The Visual Computer (2021) 37:1769–1780
https://doi.org/10.1007/s00371-020-01937-y

ORIGINAL ARTICLE

Trajectory and image-based detection and identification of UAV

Yicheng Liu[1] · Luchuan Liao[1] · Hao Wu[2] · Jing Qin[3] · Ling He[1] · Gang Yang[1] · Han Zhang[1] · Jing Zhang[1]

Published online: 29 July 2020

Abstract
Much more attentions have been attracted to the inspection and prevention of unmanned aerial vehicle (UAV) in the wake of increasing high frequency of security accident. Many factors like the interferences and the small fuselage of UAV pose challenges to the timely detection of the UAV. In our work, we present a system that is capable of detecting, recognizing, and tracking an UAV using single camera automatically. For our method, a single pan–tilt–zoom (PTZ) camera detects flying objects and gets their trajectories; then, the trajectory identified as a UAV guides the camera and PTZ to capture the detailed region image of the target. Therefore, the images can be classified into the UAV and interference classes (such as birds) by the convolution neural network classifier trained with our image dataset. For the target recognized as a UAV with the double verification, the radio jammer emits the interferential radio to disturb its control radio and GPS. This system could be applied in some complex environment where many birds and UAV appear simultaneously.

Keywords UAV · Drone detection · Trajectory identification · Tracking · Object recognition · Deep learning

1 Introduction

In recent years, the popularization of the UAV rises in an ever-increasing speed due to the great maneuverability and the reduction of price. They are widely adopted by private individuals and can cause security issues for security-sensitive locations such as nuclear facilities, airports, and some private residences. For instance, two UAVs attacked on Venezuelan President during a speech at a military parade on August 4, 2018 [1]. To address these risks, we propose a framework based on a single PTZ camera for detecting UAV in real time and warning the watchman as early as possible.

Several approaches are applied to the problem of the UAV detection and recognition [2], such as Doppler radar [3–6], computer vision [7–15], acoustics [16,17], infrared [18], and radio frequency (RF) detection [19].

Radar detection Radar has been applied to detect and recognize small objects like UAV in addition to the detection of big flight objects. In [3], some features were extracted from micro-Doppler signatures and distinguished into UAV or clutter with machine learning methods like support vector machines (SVM) [20] and Naive Bayes [21]. In [5], a series of points formed by radar were used to the formation of trajectory, and it was divided into two classes (bird and UAV) with multilayer perceptron neural network (MLP). The distinctive advantage of radar lies in its long-distance detection and is basically immune to the bad weather like haze and rainy day. However, it is also easy to be disturbed by the clutter, such as the sea clutter and the ground clutter.

Acoustic detection Features based on acoustic have also been adopted to the detection and identification of the UAV. In [16], the authors proposed a sound-based UAV detection that extracts features of UAV sounds from a variety of ambient sounds for SVM classification. However, due to the noisy environment near the detector, the method could be limited by the detection distance. Another approach: the UAV is detected by the acoustic sensor network, then the equipment expelled the UAV with the defined radio [17]. The low cost of acoustic equipment is the superiority of these methods based on acoustic. But they are easy to be obstructed by some noise.

Video approaches The camera can detect UAVs effectively. the UAV. Many methods of feature extraction are the basis of imagery detection and classification, and they can be segregated into two major categories: the handcrafted feature and the high-level semantic feature of the image.

✉ Jing Zhang
jing_zhang@scu.edu.cn

1 College of Electrical Engineering, Sichuan University, Chengdu 610065, China

2 Sky Defence Technology, Chengdu 610213, China

3 Centre for Smart Health, School of Nursing, The Hong Kong Polytechnic University, Hong Kong, Hung Hom, China

Before the extensive application of deep learning in the detection of small flying objects, the hand-designed features of the image (such as HOG [22], Fourier descriptors [23] and SURF [24]) are generally applied in the detection and classification of UAV. Eren et al. [14] extracted the generic Fourier descriptor from the imagery of the objects, and used a neural network to classify the object to the bird and the UAV. For this method, the feature extraction and the training of the classifier take less time than the methods based on the CNN, but its recognition rate is lower than the classifiers based on CNN.

Image feature, which was extracted from CNN, is generally more detailed compared to the traditional way. In [11], the method based on the hybrid deep network and sparse autoencoder model was applied to get the high layer visualizing feature of the image. The experiment indicated that the performance of the higher layer visualizing feature is better than the lower layer visualizing feature in image classification, but the process of feature extraction is complex. In [15], many state-of-the-arts convolutional object detectors (such as Fast r-CNN [25], Faster r-CNN [26] and YOLOv2 [27]) were tested for the detection and recognition of the UAV. However, these detectors are not ideal in the performance of accuracy and real time.

For the majority of detection methods applied in the video, they normally require a higher resolution image to extract more detailed features in order to ensure a higher recognition rate. However, the recognition rate of these methods for small and low-resolution images is not guaranteed. For instance, in Fig. 1, three flying objects are too small to be directly identified with these methods based on CNN. Therefore, [28] used the U-Net network to detect suspected UAV targets, then applied residual network to identify target candidates, and finally utilized the spatiotemporal characteristics of the UAV to filter out interference. In [29], super-resolution technology is utilized to enlarge the small target image. The Faster-RCNN model is then used to detect the UAV target. The above two methods achieved the first and second results of the 2019 Drone-vs-Bird Detection Challenge [30], respectively. It can be seen from these methods that combining various methods according to application scenarios can effectively improve target detection performance. For example, [7] proposed a method that extracts the long trajectories of flying objects to confirm the rough location of a UAV. The image including the regions of interest was captured by a zoom camera to locate and recognize the target using a popular SSD detector [31]. However, this method cannot distinguish the trajectories of flying birds and UAVs well, and it is easy to cause false alarms in the environment where there are a large number of flying birds.

In this article, in view of the problem of insufficient pixels and flying birds interference in the process of long-distance detection of UAVs, we proposed a long-range UAV detection method based on target trajectory and shape recognition. The system can detect, identify and track small UAVs in real time. This method extracts the spatiotemporal features of trajectory and morphology according to the flying characteristics of UAVs and birds and then uses the neural network to classify features. Therefore, this method does not depend on the image features and can accurately identify remote UAVs in real time. In addition, in order to reduce false positives of trajectory recognition and increase system stability, a single PTZ camera is applied to zoom in on suspicious targets and then further confirm the target attributes with image recognition.

