ATLANTIS PRESS

Advances in Computer Science Research, volume 80

International Conference on Computer Science, Electronics and Communication Engineering (CSECE 2018)

The Application of Real-Time Object Detection on Aerial HD Videos Based on Deep CNN

Guanghui Xu, Shuai Xie*, Jun Wang and Guodong Wu

College of Communication Engineering, Army Engineering University of PLA, Nanjing, Jiangsu 210007, P.R. China

*Corresponding author

***Abstract*—Efficient real-time object detection in aerial HD videos is an urgent need, with the increasing use of UAV(unmanned aerial vehicles) in various fields. But it is still challenging to detect objects accurately and timely due to its large pixel size and relatively small objects. We use the popular SSD to detect people, cars, boats and so on from real-time HD videos which behave better than traditional methods. Moreover, we improve the deep learning algorithm by tailoring its networks and enlarging its size. We apply this algorithm to the real time detection of UAV aerial HD video and experiments show that our method can realize the organic combination of detection efficiency and detection effect.**

***Keywords*—real-time object detection; SSD; HD videos; UAV**

I. INTRODUCTION

High resolution cameras is being used in areas of video surveillance like security of public places, traffic monitoring, military and satellite imaging. This leads to a demand for computational algorithms for real time processing of high resolution videos. Recently, detection methods based on deep learning, such as Faster R-CNN, YOLO and SSD [1-4], have rapidly advanced. However, they require numerous samples to train the detection model and cannot be directly used to efficiently handle large-area remote sensing HD videos.

Simonyan et al. began to pay attention to the application of deep convolutional networks in high resolution remote sensing images [5]. Bin Pan et al. proposed a cascade convolutional neural network based on transfer-learning for aircraft detection on high-resolution remote sensing images [6]. Chongyuan Tao et al. proposed a pedestrian detection in a down-looking perspective using deep learning [7]. Nevertheless, all the algorithms mentioned can not accurately detect relatively small targets on high-resolution remote sensing images. To address this problem, we propose an algorithm that performs better in this respect by tailoring and enlarging SSD. Then we apply this algorithm to the real time detection of UAV aerial video.

This paper is organized as follows. Section II presents the idea, model architecture and training details about the improved networks. Section III gives experimental results from comparisons. Conclusions are drawn in Section IV.

II. APPROACH

Our approach is based on SSD which is a particular successful method for general object detection. And we tailor the last two feat layers and enlarge the model size to detect small objects and get higher detection efficiency, as introduced in the following.

A. Enlarged the Input Layer and Tailored the Last Two Feat Layers

The original SSD with a 300×300 input size is not suitable for aerial HD videos, because it is hard to detect relatively small object from a high altitude perspective when the HD videos which is resized to a size of 300×300 was inputted to original SSD. As shown in Figure *, the input size of SSD is enlarged from 300×300 to 1280×720, which leads to the decline of detection speeds accompanying with the increase of anchors. However, it is concerned a lot in real-time object detection, so that we tailored the last two feat layers whose receptive field is relatively larger to improve detection efficiency. As a reason of that the targets in aerial videos, such as people, cars and boats, are small, the accuracy of object detection is almost not affected.

B. Model Architecture

This is implemented using TensorFlow. The framework is shown in Figure 1. The SSD approach is based on a feed-forward convolutional network that produces a fixed-size collection of bounding boxes and scores for the presence of object class instances in those boxes, followed by a non-maximum suppression step to produce the final detections.

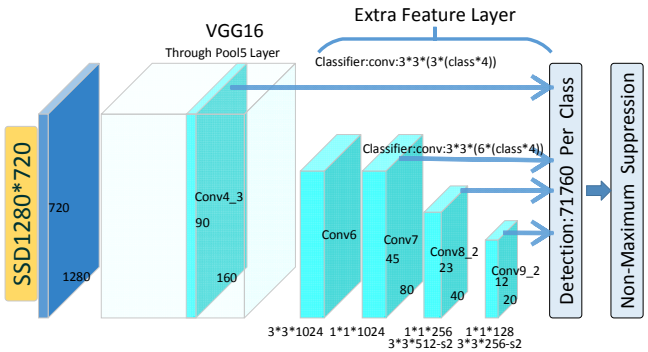

FIGURE I. OUR SINGLE SHOT DETECTION MODELS WITH A SIZE OF 1280×720

Figure I shows that the first layer is input layer with a size of 1280×720. The early network layers are based on a standard architecture, a part of VGG16 [3], used for high quality image classification (truncated before any classification layers), which consists of five max-pooling layers and thirteen convolutional

