Journal of Information Hiding and Multimedia Signal Processing ©2020 ISSN 2073-4212
Ubiquitous International Volume **11**, Number **4**, December **2020**

Target Detection and Recognition for Wide Area Border Defense Intelligent Surveillance

Hao Luo

School of Aeronautics and Astronautics
Zhejiang University
38 ZheDa Road, Hangzhou, 310027, China
luohao@zju.edu.cn

Yundong Guo

School of Mechanical Engineering
Zhejiang University
38 ZheDa Road, Hangzhou, 310027, China
yundong88@zju.edu.cn

Tien-Szu Pan

Department of Electronic Engineering
National Kaohsiung University of Science and Technology
415 Chien-Kung Road, Kaohsiung, 807, Taiwan
tpan@nkust.edu.tw

Received November 2020; revised December 2020

Abstract. *A lot of exploration and practice on the construction of border video surveillance under the conditions of informatization have been investigated. However, there are still some problems in the video surveillance system, such as insufficient application depth, insufficient information analysis and processing capabilities, and low level of intelligent application. With the rapid development of deep learning and computer vision technology, intelligent video surveillance systems are widely used in various fields such as transportation and security. How to apply intelligent video surveillance to border defense systems to meet the requirements of practical, effective, reliable, advanced and economical. In response to the above needs, based on the three-camera array video surveillance image, this paper sets five types of monitoring targets for people, car, airplane, rotorcraft and birds, and studies the application of MobileNet-SSD network for target detection and recognition under the Caffe framework. The paper firstly choose the Caffe framework, MobileNet-SSD network and OpenCV to build a working environment. Secondly, build a specific dataset based on five types of images with established targets and VOC data sets. Finally, image and video target detection tests are carried out on the model generated by training.*
Keywords: Target detection, Target recognition, Deep learning, MobileNet-SSD algorithm

1. **Introduction.** The border is a frontier position to resist foreign invasions, a channel for communication with neighboring countries, and a window to show the might of the country. Due to the special geographical location of the border, border defense plays a pivotal role in the national security strategy and is an important part of national defense. It is an important barrier to national security.

With the rapid development of video surveillance technology, its application has penetrated into all aspects of daily work and life. With the development of the Internet of Things, the research and development space of video surveillance has also been greatly adapted and expanded. In the wave of informatization since the end of the twentieth century, many countries have studied and explored a large number of border defense control methods in response to the effectiveness, real-time and shared needs of border control. Some fruitful exploration and construction have been developed in the real-time collection and efficient transmission of border video surveillance information, but the video surveillance system application level is still relatively low, including insufficient application depth, insufficient integration, insufficient information analysis and processing capabilities, low level of intelligent application, imperfect management mechanism, etc.

With the continuous development of network technology, data storage technology and computer vision technology, with the Internet as a platform, digital video surveillance systems with intelligent image analysis functions based on embedded technology are being widely used in various fields. At present, researchers have built a monitoring system in some difficult areas and observing blind spots. The system has set up cameras and pan-tilts, data transmission systems, solar power supply systems, etc. on unattended watch towers to achieve all-weather visibility in the jurisdiction monitor. Judging from the current application situation, how to not only meet the specific video surveillance requirements in the border defense wide-area scenario, but also take into account the requirements of practical, useful, reliable, advanced and economical, is the research of border defense intelligent surveillance important question.

Therefore, this article focuses on the high cost of the monitoring system pan/tilt. The conventional pan/tilt has a small load, is affected by the wind when it is rotated, and its own load limitation can easily cause the camera to swing or rotate instability with the wind, and it is difficult to ensure the high-precision positioning and high quality of the system camera Based on the three-camera array video surveillance image, a solution for target detection and recognition using the MobileNet-SSD network under the Caffe framework is proposed. With the help of neural network and computer vision technology, the effective information in the video image is fully extracted in order to achieve Early warning monitoring for key targets of border defense, solving the high cost of monitoring pan-tilt, and conventional servo mechanisms that are difficult to meet high-precision positioning requirements, realizing in-depth, intelligent, and high-quality applications of border defense video surveillance, providing reliable and efficient real-time information Reduce the workload of manual monitoring and assist border guards to ensure border security.

