

Uncertainty and Prediction Quality Estimation for Semantic Segmentation via Graph Neural Networks

Edgar Heinert*¹
heinert@uni-wuppertal.de

Stephan Tilgner*²
tilgner@uni-wuppertal.de

Timo Palm¹
palm@math.uni-wuppertal.de

Matthias Rottmann¹
rottmann@uni-wuppertal.de

¹ School of Mathematics and Natural Sciences
University of Wuppertal
Wuppertal, Germany

² School of Electrical, Information and Media Engineering
University of Wuppertal
Wuppertal, Germany

Abstract

When employing deep neural networks (DNNs) for semantic segmentation in safety-critical applications like automotive perception or medical imaging, it is important to estimate their performance at runtime, e.g. via uncertainty estimates or prediction quality estimates. Previous works mostly performed uncertainty estimation on pixel-level. In a line of research, a connected-component-wise (segment-wise) perspective was taken, approaching uncertainty estimation on an object-level by performing so-called meta classification and regression to estimate uncertainty and prediction quality, respectively. In those works, each predicted segment is considered individually to estimate its uncertainty or prediction quality. However, the neighboring segments may provide additional hints on whether a given predicted segment is of high quality, which we study in the present work. On the basis of uncertainty indicating metrics on segment-level, we use graph neural networks (GNNs) to model the relationship of a given segment's quality as a function of the given segment's metrics as well as those of its neighboring segments. We compare different GNN architectures and achieve a notable performance improvement.

1 Introduction

In recent years, DNNs for visual perception tasks have increasingly been used in safety-critical applications such as advanced driver assistance systems [30, 37], autonomous driving [27, 36, 52], or medical diagnostics [20, 54, 47, 55, 56].

© 2024. The copyright of this document resides with its authors.

It may be distributed unchanged freely in print or electronic forms.

* Equal contribution. This work has been supported by the German Federal Ministry of Education and Research within the junior research group project “UnrEAL”, grant no. 01IS22069, and by the VolkswagenStiftung, “DigiData”, grant no. 9C436.

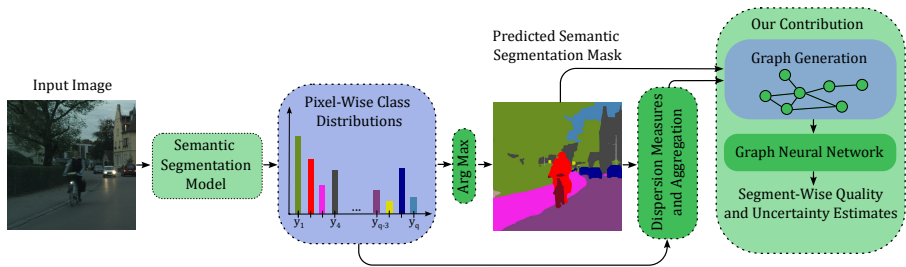


Figure 1: An illustration of our graph-based uncertainty estimation and prediction quality estimation approach.

However, their decision-making process for predictions is mostly opaque, and the quality of a particular prediction is not quantified in standard semantic segmentation DNNs. Typically, the quality of such models is evaluated empirically by using collected data samples to test the model’s predictive ability, resulting in a quantity averaged over the whole test set. When a semantic segmentation DNN is deployed for a given safety-critical task, it is of importance to have an indication of the performance of the DNN for the given data rather than an average performance over a greater test set [29]. Most of the early works in uncertainty estimation for semantic segmentation DNNs focused on a pixel-wise representation of the uncertainty estimates [52, 53] also rather focusing on a Bayesian point of view. It can be argued, however, that in practice a strict separation of aleatoric and epistemic uncertainty is not necessarily of importance [29].

Inspired by the idea that an uncertainty estimate on an object level, rather than pixel level, would be desirable, a line of research pursues uncertainty estimates and prediction quality estimation on the level of class-wise connected components (segments) in the predicted semantic segmentation masks obtained by the DNN [5, 6, 50, 52].

These works focus on extracting features or hand-crafted metrics from the DNN’s softmax output, therefrom performing a so-called meta classification and meta regression. Both tasks require a little hold out dataset to train on. In the given context, meta classification refers to discriminating true positive from false positive predictions and meta regression refers to the task of estimating the segment-wise intersection over union (IoU) of the given predicted segment with the ground truth. In both tasks, small-scale post-processing models perform predictions based on the hand-crafted metrics that are aggregated over the given segment under inspection. The works [19, 40] extend this post-processing framework by a temporal component. However, the spatial component has not been considered up to now. Neighboring segments may provide additional hints on whether a given predicted segment is of uncertain or of low quality, e.g. in street scenes it might be unlikely to observe a human in the sky.

To study the neighbor dependence in the tasks of meta classification and meta regression for semantic segmentation DNNs, we use different GNNs [23, 60] to model the neighbor relations of the predicted segments and their effect on the segment-wise IoU as well as the correctness of the given predicted segment. Those different GNN models include the utilization of basic aggregation functions [23] and attention-mechanisms [60]. The overall method is illustrated in Fig. 1. We evaluate varying architectures of the different GNN types on the publicly available Cityscapes [4] dataset. The best GNN models outperform the baseline models, not considering any neighboring segments, by up to 1.78 percentage points in terms of AUROC in the classification task.

Our main contributions in this work are:

- We incorporate information from neighboring predicted segments into meta classification and meta regression for semantic segmentation DNNs by modeling the spatial relationship between neighboring segments as edges in a graph.
- We perform a comprehensive study of the performance of several GNN architectures as well as models for semantic segmentation of street scenes.
- Our method outperforms baseline models that do not consider neighboring segments by up to 1.78 percentage points in the classification task.

We make our implementation publicly available under <https://github.com/eheinert/Uncertainty-and-Prediction-Quality-Estimation-for-SemSeg-via-GNNs>.

2 Related Work

Quality and uncertainty estimation for DNN’s prediction are both closely related tasks, as a high quality prediction is often associated with low uncertainty. Accordingly, both tasks are addressed in the related work section, which is divided into two parts: pixel-wise and segment-wise approaches for semantic segmentation and related tasks.

