Pedestrian Feature Generation in Fish-eye Images via Adversary

Yeqiang Qian[1], Ming Yang[1], Chunxiang Wang[2] and Bing Wang[1]

Abstract— **Pedestrian detection in fish-eye images is always an important problem in advanced driver assistance systems (ADAS). One main problem is the lack of pedestrian datasets of fish-eye images. Without massive fish-eye images, universal pedestrian detectors can not show all the qualities. Instead of collecting fish-eye images and labeling manually, another strategy is proposed in this work. (1) Firstly, the spatial transformer network (STN) is designed to generate pedestrian features in fish-eye images. Two STNs (Horizontal STN and Vertical STN) are combined to simulate distorted pedestrian features on the feature maps. In this way, fish-eye pedestrian detectors can be trained using normal pedestrian datasets. (2) Moreover, the entire network is trained via adversary. In particular, the designed STNs are trained to generate examples which are difficult for pedestrian detectors to classify. So that the detectors are more robust to the deformation. The STN can be embed into state-of-art detectors easily. And the entire pedestrian detector, where the STN embeded, can be trained end to end via adversary. Experiments on ETH and KITTI pedestrian datasets show the accuracy improvement of pedestrian detection in fish-eye images using adversarial network.**

I. INTRODUCTION

Nowadays, fish-eye cameras become essential sensors in intelligent vehicles because of 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. The panoramic system, which is widely applied in advanced vehicles, is composed of 4 or more fish-eye cameras [1]. So pedestrian detection in the fish-eye images has a strong practical significance. However, problems expose when performing pedestrian detection in fish-eye images, the most important one is the lack of pedestrian datasets of fish-eye images.

Datasets play a key role in the development of object detection algorithm. The appearance of numerous datasets bring the boom of deep learning algorithms. A quality dataset is not only useful for evaluating a specific approach, but is also improves the performance on the object detection in the training process. 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. In fact, massive pedestrian datasets spring up these years and benefit many object detectors, such as ETH [2], KITTI [3] and COCO [4] benchmarks. But they are all normal images instead of fish-eye images.

To solve the problem, most papers collected their own fish-eye image datasets and labeled pedestrians manually such as [5][6][7][8]. In [5] and [6], Shai 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 [7], Manh-Tuan Bui et al. tested their algorithm using different datasets including their own dataset, which contained 1520 pedestrian examples. Similarly, in [8] authors combined KITTI, INRIA with private datasets. All the methods got great feedback, but cost too much time and effort. Moreover, it is difficult for these datasets to cover all kinds of weathers, backgrounds and pedestrian postures compared with standard pedestrian benchmarks. Also, it is difficult to guarantee the quality of private datasets, so the datasets can not benefit others' works.

So another strategy is taken to deal with the problem. The spatial transformer network (STN) is designed to distort normal pedestrians on the feature maps, which aims to simulate distorted pedestrian features in the fish-eye images. STN is first proposed in [9], it is instantiated and optimized when applying to the field of fish-eye images. The difference between [9] and our work is that, in [9] the STN is used to deform the features to make the classification easier but our network is doing the exact opposite task. Moreover, STN designed in this paper operates on each pixel discriminatively instead of the single transformation for the entire image. 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 the designed STN. Because the network is trained aiming to decreasing the classification error. But original upright pedestrians are easiest for pedestrian detectors to classify. In the end, the designed STN will generate examples which are similar to normal pedestrians, that is surely not desired. Moreover, false negatives are often occurred in similar scenes, where pedestrians show severe distortion and are more important for pedestrian detectors. So it is unwise to distort pedestrian features using STN randomly, and it is difficult to choose pedestrian features which are hard for detectors artificially.

In this paper, adversarial strategy is taken. Pedestrian features are generated actively which are hard for the pedestrian detectors to recognize via adversary. Our work is inspired by Generative Adversarial Networks (GAN) [10]. In [11], adversarial learning showed effective results for image generation. In [12], authors augured that image classification in a semi-supervised setting can be improved using adver-

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.

