Оптический журнал

Иконика

Real-time and efficient eyes and mouth state detection: an artificial intelligence application based on embedded systems

© 2020 Fei Liu*, **, Changcheng Qin*, **, Hongliu Yu*, **, ***

*Rehabilitation Engineering and Technology Institute, University of Shanghai for Science and Technology, Shanghai, China,200093

**Shanghai Engineering Research Center of Assistive Devices, Shanghai, China,200093

***Key Laboratory of Neural-functional Information and Rehabilitation Engineering of the Ministry of Civil Affairs, Shanghai, China,200093

E-mail: yhl98@hotmail.com

Submitted 14.07.2020

DOI:10.17586/1023-5086-2020-87-12-57-66

It is reported that many traffic accidents are caused by fatigued driving. The state detection of eyes and mouth is usually employed to judge whether the driver is fatigue. However, the traditional image processing methods cannot achieve satisfactory detection accuracy due to the changes of illumination, head posture, and other factors in the actual environment. To date, although the methods based on deep learning have reached adequate accuracy in the tasks of object detection, they are obliged to rely on high hardware configuration to meet the real-time requirements. To achieve satisfactory detection accuracy in eyes and mouth detection task and obtain good real-time performance on embedded platforms such as NVIDIA Jetson TX2 by improving the object detection algorithm based on deep learning. In this paper, based on the original YOLOv3-Tiny structure, we not only added the structure of deep residual learning, but also calculated six new anchor boxes according to the training set using the K-means algorithm. We also used six data augmentation methods in the training set to improve the detection accuracy. The improved algorithm proposed in this paper can achieve satisfactory detection speed and accuracy on the TX2 embedded hardware configuration platform, validating that the ameliorated scheme is effective.

Keywords: *eyes state, mouth state, YOLOv3-Tiny, deep learning, object detection, real-time embedded systems.*

OCIS codes: 100.3008.

Определение состояния глаз и рта в реальном масштабе времени: применение искусственного интеллекта на основе встроенных систем

© 2020 г. Fei Liu, Changcheng Qin, Hongliu Yu

Известно, что многие из дорожных происшествий вызываются утомлением водителя. Часто для определения состояния утомления используется анализ состояния глаз и рта. Однако традиционные способы обработки изображений не дают удовлетворительной точности из-за изменений

освещённости, положения головы и других факторов реального окружения. Хотя методы, основанные на глубоком обучении, достигли надлежащей точности обнаружения объектов, для работы в реальном масштабе времени они требуют применения мощных вычислителей. Для достижения удовлетворительной точности обнаружения глаз и рта в реальном масштабе времени в случае применения встроенных платформ, таких как NVIDIA Jetson TX2, необходимо улучшение алгоритмов распознавания объектов на основе глубокого обучения. В настоящей работе, основанной на оригинальной архитектуре YOLOv3-Tiny, мы не только дополнительно применили глубокое разностное обучение (deep residual learning), но также вычислили на основе обучающих наборов шесть новых опорных рамок (anchor boxes) с использованием алгоритма K-средних. Для повышения точности обнаружения были также использованы шесть методов дополнения наборов обучающих данных. Усовершенствованный алгоритм, предложенный в работе, обеспечивает удовлетворительную точность и скорость обнаружения на встроенной платформе типа TX2, подтвердив тем самым эффективность предлагаемых решений.

Ключевые слова: *состояние глаз, состояние рта, архитектура YOLOv3-Tiny, глубокое обучение, распознавание объектов, встроенные системы реального времени.*

1. INTRODUCTION

In recent years, there are more and more traffic accidents caused by fatigued driving, leading to great damage to the physical and mental health of the injured. In order to diminish the road accidents caused by fatigued driving, researchers have put forward many solutions to detect driver fatigue. Some of them proposed to wear EEG (electroencephalogram) [1], ECG (electrocardiograph) [2], EOG (electrooculogram) [3] et al. monitoring devices on the driver to obtain the physiological parameters of the drivers, but they could affect the comfort of the driver. Some of them predicted whether the driver is drowsy according to the angle of steering wheel rotation and the speed of the vehicle [4], whereas the predicted results are often affected by driving proficiency, driving conditions, and other factors. Moreover, some other researchers analyzed the fatigue state of drivers through the state of eyes and mouth in a real-time video without installing monitoring devices on the drivers [5–7], which is highly praised by users and thus promising.

Ji et al. [8] proposed an algorithm to analyze the state of eyes and mouth by extracting contour features, which could adapt to different lighting and background environments. Nevertheless, if the tested person wears glasses, the test results will be affected. Zhao et al. [9] reported a state detection method for drowsiness based on the deep convolution neural network, which could avoid the complex manual feature extraction and effectively reduced the interference from environmental factors. However, when the head moves, the detection accuracy will be reduced. Jakubowski J. and J. Chmielińska [10] developed the method of convolutional neural network and transfer learning to judge the driver fatigue by detecting the characteristics of eyes, mouth, and eyebrows. However, the detection accuracy of this algorithm will be lower when the light is darker.

Since AlexNet [11] achieved great success in the ImageNet image classification competition in 2012, the convolutional neural network (CNN) has been widely used in the field of computer vision. Object detection algorithms based on deep learning not only do not need numerous artificial feature designs but also show good feature expressions and detection accuracy. Based on different design ideas, these algorithms can be classified into two categories. One is the two-stage object detection algorithm represented by R-CNN [12] (Region-CNN), Fast R-CNN [13], and Faster R-CNN [14]. The other is the single-stage object detection algorithm represented by YOLO (You Only Look Once) [15–17], SSD (Single Shot MultiBox Detector) [18], and RetinaNet [19].

