IIETA International Information and Engineering Technology Association
Advancing the World of Information and Engineering

Traitement du Signal
Vol. 39, No. 2, April, 2022, pp. 677-682
Journal homepage: http://iieta.org/journals/ts

Recognition of Cheating Behaviors Based on Finetuning of Model Parameters

Fengyun Cao*, Shijie Lu, Jin Zhong

School of Computer Science and Technology, Hefei Normal University, Hefei 230061, China

Corresponding Author Email: caofengyun@hfnu.edu.cn

https://doi.org/10.18280/ts.390231

Received: 17 December 2021
Accepted: 10 February 2022

Keywords:
behavior recognition, transfer learning, parameter finetuning, pretrained network model

ABSTRACT

There are many problems with the current recognition methods of test cheating behaviors, namely, low accuracy, poor efficiency, and imbalance between positive and negative samples. To solve the problems, this paper proposes a classification and recognition method for test cheating behaviors through the transfer learning of pretrained models. Firstly, cheating samples, which mainly cover three cheating behaviors (peeking, passing notes, and checking cellphone) were collected from surveillance videos of exam rooms. The samples were enhanced through size transform and image synthesis. Next, multiple strategies were adopted to freeze the feature weights of the convolutional layers in the Darknet, before retraining the cheating classifier. In this way, a classification and recognition model was obtained for cheating behaviors. The model was tested on a self-designed dataset of test cheating behaviors. The results show that our method recognized 95.57% of cheating behaviors accurately, which is much better than the accuracy of the other methods. The real-time performance and accuracy of our method meet the application requirements.

1. INTRODUCTION

Subject tests are an important means to appraise the students' academic performance. They are widely adopted to measure the students' mastery of the knowledge of each subject. In academic performance measurement, the effectiveness of subject tests depends on the surveillance of the test room, and the honesty of the students. A well-disciplined and fair subject test gives all students equal opportunity, and ensures the integrity of academic assessment. If cheating goes unchecked, the students will become opportunistic, lazy, and dishonest, making test results less fair and credible. As an essential step in school education and teaching, the prevention of cheating plays an important role in enhancing the quality of students, and in the development of any teaching curriculum.

Invigilation, as a means of preventing predetermined cheating behaviors, is a tedious and time-consuming task. During the test, each invigilator needs to watch one person or a group of people. Cheating behaviors may arise, due to the neglection of the invigilator. To prevent cheating and ensure test fairness, the best solution is to videotape the entire test in real-time, process the real-time image data on computer, and classify and detect cheating behaviors. However, the traditional invigilation system cannot detect or prewarn cheating behaviors in real-time images, failing to realize intelligent, automatic detection of the test room. Luckily, computer vision and deep learning, two emerging techniques, have achieved immense success in image classification, target detection, and behavior recognition. Many cheating behavior recognition models have been developed based on deep learning, and applied to detect cheating behaviors in the test room intelligently.

In essence, the recognition of cheating behaviors in the test room is the recognition of human behaviors. The current recognition approaches of human behaviors rely on either the traditional manual feature expression or the automatic feature extraction by deep learning neural networks.

The traditional recognition methods for test cheating behaviors combine human target detection and classification models, based on manual feature expression. These methods commonly adopt feature expressions like inter-frame difference [1], histogram of oriented gradients (HOG) [2], feature background subtraction [3], mixed Gaussian modeling [4], and optical flow [5]. Then, the behaviors are classified and recognized by the support vector machine (SVM) classifiers [6] trained by feature expressions. Based on the statistical learning theory, the SVM classifiers have a low prediction accuracy, depend heavily on the data samples, and require the training samples and test samples to obey strictly the same distribution. Therefore, the traditional recognition methods cannot meet the application requirements in terms of accuracy and real-time performance, owing to the defects of manual feature expression, and SVM classifiers.

