Real-Time and Accurate UAV Pedestrian Detection for Social Distancing Monitoring in COVID-19 Pandemic

Zhenfeng Shao, Gui Cheng, Jiayi Ma, *Member, IEEE*, Zhongyuan Wang, *Member, IEEE*, Jiaming Wang, and Deren Li

***Abstract*—Coronavirus Disease 2019 (COVID-19) is a highly infectious virus that has created a health crisis for people all over the world. Social distancing has proved to be an effective non-pharmaceutical measure to slow down the spread of COVID-19. As unmanned aerial vehicle (UAV) is a flexible mobile platform, it is a promising option to use UAV for social distance monitoring. Therefore, we propose a lightweight pedestrian detection network to accurately detect pedestrians by human head detection in real-time and then calculate the social distancing between pedestrians on UAV images. In particular, our network follows the PeleeNet as backbone and further incorporates the multi-scale features and spatial attention to enhance the features of small objects, like human heads. The experimental results on Merge-Head dataset show that our method achieves 92.22% AP (average precision) and 76 FPS (frames per second), outperforming YOLOv3 models and SSD models and enabling real-time detection in actual applications. The ablation experiments also indicate that multi-scale feature and spatial attention significantly contribute the performance of pedestrian detection. The test results on UAV-Head dataset show that our method can also achieve high precision pedestrian detection on UAV images with 88.5% AP and 75 FPS. In addition, we have conducted a precision calibration test to obtain the transformation matrix from images (vertical images and tilted images) to real-world coordinate. Based on the accurate pedestrian detection and the transformation matrix, the social distancing monitoring between individuals is reliably achieved.**

***Index Terms*—UAV, COVID-19, pedestrian detection, spatial attention, social distancing monitoring.**

Manuscript received September 24, 2020; revised December 15, 2020 and February 25, 2021; accepted April 19, 2021. Date of publication April 28, 2021; date of current version April 6, 2022. This work was supported in part by the National Key Research and Development Program of China under Grant 2018YFB2100501, in part by the Key Research and Development Program of Yunnan province in China under Grant 2018IB023, in part by the National Natural Science Foundation of China under Grants 42090012, 41771452, 41771454, and 41901340, and in part by the Consulting research project of Chinese Academy of Engineering under Grant 2020ZD16. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Abderrahim Benslimane. (*Corresponding author: Gui Cheng.*)

Zhenfeng Shao, Gui Cheng, Jiaming Wang, and Deren Li are with the State Key Laboratory of Information Engineering in Surveying, Mapping, and Remote Sensing, Wuhan University, Wuhan 430079, China (e-mail: shaozhenfeng@whu.edu.cn; 2019206190077@whu.edu.cn; wjmecho@163.com; drli@whu.edu.cn).

Jiayi Ma is with Electronic Information School, Wuhan University, Wuhan 430072, China (e-mail: jyma2010@gmail.com).

Zhongyuan Wang is with the National Engineering Research Center for Multimedia Software, Wuhan 430079, China (e-mail: wzy_hope@163.com).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TMM.2021.3075566.

Digital Object Identifier 10.1109/TMM.2021.3075566

I. Introduction

SINCE December 2019, the novel Coronavirus Disease 2019 (COVID-19) has caused severe acute respiratory syndrome globally and created a health crisis for people all over the world [1]–[3]. Social distancing is considered the most effective non-pharmaceutical measure to slow down the spread of COVID-19 [4]. How to calculate the social distancing between pedestrians in real time and accurately is of great significance.

With the rapid development of artificial intelligence, big data and other digital technologies, these technologies have played a supporting role in social distancing monitoring. The ordinary methods of social distancing monitoring [5], [6] are based on a monocular and intelligent surveillance camera that can only capture a certain area. In contrast, unmanned aerial vehicle (UAV) is flexible, convenient and wide in coverage. Social distancing monitoring based on UAV mobile platform is thus an ideal choice. UAV-based social distancing monitoring can basically include two steps, such as pedestrian detection and social distancing calculating.

In recent years, due to the strong learning ability of convolution neural networks (CNN), the state-of-the-art object detection algorithms are all based on deep learning [7]–[20]. These algorithms can be divided into two main categories. One type is two-stage methods [7]–[12], which divide detection into two parts, region proposal and classification. These methods can achieve high detection accuracy but consume time. Another one refers to single-stage methods [13]–[18], which treat detection as an end-to-end process to directly predict the location and categories of targets. These methods can achieve fast object detection, but have a lower accuracy than that of two-stage methods. A good detector should provide high accuracy as well as fast inference speed. The current pedestrian detection methods based on deep learning, include body detection [21], [22], shoulder detection [23], [24], and head detection [25]–[27]. In actual scenes, there exits occlusion between pedestrian, where only human heads can be seen. Therefore, the body detection methods are confronted with great limitations, while the human head detection is relatively more accurate.

Human heads belong to small objects in UAV images. As human head is larger in the vicinity and smaller in the distance in the UAV image, it still has multi-scale features. Multi-scale feature is an important feature of human head. The performance

of human head detection can be improved by multi-scale feature enhancement. Generally, multi-scale feature fusion and super-resolution reconstruction are used to enhance the feature of small targets like human heads. The methods based on multi-scale feature fusion, such as [28]–[30], are to expand the size of small-scale feature map through up-sampling, and then fuse it with feature map of the same size. The super-resolution reconstruction methods, such as [31]–[35] are based on the idea of image super-resolution, which can increase the resolution of small-scale feature map by super-resolution reconstruction, and obtain large-scale feature map to enhance the feature of small targets.

UAV images exhibit complex backgrounds due to diversified illumination, viewpoints, altitudes and scenarios. The complex background has great interference on the object detection. It is hard to detect the target quickly and accurately in such a condition. Can we pay more attention to our targets in the complex background? The spatial attention mechanism can achieve this purpose. Recent studies [36]–[40] have shown that spatial attention can enhance the feature that we are interested in and ignore the irrelevant information. The UAV is a flexible mobile device, whose memory and computing power are very limited during the timely image processing, which makes real-time object detection based on UAV image a great challenge. However, for the huge amount of data collected by UAV, it is necessary to process such data in order to obtain useful information in a timely manner. Although the current state-of-the-art object detection algorithms, such as Faster R-CNN [10], YOLOv3 [16], R-FCN [11], RetinaNet [13], perform well on natural images, their model size are relatively large, so that they are unsuitable for mobile devices like UAV. In recent years, numerous lightweight and accurate CNN based models have been proposed for mobile platforms, such as MobileNet [41] and ShuffleNet [42]. However, these networks rely heavily on deep separable convolution and lack effective implementations in some deep learning frameworks. Recently, an efficient lightweight network, PeleeNet [43], is proposed, which can be achieved just with traditional convolution and can fully extract features with fewer parameters.

From the above analysis, we perform UAV based pedestrian detection by fast and accurate human head detection in this paper. In particular, we propose a lightweight pedestrian detection network specifically for head detection, which follows the PeleeNet as the backbone and further incorporates the multi-scale features and spatial attention. Because there exist few large-scale pedestrian datasets for UAV images, in order to train an efficient pedestrian detection model based on UAV images, we firstly employ a large number of existing video surveillance datasets for network training and then make use of a small UAV image dataset that we built to fine-tune the model trained on the surveillance video datasets. As a matter of fact, the UAV image is highly similar to the video surveillance image, as shown in Fig. 1. Therefore, using the existing human head datasets from different video surveillance for pre-training and then fine-tuning the network model on the small UAV dataset can save a lot of resources and obtain an efficient model that we want. The video surveillance datasets consist of three human head datasets, including Brainwash [44], SCUT-HEAD [45] and our FerryHead,

Fig. 1. (a) are the samples from VisDrone2018 [46] UAV dataset. (b) are the samples from video surveillance. The samples in these two data sources look very similar.

which are from various scenes with diversified viewpoints, illuminations and scales. The small UAV dataset that we built, called UAV-Head, contains 745 images in size of 1920×1080 pixels.

Pedestrian detection is used for social distancing monitoring in the COVID-19 prevention. Social distancing is defined as the physical contact distance between each individual. Generally speaking, the safety distance between individuals needs to be more than 2 m. In order to calculate correctly the social distance, we need detect the position of each pedestrian accurately in UAV images and then compute the projection transformation matrix from UAV images to real-world coordinate.

UAV is a flexible mobile photography platform, which can be used to photograph ground objects at different altitudes and in different attitudes. The projection transformation matrix from the image to real-world coordinate system will vary with the altitude and attitude of the UAV. Therefore, the transformation projection matrix cannot be obtained in real time under the condition of unconstrained photography. In this paper, we first make a precision calibration test including vertical images and tilted images to gain the relationship between image and real-world coordinate. Then the UAV takes photography at different scenarios under the condition of calibration. Based on calibration results, the position of each pedestrian in real-world coordinate system can be readily obtained.

Compared with fixed surveillance video cameras on the ground, UAVs are more flexible and can provide real-time dynamic information in any area. Based on these advantages, UAV is an idea social distance monitoring platform. Our proposed pedestrian detection network is lightweight that can be directly used in small mobile platforms like UAVs, to achieve real-time and accurate pedestrian detection. Therefore, we designed a social distancing monitoring system based on UAV, the whole process of which is shown in Fig. 2, including 5 steps. 1) The UAV takes photography under the condition of calibration. 2) By using our pedestrian detection algorithm, the image pixel coordinate of each pedestrian's head can be calculated. 3) The coordinate of each pedestrian in the real world can be quickly calculated by projection transformation matrix. 4) Based on the real-world coordinate, the distance between each pedestrian is calculated and stored in a distance upper triangular matrix. 5) Determine whether each distance in the matrix is less than 2 meters. If so, it indicates that there is a clustering situation in the area. Then alert is performed. In short, this system can monitor the social distance of each area in real time while UAV flying and plays an important in COVID-19 pandemic prevention.

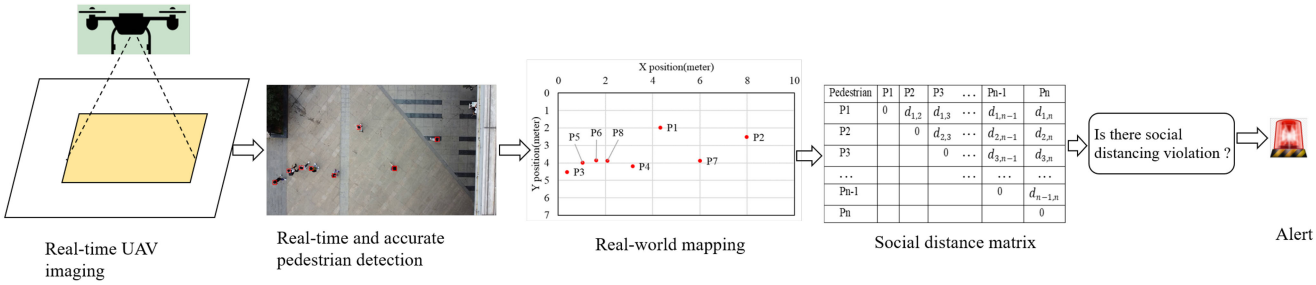


Fig. 2. The overview of social distancing monitoring system.

