SELF-BOOTSTRAPPING PEDESTRIAN DETECTION IN DOWNWARD-VIEWING FISHEYE CAMERAS USING PSEUDO-LABELING

Kaishi Gao[1], *Qun Niu*[1],*Haoquan You*[2], *Chengying Gao*[1,*]

[1]School of Data and Computer Science, Sun Yat-sen University, China
[2]Winner Technology Co., Inc., China
{gaoksh,niuqun}@mail2.sysu.edu.cn,youhq@winnerinf.com,mcsgcy@mail.sysu.edu.cn

ABSTRACT

Downward-viewing fisheye cameras have attracted much attention in surveillance systems due to the wide coverage and less occlusion. However, pedestrian detection in downward-viewing fisheye cameras remains an open problem due to a lack of large-scale labeled dataset. Furthermore, it's time-consuming and labor-intensive to label a downward-viewing fisheye dataset manually. To address this, we propose a self-bootstrapping pedestrian detection method, which automatically pseudo-labels downward-viewing fisheye images by making full use of *spatial* and *temporal* consistency of pedestrians in the cameras to improve the accuracy of pedestrian detection. We segment the downward-viewing fisheye images into two regions and propose the pseudo-labeling methods for them progressively: a cyclic fine-tuned detector for the oblique region and a visual tracking method for the vertical region. Combining the pseudo-labels from two regions, we fine-tune the network for better accuracy. Experimental results show that the proposed approach reduces time consumption by about 95% compared with the labor-intensive manual labeling while it still reaches competitive and comparable Average Precision (AP).

Index Terms— Pedestrian detection, automatic labeling, downward-viewing fisheye camera

1. INTRODUCTION

Fisheye cameras have been widely used in real-world surveillance systems and a variety of computer vision tasks, because they provide a wide field of view (FOV). Furthermore, downward-viewing fisheye cameras also benefit from less occlusion among objects. Recently, many pedestrian detection networks are based on fully-supervised detectors (FSDs). They take advantages of large-scale labeled pedestrian datasets and have ability to detect pedestrians which are visually similar to the training set. However, different from existing datasets, visual appearances of pedestrians vary drastically in cameras because of different camera angle, as shown in Figure 1.

(a)

(b)

Fig. 1. (a) Pedestrians from downward-viewing fisheye cameras. (b) Pedestrians from existing dataset.

Former works in literatures related to pedestrian detection in downward-viewing fisheye cameras are usually based on hand-crafted features [1][2] and foreground information [3][4]. These methods would be limited by the foreground information extraction and lead to approximation errors, which are common with hand-crafted features. Recently, a series of labeled downward-viewing images are generated by transforming existing datasets [5]. However, the data gap between downward-viewing fisheye pedestrians and existing datasets remains unsolved. To address this, in real-world surveillance systems, some works resort to non-technical solutions, such as manual labeling. However, it is usually time-consuming, labor-intensive and not scalable to large datasets. Su et al. [6] reports that it usually takes at least 35s to draw a bounding box manually, adding up to 180 hours with a dataset with 10,000 images. Other works devote to detecting pedestrians with weakly supervised object detectors (WSODs) [7][8], which still suffer from performance degradation with drastic appearance changes of pedestrians. Therefore, the wide application of fisheye cameras is largely hindered by a lack of high-quality and labeled training data with pedestrians.

In this paper, we propose a self-bootstrapping method for pedestrian detection in downward-viewing fisheye cameras, which progressively labels images without intervention after initialization, to address the problem. We propose to segment downward-viewing fisheye images into two regions: an

*The corresponding author is Chengying Gao.

978-1-7281-1331-9/20/$31.00

oblique region and a vertical region, according to the angle θ between the incoming ray and the camera's optical axis. An oblique region is the one with large θ values, in which the body of a pedestrian is visible and visually similar to pedestrians in perspective dataset. While in the vertical region, which refers to the region with small θ values, only heads and shoulders of pedestrians are visible. Then we conduct further pseudo-labeling method separately in these regions, as shown in Figure 2. The objective of pseudo-labeling is to obtain a pseudo instance-level annotations for images without labor-intensive manual labeling.

