A robust monocular depth estimation framework based on light-weight ERF-PSPNet for day-night driving scenes

Keyang Zhou, Kaiwei Wang* and Kailun Yang

State Key Laboratory of Modern Optical Instrumentation, Zhejiang University

wangkaiwei@zju.edu.cn +86-571-8795-3154

Abstract. With the development of deep learning, various fields of computer vision have made huge progress. Among them, depth estimation is an important part of scene perception, therefore receives much interest and is widely used in daily life with the assistance of GPUs. Besides, the ways to obtain depth maps have also been improved, from using multiple images to a single image to obtain depth, which is called monocular depth estimation task. In this paper, we design a convolutional neural network called ERF-PSPNet to perform the task. We prove that by using unsupervised training, monocular depth estimation's result learned from large-scale dataset is close to the result of stereo matching. We also show that the monocular depth estimation model proposed in this paper can achieve a satisfying precision while maintaining a certain real-time frame rate for day-night driving scenes, which confirms the practical applicability of our design and result.

1. Introduction

Vision is one of the basic ways in which human rely on. Nowadays, the amount of visual information is increasing in a massive and explosive way. As an important and basic part of the vision information, depth plays an important role to transform two-dimensional scenes into three-dimensional information. Therefore, researching how to effectively obtain depth information will help us locate the obstacles and understand road conditions, which is essential for safety-desired autonomous vehicles [1].

The monocular depth estimation is proposed to overcome a series of shortcomings using stereo matching algorithm. In early days, most of monocular methods are restricted to simple scenes by adopting the theory of random field [2]. Nowadays, with the rapid development of autonomous vehicles and robots, higher precision of depth estimation is required with the complexity of scenes increasing. Besides, with the continuous development of equipment integration and miniaturization, the traditional multi-camera solution has gradually revealed its drawbacks, and the monocular depth estimation solution using deep learning, as an alternative, has gained more attention due to the advantages of simple equipments. Therefore, applying deep learning technology to monocular depth estimation has become a major research field now [3].

This paper focuses on training monocular depth estimation model which can be deployed in embedded processors with portable vision sensors [4]. The purpose is to use computer vision and deep learning methods in order to give high precision and robustness of monocular depth prediction results. At the same time, it should meet the requirements of real-time detection. Because we plan to use vision sensors mostly in car's perspectives, we constrain the area to the depth acquisition of outdoor scenes in this paper.

Our contribution can be summarized as follows: (1) An effective and efficient monocular depth estimation network is proposed. Specifically, we put forward a fully-convolutional network for the balance of inference accuracy and speed. (2) A comparison about supervised and unsupervised training strategies chosen for monocular depth estimation task. We prove that unsupervised training methods can reach higher ability of generalization despite its slightly lower accuracy on the dataset compared to supervised training methods. (3) A method of deploying model to a real-time device. We build a C++ program to make it deployable on real-time vision sensor, and test its performance in real-world environments across the day and night.

2. Related Work

Monocular depth estimation task has been proposed for some years. Recently, it receives much attention due to the development of deep learning.

2.1. Monocular estimation depth based on Markov random field

The method is to treat the pixel distribution of the whole image as a random field, and use the characteristics of the color, texture and other features to model different images based on the conditional random field (CRF) and Markov random field (MRF), proposed by Saxena et al [1]. In this way, the eigenvector is used to measure the pixel similarity. Then maximum likelihood estimation (MAP) is performed to give the depth prediction value.

2.2. Monocular depth estimation based on supervised deep learning methods

Eigen et al. [3] first introduced convolutional neural networks into monocular depth estimation. The authors divide convolutional neural network into Coarse Network and Fine Network. Coarse Network uses the classic AlexNet [5] structure while Fine Network applies multiple convolutional layers to optimize depth map obtained before. Jiao et al. [6] proposed a monocular depth estimation with the assistance of semantic booster and attention-driven loss. Fu et al. [7] discretized the continuous depth value into several bins and applied an additional classifier to improve the result.