The main contributions of our work are summarized as follows:

1) We performed UAV based pedestrian detection by fast and accurate human head detection. Particularly, we proposed an efficient and lightweight pedestrian detection network that adopts the PeleeNet as the backbone and combines the multi-scale features and spatial attention. It can achieve a real-time and accurate pedestrian detection on the UAV mobile platform.

2) To alleviate the difficulty of insufficient UAV training samples, we firstly used a large number of existing video surveillance datasets for network training and then leveraged a small UAV image dataset that we built to fine-tune the pretrained model.

3) We proposed a social distancing monitoring method based on the map relationship from UAV images to the real-world coordinate system.

The rest of this paper is organized as follows. In section II, we review the related work of pedestrian detection based on UAV, object detection based on spatial attention and social distancing monitoring. In section III, we present our method in detail. Section IV illustrates the experiments and analysis of our method in comparison with state-of-the-art methods and shows results of social distancing monitoring. Finally, Section V draws a conclusion of this paper.

II. Related Work

A. Pedestrian Detection Based on UAV

The traditional object detection methods usually extract the hand-crafted features and then utilize a classifier to predict the location and category of targets, such as HOG features with SVM classifier [47], and Haar-like features with AdaBoost classifier [48]. However, most of the traditional object detection methods are time-consuming, labor-consuming and poorly robust [49]. These low-level automation methods often fail to meet the requirements in the practical applications. Recently, pedestrian detection based on deep convolutional neural network (CNN) has made great improvements [50]–[56]. In natural pedestrian detection scenes, pedestrians are photographed from the side and most of their scales are relatively large. Li *et al.* [50] designed a scale-aware Fast R-CNN framework to detect pedestrians with scales from disjoint ranges. Wang *et al.* [51] made good use of body part semantic information and contextual information to design a high accurate pedestrian detector. Zhang *et al.* [52] explicitly model people's semantic attributes in a high-level feature detection fashion to accurately detect pedestrian in a crowded group. However, pedestrians in UAV images are different from ordinary pedestrians in video images. They are captured from diversified perspectives, with various scales and shapes, which leads to more complex scenarios. For pedestrian detection in UAV images, Ma *et al.* [57] proposed a two-stage blob-based approach (first extracting pedestrian blobs and then classifying the detected blobs) for pedestrian detection using thermal infrared images recorded from UAVs. Aguilar *et al.* [53] utilized HAAR-LBP cascade classifiers with AdaBoost training and saliency maps to detect pedestrian in UAV. AlDahoul *et al.* [54] achieved real-time human detection from aerial captured video with different altitudes using automatic feature learning methods which combine optical flow and three different deep models. In order to achieve the accurate object detection in UAV image on the premise of real-time processing, Zhang *et al.* [55] proposed a coarse-to-fine object detection method for UAV image which combines lightweight convolutional neural network and deep motion saliency. However, these pedestrian detection methods do not solve the occlusion situation very well.

For the reason that there are always occlusions in UAV images where only the human heads can be visible, human head detection is a good choice to accurately detect pedestrian. In recent years, various head detectors based on deep learning have emerged. Vu *et al.* [58] proposed a context-aware CNN-based model that extends the R-CNN [9] object detection model by using two types of contextual cues for head detection from video data. Gao *et al.* [59] designed a cascade AdaBoost head detector based on CNN that uses HOG as the feature representation and has fewer head region proposals. Li *et al.* [60] combined the regional context with the feature fusion strategy to improve the head detection performance. [27] proposed an end-to-end head detection method that integrates low-level local information with the semantic features of the upper layer. This method based on SSD [15] shows a good performance on the detection of small-size human heads. For quick and accurate head detection in crowded scenes, [26] proposed a lightweight model called fully constitutional head detector which can perform both classification and bounding box prediction. This method uses a series of anchor scales that can adapt to the size of human heads. Li *et al.* [25] used an adaptive relational network for head detection, which can capture context information.

B. Object Detection Based on Spatial Attention

An important feature of the human visual system is that people do not try to deal with the whole scene at once. Instead,

humans take advantages of a series of partial glimpses, selectively focusing on the salient parts in order to better capture the visual structure [61].

In recent years, to improve the performance of CNNs, numerous studies on object detection combined with spatial attention have emerged [36]–[40]. Wang *et al.* [40] designed a residual attention network, a convolutional network with mixed attention mechanism in a very deep structure, which can not only generate attention-sensing features, but also show strong robustness against noisy labels. To adaptively refine the intermediate feature map, [39] proposed a lightweight convolutional block attention module (CBAM) by combining the channel and spatial attention. To relax the local neighborhood constraint, Zhao *et al.* [38] used a self-adaptively learned attention map to connect each position in the feature with all the others. Chen *et al.* [37] used multi-scale spatial and channel-wise attention mechanism to improve the performance in detecting objects with different backgrounds and sizes in remote sensing images. To deal with the scale variation, [36] designed a spatial-refinement module in which the spatial details of multi-scale objects in images can be repaired. Spatial attention can be used to extract important feature information that we are interested in and ignore irrelevant information.

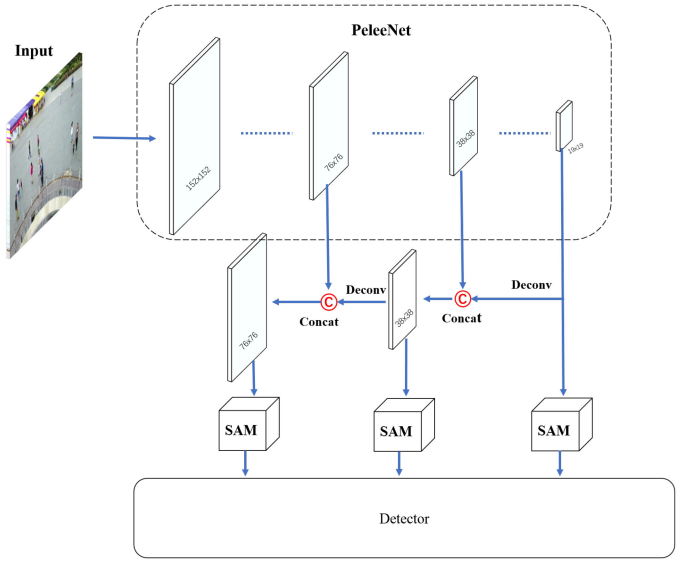


Fig. 3. The architecture of our network. It consists of three parts, including PeleeNet, multi-scale spatial attention module and detection layer. The PeleeNet is used to extract features as backbone. There are three scales of features, namely 19×19, 38×38 and 76×76. Each scale of feature is processed by spatial attention module (SAM) before putting into detection layer.

C. Social Distancing Monitoring

Social distancing refers to the physical contact distance between people. The spread of disease can be reduced through social distancing control, such as closing public places (e.g, schools and workplaces), avoiding crowds and keeping enough distance between people. The minimum social distancing between people is usually 2 meters, which can effectively reduce and avoid possible contact. In recent years, a number of emerging technologies have contributed to the monitoring of social distancing. In work [62], Nguyen *et al.* showed how emerging technologies (e.g, wireless, networking, and artificial intelligence) can enable or even enforce social distancing. The basic concepts, measurements, models and practical scenarios of social distancing were discussed in this work. A specific approach of social distancing monitoring was proposed in [5]. First, they used YOLOv3 and Deepsort respectively to detect and track pedestrians under surveillance video. Then, the social distancing was obtained by computing the pair-wise vectorized L2 norm. Finally, a violation index was calculated for non-social distancing behaviors. A similar work was done by Yang *et al* [6]. They proposed an artificial intelligence based real-time social distancing detection and warning system by using a monocular camera. However, these two works are based on a fixed surveillance camera that can only capture a certain area. In contrast, social distancing monitoring based on UAVs is more flexible and wide-ranging than surveillance cameras.

III. Method

A. Pedestrian Detection Network

We detect pedestrians by detecting human heads on UAV mobile platform, and there are three aspects to consider. 1) In order to perform pedestrian detection on a small mobile platform such as UAV, the algorithm must be lightweight to achieve real-time performance. 2) For such a small target as human head, in the process of perspective imaging, it still has multi-scale features and the use of a single scale often leads to missed detection. 3) In a complex background, the head is easy to be confused with other objects. If we can effectively enhance the features of human head while ignoring other irrelevant information, we can effectively distinguish human heads from other objects. To this end, we proposed a real-time and accurate pedestrian detection network which adopts the PeleeNet as backbone and combines the multi-scale feature and spatial attention.

Our network consists of three parts, including PeleeNet, multi-scale spatial attention module and detection layer, as shown in Fig. 3. As the backbone, PeleeNet can fully extract features based on an improved dense connection. The multi-scale spatial attention module is used to integrate the features of multiple scales to enhance the information of small targets. Meanwhile, the spatial attention information of different scales is conducive to object detection. The detection layer is used to predict the location of target.

1) Backbone: We use PeleeNet [43] as backbone which is a lightweight network variant based on DenseNet [63]. PeleeNet follows DenseNet's innovative connectivity patterns and some key design principles. There are several tricks in PeleeNet that contribute to feature learning. More details can be found in [43]. First, the Stem block adopts convolution and maxpooling branches for down sampling, which can ensure strong feature expression ability without too much computational complexity. Then the feature information is extracted using the Dense Layer + Transition structure iteratively. Different from the original DenseNet, a Two-way Dense layer is designed in PeleeNet

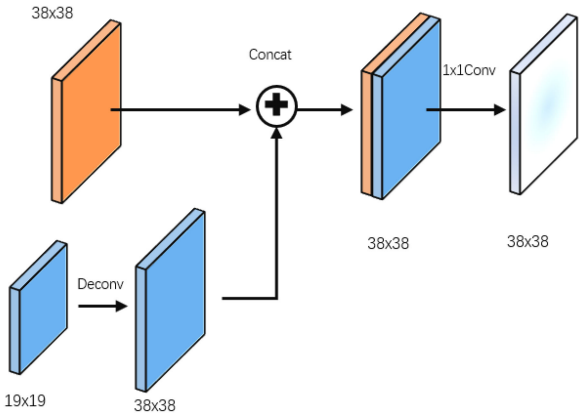


Fig. 4. The multi-scale feature fusion.

which combines 1×1 and 3×3 convolution to extract the different scales of receptive field. At the same time, the number of output channels in Bottleneck Layer changes with the shape of input, so as to ensure that the number of output channels does not exceed the input channels and the calculation amount of Bottleneck Layer does not increase significantly. In the Transition Layer, the number of output channels is not compressed, in other words the output channels have the same number as the input.

