UAV Inspection Technology Based on Lightweight Edge Computing Framework

Zifeng Qiu¹, Yahan Wang¹, Feng Gao¹, Sha Liu², Junkai Zhang², Fuchen Yang¹ and Yan Hu*¹

¹ Key Laboratory of Aerospace Information Applications of CETC, Shijiazhuang, Hebei Province, 050081, China
² The 54th Research Institute of CETC, Shijiazhuang, Hebei Province, 050081, China

*Corresponding author’s e-mail: qzf93@foxmail.com

Abstract. At present, unmanned aerial vehicle (UAV) is usually used to capture the ground images for power inspections, then transmit the images to the fixed ground station for analysis. That method is not conducive to the rapid positioning of key parts or timely treatment. At the same time, the automatic sensing of power inspections mainly adopts the targets detection and classification algorithms based on deep learning, which has a large amount of calculation, and the processor installed in the UAV terminal is difficult to achieve the effect of real-time detection. In order to improve the accuracy and real-time of key targets detection and classification of power facilities in the inspection process, a lightweight edge computing framework AirNet is proposed. In AirNet, simple linear iterative clustering (SLIC) algorithm and direct convolution method are used to optimize the UAV input image to simplify multiple granularity feature information and improve the accuracy of the algorithm. Real time intelligent analysis of the algorithm model is carried out in UAV terminal, and the key parts of transmission towers, houses and vehicles are selected for test. The results show that the algorithm can achieve 64 ms detection speed and 85% accuracy on Huawei Atlas 200 chip equipment.

1. Introduction

Power line is an important part of power transmission, so it is necessary to inspect the power line regularly [1]. At present, there are cases of applying UAV remote sensing technology to power inspection [2], which can greatly improve the efficiency and safety of detection. In the field of traditional UAV power inspection, the intelligent algorithms are deployed in the ground computer. The ground receives the image information transmitted by the UAV for intelligent recognition, and then feeds back to the UAV through the link to realize intelligent control and perception [3]. This kind of application scenario seriously depends on the stability of the link. If the link is interfered or there is a bit error, there will be abnormal ground image parsing, deviation or failure of intelligent processing. Besides, the control instructions uploaded by the intelligent system will also leak, which will greatly reduce the effect of intelligent applications. Simultaneously, the load data needs to be transmitted continuously, so the link needs to reserve enough bandwidth for image data. The increase of bandwidth leads to the decline of anti-interference of the link and the shortening of communication distance, which also brings certain pressure to the communication link, and will face the problems of transmission delay.

In addition, the visual targets recognition and location based on UAV platform are different from the traditional monitoring system. The flexibility of UAV will lead to poor stability of images taken by the camera onboard, such as complex background, variable target scale, frequent occlusion and other...
problems, which bring difficulties to image intelligent perception. Concurrently, due to the images sequence obtained by aerial photography, the images resolution is larger. Therefore, in order to ensure the processing efficiency and analysis accuracy of these images, offline analysis is usually carried out on the asynchronous operation terminal with ultra-high data throughput, which reduces the real-time performance of analysis and limits the practical applications of UAV in targets detection to a certain extent.

Therefore, it is necessary to study the embedded detection algorithms based on edge computing [4]. As a new computing mode, edge computing decomposes the computing tasks and migrates them to the edge nodes for processing. Intelligent algorithms are deployed on the airborne terminal to reduce the computing load on the ground side and the transmission delay of the data link, so as to achieve the goals of reducing energy consumption, improving the real-time performance and reliability of the UAV, and enhancing the anti-interference of the data link.

In the intelligent computing scenario of the UAV edge computing for the ground network, it provides the computing unloading service for the user equipment on the ground by providing the edge computing terminal on the UAV. In the three-tier architecture of user equipment, UAV and ground base station, user equipment can unload the computing intensive tasks to the UAV, or unload some tasks to the server with stronger computing power in the ground base station through UAV relay when the user equipment is unable to unload the computing tasks directly to the server of ground base station due to blocking or other reasons. In the air-ground integrated mobile edge network proposed by [5], in addition to providing communication and caching services, UAV can also serve as an edge server to provide ground users with high bandwidth and low delay computing task unloading services. Ground users can unload computing intensive tasks such as virtual reality and image processing to the edge server of UAV for processing. In reference [6], a multi deployment mechanism based on differential evolution is proposed to realize the load balancing of UAVs. On this basis, a UAV task scheduling algorithm based on deep reinforcement learning is used to improve the efficiency of UAV task execution. In the study of power line inspection, a U-net-based method for detecting aerial photo insulators is presented [7], which can achieve automatic layering and feature extraction of different layers of information. This overlay method can effectively combine the shallow features with the high-dimensional features for joint training and learning. In order to avoid the problems of omission and false positives in hidden danger detection caused by occlusion or loss of some key information. Yang presents a detection algorithm for key components of transmission lines based on multiscale feature fusion [8], which uses deep-level convolution neural network for feature extraction and applies the algorithm to mobile ARM devices. However, the above literature does not take into account the limited computing power of embedded devices in edge computing model and the relatively small amount of training data, so the detection performance of these detection algorithms based on neural network needs to be improved. At present, the patrol algorithm carried on the UAV cannot meet the requirements of high quality, high efficiency and low delay detection in key parts of transmission lines.

