Identification of Uncertainty in Artificial Neural Networks

Nicolas Jourdan* Eike Rehder* Uwe Franke*

Abstract: Neural networks are the backbone of environment perception systems for autonomous driving. While they achieve state-of-the-art performance in most computer vision tasks, they typically do not provide self-evaluation with respect to their predictions. For autonomous vehicles, though, it is vital that the system actively reasons about its limitations. The aim of this work is to identify uncertainty in neural network decisions for semantic segmentation. To systematically evaluate this, we develop a methodology to compare neural networks' performance in out-of-distribution detection and uncertainty estimation. As the core contribution of our work, we propose a novel approach to learn uncertainty estimation for out-of-distribution detection from unlabeled parts of the training data. Our approach only extends the training strategy and therefore does not require any changes to network architecture or runtime. We show that resulting networks perform en par with state-of-the-art methods that require much greater computational efforts. Consequently, any given architecture for segmentation can be trained to also provide out-of-distribution detection.

Keywords: neural networks, out-of-distribution detection, semantic segmentation, uncertainty

1 Introduction

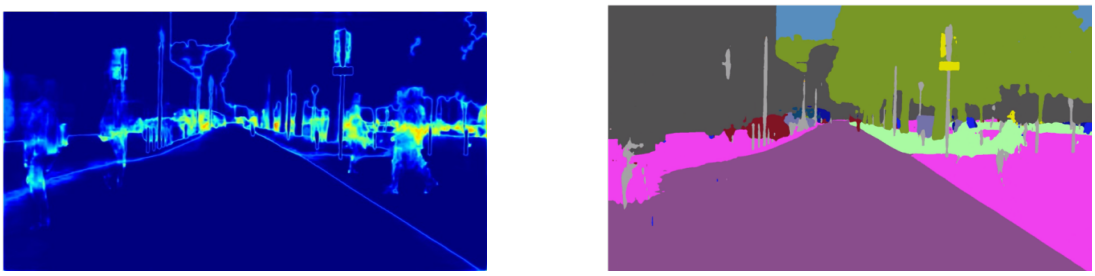
Neural networks have achieved state-of-the-art in most computer vision tasks and are the foundation of modern environment perception systems. While they have been deployed with great success, neural networks typically do not provide self-evaluation with respect to their predictions. If neural networks are used in environment perception systems of autonomous vehicles, though, it is mandatory that the system actively self-identifies its limitations as the human passenger does not provide a fallback option [13].

This work focuses on neural networks for road scene understanding from camera images. Thus, the aforementioned limitations include scene configurations the network was not trained to comprehend. In such situations the desired behavior of the classification system would be to express low confidence or high uncertainty for the detection. As neural networks can only be trained on a finite dataset, not every possible situation or even class configuration could be contained in a finite-sized and finite-class dataset. Thus, in deployment, a neural network will encounter situations that significantly differ from the data distribution it was trained with. For autonomous vehicles, it is crucially important

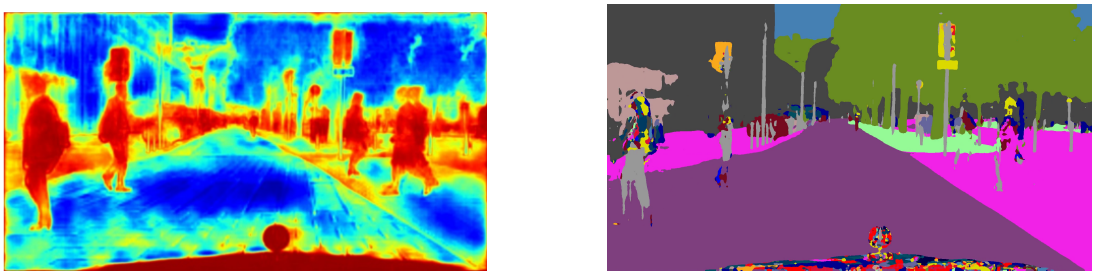
*Daimler AG, Image Understanding Group (e-mail: {nicolas.jourdan, eike.rehder, uwe.franke}@daimler.com).

to detect these out-of-distribution (OOD) samples since failures may have catastrophic consequences.
As an example, we provide semantic segmentation results of three different models, all trained on the Cityscapes dataset [1], in Figure 1. A standard network segments the image into relevant classes such as *road*, *infrastructure*, *vegetation* or *pedestrian* (Fig. 1 (a)). Now, suppose the same network was trained on data that did not include any *pedestrian* labels. Alarmingly, this network fills in the unknown pedestrians with highly confident predictions for the respective background, shown in Fig. 1 (b). An autonomous vehicle relying on this perception consequently would cause severe accidents. If we apply our proposed training method, the network achieves decent prediction in known regions while tagging the pedestrians as highly inconfident (*cf.* Fig. 1 (c)).

