Gradient-Based Quantification of Epistemic Uncertainty for Deep Object Detectors

Tobias Riedlinger[1] Matthias Rottmann[1,2] Marius Schubert[1] Hanno Gottschalk[1]

[1] School of Mathematics and Natural Sciences, IZMD, University of Wuppertal, Germany
[2] School of Computer and Communication Sciences, CVLab, EPFL, Switzerland
{riedlinger@math., rottmann@math., mschubert@, hgottsch@}uni-wuppertal.de

Abstract

The majority of uncertainty quantification methods for deep object detectors are based on the network output, such as sampling strategies like Monte-Carlo dropout or deep ensembles with straight-forward transfers to object detection. Here, we study gradient-based uncertainty features for object detection. We show that they contain information orthogonal to that of common, output-based uncertainty approximation methods. Meta classification and meta regression are used to produce confidence estimates using gradient features and other methods which are applicable to numerous object detection architectures. Our results show that gradient uncertainty itself performs on par with state-of-the-art methods across different detectors and datasets. We find that combined meta classifiers outperform standalone models. This suggests that sampling strategies may be supplemented by gradient-based uncertainty to obtain improved confidences, contributing to the probabilistic reliability of object detectors in down-stream applications.

1. Introduction

Deep artificial neural networks (DNNs) designed for tasks such as object detection or semantic segmentation provide a probabilistic prediction on given feature data such as camera images. Modern deep object detection architectures [28, 43, 44, 26, 1] predict bounding boxes for instances of a set of learned classes on an input image. The so-called *objectness or confidence score* indicates the probability of the existence of an object for each predicted bounding box. Throughout this work, we will refer to this quantity which the DNN learns by the term "score". For applications of deep object detectors such as automated surgery or driving, the reliability of this component is crucial. See, for example the detection in the top panel of fig. 1 where each box is colored from red (low score) to green (high score). Apart from the accurate, green boxes, boxes with a score below 0.3 (dashed) contain true and false predictions which *cannot be reliably separated in terms of their score*. In addition, it is well-known that DNNs tend to give mis-calibrated scores [52, 12, 13] that are oftentimes over-confident and may also lead to unreliable predictions. Over-confident predictions might render an autonomous driving system inoperable by perceiving non-existent instances (false positives / FP). Perhaps even more detrimental, under-confidence may lead to overlooked (false negative / FN) predictions possibly endangering humans outside autonomous vehicles like pedestrians and cyclists, as well as the passengers.

Figure 1. Object detection in a street scene. Top coloration: Score $\hat{s}$; bottom coloration: instance-wise gradient-based confidence $\hat{\tau}$ obtained by our method. Dashed boxes here indicate the discarding at any confidence threshold in $[0.3, 0.85]$. The top image contains FNs which are not separable from correctly discarded boxes based on the score (lower threshold would lead to FPs). In the bottom image, those $\hat{s}$-FNs are assigned higher confidences and there is a large range of thresholds with no FPs.

Apart from modifying and improving the detection architecture or the loss function, there exist methods to estimate prediction confidence which are more involved than the score in order to remedy these issues [36, 30, 48]. We use the term "confidence" more broadly than "score" to refer to quantities which represent the estimated probability of a detection being correct. Such a quantity should reflect the model's overall level of competency when confronted with a given input and is intimately linked to prediction uncertainty. Uncertainty for statistical models, in particular DNNs, can broadly be divided into two types [18] depending on their primary source [9, 20]. Whereas aleatoric uncertainty is mainly founded in the stochastic nature of the data generating process, epistemic uncertainty stems from the probabilistic nature of sampling data for training, as well, as the choice of model and the training algorithm. The latter is technically reducible by obtaining additional training data and is the central subject of our method.

Due to the instance-based nature of deep object detection, modern ways of capturing epistemic uncertainty are mainly based on the instance-wise DNN output. From a theoretical point of view, Bayesian DNNs [5, 33] represent an attractive framework for capturing epistemic uncertainty for DNNs by modeling their weights as random variables. Practically, this approach introduces a large computational overhead making its application infeasible for object detection. Therefore, in variational inference approaches, weights are sampled from predefined distributions to address this. These famously include methods like Monte-Carlo (MC) dropout [50, 10] generating prediction variance by performing several forward passes under active dropout. The same idea underlies deep ensemble sampling [25] where separately trained models with the same architecture produce variational forward passes. Other methods based on the classification output can also be applied to object detection such as softmax entropy or energy methods.

A number of other, strong uncertainty quantification methods that do not only rely on the classification output has also been developed for image classification architectures [4, 34, 40, 42]. However, the *transfer of such methods to object detection* frameworks can pose serious challenges, if at all possible, due to architectural restrictions. For example, the usage of a learning gradient evaluated at the network's own prediction was proposed [40] to contain epistemic uncertainty information for image classification and investigated for out-of-distribution (OoD) data. The method has also been applied natural language understanding [54] where gradient features and deep ensemble uncertainty were aggregated to obtain well-calibrated confidence measures on OoD data. The epistemic content of gradient uncertainty has further been explored in [17] in the classification setting by observing shifts in the data distribution.

We propose a way to compute gradient features for the prediction of deep object detectors. We show that they perform on par with state-of-the-art uncertainty quantification methods and that they contain information that can not be obtained from output- or sampling-based methods. In particular, *we summarize our main contributions as follows*:

- We introduce a way of generating gradient-based uncertainty features for modern object detection architectures, allowing to generate uncertainty information from hidden network layers.
- We investigate the performance of gradient features in terms of meta classification (FP detection), calibration and meta regression (prediction of intersection over union IoU with the ground truth) and compare them to other means to quantify/approximate epistemic uncertainty and investigate mutual redundancy as well as detection performance through gradient uncertainty.
- We explicitly investigate the FP/FN-tradeoff for pedestrian detection based on the score and meta classifiers.
- We provide a theoretical treatment of the computational complexity of gradient features in comparison with MC dropout and deep ensembles and show that their FLOP count is similar at worst. Explicit runtime measurements are performed for verification.

An implementation of our method is publicly available at https://github.com/tobiasriedlinger/gradient-metrics-od. A video illustration of our method is publicly available at https://youtu.be/L4oVNQAGiBc.

