Hindawi
Mobile Information Systems
Volume 2022, Article ID 1828848, 11 pages
https://doi.org/10.1155/2022/1828848

Research Article

YOLOv5-Based Vehicle Detection Method for High-Resolution UAV Images

Ziwen Chen,[1] **Lijie Cao**,[2] **and Qihua Wang**[3]

[1]*College of Mechanical and Power Engineering, Dalian Ocean University, China*
[2]*School of Information Engineering, Dalian Ocean University, China*
[3]*School of Medical Information Engineering, Jining Medical University, China*

Correspondence should be addressed to Lijie Cao; caolijie@dlou.edu.cn

Received 22 March 2022; Revised 2 April 2022; Accepted 6 April 2022; Published 2 May 2022

Academic Editor: Hasan Ali Khattak

Copyright © 2022 Ziwen Chen et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

To solve the feature loss caused by the compression of high-resolution images during the normalization stage, an adaptive clipping algorithm based on the You Only Look Once (YOLO) object detection algorithm is proposed for the data preprocessing and detection stage. First, a high-resolution training dataset is augmented with the adaptive clipping algorithm. Then, a new training set is generated to retain the detailed features that the object detection network needs to learn. During the network detection process, the image is detected in chunks via the adaptive clipping algorithm, and the coordinates of the detection results are merged by position mapping. Finally, the chunked detection results are collocated with the global detection results and outputted. The improved YOLO algorithm is used to conduct experiments comparing this algorithm with the original algorithm for the detection of test set vehicles. The experimental results show that compared with the original YOLO object detection algorithm, the precision of our algorithm is increased from 79.5% to 91.9%, the recall is increased from 44.2% to 82.5%, and the mAP@0.5 is increased from 47.9% to 89.6%. The application of the adaptive clipping algorithm in the vehicle detection process effectively improves the performance of the traditional object detection algorithm.

1. Introduction

With the rapid development of the social economy and accelerated urbanization, traffic problems are becoming more and more serious [1]. Effective traffic monitoring helps to solve increasingly serious traffic problems. Once AI enters Agenda at the national level, intelligent transportation systems will become the development in the trend [2–4]. An unmanned aircraft has wide application prospects in the field of transportation, and the UAV equipped with high-definition cameras has great development potential and advantages in parking lot management, intelligent traffic control, and disaster rescue [5–9]. Using the improved YOLO algorithm, according to the characteristics of fast recognition speed, high accuracy, and good detection effect, it can give full play to the advantages of auxiliary decision-making in a variety of complex traffic conditions.

Compared with vehicle detection through ground images, aerial image taken by UAV is slightly different: the ground view is mainly taken by a fixed camera. The aerial view is taken from the top view by a mobile UAV with a camera. Therefore, some side information about the vehicle is lost [10]. The image quality of the camera carried by the UAV is much higher than that of the ground camera (most cameras are 4 K, and some high-end models can output images with a resolution of 8 K), and the amount of information carried by the image is huge. Therefore, images need to be used correctly and reasonably. In addition, in aerial images, objects of interest are usually small and dense. For example, when a DJI Inspire 2 Zenmuse X7 drone is used, the output image size is 5760×3240 pixels; for such a high resolution, a vehicle may only be 50×50 pixels or less [11], and it is very challenging to detect such a small vehicle in large images.

In the field of deep learning algorithms, image classification networks based on convolutional neural networks such as AlexNet, VGG, and ResNet [12–15] have been developed to enhance ImageNet classification competition to achieve

higher scores. Convolutional neural networks have been increasingly used in the object detection field [16, 17]. Redmon et al. [18] proposed the You Only Look Once (YOLO) object detection network; it treats object detection as a regression problem and uses an end-to-end framework to directly predict category and location information. The following year, Redmon and Farhadi [19] proposed an improved version named YOLO9000, which added anchor boxes to make it easier for the detection head to predict the target box and added batch normalization (BN) to reduce the overfitting of the model. The most recent version of the YOLO object detection algorithm is YOLOv5, which significantly improves the accuracy and efficiency of the object detection algorithm by replacing the backbone to CSP-DarkNet and adding some data augmentation methods like mosaic.

Ground target detection based on the deep learning method has been well developed. However, the current technology still has some shortcomings in vehicle detection from UAVs, such as a small set of targets consisting of pieces of cars in parking lots. Taking the YOLO object detection network as an example, the downsampling factor of YOLO is 32, and the network outputs a 13×13 prediction grid. If the distance between two target objects is less than 32 pixels, then the network has errors when the targets are differentiated [11].

Therefore, some researchers are committed to improving the network structure. Zhong et al. [20] used convolutional neural networks to generate vehicle-like regions from the feature maps of different layers in the backbone and pooled the features of the deep and shallow layers, which is helpful to detect small objects more effectively. Yang et al. [21] used cross-layer skip connections to overcome the feature loss caused by deep convolutional neural networks for small objects. Sommer et al. [22] showed that the current region proposal network (RPN) did not work effectively for small objects, so the RPN network, including the fast R-CNN improvement, was used to detect small objects. The above researchers have conducted in-depth studies on network structures. However, due to the strict limitation of the input size of the convolutional neural network, the above algorithms are weak in terms of enhancing the vehicle detection process of high-resolution images.

