Universal Semantic Segmentation for Fisheye Urban Driving Images

Yaozu Ye[1], Kailun Yang[2], Kaite Xiang[1], Juan Wang[1] and Kaiwei Wang[3]

Abstract— Semantic segmentation is a critical method in the field of autonomous driving. When performing semantic image segmentation, a wider field of view (FoV) helps to obtain more information about the surrounding environment, making automatic driving safer and more reliable, which could be offered by fisheye cameras. However, large public fisheye datasets are not available, and the fisheye images captured by the fisheye camera with large FoV comes with large distortion, so commonly-used semantic segmentation model cannot be directly utilized. In this paper, a seven degrees of freedom (DoF) augmentation method is proposed to transform rectilinear image to fisheye image in a more comprehensive way. In the training process, rectilinear images are transformed into fisheye images in seven DoF, which simulates the fisheye images taken by cameras of different positions, orientations and focal lengths. The result shows that training with the seven-DoF augmentation can improve the model's accuracy and robustness against different distorted fisheye data. This seven-DoF augmentation provides a universal semantic segmentation solution for fisheye cameras in different autonomous driving applications. Also, we provide specific parameter settings of the augmentation for autonomous driving. At last, we tested our universal semantic segmentation model on real fisheye images and obtained satisfactory results. The code and configurations are released at `https://github.com/Yaozhuwa/FisheyeSeg`.

I. INTRODUCTION

With the research boom of autonomous driving, scene understanding has become a hot topic. Semantic segmentation enables pixel-by-pixel tagging of images, completing several fine detection tasks at the same time, which makes it ideal for intelligent vehicles, advanced driver assistance systems, as well as personal wearable navigation tools [1] [2].

Thanks to the emergence of large-scale natural datasets [3] [4] and architectural advances of convolutional neural networks [5] [6], most current semantic segmentation studies are based on images taken by pinhole cameras. The urban traffic environment is so complex that more information of the surroundings is required, while the pinhole camera only has a narrow FoV. If a vehicle or pedestrian suddenly appears from the blind spot, the safety of autonomous driving is difficult to guarantee. One of the approach is to increase the amount of information obtained for complete scene comprehension.

In this line, panoramic camera and multi-sensor fusion are good solutions [7] [8]. For example, by installing multiple cameras on the vehicle, or attaching additional ultrasonic

radar and LiDAR sensors, we can increase the amount of acquired information [9] [10]. However, these methods entail additional semantic mapping and fusion of data from multiple sensors, where repeated calibration and matching are required for different designed hardware collocation methods [11]. A vital subset of systems [12] [13] [14] mapped semantic segmentation from a stack of surround-view images into a bird-eye space, which works well for estimating road layout but sacrifices safety-critical view above the horizon. Nevertheless, generating on the fly a holistic representation incurs significant latency and computational cost to process multi-view images. In addition, the multi-sensor approach is cumbersome and very expensive, which could be problematic in some application scenarios [7]. A simpler and more direct way is to utilize a fisheye camera which naturally images a wide FoV [15].

While the fisheye image has a large FoV, it has large distortion. The distortion of the object depends on the view angle of the object relative to the fisheye camera. Moreover, the distortion of fisheye lenses with different focal lengths are also different. Due to the existence of distortion, the commonly-used semantic segmentation model for pinhole camera cannot be directly applied in fisheye image segmentation. There are two solutions to it. One is to rectify the fisheye image to rectilinear image, followed by common segmentation methods to obtain the wide-FoV semantic map. However, the rectification process will lead to a loss of boundary information. The output image will have a smaller FoV, which is against the original intention of using a fisheye camera.

The second approach is to carry out the segmentation directly on the fisheye images. This approach works on condition that we have a large-scale finely-annotated fisheye dataset to train our model. However, currently there is not a fisheye image dataset that fits exactly the purpose, while collecting and annotating such a dataset is expensive and laborious. WoodScape [16], a multi-task, multi-camera fisheye dataset for autonomous driving, and OmniScape [17], a synthetic dataset, could meet our requirement, but they have not been released yet. In addition, they are not as diverse and realistic as large-scale pinhole datasets [3] [4], while the large gap between synthetic and real-world domain necessitates further domain adaptation strategies. Even if we get it, there are also many problems to resolve. For example, how can we apply the model to a different fisheye camera (for they have different distortion). And is the model fits the situation that a fisheye camera installed on a different height and orientation.

Some other researches begin to train their model on synthetic fisheye dataset. The early research [18] relied on a

[1]Y. Yao, K. Xiang and J. Wang are with State Key Laboratory of Modern Optical Instrumentation, Zhejiang University, China {yaozuye, katexiang, zjuwjopt}@zju.edu.cn

[2]K. Yang is with Institute for Anthropomatics and Robotics, Karlsruhe Institute of Technology, Germany kailun.yang@kit.edu

[3]K. Wang is with National Optical Instrumentation Engineering Technology Research Center, Zhejiang University, China wangkaiwei@zju.edu.cn

$r = f\theta$ model to transform the rectilinear images to fisheye images (which they called it zoom augmentation), and trained their model with the synthetic dataset. This zoom augmentation deals the lack of the fisheye dataset to some extent, but the synthetic dataset is not rich enough. The follow-up researches [19] [20] inherited the zoom augmentation to synthesis virtual fisheye dataset, but focused on the CNN (Convolutional Neural Networks) structure design. Blott et al. [21] proposed to use the projection model transformation (PMT) to synthesize fisheye data, which they called it six-DoF (six degrees of freedom) augmentation. This six-DoF augmentation simulated the situation that camera rotated (three DoF) and shifted (three DoF) with respect to the coordinate system origin and axes. It gives a better way to generate fisheye dataset. However the research aimed at general semantic segmentation and didn't realize the great potential of the six-DoF method in the field of autonomous driving. Another work [22] on pedestrian detection also utilized the PMT to generate their fisheye data, but they only explicated the camera rotation around the vertical line (one DoF). To deal with the distortion of fisheye images, Deng et at. [14] designed a CNN structure based on deformable convolution [23] and obtained a better performance.

