2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
Macau, China, November 4-8, 2019

Inferring Distributions Over Depth from a Single Image

Gengshan Yang, Peiyun Hu, Deva Ramanan

***Abstract*— When building a geometric scene understanding system for autonomous vehicles, it is crucial to know when the system might fail. Most contemporary approaches cast the problem as depth regression, whose output is a depth value for each pixel. Such approaches cannot diagnose when failures might occur. One attractive alternative is a deep Bayesian network, which captures uncertainty in both model parameters and ambiguous sensor measurements. However, estimating uncertainties is often slow and the distributions are often limited to be uni-modal. In this paper, we recast the continuous problem of depth regression as discrete binary classification, whose output is an un-normalized distribution over possible depths for each pixel. Such output allows one to reliably and efficiently capture multi-modal depth distributions in ambiguous cases, such as depth discontinuities and reflective surfaces. Results on standard benchmarks show that our method produces accurate depth predictions and significantly better uncertainty estimations than prior art while running near real-time. Finally, by making use of uncertainties of the predicted distribution, we significantly reduce streak-like artifacts and improves accuracy as well as memory efficiency in 3D map reconstruction. Video and code can be found on the project website[1].**

I. Introduction

Most contemporary architectures for geometric scene understanding cast the problem as one of regression - given an image, infer a depth for each pixel. However, in safety-critical systems such as autonomous vehicles, such perceptual inferences will be used to make critical decisions and motion plans with considerable implications for safety. For example, what if the estimated depth of an obstacle on the road is incorrect? Here, it is crucial to build recognition systems that (1) allow for safety-critical graceful-degradation in functionality, rather than catastrophic failures; (2) are self-aware enough to diagnose when such failures occur; and (3) extract enough information to take an appropriate action, e.g. a slow-down, pull-over, or alerting of a manual operator. Such requirements are explicitly laid out in Automotive Safety Integrity Level (ASIL) standards which self-driving vehicles will be required to satisfy [18].

Such safety standards represent significant challenges for data-driven machine vision algorithms, which are unlikely to provide formal guarantees of performance [27]. One attractive solution is that of probabilistic modeling, where uncertainty estimates are propagated throughout a model. In the contemporary world of deep learning, deep Bayesian methods [6], [17] provide uncertainty estimates over model parameters (e.g., observing a scene that looks different than experience) and uncertainty estimates arising from ambiguous data (e.g., a sensor failure). We apply such approaches to the problem of depth estimation from a single camera. Our particular approach differs from prior work in two notable aspects. First, prior methods often require Monte Carlo sampling to compute uncertainty estimates [6], which can be slow for real-time safety-critical applications. Second, while certainty estimates provide some degree of self-awareness, they are limited to *uni*modal estimates of scene structure, implicitly producing a Gaussian estimate of depth represented by a regressed mean and regressed variance (or confidence) [17]. Instead, we develop representations that report back *multi*modal distributions that allow us to ask more nuanced questions (e.g., “what is the second possible depth of a pixel?”, “how many modes exist in the distribution?”), as shown in Fig. 1.

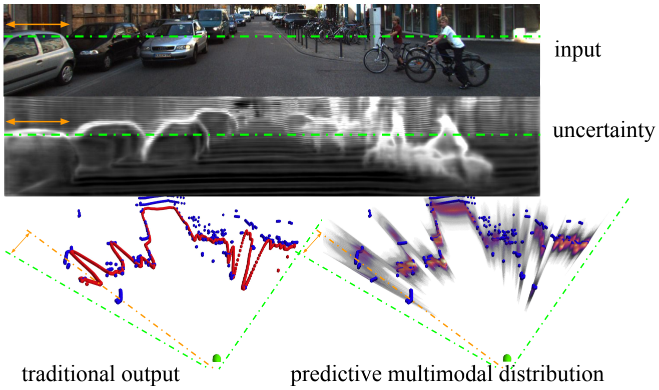


Fig. 1: Given an input image, traditional methods predict a single depth value for each pixel. In this paper, we describe an approach that predicts a per-pixel multi-modal distribution over depth. In the example above, we zoom in onto depth predictions along the dashed green line. Inside the input image, we highlight a segment filled with depth continuities marked with a yellow double-head arrow, where pixels could come from the car in the front, the car behind, or even the building in the back. In the output at the bottom, we mark ground truth depth with blue and depth with higher probabilities with red. While traditional methods incorrectly yield the mean of different modes, our approach successfully captures the multi-modal nature.

From a practical perspective, one may ask why bother estimating depth from a single camera when special-purpose sensors for depth estimation exist (such as LIDAR or multi-view camera rigs)? Common arguments include cost, payload and power consumption of robots [29], but we motivate this problem from a safety perspective. One crucial method for ensuring ASIL certification is redundancy, and so estimates of scene geometry that are independently produced from

Robotics Institute, Carnegie Mellon University, Pittsburgh, PA 15213, USA. {gengshay,peiyunh,deva} at cs.cmu.edu