2. Related work

Epistemic uncertainty for deep object detection. Sampling-based uncertainty quantification such as MC dropout and deep ensembles have been investigated in the context of object detection by several authors in the past. They are straight-forward to implement into any architecture and yield output variance for all bounding box features. Harakeh *et al*. [14] employed MC dropout and Bayesian inference as a replacement of Non-Maximum Suppression (NMS) to get a joint estimation of epistemic and aleatoric uncertainty. Similarly, epistemic uncertainty measures were obtained by Kraus and Dietmayer [23] from MC dropout. Miller *et al*. [36] investigated MC dropout as a means to improve object detection performance in open-set conditions. Different merging strategies for samples from MC dropout were investigated by Miller *et al*. [35] and compared with the influence of merging boxes in deep ensembles [37]. Lyu *et al*. [30] aggregated deep ensemble samples as if produced from a single detector to obtain improved detection performance. A variety of uncertainty measures generated from proposal box variance pre-NMS called MetaDetect was investigated by Schubert *et al*. [48]. In generating advanced scores and IoU estimates,

it was reported that the obtained information is largely redundant with MC dropout uncertainty features. All of the above methods are based on the network output and generate variance by aggregating prediction proposals in some manner. Moreover, a large amount of uncertainty quantification methods based on classification outputs can be directly applied to object detection [16, 29]. Little is known about other methods developed for image classification that are not directly transferable to object detection due to architectural constraints (*e.g.*, activation-based [4] or gradient-based [40] uncertainty). The central difficulty lies in the fact that *different predicted instances depend on shared latent features or DNN weights* such that the base method can only estimate uncertainty for the entire prediction (all instances) instead of individual estimates per instance. We show that gradient uncertainty information can be extracted from hidden layers in object detectors, seek to determine how they compare to output-based methods and show that they contain orthogonal information.

Meta classification and meta regression. The term meta classification refers to the discrimination of TPs from FPs on the basis of uncertainty features which was first explored by Hendrycks and Gimpel [16] to detect OoD samples based on the maximum softmax probability. Since then, the approach has been applied to natural language processing [54], semantic segmentation [2, 31, 46, 45, 47], instance segmentation in videos [32] and object detection [48, 22] to detect FP predictions on the basis of uncertainty features accessible during inference. Moreover, meta regression (the estimation of IoU based on uncertainty in the same manner) was also investigated [31, 32, 46, 47, 48] showing large correlations between estimates and the true localization quality. Chan *et al*. [2] have shown that meta classification can be used to improve network accuracy, an idea that so-far has not been achieved for object detection. Previous studies have overlooked class-restricted meta classification performance, *e.g.*, when restricting to safety-relevant instance classes. Moreover, in order to base downstream applications on meta classification outputs, resulting confidences need to be statistically reliable, *i.e.*, calibrated which has also escaped previous research.

3. Gradient-based epistemic uncertainty

In instance-based recognition tasks, such as object detection or instance segmentation, the prediction

$$\hat{y} = (\hat{y}^1, \dots, \hat{y}^{N_{\boldsymbol{x}}}) \tag{1}$$

consists of a list of instances (*e.g.*, bounding boxes). The length of $\hat{y}$ usually depends on the corresponding input $\boldsymbol{x}$ and on hyperparameters (*e.g.*, confidence / overlap thresholds). Uncertainty information which is not generated directly from instance-wise data such as activation- or gradient-based information can *at best yield statements about the entirety of* $\hat{y}$ but not immediately about any individual instance $\hat{y}^j$. This issue is especially apparent for uncertainty generated from deep features which potentially all contribute to an instance $\hat{y}^j$. Here, we introduce an approach to generate gradient-based uncertainty features for the instance-based setting. To this end, we sketch how gradient uncertainty is generated for classification tasks.

Generically, given an input $\boldsymbol{x}$, a classification network predicts a class distribution $\hat{y}(\boldsymbol{x}, \boldsymbol{w}) = (\hat{p}_1, \dots, \hat{p}_C)$ of fixed length C given a set of weights $\boldsymbol{w}$. During training, the latter is compared to the ground truth label y belonging to $\boldsymbol{x}$ by means of some loss function $\mathcal{L}(\cdot, \cdot)$, which is minimized by optimizing $\boldsymbol{w}$, *e.g.*, by standard stochastic gradient descent. The $\boldsymbol{w}$-step is proportional to the gradient $g(\boldsymbol{x}, \boldsymbol{w}, y) := \nabla_{\boldsymbol{w}} \mathcal{L}(\hat{y}(\boldsymbol{x}, \boldsymbol{w}), y)$ which can also be regarded as a measure of *learning stress* imposed upon $\boldsymbol{w}$. Gradient uncertainty features are generated by substituting the non-accessible ground truth y with the network's class prediction $\overline{y} := \arg\max_c \{\hat{p}_c\}_{c=1}^C$ and disregarding the dependence of the latter on $\boldsymbol{w}$. In the following we will identify $\overline{y}$ with its one-hot encoding. Scalar values are obtained by computing some magnitude of

$$g(\boldsymbol{x}, \boldsymbol{w}, \overline{y}) = \nabla_{\boldsymbol{w}} \mathcal{L}(\hat{y}(\boldsymbol{x}, \boldsymbol{w}), \overline{y}). \tag{2}$$

To this end, in our experiments we employ the maps

$$\{\min(\cdot), \max(\cdot), \mathrm{mean}(\cdot), \mathrm{std}(\cdot), \|\cdot\|_1, \|\cdot\|_2\}. \tag{3}$$

We discuss the latter choice in our supplementary material and first illuminate some points about eq. (2).

Intuition and discussion of (2)**.** First of all, eq. (2) can be regarded as the *self-learning gradient* of the network. It, therefore, expresses the learning stress on $\boldsymbol{w}$ under the condition that the class prediction $\overline{y}$ were given as the ground truth label. The collapse of the (*e.g.*, softmax) prediction $\hat{y}$ to $\overline{y}$ implies that (2) does not generally vanish in the classification setting. However, this consideration poses a problem for (bounding box) regression which we will address in the next paragraph. We also note that it is possible to generate fine-grained features by restricting $\boldsymbol{w}$ in eq. (2) to sub-sets of weights $\boldsymbol{w}_\ell$, *e.g.* individual layers, convolutional filters or singular weights (computing partial gradients of $\mathcal{L}$).

Using eq. (2) as a measure of uncertainty may be understood by regarding true and false predictions. A well-performing network which has $\overline{y}$ already close to the true label y tends to experience little stress when trained on $(y, \boldsymbol{x})$ with the usual learning gradient. This reflects *confidence* in the prediction $\overline{y}$ and the difference between eq. (2) and the true gradient is then small. In the case of false predictions $\overline{y} \neq y$, the true learning gradient enforces large adjustments in $\boldsymbol{w}$. The self-learning gradient eq. (2) behaves differently in that it is *large for non-peaked / uncertain* (high entropy) predictions $\hat{y}$ and small for highly peaked distributions.

The following consideration establishes a link to empirical findings. Assuming that we draw data $(y, \boldsymbol{x}) \sim p$ from a fixed distribution p, we regard $g(\boldsymbol{x}) := g(\boldsymbol{x}, \boldsymbol{w}, \overline{y})$. A simple computation (cf. appendix E) shows that

$$\mathbb{E}_{(y,\boldsymbol{x})}[\|g(\boldsymbol{x})\| \, | y = \overline{y}(\boldsymbol{x})] < \mathbb{E}_{(y,\boldsymbol{x})}[\|g(\boldsymbol{x})\| \, | y \neq \overline{y}(\boldsymbol{x})] \quad (4)$$

holds, if and only if $\mathrm{Cov}(\|g(\boldsymbol{x})\|, \varepsilon(\boldsymbol{x})) > 0$, where $\varepsilon(\boldsymbol{x}) = \sum_{c \neq \overline{y}(\boldsymbol{x})} p(c|\boldsymbol{x})$ is the model's conditional error rate. Such an actual positive correlation between $\|g(\boldsymbol{x})\|$ and the local error rate has been established independently in experiments before[40, 51]. Note further, that for precise models where $g(\boldsymbol{x}) \approx g(\boldsymbol{x}, \boldsymbol{w}, y)$, this relation is indicative of epistemic uncertainty as the model will adapt more strongly to instances where $\varepsilon(\boldsymbol{x})$ is (still) large.