A big obstacle to the practical application of semantic segmentation in automatic driving is the robustness of semantic segmentation algorithm, which requires high segmentation accuracy in different scenarios. In particular, the robustness of the algorithm is critical for the safety requirements of automatic driving. The robustness of the algorithm not only depends on the design of CNN network structure, but also largely depends on the dataset we feed it.

In this paper, we focus on the data data augmentation method to transform the rectilinear dataset to synthetic fisheye dataset. Based on the six-DoF augmentation, we designed seven fisheye augmentation methods and tested them on testing sets of different distortion, and the results showed that the seven-DoF method have the best generalization capacity. So we proposed the seven-DoF augmentation method and apply it to fisheye urban driving images. This proposed seven-DoF augmentation transforms rectilinear images to virtual fisheye images taken by fisheye cameras with different angles, positions and distortion parameters during training, which is perfectly suitable for the urban driving images. It provides a universal semantic segmentation solution for fisheye cameras in different autonomous driving applications. Also, we conduct extensive experiment to investigate the hyper-parameters settings for the seven-DoF data augmentation and give specific values of our hyper-parameters settings.

In the method section, we explain the principle and intuitive understanding of the seven-DoF augmentation in detail. In the experiments section, we conduct several experiments to prove the superiority of the seven-DoF augmentation. Also, the setting of hyper-parameters for data augmentation is discussed there. Next, we test our universal semantic segmentation model on real fisheye images which are captured by a smart phone with an external fisheye lens and get satisfactory results. At last, we make a summary of the
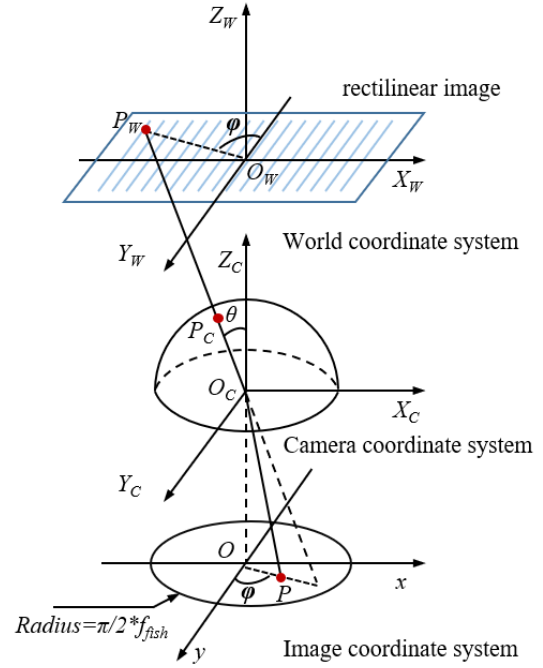

Fig. 1. Projection model of fisheye camera. P_W is a point on a rectilinear image that we place on the x-y plane of the world coordinate system. θ is the Angle of incidence of the point relative to the fisheye camera. P is the imaging point of P_W on the fisheye image. $|OP| = f\theta$. The relative rotation and translation between the world coordinate system and the camera coordinate system results in six degrees of freedom.

article and suggest some possible future research directions of fisheye segmentation.

II. METHOD

A. The principle of seven-DoF augmentation

The basic principle of data augmentation approach in this paper is to convert the rectilinear image of the world coordinate system into the synthetic fisheye image by using the projection model of the camera and a virtual fisheye camera (see Fig. 1). The so-called seven-DoF augmentation contains the spatial relationship between the world coordinate system and the fisheye coordinate system (six DoF) and the variation in the focal length of the virtual fisheye camera (one DoF). The relative rotation between the two coordinate systems contains three degrees of freedom, and the relative translation between the two coordinate systems also contains three degrees of freedom.

In actual implementation, a point on the fisheye image is mapped to a normal image as follows:

1. For a point (x_0, y_0) in the fisheye plane, it is first mapped to (x_1, y_1, z_1) in world coordinate system using the $r = f \cdot \theta$ principle. (z_1 is a relative quantity. We set it to 500 here.)

$$\theta = \frac{\sqrt{x_0^2 + y_0^2}}{f_{fish}} \tag{1}$$

$$(x_1, y_1) = (\frac{x_0}{\sqrt{x_0^2 + y_0^2}}tan(\theta), \frac{y_0}{\sqrt{x_0^2 + y_0^2}}tan(\theta)) \tag{2}$$

(a) Camera moves to the left (X) (b) Camera moves to the right (X) (c) Camera moves up (Y) (d) Camera moves down (Y)

(e) Camera moves forward (Z) (f) Camera moves back (Z) (g) Camera turns left (Y) (h) Camera turns right (Y)

