Real-time Monocular Depth Estimation with Extremely Light-Weight Neural Network

Mian-Jhong Chiu
Institute of Multimedia Engineering
National Chiao Tung University
Hsinchu County, Taiwan
0210028@gmail.com

Wei-Chen Chiu
Department of Computer Science
National Chiao Tung University
Hsinchu County, Taiwan
walon@cs.nctu.edu.tw

Hua-Tsung Chen
Department of Computer Science
National Chiao Tung University
Hsinchu County, Taiwan
huatsung@cs.nctu.edu.tw

Jen-Hui Chuang
Department of Computer Science
National Chiao Tung University
Hsinchu County, Taiwan
jchuang@cs.nctu.edu.tw

Abstract—**Obstacle avoidance and environment sensing are crucial applications in autonomous driving and robotics. Among all types of sensors, RGB camera is widely used in these applications as it can offer rich visual contents with relatively low-cost, and using a single image to perform depth estimation has become one of the main focuses in resent research works. However, prior works usually rely on highly complicated computation and power-consuming GPU to achieve such task; therefore, we focus on developing a real-time light-weight system for depth prediction in this paper. Based on the well-known encoder-decoder architecture, we propose a supervised learning-based CNN with detachable decoders that produce depth predictions with different scales. We also formulate a novel log-depth loss function that computes the difference of predicted depth map and ground truth depth map in log space, so as to increase the prediction accuracy for nearby locations. To train our model efficiently, we generate depth map and semantic segmentation with complex teacher models. Via a series of ablation studies and experiments, it is validated that our model can efficiently performs real-time depth prediction with only 0.32M parameters, with the best trained model outperforms previous works on KITTI dataset for various evaluation matrices.**

Keywords—Monocular depth estimation, Light-weight CNN, Real-time, Edge device, Supervised learning

I. INTRODUCTION

Depth estimation is an important task in computer vision, which generates depth maps of corresponding scenes to provide statistical representation of object appearance and the surrounding environment. Such information is crucial for a variety of applications such as autonomous driving, robotics, augmented reality, and 3D modeling.

Traditional approaches to pixel-wise depth estimation are achieved through stereo matching [1], which aims at finding the corresponding points in the stereo images for every image pixel to compute disparity and depth. However, multiple viewpoints are required for such methods. To overcome this limitation, many researches are focused on developing monocular depth estimation in recent years. Some supervised learning approaches [2, 3, 4, 29] attempt to predict pixel-wise depth map for the corresponding input image by using a CNN (Convolutional Neural Network) trained on a large-scale dataset, which contains RGB images and corresponding ground truth depth maps. However, the ground truth collection of pixel-wise depth map still remains a challenge due to the hardware limitation of LiDaR depth sensor. Specifically, the depth maps collected by LiDaR sensor are not only sparse but may also contain incorrect depth values due to transparent surfaces, moving objects, and occlusion in the scene. Using such incomplete and possibly incorrect training data, supervised learning approaches may fail to predict correct depth for the corresponding image pixels.

Accordingly, some other methods [5, 6, 7, 8] take alternative approaches by investigating unsupervised learning for depth prediction. By treating depth prediction as an image reconstruction problem, such methods can be trained without ground truth of depth map, while also achieving high prediction accuracy for inference. Recently, some prior works [9, 10, 11] also combine unsupervised and supervised loss during training to further improve the model performance. Despite the great success of these learning-based methods, their models are often designed with complex network structures and inefficient convolution layers, which is a major concern in nowadays real-time applications.

Therefore, methods proposed in [12, 13, 14] are focused on developing light-weight models for real-time depth prediction, involving efficient convolution layer design, network pruning, and assistance of teacher model. These methods have compressed their models to less than 10M parameters for real-time depth estimation on CPU. In this paper, we further investigate several light-weight layer designs of CNN and develop an extremely small model with less than 0.5M parameters for real-time and high-resolution depth estimation even on edge devices. To improve depth prediction accuracy of such small network, we implement a multi-task network structure that output depth map and semantic segmentation simultaneously during the training process. With the help of complex teacher models that generate dense depth map and semantic segmentation map in the training stage, the system performance in terms of depth prediction accuracy can be further improved. Evaluated on KITTI dataset, our best trained model not only has much fewer model parameters and much lower computation cost compared with other real-time methods, but also outperforms some previous works with several evaluation matrices. In summary, contributions of the proposed approach include:

- Design a light-weight CNN that can generate high resolution depth map in real-time with much less parameters and computations.
- Propose effective training strategies for such small model: (i) joint-training, (ii) data generation by complex teacher model, and (iii) using a novel, multi-resolution depth loss.

II. Related Work

A. *Learning Based Method for Depth Prediction*

In recent years, several learning-based approaches for single image depth estimation has been proposed. Eigen et al. [15] introduced the first supervised method, in which a coarse depth map is generated and refined using convolutional neural network. Following [15], Laina et al. [2] proposed an end-to-end deep neural network structure to produce dense depth map using indoor RGB-D image. They demonstrated the usage of fully convolutional network (FCN) for depth estimation and the efficient design of up-sampling layer in the encoder-decoder structure. Huan et al. proposed deep ordinal regression that treated depth estimation as a classification problem and outperformed previous supervised methods. Despite the success of aforementioned supervised methods, insufficient ground truth provided by LiDAR remains to be a problem. Therefore, some unsupervised approaches have been proposed to avoid using depth ground truth during training. Firstly, Garg et al. [11] proposed an unsupervised method by introducing photometric loss. Based on [11], Godard et al. [5] introduced a left-right consistency loss to enforce stereo reconstruction and achieved higher prediction accuracy. One can also adopt both supervised and unsupervised losses to further improve model performance. Inspired from previous works [2, 5], Kuznietsov et al. [9] proposed a semi-supervised deep learning approach by using sparse ground-truth depth map for supervised learning and stereo image construction loss for unsupervised learning.

