Controlled False Negative Reduction of Minority Classes in Semantic Segmentation

1st Robin Chan
University of Wuppertal & IZMD
chan@math.uni-wuppertal.de

2nd Matthias Rottmann
University of Wuppertal & IZMD
rottmann@math.uni-wuppertal.de

3rd Fabian Hüger
Volkswagen Group Innovation
fabian.hueger@volkswagen.de

4th Peter Schlicht
Volkswagen Group Innovation
peter.schlicht@volkswagen.de

5th Hanno Gottschalk
University of Wuppertal & IZMD
hanno.gottschalk@uni-wuppertal.de

***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.**

***Index Terms*—computer vision, convolutional neural networks, class imbalance, false negative reduction**

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 [1] 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 [2], [3] 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 [1], [2], [4]: *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 [5], respectively. They can lead to performance improvement, in particular with random oversampling [2], [3], [6] unless there is no overfitting [7]. A more advanced approach called SMOTE [8] 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 [9] 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 [10], [11], [12]. Instead of the total error, the average misclassifcation cost is minimized. In addition, methods learning the cost parameters throughout training have been proposed [13], [14] and thus circumventing the ethical problem of predefining them [15]. 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 [2], [16] which

978-1-7281-6926-2/20/$31.00 ©2020 IEEE

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 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) [18] 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 [9] 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 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 [20]. More advanced techniques include Bayesian neural networks (BNNs) that yield posterior distributions over the model's weight parameters [21]. As BNNs come with a prohibitive computational cost, recent works developed approximations such as Monte-Carlo dropout [22] or stochastic batch normalization [23]. 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 [24], [25] 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 [26]. Therefore, auxiliary machine learning models for predicting the segmentation quality [14], [27] 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 [28], [29]. A segment based prediction rating method for semantic segmentation was proposed in [30] and extended in [31], [32]. 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 [16] and [30], 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 [15] 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 $\mathcal{Y} := \{1, \dots, l\}$ 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

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

we have $R_{sym}(d_{Bayes}) \leq R_{sym}(d)$. 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 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:

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

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 [33]).

Analogously to [16], 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 [30], [31]. 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 [30]. For instance, instead of computing logistic least absolute shrinkage and selection operator (LASSO [34]) regression fits, we use gradient-boosting trees (GB [35]). 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 $k = \{(h_s, v_s) \in \mathcal{Z}, s = 1, \dots, |k|\} \in \hat{\mathcal{K}}_x$ being the pixel coordinates of one *segment* (or *connected component*)

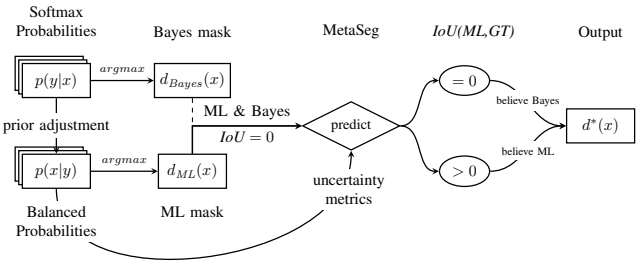


Fig. 1. Overview of our method for controlled false negative reduction of minority classes which we term "*MetaFusion*". Note that *IoU* denotes the intersection over union measure of two segmentation masks.

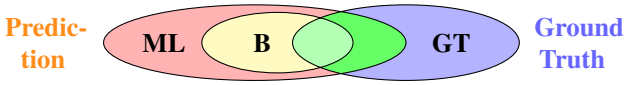


Fig. 2. Graphical illustration of the relation between Bayes and ML prediction segments for rare classes.

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

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

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 [30] 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.

