

Received December 23, 2020, accepted January 15, 2021, date of publication January 27, 2021, date of current version February 8, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3054938

Real-Time Enumeration of Metro Passenger Volume Using Anchor-Free Object Detection Network on Edge Devices

ZHONGXING ZHENG¹, WEIMING LIU¹, HENG WANG², GUICI FAN³,
AND YUAN DAI¹, (Graduate Student Member, IEEE)

¹School of Civil Engineering and Transportation, South China University of Technology, Guangzhou 510641, China

²Shenzhen Metro Group Company Ltd., Shenzhen 518026, China

³Guangzhou Metro Corporation, Guangzhou 510335, China

Corresponding author: Weiming Liu (mingweiliu@126.com)

This work was supported by the China's 13th Five-Year Key Research and Development Plan for the Safety Guarantee Technology of Urban Rail Transit System under Grant 2016YFB1200402.

ABSTRACT Enumeration of metro passenger volume is essential in providing effective passenger guidance and improving the usage rate of each carriage. However, existing methods cannot provide the accurate number of alighting and boarding passengers at each gate on the platform. For existing visual methods, the occlusion problem seriously affected the results caused by the angle of view. In this study, we introduce a real-time metro passenger volume enumerating algorithm that is simple, effective, and fast enough to run on edge devices mounted above the platform gate. First, we capture videos from the cameras and design an anchor-free object detection network called CircleDet to detect passengers' heads. CircleDet predicts a circle to localize and bound the target instead of traditional bounding box. Then, we apply a simple but effective circle IoU-based method to identify and track passengers in the videos. CircleDet can achieve up to 111 frames per second (FPS) running on NVIDIA RTX 2080 and 7.8 FPS on an NVIDIA Jetson Nano device. The accuracy of enumeration is as high as 97.1% on our own metro object detection (MOD) dataset.

INDEX TERMS Metro passenger volume, edge device, deep learning, computer vision.

I. INTRODUCTION

In cities around the world, the metro train system is one of the most important urban transportation system. The main advantage of metro is the overwhelming payload capacity compared with other means. However, this capacity relies on flexible train dispatching and effective guidance that makes each carriage payload full and balanced on the platforms. As the metro trains become longer and longer, balancing payload of each carriage is of great importance. As a result, passenger volume at each gate is a key factor that is essential to enumerate metro passenger volume in real time.

Over the years, many metro corporations mainly use these common types of enumeration for passenger volume: video surveillance, weighing, RFID reader and turnstile enumeration. However, these methods may suffer from low accuracy and efficiency. Moreover, aforementioned methods cannot precisely provide the number of passengers entering or leav-

ing each carriage and train gate. As a result, they cannot be reliable real-time evidence of passenger guidance and train dispatching.

With the development of sensor and computer technology, the video-sensor-based detector has become popular. Traditional computer vision were applied to the task mentioned but they heavily depended on controlled condition and vulnerable to changes. Early works try to extract global features in the images first, such as texture, gradient, edge features, or local features, such as SIFT [1], HOG [2] and LBP [3]. Then, they directly learn the mapping from an image patch to the count. However, these methods based on hand-craft features not only suffer from low accuracy but also always ignore spatial information.

In recent years, many scholars have studied the possibility of using deep neural networks to solve the detection problem. Many excellent works have been conducted, such as R-CNN [4], Fast RCNN [5], Faster RCNN [6], YOLO series [7]–[9], SSD [10] and CenterNet [11]. Although these methods detect the targets in sight with high accuracy, they


The associate editor coordinating the review of this manuscript and approving it for publication was Yongqiang Zhao .



FIGURE 1. An example image of passengers captured by overhead camera.

still do not work well under a more complex situation such as occlusion, scale variation and rotation of targets.

Some scholars try to use CNN to count crowds at a coarser granular level [12]–[18]. Researchers train CNN to learn multi-scale features from a whole image and directly predict the number of people in a crowd in a supervised or self-supervised manner.

However, the computation consumption of these methods may not be suitable for mobile devices, and balancing accuracy and speed on the edge devices is difficult. Although the latter methods using multi-scale information work well under some situation of occlusion and multi-scale variation, they cannot provide a precise number of passengers under different shooting angles.

In this study, considering the trade-off between accuracy and hardware capacity constrain, we introduce an enumerating paradigm for real-time metro passenger volume. This paradigm is simple, effective, and fast enough to run on edge devices. First, a dataset containing various types of passengers from the real world is developed. The video data are in the videos captured by the cameras mounted on top of a platform screen door (PSD) and aimed at the gap between the metro door and PSD. As a result of this shooting angle, the occlusion problem can be eliminated. Then, we design a new CNN network to detect passengers. Our design is based on the deep layer aggregation (DLA) [19] network structure, and simplify it according to our task. We also apply MBConv block [20] as basic block that helps network run on edge devices. The network takes the RGB image as input and predicts the position and size of the head and body of passengers in an anchor-free manner. In particular, we design the network predicting bounding circles instead of boxes to present the size of the target, so we named it CircleDet. To mark and track passengers in the videos, we propose an algorithm based on the predictions of head, which basically rely on the relationship of the heads measured by circle intersection over union (IoU). Finally, we record every passenger moving track and enumerate the passenger volume in the metro platform.

The contributions of this study can be summarized as follows:

- To the best of our knowledge, this study is the first to adopt anchor-free style detection network to solve the problem of enumerating metro passenger volume.
- CircleDet: We propose a circle representation for human head and body detection in a neural network. To a certain extent, CircleDet reduces computation cost and makes inference faster. The proposed CircleDet network has better rotation consistency in detecting human heads.
- We propose a human marking and tracking algorithm to enumerate passenger volume. It is based on the circle representation prediction on the head.

II. RELATED WORKS

A. CONVOLUTIONAL NEURAL NETWORKS WORK ON EDGE DEVICE

As the edge devices usually needs to be small enough or portable in practical application, the performance encounters difficulty in meeting modern convolutional neural network (CNN) requirements. As a result, improving the resource efficiency of CNN models has been a hot research topic in recent years. There are two types of mainstream approaches: 1) quantizing the weights and activation layers of the baseline CNN model into lower bits format, or pruning some less important layers and filters of in the network to reduce computation, and 2) designing more efficient mobile architectures manually or using network architecture search (NAS): SqueezeNet [21] uses 1×1 convolutions and reduces filters sizes to reduce parameters and computations. MobileNet series [22], [23] apply depthwise separable convolution and redesign the residual architecture to balance efficiency and accuracy. In addition, NAS is applied to train the network itself to find out the best network model architecture under designed constraint condition, such as MnasNet [20], MobileNetV3 [24].

