Vehicle-specific retrieval based on aerial images

Guoliang Yang, Zhigang Tang* and Jianchen Huang

Department of Electrical Engineering and Automation, Jiangxi University of Science and Technology, Ganzhou, Jiangxi, 341000, China

*Corresponding author’s e-mail: 1397414695@qq.com

Abstract. UAVs (Unmanned Aerial Vehicle) are currently widely used in civil and military fields. Collecting ground information through UAVs is their main application direction. This paper proposes a vehicle retrieval system based on aerial images, which can retrieve the corresponding target vehicle according to the set conditions. For the retrieval algorithm, we optimized its feature extraction network based on YOLO v3, and optimized it as a two-way residual based on the original residual module to improve its ability to detect small targets. Considering that the position deviation of the prediction frame in the actual detection will greatly affect the detection result, the GIoU position calculation index is adopted to improve the network's sensitivity to the position information. At the same time, retrieval conditions such as vehicle type and color are added on the basis of detection, so that the model can complete the retrieval of specific vehicle targets. The test results of the finally optimized network on the VEDAI dataset reached 91.2% mAP.

1. Introduction

UAVs have been popular in our lives from the initial military use to the present. In recent years, the rapid development of drones has made their application areas not limited to military operations, terrain surveys, and high-altitude gas detection. Drones rely on their high-altitude advantages to obtain some perspectives that we usually cannot obtain. Aerial images obtained by drones can contain more ground information. Although the remote sensing images taken by satellites are similar to high-altitude aerial photos taken by drones, they contain rich ground information, but the image quality of remote sensing images is not ideal due to the influence of external factors such as clouds and light. The addition of additional noise makes convolution. The neural network has problems such as missed detection and false detection. The drone can improve the overall imaging quality by changing the position and aerial photography height, and it is flexible and easy to use. Therefore, more and more high-altitude image acquisition methods are obtained by aerial photography of mobile devices such as drones. Aerial target detection is mainly aimed at aerial images, and detects targets of interest, such as vehicles, ships, roads, and natural landforms. The aerial image can clearly express the geographical form, and it is less affected by the terrain and ground architecture, and can hold more information. However, because of the small proportion of the target in the image due to high-altitude shooting, it is difficult to extract target features, resulting in poor detection results.

Although the concept of convolutional neural networks was proposed as early as the 1980s, it was limited by the hardware equipment at the time and could not complete the calculation of large-scale convolutional neural networks, making it not a research hotspot. In recent years, the rapid development of GPU hardware technology can rely on its high-bandwidth, multi-threaded parallel computing capabilities to easily complete inference calculations on a complex network. Especially in
the 2012 ILSVRC competition, Alex Krizhevsky won the championship with the AlexNet model, and brought the field of deep learning back to everyone's perspective. The automatic learning ability of convolutional neural network for target features is more robust and the detection effect is better than that of the traditional target detection method through artificially designed features. Therefore, we use YOLO v3 convolutional neural network to detect vehicles in aerial images according to the retrieval conditions. Compared with other vehicles, we can further narrow the detection range by setting the rigid characteristics of the vehicle, and the detection results obtained will have more Targeted.

2. Small target detection and problem analysis based on YOLO v3

The YOLO series of algorithms rely on the end-to-end network structure, and can complete the regression of target locations and category predictions in a network. The overall model has a small amount of calculations and can quickly complete inference, but it sacrifices detection accuracy. Based on the actual needs, we hope to be able to detect the video returned by the drone in real time, and require the algorithm to reason quickly. And different from the RCNN series of algorithms based on sliding window target detection, YOLOs target detection is for a complete feature map, so it can reduce the false detection due to the background of the image, especially the small target area in aerial images Most of them are background images. In summary, in the end, we chose YOLO v3 algorithm as the basic algorithm for this design, and optimized its network structure to improve the detection ability for small targets.

2.1. Principle of YOLO v3

The YOLO algorithm inputs a complete picture into an end-to-end network. Because the image is not segmented, the detection is performed based on a complete feature map. The feature map input to the YOLO detector is first divided into S * S grids. When the center point of a potential target is in a certain grid, the grid will generate B bounding boxes to predict them, and each bounding box has its own confidence. At the same time, each bounding box is composed of five parts: (x, y, w, h, and c). (x, y) represents the center position of the bounding box relative to the grid. And height; c indicates the ratio of the area of the boundary prediction box to the actual boundary box. In addition to the prediction of position information, the boundary prediction box also predicts the category of the target in the box. The final confidence score of the boundary prediction box is composed of the prediction results of position and category. The calculation formula is as follows:

\[ C = \Pr(\text{class}_i) \times \Pr(\text{object}) \times \text{IOU}_{\text{pred}}^{\text{truth}} = \Pr(\text{class}_i) \times \text{IOU}_{\text{pred}}^{\text{truth}} \]

In the above formula, \( \Pr(\text{class}_i) \) Indicates the category prediction score for the target in the box; \( \text{IOU}_{\text{pred}}^{\text{truth}} \) The ratio of the predicted bounding box to the actual bounding box area is the location prediction score.

