Journal of Computers Vol. 33 No. 3, June 2022, pp. 147-158
doi: 10.53106/199115992022063303012

# An Improved Algorithm for Moving Object Detection in YOLO UAV Videos

Juewen Hu[1] , Pei Wang[1*], Jian Yang[2], Longqiang Ni[3]

[1] School of Science, Beijing Forestry University, Beijing, China
{wangpei, hujuewen}@bjfu.edu.cn
[2] China Research and Development Academy of Machinery Equipment, Beijing, China
buaayangjian@126.com
[3] Northwest Institute of Mechanical & Engineering, Xianyang, China
shepherdni@163.com

Received 17 August 2021; Revised 13 December 2021; Accepted 13 January 2022

**Abstract.** Recently, moving object detection (MOD) in UAV (Unmanned Aerial. Vehicle) videos has been widely used in many fields. However, different objects and different algorithms often result in different detection accuracy. SSD (Single Shot MultiBox Detector) series and YOLO (You Only Look Once) version 5 are two popular object detection model, and their performance are always evaluated and compared with other improved method for optimizing detection accuracy. In this paper, an improved YOLO_v5 detection algorithm was proposed to further improve the detection accuracy. It adopted a cascaded inter-frame verification mechanism which is based on the neural network and uses spatial information and integrates object speed and direction as well to improve the detection accuracy of moving objects. To evaluate its performance, the open UAV video data from Stanford University was used to test the algorithms, and three types of moving objects were analyzed. The experimental results demonstrate that the proposed MOD method can improve the detection accuracy of small moving objects, which have a good application value, and can lay a foundation for subsequent related studies.

**Keywords:** moving object detection, neural network, SSD, YOLO, inter-frame verification algorithm

## 1 Introduction

Moving objects detection (MOD) is a key technique in many vision applications, such as motion recognition, human body detection, video surveillance and tracking, manual tracking, traffic surveillance and semantic annotation of video etc. Recently, the images and videos of UAV have been widely used in vast areas, such as high-altitude remote sensing image acquisition [1], geographic information collection [2], surveying and mapping system development [3], agricultural applications [4] etc. The growing interest in UAVs has made UAV MOD an hot topic, and more and more important for UAV applications [5].

UAV MOD usually include two main parts: UAV image pre-processing and moving object detection. UAV image pre-processing should be carefully considered due to lots of interference when UAV images are obtained. UAV image acquisition is often carried out at high altitudes, which inevitably results in lower clarity [6] and higher noise level [7]. Therefore, it is necessary to perform pre-processing on UAV images to reduce the noise. For instance, a UAV image defogging enhancement method was proposed, which was based on a wavelength-adaptive image formation model and geometric classification algorithm to generate a modified transmission map from the scattering coefficient, by which atmospheric light can be estimated, the visibility of the foggy UAV images can be significantly improved [7]. Simone Milani also proposed a low-complexity adaptive filtering strategy, which relies on the local image characteristics to protect low contrast areas due to excessive smoothing [8]. A fast and flexible denoising method using neural network (FFDnet) was proposed to denoise images by using the downsampled sub-images space, and this method can achieve a good balance between inference speed and denoising performance [9]. Video denoising neural network (VideNN) was proposed to denoise UAV images without prior knowledge of noise distribution. It can adapt to different noise models because it use a combination of spatial and temporal filtering [10]. In one word, the pre-processing of UAV images is important to improve the efficiency and accuracy of object detection.

It's worth to mention that the feature extraction in the pre-processed image series got more attention in MOD. Traditional image feature extractions mainly focus on static images, such as image registration, image fusion, image segmentation, and image classification etc., which essentially need large volume of calculation [11].

[*] Corresponding Author

However, the UAV image series need fast processing and light-weighted computing burden. The emergence of convolutional neural networks (CNN) has brought new vitality to image features extraction over the last decade. For example, AlexNet [12], Fast RCNN (Fast Region-CNN) [13], LSTM (Long Short-Term Memory) [14], and other algorithms have been developed, which adopted more and deeper neural networks and brought the image features extraction to a new level. Wenzhe Shi proposed an efficient sub-pixel CNN to improve the speed and quality of super-resolution reconstruction by introducing an effective sub-pixel convolutional layer and reducing the complexity of the system [15]. Mohamed A. Kassab used approaching and chasing networks to implement a real-time tracking control system based on full-vision depth object, which solves the problem of ambiguity between UAV yaw and lateral movement of moving objects [16]. Swathikiran Sudhakaran used LSTM with a pre-convolutional network to extract frame-level features, and proposed a simulated frame change model to improve the accuracy significantly [17]. Xiaohang Shi introduced a channel attention mechanism in YOLO_v5 to detect birds, which can make the model pay more attention to information-rich features, and ultimately improve the accuracy of small object detection [18]. Abbas B. Sadkhan proposed an improved Kalman filter method to track the moving target This method improved the initial parameters by using the intrusion weed optimization algorithm, which improved the detection efficiency and detection rate [19]. Xiang Zhang proposed another new moving target detection method based on the prior knowledge of the airport apron which can make the classification biased towards the foreground, so as to make up for the detection defect [20]. Irvine Valiant Fanthony used the YOLO network to help auto driving vehicles detect targets in real time with high accuracy and good results [21]. Obviously, the network cascade and migration played a major role on promoting the detection ability of UAV images.

Despite the above-mentioned efforts, there is still a room for improvement on the MOD because of serious interference, such as various types of noise and low resolution images, which hamper the detection accuracy [5] . Besides, different sizes and numbers of the moving objects, also introduced different impacts on the algorithm analysis and establishment of the model [6]. What's more, some researchers established models with a larger training-set than test-set which makes the model less robust [22].

In this paper, we present an improved algorithm based on Yolo network, which uses the motion characteristics of moving targets in UAV videos combining time domain and space domain information. The proposed algorithm not only effectively reduces the interference caused by low resolution of UAV videos, but also improves the robustness and efficiency due to the consideration of time-domain information collaborative identification.

