Research on Simulation System of Small Unmanned Air-to-Ground Object Detection Platform

Xing Liu, Jian Chen, Dongfang Yang, Yongfei Li
Xi’an Research Institute of High Technology, Xi’an, China
Email: 549645043@qq.com

Abstract. Air-to-ground object detection has important application value in many fields such as military reconnaissance, air strike, personnel rescue, and natural exploration. Object detection algorithm based on deep learning solves the problems of poor generality and low robustness of traditional manual detection algorithms. It has great advantages in air-to-ground detection tasks, but for small unmanned platforms, currently deep learning object detection algorithms have high hardware requirements and are difficult to deploy. In this paper, a deep learning object detection simulation system is built based on the TX2 embedded development board with the features of low power consumption, light weight and fast calculation speed. This object detection simulation experiment is performed for two mainstream deep learning object detection algorithms. The experimental results showed that the regression-based SSD algorithm combined with MobileNets feature extraction network could perform real-time in the embedded system, which laid the foundation for the actual construction of the next unmanned aerial platform.

1. Introduction

Today is an era of big data and an era of images. The appearance of various intelligent terminals has resulted in a large amount of image and video data in human life. Object detection is the premise of a large number of advanced visual tasks, and it is the key to the research and application of the current academic and industrial circles. With the continuous development of machine learning theory and artificial intelligence technology, coupled with the promotion of big data and high-performance computing, the field of object detection has rapidly advanced in the past decade. A number of object detection algorithms based on deep learning have become mainstream in the field. Two-stage detection algorithms such as R-CNN[1], SPPnet[2], Fast R-CNN[3], Faster R-CNN[4], and Mask R-CNN[5] and one-stage algorithms based on regression learning such as YOLO[6], YOLOv2[7], SSD[8], DSSD[9] and so on, which make the object detection from the low detection accuracy to the higher detection accuracy while ensuring better real-time performance which provide conditions for miniaturized deployment in embedded systems.

Air-to-ground object detection is a branch of object detection. It has great application value in military affairs such as battlefield reconnaissance, enemy situation analysis, target tracking and civil use such as post-disaster search and rescue, disaster prevention and traffic grooming. At present, the object detection based on deep learning meets the requirement of real-time while pursuing high precision development to ensure accuracy, which provides preconditions for engineering applications. A simulation platform for autonomous UAV detection was built in this paper. We have carried out experiments on two types of mainstream detection algorithms. A lightweight detection algorithm based on regression learning was obtained to realize real-time detection.
2. Simulation platform based on TX2

2.1. TX2 Development Board
When it comes to artificial intelligence and machine learning (especially deep learning), the first impression left behind is the powerful hardware platform in the background. Nvidia launched the Jetson computing platform which used to deal with the challenges of the front end. Jetson TX2 is the latest computing platform in the Jetson series, which is an embedded computing platform based on the Pascal architecture and a common credit card size that combines high performance computing with low power consumption can be viewed as a small AI workstation. The TX2 incorporates components such as a 4-core ARM A57 CPU, a Pascal-based GPU (16-nm process), up to 8G memory, and 32G solid-state memory. The standard power consumption of this device is 7.5W, which can bring more performance improvements by increasing power consumption. When the power is increased to 15W, it will reach twice the computing power of TX1. TX2 also has the ability of automatic frequency control, can control the CPU core, GPU core and memory frequency, the voltage, frequency control at a more appropriate level. When dealing with some of the heavier tasks, such as object detection, the selection of dense targets in the scene can be automatically raised in voltage to achieve optimal performance. When dealing with simple tasks, such as drones cruising in the air, it can automatically reduce the frequency, reduce power consumption to achieve the purpose of continuation of Battery life.

