Bergische Universität Wuppertal

Fakultät für Mathematik und Naturwissenschaften

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

Preprint BUW-IMACM 20/58

Robin Chan, Matthias Rottmann und Hanno Gottschalk

Entropy Maximization and Meta Classification for Out-Of-Distribution Detection in Semantic Segmentation

December 9, 2020

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

Entropy Maximization and Meta Classification for Out-of-Distribution Detection in Semantic Segmentation

Robin Chan, Matthias Rottman and Hanno Gottschalk
University of Wuppertal, Germany
Faculty of Mathematics und Natural Sciences
{rchan,rottmann,hgottsch}@uni-wuppertal.de

Abstract

Deep neural networks (DNNs) for the semantic segmentation of images are usually trained to operate on a predefined closed set of object classes. This is in contrast to the "open world" setting where DNNs are envisioned to be deployed to. From a functional safety point of view, the ability to detect so-called "out-of-distribution" (OoD) samples, i.e., objects outside of a DNN's semantic space, is crucial for many applications such as automated driving. A natural baseline approach to OoD detection is to threshold on the pixel-wise softmax entropy. We present a two-step procedure that significantly improves that approach. Firstly, we utilize samples from the COCO dataset as OoD proxy and introduce a second training objective to maximize the softmax entropy on these samples. Starting from pretrained semantic segmentation networks we re-train a number of DNNs on different in-distribution datasets and consistently observe improved OoD detection performance when evaluating on completely disjoint OoD datasets. Secondly, we perform a transparent post-processing step to discard false positive OoD samples by so-called "meta classification". To this end, we apply linear models to a set of hand-crafted metrics derived from the DNN's softmax probabilities. In our experiments we consistently observe a clear additional gain in OoD detection performance, cutting down the number of detection errors by 52% when comparing the best baseline with our results. We achieve this improvement sacrificing only marginally in original segmentation performance. Therefore, our method contributes to safer DNNs with more reliable overall system performance.

1. Introduction

In recent years spectacular advances in the computer vision task semantic segmentation have been achieved by deep learning [43, 46]. Deep convolutional neural networks (CNNs) are envisioned to be deployed to real world applications, where they are likely to be exposed to data that is substantially different from the model's training data. We consider data samples that are not included in the set of a model's semantic space as *out-of-distribution* (OoD) samples. State-of-the-art neural networks for semantic segmentation, however, are trained to recognize a predefined closed set of object classes [11, 29], e.g. for the usage in environment perception systems of autonomous vehicles [22]. In open world settings there are countless possibly occurring objects. Defining additional classes requires a large amount of annotated data (cf. [10, 47]) and may even lead to performance drops [13]. One natural approach is to introduce a *none-of-the-known* output for objects not belonging to any of the predefined classes [45]. In other words, one uses a set of object classes that is sufficient for most scenarios and cover all OoD objects by enforcing a specific model output for such samples. This additional output can be imple-

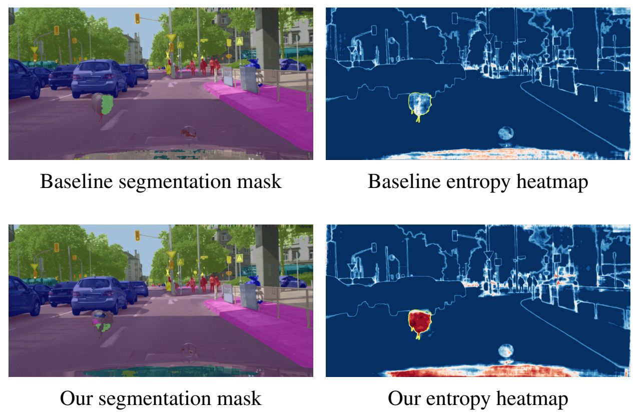


Figure 1: Comparison of segmentation mask and softmax entropy before our OoD training (*top row*) and after (*bottom row*). While there are minor differences in the segmentation masks, the annotated unknown object (marked with yellow lines) becomes clearly recognizable in the entropy heatmap due to our OoD training. In the heatamp high values are red.

mented by introducing an additional class or by setting a threshold on the softmax entropy as well as any other dispersion or uncertainty measure. From a functional safety point of view, it is a crucial but yet missing prerequisite that neural networks are capable of reliably indicating when they are operating out of their proper domain, i.e., detecting OoD objects, in order to initiate a fallback policy.

As images from everyday scenes usually contain many different objects, of which only some could be out-of-distribution, knowing the location where the OoD object occurs is desired for practical application. Therefore, we address the problem of detecting anomalous regions in an image, which is the case if an OoD object is present (see figure 1) and which is a research area of high interest [5, 18, 30, 39].

This so-called *anomaly segmentation* [4, 18] can be pursued, for instance, by incorporating sophisticated uncertainty estimates [2, 16] or by adding an extra class to the model's learnable set of classes [45].

In this work, we detect OoD objects in semantic segmentation with a different approach which is composed of two steps: As fist step, we re-train the segmentation network to predict class labels with low confidence scores on OoD inputs by enforcing the segmentation model to output high prediction uncertainty. In order to quantify uncertainty, we compute the softmax entropy which is maximized when a model outputs uniform probability scores over all classes [27]. By deliberately including annotated OoD objects as *known unknowns* into the training process and employing a modified multi-objective loss function, we observe that the semantic segmentation network generalizes learnt uncertainty to unseen OoD samples (*unknown unknowns*) without significantly sacrificing in original performance on the primary task, see figure 1.

The primal model for semantic segmentation is trained on the Cityscapes data [11]. As proxy for OoD samples we randomly pick images from the COCO dataset [29] excluding the ones with instances that are also available in Cityscapes, cf. [17, 20, 34] for a related approach in image classification. We evaluate the pixel-wise OoD detection performance via entropy thresholding for OoD samples from the LostAndFound [39] and Fishyscapes [5] dataset, respectively. Both datasets share the same setup as Cityscapes but include OoD road obstacles.

The second step incorporates a *meta classifier* flagging incorrect class predictions at segment level, similar as proposed in [31, 41, 42] for the detection of false positive instances in semantic segmentation. After increasing the sensitivity towards predicting OoD objects, we aim at removing false predictions which are produced due to the preceding entropy boost (cf. [8]). The removal of false positive OoD object predictions is based on aggregated dispersion measures and geometry features within segments (connected components of pixels), with all information derived solely from the neural network's softmax output. As meta classifier we employ a simple linear model allowing to track and understand the impact of each metric.

To sum up our contributions, we show that only a little modification of training is required to make semantic segmentation networks much more sensitive to the detection of OoD samples. Re-training segmentation networks with a specific choice of OoD images from COCO [29] clearly outperforms the natural baseline approach of plain softmax entropy thresholding [19] by up to 73 percent points in average precision. In addition, we are the first to demonstrate that entropy based OoD object predictions in semantic segmentation can be meta classified reliably, i.e., classified whether one considered OoD prediction is true positive or false positive without access to the ground truth. For this task we employ simple logistic regression. Combining entropy maximization and meta classification therefore is an efficient and yet lightweight method, particularly suitable as an integrated monitoring system of safety-critical real world applications based on deep learning.

2. Related Work

Methods from prior works have already proven their efficiency in identifying OoD input for image data. The proposed methods are either modifications of the training procedure [17, 20, 27, 28, 34] or post-processing techniques adjusting the estimated confidence [14, 19, 27]. However, most of these works treat entire images as out-of-distribution.

When considering the semantic space to be fixed, anomaly segmentation, i.e., treating pixels as OoD, is necessarily based on estimates of uncertainty for neural networks. Early approaches to uncertainty estimation involve Bayesian neural networks (BNNs) yielding posterior distributions over the model's weight parameters [32, 37]. In practice, approximations such as Monte-Carlo dropout [16] or stochastic batch normalization [2] are mainly used due to cheaper computational cost. Frameworks using dropout for uncertainty estimation applied to semantic segmentation have been developed in [3, 24]. Other approaches to model uncertainty consist of using an ensemble of neural networks [26], which captures model uncertainty by averaging predictions over multiple models, and density estimation [5, 9, 36, 40] via estimating the likelihood of samples with respect to the training distribution. Methods for OoD detection in semantic segmentation based on label-prediction (or classification) uncertainty have been analyzed in [6, 21, 23, 30, 33, 38].

Using BNNs for estimating uncertainty in deep neural networks is associated with prohibitive computational cost. Uncertainty estimates that are generated by multiple models or by multiple forward passes are still computationally expensive compared to single inference based ones. In our approach, we unite semantic segmentation and OoD detection in one model without any modifications of the underlying network's architecture. Therefore, our re-training approach can be even combined with existing OoD detection techniques and potentially enhance their efficiency.

Works with similar training approaches as ours use a different OoD proxy and are presented in [5, 23]. They train neural networks on the unlabeled objects in Cityscapes as OoD approximation. The training process includes only one single dataset, but in our experiments we observe that the unlabeled data lacks in diversity and therefore tends to be too dataset specific. With respect to other OoD datasets, such as LostAndFound and Fishyscapes on which we perform our experiments, we observe in our tests that these methods fail to generalize. Furthermore, in contrast to those works we incorporate a post-processing step that significantly leverages the OoD detection performance.

Another line of work detects OoD samples in semantic segmentation by incorporating autoencoders [1, 4, 12, 30]. Training such a model only on specific samples from a closed set of classes, it is assumed that the autoencoder model performs less accurately when fed with samples from never-seen-before classes. The identification of an OoD sample then relies on the reconstruction quality. In this way, no OoD data is required, except for further adjusting the sensitivity of the method.

