2018 IEEE Intelligent Vehicles Symposium (IV)
Changshu, Suzhou, China, June 26-30, 2018

Understanding 3D Semantic Structure around the Vehicle with Monocular Cameras

Kenichi Narioka[1] Hiroki Nishimura[1] Takayuki Itamochi[1] Teppei Inomata[2]

Abstract— **In this paper, we propose a method to recognize semantic and geometric structure of a traffic scene using monocular cameras. We designed Deep Neural Networks (DNNs) for semantic segmentation and depth estimation and trained them using data collected with a test vehicle, on top of which a 360-degree panoramic camera system and a LIDAR are mounted. Collected images were manually annotated for semantic segmentation. Experimental results show that the trained DNNs can accurately classify each pixel and also accurately estimate depth of each pixel of images in validation data. Global average of semantic segmentation reached 96.4%, while overall accuracy of depth estimation was 88.7%. Generalization capability for both tasks was also tested with DNNs trained only with front facing camera images, resulting that semantic segmentation and depth estimation were successfully executed at slightly less accuracy. We also developed a novel interface using a head mount display, that enables us to evaluate results of estimation intuitively for checking how well the estimation of proposed DNNs is.**

I. INTRODUCTION

An ability of understanding traffic scenes is one of the essential features that autonomous driving systems and advanced driver assistant systems should have in order to replace or support driver's cognitive function by such systems. In traffic scenes, a driver is recognizing carefully and constantly what is around his/her vehicle and also where it is in the 3D space around the vehicle. In other words, 2D semantic information and 3D geometric information should be grasped simultaneously while driving.

A big challenge in developing such a sensing system for a serious commercial product is to build a reliable one at low cost. Camera is widely used in sensing systems of commercially successful products, as it is a non-expensive and promising device for 2D recognition. On the other hand, 3D recognition is commonly handled by stereo camera, millimeter-wave radar, and/or laser radar, that have advantage in precision in distance measurement, but disadvantage in cost. The trade off between sensing performance and its cost must be solved in order to make safety devices accessible for more people.

In this paper, we propose a method to recognize traffic scene in terms of both 2D semantic and 3D geometric structure with monocular camera(s) based on Deep Neural Networks (DNNs). The overview is briefly shown in Fig. 1. A single still image from a monocular camera is taken as an input, which is transferred independently to 2D process that infers semantic labels for all pixels and 3D process that infers depth for all pixels. The recognized 2D and 3D structures are then integrated into 3D semantic scene, that is displayed in a virtual reality (VR) interface to help us evaluating the recognition intuitively.

There are some related works that propose 3D semantic scene recognition. Schneider et al. [1] integrated semantic segmentation and stereo depth information using a stixel expression. Song et al. [2] proposed segmentation for 3D voxel, by taking image and depth map as input. In contrast, our method estimates both 2D and 3D information from a monocular camera image. We also tackle a challenge of comprehensive, or 360-degree recognition, that few papers have sufficiently addressed. We trained DNNs using panoramic camera data with annotation of 2D semantic information and 3D geometric information as ground truth. First question is whether a single DNN, that should be compact enough to be implemented in an in-car chip with limited resources, could have a sufficient ability to execute each task for all directions. This is crucial because computational cost can seriously increase if multiple DNNs are required to cover respective directions. Second question is whether a DNN trained with data of limited range of view, i.e. only front facing camera images, could have enough generalization capability for comprehensive recognition. This is even more crucial because placement of cameras may vary among vehicles and the cost of gathering annotated data for each placement is non-negligible.

Semantic segmentation, depth estimation, and those integration through the evaluation interface are described in section 2, 3 and 4, respectively. Section 5 includes conclusion and discussion.

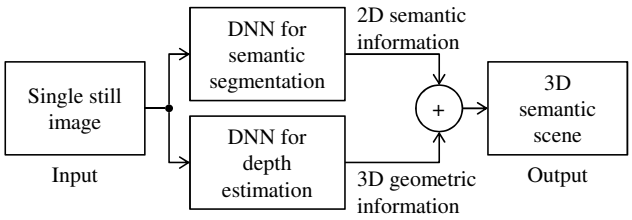


Fig. 1: Overview of DNNs for understanding 3D semantic scene. Both 2D semantic information and 3D geometric information are inferred by DNNs and integrated to get 3D semantic structure of environment.

