2017 IEEE Intelligent Vehicles Symposium (IV)
June 11-14, 2017, Redondo Beach, CA, USA

CNN based Semantic Segmentation for Urban Traffic Scenes using Fisheye Camera

Liuyuan Deng, Ming Yang, Yeqiang Qian, Chunxiang Wang, and Bing Wang

***Abstract*—Semantic segmentation is an important step of visual scene understanding for autonomous driving. Recently, Convolutional Neural Network (CNN) based methods have successfully applied in semantic segmentation using narrow-angle or even wide-angle pinhole camera. However, in urban traffic environments, autonomous vehicles need wider field of view to perceive surrounding things and stuff, especially at intersections. This paper describes a CNN-based semantic segmentation solution using fisheye camera which covers a large field of view. To handle the complex scene in the fisheye image, Overlapping Pyramid Pooling (OPP) module is proposed to explore local, global and pyramid local region context information. Based on the OPP module, a network structure called OPP-net is proposed for semantic segmentation. The net is trained and evaluated on a fisheye image dataset for semantic segmentation which is generated from an existing dataset of urban traffic scenes. In addition, zoom augmentation, a novel data augmentation policy specially designed for fisheye image, is proposed to improve the net's generalization performance. Experiments demonstrate the outstanding performance of the OPP-net for urban traffic scenes and the effectiveness of the zoom augmentation.**

I. Introduction

It is an essential task for autonomous vehicles to perceive the surrounding environments. Semantic segmentation takes an important step towards traffic scene understanding by parsing an image into different regions with specific semantic categories, such as road, vehicles, pedestrians, etc. It has a variety of applications for autonomous driving, such as scene representation [1, 2], road detection [3], semantic mapping [4], etc. The research [1] proposed semantic stixels, a scene model geared towards the needs of autonomous driving applications, which is produced by leveraging pixel-wise semantic segmentation results and dense depth maps. The research [5] attempted to simplify and unify detection tasks in the perception module of an autonomous vehicle by using pixel-wise semantic segmentation.

Early semantic segmentation methods relied upon handcrafted features, using Random Decision Forest [6] or Boosting [7] to predict the class probabilities. And using probabilistic models known as Conditional Random Fields (CRFs) to handle uncertainties and propagate contextual information across the image. Recent years, Convolutional Neural Networks (CNNs) have made a huge step forward in vision recognition due to large-scale training datasets [8-10] and high performance Graphics Processing Unit (GPU). In addition, excellent open source deep learning frameworks [11-13] speed up algorithm development. Powerful deep neural networks [14-17] largely reduced the classification errors on ImageNet [8], which also benefit semantic segmentation. The state-of-the-art semantic segmentation methods all relied on deep convolutional neural networks, which advanced by a large gap to early methods. FCN [18], which allows arbitrary-sized input, adapted classification networks for semantic segmentation by replacing fully-connected layers with convolutional layers. Following this line, subsequent works [19-21] have extended FCN to achieve a state-of-the-art performance.

In urban environments, the traffic situations can be very complex with unpredictable behaviors of dynamic traffic participants and more challenging compared to highways and rural roads, especially at intersections. A beneficial solution in urban environments is to perceive wide-angle views for semantic segmentation. However, the traditional methods are usually performed using pinhole camera whose field of view is limited. This paper thus focuses on semantic perception of urban traffic scenes using fisheye camera which theoretically provides the entire frontal hemispheric view of 180°.

Fisheye camera is widely applied in areas of intelligent vehicles for its coverage of large field of view, such as parking [22], vehicle surrounding monitoring [23], object detection [24], scene understanding [25], etc. Due to the strong distortions, the fisheye image is usually undistorted in practical usage. However, un-warping hurts the image quality and impacts the detection process especially at the image boundaries. Learning based methods trained on conventional images is difficult to be applied to the undistorted images. Therefore, some studies adapted its image processing operations to handle the uncorrected image directly [24]. This paper will also perform semantic segmentation directly on uncorrected fisheye images.

As we know, few studies attempted to perform semantic segmentation directly on fisheye images using deep learning method. To address the issue, this paper considers two prominent challenges in the application of CNNs to fisheye image semantic segmentation: (1) huge variations of objects in the fisheye image, (2) lack of large-scale dataset. The first challenge is due to the inherent distortion of fisheye cameras, which is inevitable in the projection from a hemispheric field onto a plane [26]. Things and stuff in the image show large variations of size with strong distortions. The second challenge is due to the requirement of supervised learning. So far, state-of-the-art CNN-based semantic segmentation

This work was supported by the National Natural Science Foundation of China (91420101) and International Chair on automated driving of ground vehicle.