Autoencoders are in fact deep neural networks themselves. For the goal of safe real-time semantic segmentation, e.g., necessary for automated driving [22], more lightweight approaches are favorable. We avoid incorporating deep auxiliary models at all and only employ a lightweight linear model instead. Furthermore, usually the more complex a model, the greater the lack of interpretability. As monitoring systems are supposed to make deep learning models safer, one seeks for simpler and thereby more explainable approaches. We post-process our entropy boosted semantic segmentation network output via logistic regression whose computational overhead is negligible. This linear model is transparent as it allows us to analyze the impact of each single feature fed into the model and it demonstrates in our experiments to efficiently reduce the number of OoD detection errors.

3. Entropy based OoD Detection

In this section, we present our training strategy to improve the detection of OoD pixels in semantic segmentation via spatial entropy heatmaps.

3.1. Training for high Entropy on OoD Samples

Let $f(x) \in (0,1)^q$ be the softmax probabilities after processing the input image $x \in \mathcal{X}$ with some machine learning model $f : \mathcal{X} \to (0,1)^q$ and let $q \in \mathbb{N}$ be the number of classes. For the sake of brevity, we omit the consideration of image pixels in this section. We compute the softmax entropy via

$$E(f(x)) = -\sum_{j=1}^{q} f_j(x) \log(f_j(x)) \,. \tag{1}$$

By $(x, y(x)) \sim \mathcal{D}_{in}$ we denote an "in-distribution" sample with $y(x) \in \{1,\ldots,q\}$ being its corresponding ground truth class label, by $x' \sim \mathcal{D}_{out}$ we denote an "out-distribution" sample for which there is no label given. We aim at minimizing the overall objective

$$\begin{aligned} L := (1-\lambda)\, & \mathbb{E}_{(x,y)\sim\mathcal{D}_{in}} [\ell_{in}(f(x), y(x))] \\ & + \lambda\, \mathbb{E}_{x'\sim\mathcal{D}_{out}} [\ell_{out}(f(x'))] \ , \ \lambda \in [0,1] \end{aligned} \tag{2}$$

where

$$\ell_{in}(f(x), y(x)) := -\sum_{j=1}^{q} \mathbb{1}_{j=y(x)} \log(f_j(x)) \ \text{ and } \tag{3}$$

$$\ell_{out}(f(x')) := -\sum_{j=1}^{q} \frac{1}{q} \log(f_j(x')) \tag{4}$$

with the indicator function $\mathbb{1}_{j=y(x)} \in \{0,1\}$ being equal to one if $j = y(x)$ and zero else. In other words, for in-distribution samples we apply the commonly used empirical cross entropy loss, i.e., the negative log likelihood of the target class. For out-distribution samples, we consider the negative log likelihood for each class, weighted inverse proportionally to the number of classes.

By that choice of out-distribution loss function, minimizing $\ell_{out}(f(x'))$ is equivalent to maximizing the softmax entropy $E(f(x))$, see equation (1). Since the softmax definition implies $f_j(x) \in (0,1)$ and $\sum_{j=1}^{q} f_j(x) = 1$, Jensen's inequality applied to the convex function $-\log(\cdot)$ yields

$$\ell_{out}(f(x)) \geq -\log\left(\sum_{i=1}^{q} \frac{1}{q} f_j(x)\right) = \log(q) \tag{5}$$

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

$$E(f(x)) \leq \log\left(\sum_{i=1}^{q} f_j(x) \frac{1}{f_j(x)}\right) = \log(q) \tag{6}$$

with equality (for both inequalities inequalities (5) and (6), respectively) if $f_j(x) = 1/q \ \forall\, j = 1,\ldots,q$, i.e., if the softmax probabilities are uniformly distributed over all classes.

In order to control the impact of each single objective on the overall objective L, the convex combination between expected in-distribution loss and expected out-distribution loss is included in equation (2) which can be adjusted by varying the parameter λ.

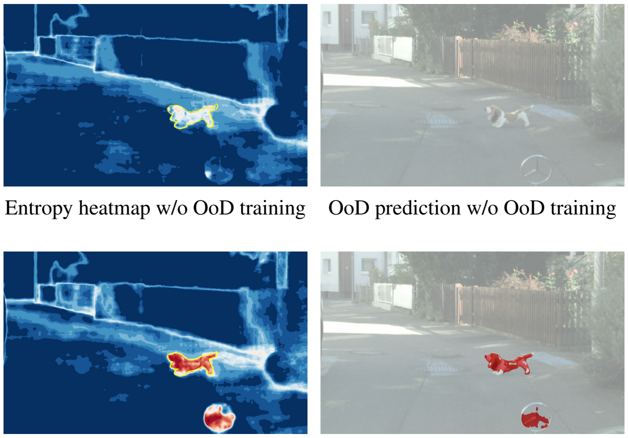


Figure 2: Comparison of softmax entropy heatmap and OoD prediction mask with our OoD training (*top row*) and without (*bottom row*). The yellow lines in the entropy heatmaps mark the annotation of the OoD object. The OoD object prediction is obtained by simply thresholding on the entropy heatmap (in this example at $t = 0.7$ yielding the red pixels in the OoD prediction masks).

3.2. OoD Object Prediction in Semantic Segmentation via Entropy Thresholding

The softmax probabilities output of neural networks for semantic segmentation $f(x) \in (0,1)^{|\mathcal{H}|\times|\mathcal{W}|\times q}, x \in \mathcal{X} \subseteq [0,1]^{|\mathcal{H}|\times|\mathcal{W}|\times 3}$, can be viewed as pixel-wise probability distributions that express how likely each potential class affiliation $j = 1, \dots, q$ of a given pixel $z \in \mathcal{H} \times \mathcal{W}$ is, according to the model f. Let $f^z(x) \in (0,1)^q$ denote the softmax output in pixel location z which we implicitly considered throughout the previous section. In semantic segmentation one minimizes the averaged pixel-wise classification loss over the image, cf. equation (2). For the sake of simplicity, we consider the normalized entropy $\bar{E}(f^z(x))$ at pixel location z in the following, which we obtain by dividing $E(f^z(x))$ by $\log(q)^{-1}$. One pixel is then assumed to be out-of-distribution if the normalized entropy $\bar{E}(f^z(x))$ at that pixel location z is greater than a chosen threshold $t \in [0,1]$, i.e., *z is predicted to be OoD* if

$$z \in \hat{\mathcal{Z}}_{out}(x) := \{z' \in \mathcal{H} \times \mathcal{W} : \bar{E}(f^{z'}(x)) \geq t\} \, . \quad (7)$$

A connected component $k \in \hat{\mathcal{K}}(x) \subseteq \mathcal{P}(\hat{\mathcal{Z}}_{out}(x))$ (the latter being the power set of $\hat{\mathcal{Z}}_{out}(x)$) consisting of neighboring pixels fulfilling the condition in equation (7) gives us an *OoD segment / object prediction*. An illustration can be viewed in figure 2. Obviously, the better an in-distribution pixel can be separated from an out-distribution pixel by means of the entropy, the more accurate the OoD object prediction will be.

4. Meta Classifier in Semantic Segmentation

By training the segmentation network to output uniform confidence scores as presented in section 3, we increase the sensitivity towards predicting OoD objects, aiming for an "entropy boost" on OoD samples. However, it is not guaranteed that only OoD samples have a high entropy. Therefore, detecting OoD samples via entropy boosting potentially comes along with a considerable number of false OoD predictions, resulting in an unfavorable trade-off.

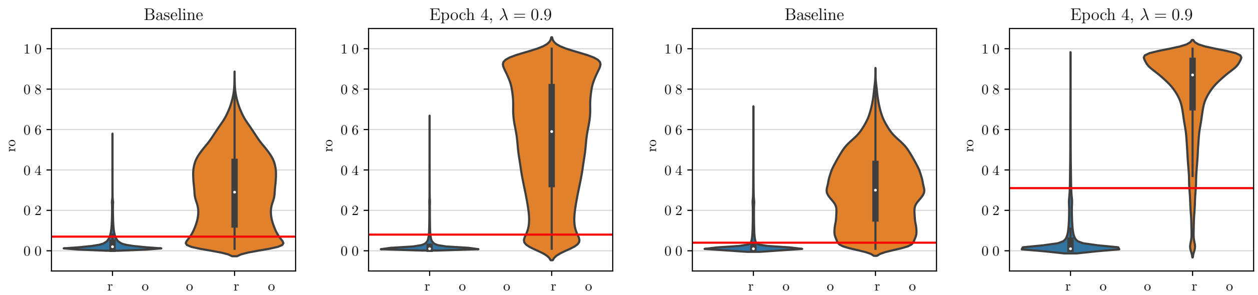
In this context, we consider one entire OoD object prediction (see section 3.2) as true positive if its intersection over union (IoU, [15]) with a ground truth OoD object is greater than zero. More formally, let $\mathcal{Z}_{out}(x)$ be the set of pixel locations in x which are labeled OoD according to ground truth. Then $k \in \hat{\mathcal{K}}(x)$ *is true positive* (TP) if

$$\begin{aligned} & \mathrm{IoU}(k, \mathcal{Z}_{out}(x)) > 0 \\ \Leftrightarrow \; & \exists \, z \in k : \bar{E}(f^z(x)) \geq t \wedge z \in \mathcal{Z}_{out}(x) \, . \end{aligned} \quad (8)$$

In [8] it has been demonstrated that false-positives due to increased prediction sensitivity can be removed based on a meta classifier's decision, achieving improved trade-offs between error rates. This meta classifier is essentially a binary classification model added on top of an underlying segmentation network [31, 41, 42]. We construct hand-crafted metrics per connected component of pixels by aggregating different pixel-wise uncertainty measures derived from the softmax probabilities, one of which is the entropy. The entropy metric has proven to be highly correlated to the segment-wise IoU and therefore contributes greatly to the meta classifier's performance, cf. [41]. Different to existing approaches, that consider neighboring pixels sharing the same class label as segment, we generate metrics per segment above the given entropy threshold t. Given the importance of entropy for meta classifiers in combination with entropy based segment generation, we expect the learned entropy maximization on OoD objects to leverage the meta classification performance.