For complex images in power patrol, a lightweight edge computing framework, AirNet, is proposed considering the UAV edge computing capability and data link transmission strength, and a simple linear iterative clustering (SLIC) algorithm is used to optimize the UAV input image. The algorithm model in AirNet is transplanted to Huawei Atlas 200 chip for real-time intelligent analysis in UAV terminal. The key targets such as wind turbines, transmission towers and illegal buildings commonly used in power inspection are selected for experiments. The results show that the algorithm achieves an average detection speed of 64 ms and an accuracy of 85% on the Atlas 200 chip device.

2. Approach

2.1. SLIC Super-pixel Segmentation

SLIC algorithm [9] divides an image based on pixel color and distance similarity. The algorithm has the advantages of high efficiency, uniform block size and regular outline, and is widely used in image segmentation such as unmanned aerial vehicle and remote sensing detection. The original SLIC super-
pixel segmentation algorithm steps are as follows: 1) dividing the mesh according to the image size and initializing the cluster center; 2) moving the cluster center to the minimum gradient position in the adjacent intervals; 3) setting the best matching pixel by the distance measurement formula in the adjacent spatial area near each cluster center; 4) calculating the new cluster center error to enforce connectivity after meeting the threshold. The distance measurements involved are as follows:

\[ J = J_{Lab} + m J_{xy} / S \] (1)

\[ J_{Lab} = \sqrt{(L_j - L_h)^2 + (a_j - a_h)^2 + (b_j - b_h)^2} \] (2)

\[ J_{xy} = \sqrt{(x_j - x_h)^2 - (y_j - y_h)^2} \] (3)

In the above formula, \( J \) is the distance between each pixel and the cluster center; \( J_{Lab} \) is the color feature distance; \((L_j, a_j, b_j)\) and \((L_h, a_h, b_h)\) are the Lab color space coordinates of \( j \) and \( h \), respectively; \( J_{xy} \) is the space feature distance; \((x_j, y_j)\) and \((x_h, y_h)\) are the two-dimensional space coordinates of \( j \) and \( h \), respectively; \( m \) is the influence factor of space distance; \( S \) is the distance between seed points.

2.2. Lightweight edge computing framework: AirNet

AirNet is a deep learning framework for UAV edge computing based on C language and CUDA language. Different from Caffe, TensorFlow, MXNet, Theano and other deep learning frameworks [10], AirNet does not have too many additional dependent libraries. When configuring the environment, it only needs to meet the basic conditions of C program and CUDA program. AirNet framework running environment is simple to build, and the program transplantation between different platforms is more convenient. For domestic platforms or embedded platforms with limited resources, the reason why most of the frameworks cannot run and build is often the lack of relevant library support, or the framework takes up too much resources. Here, the advantages of AirNet can be reflected.

The deep learning detection models developed based on AirNet run fast and support CPU/GPU computing devices. In addition, the applications from AirNet in image classification and object detection working fast, especially in the field of object detection, the proposed YOLO series networks can basically achieve real-time detection of targets on some platforms.

The main components of the AirNet framework are shown in Figure 1, which is also the core process of the framework when performing training tasks. It is mainly composed of four parts: network parameter initialization, training data loading, network layer calculation, and training result output.

![Figure 1. The main components of the AirNet framework.](image)

The core of network parameter initialization is network component. When the training data is loaded, a thread is started in load_data to call load_threads, which is the thread actually loads the data. The main
thread will wait until the load_threads is complete. After that, the load_data_in_thread function is called 64 times in the load_threads loop, and the data loading process is completed in the load_data_in_thread function. Then, start the thread in the load_data_in_thread function, call the load_thread function, and the main thread returns to load_threads to wait. Finally, the load_thread function calls the load_data_detection function to load the data. The functions involved in the network layer calculation and training result output mainly include train_network, train_network_datum, fill_truth_detection, etc. in the training process, the average loss is returned through the train_network function and the weights are updated through forward and backward circular calculation.

