Detecting Anything Overlooked in Semantic Segmentation

Dissertation
Faculty 4 - Mathematics and Computer Science

BERGISCHE UNIVERSITÄT WUPPERTAL

submitted by

Robin Chan

for the degree of Doctor of Natural Sciences (Dr. rer. nat.)

Supervisor: Prof. Dr. Hanno Gottschalk
Co-Supervisor: Dr. Matthias Rottmann

Wuppertal, February 8, 2022

The PhD thesis can be quoted as follows:

urn:nbn:de:hbz:468-20220512-123826-8
[http://nbn-resolving.de/urn/resolver.pl?urn=urn%3Anbn%3Ade%3Ahbz%3A468-20220512-123826-8]

DOI: 10.25926/spmr-x468
[https://doi.org/10.25926/spmr-x468]

Contents

Part II: Research

Acknowledgement

First of all, I would like to thank my supervisors, Hanno Gottschalk and Matthias Rottmann, for their constant support and guidance during my whole time as doctoral student. We had many pleasant discussions due to their unique proximity to employees, which is something I really appreciate. In the last three years, we worked together on several projects, eventually jointly publishing at top-tier conferences. In this time, I have personally learned a lot, which forms an integral part of this dissertation, and I am happy to finally share the gained knowledge.

I would also like to thank my supervisors during my internship at Volkswagen Group Research, Peter Schlicht and Fabian Hüger, who actually gave me the opportunity to start as a research assistant. Their trust in me was not to be taken for granted, and that is why I am deeply grateful for it.

The research leading to the results presented in this dissertation was in part funded by Volkswagen Group Research through the contract "Maximum Likelihood and Cost-based Decision Rules in Semantic Segmentation" and by the German Federal Ministry for Economic Affairs and Energy within the project "KI Absicherung – Safe AI for Automated Driving", grant no. 19A19005R. I would like to thank all the partners for the successful cooperation.

Further, I would like to thank the whole team, of which I was part of, for creating such a great and relaxed working atmosphere. In particular, I thank Kira Maag and Pascal Colling, who hopefully enjoyed sharing the same office as much as I did and who accompanied me throughout writing this dissertation.

Lastly, I would like to thank Philipp Oberdiek, Annika Mütze, and Tobias Riedlinger for proofreading this dissertation as well as for always being available for fruitful discussions.

Abstract

Computer vision is the field that deals with enabling machines to gain an understanding of the content in digital images. This field has seen spectacular advances in recent years through the introduction of deep learning. However, deep learning models are all driven by data and as such they are ill-equipped to handle data samples from categories that they rarely or even never have previously encountered. Consequently, these models are potentially prone to overlook relevant examples.

The focus of this dissertation lies on detecting objects that initially have been overlooked in the semantic segmentation of street scenes. Semantic segmentation can be described as pixel-wise image classification problem, which is typically approached by using deep convolutional neural networks (CNNs). These models are usually trained on a closed set of predefined object classes. This procedure, however, could be a source of two types of classification failures. On the one hand, CNNs might overfit their training data, which possibly results in an undesirable bias towards predicting object classes that dominate the dataset in terms of frequency of occurrence. Methods that properly cope with the imbalance problem is crucial in order to prevent the non-detection of naturally underrepresented but particularly relevant instances, such as humans in street scenes. On the other hand, CNNs in real-world applications have to face diversity of the real-world, which necessarily includes previously unseen and semantically unknown object types, *i.e.* objects from categories on which a model is not trained and thus unable to operate reliably. Detecting and localizing such unknown instances is extremely safety critical in high-stake applications relying on semantic segmentation, especially in automated driving, if they appear as obstacles on the road ahead of the self-driving car.

In this work, we present methods addressing both shortcomings. Regarding the problem of class imbalance, we investigate the concept of cost-based decision rules in CNNs and aim to make their ethical difficulties transparent. Furthermore, we introduce and extensively inspect the maximum likelihood decision rule for the reduction of false negative instances of humans as underrepresented class in the semantic segmentation of street scenes. We additionally combine this approach with methods for false positive detection based on aggregated pixel-wise uncertainty metrics. This yields a novel final decision rule, which significantly improves the trade-off between false negative and false positive error rates with respect to infrequent classes. Concerning the problem of unknown objects, we examine their pixel-level identification by thresholding on the softmax entropy as measure for prediction uncertainty of CNNs. We subsequently present a retraining approach using a multi objective loss function that maximizes the entropy of semantic segmentation models on unknown objects while allowing only a marginal drop in original semantic segmentation performance. To this end, we utilize training samples extracted from a large database consisting of images with everyday objects. Compared to pure entropy thresholding, we achieve a clear gain in effectiveness at detecting unknown obstacles on roads, which is additionally enhanced by again combining with methods for false positive detection. Lastly, we create a public benchmark for the pixel-level identification of unknown objects. On two new datasets, consisting of real-world and pixel-annotated images, we provide an in-depth evaluation of multiple state-of-the-art methods using established as well as novel and practically relevant performance metrics.

Introduction

Deep learning, which is considered as a form of artificial intelligence, comprises a set of algorithms that enable computers to learn from data. Given these algorithms, scientists have recently achieved remarkable progress in various research fields. In this light, neural networks [McC43; Ros61; Rum86; Kri12] constitute the basis of deep learning. They form a class of models whose architecture is inspired by the structure of the human brain. This model type can be described as a directed acyclic graph with nodes representing neurons and edges representing interactions between neurons. Each node consists of a computing unit passing incoming signals as linear combination through a non-linear scalar function. Hence, neural networks receive some data as input and predict some numerical values accordingly as output. By feeding them with observational data and by allowing them to adjust their model parameters, neural networks are capable of being trained to recognize complex patterns. In theory [Cyb89; Hor91; Les93] as well as in practice [Den09; Kri09; LeC10], neural networks have long been known to yield highly expressive models. Due to the development of novel network architectures, more efficient training algorithms, and particularly the increased computing power as well as the amount of labeled data over the time, deep learning has become the dominating approach for complex applications like computer vision, which is the field that deals with enabling machines to automatically gain an high-level understanding of the content shown in digital images. Recent success includes large-scale vision tasks such as image classification [Kri12; He16b; Sze16; Zop18; Pha21], object detection [Gir15; Red16; He17; Mis20; Liu21], and semantic segmentation [Lon15; Zha17; Che18b; Yua20].

Semantic Segmentation. In this context, semantic segmentation is the task of assigning a category to each pixel of an image. To this end, neural networks are used as statistical models to estimate the probability that a given pixel corresponds to a certain predefined class. These models are all driven by data and require large amount thereof to be trained properly. For this reason, they are ill-equipped to handle data samples from classes that they rarely or even never encounter during training. As a consequence, when deployed to high-stake applications, such as medical diagnostics or automated driving, deep learning models might be prone to overlook examples from rare and unknown classes. Errors of that kind could possibly lead to fatal consequences and therefore should be prevented. The current main research trend in deep learning, however, lies on developing novel model architectures and adapt them to ever more complex datasets [Gei12; Lin14; Men15; Cor16; Yu20]. Even though improving general model performance is important, addressing the mentioned issues is important as well in order ensure safe and reasonable usage of deep learning in real-world applications.

Class Imbalance. The problem of rare examples existing in classification datasets is often associated with the term class imbalance. In other words, class imbalance occurs if at least one class has significantly less examples than another class [Jap02; Kra16; Joh19]. Neural networks as statistical models are trained to learn the underlying distribution of data. This particularly includes the low

prior probabilities of infrequent classes when trained on imbalanced datasets. In extreme cases, instances from such minority classes may be completely ignored if models exhibit an extremely strong bias towards majority classes. Moreover, minority classes are often of special interest in many practical applications, *e.g.* rare but serious diseases in medical diagnostics, which intuitively increases the severity of the mentioned kind of classification failure.

However, inaccurate predictions with respect to an underrepresented class are not always identified since performance metrics for classification may be misleading. For instance, neural networks are typically trained to minimize the overall risk of an incorrect prediction, hence the inference of neural networks minimizes the total number of errors. This is commonly evaluated using the accuracy measure [Faw06; ShS14; Joh19], which takes all errors into account irrespective of the type of error. Therefore, this metric is highly biased towards overrepresented classes. Yet only changing the performance metric does have an impact on a model's actual classification capability. In general, classification models, including neural networks, have empirically shown to be detrimentally affected by class imbalance regarding their performance on minority classes [Jap00; Lóp13; Kra16; Bud18]. Thus, neural networks for semantic segmentation, which can be seen as a pixel-wise classification problem, are likely to also exhibit the same set of problems in presence of imbalanced datasets.

On the one hand, existing approaches to handle class imbalance are based on data sampling techniques. These techniques are applied directly to a dataset with the goal to balance its class distribution. Naïve but commonly applied methods are random oversampling [Cha04; Lóp13; Mas15] and random undersampling [Van07; Joh19]. In this context, data of minority classes are randomly duplicated or, respectively, data of majority classes are randomly removed. A more sophisticated method to oversampling consists of generating synthetic examples of minority classes [Cha02], whereas advanced techniques to undersampling search for redundant instances of majority classes to discard from the dataset [Wil72; Zha03]. On the other hand, algorithm based techniques aim at class balancing by incorporating costs for different types of classification mistakes during a model's training phase [Cae15; Wan16; Bul17]. Instead of minimizing the absolute number of misclassifications, the averaged misclassification costs are minimized. Such cost assignments are not necessarily required to be predefined as they can also be learned as additional model parameters [Zha16; Kha18]. Although class imbalance is a well known problem in machine learning, the outlined works indicate that only little work has been done to examine and handle this issue, particularly in deep learning, despite its practical relevance.

Anomaly Segmentation. Neural networks in real-world applications are likely exposed to data from classes that are unknown to them and therefore beyond their capabilities to process reliably. In deep learning terminology, such examples are considered as anomalies since they deviate from anything that is regarded as common and assumed to be known given the training dataset. In this respect, neural networks for semantic segmentation are usually trained with an implicit closed world assumption, *i.e.* they are designed to operate on a closed and predefined set of semantic categories. These models lack a reject option and fail on semantically unknown instances by design since they must predict a class for any instance, regardless whether the corresponding true class label is contained in the learnable semantic space or not. Naturally, reliably identifying such unknown objects is a crucial prerequisite in safety critical applications, *e.g.* consider the detection of children's toys as anomalies on the street in the setting of automated driving.

The detection of anomalies requires neural networks to be equipped with additional techniques. This problem has been mostly tackled in the context of image classification, *i.e.* identifying whether whole images are anomalous. The introduced methods aim at adjusting the model confidence scores on anomalous inputs [Hen17; Lee18; Lia18; Hei19; Mei20] such that low values indicate the presence of an anomaly. Obviously, images could also be anomalous only in certain parts, *e.g.* if an unknown object

appears in a familiar surrounding. This problem is commonly referred to as anomaly segmentation, where each individual pixel has to be classified as anomaly or non-anomaly. Methods for image-level anomaly detection can straightforwardly be adapted to anomaly segmentation. However, this approach has shown to scale poorly to pixel-wise anomaly detection, mainly due to imprecise localization of anomalous objects [Ang19; Blu21]. More dedicated approaches to anomaly segmentation mitigate this localization issue by incorporating context information of the given scene into their methods. Among those, many solely rely on uncertainty estimates [Gal16; Ken17; Lak17; Muk19; Gus20; Jun21] since anomalies are intuitively assumed to correlate with high prediction uncertainty. Another popular line of work consists of tuning semantic segmentation models to anomaly segmentation. This can be accomplished by either adding a separate branch to the semantic segmentation model for anomaly prediction [DeV18; Lis19; Di 21] or by enforcing a specific model output on anomalous inputs using auxiliary datasets [Bev19; Hen19; Jou20; Mei20].

Despite these recent efforts in anomaly segmentation, in general only few scientific works exist. This is also a consequence of the clear shortage of proper datasets with realistic and safety relevant scenes. Existing datasets either rely on synthetically generated anomalies [Hen20; Blu21] or suffer from insufficient labeling quality [Pin16; Lis19], besides all providing only a limited variety of anomaly types. Furthermore, missing benchmarks on proper anomaly segmentation datasets to reliably compare methods additionally hinder progress in this research direction.

Contributions. In this dissertation, we focus on the detection of objects that have initially been overlooked in the semantic segmentation of street scenes since the corresponding object classes are rare or unknown to the model. The detection of rare objects corresponds to the problem of class imbalance, while the detection of unknown objects corresponds to anomaly segmentation. In both research directions there is a shortage of scientific works, particularly in the context of deep learning. However, these two issues are of special interest in order to advance towards safe deployment of semantic segmentation to systems for automated driving.

For instance, humans typically correspond to a minority class in street scene datasets in comparison to the class street. The class human appears less frequently besides comprising significantly less image pixels due to their smaller object size. Intuitively, ignoring humans in favor of predicting street as a consequence of class imbalance is highly undesirable. In this regard, existing class balancing methods are difficult to apply since class imbalance is often inherent in real-world datasets. Any change to the training data or modification to the training procedure could yield an unnatural bias compromising the overall performance of semantic segmentation models.

One efficient approach to class balancing, which circumvents the latter issue, relates to the classification decision rule incorporated in neural networks for semantic segmentation. Given the probabilistic output of neural networks, the standard maximum a-posteriori probability decision rule selects the class as final prediction that has been assigned the highest estimated probability. The sensitivity towards predicting minority classes can easily be increased by introducing misclassification costs. We reveal, however, that this approach includes serious ethical difficulties when it comes to providing explicit cost values [Cha19]. As an alternative and mathematically natural way of decision making, we introduce the maximum likelihood decision rule for semantic segmentation [Cha20a], which differs from the standard maximum a-posteriori probability principle by taking the likelihoods into account instead of the posterior class probabilities. In other words, the maximum likelihood principle selects the class as final prediction for which observed features are most typical without including any prior belief about class occurrence frequencies. As light-weight post-processing techniques, multiple decision rules can even be applied simultaneously during inference with negligible computational overhead. By comparing the final semantic segmentation masks, we further aim to identify false indications, which

are produced due to an increased prediction sensitivity induced by a decision rule. To this end, we construct metrics based on geometry properties and location information of predicted objects as well as dispersion measures derived from the probabilistic output of neural networks [Rot20]. Using these hand-crafted metrics, we first identify and then remove false predictions of minority class objects that are produced by using the maximum likelihood in place of the maximum a-posteriori probability decision rule. This yields a sophisticated alternative for the final class prediction, which significantly reduces the non-detection of humans while at the same time controlling the amount of false indications thereof in the semantic segmentation of street scenes [Cha20b].

Another critical scenario in automated driving occurs if unknown objects appear on the road ahead of the vehicle. As already stated, neural networks for semantic segmentation are trained on a closed set of object classes that is assumed to be sufficient for perceiving most street scenes. However, when deployed to the real world, these models necessarily also have to face the diversity of the real world with a boundless set of possible object types. From a functional safety point of view, anomaly segmentation is therefore mandatory in order to determine whether neural networks for semantic segmentation are operating out of their proper domain. Existing methods in this direction are either computationally too expensive for safe real-time semantic segmentation or they are too dataset specific such that they fail to generalize at detecting arbitrary types of anomaly objects.

We address both mentioned shortcomings by enforcing semantic segmentation models to output high prediction uncertainty on anomalous inputs. To quantify prediction uncertainty, we rely on the intuitive entropy measure, which can be easily computed given the probabilistic output of neural networks. In a first step, we study the effectiveness of the entropy measure at detecting unknown obstacles in street scenes and generally observe high entropy responses on those anomaly objects [Brü20]. At the same time, we realize that this uncertainty measure also yields a substantial amount of false anomaly indications, particularly on the road if different road surfaces are shown. To improve the separability between anomalies and non-anomalies, we deliberately include unknown objects in a retraining process of semantic segmentation models and employ a modified training objective to train for high entropy on those induced examples [Cha21a]. To this end, these known unknowns are randomly sampled from a large collection of images showing everyday objects in everyday scenes. In this manner, we obtain an anomaly segmentation model, which generalizes learned uncertainty concepts to truly unseen anomalies without significantly sacrificing in original performance on the primary task of semantic segmentation. Our approach outperforms many other established state-of-the-art methods in terms of effectiveness but particularly in computation time. Furthermore, we equip our anomaly segmentation method with a post-processing step to remove false anomaly indications. This additional step is based on the same set of hand-crafted metrics that is also used in our class balancing method. Again, these metrics reliably indicate false predictions, which we then remove to further enhance anomaly segmentation performance.

Motivated by the shortage and limitations of existing datasets in this research direction, we further introduce an anomaly segmentation benchmark, which is publicly available and accompanied with two new real-world datasets [Cha21b]. Our benchmark provides a public leader board and an evaluation suite including established as well as novel performance metrics. In our datasets, we included images showing anomalies in safety critical scenes and we made sure that the anomalies are clearly defined as well as properly labeled. Moreover, our datasets consist of high-quality images with real anomalies from a wide variety of anomaly types, which have not previously existed in this form.

Outline. The remainder of this dissertation is structured as follows: in Chapter 2, we recapitulate the basic theory on neural networks with a special focus on classification problems. We introduce the building blocks of neural networks, how these models are trained, and how they are adapted to image classification. In Chapter 3, we give an overview on the current research status in the field of semantic segmentation. This encompasses possible real-world applications relying on semantic segmentation and also their corresponding datasets. Furthermore, we review the evolution of state-of-the-art neural network architectures, which consistently improved the state-of-the-art in semantic segmentation over the time. In Chapter 4 and Chapter 5, we provide brief summaries of our works on handling class imbalance and on anomaly segmentation, respectively. This dissertation is cumulative comprising the following seven peer reviewed conference articles, which all appear in their full form including the description of own contributions in Chapter 6:

> Robin Chan, Matthias Rottmann, Radin Dardashti, Fabian Hüger, Peter Schlicht and Hanno Gottschalk, *"The Ethical Dilemma when (not) Setting up Cost-based Decision Rules in Semantic Segmentation"* in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops, Safe Artificial Intelligence for Automated Driving (SAIAD)* © 2019

> Robin Chan, Matthias Rottmann, Fabian Hüger, Peter Schlicht and Hanno Gottschalk, *"Application of Decision Rules for Handling Class Imbalance in Semantic Segmentation"* in *The 30th European Safety and Reliability Conference (ESREL)* © 2020

> Matthias Rottmann, Pascal Colling, Thomas Paul Hack, Robin Chan, Fabian Hüger, Peter Schlicht and Hanno Gottschalk, *"Prediction Error Meta Classification in Semantic Segmentation: Detection via Aggregated Dispersion Measures of Softmax Probabilities"* in *The IEEE International Joint Conference on Neural Networks (IJCNN)* © 2020

> Robin Chan, Matthias Rottmann, Fabian Hüger, Peter Schlicht and Hanno Gottschalk, *"Controlled False Negative Reduction of Minority Classes in Semantic Segmentation"* in *The IEEE International Joint Conference on Neural Networks (IJCNN)* © 2020

> Dominik Brüggemann, Robin Chan, Matthias Rottmann, Hanno Gottschalk and Stefan Bracke, *"Detecting Out-of-Distribution Objects in Semantic Segmentation of Street Scenes"* in *The 30th European Safety and Reliability Conference (ESREL)* © 2020

> Robin Chan, Matthias Rottmann and Hanno Gottschalk, *"Entropy Maximization and Meta Classification for Out-of-Distribution Detection in Semantic Segmentation"* in *The IEEE/CVF International Conference on Computer Vision (ICCV)* © 2021

> Robin Chan, Krzysztof Lis, Svenja Uhlemeyer, Hermann Blum, Sina Honari, Roland Siegwart, Pascal Fua, Mathieu Salzmann and Matthias Rottmann, *"SegmentMeIfYouCan: A Benchmark for Anomaly Segmentation"*, *Thirty-fifth Conference on Neural Information Processing Systems (NeurIPS) Datasets and Benchmarks Track* © 2021

Finally, we conclude this dissertation in Chapter 7 by pointing out the main contributions of the above articles and giving an outlook for future work directions.

Theoretical Foundation

Deep learning is currently the best approach to tackle many applications such as computer vision, speech recognition or natural language processing. The algorithms solving the mentioned tasks are often regarded as kind of artificial intelligence. They mostly encompass artificial neural networks, which are biologically-inspired statistical and parameterized models. Then, by means of observational data and optimization algorithms, machines are enabled to automatically find their appropriate model parameters in order to solve highly complex tasks. In other words, machines "learn" from experience provided by data. In this chapter, we introduce the theoretical foundations behind deep learning techniques and motivate why they work so well in practice, with special focus on neural networks specifically designed for classification problems on image data.

2.1 Feedforward Neural Networks

The field of deep learning incorporates neural networks (NNs) as the most commonly used type of model. In general, NNs for classification can be understood as statistical models to approximate some probability function p, that is estimating the probability $p(y|\mathbf{x})$ for an input $\mathbf{x}$ to have class affiliation y. In this context, so-called *feedforward neural networks* are one particular type of model. They are called feedforward as the information flows in one direction, from the input $\mathbf{x}$, through some intermediate computations, and finally to the output of the model $f(\mathbf{x})$. Then, a feedforward NN for classification is an input-target mapping, where $f_y(\mathbf{x}|\theta)$ is a parameterized approximation function of the target function $p(y|\mathbf{x})$. The design of feedforward NNs enables them to automatically and efficiently "learn" its appropriate model parameters θ. This is accomplished via optimization algorithms with the objective that the approximation fits the target as well as possible.

2.1.1 Perceptron Neuron

A feedforward neural network in its simplest form consists of one neuron, the *perceptron* neuron. The latter model is a binary classifier consisting of a single computing unit, which processes multiple scalar inputs and returns a binary output, see also Figure 2.1.

Definition 2.1. *Perceptron Neuron [McC43; Ros58]*
Let $\mathbf{x} \in \mathbb{R}^n$, $n \in \mathbb{N}$ be an input vector. A perceptron neuron is a mapping $f : \mathbb{R}^n \to \{0, 1\}$,

$$f(\mathbf{x}) := \begin{cases} 1 & , \text{ if } \quad \mathbf{w}^\top \mathbf{x} + b > 0 \\ 0 & , \text{ otherwise} \end{cases} \tag{2.1}$$

where $\mathbf{w} \in \mathbb{R}^n$ is the weights vector and $b \in \mathbb{R}$ the bias.

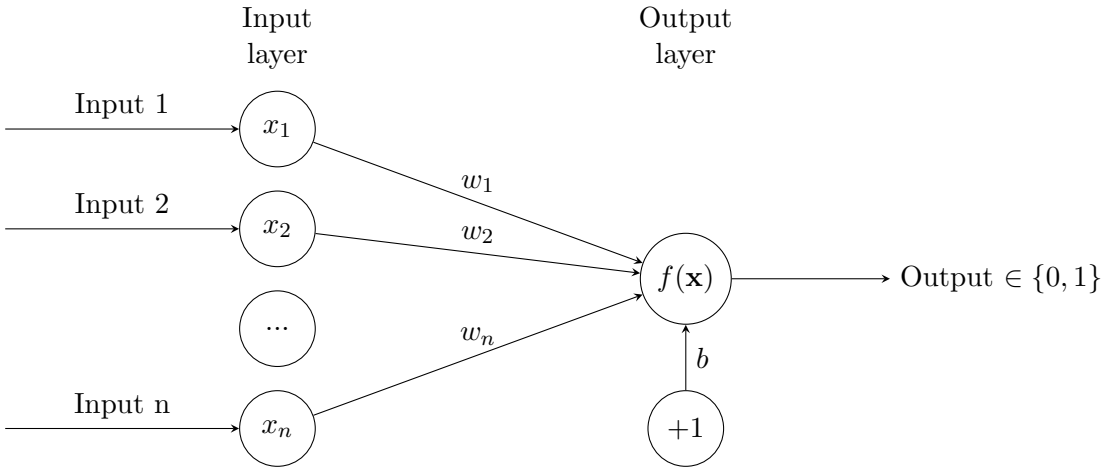


Figure 2.1: A feedforward neural network as perceptron neuron. Here, $\mathbf{x} = (x_1, \ldots, x_n)^\top$ denotes the input vector, $\mathbf{w} = (w_1, \ldots, w_n)^\top$ the weights vector and b the bias.

A perceptron neuron can also be understood as a threshold gate for the sum of weighted inputs $\mathbf{w}^\top \mathbf{x}$, given the threshold $-b$. By adjusting the parameters in the set $\theta = \{\mathbf{w}^\top, b\}$, this simple perceptron model is capable of solving any binary classification problem with data that is linearly separable [Bis06; Has09; ShS14].

2.1.2 Sigmoid Neuron

Perceptron neurons provide two output states, either 0 or 1, see Equation (2.1). Thus, small changes in the weights might cause the output to completely flip, which is obviously undesirable when tuning the weight parameters. The *sigmoid neuron* avoids this problem by coupling perceptrons with the logistic sigmoid function as an additional computation step.

Definition 2.2. *Sigmoid Neuron [Has09; Nie17; Cal20]*
Let $\mathbf{x} \in \mathbb{R}^n, n \in \mathbb{N}$ be an input vector and let $\sigma : \mathbb{R} \to (0, 1)$, $\sigma(z) = 1/(1 + e^{-z})$ denote the logistic sigmoid function. A sigmoid neuron is a mapping $f : \mathbb{R}^n \to (0, 1)$,

$$f(\mathbf{x}) := \sigma(\mathbf{w}^\top \mathbf{x} + b) = \frac{1}{1 + e^{-\mathbf{w}^\top \mathbf{x} - b}} \tag{2.2}$$

where $\mathbf{w} \in \mathbb{R}^n$ is the weights vector and $b \in \mathbb{R}$ the bias.

As the logistic sigmoid function σ is continuous with respect to the parameters $\theta = \{\mathbf{w}^\top, b\}$, the output of a sigmoid neuron does not have a jump discontinuity as opposed in the perceptron neuron. This property is crucial for learning / optimization algorithms to work. Moreover, sigmoid neurons approximate perceptrons, *cf.* Figure 2.2, therefore they are capable of solving any problem with linearly separable data as well. Although the limitation of only solving linear classification problems still remains, the sigmoid neuron output provides a smoother decision boundary than the perceptron neuron. This continuous output can be interpreted as confidence of taking a certain decision, thus allowing for a more adequate treatment of examples close to the decision boundary.

2.1.3 Multi Layer Neural Network

In order to handle more complex tasks than binary classification, the single neuron models have been extended to *multi layer neural networks*. This latter model type is a feedforward neural network that

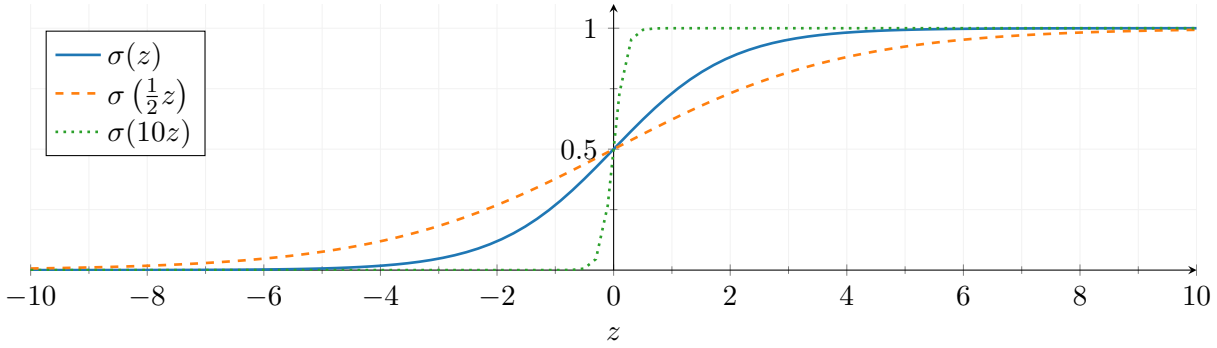


Figure 2.2: Plot of the logistic sigmoid function $\sigma(z) = 1/(1+e^{-z})$. As comparison, the scaled logistic sigmoid function $\sigma(sz) = 1/(1+e^{-sz})$ with scale parameters $s = 1/4$ and $s = 10$ are included as well. The larger the scale parameter, the more the logistic sigmoid function amounts to a unit step function at $z = 0$ (which is used in perceptron neurons).

is composed of multiple layers of neurons. The layers are linked together in the feedforward sense such that the output of neurons in one layer form the input of neurons in the next one. In this regard, layers residing in-between the network input and network output layer are called *hidden* since their inputs and outputs are not directly observed. Typically, the neurons in the hidden layer perform non-linear transformations to the input $\mathbf{x}$, *e.g.* by employing sigmoid neurons, *cf.* Definition 2.2. In this way, multi layer neural networks are capable of distinguishing non-linearly separable data, thus overcoming a major limitation of single neuron models.

Definition 2.3. *Output per layer [Has09; Nie17; Cal20]*
Let $n_\ell \in \mathbb{N}$ denote the number of neurons in layer $\ell \in \mathbb{N}$ of a feedforward neural network. Then, the output for layer ℓ is defined as

$$\begin{aligned} f^{(\ell)} : \mathbb{R}^{n_{\ell-1}} &\to \mathbb{R}^{n_\ell} \\ \mathbf{x}^{(\ell-1)} &\mapsto \mathbf{x}^{(\ell)} \end{aligned}$$

$$\mathbf{x}^{(\ell)} = f^{(\ell)}(\mathbf{x}^{(\ell-1)}) := \boldsymbol{\phi}^{(\ell)}\left(\mathbf{W}^{(\ell)}\mathbf{x}^{(\ell-1)} + \mathbf{b}^{(\ell)}\right) \;\forall\, \ell \in \mathbb{N}, \ell > 0 \quad \textit{and} \quad \mathbf{x}^{(0)} = \mathbf{x} \in \mathbb{R}^n \;, \tag{2.3}$$

where $\boldsymbol{\phi}^{(\ell)} : \mathbb{R}^{n_\ell} \to \mathbb{R}^{n_\ell}$ is a transformation function, $\mathbf{W}^{(\ell)} \in \mathbb{R}^{n_\ell \times n_{\ell-1}}$ the weights matrix and $\mathbf{b}^{(\ell)} \in \mathbb{R}^{n_\ell}$ the biases vector for layer ℓ.

Remark 2.4. Note, that in practice $\boldsymbol{\phi}^{(\ell)}$ consists of non-linear component-wise operations. More precisely, $x_j^{(\ell)} = \phi_j^{(\ell)}(\mathbf{W}_j^{(\ell)}\mathbf{x}^{(\ell-1)} + \mathbf{b}_j^{(\ell)})$ with $\phi_j^{(\ell)} : \mathbb{R} \to \mathbb{R}\ \forall\ j \in \{1, \ldots, n_\ell\}$. The logistic sigmoid function $\sigma(z) = 1/(1+e^{-z})$, see Equation (2.2), is one such non-linear transformation.

Definition 2.5. *Multi Layer Neural Network [ShS14; Goo16; Dei20]*
Let $\mathbf{x} \in \mathbb{R}^n, n \in \mathbb{N}$ be an input vector and let $f^{(\ell)}(\cdot|\theta^{(\ell)})$ denote the output per layer function with parameters $\theta^{(\ell)} = \{\mathbf{W}^{(\ell)}, \mathbf{b}^{(\ell)}\}$ for $\ell \in \{1, 2, \ldots, L\}, L \in \mathbb{N}, L \geq 2$, as defined in Definition 2.3. Then, a multi layer neural network is a mapping $f : \mathbb{R}^n \to \mathbb{R}^K, K \in \mathbb{N}$ which is composed of a chain of output per layer functions

$$f(\mathbf{x}|\theta) := \left(f^{(L)} \circ f^{(L-1)} \circ \ldots \circ f^{(1)}\right)(\mathbf{x}) \tag{2.4}$$

where $\ell = 0$ denotes the layer of the network input, $\ell = L$ the layer of the network output, and $\theta = \{\theta^{(\ell)}\}_{\ell=1}^{L}$ the set of model parameters.

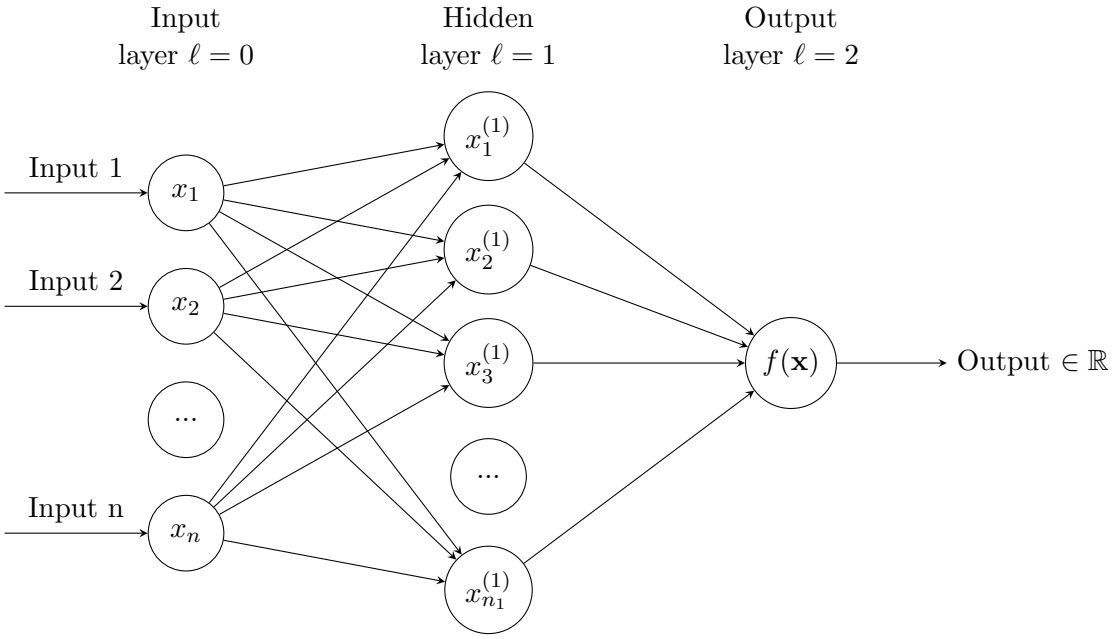


Figure 2.3: A vanilla neural network, *i.e.* a multi layer neural network with *one* hidden layer and 1-dimensional output ($K = 1$). In the layer $\ell = 1$, the corresponding neurons are coupled with the logistic sigmoid function $\sigma(z) = \frac{1}{1+e^{-z}}$ as transformation step $\boldsymbol{\phi}^{(1)}$ and in the final layer $\ell = 2$, the identity function $\mathrm{id}(z) = z$ is employed as $\boldsymbol{\phi}^{(2)}$. For the sake of clarity, the weights are omitted in this illustration. With the notation introduced in Definition 2.3, the network output is equivalent to $f(\mathbf{x}) = x^{(2)}$.

The most commonly-known type of multi layer neural networks is the so-called "vanilla" neural network [Has09]. With Definition 2.5, this model can be understood as a single hidden layer NN with $L = 2, K = 1$ where $\boldsymbol{\phi}$ is the sigmoid function in $f^{(1)}$ and the identity function in $f^{(2)}$, respectively, *cf.* Figure 2.3. This particular architecture is a powerful approximator for a large family of real-life functions. In fact, a vanilla NN are so-called *universal approximator* [Cyb89] (see also next Section 2.1.5). The vanilla NN is also the type of NN for which the universal approximation property has been determined first. Consequently, this model type has a certain historical importance in the development of deep learning as it inspired a series of other NN architectures having the universal approximation property as well but employing different transformation functions for $\boldsymbol{\phi}^{(1)}$.

2.1.4 Activation Functions

In each neuron of a neural network, the computation of the output consists of two steps. First, the incoming inputs are weighted and summed up. As second step, a function performing transformations to the weighted sum of inputs is applied. In Definition 2.3, this function is denoted by $\boldsymbol{\phi}$, which is also commonly referred to as *activation function*. From the single neuron models, we already know some of their types. These are for instance the Heavyside function, *i.e.*

$$\phi(z) = \mathbb{1}_{\{z>0\}}, \ z \in \mathbb{R} \ , \tag{2.5}$$

which is used in the perceptron neuron (Definition 2.1), and the logistic sigmoid function, *i.e.*

$$\phi(z) = \frac{1}{1+e^{-z}}, \ z \in \mathbb{R} \ , \tag{2.6}$$

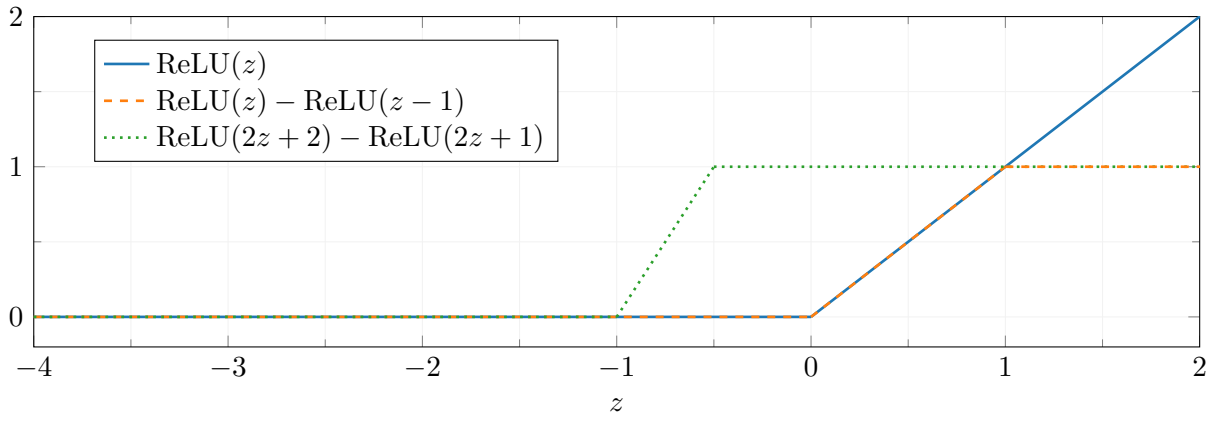


Figure 2.4: Plot of the rectified linear unit activation function $\mathrm{ReLU}(z) = \max\{0, z\}$ (blue line). Additionally, the difference of two (scaled and shifted) ReLU functions are included.

which is used in the sigmoid neuron (Definition 2.2), respectively.

As neural networks are tuned via gradient-based optimization algorithms, the activation functions should ideally be differentiable. Other desired properties, besides non-linearity and differentiability, is computationally cheap derivatives. In practice, the most commonly-used activation function is the so-called *rectified linear unit activation function.*

Definition 2.6. *Rectified Linear Unit Neuron [Nai10; Maa13; Goo16]*
Let $\mathbf{x} \in \mathbb{R}^n, n \in \mathbb{N}$ *be an input vector. A rectified linear unit (ReLU) neuron is a mapping* $f : \mathbb{R}^n \to \mathbb{R}$,

$$f(\mathbf{x}) := \mathrm{ReLU}(\mathbf{w}^\top \mathbf{x} + b) \tag{2.7}$$

where $\mathbf{w} \in \mathbb{R}^n$ *is the weights vector,* $b \in \mathbb{R}$ *the bias and* $\mathrm{ReLU} : \mathbb{R} \to \mathbb{R}$ *the ReLU activation function. The latter function is defined as the positive part of its inputs,* i.e.

$$\mathrm{ReLU}(z) := \max\{0, z\}, \quad z \in \mathbb{R} \ . \tag{2.8}$$

Remark 2.7. Considered strictly, the ReLU activation function is not differentiable at every point $z \in \mathbb{R}$ since for $\delta > 0$ sufficiently small

$$\lim_{\delta \to 0} \lim_{z \nearrow 0} \frac{\max\{0, z+\delta\} - \max\{0, z\}}{\delta} = 0 \neq 1 = \lim_{\delta \to 0} \lim_{z \searrow 0} \frac{\max\{0, z+\delta\} - \max\{0, z\}}{\delta} \ , \tag{2.9}$$

see Figure 2.4 blue graph. In implementations of neural networks, the derivative ReLU' is therefore set to 0 at $z = 0$, *i.e.* the Heavyside function is used as derivative $\mathrm{ReLU}'(z) \approx \mathbb{1}_{\{z>0\}}$.

The ReLU activation function is non-linear although consisting of two linear pieces, see Figure 2.4. Thus, while accounting for non-linearity, the derivative of ReLU can be very easily computed. Moreover, replacing sigmoid neurons in multi layer neural networks for ReLUs yield another universal approximator, which is another reason for the popularity of the ReLU activation function (see next Section 2.1.5).

Multi layer neural networks can be extended in a straightforward manner from single-class classification to multi-class classification. In Definition 2.5, $K \in \mathbb{N}$ corresponds to the number of final network outputs, in other words it corresponds to the number of classes to be predicted. The activation function in the last layer is usually the softmax function, which provides a categorical probability distribution over all classes.

Definition 2.8. *Softmax Activation Function [Bri90; Bis06; Has09; Gib10; Goo16; Dei20]*
Let $K \in \mathbb{N},\ K \geq 2$ denote the number of network outputs, i.e. *the number of neurons in a network's last layer. Then, the softmax activation function is defined as*

$$\text{softmax} : \mathbb{R}^K \to \left\{ \mathbf{z} \in (0,1)^K : \|\mathbf{z}\|_1 := \sum_{k=1}^{K} z_k = 1 \right\}$$

$$\text{softmax}_k(\mathbf{z}) := \frac{e^{z_k}}{\sum_{j=1}^{K} e^{z_j}} \quad \forall\ k = 1, \ldots, K\ . \tag{2.10}$$

By applying the softmax function, the components of the network output $f(\mathbf{x})$ are scaled into the range $(0, 1)$ such that the sum of all outputs is equal to 1. In other words, $f(\mathbf{x})$ is normalized to a probability distribution over K predicted outputs. In this way, the k-th component $f_k(\mathbf{x})$ is oftenинterpreted as the NN's confidence in the class prediction $k \in \{1, \ldots, K\}$, *i.e.* an estimate of $p(k|\mathbf{x})$.

Note that for the final class prediction, a decision function $D : \mathbb{R}^K \to \{1, \ldots, K\}$ is applied. According to the maximum a-posteriori principle [Bis06], the final class prediction for input $\mathbf{x}$ is given by

$$D^{\text{MAP}}(\mathbf{x}) := \underset{k \in \{1,\ldots,K\}}{\arg\max}\ p(k|\mathbf{x})\ . \tag{2.11}$$

Replacing the unknown $p(k|\mathbf{x})$ by the estimate $f_k(\mathbf{x})$ then yields the final prediction $\hat{y} = \underset{k \in \{1,\ldots,K\}}{\arg\max}\ f_k(\mathbf{x})$.

2.1.5 Universal Approximators for Continuous Functions

In Section 2.1.3, we have introduced multi layer neural networks as parametric models $f(\cdot|\theta)$ and in Section 2.1.4 we have seen that they can be equipped with different activation functions. Such models possess the capability of "learning", *i.e.* approximating a target function f^* by automatically modifying their parameters θ. In this light, so-called *universal approximators* are capable of approximating any function, from a certain space of functions, to any degree of accuracy. In what follows, we will examine the expressivity of multi layer neural networks given the known activation functions.

Definition 2.9. *Universal Approximator [Cyb89; Cal20]*
Let $(\mathcal{S}, \mathrm{d})$ be a function space $\mathcal{S}$ equipped with a distance function $\mathrm{d} : \mathcal{S} \times \mathcal{S} \to \mathbb{R}$, and let $\mathcal{U}$ be the set of functions in $\mathcal{S}$ than can be represented by neural networks. Then, the class $\mathcal{U}$ of neural networks is a universal approximator for $(\mathcal{S}, \mathrm{d})$ if

$$\forall\ f^* \in \mathcal{S}\ \forall\ \varepsilon > 0\ \exists\ f \in \mathcal{U} : \mathrm{d}(f^* - f) < \varepsilon\ . \tag{2.12}$$

In other words, a class of neural networks is a universal approximator for $(\mathcal{S}, \mathrm{d})$ if the function space $\mathcal{U}$ is d-dense in $\mathcal{S}$.

Universal approximation theorems state the existence of NNs fulfilling the property in Definition 2.9. It has been proven that even single hidden layer neural networks can have the universal approximation property. The most classical form of a universal approximator is the vanilla NN for continuous functions $f^* \in C(\mathbb{I}^n)$, where $C(\mathbb{I}^n)$ denotes the space of real-valued continuous functions on the n-dimensional unit cube $\mathbb{I}^n := [0, 1]^n \subset \mathbb{R}^n$.

Theorem 2.10. *Sigmoid Neurons as Universal Approximator [Cyb89]*
Let $\mathbf{x} \in \mathbb{I}^n, n \in \mathbb{N}$ *and let* $\sigma(z) = 1/(1+e^{-z})$ *denote the logistic sigmoid function. Then, the space of functions of the form*

$$f(\mathbf{x}) = (\mathbf{w}^{(2)})^\top \boldsymbol{\sigma}\left(\mathbf{W}^{(1)}\mathbf{x} + \mathbf{b}^{(1)}\right) = \sum_{j=1}^{n_1} w_j^{(2)} \sigma\left(\mathbf{w}_j^{(1)}\mathbf{x} + b_j^{(1)}\right) \tag{2.13}$$

$$w_j^{(2)}, b_j^{(1)} \in \mathbb{R},\ (\mathbf{w}_j^{(1)})^\top \in \mathbb{R}^n,\ j = 1, \ldots, n_1 \in \mathbb{N}$$

is $\|\cdot\|_\infty$*-dense in* $C(\mathbb{I}^n)$.

Proof. For the proof we will utilize the discriminatory property of the logistic sigmoid function σ. Note that σ is one type of continuous sigmoid functions since $\sigma(z) \xrightarrow{z\to-\infty} 0$ and $\sigma(z) \xrightarrow{z\to+\infty} 1$, see Definition A.1. Then, σ is discriminatory since any arbitrary continuous sigmoid function is discriminatory, *i.e.* the zero measure $\nu = 0$ is the only finite signed Borel measure $\nu \in M(\mathbb{I}^n)$ on $\mathbb{I}^n$ with

$$\int_{\mathbb{I}^n} \sigma(\mathbf{w}^\top x + b)\, d\nu(\mathbf{x}) = 0 \quad \forall\ \mathbf{w} \in \mathbb{R}^n, b \in \mathbb{R}\ , \tag{2.14}$$

see also Definition A.2 and Proposition A.3. Let $\mathcal{U}$ be the set of functions of the form as in Equation (2.13), *i.e.*

$$\mathcal{U} = \left\{ f : f(\mathbf{x}) = \sum_{j=1}^{n_1} w_j^{(2)} \sigma\left(\mathbf{w}_j^{(1)}\mathbf{x} + b_j^{(1)}\right), \right. \\ \left. w_j^{(2)}, b_j^{(1)} \in \mathbb{R},\ (\mathbf{w}_j^{(1)})^\top \in \mathbb{R}^n,\ j = 1, \ldots, n_1 \in \mathbb{N} \right\}\ , \tag{2.15}$$

which is a linear subspace of $C(\mathbb{I}^n)$.

We show that $\mathcal{U}$ is $\|\cdot\|_\infty$-dense in $C(\mathbb{I}^n)$ by contradiction. To this end, let us assume that $\mathcal{U}$ is not $\|\cdot\|_\infty$-dense in $C(\mathbb{I}^n)$. Then, the closure $\mathcal{R}$ of $\mathcal{U}$ is a closed proper subspace of $C(\mathbb{I}^n)$. By the Hahn-Banach theorem, *cf.* Theorem A.4, there exists a bounded continuous linear functional $\mathrm{L} : C(\mathbb{I}^n) \to \mathbb{R}$, with the property that $\mathrm{L} \neq 0$ but $\mathrm{L}(f) = 0\ \forall\ f \in \mathcal{U}$. By the Riesz representation theorem, see Theorem A.5, there is a measure $\nu \in M(\mathbb{I}^n)$, such that

$$\mathrm{L}(g) = \int_{\mathbb{I}^n} g(\mathbf{x})\, d\nu(\mathbf{x}) \quad \forall\ g \in C(\mathbb{I}^n)\ . \tag{2.16}$$

In particular,

$$\mathrm{L}(f) = \int_{\mathbb{I}^n} f(\mathbf{x})\, d\nu(\mathbf{x}) = 0 \quad \forall\ f \in \mathcal{U} \tag{2.17}$$

$$\Leftrightarrow \sum_{j=1}^{n_1} w_j^{(2)} \int_{\mathbb{I}^n} \sigma\left(\mathbf{w}_j^{(1)}\mathbf{x} + b_j^{(1)}\right)\, d\nu(\mathbf{x}) = 0 \quad \forall\ w_j^{(2)}, b_j^{(1)} \in \mathbb{R}, (\mathbf{w}_j^{(1)})^\top \in \mathbb{R}^n, \tag{2.18}$$

see also Lemma A.6. Since $\sigma(\mathbf{w}^\top\mathbf{x} + b) \in \mathcal{R}\ \forall\ \mathbf{w} \in \mathbb{R}^n, b \in \mathbb{R}$, we have

$$\int_{\mathbb{I}^n} \sigma\left(\mathbf{w}^\top\mathbf{x} + b\right)\, d\nu(\mathbf{x}) = 0 \quad \forall\ \mathbf{w} \in \mathbb{R}^n, b \in \mathbb{R}, \tag{2.19}$$

which means that $\nu = 0$ according to the discriminatory property of σ. However, $\nu = 0$ being the zero measure would imply $\mathrm{L}(g) = 0\ \forall\ g \in C(\mathbb{I}^n)$, contradicting Equation (2.16) and therefore also our assumption. Hence, we conclude that $\mathcal{U}$ is $\|\cdot\|_\infty$-dense in $C(\mathbb{I}^n)$. □

As the output of a vanilla neural network is of the form as in Equation (2.13), a vanilla NN consequently is a universal approximator. Summarizing Theorem 2.10, the statement is that no matter what continuous function we wish to be approximate, there is a vanilla NN whose output is arbitrarily close. In practice, vanilla neural networks are not the first choice of network design, inter alia due to the logistic sigmoid function as activation function. In this context, the most popular activation function is ReLU. While replacing the sigmoid neurons with ReLU neurons in vanilla NNs, the universal approximation property still remains.

Theorem 2.11. *ReLU Neurons as Universal Approximator [Hor91; Gui18]*
Let $\mathbf{x} \in \mathbb{I}^n, n \in \mathbb{N}$ *and let* $\mathrm{ReLU}(z) = \max\{0, z\}$ *denote the ReLU activation function. Then, the space of functions of the form*

$$f(\mathbf{x}) = (\mathbf{w}^{(2)})^\top \mathbf{ReLU}\left(\mathbf{W}^{(1)}\mathbf{x} + \mathbf{b}^{(1)}\right) = \sum_{j=1}^{n_1} w_j^{(2)} \mathrm{ReLU}\left(\mathbf{w}_j^{(1)}\mathbf{x} + b_j^{(1)}\right) \tag{2.20}$$

$$w_j^{(2)}, b_j^{(1)} \in \mathbb{R},\ (\mathbf{w}_j^{(1)})^\top \in \mathbb{R}^n,\ j = 1, \dots, n_1 \in \mathbb{N}$$

is $\|\cdot\|_\infty$*-dense in* $C(\mathbb{I}^n)$.

Proof. According to the proof of Theorem 2.10, it is sufficient for us to show that ReLU is discriminatory for $\nu \in M(\mathbb{I}^n)$, *i.e.* we show that

$$\int_{\mathbb{I}^n} \mathrm{ReLU}(\mathbf{w}^\top \mathbf{x} + b)\, d\nu(\mathbf{x}) = 0 \quad \forall\ \mathbf{w} \in \mathbb{R}^n, b \in \mathbb{R} \implies \nu = 0\ . \tag{2.21}$$

Consider the continuous sigmoid function of the form

$$\psi(z) = \begin{cases} 0 & , \text{ if } \quad z < 0 \\ z & , \text{ if } \quad 0 \le z \le 1 \\ 1 & , \text{ if } \quad z > 1 \end{cases}, \quad z \in \mathbb{R}\ . \tag{2.22}$$

This activation function ψ can be described as the difference of two ReLU activation functions, *cf.* Figure 2.4 green dotted graph,

$$\psi(z) = \mathrm{ReLU}\,(z) - \mathrm{ReLU}\,(z-1) \qquad \forall\ z \in \mathbb{R} \tag{2.23}$$

$$\Leftrightarrow \quad \psi(\mathbf{w}^\top \mathbf{x} + b) = \mathrm{ReLU}\left(\mathbf{w}^\top \mathbf{x} + b\right) - \mathrm{ReLU}\left(\mathbf{w}^\top \mathbf{x} + b - 1\right) \qquad \forall\ \mathbf{x}, \mathbf{w} \in \mathbb{R}^n, b \in \mathbb{R}\ . \tag{2.24}$$

Since any continuous sigmoid function is discriminatory, see Proposition A.3, we have

$$\int_{\mathbb{I}^n} \psi(\mathbf{w}^\top \mathbf{x} + b)\, d\nu(\mathbf{x}) = 0 \quad \forall\ \mathbf{w} \in \mathbb{R}^n, b \in \mathbb{R} \implies \nu = 0\ , \tag{2.25}$$

which yields

$$\int_{\mathbb{I}^n} \psi(\mathbf{w}^\top \mathbf{x} + b)\, d\nu(\mathbf{x}) \tag{2.26}$$

$$= \int_{\mathbb{I}^n} \left(\mathrm{ReLU}\left(\mathbf{w}^\top \mathbf{x} + b\right) - \mathrm{ReLU}\left(\mathbf{w}^\top \mathbf{x} + b - 1\right)\right) d\nu(\mathbf{x}) \tag{2.27}$$

$$= \int_{\mathbb{I}^n} \mathrm{ReLU}\left(\mathbf{w}^\top \mathbf{x} + b\right)\, d\nu(\mathbf{x}) - \int_{\mathbb{I}^n} \mathrm{ReLU}\left(\mathbf{w}^\top \mathbf{x} + b - 1\right)\, d\nu(\mathbf{x}) \tag{2.28}$$

$$= 0 \quad \forall\ \mathbf{w} \in \mathbb{R}^n, b \in \mathbb{R} \implies \nu = 0 \tag{2.29}$$

$$\Leftrightarrow \int_{\mathbb{I}^n} \mathrm{ReLU}\left(\mathbf{w}^\top \mathbf{x} + b\right)\, d\nu(\mathbf{x}) = 0 \quad \forall\ \mathbf{w} \in \mathbb{R}^n, b \in \mathbb{R} \implies \nu = 0 \tag{2.30}$$

and concludes the proof, see Equation (2.21). □

In this Section 2.1.5, we have now seen two universal approximation theorems for different activation functions. An extension to the presented theorems and a more general statement for single hidden layer neural networks is provided by [Les93]. The authors of the work state that the universal approximation property holds for any function from $C(\mathcal{K})$, $\mathcal{K} \subseteq \mathbb{R}^n$ being a compact set, with every continuous and non-polynomial activation function. However, all the outlined theorems do not put any constraint on the number of neurons in the hidden layer, which is referred to as *width* of NNs. In general, the wider a NN is designed, the larger the capacity of the NN. This comes at the cost of increasing the number of model parameters and therefore potentially facing the *curse of dimensionality* [Bel57; Bis06]. Hence, finding the best model might encounter some computational limitations due to the infinitely many possibilities of network architectures and corresponding parameter values.

Dealing with an arbitrary width and a fixed *depth* of NNs are considered as the classical type of universal approximation theorems. More recent works focus on the opposite type, that is NNs with an arbitrary depth and a fixed width. In general, adding more layers of neurons to a NN, *i.e.* making the network deeper, does not reduce its capacity (think of *e.g.* additional layers with the identity as activation function). On the contrary, compared to wider NNs it has been observed that deeper NNs are beneficial with respect to the parameters count and the convergence speed of the learning process [He16b]. In fact, in [Lu17] the authors have proven that multi layer neural networks that are of width $n + 4$ and equipped with the ReLU activation function are universal approximators for any Lebesgue integrable function on $\mathbb{R}^n$ (with respect to L_1 as distance measure). The latter result has recently been extended to width of $\max\{n + 1, K\}$ and L^p functions, *i.e.* functions that are p-integrable with respect to the Lebesgue measure [Par21]. Therefore, ever deeper network architectures have gained popularity in practice and that is why this family of models is associated with the term *deep learning*.

Nevertheless, none of these universal approximation theorems provide the design of a NN for a specific task, in particular none of them provide the appropriate model parameters. To this end, labeled data is typically presented to a NN, from which it can learn how to adjust its weights such that its output fits best the target. This process of feeding neural networks with data, therefore enabling them to learn, is known as "training".

2.2 Training of Neural Networks

In the preceding Section 2.1, the ability of neural networks to learn from data has already been stated. In the supervised learning setting for classification, a dataset $\mathcal{D}_N$ with manually annotated data examples is typically available. More formally, $\mathcal{D}_N = \{(\mathbf{x}^{(i)}, y^{(i)})\}_{i=1}^N$ consists of N randomly sampled data pairs with $\mathbf{x}^{(i)} \in \mathbb{R}^n, y^{(i)} \in \{1, \dots, K\}\ \forall\ i = 1, \dots, N$ denoting the i-th input and target, respectively.

Since the true data generating process for every data pair $(\mathbf{x}, y)$ of $\mathcal{D}_N$ is usually unknown, the goal in classification problems is to approximate the conditional probability $p(y|\mathbf{x})$, *i.e.* the probability for target class y given input $\mathbf{x}$, *e.g.* by using a neural network $f(\mathbf{x}|\theta)$. A neural network is a parametric model whose parameters θ are optimized to fit the distribution provided by $\mathcal{D}_N$. This optimization procedure is known as *training of neural networks*, therefore $\mathcal{D}_N$ is also referred to as training dataset. To this end, $\mathcal{D}_N$ is ideally as representative as possible for the true underlying distribution of $(\mathbf{x}, y)$.

2.2.1 Distance between Probability Distributions

One question, which naturally arises when training neural networks for classification, is what measure to use to quantify the "closeness" between the neural network output $f_y(\mathbf{x}|\theta)$ and the target probability $p(y|\mathbf{x})$ for classes $y \in \{1, \dots, K\}$. As we will see, this can be regarded as quantifying

the distance between two conditional probability distributions (or, more generally, the distance between two Markov kernels). For the sake of simplicity, let us first consider the case of unconditional probability distributions.

Definition 2.12. *Kullback-Leibler Divergence [Kul51; Kul59; Han16]*
Let μ, ν be two probability distributions over $\mathbb{R}^n$, and let ν be absolutely continuous with respect to μ. Then, the Kullback-Leibler divergence between μ and ν is defined as

$$\mathrm{KL}(\mu\|\nu) := -\int_{\mathbb{R}^n} \ln\left(\frac{d\nu}{d\mu}(\mathbf{z})\right) d\mu(\mathbf{z}) = \mathbb{E}_{Z\sim\mu}\left[-\ln\left(\frac{d\nu}{d\mu}(Z)\right)\right] . \tag{2.31}$$

Here, $\frac{d\nu}{d\mu}$ denotes the Radon–Nikodym derivative and $\mathbb{E}_{Z\sim\mu}$[1] the expectation operator with respect to the random variable $Z \sim \mu$.

The Kullback-Leibler divergence is also sometimes called *relative entropy* and defines a form of distance between two probability distributions. As $-\ln(\cdot)$ is a convex function, applying Jensen's inequality [Jen06] yields

$$\mathrm{KL}(\mu\|\nu) \geq -\ln\left(\int_{\mathbb{R}^n} \frac{d\nu}{d\mu}(\mathbf{z})\, d\mu(\mathbf{z})\right) = 0 \quad \text{with equality if and only if} \quad \mu = \nu . \tag{2.32}$$

Note that KL is not a metric since it is neither symmetric, *i.e.* $\mathrm{KL}(\mu\|\nu) \neq \mathrm{KL}(\nu\|\mu)$, nor does it satisfy a triangle inequality. Pinsker's inequality [Pin64], however, allows for comparison between the Kullback-Leibler divergence and other common distance metrics by means of their dual form [Han16]. More precisely, for the two probability measures μ, ν on $(\mathbb{R}^n, \mathcal{B}(\mathbb{R}^n))$, with $\mathcal{B}(\mathbb{R}^n)$ denoting the Borel algebra on $\mathbb{R}^n$, the commonly used total variation distance is given as

$$\|\mu-\nu\|_{\mathrm{TV}} := \sup_{A\in\mathcal{B}(\mathbb{R}^n)} |\mu(A)-\nu(A)| = \frac{1}{2}\sup_{g\in\mathrm{M}}\left|\int_{\mathbb{R}^n} g(\mathbf{z})\; d\mu(\mathbf{z}) - \int_{\mathbb{R}^n} g(\mathbf{z})\; d\nu(\mathbf{z})\right| , \tag{2.33}$$

where $\mathrm{M} := \{g : \mathbb{R}^n \to [-1,1] : g \text{ is measureable}\}$. Via the Wasserstein distance

$$\|\mu-\nu\|_{W_1(\mathbb{1}_{\mathbf{x}\neq\mathbf{x}'})} := \sup_{g\in\mathrm{Lip}}\left|\int_{\mathbb{R}^n} g(\mathbf{z})\; d\mu(\mathbf{z}) - \int_{\mathbb{R}^n} g(\mathbf{z})\; d\nu(\mathbf{z})\right| \tag{2.34}$$

where $\mathrm{Lip} := \{g : \mathbb{R}^n \to \mathbb{R} : |g(\mathbf{x}) - g(\mathbf{x}')| \leq \mathbb{1}_{\mathbf{x}\neq\mathbf{x}'}\} = \{g \in \mathrm{M} : \sup_{\mathbf{x}\in\mathbb{R}^n} g(\mathbf{x}) - \inf_{\mathbf{x}\in\mathbb{R}^n} g(\mathbf{x}) \leq 1\}$ denotes the family of 1-Lipschitz functions with respect to the trivial metric $\mathbb{1}_{\mathbf{x}\neq\mathbf{x}'}$, Pinsker's inequality bounds the total variation distance with the Kullback-Leibler divergence [Bob99], that is

$$\|\mu-\nu\|_{\mathrm{TV}} = \|\mu-\nu\|_{W_1(\mathbb{1}_{\mathbf{x}\neq\mathbf{x}'})} \leq \sqrt{\frac{1}{2}\,\mathrm{KL}(\mu\|\nu)} . \tag{2.35}$$

Consequently, this result implies that the topology generated by the Kullback-Leibler divergence is stronger than the one of the total variation distance, *i.e.*

$$\mathrm{KL}(\mu\|\nu_m) \xrightarrow{m\to\infty} 0 \implies \|\mu-\nu_m\|_{\mathrm{TV}} \xrightarrow{m\to\infty} 0 \tag{2.36}$$

for any sequence $\{\nu_1, \nu_2, \ldots\}$ of probability measures for which $\mathrm{KL}(\mu\|\nu_m) \xrightarrow{m\to\infty} 0$.

[1] Note that, for the sake of brevity, we sometimes omit the indication to the random variable in the notation of the expectation operator if it becomes clear from the context. We then *e.g.* write $\mathbb{E}_\mu$ instead of $\mathbb{E}_{Z\sim\mu}$.

Having now the Kullback-Leibler divergence as "strong" distance at hand, the two probability distributions, for which the distance is desired to be quantified, need to be identified next. In the setting of statistical learning, the learning target is the underlying distribution of a data generating process, provided that empirically observed samples produced by the data generating process are given. More formally, assume that μ denotes the true but unknown target distribution. Sampling from μ yields the training dataset $\mathcal{D}_N$ and accordingly the *empirical distribution* $\hat{\mu}$. Processing the training data via a parametric model $f(\cdot|\theta)$ yields the *model distribution* ν. Then, we want to minimize the Kullback-Leibler divergence between μ and ν.

To this end, let us consider the densities $d\mu(\mathbf{z}) = p(\mathbf{z})d\mathbf{z}$ and $d\nu(\mathbf{z}) = f(\mathbf{z}|\theta)d\mathbf{z}$ corresponding to the distributions μ and ν, respectively, both with respect to the Lebesgue measure $d\mathbf{z}$ on $\mathbb{R}^n$. Then, we minimize the Kullback-Leibler divergence between μ and ν by minimizing the expected value of the log density of the model distribution since

$$\mathrm{KL}(\mu\|\nu) = -\mathbb{E}_\mu\left[\ln\left(\frac{d\nu}{d\mu}(Z)\right)\right] = -\mathbb{E}_\mu\left[\ln\left(\frac{f(Z|\theta)}{p(Z)}\right)\right] = \underbrace{\mathbb{E}_\mu\left[-\ln f(Z|\theta)\right]}_{\text{expected risk}} + \underbrace{\mathbb{E}_\mu\left[\ln p(Z)\right]}_{\text{constant}}. \tag{2.37}$$

Here, $\mathbb{E}_\mu\left[-\ln f(Z|\theta)\right]$ is called the expected value of the log density of the model distribution ν, which we will also refer to as the *expected risk* in the following. The choice of parameters (as well as the choice of model class) for $f(Z|\theta)$ determines what distributions ν is capable to express. In particular, the only way to alter $\mathrm{KL}(\mu\|\nu)$ is by modifying θ. Since we do not have access to the true target distribution μ, we minimize the Kullback-Leibler divergence between the empirical distribution $\hat{\mu}$ and ν instead during training.

This approach incorporates several sources of errors. For instance, data distributed according to the target distribution μ can be collected in a dataset $\mathcal{D}_N$ but the empirical distribution $\hat{\mu}$ obtained by $\mathcal{D}_N$ is an estimation of μ and does not necessarily represent the full target distribution. Therefore, an empirical distribution introduces the so-called *estimation (or sampling) error*. Naturally, the estimation error is reduced the greater N, *i.e.* the larger the training dataset $\mathcal{D}_N$. Furthermore, the probability density function p of the target distribution is usually unknown. For that reason, p must be approximated using the training dataset $\mathcal{D}_N$ and some model $f(\cdot|\theta)$. The error associated with a misspecified model is referred to as *approximation (or model) error*. The approximation error obviously becomes smaller by better approximation, for which models of high expressivity are favorable. Given the universal approximation property, see Section 2.1.5, neural networks are an obvious choice in order to have a low approximation error. However, determining the optimal parameters of the model is another challenge, which is essentially an optimization problem. The error produced by the inability to optimize some training objective is then called *optimization error*. These three mentioned error terms are known as *error decomposition* of statistical learning methods [ShS14].

Regardless of the stated error decomposition, the minimization process of $\mathrm{KL}(\mu\|\nu)$ and therefore the success of learning algorithms, further depends on a proper estimate of the expected risk, *cf.* Equation (2.37). In order to tune the model parameters θ via optimization, an objective function is necessary, which explicitly computes an estimate of the expected risk by means of the available training data $\mathcal{D}_N$. Such a function is then called *empirical risk function*. Preferably, this function should have the property that for a large training dataset the empirical risk converges to the true expected risk. We will see that the *empirical negative log-likelihood* is suitable as such empirical risk function, which we will discuss in more detail after the formal introduction of conditional distributions in the following remark.

Remark 2.13. For the sake of simplicity, we have so far only considered probability measures on $(\mathbb{R}^n, \mathcal{B}(\mathbb{R}^n))$, with $\mathcal{B}(\mathbb{R}^n)$ denoting the Borel algebra on $\mathbb{R}^n$. This is valid for the case of unsupervised

learning but not for the case of supervised learning, thus also not for the learning task of classification. In classification problems we are interested in conditional densities, consequently we have to deal with conditional probability distributions. In this regard, the joint probability distribution $P_{X,Y}$ of a pair of random variables $(X, Y) : (\Omega, \mathcal{A}, P) \to (\mathbb{R}^n \times \mathbb{R}, \mathcal{B}(\mathbb{R}^n \times \mathbb{R}))$ is uniquely determined by

$$P_{X,Y}(B) = \int_{\mathbb{R}^n} \left(\int_{\mathbb{R}} \mathbb{1}_B(\mathbf{x}, y)\ P_{Y|X=\mathbf{x}}(y) \right) dP_X(\mathbf{x}) \quad \forall\ B \in \mathcal{B}(\mathbb{R}^n \times \mathbb{R})\ , \tag{2.38}$$

where P_X denotes the (marginal) distribution of X and $P_{Y|X}$ the (regular) conditional distribution of Y given X, *cf.* [Got20]. Given the densities $p_{X,Y}$ and p_X corresponding to the distributions $P_{X,Y}$ and P_X, respectively, $p(y|\mathbf{x}) = p_{X,Y}(\mathbf{x}, y)/p_X(\mathbf{x})$ (for $p_X(\mathbf{x}) > 0$) then defines the density of the conditional distribution $P_{Y|X}$. Seen from a more technical perspective, the latter is a so-called *Markov kernel*. In the following, we will be concerned about Markov kernels as learning targets (which we will however refer to as conditional probability distributions for the sake of clarity). Note that all statements in this section still hold analogously when probability measures are replaced by Markov kernels.

Definition 2.14. *Likelihood Function of Conditional Distributions [Fis25; Sha03]*
Let $\mu_{Y|X}$ be a conditional probability distribution over $\mathbb{R}$, and let $d\mu_{Y|X=\mathbf{x}}(y) = p(y|\mathbf{x})dy$ denote the corresponding conditional density with respect to the Lebesgue measure dy at the point $X = \mathbf{x}$. Moreover, let $\mathcal{D}_N = \{(\mathbf{x}^{(i)}, y^{(i)})\}_{i=1,}^{N}$ denote a set of examples independently drawn from the corresponding joint distribution. Then, we define the likelihood function as

$$\mathscr{L}(\mathcal{D}_N|\mu_{Y|X}) := \prod_{i=1}^{N} p(y^{(i)}|\mathbf{x}^{(i)})\ . \tag{2.39}$$

We note that the likelihood can be evaluated explicitly by means of the model density function and the training dataset. With $d\nu_{Y|X=\mathbf{x}}(y) = f_y(\mathbf{x}|\theta)dy$ denoting the probability density function that is parameterized by θ and corresponds to the conditional model distribution $\nu_{Y|X}$, the *empirical risk minimizer* [ShS14] for the negative log-likelihood as empirical risk function is given as

$$\hat{\theta} := \arg\min_{\theta} -\ln \mathscr{L}(\mathcal{D}_N|\nu_{Y|X}, \theta) = \arg\min_{\theta} -\ln \prod_{i=1}^{N} f_{y^{(i)}}(\mathbf{x}^{(i)}|\theta) = \arg\min_{\theta} -\sum_{i=1}^{N} \ln f_{y^{(i)}}(\mathbf{x}^{(i)}|\theta) \tag{2.40}$$

which is defined as the optimal estimate for the model parameters θ with respect to the empirical distribution. In this particular context, empirical risk minimization is equivalent to the maximum likelihood approach [Pfa11] since $\arg\min -\ln \mathscr{L}(\mathcal{D}_N|\nu_{Y|X}, \theta) = \arg\max \mathscr{L}(\mathcal{D}_N|\nu_{Y|X}, \theta)$. Moreover, by the weak law of large numbers [Ros06] the empirical risk computed via the negative log-likelihood converges in probability to the expected risk. More formally,

$$-\frac{1}{N}\sum_{i=1}^{N} \ln f_{y^{(i)}}(\mathbf{x}^{(i)}|\theta) = \mathbb{E}_{(X,Y)\sim\hat{\mu}}\left[-\ln f_Y(X|\theta)\right] \xrightarrow{P} \mathbb{E}_{(X,Y)\sim\mu}\left[-\ln f_Y(X|\theta)\right] \quad \text{when } N \to \infty\ , \tag{2.41}$$

where $\mu = \mu_{Y|X}P_X$ now denotes the joint probability distribution of a random data pair (X, Y) and comprises the conditional distribution $\mu_{Y|X}$ of Y given X and the marginal distribution P_X of X (analogously for the empirical distribution $\hat{\mu}$). The result of (2.41) holds for any fixed set of parameters θ and depends only on the size of the dataset $\mathcal{D}_N$. For a sufficiently large training dataset and according to the uniform law of large numbers [New86; Vaa96] (which allows us to change the order of $\arg\min$ and $\lim$) we have

$$\lim_{N\to\infty} \hat{\theta} = \arg\min_{\theta} \lim_{N\to\infty} -\frac{1}{N}\sum_{i=1}^{N} \ln f_{y^{(i)}}(\mathbf{x}^{(i)}|\theta) = \arg\min_{\theta} \mathbb{E}_{(X,Y)\sim\mu}\left[-\ln f_Y(X|\theta)\right] =: \theta^*, \tag{2.42}$$

with θ^* denoting the expected risk minimizer. Furthermore, the negative log-likelihood function represents an unbiased empirical risk function for the Kullback-Leibler divergence, which is a further favorable property of an empirical risk function.