2.3. Monocular depth estimation based on unsupervised deep learning methods

Garg et al. [8] first proposed a monocular depth estimation scheme based on unsupervised learning. This network training strategy only requires left and right views as training data. It mainly relies on the principle that the depth map has a one-to-one correspondence with the stereo disparity map. Godard et al. [9] further improved this idea, strengthening the constraints of the loss function during training, which introduces left-right RGB view reconstruction loss, image smoothness loss and left-right disparity map matching loss. Luo et al. [10] employed the network to generate a corresponding image view and calculate the depth information using binocular clues. Zhan et al. [11] used visual odometry information to assist the unsupervised training process. Aleotti et al. [12] added a generative adversarial network to measure the quality of generated image view. Guo et al. [13] applied domain adaption methods to enhance network's robustness about indoor and outdoor scenes.

However, previous works usually focus on enhancing the accuracy of prediction depth, which may lead to the decrease of inference speed. Moreover, those works usually focus on daytime scenes and perform relative badly at night. To solve those shortcomings, we proposed an ERF-PSPNet structure based on attention mechanism, and train it using unsupervised methods, which is described in the following section.

3. Methodology

Inspired by [14, 15], we design a new network named ERF-PSPNet, which uses light-weight ERFNet proposed by Romera et al. [16] as our feature extraction network and PSPNet proposed by Zhao et al. [17] as our feature fusion network. We also add attention mechanism and modify the loss function adapted to unsupervised training. The details of network design are presented below.

3.1. Feature extraction network: ERFNet

The feature extraction network is a neural network that down-samples the original input through a series of convolutional layers and pooling layers, extracting multi-level information of the image through expanding channel numbers of the convolutional layer.

As the depth of the network continues to increase for the need of higher precision, the running time and resources of the entire network are also increasing, which becomes a major difficulty in deploying it to embedded devices. Therefore, a feasible feature extraction network requires a balance between high quality and computational speed. We finally use ERFNet structure, which can significantly reduce the running time of the network while not greatly reducing the feature extraction ability. ERFNet's network design is shown in figure 1:

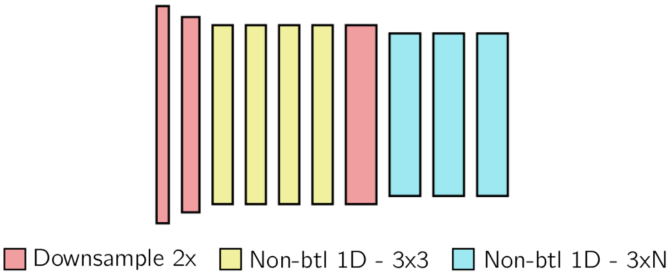

Figure 1. Structure of ERFNet

Specifically, ERFNet network can be divided into two parts:

a) Reduce the resolution of the original image by down-sampling. Although down-sampling operation loses some details, network's occupied resources can be cut down by reducing the resolution of the feature map. Here, the down-sampling process performs a convolutional layer and an average pooling layer with a step size of 2 in parallel. Afterwards, two branch's output are concatenated to compose the final result. We design this mechanism so as to better preserve the details of the original image.

b) After down-sampling, the network needs to learn the mapping relationship from RGB information to depth information. Therefore, the network uses the Non-Bottleneck connection layer. It replaces the original 3*3 convolution kernel with a pair of series-connected 1*3 and 3*1 convolution kernels. This method effectively reduces the parameter amount of the network while maintaining the extraction ability. At the same time, in order to make the gradient backward-propagation work smoothly, the Non-Bottleneck connection layer uses residual connection, which adds the original input to the final output feature map.

3.2. Multiscale Feature Fusion Network: PSPNet

The core of PSPNet is the merging of multiple features. After ERFNet extracts features, if the up-sampling layer is concatenated directly, the context information in the feature map cannot be used thoroughly and appropriately. Therefore, we apply PSPNet to operate on the extracted feature map in parallel. Here, 4 convolutional layers with different strides are applied on the same input. The convolution layers with small strides can learn detailed information, while ones with large strides can learn abstract information. The results obtained by these convolutions are then resized through bilinear up-sampling and stacked into a new feature extraction layer, which is sent to the subsequent up-sampling process. The detailed network design is shown in figure 2.

