2018 IEEE International Conference on Robotics and Automation (ICRA)
May 21-25, 2018, Brisbane, Australia

Pedestrian Feature Generation in Fish-Eye Images via Adversary

Yeqiang Qian, Ming Yang, Chunxiang Wang and Bing Wang

Abstract— **Pedestrian detection in fish-eye images is always an important problem in advanced driver assistance systems (ADAS). In conventional methods, pedestrian detectors will be trained using fish-eye images. But it is hard to collect and label enough fish-eye images manually. Therefore, a new strategy for training fish-eye pedestrian detectors using images from normal pedestrian datasets is proposed in this work. Concretely, Fish-eye Spatial Transformer Network (FSTN) is designed to generate pedestrian features in fish-eye images. FSTN aims to simulate distorted pedestrian features on the feature maps. Then the entire network is trained via adversary. FSTN is trained to generate examples which are difficult for pedestrian detectors to classify. So that the detectors are more robust to the deformation. FSTN can be embedded into state-of-the-art detectors easily. And the entire pedestrian detector, where the FSTN embedded, can be trained end to end via adversary. Moreover, experiments on ETH and KITTI pedestrian datasets show the slight accuracy improvement of pedestrian detection in fish-eye images using adversarial network compared with conventional methods.**

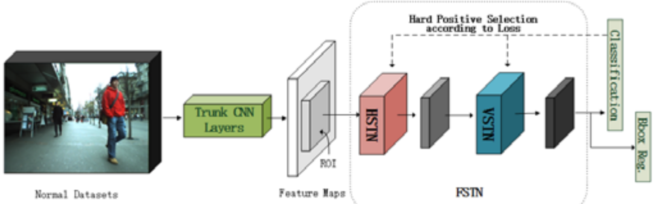

Fig. 1. Overview of proposed approach. Two contributions are in the dashed frames. FSTN is used to generate pedestrian features in the fish-eye images. The dotted line shows the adversarial learning process.

I. INTRODUCTION

Nowadays, fish-eye cameras become essential sensors in intelligent vehicles, which benefit from their wide field of view for almost 180 degrees. Fish-eye cameras provide a broader perspective compared with normal cameras and they have lower cost compared with other sensors. So pedestrian detection in the fish-eye images has a strong practical significance. In conventional methods of pedestrian detection in fish-eye images, pedestrians detectors will be trained using fish-eye images.

However, many problems exist in pedestrian detection in fish-eye images, the most serious one is the lack of pedestrian datasets of fish-eye images. Datasets play a key role in the development of deep learning based object detection algorithm. However, to the best of our knowledge, there are no available fish-eye pedestrian datasets that contain enough quantity and rich scenes like standard pedestrian benchmarks.

To solve the problem, most papers collected private fish-eye image datasets and labeled pedestrians manually such as [1][2][3][4]. In [1] and [2], Fremont and Silberstein et al. made private dataset, which contained 250 clips with a total duration of 76 minutes and over $200K$ annotated pedestrian bounding boxes. In [3], Bui et al. tested their algorithm using different datasets including their own dataset, which contained 1520 pedestrian examples. Similarly, in [4] authors combined KITTI, INRIA datasets with private datasets. All the methods produced good performance, but require too much time and dedication. Moreover, it is difficult for these datasets to cover all kinds of weathers, backgrounds and pedestrian postures compared with standard pedestrian benchmarks. Therefore, private datasets can hardly benefit other researchers.

Another strategy is taken to deal with the problem in this paper. Since normal images can be obtained easily, e.g. ETH and KITTI pedestrian dataset, fish-eye pedestrian detectors are trained using normal images instead of fish-eye images in this work. But how can pedestrian detectors learn pedestrian features in fish-eye images from normal images? A new network and its corresponding training method is proposed.

