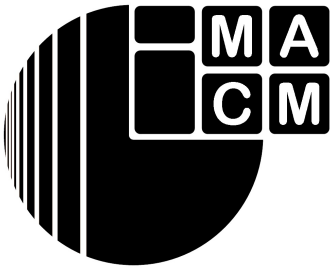


Bergische Universität Wuppertal

Fachbereich Mathematik und Naturwissenschaften

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

Preprint BUW-IMACM 19/29

K. Maag, M. Rottmann and H. Gottschalk

Time-Dynamic Estimates of the Reliability of Deep Semantic Segmentation Networks

November 25, 2019

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

Time-Dynamic Estimates of the Reliability of Deep Semantic Segmentation Networks

Kira Maag[1] and **Matthias Rottmann**[1] and **Hanno Gottschalk**[1]

Abstract. In the semantic segmentation of street scenes, the reliability of a prediction is of highest interest. The assessment of neural networks by means of uncertainties is a common ansatz to prevent safety issues. As in online applications like automated driving, a video stream of images is available, we present a time-dynamical approach to investigate uncertainties and assess the prediction quality of neural networks. To this end, we track segments over time and gather aggregated metrics per segment, e.g. mean dispersion metrics derived from the softmax output and segment sizes. Due to identifying segments over consecutive frames, we obtain time series of metrics from which we assess prediction quality. We do so by either classifying between intersection over union $(IoU) = 0$ and $IoU > 0$ (*meta classification*) or predicting the IoU directly (*meta regression*). In our tests, we analyze the influence of the length of the time series on the predictive power of metrics and study different models for meta classification and regression. We use two publicly available DeepLabv3+ networks as well as two street scene datasets, i.e., VIPER as a synthetic one and KITTI based on real data. We achieve classification accuracies of up to 81.20% and AUROC values of up to 88.68% for the task of meta classification. For meta regression we obtain R^2 values of up to 87.51%. We show that these results yield improvements compared to other approaches.

1 Introduction

Semantic segmentation, i.e., the pixel-wise classification image content, is an important tool for scene understanding. In recent years, neural networks have demonstrated outstanding performance for this task. In safety relevant applications like automated driving [26] and medical imaging [48], the reliability of predictions and thus uncertainty quantification is of highest interest. While most works focus on uncertainty quantification for single frames, there is often video data available. In this work, we investigate uncertainties taking the temporal information into account. To this end, we track objects over time and construct metrics that express the model's uncertainty.

Uncertainty measures. A very important type of uncertainty is the model uncertainty resulting from the fact that the ideal parameters are unknown and have to be estimated from data. Bayesian models are one possibility to consider these uncertainties [35]. Therefore, different frameworks based on variational approximations for Bayesian inference exist [3, 16]. Recently, Monte-Carlo (MC) Dropout [19] as approximation to Bayesian inference has aroused a lot of interest. In classification tasks, the uncertainty score can be directly determined on the network's output [19]. Threshold values for the highest softmax probability or threshold values for the entropy of the classification distributions (softmax output) are common approaches for the detection of false predictions (false positive) of neural networks, see e.g. [24, 34]. Uncertainty metrics like classification entropy or the highest softmax probability are usually combined with model uncertainty (MC Dropout inference) or input uncertainty, cf. [19] and [34], respectively. Alternatively, gradient-based uncertainty metrics are proposed in [37] and an alternative to Bayesian neural networks is introduced in [32] where the idea of ensemble learning is used to consider uncertainties. These uncertainty measures have proven to be practically efficient for detecting uncertainty and some of them have also been transferred to semantic segmentation tasks, such as MC Dropout, which also achieves performance improvements in terms of segmentation accuracy, see [30]. The works presented in [29] and [48] also make use of MC Dropout to model the uncertainty and filter out predictions with low reliability. This line of research is further developed in [26] to detect spacial and temporal uncertainty in the semantic segmentation of videos. In semantic segmentation tasks the concept of *meta classification* and *meta regression* is introduced in [39]. Meta classification refers to the task of predicting whether a predicted segment intersects with the ground truth or not. Therefore, the intersection over union (IoU, also known as Jaccard index [27]), a commonly used performance measure for semantic segmentation, is considered. The IoU quantifies the degree of overlap of prediction and ground truth, it is equal to zero if and only if the predicted segment does not intersect with the ground truth. The meta-classification task corresponds to (meta-)classifying between $IoU = 0$ and $IoU > 0$ for every predicted segment. Meta regression is the task of predicting the IoU (e.g. via linear regression) directly. The main aim of both tasks is to have a model that is able to reliably assess the quality of a semantic segmentation obtained from a neural network. The predicted IoU therefore also serves as a performance estimate. As input both methods use segment-wise metrics extracted from the segmentation network's softmax output. The same tasks are pursued in [14, 25] for images containing only a single object, instead of metrics they utilize additional CNNs. In [40] the work of [39] is extended by adding resolution dependent uncertainty and further metrics. In [17] performance measures for the segmentation of videos are introduced, these measures are also based on image statistics and can be calculated without ground truth.

Visual object tracking. Object tracking is an essential task in video applications, such as automated driving, robot navigation and many others. The tasks of object tracking consist of detecting the objects and then tracking them in consecutive frames, eventually studying their behavior [54]. In most works, the target object

[1] University of Wuppertal, School of Mathematics and Natural Sciences, Germany, email: {kmaag, rottmann, hgottsch}@uni-wuppertal.de

is represented as an axis-aligned bounding box [49] or rotated bounding box [31]. Labeling objects with bounding boxes keeps annotation costs low and allows a fast and simple initialization of the target object. The approaches described in the following work with bounding boxes. A popular strategy for object tracking is the tracking-by-detection approach [4]. A discriminative classifier is trained online while performing the tracking, to separate the object from the background only by means of the information where the object is located in the first frame. Another approach for tracking-by-detection uses adaptive correlation filters that model the targets appearance, the tracking is then performed via convolution with the filters [7]. In [13] and [46], the trackers based on correlation filters are improved with spatial constraints and deep features, respectively. Another object tracking algorithm [36] combines Kalman filters and adaptive least squares to predict occluded objects where the detector shows deficits. In contrast to online learning, there are also tracking algorithms that learn the tracking task offline and perform tracking as inference, only. These methods differ greatly from the tracking-by-detection procedure. The idea behind these approaches [23, 6] is to train offline a similarity function on pairs of video frames instead of training a discriminative classifier online. In [6] a fully-convolutional siamese network is used and this approach is improved by making use of region proposals ([33]), angle estimation and spatial masking ([22]) as well as memory networks ([51]). Another approach for object tracking with bounding boxes is presented in [52] where semantic information is used for tracking. Most algorithms and also the ones described here use bounding boxes, mostly for initializing and predicting the position of an object in the subsequent frames. In contrast, [11] uses coarse binary masks of target objects instead of rectangles. There are other procedures that initialize and/or track an object without bounding boxes, since a rectangular box does not necessarily capture the shape of every object well. In [28] a temporal quad-tree algorithm is applied, where the objects are divided into squares getting smaller and smaller. Other approaches use semantic image segmentation such as [21], where the initialization includes a segmentation for predicting object boundaries. Segmentation-based tracking algorithms are presented in [2] and [15] based on a pixel-level probability model and an adaptive model, respectively. In the latter case, co-training takes place between detector and segmentation. The approaches presented in [5] and [43] are also based on segmentation and use particle filters for the tracking process. There are also a superpixel-based approaches, see e.g. [53], and a fully-convolutional siamese approach [47] that creates binary masks and starts from a bounding box initialization.