2) Multi-Scale Spatial Attention: In this paper, we used three scales to conduct head detection, 19×19, 38×38 and 76×76 respectively. Human head is a small target in UAV image so that we need to retain and enhance the feature information as much as possible. We adopt multi-scale feature fusion in our network. In order to extract the deep semantic feature information, the general neural network will carry out multiple down-sampling operations to obtain small-size feature maps. For example, the size of feature maps in SSD [15] are respectively 38, 19, 10, 5, 3, 1. The minimum size of feature is 1×1. Although deep semantic features can be extracted in this way, a large amount of feature information will be lost for small targets. Therefore, in this paper, in order to retain the feature information of small targets, excessive down-sampling is avoided. At the same time, larger feature maps are used for prediction. The minimum size of the feature map is 19×19.

For multi-scale feature fusion, we use deconvolution [64] as up-sampling. In general, the interpolation methods are used for up-sampling, but deconvolution carries out up-sampling based on network learning method which can be trained to obtain better up-sampling parameters. The multi-scale feature fusion process is shown in Fig. 4. Firstly, feature map of 19×19 is up-sampled via deconvolution to obtain feature map of 38×38. Then, concatenation is conducted between it and the previous feature map. Finally, 1×1 convolution is adopted for feature fusion.

Spatial attention is an important feature that can enhance the feature information and ignore the irrelevant information. The architecture of spatial attention module can be seen in Fig. 5.

The process of spatial attention module can be denoted as

$$F' = M_s(F) \otimes F \tag{1}$$

Where the input feature map is denoted as $F \in \mathbb{R}^{C \times W \times H}$. The spatial attention map is a 2D map denoted as $M_s \in \mathbb{R}^{1 \times W \times H}$. After the spatial attention process, the output refined feature maps are denoted as $F' \in \mathbb{R}^{C \times W \times H}$.

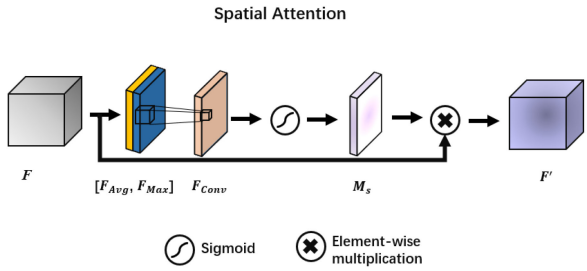


Fig. 5. The architecture of spatial attention.

We use AvgPooling and MaxPooling to aggregate channel information of input feature maps respectively, generating two maps, denoted as $F_{Avg} \in \mathbb{R}^{1 \times W \times H}$ and $F_{Max} \in \mathbb{R}^{1 \times W \times H}$, which represent the average pooling characteristic and the maximum pooling characteristic of the whole channels respectively. Then the F_{Avg} and F_{Max} are concatenated. Finally, the spatial attention map M_s is generated by a convolution layer, calculated by

$$\begin{aligned} M_s(F) &= \sigma(f^{(7\times7)}([AvgPool(F); MaxPool(F)])) \\ &= \sigma(f^{(7\times7)}([F_{Avg}; F_{Max}])) \end{aligned} \tag{2}$$

Where σ represents the Sigmoid function and $f^{7\times7}$ represents a convolution operation with filter size of 7×7.

3) Detection Layer: We follow the detection principle of YOLOv3 [16]. Firstly, feature extraction is carried out on the input image through the network to obtain feature maps of different sizes. The detection principle is to divide the feature map into $S \times S$ grids, such as 19×19, 38×38, 76×76, and then determine whether each grid contains the center of the target. If so, then this grid is responsible for detecting the target. Assume that each grid can predict 3 bounding boxes of different sizes, of which only the largest intersection over union (IoU) value with the ground truth is used to predict this object. IoU is a common metric in object detection. The difference between the predicted results and ground truths can be measured by calculating the IoU. The larger the IoU value is, the closer the predicted result is to the ground truth. Each bounding box consists of 6 predicted values: x, y, w, h, c, p. Where, (x, y) represents the center point of the bounding box. (w, h) denotes the ratio of the width and height of the bounding box to the entire image. c is the bounding box confidence, and indicates the IoU between the predicted bounding box and ground truth. At the end, each bounding box contains a probability p of head.

In order to refresh the weights of network parameters, the cost loss will be evaluated after each iteration in training. When calculating the loss, x^*, y^*, w^*, h^*, c^*, p^* are considered as the value of ground truth of object. The size of feature map is $S \times S$, and each grid can predict B bounding boxes. For each grid, we need to know whether it contains center point of object. So we set the P_{obj} as the measure. If the grid contains center of object, the value of P_{obj} is 1; otherwise, the value of P_{obj} is 0. The loss of our model in each image includes $Loss_{xy}$, $Loss_{wh}$, $Loss_c$, $Loss_p$.

1) The loss of (x, y) can be calculated in Eq. (3). Where the *BCE* is binary cross entropy loss function, λ is defined as Eq. (4).

$$Loss_{xy} = \lambda \sum_{i=1}^{S^2} \sum_{j=1}^{B} P_{obj} \times [BCE(x_{ij}) + BCE(y_{ij})]$$

$$BCE(x_{ij}) = x_{ij}^* log x_{ij} + (1 - x_{ij}^*) log(1 - x_{ij})$$

$$BCE(y_{ij}) = y_{ij}^* \log y_{ij} + (1 - y_{ij}^*) log(1 - y_{ij}) \quad (3)$$

$$\lambda = (2 - w^* \times h^*) \quad (4)$$

2) The loss of (w, h) can be calculated in Eq. (5).

$$Loss_{wh} = \frac{1}{2} \lambda \sum_{i=1}^{S^2} \sum_{j=1}^{B} P_{obj} \times [(w_{ij} - w_{ij}^*)^2 + (h_{ij} - h_{ij}^*)^2] \quad (5)$$

3) The loss of c can be calculated in Eq. (6). Where the c* is defined in Eq. (7).

$$Loss_c = \sum_{i=1}^{S^2} \sum_{j=1}^{B} P_{obj} \times BCE(c_{ij}) + (1 - P_{obj}) \times BCE(c_{ij})$$

$$BCE(c_{ij}) = c_{ij}^* \log c_{ij} + (1 - c_{ij}^*) log(1 - c_{ij}) \quad (6)$$

$$c^* = P_{obj} \times IoU \quad (7)$$

4) The loss of p can be calculated Eq. (8).

$$Loss_p = \sum_{i=1}^{S^2} \sum_{j=1}^{B} P_{obj} \times BCE(p_{ij}) \quad (8)$$

$$BCE(p_{ij}) = p_{ij}^* \log p_{ij} + (1 - p_{ij}^*) log(1 - p_{ij})$$

The total loss of each image is the sum of each loss, which is defined as Eq. (9). In addition, the loss of each batch of images is defined as Eq. (10), where the b is the batch size.

$$Loss\_img = Loss_{xy} + Loss_{wh} + Loss_c + Loss_p \quad (9)$$

$$Loss\_batch = \frac{1}{b} \sum_{k=1}^{b} Loss\_img_k \quad (10)$$

B. Social Distancing

1) Image to Real-World Coordinate: Image to real-world coordinate refers to the coordinate transformation between two planes. Here, we take the ground plane as the plane of the real world, and one of the planes in the image is needed to be transformed. The coordinates of any point on the pedestrian body are the same in the real-world plane while different in images. This is caused by the image point displacement in the process of camera perspective projection imaging due to the height of pedestrian, as shown in Fig. 6. The more details about image point displacement can been seen in [65]. Therefore, in order to avoid the error caused by the image point displacement, we need determine a plane parallel to the ground in the image to carry out coordinate transformation. The two most obvious planes in the image are the planes of the pedestrian's head and foot. Actually, the plane of the pedestrian foot is the ground plane in image.

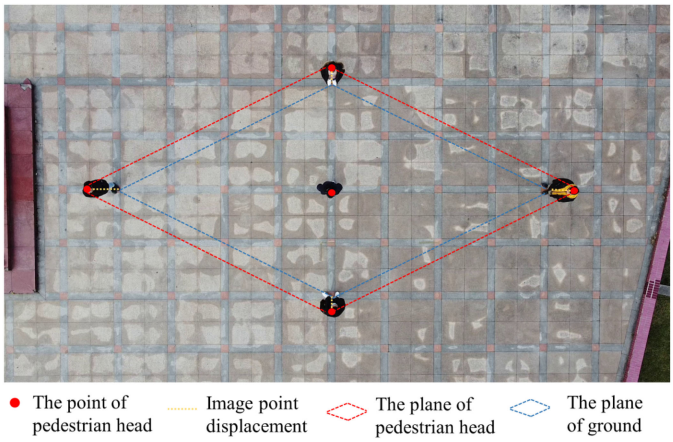


Fig. 6. The image point displacement in vertical image due to the height of pedestrian.

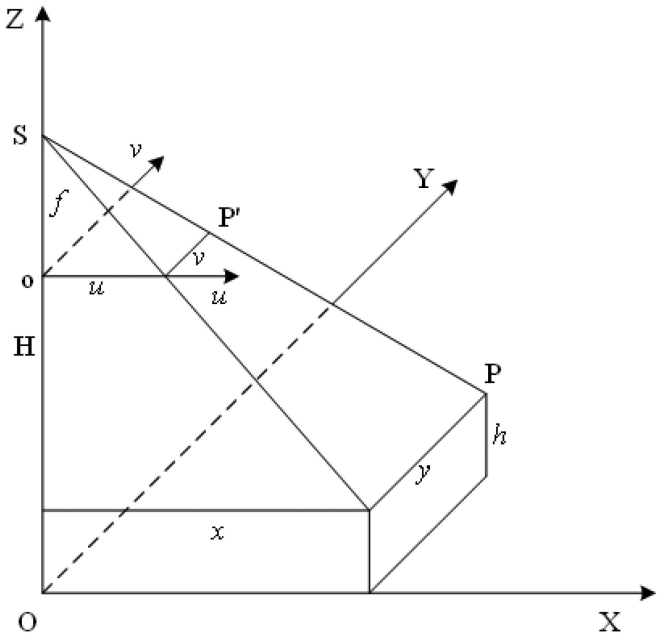


Fig. 7. The imaging principle of vertical photography at a fixed altitude. *O-XYZ* represents the real-world coordinate system and *u-o-v* represents the image coordinate system. **P** is the ground object, and **P′** is the corresponding image point.