B. ANCHOR-BASED AND ANCHOR-FREE DETECTION NETWORK

Object detection task involves predicting objects location, size and categories. Location and size are usually described as bounding box. To predict bounding box of objects, researches use designed anchors and regress each anchor size and location to fit ground true bounding boxes. Although these methods achieved good performance, they still suffer from problems caused by non-maximum suppression in selecting output bounding box and need well-designed initial anchor scales and ratios in different scenes. To solve the problems mentioned above, several anchor-free detection networks have been proposed recently. They tried different ways to avoid usage of anchor. Using key-points estimation to detect objects: CornerNet [25] detects two bounding box corners as key-points. CenterNet [11] and FCOS [26] predict the center point of the target object in the image. Some networks apply semantic segmentation skill to object detection task [27].

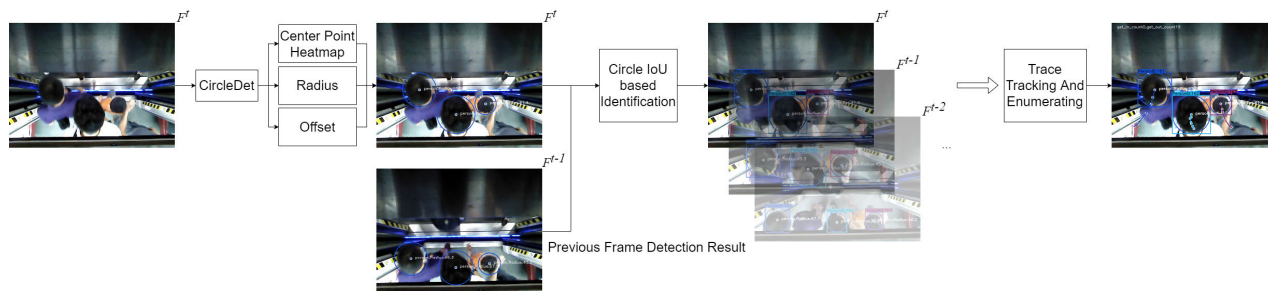


FIGURE 2. The pipeline of enumerating passengers in metro.

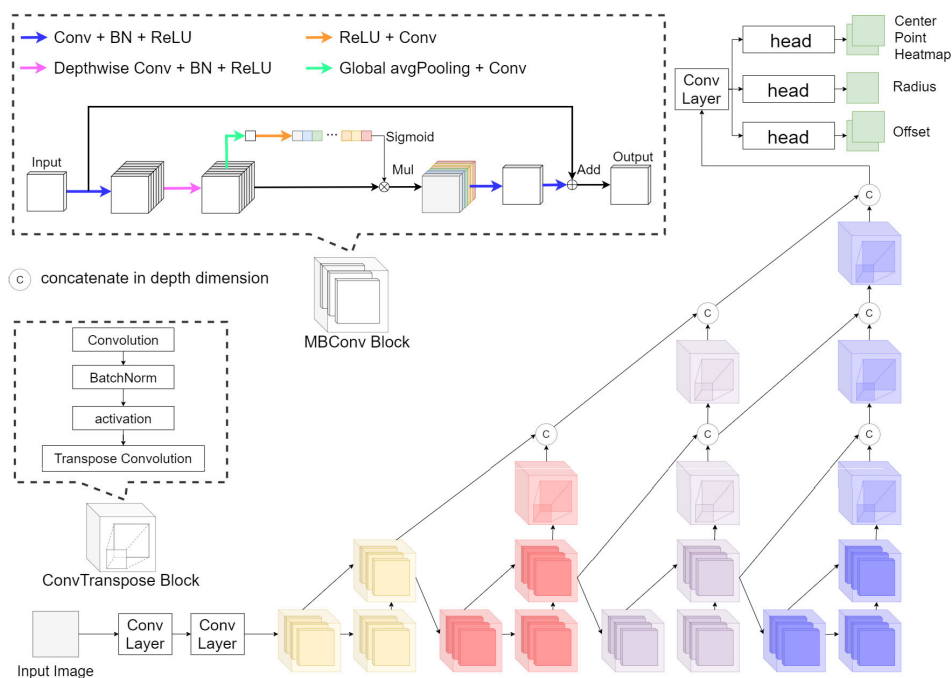


FIGURE 3. CircleDet architecture.

Extracting multiple representative points of objects in image is also proposed [28]–[30].

C. PASSENGER ENUMERATION IN TRANSPORTATION

It is crucial for public transportation to develop an efficient management system. If the passenger can be enumerated in different transportation situations, then the routes and schedules can be effectively improved. Many scholars have exerted effort to meet the needs of enumerating passengers. Hsu *et al.* [31] propose a passenger flow counting model for buses based on deep learning. Using image analysis and shape detection, Grönman *et al.* [32] developed a system to collect statistics about bus passengers. Nakashima *et al.* [33] used GPS, drive recorders and other sensors that are already equipped in the bus to estimate the number of passengers.

III. METHODOLOGY

The whole pipeline of our algorithm is shown in Fig.2. First a frame of video is obtained by CircleDet to detect human

heads. After detection, circle IoU-based identification is applied to result and previous one to track passengers. Finally, when a multiple frame of trace tracking is finished, the number of passengers alighting or boarding can be obtained.

A. CircleDet

1) ANCHOR-FREE BACKBONE

In Fig.3, the backbone network is designed in the anchor-free style. Although CenterNet possesses a combination of high performance and simplicity, it still computational expensive in edge devices. Thus, we simplify the backbone network and reduce the predicting heatmap number by using circle representation. Stages in CircleDet backbone are marked in different colors in Fig.2.

As the camera detecting passengers is at a fixed pitch angle, the size and appearance of the human head in image is relatively fixed. Therefore, multi-scale fusion of feature maps is essential but can be reduced in this scene. In CircleDet, we follow an idea of DLA [19] network to design ours. DLA

can learn to better extract the full spectrum of semantic and spatial information from a network. However, it not only requires too much computation resource and memory of edge devices in the hierarchical deep aggregation (HDA) of DLA, but also the task of detecting passengers does not require extremely deep aggregation of feature maps. As a result, we shrunk the depth of each stage into only 1 in HDA. Furthermore, we also cut down some channels inside the network to make it slimmer and fit for the edge device.

For each stage in the network, we use transpose convolution to upsample feature maps and concatenate the same size output feature maps of each stage. To further fasten the inference speed and balance the average accuracy and speed, we use the MBConv block as basic block in CircleDet. MBConv block is first used in MnasNet [20]. MBConv block is searched by NAS in consideration of balancing accurate and source consumption. MBConv block not only applies residual design and squeeze and excitation (SE) module, but also uses depthwise convolution. These techniques reduce computation and parameters but keep the model accurate at the same time.

2) CIRCLE REPRESENTATION

Commonly, object detection tasks use a bounding box to mark targets in images. Once the center point of the bounding box is determined, bounding box representation requires a network to predict two dimension variables, which are width and height. However, in the passengers enumeration task, targets are human heads which usually appears in round shape at an overhead view. In this case, not only using circle representation need to predict only one dimension variable which is radius, but also has multiple benefits in the detection task of human heads.