Extension to object detectors. We first clarify the aforementioned complications in generating uncertainty information for object detection. Generally, the prediction (1) is the filtering result of a larger, often fixed number $\hat{N}_{\mathrm{out}}$ of output bounding boxes $\widetilde{y}(\boldsymbol{x}, \boldsymbol{w})$. Given a ground truth list y of bounding boxes, the loss function usually has the form

$$\mathcal{L} = \mathcal{L}(\widetilde{y}(\boldsymbol{x}, \boldsymbol{w}), y), \quad (5)$$

such that all $\hat{N}_{\mathrm{out}}$ output bounding boxes potentially contribute to $g(\boldsymbol{x}, \boldsymbol{w}, y)$. Again, when filtering $\widetilde{y}$ to a smaller number of predicted boxes $\hat{y}$ and converting them to ground truth format $\overline{y}$, we can compute the self-learning gradient $g(\boldsymbol{x}, \boldsymbol{w}, \overline{y})$. This quantity, however, does not refer to any individual prediction $\hat{y}^j$, but rather to all boxes in $\overline{y}$ simultaneously. We take *two steps to obtain meaningful gradient information* for one particular box $\hat{y}^j$ from this approach.

Firstly, we restrict the ground truth slot to only contain the length-one list $\overline{y}^j$, regarding it as the hypothetical label. This alone is insufficient since other, correctly predicted instances in $\widetilde{y}(\boldsymbol{x}, \boldsymbol{w})$ would lead to a penalization and "overcorrecting" gradient $g(\boldsymbol{x}, \boldsymbol{w}, \overline{y}^j)$, given $\overline{y}^j$ as label. This gradient's optimization goal is, figuratively speaking, to forget to predict everything but $\hat{y}^j$ when presented with $\boldsymbol{x}$. Note that we cannot simply compute $\nabla_{\boldsymbol{w}} \mathcal{L}(\hat{y}^j(\boldsymbol{x}, \boldsymbol{w}), \overline{y}^j)$ since regression losses, such as for bounding box regression, are frequently norm-based (*e.g.* L^p-losses) such that the respective loss and gradient would both vanish. Therefore, we *secondly* mask $\widetilde{y}$ such that the result is likely to only contain output boxes meaning to predict the same instance as $\overline{y}^j$. Our conditions for this mask are *sufficient score*, *sufficient overlap* with $\overline{y}^j$ and *same indicated class* as $\overline{y}^j$ (the predictions which would be suppressed by $\overline{y}^j$ in NMS). We call the subset of $\widetilde{y}$ that satisfies these conditions *candidate boxes* for $\overline{y}^j$, denoted $\mathrm{cand}[\overline{y}^j]$. We, thus, propose the candidate-restriced self-learning gradient

$$g^{\mathrm{cand}}(\boldsymbol{x}, \boldsymbol{w}, \hat{y}^j) := \nabla_{\boldsymbol{w}} \mathcal{L}\left(\mathrm{cand}[\hat{y}^j](\boldsymbol{x}, \boldsymbol{w}), \overline{y}^j\right) \quad (6)$$

of $\hat{y}^j$ for computing instance-wise uncertainty. This approach is in line with the motivation for the classification setting and extends it when computing (6) for the multi-criterial loss function in object detection.

Computational complexity. Sampling-based epistemic uncertainty quantification methods such as MC dropout and deep ensembles tend to generate a significant computational overhead as several forward passes are required. Here, we provide a theoretical result on the count of floating point operations (FLOP) of gradient uncertainty features which is supported with a proof and additional detail in appendix D. In our experiments, we use the gradients computed over the last one, resp. two layers of each network architecture (of different architectural branches, as well, if applicable). For layer t, we assume stride-1, $(2s_t + 1) \times (2s_t + 1)$-convolutional layers acting on features maps of spatial size $w_t \times h_t$. These assumptions hold for all architectures in our experiments. We denote the number of input channels by k_{t-1} and of output channels by k_t.

Theorem 1 *The number of FLOP required to compute the last layer ($t = T$) gradient in eq.* (6) *is* $\mathcal{O}(k_T hw + k_T k_{T-1}(2s_T + 1)^4)$. *Similarly, for earlier layers t, we have* $\mathcal{O}(k_{t+1}k_t + k_t k_{t-1})$*, provided that we have previously computed the gradient for the consecutive layer $t + 1$. Performing variational inference only on the last layer requires* $\mathcal{O}(k_T k_{T-1} hw)$ *FLOP per sample.*

Theorem 1 provides that even for MC dropout only before the last layer, or the use of efficient deep sub-ensembles [53] sharing the entire architecture but the last layer, gradient features require fewer or at worst similar FLOP counts. Earlier sampling, especially entire deep ensembles, have even higher FLOP counts than these variants. Note, that computing gradient features have somewhat larger computational latency since the full forward pass needs to be computed before gradients can be computed. Moreover, while sampling strategies can in principle be implemented to run all sample forward passes in parallel, the computation of gradients can run in parallel for predicted boxes per image. We compare explicit time measurements for different methods in section 5 and provide a proof of theorem 1 in appendix D.

4. Meta classification and meta regression

We evaluate the efficacy of gradient scores in terms of meta classification and meta regression. These two approaches allow for the *aggregation of potentially large feature vectors to obtain uncertainty estimates* for a respective prediction (*e.g.*, a bounding box). The aim of meta classification is to detect FP predictions by generating confidence estimates while meta regression directly estimates the prediction quality (*e.g.*, IoU). This, in turn, allows for the unified comparison of different uncertainty quantification methods and combinations thereof by regarding meta classifiers and meta regression models based on different features. Moreover, we are able to investigate the degree of mutual redun-