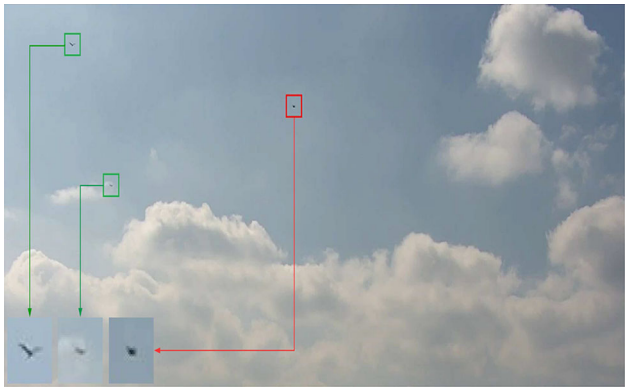


Fig. 1 Three small flying objects appear simultaneously in the sky. They are too small to recognize for human. The object in the red box is a UAV. These objects in the green box are flying birds

2 Detection and identification methods

Initially, the single PTZ camera remains at rest. The system flow is then divided into two processing flows according to the size of the target. For big objects, the targets are directly detected and recognized without zooming in. On the contrary, there are four steps to the detection of small objects. Firstly, the small flying objects are detected for the formation of trajectory. Secondly, the Non-interfering trajectories that have been screened out from the set of trajectory are classified into the bird and the UAV. Thirdly, the single PTZ camera is used to capture the clear image of the target which has been recognized as UAV by its trajectory. Finally, these sharper images are classified into birds, UAVs, and clutters.

2.1 System architecture

Figure 2 shows the architecture of our system which contains four modules: flying objects detection, trajectory identification, PTZ camera control, and image recognition.

Flying objects detection: the average background modeling is applied to get the points of trajectory or the sharp images of objects.

Trajectory identification: the trajectory of flying objects formed by a series of points are screened from the set of the trajectory which may contain many trajectories of interference like shaking leaves and moving clouds. After that, the features of the non-interfering trajectories are extracted and classified into the bird and the UAV when there are not enough details to recognize the image.

PTZ control: the motion of the PTZ system and the adjustment of the camera aperture are guided by the trajectory of UAV to capture the sharp image of it.

Image recognition: the detailed images may come from two ways. One type of images is captured with the detection algorithm when the object has enough pixels in the screen. In another section, after the identification of trajectory, the detail images are captured by the camera with the help of the PTZ camera system. Eventually, all the detailed images are classified into three classes: the bird, the UAV, and the interference.

2.2 Flying objects detection

For our system, the camera always stares at the surveillance space until it discovers an intrusive UAV. To detect flying objects quickly, the average background modeling is adopted to extract flying objects from still-camera video.

The first N grayscale frames of the video are used to build the average background model, as follows in formula (1).

$$u(x, y) = \frac{1}{N} \sum_{i=0}^{N-1} I_i(x, y). \tag{1}$$

In equation (1), N is usually 100, and $I_i(x, y)$ is the grayscale value of one pixel in the ith frame. $u(x, y)$ is the grayscale value at (x, y) in the background image. In order to adapt the change of the background, the adaptive algorithm is utilized to update the threshold (TH) used in the foreground detection. Firstly, between different frames, the average (u_{ave}) and the standard deviation (ave_{std}) of every pixel are computed by formulas (2), (3), and (4), respectively.

$$F_i(x, y) = |I_i(x, y) - I_{i-t}(x, y)| \tag{2}$$

$$u_{ave}(x, y) = \frac{1}{M} \sum_{i=s+1}^{M} F_i(x, y) \tag{3}$$

$$ave_{std} = \sqrt{\frac{1}{M} \sum\nolimits_{i=s+1}^{M} \sum (F_i(x, y) - u_{ave}(x, y))}. \tag{4}$$

In equation (2), t, set as 3, represents the interval between two frames. To ensure the accuracy of u_{ave} and ave_{std}, M represents the number of consecutive frames, which was set to 30 in equations (3) and (4). The threshold of every pixel is calculated using formula (5):

$$TH = u_{ave} + \beta * ave_{std}. \tag{5}$$

In order to increase the robustness of detection, the background model is updated after a single detection. For each pixel, $u(x, y)$ is updated to $u'(x, y)$ using formula (6):

$$u'(x, y) = (1 - \alpha)u(x, y) + \alpha I(x, y) \tag{6}$$

where $\alpha (0 \leq \alpha \leq 1)$ is the background update factor.

The size of the object reflected on the screen is determined by the number of the pixel (δ) in its connected domain. In order to simplify the calculation process, the operation is segmented into two pipelines after the detection of flying objects. Firstly, the detailed image is directly classified into the interference and the UAV if its δ is more than 20. Secondly, the trajectory is formed for the preliminary identification of the object when its δ is less than 20.

2.3 Trajectory association

After the detection of flying objects, the coordinate and δ of the effective detection point are recorded. Each coordinate is then associated with the corresponding trajectory. Meanwhile, the trajectories caused by the clutter like moving leaves and clouds were removed from the set of trajectory.

The points in each frame are numbered as k, and $(k = 1, 2, 3, \ldots, K)$. (x_k, y_k) is the coordinate of the kth point. For the first frame, every point is regarded as an initial point of a trajectory. The search range R ($[x_k \pm w, y_k \pm h]$, where w and h are lengths in horizontal and vertical, is set to match the optimum correlation point of a trajectory in current frame quickly. However, there may be many points in the range R. Consequently, best-associated point is defined as the point which has the shortest distance with the latest point in a trajectory.

Except for the trajectory of the flying objects, there may be some interferential trajectories caused by moving leaves and clouds, and they usually swing back and forth in the horizontal direction. Besides, the area of bounding boxes around the trajectory caused by noise is generally small than others. Therefore, these two characteristics of the interference trajectory are used to distinguish between the interference trajectory and the non-interference trajectory.