This paper is organized as follows: The second section describes the improved method, and introduces the network cascading inter-frame verification algorithm. The third section demonstrates the experimental results and performance analysis. The fourth section discuss and analyzes the advantages and disadvantages of the proposed method. Finally, the last section draws some conclusions and look forward to the future work.

## 2 Methodology

In this section, we present a CNN-based cascaded inter-frame verification algorithm that is able to increase the detection accuracy of the moving object of the UAV image. The dataset is introduced in Section 2.1. Basic neural network methods are reviewed in Section 2.2 and our proposed inter-frame verification algorithm are described in Section 2.3. Finally, Section 2.4 demonstrated the performance evaluation.

### 2.1 Dataset

This study is based on the SDD (Stanford Drone Dataset) [23] of Stanford University's Computation Vision and Geometry Laboratory. The SDD contains videos include eight outdoor scenes, and each scene is composed of multiple videos of varying lengths taken by UAVs. The information about the dataset is summarized in Table 1. Moreover, each scene has the action trajectories of many different groups of people, such as pedestrians, cyclists, vehicles, etc. Eight videos from four scenes were used in this study, and some representative demo images are shown in Fig. 1. Fig. 1(a) to Fig. 1(d) are screenshots of the Bookstore scene, the Death Circle scene, Hyang scene, and the Gates scene respectively.

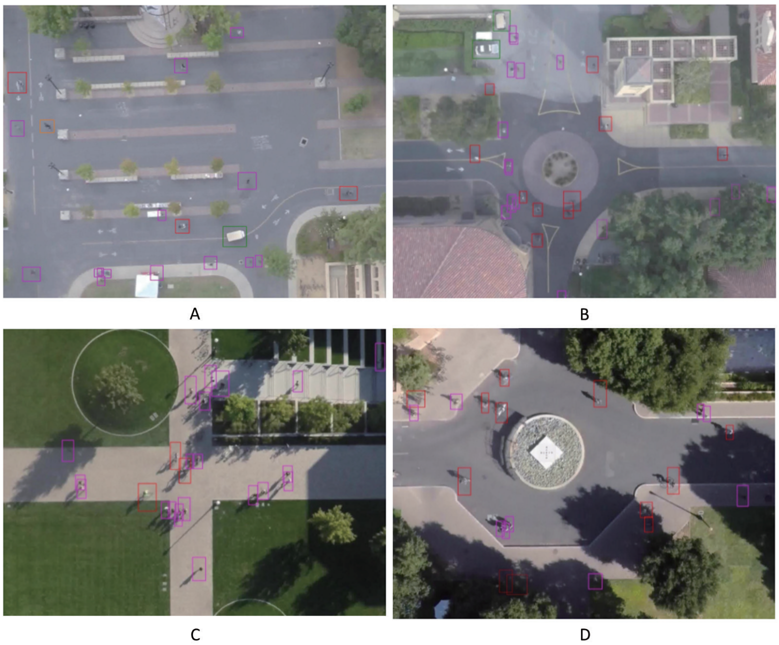

**Fig. 1.** Demonstration of several representative dataset samples [23]

**Table 1.** The summary information of the scene videos used in this study

| Scenes | No. of Videos | Agent percentage | | | | | |
|---|---|---|---|---|---|---|---|
| | | Bicyclist | Pedestrian | Skateboarder | Cart | Car | Bus |
| Gates | 9 | 51.94 | 43.36 | 2.55 | 0.29 | 1.08 | 0.78 |
| Little | 4 | 56.04 | 42.46 | 0.67 | 0 | 0.17 | 0.67 |
| Nexus | 12 | 4.22 | 64.02 | 0.60 | 0.40 | 29.51 | 1.25 |
| Coupa | 4 | 18.89 | 80.61 | 0.17 | 0.17 | 0.17 | 0 |
| Bookstore | 7 | 32.89 | 63.94 | 1.63 | 0.34 | 0.83 | 0.37 |
| DeathCircle | 5 | 56.30 | 33.13 | 2.33 | 3.10 | 4.71 | 0.42 |
| Quad | 4 | 12.50 | 87.50 | 0 | 0 | 0 | 0 |
| Hyang | 15 | 27.68 | 70.01 | 1.29 | 0.43 | 0.50 | 0.09 |

In pre-processing, each video was filtered to reduce the noise using the Gaussian filtering, and then split into frames and re-sampled to images with the size of 640*640 pixels. Considering the low proportion of some types of objects, we enhanced the image quality using cropping, random shift, and inversion measures while avoiding the overfitting. Then, the experimental data was divided into a training set, a verification set, and a test set at a ratio of 6:2:2 [24].

### 2.2 Basic Neural Network Algorithm

In order to analyze the performance differences of different networks for MOD, we select the two typical object detection networks SSD series and the YOLO series.

**Algorithm for SSD Series.** The SSD series algorithm is based on the Faster-RCNN model while YOLO algorithm uses a regression-based model to directly return the category and location of the object in a network. [25]. The classic pre-processing framework for SSD is shown in Fig. 2. In the detection process, many candidate regions are considered as ROI (Region Of Interest). The defined ROIs of different sizes are retrieved after different layers of convolution and pooling, and subjected to regression processing to get the positions, categories and scores [26]. Finally, NMS (Non Maximum Suppression) is used to process all the generated default boxes and output the results.

Inception_v2, Mobilenet_v2, Resnet50_v1 all belong to the internal classification network model of the SSD algorithm. The Inception_v2 have deeper inception structures than inception_v1 [27]. Moreover, it decomposes a relatively large convolution kernel into a small volume and asymmetric convolution kernels. Furthermore, the adopted batch_normal module makes the input data of each layer have the same mean and variance, increases the convergence speed of model training, and has a significant effect on the selection of activation function and the fine-tuning of learning rate [28]. Mobilenet_v2 is a lightweight CNN model [27], which model size is 1/30 of Inception and its speed is about 3 times of Inception [29]. While deep networks are prone to vanishing gradient, the emergence of Resnet50_v1 enables deeper networks to be better trained, and the residuals are used to reconstruct the mapping of the network, which is used to solve the problem that the training error becomes larger after increasing the number of layers [26].