Ming Yang is the corresponding author.

L. Deng, M. Yang, Y. Qian and B. Wang are with the Department of Automation, Shanghai Jiao Tong University, Key Laboratory of System Control and Information Processing, Ministry of Education of China, Shanghai, 200240, China (phone: +86-21-34204533; email: MingYang@sjtu.edu.cn).

C. Wang is with Research Institute of Robotics, Shanghai Jiao Tong University, Shanghai, 200240, China.

models require large-scale pixel-level annotated images to optimize parameters and avoid overfitting. However, the generation of the dataset requires expensive and time-consuming annotation effort, especially for fisheye image.

This paper presents a CNN-based semantic segmentation solution for urban traffic scenes using fisheye camera. To handle the complex scene in the fisheye image, local, global and pyramid local region features are integrated by an overlapping pyramid pooling (OPP) module which is detailed in Sec. II-A. A network structure for semantic segmentation based on the OPP module, called OPP-net, is proposed in Sec. II-B. To compensate for the lack of large-scale annotated dataset, a fisheye image dataset for semantic segmentation is generated from an existing dataset of urban scenes in Sec. II-C. Besides, in Sec. II-D, a data augmentation policy specially designed for fisheye image is proposed to improve the generalization performance. Experiments are carried out in sec. III, where the OPP-net was trained and evaluated on the fisheye image dataset.

II. METHOD

A. Overlapping Pyramid Pooling Module

Contextual information can helps clarify local confusions, as discussed in [27]. Parsenet [27] aggregated local features with global pooling to augment the features at each location and achieved success in VOC2012 [9] and PASCAL-Context [28]. But it is not representative enough for more challenging scene. Inspired by spatial pyramid pooling which is successful used in object detection [29], PSPNet [21] expanded the capability of global context information by fusing information from different sub-regions. The sub-region pyramid pooling module as illustrated in Fig. 1 has four different pyramid scales: one global pooling layer and three finer non-overlapping pooling layers with bin sizes of 2x2, 3x3 and 6x6 respectively. For non-overlapping pooling, the kernel size and the stride is identical. Thus the spatial size of feature map is divided by the kernel size after using the non-overlapping pooling operation. This leads to a constraint for the module that the size of input feature map should be exactly divisible by all the kernel sizes of the non-overlapping pooling layers.

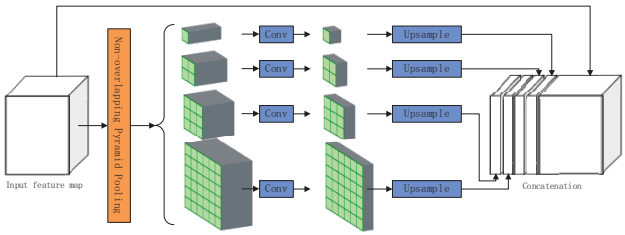
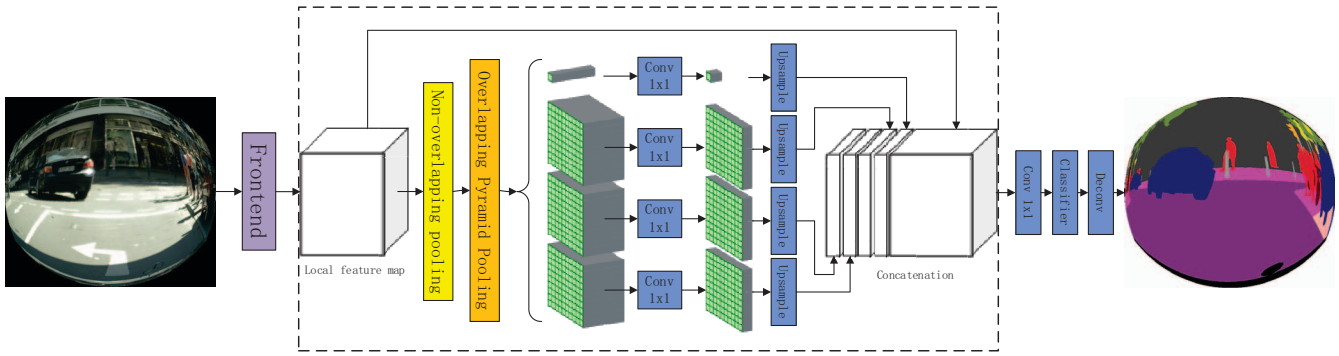


Figure 1. The sub-region pyramid pooling module (called pyramid pooling module in PSPNet [21]). Here the non-overlapping pyramid pooling consists of four different pyramid scales.