In the oblique region, we detect pedestrians with an off-the-shelf pedestrian detection network, and fine-tune the network on detected pedestrians' proposals afterwards. While in the vertical region, we track pedestrians across frames on the basis of detected pedestrians in the oblique region. Combining the pseudo-labels from two regions, we obtain pseudo-labels of the entire images and create a downward-viewing fisheye dataset. Then we fine-tune FSDs on the dataset for better accuracy.

To summarize, we make the following major contributions:

- We propose a self-bootstrapping method for pedestrian detection in downward-viewing fisheye cameras. Without the need of manual annotation, our approach achieves high applicability.
- We segment downward-viewing fisheye images into two regions and propose a temporal continuity-based labeling algorithm to label pedestrians effectively.
- Experimental results demonstrate that the proposed approach reduces time consumption by 95% and reaches competitive Average Precision (AP) of 84.8%, which outperforms the state-of-the-art methods.

In addition to pedestrian detection, the proposed method can be extended to other visual tasks in downward-viewing fisheye cameras as well.

2. RELATED WORK

Downward-viewing fisheye cameras have attracted more and more attention in recent years since they provide a wide field of view and can avoid the occlusion problem. While there are outstanding works [9][10][11] for pedestrian detection, they are mainly based on existing perspective datasets such as MSCOCO [12]. The study of pedestrian detection in downward-viewing cameras is limited due to the lack of labeled downward-viewing fisheye images. Former works make use of foreground information [3][4] and are laborious to generalize to other cameras. Krams et al. [2] unwrap the aggregated channel features of pedestrians in downward-viewing fisheye cameras to perspective-like feature maps so to detect pedestrians with existing classifier. However, they are prone to ignore pedestrians with only heads and shoulders

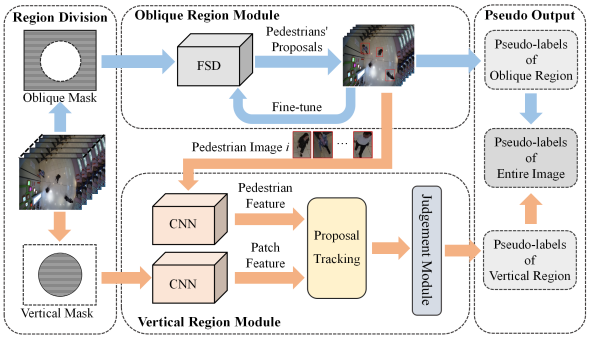

Fig. 2. The overall workflow of our proposed method.

as they are significantly different from those in the training dataset. Wang et al. [1] propose a HOG+SVM based detector trained with both the images and the foreground masks for a single camera. Although its high accuracy in specific experimental settings, the method is still dependent on manually labeled images to train the SVM classifier. Chen et al. [5] apply a CNN-based method to downward-viewing images without manually labeled image. They generate labeled downward-viewing fisheye images by transforming existing oblique-viewing datasets so to train FSDs. However, it is hard to generate all kinds of downward-viewing pedestrians and the data gap remains unsolved.

A large body of recent arts [7][8][13] leverage weakly supervised learning to learn a model with limited training data, which intends to detect objects with only image-level labels. Multiple Instance Learning (MIL) has been widely deployed in weakly supervised object detection methods. WSDDN [7] proposes a smoothed version of MIL that softly labels object proposals instead of choosing the one with the highest score. C-MIL [8] integrates a continuation optimization method into MIL to create a continuation multiple instance learning. But these methods still have large performance gap with fully-supervised detectors. Recently, Inoue et al. [13] propose a high-performance cross-domain weakly supervised object detection pipeline, which fine-tunes FSDs on generated labeled images. A former work [5] fine-tunes FSDs on images after perspective transformation to detect pedestrian in aerial images, which also shows the advantage of fine-tuning methods.