Due to the limitations of the convolutional neural network, the current mainstream target detection network has strict requirements for the size of the input image. Different object detection networks have different requirements for the resolution of the input image. Images that do not meet the corresponding resolution need to be compressed or zero-padded and adjusted to meet the requirements before being detected again. The faster R-CNN [23] uses 1000×600 pixel images as the regular input, SSD [24] uses 300×300 or 512×512 pixel images as the input, and the latest YOLOv5 algorithm uses 640×640 pixel images as the input. However, the resolution of images captured by UAVs is much higher than the image size acceptable for the above object detection models. The loss in the process of image compression will seriously affect the detection of small targets in the target detection network.

In order to solve the problem of feature loss in the process of UAV high-resolution image target detection, an adaptive clipping algorithm based on UAV image as the input of training and detection is proposed in this paper. The algorithm is based on the YOLOv5 object detection network. During the process, high-resolution images are input to the network for training after being adaptively clipped according to the input size requirements. After training, the small object detection problem is transformed into a standard problem using a sliding window for sliding chunk detection through the step size calculated by the adaptive clipping algorithm. The algorithm is evaluated by using accuracy, recall, and map, and the effect of the algorithm is verified by testing actual vehicle detection images.

The rest of this paper is organized as follows: Section 2 presents the principles and implementation of the YOLOv5-based adaptive clipping algorithm, Section 3 describes the experimental procedure of the algorithm in this paper based on a modified VisDrone dataset, and Section 4 presents and analyzes the results of the operation of the proposed algorithm. Finally, a conclusion is drawn in Section 5.

2. Description of the Methodology

The workflow of the proposed YOLOv5-based high-resolution UAV image vehicle detection algorithm is shown in Figure 1.

The drone acquires high-resolution images or videos, which are processed to form an image library, organized into an initial training dataset using manual labeling, and split into a final training dataset after processing by the proposed adaptive clipping algorithm that is used to train the YOLOv5 object detection algorithm. The corresponding model weights are obtained.

The detection process uses the improved adaptive clipping detection algorithm to take chunks of the images on the test set. After obtaining the coordinate position of the current image's clipping detection frame, the coordinates are adjusted according to the sliding window step given by the adaptive clipping algorithm. Then, the adaptive clipping detection coordinate frame is merged with the coordinate frame of the original image detection after nonmaximum suppression. Finally, the complete object detection image is outputted.

3. The Proposed Adaptive Clipping Method

3.1. YOLOv5 Object Detection Algorithm. The proposed adaptive clipping algorithm applies to both the training data preprocessing process and the detection process of the object detection algorithm. The YOLOv5 algorithm, as the latest version of the YOLO algorithm, is known for its breakneck detection speed and high accuracy. Currently, the YOLOv5 model has a detection speed as low as 2 ms per image on a single NVIDIA Tesla v100. The proposed algorithm requires the input image to be detected in chunks and then combined into a single image; therefore, the YOLOv5 algorithm is

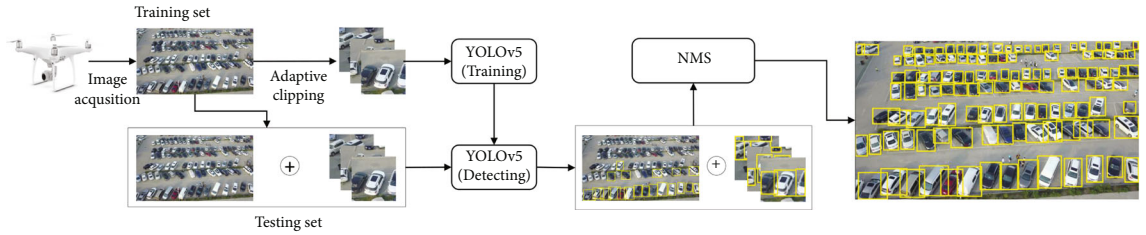

Figure 1: As can be seen from the figure, the vehicle images collected by the UAV are divided into a training set and a test set. The training set is processed by an adaptive clipping algorithm and then trained using YOLOv5. The images from the test set are adaptively chunked and fed into the improved YOLOv5 for detection. The detection results of both branches are nonmaximum suppressed to obtain the final detection frame.

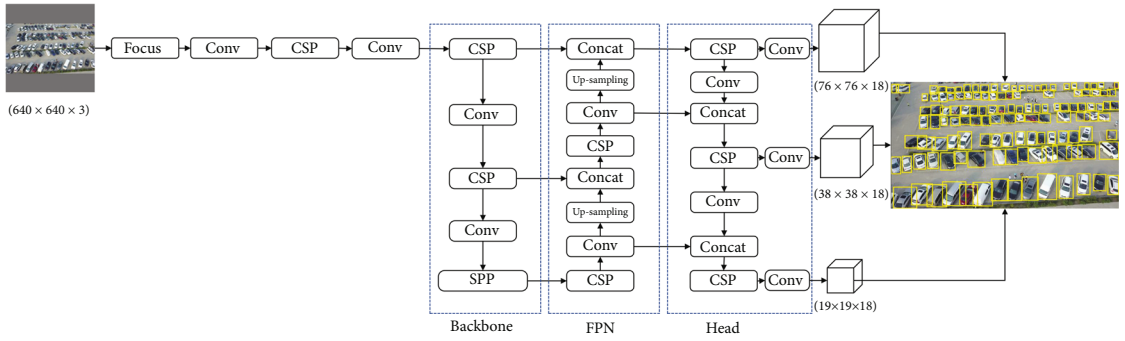

Figure 2: The structure of YOLOv5.

chosen as the object detection algorithm to ensure a high detection speed during real-time performance.