[1] DENSO CORPORATION, 1-1 Showa-cho, Kariya, Aichi 448-8661, Japan kenichi_narioka@denso.co.jp
[2] Morpho, Inc., 3-8-1 Nishi-kanda, Chiyoda-ku, Tokyo, 101-0065, Japan

II. SEMANTIC SEGMENTATION

A. Task overview

Semantic segmentation is one of the most popular tasks in computer vision. In this task, each pixel of an input image should be classified into one of pre-defined classes, so as to divide the input image into semantic segments.

Comparing to conventional 2D recognition tasks such as object detection, semantic segmentation can deal with more detail information of 2D scene and handle classes with a variety of shapes, e.g. road, that is hard to *detect*. It is recently receiving even more attention since the progress of deep learning has been yielding remarkable improvements in this task [3], [4], [5].

B. DNN architecture

A convolutional neural network, particularly that with an encoder-decoder structure such as SegNet [6] is one of the promising architectures for semantic segmentation. Convolutional layers and pooling layers are iteratively processed in the encoding part to compress the information from input data, while upsampling layers and convolutional layers in the following decoding part make the compressed information high-resolution. Each pixel is then classified through a softmax layer.

We designed a DNN shown in Fig. 2, which has less number of layers and channels compared to SegNet. The proposed DNN aims at a good balance between performance and computational cost, so as to be executable with sufficient frequency in an in-car chip, that generally has limited computational resources.

C. Data for learning

Although open benchmark suites such as CamVid [7] and cityscapes [8] are often used for many semantic segmentation researches, those datasets are not suitable for a research of the comprehensive recognition since those include only front facing camera images.

In contrast, we collected data with our test vehicle that has a panoramic camera system on top of it, covering a 360-degree view around the vehicle. This camera system consists of five monocular cameras, each of which is arrayed equiangularly as shown in Fig. 3. We manually annotated each pixel of the collected images with its class label from a list that is tailored for traffic scenes, including pedestrian, vehicle, road and non-drivable area.

D. Experiment I: Training with all cameras

First, we divided a part of the annotated data for learning into training data and validation data, so that both include images from all cameras. A set of weight parameters of the DNN was trained with the training data in supervised learning manner. Note that we did not train independent DNNs for the respective cameras, because it may cause serious rise in computational cost. Instead, we trained a single DNN for all the cameras.

TABLE I: Result of Experiment II

Camera	Cam3	Cam4	Cam0	Cam1	Cam2
Global Accuracy[%]	87.2	76.7	90.0	88.3	89.9
Class Accuracy[%]	65.1	56.7	72.8	60.8	62.2
IoU[%]	44.0	37.2	53.2	41.0	41.2

Typical results of the inference for the validation data are shown in Fig. 5, where input images from all cameras, annotated label maps, and estimated label maps are displayed, respectively. It is found that estimated label maps are close to ground truth maps for all directions. It is notable that drivable areas and non-drivable areas are well segmented, suggesting that the context around each pixel is referred properly due to the encoder-decoder structure. In the quantitative evaluation, global accuracy, mean class accuracy, and mean IoU reached 96.4%, 88.7%, and 65.4%, respectively.

E. Experiment II: Training with front camera

Second, we divided other annotated data for learning into training data and validation data in a different way from that of the previous experiment, so that the training data includes only images from the front facing camera, called Cam0, while the validation data includes images of all the cameras. A set of weight parameters of the DNN was trained with the training data in supervised learning manner.

Qualitative results by the trained DNN are shown in Fig. 5, while quantitative result is shown in Table I, where global accuracy, mean class accuracy, and mean IoU for each camera are shown. It is found that label maps are estimated well not only for Cam0 which is used in the training, but also for the other cameras in the most of the validation scenes. It is, however, inevitable that the accuracy slightly decreases for side facing cameras, particularly Cam4, probably because there were not enough similar images in the training data. Accuracy for images from Cam4 dropped by more than 10% in all criteria. The typical error is shown in the second column of Fig. 5 (b), where a part of the sidewalk is misclassified as the road class.

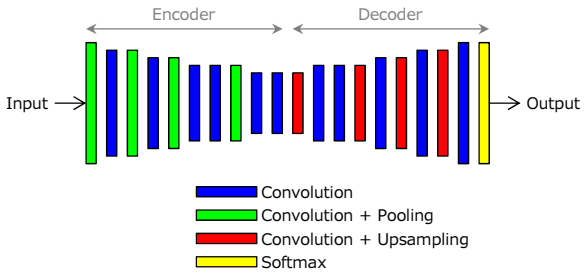


Fig. 2: Proposed DNN for semantic segmentation. The encoder part has iterative convolution al layers and pooling layers, while the decoder part has iterative convolutional layers and unpooling layers.