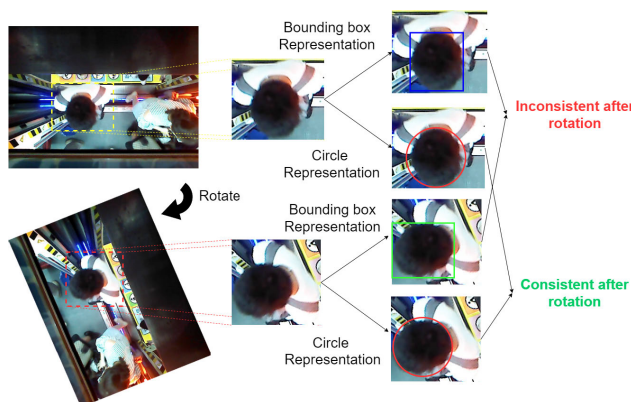


FIGURE 4. Circle representation.

First, as shown in Fig.4, circle representation has only one degree of freedom compared with bounding box representation. Although human heads are round in most cases, bounding box representation is mainly designed for oriented detection approaches. In other words, it is not necessarily optimized for round-shape target detection. Second, circle representation requires less computation cost in inference

time. Third, less predicting outputs mean less difficulty for networks to train and better performance on simple network model. As a result, there is more design room for simplification of the backbone network. Converting the bounding box representation to circle representation is straightforward and simple. As it uses Gaussian heatmap to describe, center point positional information can be combined with target radius.

The original ground-truth bounding box can be described as x_b, y_b, w, h , which stand for center point coordinates x_b, y_b and bounding box width and height. We transform the bounding box to circle representation as (x_c, y_c, r) :

$$(x_c, y_c) = (x_b, y_b),$$

$$r = \begin{cases} \frac{\sqrt{w \cdot h}}{2}, & \frac{w}{h} > 0.7 \text{ or } \frac{w}{h} < 1.3, \\ \frac{\pi}{\sqrt{w \cdot h}}, & \text{otherwise.} \end{cases} \quad (1)$$

where (x_c, y_c, r) stands for coordinates of center of bounding circle and its radius. The region of width and height ratio is $(0.7, 1.3)$ for the round-shape targets. For the ground truth heatmap of the center point, the variance σ_x, σ_y of Gaussian distribution used to produce heatmap is set to

$$\sigma_x = \sigma_y = \begin{cases} \frac{r}{6}, & \frac{w}{h} > 0.7 \text{ or } \frac{w}{h} < 1.3, \\ \frac{0.6 \times \min(w, h)}{12}, & \text{otherwise.} \end{cases} \quad (2)$$

3) LOSS FUNCTION

CircleDet follows the basic definition of terms of CenterNet. The input image I is defined as $I \in R^{W \times H \times 3}$, where W and H are width and height of the image. The output of CircleDet has three branches: heatmaps of center, radius of circle, and offset.

The branch of the center point heatmap shows the center point localization of each object, which is defined as $\hat{Y} \in [0, 1]^{\frac{W}{R} \times \frac{H}{R} \times C}$. C is the number of classes and R is the downsample ratio of predicted heatmap output size. In predicted heatmap \hat{Y} , the center point pixel of the head or body ideal value should be 1. To let CircleDet learning center point easier, the ground truth of the target center point is modeled as a 2D Gaussian kernel:

$$Y_{xyc} = \exp\left(-\frac{(x - \tilde{p}_x)^2 + (y - \tilde{p}_y)^2}{2\sigma_r^2}\right). \quad (3)$$

where \tilde{p}_x and \tilde{p}_y are the target class center point position in downsampled heatmap, the size of which is $\frac{W}{R} \times \frac{H}{R}$. σ_r^2 is standard deviation of kernel. Different from CenterNet, the σ_r^2 is based on the area of original bounding box in ground truth data. Details are presented in Section III-A2. The predicted heatmap is optimized using focal loss as

$$L_{hm} = \frac{-1}{N} \sum_{xyc} \begin{cases} (1 - \hat{Y}_{xyc})^\alpha \log(\hat{Y}_{xyc}) & \text{if } Y_{xyc} = 1 \\ (1 - Y_{xyc})^\beta (\hat{Y}_{xyc})^\alpha \log(1 - \hat{Y}_{xyc}) & \text{otherwise.} \end{cases} \quad (4)$$

where α and β are hyperparameters of focal loss.

Both branches of predicting circle radius and offset use L1 loss to meet the regression task and described as L_r and L_{off} . The offset predicting branch is the same in CenterNet, which is formulated to further refine the center point prediction caused by the difference between input and output map size. The supervision acts only at keypoints location \tilde{p} .

$$L_{off} = \frac{1}{N} \sum \left| \hat{O}_{\tilde{p}} - \left(\frac{p}{R} - \tilde{p} \right) \right| \quad (5)$$

$$L_r = \frac{1}{N} \sum \left| \hat{r} - r \right| \quad (6)$$

The overall loss is:

$$L_{all} = L_{hm} + \lambda_r L_r + \lambda_{off} L_{off} \quad (7)$$

where $\lambda_r = 0.1$ and $\lambda_{off} = 1$ to balance different task importance during training.

B. IDENTIFICATION AND TRACKING

To enumerate the passengers in a video, we need a method to identify and track each one recorded by camera. The method should be effective but simple enough to run on the edge device. We proposed an identification and tracking method based on intersection over union (IoU) using circle representation.

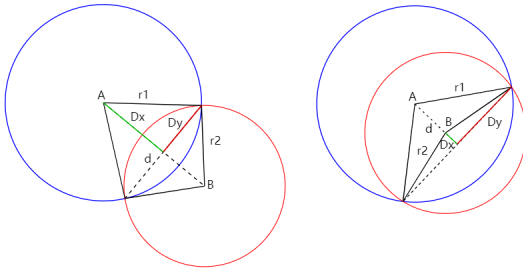


FIGURE 5. Circle IoU.

Calculating the IoU in circle representation is similar to the one in bounding box representation. In Fig. 5, we show two condition of calculation the IoU in the blue and red circles.

$$\text{circle IoU} = \frac{\text{Area}(A \cap B)}{\text{Area}(A \cup B)} \quad (8)$$

$$d = \sqrt{(x_A - x_B)^2 + (y_A - y_B)^2}$$

$$D_x = \frac{r_1^2 - r_2^2 + d^2}{2d}$$

$$D_y = \sqrt{r_1^2 - D_x^2} \quad (9)$$

$$\begin{aligned} \text{Area}(A \cap B) &= r_1^2 \sin^{-1}\left(\frac{D_y}{r_1}\right) + r_2^2 \sin^{-1}\left(\frac{D_y}{r_2}\right) \\ &\quad - D_y(D_x + \sqrt{r_2^2 - r_1^2 + D_x^2}) \end{aligned} \quad (10)$$

$$\text{Area}(A \cup B) = \pi r_1^2 + \pi r_2^2 - \text{Area}(A \cap B) \quad (11)$$

where $r_1, (x_A, y_A), r_2, (x_B, y_B)$ are radius and the center point coordinates of red and blue circles.