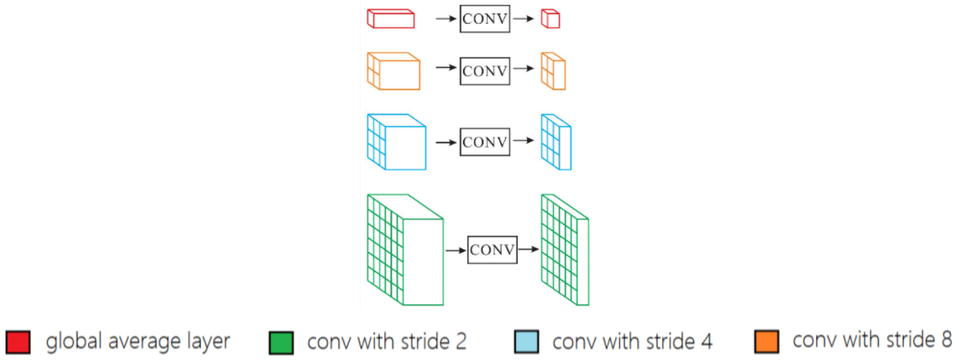

Figure 2. Structure of PSPNet

Specifically, the feature extraction layer is down-sampled by 1/2, 1/4, 1/8, respectively. At the same time, there is another branch to calculate the global feature vector by taking the average value of each channel. The global feature vector and the 1/2, 1/4, and 1/8 feature layers' outputs are learned by an additional convolution layer respectively and stacked to compose the final output. Here, in order to keep the total amount of information constant, the number of channels in each convolutional result layer is reduced, facilitating the neural network to integrate and compress information.

3.3. Application of Attention Mechanism in Fully Convolutional Networks

The fully convolutional network has developed rapidly since Long et al. [18] first proposed in 2015. Prior to this, convolutional neural network can only down-sample to reduce the image resolution. The fully convolution network applies the same number of up-sampling layers after down-sampling operations, and the output resolution is finally restored. This type of mechanism is often referred as the Encoder-Decoder mechanism [19].

Although the deconvolution operation enables image input of any size to be accepted by the network, the results are not accurate enough. In order to solve this problem, the attention mechanism is introduced. The attention mechanism is to generate layer and force network to pay more attention to the main information. Attention mechanism can also help neural networks better learn important features rather than fit some useless information. Attention mechanism was first widely used in Natural Language Processing and achieved good results. Recently, it is introduced into computer vision and improve the quality of prediction [20].

In practice, the attention mechanism adds a side branch to the backbone neural network, which is generally composed of several convolutional layers and the final sigmoid layer, which normalizes the result to a range of 0-1. The output is finally multiplied by the backbone result to obtain the final result. The specific flow char indicating how to apply the attention mechanism is shown in figure 3. We finally combined the attention mechanism into ERF-PSPNet's forward propagation process.

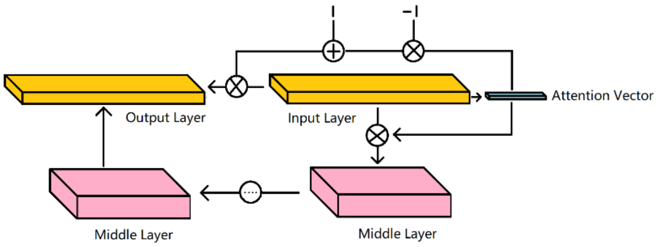

Figure 3. The mechanism of attention layer used in ERF-PSPNet's forward propagation

3.4. Monocular depth estimation loss function

The traditional loss function of monocular depth estimation is adapted to the supervised network, where true depth information is included in training dataset. Then predicted results and real data are directly compared by the loss function. L1 loss function and an L2 loss function are usually used, where L1 function is an absolute value function and L2 function is a square function.