layers where we use filters with a very small receptive field: 3×3(which is the smallest size to capture the notion of left/right, up/down, center). As the fifth max-pooling layer uses kernels of size 3×3 with a stride of 1, its output size is 80×45. Then we add auxiliary structure to the network to produce detections. The layer named conv6 uses 1024 kernels of size 3×3, but the layer named conv7 uses 1024 kernels of size 1×1. The layer named conv8_1 uses 256 kernels of size 1×1. The layer named conv8_2 uses 512 kernels of size 3×3. A max-pooling layer is between conv8_1 and conv8_2, which is the same as conv9_1 and conv9_2. These layers decrease in size progressively and allow predictions of detections at multiple scales. Next a multiple boxes detection layer is followed, which is also a convolution layer where we use k filters with a size of 3×3.We choose conv4_3, conv7, conv8_2 and conv9_2 as features to be sent to multiple boxes layer. Finally non-maximum suppression layer eliminate the boxes whose IoU [4] values greater than 4.5.Equations

C. *Parameters Interoperability*

The model architecture except non-maximum suppression layer is a fully convolutional network. And all the parameters are convolution kernels with a size of 3×3, which is the same as original SSD. On the one hand, the number and size of the convolution kernels is not changed, although we enlarge the size of the input layers so that the size of the feature map [8] layers are correspondingly increased. On the other hand, the abstract levels of the convolution kernels is not changed. Our model with a size of 1280×720 can reuse the relative parameters of the original SSD when we adjustment the number and position of default boxes [3]. As a result, our model with a size of 1280×720 reuse the parameters which is trained beforehand on a model with a size of 300×300, instead of being trained directly on itself.

D. *Training Details*

Primarily, the popular datasets can not include all the categories, while we are usually required to discover specially designated objects in the target detection task. In order to test the practicability of the model better, we make part of our datasets through collecting 200 images of tank which we have chosen as a target category. Then we chose the three other categories, person, car and boat (646 images), which are part of the popular datasets, PASCAL VOC2012. We resized all the images to 300×300 before training. At last, we fine-tune the model which is based on pre-trained VGG16 network, with 10^{-3} learning rate for 60k iterations, then 10^{-4} for 20k iterations.

III. EXPERIMENT

A. *Difference in Speed of Different Model Sizes*

The difference in speed is obvious when we apply the trained parameters to different size models. We use a system Windows 10 with a CPU Inter Core i7-7700k(4.2GHz), memory 16 GB, and a graphics card NVIDIA GeForce GTX 1080Ti.

TABLE I. THE DETECTION SPEED SLOWS DOWN WITH THE INCREASE OF THE MODEL SIZES

Input Resolution	*Iterms*		
	Default boxes	***FPS***	***mAP***
300×300	8732	53	78.3
512×512	24564	29	82.2
1280×720	71760	11	76.6
1920×1080	193860	53	75.9

Table II shows that the detection speed slows down with the increase of the model sizes. Especially, the detection speed drops from 11FPS to less than 1FPS when size changes from 1280×720 to 1920×1080. It is more than theoretical expectations. In fact, the total variable of the model with a size of 1920×1080 outnumbers the memory of our GPU so that a part of the calculation is put on the CPU. However, the calculation speed of CPU is much slower than that of GPU, which conduct that the real-time performance of 1920×1080 model is far worse than that expected. Our application use the model with a size of 1280×720 which is a little bit visually discontinuous with a speed of 11FPS, but it is enough for searching.

B. *Application*

The application as shown in figure II intends to use unmanned aerial vehicles to patrol independently, in order to find specially designated objects. When the targets is detected, the drone sends out an alarm and then returns home automatically. Figure 2 shows the block diagram of the system. We use a UAV DJI Phantom4 Pro with a remote control, a HDMI-to-USB3.0 acquisition card, an android mobile phone and a notebook computer.