To identify the same person between adjacent frames, we assume that the camera frame rate is fast enough and

bounding circles of the same person head in adjacent frames would overlap. As a result, the circle IoU of two bounding circles would be large enough to identify the same person. Based on this assumption, a passenger identification and tracking algorithm is proposed.

CircleDet possesses frames of length T , which $\{F^t | t = 0, \dots, T\}$. For the moment of t , it give a result of detection in frame F^t , and we mark it as

$$D^t = \{H_i^t(c_i^t, r_i^t, M_i^t) | i = 0, \dots, n; n \in \mathbb{N}\} \quad (12)$$

which means n human heads are detected in frame F^t and the i^{th} human head is marked as H_i^t . H_i^t has three attributions (c_i^t, r_i^t, M_i^t) where (c_i^t, r_i^t) stands for coordinates of center point and radius for bounding circle and M_i^t for the identification mark of the person. Therefore, the result of detection in frame F^{t-1} is $D^{t-1} = \{H_j^{t-1}(c_j^{t-1}, r_j^{t-1}, M_j^{t-1}) | j = 0, \dots, m; m \in \mathbb{N}\}$. The calculation of circle IoU of H_i and H_j adjacent frames F^{t-1} and F^t can be written as

$$\eta_{ij} = \text{circleIoU}(H_i, H_j) \quad (13)$$

where η_{ij} is the circle IoU of bounding circles of H_i and H_j . If circle IoU result η_{ij} is more than the threshold value α , then the two of bounding circles in adjacent frames can be identified as the same person and share the same identification mark M , which means $M_i^t = M_j^{t-1}$.

After the identification step, tracking a passenger in the sight of the camera means tracking the bounding circle with the same identification mark M_k . The track can be written as

$$\text{Track}_{M_k} = \{H^\tau, \dots, H^{\tau+\Delta t}\} \quad (14)$$

where τ is the moment when $H^\tau(c^\tau, r^\tau, M_k^\tau)$ is first detected and M_k^τ is assigned to the $M^{\tau+\Delta t}$ of $H^{\tau+\Delta t}$ in the lasting Δt frames. When $\Delta t \geq \beta$, Track_{M_k} can be considered as an effective moving track of passengers to prevent a false positive detection result in which CircleDet detected some other things as human heads or a person wandering around the door. Thus, a passenger Track_{M_k} is recorded therefore the center point moving distance and direction are calculated to recognize passengers who are alighting or boarding.

IV. EXPERIMENTS

A. DATASET

In this section, the dataset acquisition is illustrated in detail.

1) DATASET CONSTRUCTION

To develop a dataset containing daily passenger volume in the metro station, a device contained high-definition (HD) camera and edge computing unit is mounted on top of the gap between the metro door and platform screen door (PSD) [34]. We mounted 24 devices in the metro station in Guangdong, China and recorded each PSD daily passengers alighting and boarding video. Among these videos, 6508 images were selected to construct a passenger detection task in our metro object detection (MOD) dataset. The MOD dataset not only contains detection task images and their labels such as passengers, foreign object and unusual behavior of passengers

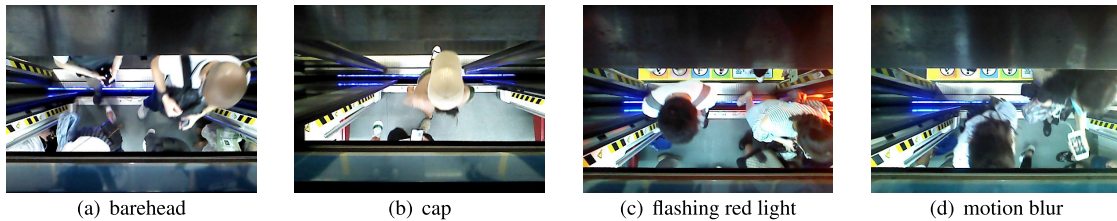


FIGURE 6. Abnormal human heads situation.

but also a large amount of daily metro videos data. In this study, we collected 6508 images from the MOD dataset as training data and 20 times of metro traveling data to validate our algorithm, $28 \times 24 = 672$ videos in total. The training dataset included all types of human heads, such as wearing a hat, bareheaded and white hair.

Although the cameras were mounted on the fixed angle, many variants can cause degradation of images. For example, as the limitation of the edge device, the frame rate of videos is approximately 8 to 10 FPS (frames per second), which causes blur image frame. Also, the illumination may vary from one PSD to another and cause a different situation of light source. Specifically red and white flashing light bands are mounted on the end of the platform, which causes challenges in the enumerating task. In Fig. 6, several examples are shown.

2) TRAINING, VALIDATION, AND TEST DATASETS

Although the CircleDet uses circle representation to predict heads of passengers, the origin ground truth of the MOD dataset still uses bounding box representation. Each labeled image frame corresponds to an XML file, which contains various information of it (file name, storage path, width, height, class, and bounding box coordinate). Our dataset consists of two clas, human head and body part, which are labeled as *head* and *person*. To generate the testing dataset, 20% of the images are randomly selected from annotation ones, namely 10% randomly selected images constructed the validation dataset and the rest constructed the training dataset.

B. IMPLEMENTATION DETAIL

1) NETWORK BLOCK DESIGN

The original CenterNet backbone used bottleneck residual block and the multi-branch block in ResNeXt [35] as a basic block combined with the DLA network design. However, limited by the edge device computation ability and memory, these network designs still consume a large amount of resources. As shown in Fig. 3, we replace the basic block with modified MBConv block. The basic block in CircleDet is designed as shown in Fig. 7.

2) TRAINING AND TESTING PLATFORM

The CircleDet and the other network models are trained on a computer with an Intel Core i9 7920X central processing unit (CPU), 48 GB DDR4 memory, and NVIDIA three GeForce RTX 2080Ti graphic processing units (GPUs). Three types of

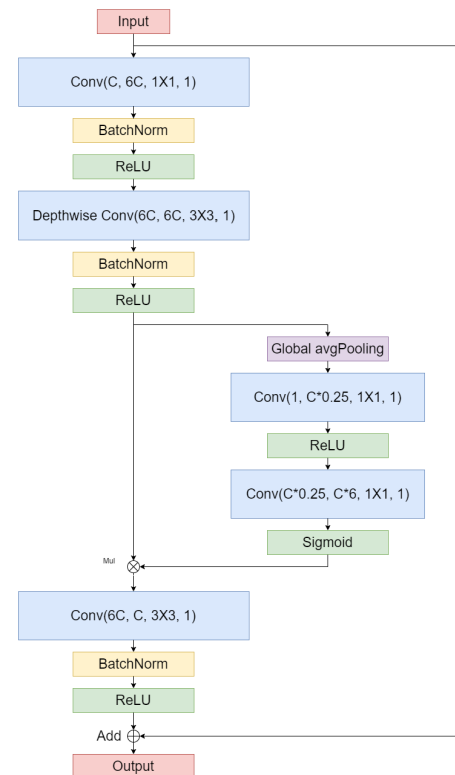


FIGURE 7. The basic block of CircleDet. The convolution layers parameter in the figure is (input channels, output channels, kernel size, stride).