The automatic feature extraction and behavior recognition based on deep learning neural networks mainly fall into three categories: (1) the two-stage approaches based on region proposals, such as regional convolutional neural network (RCNN) [7], Fast RCNN [8], Faster RCNN [9], and region-based fully convolutional network (R-FCN) [10]; (2) regression-based one-stage approaches, represented by single shot multi-box detector (SSD) [11], you only look once (YOLOvX) [12-16] and VGG16(Visual Geometry Group)[17]; (3) The long short-term memory (LSTM) network for target detection in the overall video series [18-20].

(1) Two-stage approaches

RCNN and Fast RCNN consume a long time in the selective search and categorical regression of candidate regions. In Faster RCNN, the selective search is replaced with the regional growth algorithm, which significantly increases the search speed. But Faster RCNN does poorly in real-time performance. These three approaches are good at recognizing large objects. However, cheating behaviors are mostly hand, shoulder, and head movements. The three approaches are not accurate enough to detect these small objects.

(2) One-stage approaches

Taking VGG16 [17] as the basic convolutional network architecture, the SSD adds an auxiliary network layer for the fusion of the prediction results of multiscale convolutional graphs, and combines object pre-selection boxes called default boxes, which are similar to the anchor boxes in the Faster RCNN, to unify the different sizes of the input images. YOLOv3 is grounded on the Darknet53 with residual structure. Whereas the SSD receives a single-level basic network, YOLOv3 receives multi-level inputs, and surpasses the SSD in the detection accuracy of small objects.

The existing solutions lack public datasets, and are susceptible to the variation of many things during the test, namely, student posture, student appearance, student dress, and the environment. This paper enhances the original data through parameter finetuning of pretrained models, and CutMix [21], trying to improve the prediction ability of our model for new classes. The main contributions are as follows:

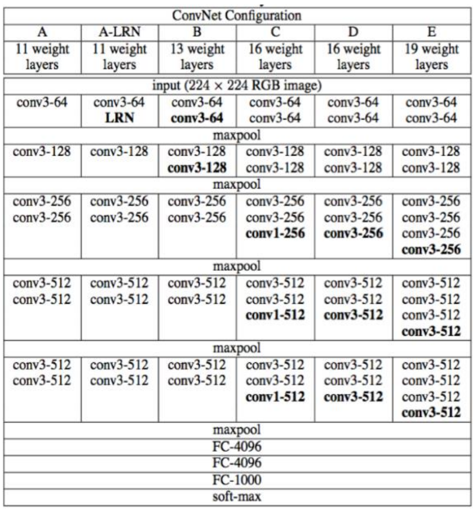
(1) Students were organized to attend a simulated test in the classroom. In such a highly unsupervised environment, the cheating behaviors during the test were collected, and prepared into a dataset of test cheating behaviors. Then, three cheating behaviors were labeled, namely, passing notes, peeking, and checking cellphone.

(2) Based on the pretrained YOLOv3, the feature parameters of shallow convolutional layers, which are relatively general and universal, were frozen, and the high-layer convolutional layers were retrained, to test the recognition rate of our model on each class of cheating behaviors.

(3) The images on the three classes of cheating behaviors, namely, passing notes, peaking, and checking cellphone, were tested separately and comprehensively, and contrastive experiments were carried out on VGG16 with multi-strategy parameter finetuning.

2. PRELIMINARIES

2.1 Structure of VGG16



ConvNet Configuration					
A	A-LRN	B	C	D	E
11 weight layers	11 weight layers	13 weight layers	16 weight layers	16 weight layers	19 weight layers
input (224 × 224 RGB image)					
conv3-64	conv3-64 **LRN**	conv3-64 **conv3-64**	conv3-64 conv3-64	conv3-64 conv3-64	conv3-64 conv3-64
maxpool					
conv3-128	conv3-128	conv3-128 **conv3-128**	conv3-128 conv3-128	conv3-128 conv3-128	conv3-128 conv3-128
maxpool					
conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256 **conv1-256**	conv3-256 conv3-256 **conv3-256**	conv3-256 conv3-256 conv3-256 **conv3-256**
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 **conv1-512**	conv3-512 conv3-512 **conv3-512**	conv3-512 conv3-512 conv3-512 **conv3-512**
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 **conv1-512**	conv3-512 conv3-512 **conv3-512**	conv3-512 conv3-512 conv3-512 **conv3-512**
maxpool					
FC-4096					
FC-4096					
FC-1000					
soft-max					

Figure 1. VGGNet with different configurations of weight layers

The CNN is mainly composed of convolutional layers, nonlinear units, pooling layers, and fully connected layers. In classification problems, the convolutional layers, nonlinear units, and pooling layers are responsible for extracting features, while the fully connected layers are responsible for classification.