These object detection algorithms use deep neural networks to detect and classify the object, with YOLOv3 adding the FPN (feature pyramid network) [20] to improve the accuracy of small object detection, showing the advantages of fast detection speed and high detection accuracy. However, although YOLOv3 exhibits excellent performance, it still needs to rely

on the hardware conditions of high configuration in the real-time detection task. In our test, using the YOLOv3 algorithm on NVIDIA TX2 platform for real-time eyes and mouth state detection could only obtain 3.2FPS (Frames Per Second) detection speed.

YOLOv3-Tiny is a simplified version of YOLOv3, with faster running speed while lower detection accuracy. In this paper, we used the improved YOLOv3-Tiny algorithm (called YOLOv3Tiny-EM) to detect the state of eyes and mouth. Compared with the YOLOv3-Tiny algorithm, the detection accuracy of this improved algorithm is significantly enhanced, although the running speed is slightly reduced.

The rest of this paper is arranged as follows: The second section introduces the way of data acquisition and several methods of data augmentation. The third section presents the improved YOLOv3-Tiny algorithm. The relevant evaluation criteria are introduced in the fourth section. The related experiments and the discussions of the experimental results are shown in the fifth section. The last section is the conclusion and prospect of this article.

2. DATASET

2.1. Image data acquisition

The acquiring dataset is a substantially important step in deep learning. The dataset of this experiment mainly came from mobile phone shootings and the Internet. We chose 28 people of different ages and collected about 562 images of eyes and mouth in different light, backgrounds, and angles with the front camera of the iPhone XS. In addition, we also used Python scripts to crawl the appropriate dataset in the Baidu web page, and got 198 images.

2.2. Image data augmentation

Data augmentation can improve the robustness of models by increasing the number and diversity of training samples. Furthermore, changing the training samples randomly can reduce the dependence of the model on some parameters and improve the generalization ability of the model. In this study, we employed six data augmentation methods (Fig. 1) and finally got 2118 images.

• Data augmentation: brightness

In the detection task, the illumination of the environment affects the brightness of the detected face, which may lead to inferior detection results. Therefore, we used brightness transformation method to change the ambient light of the detected objects, so as to improve the robustness of the model. In this study, the RGB value of the training images were multiplied by 0.5 or 1.5 randomly to change the brightness of the original images.

• Data augmentation: color

In this experiment, the dataset was mainly from the same race. Since people in different regions exhibit different skin colors, we converted some of the images into gray-scale images to enhance the generalization ability of the model.

• Data augmentation: contrast

In our study, the external ambient light might cause the difference between the camera detection result and the actual image. In this case, the eyes and mouth contour of the detected face will be blurred, which could affect the detection results. Therefore, we used a histogram equalization algorithm to change the contrast of the original image.

• Data augmentation: blur

In the process of real-time detection, the results may become blurred due to facial movements and camera focusing problems, affecting the recognition accuracy. Therefore, this study

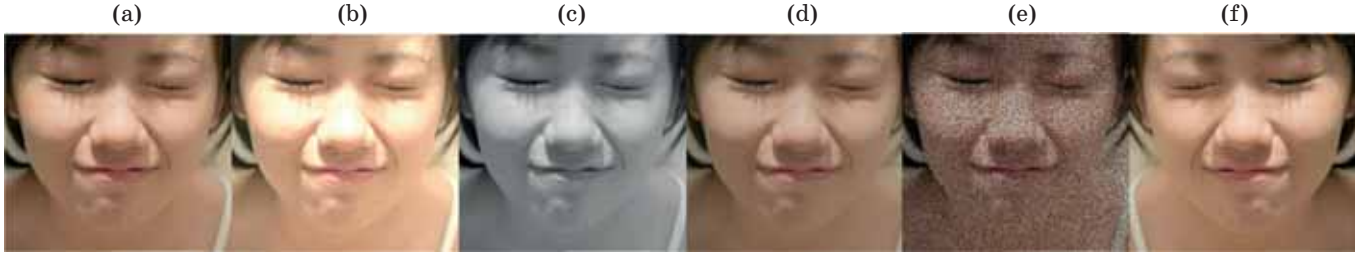

Fig. 1. Image augmentation methods. Brightness transformation (a), contrast enhancement transformation (b), color transformation (c), blur processing (d), add noises (e), horizontal mirror (f).

adopted the mean template method to randomly blur the original images with the convolution kernel of 7×7.

• Data augmentation: noise

Overfitting usually occurs when the neural network handles high-frequency data, which not only affects the tasks of the neural network but also has an impact on low-frequency features. In order to eliminate high-frequency features, we randomly added salt and pepper noise data with SNR (signal-to-noise ratio) of 0.9.

• Data augmentation: mirror image

In order to obtain more training sets to improve the performance of the model, we also horizontally mirrored the original images to get more dataset.