test platforms are used. One is a personal computer (PC) with Intel Core i5 6500 CPU, 16 GB DDR4 memory, NVIDIA GeForce 2080 GPU. Another one is NVIDIA Jetson Nano with Quad-core ARM Cortex-A57 MPCore CPU, 4 GB LPDDR4 memory and NVIDIA CUDA core as testing edge device platform. The computation ability of different devices can vary from one to another. Specifically, the edge device of our testing platform, namely, Jetson Nano, can only provide maximum 472 GFLOPS (giga floating-point operations per second).

- Training stage: We train our model using Pytorch. We use Adam optimizer with an initial learning rate of 0.000125, and drop learning rate by 10% in step 90 and 120 epochs. We train on our MOD dataset with a batch size of 32.

- Testing stage: We test and compare our networks against other network models in detection task. To validate the efficiency of our algorithm for the whole pipeline, we used

TABLE 1. Experiments results of networks in detection task.

model	AP@IoU=0.50	Parameters (M)	FLOPs (G)	CPU Inference Time (ms/frame)	GPU Inference Time (ms/frame)
YOLOv3 [9]	89.66	-	-	417	9
SSDLite [10]	89.84	-	-	58	4
ppYOLO [36]	89.50	-	-	345	5
CenterNet [11]	98.1	16.373	25.890	1906	12
CircleDet	92.2	2.874	7.849	130	9

TABLE 2. Experiments results of different modification in CircleDet.

model	AP@IoU=0.50	Parameters (M)	FLOPs (G)	CPU Inference Time (ms/frame)	GPU Inference Time (ms/frame)
CircleDet_Res_DLA	92.9	16.373	25.886	1965	13
CircleDet_MBConv_DLA	92.6	13.714	26.868	340	13
CircleDet_MBConv_d1	92.6	10.901	23.048	268	11
CircleDet_MBConv_c8	91.1	3.594	8.837	147	11
CircleDet_MBConv_d1c8	92.2	2.874	7.849	130	9
CircleDet_MBConv_d1c8Leaky	91.6	2.874	7.876	129	9

672 videos to test the accuracy and recall. These videos contained video parts of passing passengers up to 13733.

C. RESULTS

As the passenger volume enumeration depends on the detector and tracking algorithm, the results of experiments are divided into two part in this section, namely, detection and enumeration tasks. The detection task section mainly shows the result of CircleDet. The enumeration task section mainly shows the test result of overall pipeline.

1) EVALUATION METRICS

The evaluation metrics often used in object detection is mAP, which means the average of Average Precision (AP) over all categories. In this study, our detection targets are human heads and bodies. We follow the calculation method in VOC2010: to calculate AP, we sample the monotonically decreasing curve at a fixed set of uniformly spaced recall values from 0 to 1 by step of 0.1. Evaluating the speed and efficiency is important because it needs to be run on the edge device. We calculate the parameters amount, time usage for each frame, and floating point operations (FLOPs). For the enumeration task, we compute the enumeration task error by

$$Error = \frac{Prediction - GroundTruth}{GroundTruth} \times 100\% \quad (15)$$

2) DETECTION TASK

The detection task in our algorithm aims to locate the heads of passengers in each frame. To further provide additional location information of passengers, we also add the *person* class in our dataset. *Person* class annotation in the dataset is a bounding box that covers the whole body of a passenger.

As shown in Table 1, we compared AP, amount of parameters, FLOPs and inference time of different models. In Table 2, we apply different network design to CircleDet. We use *network_basic block_backbone design* to identify. In *basic block*, *Res* stands for using the residual bottleneck block as basic block in the network. *MBConv* stands for applying MBConv block which is shown in Fig.7. In *backbone design*, there are different version of CircleDet, *DLA* stands for using basic DLA network architecture design. The suffix stand for modifications in DLA network: *l* for shallower HDA depth as shown in Fig.3, *c8* for reduced channel design, *Leaky* for using leaky ReLU as activation function in the network.

As shown in Table 1, the methods based on deep learning can achieve impressive results while maintaining a fast detection speed on GPU. The results are compared with CenterNet with DLA backbone, YOLOv3 with mobilenetV3 backbone, SSDLite with mobilenetV3 and FPN backbone, and ppYOLO. At first glance, CircleDet is about 6% behind CenterNet in accuracy. As we further look into our detection result, we found that is the *person* class is not appropriate to circle representation. As the body part is not rotation consistent, it will lower the whole performance in our test. However, this condition reveals that the head and body part do not have the same rotation consistent property. Nevertheless, the accuracy of CircleDet is sufficient for our subsequent tracking and enumeration task.

CircleDet_MBConv_DLA, CircleDet_MBConv_d1, CircleDet_MBConv_c8 are the models that show different design and modifications mentioned in Section III-A on DLA34 effect on accuracy, amount of parameters, and FLOPs. The CircleDet_MBConv_d1c8 reduces the channels

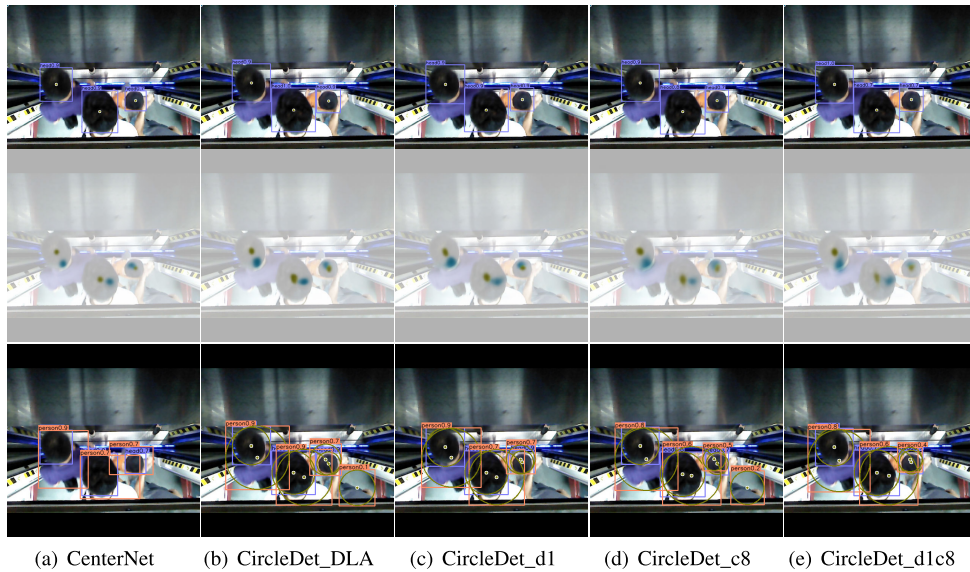


FIGURE 8. Detection result.