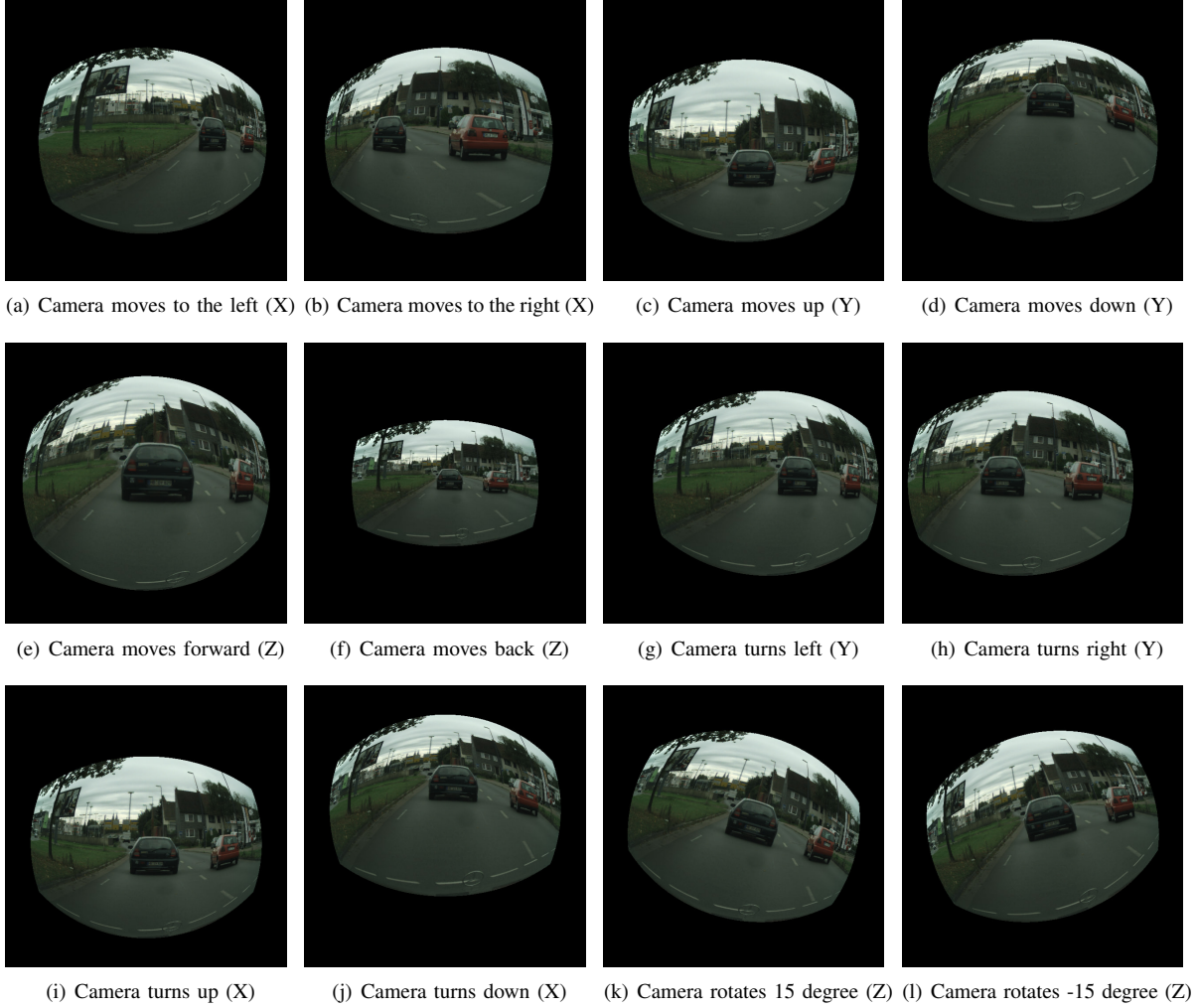(i) Camera turns up (X) (j) Camera turns down (X) (k) Camera rotates 15 degree (Z) (l) Camera rotates -15 degree (Z)

Fig. 2. The six DoF augmentation. Except the first row, every image is transformed using a virtual fisheye camera with focal length of 300 pixels. The letter in brackets means that which axis the camera is panning along or rotating around.

2. (x_1, y_1, z_1) is mapped to (x_2, y_2, z_2) in pinhole camera coordinate system by a transform matrix.

$$\begin{pmatrix} x_2 \\ y_2 \\ z_2 \\ 1 \end{pmatrix} = \begin{pmatrix} R & t \\ 0^T & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ y_1 \\ z_1 \\ 1 \end{pmatrix} \quad (3)$$

3. (x_2, y_2, z_2) is mapped to (u, v) in the rectilinear image by a camera Intrinsics. The parameters of cols and rows mean the number of column pixels and row pixels representing the rectilinear image respectively.

$$\begin{pmatrix} u \\ v \\ 1 \end{pmatrix} = \frac{1}{z_2} \begin{pmatrix} z_1 & 0 & cols/2 \\ 0 & z_1 & rows/2 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_2 \\ y_2 \\ z_2 \end{pmatrix} \quad (4)$$

By following the steps above, we can map every pixel in the fisheye image to the rectilinear image.