[1]Yeqiang Qian, Ming Yang 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).

[2]Chunxiang Wang is with Research Institute of Robotics, Shanghai Jiao Tong University, Shanghai 200240, China.

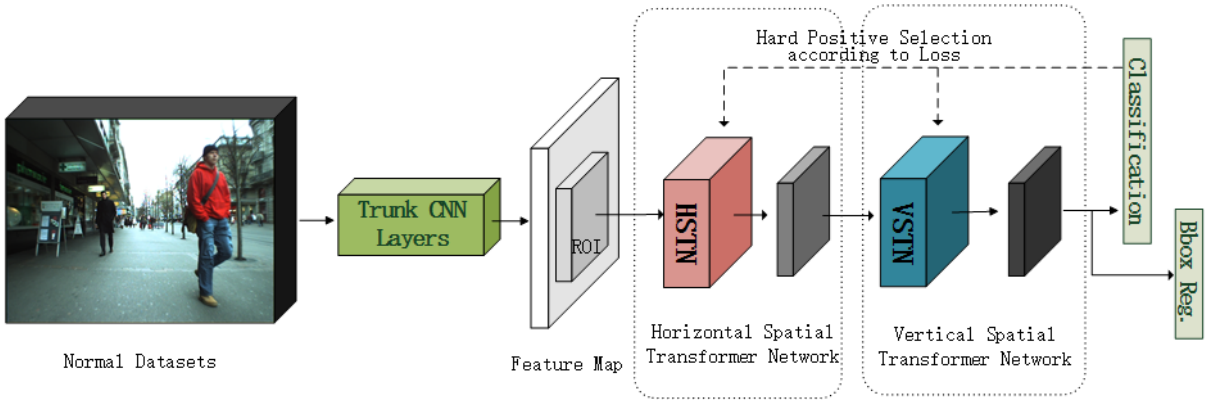

Fig. 1. Overview of proposed approach. Designed STN is in the dashed frames. The dotted line shows the adversarial learning.

sarial learning. Compared with [11], which is also the work on adversary training, our work focus on generating hard examples instead of better supervision. This work is also inspired by a recent work [13]. 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 proposed approach. Firstly, normal pedestrian images (here ETH dataset is used as the example) are send into CNN layers to extract features. Then features inside Region of Interest (ROI) on the feature map are send into STNs. Here STNs are divided into horizontal spatial transformer network (red part) and vertical spatial transformer network (blue part). 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 network, STNs and classification are optimized in the opposite direction, which is the key of adversarial learning.

This paper is organized as follows: the spatial transformer network is introduced in Section II; the adversarial network is detailed in Section III; experimental results are demonstrated in Section IV, followed by a conclusion in Section V.

II. STN BASED NETWORK

Max Jaderberg et al. first proposed STN in [9] and it had the ability to spatial transformation. The original spatial transformer mechanism was divided into three parts, including a localization network, a gird generator and a sampler. The localization network was used to learn parameters of the spatial transformation, which were trained from image features. The gird generator created a sampling grid, that was a set of points where the input map should be sampled to produce the transformed output. In the end, the rectified features were produced by the sampler. Note that the entire network was differentiable, so it could be trained using back propagation algorithm easily.

STN was proposed with a universal model, without concrete structure and methods of combining with state-of-art detectors. Moreover, STN in [9] was used to rectify distorted MNIST datasets, where a single transformation could be applied for the entire image. But the imaging mechanism of fish-eye images can not be explained with a single transformation. Therefore in this work, STN is optimized when applying to the field of fish-eye images:

- Each pixel in feature maps is operated distinguishably. In particular, different pixels will take different actions depending on parameters learned.
- Two STNs (horizontal STN and vertical STN) are proposed to simulate the imaging mechanism of fish-eye images. They are also used to limit the transform operations, guaranteeing convergence.
- The input and output of horizontal STN are combined to send to vertical STN. The design improves the detection accuracy in experiments compared with simple tandem.