2.3. UAV edge computing paradigm based on SLIC optimization

Aiming at the problem of low computing speed of UAV edge, an intelligent sensing algorithm based on SLIC super-pixel segmentation optimization of AirNet is proposed to solve the problems of target rotation and scale transformation. The specific steps are as follows:

1) Firstly, an appearance model based on SLIC super-pixel segmentation optimization is established by using the front images of UAV;

2) The input images of each frame are segmented by SLIC, and the confidence maps of the segmented image is obtained by comparing with the apparent model;

3) The above confidence maps are fed into the AirNet edge computing model trained in advance, and the perception result of the model is modified by the SLIC super-pixel segmentation result. The specific method is as follows: the classification category of all pixels belonging to the same super-pixel block with the most frequent occurrences is assigned to the classification category of all pixels of the super-pixel block and marked as Result-SLIC;

4) According to the information entropy, the classification results with high confidence are selected, and the classification results of the corresponding region in Result-SLIC is modified. The final classification result is marked as Result-entropy, the calculation method of information entropy is as follows:

\[
    h(x) = -\sum_{i=1}^{N} p(i)\ln(p(i))
\]

In the above formula, \(N\) stands for the number of categories; \(i\) stands for the \(i\)-th category; \(p(i)\) stands for the probability of dividing \(x\) into the \(i\)-th category; the smaller \(h(x)\), the higher the confidence of the classification result of \(x\), and the position where the maximum confidence is obtained is the position of the perceptual target.

In the process of edge computing, the direct convolution method is used to convolute multiple groups of convolutions on the same multi-channel feature map, and an output feature map with the same number of channels and convolution groups is obtained. In order to facilitate the task partition of AirNet serial code in GPU, the direct convolution method is used for optimization, as shown in Figure 2.

Figure 2. Direct convolution method and SLIC in AirNet.
In Figure 2, \( n_1 - n_i \) represents the input characteristic diagrams of the convolution layer, with \( i \) channels in total. \( K_1 - K_j \) represents all convolution kernels of this layer, and there are \( j \) groups in total, representing \( J \) channels of output. Each group of convolution kernel has \( i \) convolution kernels, which is the same as the input channel. The output characteristic graphs of convolution layer are expressed as \( m_1 - m_j \), and there are \( j \) channels in total. For the output characteristic graph \( m_a \) of one of the channels, it is calculated by the input characteristic graph and the convolution kernel \( K_a \) of group \( a \): the results of convolution operation of \( K_{a1} - K_{ai} \) at \( n_1 - n_i \), and \( m_a \) is obtained after accumulation. According to the convolution calculation, the number of thread blocks is set to \( j \), which is equal to the number of channels. In this way, each thread block completes the calculation of an output characteristic graph. In the thread block, the number of threads is divided according to the size of the feature graph. A thread is responsible for the calculation of one pixel of the output feature graph. In this way, the task partition is equal, the computing power of each thread can be fully and equally utilized, and the time cost for synchronization of parallel programs is reduced.

3. Experiment

3.1. Environment and data

In this paper, the proposed AirNet edge computing framework is tested. The operating system is Ubuntu16.04, using the AirNet adapted Yolo series neural networks. Hardware conditions are Xeon E5 2660 processor, 128G memory, and Tesla P100 graphics card. The core configuration of mobile device with edge computing capability is Atlas 200 develop kit, Hi3559 module and Atlas 200 AI acceleration module. The Ubuntu server is connected with Atlas DK developer board through USB interface. The composition of edge computing module is shown in Figure 3.

![Figure 3. The composition of edge computing module.](image)

The data set used in this paper comes from the pictures taken by the UAV during the inspection of the State Grid, covering many regions, different time periods and various scenes. The camera is equipped with a Zenmuse Z30 pan-tilt camera and a 1/2.8 in. progressive CMOS sensor with an effective pixel of 2.13 million. There are 79273 images in training set and 8033 in test set.
3.2. Comparison of results

Yolo-v4 and tiny-yolo-v4 trained by AirNet are used to count the accuracy and speed of target perception, and compared with [7] and [8]. The results are shown in Table 1.