Pixel-Wise Uncertainty. One of the first uncertainty estimates at inference time in semantic segmentation models was Monte Carlo (MC) dropout proposed in [33] as an extension of [21]. The authors employ Monte Carlo sampling, implemented through the use of dropout [67] during inference, to estimate uncertainty for each pixel of the model’s prediction. Dropout for uncertainty estimation for DNNs was initially proposed in [21]. A variety of approaches [16, 52, 53] utilize an uncertainty measure in the prediction of semantic segmentation models through MC dropout. However, this approach has the disadvantage of requiring multiple inferences on an image to estimate the uncertainty. In real-time applications such as autonomous driving, the computational burden associated with the utilization of MC dropout as an uncertainty estimation method may be considerable.

In [51], the authors propose to use a five ensemble for clinical target volume segmentation of the prostate in ultrasound images, providing a predictive pixel-wise quality estimation. Therein, the pairwise Dice Similarity Coefficient [17] of the ensemble predictions is used as a quality estimate. In [28], knowledge distillation methods [76] transfer uncertainty estimates from a teacher ensemble segmentation to a single student model, eliminating the necessity for multiple inferences. A semantic segmentation approach that leverages hyperbolic embeddings instead of the conventional Euclidean ones is proposed in [11], providing a pixel-wise quality estimate. All of these methods ultimately yield pixel-wise or (via averaging) image-wise uncertainty measures or quality estimates.

Segment-Wise Uncertainty. Initially, [51] proposed a methodology for estimating the quality and uncertainty of each segment of a segmentation model’s prediction, as opposed to the pixel or image-level. Estimates at the segment-level are more conducive to integration into a downstream automated decision-making process, such as that employed in autonomous driving. Based on segment-wise features carrying uncertainty, the segment-wise IoU with the ground truth is estimated (termed meta regression) as well as the probability of a correct prediction (termed meta classification). The entire framework is termed MetaSeg. A variety of methods make use of MetaSeg for the purpose of detecting or reducing false positive and false negative samples [8, 40, 52], identifying out-of-distribution objects [9, 6, 43], implementing active learning strategies [11], anomaly detection [7, 53], detecting label errors in annotated datasets [49] or serving for segment-wise analysis purposes [25, 43].

MetaSeg is employed in [50] on nested image crops of an image, yielding a batch of pixel-wise class distributions, which are combined before applying MetaSeg. Using multiple crops enhances the performance of the segmentation model and of the meta models - a term we use to refer to both meta classification as well as regression models. In [41], an extension of MetaSeg incorporates temporal aspects, integrating the segment-wise uncertainty derived from preceding frames (in the context of video data) into the current frame’s segment-wise estimation of uncertainty and quality. Furthermore, temporal MetaSeg is employed within the domain of instance segmentation [39, 42]. Also extensions of MetaSeg to 2D objects detection [45, 54], 3D object detection [46] and 3D Lidar point cloud segmentation [12, 13] are available. However, none of the approaches to uncertainty estimation at the segment-level make use of information from neighboring segments.

Furthermore, in a recent line of work, there are approaches that estimate mask-wide uncertainty for anomaly segmentation and predict outliers [15, 22].

3 Neighbor-Aware Meta Classification and Regression

Our approach extends a semantic segmentation post-processing method by Rottmann et al. [51], which estimates the quality of each predicted segment using metrics, or in other words constructed features that carry uncertainty information, derived from the softmax output. However, [51] treats the derived metrics of each segment independently and does not utilize information about a segment’s neighborhood. To incorporate neighborhood information of predicted segments, we introduce a data structure that captures this information and apply meta-estimators capable of leveraging it. The natural choice for such a data structure is an undirected graph representing the information of each frame. For the estimators, we employ inductive GNNs, such as GraphSAGE and Graph Attention Networks (GAT), which are suitable for graphs of arbitrary shapes.

Graph and Feature Construction. In this paragraph, except for the graph construction, we mainly follow [51]. For an image $x \in [0, 1]^{h \times b \times d}$ and one of its pixels z , the ground truth semantic segmentation with $q + 1$ classes is $y \in \mathcal{C} = \{y_1, \dots, y_{q+1}\}$ with class y_{q+1} being unlabeled. The pixel-wise raw softmax output of a semantic segmentation network for z , $f_z(y|x, w)$ is a probability distribution over the first q classes \mathcal{C}' and the networks pixel-wise prediction $\hat{y}_z \in \mathcal{C}'$ is given by $\hat{y}_z = \arg \max_{y \in \mathcal{C}'} f_z(y|x, w)$.

Let $\hat{\mathcal{K}}_x = \{\hat{k}_1, \dots, \hat{k}_M\}$ be the set of predicted segments and \mathcal{K}_x the set of ground truth segments of a given image. Herein, a segment refers to a connected component of a given class in a segmentation mask where all eight surrounding pixels are considered to be adjacent. We then define the image-wise graph $\mathbf{G}_x = (\hat{\mathcal{K}}_x, \hat{\mathcal{E}}_x)$ with $\hat{e}_{l,m} := \{\hat{k}_l, \hat{k}_m\} \in \hat{\mathcal{E}}_x$, iff a pixel in \hat{k}_l neighbors a pixel in \hat{k}_m . An illustration of such a graph can be found in Fig. 2. As before,

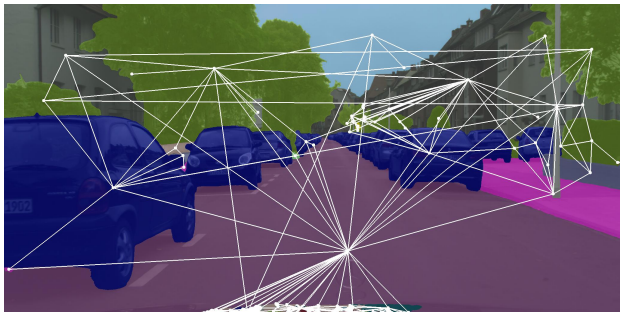


Figure 2: Cityscapes image overlaid with a graph generated from segment barycenters. Each segment is connected to adjacent ones.