Our contribution. In this work we elaborate on the meta classification and regression approach from [39] that provides a framework for post processing a semantic segmentation. This method generates uncertainty heat maps from the softmax output of the semantic segmentation network, such as pixel-wise entropy, probability margin or variation ratio. In fig. 1 a visualization of the segment-wise variation ratio is given. In addition to these segment-wise metrics, further quantities derived from the predicted segments are used, for instance various measures corresponding to the segments geometry. This set of metrics, yielding a structured dataset where each row corresponds to a predicted segment, is presented to meta classifier/regressor to either classify between $IoU = 0$ and $IoU > 0$ or predict the IoU directly. In contrast to [39] we use the additional metrics proposed in [40]. In this paper, we extend the work presented in [39] by taking time-dynamics into account. A core assumption is that a semantic segmentation network and a video stream of input data are available. We present a light-weight approach that tracks semantic segments over time. The segments are matched according to their overlap in multiple frames and we improve these measures due to shifting segments according to expected location in the next frame. We gather time series of metrics that are presented as input to meta classifiers and regressors. For the latter we study different types of models and their dependence on the length of the time series.

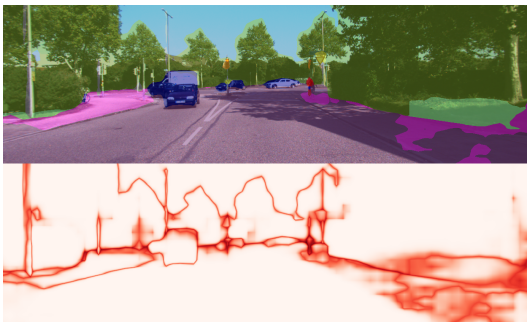


Figure 1. Segmentation predicted by a neural network (top) and heat map V_z (bottom).

In our tests, we employ two publicly available DeepLabv3+ networks [9] and we perform all tests on the VIsual PERception (VIPER) dataset [38] as well as on the KITTI dataset [20]. For the synthetic VIPER dataset we train a DeepLabv3+ network and demonstrate that the additional information from our time-dynamical approach improves over its single frame counterpart w.r.t. meta classification and regression (meta tasks). Furthermore, the different methods for classification and regression improve the prediction accuracy of the $IoU = 0$. For the task of meta regression we obtain an R^2 value of up to 85.82% and for the meta classification AUROC values of up to 86.01% as well as classification accuracies of up to 77.88%. For the VIPER dataset there are labeled ground truth images for each frame, while for the KITTI dataset only a few frames are labeled with ground truth. For the KITTI datset we use alternative sources of useful information besides the real ground truth to train the meta tasks and we employ both networks for investigations. For meta regression we achieve R^2 values of up to 87.51% and for the meta classification AUROC values of up to 88.68% as well as a classification accuracies of up to 81.20%. We also show that these results yield significant improvements compared to the results obtained by the predecessor method introduced in [39].

Related work. Most works [49, 31, 4, 7, 13, 46, 36, 50, 23, 6, 33, 22, 51, 52] in the field of object tracking make use of bounding boxes while our approach is based on semantic segmentation. There are some approaches that make use of segmentation masks. However, only a coarse binary mask is used in [11] and in [21] the segmentation is only used for initialization. In [2, 15] not only information of the semantic segmentations are included in the tracking algorithm, but the segmentation and the tracking are executed depending on each other. In our procedures, a segmentation is inferred first, tracking is performed afterwards. In addition to the different forms of object representations, there are various algorithms for object tracking. In the tracking-by-detection methods a classifier for the difference between object and background is trained and therefore only information about the location of the object in the first frame is given [4, 7, 13, 46]. We do not train classifiers as this information is contained in the inferred segmentations. Another

approach is to learn a similarity function offline [23, 6, 33, 22, 51]. The works of [2, 15, 5, 43, 47] are based on segmentation and they use different tracking methods, like probability models, particle filters and fully-convolutional siamese network, respectively. Our algorithm is solely based on the degree of overlap of predicted segments.

With respect to uncertainty quantification, MC dropout – many forward passes under dropout at inference time – is widely used, c.f. [19, 30, 29, 48]. Whenever dropout is used in a segmentation network, the resulting heat-map can be equipped by our framework. However, in this work we do not include MC dropout. There are alternative measures of uncertainty like gradient based ones [37] or measures based on spatial and temporal differences between the colors and movements of the objects [17]. We construct metrics based on aggregated dispersion measures from the softmax output of a neural network at segment level. The works [14, 25] closest to ours are constructed to work with one object per image, instead on hand crafted metrics they are based on post-processing CNNs. We extend the work of [39] by a temporal component and further investigate methods for the meta classification and regression, e.g. gradient boosting and neural networks.

Outline. The remainder of this work is organized as follows. In section 2 we introduce a tracking algorithm for semantic segmentation. This is followed by the construction of segment-wise metrics using uncertainty and geometry information in section 3. In section 4 we describe the meta regression and classification methods including the construction of their inputs consisting of time series of metrics. Finally, we present numerical results in section 5. We study the influence of time-dynamics on meta classification and regression as well as the incorporation of various classification and regression methods.

2 Tracking Segments over Time

In this section we introduce a light-weight tracking method for the case where a semantic segmentation is available in each frame of a video. Semantic image segmentation aims at segmenting objects in an image, to this end it can be defined as a pixel-wise classification of image content (cf. top panel of fig. 1). To obtain a semantic segmentation, the goal is to assign to each image pixel z of an input image x a label y within a prescribed label space $\mathcal{C} = \{y_1, \dots, y_c\}$. Here, this task is performed by a neural networks that provides for each pixel z a probability distribution $f_z(y|x, w)$ over the class labels $y \in \mathcal{C}$, given learned weights w and an input image x. The predicted class for each pixel z is obtained by

$$\hat{y}_z(x, w) = \text{argmax}_{y \in \mathcal{C}} f_z(y|x, w)\,. \tag{1}$$

Let $\hat{\mathcal{S}}_x = \{\hat{y}_z(x, w) | z \in x\}$ denote the predicted segmentation and $\hat{\mathcal{K}}_x$ the set of predicted segments. The idea of our algorithm is to match segments of the same class according to their overlap in consecutive frames. So we denote by $\{x_1, \dots, x_T\}$ the image sequence with a length of T and x_t corresponds to the t^{th} image. Furthermore, we formulate the overlap of a segment k with a segment j through

$$O_{j,k} = \frac{|j \cap k|}{|j|}. \tag{2}$$

To account for the motion of objects, we also register geometric centers of predicted segments. The geometric center of a segment $k \in \hat{\mathcal{K}}_{x_t}$ in frame t is defined as

$$\bar{k}_t = \frac{1}{|k|} \sum_{z \in k} z \tag{3}$$

where $z = (z_1, z_2)$ is given by its vertical and horizontal coordinates of pixel z.