Otherwise, after pooling and upsampling, the module will encounter alignment problem. For example, if the kernel sizes are 45, 30 and 15, the size of input feature map should be a multiple of 90.

Different from the sub-region pyramid pooling module which is done over non-overlapping regions of feature map, the proposed OPP module illustrated in Fig. 2(c) employs overlapping pyramid pooling. The overlapping pyramid pooling contains four levels: one global pooling layer and three overlapping pooling layers with increasingly finer kernel sizes. The size of levels and kernel sizes can be changed as hyperparameters. The global pooling layer generates a single bin output and produces global features. Local region feature maps with different region sizes are produced by the overlapping pooling layers. The stride of each overlapping pooling layer is fixed to 1 and padding of corresponding size is set to preserve the spatial size of the input feature map. Therefore, the overlapping pyramid pooling does not alter the spatial size of its input feature map. Note that the padding will not involve computing actually, either in max or average pooling. The overlapping pooling with stride 1 can preserve the original spatial dimension, but it will lead to larger amount of computation compared to non-overlapping pooling. To make the computing efficient, a non-overlapping pooling operation with a small kernel size before the overlapping pyramid pooling is employed to downsample the feature map. The type of pooling operation can be max or average. The 1×1 convolution layers appended to each pyramid level are used to



(a) Input image (b) Frontend (c) Overlapping Pyramid Pooling (OPP) Module (d) Final Prediction

Figure 2. The framework of OPP-net. The frontend (b) generates local feature map from input image (a), and then the overlapping pooling pyramid module (c) gathers local, global and pyramid region context information. The non-overlapping pooling in the OPP module (c) is used to downsample the feautre map. Global and pyramid region feature maps are generated by the overlapping pyramid pooing. The concatenated feature map is reduced by a 1x1 convolutional layer with channel dimension 512 and then classified by a 3x3 convolutional layer. Finally, final prediction (d) is obatained by a deconvolutional layer.

reduce the dimension of the context representation to 1/*N* of the original feature map. N equal to 4 denotes the level size of pyramid. All the low-dimension pyramid feature maps are upsampled to the size of the original feature map via bilinear interpolation. Finally, the local, global and pyramid local region features are concatenated.

Note that the OPP module has several remarkable properties. Firstly, there is no constraint for the size of input feature map due to the overlapping pooling with a stride of 1. Thus the OPP module is well fitting for the fully convolutional networks. Besides, the module still keeps good translation invariant property. This property is harmed in the sub-region pyramid pooling module [21] because of the large kernel size of non-overlapping pooling in the module, to the maximum of half of the size of input feature map. The spatial size is largely reduced by the non-overlapping pooling, but the upsampling operation can not fully restore local region features. However, for OPP module, the overlapping pooling can help accurately obtain local region features for each location if neglecting the downsampling of the non-overlapping pooling with small kernel size. Thirdly, the module keeps efficient due to the existence of the non-overlapping pooling operation before the overlapping pyramid pooling. And the non-overlapping pooling with small kernel size won't lead to the alignment problem, just like common CNNs. Finally, local features from the input feature map, global features and pyramid local region features from the overlapping pyramid pooling are all integrated to one feature map.

B. Network Architecture

The whole framework of proposed OPP-net is illustrated in Fig. 2. The complete architecture mainly consists of the frontend and the OPP module. The frontend provides the input image to the network and extracts feature map of local features. And the OPP module is used to gather the local, global and pyramid local region context information. The frontend is a dilated fully convolutional network. It is built up following fully convolutional residual network (FCRN) [19] which is mainly based on the ResNet [17] and FCN [18]. First, an 18-layer ResNet is turned to 26-layer ResNet by replacing each 2-layer block in the 18-layer net with a 3-layer bottleneck block. Then the last 7×7 pooling layer and the classifier are removed. The last two downsampling operations are cancelled by setting the stride of the convolutional layer to 1 and then increase the dilations of subsequent convolution kernels by a factor of 2 for each downsampling operation. Thus a higher-resolution feature map is produced with a resolution of 1/8. Then the OPP module is used to integrate the local, global and local region features to a single feature map. Eventually, after the 1×1 convolutional layer and the classifier, the feature map is upsampled to get the final prediction with the original resolution by a factor of 8 using a deconvolutional layer adopted by [18]. Batch normalization [30] is used after each convolutional layer except the classifier.

The OPP-net is a kind of fully convolutional networks that take input of arbitrary size. It has a remarkable representing ability by integrating local, global and pyramid local region features via OPP module. And the net can be trained end-to-end using stochastic gradient descent (SGD). The OPP-net is adopted to perform semantic segmentation for urban traffic scenes using fisheye camera.