Some other works have tried to tackle the problem of depth prediction with joint learning. As semantic segmentation also provides rich information of scene understanding and geometric representation, Chen et al. [8] introduced a training strategy that takes such information into consideration. They conducted left-right consistency loss for depth and semantic respectively based on [5]. Ramirez et al. [11] proposed a semantics-guided disparity smoothness loss that uses semantic information to improve disparity prediction. Nekrasov et al. [14] also trained a small model jointly with semantic and depth ground truth generated by a large teacher model. (Our model is also jointly trained using semantic and depth ground truth generated by a complex teacher model, which is similar to [14]. However, we adopt a multi-resolution loss that further expedite the training convergence and achieve similar accuracy with a much smaller network structure.)

While the aforementioned prior works enjoy high accuracy for depth prediction, their computation cost and number of model parameters are inevitably large. For example, the models presented in [5], [15], and [9] have about 30, 70, and 80 million parameters, respectively, making them hard to fit in edge devices for real-time inference. In this paper, we focus on developing highly efficient neural network that can achieve high prediction accuracy with much fewer parameters.

B. *Light-weight Neural Network Design*

In the past few years, the increasing needs of implementing high quality and real-time neural networks on edge devices have encouraged the research on more efficient network design. Several approaches [16, 17, 18] have been proposed to create efficient neural networks. In SqueezeNet, Iandola et al. [16] replace 3×3 convolution with pointwise convolution, and decrease the number of channels for convolution layers, so as to reduce computation cost. Presented in MobileNet [17], depthwise separable convolutions are adopted to decompose a standard convolution into depthwise and pointwise convolutions, and drastically increase model efficiency. Several vision tasks such as image classification and object detection have been implemented using MobileNet network as backend while achieving comparable results with less computation cost (and total number of model parameters) compared with other state-of-the-art works. Based on [17], MobileNetV2 [18] further improves the depthwise separable convolution by introducing the inverted residual (IR) and linear bottlenecks design (LBD). While IR increases the complexity of the information flow during depthwise convolution, which in turn increases the feature expressiveness of the model, LBD replaces the non-linear transform layer with a linear one to maintain the feature information in low dimension, after pointwise convolution is used to extract image features, and further improve the model performance. Inspired by MobileNetV2, a further streamlined encoder-decoder architecture is developed in this paper, which adopts similar concepts of IR and LBD, for depth estimation.

C. *Real-time Depth Estimation*

Based on the development of several light-weight neural network structure designs, some prior works for real-time depth prediction have been proposed in recent years. Poggie et al. [12] introduced a compact network designed with pyramid structure and achieved real-time performance on CPU with only 1.9M parameters. Elkerdawy et al. [13] trained a complex model first and pruned least important filters later by learned binary masks. Their method achieves better depth accuracy with 5.9M parameters. Using MobileNetV2 as framework, Nekrasov et al.

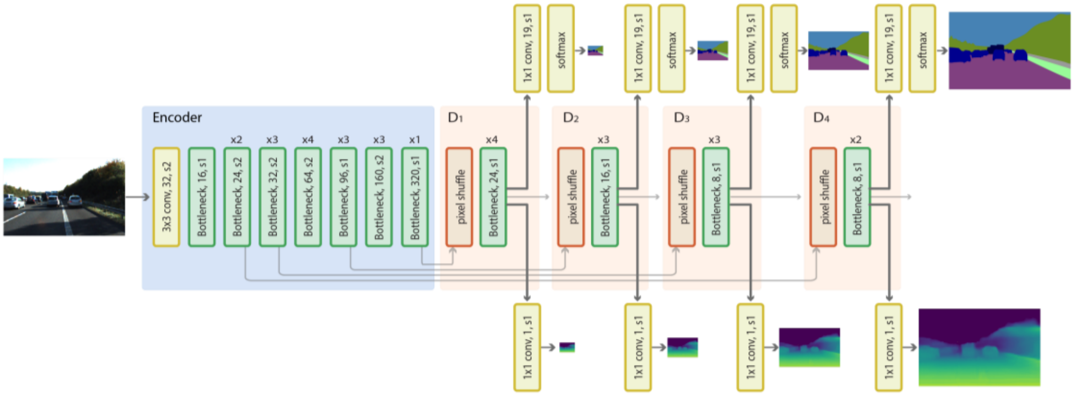
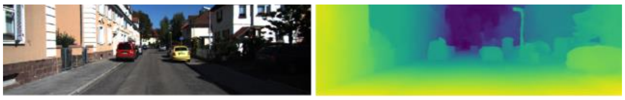


Fig. 3. The proposed CNN architecture. Each layer is specified by layer name, number of channels, and strike size. The multiplication (×) on top of a layer gives the repetition of that layer.



(a) (b)

Fig. 1. An example of the pre-processed depth maps on KITTI dataset. (a) Input image. (b) Depth map generated by PSMNet.