In this paper, we modified the model as follows: Firstly, the three object detection models of the SSD series were pre-trained under the same dataset COCO (Common Objects in Context). Then we retained the trained convolutional layer and pooling layer, deleted the output classifier and modified the remaining model by deleting the last max-pooling layer of the internal classifier to the average-pooling layer, which can improve the sensitivity to global features by fine-tuning the transferred network parameters.

**Algorithm for YOLO_v5.** With the advent of new methods and technologies, YOLO iteration become faster, stronger, and better. Compared with other object detection methods, YOLO integrates object region prediction and object category prediction into a single neural network. YOLO is fast on detecting objects with high accuracy and suitable many actual application environments [30].

YOLO_v5 is the latest product of the YOLO series,whose pre-processing structure is shown in Fig. 3. YOLO_v5 series has been improved over YOLO series, and its running speed can reach to 140 frames per second. Moreover, the size of YOLO_v5 is nearly 90% smaller than the previous version [31].

The input end of YOLO_v5 adopts the Mosaic data enhancement method to improve the detection effect for small objects using random scaling, cropping and arrangement. The algorithm for the different datasets are implemented by the anchor boxes with different lengths and widths in the initial set. During training, the network outputs the prediction boxes based on the initial anchor boxes, and then compares it with the real boxes, calculates the loss between the two boxes, and then reverses the update to iterate the network parameters. Furthermore, two CSP (Cross Stage Partial) structures are designed in YOLO_v5. CSPNet splits the feature map into two parts. One part performs convolution operation, while the other part performs concatenation with the result of the previous part convolution operation. By integrating the gradient changes into the feature map from beginning to end, the accuracy is guaranteed to the greatest extent while reducing the amount of calculation [19].

**Evaluation of basic network.** As shown in Fig. 2, the UAV videos were pre-processed to filter out the noise and enhance the features. Meanwhile, the data stream format is converted to the network input format. Then the pre-processed data is put into the neural network for training and inference. In this step, we can obtain the training freezing graph or weight file of each neural network. According to the best training checkpoint, the performance of each network is evaluated to analyze moving objection detection in the dataset.

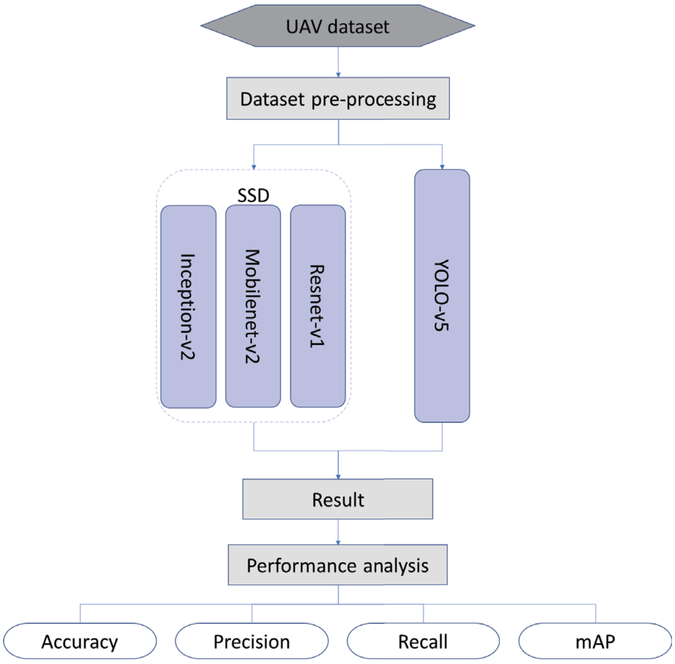

**Fig. 2.** The evaluation flowchart of the four networks

### 2.3 Improved YOLO Detection Algorithm

The features of moving objects not only exist in the spatial domain, but also in the time domain. Most pre-processing ways only focus on the image processing in the spatial domain. As an important feature, the speed of the moving objects is introduced using the proposed the inter-frame verification algorithm on the basis of YOLO_v5 to improve the MOD performance in UAV videos.

As shown in Fig. 3, YOLO_V5 is selected as the basic network because of its good performance in the above-mentioned evaluation. In the same way, the detection results of the basic network are checked by the proposed inter-frame verification algorithm, and the final results are evaluated using four indicators shown in Fig. 5.

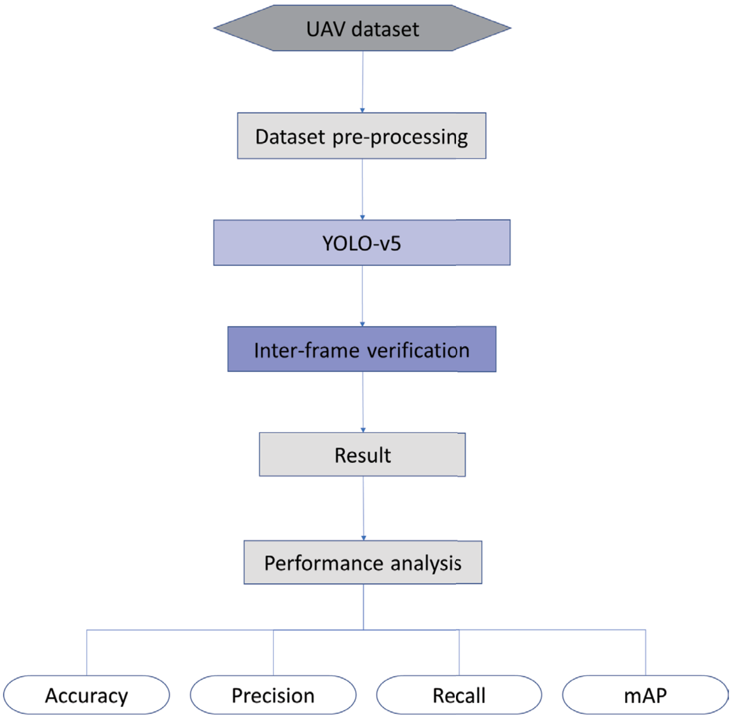

**Fig. 3.** The flowchart of the improved YOLO_v5

The specific process of the inter-frame verification algorithm is shown in Fig. 4. The speed and direction of the moving object are considered and used to increase the detection accuracy.