[1] https://github.com/gengshan-y/monodepth-uncertainty

978-1-7281-4003-2/19/$31.00 ©2019 IEEE

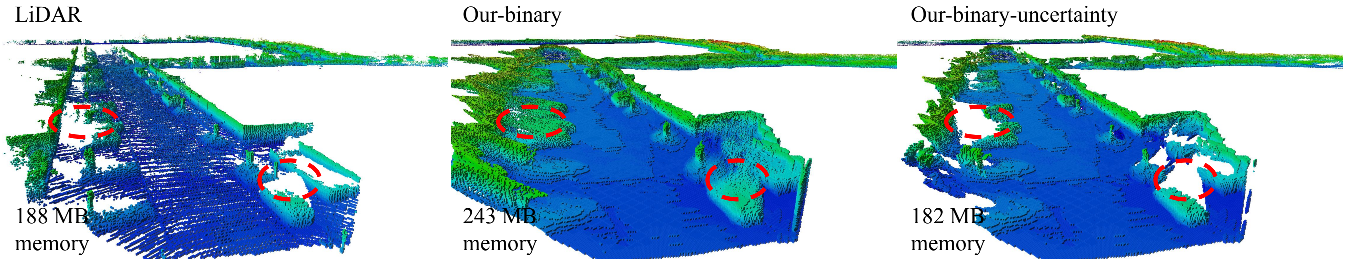


Fig. 2: From left to right, we show 3D maps built with LiDAR measurements (left), vanilla monocular depth predictions (middle), and *most certain* monocular depth predictions (right). Color encodes normalized heights. By thresholding depth predictions with uncertainty, we can remove streak-like artifacts (red dotted circles) and reduce memory usage by a quarter. To generate these maps, we feed depth measurements/predictions into OctoMap [14] and use odometry measurements as provided. LiDAR and monocular images come from the KITTI odometry sequence-00, which is not included in training.

various sensors (e.g., independently from LIDAR and independently from cameras) *and* that agree provide additional fault tolerance. In Fig. 3, we illustrate a situation in which monocular depth estimation complements range sensing.

Our overall approach to probabilistic reasoning is to recast the continuous problem of depth regression (given an image patch x, regress a depth value $y \in R$) as a discrete problem of selecting one out of many possible discretized depths $y \in \{1,2,\ldots K\}$. Previous work [3] has already demonstrated that discretization can improve the *accuracy* of the underlying depth regression task, but we show that discretization is even more useful for producing simple and efficient (and possibility multimodal) *uncertainty* estimates of depth. Intuitively, K-way classifiers are often trained with softmax loss functions, and so naturally report a distribution over K possible discrete depths. Importantly, we find that such distributions can be further improved by recasting the multiclass formulation as a binary *multilabel* task $y \in \{0,1\}^K$ - essentially, train K independent binary classifiers that classify patches at particular discrete depths. It is straightforward to show that the binary multilabel formulation can be seen as a relaxation of the multiclass problem that removes a linear constraint. Removing this constraint creates a more challenging learning problem that appears to be better regularized in terms of uncertainty reports. At test-time, we use the K logits as an unnormalized distribution over possible depths, though they can easily be normalized post-hoc (to compute summary statistics such as the expected depth).

Our main contributions are as follows:

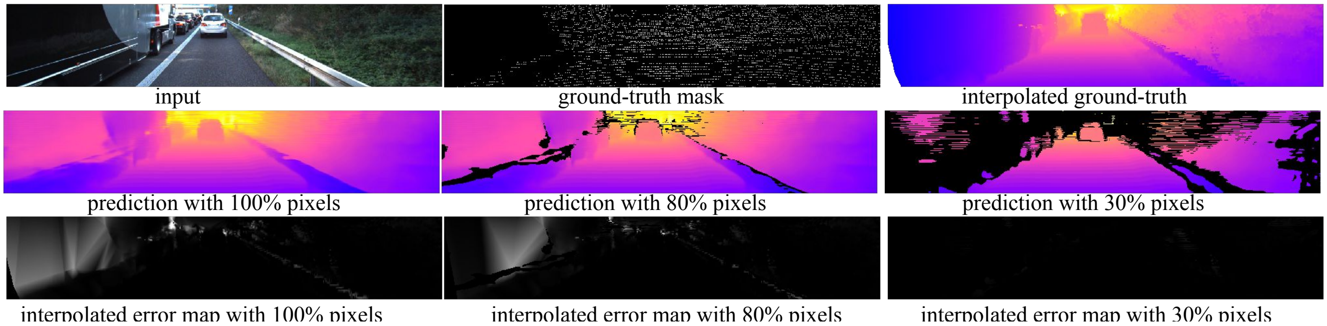
- We formulate the problem of monocular depth estimation in a probabilistic framework, which gives us confidence intervals of depth instead of point estimations.
- We recast the problem of depth regression as *multi-label* depth classification, which yields *reliable*, *multi-modal* distributions over depth.
- Our method produces accurate depth and significantly better uncertainty estimation over prior art on KITTI and NYU-depth while running *near real-time*.
- Our predicted distribution over depths improves monocular 3D map reconstruction, reducing streak-like artifacts and improving accuracy as well as memory efficiency.

II. Related Work

Single Image Depth Estimation: Early works [13], [26] popularize the problem of inferring scene depth maps from a single image, making use of handcrafted features. Eigen et al. [4] take a data-driven approach to learn features in a coarse-to-fine network that refines global structure with local predictions. Some recent work substantially improves the performance on single image depth estimation using better deep neural network architectures [20], [23], [28].