C. The Generation of Fisheye Image Dataset for Semantic Segmentation

The fisheye image dataset is generated from an existing conventional image dataset. A mapping is built from the fisheye image plane to the conventional image plane. Thus the scene in conventional image can be remapped into fisheye image.

The conventional image is captured from a pinhole camera. The perspective projection of a pinhole camera model can be described by (1); for fisheye cameras, perhaps the most common model is the equidistance projection [31], as in (2):

$$r = f \tan\theta, \tag{1}$$

$$r = f\theta, \tag{2}$$

Where θ is the angle between the principal axis and the incoming ray, r is the distance between the image point and the principal point, and f is the focal length. Both the conventional image and the fisheye image are considered as a result of projection of a hemisphere on a plane with different projection models and view angles. The details of the geometrical imaging model are described by [26, 31]. With the settings that the focal lengths of the perspective projection and the equidistance projection are identical and the max viewing angle $\theta_{\max}$ is equal to 180°. The mapping from the fisheye image point $P_f = (x_f, y_f)$ to the conventional image point $P_c = (x_c, y_c)$ is described by (3):

$$r_c = f \tan(r_f / f), \tag{3}$$

where $r_c = \text{sqrt}((x_c - u_{cx})^2 + (y_c - u_{cy})^2)$ denotes the distance between the image point P_c and the principal point $U_c = (u_{cx}, u_{cy})$ in the conventional image, and $r_f = \text{sqrt}((x_f - u_{fx})^2 + (y_f - u_{fy})^2)$ correspondingly denotes the distance between the image point P_f and the principal point $U_f = (u_{fx}, u_{fy})$ in the fisheye image.

The mapping relationship (3) is determined by focal length f. A base focal length f_0 is set, thus the fisheye camera model approximately covers a hemispherical field. Each image and its corresponding annotation in the existing segmentation dataset are transformed using the same mapping function to generate the fisheye image dataset, except different interpolation methods: bilinear interpolation for images and nearest-neighbor interpolation for annotations.

D. Zoom Augmentation for Fisheye Image

Deep networks require huge number of training images during training to get better performance. But the training dataset is always limited. Data argumentation is adopted to enlarge the training data using label-preserving transformations. Many forms are employed to do data augmentation for semantic segmentation, such as horizontally flipping, scaling, rotation, cropping and color jittering. Among them, scaling (zoom-in/zoom-out) is one of the most effective forms. DeepLab [32] augment training data by random scaling the input images (from 0.5 to 1.5). PSPNet [21] adopted random resize between 0.5 and 2 combining with other augmentation policies. In fact, the scaling means the operation

of changing the image's size. However, another reasonable explanation is given in this paper that scaling the image is the action of changing the focal length of the camera.

With this idea, a new data augmentation policy called zoom augmentation which is specially designed for fisheye image is proposed. Instead of simply resizing the image, the zoom augmentation means augmenting training dataset with additional data that is derived from existing source by changing the focal length of the fisheye camera, as illustrated in Fig. 4. Experiments validate that the zoom augmentation helps to improve generalization performance.