Given the softmax output, we include further pixel-wise dispersion measures such as the probability margin, the difference between highest and second highest softmax probability, and variation ratio, the maximum softmax probability. They all have proven their efficiency in terms of meta classification performance [8, 31, 42]. Moreover, we also consider geometry features, such as the segment's size or its ratio between interior and boundary [41]. These metrics serve as inputs for the auxiliary *meta* model that classifies into *true positive* and *false positive* (FP) OoD object prediction, i.e., classifying $k \in \hat{\mathcal{K}}(x)$ into the classes / sets

$$\begin{aligned} C_1 &:= \{k' \in \hat{\mathcal{K}}(x) : \mathrm{IoU}(k, \mathcal{Z}_{out}(x)) > 0\} \;\text{ and} \\ C_2 &:= \{k' \in \hat{\mathcal{K}}(x) : \mathrm{IoU}(k, \mathcal{Z}_{out}(x)) = 0\} \, . \end{aligned} \quad (9)$$



(a) LostAndFound (b) Fishyscapes

Figure 3: Relative pixel frequencies of LostAndFound (a) and Fishyscapes (b) OoD pixels, respectively, at different entropy values for the baseline model, i.e., before OoD training (*a & b left*), and after OoD training (*a & b right*). The red lines indicate the thresholds of highest accuracy. See also appendix A for more details and see appendix E for a visualization.

The outlined hand-crafted metrics form a structured dataset of features where the rows correspond to predicted segments and the columns to metrics, see also appendix B.

5. Setup of Experiments

Semantic segmentation is one of the basic components in environment perception systems of autonomous vehicles [22]. We therefore consider the semantic segmentation for the Cityscapes dataset [11] as original task, i.e., we consider Cityscapes as in-distribution data $\mathcal{D}_{in}$. The (standard) training split consists of 2,975 pixel-annotated urban street scene images. As original model, we use the state-of-the-art semantic segmentation DeepLabv3+ model with a WideResNet38 backbone trained by Nvidia [46]. This model is initialized with publicly available weights and serves as our *baseline* model. For testing, we evaluate the OoD detection performance on two datasets comprising street scene images and unexpected obstacles. We consider images from the LostAndFound test split [39], containing 1,203 images with annotations of small obstacles and road in front of the (ego-)car, and Fishyscapes Static [5], containing 30 images with annotated anomalous objects extracted from Pascal VOC [15] which are then overlayed in Cityscapes images. Both datasets share the same setup as Cityscapes but include small road obstacles.

In order to perform the *OoD training* as proposed in section 3.1, we approximate the out-distribution via images from the COCO [29] dataset. This dataset contains images of everyday objects captured from everyday scenes. Besides that, we only consider COCO images with instances that are not included in Cityscapes (no persons, no cars, no traffic lights, ...) and images that have a minimum height and width of at least 480 pixels. After filtering, there remain 1,489 images serving as our proxy for $\mathcal{D}_{out}$ (see appendix C for experiments with another OoD proxy).

We finetune the DeepLabv3+ model with loss functions according to equation (3) and equation (4). As training data we randomly sample 297 images from our COCO subset per epoch and mix them into all 2,975 Cityscapes training images (1:10 ratio of out-distribution to in-distribution images). We train the model's weight parameters on random crops of size 480 pixels for 4 epochs in total and set the (out-distribution) loss weight $\lambda = 0.9$ (see equation (2)). As optimizer we use Adam [25] with a learning rate of 10^{-5}.

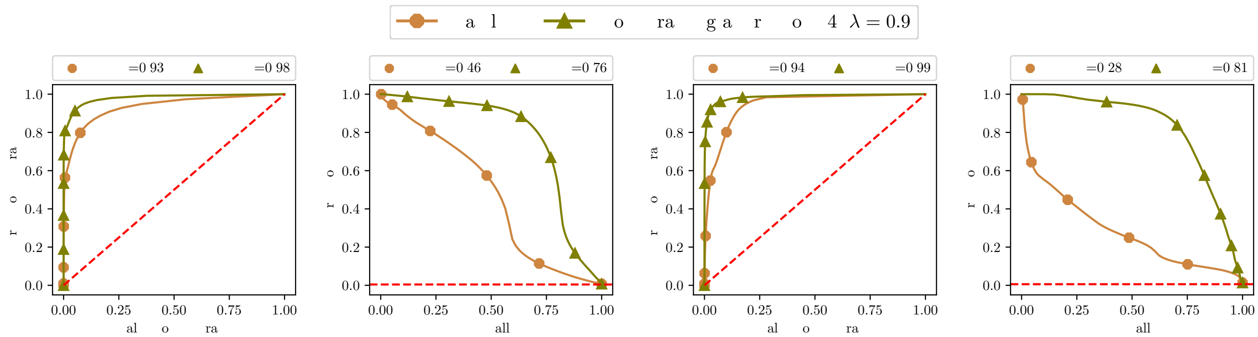
6. Pixel-wise Evaluation

Based on the softmax probabilities, we compute the normalized entropy $\bar{E}$ for all pixels in the respective test dataset. This gives us a per-pixel anomaly / OoD score which we compare with the ground truth anomaly segmentation. For the sake of clarity, in this section we refer to in-distribution pixels as *samples of the negative class* and to out-distribution pixels as *samples of the positive class*. We emphasize that none of the OoD objects in the test data have been seen during our OoD training since we use separate datasets for training and testing, with different objects corresponding to completely disjoint semantic class labels.

6.1. Separability by means of AUROC

On basis of the violin plots in figure 3, one already notices the improved separability of in-distribution and out-distribution pixels as large masses of the distributions corresponding to the respective classes can be well separated via multiple entropy thresholds. One also notices, that our OoD training is beneficial with respect to the separability. This effect can be further quantified with the aid of receiver operating characteristic (ROC) curves, see figure 4 (a) & (b) left. The area under the curve (AUC) of ROC curves (AUROC) then represents the degree of separability. The higher the AUC, the better the separability.

By comparing the ROC curves for LostAndFound (figure 4 (a)), we observe that there is a performance gain over



(a) LostAndFound (*left:* AUROC, *right:* AUPRC) (b) Fishyscapes (*left:* AUROC, *right:* AUPRC)

Figure 4: Detection ability for LostAndFound (a) and Fishyscapes (b) OoD pixels, respectively, evaluated by means of receiver operating characteristic curve (*a & b left*) and precision recall curve (*a & b right*). The red lines indicate the performance according to random guessing.

the baseline model when OoD training is applied. The baseline curve indicates that the corresponding model has a lower true positive rate across various fixed false positive rates, i.e., our model after OoD training assigns higher uncertainty / entropy values to OoD samples which is beneficial for OoD detection. Although the AUC of the baseline model is already decent at 0.93, we outperform this AUC significantly with a value of 0.98, which is $5/7$ of what we could gain due to our OoD training.

We observe the same effect for Fishyscapes. From the violins already, the discrimination performance seems already close to perfect due our OoD training. This is confirmed by means of the corresponding ROC curve as the AUC score has increased up to 0.99. In comparison to the baseline model, there is a gain of 5 percent points which makes considerable $5/6$ of the possible performance gain.

6.2. Separability by means of AUPRC

As the AUROC essentially measures the overlap of distributions corresponding to negative and positive samples, this score does not place more emphasis on one class over the other in case of class imbalance. In both our test datasets, there is a considerably strong class imbalance, 0.7% and 1.3% OoD samples in LostAndFound and Fishyscapes, respectively. Therefore, we additionally measure the separability by means of precision recall curves (PRC), see figure 4 (a) & (b) right, thus ignoring true negatives and emphasizing the detection of the positive class / OoD samples. Now the AUC of PRC (AUPRC) serves as measure of separability.

For LostAndFound as well as for Fishyscapes objects the re-trained model is superior over the baseline model in terms of precision when we fix recall to any score. The AUC quantifies this performance gain and thus further clarifies the improved capability at detecting pixels corresponding to an OoD object. Regarding LostAndFound, the OoD training increases the AUC by 0.30 up to a score of 0.76. This is a relative change with respect to the baseline model of roughly 66%. Regarding Fishyscapes, the performance gain is even more significant. We raise the AUC from 0.28 up to 0.81, which is nearly a threefold performance increase. We conclude that measured by AUPRC scores our OoD training is highly beneficial for detecting OoD samples.

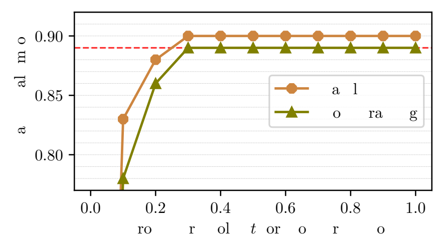


Figure 5: Mean intersection over union (mIoU) for Cityscapes validation split with OoD predictions at different entropy thresholds t. The dashed red line indicates the performance loss that we consider to be "acceptable" ($\sim 1\%$).

