Proceeding of the IEEE
International Conference on Robotics and Biomimetics
Dali, China, December 2019

# Hierarchical Binary Classification for Monocular Depth Estimation

Hualie Jiang and Rui Huang*

*Abstract*— **Extracting dense depth from a single image is an important and challenging task in computer vision. Recently, significant progress has been made in monocular depth estimation (MDE) by formulating this problem as a multi-label classification instead of traditional regression. However, to discretize depth into hundreds of labels results in complicate network output and a large number of parameters. In this paper, we propose to encode the discretized depth into a binary code and formulate MDE as a hierarchical binary classification (HBC) problem to reduce the complexity. Besides, by studying the distribution of individual bit of the encoded depth codes, we find that our encoding scheme can also solve the problem of depth data imbalance. We conduct experiments on KITTI and NYU Depth V2 datasets, which show that our simplified approach still achieves a comparable performance with state-of-the-art methods.**

## I. INTRODUCTION

Depth prediction is to densely estimate the distance between the actual 3D point corresponding to an image pixel and camera center along the viewing direction of the camera, and this is a crucial step to recover the 3D structure of images. 3D reconstruction is a fundamental problem in computer vision, having many important applications, like 3D modeling [16] and scene understanding [17]. For example, depth prediction is applied to recover the absolute scale and avoid scale drift [28] in monocular simultaneous localization and mapping (SLAM), which is a critical technology for autonomous robot navigation.

Depth estimation from a single image is an ill-posed and very challenging problem, as only the direction of a 3D point is constrained excluding the distance. Thus, it is not well studied as many traditional techniques for acquiring depth from images, for instance, stereo-vision [14], structure-from-motion [12], and structured light techniques [27]. However, these techniques have limits and disadvantages compared to MDE. For example, in stereo-vision, the measurement range is limited by the baseline and the precision is easily affected by texture-less scenes. On the other hand, although depth estimation from a single image lacks geometrical constraints, it still can be addressed because the rich context information from images can be extracted and mapped into depth by deep convolutional neural networks (CNNs) [8]. Moreover, the phenomenon that human beings and animals can use only one eye to roughly precept the structure of the environment by visual context indicates that the MDE is solvable.

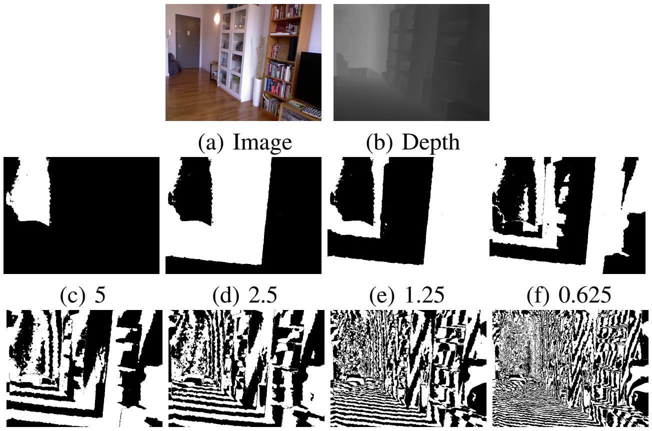

(a) Image (b) Depth
(c) 5 (d) 2.5 (e) 1.25 (f) 0.625
(g) 0.3125 (h) 0.15625 (i) 0.078125 (j) 0.0390625

Fig. 1: **Illustration of Binarily Encoded Depth Map.** (a) is a RGB image in NYU Depth V2 [26] dataset and (b) is its corresponding depth map, and both make up a sample for the MDE task. The depth bit maps (c), (d), (e), (f), (g), (h), (i), (j) and (k) are obtained by uniformly quantizing and binarily encoding the depth map (f), associating with depth weight 5m, 2.5m, 1.25m, 0.625m, 0.3125m, 0.15625m, 0.078125m and 0.0390625m, respectively.

In recent years, CNN-based methods have largely advanced the performance of MDE [8], [7], [19], [24], [11], [2], [23], [21], [9], [15]. While MDE is conventionally treated as a dense regression problem, it is also formulated as a multi-label classification problem in [2], [9], [23], [21] by discretizing depth into many bins as different labels. As CNNs are designed to solve classification problems, such kind of approaches enjoys some performance gain than the regression techniques. However, in these methods, depth has to be discretized into about 100 levels to achieve an optimal performance, which results in inefficiency.