Definition 2.15. *Unbiased Empirical Risk Function*
Let $\mathcal{D}_N \in \mathbb{R}^{n\times N}$ denote a dataset consisting of $N \in \mathbb{N}$ independent data samples from μ. An empirical risk function $R = \{R_N\}_{N\in\mathbb{N}}$ for a given distance d between two probability measures $\nu \in \mathcal{H}_N, \mu \in \mathcal{T}$ is a family of functions $R_N : \mathcal{H}_N \times \mathbb{R}^{n\times N} \to \mathbb{R}$ for which there exists $c \in \mathbb{R}_{>0}$ and $g : \mathcal{T} \to \mathbb{R}$ such that

$$c_N R_N(\nu, \mathcal{D}_N) + g(\mu) \xrightarrow{P} \mathrm{d}(\nu, \mu) \quad \text{when} \quad N \to \infty \quad \forall\ \nu \in \mathcal{H}_N, \mu \in \mathcal{T}\ . \tag{2.43}$$

An empirical risk function is unbiased if

$$\mathbb{E}_\mu[c_N R_N(\nu, \mathcal{D}_N) + g(\mu)] = \mathrm{d}(\nu, \mu) \quad \forall\ \nu \in \mathcal{H}_N, \mu \in \mathcal{T}\ . \tag{2.44}$$

Lemma 2.16. *Unbiased Empirical Risk Function for the Kullback-Leibler Divergence [Got20]*
Let $\mu_{Y|X}$, $\nu_{Y|X}$ be two conditional probability distributions over $\mathbb{R}$, and let $d\mu_{Y|X=\mathbf{x}}(y) = p(y|\mathbf{x})dy$ and $d\nu_{Y|X=\mathbf{x}}(y) = q(y|\mathbf{x})dy$, respectively, denote the corresponding conditional densities with respect to the Lebesgue measure dy at the point $X = \mathbf{x}$. Further, let $\mathcal{D}_N = \{(\mathbf{x}^{(i)}, y^{(i)})\}_{i=1}^N$ be a dataset consisting of $\mathbb{R}^{n+1}$-valued random examples independently drawn from the joint distribution $\mu = \mu_{Y|X}P_X$. Then, the negative log-likelihood function

$$-\ln \mathscr{L}(\mathcal{D}_N|\nu_{Y|X}) = -\ln \prod_{i=1}^N q(y^{(i)}|\mathbf{x}^{(i)}) = -\sum_{i=1}^N \ln q(y^{(i)}|\mathbf{x}^{(i)}) \tag{2.45}$$

with $c = \frac{1}{N}$ and $g(\mu) = \mathbb{E}_\mu[\ln p(Y|X)]$ is an unbiased empirical risk function for the Kullback-Leibler divergence between $\mu_{Y|X}$ and $\nu_{Y|X}$, with the Kullback-Leibler divergence between these two conditional probability distributions given as

$$\mathrm{KL}_{P_X}(\mu_{Y|X}\|\nu_{Y|X}) := \int_{\mathbb{R}^n} \mathrm{KL}(\mu_{Y|X=\mathbf{x}}\|\nu_{Y|X=\mathbf{x}})\, dP_X(\mathbf{x}) = \mathbb{E}_{X\sim P_X}\left[\mathrm{KL}(\mu_{Y|X}\|\nu_{Y|X})\right]\ . \tag{2.46}$$

Proof. We start the proof by showing that the negative log-likelihood is an empirical risk function for the Kullback-Leibler divergence, *i.e.* we first show:

$$-\frac{1}{N}\ln \mathscr{L}(\mathcal{D}_N|\nu_{Y|X}) + \mathbb{E}_{(X,Y)\sim\mu}[\ln p(Y|X)] \xrightarrow{N\to\infty} \mathrm{KL}_{P_X}(\mu_{Y|X}\|\nu_{Y|X})\ . \tag{2.47}$$

Using the law of large numbers [Ros06], *cf.* (2.41), we have

$$\begin{aligned} &-\frac{1}{N}\ln \mathscr{L}(\mathcal{D}_N|\nu_{Y|X}) + \mathbb{E}_{(X,Y)\sim\mu}[\ln p(Y|X)] \\ \xrightarrow{N\to\infty}\ &\mathbb{E}_{(X,Y)\sim\mu}\left[-\ln q(Y|X)\right] + \mathbb{E}_{(X,Y)\sim\mu}[\ln p(Y|X)] \end{aligned} \tag{2.48}$$

and it follows with Remark 2.13

$$\mathbb{E}_{(X,Y)\sim\mu}\left[-\ln q(Y|X)\right] + \mathbb{E}_{(X,Y)\sim\mu}[\ln p(Y|X)] \tag{2.49}$$

$$= -\int_{\mathbb{R}^n\times\mathbb{R}} \ln q(y|\mathbf{x})\, d\mu(\mathbf{x}, y) + \int_{\mathbb{R}^n\times\mathbb{R}} \ln p(y|\mathbf{x})\, d\mu(\mathbf{x}, y) \tag{2.50}$$

$$= -\int_{\mathbb{R}^n\times\mathbb{R}} \ln\left(\frac{q(y|\mathbf{x})}{p(y|\mathbf{x})}\right) d\mu_{Y|X}P_X(\mathbf{x}, y) = \int_{\mathbb{R}^n}\left(-\int_{\mathbb{R}} \ln\left(\frac{d\nu_{Y|X=\mathbf{x}}}{d\mu_{Y|X=\mathbf{x}}}(y)\right) d\mu_{Y|X=\mathbf{x}}(y)\right) dP_X(\mathbf{x}) \tag{2.51}$$

$$= \int_{\mathbb{R}^n} \mathrm{KL}(\mu_{Y|X=\mathbf{x}}\|\nu_{Y|X=\mathbf{x}})\, dP_X(\mathbf{x}) = \mathbb{E}_{X\sim P_X}\left[\mathrm{KL}(\mu_{Y|X}\|\nu_{Y|X})\right] = \mathrm{KL}_{P_X}(\mu_{Y|X}\|\nu_{Y|X})\ . \tag{2.52}$$

Next, we show that negative log-likelihood as empirical risk function is unbiased with respect to the Kullback-Leibler divergence, *i.e.* we show:

$$\mathbb{E}_{(X,Y)\sim\mu}\left[-\frac{1}{N}\ln\mathscr{L}(\mathcal{D}_N|\nu_{Y|X}) + \mathbb{E}_{(X,Y)\sim\mu}[\ln p(Y|X)]\right] = \mathrm{KL}_{P_X}(\mu_{Y|X}\|\nu_{Y|X}) \ . \tag{2.53}$$

Analogously to the previous steps, we have

$$\mathbb{E}_{(X,Y)\sim\mu}\left[-\frac{1}{N}\sum_{i=1}^{N}\ln q(Y|X) + \mathbb{E}_{(X,Y)\sim\mu}[\ln p(Y|X)]\right] \tag{2.54}$$

$$= \mathbb{E}_{(X,Y)\sim\mu}\Big[-\ln q(Y|X)\Big] + \mathbb{E}_{(X,Y)\sim\mu}\Big[\ln p(Y|X)\Big] = \mathbb{E}_{(X,Y)\sim\mu}\Big[-\ln q(Y|X) + \ln p(Y|X)\Big] \tag{2.55}$$

$$= \mathbb{E}_{(X,Y)\sim\mu}\left[-\ln\left(\frac{q(Y|X)}{p(Y|X)}\right)\right] = \mathbb{E}_{(X,Y)\sim\mu}\left[-\ln\left(\frac{d\nu_{Y|X}}{d\mu_{Y|X}}(Y)\right)\right] \tag{2.56}$$

$$= \mathbb{E}_{X\sim P_X}\left[\mathbb{E}_{Y\sim\mu_{Y|X}}\left[-\ln\left(\frac{d\nu_{Y|X}}{d\mu_{Y|X}}(Y)\right)\right]\right] = \mathbb{E}_{X\sim P_X}\Big[\mathrm{KL}(\mu_{Y|X}\|\nu_{Y|X})\Big] \tag{2.57}$$

$$= \mathrm{KL}_{P_X}(\mu_{Y|X}\|\nu_{Y|X}) \ . \tag{2.58}$$

□

In this subsection, we have introduced the Kullback-Leibler divergence to quantify the closeness between two probability distributions. We have seen that the Kullback-Leibler divergence represents a strong distance since its minimization implies the minimization of other commonly-used distance metrics, such as the total variation. In the setting of classification problems, we aim at minimizing the Kullback-Leibler divergence between a given conditional class distribution as target and a conditional class distribution according to some statistical model as estimation. As the true target distribution is usually unknown, data examples can be sampled and collected in a dataset yielding the empirical distribution as an approximation. This allows for an explicit evaluation of the empirical risk, which ideally should have properties of a proper estimate for the expected risk with respect to the unknown target distribution. In this light, we have seen that the negative log-likelihood is an unbiased empirical risk function for the Kullback-Leibler divergence. In other words, for sufficiently large datasets minimizing the empirical risk computed via the negative log-likelihood also minimizes the true Kullback-Leibler divergence between the unknown target distribution and the estimated model distribution. As probability distributions can be fully determined by their respective densities, neural networks as statistical models are employed to learn the density of the empirical (and thus the target) distribution. In what follows, we focus on how the negative log-likelihood is used in practice in combination with neural networks for classification.

2.2.2 Loss Function in Classification Problems

After gaining an intuition and an understanding of the learning objective of neural networks as statistical models, we discuss the explicit implementation for classification models in this section. As already indicated, tuning model parameters is essentially an optimization problem. The objective to optimize is the empirical risk function, which is more commonly known under the term *loss function* [Goo16; Cal20] in deep learning terminology. Just like a risk function, the loss function evaluates the proximity between a target distribution and an approximated distribution. In classification problems, these distributions are given over a finite set of classes, *i.e.* the model and target distributions are both discrete. In this regard, the discrete cross-entropy is the most popular loss function for training classification neural networks.

Definition 2.17. *Discrete Cross-Entropy Loss Function [Bis06; Has09; Cal20]*
Let μ and ν be two discrete probability distribution over $\mathbb{Z}$ with probability mass functions p and q, respectively. Then, the cross-entropy of μ relative to ν is defined as

$$H(\mu, \nu) := -\sum_{y \in \mathbb{Z}} p(y) \ln q(y) = \mathbb{E}_{Y \sim \mu}[-\ln q(Y)] \ . \tag{2.59}$$

The motivation for using the cross-entropy is due to its connection to the Kullback-Leibler divergence, *cf*. Equation (2.37), and therefore also its relation to maximum likelihood estimation, *cf*. Equation (2.40). In order to apply the cross-entropy as loss function for training classification NNs, the model output and the target must be in a format such that they can be understood as input conditional probability distributions over a finite set of classes. By employing the softmax activation in the last layer of a NN, see Definition 2.8, the model output becomes a categorical probability distribution $f(\mathbf{x}|\theta) \in (0,1)^K, \|f(\mathbf{x}|\theta)\|_1 = 1$ over classes $\{1, \dots, K\}$ for all $\mathbf{x} \in \mathbb{R}^n$. In practice, given some labeled data in the training dataset $\mathcal{D}_N$, the cross-entropy as loss function is then employed via

$$\mathcal{L}(y, f(\mathbf{x}|\theta)) = -\sum_{k=1}^{K} \mathbb{1}_{k=y} \ln f_k(\mathbf{x}|\theta) = -\ln f_y(\mathbf{x}|\theta) \quad \forall \ (\mathbf{x}, y) \in \mathcal{D}_N. \tag{2.60}$$

Here, $p(y|\mathbf{x}) = 1 \ \forall \ (\mathbf{x}, y) \in \mathcal{D}_N$ (so-called *one-hot encoding*) is considered as input conditional probability of the target distribution, implying $p(y'|\mathbf{x}) = 0 \ \forall \ y' \neq y, (\mathbf{x}, y) \in \mathcal{D}_N$. Then, averaging over all training examples yields the *average empirical loss*

$$\bar{\mathcal{L}}(\mathcal{D}_N, f(\mathbf{x}|\theta)) := \frac{1}{|\mathcal{D}_N|} \sum_{(\mathbf{x},y) \in \mathcal{D}_N} \mathcal{L}(y, f(\mathbf{x}|\theta)) = \frac{1}{N} \sum_{i=1}^{N} -\ln f_{y^{(i)}}(\mathbf{x}^{(i)}|\theta) \ . \tag{2.61}$$

Comparing Equation (2.61) and (2.41), we directly realize that the cross-entropy loss (with one-hot encoded targets) is essentially the same as employing the negative log-likelihood as empirical risk function, thus also sharing the same properties of a proper estimate for the expected loss / risk.

In the field of statistical learning, many other loss functions exist and depending on the type of task, other loss functions may be preferred over the cross-entropy. For instance, for simple regression problems the mean squared error [Bis06] is commonly used, whereas in cases of unbalanced label distributions the Dice loss [Sør48; Mil16] is the first choice.

2.2.3 Gradient Descent

The process of learning incorporates tuning parameters of a NN such that the empirical average loss is minimized. The most commonly used optimization algorithm in deep learning is *gradient descent*. This algorithm is an iterative greedy approach, searching a minimum by directing a step at each iteration into the direction that decreases the loss the most (given some loss function $\mathcal{L}$). The direction to reach the minimum is given by the negative of the gradient of the average empirical loss $\bar{\mathcal{L}}$ with respect to the model parameters. Then, the standard parameters update from iteration r to $r+1$ for the parameters θ is defined as

$$\theta^{[r+1]} := \theta^{[r]} - \eta \cdot \nabla_\theta \bar{\mathcal{L}} \restriction_{\theta=\theta^{[r]}} \ , \ \eta \in \mathbb{R}, \ r \in \mathbb{N} \tag{2.62}$$

where the starting solution $\theta^{[0]}$ is randomly initialized, and $\eta > 0$ denotes the learning rate, *i.e.* the step size taken along the negative of the gradient $\nabla_\theta \bar{\mathcal{L}}$. This standard procedure, however, requires to compute the gradients for the whole dataset, which becomes more computationally expensive and impractical the larger the dataset.

In contrast, *mini-batch gradient descent* performs the parameters update for a random subset / batch $\mathcal{I} \subset \{1, \ldots, N\}$ of the training dataset $\mathcal{D}_N$, *i.e.*

$$\theta^{[r+1]} := \theta^{[r]} - \eta \cdot \nabla_\theta \widetilde{\mathcal{L}} \restriction_{\theta=\theta^{[r]}} \quad \text{with} \quad \widetilde{\mathcal{L}} = \frac{1}{|\mathcal{I}|} \sum_{i \in \mathcal{I}} \mathcal{L}(y^{(i)}, f(\mathbf{x}^{(i)}|\theta)),\ |\mathcal{I}| \ll N = |\mathcal{D}_N| \ . \tag{2.63}$$

The latter update procedure computes an average of gradients on randomly selected data examples. In the most extreme form, the mini-batch consists of one single data example, *i.e.* $|\mathcal{I}| = 1$. Choosing a subset as approximation to perform gradient descent is also known as *stochastic gradient descent* (SGD). This approach does not necessarily require the full gradient of the entire training dataset but a random descent direction. The only requirement in order to converge on convex loss surfaces is that the expected value of the random direction is the negative of the true gradient of the expected loss [ShS14; Dei20]. More precisely, for random examples $(X^{(i)}, Y^{(i)}) \sim \mu,\ i \in \mathcal{I}$ independently sampled from the target distribution

$$\mathbb{E}_\mu \left[\nabla_\theta \widetilde{\mathcal{L}} \right] = \nabla_\theta \mathbb{E}_\mu \left[\widetilde{\mathcal{L}} \right] = \nabla_\theta \mathbb{E}_\mu \left[\frac{1}{|\mathcal{I}|} \sum_{i \in \mathcal{I}} \mathcal{L}(Y^{(i)}, f(X^{(i)}|\theta)) \right] = \nabla_\theta \underbrace{\mathbb{E}_\mu \left[\mathcal{L}(Y^{(1)}, f(X^{(1)}|\theta)) \right]}_{\text{expected loss}} , \tag{2.64}$$

which is a direct consequence of the linearity of gradients and expectation. Hence, we can understand $\nabla_\theta \mathcal{L}(Y, f(X|\theta))$ as an unbiased estimate for the gradient of the expected loss. In this way, SGD circumvents the major problem of memory restrictions of standard gradient descent. To this end, the random examples need to be sampled again in each iteration of the SGD algorithm. As in practice the training dataset is typically assumed to consist of independent and identically distributed examples of the target distribution, SGD is often applied on the training data only. In this context, an *epoch* refers to one cycle through the whole training dataset. A neural network is usually trained for multiple epochs, allowing the model to learn different patterns in each cycle in order to better generalize to unseen data.

One key challenge in using gradient descent is the choice of the learning rate η. The learning rate controls the convergence of gradient descent. This might include the rate of convergence, but also whether the optimizer gets trapped in local minima. In practice, this is a hyperparameter which is mainly tuned by trial and error. To this end, *learning rate schedules* adjust η depending on the progress of training [Dar92]. Another extension to gradient descent is *momentum* [Qia99] which is another hyperparameter. By using momentum, gradients of past updates are accumulated to determine the parameters update at a current step, aiming at accelerating gradient descent. The latter approach has motivated *adaptive moment estimation*, also known as Adam [Kin15]. The Adam algorithm computes an individual learning rate for each parameter and it represents the main optimization algorithm in deep learning.

In general, however, the convergence of optimization is mostly neglected in this field. This is for instance apparent by the fact that gradient-based algorithms are motivated by ideas for convex optimization, whereas most loss surfaces in deep learning are known to be non-convex. These algorithms may provably converge to local minima, but finding the global minimum, *i.e.* the empirical risk minimizer (see again Equation (2.40)), is generally known to be NP-hard [ShS14]. That is why the success of deep learning tends to be based on the loss surfaces having many local minima that are sufficiently good rather than sophisticated optimization.

2.2.4 Backpropagation

Gradient descent and its variants are the most-used optimization algorithms to tune the weights in neural networks. Their popularity is not only based on their simplicity, but also their straightforward

applicability to NNs, since the network design allows for an efficient computation of gradients. Recall that in order to apply gradient descent, the gradient of the loss function $\mathcal{L}$ with respect to all parameters in the model $f(\cdot|\theta)$ are needed. To this end, *backpropagation* is an algorithm for computing such gradients in neural networks and it is key to the success of deep learning.

As introduced in Definition 2.5, a multi layer neural network is composed of a chain of functions

$$f(\mathbf{x}) = \left(f^{(L)} \circ f^{(L-1)} \circ ... \circ f^{(1)}\right)(\mathbf{x}) \ , \tag{2.65}$$

see also Equation (2.4). Each layer function possesses its own set of parameters $\theta^{(\ell)} = \{\mathbf{W}^{(\ell)}, \mathbf{b}^{(\ell)}\}$ for weights as well biases and its own activation functions $\boldsymbol{\phi}^{(\ell)}$, *i.e.* for all $\ell \in 1, \dots, L$

$$f^{(\ell)}(\mathbf{x}|\theta^{(\ell)}) = \boldsymbol{\phi}^{(\ell)}\left(\mathbf{W}^{(\ell)}\mathbf{x} + \mathbf{b}^{(\ell)}\right), \ \mathbf{W}^{(\ell)} = \left(\mathbf{w}^{(\ell)}_{\bullet 1}, \dots, \mathbf{w}^{(\ell)}_{\bullet n_{\ell-1}}\right) \in \mathbb{R}^{n_\ell \times n_{\ell-1}}, \mathbf{b}^{(\ell)} \in \mathbb{R}^{n_\ell} \ . \tag{2.66}$$

Furthermore, in feedforward NNs the output of a layer represents the input of the next one, *i.e.*

$$\mathbf{x}^{(\ell)} = f^{(\ell)}(\mathbf{x}^{(\ell-1)}) = \boldsymbol{\phi}^{(\ell)} \underbrace{\left(\mathbf{W}^{(\ell)}\mathbf{x}^{(\ell-1)} + \mathbf{b}^{(\ell)}\right)}_{=:\mathbf{z}^{(\ell)}} \ \forall \ \ell = 1, \dots, L \ \text{ and } \ \mathbf{x}^{(0)} = \mathbf{x} \ , \tag{2.67}$$

cf. Equation (2.3). For the sake of simplified notation, consider the set $\Theta^{(\ell)} = \{\mathbf{w}^{(\ell)}_{\bullet 1}, \dots, \mathbf{w}^{(\ell)}_{\bullet n_{\ell-1}}, \mathbf{b}^{(\ell)}\}$ consisting of the column vectors of the weights matrix $\mathbf{W}^{(\ell)}$ and also the biases vector $\mathbf{b}^{(\ell)}$ at layer ℓ. Then, using calculus and the chain rule, the gradient of some loss function $\mathcal{L} : \mathbb{R}^K \to \mathbb{R}$ with respect to all $\boldsymbol{\theta}^{(\ell)} \in \Theta^{(\ell)}$ is obtained via

$$\nabla^\top_{\boldsymbol{\theta}^{(\ell)}} \mathcal{L} \circ f = \nabla^\top_{f^{(L)}} \mathcal{L} \circ f \cdot \mathbf{J}_{f^{(L-1)}} f^{(L)} \cdots \ \mathbf{J}_{f^{(\ell)}} f^{(\ell+1)} \cdot \mathbf{J}_{\boldsymbol{\theta}^{(\ell)}} f^{(\ell)} \tag{2.68}$$

$$= \underbrace{\nabla^\top_{f^{(L)}} \mathcal{L} \circ f \cdot \mathbf{J}_{\mathbf{z}^{(L)}} \boldsymbol{\phi}^{(L)} \mathbf{J}_{\mathbf{x}^{(L-1)}} \mathbf{z}^{(L)} \cdots \ \mathbf{J}_{\mathbf{z}^{(\ell+1)}} \boldsymbol{\phi}^{(\ell+1)}}_{= \nabla^\top_{\mathbf{z}^{(\ell+1)}} \mathcal{L} \circ f, \text{ also required to compute } \nabla^\top_{\boldsymbol{\theta}^{(\ell+1)}} \mathcal{L} \circ f} \mathbf{J}_{\mathbf{x}^{(\ell)}} \mathbf{z}^{(\ell+1)} \cdot \mathbf{J}_{\mathbf{z}^{(\ell)}} \boldsymbol{\phi}^{(\ell)} \mathbf{J}_{\boldsymbol{\theta}^{(\ell)}} \mathbf{z}^{(\ell)} \tag{2.69}$$

for each neural network layer $\ell = 1, \dots, L$. Here, ∇g denotes the gradient and $\mathbf{J}g$ the Jacobian of a function g. Clearly, the feedforward design of neural networks allows for reusing most of the computations at layer ℓ if the gradients of the deeper layer $\ell + 1$ have already been computed. More specifically, in order to determine $\nabla_{\boldsymbol{\theta}^{(\ell)}} \mathcal{L} \circ f$, the gradient $\nabla_{\mathbf{z}^{(\ell+1)}} \mathcal{L} \circ f$ represents a suitable auxiliary factor, which in turn itself shares common computations with $\nabla_{\mathbf{z}^{(\ell+2)}} \mathcal{L} \circ f$, and so forth, *cf.* Equation (2.69). Hence, starting from the last layer, backpropagation essentially consists of recursively computing the gradient of some loss with respect to the parameters of one layer at a time. In each iteration, the algorithm reuses intermediate terms in the chain rule that have already been computed and stored in previous iterations. As the number of parameters of NNs in deep learning could run into millions, backpropagation is considerably more efficient than direct computation of each single gradient, therefore this procedure assures more reasonable computational cost when training NNs. In the following, we examine the implementation of the backpropagation algorithm in more detail.

Definition 2.18. *Network Output Error per Layer [Bis06; ShS14; Nie17]*
Let $f : \mathbb{R}^n \to \mathbb{R}^K$, $n, K \in \mathbb{N}$ denote a multi layer neural network with $L \in \mathbb{N}$ layers and let $\mathcal{L} : \mathbb{R}^K \to [0, \infty)$ denote a loss function. The network output error function $\boldsymbol{\delta}^{(\ell)} : \mathbb{R}^n \to \mathbb{R}^{n_\ell}$ associated with the ℓ-th layer of the network f is defined as

$$\boldsymbol{\delta}^{(\ell)} := \nabla_{\mathbf{z}^{(\ell)}} \mathcal{L} \circ f \quad \forall \ \ell = 1, \dots, L \ , \tag{2.70}$$

which is the gradient of $\mathcal{L}$ with respect to the weighted input $\mathbf{z}^{(\ell)} = \mathbf{W}^{(l)}\mathbf{x}^{(l-1)} + \mathbf{b}^{(\ell)} \in \mathbb{R}^{n_\ell}$ in the ℓ-th network layer.

Proposition 2.19. *Error Backpropagation [Dre62; Rum86; Bis06; ShS14; Nie17; Dei20]*
Let $f : \mathbb{R}^n \to \mathbb{R}^K$, $n, K \in \mathbb{N}$ denote a multi layer neural network with $L \in \mathbb{N}$ layers and component-wise activation functions $\boldsymbol{\phi}^{(\ell)} = (\phi_1^{(\ell)}, \dots, \phi_{n_\ell}^{(\ell)})$, $\ell = 1, \dots, L$. Further, let $\mathcal{L} : \mathbb{R}^K \to [0, \infty)$ denote a loss function. Then, the network output error function for the L-th layer of f is obtained by

$$\boldsymbol{\delta}^{(L)} = \operatorname{diag}\left(\boldsymbol{\phi}'^{(L)}(\mathbf{z}^{(L)})\right) \nabla_f \mathcal{L} \circ f \ , \tag{2.71}$$

and for the remaining layers the error is propagated backwards through the network yielding

$$\boldsymbol{\delta}^{(\ell)} = \operatorname{diag}\left(\boldsymbol{\phi}'^{(\ell)}(\mathbf{z}^{(\ell)})\right) (\mathbf{W}^{(\ell+1)})^\top \boldsymbol{\delta}^{(\ell+1)} \quad \forall\ \ell = (L-1), \dots, 1 \ , \tag{2.72}$$

where

$$\operatorname{diag}\left(\boldsymbol{\phi}'^{(\ell)}(\mathbf{z}^{(\ell)})\right) := \begin{pmatrix} \frac{\partial \phi_1^{(\ell)}}{\partial z_1^{(\ell)}}(z_1^{(\ell)}) & 0 & \dots & 0 \\ 0 & \frac{\partial \phi_2^{(\ell)}}{\partial z_2^{(\ell)}}(z_2^{(\ell)}) & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \frac{\partial \phi_{n_\ell}^{(\ell)}}{\partial z_{n_\ell}^{(\ell)}}(z_{n_\ell}^{(\ell)}) \end{pmatrix} \quad \forall\ \ell = 1, \dots, L \ . \tag{2.73}$$

Proof. Starting with Equation (2.71), we note that the network output, *i.e.* the output of layer L, can also be written as $f(\mathbf{x}) = f^{(L)}(\mathbf{x}^{(L-1)}) = \boldsymbol{\phi}^{(L)}(\mathbf{z}^{(L)})$. Moreover, since $\boldsymbol{\phi}^{(L)}(\mathbf{z}^{(L)}) = (\phi_k^{(L)}(\mathbf{z}_k^{(L)}))_{k=1}^{n_L}$ consists of component-wise activation functions,

$$\mathbf{J}_{\mathbf{z}^{(L)}} \boldsymbol{\phi}^{(L)} = \begin{pmatrix} \frac{\partial \phi_1^{(L)}}{\partial z_1^{(L)}}(z_1^{(L)}) & \frac{\partial \phi_1^{(L)}}{\partial z_2^{(L)}}(z_1^{(L)}) & \dots & \frac{\partial \phi_1^{(L)}}{\partial z_{n_L}^{(L)}}(z_1^{(L)}) \\ \frac{\partial \phi_2^{(L)}}{\partial z_1^{(L)}}(z_2^{(L)}) & \frac{\partial \phi_2^{(L)}}{\partial z_2^{(L)}}(z_2^{(L)}) & \dots & \frac{\partial \phi_2^{(L)}}{\partial z_{n_L}^{(L)}}(z_2^{(L)}) \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial \phi_{n_L}^{(L)}}{\partial z_1^{(L)}}(z_{n_L}^{(L)}) & \frac{\partial \phi_{n_L}^{(L)}}{\partial z_2^{(L)}}(z_{n_L}^{(L)}) & \dots & \frac{\partial \phi_{n_L}^{(L)}}{\partial z_{n_L}^{(L)}}(z_{n_L}^{(L)}) \end{pmatrix} \tag{2.74}$$

$$= \begin{pmatrix} \frac{\partial \phi_1^{(L)}}{\partial z_1^{(L)}}(z_1^{(L)}) & 0 & \dots & 0 \\ 0 & \frac{\partial \phi_2^{(L)}}{\partial z_2^{(L)}}(z_2^{(L)}) & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \frac{\partial \phi_{n_L}^{(L)}}{\partial z_{n_L}^{(L)}}(z_{n_L}^{(L)}) \end{pmatrix} = \operatorname{diag}\left(\boldsymbol{\phi}'^{(L)}(\mathbf{z}^{(L)})\right) \ . \tag{2.75}$$

Then, by applying the chain rule from calculus

$$\boldsymbol{\delta}^{(L)} = \nabla_{\mathbf{z}^{(L)}} \mathcal{L} \circ f = \left(\nabla_{f^{(L)}}^\top \mathcal{L} \circ f \cdot \mathbf{J}_{\mathbf{z}^{(L)}} \boldsymbol{\phi}^{(L)}\right)^\top \tag{2.76}$$

$$= \mathbf{J}_{\mathbf{z}^{(L)}}^\top \boldsymbol{\phi}^{(L)} \cdot \nabla_{f^{(L)}} \mathcal{L} \circ f = \operatorname{diag}\left(\boldsymbol{\phi}'^{(L)}(\mathbf{z}^{(L)})\right) \nabla_f \mathcal{L} \circ f \ . \tag{2.77}$$

For the remaining layer, it follows immediately with

$$\mathbf{J}_{\mathbf{x}^{(\ell)}} \mathbf{z}^{(\ell+1)} = \mathbf{J}_{\mathbf{x}^{(\ell)}} (\mathbf{W}^{(\ell+1)} \mathbf{x}^{(\ell)} + \mathbf{b}^{(\ell+1)}) = \mathbf{W}^{(\ell+1)} \tag{2.78}$$

and analogous steps as in the first part of this proof

$$\boldsymbol{\delta}^{(\ell)} = \nabla_{\mathbf{z}^{(\ell)}} \mathcal{L} \circ f = \left(\nabla^\top_{\mathbf{z}^{(\ell+1)}} \mathcal{L} \circ f \cdot \mathbf{J}_{\mathbf{x}^{(\ell)}} \mathbf{z}^{(\ell+1)} \cdot \mathbf{J}_{\mathbf{z}^{(\ell)}} \boldsymbol{\phi}^{(\ell)} \right)^\top \tag{2.79}$$

$$= \mathbf{J}^\top_{\mathbf{z}^{(\ell)}} \boldsymbol{\phi}^{(\ell)} \cdot \mathbf{J}^\top_{\mathbf{x}^{(\ell)}} \mathbf{z}^{(\ell+1)} \cdot \nabla_{\mathbf{z}^{(\ell+1)}} \mathcal{L} \circ f \tag{2.80}$$

$$= \operatorname{diag}\left(\boldsymbol{\phi}'^{(\ell)}(\mathbf{z}^{(\ell)}) \right) (\mathbf{W}^{(\ell+1)})^\top \boldsymbol{\delta}^{(\ell+1)} \quad \forall\ \ell = (L-1), \dots, 1\ . \tag{2.81}$$

□

Corollary 2.20. *Gradients via Backpropagation [Dre62; Rum86; Bis06; ShS14; Nie17; Dei20]*
Let $f : \mathbb{R}^n \to (0,1)^K,\ n, K \in \mathbb{N}$ denote a multi layer neural network with $L \in \mathbb{N}$ layers and let $\mathcal{L} : \mathbb{R}^K \to [0, \infty)$ denote a loss function. The gradient of $\mathcal{L}$ with respect to the network weights at layer ℓ is given by

$$\nabla_{\mathbf{w}^{(\ell)}_{\bullet j}} \mathcal{L} \circ f = x_j^{(\ell-1)} \boldsymbol{\delta}^{(\ell)} \quad \forall\ j = 1, \dots, n_{\ell-1},\ \ell = 1, \dots, L\ , \tag{2.82}$$

and with respect to the biases at layer ℓ by

$$\nabla_{\mathbf{b}^{(\ell)}} \mathcal{L} \circ f = \boldsymbol{\delta}^{(\ell)} \quad \forall\ \ell = 1, \dots, L\ . \tag{2.83}$$

Proof. First, we determine the Jacobian of the weighted input $\mathbf{z}^{(\ell)}$ at layer $\ell \in \{1, \dots, L\}$ with respect to the column vectors $\mathbf{w}^{(\ell)}_{\bullet j}$ of the weights matrix $\mathbf{W}^{(\ell)} = (\mathbf{w}^{(\ell)}_{\bullet j})_{j=1}^{n_{\ell-1}}$, that is

$$\mathbf{J}_{\mathbf{w}^{(\ell)}_{\bullet j}} \mathbf{z}^{(\ell)} = \mathbf{J}_{\mathbf{w}^{(\ell)}_{\bullet j}} \left(\mathbf{W}^{(\ell)} \mathbf{x}^{(\ell-1)} + \mathbf{b}^{(\ell)} \right) = x_j^{(\ell-1)} \boldsymbol{I}_{n_\ell} \quad \forall\ j = 1, \dots, n_{\ell-1} \tag{2.84}$$

where $\boldsymbol{I}_{n_\ell}$ denotes the identity matrix of size n_ℓ. Thus, the Jacobian can also be seen as a squared matrix whose diagonal entries are all $x_j^{(\ell-1)}$. This implies $\forall\ j = 1, \dots, n_{\ell-1},\ \ell = 1, \dots, L$

$$\nabla_{\mathbf{w}^{(\ell)}_{\bullet j}} \mathcal{L} \circ f = \left(\nabla^\top_{\mathbf{z}^{(\ell)}} \mathcal{L} \circ f \cdot \mathbf{J}_{\mathbf{w}^{(\ell)}_{\bullet j}} \mathbf{z}^{(\ell)} \right)^\top = x_j^{(\ell-1)} \boldsymbol{I}_{n_\ell}\ \boldsymbol{\delta}^{(\ell)} = x_j^{(\ell-1)} \boldsymbol{\delta}^{(\ell)}\ . \tag{2.85}$$

For the biases, the Jacobian is given by $\mathbf{J}_{\mathbf{b}^{(\ell)}} \mathbf{z}^{(\ell)} = \boldsymbol{I}_{n_\ell}$ and it follows analogously $\forall\ \ell = 1, \dots, L$

$$\nabla_{\mathbf{b}^{(\ell)}} \mathcal{L} \circ f = \left(\nabla^\top_{\mathbf{z}^{(\ell)}} \mathcal{L} \circ f \cdot \mathbf{J}_{\mathbf{b}^{(\ell)}} \mathbf{z}^{(\ell)} \right)^\top = \boldsymbol{I}_{n_\ell}\ \boldsymbol{\delta}^{(\ell)} = \boldsymbol{\delta}^{(\ell)}\ . \tag{2.86}$$

□

All quantities required for the backpropagation algorithm, namely the activations of each network layer, are computed during one forward pass, and they can all be stored in the memory of a machine. Then, for computing the deltas as in Proposition 2.19, the first derivative of the activation function needs to be computed. This is computationally cheap in general, but especially for the ReLU activation function, *cf*. Definition 2.6. Finally, the computation of the desired gradients with respect to all model parameters amounts to cheap elementary multiplication operations. Thus, backpropagation can be seen as a special case of efficient *automatic differentiation* [Pas17; Dei20].

2.3 Convolutional Neural Networks

The multi layer neural networks we know up to now, *cf.* Section 2.1, perform well on many tasks, for instance on image classification of low-resolution images. However, for more complex tasks with higher dimensional inputs, *e.g.* pixel-wise image classification of high-resolution images, these traditional networks with fully connected layers fail due to the so-called *curse of dimensionality* [Bel57; Bis06]. That is why modern network architectures focus on reducing the parameters count while maintaining (or even improving) their performance on large-scale tasks. In the context of computer vision, *convolutional neural networks* (CNNs) are mostly preferred over traditional fully connected neural networks (FNNs). These two network types differ only in that matrix multiplications are replaced by *convolutions* in at least one layer.

By adding more and more layers to a multi layer neural network, the capacity of the model is increased. However, this increase in capacity is obviously limited by the computational resources. Since in traditional NNs the layers are fully connected, each neuron interacts with every neuron in the next layer. In contrast, neurons in convolutional layers typically have sparse interactions, aiming at reducing the number of parameters but, ideally, extracting the same meaningful information as in fully connected ones. The reduction of the parameters count is achieved by *parameter sharing*. Instead of using each model parameter once when computing the network layer output, in convolutional layers the same parameters are used multiple times. More precisely, each weight of a layer's set of parameters is applied to every position of the layer's input. Thus, instead of learning one set of parameters for each neuron, through convolutions one set of parameters is learned for each layer.

2.3.1 Convolution Operation

A convolution is a mathematical operation, mainly known from signal processing, whose output is derived from two given functions by integration.

Definition 2.21. *Convolution Operation [Goo16; Cal20]*
Let g and h be some integrable functions on $\mathbb{R}$. The convolution operation is denoted by $*$ and it is defined as

$$(g * h)(t) := \int_{\mathbb{R}} g(u)h(t-u)du \ . \tag{2.87}$$

Therefore, $(g*h)$ expresses how the shape of g is modified by h, and vice versa. Definition 2.21 is a very general formulation of the convolution operation. In the terminology of CNNs, g relates to some input $\mathbf{x}$ while h corresponds to some network weights $\mathbf{w}$. In addition, data on machines is usually discretized, that is why we introduce the following adaption of the convolution operation.

Definition 2.22. *Discrete Convolution Operation in Neural Networks*

1. *Let $\mathbf{x} = (x_i)_{i=1}^n \in \mathbb{R}^n,\ n \in \mathbb{N}$ be an input vector and let $\mathbf{w} = (w_k)_{k=-r}^r \in \mathbb{R}^s,\ s = 2r+1, r \in \mathbb{N}$ be a weights kernel. The discrete convolution operation is defined as*

$$z_i = (\mathbf{x} * \mathbf{w})_i := \sum_{k \in \mathbb{Z}} x_k \cdot w_{i-k} \quad \forall\ i = 1, \dots, n \tag{2.88}$$

   where $x_i = 0\ \forall\ i \notin \{1, \dots, n\}$ and $w_j = 0\ \forall\ k \notin \{-r, \dots, r\}$.

2. *The discrete convolution operation can be easily extended to two-dimensional data. Let $\mathbf{X} = (x_{ij})_{i,j=1}^n \in \mathbb{R}^{n \times n},\ n \in \mathbb{N}$ be an input matrix and let $\mathbf{W} = (w_{kl})_{k,l=-r}^r \in \mathbb{R}^{s \times s},\ s = 2r+1, r \in \mathbb{N}$*

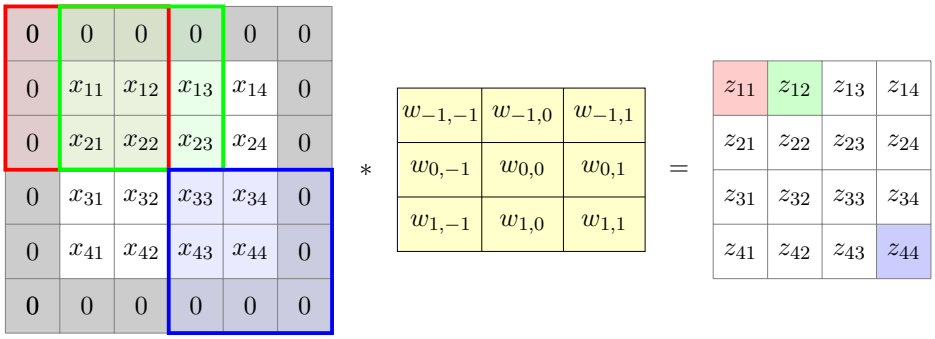


Figure 2.5: An illustrative example of a *same* convolution (or cross correlation) on 2D input data. The colored squares in $\mathbf{X}$ are also commonly referred to as *receptive field* corresponding the same colored components in $\mathbf{Z}$.

be a two-dimensional weights kernel. The two-dimensional discrete convolution operation is defined as

$$z_{ij} = (\mathbf{X} * \mathbf{W})_{ij} := \sum_{k\in\mathbb{Z}} \sum_{l\in\mathbb{Z}} x_{kl} \cdot w_{i-k,j-l} \quad \forall\ i,j = 1,\dots,n\ , \tag{2.89}$$

where $x_{ij} = 0\ \forall\ i,j \notin \{1,\dots,n\}$ *and* $w_{kl} = 0\ \forall\ k,l \notin \{-r,\dots,r\}$.

Remark 2.23. By convention, the kernel $\mathbf{W}$ for the two dimensional (2D) convolution is a quadratic matrix of odd size. This convolution operation is commonly applied to images since they are typically available as 2D data, that is the image height and image width. For the sake of simplified notation, we consider the inputs $\mathbf{X}$ to be squared as well in the 2D case. Nonetheless, the discrete convolution can also be applied on even more dimensional data and without restrictions on the input's or kernel's shape.

2.3.2 Convolution on Image Data

In practice, $\mathbf{X}$ could be an input image to a neural network, *e.g.* $\mathbf{X} \in [0,1]^{n\times n}$ for a normalized gray-scaled image of resolution $n \times n$ pixels. Such operations as in Definition 2.22 are called "same" convolutions as their input and output are of the same size, *i.e.* $\mathbf{X} \in \mathbb{R}^{n\times n}$ and $\mathbf{Z} = (\mathbf{X} * \mathbf{W}) \in \mathbb{R}^{n\times n}$. To this end, the input and the kernel are extended with zeros, which is referred to as *zero padding*.

Moreover, the kernel $\mathbf{W}$ is usually chosen to be of substantially smaller size than $\mathbf{X}$, *i.e.* $s \ll n$. Together with the commutativity and translation invariance of convolutions, this operation is usually implemented on machines for all $i,j = 1,\dots,n$ as

$$(\mathbf{X} * \mathbf{W})_{ij} = \sum_{k,l\in\mathbb{Z}} x_{i+k,j+l} \cdot w_{kl} = \sum_{k=-r}^{r} \sum_{l=-r}^{r} x_{i+k,j+l} \cdot w_{kl}\ , \tag{2.90}$$

cf. Figure 2.5 for an illustrative example of a convolution. In this light, all the elements in $\mathbf{X}$ having an effect on the value of pixel $z \in (\mathbf{X} * \mathbf{W})$ are considered as *receptive field* of z.

Note that the weighted sum in Equation (2.90) is computed more efficiently compared to Equation (2.89) as the number of operations is reduced (if $s < n$). Considered strictly, Equation (2.90) is the *cross correlation* [Cal20], which is the same as convolution but with the kernel $\mathbf{W}$ being flipped

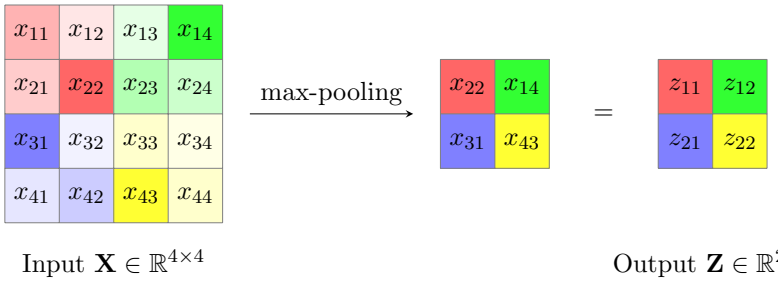


Figure 2.6: An illustrative example of a max-pooling operation on 2D input data. Higher color intensities indicate higher values. In practice, a pooling function is usually applied to a network layer's activation map, which is down sampled in that way and passed as input to the next layer.

on both axis. In deep learning terminology, this implementation is called convolution anyway since learning algorithms will then simply learn a kernel that is flipped relative to the kernel without flipping [Goo16].

2.3.3 Pooling Operation

A typical convolutional layer is additionally accompanied by a pooling function, which reduces the spatial dimension of its input by summarizing and selecting some essential features within a neighborhood [Cal20]. In convolutional neural networks *max-pooling* is mostly applied. This pooling strategy can be extended to multiple dimensions and can be applied to discretized data. In practice on image data, the input is partitioned into non-overlapping neighborhoods, almost always chosen to be of size 2×2. Within each such neighborhood of data points, the maximum value is then pooled [Zho88], *i.e.* for some input $\mathbf{X} \in \mathbb{R}^{n\times n}$, $n \in \{2r : r \in \mathbb{N}\}$, the pooling output $\mathbf{Z} \in \mathbb{R}^{(n/2)\times(n/2)}$ has the entries

$$z_{ij} = \max_{k,l\in\{1,2\}} \{x_{2(i-1)+k,2(j-1)+l}\} \quad \forall\, i,j = 1,\ldots,n/2\ . \tag{2.91}$$

This operation reduces the input size by 75%, see also Figure 2.6 for an illustrative max-pooling example. Note that the pooling operation usually follows after applying a convolution and an activation function, thus the input to the pooling function is an activation map. Moreover, different partition sizes could also be applied even though this is very uncommon in convolutional neural networks.

While convolutions extract features from input signals, max-pooling selects the most representative value. The main intuition is that features are important when they are active. Moreover, as max-pooling functions are only sensitive to the maximum value, they are approximately invariant to translations of the input, *i.e.* most pooled outputs are not affected by small changes to the input. Hence, max-pooling provides information whether one particular feature is present rather than where exactly this feature is [Goo16], which is a useful property particularly in image classification. Other pooling strategies include for instance average pooling or minimum pooling, which are, however, not commonly used in practice.

We provide a summary of the building blocks of a convolutional neural network in Figure 2.7 in form of a very simplified CNN.

2.3.4 Universal Approximators for Continuous Functions

One important question, which we have not discussed yet, is whether CNNs have the universal approximation property, *cf*. Definition 2.9. Some works provide answers to that question such as *e.g.*

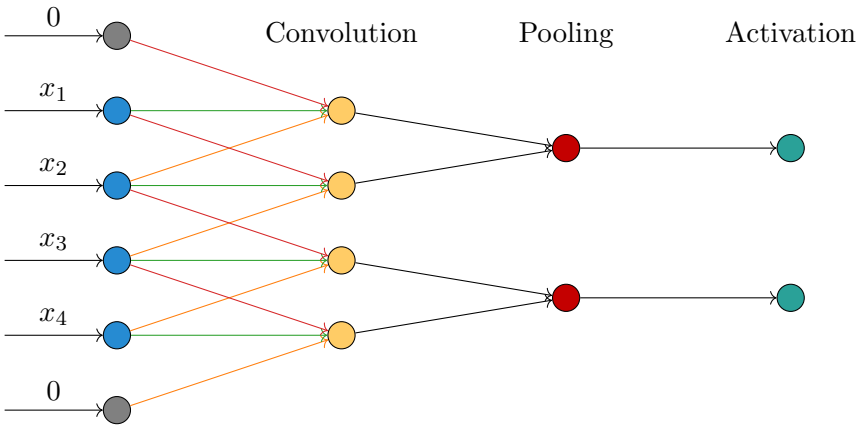


Figure 2.7: A simplified convolutional neural network consisting of a (same) convolutional layer, a pooling layer and an activation function. Note that the colors of the edges connecting the input and convolutional layer indicate shared weights.

[Pet19]. In the latter work, the authors introduce a connection between fully connected NNs and convolutional NNs by considering projections of the input onto one single component of the input. More precisely, they show that every FNN can be transformed into a CNN and vice versa such that these two types can be considered equivalent in terms of approximation capabilities, *e.g.* as presented in Theorem 2.11. In another work, the authors show explicitly that fully convolutional neural networks are universal approximators for $C(\Omega), \Omega \subset \mathbb{R}^n$ being a compact set, given sufficient depth of the CNN [Zho20], *i.e.* CNNs are able approximate any continuous function on Ω to any degree of accuracy. In analogy to the classical universal approximation theorem by [Cyb89], the author of [Yar21] presents a result given sufficient width of the CNN. He shows that any continuous and translation-equivariant function can be approximated by CNNs if no pooling operations are incorporated. In the same work, he also proves that CNNs with pooling operations are able to approximate any continuous function. For further details on the universal approximation property with respect to CNNs, we refer the reader to the outlined works.

In general, the whole field of research in computer vision has seen spectacular advances through the introduction of CNNs. Such models are efficient to train while being highly expressive. Therefore, the development of convolutional neural networks has not only impacted the task of image classification, but also led to successful approaches on more complex computer vision tasks such as semantic segmentation, which had been considered to be intractable for a long time.

Semantic Segmentation

Semantic segmentation is the computer vision task of assigning each pixel in an image to a certain class. Hence, objects are classified at pixel level, providing the finest possible localization level of objects within image dimensions. For this task, deep learning methods have been successfully employed in recent years, particularly convolutional neural networks contributed greatly to that field's progress. In this chapter, we first present possible use cases of semantic segmentation, followed by an overview of corresponding public datasets. Then, we briefly review the evolution of neural network architectures for semantic segmentation, including recent state-of-the-art, by highlighting their key contributions. Finally, we discuss the main performance metrics in order evaluate semantic segmentation.

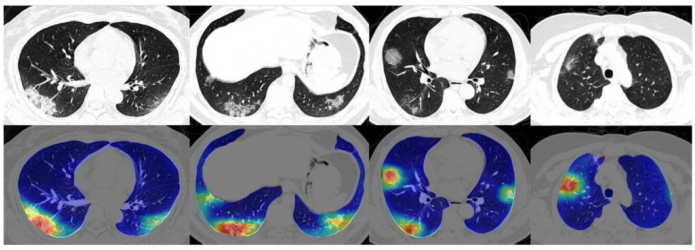
3.1 Real-World Applications

The general objective of semantic segmentation is to gain a pixel-level *understanding* of contents shown in images. This includes information on what objects are present in a given image, just like in image classification. Additionally, semantic segmentation includes information on where in the scene these objects appear as well as what shapes they have. Therefore, this computer vision task plays a central role in a wide range of applications as it enables machines to have precise visual perception.

Such applications are already partly available in our everyday lives. For instance, camera apps on modern smartphones have a portrait mode based on semantic segmentation. The portrait mode keeps the subject in the scene sharp while the background is blurred. Often this effect is not achieved by the camera lens but by software, which automatically separates foreground and background to artificially create the depth-of-field effect.

There is also great potential of computer vision in the healthcare sector. More concretely, **medical diagnosis** is heavily based on analysis of image data captured by X-ray or computer tomography (CT) scans. Semantic segmentation is one promising approach in this regard to facilitate and quicken the process of diagnosis besides helping to prevent misdiagnosis. This technique is not supposed to replace the tasks of doctors but to support their work. A recent important use case due to the ongoing COVID-19 pandemic is, for instance, identifying infected patients and quantifying the disease burden by analyzing thoracic CT scans (chest scans to analyze the lungs), see Figure 3.1 (a).

Another real-world application, which increasingly gained attention in past years, is **automated driving**. This is not a futuristic vision anymore, but part of the world's upcoming and most technological advances. In this light, semantic segmentation as perception module represents a crucial building block in such a system based on artificial intelligence (AI). Sensors like cameras capture a car's environment, producing images that are then analyzed by the AI (ideally in real-time) to control the vehicle. Gaining a reliable understanding of street scenes, *i.e.* knowing what and where objects appear, see Figure 3.1 (b), is a critical prerequisite to enable self-driving cars to navigate safely in



(a) COVID-19 disease detection, source: [Goz20]

(b) Perception in street scenes, source: [Cor16]

Figure 3.1: Two exemplary real-world use cases of semantic segmentation. (**a**) Thoracic CT scans to identify COVID-19 infected patients and to quantify the follow-up of disease. Top row shows scans classified as positive for the virus, bottom row depicts the corresponding pixel-wise abnormality scores as heatmap. (**b**) A typical urban street scene image captured by a camera from a driver's perspective. The overlayed colors indicate the corresponding pixel-wise object labels, which enable self-driving cars to perceive their surroundings.

everyday traffic. Currently, this technology is heavily driven by industry and therefore represents the most demanded and popular use case of semantic segmentation.

3.2 Public Datasets and Benchmarks

Deep learning requires large amount of data in order to work properly. This includes capturing images on the one hand and annotating them on the other. In this context, there exists relatively few semantic segmentation data as the manual labeling effort is extremely costly. Nonetheless, mainly due industrial interest, several high-quality datasets tackling specific tasks have been made publicly available in the past years. Some of them are also accompanied with a public benchmark, providing a unified evaluation framework and accelerating progress of research. In the following, we present a selection of different semantic segmentation datasets based on their popularity and their impact in the field of research.

General Scene Understanding. To understand visual scenes involves recognizing and localizing several objects that are captured in images. This is accomplished by determining characteristics of scenes and attributes of objects. Microsoft's *Common Objects in Context (COCO)* [Lin14] is a large-scale dataset containing photos of everyday objects in their natural context. Another such large-scale dataset with even more scenes is *PASCAL Visual Object Classes (VOC)* [Eve15], consisting of annotated photos collected from the web. Both datasets are well-known in the computer vision community and also greatly contribute to the progress of semantic segmentation as most methods report results on the respective benchmarks. An even broader collection of images in terms of scenes and object categories is available in MIT's *ADE20k* [Zho17], which is also accompanied with a public benchmark.

Medical Diagnosis. In many current clinical processes, medical images are studied for diagnosing diseases. This is often time-consuming and might exceed available resources, possibly leading to delays of life-saving diagnoses. Deep learning has the potential to support this task by automating medical diagnoses. To this end, deep learning algorithms can be trained with data provided by human experts such as doctors. The multimodal *Brain Tumor Image Segmentation (BRATS)* [Men15] dataset

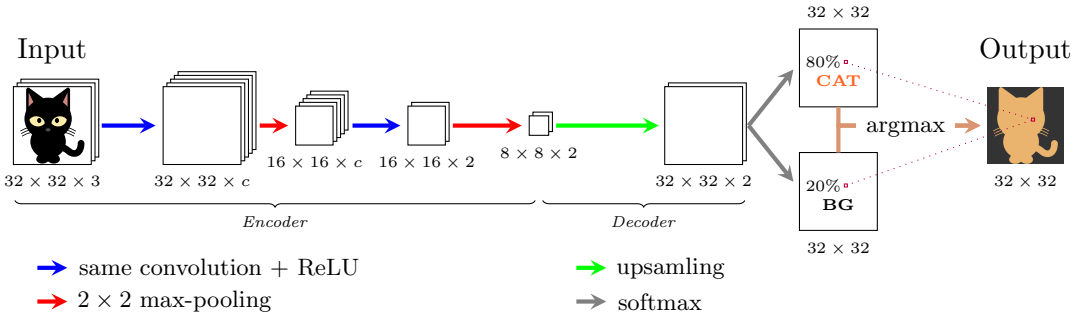


Figure 3.2: A very simplified illustration of a fully convolutional like neural network for the semantic segmentation of an RGB image and two classes (cat and background). The depicted network consists of two convolutional layers, each followed by a max-pooling operation.

provides examples of magnetic resonance images (MRIs) of brains with gliomas (type of tumor). Tumor structures can vary in size, shape, location or severity, which makes the detection of tumors difficult even for human experts. On the corresponding BRATS benchmark, semantic segmentation has demonstrated promising results in order to accelerate diagnosis processes by doctors. Another similar public dataset focusing on the heart function is provided by the *Automated Cardiac Diagnosis Challenge (ACDC)* [Ber18]. Here, MRIs are used to diagnose muscle diseases such as myocardial infarction. As the coronavirus has drastically overwhelmed the world, deep learning based diagnostic solutions have recently gained an increased interest, for which several *COVID-19 datasets* comprising CT scans have been created, see *e.g.* [Afs21; Rah21; Sar21]. Rapid and automatic diagnoses based on semantic segmentation are highly desired in order to slow down spreading the disease.

Automated Driving. Most existing public semantic segmentation datasets are used to advance the development of self-driving cars. Deep learning in general has taken over the major tasks of automated driving, where semantic segmentation is mainly employed for perception. One of the first and most popular datasets is *KITTI* [Gei12], containing hours of videos of traffic scenes with data from a variety of sensors. Although the latter dataset rather focuses on sensor fusion, a few semantic segmentation labels are still available. Probably the most prominent semantic segmentation dataset is Daimler's *Cityscapes* [Cor16], which contains high quality images of urban street scenes and their corresponding pixel-level annotations. Moreover, the images show street scenes of varying complexity in several European cities. Nearly every state-of-the-art semantic segmentation model is applied on the accompanied Cityscapes benchmark, therefore underlining the importance of the dataset in research but also in industry. Another mentionable dataset is *Berkeley Deep Drive (BDD) 100k* [Yu20] since it is the largest and most diverse dataset for automated driving up to date. This is particularly important in order to extensively test the robustness of potential deep learning algorithms.

3.3 Evolution of Neural Network Architectures

Because of their success on image classification problems, CNNs are also the preferred type of neural network for semantic segmentation. The basic architecture consists of an *encoder*, which produces a low-resolution representation of the input, and a *decoder*, which projects the compressed representation of the input back onto full pixel space. In the following, some semantic segmentation models of past years and their major contributions to recent state-of-the-art CNNs are presented.

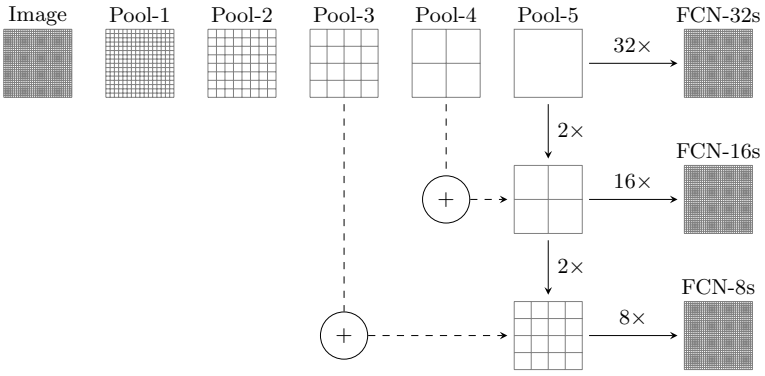


Figure 3.3: Skip connections in the fully convolutional neural network (FCN), which define its different variants FCN-32s, FCN-16s and FCN-8s. Here, the output of a FCN network layer is shown as grid, where the grid size depicts the spatial coarseness, *i.e.* feature representations in deeper layers contain less location information. Instead of upsampling feature maps in one step (solid line, upsampling factor 32 for FCN-32s), outputs from more shallow layers can also be combined with the final prediction. This results in smaller upsampling factors and potentially finer details in the final segmentation. Source: [Lon15]

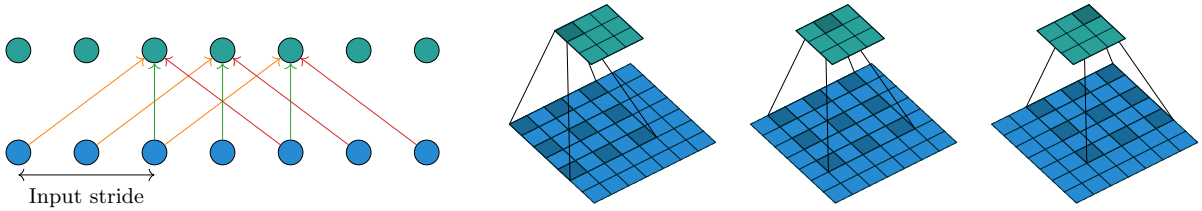
Fully Convolutional Network. Many semantic segmentation models build upon CNNs that were originally designed for image classification and already have proven to successfully extract meaningful features from image data. Some well-known neural network designs are for instance AlexNet [Kri12], VGGNet [Sim15] or GoogleNet [Sze15]. These mentioned models were the first that have been transformed to network architectures consisting only of convolutional layers, *cf.* Figure 3.2 for an basic overview of that network design. More precisely, in the *Fully Convolutional Network* (FCN) [Lon15] fully connected layers are replaced by 1×1 convolutions, allowing networks to process images of any size. To adapt to image segmentation, which requires dense pixel predictions, spatial low-resolution feature maps are upsampled, *e.g.* via bilinear interpolation.

However, image classification networks apply successive subsampling layers, such as pooling, *cf.* Section 2.3.3, which reduce the resolution of feature maps until a global prediction on full image level can be obtained. This is in contrast to semantic segmentation, where such contextual information is required for each image pixel. In order to combine hierarchies of features and their spatial location in fully convolutional networks, connections combining outputs of early network layers and the final prediction layer were included in FCN. These links are known as "skip connections", see Figure 3.3, which preserve location information from feature representations in shallow layers as they contain more global context than deeper ones after downsampling. This idea of skip connections, but generally also of FCN, had a significant impact on the progress in semantic segmentation, since this concept initiated the development of many subsequent state-of-the-art network architectures.

Atrous Convolution. The use of fully convolutional networks led to a significant performance gain in semantic segmentation by repurposing network architectures from image classification to dense pixel prediction. However, location information of features is usually lost when pooling is employed multiple times. One way to mitigate this issue is by adding skip connections as in FCN, see again Figure 3.3.

Another way is to replace convolution operations in existing network architectures with *atrous* convolutions. The latter operation can be formalized in analogy to Equation (2.90) as

$$(\mathbf{X} * \mathbf{W})_{ij} = \sum_{k=-r}^{r} \sum_{l=-r}^{r} x_{i+\alpha k, j+\alpha l} \cdot w_{k,l} \ , \tag{3.1}$$



(a) 1D atrous convolution **(b)** 2D atrous convolutions

Figure 3.4: An illustration of one-dimensional (1D) and two-dimensional (2D) atrous convolution. **(a)** 1D example where the kernel is of size 3 and an input stride of 2 is employed, *i.e.* kernel weights are applied to every second input point when performing a convolution. **(b)** 2D example with a 3×3 kernel and input stride 2. Source: [Dum16]

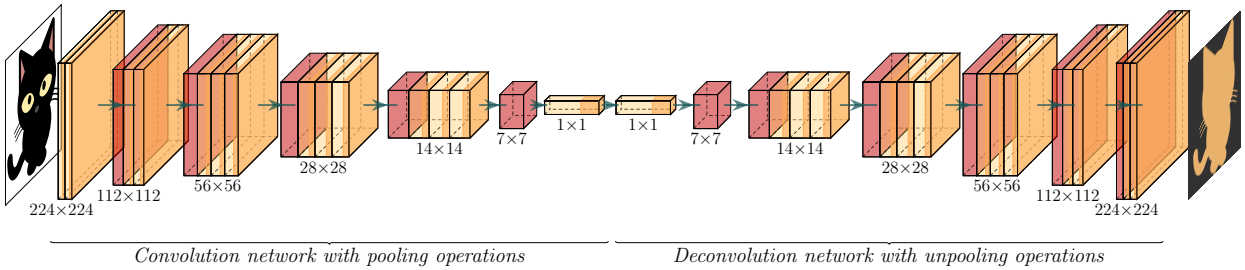


Figure 3.5: Architecture of an entire semantic segmentation neural network that incorporates a deconvolutional network. The deconvolution network is added on top of the classification model, the convolution network, and consists of multiple layers performing unpooling and deconvolutions for dense pixel predictions. Source: [Noh15]

where $\alpha \in \mathbb{N}$ is called *input stride*, *i.e.* the kernel $\mathbf{W}$ is applied to every α-th component of input $\mathbf{X}$, *cf.* Figure 3.4. The multi scale context aggregation module introduced in [Yu16] is based on atrous convolutions with different input strides. Via this module, the receptive field of pixels in feature maps is significantly expanded, which consistently improves semantic segmentation accuracy when plugged into FCNs.

This paved the way towards more sophisticated neural network architectures for semantic segmentation, such as the *DeepLab* models. Its first version employs atrous convolutions in CNNs and additionally conditional random fields (CRF) [Che18a]. Traditionally, CRFs are probabilistic graphical models used to smooth noisy segmentations. In DeepLabV1, fully connected CRFs are introduced to recover detailed local structures of objects, which in the end substantially improved classification accuracy at object boundaries. Although the CRF in the decoder showed promising results, they were soon outperformed by network-like approaches.

Deconvolutional Network. Instead of upsampling coarse activation maps via interpolation, this step can also be performed by neural networks whose parameters can be learned for dense prediction. These upsampling networks are known as *Deconvolutional Networks* [Noh15], *cf.* Figure 3.5 for such an entire semantic segmentation model architecture. Deconvolutional networks serve as object structure generator from low-resolution feature representations. Their architecture is typically a mirrored version of the employed convolutional part but with *unpooling* and *deconvolution* operations.

Unpooling is the reverse operation of pooling, in that way reconstructing activation maps to their original size. To this end, locations of maximum values during max-pooling are stored and then

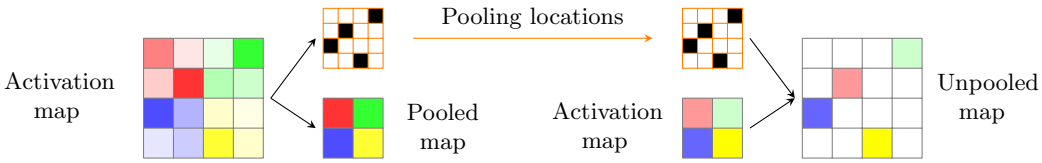


Figure 3.6: An illustration of a max-unpooling operation. After max-pooling is applied, the pooling locations are stored and used during max-unpooling in the decoder. Different colors represent different neighborhoods, higher color intensities indicate higher values.

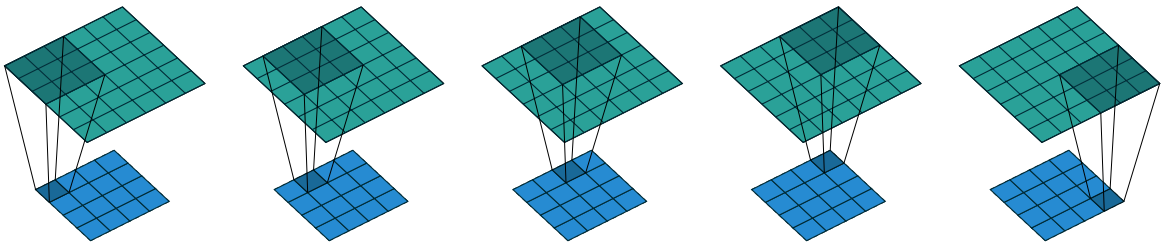


Figure 3.7: An illustration of deconvolution operations, which are used in neural networks to densify sparse feature maps. Instead of connecting multiple feature inputs to one single output like in convolutions, deconvolutions associate one single input to multiple outputs. In practice, the obtained output is usually cropped to keep the size identical to the input. Note that these operations are also called *transposed* or *fractionally strided* convolutions since they can reformulate as conventional convolutions with padding. Source: [Dum16]

recovered during max-unpooling to place activations back to the locations according to the mirrored layer in the convolutional part, see Figure 3.6 for an illustration. Unpooled activation maps are sparse, which however can be densified by deconvolutions[1]. These are convolution-like operations, but in place of connecting multiple feature inputs to one single output, deconvolutions associate one single input to multiple outputs, see Figure 3.7 for an illustration. Therefore, deconvolution kernels capture different levels of details in order to reconstruct shapes of objects at full resolution.

Learning upsampling via deconvolution networks demonstrated to help classifying fine details of objects in semantic segmentation. Prominent models based on deconvolutional networks are U-Net [Ron15] and SegNet [Bad17]. These two network architectures also employ mirrored versions of the encoder as decoder. However, this approach faces computational limitations when dealing with large-scale datasets. That is why modern architectures for semantic segmentation still perform upsampling via deconvolutional layers, but with as few of those as possible.

Multi scale Context. One promising approach in order to better capture global context of objects is pooling feature maps at different scales. To this end, the Pyramid Scene Parsing Network (PSPNet) [Zha17] introduced a *pyramid pooling* (PP) module with kernels varying in size, aggregating sub-regions covering up to the whole feature map, see Figure 3.8. These pooled outputs are then fused, in that way exploiting local and global context information. In PSPNet the pyramid pooling module is applied to the encoder's output, yielding outstanding results on many semantic segmentation benchmarks at the time it was published.

[1]The term deconvolution is widely known as the operation to undo a convolution. This is however quite different to deconvolutions in deep learning terminology, which is why these operations are also often referred to as *transposed* or *fractionally strided* convolutions for technical correctness.

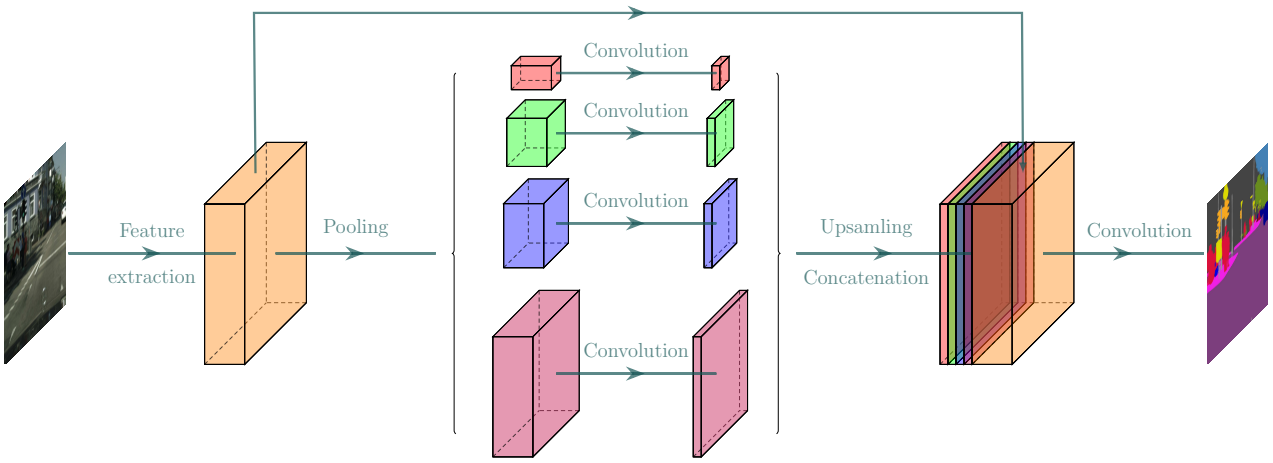


Figure 3.8: An overview of PSPNet including the pyramid pooling module. In this module, pooling is applied at different scales, followed by convolutions and a concatenation layer. Source: [Zha17]

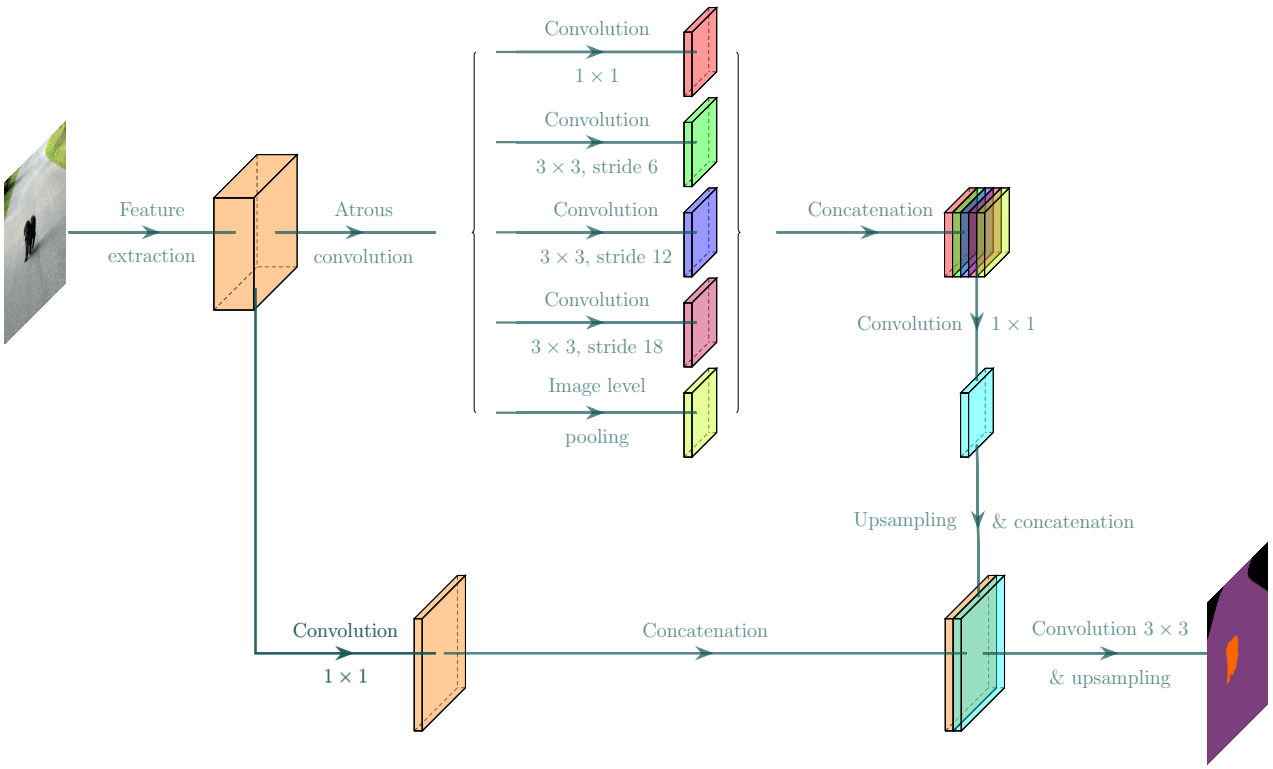


Figure 3.9: An overview of DeepLabV3+, where an encoder-decoder architecture is employed. In the encoder, atrous convolutions with different input strides in combination with image level pooling are applied to extract contextual information, which are then fused and passed to the decoder to refine segmentation at object boundaries. Source: [Che18b]

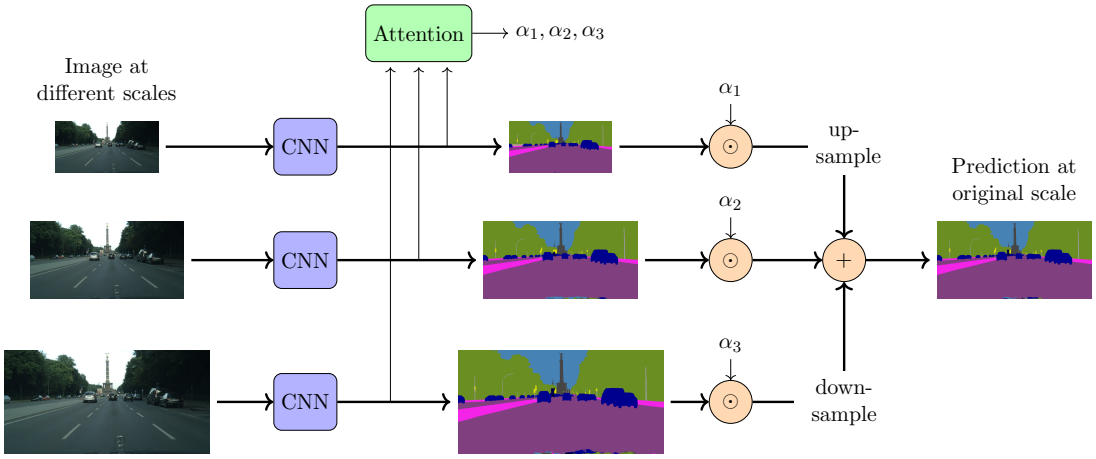


Figure 3.10: A basic overview of a multi scale attention semantic segmentation network. One input image at different resolutions is fed through the same convolutional neural network (CNN), yielding outputs at different scales. For the final segmentation, attention maps $\alpha_1, \alpha_2, \alpha_3$, which are learned for each scale, are then used to combine the different predictions via a pixel-wise weighted sum. Note that in this illustration semantic segmentation masks depict the CNN's outputs, but in practice attention is usually applied to softmax probabilities.