B. Intuitive understanding of seven-DoF augmentation

This seven-DoF augmentation, while not strictly simulating a fisheye image, can simulate its distortion pattern to some extent. Fig. 2 illustrates the effects of the six-DoF augmentation. And Fig.3 illustrates the virtual fisheye images with different focal lengths.

In the process of training neural networks, we usually use some data augmentation methods to extend the training dataset and reduce overfitting. Data augmentation methods for ordinary rectilinear data include random cropping, random horizontal flipping, panning, rotation, scaling, and color jittering. For fisheye data augmentation based on zoom augmentation, we can use all of these augmentation methods. However, when we use panning or scaling to augment our data, we just augment the data in rectilinear style, not in fisheye style. For a fisheye image, the distortion increases with the distance between the pixel and the center of the image. Besides, for the frames taken by the same fisheye camera, the distortion of the same position of each picture is the same. Therefore, once the fisheye image is translated, the distortion feature of the image will be destroyed.
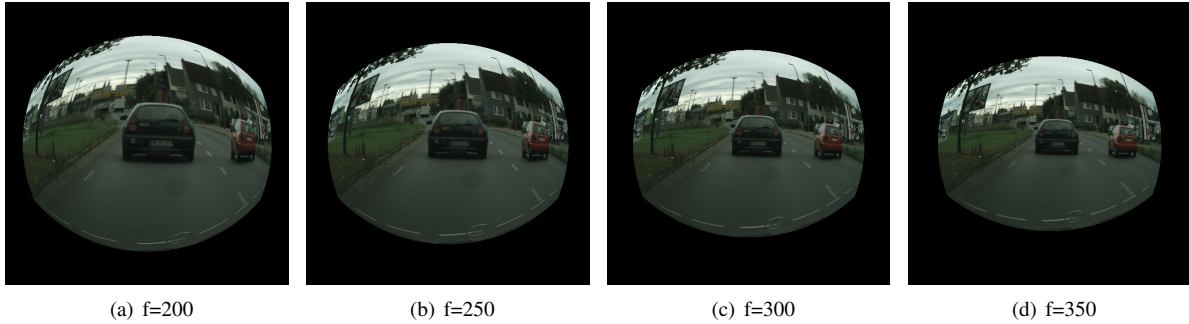
However, we can find that the seven-DoF augmentation

(a) f=200   (b) f=250   (c) f=300   (d) f=350

Fig. 3.   the synthetic fisheye images with different f(focal length)

method naturally do the data augmentation in fisheye style.

As shown in Fig. 2(e) and Fig. 2(f), when we change the relative z-axis position of the virtual fisheye camera coordinate system and the world coordinate system, it simulates the scene of fish-eye camera moving forward and backward. It makes the object closer to the fish-eye camera, which results in the object bigger in the image. The variation of the relative positions of X-axis and Y-axis between the fisheye coordinate system and the world coordinate system actually simulates the position changes of the virtual fisheye camera. Specifically, the augmentation of X-axis translation simulates the changes of the left and right position of the car on the road, while the Y-axis translation simulates the changes of the height of the fisheye camera on the car. It's also understandable that the data augmentation of rotation around three axis can simulate the orientation changes of the fisheye camera.

In practice, the fisheye camera will be placed on a car, and the attitude of the camera is always changing with the time and the turbulence of the car. Also, the position and orientation of the camera will vary from vehicle to vehicle, which results in a different view of the image. However, the neural network is not very good at handling these situation, because it is invariably trained from an existing dataset, and the accuracy of the neural network will decrease if the situation is not similar with the existing dataset. For example, when we use Cityscapes dataset [3] to train a semantic segmentation network, if we place the camera at a lower position in the actual application, the perspective of the actual image will be different from the training set, which will lead to a decrease in the accuracy. If we have a dataset which contains frames taken by cameras in different orientations and positions, that won't be a problem. But it's a huge project to collect and annotate such a dataset with cameras of different orientations and different positions, especially for fisheye cameras, as fisheye camera has a parameter of focal length and the distortion of fisheye camera varies with focal length. With the seven-DoF augmentation, we can synthesize fisheye images of the camera of different positions, orientations and focal lengths, so that a general semantic segmentation dataset of fisheye camera could be obtained.

### C. Comparison of different augmentation methods

For the seven-DoF augmentation is not generating fisheye images optically, it only simulated fisheye image to some extent. The more dimensions of freedoms it has, the bigger the difference between the synthetic image and the real fisheye scene. We can't directly conclude that seven-DoF is best for fisheye segmentation, so we conducted a series of experiments.

Now we have the following data augmentation for fisheye semantic segmentation: random cropping, random flipping, color jitter, z-aug, six-DoF augmentation and seven-DoF augmentation.For the benchmark, we adopted the data augmentation means of random clipping, random flip, color jitter and fixed f (virtual fisheye camera's focal length), and used the SwiftNet-18 as our semantic segmentation structure. The data augmentation methods are divided into three types: random focal length, random rotation and random translation.
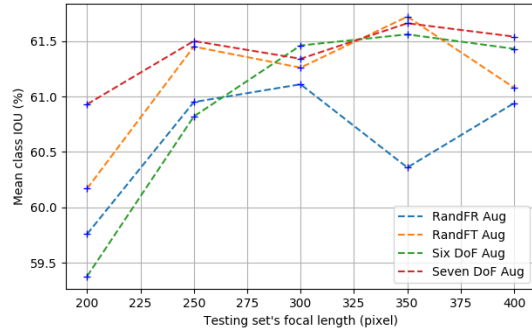
We designed the following data augmentation methods:

1. **Base Aug**: random clipping + random flip + color jitter + z-aug of fixed focal length

2. **RandF Aug**: Base Aug + random focal length

3. **RandR Aug**: Base Aug + random rotation

4. **RandT Aug**: Base Aug + random translation

5. **RandFR Aug**: Base Aug + random focal length + random rotation

6. **RandFT Aug**: Base Aug + random focal length + random translation

7. **Six-DoF Aug**: Base Aug + random rotation + random translation

8. **Seven-DoF Aug**: Base Aug + random focal length + random rotation + random translation

First, methods 1 to 4 are tested to compare the performance of a single data augmentation approach (see Fig. 4(a)). As it can be seen, when the focal length of the virtual fisheye image of the testing set is larger (the distortion is smaller), the segmentation ability of these models for the distorted image is better. Compared with Base Aug, the RandF Aug, RandR Aug and RandT Aug can evidently improve the accuracy of the model. Moreover, the RandF Aug had the best performance, while RandR Aug made the least improvement on mIoU.
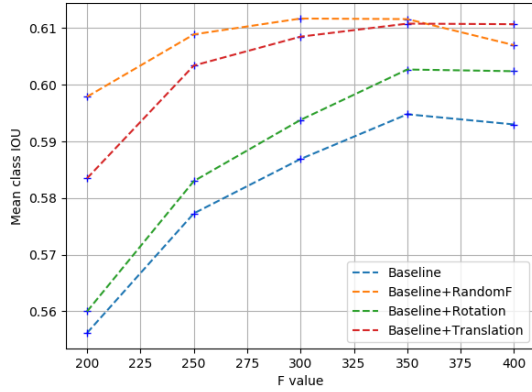
(a) Results of augmentation methods 1-4.

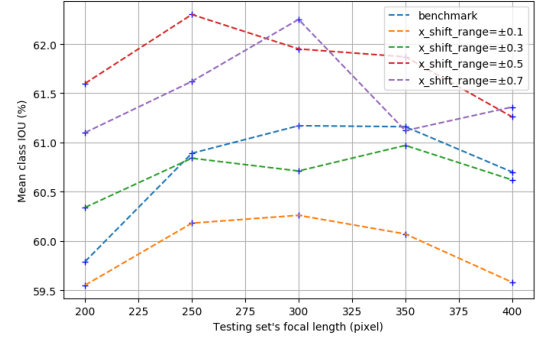(b) Results of augmentation methods 5-8.

Fig. 4.   Result of all augmentation methods.

(a) Results of different translation range along x-axis
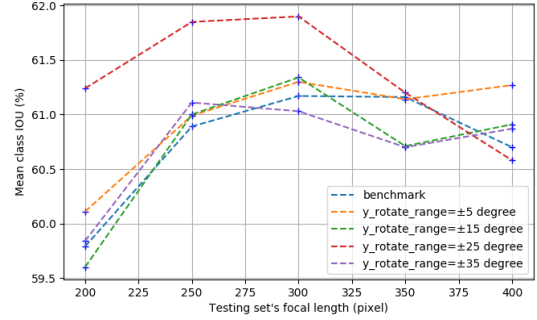
(b) Results of different rotation range around y-axis

Fig. 5.   Result of hyper-parameters settings.

Next, we test the more complex data augmentation methods (see Fig. 4(b)). The performance of the model obtained by the combination of multiple data augmentation is better than that of the single method for the testing set of different distortion parameters. It indicates that the seven-DoF Aug achieves the best performance and reaches a high mIoU in every testing sets with different degrees of distortion, which proves the robustness of the seven-DoF augmentation. However, the six-DoF Aug has a worst performance compared with other approaches. The previous experiment already shows that random focal length improves the mIoU most, while random rotation improves the mIoU least. Therefore, it's understandable that the six-DoF Aug without random focal length augmentation performs the worst.

To sum up, the performance of different data augmentation methods are shown in Table I. While the fixed z-aug performs worst, the seven-DoF augmentation we proposed almost performs best in all testing datasets. Compared with other augmentation methods, the seven-DoF has significant advantages, especially in testing datasets with larger distortion (smaller f value).

D. Hyper-parameters settings

The above experiments demonstrate the effectiveness of the seven-DoF data augmentation method. However, arbitrarily setting the parameter values of different degrees of

freedom cannot maximize the superiority of the method. In some cases, the value of a parameter is too unreasonable and may even cause the accuracy of the model to decrease. In order to set our hyper-parameters more scientifically, we conduct several experiments to test the model with different values of the hyper-parameters.

The first experiment explores the setting of the translation parameters. The translation parameters include translations along the x, y, and z axes. The translation along the x and y axes have similar effects on image distortion. We first discuss translation along the x-axis. We do not directly set an absolute pixel value as the variation range of the translation parameter, but uses a normalized value v of [0,1] to represent the translation. In the code implementation, we set $v * fish\_width$ to the pixel value of the camera coordinate system's final translation, where $fish\_width$ is the width of the virtual fisheye image finally generated. Taking the experiment with f value variation range of [200, 400] as the benchmark, the range of the camera coordinate system translation along the x axis is set to [-0.1, 0.1], [-0.3, 0.3], [-0.5, 0.5], [-0.7, 0.7] respectively. The results reveal that the model works best when $v = [-0.5, 0.5]$ (see Fig. 5(a)).