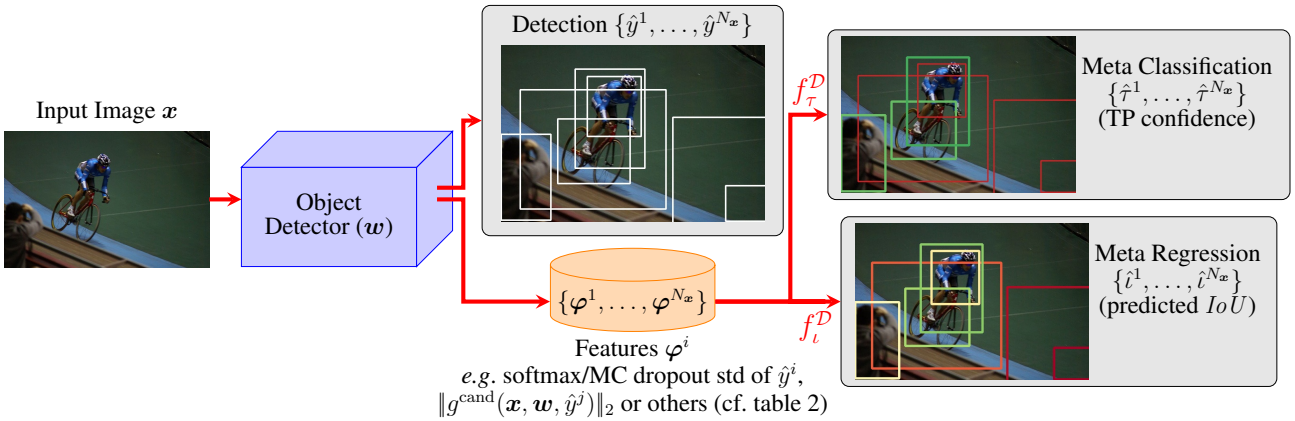

Figure 2. Meta classification and meta regression pipeline: An uncertainty feature vector $\boldsymbol{\varphi}^j$ is assigned to each detected box $\hat{y}^j$. During training, we fit $f_\tau^{\mathcal{D}}$ and $f_\iota^{\mathcal{D}}$ to map $\boldsymbol{\varphi}^j$ to τ^j (TP/FP) and max. IoU ι^j of $\hat{y}^j$, resp. At inference, $f_\tau^{\mathcal{D}}$ and $f_\iota^{\mathcal{D}}$ yield confidence and IoU estimates $\hat{\tau}^j$ and $\hat{\iota}^j$ for $\hat{y}^j$ based on $\boldsymbol{\varphi}^j$.

dancy of different sources of uncertainty. In the following, we summarize this method for bounding box detection and illustrate the scheme in fig. 2.

We regard an object detector generating a list of $N_{\boldsymbol{x}}$ detections along with a vector $\boldsymbol{\varphi}^j$ for each predicted bounding box $\hat{y}^j$. This vector $\boldsymbol{\varphi}^j \in \mathbb{R}^n$ of n "features" may contain gradient scores, but also, *e.g.*, bounding box features, MC dropout or deep ensemble features or combinations thereof (*e.g.*, by concatenation of dropout and ensemble feature vectors). On training data $\mathcal{D}$, we compute boxes $\hat{y}$ and corresponding features $\varphi = (\boldsymbol{\varphi}^1, \ldots, \boldsymbol{\varphi}^{N_{\boldsymbol{x}}})$. We evaluate each predicted instance $\hat{y}^j$ corresponding to the features φ^j in terms of their maximal IoU, denoted $\iota^j \in [0, 1]$ with the respective ground truth and determine FP/TP labels $\tau^j \in \{0, 1\}$. A *meta classifier* is a lightweight classification model $f_\tau : \mathbb{R}^n \to (0, 1)$ giving probabilities for the classification of $\boldsymbol{\varphi}^j$ (vicariously for the uncertainty of $\hat{y}^j$) as TP which we fit on $\mathcal{D}$. Similarly, a *meta regression* model $f_\iota : \mathbb{R}^n \to \mathbb{R}$ is fit to the maximum IoU ι^j of $\hat{y}^j$ with the ground truth of $\boldsymbol{x}$. The models $f_\tau^{\mathcal{D}}$ and $f_\iota^{\mathcal{D}}$ can be regarded as post-processing modules which generate confidence measures given an input to an object detector leading to features φ^j. At inference time, we then obtain box-wise classification probabilities $\hat{\tau}^k = f_\tau^{\mathcal{D}}(\varphi^k)$ and IoU predictions $\hat{\iota}^k = f_\iota^{\mathcal{D}}(\varphi^k)$. We then determine the predictive power of $f_\tau^{\mathcal{D}}$ and $f_\iota^{\mathcal{D}}$ in terms of their area under receiver operating characteristic ($AuROC$) or average precision (AP) metrics and the determination coefficient (R^2), respectively.

MetaFusion (object detection post-processing). As a direct application of uncertainty quantification, we investigate an approach inspired by [2]. We *implement meta classification into the object detection pipeline* by assigning each output box in $\widetilde{y}$ its meta classification probability as prediction confidence as shown in fig. 1. State-of-the-art object detectors use score thresholding in addition to NMS which we compare with confidence filtering based on meta classification. Since for most competitive uncertainty baselines in our experiments, computation for the entire pre-filtering network output $\widetilde{y}$ is expensive, we implement a small score threshold which still allows for a large amount of predicted boxes (of $\sim$ 150 bounding boxes per image). This way, well-performing meta classifiers (which accurately detect FPs) together with an increase in detection sensitivity offer a way to "trade" uncertainty information for detection performance. In most object detection pipelines, score thresholding is carried out before NMS. We choose to interchange them here as they commute for the baseline approach. The resulting predictions are compared for a range of confidence thresholds in terms of mean Average Precision (mAP [8]).

Table 1. Number of layers and losses utilized and resulting numbers of gradients per box. Multiplication in # layers denotes parallel output strands of the resp. DNN (no additional gradients).

Architecture	# layers	# Losses	# gradients
YOLOv3	2×3	3	6
Faster R-CNN	2×4	4	8
RetinaNet	2×2	2	4
Cascade R-CNN	2×8	8	16

5. Experiments

In this section, we report our numerical methods and experimental findings. We investigate meta classification and meta regression on three object detection datasets, namely Pascal VOC [8], MS COCO [27] and KITTI [11]. We investigate for gradient-based meta classification and meta regression for only 2-norm scalars, denoted $\text{GS}_{\|\cdot\|_2}$ (refer to section 3) as well as the larger model for all maps listed in eq. (3), denoted GS_{full}. GS_{full} is always computed for the last two network layers (unless specified otherwise) of each architectural branch and for each contribution to the loss

function $\mathcal{L}$ separately, *i.e.*, for classification, bounding box regression and, if applicable, objectness score. We list the resulting counts and number of gradients per investigated architecture in table 1. As meta classifiers and meta regressors, we use gradient boosting models which have been shown[54, 48, 32] to perform well as such. Whenever we indicate means and standard deviations, we obtained those by 10-fold image-wise cross validation (cv) for the training split $\mathcal{D}$ of the meta classifier / meta regression model. Evaluation is done on the complement of $\mathcal{D}$.