In 2014, the Oxford Visual Geometry Group put forward the VGGNet, which replaces the large kernels in convolutional layers with multiple small kernels. After each convolution, the rectified linear unit (ReLU) is called for activation. To put it simply, each 11×11 large convolutional layer is substituted by multiple continuous 3×3 convolutional layers. The substitution can increase the depth of the neural network to a certain degree. In addition, the VGGNet has many nonlinear transforms, which reduce the computing load and enhance the extraction efficiency of the CNN for image features. Figure 1 shows the six different configurations of weight layers for the VGGNet [17].

2.2 Structure of YOLOv3

As a mainstream real-time target detection algorithm, the YOLO transforms target detection into a regression problem of locating bounding boxes and solving class probabilities. The YOLOv3 network model consists of Darknet53 for basic feature extraction, a multiscale feature fusion layer, and an output layer. Among them, Darknet53 is a fully convolutional network with a residual module containing 3×3 and 1×1 convolutional kernels (Figure 2).

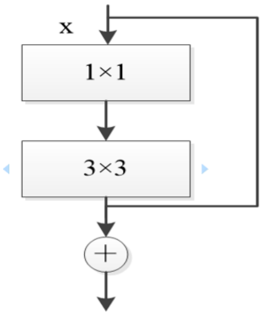
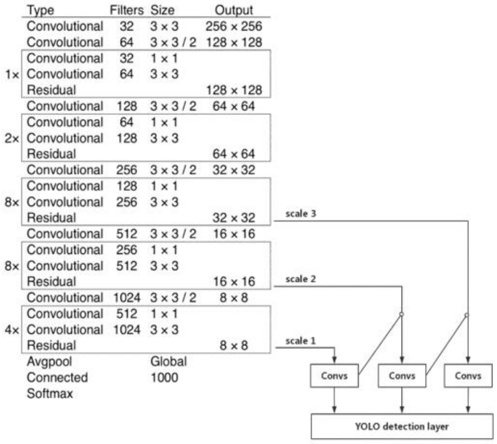


Figure 2. Residual module

The large, medium, and small targets are detected by feature maps of three different scales: 13×13, 26×26, and 52×52. The feature maps are then fused to obtain the final output. Figure 3 shows the structure of YOLOv3.



	Type	Filters	Size	Output
	Convolutional	32	3 × 3	256 × 256
	Convolutional	64	3 × 3 / 2	128 × 128
1×	Convolutional	32	1 × 1	
	Convolutional	64	3 × 3	
	Residual			128 × 128
	Convolutional	128	3 × 3 / 2	64 × 64
2×	Convolutional	64	1 × 1	
	Convolutional	128	3 × 3	
	Residual			64 × 64
	Convolutional	256	3 × 3 / 2	32 × 32
8×	Convolutional	128	1 × 1	
	Convolutional	256	3 × 3	
	Residual			32 × 32
	Convolutional	512	3 × 3 / 2	16 × 16
8×	Convolutional	256	1 × 1	
	Convolutional	512	3 × 3	
	Residual			16 × 16
	Convolutional	1024	3 × 3 / 2	8 × 8
4×	Convolutional	512	1 × 1	
	Convolutional	1024	3 × 3	
	Residual			8 × 8
	Avgpool		Global	
	Connected		1000	
	Softmax			

Figure 3. Structure of YOLOv3 [14]

2.3 Overview of transfer learning

It is costly and time-consuming to design and test deep neural networks, due to their massive scale and high complexity. The efficiency of model training can be enhanced easily and cheaply through transfer learning, especially in the presence of a few samples. Normally, the transfer learning of pretrained model can be divided into feature extraction and parameter finetuning. The specific type of transfer learning could be selected according to the sample size and properties in the application domain. Through parameter finetuning-based transfer learning, this paper freezes the weights of shallow layers, finetunes the deep network, and retrains the model. In this way, the pretrained model can be trained rapidly and achieve a high recognition rate of the dataset in a specific domain.