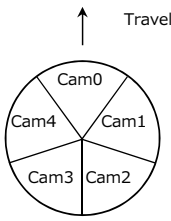


Fig. 3: Configuration of cameras. All cameras are placed equiangularly. Cam0 faces the front of the test vehicle.

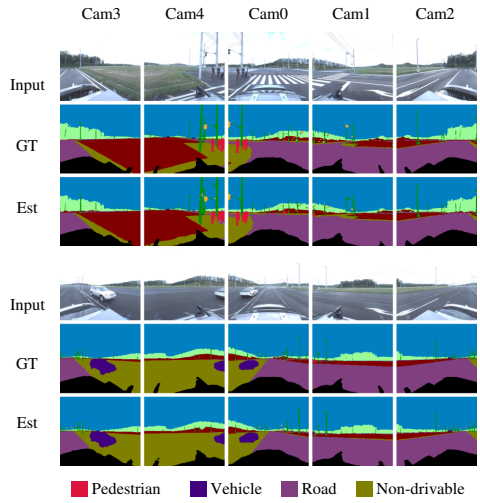


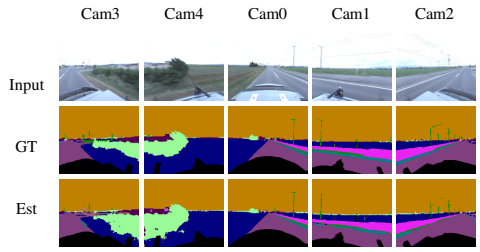
Fig. 4: Result of semantic segmentation in the experiment I. Input images, manually annotated label maps (ground truth), and estimated label maps are shown in the top, middle, and bottom, respectively in each figure. Each class is colorized with its pre-defined color, which is shown in the bottom.

III. DEPTH ESTIMATION

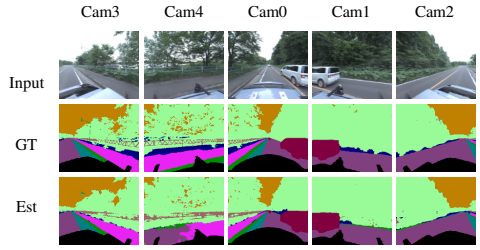
A. Task overview

Triangulation approaches are often used for depth estimation task, where depth of each pixel in an input image should be estimated to obtain a depth map. For example, a stereo camera system is often used to gain a disparity map, or a depth map, by matching feature points in paired images. Structure from motion (SfM) is a well known method to estimate depth with a monocular camera, also by the principle of triangulation. SfM obtains a depth map by matching feature points in neighbor frames and estimating its camera motion simultaneously.

On the other hand, it has been reported that a depth map of an input image can be well estimated directly by a DNN without using triangulation [9], [10], [11]. In this approach, a DNN can be trained with a number of images and corresponding depth maps in supervised learning manner. It has an advantage over a stereo camera in cost of device and cost of maintenance, as it requires only a monocular camera. It can estimate depth well even when the camera does not move and objects in the environment are moving, where SfM in principle does not work well in such situations.



(a)



(b)

Fig. 5: Result of semantic segmentation in the experiment II.

B. DNN architecture

Eigen et al. [9] proposed multi-scale CNN that can be applied for multiple tasks including semantic segmentation, normal estimation, and depth estimation. An output of a scale is taken by the following scale as an input, together with the original input image. They proposed a DNN composed of three scales, where the scale 1 outputs image feature, scale 2 does a depth map, and scale 3 make the depth map high-resolution.

As discussed in the previous section, it is important to reduce the amount of computation and the model size of the network, while maintaining the accuracy of the estimation as much as possible, since an in-car chip generally has limited computational resources. For the depth estimation task, we used multi-scale CNN architecture shown in Fig. 6, where the number of layers and number of maps in each layer were reduced comparing to [9], and also some convolutional layers were replaced by Firemodule [12], targeting at a good balance between accuracy and computational cost.

C. Data for learning

Most of open datasets available for training a depth estimator provide depth data of indoor environment [13]. Although there are some datasets providing depth data taken by in-car

sensors, such as KITTI [14] and cityscapes [8], those datasets are not suitable for a research of the comprehensive depth estimation since those include only front facing camera data.

