Gradient-Based Uncertainty
for Monocular Depth Estimation

Julia Hornauer[1]([✉]) and Vasileios Belagiannis[2]

[1] Institute of Measurement, Control and Microtechnology, Ulm University, Ulm,
Germany
`julia.hornauer@uni-ulm.de`
[2] Department of Simulation and Graphics, Otto von Guericke University,
Magdeburg, Germany

Abstract. In monocular depth estimation, disturbances in the image
context, like moving objects or reflecting materials, can easily lead to
erroneous predictions. For that reason, uncertainty estimates for each
pixel are necessary, in particular for safety-critical applications such as
automated driving. We propose a post hoc uncertainty estimation app-
roach for an already trained and thus fixed depth estimation model, rep-
resented by a deep neural network. The uncertainty is estimated with the
gradients which are extracted with an auxiliary loss function. To avoid
relying on ground-truth information for the loss definition, we present an
auxiliary loss function based on the correspondence of the depth predic-
tion for an image and its horizontally flipped counterpart. Our approach
achieves state-of-the-art uncertainty estimation results on the KITTI
and NYU Depth V2 benchmarks without the need to retrain the neural
network. Models and code are publicly available at https://github.com/
jhornauer/GrUMoDepth.

Keywords: Depth estimation · Uncertainty estimation · Training-free

1 Introduction

Deep neural networks have shown astonishing performance in 3D perception
tasks such as depth prediction [9,15]. Depth estimation from a single image has
particularly attracted attention because RGB cameras are cheaper compared to
LiDAR sensors while offering higher resolution and frame rates. Nevertheless,
disturbances in the image context, like occlusions, moving objects, or reflecting
materials, can easily affect the neural network and therefore lead to erroneous

---

Most of this work was done while Vasileios Belagiannis was with Ulm University.

predictions [20]. It is thus crucial to estimate the uncertainty of the depth estimates, especially for safety-critical applications such as automated driving.

Bootstrapped ensembles [16] and Monte Carlo Dropout [6], which are the well-known uncertainty estimation approaches, have been explored for computationally expensive tasks, such as depth estimation [22]. Both approaches rely on sampling from the model distribution, which is not suitable for real-time applications due to the high computational cost. Predictive methods, such as maximum likelihood maximization [13,14], on the other hand, require the adaptation of the training procedure to learn an estimate of the uncertainty contained in the data. Also, it is not always wished or even possible to retrain the neural network with the objective of uncertainty estimation. For instance, the parameters of models provided by external sources are usually not accessible and therefore not modifiable. The parameter modification is not feasible for neural network models that are either specialized to the target system (e.g., by pruning) or have to meet certain system requirements. For these reasons, we target the problem of uncertainty estimation of an already trained model in a *post hoc* manner.

One approach to training-free uncertainty estimation is the usage of dropout only during inference [18]. Although this approach does not require any adaptation of the training protocol, the uncertainty estimation is still based on sampling from the model distribution and is therefore not suitable for real-time applications. In contrast, we propose to estimate uncertainty based on gradients extracted from the neural network, which is much less computationally intensive and requires only one additional backward pass.

Inspired by gradient-based approaches for out-of-distribution detection in classification [11,17], we use gradients to estimate the uncertainty associated with pixel-wise depth, obtained from monocular depth estimation models. Given a trained model, we aim to extract meaningful gradients without relying on the ground-truth depth values. Therefore, we define an auxiliary loss function based on the correspondence of the depth prediction for an image and its horizontally flipped counterpart. With the loss defined as the error between both depth predictions, we calculate the derivative *w.r.t.* feature maps, by back-propagating through the neural network. We then rely on the gradients extracted from the feature maps of a decoder layer of the depth estimation model to obtain the final uncertainty score. Importantly, our approach achieves state-of-the-art uncertainty estimation results without the need to retrain the neural network.

Overall, we summarize the contributions of our paper as follows: Firstly, we propose a *post hoc* uncertainty estimation approach for depth prediction based on the gradients extracted from already trained models. Essentially, our approach is independent of how the model was trained. Secondly, for the gradient generation, we define an auxiliary loss function without relying on ground-truth depth values. In this context, we empirically show the meaningfulness of the gradient-based uncertainty estimation. Lastly, in an extensive comparison to existing approaches on the two common depth estimation benchmarks KITTI [7] and NYU Depth V2 [19], we demonstrate *state-of-the-art* uncertainty estimation results.

2   Related Work

Depth Estimation Uncertainty. Compared to classification tasks, it is more difficult to determine the uncertainty of high-dimensional predictions, such as in depth estimation. In general, a distinction can be made between epistemic and aleatoric uncertainty. While epistemic uncertainty arises from model weights and is reducible with more training data, aleatoric uncertainty is due to noise in the input data [13]. Bootstrapped ensembles [16] and Monte Carlo (MC) Dropout [6] estimate the epistemic uncertainty by modeling the distribution over parameters. For bootstrapped ensembles, this is achieved by training multiple models, sampling initial weights from a specified distribution, while for MC Dropout dropout layers are applied during training and inference. The depth and uncertainty estimates are obtained by sampling from the model distribution and computing the mean and variance, respectively. A predictive approach that accounts for aleatory uncertainty is to learn a distribution with mean and variance that represents the data-dependent error instead of a single output value by maximizing the negative log-likelihood [14]. Kendall and Gal [13] demonstrate how to combine both types of uncertainty. Recent works explore the integration of uncertainty for computationally intensive tasks such as depth estimation [4,28], semantic segmentation [10], optical flow [26], or multi-task learning [31]. Moreover, Poggi et al. [22] extensively compare different empirical and predictive uncertainty estimation approaches for self-supervised depth estimation. Among others, image flipping post-processing proposed by Godard et al. [8] serves as a simple baseline to obtain the variance over two outputs as an uncertainty measure. In addition, Poggi et al. [22] present their approach of self-training, where variance learned with log-likelihood maximization is improved by using knowledge obtained from a teacher model using its predictions as labels. One downside of those approaches is the adaptation of the model design [14] or the specific training pipeline [6,16]. Furthermore, empirical approaches have additional computational overhead [6,16] and increased memory footprint [16], making them unsuitable for real-time applications such as autonomous driving or robotics. By contrast, in this work, we explore the usage of gradients as a *post hoc* uncertainty estimation approach that is independent of the conducted training procedure. Since model re-training is not always feasible, Mi et al. [18] explore different training-free strategies to generate a distribution over the model output by data augmentation, inference-time dropout, and additive noise in intermediate network layers. We also explore training-free uncertainty estimation but forgo computationally intensive sampling by relying on gradients extracted with an auxiliary loss function.