The rest part of this paper is organized as follows. Section 2 gives a brief review of the related work. Section 3 and 4 extensively describes the proposed technique. Section 5 demonstrates the experimental results and gives some discussions. Finally, conclusions are drawn and the associated future work is given in Section 6.

2. **Related Work.**

The existing target detection and recognition technologies [1, 2, 3, 4, 5, 6, 7, 8, 10, 11, 12, 13, 14, 22, 23, 24, 25, 30] are mainly divided into two types:

(1) The traditional method based on manual feature annotation is mainly divided into six steps: preprocessing, window sliding, feature extraction, feature selection, feature classification and post-processing. The research focus is on feature extraction and feature classification. The selection of features is the most important part of traditional algorithms. Features include color features, texture features, shape features, etc. Generally, the target and the background have a large color difference. You can choose a suitable

color space (RGB, HIS, Lab, etc.) The background and the target are separated to complete the target detection. Traditional target detection methods using designed features have the following shortcomings: a) The designed features are low-level features, and the ability to express the target is insufficient; b) The designed features are poorly separable, resulting in a higher classification error rate; c) Design The features of are pertinent, and it is difficult to choose a single feature for multi-target detection.

(2) Method based on deep learning technology. In order to extract features better, Hinton first proposed the concept of deep learning in 2006, using deep neural networks to iteratively learn from a large amount of data to high-level features. Compared with traditional methods, the acquired image features are richer and more expressive. Convolutional neural networks are mainly used to process images. On top of the most basic neural network structure, a convolutional layer (sufficient extraction of features) and a pooling layer (reduce the amount of data processed by the convolutional layer) are added. The accuracy of target detection can be greatly improved; the convolutional neural network not only extracts high-level features, but also improves the high-level semantic expression capabilities of features. It also integrates feature extraction, feature selection, and feature classification into the same model to enhance the separability of features.

In practical applications, target detection algorithms based on deep learning can be divided into two types according to model training methods [15].

(1) Two-stage target detection algorithm based on candidate regions: The detection process is divided into two steps. First, the candidate regions (RegionProposals) are generated and the features are extracted, and then the feature maps (Feature Map) generated by all the regions to be selected are passed through a series of The classifier, classification and linear regression, refined positioning box (Bounding Boxes); specific algorithms are mainly: Girshick et al. proposed the first algorithm R-CNN (Regions with Convolutional Neural Network) that can truly achieve industrial applications) [16]; In 2015, He Kaiming et al. proposed SPP-net (Spatial Pyramid Pooling Networks) based on R-CNN and for the repetitive operations and shape distortion of convolutional neural networks [17]; Ross Fast R-CNN [18] proposed by Girshick et al. in 2015; Shaoqin Ren et al. on the basis of Fast R-CNN solve the problem of SPP-Net and Fast R-CNN, both of which have separate candidate regions CNN [19]; Region-based full convolutional network (R-FCN), feature pyramid network (FPN), mask R-CNN, etc.

(2) A single-stage target detection algorithm based on classification/regression. This type of algorithm directly extracts features from the original image to predict the classification and position of the object, and converts the target frame positioning problem into a regression problem. The main methods are MultiBox, AttentionNet, and G- CNN, Joseph Redmon et al. in 2015 inherited OverFeat's algorithm YOLO [20] Wei Liu et al. proposed a combination of YOLO regression ideas and Faster R-CNN's anchor box mechanism for the problem of poor positioning accuracy of the YOLO algorithm. SSD (Single shot multi box detector (Single shot multi box detector) [21], YOLOv2, Deconvolution Single Point Detector (DSSD), Deep Supervised Object Detector (DSOD), etc.

The two-stage detection algorithm can accurately detect dense targets and small targets, but the detection speed is slow. The single-stage algorithm omits the process of candidate regions, has a simple structure, and improves the detection speed. However, for scenes where the target density is too large or the target overlap is high, the missed detection of small and dense targets occurs, which is reduced to a certain extent Improve the detection accuracy.