The flow of the trajectory association is shown in Fig. 3. Firstly, the existing trajectory matches the associated point in its search range. If a point to be linked cannot be associated with all existing trajectories, it is regarded as an initial point

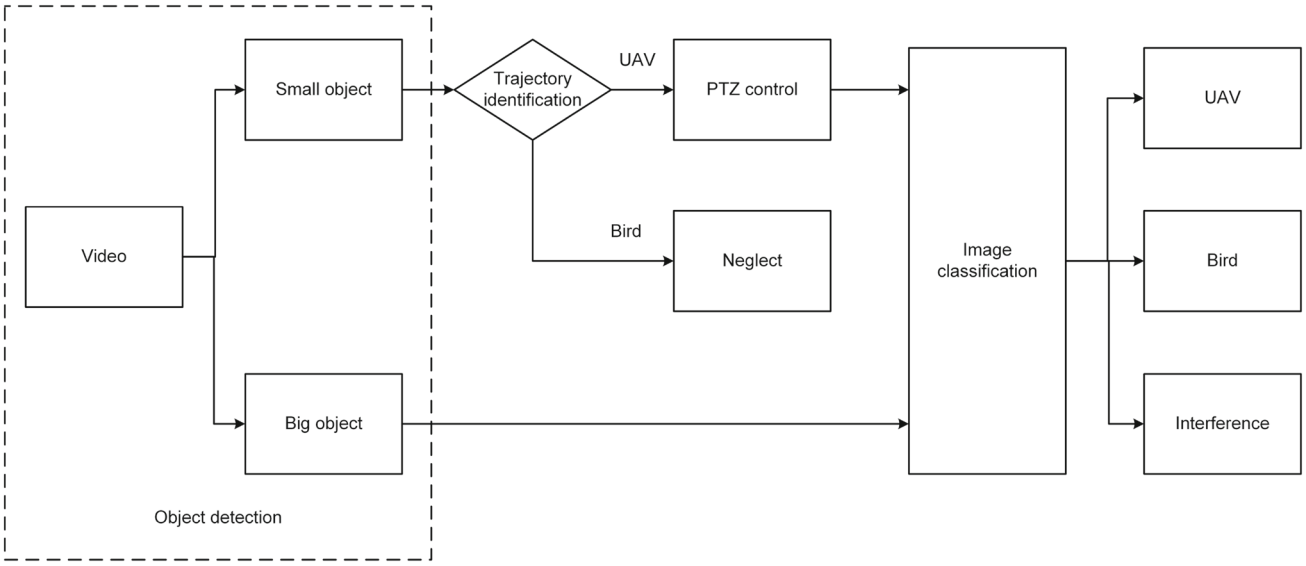


Fig. 2 System architecture

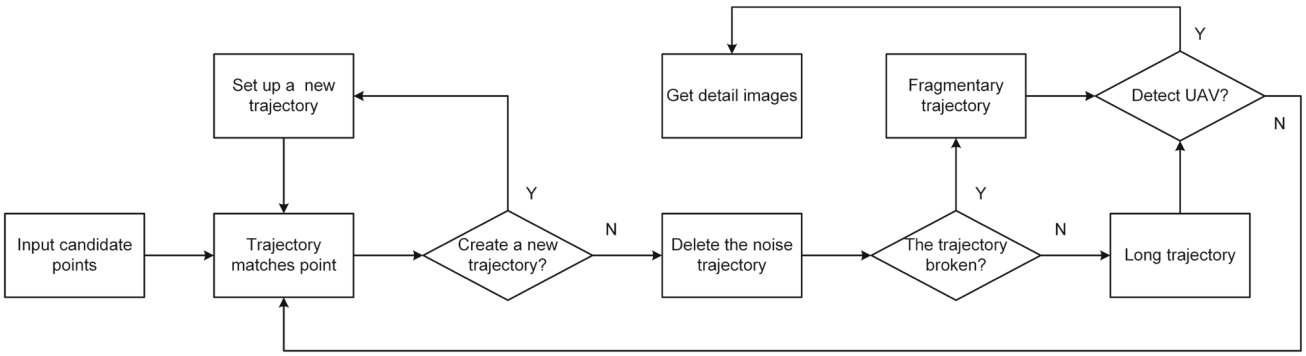


Fig. 3 The flow of trajectory association

of a new trajectory in the next matching mission. Secondly, the trajectory of noise is eliminated from the set of trajectories. Thirdly, when the trajectory has no associated point in past five successive frames, and the number of points in it exceeds λ, it is deemed as a fragmentary trajectory of the potential target. Finally, the long trajectories are first used for trajectory identification. If no UAV is detected, the fragmentary trajectories are classified to avoid missing any potential UAV.

2.4 Trajectory features extraction

In order to identify the trajectory in real time, the trajectory is divided into small segments which contain n points. Since each point of trajectory could be the starting point of a segment, a segment marquee is set to slide along a trajectory with the step size of 1, as shown in Fig. 4b. Long trajectories are classed into the bird and the UAV, as shown in Fig. 4a. Because the flight posture of the bird constantly changes while its wings are spreading and closing. The points in the trajectories of birds may fluctuate vertically and disappear in some frames. Comparing with the bird, however, the flight posture of UAV is more smooth and uniform, as shown in the picture (b) of Fig. 4. Meanwhile, there is a significant change in the number of effective pixels of the bird filmed on the screen. Therefore, features are extracted from two pipelines: the trajectory and shape of the object.

Trajectory features The displacement of two adjacent points in a trajectory is decomposed in horizontal and vertical directions. The features of horizontal and vertical are then extracted, respectively.

The average displacement in horizontal is D_{ave}:

$$D_{ave} = \frac{1}{n-1}\sum_{i=1}^{n-1} d_{xi} \qquad d_{xi} = x_i - x_{i-1} \tag{7}$$

where d_{xi} is distance between the ith point and $(i+1)$th point in horizontal, (x_i, y_i) is the coordinate of ith point in

