Bergische Universität Wuppertal

Fakultät für Mathematik und Naturwissenschaften

Institute of Mathematical Modelling, Analysis and Computational Mathematics (IMACM)

Preprint BUW-IMACM 19/39

Robin Chan, Matthias Rottmann, Fabian Hüeger, Peter Schlicht and Hanno Gottschalk

MetaFusion: Controlled False-Negative Reduction of Minority Classes in Semantic Segmentation

December 17, 2019

http://www.math.uni-wuppertal.de

MetaFusion: Controlled False-Negative Reduction of Minority Classes in Semantic Segmentation

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

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 which assign larger 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. We therefore fuse false-negative detection with uncertainty based false-positive meta classification. We present proof-of-concept results for CIFAR-10, and prove 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.

1 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, see however [23] 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 [3, 30] by revealing a bias towards the overrepresented class. Being an classification problem at pixel-level, semantic segmentation therefore is exhibited to the same set of problems when class imbalance is present. As the imbalance naturally exist in most datasets for "real world" applications, finding the underrepresented class is of highest interest.

Methods for handling class imbalance have been developed and can be divided into two main categories: *sampling-based* and *algorithm-based* techniques [3, 23, 28]. 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, balancing data is done 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 [38], respectively. They can lead to performance improvement, in particular with random oversampling [3, 30, 32] unless there is no overfitting [9]. A more advanced approach called SMOTE [8] alleviates the latter issue by creating synthetic examples of minority classes.

Oversampling methods are difficult to apply on semantic segmentation datasets due to naturally occurring class frequencies on single input frames. Considering the Cityscapes [13] 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 already are strongly represented in this datatset as street scenarios are shown from a car driver's perspective.

The training approach is to assign costs to different classification mistakes for different classes and include them in the loss function [4, 5, 39]. Instead of minimizing the total error, the average misclassifcation cost is minimized. In addition, methods have been proposed learning the cost parameters throughout training [26, 40] and thus eliminating the ethical problem of predefining them [6]. 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 correct class imbalance during inference is output thresholding, thus interchanging the standard maximum a-posteriori probability (MAP) principle for an alternate decision rule. Dividing the CNN's output by the estimated prior probabilities for each class was proposed in [3, 7] which is also known as Maximum Likelihood rule in decision theory [17]. This results in a reduced likelihood of misclassifying minority class objects and a performance gain in particular with respect to the sensitivity of rare classes. Output thresholding does neither affect training time nor the model's capability to discriminate between different groups. It is still a suitable technique for reducing class bias as it shifts the priority to predicting certain classes that can be easily added on top of every CNN.

In the field of semantic segmentation of street scenes the overall performance metric intersection over union (IoU) [16] is mainly used. This metric is highly biased towards large and therefore majority class objects such as street or buildings. Currently, state-of-the-art models achieve class IoU scores of 83% for Cityscapes [13] and 73% for Kitti [19]. 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 con-

[1] University of Wuppertal, School of Mathematics and Natural Sciences, Germany, email: {rchan, rottmann, hgottsch}@uni-wuppertal.de
[2] Volkswagen Group Innovation, Center of Innovation Automation, Germany, email: {fabian.hueger, peter.schlicht}@volkswagen.de

sequences in applications like automated driving.

In this context, uncertainty estimates are helpful as they can be used to quantify how likely an incorrect prediction has been made. 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 [21]. More advanced techniques include Bayesian neural networks (BNNs) that are supposed to output distributions over the model's weight parameters [33]. As BNNs come with a prohibitive computational cost, recent works developed approximations such as Monte-Carlo dropout [18] or stochastic batch normalization [2]. 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 [14, 25] which seems to be more appropriate in terms of computational efficiency for real-time inference.

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 [1]. Therefore, auxiliary machine learning models for predicting the segmentation quality [27, 40] have been proposed. While some methods built upon hand-crafted features, some other methods apply CNNs for that task by learning a mapping from the final segmentation to its prediction quality [15, 22]. A segment-based prediction rating method for semantic segmentation was proposed in [34] and extended in [35, 31]. They derive aggregated dispersion metrics from the CNN's softmax output and pass them through a classifier that discriminates whether one segment intersects with the ground truth or not. These hand-crafted metrics have shown to be well-correlated to the calculated segment-wise IoU. The method is termed "*MetaSeg*" which we use from now on to refer to that procedure.