Table 1. A slightly more complex table with a narrow caption.

| Methods          | Transmission tower | Vehicle | Building | Average Value |
|------------------|--------------------|---------|----------|---------------|
|                  | Accuracy | Speed(ms) | Accuracy | Speed(ms) | Accuracy | Speed(ms) | Accuracy | Speed(ms) |
| AirNet (Yolo-V4) | 0.93     | 0.141    | 0.89     | 0.138     | 0.95     | 0.143     | 0.92     | 0.141     |
| AirNet (Tiny-Yolo-V4) | 0.86    | 0.061    | 0.81     | 0.058     | 0.89     | 0.072     | 0.85     | 0.064     |
| Ref. [7]         | 0.75     | 0.318    | 0.69     | 0.301     | 0.71     | 0.241     | 0.72     | 0.287     |
| Ref. [8]         | 0.78     | 0.276    | 0.71     | 0.298     | 0.73     | 0.253     | 0.74     | 0.276     |

Comparison results with the four typical existing methods, AirNet (Yolo-V4), AirNet (Tiny-Yolo-V4), U-net [7] and multiscale feature fusion [8] for detecting the three types targets are listed in Table 1. It should be noted that these four algorithms are trained in the same experimental environment and use the identical test sets. From it we can see, the detection accuracy of the AirNet (Yolo-V4) is validated with the highest 0.92 compared with the AirNet (Tiny-Yolo-V4) 0.85, U-net 0.72, and multiscale feature fusion 0.74. But in terms of detection speed, AirNet (Tiny-Yolo-v4) is the fastest, its accuracy is slightly low, but also in the acceptable range.

The evolution metrics in Table 1 are explained as follows. Accuracy rate represents the proportion of positive and negative samples detected correctly to all samples. Speed refers to the time taken to input an image and output the perception result. It can be concluded the proposed AirNet possessed the following advantages over the traditional edge computing algorithms.

1. Compared with the general neural networks in machine learning, the AirNet can accurately classify and identify various targets with a desired time consumption and robustness.

2. In contrast to the multiscale deep learning algorithm, the AirNet with SLIC and direct convolution me achieves highly detection accuracy upon UAV images. It can increase the detection speed by two orders of magnitude.

4. Conclusion

This paper proposes a UAV inspection approach based on AirNet, a lightweight edge computing framework, which processes multi granularity feature information based on SLIC super pixel segmentation. The UAV is equipped with Huawei Atlas 200 artificial intelligence computing acceleration module, which has very powerful computing power and provides a better computing platform for detection algorithms. After the back-end server pre-trained the model, it was migrated to the UAV for the calculation of the edge detection model, and the accuracy and speed of the perception of the key components of power inspection were improved. Three key parts, such as transmission tower, vehicle and building, are selected to test. The results show that the approach can achieve 64 ms detection speed and 85% accuracy. This paper solves the problems of slow speed and low precision of traditional UAV inspection. The algorithm adapts to various backgrounds and some complex environmental changes, which may be applied to aerial photography of UAV and ground mobile robot shooting in actual electric power patrol.

Acknowledgments

This work was supported by the National Key Research & Development Program of China (2017YFB0503003).
References

[1] Zhang T, Dai J. (2021) Electric Power Intelligent Inspection Robot: a Review. Journal of Physics: Conference Series, 1750(1): 012-023.

[2] Zifeng Qiu, Shuangxin Wang, Zhaoxi Zeng, Dingli Yu. (2019) Automatic visual defects inspection of wind turbine blades via YOLO-based small object detection approach. Journal of Electronic Imaging, 043023:1-11.

[3] Feng Y, Song B, Liu F, Jiang Q. (2019) Performance Analysis of UAV Uplink Communication Technology. Journal of Physics: Conference Series,1237(3): 1-7.

[4] Mach P, Becvar Z. (2017) Mobile edge computing: a survey on architecture and computation offloading. IEEE Communications Surveys & Tutorials, 19(3): 1628-1656.

[5] Cheng N, Xu W, Shi W. (2018) Air-ground integrated mobile edge networks: architecture, challenges, and opportunities. IEEE Communications Magazine, 56(8): 26-32.

[6] Yang L, Yao H, Wang J. (2020) Multi-UAV enabled load-balance mobile edge computing for IoT networks. IEEE Internet of Things Journal, 99: 1-7.

[7] Chen JW, Zhou X, Zhang R, Zhang D. (2018) Aerial insulator detection based on U-net network. Journal of Shaanxi University of Science & Technology, 36(4): 153-157.

[8] Yang G, Sun C, Zhang N, Jin T, Xu C, et al. (2020) Detection of key components of transmission lines based on multi-scale feature fusion. Electrical Measurement & Instrumentation. 57(3): 54-59.

[9] Cong J, Wei B, Yin Y. (2014) Performance evaluation of simple linear iterative clustering algorithm on medical image processing. Bio-medical Materials and Engineering, 24(6): 3231-3238.

[10] J. Schmidhuber. (2015) Deep Learning in Neural Networks: An Overview. Neural Networks, 61: 85–117.