Spatial pyramid pooling also helped to further improve the performance of the DeepLab model. Its latest version DeepLabV3+ [Che18b] captures global context via *atrous spatial pyramid pooling* (ASPP), where parallel atrous convolutions with different input strides are applied. Similar to the pyramid pooling module in PSPNet, ASPP is placed on top of the encoder, thereafter followed by fusing the ASPP outputs. Moreover, this extracted information is additionally concatenated with a feature map of a shallow layer before upsampling and generating the final prediction, see Figure 3.9. This is a novel part of the decoder in DeepLabV3+, that demonstrated remarkable performance gains by topping the leaderboards on many benchmarks, in particular on the challenging Cityscapes benchmark. DeepLabV3+ had a significant impact on semantic segmentation in general and it is still a very popular model as many recent segmentation networks, including recent state-of-the-art, build upon this architecture.

Multi Scale Attention. Another common way to extract mutli-scale features is to feed CNNs with multiple resized images, which are then merged for the final prediction. The intuition behind this approach is that fine details of objects are better classified with scaled up images and more local context, while structures of large objects are better classified with scaled down images and more global context. To this end, *attention* maps are pixel-wise weightings in order to combine predictions at different scales and the parameters of such a model producing these attention maps can be learned as well, see Figure 3.10.

Using attention to combine multiple scaled predictions was examined early in [Che16]. Besides improving segmentation performance, attention also provides useful visualizations of the importance of features at each image location. Since in times in which deep neural networks are considered more and more as black boxes, such diagnostic tools become ever crucial for the deployability of deep learning.

For large-scale datasets such as Cityscapes, the attention-based segmentation approach has been successfully adopted as well. In [Tao20], the authors propose an efficient *hierarchical multi scale attention* mechanism. Instead of learning a fixed set of scales, their proposed module allows to learn relative attention masks between adjacent scales. This enables a significant gain in memory efficiency

		True / Actual	
		Positive	Negative
Predicted	Positive	True Positive	False Positive
	Negative	False Negative	True Negative

Figure 3.11: Confusion matrix for binary classification.

during training and combination of varying, non-fixed scales during inference. At the time of writing, hierarchical multi scale attention segmentation yields the best performing network architecture on the Cityscapes benchmark.

Note that while convolutional neural networks dedicated for semantic segmentation rapidly evolved, networks for image classification have been developed at a similar pace. Therefore, more advanced network backbones became available and updated the encoder of segmentation networks over the time, which generally improved segmentation performance. In this regard, the most commonly-used network backbone is by far ResNet [He16a], which eases the training of ever deeper networks via residual learning. Just recently, the HRNet [Yua20] gained substantial popularity by considering object-contextual representations (OCR). The latter module takes the relation between positions of objects and their context into account for the final segmentation prediction. OCR differentiates context at object level, while other conventional multi scale modules, such PP or ASPP, only differentiates pixels by means of spatial locations.

3.4 Evaluation Metrics

Semantic segmentation is a multi class pixel-wise image classification problem. This allows for evaluating the performance of a semantic segmentation model by means of a so-called *confusion matrix* [Ste97; Faw06; Has09]. This matrix summarizes the number of correct and incorrect predictions with respect to each class. In particular, a confusion matrix displays which types of errors have been made by a classification model. To this end, each row corresponds to a predicted class and each column to an actual class label. This yields a squared matrix of size K, where $K \in \mathbb{N}$ denotes the number of classes.

In its simplest form in the case of binary classification, a confusion matrix provides the following quantities, *cf*. Figure 3.11:

- **true-positives** (TP): number of predictions where the model *correctly predicts* the positive class
- **false-positives** (FP): number of predictions where the model *incorrectly predicts* the negative class as positive (type I error)
- **false-negatives** (FN): number of predictions where the model *incorrectly predicts* the positive class as negative (type II error) and
- **true-negatives** (TN): number of predictions where the model *correctly predicts* the negative class

In binary classification, the terms *positive* and *negative* are typically used to denote the two classes to discriminate, where positive often refers to the class of interest. These quantities can be adapted

		True / Actual			
		Class 1	Class 2	...	Class K
Predicted	Class 1	TP_1	FN_{12}	...	FN_{1K}
	Class 2	FP_{12}	TP_2	$\ddots$	FN_{2K}
	$\vdots$	$\vdots$	$\ddots$	$\ddots$	$\vdots$
	Class K	FP_{1K}	FP_{2K}	...	TP_K

Figure 3.12: Confusion matrix for multi class classification.

to multi class classification by considering the introduced quantities for each individual class. Furthermore, in semantic segmentation each pixel is also treated individually, which yields for all classes $k = \{1, \dots, K\}$ the quantities, *cf*. Figure 3.12,

$$\mathrm{TP}_k = \sum_{i=1}^{N} \mathbb{1}_{y^{(i)}=k} \mathbb{1}_{\hat{y}^{(i)}=k} \tag{3.2}$$

$$\mathrm{FP}_k = \sum_{\substack{k'=1 \\ k' \neq k}}^{K} \mathrm{FP}_{kk'} = \sum_{\substack{k'=1 \\ k' \neq k}}^{K} \sum_{i=1}^{N} \mathbb{1}_{y^{(i)}=k'} \mathbb{1}_{\hat{y}^{(i)}=k} = \sum_{i=1}^{N} \mathbb{1}_{y^{(i)} \neq k} \mathbb{1}_{\hat{y}^{(i)}=k} \tag{3.3}$$

$$\mathrm{FN}_k = \sum_{\substack{k'=1 \\ k' \neq k}}^{K} \mathrm{FN}_{kk'} = \sum_{\substack{k'=1 \\ k' \neq k}}^{K} \sum_{i=1}^{N} \mathbb{1}_{y^{(i)}=k} \mathbb{1}_{\hat{y}^{(i)}=k'} = \sum_{i=1}^{N} \mathbb{1}_{y^{(i)}=k} \mathbb{1}_{\hat{y}^{(i)} \neq k} \tag{3.4}$$

where $y^{(i)} \in \{1, \dots, K\}$ denotes the label of the i-th ground truth pixel and $\hat{y}^{(i)} \in \{1, \dots, K\}$ the corresponding class prediction in a semantic segmentation dataset with $N \in \mathbb{N}$ annotated pixels.

Given those quantities, various performance metrics can be computed in order to determine how well a semantic segmentation model performs. In this context, *precision* [Ken55; Ols08] measures the percentage of predictions that are actually correct, *i.e.*

$$\mathrm{precision}_k = \frac{\mathrm{TP}_k}{\mathrm{TP}_k + \mathrm{FP}_k} \quad \forall\ k \in \{1, \dots, K\}\ , \tag{3.5}$$

while *recall* [Ken55; Ols08; Has09] measures the percentage of actual instances that are correctly identified, *i.e.*

$$\mathrm{recall}_k = \frac{\mathrm{TP}_k}{\mathrm{TP}_k + \mathrm{FN}_k} \quad \forall\ k \in \{1, \dots, K\}\ . \tag{3.6}$$

Another popular performance metric, particularly in semantic segmentation, is the *intersection over union* (IoU, also known as *Jaccard Index* [Jac12]), which measures the overlap between prediction and ground truth relative to the union of prediction and ground truth, *i.e.*

$$\mathrm{IoU}_k = \frac{\mathrm{TP}_k}{\mathrm{TP}_k + \mathrm{FP}_k + \mathrm{FN}_k} \quad \forall\ k \in \{1, \dots, K\}\ . \tag{3.7}$$

As overall metric the class-wise IoU is typically averaged over all classes, yielding the *mean intersection over union* [Cor16]

$$\mathrm{mIoU} = \frac{1}{K} \sum_{k=1}^{K} \mathrm{IoU}_k\ . \tag{3.8}$$

Bibliography Part I

[Jen06] J. L. W. V. Jensen. "Sur les fonctions convexes et les inégalités entre les valeurs moyennes (in French)". In: *Acta Mathematica* 30 (1906), pp. 175–193. DOI: 10.1007/BF02418571 (cit. on p. 16).

[Jac12] Paul Jaccard. "The distribution of the flora in the alpine zone. 1". In: *New phytologist* 11.2 (1912), pp. 37–50 (cit. on p. 40).

[Fis25] R.A. Fisher. *Statistical methods for research workers*. Edinburgh Oliver & Boyd, 1925 (cit. on p. 18).

[McC43] Warren S McCulloch and Walter Pitts. "A logical calculus of the ideas immanent in nervous activity". In: *The bulletin of mathematical biophysics* 5.4 (1943), pp. 115–133 (cit. on pp. 1, 7).

[Sør48] T. Sørenson. *A Method of Establishing Groups of Equal Amplitude in Plant Sociology Based on Similarity of Species Content and Its Application to Analyses of the Vegetation on Danish Commons*. Biologiske skrifter. I kommission hos E. Munksgaard, 1948 (cit. on p. 21).

[Kul51] S. Kullback and R. A. Leibler. "On Information and Sufficiency". In: *The Annals of Mathematical Statistics* 22.1 (1951), pp. 79–86. DOI: 10.1214/aoms/1177729694 (cit. on p. 16).

[Ken55] A. Kent et al. "Machine literature searching VIII. Operational criteria for designing information retrieval systems". In: *American Documentation* 6 (1955), pp. 93–101 (cit. on p. 40).

[Bel57] R. Bellman, Rand Corporation, and Karreman Mathematics Research Collection. *Dynamic Programming*. Rand Corporation research study. Princeton University Press, 1957. ISBN: 9780691079516. URL: https://books.google.de/books?id=wdtoPwAACAAJ (cit. on pp. 15, 26).

[Ros58] F. Rosenblatt. "The perceptron: A probabilistic model for information storage and organization in the brain." In: *Psychological Review* 65.6 (1958), pp. 386–408. ISSN: 0033-295X. DOI: 10.1037/h0042519. URL: http://dx.doi.org/10.1037/h0042519 (cit. on p. 7).

[Kul59] Solomon Kullback. *Information Theory and Statistics*. John Wiley & Sons, 1959 (cit. on p. 16).

[Ros61] Frank Rosenblatt. *Principles of neurodynamics. perceptrons and the theory of brain mechanisms*. Tech. rep. Cornell Aeronautical Lab Inc Buffalo NY, 1961 (cit. on p. 1).

[Dre62] Stuart Dreyfus. "The numerical solution of variational problems". In: *Journal of Mathematical Analysis and Applications* 5.1 (1962), pp. 30–45. ISSN: 0022-247X. DOI: https://doi.org/10.1016/0022-247X(62)90004-5 (cit. on pp. 24, 25).

[Pin64] Mark S Pinsker. *Information and information stability of random variables and processes*. Holden-Day, 1964 (cit. on p. 16).

[Wil72] Dennis L. Wilson. "Asymptotic Properties of Nearest Neighbor Rules Using Edited Data". In: *IEEE Transactions on Systems, Man, and Cybernetics* SMC-2.3 (1972), pp. 408–421. DOI: 10.1109/TSMC.1972.4309137 (cit. on p. 2).

[New86] Whitney Newey and Daniel McFadden. "Large sample estimation and hypothesis testing". In: *Handbook of Econometrics*. Ed. by R. F. Engle and D. McFadden. 1st ed. Vol. 4. Elsevier, 1986. Chap. 36, pp. 2111–2245 (cit. on p. 18).

[Rum86] David E. Rumelhart, James L. McClelland, and CORPORATE PDP Research Group, eds. *8. Learning Internal Representations by Error Propagation*. Cambridge, MA, USA: MIT Press, 1986. ISBN: 0262132184 (cit. on pp. 1, 24, 25).

[Zho88] Zhou and Chellappa. "Computation of optical flow using a neural network". In: *IEEE 1988 International Conference on Neural Networks*. 1988, 71–78 vol.2. DOI: 10.1109/ICNN.1988.23914 (cit. on p. 28).

[Cyb89] G. Cybenko. "Approximation by superpositions of a sigmoidal function". In: *Mathematics of Control, Signals and Systems* 2.4 (Dec. 1989), pp. 303–314. ISSN: 1435-568X. DOI: 10.1007/BF02551274 (cit. on pp. 1, 10, 12, 13, 29).

[Bri90] John S. Bridle. "Probabilistic Interpretation of Feedforward Classification Network Outputs, with Relationships to Statistical Pattern Recognition". In: *Neurocomputing*. Ed. by Françoise Fogelman Soulié and Jeanny Hérault. Berlin, Heidelberg: Springer Berlin Heidelberg, 1990, pp. 227–236. ISBN: 978-3-642-76153-9 (cit. on p. 12).

[Hor91] Kurt Hornik. "Approximation Capabilities of Multilayer Feedforward Networks". In: *Neural Netwworks* 4.2 (Mar. 1991), pp. 251–257. ISSN: 0893-6080. DOI: 10.1016/0893-6080(91)90009-T (cit. on pp. 1, 14).

[Dar92] C. Darken, J. Chang, and J. Moody. "Learning rate schedules for faster stochastic gradient search". In: *Neural Networks for Signal Processing II Proceedings of the 1992 IEEE Workshop*. 1992, pp. 3–12. DOI: 10.1109/NNSP.1992.253713 (cit. on p. 22).

[Les93] Moshe Leshno et al. "Multilayer feedforward networks with a nonpolynomial activation function can approximate any function". In: *Neural Networks* 6.6 (1993), pp. 861–867. ISSN: 0893-6080. DOI: 10.1016/S0893-6080(05)80131-5 (cit. on pp. 1, 15).

[Vaa96] AW van der Vaart et al. *Weak Convergence and Empirical Processes: With Applications to Statistics*. Springer Series in Statistics. Springer, 1996. ISBN: 9780387946405 (cit. on p. 18).

[Ste97] Stephen V. Stehman. "Selecting and interpreting measures of thematic classification accuracy". In: *Remote Sensing of Environment* 62.1 (1997), pp. 77–89. ISSN: 0034-4257. DOI: https://doi.org/10.1016/S0034-4257(97)00083-7 (cit. on p. 39).

[Bob99] S.G Bobkov and F Götze. "Exponential Integrability and Transportation Cost Related to Logarithmic Sobolev Inequalities". In: *Journal of Functional Analysis* 163.1 (1999), pp. 1–28. ISSN: 0022-1236. DOI: https://doi.org/10.1006/jfan.1998.3326 (cit. on p. 16).

[Qia99] Ning Qian. "On the momentum term in gradient descent learning algorithms". In: *Neural Networks* 12.1 (1999), pp. 145–151. ISSN: 0893-6080. DOI: 10.1016/S0893-6080(98)00116-6 (cit. on p. 22).

[Jap00] Nathalie Japkowicz. "The Class Imbalance Problem: Significance and Strategies". In: *In Proceedings of the 2000 International Conference on Artificial Intelligence (ICAI.* 2000, pp. 111–117 (cit. on p. 2).

[Cha02] Nitesh Chawla et al. "SMOTE: Synthetic Minority Over-sampling Technique". In: *J. Artif. Intell. Res. (JAIR)* 16 (Jan. 2002), pp. 321–357. DOI: 10.1613/jair.953 (cit. on p. 2).

[Jap02] Nathalie Japkowicz and Shaju Stephen. "The class imbalance problem: A systematic study". In: *Intelligent data analysis* 6.5 (2002), pp. 429–449 (cit. on p. 1).

[Sha03] Jun Shao. *Mathematical Statistics*. 2nd. Springer-Verlag New York Inc, 2003 (cit. on p. 18).

[Zha03] J. Zhang and I. Mani. "KNN Approach to Unbalanced Data Distributions: A Case Study Involving Information Extraction". In: *Proceedings of the ICML 2003 Workshop on Learning from Imbalanced Datasets*. 2003 (cit. on p. 2).

[Cha04] Nitesh V. Chawla, Nathalie Japkowicz, and Aleksander Kotcz. "Editorial: Special Issue on Learning from Imbalanced Data Sets". In: *SIGKDD Explor. Newsl.* 6.1 (June 2004), pp. 1–6. ISSN: 1931-0145. DOI: 10.1145/1007730.1007733. URL: https://doi.org/10.1145/1007730.1007733 (cit. on p. 2).

[Bis06] Christopher M. Bishop. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Berlin, Heidelberg: Springer-Verlag, 2006. ISBN: 0387310738 (cit. on pp. 8, 12, 15, 21, 23–26).

[Faw06] Tom Fawcett. "An introduction to ROC analysis". In: *Pattern Recognition Letters* 27.8 (2006). ROC Analysis in Pattern Recognition, pp. 861–874. ISSN: 0167-8655. DOI: https://doi.org/10.1016/j.patrec.2005.10.010 (cit. on pp. 2, 39).

[Ros06] Jeffrey S Rosenthal. *A First Look at Rigorous Probability Theory*. 2nd. WORLD SCIENTIFIC, 2006. DOI: 10.1142/6300 (cit. on pp. 18, 19).

[Van07] Jason Van Hulse, Taghi M. Khoshgoftaar, and Amri Napolitano. "Experimental Perspectives on Learning from Imbalanced Data". In: *Proceedings of the 24th International Conference on Machine Learning*. ICML '07. Corvalis, Oregon, USA: Association for Computing Machinery, 2007, pp. 935–942. ISBN: 9781595937933. DOI: 10.1145/1273496.1273614 (cit. on p. 2).

[Ols08] David Olson and Dursun Delen. *Advanced Data Mining Techniques*. Jan. 2008. ISBN: 978-3-540-76916-3. DOI: 10.1007/978-3-540-76917-0 (cit. on p. 40).

[Den09] Jia Deng et al. "Imagenet: A large-scale hierarchical image database". In: *IEEE conference on computer vision and pattern recognition*. IEEE. 2009, pp. 248–255 (cit. on p. 1).

[Has09] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The elements of statistical learning: data mining, inference and prediction*. 2nd ed. Springer New York, 2009. URL: http://www-stat.stanford.edu/~tibs/ElemStatLearn/ (cit. on pp. 8–10, 12, 21, 39, 40).

[Kri09] Alex Krizhevsky, Vinod Nair, and Geoffrey Hinton. "CIFAR-10 (Canadian Institute for Advanced Research)". In: (2009). URL: http://www.cs.toronto.edu/~kriz/cifar.html (cit. on p. 1).

[Gib10] Josiah Willard Gibbs. *Elementary Principles in Statistical Mechanics: Developed with Especial Reference to the Rational Foundation of Thermodynamics*. Cambridge Library Collection - Mathematics. Cambridge University Press, 2010. DOI: 10.1017/CBO9780511686948 (cit. on p. 12).

[LeC10] Yann LeCun and Corinna Cortes. “MNIST handwritten digit database”. In: (2010). URL: http://yann.lecun.com/exdb/mnist/ (cit. on p. 1).

[Nai10] Vinod Nair and Geoffrey E. Hinton. “Rectified Linear Units Improve Restricted Boltzmann Machines”. In: *ICML*. 2010, pp. 807–814. URL: https://icml.cc/Conferences/2010/papers/432.pdf (cit. on p. 11).

[Pfa11] Johann Pfanzagl. *Parametric Statistical Theory*. De Gruyter, 2011. ISBN: 9783110889765. DOI: doi:10.1515/9783110889765 (cit. on p. 18).

[Gei12] Andreas Geiger, Philip Lenz, and Raquel Urtasun. “Are we ready for Autonomous Driving? The KITTI Vision Benchmark Suite”. In: *Conference on Computer Vision and Pattern Recognition (CVPR)*. 2012 (cit. on pp. 1, 33).

[Kri12] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. “ImageNet Classification with Deep Convolutional Neural Networks”. In: *Advances in Neural Information Processing Systems 25*. Ed. by F. Pereira et al. Curran Associates, Inc., 2012, pp. 1097–1105. URL: http://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks.pdf (cit. on pp. 1, 34).

[Lóp13] Victoria López et al. “An insight into classification with imbalanced data: Empirical results and current trends on using data intrinsic characteristics”. In: *Information Sciences* 250 (2013), pp. 113–141. ISSN: 0020-0255. DOI: https://doi.org/10.1016/j.ins.2013.07.007. URL: https://www.sciencedirect.com/science/article/pii/S0020025513005124 (cit. on p. 2).

[Maa13] Andrew L. Maas, Awni Y. Hannun, and Andrew Y. Ng. “Rectifier nonlinearities improve neural network acoustic models”. In: *in ICML Workshop on Deep Learning for Audio, Speech and Language Processing*. 2013 (cit. on p. 11).

[Lin14] Tsung-Yi Lin, Michael Maire, Serge Belongie, et al. “Microsoft COCO: Common Objects in Context”. In: *Computer Vision – ECCV 2014*. Springer International Publishing, 2014, pp. 740–755. ISBN: 978-3-319-10602-1 (cit. on pp. 1, 32).

[ShS14] Shai Shalev-Shwartz and Shai Ben-David. *Understanding Machine Learning - From Theory to Algorithms*. Cambridge University Press, 2014, pp. I–XVI, 1–397. ISBN: 978-1-10-705713-5 (cit. on pp. 2, 8, 9, 17, 18, 22–25).

[Cae15] Holger Caesar, Jasper Uijlings, and Vittorio Ferrari. “Joint Calibration for Semantic Segmentation”. In: *Proceedings of the British Machine Vision Conference (BMVC)*. Ed. by Mark W. Jones Xianghua Xie and Gary K. L. Tam. BMVA Press, Sept. 2015, pp. 29.1–29.13. ISBN: 1-901725-53-7. DOI: 10.5244/C.29.29 (cit. on p. 2).

[Eve15] Mark Everingham et al. “The Pascal Visual Object Classes Challenge: A Retrospective”. In: *Int. J. Comput. Vision* 111.1 (Jan. 2015), pp. 98–136. ISSN: 0920-5691. DOI: 10.1007/s11263-014-0733-5 (cit. on p. 32).

[Gir15] Ross Girshick. “Fast R-CNN”. In: *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*. Dec. 2015 (cit. on p. 1).

[Kin15] Diederik P. Kingma and Jimmy Ba. “Adam: A Method for Stochastic Optimization”. In: *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*. Ed. by Yoshua Bengio and Yann LeCun. 2015 (cit. on p. 22).

[Lon15] Jonathan Long, Evan Shelhamer, and Trevor Darrell. "Fully convolutional networks for semantic segmentation". In: *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2015, pp. 3431–3440. DOI: 10.1109/CVPR.2015.7298965 (cit. on pp. 1, 34).

[Mas15] David Masko and Paulina Hensman. "The Impact of Imbalanced Training Data for Convolutional Neural Networks". In: 2015 (cit. on p. 2).

[Men15] Bjoern H. Menze et al. "The Multimodal Brain Tumor Image Segmentation Benchmark (BRATS)". In: *IEEE Transactions on Medical Imaging* 34.10 (2015), pp. 1993–2024. DOI: 10.1109/TMI.2014.2377694 (cit. on pp. 1, 32).

[Noh15] Hyeonwoo Noh, Seunghoon Hong, and Bohyung Han. "Learning Deconvolution Network for Semantic Segmentation". In: *Computer Vision (ICCV), 2015 IEEE International Conference on*. 2015 (cit. on p. 35).

[Ron15] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. "U-Net: Convolutional Networks for Biomedical Image Segmentation". In: *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III*. Ed. by Nassir Navab et al. Springer International Publishing, 2015, pp. 234–241. ISBN: 978-3-319-24574-4. DOI: 10.1007/978-3-319-24574-4_28 (cit. on p. 36).

[Sim15] Karen Simonyan and Andrew Zisserman. "Very Deep Convolutional Networks for Large-Scale Image Recognition". In: *International Conference on Learning Representations*. 2015 (cit. on p. 34).

[Sze15] Christian Szegedy et al. "Going Deeper with Convolutions". In: *Computer Vision and Pattern Recognition (CVPR)*. 2015 (cit. on p. 34).

[Che16] Liang-Chieh Chen et al. "Attention to Scale: Scale-Aware Semantic Image Segmentation". In: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2016, pp. 3640–3649. DOI: 10.1109/CVPR.2016.396 (cit. on p. 38).

[Cor16] Marius Cordts, Mohamed Omran, Sebastian Ramos, et al. "The Cityscapes Dataset for Semantic Urban Scene Understanding". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2016 (cit. on pp. 1, 32, 33, 40).

[Dum16] Vincent Dumoulin and Francesco Visin. "A guide to convolution arithmetic for deep learning". In: *ArXiv e-prints* (Mar. 2016). eprint: 1603.07285 (cit. on pp. 35, 36).

[Gal16] Yarin Gal and Zoubin Ghahramani. "Dropout as a Bayesian Approximation: Representing Model Uncertainty in Deep Learning". In: *Proceedings of The 33rd International Conference on Machine Learning*. Vol. 48. Proceedings of Machine Learning Research. New York, New York, USA: PMLR, June 2016, pp. 1050–1059. URL: http://proceedings.mlr.press/v48/gal16.html (cit. on p. 3).

[Goo16] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016. URL: http://www.deeplearningbook.org (cit. on pp. 9, 11, 12, 20, 26, 28).

[Han16] Ramon van Handel. *Probability in High Dimension*. https://web.math.princeton.edu/~rvan/APC550.pdf. Dec. 2016 (cit. on p. 16).

[He16a] Kaiming He et al. "Deep Residual Learning for Image Recognition". In: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2016, pp. 770–778. DOI: 10.1109/CVPR.2016.90 (cit. on p. 39).

[He16b] Kaiming He et al. "Identity Mappings in Deep Residual Networks". In: *Computer Vision – ECCV 2016*. Ed. by Bastian Leibe et al. Cham: Springer International Publishing, 2016, pp. 630–645. ISBN: 978-3-319-46493-0 (cit. on pp. 1, 15).

[Kra16] Bartosz Krawczyk. "Learning from imbalanced data: open challenges and future directions". In: *Progress in Artificial Intelligence* 5.4 (Nov. 2016), pp. 221–232. ISSN: 2192-6360. DOI: 10.1007/s13748-016-0094-0 (cit. on pp. 1, 2).

[Mil16] Fausto Milletari, Nassir Navab, and Seyed-Ahmad Ahmadi. "V-Net: Fully Convolutional Neural Networks for Volumetric Medical Image Segmentation". In: *2016 Fourth International Conference on 3D Vision (3DV)* (2016), pp. 565–571 (cit. on p. 21).

[Pin16] Peter Pinggera et al. "Lost and found: detecting small road hazards for self-driving vehicles". In: *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 2016 (cit. on p. 3).

[Red16] Joseph Redmon et al. "You Only Look Once: Unified, Real-Time Object Detection". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2016 (cit. on p. 1).

[Sze16] Christian Szegedy et al. "Rethinking the Inception Architecture for Computer Vision". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2016 (cit. on p. 1).

[Wan16] S. Wang, W. Liu, J. Wu, et al. "Training deep neural networks on imbalanced data sets". In: *2016 International Joint Conference on Neural Networks (IJCNN)*. July 2016, pp. 4368–4374. DOI: 10.1109/IJCNN.2016.7727770 (cit. on p. 2).

[Yu16] Fisher Yu and Vladlen Koltun. "Multi-Scale Context Aggregation by Dilated Convolutions". In: *International Conference on Learning Representations (ICLR)*. May 2016 (cit. on p. 35).

[Zha16] C. Zhang, K. C. Tan, and R. Ren. "Training cost-sensitive Deep Belief Networks on imbalance data problems". In: *2016 International Joint Conference on Neural Networks (IJCNN)*. July 2016, pp. 4362–4367 (cit. on p. 2).

[Bad17] Vijay Badrinarayanan, Alex Kendall, and Roberto Cipolla. "SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 39.12 (2017), pp. 2481–2495. DOI: 10.1109/TPAMI.2016.2644615 (cit. on p. 36).

[Bul17] Samuel Rota Bulò, Gerhard Neuhold, and Peter Kontschieder. "Loss Max-Pooling for Semantic Image Segmentation". In: *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2017), pp. 7082–7091 (cit. on p. 2).

[He17] Kaiming He et al. "Mask R-CNN". In: *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*. Oct. 2017 (cit. on p. 1).

[Hen17] Dan Hendrycks and Kevin Gimpel. "A Baseline for Detecting Misclassified and Out-of-Distribution Examples in Neural Networks". In: *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. 2017. URL: https://openreview.net/forum?id=Hkg4TI9xl (cit. on p. 2).

[Ken17] Alex Kendall and Yarin Gal. "What Uncertainties Do We Need in Bayesian Deep Learning for Computer Vision?" In: *Advances in Neural Information Processing Systems 30*. Curran Associates, Inc., 2017, pp. 5574–5584. URL: http://papers.nips.cc/paper/7141-what-uncertainties-do-we-need-in-bayesian-deep-learning-for-computer-vision.pdf (cit. on p. 3).

[Lak17] Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. "Simple and Scalable Predictive Uncertainty Estimation using Deep Ensembles". In: *Advances in Neural Information Processing Systems 30*. Curran Associates, Inc., 2017, pp. 6402–6413. URL: http://papers.nips.cc/paper/7219-simple-and-scalable-predictive-uncertainty-estimation-using-deep-ensembles.pdf (cit. on p. 3).

[Lu17] Zhou Lu et al. "The Expressive Power of Neural Networks: A View from the Width". In: *Advances in Neural Information Processing Systems*. Ed. by I. Guyon et al. Vol. 30. Curran Associates, Inc., 2017. URL: https://arxiv.org/abs/1709.02540 (cit. on p. 15).

[Nie17] Michael Nielsen. *Neural Networks and Deep Learning*. 2017. URL: http://www.neuralnetworksanddeeplearning.com (cit. on pp. 8, 9, 23–25).

[Pas17] Adam Paszke et al. "Automatic Differentiation in PyTorch". In: *NIPS 2017 Workshop on Autodiff*. Long Beach, California, USA, 2017. URL: https://openreview.net/forum?id=BJJsrmfCZ (cit. on p. 25).

[Zha17] Hengshuang Zhao et al. "Pyramid Scene Parsing Network". In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2017 (cit. on pp. 1, 36, 37).

[Zho17] Bolei Zhou et al. "Scene Parsing through ADE20K Dataset". In: *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2017, pp. 5122–5130. DOI: 10.1109/CVPR.2017.544 (cit. on p. 32).

[Ber18] Olivier Bernard et al. "Deep Learning Techniques for Automatic MRI Cardiac Multi-Structures Segmentation and Diagnosis: Is the Problem Solved?" In: *IEEE Transactions on Medical Imaging* 37.11 (2018), pp. 2514–2525. DOI: 10.1109/TMI.2018.2837502 (cit. on p. 33).

[Bud18] Mateusz Buda, Atsuto Maki, and Maciej A. Mazurowski. "A systematic study of the class imbalance problem in convolutional neural networks". In: *Neural Networks* 106 (2018), pp. 249–259. ISSN: 0893-6080. DOI: https://doi.org/10.1016/j.neunet.2018.07.011. URL: https://www.sciencedirect.com/science/article/pii/S0893608018302107 (cit. on p. 2).

[Che18a] Liang-Chieh Chen et al. "DeepLab: Semantic Image Segmentation with Deep Convolutional Nets, Atrous Convolution, and Fully Connected CRFs". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 40.4 (2018), pp. 834–848. DOI: 10.1109/TPAMI.2017.2699184 (cit. on p. 35).

[Che18b] Liang-Chieh Chen et al. "Encoder-Decoder with Atrous Separable Convolution for Semantic Image Segmentation". In: *Proceedings of the European Conference on Computer Vision (ECCV)*. Sept. 2018 (cit. on pp. 1, 37, 38).

[DeV18] Terrance DeVries and Graham W. Taylor. *Learning Confidence for Out-of-Distribution Detection in Neural Networks*. Feb. 2018. arXiv: 1802.04865. URL: http://arxiv.org/abs/1802.04865 (cit. on p. 3).

[Gui18] Leonardo Ferreira Guilhoto. "An Overview Of Artificial Neural Networks for Mathematicians". In: *The University of Chicago Mathematics REU*. 2018. URL: http://math.uchicago.edu/~may/REU2018 (cit. on p. 14).

[Kha18] S. H. Khan et al. "Cost-Sensitive Learning of Deep Feature Representations From Imbalanced Data". In: *IEEE Transactions on Neural Networks and Learning Systems* 29.8 (Aug. 2018), pp. 3573–3587. ISSN: 2162-237X. DOI: 10.1109/TNNLS.2017.2732482 (cit. on p. 2).

[Lee18] Kimin Lee et al. "A Simple Unified Framework for Detecting Out-of-Distribution Samples and Adversarial Attacks". In: *Advances in Neural Information Processing Systems*. Ed. by S Bengio et al. Vol. 31. Curran Associates, Inc., 2018, pp. 7167–7177. URL: https://proceedings.neurips.cc/paper/2018/file/abdeb6f575ac5c6676b747bca8d09cc2-Paper.pdf (cit. on p. 2).

[Lia18] Shiyu Liang, Yixuan Li, and R. Srikant. "Enhancing The Reliability of Out-of-distribution Image Detection in Neural Networks". In: *International Conference on Learning Representations*. 2018. URL: https://openreview.net/forum?id=H1VGkIxRZ (cit. on p. 2).

[Zop18] Barret Zoph et al. "Learning Transferable Architectures for Scalable Image Recognition". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2018 (cit. on p. 1).

[Ang19] Matt Angus, Krzysztof Czarnecki, and Rick Salay. "Efficacy of Pixel-Level OOD Detection for Semantic Segmentation". In: *CoRR* abs/1911.02897 (2019). arXiv: 1911.02897. URL: http://arxiv.org/abs/1911.02897 (cit. on p. 3).

[Bev19] Petra Bevandić et al. "Simultaneous Semantic Segmentation and Outlier Detection in Presence of Domain Shift". In: *Pattern Recognition*. Ed. by Gernot A. Fink, Simone Frintrop, and Xiaoyi Jiang. Cham: Springer International Publishing, 2019, pp. 33–47 (cit. on p. 3).

[Cha19] Robin Chan et al. "The ethical dilemma when (not) setting up cost-based decision rules in semantic segmentation". In: *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*. Vol. 2019-June. 2019, pp. 1395–1403. ISBN: 9781728125060. DOI: 10.1109/CVPRW.2019.00180. eprint: 1907.01342 (cit. on p. 3).

[Hei19] Matthias Hein, Maksym Andriushchenko, and Julian Bitterwolf. "Why ReLU Networks Yield High-Confidence Predictions Far Away From the Training Data and How to Mitigate the Problem". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2019 (cit. on p. 2).

[Hen19] Dan Hendrycks, Mantas Mazeika, and Thomas Dietterich. "Deep Anomaly Detection with Outlier Exposure". In: *Proceedings of the International Conference on Learning Representations* (2019) (cit. on p. 3).

[Joh19] Justin M. Johnson and Taghi M. Khoshgoftaar. "Survey on deep learning with class imbalance". In: *Journal of Big Data* 6.1 (Mar. 2019), p. 27. ISSN: 2196-1115. DOI: 10.1186/s40537-019-0192-5 (cit. on pp. 1, 2).

[Lis19] Krzysztof Lis et al. "Detecting the Unexpected via Image Resynthesis". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*. Oct. 2019 (cit. on p. 3).

[Muk19] Jishnu Mukhoti and Yarin Gal. *Evaluating Bayesian Deep Learning Methods for Semantic Segmentation*. 2019. arXiv: 1811.12709 [cs.CV] (cit. on p. 3).

[Pet19] Philipp Petersen and Felix Voigtlaender. "Equivalence of approximation by convolutional neural networks and fully-connected networks". In: *Proceedings of the American Mathematical Society*. 2019. DOI: https://doi.org/10.1090/proc/14789 (cit. on p. 29).

[Brü20] Dominik Brüggemann et al. "Detecting out of distribution objects in semantic segmentation of street scenes". In: *Proceedings of the 30th European Safety and Reliability Conference (ESREL) and the 15th Probabilistic Safety Assessment and Management Conference*. 2020, pp. 3023–3030. ISBN: 9789811485930. DOI: 10.3850/978-981-14-8593-0_4518-cd (cit. on p. 4).

[Cal20] Ovidiu Calin. *Deep Learning Architectures - A Mathematical Approach*. 1st ed. Springer International Publishing, 2020 (cit. on pp. 8, 9, 12, 20, 21, 26–28).

[Cha20a] Robin Chan et al. "Application of maximum likelihood decision rules for handling class imbalance in semantic segmentation". In: *Proceedings of the 30th European Safety and Reliability Conference (ESREL) and the 15th Probabilistic Safety Assessment and Management Conference*. 2020, pp. 3065–3072. ISBN: 9789811485930. DOI: 10.3850/978-981-14-8593-0_5748-cd. eprint: 1901.08394 (cit. on p. 3).

[Cha20b] Robin Chan et al. "Controlled False Negative Reduction of Minority Classes in Semantic Segmentation". In: *Proceedings of the International Joint Conference on Neural Networks (IJCNN)*. 2020. ISBN: 9781728169262. DOI: 10.1109/IJCNN48605.2020.9207104 (cit. on p. 4).

[Dei20] Marc Peter Deisenroth, A. Aldo Faisal, and Cheng Soon Ong. *Mathematics for Machine Learning*. Cambridge University Press, 2020 (cit. on pp. 9, 12, 22, 24, 25).

[Got20] Hanno Gottschalk. *Hochdimensionale Wahrscheinlichkeitstheorie und maschinelles Lernen (in German)*. University of Wuppertal, 2020 (cit. on pp. 18, 19).

[Goz20] Ophir Gozes et al. *Rapid AI Development Cycle for the Coronavirus (COVID-19) Pandemic: Initial Results for Automated Detection & Patient Monitoring using Deep Learning CT Image Analysis*. 2020. arXiv: 2003.05037 [eess.IV] (cit. on p. 32).

[Gus20] Fredrik K. Gustafsson, Martin Danelljan, and Thomas Bo Schön. "Evaluating Scalable Bayesian Deep Learning Methods for Robust Computer Vision". In: *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*. 2020, pp. 1289–1298 (cit. on p. 3).

[Hen20] Dan Hendrycks et al. *Scaling Out-of-Distribution Detection for Real-World Settings*. 2020. arXiv: 1911.11132 [cs.CV] (cit. on p. 3).

[Jou20] Nicolas Jourdan, Eike Rehder, and Uwe Franke. "Identification of Uncertainty in Artificial Neural Networks". In: *Proceedings of the 13th Uni-DAS e.V. Workshop Fahrerassistenz und automatisiertes Fahren*. July 2020 (cit. on p. 3).

[Mei20] Alexander Meinke and Matthias Hein. "Towards neural networks that provably know when they don't know". In: *International Conference on Learning Representations*. 2020. URL: https://openreview.net/forum?id=ByxGkySKwH (cit. on pp. 2, 3).

[Mis20] Diganta Misra. "Mish: A Self Regularized Non-Monotonic Activation Function". In: *31st British Machine Vision Conference 2020, BMVC 2020, Virtual Event, UK, September 7-10, 2020*. BMVA Press, 2020. URL: https://www.bmvc2020-conference.com/assets/papers/0928.pdf (cit. on p. 1).

[Rot20] Matthias Rottmann et al. “Prediction Error Meta Classification in Semantic Segmentation: Detection via Aggregated Dispersion Measures of Softmax Probabilities”. In: *Proceedings of the International Joint Conference on Neural Networks (IJCNN)*. 2020. ISBN: 9781728169262. DOI: 10.1109/IJCNN48605.2020.9206659. eprint: 1811.00648 (cit. on p. 4).

[Tao20] Andrew Tao, Karan Sapra, and Bryan Catanzaro. “Hierarchical Multi-Scale Attention for Semantic Segmentation”. In: *CoRR* abs/2005.10821 (2020). arXiv: 2005.10821 (cit. on p. 38).

[Yu20] Fisher Yu et al. “Bdd100k: A diverse driving dataset for heterogeneous multitask learning”. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2020, pp. 2636–2645 (cit. on pp. 1, 33).

[Yua20] Yuhui Yuan, Xilin Chen, and Jingdong Wang. “Object-Contextual Representations for Semantic Segmentation”. In: *Computer Vision - ECCV 2020 - 16th European Conference, Glasgow, UK, August 23-28, 2020, Proceedings, Part VI*. Ed. by Andrea Vedaldi et al. Vol. 12351. Lecture Notes in Computer Science. Springer, 2020, pp. 173–190. DOI: 10.1007/978-3-030-58539-6_11 (cit. on pp. 1, 39).

[Zho20] Ding-Xuan Zhou. “Universality of deep convolutional neural networks”. In: *Applied and Computational Harmonic Analysis* 48.2 (2020), pp. 787–794. ISSN: 1063-5203. DOI: https://doi.org/10.1016/j.acha.2019.06.004 (cit. on p. 29).

[Afs21] Parnian Afshar et al. “COVID-CT-MD, COVID-19 computed tomography scan dataset applicable in machine learning and deep learning”. In: *Scientific Data* 8.1 (Apr. 2021), p. 121. ISSN: 2052-4463. DOI: 10.1038/s41597-021-00900-3 (cit. on p. 33).

[Blu21] Hermann Blum et al. “The Fishyscapes Benchmark: Measuring Blind Spots in Semantic Segmentation”. In: *International Journal of Computer Vision* 129.11 (Nov. 2021), pp. 3119–3135. ISSN: 1573-1405. DOI: 10.1007/s11263-021-01511-6 (cit. on p. 3).

[Cha21a] Robin Chan, Matthias Rottmann, and Hanno Gottschalk. “Entropy Maximization and Meta Classification for Out-Of-Distribution Detection in Semantic Segmentation”. In: *The IEEE/CVF International Conference on Computer Vision (ICCV)*. 2021. eprint: 2012.06575 (cit. on p. 4).

[Cha21b] Robin Chan et al. “SegmentMeIfYouCan: A Benchmark for Anomaly Segmentation”. In: *Thirty-fifth Conference on Neural Information Processing Systems (NeurIPS) Datasets and Benchmarks Track*. 2021 (cit. on p. 4).

[Di 21] Giancarlo Di Biase et al. “Pixel-Wise Anomaly Detection in Complex Driving Scenes”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2021, pp. 16918–16927 (cit. on p. 3).

[Jun21] Sanghun Jung et al. “Standardized max logits: A simple yet effective approach for identifying unexpected road obstacles in urban-scene segmentation”. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2021, pp. 15425–15434 (cit. on p. 3).

[Liu21] Ze Liu et al. “Swin Transformer: Hierarchical Vision Transformer Using Shifted Windows”. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*. Oct. 2021, pp. 10012–10022 (cit. on p. 1).

[Par21] Sejun Park et al. “Minimum Width for Universal Approximation”. In: *International Conference on Learning Representations (ICLR)*. 2021. URL: https://openreview.net/forum?id=O-XJwyoIF-k (cit. on p. 15).

[Pha21] Hieu Pham et al. “Meta Pseudo Labels”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2021, pp. 11557–11568 (cit. on p. 1).

[Rah21] Sejuti Rahman et al. “Deep Learning–Driven Automated Detection of COVID-19 from Radiography Images: a Comparative Analysis”. In: *Cognitive Computation* (Mar. 2021). ISSN: 1866-9964. DOI: 10.1007/s12559-020-09779-5 (cit. on p. 33).

[Sar21] Arjun Sarkar et al. “Identification of Images of COVID-19 from Chest X-rays Using Deep Learning: Comparing COGNEX VisionPro Deep Learning 1.0™ Software with Open Source Convolutional Neural Networks”. In: *SN Computer Science* 2.3 (Mar. 2021), p. 130. ISSN: 2661-8907. DOI: 10.1007/s42979-021-00496-w (cit. on p. 33).

[Yar21] Dmitry Yarotsky. “Universal Approximations of Invariant Maps by Neural Networks”. In: *Constructive Approximation* (Apr. 2021). ISSN: 1432-0940. DOI: 10.1007/s00365-021-09546-1. URL: https://doi.org/10.1007/s00365-021-09546-1 (cit. on p. 29).

False Negative Detection and Reduction in Semantic Segmentation

In this chapter, we briefly describe our works dealing with objects that have been overlooked in the semantic segmentation of street scenes due to class imbalance. More precisely, we aim to detect and reduce pixel class predictions failing to indicate the presence of certain objects, which fall under the category of *false negative* errors. This is particularly safety relevant when these false negative instances belong to important classes like humans or traffic signs.

In this light, we consider neural networks as statistical models providing a probability distribution over predefined classes for each pixel in an image. Therefore, to obtain the final class predictions, *decision rules* must be incorporated, which are inevitably accompanied by weightings of classes. In the following, we discuss why decision rules reveal to be ethically difficult, besides not being necessarily optimal, and propose alternative approaches.

All the following outlined works can be found in detailed form in Chapter 6.

4.1 Cost-based Decision Rules

Most neural networks for classification are usually equipped with a softmax function in the final network layer, as explained in Definition 2.8. The probabilistic output of such a layer can be seen as posterior distribution expressing a model's confidence to label the input correctly. In deep learning the final prediction is then typically obtained by applying the argmax function as decision rule, see Equation (2.11). From decision theory, this kind of decision making is known as *maximum a-posteriori probability* (MAP) principle. However, the MAP principle is one particular type of cost-based decision rules and it is known to consider each type of error equally serious. The latter fact usually conflicts with human common sense. For example, the confusion of person and street should intuitively be assessed differently compared to the confusion of pole and building.

More formally, consider the class probabilities $p(k|\mathbf{x})$ for classes $k \in \{1, \dots, K\}$ conditioned on some input $\mathbf{x}$. Then, given some confusion cost function $c : \{1, \dots, K\} \times \{1, \dots, K\} \to \mathbb{R}_{\geq 0} := [0, \infty)$, an optimal cost-based decision rule minimizes the expected confusion costs, that is

$$D(\mathbf{x}, c) := \underset{k' \in \{1,\dots,K\}}{\arg\min} \; \mathbb{E}\big[\; c(Y' = k', Y) \mid \mathbf{x} \;\big] = \underset{k' \in \{1,\dots,K\}}{\arg\min} \; \sum_{k=1}^{K} c(k', k)\; p(k|\mathbf{x}) \;. \tag{4.1}$$

It is easy to prove that the decision rule according to the MAP principle is equivalent to an optimal cost-based decision rule incorporating the simple symmetric cost function

$$c_s(k', k) = \begin{cases} 0 & , \quad \text{if } k' = k \\ \lambda & , \quad \text{if } k' \neq k \end{cases} , \quad \lambda \in \mathbb{R}_{\geq 0} \;. \tag{4.2}$$

(a) Altruistic attitude **(b)** "Robotistic" attitude **(c)** Egoistic attitude

Figure 4.1: Different visual perception due to different decision rules. The obtained segmentation masks are obtained with the same semantic segmentation network but different (ad-hoc defined) confusion costs representing **(a)** an altruistic attitude, **(b)** a robotistic attitude (the standard maximum a-posteriori probability principle) and **(c)** an egoistic attitude.

We recall that semantic segmentation models essentially estimate class probabilities for each single pixel of an image. Using these estimates, the final class predictions are obtained by applying the MAP principle pixel-wise, *cf*. Equation (2.11). Thus, we realize that in most semantic segmentation models, but generally also in machine learning models, all types of confusions between different classes have an equal weighting with a constant λ as the MAP principle is usually employed by default. While it seems reasonable that confusion costs should be different from this "robotistic" cost assignment, *i.e.* different from being constant, ethically it is very unclear which numbers should explicitly be used.

In this work, we investigate how different confusion cost assessments change the final predictions in semantic segmentation and therefore also the obtained visual perception. A numerical study demonstrates the practical relevance of confusion cost functions as they might represent different ethical attitudes, see *e.g.* Figure 4.1. In particular, we show that employing different confusion costs might yield significant changes of safety-relevant quantities, such as precision and recall, of a state-of-the-art semantic segmentation model trained on Cityscapes. Our main aim of this work is not to provide a solution to this problem but to make the ethical dimension transparent that is involved in the necessary choice of decision rules.

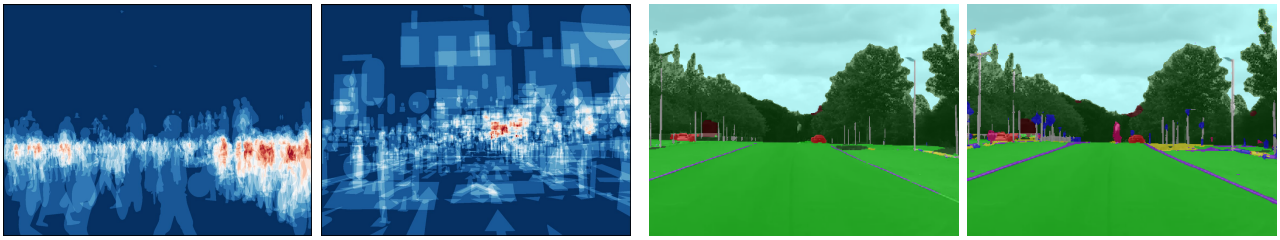
The corresponding full article appears in Section 6.1.

4.2 Maximum Likelihood Decision Rules

In decision theory, the MAP principle is merely one example of cost-based decision rules, which is also known as *Bayes* decision rule. This decision rule implies an equal class weighting, *i.e.* weighting every type of misclassification equally, and therefore might ignore classes that are rare in terms of their frequency of occurrence in the training dataset. In fact, class imbalance is often inherent in a real-world dataset itself. For instance, when considering a street scene dataset, pixels belonging to safety relevant classes, such as humans or traffic signs, are underrepresented compared to the class road, since the latter appears in every image besides typically being substantially larger in terms of object size. Consequently, neural networks might be trained on unbalanced data which possibly results in an undesirable bias to the disadvantage of ignoring classes of highest interest.

A mathematically natural approach from decision theory to tackle this issue of class imbalance is employing the *maximum likelihood* (ML) decision rule, which is defined for an input $\mathbf{x}$ as

$$D^{\mathrm{ML}}(\mathbf{x}) := \underset{k\in\{1,\dots,K\}}{\arg\max}\ p(\mathbf{x}|k) \overset{\text{Bayes Theorem}}{=} \underset{k\in\{1,\dots,K\}}{\arg\max}\ \frac{p(k|\mathbf{x})\,p(\mathbf{x})}{p(k)} = \underset{k\in\{1,\dots,K\}}{\arg\max}\ \frac{p(k|\mathbf{x})}{p(k)}\ . \tag{4.3}$$



(a) Pixel-wise priors in the DS20K dataset **(b)** Segmentation maks with different decision rules

Figure 4.2: **(a)** Pixel-wise class distributions for humans (left) and traffic signs (right) in the DS20K dataset as heatmaps. **(b)** Semantic segmentation with the MAP / Bayes decision rule (left) and with the position-specific maximum likelihood decision rule (right). Note that the person in the scene is completely overlooked by Bayes, but fully detected by Maximum Likelihood.

Seen from the angle of cost-based decision rules, the ML rule assigns costs inverse proportional to the class priors, *i.e.*

$$c_p(k', k) = \begin{cases} 0 & , \quad \text{if } k' = k \\ \frac{1}{p(k)} & , \quad \text{if } k' \neq k \end{cases} \tag{4.4}$$

where $p(k)$ is called *a-priori probability* or simply *prior* of class $k \in \{1, \ldots, K\}$.

We notice that the ML rule is the same as the MAP principle, *cf*. Equation (2.11), but with a weight adjustment $p(k)$. This adjustment puts higher weight on classes with a low a-priori probability. Since these priors can be estimated via class frequencies in the training dataset, the ML decision rule therefore puts more emphasis on finding classes that rarely appear than the Bayes decision rule. In other words, the ML rule balances class probabilities and thus aims at finding the class k for which the input $\mathbf{x}$ is most typical without any prior belief about the underlying class frequency.

In this work, we approach the reduction of false negatives in semantic segmentation by applying different variants of the maximum likelihood decision rule. In more detail, we employ a global ML rule and position-specific one, which differ in the estimation of priors. We provide an in-depth comparison between these two decision rules with respect to the vulnerable class human in the street scene datasets Cityscapes and DS20k. The obtained results show that ML rules can significantly improve recall, however only with some sacrifice in precision. Nevertheless, ML decision rules as post-processing step are computationally nearly for free. They can easily be applied simultaneously to the standard MAP principle / Bayes decision rule and offer great potential when combined with methods checking the plausibility of ML predictions.

The corresponding full article appears in Section 6.2.

4.3 Prediction Error Meta Classification

While deep learning continuously improves the performance of safety critical tasks like automated driving or medical diagnosis, one crucial aspect that remains is providing reliable predictions. Neural networks as statistical models produce errors with a certain probability. When incorrect predictions cannot be avoided, it is desirable to at least know when they occur. In the special case of semantic segmentation, it is further helpful to locate errors within image dimensions. To this end, we consider the task of *meta classification*, which can be described as discriminating between a true positive prediction and a false positive prediction without requiring access to the ground truth, see Figure 4.3, and in that way performing a prediction quality rating.

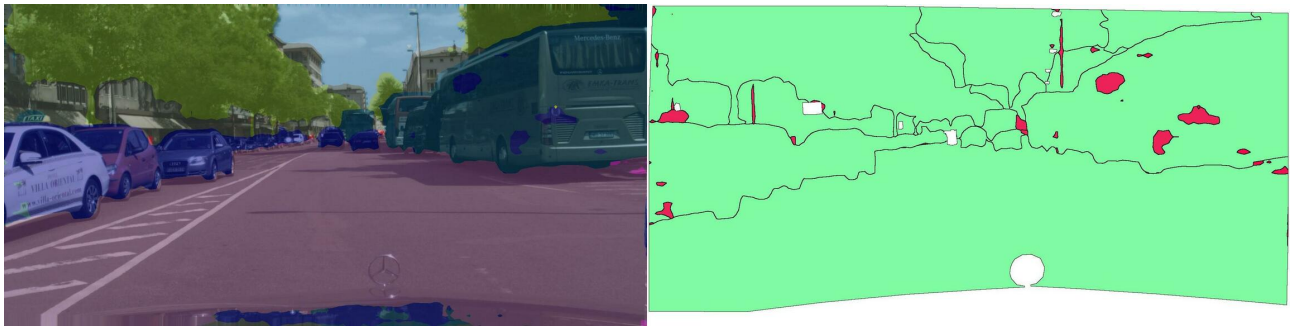


Figure 4.3: An example of meta classification in semantic segmentation. After an image is processed by a semantic segmentation network (left), the meta classifier applied afterwards is able to identify incorrectly predicted segments (right, errors marked with red). Note that meta classification is performed without having access to the ground truth.

Meta classification is mainly based on prediction uncertainty. In semantic segmentation, uncertainty is typically quantified at pixel-level. Intuitive measures are for instance the probability margin

$$M(\mathbf{x}) = 1 - \max_{k \in \{1,\dots,K\}} p(k|\mathbf{x}) \tag{4.5}$$

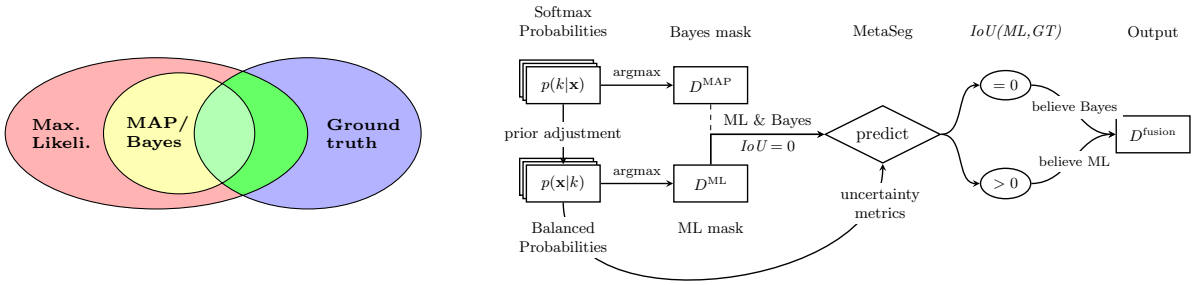
or the Shannon entropy

$$E(\mathbf{x}) = -\sum_{k=1}^{K} p(k|\mathbf{x}) \ln(p(k|\mathbf{x})) \ . \tag{4.6}$$

Note again that in practice the pixel-wise class probability $p(k|\mathbf{x})$ is estimated by the output of a neural network for semantic segmentation. We observe that these dispersion measures applied on pixel-level are not sufficient for reliable pixel-wise meta classification. However, aggregating these metrics over predicted segments, *i.e.* connected components of pixels in semantic segmentation masks sharing the same class labels, yields improved correlation with prediction errors for entire image regions. Besides, a collection of aggregated metrics form a structured dataset that can be efficiently processed with simple regression models such as logistic regression. This allows us to additionally quantify how well a segment is predicted and thus also analyze the impact of each single metric on the identification of classification mistakes.

In this work, we present a procedure for meta classification specifically designed for semantic segmentation. We perform segment-wise meta classification by aggregating pixel-wise metrics, which all can be derived from the softmax probabilities obtained by neural networks. The hand-crafted metrics include pixel-wise dispersion measures as well as geometry information of predicted segments. Those metrics are then fed into a logistic regression model and show to reliably quantify the prediction quality of predicted segments. In order to show the generalization capabilities of our approach, we perform tests on Cityscapes data as well as on BRATS2017 data, which cover two extremely safety-critical use cases of semantic segmentation. Moreover, we investigate the predictive power of different metrics and also different sets of metrics. The proposed post-processing method, which we term *MetaSeg*, outperforms standard approaches and yield a plug-and-play tool that can easily be extended with other pixel-wise uncertainty maps.

The corresponding full article appears in Section 6.3.



(a) Decision rules assumption

(b) Overview of MetaFusion

Figure 4.4: An overview of our *MetaFusion* method. **(a)** With respect to a minority class in semantic segmentation, segment class predictions by the maximum a-psoertiori probability (MAP) / Bayes decision rule lie entirely within segments assigned to the same class by the maximum likelihood (ML) decision rule (both rules applied to the same input). **(b)** Based on the latter assumption, MetaFusion selects predicted segments that are further processed via MetaSeg, yielding a fused decision rule to reduce false negative errors.

4.4 Controlled False-Negative Reduction of Minority Classes

In our studies on decision rules we observed that the recall with respect to underrepresented classes in semantic segmentation can significantly be improved by applying the position-specific maximum likelihood decision rule. However, this gain in recall is accompanied by an overproduction of false positive indications and therefore leading to a substantial performance loss in terms of precision. One intuitive solution to this problem would be to post-process ML predictions with MetaSeg, since the latter is specifically tailored to false positive detection. As the original MetaSeg method does not provide alternative predictions for identified false positives, we propose the following two-step procedure.

First, we apply the maximum likelihood as well as the standard Bayes decision rule to the same semantic segmentation network, which will then produce two different semantic segmentation masks. They differ because ML performs a prior adjustment assigning higher weights to minority classes than without that adjustment. Hence, we conclude that with respect to the most underrepresented class, every predicted Bayes segment lies entirely inside predicted ML segments, see Figure 4.4 (a). Based on this finding for such a minority class, we assume that a non-empty intersection between ML and Bayes segments, both predicted to belong to the same minority class, indicates a confirmation for the presence of a minority class object that Bayes already has detected. On the contrary, ML segments, that do not intersect with any Bayes segment, then indicate image regions where minority class objects might be overlooked by Bayes.

Based on the observation whether the *decision rules agree or disagree*, segments are selected that are further processed by a modified version of MetaSeg. We employ a meta classifier whose metrics are derived from the balanced softmax probabilities given by a neural network for semantic segmentation. Whenever the meta classier now identifies a false positive prediction, we stick to the standard MAP principle. Otherwise, we replace the Bayes segment by the ML segment in the Bayes semantic segmentation mask. Thus, the combination of segment selection based on multiple decision rules and meta classification yield a novel decision rule, which we term *MetaFusion*, see also Figure 4.4 (b). In this context, the ultimate goal is to discard all additional false positives while keeping all additional true positives, which are generated due the application of the ML decision rule.

In this work, we focus on pedestrians as minority class of highest interest in the semantic segmentation of street scenes. We perform our experiments on the Cityscapes dataset for which the class human is significantly underrepresented. Our proposed procedure combines decision rules for false negative detection with methods for false positive detection based on aggregated uncertainty metrics. The obtained results demonstrate that our final fused decision rule achieves more favorable trade-offs between error rates, in particular compared to other methods for false negative reduction. In other words, we reduce the number of false negatives while controlling the (over-)production of false positives. As pure post-processing method, MetaFusion can easily be added on top of any semantic segmentation neural network. In addition, our method is designed to have online capabilities, having a computational overhead that is negligible relative to the complexity of inferences of semantic segmentation models.

The corresponding full article appears in Section 6.4.

Out-of-Distribution Detection in Semantic Segmentation

In this chapter, we briefly describe our works dealing with objects that have been overlooked in the semantic segmentation of street scenes since they belong to an unknown object category. As neural networks are usually trained to always predict one class from a closed and predefined set of object classes, these models are prevented from predicting anything outside of this semantic space. However, the detection of such semantically unknown objects in street scenes is extremely safety relevant, particularly in automated driving when they appear on the road ahead of the self-driving car that relies on semantic segmentation for perception.

In this regard, unknown or unusual objects shown in images are commonly considered as *anomalies*, which is a generic term for anything deviating from what is regarded as normal. The corresponding task of detecting and localizing anomalies at pixel-level is referred to as *anomaly segmentation*. To be more precise, we are interested in finding objects that are semantically novel and appear after training a model. As these objects lie outside of the learnable distribution provided by the training data, such anomalies are more specifically referred to as *out-of-distribution*[1] (OoD) objects.

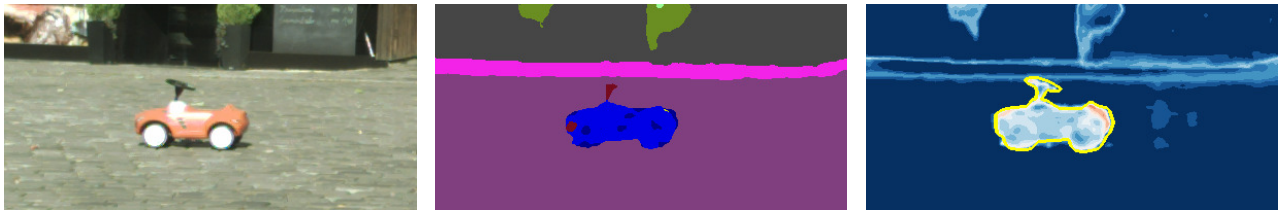
In this chapter, we discuss how OoD objects can be detected and localized at pixel-level using semantic segmentation models. To this end, we propose an anomaly segmentation approach based on prediction uncertainty, *i.e.* the architecture of the underlying semantic segmentation model remains untouched. Via a specific training regime, we considerably increase the sensitivity towards the identification of semantically unknown objects without significantly sacrificing in original semantic segmentation performance. Additionally, we introduce a benchmark to anomaly segmentation, which is an important yet not widely recognized task at the time of writing.

All the following outlined works can be found in detailed form in Chapter 6.

5.1 Softmax Entropy Thresholding

When feeding images with OoD objects to a convolutional neural network (CNN) for semantic segmentation, first experiments revealed that the model has low confidence scores with respect to multiple classes on the OoD objects. Thus, the model can be interpreted as being uncertain about its prediction, which is a desirable property when exposed to anomalies. For that matter, one intuitive measure to quantify uncertainty from softmax probabilities is given by the entropy, see Equation (4.6). In semantic segmentation, computing the softmax entropy yields a pixel-wise anomaly score map. Ide-

[1] In deep learning terminology, the terms "anomaly" and "out-of-distribution", respectively, are often used interchangeably for semantically unknown objects. In this chapter, we stick to the more general term anomaly and only specify when the distinction is relevant.



(a) Original input **(b)** Prediction mask **(c)** Softmax entropy

Figure 5.1: An example of the softmax entropy for the detection of a bobby car, which is a semantically unknown object for semantic segmentation models trained on Cityscapes. **(a)** An input image containing a bobby car. **(b)** The corresponding semantic segmentation mask, in which the bobby car is identified as car. **(c)** The softmax entropy, however, indicates an increased prediction uncertainty in the region of the bobby car (brighter pixels in the corresponding heatmap).

ally, a high value indicates the presence of a pixel belonging to an anomaly. Then, by choosing an appropriate threshold on the score map, we perform anomaly segmentation without any modification of the existing semantic segmentation model.

In this work, we extensively study the capabilities of softmax entropy thresholding with regard to anomaly segmentation. We employ a state-of-the-art CNN for semantic segmentation, which is trained on Cityscapes, for the detection of OoD objects. We evaluate anomaly segmentation performance on the LostAndFound dataset, which shares the same setup as Cityscapes but contains small and semantically unknown obstacles placed on the road. We observe that most OoD objects are attached with high pixel-wise entropy, however, at the same time there is a substantial amount of pixels with high entropy in image regions where no anomalies are present. To account for small and false OoD object predictions, we additionally apply morphological image operations to connect segments within a specified distance, which alleviates the issue of false positive OoD indications only slightly. Hence, the limitations of simply thresholding on the softmax entropy still persist but provide first valuable insights to further tackle anomaly segmentation.

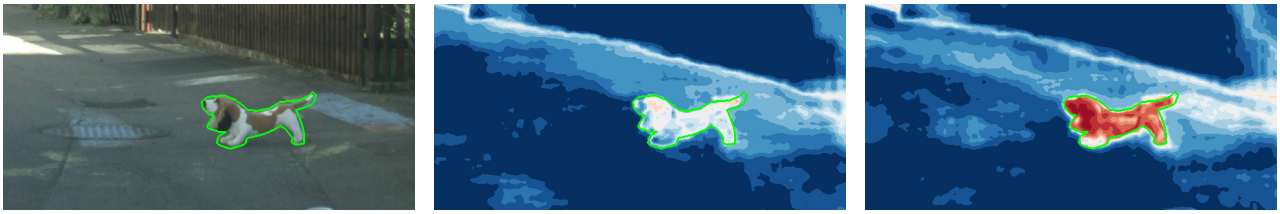
The corresponding full article appears in Section 6.5.

5.2 Entropy Maximization and Meta Classification

After investigating softmax entropy thresholding for the detection of OoD objects, we concluded that OoD pixels are indeed assigned high entropy scores, but to the detriment of many faulty anomaly indications at the same time. Obviously, a better separation between out-distribution and in-distribution pixels is necessary. To this end, we approach to improve separation by deliberately inducing OoD objects as *known unknowns* into a retraining process of a pretrained semantic segmentation model. On these known unknowns, we enforce the semantic segmentation model to maximize the softmax entropy by means of a multi objective loss function. The general aim of this retraining approach is that the semantic segmentation model generalizes newly learned uncertainty concepts to truly unseen OoD examples, which are referred to as *unknown unknowns*, without sacrificing in original performance on the primary task of semantic segmentation.

In what follows, let us omit the consideration of image pixels[2] for the sake of simplicity and assume

[2]This is unproblematic as the softmax output of semantic segmentation models yield pixel-wise probability distributions over all classes for each single pixel of a given input image. Then, a loss function is applied individually to each single pixel-wise probability distribution. To compute the loss per image, the pixel-wise scalar losses are simply averaged over pixel locations of the input. Thus, each entry of the softmax output is treated individually.



(a) Input with OoD instance **(b)** Before OoD training **(c)** After OoD training

Figure 5.2: An example of OoD training with entropy maximization for the detection of unknown objects. In this example the unknown object is represented by a dog (marked with green contours) which is unknown to a semantic segmentation network trained on Cityscapes. **(a)** An image containing a dog. **(b)** The pixel-wise softmax entropy as heatmap before OoD training is applied. **(c)** The pixel-wise softmax entropy as heatmap after OoD training is applied. The dog as unknown object becomes visible clearly visible.

that $f(\mathbf{x}) \in (0,1)^K$ denotes the softmax output over K classes for one arbitrary pixel of an input $\mathbf{x}$. We realize that maximizing the softmax entropy on anomaly data is accomplished by incorporating the training objective

$$\mathcal{L}_{out}(f(\mathbf{x})) := -\sum_{k=1}^{K} \frac{1}{K} \ln(f_k(\mathbf{x})), \quad f(\mathbf{x}) \in (0,1)^K \, , K \in \mathbb{N} \, , \tag{5.1}$$

which represents the negative log-likelihood averaged over all classes $\{1, \ldots, K\}$. Minimizing Equation (5.1) is equivalent to maximizing the softmax entropy

$$E(f(\mathbf{x})) = -\sum_{k=1}^{K} f_k(\mathbf{x}) \ln(f_k(\mathbf{x})) \, , \quad f(\mathbf{x}) \in (0,1)^K \, , \tag{5.2}$$

cf. also Equation (4.6). The latter statement is straightforward to prove with Jensen's inequality.

Since the softmax definition implies $f_k(\mathbf{x}) \in (0,1) \; \forall \; k \in \{1, \ldots, K\}$ and $\|f(\mathbf{x})\|_1 = 1$, Jensen's inequality applied to the convex function $-\ln(\cdot)$ yields

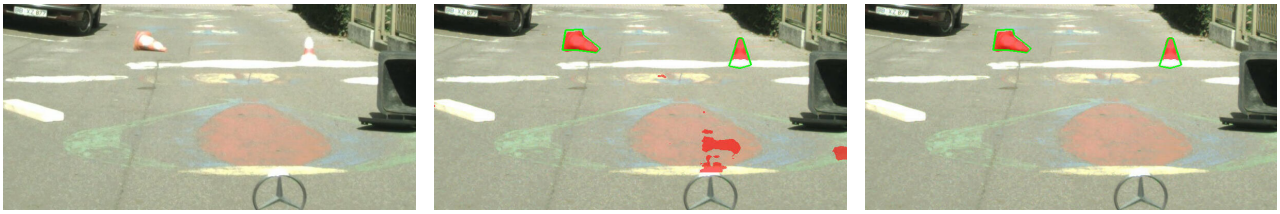
$$\mathcal{L}_{out}(f(\mathbf{x})) = \frac{1}{K} \sum_{k=1}^{K} -\ln\left(f_k(\mathbf{x})\right) \geq -\ln\left(\frac{1}{K} \sum_{k=1}^{K} f_k(\mathbf{x})\right) = \ln(K) \tag{5.3}$$

and applied to the concave function $\ln(\cdot)$

$$E(f(\mathbf{x})) = \sum_{k=1}^{K} f_k(\mathbf{x}) \ln\left(\frac{1}{f_k(\mathbf{x})}\right) \leq \ln\left(\sum_{k=1}^{K} f_k(\mathbf{x}) \frac{1}{f_k(\mathbf{x})}\right) = \ln(K) \tag{5.4}$$

with equality if and only if $f_k(\mathbf{x}) = \frac{1}{K} \; \forall \; k \in \{1, \ldots, K\}$. Then, the overall training objective is given by a multi criteria loss function, where *e.g.* the standard cross-entropy loss function, *cf.* Equation (2.60), is applied to in-distribution pixels and Equation (5.1) to out-distribution pixels.

With entropy maximization the entropy measure is trained to be more sensitive to anomalies, ideally without producing any false indications. Motivated by the findings of the MetaFusion approach, *cf.* Section 4.4, we additionally post-process the anomaly segmentation via meta classification to prevent an overproduction of false positive anomaly predictions.