(a) Original image and prediction of the baseline network trained with all 19 classes.



(b) Uncertainty and prediction of the baseline network trained without *pedestrian* and *rider*.



(c) Uncertainty and prediction using our approach.

Figure 1: Example showcasing the behavior of the semantic segmentation network in the presence of unknown classes. Blue indicates low uncertainty while red indicates high uncertainty.

In this work, we present a method to extend any given segmentation architecture with uncertainty estimation. To put this into perspective, we initially review approaches from literature. We then present our simple yet effective training strategy. In order to evaluate and compare respective results, a methodology for out-of-distribution detection and network self-assessment is presented and used to benchmark the uncertainty estimation methods.

2 Uncertainty Estimation in Neural Networks

Any neural network that is used in the perception stage of an autonomous system should provide some means of self-evaluation. Unfortunately, the categorical decisions of neural networks do not provide this directly. Yet, the typical activation function for classification is the softmax $s_i(\boldsymbol{a}) = \exp(a_i) / \sum_{j=1}^{C} \exp(a_j)$, where a_i is the network's output logit for the i-th of C classes. The softmaxes can be interpreted as a categorical probability distribution. In the following, we will briefly explain how these outputs have been used to estimate uncertainty of network decisions in literature.

2.1 Reference Methods

Naïve Baseline The maximum softmax probability can be used as a confidence measure for out-of-distribution samples as well as misclassifications [6, 11]. This provides the baseline for the experiments of this work (*cf* Fig. 1 (b)). In order to incorporate the prediction of all classes, the per-pixel entropy can be used instead. The entropy for this case with an input sample $\boldsymbol{x}$ is defined as $H(\boldsymbol{x}) = -\sum_{i=1}^{C} p_i(\boldsymbol{x}) \log p_i(\boldsymbol{x})$. While the confidence score is easily acquired, it exhibits significant limitations. Since the softmax function normalizes the output distribution to $\sum_{i=1}^{C} p_i(\boldsymbol{x}) = 1$, the confidences by definition cannot be calibrated in open set conditions with unknown classes that are not accounted for during training. Consequently, the softmax probabilities only express relative confidences for the known classes (e.g. the image more likely shows a dog than a cat) but no overall confidence in the classification (e.g. the image shows neither a dog nor a cat).

Temperature Scaling Temperature scaling is an extension to the softmax function that is commonly used as the final layer in classification networks. For this, the inputs $\boldsymbol{a}$ to the softmax are divided by a scalar constant *temperature* $T \in \mathbb{R}^+$,

$$s_i(\boldsymbol{a}, T) = \frac{\exp\left(a_i / T\right)}{\sum_{j=1}^{C} \exp\left(a_j / T\right)}. \tag{1}$$

Since it only modifies the logits linearly, it can be applied to existing models without the need for architecture changes or retraining [11].

Monte-Carlo Dropout In Monte-Carlo dropout, multiple forward passes are performed for a test image. In each of the forward passes, random units are dropped by using dropout [8] layers in the network architecture, introducing variations in the predictions. Thus, they can be considered stochastic samples approximating a *Bayesian Neural Network* [4, 9]. The resulting N predictions are combined by class-wise averaging the predicted probabilities $P_{final} = \frac{1}{N} \sum_{i=1}^{N} P_i$. The combined per-pixel

probability distribution P_{final} is consequently used for uncertainty estimation. To enable the use of Monte-Carlo dropout for uncertainty estimation with the model used in this work, dropout layers are added to the central stages of the architecture, roughly following the setup of *Bayesian Segnet* but applied to an FCN [9]. We compute $N = 10$ samples per image during testing. The major drawback of using Monte-Carlo dropout is the significantly increased computational cost since multiple forward passes are performed for each input image.