The YOLOv5 network model consists of three main structures: the backbone, the feature pyramid network, and the detection head. The backbone network is responsible for extracting features from different images at different scales, the feature pyramid network is responsible for fusing features from different scales and passing them to the detection network, and the detection network is responsible for predicting the object category in it using the image features and generating the object bounding box. The YOLOv5 network structure is shown in Figure 2.

3.2. Adaptive Clipping of Datasets. Taking a DJI Inspire 2 Zenmuse X7 UAV as an example, the maximum image size output by the camera is 5760×3240 pixels, and the size of a vehicle on the ground is only approximately 30-50 pixels when the UAV is flying at an altitude of 50-100 meters. The algorithm compresses the input image to 640×640 pixels during the object detection process. At this time, the length of the vehicle on the ground is only 4-6 pixels, and the image detail features of the vehicle suffer a large amount of loss. Figure 3 shows the detailed features of the vehicle in the same area before and after the compression of the original image.

In this paper, we propose an adaptive image clipping algorithm for the training set of high-resolution images captured by UAVs. In the process, the high-resolution images are slid and clipped with overlap according to the output size required by the object detection network to generate a new dataset after data augmentation. The clipping frame coordinates are calculated as follows:

$$N_w = \frac{I_w}{F_w} + 1, \tag{1}$$

$$N_h = \frac{I_h}{F_h} + 1, \tag{2}$$

$$S_w = F_w - \left[\frac{F_w - (I_w \% F_w)}{N_w}\right], \tag{3}$$

$$S_h = F_h - \left[\frac{F_h - (I_h \% F_h)}{N_h}\right], \tag{4}$$

where I_w denotes the number of horizontal pixels in the original image, I_h denotes the number of vertical pixels in the original image, F_w represents the width of the input image of the object detection network, F_h represents the height of the input image of the object detection network, N_w denotes the number of clip frames finally generated in the horizontal direction, and the calculation results in parentheses are rounded down, and N_h represents the final number of clip boxes generated in the vertical direction. The calculated results in parentheses are rounded down. S_w is the step length of the horizontal sliding of the clip frame, and S_h is the step length of the vertical sliding of the clip frame.

The workflow of the sliding window equations is shown in Figure 4. First, we calculate how many windows are

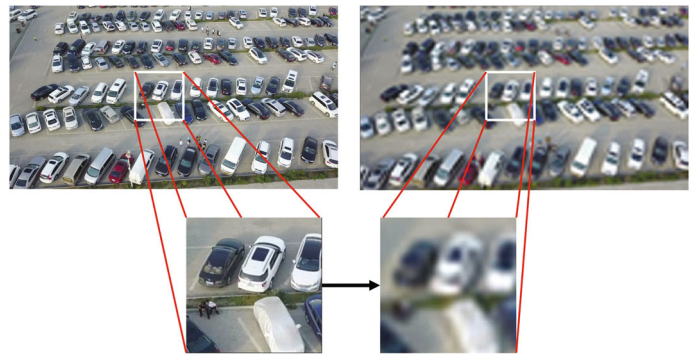

Figure 3: Detail damage in compressed images.

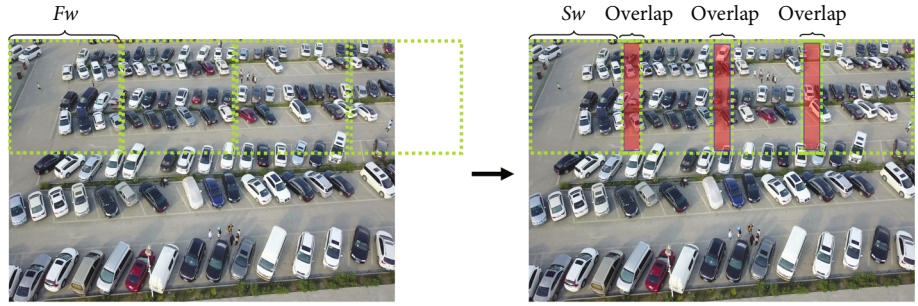

Figure 4: Detailed process of implementing the equation.

```
Input: ObjectBox(x1,y1,x2,y2),SlidingBox(x3,y3,x4,y4)
Output: ClippingBox(x-top,y-top,x-bot,y-bot)
  function Label mapping(x1,y1,x2,y2,x3,y3,x4,y4)
  if IoU([x1,y1,x2,y2], [x3,y3,x4,y4])>0 then
    x-top = max(x1,x3)
    x-bot = max(x2,x4)
    y-top = max(y1,y3)
    y-bot = max(y2,y4)
  end if
  return x-top,y-top,x-bot,y-bot
```

Algorithm 1: Label mapping.

needed to cover all the pixels at the current crop size according to Formulas (1) and (2). We allow the window to exceed a portion of the image. We then distribute the excess equally as the overlap of the sliding window in Formulas (3) and (4). Note that when the image size is just divisible by the sliding window, we add an extra window and then divide the entire window equally for overlap.