(a) Before OoD training **(b)** OoD training only **(c)** OoD training + meta classifier

Figure 5.3: An example of OoD training with entropy maximization combined with meta classification. In this example the semantically unknown objects are represented by pylons (marked with green contours), which are truly unseen OoD objects to our anomaly segmentation model. **(a)** An image with pylons that are entirely non-detected before OoD training. **(b)** After OoD training, the pylons are well segmented (red color) but at the same some false positives are produced on the street. **(c)** The false positive OoD predictions on the street are removed by applying meta classification while the true positive OoD predictions with respect to the pylons still remain.

In this work, we utilize samples randomly drawn from the COCO dataset in order to perform entropy maximization on CNNs for semantic segmentation. Starting from models pretrained on Cityscapes, we demonstrate that retraining for only a few epochs with entropy maximization yields a significant gain in anomaly segmentation performance over plain softmax entropy thresholding. To this end, we use the LostAndFound and Fishyscapes dataset, which both share the same setup as Cityscapes but contain real as well as synthetic OoD objects in their images, to evaluate anomaly segmentation. Furthermore, we observe that our retraining approach also generalizes to the detection of truly unseen OoD objects, outperforming many other state-of-the-art anomaly segmentation methods. As post-processing step to discard false positive OoD instance predictions, we apply lightweight and transparent linear models as meta classifiers. These meta classifiers operate based on hand-crafted metrics derived from softmax probabilities and additionally enhance anomaly segmentation quality. The combination of these two described steps show to consistently improve anomaly segmentation performance with only marginally sacrificing in original performance on the primary task of semantic segmentation on Cityscapes. Therefore, our presented approach is an efficient and yet lightweight procedure contributing to safer perception of street scenes via semantic segmentation.

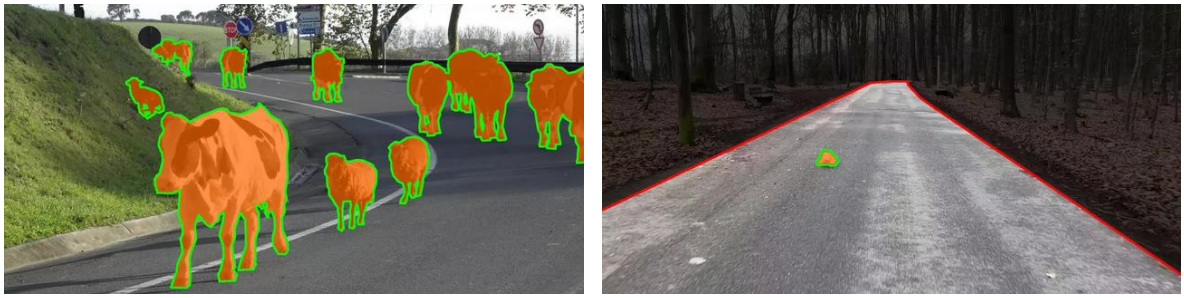
The corresponding full article appears in Section 6.6.

5.3 Benchmark for the Segmentation of Out-of-Distribution Objects

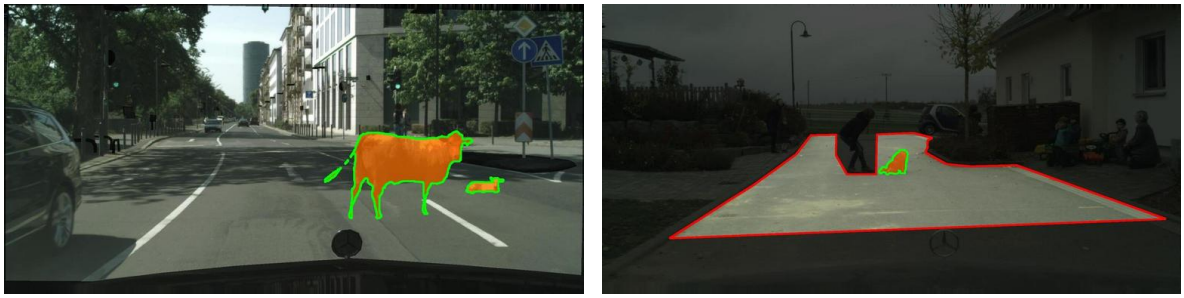
While there exists some methods tackling the task of detecting and localizing out-of-distribution objects at pixel-level, *i.e.* the task of *anomaly segmentation*, progress in this field remains slow despite its importance, which is mainly due to the lack of proper datasets and corresponding public benchmarks.

A prominent dataset for anomaly segmentation is LostAndFound, which contains images of street scenes in the same setup as Cityscapes but with real-world anomalous objects. This latter dataset is therefore particularly suitable to evaluate CNNs that are trained on the popular Cityscapes semantic segmentation dataset. However, LostAndFound suffers from low diversity of anomaly types, besides inconsistent ground truth labels. For instance, in this dataset children are considered as anomalies but other humans not. The missing benchmark suite additionally hinders proper evaluation and reliable comparison of proposed methods.

The best-known effort to benchmark anomaly segmentation methods is made in Fishyscapes. The accompanied dataset includes Cityscapes scenes with synthetic anomalous objects pasted into the



(a) Ours: RoadAnomaly21 **(b)** Ours: RoadObstacle21



(c) Fishyscapes **(d)** LostAndFound

Figure 5.4: Comparison of example images and their corresponding ground truth annotations from different anomaly segmentation datasets. In all displayed images, the anomalies of interest are highlighted in orange and additionally with green contours, whereas darkened regions are excluded from evaluation. Note that in Fishyscapes **(c)** anomalous objects are synthetic and in LostAndFound **(d)** anomalies are not labeled consistently (children vs. other humans).

images. Although the latter benchmark enables reliable comparison of methods, anomalies in this dataset look unrealistic due to the mixture of real and synthetic data. Therefore, such a dataset is not suitable to represent OoD objects that arise in the real world.

The mentioned works shed light on the shortage of real-world datasets with real and diverse anomalies in order to properly benchmark anomaly segmentation. Other difficulties that arise in the creation of a benchmark is the proper labeling of anomalies, since differences between known and unknown are oftentimes blurry. Moreover, there are no established and practically relevant performance metrics available to measure the quality of anomaly segmentation, not to mention a unified evaluation framework.

In this work, we introduce the *SegmentMeIfYouCan* benchmark. Our public benchmark encompasses two separated tasks of anomaly segmentation in the context of street scenes: firstly, strict OoD object segmentation, where any object category not included in Cityscapes is considered; and secondly, obstacle segmentation, whose goal is to segment any obstacle on the road, irrespective of whether the corresponding object category may be known. Alongside a unified and publicly-available test suite with classical pixel-wise as well as recent component-wise performance metrics, we provide two datasets addressing the two benchmark tracks. Both datasets consist of high-resolution real images with pixel-level annotations. By focusing on a wide variety in anomaly types and sizes, our datasets contribute to greater diversity in data than any other existing anomaly segmentation dataset. Furthermore, we empirically evaluate multiple state-of-the-art anomaly segmentation methods, forming an initial public leader board, which is available on https://www.segmentmeifyoucan.com.

The corresponding full article appears in Section 6.7.

Publications

List of all publications included in this dissertation:

[1] Robin Chan, Matthias Rottmann, Radin Dardashti, Fabian Hüger, Peter Schlicht and Hanno Gottschalk, *"The Ethical Dilemma when (not) Setting up Cost-based Decision Rules in Semantic Segmentation"* in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops, Safe Artificial Intelligence for Automated Driving (SAIAD)* © 2019

[2] Robin Chan, Matthias Rottmann, Fabian Hüger, Peter Schlicht and Hanno Gottschalk, *"Application of Decision Rules for Handling Class Imbalance in Semantic Segmentation"* in *The 30th European Safety and Reliability Conference (ESREL)* © 2020

[3] Matthias Rottmann, Pascal Colling, Thomas Paul Hack, Robin Chan, Fabian Hüger, Peter Schlicht and Hanno Gottschalk, *"Prediction Error Meta Classification in Semantic Segmentation: Detection via Aggregated Dispersion Measures of Softmax Probabilities"* in *The IEEE International Joint Conference on Neural Networks (IJCNN)* © 2020

[4] Robin Chan, Matthias Rottmann, Fabian Hüger, Peter Schlicht and Hanno Gottschalk, *"Controlled False Negative Reduction of Minority Classes in Semantic Segmentation"* in *The IEEE International Joint Conference on Neural Networks (IJCNN)* © 2020

[5] Dominik Brüggemann, Robin Chan, Matthias Rottmann, Hanno Gottschalk and Stefan Bracke, *"Detecting Out-of-Distribution Objects in Semantic Segmentation of Street Scenes"* in *The 30th European Safety and Reliability Conference (ESREL)* © 2020

[6] Robin Chan, Matthias Rottmann and Hanno Gottschalk, *"Entropy Maximization and Meta Classification for Out-of-Distribution Detection in Semantic Segmentation"* in *The IEEE/CVF International Conference on Computer Vision (ICCV)* © 2021

[7] Robin Chan, Krzysztof Lis, Svenja Uhlemeyer, Hermann Blum, Sina Honari, Roland Siegwart, Pascal Fua, Mathieu Salzmann and Matthias Rottmann, *"SegmentMeIfYouCan: A Benchmark for Anomaly Segmentation"*, *Thirty-fifth Conference on Neural Information Processing Systems (NeurIPS) Datasets and Benchmarks Track* © 2021

Own contributions to:

[1] The Ethical Dilemma when (not) Setting up Cost-based Decision Rules in Semantic Segmentation:

- major contributions to the idea of the article's content
- all experiments were implemented and conducted by myself
- sections II,III,IV and V were mostly written by myself
- general editing of sections I and VI

---

[2] Application of Maximum Likelihood Decision Rules for Handling Class Imbalance in Semantic Segmentation:

- major contributions to the idea of the article's content
- all experiments were implemented and conducted by myself
- the whole article was mostly written by myself

---

[3] Prediction Error Meta Classification in Semantic Segmentation: Detection via Aggregated Dispersion Measures of Softmax Probabilities:

- the experiments in section V were implemented and conducted by myself
- section V was mostly written by myself
- improvement of the article's quality by revision and restructuring
- clearer explanation of the adjusted IoU

---

[4] Controlled False Negative Reduction of Minority Classes in Semantic Segmentation:

- major contributions to the methodological ideas
- all experiments were implemented and conducted by myself
- the whole article was mostly written by myself

[5] Detecting Out-of-Distribution Objects in Semantic Segmentation of Street Scenes:

- major contributions to the ideas of the master's thesis that resulted in this article
- guidance on development and experiments during the main author's time as master student
- section I was entirely written by myself
- general editing of sections II,III

---

[6] Entropy Maximization and Meta Classification for Out-of-Distribution Detection in Semantic Segmentation:

- major contributions to the methodological ideas
- all experiments were implemented and conducted by myself
- the whole article was mostly written by myself
- the whole the appendix was entirely wirtten by myself (not included in this dissertation but available as open access version on `https://openaccess.thecvf.com/content/ICCV2021/`)

---

[7] SegmentMeIfYouCan: A Benchmark for Anomaly Segmentation:

- major contribution to the idea of the article's content
- implementation of several baseline methods
- integration of component-wise metrics as well as visualization functions into benchmark suite
- sections II.ii, III.ii, III.iii, IV, V were mostly written by myself
- general editing of sections I, II.i, III.i
- major contribution to experimental results
- major contribution to collecting and annotating images in order to create test datasets
- major contribution in setting up the benchmark webpage `https://segmentmeifyoucan.com/`
- section C, D, E, F of the appendix were mostly written by myself (not included in this dissertation but available as open access version on `https://datasets-benchmarks-proceedings.neurips.cc/paper/2021`)

The Ethical Dilemma when (not) Setting up Cost-based Decision Rules in Semantic Segmentation

Robin Chan[1], Matthias Rottmann[1], Radin Dardashti[2],
Fabian Hüger[3], Peter Schlicht[3] and Hanno Gottschalk[1]

[1]School of Mathematics and Natural Sciences, University of Wuppertal
[2]Philosophy of Science, Philosophical Seminar, University of Wuppertal
[3]Architecture and AI Technologies, Automated Driving, Volkswagen Group Research

Abstract. Neural networks for semantic segmentation can be seen as statistical models that provide for each pixel of one image a probability distribution on predefined classes. The predicted class is then usually obtained by the maximum a-posteriori probability (MAP) which is known as Bayes rule in decision theory. From decision theory we also know that the Bayes rule is optimal regarding the simple symmetric cost function. Therefore, it weights each type of confusion between two different classes equally, e.g., given images of urban street scenes there is no distinction in the cost function if the network confuses a person with a street or a building with a tree. Intuitively, there might be confusions of classes that are more important to avoid than others. In this work, we want to raise awareness of the possibility of explicitly defining confusion costs and the associated ethical difficulties if it comes down to providing numbers. We define two cost functions from different extreme perspectives, an egoistic and an altruistic one, and show how safety relevant quantities like precision / recall and (segment-wise) false positive / negative rate change when interpolating between MAP, egoistic and altruistic decision rules.

I Introduction

Machines acting autonomously in spaces co-populated by humans and robots are no longer a futuristic vision, but are part of the agenda of the world's technologically most advanced corporations. Autonomous car driving has seen spectacular advances due to recent progress in artificial intelligence (AI) and therefore is one of the corner-cases for this development. As street traffic, according to the world health organization (WHO), causes an annual death toll of 1.35M persons at the time of writing [WHO18], it is expected that also autonomous driving cars will be involved in such tragic events. While there are reasons to believe that autonomous driving can reduce the overall numbers of deaths and heavy injuries, besides being required by *e.g.* the Ethics Commission instated by the German Federal Ministry of Transport and Digital Infrastructure [Eth17], many further ethical issues remain in the choices of programming an autonomous vehicle. Therefore, autonomous cars have been a much-discussed topic in robot ethics [Lin17], ranging from inevitable ethical dilemmas like the trolley problem [Foo67; Lin16] to more mundane ethical situations [Him18].

In most of these ethical situations discussed in the literature, the robots and the AI algorithms controlling them are assumed to know the situation they decide on, whereas most deadly accidents with the involvement of self-driving cars in some way or another are connected with the (insufficient) perception of the vehicle's surrounding (see [Boa18] for a preliminary report). Whether the AI algorithms of perception themselves depend on choices that involve ethical decisions is therefore a legitimate question.

For a practitioner in the field it is quite obvious that the answer is "yes": In semantic segmentation, the choice of training data, the selection of classes, potential class imbalance, the amount of data, the capacity of the learning algorithm and the performance of the hardware all determine what a

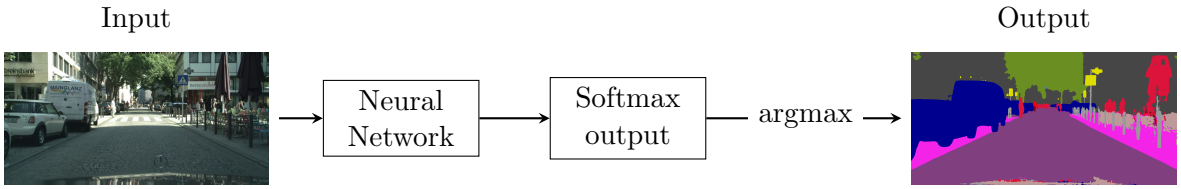


Figure 1: Illustration of semantic segmentation performed on an image of the Cityscapes dataset with a neural network in combination with (pixel-wise) maximum a-posteriori probability classification.

contemporary AI algorithm is able to "see" and how error prone its perception will be. As errors in perception are potential root causes of accidents, ethical implications clearly exist.

In this work, we draw the attention to one further issue that is connected to the probabilistic output of semantic segmentation neural networks that are mostly used for the perceptive task. As the softmax output of a segmentation network gives a pixel-wise class distribution, the maximum a-posteriori probability (MAP) principle, also known as Bayes decision rule, selects the class of highest probability. This is however not the only selection principle, as one could also apply the Maximum Likelihood (ML) decision rule that picks the class for which the input data is most representative [Fah96]. While both rules have the appeal of being mathematically "natural", they are merely two examples of cost-based decision rules, where each confusion event is penalized by a specific quantity $c(\hat{k}, k)$ that valuates the aversion of a decision maker towards the confusion of the predicted class $\hat{k}$ with the actual class k. The decision on the predicted class now minimizes the expected cost.

Seen from this angle, the MAP principle corresponds to the cost matrix that attributes equal cost to any confusion event. We call this the robotistic valuation of the segmentation network's output. Human common sense would valuate the confusion of the street with a pedestrian differently from the confusion event with the roles interchanged: an unjustified emergency brake is a much weaker consequence as potential harm than overlooking a person on the street and therefore should come with a significantly lower cost. While it seems reasonable to assume that the confusion cost should be different from constant, it is ethically much less evident, which numbers should explicitly be used. In these situations of moral uncertainty, different ethical schools of thought may provide different answers, with some refusing to weigh lives at all [Bro04]. In addition, legislation can put strong constraints on the choice as well. However, as the MAP principle and the ML decision rule already define confusion cost matrices, choices about these numbers have already been made. We, therefore, aim to make more transparent the ethical dimension involved in making a choice regarding a decision rule with its corresponding cost matrix.

We realize that the ultimate step from probabilities to perception depends on cost matrices in a high dimensional value space $\mathcal{V}$ and that the selected valuation changes the perception. Thereby, it also changes the consequences, as, *e.g.*, the precision and recall rates of specific classes. Furthermore, different cost matrices $C \in \mathcal{V}$ might express different ethical attitudes, like more egoistic (centred on the passenger in the (ego-) car) or altruistic (centred on public safety). Putting drivers first vs. putting the public first has already been subject to intense public debate [Tay16].

In this paper, we do not intend to resolve the problem outlined above in any way. We present a numerical study that demonstrates the practical relevance of the problem by traveling through the value space within a triangle of robotistic and approximately egoistic and altruistic, respectively, cost value systems. Here the egoistic and altruistic cost matrices are set up in an *ad hoc* manner and are not meant to accurately represent these attitudes. Also, the matrices are by no means the most extreme ones spanning the value space. Nevertheless, when traveling through this small triangle in the large space of valuations $\mathcal{V}$, we see significant and relevant differences in the perception and measure

consequences like the precision / recall and (segment-wise) false positive / negative rates for specific classes.

The remainder of this paper is organized as follows: In Section II we describe our use-case for decision rules in neural networks, in particular in semantic segmentation neural networks. Next, in Section III we explain the concept of decision rules in general and how they can be modified by valuating confusion costs between classes. We see various possibilities of defining the mentioned costs and provide two concrete examples in form of matrices in Section IV. Moreover, we present our spanned value space of confusion cost matrices and the setup for our experiments which follow in Section V. We show that different cost matrices are capable of considerably affecting the perception of a state-of-the-art semantic segmentation network in the setting of urban street scenes.

II Standard decision rule in neural networks

Semantic segmentation is the task of assigning each pixel of an image to one of the predefined classes $\mathcal{K} = \{1, \dots, N\}$. Suppose, we use a neural network for solving this task. Let $x \in \{(r, g, b)\}^{m \times n}$, $(r, g, b) \in \{0, \dots, 255\}^3$ be an "rgb" (red, green and blue light additively colored) input image with resolution $m \times n$. After processing the image x with a neural network we obtain a posterior probability distribution $p_{ij}(k|x)$ over all classes $k \in \{1, \dots N\}$ at location (pixel position in the image) $(i, j) \in \{1, \dots, m\} \times \{1, \dots, n\}$. The 3D tensor $p_{ij}(k|x)$ represents the softmax output of a neural network for semantic segmentation. The third dimension is given by the choice of $k \in \{1, \dots N\}$. This provided probability distribution expresses the confidence of the neural network as statistical prediction model to label the input correctly given the class k. The pixel-wise classification is then performed by applying the argmax function (pixel-wise) on the posterior probabilities / softmax output. This kind of decision making is called maximum a-posteriori probability (MAP) principle.

In the field of Deep Learning, following the MAP as *decision rule* is by far the most commonly used one. It maximizes the overall performance of a neural network, meaning in cases of large prediction uncertainty, this rule tends to predict classes that appear frequently in the dataset. However, classes of potential high importance, like in autonomous driving the classes traffic signs and humans, usually appear less frequently. These classes are rare in terms of the number of instances and the number of pixels in the dataset. This problem is in close connection to the fact that the MAP estimation considers all prediction mistakes to be equally serious which is in conflict with human intuition. Thus, a natural approach is to weight different prediction mistakes against each other.

III Cost-based decision rules in neural networks

Let Ω be a population consisting of $N \geq 2$ disjoint subsets. For each element $\omega \in \Omega$ we assume there exists one feature vector $x(\omega) \in S \subset \mathbb{R}^n$. Let

$$X : \Omega \to S \qquad K : \Omega \to \{1, \dots, N\} = \mathcal{K} \tag{1}$$

be random variables for feature vector x and class affiliation k, respectively. A *decision rule* can be defined as a map

$$d : \quad S \ \to \ \mathcal{K} \tag{2}$$

$$x(\omega) \mapsto \hat{k}(\omega) \tag{3}$$

which assigns an element from the feature space to one class. We say, $d(x) = \hat{k}$ is the predicted class for feature vector x. Furthermore, we describe the a-posteriori probability of an object to belong to

class k given feature x as

$$p(k|x) := P(K = k \mid X = x). \tag{4}$$

Usually, this probability is not known and needs to be estimated. We assume in the following that this is already accomplished, *e.g.*, $p(k|x)$ is approximated by the softmax output of a neural network.

Cost-based decision rules follow the idea of assigning one input to the class which minimizes the expected cost given one confusion cost function

$$c : \mathcal{K} \times \mathcal{K} \to \mathbb{R}_{\geq 0} := [\ 0, \infty\ ).$$

Considering all possible confusion cases we obtain a confusion cost matrix

$$C := (c(\hat{k}, k))_{\hat{k},k=1,\dots,N} \in \mathcal{V} \subset \mathbb{R}_{\geq 0}^{N \times N} \tag{5}$$

with $\hat{k}$ being the predicted class while k being the target class and

$$\mathcal{V} := \{\ C \in \mathbb{R}^{N \times N} \mid C_{jj} = 0, C_{ij} > 0, i, j \in \mathcal{K}\ \} \tag{6}$$

being the value space of all valid matrices C for cost-based decision rules. Hence, all elements of a valid matrix must be positive except the diagonal elements, which must equal 0, according to $\mathcal{V}$. Strictly speaking, $\mathcal{V}$ consists of equivalence classes since each C in combination with cost-based decision rules will produce the same output as $\mu C, \mu > 0$, *i.e.*, different scales of C do not change the output. Therefore, rather the costs of the classes relative to each other are decisive for the output instead of the absolute values.

In order to understand the just stated fact we define the expected cost with respect to confusion cost functions via

$$\mathbb{E}[\ c(k', K) \mid X = x\ ] = \sum_{k=1}^{N} c(k', k)\ p(k|x) \tag{7}$$

and the corresponding *cost-based* decision rule as

$$d(x; C) := \underset{k' \in \{1,\dots,N\}}{\arg\min} \sum_{k=1}^{N} c(k', k)\ p(k|x) \overset{(5)}{=} \underset{k' \in \{1,\dots,N\}}{\arg\min}\ C_{k'} \cdot \vec{p}(x) = \hat{k} \tag{8}$$

with $C_k := (C_{k1}, \dots, C_{kN})$ being the k-th row vector of $C \in \mathcal{V}$ and $\vec{p}(x) := (p(1|x), \dots, p(N|x))^T$ being the posterior probabilities vector conditioned on the feature x. This rule is optimal considering the expected costs.

Cost-based decision rules are strongly related to probability thresholding. The aim of probability thresholding is to make class predictions cost-sensitive during inference by moving the output threshold towards inexpensive classes. This is achieved by defining a confusion cost function of the form

$$c\left(\hat{k}, k\right) := \begin{cases} 0 & ,\ \text{if} \quad \hat{k} = k \\ \psi(k) & ,\ \text{if} \quad \hat{k} \neq k \end{cases},\ \psi(k) \in \mathbb{R}_{\geq 0} \tag{9}$$

with $\psi(k) > \psi(k')$ if we want the network to prefer predicting class k to predicting class k'. One special type of c is the simple symmetric cost function [Fah96]

$$c_s\left(\hat{k}, k\right) := \begin{cases} 0 & ,\ \text{if} \quad \hat{k} = k \\ \lambda & ,\ \text{if} \quad \hat{k} \neq k \end{cases},\ \lambda \in \mathbb{R}_{\geq 0} \tag{10}$$

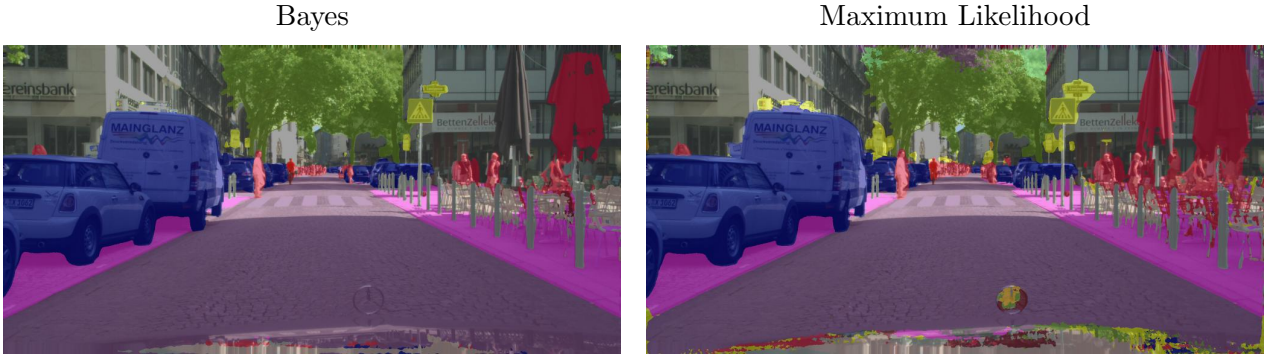


Figure 2: Illustration of two segmentation masks obtained with the Bayes decision rule (right) and the Maximum Likelihood decision rule (left). The difference between these two masks lies in the adjustment with the (pixel-wise) prior class probabilities in the decision rule during inference.

whose incorporation in the cost-based decision rule is equivalent to the MAP principle. Given $c_s(\hat{k}, k)$ all elements in the confusion cost matrix C_s are equal to the constant λ except the diagonal elements which are equal 0. Accordingly, the cost-based decision rule takes the form:

$$d(x; C_s) \overset{(10)}{=} \underset{k\in\{1,\dots,N\}}{\arg\min} \sum_{\substack{k'=1 \\ k'\neq k}}^{N} \lambda \cdot p(k'|x) = \underset{k\in\{1,\dots,N\}}{\arg\min}\ 1 - p(k|x) = \underset{k\in\{1,\dots,N\}}{\arg\max}\ p(k|x) \ =: \ d_{Bayes}(x). \quad (11)$$

In decision theory Equation (11) is the definition of the *Bayes* decision rule which is equivalent to the MAP principle and therefore also to the default classification principle in neural networks. However, the simple symmetric cost function implies an equal class weighting, *i.e.*, weighting every confusion between two classes (or each type of misclassification) equally. Depending on the purpose, this setting does not reflect the intuition of most people but is still applied in most deep learning state-of-the-art models.

A mathematically natural way to approach this problem is exchanging the simple symmetric with the inverse proportional cost function [Fah96] which is another special type of c. In light of confusion costs the latter cost function

$$c_p\left(\hat{k}, k\right) := \begin{cases} 0 & , \text{ if } \quad \hat{k} = k \\ \lambda/p(k) & , \text{ if } \quad \hat{k} \neq k \end{cases}, \ \lambda \in \mathbb{R}_{\geq 0} \quad (12)$$

weights each confusion with the inverse prior probability $1/p(k), p(k) \in (0, 1)$ of the potential target class k. In neural networks the class appearance frequencies in the training data correspond approximately to the priors. Considering the priors, we can put more emphasis on finding classes which are rare, *i.e.*, classes which have a low prior probability. The decision rule resulting from this is the *Maximum Likelihood* (ML) decision rule

$$d_{ML}(x) := \underset{k\in\{1,\dots,N\}}{\arg\max}\ p(x|k). \quad (13)$$

Now x is mapped to the class k for which the observed features are most typical, independent of a prior belief about the class frequencies. As presented in [Cha19], with respect to rare classes the application of the ML rule significantly reduces the number of false negative (overlooked) segments for rare classes, but to the detriment of producing substantially more false positive segment predictions.

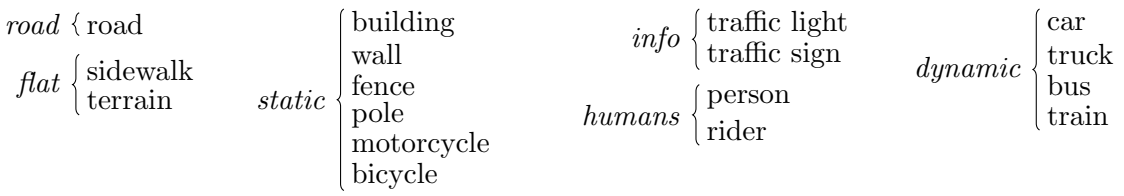


Figure 3: Class aggregates of Cityscapes classes that we use for simplicity in our experiments. Note that in the Cityscapes labeling motorcycles and bicycles in motion adhere to the class "rider".

One might argue that there is a "sweet spot" where the two error rates, the positive and negative one, are optimal. However, one might also argue that certain classes are still underweighted relative to others. We address both problems by applying the cost-based decision rule in combination with adjusting the confusion cost matrix C.

IV Setup of experiments

For our experiments we use the Cityscapes dataset with 19 semantic classes. In order to reduce the number of confusion cost values to be specified for the matrix C we aggregate classes that are treated similarly considering confusion costs, see Figure 3 for a first attempt although refined aggregations are probably more appropriate.

With 6 aggregated classes we define a 6×6 matrix. For performance evaluation we map the reduced matrix back to full 19×19 size such that all combinations between classes out of two aggregates have an equal confusion cost, *i.e.*, for two different non-empty aggregates $\mathcal{I}, \mathcal{J} \subset \mathcal{K}$ it holds

$$\mathcal{I} \cap \mathcal{J} = \emptyset \quad \Leftrightarrow \quad c(i,j) = c(i',j') \;\forall\; i,i' \in \mathcal{I},\; j,j' \in \mathcal{J}. \tag{14}$$

In addition, we set a small $\epsilon = 0.1$ for all confusions between different classes within an aggregate so that we apply the Bayes decision rule (only within an aggregate) without affecting the cost-based decision between aggregated classes, *i.e.*, for each non-empty aggregate $\mathcal{I} \in \mathcal{K}$ it holds

$$c(i,i') = \epsilon \;\forall\; i \neq i' \in \mathcal{I}, \quad c(i,i) = 0 \;\forall\; i \in \mathcal{I}. \tag{15}$$

Note that we suppress the "sky" class in our class aggregation although it is one of the originally trained classes. The reason is that we believe that overlooking the sky does not result in dangerous traffic scenarios. Therefore, we prevent the network from predicting sky by setting $C^T_{sky} = \{M\}^N$ with $M = 1000$ being a sufficiently large cost value. This implies that the confusion of any (target) class with sky is valuated with high cost. We set the cost for the converse confusion, when sky is the target class, to a constant value in order to not affect the class prediction between the remaining classes.

To gain further insight we define image regions of interest (RoI). These regions are derived from the pixel-wise class frequencies (priors) of the classes "road", "sidewalk", "building" and "sky" in the Cityscapes dataset. We obtain the 4 regions of interest (or 5 regions as the sidewalk RoI consists of two connected components) by assigning each pixel to the class with the highest class appearance frequency at the corresponding pixel location, see Figure 4.

For our experiments we further define two confusion cost matrices representing two extreme views in traffic scenes. On the one hand, we define the "altruistic" matrix C_A that prioritizes all traffic participants and particularly humans. On the other hand, we define the "egoistic" matrix C_E that only prioritizes the safety and comfort of the passenger inside the (ego-) car. The chosen cost values can be viewed in Figure 6. We compare the corresponding predictions with each other and also with the Bayes

Figure 4: Regions of interest derived from the priors of the classes building, road, sidewalk and sky in the Cityscapes dataset.

Cost matrix	Class	RoI	Precision	Recall
Altruistic	Person	1	41.12%	**99.81**%
Robotistic	Person	1	89.87%	94.98%
Egoistic	Person	1	**93.88**%	70.07%
Altruistic	Person	2	39.42%	**99.86**%
Robotistic	Person	2	88.36%	93.93%
Egoistic	Person	2	**95.07**%	54.81%
Altruistic	Building	1	22.56%	93.65%
Robotistic	Building	1	**80.99**%	94.94%
Egoistic	Building	1	15.15%	**99.93**%
Altruistic	Building	2	24.94%	95.22%
Robotistic	Building	2	**87.76**%	94.58%
Egoistic	Building	2	18.48%	**99.90**%

Table 1: Precision and recall rates for the different cost matrices. The rates are computed for the classes person and building in the street and the sidewalk RoIs, *i.e.*, RoI 1 and 2.

rule's prediction, respectively. The Bayes decision rule implies the matrix $C_R := (c_s(\hat{k}, k))_{\hat{k},k=1,\dots,N}$ which we term in the following the "robotistic" confusion cost matrix. This method is robotistic in the sense that, in any event, the only goal is to minimize all error rates. The convex combinations of these three presented matrices span a confusion value space

$$V := \{\ C \in \mathcal{V} \mid \alpha C_R + \beta C_A + \gamma C_E = C, \alpha + \beta + \gamma = 1,\ \alpha, \beta, \gamma \geq 0\ \} \tag{16}$$

(see Figure 7 and Figure 8). It is important to emphasize that $V \subset \mathcal{V}$ is only one subspace of a far bigger possible value space. There are even more extreme cost matrices that enlarge the space dramatically. There are also cost matrices expressing views in a completely different direction and therefore increasing the dimensionality of the space. However, our presented V is sufficient in order to show that it is already capable of changing our model's perception significantly.

V Experiments

As part of autonomous car driving systems, interpreting visual inputs is crucial in order to obtain a full understanding of the car's environment. The inference of an image in semantic segmentation [Eve15; Cor16] is performed at pixel level combining object detection and localization. In recent years, deep learning has achieved great success in a wide range of problems including semantic segmentation. Most state-of-the-art models are built on deep convolutional neural networks (CNNs) [Kri12; Sim14]. One important contribution to CNNs for semantic segmentation is the Fully Convolutional Network (FCN) [She16] which introduces end-to-end training taking input of arbitrary size and producing output of equal size. The network is one of the first using an encoder-decoder structure [Bad15; Ron15] whose encoder part is a classification network followed by the decoder part that projects convolved learned features back onto full pixel resolution. With the integration of atrous (also called dilated) convolutions [Yu15], that allows an exponential increase of the network's receptive field without loss of resolution, the performance of semantic segmentation networks is further significantly improved. One advanced module based on the latter operation is atrous spatial pyramid pooling (ASPP) [Che16]. It is one of the main contributions to the network DeepLabv3+ [Che18] which we use in the following in our experiments.

We demonstrate the performance of cost-based decision rules with different confusion cost matrices on the Cityscapes [Cor16] validation dataset. DeepLabv3+ is already pretrained on the latter dataset

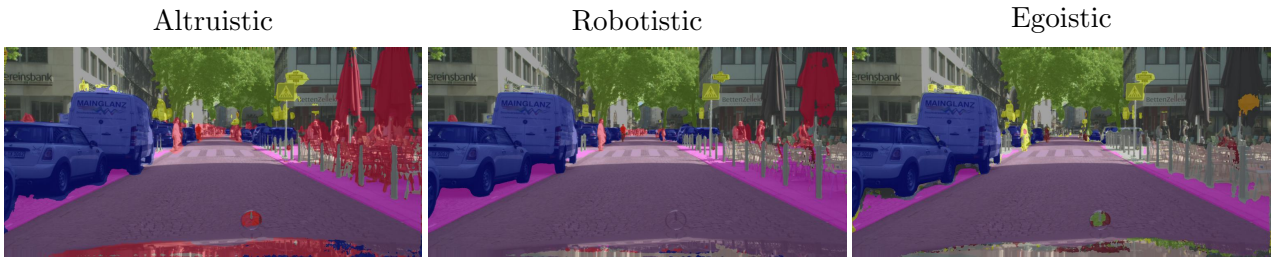


Figure 5: Illustration of three semantic segmentation masks and different perception obtained by the application of cost-based decision rules with an altruistic, a simple symmetric (robotistic) and an egoistic cost matrix.

$$C_A = \begin{array}{c} \begin{matrix} \text{“road”} & \text{“flat”} & \text{“static”} & \text{“info”} & \text{“human”} & \text{“dynamic”} \end{matrix} \\ \begin{pmatrix} 0 & 10^0 & 10^1 & 10^2 & 10^3 & 10^2 \\ 10^0 & 0 & 10^1 & 10^2 & 10^3 & 10^2 \\ 10^0 & 10^0 & 0 & 10^2 & 10^2 & 10^1 \\ 10^0 & 10^0 & 10^0 & 0 & 10^3 & 10^2 \\ 10^0 & 10^0 & 10^0 & 10^2 & 0 & 10^1 \\ 10^0 & 10^0 & 10^0 & 10^2 & 10^3 & 0 \end{pmatrix} \end{array} \begin{matrix} \text{“road”} \\ \text{“flat”} \\ \text{“static”} \\ \text{“info”} \\ \text{“human”} \\ \text{“dynamic”} \end{matrix}$$

$$C_E = \begin{array}{c} \overbrace{\text{potential target in columns}}^{} \\ \begin{pmatrix} 0 & 10^0 & 10^3 & 10^2 & 10^1 & 10^2 \\ 10^0 & 0 & 10^3 & 10^2 & 10^1 & 10^2 \\ 10^1 & 10^0 & 0 & 10^3 & 10^0 & 10^1 \\ 10^1 & 10^0 & 10^3 & 0 & 10^0 & 10^1 \\ 10^1 & 10^0 & 10^3 & 10^2 & 0 & 10^2 \\ 10^1 & 10^1 & 10^3 & 10^2 & 10^2 & 0 \end{pmatrix} \end{array} \Bigg\} \text{prediction in rows}$$

Figure 6: Two extreme confusion cost matrices that we study in our experiments. C_A represents the altruistic view prioritizing all traffic participants and particularly pedestrians. C_E represents the egoistic view prioritizing only the passenger in the (ego-) car. One element in the matrix expresses the cost that arises if we predict the class corresponding to the row and we confuse it with the potential target class corresponding to the column.

and implemented in TensorFlow [TF15]. The implementation and tuned weights are publicly available on GitHub. As network backbone, we choose the modified version of the Xception model [Cho16] that attains an mIoU score of 79.55% on the Cityscapes validation set with the application of the MAP / Bayes decision rule.

In the following, we perform our analysis for the classes “person” and “building” which are key classes in our problem setting of autonomous driving for the altruistic and egoistic view, respectively. Furthermore, we focus our studies on the regions of interest 1 & 2, the near field perception in front of the (ego-) car and to the side of the (ego-) car.

Pixel-wise precision vs. recall. For evaluation we first consider precision and recall. These two metrics are closely connected to the quantities false positive and false negative pixel predictions. A predicted pixel is a false positive (FP) if it falsely indicates an object's presence. A predicted pixel ignoring the presence of a present object is a false negative (FN). Therefore, precision is the percentage of a model's predicted pixels that match the ground truth, while recall is the percentage of ground truth pixels that a model predicts correctly, *i.e.*,

$$prc = TP \;/\; (TP + FP), \quad rec = TP \;/\; (TP + FN) \tag{17}$$

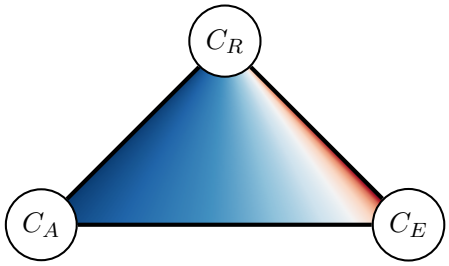


Figure 7: Confusion cost matrix space V spanned by our exemplary altruistic (C_A) and egoistic (C_E) cost matrix and the robotistic (C_R) cost matrix. Inside the triangle as heatmap the behavior of *rec*($V(C)$ | *person*), the recall of person pixels. Blue indicates high recall, red indicates low recall.

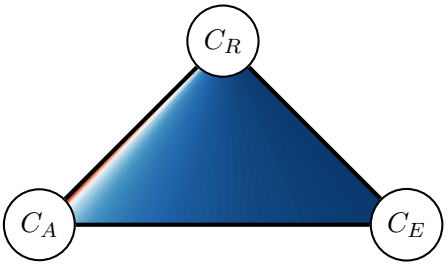


Figure 8: Confusion cost matrix space V spanned by our exemplary altruistic (C_A) and egoistic (C_E) cost matrix and the robotistic (C_R) cost matrix. Inside the triangle as heatmap the behavior of *rec*($V(C)$ | *building*), the recall of building pixels. Blue indicates high recall, red indicates low recall.

with TP being the true positives (pixels correctly classified according to the ground truth). The two evaluation metrics can be formulated as maps

$$prc : V \to [\ 0, 1\ ], \quad rec : V \to [\ 0, 1\ ] \tag{18}$$

expressing the neural network's predictive power depending on $C \in V$. The higher the value, the less prediction mistakes we obtain regarding falsely detected and non-detected pixels, respectively. The precision and recall scores of the different cost matrices in different regions of interest can be found in Table 1.

For the class person we observe that the recall is maximized when using C_A. Compared to C_R the reduction is 4.83 percent points in the street RoI and even 5.93 percent points in the sidewalk RoI. Even if the recall of person instances is already impressively high, C_A is still capable of boosting the performance in this metric such that nearly no person pixels are missed. However, to a striking detriment, the precision is reduced by about 48 percent points in both RoIs down to 41.12% and 39.42%, respectively. When using C_E persons are ignored to a large extent leading to a recall reduction of 24.91 percent points in the frontal RoI and 39.12 percent points in the sidewalk RoI in comparison to C_R. Consequently, the precision is increased by 4.01 and 6.71 percent points, respectively. With C_E DeepLabv3+ only predicts persons if the network indicates a high confidence about its decision. As expected there is a trade off between the metrics, *i.e.*, increasing one performance measure decreases the other and vice versa. Also noteworthy from this analysis is that DeepLabv3+ confuses only persons which are not completely visible, *e.g.*, persons standing behind cars or around corners. Only small parts of person instances are mainly overlooked, see also Figure 9.

For the class building we also observe this trade off but only between C_E and C_R. C_E improves the recall by 4.99 and 5.32 percent points while reducing the precision substantially by 65.84 and 69.28 percent points, respectively, for the street and the sidewalk RoI.

The behavior is different with respect to C_A. Regarding building segments, C_R performs better in both metrics in the frontal RoI. The recall is reduced by 1.29 and the precision by significant $58,43$ percent points. In the sidewalk RoI, the recall of C_A is slightly improved (0.64%) but the precision is again drastically reduced to 24.94%. Noteworthy from this analysis is that DeepLabv3+ has difficulties in detecting separated ground truth segments of building instances which arise from objects in front of buildings and splitting the instance's actual connected component in the ground truth, see also Figure 10.

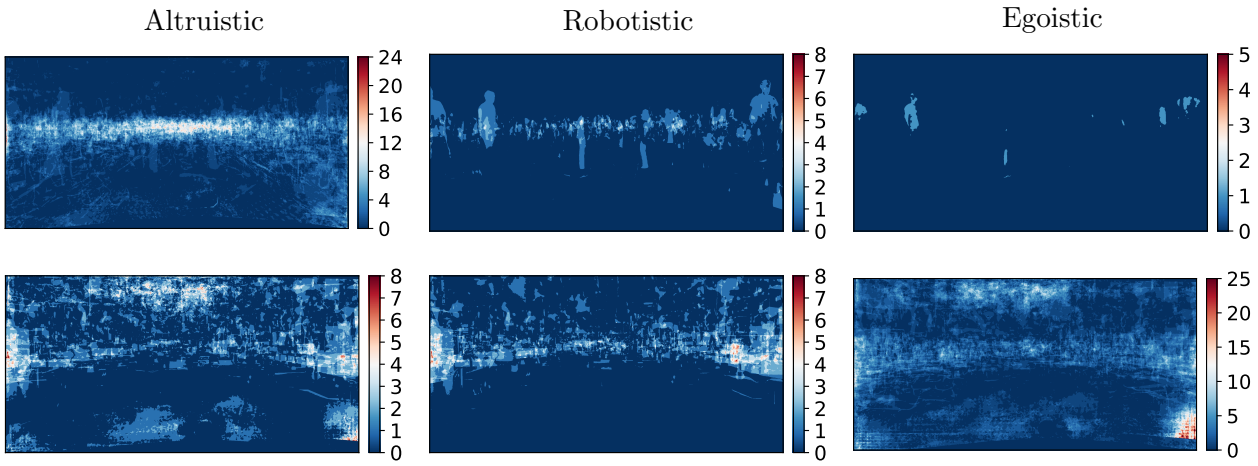


Figure 9: Falsely detected (false positive) person (top row) and building (bottom row) segments.

Segment-wise false-detection vs. non-detection. Another interesting quantity are the entire false-detections and non-detections of person and building segments when using the different cost matrices. In this regard, we now define a segment to be, depending on the considered prediction or ground truth mask, a false positive / negative if the segment's intersection over union (IoU) equals 0. Figure 9 and Figure 10 visualize the segments with $IoU = 0$ in the prediction mask and ground truth mask, again for the classes person and building. The presented heatmaps visibly confirm the findings from the precision and recall analysis. The application of cost-based decision rules changes the perception of DeepLabv3+ significantly. For instance, for the class person the altruistic cost matrix overproduces false positives but there are almost no overlooked person segments. On the contrary, the egoistic cost matrix almost completely refuses to predict the class person but is mostly correct in case it predicts a person segment. The robotistic cost matrix offers a balanced compromise between both prediction mistakes. Depending on people's individual sense of how the cost matrix should be defined, the presented observations will change again. Thus, what will remain open is a concrete suggestion to the inevitable definition of a confusion cost matrix.

VI Discussion

In this paper we illustrated the impact of cost-based decision rules on the perception of a state-of-the-art semantic segmentation neural network. In this framework, we discussed options for setting up cost-based decision rules ranging from the classical "robotistic" maximum a-posteriori probability principle over probability thresholding and the Maximum Likelihood decision rule to *ad hoc* "egoistic" and "altruistic" cost assignments to confusion events. Within the triangle of robotistic, egoistic and altruistic attitudes, we investigated precision and recall and also false positive and negative rates in two regions of interest for the classes "person" and "building" in the Cityscapes dataset. We demonstrated the metrics' dependence on the convex combination of the cost matrices from the three mentioned ethical attitudes spanning a triangle within a larger space of values.

On the technical side, many questions concerning the use of cost-based decision rules have to be clarified, *e.g.* the adaptation of cost matrices to prior probabilities or the impact on "downstream" modules like data fusion with other sensors and trajectory planning.

Let us turn to the ethical side of the discussion. The probabilistic nature of the output of the segmentation network makes a decision rule necessary. As different decision rules have non-converging

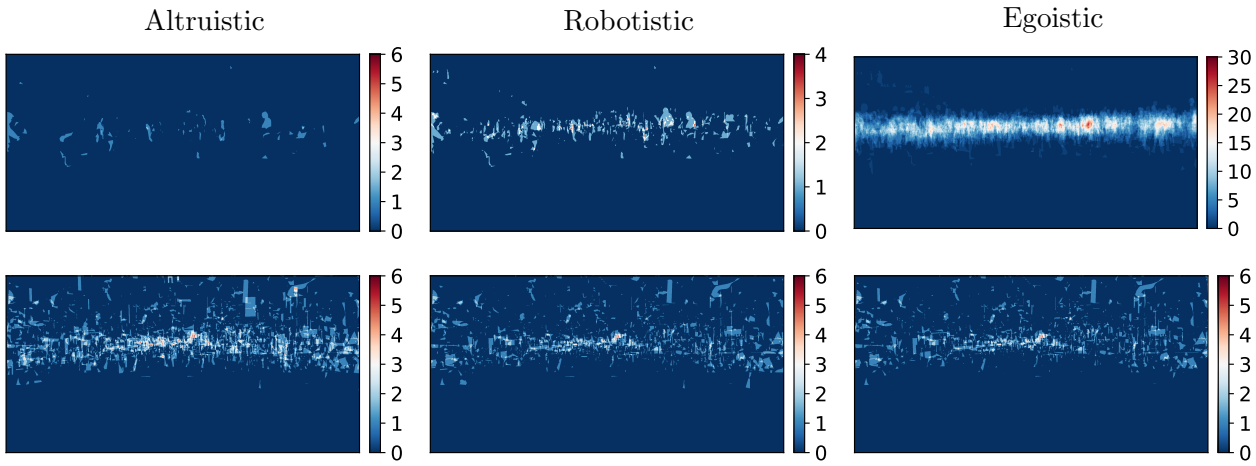


Figure 10: Non-detected (false negative) person (top row) and building (bottom row) segments.

consequences, a choice for a decision rule amounts to a choice where in the long run human lives are weighted against other considerations. This choice is therefore not one to be made from a purely technical side (by *e.g.* choosing the mathematically "natural" decision rule) but one that needs to recognize its ethical dimension. While technological advances may have an impact on these considerations they will not make the need for a decision rule obsolete.

This leads to the question: Which decision rule is the "right" one? As in most cases of moral uncertainty, different normative ethical schools of thought will provide different answers (see [Lin14, Ch.3] for a short non-technical introduction in the context of robot ethics). A deontological strategy would try to justify a certain choice of a decision rule by arguing for the rule itself being ethically "good", not considering what may follow from that choice. For instance, a strict rule-based implementation of the requirement by the ethics commission that "[t]he protection of individuals takes precedence over all other utilitarian considerations." [Eth17] may be interpreted to lead to a cost function that is never allowed to confuse a human for another object. A consequentialist strategy justifies a cost function by focusing on the consequences of a certain choice. This would involve the above analysis of the consequences of the egoistic and altruistic cost functions. Another approach refers to polling, using the ethical intuition of the majority of the people being asked. This can lead to strong cultural differences, as resulted in an analysis of Awad et al. in the context of trolley-like problems [Awa18].

It is not the aim of this paper to defend any specific approach or to provide an alternative answer to the above problem of choosing the "right" decision rule, but to make transparent the underlying ethical dimension of what may seem as mathematically innocuous "natural" choices. This transparency is a precondition for a responsible handling and open debate on these issues.

Acknowledgment. Robin Chan, Matthias Rottmann and Hanno Gottschalk acknowledge (partial) funding by Volkswagen Group Research through the contract "Maximum likelihood and cost-based decision rules in semantic segmentation".

References

[Foo67] Philippa Foot. "The Problem of Abortion and the Doctrine of Double Effect". In: *Oxford Review* 5 (1967), pp. 5–15 (cit. on p. 69).

[Fah96] L. Fahrmeir, A. Hamerle, and W. Häussler. *Multivariate statistical Methods (in German).* 2nd ed. Walter De Gruyter, 1996. ISBN: 978-3110138061 (cit. on pp. 70, 72, 73).

[Bro04] John Broome. *Weighing lives.* Oxford University Press, 2004 (cit. on p. 70).

[Kri12] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. "ImageNet Classification with Deep Convolutional Neural Networks". In: *Advances in Neural Information Processing Systems 25.* Ed. by F. Pereira et al. Curran Associates, Inc., 2012, pp. 1097–1105. URL: http://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks.pdf (cit. on p. 75).

[Lin14] Patrick Lin, Keith Abney, and George A Bekey. *Robot ethics: the ethical and social implications of robotics.* The MIT Press, 2014 (cit. on p. 79).

[Sim14] Karen Simonyan and Andrew Zisserman. "Very Deep Convolutional Networks for Large-Scale Image Recognition". In: *CoRR* abs/1409.1556 (2014). arXiv: 1409.1556. URL: http://arxiv.org/abs/1409.1556 (cit. on p. 75).

[Bad15] Vijay Badrinarayanan, Alex Kendall, and Roberto Cipolla. "SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation". In: *CoRR* abs/1511.00561 (2015). arXiv: 1511.00561. URL: http://arxiv.org/abs/1511.00561 (cit. on p. 75).

[Eve15] Mark Everingham et al. "The Pascal Visual Object Classes Challenge: A Retrospective". In: *International Journal of Computer Vision* 111.1 (Jan. 2015), pp. 98–136. ISSN: 1573-1405. DOI: 10.1007/s11263-014-0733-5 (cit. on p. 75).

[TF15] Martin Abadi et al. *TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems.* Software available from tensorflow.org. 2015. URL: https://www.tensorflow.org/ (cit. on p. 76).

[Ron15] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. "U-Net: Convolutional Networks for Biomedical Image Segmentation". In: *CoRR* abs/1505.04597 (2015). arXiv: 1505.04597. URL: http://arxiv.org/abs/1505.04597 (cit. on p. 75).

[Yu15] Fisher Yu and Vladlen Koltun. "Multi-Scale Context Aggregation by Dilated Convolutions". In: *CoRR* abs/1511.07122 (2015). arXiv: 1511.07122. URL: http://arxiv.org/abs/1511.07122 (cit. on p. 75).

[Che16] Liang-Chieh Chen et al. "DeepLab: Semantic Image Segmentation with Deep Convolutional Nets, Atrous Convolution, and Fully Connected CRFs". In: *CoRR* abs/1606.00915 (2016). arXiv: 1606.00915. URL: http://arxiv.org/abs/1606.00915 (cit. on p. 75).

[Cho16] François Chollet. "Xception: Deep Learning with Depthwise Separable Convolutions". In: *CoRR* abs/1610.02357 (2016). arXiv: 1610.02357. URL: http://arxiv.org/abs/1610.02357 (cit. on p. 76).

[Cor16] Marius Cordts et al. "The Cityscapes Dataset for Semantic Urban Scene Understanding". In: *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR).* 2016 (cit. on p. 75).

[Lin16] Patrick Lin. "Why ethics matters for autonomous cars". In: *Autonomous driving.* Springer, Berlin, Heidelberg, 2016, pp. 69–85 (cit. on p. 69).

[She16] Evan Shelhamer, Jonathan Long, and Trevor Darrell. “Fully Convolutional Networks for Semantic Segmentation”. In: *PAMI* (2016). URL: http://arxiv.org/abs/1605.06211 (cit. on p. 75).

[Tay16] Michael Taylor. “Self-driving mercedes-benzes will prioritize occupant safety over pedestrians”. In: *Car and Driver* Oct. 7 (2016). URL: https://www.caranddriver.com/news/a15344706/self-driving-mercedes-will-prioritize-occupant-safety-over-pedestrians/ (cit. on p. 70).

[Eth17] Ethics Commission on automated, networked driving of the German Federal Ministry for Transport, and Infrastructure. *Report of the ethics commission automated and networked driving (in German)*. Berlin, 2017 (cit. on pp. 69, 79).

[Lin17] Patrick Lin, Keith Abney, and Ryan Jenkins. *Robot Ethics 2.0: From Autonomous Cars to Artificial Intelligence*. Oxford University Press, 2017 (cit. on p. 69).

[Awa18] Edmond Awad et al. “The Moral Machine experiment”. In: *Nature* 563 (2018), pp. 59–64 (cit. on p. 79).

[Boa18] National Transportation Safety Board. *Preliminary Report Highway HWY18MH010*. 2018 (cit. on p. 69).

[Che18] Liang-Chieh Chen et al. “Encoder-Decoder with Atrous Separable Convolution for Semantic Image Segmentation”. In: *CoRR* abs/1802.02611 (2018). arXiv: 1802.02611. URL: http://arxiv.org/abs/1802.02611 (cit. on p. 75).

[Him18] Johannes Himmelreich. “Never Mind the Trolley: The Ethics of Autonomous Vehicles in Mundane Situations”. In: *Ethical Theory and Moral Practice* 21.3 (2018), pp. 669–684 (cit. on p. 69).

[WHO18] World Health Organization. *Road traffic injuries*. 2018. URL: https://www.who.int/news-room/fact-sheets/detail/road-traffic-injuries (visited on 02/27/2019) (cit. on p. 69).

[Cha19] Robin Chan et al. “Application of Decision Rules for Handling Class Imbalance in Semantic Segmentation”. In: *CoRR* abs/1901.08394 (2019). arXiv: 1901.08394. URL: https://arxiv.org/abs/1901.08394 (cit. on p. 73).

Application of Maximum Likelihood Decision Rules for Handling Class Imbalance in Semantic Segmentation

Robin Chan[1], Matthias Rottmann[1], Fabian Hüger[2], Peter Schlicht[2] and Hanno Gottschalk[1]

[1]School of Mathematics and Natural Sciences, University of Wuppertal
[2]Architecture and AI Technologies, Automated Driving, Volkswagen Group Research

Abstract. One difficulty, that occurs while training neural networks for semantic segmentation, is class imbalance in "real world" datasets. Consequently, a neural network trained on unbalanced data in combination with standard maximum a-posteriori classification may ignore classes that are rare in terms of their frequency in the dataset. However, these classes are often of highest interest. We approach such potential misclassifications in semantic segmentation by applying different variants of the *Maximum Likelihood* decision rule which shows to significantly improve recall with some sacrifice in precision. In particular, we provide an in-depth comparison between a global and a localized version for the semantic segmentation of street scenes.

I Introduction

A common issue with "real world" datasets and classification tasks is an unbalanced class distribution, i.e., a dominant portion of examples is assigned to only a few groups. Class imbalance in datasets can have a detrimental effect on classification performance of neural networks (NNs) as they tend to predict classes that appear frequently [Lóp13; Bud18]. With respect to safety relevance, the semantic segmentation of street scenes is one instance of applications where NNs are constantly setting ever new performance records. As semantic segmentation tasks can be viewed as pixel-wise classification problems, they encounter the same difficulties when dealing with class imbalance in the data. Methods overcoming class imbalance are mostly adopted from traditional machine learning and can be divided into two main categories [Kra16; Bud18; Joh19].

The first category are *sampling-based* methods that operate directly on a dataset with the aim to balance its class distribution. Oversampling and undersampling are basic strategies that have been proposed by [Van07]. In their simplest form, the dataset is balanced by increasing the number of instances from "minority" classes and by decreasing the number of instances from "majority" classes, respectively. Consequently, when applying sampling techniques one either discards useful training data or one duplicates data that might cause overfitting. Moreover, sampling approaches are hardly applicable for semantic segmentation datasets as the class imbalance is mostly intrinsic [Lee18].

The second category are *algorithm-based* methods that do not modify the data and make use of cost-based training and output thresholding. The idea behind these strategies is to assign different costs to classification mistakes for different classes. Accordingly, one possibility is to minimize the misclassification cost instead of the standard loss function during training [Cae15; Wan16; Bul17]. However, this biases the softmax probability output of the NN and requires the cost parameters to be carefully calibrated, resulting in an increased training time. Another possibility is to make class predictions cost-sensitive during inference after the network is fully trained by moving the output threshold towards inexpensive classes [Bud18]. The application of decision rules falls under the latter category. Although this approach does not change a model's classification capabilities, it is still appropriate for handling class imbalance since it shifts the priority to predicting certain classes.

Using convolutional neural networks (CNNs) trained for the task of semantic segmentation, the output mask is usually obtained by the maximum a-posteriori probability (MAP) principle, i.e., by

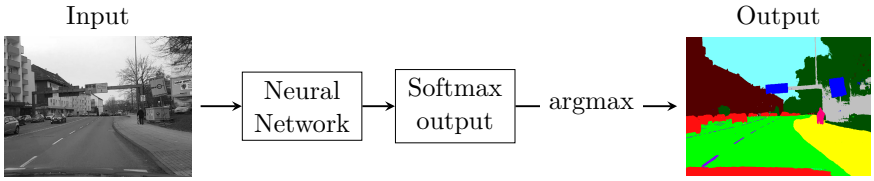


Figure 1: Illustration of the standard procedure for semantic segmentation using the maximum a-posteriori principle (argmax function applied on softmax probabilities output).

applying the argmax function to the pixel-wise softmax output. CNNs as statistical models in combination with MAP aim at minimizing the chance of a misclassification which in decision theory is known as Bayes rule. Another mathematically natural approach from decision theory is the *Maximum Likelihood* (ML) rule [Fah96]. While the MAP / Bayes rule incorporates a prior belief about the semantic classes, the ML rule decides only by means of the observed features and chooses the class that is most typical for the given pattern. This is particularly helpful when features are observed outside of the range of learned concepts.

In this work we deal with different CNNs for semantic segmentation trained on the publicly available street scenes dataset Cityscapes [Cor16] and a proprietary street scenes dataset called DS20k by Volkswagen Group Innovation. By that choice of dataset, we focus on the following class imbalance scenario:

- the task of segmenting person instances in street scene images

in which these instances usually cover only a small portion of the image data. In this kind of situations, detecting the minority classes is crucial for safety reasons. We analyze the impact of applying ML instead of Bayes decision rule for CNNs for semantic segmentation. In other words, the dataset and the training procedure remain unchanged and the decision rules are only interchanged at inference. This provides the most lightweight solution that can be easily added on top of every NN to combat the effects of class imbalance. As ML follows the mathematical concept that only uses the likelihood, i.e., ML only uses observed data as evidence for decision making, it is a legitimate approach to replace the standard MAP principle with ML. We do not apply other arbitrary decision rules as it may lead to ethical difficulties due to explicitly defining costs. Our main focus in this work is to reduce false-negative predictions of rare classes.

The remainder of this paper is structured as follows: In Section II we introduce decision rules in general and how they are employed in combination with neural networks. Afterwards, we describe our prior class probability estimation approach and setup of experiments in Section III and Section IV, respectively. We discuss the performance of ML in particular in comparison with Bayes for different datasets and priors in Section V. In a sense of an outlook, we discuss further possible constructions of priors in Section VI and provide an overview of safety challenges in Section VII.

II Decision Rules in Neural Networks

A neural network for semantic segmentation with a softmax output layer can be seen as a statistical model that provides for each pixel z of one image a probability distribution $p_z(y|x,w)$ on q predefined semantic class labels $y \in \mathcal{Y} = \{y_1, \ldots, y_q\}$, given the data $x \in \mathcal{X}$ and the weights w. The predicted class in z is then usually obtained by

$$\hat{y}_z^b(x,w) = \text{argmax}_{y \in \mathcal{Y}} p_z(y|x,w). \tag{1}$$

By assigning the features x to the class with the largest posterior probability $p_z(y|x, w)$, the chance of an incorrect class estimation is minimized which is equivalent to the *Bayes* rule from decision theory. Following the Bayes rule is by far the most commonly used decision principle in the field of deep learning, i.e., in cases of large prediction uncertainty, this rule tends to predict classes that appear frequently in the dataset. However, classes of high interest are often minority classes and appear less frequently. In this context, the *Maximum Likelihood* rule maps features x to the class with the largest conditional likelihood:

$$\hat{y}_z^{ml}(x, w) = \text{argmax}_{y\in\mathcal{Y}} p_z(x|y, w) = \text{argmax}_{y\in\mathcal{Y}} p_z(y|x, w)/p_z(y) \tag{2}$$

In the latter, the class affiliation y is an unknown parameter that needs to be estimated using the principle of maximum likelihood. The ML rule aims at finding the class y for which the features x are most typical (according to the observed features in the training set) without incorporating the prior probability (or prior belief about the frequency) $p_z(y)$ of the particular class. The difference between these two decision rules only lies in the adjustment with the prior class probabilities. In addition, Bayes weights every type of confusion between two different classes equally while ML weights one confusion with the inverse prior probability of the possible target class, therefore having optimality properties with respect to the simple symmetric and inverse proportional cost function, respectively [Fah96]. Obviously, both decision rules are equal when the prior class distribution is balanced, i.e., $p_z(y_1) = \ldots = p_z(y_q)$.

III Choice of Priors

Priors are essential for the implementation of the Maximum Likelihood (ML) decision rule. We approximate them using the training set since our network is trained on these unbalanced data.

The most intuitive approach for balancing data, following traditional machine learning methods, is to count the pixels in the training set's annotations belonging to each class and use the resulting normalized class frequencies as priors. In this way location information is not taken into account by the priors. Since a model's performance in semantic segmentation is highly dependent on localization capabilities, we believe that more advanced methods are needed.

One possibility in order to maintain information about position-specific data imbalance is to provide the class frequencies at every pixel. Therefore, the obtained priors have the same 3D tensor shape and dimensions as the softmax output. They do not only depend on the class but also on the location. Note that this definition of priors is consistent with a pixel-wise realization of ML, cf. Equation (2).

In our experiments, we compare the use of ML with these two priors against each other as well as against the standard Bayes rule, which can be considered as ML using constant priors.

IV Setup of Experiments

Given a dataset to be examined, we first conduct a statistical analysis on the training set's class distribution. We construct two different types of priors, according to the concepts presented in Section III, by deriving the class frequencies of the training set's annotations.

Let $y_z(x) \in \mathcal{Y}$ denote the target class at pixel location $z \in \mathcal{H} \times \mathcal{W} = \{1, \ldots, m\} \times \{1, \ldots, n\}$ given features $x \in \mathcal{X}$. For all $y \in \{y_1, \ldots, y_q\}$ and for all $z \in \mathcal{H} \times \mathcal{W}$ we term:

(i) the normalized class frequencies as *global priors*

$$\hat{p}_z^{glob}(y) = \sum_{x\in\mathcal{X}} \sum_{z'\in\mathcal{H}\times\mathcal{W}} \frac{1_{\{y=y_{z'}(x)\}}}{(nm|\mathcal{X}|)} \tag{3}$$

(ii) and the position-specific normalized class frequencies as *local priors*

$$\hat{p}_z^{loc}(y) = \sum_{x \in \mathcal{X}} 1_{\{y=y_z(x)\}} \ / \ |\mathcal{X}| \quad . \tag{4}$$

For the purpose of avoiding divisions by zero, see Equation (2), we clip all prior values below $\varepsilon = 10^{-5}$ to this lower bound. In our experiments, we use two street scene datasets, namely Cityscapes [Cor16] and DS20k by Volkswagen Group Innovation. In both of these datasets the pedestrian class is underrepresented and evidently of high interest, consequently this class serves as canonical candidate for our analysis of decision rules. We evaluate the performance of different decision rules at pixel and at segment level for the pedestrian class.

V Quantitative Analysis

As applying ML with global priors is a commonly used procedure in machine learning for handling class imbalance, this approach can serve as baseline and we first focus on the performance comparison between our contribution, ML with local priors (in the following three subsections abbreviated as ML), and Bayes.

V.i Experiments: Cityscapes

The Cityscapes dataset [Cor16] consists of 5,000 pixel-annotated street scene images (resolution 2048×1024 pixel) recorded in 50 different cities from the perspective of a car driver. A subset of 2,975 images are used for training, another subset of 500 images for validation purposes. The annotation comprises 29 semantic classes of which 19 are trained. The dataset also provides coarse categories, the 19 trained classes are aggregated into 8 categories. Many state-of-the-art models for visual perception of street scenes are assessed with Cityscapes.

Class Imbalance in Dataset. The authors of the dataset report the number of finely annotated pixels per class in the whole dataset which reveals an unbalanced class distribution. For instance, the total number of pixels belonging to the class *person* is $\sim 10^8$, whereas $\sim 3 \cdot 10^9$ pixels belong to the class *road*. This is a difference greater than one order of magnitude. The confusion of these two classes would possibly lead to fatal situations and must be avoided, especially in the domain of near field perception. However, there are labels such as *wall*, *fence* or *pole* that are as rare as *person* in terms of pixel frequency. This is due to the fact that the images are recorded in urban street scenarios and therefore naturally boost the number of pedestrian instances. Applying priors according to that distribution might lead to class weightings preferring static objects over humans. This policy does not reflect the intuition of most people. For this reason, we deal with the associated categories of the training classes that distinguish objects more superficially. The category distribution remains unbalanced with *humans* being significantly underrepresented. A corresponding visualization can be found in Figure 2.

Prior Class Distribution. In contrast to Figure 2 that shows the category distribution on full image level, Figure 3 visualizes the distribution for class *human* per pixel. For instance, from the latter figure we conclude that during training there are no persons seen at the top or the bottom part of the image as the local prior, according to Equation (4), is primarily zero in these regions. Thus, we expect the network to be biased towards not predicting a person in that area which might be misleading, e.g., when the street is ascending or descending.

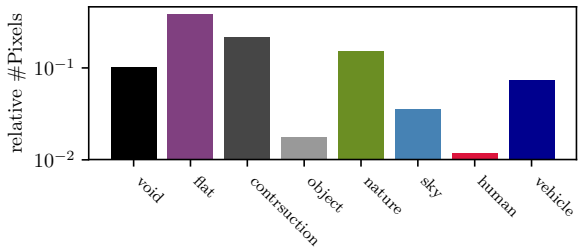


Figure 2: Pixel-wise distribution of category in the Cityscapes training set.

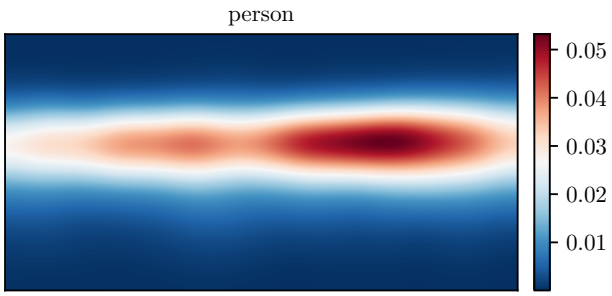


Figure 3: Estimated localized and pixel-wise prior distribution of class *human*.

Numerical Results. We evaluate the performance of Bayes and ML on the validation images that were not used during training. The networks we use in our experiments are DeepLabv3+ models [Che18] with MobileNetV2 [San18] and Xception [Cho17] as network backbones. We use pretrained weights that are publicly available such that no training is required.

The intersection over union (IoU), also referred to as Jaccard index [Jac12], is the standard evaluation metric in semantic segmentation and quantifies the similarity between the target mask and the NNs output. The lightweight MobileNetV2 model (MN) achieves an IoU score for class *human* of 60.4% while the more complex Xception model (XC) achieves an IoU of 79.5% which is close to state-of-the-art at the time of writing. Furthermore, for MN we observe a superiority of Bayes over ML in this metric by 5.2 percent points. For XC the superiority is even clearer with 10.8%. This finding is not unexpected since the IoU is an overall performance measure and Bayes maximizes the overall probability of a correct class prediction. ML aims at decreasing the risk of missing any pixels belonging to class *human*. Thus, the lower IoU achieved by ML is primarily caused by an overproduction of false-positives.

While precision expresses the ratio of correct predictions and all predicted instances, recall expresses the the ratio of correct predictions and actually existing (ground truth) instances [Faw06]. The amplification of predicting *human* segments by introducing priors leads to producing additional false-positives compared to Bayes. Since the strength of ML lies in the ability of finding minority classes, we investigate to which extent precision is sacrificed in order to improve recall. The pixel-wise scores are summarized in Table 1. For MN, we observe that ML increases the recall by 8.4% up to 84.9% whereas the precision decreases by 13 percent points down to 61.2%. With XC, the gain in recall of 5 pp. up to 94.7% result in a reduction of precision of about 15.9 pp. down to 71.5%. For both networks the observed effects are similar showing the trade-off between the two metrics when using ML. However, the trade-off for XC is more pronounced and might indicate that ML is more appropriate for weak networks. This is also in accordance with the observed IoU performance reduction which is more distinct for XC than for MN.

As we expect whole segments to be false-positive in ML prediction masks but also to recognize more person instances compared to Bayes, we extend our analysis to segment-wise precision and recall. The empirical cumulative distribution functions (CDFs) of the class *human* for segment-wise precision and recall can be found in Figure 4. Let F_1 and F_2 be two CDFs, then F_1 is *dominated stochastically to 1st order* by F_2 [Pfl07],

$$F_1 \prec F_2, \text{ if } F_1(\eta) \geq F_2(\eta) \; \forall \; \eta. \tag{5}$$

In the following, we denote the CDFs of the Bayes decision rule regarding precision and recall by F_B^p and F_B^r, respectively. Analogously, F_{ML}^p, F_{ML}^r refer to the ML decision rule.

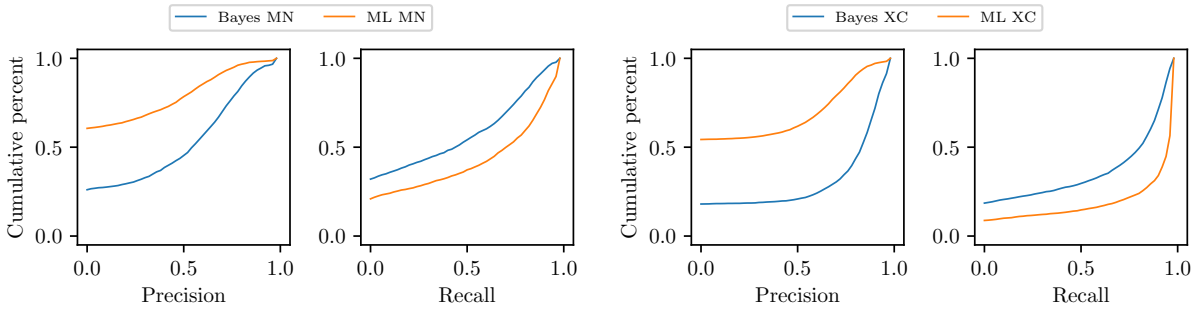


Figure 4: Empirical cumulative distribution functions (CDFs) in Cityscapes for segment-wise precision and recall of class *human* using DeeplabV3+ models. In the left two panels the CDFs for MobileNetV2 (MN) and in the right two panels the CDFs for Xception (XC).