For a neural network trained by largescale multi-class samples, the features learned by the convolutional layers boast a high generality for different samples. These features are more obviously displayed in shallow layers. This is because the shallow convolutional layers mainly learn local subtle features, while the deep layers tend to capture the global or local target contours. The parameter finetuning mechanism effectively prevent the over-fitting problem, which arises from the high complexity of model parameters, a result of the small sample size.

3. RECOGNITION OF CHEATING BEHAVIORS

Deep learning has achieved remarkable progress in classification tasks. The deep structure of the network, huge computing power of hardware, and the massiveness of the available training data provide a tremendous driving force for the development of deep learning. The excellent performance of deep learning on small samples fully reveals the importance of data. The transfer mechanism improves the training speed and recognition rate of the pretrained model on the sample dataset of a specific domain by sharing the universal visual features and weight parameters of the shallow layers, providing a good solution to the deep learning of small samples. In this section, the proposed methodology is detailed in terms of the construction of cheating behavior dataset, sample enhancement, and parameter finetuning mechanism, followed by a deep discussion on the application of transfer learning to cheating behavior recognition.

3.1 Types of cheating behaviors and sample construction

Transfer learning, as a machine learning method, can be viewed as reusing a model developed for a task in the second task. To solve the second task, it is not necessary to build a brand-new model, i.e., adopt a pretrained model of similar problems. Instead, a model pretrained on the other problems can be taken as the starting point. Transfer learning uses a network architecture with preloaded weights. The classifier is modified before retraining part or all of the model, in order to output the prediction of the new task.

There is no open-source dataset available for cheating behaviors in the test room. To build a training set of cheating behaviors, it is necessary to collect a huge sum of image samples. Ideally, the collected images should be clear and sharp, eliminating the need for preprocessing (e.g., denoising). Due to lighting conditions and other conditions, the collected data may face problems like vague features and strong interference (e.g., heavy occlusions between students, and between students and desks). The effective preparation of sample data is crucial to deep learning.

In the actual scene of the test room, this paper selects four types of test room behaviors: passing notes, peeking, checking cellphone, and non-cheating. The final dataset was obtained through data preprocessing, labeling, and image enhancement. There are 6,000 cheating images and 2,000 non-cheating images in the dataset. Two labels were assigned: cheating and non-cheating. The cheating images include extensive data on passing notes (2,000 images), peeking (2,000 images), and checking cellphone (2,000 images). The non-cheating images are about non-cheating behavior. Figure 4 shows some of the samples.

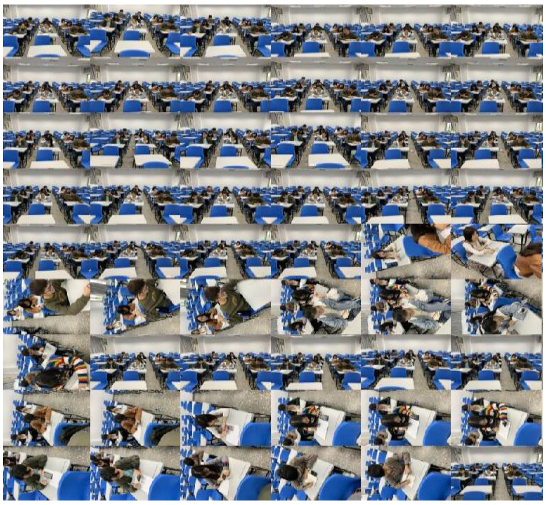


Figure 4. Some of the samples

3.2 Data preprocessing

Owing to the diversity of data sources, the collected dataset contains images of different resolutions. To train the network with the dataset, the size of each input image must match the input size of the network model. According to the data situation and classification goal, the sample data were preprocessed, and enhanced. The preprocessing greatly improves the prediction accuracy of the algorithm.