Our tracking algorithm is applied sequentially to each frame t, $t = 1, \dots, T$, and we aim at tracking all segments present in at least one of the frames. To give the segments different priorities for matching, the segments of each frame are sorted by size and treated in descending order. As is the case when a segment in frame t has been matched with a segment from previous frames, it is ignored in further steps and matched segments are assigned an id. Within the description of the matching procedure, we introduce parameters c_{near}, c_{over}, c_{dist} and c_{lin}, the respective numerical choices are given in section 5. More formally, our algorithm consists of the following five steps:

Step 1 (aggregation of segments). The minimum distance between segment $i \in \hat{\mathcal{K}}_{x_t}$ and all $j \in \hat{\mathcal{K}}_{x_t} \setminus \{i\}$ of the same class is calculated. If the distance is less than a constant c_{near}, the segments are so close to each other that they are regarded as one segment and receive a common id.

Step 2 (shift). If the algorithm was applied to at least two previous frames, the geometric centers $(\bar{k}_{t-2})$ and $(\bar{k}_{t-1})$ of segment $k \in \hat{\mathcal{K}}_{x_{t-1}}$ are computed. The segment from frame $t-1$ is shifted by the vector $(\bar{k}_{t-1} - \bar{k}_{t-2})$ and the overlap $O_{j,k}$ with each segment $j \in \hat{\mathcal{K}}_{x_t}$ from frame t is determined. If $O_{j,k} \geq c_{over}$ or $j = \text{argmax}_{i \in \hat{\mathcal{K}}_{x_t}} O_{i,k}$, the segments k and j are matched and receive the same id. If there is no match found for segment k during this procedure, the quantity

$$\begin{aligned} d = \min_{j \in \hat{\mathcal{K}}_{x_t}} & \left\| \bar{j}_t - \bar{k}_{t-1} \right\|_2 \\ & + \left\| (\bar{k}_{t-1} - \bar{k}_{t-2}) - (\bar{j}_t - \bar{k}_{t-1}) \right\|_2 \end{aligned} \tag{4}$$

is calculated for each available j and both segments are matched if $d \leq c_{dist}$. This allows for matching segments that are closer to $\bar{k}_{t-1}$ than expected. If segment k exists in frame $t-1$, but not in $t-2$, then step 2 is simplified: only the distance between the geometric center of $k \in \hat{\mathcal{K}}_{x_{t-1}}$ and $j \in \hat{\mathcal{K}}_{x_t}$ is computed and the segments are matched if the distance is smaller than c_{dist}.

Step 3 (overlap). If $t \geq 2$, The overlap $O_{j,k}$ of the segments $k \in \hat{\mathcal{K}}_{x_{t-1}}$ and $j \in \hat{\mathcal{K}}_{x_t}$ of two consecutive frames is calculated. If $O_{j,k} \geq c_{over}$ or $j = \text{argmax}_{i \in \hat{\mathcal{K}}_{x_t}} O_{i,k}$, the segments k and j are matched.

Step 4 (regression). In order to account for flashing predicted segments, either due to false prediction or occlusion, we implement a linear regression and match segments that are more than one, but at most $n_{lr} - 2$, frames apart in temporal direction. If the id of segment $k \in \hat{\mathcal{K}}_{x_*}, * \in \{t - n_{lr}, \dots, t-1\}$, in frame t has not yet been assigned and $t \geq 4$, i.e., three frames have already been processed, then the geometric centers of segment k are computed in frames $t - n_{lr}$ to $t-1$ (in case k exists in all these frames). If at least two geometric centers are available, a linear regression is performed to predict the geometric center $(\hat{\bar{k}}_t)$. If the distance between the predicted geometric center and the calculated geometric center of the segment $j \in \hat{\mathcal{K}}_{x_t}$ is less than a constant value c_{lin}, k and j are matched. If no match was found for segment k, segment $k \in \hat{\mathcal{K}}_{x_{t_{max}}}$ is shifted by the vector $\left(\hat{\bar{k}}_t - \bar{k}_{t_{max}}\right)$, where $t_{max} \in \{t - n_{lr}, \dots, t-1\}$ denotes the frame where k contains the maximum number of pixels.

If $O_{j,k} \geq c_{over}$ or $j = \text{argmax}_{i \in \hat{\mathcal{K}}_{x_t}} O_{i,k}$ applies to the resulting overlap, k and j are matched.

Step 5 (new ids). All segments $j \in \hat{\mathcal{K}}_{x_t}$ that have not yet received an id are assigned with a new one.

3 Segment-wise Metrics and Time Series

In the previous chapter, we presented the semantic segmentation and the resulting probability distribution $f_z(y|x,w)$ for an image x, pixel z and weights w. The degree of randomness in $f_z(y|x,w)$ is quantified by dispersion measures. In this work, we consider three pixel-wise dispersion measures: the entropy E_z, the variation ratio V_z and the probability margin M_z. The measures are given by

$$E_z(x,w) = -\frac{1}{\log(q)} \sum_{y \in \mathcal{C}} f_z(y|x,w) \log f_z(y|x,w), \tag{5}$$

$$V_z(x,w) = 1 - f_z(\hat{y}_z(x,w)|x,w) \tag{6}$$

and

$$\begin{aligned} M_z(x,w) =& 1 - f_z(\hat{y}_z(x,w)|x,w) \\ &+ \max_{y \in \mathcal{C} \setminus \{\hat{y}_z(x,w)\}} f_z(y|x,w) \end{aligned} \tag{7}$$

and a visualization of segment-wise variation ratio is shown in fig. 1. Note that also other heat maps (like MC Dropout variance) can be processed. At segment level, we define for each $k \in \hat{\mathcal{K}}_x$ the interior $k_{in} \subset k$ where a pixel z is an element of k_{in} if all eight neighbouring pixels are an element of k, the boundary $k_{bd} = k \setminus k_{in}$ and the following metrics (see [39, 40]):

- the segment sizes $S = |k|$, $S_{in} = |k_{in}|$, $S_{bd} = |k_{bd}|$
- the mean dispersions $\bar{D}$, $\bar{D}_{in}$, $\bar{D}_{bd}$ defined as

$$\bar{D}_\sharp(k) = \frac{1}{S_\sharp} \sum_{z \in k_\sharp} D_z(x), \qquad \sharp \in \{\_, in, bd\}$$

  where $D_z \in \{E_z, V_z, M_z\}$
- the relative segment sizes $\tilde{S} = S/S_{bd}$, $\tilde{S}_{in} = S_{in}/S_{bd}$
- the relative mean dispersions $\tilde{\bar{D}} = \bar{D}\tilde{S}$, $\tilde{\bar{D}}_{in} = \bar{D}_{in}\tilde{S}_{in}$ where $D \in \{E, V, M\}$
- the geometric center $\bar{k}$ defined in equation (3)
- the mean class probabilities for each class $y \in \{1, \ldots, c\}$

$$P(y|k) = \frac{1}{S} \sum_{z \in k} f_z(y|x,w)$$

Additionally we define the set of metrics by

$$\begin{aligned} U^k = \{&S, S_{in}, S_{bd}, \tilde{S}, \tilde{S}_{in}, \bar{D}, \bar{D}_{in}, \bar{D}_{bd}, \tilde{\bar{D}}, \tilde{\bar{D}}_{in}| \\ &D \in \{E, V, M\}\} \cup \{\bar{k}\} \cup \{P(y|k) \,:\, y = 1, \ldots, c\}. \end{aligned}$$

The separate treatment of interior and boundary in all dispersion measures is motivated by typically large values of D_z for $z \in k_{bd}$. In addition, we find that poor or false predictions are often accompanied by fractal segment shapes (which have a relatively large amount of boundary pixels, measurable by $\tilde{S} = S/S_{bd}$ and $\tilde{S}_{in} = S_{in}/S_{bd}$) and/or high dispersions $\bar{D}_{in}$ on the segment's interior.