3. **Proposed Technique.** According to the five types of detection targets set for fixed-wing aircraft, birds, vehicles, people, and rotorcraft, the Mobilenet-SSD network is used

for target detection and recognition under the Caffe framework, and the OpenCV computer vision library is used for image processing

3.1. **MobileNet-SSD algorithm.** The SSD proposed by Wei Liu et al. is a single-stage target detection algorithm. As an improved model of the current mainstream network, SSD is derived from the Faster R-CNN series of detection models. It not only inherits the good performance of the original detection framework, but also improves and updates on the basis of the original model. It has a good balance between timeliness and accuracy and is widely used . The network structure of SSD is to change the two fully connected layers in the VGG16 network into convolutional layers, remove the soft-max layer and add 4 convolutional layers, and obtain feature maps of different scales from different convolutional layers of the convolutional neural network. , And obtain the high-level semantic information of the image, and then perform non-maximum suppression to complete the prediction, thereby improving the accuracy of the target detection result.

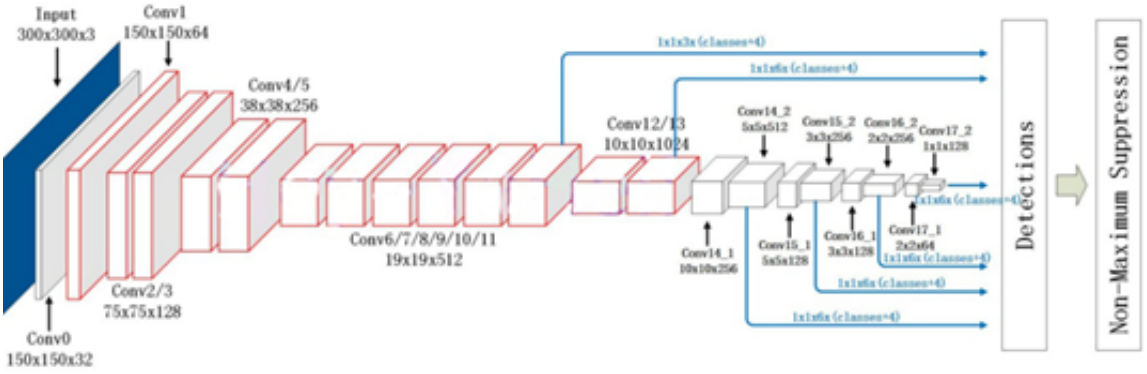

FIGURE 1. Structure of MobileNet-SSD

MobileNets [26] proposed by Andrew G. Howard et al. is a lightweight deep neural network based on a streamlined architecture and using deep separable convolutions proposed by Google for embedded devices such as mobile phones for mobile and embedded vision applications. Depth separable convolution is to solve the standard convolution into deep convolution and 1x1 point-by-point convolution. Through the convolution solution, a large number of parameters are saved, the amount of calculation is greatly reduced, the calculation speed is increased, and the accuracy of the model is less affected. The MobileNets structure is based on the above-mentioned deep decomposable convolution. Only the first layer is standard convolution. Except for the last fully connected layer, all layers are followed by batchnorm and ReLU, which are finally input to softmax for classification. Considering that deep convolution and point convolution are different layers, MobileNets has a total of 28 layers. In addition to v1, two improved version of MobileNet has been reported in [27, 28].

Since the SSD model uses a single-stage target detection algorithm, no candidate regions are required, and feature maps of different scales are directly obtained from different convolutional layers of the convolutional neural network, while the MobileNets lightweight deep neural network uses deep separable volumes The specific network structure established by product, which allows the two to be combined. As shown in Figure 1, the configuration of MobileNet-SSD from Conv0 to Conv13 is completely consistent with the MobileNet v1 model. After the final convolution layer Conv13, 8 convolution layers are added, and a total of 6 layers of feature maps are extracted for target detection. The Mobilenet-SSD network model uses the SSD model as the basic model, combined with

the characteristics of Mobilenet using fewer parameters and reducing the amount of calculation, using a small-scale parameter network on the basis of ensuring good accuracy, reducing the amount of calculation and reducing resource consumption.

3.2. **Caffe framework.** With the development of deep learning, researchers have developed many open source deep learning frameworks, such as Caffe, Tensorflow, MXNet, Theano, etc. Caffe [29] is a deep learning framework developed and released by the University of Berkeley Vision and Learning Center in 2014. The framework is written in clean and efficient C++ code, with layers as the unit, which makes the framework structure clear, and the code execution efficiency is high. Loss of flexibility; CUDA is used for GPU computing, and has almost completed the good support binding with Python/Numpy and Matlab, supports command line, Python and MATLAB interfaces, and can directly and seamlessly switch between CPU and GPU.

