Comparative Study and Analysis of Different Target Detection Algorithms in Traffic Sign Detection

Chunli Wang, Long Xiangyu, Chu Zhongmin, Pan Mingfang
School of Tourism Data, Guilin Tourism University, Guilin, Guangxi, China

Abstract: The advancement of artificial intelligence in transportation has led to a burgeoning interest in the research of automatic identification technologies, particularly in the realm of traffic signs. It is an important pioneer technology of unmanned driving technology and has great theoretical value and application prospect. However, traffic sign detection is faced with the influence of complex weather factors such as rain, snow and fog, as well as the problem that the target is partially blocked and the size of the target is very small. Hence, selecting a target detection algorithm capable of swiftly and precisely identifying traffic sign categories is imperative. This paper compared various target detection algorithms, trained and tested YOLO v3, YOLO v4, SSD and other algorithms using the same traffic sign data set (30 classes), and finally concluded that the YOLO v4 network had the best effect, with a mAP value of 83.28% and a convergence interval of total loss between 3.5 and 4.

Keywords: Traffic Sign Recognition; YOLO; SSD; Algorithm Contrast

1. Introduction

Road facilities known as traffic signs utilize words or symbols to communicate guidance, restrictions, warnings, or instructions. These signs represent crucial measures in enforcing traffic management, guaranteeing road traffic safety, and ensuring smooth flow. Deep learning technology is used to accurately identify traffic signs and immediately convey relevant information to drivers, which can effectively avoid serious consequences such as traffic violations and traffic accidents caused by drivers' misreading or omission. At the same time, the technology is conducive to the development of driverless technology, driverless vehicles can determine their driving speed and driving path according to the relevant traffic signs identified.

Presently, the prevalent techniques employed for the identification of traffic signs encompass template matching, conventional machine learning, and the paradigm of deep learning.

1.1 Template Matching

The template matching method has a very high recognition rate and robustness for the identification of traffic signs in still pictures. However, the overall efficiency of the identification of deformed traffic signs in the pictures taken by cameras is low in the process of rapid vehicle running. Among the domestic scholars who use template matching method, the template matching method based on mathematical morphology recognition algorithm of Jiang Gangyi and Zheng Yi is the most representative.[1]

1.2 Traditional Machine Learning

Compared with the template matching method, the traditional machine learning method uses the principle that some features of the picture do not deform after translation, rotation and scaling, and then classifies the relevant features through the trained classifier, and finally achieves the effect of accurate classification of traffic signs.

1.3 Deep Learning

The deep learning approach, through its training methodology, autonomously extracts pertinent features from the provided dataset, enhancing the precision of selected features. Possessing a discernible degree of self-adaptability, it significantly elevates the accuracy in recognizing traffic signs. After the reconstruction and amplification of small-size traffic signs, domestic scholar Wu Haomin used feature extraction network and recognition network to extract and recognize relevant features, and successfully realized the recognition of small-size traffic sign images.

When testing driving videos at night, the recognition accuracy could reach 90.17%.[2]
In this paper, the adoption of the deep learning approach for traffic sign identification is opted for. Furthermore, the same dataset of traffic signs is employed for training and testing various target detection methods.

2. Traffic Sign Data Set Production

2.1 Data Source

The traffic sign image in the experiment comes from the network, and the picture is in PNG format. Special weather conditions such as sunny day, rainy day and foggy day are included. The foggy day picture is simulated by adding noise to the sunny day picture. Figure 1 is an example diagram of several typical traffic signs under different weather conditions.

(a)foggy day (b)rainy day (c)sunny day

Figure 1. Images of Traffic Signs in Different Weather Conditions in the Data Set

2.2 Construction of Traffic Sign Detection Data Set

In order to avoid over fitting phenomenon appeared in the process of model training, should as far as possible to join all kinds of traffic signs images under different weather conditions, increase the diversity of samples, make the trained model is reliable, stable, namely to reduce the image brightness, increase noise, shift and tilt of the image to simulate traffic sign pictures under different weather conditions, as shown in figure 2.

(a) reduce brightness (b) translation & inclination (c) add noise

Figure 2. Schematic Diagram of Image Preprocessing

This resulted in 15300 images.