Fig. 2 shows the concrete structure of proposed STN. The two STN networks are combined in a sequential manner. The horizontal STN is used to transform features in ROI horizontally and all the pixels in the same horizontal line will perform the same operation. Similar theories can be extended into vertical STN. In the fish-eye images, the transform operation is performed into individual pixels instead of the entire image. The original STN is divided into two parts to limit transform operations of pixels, which is also conducive to the convergence in the training process.

In each STN, we follow the three parts structure in [9]. Concretely, two convolution layers followed by a fully connected layer are performed to get the parameters in the localization net. The gird generator contains translation and scaling process. Finally, the sampler produces the output map sampled from the input at the grid points. Note that since the output put of horizontal STN has lost some pixels, the original feature map and the output feature map of the horizontal STN are concatenate to initialize the vertical STN. The convolution layer in the following is used to reduce the dimensions of features.

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

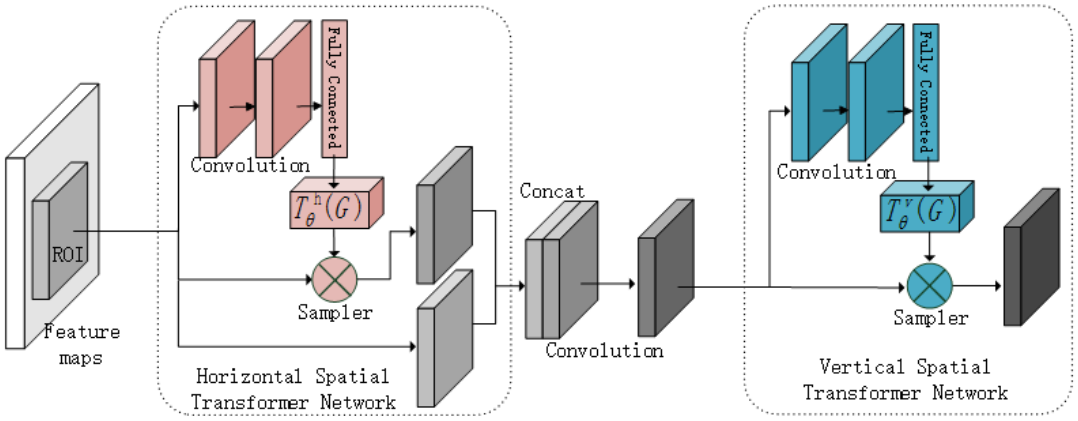

Fig. 2. The concrete structure of designed STN, including a horizontal STN and a vertical STN.

feature map has the same shape of input one and the output pixels are defined as $G_i = (X_i, Y_i)$.

For the horizontal STN, the translation and scaling transformation is

$$\begin{aligned} &\begin{bmatrix} x_i & y_i & 1 \end{bmatrix} \\ &= T_\theta^h(G) \\ &= \begin{bmatrix} X_i & Y_i & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ dx & 0 & 1 \end{bmatrix} \begin{bmatrix} sx & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \\ &= \begin{bmatrix} X_i & Y_i & 1 \end{bmatrix} \begin{bmatrix} sx & 0 & 0 \\ 0 & 1 & 0 \\ dx \cdot sx & 0 & 1 \end{bmatrix} \end{aligned} \tag{1}$$

In the equation (1), dx is the translation value in the X direction, and sx is the scaling value in the X direction. Parameters in the same horizontal line share the same value. In the horizontal STN, the translation and scaling values in the Y direction are set to 0.