Depth Estimation as Classification: Closely related to our work, Cao et al. [3] formulates depth estimation as a multi-class classification problem and use soft targets to train the model. However, they make inference by choosing the most likely depth class, which does not take full advantage of the depth distribution, while we explore richer inference methods based on the predicted depth distributions. More importantly, the standard multi-class classification approach tends to make confident errors and does not yield reliable uncertainty estimations. Instead, we learn the classification model as K independent binary classifiers, which regularizes the model and gives us much better uncertainty estimation as well as noticeable performance improvement on standard benchmarks. Fu et al. [5] formulate depth estimation as ordinal regression, aiming to predict a CDF over depth. However, they do not ensure the predicted CDF to be monotonically non-decreasing. This makes it ungrounded to apply probabilistic reasoning for uncertainty estimation. In contrast, we formulate depth estimation as a discrete classification problem, aiming to predict a valid depth PDF.

Uncertainty in Depth Estimation: Kendall et al. [17] introduce two kinds of uncertainties: epistemic uncertainty (over model parameters) and aleatoric uncertainty (over output distributions). They show that epistemic uncertainty is data-dependent while aleatoric uncertainty is not. They model aleatoric uncertainty by fitting the variance of Gaussian distributions (also proposed in recent work on lightweight probabilistic extensions for deep networks [7]). However, this might lead to unstable training and suboptimal performance. More importantly, this ignored the fact that depth distributions are multi-modal in many cases (for example at depth discontinuities and reflective surfaces). They capture epistemic uncertainty by Bayesian neural networks


input

ground-truth mask

interpolated ground-truth

prediction with 100% pixels

prediction with 80% pixels

prediction with 30% pixels

interpolated error map with 100% pixels

interpolated error map with 80% pixels

interpolated error map with 30% pixels

Fig. 3: A situation in which monocular depth estimation complements range sensing. In the top row, from left to right, we show a monocular image, a binary mask, and an entropy map. The binary mask shows where LiDAR readings are available and the entropy map summarizes the uncertainty of each pixel's predicted distribution. Note that a large chunk of the truck body with black paint has no LiDAR returns since LiDAR sensors are less reliable with less-reflective materials. Our monocular depth estimator successfully predicts high entropy in the area with black paint. In the bottom row, we show depth predictions with uncertain pixels removed. From left to right, we gradually increase the confidence threshold. The rightmost one plots 30% pixels with the most confident depth predictions, in which we see most predictions on the truck body are removed. If a perception system solely relies on LiDAR measurements, it will perceive plenty of free space on the left side, which might lead to catastrophic decisions. If a perception system is designed with redundancy, it would trust LiDAR measurements less at pixels where the monocular estimator predicts uncertainties.

[6]. However, it requires expensive Monte Carlo sampling to obtain depth predictions and uncertainty estimations. Instead, we focus on modeling the multi-modal distributions over depth, which gives us more reliable uncertainty metrics without the additional computational overhead.

Multiple Hypotheses Learning (MHL): Prior works [11], [21] formulate the problem of learning to predict a set of plausible hypotheses as multiple-choice learning. They train an ensemble of models to produce multiple possibilities and define an oracle to pick up the best hypothesis. Rupprecht et al. [25] uses a shared architecture to produce multiple hypotheses and train the network by assigning each sample to the closest hypothesis. Different from these approaches, we train a single network to produce a multi-modal distribution, from which we can obtain multiple predictions without directly optimizing an oracle loss in training.

III. Method

We solve the problem of inferring continuous depth through discrete classification. To illustrate the method, we first introduce how we discretize continuous depth into discrete categories. Then we show the formulation of depth estimation as a multi-class classification task (mutual exclusive) and a multi-label classification task (not mutually exclusive). Then we discuss the output of our model, i.e. a probabilistic categorical distribution over discrete depths, and how we will evaluate the output, including evaluating as a standard depth estimation task and as a depth estimation with uncertainty.

Discretization: We discretize continuous depth values in the log space. Given a continuous range of depth $[a,b)$, we discretize it into K intervals, i.e. $[d_1,d_2),[d_2,d_3),..,[d_K,d_{K+1})$, with

$$d_k = \log a + \frac{k-1}{K}(\log b - \log a), k \in \{1\dots K\}. \qquad (1)$$

This captures the perceptual difference in human visual systems, i.e., we care more about differences in depths of close objects than distant ones. Furthermore, due to sensor sampling effects, we tend to encounter more close points rather than far away ones. Working in log space partially alleviates this class imbalance problem.

Multi-class Classification: As a baseline method, we first show how we recast the continuous regression as a multi-class classification problem. A discrete distribution over depth y can be parameterized by a categorical distribution $Cat(K,\boldsymbol{\mu})$. We learn to predict the probability $\mu_k = p(y = k)$ of each depth label by minimizing the negative log likelihood. Since we use the output of a softmax layer as the predicted probability, we will also refer to this variant as "Softmax" in the following text. Given a ground truth label y^*, image feature $\mathbf{x}$, and the model parameters $\mathbf{w}$, the loss function can be written as,

$$L(\boldsymbol{\mu}|y^*) = -\sum_{k=1}^{K} \mathbb{1}(k=y^*)\log \mu_k(\mathbf{x};\mathbf{w}). \qquad (2)$$

Here the distribution $\boldsymbol{\mu}(\mathbf{x};\mathbf{w})$ is predicted from a K-way multi-class classifier.