3. Target Detection Algorithm

3.1 Yolo V3

The YOLO v3 algorithm, the most extensively utilized in the YOLO series, has outperformed accuracy in target detection algorithms, including Faster R-CNN. Building upon Darknet-19 from YOLO v2, YOLO v3 introduces the residual module and extends the network. With 53 convolutional layers, the enhanced network reduces the layer count while upholding classification accuracy. The calculation speed is greatly improved. Darknet-53 outperforms alternative network architectures in terms of floating-point calculations per second, enabling more efficient utilization of GPU resources. The graphical representation of its network structure is illustrated in Figure 3:

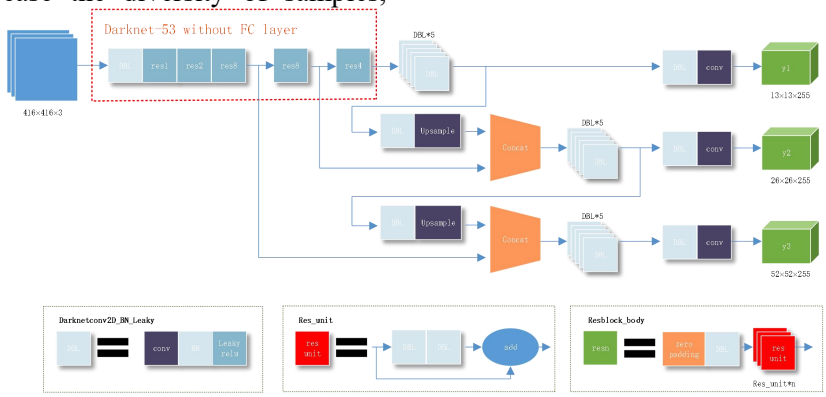

Figure 3. Network Structure Diagram of Yolo V3

DBL, the smallest constituent of YOLO v3, comprises three operations: convolution (DarknetConv), normalization (BN), and nonlinear (Leaky ReLU) activation functions. Resn is a residual module, with "n" denoting the quantity of residual units embedded within the module. Concat is a tensor splicing operation. Its function is to splice the up-sampled feature maps of the Darknet middle layer and a later layer. The splicing will expand the tensor dimension. Among them, y1, y2, and y3 serve the purpose of detecting targets of large, medium, and small targets, respectively.[3]

The loss function in YOLO v3 encompasses the positional error, confidence error, and categorical error. The precise formula is delineated as follows:

$$\begin{aligned} loss = {} & \lambda_{coord}\sum_{i=0}^{S^2}\sum_{j=0}^{B} I_{ij}^{obj}\left[\left(x_i-\hat{x}_i\right)^2+\left(y_i-\hat{y}_i\right)^2\right]+ \\ & \lambda_{coord}\sum_{i=0}^{S^2}\sum_{j=0}^{B} I_{ij}^{obj}\left(2-w_i\times h_i\right)\left[\left(w_i-\hat{w}_i\right)^2+\left(h_i-\hat{h}_i\right)^2\right]- \\ & \lambda_{obj}\sum_{i=0}^{S^2}\sum_{j=0}^{B} I_{ij}^{obj}\left[\hat{C}_i\ln\left(C_i\right)+\left(1-\hat{C}_i\right)\ln\left(1-C_i\right)\right]- \\ & \lambda_{noobj}\sum_{i=0}^{S^2}\sum_{j=0}^{B} I_{ij}^{noobj}\left[\hat{C}_i\ln\left(C_i\right)+\left(1-\hat{C}_i\right)\ln\left(1-C_i\right)\right]- \\ & \sum_{i=0}^{S^2}\sum_{j=0}^{B} I_{ij}^{obj}\sum\left[\hat{p}_i(c)\ln\left(\hat{p}_i(c)\right)+\left(1-\hat{p}_i(c)\right)\ln\left(1-p_i(c)\right)\right] \end{aligned} \quad (1)$$

λ_{coord} , λ_{obj} and λ_{noobj} respectively represent the proportion of each loss, S represents the grid size, B represents the number of candidate frames, I_{ij}^{obj} judges whether the i candidate frame in the i grid detects the target, if there is, the return value is 1, otherwise it is 0, I_{ij}^{noobj} is the opposite. x_i 、 y_i 、 w_i and h_i represent the predicted location value, $\hat{x}_i$ 、 $\hat{y}_i$ 、 $\hat{w}_i$ and $\hat{h}_i$ represent the true location value, C_i 、 p_i and $\hat{C}_i$ 、 $\hat{p}_i$ represent the predicted and true value of confidence and classification information, respectively.