3. APPROACH

We motivate the region division and define the oblique and vertical regions in Section 3.1. Afterwards, we elaborate the cyclic fine-tuned detector for pseudo-labeling method in the oblique region (Section 3.2) and the tracking-based pseudo-labeling method in the vertical region (Section 3.3).

3.1. Region Division

Prior pedestrian detection works usually directly generate the detection models with existing labeled datasets as they prac-

tically cover all kinds of pedestrians in the target images. However, due to the wide field of view and different viewing angles, visual appearances of pedestrians vary a lot in downward-viewing fisheye images. As a result, there are some pedestrians, which only heads and shoulders are visible, never or seldom appears in existing labeled datasets. Additionally, these pedestrians are more likely to be observed in the region just below the cameras. Consequently, in this work, we propose to segment downward-viewing fisheye images into two regions based on visual appearances of pedestrians in downward-viewing fisheye cameras. We conduct pseudo-labeling of pedestrians in two regions separately.

We start by presenting the camera models. Given a standard perspective camera, the projection model can be written as [14]:

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

where f indicates the focal length, r is the projection distance between the principal point and the pixels in the image and θ measures the angle between the incoming ray and the camera's optical axis.

However, for fisheye cameras, they don't obey this rule due to the unique structural design of lens. Generally, a quintic polynomial is normally used to describe the fisheye camera projection model [14]:

$$r(\theta) = \sum_{k=1}^{n} k_i \theta^{2i-1}, n = 5. \tag{2}$$

As for downward-viewing fisheye images, the principal point is the center of the image. The distortion parameters k_i can be computed by using calibration points in the images or may be directly provided by the producer. Figure 3 shows the model of the downward-viewing fisheye camera.

Visual appearances of pedestrians change with different θ values in downward-viewing fisheye cameras. When the θ is large, pedestrians can be detected by FSDs with low confirm threshold. Otherwise, we are only able to observe head or upper body of a pedestrian, which is challenging for off-the-shelf FSDs. The reason is that when θ is large, we can observe distorted oblique-viewing bodies of pedestrians, which are similar to oblique-viewing pedestrians. However, when θ is small, only heads and shoulders of pedestrians are visible.

In accordance with the characteristics of pedestrians with different θ, we discriminate each pixel of downward-viewing fisheye images into two regions, an oblique region and a vertical region, according to a certain angle α. R is the Euclidean distance from the pixel to the principal point in images. The regions are defined as following:

$$r(\alpha) < R \leq r_{max}, \tag{3}$$

the corresponding pixel is in oblique region.

$$r(0) \leq R \leq r(\alpha), \tag{4}$$

the corresponding pixel is in the vertical region. Furthermore, we propose two masks for two regions, an oblique mask m_o and a vertical mask m_v separately.

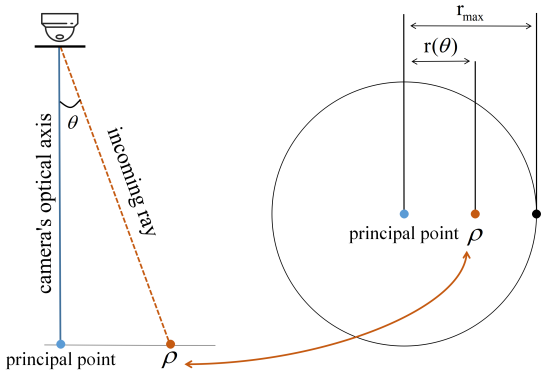

Fig. 3. The model of the downward-viewing fisheye camera.

The choice of α influences the final pedestrian detection performance. We study the influences of α on the detection accuracy in Section 4.4.

3.2. Pseudo-Labeling in the Oblique Region

In oblique region, pedestrians outside the region determined by α are able to be detected by FSDs trained on existing oblique-viewing dataset. However, the confirm scores of some pedestrians are low as the dataset bias is small but still exist. If we directly decrease the confirm threshold of the FSDs, detected proposals may be erroneous. So we fine-tune the FSDs on downward-viewing fisheye images and the detected pedestrian proposals with higher confirm scores to overcome the dataset bias.