Fish-eye Spatial Transformer Network (FSTN) is designed to generate pedestrian features in fish-eye images. Spatial Transformer Network (STN) was first proposed in [5]. The difference between [5] and this work is that, in [5] STN is used to deform features to make the classification easier but FSTN is doing the exact opposite task. Moreover, many new structures have been designed in FSTN according to image formation model of fish-eye cameras. In this way, pedestrian detectors can be trained using normal pedestrian datasets directly, instead of fish-eye pedestrian datasets.

An attendant problem is that it is hard to train FSTN. Because the network is trained aiming to decrease the classification error. But original upright pedestrians are easiest for pedestrian detectors to classify. In the end, FSTN will generate examples which are similar to normal pedestrians, which is surely not desired. Moreover, false negatives often occur in similar scenes, where pedestrians show severe distortion and are more important for pedestrian detectors. So it is unwise to distort pedestrian features using FSTN randomly, and it is difficult to choose pedestrian features which are hard for detectors artificially.

In this paper, adversarial strategy is taken. Pedestrian fea-

This work was supported by the National Natural Science Foundation of China (91420101), International Chair on automated driving of ground vehicle. Ming Yang is the corresponding author.

Yeqiang Qian, Ming Yang, Chunxiang Wang and Bing Wang are with Department of Automation, Shanghai Jiao Tong University, Key Laboratory of System Control and Information Processing, Ministry of Education of China, Shanghai, 200240, China (phone: +86-21-34204553; e-mail: MingYang@sjtu.edu.cn).

978-1-5386-3081-5/18/$31.00 ©2018 IEEE

tures are generated actively, which are hard for the pedestrian detectors to recognize via adversary. Our work is inspired by [6][7][8][9]. In [6], virtual but vivid images were generated through adversarial network. In [7], adversarial learning showed effective results for image generation. In [8], authors augured that image classification in a semi-supervised setting could be improved using adversarial learning. Compared with [7], which was also the work on adversarial training, our work focuses on generating hard examples instead of better supervision. This work is also inspired by a recent work [9]. The difference is that our work designs specialised STN to distort pedestrian features instead of using the original one.

Fig. 1 shows the overview of the proposed approach. Firstly, normal pedestrian images (here ETH dataset is used as the example) are sent into CNN layers to extract features. Then features inside Region of Interest (ROI) on the feature map are send into FSTN. Here FSTN is divided into Horizontal Spatial Transformer Network (HSTN) and Vertical Spatial Transformer Network (VSTN). Two parts are combined in a sequential manner. Finally, classification and bounding box regression are conducted to get the result. The dotted line means hard positive selection according to loss. Through this feedback process, FSTN and classification are optimized in the opposite direction, which is the key of adversarial learning.

This paper is organized as follows: Fish-eye Spatial Transformer Network is introduced in Section II; the adversarial learning is detailed in Section III; experimental results are demonstrated in Section IV, followed by a conclusion in Section V.

II. FISH-EYE SPATIAL TRANSFORMER NETWORK

Max Jaderberg et al. first proposed STN in [5] and it had the great ability in spatial transformation and this work benefits a lot from it. A new STN is designed in this work to generate pedestrian features in fish-eye images. Since the imaging mechanism of fish-eye images can not be explained with a single transformation, many new structures are added. The network aims to simulate pedestrian features in fish-eye images, so it is called Fish-eye Spatial Transformer Network (FSTN).

A. FSTN Design

Fig. 2 shows the concrete structure of FSTN. FSTN has the following characteristics:

- Two STNs (HSTN and VSTN) are combined in a sequential manner.
- Each STN has three modules, including a localization network, a grid generator and a sampler.
- The input and output of HSTN are combined to send to VSTN.

The imaging mechanism of fish-eye images can not be explained with a single transformation, so FSTN is divided into two networks and each network contains two spatial transformation operations.

HSTN is used to transform features horizontally and all the pixels in the same horizontal line will perform the same operation. Similar theories can be extended to VSTN.