The label format of the YOLOv5 algorithm is the normalized relative coordinate value. For example, $(0.5, 0.5)$ represents the center point of an image, and $(1, 1)$ represents the point in the bottom right corner of an image. Therefore, the original labels need to be mapped according to the rules of adaptive clipping to generate the labels of the new image, and the algorithm flow of label mapping proposed in this paper is shown in Algorithm 1.

```
Input: Image, DetectSize
Output: PredictionBox
  function Detect (img, detect_size)
  Nw,Nh,Sw,Sh = Adaptive_clipping(img.size, detect_size)
  for h in range (Nh) do
    for w in range (Nw) do
      y3 = h * Sh do
      y4 = h * Sh + detect_size
      x3 = w * Sw
      x4 = w * Sw + detect_size
      clip_img = img[:,y3:y4,x3:x4]
      pred_clip = Model(clip_img)
      pred_clip[:,:,0] += x3
      pred_clip[:,:,1] += y3
      pred_all = Concat (pred_clip,pred_all)
    end for
    pred = Model(img)
    pred_all=Concat (pred, pred_all)
    pred_all=NMS (pred_all)
  end for
  return pred_all
```

Algorithm 2: Adaptive clipping detection.

The original object box $(x1, y1, x2, y2)$ represents the top-left and bottom-right coordinates of an object box in the original map, $(x3, y3, x4, y4)$ represents the top-left and bottom-right coordinates of the current sliding window, and $(x - \text{top}, y - \text{top}, x - \text{bot}, y - \text{bot})$ represents the top-left and bottom-right coordinates of the object box of an object

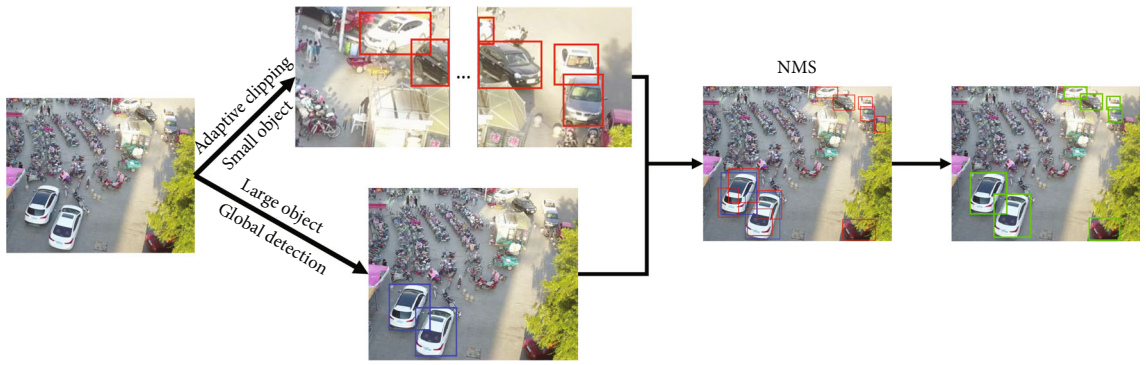

Figure 5: Adaptive clipping detection algorithm flow.

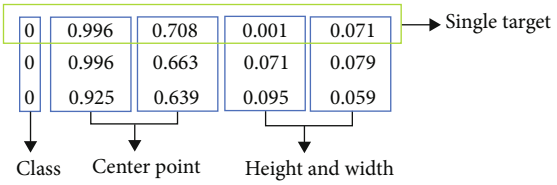

Figure 6: YOLO-TXT format.

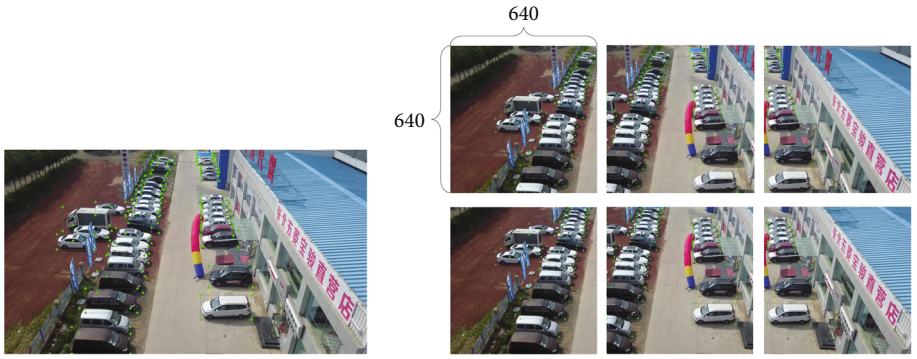

Figure 7: Adaptive clipping of datasets. The image on the left is the labeled original image, and the image on right is the labeled adaptive clipping image.