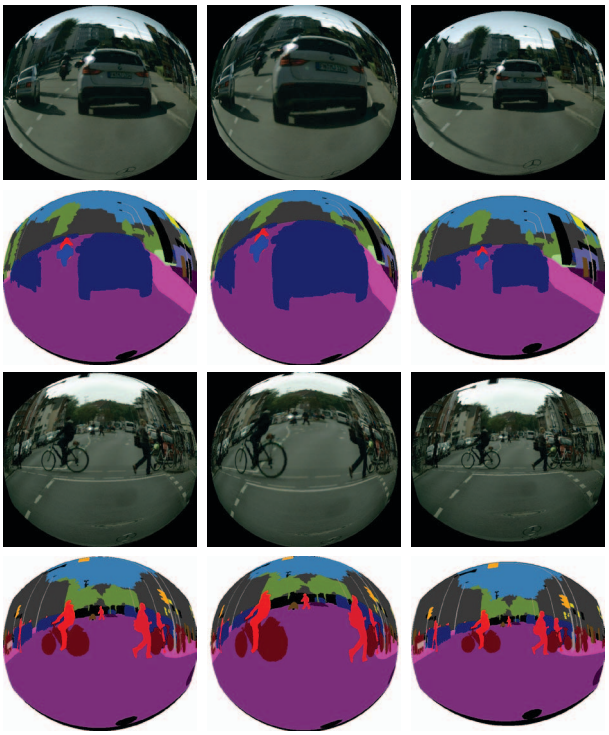
III. EXPERIMENTS

A. *Fisheye Image Dataset for Semantic Segmentation*

The fisheye image dataset for semantic segmentation is generated from the Cityscapes dataset [33]. Cityscapes is a large-scale dataset for semantic urban scene understanding. It contains 5,000 dense pixel-level annotated images selected from 27 cities with 19 classes for evaluation. The fine annotated images are split into three parts: 2975 training, 500 validation and 1525 test images. The training part and validation part are transformed to fisheye style dataset by (3) in Sec. II-C, forming the training set and validation set of the fisheye image dataset. With additional scaling operations in the mapping process, finally, the fisheye image dataset with a resolution of 576×640 is obtained by using the base focal length $f_0 = 159$. Some results of the transformation are shown in Fig. 3. As illustrated in Fig. 3, the scene of a whole intersection can be mapped into the fisheye image with a hemispheric view of about 180°.

For zoom augmentation, randomly changing the focal length within a certain range is theoretically the best choice. Nevertheless, for simplicity, this paper adopts another two scales of focal length (A smaller one $f_1 = 96$ and a bigger one $f_2 = 242$) to augment the training set. And this simplified version of zoom augmentation works pretty well. Note that, f_0 , f_1 and f_2 are set empirically. As illustrated in Fig. 4, different scales of focal length introduce different degrees of distortion.

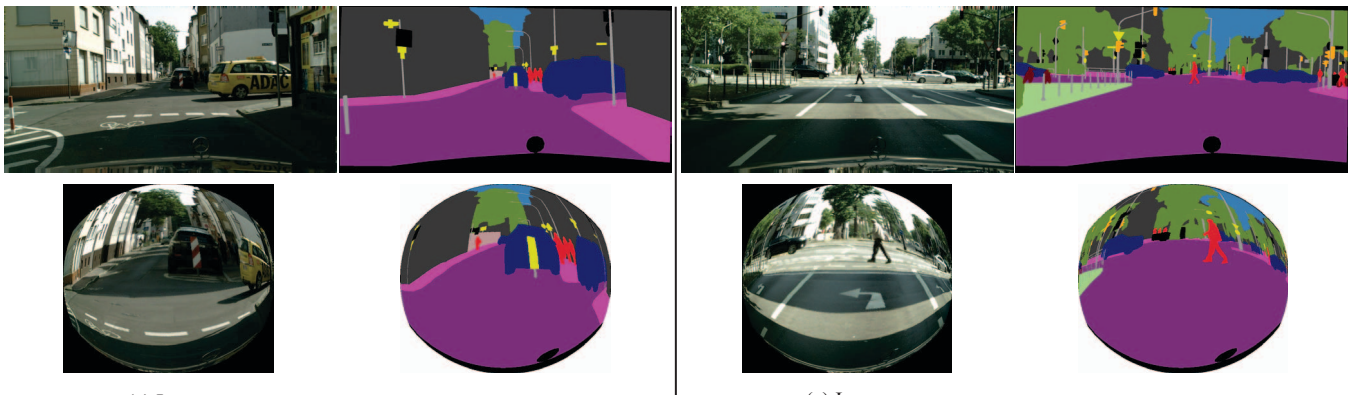
B. *Evaluation*



(a) $f_0 = 159$ (b) $f_1 = 96$ (c) $f_2 = 242$

Figure 4. Zoom augmentation for training set by employing different scales of focal length. Smaller focal length (b) introduces stronger distortions and bigger focal length (c) introduces weaker distortions.

In this section, the OPP-net is evaluated on the fisheye image dataset, which has 2975 training and 500 validation annotated images. The implementation is based on the public deep learning frame work Caffe [11]. The OPP-net was trained using SGD with momentum. The "poly" learning rate policy [11] (the learning rate is multiplied by $(1-iter/max\_iter)^{power}$) is used, which was also adopted by [21] [32]. The based learning rate and power are set to 0.01 and 0.9. A mini-batch size of 10 images and 25K iterations are employed for training. The frontend of OPP-net is based on a dilated ResNet. The first 25 convolutional layers are initialized from weights of ResNet trained for classification on a large dataset [8]. Subsequent layers are initialized by the



(a) Image (b) Annotation (a) Image (b) Annotation

Figure 3. Results of transformation from conventional image to fisheye image for urban traffic scenes. The first row is the original images and corresponding annotations in the Cityscapes [33]. The second row shows the transformed fisheye images and annotations. The same mapping function is used for (a) images and (b) annotations except different interpolations.

"xavier" method [11].