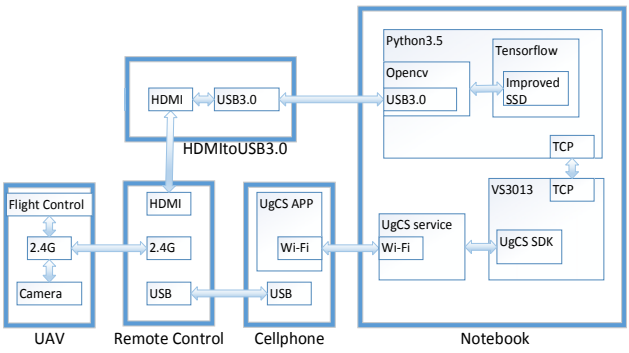

FIGURE II. SYSTEM BLOCK DIAGRAM OF THE APPLICATION

UgCS is a complete software to plan and fly drone survey missions, supporting most popular UAV platform, providing convenient tools for areal and linear surveys and enabling direct drone control. The preplanned path is introduced into the UAV through UgCS, and the UAV can completely patrol independently. At the same time, the real-time videos with a pixel size of 1280×720 captured by the UAV camera is transmitted to the remote control by the graphic transmission module. Through the HDMI interface on the remote control, the acquisition card converts the video into the UVC format and sends it into the notebook, which shows excellent real-time

performance. There are two main parts in the notebook, which are completed by Python and C#, respectively. The former completes the detection of video frames (not frame by frame but only real-time frames), and the latter uses UgCS SDK to control the UAV. The communication between the two depends on the socket. Once the target is found, it sends a voice alarm and controls the automatic return of the UAV. In the mission, the UAV is completely separated from human surveillance and control after taking off. The system delay is 0.24 seconds (Detection Processing + Transmission Delay). That is to say, the UAV finds the target and reacts accordingly when it flies to the target area for 0.24 seconds.

IV. CONCLUSION

Efficient real-time object detection in aerial HD videos is an urgent need, with the increasing use of UAV(Unmanned Aerial Vehicles) in various fields. But it is still challenging to detect objects accurately and timely due to its large pixel size and relatively small objects. Recently, detection methods based on deep learning have rapidly advanced. However, they require numerous samples to train the detection model and cannot be directly used to efficiently handle large-area remote sensing HD videos. We use the popular SSD to detect people, cars, boats and so on from real-time HD videos which behave better than traditional methods. Moreover, we improve the deep learning algorithm by tailoring its networks and enlarging its size. We apply this algorithm to the real time detection of UAV aerial HD video and experiments show that our method can realize the organic combination of detection efficiency and detection effect. The detection results of some video frames is shown in Figure III.

FIGURE III. EXAMPLES OF EXPERIMENTAL RESULTS

REFERENCES

[1] Girshick, R.: Fast R-CNN. In: ICCV,2015.

[2] Ren, S., He, K., Girshick, R., Sun, J.: Faster R-CNN: Towards real-time object detection with region proposal networks. In: NIPS,2015.

[3] Redmon, J., Divvala, S., Girshick, R., Farhadi, A.: You only look once: Unified, real-time object detection. In: CVPR,2016.

[4] Wei Liu. SSD: Single Shot MultiBox Detector[J]. ECCV16, 2016.

[5] Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. In: NIPS,2015.

[6] Bin Pan,Jianhao Tai,Qi Zheng,Shanshan Zhao,María Guijarro. Cascade Convolutional Neural Network Based on Transfer-Learning for Aircraft Detection on High-Resolution Remote Sensing Images[J]. Journal of Sensors,2017.

[7] Chongyuan Tao. Pedestrian Detection in A Down-Looking Perspective Using Deep Learning[A]. Proceeding of The 2nd Asia-Pacific Computational Intelligence and Information Technology Conference (APCIIT2017)[C],2017.

[8] Zhang, L., Lin, L., Liang, X., He, K.: Is faster r-cnn doing well for pedestrian detection. In:ECCV,2016.