Model Robustness by Gradient Analysis. Deep neural networks are mostly trained using gradient-based optimization. The informativeness of gradients for the task of out-of-distribution detection is explored in [11,17,21]. To obtain the gradients, a loss function must be defined before backpropagating through the model. Oberdiek et al. [21] use the negative log-likelihood at the predicted class since no labels are present during testing. Lee and AlRegib [17], on the other

hand, define confounding labels with only positive entries for the cross-entropy loss to show whether the model can associate features with any of the learned classes. Huang et al. [11] determine the KL divergence between the softmax output and the Uniform distribution, which indicates whether the predicted probabilities are distributed across all classes or concentrated in one class. Unlike, we use gradients of a regression neural network to determine the pixel-wise uncertainty for depth estimation models. More precisely, we need an uncertainty score for each input pixel and not just one score for the entire input image. For this purpose, we define the pixel-wise distance between the depth prediction and a reference depth obtained by image transformation as an auxiliary loss function.

Monocular Depth Estimation. The recent monocular depth estimation approaches train deep neural networks with supervision [1,2,5,15,23]. While Eigen et al. [5] use information from local and global features, Laina et al. [15] propose a fully convolutional deep neural network and Bauer et al. [1] integrate novel view synthesis. However, a large amount of data, which is time-consuming and expensive to obtain, is necessary for neural network training. Especially the generation of ground truth depth requires different well calibrated sensors. Therefore, self-supervised approaches leveraging monocular sequences [9,32], or stereo image pairs [8,29,30] are proposed. In the works of Godard et al. [8] as well as Yang et al. [29], learning depth from stereo image pairs by leveraging the scene geometry is introduced. In contrast, the use of monocular sequences for supervision by image reprojection [32], which is beneficial in practical applications, is more challenging due to the unknown scale and camera position that must be learned simultaneously with depth. Godard et al. [9] adapt the reprojection loss to handle occlusions and moving objects, while semantic or scale consistency is proposed by Toasi et al. [24] and Bian et al. [3], respectively. Xu et al. [27], on the other hand, introduce Region Deformer Networks to take moving objects into account. We propose an uncertainty estimation approach for monocular depth estimation models that are trained in a supervised or self-supervised manner. For the self-supervised case, the supervision can be provided by both monocular as well as stereo image pairs.

3   Method

Consider a deep neural network $\mathbf{d} = f(\mathbf{x}; \theta)$, parameterized by θ, that takes an image $\mathbf{x} \in \mathbb{R}^{w \times h \times 3}$ with width w and height h as input and predicts the pixel-wise depth $\mathbf{d} \in \mathbb{R}^{w \times h \times 1}$. Based on the trained depth prediction model f_θ, we aim to predict the uncertainty $\mathbf{u} \in \mathbb{R}^{w \times h \times 1}$ for each predicted depth value.

Post Hoc Uncertainty Estimation. In this work, we target *post hoc* uncertainty estimation. We do not adjust the model parameters, but only assume access to the internal model representation given by the feature maps $\mathbf{a}_i$, where i denotes the feature map obtained from the i-th layer of the depth decoder. Essentially, we estimate the uncertainty of the already trained depth estimation model and therefore do not need to adjust the training protocol or the network
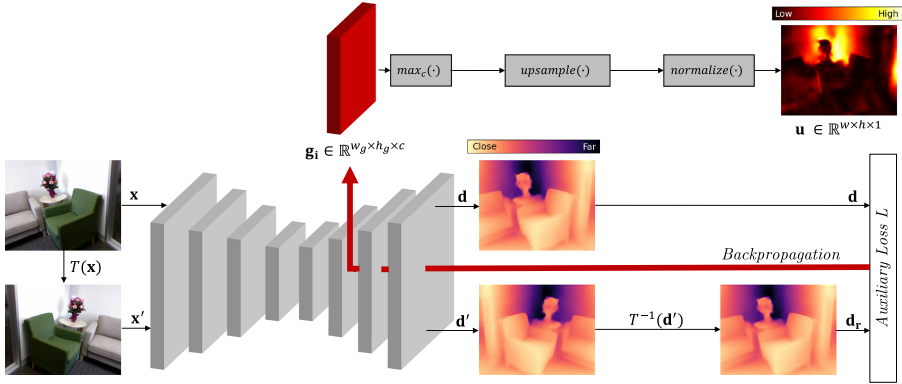
Fig. 1. First, the image $\mathbf{x}$ and its flipped counterpart $\mathbf{x}'$ are forwarded through the neural network. Then, the depth prediction $\mathbf{d}'$ of the flipped counterpart is flipped back to obtain the reference depth $\mathbf{d}_r$. Afterwards an auxiliary loss based on the two depth predictions is calculated to back-propagate through the neural network. Finally, the gradients $\mathbf{g}_i$ of the i-th intermediate layer are extracted. To determine the pixel-wise uncertainty score $\mathbf{u}$, the channel number is reduced from c to 1 by taking the maximum value of each pixel in $\mathbf{g}_i$ in the channel dimension. Then, the gradient map is upsampled with nearest neighbor interpolation to match the depth map resolution.

design. Moreover, our approach is independent of the underlying model training strategy. For instance, the depth estimation model may have been trained in a supervised or self-supervised manner with monocular or stereo supervision. Figure 1 illustrates an overview of our method.