Note that since the output of HSTN has lost some pixels, the original feature map and the output feature map of HSTN are concatenated to add detail information. The convolution layer in the following is used to reduce the dimensions of features.

B. Three Parts Structure

In HSTN and VSTN, three parts structure in [5] is followed. Concretely, two convolution layers followed by a fully connected layer are performed to get parameters in the localization net. The grid generator contains translation and scaling process. Finally, the sampler produces the output map sampled from the input at the grid points.

To distort the pedestrian features in fish-eye images, translation and scaling process are performed. In general, the input feature map is defined as $FI \in R^{H\times W\times C}$, where H, W and C are the height, width and channels of the input feature map. The input pixels are defined to lie on a regular grid $G = \{G_i\}$ of pixels $G_i = (x_i, y_i)$. Similarly, the output feature map has the same shape of input one and the output pixels are defined as $G_i = (X_i, Y_i)$.

For HSTN, the translation and scaling transformation is

$$T^h_\theta(G) = \begin{bmatrix} X_i & Y_i & 1 \end{bmatrix} \begin{bmatrix} sx & 0 & 0 \\ 0 & 1 & 0 \\ dx \cdot sx & 0 & 1 \end{bmatrix} \quad (1)$$

In the equation (1), dx and sx are the translation value and the scaling value in X direction respectively. Similarly, dy and sy can be defined in Y direction in VSTN. Parameters in the same horizontal line share the same value. In HSTN, the translation and scaling values in Y direction are set to 0.

After the parameterized sampling grid, the sampler is used to produce the output feature map FO. This can be written as

$$FO_i = \sum_{h=1}^{H} \sum_{w=1}^{W} FI_{hw}(1 - |x_i - h|)(1 - |y_i - w|) \quad (2)$$

Here equation (2) shows a bilinear sampling kernel. FI_{hw} is the value at location (h, w) in the input feature map. All the positions will be traversed in FI and (x_i, y_i) is calculated from formula above. The same operation will be performed in all channels, so channels are omitted in the equation. The sampler in HSTN is the same as one in VSTN. The normalization is carried out in X and Y coordinates, such that $0 \leq x_i, y_i \leq 1$ in FI.

C. Back Propagation

Gradients with respect to FI and G are defined to allow backpropagation of the loss through the sampler as shown in the following.

$$\frac{\partial FO_i}{\partial FI_{hw}} = \sum_{h=1}^{H} \sum_{w=1}^{W} (1 - |x_i - h|)(1 - |y_i - w|) \quad (3)$$

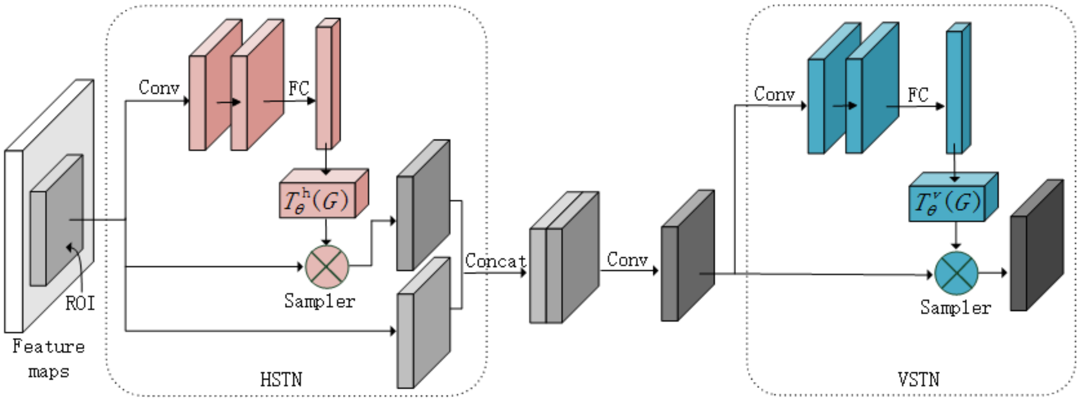

Fig. 2. The concrete structure of FSTN, including a HSTN and a VSTN. Each STN has three modules. The input and output of HSTN are combined to send to VSTN. **Conv** is convolution layer and **FC** is fully connected layer.