3.2.1 Random geometric transform

The preprocessing steps include size adjustment, random rotation, random cropping, random translation, and color transform. Because the operation is random, the data generated in each round are unique. For example, the same image may be flipped in some rounds, and not flipped in the other rounds. Thus, the training data vary with the rounds, producing enhanced samples.

3.2.2 CutMix sample enhancement

CutMix generates a new training sample $(\tilde{x},\tilde{y})$ by combining two training samples (x_a,y_a) and (x_b,y_b). The new sample is used to train the network model with the original loss function. The sample combination operation can be defined as:

$$\begin{gathered}\tilde{x}=M\odot x_a+(1-M)\odot x_b\\ \tilde{y}=\lambda y_a+(1-\lambda)y_b\end{gathered}\tag{1}$$

where, $M \in \{0,1\}^{W \times H}$ is the label mask for the cropped and reserved areas of the image (1 for the cropped area; 0 for the other areas); $\odot$ is the pixel-wise operation; λ belongs to $Beta(\partial, \partial)$. Since the value of ∂ is usually set to 1 in experiments, λ is uniformly distributed in (0, 1).

3.3 Training process of transfer learning

In the case of insufficient sample data, the transfer learning of pretrained models is a convenient and efficient method. If the pretrained model is trained on large multi-scale image datasets like ImageNet, the features learned by the convolutional layers boast a high generality for different samples. As mentioned before, these features are more obviously displayed in shallow layers. This is because the shallow convolutional layers mainly learn local subtle features, while the deep layers tend to capture the global or local target contours. To speed up the training speed of the pretrained model on a dataset of the specific domain, a viable option is to freeze the parameter weights of the convolutional layers in the pretrained model, and only retrain the weights of the fully connected layer.

This paper compares the multi-strategy finetuning of weight parameters of two pretrained models: VGG16 and YOLOv3. The finetuning strategies mainly include freezing shallow layer weights, re-learning weights, and freezing all weights except the fully connected layer.

(1) Retraining weight parameters

The classifier of the pretrained model is replaced to form a complete network model for retraining. The new model only differs from the pretrained model in the type of the classifier. This time-consuming strategy is only suitable, when the source domain varies significantly from the target domain.

(2) Freezing the weight parameters of shallow layers

The weight parameters of shallow convolutional layers are frozen, and the remaining learnable layers are retrained. During model training, the weight parameters of the frozen layers are not updated. Thus, the network training will converge much faster, and the overfitting on small samples will be solved.

(3) Freezing all weight parameters except the fully connected layer

The fully connected layer is the last learnable layer in most network models. By this strategy, the fully connected layer is replaced, and its weight parameters are frozen. The new fully connected layer will output a class parameter about the class of the target dataset. In this paper, the 1,000 classes of ImageNet are revised into 4 classes, and the pretrained model is retrained on the cheating behavior dataset. The expansion of more classes of cheating behaviors can be realized through further iteration. It is only necessary to change the corresponding classes.

3.4 Classification of cheating behaviors based on transfer learning

Figure 5 shows the flow of the training framework for the classification model. There are two parts of the framework: model training and classification model deployment. During model training, the pretrained model is loaded, the parameter finetuning strategy is selected, retraining is conducted on samples, and the model accuracy is predicted and evaluated on the test set. During the deployment of the classification model, the trained classification model is adopted to predict the classes of the input images, and output the predictions. The network model can be iteratively optimized according to the classification samples.