Comparison with output-based uncertainty. We compare gradient-based uncertainty with various uncertainty baselines in terms of meta classification (table 2) and meta regression (table 3) for a YOLOv3 model with standard Darknet53 backbone [43]. As class probability baselines, we consider objectness score, softmax entropy, energy score [29] and the full softmax distribution per box. Since the full softmax baseline fits a model directly to all class probabilities (as opposed to relying on hand-crafted functions), it can be considered an *enveloping model* to both, entropy and energy score. Moreover, we consider other output baselines in MC dropout (MC), deep ensembles (E) and MetaDetect (MD). Since MetaDetect involves the entire network output of a bounding box, it leads to meta classifiers fitted on more variables than class probability baselines. It is, thus, an enveloping model of the full softmax baseline and, therefore, all classification baselines. The results in table 2 indicate that $\mathrm{GS}_{\mathrm{full}}$ is roughly in the same $AuROC$ range as sampling-based uncertainty methods, while being consistently among the two best methods in terms of AP. The smaller gradient-based model $\mathrm{GS}_{\|\cdot\|_2}$ is consistently better than the full softmax baseline, by up to 3.14 $AuROC$ percentage points (ppts) and up to 5.60 AP ppts. We also find that $\mathrm{GS}_{\mathrm{full}}$ tends to rank lower in terms of $AuROC$. Note also, that MetaDetect is roughly on par with the sampling approaches MC and E throughout. While the latter methods aim at capturing epistemic uncertainty they constitute approximations and are, not necessarily mutually redundant.

In addition, we compare the largest sampling and output based model in MC+E+MD and add the gradient features $\mathrm{GS}_{\mathrm{full}}$ to find out about the degree of redundancy between the approximated epistemic uncertainty in MC+E+MD and our method. We note significant boosts to the already well-performing model MC+E+MD across all metrics. Table 3 suggests that gradient uncertainty is especially informative for meta regression with $\mathrm{GS}_{\mathrm{full}}$ being consistently among the best two models and achieving R^2 scores of up to 85.4 on the KITTI dataset. Adding $\mathrm{GS}_{\mathrm{full}}$ to MC+E+MD always leads to a gain of more than one R^2 ppt indicating non-redundancy of gradient- and sampling-based features.

Object detection architectures. We investigate the applicability and viability of gradient uncertainty for a variety of different architectures. In addition to the YOLOv3 model, we investigate two more standard object detectors in Faster R-CNN [44] and RetinaNet [26] both with a ResNet50 backbone [15]. Moreover, we investigate a stronger object detector in Cascade R-CNN [1] with a large ResNeSt200 [57] backbone which at the time of writing was ranked among the top 10 on the official COCO Detection Leaderboard. With a COCO detection AP of 49.03, this is in the state-of-the-art range for pure, non-hybrid-task object detectors. In table 4, we list meta classification $AuROC$ and meta regression R^2 for the score, MetaDetect (representing output-based methods), $\mathrm{GS}_{\mathrm{full}}$ and the combined model $\mathrm{GS}_{\mathrm{full}}$+MD. We see $\mathrm{GS}_{\mathrm{full}}$ again being on par with MD, in the majority of cases even surpassing it by up to 2.01 $AuROC$ ppts and up to 11.52 R^2 ppts. When added to MD, we find again boosts in both performance metrics, especially in R^2. On the COCO dataset, the high performance model Cascade R-CNN delivers a remarkably strong Score baseline completely redundant with MD and surpassing $\mathrm{GS}_{\mathrm{full}}$ on its own. However, here we also find an improvement of 0.68 ppts by adding gradient information.

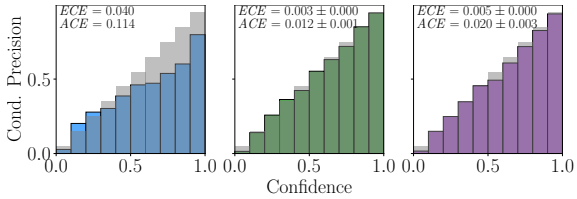

Figure 3. Reliability plots of the Score (left) and meta classifiers for MD (center) and $\mathrm{GS}_{\mathrm{full}}$ (right) on the VOC dataset (YOLOv3) with calibration errors ($\mathrm{mean} \pm \mathrm{std}$). The gray diagonal shows optimal calibration.

Calibration. We evaluate the meta classifier confidences obtained above in terms of their calibration errors when divided into 10 confidence bins. Reliability plots are shown in fig. 3 for the Score, MD and $\mathrm{GS}_{\mathrm{full}}$ together with corresponding expected (ECE [38]) and average (ACE [39]) calibration errors. The Score is clearly over-confident in the upper confidence range and both meta classifiers are well-calibrated. Both calibration errors of the latter are about one order of magnitude smaller than for the Score.

Pedestrian detection. The statistical improvement seen in table 2 may not hold for non-majority classes within a dataset which are regularly safety-relevant. We investigate meta classification of the "Pedestrian" class in KITTI and explicitly study the FP/FN trade-off. This can be accomplished by sweeping the confidence threshold between 0 and 1 and counting the resulting FPs and FNs. We choose increments of 10^{-2} for meta classifiers and 10^{-4} for the scores as to not interpolate too roughly in the range of very small score values where a significant number of predictions cluster. The resulting curves are depicted in fig. 4. For applications in safety-critical environments, not all errors need to be equally important. We may, for example,

Table 2. Meta classification performance in terms of $AuROC$ and AP per confidence model over 10-fold cv (mean $\pm$ std).

YOLOv3	Pascal VOC		COCO		KITTI	
	AuROC	*AP*	*AuROC*	*AP*	*AuROC*	*AP*
Score	90.68 ± 0.06	69.56 ± 0.12	82.97 ± 0.04	62.31 ± 0.05	96.53 ± 0.05	96.87 ± 0.03
Entropy	91.30 ± 0.02	61.94 ± 0.06	76.52 ± 0.02	42.52 ± 0.04	94.79 ± 0.06	94.83 ± 0.05
Energy Score [29]	92.59 ± 0.02	64.65 ± 0.06	75.39 ± 0.02	39.72 ± 0.06	95.66 ± 0.02	95.33 ± 0.03
Full Softmax	93.81 ± 0.06	72.08 ± 0.15	82.91 ± 0.06	58.65 ± 0.10	97.07 ± 0.03	96.85 ± 0.03
MC Dropout [50] (MC, $N_{\text{MC}} = 30$)	96.72 ± 0.02	78.15 ± 0.09	$\mathbf{89.04 \pm 0.02}$	64.94 ± 0.11	97.60 ± 0.07	97.17 ± 0.10
Ensemble[25] (E, $N_{\text{ens}} = 5$)	$\mathbf{96.87 \pm 0.02}$	77.86 ± 0.11	88.97 ± 0.02	64.05 ± 0.12	97.98 ± 0.03	97.69 ± 0.04
MetaDetect [48] (MD)	95.78 ± 0.05	$\mathbf{78.64 \pm 0.08}$	87.16 ± 0.04	69.41 ± 0.07	$\mathbf{98.23 \pm 0.02}$	$\mathbf{98.06 \pm 0.02}$
Grad. Score$_{\|\cdot\|_2}$ ($\text{GS}_{\|\cdot\|_2}$; ours)	94.76 ± 0.03	74.86 ± 0.10	86.05 ± 0.04	64.25 ± 0.06	97.31 ± 0.05	96.86 ± 0.10
Grad. Score$_{\text{full}}$ (GS_{full}; ours)	95.80 ± 0.04	78.57 ± 0.11	88.07 ± 0.03	$\mathbf{69.62 \pm 0.07}$	98.04 ± 0.03	97.81 ± 0.06
MC+E+MD	97.66 ± 0.02	85.13 ± 0.12	91.14 ± 0.02	73.82 ± 0.05	98.56 ± 0.03	98.45 ± 0.03
GS_{full}+MC+E+MD	$\mathbf{97.95 \pm 0.02}$	$\mathbf{86.69 \pm 0.09}$	$\mathbf{91.65 \pm 0.03}$	$\mathbf{74.88 \pm 0.07}$	$\mathbf{98.74 \pm 0.02}$	$\mathbf{98.62 \pm 0.01}$

Table 3. Meta regression performance in terms of R^2 per confidence model over 10-fold cv (mean $\pm$ std).