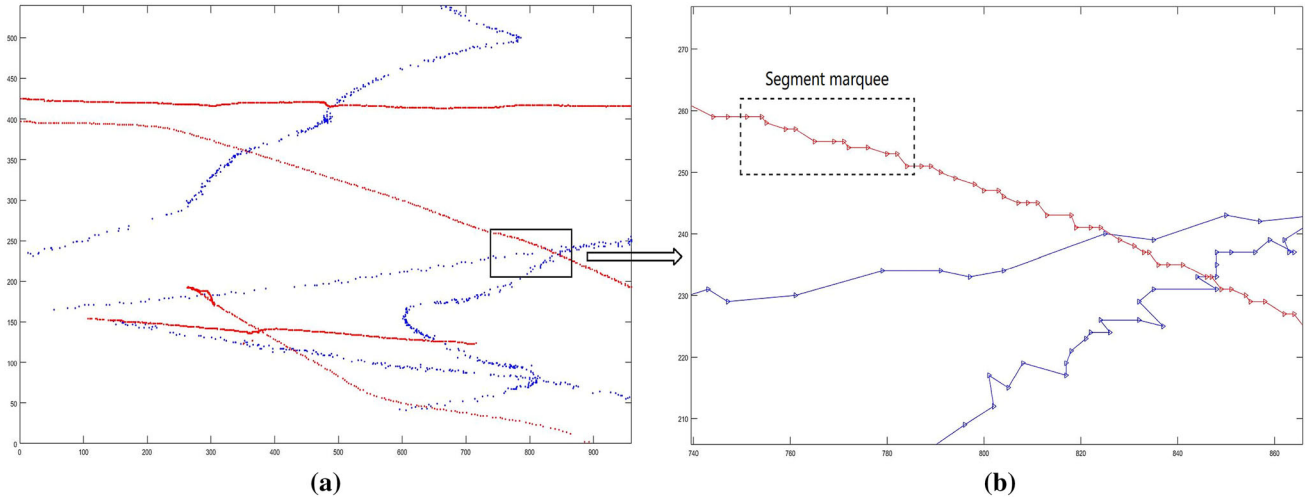


Fig. 4 Red and blue trajectories are formed by a UAV and birds, respectively. The picture on the left side is **a** a global graph of trajectories on the screen. The picture on the right side **b** is a larger version of the box in the left picture, and the dotted box is a segment marquee which contains 13 points

a segment, and n is the number of points in a segment. D_r is the ratio of the minimum horizontal displacement to the maximum horizontal displacement.

$$D_r = \frac{\min d_{xi}}{\max d_{xi}}, \quad (i = 1, 2, \ldots, n-1). \tag{8}$$

MSE and σ: The variance and standard deviation of d_{xi} in horizontal.

$$MSE(D) = \frac{1}{n-1} \sum_{i=0}^{n-1} (d_{xi} - D_{ave})^2 \tag{9}$$

$$\sigma = \sqrt{MSE(D)}. \tag{10}$$

θ is defined as the average deviation.

$$\theta = \frac{1}{n-1} \sum_{i=0}^{n-1} |d_{xi} - D_{ave}| . \tag{11}$$

After that, five features of the vertical displacement are extracted in the same way.

To highlight the angular variation between the bird's trajectory and the trajectory of the UAV, the second-order differential features between the three adjacent points were extracted. The second-order differential of three successive detections of each segment is β_i:

$$\beta_i = \frac{y_{i+2} + y_i - 2 * y_{i+1}}{x_{i+2} - x_i}, \quad (i = 0, 1, \ldots, n-2). \tag{12}$$

Same as the above method, five features of the second-order differential are extracted. Therefore, 15 features have been extracted from the trajectory.

Shape features As mentioned earlier, the shape of the bird in successive frames may change too fast that it could cause the fluctuation of the pixel number which reflected the morphological changes of the bird. Consequently, the number of the pixels of the object for each point in a segment (p_i) is also recorded along with coordinate.

P_r is the ratio between the minimum and maximum of p_i.

$$P_r = \frac{\min p_i}{\max p_i}, \quad (i = 0, 1, \ldots, n-1) \tag{13}$$

P_{ave}, $MSE(P)$ and σ' are the mean, the variance and the standard deviation of the number of pixels in each segment, respectively.

$$P_{ave} = \frac{1}{n-1} \sum_{i=0}^{n-1} p_i \tag{14}$$

$$MSE(P) = \frac{1}{n-1} \sum_{i=0}^{n-1} (p_i - P_{ave})^2 \tag{15}$$

$$\sigma' = \sqrt{MSE(P)}. \tag{16}$$

2.5 PTZ and zoom control

The latest position of the trajectory identified as the UAV is used to guide the camera and PTZ to capture the detailed images of the target. However, it is possible that the target may move out from the view of the camera with the adjustment of the focal length and the movement of the target. To handle this problem, the PTZ is controlled to rotate with the move of the target automatically. Therefore, the field of camera view is separated into three rectangular sections (R_1, R_2, R_3), where R_1 is the central zone, R_3 is the area of the edge,

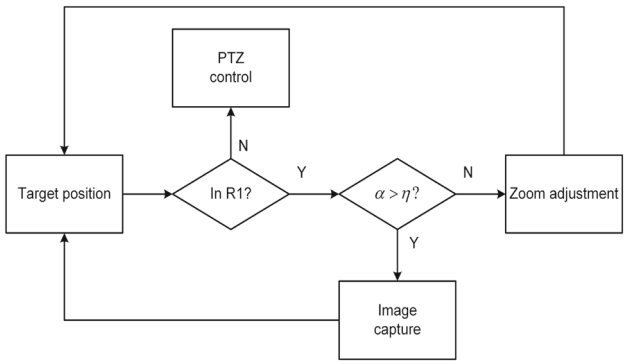


Fig. 5 The flow of PTZ and camera control

and R_2 is between R_1 and R_3. At the same time, the three regions correspond to three different rotation speeds of PTZ (S_1, S_2, S_3). As the PTZ rotates, the focal length of the camera is automatically adjusted according to the size of the δ, so it is also divided into several levels.

The control flow is shown in Fig. 5. If the position (P) of the target on the screen is in the region R_1, the flow goes directly to the next step; otherwise, when P is in the $R_2(R_3)$ region, the PTZ rotates at speed $S_2(S_3)$ to bring the target in region R_1, and the flow proceeds to the next step.

The next step is to determine the relationship between δ and the threshold η (where it is 20). If δ is higher than η, the detail image of the target is acquired for image recognition; otherwise, the focal length of the camera is adjusted to the next level to zoom in on the target. As described above, by adjusting the focal length of the camera and the rotation of the PTZ, the system can obtain a clear target image and continuously track the target without losing.

2.6 Image recognition

In this section, there are two types of detailed images for the target recognition. If the targets are close enough to the camera, they can be filmed by the camera without amplification. Afterward, the filmed object is directly classified as the UAV or the interference without analyzing trajectory. Otherwise, the object is far from the camera. If it is identified as UAV in the trajectory pre-identification, the PTZ is applied to obtain the magnified image.

In both situations, the captured object images have the same size. To confirm the target type, the CNN is adopted to classify the images. Compared with other network structures, the residual network (ResNet) [32] can solve the problem of gradient disappearance caused by the deepening of the network, and its shortcut can further extract more information under the premise of ensuring network performance. Therefore, a ResNet is constructed to train the image classifier, which includes convolutional layer, residual block, pooling, and fully connected layer, as illustrated in Fig. 6.