In this paper, we investigate the task of MDE by formulating it as a problem of hierarchical binary classification. For the sake of efficiency, we propose to further binarily encode the discretized depth labels into some depth bit maps as illustrated by Figure 1. Thus we can use CNNs to predict these depth bit maps instead of multiple discretized depth labels and thus the number of logits in CNNs will decrease logarithmically. Accordingly, we transform a multi-class classification problem into a multiple binary-class classification problem. By examining the soft-weighted-sum inference [21] in our encoding framework, we find we can translate the probability maps output by the network into depth much

Hualie Jiang and Rui Huang are with School of Science and Engineering, The Chinese University of Hong Kong, Shenzhen. Rui Huang is also with Shenzhen Institute of Artificial Intelligence and Robotics for Society. *Corresponding author: Rui Huang (email: ruihuang@cuhk.edu.cn).

This work is supported in part by National Key R&D Program of China (Grant No. 2017YFB1303701), funding from Shenzhen Institute of Artificial Intelligence and Robotics for Society, and Robotics Discipline Development Fund from Shenzhen Municipal Government (Grant No. 2016-1418).

978-1-7281-6321-5/19//$31.00 ©2019 IEEE

more efficiently. To be specific, we reduce the complexity from $O(L \log L)$ to $O(\log L)$ for the translation, if the depth is discretized into $L$ levels. Additionally, the reconstructed depth can be represented as a linear combination of the depth bits with an exponentially decreasing weight. Thus, to reconstruct depth from bits is a process of a hierarchical binary decision. In the reconstruction process, we first judge whether the depth is in the far half or near half in the entire depth range according to the first bit, and it inductively makes a binary decision on the base on the depth range decided by the former bits. This process of coarse-to-fine decision is similar to human's perception of distance, as we usually first globally estimate the distance of an object and then make better estimation by looking locally around this object.

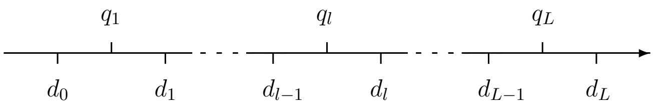
Our representation of depth is also helpful to address the problem of serious data imbalance due to the perspective effect, which has been pointed out in [21]. Large depth is the minority in a depth map, and as indicated by (b) in Figure 1, the depth that is larger than $5m$ in a $10m$ range in only accounts for a small proportion. Li *et al.* [21] found that to equally discretizing the depth value in the log space performs better than simple uniform quantization. This is because larger depth can tolerate larger quantization error and depth labels become less concentrated after the log transformation. However, even after the log transformation, the depth distribution still exhibits obvious nonuniformity [21]. In our approach, only the first or second depth bit map exhibits imbalance if uniform quantization is applied and if we adopt uniform quantization in log space, almost no imbalance appears. Therefore, our encoding scheme can largely alleviate depth imbalance. Although it is not eliminated, we believe that the research line of our approach has the potential to solve the data imbalance in MDE.

As we have formulated MDE as a dense classification problem, we can use existing network architecture for dense classification problems, such as semantic image segmentation. In this paper, we use a state-of-the-art semantic segmentation network, DeepLab V3 [4] which can capture multi-scale information efficiently by using the atrous spatial pyramid pooling (ASPP) module, and ResNet [13] as the backbone. In training the network, we dynamically adjust the cross entropies of different depth bits, to guide the network to coarsely concentrate on bits with bigger depth weights initially, and gradually treat all bits more equally to make a refined prediction.

Our contribution can be summarized as follows. First, to our best knowledge, we first propose to binarily encode depth for MDE and model this complicated dense regression problem to an easier dense binary classification problem. Second, we find that data imbalance can be largely reduced by our encoding scheme, and our encoding scheme can solve the data imbalance problem in MDE.

## II. Approach

In this section, we present our approach in detail, including depth representation and reconstruction, the network that we adopt and the loss function to train the network.



Quantization Level: $L$
Partition Value: $d_l$
Partition Region: $D_l = [d_{l-1}, d_l)$
Quantized Index : $Q_i(x) = l, if\ x \in D_l$
Quantized Value : $Q(x) = q_l, if\ x \in D_l$

Fig. 2: **Illustration of Quantization.** The minimum and maximum depth are $d_0$ and $d_L$, respectively. We divide the depth range into $L$ bins, with $d_l$ as the boundary. Any depth in region $D_l$ will be quantized to an index $l$ and a value $q_l$.

### A. Depth Representation and Reconstruction

Analog signals are usually digitalized by sampling, quantization and binary encoding for efficient storage and transmission. We briefly show how to quantize and encode depth and more detailedly show how to decode network output into the predicted depth map efficiently.