$$\frac{\partial FO_i}{\partial x_i} = \sum_{h=1}^{H}\sum_{w=1}^{W} FI_{hw}(1-|y_i-w|)k_x, k_x = \begin{cases} -1, if\ x_i \geq h \\ 1, if\ x_i < h \end{cases} \quad (4)$$

$$\frac{\partial FO_i}{\partial y_i} = \sum_{h=1}^{H}\sum_{w=1}^{W} FI_{hw}(1-|x_i-h|)k_y, k_y = \begin{cases} -1, if\ y_i \geq w \\ 1, if\ y_i < w \end{cases} \quad (5)$$

III. ADVERSARIAL NETWORK

With FSTN, distorted pedestrian features in fish-eye images can be generated easily. So real fish-eye images are not necessary in the training process. However, it is still hard to train FSTN. Since the network will be trained in order to decrease the classification error, FSTN will generate examples which are easy for detectors to classify. In fact, the original upright pedestrians are easiest to recognize. In other words, FSTN fails to work in reality.

In this paper, adversarial learning method is used to train the entire network. It is assumed that FSTN should generate examples which are hard for detectors to recognize, and these examples are more important for detectors. Specifically, FSTN is trained in order to increase the classification error and FSTN based detector is trained in the opposite direction.

A. Adversarial Learning

Mathematically, the original detector is trained to minimize an objective function following the multi-task loss, which is defined as:

$$L_D = \sum_{i=1}^{N} L_{cls}(D_c(P^i), C^i) + \sum_{i=1}^{N} C^i L_{reg}(D_l(P^i), L^i) \quad (6)$$

In equation (6), L_{cls} is the loss function of classification and L_{reg} is the loss function of bounding box regression. P^i is one of proposals with order i and there are N proposals in total. The original detector network is represented as $D(P)$, including $D_c(P)$ for calculating the class and $D_l(P)$ for calculating the location. C^i is the ground-truth class for proposal P^i and L^i is the ground-truth location. Note that, if proposal P^i is defined as background, bounding box regression will not be calculated because C^i is 0.

The loss function of adversarial network can be defined as:

$$L_S = -\sum_{i=1}^{N} L_{cls}(D_c(S(P^i)), C^i) \quad (7)$$

In equation (7), FSTN is represented as $S(P)$. $D(S(P))$ is the entire detector where FSTN embedded. Note that bounding box regression is omitted in the adversarial network. Minus sign means that FSTN has the opposite direction of optimization compared with original detector, which is the key idea of adversarial learning.

Concretely, if examples generated by FSTN are hard for the detector to recognize, the detector will get a high loss and the adversarial network will get a low one. On the contrary, if examples generated by FSTN are easy for the detector to recognize, the adversarial network will get a high loss and the detector will get a low one.

B. Training Steps

Motivated by the Faster R-CNN [10], stage-wise training method is applied in this paper. Here training process is divided into three steps.

- Firstly, original detector is trained without FSTN. Now the detector has a sense of the normal upright pedestrians.
- Then FSTN is inserted into original detector and we train FSTN for 10K iterations. Note that all the parameters in original detectors will be fixed in this step. Now the FSTN has the ability to generate warped pedestrian features.
- Finally, the FSTN based detector is trained for extra 80K iterations.

Note that FSTN and adversarial learning are only applied during training process. And all the images used in the training process are normal images. In other words, fish-eye images are unnecessary, which is main contribution of this work.