for each predicted segment \hat{k} in \mathcal{K}_x , a number of scalar metrics $h \in \mathbb{R}^p$ are derived, which are the node-wise input vectors of our meta estimators. The metrics are based on geometric properties of the respective predicted segments like the set of inner pixels \hat{k}_{in} , the boundary $\hat{k}_{bd} = \hat{k} \setminus \hat{k}_{in}$, as well as dispersion metrics like the pixel-wise entropy and difference in two highest probabilities,

$$E_z(x, w) = -\frac{1}{\log(q)} \sum_{y \in \mathcal{C}'} f_z(y|x, w) \log f_z(y|x, w), \quad (1)$$

$$D_z(x, w) = 1 - f_z(\hat{y}_z(x, w) | x, w) + \max_{y \in \mathcal{C}' \setminus \{\hat{y}_z(x, w)\}} f_z(y|x, w). \quad (2)$$

Further metrics can be found in [6, 51] and our implementation as we only present an excerpt:

- the number of all segments $|\hat{\mathcal{K}}_x|$ and the specific segment's barycenter. $(\bar{h}_{\hat{k}}, \bar{w}_{\hat{k}})$
- the number of all, inner, and boundary pixels of a segment and their relations $S = |\hat{k}|$, $S_{in} = |\hat{k}_{in}|$, $S_{bd} = |\hat{k}_{bd}|$, $S_{rel} = S/S_{bd}$ etc.
- average entropies and probability distances \bar{E} , \bar{E}_{in} , \bar{E}_{bd} , \bar{D} , \bar{D}_{in} , \bar{D}_{bd} defined as

$$\bar{E}_{\#} = \frac{1}{S_{\#}} \sum_{z \in \hat{k}_{\#}} E_z(x), \quad \bar{D}_{\#} = \frac{1}{S_{\#}} \sum_{z \in \hat{k}_{\#}} D_z(x) \quad \text{for } \# \in \{-, in, bd\} \quad (3)$$

and analogous variances.

- the class-wise average softmax probabilities $\bar{p}_y = \frac{1}{S} \sum_{z \in \hat{k}} f_z(y|x, w)$ for $y \in \mathcal{C}'$.

For $\mathcal{K}_x|_{\hat{k}}$, the set of ground truth segments that intersect with \hat{k} and have the same ground truth class as the predicted class for \hat{k} , $K' = \cup_{k' \in \mathcal{K}_x|_{\hat{k}}} k'$ and $Q = \{q \in \hat{\mathcal{K}}_x \setminus \{\hat{k}\} : q \cap K' \neq \emptyset\}$ we finally define the adjusted IoU [51]

$$\text{IoU}_{\text{adj}}(\hat{k}) = \frac{|\hat{k} \cap K'|}{|\hat{k} \cup (K' \setminus Q)|} \quad (4)$$

as the target metric. The adjusted IoU was proposed to take into account that more than one of the predicted segments with the same class can significantly intersect with the same ground truth segment.

Meta Classification and Regression. Meta regression and classification are both segment-wise tasks. For each predicted segment, we perform two learning tasks. *Meta Regression* refers to the task of predicting $\text{IoU}_{\text{adj}}(\hat{k})$ for a given segment \hat{k} based on its metrics / constructed features introduced in the previous subsection. *Meta Classification* refers in analogy to meta regression to a corresponding classification task where the $\text{IoU}_{\text{adj}}(\hat{k})$ is replaced by the binarized target quantity

$$\text{IoU}_0(\hat{k}) = \begin{cases} 0 & \text{if } \text{IoU}_{\text{adj}}(\hat{k}) = 0, \\ 1 & \text{if } \text{IoU}_{\text{adj}}(\hat{k}) > 0. \end{cases} \quad (5)$$

While meta regression aims at estimating prediction quality of localization quality for a predicted segment, meta classification aims at estimating the uncertainty about the correctness of the prediction \hat{k} .

Graph Neural Networks. In the following we use the metrics from above as covariables to estimate certain quantities, giving rise to uncertainty as well as prediction / localization quality. To also incorporate the features of neighboring predicted segments, we utilize GNNs. In the following we introduce the types of GNN layers used in our method.

GraphSAGE [23] was designed for inductive learning, which means that the model is generally applicable to undirected finite graphs of arbitrary shape and, therefore, aligns well with the objective of meta classification and meta regression. The fundamental concept of a GraphSAGE layer is the aggregation of node features from a uniformly sampled subset of all one-hop neighbors for each node, which are then concatenated with the node’s own features. In this context, the term *k-hop neighbor* is defined as a node that can be reached via the shortest path of length k from the node under consideration. The input-output characteristic of a layer for each node $\hat{k} \in \hat{\mathcal{K}}_x$ is described by (cf. [23])

$$h_{\tilde{\mathcal{N}}(\hat{k})} = \text{AGGREGATE}\left(\{h_{\hat{v}} \mid \hat{v} \in \tilde{\mathcal{N}}(\hat{k})\}\right), \quad (6a)$$

$$\tilde{h}_{\hat{k}} = \sigma\left(W \cdot \text{CONCAT}(h_{\hat{k}}, h_{\tilde{\mathcal{N}}(\hat{k})})\right), \quad (6b)$$

where $h_i \in \mathbb{R}^p$ is the feature vector of node i , W a learnable weight matrix, σ an activation function, and the function $\tilde{\mathcal{N}}(\hat{k})$ gives a uniformly sampled subset of all one-hop neighborhood node indices of node \hat{k} . As the set of one-hop neighbors feature vectors is unordered, the authors propose the use of aggregation functions that are invariant to permutations of their inputs. Examples of such functions include the mean and the maximum.

Graph Attention Networks (GATs) [60] are analogous to GraphSAGE in that they can be utilized for inductive learning and leverage the local neighborhood structure. However, the fundamental concept is to employ an attention mechanism [2] for the aggregation of the one-hop neighbor’s features. A pairwise attention weight $\alpha_{\hat{k}\hat{v}}$ between a node $\hat{k} \in \hat{\mathcal{K}}_x$ and its one-hop neighbors $\hat{v} \in \mathcal{N}(\hat{k}) \cup \{\hat{k}\}$ (including \hat{k}) is employed to form the output (cf. [60])

$$\tilde{h}_{\hat{k}} = \sum_{\hat{v} \in \mathcal{N}(\hat{k}) \cup \{\hat{k}\}} \alpha_{\hat{k}\hat{v}} W h_{\hat{v}} \quad (7)$$

of the GAT layer for each node. This is extended to a multi-head attention mechanism, where Eq. (7) is applied K -fold in parallel with different weight matrices $\{W^l\}$ and attention weights $\{\alpha_{\hat{k}\hat{v}}^l\}$. The resulting set $\{\tilde{h}_{\hat{k}}^l\}$ is then averaged to produce the final output $\tilde{h}_{\hat{k}}$.