Similarly for the vertical STN, $T_\theta^v(G)$ is

$$\begin{aligned} \begin{bmatrix} x_i & y_i & 1 \end{bmatrix} &= T_\theta^v(G) \\ &= \begin{bmatrix} X_i & Y_i & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & sy & 0 \\ 0 & dy \cdot sy & 0 \end{bmatrix} \end{aligned} \tag{2}$$

Here sx and dy are the translation value and the scaling value in the Y direction respectively. After the parameterised sampling grid, the sampler is used to produce the output feature map FO. This can be written as

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

Here equation (3) 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 the FI and (x_i, y_i) is calculated from formula above. The same operation will be performed in all channels, so channels is omitted in the equation. The sampler in the horizontal STN is the same as one in the vertical STN. The normalization is carried out in X and Y coordinates, such that $0 \leq x_i, y_i \leq 1$ in FI.

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

$$\frac{\partial FO_i}{\partial FI_{hw}} = \sum_{h=1}^{H} \sum_{w=1}^{W} (1 - |x_i - h|)(1 - |y_i - w|) \tag{4}$$

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

$$\frac{\partial FO_i}{\partial y_i} = \sum_{h=1}^{H} \sum_{w=1}^{W} FI_{hw}(1 - |x_i - w|)k_y, k_y = \begin{cases} -1, if\ y_i \geq w \\ 1, if\ y_i < w \end{cases} \tag{6}$$

III. ADVERSARIAL NETWORK

With designed STN, 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 STN. Since the network will be trained in order to decrease the classification error, STN will generate examples which are easy for detectors to classify. In fact, the original upright pedestrians is easiest to recognize. In the other words, the designed STN fails to work in reality.

In this paper, adversarial learning method is used to train the entire network. It is hypothesized that STN should generate examples which are hard for detectors to recognize, and these examples are more important for detectors compared with normal ones. In particular, STN is trained in order to increase the classification error and detectors are trained in the opposite direction.

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

$$L_D = \sum_{i=1}^{N} L_{cls}(D_c(P^i), C^i) + \sum_{i=1}^{N} C^i L_{reg}(D_l(P^i), L^i) \quad (7)$$

In equation (7), L_{cls} is the loss function of classification and L_{reg} is the loss function of bounding box regression. P^i is one of proposes with order i and there are N proposes in total. The original detector network is represented as $D(P)$, including $D_c(P)$ for calculating the class $D_l(P)$ for calculating the location. C^i is the ground-truth class for proposal P^i and L^i is the ground-truth location for it. Note that, if proposal P^i is defined as background, bounding box regression will not be calculated because C^i is 0.

Similarly, the loss function of adversarial network can be defined as:

$$L_S = -\sum_{i=1}^{N} L_{cls}(D_c(S(P^i)), C^i) \quad (8)$$

In equation (8), STN is represented as $S(P)$. $D(S(P))$ is the entire detectors where STN embedded. Note that bounding box regression is omitted in the adversarial network. Minus sign means that STN has the opposite direction of optimization compared with original detector, which is the key idea of adversarial learning.

Concretely, if examples generated by STN is hard for the detectors 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 STN is easy for detectors to recognize, the adversarial network will get a high loss and the detector will get a loss one.

Training method

Motivated by the Faster R-CNN [14], stage-wise training method is applied in this paper. Here training process is divided into three steps.

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

Note that the designed STN and adversarial learning is only applied during training process. And all the images used in the training process are normal images. On the other words, fish-eye images are unnecessary, which is one of the contributions in this work.

Combining with Universal Detectors

It is convenient for universal detectors to combing with designed STN, forming a complete adversarial network. Here three state-of-the-art object detector are used as examples, including Faster R-CNN [14], MS-CNN [15] and SSD [16]. STN can be inserted into Faster R-CNN and MS-CNN before ROI pooling layer. Since MS-CNN has multiple branches, multiple STNs are used and they do not share values. SSD conducts object classification and bounding box regression directly, without ROI pooling layer. So STN is inserted after trunk CNN layers.

Implementation Details

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

It is hypothesized 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) and equation (2). Therefore, the real outputs of localization network in STN 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 process, which is similar to $dropout$ process. In particular, only 30% samples are send into STN, which aims that the detector still has s sense of the normal upright pedestrians through adversarial learning.