The overlapping pyramid pooling in the OPP-net has four levels: one global pooling and three overlapping pooling layers with stride 1. The kernel sizes of the three overlapping pooling layers are set to about 3/4, 1/2 and 1/4 of the kernel size of the global pooling and the sizes are restricted to odd numbers. The types of the four pooling layer are set to average operations following PSPNet [21]. For the non-overlapping pooling before the overlapping pyramid pooling, experiments are conducted with two settings, including kernel sizes of 2 and 4, pooling types of max and average. The larger the kernel size is, the more efficient the module is. As listed in Table I, the kernel size of 4 with pooling type of average work better than other settings.

On the generated fisheye image dataset for urban traffic scene, the OPP-net is compared with Dilation10 [20], Model-A and Model-B. Dilation10 is retrained with parameters initialized from weights trained on the Cityscapes dataset [33]. The complete model is jointly trained for 20K iterations on whole image, with learning rate 10^{-4}, momentum 0.99, and then another 40K iterations with learning rate multiplied by 0.1. Model-A is built up following FCRN [19] with 26-layer ResNet. A kernel size of 5 and a dilation of 12 were employed for the classifier according to [19]. Model-B is built up following PSPNet with 26-layer ResNet, and without auxiliary loss and dropout. The sub-region pyramid pooling module of Model-B is set as a four-level one with bin sizes of 1×1, 2×2, 4×3 and 8×6 respectively, so that it can fit the input image with a resolution of 576×640. Note that it didn't achieve a high mIoU score as in PSPNet on account of a relatively shallow version. Model-A and Model-B used the same training policy as the OPP-net. As shown in Table II, the OPP-net achieves the highest mIoU. The OPP-net is slightly better than the Model-B which adopted sub-region pyramid pooling module [21]. And zoom augmentation during training is effective (about 1.9% improvement). Scaling augmentation from 0.5 to 1.5 is also tested, however, it did not give such an improvement. Several examples are shown in Fig. 5.

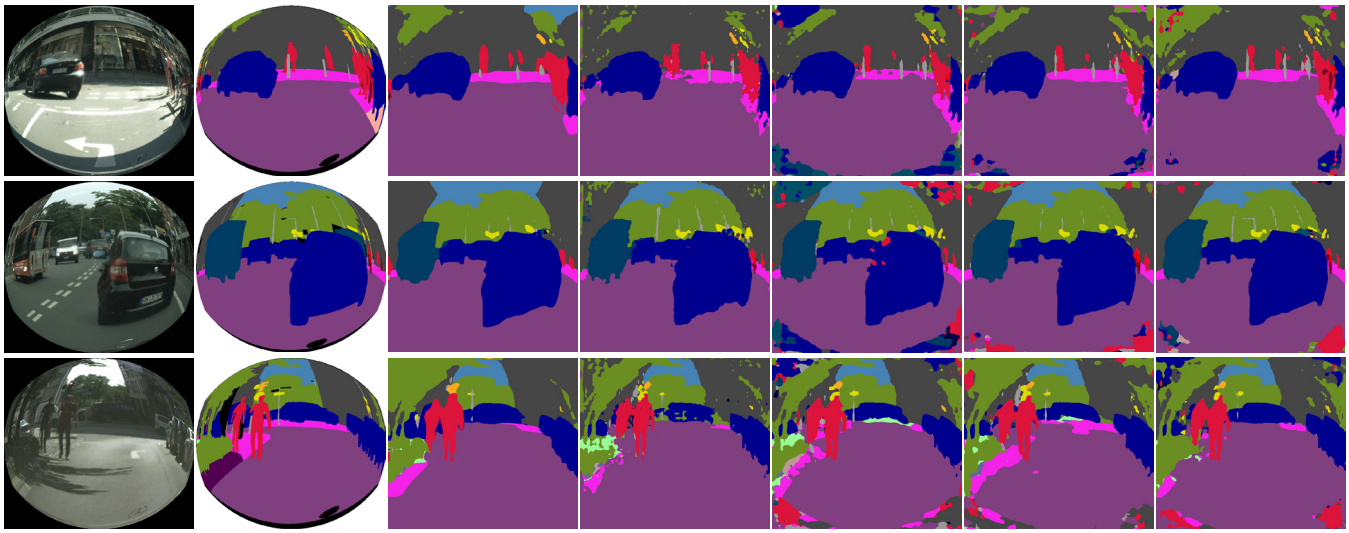
TABLE I. INVESGATION OF OPP-NET WITH DIFFTERENT SETTINGS OF THE NON-OVERLAPPING POOLING