*1) Depth Quantization and Encoding:* Figure 2 shows how to perform quantization. In source coding [31], the optimal quantization is found by minimizing the quantized error and uniform quantization is optimal if the signal is uniformly distributed under the criterion of MSE. The partition value and quantized value of uniform quantization must satisfy,

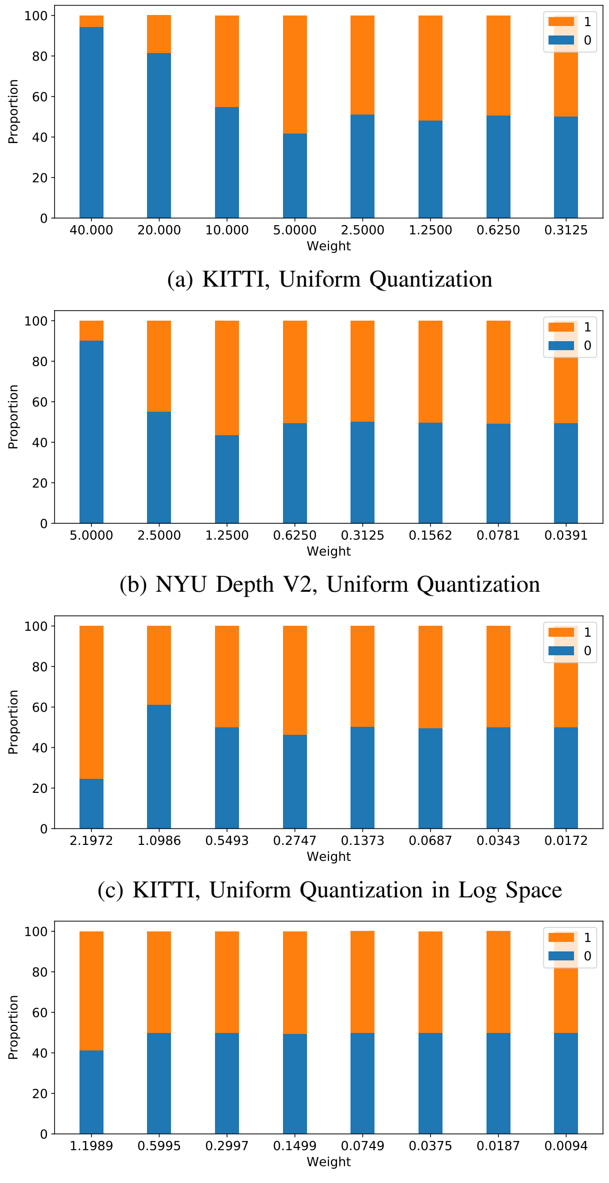
$$d_l - d_{l-1} = \frac{d_L - d_0}{L} \text{ and } q_l = \frac{d_l + d_{l-1}}{2}, \ \forall l \tag{1}$$

Non-uniform quantization is usually implemented by first applying a non-linear transformation on a signal and uniformly quantizing it. Specifically, we can perform uniform quantization depth on log space by first applying a log function on the depth and uniformly quantizing the log depth.

Binary encoding requests that quantization level satisfy $L = 2^N$, where $N$ is a positive integer representing the number of bits of the depth code. Then we can further represent the quantized label $l$ as $N$ bits, $b_1, b_2, \cdots, b_N$,

$$l = 1 + \sum_{n=1}^{N} b_n 2^{n-1} \tag{2}$$

*2) Depth Bit Distribution:* As we can see in Figure 1, 0 dominates the depth bit map with the biggest depth weight, which is due to the long-tailed distribution of depth as found by [21]. To further understand data imbalance in our depth encoding scheme, we also count the proportion of 0 and 1 in the depth bits of the whole training set of two datasets, KITTI [10] and NYU Depth V2 [26], as illustrated in Figure 3. When we perform uniform quantization, we observe that data imbalance only exhibits in one or two bits with the biggest depth weight. When we perform uniform quantization after applying the log transformation (we use $\log(1+x)$ to map 0 to 0), we find that the data imbalance can be further reduced by our encoding scheme. In KITTI, large enhance in balance occurs in the first two bits, for example, the ratio between 0



(a) KITTI, Uniform Quantization

(b) NYU Depth V2, Uniform Quantization

(c) KITTI, Uniform Quantization in Log Space

(d) NYU Depth V2, Uniform Quantization in Log Space

Fig. 3: **The Distribution of $0$ and $1$ in Depth Bits.** The proportion of $0$ and $1$ in bits with different depth weight of our encoding scheme in two Datasets: KITTI [10] and NYU Depth V2 [26]. **Uniform Quantization:** depth weight represents actual distance. (a) In KITTI, $0$ dominates the bits of the two biggest depth weights with about $95\%$ and $80\%$ respectively, while $0$ and $1$ hold a proportion between $40\%$ and $60\%$ in other bits. (b) In NYU Depth v2, $0$ accounts for about $90\%$ in the bit associating with the biggest depth weight $5$, while $0$ and $1$ hold a proportion around $50\%$ in other bits. **Uniform Quantization in Log Space:** depth weight represents $\log(1 + \text{actual distance})$. (c) In KITTI, $0$ unevenly occupies only about $25\%$ in the first bit, and sightly unevenly accounts for $60\%$ in second bit, while $0$ and $1$ approximately equally distributed in other bits. (d) In NYU Depth v2, almost all bits distribute perfectly evenly except the first bit with the biggest depth weight, in which $0$ accounts for just above $40\%$.