6.3. Original Task Performance

In order to monitor that the baseline model does not unlearn its original task due to OoD training, we evaluate the model's performance on in-distribution data with OoD predictions at different entropy thresholds. The original task is the semantic segmentation of the Cityscapes images and we evaluate by means of the most commonly used performance metric *mean Intersection over Union* (mIoU, [15]). Additionally to the Cityscapes class predictions, that is obtained via the standard maximum a posteriori (MAP) decision principle [7, 35], we consider an extra OoD class prediction if the softmax entropy is above the given threshold. We compute the mIoU for the Cityscapes validation dataset, but average only over the 19 Cityscapes class IoUs.

Method	AUROC ↑	FPR_{95} ↓	AUPRC ↑	mIoU ↑
	LostAndFound Test			Cityscapes Val.
Lis et al. [30] Best	0.93	-	-	0.80
Li et al. [44] Plain	0.91	0.30	0.36	0.80
Zhu et al. [46] Plain	0.93	0.35	0.46	**0.90**
Ours: Li et al. + OoD T.	0.94	0.12	0.51	0.76
Ours: Zhu et. al + OoD T.	**0.98**	**0.09**	**0.76**	0.89
	Fishyscapes Static			Cityscapes Val.
Blum et al. [5] Best	-	0.13	0.62	0.80
Li et al. [44] Plain	0.85	0.46	0.07	0.80
Zhu et al. [46] Plain	0.94	0.18	0.28	**0.90**
Ours: Li et al. + OoD T.	0.94	0.21	0.38	0.76
Ours: Zhu et. al + OoD T.	**0.99**	**0.05**	**0.81**	0.89

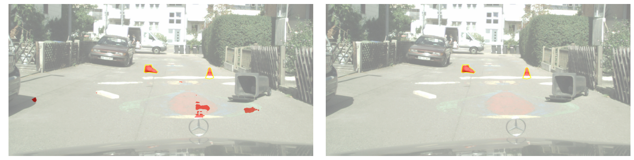
Table 1: Benchmark results for LostAndFound and Fishyscapes with DeepLabv3+ (Zhu et al. [46]) and a slightly weaker DualGCNNet (Li et al. [44]) CNNs. The gray rows mark scores with OoD training, otherwise only entropy thresholding is applied (Plain). For comparison, we included scores reported in [30] and [5] which are, to the best of our knowledge, the only works comparable to ours.

The state-of-the-art DeepLabv3+ model [46], which serves as our baseline throughout our experiments, achieves an mIoU of 0.90 on the Cityscapes validation dataset without OoD predictions (implying $t = 1.0$). By re-training the neural network with entropy maximization on OoD inputs, we observe improved OoD-AUPRC scores over the course of training peaking at 0.76. This gain at detecting OoD samples in LostAndFound comes with a marginal loss in Cityscapes validation mIoU down to 0.89. These two mIoU scores remain nearly constant (deviations less than 1 percent point) for the evaluated thresholds $t = 0.3, \ldots, 1.0$. In general, the lower the entropy threshold, the more pixels are predicted to be OoD. For $t = 0.2$ this results in noticeable performance decrease, 0.05 for the baseline model and 0.03 for the re-trained model, respectively. As displayed in figure 5 further lowering the threshold leads to even more significant sacrifice of original performance. Consequently, we consider in the following experiments entropy thresholds of $t \geq 0.3$ as the performance loss seems acceptable, especially in view of substantially improved OoD detection capability. We refer to appendix F for more details.

All the results presented in this section are summarized in table 1 where we additionally provide the false positive rates at 95% true positive rate (FPR_{95}). Moreover, we conducted the same experiments as for the DeepLabv3+ model [46] also for the weaker DualGCNNet [44], see appendix D. We re-trained the latter model with $\lambda = 0.25$ for 11 epochs in total and report the scores in the table 1 as well.

7. Segment-wise Evaluation

In this section we evaluate the meta classification performance on LostAndFound. The main metrics of the segment



OoD Training | OoD Training + meta classifier

Figure 6: OoD detection with $t = 0.7$ after OoD training and meta classification. The yellow lines mark the annotations of the OoD objects. OoD predictions labeled as background area according to the ground truth are ignored (this includes e.g. the garbage bin). See appendix G for more examples.

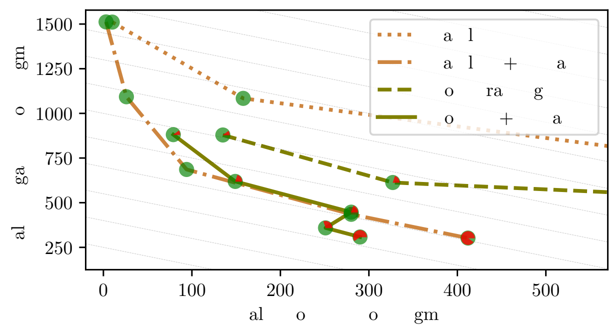


Figure 7: Detection errors of LostAndFound OoD objects. In this plot, the number of errors when $t = 0.7, \ldots, 0.3$ are displayed (when in the axes' range). The pie-chart markers indicate the road miss rate ε, being entirely red if $\varepsilon \geq 0.001$. See also table 2 for exact numbers.

-wise evaluation are the numbers of FPs and FNs with respect to an OoD object prediction, cf. equation (8). As the removal of FP OoD predictions should not come at cost of a significant loss in original performance, see figure 6, we additionally consider the *miss rate of road pixels*:

$$\varepsilon := 1 - \left| \bigcup_{x \in \mathcal{X}} \left(\hat{\mathcal{Z}}_{in}(x) \cap \mathcal{Z}_{in}(x) \right) \right| \left| \bigcup_{x \in \mathcal{X}} \mathcal{Z}_{in}(x) \right|^{-1} \quad (10)$$

with pixel locations predicted to be in-distribution in $\hat{\mathcal{Z}}_{in}$ and annotated as in-distribution in $\mathcal{Z}_{in}$. The road miss rate ϵ measures the proportion of actual road pixels in the whole dataset which are incorrectly identified.

We compute per-segment metrics as outlined in section 4 for OoD object predictions in the LostAndFound test set and feed them through meta classification models, which are simple logistic regressions throughout our experiments. The segments are then leave-one-out cross validated whether they are TP or FP, see equation (9).

We consistently observe a gain in OoD detection performance due to meta classification. The number of detection

Entropy Threshold	Baseline				Baseline + Meta Classifier				OoD Training				OoD Training + Meta Classifier			
$\bar{E} \geq t$	FP ↓	FN ↓	$\sum$ ↓	ε in % ↓	FP ↓	FN ↓	$\sum$ ↓	ε in % ↓	FP ↓	FN ↓	$\sum$ ↓	ε in % ↓	FP ↓	FN ↓	$\sum$ ↓	ε in % ↓
$t = 0.10$	33,584	77	33,661	7.60	386	314	700	3.24	21,967	99	8277	5.22	245	302	547	2.70
$t = 0.20$	19,456	136	19,592	2.48	454	307	761	0.93	17,000	127	17,127	2.14	288	316	604	0.18
$t = 0.30$	7,349	218	7,567	0.38	412	302	714	0.09	8,068	**191**	8,277	0.30	**290**	308	**598**	**0.06**
$t = 0.40$	3,214	377	3,591	0.08	280	435	715	0.03	4,035	**289**	4,324	0.11	**251**	359	**610**	**0.03**
$t = 0.50$	809	662	1,471	0.01	**94**	686	780	< **0.01**	1,215	**415**	1,630	0.04	280	447	**727**	0.02
$t = 0.60$	158	1,084	1,242	< 0.01	**26**	1,093	1,119	< **0.01**	327	**613**	940	0.02	149	619	**768**	0.02
$t = 0.70$	10	1,511	1,521	< 0.01	**3**	1,512	1,515	< **0.01**	135	**879**	1,014	0.01	79	881	**960**	0.01

Table 2: Detection errors for LostAndFound OoD objects at different entropy thresholds t. We consider the road miss rate ε, see equation (10), as further measure of loss in original performance (for Cityscapes mIoU, see figure 5). Below the horizontal line, i.e., $t \geq 0.3$, we consider the loss in original performance to be acceptable, see section 6.3 for further details.

Entropy Threshold t	Baseline + MSP [19] AUROC	AUPRC	Baseline + Meta C. AUROC	AUPRC	OoD T. + Meta C. AUROC	AUPRC
$t = 0.10$	0.8509	0.9817	0.9894	0.9993	0.9915	0.9993
$t = 0.20$	0.6470	0.9119	0.9859	0.9980	0.9898	0.9980
$t = 0.30$	0.5333	0.7376	0.9742	0.9884	**0.9847**	**0.9953**
$t = 0.40$	0.3847	0.4671	0.9715	0.9740	**0.9808**	**0.9807**
$t = 0.50$	0.4172	0.2286	0.9628	0.9214	**0.9665**	**0.9536**
$t = 0.60$	0.4906	0.1228	0.9291	0.7252	**0.9511**	**0.8405**
$t = 0.70$	0.5932	0.1334	0.9140	0.5283	**0.9444**	**0.7185**

Table 3: Meta classification performance on LostAndFound at different entropy thresholds t. As comparison to the meta classifier, we include the detection of OoD prediction errors via the maximum softmax probability (MSP, [19]).

errors as well as road miss rate ε at different entropy thresholds t are summarized in table 2. The performance of FP OoD removal is given in table 3.