C. Combining FSTN with Universal Detectors

It is convenient to combine FSTN with universal detectors, forming complete adversarial networks. Here three state-of-the-art object detectors are used as examples, including Faster R-CNN [10], MS-CNN [11] and SSD [12]. FSTN can be inserted into Faster R-CNN and MS-CNN before ROI pooling layer. Since MS-CNN has multiple branches, multiple FSTNs are used and they do not share values. SSD conducts object classification and bounding box regression directly, without ROI pooling layer. So FSTN is inserted after trunk CNN layers.

D. Implementation Details

Experiments show that it is very important to limit the translation and scaling value in FSTN, because pedestrians in fish-eye images will not deform randomly. If adversarial network is trained without limitation, features generated by FSTN will show very large deformation and strange gestures. This is also the reason why two STNs are designed instead of one. However, it is not enough.

It is assumed that translation and scaling value between adjacent rows and columns should change consecutively in fish-eye images. More strictly, they obey normal distribution, i.e., $X \sim N(\mu, \sigma^2)$, $X \in \{sx, dx, sy, dy\}$. Here sx, dx, sy and dy are defined in equation (1). Therefore, real outputs of localization network in FSTN are μ_i^j and σ_i^j, $i \in \{x, y\}$ and $j \in \{s, d\}$.

Besides, random selection is applied in the third training step, which is similar to *dropout* process. Specifically, only 30% samples are sent into FSTN, so that the detector still has a sense of the normal upright pedestrians through adversarial learning.

IV. EXPERIMENTS

The proposed approach is evaluated using three deep learning based detectors, including Faster R-CNN [10], MS-CNN [11] and SSD [12]. Experiments are based on ETH [13] and KITTI [14] pedestrian datasets. Our experiments are all based on VGG-16 model [15], which is pre-trained on the ILSVRC CLS-LOC dataset [16].

A. Evaluation Methods and Experimental Settings

Since there is no available fish-eye image dataset, an imaging model conversion algorithm from [17] is used to generate fish-eye images and corresponding pedestrian labels based on normal images. In this paper, ETH and KITTI datasets are used to generate fish-eye image datasets. Please refer to [17] for more technical details.

The evaluation method used in experiments refers to the KITTI standard, please refer to [14] for more technical details. KITTI evaluation standard contains three different difficulty levels and moderate level is used when drawing the PR-curve graph. Our CPU is Intel(R) Xeon(R) E5-2620 v4 @2.10GHz, and our GPU is NVIDIA GTX 1080.

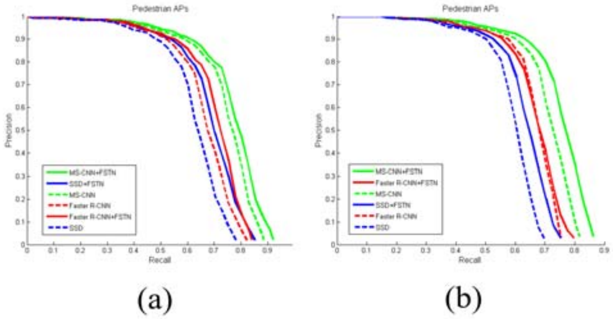

Fig. 3. Comparison between original pedestrian detectors and FSTN based pedestrian detectors. (a) is based on ETH dataset and (b) is based on KITTI dataset.

B. Experiments on ETH

According to [17], ETH dataset is converted to corresponding fish-eye dataset. The 5361 images are divided into two parts, including 4000 images for training and 1361 images for testing. Note that FSTN is only used in the training process. Proposed approach is trained using normal images while corresponding original detector is trained using fish-eye images.

Fig. 4 (a) shows some detection results on the converted fish-eye images based on ETH. Fig. 3 (a) is the comparison between proposed approach and conventional one. The FSTN based detectors are trained using normal images via adversary while original detectors are trained using fish-eye images. From the figure, FSTN based approach improves the accuracy of pedestrian detection slightly.