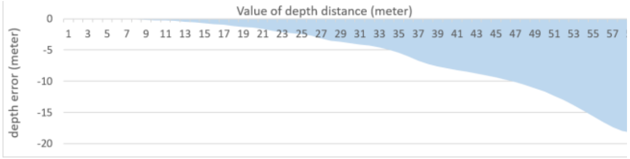


Fig. 2. Error distribution per distance of PSMNet depth map.

[14] proposed an efficient model which is jointly trained using depth map and semantic segmentation. By using large pretrained teacher model to produce depth and semantic segmentation ground truth for training, their model achieved high accuracy with only 2.99M parameters. Our proposed method is similar to Nekrasov et al., which is also jointly trained with ground truth provided by complex teacher networks. However, with a much more compact structure design and the usage of multi-task loss under multiple resolutions, we further push forward the limits of real-time depth prediction and achieve real-time performance on edge device with only 0.32M parameters and slightly reduce of depth accuracy.

III. Approach

In this section, our light-weight neural network for real-time monocular depth estimation is described. We first introduce depth map and segmentation map generation by complex teacher model on KITTI dataset, which can speed up the model convergence during training. Then, we propose a highly efficient architecture design for depth prediction, which has lower computation cost and requires less model parameters compared with other state-of-the-art methods. Finally, we demonstrate the design of a multi-task loss function which is composed of depth loss and semantic segmentation loss.

A. *Data Pre-processing*

In this paper, we train, evaluate, and compare our approach with previous works on the public KITTI dataset, wherein the data are collected in the streets of cities and campuses. During the training process, a multi-task learning method is conducted to generate depth map and semantic segmentation, wherein the aforementioned multi-task loss function is evaluated to update the model parameters. Therefore, three types of data are needed for the training process: (i) original images as model input, (ii) semantic segmentation of (i) as one type of ground truth, and (iii) depth map of (i) as another type of ground truth.

For (i), we use monocular RGB images in KITTI raw dataset as the input of our CNN model. For (ii), since the raw data provided by KITTI do not include the annotations for semantic segmentation, we generate such segmentation data using a teacher model - DeepLabV3 [19], a CNN model which generates the semantic segmentation with pixel-wise label for 19 different classes for each input image. As for the generation of depth ground truth in (iii), we employ Pyramid Stereo Matching Network (PSMNet) [20] as a teacher model, which uses stereo images pair from the KITTI raw data as input and generates disparity maps as output. Such maps are then converted to depth maps and used as ground truth during the training process. An example of the depth map thus generated is shown in Fig. 1.

However, the above PSMNet depth map is generated via stereo matching, which is certainly not the measured ground truth; therefore, we further evaluate the PSMNet depth map by comparing it with the sparse ground truth depth map measured with a LiDaR (provided in the KITTI dataset). In particular, we subtract PSMNet depth map from the sparse depth map to compute mean error per distance for the training set, as shown in Fig. 2. It is readily observable that the mean error is always negative with its magnitude increasing with the distance value. To compensate such error, a look-up table is built from Fig. 2 and used to adjust the depth value of each pixel of the PSMNet depth map to a more accurate depth value.

B. *Network Architecture*

The proposed CNN design is based on the well-known encoder-decoder structure. We choose MobileNetV2 (MNv2) as our encoder. The decoder is designed with up-scaling layers to enlarge the extracted feature maps, to form high resolution outputs of depth estimation as well as the corresponding semantic segmentation, and will be elaborated next.

To design a light-weight CNN structure for real-time applications, an up-scaling method need to be chosen for our decoder design. Three common up-scaling strategies in neural network design include: (i) deconvolution, (ii) unpooling, and (iii) pixel shuffle. With (i), a deconvolution layer will up-scale the feature map by reversing the standard convolution process with a stride equal to 2, which maintains good gradient information but has high computation costs due to gradient descent-based updates of its kernel weights. For (ii), the feature map is enlarged directly by filling in zeros or repetitive values to empty cells of the enlarged feature map, which has less multiplication-add operations but will increase extraneous information flow while filling in unnecessary values. As for (iii), as presented by Shi et al. [21], the feature map is up-scaled by reshaping the feature matrix, e.g., a feature map with size (H, W, C) can be reshape to (2H, 2W, C/4) by a pixel shuffle layer. The feature reordering process can be efficiently done without any multiplication-add computation. Furthermore, the enlarged feature map will have all original feature values, maintaining good information flow in the neural network. Thus, strategy (iii) is adopted in the proposed decoder for feature map up-scaling.

The proposed neural network architecture is shown Fig. 3, wherein each layer is denoted with its layer name, number of output channels, and strides (s), while the mark (×) on top of each layer gives the repetition of that layer. The encoder is composed of one 3×3 full convolution layer and 17 bottlenecks, which first down-samples input RGB image and eventually outputs a set of low-resolution and highly expressive feature maps. On the other hand, the decoder is represented by four regions (D1~D4), each composed of a pixel shuffle layer and 2 to 4 bottlenecks, which eventually generates depth map and semantic segmentation map with multiple resolutions. For each of the four decoder blocks, a feed-forward feature map from previous layer is concatenated with a feature map passed from some encoder layers via skip connections and fed into a pixel shuffle layer for up-sampling. Next, the feature maps are fed into the bottlenecks to increase the feature expressiveness before further passed to two separate output layers, one for depth prediction and the other for semantic segmentation. While the former is done by using a pointwise convolution that compresses high-dimensional feature map into a single channel output as the pixel-wise depth map, the latter is done by using a pointwise convolution and a Softmax function that outputs a feature map with 19 channels, each represented by the probability of a specific category of segmentation.