In this work, we present 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. In 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 mainly builds on methods already presented in [7] and [34]. Our main contribution is the fusion of these two components providing an additional segmentation mask that is more sensitive to finding rare class objects, but keeps false-positive instances in check. Compared to different class weightings for decision thresholding, we obtain a more favorable trade-off between error rates. As inference post-processing tool, our method does not touch the underlying CNN architecture used for semantic segmentation, it is computationally cheap, easily interpretable and can be seamlessly added on top of other CNNs for semantic segmentation.

This work is structured as follows: In sections 2 and 3, we recall the building blocks of our approach, namely the Maximum Likelihood decision rule for the reduction false-negatives and MetaSeg for false-positive segments detection, respectively. In section 4, we combine the latter components and show proof-of-concept results for CIFAR-10 in section 5. We complement this work by extending our approach to the application-relevant task of semantic segmentation and show numerical results for the Cityscapes data in section 6.

2 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 $\mathcal{Y} := \{1, \dots, l\}$ 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:

$$d_{Bayes}(x)_z := \underset{y \in \mathcal{Y}}{\operatorname{argmax}}\, p_z(y|x) \, . \tag{1}$$

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

$$R_{sym}(d) := \frac{1}{|\mathcal{Z}|} \sum_{z \in Z} \sum_{y \in Y} 1_{\{d(x)_z \neq y\}} p_z(y|x) \;\forall\, x \in \mathcal{X} \tag{2}$$

it holds $R_{sym}(d_{Bayes}) \leq R_{sym}(d)$. In decision theory, this principle is also known as Bayes decision rule [17] 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 applied 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:

$$\hat{y}_z = d_{ML}(x)_z := \underset{y \in \mathcal{Y}}{\operatorname{argmax}}\, p_z(x|y) = \underset{y \in \mathcal{Y}}{\operatorname{argmax}}\, \frac{p_z(y|x)}{p_z(y)} \, . \tag{3}$$

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 [24]).

Analogously to [7], we approximate $p_z(y)$ in a position-specific manner using the pixel-wise class frequencies of the training set:

$$\hat{p}_z(y) = \frac{1}{|\mathcal{X}|} \sum_{x \in \mathcal{X}} 1_{\{y_z(x) = y\}} \;\forall\, y \in \mathcal{Y}, z \in \mathcal{Z} \, . \tag{5}$$

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-predictions 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.

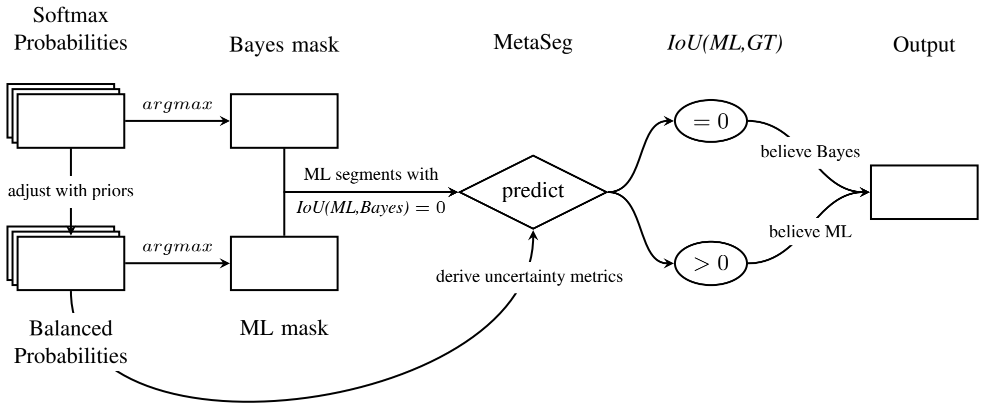


Figure 1. Overview of our method for controlled false-negative reduction of minority classes which we term "*MetaFusion*". Note that *IoU* denotes the intersection over union measure of two segmentation masks.

3 PREDICTION ERROR 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 [34, 35]. Given the conditional likelihood (softmax output adjusted with priors), we estimate uncertainty per segment by aggregating different pixel-wise dispersion measures, such as entropy

$$E_z(x) = -\frac{1}{\log(|\mathcal{Y}|)} \sum_{y \in \mathcal{Y}} p_z(x|y) \log(p_z(x|y)) \; \forall \, z \in \mathcal{Z}, \qquad (6)$$

probability margin

$$M_z(x) = 1 - p_z(x|\hat{y}_z) + \max_{y \in \mathcal{Y} \setminus \{\hat{y}_z\}} p_z(x|y) \; \forall \, z \in \mathcal{Z} \qquad (7)$$

or variation ratio

$$V_z(x) = 1 - p_z(x|\hat{y}_z) \; \forall \, z \in \mathcal{Z} \, . \qquad (8)$$

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 classification, in the following abbreviated as "meta" classification, compared to [34]. For instance, instead of computing logistic least absolute shrinkage and selection operator (LASSO [37]) regression fits, we use gradient-boosting trees (GB [20]). GB has shown to be a powerful classifier on binary classification problems and structured data with modest dataset size which both match our problem setting.