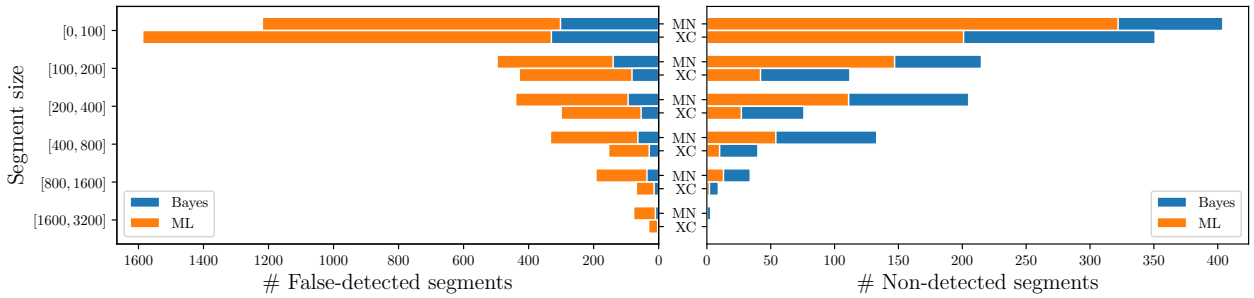


Figure 5: Bar plot of false-detected and non-detected human segments with the Bayes and localized Maximum Likelihood (ML) decision rule depending on the size in Cityscapes.

We observe a clear advantage of Bayes in terms of precision since $F^p_{ML} \prec F^p_B$ for both MN and XC. For any precision value η, in particular for low precision values, the frequency that one instance's precision is below η is significantly less with Bayes than with ML. The average difference is about 25% for MN and even 37% for XC. Hence, Bayes predicts *human* segments with better precision than ML.

In terms of recall, we observe the opposite behavior: $F^r_B \prec F^r_{ML}$, i.e., ML is superior over Bayes in this metric. The average difference is about 15% and 17%, respectively. The steep ascent of the ML curves in Figure 4 additionally indicates that most ground truth segments are predicted with high recall, i.e., most ground truth segments are almost entirely detected. More relevantly, ML significantly reduces the number of non-detected segments, i.e., $F^r_B(0) > F^r_{ML}(0)$.

To this end, we define that a connected component k is predicted correctly (and thus is a true positive), if there is a ground truth connected component k' of the same class with $k \cap k' \neq \emptyset$. Hence, we consider a predicted segment with precision equal to zero as *false-detection* (false-positive) and a ground truth segment with recall equal to zero as *non-detection* (false-negative). The corresponding quantities for different segment size bins are summarized in Figure 5. There is a noticeable decrease in the frequency of ML false-detections if the segment size increases, i.e., larger predicted segments are less likely to be entirely incorrect. For Bayes segments, the same tendency holds. Moreover, we see for every segment size bin that the amount of ML false-detections considerably exceeds the amount of Bayes false-detections. For the non-detection of humans we find a clear advantage in favor of ML or all component sizes. The amount of ML non-detections relative to the amount of Bayes non-detections decreases for increasing component sizes. This result indicates an uncertainty of the network in finding humans which can be alleviated by using ML.

	IoU			Precision			Recall		
Decision Rule	Bayes	ML		Bayes	ML		Bayes	ML	
Priors Adjustment	none	local	global	none	local	global	none	local	global
Dataset Cityscapes									
MobileNetV2	60.4	55.2	54.8	74.2	61.2	59.7	76.5	84.9	87.0
Xception	79.5	68.7	66.3	87.4	71.5	68.5	89.7	94.7	95.3
Dataset DS20k									
FRRN	78.9	74.6	71.0	83.8	78.3	75.1	92.7	94.0	94.4

Table 1: Pixel-wise precision, recall and intersection over union (IoU) for Bayes and ML with different priors.

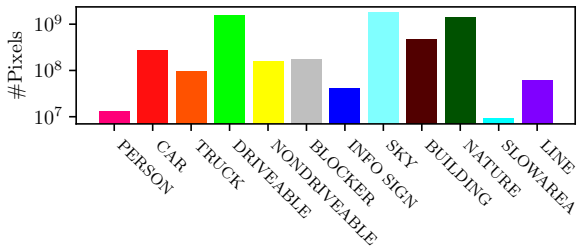


Figure 6: Pixel-wise class distribution of the DS20k training set.

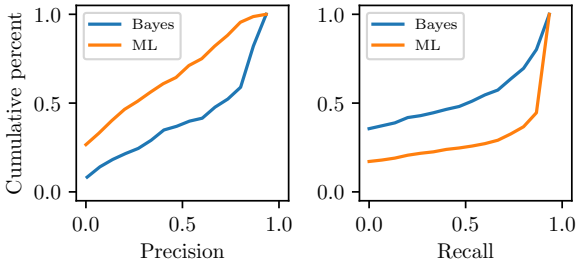


Figure 7: CDFs in DS20k for segment-wise precision and recall of class *person*.

V.ii Experiments: DS20k

The second street scenes dataset for our experiments is the proprietary DS20k by Volkswagen Group Innovation. Unlike the Cityscapes dataset, DS20k also contains scenes from highways and country roads. Consequently, pedestrians are even more underrepresented. The class distribution of the training set, see Figure 6, differs significantly from a uniform distribution. We perform tests analogously to Cityscapes using the Full-Resolution Residual Network (FRRN) by [Poh17] on 200 images that were not used during training.

Due to the more drastic class imbalance in DS20k compared to Cityscapes and the fact that we trained the FRRN on DS20k only, we expect ML to better unfold its strength on this dataset. The pixel-wise evaluation scores are reported in Table 1 and the CDFs of segment-wise precision and recall are shown in Figure 7. In general, we observe a similar behavior for DS20k as for Cityscapes regarding the evaluation metrics and thus focus on discussing the differences. In comparison to the DeeplabV3+ Xception model for Cityscapes, the Bayes performance of the FRRN on DS20k is in the same range for IoU, precision and recall. In contrast to this, when using ML the FRRN sacrifices less precision (15.9% vs. 5.6%) to achieve a similar recall of 94%. Noteworthy, the FRRN's Bayes recall of 92.7% is already quite high.

Again, more relevantly are the person instances that are entirely overlooked and the CDFs, i.e., $F_B^r(0) = 0.36$ and $F_{ML}^r(0) = 0.17$, indicate that ML significantly decreases the number of non-detections. Indeed, while Bayes misses 125 segments, ML only misses 55 which is less than half of the amount. Figure 9 visualizes the overlooked person segments over the whole test set. We notice that the additional instances detected by ML are rather small segments and conclude that ML is particularly powerful on DS20k for small object sizes.

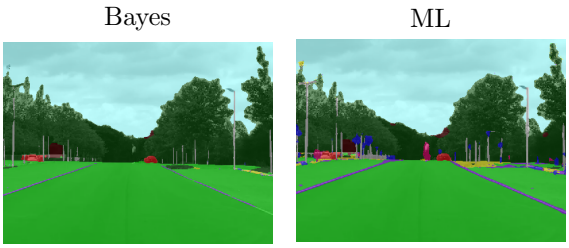


Figure 8: Effects of using ML with local priors in combination with the FRRN trained on DS20k. While Bayes (left) entirely overlooks the person in the image, ML (right) fully detects that instance.

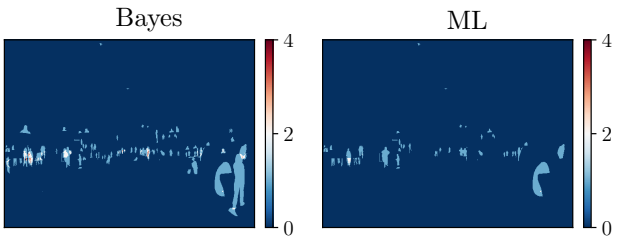


Figure 9: Heatmap of non-detected / overlooked person instances in DS20k. ML misses significantly less persons segments than Bayes. This holds particularly for instances that are rather small.

V.iii Local vs. Global Priors

In the following we compare the differences in performance when employing ML with either local or global priors. Throughout all investigated datasets, we observe that ML with global priors sacrifices more in terms of precision but achieves higher recall compared to ML with local priors, see Table 1. This is primarily due to the similar position-specific class distribution between training and validation set. In other words, objects of rare classes in the validation set appear in the same image region as in the training set and consequently where the localized prior class probability is high (in a relative sense). Global priors, on the contrary, are obtained by averaging over all image parts leading to a lower prior class probability (compared to local priors) and a stronger boost for rarest class objects in the validation set. More interesting would be cases in which the class distribution differs significantly from the one given by the training data as an opposite behavior is to be expected.

VI Rethinking the Choice of Priors

For NNs with a fully connected layer for dense pixel prediction, using local priors might be appropriate as every output pixel has its own weights and can learn dependent on its location. However, NNs for semantic segmentation are mostly fully convolutional, i.e., they do not use any fully connected layers.

In convolutional layers, weights are shared across all patches of the input such that the learned weights are invariant with respect to image positions, i.e., convolution and position translations theoretically commute. In practice, this does not strictly apply to CNNs for semantic segmentation due to boundary effects and max-pooling operations. The receptive field describes the region in the input space that a particular (feature) pixel is looking at and it enlarges the more convolutions and pooling layers are stacked. Consequently, large receptive fields might include parts outside of the input image's dimension. Although it has been observed that the pixels in the center of the receptive field contribute most to the output and that the effective receptive field size is only a fraction of the full theoretical size [Luo16], it is not evident to exclude that CNNs use boundary information for approximating pixel locations. In general, max-pooling is only translation invariant under shifts that are multiples of the pooling regions size. For CNNs an averaging of pixel-wise priors along the orbits of the translation group (adequately coarsened by pooling) could be appropriate [Coh16].

We believe that priors in NNs do exist and that different NN architectures learn different priors. However, according to the listed reasons in this section and Section III, the main difficulty lies in finding the "right" priors.

VII Safe Neural Networks in Automated Driving Systems

Neural networks are supposed to be used for executing safety-relevant tasks in a broad field of applications. As such machine learning algorithms are not explicitly programmed and are often even defined through non-understandable rules, this lead to several challenges in order to assure safety in applications like automated driving. Moreover, the safety standards available within the automotive industry have been defined without clearly considering the specifics of machine learning algorithms, cf., ISO 26262 *Road Vehicles – Functional Safety*.

In this work, we particularly address the difficulty of collecting highly representative and complete data. A uniformly distributed dataset is often desired, however, in practice it is not always realistic, e.g., due to the class imbalance inherent in street scenes data itself. This makes additional software components working in parallel to NNs at runtime crucial. The Maximum Likelihood rules can be easily integrated in such so-called *observers* for checking plausible sensitivity of NNs via saliency maps (i.e. in semantic segmentation by comparing the Bayes and Maximum Likelihood mask). In this way, challenges like whether NNs replicate the high confidence labels presented during training or whether their decisions are based on meaningful features observed in the input data can be monitored. To ensure safety, the above challenges must be taken into consideration. As decision rules are computationally cheap, they are therefore highly suitable for the runtime monitoring task of the deployed system.

VIII Conclusion & Outlook

In this work, we conducted an in-depth comparison of the ML and Bayes decision rule for semantic segmentation networks trained on unbalanced street scene datasets. In our tests, we observe that ML is able to detect the underrepresented classes more frequently than Bayes. Indeed, the pixels that Bayes and ML classify differently indicate that a less frequent class might be overlooked. ML detects significantly more instances of the rare class in comparison to Bayes, but to the detriment of producing substantially more false-detections which makes ML not reliable for always predicting the rare class correctly. We introduced pixel-wise priors in order to handle class imbalance depending on the location in the image. They have shown to be less prone to false-positives than averaged priors over all pixels in turn with moderately more false-negatives.

We believe that priors exist in CNNs and plan on developing more sophisticated techniques for estimating them. Apart from this, it is important to emphasize that ML only post-processes the softmax output of a neural network. This can be done simultaneously while applying the usual Bayes rule. In the end, we obtain two prediction masks of which the additional ML mask is produced computationally almost for free. What remains is to develop methods to draw plausible conclusions in order to combine both segmentation masks. Using dispersion measures in combination with meta-classification [Rot18] would be a promising automated solution approach. Furthermore, the ML prediction can serve as uncertainty mask revealing labeling mistakes of training data or indicating new unlabeled images of high prediction uncertainty which then can be annotated and included in the training process in the manner of active learning.

Acknowledgment. Robin Chan, Matthias Rottmann and Hanno Gottschalk acknowledge (partial) funding by Volkswagen Group Innovation through the contract “Maximum likelihood and cost-based decision rules in semantic segmentation”.

References

[Jac12] Paul Jaccard. "The Distribution of the Flora in the Alpine Zone". In: *New Phytologist* 11.2 (1912), pp. 37–50. DOI: 10.1111/j.1469-8137.1912.tb05611.x. URL: https://nph.onlinelibrary.wiley.com/doi/abs/10.1111/j.1469-8137.1912.tb05611.x (cit. on p. 87).

[Fah96] L. Fahrmeir, A. Hamerle, and W. Häussler. . German. 2nd ed. Walter De Gruyter, 1996. ISBN: 978-3110138061 (cit. on pp. 84, 85).

[Faw06] Tom Fawcett. "An introduction to ROC analysis". In: *Pattern Recognition Letters* 27.8 (2006), pp. 861–874. ISSN: 0167-8655. DOI: https://doi.org/10.1016/j.patrec.2005.10.010. URL: http://www.sciencedirect.com/science/article/pii/S016786550500303X (cit. on p. 87).

[Pfl07] Georg Christian Pflug and Werner Römisch. *Modeling Maeasuring and Managing Risk*. 1st ed. World Scientific, 2007. ISBN: 978-9812707406 (cit. on p. 87).

[Van07] Jason Van Hulse, Taghi M. Khoshgoftaar, and Amri Napolitano. "Experimental Perspectives on Learning from Imbalanced Data". In: *Proceedings of the 24th International Conference on Machine Learning*. ICML '07. ACM, 2007, pp. 935–942. ISBN: 978-1-59593-793-3. DOI: 10.1145/1273496.1273614. URL: http://doi.acm.org/10.1145/1273496.1273614 (cit. on p. 83).

[Lóp13] Victoria López, Alberto Fernández, Salvador García, et al. "An insight into classification with imbalanced data: Empirical results and current trends on using data intrinsic characteristics". In: *Information Sciences* 250 (2013), pp. 113–141. ISSN: 0020-0255. DOI: https://doi.org/10.1016/j.ins.2013.07.007. URL: http://www.sciencedirect.com/science/article/pii/S0020025513005124 (cit. on p. 83).

[Cae15] Holger Caesar, Jasper Uijlings, and Vittorio Ferrari. "Joint Calibration for Semantic Segmentation". In: *Proceedings of the British Machine Vision Conference (BMVC)*. Ed. by Mark W. Jones Xianghua Xie and Gary K. L. Tam. BMVA Press, Sept. 2015, pp. 29.1–29.13. ISBN: 1-901725-53-7. DOI: 10.5244/C.29.29 (cit. on p. 83).

[Coh16] Taco Cohen and Max Welling. "Group Equivariant Convolutional Networks". In: *Proceedings of The 33rd International Conference on Machine Learning*. Ed. by Maria Florina Balcan and Kilian Q. Weinberger. Vol. 48. Proceedings of Machine Learning Research. New York, New York, USA: PMLR, June 2016, pp. 2990–2999. URL: http://proceedings.mlr.press/v48/cohenc16.html (cit. on p. 90).

[Cor16] Marius Cordts, Mohamed Omran, Sebastian Ramos, et al. "The Cityscapes Dataset for Semantic Urban Scene Understanding". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2016 (cit. on pp. 84, 86).

[Kra16] Bartosz Krawczyk. "Learning from imbalanced data: open challenges and future directions". In: *Progress in Artificial Intelligence* 5.4 (Nov. 2016), pp. 221–232. ISSN: 2192-6360. DOI: 10.1007/s13748-016-0094-0 (cit. on p. 83).

[Luo16] Wenjie Luo et al. "Understanding the Effective Receptive Field in Deep Convolutional Neural Networks". In: *Advances in Neural Information Processing Systems 29*. Ed. by D. D. Lee et al. Curran Associates, Inc., 2016, pp. 4898–4906 (cit. on p. 90).

[Wan16] S. Wang, W. Liu, J. Wu, et al. "Training deep neural networks on imbalanced data sets". In: *2016 International Joint Conference on Neural Networks (IJCNN)*. July 2016, pp. 4368–4374. DOI: 10.1109/IJCNN.2016.7727770 (cit. on p. 83).

[Bul17] Samuel Rota Bulò, Gerhard Neuhold, and Peter Kontschieder. "Loss Max-Pooling for Semantic Image Segmentation". In: *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2017), pp. 7082–7091 (cit. on p. 83).

[Cho17] Francois Chollet. "Xception: Deep Learning With Depthwise Separable Convolutions". In: *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. July 2017 (cit. on p. 87).

[Poh17] Tobias Pohlen et al. "Full-Resolution Residual Networks for Semantic Segmentation in Street Scenes". In: *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. July 2017 (cit. on p. 89).

[Bud18] Mateusz Buda, Atsuto Maki, and Maciej A. Mazurowski. "A systematic study of the class imbalance problem in convolutional neural networks". In: *Neural Networks* 106 (2018), pp. 249–259. ISSN: 0893-6080. DOI: https://doi.org/10.1016/j.neunet.2018.07.011. URL: http://www.sciencedirect.com/science/article/pii/S0893608018302107 (cit. on p. 83).

[Che18] Liang-Chieh Chen et al. "Encoder-Decoder with Atrous Separable Convolution for Semantic Image Segmentation". In: *The European Conference on Computer Vision (ECCV)*. Sept. 2018 (cit. on p. 87).

[Lee18] Joffrey L. Leevy et al. "A survey on addressing high-class imbalance in big data". In: *Journal of Big Data* 5.1 (Nov. 2018), p. 42. ISSN: 2196-1115. DOI: 10.1186/s40537-018-0151-6. URL: https://doi.org/10.1186/s40537-018-0151-6 (cit. on p. 83).

[Rot18] Matthias Rottmann, Pascal Colling, Thomas-Paul Hack, et al. "Prediction Error Meta Classification in Semantic Segmentation: Detection via Aggregated Dispersion Measures of Softmax Probabilities". In: *CoRR* (2018). arXiv: 1811.00648. URL: http://arxiv.org/abs/1811.00648 (cit. on p. 91).

[San18] Mark Sandler et al. "MobileNetV2: Inverted Residuals and Linear Bottlenecks". In: *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2018 (cit. on p. 87).

[Joh19] Justin M. Johnson and Taghi M. Khoshgoftaar. "Survey on deep learning with class imbalance". In: *Journal of Big Data* 6.1 (Mar. 2019), p. 27. ISSN: 2196-1115. DOI: 10.1186/s40537-019-0192-5 (cit. on p. 83).

Prediction Error Meta Classification in Semantic Segmentation: Detection via Aggregated Dispersion Measures of Softmax Probabilities

Matthias Rottmann[1], Pascal Colling[1], Thomas Paul Hack[2],
Robin Chan[1], Fabian Hüger[3], Peter Schlicht[3] and Hanno Gottschalk[1]

[1]School of Mathematics and Natural Sciences, University of Wuppertal
[2]Faculty of Physics, University of Leipzig
[3]Architecture and AI Technologies, COI Automation, Volkswagen Group Innovation

Abstract. We present a method that "meta" classifies whether segments predicted by a semantic segmentation neural network intersect with the ground truth. For this purpose, we employ measures of dispersion for predicted pixel-wise class probability distributions, like classification entropy, that yield heat maps of the input scene's size. We aggregate these dispersion measures segment-wise and derive metrics that are well correlated with the segment-wise IoU of prediction and ground truth. This procedure yields an almost plug and play post-processing tool to rate the prediction quality of semantic segmentation networks on segment level. This is especially relevant for monitoring neural networks in online applications like automated driving or medical imaging where reliability is of utmost importance. In our tests, we use publicly available state-of-the-art networks trained on the Cityscapes dataset and the BraTS2017 dataset and analyze the predictive power of different metrics as well as different sets of metrics. To this end, we compute logistic LASSO regression fits for the task of classifying IoU $= 0$ vs. IoU > 0 per segment and obtain AUROC values of up to 91.55%. We complement these tests with linear regression fits to predict the segment-wise IoU and obtain prediction standard deviations of down to 0.130 as well as R^2 values of up to 84.15%. We show that these results clearly outperform standard approaches.

I Introduction

In recent years, deep learning has outperformed other classes of predictive models in many applications. In some of these, e.g. autonomous driving or diagnostics in medicine, the reliability of a prediction is of highest interest. In classification tasks, thresholding on the highest softmax probability or thresholding on the entropy of the classification distributions (softmax output) are commonly used approaches to detect false predictions of neural networks, see e.g. [Hen16; Lia17]. Metrics like classification entropy or the highest softmax probability are usually combined with model uncertainty (Monte-Carlo (MC) dropout inference) and sometimes input uncertainty, cf. [Gal16] and [Lia17], respectively. These approaches have proven to be practically efficient for detecting uncertainty. Such methods have also been transferred to semantic segmentation tasks. See also [Obe18] for further uncertainty metrics. The work presented in [Ken15] makes use of MC dropout to model the uncertainty of segmentation networks and also shows performance improvements in terms of segmentation accuracy. This approach was applied in other works to model the uncertainty and filter out predictions with low reliability, cf. e.g. [Kam16; Wic18]. In [Hua18] this line of research was further developed to detect spacial and temporal uncertainty in the semantic segmentation of videos.

In this work we establish an approach for efficiently meta classifying whether an inferred segment (representing a predicted object) of a semantic segmentation intersects with the ground truth or not. This task was first proposed for classification problems in [Hen16] and transferred to semantic

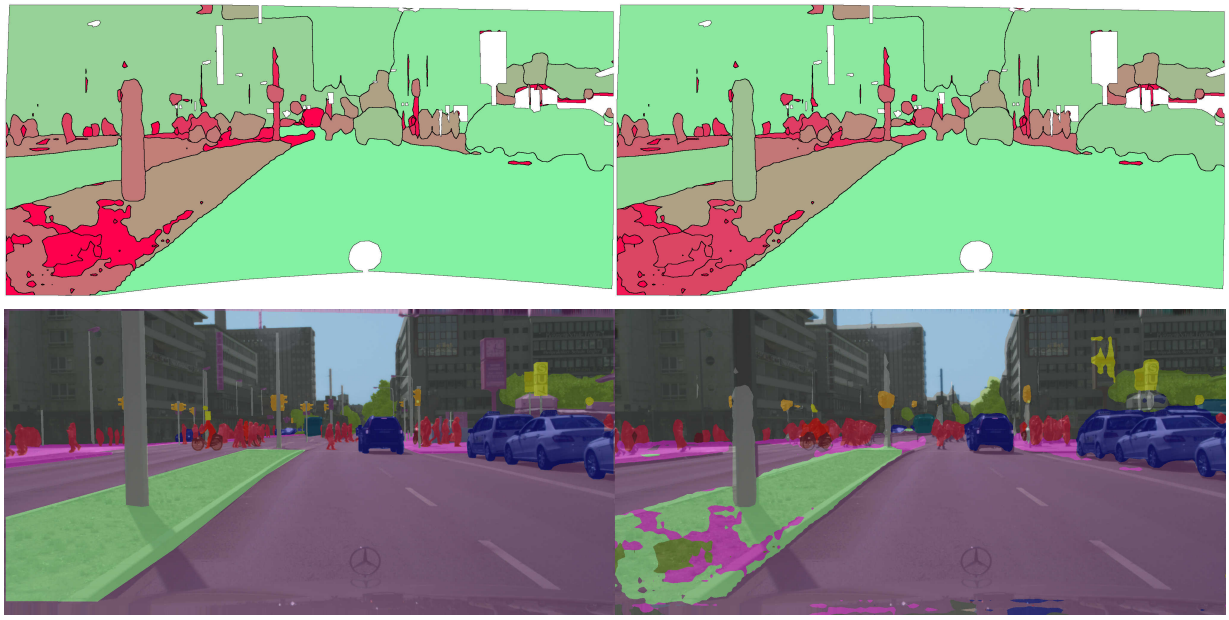


Figure 1: A demonstrations of the proposed method's performance of predicting the segment-wise IoU as a quality measure. The figure consists of ground truth (bottom left), segmentation predicted by DeepLabv3+ MobilenetV2 (bottom right), true IoU for each predicted segment (top left) and prediction of the IoU for each predicted segment obtained by our method (top right). In the top row, green color corresponds to high IoU values and red color to low ones. For the white regions there is no ground truth available. These regions are excluded from statistical evaluations.

segmentation [Hua16; DeV18], however not on segment level but for estimating the quality of a segmentation for an entire image that contains only a single object of interest. Segment level quality control for brain segmentation by means of metrics computed from MC dropout inferences is introduced in [Roy18] and another MC dropout based approach for object detection is presented in [Ozd17].

We term the task of classifying whether a predicted segment intersects with the ground truth or not as *meta classification*. This term has been used in the context of classical machine learning for learning the weights for each member of a committee of classifiers [Lin03]. In terms of deep learning we use this term as a shorthand to distinguish between a network's own classification and the classification whether a prediction is "true" or "false". In contrast to the work cited above, we aim at judging the statistical reliability of each segment inferred by the neural network. To the best of our knowledge, this is the first work that detects false positive segments (objects) in the semantic segmentation with multiple segments per image.

For meta classification we utilize dispersion measures, like entropy, applied to the softmax probabilities (the network's output) on pixel level yielding dispersion heat maps. We aggregate these heat maps over predicted segments alongside with other quantities derived from the network's prediction like the segment size and predicted class. From this, we construct per-segment metrics. A commonly used performance measure for the quality of a segmentation is the intersection over union (IoU a.k.a. Jaccard index [Jac12]) of prediction and ground truth. We use the constructed metrics as inputs to logistic regression models for *meta classifying*, whether an inferred segment's IoU vanishes or not, i.e., predicting $\mathrm{IoU} = 0$ or $\mathrm{IoU} > 0$. Also, we use linear regression models for predicting a segment's IoU directly, thus obtaining statements about the reliability of the network's prediction. We term this task *meta regression* and also introduce a modified version of the IoU (adjusted IoU) that is more suitable for this task. The same task is pursued in [Hua16; DeV18] for images containing only a single object,

instead of metrics they utilize additional CNNs. The approach presented in [Roy18] is inherently based on MC dropout while our approach is independent of this.

Our method only uses the softmax output of a semantic segmentation network and the corresponding ground truth. It is a pure post-processing tool that is trained once and offline, there is no additional training of segmentation networks involved. The segmentation network's output is not changed, only assessed. Our approach can be equipped with any heat map obtained from pixel-wise uncertainty measures. Thus, any work on uncertainty quantification for semantic segmentation that yields new improved dispersion heat maps can be seamlessly integrated and leverages our method. Hence, we also provide a framework to evaluate the information contained in pixel-wise uncertainty measures for semantic segmentation. To the best of our knowledge, this is the first work that estimates the quality of each predicted segment in a fully segmented image. A demonstration of its performance is given in Figure 1.

The work presented in this publication has initiated further research on resolution dependent uncertainty [Rot19] as well as time-dynamic quality estimates [Maa19].

In our tests we use two publicly available datasets: Cityscapes [Cor16] for the semantic segmentation of street scenes and BraTS2017 [Men15; Bak17] for brain tumor segmentation. For each of the two datasets we employ two state-of-the-art networks. We perform tests on validation sets and demonstrate that our segment-wise metrics are well correlated with the IoU; thus they are suitable for detecting false positives on segment level. For logistic regression fits we obtain values of up to 91.55% for the area under curve corresponding to the receiver operator characteristic curve (AUROC, see [Dav06]). Predicting the segment-wise IoU via linear regression we obtain prediction standard deviations of down to 0.130 and R^2 values of up to 84.15%.

II False positives and segment-wise quality measures for semantic segmentation

In order to perform meta classification and regression we first define the corresponding measures that can be deduced from prediction and ground truth.

A segmentation network with a softmax output layer can be seen as a statistical model that provides for each pixel z of the image a probability distribution $f_z(y|x,w)$ on the q class labels $y \in \mathcal{C} = \{y_1, \ldots, y_q\}$, given the weights w and the data x. The predicted class in y is then given by

$$\hat{y}_z(x,w) = \arg\max_{y \in \mathcal{C}} f_z(y|x,w)\,. \tag{1}$$

For a given image x we denote by $\hat{\mathcal{K}}_x$ the set of connected components (segments) in the predicted segmentation $\hat{\mathcal{S}}_x = \{\hat{y}_z(x,w) | z \in x\}$ (omitting the dependence on the weights w). Analogously we denote by $\mathcal{K}_x$ the set of connected components in the ground truth $\mathcal{S}_x$. For each $k \in \hat{\mathcal{K}}_x$, the intersection over union IoU is defined as follows: Let $\mathcal{K}_x|_k$ be the set of all $k' \in \mathcal{K}_x$ that have non-trivial intersection with k and whose class label are equal to the predicted class of k, then

$$\mathrm{IoU}(k) = \frac{|k \cap K'|}{|k \cup K'|}\,, \qquad K' = \bigcup_{k' \in \mathcal{K}_x|_k} k'\,. \tag{2}$$

High values of $\mathrm{IoU}(k)$ correspond to good predictions, low values to bad predictions. The task of meta classification can now be defined as predicting for each $k \in \hat{\mathcal{K}}_x$, whether $\mathrm{IoU}(k) = 0$ or $\mathrm{IoU}(k) > 0$. Meta regression amounts to predicting $\mathrm{IoU}(k)$ quantitatively. For the latter task, however, in specific scenarios the $\mathrm{IoU}(k)$ can have low values while the prediction looks fine. This is the case, when a ground truth segment is covered by more than one predicted segment. In this case the predicted segments can have a low $\mathrm{IoU}(k)$ although together they provide a good prediction. To this end

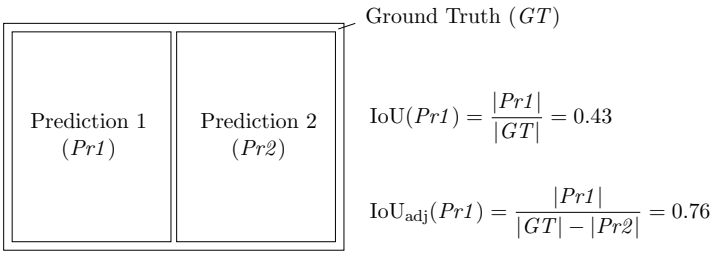


Figure 2: Illustration of different behaviors of IoU and $\mathrm{IoU}_{\mathrm{adj}}$. Two disjoint predictions (of the same size and assumed to be assigned to the same class) are enclosed by their corresponding ground truth component. Each predicted component achieves an IoU of 43%. However, this value seems rather low as the ground truth is well covered. Thus, we modify the quality measure for each prediction by excluding that part of the ground truth covered by other predicted components (of the same class), yielding an $\mathrm{IoU}_{\mathrm{adj}}$ of 76%.

we introduce as a segment-wise quality measure the *adjusted intersection over union* $\mathrm{IoU}_{\mathrm{adj}}$: Let $Q = \{q \in \hat{\mathcal{K}}_x \setminus \{k\} : q \cap K' \neq \emptyset\}$, then

$$\mathrm{IoU}_{\mathrm{adj}}(k) = \frac{|k \cap K'|}{|k \cup (K' \setminus Q)|}\,. \tag{3}$$

The $\mathrm{IoU}_{\mathrm{adj}}(k)$ does not punish different predicted segments that share a common bigger ground truth segment, for an illustration of this see Figure 2. Clearly, we have $\mathrm{IoU}_{\mathrm{adj}}(k) = \mathrm{IoU}(k) = 1$ if and only if the predicted segment k and the ground truth K' match for each pixel, $\mathrm{IoU}_{\mathrm{adj}} = \mathrm{IoU} = |k \cap K'| = 0$ when ground truth and predicted segment do not overlap, i.e., $k \cap K' = \emptyset$, and it holds $\mathrm{IoU}_{\mathrm{adj}} \geq \mathrm{IoU}$. Thus, the meta classification task is invariant under interchanging IoU and $\mathrm{IoU}_{\mathrm{adj}}$. However, the meta regression task for directly predicting IoU and $\mathrm{IoU}_{\mathrm{adj}}$, respectively, is not invariant. In our experiments we found that the $\mathrm{IoU}_{\mathrm{adj}}(k)$ is indeed more suitable for the task of meta regression which is manifested by higher performance in terms of R^2 values. For this discussion we refer to Section VI.

III Pixel-wise dispersion metrics and aggregation over segments

In this section we introduce the metrics that are used as input quantities for performing meta classification and regression. They are based on dispersion measures as well as different size measures that are aggregated for each predicted segment.

Dispersion or concentration measures quantify the degree of randomness in $f_z(y|x,w)$. Here, we consider two of those measures: *entropy* E_z (also known as *Shannon information* [Sha48]) and *difference in probability* D_z, i.e., the difference between the two largest softmax values:

$$E_z(x,w) = -\frac{1}{\log(q)} \sum_{y \in \mathcal{C}} f_z(y|x,w) \log f_z(y|x,w)\,, \tag{4}$$

$$D_z(x,w) = 1 - f_z(\hat{y}_z(x,w)|x,w) + \max_{y \in \mathcal{C} \setminus \{\hat{y}_z(x,w)\}} f_z(y|x,w)\,. \tag{5}$$

For better comparison, both quantities have been written as dispersion measures and been normalized to the interval $[0,1]$: One has $E_z = D_z = 1$ for the equiprobability distribution $f_z(y|x,w) = \frac{1}{q}$, $y \in \mathcal{C}$, and $E_z = D_z = 0$ on the deterministic probability distribution ($f_z(y|x,w) = 1$ for one class and 0 otherwise). For further discussion on dispersion measures, see [Cow11]. The most direct method of uncertainty quantification on an image is the heat mapping of a dispersion measure as in Figure 3.

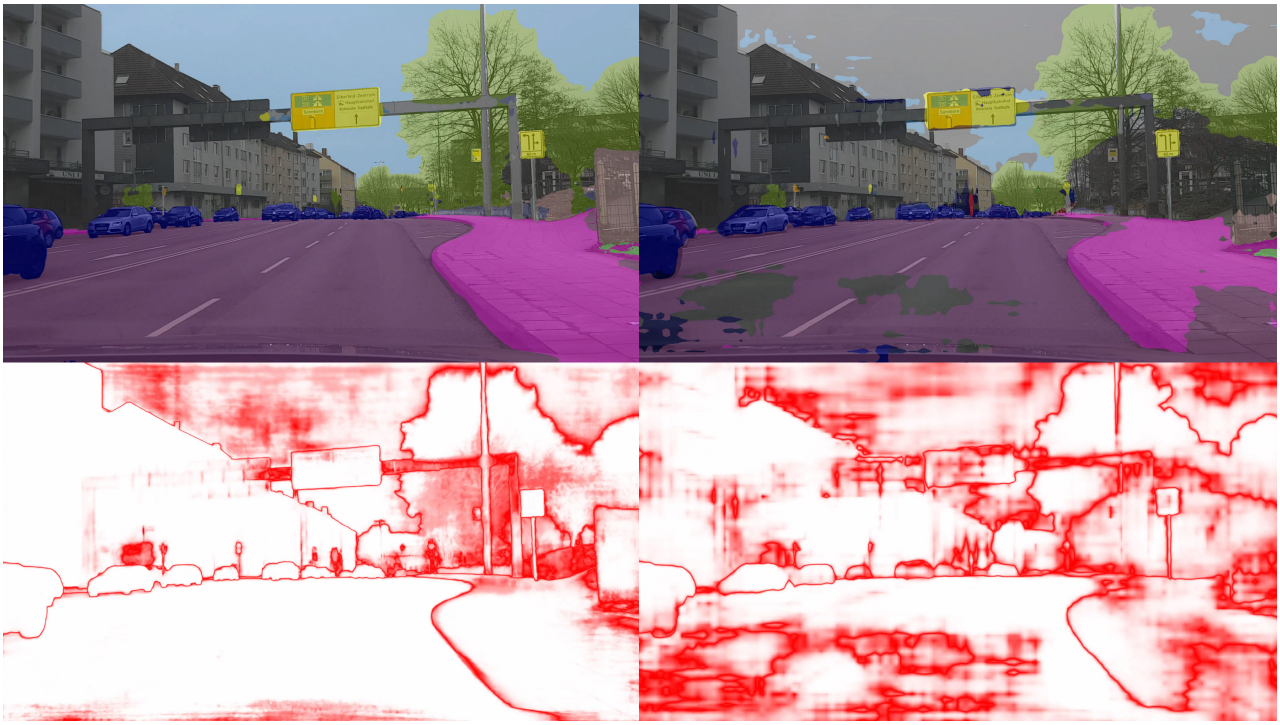


Figure 3: Segmentation example (top line) and heat map D_z (bottom line) for Xception65 (left column) and MobilenetV2 (right column). Original image is not part of the Cityscapes dataset.

We now aggregate these measures over predicted segments. Therefore, for each $k \in \hat{\mathcal{K}}_x$, we define the following quantities:

- the interior $k_{in} \subset k$ where a pixel z is an element of k_{in} if all eight neighbouring pixels are an element of k
- the boundary $k_{bd} = k \setminus k_{in}$
- the pixel sizes $S = |k|$, $S_{in} = |k_{in}|$, $S_{bd} = |k_{bd}|$
- the mean entropies $\bar{E}$, $\bar{E}_{in}$, $\bar{E}_{bd}$ defined as

$$\bar{E}_\sharp(k) = \frac{1}{S_\sharp} \sum_{z \in k_\sharp} E_z(x)\,, \qquad \sharp \in \{\_, in, bd\}$$

- the mean distances $\bar{D}$, $\bar{D}_{in}$, $\bar{D}_{bd}$ defined in analogy to the mean entropies
- the relative sizes $\tilde{S} = S/S_{bd}$, $\tilde{S}_{in} = S_{in}/S_{bd}$
- the relative mean entropies $\tilde{\bar{E}} = \bar{E}\tilde{S}$, $\tilde{\bar{E}}_{in} = \bar{E}_{in}\tilde{S}_{in}$, and
- the relative mean distances $\tilde{\bar{D}} = \bar{D}\tilde{S}$, $\tilde{\bar{D}}_{in} = \bar{D}_{in}\tilde{S}_{in}$.

Typically, E_z and D_z are large for $z \in k_{bd}$. This motivates the separate treatment of interior and boundary measures. With the exception of IoU and $\mathrm{IoU}_{\mathrm{adj}}$, all scalar quantities defined above can be computed without the knowledge of the ground truth. Our aim is to analyze to which extent they are able to predict $\mathrm{IoU}_{\mathrm{adj}}$.

	XC	MN		XC	MN		XC	MN
$\bar{E}$	-0.70139	-0.70162	$\bar{D}$	-0.85211	-0.84858	S	0.30442	0.47978
$\bar{E}_{bd}$	-0.44065	-0.41845	$\bar{D}_{bd}$	-0.60308	-0.52163	S_{bd}	0.44625	0.62713
$\bar{E}_{in}$	-0.71623	-0.69884	$\bar{D}_{in}$	-0.85458	-0.82171	S_{in}	0.30201	0.47708
$\tilde{\bar{E}}$	0.31219	0.36261	$\tilde{\bar{D}}$	0.22797	0.30245	$\tilde{S}$	0.50758	0.71071
$\tilde{\bar{E}}_{in}$	0.39195	0.42806	$\tilde{\bar{D}}_{in}$	0.29279	0.35131	$\tilde{S}_{in}$	0.50758	0.71071

Table 1: Correlation coefficients ρ of aggregated dispersion metrics with respect to $\mathrm{IoU}_{\mathrm{adj}}$. The shown results are computed on the Cityscapes validation set, XC: DeepLabv3+Xception65 and MN: DeepLabv3+MobilenetV2.

IV Numerical Experiments: Street Scenes

We investigate the properties of the metrics defined in the previous section for the example of a semantic segmentation of street scenes. In order to investigate the predictive power of the metrics, we first compute the Pearson correlation $\rho \in [-1, 1]$ between each feature and $\mathrm{IoU}_{\mathrm{adj}}$. We report the results of this analysis in Table 1 and Figure 4.

In our experiments we consider the DeepLabv3+ model [Che18] for which we use a reference implementation in Tensorflow [Mar15] as well as weights pretrained on the Cityscapes dataset [Cor16] that are available on GitHub. The DeepLabv3+ implementation and weights are available for two network backbones: Xception65, which is a modified version of Xception [Cho17] and is a powerful structure intended for server-side deployment, and MobilenetV2 [San18], a fast structure designed for mobile devices. Each of these implementations have parameters tuning the segmentation accuracy. We choose the following best (for Xception65) and worst (for MobilenetV2) parameters in order to perform our analysis on two very distinct networks. Note, that the parameter set for the Xception65 setting also includes the evaluation of the input on multiple scales (averaging the results) which increases the accuracy and also leverages classification uncertainty. We refer to [Che18] for a detailed explanation of the chosen parameters.

For both networks, we consider the output probabilities and predictions on the Cityscapes validation set, which consists of 500 street scene images at a resolution of 2048×1024. We compute the 15 constructed metrics as well as $\mathrm{IoU}_{\mathrm{adj}}$ for each segment in the segmentations of the images. Note, that in all computations, we only consider connected components with non-empty interior.

- DeepLabv3+Xception65: output stride 8, decoder output stride 4, evaluation on input scales $0.75, 1.00, 1.25$ – mIoU = 79.72% on the Cityscapes validation set
- DeepLabv3+MobilenetV2: output stride 16, evaluation on input scale 1.00 – mIoU = 61.85% on the Cityscapes validation set

For both networks $\mathrm{IoU}_{\mathrm{adj}}$ shows a strong correlation with the mean distances $\bar{D}$ and $\bar{D}_{in}$ as well as with the mean entropies $\bar{E}$ and $\bar{E}_{in}$. On the other hand, the relative counterparts are less correlated with $\mathrm{IoU}_{\mathrm{adj}}$. The relative segment size $\tilde{S}$ for the DeepLabv3+MobilenetV2 network shows a clear correlation whereas this is not the case for the more powerful DeepLabv3+Xception65 network.

In order to find more indicative measures, we now investigate the predictive power of the metrics when they are combined. For the Xception65 net, we obtain 45,194 segments with non-empty interior of which 11,331 have $\mathrm{IoU}_{\mathrm{adj}} = 0$. For the weaker MobilenetV2 this ratio is 42,261/17,671. We would first like to detect segments with $\mathrm{IoU}_{\mathrm{adj}} = 0$, i.e., learn the meta classification task of identifying false positive segments based on our 15 metrics and the segment-wise averaged probability distribution

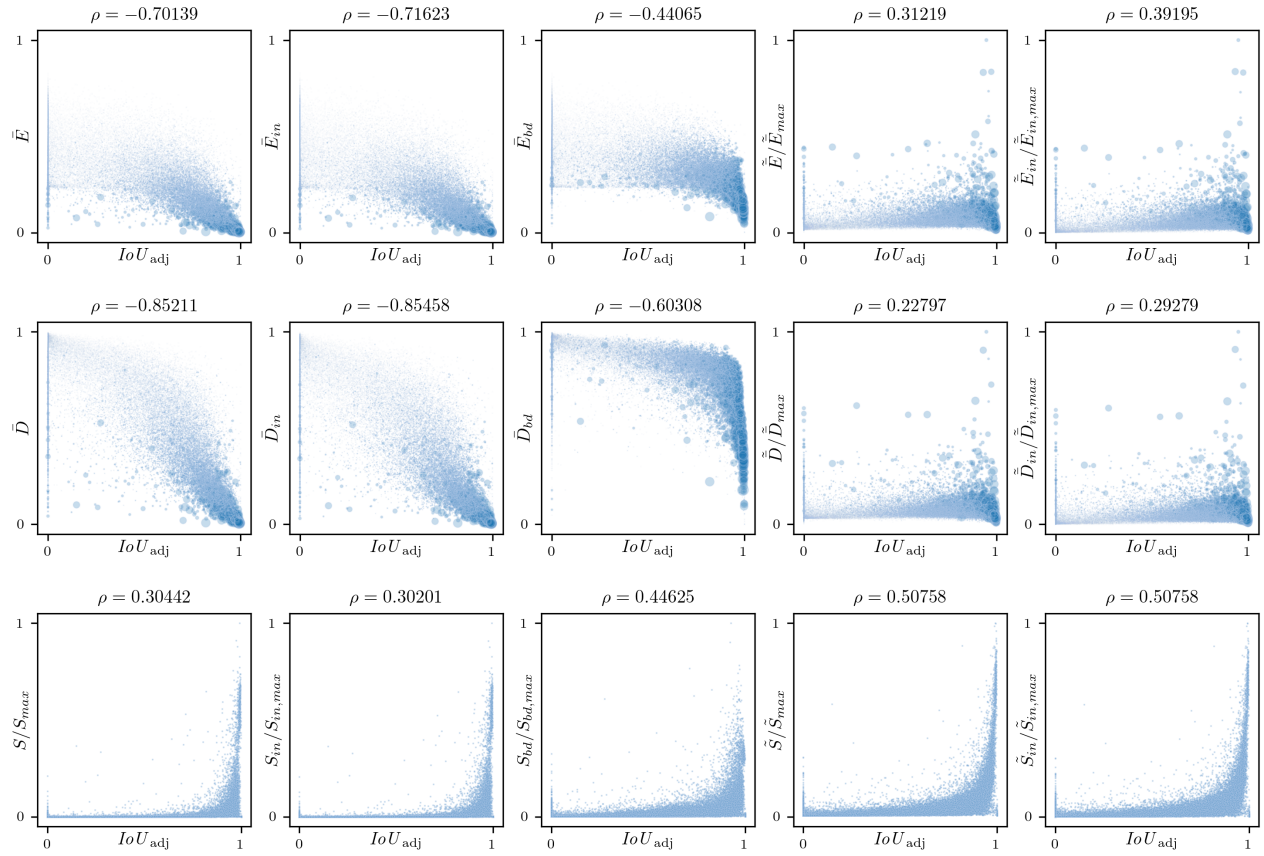


Figure 4: Correlation scatter plots of $\mathrm{IoU}_{\mathrm{adj}}$ and rescaled features for the DeepLabv3+Xception65 network. Dot sizes in the first two rows are proportional to S.

vectors. We term these (standardized) inputs x_k for a segment k. Further, let $y_k = ceil(\mathrm{IoU}_{\mathrm{adj}}) = \{0 \text{ if } \mathrm{IoU}_{\mathrm{adj}} = 0, 1 \text{ if } \mathrm{IoU}_{\mathrm{adj}} > 0\}$.

The least absolute shrinkage and selection operator (LASSO, [Tib96]) is a popular tool for investigating the predictive power of different combinations of input variables. We compute a series of LASSO fits, i.e., ℓ_1-penalized logistic regression fits

$$\min_w \sum_i [-y_i \log(\tau(w^T x_i)) - (1 - y_i)(1 - \log(\tau(w^T x_i))) + \lambda \|w\|_1] \ , \tag{6}$$

for different regularization parameters λ and standardized inputs (zero mean and unit standard deviation). Here, $\tau(\cdot)$ is the logistic function. Results for the Xception65 net are shown in Figure 5.

The top left and top right panels show, in which order the weight coefficients w for each metric/predicted class become active. At the same time the bottom left and bottom right panels show, which weight coefficient causes which amount of increase in predictive performance in terms of meta classification rate and AUROC, respectively. The AUROC is obtained by varying the decision threshold of the logistic regression output for deciding whether $\mathrm{IoU} = 0$ or $\mathrm{IoU} > 0$.

The first non-zero coefficient activates the $\bar{D}_{in}$ metric, which elevates the predictive power above our reference benchmark of choice, the mean entropy per component $\bar{E}$. Another significant gain is achieved when $\bar{D}_{bd}$ and the predicted classes come into play. In the numerical experiments we randomly choose 10 50/50 training/validation data splits and average the results. Additionally, the bottom line

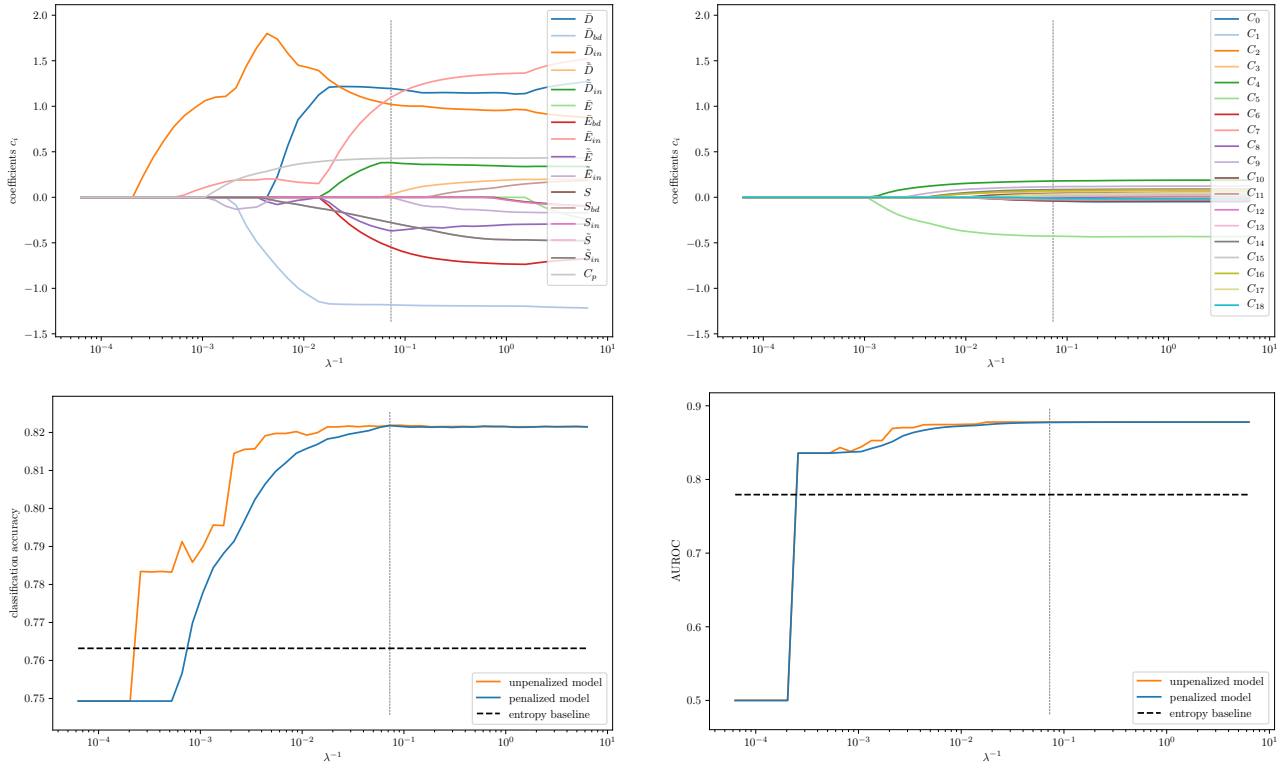


Figure 5: Results for the meta classification task $\text{IoU}_{\text{adj}} = 0, > 0$ for predictions obtained from the Xception65 net. (Top left): the weights coefficients for the 15 metrics computed with LASSO fits as function of λ^{-1}, C_p denotes the maximum of the absolute values of all weight coefficients for predicted classes. (Top right): like top left but showing coefficients for the 18 predicted classes. (Bottom left): meta classification rates for $\text{IoU}_{\text{adj}} = 0, > 0$. The blue line are the LASSO fits for different λ values, the orange line shows the performance of regular logistic regression fits ($\lambda = 0$) where the input metrics are only those that have non-zero coefficients in the LASSO fit for the current λ. (Bottom right) same as bottom left, but for AUROC. The vertical dashed lines indicate the λ value for which we obtained the best validation accuracy.

of Figure 5 shows that there is almost no performance loss when only incorporating some of the metrics proposed by the LASSO trajectory. For both networks the classification accuracy corresponds to a logistic regression trained with unbalanced meta classes $\text{IoU}_{\text{adj}} = 0$ and $\text{IoU}_{\text{adj}} > 0$, i.e., we did not adjust the class weights. On average (over the 10 training/validation splits) 6851 components with vanishing IoU_{adj} are detected for Xception65 while 4480 remain undetected, for MobilenetV2 this ratio is 14976/2695. These ratios can be adjusted by varying the probability thresholds for deciding between $\text{IoU}_{\text{adj}} = 0$ and $\text{IoU}_{\text{adj}} > 0$. For this reason we state results in terms of AUROC which is threshold independent.

We compare our results with two different baselines. The naive baseline is given by random guessing (randomly assigning a probability to each segment k and then thresholding on it). The best meta classification accuracy is achieved for the threshold being either 0 or 1. For $I_0 := |\{k : \text{IoU}_{\text{adj}} = 0\}|$ and $I_1 := |\{k : \text{IoU}_{\text{adj}} > 0\}|$ the naive baseline accuracy is then given by $\frac{\max(I_0, I_1)}{I_0 + I_1}$. The corresponding AUROC value is 50%. Another baseline is to equip our approach only with a single metric. For this purpose we choose the entropy as it is commonly used for uncertainty quantification.

The meta classification results averaged over 10 runs with different training/validation splits are reported in Table 2. We obtain a meta classification validation accuracy of up to 81.91%(±0.13%)

	Xception65		MobilenetV2	
Cityscapes	training	validation	training	validation
	Classification $\text{IoU}_{\text{adj}} = 0, > 0$			
ACC, penalized	81.88%(±0.13%)	81.91%(±0.13%)	78.87%(±0.13%)	78.93%(±0.17%)
ACC, unpenalized	81.91%(±0.12%)	81.92%(±0.12%)	78.84%(±0.14%)	78.93%(±0.18%)
ACC, entropy only	76.36%(±0.17%)	76.32%(±0.17%)	68.33%(±0.27%)	68.57%(±0.25%)
ACC, naive baseline	74.93%		58.19%	
AUROC, penalized	87.71%(±0.14%)	87.71%(±0.15%)	86.74%(±0.18%)	86.77%(±0.17%)
AUROC, unpenalized	87.72%(±0.14%)	87.72%(±0.15%)	86.74%(±0.18%)	86.76%(±0.18%)
AUROC, entropy only	77.81%(±0.16%)	77.94%(±0.15%)	76.63%(±0.24%)	76.74%(±0.24%)
	Regression IoU_{adj}			
σ, all metrics	0.181(±0.001)	0.182(±0.001)	0.130(±0.001)	0.130(±0.001)
σ, entropy only	0.258(±0.001)	0.259(±0.001)	0.215(±0.001)	0.215(±0.001)
R^2, all metrics	75.06%(±0.22%)	74.97%(±0.22%)	81.50%(±0.23%)	81.48%(±0.23%)
R^2, entropy only	49.37%(±0.32%)	49.02%(±0.32%)	49.32%(±0.31%)	49.12%(±0.32%)

Table 2: Summarized results for the meta classification and regression task for Cityscapes. The results are averaged over 10 runs. The numbers in brackets denote standard deviations of the computed mean values.

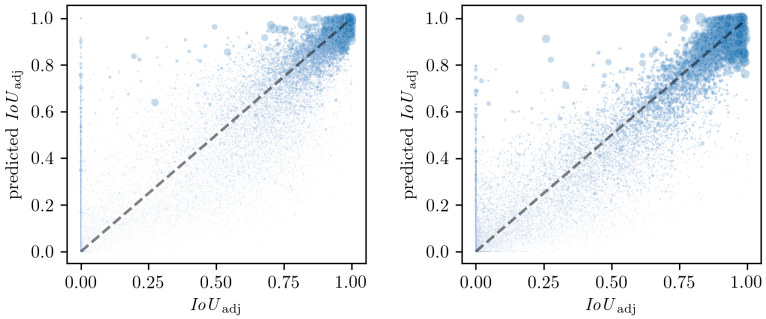
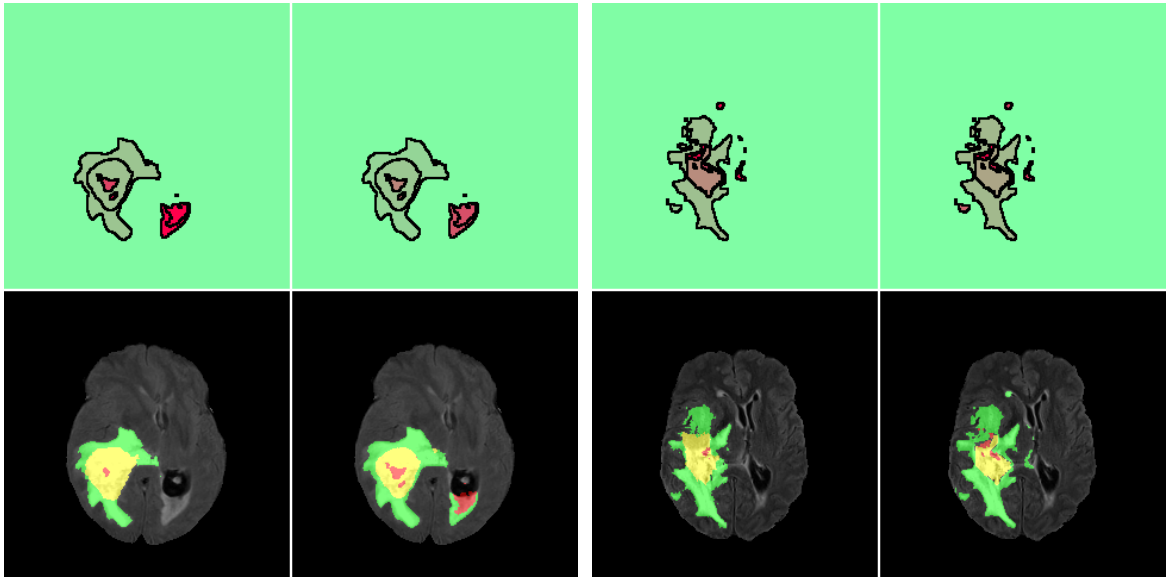


Figure 6: Relationship between IoU_{adj} and predicted IoU_{adj} for all connected components predicted by Xception65 (left) and MobilenetV2 (right). Dot sizes are proportional to connected component's size S.

and an AUROC of up to 87.71%(±0.15%) for Xception65. And also for the weaker MobilenetV2 we obtain 78.93%(±0.17%) classification accuracy and 86.77%(±0.17%) AUROC. The numbers in brackets denote standard deviations of the performance scores. The classification accuracy and AUROC results are slightly biased towards the validation results as they correspond to the particular λ value that maximizes the validation accuracy. Both baselines (random guessing and entropy) are clearly outperformed and indicate that the computed set of dispersion measures contains rich information for detecting unreliably predicted segments.

Ultimately, we want to perform meta regression, i.e., predict IoU_{adj} values for all connected components and thus model a quality measure. We now resign from regularization and use a linear regression model to predict the IoU_{adj}. Figure 6 depicts the quality of a single linear regression fit for each of the two segmentation networks.

For Xception65 we obtain an R^2 value of 74.93%(±0.22%) and for MobilenetV2 81.48%(±0.23%). Averaged results over 10 runs including standard deviations σ are summarized in Table 2. In both cases, our presented approach clearly outperforms the entropy. The linear regression models do not overfit the data and note-worthily we obtain prediction standard deviations of down to 0.130 and almost no standard deviation for the averages.



(a) 2D U-Net by Kermi et al. (b) 3D Net by Myronenko et al.

Figure 7: Two demonstrations (left and right four panels, analogously to Figure 1) of our method's performance of predicting IoU_{adj} on BraTS2017. In the bottom row, the whole tumor (WT) includes all colored segments (union of green, yellow & red), the tumor core (TC) the yellow joined with the red colored segments and the enhancing tumor (ET) only the yellow colored segments. In the top row, green color corresponds to high IoU_{adj} values and red color to low ones. In both examples, predicted quality and true quality look very similar.

V Numerical Experiments: Brain Tumor Segmentation

The method we propose only uses dispersion heat maps and softmax probabilities as inputs. Any additional heat map increases the performance as long as there is no overfitting. Thus, we expect our approach to generalize across different datasets even from different domains. To demonstrate this, we perform additional tests with the brain tumor segmentation dataset BraTS2017 [Men15; Bak17] and two different networks, i.e., a simple 2D network and a more complex 3D network. Compared to the segmentation of street scenes, brain tumor segmentation involves way fewer classes. The background class is usually dominant. In BraTS2017, roughly 98% of all pixels are background, the remaining classes comprise necrotic/non-enhancing tumor, peritumoral edema and enhancing tumor. For benchmarks of predictive methods, these labels are combined into three nested classes: whole tumor (WT), tumor core (TC) and enhancing tumor (ET) (see Figure 7). The most commonly used evaluation metric is the so-called *Dice-Coefficient* [Zou04] that is defined as

$$\text{Dice} := 2\,TP/(2\,TP + FP + FN) \tag{7}$$

where TP, FP and FN denote all true positive, false positive and false negative pixels, respectively, for a chosen class.

The BraTS data is available as magnetic resonance imaging (MRI) brain scans from three viewing angles and with four modalities of higher grade gliomas (HGG) and lower grade gliomas (LGG). For training and validation, we combine HGG and LGG images and randomly split the data 80/20. We train the networks from scratch with the different scan modalities stacked as the network's input channels. Once this is done, we perform tests analogously to the previous section. The performance

Metric	Dice Coefficient				Intersection over Union	
	WT	TC	ET	*mDice*	mIoU nested	mIoU single
2D U-Net by Kermi et al.	88.09%	77.38%	78.89%	81.45%	68.99%	67.14%
3D Net by Myronenko et al.	88.83%	81.07%	79.63%	83.18%	71.40%	69.64%

Table 3: BraTS2017 performance scores on validation data split for the two networks used in the numerical experiments. The nested classes whole tumor (WT), tumor core (TC) and enhancing tumor (ET) are evaluated with the Dice coefficient. For comparison purposes, the mean Dice score is reported as well as mean Intersection over Union for nested classes and single classes (background, non-enhancing tumor, peritumoral edema and enhancing tumor).

	2D U-Net by Kermi et al.		3D Net by Myronenko et al.	
BraTS2017	training	validation	training	validation
	Classification $\mathrm{IoU}_{\mathrm{adj}} = 0, > 0$			
ACC, penalized	89.30%(±0.18%)	89.39%(±0.17%)	88.40%(±0.27%)	88.42%(±0.27%)
ACC, unpenalized	89.29%(±0.19%)	89.40%(±0.18%)	88.38%(±0.27%)	88.40%(±0.28%)
ACC, entropy only	87.96%(±0.12%)	87.96%(±0.12%)	86.69%(±0.20%)	86.69%(±0.20%)
ACC, naive baseline	88.30%		86.35%	
AUROC, penalized	91.84%(±0.25%)	91.93%(±0.24%)	91.51%(±0.22%)	91.55%(±0.22%)
AUROC, unpenalized	91.83%(±0.25%)	91.93%(±0.24%)	91.49%(±0.22%)	91.53%(±0.22%)
AUROC, entropy only	86.68%(±0.25%)	86.73%(±0.25%)	86.59%(±0.28%)	86.74%(±0.28%)
	Regression $\mathrm{IoU}_{\mathrm{adj}}$			
σ, all metrics	0.148(±0.001)	0.149(±0.001)	0.171(±0.001)	0.171(±0.001)
σ, entropy only	0.178(±0.001)	0.178(±0.001)	0.198(±0.001)	0.197(±0.001)
R^2, all metrics	84.22%(±0.21%)	84.15%(±0.21%)	79.53%(±0.28%)	79.64%(±0.28%)
R^2, entropy only	77.18%(±0.18%)	77.30%(±0.17%)	72.63%(±0.27%)	72.91%(±0.27%)

Table 4: Summarized results for the meta classification and regression task for BraTS2017. The results are averaged over 10 runs. The numbers in brackets denote standard deviations of the computed mean values.

of the two networks being used for our validation split are reported in Table 3, the results for meta classification and regression are summarized in Table 4.

For the first test we use the network by Kermi et al. [Ker18]. It is based on the U-Net [Ron15] which is originally well known for its performance on biomedical image segmentation. We train the network on randomly sampled 2D patches from axial (top view) slices of the brain scans. The results of our prediction rating methods are computed for 22,242 non-empty segments of which 2,603 have $\mathrm{IoU}_{\mathrm{adj}} = 0$. Indeed, we obtain higher accuracy values compared to Cityscapes, however the gain over the single metric baseline is not as big. This is primarily due to a strong correlation between E and $\mathrm{IoU}_{\mathrm{adj}}$ (-0.87794). In this case, the gain over the naive baseline is marginal. This may be misleading to the disadvantage of our method as the high naive accuracy is caused by the strong sample imbalance of the meta classes. The corresponding AUROC value of 91.93% shows that our method meta classifies with significantly higher confidence when incorporating all metrics. Regarding the R^2 value of our regression model for predicting $\mathrm{IoU}_{\mathrm{adj}}$, we observe an increase from 77.30%(±0.17%) to 84.15%(±0.21%) when incorporating all metrics instead of only the entropy.

Next, we compare the U-Net's performance to the state-of-the-art network by Myronenko et al. [Myr19]. One main difference is that the latter network considers the MRI scans' 3D contextual information by processing multiple contiguous 2D slices at once, i.e., we train the network on randomly sampled 3D patches. As a consequence, the model is more complex and the number of trainable parameters is noticeably increased (10.1M vs. 17.3M). We perform the evaluation in the same 2D

	Xception65		MobilenetV2	
	training	validation	training	validation
	Regression $\mathrm{IoU}_{\mathrm{adj}}$			
σ, all metrics	0.181(±0.001)	0.182(±0.001)	0.130(±0.001)	0.130(±0.001)
σ, entropy only	0.258(±0.001)	0.259(±0.001)	0.215(±0.001)	0.215(±0.001)
R^2, all metrics	75.06%(±0.22%)	74.97%(±0.22%)	81.50%(±0.23%)	81.48%(±0.23%)
R^2, entropy only	49.37%(±0.32%)	49.02%(±0.32%)	49.32%(±0.31%)	49.12%(±0.32%)
	Regression IoU			
σ, all metrics	0.192(±0.001)	0.192(±0.001)	0.135(±0.001)	0.135(±0.001)
σ, entropy only	0.267(±0.001)	0.268(±0.001)	0.217(±0.001)	0.217(±0.001)
R^2, all metrics	72.90%(±0.21%)	72.77%(±0.21%)	79.63%(±0.27%)	79.58%(±0.27%)
R^2, entropy only	47.43%(±0.28%)	47.07%(±0.28%)	47.73%(±0.37%)	47.50%(±0.38%)

Table 5: Comparison of regression results for segment-wise fitting $\mathrm{IoU}_{\mathrm{adj}}$ and IoU, averaged over 10 runs. The numbers in brackets denote standard deviations of the computed mean values.

slice-wise manner as for the U-Net. The results are now computed for 24,397 non-empty segments of which 3331 have $\mathrm{IoU}_{\mathrm{adj}} = 0$. Again, we observe a strong correlation between E and $\mathrm{IoU}_{\mathrm{adj}}$ of -0.84294 which results in a nearly identical gain in terms of percent points over the single metric baseline as for the U-Net. Also with respect to the R^2 value of our regression model, the gain is again around 7 percent points, whereas the absolute value with 79.64%(±0.28%) for all metrics is not as high as for the U-Net.

VI Comparison of IoU and $\mathrm{IoU}_{\mathrm{adj}}$

Recall from Section II, the $\mathrm{IoU}_{\mathrm{adj}}(k)$ does not punish differently predicted segments that share a common bigger ground truth segment, whereas the standard IoU measure does. As the meta regression task is not invariant under interchanging IoU and $\mathrm{IoU}_{\mathrm{adj}}$, we compare performance differences when using either of these measures. Carrying out the regression tests from Section IV for the $\mathrm{IoU}_{\mathrm{adj}}$ with the IoU as well, we observe that the regression fit for the $\mathrm{IoU}_{\mathrm{adj}}$ achieves R^2 values that are roughly 2% higher than those for the IoU, cf. Table 5. Usually, for performance measures in semantic segmentation, the IoU is computed for a chosen class over the whole image. This means that each pixel of the union of prediction and ground truth is only counted once in the denominator of the image-wise IoU. On the other hand, a ground truth pixel may contribute to segment-wise IoUs of several segments, a practical example is given in Figure 8. In this sense, in the context of semantic segmentation, the adjusted $\mathrm{IoU}_{\mathrm{adj}}$ is in spirit closer to the regular image-wise IoU.

VII Conclusion and Outlook

We have shown statistically that per-segment metrics derived from entropy, probability difference, segment size and the predicted class clearly contain information about the reliability of the segments and constructed an approach for detecting unreliable segments in the network's prediction. In our tests with publicly available networks and datasets, Cityscapes and BraTS2017, the computed logistic LASSO fits for meta classification task $\mathrm{IoU}_{\mathrm{adj}} = 0$ vs. $\mathrm{IoU}_{\mathrm{adj}} > 0$ achieve AUROC values of up to 91.55%. When predicting the $\mathrm{IoU}_{\mathrm{adj}}$ with a linear regression fit we obtain a prediction standard deviation of down to 0.130, as well as R^2 values of up to 84.15%. These results could be further improved when incorporating model uncertainty in heat map generation. We believe that using MC dropout will further improve these results, just like the development of ever more accurate networks. We plan to use our method for detecting labeling errors, for label acquisition in active learning and we

(a) (b) (c)

Figure 8: Illustration of the different behaviors of IoU and IoU_{adj}. We have (a): IoU per predicted segment, (b) left: ground truth, right: detail views for the crucial area of the predicted segmentation (top) and the corresponding ground truth (bottom) and (c): IoU_{adj} per segment. Green stands for high IoU and IoU_{adj} values, red for low ones, respectively. The top right panel in (b) shows that the prediction for the class 'nature' is decoupled into two components by the traffic light's prediction. The IoU rates this small part on the left from the traffic light very badly even though the prediction is absolutely fine. The adjusted IoU_{adj} circumvents this type of problems.

plan to investigate further metrics that may leverage detection accuracy. Apart from that, detection mechanisms built on the softmax input and even earlier layers could be thought of. Furthermore, when reducing the number of false negatives for a chosen class by adjusting softmax thresholds, our method can be used to keep the production of new false positives under control. The source code of our method is publicly available at https://github.com/mrottmann/MetaSeg.

Acknowledgment. This work is in part funded by Volkswagen Group Innovation.

References

[Jac12] Paul Jaccard. "The Distribution of the Flora in the Alpine Zone". In: *New Phytologist* 11.2 (Feb. 1912), pp. 37–50. URL: http://www.jstor.org/stable/2427226?seq=3 (cit. on p. 96).

[Sha48] C. E. Shannon. "A Mathematical Theory of Communication". In: *The Bell System Technical Journal* 27 (1948), pp. 379–423, 623–656. URL: http://math.harvard.edu/~ctm/home/text/others/shannon/entropy/entropy.pdf (cit. on p. 98).

[Tib96] Robert Tibshirani. "Regression Shrinkage and Selection via the Lasso". In: *Journal of the Royal Statistical Society: Series B* 58 (1996), pp. 267–288. URL: https://www.bibsonomy.org/bibtex/290e648276aa6cd3c601e7c0a54366233/dieudonnew (cit. on p. 101).

[Lin03] WH. Lin and A. Hauptmann. "Meta-classification: Combining Multimodal Classifiers". In: *Mining Multimedia and Complex Data. PAKDD 2002. Lecture Notes in Computer Science* 2797 (2003) (cit. on p. 96).

[Zou04] Kelly H. Zou et al. "Statistical validation of image segmentation quality based on a spatial overlap index". In: *Academic radiology* 11.2 (Feb. 2004), pp. 178–189. ISSN: 1076-6332. DOI: 10.1016/S1076-6332(03)00671-8. URL: https://www.ncbi.nlm.nih.gov/pubmed/14974593 (cit. on p. 104).

[Dav06] Jesse Davis and Mark Goadrich. "The relationship between Precision-Recall and ROC curves". In: *Machine Learning, Proceedings of the Twenty-Third International Conference (ICML 2006), Pittsburgh, Pennsylvania, USA, June 25-29, 2006*. 2006, pp. 233–240. DOI: 10.1145/1143844.1143874. URL: http://doi.acm.org/10.1145/1143844.1143874 (cit. on p. 97).

[Cow11] Frank Cowell. *Measuring Inequality*. 3rd ed. Oxford University Press, 2011 (cit. on p. 98).

[Ken15] Alex Kendall, Vijay Badrinarayanan, and Roberto Cipolla. "Bayesian SegNet: Model Uncertainty in Deep Convolutional Encoder-Decoder Architectures for Scene Understanding". In: *CoRR* abs/1511.02680 (2015). arXiv: 1511.02680 (cit. on p. 95).

[Mar15] Martin Abadi et al. *TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems*. Software available from tensorflow.org. 2015. URL: https://www.tensorflow.org/ (cit. on p. 100).

[Men15] B. Menze et al. "The Multimodal Brain Tumor Image Segmentation Benchmark (BRATS)". In: *IEEE Transactions on Medical Imaging* 34.10 (Oct. 2015), pp. 1993–2024. ISSN: 0278-0062. DOI: 10.1109/TMI.2014.2377694 (cit. on pp. 97, 104).