Detailed results are shown in the Table I. In Table I, **AD** means using the adversarial learning in the training process and **NON** means a non-adversarial training process. With FSTN and adversarial learning strategy, the three detectors achieve 3.88% AP improvement averagely compared with conventional methods.

C. Experiments on KITTI

Fish-eye image dataset based on KITTI [14] can be generated similarly. Note that since we only have the annotated files of training set of KITTI dataset, the whole training part of KITTI dataset is divided into two parts, including 3682 images for training and 3799 images for testing. Fig. 4 (b) shows some results on the converted fish-eye images based on KITTI. Fig. 3 (b) is the similar results with Fig. 3, but it is based on KITTI dataset. Note that moderate level is used when drawing the PR-curve graph. Except Faster R-CNN, two other detectors show slight accuracy improvement.

Detailed results are shown in the Table II. With FSTN and adversarial learning strategy, the three detectors achieve 2.24% AP (moderate level) improvement averagely compared with conventional methods. The results prove the feasibility of using proposed approach to replace conventional outlines, i.e., training fish-eye pedestrian detectors using normal images. Note that, FSTN based Faster R-CNN does not show the sound result. But the difference is little compared with original one (0.45%).

TABLE I

Results based on ETH dataset. **AD** means using the adversarial learning in the training process and **NON** means a non-adversarial training process.

Training Process				Testing Process		
Detector		Training Data	Training Method	Testing Data	AP(%)	Time($ms/frame$)
Faster R-CNN		Fish-eye Image	NON	Fish-eye Image	70.39	109
	+FSTN	Normal Image	AD	Fish-eye Image	**74.66**	109
MS-CNN		Fish-eye Image	NON	Fish-eye Image	76.79	136
	+FSTN	Normal Image	AD	Fish-eye Image	**79.41**	136
SSD		Fish-eye Image	NON	Fish-eye Image	67.89	51
	+FSTN	Normal Image	AD	Fish-eye Image	**72.64**	51

TABLE II

Results based on KITTI dataset.

Training Process				Testing Process				
Detector		Training Data	Training Method	Testing Data	AP(%)			Time
					easy	moderate	hard	($ms/frame$)
Faster R-CNN		Fish-eye Image	NON	Fish-eye Image	**76.44**	**64.58**	59.97	107
	+FSTN	Normal Image	AD	Fish-eye Image	75.72	64.13	**60.87**	107
MS-CNN		Fish-eye Image	NON	Fish-eye Image	77.73	67.29	61.42	186
	+FSTN	Normal Image	AD	Fish-eye Image	**79.51**	**71.24**	**63.73**	186
SSD		Fish-eye Image	NON	Fish-eye Image	65.93	59.76	51.49	68
	+FSTN	Normal Image	AD	Fish-eye Image	**72.83**	**63.98**	**59.28**	68

TABLE III

Analysis of FSTN based on ETH dataset. **Concat** means combining the input and output of HSTN. **HSTN+VSTN+Concat** is equal to **FSTN**.

Training Process				Testing Process		
Detector		Training Data	Training Method	Testing Data	AP(%)	Time ($ms/frame$)
Faster R-CNN		Normal Image	NON	Fish-eye Image	65.77	109
	+HSTN	Normal Image	AD	Fish-eye Image	70.23	109
	+VSTN	Normal Image	AD	Fish-eye Image	69.27	109
	+HSTN +VSTN	Normal Image	AD	Fish-eye Image	72.63	109
	+HSTN +VSTN +Concat (FSTN)	Normal Image	NON	Fish-eye Image	70.85	109
	+HSTN +VSTN +Concat (FSTN)	Normal Image	AD	Fish-eye Image	**74.66**	109
MS-CNN		Normal Image	NON	Fish-eye Image	66.29	136
	+HSTN	Normal Image	AD	Fish-eye Image	74.98	136
	+VSTN	Normal Image	AD	Fish-eye Image	73.21	136
	+HSTN +VSTN +Concat (FSTN)	Normal Image	NON	Fish-eye Image	73.80	136
	+HSTN +VSTN +Concat (FSTN)	Normal Image	AD	Fish-eye Image	**79.41**	136
SSD		Normal Image	NON	Fish-eye Image	61.98	51
	+HSTN	Normal Image	AD	Fish-eye Image	70.68	51
	+VSTN	Normal Image	AD	Fish-eye Image	69.93	51
	+HSTN +VSTN +Concat (FSTN)	Normal Image	NON	Fish-eye Image	67.98	51
	+HSTN +VSTN +Concat (FSTN)	Normal Image	AD	Fish-eye Image	**72.64**	51