Kernel Size	Pooling Type	mIOU
2	Max	52.29
2	Average	51.98
4	Max	52.17
4	**Average**	**52.64**

TABLE II. RESULTS ON THE VALIDATION SET OF THE GENERATED FISHEYE IMAGE DATASET FOR SEMANTIC SEGMENATION

Method	Road	Swalk	Build.	Wall	Fence	Pole	Tlight	Sign	Veg.	Terrain	Sky	Persion	Rider	Car	Truck	Bus	Train	Mbike	Bike	mIoU
Dilation10	95.4	59.9	78.1	36.6	25.4	19.1	17.6	34.6	80.5	42.0	80.7	61.0	36.0	81.3	46.8	63.9	40.5	32.0	51.5	51.7
Model A	96.3	60.6	77.9	22.7	20.3	24.4	27.0	39.9	82.5	41.5	86.6	63.1	36.3	84.8	40.3	53.9	39.3	31.8	53.6	51.7
Model B	96.4	61.1	78.3	27.3	23.4	24.9	29.0	41.7	82.7	41.3	87.3	63.6	35.7	85.3	39.3	46.4	30.7	38.1	54.6	52.0
OPP-net	96.5	61.4	78.4	23.7	22.8	24.6	28.4	41.1	82.5	39.1	87.2	63.3	34.2	85.8	40.2	56.7	39.2	41.2	53.9	52.6
OPP-net+AUG[a]	**96.7**	**63.5**	**79.6**	**26.9**	**25.4**	**25.6**	**30.6**	**44.0**	**83.2**	**43.0**	**88.8**	**65.7**	**39.4**	**86.7**	**48.6**	**55.3**	**37.2**	**40.1**	**55.4**	**54.5**

a. Random mirror is adopted during training for all the nets, 'AUG' means additional zoom augmentation is added.



Color image Ground truth Dilation10 Model-A Model-B OPP-net OPP-net+AUG

Figure 5. Results on the fisheye image dataset. Opp-net with zoom augmentation shows the best results, which is able to capture details in the edge of the image. The invalid area in the corners is not included in evaluation and Dilation10 seems to yield visually more pleasing results in the corners.

IV. CONCLUSION

This paper provided a solution for CNN-based semantic segmentation using fisheye camera in urban traffic environments. For pixel-wise semantic reasoning in the fisheye image, the OPP-net is proposed to integrate local, global and pyramid local region features in an elegant way by employing the overlapping pyramid pooling module. The OPP module allows arbitrary-sized input, keeps good translation invariant property and shows better performance than sub-region pyramid pooling module. A fisheye image dataset for semantic segmentation is generated from an existing conventional image dataset of urban traffic scenes to compensate for the lack of large-scale training dataset. In addition, zoom augmentation which is specifically designed for fisheye image is proposed by the idea of changing the focus length of the fisheye camera. A simplified version of the zoom augmentation showed the improvement for generalization. Future work needs to make random changes of focal length for zoom augmentation, and the focus of the next step is quantitative performance evaluation on real fisheye camera with annotated images.

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," in IEEE Intelligent Vehicles Symposium (IV), 2016, pp. 110-117.

[2] T. Scharwachter, M. Enzweiler, U. Franke, and S. Roth, "Stixmantics: A medium-level model for real-time semantic scene understanding," in 13th European Conference on Computer Vision. vol. 8693 LNCS Zurich, Switzerland: Springer Verlag, 2014, pp. 533-548.

[3] C. Brust, S. Sickert, M. Simon, E. Rodner, and J. Denzler, "Convolutional patch networks with spatial prior for road detection and urban scene understanding," arXiv preprint arXiv:1502.06344, 2015.

[4] F. Bernuy and J. R. D. Solar, "Semantic Mapping of Large-Scale Outdoor Scenes for Autonomous Off-Road Driving," in 2015 IEEE International Conference on Computer Vision Workshop (ICCVW), 2015, pp. 124-130.

[5] E. Romera, L. M. Bergasa and R. Arroyo, "Can we unify monocular detectors for autonomous driving by using the pixel-wise semantic segmentation of CNNs?" CoRR, vol. abs/1607.00971, 2016.

[6] T. Scharwachter and U. Franke, "Low-level fusion of color, texture and depth for robust road scene understanding," in Intelligent Vehicles Symposium (IV), 2015 IEEE: IEEE, 2015, pp. 599--604.

[7] P. Sturgess, K. Alahari, P. H. S. Torr, and E. Al., "Combining Appearance and Structure from Motion Features for Road Scene Understanding," in BMVC, 2009.