4 Numerical Experiments

In this section, we investigate the performance of our method and present an analysis of the impact of the choice of hyperparameters, such as depth, width, layer types, and learning rate, of graph-based meta classification and meta regression models. We utilized the same segment-wise metrics for all experiments, and a fixed 80/20 split of the validation set of the publicly available Cityscapes dataset [14] was employed for training and evaluation. The validation set comprises 500 pixel-wise labeled street scene images with a resolution of 2048×1024 pixels. Moreover, we utilized the DeepLabv3+ [8] network for semantic segmentation with two backbones: WideResNet38 [61] and ResNet18 [24]. For the sake of brevity, they will be referred to as DV3+WRN38 and DV3+RN18, respectively. We use trained checkpoints from [63] for DV3+WRN38, while we trained DV3+RN18 with only 50 annotated images to obtain a model with low mIoU, which may affect our downstream task. The model DV3+WRN38 demonstrates excellent semantic segmentation performance with a mIoU of 83.45% on the Cityscapes validation set. DV3+RN18 reaches a comparatively low mIoU of 46.29% on the same validation data. Our graph-based meta model employs a combination of GAT, GraphSAGE, and linear layers (fully connected layers). Specifically,

Model	LR	1. Layer	2. Layer	3. Layer	4. Layer
LLLL _C	0.021	Linear (77)	Linear (57)	Linear (144)	Linear (1)
GSSL _C	0.015	GAT (91)	SAGE-mean (93)	GAT (154)	Linear (1)
LLSG _C	0.007	Linear (293)	Linear (39)	SAGE-mean (179)	GAT (1)
LS _C	0.002	Linear (242)	SAGE-mean (1)	-	-
LSS _C	0.009	Linear (373)	SAGE-mean (269)	SAGE-mean (1)	-
SSS _C	0.002	SAGE-mean (371)	SAGE-mean (283)	SAGE-mean (1)	-
LLSS _C	0.013	Linear (304)	Linear (57)	SAGE-mean (161)	SAGE-mean (1)
LLSL _C	0.015	Linear (117)	Linear (22)	SAGE-mean (154)	Linear (1)
LL _R	0.001	Linear (317)	Linear (1)	-	-
LS _R	0.001	Linear (314)	SAGE-mean (1)	-	-
LLS _R	0.006	Linear (145)	Linear (103)	SAGE-mean (1)	-
LSS _R	0.009	Linear (142)	SAGE-mean (137)	SAGE-mean (1)	-
LLSL _R	0.003	Linear (337)	Linear (292)	SAGE-mean (99)	Linear (1)

Table 1: Selected meta classification (top) and meta regression (bottom) architectures. Learning rate (LR) and the layer configuration of each selected model architecture are presented. The number of neurons in each layer is indicated in parentheses. The sequence of letters in the model name indicates the number of layers and the types. The letter L represents a linear layer, S denotes a GraphSAGE layer, and G represents a GAT layer.

a Bayesian optimization method from the SMAC3 package [58] was employed to identify optimal model architectures by using the validation data of Cityscapes.

The remainder of this section is organized in the following manner: First, the results of the graph-based meta classification and graph-based meta regression experiments are shown in a baseline comparison. Thereafter, the process of selecting the GNN architectures for the graph-based meta estimators is described in detail, and an analysis of the impact of the choice of hyperparameters is provided. Finally, a comprehensive examination of the performance of the meta models with a particular focus on class-wise evaluation is presented.

Evaluation Procedure. All model architectures (cf. Table 1) identified in the selection process were trained five times to account for the randomness inherent in the learnable parameter initialization process and the resulting randomness in the optimization trajectory in parameter space. Moreover, the Mean Square Error (MSE) loss function and the Adam optimizer [65] were employed during the training phase. The learning rates (cf. Table 1) identified by the Bayesian optimization process for selecting the model architectures were utilized for training. The meta models were trained for 200 epochs and the performance metrics of the best model instances within each run are considered for evaluation. In addition, the performance of our proposed graph-based meta models is compared to a baseline comprising a set of models that do not incorporate information from neighboring segments. The baseline models are: logistic and linear regression as employed in the original MetaSeg framework [61], a gradient boosting model, GNNs without neighbors (in essence fully connected neural networks), and the two best-performing GNNs, which are utilized with graphs without edges. All meta models have been implemented in Python by using either Pytorch Geometric (PyG) [68] or Scikit-learn [64].

Graph-Based Meta Classification. In order to evaluate the meta classification models that estimate the segment-wise uncertainty of the segmentation models’ predictions, the area under the receiver operating characteristic curve (AUROC) [9] and the F1-score [10, 69] are employed as performance metrics. The results are presented in Table 2 where in each row the means and the standard deviations of the performance metrics of the five model instances are given. Furthermore, the results of the five baseline models are presented in the top section of the table, while the GNNs’ utilising neighboring segments’ information are at the bottom.

The best-performing model for the DV3+WRN38 case is the LSS_C GNN with an av-

Model	DV3+WRN38		DV3+RN18	
	AUROC (%)	F1-Score (%)	AUROC (%)	F1-Score (%)
logistic regression	95.06 ± 0.00	82.09 ± 0.04	83.22 ± 0.00	65.11 ± 0.00
gradient boosting	95.95 ± 0.00	83.54 ± 0.00	84.46 ± 0.00	66.06 ± 0.00
LLLL _C	95.87 ± 0.02	83.46 ± 0.08	84.59 ± 0.04	66.57 ± 0.32
LS _{C,no edges}	95.95 ± 0.02	83.48 ± 0.12	84.69 ± 0.06	66.73 ± 0.40
LSS _{C,no edges}	95.89 ± 0.01	83.33 ± 0.08	84.65 ± 0.04	66.08 ± 0.26
LLSS _{C,no edges}	95.93 ± 0.01	83.51 ± 0.18	84.65 ± 0.09	66.32 ± 0.54
GSGL _C	87.95 ± 12.32	64.06 ± 32.05	80.79 ± 0.10	59.43 ± 1.12
LLSG _C	95.20 ± 0.58	82.48 ± 1.25	83.11 ± 0.58	65.31 ± 0.88
LS _C	96.10 ± 0.02	83.72 ± 0.11	85.00 ± 0.06	67.14 ± 0.26
LSS _C	96.11 ± 0.02	83.81 ± 0.21	84.94 ± 0.04	66.69 ± 0.30
SSS _C	96.04 ± 0.02	83.65 ± 0.30	84.73 ± 0.42	66.22 ± 0.56
LLSS _C	96.10 ± 0.03	83.77 ± 0.10	84.83 ± 0.12	66.77 ± 0.51
LLSL _C	96.05 ± 0.02	83.69 ± 0.06	84.84 ± 0.06	66.35 ± 0.29