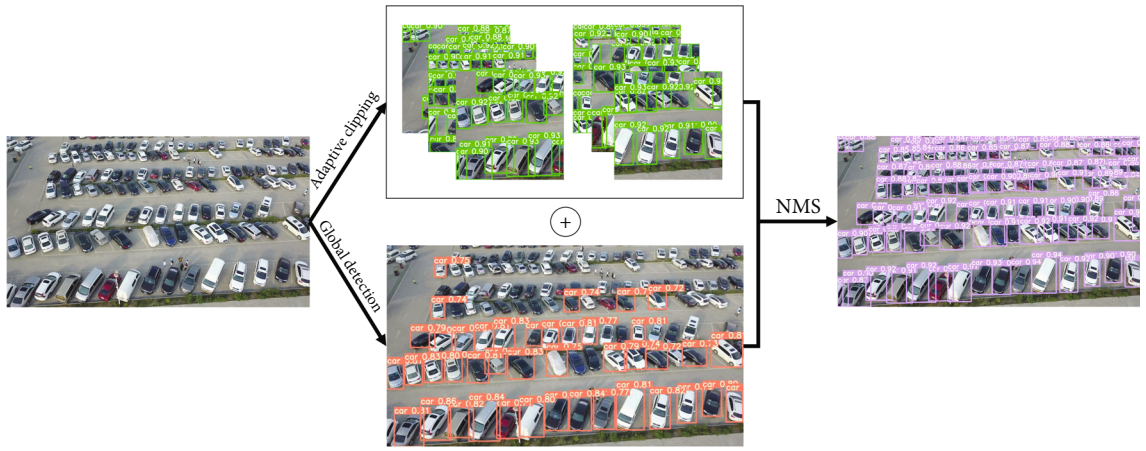

Figure 8: Algorithm detection process.

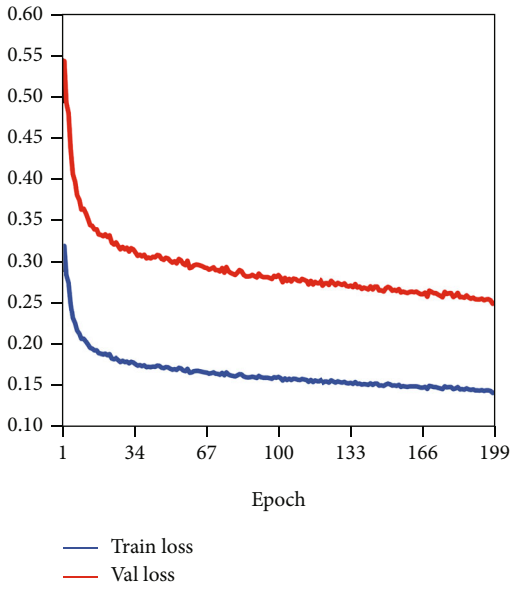

Figure 9: Loss function curves.

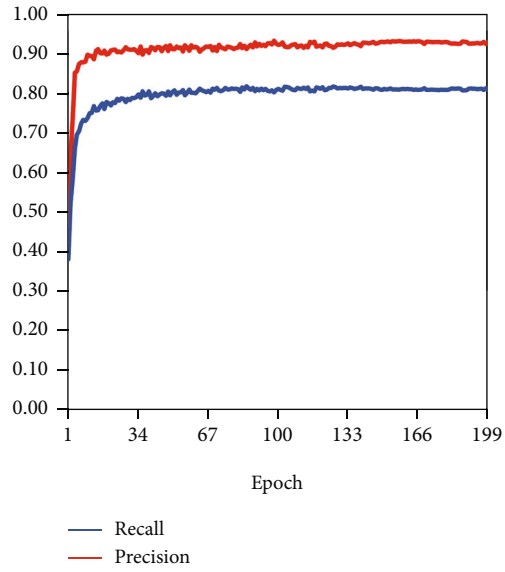

Figure 10: Recall and precision curves.

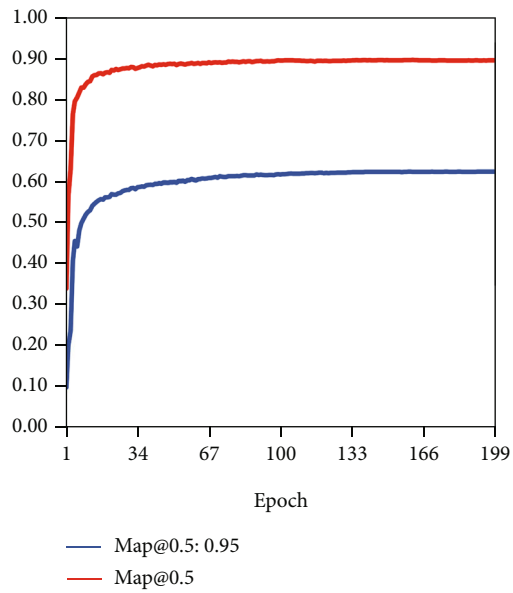

Figure 11: Mean average precision curves.

output from the clip map. IoU is the intersection over the union of the ratio discriminant function, which is responsible for calculating the ratio of the intersection over the union of two regions. IoU is calculated as

$$\mathrm{IOU} = \frac{\mathrm{Area}(A \cap B)}{\mathrm{Area}(A \cup B)}. \tag{5}$$

3.3. Adaptive Clipping Detection. The network structure of the YOLOv5 object detection algorithm has strict requirements concerning the resolution of the input raw images. The default input image size in YOLOv5 is 640×640; thus, all images larger than this resolution will be compressed, and image detail features are inevitably lost during the compression process.

This paper proposes adaptive clipping of images in the inference process using the adaptively clipped image coordinates calculated using Formulas (1)–(4) to address the above issues. The algorithm uses the input image width required by the network during the inference process F_w, as in Formula (1); the input image height required by the network F_h, as in Formula (2); and the calculated chunk detection frame coordinates to perform clipping with overlap on the original images and detect the clipped images separately. The algorithm flow is shown in Algorithm 2.