3.1 Gradient-Based Uncertainty

Gradient Generation. We propose to use gradients extracted from the feature maps $\mathbf{a}_i$ to estimate the pixel-wise uncertainty of depth predictions. For the gradient generation, the pixel-level loss function must be defined first, followed by the derivative of the loss function $w.r.t$ the respective feature map using backpropagation. Our goal is to convert the model loss to pixel-wise uncertainty estimates. We assume that the pixel-wise uncertainty is in accordance with the depth estimation error. We later empirically show our claims in our evaluations.

In general, the most meaningful loss function is the error between the depth prediction $\mathbf{d}$ and the ground-truth depth $\mathbf{y} \in \mathbb{R}^{w \times h \times 1}$. Nevertheless, we have access to ground-truth information during training, but not during inference. Therefore, we define an auxiliary loss function $\mathcal{L}$ for the gradient generation. For the definition of this loss function, we consider the reference image $\mathbf{x}'$ whose structure equals the structure of the original image $\mathbf{x}$. We make the assumption that the depth maps $\mathbf{d}$ and $\mathbf{d}'$ predicted for $\mathbf{x}$ and $\mathbf{x}'$ match. To generate the reference image $\mathbf{x}'$, we define the horizontal flip operation $T(\cdot)$, which performs a left-right flip. The inverse function $T^{-1}(\cdot)$ reverts the flip operation. The depth prediction on an image $\mathbf{x}$ and the horizontally flipped version $\mathbf{x}' = T(\mathbf{x})$ should

be consistent, since the linear transformation preserves the pixel information of the original image and thus the structure of the scene.

Overall, the reference depth $\mathbf{d}_r$ is the depth $\mathbf{d}'$ predicted for the flipped image $\mathbf{x}'$, which is flipped back to match the depth $\mathbf{d}$ obtained for the original input image $\mathbf{x}$. More precisely, we conduct two forward passes, one with the original image $\mathbf{x}$ to obtain $\mathbf{d} = f(\mathbf{x}; \theta)$ and one with the flipped version $\mathbf{x}' = T(\mathbf{x})$ to obtain $\mathbf{d}' = f(\mathbf{x}'; \theta)$. Then, we apply the inverse function $T^{-1}(\mathbf{d}')$ to obtain the reference depth $\mathbf{d}_r$. The auxiliary loss function is then defined based on the difference between the depth $\mathbf{d}$ and the reference depth $\mathbf{d}_r$ (see Sec. 3.2). To obtain the gradients, we calculate the derivative of the loss $w.r.t$ the respective feature map $\mathbf{a}_i$:

$$\mathbf{g}_i = \frac{\partial \mathcal{L}}{\partial \mathbf{a}_i}, \tag{1}$$

where $\mathbf{g}_i \in \mathbb{R}^{w_g \times h_g \times c}$ with w_g, h_g and c being the feature map width, feature map height and the channel number, respectively.

Uncertainty Score. Next, the pixel-wise uncertainty estimate $\mathbf{u}$ is determined from the gradients $\mathbf{g}_i$ extracted from the neural network. Since an uncertainty value is required for each pixel, the gradient map must match the resolution of the depth map. First, we define the function $max_c(\cdot) : \mathbb{R}^{w_g \times h_g \times c} \rightarrow \mathbb{R}^{w_g \times h_g \times 1}$ to perform max-pooling over the channel dimensions:

$$\mathbf{g}_i^{(\mathrm{max})} = max_c(\mathbf{g}_i), \tag{2}$$

where $\mathbf{g}_i^{(\mathrm{max})} \in \mathbb{R}^{w_g \times h_g \times 1}$. We are interested in gathering the gradients with the largest magnitude for the uncertainty estimation using the max-pooling operation. Afterward, we define the function $upsample(\cdot) : \mathbb{R}^{w_g \times h_g \times 1} \rightarrow \mathbb{R}^{w \times h \times 1}$ to upsample $\mathbf{g}_i^{(\mathrm{max})}$ by nearest-neighbor interpolation. Finally, we propose the uncertainty estimate as the self-normalized gradient map:

$$\mathbf{u} = \frac{upsample(\mathbf{g}_i^{(\mathrm{max})}) - \min \mathbf{g}_i^{(\mathrm{max})}}{\max \mathbf{g}_i^{(\mathrm{max})} - \min \mathbf{g}_i^{(\mathrm{max})}}, \tag{3}$$

where $\max \mathbf{g}_i^{(\mathrm{max})}$ and $\min \mathbf{g}_i^{(\mathrm{max})}$ are the maximum and the minimum value of the gradient map, respectively. The final uncertainty is then normalized to a range of $[0, 1]$.

3.2   Training Strategy

We instantiate the auxiliary loss function for gradient generation based on the predicted depth $\mathbf{d}$ and the reference depth $\mathbf{d}_r$. First, we define our loss function for depth estimation models that only predict the depth $\mathbf{d} = f_r(\mathbf{x}; \theta_r)$. Then, we consider a Bayesian depth estimation model $\mathbf{d}, \sigma = f_b(\mathbf{x}; \theta_b)$ with $\mathbf{d}$ being the depth and σ the variance as a measure of uncertainty.

Depth Estimation Model. For standard depth estimation models f_{θ_r}, that do not predict the uncertainty but only the depth, the auxiliary loss function is defined as follows:

$$\mathcal{L}_r = (\mathbf{d} - \mathbf{d}_r)^2. \tag{4}$$

An uncertainty estimate can be generated by a post-processing step in which image flipping is applied to estimate the uncertainty as the variance over two outputs (*Post*) [9]. Here, the variance is also the pixel-wise difference of the two depth predictions. We claim that the information given by the gradients of the feature maps is higher compared to the uncertainty solely given by the loss. This is supported by the results in Sect. 4.2, where we empirically demonstrate that the information given by the gradients of the feature maps is more effective for uncertainty estimation compared to simply relying on image flipping as a post-processing step.