IV. EXPERIMENTS

Evaluation Methods and Experimental Settings

Proposed approach is evaluated using three deep learning based detectors, including Faster R-CNN [14], MS-CNN [15] and SSD [16]. Experiments are based on ETH [2] and KITTI [3] pedestrian datasets. Our experiments are all based on VGG-16 model [17], which is pre-trained on the ILSVRC CLS-LOC dataset [18].

Since there is no available fish-eye image dataset, an imaging model conversion algorithm from [19] 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 [19] for more technical details.

The evaluation method used in experiments refers to the KITTI standard, please refer to [3] 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.

A. Experiments on ETH

According to [19], ETH dataset is convert to corresponding fish-eye dataset. The 5361 images are divided into two parts, including 4000 images for training and 1361 images for testing.

Fig. 5 (a) shows some detection results on the converted fish-eye images based on ETH. Fig. 3 is the comparison

TABLE I

RESULTS BASED ON ETH FISH-EYE DATASET. **HSTN** AND **VSTN** ARE THE HORIZONTAL STN AND THE VERTICAL STN. **CONCAT** MEANS COMBINING THE INPUT AND OUTPUT OF HORIZONTAL STN. **AD** MEANS USING THE ADVERSARIAL LEARNING IN THE TRAINING PROCESS AND **NON** MEANS A NON-ADVERSARIAL TRAINING PROCESS.

Training Process			Testing Process	
Detector		Training Method	mAP(%)	Time($ms/frame$)
Faster R-CNN		NON	68.77	109
	+HSTN	AD	70.23	109
	+VSTN	AD	69.27	109
	+HSTN +VSTN	AD	72.63	109
	+HSTN +VSTN +Concat	NON	68.92	109
	+HSTN +VSTN +Concat	AD	**74.66**	109
MS-CNN		NON	71.79	136
	+HSTN	AD	74.98	136
	+VSTN	AD	73.21	136
	+HSTN +VSTN +Concat	NON	71.80	136
	+HSTN +VSTN +Concat	AD	**79.41**	136
SSD		NON	67.89	51
	+HSTN	AD	70.68	51
	+VSTN	AD	69.93	51
	+HSTN +VSTN +Concat	NON	**67.98**	51
	+HSTN +VSTN +Concat	AD	**72.64**	51

TABLE II

RESULTS BASED ON KITTI FISH-EYE DATASET.

Training Process			Testing Process			
Detector		Training Method	mAP(%)			Time($ms/frame$)
			easy	moderate	hard	
Faster R-CNN		NON	70.44	60.58	58.97	107
	+HSTN	AD	74.09	63.68	59.80	107
	+VSTN	AD	74.08	62.11	59.74	107
	+HSTN +VSTN	AD	74.67	63.97	60.13	107
	+HSTN +VSTN +Concat	NON	70.41	60.67	58.99	107
	+HSTN +VSTN +Concat	AD	**76.72**	**64.13**	**60.87**	107
MS-CNN		NON	71.37	67.29	60.42	186
	+HSTN	AD	74.87	70.33	60.99	186
	+VSTN	AD	73.06	68.34	60.12	186
	+HSTN +VSTN +Concat	NON	71.29	68.09	60.33	186
	+HSTN +VSTN +Concat	AD	**77.51**	**71.24**	**62.73**	186
SSD		NON	65.93	59.76	51.49	68
	+HSTN	AD	71.34	62.71	56.83	68
	+VSTN	AD	69.58	61.20	54.50	68
	+HSTN +VSTN +Concat	NON	65.30	59.77	51.69	68
	+HSTN +VSTN +Concat	AD	**72.83**	**63.98**	**59.28**	68

original three detectors and new detectors with STN embedded. The new detectors are trained using adversarial method. From the figure, designed STN and adversarial learning improve the accuracy of pedestrian detection obviously.