Network Ensembles Deep ensembles have been introduced for uncertainty estimation in image classification by Lakshminarayanan et al. [10]. In our implementation we employ an ensemble of $N = 9$ network models for semantic segmentation. All ensemble members share the same architecture and training data and only differ in the random seed used for training, following the configuration for image classification in [10]. The random seed influences the initialization of the network weights as well as the shuffling of the training images. The predictions of the ensemble members are combined to P_{final} by class-wise averaging, equal to the combination of samples obtained by Monte-Carlo dropout. Consequently, the major drawback of this method is the significantly increased computational cost due to multiple forward passes being performed per image.

2.2 Margin-Entropy Loss

While sampling based uncertainty estimation methods like ensembles and Monte-Carlo dropout have shown decent results in recent publications, the need to process multiple network forward passes per image usually prohibits the deployment in autonomous vehicles due to the significantly increased runtime. In practice, an uncertainty estimation method is required that can be used to extend an existing network architecture without increasing the computational cost.

We employ the margin-entropy loss function that enforces a margin in predictive entropy between predictions on known classes and those on the unknown data. The margin-entropy loss function is defined as $L_{\text{me}} = \max\left(m + \bar{H}_{\text{id}} - \bar{H}_{\text{void}}, 0\right)$, where $\bar{H}_{\text{id}}$ and $\bar{H}_{\text{void}}$ are the average entropy of in-distribution and OOD samples, respectively. The hyperparameter m controls the margin between the two. In contrast to entropy maximization, the margin-entropy loss term L_{me} prevents overfitting on the OOD areas and limits negative effects on the performance on the original classification task. The margin-entropy loss is designed to punish high confidence values on out-of-distribution samples for image classification. The existing implementations of margin-entropy type loss functions require either a computationally expensive ensemble [15], or a separate dataset providing OOD samples for training [7]. In this work, we require neither of the two. Instead, we use Cityscapes labels as known classes while pixels that are unlabeled or ignored in Cityscapes images, $\boldsymbol{X}_{\text{void}}$, can be treated as OOD (shown in red in Fig. 2). The complete loss function used to train the network is the sum $L = L_{\text{ce}} + \beta \cdot L_{\text{me}}$, where L_{ce} is defined as the standard cross-entropy loss limited to the ID samples $\boldsymbol{X}_{\text{id}}$ and β is a hyperparameter used to control the influence of the margin-entropy loss.

Note that the additional margin-entropy loss does not induce any changes to network architecture or required data. It can readily be applied to any given training pipeline and, thus, does not impact the network runtime in deployment.

Figure 2: Example for unused image areas in the Cityscapes dataset. Unused areas are highlighted in red.[1]

3 Experiments

In order to show the potential of our approach, we evaluate it experimentally in comparison to the presented baseline methods. For the first experiment, we construct a meaningful benchmark for Out-of-Distribution Detection. Secondly, we compare the networks' performance in Self-Assessment. All experiments in this work use the same FCN-based [12] architecture for semantic segmentation with a GoogLeNet [14] feature extractor trained with equal hyperparameter settings.

3.1 Out-of-Distribution Detection

Evaluation of out-of-distribution detection poses a severe challenge since well-defined unknowns are needed. In many works, it is customary to use out-of-distribution test data from a source other than the one the network was trained with. In segmentation, however, images are required that resemble the training data and comprise both, known and unknown classes. Unfortunately, Cityscapes void labels are not guaranteed to not contain known classes. This raises the need of *known unknowns* for evaluation. As a solution, we leave out well-defined classes in the dataset during training which will then be used as *known unknown* OOD samples during testing.

For evaluation, we employ the Receiver Operating Characteristic (ROC), which is commonly used to evaluate out-of-distribution detectors in recent literature [11]. The ROC describes the relative tradeoff between *true positive rate* $\mathrm{TPR} = \mathrm{TP}/(\mathrm{TP} + \mathrm{FN})$ and *false positive rate* $\mathrm{FPR} = \mathrm{FP}/(\mathrm{FP} + \mathrm{TN})$ of binary classifiers [2]. In the out-of-distribution experiment, in-distribution samples are labeled as positive, while out-of-distribution samples are labeled as negative. We use two metrics that summarize the ROC performance of a classifier:

- **AUROC:** The ROC curve of a continuous score classifier can be summarized in a single scalar by calculating the Area under the Receiver Operating Characteristic (AUROC). A perfect detector corresponds to an AUROC score of 1.0 [3].
- **FPR@0.95TPR:** The false positive rate at 0.95 true positive rate describes the probability that a negative sample is classified as positive when the TPR equals 0.95 [11].