Due to the fact that UAV photographs from overhead and there is an occlusion among pedestrians, it is difficult to obtain the image coordinate of the foot of each pedestrian. Therefore, it is inappropriate to use the plane of pedestrian foot as the plane to be transformed. Instead, the head of each pedestrian is easy to be recognized whose image coordinate can be obtained through pedestrian detection method. Therefore, we establish the coordinate transformation relationship between the plane of pedestrian's head and the real-world plane. Here, we assume that the pedestrian's head is in the same plane (About 1.7 m above the ground).

Vertical images. The vertical imaging principle can be simplified as shown in Fig. 7 where *O-XYZ* represents the real-world coordinate system and *u-o-v* represents the image coordinate

system. $\mathbf{P}$ is the real-world coordinate point, and $\mathbf{P}'$ is the corresponding image point. *f* is the focal length of the camera and *H* is the distance between the focus of the camera and the ground.

Assuming that $\mathbf{P}_i^{rw} = [x_i, y_i, 1]$ is the homogeneous representation of the 2D plane real-word coordinate of each head, and $\mathbf{P}_i^{v} = [u_i, v_i, 1]$ is the homogeneous representation of the corresponding vertical image coordinate. The following formula can be derived from the vertical imaging principle.

$$\frac{x_i}{u_i} = \frac{y_i}{v_i} = \lambda \tag{11}$$

The λ refers to the ratio of pixel to meter. The mapping from $\mathbf{P}_i^{v}$ to $\mathbf{P}_i^{rw}$ can be expressed as

$$\begin{bmatrix} x_i \\ y_i \\ 1 \end{bmatrix} = \begin{bmatrix} \lambda & 0 & 0 \\ 0 & \lambda & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} u_i \\ v_i \\ 1 \end{bmatrix} \tag{12}$$

Tilted images. Tilted image is more complex than vertical image and can be hardly transformed to the real-world coordinate directly. However, according to the research in [66], there is a homography between two images of the same area taken by the same camera at different angles or positions. In other words, there is a homography between two planes of the same region corresponding to the tilted image and the vertical image, which can be transformed by homography transformation. Therefore to solve this task, we first transform the tilted image to the vertical image by a homography matrix, and then transform the vertical image to the real-world coordinate based on the transformation principle of vertical image to real-world coordinate, as shown in Fig. 10.

A homography is an invertible mapping of points and lines on the projective plane. Assuming $\mathbf{P}_i^{t} = [u_i', v_i', 1]$ is the homogeneous representation of the titled image coordinate point. The transformation relationship between $\mathbf{P}_i^{v}$ and $\mathbf{P}_i^{t}$ can be represented by the following formula:

$$\mathbf{P}_i^{v} = \mathbf{H}\mathbf{P}_i^{t} \tag{13}$$

where $\mathbf{H}$ is a homography matrix, and can be represented as:

$$\mathbf{H} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & 1 \end{bmatrix} \tag{14}$$

As $\mathbf{H}$ has eight unknowns, at least four pairs of noncollinear titled image points and vertical image points are required to calculate the parameters of $\mathbf{H}$. Once the $\mathbf{H}$ is determined, the coordinates of any point in titled image coordinate can be projected into the vertical image coordinate.

$$\begin{bmatrix} u_i \\ v_i \\ 1 \end{bmatrix} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & 1 \end{bmatrix} \begin{bmatrix} u_i' \\ v_i' \\ 1 \end{bmatrix} \tag{15}$$

According to Eq. (12) and Eq. (15), the transformation of tilted image coordinate to real-world coordinate can be realized by

$$\begin{bmatrix} x_i \\ y_i \\ 1 \end{bmatrix} = \begin{bmatrix} \lambda & 0 & 0 \\ 0 & \lambda & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & 1 \end{bmatrix} \begin{bmatrix} u_i' \\ v_i' \\ 1 \end{bmatrix} \tag{16}$$

2) Social Distancing Monitoring: The image coordinate of each head $\mathbf{P}_i'(u_i, v_i)$ can be derived from the coordinate of bounding box of head that is predicted by our proposed head detection method. Then the corresponding real-word coordinate $\mathbf{P}_i(x_i, y_i)$ can be calculated by Eq. (12) or Eq. (16). Based on the real-word coordinate $\mathbf{P}_i(x_i, y_i)$, the distance $d_{i,j}$ for pedestrian *i* and *j* can be obtained by pairwise *L*2 norm between vector $\mathbf{P}_i$ and $\mathbf{P}_j$:

$$d_{i,j} = \| \mathbf{P}_i - \mathbf{P}_j \|_2 \tag{17}$$

where $i, j \in \{1, 2, \ldots, n\}$, *n* is the number of detected pedestrians.

All the inter-pedestrian distance *D* can be represented by an upper triangular matrix of distance:

$$D = \begin{bmatrix} 0 & d_{1,2} & \cdots & d_{1,j} & \cdots & d_{1,n} \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & d_{i,j} & \cdots & d_{i,n} \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}_{n \times n} \tag{18}$$
$$i \in \{1, 2, \ldots, n-1\}, j > i, \in \{2, \ldots, n\}$$

Because the safe social distancing is about 2 meters away, we set the threshold for social distancing as $d_c = 2$. We use the total number of social distancing violation *v* to measure the condition of social distancing of pedestrian in a scene, which can be calculated by

$$v = \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} f(d_{i,j}) \tag{19}$$

where, $f(d_{i,j}) = 1$ if $d_{i,j} < d_c$, otherwise 0.

IV. Experimental Results

A. Human Head Datasets

1) Video Surveillance Datasets: Since the UAV image is similar to the video surveillance data, to alleviate the difficulty of insufficient UAV training samples, we adopted the common head dataset under video surveillance for experiments firstly, including Brainwash [44], SCUT-HEAD [45] and our FerryHead. They are all obtained from video surveillance and are basically similar to UAV image. The samples from these datasets can be seen in Fig. 8, which show the following characteristics: 1) the size of human head is small, 2) they are from diversified scenes with various viewpoints, 3) there exists occlusion, 4) many pedestrians wear hats and helmets, and 5) some scenes show complex backgrounds. Based on these characteristics, we can draw a conclusion that these samples include almost the heads in daily scenes. In other words, these three datasets make up a representative head dataset.

Brainwash is a large dataset of human heads, derived from surveillance video footage from a coffee shop. The Brainwash dataset contains 11 917 images with 91 146 annotated heads, which is divided into three parts. The training set includes 10 917 images with 82 906 annotations. The validation set includes

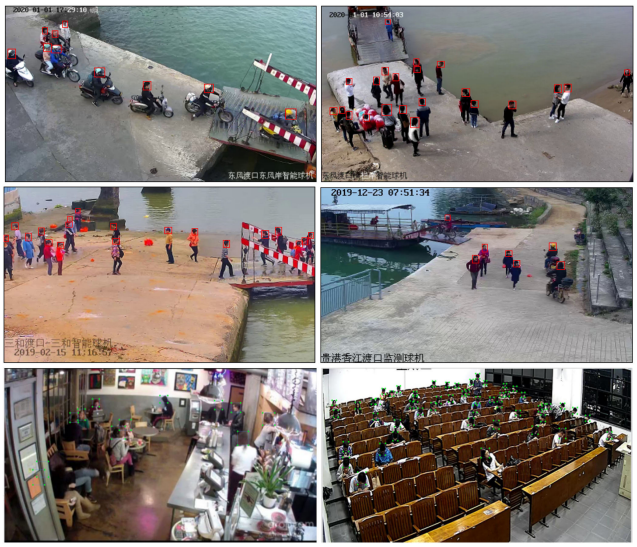


Fig. 8. The samples from a merged human head dataset (Merge-Head).

TABLE I
THE DETAIL INFORMATION OF HUMAN HEAD DATASETS

Dataset	Training set	Validation set	Testing set	Overall
Brainwash	10917	500	500	11917
SCUT-HEAD-PartA	1100	400	500	2000
FerryHead	3923	440	487	4850
Merge-Head	15940	1340	1487	18767

500 images with 3318 annotations. The testing set includes 500 images with 4922 annotations.

SCUT-HEAD is a common human head dataset, including PartA and PartB that are from monitor videos of classrooms in a university and images of Internet respectively. In this paper, we just use the PartA whose samples are similar to the UAV image. The SCUT-HEAD dataset contains 2000 images with 67 321 annotated heads. The number of training set, validation set, and testing set are respectively 1100, 400, 500.

Ferryhead is a universal human heads dataset that we built, whose samples are captured form the video surveillance of rural ferries in China. The FerryHead dataset has 4850 images in 8 scenes, with a total number of 27 289 labeled heads, containing heads from different scales, directions and viewpoints.

We merge these three datasets to obtain a large and universal human head dataset, call Merge-Head. The detail information is shown in Table I.

2) UAV Datasets: We built a small human head dataset, called UAV-Head, whose samples were captured from UAV images, containing both tilted images and vertical images. The UAV-Head contains 745 images in size of 1920×1080 pixels, with 564 images for training and 181 images for validation. The UAV-Head dataset is used to fine-tune these detection models that have been trained on video surveillance dataset (Merge-Head dataset).

B. Evaluation Metrics

To evaluate the detection speed, one of the most common metrics is frames per second (FPS), which means the number of images that each detection algorithm can process with the specified hardware.

We regard the IoU value of the predicted bounding boxes more than 0.5 and the correct classification results as the true results, and other predicted results as false results. The predicted results can be divided into four categories: True Positive (TP), False Positive (FP), False Negative (FN) and True Negative (TN). The precision is defined as $precision = TP/(TP + FP)$, and recall is defined as $recall = TP/(TP + FN)$. Average precision (AP) is used to measure the accuracy of the detection model, which is the average of precision values under different recall values.

C. Implementation Details

These experiments run at a desktop with 3. 60 GHz Intel Core i7-7820X CPU, 32 GB RAM, Ubuntu 18. 04 systems. Our proposed method and the baseline methods are implemented by PyTorch 1.5.1 library with Python 3. 6. 9, accelerating by a NVIDIA GTX 2080Ti GPU with 12 GB GPU memory, CUDA 10.1, and CUDNN 7.5.

1) Detection Details: When detecting small targets, the size of normalized input images and anchor box have certain influence on the detection results. Therefore, we chose the larger input image size and the smaller anchor box compared to YOLOv3. We resized all images to 608×608. As the feature information will be lost as the size of feature maps decrease in down-sampling operation, we used larger size of feature maps for prediction, which are respectively to $19 \times 19, 38 \times 38, 76 \times 76$.