However, in the actual training process, it is found that the while supervised training strategy can achieve better results on the training dataset, it does not have strong robustness and generalization ability. Considering the traditional matching algorithm, the disparity information based on stereo or multi-view information is mostly used to obtain final depth information through a simple geometric relationship between depth information and disparity information, as formula 1 illustrates.

$$depth = \frac{focal\_length * camera\_distance}{disparity} \tag{1}$$

Inspired by [9] and formula 1, the unsupervised training's loss function is proposed as the sum of the following three parts:

a) Left-right view reconstruction loss. The loss function is designed to measure the deviation of true views and generated views. Specifically, each pixel in the left view is translated to the corresponding left-view difference, and finally a new right view is obtained. The more accurate the left disparity map, the smaller the difference between the new right view and the true right view, vice versa.

$$\mathrm{C}_{ap}^{l} = \frac{1}{N}\sum_{i,j}\left\|I_{ij}^{l} - \tilde{I}_{ij}^{l}\right\| \tag{2}$$

b) Left-right disparity consistency loss. The loss function is designed to measure the deviation of left-right disparity map's consistency since the disparity maps are also separated to left disparity maps and right disparity maps. Therefore, left and right disparity maps obey the same relationship in part(a).

$$\mathrm{C}_{lr}^{l} = \frac{1}{N}\sum_{i,j}\left\|d_{ij}^{l} - d_{ij+d_{ij}^{l}}^{l}\right\| \tag{3}$$

c) Disparity gradient loss. The loss function is designed to ensure most areas should be smooth. The disparity map is usually more precise when the sum of gradients is less.

$$\mathrm{C}_{\mathrm{ds}}^{l} = \frac{1}{N}\sum_{i,j}\left|\partial_x d_{ij}^{l}\right| e^{-\left\|\partial_x I_{ij}^{l}\right\|} + \left|\partial_y d_{ij}^{l}\right| e^{-\left\|\partial_y I_{ij}^{l}\right\|} \tag{4}$$

Finally, the total loss function is the weighted sum of the above:

$$Loss = C_{ap}^{l} + C_{ap}^{r} + \alpha_{lr} * (C_{lr}^{l} + C_{lr}^{r}) + \alpha_{ds} * (C_{ds}^{l} + C_{ds}^{r}) \tag{5}$$

4. Experiment

We mainly use Cityscapes [21] and KITTI [22] dataset to train networks. There are 2975 training samples, 1525 validation samples, and 500 test samples in the Cityscapes dataset. We use Cityscapes dataset to verify the convergence ability of the network structure in early stage. KITTI dataset is a dataset with a very wide range of scenes, having approximately 40,000 images available for training. We use KITTI dataset as the training dataset to train model, which also serves as the benchmark for the evaluation of the final training results.

We implement ERF-PSPNet using TensorFlow [23] and train the whole network on a NVIDIA 1080Ti.

We firstly trained the model using supervised method on Cityscapes. Despite the competitive results on Cityscapes dataset, its performance is very low when we test the trained model on real-world scenes. Therefore, we changed our training strategy to unsupervised method and use KITTI dataset as our unsupervised training dataset to obtain a better result afterwards.

4.1. Supervised training method based on ERF-PSPNet

We use ERF-PSPNet with attention mechanism as prediction network. To test the preliminary effects of the network, we use Cityscapes dataset here. In the training process, the initial learning rate is adjusted to 1e-3 and multiplied by 0.8 after every 5 iterations. We use Adam optimizer [24]. During the training process, it was found that the network's monocular estimation results for depth were comparably precise. Some examples are shown in figure 4.