Table 1. Overview of meta classification and regression methods. For classification we consider **LR L1**, **GB**, **NN L2** and for regression all of them.

methods	
LR	linear regression
LR L1	linear / logistic regression with ℓ_1-penalization
LR L2	linear regression with ℓ_2-penalization
GB	gradient boosting
NN L1	neural network with ℓ_1-penalization
NN L2	neural network with ℓ_2-penalization

4 Prediction of IoU from Time Series

A measure to determine the prediction accuracy of the segmentation network with respect to the ground truth is the IoU. Therefore we define $\mathcal{K}_x$ the set of connected components in the ground truth $\mathcal{S}_x$, analogously to $\hat{\mathcal{K}}_x$ (the set of connected components in the predicted segmentation $\hat{\mathcal{S}}_x$). The corresponding class of $k' \in \mathcal{K}_x$ is denoted by $y_z(x) \in \mathcal{C}$ (for any $z \in k'$). For $k \in \hat{\mathcal{K}}_x$ let $\mathcal{K}_x^k = \{k' \in \mathcal{K}_x : \hat{y}_z(x,w) = y_z(x)$ for $z \in k \cap k'$ and $k \cap k' \neq \emptyset\}$. For each $k \in \hat{\mathcal{K}}_x$ the IoU is given by

$$IoU(k) = \frac{|k \cap K'|}{|k \cup K'|}, \qquad K' = \bigcup_{k' \in \mathcal{K}_x^k} k'. \tag{8}$$

In our test we use a slight modification, i.e., the adjusted IoU

$$IoU_{\text{adj}}(k) = \frac{|k \cap K'|}{|k \cup (K' \setminus Q)|}, \qquad Q = \bigcup_{q \in \mathcal{Q}_x^k} q. \tag{9}$$

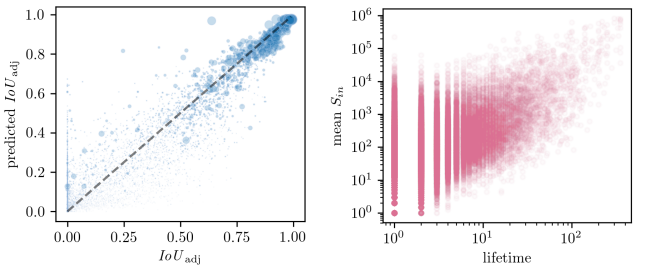
with $\mathcal{Q}_x^k = \{q \in \hat{\mathcal{K}}_x : \hat{y}_z(x,w) = y_z(x),\ z \in q \cap K' \neq \emptyset\}$ proposed in [39]. In this work, we make segment-wise predictions of the IoU_{adj} via different regression approaches and classify between $IoU_{\text{adj}} = 0$ and $IoU_{\text{adj}} > 0$ (meta classification) for every predicted segment. These prediction tasks are performed by means of the metrics introduced in section 3. Note that these metrics can be computed without the knowledge of the ground truth. Our aim is to analyze to which extent they are suitable for the meta tasks and what influence the temporal information has. For each segment $k \in \hat{\mathcal{K}}_{x_t}$ in frame t we have the metrics U_t^k and further measures from the previous frames due to the segment matching. For meta regression and classification we make use of these metrics U_i^k, $i = t - n_\tau, \ldots, t$, where n_τ describes the number of considered frames. For meta classification we define $y_k = \{0$ if $IoU_{\text{adj}}(k) = 0, 1$ if $IoU_{\text{adj}}(k) > 0\}$ and we want to predict this value by three different methods. One method used is the least absolute shrinkage and selection operator (LASSO [44]) method. The LASSO method makes use of ℓ_1-penalization and investigates the predictive power of different combinations of input variables. Here, we study the influence of the various metrics of previous frames. The two other methods we apply to meta classification are gradient boosting [18] and a shallow neural network, which contains only one hidden layer with 50 neurons. For the meta regression we follow a similar procedure and compare six methods for this task. The procedure includes simple linear regression, as well as linear regression with ℓ_1- and ℓ_2-penalization. Furthermore we use gradient boosting and shallow neural networks. One net with ℓ_1-penalization and another one with ℓ_2. An overview of the different methods for regression and classification is given in table 1.

5 Numerical Results

In this section, we investigate the properties of the metrics defined in the previous sections, the influence of the length of the time series considered and of different meta classification and regression

Table 2. Results for meta classification and regression for the different methods. The super script denotes the number of frames where the best performance and in particular the given values are reached. The best classification and regression results are highlighted.

Meta Classification $IoU_{\text{adj}} = 0, > 0$					
LR L1		GB		NN L2	
ACC	AUROC	ACC	AUROC	ACC	AUROC
$75.75\%(\pm 0.49\%)^{8}$	$83.44\%(\pm 0.47\%)^{7}$	$\mathbf{77.88\%}(\pm 0.60\%)^{2}$	$\mathbf{86.01\%}(\pm 0.56\%)^{4}$	$76.62\%(\pm 0.51\%)^{6}$	$84.52\%(\pm 0.50\%)^{11}$
Meta Regression IoU_{adj}					
LR		LR L1		LR L2	
σ	R^2	σ	R^2	σ	R^2
$0.124(\pm 0.002)^{6}$	$82.58\%(\pm 0.45\%)^{6}$	$0.124(\pm 0.002)^{7}$	$82.56\%(\pm 0.43\%)^{7}$	$0.124(\pm 0.002)^{5}$	$82.57\%(\pm 0.44\%)^{5}$
GB		NN L1		NN L2	
σ	R^2	σ	R^2	σ	R^2
$\mathbf{0.112}(\pm 0.002)^{6}$	$\mathbf{85.82\%}(\pm 0.36\%)^{6}$	$0.118(\pm 0.002)^{4}$	$84.36\%(\pm 0.51\%)^{4}$	$0.117(\pm 0.002)^{2}$	$84.58\%(\pm 0.44\%)^{2}$



(a) (b)

Figure 2. (a): Predicted IoU_{adj} vs. IoU_{adj} for all non-empty segments. The dot size is proportional to the segment size. (b): Segment lifetime (time series length) vs. mean interior segment size, both on log scale.