In our method, the basic YOLO_v5 network output the object detection box in every image. The coordinates of the center point of the object detection box in the current and subsequent image frames are $(x_1, y_1)$ and $(x_2, y_2)$. The speed of the object is calculated as follows:

$$v = \frac{\sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}}{\triangle t} \quad . \tag{1}$$

When the detected speed v is greater than the defined critical speed $v_c$, that is:

$$v > v_c \quad . \tag{2}$$

In this case, we consider that the object does not belong to the object category whose speed is lower than the critical speed, and the category with the highest probability will be selected as the detection result from the object category whose critical speed is higher than the moving speed. This paper uses pedestrians, cyclists, and cars as the detection objects. Typically, the normal pedestrian walking speed varied ed between 0.5~1.5m/s and the bicycle riding speed is between 2.8~4.2m/s. Therefore, combined with the dataset, we set the pedestrian critical speed as 0.32 pixel/$\Delta t$ and the bicycle critical speed as 0.90 pixel/ $\Delta t$ [32].

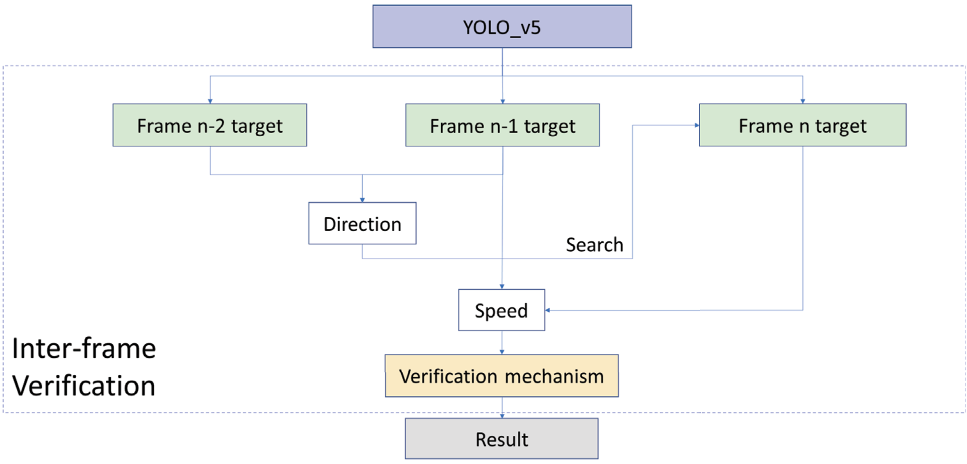

**Fig. 4.** The flowchart of the inter-frame verification

Due to the influence of the environment as the presence of many small objects in the UAV videos, it is challenging to determine the object speed. Considering the detection efficiency, we tracked the objects by narrowing the detection period and the probability direction angle. The object motion process is analogous to a random motion function, and its auto-correlation function is as follows:

$$R\left(t_1,t_2\right)=\int_{-\infty}^{\infty}\int_{-\infty}^{\infty}x_1x_2f_2\left(x_1,x_2,t_1,t_2\right)dx_1dx_2 \quad . \tag{3}$$

Obviously, it can be seen that the reduction of the detection period can increase the correlation between the front and rear positions $\left(x_1,y_1\right)$ and $\left(x_2,y_2\right)$ of the object, thereby maintaining the stability of object tracking. So we set the decision period $\Delta t$ as 0.03 seconds (one frame). Besides, there is a certain regularity in the movement of the object, and the detected direction angle can be calculated as follows:

$$\tan\theta=\frac{y_2-y_1}{x_2-x_1} \quad . \tag{4}$$

For the moving objects, we also supposed that the probability of moving direction angle tends to a Gaussian distribution with a mean of $\theta$ and a variance of $\frac{\pi^2}{9}$, which is:

$$F\left(x\right)=\frac{1}{\sqrt{2\pi}\frac{\pi}{3}}\int_{-\infty}^{x}e^{-\frac{(t-\theta)^2}{2\frac{\pi^2}{9}}} \quad . \tag{5}$$

So far, we can obtain the probabilities of all possible positions in the detection box, and the one with the highest probability is taken as the position of the tracking, while reducing the interference of other object motions.

The specific steps of the inter-frame verification are as follows (assuming that the nth frame object needs to be verified):

- Reading the object position of the n-2 frame and the n-1 frame of the object position, and calculating the object movement direction angle $\theta$.
- Taking the object detection frame of the n-1 box as the search range, the object position is determined according to the Gaussian distribution with the mean value $\theta$ and the variance $\frac{\pi^2}{9}$.
- Obtaining the object moving speed v from the object position in the n-1 frame and the object position in the n$^{th}$ frame.
- Verifying the relationship between the speed of the object and each critical speed, then setting the probability of the category lower than the object speed as 0, normalizing the probability of the remaining categories. Finally, taking the largest probability category as result.

### 2.4 Performance Evaluation

To evaluate the performance of the proposed method, accuracy, precision, recall, and Map (Mean Average Precision) are used, and the first three are calculated as follow:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} . \tag{6}$$

$$Precision = \frac{TP}{TP + FP} . \tag{7}$$

$$Recall = \frac{TP}{TP + FN} . \tag{8}$$

Where TP (True Positive) is the number of detection boxes with IoU (Intersection over Union) greater than 0.5 (the same ground truth is only calculated once), TN (True Negative) is the number of detection boxes with IoU less than or equal to 0.5 or the number of redundant detection boxes with the same ground truth detected, and FN (False Negative) is the number of ground truth not detected, FP (False Positive) is the number of false detection boxes [33].