Formally, we train an FSD G only on existing labeled datasets. We denote RGB images from downward-viewing fisheye cameras by $x \in R^{H \times W \times 3}$, where H and W are the height and width of the image, respectively. Pedestrians in images are defined as $P_i \in \{(x_i, y_i, w_i, h_i) | i = 1, 2, 3....N\}$, where (x, y) is the center point of a pedestrian's bounding box and w is weight and h is height. The oblique region mask m_o of each image is made up of concentric circles with radius $r(\theta_o)$, where $\alpha \leq \theta_o \leq \theta_{max}$, as shown in Figure 2. We compute the pseudo-labels $\widetilde{P}_i$ with confirm threshold δ for each image as:

$$\left\{\widetilde{P}_i\right\} = G(x \otimes m_o, \delta), i = 1, 2, 3...N, \tag{5}$$

where $\otimes$ is the Hadamard product, i is the index of a pedestrian. Then we fine-tune the FSD on the pseudo-labels $\widetilde{P}_i$ and images. Based on the number of training samples, we fine-tune the model cyclically until we are able to achieve sufficient accuracy.

3.3. Pseudo-Labeling in the Vertical Region

Visual differences between pedestrians in vertical region and in existing labeled datasets are tremendous and the pedestrians are hard to be detected by off-the-shelf FSDs. Notice

that the original data obtained by downward-viewing fisheye cameras are videos, which are continuous in time and entire in space. Furthermore, inspired by the multi-object tracking method in videos [15], we combine pedestrian detection and visual tracking for pedestrian pseudo-labeling in the vertical region. The core concept of our method is to find the target pedestrian, which is detected in the oblique region, based on temporal continuity in the vertical region. The vertical mask m_v covers the vertical region and is a bit larger than it. The mask consists of concentric circles with radius $r(\theta_v)$, where $\theta_v < \beta$ and $\alpha < \beta$. In addition, we define the corresponding area where $\alpha < \theta < \beta$ as detection area. Our method can be subdivided into 3 steps: feature extraction, proposal tracking and judgement module.

Feature extraction: Firstly, we use a fine-tuned FSD $\hat{G}$ to detect pedestrians in the detection area in frame T^t. We pick up the pedestrian proposals $P_i^t \in \{(x_i^t, y_i^t, w_i^t, h_i^t) | i = 1, 2...N\}$ and the corresponding image I_i of pedestrians i. Then, we extract the feature of each pedestrian image by a CNN F, as:

$$F_{detect} = F(I_i). \tag{6}$$

For the next frame T^{t+1}, we extract the feature of the image of frame T^{t+1}, denoted as $F\left(T^{t+1}\right)$. Afterwards, we use a sliding window $Win = (w_{win}, h_{win})$ within the vertical mask m_v to scan the feature, where (x_{win}, y_{win}) is the center point of the window. We denote each image patch cropped from T^{t+1} on Win by $C\left(T^{t+1}, Win\right)$ and the corresponding feature of each tracking patch as:

$$F_{track} = F\left(C\left(T^{t+1}, Win\right)\right). \tag{7}$$

Proposal tracking: We compute the matching score s of each pedestrian image and tracking patch by proposal cross-correlation:

$$S_i^{Win}(F_{detect}, F_{track}) = F_{detect} \star F_{track}. \tag{8}$$

Since the walking speed of pedestrians is usually limited, the displacement of a pedestrian between two consecutive frames is usually small. So the matching score s is relevant to the distance from the center point (x_{t+1}, y_{t+1}) of the tracking patch in frame T^{t+1} to the center point (x_t, y_t) of the tracked pedestrian in frame T^t. We further combine the matching score s with the displacement distance D. Usually, the displacement D is smaller than a threshold N. When $D > N$, the matching score s declines as D gets larger:

$$S_i^{Win} = \begin{cases} F_{detect} \star F_{track}, & D \leq N; \\ \frac{1}{|D-N|}\left(F_{detect} \star F_{track}\right), & D > N, \end{cases} \tag{9}$$

$$D = \sqrt{(x_{t+1} - x_t)^2 + (x_{t+1} - x_t)^2}. \tag{10}$$

We select the best matching proposal with the highest score s as the bounding box of pedestrian i. Afterward, we update the bounding box as the position of pedestrian i in frame T^{t+1}. Nevertheless, we keep image I_i of pedestrian i detected in the detection area.