3. METHODOLOGIES

3.1. YOLO

The object detection of the RCNN series contain two steps: object proposal and object classification. YOLO series retain both two branches, and only uses one network to output object location and classification simultaneously at one time, thus it is faster than Faster-RCNN. YOLO divides the input image into $s \times s$ grids, and each grid outputs an array with dimension $B * 5 + C$, where B is the number of bounding boxes predicted per grid and C is the number of categories to be predicted. YOLO network structure mainly comprises two parts: the first part is feature extraction network, which is primarily used to extract the general features of the object; the second part is the post-processing network, whose purpose is to return the coordinates and categories of the object being detected. Figure 2 is the YOLO detection model.

3.2. K-means algorithm

The second version of the YOLO series begins to add the anchor boxes which are for the prediction of the bounding boxes. YOLOv3 uses nine anchors and YOLOv3-Tiny uses six anchors. These anchors are obtained by clustering the VOC dataset. There is a big gap in the size of 20 categories in VOC dataset, among which the big objects are bicycles and buses, and the small objects are birds and cats. In the task of

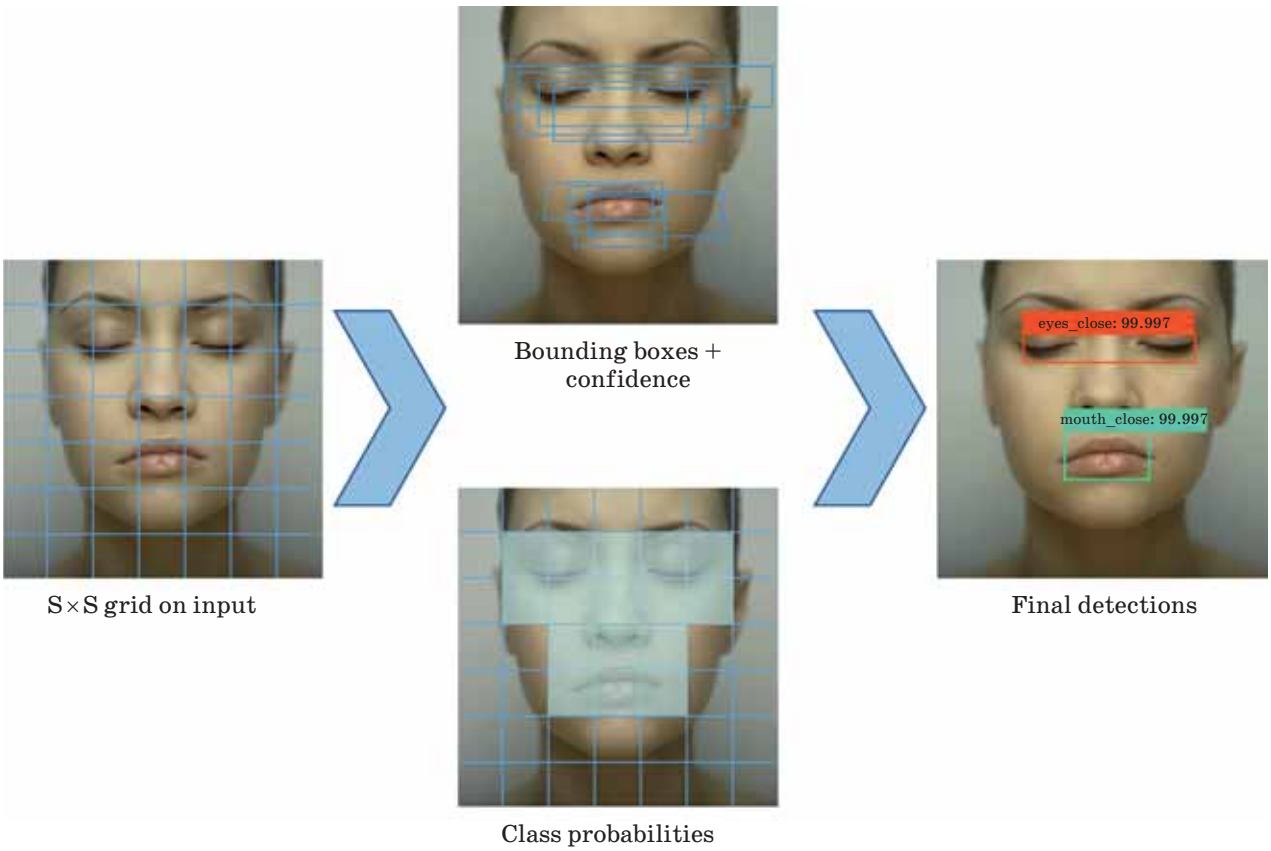

Fig. 2. YOLO detection.

eyes and mouth state detection, some of the anchors proposed by previous studies are unreasonable, thus we calculated new anchors according to the training set using the k-means algorithm to improve the detection effect of the model.

K-means algorithm mainly contains the following four processes:

1. Select the number of clusters, and initialize the center location of the cluster.

2. For each sample in the dataset, calculate its distance to the center of each cluster, and classify the sample into the cluster with the smallest distance.