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 [16], 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

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

that serve as input for the meta classifier, see also section III and cf. [30], [31]. The classifier we use in our meta model is the gradient-boosting tree algorithm (GB [35]) 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

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

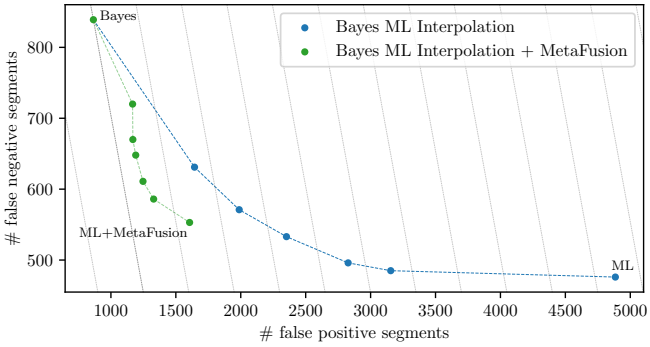


Fig. 3. False positives vs. false negatives of human segments for MobileNetV2 on Cityscapes. The blue curve is obtained by applying adjusted decision rules according to equation (16) with varied degrees of prior interpolation α. For each of the points given on the blue curve we additionally apply MetaFusion 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.

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 [9] 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 [36] with MobileNetV2 [37] and Xception65 [38] 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 I.

TABLE I

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)).

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**

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

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

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 $FN(d_{Bayes}) - FN(d) \neq 0$, 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' 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

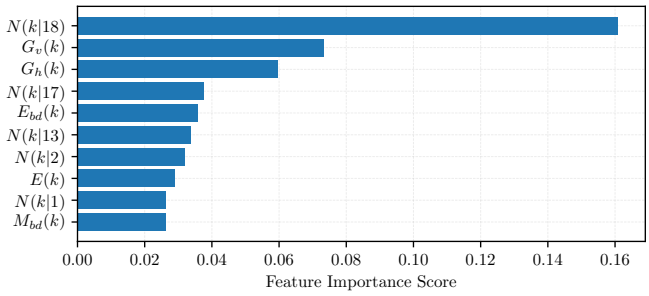


Fig. 4. Feature importance scores of the gradient-boosting classifier for MobileNetV2 applied to all disjoint ML and Bayes human segments. The score is averaged over all random cross-validation splits and only the ten features with the highest score are stated. In total we used $q = 56$ metrics as meta 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.

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

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

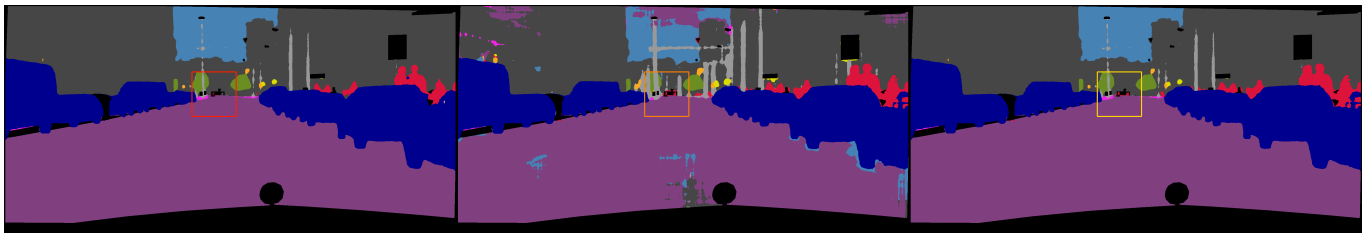
with $Q(t)$ the Gini impurity [35] and $n(t)$ the weighted number of samples in node $t \in \mathcal{T}$ (the weighting corresponds to the portion of all samples reaching node t). Moreover, by *left* and *right* we denote the respective child nodes. The importance for $\hat{f}$ of feature / uncertainty metric $m \in [0,1]$ is then computed as

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

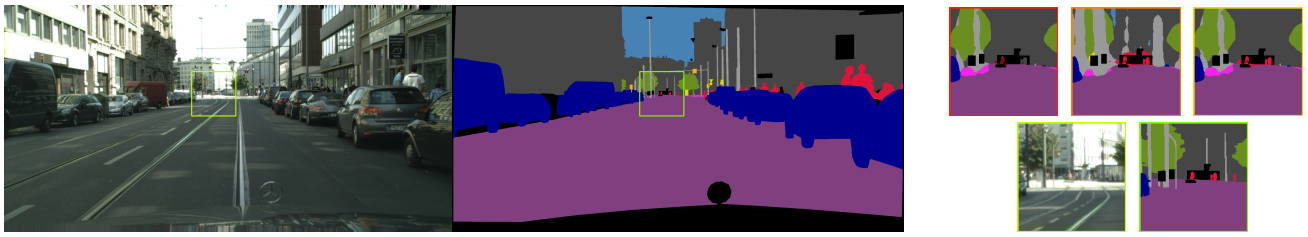
with

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

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



(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

Fig. 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.