Judgement module: We add a judgement module to determine if a pedestrian has been out of the vertical region. We keep detecting pedestrian in vertical mask m_v of each frame and pedestrians would be detected when they walk closed to the detection area.

We compute the IoU between each detected pedestrian proposal P_{detect} and the tracking proposal P_{track} of pedestrian i. If the IoU is larger than 0.5 and the distance D between the tracking proposal P_{track} and the original proposal in the detection area of pedestrian i is lager than $r(\theta_v)$, we consider pedestrian i has walked through the vertical region and will not compute it in the following frames. What's more, we further limit the number of tracking frames for one pedestrian to 30, according to our training dataset.

Ultimately, we combine the pseudo-labels from two regions and obtain pseudo-labels of the entire downward-viewing fisheye images. We construct a downward-viewing fisheye dataset with pseudo-labeled images and fine-tune FSDs on this dataset.

Table 1. Detection AP results on the downward-viewing fisheye dataset compare with Baseline.

Method	$AP\%$	$AP_{50}\%$	$AP_{75}\%$	time cost
Baseline	90.1	86.6	62.2	two weeks
Ours	84.8	80.8	39.4	**10 hours**

4. EXPERIMENT

4.1. Dataset

We provide two datasets for following experiments: a validation set and a training set for pseudo-labeling and fine-tuning. We create a hand-crafted downward-viewing fisheye image dataset with full instance-level annotations as a validation set, which contains 1700 images and about 4500 pedestrians in total. The image size is 640×360. The number of pedestrians ranges from 1 to 5 in each image.

As for the training dataset, we select a number of videos. Each video is about 1 hour in length and the resolution is 1080p. We pick up videos with pedestrians walking through and cut them with 3s interval. At last we have 14400 images which are continuous in time and entire in space.

4.2. Implementation Details

Fisheye camera model: In this work, we use a spherical fisheye camera. Its focal configurations is 2.1mm (FOV 160°). The installation height of the camera ranges from 3.1m to 3.5m. The height in our experiment is 3.5m.

Oblique region module: We use VGG-16 and a multi-level feature pyramid network to extract features from the input images as mentioned in M2Det [16], because experiments demonstrate that multi-level features are beneficial to detection results. The input size of images is 512x512 and the batch

Table 2. Detection results on the downward-viewing fisheye dataset compare with state-of-the-art methods.

Method		$AP\%$
FSD	Yolo v3 [18]	31.7
	M2Det [16]	43.2
WSOD	WSDDN [7]	13.2
	C-MIL [8]	21.4
PL	Transformed[5]	55.2
	Ours-O	**61.3**
	Ours	**84.8**

size is 8. We replace TUM module by a conv-layer with 1×1 filter to generate multi-scale features since the size of pedestrians in downward-viewing fisheye camera keeps almost unchanged compared with oblique-viewing cameras. We train the network on MSCOCO [12]. In the fine-tuning step, the confirm threshold δ is 0.6. The confirm threshold here is chosen as a matter of experience and could be changed according to different downward-viewing fisheye cameras.

Vertical region module: We follow the structure of SiamFC architecture [17] for tracking, as its performance is less dependent on the training dataset. Instead of AlexNet, we attach a pre-trained ResNet-50 to extract features of pedestrian images and input frames. And we compare the matching scores with sliding windows of 5 anchors. Our training step is the same as SiamFC [17]. We apply the tracking method to each recorded pedestrian independently.