3. Recalculate the central location of each cluster.

4. Repeat step 2 and step 3 until the center of the clusters no longer changes.

The traditional K-means algorithm uses the standard Euclidean distance method, which is deficient in that: the big box will produce more errors than the small box. Therefore, we replaced the standard Euclidean distance function with the other distance measure function. The purpose of clustering is to make the anchor box and the adjacent ground truth have larger IoU (Intersection over Union) value, which is not directly related to the size of the anchor box. The new distance measurement formula is:

$$d(\text{box}, \text{centroid}) = 1 - IOU(\text{box}, \text{centroid}). \quad \textbf{(1)}$$

3.3. Loss function

The loss function of the improved YOLOv3-Tiny is defined as follows:

$$\begin{aligned} Loss(\text{object}) = {} & \lambda_{\text{coord}} \sum_{i=0}^{K\times K}\sum_{j=0}^{M} I_{ij}^{\text{obj}} \left[(x_i-\hat{x}_i)^2+(y_i-\hat{y}_i)^2\right]+ \\ & +\lambda_{\text{coord}} \sum_{i=0}^{K\times K}\sum_{j=0}^{M} I_{ij}^{\text{obj}} (2-w_i\times h_i)\left[(w_i-\hat{w}_i)^2+(h_i-\hat{h}_i)^2\right]- \\ & \lambda_{\text{noobj}} \sum_{i=0}^{K\times K}\sum_{j=0}^{M} I_{ij}^{\text{noobj}} \left[\hat{C}_i \log(C_i)+(1-\hat{C}_i)\log(1-C_i)\right]- \\ & -\lambda_{\text{noobj}} \sum_{i=0}^{K\times K}\sum_{j=0}^{M} I_{ij}^{\text{noobj}} \left[\hat{C}_i \log(C_i)+(1-\hat{C}_i)\log(1-C_i)\right]- \\ & \sum_{i=0}^{K\times K} I_{ij}^{\text{obj}} \sum_{c\in\text{classes}}^{K\times K} \left[\hat{p}_i(c)\log(p_i(c))+(1-\hat{p}_i(c))\log(1-p_i(c))\right]. \end{aligned} \quad \textbf{(2)}$$

In formula 2, K is the number of grids, M represents the number of bounding boxes per grid; $\hat{x}_i, \hat{y}_i$ are values of the center coordinate of the predicted bounding box; $\hat{w}_i$, $\hat{h}_i$ are values of height and width of the predicted bounding box; x_i, y_i, w_i, h_i are true values. The value of $I_{\text{ij}}^{\text{obj}}$ s either 0 or 1,when the j-th anchor box of the i-th grid is responsible for this object, its value is 1, otherwise its value is 0; $I_{\text{ij}}^{\text{noobj}}$ means that the j-th anchor box of the i-th grid is not responsible for this object; $\hat{C}_i$ is the predicted confidence, C_i is the true confidence; $p_i(c)$ represents the true probability that the object belonging to class c is in grid i, $\hat{p}_i(c)$ is the predicted value; λ_{coord} is defined as the weight of the coordinate error, λ_{noobj} is the weight of the IoU error, the value of λ_{coord} is 5 and the value of λ_{noobj} is 0.5 in this study.

The first part of formula 2 is used to calculate the error of the central coordinate. The second part is used to calculate the error of the width and height coordinate. The third part is used to calculate the error of confidence. The last part is used to calculate the error of the classification.

3.4. Proposed algorithm

YOLOv3-Tiny is a simplified version of YOLOv3, the primary difference is that: YOLOv3-Tiny does not contain residual layers [21], instead, the backbone network uses 7-layer Conv+Max networks to extract features (similar to darknet 19), and it includes feature maps of 13×13 and 26×26 for object detection. After compression, YOLOv3-Tiny runs extremely fast, but its precision decreases obviously. One of the crucial rea-

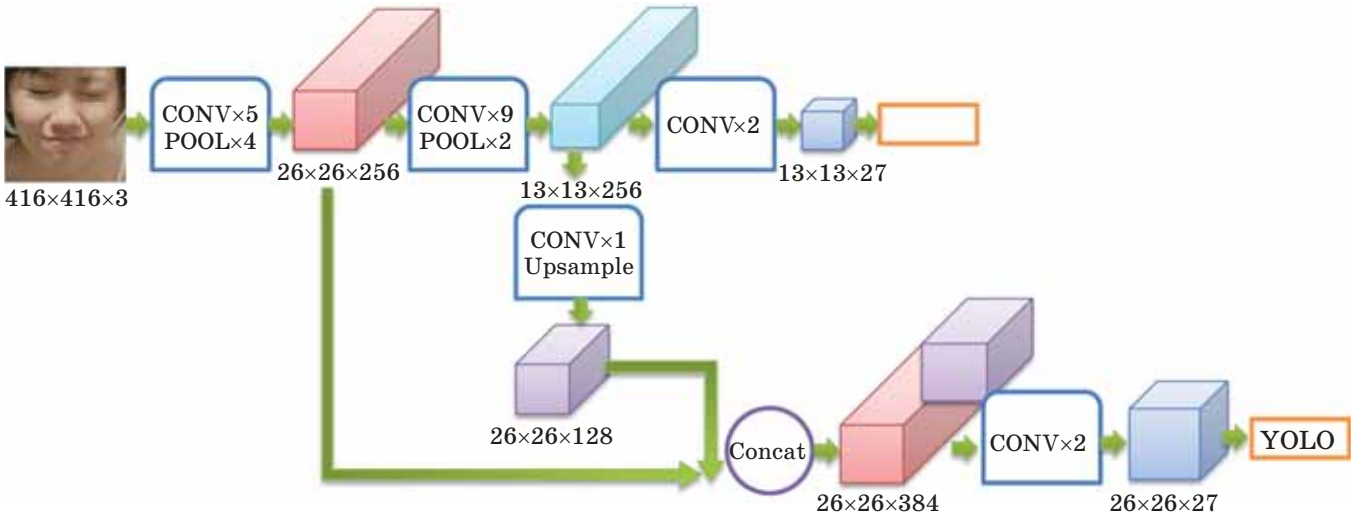

Fig. 3. YOLOv3Tiny-EM network structure diagram.