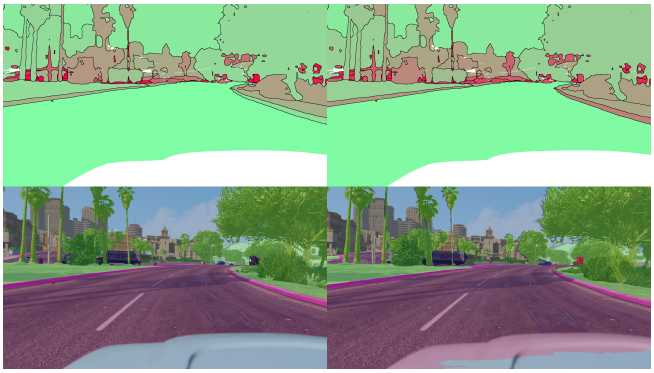


Figure 3. Ground truth image (bottom left), prediction obtained by a neural network (bottom right), a visualization of the true segment-wise IoU_{adj} of prediction and ground truth (top left) and its prediction obtained from meta regression (top right). Green color corresponds to high IoU_{adj} values and red color to low ones. For the white regions there is no ground truth available, these regions are not included in the statistical evaluation.

methods. We perform our tests on two different datasets for the semantic segmentation of street scenes where also videos are available, the synthetic VIPER dataset [38] obtained from the computer game GTA V and the KITTI dataset [20] with real street scene images from Karlsruhe, Germany. In all our tests we consider two different DeepLabv3+ networks [9] for semantic segmentation for which we use a reference implementation in Tensorflow [1]. The DeepLabv3+ implementation and weights are available for two network backbones. First, there is the Xception65 network, a modified version of Xception [10] and it is a powerful structure for server-side deployment. On the other hand, is MobilenetV2 [42] a fast structure designed for mobile devices. Primarily we use Xception65 for VIPER and MobilenetV2 for KITTI, for the latter we also use Xception65 as a reference network to generate pseudo ground truth for the meta classification and regression tasks. For tests with KITTI we used the publicly available weights for both networks.

For tracking segments with our procedure, we assign the parameters defined in section 2 with the following values: $c_{near} = 10$, $c_{over} = 0.35$, $c_{dist} = 100$ and $c_{lin} = 50$. We study the predictive power of our 22 metrics and segment-wise averaged class probabilities per segment and frame. From our tracking algorithm we get these metrics additionally from previous frames for every segment.

VIPER dataset. The VIPER dataset consists of more than 250K high-resolution $1{,}920 \times 1{,}080$ video frames and for all frames there is ground truth available for 23 classes. We trained an Xception65 network starting from the weights for ImageNet [41]. We choose an output stride of 16 and the input image is evaluated within the framework only on its original scale (deeplab allows for evaluation on different scales and averaging the results). For a detailed explanation of the chosen parameters we refer to [9]. We retrain the Xception65 net on the VIPER dataset on 5,147 training images and 847 validation images. We only use images from the day category (i.e., bright images, no rain) for training and further processing. We achieve a mean IoU of 50.33%. If we take out the classes with a mean IoU below 10%, the total mean IoU rises to 57.38%. This case applies to the three classes mobile barrier, chair and van, classes that are also underrepresented in the dataset. For meta classification and regression we use only 13 video sequences consisting of 3,593 images in total. From these images we obtain roughly 309,874 segments (not yet matched over time) of which 251,368 have non-empty interior. The latter are used in all numerical tests. We investigate the influence of time-dynamics on meta classification and regression, i.e., we firstly only present the segment-wise metrics U_t^k of a single frame t into the meta classifier/regressor, secondly we extend the metrics to time series with a length of up to 10 previous time steps U_i^k, $i = t-10, \ldots, t-1$. In summary, we obtain 11 different inputs for the meta classification and regression models. The presented results are averaged over 10 runs obtained by random sampling of the train/validation/test splitting. In tables and figures, the corresponding standard deviations are given in brackets and by shades, respectively. Out of the 251,368 segments with non-empty interior, 85,291 have an $IoU_{\text{adj}} = 0$. We start with the detection of the segments with $IoU_{\text{adj}} = 0$, i.e., we perform meta classification to detect false positive segments. To this extent, we use 38,000 segments that are not presented to the segmentation network during training and apply a train/validation/test splitting of 70%/10%/20%. To evaluate the performance of different models for meta classification we consider classification accuracy and AUROC values. The AUROC is obtained by varying the decision threshold

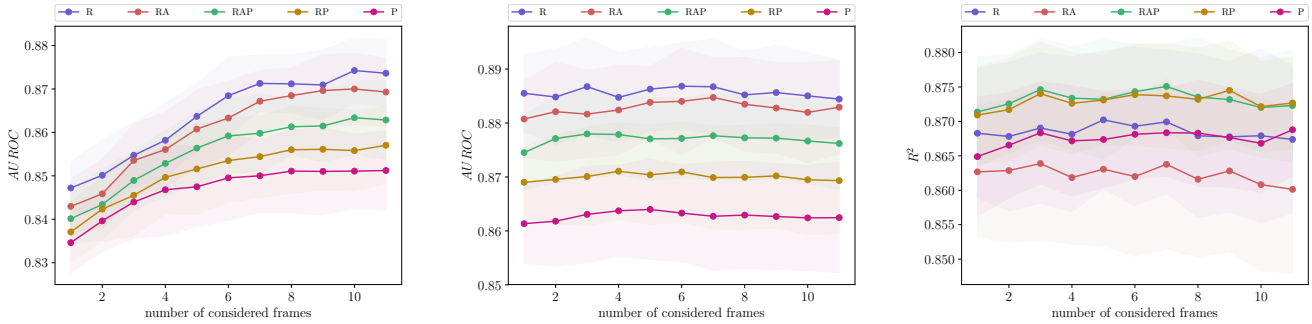


Figure 4. A selection of results for meta classification AUROC and regression R^2 as functions of the number of frames and for different compositions of training data (cf. table 3). (a): meta classification via a neural network with ℓ_2-penalization, (b): meta classification via gradient boosting, (c): meta regression via gradient boosting.

in a binary classification problem, here for the decision between $IoU = 0$ and $IoU > 0$. We achieve test AUROC values of up to $86.01\%(\pm 0.56\%)$ and accuracies of up to $77.88\%(\pm 0.60\%)$. Table 2 shows the best results for three different meta classification methods, i.e., logistic regression, a neural network and gradient boosting, cf. table 1. The super script denotes the number of frames where the best performance and in particular the given values are reached. On the one hand, we observe that the best results are achieved when considering more than one frame. On the other hand, significant differences between the methods for meta classification can be observed, gradient boosting shows the best performance with respect to classification accuracy and AUROC.

In the next step, we predict IoU_{adj} values via meta regression to get an uncertainty measure. For this task we indicate resulting standard deviations σ and R^2 values. We achieve R^2 values of up to $85.82\%(\pm 0.36\%)$. This value is obtained by gradient boosting incorporating 5 previous frames. For this particular study, the relationship between the calculated and predicted IoU_{adj} is shown in fig. 2 (a), an illustration of the resulting uncertainty measure is given in fig. 3. We also provide video sequences that visualize the IoU_{adj} prediction and the segment tracking, see `https://youtu.be/TQaV5ONCV-Y`. Result for meta regression are also summarized in table 2, the findings are in analogy to those for meta classification. Gradient boosting performs best, and more frames yield better results than a single one. Figure 2 (b) depicts the time series length vs. the mean interior segment size. On average, a predicted segment exists for 4.4 frames, however when we consider only segments that contain at least 1,000 interior pixels, the average life time increases to 19.9 frames.