2.2. Simulation System

![Figure 1. Air-to-ground object detection R&D simulation system](image)

The simulation system consists of three parts: a data acquisition terminal based on laboratory UAVs, a network development terminal based on desktop workstations, and a deployment application terminal based on the TX2 embedded development board (shown in Figure 1). First, we use UAV for air-to-ground object detection image acquisition and simulation video capture. Then we use the desktop workstation to perform data set production while training and verifying the designed network; Finally, the optimal network model is simulated in the TX2 development board, which using the collected simulation video to simulate the drone-equipped TX2 autonomous detection system for air-to-ground object detection. Desktop workstations and TX2 are based on the Ubuntu16.04 system, the programming language is python, and the Tensorflow deep learning framework is used to deploy and build the air-to-ground object autonomous detection platform. Tensorflow is Google's open source deep learning framework. The framework is easy to deploy and easy to develop. The visual function of tensorboard can visually display the network structure, train verification results, and facilitate research and comparison. At the same time tensorflow open source its object detection programming interface, Tensorflow object detection[10] programming interface includes mainstream object detection models such as Faster R-CNN, SSD, the entire programming interface set research, development, deployment in one, making the object simulation experiment more convenient.

3. Mainstream object detection algorithms

3.1. Faster RCNN
Faster RCNN is based on the Fast RCNN which introduces the RPN network. It discards the selective research strategy that originally used a lot of time and expenses. It improves the detection accuracy and
also improves the detection speed. In the network architecture, this network belongs to a multi-task model architecture. The entire network shares a convolutional network backbone (such as VGG-16) and performs RPN and Fast R-CNN processes respectively, which improves the generalization ability of the network. In the field of object detection belongs to a two-stage network. In the first level network (RPN), they propose to use a region of interests to locate significant information in a picture. These regions of interest can be understood as an attention model, adopting an anchor mechanism, and using the feature map mapping relationship. A large number of detection frames are generated with a fixed scale and ratio rate. The RPN network predicts a score for each anchor point. This score is used to measure the probability that an anchor point contains a target. At the same time, the RPN network also uses a regression mechanism to continuously modify the coordinates and scale of the bounding box as close as possible to the real groundtruth box. The revised anchor points will be sorted by the score, combined with non-maximal suppression, and the highest scoring anchor point will be reserved as the target area and sent to the second level network (Fast R-CNN). The optimization loss function is defined as follows:

$$L(p_i; t_i) = \frac{1}{N_c} \sum_i L_{cls}(p_i; t_i^*) + \lambda \frac{1}{N_{reg}} \sum_i p_i^* L_{reg}(t_i; t_i^*)$$

(1)

Where $i$ is the index value of the anchor and $p_i$ is the prediction probability that the $i$-th anchor is the object. When an anchor is positive, $p_i^* = 1$ (the negative is 0), $t_i = (t_x, t_y, t_w, t_h)$ which represents the parameter of a bounding box (the bounding box coordinates $x, y$ and the bounding box length and width $w, h$), $t_i^*$ represents the parameter of the groundtruth box. The classification loss function $L_{cls}$ uses the form of a logistic regression function to distinguish the target from the non-target.

$$p_i^* L_{reg}$$

indicates that the regression loss is only affected by anchors that are positive.

3.2. SSD

The SSD detection model is a one-step detection algorithm. Based on the YOLO algorithm, the anchor detection mechanism of Faster R-CNN is introduced to extract object feature information on different feature layers. The structure of the SSD is mainly composed of two parts. The front end uses VGG16 as the feature extraction network, and the back end adds an extra feature layer to obtain the prediction boxes on multiple features and multiple scales. A total of 8732 a priori boxes of different scale sizes are generated for each class of object. Then, the priori box is matched with the groundtruth box, and the a priori box with the largest IOU ratio and the IOU ratio exceeding the given threshold is also considered as Positive examples. These matched positive examples perform the regression calculation of the coordinates. This multi-scale and multi-feature detection process greatly improves the accuracy of the algorithm, and the real-time performance is also well guaranteed.