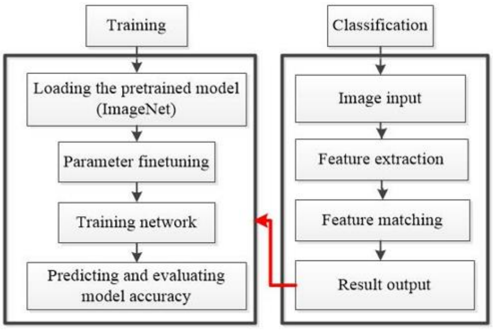


Figure 5. Training framework

4. EXPERIMENTS AND RESULTS ANALYSIS

4.1 Experimental setting

Two pretrained network models were compared in our experiments, namely, VGG16 and YOLOv3. Each model was subjected to parameter finetuning and training on a workstation with NVIDIA 2080Ti GPU and Intel(R) Core(TM)i7-9700 CPU @ 3.00GHz 3.00GHz. Then, the parameter finetuning and training under CutMix data enhancement was added. The models were realized on Matlab R2020a. The classes of the self-developed dataset are shown in Table 1.

Table 1. Cheating behaviors and their classes

Cheating/Non-cheating	Class	Size of verification set	Size of test set
Cheating	Passing notes	1400	600
Cheating	Peeking	1400	600
Cheating	Checking cellphone	1400	600
Non-cheating	Normal	1400	600

4.2 Results analysis

With the initial learning rate of 0.0001, the stochastic gradient descent with momentum (SGDM) was chosen as the network optimizer. In the presence or absence of sample data enhancement, VGG16_1 (freezing the weight parameters of all layers except the fully connected layer), YOLOv3_1 (freezing the weight parameters of all layers except the fully connected layer), and VGG16_2 (freezing the weight parameters of shallow convolutional layers) were applied on each class of samples. Table 2 compares the recognition accuracies of these models.

Table 2. Comparison of parameter finetuning results

Model	Single-class accuracy	Multi-class accuracy
VGG16_1	91.30%	83.97%
VGG16_1+Cutmix	94.2%	92.1%
Yolov3_1	95.15%	89.6%
Yolov3_1+CutMix	97.3%	93.81%
VGG16_2	92.34%	87.57%
VGG16_2+CutMix	95.34%	92.67%