Bayesian Depth Estimation Model. Since our approach is independent of the underlying training strategy, we do not only consider conventional depth estimation models but also the Bayesian version f_{θ_b} that already predicts the variance as an uncertainty measure. Different strategies are proposed to train Bayesian models. One is training with log-likelihood maximization objective (*Log*), while another is the self-teaching paradigm (*Self*) as proposed by Poggi et al. [22]. For Bayesian depth prediction networks, we aim to improve the uncertainty estimate given by the variance. Since these models do not only output the pixel-wise depth $\mathbf{d}$ but also the pixel-wise variance σ, we also make use of the variance for the gradient generation. In this context, we add the squared variance to the loss term. The auxiliary loss for the Bayesian model is given in the following:

$$\mathcal{L}_b = \mathcal{L}_r + \lambda\sigma^2, \tag{5}$$

where λ controls the influence of the variance loss term. Since σ represents the error inherent in the data, high variance leads to larger gradients, while small variance leads to small gradients.

4   Experiments

We evaluate our approach on two standard monocular depth estimation benchmarks. First, we describe the experimental setup including datasets, models, metrics, and implementation. Then, we demonstrate our uncertainty estimation results and provide visual illustrations. Ultimately, we demonstrate the informativeness of the gradient space when ground truth is given for the loss computation and validate horizontal flipping as the chosen image transformation.

4.1   Experimental Setup

Datasets. Since there is no other work that performs a detailed analysis of different approaches to uncertainty estimation using a well-defined protocol, we follow the same evaluation as proposed by Poggi et al. [22]. Therefore, we evaluate

our approach on the KITTI [7] dataset. KITTI is an autonomous driving dataset taken at 61 scenes with an average image resolution of 375×1242. Similar to Poggi et al. [22], we use the Eigen split [5] with maximum depth set to 80 m and use the improved ground truth depth provided in [25] for evaluation. In addition, we propose to evaluate the uncertainty estimation on the NYU Depth V2 dataset. This is an indoor depth estimation dataset taken at 494 scenes. The original image resolution is 480×640. Here, we adapt the evaluation protocol proposed by Poggi et al. [22] but set the maximum depth to 10 m.

Models. We choose Monodepth2 [9] as the depth estimation model to make a fair comparison to the numerous uncertainty estimation approaches in [22]. Similar to Poggi et al. [22], we use both Monodepth2 trained with monocular and stereo supervision. Importantly, networks trained with monocular sequences are not only subject to depth uncertainty but also the uncertainty of the estimated camera pose. For NYU Depth V2 [19] we train Monodepth2 [9] in a supervised manner with the provided ground truth depth maps.

Evaluation Metrics. Similar to prior work [12,22], our goal is to estimate the uncertainty with respect to the prediction error. Therefore, we evaluate the estimated uncertainty on in-distribution data by *sparsification* plots that measure if the uncertainty agrees with the true error. Given a data sample and an error metric, the uncertainty is ranked in descending order. Then, the most uncertain pixels are removed to calculated the error metric for the remaining pixels to obtain the sparsification over the fraction of removed pixels. As in [12,22], we compute the Area Under the Sparsification Error (AUSE) that is the difference between the sparsification and the oracle sparsification. The oracle sparsification is given if the uncertainty ranking corresponds to the ranking of the true error. We evaluate the AUSE in terms of the depth estimation metrics Absolute Relative Error (Abs Rel), Root Mean Squared Error (RMSE), and Accuracy ($\delta \geq 1.25$). In addition, as proposed by Poggi et al. [22], we compute the Area Under the Random Gain (AURG) that considers the difference between the sparsification and the random sparsification that quantifies whether the estimated uncertainty is better than not modeling the uncertainty. Again, we calculate the AURG in terms of Abs Rel, RMSE, and $\delta \geq 1.25$.

Comparison to Related Work. Since our approach is a *post hoc* uncertainty estimation method, we apply our approach to already trained depth estimation models. As described in Sect. 3.2, the base models (Base) are a standard depth estimation model with post-processing applied (*Post*) [9], and the two Bayesian depth estimation models *Log* [22] and *Self* [22], that also predict the variance. We compare our method to the uncertainties obtained from the base models themselves as well as inference only dropout (*In-Drop*) [18], MC Dropout (*Drop*) [6] and bootstrapped ensembles (*Boot*) [16]. Note that *In-Drop* is also applied to the base models as a training-free approach, while *Drop* and *Boot* are separate models. Furthermore, we compare our approach to the uncertainty obtained with the variance over different test-time augmentations (*Var*) as additional *post hoc* baseline.

Implementation Details. All base models are implemented based on Monodepth2 [9] as depth estimation network. Since Poggi et al. [22] provide trained models for the uncertainly estimation baselines *Post*, *Log*, *Self*, *Boot*, and *Drop* trained with monocular as well as stereo supervision, we evaluate our model with the provided weights for a fair comparison. For *In-Drop* we apply dropout with probability 0.2 at the same locations as used for *Drop* but only during inference. As in [22], we perform 8 forward passes for *Drop* and *In-Drop*. For NYU Depth V2 [19], we train Monodepth2 [9] as the base model with random rotation, random scaling, horizontal flipping, color jittering and cropped to a final resolution of 224×288. For this setup, we additionally implement the Bayesian model trained with log-likelihood maximization (*Log*) as well as MC Dropout (*Drop*) next to the standard depth estimation network. Moreover, *In-Drop* is implemented as for KITTI [7]. For *Var*, we apply grayscale, flipping, noise and rotation as augmentations to calculate the variance over the resulting outputs. For all setups, we choose the 6th layer of the decoder to extract the gradients for our gradient-based uncertainty estimation approach. Results with gradients extracted from different layers are provided in the supplementary material. In the case of the Bayesian depth estimation networks, we choose λ to be 2.0.