In PeleeNet module, the number of dense layers in four dense-blok are set to 3, 4, 8, 6 respectively. The growth rate is set to 32. We adopt the deconvolution with kernel size of 4×4 and stride of 2 to carry out up-sampling in multi-scale feature fusion processing. The learning rate is set to 0.001, the momentum parameter is chosen as 0.9, and the weight decay is 0.0005.

Our method and the baseline methods are firstly trained by Merge-Head dataset. Then we used a small UAV dataset to fine-tune these models.

2) Calibration Test: We calibrated on a square where the plane of pedestrian head in the image is calibrated with the plane in the real world. Five pedestrians with an average height of 1.7 m were selected to be calibration points and stood at different positions to obtain vertical and tilted images of the UAV. Here, the focal length of the camera (f) is 24 mm, the height indicating the distance between the camera and ground (H) is 14 m, and the tilted angle is $45°$. The vertical image calibration is shown in the Fig. 9, where the pixel distance between the pedestrian's head in the image and the corresponding actual distance are obtained by actual measurement. Thus, we can gain the ratio of pixel to meter that is from the plane of pedestrian's head in image to real world, and that is $\lambda = 0.00783$ m/pixel. Fig. 10 shows the calibration from the tilted image to the real world. First, 5 pairs of homonymic points (pedestrian's head) are selected between the tilted image and the vertical image and corresponding pixel coordinates are obtained. Through these 5 pairs of homonymic point pairs, **H**, the homography matrix for the transformation from the tilted image to the vertical image, can be calculated.

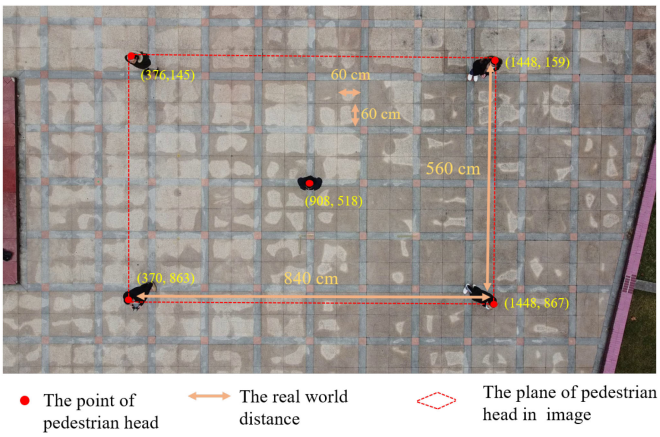


Fig. 9. The calibration of vertical images.

Note that **H** here represents the transform relationship between the planes of pedestrian's head in tilted images and vertical images. Then, the tilted image can be transformed to the real world using the vertical image calibration results.

After the calibration, the vertical images and the tilted images were acquired under the condition of calibration. The pedestrian's heads in images were detected by our pedestrian detection algorithm, then the social distance was calculated.

D. Results and Analysis

1) Detection Results on Video Surveillance Datasets: In this part, we compare our method with two state-of-the-art object detection models (SSD [15] and YOLOv3 [16]), both of which show a good performance on Pascal VOC 2007 dataset and can achieve real-time end-to-end object detection. Here, the SSD300 model and SSD512 model are both based on VGG16, with the same architecture, and the only difference is the input image size (300×300 and 512×512). The SSD320 utilizes the MobileNet_v2 [39] as backbone, which is a lightweight network, with the input image size of 320×320. The YOLOv3 uses the Darknet-53 to extract feature information and have three prediction layers with different scales. The YOLOv3-Tiny is the reduced version of YOLOv3, which is a lightweight network with smaller size of model and fast detection speed. YOLOv3, YOLOv3-Tiny and our method have the same size of input image (608×608). All methods are trained and tested on Merge-Head dataset with one NVIDIA GTX 2080Ti GPU. The precision-recall curves of these methods on Merge-Head dataset are shown in Fig. 11, from which we can see that our proposed method gives the highest average precision (AP) than others.

Speed and accuracy analysis. The speed and accuracy of each methods can be seen in Table II. Our model is a lightweight CNN object detection network, whose size (8.30 M) is only larger than SDD320 (3.02 M). Compared with the two lightweight models SSD320 and YOLOv3-Tiny, the speed of our method is 76 FPS, which is slower than SSD320 (126 FPS) and YOLOv3-Tiny (261 FPS) but faster than the other methods. However, our method can achieve highest AP. The AP of our method is 92.22%, which is higher than that of YOLOv3 by 1.2% and higher than that of YOLOv3-Tiny by 5.25% and much higher than that of SSD models. The size of input image has a certain impact on the detection results. Comparing the AP of SSD512 (78.74%) and SSD 300 (44. 78%), we can know that the larger the input size, the higher the AP, but the lower speed.

There is a trade-off problem between accuracy and speed. In practical applications, we give priority to accuracy and then consider its speed and model size. The algorithm with high accuracy, fast speed and small size is the best. The experimental results can explain that our method enjoys the best performance and can achieve real-time and accurate head detection.

TABLE II
THE COMPARISON RESULTS ON MERGE-HEAD DATASET OF DIFFERENT METHODS

Model	Backbone	Input size	Model size	AP (%)	FPS
SSD512	VGG16	512×512	24.38M	78.74	58
SSD300	VGG16	300×300	23.74M	44.78	105
SSD320	MobileNet_V2	320×320	**3.02M**	17.15	126
YOLOv3	Darknet-53	608×608	61.52M	91.02	64
YOLOv3-Tiny	Darknet-53-Tiny	608×608	9.00M	86.97	**261**
Ours	PeleeNet	608×608	8.30M	**92.22**	76

TABLE III
THE ABLATION EXPERIMENTAL RESULTS BASED ON MERGE-HEAD DATASET

Model	Backbone	Input size	Multi-scale	Spatial attention	AP (%)
PeleeNet	PeleeNet	608×608	×	×	15.40
PeleeNet_M	PeleeNet	608×608	√	×	91.12
Ours	PeleeNet	608×608	√	√	92.22

Visualization results on Merge-Head dataset. The test results of our method in Merge-Head are shown in Fig. 12. The test scenarios include rural ferries, cafe, classrooms, with diversified viewpoints, illuminations, and scales. In the video surveillance of the rural ferries, people wear all kinds of hats, carry various luggage and take different travel tools, which increases the difficulty of pedestrian detection. However, our method can almost accurately detect all the people and can distinguish the human heads from background. In the cafe, the human head with various scales and different illumination can also be detected accurately by our method. The classroom is a crowded scene with numerous people in a room, where the human heads are small size and blocked by each other. It is a hard task to accurately detect small-size object in a crowded scene, but our method can achieve it and detect each human head in such a complicated scene. Generally speaking, our method shows a great performance for human head detection with a high accuracy in diversified scenarios.

Ablation analysis. In order to verify the effectiveness of our network, we made specific analysis from the following two aspects: multi-scale feature and spatial attention. The experimental results are based on Merge-Head dataset, which can be seen in Table III. Fig. 13 shows the visualization of ablation experiments.

1) Multi-scale features. We use the PeleeNet as the baseline which only uses one scale for prediction. We defined PeleeNet_M as the network that uses the PeleeNet as backbone and has three scales of features for prediction. The AP of PeleeNet is 15.40%, which is much low. While the PeleeNet_M

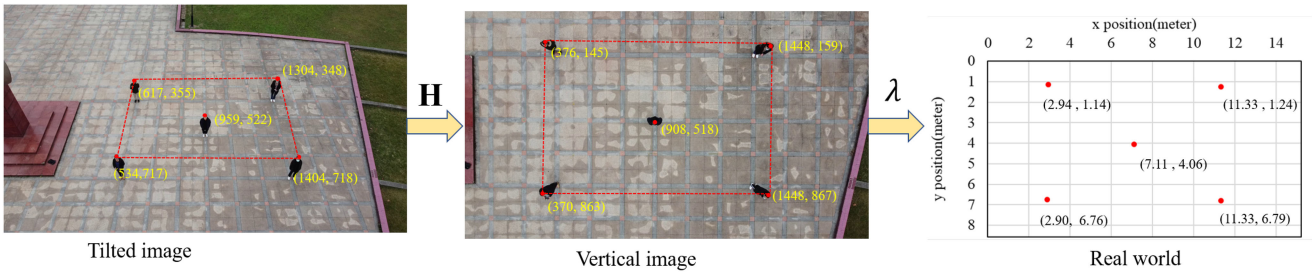


Fig. 10. The calibration of tilted images.

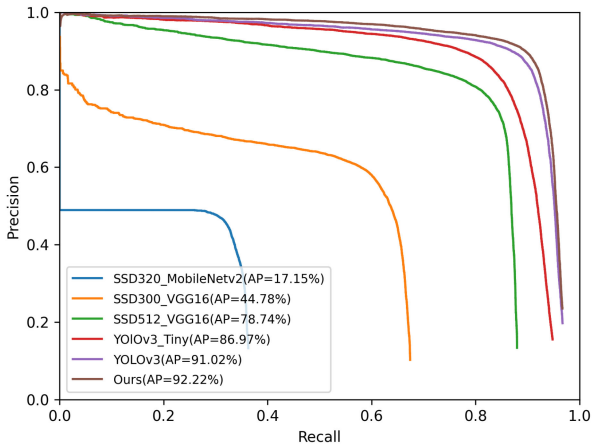


Fig. 11. The precision-recall curves of different algorithms.

TABLE IV
THE COMPARISON RESULTS ON UAV-HEAD DATASET OF DIFFERENT METHODS

Model	AP(%)	FPS
SSD512	80.2	48
YOLOv3-Tiny	82.3	250
YOLOv3	87.9	66
Ours	88.5	75

can achieve 91.12% AP, which is significantly improved compared with PeleeNet. The detection results in Fig. 13(a) shows that there are numerous miss detections compared to that of in Fig. 13(b). The results between PeleeNet and PeleeNet_M can fully confirm that the network only uses one scale can hardly detect human heads but multi-scale features are much useful for improving the accuracy of human head detection.

2) Spatial attention. The AP of PeleeNet_M is 91.12%, while ours has a further improvement of 1.10% compared with PeleeNet_M. We use three scales of features for prediction in our network as same as PeleeNet_M, but we apply spatial attention module to each feature while PeleeNet_M does not. And the visualized results in Fig. 13 show that there are some false detections by PeleeNet_M. The results indicate that spatial attention module indeed improves the accuracy of human head detection. In short, both multi-scale feature and spatial attention lead to a significant improvement to the accuracy of human head detection, and our proposed human head detection method outperforms other methods.