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 [30], 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 [30], 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 J.D. Schneider and M. Fahrland for fruitful discussions.

References

[1] J. M. Johnson and T. M. Khoshgoftaar, "Survey on deep learning with class imbalance," *Journal of Big Data*, vol. 6, no. 1, p. 27, Mar 2019. [Online]. Available: https://doi.org/10.1186/s40537-019-0192-5 1

[2] M. Buda, A. Maki, and M. A. Mazurowski, "A systematic study of the class imbalance problem in convolutional neural networks," *Neural Networks*, vol. 106, pp. 249 – 259, 2018. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0893608018302107 1

[3] V. López, A. Fernández, S. García, V. Palade, and F. Herrera, "An insight into classification with imbalanced data: Empirical results and current trends on using data intrinsic characteristics," *Information Sciences*, vol. 250, pp. 113 – 141, 2013. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0020025513005124 1

[4] B. Krawczyk, "Learning from imbalanced data: open challenges and future directions," *Progress in Artificial Intelligence*, vol. 5, no. 4, pp. 221–232, Nov 2016. [Online]. Available: https://doi.org/10.1007/s13748-016-0094-0 1

[5] J. Van Hulse, T. M. Khoshgoftaar, and A. Napolitano, "Experimental perspectives on learning from imbalanced data," in *Proceedings of the 24th International Conference on Machine Learning*, ser. ICML '07. New York, NY, USA: ACM, 2007, pp. 935–942. [Online]. Available: http://doi.acm.org/10.1145/1273496.1273614 1

[6] D. Masko and P. Hensman, "The impact of imbalanced training data for convolutional neural networks," Ph.D. dissertation, KTH Royal School of Technology, 2015. 1

[7] N. V. Chawla, N. Japkowicz, and A. Kotcz, "Editorial: Special issue on learning from imbalanced data sets," *SIGKDD Explor. Newsl.*, vol. 6, no. 1, pp. 1–6, Jun. 2004. [Online]. Available: http://doi.acm.org/10.1145/1007730.1007733 1

[8] N. Chawla, K. Bowyer, L. O. Hall, and W. Philip Kegelmeyer, "Smote: Synthetic minority over-sampling technique," *J. Artif. Intell. Res. (JAIR)*, vol. 16, pp. 321–357, 01 2002. 1

[9] M. Cordts, M. Omran, S. Ramos *et al.*, "The cityscapes dataset for semantic urban scene understanding," in *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 1, 2, 5

[10] S. R. Bulò, G. Neuhold, and P. Kontschieder, "Loss max-pooling for semantic image segmentation," *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 7082–7091, 2017. 1

[11] H. Caesar, J. Uijlings, and V. Ferrari, "Joint calibration for semantic segmentation," in *Proceedings of the British Machine Vision Conference (BMVC)*, M. W. J. Xianghua Xie and G. K. L. Tam, Eds. BMVA Press, September 2015, pp. 29.1–29.13. [Online]. Available: https://dx.doi.org/10.5244/C.29.29 1

[12] 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. 1

[13] 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*, vol. 29, no. 8, pp. 3573–3587, Aug 2018. 1

[14] 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. 1, 2

[15] R. Chan, M. Rottmann, R. Dardashti *et al.*, "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, 2

[16] R. Chan, M. Rottmann, F. Hüger, P. Schlicht, and H. Gottschalk, "Application of decision rules for handling class imbalance in semantic segmentation," *CoRR*, vol. abs/1901.08394, 2019. [Online]. Available: http://arxiv.org/abs/1901.08394 1, 2, 3, 4

[17] L. Fahrmeir, A. Hamerle, and W. Häussler, *Multivariate statistische Verfahren (in German)*, 2nd ed. Walter De Gruyter, 1996. 2, 3