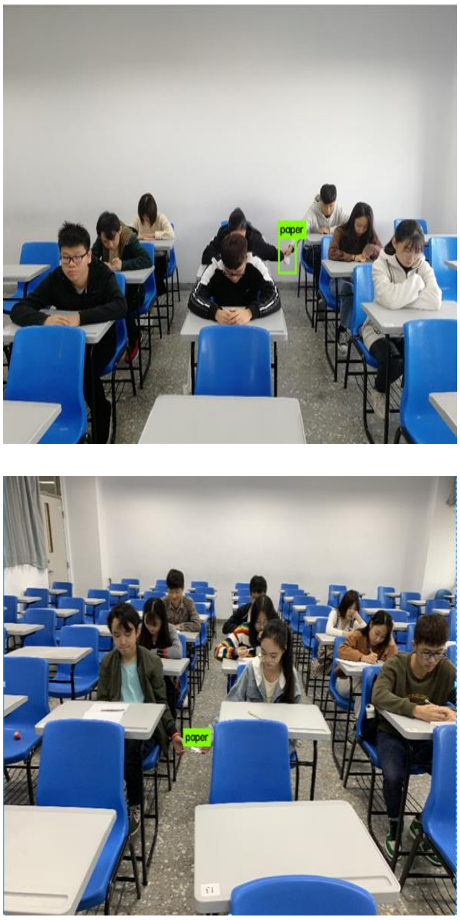


Figure 6. Prediction results on a single class

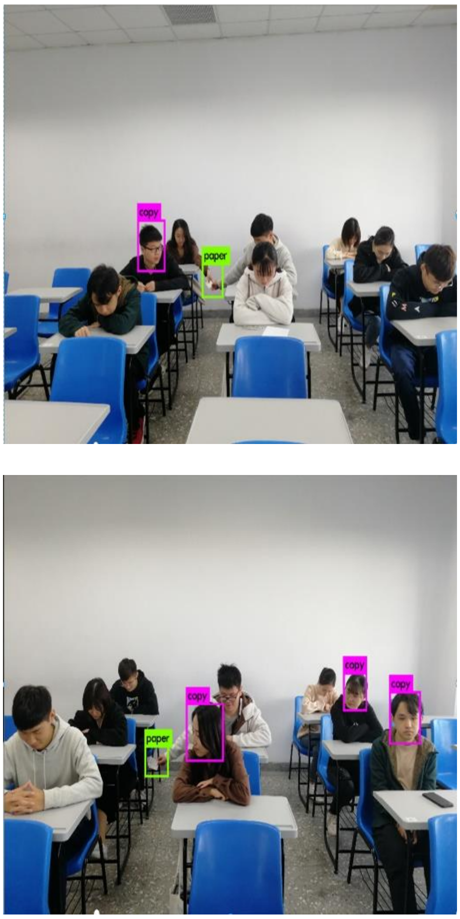


Figure 7. Prediction results on multiple classes

Figures 6 and 7 show the prediction results on a single class and multiple classes, respectively. Figure 8 illustrates the training process of YOLOv3_1.

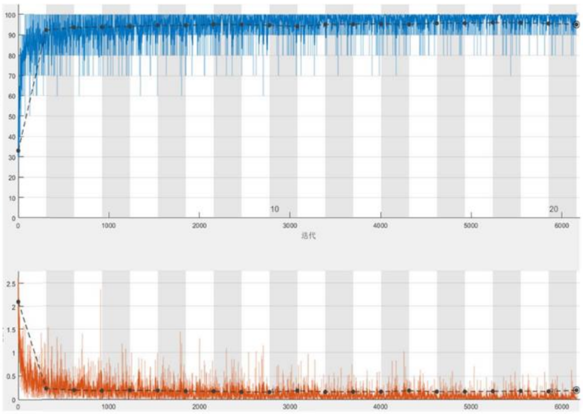


Figure 8. Training results of YOLOv3_1

5. CONCLUSIONS

Under the framework of parameter finetuning of deep neural networks, this paper presents an automatic detection and characterization method for test room cheating behaviors. Firstly, a dataset was developed by the research team, and assigned labels of cheating and non-cheating. Based on the pretrained model, further training was conducted with different strategies for parameter finetuning. Then, a cheating behavior recognition model was designed on the basis of model parameter finetuning. Finally, experiments were carried out to compare the multiple transfer strategies on sample data enhancement. The experimental results show that the proposed model has a good generalizability, and correctly recognizes over 92% of the classes of the data in the target domain. Thus, our model is very suitable for recognizing the targets, i.e., the cheating behaviors in the test room.

ACKNOWLEDGMENT

This paper is funded by Anhui University collaborative innovation project (Grant No.: GXXT-2021-091). School level key scientific research projects of Hefei Normal University (Grant No.: 2021KJZD12).

REFERENCES

[1] Guo, J., Wang, J., Bai, R., Zhang, Y., Li, Y. (2017). A new moving object detection method based on frame-difference and background subtraction. In IOP Conference Series: Materials Science and Engineering, 242(1): 012115. https://doi.org/10.1088/1757-899X/242/1/012115

[2] Chen, T., Gao, T., Li, S., Zhang, X., Cao, J., Yao, D., Li, Y. (2021). A novel face recognition method based on fusion of LBP and HOG. IET Image Processing, 15(14): 3559-3572. https://doi.org/10.1049/ipr2.12192

[3] Haifei, S., Xingliu, H., Zhen, S. (2020). Fall recognition method based on background subtraction and feature extraction. Journal of Electronic Measurement and Instrumentation, 34(10): 33-39.

https://doi.org/10.13382/j.jemi.B2003151

[4] Anand, V., Pushp, D., Raj, R., Das, K. (2019). Gaussian Mixture Model (GMM) Based object detection and tracking using dynamic patch estimation. In 2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 4474-4481. https://doi.org/10.1109/IROS40897.2019.8968275