The motivation of the multi-task decoder design is that certain dependency and relationship do exist between many vision tasks [22]. With such design, we explore the possibility of learning depth estimation together with semantic segmentation to further improve the performance of depth estimation task, as will be observed from some results provided in experiments, while the multi-scale decoder design is also shown to be able to improve the model performance by improving the features of each resolution via the consideration of a multi-scale loss, as discussed next. In addition, skip connections between corresponding layers in encoder and multi-scale decoder also benefit the information flow and speed up the model convergence. Finally, the proposed multi-scale multi-task model is detachable and highly customizable for different application needs after the training procedure, which enables us to deal with more easily the trade-off between resolution, run-time speed, and types of output.

C. Loss function

In this paper, a loss function $\mathcal{L}_{total}$ (1), formulated as a weighted sum of depth loss and segmentation loss, is applied to each decoder block jointly for depth prediction as well as semantic segmentation, i.e.,

$$\mathcal{L}_{total} = \sum_{i=1}^{4} \alpha_i \mathcal{L}_{D_i} + \beta_i \mathcal{L}_{S_i}, \tag{1}$$

where parameter i denotes the block number of the multi-scale decoder, $\mathcal{L}_{D_i}$ and $\mathcal{L}_{S_i}$ give the depth loss and segmentation loss for block i, with α_i and β_i being their weights and set to 0.25 and 0.75, respectively.

The depth loss $\mathcal{L}_{D_i}$ measures the average difference of depth value between predicted depth map and ground truth depth. Most prior works [2, 4, 9] design the depth loss function by using L1 distance or L2 distance directly to represent the difference between predicted and the ground truth depth. However, error values should be considered differently for nearby and distant pixels. Specifically, the depth difference for nearby pixels should be more sensitive to depth difference than distant pixels. By analyzing the depth value distribution of ground truth depth map on KITTI dataset, we discovered that most of ground truth pixels has small depth value while only few pixels have large depth value, which means that the model should be more focused on the depth prediction for nearby objects. Therefore, our novel depth loss focuses more on nearby pixels by computing the difference of predicted and ground truth depth values in log space. To transform depth values from linear space to log space, we apply a log transform $G(d)$ to each pixel of depth map. Moreover, BerHu normalization $F(x)$ [23] is also used to balance the training convergence between nearby depth and distant depth. $\mathcal{L}_{D_i}$ (2), $F(x)$ (3), and $G(d)$ (5) can be represented as:

$$\mathcal{L}_{D_i} = \frac{1}{W_i H_i} \sum_{x=1}^{W_i} \sum_{y=1}^{H_i} F\big(G(\tilde{d}_{x,y}) - G(d_{x,y})\big), \tag{2}$$

$$F(x) = \begin{cases} |x| & |x| \leq c, \\ \frac{x^2+c^2}{2c} & |x| > c. \end{cases} \tag{3}$$

$$c = \frac{1}{5} \max_i \big(\big|G(\tilde{d}_{x,y}) - G(d_{x,y})\big|\big), \tag{4}$$

$$G(d) = \frac{(\log d - \log m) \times M}{\log M - \log m}; m = 4, M = 80, \tag{5}$$

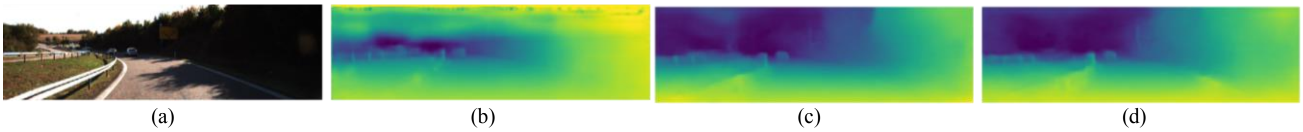


Fig. 4. An example of depth prediction results trained with various pre-processed data. (a) Input image. (b), (c), and (d) are the predicted results trained with Sparse depth map, Dense depth map, PSMNet (origin), and PSMNet (compensated), respectively.

TABLE I. EVALUATION OF MODELS TRAINED WITH DIFFERENTLY PRE-PROCESSED TRAINING DATA.

Data type	Training data	RMSE (meter)
Baseline	Sparse depth map	4.922
Pre-processed	PSMNet (origin)	4.716
	PSMNet (compensated)	3.945

TABLE II. EVALUATION OF MODELS TRAINED WITH DIFFERENTLY PRE-PROCESSED TRAINING DATA.

Improvement			RMSE (meter)
Depth Teacher	*Log Depth*	*Segment Teacher*	
✓			3.945
✓	✓		3.884
✓	✓	✓	3.871

where $W_i \mathrm{x} H_i$ gives the resolution of output depth map from decoder block i , $\tilde{d}_{x,y}$ and $d_{x,y}$ give depth values for the corresponding pixels of output and ground truth, respectively. For the BerHu normalization, $F(x)$, the boundary c between L1 distance and L2 distance is determined by $\tilde{d}_{x,y}$ and $d_{x,y}$ in current batch. As for the log transform function $G(d)$, the lower bound m and upper bound M are set to 4 and 80, respectively, according to the limitation of the LiDaR sensor.