KITTI dataset. For the KITTI dataset, we use both DeepLabv3+ networks (pre-trained on the Cityscapes dataset [12], available on GitHub). As parameters for the Xception65 network we choose an output stride of 8, a decoder output stride of 4 and an evaluation of the input on scales of 0.75, 1.00 and 1.25 (averaging the results). For the MobilenetV2 we use an output stride of 16 and the input image is evaluated within the framework only on its original scale. We use both nets to generate the output probabilities on the KITTI dataset. In our tests we use 29 street scene videos consisting of 12,223 images with a resolution of 1392×512. Of these images, only 142 are labelled. An evaluation of meta regression and classification requires a train/validation/test splitting. Therefore, the small number of labeled images seems almost insufficient. Hence,

Table 3. Train/val/test splitting, different compositions of training data and their approximate number of segments.

splitting	types of data / annotation		no. of segments
train	R	real	∼ 3,400
	RA	real and augmented	∼ 27,000
	RAP	real, augmented and pseudo	∼ 27,000
	RP	real and pseudo	∼ 27,000
	P	pseudo	∼ 27,000
val		real	∼ 500
test		real	∼ 1,000

we acquire alternative sources of useful information besides the (real) ground truth. First, we utilize the Xception65 net with high predictive performance, its predicted segmentations we term *pseudo ground truth*. We generate pseudo ground truth for all images where no ground truth is available. The mean IoU performance of the Xception65 net for the 142 labelled images is roughly 65% (and for the MobilenetV2 the mean IoU is about 50%). In addition, to augment the structured dataset of metrics, we apply a variant of SMOTE for continuous target variables for data augmentation (see [8, 45]). An overview of the different compositions of training data and the train/val/test splitting are given in table 3. The train/val/test splitting of the data with ground truth available is the same as for the VIPER dataset, i.e., 70%/10%/20%. The shorthand "augmented" refers to data obtained from SMOTE, "pseudo" refers to pseudo ground truth obtained from the Xception65 net and "real" refers to ground truth obtained from a human annotator. These additions are only used during training. We utilize the Xception65 network only for the generation of pseudo ground truth, all tests are performed using the MobilenetV2. The KITTI dataset consists of 19 classes (4 classes less than VIPER), thus we have 41 metrics in total.

From the 12,223 chosen images, we obtain 452,287 segments of which 378,984 have non-empty interior. Of these segments, 129,033 have an $IoU_{\text{adj}} = 0$. A selection of results for meta classification AUROC and regression R^2 as functions of the number of frames, i.e., the maximum time series length, is given in fig. 4. The meta classification results for neural networks presented in subfigure (a) indeed show, that an increasing length of time series has a positive effect on meta classification. On the other hand, the results in subfigure (b) show that gradient boosting does not benefit as much from time series. In both cases augmentation and pseudo ground truth do not improve the models' performance on the test set and although the neural network benefits a lot from time series, its best performance is still about 1% below that of gradient boosting. With respect

Table 4. Results for meta classification and regression for different compositions of training data and methods. The super script denotes the number of frames where the best performance and thus the given value is reached. The best results for each data composition are highlighted.

Meta Classification $IoU_{\text{adj}} = 0, > 0$						
	LR L1		GB		NN L2	
	ACC	AUROC	ACC	AUROC	ACC	AUROC
R	**76.69%**$(\pm 1.68\%)^{10}$	$85.13\%(\pm 0.84\%)^{1}$	**81.20%**$(\pm 1.02\%)^{4}$	**88.68%**$(\pm 0.80\%)^{6}$	**79.67%**$(\pm 0.93\%)^{10}$	**87.42%**$(\pm 0.75\%)^{10}$
RA	$76.60\%(\pm 1.31\%)^{7}$	$85.00\%(\pm 1.05\%)^{7}$	$80.73\%(\pm 1.03\%)^{9}$	$88.47\%(\pm 0.73\%)^{7}$	$78.62\%(\pm 0.61\%)^{11}$	$87.00\%(\pm 0.81\%)^{10}$
RAP	$76.18\%(\pm 1.22\%)^{7}$	**85.39%**$(\pm 0.97\%)^{6}$	$79.64\%(\pm 1.03\%)^{7}$	$87.80\%(\pm 0.82\%)^{3}$	$77.08\%(\pm 1.05\%)^{9}$	$86.34\%(\pm 0.84\%)^{10}$
RP	$76.52\%(\pm 0.80\%)^{8}$	$85.38\%(\pm 0.87\%)^{8}$	$78.45\%(\pm 0.88\%)^{8}$	$87.11\%(\pm 0.90\%)^{4}$	$76.35\%(\pm 0.67\%)^{9}$	$85.70\%(\pm 0.88\%)^{11}$
P	$75.96\%(\pm 0.80\%)^{11}$	$84.94\%(\pm 1.03\%)^{6}$	$77.56\%(\pm 0.95\%)^{5}$	$86.40\%(\pm 0.93\%)^{5}$	$75.68\%(\pm 0.67\%)^{11}$	$85.12\%(\pm 0.92\%)^{11}$
Meta Regression IoU_{adj}						
	LR		LR L1		LR L2	
	σ	R^2	σ	R^2	σ	R^2
R	**0.128**$(\pm 0.003)^{2}$	$83.48\%(\pm 0.99\%)^{2}$	$0.129(\pm 0.003)^{2}$	$83.37\%(\pm 0.92\%)^{2}$	**0.128**$(\pm 0.003)^{2}$	$83.49\%(\pm 0.96\%)^{2}$
RA	$0.134(\pm 0.003)^{2}$	$82.06\%(\pm 0.96\%)^{2}$	$0.134(\pm 0.003)^{3}$	$82.09\%(\pm 0.94\%)^{3}$	$0.134(\pm 0.003)^{2}$	$82.08\%(\pm 0.95\%)^{2}$
RAP	$0.129(\pm 0.003)^{7}$	$83.38\%(\pm 0.89\%)^{7}$	$0.129(\pm 0.003)^{7}$	$83.35\%(\pm 0.90\%)^{7}$	$0.129(\pm 0.003)^{7}$	$83.40\%(\pm 0.92\%)^{7}$
RP	**0.128**$(\pm 0.003)^{7}$	**83.62%**$(\pm 0.91\%)^{7}$	**0.128**$(\pm 0.002)^{7}$	**83.54%**$(\pm 0.88\%)^{7}$	**0.128**$(\pm 0.003)^{7}$	**83.61%**$(\pm 0.91\%)^{7}$
P	**0.128**$(\pm 0.003)^{7}$	$83.43\%(\pm 0.90\%)^{7}$	$0.129(\pm 0.002)^{7}$	$83.36\%(\pm 0.86\%)^{7}$	$0.129(\pm 0.003)^{7}$	$83.41\%(\pm 0.91\%)^{7}$
	GB		NN L1		NN L2	
	σ	R^2	σ	R^2	σ	R^2
R	$0.114(\pm 0.004)^{5}$	$87.02\%(\pm 1.00\%)^{5}$	**0.114**$(\pm 0.005)^{1}$	$86.98\%(\pm 1.07\%)^{1}$	**0.113**$(\pm 0.005)^{1}$	**87.16%**$(\pm 1.25\%)^{1}$
RA	$0.116(\pm 0.004)^{3}$	$86.39\%(\pm 1.11\%)^{3}$	$0.118(\pm 0.007)^{1}$	$85.94\%(\pm 1.76\%)^{1}$	$0.116(\pm 0.005)^{1}$	$86.46\%(\pm 1.32\%)^{1}$
RAP	**0.112**$(\pm 0.003)^{7}$	**87.51%**$(\pm 0.61\%)^{7}$	**0.114**$(\pm 0.003)^{1}$	**87.03%**$(\pm 0.71\%)^{1}$	$0.114(\pm 0.005)^{1}$	$86.97\%(\pm 1.10\%)^{1}$
RP	**0.112**$(\pm 0.002)^{9}$	$87.45\%(\pm 0.72\%)^{9}$	$0.116(\pm 0.004)^{1}$	$86.51\%(\pm 0.88\%)^{1}$	$0.115(\pm 0.003)^{2}$	$86.69\%(\pm 0.85\%)^{2}$
P	$0.114(\pm 0.002)^{11}$	$86.88\%(\pm 0.67\%)^{11}$	$0.118(\pm 0.004)^{1}$	$86.13\%(\pm 0.95\%)^{1}$	$0.117(\pm 0.004)^{3}$	$86.24\%(\pm 0.99\%)^{3}$