YOLOv3	Pascal VOC	COCO	KITTI
Score	48.29 ± 0.04	32.60 ± 0.02	78.86 ± 0.05
Entropy	43.24 ± 0.03	21.10 ± 0.04	69.33 ± 0.04
Energy Score	47.18 ± 0.03	17.94 ± 0.02	71.53 ± 0.10
Full Softmax	53.86 ± 0.11	36.95 ± 0.13	78.92 ± 0.11
MC	61.63 ± 0.15	43.85 ± 0.09	82.10 ± 0.11
E	61.48 ± 0.07	43.53 ± 0.13	84.18 ± 0.12
MD	60.36 ± 0.14	44.22 ± 0.11	$\mathbf{85.88 \pm 0.10}$
$\text{GS}_{\|\cdot\|_2}$ (ours)	58.05 ± 0.13	38.77 ± 0.04	81.21 ± 0.05
GS_{full} (ours)	$\mathbf{62.50 \pm 0.11}$	$\mathbf{44.90 \pm 0.09}$	85.40 ± 0.11
MC+E+MD	69.38 ± 0.11	54.07 ± 0.08	87.78 ± 0.11
GS_{full}+MC+E+MD	$\mathbf{72.26 \pm 0.08}$	$\mathbf{56.14 \pm 0.11}$	$\mathbf{88.80 \pm 0.07}$

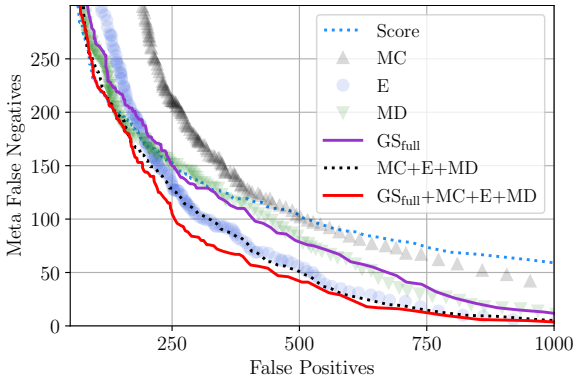

Figure 4. Meta classification for the class "Pedestrian". Curves obtained by sweeping the threshold on score / meta classification probability. Note the FP gaps for ≤ 100 FNs.

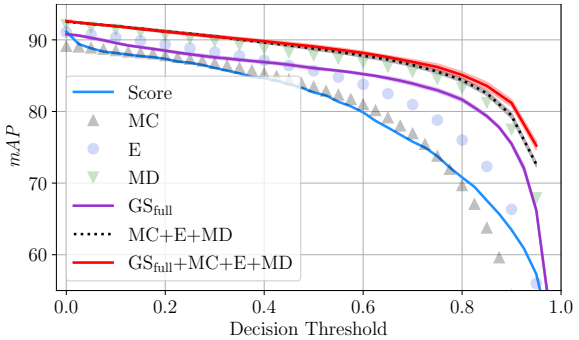

Figure 5. Score baseline and MetaFusion mAP. Error bands we draw around meta classifiers indicate cv-std.

demand a good trade-off at a given FN count which is usually desired to be especially small. Our present evaluation split contains a total of 1152 pedestrian instances. Assume that we allowed for a detector to miss around 100 pedestrians ($\sim 10\%$), we see a reduction in FPs for some meta classifiers. MD and GS_{full} are very roughly on par, leading to a reduction of close to 100 FPs. The ensemble E turns out to be about as effective as the entire output-based model MC+E+MD, only falling behind above 150 FNs. This indicates some degree of redundancy between output-based methods. Adding GS_{full} to MC+E+MD, however, reduces the number of FPs again by about 100 leading to an FP difference of about 250 as compared to the Score baseline. Observing the trend, the improvements become even more effective for smaller numbers of FNs (small thresholds) but diminish for larger numbers of above 200 FNs.

MetaFusion. In regarding fig. 2, meta classifiers naturally fit as post processing modules on top of object detection pipelines. Doing so does not generate new bounding boxes, but modifies the confidence ranking as shown in fig. 1 and may also lead to calibrated confidences. Therefore, the score baseline and meta classifiers are not comparable for fixed decision thresholds. We obtain a comparison of the resulting object detection performance by sweeping the decision threshold with a step size of 0.05 (resp. 0.025 for Score). The mAP curves are shown in fig. 5. We draw error bands showing cv-std for GS_{full}, MC+E+MD and GS_{full}+MC+E+MD. Meta classification-based decision rules are either on par (MC) with the score threshold or consistently allow for an mAP improvement of at least 1 to 2 mAP ppts. In particular, MD performs well, gaining around 2 ppts in the maximum mAP. When comparing the addition of GS_{full} to MC+E+MD, we still find slim improvements for thresholds ≥ 0.75. The score curve shows a kink at a threshold of 0.05 and ends at the same maximum mAP as GS_{full} while the confidence ranking is clearly improved for MC+E+MD and GS_{full}+MC+E+MD. Note that meta classification based on GS_{full} is less sensitive to the choice of threshold than the score in the medium range. At a threshold of 0.3 we have an mAP gap of about 1.4 ppts

Table 4. Meta classification and meta regression performance in terms of $AuROC$ and R^2, respectively, for different object detection architectures. Results (mean $\pm$ std) obtained from 10-fold cv as above.