4.2   Uncertainty Estimation Results

KITTI Monocular Supervision. The uncertainty estimation results for Monodepth2 trained with monocular sequences from KITTI are reported in Tab. 1. For the monocular setup, where the uncertainty stems not only from the depth estimation model but also from the pose prediction, our method outperforms the base models as well as *In-Drop*. Especially, AUSE and AURG in terms of RMSE are clearly improved compared to the base models *Post* [9], *Log* [22], and *Self* [22]. Moreover, all methods obtain better uncertainty estimation results compared to the two empirical methods *Drop* [6] and *Boot* [16]. In terms of inference time, the base methods are slightly faster, but our gradient-based uncertainty estimation achieves better uncertainty estimation results in comparison. *In-Drop*, on the other hand, has significantly higher inference times due to the sampling procedure and does not always outperform the baseline model when considering AUSE and AURG in terms of Abs Rel and $\delta \geq 1.25$.

KITTI Stereo Supervision. In Table 2, the uncertainty estimation results for Monodepth2 trained with stereo pair supervision, where the uncertainty results only from the depth estimation model, are listed. In this setup, the worst results are obtained from the computationally expensive MC Dropout [6]. For *Post* [9] and *Log* [22], the gradient-based uncertainty estimation especially improves over the base model in terms of AUSE and AURG for Abs Rel and RMSE. Moreover, our method is on par with the *Self* [22] model while being a *post hoc* approach. One downside of self-teaching is the elaborate training procedure that is required to obtain the final depth estimation model. Furthermore, the training-free *In-Drop* [18] shows no improvement over the base models while requiring a significantly higher inference time. Furthermore, *In-Drop* [18] is barely better than random chance when observing AURG.

Table 1. Uncertainty evaluation results for Monodepth2 [9] trained with monocular supervision on KITTI [7].

Model	Method	Abs Rel		RMSE		$\delta \geq 1.25$		Inf [ms]
		AUSE ↓	AURG ↑	AUSE ↓	AURG ↑	AUSE ↓	AURG ↑	Inf [ms]
Drop	Base [22]	0.056	0.000	2.568	0.944	0.097	0.002	47.12
Boot	Base [22]	0.058	0.001	3.982	−0.743	0.084	−0.001	47.12
Post	Base [9]	0.044	0.012	2.864	0.412	0.056	0.022	11.78
	Var	0.055	0.003	3.575	−0.210	0.074	0.006	29.45
	In-Drop [18]	0.031	0.027	0.871	2.495	0.029	0.051	53.01
	Ours	**0.029**	**0.029**	**0.540**	**2.825**	**0.025**	**0.054**	18.84
Log	Base [22]	0.039	0.020	2.562	0.916	0.044	0.038	5.89
	Var	0.055	0.004	3.686	−0.209	0.075	0.008	29.45
	In-Drop [18]	0.045	0.014	1.916	1.561	0.056	0.027	53.01
	Ours	**0.026**	**0.033**	**0.819**	**2.658**	**0.024**	**0.059**	18.84
Self	Base [22]	0.030	0.026	2.009	1.266	0.030	0.045	5.89
	Var	0.055	0.001	3.632	−0.357	0.073	0.002	29.45
	In-Drop [18]	0.033	0.024	1.091	2.184	0.031	0.044	53.01
	Ours	**0.024**	**0.032**	**0.494**	**2.780**	**0.017**	**0.057**	18.84

Table 2. Uncertainty evaluation results for Monodepth2 [9] trained with stereo pair supervision on KITTI [7].

Model	Method	Abs Rel		RMSE		$\delta \geq 1.25$		Inf [ms]
		AUSE ↓	AURG ↑	AUSE ↓	AURG ↑	AUSE ↓	AURG ↑	
Drop	Base [22]	0.103	−0.029	6.163	−2.169	0.231	−0.080	47.12
Boot	Base [22]	0.028	0.029	2.291	0.964	0.031	0.048	47.12
Post	Base [9]	0.036	0.020	2.523	0.736	0.044	0.034	11.78
	Var	0.054	0.003	3.672	−0.327	0.074	0.006	29.45
	In-Drop [18]	0.061	−0.004	3.366	−0.021	0.089	−0.009	53.01
	Ours	**0.022**	**0.035**	**0.510**	**2.835**	**0.023**	**0.057**	18.84
Log	Base [22]	0.022	0.036	0.938	2.402	**0.018**	**0.061**	5.89
	Var	0.054	0.003	3.639	−0.299	0.073	0.006	29.45
	In-Drop [18]	0.073	−0.016	4.063	−0.724	0.107	−0.028	53.01
	Ours	**0.019**	**0.038**	**0.490**	**2.849**	**0.018**	**0.061**	18.84
Self	Base [22]	**0.022**	**0.035**	1.679	1.642	**0.022**	**0.056**	5.89
	Var	0.056	0.001	3.728	−0.408	0.076	0.002	29.45
	In-Drop [18]	0.076	−0.019	4.173	−0.852	0.109	−0.031	53.01
	Ours	**0.022**	**0.035**	**0.515**	**2.806**	0.023	0.055	18.84

Table 3. Uncertainty evaluation results for Monodepth2 [9] trained on NYU Depth V2 [19].