The segmentation loss $\mathcal{L}_{S_i}$ is defined as the pixel-wise categorical cross-entropy:

$$\mathcal{L}_{S_i} = -\frac{1}{W_iH_i}\sum_{x=1}^{W_i}\sum_{y=1}^{H_i}\sum_{c=1}^{C}\tilde{s}_{x,y}^{c}\log s_{x,y}^{c}, \tag{5}$$

where C denotes the number of classes for semantic segmentation, $\tilde{s}_{x,y}^{c}$ and $s_{x,y}^{c}$ respectively give the predicted segmentation and ground truth segmentation of class c at the corresponding pixel location (x, y).

IV. EXPERIMENTS

A. Implementation details

In our experiments, the MNv2-based encoder of the proposed CNN network is initialized with pretrained weights for the ImageNet classification task, while the convolution filters in the decoder blocks are initialized with normal initializer. For the gradient descent procedure in the training process, Adam optimizer is employed with an initial learning rate of 0.01, and with the rate reduced by 10% for every epoch. The proposed network is trained for at least 20 epochs, with 1000 iterations for each epoch and a batch size of 8, on NVIDIA TITAN Xp with 12GB of memory. All the experiments are conducted using Keras as high-level API and TensorFlow as the backend platform.

To train the CNN on KITTI dataset, we first downsample RGB images from the original size of (1241, 376) to (1056, 320) as the model input, while resolutions of the depth maps and semantic segmentation outputs of decoder blocks D1~D4 are (66, 20), (132, 40), (264, 80), and (528, 160), respectively. For performance evaluation, we upsample all the predicted depth maps back to (1241, 376), i.e., the original resolution of the ground truth depth.

For the purpose of designing a light-weight model with high efficiency, we set the hyper-parameter, i.e., width multiplier of the bottlenecks, to 0.35, which is used to uniformly down-scale the number of channels for each convolution layer. The total number of parameters of our model are thus reduced to 0.32 million, with the multiplication-add operations of the model limited to 2.1 GFLOPs. The average inference time, for one input image is 21ms on TITAN Xp GPU.

B. Results

1) Ablation Study: To demonstrate that our approach does benefit from using depth map generated by a teacher model, different approaches mentioned in Sec.Ⅲ-A. are evaluated for the depth prediction results obtained with the proposed CNN model. As shown in Table 1, the model trained directly with sparse ground truth depth map (SparseGT) is regarded as baseline, which results in an RMSE of 4.922 meters. However, the prediction results trained with SparseGT often give false predictions near object boundaries and upper part of the corresponding input images, as shown in Fig. 4. This is because such ground truths are lack of depth value at the upper part due to LiDaR projection limitation, while occlusion caused by sensor fusion between RGB camera and LiDaR often occur near the image boundary.

To overcome the above problems, PSMNet is employed to generate pixel-wise depth map as training ground truth for the training of our model. Nevertheless, the PSMNet depth map is not the real ground truth and has a biased estimation with respect to different distances, and the model trained with PSMNet depth map results in 4.716 meters in RMSE. After using the look-up table mentioned in Sec. Ⅲ-A to compensate for the bias in the depth estimation of PSMNet, the model achieves an RMSE of 3.945 meters and outperforms other preprocessing methods. It is readily observable from Fig. 4 that the models trained with PSMNet depth map predicts more accurate depth value at image boundaries and the upper part of the image. Moreover, sharper object boundaries can also be obtained compared with the results generated by models trained with SparseGT.

TABLE IV. COMPARED WITH PREVIOUS WORKS ON KITTI DATASET.

Model	Abs Rel	Sq Rel	RMSE	RMSE log	Threshold		
					<1.25	<1.56	<1.95
Kuznietsov et al. [9]	0.113	0.741	4.621	0.189	0.862	0.960	0.986
Godard et al. [5]	0.133	1.158	5.370	0.208	0.841	0.949	0.978
Eigen et al. [15]	0.203	1.548	6.307	0.282	0.702	0.890	0.958
Liu et al. [4]	0.201	1.584	6.471	0.273	0.680	0.898	0.967
Luo et al. [24]	0.094	0.626	4.252	0.177	0.891	0.965	0.984
Godard et al. [6]	0.114	0.991	5.029	0.203	0.864	0.951	0.978
Luo et al. [25]	0.128	0.935	5.011	0.209	0.831	0.945	0.979
Yin et al. [7]	0.149	1.060	5.567	0.226	0.796	0.935	0.975
Guo et al. [26]	0.111	0.771	4.449	0.185	0.868	0.958	0.983
Yang et al. [28]	0.092	0.547	3.390	0.177	0.898	0.962	0.982
Amiri et al. [10]	0.078	0.417	3.464	0.126	0.923	0.984	0.995
Fu et al. [3]	**0.072**	**0.307**	**2.727**	**0.120**	**0.932**	0.984	0.994
Ours	0.106	0.502	3.871	0.160	0.897	**0.998**	**0.9997**

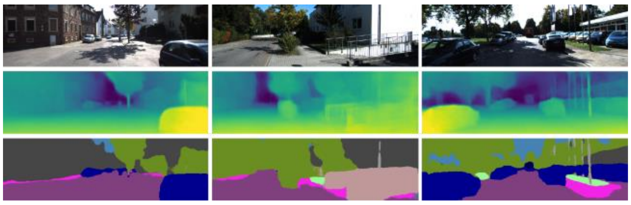


Fig. 5. Some qualitative results of our multi-tasks model. Top to bottom: input images, predicted depth map, and predicted semantic segmentation.

TABLE III. EVALUATION OF MODELS TRAINED WITH VARIOUS DECODER STRUCTURES.