Additionally to the uncertainty measures, we introduce further metrics indicating incorrect predictions. For localization purposes we include a segment's geometric center

$$G_h(k) = \frac{1}{|k|} \sum_{i=1}^{|k|} h_i \; , \; G_v(k) = \frac{1}{|k|} \sum_{j=1}^{|k|} v_j \qquad (9)$$

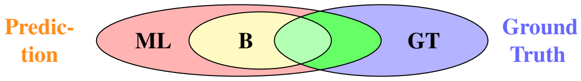


Figure 2. Graphical illustration of the relation between Bayes and ML prediction segments for rare classes.

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 / 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 surrounding area 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, regarding segment k,

$$N(k|y) = \frac{1}{|k_{bd}|} \sum_{z \in k_{bd}} 1_{\{\hat{y}_z = y\}} \; \forall \, y \in \mathcal{Y} \qquad (10)$$

expresses the ratio of the amount of pixels in the neighborhood predicted to belong to class y to neighborhood size.

4 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. For the most underrepresented class $c \in \mathcal{Y}$ in an unbalanced semantic segmentation dataset, in many real-world applications often also the class of highest interest, all predicted Bayes segments are inside ML segments [7], see figure 2. Consequently, for c we assume that a non-empty intersection between an ML segment and any Bayes segment (predicted to belong to 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 in 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

$$U_k := \mu_k((\hat{p}(x|y))_{y \in \mathcal{Y}}) \; \forall \, k \in \mathcal{D}_{x|c} \tag{12}$$

that serve as input for the meta classifier, see also section 3 and cf. [34, 35]. The classifier we use in our meta model is gradient-boosting tree algorithm (GB [20]) and it is trained to discriminate between true-positive and false-positive segment prediction. 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 $k' \in \hat{\mathcal{K}}_{x,ML} \backslash \mathcal{D}_{x|c}$, whether or not they are minority class segments, we stick to the Bayes decision rule as well as it is optimal with respect to the expected total number of errors, see equation (2). Therefore, the final segmentation output

$$d_{Fusion}(x)_z = \begin{cases} d_{ML}(x)_z, & \text{if } \hat{f}(U_k) = 1 \land z \in k \in \mathcal{D}_{x|c} \\ d_{Bayes}(x)_z, & \text{else} \end{cases} \tag{14}$$

fuses 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. We term our approach "*MetaFusion*" and provide a summary as graphical illustration in figure 1.

5 NUMERICAL RESULTS FOR CIFAR-10

In order to test the general concept of MetaFusion, we perform experiments with CIFAR-10 [29]. The dataset is commonly used for image classification and contains 60k color images of resolution 32 × 32 pixels in 10 classes, each class having the same amount of samples. The CNN architecture we use in our experiments for this task (figure 3) is adopted from the Keras documentation [12], the network is reported to achieve a validation accuracy of 79% after 50 epochs of training.

To evaluate our method, we construct a rare class setup with ten CNNs. In this setup, the i-th CNN is trained on a CIFAR-10 (training data) subset assembled by randomly leaving out 90% of the samples of class i. In [3] it already has been shown empirically that the application of the Maximum Likelihood decision rule positively affects the classification performance, not only increasing area under the receiver operating characteristic curve (AUROC) but also total accuracy. In particular, compensating for prior class probabilities increases the number of properly classified minority class samples. Based on this finding, we examine MetaFusion's behavior.

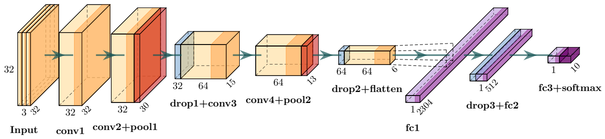


Figure 3. Convolutional neural network architecture applied in our tests on CIFAR-10. Each convolution is followed by a ReLU (rectified linear unit) activation. The dropout rate is 0.25 for the first two dropout layers and 0.50 for the third one, respectively.