[1] Note that this is an extreme example for unused regions.

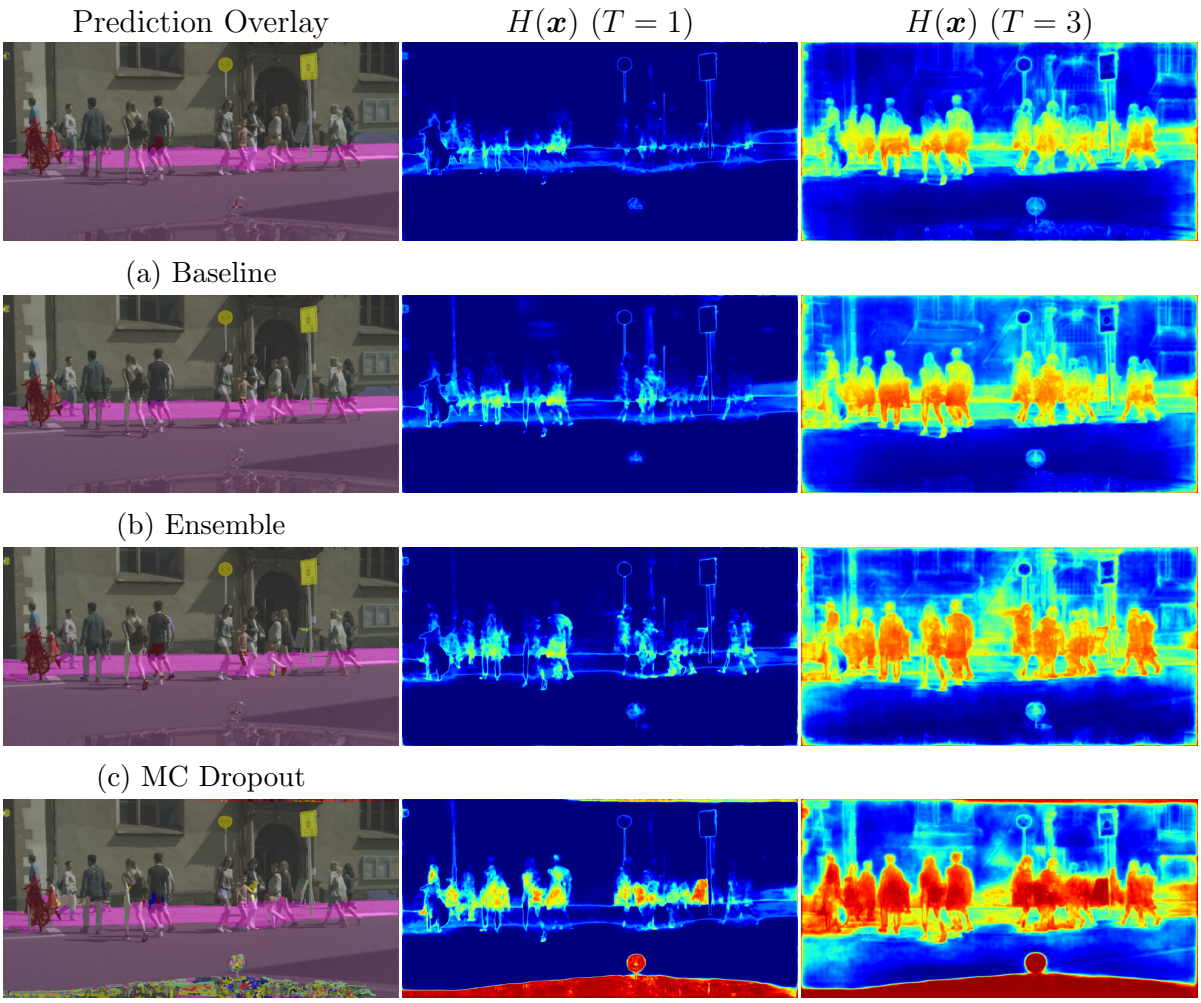


Figure 3: Uncertainty visualizations for OOD detection using the classes *pedestrian* and *rider* as out-of-distribution samples for evaluation. Blue indicates low uncertainty while red indicates high uncertainty.

The OOD detection performance is visualized for an example scene in Figure 3. In this experiment, the network is trained on the Cityscapes [1] training set, treating *pedestrian* and *rider* as *known unkowns* for evaluation. These classes are therefore not used in any of the loss calculations. Figure 3 shows the image with the model prediction as overlay (left), the predictive entropy as a heatmap (middle) and the same with temperature scaling applied (right).