2) Detection Results on UAV Images: The UAV image is very similar to the video surveillance image. However, the scenarios of UAV images are still more complex than those under video surveillance. Our method works very well on video surveillance data. Whether can it also work well on low-altitude UAV images? To validate it, we conducted a detection experiment on UAV-Head dataset using our method and comparison methods. The UAV-Head dataset contains vertical images and tilted images.

The performance of object detection based on deep learning much depends on training data. The data sources, objects, and scenarios of the dataset are different depending on the requirements. There is currently no dataset that meets the requirements for all data sources, objects and scenarios. Therefore, according to requirements, corresponding datasets should be made for specific data sources, objects and scenarios. However, it takes a lot of manpower and material resources to make such a dataset. In order to save resources, according to the idea of transfer learning [67], we used a small UAV dataset to fine-tune these detection models trained on Merge-Head dataset and conducted a comparison experiment. Using the existing datasets from different sources for pre-training and then fine-tuning the network model on the small dataset in a specific scenario can save a lot of resources and obtain an applicable network model.

The test results on UAV-Head dataset based on our method and comparison methods are shown in Fig. 14 and Table IV. The results in Table IV show that our pedestrian detection method achieves 88.5% AP that is higher than the comparison methods: YOLOv3 (87.9%), YOLOv3-Tiny (82.3%), SSD512 (80.2%). The FPS of our method is 75. It indicates that our pedestrian detection method can also achieve real time and accurate detection on UAV images.

We visualized the detection results on UAV images (including vertical images and tilted images), as shown in Fig. 14. There are some missed and false detections in the results of comparison methods while our method can achieve high-precision detection of each pedestrian in both vertical images and tilted images. These results can explain that our method enjoys the best performance on UAV pedestrian detection.

3) Social Distancing Monitoring: Social distancing can be measure by calculating inter-pedestrian distance. We have experimented with social distancing monitoring on UAV images. Since we have conducted a calibration test and obtained the transformation matrix from image to real-world coordinate, the social distance in UAV can be easily calculated. Fig. 15

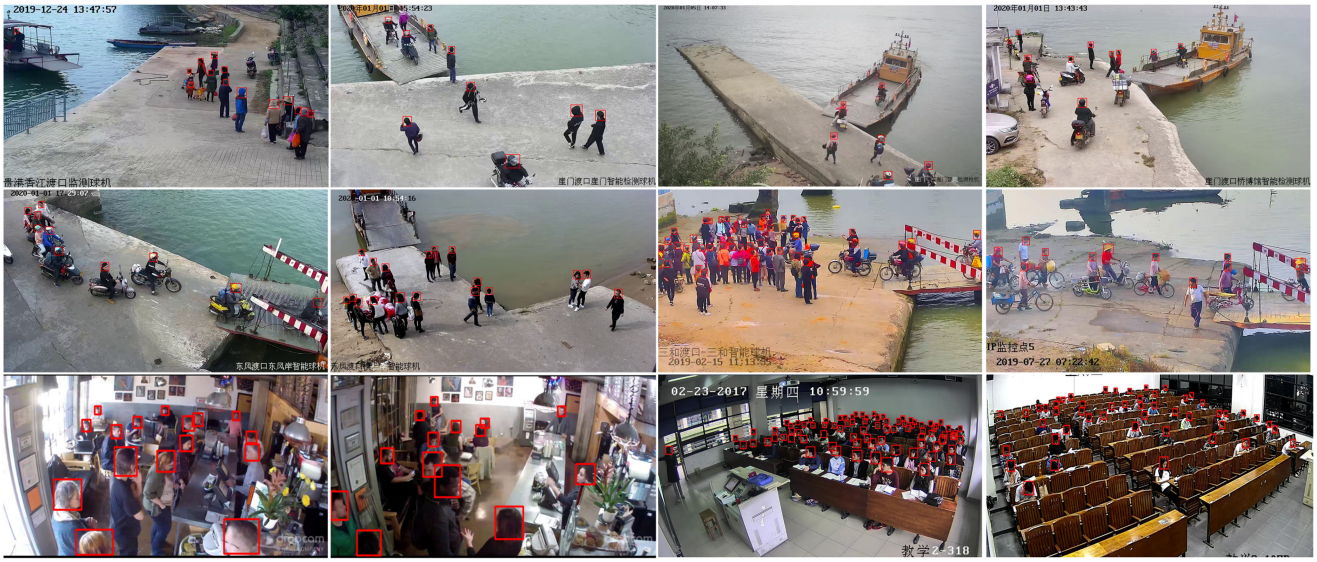


Fig. 12. The visualization results of our method on Merge-Head dataset.

Fig. 13. The visualization of ablation experiments. (a), (b) and (c) are the detection results of PeleeNet, PeleeNet_M and Ours respectively.

TABLE V
THE TILTED IMAGE AND REAL-WORLD COORDINATE OF EACH PEDESTRIAN

Pedestrian	u (pixel)	v (pixel)	X (meter)	Y (meter)
P1	1077	518	12.416	8.055
P2	1005	180	11.874	1.481
P3	290	258	2.163	2.952
P4	800	979	9.686	13.548
P5	851	953	10.115	13.3
P6	1027	529	11.866	8.214
P7	1078	166	12.899	1.177
P8	1302	205	15.799	2.241
P9	956	713	11.06	10.707
P10	1001	717	11.496	10.758
P11	1520	248	18.369	3.302
P12	916	739	10.671	11.019

shows the pedestrian detection results on UAV image by using our proposed method and the calculated corresponding coordinates in the real world. Taking the tilted image for example, the concrete corresponding coordinate of each pedestrian in Fig. 15(c) and Fig. 15(d) can be seen in Table V. **Pi** indicates the ID of each detected pedestrian. (u, v) represents the image pixel coordinate of each pedestrian. (X, Y) denotes the coordinate of each pedestrian in the real world that was calculated by projection transformation matrix **H** and the ratio of pixel to meter λ. The inter-pedestrian distance upper triangular matrix D is shown in Table VI. As seen from the Table VI, there are 6 couple of pedestrians less than 2 meters apart, namely $d_{1,6}, d_{2,7}, d_{4,5}, d_{9,10}, d_{9,12}, d_{10,12}$. The total number of social distancing violation in this scene is $v = 6$. That means there are 6 couple of pedestrians in a close social distancing.

We conducted experiments of social distancing monitoring on numerous UAV images, and the monitoring results are shown in Fig. 16. We display the total number of pedestrians in each scene and calculate the value of v. If $v = 0$, it means that there is no pedestrian with close social distancing in the scene and the pedestrian keep a safe social distancing, then the Normal is showed. If $v > 0$, it means there is a pedestrian with a close social distancing in the scene, then the Warning is shown. Based on our social distancing monitoring algorithm, the UAV can quickly detect the crowd.

Furthermore, we have evaluated the accuracy of social distance estimation with four different pedestrian position patterns. Fig. 17 is the results of the social distance monitoring on tilted UAV images using our method. We compared the detected social distance with the ground truth and calculated the absolute errors. Taking the Fig. 17(a) and Fig. 17(b) for example, the results are shown in Table VII and Table VIII, from which we can calculate the mean absolute error respectively. The mean absolute errors in Fig. 17(a) and Fig. 17(b) are respectively 0.1099 m and 0.1050 m. In the same way, the mean absolute errors in Fig. 17(c) and Fig. 17(d) are respectively 0.1061 m and 0.1084 m. Finally, the mean absolute error in these four different pedestrian position patterns is 0.1073 m. Considering that each pedestrian has a volume, we treated each pedestrian as a point in our experiment, which is an ideal state. Therefore, in the real world, the mean absolute error of 0.1073 m is within the allowable error range. Form the above analysis, we can make a conclusion that our method has a good performance in social distancing estimation.

Although our method has a good performance in social distancing estimation, there are some limitations. On the one hand, when UAVs fly in the air, they are inevitably disturbed by the wind. The issue of instability caused by the wind is not solved by algorithms at present, generally through hardware devices to

Fig. 14. The visualization detection results on UAV-Head dataset (including tilted images and vertical images).

TABLE VI
THE INTER-PEDESTRIAN DISTANCE UPPER TRIANGULAR MATRIX D (METER)

Pedestrian	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	P11	P12
P1	0	6.596	11.453	6.134	5.728	**0.573**	6.895	6.727	2.979	2.855	7.618	3.44
P2	0	0	9.822	12.264	11.949	6.733	**1.069**	3.998	9.262	9.285	6.745	9.614
P3	0	0	0	12.995	13.050	11.038	10.882	13.655	11.802	12.167	16.210	11.724
P4	0	0	0	0	**0.496**	5.762	12.781	12.854	3.156	3.326	13.430	2.714
P5	0	0	0	0	0	5.379	12.439	12.434	2.760	2.893	12.965	2.348
P6	0	0	0	0	0	0	7.112	7.152	2.620	2.571	8.150	3.049
P7	0	0	0	0	0	0	0	3.089	9.706	9.683	5.868	10.091
P8	0	0	0	0	0	0	0	0	9.702	9.542	2.780	10.166
P9	0	0	0	0	0	0	0	0	0	**0.439**	10.405	**0.499**
P10	0	0	0	0	0	0	0	0	0	0	10.141	**0.865**
P11	0	0	0	0	0	0	0	0	0	0	0	10.900
P12	0	0	0	0	0	0	0	0	0	0	0	0

reduce the interference as much as possible. With the rapid development of UAV technology, more and more UAVs can withstand the instability caused by a certain intensity wind. On the other hand, in our study, we assume that the surface is flat. This is the simplest situation. However, sometimes the surface is not always flat or there are different levels in one scene. Our social distancing estimation method is unsuitable in this situation and it is a huge challenge to make a precise social distancing estimation. In addition, there is no similar solution so far. In our future work, this will be a direction worthy of further study.

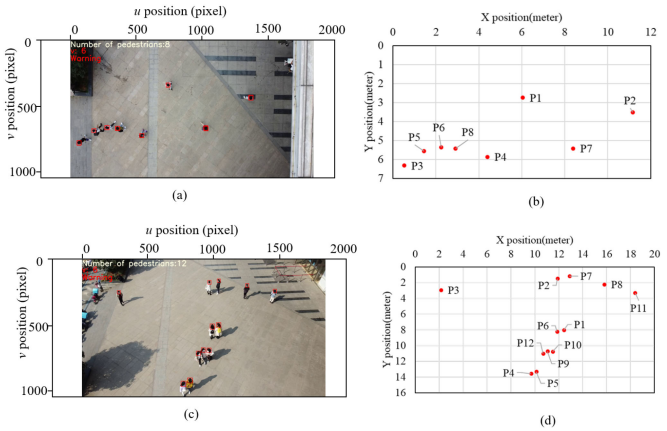


Fig. 15. The pedestrian detection results on UAV images and the mapped corresponding coordinates in the real world.