3.2 Yolo V4

YOLO v4 is composed of many Tricks, which reduces training requirements and can be trained using a single GPU, while ensuring excellent performance of detection speed and accuracy. The YOLO v4 model incorporates CSPDarkNet53 as its backbone, integrates SPP as the supplementary module for the Neck, employs PANET as the feature fusion module of the Neck, and utilizes YOLO v3 as the Head. The detailed network architecture is illustrated in Figure 4:

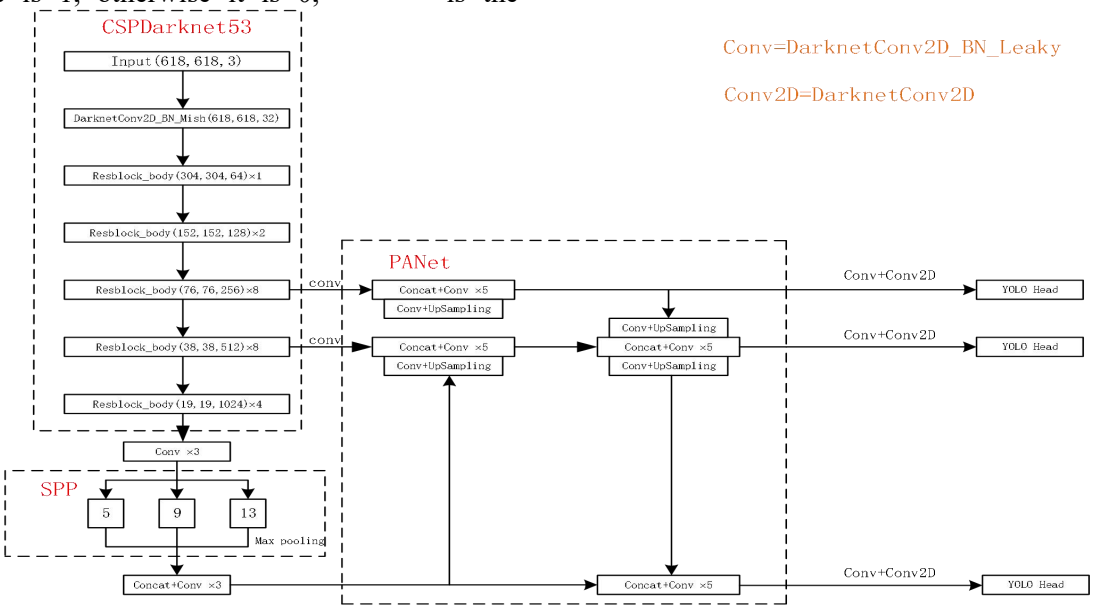

Figure 4. Network Structure Diagram of Yolo V4

In contrast to YOLO v3, YOLO v4 employs the ciou function in the positional loss component of its loss function, articulated as follows:

$$\begin{aligned} \Re_{CIoU} &= \frac{\rho^2\left(b,b^{gt}\right)}{c^2}+\alpha v \\ v &= \frac{4}{\pi^2}\left(\arctan\frac{w^{gt}}{h^{gt}}-\arctan\frac{w}{h}\right)^2 \\ \alpha &= \frac{v}{\left(1-IoU\right)+v} \\ \mathrm{L}_{CoU} &= 1-IoU+\frac{d^2}{c^2}+\alpha v \end{aligned} \quad .....(2)$$

v measures the consistency of aspect ratio, w^{gt} and w are the widths of the real frame and the predicted frame, and h^{gt} and h are the heights of the real frame and the predicted frame. [4]

3.3 SSD

SSD, an acronym for Single Shot Multibox Detector, stands out as one of the predominant frameworks for target detection. The SSD algorithm amalgamates the principles of Faster

R-CNN and YOLO, creating a synergistic approach to target detection. It not only uses a regression-based calculation model similar to YOLO, but also uses a concept based on region detection similar to Faster R-CNN. Compared with the former two, its recognition speed is Performance has significant advantages. The network structure diagram is as follows:

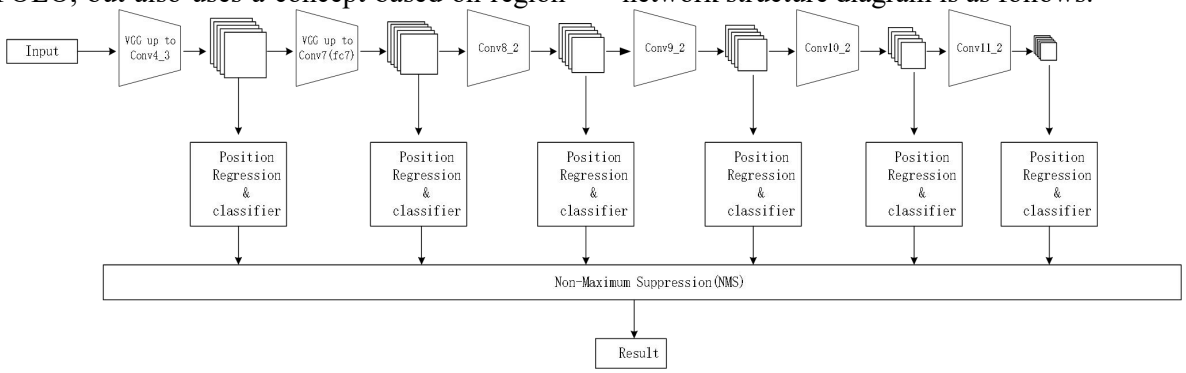

Figure 5. Network Structure Diagram of SSD

Derived from the VGG16 network structure, SSD transforms FC6 and FC7 into Conv6, a 3×3 convolutional layer, and Conv7, a 1 × 1 convolutional layer. This process involves eliminating all Dropout and FC8 layers, while simultaneously introducing Conv9, Conv10, and Conv11. The SSD loss function encompasses both confidence and location losses. The formula is:

$$\left.\begin{aligned} L(x,c,l,g) &= \frac{1}{N}\left(L_{conf}(x,c)+\alpha L_{loc}(x,l,g)\right) \\ L_{loc}(x,l,g) &= \sum_{\in Pos}^{N}\sum_{m\in cx,cy,w,h} x_{kij}\, smooth_{L1}\left(l_{mi}-\hat{g}_{mj}\right) \\ L_{conf(x,c)} &= -\sum_{i\in Pos}^{N} x_{pij}\ln\left(\hat{c}_{pi}\right)-\sum_{i\in Neg}\ln(\hat{c}_{0i}) \end{aligned}\right\} \quad \text{.....(3)}$$

$L_{loc}(x,l,g)$ represents location loss, $L_{conf(x,c)}$ represents confidence loss, α is the proportional coefficient that adjusts the proportion of these two losses, i is the prediction box number, j is the real box number, cx 、cy is the center coordinate of the default box, w 、h are the width and height of the default box. p represents the category, $\hat{c}_{pi}=\frac{e^{c_{pi}}}{\sum_{p} e^{c_{pi}}}$, the higher the $\hat{c}_{pi}$ the greater the probability that the object is p, and the $\hat{c}_{0i}$ is just the opposite.

4. Results and Discussion

4.1 Experiment Platform

This article uses the TensorFlow open source framework, the CPU model is Intel i5-10400, the CPU Clock Speed is 2.9GHz, the memory is 16GB, the GPU model is GTX1080TI, the operating system is ubuntu, the CUDA10.0 version, and the compiled language is python3.6.

4.2 The Evaluation Index

The performance assessment of a target detection algorithm frequently involves the utilization of accuracy rate and recall rate.[5] The accuracy rate represents the proportion of true and correct samples within the overall predicted correct outcomes. The calculation formula is as follows:

$$P=\frac{TP}{TP+FP}\times 100\% \quad \text{.....(4)}$$

Where P represents the accuracy rate, TP is the number of samples that divide the sample positive cases into positive cases, and FP is the number of samples that divide the negative samples into positive cases. The recall rate signifies the proportion of actual positive samples within the predicted samples relative to the number of predicted positive samples. Its calculation formula is expressed as follows:

$$R=\frac{TP}{TP+FN}\times 100\% \quad \text{.....(5)}$$

R is the recall rate, and FN is the number of samples that divide the sample positive cases into negative cases. In general, an increase in the recall rate is typically associated with a decrease in the accuracy rate. Therefore, we need to comprehensively consider these two parameters and use the AP (Average Precision) value to measure the performance of the algorithm. The formula is as follows:

$$AP=\sum_{k=1}^{N} P(k)\Delta R(k) \quad \text{.....(6)}$$

N represents the number of samples in the data set, $P(k)$ represents the accuracy of identifying different samples, and $\Delta R(k)$ represents the change of Recall when the number of samples changes from $k-1$ to k. For scenarios where the detection target spans multiple categories, the evaluation of algorithm models often involves the use of mAP (Mean Average Precision). The formula is articulated as follows:

$$mAP = \frac{AP}{C} \quad \text{.....(7)}$$

C is the total number of classes.