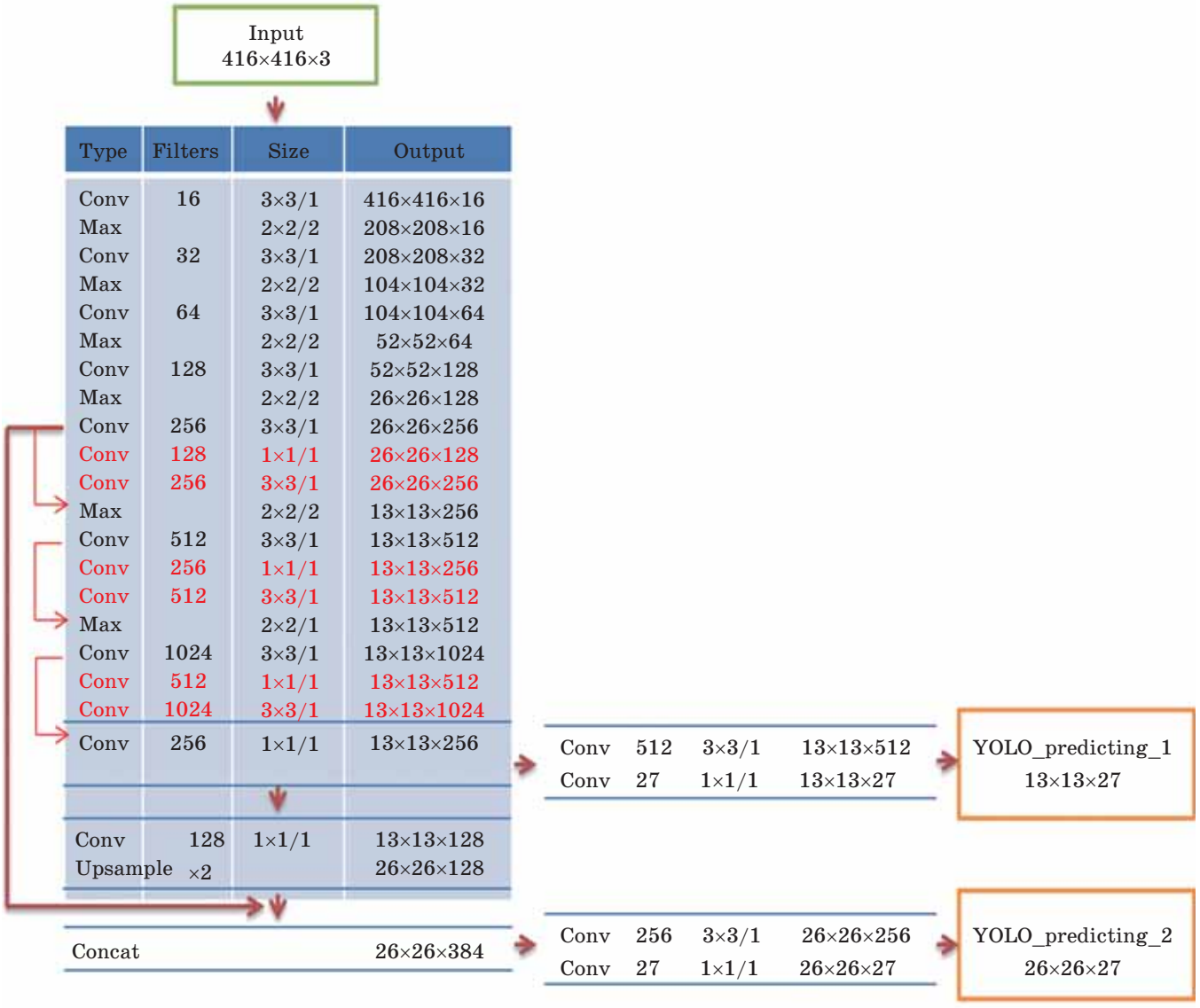

Fig. 4. Network parameters of YOLOv3Tiny-EM.

sons for the low accuracy of detection is that the backbone network of YOLOv3-Tiny is shallow and it cannot extract higher-level semantic features. Therefore, we proposed a new model (YOLOv3Tiny-EM) to improve the performance of YOLOv3-Tiny, and Fig. 3 is the new network structure diagram.

The same as YOLOv3-Tiny, our improved model uses two feature maps with different scales for detection to achieve satisfactory detection speed on the TX2 embedded platform.

In order to enhance the accuracy of detection, we will think of increasing the number of network layers to extract the features of deeper networks. However, simply increasing the depth in the original network will lead to the problem of vanishing gradient or exploding gradient. The method of adding residual blocks [21] can not only suppress the above problems caused by directly adding ordinary convolution layer, but also improve the performance of the model. Therefore, in this study, we added three residual blocks (the red part of Fig. 4) based on the original YOLOv3-Tiny to extract more abundant semantic information and improve the detection accuracy.