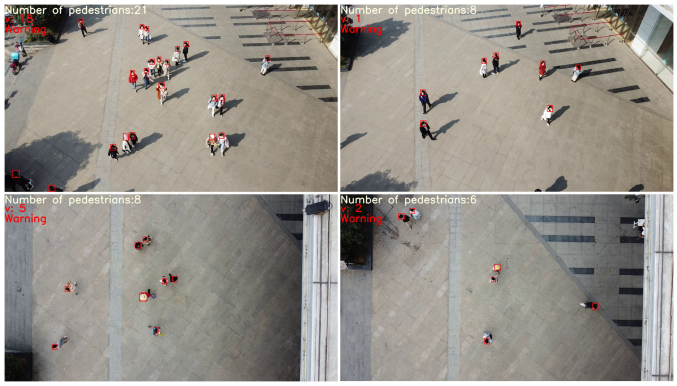


Fig. 16. The results of social distancing monitoring on UAV images (including tilted images and vertical images).

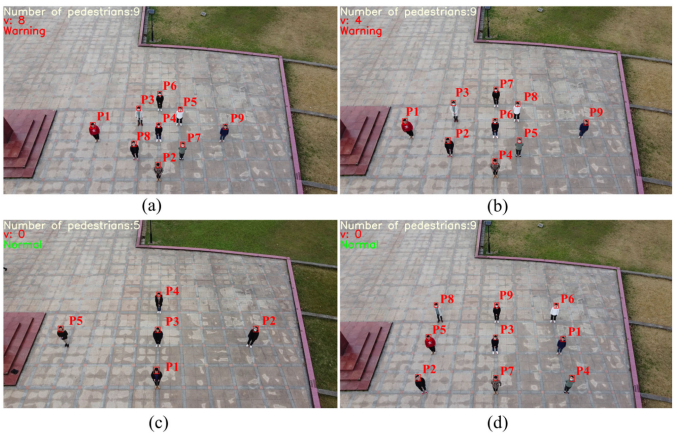


Fig. 17. The results of social distancing monitoring on tilted images with four different pedestrian position patterns.

V. Conclusion

In this paper, we have proposed a lightweight pedestrian detection network which can accurately detect pedestrian by human head in real time, and then calculated the social distancing of each pedestrian on UAV images. Our pedestrian detection network consists of three parts, PeleeNet, multi-scale spatial attention module and detection layer. In order to explore the features of small-size object like human head, we fuse three scales of feature maps (19×19, 38×38, 76×76) by deconvolution and concatenation. The spatial attention module is particularly used to enhance the feature information and ignore the irrelevant information. Then the location of human head is predicted in detection layer.

We compared our method with the state-of-the-art object detection methods (SSD model and YOLOv3 model) on a merged human head dataset. The experimental results show that our method achieves 92.22% AP and 76 FPS, which turns out accurate and real-time detection in actual applications. Especially, the ablation experiments show that multi-scale feature and spatial attention can substantially improve the performance of pedestrian detection. The test results on UAV-Head dataset show that our method can also achieve high precision pedestrian detection on UAV images with 88.5% AP and 75 FPS. The visualization results of our method on UAV images also show that our method can detect each individual with different viewpoints, illuminations and scales in various scenes. In addition, we conducted a precision calibration test to obtain the transformation matrix from tilted image and vertical image to real-world coordinate. Based on the accurate pedestrian detection and the map relationship from image to real-world coordinate system, the social distancing monitoring is achieved reliably, enabling an automatic distance-sensing approach for preventing COVID-19.

TABLE VII
The Comparison Results Between the Detected Social Distance and the Ground Truth in Fig. 17 (a) (meter)

Inter-pedestrian distance	Ground truth	Distance estimated from our method	Absolute error
$d_{3,4}$	1.980	1.905	0.075
$d_{4,5}$	1.980	1.855	0.125
$d_{3,6}$	1.980	1.877	0.103
$d_{5,6}$	1.980	1.860	0.120
$d_{2,7}$	1.980	1.897	0.083
$d_{4,7}$	1.980	1.859	0.121
$d_{2,8}$	1.980	1.855	0.125
$d_{4,8}$	1.980	1.853	0.127

TABLE VIII
The Comparison Results Between the Detected Social Distance and the Ground Truth in Fig. 17 (b) (meter)

Inter-pedestrian distance	Ground truth	Distance estimated from our method	Absolute error
$d_{4,5}$	1.980	1.831	0.149
$d_{5,6}$	1.980	1.898	0.082
$d_{6,8}$	1.980	1.902	0.078
$d_{7,8}$	1.980	1.869	0.111

References

[1] J.F-W. Chan *et al.*, "A familial cluster of pneumonia associated with the 2019 novel coronavirus indicating person-to-person transmission: A study of a family cluster," *Lancet*, vol. 395, no. 10223, pp. 514–523, 2020.

[2] J. Riou and C. L. Althaus, "Pattern of early human-to-human transmission of wuhan 2019 novel coronavirus (2019-nCoV), december 2019 to january 2020," *Eurosurveillance*, vol. 25, no. 4, p. 2000058, 2020.

[3] Q. Li *et al.*, "Early transmission dynamics in Wuhan, China, of novel coronavirus-infected pneumonia," *New England J. Med.*, 2020.

[4] C. Courtemanche, J. Garuccio, A. Le, J. Pinkston, and A. Yelowitz, "Strong Social Distancing Measures in the United States Reduced the covid-19 Growth Rate: Study Evaluates the Impact of Social Distancing Measures on the Growth Rate of Confirmed covid-19 Cases Across the United States." *Health Affairs*, pp. 10–1377, 2020.

[5] N. S. Punn, S. K. Sonbhadra, and S. Agarwal, "Monitoring covid-19 social distancing with person detection and tracking via fine-tuned YOLO V3 and deepsort techniques," 2020 *arXiv:2005.01385*.

[6] D. Yang, E. Yurtsever, V. Renganathan, K. A. Redmill, and Ü. Özgüner, "A vision-based social distance and critical density detection system for covid-19," , 2020, *arXiv:2007.03578* .

[7] P. Tang *et al.*, "PCL: Proposal Cluster Learning for Weakly Supervised Object Detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 42, no. 1, pp. 176–191, Jan. 2020.

[8] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Region-based convolutional networks for accurate object detection and segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 38, no. 1, pp. 142–158, Jan. 2016.

[9] R. Girshick, "Fast R-CNN," in *Proc. IEEE Int. Conf. computer vision*, 2015, pp. 1440–1448.

[10] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2015, pp. 91–99.

[11] J. Dai, Y. Li, K. He, and J. Sun, "R-FCN: Object detection via region-based fully convolutional networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2016, pp. 379–387.

[12] J. Li *et al.*, "Multistage object detection with group recursive learning," *IEEE Trans. Multimedia*, vol. 20, no. 7, pp. 1645–1655, Jul. 2018.

[13] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," in *Proc. IEEE Int. Conf. Computer vision*, 2017, pp. 2980–2988.

[14] Z. Shao, L. Wang, Z. Wang, W. Du, and W. Wu, "Saliency-aware convolution neural network for ship detection in surveillance video," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 30, no. 3, pp. 781–794, Mar. 2020.

[15] W. Liu *et al.*, "Single shot multibox detector," in *Proc. Eur. Conf. Computer vision*. Springer, 2016, pp. 21–37.

[16] J. Redmon and A. Farhadi, "YOLOV3: An Incremental Improvement," 2018, *arXiv:1804.02767* .

[17] J. Redmon, and A. Farhadi, "YOLO9000: Better, faster, stronger," in *Proc. IEEE Conf. Computer vision Pattern Recognit.*, 2017, pp. 7263–7271.

[18] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proc. IEEE Conf. Computer vision Pattern Recognit.*, 2016, pp. 779–788.

[19] Z. Shao, W. Wu, Z. Wang, W. Du, and C. Li, "Seaships: A large-scale precisely annotated dataset for ship detection," *IEEE Trans. Multimedia*, vol. 20, no. 10, pp. 2593–2604, Oct. 2018.

[20] J. Li *et al.*, "Attentive contexts for object detection," *IEEE Trans. Multimedia*, vol. 19, no. 5, pp. 944–954, may 2017.

[21] S. Zhang, L. Wen, X. Bian, Z. Lei, and S. Z. Li, "Occlusion-aware r-cnn: Detecting pedestrians in a crowd," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 637–653.

[22] W. Ouyang and X. Wang, "Joint deep learning for pedestrian detection," in *Proc. IEEE international conference computer vision*, 2013, pp. 2056–2063.

[23] S. Wang, J. Zhang, and Z. Miao, "A new edge feature for head-shoulder detection," in *Proc. IEEE Int. Conf. Image Process.*IEEE, 2013, pp. 2822–2826.

[24] C. Zeng and H. Ma, "Robust head-shoulder detection by PCA-based multi-level hog-lbp detector for people counting," in *2010 20th Int. Conf. Pattern Recognit*. IEEE, 2010, pp. 2069–2072.

[25] W. Li *et al.*, "HeadNet: An end-to-end adaptive relational network for head detection," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 30, no. 2, pp. 482–494, Feb. 2020.

[26] A. Vora and V. Chilaka, "FCHD: Fast and accurate head detection in crowded scenes,"2018, *arXiv:1809.08766*.

[27] Y. Wang, Y. Yin, W. Wu, S. Sun, and X. Wang, "Robust person head detection based on multi-scale representation fusion of deep convolution neural network," in *IEEE Int. Conf. Robot. Biomimetics (ROBIO)*. IEEE, 2017, pp. 296–301.

[28] J. Jeong, H. Park, and N. Kwak, "Enhancement of SSD by concatenating feature maps for object detection," 2018, *arXiv:1705.09587*.

[29] Z. Li and F. Zhou, "FSSD: Feature fusion single shot multibox detector," 2017, *arXiv:1712.00960*.

[30] C.-Y. Fu, W. Liu, A. Ranga, A. Tyagi, and A. C. Berg, "DSSD: Deconvolutional single shot detector," 2017, *arXiv:1701.06659*.

[31] J. Ma, X. Wang, and J. Jiang, "Image superresolution via dense discriminative network," *IEEE Trans. Ind. Electron.*, vol. 67, no. 7, pp. 5687–5695, Jul. 2020.

[32] K. Jiang *et al.*, "Edge-enhanced gan for remote sensing image superresolution," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 8, pp. 5799–5812, Aug. 2019.