4.3 Comparison of Algorithm Performance

To ensure impartiality and precision, the YOLO v3, Tiny-YOLO v3, YOLO v4, and SSD networks undergo training and testing on an identical dataset. The distribution of each category of traffic signs within the test set is visually presented in Figure 6:

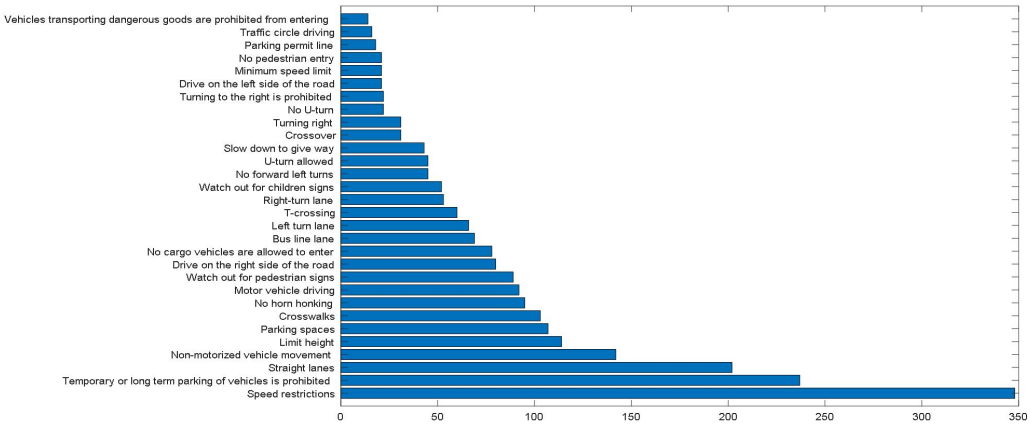

Figure 6. Number of Objects Per Class

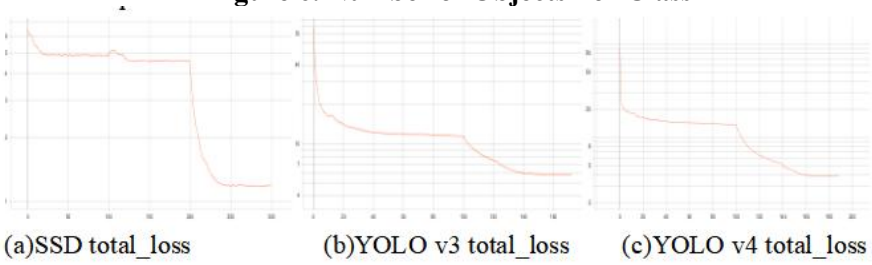

Figure 7.Comparison of Loss Curves

The loss curve of each model is shown in Figure 7, where the ordinate represents the loss value, and the abscissa represents the number of iterations.

The figure illustrates that both YOLO v3 and YOLO v4 exhibit a faster convergence of total loss, yet the total loss post-convergence is higher. The convergence interval for the total loss of YOLO v3 spans from 5~6, whereas the convergence interval for the total loss of YOLO v4 narrows down to 3~4, and the initial convergence of SSD The speed is faster, and the total loss convergence interval is 1~2. See Table 1, Table 2 and Table 3 for the mAP statistics table of all network models:

Table 1. The AP Comparison of Each Target Detection Framework on the Data Set

Detection framework	Pre-training	mAP	T-crossing	Non-motorized vehicle movement	Bus line lane	Traffic circle driving	Motor vehicle driving
SSD	Yes	60.77	69.45	67.06	77.42	27.08	62.27
YOLO v3	Yes	83.09	98.14	82.81	95.61	78.57	84.40
YOLO v4	Yes	83.28	93.48	83.18	98.47	72.60	86.20

Table 2. The ap Comparison of Each Target Detection Framework on the Data Set

Detection framework	Slow down to give way	Temporary or long term parking of vehicles is prohibited	No U-turn	No horn honking	Turning to the right is prohibited	No forward left turns	No pedestrian entry
SSD	66.20	77.33	45.45	60.48	62.23	64.07	76.22
YOLO v3	91.96	82.44	95.04	78.58	83.73	88.51	89.65
YOLOv4	94.57	88.33	84.28	77.26	90.08	90.07	90.24