and $1$ changes from about $19 : 1$ to about $1 : 4$ in the first bit. In NYU Depth V2, the percentage of $0$ in the bit with biggest depth weight, decreases from just above $90\%$ to just above $40\%$, and the following two bits with smaller depth weight become more uniform too. Therefore, by combining the technique of uniform quantization in log space with our encoding scheme, we can almost eliminate depth imbalance in the KITTI and NYU Depth V2 datasets.

*3) Depth Decoding:* The network usually outputs a likelihood that predicts depth bit $b_n$ is $1$. There are two strategies to decoding the probability to the depth. One is hard inference, which computes $b_n$ by $u(p_n - 0.5)$, where $u$ is the unit step function. Then we reconstruct the depth by,

$$\hat{q} = q_{1+\sum_{n=1}^{N} u(p_n - 0.5) 2^{n-1}} \tag{3}$$

However, although the hard inference is efficient in computation, is it less robust and accurate than soft inference than that compute an expectation of all possible cases to predict depth [21]. The expected depth by soft inference is computed by,

$$\hat{q} = \sum_{i_i \sim i_N} q_{1+\sum_{n=1}^{N} i_n 2^{n-1}} \prod_{n=1}^{N} p_n^{i_n} (1-p_n)^{1-i_n}, \ \ i_n \in \{0, 1\} \tag{4}$$

This formula has a complexity of $O(N2^N)$ or $O(L \log L)$. Fortunately, we simplify the complexity to $O(N)$ or $O(\log L)$. Suppose we uniformly quantize depth, so we can define max quantization error $\Delta \triangleq q_1 - b_0$. Then by using Equations 1 we have,

$$\begin{aligned} \hat{q} &= \sum_{i_1 \sim i_N} (q_1 + \sum_{n=1}^{N} i_n 2^n \Delta) \prod_{n=1}^{N} p_n^{i_n} (1-p_n)^{1-i_n} \\ &= q_1 + \Delta \sum_{n=1}^{N} 2^n \sum_{i_i \sim i_N} i_n \prod_{n=1}^{N} p_n^{i_n} (1-p_n)^{1-i_n} \\ &= q_1 + \Delta \sum_{n=1}^{N} 2^n p_n = q_1 + \sum_{n=1}^{N} p_n (d_{2^{n-1}} - d_0) \end{aligned} \tag{5}$$

Equations 5 also indicates every depth bit $b_n$ associated with a depth weight $\Delta 2^n$.

Furthermore, when depth is uniformly quantized in log space, the quantized value $g_l$ satisf,

$$q_l = \exp\left(\frac{\log d_l + \log d_{l-1}}{2}\right), \forall l \tag{6}$$

then we still have a similar simple form to decode the network output to depth,

$$\hat{q} = q_1 \prod_{n=1}^{N} \left(\frac{d_{2^{n-1}}}{d_0}\right)^{p_n} \tag{7}$$

However, instead of the quantized as Equations 6, Li *et al.* [21] just simply set it as the upper partition value and Fu *et al.* [9] set it as the mean value of its lower and upper partition values in linear space.

## B. Network and Objective

As we have formulated MDE as a classification problem, we should use a network architecture for dense classification problems in our implementation and the objective function consists of multiple binary classification losses.