Convolutional layer Since the size of the input image does not exceed $64 * 64$, There are only 9 convolutional layers in our network. The structure of convolution layers is $(N, M, W * H)$, where M and N are the numbers of channels of the input and output for the convolutional layers, and W and H are the width and length of each convolutional filter, respectively.

In order not to discard the information of the original image, each edge of the input image is padded with a pixel having a value of 0. Then, it is convolved by $c * 3 * 3$-sized convolutional filters with stride of 1 to extract more local information. Thus, the size of the output feature maps of the first convolutional layer (Conv1) is $c*(w*h)$, where w and h are the width and length of input image, respectively. Except for the 8th and 9th convolutional layers, the output of each other convolutional layer is normalized by the batch normalization (BN) [33] to reduce the internal covariate shift and is then activated by the rectified linear unit (ReLU) [34].

Residual block There are two residual blocks behind the Conv1. For the first residual block, the Conv4 is the convolution shortcut, and the Conv2 and Conv3 are called the plain convolution layers. In order to extract high-dimensional features while reducing the output dimension, the size and stride of Conv2 are set to $(2c, c, 3 * 3)$ and 2, respectively. The size and stride of Conv3 are set to $(2c, 2c, 3 * 3)$ and 1, respectively, to extract more details in the receptive field without increasing the number of channels. After the convolution operation of the Conv2 and Conv3, the size of the input feature map is downsampled to $2c * \frac{1}{2}w * \frac{1}{2}h$. However, the input map size of shortcut is $c * w * h$. Therefore, the output feature map size of shortcut is also resized to $2c * \frac{1}{2}w * \frac{1}{2}h$ by Conv4 with stride of 2 to make it can be superimposed with the output of the plain convolutional layer. The residual block 2 is similar to the residual block 1. Consequently, for the Conv8, the size of the input feature map is $4c * \frac{1}{4}w * \frac{1}{4}h$.

Pooling layer and dropout layer To reduce the number of computational parameters while maintaining the main features of the feature map, two max-pooling layers are used in front of the fully connected layer, and the size of pooling filter is $p_1 * p_2$. At the same time, a Dropout layer [35], which follows the pooling layer, is utilized to alleviate overfitting.

Fully connected layer After pooling and dropout operations, the fully connected layer maps the high-level features obtained by convolution and pooling to the sample tag space. There are two fully connected layers behind the final dropout layer. For the first fully connected layer ($FC1$), the number of neurons depends on the length of the fixed-length vector obtained from the previous convolution and pooling operations. For the second fully connected layer ($FC2$), its number of neurons is half of $FC1$.

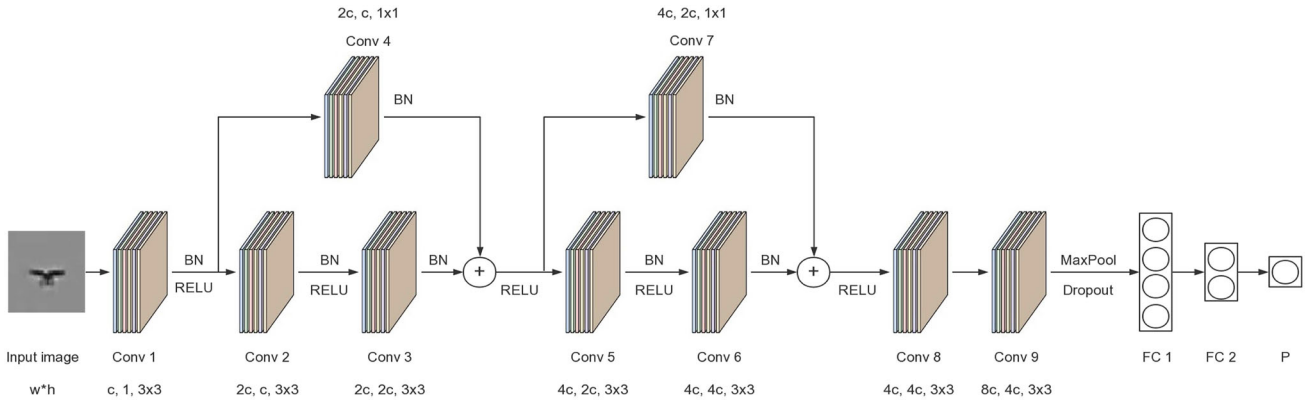


Fig. 6 The structure of ResNet

3 Experiment and result

3.1 Datasets

Trajectory dataset In our experiment, 19 videos with a framerate of 25 fps were filmed by a stationary camera. Ten videos recorded the irregular flight of UAV (DJI Phantom3 SE) and were used to extract UAV trajectory segment, three of which (including 2150 1280 $*$ 720 frames) were taken on a cloudy day and seven (containing 6900 1280*720 frames) were obtained on a sunny day. Nine videos were utilized to obtain bird trajectory segment, seven of which (including 2325 1280 $*$ 720 frames) were filmed on a cloudy day and two (containing 7075 1280 $*$ 720 frames) were filmed on a sunny day. For all birds and UAV in these videos, their trajectories were recorded when their effective pixels are less than 20. Finally, 3846 trajectory segments of UAV and 4198 trajectory segments of birds are acquired and used for the extraction of trajectory feature.

In a trajectory segment, 15 features were extracted from the trajectory, and four shape features were taken from the morphological of the object. The features have different dimensions. In order to speed up data analysis and improve the accuracy, they were normalized to a mean of 0 and a variance of 1. Eventually, 70% of the segments were randomly selected as the training set, and 30% as the test set.

Image datasets In this article, we have built an image library which containing UAVs, flying birds, and other background objects, as shown in Fig. 7. The image dataset was built from 25 video sequences whose frame rates are 25 fps. There are 20,000 images of UAV, birds, and interference images, respectively. Every 10 consecutive frames were grouped as one sample. From the total 6000 samples, we have randomly chosen 70% as the training set and 30% as the test set. Since the image captured by the PTZ camera in long distance or the haze contains little color information, all the pictures in the experiment were transformed to grayscale images.

3.2 Trajectory identification

After the trajectory features were extracted and normalized, the neural network was used to train the trajectory classifier. Then, the identification performance based on different number points in the trajectory segment was evaluated.