For the translation range v along the y-axis, by analogy, it should also be set to [-0.5, 0.5]. However, considering that our application scenario is urban autonomous driving, the actual meaning of the camera coordinate system translation along the y-axis is the height variation of the fisheye camera. In practice, when the vehicle is driving in different lanes,

TABLE I

PERFORMANCE OF DIFFERENT DATA AUGMENTATION

Data augmentation	mIoU ($f = 200$)	mIoU ($f = 250$)	mIoU ($f = 300$)	mIoU ($f = 350$)	mIoU ($f = 400$)
Fixed z-aug	0.5562	0.5773	0.5869	0.5948	0.5930
Random z-aug	0.5979	0.6089	0.6117	0.6116	0.6070
Six-DoF aug	0.5938	0.6082	**0.6146**	0.6156	0.6143
Seven-DoF aug	**0.6093**	**0.6150**	0.6134	**0.6166**	**0.6154**

the left and right positions of the camera may vary greatly, but the height of the camera does not change much, even for different models of cars. Therefore, the parameter v for translation along the y-axis is set to [-0.1, 0.1]. For the parameter v of translation along the z axis, in the code implementation, we normalize it to (-1, 1), and the actual translation distance is the parameter v multiplied by the focal length of the pinhole camera. Here we set it to [-0.4, 0.4]. Under this parameter range, the distortion of the virtual fisheye image will not be too much.

Similarly, for the setting of the rotation parameters, based on the experiment where the f value variation range is [200, 400] as the benchmark, the variation range of the fisheye camera coordinate system rotation around the y axis is set to [-5, 5], [-15, 15], [-25, 25], [-35, 35] degrees respectively. The results (Fig. 5(b)) indicate that the model performs best when the rotation parameter range is set to [-25, 25]. For the parameter range of the camera coordinate system rotating around the x axis, we also set it to [-25, 25] degrees. For the rotation parameter setting around the z-axis, the effect it produces is the rotation of the fisheye image that is ultimately generated. We set it to [-25, 25] degrees.

III. Experiments

A. dataset and CNN structure

CityScapes dataset [3] is a well-known dataset in the field of autonomous driving. It was recorded in street scenes from 50 different cities, and provides 5000 finely annotated frames, in addition to a larger set of 20000 coarsely annotated frames. Within the 5,000 pixel-level annotated frames, 2,975 frames were used for training, 500 frames for validation, and 1,525 frames for testing. We used the 2,975 training data and 500 validation data to conduct our experiments.

1) Training dataset: We directly use the 2975 dataset as our raw dataset, with the method of online data augmentation to transform the rectilinear data to fisheye images to train our neural networks. The original training set is the rectilinear image of 1024*2048 pixels. After data augmentation, we unified them into fisheye images of 640*640 pixels. This paper uses the method of online augmentation, that is, the parameters of seven-DoF augmentation change in every batch. The advantage of this method is that each image of the training set is transformed into a different fisheye image each time it is fed to the semantic segmentation network (the data augmentation part contains random parameters), which can greatly increase the richness of the training set.

2) Testing dataset: Testing set is to use the z-aug (zoom augmentation) to transform the 500 pieces of rectilinear cityscapes' validation data into virtual fisheye data. Different focal lengths are used to generate testing sets for a better evaluation of our models. We generate five testing sets and their focal length is 200, 250, 300, 350, 400 respectively. If the model has superior generalization performance, it should perform well on all testing sets.

3) CNN structure and training details: As this work focuses on the data augmentation, we simply choose SwiftNet-18 [6] (a lightweight CNN structure with ResNet 18 as it's backbone), which has a U-net [24] structure, to conduct our experiment. We use the SwiftNet-18 with ImageNet pre-training. Just the same as the paper [6], the pre-trained parameters are updated with Adam optimizer with learning rate of $1 \cdot 10^{-4}$. The learning rate decays with cosine annealing to the minimum value of $2.5 \cdot 10^{-5}$. And the other parameters are updated with 4 times bigger learning rate and 4 times bigger weight decay. We utilize the focal loss [25] as the loss function of the semantic segmentation. Batch size is set to 12 and we train for 200 epochs on Cityscapes. We choose the last epoch's parameters as the final model.

B. Real fisheye image test

To test the generalization performance of our model, we collected fisheye images of real urban street scenes. For convenience, we adopted an external fisheye lens for a mobile phone with a field of view of about 180 degrees and clip it to a smartphone. On the bus, we held the smartphone equipped with the external fisheye lens. As the bus navigated, we collected a series of fisheye image data of urban street scenes. We resized the obtained images to 640 * 640 resolution, and applied our model (based on seven-DoF augmentation) to the obtained images. Fig. 6 depicts the segmentation performance of our model in different scenes. As it can be seen, basically all categories are well segmented.