Model	Method	Abs Rel		RMSE		$\delta \geq 1.25$		Inf [ms]
		AUSE ↓	AURG ↑	AUSE ↓	AURG ↑	AUSE ↓	AURG ↑	
Drop	Base [6]	0.086	0.002	0.214	0.160	0.169	−0.001	26.32
Post	Base [9]	0.066	0.018	0.267	0.096	0.116	0.037	6.58
	Var	0.063	0.024	0.221	0.151	0.110	0.045	16.45
	In-Drop [18]	0.086	0.000	**0.223**	**0.149**	0.163	0.000	29.61
	Ours	**0.061**	**0.025**	0.252	0.120	**0.106**	**0.048**	12.05
Log	Base [22]	0.055	0.030	**0.159**	**0.210**	0.089	0.066	3.29
	Var	0.061	0.024	0.220	0.149	0.106	0.049	16.45
	In-Drop [18]	0.087	−0.002	0.215	0.154	0.166	−0.012	29.61
	Ours	**0.053**	**0.032**	0.176	0.193	**0.086**	**0.069**	12.05

NYU Depth V2. The uncertainty estimation performance of Monodepth2 trained on NYU Depth V2 is reported in Table 3. Again, the uncertainty only stems from the depth estimation itself due to the supervised training. We implement our approach on the *Post* [9] and *Log* [22] models and state the MC Dropout [6] model as baseline. As in the previous setups, uncertainty estimation with dropout sampling results in the worst uncertainty estimation performance. Here, both versions, with dropout during training as well as only during inference, do obtain worse uncertainty estimates compared to the base models in almost all metrics. By contrast, our gradient-based method does improve the performance for AUSE and ARUG in terms of Abs Rel as well as $\delta \geq 1.25$.

Table 4. Comparison of the different uncertainty estimation methods regarding number of trained models (#Train), whether a specific training strategy is required (Specialized Training), number of models for uncertainty estimation (#Models), number of forward passed required to obtain depth and uncertainty (#Forward) and number of backward passes (#Backward). Note that N is set to 8.

Method	#Train	Specialized Training	#Models	#Forward	#Backward
Drop [6]	1	Yes	1	N	–
Boot [16]	N	Yes	N	N	–
Post [9]	1	No	1	2	–
Log [22]	1	Yes	1	1	–
Self [22]	2	Yes	1	1	–
Var	–	No	–	4 + 1	–
In-Drop [18]	–	No	–	N + 1	–
Ours	–	No	–	2	1

Sparsification Error Plots. In Fig. 2, we illustrate the sparsification error curves in terms of RMSE for KITTI with monocular supervision (Fig. 2a) and with stereo pair supervision (Fig. 2b) averaged over the KITTI test set. The sparsification error represents the deviation of the sparsification curve from the respective oracle sparsification. Therefore, a smaller area under the curve means better performance in uncertainty estimation. For our *post hoc* approach, the uncertainty best matches the true error, while *In-Drop* [18] does not improve on the baseline model for the stereo setup. In the monocular setup, *Boot* [16] performs the worst, while the uncertainty modeled with *Drop* [6] in the stereo case is the least consistent with the true error.

Summary. Overall, our method improves the uncertainty estimates for most of the cases over the base model while only adding a slight overhead in inference time. The empirical approaches *Boot* [16], *Drop* [6], and *In-Drop* [18], on the other hand, add a huge computational overhead and therefore require inference times that are not feasible in real-time applications such as autonomous driving or robotics. In addition, these methods do not outperform the uncertainty
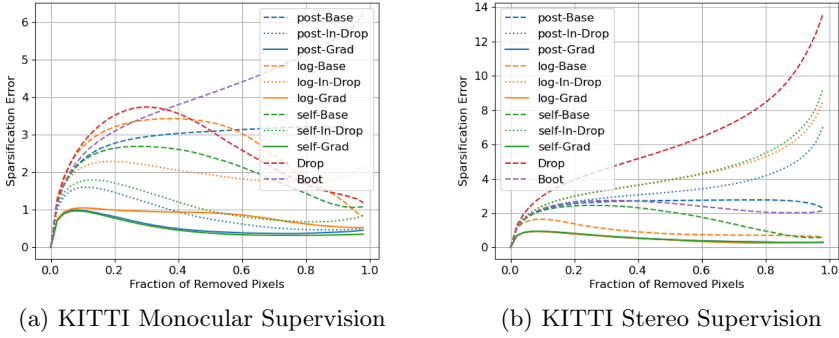
Fig. 2. The sparsification error in terms of RMSE over the fraction of removed pixels is shown for Monodepth2 [9] trained on KITTI [7] with monocular (a) and stereo (b) supervision.

of the *Post* [9], *Log* [22], and *Self* [22] models. The comparison of our method with the *Post* baseline highlights the higher significance of gradients extracted from the model compared to using image flipping alone as the difference in pixel space. In Tab. 4, the main characteristics of the different methods are compared regarding the number of trained models (#Train), the adaptation of the training strategy (Specialized Training), the number of models that must be saved (#Models), the number of forward passes necessary to obtain the depth as well as the uncertainty estimate (#Forward) and the number of required backward passes (# Backward). In general, only *In-Drop*, *Var* and our method are *post hoc* approaches applicable to all kinds of depth estimation networks. Moreover, all approaches besides post-processing require an adaptation to the training strategy or the network architecture. While the empirical methods result in a large computational overhead, our approach only needs one additional backward pass to obtain the gradients. Moreover, empirical methods are known to decrease the depth estimation performance [22]. When comparing our approach with the other training-free methods *In-Drop* [18] and *Var*, these approaches need one forward pass to obtain the depth prediction and N or 4 (depending on the number of augmentations) forward passes to get the uncertainty estimate, while our approach only needs two forward and one backward pass to obtain depth and uncertainty estimates.