The Caffe framework is very efficient when processing large amounts of data. In addition to using OpenBLAS, MKL, cuBLAS and other computing libraries, it is also compatible with GPU acceleration, which makes it very suitable for feature extraction of two-dimensional image data; in addition, Caffe provides training and prediction A complete set of tools, such as, fine-tuning, publishing, data preprocessing and automatic detection, greatly reduces the difficulty of deep learning research and development, and is very user-friendly. There are three core modules of Caffe, namely Blobs, Layers, and Nets. Among them, Blobs is a four-dimensional continuous array that represents the amount of input data, the number of channels, and the length and width of the image. It can represent input and output data, parameter data; Layers are some The representation of the network layer includes convolutional layer, pooling layer, activation layer, etc.; Nets is a collection of Layers, which associates Layers layers together.

4. Model Training and Testing Based on Self-collected Data Sets.

4.1. **Self-collected data set based on VOC2007.** To detect the five types of detection targets set for people, vehicles, fixed-wing aircraft, rotorcraft, and birds, you need to create your own data set and add it to MobileNet-SSD for training. The five types of targets that need to be detected are not completely consistent with the existing data set and cannot be used directly; this article makes its own data set according to the format of the VOC data set [9] to connect to practical applications. The specific production method is as follows:

(1) Create a dataset folder: Create a self-made dataset folder VOC devkit and subfolders Annotations, ImageSets/Main and JPEG Images. The JPEG Images folder stores the five types of established target images prepared by yourself, and the image formats must be consistent; the Annotations folder is used to store the xml files generated by tagging the images in JPEG Images, and the two must be one-to-one correspondence; the ImageSets folder contains Main The subfolder stores txt files. The file is the name of the picture used for training verification and testing, excluding the suffix.

(2) Annotate the target picture: Since the names of pictures downloaded or crawled on the Internet are mostly irregular and inconvenient to use, rename the picture first. For example, researcher can refer to "000001.jpg" for naming for easy use. Commonly used tools for labeling pictures include: labelme and IAT for image segmentation tasks, labelImg, and Vatic (which can be used for video labeling) for image detection tasks; this article uses labelImg for image labeling, and after labeling the target frame Generate an xml file, as shown in Figure 2, the file contains information such as the source image storage path, image size, tag name, and destination location. This article first obtained 1344 images of five types of targets from the Internet for the first training; later, 327

additional images from the Internet, extracted from the VOC2007 data set include the original images and corresponding xml of four types of targets including people, vehicles, birds, and fixed-wing aircraft 2856 files are used for the second training. The extracted source images and corresponding xml files should only include one or more of the above four types of targets, so as to avoid mixing with other categories to affect training. See Figure 3 for pictures of self-made dataset and target statistics.

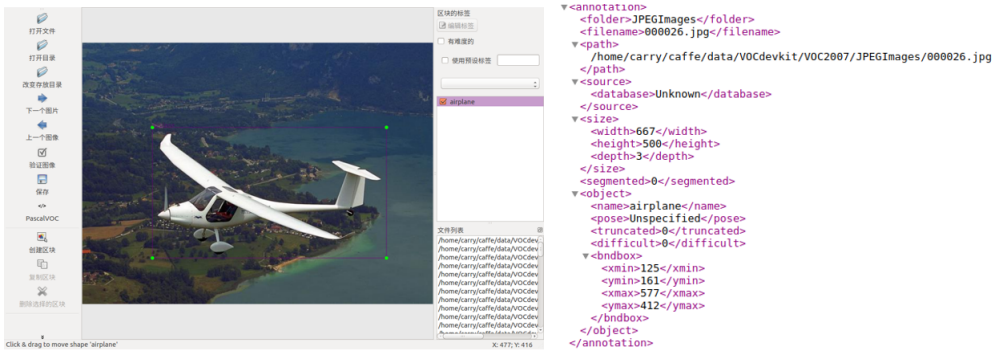

FIGURE 2. Image annotation based on LabelImg (left) and the xml file (right)

(1) Generate image set txt file: After obtaining the xml file of the source image, you need to divide the training verification set (trainval) and the test set (test) according to the ratio. The training verification set includes the training set (train) and the verification set (val). And write the corresponding xml file names into four txt files: trainval, train, val, test; due to the large number of pictures.