Compared with YOLO v1 and YOLO v2, YOLO v3 has greatly improved the feature extraction network structure[1]. First of all, the idea of residual network is borrowed, and the residual module is constructed through the fast link layer (Figure 1), so that the total number of layers in the network reaches 106 layers[2]. The more network layers, the stronger the ability to extract features, but blindly increasing the number of network layers will cause problems such as disappearance of gradients during training, and the residual module solves the network degradation during training by means of identity mapping. This problem also makes YOLO v3 with more network layers have stronger feature extraction capabilities.
In addition, in order to improve the detection ability of small targets, YOLO v3 uses a multi-scale detection mechanism, which is set up with 13 * 13, 26 * 26, and 52 * 52 three grid division scales. Among them, the 52 * 52 scale is responsible for detecting small targets. The addition of fine-grained scale detection effectively reduces the occurrence of missed detection of small targets and improves the overall network robustness.

2.2. Test and analysis of YOLO v3 algorithm based on aerial image
In order to determine YOLO v3’s ability to detect small targets in aerial images, we use the VEDAI data set as the training and testing set for this experiment, and at the same time train the YOLO v3-tiny network model as a reference. The test results are shown in Table 1.

| Model       | mAP  |
|-------------|------|
| YOLO v3     | 76.5%|
| YOLO v3-tiny| 71.7%|

Refer to the test results of YOLO v3 and YOLO v3-tiny on the COCO dataset. The test results are shown in Table 2.

| Model       | mAP  |
|-------------|------|
| YOLO v3     | 55.3%|
| YOLO v3-tiny| 33.1%|

Compare the test results of the two data sets. In the comparison of the test results of the VEDAI data set, the detection accuracy of YOLO v3 is not much higher than that of YOLO v3-tiny.

Because YOLO v3 uses a multi-scale detection mechanism, the object detected based on the actual application scene is a small target. In order to verify the ability of three scales to detect small targets in aerial images, we split YOLO v3 into three single scales for testing. The test results are shown in Table 3.

| Model       | mAP  |
|-------------|------|
| YOLO v3-13*13| 33.4%|
| YOLO v3-26*26| 48.6%|
| YOLO v3-52*52| 70.2%|

It can be seen from the above table that the detection accuracy of 13 * 13 and 26 * 26 scales is much lower than 52 * 52 scales. It shows that the detection of small targets is mainly performed on the 52 * 52 scale, and the contribution rate of the 13 * 13 and 26 * 26 scale detection is not large.

3. Optimization based on YOLO v3 algorithm
According to the test results and analysis in section 1.2, we mainly optimize the YOLO v3 algorithm from the following aspects. First, streamline the overall network structure to avoid loss of features caused by excessively deep network layers; Second, add more fine-grained YOLO detection to further...
improve the detection ability for small targets; Third, replace the original IOU calculation method with the GIOU calculation method to optimize the accuracy of the target position prediction.

3.1. Network Structure Optimimization
The reason why the original YOLO v3’s backbone network is deeper is that the residual module used by it is smaller. Each scale feature detection has multiple residual modules for stacking. First, we perform the convolution structure in the residual module. In a large residual module, there are residual modules of two receptive fields. The structure of the optimized residual module is shown in Figure 2.

![Figure 2. Two-way residual module](image)

It can be seen from the above figure that when a feature map is input into a residual module, two small residual modules will perform feature extraction operations on it simultaneously. In the residual module on the left, the receptive field size of its convolution kernel is $3 \times 3$, which mainly extracts detailed features. The size of the convolution kernel receptive field in the residual module on the right is $5 \times 5$, which is responsible for extracting the overall features of the target. Finally, the $1 \times 1$ convolution layer is used to extract the two channels to the features for fusion, so as to obtain more comprehensive features[3]. At the same time, in order to reduce the amount of calculation caused by the increase of the module structure, we perform a low-rank decomposition on the $5 \times 5$ convolution kernel and decompose it into two serial convolution kernels of $5 \times 1$ and $1 \times 5$. The low-rank decomposition method can greatly reduce the calculation amount, and the effect of feature extraction is the same as the effect before decomposition[4]. The test results can reduce the calculation amount by using the low-rank decomposition method, and the effect of feature extraction is the same as that before decomposition. The test results are shown in Table 4.

| Model                  | mAP   |
|------------------------|-------|
| 1 residual module per scale | 78.6% |
| 2 residual module per scale | 88.5% |
| 3 residual module per scale | 82.4% |
| 4 residual module per scale | 72.9% |

From the data analysis results in the table above, it can be concluded that the four scales 104 * 104, 52 * 52, 26 * 26, and 13 * 13 are better for feature extraction by two optimized residual modules.