Visual Results. In Fig. 3, an example from Monodepth2 trained with the standard depth estimation protocol on NYU Depth V2 is illustrated. Overall, the estimated uncertainty for high error regions should be high, while the uncertainty for low error regions should be low. When comparing the depth prediction (Fig. 3b) and the RMSE (Fig. 3c), it becomes apparent that the region around the two lamps has the greatest error. Both, the post-processing [9] (Fig. 3d) and our gradient-based method (Fig. 3f) highlight the high uncertainty at the lamp. Moreover, our approach improves over the uncertainty estimation by post-processing in regions with low error. *In-Drop* [18] (Fig. 3e), on the other hand,

overestimates uncertainty in most regions, especially in the lower right corner. More visual results can be found in the supplementary material.
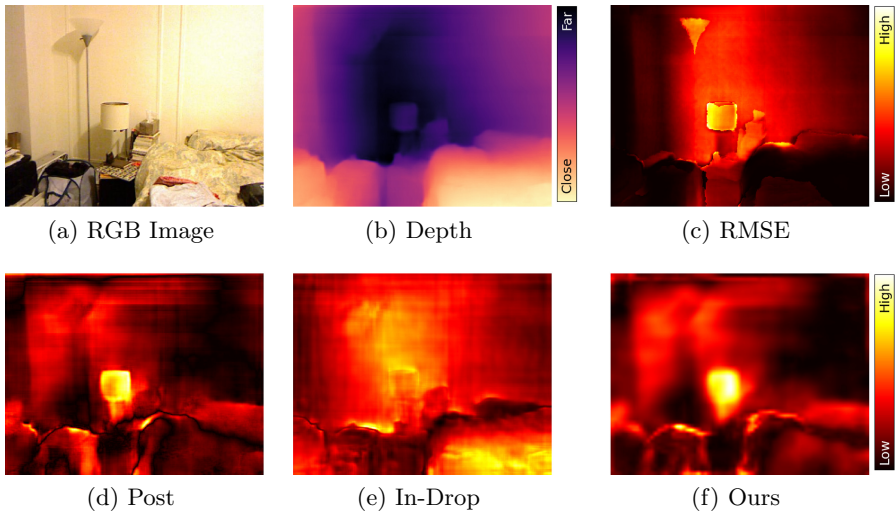

(a) RGB Image   (b) Depth   (c) RMSE

(d) Post   (e) In-Drop   (f) Ours

Fig. 3. Uncertainty estimation example from Monodepth2 [9] trained on NYU Depth V2 [19]. In (a), the input image is shown. The depth prediction (b) and the root mean squared error (c) highlight that the lamp and the region around it are not estimated correctly. The second row demonstrates the uncertainty estimated by post-processing (d), inference dropout (e) and gradients (f).

4.3   Ablation Study

In Table 5, the uncertainty estimation results of our gradient-based approach on Monodepth2 trained with NYU Depth V2 are reported for the *Post* model. We demonstrate different configurations to define the loss for the gradient generation. We consider the squared difference of the prediction to the ground truth depth (GT) and to transformed images by image flipping (Flip), gray-scale conversion (Gray), additive Gaussian noise (Noise) and rotation (Rot) with different rotation angles. Note that the selected augmentations are not necessarily within the training time augmentation. The uncertainty estimation results for all three depth estimation metrics clearly show that the loss between the prediction and the ground truth depth results in the best uncertainty estimation performance. Nevertheless, the ground truth is not available during inference in real-world application. However, results in Sect. 4.2 demonstrate promising results for uncertainty estimation based on the information given by the extracted gradients. Comparing the different transformation operations, the flipping operation performs best. Compared to the other augmentations considered, when flipping

horizontally, the model observes the scene from a different context, while preserving the geometry of the scene. In case of the rotation operation, the uncertainty cannot be determined for each pixel. This confirms the use of the flipping operation to create a reference depth.

Table 5. Uncertainty estimation results for Monodepth2 [9] Post trained on NYU Depth V2 [19] when using different loss functions for the gradient generation. We compare the error of the prediction to the ground truth depth (GT) and depth predictions obtained by different image transformations. We consider image flipping (Flip), grayscale conversion (Gray), additive Gaussian noise (Noise) and rotation (Rot-[angle]) with different rotation angles.

	Abs Rel		RMSE		$\delta \geq 1.25$	
Loss	AUSE ↓	AURG ↑	AUSE ↓	AURG ↑	AUSE ↓	AURG ↑
GT	**0.020**	**0.066**	**0.106**	**0.266**	**0.031**	**0.123**
Flip	<u>0.061</u>	<u>0.025</u>	<u>0.252</u>	<u>0.120</u>	<u>0.106</u>	<u>0.048</u>
Gray	0.067	0.020	0.253	0.119	0.111	0.044
Noise	0.066	0.020	0.261	0.111	0.108	0.046
Rot-5°	0.066	0.020	0.267	0.105	0.114	0.041
Rot-10°	0.069	0.017	0.269	0.103	0.124	0.031
Rot-20°	0.076	0.011	0.270	0.102	0.143	0.012

5   Conclusions

We present the usage of gradients for uncertainty estimation of depth predictions. Given an already trained model, we introduce an auxiliary loss function, that is independent of ground truth depth, based on the correspondence of the depth prediction for an image and its horizontally flipped counterpart. Note that our method does not adjust the neural network weights but estimates the uncertainty in a *post hoc* manner. With the loss defined as the error between the depth predictions, we calculate the derivative *w.r.t.* feature maps by backpropagation. Finally, we determine the pixel-wise uncertainty based on the gradients of an inter-mediate feature map extracted from the depth decoder. We performed a comprehensive comparison with related work using two depth estimation benchmarks, namely the KITTI and NYU Depth V2 datasets. Our evaluation shows the high potential to estimate uncertainty based on gradients extracted from an already trained model.