[18] M. Everingham, S. M. A. Eslami, L. Van Gool *et al.*, "The pascal visual object classes challenge: A retrospective," *International Journal of Computer Vision*, vol. 111, no. 1, pp. 98–136, Jan 2015. [Online]. Available: https://doi.org/10.1007/s11263-014-0733-5 2

[19] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, "Vision meets robotics: The kitti dataset," *International Journal of Robotics Research (IJRR)*, 2013. 2

[20] D. Hendrycks and K. 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. [Online]. Available: https://openreview.net/forum?id=Hkg4TI9xl 2

[21] R. M. Neal, *Bayesian learning for neural networks*. Springer Science & Business Media, 2012, vol. 118. 2

[22] Y. Gal and Z. Ghahramani, "Dropout as a bayesian approximation: Representing model uncertainty in deep learning," in *Proceedings of The 33rd International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, vol. 48. New York, New York, USA: PMLR, 20–22 Jun 2016, pp. 1050–1059. [Online]. Available: http://proceedings.mlr.press/v48/gal16.html 2

[23] A. Atanov, A. Ashukha, D. Molchanov, K. Neklyudov, and D. Vetrov, "Uncertainty estimation via stochastic batch normalization," in *Advances in Neural Networks – ISNN 2019*, H. Lu, H. Tang, and Z. Wang, Eds. Cham: Springer International Publishing, 2019, pp. 261–269. 2

[24] T. DeVries and G. W. Taylor, "Learning confidence for out-of-distribution detection in neural networks," *arXiv preprint arXiv:1802.04865*, 2018. 2

[25] A. Kendall and Y. 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. 2

[26] D. Amodei, C. Olah, J. Steinhardt, P. F. Christiano, J. Schulman, and D. Mané, "Concrete problems in AI safety," *CoRR*, vol. abs/1606.06565, 2016. [Online]. Available: http://arxiv.org/abs/1606.06565 2

[27] T. Kohlberger, V. Singh, C. 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. 2

[28] 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. 2

[29] T. DeVries and G. W. Taylor, "Leveraging uncertainty estimates for predicting segmentation quality," *CoRR*, vol. abs/1807.00502, 2018. [Online]. Available: http://arxiv.org/abs/1807.00502 2

[30] M. Rottmann, P. Colling, T. Hack *et al.*, "Prediction error meta classification in semantic segmentation: Detection via aggregated dispersion measures of softmax probabilities," *CoRR*, vol. abs/1811.00648, 2018. [Online]. Available: http://arxiv.org/abs/1811.00648 2, 3, 4, 7

[31] M. Rottmann and M. Schubert, "Uncertainty measures and prediction quality rating for the semantic segmentation of nested multi resolution street scene images," *CoRR*, vol. abs/1904.04516, 2019. [Online]. Available: http://arxiv.org/abs/1904.04516 2, 3, 4

[32] K. Maag, M. Rottmann, and H. Gottschalk, "Time-dynamic estimates of the reliability of deep semantic segmentation networks," *CoRR*, vol. abs/1911.05075, 2019. [Online]. Available: http://arxiv.org/abs/1911.05075 2

[33] J. Joyce, "Bayes' theorem," in *The Stanford Encyclopedia of Philosophy*, E. N. Zalta, Ed. Metaphysics Research Lab, Stanford University, 2019. 3

[34] R. Tibshirani, "Regression shrinkage and selection via the lasso," *Journal of the Royal Statistical Society: Series B*, vol. 58, pp. 267–288, 1996. [Online]. Available: https://www.bibsonomy.org/bibtex/290e648276aa6cd3c601e7c0a54366233/dieudonnew 3

[35] T. Hastie, R. Tibshirani, and J. Friedman, *The elements of statistical learning: data mining, inference and prediction*, 2nd ed. Springer, 2009. [Online]. Available: http://www-stat.stanford.edu/~tibs/ElemStatLearn/ 3, 4, 6

[36] L. C. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam, "Encoder-decoder with atrous separable convolution for semantic image segmentation," in *The European Conference on Computer Vision (ECCV)*, 9 2018. 5

[37] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "Mobilenetv2: Inverted residuals and linear bottlenecks," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 6 2018. 5

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