Moreover, according to the precision and recall, the P-R curve can be drawn. AP is the area under the P-R curve of a certain category. Then mAP is calculated by using the P-R curve, which is the average area under the P-R curve for all categories [34].

## 3 Results

To evaluate the performance of the above-mentioned four neural networks, we trained each network for 30 epochs, and the training losses are shown in Fig. 5.

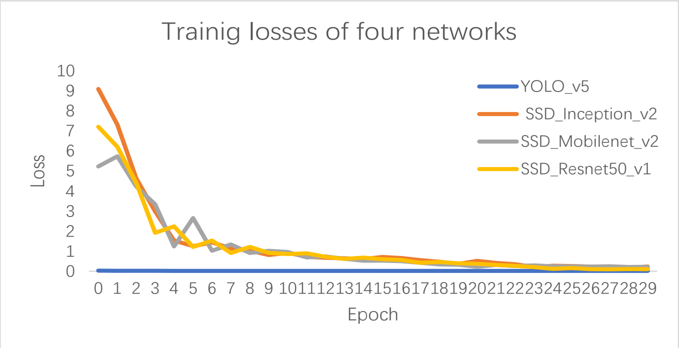

**Fig. 5.** Training losses of four networks

### 3.1 Basic Network Results

The accuracy, precision, recall, and AP (Average Precision) of different types of objects detected by the four neural networks are calculated and showed in Tables 2 to Table 5 respectively.

**Table 2.** Pedestrian detection evaluation of four networks

| | Accuracy | Precision | Recall | AP |
|---|---|---|---|---|
| SSD_Resnet50_v1 | 0.6148 | 0.6183 | 0.6378 | 0.6182 |
| SSD_Mobilenet_v2 | 0.4328 | 0.4341 | 0.4326 | 0.4339 |
| SSD_Inception_v2 | 0.4781 | 0.4810 | 0.4853 | 0.4811 |
| YOLO_v5 | 0.7253 | 0.7592 | 0.7436 | 0.7524 |

**Table 3.** Bicyclist detection evaluation of four networks

| | Accuracy | Precision | Recall | AP |
|---|---|---|---|---|
| SSD_Resnet50_v1 | 0.7618 | 0.7670 | 0.7712 | 0.7672 |
| SSD_Mobilenet_v2 | 0.6033 | 0.6204 | 0.6117 | 0.6197 |
| SSD_Inception_v2 | 0.6891 | 0.6903 | 0.6853 | 0.6891 |
| YOLO_v5 | 0.7490 | 0.7753 | 0.7594 | 0.7750 |

**Table 4.** Car detection evaluation of four networks

| | Accuracy | Precision | Recall | AP |
|---|---|---|---|---|
| SSD_Resnet50_v1 | 0.9835 | 0.9906 | 0.9889 | 0.9906 |
| SSD_Mobilenet_v2 | 0.4015 | 0.4037 | 0.4153 | 0.4043 |
| SSD_Inception_v2 | 0.9832 | 0.9874 | 0.9890 | 0.9884 |
| YOLO_v5 | 0.9512 | 0.9534 | 0.9351 | 0.9489 |

**Table 5.** Overall detection evaluation of four networks

| | Accuracy | Precision | Recall | mAP |
|---|---|---|---|---|
| SSD_Resnet50_v1 | 0.7867 | 0.7920 | 0.7893 | 0.7920 |
| SSD_Mobilenet_v2 | 0.4943 | 0.5017 | 0.5041 | 0.5017 |
| SSD_Inception_v2 | 0.7017 | 0.7039 | 0.7023 | 0.7038 |
| YOLO_v5 | 0.8085 | 0.8293 | 0.8127 | 0.8257 |

As shown in Table 2, YOLO_v5 had the best performance on pedestrians detection using the four indicators. The indicator values of the other three networks decrease in the order of SSD_Resnet50_v1, SSD_Inception_v2 and SSD_Mobilenet_v2. In particular, the evaluation indicators of SSD_Inception_v2 and SSD_Mobilenet_v2 are less than 50%.

Similarly, the performance of four methods on bicyclist and car detection are shown in Table 3 to Table 4. As shown in Table 3, SSD_Resnet50_v1 performed best and YOLO_v5 got the second place on bicyclist detection. The results of SSD_Resnet50_v1 and YOLO_v5 are very close, which is up to 77%. It can be found that in Table 4, SSD_Resnet50_v1 and SSD_Inception_v2 are the best two on car detection, and YOLO_v5 also got a similar performance. The three types of networks got very good performance on the car detection, which are about 95%-98%. Obviously, the three types of networks maintained high detection accuracy for cars and relative low detection accuracy for pedestrians and cyclists.

From the overall evaluation shown in Table 5, YOLO_v5 had the best comprehensive performance on three types of objects. However, the detection accuracy of YOLO_v5, fluctuates between 80% and 85%, which still have rooms to be improved.

To evaluate different networks in detail, some test results are shown in Fig. 6. It is noticed that different networks displayed different results even on the same test scene. The smaller and denser the objects are, the worse the detection performance are.

Specifically, the SSD_Resnet50_v1 network only missed two cyclists in scene A in the three test scenes. Using SSD_Inception_v2, a pedestrian was missed in the shade of scene A, and a pedestrian was missed in scene B, and a car was missed in the scene C. Especially, SSD_Inception_v2 is the only network that missed car object. The

SSD_Mobilenet_v2 network had the worst detection performance. Because of tree shades, two pedestrians were missed in scene A. Another two pedestrians were missed in scene B even without shadows. Three cyclists and two pedestrians were missed in scene C. YOLO_v5 performed best in the four networks. All objects in the three scenes were detected with a small flaw that a pedestrian was mistakenly detected as a cyclist in scene C.