Table 2: Evaluation results of the meta classification models for WideResNet38 (column 2+3) and ResNet18 (column 4+5) as respective backend of the DeepLabV3+ segmentation model. The results are averages over five runs and the numbers are in percent.

average AUROC of 96.11% and a mean F1-score of 83.81%. The LSS_C model is composed of a linear layer followed by two GraphSAGE layers with mean aggregation functions (cf. Table 1). All baseline models are outperformed by LSS_C, with the largest difference being 1.05 percent points relative to the logistic regression model and the smallest difference being 0.16 percent points relative to the gradient boosting model. It should be noted that five out of the seven GNN-based models demonstrate superior performance compared to the strongest baseline models, namely the gradient boosting and LS_{C, no edges}. Furthermore, the standard deviations are notably small, with the exception of the GNNs that contain GAT layers, specifically GSGL_C and LLSG_C. Indeed, the GSGL_C model, comprising two GAT layers, exhibits the poorest performance among all models, while LLSG_C, which contains a single GAT layer, outperforms only the logistic regression baseline.

A comparable image can be derived from the evaluation results for the weaker DV3+RN18 model. Five of the seven GNNs demonstrate superior performance compared to all baseline models. The LS_C GNN exhibits the highest performance, achieving a mean AUROC of 85% and an average F1-score of 67.14%. The difference between this result and that of the weakest baseline, namely logistic regression, is 1.78 percentage points. As was the case with the DV3+WRN38 result, the GNNs containing GAT layers perform the worst.

Graph-Based Meta Regression.

In the evaluation of the graph-based meta regression models that estimate the segment-wise prediction quality of a segmentation model, the coefficient of determination (R^2 -score cf. [9]) is employed as a performance metric. As with the meta classification case, all GNN architectures for graph-based meta regression (cf. Table 1) that were identified in the selection process, were trained and evaluated five times and the means and the standard deviations of the R^2 -

scores are reported. The results for the DV3+WRN38 and DV3+RN18 cases are presented in Table 3.

Model	DV3+WRN38	DV3+RN18
	R^2 -Score (%)	R^2 -Score (%)
Linear Regression	83.45	74.59
Gradient Boosting	85.05 ± 0.00	77.41 ± 0.00
LL _R	85.71 ± 0.04	78.27 ± 0.07
LS _{R,no edges}	85.67 ± 0.03	78.25 ± 0.07
LLSL _{R,no edges}	85.32 ± 0.03	77.92 ± 0.14
LS _R	85.58 ± 0.04	78.29 ± 0.06
LLS _R	85.39 ± 0.04	77.97 ± 0.10
LSS _R	85.33 ± 0.07	53.05 ± 43.07
LLSL _R	85.47 ± 0.09	78.20 ± 0.10

Table 3: Evaluation results of the meta regression models for WideResNet38 (column 2) and ResNet18 (column 3) as respective backend of the DeepLabV3+ segmentation model. The results are averages over five runs and displayed in percent.

The best-performing meta regression model in the DV3+WRN38 case with an average R^2 -score of 85.71% is the baseline model LL_R that is composed of two linear layers. The LS_R model, which is the best-performing GNN that considers information from neighboring segments, shows a 0.13 percent point inferiority, and, unexpectedly, a 0.09 percent point lower performance as well, in comparison to the corresponding $LS_{R, \text{no edges}}$ model that does not incorporate information from the segments’ neighborhoods. In the DV3+RN18 case, the best-performing model is the LS_R GNN that is composed of a linear and GraphSAGE layer with a mean aggregation function. The model archives an average R^2 -score of 78.29%, which is slightly superior to the strong LL_R baseline model.

Setup of the Architecture Selection Process and the Hyperparameter Analysis. A Bayesian optimization method from the SMAC3 package [53] was employed to identify model architectures for evaluation. This utilized the fixed 80/20 data split from the Cityscapes [14] validation set and the trained DV3+WRN38 segmentation model. The hyperparameters of the model’s architecture that were optimized include the types of layers, the depth of the model, the width of the layers, and the learning rate. The algorithm has the option to select for each layer stage (independently) either a linear layer, that does not consider information from neighboring segments, or one of the three GNN layers: GAT, GraphSAGE with a maximum aggregation function, and GraphSAGE with a mean aggregation function, respectively. Additionally, a rectified linear unit (ReLU) activation function was incorporated between each layer of the model. The remaining hyperparameter values were constrained as follows: The maximum depth of the model architecture was fixed at four, as a GNN comprising a total of four GraphSAGE and GAT layers, respectively, incorporates four-hop neighboring features in the node embeddings (cf. [23, 60]). We observed in our experiments that depth four is sufficient for the purposes of our investigation. Moreover, the minimum depth was set at two. The number of neurons in each layer was explored in a range between 10 and 400. Consequently, the identified model architectures through Bayesian optimization comprise a depth of up to four layers, with up to 400 neurons in each layer. Each layer is either a linear layer, a GAT layer, or a GraphSAGE layer with mean or maximum aggregation function. It should be noted, however, that in all cases the final layer is composed of a single neuron and that the subsequent activation function is always the sigmoid. Additionally, the learning rate was explored in the range of [0.001, 0.2].

Selection Process. From the optimization results, the best-performing model architectures were selected for each model depth value. For a given depth value, if multiple architectures obtained similar top performances, we selected two considerably different architectures among them for further experiments. Moreover, in the graph-based meta classification case, the two best-performing architectures that contain GAT layers were selected for further evaluation. Furthermore, each best-performing model for meta classification and meta regression that contains only linear layers ($LLLL_C$ and LL_R) was chosen as baseline. The selected architectures are shown in Table 1.