In Algorithm 2, img is the image input with the original resolution, and the clipped image size is the input image size of the object detection algorithm (640 in this paper). The output of the Adaptive_clipping function is calculated by Formulas (1)–(4). The Model function is the YOLOv5 network training model, which returns the prediction frame information of the input image. The Concat function is the combination function, which outputs the tensor after the combination of multiple tensors. Finally, the NMS function is the nonmaximum suppression function, which eliminates the redundant prediction frames by removing the object frame with the greatest overlap with the confidence value.

Since single images inevitably contain some large objects, to avoid detection errors caused by the incomplete combination of object features when a single large object is split into multiple clips, the algorithm inputs the whole image for inference after the inference of the clips. Finally, nonmaximum suppression is used for all inference results, including the clipped images and the whole images. The principle of this part of the algorithm flow is shown in Figure 5.

4. Experiments

The VisDrone drone dataset [25] was filmed and produced by the AISKYEYE team at Tianjin University, and the base dataset consists of 260,000 frames of video, with more than

Table 1: Comparison of different algorithms.

Model	mAP@0.5	mAP@0.5:0.95	Precision	Recall	Average speed (ms)
FRCNN	0.494	0.243	0.525	0.480	96
Cascade RCNN	0.586	0.330	0.592	0.538	117
YOLOv5(s)	0.479	0.261	0.795	0.442	46
FRCNN+AC	0.784	0.488	0.715	0.655	341
Cascade RCNN+AC	0.815	0.540	0.742	0.657	788
YOLOv5(s)+AC	0.896	0.624	0.919	0.825	212

10,000 still images from 14 different cities collected by various models of drones.

The VisDrone dataset is labeled with ten categories, namely, pedestrian, person, car, van, bus, truck, motor, bicycle, awning-tricycle, and tricycle. However, it suffers from an imbalance in the data distribution of different classes. To overcome this problem, uniform variables are used to verify the validity of the algorithm. We have removed the category labels for people and nonmotorized vehicles. According to various vehicle characteristics, retain only the car, van, bus, and truck categories, and unify the names of the above categories into one named car by modifying the labels. Facilitate the monitoring and identification of objectives. The adjusted training set has a total of 6471 images, the validation set has a total of 548 images, and a total of approximately 175,000 cars are labeled. We use an Intel I7-7700 CPU with 16 GB of memory and an NVIDIA RTX 2070 GPU (8 GB) for experiments, and the deep learning framework is Python 3.7 with PyTorch1.8.

4.1. Data Preprocessing Results. The training set is adaptively clipped using the proposed algorithm, and the clipping process discards the images that do not contain the object in the generated clipping map. The algorithm generates 35,742 images for the training set and 2656 images for the validation set. The labels of the clipped training set are reassigned using Algorithm 1 according to the YOLO-TXT format. The format requirements are shown in Figure 6.

Each image generates a txt file of the same name, and each line in the txt file represents the label of an individual object. The first column is the object class, numbered from 0. Since all classes were merged, only one class is included in the dataset. The second and third columns are the XY coordinates of the object frame, and the coordinate positions are normalized using the aspect pixel values of the original image as the denominator. The fourth and fifth columns are the aspect pixel values of the object frame, which are also normalized using the aspect pixel values of the original image as the denominator. The converted label image is shown in Figure 7.

4.2. Clipping Test Results. The YOLOv5 model is modified using Algorithm 2. We use transfer learning to initialize the model parameters, and the pretrained model is trained on the MS COCO dataset. The detection process of the algorithm is shown in Figure 8. We chunk the input image according to its size and the model's hyperparameters. The global detection branch takes the original image and infers it directly, while the chunking detection branch uses image chunks for detection. For example, the original map in Figure 8 is calculated using the algorithm to be divided into six blocks for inference. After the inference, the target boxes of the two detection branches are combined, and the redundant target boxes are removed using a nonmaximum suppression algorithm. The final result will be marked on the image at the end of the above process.

4.3. Comparative Tests. The transformed VisDrone dataset comprises 35,742 images in the training set and 2656 images in the validation set. The network parameters are updated using stochastic gradient descent (SGD). The learning rate uses the warm-up method and is updated by the cosine annealing algorithm. The number of iterations is set to 200, and the batch size is set to 16 for training.

To verify the generalization performance of the adaptive clipping algorithm, we compare the performance to the faster RCNN [23] and cascade RCNN [26] on the transformed VisDrone dataset. The experimental group uses the adaptive clipping algorithm to train and detect the data. In contrast, the control group uses the original algorithm to train and detect the high-resolution images directly.

5. Results and Analysis

We use different metrics, including the precision, recall, and mean average precision (mAP), to verify the effectiveness of the network. For a classification problem, the samples can be classified as true positives (TP), false positives (FP), true negatives (TN), and false negatives (FN) according to the combination of the ground truth and the prediction from the neural networks. The formulas for the precision and recall are shown in Formulas (6) and (7), respectively.

$$P = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP}} 100\%, \tag{6}$$

$$R = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}} 100\%. \tag{7}$$

The mAP is the average of the detection precision for all categories and is calculated as

$$\mathrm{mAP} = \frac{1}{n} \sum_{k=1}^{n} J(P, R)_k, \tag{8}$$

(a) (b) (c) (d) (e) (f) (g) (h) (i) (j)

Figure 12: Continued.