Trajectory identification using neural network The neural network is very flexible and can extract features from the data directly. In this section, it was used to train the trajectory classifier. The structure of the neural network mainly included three parts: one input layer, three hidden layers, and one output layer, where each hidden layer contains 64 neurons.

The Adam optimization algorithm [36] was applied to train the neural network, and the ReLU was set as the activate function. The decay rates of the first and second moments, d_1 and d_2, were deployed as 0.9 and 0.999, respectively. The initial learning remained constant as 0.0001. The trajectory classification model is obtained after 350 epochs.

Identification performance analysis The performance of the recognition in this paper is evaluated by $precision$, $recall$, and $F1\_score$.

$$Precision = \frac{TP}{TP + FP} \times 100\% \tag{17}$$

$$Recall = \frac{TP}{TP + FN} \times 100\% \tag{18}$$

$$F1\_score = \frac{2 \times TP}{2 \times TP + FP + TN} \times 100\% \tag{19}$$

where TP is the number of positive samples in the correct predictions, FP is the number of positive samples in the wrong predictions, TN is the number of negative samples in the error predictions, and FN is the number of positive samples in the error predictions.

In this paper, these experiments are completed on a computer equipped with Windows 10 system, which is configured as Inter (R)Core(TM)i7-8550U@1.8GHz CPU, 8G memory.

Fig. 7 The left side of the figure is bird imagery **a** detected by the camera; the middle of the figure shows the target imagery of UAV **b** in our picture database; on the right, these images are interference **c** captured by the camera

Table 1 $Precision$, $Recall$, and $F1\_score$ for the various number of points in the segment

Number of points in the trajectory segment (n)	12 (%)	13 (%)	14 (%)	15 (%)	16 (%)	17 (%)	18 (%)
Precision (bird)	98.96	98.79	99.58	98.91	99.59	**99.89**	99.27
Recall (bird)	98.75	99.59	99.27	99.50	**99.80**	99.50	99.79
F1_score (bird)	98.85	99.69	99.43	99.21	99.69	**99.70**	99.53
Precision(UAV)	98.81	99.55	99.30	99.47	**99.79**	99.47	99.79
Recall (UAV)	99.01	99.76	99.60	98.83	99.59	**99.89**	99.27
F1_score (UAV)	98.91	98.66	99.45	99.15	**99.69**	99.68	99.53

The bolded data are the maximum value of the line in which it is located

The graphics card is NVIDIA GeForce MX150 with a memory of 2048M. The experimental development platform is Microsoft Visual Studio 2015. And the development language is C++.

As shown in Table 1, it shows the differential performance of the different numbers of points in each segment. The three evaluation metrics are very high for trajectory identification. In some cases, the three metrics of bird and UAV may fluctuate slightly, but they all exceed 99.5% when n reaches 16. In our system, n was set as 16, which is able to meet the requirements for the initial identification of the potential UAV.

When the small UAV appears on the screen, the system could detect and identify it within 1.5 seconds. Figure 8 is the detection results of the 57th, 364th, and 379th frames in a video. Figure (a) is the foregrounds detected by the adaptive background update algorithm. It can be seen that there is a lot of noise in addition to the flying objects. Figure (b) is the resulting graph after trajectory clustering and recognition. It can be seen that the original noise point has been completely cleared. The UAV and flying birds in the same figure could be accurately identified in real time. Experiments show that the trajectory identification algorithm in this paper could process videos with a resolution of $1280 * 720$ at a speed of 36fps.

3.3 Image recognition

In image processing, we changed the network structure and the input image size, and then evaluated the performance of the different combinations.

Model training Pytorch [37] was utilized to build and train the CNN presented in Sect. 2.5. In order to analyze the effect of different image sizes on the recognition rate, the size of the input image was changed from $32 * 32$ to $48 * 48$ and $64*64$. In addition, for the different sizes of the input images, we adjusted the CNN structure to explore the impact of the different parameters on the network performance. These different parameters contain the size of Conv1, the pooling filter size of two pooling layers, and the number of neurons in two fully connected layers, as shown in Table 2.

In our experiment, the Adam optimization algorithm was also utilized to train the CNN. The learning rate, named as γ, was initialized as 0.0005, and then, it was reduced by half every 50 epochs during the training process to ensure the training efficiency of the network and the stability in the later stage. During the network training process, the batch size was set to 128. Finally, the classifier was obtained after 100 epochs.

Evaluation In this section, the performance of image recognition was also evaluated by the $Precision$, $Recall$,

(a) The foreground obtained by the adaptive background update

(b) Trajectory recognition result. Green and red trajectories are birds and UAV, respectively.

Fig. 8 The detection and recognization of the small UAV

Table 2 $Precision$, $Recall$, and $F1\_score$ under various input image sizes and CNN architectures

Input image size	Conv1 size	Pooling filter size		Fully connected layers		Macro-averaging of metrics		
		Pooling1	Pooling2	FC1	FC2	$Precision$ (%)	$Recall$ (%)	$F1\_score$ (%)
32*32	32*3*3	2*2	2*2	256	128	99.27	99.28	99.27
48*48	32*3*3	2*2	3*3	256	128	99.34	99.38	99.36
64*64	32*3*3	2*2	2*2	1024	512	**99.46**	99.48	**99.48**
32*32	64*3*3	2*2	2*2	512	256	99.28	99.33	99.31
48*48	64*3*3	2*2	3*3	512	256	99.38	99.45	99.41
64*64	64*3*3	2*2	2*2	2048	1024	99.44	**99.52**	99.47

The bolded data are the maximum value of the column in which it is located

and $F1\_score$. To simplify the evaluation process, the macro-averaging, which calculates metrics for each label and finds their unweighted mean, was applied to compute each evaluation metrics.

According to Table 2, the three metrics reach 99.20% for different CNN network, and the variations between them are very slight. When the input image size and Conv1 size are $64*64$ and $32*3*3$, the $Precision$ and $F1\_score$ reached the highest of 99.46% and 99.48%, respectively.