[Ron15] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. "U-Net: Convolutional Networks for Biomedical Image Segmentation". In: *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*. Ed. by Nassir Navab et al. Cham: Springer International Publishing, 2015, pp. 234–241. ISBN: 978-3-319-24574-4 (cit. on p. 105).

[Cor16] Marius Cordts et al. "The Cityscapes Dataset for Semantic Urban Scene Understanding". In: *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2016 (cit. on pp. 97, 100).

[Gal16] Yarin Gal and Zoubin Ghahramani. “Dropout As a Bayesian Approximation: Representing Model Uncertainty in Deep Learning”. In: *Proceedings of the 33rd International Conference on International Conference on Machine Learning - Volume 48*. ICML’16. New York, NY, USA: JMLR.org, 2016, pp. 1050–1059. URL: http://dl.acm.org/citation.cfm?id=3045390.3045502 (cit. on p. 95).

[Hen16] Dan Hendrycks and Kevin Gimpel. “A Baseline for Detecting Misclassified and Out-of-Distribution Examples in Neural Networks”. In: *CoRR* abs/1610.02136 (2016). arXiv: 1610.02136. URL: http://arxiv.org/abs/1610.02136 (cit. on p. 95).

[Hua16] C. Huang, Q. Wu, and F. Meng. “QualityNet: Segmentation quality evaluation with deep convolutional networks”. In: *2016 Visual Communications and Image Processing (VCIP)*. Nov. 2016, pp. 1–4. DOI: 10.1109/VCIP.2016.7805585 (cit. on p. 96).

[Kam16] Michael Kampffmeyer, Arnt-Borre Salberg, and Robert Jenssen. “Semantic Segmentation of Small Objects and Modeling of Uncertainty in Urban Remote Sensing Images Using Deep Convolutional Neural Networks”. In: *2016 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)* (2016), pp. 680–688 (cit. on p. 95).

[Bak17] Spyridon Bakas et al. “Advancing The Cancer Genome Atlas glioma MRI collections with expert segmentation labels and radiomic features”. In: *Scientific Data* 4 (Sept. 2017). Data Descriptor. URL: https://doi.org/10.1038/sdata.2017.117 (cit. on pp. 97, 104).

[Cho17] François Chollet. “Xception: Deep Learning with Depthwise Separable Convolutions”. In: *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2017), pp. 1800–1807 (cit. on p. 100).

[Lia17] Shiyu Liang, Yixuan Li, and R. Srikant. “Principled Detection of Out-of-Distribution Examples in Neural Networks”. In: *CoRR* abs/1706.02690 (2017). arXiv: 1706.02690. URL: http://arxiv.org/abs/1706.02690 (cit. on p. 95).

[Ozd17] Onur Ozdemir, Benjamin Woodward, and Andrew A. Berlin. “Propagating Uncertainty in Multi-Stage Bayesian Convolutional Neural Networks with Application to Pulmonary Nodule Detection”. In: *CoRR* abs/1712.00497 (2017). arXiv: 1712.00497. URL: http://arxiv.org/abs/1712.00497 (cit. on p. 96).

[Che18] Liang-Chieh Chen et al. “Encoder-Decoder with Atrous Separable Convolution for Semantic Image Segmentation”. In: *CoRR* abs/1802.02611 (2018) (cit. on p. 100).

[DeV18] Terrance DeVries and Graham W. Taylor. “Leveraging Uncertainty Estimates for Predicting Segmentation Quality”. In: *CoRR* abs/1807.00502 (2018). arXiv: 1807.00502. URL: http://arxiv.org/abs/1807.00502 (cit. on p. 96).

[Hua18] P.Y. Huang et al. “Efficient Uncertainty Estimation for Semantic Segmentation in Videos”. In: *European Conference on Computer Vision (ECCV)*. 2018 (cit. on p. 95).

[Ker18] Adel Kermi, Issam Mahmoudi, and Mohamed Tarek Khadir. “Deep Convolutional Neural Networks Using U-Net for Automatic Brain Tumor Segmentation in Multimodal MRI Volumes”. In: *International MICCAI Brainlesion Workshop*. Springer. 2018, pp. 37–48 (cit. on p. 105).

[Obe18] Philip Oberdiek, Matthias Rottmann, and Hanno Gottschalk. “Classification Uncertainty of Deep Neural Networks Based on Gradient Information”. In: *Artificial Neural networks and Pattern Recognition (ANNPR)*. 2018 (cit. on p. 95).

[Roy18] Abhijit Guha Roy et al. “Inherent brain segmentation quality control from fully convnet Monte Carlo sampling”. In: *International Conference on Medical Image Computing and Computer-Assisted Intervention*. Springer. 2018, pp. 664–672 (cit. on pp. 96, 97).

[San18] Mark Sandler et al. “Inverted Residuals and Linear Bottlenecks: Mobile Networks for Classification, Detection and Segmentation”. In: *CoRR* abs/1801.04381 (2018) (cit. on p. 100).

[Wic18] Kristoffer Wickstrøm, Michael Kampffmeyer, and Robert Jenssen. “Uncertainty and Interpretability in Convolutional Neural Networks for Semantic Segmentation of Colorectal Polyps”. In: *CoRR* abs/1807.10584 (2018). arXiv: 1807.10584. URL: http://arxiv.org/abs/1807.10584 (cit. on p. 95).

[Maa19] Kira Maag, Matthias Rottmann, and Hanno Gottschalk. “Time-Dynamic Estimates of the Reliability of Deep Semantic Segmentation Networks”. In: *CoRR* abs/1911.05075 (2019). arXiv: 1911.05075. URL: http://arxiv.org/abs/1911.05075 (cit. on p. 97).

[Myr19] Andriy Myronenko. “3D MRI Brain Tumor Segmentation Using Autoencoder Regularization”. In: *Brainlesion: Glioma, Multiple Sclerosis, Stroke and Traumatic Brain Injuries*. Ed. by Alessandro Crimi et al. Cham: Springer International Publishing, 2019, pp. 311–320. ISBN: 978-3-030-11726-9 (cit. on p. 105).

[Rot19] Matthias Rottmann and Marius Schubert. “Uncertainty Measures and Prediction Quality Rating for the Semantic Segmentation of Nested Multi Resolution Street Scene Images”. In: *CoRR* abs/1904.04516 (2019). arXiv: 1904.04516. URL: http://arxiv.org/abs/1904.04516 (cit. on p. 97).

Controlled False Negative Reduction of Minority Classes in Semantic Segmentation

Robin Chan[1], Matthias Rottmann[1], Fabian Hüger[2], Peter Schlicht[2] and Hanno Gottschalk[1]

[1]School of Mathematics and Natural Sciences, University of Wuppertal
[2]Architecture and AI Technologies, COI Automation, Volkswagen Group Innovation

Abstract. In semantic segmentation datasets, classes of high importance are oftentimes underrepresented, e.g., humans in street scenes. Neural networks are usually trained to reduce the overall number of errors, attaching identical loss to errors of all kinds. However, this is not necessarily aligned with human intuition. For instance, an overlooked pedestrian seems more severe than an incorrectly detected one. One possible remedy is to deploy different decision rules by introducing class priors that assign more weight to underrepresented classes. While reducing the false negatives of the underrepresented class, at the same time this leads to a considerable increase of false positive indications. In this work, we combine decision rules with methods for false positive detection. Therefore, we fuse false negative detection with uncertainty based false positive meta classification. We present the efficiency of our method for the semantic segmentation of street scenes on the Cityscapes dataset based on predicted instances of the "human" class. In the latter we employ an advanced false positive detection method using uncertainty measures aggregated over instances. We, thereby, achieve improved trade-offs between false negative and false positive samples of the underrepresented classes.

I Introduction

Deep learning has improved the state-of-the-art in a broad field of applications such as computer vision, speech recognition and natural language processing by introducing deep convolutional neural networks (CNNs). Although class imbalance is a well-known problem of traditional machine learning models, little work has been done to examine and handle the effects on deep learning models; however, see [Joh19] for a recent review. Class imbalance in a dataset occurs when at least one class contains significantly less examples than another class. The performance of CNNs for classification problems has empirically been shown to be detrimentally affected when applied on skewed training data [Lóp13; Bud18] by revealing a bias towards the overrepresented class. Semantic segmentation, seen as a pixel-wise classification problem, thus exhibits the same set of problems when class imbalance is present. As imbalance naturally exists in most datasets for "real world" applications, finding the underrepresented class is oftentimes of highest interest.

Methods for handling class imbalance have been developed and can be divided into two main categories [Kra16; Bud18; Joh19]: *sampling based* and *algorithm based* techniques. While sampling based methods operate directly on a dataset with the aim to balance its class distribution, algorithm based methods include a cost scheme to modify the learning process or decision making of a classifier.

In the simplest form, data is balanced by randomly discarding samples from frequent (majority) groups and/or randomly duplicating samples from less frequent (minority) groups. These techniques are known as oversampling and undersampling [Van07], respectively. They can lead to performance improvement, in particular with random oversampling [Lóp13; Mas15; Bud18] unless there is no overfitting [Cha04]. A more advanced approach called SMOTE [Cha02] alleviates the latter issue by creating synthetic examples of minority classes.

Sampling based methods are difficult to apply to semantic segmentation datasets due to inherent class frequencies within single input images. Considering the Cityscapes [Cor16] dataset of urban

street scenes for instance, the number of annotated road pixels exceeds the number of annotated person pixels by a factor of roughly 25 despite the fact that persons are already strongly represented in this datatset as exclusively urban street scenes are shown from a car driver's perspective.

In general, class imbalance can be tackled during training by assigning costs to different classification mistakes for different classes and including them in the loss function [Cae15; Wan16; Bul17]. Instead of the total error, the average misclassifcation cost is minimized. In addition, methods learning the cost parameters throughout training have been proposed [Zha16; Kha18] and thus circumventing the ethical problem of predefining them [Cha19b]. These methods require only little tuning and outperform sampling based approaches without significantly affecting training time. Modifying the loss function, however, biases the CNN's output.

One approach to address class imbalance during inference is output thresholding, thus interchanging the standard maximum a-posteriori probability (MAP) principle for an alternative decision rule. Dividing the CNN's output by the estimated prior probabilities for each class is proposed in [Bud18; Cha19a] which is also known as Maximum Likelihood rule in decision theory [Fah96]. This results in a reduced likelihood of misclassifying minority class objects and a performance gain in particular with respect to rare classes. Output thresholding affects neither training time nor the model's capability to discriminate between different groups. It is still a suitable technique for reducing class biases as it shifts the priority to predicting certain classes and it can be easily added on top of any CNN.

In the field of semantic segmentation of street scenes, the overall performance metric intersection over union (IoU) [Eve15] is used primarily. This metric is highly biased towards large and therefore majority class objects such as street or buildings. As a remedy, IoU scores are calculated per class and then averaged. Currently, state-of-the-art models achieve mean class IoU scores of 83% for Cityscapes [Cor16] and 73% for Kitti [Gei13]. Further maximizing global performance measures is important but does not necessarily improve the overall system performance. The priority shifts to rare and potentially more important classes, where the lack of reliable detection has potentially fatal consequences in applications like automated driving.

In this context, uncertainty estimates are helpful as they can be used to quantify the likelihood of predictions being incorrect. Using the maximum softmax probability as confidence estimate has been shown to effectively identify misclassifications in image classification problems which can serve as baseline across many other applications [Hen17]. More advanced techniques include Bayesian neural networks (BNNs) that yield posterior distributions over the model's weight parameters [Nea12]. As BNNs come with a prohibitive computational cost, recent works developed approximations such as Monte-Carlo dropout [Gal16] or stochastic batch normalization [Ata19]. These methods generate uncertainty estimates by sampling, i.e., through multiple forward passes. These sampling approaches are applicable for most CNNs as they do not assume any specific network architecture, but they tend to be computationally expensive during inference. Other frameworks include learning uncertainty estimates via a separate output branch in CNNs [Ken17; DeV18a] which seems to be more adequate in terms of computational efficiency.

In semantic segmentation, uncertainty estimates are usually visualized as spatial heatmaps. Nevertheless, it is possible that CNNs show poor performance but also high confidence scores [Amo16]. Therefore, auxiliary machine learning models for predicting the segmentation quality [Koh12; Zha16] have been proposed. While some methods build upon hand-crafted features, some other methods apply CNNs for that task by learning a mapping from the final segmentation to its prediction quality [Hua16; DeV18b]. A segment based prediction rating method for semantic segmentation was proposed in [Rot18] and extended in [Maa19; Rot19]. They derive aggregated dispersion metrics from the CNN's softmax output and pass them through a classifier that discriminates whether a predicted segment intersects with the ground truth or not. These hand-crafted metrics have shown to be

well-correlated with the calculated segment-wise IoU. This method is termed "*MetaSeg*".

In this present work, we introduce a novel method for semantic segmentation in order to reduce the false negative rate of rare class objects and alleviate the effects of strong class imbalance in data. The proposed method consists of two steps: First, we apply the Maximum Likelihood decision rule that adjusts the neural network's probabilistic / softmax output with the prior class distribution estimated from the training set. This way, less instances of rare classes are overlooked but to the detriment of producing more false positive predictions of the same class. Afterwards, we apply MetaSeg to extract dispersion measures from the balanced softmax output and, based upon that, discard the additional false positive segments in the generated segmentation mask.

This work combines the methods presented in [Cha19a] and [Rot18], resulting in a novel approach to reducing false negatives corresponding to rare classes. Some of the techniques used by us for the detection of false positive and false negative samples separately emerge from a quite recent line of development and the present paper contributes to showing their potential when combining them.

In many situations where CNNs are applied in a safety-critical context, weighting all errors equally for pure performance [Cha19b] might be inappropriate. For instance in the use case of autonomous driving, confusing a pedestrian (minority class) with the street (majority class) is more severe than the other way round. The potential consequences of a single event of the first kind (accident with a pedestrian) far outweigh the event's consequences of the second kind (unnecessary emergency stop). Nevertheless, a too frequent occurrence of false positive person indications will considerably degrade the customers' experience. Compared to other methods for false negative reduction, like using different class weightings for decision thresholding, our method provides a more favorable trade-off between error rates. Hence, this work contributes to making alternative decision rules much more favorable in practical applications.

As a pure post-processing tool (no additional CNN inferences, no CNN retraining with modified cost functions and no resampling the dataset are required), our method can be seamlessly added on top of any CNN for semantic segmentation. Compared to a CNN's inference complexity, the complexity of our post-processing step is negligible. We believe that the envisioned use case in automated driving is a consumers' market in which inference cost matters. Hence, our presented method is designed to have online capabilities that are in reach. To the best of our knowledge, in the context of semantic segmentation this is the first work on segment based false negative reduction by purely post-processing CNN inferences.

The remainder of this work is structured as follows: In Sections II and III, we recall the building blocks of our approach, namely the Maximum Likelihood decision rule for the reduction of false negatives and MetaSeg for false positive detection, respectively. In Section IV, we present how these two components are combined. We apply our approach to the application-relevant task of semantic segmentation and show numerical results for the Cityscapes dataset in Section V.

II Maximum likelihood decision rule

Neural Networks for semantic segmentation can be viewed as statistical models providing pixel-wise probability distributions that express the confidence of predicting the correct class label y within a set $\mathcal{Y} := \{1, \ldots, l\}$ of predefined classes. The classification at pixel location $z \in \mathcal{Z}$ is then performed by applying the $argmax$ function to the posterior probabilities / softmax output $p_z(y|x) \in [0, 1]$ after processing image $x \in \mathcal{X}$. In the field of Deep Learning, this decision principle, called the maximum a-posteriori probability (MAP) principle, is by far the most commonly used one:

$$d_{Bayes}(x)_z := \arg\max_{y \in \mathcal{Y}} p_z(y|x) \ . \tag{1}$$

In this way, the overall risk of incorrect classifications is minimized, i.e., for any other decision rule $d : [0,1]^{|\mathcal{Z}|} \mapsto \mathcal{Y}^{|\mathcal{Z}|}$ and with

$$R_{sym}(d) := \frac{1}{|\mathcal{Z}|} \sum_{z \in Z} \sum_{y \in Y} 1_{\{d(x)_z \neq y\}} p_z(y|x) \ \forall \ x \in \mathcal{X} \tag{2}$$

we have $R_{sym}(d_{Bayes}) \leq R_{sym}(d)$. In decision theory, this principle is also known as Bayes decision rule [Fah96] and it incorporates knowledge about the prior class distribution $p(y)$. As a consequence, in cases of large prediction uncertainty the MAP / Bayes rule tends to predict classes that appear frequently in the training dataset when used in combination with CNNs. However, classes of high interest might appear less frequently. Regarding highly unbalanced datasets the Maximum Likelihood (ML) decision rule oftentimes is a good choice as it compensates for the weights of classes induced by priors:

$$\hat{y}_z = d_{ML}(x)_z := \arg\max_{y \in \mathcal{Y}} p_z(x|y) = \arg\max_{y \in \mathcal{Y}} \frac{p_z(y|x)}{p_z(y)} \ . \tag{3}$$

Instead of choosing the class with the largest a-posteriori probability $p_z(y|x)$, the ML rule chooses the class with the largest conditional likelihood $p_z(x|y)$. It is optimal regarding the risk function

$$R_{inv}(d) := \frac{1}{|\mathcal{Z}|} \sum_{z \in Z} \sum_{y \in Y} 1_{\{d(x)_z \neq y\}} p_z(x|y) \ \forall \ x \in \mathcal{X} \tag{4}$$

and in particular $R_{inv}(d_{ML}) \leq R_{inv}(d_{Bayes})$ is satisfied. The ML rule corresponds to the Maximum Likelihood parameter estimation in the sense that it aims at finding the distribution that fits best the observation. In our use case, the ML rule chooses the class that is most typical for a given pattern observed in an image independently of any prior belief, such as the frequency, about the semantic classes. Moreover, the only difference between these two decision rules lies in the adjustment by the priors $p_z(y)$ (see Equation (3) and Bayes' theorem [Joy19]).

Analogously to [Cha19a], we approximate $p_z(y)$ in a position-specific manner using the pixel-wise class frequencies of the training set:

$$\hat{p}_z(y) = \frac{1}{|\mathcal{X}|} \sum_{x \in \mathcal{X}} 1_{\{y_z(x)=y\}} \ \forall \ y \in \mathcal{Y}, z \in \mathcal{Z} \ . \tag{5}$$

Note that there is no training required for the ML rule. Having calculated the priors (see Equation (5)) from the dataset's ground truth once and offline, each ML mask is obtained via one Hadamard product (see Equation (3)), i.e., there is also no additional CNN inference required.

After applying the ML rule, the amount of overlooked rare class objects is reduced compared to the Bayes rule, but to the detriment of overproducing false positives of the same class. Hence, our ultimate goal is to discard as many additionally produced false positive segments as possible while keeping almost all additionally produced true positive segments (that were overlooked by the Bayes rule).

III Prediction error meta classification

In order to decide which additional segments – predicted by ML but not by Bayes – to discard in an automated fashion, we train a binary classifier performing on top of the CNN for semantic segmentation analogously to [Rot18; Rot19]. Given the conditional likelihood (softmax output adjusted with priors),

we estimate uncertainties per segment by aggregating different pixel-wise dispersion measures, such as entropy

$$E_z(x) = -\frac{1}{\log(|\mathcal{Y}|)} \sum_{y \in \mathcal{Y}} p_z(x|y) \log(p_z(x|y)) \ \forall \ z \in \mathcal{Z}, \tag{6}$$

probability margin

$$M_z(x) = 1 - p_z(x|\hat{y}_z) + \max_{y \in \mathcal{Y} \setminus \{\hat{y}_z\}} p_z(x|y) \ \forall \ z \in \mathcal{Z} \tag{7}$$

and variation ratio

$$V_z(x) = 1 - p_z(x|\hat{y}_z) \ \forall \ z \in \mathcal{Z} \ . \tag{8}$$

As uncertainty is typically large at transitions from one class to another (in pixel space, i.e., at transitions between different predicted objects), we additionally treat these dispersion measures separately for each segment's interior and boundary. The generated uncertainty estimates serve as inputs for the auxiliary "meta" model which classifies into the classes $\{IoU = 0\}$ and $\{IoU > 0\}$. Since the classification is employed on segment-level, the method is also termed *MetaSeg*.

We only add minor modifications to the approach for prediction error meta classification, in the following abbreviated as *meta classification*, as in [Rot18]. For instance, instead of computing logistic least absolute shrinkage and selection operator (LASSO [Tib96]) regression fits, we use gradient-boosting trees (GB [Has09]). GB has proven to be a powerful classifier on binary classification problems and structured data with moderate dataset size which both match our problem setting.

In addition to the uncertainty measures, we introduce further metrics indicating incorrect predictions. For localization purposes we include the segment's geometric center

$$G_h(k) = \frac{1}{|k|} \sum_{i=1}^{|k|} h_i \ , \ G_v(k) = \frac{1}{|k|} \sum_{j=1}^{|k|} v_j \tag{9}$$

with $k = \{(h_s, v_s) \in \mathcal{Z}, s = 1, \ldots, |k|\} \in \hat{\mathcal{K}}_x$ being the pixel coordinates of one *segment* (or *connected component*) in the predicted segmentation mask, i.e., a set consisting of neighboring pixel locations with the same predicted class. The geometric center is the mean of all coordinates of a segment in all directions, in our case in horizontal and vertical direction.

Another metric to be included makes use of a segment's vicinity to determine if an object prediction is misplaced. Let $k_{nb} = \{(h', v') \in [h \pm 1] \times [v \pm 1] \subset \mathcal{Z} : (h', v') \notin k, (h, v) \in k\}$ be the neighborhood of $k \in \hat{\mathcal{K}}_x$. Then

$$N(k|y) = \frac{1}{|k_{nb}|} \sum_{z \in k_{nb}} 1_{\{\hat{y}_z = y\}} \ \forall \ y \in \mathcal{Y} \tag{10}$$

states the fraction of pixels predicted to belong to class y in the neighborhood of k.

After computing the aggregated metrics, we obtain a structured dataset with a fixed number $q \in \mathbb{N}$ of features for each single segment $k \in \hat{\mathcal{K}}_x$. Given this dataset, we perform the meta classification with an auxiliary binary classifier.

Note that if one would use the original MetaSeg method [Rot18] and reject false positives, there would only remain holes in the segmentation masks as MetaSeg does not further process the identified false positives. In contrast, our proposed method presented in the next Section IV utilizes two decision rules in combination with the rejection step performed by MetaSeg yielding a simple but powerful tool for producing a new segmentation mask.

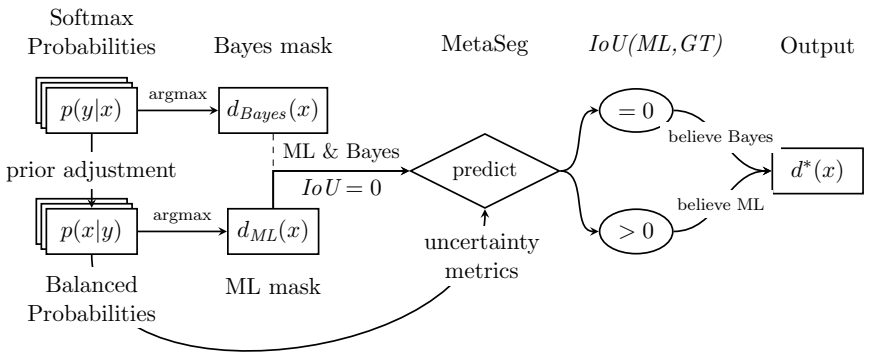


Figure 1: Overview of our method for controlled false negative reduction of minority classes which we term "*MetaFusion*". Note that IoU denotes the intersection over union measure of two segmentation masks.

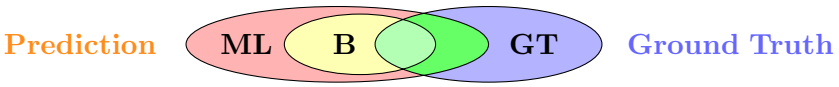


Figure 2: Graphical illustration of the relation between Bayes and ML prediction segments for rare classes.

IV Combining Maximum Likelihood Rule and Meta Classification

After describing the key components of our method for controlled false negative reduction in the preceding sections, we now present our approach as combination of the Maximum Likelihood decision rule and prediction error meta classification for semantic segmentation in more detail. A graphical illustration is provided in Figure 1.

Applying either the Bayes or Maximum Likelihood decision rule may lead to two different prediction masks. They may differ because ML performs a prior adjustment assigning higher weight to underrepresented classes than without this adjustment, consequently increasing the sensitivity towards predicting underrepresented classes. With respect to the most underrepresented class $c \in \mathcal{Y}$ in an unbalanced semantic segmentation dataset, it holds that all predicted Bayes segments are inside ML segments [Cha19a], see Figure 2 for a graphical illustration.

Therefore, we assume that a non-empty intersection between an ML segment and any Bayes segment, which are both assigned to class c, indicates a confirmation for the presence of a minority class object that was already detected by Bayes. In this case, we say *the decision rules agree*. More crucial are predicted ML segments that do not intersect with any Bayes segment of the same class, i.e., *the decision rules disagree*, as these indicate a CNN's uncertain regions where rare instances are potentially overlooked.

The observation whether the decision rules agree or not builds the basis for segment selection for further processing. Let $k \in \hat{\mathcal{K}}_{x,ML}$ be the pixel coordinates of one connected component in the ML mask. Then, given input x,

$$\mathcal{D}_x = \{k \in \hat{\mathcal{K}}_{x,ML} : d_{ML}(x)_z \neq d_{Bayes}(x)_z \ \forall \ z \in k\} \tag{11}$$

denotes the set of segments for which Bayes and ML disagree. Restricting $\mathcal{D}_x$ to a single minority class $c \in \mathcal{Y}$, we obtain the subset $\mathcal{D}_{x|c} = \{k_c \in \mathcal{D}_x : d_{ML}(x)_z = c \ \forall \, z \in k_c\}$. The obtained subset contains the candidates we process with MetaSeg. Let $\mu_k : [0,1]^{|\mathcal{Z}|\times|\mathcal{Y}|} \mapsto \mathbb{R}^q$ be a vector-valued function that

returns a vector containing all generated input metrics for MetaSeg restricted to segment $k \in \mathcal{D}_{x|c}$. We derive aggregated uncertainty metrics per segment

$$U_k := \mu_k\big((\hat{p}(x|y))_{y\in\mathcal{Y}}\big) \;\forall\; k \in \mathcal{D}_{x|c} \tag{12}$$

that serve as input for the meta classifier, see also Section III and cf. [Rot18; Rot19]. The classifier we use in our meta model is the gradient-boosting tree algorithm (GB [Has09]) and it is trained to discriminate between true positive (*detected false negative*) and false positive segment predictions. Thus, we seek a function $\hat{f} : \mathbb{R}^q \mapsto \{0, 1\}$ that learns the mapping

$$f(U_k) = \begin{cases} 1 & \text{if } \exists\; z \in k : d_{ML}(x)_z = y_z \\ 0 & \text{else} \end{cases} \tag{13}$$

with one connected component $k \in \mathcal{D}_{x|c}$ being considered as true positive if there exists (at least) one pixel assigned to the correct class label and as false positive otherwise. In the latter case, we remove that segment from the ML mask and replace it with the Bayes prediction. For the remaining connected components $k' \in \hat{\mathcal{K}}_{x,ML} \setminus \mathcal{D}_{x|c}$, whether or not they are minority class segments, we stick to the Bayes decision rule as it is optimal with respect to the expected total number of errors, see Equation (2). Therefore, the final segmentation output

$$d^*(x)_z = \begin{cases} d_{ML}(x)_z & \text{if } \hat{f}(U_k) = 1 \wedge z \in k \in \mathcal{D}_{x|c} \\ d_{Bayes}(x)_z & \text{else} \end{cases} \tag{14}$$

combines Maximum Likelihood and Bayes decision rule. In this way, compared to standard MAP principle, we sacrifice little in overall performance but significantly improve performance on segment recall for class c. We term our approach *MetaFusion*.

V Numerical Results for Cityscapes

Semantic segmentation is a crucial step in the process of perceiving a vehicle's environment for automated driving. Therefore, we perform tests on the Cityscapes dataset [Cor16] which consists of 2,975 pixel-annotated street scene images of resolution 2048×1024 pixels used for training and further 500 images for validation purposes. CNNs can be trained either on 19 classes or 8 aggregated coarse categories. Our main focus lies on avoiding non-detected humans (ideally without producing any false positive predictions). As all images are recorded in urban street scenes (thus naturally boosting the occurrence of persons), classes like wall, fence or pole are as rare as pedestrians in terms of pixel frequency in the dataset. Therefore, estimating class priors via pixel-wise frequency leads to a weighting not in line with human common sense due to the possible preference of static objects over persons. Therefore, we use category priors treating objects more superficially (by aggregating all classes into the 8 predefined categories), with pedestrians and rider aggregated to the class *human*, then being significantly underrepresented compared to all remaining categories.

We perform the Cityscapes experiments using DeeplabV3+ networks [Che18] with MobileNetV2 [San18] and Xception65 [Cho17] backbones. We apply MetaFusion per predicted human segment as presented in Section IV and evaluate the modified predictions with respect to the human class in the Cityscapes validation data. As meta classifier we employ GB with $q = 56$ inputs, 27 boosting stages, maximum depth of 3 per tree, exponential loss and 5 features to consider when looking for the best split. MetaFusion is 5-fold cross-validated. Numerical results are listed in Table 1.

Priors interpol. degree α	Adjusted Decision Rule $mIoU$	FP	FN	Δ	MetaFusion $mIoU$	FP	FN	Δ
	DeeplabV3+ MobileNetV2 on Cityscapes validation set							
0.000 (Bayes)	0.684	865	839	-	0.684	865	839	-
0.900	0.675	1644	**631**	3.735	0.683	**1167**	720	**2.538**
0.950	0.668	1988	**571**	4.190	0.682	**1169**	670	**1.799**
0.975	0.661	2352	**533**	4.860	0.681	**1191**	648	**1.701**
0.990	0.653	2827	**496**	5.720	0.680	**1247**	611	**1.676**
0.995	0.649	3155	**485**	6.469	0.680	**1329**	586	**1.834**
1.000 (ML)	0.600	4885	**476**	11.074	0.680	**1606**	553	**2.590**
	DeeplabV3+ Xception on Cityscapes validation set							
0.000 (Bayes)	0.753	774	679	-	0.753	774	679	-
0.900	0.746	1314	**530**	**3.624**	0.752	**1055**	614	4.323
0.950	0.742	1579	**487**	4.193	0.752	**1079**	583	**3.177**
0.975	0.737	1783	**458**	4.566	0.751	**1118**	571	**3.185**
0.990	0.732	2068	**433**	5.260	0.751	**1103**	549	**2.531**
0.995	0.731	2219	**425**	5.689	0.750	**1154**	532	**2.585**
1.000 (ML)	0.705	3003	**421**	8.640	0.750	**1272**	508	**2.912**

Table 1: Performance comparison of different decision rules and MetaFusion for DeeplabV3+ with MobileNetV2 and Xception65 backbones on Cityscapes. The decision rules are obtained according to Equation (16) by differently interpolating between priors. The performance is measured using the mean intersection over union ($mIoU$), number of false positive (FP) / false negative (FN) human segments and the trade-off slope Δ (see Equation (17)).

As a baseline we interpolate the priors between the Bayes and Maximum Likelihood decision rules in order to understand how they translate into each other, i.e., we use the priors

$$p_{z,\alpha}(y) = (1-\alpha)1 + \alpha p_z(y) \;\forall\; y \in \mathcal{Y}, z \in \mathcal{Z}, \tag{15}$$

with $\alpha \in [0,1]$, resulting in the adjusted decision rule

$$d_{adj}(x,\alpha)_z := \arg\max_{y\in\mathcal{Y}} \frac{p_z(y|x)}{p_{z,\alpha}(y)} \tag{16}$$

with $d_{adj}(x,0) = d_{Bayes}(x)$ and $d_{adj}(x,1) = d_{ML}(x)$. By varying the coefficient α we obtain the blue line in Figure 3 that may serve as an intuitive approach to balance false negatives (FNs) and false positives (FPs). For each of the points given on the blue curve we apply MetaFusion (green line). Thus, many of the overproduced FPs are removed, however, at the same time we also have to sacrifice some of the detected FNs. In other words, we sacrifice only a small number of the newly found true positives which MetaFusion incorrectly discards.

Although there exist different techniques from traditional machine learning for handling class imbalance, they cannot be applied offhand in semantic segmentation. This includes sampling-based methods as the class imbalance is often inherent in street scene images. Algorithm-based techniques are computationally expensive since good reweighting factors are not known a-priori. Thus, we choose probability thresholding as the only baseline.

The main evaluation metrics that serve for our evaluation are the numbers of false positives (FP) and false negatives (FN) with respect to the minority class "human". Another measure for MetaFusion is the ratio between prediction errors: For any decision rule $d : [0,1]^{|\mathcal{Y}|} \times \mathbb{R} \mapsto \mathcal{Y}$, such that

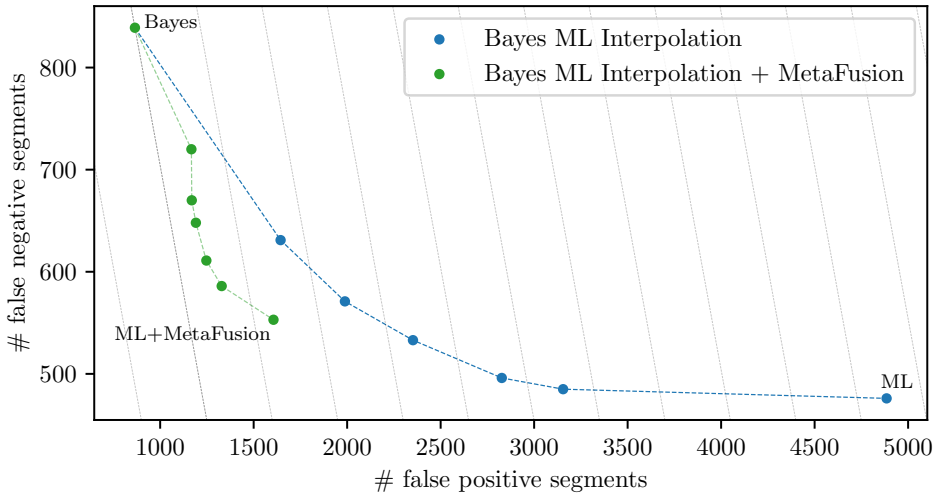


Figure 3: False positives vs. false negatives of human segments for MobileNetV2 on Cityscapes. The blue curve is obtained by applying adjusted decision rules according to Equation (16) with varied degrees of prior interpolation α. For each of the points given on the blue curve we additionally apply MetaFusion which results in the green points given in the figure. The diagonal gray lines depict level sets along which the sum of both errors is constant.

$FN(d_{Bayes}) - FN(d) \neq 0$, the slope

$$\Delta(d) = \frac{FP(d) - FP(d_{Bayes})}{FN(d_{Bayes}) - FN(d)} \tag{17}$$

describes how many additional FPs we have to accept for removing a single FN compared to the Bayes decision rule. The smaller Δ, the more favorable the trade-off between the two error rates. In fact, $\Delta < 1$ indicates that for the considered minority class the total number of errors is decreased by applying d compared to d_{Bayes} (whereas it may increase for the other classes).

We interpolate between Bayes and ML priors according to Equation (15) for every pixel location $z \in \mathcal{Z}$. We observe that an interpolation degree of $\alpha < 0.9$ for the adjusted decision rules (see Equation (16)) leads to a lack of meta training data as their predictions do not differ substantially. Moreover, we choose unevenly spaced steps $\alpha \in \{0.9, 0.95, 0.975, 0.99, 0.995, 1\}$ due to a drastic increase in error rates for interpolation degrees close to 1.

For MobileNetV2, see also Figure 3, we observe that the number of FPs increases from 865 up to 4885 when applying ML instead of Bayes while the number of FNs decreases from 839 down to 476. This results in a large $\Delta = 11.07$ expressing that roughly 11 FPs are paid for removing a single FN. Clearly, there is an overproduction of predicted human segments that we can keep under control using MetaFusion.

By applying MetaFusion, the number of FPs is reduced to a third of ML's FPs while maintaining more than two thirds (78.79%) of the additional true positives. This results in $\Delta = 2.59$ which is a significant decrease compared to plain ML without MetaFusion. With respect to the overall performance, measured by *mean* IoU, MetaFusion sacrifices 0.4 percent points and ML 8.4 percent points. In our experiments we observe that our approach works better the more segments are available for which the decision rules disagree. Therefore, the performance gain with respect to the total number of errors is most significant for $\alpha = 1.0$. For decreasing interpolation degrees, we observe a successive reduction of the total number of errors for the adjusted decision rules. The class weightings'

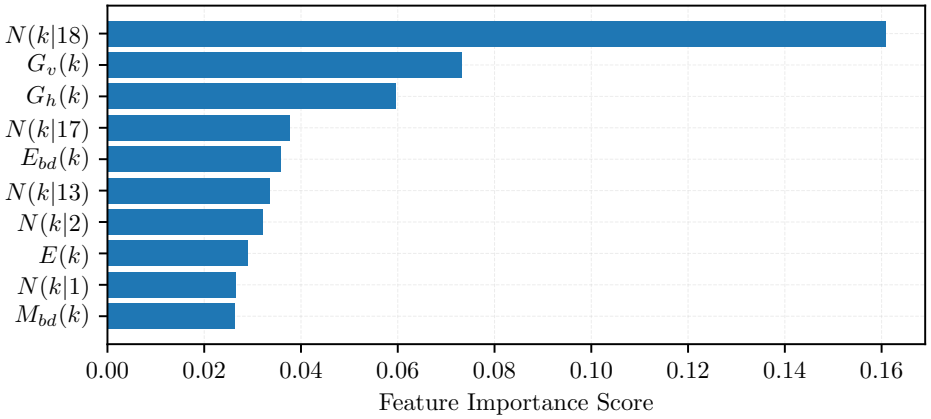


Figure 4: Feature importance scores of the gradient-boosting classifier for MobileNetV2 applied to all disjoint ML and Bayes human segments. The score is averaged over all random cross-validation splits and only the ten features with the highest score are stated. In total we used $q = 56$ metrics as meta model input. N and G are defined in Section III. E and M denote the segment-wise averaged entropy and probability margin, respectively, with bd indicating the restriction on the segment's boundary.

adjustment does not lead to a better performance than Bayes with respect to the absolute number of errors. However, when avoiding FNs is considered to be more important than FPs, our method proposes alternative decision rules that are more attractive than plain decision rules.

For every investigated α MetaFusion is superior to ML regarding the failure trade-off Δ, producing 1.68 additional FPs for removing one single FN as its best performance. In addition, we can conclude that our approach outperforms probability thresholding with respect to the error rates on human segments.

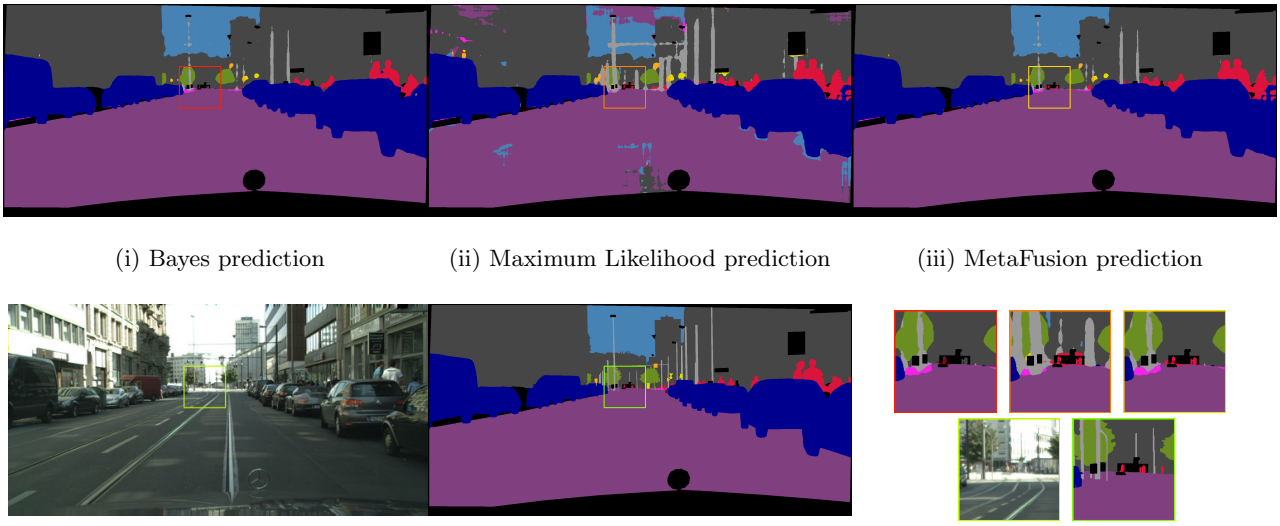
For the stronger DeeplabV3+ model with Xception65 network backbone, we observe similar effects in general. Compared to MobileNetV2, MetaFusion's performance gain over adjusted decision rules is not as great. This is primarily due to the higher confidence scores in the softmax output of the underlying CNN. They prevent the adjusted decision rules from producing segments for which the decision rules disagree. Therefore, the training set size for the meta classifier is rather small even resulting in a worse Δ for MetaFusion than for the adjusted decision rule when $\alpha = 0.90$. Nevertheless, the latter does not hold for the remaining investigated interpolation degrees. Indeed, across the remaining investigated α MetaFusion accepts on average 2.8 FPs for removing a single FN which is less than half of the average Δ (5.7 FPs) for the adjusted decision rules.

In order to find out which of the constructed metrics contribute most to meta classification performance, we analyze our trained GB with respect to feature importance. The latter is a measure indicating the relative importance of each feature variable in a GB model. In a decision tree the importance is computed via

$$I_n(t) = n(t)Q(t) - n_{left}(t)Q_{left}(t) - n_{right}(t)Q_{right}(t) \tag{18}$$

with $Q(t)$ the Gini impurity [Has09] and $n(t)$ the weighted number of samples in node $t \in \mathcal{T}$ (the weighting corresponds to the portion of all samples reaching node t). Moreover, by *left* and *right* we denote the respective child nodes. The importance for $\hat{f}$ of feature / uncertainty metric $m \in [0, 1]$ is then computed as

$$I(m) = \sum_{t \in \mathcal{T}} \chi(t|m) I_n(t) \Big/ \sum_{t \in \mathcal{T}} I_n(t) \tag{19}$$



(i) Bayes prediction (ii) Maximum Likelihood prediction (iii) MetaFusion prediction

(iv) Original raw input image (v) Ground truth segmentation mask (vi) Image zoom into region of interest

Figure 5: Example of generated segmentation masks with MobileNetV2. In the top row: prediction masks using Bayes (i), ML (ii) and MetaFusion (iii). In the bottom row: raw input image (iv), corresponding annotated ground truth mask (v) and zoomed views into the region of interest marked in the latter images (vi). By comparing the prediction masks, we observe a couple of person segments (red color) for which the decision rules disagree and which are correctly identified as false positive according to the ground truth by using MetaSeg. In the end, with MetaFusion we obtain a segmentation mask similar at large to the standard Bayes mask but with some additionally detected person instances (in numbers 3) that are rather small and barely visible in the original image.

with

$$\chi(t|m) = \begin{cases} 1, & \text{if node } t \text{ splits on feature } m \\ 0, & \text{else} \end{cases} . \tag{20}$$

The ten features of highest importance (in experiments with MobileNetV2) are reported in Figure 4. By a large margin, a segment's neighborhood including class id 18, which corresponds to bicycles, has the strongest effect on GB. This is plausible since a bicycle segment adjacent to a human segment can be viewed as an indicator that this human segment is indeed present, i.e., a true positive. Having less than half the importance score, the geometric center still has a relatively high impact on GB. We notice that ML produces many (false positive) segments close to the image borders. This is a consequence of applying pixel-wise ML which GB takes into account. The dispersion measures entropy and probability margin are considered as important features as well expressing the CNN's uncertainty about its prediction. In [Rot18], it already has been shown that these two metrics are well-correlated with the segment-wise IoU. GB also uses these correlations to perform the meta classification. In contrast to the findings in [Rot18], dispersion measures at segment boundaries have greater impact than the dispersion of the interior. This high uncertainty at the boundaries can be interpreted as disturbances for class predictions in a segment's vicinity and may indicate that the investigated segment is a false positive. Moreover, the remaining features in the top ten of highest importance are neighborhood statistics for the classes (in descending order) motorcycle, car, building and sidewalk.

VI Conclusion

In this work, we presented a novel pure post-processing method for semantic segmentation that further processes only the softmax output of any given model. As minority classes are often of highest interest in many real-world applications, the non-detection of their instances might lead to fatal situations and therefore must be treated carefully. In particular, the class person is one such minority class in street scene datasets. We compensate unbalanced class distributions by applying the Maximum Likelihood decision rule that detects a significantly larger number of humans, but also causes an overproduction of false positive indications of the same class. With our method, we are able to detect false positive segment predictions in the ML mask in an automated fashion. These detected false positives are replaced by the Bayes mask. Both, the Bayes and ML mask are obtained from the same inference. Also, the final decision step is not performed by weighting, but by using uncertainty, geometry and location features of the additional minority class segments proposed by ML and passing them through a (in comparison to deep learning models lightweight) gradient-boosting classifier. In our tests with the Cityscapes dataset, we significantly reduce the number of false positives induced by the modification of the decision rule. At the same time, we sacrifice only a small number of newly found true positives which also results only in a minor overall performance loss compared to the standard Bayes decision rule. In fact, our method, which we term *MetaFusion*, clearly outperforms decision rules with different class weightings obtained by interpolating between Bayes and ML rule, i.e., MetaFusion outperforms pure probability thresholding with respect to both error rates, false positive and false negative, of class human. This result holds for the investigated DeeplabV3+ models with MobileNetV2 and Xception65 backbones. The performance gain is more substantial the greater the difference between the Bayes and ML mask. MetaFusion can be viewed as a general concept for trading improved false positive detection for additional performance on rare classes.

For future work we plan to improve our meta classification approach with further heatmaps, metrics as well as component sensitivity to time dynamics. Our approach might also be suitable to serve for query strategies in active learning. Our source code for reproducing experiments is publicly available on GitHub, see `https://github.com/robin-chan/MetaFusion`.

Acknowledgment. This work is funded by Volkswagen Group Innovation. We thank Jan David Schneider and Matthias Fahrland for fruitful discussions.

References

[Fah96] L. Fahrmeir, A. Hamerle, and W. Häussler. . German. 2nd ed. Walter De Gruyter, 1996. ISBN: 978-3110138061 (cit. on pp. 112, 114).

[Tib96] Robert Tibshirani. "Regression Shrinkage and Selection via the Lasso". In: *Journal of the Royal Statistical Society: Series B* 58 (1996), pp. 267–288. URL: https://www.bibsonomy.org/bibtex/290e648276aa6cd3c601e7c0a54366233/dieudonnew (cit. on p. 115).

[Cha02] Nitesh Chawla et al. "SMOTE: Synthetic Minority Over-sampling Technique". In: *J. Artif. Intell. Res. (JAIR)* 16 (Jan. 2002), pp. 321–357. DOI: 10.1613/jair.953 (cit. on p. 111).

[Cha04] Nitesh V. Chawla, Nathalie Japkowicz, and Aleksander Kotcz. "Editorial: Special Issue on Learning from Imbalanced Data Sets". In: *SIGKDD Explor. Newsl.* 6.1 (June 2004), pp. 1–6. ISSN: 1931-0145. DOI: 10.1145/1007730.1007733. URL: http://doi.acm.org/10.1145/1007730.1007733 (cit. on p. 111).

[Van07] Jason Van Hulse, Taghi M. Khoshgoftaar, and Amri Napolitano. "Experimental Perspectives on Learning from Imbalanced Data". In: *Proceedings of the 24th International Conference on Machine Learning*. ICML '07. ACM, 2007, pp. 935–942. ISBN: 978-1-59593-793-3. DOI: 10.1145/1273496.1273614. URL: http://doi.acm.org/10.1145/1273496.1273614 (cit. on p. 111).

[Has09] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The elements of statistical learning: data mining, inference and prediction*. 2nd ed. Springer, 2009. URL: http://www-stat.stanford.edu/~tibs/ElemStatLearn/ (cit. on pp. 115, 117, 120).

[Koh12] Timo Kohlberger, Vivek Singh, Chris Alvino, et al. "Evaluating Segmentation Error without Ground Truth". In: *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2012*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, pp. 528–536 (cit. on p. 112).

[Nea12] Radford M Neal. *Bayesian learning for neural networks*. Vol. 118. Springer Science & Business Media, 2012 (cit. on p. 112).

[Gei13] Andreas Geiger et al. "Vision meets Robotics: The KITTI Dataset". In: *International Journal of Robotics Research (IJRR)* (2013) (cit. on p. 112).

[Lóp13] Victoria López, Alberto Fernández, Salvador García, et al. "An insight into classification with imbalanced data: Empirical results and current trends on using data intrinsic characteristics". In: *Information Sciences* 250 (2013), pp. 113–141. ISSN: 0020-0255. DOI: https://doi.org/10.1016/j.ins.2013.07.007. URL: http://www.sciencedirect.com/science/article/pii/S0020025513005124 (cit. on p. 111).

[Cae15] Holger Caesar, Jasper Uijlings, and Vittorio Ferrari. "Joint Calibration for Semantic Segmentation". In: *Proceedings of the British Machine Vision Conference (BMVC)*. Ed. by Mark W. Jones Xianghua Xie and Gary K. L. Tam. BMVA Press, Sept. 2015, pp. 29.1–29.13. ISBN: 1-901725-53-7. DOI: 10.5244/C.29.29 (cit. on p. 112).

[Eve15] Mark Everingham et al. "The Pascal Visual Object Classes Challenge: A Retrospective". In: *International Journal of Computer Vision* 111.1 (Jan. 2015), pp. 98–136. ISSN: 1573-1405. DOI: 10.1007/s11263-014-0733-5 (cit. on p. 112).

[Mas15] David Masko and Paulina Hensman. "The Impact of Imbalanced Training Data for Convolutional Neural Networks". In: 2015 (cit. on p. 111).

[Amo16] Dario Amodei et al. “Concrete Problems in AI Safety”. In: *CoRR* abs/1606.06565 (2016). arXiv: 1606.06565. URL: http://arxiv.org/abs/1606.06565 (cit. on p. 112).

[Cor16] Marius Cordts, Mohamed Omran, Sebastian Ramos, et al. “The Cityscapes Dataset for Semantic Urban Scene Understanding”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2016 (cit. on pp. 111, 112, 117).

[Gal16] Yarin Gal and Zoubin Ghahramani. “Dropout as a Bayesian Approximation: Representing Model Uncertainty in Deep Learning”. In: *Proceedings of The 33rd International Conference on Machine Learning*. Vol. 48. Proceedings of Machine Learning Research. New York, New York, USA: PMLR, June 2016, pp. 1050–1059. URL: http://proceedings.mlr.press/v48/gal16.html (cit. on p. 112).

[Hua16] C. Huang, Q. Wu, and F. Meng. “QualityNet: Segmentation quality evaluation with deep convolutional networks”. In: *2016 Visual Communications and Image Processing (VCIP)*. Nov. 2016, pp. 1–4. DOI: 10.1109/VCIP.2016.7805585 (cit. on p. 112).

[Kra16] Bartosz Krawczyk. “Learning from imbalanced data: open challenges and future directions”. In: *Progress in Artificial Intelligence* 5.4 (Nov. 2016), pp. 221–232. ISSN: 2192-6360. DOI: 10.1007/s13748-016-0094-0 (cit. on p. 111).

[Wan16] S. Wang, W. Liu, J. Wu, et al. “Training deep neural networks on imbalanced data sets”. In: *2016 International Joint Conference on Neural Networks (IJCNN)*. July 2016, pp. 4368–4374. DOI: 10.1109/IJCNN.2016.7727770 (cit. on p. 112).

[Zha16] C. Zhang, K. C. Tan, and R. Ren. “Training cost-sensitive Deep Belief Networks on imbalance data problems”. In: *2016 International Joint Conference on Neural Networks (IJCNN)*. July 2016, pp. 4362–4367 (cit. on p. 112).

[Bul17] Samuel Rota Bulò, Gerhard Neuhold, and Peter Kontschieder. “Loss Max-Pooling for Semantic Image Segmentation”. In: *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2017), pp. 7082–7091 (cit. on p. 112).

[Cho17] Francois Chollet. “Xception: Deep Learning With Depthwise Separable Convolutions”. In: *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. July 2017 (cit. on p. 117).

[Hen17] Dan Hendrycks and Kevin Gimpel. “A Baseline for Detecting Misclassified and Out-of-Distribution Examples in Neural Networks”. In: *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. 2017. URL: https://openreview.net/forum?id=Hkg4TI9xl (cit. on p. 112).

[Ken17] Alex Kendall and Yarin Gal. “What Uncertainties Do We Need in Bayesian Deep Learning for Computer Vision?” In: *Advances in Neural Information Processing Systems 30*. Curran Associates, Inc., 2017, pp. 5574–5584 (cit. on p. 112).

[Bud18] Mateusz Buda, Atsuto Maki, and Maciej A. Mazurowski. “A systematic study of the class imbalance problem in convolutional neural networks”. In: *Neural Networks* 106 (2018), pp. 249–259. ISSN: 0893-6080. DOI: https://doi.org/10.1016/j.neunet.2018.07.011. URL: http://www.sciencedirect.com/science/article/pii/S0893608018302107 (cit. on pp. 111, 112).

[Che18] Liang-Chieh Chen et al. “Encoder-Decoder with Atrous Separable Convolution for Semantic Image Segmentation”. In: *The European Conference on Computer Vision (ECCV)*. Sept. 2018 (cit. on p. 117).

[DeV18a] Terrance DeVries and Graham W Taylor. "Learning Confidence for Out-of-Distribution Detection in Neural Networks". In: *arXiv preprint arXiv:1802.04865* (2018) (cit. on p. 112).

[DeV18b] Terrance DeVries and Graham W. Taylor. *Learning Confidence for Out-of-Distribution Detection in Neural Networks*. Feb. 2018. arXiv: 1802.04865. URL: http://arxiv.org/abs/1802.04865 (cit. on p. 112).

[Kha18] S. H. Khan et al. "Cost-Sensitive Learning of Deep Feature Representations From Imbalanced Data". In: *IEEE Transactions on Neural Networks and Learning Systems* 29.8 (Aug. 2018), pp. 3573–3587. ISSN: 2162-237X. DOI: 10.1109/TNNLS.2017.2732482 (cit. on p. 112).

[Rot18] Matthias Rottmann, Pascal Colling, Thomas-Paul Hack, et al. "Prediction Error Meta Classification in Semantic Segmentation: Detection via Aggregated Dispersion Measures of Softmax Probabilities". In: *CoRR* (2018). arXiv: 1811.00648. URL: http://arxiv.org/abs/1811.00648 (cit. on pp. 112–115, 117, 121).

[San18] Mark Sandler et al. "MobileNetV2: Inverted Residuals and Linear Bottlenecks". In: *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2018 (cit. on p. 117).

[Ata19] Andrei Atanov et al. "Uncertainty Estimation via Stochastic Batch Normalization". In: *Advances in Neural Networks – ISNN 2019*. Ed. by Huchuan Lu, Huajin Tang, and Zhanshan Wang. Cham: Springer International Publishing, 2019, pp. 261–269 (cit. on p. 112).

[Cha19a] Robin Chan et al. "Application of Decision Rules for Handling Class Imbalance in Semantic Segmentation". In: *CoRR* abs/1901.08394 (2019). arXiv: 1901.08394. URL: http://arxiv.org/abs/1901.08394 (cit. on pp. 112–114, 116).

[Cha19b] Robin Chan et al. "The ethical dilemma when (not) setting up cost-based decision rules in semantic segmentation". In: *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*. Vol. 2019-June. 2019, pp. 1395–1403. ISBN: 9781728125060. DOI: 10.1109/CVPRW.2019.00180. eprint: 1907.01342 (cit. on pp. 112, 113).

[Joh19] Justin M. Johnson and Taghi M. Khoshgoftaar. "Survey on deep learning with class imbalance". In: *Journal of Big Data* 6.1 (Mar. 2019), p. 27. ISSN: 2196-1115. DOI: 10.1186/s40537-019-0192-5 (cit. on p. 111).

[Joy19] James Joyce. "Bayes' Theorem". In: *The Stanford Encyclopedia of Philosophy*. Ed. by Edward N. Zalta. Metaphysics Research Lab, Stanford University, 2019 (cit. on p. 114).

[Maa19] Kira Maag, Matthias Rottmann, and Hanno Gottschalk. "Time-Dynamic Estimates of the Reliability of Deep Semantic Segmentation Networks". In: *CoRR* abs/1911.05075 (2019). arXiv: 1911.05075. URL: http://arxiv.org/abs/1911.05075 (cit. on p. 112).

[Rot19] Matthias Rottmann and Marius Schubert. "Uncertainty Measures and Prediction Quality Rating for the Semantic Segmentation of Nested Multi Resolution Street Scene Images". In: *CoRR* abs/1904.04516 (2019). arXiv: 1904.04516. URL: http://arxiv.org/abs/1904.04516 (cit. on pp. 112, 114, 117).

Detecting Out of Distribution Objects in Semantic Segmentation of Street Scenes

Dominik Brüggemann[1], Robin Chan[2], Matthias Rottmann[2], Hanno Gottschalk[2] and Stefan Bracke[1]

[1]Chair of Reliability Engineering and Risk Analytics, University of Wuppertal
[2]School of Mathematics and Natural Sciences, University of Wuppertal

Abstract. Convolutional neural networks (CNNs) have seen spectacular advances over the past century, particularly improving the state-of-the-art in computer vision tasks. Semantic segmentation, an image classification at pixel-level, is an essential step in understanding a vehicle's surroundings via camera images for autonomous driving. While CNNs keep becoming more and more powerful predictive models, they still often fail if an input is outside of their learned concepts. The non-detection of objects in street scenes, including out-of-distribution (OOD) objects, poses serious hazard and may cause public harm. Therefore, methods determining when a model has failed are crucial in order to ensure a safe and responsible usage of CNNs in real-world applications. In this work we present a method for the detection of OOD objects. We extend work from image classification to more complex semantic segmentation. Our approach is based on pixel-wise entropy derived from the CNN's probabilistic Softmax output. This dispersion measure can be understood as prediction uncertainty indicating a failure per pixel. Paired with methods from image processing, we determine image regions in which an OOD object might be present but overlooked by the CNN. We show the course of our method's development performed on the semantic segmentation Lost and Found dataset that was inferred using the state-of-the-art CNN DeeplabV3+ with Xception65 network backbone. We provide an in-depth statistical evaluation and discuss strength, but also weakness, of our presented method. Additionally, we perform a brief analysis of the topic from the point of view of safety engineering, including a critical evaluation why common standards like ISO 26262 cannot be applied.

I Introduction

Deep learning algorithms are to be deployed in a variety of real world tasks as they consistently improve the state-of-the-art in fields such as natural language processing, image processing or medical diagnosis. In particular in the field of computer vision the introduction of convolutional neural networks (CNNs, [Kri12]) led to a great impact on recent advances as most deep learning models are based on the convolutional architecture. Even for difficult problems like semantic segmentation, the task of assigning the object class for each pixel within an image, CNNs are the most commonly used network architecture [Lon14]. Despite the high performance that such CNNs provide, the adoption of deep learning in everyday applications is yet to come. One reason is the lack of mandatory safety guarantees in systems that are becoming more and more complex. For instance, the capability of indicating when the underlying model is likely mistaken is still a missing but safety critical prerequisite for the usage of deep learning. Therefore, the behavior of CNNs when they encounter data unlike the training data is an important topic in this area. In this context, samples, that are possibly seen at test time but are different to the ones seen during training, are called out-of-distribution (OOD) samples. One property that one expects CNNs to satisfy is to output high prediction uncertainty for these OOD samples. However, the opposite behavior, i.e., producing high confidence predictions, have been reported in several works, see [Yos15; Hen16; Guo17].

Tackling the OOD detection in CNNs is usually approached by either modifying the training process or by post processing given CNN outputs. The training approach aims at improving the models'

discrimination capabilities between out-of-distribution and in-distribution data [Lee18] or at enforcing models to output uniform confidence scores on OOD inputs [Hen18]. Mathematical guarantees for low confidence scores on OOD samples are additionally provided in the work by [Mei19]. On the contrary, post processing methods do not require any changes to a pre-trained model and identify OOD samples by deriving patterns from the CNNs' output. As baseline approach [Hen16] introduced thresholding on the maximum softmax probability that is outperformed by techniques adjusting the estimated confidence [DeV18; Lia18]. However, all these presented methods are limited to simple image classification problems only.

In this work, we extend OOD detection from image classification to the more demanding and application-relevant task of semantic segmentation. Specifically, we present a post processing method that extracts (pixel-wise) inference uncertainty via the entropy measure from the pixel-wise probabilistic softmax output. Connected components of pixels with high entropy values, that are additionally aggregated using methods from image processing, consequently are supposed to indicate the presence of an (possibly overlooked) OOD object. We perform numerical experiments with the state-of-the-art semantic segmentation network DeeplabV3+ with Xception65 backbone [Che18] trained on the Cityscapes street scenes dataset [Cor16] and test the performance of our presented method on the Lost and Found dataset [Pin16] which includes OOD objects. We provide an in-depth statistical evaluation and discuss strength but also weakness of our presented method.

The remainder of this work is structured as follows: In the second chapter the methodology is presented by building up the necessary fundamental knowledge and explaining it step by step. Moreover the thought process is presented in a traceable way. In the third chapter the results obtained by applying our method to the test set of the Lost and Found data set are presented and analyzed. Moreover strengths and weaknesses of the method and the used evaluations are discussed. In the following, a brief consideration of the problem from a safety perspective is given. The work is concluded with a summary of the results and an outlook.

II Methodology

The method of post processing the semantic segmentation is divided into three sub-processes. During the first sub-process the pixelwise entropy of the Softmax-outputs is determined. The entropy can be used as a measure of the uncertainty of a neural network. In the second sub-process the entropy is filtered in order to find dense regions of high uncertainty. These regions are assumed to contain an overlooked object. In the third sub-process the prediction, i.e. the semantic segmentation, is modified using the information obtained in the second sub-process.

II.i Determination of the entropy

At the beginning of the first sub process the Softmax-output of a segmentation-CNN is given for every image of the data set. The Softmax function is given by the following formula.

$$S(z_i^l) = \frac{e^{(z_i^l)}}{\sum_j e^{(z_j^l)}} \tag{1}$$

Here, z_i^l is the activation of the i-th neuron, of the l-th layer. Applying the Softmax function to the activations of a layer converts these activations into easy-to-read probability-like values. The Softmax function is usually applied to the output layer of a neural network.
In this case, the CNN was trained to distinguish 19 different classes. Consequently, there are 19

softmax probabilities for each pixel of an image. To measure the uncertainty of the CNN for a single pixel, the entropy is calculated.
Entropy is a measure for the average information value of a message and was defined by Claude E. Shannon [Sha48]. In general, the entropy H of a message is calculated as follows.

$$H_I = -\sum_{z \in Z} p_z log_2(p_z) \tag{2}$$

Here, H_I denotes the entropy of a message I, which is built from the letters z of the finite alphabet Z. p_z denotes the probability of occurrence of the letter z.
This formula is adapted to measure the uncertainty of the CNN. As stated before, the CNN was trained to distinguish 19 different classes, which leads to arrays of the size $2048 \times 1024 \times 19$ as the Softmax-output. The entropy is calculated for every individual pixel, using the following formula.

$$H_{pixelwise} = \frac{-\sum_{i=1}^{19} p_i log_2(p_i)}{log_2(19)} = \sum_{i=1}^{19} \frac{p_i log_2(p_i)}{log_2(\frac{1}{19})} \tag{3}$$

Here, H denotes the entropy again. More specifically, the normalized entropy, since the value is divided by $log_2(\frac{1}{19})$, leading to values in the range of $[0; 1]$. p_i denotes the Softmax-Output of the i-th class that the CNN was trained on.
An exemplary result of this first sub process can be seen in figure 1. It can be observed that high entropy values occur in the region of the left object near the center of the image.

II.ii Filtering of the entropy

During the next sub-process, the entropy is filtered in order to find dense regions of high uncertainty. To do this, the relevant area of the image is narrowed down first with the goal to obtain the driveable area seen in the image. We propose two different methods to do so.

Defining the relevant area of the image. The first method uses a fixed mask to define the relevant area of the image (hereinafter referred to as "**fixed** restriction"). The fixed mask, that is used to define the relevant area, was determined by "adding up" all objects that can be seen in a pixel over all images in the training portion of the data set. The resulting image is then blurred using `gaussian_filter(sigma = 25)` from the scipy.ndimage.filters library. Afterwards all values below the threshold of $e - 1$ are discarded. The pixels whose value exceeds the limit form the mask, that is used to define the relevant area.

The second method to define the relevant area of the image uses an adaptive approach to define the relevant area of the image (hereinafter referred to as **adaptive** restriction). As the name suggests, the adaptive restriction determines a different relevant area for every image. The approach uses the Softmax output by defining the relevant area as all pixels, that are in the convex hull of pixels, that are labeled as either street or sidewalk. This area is extended by the area of all pixels that have been labelled as terrain. A pixel is considered to be labeled as category X if the corresponding Softmax output is maximal for the pixel. An exemplary result of this method can be seen in figure 2.

Note that both methods have not been optimized yet. The parameters used were determined by randomly testing some of the values and making a decision based on sound judgment.

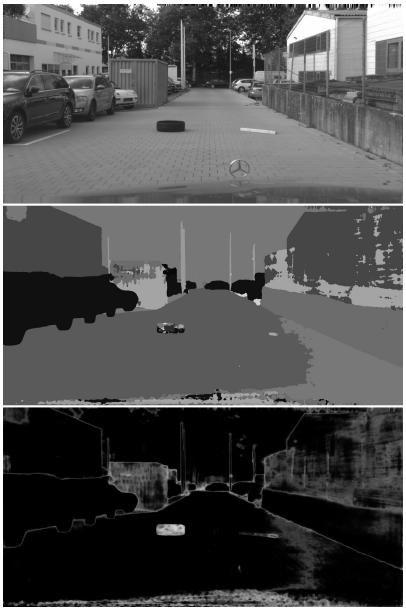


Figure 1: The Softmax probabilities that were obtained by inferring the real image (top) visualized in the semantic segmentation (middle) provide the basis for the calculation of the entropy. The values of the entropy are visualized (bottom) from white for $H = 1$ to black for $H = 0$.

Figure 2: Real image (top), semantic segmentation (middle) and adaptive restriction of the relevant area of this image (bottom).

Procedure of the method. After the relevant area has been narrowed down, the entropy in that relevant area is filtered. Here, the following observations on the obtained entropy images are considered:

1. In multiple cases, the neural network seems to be unsure, whether the road area is to be considered as "road" or "sidewalk" resulting in low amounts of entropy in the area. The cause for this uncertainty is most likely, that in several images the road surface is made out of paving stones, which is typically used for sidewalks. To ignore the entropy values resulting from the uncertainty between two classes, the maximum value of entropy, that can occur in this case, is calculated.

$$\max(H)|[2C] = \sum_{i=1}^{19} \frac{p_i log_2(p_i)}{log_2(19)} \approx \sum_{i=1}^{2} \frac{p_i log_2(p_i)}{log_2(19)} = \frac{2 \cdot 0.5 \cdot log_2(0.5)}{log_2(\frac{1}{19})} = 0,2354 \tag{4}$$

   Here, $\max(H)|[2C]$ denotes the maximum entropy value under the assumption, that all Softmax outputs except two are approximately zero. In order not to take into account entropy values that occur due to the uncertainty between two classes, a threshold value greater than 0,2354 is applied in the method.

2. Between two semantic segments there is almost always a fine line, at which the output of the neural network transitions from one segments class to the other segments class. This results in a fine line of high entropy values. In order to discard those fine lines of high entropy and instead focus on dense regions of high entropy, a series of image filters is applied in the method.

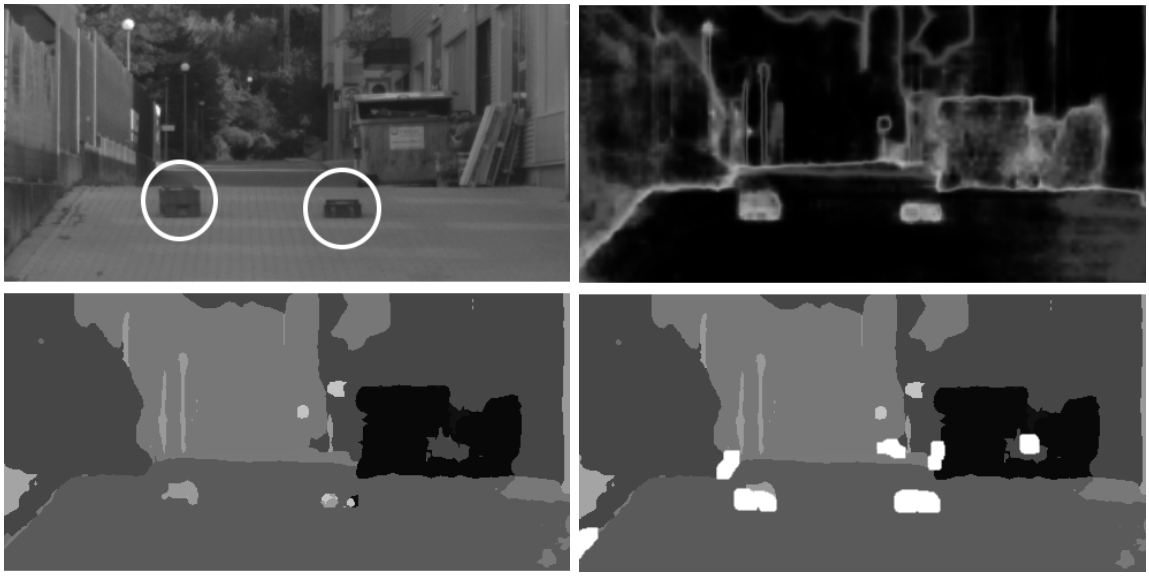


Figure 3: Top left) Real image (with the two objects circled in white for enhanced visibility), bottom left) semantic segmentation / Inference, top right) visualization of the entropy, bottom right) modified segmentation (new segments in white for enhanced visibility).
The segmentation was modified and improved in the region of the two objects that were placed on the street. Further segments that were labeled as "USO" can be found in the region of the dumpster on the right side, slightly above the two objects and in the left foreground of the picture. (total of five false-positives).

The method starts with importing the Softmax outputs and calculating the pixelwise entropy. In the following first filtering step all entropy values that are below the threshold of 0.5 or outside of the defined relevant area of the image are discarded. The values are than scaled up to brightness values by multiplying them by 255. Afterwards a gaussian blur is applied using `cv2.GaussianBlur` with a kernel-size of 11×11. After that, all pixels with a value greater or equal 90 are set to 255 by applying `cv2.threshold(im, 90, 255, cv2.THRESH_BINARY)`. Thereafter `cv2.erode` is applied with two iterations and `cv2.dilate` is applied with four iterations. The idea for this method of filtering entropy values is taken from [Ros16]. The image filters used are taken from the OpenCV library. As before, all used parameters were determined by randomly testing some values and making a decision based on sound judgment. An optimization of the parameters should be done in the future.

Modifying the segmentation. The remaining entropy segments have their origin in dense fields of high entropy. Therefore they can be used to modify the original segmentation in the third and final sub-process of the method. For this purpose a new 20^{th} class is introduced. All pixels that are filtered out by the method are then labeled with that 20^{th} class that was named "USO" (for "unidentified street object"). An exemplary result of the method can be seen in figure 3.

III Results

The evaluations are performed using the `compute_metrics_per_image` function from the MetaSeg-framework by Matthias Rottmann [Rot18]. The function compares two images by scanning all segments of the first image and comparing them to the segments of the second image. The most important

	adaptive	(%)	fixed	(%)
> 0%	1864	100	1719	92.22
> 20%	1864	100	1688	90.56
> 40%	1857	99.62	1667	89.43
> 60%	1842	98.82	1647	88.36
> 80%	1823	97.80	1623	87.07
100%	1751	93.94	1566	84.01

Table 1: Number of objects that are contained in the relevant area for the specified proportion. Note that the columns headed with "(%)" indicate the proportion of the total quantity represented by the respective quantity to the left.

	USO_a	USO_f
# obj	10170	5210
IoU = 0.00	9310	4428
IoU < 0.25	9905	4971
IoU < 0.50	10061	5110
IoU < 0.75	10148	5188

Table 2: Number of false-positives. Note that the first line shows the total number of segments in the respective predictions.

measure, that is taken is the Intersection-over-Union (abbr: IoU), which indicates the ratio of intersection to union. An exemplary application of that function is to scan a Ground Truth image and comparing its segments with the segments of a prediction for the same image. By doing so, the quality of the prediction can be evaluated.

III.i Evaluation of the restriction of the relevant area

To evaluate the two different restriction methods, the number of Ground Truth Objects that are contained in the relevant area to a certain percentage is considered. The results of this evaluation are shown in Table 1.