The multitask loss function of the SSD model is defined as follows:

$$L(x, p, l, g) = \frac{1}{N} (L_{conf}(x, p) + \alpha L_{loc}(x, l, g))$$

(2)

Where $x$ is the matching indicator function, $p$ is the class Category confidence, $l$ is the prediction box parameter, $g$ is the groundtruth box parameter, $N$ is the number of matching positive instances, $L_{conf}$ is the confidence loss function, and $L_{loc}$ is the location regression loss function.
4. Simulation experiment

4.1. Datasets
The air-to-ground object dataset requires professional shooting equipment. Therefore, the dataset is relatively scarce. We have collected data from the air-to-ground dataset through online gathering and shooting in laboratory equipment. Among them, the online datasets mainly come from Google Earth\(^1\) and the 3K Germany Munich vehicle dataset\(^2\). Google Earth has collected the streets and airports of many cities, and the elevation of perspectives is limited to within 500 meters (Figure 2). The laboratory used drone aerial photography to collect some street data sets in Xi’an.

Figure 2. Air-to-ground detection datasets

Rename the collected data set first, the naming format is: 000001.jpg, 000002.jpg, and so on. Then use labelImg\(^2\) software for labeling. The labeling box should be as close as possible to the target while not marking the targets with more than two-thirds of the obstructed area. A photo generated An xml file corresponding to the picture naming format (eg 000001.xml). This data set includes a total of two categories (cars and airplanes) and the total number is 2000, with 1,000 in each category.

The system uses the Pascal VOC format to store data sets. The Pascal VOC format divides the data set into three folders, the first namely the JPEGImages, which stores the original jpg format pictures; and the second, the annotation information folder (Annotations) which stores tagging information xml files generated by labelImg, and the last is ImageSets folder which uses two txt files to divide the data set into a training set and a validation set, respectively. Tensorflow object detection api itself does not support reading of Pascal VOC format, so you need to use the data format conversion module provided by the api interface to convert the to the tensorflow-recognizable tfrecords format, which generates the training and the verification tfrecords file.

4.2. Model R&D training

The training and verification phase of the object detection network was performed on Ubuntu 16.04, HP Z840 workstation (Nvidia GTX 1080, 8G). Training deployment requires preparation of training and validation datasets in tfrecord format, category tag files, model profiles, and pre-training models.

The feature extraction network of the Faster RCNN uses the inception\(^{12}\) proposed by the Google team. The basic idea of Inception is to use the 1x1, 3x3, 5x5 convolution kernel and the pooling layer at the same time to let the network self-learn how to choose four types of use. At the same time, using the bottleneck idea\(^{13}\), this CNN introduce a 1x1 point convolution for channel compression, and thus Decrease the amount of parameters and increase the speed of detection.

The SSD feature extraction network uses the MobileNets\(^{14}\) model proposed by the Google team. The model uses depth separable convolution to separate an original convolution with depth M into M/S

---

\(^{1}\) https://earth.google.com

\(^{2}\) https://github.com/tzutalin/labelImg
convolutions with depth S or M convolutions with depth 1 (figure 4(b)). Then, using the 1x1 dot convolution, the feature map obtained by convolving M depths of 1 is again extended along the depth direction to the feature map with the number of channels (Figure 4(c)).

The training process uses migration learning, and the weights of the training and powernets training under the VOC data set are used for initialization. A total of 200000 rounds of training are performed. The loss of the two types of target detection models during the training process is shown in the figure. It can be seen from the figure that the target detection model based on region extraction is better than the target detection model based on regression, Faster RCNN, and it is more adaptable to air-to-ground scenarios, and the loss decreases rapidly.

![Figure 5. loss of Faster RCNN and SSD](image)

The trained model is verified on the validation set. It is found that because of the unique appearance of the aircraft, which is easily discernible under the air-to-ground field of view, the overall accuracy of the aircraft is higher than that of cars. At the same time, detection accuracy based on the regional proposed detection model is higher than that based on the regression model.