Figure 9 shows the accuracy and loss curves of training and testing in the third experiment. It can be seen that the model gradually converges during the training process. The loss and accuracy tend to stabilize after the 50th epochs. In addition, by analyzing these curves, we can know that the model does not have overfitting and underfitting, which indicates that the network structure and parameters are set properly. The final

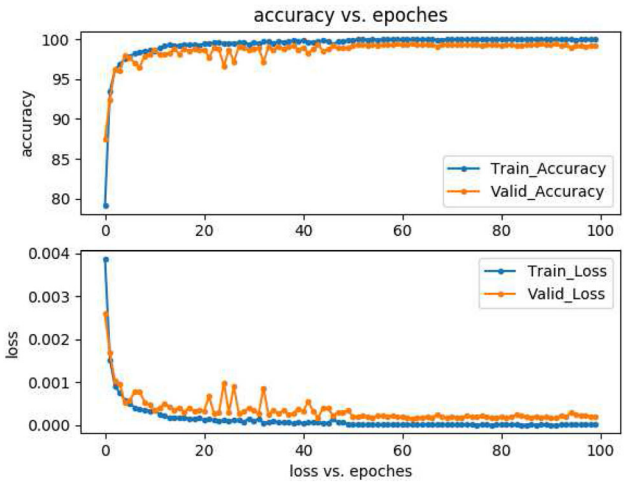


Fig. 9 The loss and accuracy during training

experiment shows that the image recognition model could recognize an image in 15.9ms.

4 Conclusion

In this paper, a system is proposed to detect the UAV with trajectory and image feature identification. Firstly, the trajectory features of flying objects are extracted and then classified into the bird and UAV with the neural network. Afterward, the trajectory, which has been identified belonging to UAV, guides the PTZ camera to capture the detailed images of the potential UAV. These detailed images are recognized by the CNN classifier. The experiment verified that the identification accuracy of trajectory and image reaches 99.50% and 99.89%, respectively. The system could be used in complex environments where many birds and UAV appear simultaneously in the sky. In addition, the system is able to early identify the potential UAV which is farther away from the camera. In the future study, we will append more kinds of UAV videos for improving the robustness of the image classifier. Since the method in this paper deals with two-dimensional images, the trajectory detection method is not effective when the UAV is flying along the vertical line of the lens plane from a distance.

Acknowledgements This research was partially supported by research grants from the National Natural Science Foundation of China (Grant No.: 61571314) and the Sky Defence Technology Co., Ltd.

References

1. Alvine, C.: Drone carrying explosives almost assassinates Venezuela's president. https://www.techzim.co.zw/2018/08/drone-carrying-explosives-almost-assassinates-venezuelas-president/ (2018). Accessed 06 Aug 2018
2. Samaras, S., Diamantidou, E., Ataloglou, D., Sakellariou, N., Vafeiadis, A., Magoulianitis, V., Lalas, A., Dimou, A., Zarpalas, D., Votis, K.: Deep learning on multi sensor data for counter uav applications: a systematic review. Sensors **19**(22), 4837 (2019)
3. Molchanov, P., Harmanny, R.I.A., de Wit, J.J.M., Egiazarian, K., Astola, J.: Classification of small UAVs and birds by micro-Doppler signatures. Int. J. Microw. Wirel. Technol. **6**(3–4), 435–444 (2014)
4. Chenchen, J.H., Ling, H.: An investigation on the radar signatures of small consumer drones. IEEE Antennas Wirel. Propag. **16**, 649–652 (2017)
5. Mohajerin, N., Histon, J., Dizaji, R., Waslander, S.L.: Feature extraction and radar track classification for detecting UAVs in civillian airspace. In: 2014 IEEE Radar Conference. 0674-0679 (2014)
6. Shin, D.H., Jung, D.H., Kim, D.C., Ham, J.W., Park, S.O.: A distributed FMCW radar system based on fiber-optic links for small drone detection. IEEE Trans. Instrum. Meas. **66**(2), 340–347 (2017)
7. Schumann, A., Sommer, L., Müller, T., Voth, S.: An image processing pipeline for long range UAV detection. Int. Soc. Opt. Photon. **10799**, 107990T (2018)
8. Unlu, E., Zenou, E., Rivière, N.: Using shape descriptors for UAV detection. Electron. Imaging **2018**(9), 1–5 (2018)
9. Rozantsev, A., Lepetit, V., Fua, P.: Detecting flying objects using a single moving camera. IEEE Trans. Pattern Anal. Mach. Intell. **39**(5), 879–892 (2017)
10. Kapoor, A., Biswas, K., Hanmandlu, M.: An evolutionary learning based fuzzy theoretic approach for salient object detection. Vis. Comput. **33**(5), 665–685 (2017)
11. Yu, C., Hongbing, M., Xinling, W., Pengge, M., Yuxin, Q., Zhengxiang, M., Zhaoyu, L.: Classification methods of a small sample target object in the sky based on the higher layer visualizing feature and transfer learning deep networks. EURASIP J. Wirel. Commun. Netw. **2018**(1), 127 (2018)
12. Aker, C., Kalkan, S.: Using deep networks for drone detection. In: 2017 14th IEEE international conference on advanced video and signal based surveillance (AVSS). pp. 1–6 (2017)
13. Yoshihashi, R., Trinh, T.T., Kawakami, R., You, S., Iida, M., Naemura, T.: Differentiating objects by motion: joint detection and tracking of small flying objects (2017). arXiv preprint arXiv:1709.04666
14. Eren, U., Emmanuel, Z., Nicolas, R.: Generic fourier descriptors for autonomous UAV detection. In: 7th International Conference on Pattern Recognition Applications and Methods (ICPRAM). vol 1, pp. 550–554 (2018). https://doi.org/10.5220/0006680105500554
15. Park, J., Kim, D.H., Shin, Y.S., Lee, S.: A comparison of convolutional object detectors for real-time drone tracking using a PTZ camera. In: 2017 17th International Conference on Control, Automation and Systems (ICCAS), pp. 696–699 (2017)
16. Anwar, M.Z., Kaleem, Z., Jamalipour, A.: Machine learning inspired sound-based amateur drone detection for public safety applications. IEEE Trans. Veh. Technol. **68**(3), 2526–2534 (2019)
17. Xuejun, Y., Yongxin, L., Jian, W., Houbing, S., Huiru, C.: Software defined radio and wireless acoustic networking for amateur drone surveillance. IEEE Commun. Mag. **56**(4), 90–97 (2018)
18. Andraši, P., Radišić, T., Muštra, M., Ivošević, J.: Night-time detection of UAVs using thermal infrare camera. Transp. Res. Procedia **28**, 183–190 (2017)
19. Nguyen, P., Ravindranatha, M., Nguyen, A., Han, R., Vu, T.: Investigating cost-effective RF-based detection of drones. In: Proceedings of the 2nd Workshop on Micro Aerial Vehicle Networks, Systems, and Applications for Civilian Use, pp. 17–22 (2016)
20. Cortes, C., Vapnik, V.: Support-vector networks. Mach. Learn. **20**(3), 273–297 (1995)
21. McCallum, A., Nigam, K.: A comparison of event models for naive Bayes text classification. In: AAAI-98 Workshop on Learning for Text Categorization. Citeseer, vol.752, no. 1, pp. 41–48 (1998)
22. Dalal, H., Triggs, B.: Histograms of oriented gradients for human detection. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), vol. 1, pp. 886–893 (2005)
23. Zahn, C.T., Roskies, R.Z.: Fourier descriptors for plane closed curves. IEEE Trans. Comput. **21**(3), 269–281 (1972)
24. Bay, H., Tuytelaars, T., Van Gool, L.: Surf: speeded up robust features. In: European Conference on Computer Vision (ECCV), pp. 404–417 (2006)
25. Girshick, R.: Fast R-CNN. In: Proceedings of the IEEE International Conference on Computer Vision (ICCV), pp. 1440–1448 (2015)
26. Shaoqing, R., Kaiming, He, Girshick, R., Jian, S.: Faster R-CNN: towards real-time object detection with region proposal networks. In: Neural Information Processing Systems (NIPS), vol. 39, no. 6, pp. 1137–1149 (2017)
27. Redmon, J., Farhadi, A.: YOLO9000: better, faster, stronger. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 6517–6525 (2017)
28. Craye, C., Ardjoune, S.: Spatio-temporal semantic segmentation for drone detection. In: 2019 16th IEEE International conference on advanced video and signal based surveillance (AVSS), pp. 1–6 (2019)