4.3. Comparative Results

We compare our method with a labor-intensive manual labeling method and state-of-the-art methods. Table 1 shows the comparison with the labor-intensive manual labeling baseline method. All models were trained and tested on a NVIDIA GeForce GTX 2080Ti but we haven't made full use of the 11G memory. It takes two volunteers around two weeks to label 14400 images manually. Experimental results demonstrate the effectiveness of the proposed method. IoU here presents the threshold to compute the AP. Our proposed method has achieved comparable AP (84.8%, IoU=0.05) compared with the baseline approach (90%, IoU=0.05). What's more, our method still achieves a competitive recall rate of 80.8% when IoU=0.5. But when IoU=0.75, AP drops to 39.4%. This is because the pseudo-labels are extracted automatically and it may lead to offset. However, our method significantly reduces time consumption by about 95% and saves human resources, which is exceedingly important in real-world application.

Table 2 shows the results of our method against the FSDs, WSODs and the state-of-art pedestrian detection method for downward-viewing aerial images [5], which shares similar data gap problem as downward-viewing fisheye cameras. We present the performance of FSDs without fine-tuning with

Fig. 4. Downward-viewing fisheye pedestrian detection results using our self-bootstrapping method.

Yolo v3 [18] and M2Det [16] with parameters of the network suggested by the authors. As for WSODs, we implement WSDDN [7] and C-MIL [8], which only require image-level labels. We fine-tune them using our dataset combining to achieve sufficient accuracy. In Transformed [5], they fine-tunes Yolo v3 [18] on VOC2007 [19] after perspective transformation to generate pedestrians with minor θ in downward-viewing aerial images. Ours-O is Yolo v3 [18] only fine-tuned on oblique region pseudo-labels.

The result shows fine-tuned FSD on transformed oblique-viewing images can promote the pedestrian detection performance of downward-viewing fisheye cameras (as 31.7% AP of Yolo v3 [18] and 55.2% of Transformed [5]). However, the dataset generated by perspective transformation is not sufficient when pedestrians are in the vertical region. Our method achieves better performance by pseudo-labeling in two different regions.

4.4. Influences of α

The choice of α has influences on the performance of pedestrian detection as it directly affects the quality of pseudo-labels. When α is too large, which means the vertical region is large, the visual tracking method of vertical region results in larger offset of bounding boxes, as shown in Figure 5, which will lead to lower AP. On the other hand, when α is small, it is difficult to detect pedestrians after cyclically fine-tuning. So the selection of α should well balance the recall rate of oblique region and the tracking precision of vertical region. Most of the time, the value of α can be estimated by the pedestrians detection results of off-the-shelf FSDs.

4.5. Studies on Camera Generalization

We have conducted another experiment to evaluate how the proposed method transfers to a camera in new environments since pedestrians' feature may vary in different cameras due to the impact of hardware and height. In Experiment A, we use a pre-trained model in our previous experiment directly. In Experiment B, we fine-tune a pre-trained Yolo v3 [18] with pseudo-labeled images in the new camera. Table 4 compares

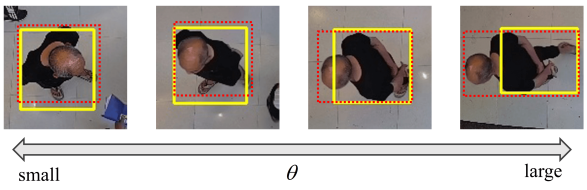

Fig. 5. Offsets occur when θ is large.

Table 3. Evaluation results of generalization studies.

Method	Recall%
Experiment A	64.0
Experiment B	87.0

the recall of 100 random pedestrians.

As shown in the table, Experiment B performs better since pedestrians' feature may vary in different cameras due to the impact of hardware and height. However, this also shows the superiority of our self-bootstrapping pseudo-labeling method for it can obtain pseudo-labeled images automatically while others have to pay an army of time and labor to achieve similar results.