Hyperparameter Analysis. In the course of Bayesian optimization of the graph-based meta classifiers, 142 distinct model architectures were evaluated in accordance with the methodology and definition of the hyperparameter space described before. The characteristics of the 80 best-performing models are analyzed in the following, as a notable decline in the AUROC value is observed within that region (cf. Fig. 3(a)).

In Fig. 3(b), a histogram of the number of layers in the model and the number of GNN-based layers within the architecture is presented. The number of layers in the model is displayed in orange, while the number of GNN-based layers within the top 80 architectures is shown in blue. Fifty-one of the top 80 ranked model architectures contain two layers.

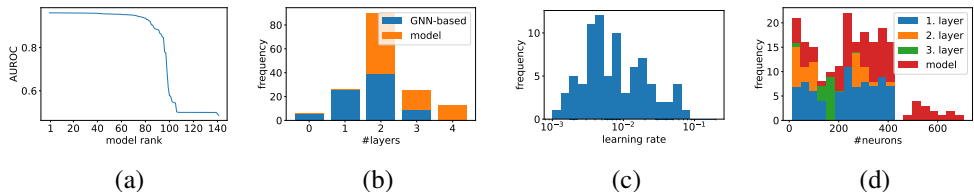


Figure 3: Illustration of the characteristics of the Bayesian optimization results for the top 80th ranked model architectures. (a) AUROC course over the model rank (models ordered from highest to lowest AUROC). (b) histogram of the number of layers in the model (orange) and the number of GNN-based layers within the architecture (blue). (c) distribution of the learning rate of the model architectures. (d) histogram of the number of neurons in the layers (blue, orange, green), except each last layer with only one fixed neuron, and the total number of neurons in the architecture (red).

The remaining 29 are distributed approximately equally between architectures with a depth of three and four, respectively. Noteworthy, among the top 80 architectures, 65 contain one or two GNN-based layers. This observation leads to the conclusion that integrating information from one- or two-hop neighboring segments is a sufficient approach. Only nine architectures comprise three GNN-based layers, while six architectures have no GNN-layer at all. Consequently, these latter six architectures are comprised by linear layers only. In the top 80 architectures, 66 models contain at least one GraphSAGE layer, with 40 having a mean and 26 having a maximum aggregation function. However, we found that maximum aggregation was generally less reliable than mean aggregation, frequently exhibiting a tendency to collapse in repeated training runs, resulting in an AUROC value of approximately 50%. Therefore, such model architectures were excluded from the final selection process. Furthermore, 30 out of the 80 top-ranked architectures contain at least one GAT layer. It should be noted that the first GNN comprising at least one GAT layer is ranked 45th. Architectures with GAT layers demonstrate a comparable outcome across multiple training runs in our experiments, as evidenced by models with GraphSAGE and maximum aggregation functions (cf. high standard deviation in Table 2).

The learning rate distribution of the top 80 architectures demonstrates a concentration around a learning rate of 3×10^{-3} (cf. Fig. 3(c)). During the Bayesian optimization process of the top 80 architectures, no models were trained with a learning rate exceeding 8×10^{-2} . The distribution of the number of neurons in the first layer is approximately uniform (cf. blue bars in Fig. 3(d)), while the distributions for the second (orange bars) and third layer (green bars) are not. The former one has got two high-density intervals at approximately $[10, 112]$ and $[250, 325]$ and the latter one at $[125, 175]$.

5 Conclusion and Outlook

In this work we have demonstrated that the relationships between neighboring predicted segments produced by DNNs are helpful for uncertainty estimation and prediction quality estimation. Notable performance improvements over the presented baselines were achieved. Our hyperparameter study revealed that GNNs of moderate size are sufficient to address corresponding estimation tasks. For future work it would be interesting to use GNNs not only for spatial but also temporal relationships in a unified manner. Furthermore, a performance investigation on out-of-distribution objects and on data with domain shift could be of interest.

Acknowledgment

We would like to thank Daniel Siemssen for his invaluable contribution in transferring the base code to Python and significantly improving its efficiency. E.H. and M.R. acknowledge support by the German Federal Ministry of Education and Research within the junior research group project “UnrEAL” (grant no. 01IS22069). S.T. is funded by the Volkswagen-Stiftung, “DigiData” (grant no. 9C436).

References

- [1] Mina Ghadimi Atigh, Julian Schoep, Erman Acar, Nanne Van Noord, and Pascal Mettes. Hyperbolic image segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4453–4462, 2022.
- [2] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*, 2014.
- [3] Andrew P Bradley. The use of the area under the roc curve in the evaluation of machine learning algorithms. *Pattern recognition*, 30(7):1145–1159, 1997.
- [4] 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)*, volume 2, page 2, 2020.
- [5] Robin Chan, Matthias Rottmann, Fabian Hüger, Peter Schlicht, and Hanno Gottschalk. Controlled false negative reduction of minority classes in semantic segmentation. In *2020 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE, 2020.
- [6] Robin Chan, Matthias Rottmann, and Hanno Gottschalk. Entropy maximization and meta classification for out-of-distribution detection in semantic segmentation. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 5128–5137, 2021.
- [7] Robin Chan, Svenja Uhlemeyer, Matthias Rottmann, and Hanno Gottschalk. Detecting and learning the unknown in semantic segmentation. In *Deep Neural Networks and Data for Automated Driving: Robustness, Uncertainty Quantification, and Insights Towards Safety*, pages 277–313. Springer International Publishing Cham, 2022.
- [8] Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation. In *Proceedings of the European conference on computer vision (ECCV)*, pages 801–818, 2018.
- [9] Davide Chicco, Matthijs J Warrens, and Giuseppe Jurman. The coefficient of determination r-squared is more informative than smape, mae, mape, mse and rmse in regression analysis evaluation. *PeerJ computer science*, 7:e623, 2021.