D. Analysis of FSTN

Different combinations are tried in experiments to analyze FSTN. The detailed results are shown in the Table III. With concatenation process, Faster R-CNN achieves 2.03% AP improvement. Note that, HSTN is more effective than VSTN (1.42% improvement), which matches imaging principle of fish-eye cameras. Because greater distortion occurs in the horizontal direction.

E. Analysis of Adversarial learning

As it is shown in the Table III, adversarial learning plays a key role in training FSTN. Without adversarial learning method, FSTN based detectors show similar results compared with original ones. And FSTN based networks achieve 4.69% AP improvement averagely using adversarial learning, which verifies the previous analysis.

V. CONCLUSIONS

In this paper, FSTN is designed to generate pedestrian features in fish-eye images. Adversarial learning method is

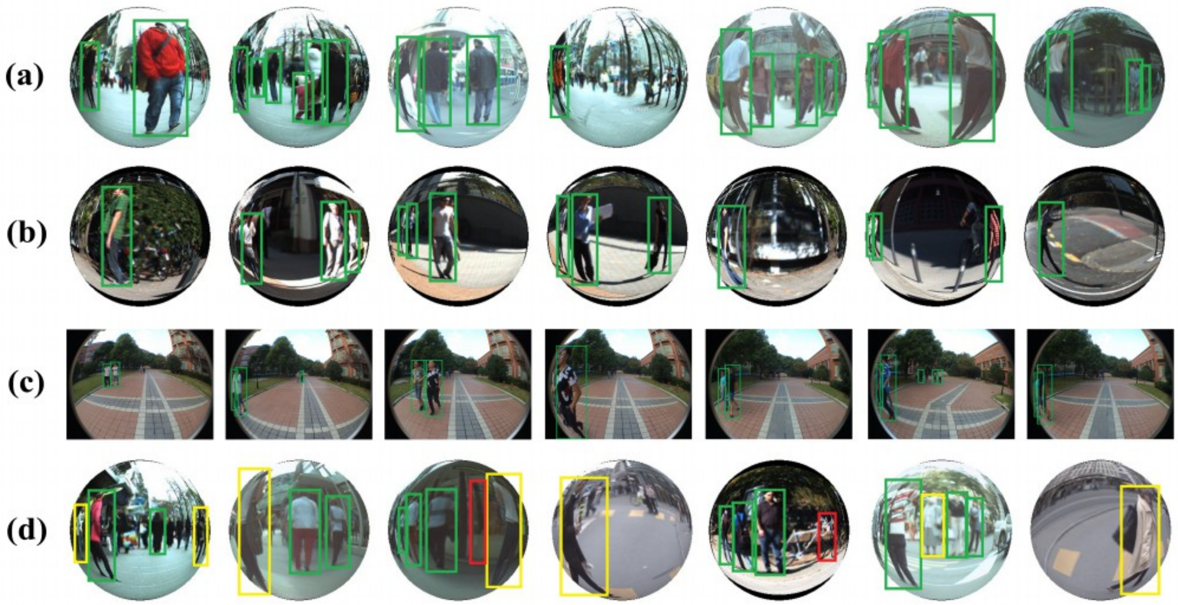

Fig. 4. Some results on the converted fish-eye images. (a): Converted ETH dataset. (b): Converted KITTI dataset. (c): Real fish-eye images. (d): False positives and false negatives. In ETH fish-eye dataset, the training part contains 4000 images and testing part contains 1361 images. In KITTI dataset, the training part contains 3682 images and testing part contains 3799 images. The red bounding boxes in (d) are false positives and the yellow ones are false negatives.