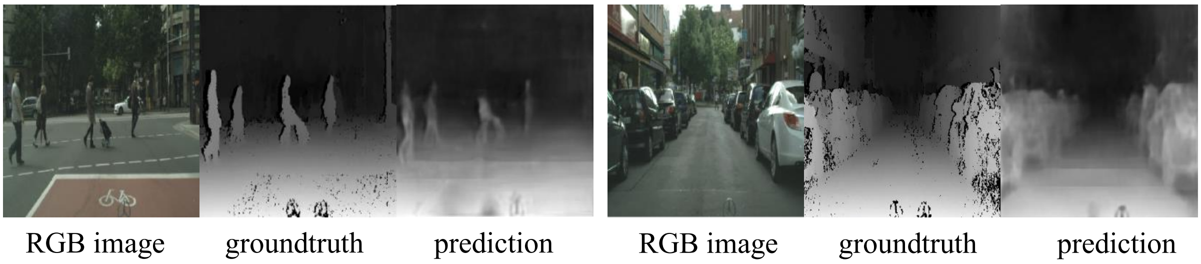

Figure 4. Supervised training results on Cityscapes dataset

From left to right is real RGB map, real depth map, and predicted depth map, respectively. In the upper row of figure 4, the depth information of pedestrian can be correctly predicted. In the lower row of figure 4, the depth information of the roadside cars remains a clear shape. Besides, depth information of the adjacent vehicles can be distinguished, too. Therefore, it shows a precise result achieved on the training dataset.

However, when the trained model is transplanted to the real-world video frames, the prediction is unsatisfactory, which showed that the model is still not good at depth prediction of the real-world scene. Therefore, we propose the following two methods for improvement: one is to use a larger dataset such as the KITTI dataset described above; the other is to use unsupervised training strategy as unsupervised loss function can better describe the characteristics of depth information than the supervised loss function.

4.2. Unsupervised training method based on ERF-PSPNet

In the unsupervised training setting, we use the same ERF-PSPNet structure as supervised training except the loss function. In addition, training dataset can be expanded to more than 40,000 image pairs for the monocular depth estimation task. Since the learning method is changed to unsupervised training, hyperparameters should be adjusted too. We set the initial learning rate to 1e-5, and then after every 3 iterations, the learning rate was decayed by 0.2. We still use Adam optimizer, and the weights in formula 5 is all set to 1.

We already acknowledge that the network has strong learning ability from previous chapter, so after switching to unsupervised training strategy, the model can still converge. The loss function that can directly explain the quality of the generated disparity map is the left-right view reconstruction loss. We observe that the final left-right view reconstruction loss of the model is approximately stable at around 0.12. Besides, inference speed on a single NVIDIA 1080Ti can reach around 50fps when the resolution of the input image is 512*256. After the training is over, we obtain the predicted results shown in figure 5:

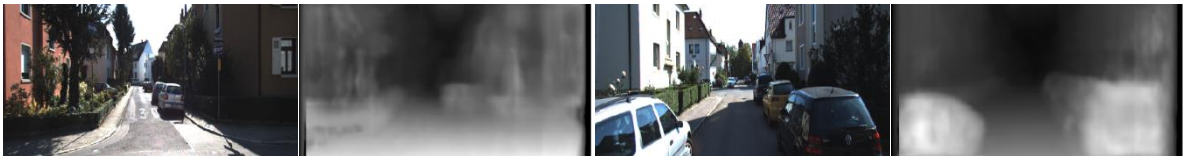

Figure 5. Unsupervised training results on KITTI dataset

From left to right is real RGB map and predicted depth map, respectively. We notice that the prediction results on the training dataset are not as good as the results of the supervised learning; but from the results of the previous section, obtaining particularly good prediction results on the training dataset sometimes contradicts to model's generalization ability. When the model is applied to the real-world video, the effect is better than the model using supervised training, which shows that the unsupervised learning can assist network to learn the features required for the depth information. Finally, we choose the unsupervised training model as our result due to its better generalization capacity and robustness in the real world, which is a key point in autonomous vehicle systems.

4.3. Evaluation metrics

The trained model is evaluated using standard metrics [22]. Results are shown in Table 1.

Table 1. Evaluation results on KITTI dataset

Methods	Abs Rel	RMSE	RMSE $_{\log}$	$\delta<1.25$	$\delta<1.25^2$	$\delta<1.25^3$
Eigen	0.203	6.307	0.282	67.3%	89.0%	95.8%
Liu	0.201	6.471	0.273	68.0%	89.8%	96.7%
Garg	0.169	5.104	0.273	74.0%	90.4%	96.2%
Godard	0.140	4.471	0.232	81.8%	93.1%	96.9%
Ours	**0.200**	**7.633**	**0.315**	**62.6%**	**85.5%**	**94.6%**

It can be seen that the results are in a medium ranking position compared to other works. Since we need to balance real-time and accuracy in the application to the real-world scenario, this model does not use a more powerful network such as ResNet [25] or DenseNet [26]. At the same time, because the training resources of this model are only one single NVIDIA 1080Ti, it is difficult to conduct large-scale training. Therefore, the result is still of practical value.