Equation (2) gives us the cross-entropy between an one-hot label vector $\mathbb{1}(k=y^*)$ and the predicted distribution $\boldsymbol{\mu}(\mathbf{x};\mathbf{w})$. To incorporate the ordinal nature of the depth labels, i.e., penalize predictions closer to the ground truth less than predictions further away, we replace the one-hot target vector $\mathbb{1}(k=y^*)$ with a discretized Gaussian centered around the ground truth, i.e.,

$$q(k;y^*) = \frac{1}{Z}e^{\frac{||k-y^*||^2}{-2\sigma^2}}, \qquad (3)$$

where Z is the partition function.

Binary Classification: To alleviate competition between depth classes, we further model continuous depth as a collection of K independent Bernoulli random variables $y_k \sim B(1,\mu_k)$, where μ_k encodes the probability of falling into the k_{th} depth interval. We also refer to this variant as multilabel

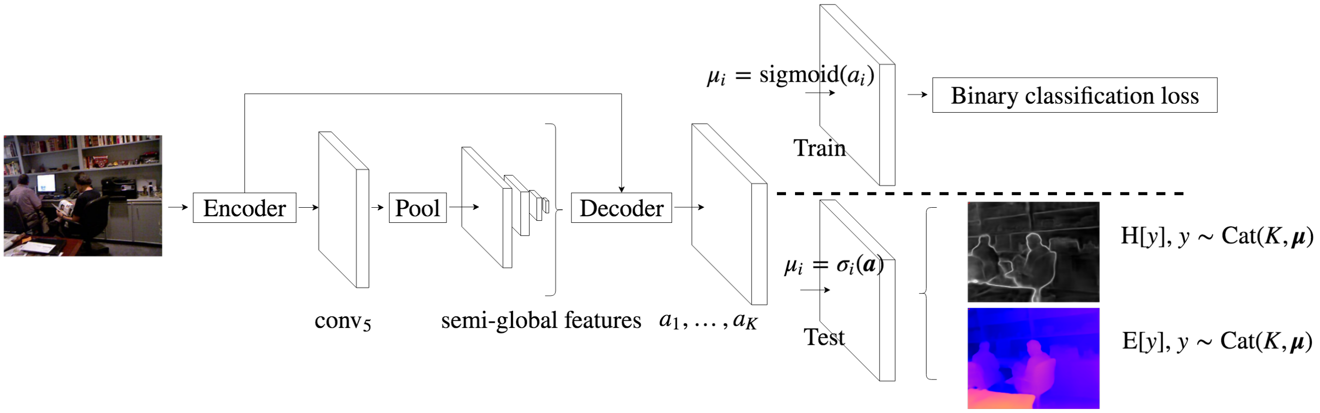

Encoder
Pool
Decoder
conv$_5$
semi-global features
$a_1, \dots, a_K$
μ_i = sigmoid(a_i)
Train
Binary classification loss
$\mu_i = \sigma_i(\boldsymbol{a})$
Test
H[y], $y \sim$ Cat($K, \boldsymbol{\mu}$)
E[y], $y \sim$ Cat($K, \boldsymbol{\mu}$)

Fig. 4: Our network architecture consists of an encoder, a spatial pyramid pooling module, and a decoder. Our encoder is a ResNet-50 truncated before global pooling. Spatial pyramid pooling takes ResNet feature then extracts global and semi-global feature through multi-scale pooling. The decoder processes pooled feature to predict a un-normalized score map for each discrete depth class $a_1, \dots, a_K$. During training, each un-normalized score map is pushed into a per-pixel soft-labeled binary cross entropy loss; at test time, we perform per-pixel normalization using softmax across all depth classes to ensure a valid per-pixel distribution over depth, from which we can make the final prediction of depth and uncertainty. Original-resolution images are used as input, and the predictions are bilinearly up-sampled to the same resolution as ground-truth.

in the paper. The loss function is written as,

$$L(\boldsymbol{\mu}|y^*) = -\sum_{k=1}^{K} [\tilde{q}(k;y^*) \log \mu_k(\mathbf{x};\mathbf{w}) \\ + (1 - \tilde{q}(k;y^*))(\log(1 - \mu_k(\mathbf{x};\mathbf{w})))], \tag{4}$$

where $\tilde{q}(k;y^*) = e^{\frac{||k-y^*||^2}{-2\sigma^2}}$ is an *unnormalized* version of soft target distribution.

One can see this as a *relaxation* of the training objective from Eq.(2) that drops the constraint that $\sum_k \mu_k = 1$ [22]. The variance σ^2 is designed such that for all depth classes within 25% difference to ground truth, their label is greater than 0.5. In test time, we push the pre-logit scores of each binary classifier through a softmax and obtain a distribution over discrete depth, as shown in Fig. 4.

Predicting Depth from a Distribution: After obtaining the distribution over depth, Cao et al. [3] report the most confident depth class, ignoring the multi-modal nature of the predicted distribution. Different from their approach, we report the *expected* depth based on the predicted distribution as $\mathbb{E}[y] = \sum_k \mu_k d_k$, which takes into account the whole distribution and yields better depth estimations.

Uncertainty and Multiple Hypotheses: We now describe various statistics that can be computed from our multimodal distribution, motivated by autonomous robotic perception. Because the perception module of robots needs to be self-aware enough to report potential failures to the downstream planner or online-mapping module when faced with ambiguous scenes, the first statistic is uncertainty, as computed with Shannon entropy:

$$H(y) = -\sum_k \mu_k \log \mu_k. \tag{5}$$

Secondly, even if the most-likely (or expected) depth of a particular pixel is far away, a robotic motion planner may wish to decrease speed if there is a non-negligible probability that its depth is in the near-field (due to say, a translucent obstacle). As such, our network can directly output *multiple* depth modes to downstream planners.