to the influence of time series length, the results for meta regression with gradient boosting in subfigure (c) are qualitatively similar to those in subfigure (b). However, we observe in this case that the incorporation of pseudo ground truth slightly increases the performance. Noteworthily, gradient boosting trained with real ground truth and gradient boosting trained only with pseudo ground truth perform almost equally well. This shows that meta regression can be learned when there is no ground truth but a strong reference model available. Note that this (except for the data augmentation part) is in accordance to our findings for the VIPER dataset. Results for a wider range of tests (including those previously discussed) are summarized in table 4. Again we provide video sequences that visualize the IoU_{adj} prediction and the segment tracking, see https://youtu.be/YcQ-i9cHjLk. For meta classification, we achieve accuracies of up to $81.20\%(\pm 1.02\%)$ and AUROC values of up to $88.68\%(\pm 0.80\%)$, for meta regression we achieve R^2 values of up to $87.51\%(\pm 0.61\%)$. As the labeled 142 images only yield 4,877 segments, we observe overfitting in our tests for all models when increasing the length of the time series. This might serve as an explanation that in some cases, time series do not increase performance. In particular, we observe overfitting in our tests when using gradient boosting, this holds for both datasets, KITTI and VIPER. It is indeed well-known that gradient boosting requires plenty of data.

6 Conclusion and Outlook

In this work we extended the approach presented in [39] by incorporating time series as input for meta classification and regression. To this end, we introduced a light-weight tracking algorithm for semantic segmentation. From matched segments we generated time series of metrics and use these as inputs for the meta tasks. In our tests we studied the influence of the time series length on different models for the meta tasks, i.e., gradient boosting, neural networks and linear ones. Our results show significant improvements in comparison to those presented in [39]. More precisely, in contrast to the single frame approach using only linear models, we increase the accuracy by 6.78 pp and the AUROC by 5.04 pp. The R^2 value for meta regression is increased by 5.63 pp. As a further improvement, we plan to develop additional time-dynamical metrics, as the presented metrics are still single-frame based. In addition, we plan to further investigate and improve the tracking algorithm by using autoregressive time series and comparing it with approaches based on bounding boxes. Another interesting direction could be to jointly performing segmentation and tracking. The source code of our method is publicly available at https://github.com/kmaag/Time-Dynamic-Prediction-Reliability.

REFERENCES

[1] Martin Abadi, Ashish Agarwal, Paul Barham, et al. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. Software available from tensorflow.org. 5

[2] Chad Aeschliman, J Park, and Avinash C. Kak, 'A probabilistic framework for joint segmentation and tracking', pp. 1371 – 1378, (07 2010). 2, 3

[3] Hagai Attias, 'A variational bayesian framework for graphical models', in *In Advances in Neural Information Processing Systems 12*, pp. 209–215. MIT Press, (2000). 1

[4] B. Babenko, M. Yang, and S. Belongie, 'Visual tracking with online multiple instance learning', in *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 983–990, (June 2009). 2

[5] Vasileios Belagiannis, Falk Schubert, Nassir Navab, et al., 'Segmentation based particle filtering for real-time 2d object tracking', in *Proceedings of the 12th European Conference on Computer Vision - Volume Part IV*, ECCV'12, pp. 842–855, Berlin, Heidelberg, (2012). Springer-Verlag. 2, 3

[6] Luca Bertinetto, Jack Valmadre, João F. Henriques, et al., 'Fully-convolutional siamese networks for object tracking', *CoRR*, **abs/1606.09549**, (2016). 2, 3

[7] D. S. Bolme, J. R. Beveridge, B. A. Draper, et al., 'Visual object tracking using adaptive correlation filters', in *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 2544–2550, (June 2010). 2

[8] Nitesh Chawla, Kevin Bowyer, Lawrence O. Hall, et al., 'Smote: Synthetic minority over-sampling technique', *J. Artif. Intell. Res. (JAIR)*, **16**, 321–357, (01 2002). 6

[9] Liang-Chieh Chen, Yukun Zhu, George Papandreou, et al., 'Encoder-decoder with atrous separable convolution for semantic image segmentation', in *ECCV*, (2018). 2, 5

[10] Franois Chollet, 'Xception: Deep learning with depthwise separable convolutions', *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 1800–1807, (2017). 5

[11] D. Comaniciu, V. Ramesh, and P. Meer, 'Real-time tracking of non-rigid objects using mean shift', in *Proceedings IEEE Confer-*

ence on Computer Vision and Pattern Recognition. CVPR 2000 (Cat. No.PR00662), volume 2, pp. 142–149 vol.2, (June 2000). 2

[12] Marius Cordts, Mohamed Omran, Sebastian Ramos, et al., 'The cityscapes dataset for semantic urban scene understanding', in *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, (2016). 6

[13] Martin Danelljan, Gustav Hager, Fahad Khan, et al., 'Learning spatially regularized correlation filters for visual tracking', pp. 4310–4318, (12 2015). 2

[14] Terrance DeVries and Graham W. Taylor, 'Leveraging uncertainty estimates for predicting segmentation quality', *CoRR*, **abs/1807.00502**, (2018). 1, 3

[15] S. Duffner and C. Garcia, 'Pixeltrack: A fast adaptive algorithm for tracking non-rigid objects', in *2013 IEEE International Conference on Computer Vision*, pp. 2480–2487, (Dec 2013). 2, 3

[16] David Duvenaud, Dougal Maclaurin, and Ryan Adams, 'Early stopping as nonparametric variational inference', in *Proceedings of the 19th International Conference on Artificial Intelligence and Statistics*, eds., Arthur Gretton and Christian C. Robert, volume 51 of *Proceedings of Machine Learning Research*, pp. 1070–1077, Cadiz, Spain, (09–11 May 2016). PMLR. 1

[17] C.E. Erdem, B. Sankur, and A.M. Tekalp, 'Performance measures for video object segmentation and tracking', *IEEE Transactions on Image Processing*, **13**, (2004). 1, 3

[18] Jerome H. Friedman, 'Stochastic gradient boosting', *Comput. Stat. Data Anal.*, **38**(4), 367–378, (February 2002). 4

[19] Yarin Gal and Zoubin Ghahramani, 'Dropout as a bayesian approximation: Representing model uncertainty in deep learning', in *Proceedings of the 33rd International Conference on International Conference on Machine Learning - Volume 48*, ICML'16, pp. 1050–1059. JMLR.org, (2016). 1, 3

[20] A Geiger, P Lenz, C Stiller, et al., 'Vision meets robotics: The kitti dataset', *The International Journal of Robotics Research*, **32**(11), 1231–1237, (2013). 2, 5