[33] Y. Bai, Y. Zhang, M. Ding, and B. Ghanem, "SOD-MTGAN: Small object detection via multi-task generative adversarial network," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 206–221.

[34] P. Yi, Z. Wang, K. Jiang, Z. Shao, and J. Ma, "Multi-temporal ultra dense memory network for video super-resolution," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 30, no. 8, pp. 2503–2516, Aug. 2020.

[35] L. Zhou, Z. Wang, Y. Luo, and Z. Xiong, "Separability and compactness network for image recognition and superresolution," *IEEE Trans. Neural Netw. learn. Syst.*, vol. 30, no. 11, pp. 3275–3286, Nov. 2019.

[36] H. Wang *et al.*, "Spatial attention for multi-scale feature refinement for object detection," in *Proc. IEEE Int. Conf. Comput. Vis. Workshops*, 2019, pp. 0–0.

[37] J. Chen, L. Wan, J. Zhu, G. Xu, and M. Deng, "Multi-scale spatial and channel-wise attention for improving object detection in remote sensing imagery," *IEEE Geosci. Remote Sens. Lett.*, vol. 17, no. 4, pp. 681–685, 2019.

[38] H. Zhao *et al.*, "PSANET: Point-wise spatial attention network for scene parsing," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 267–283.

[39] S. Woo, J. Park, J.-Y. Lee, and I. S. Kweon, "CBAM: Convolutional block attention module," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, Sep. 2018.

[40] F. Wang *et al.*, "Residual attention network for image classification," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017.

[41] A. G. Howard *et al.*, "MobileNets: Efficient convolutional neural networks for mobile vision applications," *arXiv preprint arXiv:1704.04861*, 2017.

[42] X. Zhang, X. Zhou, M. Lin, and J. Sun, "ShuffleNet: An extremely efficient convolutional neural network for mobile devices," in *Proc. IEEE Conf. Computer vision Pattern Recognit.*, 2018, pp. 6848–6856.

[43] R. J. Wang, X. Li, and C. X. Ling, "Pelee: A real-time object detection system on mobile devices," in *Proc. Adv. Neural Inf. Process. Syst.*, 2018, pp. 1963–1972.

[44] R. Stewart, M. Andriluka, and A. Y. Ng, "End-to-end people detection in crowded scenes," in *Proc. IEEE Conf. Computer vision Pattern Recognit.*, 2016, pp. 2325–2333.

[45] D. Peng *et al.*, "Detecting heads using feature refine net and cascaded multi-scale architecture," in *Proc. 24th Int. Conf. Pattern Recognit. (ICPR)*. IEEE, 2018, pp. 2528–2533.

[46] P. Zhu, L. Wen, X. Bian, H. Ling, and Q. Hu, "Vision meets drones: A challenge," 2018, *arXiv:1804.07437*.

[47] T. Moranduzzo and F. Melgani, "Detecting cars in UAV images with a catalog-based approach," *IEEE Trans. Geosci. remote sensing*, vol. 52, no. 10, pp. 6356–6367, Oct. 2014.

[48] M. A. Ma'Sum *et al.*, "Simulation of intelligent unmanned aerial vehicle (UAV) for military surveillance," in *Proc. Int. Conf. Adv. Comput. Sci. Inf. Syst. (ICACSIS)*. IEEE, 2013, pp. 161–166.

[49] W. Zhiqiang and L. Jun, "A review of object detection based on convolutional neural network," in *Proc. 36th Chin. Control Conf. (CCC)*. IEEE, 2017, pp. 11 104–11 109.

[50] J. Li *et al.*, "Scale-aware fast r-cnn for pedestrian detection," *IEEE Trans. Multimedia*, vol. 20, no. 4, pp. 985–996, Apr. 2018.

[51] S. Wang, J. Cheng, H. Liu, F. Wang, and H. Zhou, "Pedestrian detection via body part semantic and contextual information with dnn," *IEEE Trans. Multimedia*, vol. 20, no. 11, pp. 3148–3159, Nov. 2018.

[52] J. Zhang *et al.*, "Attribute-aware pedestrian detection in a crowd," *IEEE Trans. Multimedia*, pp. 1–1, 2020, doi: 10.1109/TMM.2020.3020691.

[53] W. G. Aguilar *et al.*, "Pedestrian detection for uavs using cascade classifiers and saliency maps," in *Proc. Int. Work-Conf. Artif. Neural Netw*. Springer, 2017, pp. 563–574.

[54] N. AlDahoul, A. Q. Md Sabri, and A. M. Mansoor, "Real-time human detection for aerial captured video sequences via deep models," *Comput. Intell. neurosci.*, vol. 2018, 2018.

[55] J. Zhang, X. Liang, M. Wang, L. Yang, and L. Zhuo, "Coarse-to-fine object detection in unmanned aerial vehicle imagery using lightweight convolutional neural network and deep motion saliency," *Neurocomputing*, vol. 398, pp. 555–565, 2020.

[56] R. Zhang, Z. Shao, X. Huang, J. Wang, and D. Li, "Object detection in UAV images via global density fused convolutional network," *Remote Sens.*, vol. 20, p. 31, 2020.

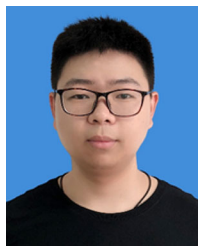
[57] Y. Ma, X. Wu, G. Yu, Y. Xu, and Y. Wang, "Pedestrian detection and tracking from low-resolution unmanned aerial vehicle thermal imagery," *Sensors*, vol. 16, no. 4, p. 446, 2016.
[58] T.-H. Vu, A. Osokin, and I. Laptev, "Context-aware cnns for person head detection," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2015, pp. 2893–2901.
[59] C. Gao, P. Li, Y. Zhang, J. Liu, and L. Wang, "People counting based on head detection combining adaboost and CNN in crowded surveillance environment," *Neurocomputing*, vol. 208, pp. 108–116, 2016.
[60] Y. Li, Y. Dou, X. Liu, and T. Li, "Localized region context and object feature fusion for people head detection," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*. IEEE, 2016, pp. 594–598.
[61] H. Larochelle and G. E. Hinton, "Learning to combine foveal glimpses with a third-order Boltzmann machine," in *Proc. Adv. Neural Inf. process. Syst.*, 2010, pp. 1243–1251.
[62] C. T. Nguyen *et al.*, "Enabling and emerging technologies for social distancing: A comprehensive survey," 2020, *arXiv:2005.02816*.
[63] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proc. IEEE Conf. Computer vision Pattern Recognit.*, 2017, pp. 4700–4708.
[64] H. Noh, S. Hong, and B. Han, "Learning deconvolution network for semantic segmentation," in *Proc. IEEE Int. Conf. Computer vision*, 2015, pp. 1520–1528.
[65] Q. Liu, W. Liu, Z. Lei, J. Wang, and Y. Liu, "A new approach to fast mosaic uav images," in *Proc. Int. Arch. Photogrammetry, Remote Sensing Spatial Inf. Sci.*, vol. 38, no. 1, 2011.
[66] E. Dubrofsky, "Homography estimation," *Diplomová práce. Vancouver: Univerzita Britské Kolumbie*, vol. 5, 2009.
[67] L. Torrey and J. Shavlik, "Transfer learning," in *Handbook of Research on Machine Learning Applications and Trends: Algorithms, Methods, and Techniques. IGI Global*, 2010, pp. 242–264.

Zhenfeng Shao received the Ph.D. degree in photogrammetry and remote sensing from Wuhan University, Wuhan, China, in 2004, working in the State Key Laboratory of Information Engineering in Surveying, Mapping and Remote Sensing (LIESMARS). Since 2009, he has been a Full Professor with LIESMARS, Wuhan University. He has authored or coauthored more than 60 peer-reviewed articles in international journals. His research interests include high-resolution image processing, pattern recognition, and urban remote sensing applications. Since 2019, he has been an Associate Editor for the PE&RS specializing in smart cities, photogrammetry and change detection. He was the recipient of the Talbert Abrams Award for the Best Paper in Image matching from the American Society for Photogrammetry and Remote Sensing in 2014 and the New Century Excellent Talents in University from the Ministry of Education of China in 2012.

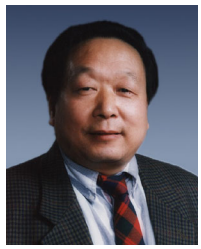
Gui Cheng received the B.S. degree in geographic information science from Chang'an University, Xi'an, China, in 2019. He is currently working toward the Ph.D. degree with the State Key Laboratory of Information Engineering in Surveying, Mapping and Remote Sensing, Wuhan University, Wuhan, China. His research interests include image processing and computer vision.

Jiayi Ma (Member, IEEE) received the B.S. degree in information and computing science and the Ph.D. degree in control science and engineering from the Huazhong University of Science and Technology, Wuhan, China, in 2008 and 2014, respectively. He is currently a Professor with the Electronic Information School, Wuhan University, Wuhan, China. He has authored or coauthored more than 150 refereed journal and conference papers, including IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, IEEE TRANSACTIONS ON IMAGE PROCESSING, *International Journal of Computer Vision*, CVPR, ICCV, and ECCV. His research interests include computer vision, machine learning, and pattern recognition. He was identified in the 2019 and 2020 Highly Cited Researchers list from the Web of Science Group. He is the Area Editor of the *Information Fusion*, an Associate Editor for the *Neurocomputing*, and the Guest Editor of the *Remote Sensing*.

Zhongyuan Wang (Member, IEEE) received the Ph.D. degree in communication and information system from Wuhan University, Wuhan, China, in 2008. He is currently a Professor with Computer School, Wuhan University. He has authored or coauthored more than 80 papers in major international conferences and journals. His research interests include image processing and multimedia analytics.



Jiaming Wang received the B.S. degree from the College of Post and Telecommunication of Wuhan Institute of Technology, Wuhan, China, in 2016 and the master's degree from the Wuhan Institute of Technology, Wuhan, China, in 2018. He is currently working toward the Ph.D. degree with the State Key Laboratory of Information Engineering in Surveying, Mapping and Remote Sensing, Wuhan University, Wuhan, China. His research interests include image or video processing and computer vision.



Deren Li is currently an Academician of Chinese Academy of Sciences, Chinese Academy of Engineering and International Eurasian Academy of Sciences, Guangzhou, China. He promotes the theory and practice of the development of digital cities in China to smart cities. He is the policy maker in smart city strategy of the Ministry of Science and Technology and the Ministry of Industry and Information Technology of China. He is also an expert on the smart city strategy of China Development and Reform Commission, the chief scientist of Optics Valley of China. He is the current Director of the Academic Committee of the State Key Laboratory of Information Engineering in Surveying, Mapping and Remote Sensing.