- [10] Peter Christen, David J Hand, and Nishadi Kirielle. A review of the f-measure: its history, properties, criticism, and alternatives. *ACM Computing Surveys*, 56(3):1–24, 2023.
- [11] Pascal Colling, Lutz Roese-Koerner, Hanno Gottschalk, and Matthias Rottmann. Metabox+: A new region based active learning method for semantic segmentation using priority maps. *arXiv preprint arXiv:2010.01884*, 2020.
- [12] Pascal Colling, Matthias Rottmann, Lutz Roese-Koerner, and Hanno Gottschalk. False positive detection and prediction quality estimation for lidar point cloud segmentation. In *2021 IEEE 33rd International Conference on Tools with Artificial Intelligence (IC-TAI)*, pages 18–25. IEEE, 2021.
- [13] Pascal Colling, Matthias Rottmann, Lutz Roese-Koerner, and Hanno Gottschalk. Prediction quality meta regression and error meta classification for segmented lidar point clouds. *International Journal on Artificial Intelligence Tools*, 32(5), 2023.
- [14] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3213–3223, 2016.
- [15] Anja Delić, Matej Grcić, and Siniša Šegvić. Outlier detection by ensembling uncertainty with negative objectness. *arXiv preprint arXiv:2402.15374*, 2024.
- [16] Terrance DeVries and Graham W. Taylor. Leveraging uncertainty estimates for predicting segmentation quality, 2018. URL <https://arxiv.org/abs/1807.00502>.
- [17] Lee R. Dice. Measures of the amount of ecologic association between species. *Ecology*, 26(3):297–302, 1945. doi: <https://doi.org/10.2307/1932409>. URL <https://esajournals.onlinelibrary.wiley.com/doi/abs/10.2307/1932409>.
- [18] Matthias Fey and Jan E. Lenssen. Fast graph representation learning with PyTorch Geometric. In *ICLR Workshop on Representation Learning on Graphs and Manifolds*, 2019.
- [19] Laura Fieback, Bidya Dash, Jakob Spiegelberg, and Hanno Gottschalk. Temporal performance prediction for deep convolutional long short-term memory networks. In *International Workshop on Advanced Analytics and Learning on Temporal Data*, pages 145–158. Springer, 2023.
- [20] Kerr Fitzgerald, Jorge Bernal, Aymeric Histace, and Bogdan J. Matuszewski. Polyp segmentation with the fcb-swinv2 transformer. *IEEE Access*, 12:38927–38943, 2024. doi: 10.1109/ACCESS.2024.3376228.
- [21] Yarin Gal and Zoubin Ghahramani. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *international conference on machine learning*, pages 1050–1059. PMLR, 2016.
- [22] Matej Grcić, Josip Šarić, and Siniša Šegvić. On advantages of mask-level recognition for outlier-aware segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2937–2947, 2023.

- [23] Will Hamilton, Zhitao Ying, and Jure Leskovec. Inductive representation learning on large graphs. *Advances in neural information processing systems*, 30, 2017.
- [24] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.
- [25] Edgar Heinert, Matthias Rottmann, Kira Maag, and Karsten Kahl. Reducing texture bias of deep neural networks via edge enhancing diffusion. *arXiv preprint arXiv:2402.09530*, 2024.
- [26] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network, 2015. URL <https://arxiv.org/abs/1503.02531>.
- [27] Stefan Hoermann, Martin Bach, and Klaus Dietmayer. Dynamic occupancy grid prediction for urban autonomous driving: A deep learning approach with fully automatic labeling. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2056–2063. IEEE, 2018.
- [28] Christopher J. Holder and Muhammad Shafique. Efficient uncertainty estimation in semantic segmentation via distillation. In *2021 IEEE/CVF International Conference on Computer Vision Workshops (ICCVW)*, pages 3080–3087, 2021. doi: 10.1109/ICCVW54120.2021.00343.
- [29] Eyke Hüllermeier and Willem Waegeman. Aleatoric and epistemic uncertainty in machine learning: An introduction to concepts and methods. *Machine learning*, 110(3): 457–506, 2021.
- [30] Chulhoon Jang and Myoungcho Sunwoo. Semantic segmentation-based parking space detection with standalone around view monitoring system. *Machine Vision and Applications*, 30(2):309–319, 2019.
- [31] Davood Karimi, Qi Zeng, Prateek Mathur, Apeksha Avinash, Sara Mahdavi, Ingrid Spadinger, Purang Abolmaesumi, and Septimiu E. Salcudean. Accurate and robust deep learning-based segmentation of the prostate clinical target volume in ultrasound images. *Medical Image Analysis*, 57:186–196, 2019. ISSN 1361-8415. doi: <https://doi.org/10.1016/j.media.2019.07.005>. URL <https://www.sciencedirect.com/science/article/pii/S1361841519300623>.
- [32] Alex Kendall and Yarin Gal. What uncertainties do we need in bayesian deep learning for computer vision? In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. URL https://proceedings.neurips.cc/paper_files/paper/2017/file/2650d6089a6d640c5e85b2b88265dc2b-Paper.pdf.
- [33] Alex Kendall, Vijay Badrinarayanan, and Roberto Cipolla. Bayesian segnet: Model uncertainty in deep convolutional encoder-decoder architectures for scene understanding. In Gabriel Brostow Tae-Kyun Kim, Stefanos Zafeiriou and Krystian Mikolajczyk, editors, *Proceedings of the British Machine Vision Conference (BMVC)*, pages 57.1–57.12. BMVA Press, September 2017. ISBN 1-901725-60-X. doi: 10.5244/C.31.57. URL <https://dx.doi.org/10.5244/C.31.57>.