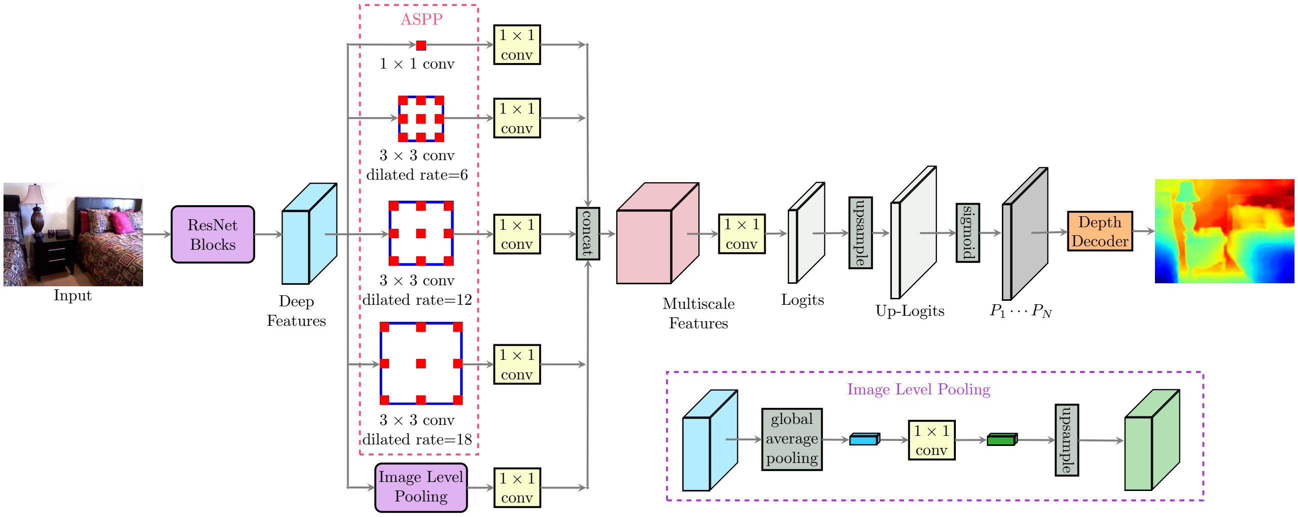


Fig. 4: **Depth Estimation Network Architecture.** The network consists of 3 parts. The first part is the ResNet blocks for extracting deep features, and atrous convolutions are applied in block3 and block4 to make $output\_stride = 8$. The second part is atrous spatial pyramid pooling (ASPP) and image pooling module followed by 1 × 1 convolutions to share information between different channels, and this part can transform deep features into multiscale features. The image-level pooling module contains 3 operations, global average pooling, a 1 × 1 convolution and bilinear upsampling. The last part is to transform the multiscale features into the predicted depth map. The multiscale features change to logits by a 1 × 1 convolution. Then bilinear upsampling and sigmoid function are applied on the logits to output the probability maps that predict the depth bit maps to be 1. At last, our proposed efficient decoding method will reconstruct a depth map from the probability maps.

*1) Depth Prediction Network:* In this paper we use a state-of-the-art semantic segmentation network, DeepLab V3 [5] which can efficiently enlarge receptive field by atrous or dilated convolution [3] and capture multi-scale information by atrous spatial pyramid pooling (ASPP) module [4], and we use ResNet [13] as backbone of the network. The depth prediction network architecture is illustrated in Figure 4.

*2) Objective Function:* In our approach, depth estimation is formulated as a problem consists of multiple binary classifications. We use conventional cross-entropy for every binary-class classification. We simply use a linear combination of these cross entropies of different depth bits as the objective function,

$$Loss = -\sum_{n=1}^{N} \lambda_n \left[ b_n \log p_n + (1 - b_n) \log(1 - p_n) \right] \quad (8)$$

where $\lambda$ is a weight to control the influence of different bits. It is reasonable to assign $\lambda$ proportional to the depth weight, which constitutes a geometric sequence with common ratio 2. We find it is better to initialize $\lambda$ in such way, and then flatten $\lambda$ in the training. We define $\lambda$ as following.

$$\lambda_n = \hat{\lambda}_n / \sum_{n=1}^{N} \hat{\lambda}_n, \;\; \hat{\lambda}_n = (1 + 100^{-iters/total})^n \quad (9)$$

where $iters$ represents the number of current iterations and $total$ represents the total number of iterations. In this way, we can make the network first coarsely focus on the bits with bigger depth weights and then refine the network with all depth bits altogether.

## III. Experiments

We demonstrate the effectiveness of our proposed approach by experiments on the publicly available KITTI [10] and NYU Depth V2 [26] datasets. Our experimental setup and results will be described as follow.

### *A. Experimental Setup*

**Datasets.** The KITTI dataset [10] is for testing computer vision algorithms in several tasks in the context of autonomous driving, contains data including stereo images and LiDAR point clouds, about outdoor scenes, like, “city”, “residential”, “road” and “campus”. There is an official split for MDE [29], containing 85898, 6852 and 500 pairs for training, validation, and testing. The official provided depth maps are much denser as neighboring 10 frames of point cloud are registered to the current frame and together reprojected to the camera.

The NYU Depth V2 [26] is a widely used indoor RGBD dataset in depth estimation and semantic segmentation. It contains 464 indoor video scenes taken by a Microsoft Kinect camera with a resolution of 640 × 480, of which 249 scenes are for training and 215 scenes for testing. 1449 pairs of aligned RGB and depth images were densely labeled, and depth maps have been filled by the colorization scheme of Levin *et al.* [20]. In the labeled pairs, 795 pairs are used for training and the rest 654 for testing. But about 407k frames were not labeled. We use the officially provided toolbox to process the unlabeled data and we obtain about 238k image-depth pairs for training. We still use the official testing in our evaluation.