IV. Conclusions And Future Work

This paper proposes a general virtual fisheye data augmentation method, the seven-DoF augmentation. This method transforms a rectilinear dataset into a fisheye dataset in a comprehensible way, synthesizing fisheye images taken by cameras with different orientations, different positions, and different f values, which significantly improves the generalization performance of fisheye semantic segmentation. It provides a universal semantic segmentation solution for fisheye cameras in different autonomous driving applications. In addition, even if you already have a fisheye dataset, this

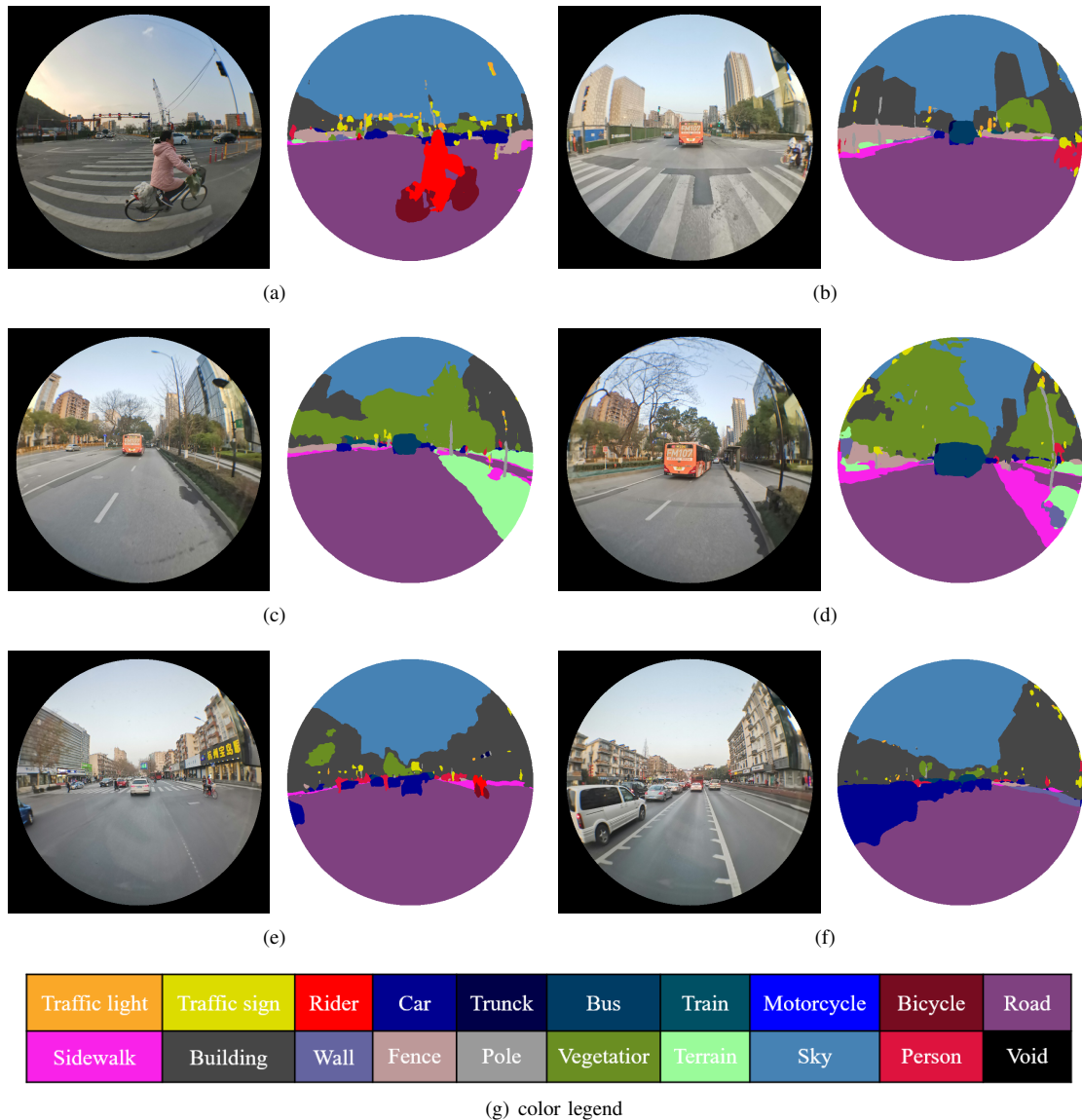| Traffic light	Traffic sign	Rider	Car	Trunck	Bus	Train	Motorcycle	Bicycle	Road
Sidewalk	Building	Wall	Fence	Pole	Vegetatior	Terrain	Sky	Person	Void

(g) color legend

Fig. 6. Semantic segmentation of real fisheye images.

method is still very meaningful. Because in practice, it is unlikely that the training set will have the same parameters as the real installed fisheye lens. The distortion parameters of different fisheye lenses are different, and the fisheye images obtained by different parameters such as orientation and camera height are also different. The dataset taken by a fisheye camera with fixed parameters cannot be well adapted to segmentation task for images taken by cameras with other different parameters.

This paper also discusses the setting of hyper-parameters for data augmentation. The parameters specially designed for urban autonomous driving scenarios evidently improves the segmentation accuracy of the model. Besides, this article proposes a convenient method to obtain fisheye images, which combines a smartphone and an external fisheye lens. Finally, when applied to real fisheye images, our model achieves precise segmentation results.

This article mainly focuses on the data augmentation method of fisheye semantic segmentation, and does not design the network structure according to the characteristics of fisheye images. In the future, we plan to make some CNN structural improvements for fisheye images specially. In addition, data augmentation methods for real fisheye datasets are also a promising research direction.

REFERENCES

[1] K. Yang, L. M. Bergasa, E. Romera, R. Cheng, T. Chen, and K. Wang, "Unifying terrain awareness through real-time semantic segmentation," in *2018 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, 2018, pp. 1033–1038.