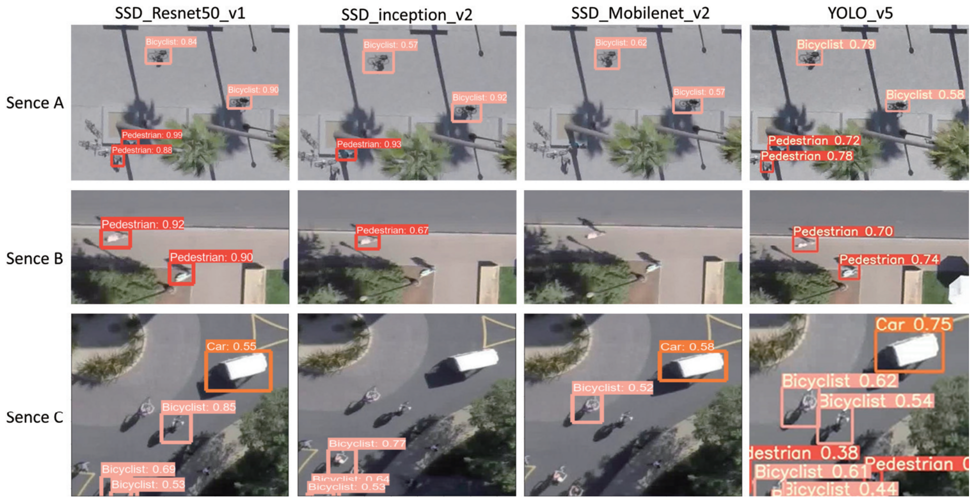

**Fig. 6.** The test of three scenes using four networks

### 3.2 Improved Method Results

According to the above evaluation, YOLO_v5 is selected as the basic network to improve the moving object detection. Similarly, the improved YOLO_v5 cascaded inter-frame verification algorithm was evaluated and the results are reported in Table 6.

**Table 6.** Detection evaluation of the improved YOLO_v5

| | Accuracy | | Precision | | Recall | | AP | |
|---|---|---|---|---|---|---|---|---|
| | Improved YOLO_v5 | YOLO_v5 | Improved YOLO_v5 | YOLO_v5 | Improved YOLO_v5 | YOLO_v5 | Improved YOLO_v5 | YOLO_v5 |
| Pedestrian | 0.9034 | 0.7253 | 0.9097 | 0.7592 | 0.9300 | 0.7436 | 0.9131 | 0.7524 |
| Bicyclist | 0.8850 | 0.7490 | 0.8893 | 0.7753 | 0.8901 | 0.7594 | 0.8899 | 0.7750 |
| Car | 0.9512 | 0.9512 | 0.9534 | 0.9534 | 0.9351 | 0.9351 | 0.9498 | 0.9489 |
| Average | 0.9132 | 0.8085 | 0.9175 | 0.8293 | 0.9184 | 0.8127 | 0.9176 | 0.8257 |

As shown in Table 6, by using the improved YOLO_v5, the accuracy, precision, recall, and AP have increased by 24.6%, 18.8%, 25.1%, and 21.4% for pedestrians and have increased by 18.2%, 14.7%, 17.2%, and 14.8% for bicyclists, respectively. The car detection performances are same using the two methods. The average overall accuracy, precision, recall, and AP have increased by 12.9%, 10.6%, 13.0%, and 11.1% respectively. Except for the car detection, the pedestrians and bicyclists were detected better significantly. And all evaluation indicators fluctuated around 90% for the improved YOLO_v5.

## 4 Discussion

Although the proposed method has achieved good results using the experimental dataset, it is needed to be improved. First, shadows are common problem in the images, which may greatly affect the actual detection as shown in Fig. 7, which may be solved by using more precise hardware to improve the quality of raw videos, or using some pre-processing measures. Second, the characteristics and parameter settings of neural network also affect the detection performance. Many attempts need to be done to get a better performance. Third, detecting small objects with high speed is a challenging problem, such as the running pedestrians and high-speed bicycles, which needs the improvements of algorithm.

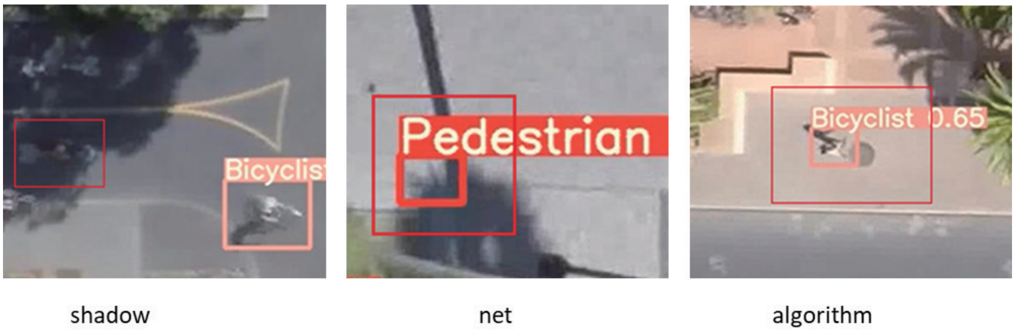

**Fig. 7.** Three error types

According to the training curves of the four neural networks shown in Fig. 8 SSD_Mobilenet_v2 (gray line) and SSD_inception_v2 (orange line) are close, while SSD_Mobilenet declines slower due to its lightweight characteristic for the adaption of mobile terminal deployment [27]. SSD_Resnet50_v1 decrease quickly in the SSD series networks because of the residual blocks in the Resnet network, which accelerates the initial mapping.

As for the YOLO_v5, BECLogits loss function is used to calculate the loss of the object score, and the binary cross-entropy loss function (BCEclsloss) is used to calculate the class probability score, and GIoU (Generalized Intersection over Union) [35] loss is used to calculate its Loss and IoU is set as the regression loss. Therefore, the value ranges of these functions are small, and the training loss of YOLO_v5 is the smallest of the four networks. As shown, the training loss image of the 25th to 29th epochs are enlarged as shown in Fig. 8.