Decoder Structure	**RMSE (meter)**
(N, 0)	4.167
(N-1, 1)	4.225
(N-2, 2)	4.465
(N-3, 3)	5.222

Next, we conduct another experiment to evaluate loss functions designed with (i) ordinary depth map and (ii) the depth map re-mapped by $G(d)$ (log depth map). To evaluate the depth prediction result, the outputs of the model trained (and tested) with log depth maps are inversely re-mapped to the original depth values by using $G^{-1}(d)$ for RMSE computation. According to the results shown in Table 2, evaluated with RMSE, the model trained with log depth map performs better than that trained with original depth map, i.e., with 6 cm decreases in RMSE.

Finally, we explore possible enhancements of depth estimation via the employment of multi-task learning that not only predicts depth map but also gives semantic segmentation during training. Based on the original model for single task, we investigate several designs by modifying the decoder blocks for multi-task prediction. Since a decoder block is composed of one pixel shuffle layer and N bottlenecks, we split the bottlenecks into m mutual layers and n individual layers for multi task learning, where the formers share the same weights and produce the same feature maps while the latter have independent weights and produce individual feature maps for each individual task. We conduct an experiment with four different settings of (m, n) for mutual convolution and individual convolution, including $(N, 0)$, $(N-1, 1)$, $(N-2, 2)$, and $(N-3, 3)$, as shown in Table. 3. We train each model for 10 epochs, and compare the depth loss $\mathcal{L}_D$ of the last epoch, which are 4.167, 4.225, 4.465, and 5.222, respectively, for the above four (m, n) setting. We can conclude that with same amount of bottleneck layers, increasing mutual layers between tasks can increase the model performance. Therefore, decoder blocks designed with N mutual bottlenecks and 0 individual bottleneck are used for our multi-task learning approach, as in the network architecture shown in Fig. 3.

We then compare the depth prediction results of single task and multi-task learning models. According to the results shown in Table 2, the model trained for both depth prediction and semantic segmentation outperforms the model only trained for depth prediction, i.e., with 1.3 cm decreases in RMSE. One possible explanation of such results is that the model trained for multiple tasks will have higher feature expressiveness and expedite the convergence of the training process, which in turn improves the depth estimation for all distance ranges.

2) Evaluation on KITTI dataset: With all improvements described in the previous subsection, we can now use KITTI public dataset to compare the performance of our method with previous works in terms of depth prediction accuracy and model size. Fig. 5 shows some qualitative results of the prediction of depth map and semantic segmentation on KITTI dataset by using our best trained multi-task multi-loss model. The outputs of depth maps are illustrated with color map, with brighter color corresponding to a closer distance, whereas the outputs of semantic segmentation are illustrated by assigning different colors to differently identified classes. One can also see that the proposed model can obtain accurate depth maps with clear object boundaries for small (far away) objects.

To compare the performance with previous works, we use our best trained model to predict depth maps for the testing

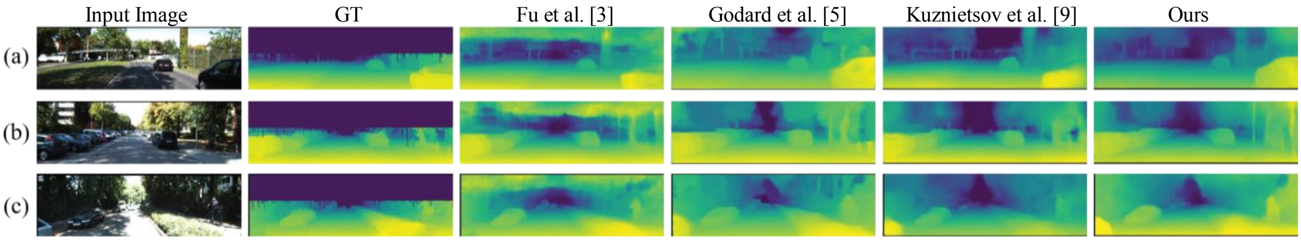


Fig. 6. Qualitative results on KITTI dataset. GT gives the ground truth depth, which is pre-processed with interpolation for visualization.

TABLE V. THE COMPARISON OF PREDICTION ERROR AND TOTAL USAGE OF MODEL PARAMETERS.

Method	**RMSE**	**Parameters (M)**	**Ratio**
Elkerdawy et al. [13]	5.891	5.9	×18.4
Poggi et al. [12]	6.030	1.9	×5.9
Nekrasov et.al [14]	**3.453**	2.99	×9.3
Ours	3.871	**0.32**	**×1.0**

TABLE VI. EVALUATION OF OUR MODEL ON GTX 1060 AND JETSON TX2.

Model	Output dim.	GTX 1060 (FPS)	Jetson TX2 (FPS)	RMSE
L	(240, 160)	121.6	33.5	4.315
M	(120, 80)	148.7	42.8	4.344
S	(60, 40)	174.8	49.5	4.619
XS	(30, 20)	179.4	54.1	4.549

images of KITTI dataset and compare the evaluation results with various approaches of monocular depth estimation, as shown in Table 4. For each evaluation metric, bold numbers and underlined numbers represent the best and the second-best results, respectively. Our model outperforms all the others for Threshold <1.56 and Threshold <1.95. Note that the current state-of-the-art method proposed by Fu et al. [3] is conducted by using ordinal regression, which has high computation cost and takes more than 500ms (less than 2 fps) for inference, while our model can produce high resolution outputs with low computation cost and only takes 21ms (47 fps) for inference.