(k) (l)

Figure 12: Detection result comparison.

where $J(P, R)$ is the average precision function, which is calculated using the current category number k. The precision rate P with the recall rate R forms the P-R area under the curve. n is the total number of categories, and k is the current category.

5.1. Analysis of Model Training Results. The loss function is used to determine the training state of the model in the current iteration and to calculate the difference between the predicted and true values during the iteration. The YOLOv5 loss function is calculated as

$$\text{Loss} = l_{\text{object}} + l_{\text{box}} + l_{\text{class}}, \tag{9}$$

where l_{object} is the confidence loss, l_{box} is the bounding box loss, and l_{class} is the category loss. Since there is only one class in the training set of this paper, l_{class} is 0. The loss function curve of the training process is shown in Figure 9.

As the loss curve shows, at 200 rounds, the curve essentially stops decreasing, and the network training is essentially complete. The value of the loss function of the training set decreased from an initial 0.3187 to approximately 0.1397, and the value of the loss function of the validation set decreased from an initial value of 0.5425 to 0.2487.

The precision measures how accurate a model is at recognizing an object. The recall rate is how much a model searches for the entire object when recognizing the object. Figure 10 shows the variation of the precision and recall during the training of the model according to the number of epochs. The highest precision achieved by the model during training is 0.93087, and the highest recall is 0.8169.

The mAP is an evaluation metric that assesses network performance in the object detection field. mAP@0.5 is the area under the P-R curve of the network when setting the detection IOU ratio threshold to 0.5. mAP@0.5:0.95 is the average value of the area under the P-R curve of the network when setting the detection positive case intersection and ratio threshold from 0.5 to 0.95, calculated individually at a step size of 0.05. Thus, mAP@0.5:0.95 is harder to achieve. Figure 11 shows the mAP curve during training. The final mAP@0.5 achieved by the algorithm is 0.894, and the mAP@0.5-0.95 is 0.623.

As shown in Table 1, we compare the original dataset with the data processed by the proposed algorithm, the faster RCNN, the cascade RCNN, and YOLOv5. The results show that before using the proposed algorithm, the mAP of the cascade RCNN exceeds the faster RCNN and YOLOv5, and the precision and inference of YOLOv5 improve over time. After the adaptive clipping algorithm is used, the parameters of all three object detection frameworks are improved to some extent, and our algorithm outperforms the other two algorithms in all metrics. The inference time is controlled within an acceptable range.

5.2. Analysis of Detection Results. To prove the rigor of the analysis, 500 images in the test set that are not involved in training are used for testing. The detection function provided by the original YOLOv5 algorithm and the improved adaptive clipping detection function are applied to the test set. The detection results are evaluated based on the label value calculation. The detection results, which are presented in Table 1, show that the original model has significant feature losses due to the input image compression problem when detection is performed on high-resolution images; therefore, the detection results of the original model are lower than those of the model with the proposed algorithm in all indices.

Figure 12 shows a comparison of the detection effect between the proposed algorithm and the original algorithm. (a–c) and (g–i) are the detection effects of the proposed algorithm, and (d–f) and (j–l) are the detection effects of the original algorithm. In (a–f), in which the UAV flies at a low altitude and is tilted, the vehicle object size is approximately 100 pixels in the close view and only 30 pixels or less in the far view. (d–f) Show that the original algorithm has a good detection effect for near vehicles, but for far vehicles, a large area is not detected. The proposed algorithm can detect both small objects at a distance and large objects nearby because of the adaptive clipping of the detection images. The images detected in (g–l) are images taken at high altitudes, and the object size is generally smaller than 50 pixels. At this point, the advantage of the proposed algorithm becomes apparent. (g, j) Show that the original algorithm detects only two buses and one car as large objects. In contrast, the proposed adaptive clipping detection algorithm detects all 45 vehicles. The second figures in (h, k) show the detection effect of large dense objects. Because the objects are too small and dense, the original algorithm detects only one vehicle, while the proposed algorithm detects 255 objects, accounting for 95.1% of all 268 objects. The vehicle targets in (i, l) are smaller than 30 pixels in size.

The original algorithm did not detect any targets, while the algorithm in this paper detected 50 targets, including all 48 objects plus some false positive detections.

6. Conclusion

This paper proposes a vehicle detection method based on high-resolution images captured by UAVs, which addresses that traditional object detection algorithms are limited by images and object size. High-resolution images can limit the performance of the network when detecting small targets. So, we take the YOLOv5 object detection algorithm as the baseline. And we proposed an adaptive clipping algorithm of high-resolution images during data preprocessing and detection to detect small object vehicles. We introduce evaluation indices such as precision, recall, and mAP to evaluate the performance of the algorithm and design comparison experiments to verify the algorithm's effectiveness. The conclusion of improving the resolution of the UAV aerial image is obtained.

The framework detection speed determines the vehicle detection efficiency and real-time performance during UAV operations, so improving the operating speed of the algorithm is the goal of future research. Furthermore, in subsequent research, the single-scale object detection process for the object detection network and the network model structure can be improved, for example, by using model pruning, backbone structure optimization, and reparameters. Therefore, UAVs can be widely used in intelligent traffic management.