4.4. Real-world result

In the real-world scene test, we use our multimodal vision sensor to collect the images across the day and night, where the dataset has been made publicly available at [27]. As is shown below, at daytime, model can give satisfactory prediction results of vehicles, pedestrians and other objects. Besides, the process of vehicle's movement can be clearly observed.

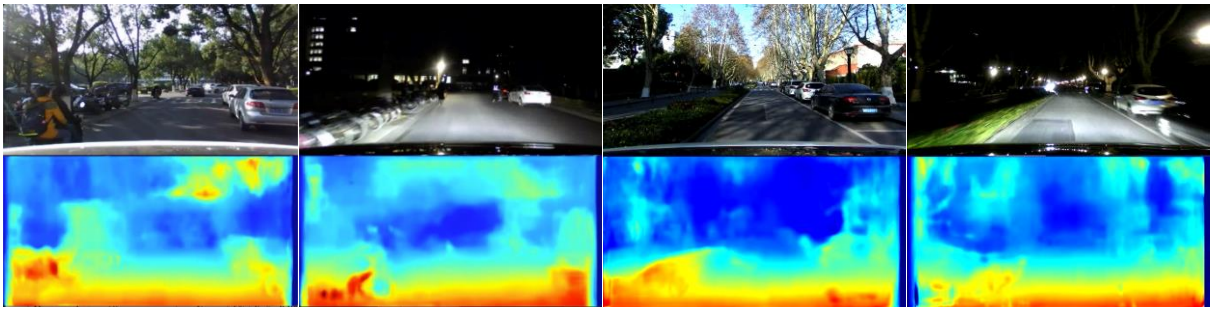

Figure 6. Results of real-world scenes depth prediction across the day and night

In addition, an applicable monocular depth estimation model should also be adaptive to different environments. We also collect the video data of the same scene at night and observe the predicted depth. The results show that the model can still reflect the approximate distribution of the depth, which indicates that the effectiveness of the proposed framework, yielding robust and efficient ERF-PSPNet model that is able to work both swiftly and reliably even at nighttime.

5. Conclusion and future work

We propose a light-weight ERF-PSPNet to achieve the target of monocular depth estimation task. According to the experiments, we finally choose the unsupervised training strategy for the pursue of a better robustness and generalization capacity. We test the trained model on real-world scenes and confirm its ability of monocular depth prediction across the day and night. Future work will focus on combining the algorithm of monocular and binocular depth estimation and further accelerate the inference speed. In addition, we are determined to research on panoramic monocular depth estimation jointly with panoramic semantic segmentation [28] in order to provide a high-level comprehensive perception for vision sensor in complex intersections and dynamic scenes.

6. References

[1] Yang, K., Wang, K., Hu, W. and Bai, J., *Expanding the detection of traversable area with RealSense for the visually impaired*. Sensors, 16(11), 2016, p.1954.

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

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

[4] Sun, D., Huang, X. and Yang, K., *A multimodal vision sensor for autonomous driving*. arXiv preprint arXiv:1908.05649, 2019.

[5] Krizhevsky, A., Sutskever, I. and Hinton, G.E., *Imagenet classification with deep convolutional neural networks*. In Advances in neural information processing systems, 2012, pp. 1097-1105.

[6] Jiao, J., Cao, Y., Song, Y. and Lau, R., *Look deeper into depth: Monocular depth estimation with semantic booster and attention-driven loss*. In Proceedings of the European Conference on Computer Vision (ECCV), 2018, pp. 53-69.

[7] Fu, H., Gong, M., Wang, C., Batmanghelich, K. and Tao, D., *Deep ordinal regression network for monocular depth estimation*. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 2002-2011.

[8] Garg, R., BG, V.K., Carneiro, G. and Reid, I., October. *Unsupervised cnn for single view depth estimation: Geometry to the rescue*. In European Conference on Computer Vision, 2016, pp. 740-756.