29. Magoulianitis, V., Ataloglou, D., Dimou, A., Zarpalas, D., Daras, P.: Does deep super-resolution enhance UAV detection?. In: 2019 16th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS), pp. 1–6 (2019)
30. Coluccia, A., Fascista, A., Schumann, A., Sommer, L., Ghenescu, M., Piatrik, T., Sharma, N.: Drone-vs-Bird detection challenge at IEEE AVSS2019. In: 2019 16th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS). pp. 1–7 (2019)
31. Wei, L., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.Y., Berg, A.C.: SSD: Single shot multibox detector. In: European Conference on Computer Vision (ECCV), pp. 21–37 (2016)
32. Kaiming, H., Xiangyu, Z., Ren, S., Jian, S.: Deep residual learning for image recognition. In: IEEE Conference on Computer Cision and Pattern Recognition (CVPR), pp. 770–778 (2016)
33. Ioffe, S., Szegedy, C.: Batch Normalization: accelerating deep network training by reducing internal covariate shift. In: Proceedings of the 32nd International Conference on Machine Learning (ICML), vol 37, no 9, pp 448–456 (2015)
34. Nair, V., Hinton, G.E.: Rectified linear units improve restricted Boltzmann machines. In: Proceedings of the 27th International Conference on Machine Learning (ICML), vol 27, pp 807–814 (2010)
35. Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., Salakhutdinov, R.: Dropout: a simple way to prevent neural networks from over. J. Mach. Learn. Res. **15**(1), 1929–1958 (2014)
36. Kingma, D.P., Ba, J.: Adam: a method for stochastic optimization (2014). arXiv preprint arXiv:1412.6980
37. Ketkar, N.: Introduction to PyTorch. In: Deep Learning with Python, pp. 195–208. Springer, Berlin (2017)

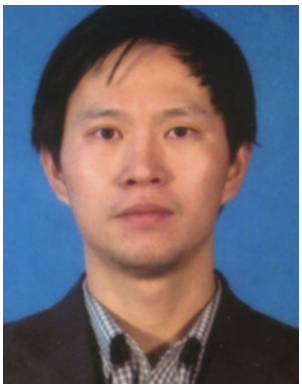
Yicheng Liu was born in China in 1975. He received his Ph.D. in control science and engineering from Tsinghua University in China. He is an associate professor in Sichuan University of Control Science and Engineering. His current interests include automation, wireless power transmission, control theory and intelligent system, nonlinear control and autonomous control, computer vision and application, power electronics and electrical drives.

Luchuan Liao received his B.E. in automation from College of Electrical Information Engineering of Panzhihua University in 2017. He is currently a graduate student in Sichuan University of Control Engineering, Chengdu, China. At present, his main research interests include computer vision, deep learning, object detection, and tracking.

Hao Wu received his B.E. degree in electronic and information engineering from Tsinghua University in 2008. He is a Ph.D. candidate in electronics and information in the Department of Electronic Engineering, Tsinghua University, Beijing, China. He is the general manger of Chengdu Sky Defence Technology Co. Ltd, Chengdu, China. His main research interest is the development of anti-drone system.

Jing Qin is currently an assistant professor in the Centre for Smart Health, School of Nursing, The Hong Kong Polytechnic University. He received his Ph.D. degree in Computer Science and Engineering from the Chinese University of Hong Kong in 2009. His research interests include medical image processing, virtual/augmented reality for healthcare and medicine training, deep learning, visualization and human–computer interaction and health informatics.

Ling He was born in China in 1981. She received the B.E. and M.S. degree from Sichuan University, Chengdu, China, in 2004 and 2007, respectively. The Ph.D. degree is received in 2011 from Royal Melbourne Institute of Technology University, Melbourne, Australia. She is currently an associate professor of College of Electrical Engineering, Sichuan University, Chengdu, China. Her research interests include speech signal processing, image processing, and machine learning.



Gang Yang received his M.E. degree in control theory and control engineering and Ph.D. degree in biomedical engineering from Sichuan University, in 2003 and 2010, respectively. He is currently an associate professor of College of Electrical Engineering, Sichuan University, Chengdu, China. His current research interests include bioelectromagnetic effects of stem cells and induction of differentiation, biomaterials, and regeneration of myocardial tissue repair.

Han Zhang received his B.S. degree in physics from University of Science and Technology of China and Ph.D. in optics from University of Arizona, in 2009 and 2014, respectively. He is currently invited associate researcher of College of Electrical Engineering, Sichuan University, Chengdu, China. His research interests include biomedical imaging and biomedical big data.

Jing Zhang was born in China in 1975. He received B.E. degree in automation from Tsinghua University and Ph.D. degree in electrical and computer engineering from the National University of Singapore, in 1999 and 2003, respectively. He is currently an associate professor of College of Electrical Engineering, Sichuan University, Chengdu, China. His research interests include image processing and virtual reality.