Data Availability

The data underlying the results presented in the study are available within the manuscript.

Conflicts of Interest

There is no potential conflict of interest in our paper, and all authors have seen the manuscript and approved to submit it to your journal.

Acknowledgments

This was supported in part by the Natural Science Foundation of Liaoning Province of China (2019-ZD-0731), the Startup Project of Doctoral Scientific Research (HDBS201802), and the Liaoning Provincial Education Department Scientific Research Funding Project (LJKZ0731).

References

[1] C. Lu, H. Zhang, and M. Chen, "Realization of high-quality development of transportation in the new era," *China Journal of Highway and Transport*, vol. 34, no. 6, p. 1, 2021.

[2] R. Li and C. Wang, "Department of advanced traffic management systems," *Journal of Tsinghua University(Science and Technology)*, pp. 1–7, 2021.

[3] G. Guo, Y. Xu, T. Xu, D. Li, Y. Wang, and W. Yuan, "A survey of connected shared vehicle-road cooperative intelligent transportation systems," *Control and Decision*, vol. 34, no. 11, pp. 2375–2389, 2019.

[4] X. Kong, J. Zhang, L. Deng, and K. Liu, "Research advances on vehicle parameter identification based on machine vision," *China Journal of Highway and Transport*, vol. 34, pp. 13–30, 2021.

[5] A. Tian, X. Cai, W. Chen, W. Luo, and Y. Yin, "Vehicle illegal parking detection and evidence collection system on uav," *Measurement & Control Technology*, vol. 40, pp. 67–74, 2021.

[6] B. S. Ali, "Traffic management for drones flying in the city," *International Journal of Critical Infrastructure Protection*, vol. 26, article 100310, 2019.

[7] D. Câmara, "Cavalry to the rescue: drones fleet to help rescuers operations over disasters scenarios," in *2014 IEEE Conference on Antenna Measurements & Applications (CAMA)*, pp. 1–4, Antibes Juan-les-Pins, France, 2014.

[8] S. Srivastava, S. Narayan, and S. Mittal, "A survey of deep learning techniques for vehicle detection from uav images," *Journal of Systems Architecture, page*, vol. 117, p. 102152, 2021.

[9] A. Bouguettaya, H. Zarzour, A. Kechida, and A. M. Taberkit, "Vehicle detection from uav imagery with deep learning: a review," *IEEE Transactions on Neural Networks and Learning Systems*, pp. 1–21, 2021.

[10] M. Cao, H. Ji, Z. Gao, and T. Mei, "Vehicle detection in remote sensing images using deep neural networks and multi-task learning," *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. 2, pp. 797–804, 2020.

[11] A. Van Etten, "You only look twice: rapid multi-scale object detection in satellite imagery," 2018, https://arxiv.org/abs/1805.09512.

[12] Y. LeCun, B. Boser, J. Denker et al., "Handwritten digit recognition with a back-propagation network," *Advances in Neural Information Processing Systems*, vol. 2, 1989.

[13] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Advances in neural information processing systems*, vol. 60, pp. 84–90, 2012.

[14] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2014, https://arxiv.org/abs/1409.1556.

[15] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778, Las Vegas, NV, USA, 2016.

[16] Z.-Q. Zhao, P. Zheng, S.-t. Xu, and X. Wu, "Object detection with deep learning: a review," *IEEE transactions on neural networks and learning systems*, vol. 30, no. 11, pp. 3212–3232, 2019.

[17] Z. Wang and L. Jun, "A review of object detection based on convolutional neural network," in *2017 36th Chinese Control Conference (CCC)*, pp. 11104–11109, Dalian Liaoning, China, 2017.

[18] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: unified, real-time object detection," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 779–788, Las Vegas, NV, USA, 2016.

[19] J. Redmon and A. Farhadi, "Yolo9000: better, faster, stronger," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 7263–7271, Honolulu, HI, USA, 2017.

[20] J. Zhong, T. Lei, and G. Yao, "Robust vehicle detection in aerial images based on cascaded convolutional neural networks," *Sensors*, vol. 17, no. 12, article 2720, 2017.

[21] M. Y. Yang, W. Liao, X. Li, and B. Rosenhahn, "Deep learning for vehicle detection in aerial images," in *2018 25th IEEE International Conference on Image Processing (ICIP)*, pp. 3079–3083, Athens, Greece, 2018.

[22] L. W. Sommer, T. Schuchert, and J. Beyerer, "Fast deep vehicle detection in aerial images," in *2017 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pp. 311–319, Santa Rosa, CA, USA, 2017.

[23] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: towards real-time object detection with region proposal networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, pp. 1137–1149, 2017.

[24] W. Liu, D. Anguelov, D. Erhan et al., "Ssd: single shot multibox detector," in *European Conference on Computer Vision*, pp. 21–37, 2016.

[25] P. Zhu, L. Wen, D. Du, X. Bian, Q. Hu, and H. Ling, "Vision meets drones: past, present and future," 2020, https://arxiv.org/abs/2001.06303.

[26] Z. Cai and N. Vasconcelos, "Cascade r-cnn: delving into high quality object detection," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6154–6162, Salt Lake City, UT, USA, 2018.