In general, the higher the entropy threshold, the less OoD objects are predicted and consequently less data is fed through the linear models. This explains the observation that meta classifiers identify FPs more reliably the lower t. However, also for larger thresholds, meta classifiers still clearly outperform the natural maximum softmax probability (MSP, [19]) approach. Due to our OoD training, the meta classifiers demonstrate to be even more effective, being most superior when $t = 0.7$ with an AUPRC score of 0.72, which is 19 percent points higher than without OoD training. In our experiments, OoD training in combination with meta classification at $t = 0.3$ turns out to be the best OoD detection approach achieving the best result with only 598 errors in total while having a road miss rate of marginally 0.06%. In comparison, there are 7,567 errors at a road miss rate of 0.38% when applying neither OoD training nor meta classifiers, which can be reduced to decent scores of 714 and 0.09%, respectively, when adding the meta classifier. Compared with the best baseline at $t = 0.6$, we decrease the number of total errors by 52% from 1,242 down to 598. More safety-relevantly, at the same time we significantly reduce the number of overlooked OoD objects by 70% from 1,084 down to 308.

8. Conclusion & Outlook

In this work, we presented a novel re-training approach for deep neural networks that unites improved OoD detection capability and state-of-the-art semantic segmentation in one model. Up to now, only a small number prior works exists for anomaly segmentation on LostAndFound and Fishyscapes, respectively. We demonstrate that our OoD training significantly improves the detection efficiency via softmax entropy thresholding, leading to a performance superior over existing methods.

Moreover, we introduced meta classifiers for entropy based OoD object predictions. By applying lightweight logistic regressions, we have shown that entire LostAndFound OoD segments are meta classified reliably. This observation already holds for the tested neural network in its plain version. Due to the increased sensitivity of OoD predictions via entropy maximization, the meta classifiers' efficiency is even more pronounced. In view of emerging safety-critical deep learning applications, the combination of OoD training and meta classification has the potential to considerable improve the overall system's performance.

For future work, we plan to apply OoD training for the retrieval of OoD objects in order to assess the importance of their occurrence and whether a new concept is required to be learned. Our code is publicly available at https://github.com/robin-chan/meta-ood.

Acknowledgement

The research leading to these results is funded by the German Federal Ministry for Economic Affairs and Energy within the project "KI Absicherung – Safe AI for Automated Driving", grant no. 19A19013Q. The authors would like to thank the consortium for the successful cooperation and acknowledge fruitful discussions with Sharat Gujamagadi & Fabian Kunst.

References

[1] Samet Akçay, Amir Atapour-Abarghouei, and Toby P Breckon. Skip-ganomaly: Skip connected and adversarially trained encoder-decoder anomaly detection. In *2019 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE, 2019. 3

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

[3] Vijay Badrinarayanan, Alex Kendall, and Roberto Cipolla. Bayesian segnet: Model uncertainty in deep convolutional encoder-decoder architectures for scene understanding. In *Proceedings of the British Machine Vision Conference (BMVC)*, pages 57.1–57.12. BMVA Press, September 2017. 2

[4] Christoph Baur, Benedikt Wiestler, Shadi Albarqouni, and Nassir Navab. Deep autoencoding models for unsupervised anomaly segmentation in brain mr images. In *International MICCAI Brainlesion Workshop*, pages 161–169. Springer, 2018. 2, 3

[5] Hermann Blum, Paul-Edouard Sarlin, Juan Nieto, Roland Siegwart, and Cesar Cadena. Fishyscapes: A benchmark for safe semantic segmentation in autonomous driving. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) Workshops*, October 2019. 2, 3, 5, 7

[6] Dominik Brüggemann, Robin Chan, Matthias Rottmann, Hanno Gottschalk, and Stefan Bracke. Detecting Out of Distribution Objects in Semantic Segmentation of Street Scenes. In *The 30th European Safety and Reliability Conference (ESREL)*, 2020. 2

[7] Robin Chan, Matthias Rottmann, Fabian Hüger, Peter Schlicht, and Hanno Gottschalk. Application of Maximum Likelihood Decision Rules for Handling Class Imbalance in Semantic Segmentation. In *The 30th European Safety and Reliability Conference (ESREL)*, 2020. 6

[8] Robin Chan, Matthias Rottmann, Fabian Hüger, Peter Schlicht, and Hanno Gottschalk. Controlled false negative reduction of minority classes in semantic segmentation. In *2020 IEEE International Joint Conference on Neural Networks (IJCNN)*, 2020. 2, 4, 11

[9] Hyunsun Choi and Eric Jang. Generative ensembles for robust anomaly detection. *ArXiv*, abs/1810.01392, 2018. 2

[10] Pascal Colling, Lutz Roese-Koerner, Hanno Gottschalk, and Matthias Rottmann. Metabox+: A new region based active learning method for semantic segmentation using priority maps, 2020. 1

[11] Marius Cordts, Mohamed Omran, Sebastian Ramos, et al. The cityscapes dataset for semantic urban scene understanding. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 1, 2, 5

[12] Clement Creusot and Asim Munawar. Real-time small obstacle detection on highways using compressive rbm road reconstruction. In *2015 IEEE Intelligent Vehicles Symposium (IV)*, pages 162–167, 2015. 3

[13] Jia Deng, Alexander C. Berg, Kai Li, and Li Fei-Fei. What does classifying more than 10,000 image categories tell us? In Kostas Daniilidis, Petros Maragos, and Nikos Paragios, editors, *Computer Vision – ECCV 2010*, pages 71–84, Berlin, Heidelberg, 2010. Springer Berlin Heidelberg. 1

[14] Terrance DeVries and Graham W. Taylor. Learning Confidence for Out-of-Distribution Detection in Neural Networks, Feb 2018. 2

[15] Mark Everingham, S. M. Ali Eslami, Luc Van Gool, et al. The pascal visual object classes challenge: A retrospective. *International Journal of Computer Vision*, 111(1):98–136, Jan 2015. 4, 5, 6

[16] 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*, volume 48 of *Proceedings of Machine Learning Research*, pages 1050–1059, New York, New York, USA, 20–22 Jun 2016. PMLR. 2

[17] Matthias Hein, Maksym Andriushchenko, and Julian Bitterwolf. Why relu networks yield high-confidence predictions far away from the training data and how to mitigate the problem. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019. 2

[18] Dan Hendrycks, Steven Basart, Mantas Mazeika, Mohammadreza Mostajabi, Jacob Steinhardt, and Dawn Song. A benchmark for anomaly segmentation. *arXiv preprint arXiv:1911.11132*, 2019. 2

[19] 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, 8

[20] Dan Hendrycks, Mantas Mazeika, and Thomas Dietterich. Deep anomaly detection with outlier exposure. *Proceedings of the International Conference on Learning Representations*, 2019. 2

[21] S. Isobe and S. Arai. Deep convolutional encoder-decoder network with model uncertainty for semantic segmentation. In *2017 IEEE International Conference on INnovations in Intelligent SysTems and Applications (INISTA)*, pages 365–370, 2017. 2

[22] Joel Janai, Fatma Güney, Aseem Behl, Andreas Geiger, et al. Computer vision for autonomous vehicles: Problems, datasets and state of the art. *Foundations and Trends® in Computer Graphics and Vision*, 12(1–3):1–308, 2020. 1, 3, 5

[23] Nicolas Jourdan, Eike Rehder, and Uwe Franke. Identification of uncertainty in artificial neural networks. In *Proceedings of the 13th Uni-DAS e.V. Workshop Fahrerassistenz und automatisiertes Fahren*, July 2020. 2, 3

[24] Alex Kendall and Yarin Gal. What Uncertainties Do We Need in Bayesian Deep Learning for Computer Vision? In I Guyon, U V Luxburg, S Bengio, H Wallach, R Fergus, S Vishwanathan, and R Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 5574–5584. Curran Associates, Inc., 2017. 2

[25] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In Yoshua Bengio and Yann LeCun, editors, *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015. 5

[26] Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 6402–6413. Curran Associates, Inc., 2017. 2

[27] Kimin Lee, Honglak Lee, Kibok Lee, and Jinwoo Shin. Training confidence-calibrated classifiers for detecting out-of-distribution samples. In *International Conference on Learning Representations*, 2018. 2

[28] Shiyu Liang, Yixuan Li, and R. Srikant. Enhancing the reliability of out-of-distribution image detection in neural networks. In *International Conference on Learning Representations*, 2018. 2

[29] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft COCO: Common Objects in Context. In David Fleet, Tomas Pajdla, Bernt Schiele, and Tinne Tuytelaars, editors, *Computer Vision – ECCV 2014*, pages 740–755. Springer International Publishing, 2014. 1, 2, 5

[30] Krzysztof Lis, Krishna Nakka, Pascal Fua, and Mathieu Salzmann. Detecting the unexpected via image resynthesis. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2019. 2, 3, 7

[31] Kira Maag, Matthias Rottmann, and Hanno Gottschalk. Time-dynamic estimates of the reliability of deep semantic segmentation networks. In *IEEE International Conference on Tools with Artificial Intelligence (ICTAI)*, November 2020. 2, 4, 11

[32] David J. C. MacKay. A practical bayesian framework for backpropagation networks. *Neural Computation*, 4(3):448–472, 1992. 2