We evaluate the ten CNNs on one and the same CIFAR-10 validation set consisting of 10k images with balanced class distribution. Each CNN is trained 50 epochs with categorical cross-entropy loss. After adjusting the softmax probabilities by the priors (cf. equation (3)) to perform the ML decision rule subsequently, we derive three dispersion measures, namely entropy E, probability margin M and variation ratio V. Note that, since CIFAR-10 is an image classification task, the priors as well as the metrics are on the level of full images (e.g., consider $\mathcal{Z} = \{1\}$). For each CNN, the candidate images for MetaFusion are samples predicted by ML to belong to the trained subset's minority class, but not by Bayes. As meta classifier we use GB that, based on E, M, V, classifies if an image is predicted correctly or incorrectly. For GB we employ 15 boosting stages with maximum depth of 3 per tree and exponential loss function. In case an image's classification result is meta classified to be incorrect, we replace it with the class prediction obtained by the Bayes rule. MetaFusion is leave-one-out cross-validated.

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. In figure 4 we see that, averaged over the ten CIFAR-10 subsets we obtain roughly 611 FNs and 39 FPs with Bayes whereas ML produces 271 FNs and 432 FPs. As a baseline we interpolate the priors between these two 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

$$d_{adj}(x,\alpha)_z := \operatorname*{argmax}_{y \in \mathcal{Y}} \frac{p_z(y|x)}{p_{z,\alpha}(y)} \tag{16}$$

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 4 that may serve as an intuitive approach to balance FNs and 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 we also have to sacrifice some of the highly desired FNs at the same time. The diagonal gray lines visualizes level sets with respect to the sum of FNs and FPs which is the absolute number of errors, i.e., on each of these line the sum is constant. In our experiments, we choose equidistant interpolation degrees α. Due to lack of data for MetaFusion the smaller α and/or the more confident the underlying image classification model, the smallest interpolation degree we use is $\alpha = 0.9$. We notice that calibrating the class weightings leads to better average performance

Table 1. Performance comparison of Bayes, ML and MetaFusion on minority classes in CIFAR-10. In total, the performance of ten CNNs are reported, each CNN trained with a different minority class. To generate the unbalanced training dataset, 90% of one class' samples were randomly removed.

	Bayes			Maximum Likelihood				MetaFusion			
y	F_1	FP	FN	F_1	FP	FN	Δ	F_1	FP	FN	Δ
0	0.66	**85**	461	0.70	619	**132**	1.62	**0.72**	175	332	**0.70**
1	0.58	**21**	585	0.78	169	**253**	0.45	**0.79**	103	280	**0.27**
2	0.41	**39**	729	**0.54**	698	**367**	1.82	0.50	112	629	**0.73**
3	0.15	**17**	915	**0.53**	502	**452**	1.05	0.42	138	691	**0.54**
4	0.46	**70**	682	0.60	852	**203**	1.63	**0.63**	195	452	**0.54**
5	0.34	**35**	787	**0.61**	528	**330**	1.08	0.56	207	532	**0.67**
6	0.67	**41**	473	0.75	336	**202**	1.09	**0.77**	137	293	**0.53**
7	0.65	**24**	506	**0.74**	248	**261**	0.91	0.73	106	368	**0.60**
8	0.69	**29**	459	0.79	183	**234**	0.68	**0.79**	126	260	**0.49**
9	0.64	**31**	516	**0.76**	185	**273**	0.63	0.76	132	309	**0.49**
$\bar{y}$	0.52	**39**	611	**0.68**	432	**271**	1.09	0.67	143	415	**0.56**
	Averaged total accuracy score on validation set										
$\bar{y}$	0.75451			**0.76563**				0.76707			

with respect to the sum of false-positives and false-negatives. We observe that applying MetaFusion reduces the sum of errors once more, nearly throughout all of investigated interpolation degrees (green line lying below blue line in figure 4). In order to further analyze this test, we state numbers for single runs in table 1, complemented with additional evaluation metrics.

As an overall performance measure, although being skewed towards majority classes, we report the score $F_1 = 2TP/(2TP + FP + TP)$ with TP being the number of true-positives. Another measure for MetaFusion is the ratio between prediction errors. For any decision rule $d_{adj} : [0,1]^{|\mathcal{Y}|} \times \mathbb{R} \mapsto \mathcal{Y}$, the slope

$$\Delta(d_{adj}) = \frac{FP(d_{adj}) - FP(d_{Bayes})}{FN(d_{Bayes}) - FN(d_{adj})} \tag{17}$$

with d_{adj} such that $FN(d_{Bayes}) - FN(d_{adj}) \neq 0$ 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 d_{adj} compared to d_{Bayes} (whereas it may increase for the other classes).