In contrast, we collected data with our test vehicle that has the panoramic camera and an additional LIDAR system on top of the camera, that enable us to get camera images and synchronized depth data of the 360-degree view around the vehicle.

D. Experiment III: Training with all cameras

As we did in the experiment I in the previous section, we firstly divided the data for learning into training data and validation data, so that both include data from all cameras. A set of weight parameters of the DNN was trained with the training data in supervised learning manner.

A typical result is shown in Fig.7, where input images, depth maps measured by the LIDAR, depth maps estimated by the proposed DNN are shown in the top, middle, and bottom row of the figure, respectively. It is found that the estimated depth maps are qualitatively close to those of ground truth taken by the LIDAR for all directions. Note that the estimated depth for the area of the ego-vehicle is not precise at all because of the random guess, as this area has no ground truth depth. The quantitative result is shown in the upper row of Table II, where the accuracy of depth estimation is shown. Each score in the table means the ratio of the properly estimated pixels that satisfy the criteria $\delta = \max(\frac{d_i^*}{d_i}, \frac{d_i}{d_i^*}) < 1.25$, where d_i^* is depth of ground truth, and d_i is estimated depth for pixel i. The way of evaluation follows that of to the related paper [9]. As the accuracies of depth estimation keep reasonable level for all cameras, we see that the proposed DNN architecture has enough potential to cover all direction by a single one, instead of using independent DNNs for every directions.

E. Experiment IV: Training with front camera

As we did in the experiment II in the previous section, we divided the data for learning into training data and validation data so that the training data includes only images from Cam0, while the validation data includes images of all the cameras. A set of weight parameters of the DNN was trained with the training data in supervised learning manner.

A qualitative result by the trained DNN is shown in Fig. 8, while quantitative result is shown in the lower row of Table II. It is found that depth maps are estimated well not only for Cam0 used in the training, but also for the other cameras in the most of the validation scenes. In Fig. 8, depth maps shown in the third row, that are by the DNN trained with images from Cam0, are comparable to those shown in the fourth row, that are by the DNN trained with images from all the cameras. In quantitative evaluation, however, it is inevitable that the accuracy decreases for side facing cameras, particularly Cam4 dropped by 23.7%, probably because there was no similar image in the training data.

IV. EVALUATION INTERFACE

In the most of studies, outputs of semantic segmentation and depth estimation are normally represented in 2D space

TABLE II: Result of Experiment III and IV

Experiment	Trained with	Estimated with Cam3	Cam4	Cam0	Cam1	Cam2
III	Cam0-4	87.2	87.8	88.4	87.0	88.3
IV	Cam0	82.0	64.1	89.0	83.5	87.5

unit: [%]

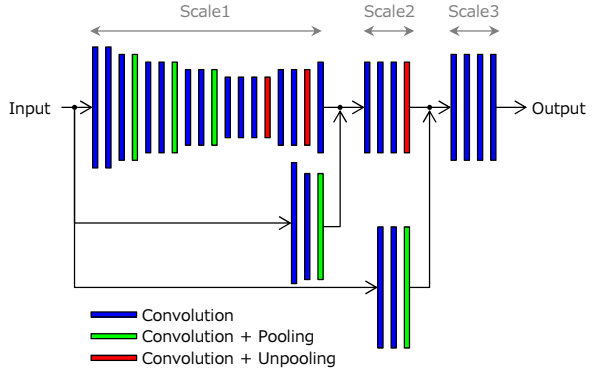


Fig. 6: Proposed DNN for monocular depth estimation. Scale1 extracts feature of an input image. Scale2 takes the output of the Scale1 together with the input filtered by several convolutional and pooling layers. Scale 3 has a function of upsampling to make the resolution of the output higher.

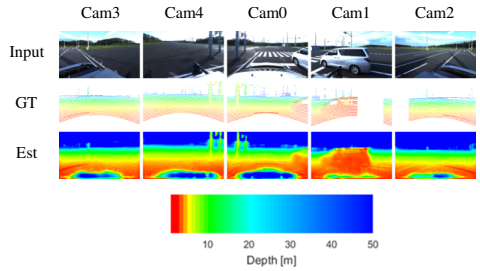


Fig. 7: Result of depth estimation in the experiment III. Input images, depth maps measured by LIDAR, depth maps estimated by the proposed network are shown in the top, middle, and bottom, respectively.