[33] A. Mehrtash, W. M. Wells, C. M. Tempany, P. Abolmaesumi, and T. Kapur. Confidence calibration and predictive uncertainty estimation for deep medical image segmentation. *IEEE Transactions on Medical Imaging*, pages 1–1, 2020. 2

[34] Alexander Meinke and Matthias Hein. Towards neural networks that provably know when they don't know. In *International Conference on Learning Representations*, 2020. 2

[35] Mehryar Mohri, Afshin Rostamizadeh, and Ameet Talwalkar. *Foundations of Machine Learning*. The MIT Press, 2012. 6

[36] Eric Nalisnick, Akihiro Matsukawa, Yee Whye Teh, Dilan Gorur, and Balaji Lakshminarayanan. Do deep generative models know what they don't know? In *International Conference on Learning Representations*, 2019. 2

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

[38] Philipp Oberdiek, Matthias Rottmann, and Gernot A. Fink. Detection and retrieval of out-of-distribution objects in semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2020. 2

[39] Peter Pinggera, Sebastian Ramos, Stefan Gehrig, Uwe Franke, Carsten Rother, and Rudolf Mester. Lost and found: detecting small road hazards for self-driving vehicles. In *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2016. 2, 5

[40] Jie Ren, Peter J. Liu, Emily Fertig, Jasper Snoek, Ryan Poplin, Mark Depristo, Joshua Dillon, and Balaji Lakshminarayanan. Likelihood ratios for out-of-distribution detection. In *Advances in Neural Information Processing Systems 32*, pages 14707–14718. Curran Associates, Inc., 2019. 2

[41] Matthias Rottmann, Pascal Colling, Thomas Paul Hack, Robin Chan, Fabian Hüger, Peter Schlicht, and Hanno Gottschalk. Prediction error meta classification in semantic segmentation: Detection via aggregated dispersion measures of softmax probabilities. In *2020 IEEE International Joint Conference on Neural Networks (IJCNN)*, 2020. 2, 4, 11

[42] Matthias Rottmann and Marius Schubert. Uncertainty measures and prediction quality rating for the semantic segmentation of nested multi resolution street scene images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2019. 2, 4, 11

[43] Jingdong Wang, Ke Sun, Tianheng Cheng, Borui Jiang, Chaorui Deng, Yang Zhao, Dong Liu, Yadong Mu, Mingkui Tan, Xinggang Wang, Wenyu Liu, and Bin Xiao. Deep high-resolution representation learning for visual recognition. *IEEE transactions on pattern analysis and machine intelligence*, PP, April 2020. 1

[44] Li Zhang, Xiangtai Li, Anurag Arnab, Kuiyuan Yang, Yunhai Tong, and Philip H. S. Torr. Dual graph convolutional network for semantic segmentation. In *Proceedings of the British Machine Vision Conference (BMVC)*, 2019. 7

[45] Xiang Zhang and Yann LeCun. Universum prescription: Regularization using unlabeled data. In *Thirty-First AAAI Conference on Artificial Intelligence*, 2017. 1, 2, 14

[46] Yi Zhu, Karan Sapra, Fitsum A. Reda, Kevin J. Shih, Shawn Newsam, Andrew Tao, and Bryan Catanzaro. Improving semantic segmentation via video propagation and label relaxation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019. 1, 5, 7

[47] Aleksandar Zlateski, Ronnachai Jaroensri, Prafull Sharma, and Frédo Durand. On the importance of label quality for semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018. 1

Appendix

A. Separability by means of Data Distribution

The violin plots in figure 3 visualize the separability of in-distribution and out-distribution pixels (binary classification) in LostAndFound and Fishyscapes, respectively. These plots summarize different statistics such as median and interquartile ranges and also show the full distribution of the data. The density corresponds to the relative pixel frequency at a given entropy value of the considered class. In the following, we refer to the shape of the violin plots as distribution.

First, we focus on evaluating LostAndFound OoD objects, see figure 3 (a). For the baseline model we observe that a large mass of data corresponding to the negative class is located at very low entropy values (median 0.02), i.e., most road pixels are classified with high confidence. Moreover, the 75th percentile is located at an entropy value of 0.04 and the sample of highest value at 0.57. Regarding the pixels of the positive class, we see that the distribution is rather dispersed. The median is at 0.29 and the interquartile ranges from 0.13 to 0.44. We conclude that, on average, positive samples have higher entropy values than negative samples, i.e., pixels of an OoD object are classified with higher uncertainty than for road pixels. However, for perfect performance one seeks a threshold such that both distributions (of positive and negative class) are separated. This is not the case for the baseline model since a substantial amount of samples still has very low entropy, e.g. the 10th percentile of the positive samples is at 0.04, which is also the median of negative samples.

After OoD training, the distribution of negative samples remains in large parts similar compared to the baseline only with little changes. Noteworthy, the median and upper quartile decrease down to entropy values of 0.1 and 0.2, respectively. The distribution's maximum is at 0.66. On the contrary, the changes of the distribution for the positive samples are significant as a large mass is concentrated at very high entropy values. The median is located at 0.59 which is roughly at the same magnitude as the maximum for negative samples. Moreover, the minimum value for positive pixels is at 0.01 which equals the median for negative samples. In particular the latter underlines the significant improvement of separability due to our OoD training.

We observe the same behavior for Fishysacpes OoD objects but even more pronounced, see figure 3 (b). After the OoD training, the medians of the two classes, 0.01 for negative samples and 0.87 for positive samples, differ by 86 percent points. Besides, the lower quartile of positive samples at an entropy value of 0.71 as well as the 1st percentile at 0.03 are still above the median of negative samples. Consequently, we conclude that our OoD training is beneficial for identifying OoD pixels.

B. Segment-wise Metrics for Meta Classifiers

As outlined in section 4, we train meta classifiers based on hand-crafted metrics. These metrics are derived from the softmax probabilities $f(x) \in (0,1)^{|\mathcal{H}|\times|\mathcal{W}|\times q}, x \in \mathcal{X}$ of deep convolutional neural networks, information we get in every forward pass. As a reminder, let $\hat{\mathcal{Z}}_{out}(x)$ be the set of pixel locations in image $x \in \mathcal{X}$ that are predicted to be OoD, see equation (7). A connected component $k \in \hat{\mathcal{K}}(x) \subseteq \mathcal{P}(\hat{\mathcal{Z}}_{out}(x))$ represents an *OoD segment / object prediction* due to the entropy being above the given threshold. This is different to other works dealing with segment-wise meta classification [8, 31, 41, 42] as they consider connected components sharing the same class label as segments.

We estimate uncertainty per OoD segment k by averaging pixel-wise scores at the segment's pixel locations $z \in k$. In addition to the plain softmax probabilities $f^z(x)$, we also incorporate three pixel-wise dispersion measures, namely $\forall z \in k$ the (normalized) entropy

$$\bar{E}(f^z(x)) = -\frac{1}{q}\sum_{j=1}^{q} f_j^z(x)\log(f_j^z(x)) \,, \qquad (11)$$

the variation ratio

$$V(f^z(x)) = 1 - f_{j^*(z)}^z(x) \,, \qquad (12)$$

and the probability margin

$$M(f^z(x)) = V(f^z(x)) + \max_{j\in\{1,\dots,q\}\setminus\{j^*(z)\}} f_j^z(x) \qquad (13)$$

with $j^*(z) := \arg\max_{j=1,\dots,q} f_j^z(x)$ being the class label according to the maximum a posteriori principle.

The segment's size $S(k) = |k|$ is not only needed for averaging but also serves as meta classification input on its own. Moreover, let $k_{in} \subset k$ be the set of pixel locations in the interior of the segment k, i.e., $k_{in} = \{(h,w) \in k : [h\pm1]\times[w\pm1] \in k\}$. This also gives us the pixel locations of the boundary $k_{bd} = k \setminus k_{in}$. In order to capture geometry features of a segment, we consider the relative sizes

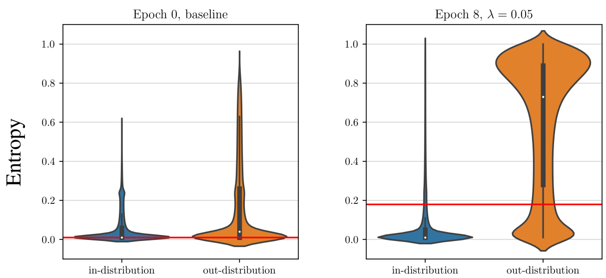
$$\tilde{S} = S/S_{bd} \text{ and } \tilde{S}_{in} = S_{in}/S_{bd} \qquad (14)$$

by treating the segment's boundary and interior separately.