[21] K. Hariharakrishnan and D. Schonfeld, 'Fast object tracking using adaptive block matching', *IEEE Transactions on Multimedia*, **7**(5), 853–859, (Oct 2005). 2

[22] Anfeng He, Chong Luo, Xinmei Tian, et al., 'Towards a better match in siamese network based visual object tracker', *CoRR*, **abs/1809.01368**, (2018). 2, 3

[23] David Held, Sebastian Thrun, and Silvio Savarese, 'Learning to track at 100 FPS with deep regression networks', *CoRR*, **abs/1604.01802**, (2016). 2, 3

[24] Dan Hendrycks and Kevin Gimpel, 'A baseline for detecting misclassified and out-of-distribution examples in neural networks', *CoRR*, **abs/1610.02136**, (2016). 1

[25] C. Huang, Q. Wu, and F. Meng, 'Qualitynet: Segmentation quality evaluation with deep convolutional networks', in *2016 Visual Communications and Image Processing (VCIP)*, pp. 1–4, (Nov 2016). 1, 3

[26] Po-Yu Huang, Wan-Ting Hsu, Chun-Yueh Chiu, et al., 'Efficient uncertainty estimation for semantic segmentation in videos', in *European Conference on Computer Vision (ECCV)*, (2018). 1

[27] Paul Jaccard, 'The distribution of the flora in the alpine zone', *New Phytologist*, **11**(2), 37–50, (February 1912). 1

[28] B. A. Jilani, T. Rabie, and M. Baziyad, 'Autonomous motion tracking for dynamic objects using a temporal quad-tree algorithm', in *2019 Advances in Science and Engineering Technology International Conferences (ASET)*, pp. 1–5, (March 2019). 2

[29] Michael Kampffmeyer, Arnt-Børre Salberg, and Robert Jenssen, 'Semantic segmentation of small objects and modeling of uncertainty in urban remote sensing images using deep convolutional neural networks', *2016 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 680–688, (2016). 1, 3

[30] Alex Kendall, Vijay Badrinarayanan, and Roberto Cipolla, 'Bayesian segnet: Model uncertainty in deep convolutional encoder-decoder architectures for scene understanding', *CoRR*, **abs/1511.02680**, (2015). 1, 3

[31] Matej Kristan, Jiri Matas, Ales Leonardis, et al., 'A novel performance evaluation methodology for single-target trackers', *CoRR*, **abs/1503.01313**, (2015). 2

[32] Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell, 'Simple and scalable predictive uncertainty estimation using deep ensembles', in *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, pp. 6405–6416, USA, (2017). Curran Associates Inc. 1

[33] B. Li, J. Yan, W. Wu, et al., 'High performance visual tracking with siamese region proposal network', in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8971–8980, (June 2018). 2, 3

[34] Shiyu Liang, Yixuan Li, and R. Srikant, 'Principled detection of out-of-distribution examples in neural networks', *CoRR*, **abs/1706.02690**, (2017). 1

[35] David J. C. MacKay, 'A practical bayesian framework for backpropagation networks', *Neural Computation*, **4**(3), 448–472, (1992). 1

[36] X. Mu, J. Che, T. Hu, et al., 'A video object tracking algorithm combined kalman filter and adaptive least squares under occlusion', in *2016 9th International Congress on Image and Signal Processing, BioMedical Engineering and Informatics (CISP-BMEI)*, pp. 6–10, (Oct 2016). 2

[37] Philip Oberdiek, Matthias Rottmann, and Hanno Gottschalk, 'Classification uncertainty of deep neural networks based on gradient information', in *Artificial Neural networks and Pattern Recognition (ANNPR)*, (2018). 1, 3

[38] Stephan R. Richter, Zeeshan Hayder, and Vladlen Koltun, 'Playing for benchmarks', in *IEEE International Conference on Computer Vision, ICCV 2017, Venice, Italy, October 22-29, 2017*, pp. 2232–2241, (2017). 2, 5

[39] Matthias Rottmann, Pascal Colling, Thomas-Paul Hack, et al., 'Prediction error meta classification in semantic segmentation: Detection via aggregated dispersion measures of softmax probabilities', *CoRR*, **abs/1811.00648**, (2018). 1, 2, 3, 4, 7

[40] Matthias Rottmann and Marius Schubert, 'Uncertainty measures and prediction quality rating for the semantic segmentation of nested multi resolution street scene images', *CoRR*, **abs/1904.04516**, (2019). 1, 2, 4

[41] Olga Russakovsky, Jia Deng, Hao Su, et al., 'ImageNet Large Scale Visual Recognition Challenge', *International Journal of Computer Vision (IJCV)*, **115**(3), 211–252, (2015). 5

[42] Mark Sandler, Andrew G. Howard, Menglong Zhu, et al., 'Inverted residuals and linear bottlenecks: Mobile networks for classification, detection and segmentation', *CoRR*, **abs/1801.04381**, (2018). 5

[43] J. Son, I. Jung, K. Park, et al., 'Tracking-by-segmentation with online gradient boosting decision tree', in *2015 IEEE International Conference on Computer Vision (ICCV)*, pp. 3056–3064, (Dec 2015). 2, 3

[44] Robert Tibshirani, 'Regression shrinkage and selection via the lasso', *Journal of the Royal Statistical Society: Series B*, **58**, 267–288, (1996). 4

[45] Luís Torgo, Rita P. Ribeiro, Bernhard Pfahringer, et al., 'Smote for regression', in *Progress in Artificial Intelligence*, pp. 378–389, Berlin, Heidelberg, (2013). Springer Berlin Heidelberg. 6

[46] Jack Valmadre, Luca Bertinetto, Joao Henriques, et al., 'End-to-end representation learning for correlation filter based tracking', in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, (July 2017). 2

[47] Qiang Wang, Li Zhang, Luca Bertinetto, et al., 'Fast online object tracking and segmentation: A unifying approach', *CoRR*, **abs/1812.05050**, (2018). 2, 3

[48] Kristoffer Wickstrøm, Michael Kampffmeyer, and Robert Jenssen, 'Uncertainty and interpretability in convolutional neural networks for semantic segmentation of colorectal polyps', *CoRR*, **abs/1807.10584**, (2018). 1, 3

[49] Y. Wu, J. Lim, and M. Yang, 'Online object tracking: A benchmark', in *2013 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2411–2418, (June 2013). 2

[50] Y. Xiang, A. Alahi, and S. Savarese, 'Learning to track: Online multi-object tracking by decision making', in *2015 IEEE International Conference on Computer Vision (ICCV)*, pp. 4705–4713, (Dec 2015). 2

[51] Tianyu Yang and Antoni B. Chan, 'Learning dynamic memory networks for object tracking', *CoRR*, **abs/1803.07268**, (2018). 2, 3

[52] R. Yao, G. Lin, C. Shen, et al., 'Semantics-aware visual object tracking', *IEEE Transactions on Circuits and Systems for Video Technology*, **29**(6), 1687–1700, (June 2019). 2

[53] D. Yeo, J. Son, B. Han, et al., 'Superpixel-based tracking-by-segmentation using markov chains', in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 511–520, (July 2017). 2

[54] Alper Yilmaz, Omar Javed, and Mubarak Shah, 'Object tracking: A survey', *ACM Computing Surveys*, **38**, 1–45, (01 2006). 1