The average F_1 score is 67%, marginally less than with ML (68%). This outcome is mainly caused by the class 3 (cat) where ML considerably outperforms MetaFusion by 11 percent points. For the remaining classes the exchange is at most 5 percent points with either MetaFusion achieving a higher score than ML or vice versa. MetaFusion is superior to ML in average Δ by taking roughly only one FP in order to reduce two FNs. Hence, with respect to average error rates, MetaFusion outperforms Bayes and ML. On average $\Delta(d_{ML}) = 1.09$, i.e., ML produces slightly more than one FP to reduce one FN in comparison to Bayes. Moreover, also compared to Bayes, the amount of predicted minority class instances is significantly increased, leading to an improved performance per minority class by 16% on average and also in total accuracy by 1.11%. This result confirms the finding from [3].

Summarizing this test, we have shown empirically that the basic concept of MetaFusion works for image classification by implementing a minimal version of the method and reporting numerical results. Next, we aim at extending our method to the application-relevant and more complex task of semantic segmentation.

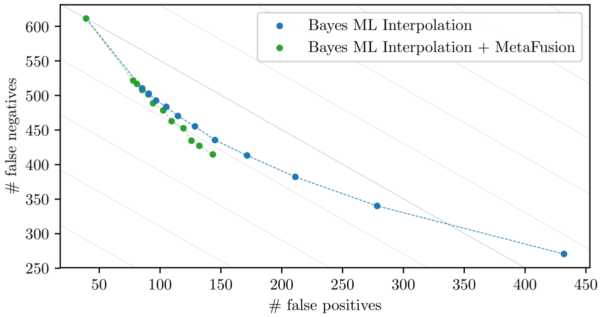


Figure 4. False-positives vs. false-negatives on CIFAR-10. The blue line is an interpolation between the Bayes and ML. Different points indicate equidistant interpolation degrees starting from 0.9, i.e., different degrees of decision thresholding. MetaFusion (green points) is applied in accordance to the interpolated class weightings. The diagonal lines denote level sets in which the sum of both errors equals the same value.

6 NUMERICAL RESULTS FOR CITYSCAPES

Semantic segmentation is a crucial step in the process of perceiving a vehicle's surroundings for automated driving. Therefore, we perform tests on the Cityscapes dataset [13] 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 in 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. This would lead to class priors, when estimating via pixel-wise frequency, conflicting with human common sense due to the possible preference of static objects over persons. Therefore, we use category priors treating objects more superficially (by aggregrating all clases into the 8 predefined categories), with pedestrians and rider aggregated to "*human*" class then being significantly underrepresented relative to all remaining categories.

We perform the Cityscapes experiments using DeeplabV3+ networks [10] with MobileNetV2 [36] and Xception65 [11] backbones. We apply MetaFusion per predicted human segment as presented in section 4 and evaluate only the human class in the Cityscapes validation data. As meta classifier we employ GB with 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 2.

Similar to the experiments in section 5, we interpolate between Bayes and ML priors according to equation (15) but now for every pixel location $z \in \mathcal{Z}$. We again 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. 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 the bigger the interpolation degree.

For MobileNetV2, see also figure 5, we observe that the number of false-positives increases from 865 up to 4885 when applying ML instead of Bayes while the number of false-negatives decreases from 839 down to 476. This results in large $\Delta = 11.07$ expressing that roughly 11 FPs are produced in order to remove one single FN. Clearly, there is an overproduction of predicted human segments that

Table 2. Performance comparison of different decision rules and MetaFusion for DeeplabV3+ with MobileNetV2 [36] and Xception65 backbones on Cityscapes. Different adjusted decision rules are obtained according to equation (16).

Priors interpol.	Adjusted Decison Rule				MetaFusion			
degree α	$mIoU$	FP	FN	Δ	$mIoU$	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**

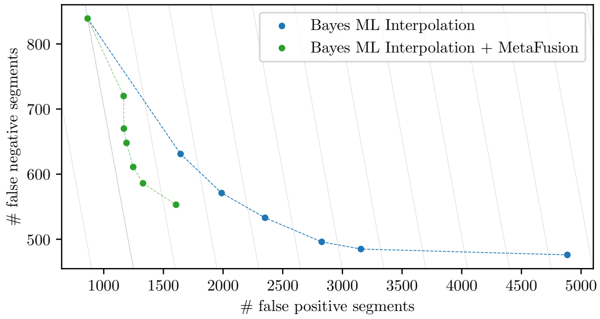


Figure 5. False-positives vs. false-negatives of person segments for MobileNetV2 on Cityscapes. The diagonal lines denote level sets in which the sum of both errors equals the same value.

we want to keep under control using MetaFusion.

By applying MetaFusion, the number of false-positives is reduced to a third of ML false-positives while keeping more than two thirds (78.79%) of 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% and ML 8.4% for detecting the false-negatives additionally to Bayes. 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 total number errors for the adjusted decision rules. Different to the findings for CIFAR-10, the class weightings' 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 a large set of error weightings. Just like for CIFAR-10, 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.