Evaluation: Evaluating the above functionality on a robotic platform is difficult. Instead, to evaluate the quality of uncertainty estimation, we make use of the area under ROC curve (AUC), which is widely used in stereo vision and optical flow [2], [15]. To assess the accuracy of the multi-hypotheses output, we follow past work on MHL [11], [21] and use an "oracle" evaluation protocol where an algorithm is allowed to report back multiple depth predictions, and the best one is chosen to compute the accuracy [11]. We also report standard metrics [4] on depth estimation benchmarks.

Implementation We follow the architecture of Kuznietsov et al. [19] as shown in Fig. 4. We further add a spatial pyramid pooling module [12] to extract global and semi-global features from the scene. We experimented with different numbers of bins on KITTI. With 32, 64, 96, 128 bins, our method achieves an absolute relative error (ARE) of 9.34%, 8.61%, 8.60%, 8.59%. As improvement becomes marginal, we pick 64 as the number of bins and used it for all experiments in this paper.

IV. Experiments

We first introduce our experimental setup, including dataset and training details. We then compare to prior estimation methods that reason about uncertainties. Finally, we compare our method with the state-of-the-art on the standard depth estimation task, as well as using multi-hypotheses evaluation [25].

Setup: We test our method on the standard depth estimation benchmarks, including KITTI [8] for outdoor scenes (1-80m) and NYU-v2 [4] for indoor scenes (0.5-10m). On KITTI, we follow Eigen's split [4] for training and testing. On NYU-v2, we sample 13k images following [20] for training and test on the official test split.

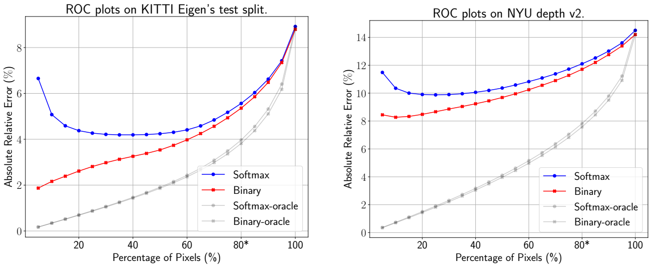


Fig. 5: How well does the predicted uncertainty correlate with the actual depth estimation performance? We first sort all predictions in ascending order of uncertainty. Then we gradually include more predictions for evaluation by increasing the uncertainty threshold (including more uncertain predictions in the evaluation). The X-axis represents the percentage of pixels we include and Y-axis represents the ARE on the selected pixels. Notice uncertainties estimated by the model trained with multi-class classification loss ("Softmax" [3]) are not well correlated with error, especially for the most confident pixels. On the contrary, the error increases monotonically as confidence drops for our proposed approach ("Binary"). At 80%, our method also achieves a lower error rate (5.4% vs. 5.6%).

Training: We first initialize the weights of our ResNet-50 backbone with the ImageNet pre-trained ones. To augment training data, we apply random gamma, brightness, and color shift, as in [10]. We fine-tune the weights with an Adam optimizer with an initial learning rate of 0.0001 and decrease the learning rate with a factor of 0.1 after 45 epochs. We train our KITTI model for a total of 60 epochs and our NYU-v2 model for a total of 160 epochs. Our experiments are run on a machine with GeForce GTX Titan X GPU using Tensorflow.

A. *Depth Estimation with Uncertainty*

Baselines: Considering most prior art do not reason about uncertainty, we compare to predictive Gaussian and predictive Gaussian with Monte Carlo dropout (Gaussian-dropout) [7], [17] in terms of depth estimation with uncertainty, as shown in Tab. I. For a fair comparison, we re-implement and train predictive Gaussian and Gaussian-dropout on KITTI and NYU depth v2. We make sure the re-implemented version has an architecture that is as close as possible to ours. For predictive Gaussian, we use the same backbone architecture but with a different prediction head, which predicts mean and variance of a Gaussian distribution over depth in log space. To train predictive Gaussian, we minimize the per-batch negative log-likelihood based on the predicted mean and variance. For Gaussian-dropout, we use the same backbone architecture and prediction head except we perform dropout with a probability of 0.5 after several convolutional layers, as in Kendall et al. [16]. During inference, we draw 32 samples to make predictions and estimate uncertainty. Following the same idea, we apply Monte Carlo dropout to our binary model, referred to as Binary-dropout.

Following Hu et al. [15], we plot ROC curves to evaluate our depth estimation with uncertainty, as shown in Fig. 5 and Fig. 6. Such curves demonstrate how well the predicted uncertainty correlates with the actual depth estimation performance. A point (x, y) on the curve indicates a performance of y on the least uncertain x (%) predictions over all pixels in the test set. Perfect uncertainty estimation, from the perspective of the ROC curve, should rank predictions as if they are ranked by the actual error. As a reference, we include curves with such oracle w.r.t. a specific error metric (absolute relative error or ARE). Below, we first compare two variants of our model (binary classification and multiclass classification). Then we will compare our model to prior art that predicts uncertainty (predictive Gaussian and Gaussian-dropout). For each sub-metric under AUC, we follow the definition in Eigen et al. [4].