Detailed results are shown in the Table I. In Table I, **HSTN** and **VSTN** are the horizontal STN and the vertical STN. **Concat** means combining the input and output of horizontal STN, which is detailed in Fig. 2. **AD** means using the adversarial learning in the training process and **NON** means a non-adversarial training process. Note that the designed STN is only used for training. With designed STN and adversarial learning strategy, the three detectors achieve 6.09% improvement in mAP averagely compared with original detectors.

B. Experiments on KITTI

Fish-eye image dataset based on KITTI [3] can be generated similarly. Note that since we only have the annotated files of training set of KITTI [3] dataset, the whole training part of KITTI [3] dataset is divided into two parts, including 3682 images for training and 3799 images for testing. Fig. 5 (b) shows some results on the converted fish-eye images based on KITTI. Fig. 4 is the similar results with Fig. 3, but it is based on KITTI [3] dataset. Note that moderate level is used when drawing the PR-curve graph.

Detailed results are shown in the Table II. With designed STN and adversarial learning strategy, the three detectors achieve 3.91% (moderate level) improvement in mAP averagely compared with original detectors.

Analysis of Two STNs

Different combinations are tried in our experiments to

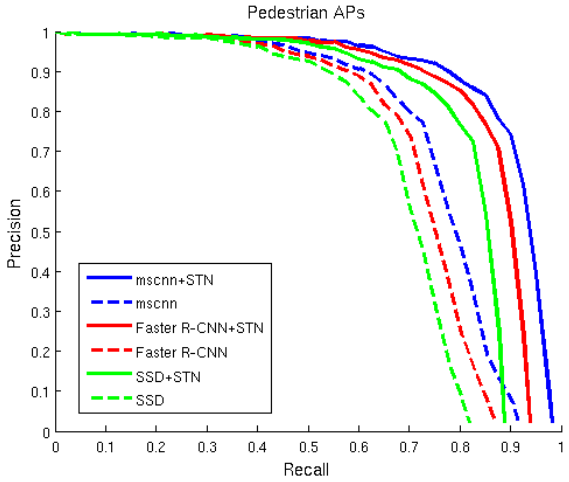

Fig. 3. Comparison between original detectors and new detectors with STN embedded. The result is based on ETH fish-eye dataset.

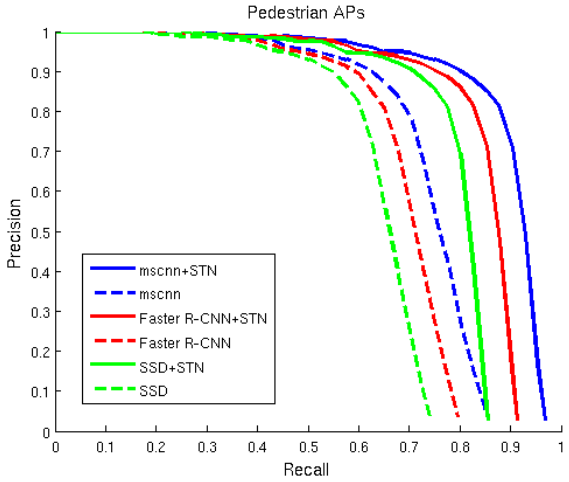

Fig. 4. Comparison between original detectors and new detectors with STN embedded. The result is based on KITTI fish-eye dataset.

prove validity of two STNs. The detailed results are shown in the Table I and II. With concatenation process, Faster R-CNN achieves 1.095% improvement in mAP. 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.

V. CONCLUSIONS

In this paper, the spatial transformer network is designed to generate pedestrian features in fish-eye images. Using this strategy, pedestrian detectors can be trained using normal pedestrian datasets, solving the problem of lack of fish-eye image datasets. Moreover, adversarial learning method is proposed to train the network. Using adversarial learning, STN designed can generate examples that are hard for detectors to classify. The entire adversarial network is more efficient and purposeful and improve the accuracy of pedestrian detection in fish-eye images.