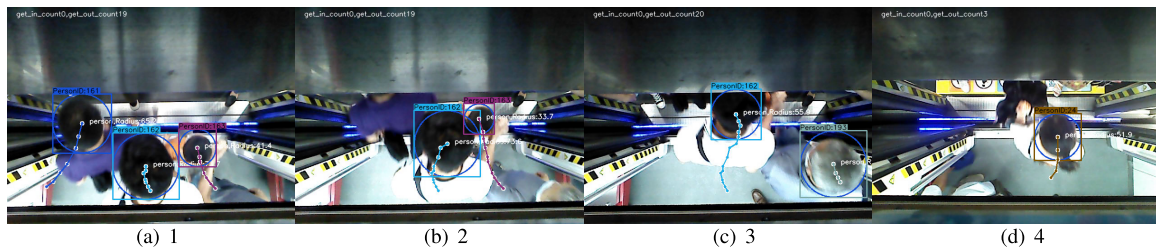


FIGURE 9. Examples of tracking and enumerating.

in the backbone, and reduce the depth of feature aggregation. It reduces large amount of parameters and inference FLOPs while maintaining relatively high accuracy of 92.2%.

This result shows that the original design of DLA is too complex and not necessary for the task of head detection. Compared with 16.373 million parameters and 25.890 FLOPs of CenterNet, CircleDet only has 2.874 million parameters and consumes 7.849 FLOPs. This means CircleDet is 5.7 times smaller and cost 3.29 times less computation resource than CenterNet.

For the inference time per frame, we test it on both CPU and GPU. Our best model costs 9 ms per frame running on GPU, in other words, 111 FPS. Although it may be seen that there is not much improvement, the situation is clearer when running on CPU (lots of edge devices have no GPU for deep neural network acceleration). CircleDet only costs 130 ms per frame while CenterNet needs 1906 ms. CircleDet is almost 14.7 times faster and maintains high accuracy. The performance of CircleDet meets the speed requirement in our real-time enumeration task.

When we further look into the detection result shown in Fig.8, we can see how circle representation affects the results of anchor-free design networks. The first row shows

the detection results of heads that will be used in subsequent tracking and enumeration. The second row shows prediction results of center point heatmaps. The third row shows all class prediction results and confidences, which includes *head* and *person* class.

3) ENUMERATION TASK

In the enumeration task, the entire algorithm pipeline applies to the real-world situation. We choose CircleDet_MBCConv_d1c8 which is mentioned in Detection Task as the detector in our pipeline. The passenger volume was enumerated in 672 videos, which recorded 28 metro trips in total. Some examples of trace tracking and enumeration are shown in Fig. 9.

The statistical data are shown in Fig. 10. The amount of passenger volume is enumerated in each metro trip. The number of passing passengers means that both directions, alighting and boarding, of passengers are enumerated and added together. Enumeration error of each trip is shown in Fig.11.

The total number is shown in the Table 3. False detection means our algorithm detect something else as a passenger and enumerates them. For example, black bags or suitcases are recognized as human heads and counted. Misclassification means

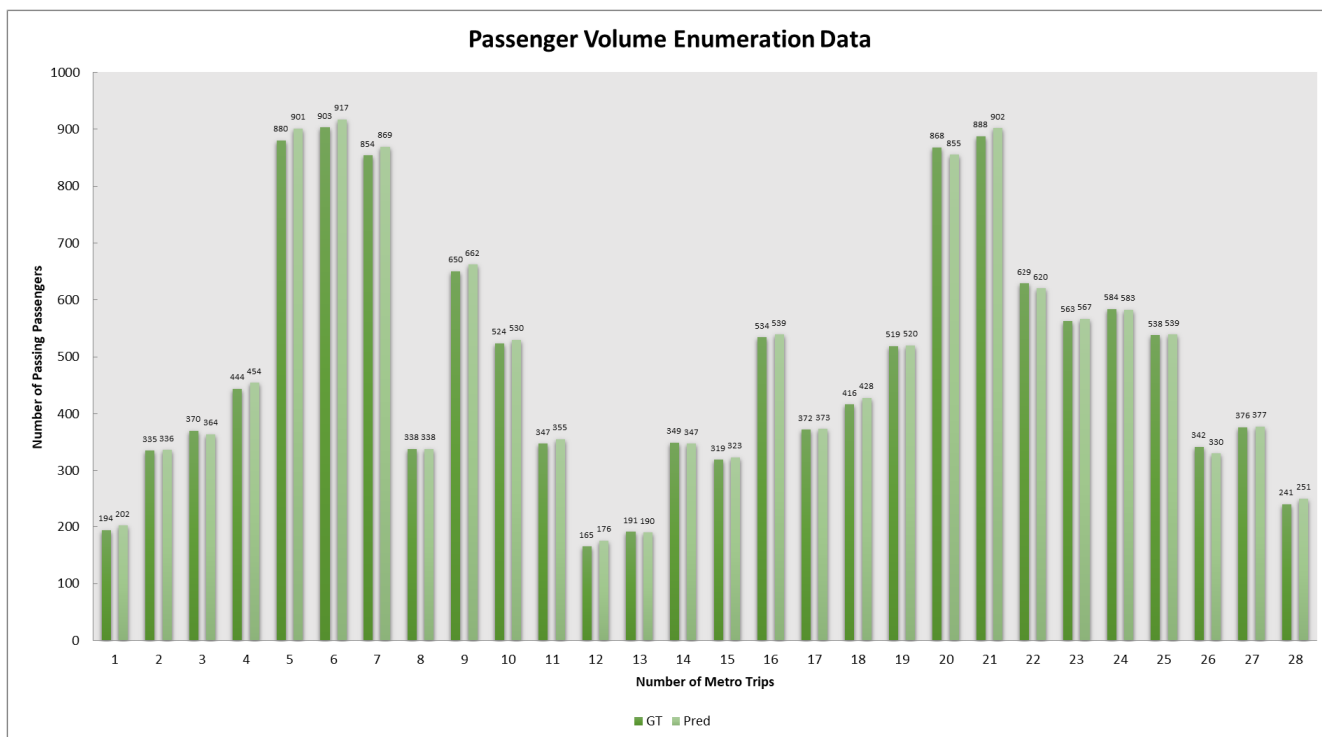


FIGURE 10. The enumerating result in 28 metro trips.