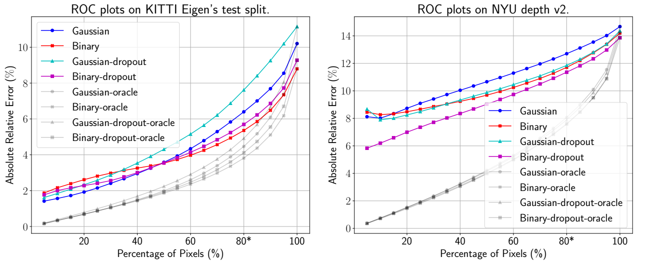


Fig. 6: Compared to predictive Gaussian [17] ("Gaussian"), our method ("Binary") yields lower error rate when more than 50% pixels are kept for KITTI, and more than 15% pixels for NYU. By applying Monte Carlo dropout, both predictive Gaussian ("Gaussian-dropout") and our approach ("Binary-dropout") see a significant improvement on NYU. While on KITTI, the performance get strictly worse for predictive Gaussian.

Binary classification vs Multiclass classification: In Fig. 5, we compare the model trained with binary classification loss ("Binary") to the model trained with multi-class classification loss ("Softmax"). As we can see on the left side of both plots, the uncertainty predicted by the multi-class classifier does not correlate well with the actual error rate, especially for those least uncertain (or most confident) pixels. In contrast, the model trained with binary classification loss produces a curve that monotonically increases as the uncertainty threshold goes up, because it is able to *correctly rank more correct pixels as more confident*. We posit that our multilabel loss (that removes a linear constraint present in the multi-class formulation) acts as an additional regularizer that improves uncertainty estimation.

Gaussian vs Binary: In Fig. 6, we find predictive Gaussian also yields reliable uncertainty estimation, as it produces a monotonically increasing curve. Overall it achieves a slightly worse performance, comparing to our model trained with binary classification. It might be due to its uni-modal assumption and optimization difficulties in training time (discussed further in our ablation study). Interestingly, adding Monte Carlo dropout significantly improves NYU performance for both predictive Gaussian ("Gaussian-dropout") and our approach ("Binary-dropout"). However, on KITTI, we see strictly worse performance for the predictive Gaussian.

Quantitative evaluation: In Tab. I, we further compare uncertainty estimation quantitatively using metrics introduced in Section III. Our binary classification method produces better performance in terms of AUC compared to predictive Gaussian and its Monte Carlo dropout variant

		AUC			time
	Method	ARE	RMSE	$1-\delta_1$	(ms)
K	Gaussian [17]	4.38	1.42	2.63	64
	Softmax	5.19	2.88	2.93	74
	Binary	<u>**4.17**</u>	**1.33**	<u>**1.79**</u>	74
	Gaussian-dropout [17]	5.18	<u>1.21</u>	3.61	467
	Binary-dropout	4.20	1.33	2.06	540
N	Gaussian [17]	10.94	**0.41**	10.95	44
	Softmax	11.17	0.53	11.09	52
	Binary	**10.28**	0.42	**9.26**	52
	Gaussian-dropout [17]	10.33	<u>0.32</u>	10.30	353
	Binary-dropout	<u>9.39</u>	0.40	<u>7.79</u>	410

TABLE I: Quantitative evaluation for uncertainty estimation on KITTI (K) and NYU-v2 (N). The best results among methods without Monte Carlo dropout are made bold, while the best considering Monte Carlo dropout are underlined. On both datasets, we compare our method trained with the binary loss ("Binary") and the multiclass loss ("Softmax") to predictive Gaussian [17] ("Gaussian"). The quantitative results are consistent with Fig. 5 and Fig. 6. In terms of AUC on ARE and $1-\delta_1$ (*the lower the better*), our binary loss consistently outperforms predictive Gaussian on both KITTI and NYU-v2. Importantly, when combined with Monte Carlo dropout, our binary model ("Binary-dropout") further reduces the AUC on NYUv2.

in terms of ARE and δ_1, without expensive Monte Carlo sampling. By adding Monte Carlo dropout to our model, we can further improve AUC of ARE, RMSE and δ_1 on NYU depth v2. Although predictive Gaussian with Monte Carlo dropout outperforms our binary loss on all metrics based on RMSE, it is too slow for real-time perception. Please refer to Tab. I for more detailed discussion.

B. Multi-hypothesis Depth Prediction

We first evaluate standard depth prediction performance on KITTI and NYU-v2 using metrics proposed in [4], as shown in Tab. II. We then extend the evaluation by allowing multiple depth hypotheses. For a fair comparison, we re-implement Fu et al. [5] and Cao et al. [3] under the same setup as ours (a light-weight backbone and no test-time ensemble). We also include numbers in the original paper as a reference. Please refer to Tab. II for detailed comparison.

To evaluate our multi-modal distributions, we follow the standard protocol in multi-hypothesis learning [21]. After computing the pre-logits scores, we report back M depth hypotheses with the highest scores, and the one with the lowest error is selected by the oracle for evaluation. Since most methods *can't* output multiple hypotheses, we compare to the ones that *can* be trained to output multiple hypotheses [25], referred to as MHL. Similar to traditional L_2 regression, we directly regress to the depth in log space. However in training time, we make M predictions and construct an oracle loss by selecting the prediction that best describe the ground-truth in terms of L_2 distance. We train the MHL baseline for $M = 1,3,5,10$, and use an oracle to select the best prediction for evaluation. Please see Fig. 7 for analysis of the results.