REFERENCES

[1] B. Zhang, V. Appia, I. Pekkucuksen, Y. Liu, A. Umit Batur, P. Shastry, S. Liu, S. Sivasankaran, and K. Chitnis, "A surround view camera solution for embedded systems," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2014, pp. 662–667.

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

[3] 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.

[4] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft coco: Common objects in context," in *European conference on computer vision*. Springer, 2014, pp. 740–755.

[5] 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.

[6] 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.

[7] 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.

[8] 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.

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

[10] 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.

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

[12] 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.

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

[14] 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.

[15] 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.

[16] 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.

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

[18] 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.

[19] 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.

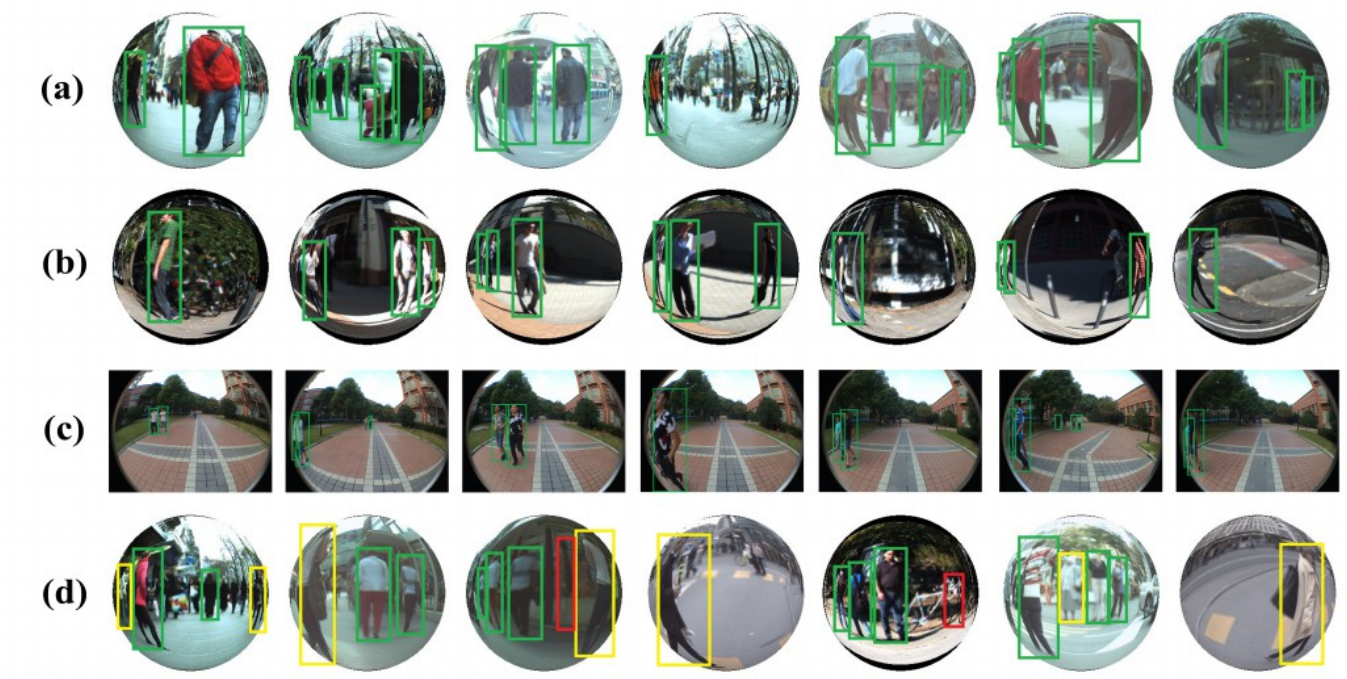

Fig. 5. Some results on the converted fish-eye images. (a): Converted ETH dataset. (b): Converted KITTI dataset. (c): Real fish-eye images. (d): False positives and false negatives. In ETH fish-eye dataset, the training part contains 4000 images and testing part contains 1361 images. In KITTI dataset, 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.