[2] J. Wang, K. Yang, W. Hu, and K. Wang, "An environmental perception and navigational assistance system for visually impaired persons based on semantic stixels and sound interaction," in *2018 IEEE International*

Conference on Systems, Man, and Cybernetics (SMC). IEEE, 2018, pp. 1921–1926.

[3] 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 *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR).* IEEE, 2016, pp. 3213–3223.

[4] X. Huang, X. Cheng, Q. Geng, B. Cao, D. Zhou, P. Wang, Y. Lin, and R. Yang, "The apolloscape dataset for autonomous driving," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2018, pp. 954–960.

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

[6] M. Orsic, I. Kreso, P. Bevandic, and S. Segvic, "In defense of pre-trained imagenet architectures for real-time semantic segmentation of road-driving images," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 12 607–12 616.

[7] K. Yang, X. Hu, L. M. Bergasa, E. Romera, X. Huang, D. Sun, and K. Wang, "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.

[8] K. Yang, X. Hu, L. M. Bergasa, E. Romera, and K. Wang, "Pass: Panoramic annular semantic segmentation," *IEEE Transactions on Intelligent Transportation Systems*, 2019.

[9] K. E. Madawy, H. Rashed, A. E. Sallab, O. Nasr, H. Kamel, and S. Yogamani, "Rgb and lidar fusion based 3d semantic segmentation for autonomous driving," *arXiv preprint arXiv:1906.00208*, 2019.

[10] K. Yang, X. Hu, H. Chen, K. Xiang, K. Wang, and R. Stiefelhagen, "Ds-pass: Detail-sensitive panoramic annular semantic segmentation through swaftnet for surrounding sensing," *arXiv preprint arXiv:1909.07721*, 2019.

[11] N. Long, K. Wang, R. Cheng, K. Yang, and J. Bai, "Fusion of millimeter wave radar and rgb-depth sensors for assisted navigation of the visually impaired," in *Millimetre Wave and Terahertz Sensors and Technology XI*, vol. 10800. International Society for Optics and Photonics, 2018, p. 1080006.

[12] K. Narioka, H. Nishimura, T. Itamochi, and T. Inomata, "Understanding 3d semantic structure around the vehicle with monocular cameras," in *2018 IEEE Intelligent Vehicles Symposium (IV).* IEEE, 2018, pp. 132–137.

[13] Y. Wu, T. Yang, J. Zhao, L. Guan, and W. Jiang, "Vh-hfcn based parking slot and lane markings segmentation on panoramic surround view," in *2018 IEEE Intelligent Vehicles Symposium (IV).* IEEE, 2018, pp. 1767–1772.

[14] L. Deng, M. Yang, H. Li, T. Li, B. Hu, and C. Wang, "Restricted deformable convolution-based road scene semantic segmentation using surround view cameras," *IEEE Transactions on Intelligent Transportation Systems*, 2019.

[15] J. Yeol Baek, I. Veronica Chelu, L. Iordache, V. Paunescu, H. Ryu, A. Ghiuta, A. Petreanu, Y. Soh, A. Leica, and B. Jeon, "Scene understanding networks for autonomous driving based on around view monitoring system," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2018, pp. 961–968.

[16] S. Yogamani, C. Hughes, J. Horgan, G. Sistu, P. Varley, D. O'Dea, M. Uricár, S. Milz, M. Simon, K. Amende *et al.*, "Woodscape: A multi-task, multi-camera fisheye dataset for autonomous driving," *arXiv preprint arXiv:1905.01489*, 2019.

[17] A. R. Sekkat, Y. Dupuis, P. Vasseur, and P. Honeine, "The omniscape dataset," in *2020 International Conference on Robotics and Automation (ICRA).* IEEE, 2020, pp. 1–6.

[18] L. Deng, M. Yang, Y. Qian, C. Wang, and B. Wang, "Cnn based semantic segmentation for urban traffic scenes using fisheye camera," in *2017 IEEE Intelligent Vehicles Symposium (IV).* IEEE, 2017, pp. 231–236.

[19] Á. Sáez, L. M. Bergasa, E. Romeral, E. López, R. Barea, and R. Sanz, "Cnn-based fisheye image real-time semantic segmentation," in *2018 IEEE Intelligent Vehicles Symposium (IV).* IEEE, 2018, pp. 1039–1044.

[20] Á. Sáez, L. M. Bergasa, E. López-Guillén, E. Romera, M. Tradacete, C. Gómez-Huélamo, and J. del Egido, "Real-time semantic segmentation for fisheye urban driving images based on erfnet," *Sensors*, vol. 19, no. 3, p. 503, 2019.

[21] G. Blott, M. Takami, and C. Heipke, "Semantic segmentation of fisheye images," in *European Conference on Computer Vision.* Springer, 2018, pp. 181–196.

[22] Y. Qian, M. Yang, X. Zhao, C. Wang, and B. Wang, "Oriented spatial transformer network for pedestrian detection using fish-eye camera," *IEEE Transactions on Multimedia*, 2019.

[23] J. Dai, H. Qi, Y. Xiong, Y. Li, G. Zhang, H. Hu, and Y. Wei, "Deformable convolutional networks," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 764–773.

[24] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *International Conference on Medical Image Computing and Computer-Assisted Intervention*, 2015.

[25] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2980–2988.