Table 3. The Ap Comparison of Each Target Detection Framework on the Data Set

Detection framework	Vehicles transporting dangerous goods are prohibited from entering	No cargo vehicles are allowed to enter	Drive on the right side of the road	Drive on the left side of the road	Crosswalks	Crossover	Parking permit line
SSD	50.00	60.96	75.60	50.17	67.86	63.93	59.40
YOLOv3	83.10	78.88	92.34	87.47	79.84	93.55	84.41
YOLOv4	82.40	81.48	85.02	89.02	80.55	85.33	85.62

Table 4. The AP Comparison of Each Target Detection Framework on the Data Set

Detection framework	Parking spaces	Limit height	Speed restrictions	Turning right	Right-turn lane	U-turn allowed	Straight lanes
SSD	46.96	57.33	60.75	62.58	45.70	54.57	51.65
YOLOv3	62.44	85.52	84.88	81.40	60.57	80.90	69.53
YOLOv4	67.62	87.20	82.16	86.40	59.94	75.34	71.23

Table 5. The AP Comparison of Each Target Detection Framework on the Data Set

Detection framework	Watch out for children signs	Watch out for pedestrian signs	Minimum speed limit	Left turn lane
SSD	75.67	61.02	61.90	61.72
YOLOv3	93.92	83.65	65.82	75.11
YOLOv4	93.25	86.46	73.60	77.05

Table 6. Comparison of Sizes of Models

Detection framework	Picture size	Modle size
SSD	416×416	110.6M
YOLOv3	416×416	246.9M
YOLOv4	416×416	257.5M

The dimensions of each model are presented in the table below:

Observing the results, the mAP value for the YOLO model surpasses 83%, reaching 83.28% for YOLO v4, while the SSD model achieves a lower mAP value at only 60.77%, indicating that the YOLO v4 network model has a higher recognition accuracy in this training. However, the SSD network model occupies a small amount of memory and is more suitable for porting to FPGA and other hardware to run. The memory occupied by the YOLO v3 model is slightly smaller than that of the YOLO v4, and the recognition accuracy differs by only 0.19%. Comprehensively, the performance of the YOLO v4 network model is even better.

5. Conclusions

The YOLO v3, YOLO v4, and SSD target detection algorithms undergo both training and testing phases utilizing an identical dataset of traffic signs. Based on the ultimate test outcomes, it is inferred that the YOLO v4 algorithm model exhibits superior performance.

Looking to the future, we hope that there will be a target detection algorithm model with higher accuracy, faster speed, and smaller model. It will be deployed in FPGA or DSP to further accelerate the speed of algorithm recognition.

Acknowledgment

This paper is supported by the student innovation project of Guangxi in 2019(201913644). 2020 Project of Three-wide education of Institute of Information Technology of GUET(2020SQ03).

References

[1]Jiang Gangyi,Zheng Yi. Automatic traffic sign recognition based on mathematical morphology. Journal of Shantou University(Natural Science Edition),1998,1998(013):90-96.

[2]Wu Haomin.Research and Implementation of Road Traffic Sign Recognition Based on Driving Video.Nanjing University of Posts and Telecommunications for the Degree of Master of Engineering,2020.

[3]Yang Yanfei,CaoYang.Improved glass insulator detection in Yolov3 drone shot.Computer Engineering and Applications,2021,1-11.

[4]Zheng Z,Wang P,Liu W,et al.Distance-IoU Loss: Faster and Better Learning for Bounding Box Regression. arXiv, 2019.

[5]Li Zhou,Huang Miaohua.Real-time vehicle detection based on YOLO_v2 model.Chinese Journal of Mechanical Engineering,2018,29(15):1869-1874.

[6]Redmon J,Divvala S,Girshick R and Farhadi A. You Only Look Once: Unified, Real-Time Object Detection.2016 IEEE Conference on Computer Vision and Pattern Recognition(CVPR), Las Vegas, NV, United States, 2016: 779-788.

[7]Garcia C,Delakis M.Convolutional face finder:A nerual architecture for fast and robust face detection.IEEE Transactions on Pattern Analysis and Machine Intelligence,2004, 26(11):1408-1423.