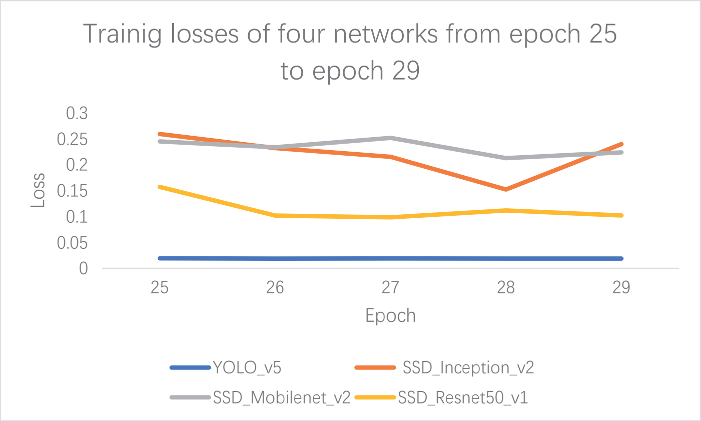

**Fig. 8.** Training losses of four networks from epoch 25 to epoch 29

According to the evaluation results, three basic networks performed well on car detection but performed relatively worse on the detection of smaller size objects such as pedestrians and cyclists. We presumed that the following factors can explain these problems well. First, for small size objects, mis-judgment is prone to occurs on a single frame due to limited image resolutions. Furthermore, noise can have strong impacts on the detection result. Modification of the network parameters for the specific dataset can improve the detection, but it is essentially complicated and will make the system less robust.

However, the proposed inter-frame verification utilized the motion information among more frames indeed increase the detection rate by, because it can provide additional information in the time domain. Moreover, the dependence on the single frame in the spatial domain can be reduced. Therefore, the detection accuracy of pedestrians and bicycles of the improved method increased significantly, while the detection accuracy of cars keep almost the same.

In the real application, the targets may stay still for a while. At this scenario, the proposed inter-frame verification mechanism will miss the targets for a moment with the occupation of computing resources and cannot be used to improve the detection accuracy. In this case, moving targets are also easy to be confused with nearby stand-still objects, which will cause verification errors and affect the final detection. Moreover, the crowded venue area is challenging for the proposed method, because lots of moving targets with small distances are hard to identify and compute accurately, which may cause incorrect detection results.

## 5 Conclusion

In the paper, a novel high-precision moving object detection method is proposed to process UAV images based on YOLO_v5 network. The inter-frame verification scheme is introduced to combine both the time domain information and the spatial domain information, which improves the detection rates for three kinds of targets. The results show that the proposed method is more accurate and more robust than simply using the four neural networks mentioned in the paper, which demonstrate us a better overall performance.

However, the time domain information helps the verification and judgment of the spatial domain information, which needs to be fused more deeply to make the proposed method more robust in our future work. Meanwhile, the joint judgment of the network output probability and the speed category probability will be considered in the verification mechanism. Shadow will also be carefully considered in analyzing the MOD of UAV images.

## Acknowledgements

The research is funded by the National Student Innovation and Entrepreneurship Training Program (No. 202010022300).

## References

[1] Q. Yang, L. Shi, L. Lin, Plot-scale rice grain yield estimation using UAV-based remotely sensed images via CNN with time-invariant deep features decomposition, in: Proc. IGARSS 2019- 2019 IEEE International Geoscience and Remote Sensing Symposium, 2019.

[2] I. Colomina, P. Molina, Unmanned aerial systems for photogrammetry and remote sensing: A review, ISPRS Journal of Photogrammetry and Remote Sensing 92(2014) 79-97.

[3] K. Choi, I. Lee, J. Hong, T. Oh, S.W. Shin, Developing a UAV-based rapid mapping system for emergency response, Proc. SPIE 7332, Unmanned Systems Technology XI (2009) 733209.

[4] F.G. Costa, J. Ueyama, T. Braun, G. Pessin, F.S. Osorio, P.A. Vargas, The use of unmanned aerial vehicles and wireless sensor network in agricultural applications, in: Proc. 2012 IEEE International Geoscience and Remote Sensing Symposium, 2012.

[5] A. Bouguettaya, H. Zarzour, A. Kechida, A. M. Taberkit, Vehicle Detection From UAV Imagery With Deep Learning: A Review, IEEE Transactions on Neural Networks and Learning Systems (2021) 1-21.

[6] X. Wang, W. Li, W. Guo, K. Cao, SPB-YOLO: An Efficient Real-Time Detector For Unmanned Aerial Vehicle Images, in: Proc. 2021 International Conference on Artificial Intelligence in Information and Communication (ICAIIC), 2021.

[7] I. Yoon, M.H. Hayes, J. Paik, Wavelength-adaptive image formation model and geometric classification for defogging unmanned aerial vehicle images, in: Proc. 2013 IEEE International Conference on Acoustics, Speech and Signal Processing, 2013.

[8] S. Milani, R. Bernardini, R. Rinaldo, Adaptive denoising filtering for object detection applications, in: Proc. 2012 19th IEEE International Conference on Image Processing, 2012.

[9] K. Zhang, W. Zuo, L. Zhang, FFDNet: Toward a Fast and Flexible Solution for CNN-Based Image Denoising, IEEE Transactions on Image Processing 27(9)(2018) 4608-4622.

[10]M. Claus, J. van Gemert, ViDeNN: Deep Blind Video Denoising, in: Proc. 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), 2019.

[11]L.M.G. Fonseca, L.M. Namikawa, E.F. Castejon, Digital Image Processing in Remote Sensing, in: Proc. 2009 Tutorials of the XXII Brazilian Symposium on Computer Graphics and Image Processing, 2009.

[12]A. Sevik, P. Erdogmus, E. Yalein, Font and Turkish Letter Recognition in Images with Deep Learning, in: Proc. 2018 International Congress on Big Data, Deep Learning and Fighting Cyber Terrorism (IBIGDELFT), 2018.