Fig. 6 illustrates qualitative results of predicted depth map produced by our method and those proposed by Godard et al. [5], Kuznietsov et al. [9], and Fu et al. [3], wherein GT gives ground truth depth map post-processed by interpolation for better visualization, with depth prediction results of each method rescaled to the resolution of input image, or $(1241, 376)$. The unsupervised learning method proposed in [5] gives depth prediction with rich detail. However, the object boundary of closed by objects has blurrier depth prediction results compared with our method. The method proposed in [9] also fails to predict correct depth for nearby objects. Moreover, the depth prediction of smaller (thinner) objects in the scene are more blurred compared with the result of our method, e.g., the lamp pole shown in Fig. 6 (b). As for the current state-of-the-art method proposed in [3], despite their high evaluation score, the depth prediction gives random value at the upper part of the image due to insufficient of ground truth value. On the other hand, our method not only gives detailed depth prediction for all image pixels, but also provides correct prediction and clear boundaries for nearby objects, and even small objects.

We further evaluate the model size (number of model parameters) of our method and several previous works of real-time depth estimation, including the methods proposed by Elkerdawy et al. [13], Poggi et al. [12], and Nekrasov et.al [14] on KITTI dataset, as shown in Table 5. One can see our (single task) model only has a total number of 0.32 million parameters, which is the least among all methods, i.e., 18.4, 5.9, and 9.3 times fewer than that of Elkerdawy, Poggi, and Nekrasov's method, respectively. As for the comparison of depth prediction error evaluated with the RMSE metric on KITTI dataset (which is also available in Table 5), our model outperforms all models except for the one proposed by Nekrasov et.al, which has a much more complex model than ours.

3) Evaluation on Edge Device: For realistic evaluation of the proposed light-weight design for computation time and model performance, we implement our model on NIVIDIA Jetson TX2 module, a power-efficient embedded AI computing device. To achieve low latency and high-throughput for inference applications, we convert our well-trained model from Keras framework to TensorFlow then optimize it using TensorRT (TRT), which combines selected layers and optimizes the kernel selection for throughput, power efficiency, and memory consumption. Finally, we evaluate the model performance (in terms of FPS), before and after the TRT optimization, on NVIDIA GTX 1060 GPU and Jetson TX2 module, as shown in Table 6.

Aiming for realistic edge applications, we provide four models of different sizes, with different computation efficiency and prediction accuracy. In particular, L, M, S, and XS represent the models with 4, 3, 2, and 1 decoder blocks, respectively. With the input resolution set to $(480, 320)$, output resolutions of L, M, S, and XS models are set to $(240, 160)$, $(120, 80)$, $(60, 40)$, and $(30, 20)$, respectively. On GTX 1060, all models can easily achieve much higher than real-time performance (>100fps) with a TensorFlow-based framework, and can further increase the inference speed (>121 fps) via TensorRT optimization. On Jetson TX2, on the other hand, all models can achieve slightly higher than real-time performance, except for the TensorFlow implementation of the model which has the highest output resolution of $(240,160)$.

V. CONCLUSION

We proposed an efficient CNN for depth estimation with real-time (over 45fps) processing rate. Our method is designed with detachable multi-resolution decoder blocks that output pixel-wise depth map and semantic segmentation with multiple scales. The detachable structure enables model customization, offering the trade-off between output resolution and computation cost (speed). To train our CNN with supervised learning techniques on KITTI dataset, we generate ground truth semantic segmentation by using DeepLabV3, and produce ground truth depth map by using PSMNet and a depth compensation scheme. Based on an encoder-decoder architecture, we also explore efficient convolution design, low-computation upsampling layers, and a series of decoder structures. Our best trained model achieves state-of-the-art performance in several evaluation matrices on KITTI dataset with extremely small model size. Finally, via the renowned TensorFlow conversion and TensorRT optimization, we show that our compressed model can perform in real-time not only on GPU but also on power-efficient computing device, and will have a great opportunity for various edge computing applications.

ACKNOWLEDGMENT

This research is supported in part by Ministry of Science and Technology under MOST 109-2634-F-009-015 and MOST 109-2636-E-009-018.

REFERENCES

[1] S. Birchfield, and C. Tomasi, “Multiway cut for stereo and motion with slanted surfaces,” in Proc. of IEEE Conference on Computer Vision, vol. 1, pp. 489-495, 1999

[2] I. Laina, C. Rupprecht, V. Belagiannis, F. Tombari, and N. Navab, “Deeper depth prediction with fully convolutional residual networks,” in Proc. of IEEE Conference on 3D Vision, pp. 239-248, 2016

[3] H. Fu, M. Gong, C. Wang, K. Batmanghelich, and D. Tao, “Deep ordinal regression network for monocular depth estimation,” in Proc. of IEEE Conference on Computer Vision and Pattern Recognition, pp. 2002-2011, 2018

[4] F. Liu, C. Shen, and G. Lin, “Deep convolutional neural fields for depth estimation from a single image,” in Proc. of IEEE Conference on Computer Vision and Pattern Recognition, pp. 5162-5170, 2015

[5] C. Godard, O. Mac Aodha, and G. J. Brostow, “Unsupervised monocular depth estimation with left-right consistency,” in Proc. of IEEE Conference on Computer Vision and Pattern Recognition, vol. 2, no. 6, pp. 7, 2017

[6] C. Godard, O. Mac Aodha, and G. J. Brostow, “Digging into self-supervised monocular depth estimation,” arXiv preprint arXiv:1806.01260, 2018