References

1. Bauer, Z., Li, Z., Orts-Escolano, S., Cazorla, M., Pollefeys, M., Oswald, M.R.: NVS-monodepth: improving monocular depth prediction with novel view synthesis. In: 2021 International Conference on 3D Vision (3DV), pp. 848–858 (2021)

2. Bhat, S., Alhashim, I., Wonka, P.: Adabins: depth estimation using adaptive bins. In: 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 4008–4017 (2021)

3. Bian, J., et al.: Unsupervised scale-consistent depth and ego-motion learning from monocular video. In: NeurIPS (2019)

4. Chanduri, S.S., Suri, Z.K., Vozniak, I., Müller, C.: Camlessmonodepth: monocular depth estimation with unknown camera parameters. arXiv:abs/2110.14347 (2021)

5. Eigen, D., Puhrsch, C., Fergus, R.: Depth map prediction from a single image using a multi-scale deep network. In: NIPS (2014)

6. Gal, Y., Ghahramani, Z.: Dropout as a bayesian approximation: representing model uncertainty in deep learning. In: Proceedings of The 33rd International Conference on Machine Learning, June 2015

7. Geiger, A., Lenz, P., Stiller, C., Urtasun, R.: Vision meets robotics: the kitti dataset. In: International Journal of Robotics Research (IJRR) (2013)

8. Godard, C., Aodha, O.M., Brostow, G.J.: Unsupervised monocular depth estimation with left-right consistency. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 6602–6611 (2017)

9. Godard, C., Aodha, O.M., Brostow, G.J.: Digging into self-supervised monocular depth estimation. In: 2019 IEEE/CVF International Conference on Computer Vision (ICCV), pp. 3827–3837 (2019)

10. Gustafsson, F.K., Danelljan, M., Schön, T.B.: Evaluating scalable Bayesian deep learning methods for robust computer vision. In: 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), pp. 1289–1298 (2020)

11. Huang, R., Geng, A., Li, Y.: On the importance of gradients for detecting distributional shifts in the wild. arXiv:abs/2110.00218 (2021)

12. Ilg, E., et al.: Uncertainty estimates and multi-hypotheses networks for optical flow. In: Ferrari, V., Hebert, M., Sminchisescu, C., Weiss, Y. (eds.) ECCV 2018. LNCS, vol. 11211, pp. 677–693. Springer, Cham (2018). https://doi.org/10.1007/978-3-030-01234-2_40

13. Kendall, A., Gal, Y.: What uncertainties do we need in Bayesian deep learning for computer vision? In: NIPS (2017)

14. Klodt, M., Vedaldi, A.: Supervising the new with the old: learning SFM from SFM. In: Ferrari, V., Hebert, M., Sminchisescu, C., Weiss, Y. (eds.) ECCV 2018. LNCS, vol. 11214, pp. 713–728. Springer, Cham (2018). https://doi.org/10.1007/978-3-030-01249-6_43

15. Laina, I., Rupprecht, C., Belagiannis, V., Tombari, F., Navab, N.: Deeper depth prediction with fully convolutional residual networks. In: 2016 Fourth International Conference on 3D Vision (3DV), pp. 239–248 (2016)

16. Lakshminarayanan, B., Pritzel, A., Blundell, C.: Simple and scalable predictive uncertainty estimation using deep ensembles. In: NIPS (2017)

17. Lee, J., AlRegib, G.: Gradients as a measure of uncertainty in neural networks. In: 2020 IEEE International Conference on Image Processing (ICIP), pp. 2416–2420 (2020)

18. Mi, L., Wang, H., Tian, Y., He, H., Shavit, N.: Training-free uncertainty estimation for dense regression: sensitivity as a surrogate (2019)
19. Silberman, N., Hoiem, D., Kohli, P., Fergus, R.: Indoor segmentation and support inference from RGBD images. In: Fitzgibbon, A., Lazebnik, S., Perona, P., Sato, Y., Schmid, C. (eds.) ECCV 2012. LNCS, vol. 7576, pp. 746–760. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-33715-4_54
20. Nguyen, A., Yosinski, J., Clune, J.: Deep neural networks are easily fooled: High confidence predictions for unrecognizable images. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), June 2015
21. Oberdiek, P., Rottmann, M., Gottschalk, H.: Classification uncertainty of deep neural networks based on gradient information. In: ANNPR (2018)
22. Poggi, M., Aleotti, F., Tosi, F., Mattoccia, S.: On the uncertainty of self-supervised monocular depth estimation. In: 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 3224–3234 (2020)
23. Song, M., Lim, S., Kim, W.: Monocular depth estimation using laplacian pyramid-based depth residuals. IEEE Trans. Circuits Syst. Video Technol. **31**, 4381–4393 (2021)
24. Tosi, F., Aleotti, F., Poggi, M., Mattoccia, S.: Learning monocular depth estimation infusing traditional stereo knowledge. In: 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 9791–9801 (2019)
25. Uhrig, J., Schneider, N., Schneider, L., Franke, U., Brox, T., Geiger, A.: Sparsity invariant CNNs. In: 2017 International Conference on 3D Vision (3DV), pp. 11–20 (2017)
26. Wannenwetsch, A.S., Keuper, M., Roth, S.: Probflow: joint optical flow and uncertainty estimation. In: 2017 IEEE International Conference on Computer Vision (ICCV), pp. 1182–1191 (2017)
27. Xu, H., Zheng, J., Cai, J., Zhang, J.: Region deformer networks for unsupervised depth estimation from unconstrained monocular videos. In: IJCAI (2019)
28. Yang, N., von Stumberg, L., Wang, R., Cremers, D.: D3vo: deep depth, deep pose and deep uncertainty for monocular visual odometry. In: 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1278–1289 (2020)
29. Yang, N., Wang, R., Stückler, J., Cremers, D.: Deep virtual stereo odometry: leveraging deep depth prediction for monocular direct sparse odometry. In: Ferrari, V., Hebert, M., Sminchisescu, C., Weiss, Y. (eds.) ECCV 2018. LNCS, vol. 11212, pp. 835–852. Springer, Cham (2018). https://doi.org/10.1007/978-3-030-01237-3_50
30. Zhou, H., Greenwood, D., Taylor, S.: Self-supervised monocular depth estimation with internal feature fusion. In: British Machine Vision Conference (BMVC) (2021)
31. Zhou, H., Taylor, S., Greenwood, D.: Sub-depth: self-distillation and uncertainty boosting self-supervised monocular depth estimation. arXiv:abs/2111.09692 (2021)
32. Zhou, T., Brown, M.A., Snavely, N., Lowe, D.G.: Unsupervised learning of depth and ego-motion from video. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 6612–6619 (2017)