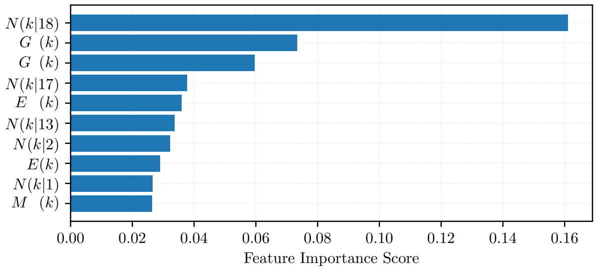


Figure 6. Feature importance scores of the gradient-boosting classifier for MobileNetV2 applied on all disjoint ML and Bayes human segments. The score is averaged over all random cross-validation splits and only the ten features with the highest score are depicted. In total we used 56 metrics as meta model input. N and G are defined in section 3. E and M denotes the segment-wise averaged entropy and probability margin, respectively, with *bd* indicating the restriction on the segment's boundary.

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, MetaFusion accepts in average 2.8 FPs for removing one single FN which is more than half of the average Δ 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 as

$$I_n(t) = n(t)Q(t) - n_{left}(t)Q_{left}(t) - n_{right}(t)Q_{right}(t) \quad (18)$$

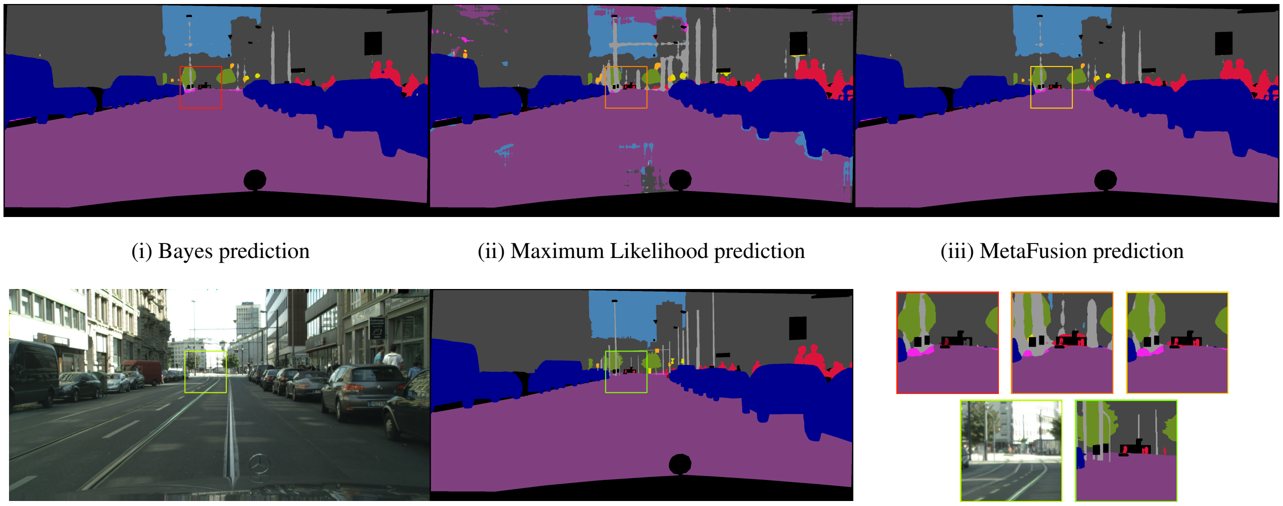
with $Q(t)$ the Gini impurity [20] and $n(t)$ the weighted number of samples in node $t \in \mathcal{T}$ (the weighting corresponds to the portion of all samples reaching node t). Moreover, by *left* and *right* we denote the respective children nodes. Then the importance of $\hat{f}$ of feature / uncertainty metric $m \in [0, 1]$ is computed as

$$I(m) = \frac{\sum_{t \in \mathcal{T}} \chi(t|m) I_n(t)}{\sum_{t \in \mathcal{T}} I_n(t)} \quad (19)$$

with

$$\chi(t|m) = \begin{cases} 1, & \text{if node } t \text{ splits on feature } m \\ 0, & \text{else} \end{cases} . \quad (20)$$

The ten features of highest importance (in experiments with MobileNetV2) are reported in figure 6. By a large margin, a segment's neighborhood including class id 18, which corresponds to bicycles,



(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 7. Example of generated segmentation masks with MobileNetV2. In the top row: prediction mask 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 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, 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 that are rather small and barely visible in the original image.