[8] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, "ImageNet Large Scale Visual Recognition Challenge," International Journal of Computer Vision, vol. 115, pp. 211-252, 2015-01-01 2015.

[9] M. Everingham, S. A. Eslami, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman, "The pascal visual object classes challenge: A retrospective," International Journal of Computer Vision, vol. 111, pp. 98--136, 2015.

[10] T. A. M. M. Lin, "Microsoft COCO: Common Objects in Context,", D. A. P. T. Fleet, Ed. Cham: Springer International Publishing, 2014, pp. 740--755.

[11] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell, "Caffe: Convolutional Architecture for Fast Feature Embedding," in Proceedings of the 22nd ACM international conference on Multimedia, Orlando, Florida, USA, 2014, pp. 675-678.

[12] M. I. N. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. J. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. J. O. Zefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. M. E, R. Monga, S. Moore, D. G. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. A. Tucker, V. Vanhoucke, V. Vasudevan, F. B. V. E. Gas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng, "TensorFlow: Large-Scale Machine Learning on Heterogeneous Distributed Systems," CoRR, vol. abs/1603.04467, 2016.

[13] R. Collobert, K. Kavukcuoglu and C. E. M. Farabet, "Torch7: A Matlab-like Environment for Machine Learning," in BigLearn, NIPS Workshop, 2011.

[14] A. Krizhevsky, I. Sutskever and G. E. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks," in Advances in Neural Information Processing Systems 25, F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, Eds.: Curran Associates, Inc., 2012, pp. 1097--1105.

[15] K. Simonyan and A. Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition," Computer Science, 2014.

[16] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," arXiv preprint arXiv:1409.4842, 2014.

[17] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," arXiv preprint arXiv:1512.03385, 2015.

[18] J. Long, E. Shelhamer and T. Darrell, "Fully convolutional networks for semantic segmentation," IEEE Transactions on Pattern Analysis & Machine Intelligence, vol. 79, pp. 1337-1342, 2014.

[19] Z. Wu, C. Shen and A. van den Hengel, "High-performance Semantic Segmentation Using Very Deep Fully Convolutional Networks," CoRR, vol. abs/1604.04339, 2016.

[20] F. Yu and V. Koltun, "Multi-Scale Context Aggregation by Dilated Convolutions," CoRR, vol. abs/1511.07122, 2015.

[21] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia, "Pyramid Scene Parsing Network," arXiv preprint arXiv:1612.01105, 2016.

[22] C. Wang, H. Zhang, M. Yang, X. Wang, L. Ye, and C. Guo, "Automatic Parking Based on a Bird's Eye View Vision System," Advances in Mechanical Engineering, vol. 6, p. 847406, 2014-01-01 2014.

[23] Y. Liu, K. Lin and Y. Chen, "Bird's-Eye View Vision System for Vehicle Surrounding Monitoring,". vol. 4931 Berlin, Heidelberg: Springer Berlin Heidelberg, 2008, pp. 207 - 218.

[24] V. Fremont, M. T. Bui, D. Boukerroui, and P. Letort, "Vision-Based People Detection System for Heavy Machine Applications," Sensors, vol. 16, p. 128, 2016.

[25] V. Haltakov, H. Belzner and S. Ilic, "Scene understanding from a moving camera for object detection and free space estimation," in Intelligent Vehicles Symposium (IV), 2012 IEEE, 2012, pp. 105-110.

[26] K. Miyamoto, "Fish Eye Lens," Journal of the Optical Society of America, vol. 54, 1964.

[27] W. Liu, A. Rabinovich and A. C. Berg, "Parsenet: Looking wider to see better," CoRR, 2015.

[28] Roozbeh Mottaghi, Xianjie Chen, Xiaobai Liu, Nam-Gyu Cho, Seong-Whan Lee, Sanja Fidler, Raquel Urtasun, and Alan Yuille, "The Role of Context for Object Detection and Semantic Segmentation in the Wild," 2014 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), vol. 00, pp. 891-898, 2014.

[29] K. He, X. Zhang, S. Ren, and J. Sun, "Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition," IEEE Transactions on Pattern Analysis & Machine Intelligence, vol. 37, pp. 1904-16, 2015.

[30] S. Ioffe and C. Szegedy, "Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift," CoRR, vol. abs/1502.03167, 2015.

[31] J. Kannala and S. S. Brandt, "A generic camera model and calibration method for conventional, wide-angle, and fish-eye lenses," IEEE Transactions on Pattern Analysis & Machine Intelligence, vol. 28, pp. 1335-40, 2006.

[32] 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," CoRR, vol. abs/1606.00915, 2016.

[33] 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 IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 3213-3223.