In the training procedures, YOLOv3Tiny-EM model outputs two feature maps with different scales. These two outputted feature maps are capable of dividing six anchor boxes equally according to their size. Moreover, Non-maximum Suppression (NMS) and confidence score are employed to generate the final output bounding box predictions.

Specifically, the number '27' in the 2 tensors that are outputted by the model is from '3×(1+4+4)', where '1' represents the target score of the box, '3' means that an output grid cell contains three prediction bounding boxes, the first '4' serves the four coordinate information of the box (center point, height, and width); the second '4' tells the number of categories contained in the dataset (eyes_open, eyes_close, mouth_open, and mouth_close).

The specific network parameters of YOLOv3Tiny-EM as well as how the improved network enhances feature propagation and promotes feature fusion are shown in Fig. 4.

4. EVALUATION CRITERION

4.1. Precision and recall

Precision and recall are usually contradictory performance measurement, where recall is low when precision is high. True classification and predicted classification of the samples may not be identical. The confusion matrix is a standard format for precision evaluation. For binary classification tasks, samples can be categorized into the following four types: True Positive (TP), False Positive (FP), True Negative (TN), and False Negative (FN). When these four indicators are presented in the table altogether, the following confusion matrix (Table 1) can be obtained.

Precision and recall are defined as follows:

$$\text{Precision} = \frac{\text{TP}}{\text{TP}+\text{FP}} = \frac{\text{TP}}{\text{all detections}}, \quad (3)$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP}+\text{FN}} = \frac{\text{TP}}{\text{all ground truths}}. \quad (4)$$

4.2. mAP (mean average precision)

The mAP is an evaluation criterion of accuracy in the object detection algorithm, which involves two concepts: precision and recall. In the object detection task, precision and recall can be calculated for each class. After calculation, each class can get a PR curve, and the area under the curve is the value of AP. The mAP is the average AP of all categories, where AP is calculated as follows: Firstly, threshold values were set as [0, 0.1, 0.2, . . . , 1]. If the recall is greater than the threshold, a corresponding maximum precision will be obtained. In this way, we finally calculated 11 precision values and obtained AP by averaging these 11 precision values. This method is called 11-point interpolated average precision. The definition of AP is shown as follows:

$$\text{AP} = \frac{1}{11} \sum_{r\in\{0,0.1,\ldots,1\}} \rho_{\text{interp}\,(r)}, \quad (5)$$

$$\rho_{\text{interp}}(r) = \max_{\tilde{r}\geq r} \rho(\tilde{r}). \quad (6)$$

Table 1. Confusion matrix for the classification

Labeled	Predicted	Confusion matrix
Positive	Positive	TP
Negative	Negative	TN
Positive	Negative	FN
Negative	Positive	FP

4.3. Speed of detection

The NVIDIA TX2 embedded development board and Logitech C920 Pro camera were applied as the test environment. When using the camera for real-time eyes and mouth state detection, the number of frames per second was indicated as the evaluation criterion.

5. EXPERIMENTS AND DISCUSSION

5.1. Experimental results

The experimental training environment is shown in Table 2, and the improved network settings of initialization parameters are shown in Table 3.

In the improved model, the input image was adjusted to 416×416 pixels, and two convolutional layers were added to the backbone network to extract higher-level semantic features. Also, the K-means algorithm was used to generate six new anchor boxes according to the training set. The model was trained after the configuration of parameters and the environment. The learning rate decreased to 0.0005 after 15000 steps, and to 0.00025 after 23000 steps.

In this work, we used improved network YOLOv3tiny-EM to detect the four state categories of eyes and mouth, which are eyes_open, eyes_close, mouth_open, and mouth_close.

Table 2. Experimental training environment

Type	Details
CPU	Intel(R) Core i9-9900k(32GB)
GPU	NVIDIA Tesla V100(32GB)
Operating system	Ubuntu 16.04
Language	C Programming Language
Development Framework	Darknet

Table 3. Initialization parameters of YOLOv3Tiny-EM network

Size of Input images	Batch/ subdivisions	Initial Learning Rate	Momentum	Training steps
416×416	64/8	0.001	0.9	30000

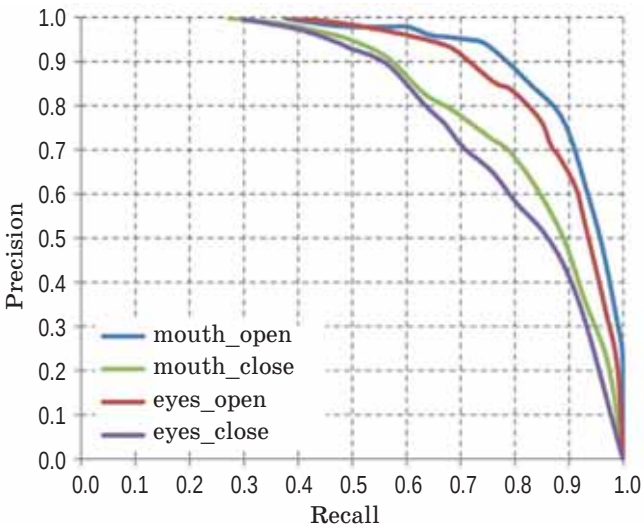

Fig. 5. P-R curves of four classes.

It can be seen from the P-R curves (Fig. 5) that the performance of the improved novel network model is satisfactory. However, it is showed that the experimental result of eyes_close is slightly worse than other classes. Through analysis, we think the reason may be that there are not enough face images with glasses in the training samples.