The confidences of the baseline network on the unknown class *pedestrian* are nearly indistinguishable from the known classes, comparable to Figure 1. The OOD detection is slightly enhanced using ensembles and Monte-Carlo dropout. Using the proposed margin-entropy loss, the pedestrians show significantly lower confidence than the known classes, increasing OOD detection performance. The margin-entropy loss additionally causes low confidence levels on image areas such as the hood of the ego vehicle and the rectification border since these areas are ignored in Cityscapes. This is a desirable result since they cannot be classified correctly with the available set of classes. Temperature scaling

notably decreases confidence on pixels showing the OOD class *pedestrian* in all of the presented methods including the baseline model. Combined with the introduced methods for uncertainty estimation, the pedestrians show low confidence, clearly separating them from the known classes in the image. Using the margin-entropy loss, this effect is most pronounced.

Table 1 shows the quantitative results of this OOD detection experiment evaluated on the Cityscapes [1] validation set, again with persons treated as *known unknowns* for evaluation. The columns beneath each OOD detection metric show the results for using either the maximum predicted confidence or the negative predictive entropy as confidence scores. Without temperature scaling, the margin-entropy loss achieves the best OOD detection results in terms of FPR@0.95TPR. Combined with temperature scaling, the detection rates are significantly improved for all methods. While the sampling based methods combined with temperature scaling achieve higher detection rates than margin-entropy in this setting, the results of margin-entropy are comparable with only requiring a single forward pass. Using the predictive entropy as uncertainty score yields noticeably better results in this experiment when compared to the predicted confidence.

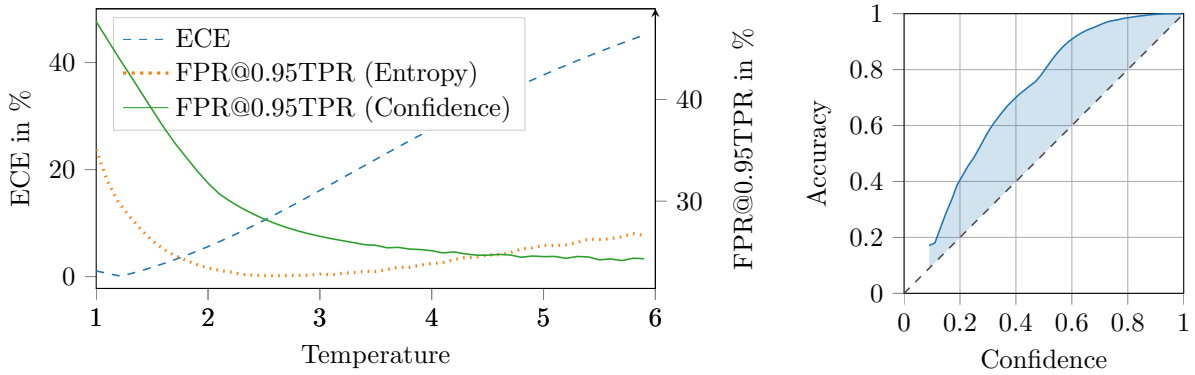
Configuration	FPR@0.95TPR in % (↓)		AUROC in % (↑)		$mIoU$ in % (↑)
	Conf.	Entr.	Conf.	Entr.	
Baseline	57.5	49.7	89.7	90.8	74.0
Ensemble	53.3	39.7	91.3	92.6	76.4
MC Dropout	51.1	37.1	89.9	93.5	72.9
Margin-Entropy	46.6	34.0	92.0	93.2	71.7
Baseline + T	37.1	30.7	93.1	94.2	74.0
Ensemble + T	32.4	24.8	94.2	95.3	76.4
MC Dropout + T	26.7	18.7	95.3	96.4	72.9
Margin-Entropy + T	28.3	26.0	94.5	94.9	71.7

Table 1: Results of OOD detection experiment on the Cityscapes [1] validation set using *pedestrian* and *rider* as out-of-distribution classes. The subsequent +T denotes the use of temperature scaling. ↑ indicates larger value is better and ↓ indicates lower value is better.