Let $k_{nb} = \{z' \in [h\pm1]\times[w\pm1] \subset |\mathcal{H}|\times|\mathcal{W}| : (h,w) \in k, z' \notin k\}$ be the neighborhood of k. As metric if one segment is misplaced we include

$$N(j|k) = \frac{1}{|k_{nb}|}\sum_{z\in k_{nb}} \mathbb{1}_{\{j=j^*(z)\}} \; \forall\, j = 1,\dots,q \qquad (15)$$

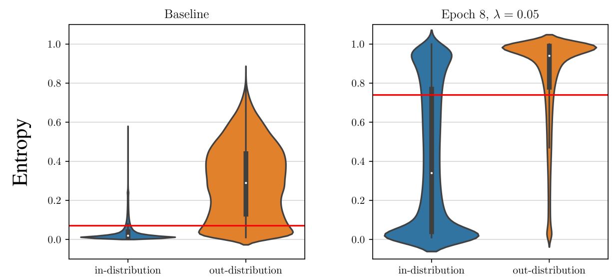
which is the proportion of neighborhood pixels, with class $j \in \{1,\dots,q\}$ having the highest softmax score, to neighborhood size. Another metric for localization purposes is



Baseline performance   Cityscapes void OoD training

(Detecting Cityscapes unlabeled objects)

Figure 8: Separability between in-distribution and out-distribution pixels in Cityscapes. Pixels labeled as train class according to the ground truth are considered as in-distribution, pixels labeled with the void class as out-of-distribution. For the results with Cityscapes void OoD training the baseline model (left) was retrained with entropy maximization on the Cityscapes void class (right), i.e., using Cityscapes unlabeled objects as OoD proxy for $\mathcal{D}_{out}$.



Baseline performance   Cityscapes void OoD training

(Detecting LostAndFound OoD objects)

Figure 9: Separability between in-distribution and out-of-distribution pixels in the OoD dataset LostAndFound. For the results with Cityscapes void OoD training the baseline model (left) was retrained with entropy maximization on the Cityscapes void class (right).

the segment's geometric center

$$C_h(k) = \frac{1}{S}\sum_{i=1}^{S} h_i \quad \text{and} \quad C_w(k) = \frac{1}{S}\sum_{i=1}^{S} w_i \qquad (16)$$

with $(h_i, w_i) \in k \ \forall \ i = 1, \dots, |k|$, i.e., averaging over the segment's pixel coordinates in vertical and horizontal direction.

For each segment k we then have 46 metrics in total (as $q = 19$ in our experiments). This forms a structured dataset

$$\mu \subseteq \mathbb{R}^{|\cup_{x \in \mathcal{X}} \hat{\mathcal{K}}(x)| \times 46} \qquad (17)$$

serving as input for the meta classification model $g : \mu \to [0, 1]$, the latter being a simple logistic regression in our case. By means of this linear model, we learn to discriminate whether a segment k has an intersection with the ground truth (while all inputs are independent of the ground truth segmentation), see also equation (9).

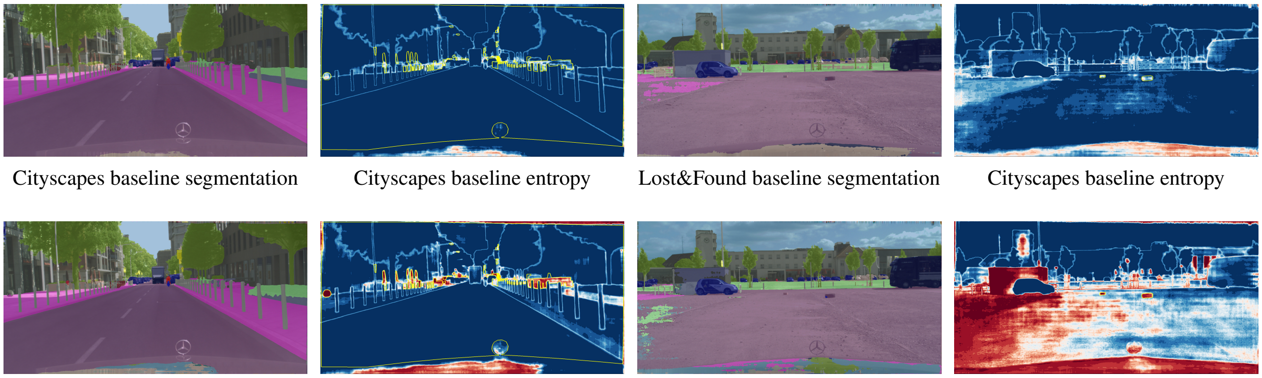
C. OoD Training with Cityscapes void Class

Before using the COCO dataset as OoD proxy, we conducted some experiments with the Cityscapes void class as OoD proxy for $\mathcal{D}_{out}$ in order to perform entropy maximization. This class includes objects that cannot be assigned to any of the Cityscapes training classes, therefore they remain unlabeled and are ignored during training. We refer to this retraining approach using the Cityscapes unlabeled objects as OoD proxy as *void OoD training*. We find the best results in our experiments for the DeepLabv3+ as baseline model after 8 epochs of void OoD training and out-distribution loss weight of $\lambda = 0.05$. With respect to the Cityscapes validation dataset, the retrained model clearly improves at identifying unseen unlabeled objects, see figure 8.

However, the same retrained model fails to generalize to unseen OoD objects available in the LostAndFound dataset, see figure 9. Not only the softmax entropy of OoD pixels is boosted but also the entropy of a significant amount of in-distribution pixels. This is even more considerable due to the strong class imbalance in LostAndFound. With respect to the AUROC, the void OoD training decreases the OoD detection score by 5 percent points down to 0.88, while decreasing the more relevant metric AUPRC by even 29 percent points down to 0.17 compared to the baseline model.

A visual comparison of the effects of void OoD training is shown is figure 10. The retraining does not noticeably impact the segmentation performance, neither for Cityscapes nor LostAndFound. In particular for the segmentation of the Cityscapes scenes, there are only minor differences visible, i.e., the difference in performance for the original task is marginal. This is in line with the observation that retraining with the multi-criteria loss function, see equation equation (2), and the COCO dataset as OoD proxy leads only to a marginal loss of mIoU for the Cityscapes validation dataset. With respect to the Cityscapes images, the softmax entropy inside unlabeled objects is clearly boosted due to void OoD training. This makes identifying such objects easier in comparison to the baseline model.

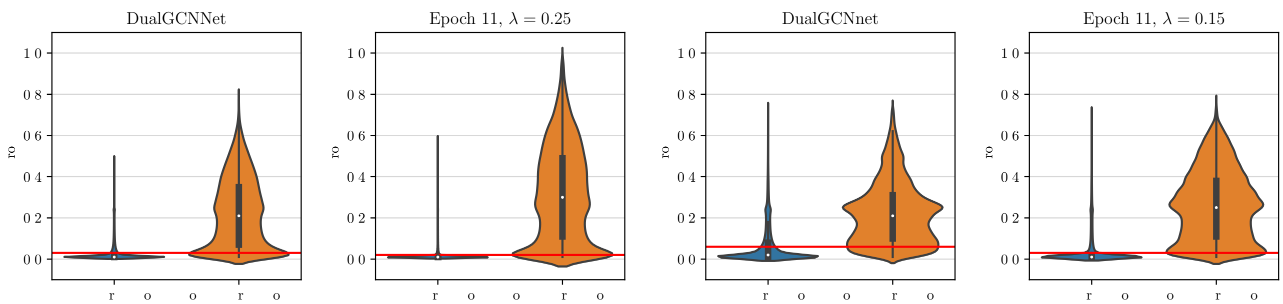
Regarding the LostAndFound segmentations the differences are more visible although still not being significant. On the contrary, by comparing the entropy heatmaps for the baseline model and the model after void OoD training, one observes that not only the entropy of pixels inside the



Cityscapes baseline segmentation | Cityscapes baseline entropy | Lost&Found baseline segmentation | Cityscapes baseline entropy

Void OoD training segmentation | Void OoD training entropy | Void OoD training segmentation | Void OoD training entropy

Figure 10: Comparison between baseline model and retrained model, with entropy maximization on Cityscapes unlabeled objects, for one Cityscapes and one LostAndFound scene. The first and third column displays the segmentations obtained by the respective models on either a Cityscapes or LostAndFound input image, the second and forth column displays the corresponding entropy heatmaps. In the entropy heatmaps, the OoD objects are marked with yellow lines.



(a) LostAndFound

(b) Fishyscapes

Figure 11: Relative pixel frequencies of LostAndFound (a) and Fishyscapes (b) OoD pixels, respectively, at different entropy values for the baseline model, i.e., before OoD training (*a & b left*), and after OoD training (*a & b right*). The red lines indicate the thresholds of highest accuracy.

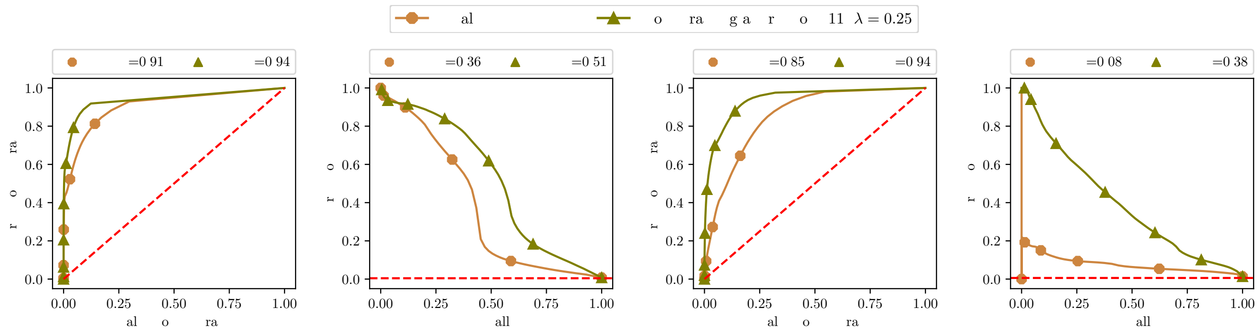
OoD objects is boosted but also many in-distribution pixels. This has an detrimental impact on the discrimination performance between in-distribution and out-distribution pixel as these two classes cannot be separated well via entropy thresholding. These visualizations support the impression of the pixel-wise evaluation that void OoD training is not suitable for the detection of objects other than the Cityscapes unlabeled objects.