TABLE I: **Quantitative Performance of classification based methods on the KITTI evaluation server.**

| Method | sqErrorRel | absErrorRel | Runtime | #Params |
|---|---|---|---|---|
| DORN [9] | 2.23 | 8.78 | 0.5 s | 98.76M |
| DABC [23] | 4.08 | 12.72 | 0.7 s | - |
| CSWS_E [21] | 3.48 | 11.84 | 0.2 s | - |
| HBC(Ours) | 3.79 | 12.33 | 0.05 s | 39.44M |

**Implementation Details.** We implement our depth prediction network using the TensorFlow framework [1] on a single Nvidia TITAN Xp GPU with 12 GB memory. We adopt ResNet-50 [13] as the deep feature extractor, and initialize the parameters in the root block and residual blocks with the pre-trained classification model on ILSVRC [6] and we fixed the parameters in the root block and the batch normalization parameters in the residual blocks. We discretize depth into 256 levels so that the number of bits in binary depth codes is 8 and after adding the ASPP module and convolution layers for output, the total number of parameters of the network is about 39.44M. We use an initial learning rate $1e-4$ and decrease it by 0.92 by every 10k steps. We train the network with using Adam optimizer [18] in TensorFlow , where $\beta_1 = 0.9$, $\beta_1 = 0.999$, and $\epsilon = 10^{-8}$.

In training, we do data augmentation like [8], but we have a bigger input size to make the ASPP module function well in extracting multiscale features. Specifically, we resize the original image to $266 \times 881$ instead of by half and randomly crop a $256 \times 848$ patch as network input for KITTI dataset and the downsampling size and input size for NVU Depth V2 are $300 \times 400$ and $288 \times 384$. We set the batch size for KITIT and NYU Depth V2 with 5 and 8, and train 40 and 10 epochs for them, costing about 36 and 60 hours respectively. In testing, We evaluate our results on pre-defined cropping by [8] for NVU Depth V2 depth dataset. We also train on KITTI official depth prediction split [29] and evaluate our approach the official evaluation server.

**Evaluation Metrics.** We adopt some conventional metrics in [8], [24] to evaluate our approach against valid ground truth depth values. Let $\hat{q}$ be the predicted depth and $d$ the ground truth depth for total $T$ pixels in an image. The metrics are: (1) Mean Absolute Relative Error ($rel$): $\frac{1}{T}\sum\frac{|\hat{q}-d|}{d}$; (2) Mean $\log_{10}$ Error ($\log_{10}$): $\frac{1}{T}\sum|\log_{10}\hat{q} - \log_{10}d|$; (3) Root Mean Squared Error ($rms$): $\sqrt{\frac{1}{T}\sum(\hat{q}-d)^2}$; (3) the accuracy with threshold $1.25^i$, i.e. the percentage of such that $\delta_i = \max(\frac{\hat{q}}{d}, \frac{d}{\hat{q}}) < 1.25^i$, where $i \in [1, 2, 3]$.

### B. Experimental Results

**Benchmark Performance.** We conduct experiments on both the KITTI and NYU Depth V2 datasets, and we will compare the performance of our approach with state of the art methods.

**KITTI.** Table I shows the results of our method trained on KITTI official split in comparison with several multi-label classification based methods. The ground truth of the official test set is not available and we have to upload the predicted depth maps to the KITTI online KITTI evaluation server [1]. Admittedly, our performance is inferior to these methods, especially DORN [9], as our approach is a simplified version of multi-label classification methods. DORN's superiority is due to the use of ordinal regression techniques, rather different from conventional multi-label classification because it considers the ordinal relation between discrete depth labels. However, the performance of HBC can approach the other two methods and our model has a much shorter runtime and smaller model size.

TABLE II: **Quantitative Performance on NYU Depth v2.**

| Method | Error | | | Accuracy | | |
|---|---|---|---|---|---|---|
| | rel | $\log_{10}$ | rms | $\delta_1$ | $\delta_2$ | $\delta_3$ |
| Make3D [25] | 0.349 | - | 1.214 | 0.447 | 0.745 | 0.897 |
| Li *et al.* [22] | 0.232 | 0.094 | 0.821 | 0.621 | 0.886 | 0.968 |
| Wang *et al.* [30] | 0.220 | - | 0.824 | 0.605 | 0.890 | 0.970 |
| Liu *et al.* [24] | 0.213 | 0.087 | 0.759 | 0.650 | 0.906 | 0.976 |
| Eigen *et al.* [8] | 0.215 | - | 0.907 | 0.611 | 0.887 | 0.971 |
| Eigen *et al.* [7] | 0.158 | - | 0.641 | 0.769 | 0.950 | 0.988 |
| Laina *et al.* [19] | 0.127 | 0.055 | 0.573 | 0.811 | 0.953 | 0.988 |
| MS-CRF [32] | 0.121 | 0.052 | 0.586 | 0.811 | 0.954 | 0.987 |
| Cao *et al.* [2] | 0.141 | 0.060 | 0.540 | 0.819 | 0.965 | 0.992 |
| CSWS_E [21] | 0.139 | 0.058 | 0.505 | 0.820 | 0.960 | 0.989 |
| DORN [9] | 0.115 | 0.051 | 0.509 | 0.828 | 0.965 | 0.992 |
| HBC (Ours) | 0.135 | 0.059 | 0.507 | 0.821 | 0.955 | 0.987 |