As stated before, the evaluation of the false-positives shown in table 2 is only there to provide an indication of the size of the relevant areas. As the values show, the segmentation that was made using the adaptive restriction contains 10170 segments and therefore almost twice as much segments as the segmentation that was made using the fixed restriction, which contains 5210 segments. Note, that there is no difference between the two methods other than the restriction of the relevant area. From the values it can be deduced that the fixed restriction provides a relevant area, that is on average significantly smaller than the relevant area provided by the adaptive restriction.
Based on this insight, the adaptive restriction should no longer be considered to be outperforming the fixed restriction. The relevant area of the adaptive restriction contains a greater number of objects, which is at the expense of an averagely larger relevant area and thereby a larger number of false-positives.

III.ii Evaluation of the different Predictions

To evaluate the method, the following three segmentations were evaluated:

- **USO:** This segmentation contains only the unidentified street objects (abbr.: USOs) found using the method described above, i.e. everything that remained after the method was applied to the entropy of the Softmax Outputs (white segments in last image of figure 3). This segmentation is used to test whether objects are actually found by filtering the entropy.

- **NN&USO:** This segmentation contains the USOs as well as everything that has been labeled as non-driveable space in the segmentation, i.e. everything that was not labeled as street, sidewalk or terrain by the neural network. This segmentation is used to evaluate the final result, i.e. the

	NU_a	NU_f	N_a	N_f	U_a	U_f
IoU > 0.0	1238	1144	1235	1142	541	494
IoU > 0.1	909	724	855	690	410	376
IoU > 0.2	795	642	738	595	333	304
IoU > 0.3	698	584	617	518	267	242
IoU > 0.4	609	525	523	442	197	176
IoU > 0.5	526	464	431	375	131	119
IoU > 0.6	419	370	325	283	69	65
IoU > 0.7	285	256	226	195	35	34
IoU > 0.8	156	139	113	97	13	13
IoU > 0.9	16	16	8	8	1	1

Table 3: Number of Ground Truth objects that fulfill the condition in the column on the left. Note that the designations for the different predictions had to be further abbreviated in order to fit the table.
NN&USO → NU; NN → N; USO → U.

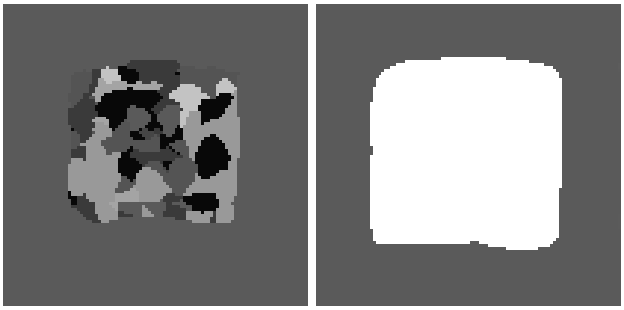


Figure 4: A recognized but unidentified object before (left) and after (right) the method was applied.

modified segmentation, and contains all objects that were recognized by the neural network and the proposed method.

- **NN:** This segmentation contains everything, that was labeled as non-driveable space in the inference of the neural network. This segmentation serves as a baseline for the aforementioned modified segmentation and contains all objects that were recognized (but not necessarily identified, see III.iii) by the neural network.

Each of these segmentations can be made with either method to restrict the relevant area, which were explained earlier (fixed relevant area will be denoted with by $_f$; adaptive relevant area will be denoted by $_a$), leading to a total of six different segmentations that are evaluated in this section.
For the evaluation, the ground truth objects were scanned for intersections with the segmentations. The results of the evaluation are shown in table 3.

As can be seen in the table, the original objective of the work was only partially achieved. The number of ground-truth objects that intersect with the segmentation was increased from 1235 (N_a, baseline) to 1238 (NU_a, modified segmentation) for the adaptive restriction, and from 1142 (N_f) to 1144 (NU_f) for the fixed restriction. This means that only three respectively two objects that were previously undetected were detected by the application of our method. However, the quality of the prediction was significantly improved. Depending on the condition, the improvement on the number of objects ranges from 6,3% (from 855 to 909 objects at 'IoU > 0.1') to 22.0% (from 431 to 526 objects at 'IoU > 0.5') to 34.5% (from 113 to 156 objects at 'IoU > 0.8') up to 100% (from 8 to 16 objects at 'IoU > 0.9') when using the adaptive restriction. For the fixed restriction the number of objects was increased, too. The improvement on the number of objects ranges from 4.9% (from 690 to 724 objects at 'IoU > 0.1') to 23.7% (from 375 to 464 objects at 'IoU > 0.5') up to 100% (from 8 to 16 objects at 'IoU > 0.9'). The relatively low numbers of intersections with the segmentations U_a and U_f indicate, that many of the 1864 objects were recognized and identified by the neural network in good quality. This is why the entropy in the region of those objects was low and thus, no USO-segment was added.

III.iii Further observations

A phenomenon that emerged was, that objects were recognized but not identified by the neural network. This resulted in a potpourri of different classes that pixels of the object were labeled with. In almost all cases of that happening, the entropy was very high in the region of the object. This resulted in a clean detection of the object by our method, which then resulted in a single segment showing the object in the modified prediction. This kind of improvement of the prediction doesn't show in the evaluations, since even though there's a potpourri of classes in the region of the object, the object is recognized and therefore treated as a single segment during the evaluations. An example for the described phenomenon is shown in figure 4.

IV Consideration from a safety perspective

From a safety perspective, the overlooking of objects during semantic segmentation represents a decisive threat to the functional safety of autonomous vehicle control. Our method represents a procedure to improve the quality of a semantic segmentation and does therefore contribute to the functional safety. However, there is currently no standard that defines the functional safety for autonomous driving. Standards to ensure functional safety in the automotive sector are usually given by ISO 26262 [Int18]. However, these standards cannot be applied to autonomously driving vehicles, since, for example, lots of the measures used to rate a hazard in the ISO 26262 evaluate the controllability of a situation by the driver.
A standard which defines requirements and methods for the validation of artificial intelligence for autonomous driving is therefore urgently needed to ensure the safe and responsible usage of artificial intelligence for autonomous driving. Furthermore, a relevance rating of false-positives is needed, which extends the outdated and primitive counting of false-positives to a more informative metric.

V Conclusion and outlook

We presented a method that can improve the quality of a semantic segmentation by calculating and filtering the entropy of the Softmax-outputs of a segmentation-CNN. As part of the method, two different methods to restrict the relevant area of the processed images were presented. Both methods have both advantages and disadvantages compared to the other method.
By applying the presented method on the Lost and Found data set, the original objectives could only be partially achieved. Only few previously undetected objects were detected by our method. However, the quality of the prediction could be significantly improved.
For the future, an optimization of the method by adjusting the used parameters is planned. Moreover, more precise metrics shall be sought in order to make the method more comparable to similar methods. Beyond that, the application of the method to other data sets will be performed. Here, we will strive to find a more completely labeled data set in order to be able to examine and evaluate our method in a more profound way.

References

[Sha48] C. E. Shannon. "A Mathematical Theory of Communication". In: *Bell System Technical Journal* 27.3 (1948), pp. 379–423. ISSN: 00058580. DOI: 10.1002/j.1538-7305.1948.tb01338.x (cit. on p. 129).

[Kri12] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. "ImageNet Classification with Deep Convolutional Neural Networks". In: *Advances in Neural Information Processing Systems 25*. Ed. by F. Pereira et al. Curran Associates, Inc., 2012, pp. 1097–1105. URL: http://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks.pdf (cit. on p. 127).

[Lon14] Jonathan Long, Evan Shelhamer, and Trevor Darrell. "Fully Convolutional Networks for Semantic Segmentation". In: *CoRR* abs/1411.4038 (2014). arXiv: 1411.4038. URL: http://arxiv.org/abs/1411.4038 (cit. on p. 127).

[Yos15] Jason Yosinski and Jeff Clune. "Deep Neural Networks are Easily Fooled : High Confidence Predictions for Unrecognizable Images". In: (2015). arXiv: arXiv:1412.1897v4 (cit. on p. 127).

[Cor16] Marius Cordts, Mohamed Omran, Sebastian Ramos, et al. "The Cityscapes Dataset for Semantic Urban Scene Understanding". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2016 (cit. on p. 128).

[Hen16] Dan Hendrycks and Kevin Gimpel. "A Baseline for Detecting Misclassified and Out-of-Distribution Examples in Neural Networks". In: *CoRR* abs/1610.02136 (2016). arXiv: 1610.02136. URL: http://arxiv.org/abs/1610.02136 (cit. on pp. 127, 128).

[Pin16] Peter Pinggera et al. "Lost and found: Detecting small road hazards for self-driving vehicles". In: *IEEE International Conference on Intelligent Robots and Systems* 2016-November (2016), pp. 1099–1106. ISSN: 21530866. DOI: 10.1109/IROS.2016.7759186. arXiv: 1609.04653 (cit. on p. 128).

[Ros16] Adrian Rosebrock. *Detecting multiple bright spots in an image with Python and OpenCV*. Ed. by pyimagesearch. www.pyimagesearch.com, 2016. URL: https://www.pyimagesearch.com/2016/10/31/detecting-multiple-bright-spots-in-an-image-with-python-and-opencv/ (cit. on p. 131).

[Guo17] Chuan Guo et al. "On calibration of modern neural networks". In: *34th International Conference on Machine Learning, ICML 2017* 3 (2017), pp. 2130–2143. arXiv: 1706.04599 (cit. on p. 127).

[Che18] Liang-Chieh Chen et al. "Encoder-Decoder with Atrous Separable Convolution for Semantic Image Segmentation". In: *The European Conference on Computer Vision (ECCV)*. Sept. 2018 (cit. on p. 128).

[DeV18] Terrance DeVries and Graham W. Taylor. *Learning Confidence for Out-of-Distribution Detection in Neural Networks*. Feb. 2018. arXiv: 1802.04865. URL: http://arxiv.org/abs/1802.04865 (cit. on p. 128).

[Hen18] Dan Hendrycks, Mantas Mazeika, and Thomas Dietterich. "Deep Anomaly Detection with Outlier Exposure". In: (Dec. 2018). arXiv: 1812.04606. URL: http://arxiv.org/abs/1812.04606 (cit. on p. 128).

[Lee18] Kimin Lee et al. “Training confidence-calibrated classifiers for detecting out-of-distribution samples”. In: *6th International Conference on Learning Representations, ICLR 2018 - Conference Track Proceedings* (2018), pp. 1–16. arXiv: 1711.09325 (cit. on p. 128).

[Lia18] Shiyu Liang, Yixuan Li, and R. Srikant. “Enhancing the reliability of out-of-distribution image detection in neural networks”. In: *6th International Conference on Learning Representations, ICLR 2018 - Conference Track Proceedings* 2017 (2018). eprint: 1706.02690 (cit. on p. 128).

[Rot18] Matthias Rottmann, Pascal Colling, Thomas-Paul Hack, et al. “Prediction Error Meta Classification in Semantic Segmentation: Detection via Aggregated Dispersion Measures of Softmax Probabilities”. In: *CoRR* (2018). arXiv: 1811.00648. URL: http://arxiv.org/abs/1811.00648 (cit. on p. 131).

[Mei19] Alexander Meinke and Matthias Hein. “Towards neural networks that provably know when they don’t know”. In: (2019), pp. 1–14. arXiv: 1909.12180. URL: http://arxiv.org/abs/1909.12180 (cit. on p. 128).

[Int18] International Organization for Standardization. *ISO 26262:2018: Road vehicles - Functional safety*. 1.12.2018 (cit. on p. 134).

Entropy Maximization and Meta Classification for Out-of-Distribution Detection in Semantic Segmentation

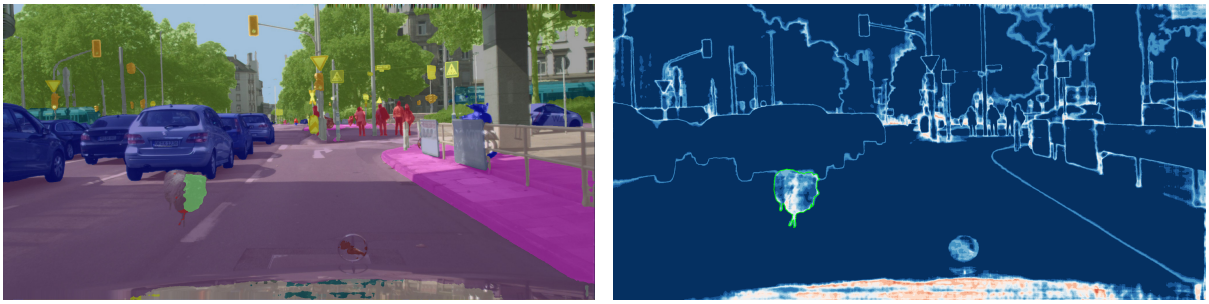
Robin Chan, Matthias Rottmann and Hanno Gottschalk
IZMD, School of Mathematics and Natural Sciences, University of Wuppertal

Abstract. Deep neural networks (DNNs) for the semantic segmentation of images are usually trained to operate on a predefined closed set of object classes. This is in contrast to the "open world" setting where DNNs are envisioned to be deployed to. From a functional safety point of view, the ability to detect so-called "out-of-distribution" (OoD) samples, *i.e.*, objects outside of a DNN's semantic space, is crucial for many applications such as automated driving. A natural baseline approach to OoD detection is to threshold on the pixel-wise softmax entropy. We present a two-step procedure that significantly improves that approach. Firstly, we utilize samples from the COCO dataset as OoD proxy and introduce a second training objective to maximize the softmax entropy on these samples. Starting from pretrained semantic segmentation networks we re-train a number of DNNs on different in-distribution datasets and consistently observe improved OoD detection performance when evaluating on completely disjoint OoD datasets. Secondly, we perform a transparent post-processing step to discard false positive OoD samples by so-called "meta classification." To this end, we apply linear models to a set of hand-crafted metrics derived from the DNN's softmax probabilities. In our experiments we consistently observe a clear additional gain in OoD detection performance, cutting down the number of detection errors by 52% when comparing the best baseline with our results. We achieve this improvement sacrificing only marginally in original segmentation performance. Therefore, our method contributes to safer DNNs with more reliable overall system performance.

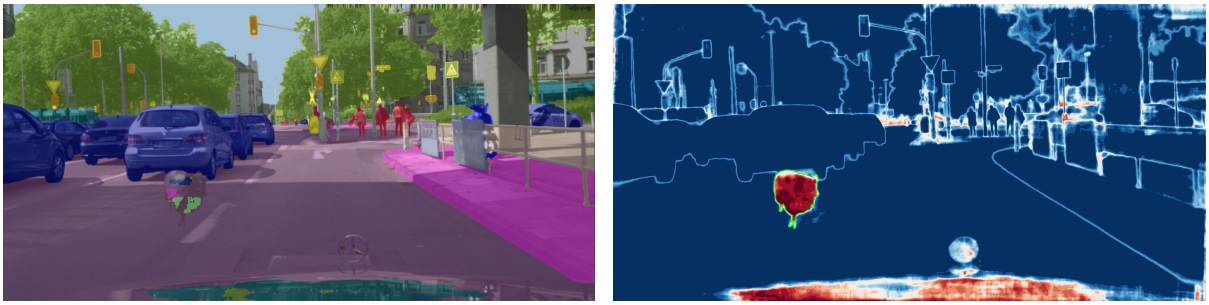
I Introduction

In recent years spectacular advances in the computer vision task semantic segmentation have been achieved by deep learning [Zhu19; Wan20]. Deep convolutional neural networks (CNNs) are envisioned to be deployed to real world applications, where they are likely to be exposed to data that is substantially different from the model's training data. We consider data samples that are not included in the set of a model's semantic space as *out-of-distribution* (OoD) samples. State-of-the-art neural networks for semantic segmentation, however, are trained to recognize a predefined closed set of object classes [Lin14; Cor16], *e.g.* for the usage in environment perception systems of autonomous vehicles [Jan20]. In open world settings there are countless possibly occurring objects. Defining additional classes requires a large amount of annotated data (cf. [Zla18; Col21]) and may even lead to performance drops [Den10]. One natural approach is to introduce a *none-of-the-known* output for objects not belonging to any of the predefined classes [Zha17a]. In other words, one uses a set of object classes that is sufficient for most scenarios and treats OoD objects by enforcing an alternative model output for such samples. From a functional safety point of view, it is a crucial but missing prerequisite that neural networks are capable of reliably indicating when they are operating out of their proper domain, *i.e.*, detecting OoD objects, in order to initiate a fallback policy.

As images from everyday scenes usually contain many different objects, of which only some could be out-of-distribution, knowing the location where the OoD object occurs is desired for practical application. Therefore, we address the problem of detecting anomalous regions in an image, which is the case if an OoD object is present (see Figure 1) and which is a research area of high interest [Pin16; Blu19; Hen19b; Lis19]. This so-called *anomaly segmentation* [Bau18; Hen19b] can be pursued, for



Baseline segmentation mask | Baseline entropy heatmap



Our segmentation mask | Our entropy heatmap

Figure 1: Comparison of segmentation mask and softmax entropy before our OoD training (*top row*) and after (*bottom row*). While there are minor differences in the segmentation masks, the annotated unknown object (marked with green contours) becomes clearly recognizable in the entropy heatmap due to our OoD training. In the heatmap high values are red.

instance, by incorporating sophisticated uncertainty estimates [Gal16; Ata19] or by adding an extra class to the model's learnable set of classes [Zha17a].

In this work, we detect OoD objects in semantic segmentation with a different approach which is composed of two steps: As first step, we re-train the segmentation CNN to predict class labels with low confidence scores on OoD inputs, by enforcing the model to output high prediction uncertainty. In order to quantify uncertainty, we compute the softmax entropy which is maximized when a model outputs uniform probability scores over all classes [Lee18b]. By deliberately including annotated OoD objects as *known unknowns* into the re-training process and employing a modified multi-objective loss function, we observe that the segmentation CNN generalizes learnt uncertainty to unseen OoD samples (*unknown unknowns*) without significantly sacrificing in original performance on the primary task, see Figure 1.

The initial model for semantic segmentation is trained on the Cityscapes data [Cor16]. As proxy for OoD samples we randomly pick images from the COCO dataset [Lin14] excluding the ones with instances that are also available in Cityscapes, cf. [Hei19; Hen19a; Mei20] for a related approach in image classification. We evaluate the pixel-wise OoD detection performance via entropy thresholding for OoD samples from the LostAndFound [Pin16] and Fishyscapes [Blu19] dataset, respectively. Both datasets share the same setup as Cityscapes but include OoD objects.

The second step incorporates a *meta classifier* flagging incorrect class predictions at segment level, similar as proposed in [Rot19; Maa20; Rot20] for the detection of false positive instances in semantic segmentation. After increasing the sensitivity towards predicting OoD objects, we aim at removing false predictions which are produced due to the preceding entropy boost (cf. [Cha20b]). The removal

of false positive OoD object predictions is based on aggregated dispersion measures and geometry features within segments (connected components of pixels), with all information derived solely from the CNN's softmax output. As meta classifier we employ a simple linear model which allows us to track and understand the impact of each metric.

To sum up our contributions, we are the first to successfully modify the training of segmentation CNNs to make them much more efficient at detecting OoD samples in LostAndFound and Fishyscapes. Re-training the CNNs with a specific choice of OoD images from COCO [Lin14] clearly outperforms the natural baseline approach of plain softmax entropy thresholding [Hen17] as well as many state-of-the-art approaches from image classification. In addition, we are the first to demonstrate that entropy based OoD object predictions in semantic segmentation can be meta classified reliably, *i.e.*, classified whether one considered OoD prediction is true positive or false positive without having access to the ground truth. For this meta task we employ simple logistic regression. Combining entropy maximization and meta classification therefore is an efficient and yet lightweight method, which is particularly suitable as an integrated monitoring system of safety-critical real world applications based on deep learning.

II Related Work

Methods from prior works have already proven their efficiency in identifying OoD inputs for image data. The proposed methods are either modifications of the training procedure [Lee18b; Lia18; Hei19; Hen19a; Mei20] or post-processing techniques adjusting the estimated confidence [Hen17; DeV18; Lee18b]. However, most of these works treat entire images as OoD.

When considering the semantic space to be fixed, one possible approach to anomaly segmentation, which we also pursue here, is to estimate uncertainty of CNNs. Early approaches to uncertainty estimation involve Bayesian neural networks (BNNs) yielding posterior distributions over the model's weight parameters [Mac92; Nea12]. In practice, approximations such as Monte-Carlo dropout [Gal16] or stochastic batch normalization [Ata19] are mainly used due to cheaper computational costs. Frameworks using dropout for uncertainty estimation applied to semantic segmentation have been developed in [Bad17; Ken17]. Other approaches to model uncertainty consist of using an ensemble of neural networks [Lak17], which captures model uncertainty by averaging predictions over multiple models, and density estimation [Cho18; Blu19; Nal19; Ren19] via estimating the likelihood of samples with respect to the training distribution. Methods for OoD detection in semantic segmentation based on classification uncertainty and processing only monocular images have been analyzed in [Iso17; Ang19; Brü20; Jou20; Meh20; Obe20].

Using BNNs for estimating uncertainty in deep neural networks is associated with prohibitive computational costs. Uncertainty estimates that are generated by multiple models or by multiple forward passes are still computationally expensive compared to single inference based ones. In our approach, we unite semantic segmentation and OoD detection in one model without any modifications of the underlying CNN's architecture. Therefore, our re-training approach can be even combined with existing OoD detection techniques and potentially enhance their efficiency.

Works with similar training approaches as ours use a different OoD proxy and are presented in [Blu19; Jou20]. They train neural networks on the unlabeled objects in Cityscapes as OoD approximation. However, in our experiments we observe that the unlabeled data in Cityscapes lacks in diversity and therefore tends to be too dataset specific. With respect to other OoD datasets, such as LostAndFound and Fishyscapes, on which we perform our experiments, we observe that these mentioned methods fail to generalize. Furthermore, in contrast to those works we incorporate a post-processing step that significantly improves the OoD detection performance.

Another line of work detects OoD samples in semantic segmentation by incorporating autoencoders [Cre15; Bau18; Akç19; Lis19]. Training such a model only on specific samples from a closed set of classes, it is assumed that the autoencoder model performs less accurately when fed with samples from never-seen-before classes. The identification of an OoD input then relies on the reconstruction quality. In this way, no OoD data is required, except for further adjusting the sensitivity of the method.

Autoencoders are in fact deep neural networks themselves and usually do not include a segmentation model. For the goal of safe real-time semantic segmentation, *e.g.* necessary for automated driving [Jan20], more lightweight approaches are favorable. We avoid incorporating deep auxiliary models at all and only employ a lightweight linear model instead. Usually the more complex a model, the greater the lack of interpretability. As monitoring systems are supposed to make deep learning models safer, one seeks for simpler and thereby more explainable approaches. We post-process our entropy boosted semantic segmentation CNN output via logistic regression whose computational overhead is negligible. This linear model is transparent as it allows us to analyze the impact of each single feature fed into the model and it demonstrates in our experiments to efficiently reduce the number of OoD detection errors.

III Entropy based OoD Detection

In this section, we present our training method to improve the detection of OoD pixels in semantic segmentation via spatial entropy heatmapping.

III.i Training for high Entropy on OoD Samples

Let $f(x) \in (0,1)^q$ denote the softmax probabilities after processing the input image $x \in \mathcal{X}$ with some deep learning model $f : \mathcal{X} \to (0,1)^q$ and let $q = |\mathcal{C}| \in \mathbb{N}$ denote the number of classes. For the sake of brevity, we omit the consideration of image pixels in this section. We compute the softmax entropy via

$$E(f(x)) = -\sum_{j \in \mathcal{C}} f_j(x) \log(f_j(x)) \ . \tag{1}$$

By $(x, y(x)) \sim \mathcal{D}_{in}$ we denote an "in-distribution" example with $y(x) \in \mathcal{C}$ being the ground truth class label of input x, and by $x' \sim \mathcal{D}_{out}$ we denote an "out-distribution" example for which no ground truth label is given. We aim at minimizing the overall objective

$$L := (1-\lambda)\ \mathbb{E}_{(x,y)\sim\mathcal{D}_{in}} \left[\ell_{in}(f(x), y(x))\right] \ + \ \lambda\ \mathbb{E}_{x'\sim\mathcal{D}_{out}} \left[\ell_{out}(f(x'))\right] \ , \ \lambda \in [0,1] \tag{2}$$

where

$$\ell_{in}(f(x), y(x)) := -\sum_{j \in \mathcal{C}} \mathbb{1}_{j=y(x)} \log(f_j(x)) \quad \text{and} \tag{3}$$

$$\ell_{out}(f(x')) := -\sum_{j \in \mathcal{C}} \frac{1}{q} \log(f_j(x')) \tag{4}$$

with the indicator function $\mathbb{1}_{j=y(x)} \in \{0,1\}$ being equal to one if $j = y(x)$ and zero else. In other words, for in-distribution samples we apply the commonly used empirical cross entropy loss, *i.e.*, the negative log-likelihood of the target class. For out-distribution samples, we consider the negative log-likelihood averaged over all classes.

By that choice of out-distribution loss function, minimizing $\ell_{out}(f(x'))$ is equivalent to maximizing the softmax entropy $E(f(x))$, see Equation (1). Since the softmax definition implies $f_j(x) \in (0,1)$ and

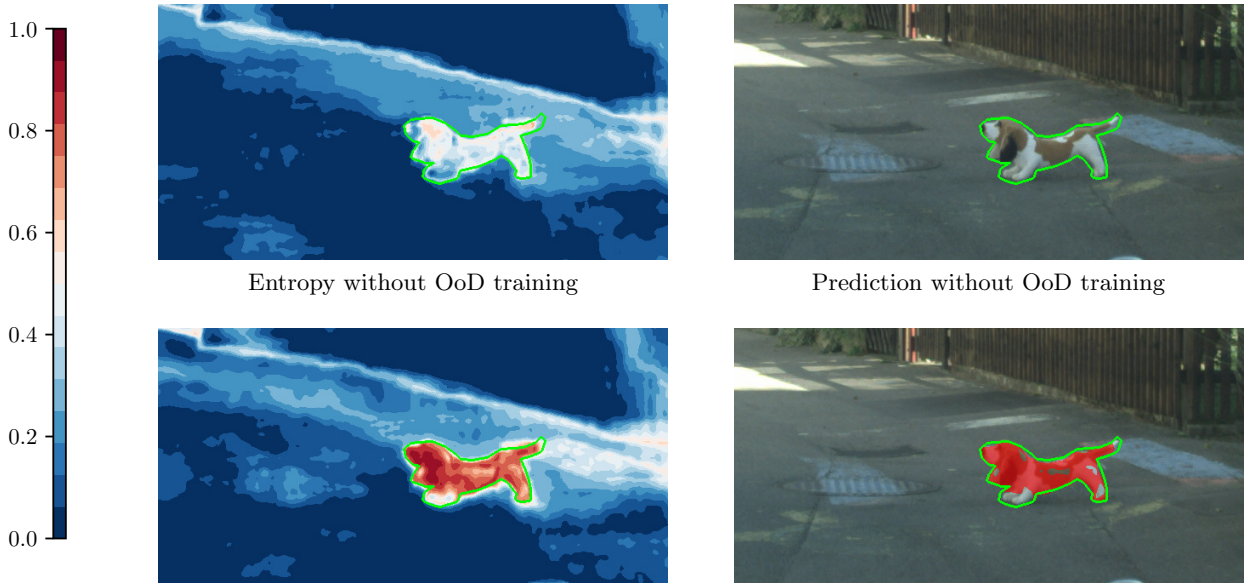


Figure 2: Comparison of softmax entropy heatmap and OoD prediction mask with our OoD training (*bottom row*) and without (*top row*). The green contours in the entropy heatmaps mark the annotation of the OoD object. The OoD object prediction is obtained by simply thresholding on the entropy heatmap (in this example at $t = 0.7$ yielding the red pixels in the OoD prediction masks).

$\sum_{j\in\mathcal{C}} f_j(x) = 1$, Jensen's inequality yields $\ell_{out}(f(x)) \geq \log(q)$ and $E(f(x)) \leq \log(q)$, with equality (for both inequalities) if $f_j(x) = 1/q \;\forall\; j \in \mathcal{C}$, *i.e.*, if the softmax probabilities are uniformly distributed over all classes.

In order to control the impact of each single objective on the overall objective L, the convex combination between expected in-distribution loss and expected out-distribution loss is included, which can be adjusted by varying the parameter λ, see Equation (2).

III.ii OoD Object Prediction in Semantic Segmentation via Entropy Thresholding

The softmax probabilities output of CNNs for semantic segmentation $f(x) \in (0,1)^{|\mathcal{Z}|\times q}, x \in \mathcal{X} \subseteq [0,1]^{|\mathcal{Z}|\times 3}$ can be viewed as pixel-wise probability distributions that express how likely each potential class affiliation $j = 1, \ldots, q$ at a given pixel $z \in \mathcal{Z}$ is, according to the model f. Let $f^z(x) \in (0,1)^q$ denote the softmax output at pixel location z which we implicitly considered throughout the previous section. In semantic segmentation one minimizes the averaged pixel-wise classification loss over the image, cf. Equation (2). For the sake of simplicity, we consider the normalized entropy $\bar{E}(f^z(x))$ at pixel location z in the following, that is $E(f^z(x))$ divided by $\log(q)$. One pixel is then assumed to be out-of-distribution (OoD) if the normalized entropy $\bar{E}(f^z(x))$ at that pixel location z is greater than a threshold $t \in [0,1]$, *i.e.*, *z is predicted to be OoD* if

$$z \in \hat{\mathcal{Z}}_{out}(x) := \{z' \in \mathcal{Z} : \bar{E}(f^{z'}(x)) \geq t\} \,. \tag{5}$$

A connected component $k \in \hat{\mathcal{K}}(x) \subseteq \mathcal{P}(\hat{\mathcal{Z}}_{out}(x))$ (the latter being the power set of $\hat{\mathcal{Z}}_{out}(x)$) consisting of neighboring pixels fulfilling the condition in Equation (5) gives us an *OoD segment / object prediction*. An illustration can be viewed in Figure 2. Obviously, the better an in-distribution pixel can be separated from an out-distribution pixel by means of the entropy, the more accurate the OoD object prediction will be.

IV Meta Classifier in Semantic Segmentation

By training the segmentation CNN to output uniform confidence scores as presented in Section III, we increase the sensitivity towards predicting OoD objects, aiming for an "entropy boost" on OoD samples. However, it is not guaranteed that only OoD samples have a high entropy. Therefore, detecting OoD samples via entropy boosting potentially comes along with a considerable number of false OoD predictions, resulting in an unfavorable trade-off.

In this context, we consider one entire OoD object prediction, *cf*. Section III.ii, as true positive if its intersection over union (IoU, [Eve15]) with a ground truth OoD object is greater than zero. More formally, let $\mathcal{Z}_{out}(x)$ be the set of pixel locations in x which are labeled OoD according to ground truth. Then $k \in \hat{\mathcal{K}}(x)$ *is true positive* (TP) if

$$\mathrm{IoU}(k, \mathcal{Z}_{out}(x)) > 0 \Leftrightarrow \ \exists\, z \in k : \bar{E}(f^z(x)) \geq t \wedge z \in \mathcal{Z}_{out}(x)\ . \tag{6}$$

One could also set a higher threshold on the IoU score, however in this work we treat every single pixel as a potential road hazard as this results in the least possible amount of overlooked OoD objects.

In [Cha20b] it has been demonstrated that false-positives due to increased prediction sensitivity can be removed based on a meta classifier's decision, achieving improved trade-offs between error rates. This meta classifier is essentially a binary classification model added on top of a segmentation CNN [Rot19; Maa20; Rot20]. We construct hand-crafted metrics per connected component of pixels by aggregating different pixel-wise uncertainty measures derived from the softmax probabilities, one of which is the entropy. The entropy metric has proven to be highly correlated to the segment-wise IoU and therefore contributes greatly to the meta classifier's performance, cf. [Rot20]. Therefore, we expect the learned entropy maximization on OoD objects to improve the meta classification performance. In contrast to existing approaches, that consider neighboring pixels sharing the same class label as segment, we generate metrics for segments above the given entropy threshold t to adapt meta classification to OoD detection. Moreover, we additionally consider the variances within segments when aggregating pixel-wise measures instead of the means only.

Given the softmax output, further pixel-wise measures we integrate into the meta classifier are the variation ratio $V(f(x)) = 1 - f_{\hat{c}}(x), \hat{c} = \arg\max_{j \in \mathcal{C}} f_j(x)$ and probability margin $M(f(x)) = V(f(x)) + \max_{j \in \mathcal{C} \setminus \{\hat{c}\}} f_j(x)$. Moreover, we also consider geometry features, such as the segment's size or its ratio between interior and boundary [Rot20]. These metrics serve as inputs for the *meta* model that classifies into *true positive* and *false positive* (FP) OoD object prediction, *i.e.*, classifying $k \in \hat{\mathcal{K}}(x)$ into the sets

$$\begin{aligned} C_{\mathrm{TP}} &:= \{k' \in \hat{\mathcal{K}}(x) : \mathrm{IoU}(k', \mathcal{Z}_{out}(x)) > 0\} \ \text{ and} \\ C_{\mathrm{FP}} &:= \{k' \in \hat{\mathcal{K}}(x) : \mathrm{IoU}(k', \mathcal{Z}_{out}(x)) = 0\}\ . \end{aligned} \tag{7}$$

The outlined hand-crafted metrics form a structured dataset of features where the rows correspond to predicted segments and the columns to metrics.

V Setup of Experiments

We consider the semantic segmentation of the Cityscapes data [Cor16] as original task, *i.e.*, we consider Cityscapes as in-distribution $\mathcal{D}_{in}$. The training split consists of 2,975 pixel-annotated urban street scene images. As original model, we use the state-of-the-art semantic segmentation DeepLabv3+ model with a WideResNet38 backbone trained by Nvidia [Zhu19]. This model is initialized with publicly available weights and serves as our *baseline* model. For testing, we evaluate the OoD detection performance on two datasets comprising street scene images and unexpected objects. We consider

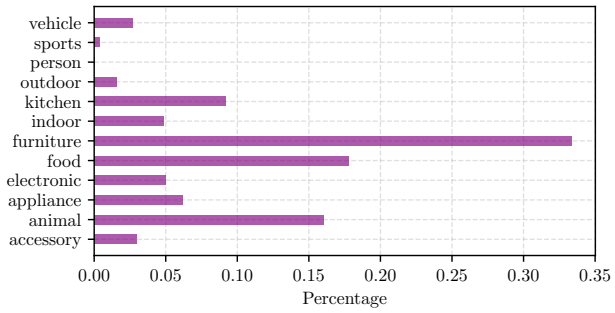
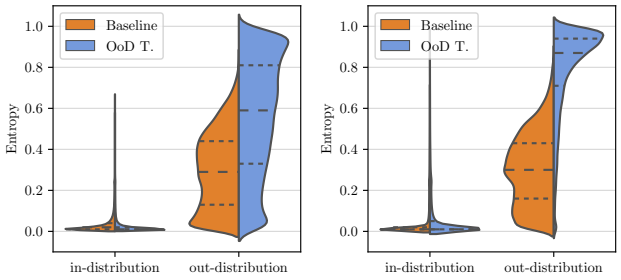


Figure 3: Relative number of pixels per supercategory in the COCO OoD proxy. In every epoch during OoD training 297 out of 46,751 images in total are randomly included.



(a) LostAndFound (b) Fishyscapes

Figure 4: Relative pixel frequencies of (a) LostAndFound and (b) Fishyscapes OoD pixels, respectively. The density at different entropy values is displayed for the baseline model, *i.e.*, before OoD training, and after OoD training.

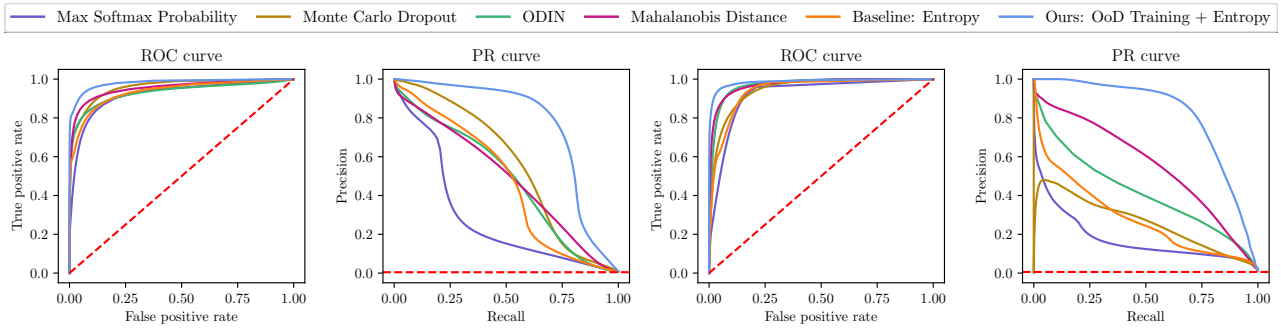
images from the LostAndFound test split [Pin16], containing 1,203 images with annotations of road and small obstacles in front of the (ego-)car, and Fishyscapes Validation [Blu19], containing 30 images with annotated anomalous objects extracted from Pascal VOC [Eve15] which are then overlayed in Cityscapes images. Both datasets share the same setup as Cityscapes but include some unknown road objects.

In order to perform the *OoD training* as proposed in Section III.i, we approximate the out-distribution via images from the COCO [Lin14] dataset. This dataset contains images of objects captured in everyday scenes. Besides, we only consider COCO images with instances that are not included in Cityscapes (no persons, no cars, no traffic lights, *etc.*) and images that have a minimum height and width of at least 480 pixels. After filtering, there remain 46,751 images serving as our proxy for $\mathcal{D}_{out}$. The pixel frequencies per class is visualized in Figure 3. We emphasize that none of the OoD objects in the test data have been seen during our OoD training since we use disjoint datasets for training and testing, that are originally also designed for completely different applications. The used OoD proxy is a mixture of true unknown unknowns (pylon, bloated plastic bag, styrofoam, *etc.*) as well as known unknowns in terms of visual similarities (*e.g.* dogs are available in the test data and share some visual features of cats which are available in the OoD proxy). Employing this COCO subset as approximation of $\mathcal{D}_{out}$ is motivated by works on OoD detection [Hen19a; Mei20] where 80 million tiny images [Tor08] serve as proxy for all possible images.

We finetune the DeepLabv3+ model with loss functions according to Equation (3) and Equation (4). As training data we randomly sample 297 images from our COCO subset per epoch and mix them into all 2,975 Cityscapes training images (1:10 ratio of out-distribution to in-distribution images). We train the model's weight parameters on random squared crops of height / width of 480 pixels for 4 epochs in total and set the (out-distribution) loss weight $\lambda = 0.9$, see Equation (2). As optimizer we use Adam [Kin15] with a learning rate of 10^{-5}.

VI Pixel-wise Evaluation

Based on the softmax probabilities, we compute the normalized entropy $\bar{E}$ for all pixels in the respective test dataset. This gives us per-pixel anomaly / OoD scores which we compare with the ground truth anomaly segmentation. For the sake of clarity, in this section we refer to in-distribution pixels as *samples of the negative class* and to out-distribution pixels as *samples of the positive class*.



(a) LostAndFound (*left:* AUROC, *right:* AUPRC) **(b)** Fishyscapes (*left:* AUROC, *right:* AUPRC)

Figure 5: Detection ability of LostAndFound (a) and Fishyscapes (b) OoD pixels, respectively, evaluated by means of receiver operating characteristic curve (*a & b left*) and precision recall curve (*a & b right*). The red lines indicate the performance according to random guessing, *i.e.*, in the PR curves the red line indicate the fraction of OoD pixels.

VI.i Separability by means of Area Under Curve

On basis of the violin plots in Figure 4, one already notices the beneficial effect of our OoD training over the baseline in separating in-distribution and out-distribution pixels as large masses of the distributions corresponding to the respective classes can be well separated for a larger range of entropy thresholds. This effect can be further quantified with the aid of receiver operating characteristic (ROC) curves and precision recall (PR) curves. The area under the curve (AUC) then represents the degree of separability. The higher the AUC, the better the separability. In addition to the baseline, we include further scores of standard OoD detection methods. Namely these are: MSP [Hen17], MC dropout [Gal16], ODIN [Lia18] and Mahalanobis distance [Lee18a].

By comparing the ROC curves for LostAndFound (Figure 5 (a) left), we observe that there is a performance gain over the baseline model when OoD training is applied. The baseline curve indicates that the corresponding model has a lower true positive rate across various fixed false positive rates, *i.e.*, our model after OoD training assigns higher uncertainty / entropy values to OoD samples which is beneficial for OoD detection. Furthermore, also with respect to all other tested methods, entropy thresholding after OoD training shows the best degree of separability measured by the AUC of ROC curves (AUROC) with a score of 0.98. We observe the same effects for Fishyscapes (Figure 5 (b) left). From the Fishyscapes violins, the discrimination performance after OoD training seems already close to perfect. This is confirmed by the AUROC of 0.99, again outperforming all other tested methods.

As the AUROC essentially measures the overlap of distributions corresponding to negative and positive samples, this score does not place more emphasis on one class over the other in case of class imbalance. As there is a considerably strong class imbalance in LostAndFound and Fishyscapes (0.7% and 1.3% OoD pixels), respectively, we also consider the PR curves, see Figure 5 (a) & (b) right. Thus, true negatives are ignored and the emphasis shifts to the detection of the positive class (OoD samples). Now the AUC of PR curves (AUPRC) serves as measure of separability. For LostAndFound as well as for Fishyscapes OoD pixels, the model after OoD training is superior not only over the baseline model but also any other tested method in terms of precision when we fix recall to any score. The AUPRC quantifies this performance gain and further clarifies the improved capability at detecting OoD pixels. Regarding LostAndFound, the OoD training increases the AUPRC over the baseline by 0.30 up to a score of 0.76. Regarding Fishyscapes, the performance gain is even more significant. We raise the AUC from 0.28 up to 0.81. We conclude that, measured by AUROC and AUPRC, our OoD training is highly beneficial for detecting OoD samples.

Moreover, we conducted the same experiments as for the DeepLabv3+ model [Zhu19] also for the weaker DualGCNNet [Zha19] which is re-trained with $\lambda = 0.25$ for 11 epochs in total. We report all benchmark scores of all tested methods in Table 1. Besides AUPRC, we also provide the false positive rates at 95% true positive rate (FPR_{95}) and the mean intersection over union (mIoU) for the semantic segmentation of the Cityscapes validation set. For further comparison, we additionally included scores of methods based on an auto-encoder [Lis19] and on density estimation [Blu19].

	FPR_{95} ↓	AUPRC ↑	mIoU ↑
Network architecture and OoD score	LostAndFound Test		Cityscapes Val.
DualGCN [Zha19] + Entropy	0.30	0.36	0.80
Ours: DualGCN + OoD T. + Entropy	0.12	0.51	0.76
PSPNet [Zha17b] + Image Resynthesis [Lis19]	N/A	0.41	0.80
DeepV3W + Max Softmax [Hen17]	0.32	0.27	**0.90**
DeepV3W + ODIN [Lia18]	0.45	0.46	**0.90**
DeepV3W + MC Dropout [Gal16]	0.21	0.55	0.88
DeepV3W + Mahalanobis [Lee18a]	0.27	0.48	**0.90**
Baseline: DeepV3W [Zhu19] + Entropy	0.35	0.46	**0.90**
Ours: DeepV3W + OoD T. + Entropy	**0.09**	**0.76**	0.89
	Fishyscapes Val.		Cityscapes Val.
DualGCN [Zha19] + Entropy	0.46	0.07	0.80
Ours: DualGCN + OoD T. + Entropy	0.21	0.38	0.76
DeepV3W + Max Softmax [Hen17]	0.21	0.17	**0.90**
DeepV3W + ODIN [Lia18]	0.12	0.39	**0.90**
DeepV3W + MC Dropout [Gal16]	0.23	0.26	0.88
DeepV3W + Mahalanobis [Lee18a]	0.14	0.55	**0.90**
Baseline: DeepV3W [Zhu19] + Entropy	0.18	0.28	**0.90**
Ours: DeepV3W + OoD T. + Entropy	**0.05**	**0.81**	0.89

Table 1: Results for LostAndFound and Fishyscapes[1].

VI.ii Original Task Performance

In order to monitor that the baseline model does not unlearn its original task due to OoD training, we evaluate the model's performance on in-distribution data with OoD predictions at different entropy thresholds. The original task is the semantic segmentation of the Cityscapes images and we evaluate by means of the most commonly used performance metric *mean Intersection over Union* (mIoU, [Eve15]). Additionally to the Cityscapes class predictions, that is obtained via the standard maximum a posteriori (MAP) decision principle [Moh12; Cha20a], we consider an extra OoD class prediction if the softmax entropy is above the given threshold t. We compute the mIoU for the Cityscapes validation dataset, but average only over the 19 Cityscapes class IoUs.

The state-of-the-art DeepLabv3+ model [Zhu19], which serves as our baseline throughout our experiments, achieves an mIoU score of 0.90 on the Cityscapes validation dataset without OoD predictions (implying $t = 1.0$). By re-training the CNN with entropy maximization on OoD inputs, we observe improved OoD-AUPRC scores. This gain at detecting OoD samples comes with a marginal drop in Cityscapes validation mIoU down to 0.89. These two mIoU scores remain nearly constant (deviations less than 1 percent point) for the thresholds $t = 0.3, \ldots, 1.0$. In general, the lower the entropy threshold, the more pixels are predicted to be OoD. For $t = 0.2$ this results in a noticeable performance decrease, 0.05 for the baseline model and 0.03 for the re-trained model, respectively. As displayed in Figure 6 further lowering the threshold leads to an even more significant sacrifice of original performance. Consequently, we consider in the following entropy thresholds of at least $t = 0.3$ since the performance loss seems acceptable, especially in view of a substantially improved OoD detection capability.

[1] At the time of writing, our method ranked 2nd best in public benchmark results: https://fishyscapes.com/results

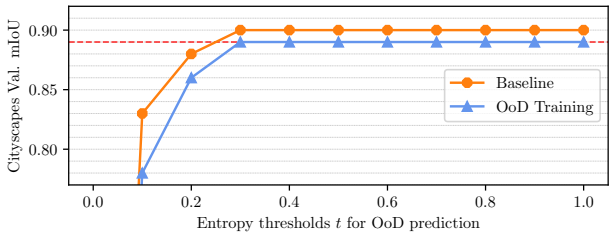


Figure 6: Mean intersection over union (mIoU) for the Cityscapes validation split with OoD predictions at entropy thresholds t. The dashed red line indicates the performance loss considered to be "acceptable" (1 percent point).

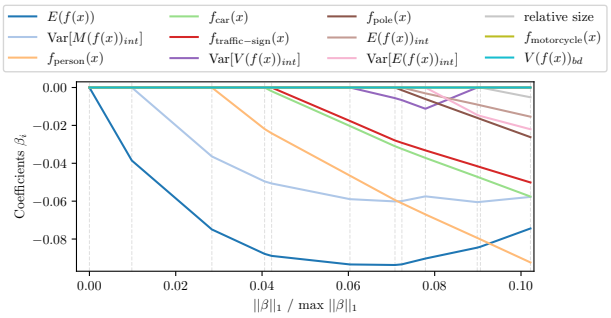


Figure 7: Least angle regression for the meta classifier with OoD training and an entropy threshold of $t = 0.3$. The 12 features becoming active first are displayed. The suffixes *int* and *bd* refer to the restriction of a metric on the segment's interior and boundary, respectively.

VII Segment-wise Evaluation

In this section we evaluate the meta classification performance on LostAndFound. The main metrics for the segment-wise evaluation are the numbers of FPs and FNs with respect to an OoD object prediction, cf. Equation (6). The F_1-score $F_1 = 2\mathrm{TP}/(2\mathrm{TP} + \mathrm{FP} + \mathrm{FN}) \in [0, 1]$ summarizes the error rates into an overall score. As the removal of FP OoD predictions should not come at cost of a significant loss in original performance, we additionally consider the *miss rate of road pixels*:

$$\varepsilon := 1 - \left| \bigcup_{x \in \mathcal{X}} \left(\hat{\mathcal{Z}}_{in}(x) \cap \mathcal{Z}_{in}(x) \right) \right| \left| \bigcup_{x \in \mathcal{X}} \mathcal{Z}_{in}(x) \right|^{-1} \tag{8}$$

with pixel locations predicted to be in-distribution in $\hat{\mathcal{Z}}_{in}$ and annotated as in-distribution in $\mathcal{Z}_{in}$. The road miss rate ϵ measures the fraction of actual road pixels in the whole dataset which are incorrectly identified.

We compute per-segment metrics as outlined in Section IV for OoD object predictions in the LostAndFound test set and feed them through meta classification models, which are simple logistic regressions throughout our experiments. The segments are then leave-one-out cross validated whether they are TP or FP, see Equation (7). Via least angle regression we analyze the metrics having the most impact on the meta classification. The analysis shows that after OoD training the entropy metric $E(f(x))$ has the most impact, see *e.g.* Figure 7 for $t = 0.3$.

In general, the higher the entropy threshold, the less OoD objects are predicted and consequently less data is fed through the linear models. This explains the observation that meta classifiers identify FPs more reliably the lower t. Due to our OoD training, the meta classifiers demonstrate to be more effective, being most superior when $t = 0.7$. In our experiments, OoD training in combination with meta classification at $t = 0.3$ turns out to be the best OoD detection approach achieving the best result with only 598 errors in total and $F_1 = 0.82$ while having a road miss rate of marginally 0.06%, see also Figure 8. Compared to the best baseline at $t = 0.6$ with $F_1 = 0.69$, we decrease the number of total errors by 52% from 1,242 down to 598. More safety-relevantly, at the same time we significantly reduce the number of overlooked OoD objects by 70% from 1,084 down to 308.

The numbers of detection errors, F_1 scores and road miss rates ε at different entropy thresholds t are summarized in Table 2. The FP OoD removal efficiency is given in Table 3.

Entropy Threshold	Baseline				Baseline + Meta Classifier				OoD Training				OoD Training + Meta Classifier			
$\bar{E} \geq t$	FP ↓	FN ↓	F_1 ↑	ε in % ↓	FP ↓	FN ↓	F_1 ↑	ε in % ↓	FP ↓	FN ↓	F_1 ↑	ε in % ↓	FP ↓	FN ↓	F_1 ↑	ε in % ↓
$t = 0.10$	33,584	77	0.09	7.60	386	314	0.80	3.24	21,967	99	0.12	5.22	245	302	0.83	2.70
$t = 0.20$	19,456	136	0.13	2.48	454	307	0.78	0.93	17,000	127	0.15	2.14	271	303	0.83	0.18
$t = 0.30$	7,349	218	0.28	0.38	412	302	0.79	0.09	8,068	**191**	0.26	0.30	**290**	308	**0.82**	**0.06**
$t = 0.40$	3,214	377	0.42	0.08	280	435	0.77	0.03	4,035	**289**	0.39	0.11	**251**	359	**0.81**	**0.03**
$t = 0.50$	809	662	0.58	0.01	**94**	686	0.71	$<$ **0.01**	1,215	**415**	0.60	0.04	145	447	**0.80**	0.02
$t = 0.60$	158	1,084	0.69	$<$ 0.01	**26**	1,093	0.50	$<$ **0.01**	327	**613**	0.69	0.02	49	619	**0.76**	0.02
$t = 0.70$	10	1,511	0.16	$<$ 0.01	**3**	1,512	0.16	$<$ **0.01**	135	**879**	0.61	0.01	21	881	**0.63**	0.01

Table 2: Detection errors for LostAndFound OoD objects at different entropy thresholds t. We consider the road miss rate ε, see Equation (8), as further measure of loss in original performance (for Cityscapes mIoU, see Figure 6). Below the horizontal line, *i.e.*, $t \geq 0.3$, we consider the loss in original performance to be acceptable, see Section VI.ii for further details.

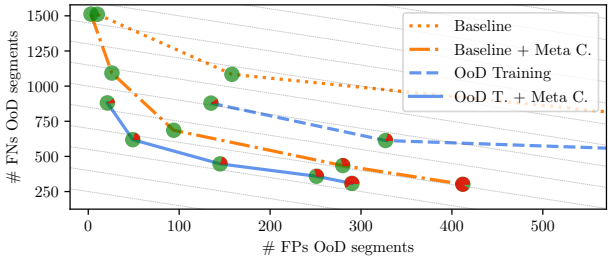


Figure 8: Detection errors of LostAndFound OoD objects. In this plot, the number of errors when $t = 0.7, \ldots, 0.3$ are displayed (when in the axes' range). The pie-chart markers indicate the road miss rate ε, being entirely red if $\varepsilon \geq 0.001$, *cf.* also Table 2.

Entropy Threshold	Base + MSP		Base + Meta		OoD T. + Meta	
	ROC	PRC	ROC	PRC	ROC	PRC
$t = 0.10$	0.850	0.981	0.989	0.999	0.9915	0.999
$t = 0.20$	0.647	0.911	0.985	0.998	0.9898	0.998
$t = 0.30$	0.533	0.737	0.974	0.988	**0.984**	**0.995**
$t = 0.40$	0.384	0.467	0.971	0.974	**0.980**	**0.980**
$t = 0.50$	0.417	0.228	0.962	0.921	**0.966**	**0.953**
$t = 0.60$	0.490	0.122	0.929	0.725	**0.951**	**0.840**
$t = 0.70$	0.593	0.133	0.914	0.528	**0.944**	**0.718**

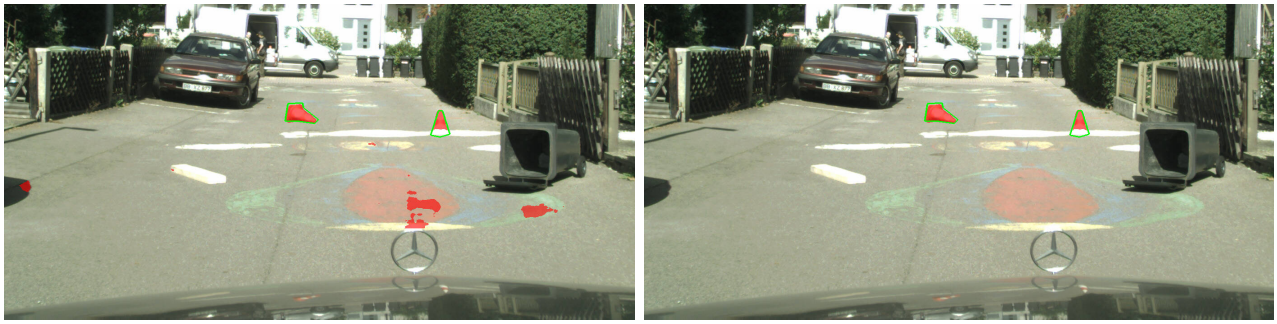
Table 3: Meta classification performance on LostAndFound measured by area under curve. As comparison to the meta classifier, we include the detection of OoD prediction errors via the maximum softmax probability (MSP, [Hen17]).

VIII Conclusion & Outlook

In this work, we presented a novel re-training approach for deep neural networks that unites improved OoD detection capability and state-of-the-art semantic segmentation in one model. Up to now, only a small number of prior works exist for anomaly segmentation on LostAndFound and Fishyscapes, respectively. We demonstrate that our OoD training significantly improves the detection efficiency via softmax entropy thresholding, leading to superior performance over existing OoD detection approaches.

Moreover, we introduced meta classifiers for entropy based OoD object predictions. By applying lightweight logistic regressions, we have demonstrated that entire LostAndFound OoD segments are meta classified reliably. This observation already holds for the tested CNN in its plain version. Due to the increased sensitivity of OoD predictions via entropy maximization, the meta classifiers' efficiency is even more pronounced. In view of emerging safety-critical deep learning applications, the combination of OoD training and meta classification has the potential to considerably improve the overall system's performance.

For future work, we plan to apply OoD training for the retrieval of OoD objects in order to assess the importance of their occurrence and whether a new concept is required to be learned. Our code is publicly available at https://github.com/robin-chan/meta-ood.



OoD Training only | OoD Training + meta classifier

Figure 9: Final OoD detection with $t = 0.5$ after OoD training and meta classification. The green contours mark the annotations of OoD objects, which initially have been entirely non-detected. OoD predictions outside the driveable area are ignored according to the ground truth (this includes *e.g.* the garbage bin even though it has been detected).

Acknowledgement. The research leading to these results is funded by the German Federal Ministry for Economic Affairs and Energy within the project "KI Absicherung – Safe AI for Automated Driving", grant no. 19A19005R. The authors would like to thank the consortium for the successful cooperation. The authors gratefully also acknowledge the Gauss Centre for Supercomputing e.V. (https://www.gauss-centre.eu) for funding this project by providing computing time through the John von Neumann Institute for Computing (NIC) on the GCS Supercomputer JUWELS at Jülich Supercomputing Centre (JSC).

References

[Mac92] David J. C. MacKay. "A Practical Bayesian Framework for Backpropagation Networks". In: *Neural Computation* 4.3 (1992), pp. 448–472 (cit. on p. 139).

[Tor08] Antonio Torralba, Rob Fergus, and William T. Freeman. "80 Million Tiny Images: A Large Data Set for Nonparametric Object and Scene Recognition". In: 30.11 (Nov. 2008), pp. 1958–1970. ISSN: 0162-8828. DOI: 10.1109/TPAMI.2008.128. URL: https://doi.org/10.1109/TPAMI.2008.128 (cit. on p. 143).

[Den10] Jia Deng et al. "What Does Classifying More Than 10,000 Image Categories Tell Us?" In: *Computer Vision – ECCV 2010*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, pp. 71–84. ISBN: 978-3-642-15555-0 (cit. on p. 137).

[Moh12] Mehryar Mohri, Afshin Rostamizadeh, and Ameet Talwalkar. *Foundations of Machine Learning*. The MIT Press, 2012. ISBN: 026201825X (cit. on p. 145).

[Nea12] Radford M Neal. *Bayesian learning for neural networks*. Vol. 118. Springer Science & Business Media, 2012 (cit. on p. 139).

[Lin14] Tsung-Yi Lin, Michael Maire, Serge Belongie, et al. "Microsoft COCO: Common Objects in Context". In: *Computer Vision – ECCV 2014*. Springer International Publishing, 2014, pp. 740–755. ISBN: 978-3-319-10602-1 (cit. on pp. 137–139, 143).

[Cre15] Clement Creusot and Asim Munawar. "Real-time small obstacle detection on highways using compressive RBM road reconstruction". In: *2015 IEEE Intelligent Vehicles Symposium (IV)*. 2015, pp. 162–167 (cit. on p. 140).

[Eve15] Mark Everingham, S. M. Ali Eslami, Luc Van Gool, et al. "The Pascal Visual Object Classes Challenge: A Retrospective". In: *International Journal of Computer Vision* 111.1 (Jan. 2015), pp. 98–136. ISSN: 1573-1405. DOI: 10.1007/s11263-014-0733-5. URL: https://doi.org/10.1007/s11263-014-0733-5 (cit. on pp. 142, 143, 145).

[Kin15] Diederik P. Kingma and Jimmy Ba. "Adam: A Method for Stochastic Optimization". In: *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*. Ed. by Yoshua Bengio and Yann LeCun. 2015. URL: http://arxiv.org/abs/1412.6980 (cit. on p. 143).

[Cor16] Marius Cordts, Mohamed Omran, Sebastian Ramos, et al. "The Cityscapes Dataset for Semantic Urban Scene Understanding". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2016 (cit. on pp. 137, 138, 142).

[Gal16] Yarin Gal and Zoubin Ghahramani. "Dropout as a Bayesian Approximation: Representing Model Uncertainty in Deep Learning". In: *Proceedings of The 33rd International Conference on Machine Learning*. Vol. 48. Proceedings of Machine Learning Research. New York, New York, USA: PMLR, June 2016, pp. 1050–1059. URL: http://proceedings.mlr.press/v48/gal16.html (cit. on pp. 138, 139, 144, 145).

[Pin16] Peter Pinggera et al. "Lost and found: detecting small road hazards for self-driving vehicles". In: *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 2016 (cit. on pp. 137, 138, 143).

[Bad17] Vijay Badrinarayanan, Alex Kendall, and Roberto Cipolla. "Bayesian SegNet: Model Uncertainty in Deep Convolutional Encoder-Decoder Architectures for Scene Understanding". In: *Proceedings of the British Machine Vision Conference (BMVC)*. BMVA Press, Sept. 2017, pp. 57.1–57.12. ISBN: 1-901725-60-X. DOI: 10.5244/C.31.57. URL: https://dx.doi.org/10.5244/C.31.57 (cit. on p. 139).

[Hen17] Dan Hendrycks and Kevin Gimpel. "A Baseline for Detecting Misclassified and Out-of-Distribution Examples in Neural Networks". In: *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. 2017. URL: https://openreview.net/forum?id=Hkg4TI9xl (cit. on pp. 139, 144, 145, 147).

[Iso17] S. Isobe and S. Arai. "Deep convolutional encoder-decoder network with model uncertainty for semantic segmentation". In: *2017 IEEE International Conference on INnovations in Intelligent SysTems and Applications (INISTA)*. 2017, pp. 365–370 (cit. on p. 139).

[Ken17] Alex Kendall and Yarin Gal. "What Uncertainties Do We Need in Bayesian Deep Learning for Computer Vision?" In: *Advances in Neural Information Processing Systems 30*. Curran Associates, Inc., 2017, pp. 5574–5584. URL: http://papers.nips.cc/paper/7141-what-uncertainties-do-we-need-in-bayesian-deep-learning-for-computer-vision.pdf (cit. on p. 139).

[Lak17] Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. "Simple and Scalable Predictive Uncertainty Estimation using Deep Ensembles". In: *Advances in Neural Information Processing Systems 30*. Curran Associates, Inc., 2017, pp. 6402–6413. URL: http://papers.nips.cc/paper/7219-simple-and-scalable-predictive-uncertainty-estimation-using-deep-ensembles.pdf (cit. on p. 139).

[Zha17a] Xiang Zhang and Yann LeCun. "Universum prescription: Regularization using unlabeled data". In: *Thirty-First AAAI Conference on Artificial Intelligence*. 2017 (cit. on pp. 137, 138).

[Zha17b] Hengshuang Zhao et al. "Pyramid Scene Parsing Network". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. July 2017 (cit. on p. 145).

[Bau18] Christoph Baur et al. "Deep autoencoding models for unsupervised anomaly segmentation in brain MR images". In: *International MICCAI Brainlesion Workshop*. Springer. 2018, pp. 161–169 (cit. on pp. 137, 140).

[Cho18] Hyunsun Choi and Eric Jang. "Generative Ensembles for Robust Anomaly Detection". In: *ArXiv* abs/1810.01392 (2018) (cit. on p. 139).

[DeV18] Terrance DeVries and Graham W. Taylor. *Learning Confidence for Out-of-Distribution Detection in Neural Networks*. Feb. 2018. arXiv: 1802.04865. URL: http://arxiv.org/abs/1802.04865 (cit. on p. 139).

[Lee18a] Kimin Lee et al. "A Simple Unified Framework for Detecting Out-of-Distribution Samples and Adversarial Attacks". In: *Advances in Neural Information Processing Systems*. Ed. by S Bengio et al. Vol. 31. Curran Associates, Inc., 2018, pp. 7167–7177. URL: https://proceedings.neurips.cc/paper/2018/file/abdeb6f575ac5c6676b747bca8d09cc2-Paper.pdf (cit. on pp. 144, 145).

[Lee18b] Kimin Lee et al. “Training Confidence-calibrated Classifiers for Detecting Out-of-Distribution Samples”. In: *International Conference on Learning Representations*. 2018. URL: https://openreview.net/forum?id=ryiAv2xAZ (cit. on pp. 138, 139).

[Lia18] Shiyu Liang, Yixuan Li, and R. Srikant. “Enhancing The Reliability of Out-of-distribution Image Detection in Neural Networks”. In: *International Conference on Learning Representations*. 2018. URL: https://openreview.net/forum?id=H1VGkIxRZ (cit. on pp. 139, 144, 145).

[Zla18] Aleksandar Zlateski et al. “On the Importance of Label Quality for Semantic Segmentation”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2018 (cit. on p. 137).

[Akç19] Samet Akçay, Amir Atapour-Abarghouei, and Toby P Breckon. “Skip-ganomaly: Skip connected and adversarially trained encoder-decoder anomaly detection”. In: *2019 International Joint Conference on Neural Networks (IJCNN)*. IEEE. 2019, pp. 1–8 (cit. on p. 140).

[Ang19] Matt Angus, Krzysztof Czarnecki, and Rick Salay. “Efficacy of Pixel-Level OOD Detection for Semantic Segmentation”. In: *CoRR* abs/1911.02897 (2019). arXiv: 1911.02897. URL: http://arxiv.org/abs/1911.02897 (cit. on p. 139).

[Ata19] Andrei Atanov et al. “Uncertainty Estimation via Stochastic Batch Normalization”. In: *Advances in Neural Networks – ISNN 2019*. Ed. by Huchuan Lu, Huajin Tang, and Zhanshan Wang. Cham: Springer International Publishing, 2019, pp. 261–269 (cit. on pp. 138, 139).

[Blu19] Hermann Blum, Paul-Edouard Sarlin, Juan Nieto, et al. “Fishyscapes: A Benchmark for Safe Semantic Segmentation in Autonomous Driving”. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) Workshops*. Oct. 2019 (cit. on pp. 137–139, 143, 145).

[Hei19] Matthias Hein, Maksym Andriushchenko, and Julian Bitterwolf. “Why ReLU Networks Yield High-Confidence Predictions Far Away From the Training Data and How to Mitigate the Problem”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2019 (cit. on pp. 138, 139).

[Hen19a] Dan Hendrycks, Mantas Mazeika, and Thomas Dietterich. “Deep Anomaly Detection with Outlier Exposure”. In: *Proceedings of the International Conference on Learning Representations* (2019) (cit. on pp. 138, 139, 143).

[Hen19b] Dan Hendrycks et al. “A Benchmark for Anomaly Segmentation”. In: *arXiv preprint arXiv:1911.11132* (2019) (cit. on p. 137).

[Lis19] Krzysztof Lis et al. “Detecting the Unexpected via Image Resynthesis”. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*. Oct. 2019 (cit. on pp. 137, 140, 145).

[Nal19] Eric Nalisnick et al. “Do Deep Generative Models Know What They Don’t Know?” In: *International Conference on Learning Representations*. 2019 (cit. on p. 139).

[Ren19] Jie Ren, Peter J. Liu, Emily Fertig, et al. “Likelihood Ratios for Out-of-Distribution Detection”. In: *Advances in Neural Information Processing Systems 32*. Curran Associates, Inc., 2019, pp. 14707–14718. URL: http://papers.nips.cc/paper/9611-likelihood-ratios-for-out-of-distribution-detection.pdf (cit. on p. 139).

[Rot19] Matthias Rottmann and Marius Schubert. "Uncertainty Measures and Prediction Quality Rating for the Semantic Segmentation of Nested Multi Resolution Street Scene Images". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*. June 2019 (cit. on pp. 138, 142).

[Zha19] Li Zhang et al. "Dual Graph Convolutional Network for Semantic Segmentation." In: *Proceedings of the British Machine Vision Conference (BMVC)*. 2019. URL: `https://bmvc2019.org/wp-content/uploads/papers/0089-paper.pdf` (cit. on p. 145).

[Zhu19] Yi Zhu, Karan Sapra, Fitsum A. Reda, et al. "Improving Semantic Segmentation via Video Propagation and Label Relaxation". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2019 (cit. on pp. 137, 142, 145).

[Brü20] Dominik Brüggemann et al. "Detecting Out of Distribution Objects in Semantic Segmentation of Street Scenes". In: *The 30th European Safety and Reliability Conference (ESREL)*. 2020 (cit. on p. 139).

[Cha20a] Robin Chan et al. "Application of Maximum Likelihood Decision Rules for Handling Class Imbalance in Semantic Segmentation". In: *The 30th European Safety and Reliability Conference (ESREL)*. 2020 (cit. on p. 145).

[Cha20b] Robin Chan et al. "Controlled False Negative Reduction of Minority Classes in Semantic Segmentation". In: *2020 IEEE International Joint Conference on Neural Networks (IJCNN)*. 2020 (cit. on pp. 138, 142).

[Jan20] Joel Janai et al. "Computer vision for autonomous vehicles: Problems, datasets and state of the art". In: *Foundations and Trends® in Computer Graphics and Vision* 12.1–3 (2020), pp. 1–308 (cit. on pp. 137, 140).

[Jou20] Nicolas Jourdan, Eike Rehder, and Uwe Franke. "Identification of Uncertainty in Artificial Neural Networks". In: *Proceedings of the 13th Uni-DAS e.V. Workshop Fahrerassistenz und automatisiertes Fahren*. July 2020 (cit. on p. 139).

[Maa20] Kira Maag, Matthias Rottmann, and Hanno Gottschalk. "Time-Dynamic Estimates of the Reliability of Deep Semantic Segmentation Networks". In: *IEEE International Conference on Tools with Artificial Intelligence (ICTAI)*. Nov. 2020 (cit. on pp. 138, 142).

[Meh20] A. Mehrtash et al. "Confidence Calibration and Predictive Uncertainty Estimation for Deep Medical Image Segmentation". In: *IEEE Transactions on Medical Imaging* (2020), pp. 1–1 (cit. on p. 139).

[Mei20] Alexander Meinke and Matthias Hein. "Towards neural networks that provably know when they don't know". In: *International Conference on Learning Representations*. 2020. URL: `https://openreview.net/forum?id=ByxGkySKwH` (cit. on pp. 138, 139, 143).

[Obe20] Philipp Oberdiek, Matthias Rottmann, and Gernot A. Fink. "Detection and Retrieval of Out-of-Distribution Objects in Semantic Segmentation". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*. June 2020 (cit. on p. 139).

[Rot20] Matthias Rottmann, Pascal Colling, Thomas Paul Hack, et al. "Prediction Error Meta Classification in Semantic Segmentation: Detection via Aggregated Dispersion Measures of Softmax Probabilities". In: *2020 IEEE International Joint Conference on Neural Networks (IJCNN)*. 2020 (cit. on pp. 138, 142).

[Wan20] Jingdong Wang, Ke Sun, Tianheng Cheng, et al. "Deep High-Resolution Representation Learning for Visual Recognition". In: *IEEE transactions on pattern analysis and machine intelligence* PP (Apr. 2020). ISSN: 0162-8828. DOI: 10.1109/tpami.2020.2983686. URL: https://doi.org/10.1109/TPAMI.2020.2983686 (cit. on p. 137).

[Col21] Pascal Colling et al. "MetaBox+: A new Region Based Active Learning Method for Semantic Segmentation using Priority Maps". In: *Proceedings of the 10th International Conference on Pattern Recognition Applications and Methods - Volume 1: ICPRAM*. SciTePress, 2021, pp. 51–62. ISBN: 978-989-758-486-2. DOI: 10.5220/0010227500510062. eprint: 2010.01884. URL: https://www.scitepress.org/PublicationsDetail.aspx?ID=r/nRUS5tLNg=&t=1 (cit. on p. 137).

SegmentMeIfYouCan: A Benchmark for Anomaly Segmentation

Robin Chan[1], Krzysztof Lis[2], Svenja Uhlemeyer[1], Hermann Blum[3], Sina Honari[2], Roland Siegwart[3], Pascal Fua[2], Mathieu Salzmann[2] and Matthias Rottmann[1]

[1]Stochastics Group, IZMD, University of Wuppertal, Wuppertal, Germany
[2]Computer Vision Laboratory, EPFL, Lausanne, Switzerland
[3]Autonomous Systems Lab, ETH, Zürich, Switzerland

Abstract. State-of-the-art semantic or instance segmentation deep neural networks (DNNs) are usually trained on a closed set of semantic classes. As such, they are ill-equipped to handle previously-unseen objects. However, detecting and localizing such objects is crucial for safety-critical applications such as perception for automated driving, especially if they appear on the road ahead. While some methods have tackled the tasks of anomalous or out-of-distribution object segmentation, progress remains slow, in large part due to the lack of solid benchmarks; existing datasets either consist of synthetic data, or suffer from label inconsistencies. In this paper, we bridge this gap by introducing the "SegmentMeIfYouCan" benchmark. Our benchmark addresses two tasks: Anomalous object segmentation, which considers any previously-unseen object category; and road obstacle segmentation, which focuses on any object on the road, may it be known or unknown. We provide two corresponding datasets together with a test suite performing an in-depth method analysis, considering both established pixel-wise performance metrics and recent component-wise ones, which are insensitive to object sizes. We empirically evaluate multiple state-of-the-art baseline methods, including several models specifically designed for anomaly / obstacle segmentation, on our datasets and on public ones, using our test suite. The anomaly and obstacle segmentation results show that our datasets contribute to the diversity and difficulty of both data landscapes.