(2) Make labelmap.prototxt: This file is used to define the types of training samples, which can be copied from the caffe root directory /data/VOC0712/ to the self-made data set folder, and modified according to the established goals. The training sample categories are background and established target categories. There are six categories in this article.

(3) Create lmdb database: Caffe supports lmdb, h5py, etc. for training data formats, and uses lmdb format data by default. The lmdb data format is often used for single-label data, such as classification; for regression and other problems or multi-label data, the h5py data format is generally used. Refer to the SSD project under the caffe framework of weiliu89, copy the two scripts create_list.sh and create_data.sh in the Caffe root directory /data/VOC0712 to the self-made data set folder, and modify the path in the script. After running the above two scripts in sequence, two folders, VOC2007_test_lmdb and VOC2007_train_lmdb, can be generated under the self-made data set folder, and the generated database files are built-in. This completes the creation of the self-made data set.

4.2. **Model training and testing.** The main files in MobileNet-SSD and their uses are shown in Figure 4. In the training model link, it is worth noting that the SSD under the Caffe framework of weiliu89 uses the python script ssd_pascal.py to automatically generate the prototxt file and starts training, while the MobileNet-SSD of chuanqi305 uses the gen_model.sh script to generate the prototxt file and use it The train.sh script starts training. The process is as follows.

(1) Regenerate network files: Since the VOC data set is 21 types (plus background), it is necessary to regenerate training, testing and running network files. Run the gen_model.sh script on the command line. Its usage is: bashgen_model.sh num, where num is the number of sample categories, which is consistent with labelmap.prototxt.

Images / Category	Train		Validation		Train/ Validation		Test	
	Image	Target	Image	Target	Image	Target	Image	Target
Airplane	280	345	102	122	382	467	165	198
Bird	315	462	148	279	463	741	174	322
Car	454	821	207	359	661	1180	274	463
Person	450	1115	187	527	637	1642	287	721
Rotorcraft	128	146	54	63	182	209	97	118
Total	1519	2889	651	1350	2170	4239	930	1822

FIGURE 3. Collected Dataset

File	Purpose
MobileNetSSD_deploy.prototxt	Run the network definition file
solver_train.prototxt	Network training hyperparameter definition file
solver_test.prototxt	Network test hyperparameter definition file
train.sh	Network training script
test.sh	Network test script
gen_model.sh	Generate custom network script
gen.py	Generate common template script
demo.py	Actual detection script
merge_bn.py	Combine bn layer scripts to generate the final caffemodel

FIGURE 4. Files in MobileNet-SSD

(2) Modify the network files: After running, three network files deploy.prototxt will be generated in the MobileNet-SSD/example directory. You need to change the source of the data_parm layer in the network file to the actual path of the self-made data set, in the file The batch_size is not easy to set too large, otherwise it will cause memory overflow and training failure.

(3) Training model: Run the train.sh script to start training. Before that, change the solver_train.prototxt network parameter file according to your needs. The model generated by training will be saved to the snapshot folder according to the definition of the parameter file. After the training, the training situation will be output on the screen. Also run merge_bn.py to merge the bn layer to generate the final model. The first model (hereinafter referred to as model 1) training iterations is 120,000 times, and the average network accuracy is 0.877; the second model (hereinafter referred to as model 2) training iterations is 100,000 times, and the average network accuracy is 0.867151.

There are two types of model testing. One is to use the test set in the data set to detect the model effect, and the other is to use the data set in the actual application outside the data set to detect the model effect; the model test here refers to the former, the latter will Do it in the next chapter.

Run the test.sh script under MobileNet-SSD to start the test. The solver_test.prototxt file is called during the test; the detection_eval value and loss value are printed on the screen after the test is over. The average accuracy of model one is 0.877272, and the average accuracy of model two is 0.866873.

5. **Target Detection and Recognition Based on Self-training Model.** In order to further test the detection effect of the self-training model, and study the recognition of various targets by the self-training model in detail, this paper randomly obtains a total of 580 pictures from the network outside the self-made data set, and makes the model one run

on this picture set. This run correctly recognized 492 pictures, the detection frame rate is basically 9-11 frames, and the total correct rate is 85.86%. The problems that occur during model 1 operation are divided into the following five categories: unrecognized target, missed target, wrong target, misrecognized other objects, correct target and misrecognized target (generate multiple bounding boxes for the same target).