V. Building Maps with Uncertainty

In this section, we demonstrate one application of geometric uncertainty estimation: robust map reconstruction.

	Method	ARE (%)	RMSE	δ_1 (%)	time (ms)
K	Binary	**8.8**	**3.88**	**91.0**	74
	Fu et al. [5]	9.1	3.90	90.5	74
	Cao et al. [3]	9.3	4.02	90.8	74
	Eigen et al. [4]	19.0	7.16	69.2	13
	Godard et al. [10]	11.4	4.94	86.1	35
	Cao et al. [3]	11.5	4.71	88.7	-
	Fu et al. [5]	<u>7.2</u>	<u>2.73</u>	<u>93.2</u>	1250
N	Binary	14.2	0.51	82.7	52
	Binary-dropout	**13.9**	<u>**0.50**</u>	<u>**82.8**</u>	410
	Kendall et al. [17]	14.4	0.51	81.5	353
	Eigen et al. [4]	15.8	0.64	76.9	10
	Laina et al. [20]	12.7	0.57	81.1	55
	Fu et al. [5]	11.5	0.51	<u>82.8</u>	-
	Kendall et al. [17]	<u>11.0</u>	0.51	81.7	7500

TABLE II: Performance on KITTI (K) Eigen's split and NYU-V2 depth (N) dataset. The best results over the light-weight setup are bolded, while the best results overall are underlined. On KITTI, our method outperforms the state-of-the-art Fu et al. [5] under the same setup. With its original setup (a heavy-weight backbone and test-time ensemble), [5] runs nearly 17x times slower (1250ms vs 75ms). On NYU-v2, our method outperforms Kendall et al. [17] with the same backbone network. With its original setup, Kendall et al. [17] runs 144x slower. Our method further improves when training with dropout and testing with MC sampling [16], referred to as Binary-dropout.

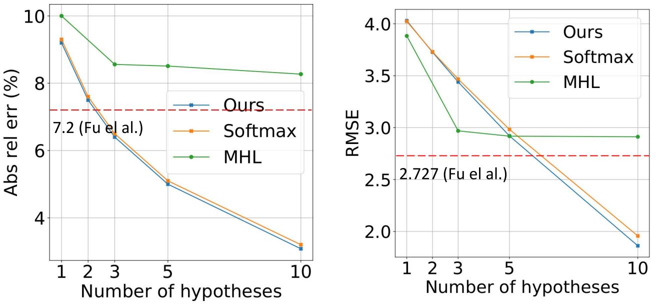


Fig. 7: Error as a function of hypotheses number on KITTI. Compared to MHL, our method always produces better results in terms of ARE. As for RMSE, our method performs worse than MHL when $M < 5$, possibly because the MHL baseline is trained to directly minimize squared error. However, MHL's error stops going down after $M \geq 3$, while we do not observe this effect for our model. Compared to softmax (Cao et al. [3]), our method also achieves slightly better performance. Also, our method consistently out-performs Fu et al. [5] in ARE using more than two hypotheses, and in terms of RMSE using more than five hypotheses.

Though maps are often constructed in an offline stage, online mapping can be an integral part of autonomous navigation in unknown/changing environments [24].

In practice, is it notoriously difficult to build 3D maps from raw depth predictions because they tend to contain "streak-like artifacts" [1], which not only affect the quality of the map but also increase memory usage (because they often result in larger occupied volumes). Empirically, we find that such artifacts often happen where ground truth depth is inherently ambiguous and follows a multi-modal distribution, e.g. depth discontinuities and reflective surfaces. Since our depth estimator is designed to predict multi-modal distributions over depth, we use it to improve the accuracy of

Method	Accuracy (%)	Memory (MB)
LiDAR-FOV†	95.9	1220.9
Ours-binary	88.3	1682.6
Ours-binary-80%	**89.9**	**1263.2**

TABLE III: Accuracy and memory usage of online mapping. LiDAR-FOV indicates the map built using LiDAR points in the left camera field of view, which is the upper-bound of our methods. The map built with top 80% most confident estimations of our model (Ours-binary-80%) significantly reduces the memory usage and also improves the mapping accuracy.

map reconstruction. By simply thresholding the uncertainty of each pixel's predicted distributions, we can significantly reduce streak artifacts and memory usage, as shown in Fig. 2.

We evaluate the performance of map reconstruction with and without uncertainty on KITTI odometry sequence-00 [9], which is not included in the training set. Specifically, we run our monocular depth estimator on left RGB images, and feed the output depth maps together with ground-truth odometry as the input of Octomap [14]. The accuracy is measured as the percentage of correctly mapped map cells, where a cell counts as correctly mapped if it has the same state (free or occupied) as the LiDAR map (ground-truth). As shown in Tab. III, applying a simple uncertainty-based ranking and selection improves the accuracy of monocular maps by 1.8% and reduces the memory usage by 25%.

Conclusion

Robotic applications of perception present new challenges for safety-critical, fault-tolerant operation. Inspired by past approaches that advocate a probabilistic Bayesian perspective, we demonstrate a simple but effective strategy of discretization (with the appropriate quantization, smoothing, and training scheme) as a mechanism for generating detailed predictions that support such safety-critical operations.