[13]H.S. Dikbayir, H.I. Bulbul, Deep Learning Based Vehicle Detection From Aerial Images, in: Proc. 2020 19th IEEE International Conference on Machine Learning and Applications (ICMLA), 2020.

[14]S. Hassan, A. Irfan, A. Mirza, I. Siddiqi, Cursive Handwritten Text Recognition using Bi-Directional LSTMs: A Case Study on Urdu Handwriting, in: Proc. 2019 International Conference on Deep Learning and Machine Learning in Emerging Applications (Deep-ML), 2019.

[15]W. Shi, J. Caballero, F. Huszar, J. Totz, A.P. Aitken, R. Bishop, D. Rueckert, Z. Wang, Real-Time Single Image and Video Super-Resolution Using an Efficient Sub-Pixel Convolutional Neural Network, arXiv:1609.05158. <https://arxiv.org/

abs/1609.05158>, September 2016 (accessed: 21.07.09).

[16]M.A. Kassab, A. Maher, F. Elkazzaz, Z. Baochang, UAV Target Tracking By Detection via Deep Neural Networks, in: Proc. 2019 IEEE International Conference on Multimedia and Expo (ICME), 2019.

[17]S. Sudhakaran, O. Lanz, Learning to detect violent videos using convolutional long short-term memory, in: Proc. 2017 14th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS), 2017.

[18]X. Shi, J. Hu, X. Lei, S. Xu, Detection of Flying Birds in Airport Monitoring Based on Improved YOLOv5, in: Proc. 2021 6th International Conference on Intelligent Computing and Signal Processing (ICSP), 2021.

[19]A.B. Sadkhan, S.R. Talebiyan, N. Farzaneh, Detection and Moving Object Tracking in images using an improved Kallman Filter (KF) by an Invasive weed optimization algorithm, in: Proc. 2021 International Conference on Advanced Computer Applications (ACA), 2021.

[20]X. Zhang, Y. Qiao, Y. Yang, S. Wang, SMod: Scene-Specific-Prior-Based Moving Object Detection for Airport Apron Surveillance Systems, IEEE Intelligent Transportation Systems Magazine (2021) 2-13.

[21]I.V. Fanthony, Z. Husin, H. Hikmarika, S. Dwijayanti, B.Y. Suprapto, YOLO Algorithm-Based Surrounding Object Identification on Autonomous Electric Vehicle, in: Proc. 2021 8th International Conference on Electrical Engineering, Computer Science and Informatics (EECSI), 2021.

[22]S. Jia, S. Jiang, Z. Lin, N. Li, M. Xu, S. Yu, A survey: Deep learning for hyperspectral image classification with few labeled samples, Neurocomputing 448(2021) 179-204.

[23]A. Robicquet, A. Sadeghian, A. Alahi, S. Savarese, Learning Social Etiquette: Human Trajectory Understanding In Crowded Scenes, in: Proc. Computer Vision – ECCV 2016, 2016.

[24]F. Cen, X. Zhao, W. Li, F. Zhu, Classification of Occluded Images for Large-Scale Datasets With Numerous Occlusion Patterns, IEEE Access 8(2020) 170883-170897.

[25]S. Zhai, D. Shang, S. Wang, S. Dong, DF-SSD: An Improved SSD Object Detection Algorithm Based on DenseNet and Feature Fusion, IEEE Access 8(2020) 24344-24357.

[26]G. Hao, Y. Yingkun, Q. Yi, General Target Detection Method Based on Improved SSD, in: Proc. 2019 IEEE 8th Joint International Information Technology and Artificial Intelligence Conference (ITAIC), 2019.

[27]Y. Zhao, J. Li, X. Li, Y. Hu, Low-Altitude UAV Imagery Based Cross-Section Geological Feature Recognition via Deep Transfer Learning, in: Proc. 2018 3rd International Conference on Robotics and Automation Engineering (ICRAE), 2018.

[28]L. J. Halawa, A. Wibowo, F. Ernawan, Face Recognition Using Faster R-CNN with Inception-V2 Architecture for CCTV Camera, in: Proc. 2019 3rd International Conference on Informatics and Computational Sciences (ICICoS), 2019.

[29]M. H. Kamrul, P. Paul, M. Rahman, Machine Vision Based Rice Disease Recognition by Deep Learning, in: 2019 22nd International Conference on Computer and Information Technology (ICCIT), 2019.

[30]J. Redmon, S. Divvala, R. Girshick, A. Farhadi, You Only Look Once: Unified, Real-Time Object Detection, in: Proc. 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016.

[31]Y. Liu, B. Lu, J. Peng, Z. Zhang, Research on the Use of YOLOv5 Object Detection Algorithm in Mask Wearing Recognition, World Scientific Research Journal 6(11)(2020) 276-284.

[32]S. Han, S. Yan, Z. Wang, Y. Zhang, Calibration of Hyperparameters for Pedestrian Flow Model Based on Bayesian Optimization, in: Proc. 2021 IEEE International Conference on Computer Science, Electronic Information Engineering and Intelligent Control Technology (CEI), 2021.

[33]S. Wang, Y. Han, J. Chen, Z. Zhang, G. Wang, N. Du, A Deep-Learning-Based Sea Search and Rescue Algorithm by UAV Remote Sensing, in: Proc. 2018 IEEE CSAA Guidance, Navigation and Control Conference (CGNCC), 2018.

[34]C. Wang, J. Wang, Q. Du, X. Yang, Dog Breed Classification Based on Deep Learning, in: Proc. 2020 13th International Symposium on Computational Intelligence and Design (ISCID), 2020.

[35]H. Rezatofighi, N. Tsoi, J. Gwak, A. Sadeghian, I. Reid, S. Savarese, Generalized Intersection over Union: A Metric and A Loss for Bounding Box Regression, arXiv:1902.09630. <https://arxiv.org/abs/1902.09630>, April 2019 (accessed: 21.07. 09)