The effects of temperature scaling are further analyzed in Figure 4. To measure the influence of temperature scaling on the calibration of the network confidences we additionally use the Expected Calibration Error (ECE) [5]. The ECE for n samples is defined as

$$\mathrm{ECE} = \sum_{m=1}^{M} \frac{|B_m|}{n} |\mathrm{acc}(B_m) - \mathrm{conf}(B_m)|. \tag{2}$$

The ECE expresses the weighted average difference between predicted confidence and achieved accuracy, discretized over M confidence intervals with their respective set of predictions B_m. Figure 4 (a) highlights the effects of temperature scaling on the calibration of the network predictions. When tuned for optimum OOD detection performance, the calibration of the network deteriorates significantly. At the optimum OOD detection temperature of $T = 2.6$, the network is strongly under-confident as visualized in the reliability curve in Figure 4 (b). Thus, the usage of temperature scaling depends on the calibration requirements of the specific usecase.



(a) ECE and FPR@0.95TPR visualized as a function of the temperature T.

(b) Reliability curve on in-distribution classes at temperature $T = 2.6$.

Figure 4: Temperature scaling effects on OOD detection and calibration using *pedestrian* and *rider* as out-of-distribution classes. Charts show the baseline model.

3.2 Network Self-Assessment

Out-of-distribution input data is not the only source of failure for classification systems. Similar to humans, classification systems based on neural networks sometimes fail to predict the correct class even for input samples that are close to the distribution of the training data. We evaluate the quality of the confidence score with regard to network self-assessment in two ways. First, the confidence score is used in a binary classification setting. In contrast to the OOD experiment, positive samples are now defined as correctly classified samples, while negative samples are defined as incorrectly classified. Second, we evaluate the calibration of the confidence scores. A network is perfectly calibrated if the accuracy of the predictions is equal to their respective confidences [5]. This property is crucial if the classification system is used for sensor fusion in conjunction with other perception systems. The confidence needs to be a meaningful, interpretable measure to be able to understand and compare predictions.

The results of the self-assessment experiments are summarized in Table 2. In addition to the binary classification metrics and the expected calibration error, we report the average predicted confidence for misclassifications $\overline{p}_{\text{err}}$. Contrary to the out-of-distribution detection experiments, temperature scaling showed no positive influence on network self-assessment and is not included in Table 2. The unchanged predicted softmax confidence of the baseline model is a competitive baseline for network self-assessment with an AUROC of 93.8 %. This score expresses a performance which significantly exceeds that of a random classifier. In general, capturing statistics about predicted confidences of correct and incorrect classifications is surprisingly effective for detecting whether an example is classified correctly, even though the prediction probability itself can be deceiving with $\overline{p}_{\text{err}} = 0.747$. This supports the results for image classification by Hendrycks & Gimpel [6].

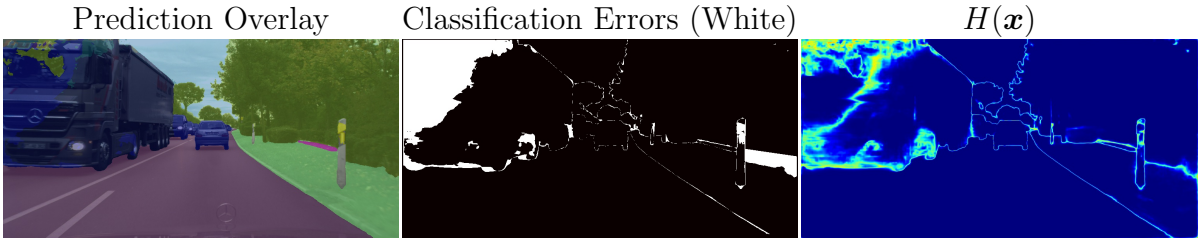
The network ensemble outperforms the other approaches in terms of AUROC and FPR@0.95TPR. There is a correlation visible between these metrics and the ECE which can intuitively be explained by the similar objectives measured by both metrics. In contrast to the OOD detection experiments, there is no clear benefit observable for choosing either the predicted confidence or the predictive entropy as uncertainty quantification.

The margin-entropy loss introduces no deterioration in this task compared the baseline, indicating that the uncertainty related to out-of-distribution samples is fundamentally different when compared to misclassifications.