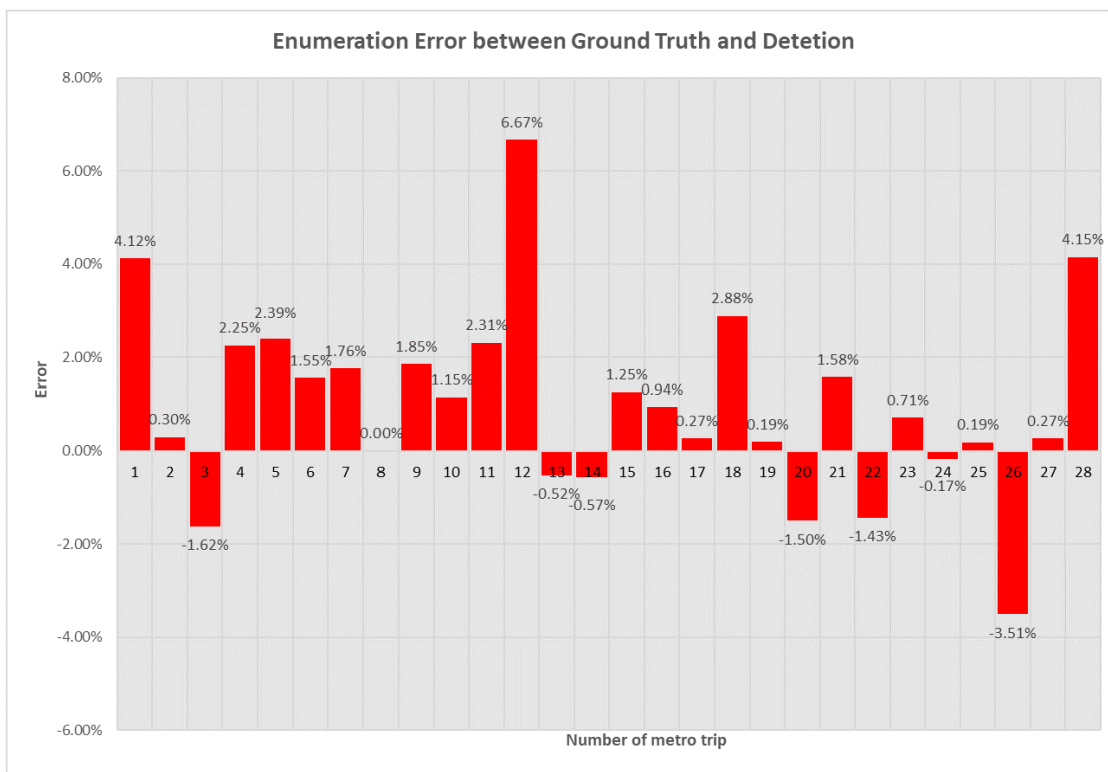


FIGURE 11. The enumerating error in 28 metro trips.

that some passengers are not counted in the videos. This situation happens when passenger heads are not recorded by the camera.

“GT” stands for real number of people and “Pred” stands for our own enumerating result. The number of passing passengers those who are alighting and boarding.

TABLE 3. Enumeration result of 672 videos.

	GT	Pred	False Detection	Miss Detection	Precision	Recall
Total In	6872	6932	232	172	0.966	0.975
Total Out	6861	6916	162	107	0.976	0.985
Totoal Count	13733	13848	394	279	0.971	0.980

We also found that our algorithm still perform well in situation that some passengers behave confusing. For example, there are some passengers wandering around the PSDs. Thanks to the tracking strategies mentioned above, our algorithm won't count them into the final result. It is because that only the passengers who walk in the correct direction and far enough can be counted in our strategies.

As shown in Table 3, the accuracy of our algorithm is high as 97.13% in 672 videos, 13733 passengers. This results mean that the algorithm is reliable and meets the real-world situation requirements. The data can be used in other studies, such as prediction of daily passenger volume, providing evidence of metro dispatching strategy and others.

Compared with traditional, manual methods to enumerate passenger volume, our method is faster, more accurate and effective. These advantages can reduce the detection time and provide real-time passenger volume as evidence of metro operation and further improve service levels.

V. CONCLUSION AND FUTURE WORKS

In this study, we propose a real-time algorithm of enumerating passenger volume based on an anchor-free backbone detection network CircleDet. First, we use cameras to record daily videos of passengers alighting and boarding in a real metro station from an overlooking angle. Then, we design an anchor-free network called CircleDet and use it as backbone of detector to detect heads of passengers in each frame. After the detection, we propose a simple but effective tracking and enumerating algorithm based on circle representation. Finally, we compare the parameters and time consumption and test our whole enumerating algorithm on real world dataset. The experimental results show that our algorithm is effective and fast enough to work in real time. Our algorithm can be used in real work and provide hard evidence of dispatching strategy and platform designing.

In the future, the following directions will be explored:

- More diverse passengers videos will be collected in different appearances to further increase the robustness of the algorithm. We will release the dataset used in this paper for research as soon as possible.
- In some situations, CircleDet would misrecognize a black bag as a human head and enumerate more passengers than real data. Therefore, the accuracy of CircleDet and the enumeration algorithm can be further improved to handle more complex situations and misleading targets.

- More robust identification and tracking algorithm are developing. Although the identification strategy in this paper is efficient and effective, it heavily relies on the accuracy of the head detector and once the detector losing the target of one frame in a continuous video may result in double counting.

ACKNOWLEDGMENT

The authors would like to thank the anonymous reviewers for their careful reading of their works and their many insightful comments.

REFERENCES

- [1] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *Int. J. Comput. Vis.*, vol. 60, no. 2, pp. 91–110, 2004.
- [2] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, vol. 1, Jun. 2005, pp. 886–893.
- [3] T. Ojala, M. Pietikäinen, and T. Mäenpää, "Gray scale and rotation invariant texture classification with local binary patterns," in *Proc. Eur. Conf. Comput. Vis. Springer*, 2000, pp. 404–420.
- [4] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 580–587.
- [5] R. Girshick, "Fast R-CNN," in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2015, pp. 1440–1448.
- [6] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 6, pp. 1137–1149, Jun. 2017.
- [7] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 779–788.
- [8] J. Redmon and A. Farhadi, "YOLO9000: Better, faster, stronger," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jul. 2017, pp. 7263–7271.
- [9] J. Redmon and A. Farhadi, "YOLOv3: An incremental improvement," 2018, *arXiv:1804.02767*. [Online]. Available: <http://arxiv.org/abs/1804.02767>
- [10] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "SSD: Single shot multibox detector," in *Proc. Eur. Conf. Comput. Vis. Springer*, 2016, pp. 21–37.
- [11] K. Duan, S. Bai, L. Xie, H. Qi, Q. Huang, and Q. Tian, "CenterNet: Keypoint triplets for object detection," in *Proc. IEEE Int. Conf. Comput. Vis.*, Oct. 2019, pp. 6569–6578.
- [12] X. Liu, J. van de Weijer, and A. D. Bagdanov, "Leveraging unlabeled data for crowd counting by learning to rank," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 7661–7669.
- [13] X. Liu, J. Van De Weijer, and A. D. Bagdanov, "Exploiting unlabeled data in CNNs by self-supervised learning to rank," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 41, no. 8, pp. 1862–1878, Aug. 2019.
- [14] M. Zhao, J. Zhang, C. Zhang, and W. Zhang, "Leveraging heterogeneous auxiliary tasks to assist crowd counting," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2019, pp. 12736–12745.
- [15] D. B. Sam, N. N. Sajjan, H. Maurya, and R. V. Babu, "Almost unsupervised learning for dense crowd counting," in *Proc. AAAI Conf. Artif. Intell.*, vol. 33, 2019, pp. 8868–8875.
- [16] A. Makhzani and B. J. Frey, "Winner-take-all autoencoders," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 28, 2015, pp. 2791–2799.
- [17] Z.-Q. Cheng, J.-X. Li, Q. Dai, X. Wu, J.-Y. He, and A. G. Hauptmann, "Improving the learning of multi-column convolutional neural network for crowd counting," in *Proc. 27th ACM Int. Conf. Multimedia*, 2019, pp. 1897–1906.
- [18] H. Cholakkal, G. Sun, F. S. Khan, and L. Shao, "Object counting and instance segmentation with image-level supervision," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 12397–12405.
- [19] F. Yu, D. Wang, E. Shelhamer, and T. Darrell, "Deep layer aggregation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 2403–2412.