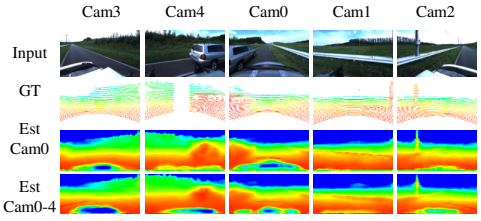


Fig. 8: Result of depth estimation in the experiment IV. From the top row, input images, depth maps measured by LIDAR, depth maps estimated by the DNN trained with images of Cam0, and depth maps estimated by the DNN trained with images of all the cameras are shown, respectively.

with a label map and a depth map. It is, however, sometimes hard to see the result of the estimation intuitively because of the gap of dimensions, as the world is indeed 3D space. For instance, it requires perceptive effort to imagine the 3D shape of an object precisely from a depth map, that may induce oversight of errors in the estimation. Although representation of 3D point clouds can be useful to grasp the 3D reconstructed world more easily, most of point cloud visualizers still require bothersome manipulation.

We developed a novel interface using a head mount display for virtual reality (VR) in order to make the evaluation of the estimators more intuitive. In this interface, 3D point clouds that are calculated using depth information earned by the depth estimation network and colorized according to the class labels earned by the semantic segmentation network, are displayed in the VR device, which provides a wearer with a sense of perspective with binocular parallax.

As an example, Fig. 9 shows a traffic scene and how it is fed into the VR interface. Input images from the all cameras are shown in Fig. 9 (a). In Fig. 9 (b), these images are put in the VR space. In Fig. 9 (c), each pixel is colorized using the result of semantic segmentation. In Fig. 9 (d), each pixel is ploted in 3D space using depth information. As shown in Fig. 9(e) and (f), the same object can be observed from the different points of view, according to the head position of a wearer of the device.

It contributes the qualitative evaluation of the estimation, such as object shape, boundary between objects. By integrating the 2D semantic information and the 3D geometric information from estimation networks through the interface, 3D semantic world is seamlessly reconstructed.

V. CONCLUSION

In this paper, we proposed a method of understanding 3D semantic scene with monocular cameras. We trained DNNs for semantic segmentation and depth estimation respectively with our comprehensive dataset that contains a number of 360-degree images, manually annotated label maps, and depth maps from LIDAR. Experimental results show that a single, compact DNN has a sufficient ability to execute each task for all directions. The other experiments were conducted to investigate the generalization capability, resulting that the trained DNN only with front facing camera images can successfully estimate 2D and 3D information at slightly less accuracy for images from the other cameras. This capability is important in practical use because the placement of cameras may vary widely among vehicles. We also developed a novel interface using a VR device, that enable us to evaluate the result of estimation intuitively.

It is crucial to improve the sensing function and performance by monocular cameras, not only for building a camera-based sensing system, but also for better fusion of multi sensors, because the semantic and geometric information from camera image provide an important clue to connect multiple modalities. We believe that our work can contribute to make in-car sensing systems more reliable and less expensive. Providing a high performance sensing at low