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 [34], it already has been shown that these two metrics are well-correlated to the segment-wise IoU. GB also uses these correlations to perform the meta classification. In contrast to the findings in [34], 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 surrounding 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) motocycle, car, building and sidewalk.

7 CONCLUSION

In this work, we presented a novel post-processing approach for semantic segmentation. 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 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 predictions of the same class. By deriving uncertainty measures per predicted segment and passing them through a gradient-boosting classifier, we are able to detect false-positive segment predictions in the ML mask in an automated and computationally cheap fashion. We remove these segments which are identified as incorrect and replace them with the Bayes mask. In this way, we significantly reduce the number of false-positives, at the same time only sacrificing a small number of detected false-negatives and also only resulting in a minor overall performance loss in comparison to the standard Bayes decision rule in the Cityscapes dataset. In fact, our method, which we term "*MetaFusion*", 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 whereby the performance gain is more substantial the greater the difference between the Bayes and ML mask. Furthermore, we tested the basic concept of MetaFusion on an image classification problem as well. Although we applied only a minimal version on the CIFAR-10 dataset, we observed similar results to Cityscapes demonstrating the method's generalization capabilities for various tasks. 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-sensitive 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`.

ACKNOWLEDGEMENTS

This work is in part funded by Volkswagen Group Innovation. We thank Jan David Schneider and Matthias Fahrland for fruitful discussions and programming support.

REFERENCES

[1] Dario Amodei, Chris Olah, Jacob Steinhardt, Paul F. Christiano, John Schulman, and Dan Mané, 'Concrete problems in AI safety', *CoRR*, **abs/1606.06565**, (2016). 2

[2] Andrei Atanov, Arsenii Ashukha, Dmitry Molchanov, Kirill Neklyudov, and Dmitry Vetrov, 'Uncertainty estimation via stochastic batch normalization', in *Advances in Neural Networks – ISNN 2019*, eds., Huchuan Lu, Huajin Tang, and Zhanshan Wang, pp. 261–269, Cham, (2019). Springer International Publishing. 2

[3] Mateusz Buda, Atsuto Maki, and Maciej A. Mazurowski, 'A systematic study of the class imbalance problem in convolutional neural networks', *Neural Networks*, **106**, 249 – 259, (2018). 1, 4, 5

[4] Samuel Rota Bulò, Gerhard Neuhold, and Peter Kontschieder, 'Loss max-pooling for semantic image segmentation', *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 7082–7091, (2017). 1

[5] Holger Caesar, Jasper Uijlings, and Vittorio Ferrari, 'Joint calibration for semantic segmentation', in *Proceedings of the British Machine Vision Conference (BMVC)*, eds., Mark W. Jones Xianghua Xie and Gary K. L. Tam, pp. 29.1–29.13. BMVA Press, (September 2015). 1

[6] Robin Chan, Matthias Rottmann, Radin Dardashti, Fabian Huger, 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*, (6 2019). 1

[7] Robin Chan, Matthias Rottmann, Fabian Hüger, Peter Schlicht, and Hanno Gottschalk, 'Application of decision rules for handling class imbalance in semantic segmentation', *CoRR*, **abs/1901.08394**, (2019). 1, 2, 3

[8] Nitesh Chawla, Kevin Bowyer, Lawrence O. Hall, and W Philip Kegelmeyer, 'Smote: Synthetic minority over-sampling technique', *J. Artif. Intell. Res. (JAIR)*, **16**, 321–357, (01 2002). 1

[9] Nitesh V. Chawla, Nathalie Japkowicz, and Aleksander Kotcz, 'Editorial: Special issue on learning from imbalanced data sets', *SIGKDD Explor. Newsl.*, **6**(1), 1–6, (June 2004). 1

[10] Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam, 'Encoder-decoder with atrous separable convolution for semantic image segmentation', in *The European Conference on Computer Vision (ECCV)*, (9 2018). 5

[11] François Chollet, 'Xception: Deep learning with depthwise separable convolutions', in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, (7 2017). 5

[12] François Chollet et al. Keras. `https://keras.io`, 2015. 4

[13] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele, 'The cityscapes dataset for semantic urban scene understanding', in *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, (2016). 1, 5

[14] Terrance DeVries and Graham W Taylor, 'Learning confidence for out-of-distribution detection in neural networks', *arXiv preprint arXiv:1802.04865*, (2018). 2

[15] Terrance DeVries and Graham W. Taylor, 'Leveraging uncertainty estimates for predicting segmentation quality', *CoRR*, **abs/1807.00502**, (2018). 2