- [34] Adel Kermi, Issam Mahmoudi, and Mohamed Tarek Khadir. Deep convolutional neural networks using u-net for automatic brain tumor segmentation in multimodal mri volumes. In *Brainlesion: Glioma, Multiple Sclerosis, Stroke and Traumatic Brain Injuries: 4th International Workshop, BrainLes 2018, Held in Conjunction with MICCAI 2018, Granada, Spain, September 16, 2018, Revised Selected Papers, Part II 4*, pages 37–48. Springer, 2019.
- [35] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.
- [36] Der-Hau Lee and Jinn-Liang Liu. End-to-end deep learning of lane detection and path prediction for real-time autonomous driving. *Signal, Image and Video Processing*, 17(1):199–205, 2023.
- [37] Jia Li and Zengfu Wang. Real-time traffic sign recognition based on efficient cnns in the wild. *IEEE Transactions on Intelligent Transportation Systems*, 20(3):975–984, 2018.
- [38] Marius Lindauer, Katharina Eggensperger, Matthias Feurer, André Biedenkapp, Difan Deng, Carolin Benjamins, Tim Ruhkopf, René Sass, and Frank Hutter. Smac3: A versatile bayesian optimization package for hyperparameter optimization. *Journal of Machine Learning Research*, 23(54):1–9, 2022. URL <http://jmlr.org/papers/v23/21-0888.html>.
- [39] Kira Maag. False negative reduction in video instance segmentation using uncertainty estimates. In *2021 IEEE 33rd International Conference on Tools with Artificial Intelligence (ICTAI)*, pages 1279–1286. IEEE, 2021.
- [40] Kira Maag and Matthias Rottmann. False negative reduction in semantic segmentation under domain shift using depth estimation. *Proceedings of the 18th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISIGRAPP)*, 2023.
- [41] Kira Maag, Matthias Rottmann, and Hanno Gottschalk. Time-dynamic estimates of the reliability of deep semantic segmentation networks. In *2020 IEEE 32nd International Conference on Tools with Artificial Intelligence (ICTAI)*, pages 502–509. IEEE, 2020.
- [42] Kira Maag, Matthias Rottmann, Serin Varghese, Fabian Hüger, Peter Schlicht, and Hanno Gottschalk. Improving video instance segmentation by light-weight temporal uncertainty estimates. In *2021 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE, 2021.
- [43] 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 workshops*, pages 328–329, 2020.
- [44] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake Vanderplas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot, and Édouard Duchesnay. Scikit-learn: Machine learning in python.

- Journal of Machine Learning Research*, 12(85):2825–2830, 2011. URL <http://jmlr.org/papers/v12/pedregosalla.html>.
- [45] Tobias Riedlinger, Matthias Rottmann, Marius Schubert, and Hanno Gottschalk. Gradient-based quantification of epistemic uncertainty for deep object detectors. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 3921–3931, 2023.
- [46] Tobias Riedlinger, Marius Schubert, Sarina Penquitt, Jan-Marcel Kezmann, Pascal Colling, Karsten Kahl, Lutz Roesse-Koerner, Michael Arnold, Urs Zimmermann, and Matthias Rottmann. Lmd: Light-weight prediction quality estimation for object detection in lidar point clouds. In *DAGM German Conference on Pattern Recognition*, pages 85–99. Springer, 2023.
- [47] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In Nassir Navab, Joachim Hornegger, William M. Wells, and Alejandro F. Frangi, editors, *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*, pages 234–241, Cham, 2015. Springer International Publishing. ISBN 978-3-319-24574-4.
- [48] Julia Rosenzweig, Eduardo Brito, Hans-Ulrich Kobialka, Maram Akila, Nico M Schmidt, Peter Schlicht, Jan David Schneider, Fabian Hüger, Matthias Rottmann, Sebastian Houben, et al. Validation of simulation-based testing: Bypassing domain shift with label-to-image synthesis. In *2021 IEEE Intelligent Vehicles Symposium Workshops (IV Workshops)*, pages 182–189. IEEE, 2021.
- [49] Matthias Rottmann and Marco Reese. Automated detection of label errors in semantic segmentation datasets via deep learning and uncertainty quantification. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 3214–3223, 2023.
- [50] 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 Workshops*, pages 0–0, 2019.
- [51] 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 International Joint Conference on Neural Networks (IJCNN)*, pages 1–9. IEEE, 2020.
- [52] Matthias Rottmann, Kira Maag, Robin Chan, Fabian Hüger, Peter Schlicht, and Hanno Gottschalk. Detection of false positive and false negative samples in semantic segmentation. In *2020 Design, Automation & Test in Europe Conference & Exhibition (DATE)*, pages 1351–1356. IEEE, 2020.
- [53] Jörg Sander, Bob D. de Vos, Jelmer M. Wolterink, and Ivana Išgum. Towards increased trustworthiness of deep learning segmentation methods on cardiac mri. In Elsa D. Angelini and Bennett A. Landman, editors, *Medical Imaging 2019: Image Processing*. SPIE, March 2019. doi: 10.1117/12.2511699. URL <http://dx.doi.org/10.1117/12.2511699>.

- [54] Marius Schubert, Karsten Kahl, and Matthias Rottmann. Metadetect: Uncertainty quantification and prediction quality estimates for object detection. In *2021 International Joint Conference on Neural Networks (IJCNN)*, pages 1–10. IEEE, 2021.
- [55] Arnaud Arindra Adiyoso Setio, Francesco Ciompi, Geert Litjens, Paul Gerke, Colin Jacobs, Sarah J. van Riel, Mathilde Marie Winkler Wille, Matiullah Naqibullah, Clara I. Sánchez, and Bram van Ginneken. Pulmonary nodule detection in ct images: False positive reduction using multi-view convolutional networks. *IEEE Transactions on Medical Imaging*, 35(5):1160–1169, 2016. doi: 10.1109/TMI.2016.2536809.
- [56] Fei Shan, Yaozong Gao, Jun Wang, Weiya Shi, Nannan Shi, Miaofei Han, Zhong Xue, Dinggang Shen, and Yuxin Shi. Abnormal lung quantification in chest ct images of covid-19 patients with deep learning and its application to severity prediction. *Medical Physics*, 48(4):1633–1645, 2021. doi: <https://doi.org/10.1002/mp.14609>. URL <https://aapm.onlinelibrary.wiley.com/doi/abs/10.1002/mp.14609>.
- [57] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15(56):1929–1958, 2014. URL <http://jmlr.org/papers/v15/srivastava14a.html>.
- [58] Svenja Uhlemeyer, Matthias Rottmann, and Hanno Gottschalk. Towards unsupervised open world semantic segmentation. In *Uncertainty in Artificial Intelligence*, pages 1981–1991. PMLR, 2022.
- [59] C.J. Van Rijsbergen. *Information Retrieval*. Butterworths, 1979. ISBN 9780408709293.
- [60] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. Graph attention networks. *arXiv preprint arXiv:1710.10903*, 2017.
- [61] Zifeng Wu, Chunhua Shen, and Anton Van Den Hengel. Wider or deeper: Revisiting the resnet model for visual recognition. *Pattern recognition*, 90:119–133, 2019.
- [62] Yi Xiao, Felipe Codevilla, Akhil Gurram, Onay Urfalioglu, and Antonio M López. Multimodal end-to-end autonomous driving. *IEEE Transactions on Intelligent Transportation Systems*, 23(1):537–547, 2020.
- [63] 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*, pages 8856–8865, 2019.