It can be seen from the running results of model 1: (1) The vehicle-type target recognition effect is the best. There are two reasons: one is that the vehicle shape is relatively single; the second is that the number of training images is slightly more than the other four types of targets. (2) The human target recognition effect is the worst, and the number of missed targets appears the most when two people are side by side or overlap. The reason for the poor recognition effect of humans is that the posture and shape of people are more complicated, and the number of training pictures is not enough to cover the postures of most people, and training is needed. (3) When recognizing fixed-wing targets, it is easy to misrecognize the target as a rotor (this is the case in 13 of the 15 problematic pictures), and it is necessary to strengthen the training distinction between these two types of targets.

Target	Model 1			Model 2		
	Train images	Test images	Accuracy	Train images	Test images	Accuracy
Airplane	127	111	86.49%	280	111	93.69%
Bird	146	132	87.12%	315	132	93.94%
Car	153	110	99.09%	454	110	100%
Person	123	110	65.45%	450	110	80.00%
Rotorcraft	110	117	90.60%	128	117	92.31%
Total	658	580	85.86%	1519	580	92.07%

FIGURE 5. Statistics of Model 1 and Model 2

In order to improve the detection effect of Model 1, 2856 source images of four types of targets including people, cars, birds, and fixed-wing aircraft and 2856 corresponding xml files were extracted from the network supplementary images and VOC2007 data set. Run model two on the same set of pictures, correctly identify 534 pictures, with a total correct rate of 92.07%. The statistical results of models one and two are shown in Figure 5 and Figure 6. It can be seen that compared with model 1, the recognition accuracy of model 2 is improved by 6.21%. The increase of the data set has made the model effect somewhat improved, but the recognition effect of people is still relatively poor.

Some test results are shown in Figure 7. Obviously we can see that almost all targets are detected and recognized correctly.

Target	Model 1		Model 2	
	Missed recognition	False recognition	Missed recognition	False recognition
Airplane	0	4	0	2
Bird	7	2	4	1
Car	1	0	0	0
Person	13	2	6	2
Rotorcraft	3	3	0	2

FIGURE 6. Performance of Model 1 and Model 2

FIGURE 7. Some test results

6. **Conclusions and Future Work.** This article analyzes the important position of border security and the application of existing border monitoring systems. Aiming at the high cost of monitoring pan-tilts and the difficulty of conventional servo mechanisms to meet high-precision positioning requirements, the application level of video monitoring systems is relatively low, insufficient application depth, and information The problem of insufficient analysis and processing capabilities and low level of intelligent application is proposed to use the Mobilenet-SSD network for target detection and recognition under the Caffe framework. With the help of neural network and computer vision technology, the effective information in the video image is fully extracted in the hope Achieve early warning and monitoring of key targets for border defense. For five types of typical monitoring targets, a model for the five types of monitoring targets is generated based on the self-made data set training under MobileNet-SSD, and the average accuracy of 87.73% and 86.69% is achieved on the self-made data set. In addition, based on the self-training model, code was written outside the self-made data set for target detection and recognition. Among 580 test images, the two self-training models reached 85.86% and 92.07% accuracy rates respectively.

Acknowledgment. This paper is supported by the Zhejiang Province Basic Public Welfare Research Program (LGG19F020021), Shanghai Automotive Industry Science and Technology Development Foundation (1815), Space Science and Technology Foundation (2019-HT-ZD-03), and Fundamental Research Funds for the Central Universities (2019QNA4059).

REFERENCES

[1] B. Liu, T. Chen, P. Fu, Y. Zhen and J. S. Pan, Method of Target Recognition in High-resolution Remote Sensing Based on Visual Saliency Mechanism and ROI Region Extraction, *Journal of Internet Technology*, vol. 20, no. 5, pp. 1333-1341, 2019.