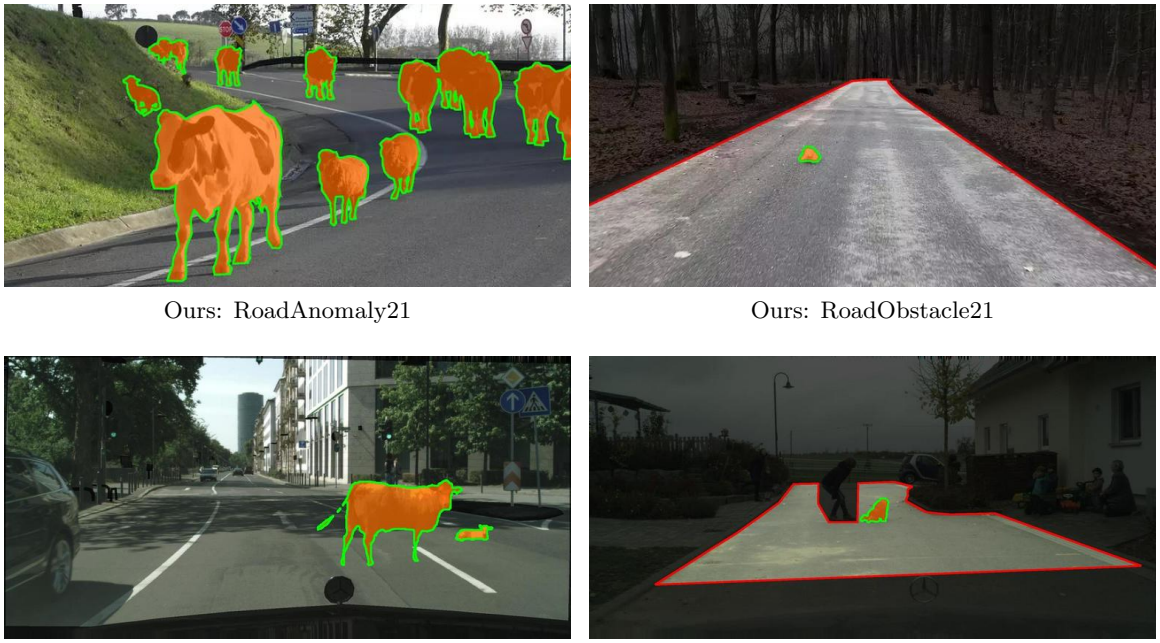
I Introduction

The advent of high-quality publicly-available datasets, such as Cityscapes [Cor16], BDD100k [Yu20], A2D2 [Gey19] and COCO [Lin14] has hugely contributed to the progress in semantic segmentation. However, while state-of-the-art deep neural networks (DNNs) yield outstanding performance on these datasets, they typically provide predictions for a closed set of semantic classes. Consequently, they are unable to classify an object as *none of the known categories* [Zha17]. Instead, they tend to be overconfident in their predictions, even in the presence of previously-unseen objects [Hei19], which precludes the use of uncertainty to identify the corresponding anomalous regions.

Nevertheless, reliability in the presence of unknown objects is key to the success of applications that have to face the diversity of the real world, *e.g.*, perception in automated driving. This has motivated the creation of benchmarks such as Fishyscapes [Blu19] or CAOS [Hen20]. While these benchmarks have enabled interesting experiments, the limited real-world diversity in Fishyscapes, the lack of a public leader board and of a benchmark suite in CAOS, and the reliance on synthetic images in both benchmarks hinder proper evaluation of and comparisons between the state-of-the-art methods.

In this paper, motivated by the limitations of existing anomaly segmentation datasets and by the emerging body of works in this direction [Iso17; Ang19; Blu19; Lis19; Brü20; Jou20; Meh20; Obe20; Cha21], we introduce the *SegmentMeIfYouCan*[2] benchmark. It is accompanied with two datasets, consisting of diverse and manually annotated real images, a public leader board and an evaluation

[2]`https://www.segmentmeifyoucan.com/`



Ours: RoadAnomaly21 | Ours: RoadObstacle21

Fishyscapes | LostAndFound

Figure 1: Comparison of images from our and existing public datasets. Anomalies / obstacles are highlighted in orange, darkened regions are excluded from the evaluation. In RoadAnomaly21, anomalies may appear everywhere in the image. In contrast to Fishyscapes, where anomalous objects are synthetic, all RoadAnomaly21 images are real. In RoadObstacle21, the region of interest is restricted to the drivable area with obstacles ahead. This is comparable to LostAndFound, where the labeling, however, is not always consistent, *e.g.* children are anomalies but other humans not.

suite, providing in-depth analysis and comparisons, to facilitate the development of road anomaly segmentation methods.

Our benchmark encompasses two separate tasks. The first one consists of strict anomaly segmentation, where any previously-unseen object is considered as an anomaly. Furthermore, motivated by the observation that the boundary between known and unknown classes can sometimes be fuzzy, for instance for *car vs. van*, we introduce the task of obstacle segmentation, whose goal is to identify all objects on the road, may they be from known classes or from unknown ones.

For the anomaly track, we provide a dataset of 100 images with pixel-wise annotations over two classes (anomaly, not anomaly) and a void class, which, in analogy to Cityscapes, signals the pixels that are excluded from the evaluation. We consider any object that strictly cannot be seen in the Cityscapes data as anomalous, appearing anywhere in the image. For the obstacle track, our dataset contains 327 images with analogous annotation (obstacle, not obstacle, void), and focuses only on the road as region of interest. The focus in this track is of more practical need, *e.g.* for automated driving systems, targeting obstacles that may cause hazardous street situations, see Figure 1. All images of our datasets are publicly available for download, together with a benchmark suite that computes both established pixel-wise metrics and recent component-wise ones.

In the remainder of this paper, we first review existing anomaly detection datasets, methods and evaluation metrics in more detail. We then describe our new benchmark and provide extensive experiments comparing state-of-the-art road anomaly / obstacle segmentation methods on our datasets and on other related ones, showing the difficulty of the models on the proposed benchmarks.

II Related Work

In this section we first review previous datasets for anomaly detection, with some of them being designed for road anomaly segmentation. Then we briefly describe some of the methods on anomaly and obstacle segmentation.

II.i Datasets and Benchmarks

Existing methods for anomaly detection have often been evaluated on their ability to separate images from two different source distributions, such as separating MNIST from FashionMNIST [Cho18; Ame20; Mei20], NotMNIST [Ame20], or Omniglot [Lak15], and separating CIFAR-10 from SVHN [Lee18; Ame20; Mei20] or LSUN [Lee18; Lia18; Mei20]. Such experiments can be found in many works, including [Hen17; Cho18; Lee18; Lia18; Ame20; Mei20].

For semantic segmentation, a similar task was therefore proposed by the WildDash benchmark [Zen18] that analyzes semantic segmentation methods trained for driving scenes on a range of failure sources, including full-image anomalies, such as images from the beach. In our work, by contrast, we focus on the problem of robustness to anomalies that only cover a small portion of the image, and on the methods that aim to segment such anomalies, *i.e.* method for the task of *anomaly segmentation*.

One prominent dataset tackling the task of anomaly segmentation is LostAndFound [Pin16], which shares the same setup as Cityscapes [Cor16] but includes anomalous objects / obstacles in various street scenes in Germany. LostAndFound contains 9 different object types as anomalies, and only has annotations for the anomaly and the road surface. Furthermore, it considers children and bicycles as anomalies, even though they are part of the Cityscapes training set, and it contains several labeling mistakes. Although we filter and refine LostAndFound in this work[3], similar to Fishyscapes [Blu19], the low diversity of anomalies persists.

The CAOS BDD-Anomaly benchmark [Hen20] suffers from a similar low-diversity issue, arising from its use of only 3 object classes sourced from the BDD100k dataset [Yu20] as anomalies (besides including several labeling mistakes). Both Fishyscapes and CAOS try to mitigate this low diversity by complementing their real images with synthetic data. Such synthetic data, however, is not realistic and not representative of the situations that can arise in the real world.

In general, the above works illustrate the shortage of diverse real-world data for anomaly segmentation. Additional efforts in this regard have been made by sourcing and annotating images of animals in street scenes [Lis19], and by leveraging multiple sensors, including mainly LiDAR, to detect obstacles on the road [Sin20]. In any event, most of the above datasets are fully published with annotations, allowing methods to overfit to the available anomalies. Furthermore, apart from Fishyscapes, we did not find any public leader boards that allow for a trustworthy comparison of new methods. To provide a more reliable test setup, we do not share the labels and request predictions of the shared images to be submitted to our servers. Furthermore, we provide a leader board, which we publish alongside two novel real-world datasets, namely RoadAnomaly21 and RoadObstacle21. A summary of the main properties of the mentioned datasets is given in Table 1. Our main contribution in both proposed datasets is the diversity of the anomaly categories and of the scenes.

In RoadAnomaly21, anomalies can appear anywhere in the image, which is comparable to Fishyscapes LostAndFound [Blu19] and CAOS BDD-Anomaly [Hen20]. Although the latter two datasets are larger, their images only show a limited diversity of anomaly types and scenes because they are usually frames of videos captured in single scenes. By contrast, in our dataset every image shows a unique scene,

[3] In the following, we refer to the LostAndFound subset without the images of children, bicycles and invalid annotations as "LostAndFound-NoKnown".

Dataset	anomaly pixels	non-anomaly pixels	diverse scenes	different anomalies	dataset size	ground truth (gt) components	mean & std of gt size relative to image size
Fishyscapes LostAndFound val [Blu19]	0.23%	81.13%	12	7	373	165	0.13% ± 0.23%
CAOS BDD-Anomaly test [Hen20]	0.83%	81.28%	810	3	810	1231	0.55% ± 1.84%
Ours: RoadAnomaly21 test	13.83%	82.17%	100	26	100	262	4.12% ± 7.29%
LostAndFound test (NoKnown) [Pin16]	0.12%	15.31%	13 (12)	9 (7)	1203 (1043)	1864 (1709)	0.08% ± 0.16%
LiDAR guided Small Obstacle test [Sin20]	0.07%	36.09%	2	6	491	1203	0.03% ± 0.07%
Ours: RoadObstacle21 test	0.12%	39.08%	8	31	327	388	0.10% ± 0.25%

Table 1: Main properties of comparable real-world anomaly (top three rows) and obstacle (bottom three rows) segmentation datasets. Our main contribution is the diversity of the anomaly (or obstacle) categories and of the scenes. Note that "void" pixels are not included in this table.

with at least one out of 26 different types of anomalous objects and each sample widely differs in size, ranging from 0.5% to 40% of the image.

In RoadObstacle21, all anomalies (or obstacles) appear on the road, making this dataset comparable to LostAndFound [Pin16] and the LiDAR guided Small Obstacle dataset [Sin20]. Again, the latter two datasets contain more images than ours, however, the high numbers of images result from densely sampling frames from videos. Consequently, those two datasets lack in object diversity (9 and 6 categories, respectively, compared to 31 in our dataset). Furthermore, the videos are recorded under perfect weather conditions, while RoadObstacle21 shows scenes in diverse situations, including night, dirty roads and snowy conditions.

II.ii Anomaly and Obstacle Segmentation

Anomaly detection was initially tackled in the context of image classification, by developing post-processing techniques aiming to adjust the confidence values produced by a classification DNN [Hen17; Lee18; Lia18; Hei19; Mei20]. Although originally designed for image-level anomaly detection, most of these methods can easily be adapted to anomaly segmentation [Ang19; Blu19] by treating each individual pixel in an image as a potential anomaly.

Another relevant approach consists of estimating the uncertainty of the predictions, leveraging the intuition that anomalous image regions should correlate with high uncertainty. One way of doing so is Bayesian (deep) learning [Mac92; Nea12], where the model parameters are treated as distributions. Because of the computational complexity, approximations to Bayesian inference have been developed [Gal16; Lak17; Ata19; Gus20] and extended to semantic segmentation [Bad17; Ken17; Muk19]. Instead of reasoning about uncertainty, other non-Bayesian approaches tune previously-trained models to the task of anomaly detection by either modifying its architecture or exploiting additional data. For example, in [DeV18], anomaly scores are learned by adding a separate branch to the DNN. Instead of modifying the DNNs's architecture, other approaches [Hen19; Mei20] incorporate an auxiliary "out-of-distribution" (OoD) dataset during training, which is disjoint from the actual training dataset. These ideas have been employed for anomaly segmentation in [Bev19; Jou20; Cha21].

A recent line of work performs anomaly segmentation via generative models that reconstruct / resynthesize the original input image. The intuition is that the reconstructed images will better preserve the appearance of regions containing known objects than those with unknown ones. Pixel-wise anomaly detection is then performed by identifying the discrepancies between the original and reconstructed image. This approach has been used not only for anomaly segmentation [Lis19; Xia20; Di 21] but also specifically for road obstacle detection [Cre15; Mun17; Lis20].

It is important to note that there are some related works with different definitions of anomaly segmentation. For example, [Ber19] evaluates the segmentation of industrial production anomalies like scratches, and in medical contexts anomaly segmentation can be understood as the detection of

diseased parts on e.g. tomography images [See20] or brain MRIs [Bau21]. What we define as anomaly segmentation will be discussed in detail in the next Section III.

III Benchmark Description

The aim of our benchmark is two-fold. On one hand, by providing diverse data with consistent annotations, we seek to facilitate progress in general semantic anomaly segmentation research. On the other hand, by focusing on road scenes, we expect our benchmark to accelerate the progress towards much needed segmentation/obstacle-detection methods for safe automated driving.

To achieve these goals, our benchmark covers two tasks. First, it tackles the general problem of anomaly segmentation, aiming to identify the image regions containing object classes that have never been seen during training, and thus for which semantic segmentation cannot be correct. This is necessary for any reliable decision making process and it is of great importance to many computer vision applications. Note that, in accordance to [Blu19; Hen20], we define anomaly as objects that do not fit any of the class definitions in the training data. In some works, anomaly may be used to describe visually different inputs like *e.g.* a car in a novel color, which does not fit our definition.

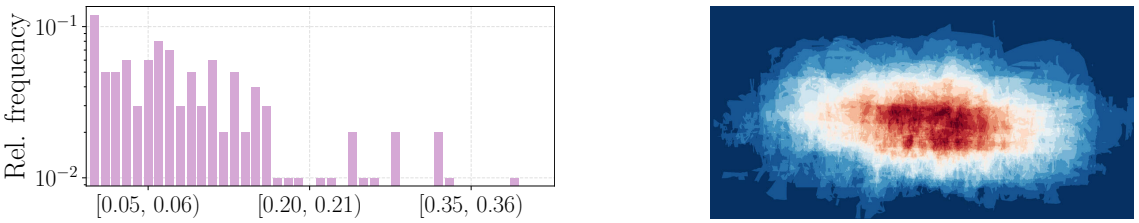
This strict definition of semantic anomalies, however, can sometimes be ill-defined because (i) existing semantic segmentation datasets, such as Cityscapes [Cor16], often contain ambiguous and ignored regions (annotated as *void*), which are not strictly anomalies since they are seen during training; (ii) the boundary of some classes is fuzzy, *e.g.*, cars *vs*. vans *vs*. rickshaws, making it unclear whether some regions should be considered as anomalous or not. To address these issues, and to account for the fact that automated driving systems need to make sure that the road ahead is free of *any* hazardous objects, we further incorporate obstacle segmentation as a second task in our benchmark, whose goal is to identify any non-drivable region on the road, may the non-drivable region correspond to a known object class or an unknown one.

III.i Benchmark Tracks and Datasets

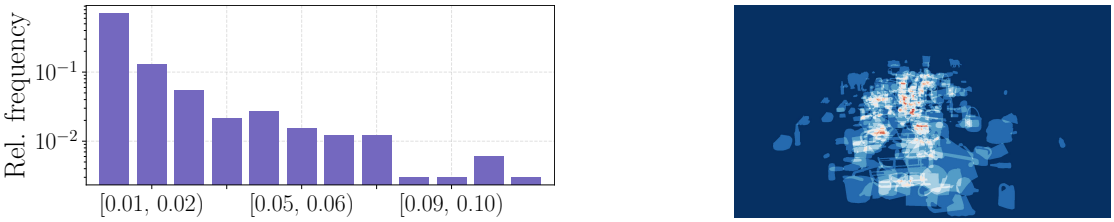
We now present the two tracks in our benchmark, corresponding to the two tasks discussed above. Each track contains its own dataset with different properties and is therefore evaluated separately in our benchmark suite. An overview comparing our datasets to related public ones is given in Table 1.

RoadAnomaly21. The road anomaly track benchmarks general anomaly segmentation in full street scenes. It consists of an evaluation dataset of 100 images with pixel-level annotations. The data is an extension of the one introduced in [Lis19], now including a broader collection of images and finer-grain labeling. In particular, we removed low quality images and ones lacking clear road scenes. Besides, we removed labeling mistakes, added the void class and included 68 newly collected images. Each image contains at least one anomalous object, *e.g.*, an animal or an unknown vehicle. The anomalies can appear anywhere in the image, which were collected from web resources and therefore depict a wide variety of environments. The distribution of object sizes and location is shown in Figure 2 (a). Moreover, we provide 10 additional images with annotations such that users can check the compatibility of their methods with our benchmark implementation.

RoadObstacle21. The road obstacle track focuses on safety for automated driving. The objects to segment in the evaluation data always appear on the road ahead, *i.e.* they represent realistic and hazardous obstacles that are critical to detect. Our dataset consists of 222 new images taken by ourselves and 105 from [Lis20], summing up to a total of 327 evaluation images with pixel-level annotations. The region of interest in these images is given by the road, which is assumed to belong to the known



(a) RoadAnomaly21



(b) RoadObstacle21

Figure 2: Relative frequency of annotated anomaly / obstacle pixels within an image over the 100 images in the RoadAnomaly21 test dataset (a) and the 327 images in the RoadObstacle21 test dataset (b), respectively. Here, the fraction of anomaly / obstacle pixels serves as a proxy for the size of the objects of interest within an image. Note that the y-axes of the histograms are log scaled.

classes on which the algorithm was trained. The obstacles in this dataset are chosen such that they all can be understood as anomalous objects as well, *e.g.*, stuffed toys, sleighs or tree stumps. They appear at different distances (one distance per image) and are surrounded by road pixels. This allows us to focus our evaluation on the obstacles, as other objects lie outside the region of interest. The distribution of object sizes and location is shown in Figure 2 (b). Moreover, this dataset incorporates different road surfaces, lighting and weather conditions, thus encompassing a broad diversity of scenes. An extra track of additional 85 images with scenes at night and in extreme weather, such as snowstorms, is also available. However, the latter subset is excluded from our numerical experiments due to the significant domain shift. Lastly, we provide 30 additional images with annotations such that users can check the compatibility of their methods with our benchmark implementation.

Labeling Policy. In both datasets, the pixel-level annotations include three classes: **1)** anomaly / obstacle, **2)** not anomaly / not obstacle, and **3)** void.

In RoadAnomaly21, the 19 Cityscapes evaluation classes [Cor16], on which most semantic segmentation DNNs are trained, serve as basis to judge whether an object is considered anomalous or not. Everything that fits in the class definitions of Cityscapes is thus labeled as *not anomaly*. This track focuses on the detection of objects which are semantically different from those in the Cityscapes training data. Therefore, if image regions cannot be clearly assigned to any of the Cityscapes classes, they are labeled as *anomaly*. The objects, which are not the main anomalies of interest in the context of street scenes, are labeled as *void* and excluded from our evaluation. The latter class include, for instance, mountains or water in the image background, and street lights. In ambiguous cases, which *e.g.* can arise from a strong domain shift to Cityscapes, we assign the void class as well to properly evaluate semantic anomaly segmentation.

In RoadObstacle21, the task is defined as distinguishing between drivable area and non-drivable

area. The goal is to make sure that the road ahead of the ego-car is free of any hazard, irrespective of the object category of potential obstacles. Therefore, the drivable area is labeled as *not obstacle*. This class particularly also includes regions on the road, which visually differ from the rest of the road. Moreover, every object, which is visually enclosed in the drivable area, is labeled as *obstacle*. All image regions outside the road are assigned to the *void* class and ignored in the evaluation.

III.ii Performance Metrics

For the sake of brevity, in what follows we refer to both anomalies and obstacles as *anomalies*.

Pixel level. Let $\mathcal{Z}$ denote the set of image pixel locations. A model with a binary classifier providing anomaly scores $s(x) \in \mathbb{R}^{|\mathcal{Z}|}$ for an image $x \in \mathcal{X}$ (from a dataset $\mathcal{X} \subseteq [0,1]^{N \times |\mathcal{Z}| \times 3}$ of N images) discriminates between the two classes *anomaly* and *non-anomaly*. We evaluate the separability of the pixel-wise anomaly scores via the area under the precision-recall curve (AuPRC), where precision and recall are considered as functions of some threshold $\delta \in \mathbb{R}$ applied to $s(x)\ \forall\ x \in \mathcal{X}$. The AuPRC puts emphasis on detecting the minority class, making it particularly well suited as our main pixel-wise evaluation metric since the pixel-wise class distributions of RoadAnomaly21 and RoadObstacle21 are considerably unbalanced, *cf*. Table 1.

To consider the safety point of view, we also include the false positive rate at 95% true positive rate (FPR_{95}) in our evaluation. The FPR_{95} metric indicates how many false positive predictions must be made to reach the desired true positive rate. Note that, any prediction which is contained in a ground-truth labeled region of the class void is not counted as false positive, *cf*. Section III.i. In particular for the RoadObstacle21 dataset the evaluation is therefore restricted to the road area.

Component level. From a practitioner's perspective, it is very important to detect all anomalous regions in the scene, regardless of their size, *i.e.*, the number of pixels they cover. However, pixel-level metrics may neglect small anomalies. While one could thus focus on object detection metrics, the notion of individual objects is in fact not relevant for anomaly (region) detection. To satisfy these requirements, we therefore consider performance metrics acting at the component level.

The main metrics for component-wise evaluation are the numbers of *true-positives* (TP), *false-negatives* (FN) and *false-positives* (FP). Considering anomalies as the positive class, we use a component-wise localization and classification quality measure to define the TP, FN and FP components. Specifically, we define this measure as an adjusted version of the component-wise intersection over union (sIoU), introduced in [Rot20]. In particular, while in [Rot20] the sIoU is computed for predicted components, we consider the sIoU for ground-truth components to compute TP and FN. To compute FP, we employ the positive predictive value (PPV, or component-wise precision) for predicted components as quality measure. We discuss the definitions of these quantities in more detail below.

Let $\mathcal{Z}_c$ be the set of pixel locations labeled with class $c =$ "anomaly" in the dataset $\mathcal{X}$. We consider a connected component of pixels (where the 8 pixels surrounding pixel z in image $x \in \mathcal{X}$ are taken to be its neighbors) that share the same class label as a *component*. Then, let us denote by $\mathcal{K} \subseteq \mathcal{P}(\mathcal{Z}_c)$, with $\mathcal{P}(\mathcal{S})$ the power set of a set $\mathcal{S}$, the set of anomaly components according to the ground truth, and by $\hat{\mathcal{K}} \subseteq \mathcal{P}(\mathcal{Z}_c)$ the set of components predicted to be anomalous by some machine learning model.

Formally, the sIoU is a mapping $\text{sIoU} : \mathcal{K} \to [0,1]$. For $k \in \mathcal{K}$, it is defined as

$$\text{sIoU}(k) := \frac{|k \cap \hat{K}(k)|}{|(k \cup \hat{K}(k)) \setminus \mathcal{A}(k)|} \qquad \text{with} \qquad \hat{K}(k) = \bigcup_{\substack{\hat{k} \in \hat{\mathcal{K}} \\ \hat{k} \cap k \neq \emptyset}} \hat{k} \tag{1}$$

Figure 3: Illustration of the ordinary component-wise intersection over union (IoU) and the adjusted one (sIoU). In both examples above, the prediction $\hat{k}$ (blue rectangle) is the same but covers different targets (green rectangles). On the left, both IoU and sIoU yield the same score. On the right, IoU punishes the prediction as it does not cover each object precisely. By contrast, sIoU checks how much the predictions cover the ground-truth regions, independently of whether prediction/ground truth belongs to a single or multiple objects. In automated driving, it is more important to detect all anomalous regions (whether they belong to single or multiple objects), rather than to detect each object precisely. Since two targets are separated by at least one pixel, IoU = sIoU = 1 if and only if the prediction covers one target perfectly.

and $\mathcal{A}(k) = \{z \in k' : k' \in \mathcal{K} \setminus \{k\}\}$. With the adjustment $\mathcal{A}(k)$, the pixels are excluded from the union if and only if they correctly intersect with another ground-truth component $k' \in \mathcal{K}(x)$, which is not equal to k. This may happen when one predicted component covers multiple ground-truth components, as illustrated in Figure 3. Given some threshold $\tau \in [0, 1)$, we then call a target $k \in \mathcal{K}$ TP if sIoU$(k) > \tau$, and FN otherwise.

For the other error type, *i.e.*, FP, we compute the PPV (or precision) for $\hat{k} \in \hat{\mathcal{K}}$, which is defined as

$$\mathrm{PPV}(\hat{k}) := \frac{|\hat{k} \cap K(\hat{k})|}{|\hat{k}|}\,, \tag{2}$$

We then call a predicted component $\hat{k} \in \hat{\mathcal{K}}$ FP if $\mathrm{PPV}(\hat{k}) \leq \tau$.

As an overall metric, we additionally include the component-wise F_1-score defined as

$$F_1(\tau) := \frac{2 \cdot \mathrm{TP}(\tau)}{2 \cdot \mathrm{TP}(\tau) + \mathrm{FN}(\tau) + \mathrm{FP}(\tau)} \in [0, 1]\,, \tag{3}$$

which summarizes the TP, FN and FP quantities (that depend on τ). The component-level metrics allow one to evaluate localization of objects irrespective of their size and hence big objects will not dominate these metrics. In addition, while object detection metrics punish predictions that cover multiple ground-truth objects or vice-versa, our component-level metric does not do so, *cf*. Figure 3.

III.iii Evaluated Methods

Several anomaly segmentation methods have already been evaluated on our benchmark and constitute our initial leader board. We evaluate at least one method per type discussed in Section II.ii, namely

- *Methods originating from image classification*: **maximum softmax probability** [Hen17], **ODIN** [Lia18], **Mahalanobis distance** [Lee18];
- *Bayesian model uncertainty*: **Monte Carlo (MC) dropout** [Muk19], **ensemble** [Lak17];
- *Learning to identify anomalies*: **learned embedding density** [Blu19], **void classifier** [Blu19], **maximized softmax entropy** [Cha21];
- *Reconstruction via generative models*: **image resynthesis** [Lis19], **SynBoost** [Di 21] and **road inpainting** (obstacle track only) [Lis20].

All methods have an underlying semantic segmentation DNN trained on Cityscapes and provide pixel-wise anomaly scores. A semantic segmentation DNN trained on Cityscapes is also our recommendation

		Pixel-level			Component-level											
	requires	Anomaly scores			$k \in \mathcal{K}$	$\hat{k} \in \hat{\mathcal{K}}$	$\tau = 0.25$			$\tau = 0.50$			$\tau = 0.75$			
Method	OoD data	AuPRC ↑	FPR_{95} ↓	F_1^* ↑	$\overline{\text{sIoU}}$ ↑	$\overline{\text{PPV}}$ ↑	FN ↓	FP ↓	F_1 ↑	FN ↓	FP ↓	F_1 ↑	FN ↓	FP ↓	F_1 ↑	$\overline{F_1}$ ↑
Maximum softmax [Hen17]	✗	28.0	72.0	34.2	15.5	15.3	204	681	11.6	233	714	5.8	256	744	1.2	5.9
ODIN [Lia18]	✗	33.1	71.7	39.1	19.6	17.9	181	924	12.8	226	985	5.6	254	1043	1.2	6.0
Mahalanobis [Lee18]	✗	20.0	87.0	31.9	14.8	10.2	206	1433	6.4	241	1478	2.4	257	1512	0.6	2.9
MC dropout [Muk19]	✗	28.9	69.5	39.0	20.5	17.3	175	1320	10.4	225	1391	4.4	252	1459	1.2	4.9
Ensemble [Lak17]	✗	17.7	91.1	27.8	16.4	20.8	197	1454	7.3	233	1511	3.2	254	1553	0.9	3.4
Void classifier [Blu19]	✓	36.8	63.5	44.3	21.1	22.1	181	797	14.2	219	845	7.5	253	879	1.6	7.6
Embedding density [Blu19]	✗	37.5	70.8	48.7	33.8	20.5	107	1437	16.7	176	1485	9.4	250	1592	1.3	9.2
Image resynthesis [Lis19]	✗	52.3	25.9	60.5	39.5	11.0	95	1187	20.7	153	1225	13.7	230	1294	4.0	12.9
SynBoost [Di 21]	✓	56.4	61.9	58.0	35.0	18.3	109	1062	20.7	178	1114	11.5	247	1216	2.0	11.5
Maximized entropy [Cha21]	✓	**85.5**	**15.0**	**77.4**	**49.2**	**39.5**	**85**	**413**	**41.5**	**115**	**421**	**35.4**	**163**	**439**	**24.8**	**34.5**

Table 2: Benchmark results for our RoadAnomaly21 dataset. This dataset contains 262 ground-truth components in total. The main performance metrics are highlighted with gray columns.

		Pixel-level			Component-level											
	requires	Anomaly (obstacle) scores			$k \in \mathcal{K}$	$\hat{k} \in \hat{\mathcal{K}}$	$\tau = 0.25$			$\tau = 0.50$			$\tau = 0.75$			
Method	OoD data	AuPRC ↑	FPR_{95} ↓	F_1^* ↑	$\overline{\text{sIoU}}$ ↑	$\overline{\text{PPV}}$ ↑	FN ↓	FP ↓	F_1 ↑	FN ↓	FP ↓	F_1 ↑	FN ↓	FP ↓	F_1 ↑	$\overline{F_1}$ ↑
Maximum softmax [Hen17]	✗	15.7	16.6	22.5	19.7	15.9	255	1494	13.2	326	1503	6.3	372	1517	1.7	6.9
ODIN [Lia18]	✗	21.2	15.4	29.2	20.7	18.5	260	1072	16.1	312	1079	9.9	362	1093	3.5	10.0
Mahalanobis [Lee18]	✗	20.9	13.1	25.8	14.0	21.8	293	1101	12.0	352	1104	4.7	385	1116	0.4	5.5
MC dropout [Muk19]	✗	3.7	50.6	8.0	6.3	5.8	351	2782	2.3	375	2784	0.8	386	2790	0.1	1.0
Ensemble [Lak17]	✗	1.1	77.2	3.1	8.6	4.7	335	3758	2.5	365	3768	1.1	382	3782	0.3	1.3
Void classifier [Blu19]	✓	9.2	41.5	23.4	6.3	20.3	350	350	9.8	365	350	6.0	381	353	1.9	5.9
Embedding density [Blu19]	✗	0.8	46.4	2.0	35.6	2.9	145	10972	4.2	244	11037	2.5	370	11191	0.3	2.4
Image resynthesis [Lis19]	✗	37.2	4.7	38.8	16.6	20.5	286	743	16.5	334	773	8.9	374	824	2.3	9.5
Road inpainting [Lis20]	✗	52.6	47.1	67.5	**57.6**	39.5	**79**	580	48.4	**131**	586	41.8	**240**	611	25.8	40.2
SynBoost [Di 21]	✓	70.3	3.1	70.1	44.3	41.8	133	352	51.3	185	363	42.6	286	414	22.6	40.4
Maximized entropy [Cha21]	✓	**85.1**	**0.8**	**79.6**	47.9	**62.6**	136	**151**	**63.7**	177	**158**	**55.7**	247	**174**	**40.1**	**54.2**

Table 3: Benchmark results for our RoadObstacle21 dataset. This dataset contains 388 ground-truth components in total. The main performance metrics are highlighted with gray columns.

as underlying model, however, we leave it up to the participants which network and training data they use. Furthermore, some evaluated methods additionally employ out-of-distribution (OoD) data to tune the anomaly detector. For our set of methods, this would be any data with labels semantically different from the Cityscapes train classes. OoD data is also allowed to be used to alleviate the effects of a potential domain shift.

IV Numerical Experiments

In our benchmark suite we integrate a default method to generate the anomaly segmentation from pixel-wise anomaly scores. We choose the threshold δ^*, at which one pixel is classified as anomaly, by means of the optimal pixel-wise F_1-score, that we denote with F_1^*. Then, δ^* is computed as

$$\delta^* = \arg\max_{\delta \in \mathbb{R}} \frac{2 \cdot \text{precision}(\delta) \cdot \text{recall}(\delta)}{\text{precision}(\delta) + \text{recall}(\delta)} \,, \tag{4}$$

subject to $\text{precision}(\delta) + \text{recall}(\delta) > 0 \;\forall\; \delta$.

Moreover, for the anomaly track, components smaller than 500 pixels are discarded from the predicted segmentation, and for the obstacle track, components smaller than 50 pixels are discarded.

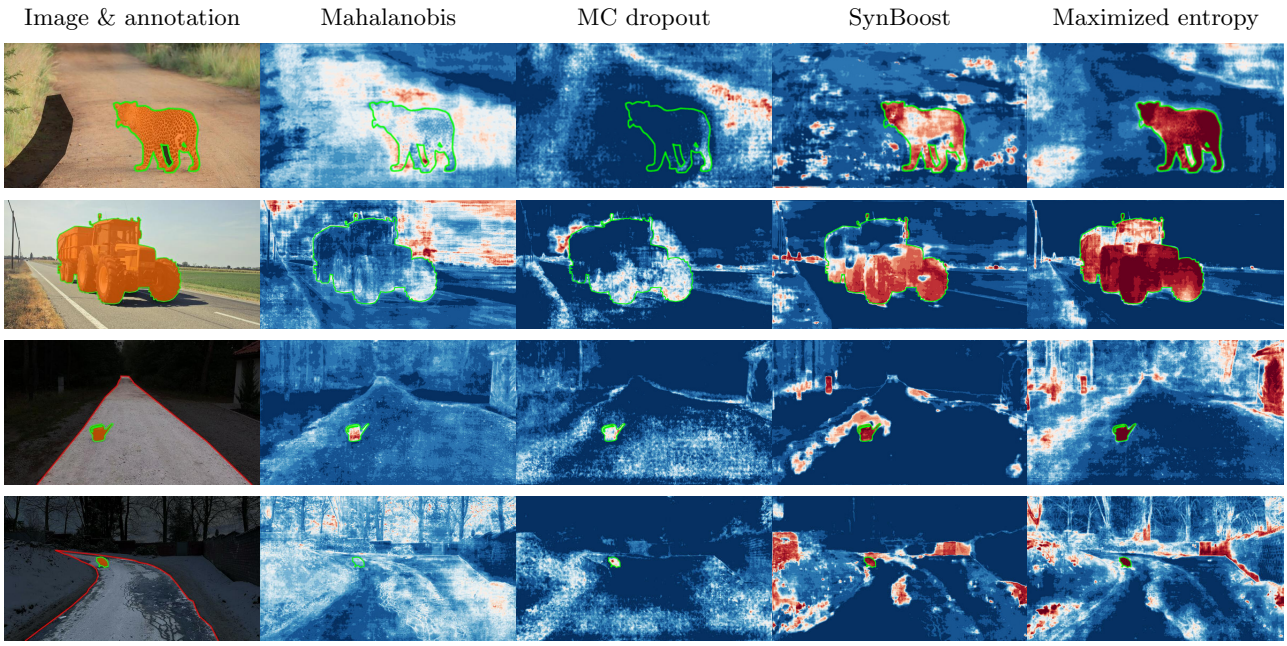


Figure 4: Qualitative comparison of the anomaly scores produced by the methods introduced in Section III.iii for example images of RoadAnomaly21 (top two rows) and examples image of RoadObstacle21 (bottom two rows). Here, red indicates higher anomaly / obstacle scores and blue lower ones. The ground-truth anomaly / obstacle component is indicated by green contours.

These sizes are chosen based on the smallest ground-truth components. All methods presented in Section III.iii produce anomaly scores for which we apply the default segmentation method. We emphasize that using our proposed default method for anomaly segmentation masks is completely optional. We allow and encourage competitors in the benchmark to submit their own anomaly segmentation masks generated via more sophisticated image operations.

In our evaluation, we additionally include the average sIoU per component $\overline{\text{sIoU}}$, which can be computed by averaging sIoU over all ground-truth components $k \in \mathcal{K}$. Analogously, we also include the average PPV per component $\overline{\text{PPV}}$ for all predicted components $\hat{k} \in \hat{\mathcal{K}}$. As the number of component-wise TP, FN and FP depends on some threshold τ for sIoU and PPV, respectively (see Section III.ii), we average these quantities over different thresholds $\tau \in \mathcal{T} = \{0.25, 0.30, \ldots, 0.75\}$, similarly to [Lin14], yielding the averaged component-wise F_1 score $\overline{F_1} = \frac{1}{|\mathcal{T}|} \sum_{\tau \in \mathcal{T}} F_1(\tau)$.

Discussion of the Results. Our benchmark results for RoadAnomaly21 and RoadObstacle21 are summarized in Table 2 and Table 3, respectively. In general, we observe that methods originally designed for image classification, including maximum softmax, ODIN, and Mahalanobis, do not generalize well to anomaly and obstacle segmentation. For methods based on statistics of the Cityscapes dataset, such as Mahalanobis as well as learned embedding density, anomaly detection is typically degraded by the presence of a domain shift. This results in a poor performance, particularly in RoadObstacle21, where various road surfaces can be observed. Interestingly, learned embedding density, MC dropout and the void classifier yield worse performance than maximum softmax on RoadObstacle21, whereas we observe the opposite on RoadAnomaly21.

The detection methods based on generative models, namely image resynthesis and SynBoost, appear to be better suited to both anomaly and obstacle segmentation at pixel as well as component

		RoadAnomaly		Fishyscapes LostAndFound Validation					RoadObstacle		LostAndFound test-NoKnown				
				Pixel-level		Component-level					Pixel-level		Component-level		
Method	OoD data	AuPRC ↑	$\overline{F_1}$ ↑	Anomaly scores AuPRC↑	FPR$_{95}$ ↓	$k \in \mathcal{K}$ $\overline{\text{sIoU}}$ ↑	$\hat{k} \in \hat{\mathcal{K}}$ $\overline{\text{PPV}}$ ↑	$\overline{F_1}$ ↑	AuPRC ↑	$\overline{F_1}$ ↑	Anomaly scores AuPRC ↑	FPR$_{95}$ ↓	$k \in \mathcal{K}$ $\overline{\text{sIoU}}$ ↑	$\hat{k} \in \hat{\mathcal{K}}$ $\overline{\text{PPV}}$ ↑	$\overline{F_1}$ ↑
Maximum softmax [Hen17]	✗	28.0	5.9	5.6	40.5	3.5	9.5	1.8	15.7	6.9	30.1	33.2	14.2	62.2	13.4
ODIN [Lia18]	✗	33.1	6.0	15.5	38.4	9.9	21.9	9.7	21.2	10.0	51.0	30.7	38.9	48.0	38.1
Mahalanobis [Lee18]	✗	20.0	2.9	32.9	**8.7**	19.6	29.4	19.2	20.9	5.5	55.0	12.9	33.8	31.7	24.6
MC dropout [Muk19]	✗	28.9	4.9	14.4	47.8	4.8	18.1	4.3	3.7	1.0	36.2	36.0	17.0	34.7	14.7
Ensemble [Lak17]	✗	17.7	3.4	0.3	90.4	3.1	1.1	0.4	1.1	1.3	2.9	82.0	6.7	7.6	2.7
Void classifier [Blu19]	✓	36.8	7.6	11.7	15.3	9.2	39.1	14.9	9.2	5.9	4.4	47.0	0.7	35.1	1.1
Embedding density [Blu19]	✗	37.5	9.2	8.9	42.2	5.9	10.8	4.9	0.8	2.4	61.7	10.4	37.8	35.2	30.8
Image resynthesis [Lis19]	✗	52.3	12.9	5.1	29.8	5.1	12.6	4.1	37.2	9.5	57.1	8.8	27.2	30.7	21.5
Road inpainting [Lis20]	✗	-	-	-	-	-	-	-	52.6	40.2	**83.0**	35.7	**49.2**	60.7	**56.9**
SynBoost [Di 21]	✓	56.4	11.5	**64.9**	30.9	**27.9**	**48.6**	**38.0**	70.3	40.4	81.8	**4.6**	37.2	**72.3**	53.0
Maximized entropy [Cha21]	✓	**85.5**	**34.5**	44.3	37.7	21.1	**48.6**	30.0	**85.1**	**54.2**	77.9	9.7	45.9	63.1	55.0

Table 4: Benchmark results for Fishyscapes LostAndFound validation and LostAndFound test-NoKnown, containing 165 and 1709 ground-truth components in total, respectively. In this table the main metrics, that are the pixel-wise AuPRC and the component-wise $\overline{F_1}$, from RoadAnomaly21 and RoadObstacle21 are additionally included for cross evaluation (gray columns, *cf*. Table 2 and Table 3).

level, clearly being superior to all the approaches discussed previously. This observation also holds for road inpainting in the obstacle track. These autoencoder-based methods are nonetheless limited by their discrepancy module, and they are outperformed in our experiments by maximized softmax entropy, which peaks at an AuPRC of 86% and a component-wise $\overline{F_1}$ of 49%. This highlights the importance of anomaly and obstacle proxy data. Illustrative example score maps produced by the discussed methods are shown in Figure 4.

In summary, the component-level evaluation highlights the methods' weaknesses even more clearly than the pixel-wise evaluation, the latter giving a stronger weight to larger anomalies and obstacles. All methods indeed tend to face difficulties in the presence of smaller anomalies and obstacles. In addition, we observe a much lower component-wise $\overline{F_1}$ score than a pixel-wise F_1^*, demonstrating the importance of evaluating at component level. The results w.r.t. the different categories of methods are challenging for models, hence leaving room for improvement.

Our benchmark suite enables a unified evaluation across different datasets whenever ground truth is available. In Table 4 we summarize our results for Fishyscapes LostAndFound [Blu19], a validation set of 100 LostAndFound images [Pin16] with refined labels fitting the anomaly track, and the LostAndFound test split, with original labels fitting the obstacle track. Note that, for the LostAndFound test split, we filtered out all images that contain humans and bicycles labeled as obstacles (yielding LostAndFound test-NoKnown) since we applied anomaly segmentation methods out of the box to the task of obstacle segmentation. These methods focus on previously-unseen objects. In comparison to our datasets, for both LostAndFound datasets we observe a less pronounced gap, in terms of both main performance metrics, the pixel-level AuPRC and component-level $\overline{F_1}$ scores, between the methods orignially designed for image classification, especially ODIN and Mahalanobis, and those specifically designed for anomaly segmentation, especially road inpainting and maximized entropy. This signals that both of our datasets contribute new challenges for anomaly and obstacle segmentation.

Finally, we also applied our benchmark suite to the LiDAR guided Small obstacle Segmentation dataset [Sin20]. Our main findings are that our whole set of methods yields weak performance on that dataset. The main purpose of this dataset is the detection of small obstacles from multiple sensors including LiDAR. Hence, the conditions for the other sensor modalities are purposely challenging (*e.g.*, low illumination), making this dataset less suitable to camera-only methods.

V Conclusion

In this work, we have introduced a unified and publicly available benchmark suite that evaluates a method's performance for anomaly segmentation with established pixel level as well as recent component level metrics. Our benchmark suite is applicable in a plug and play fashion to any dataset for anomaly segmentation that comes with ground truth, such as LostAndFound and Fishyscapes LostAndFound, allowing for a better comparison of new methods. Moreover, our benchmark is accompanied with two publicly available datasets, RoadAnomaly21 for anomaly segmentation and RoadObstacle21 for obstacle segmentation.

These two datasets challenge two important abilities of computer vision systems: On one hand their ability to detect and localize unknown objects; on the other hand their ability to reliably detect and localize obstacles on the road, may they be known or unknown. Our datasets consist of real images with pixel-level annotations and depict street scenes with higher variability in object types and object sizes than existing datasets. Our experiments have demonstrated that both of our datasets show a distinct separation in terms of performance between the methods that are specifically designed for anomaly / obstacle segmentation and those that are not. However, there remains much room for performance improvement, particularly in terms of component-wise metrics, which stresses the need for future research in the direction of anomaly segmentation.

The images of the datasets and the software are available at https://www.segmentmeifyoucan.com/.

Broader Impact

This benchmark advances research towards the safe deployment of autonomous vehicles. This ultimately will have many consequences, *e.g.*, reducing the number of jobs in the transport sector. More immediately, the benchmark measures the reliability of algorithms and therefore may be misunderstood as giving safety guarantees. This benchmark however only works for the specified training regime *i.e.* it cannot certify fitness for real-world deployment and should not be misunderstood as such. In particular, while our datasets greatly contribute to the diversity of anomalies, the scale of the datasets is still not even close to sufficient in order to represent every possible type of an anomaly. Furthermore, although we do not publicly provide test labels, there remains a risk, common to any other benchmark, of the community designing methods that overfit on our benchmark tasks.

Acknowledgement. Robin Chan and Svenja Uhlemeyer acknowledge funding by the German Federal Ministry for Economic Affairs and Energy, within the projects "KI Absicherung - Safe AI for Automated Driving", grant no. 19A19005R, and "KI Delta Learning - Scalable AI for Automated Driving", grant no. 19A19013Q, respectively. We thank the consortiums for the successful cooperation. We would also like to thank the "BUW-KI" team who substantially contributed to collecting and labeling of data.

References

[Mac92] David J. C. MacKay. "A Practical Bayesian Framework for Backpropagation Networks". In: *Neural Computation* 4.3 (1992), pp. 448–472 (cit. on p. 158).

[Nea12] Radford M Neal. *Bayesian learning for neural networks*. Vol. 118. Springer Science & Business Media, 2012 (cit. on p. 158).

[Lin14] Tsung-Yi Lin, Michael Maire, Serge Belongie, et al. "Microsoft COCO: Common Objects in Context". In: *Computer Vision – ECCV 2014*. Springer International Publishing, 2014, pp. 740–755. ISBN: 978-3-319-10602-1 (cit. on pp. 155, 164).

[Cre15] Clement Creusot and Asim Munawar. "Real-time small obstacle detection on highways using compressive RBM road reconstruction". In: *2015 IEEE Intelligent Vehicles Symposium (IV)*. 2015, pp. 162–167 (cit. on p. 158).

[Lak15] Brenden M Lake, Ruslan Salakhutdinov, and Joshua B Tenenbaum. "Human-level concept learning through probabilistic program induction". In: *Science* (2015), pp. 1332–1338 (cit. on p. 157).

[Cor16] Marius Cordts, Mohamed Omran, Sebastian Ramos, et al. "The Cityscapes Dataset for Semantic Urban Scene Understanding". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2016 (cit. on pp. 155, 157, 159, 160).

[Gal16] Yarin Gal and Zoubin Ghahramani. "Dropout as a Bayesian Approximation: Representing Model Uncertainty in Deep Learning". In: *Proceedings of The 33rd International Conference on Machine Learning*. Vol. 48. Proceedings of Machine Learning Research. New York, New York, USA: PMLR, June 2016, pp. 1050–1059. URL: http://proceedings.mlr.press/v48/gal16.html (cit. on p. 158).

[Pin16] Peter Pinggera et al. "Lost and found: detecting small road hazards for self-driving vehicles". In: *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 2016 (cit. on pp. 157, 158, 165).

[Bad17] Vijay Badrinarayanan, Alex Kendall, and Roberto Cipolla. "Bayesian SegNet: Model Uncertainty in Deep Convolutional Encoder-Decoder Architectures for Scene Understanding". In: *Proceedings of the British Machine Vision Conference (BMVC)*. BMVA Press, Sept. 2017, pp. 57.1–57.12. ISBN: 1-901725-60-X. DOI: 10.5244/C.31.57. URL: https://dx.doi.org/10.5244/C.31.57 (cit. on p. 158).

[Hen17] Dan Hendrycks and Kevin Gimpel. "A Baseline for Detecting Misclassified and Out-of-Distribution Examples in Neural Networks". In: *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. 2017. URL: https://openreview.net/forum?id=Hkg4TI9xl (cit. on pp. 157, 158, 162, 163, 165).

[Iso17] S. Isobe and S. Arai. "Deep convolutional encoder-decoder network with model uncertainty for semantic segmentation". In: *2017 IEEE International Conference on INnovations in Intelligent SysTems and Applications (INISTA)*. 2017, pp. 365–370 (cit. on p. 155).

[Ken17] Alex Kendall and Yarin Gal. "What Uncertainties Do We Need in Bayesian Deep Learning for Computer Vision?" In: *Advances in Neural Information Processing Systems 30*. Curran Associates, Inc., 2017, pp. 5574–5584. URL: http://papers.nips.cc/paper/7141-what-uncertainties-do-we-need-in-bayesian-deep-learning-for-computer-vision.pdf (cit. on p. 158).

[Lak17] Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. "Simple and Scalable Predictive Uncertainty Estimation using Deep Ensembles". In: *Advances in Neural Information Processing Systems 30*. Curran Associates, Inc., 2017, pp. 6402–6413. URL: http://papers.nips.cc/paper/7219-simple-and-scalable-predictive-uncertainty-estimation-using-deep-ensembles.pdf (cit. on pp. 158, 162, 163, 165).

[Mun17] A. Munawar, P. Vinayavekhin, and G. De Magistris. "Limiting the reconstruction capability of generative neural network using negative learning". In: *2017 IEEE 27th International Workshop on Machine Learning for Signal Processing (MLSP)*. 2017, pp. 1–6. DOI: 10.1109/MLSP.2017.8168155 (cit. on p. 158).

[Zha17] Xiang Zhang and Yann LeCun. "Universum prescription: Regularization using unlabeled data". In: *Thirty-First AAAI Conference on Artificial Intelligence*. 2017 (cit. on p. 155).

[Cho18] Hyunsun Choi, Eric Jang, and Alexander A Alemi. *WAIC, but Why? Generative Ensembles for Robust Anomaly Detection*. Oct. 2018. arXiv: 1810.01392 [stat.ML] (cit. on p. 157).

[DeV18] Terrance DeVries and Graham W. Taylor. *Learning Confidence for Out-of-Distribution Detection in Neural Networks*. Feb. 2018. arXiv: 1802.04865. URL: http://arxiv.org/abs/1802.04865 (cit. on p. 158).

[Lee18] Kimin Lee et al. "A Simple Unified Framework for Detecting Out-of-Distribution Samples and Adversarial Attacks". In: *Advances in Neural Information Processing Systems*. Ed. by S Bengio et al. Vol. 31. Curran Associates, Inc., 2018, pp. 7167–7177. URL: https://proceedings.neurips.cc/paper/2018/file/abdeb6f575ac5c6676b747bca8d09cc2-Paper.pdf (cit. on pp. 157, 158, 162, 163, 165).

[Lia18] Shiyu Liang, Yixuan Li, and R. Srikant. "Enhancing The Reliability of Out-of-distribution Image Detection in Neural Networks". In: *International Conference on Learning Representations*. 2018. URL: https://openreview.net/forum?id=H1VGkIxRZ (cit. on pp. 157, 158, 162, 163, 165).

[Zen18] Oliver Zendel et al. "Wilddash-creating hazard-aware benchmarks". In: *Proceedings of the European Conference on Computer Vision (ECCV)*. openaccess.thecvf.com, 2018, pp. 402–416. DOI: 10.1007/978-3-030-01231-1_25 (cit. on p. 157).

[Ang19] Matt Angus, Krzysztof Czarnecki, and Rick Salay. "Efficacy of Pixel-Level OOD Detection for Semantic Segmentation". In: *CoRR* abs/1911.02897 (2019). arXiv: 1911.02897. URL: http://arxiv.org/abs/1911.02897 (cit. on pp. 155, 158).

[Ata19] Andrei Atanov et al. "Uncertainty Estimation via Stochastic Batch Normalization". In: *Advances in Neural Networks – ISNN 2019*. Ed. by Huchuan Lu, Huajin Tang, and Zhanshan Wang. Cham: Springer International Publishing, 2019, pp. 261–269 (cit. on p. 158).

[Ber19] Paul Bergmann et al. "MVTec AD–A Comprehensive Real-World Dataset for Unsupervised Anomaly Detection". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2019, pp. 9592–9600 (cit. on p. 158).

[Bev19] Petra Bevandić et al. "Simultaneous Semantic Segmentation and Outlier Detection in Presence of Domain Shift". In: *Pattern Recognition*. Ed. by Gernot A. Fink, Simone Frintrop, and Xiaoyi Jiang. Cham: Springer International Publishing, 2019, pp. 33–47 (cit. on p. 158).

[Blu19] Hermann Blum, Paul-Edouard Sarlin, Juan Nieto, et al. "Fishyscapes: A Benchmark for Safe Semantic Segmentation in Autonomous Driving". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) Workshops*. Oct. 2019 (cit. on pp. 155, 157–159, 162, 163, 165).

[Gey19] Jakob Geyer et al. *A2D2: AEV Autonomous Driving Dataset*. `http://www.a2d2.audi`. 2019 (cit. on p. 155).

[Hei19] Matthias Hein, Maksym Andriushchenko, and Julian Bitterwolf. "Why ReLU Networks Yield High-Confidence Predictions Far Away From the Training Data and How to Mitigate the Problem". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2019 (cit. on pp. 155, 158).

[Hen19] Dan Hendrycks, Mantas Mazeika, and Thomas Dietterich. "Deep Anomaly Detection with Outlier Exposure". In: *Proceedings of the International Conference on Learning Representations* (2019) (cit. on p. 158).

[Lis19] Krzysztof Lis et al. "Detecting the Unexpected via Image Resynthesis". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*. Oct. 2019 (cit. on pp. 155, 157–159, 162, 163, 165).

[Muk19] Jishnu Mukhoti and Yarin Gal. *Evaluating Bayesian Deep Learning Methods for Semantic Segmentation*. 2019. arXiv: `1811.12709 [cs.CV]` (cit. on pp. 158, 162, 163, 165).

[Ame20] Joost van Amersfoort et al. *Simple and Scalable Epistemic Uncertainty Estimation Using a Single Deep Deterministic Neural Network*. Mar. 2020. arXiv: `2003.02037 [cs.LG]` (cit. on p. 157).

[Brü20] Dominik Brüggemann et al. "Detecting Out of Distribution Objects in Semantic Segmentation of Street Scenes". In: *The 30th European Safety and Reliability Conference (ESREL)*. 2020 (cit. on p. 155).

[Gus20] Fredrik K. Gustafsson, Martin Danelljan, and Thomas Bo Schön. "Evaluating Scalable Bayesian Deep Learning Methods for Robust Computer Vision". In: *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*. 2020, pp. 1289–1298 (cit. on p. 158).

[Hen20] Dan Hendrycks et al. *Scaling Out-of-Distribution Detection for Real-World Settings*. 2020. arXiv: `1911.11132 [cs.CV]` (cit. on pp. 155, 157–159).

[Jou20] Nicolas Jourdan, Eike Rehder, and Uwe Franke. "Identification of Uncertainty in Artificial Neural Networks". In: *Proceedings of the 13th Uni-DAS e.V. Workshop Fahrerassistenz und automatisiertes Fahren*. July 2020 (cit. on pp. 155, 158).

[Lis20] Krzysztof Lis et al. *Detecting Road Obstacles by Erasing Them*. 2020. arXiv: `2012.13633 [cs.CV]` (cit. on pp. 158, 159, 162, 163, 165).

[Meh20] A. Mehrtash et al. "Confidence Calibration and Predictive Uncertainty Estimation for Deep Medical Image Segmentation". In: *IEEE Transactions on Medical Imaging* (2020), pp. 1–1 (cit. on p. 155).

[Mei20] Alexander Meinke and Matthias Hein. "Towards neural networks that provably know when they don't know". In: *International Conference on Learning Representations*. 2020. URL: `https://openreview.net/forum?id=ByxGkySKwH` (cit. on pp. 157, 158).

[Obe20] Philipp Oberdiek, Matthias Rottmann, and Gernot A. Fink. "Detection and Retrieval of Out-of-Distribution Objects in Semantic Segmentation". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*. June 2020 (cit. on p. 155).

[Rot20] Matthias Rottmann et al. "Prediction Error Meta Classification in Semantic Segmentation: Detection via Aggregated Dispersion Measures of Softmax Probabilities". In: *Proceedings of the International Joint Conference on Neural Networks (IJCNN)*. 2020. ISBN: 9781728169262. DOI: 10.1109/IJCNN48605.2020.9206659. eprint: 1811.00648 (cit. on p. 161).

[See20] Philipp Seeböck et al. "Exploiting Epistemic Uncertainty of Anatomy Segmentation for Anomaly Detection in Retinal OCT". In: *IEEE Trans. Medical Imaging* 39.1 (2020), pp. 87–98 (cit. on p. 159).

[Sin20] Aasheesh Singh et al. *LiDAR guided Small obstacle Segmentation*. Mar. 2020. arXiv: 2003.05970 [cs.RO] (cit. on pp. 157, 158, 165).

[Xia20] Yingda Xia et al. "Synthesize then Compare: Detecting Failures and Anomalies for Semantic Segmentation". In: *Proceedings of the European Conference on Computer Vision (ECCV)*. 2020 (cit. on p. 158).

[Yu20] Fisher Yu et al. "Bdd100k: A diverse driving dataset for heterogeneous multitask learning". In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2020, pp. 2636–2645 (cit. on pp. 155, 157).

[Bau21] Christoph Baur et al. "Autoencoders for unsupervised anomaly segmentation in brain MR images: A comparative study". In: *Medical Image Analysis* 69 (2021), p. 101952. ISSN: 1361-8415. DOI: https://doi.org/10.1016/j.media.2020.101952. URL: https://www.sciencedirect.com/science/article/pii/S1361841520303169 (cit. on p. 159).

[Cha21] Robin Chan, Matthias Rottmann, and Hanno Gottschalk. "Entropy Maximization and Meta Classification for Out-Of-Distribution Detection in Semantic Segmentation". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*. Oct. 2021, pp. 5128–5137 (cit. on pp. 155, 158, 162, 163, 165).

[Di 21] Giancarlo Di Biase et al. "Pixel-Wise Anomaly Detection in Complex Driving Scenes". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2021, pp. 16918–16927 (cit. on pp. 158, 162, 163, 165).

Conclusion

In this dissertation, we dealt with the problem of overlooking relevant instances in the semantic segmentation of street scenes. In this regard, we focussed our research on objects from classes that are rare or unknown to the underlying model for semantic segmentation. For this latter computer vision task, convolutional neural networks (CNNs) are commonly employed, having achieved remarkable advances in recent years. However, due to the data-driven way they are trained, CNNs have shown to be ill-equipped to process rare and unknown object types. This is particular severe in high-stake applications relying on semantic segmentation, such as perception systems for automated driving, where overlooking certain objects could result in hazardous street scenarios.

We realize that the non-detection of objects from rare classes is associated with the problem of class imbalance, which incorporates an unfavorable bias towards predicting classes in training datasets that appear frequently. Regarding the non-detection of objects from unknown classes, we clarified that this is associated with the problem of data anomalies. In semantic segmentation, the task of detecting and localizing such anomalies is also known as anomaly segmentation. In both research directions there was a lack of scientific work, particularly in the field of deep learning. We addressed both outlined problems in the context of CNNs for semantic segmentation.

With respect to the problem of class imbalance, we introduced the concept of cost-based decision rules. Given the probabilistic nature of the output of CNNs for semantic segmentation, cost-based decision rules include different weightings for different types of confusions between classes. Obviously, the sensitivity towards relevant classes can easily be increased by this approach, but we made transparent the serious ethical difficulties associated with class weightings in practical applications, particularly when it comes down to providing numbers. Furthermore, we revealed that the standard maximum a-posteriori decision principle is a special case of a cost-based decision rule itself, which, according to common human sense, unnaturally considers each type of confusion equally serious.

As a mathematical natural alternative, we presented the maximum likelihood decision rule for semantic segmentation. The intuition was that instead of assigning a pixel to the class of the highest probability, it should be assigned to the class for which observed features are most typical. In this way, we achieved a significant improvement in detecting humans as instances of an underrepresented class in street scenes. However, this came to the detriment of overproducing false indications of the same class. For the identification of such false indications we constructed metrics based on location as well as geometry information of segments and based on pixel-wise dispersion measures aggregated over segments. We demonstrated that these hand-crafted metrics are strongly correlated to the prediction quality of segments in semantic segmentation and thus reliably indicate false objects predictions. By combining the maximum likelihood decision rule with techniques for false positive detection, we presented a sophisticated decision rule for semantic segmentation, which substantially reduced the non-detection of humans in street scenes while controlling the amount of false indications thereof.

With respect to the problem of anomaly segmentation, we extensively investigated the softmax entropy of semantic segmentation CNNs as uncertainty measure. The intuition was that objects from semantically unknown classes should come with higher uncertainty than for objects from trained and therefore known classes. Our main finding was that anomalous objects can indeed be detected and localized via the softmax entropy, however to the detriment of overproducing false indications. To tackle this issue, we retrained semantic segmentation CNNs for high softmax entropy on anomalous inputs. To this end, we deliberately included images with semantically unknown objects in the retraining process of CNNs and employed a modified multi-objective loss function to enforce the models to output high softmax entropy on those anomalous examples. This approach significantly improved anomaly segmentation performance, while sacrificing only marginally on the original task of semantic segmentation. Moreover, we further enhanced the performance by combining the entropy maximization approach again with our techniques for false positive detection.

Motivated by the lack of proper datasets and by the missing possibility to reliably compare methods against each other, we created the SegmentMeIfYouCan benchmark. The aim of SegmentMeIfYouCan was mainly to accelerate the progress in the research direction of anomaly segmentation. Within the benchmark, we evaluated multiple state-of-the-art anomaly segmentation methods on newly collected and annotated data, providing valuable insights on the strengths and weaknesses of different kinds of approaches. In this regard, our novel datasets consist of high-quality and real-world images depicting a variety of anomaly types in various scenes, which did not exist in this diversity in other related datasets before. In addition, we provided a public leaderboard and an evaluation suite with established pixel-wise as well as segment-wise performance metrics, which are relevant from perspectives of both scientists and practitioners.

All our presented methods dealing with the detection of relevant instances in semantic segmentation are based on uncertainty estimates. Therefore, more advanced methods to quantify uncertainty are expected to further improve our approaches for handling class imbalance as well as anomaly segmentation. In this light, the field of generative models is an interesting research direction. In general, generative models are typically used to model the distribution of data given only samples drawn from that distribution. Assuming that such a model captures all dependencies in the given data in form of a probability density, uncertainty can then be quantified by means of the negative log-likelihood.

The best known types of a generative model are arguably the generative adversarial network (GAN) [Goo14] or the variational autoencoder (VAE) [Kin14]. Both types have extensively been investigated in the past, achieving impressive results on complex tasks like *e.g.* image reconstruction. However, none of these two model types allow for an exact evaluation of the negative log-likelihood. The family of generative models including so-called normalizing flows [Tab13; Din15; Rez15] does have the mentioned limitation. The key idea of that model type is to learn an invertible mapping between a target distribution and a simple prior distribution. To this end, the prior distribution can be chosen to be of any type, like *e.g.* a Gaussian, since there always exists an invertible mapping according the change-of-variables formula [Rud87]. This allows for an exact evaluation of the negative log-likelihood with respect to the desired target distribution. Thus, applying normalizing flows on deep latent features of CNNs could provide promising uncertainty estimates, which indicates how well observed features in certain parts of an image fit to those observed during training. Normalizing flows have not gained as much attention as GANs and VAEs, yielding a clear shortage of works particularly on high dimensional data. Nevertheless, such likelihood-based generative models represent a reasonable approach for uncertainty quantification in CNNs that are not only employed for semantic segmentation but also other related computer vision tasks, such as instance segmentation or 6D pose estimation. This leaves sufficient room for future work on safe deployment of methods based on deep learning.

Bibliography Part II

[Rud87] W. Rudin. *Real and Complex Analysis*. 3rd ed. Mathematics series. McGraw-Hill, 1987. ISBN: 9780071002769 (cit. on pp. 172, 179).

[Cyb89] G. Cybenko. "Approximation by superpositions of a sigmoidal function". In: *Mathematics of Control, Signals and Systems* 2.4 (Dec. 1989), pp. 303–314. ISSN: 1435-568X. DOI: 10.1007/BF02551274 (cit. on p. 179).

[Tab13] E. G. Tabak and Cristina V. Turner. "A Family of Nonparametric Density Estimation Algorithms". In: *Communications on Pure and Applied Mathematics* 66.2 (2013), pp. 145–164. DOI: https://doi.org/10.1002/cpa.21423. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/cpa.21423. URL: https://onlinelibrary.wiley.com/doi/abs/10.1002/cpa.21423 (cit. on p. 172).

[Goo14] Ian Goodfellow et al. "Generative Adversarial Nets". In: *Advances in Neural Information Processing Systems*. Ed. by Z. Ghahramani et al. Vol. 27. Curran Associates, Inc., 2014. URL: https://proceedings.neurips.cc/paper/2014/file/5ca3e9b122f61f8f06494c97b1afccf3-Paper.pdf (cit. on p. 172).

[Kin14] Diederik P. Kingma and Max Welling. "Auto-Encoding Variational Bayes". In: *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*. Ed. by Yoshua Bengio and Yann LeCun. 2014 (cit. on p. 172).

[Din15] Laurent Dinh, David Krueger, and Yoshua Bengio. "NICE: Non-linear Independent Components Estimation". In: *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Workshop Track Proceedings*. Ed. by Yoshua Bengio and Yann LeCun. 2015 (cit. on p. 172).

[Rez15] Danilo Rezende and Shakir Mohamed. "Variational Inference with Normalizing Flows". In: *Proceedings of the 32nd International Conference on Machine Learning*. Ed. by Francis Bach and David Blei. Vol. 37. Proceedings of Machine Learning Research. Lille, France: PMLR, July 2015, pp. 1530–1538. URL: https://proceedings.mlr.press/v37/rezende15.html (cit. on p. 172).

[Cal20] Ovidiu Calin. *Deep Learning Architectures - A Mathematical Approach*. 1st ed. Springer International Publishing, 2020 (cit. on pp. 179, 180).

List of Figures

Notations and Symbols

Neural Networks

$\mathbf{w}$	weights vector
$\mathbf{W}$	weights matrix / two dimensional kernel
$\mathbf{b}$	biases vector
$\mathbf{z} = \mathbf{W}\mathbf{x} + \mathbf{b}$	weighted inputs vector
$\phi^{(\ell)}(x)$	activation function (in the ℓ-th layer of a network)
$\sigma(x) = \frac{1}{1+e^{-x}}$	logistic sigmoid function
$\mathrm{id}(x) = x$	identity function
$\mathrm{ReLU}(x) = \max\{0, x\}$	rectified linear unit (ReLU) activation function
$\mathrm{softmax}(\mathbf{x})$	softmax activation function
$f^{(\ell)}(\mathbf{x}^{(\ell-1)}) = \mathbf{x}^{(\ell)}$	output / feature vector of the ℓ-th layer
$f(\mathbf{x}) = \mathbf{x}^{(L)} = f^{(L)}(\mathbf{x}^{(L-1)})$	neural network output, output of the L-th layer
K	number of classes
L	number of layers
$\mathcal{L}$	loss function

Statistical Learning

$\mathbf{x}$	input vector
y	ground truth class label
$\mathcal{D}_N = \{(\mathbf{x}^{(i)}, y^{(i)})\}_{i=1}^{N}$	dataset consisting of N data pairs $(\mathbf{x}, y)$

$p(y\|\mathbf{x})$	conditional probability of label y given input $\mathbf{x}$
$f_y(\mathbf{x})$	estimated probability of label y given input $\mathbf{x}$
$f(\cdot\|\theta)$	statistical model parameterized by θ
$\mathscr{L}(\mathcal{D}_N)$	likelihood function for the dataset $\mathcal{D}_N$

Functional Analysis

$\mathcal{R}, \mathcal{S}, \mathcal{U}$	spaces of functions
$(\mathcal{S}, \mathrm{d}), (\mathcal{U}, \mathrm{d})$	metric spaces with distance function d
$C(\mathbb{I}^n)$	real-valued continuous functions on the unit cube in $\mathbb{R}^n$
$\\|f\\|_\infty = \sup_{x\in\mathcal{S}} \|f(x)\|$	uniform norm or sup norm
$\mathrm{L}_{\|\mathcal{S}}$	linear functional restricted to space $\mathcal{S}$

Measure Theory

$\mathbb{I}^n$	n-dimensional unit cube
$\mathbb{1}_A$	indicator function of a set A
μ, ν	(probability) measures
$M(\mathbb{I}^n)$	space of finite signed Borel measures on $\mathbb{I}^n$

Evaluation Quantities and Metrics

TP	number of true positives
FN	number of false negatives
FP	number of false positive
$\mathrm{IoU} = \frac{\mathrm{TP}}{\mathrm{TP+FP+FN}}$	Intersection over Union
$\mathrm{precision} = \frac{\mathrm{TP}}{\mathrm{TP+FP}}$	precision or positive predictive value
$\mathrm{recall} = \frac{\mathrm{TP}}{\mathrm{TP+FN}}$	recall or sensitivity

Definitions, Propositions, Lemmas and Theorems

Definition A.1. *A function $S : \mathbb{R} \to [0,1]$ is called sigmoid if*

$$S(z) \xrightarrow{z \to -\infty} 0 \quad \textit{and} \quad S(z) \xrightarrow{z \to +\infty} 1 \ . \tag{A.1}$$

Definition A.2. *Let $\mathbf{x} \in \mathbb{I}^n := [0,1]^n$ ($\mathbb{I}^n$ the n-dimensional unit cube in $\mathbb{R}^n$) and let $M(\mathbb{I}^n)$ denote the space of finite signed Borel measures on $\mathbb{I}^n$. A function $g : \mathbb{R} \to [0,1]$ is called discriminatory for the measure $\nu \in M(\mathbb{I}^n)$ if*

$$\int_{\mathbb{I}^n} g(\mathbf{w}^\top x + b)\, d\nu(\mathbf{x}) = 0 \quad \forall\ \mathbf{w} \in \mathbb{R}^n, b \in \mathbb{R} \quad \Longrightarrow \quad \nu = 0 \ . \tag{A.2}$$

Proposition A.3. *Let $\mathbb{I}^n$ denote the n-dimensional unit cube in $\mathbb{R}^n$ and let $M(\mathbb{I}^n)$ denote the space of finite signed Borel measures on $\mathbb{I}^n$. Then, any continuous sigmoid function is discriminatory for all measures $\nu \in M(\mathbb{I}^n)$.*

Proof. See [Cal20] page 34, proposition 2.2.4 or [Cyb89] page 307, Lemma 1. □

Theorem A.4. *Hahn-Banach Theorem*
Let $\mathcal{S}$ be a real linear vector space, $\mathcal{U} \subseteq \mathcal{S}$ a linear subspace, $g : \mathcal{S} \to \mathbb{R}$ a linear convex functional and $f : \mathcal{U} \to \mathbb{R}$ another linear functional subject to $f(x) \leq g(x)\ \forall\ x \in \mathcal{U}$. Then, there is a linear functional $\mathrm{L} : \mathcal{S} \to \mathbb{R}$ such that $\mathrm{L}_{|\mathcal{U}} = f$, where $\mathrm{L}_{|\mathcal{U}}$ denotes the restriction of L to $\mathcal{U}$, and $\mathrm{L}(x) \leq g(x)\ \forall\ x \in \mathcal{S}$.

Proof. For a proof of the theorem, see *e.g.* [Rud87]. □

Theorem A.5. *Riesz Representation Theorem*
Let L be a bounded linear functional on $C(\mathcal{K})$, where $\mathcal{K}$ denotes a compact set in $\mathbb{R}^n$ and $C(\mathcal{K})$ the set of real-valued continuous functions on $\mathcal{K}$. Then, there exists a unique finite signed Borel measure ν on $\mathcal{K}$, such that

$$\mathrm{L}(g) = \int_{\mathcal{K}} g\, d\nu \quad \forall\ g \in C(\mathcal{K}) \tag{A.3}$$

Proof. For a proof of the theorem, see *e.g.* [Rud87]. □

Lemma A.6. *Let $C(\mathbb{I}^n)$ denote the space of continuous functions and let $M(\mathbb{I}^n)$ denote the space of finite signed Borel measures on the n-dimensional unit cube $\mathbb{I}^n$. Moreover, let $\mathcal{U}$ be a linear and non-dense subspace of $C(\mathbb{I}^n)$. Then, there is a measure $\nu \in M(\mathbb{I}^n)$ such that*

$$\int_{\mathbb{I}^n} h\, d\nu = 0 \quad \forall\ h \in \mathcal{U} \ . \tag{A.4}$$

Proof. By the Hahn-Banach theorem, *cf*. theorem A.4, there is a bounded linear functional $\mathrm{L} : C(\mathbb{I}^n) \to \mathbb{R}$, with the property that $\mathrm{L} \neq 0$ but $\mathrm{L}(h) = 0 \ \forall \ h \in \mathcal{U}$. By the representation theorem of linear bounded functionals on $C(\mathbb{I}^n)$, *cf*. theorem A.5, there is a measure $\nu \in M(\mathbb{I}^n)$ such that

$$\mathrm{L}(g) = \int_{\mathbb{I}^n} g \, d\nu \quad \forall \ g \in C(\mathbb{I}^n) \ . \tag{A.5}$$

In particular, equation (A.5) holds for any $h \in \mathcal{U}$, since

$$\mathcal{U} \subset C(\mathbb{I}^n) \implies \mathrm{L}(h) = \int_{\mathbb{I}^n} h \, d\nu = 0 \quad \forall \ h \in \mathcal{U} \ . \tag{A.6}$$

For further details, see [Cal20] page 257, lemma 9.3.3. □