5.2. Comparison of two algorithms

In order to validate that the new model YOLOv3Tiny-EM proposed in this paper has good performance, we also used the original YOLOv3-Tiny to train the labeled dataset.

In the YOLOv3Tiny-EM algorithm, we added three residual blocks based on the original YOLOv3-Tiny and used the anchor boxes generated by K-means algorithm to replace the original ones.

The experimental results (Table 4) show that the effect of YOLOv3Tiny-EM algorithm is better than YOLOv3-Tiny in the task of eyes and mouth state detection. Furthermore, it also shows that the running speed of YOLOv3Tiny-EM is not significantly reduced.

Table 4. Test results of two methods

Method	Input	Transmission rate/(FPS)	mAP, %
YOLOv3-Tiny	416×416	23.8	79.3
YOLOv3Tiny-EM	416×416	21.2	87.7

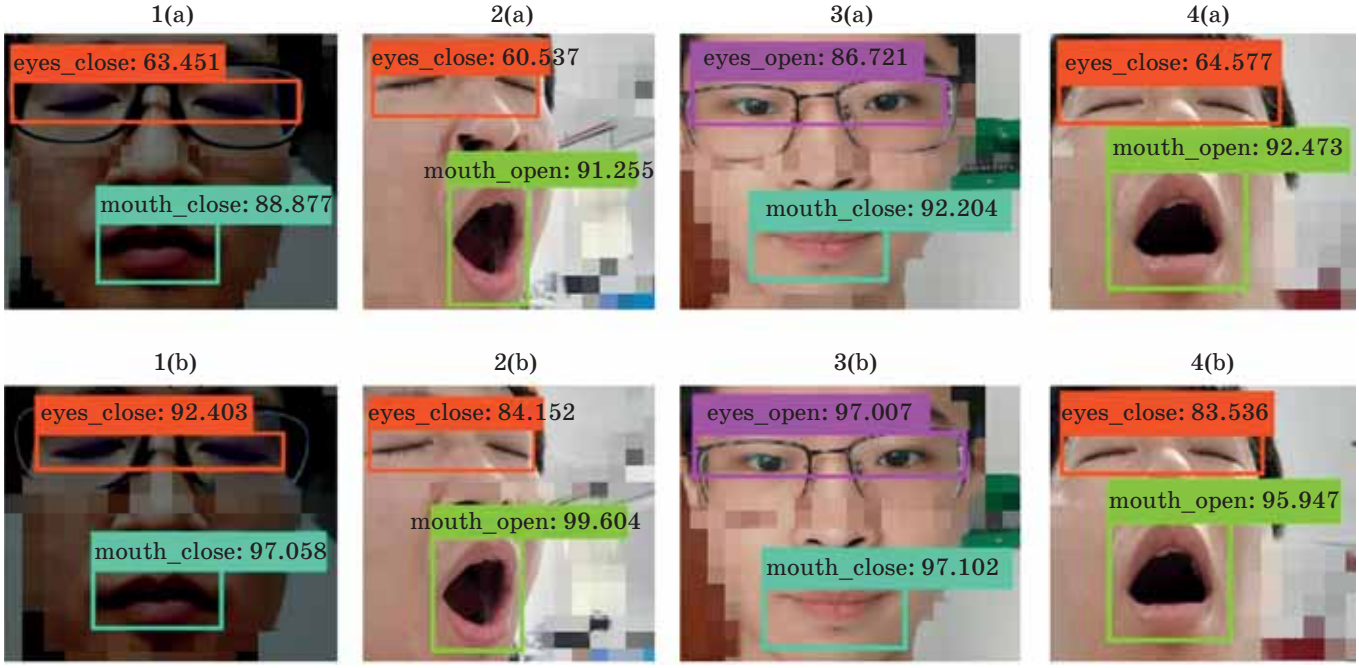

Fig. 6. Test results: 1a, 2a, 3a, 4a are detected by YOLOv3-Tiny; 1b, 2b, 3b, 4b are detected by YOLOv3Tiny-EM.

In order to display the performance of two models more intuitively, we chose different scenarios (distance, brightness, angle, etc.) for model testing. These results are shown in Fig. 6.

It can also be seen from Fig. 6 that the detection performance of YOLOv3Tiny-EM is better than YOLOv3-Tiny under the same condition.

5.3. Analysis of influencing factors

• Influence of data augmentation methods

In this study, we used 6 data augmentation methods. To verify the effect of data augmentation methods, we firstly used YOLOv3Tiny-EM model to train the original images without data augmentation methods, then removed one of the methods and trained again. The obtained mAP values are shown in Table 5. Based on the mAP values, it suggests that the performance of YOLOv3Tiny-EM is significantly improved after using the data augmentation methods. Table 5 shows the mAP values for models trained using the control variable method.

• Influence of using new anchor boxes

Before training, we used the k-means algorithm to calculate six new anchor boxes (44 28,68 53,144 36,108 53,293 58,151 122) based on our own dataset, aiming to improve the detection accuracy. The initial anchor boxes and the six new anchor boxes were used to test the performance respectively. As the final results shown in the table, the mAP improved by 2.9% after using the new anchors. Table 6 shows the mAP values of two different situations.