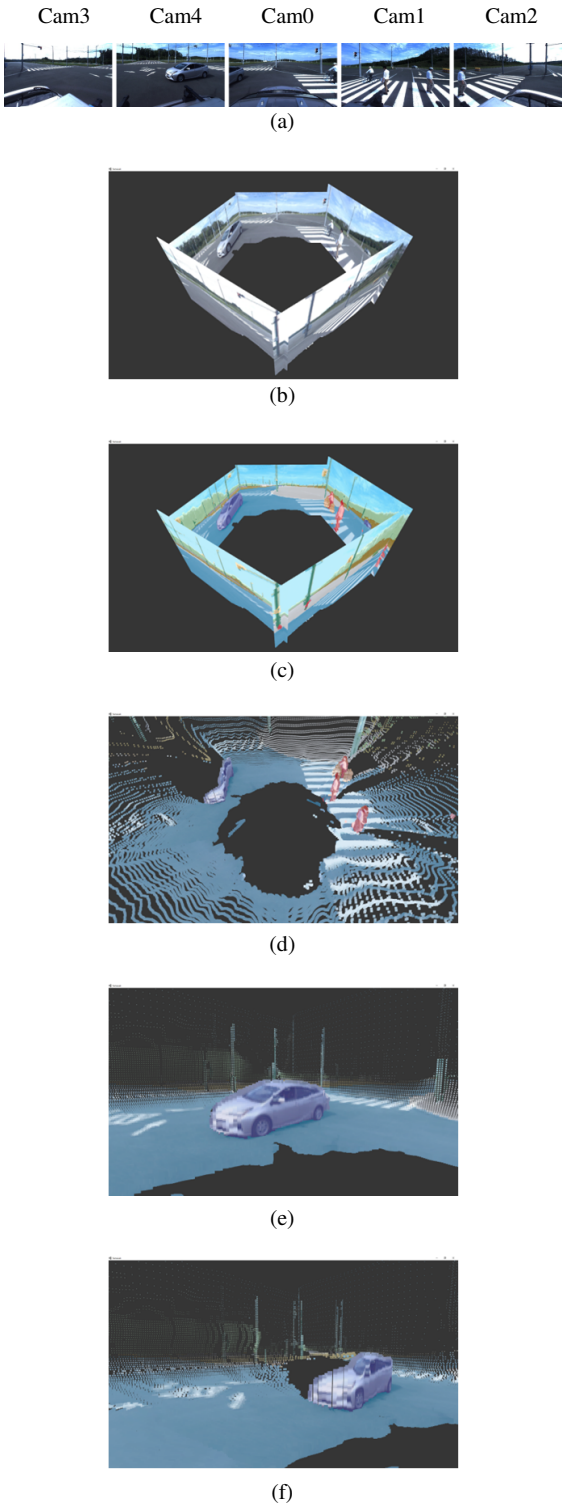


Fig. 9: 3D evaluation tool. (a) and (b) show input images from five cameras. (c) Result of semantic segmentation. (d) Result of semantic segmentation and depth estimation. (e) and (f) show the same vehicle from the different points of view.

cost will enlarge the safety system not only for expensive line but also standard line of vehicles, which is supporting the realization of safe car society.

REFERENCES

[1] L. Schneider, M. Cordts, T. Rehfeld, D. Pfeiffer, M. Enzweiler, U. Franke, M. Pollefeys, and S. Roth, "Semantic stixels: Depth is not enough," *IEEE Intelligent Vehicles Symposium (IV)*, 2016.

[2] S. Song, F. Yu, A. Zeng, A. X. Chang, M. Savva, and T. Funkhouser, "Semantic scene completion from a single depth image," *arXiv preprint arXiv:1611.08974*, 2016.

[3] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," *arXiv preprint arXiv:1411.4038*, 2014.

[4] J. Dai, K. He, and J. Sun, "Instance-aware semantic segmentation via multi-task network cascades," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 3150–3158.

[5] A. Paszke, A. Chaurasia, S. Kim, and E. Culurciello, "Enet: A deep neural network architecture for real-time semantic segmentation," *arXiv preprint arXiv:1606.02147*, 2016.

[6] V. Badrinarayanan, A. Kendall, and R. Cipolla, "Segnet: A deep convolutional encoder-decoder architecture for image segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2017.

[7] G. J. Brostow, J. Fauqueur, and R. Cipolla, "Semantic object classes in video: A high-definition ground truth database," *Pattern Recognition Letters*, vol. 30, no. 2, pp. 88–97, 2009.

[8] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele, "The cityscapes dataset for semantic urban scene understanding," in *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[9] D. Eigen and R. Fergus, "Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 2650–2658.

[10] 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*. IEEE, 2016, pp. 239–248.

[11] B. Ummenhofer, H. Zhou, J. Uhrig, N. Mayer, E. Ilg, A. Dosovitskiy, and T. Brox, "Demon: Depth and motion network for learning monocular stereo," *arXiv preprint arXiv:1612.02401*, 2016.

[12] F. N. Iandola, M. W. Moskewicz, K. Ashraf, S. Han, W. J. Dally, and K. Keutzer, "Squeezenet: Alexnet-level accuracy with 50x fewer parameters and¡ 1mb model size," *arXiv preprint arXiv:1602.07360*, 2016.

[13] N. Silberman, D. Hoiem, P. Kohli, and R. Fergus, "Indoor segmentation and support inference from rgbd images," in *European Conference on Computer Vision*. Springer, 2012, pp. 746–760.

[14] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, "Vision meets robotics: The kitti dataset," *The International Journal of Robotics Research*, p. 0278364913491297, 2013.