	Pascal VOC		COCO		KITTI	
	$AuROC$	R^2	$AuROC$	R^2	$AuROC$	R^2
Faster R-CNN						
Score	89.77 ± 0.05	39.94 ± 0.02	83.82 ± 0.03	40.50 ± 0.01	96.53 ± 0.05	72.29 ± 0.02
MD	94.43 ± 0.02	47.92 ± 0.09	91.31 ± 0.02	44.41 ± 0.04	98.86 ± 0.02	79.92 ± 0.04
$\mathrm{GS}_{\mathrm{full}}$	$\mathbf{95.88 \pm 0.05}$	$\mathbf{59.40 \pm 0.03}$	$\mathbf{91.38 \pm 0.03}$	$\mathbf{50.44 \pm 0.04}$	$\mathbf{99.20 \pm 0.01}$	$\mathbf{86.31 \pm 0.07}$
$\mathrm{GS}_{\mathrm{full}}$ + MD	96.77 ± 0.05	63.64 ± 0.08	92.30 ± 0.02	52.30 ± 0.04	99.37 ± 0.02	87.46 ± 0.05
RetinaNet						
Score	87.53 ± 0.03	40.43 ± 0.01	84.95 ± 0.02	39.88 ± 0.02	95.91 ± 0.02	73.44 ± 0.02
MD	89.57 ± 0.04	50.27 ± 0.10	85.09 ± 0.01	42.45 ± 0.12	96.19 ± 0.02	77.53 ± 0.08
$\mathrm{GS}_{\mathrm{full}}$	$\mathbf{91.58 \pm 0.04}$	$\mathbf{57.23 \pm 0.07}$	$\mathbf{85.59 \pm 0.02}$	$\mathbf{47.74 \pm 0.06}$	$\mathbf{97.26 \pm 0.03}$	$\mathbf{84.47 \pm 0.04}$
$\mathrm{GS}_{\mathrm{full}}$ + MD	92.99 ± 0.03	64.32 ± 0.07	87.15 ± 0.05	51.07 ± 0.09	97.61 ± 0.02	85.73 ± 0.09
Cascade R-CNN						
Score	95.70 ± 0.04	57.90 ± 0.09	$\mathbf{94.11 \pm 0.01}$	56.31 ± 0.01	98.67 ± 0.02	83.31 ± 0.03
MD	96.32 ± 0.05	63.62 ± 0.12	94.10 ± 0.02	$\mathbf{58.74 \pm 0.08}$	99.18 ± 0.01	86.22 ± 0.08
$\mathrm{GS}_{\mathrm{full}}$	$\mathbf{96.66 \pm 0.05}$	$\mathbf{63.94 \pm 0.13}$	93.97 ± 0.01	57.80 ± 0.08	$\mathbf{99.34 \pm 0.01}$	$\mathbf{87.39 \pm 0.08}$
$\mathrm{GS}_{\mathrm{full}}$ + MD	97.24 ± 0.05	69.78 ± 0.13	94.78 ± 0.02	62.13 ± 0.06	99.48 ± 0.01	89.59 ± 0.04

Table 5. Computation timing of different methods at $\varepsilon_s = 10^{-4}$.

Method	Parameters	$AuROC$	AP	R^2	FPS
Score	—	96.53	96.53	78.86	**43.48**
MC	$N = 30$, par.	97.60	97.17	82.10	31.45
E	$N = 5$, seq.	<u>97.98</u>	<u>97.69</u>	<u>84.18</u>	9.17
$\mathrm{GS}_{\mathrm{full}}$	1 layer	**98.04**	**97.81**	**84.35**	<u>34.77</u>

which widens to 5.2 ppts at 0.6.

Runtime. We compare the runtime of our method with MC dropout and deep ensembles for YOLOv3 running on a Nvidia Quadro P6000 GPU at batch size 1. Table 5 shows the average performance on the KITTI dataset and throughput in frames per second (FPS). MC is batch-parallelized within dropout layers, while E runs sequentially. $\mathrm{GS}_{\mathrm{full}}$ is parallelized over predicted boxes and backpropagation performed explicitly by convolution (cf. section 4). We see that at slightly better meta classification, last layer gradient scores achieve around 3 additional FPS over MC which is in line with theorem 1. This is possible due to the initial score threshold on the prediction. Computing deeper gradients amounts to performing one more transposed convolution per layer which does not obstruct parallelism.

6. Conclusion

Applications of modern DNNs in safety-critical environments demand high performance on the one hand, but also reliable confidence estimation indicating where a model is not competent. We have proposed and investigated a way of implementing gradient-based uncertainty quantification for deep object detection which complements output-based methods well and is on par with established epistemic uncertainty quantification methods. Experiments involving a number of different architectures suggest that our method can be applied to significant benefit across architectures, even for high performance state-of-the-art models. We showed that meta classification performance carries over to object detection performance when employed as post-processing and that meta classification naturally leads to well-calibrated gradient confidences improving probabilistic reliability. Equation (6) can in principle be augmented to fit any DNN inferring and learning on an instance-based logic (*e.g.*, 3D bounding box detection, instance segmentation). Industrial applications of our method may include uncertainty-based querying in active learning or the probabilistic detection of data annotation errors. We hope that this work will inspire future progress in uncertainty quantification, probabilistic object detection and related areas.

Limitations. While our experiments indicate that gradient-based uncertainty can be used beneficially to estimate prediction quality and confidence, a comparison of gradient features in terms of OoD (or "open set condition") detections would also be of great interest and in line with previous work on gradient uncertainty [40, 54, 17]. However, the very definition of OoD in the instance-based setting itself is still subject of contemporary research [36, 6, 19] and lacks a widely established definition.

Acknowledgement. The research leading to these results is funded by the German Federal Ministry for Economic Affairs and Climate Action" within the projects "KI-Absicherung - Safe AI for Automated Driving", grant no. 19A19005R. We thank the consortium for the successful cooperation. We gratefully acknowledge financial support by the state Ministry of Economy, Innovation and Energy of Northrhine Westphalia (MWIDE) and the European Fund for Regional Development via the FIS.NRW project BIT-KI, grant no. EFRE-0400216. The authors gratefully acknowledge the Gauss Centre for Supercomputing e.V. for funding this project by providing computing time through the John von Neumann Institute for Computing on the GCS Supercomputer JUWELS at Jülich Supercomputing Centre.

References

[1] Zhaowei Cai and Nuno Vasconcelos. Cascade r-cnn: Delving into high quality object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6154–6162, 2018.

[2] Robin Chan, Matthias Rottmann, Fabian Hüger, Peter Schlicht, and Hanno Gottschalk. Metafusion: Controlled false-negative reduction of minority classes in semantic segmentation. *arXiv preprint arXiv:1912.07420*, 2019.

[3] Tianqi Chen and Carlos Guestrin. XGBoost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '16, pages 785–794, New York, NY, USA, 2016. ACM.

[4] Charles Corbière, Nicolas Thome, Avner Bar-Hen, Matthieu Cord, and Patrick Pérez. Addressing failure prediction by learning model confidence. In *33rd Conference on Neural Information Processing Systems (NeurIPS 2019)*, pages 2898–2909. Curran Associates, Inc., 2019.

[5] John S Denker and Yann LeCun. Transforming neural-net output levels to probability distributions. In *Proceedings of the 3rd International Conference on Neural Information Processing Systems*, pages 853–859, 1990.

[6] Akshay Dhamija, Manuel Gunther, Jonathan Ventura, and Terrance Boult. The overlooked elephant of object detection: Open set. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 1021–1030, 2020.

[7] John Duchi, Elad Hazan, and Yoram Singer. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of machine learning research*, 12(7), 2011.

[8] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The pascal visual object classes (voc) challenge. *International Journal of Computer Vision*, 88(2):303–338, June 2010.

[9] Yarin Gal. *Uncertainty in deep learning*. PhD thesis, University of Cambridge, 2016.

[10] 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.

[11] Andreas Geiger, P Lenz, Christoph Stiller, and Raquel Urtasun. The kitti vision benchmark suite. *URL http://www. cvlibs. net/datasets/kitti*, 2, 2015.