[2] Y. Li, J. Li, J. S. Pan, Hyperspectral Image Recognition Using SVM Combined Deep Learning, *Journal of Internet Technology*, vol. 20, no. 3, pp. 851-859, 2019.
[3] L. Liang, J. S. Pan, Y. Zhuang, A Fast Specific Object Recognition Algorithm in a Cluttered Scene, *Journal of Internet Technology*, vol. 20, no. 7. , pp. 2023-2031, 2019.
[4] Q. Feng, C. Yuan, J. S. Pan, J. F. Yang, Y. T. Chou, Y. Zhou and W. Li, Superimposed Sparse Parameter Classifiers for Face Recognition, *IEEE Trans. On Cybernetics*, vol. 47, no. 2, pp. 378-390, 2017.
[5] H. Luo, F. X. Yu, J. S. Pan, S. C. Chu and P. Tsai, A Survey of Vein Recognition Techniques, *Information Technology Journal*, vol. 9., no. 6, pp. 1142-1149, 2010.
[6] J. S. Pan, Q. Feng, L. Yan and J. F. Yang, Neighborhood Feature Line Segment for Image Classification, *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 25, no. 3, pp. 387-398, 2015.
[7] J. B. Li, S. C. Chu, J. S. Pan, and L. C. Jain, Multiple Viewpoints Based Overview for Face Recognition, *Journal of Information Hiding and Multimedia Signal Processing*, vol. 3, no. 4, pp. 352-369, 2012.
[8] S. C. Chu, H. C. Huang, J. F. Roddick, J. S. Pan, Overview of Algorithms for Swarm Intelligence, ICCCI (1), pp. 28-41, 2011
[9] M. Everingham, L. V. Gool, C. K. I. Williams, J. Winn, A. Zisserman, The PASCAL Visual Object Classes (VOC) Challenge, *International Journal of Computer Vision*, vol. 88, no. 2, pp. 303-338, 2010.
[10] https://github.com/ultralytics/yolov5
[11] J. Redmon, A. Farhadi, YOLO9000: Better, Faster, Stronger, CVPR, 2017.
[12] J. Redmon, A. Farhadi, YOLOv3: An Incremental Improvement, CVPR, 2018.
[13] X. Zhang, X. Zhou, M. Lin, J. Sun, ShuffleNet: An Extremely Efficient Convolutional Neural Network for Mobile Devices, CVPR, 2018.
[14] F. N. Iandola, S. Han, M. W. Moskewicz, K. Ashraf, W. J. Dally, K. Keutzer, SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and <0.5MB model size, ICLR, 2017.
[15] Z. Zhao, P. Zheng, S. Xu, et al. Object Detection With Deep Learning: A Review. *IEEE Transactions on Neural Networks and Learning Systems*, vol. 30, no. 11, pp. 3212-3232, 2019.
[16] Girshick R, Donahue J, Darrell T, et al. Rich feature hierarchies for accurate object detection and semantic segmentation. 2013.
[17] K. He, X. Zhang, S. Ren, et al. Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. vol. 37, no. 9, pp. 1904-1916, 2015.
[18] R. Girshick, Fast R-CNN. ICCV, 2015.
[19] S Ren, K He, R Girshick, et al. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks, *IEEE Transactions on Pattern Analysis and Machine Intelligence*. vol. 39, no. 6, pp. 1137-1149, 2017.
[20] J. Redmon, S. Divvala, R. Girshick, A. Farhadi, You Only Look Once: Unified, Real-Time Object Detection, CVPR, 2015.
[21] W Liu, D Anguelov, D Erhan , et al. SSD: Single Shot MultiBox Detector, 2015.
[22] C. Szegedy, S. Ioffe, V. Vanhoucke, A. Alemi, Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning, AAAI, 2016.
[23] M. Lin, Q. Chen, S. Yan, Network in Network, ICLR, 2014.
[24] K. Simonyan, A. Zisserman, Very Deep Convolutional Networks for Large-Scale Image Recognition, ICLR, 2015.
[25] M. D. Zeiler, R. Fergus, Visualizing and Understanding Convolutional Networks, ECCV, 2014.
[26] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, H. Adam, MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications, CVPR, 2017.
[27] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, L. C. Chen, MobileNetV2: Inverted Residuals and Linear Bottlenecks, CVPR, 4510-4520, 2018.
[28] A. Howard, M. Sandler, G. Chu, L. C. Chen, B. Chen, M. Tan, W. Wang, Y. Zhu, R. Pang, V. Vasudevan, Q. V. Le, Hartwig Adam, Searching for MobileNetV3, ICCV 2019.
[29] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, T. Darrell, Caffe: Convolutional Architecture for Fast Feature Embedding, CVPR, 2014.
[30] J. Schmidhuber, Deep learning in neural networks: An overview, *Neural Networks*, vol. 61, pp. 85-117, 2015.