- [20] M. Tan, B. Chen, R. Pang, V. Vasudevan, M. Sandler, A. Howard, and Q. V. Le, "MnasNet: Platform-aware neural architecture search for mobile," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2019, pp. 2820–2828.
- [21] F. N. Iandola, S. Han, M. W. Moskewicz, K. Ashraf, W. J. Dally, and K. Keutzer, "SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and <0.5 MB model size," 2016, *arXiv:1602.07360*. [Online]. Available: <http://arxiv.org/abs/1602.07360>
- [22] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, "MobileNets: Efficient convolutional neural networks for mobile vision applications," 2017, *arXiv:1704.04861*. [Online]. Available: <http://arxiv.org/abs/1704.04861>
- [23] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "MobileNetV2: Inverted residuals and linear bottlenecks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 4510–4520.
- [24] A. Howard, M. Sandler, B. Chen, W. Wang, L.-C. Chen, M. Tan, G. Chu, V. Vasudevan, Y. Zhu, R. Pang, H. Adam, and Q. Le, "Searching for MobileNetV3," in *Proc. IEEE Int. Conf. Comput. Vis.*, Oct. 2019, pp. 1314–1324.
- [25] H. Law and J. Deng, "Cornernet: Detecting objects as paired keypoints," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 734–750.
- [26] Z. Tian, C. Shen, H. Chen, and T. He, "FCOS: Fully convolutional one-stage object detection," in *Proc. IEEE Int. Conf. Comput. Vis.*, Oct. 2019, pp. 9627–9636.
- [27] Z. Cheng, Y. Wu, Z. Xu, T. Lukasiewicz, and W. Wang, "Segmentation is all you need," 2019, *arXiv:1904.13300*. [Online]. Available: <http://arxiv.org/abs/1904.13300>
- [28] Z. Yang, S. Liu, H. Hu, L. Wang, and S. Lin, "RepPoints: Point set representation for object detection," in *Proc. IEEE Int. Conf. Comput. Vis.*, Oct. 2019, pp. 9657–9666.
- [29] Y. Chen, Z. Zhang, Y. Cao, L. Wang, S. Lin, and H. Hu, "RepPoints V2: Verification meets regression for object detection," in *Proc. NeurIPS*, 2020.
- [30] T. Kong, F. Sun, H. Liu, Y. Jiang, L. Li, and J. Shi, "FoveaBox: Beyond anchor-based object detection," *IEEE Trans. Image Process.*, vol. 29, pp. 7389–7398, 2020.
- [31] Y.-W. Hsu, T.-Y. Wang, and J.-W. Perng, "Passenger flow counting in buses based on deep learning using surveillance video," *Optik*, vol. 202, Feb. 2020, Art. no. 163675.
- [32] J. Grönman, P. Sillberg, P. Rantanen, and M. Saari, "People counting in a public event—Use case: Free-to-ride bus," in *Proc. 42nd Int. Conv. Inf. Commun. Technol., Electron. Microelectron. (MIPRO)*, 2019, pp. 1055–1059.
- [33] H. Nakashima, I. Arai, and K. Fujikawa, "Proposal of a method for estimating the number of passengers with using drive recorder and sensors equipped in buses," in *Proc. IEEE Int. Conf. Big Data (Big Data)*, Dec. 2018, pp. 5396–5398.
- [34] Y. Dai, W. Liu, H. Li, and L. Liu, "Efficient foreign object detection between PSDs and metro doors via deep neural networks," *IEEE Access*, vol. 8, pp. 46723–46734, 2020.
- [35] S. Xie, R. Girshick, P. Dollar, Z. Tu, and K. He, "Aggregated residual transformations for deep neural networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 1492–1500.
- [36] X. Long, K. Deng, G. Wang, Y. Zhang, Q. Dang, Y. Gao, H. Shen, J. Ren, S. Han, E. Ding, and S. Wen, "PP-YOLO: An effective and efficient implementation of object detector," 2020, *arXiv:2007.12099*. [Online]. Available: <http://arxiv.org/abs/2007.12099>



WEIMING LIU received the Ph.D. degree from the National University of Defense Technology, China, in 2004. He is currently a Professor with the School of Civil Engineering and Transportation, South China University of Technology, Guangzhou, China. His research interests include digital image monitoring and identification, intelligent traffic system engineering theory and application, and intelligent road traffic control and management.



HENG WANG received the B.Sc. degree from Central South University Railway Campus. He is currently a Senior Engineer of railway traffic with Shenzhen Metro Group Company Ltd. His research interest includes computerized control technology for metro platform screen doors and management.



GUICI FAN received the B.Sc. degree from the Guangdong University of Technology. He is currently a Senior Engineer of railway traffic with Guangzhou Metro Corporation. His research interest includes automation of metro station doors and management.



ZHONGXING ZHENG was born in Foshan, Guangdong, China, in 1994. He received the B.S. degree from the Wuhan University of Technology, Wuhan, Hubei, China, in 2017. He is currently pursuing the Ph.D. degree in traffic information engineering and control with the South China University of Technology. His research interests include intelligent transportation, computer vision, and deep learning.



YUAN DAI (Graduate Student Member, IEEE) was born in Loudi, Hunan, China, in 1995. He received the B.S. and M.S. degrees from the Changsha University of Science and Technology, Changsha, Hunan, in 2016 and 2018, respectively. He is currently pursuing the Ph.D. degree in traffic information engineering and control with the South China University of Technology. His research interests include intelligent transportation, computer vision, and deep learning.