[5] Zhao, B., Huang, Y., Wei, H., Hu, X. (2021). Ego-motion estimation using recurrent convolutional neural networks through optical flow learning. Electronics, 10(3): 222. https://doi.org/10.3390/electronics10030222

[6] Zhang, G.L., Jia, S.M., Zhang, X.Y., XU, T. (2017). Action recognition based on adaptive mutation particle swarm optimization for SVM. Opt. Precis. Eng, 25(6): 1669-1678. https://doi.org/10.3788/OPE.20172506.1669

[7] Girshick, R., Donahue, J., Darrell, T., Malik, J. (2014). Rich feature hierarchies for accurate object detection and semantic segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 580-587. https://doi.org/10.1109/CVPR.2014.81

[8] Girshick, R. (2015). Fast R-CNN. In Proceedings of the IEEE International Conference on Computer Vision, pp. 1440-1448. https://doi.org/10.1109/ICCV.2015.169

[9] Ren, S., He, K., Girshick, R., Sun, J. (2017). Faster R-CNN: Towards real-time object detection with region proposal networks. IEEE Transactions on Pattern Analysis & Machine Intelligence. 39(6): 1137-1149. https://doi.org/10.1109/TPAMI.2016.2577031

[10] Dai, J., Li, Y., He, K., Sun, J. (2016). R-FCN: Object detection via region-based fully convolutional networks. Proceedings of the 30th International Conference on Neural Information Processing Systems, pp. 379-387. https://doi.org/10.48550/arXiv.1605.06409

[11] Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.Y., Berg, A.C. (2016). Ssd: Single shot multibox detector. In European Conference on Computer Vision, pp. 21-37. https://doi.org/10.1007/978-3-319-46448-0_2

[12] Redmon, J., Divvala, S., Girshick, R., Farhadi, A. (2016). You only look once: Unified, real-time object detection. In Proceedings of the IEEE Conference on Computer Vision And Pattern Recognition, pp. 779-788. https://doi.org/10.1109/CVPR.2016.91

[13] Redmon, J., Farhadi, A. (2017). YOLO9000: Better, faster, stronger. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 7263-7271. https://doi.org/10.1109/CVPR.2017.690

[14] Redmon, J., Farhadi, A. (2018). Yolov3: An incremental improvement. arXiv preprint arXiv:1804.02767. https://doi.org/10.48550/arxiv.1804.02767

[15] Wang, C.Y., Bochkovskiy, A., Liao, H.Y.M. (2021). Scaled-yolov4: Scaling cross stage partial network. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 13029-13038. https://doi.org/10.1109/CVPR46437.2021.01283

[16] Feng, T.Y., Zhang, Y.H., Zhang, K., Fei, M.R., Xu, S. (2021). Research on multi-person detection algorithm in classroom in complex environment. Journal of Electronic Measurement and Instrumentation, 35(6): 53-62. https://doi.org/10.13382/j.jemi.B2003594

[17] Simonyan, K., Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv: 1409.1556. https://doi.org/10.48550/arXiv.1409.1556

[18] Sarabu, A., Santra, A.K. (2021). Human action recognition in videos using convolution long short-term memory network with spatio-temporal networks. Emerging Science Journal, 5(1): 25-33. https://doi.org/10.28991/esj-2021-01254

[19] Sarabu, A., Santra, A.K. (2020). Distinct two-stream convolutional networks for human action recognition in videos using segment-based temporal modeling. Data, 5(4): 104. https://doi.org/10.3390/data5040104

[20] Jia, J.G., Zhou, Y.F., Hao, X.W., Li, F., Desrosiers, C., Zhang, C.M. (2020). Two-stream temporal convolutional networks for skeleton-based human action recognition. Journal of Computer Science and Technology, 35(3): 538-550. https://doi.org/10.1007/s11390-020-0405-6

[21] Yun, S., Han, D., Oh, S.J., Chun, S., Choe, J., Yoo, Y. (2019). Cutmix: Regularization strategy to train strong classifiers with localizable features. In Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 6023-6032. https://doi.org/10.3390/electronics10131601