References

[1] I. A. Bârsan, P. Liu, M. Pollefeys, and A. Geiger. Robust dense mapping for large-scale dynamic environments. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 7510–7517. IEEE, 2018.

[2] A. Bruhn and J. Weickert. A confidence measure for variational optic flow methods. In *Geometric Properties for Incomplete Data*, pages 283–298. Springer, 2006.

[3] Y. Cao, Z. Wu, and C. Shen. Estimating depth from monocular images as classification using deep fully convolutional residual networks. *IEEE Transactions on Circuits and Systems for Video Technology*, 2017.

[4] D. Eigen, C. Puhrsch, and R. Fergus. Depth map prediction from a single image using a multi-scale deep network. In *Advances in neural information processing systems*, pages 2366–2374, 2014.

[5] H. Fu, M. Gong, C. Wang, K. Batmanghelich, and D. Tao. Deep ordinal regression network for monocular depth estimation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[6] Y. Gal and Z. Ghahramani. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *international conference on machine learning*, pages 1050–1059, 2016.

[7] J. Gast and S. Roth. Lightweight probabilistic deep networks. In *Proceedings of CVPR*, volume 1, 2018.

[8] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun. Vision meets robotics: The kitti dataset. *The International Journal of Robotics Research*, 32(11):1231–1237, 2013.

[9] A. Geiger, P. Lenz, and R. Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012.

[10] C. Godard, O. Mac Aodha, and G. J. Brostow. Unsupervised monocular depth estimation with left-right consistency. In *CVPR*, volume 2, page 7, 2017.

[11] A. Guzman-Rivera, D. Batra, and P. Kohli. Multiple choice learning: Learning to produce multiple structured outputs. In *Advances in Neural Information Processing Systems*, pages 1799–1807, 2012.

[12] K. He, X. Zhang, S. Ren, and J. Sun. Spatial pyramid pooling in deep convolutional networks for visual recognition. In *European conference on computer vision*, pages 346–361. Springer, 2014.

[13] D. Hoiem, A. A. Efros, and M. Hebert. Geometric context from a single image. In *Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on*, volume 1, pages 654–661. IEEE, 2005.

[14] A. Hornung, K. M. Wurm, M. Bennewitz, C. Stachniss, and W. Burgard. Octomap: An efficient probabilistic 3d mapping framework based on octrees. *Autonomous robots*, 34(3):189–206, 2013.

[15] X. Hu and P. Mordohai. A quantitative evaluation of confidence measures for stereo vision. *IEEE transactions on pattern analysis and machine intelligence*, 34(11):2121–2133, 2012.

[16] A. Kendall, V. Badrinarayanan, and R. Cipolla. Bayesian segnet: Model uncertainty in deep convolutional encoder-decoder architectures for scene understanding. *arXiv preprint arXiv:1511.02680*, 2015.

[17] A. Kendall and Y. Gal. What uncertainties do we need in bayesian deep learning for computer vision? In *Advances in Neural Information Processing Systems*, pages 5580–5590, 2017.

[18] P. Koopman and M. Wagner. Challenges in autonomous vehicle testing and validation. *SAE International Journal of Transportation Safety*, 4(1):15–24, 2016.

[19] Y. Kuznietsov, J. Stückler, and B. Leibe. Semi-supervised deep learning for monocular depth map prediction. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6647–6655, 2017.

[20] I. Laina, C. Rupprecht, V. Belagiannis, F. Tombari, and N. Navab. Deeper depth prediction with fully convolutional residual networks. In *3D Vision (3DV), 2016 Fourth International Conference on*, pages 239–248. IEEE, 2016.

[21] S. Lee, S. P. S. Prakash, M. Cogswell, V. Ranjan, D. Crandall, and D. Batra. Stochastic multiple choice learning for training diverse deep ensembles. In *Advances in Neural Information Processing Systems*, pages 2119–2127, 2016.

[22] M. Li, L. Jeni, and D. Ramanan. Brute-force facial landmark analysis with a 140,000-way classifier. *AAAI*, 2018.

[23] F. Liu, C. Shen, G. Lin, and I. D. Reid. Learning depth from single monocular images using deep convolutional neural fields. *IEEE Trans. Pattern Anal. Mach. Intell.*, 38(10):2024–2039, 2016.

[24] T. Ort, L. Paull, and D. Rus. Autonomous vehicle navigation in rural environments without detailed prior maps. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2040–2047. IEEE, 2018.

[25] C. Rupprecht, I. Laina, R. DiPietro, M. Baust, F. Tombari, N. Navab, and G. D. Hager. Learning in an uncertain world: Representing ambiguity through multiple hypotheses. In *International Conference on Computer Vision (ICCV)*, 2017.

[26] A. Saxena, S. H. Chung, and A. Y. Ng. Learning depth from single monocular images. In *Advances in neural information processing systems*, pages 1161–1168, 2006.

[27] S. Shalev-Shwartz, S. Shammah, and A. Shashua. On a formal model of safe and scalable self-driving cars. *CoRR*, abs/1708.06374, 2017.

[28] D. Xu, E. Ricci, W. Ouyang, X. Wang, and N. Sebe. Multi-scale continuous crfs as sequential deep networks for monocular depth estimation. In *Proceedings of CVPR*, volume 1, 2017.

[29] Z. Yang, F. Gao, and S. Shen. Real-time monocular dense mapping on aerial robots using visual-inertial fusion. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pages 4552–4559. IEEE, 2017.