![Figure 6. Visualization results](image)

| Model   | Feature Extraction Network | mAP(airplane) | mAP(car) | mAP   | FPS |
|---------|----------------------------|---------------|----------|-------|-----|
| Faster RCNN | Inceptionv1                      | 0.94          | 0.85     | 0.895 | 3   |
| SSD     | MobileNets                     | 0.89          | 0.70     | 0.795 | 23  |

### 4.3. Model application deployment
Figure 7 Faster RCNN-Inception (left) and SSD-MobileNets (right) real-time detection results

We use tensorflow object detection api to generate the pb file from the above files. Pb file records the entire network information, including the network structure and parameter weights, making the network model very convenient to deploy on other platforms. After the obtained pb file is copied to the TX2 air-to-ground simulation platform, the tensorflow object detection api is used again, and the video aerial data collected by the laboratory drone is simulated on the tx2 using the simulation platform. By observing the video detection results for the air-to-ground field of view, it is found that the TX2 platform is very efficient in processing deep learning object detection models, and the SSD detection model combined with MobileNets can perform real-time air-ground object detection.

5. Conclusion

Deep learning has developed rapidly in recent years, and has achieved great success in the field of detection. Air-to-ground object detection is an important branch of the object detection field, but detection algorithm based on deep learning has high requirements for computer hardware. With high power consumption and heavy quality, it is absolutely impossible to deploy small unmanned platforms. The Jetson TX2, a deep-learning embedded development module launched by NVIDIA in 2017, has achieved miniaturization, low power consumption, and light weight. A simulation platform for autonomous UAV detection was built in this paper. We have carried out experiments on two types of mainstream detection algorithms and a lightweight detection algorithm based on regression learning is obtained to realize real-time detection.

References

[1] Girshick R, Donahue J, Darrell T, et al. Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation[C]// IEEE Conference on Computer Vision and Pattern Recognition. IEEE Computer Society, 2014:580-587.
[2] He K, Zhang X, Ren S, et al. Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition[J]. IEEE Transactions on Pattern Analysis & Machine Intelligence, 2015, 37(9):1904-1916.
[3] Girshick R. Fast R-CNN[C]// IEEE International Conference on Computer Vision. IEEE Computer Society, 2015:1440-1448.
[4] Ren S, He K, Girshick R, et al. Faster R-CNN: towards real-time object detection with region proposal networks[C]// International Conference on Neural Information Processing Systems. MIT Press, 2015:91-99.
[5] He K, Gkioxari G, Dollár P, et al. Mask R-CNN[J]. 2017.
[6] Redmon J, Divvala S, Girshick R, et al. You Only Look Once: Unified, Real-Time Object Detection[J]. 2015:779-788.
[7] Redmon J, Farhadi A. YOLO9000: Better, Faster, Stronger[J]. 2016:6517-6525.
[8] Liu W, Anguelov D, Erhan D, et al. SSD: Single Shot MultiBox Detector[J]. 2015:21-37.
[9] Fu C Y, Liu W, Ranga A, et al. DSSD : Deconvolutional Single Shot Detector[J]. 2017.
[10] Huang J, Guadarrama S, Murphy K, et al. Speed/Accuracy Trade-Offs for Modern Convolutional Object Detectors[J]. 2016:3296-3297.
[11] Liu, K.; Mattyus, G. DLR 3k Munich Vehicle Aerial Image Dataset. Available online: http://pba-freesoftware.eco.dlr.de/3K_VehicleDetection_dataset.zip(accessed on 31 December 2015)
[12] Szegedy C, Liu W, Jia Y, et al. Going deeper with convolutions[C]// IEEE Conference on Computer
Vision and Pattern Recognition. IEEE Computer Society, 2015:1-9.
[13] Lin M, Chen Q, Yan S. Network In Network[J]. Computer Science, 2013.
[14] Howard A G, Zhu M, Chen B, et al. MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications[J]. 2017.