**NYU Depth V2.** HBC achieves comparable performance with the state of the art methods on the NYU Depth V2 dataset, as illustrated in Table II. Our approach can greatly outperform traditional methods by Saxena *et al.* [25]. Some early methods that combine CNNs and graphical models by Li *et al.* [22], Wang *et al.* [30] and Liu *et al.* [24] and the first method purely using deep learning by Eigen *et al.* [8] largely enhance the performance but the $rel$ error is still bigger than 20%. The second method by Eigen *et al.* [7] further advances the performance in a large extent. HBC can perform comparably with some recent state of the art methods, including two regression methods by Laina *et al.* [19] and Xu *et al.* [32], and two multi-label classification methods by Cao *et al.* [2] and Li *et al.* [21]. Compared to the most advanced DORN [9], HBC can approach it in almost all metrics except $rel$ and can win it in $rms$ sightly. As shown in Figure 5, compared with DORN [9] visually, HBC can also recover better structures in the predicted depths even with smaller complexity than earlier methods [8], [7], [19]. Therefore, HBC performs well in the indoor NYU Depth V2 dataset, which may largely due to the fact that this dataset shows unserious depth data imbalance in our encoding scheme.

## IV. Conclusion

In this paper, we present our formulation of hierarchical binary classification for monocular depth estimation by binarily encoding depth and the experiments indicating its effectiveness. We intend to to simplify the multi-label classification based methods and improve the efficiency, and HBC is proved to be not too inferior while showing much higher efficiency, especially on the NYU Depth V2 dataset. In the future, we will explore to embed HBC into a

[1] http://www.cvlibs.net/datasets/kitti/eval_depth.php?benchmark=depth_prediction

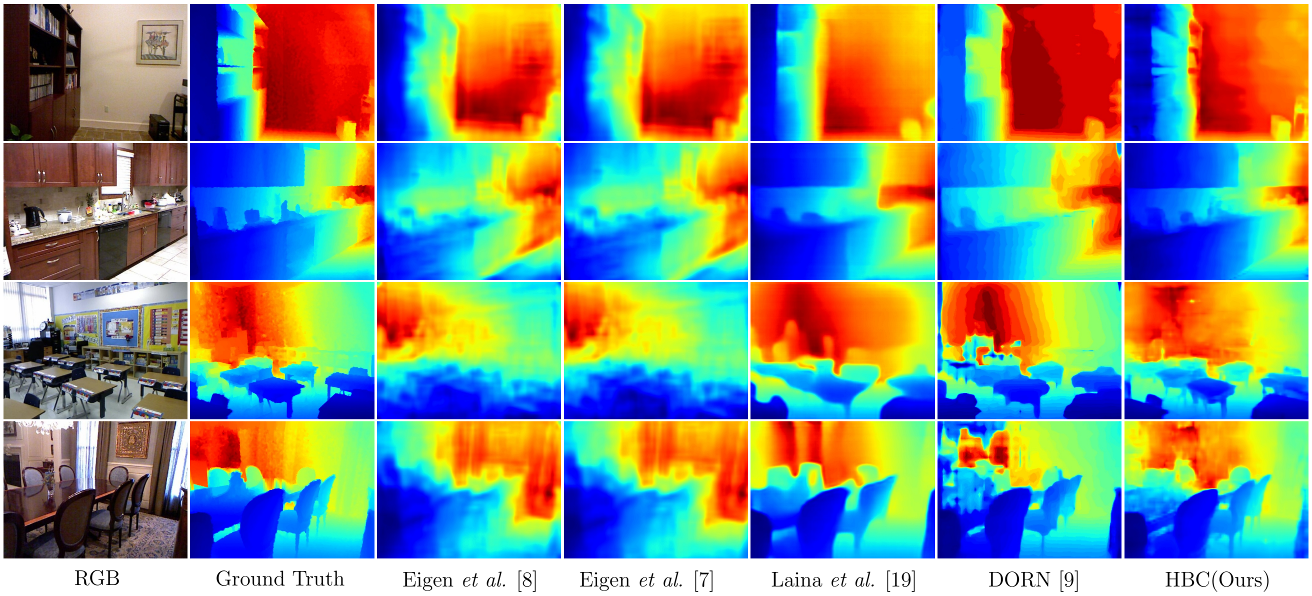


Fig. 5: **Qualitative Comparison on Some Examples of NYU Depth V2 Test Set.**

monocular visual SLAM system to increase the robustness for autonomous robot navigation.