Table 5. mAP values for models trained using the control variable method

Method	mAP, %
Dataset after augmentation	87.7
Remove brightness transformation	84.9
Remove contrast enhancement transformation	85.6
Remove color transformation	83.6
Remove blur processing	83.1
Remove noise processing	82.7
Remove mirror processing	86.3

Table 6. mAP values of two different situations

Method	mAP, %
Initial anchor boxes	84.8
New anchor boxes	87.7

CONCLUSIONS

In this study, we added three residual blocks to the initial YOLOv3-Tiny model to enhance the feature extraction performance of the neural network. Furthermore, we applied the six data augmentation methods to the dataset, and six new anchor boxes were generated by the K-means algorithm based on our dataset. The experimental results suggested that the detection accuracy of the improved YOLOv3-Tiny algorithm was significantly enhanced in the PR curve, tabular data, and detection results from the corresponding models, compared to the original YOLOv3-Tiny. Our next task is to apply the proposed algorithm to the fatigue driving detection task. When the frequency of driver's mouth opening and eyes closing is abnormal, the system will generate an alarm signal. Moreover, we will further optimize the YOLOv3Tiny-EM model to improve the speed and accuracy of detection.

The work reported in this paper is sponsored by Shanghai Pujiang Program, number:16PJC063.

REFERENCES

1. *Mardi Z., Ashtiani S.N.M., Mikaili M.* EEG-based drowsiness detection for safe driving using chaotic features and statistical tests // Journal of medical signals and sensors. 2011. V. 1. P. 130.
2. *Jung S.-J., Shin H.-S., Chung W.-Y.* Driver fatigue and drowsiness monitoring system with embedded electrocardiogram sensor on steering wheel // IET Intelligent Transport Systems. 2014. V. 8. P. 43–50.
3. Fatourechi M., Bashashati A., Ward R.K., Birch G.E. EMG and EOG artifacts in brain computer interface systems: A survey // Clinical neurophysiology. 2007. V. 118. P. 480–494.
4. *Ma J., Murphey Y.L., Zhao H.* Real time drowsiness detection based on lateral distance using wavelet transform and neural network // IEEE symposium series on computational intelligence. 2015. P. 411–418.
5. *Yazdi M.Z., Soryani M.* Driver drowsiness detection by identification of yawning and eye closure // International journal of automotive engineering. 2019. V. 9. P. 3033–3044.
6. *Xiao Z., Hu Z., Geng L., Zhang F., Wu J., Li Y.* Fatigue driving recognition network: fatigue driving recognition via convolutional neural network and long short-term memory units // Iet Intelligent Transport Systems. 2019. V. 13. P. 1410–1416.
7. *Zhang F., Su J., Geng L., Xiao Z.* Driver fatigue detection based on eye state recognition // International Conference on Machine Vision and Information Technology (CMVIT). 17–19 Feb. 2017. Singapore. P. 105–110.
8. *Ji Y., Wang S., Lu Y., Wei J., Zhao Y.* Eye and mouth state detection algorithm based on contour feature extraction // Journal of Electronic Imaging 2018. V. 27. P. 051205.
9. *Zhao G., Liu S., Wang Q., Hu T., Chen Y., Lin L., Zhao D.* Deep convolutional neural network for drowsy student state detection // Concurrency and Computation: Practice and Experience. 2018. V. 30(23). e4457.
10. *Jakubowski J., Chmielinska J.* Detection of driver fatigue symptoms using transfer learning // Bulletin of the Polish Academy of Sciences-technical Sciences. 2018. P. 869–874.
11. *Krizhevsky A., Sutskever I., Hinton G.E.* Imagenet classification with deep convolutional neural networks // Advances in neural information processing systems. 2012. P. 1097–1105.
12. *Girshick R., Donahue J., Darrell T., Malik J.* Rich feature hierarchies for accurate object detection and semantic segmentation // Computer vision and pattern recognition. 2014. P. 580–587.
13. *Girshick R.* Fast R-CNN // International conference on computer vision. Dec. 12. 2015. Santiago, Chile. P. 1440–1448.
14. *Ren S., He K., Girshick R., Sun J.* Faster R-CNN: towards real-time object detection with region proposal networks // Neural information processing systems. 2015. P. 91–99.
15. *Redmon J., Divvala S.K., Girshick R., Farhadi A.* You only look once: unified, real-time object detection // Computer vision and pattern recognition. 2016. P. 779–788.
16. *Redmon J., Farhadi A.* YOLO9000: better, faster, stronger // Computer vision and pattern recognition. 2017. P. 6517–6525.
17. *Redmon J., Farhadi A.* YOLOv3: An incremental improvement // arXiv: Computer vision and pattern recognition. 2018.
18. *Liu W., Anguelov D., Erhan D., Szegedy C., Reed S., Fu C., Berg A.C.* SSD: single shot multibox detector // European conference on computer vision. Oct. 8–16. 2016. Amsterdam. P. 21–37.
19. *Lin T., Goyal P., Girshick R., He K., Dollar P.* Focal loss for dense object detection // International conference on computer vision. Oct. 22–29. 2017. Venice. Italy. P. 2999–3007.
20. *Lin T., Dollar P., Girshick R., He K., Hariharan B., Belongie S.* Feature pyramid networks for object detection // Computer vision and pattern recognition. 2017. P. 936–944.
21. *He K., Zhang X., Ren S., Sun J.* Deep residual learning for image recognition // Computer vision and pattern recognition. 2016. P. 770–778.