[7] Z. Yin, and J. Shi, “Geonet: Unsupervised learning of dense depth, optical flow and camera pose,” in Proc. of IEEE Conference on Computer Vision and Pattern Recognition, pp. 1983-1992, 2018

[8] P. Y. Chen, A. H. Liu, Y. C. Liu, and Y. C. F. Wang, “Towards scene understanding: Unsupervised monocular depth estimation with semantic-aware representation,” in Proc. of IEEE Conference on Computer Vision and Pattern Recognition, pp. 2624-2632, 2019

[9] Y. Kuznietsov, J. Stückler, and B. Leibe, “Semi-supervised deep learning for monocular depth map prediction,” in Proc. of IEEE Conference on Computer Vision and Pattern Recognition, pp. 2215-2223, 2017

[10] A. J. Amiri, S. Y. Loo, and H. Zhang, “Semi-supervised monocular depth estimation with left-right consistency using deep neural network,” in IEEE International Conference on Robotics and Biomimetics, pp. 602-607, 2019

[11] P. Z. Ramirez, M. Poggi, F. Tosi, S. Mattoccia, and L. Di Stefano, “Geometry meets semantics for semi-supervised monocular depth estimation,” in Asian Conference on Computer Vision, pp. 298-313, 2018 R. Garg, V. K. Bg, G. Carneiro, and I. Reid, “Unsupervised cnn for single view depth estimation: Geometry to the rescue,” in European conference on computer vision, pp. 740-756, 2016

[12] M. Poggi, F. Aleotti, F. Tosi, and S. Mattoccia, “Towards real-time unsupervised monocular depth estimation on cpu,” in Proc. of IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 5848-5854, 2018

[13] S. Elkerdawy, H. Zhang, and N. Ray, “Lightweight monocular depth estimation model by joint end-to-end filter pruning,” in Proc. of IEEE International Conference on Image Processing, pp. 4290-4294, 2019

[14] V. Nekrasov, T. Dharmasiri, A. Spek, T. Drummond, C. Shen, and I. Reid, “Real-time joint semantic segmentation and depth estimation using asymmetric annotations,” in Proc. of IEEE Conference on International Conference on Robotics and Automation, pp. 7101-7107, 2019

[15] D. Eigen, C. Puhrsch, and R. Fergus, “Depth map prediction from a single image using a multi-scale deep network,” in Proc. of Advances in Neural Information Processing Systems, pp. 2366-2374, 2014

[16] F. N. Iandola, S. Han, M. W. Moskewicz, K. Ashraf, W. J. Dally, and K. Keutzer, “SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and< 0.5 MB model size,” arXiv preprint arXiv:1602.07360, 2016

[17] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, “Mobilenets: Efficient convolutional neural networks for mobile vision applications,” arXiv preprint arXiv:1704.04861, 2017

[18] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L. C. Chen, “Mobilenetv2: Inverted residuals and linear bottlenecks,” in Proc. of IEEE Conference on Computer Vision and Pattern Recognition, pp. 4510-4520, 2018

[19] 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,” in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 40, no. 4, pp. 834-848, 2018

[20] J. R. Chang, Y. S. Chen, “Pyramid stereo matching network,” in Proc. of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 5410–5418, 2018

[21] W. Shi, J. Caballero, F. Huszár, J. Totz, A. P. Aitken, R. Bishop, D. Rueckert, and Z. Wang, “Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network,” in Proc. of IEEE Conference on Computer Vision and Pattern Recognition, pp. 1874-1883, 2016

[22] A. R. Zamir, A. Sax, W. Shen, L. J. Guibas, J. Malik, and S. Savarese, “Taskonomy: Disentangling task transfer learning,” in Proc. of IEEE Conference on Computer Vision and Pattern Recognition, pp. 3712-3722, 2018

[23] L. Zwald, and S. Lambert-Lacroix, “The berhu penalty and the grouped effect,” arXiv preprint arXiv:1207.6868, 2012

[24] Y. Luo, J. Ren, M. Lin, J. Pang, W. Sun, H. Li, and L. Lin, “Single view stereo matching,” in Proc. of IEEE Conference on Computer Vision and Pattern Recognition, pp. 155-163, 2018

[25] C. Luo, Z. Yang, P. Wang, Y. Wang, W. Xu, R. Nevatia, and A. Yuille, “Every pixel count++: Joint learning of geometry and motion with 3D holistic understanding,” arXiv preprint arXiv:1810.06125, 2018

[26] X. Guo, H. Li, S. Yi, J. Ren, and X. Wang, “Learning monocular depth by distilling cross-domain stereo networks,” in Proc. of European Conference on Computer Vision, pp. 484-500, 2018

[27] A. Atapour-Abarghouei, and T. P. Breckon, “Real-time monocular depth estimation using synthetic data with domain adaptation via image style transfer,” in Proc. of IEEE Conference on Computer Vision and Pattern Recognition, pp. 2800-2810, 2018

[28] N. Yang, R. Wang, J. Stuckler, and D. Cremers, “Deep virtual stereo odometry: Leveraging deep depth prediction for monocular direct sparse odometry,” in Proc. of European Conference on Computer Vision, pp. 817-833, 2018

[29] A. Saxena, S. H. Chung, and A. Y. Ng, “Learning depth from single monocular images,” in Proc. of Advances in Neural Information Processing Systems, pp. 1161-1168, 2006