## References

[1] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard, et al. Tensorflow: A system for large-scale machine learning. *arXiv preprint arXiv:1605.08695*, 2016.

[2] Y. Cao, Z. Wu, and C. Shen. Estimating depth from monocular images as classification using deep fully convolutional residual networks. *IEEE TCSVT*, 2017.

[3] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille. Semantic image segmentation with deep convolutional nets and fully connected crfs. *ICLR*, 2015.

[4] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE TPAMI*, 40(4):834–848, 2018.

[5] L.-C. Chen, G. Papandreou, F. Schroff, and H. Adam. Rethinking atrous convolution for semantic image segmentation. *arXiv preprint arXiv:1706.05587*, 2017.

[6] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, 2009.

[7] D. Eigen and R. Fergus. Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture. In *CVPR*, 2015.

[8] D. Eigen, C. Puhrsch, and R. Fergus. Depth map prediction from a single image using a multi-scale deep network. In *NIPS*, 2014.

[9] H. Fu, M. Gong, C. Wang, K. Batmanghelich, and D. Tao. Deep ordinal regression network for monocular depth estimation. In *CVPR*, 2018.

[10] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun. Vision meets robotics: The kitti dataset. *IJRR*, 32(11):1231–1237, 2013.

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

[12] R. Hartley and A. Zisserman. *Multiple view geometry in computer vision*. Cambridge university press, 2003.

[13] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016.

[14] H. Hirschmuller. Stereo processing by semiglobal matching and mutual information. *IEEE TPAMI*, 30(2):328–341, 2008.

[15] H. Jiang and R. Huang. High quality monocular depth estimation via a multi-scale network and a detail-preserving objective. In *2019 IEEE International Conference on Image Processing (ICIP)*, pages 1920–1924. IEEE, 2019.

[16] H. Jiang, Y. Ye, Z. Song, S. Tang, and Y. Dong. An automatic 3d textured model building method using stripe structured light system. In *International Conference on Computer Vision Systems*, pages 615–625. Springer, 2017.

[17] S. H. Khan, M. Bennamoun, F. Sohel, and R. Togneri. Geometry driven semantic labeling of indoor scenes. In *ECCV*, 2014.

[18] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[19] I. Laina, C. Rupprecht, V. Belagiannis, F. Tombari, and N. Navab. Deeper depth prediction with fully convolutional residual networks. In *3DV*, 2016.

[20] A. Levin, D. Lischinski, and Y. Weiss. Colorization using optimization. In *ACM TOG*. ACM, 2004.

[21] B. Li, Y. Dai, and M. He. Monocular depth estimation with hierarchical fusion of dilated cnns and soft-weighted-sum inference. *Pattern Recognition*, 2018.

[22] B. Li, C. Shen, Y. Dai, A. van den Hengel, and M. He. Depth and surface normal estimation from monocular images using regression on deep features and hierarchical crfs. In *CVPR*, 2015.

[23] R. Li, K. Xian, C. Shen, Z. Cao, H. Lu, and L. Hang. Deep attention-based classification network for robust depth prediction. *arXiv preprint arXiv:1807.03959*, 2018.

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

[25] A. Saxena, M. Sun, and A. Y. Ng. Make3d: Learning 3d scene structure from a single still image. *IEEE TPAMI*, 31(5):824–840, 2009.

[26] N. Silberman, D. Hoiem, P. Kohli, and R. Fergus. Indoor segmentation and support inference from rgbd images. In *ECCV*, 2012.

[27] S. Tang, X. Zhang, Z. Song, H. Jiang, and L. Nie. Three-dimensional surface reconstruction via a robust binary shape-coded structured light method. *Optical Engineering*, 56(1):014102, 2017.

[28] K. Tateno, F. Tombari, I. Laina, and N. Navab. Cnn-slam: Real-time dense monocular slam with learned depth prediction. In *CVPR*, 2017.

[29] J. Uhrig, N. Schneider, L. Schneidre, U. Franke, T. Brox, and A. Geiger. Sparsity invariant cnns. In *3DV*, 2017.

[30] P. Wang, X. Shen, Z. Lin, S. Cohen, B. Price, and A. L. Yuille. Towards unified depth and semantic prediction from a single image. In *CVPR*, 2015.

[31] T. Wiegand and H. Schwarz. *Source Coding: Part I of Fundamentals of Source and Video Coding*. Now Publishers Inc, 2011.

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