According to the test results of detection accuracy of different scales in Chapter 1.2, in the new network structure, we have removed the detection of two dimensions of 13 * 13 and 26 * 26, and increased the scale of 104 * 104 to achieve the purpose of more fine-grained detection. There are two main improvements in this step. The first is to add an inverted pyramid structure after the 13 * 13 scale residual module, and use the deconvolution layer to re-expand the feature map to 52 * 52 and 104 * 104, and then perform YOLO Detection. After the expansion of the feature map is completed, the shallow-scale features of the same scale are merged into the deep layer, and the shallow and deep feature information is synthesized. For shallow neural networks, because the feature map has not been
down-sampled many times, more target position information can be extracted. In deep networks, feature maps contain rich semantic information after multiple samplings. Before the YOLO layer is input, the features extracted from the shallow layer and the deep layer are fused to improve the richness of the feature information and thereby enhance the detection capability of the YOLO layer. We call the optimized YOLO v3 YOLO-E, and its network structure is shown in Figure 3.

Figure 3. YOLO-E network structure

3.2. Use of GIOU indicators

The accuracy of object detection in this application scenario for the prediction of the bounding box position is more accurate than that in ordinary scenes. This is because the detected target occupies a small pixel area in the picture, and even if the bounding box appears slightly offset, there will also be significant visual errors in the display on the prediction map. In order to enhance the sensitivity of the network to the location, we replaced the original IoU with a GIOU indicator and improved its loss function[5].

In object detection algorithms, IoU is widely used as an indicator of the accuracy of position prediction. Its advantages are mainly that the calculation method is simple to meet most calculation scenarios, but its disadvantage is also obvious, that is, it is not sensitive to position deviation. GIOU adds the calculation of the non-intersecting area on the basis of IoU, which is more sensitive to the offset between the prediction box and the actual bounding box. The calculation formula and schematic diagram are shown in Figure 4:

Figure 4. GIOU schematic

In the above formula, C represents the minimum rectangular area including the predicted bounding box A and the actual bounding box B,

$$|C\cap(A\cup B)|$$

is represents the area of C that does not include A and B. Finally, it is compared with the area of C, and then subtracted from IoU to obtain the final GIOU value. The value of IoU ranges from [0,1]. When IoU is 0, the gradient value is also 0 during the back propagation calculation. The model cannot be further optimized. The value range of GIOU is [-1,0) U (0,1], which avoids the problem of a gradient value of 0. At the same time, the introduction of C makes the non-overlapping regions get the attention of the network. We added GIOU loss calculation to the original YOLO v3 loss function calculation. The calculation formula of the loss function is as follows.

$$L_{GIOU} = 1 - GIOU$$
3.3. Fine search process design

In addition to requiring vehicles to be detected in aerial images, we also need to further narrow down the detection range based on search conditions. Analyze the characteristics of the samples on the aerial image, and filter the detected vehicle targets based on the model and color. The vehicles in the VEDAI dataset are divided into the following three categories: Cars, pickup trucks and SUVs. First, we cut out the vehicles in VEDAI through the coordinate information in the label file, and divided these vehicles into three types: cars, pickup trucks, and SUVs. Then, based on the trained YOLO-E network model, transfer learning is performed to complete the training of the classification network. The color selection of vehicles is relatively simple. When YOLO-E detects the vehicle, it extracts nine pixels centered on the center point of the predicted bounding box and sends them to the opencv color classifier for color determination. We set the upper and lower RGB thresholds for the six common colors of red, green, blue, white, gray, and black. The color of the vehicle can be determined when the color of the extracted pixel block belongs to a certain color interval.

4. Experimental testing and analysis

We make clear sample from VEDAI the data sets, picking out 2000 pictures which contain the vehicle 8:2, dividing the data set into training. The first is the comprehensive test of YOLO-E network with multiple indicators to improve the overall robustness of the network after a more intuitive response. We have added YOLO v3 to YOLO v3-tiny for comparison, and the comparison test results of the three network models are shown in Table 5:

|       | precision | recall | F1-score | mAP    | FPS |
|-------|-----------|--------|----------|--------|-----|
| YOLO v3 | 74%       | 67%    | 70%      | 76.5%  | 45  |
| YOLO v3-tiny | 65%      | 59%    | 61%      | 71.7%  | 69  |
| YOLO-E  | 88%       | 81%    | 84%      | 91.2%  | 42  |

The calculation formulas for the above five indicators are as follows.

\[
\text{precision} = \frac{TP}{TP + FP}
\]

\[
\text{recall} = \frac{TP}{TP + FN}
\]

\[
F1\text{-score} = \frac{2 \times \text{precision} \times \text{recall}}{\text{precision} + \text{recall}}
\]

Precision reflects the accuracy and recall of the model. F1-score indicator is a comprehensive indicator based on precision and recall, reflecting the robustness of the network, mAP is the average accuracy, which is also the detection accuracy of the vehicle in this experiment. We use the FPS frame rate indicator to reflect the model's inference speed, the values vary depending on the hardware equipment test. This test was performed on a NVIDIA 2080Ti graphics card.