D. OoD Training for DualGCNNet

As a second model complementary to the DeepLabv3+ model, we performed the same experiments of OoD training, i.e., retraining with the COCO dataset as OoD proxy, for the DualGCNNet which is a weaker and more lightweight network compared to the state-of-the-art DeepLabv3+ segmentation network. We find the best results after 11 epochs of OoD training with out-distribution loss weight of $\lambda = 0.25$. As optimizer we used Adam with a learning rate of 10^{-6}.

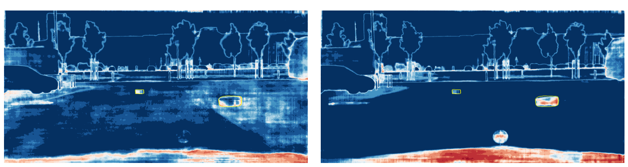
The pixel-wise evaluation results are presented by means of the violin plots in figure 11 and by ROC as well as PR curves in figure 12. We evaluated the OoD detection for the LostAndFound test and Fishyscapes static dataset in the same manner as for the experiments for DeepLabv3+.

We observe that OoD training is not as effective as for the DeepLabv3+ model in terms of absolute performance gain. However, we still observe a decent improvement in separability. By applying OoD training, the AUROC increases by 3 percent points for LostAndFound and even 9 percent points for Fishyscapes up to a score of 0.94 for both datasets. With respect to the PR curves, the AUC improves by 15 percent points up to 0.51 for LostAndFound and by 20 percent points up to 0.38 for Fishyscapes. Noteworthy,



(a) LostAndFound (*left:* AUROC, *right:* AUPRC)   (b) Fishyscapes (*left:* AUROC, *right:* AUPRC)

Figure 12: Detection ability of LostAndFound (a) and Fishyscapes (b) OoD pixels, respectively, evaluated by means of receiver operating characteristic curve (*a & b left*) and precision recall curve (*a & b right*). The red lines indicate the performance according to random guessing.



Before OoD Training   After OoD training

Figure 13: Comparison of softmax entropy heatmaps before (left) and after OoD training (right). The yellow lines mark the OoD objects according to their ground truth annotation.

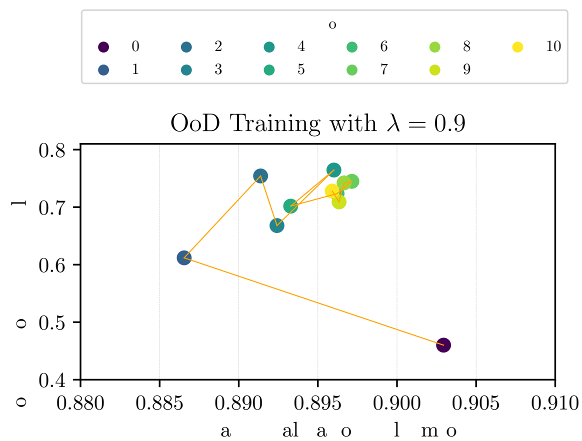


Figure 14: Mean intersection over union (mIoU) for the Cityscapes validation dataset split over the course of OoD training.

these AUC scores after OoD training are higher than for the plain DeepLabv3+ (baseline) model which is already a strong OoD detection model.

These results for the weaker DualGCNNet model further demonstrate the positive effect on the OoD detection ability when performing OoD training with the COCO dataset as OoD proxy.

E. OoD Training Visualization

The improved separation ability due to OoD training is not only achieved by increasing the softmax entropy of OoD pixels but also by decreasing the softmax entropy for in-distribution pixels. This can be also observed by means of the in-distribution violins, for instance in figure 11. By comparing the shapes of the violins corresponding to the DualGCNNet plain model and the model after OoD training, we notice that the violin shapes remain similar in large parts. The median and the upper quartile, however, decrease down to lower entropy values after OoD training. This indicates that after entropy maximization the model is on the one hand more uncertain at OoD pixel locations and on the other hand more certain about its prediction at in-distribution pixel locations. The same observation also holds for the DeepLabv3+ model, see figure 3. This is in line with the observation made in [45] that training with an OoD proxy may have a regularizing effect.

An illustration is provided in figure 13. For comparison purposes, we refer to the entropy heatmaps provided in figure 10 as both figures show the same scene. The visualization of heatmaps clearly shows that due to OoD training pixels with high entropy are more concentrated inside OoD objects. Moreover, the in-distribution objects, especially the pixels corresponding to the road, have lower entropy values than before OoD training. This makes the road seem cleaner with respect to the possible occurrence of OoD objects. After entropy maximization the OoD objects are (visibly) better recognizable within the softmax entropy heatmaps. Therefore, we expect that the meta classification performance is leveraged as the meta classifiers are able to

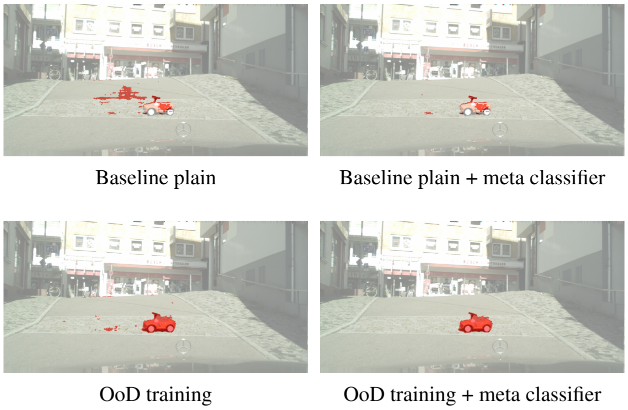


Figure 15: OoD detection for one scene with different combinations of entropy thresholding for the plain model, entropy thresholding after OoD training and meta classification. For all the OoD predictions the same threshold score of $t = 0.5$ was used. The red segments indicate OoD object predictions.

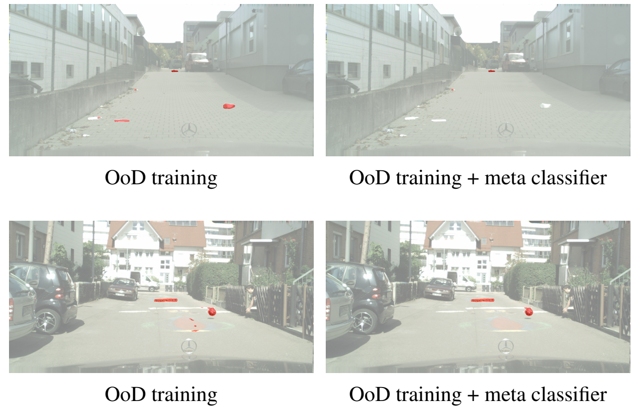


Figure 16: OoD detection performed by the OoD-trained network with and without meta classifiers. The red segments indicate OoD object predictions.

estimate the shape of OoD objects even better. Moreover, higher entropy values are stronger correlated with the presence of OoD objects.

F. Course of OoD Training

In order to monitor that the baseline model does not unlearn its original task due to OoD training, we evaluate the model's original task performance over the training epochs. We evaluate the mIoU on the Cityscapes validation dataset against the AUPRC on the LostAndFound test dataset, displayed in figure 14. The state-of-the-art DeepLabv3+ model, which serves as baseline throughout our experiments, achieves an mIoU of 90.30% when equipped only with the standard maximum a posteriori (MAP) decision principle while the same model has an entropy based OoD detection performance of 46.01% in AUPRC. By fine tuning the neural network with entropy maximization on OoD inputs, we on the one hand sacrifice only little in mIoU (of the original task). On the other hand, we observe improved AUPRC scores over the course of training epochs peaking at 76.45%. This considerable gain at detecting OoD samples in LostAndFound comes with a marginal loss in Cityscapes validation mIoU of less than 1 percent point. Moreover, the course of the OoD training illustrates convergence around the best AUPRC score with an mIoU loss that is in the same range as for the best score after OoD training. Concerning the overall performance of perception systems that rely on semantic segmentation, e.g., in applications like automated driving, this is a favorable trade-off in terms of safety that comes with very little computational overhead.

G. Meta Classification Visualization

The logistic regressions as meta classifiers have proven their efficiency in identifying and afterwards removing false positive (FP) / incorrect OoD object predictions. In this section we intend to show further examples for the FP OoD removal and thus show the final output of our two-step procedure for OoD detection.

For the plain model the meta classifiers are already able to remove FP OoD predictions reliably, see figure 15 top row. However, some false positive OoD predictions still remain. As pixels with high entropy are more concentrated inside OoD objects after the entropy maximization of the OoD training, the combination of OoD training and meta classification yields the best result in terms of the number of FP OoD predictions, see figure 15 bottom row. The examples in figure 16 further illustrate that the improved OoD detection performance after OoD training can even be enhanced by employing meta classifiers. The removed FP OoD predictions are rather small. However, we already consider one single pixel as FP OoD object prediction if that pixel is incorrectly predicted to be OoD. One could also define an OoD prediction to have a minimum amount of pixels. As our main focus is the reduction of overlooked OoD objects, we stick to the definition of equation (8) and consider an OoD object to be found if at least one pixel of that object is correctly classified as OoD. Therefore, small OoD segments are also fed through the meta classification model.

To conclude, our two step method, consisting of entropy maximization and meta classification, extends segmentation networks by an improved OoD detection capability and unites both tasks in one model.