	FPR@0.95TPR (↓) in %		AUROC (↑) in %		$\overline{p}_{\text{err}}$	ECE (↓) in %	$mIoU$ (↑) in %
Configuration	Conf.	Entr.	Conf.	Entr.			
Baseline	32.8	32.9	93.8	94.1	0.747	1.47	74.3
Ensemble	28.7	29.2	95.2	95.1	0.692	0.44	77.4
MC Dropout	32.8	33.5	94.7	94.7	0.697	0.78	72.4
Margin-Entropy	33.9	34.7	94.2	94.3	0.716	1.19	72.3

Table 2: Results of network self-assessment experiments on the Cityscapes [1] validation set. ↑ indicates larger value is better and ↓ indicates lower value is better.

The performance of the network ensemble is visualized in comparison to the baseline model in Figure 5. A large area of the front of the truck is misclassified as *traffic sign* and *passenger car*. The ensemble (b) exhibits significantly lower confidence for the misclassified areas compared to the baseline model (a) while at the same time reducing the misclassified area.



(a) Baseline

(b) Ensemble

Figure 5: Example scene for misclassification detection

4 Conclusion

In this work, the uncertainty estimation of neural networks for semantic segmentation was analyzed. We provide an evaluation methodology for out-of-distribution detection and network self-assessment and compared multiple approaches experimentally. The results of the baseline out-of-distribution detection experiments in this work highlight the need for uncertainty estimation in black-box perception systems like neural networks. Recent research on uncertainty estimation in neural networks has shown the effectiveness of

sampling based methods like network ensembles [10] and Monte-Carlo dropout [9]. This, however, increases computational demands, potentially beyond real-time requirements. We, instead, propose a simple yet effective training strategy that can enable any given architecture to detect out-of-distribution data at similar detection accuracy. This enables existing systems to be extended by uncertainty estimation of which the downstream processing can benefit greatly.

Acknowledgements

We would like to thank Maren Henzel and Prof. Dr. rer. nat. Hermann Winner from the Institute of Automotive Engineering (FZD) at the Technical University of Darmstadt for their support and supervision of this work.

References

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

[2] James P Egan. *Signal detection theory and ROC-analysis*. English. Includes bibliographies and index. New York : Academic Press, 1975.

[3] Tom Fawcett. "An introduction to ROC analysis". In: *Pattern recognition letters* 27.8 (2006).

[4] Yarin Gal and Zoubin Ghahramani. "Dropout as a Bayesian Approximation: Representing Model Uncertainty in Deep Learning". In: *Proc. of the International Conference on Machine Learning (ICML)*. 2016.

[5] Chuan Guo et al. "On Calibration of Modern Neural Networks". In: *Proc. of the International Conference on Machine Learning, (ICML)*. 2017.

[6] Dan Hendrycks and Kevin Gimpel. "A Baseline for Detecting Misclassified and Out-of-Distribution Examples in Neural Networks". In: *Proc. of the International Conference on Learning Representations (ICLR)*. 2017.

[7] Dan Hendrycks, Mantas Mazeika, and Thomas G. Dietterich. "Deep Anomaly Detection with Outlier Exposure". In: *Proc. of the International Conference on Learning Representations (ICLR)*. 2019.

[8] Geoffrey E Hinton et al. "Improving neural networks by preventing co-adaptation of feature detectors". In: *arXiv preprint arXiv:1207.0580* (2012).

[9] Alex Kendall, Vijay Badrinarayanan, and Roberto Cipolla. "Bayesian SegNet: Model Uncertainty in Deep Convolutional Encoder-Decoder Architectures for Scene Understanding". In: *Proc. of the British Machine Vision Conference (BMVC)*. 2017.

[10] Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. "Simple and Scalable Predictive Uncertainty Estimation using Deep Ensembles". In: *Proc. of the Annual Conference on Neural Information Processing Systems (NIPS)*. 2017.

[11] Shiyu Liang, Yixuan Li, and R. Srikant. "Enhancing The Reliability of Out-of-distribution Image Detection in Neural Networks". In: *Proc. of the International Conference on Learning Representations (ICLR)*. 2018.

[12] Jonathan Long, Evan Shelhamer, and Trevor Darrell. "Fully convolutional networks for semantic segmentation". In: *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2015.

[13] Markus Maurer et al. *Autonomes Fahren: technische, rechtliche und gesellschaftliche Aspekte*. Springer, 2015.

[14] Christian Szegedy et al. "Going deeper with convolutions". In: *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2015.

[15] Apoorv Vyas et al. "Out-of-Distribution Detection Using an Ensemble of Self Supervised Leave-Out Classifiers". In: *Proc. of the European Conference on Computer Vision (ECCV)*. 2018.