used to train the network. Using adversarial learning, FSTN can generate examples that are hard for detectors to classify. Therefore, FSTN based pedestrian detectors are more robust to pedestrian deformation in fish-eye images. Using this strategy, pedestrian detectors can be trained using normal pedestrian datasets instead of fish-eye image datasets. The entire FSTN based pedestrian detector shows considerably stronger ability to detect pedestrians in fish-eye images compared with conventional methods. Moreover, it is believed that, with slight modification, proposed approach will adapt to various images of different field of views using only normal images for training.

References

[1] V. Fremont, M. T. Bui, D. Boukerroui, and P. Letort, "Vision-based people detection system for heavy machine applications," *Sensors*, vol. 16, no. 1, p. 128, 2016.

[2] D. Levi and S. Silberstein, "Tracking and motion cues for rear-view pedestrian detection," in *Intelligent Transportation Systems (ITSC), 2015 IEEE 18th International Conference on*. IEEE, 2015, pp. 664–671.

[3] M.-T. Bui, V. Frémont, D. Boukerroui, and P. Letort, "Deformable parts model for people detection in heavy machines applications," in *Control Automation Robotics & Vision (ICARCV), 2014 13th International Conference on*. IEEE, 2014, pp. 389–394.

[4] M. Bertozzi, L. Castangia, S. Cattani, A. Prioletti, and P. Versari, "360 detection and tracking algorithm of both pedestrian and vehicle using fisheye images," in *Intelligent Vehicles Symposium (IV), 2015 IEEE*. IEEE, 2015, pp. 132–137.

[5] M. Jaderberg, K. Simonyan, A. Zisserman *et al.*, "Spatial transformer networks," in *Advances in Neural Information Processing Systems*, 2015, pp. 2017–2025.

[6] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Advances in neural information processing systems*, 2014, pp. 2672–2680.

[7] A. Radford, L. Metz, and S. Chintala, "Unsupervised representation learning with deep convolutional generative adversarial networks," *arXiv preprint arXiv:1511.06434*, 2015.

[8] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. Chen, "Improved techniques for training gans," in *Advances in Neural Information Processing Systems*, 2016, pp. 2234–2242.

[9] X. Wang, A. Shrivastava, and A. Gupta, "A-fast-rcnn: Hard positive generation via adversary for object detection," *arXiv preprint arXiv:1704.03414*, 2017.

[10] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," in *Advances in neural information processing systems*, 2015, pp. 91–99.

[11] Z. Cai, Q. Fan, R. S. Feris, and N. Vasconcelos, "A unified multi-scale deep convolutional neural network for fast object detection," in *European Conference on Computer Vision*. Springer, 2016, pp. 354–370.

[12] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "Ssd: Single shot multibox detector," in *European conference on computer vision*. Springer, 2016, pp. 21–37.

[13] E. benchmark, "https://data.vision.ee.ethz.ch/cvl/aess/dataset/."

[14] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? the kitti vision benchmark suite," in *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*. IEEE, 2012, pp. 3354–3361.

[15] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[16] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein *et al.*, "Imagenet large scale visual recognition challenge," *International Journal of Computer Vision*, vol. 115, no. 3, pp. 211–252, 2015.

[17] Y. Qian, M. Yang, C. Wang, and B. Wang, "Self-adapting part-based pedestrian detection using a fish-eye camera," in *Intelligent Vehicles Symposium (IV), 2017 IEEE*. IEEE, 2017, pp. 33–38.