5. CONCLUSION

In this paper, we propose an efficient and effective pedestrian detection method for downward-viewing fisheye cameras by progressive tuning of pedestrian detection networks. We segment downward-viewing fisheye images into two regions and provide a pseudo-labeling method for each. We augment our pseudo-labels by detecting pedestrians in the oblique region. By tracking pedestrians in the vertical region, we associate them with detected pedestrians in the oblique region and augment our pseudo-labels with pedestrians detected in these regions. By fine-tuning FSD on our pseudo-labels and images, the proposed method is more effective to achieve high detection performance. Our method achieves comparable accuracy compared with the network trained on hand-crafted dataset.

6. ACKNOWLEDGEMENT

This work was supported by National Key Research and Development Plan in China (Grand No. 2018YFC0830500), National Natural Science Foundation of China (Grand No. 61972433), Fundamental Research Funds for the Central Universities (Grand No. 19lgjc11), Natural Science Foundation of Guangdong Province, China (Grant No. 2019A1515011075).

7. REFERENCES

[1] T. Wang, C. Chang, and Y. Wu, "Template-based people detection using a single downward-viewing fisheye camera," in *ISPACS*, 2017.

[2] O. Krams and N. Kiryati, "People detection in top-view fisheye imaging," in *AVSS*, 2017.

[3] V. T. Nguyen, T. B. Nguyen, and S. Chung, "Convnets and AGMM based real-time human detection under fisheye camera for embedded surveillance," in *ICTC*, 2016.

[4] L. Meinel, M. Findeisen, M. Heß, A. Apitzsch, and G. Hirtz, "Automated real-time surveillance for ambient assisted living using an omnidirectional camera," in *ICCE*, 2014.

[5] H. Chen, C. Liu, and W. Tsai, "Data augmentation for cnn-based people detection in aerial images," in *ICME Workshops*, 2018.

[6] H. Su, J. Deng, and L. Fei-Fei, "Crowdsourcing annotations for visual object detection," in *AAAI Workshops*, 2012.

[7] H. Bilen and A. Vedaldi, "Weakly supervised deep detection networks," in *CVPR*, 2016.

[8] F. Wan, C. Liu, W. Ke, X. Ji, J. Jiao, and Q. Ye, "C-MIL: continuation multiple instance learning for weakly supervised object detection," in *CVPR*, 2019.

[9] C.Zhou and J. Yuan, "Bi-box regression for pedestrian detection and occlusion estimation," in *ECCV*, 2018.

[10] S. Liu, D. Huang, and Y. Wang, "Adaptive NMS: refining pedestrian detection in a crowd," in *CVPR*, 2019.

[11] W. Liu, S. Liao, W. Ren, W. Hu, and Y. Yu, "High-level semantic feature detection: A new perspective for pedestrian detection," in *CVPR*, 2019.

[12] T. Lin, M. Maire, S. J. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft COCO: common objects in context," in *ECCV*, 2014.

[13] N. Inoue, R. Furuta, T. Yamasaki, and K. Aizawa, "Cross-domain weakly-supervised object detection through progressive domain adaptation," in *CVPR*, 2018.

[14] J. Kannala and S. S. Brandt, "A generic camera model and calibration method for conventional, wide-angle, and fish-eye lenses," *PAMI*, vol. 28, no. 8, pp. 1335–1340, 2006.

[15] W. Yang, B. Liu, W. Li, and N. Yu, "Tracking assisted faster video object detection," in *ICME*, 2019.

[16] Q. Zhao, T. Sheng, Y. Wang, Z. Tang, Y.Chen, L. Cai, and H.Ling, "M2det: A single-shot object detector based on multi-level feature pyramid network," in *AAAI*, 2019.

[17] L. Bertinetto, J. Valmadre, J. Henriques, A. Vedaldi, and P. Torr, "Fully-convolutional siamese networks for object tracking," in *ECCV 2016 Workshops*, 2016.

[18] J. Redmon and A. Farhadi, "Yolov3: An incremental improvement," *arXiv preprint, arXiv:1804.02767*, 2018.

[19] M. Everingham, L. V. Gool, C. K. I. Williams, J. M. Winn, and A. Zisserman, "The pascal visual object classes (VOC) challenge," *International Journal of Computer Vision*, vol. 88, no. 2, pp. 303–338, 2010.