[12] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*, 2014.

[13] Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q Weinberger. On calibration of modern neural networks. In *International Conference on Machine Learning*, pages 1321–1330. PMLR, 2017.

[14] Ali Harakeh, Michael Smart, and Steven L Waslander. Bayesod: A bayesian approach for uncertainty estimation in deep object detectors. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 87–93. IEEE, 2020.

[15] 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.

[16] Dan Hendrycks and Kevin Gimpel. A baseline for detecting misclassified and out-of-distribution examples in neural networks. *arXiv preprint arXiv:1610.02136*, 2016.

[17] Rui Huang, Andrew Geng, and Yixuan Li. On the importance of gradients for detecting distributional shifts in the wild. *Advances in Neural Information Processing Systems*, 34, 2021.

[18] Eyke Hüllermeier and Willem Waegeman. Aleatoric and epistemic uncertainty in machine learning: An introduction to concepts and methods. *Machine Learning*, pages 1–50, 2021.

[19] KJ Joseph, Salman Khan, Fahad Shahbaz Khan, and Vineeth N Balasubramanian. Towards open world object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5830–5840, 2021.

[20] Alex Kendall and Yarin Gal. What uncertainties do we need in bayesian deep learning for computer vision? In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.

[21] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[22] Kamil Kowol., Matthias Rottmann., Stefan Bracke., and Hanno Gottschalk. Yodar: Uncertainty-based sensor fusion for vehicle detection with camera and radar sensors. In *Proceedings of the 13th International Conference on Agents and Artificial Intelligence - Volume 2: ICAART,*, pages 177–186. INSTICC, SciTePress, 2021.

[23] Florian Kraus and Klaus Dietmayer. Uncertainty estimation in one-stage object detection. In *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*, pages 53–60. IEEE, 2019.

[24] Fabian Kuppers, Jan Kronenberger, Amirhossein Shantia, and Anselm Haselhoff. Multivariate confidence calibration for object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 326–327, 2020.

[25] 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, page 6405–6416, Red Hook, NY, USA, 2017. Curran Associates Inc.

[26] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*, pages 2980–2988, 2017.

[27] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In

European conference on computer vision, pages 740–755. Springer, 2014.

[28] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. Ssd: Single shot multibox detector. In *European conference on computer vision*, pages 21–37. Springer, 2016.

[29] Weitang Liu, Xiaoyun Wang, John Owens, and Yixuan Li. Energy-based out-of-distribution detection. *Advances in Neural Information Processing Systems*, 33, 2020.

[30] Zongyao Lyu, Nolan Gutierrez, Aditya Rajguru, and William J Beksi. Probabilistic object detection via deep ensembles. In *European Conference on Computer Vision*, pages 67–75. Springer, 2020.

[31] 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.

[32] Kira Maag, Matthias Rottmann, Fabian Hüger, Peter Schlicht, and Hanno Gottschalk. Improving video instance segmentation by light-weight temporal uncertainty estimates. *arXiv preprint arXiv:2012.07504*, 2020.

[33] David JC MacKay. A practical bayesian framework for backpropagation networks. *Neural computation*, 4(3):448–472, 1992.

[34] Andrey Malinin and Mark Gales. Predictive uncertainty estimation via prior networks. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, pages 7047–7058, 2018.

[35] Dimity Miller, Feras Dayoub, Michael Milford, and Niko Sünderhauf. Evaluating merging strategies for sampling-based uncertainty techniques in object detection. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 2348–2354, 2019.

[36] Dimity Miller, Lachlan Nicholson, Feras Dayoub, and Niko Sünderhauf. Dropout sampling for robust object detection in open-set conditions. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3243–3249. IEEE, 2018.

[37] Dimity Miller, Niko Sünderhauf, Haoyang Zhang, David Hall, and Feras Dayoub. Benchmarking sampling-based probabilistic object detectors. In *CVPR Workshops*, volume 3, page 6, 2019.

[38] Mahdi Pakdaman Naeini, Gregory Cooper, and Milos Hauskrecht. Obtaining well calibrated probabilities using bayesian binning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 29, 2015.

[39] Lukas Neumann, Andrew Zisserman, and Andrea Vedaldi. Relaxed softmax: Efficient confidence auto-calibration for safe pedestrian detection. In *NIPS 2018 Workshop MLITS*, 2018.

[40] Philipp Oberdiek, Matthias Rottmann, and Hanno Gottschalk. Classification uncertainty of deep neural networks based on gradient information. In *IAPR Workshop on Artificial Neural Networks in Pattern Recognition*, pages 113–125. Springer, 2018.

[41] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d' Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc., 2019.

[42] Tiago Ramalho and Miguel Miranda. Density estimation in representation space to predict model uncertainty. In *International Workshop on Engineering Dependable and Secure Machine Learning Systems*, pages 84–96. Springer, 2020.

[43] Joseph Redmon and Ali Farhadi. Yolov3: An incremental improvement. *arXiv preprint arXiv:1804.02767*, 2018.

[44] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 28. Curran Associates, Inc., 2015.

[45] 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.

[46] 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 *Proceedings of the 23rd Conference on Design, Automation and Test in Europe*, DATE '20, page 1351–1356, San Jose, CA, USA, 2020. EDA Consortium.

[47] 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*, 2019.

[48] Marius Schubert, Karsten Kahl, and Matthias Rottmann. Metadetect: Uncertainty quantification and prediction quality estimates for object detection. *arXiv preprint arXiv:2010.01695*, 2020.

[49] Shai Shalev-Shwartz and Shai Ben-David. *Understanding machine learning: From theory to algorithms*. Cambridge university press, 2014.

[50] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1):1929–1958, 2014.

[51] Niclas Ståhl, Göran Falkman, Alexander Karlsson, and Gunnar Mathiason. Evaluation of uncertainty quantification in deep learning. In *International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems*, pages 556–568. Springer, 2020.

[52] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199*, 2013.

[53] Matias Valdenegro-Toro. Deep sub-ensembles for fast uncertainty estimation in image classification. *4th workshop on Bayesian Deep Learning (NeurIPS 2019)*, 2019.

[54] Vishal Thanvantri Vasudevan, Abhinav Sethy, and Alireza Roshan Ghias. Towards better confidence estimation for neural models. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 7335–7339. IEEE, 2019.

[55] Haoyu Wu. Yolov3-in-pytorch. `https://github.com/westerndigitalcorporation/YOLOv3-in-PyTorch`, 2018.

[56] Yuxin Wu, Alexander Kirillov, Francisco Massa, Wan-Yen Lo, and Ross Girshick. Detectron2. `https://github.com/facebookresearch/detectron2`, 2019.

[57] Hang Zhang, Chongruo Wu, Zhongyue Zhang, Yi Zhu, Zhi Zhang, Haibin Lin, Yue Sun, Tong He, Jonas Muller, R. Manmatha, Mu Li, and Alexander Smola. Resnest: Split-attention networks. *arXiv preprint arXiv:2004.08955*, 2020.