[16] Mark Everingham, S. M. Ali Eslami, Luc Van Gool, Christopher K. I. Williams, John Winn, and Andrew Zisserman, 'The pascal visual object classes challenge: A retrospective', *International Journal of Computer Vision*, **111**(1), 98–136, (Jan 2015). 1

[17] L. Fahrmeir, A. Hamerle, and W. Häussler, *Multivariate statistische Verfahren (in German)*, Walter De Gruyter, 2 edn., 1996. 1, 2

[18] 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*, eds., Maria Florina Balcan and Kilian Q. Weinberger, volume 48 of *Proceedings of Machine Learning Research*, pp. 1050–1059, New York, New York, USA, (20–22 Jun 2016). PMLR. 2

[19] Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun, 'Vision meets robotics: The kitti dataset', *International Journal of Robotics Research (IJRR)*, (2013). 1

[20] Trevor Hastie, Robert Tibshirani, and Jerome Friedman, *The elements of statistical learning: data mining, inference and prediction*, Springer, 2 edn., 2009. 3, 4, 6

[21] 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). 2

[22] C. Huang, Q. Wu, and F. Meng, 'Qualitynet: Segmentation quality evaluation with deep convolutional networks', in *2016 Visual Communications and Image Processing (VCIP)*, pp. 1–4, (Nov 2016). 2

[23] Taghi M. Johnson, Justin M.and Khoshgoftaar, 'Survey on deep learning with class imbalance', *Journal of Big Data*, **6**(1), 27, (Mar 2019). 1

[24] James Joyce, 'Bayes' theorem', in *The Stanford Encyclopedia of Philosophy*, ed., Edward N. Zalta, Metaphysics Research Lab, Stanford University, spring 2019 edn., (2019). 2

[25] 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*, eds., I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, 5574–5584, Curran Associates, Inc., (2017). 2

[26] S. H. Khan, M. Hayat, M. Bennamoun, F. A. Sohel, and R. Togneri, 'Cost-sensitive learning of deep feature representations from imbalanced data', *IEEE Transactions on Neural Networks and Learning Systems*, **29**(8), 3573–3587, (Aug 2018). 1

[27] Timo Kohlberger, Vivek Singh, Chris Alvino, Claus Bahlmann, and Leo Grady, 'Evaluating segmentation error without ground truth', in *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2012*, eds., Nicholas Ayache, Hervé Delingette, Polina Golland, and Kensaku Mori, pp. 528–536, Berlin, Heidelberg, (2012). Springer Berlin Heidelberg. 2

[28] Bartosz Krawczyk, 'Learning from imbalanced data: open challenges and future directions', *Progress in Artificial Intelligence*, **5**(4), 221–232, (Nov 2016). 1

[29] Alex Krizhevsky, 'Learning multiple layers of features from tiny images', *University of Toronto*, (05 2012). 4

[30] Victoria López, Alberto Fernández, Salvador García, Vasile Palade, and Francisco Herrera, 'An insight into classification with imbalanced data: Empirical results and current trends on using data intrinsic characteristics', *Information Sciences*, **250**, 113 – 141, (2013). 1

[31] Kira Maag, Matthias Rottmann, and Hanno Gottschalk, 'Time-dynamic estimates of the reliability of deep semantic segmentation networks', *CoRR*, **abs/1911.05075**, (2019). 2

[32] David Masko and Paulina Hensman, *The Impact of Imbalanced Training Data for Convolutional Neural Networks*, Ph.D. dissertation, KTH Royal School of Technology, 2015. 1

[33] Radford M Neal, *Bayesian learning for neural networks*, volume 118, Springer Science & Business Media, 2012. 2

[34] 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', *CoRR*, **abs/1811.00648**, (2018). 2, 3, 4, 7

[35] Matthias Rottmann and Marius Schubert, 'Uncertainty measures and prediction quality rating for the semantic segmentation of nested multi resolution street scene images', *CoRR*, **abs/1904.04516**, (2019). 2, 3, 4

[36] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen, 'Mobilenetv2: Inverted residuals and linear bottlenecks', in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, (6 2018). 5, 6

[37] Robert Tibshirani, 'Regression shrinkage and selection via the lasso', *Journal of the Royal Statistical Society: Series B*, **58**, 267–288, (1996). 3

[38] 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, pp. 935–942, New York, NY, USA, (2007). ACM. 1

[39] S. Wang, W. Liu, J. Wu, L. Cao, Q. Meng, and P. J. Kennedy, 'Training deep neural networks on imbalanced data sets', in *2016 International Joint Conference on Neural Networks (IJCNN)*, pp. 4368–4374, (July 2016). 1

[40] 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)*, pp. 4362–4367, (July 2016). 1, 2