From the table above, it can be directly seen that the overall performance of YOLO-E is better than YOLO v3 and YOLO v3-tiny. Although we have reduced the network layer, the network structure in the new residual module is more complex, which makes it less obvious for the optimization of inference speed.

In order to verify the improvement of the model of the GIoU indicator and the optimized loss function, we set up two different YOLO layers based on the YOLO-E network structure, one using the IoU indicator and the original loss function, and the other using the GIoU indicator and newly added \( L_{GioU} \) loss function, the comparison test results are shown in Table 6.
Table 6. IOU and GIOU test comparison

|               | precision | recall | F1-score | mAP   | FPS |
|---------------|-----------|--------|----------|-------|-----|
| YOLO-E(IoU)  | 87%       | 77%    | 81%      | 88.5% | 42  |
| YOLO-E(GIoU) | 88%       | 81%    | 84%      | 91.2% | 42  |

According to the data in the above table, it is proved that the optimization of GIoU and the loss function will improve the accuracy of the network.

The actual results of the three networks in the test set are shown in Figure 5.

From the test comparison chart above, you can also see the pros and cons of the three models. The main problem with YOLO v3-tiny and YOLO v3 is the feature extraction network. The simple YOLO v3-tiny backbone network leads to insufficient extraction of features. Some houses will be detected as vehicles by mistake. YOLO v3 has a deep backbone structure, because the detected targets are small, and the size of the feature map of the deep network is also small. It is easy to lose the small target information, causing the final detection result to be leaky check the situation. For the YOLO-E network, the network structure and the YOLO layer have been optimized accordingly, and the above problems of false detection and missing detection have not occurred.

At the same time, we test the accuracy of the network classification on the transfer learning. The test results are shown in Table 7:

Table 7. Classification accuracy test

|               | car       | pickup trucks | SUV    |
|---------------|-----------|---------------|--------|
| AP            | 98.7%     | 96.2%         | 95.9%  |
| mAP           |           |               | 96.8%  |

The classification accuracy of the three categories is as high as 93% or more. Pickup trucks and SUVs have certain similarities in the vehicle models, resulting in slightly lower accuracy than the cars, but the overall average accuracy also reaches 96.8%, which can be effectively classified. Then we tested the exact target retrieval, and set the retrieval targets to red cars and white pickup trucks respectively. The detection results are shown in Figure 6.

Figure 5. Comparison of several detection results

Figure 6. Accurate target detection results
5. Conclusion
This paper is based on the YOLO v3 algorithm to optimize the design of its feature extraction network and YOLO layer, which improves the performance of its backbone network for small target detection and the sensitivity of the YOLO layer to target locations. At the same time, based on the optimized YOLO-E network model, a small target vehicle classification network was migrated and trained. Later, the detected vehicle targets can be further detected according to the model and vehicle color, and reduce the target detected by YOLO-E network to the specified interval. Compared with ordinary vehicle detection, this experiment is aimed at vehicle detection of satellite images, which can detect multiple targets simultaneously in a large range. It has some practical value in real life. The optimized neural network has good robustness for the detection of such small targets. However, due to the richer structure of the improved residual module, the overall computing speed of the network has not been greatly improved. This is also the direction to continue to optimize the network. Compressing the network and improving the speed of network inference will enable it to be carried in the aerial cameras running on embedded devices, which will make the network more practical.

References
[1] Redmon J, Farhadi A. YOLOv3: An Incremental Improvement[J]. 2018.
[2] He K, Zhang X, Ren S, et al. Deep Residual Learning for Image Recognition[J]. 2015.
[3] Zhu,Yousong, zhao, Chaoyang, Wang, Jinqiao, et al. CoupleNet: Coupling Global Structure with Local Parts for Object Detection[J].
[4] Shafiee, Mohammad Javad, Chywl, Brendan, Li, Francis, et al. Fast YOLO: A Fast You Only Look Once System for Real-time Embedded Object Detection In Video[J].
[5] Rezatofighi H, Tsoi N, Gwak J Y, et al. Generalized Intersection over Union: A Metric and A Loss for Bounding Box Regression[J]. 2019.