[9] Godard, C., Mac Aodha, O. and Brostow, G.J., *Unsupervised monocular depth estimation with left-right consistency*. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 270-279.

[10] Luo, Y., Ren, J., Lin, M., Pang, J., Sun, W., Li, H. and Lin, L., *Single view stereo matching*. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 155-163.

[11] Zhan, H., Garg, R., Saroj Weerasekera, C., Li, K., Agarwal, H. and Reid, I., *Unsupervised learning of monocular depth estimation and visual odometry with deep feature reconstruction*. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 340-349.

[12] Aleotti, F., Tosi, F., Poggi, M. and Mattoccia, S., *Generative adversarial networks for unsupervised monocular depth prediction*. In Proceedings of the European Conference on Computer Vision (ECCV), 2018, pp. 0-0.

[13] Guo, X., Li, H., Yi, S., Ren, J. and Wang, X., *Learning monocular depth by distilling*

cross-domain stereo networks. In Proceedings of the European Conference on Computer Vision (ECCV), 2018, pp. 484-500.

[14] Yang, K., Wang, K., Bergasa, L., Romera, E., Hu, W., Sun, D., Sun, J., Cheng, R., Chen, T. and López, E., *Unifying terrain awareness for the visually impaired through real-time semantic segmentation*. Sensors, 18(5), 2018, p.1506.

[15] Yang, K., Hu, X., Bergasa, L.M., Romera, E., Huang, X., Sun, D. and Wang, K., *Can we pass beyond the field of view? panoramic annular semantic segmentation for real-world surrounding perception*. In 2019 IEEE Intelligent Vehicles Symposium (IV). IEEE. 2019, pp. 446-453.

[16] Romera, E., Alvarez, J.M., Bergasa, L.M. and Arroyo, R., *Erfnet: Efficient residual factorized convnet for real-time semantic segmentation*. IEEE Transactions on Intelligent Transportation Systems, 19(1), 2017, pp.263-272.

[17] Zhao, H., Shi, J., Qi, X., Wang, X. and Jia, J., *Pyramid scene parsing network*. In Proceedings of the IEEE conference on computer vision and pattern recognition, 2017, pp. 2881-2890.

[18] Long, J., Shelhamer, E. and Darrell, T., *Fully convolutional networks for semantic segmentation*. In Proceedings of the IEEE conference on computer vision and pattern recognition, 2015, pp. 3431-3440.

[19] Badrinarayanan, V., Kendall, A. and Cipolla, R., *Segnet: A deep convolutional encoder-decoder architecture for image segmentation*. IEEE transactions on pattern analysis and machine intelligence, 39(12), 2017, pp.2481-2495.

[20] Hu, J., Shen, L. and Sun, G., *Squeeze-and-excitation networks*. In Proceedings of the IEEE conference on computer vision and pattern recognition, 2018, pp. 7132-7141.

[21] Cordts, M., Omran, M., Ramos, S., Rehfeld, T., Enzweiler, M., Benenson, R., Franke, U., Roth, S. and Schiele, B., *The cityscapes dataset for semantic urban scene understanding*. In Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 3213-3223.

[22] Geiger, A., Lenz, P. and Urtasun, R., *Are we ready for autonomous driving? the kitti vision benchmark suite*. In 2012 IEEE Conference on Computer Vision and Pattern Recognition, IEEE. 2012, pp. 3354-3361.

[23] TensorFlow Framework: *https://www.tensorflow.org/.*

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

[25] He, K., Zhang, X., Ren, S. and Sun, J., *Deep residual learning for image recognition*. In Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 770-778.

[26] Iandola, F., Moskewicz, M., Karayev, S., Girshick, R., Darrell, T. and Keutzer, K., *Densenet: Implementing efficient convnet descriptor pyramids*. arXiv preprint arXiv:1404.1869, 2014.

[27] ZJU Dataset: *https://github.com/elnino9ykl/ZJU-Dataset.*

[28] Yang, K., Hu, X., Bergasa, L.M., Romera, E. and Wang, K., *PASS: Panoramic Annular Semantic Segmentation*. IEEE Transactions on Intelligent Transportation Systems, 2019.