Journal of Network Intelligence ©2023 ISSN 2414-8105 (Online)
Taiwan Ubiquitous Information Volume **8**, Number **2**, May **2023**

Real-Time Multiple Face Identity Recognition Based on Face-tracking Feature Extraction

Yi-Tong Jiang

School of Aeronautics and Astronautics
Zhejiang University
No. 38, Zheda Road, Hangzhou 310027, P. R. China
22024070@zju.edu.cn

Zhe-Ming Lu*

Ningbo Innovation Center,
Zhejiang University
No. 5, Xuefu Road, Hangzhou 315000, P. R. China
zheminglu@zju.edu.cn

Zhe Wang

School of Aeronautics and Astronautics
Zhejiang University
No. 38, Zheda Road, Hangzhou 310027, P. R. China
wangzhe0703@zju.edu.cn

Guan-Zhong Tian

Ningbo Innovation Center,
Zhejiang University
No. 5, Xuefu Road, Hangzhou 315000, P. R. China
gztian@zju.edu.cn

*Corresponding author: Zhe-Ming Lu

Received April 5, 2022, revised June 11, 2022, accepted August 4, 2022.

Abstract. *Face recognition technology is an important branch of object detection, it is actually a general term which includes face detection, face recognition or face identification. In industrial applications, such as the access control system or check-in system, usually use face identification technology to determine the face ID. Face identification needs to rely on basic face detection and recognition technology, although it is widely studied, limited by these two aspects, its function relatively single, for example, it often only supports the identification of a single face or short-range and static detection. Therefore, based on Multi-task convolutional neural network and Inception-resnet-v1, this paper proposes a method named Face-tracking Feature Extraction, which use tracking algorithm instead of partial feature extraction and can realize the function of real-time multi-target face identification with higher speed and lower resource occupation than before. The feasibility of this method is confirmed by the comparison between before and after the experiment.*

Keywords: Multi-target face recognition, tracking, Multi-task convolutional neural network, Inception-resnet-V1, Face-tracking Feature Extraction

1. **Introduction.**

In recent years, face recognition technology based on deep learning has made great achievements. Face recognition technology is actually a relatively broad term, which includes face detection, face feature extraction and classification, face identification and other technologies. Different scholars have proposed different methods to better realize face detection and face feature extraction. The shape and expression of human face are different, and the corresponding image feature information is also different, so the primary goal of face recognition is to locate all kinds of faces, maybe rely on facial expression, key points and the other things[1,2,3]. Then, At the stage of training, face related visual tasks often rely on an extremely large number of data sets, and often involve privacy issues, how to reduce the training samples has become a hot research point[4]. The feature extraction network of human face is used to extract the feature vector of human face, which can be used for classification or identification[5,6].

We can see that most scholars still focus on changing the quality of a single face, the points targeted are also often in the accuracy of detection and facial feature vector, do not pay attention to the speed of multi face collaborative processing. Actually, this will bring great problems. It only aims at the improvement of the quality of a single face. Although there will be no problem with the accuracy of face recognition, the algorithm is still difficult to apply to multi-objective situations. The most direct consequence is that the processing speed is very slow and there is no way to apply it. Therefore, it is very important to improve the speed of the algorithm. For improving the speed of the algorithm, there are various methods. You can choose to simplify or optimize the structure of the algorithm itself, or choose the method of combining the algorithm with the other algorithm. The structure of optimization algorithm is a complex process. It has high uncertainty, and its robustness also needs the support of a large number of experiments. Considering this problem, we choose to focus on the combination of algorithms. Based on face detection and recognition, using the principle of face tracking, this paper discusses a method of using face tracking to replace the face feature extraction of some frames, so as to improve the number of frames of multi-target face real-time recognition.

In short, our methods and contribution are mainly:

(1)Using the two-stage face detection and recognition method, Multi-task convolutional neural network(MTCNN) is used to detect the face, and Inception-resnet-v1 is used for feature extraction. MTCNN can accurately extract the face information of various sizes, while Inception-resnet-v1 has a good effect on face feature extraction. These two reliable methods provide great help for our follow-up work. Of course, we also use some data to fine tune the two networks and modify some parameters, which can better adapt to the follow-up work.

(2)We propose an improved method for the two-stage face detection and recognition algorithm, namely face tracking feature extraction. This method applies the face tracking method to the detection and recognition process, improves the real-time frame rate of the algorithm without affecting the accuracy of the algorithm, and makes up for the deficiency of the two-stage network speed

(3)In this paper, we propose an idea, that is, how to optimize the repeated feature extraction process in real-time system. We know that the object recognized (face in this paper) in the real-time system does not need repeated feature extraction, which not only wastes resources but also affects the speed. For the obtained features, we can use some other algorithms, such as tracking algorithm, to retain the feature information in real time. For the recognized face, we can use the tracking method to lock the position of the face, identify the identity, and decide to continue tracking or feature extraction. The tracking algorithm is inserted between feature extraction to replace the repeated feature extraction of some frames, which can improve the running speed.

(4)In addition to the innovation of methods and ideas mentioned above, this paper also makes a certain contribution to industrial application. Industry often has a very high demand for multi face identification. Many occasions involving the passage of large traffic people, such as sidewalk monitoring, runway personnel detection, illegal capture of personnel in vehicles, etc. These occasions involve a large number of objectives. If the previous algorithms are not optimized and adjusted, it is very difficult to achieve the goal of real-time detection. The methods discussed in this paper also hope to solve the problems of industrial application.

2. **Related Works.**

2.1. **Face detection.**

This process is similar to the principle of ordinary object detection, so most object detection algorithms often have good results in detecting faces. Single Shot MultiBox Detector(SSD)[7]proposed in 2016 is one of the popular detection algorithms in recent years. As a one-stage detection algorithm, SSD algorithm has the greatest advantage of being very fast and easy to train, which is suitable for the actual needs of face detection. Some scholars have also changed it into a lightweight SSD algorithm revision, that is, the Mobilenet SSD[8]. Because of its lightweight characteristics, mobilenet SSD is more suitable for the needs of embedded systems such as mobile terminals. J. Redmon et al. proposed the YOLO algorithm, YOLO means you only look once. The earliest YOLO algorithm[9] left other detection algorithms far behind at that time, even though YOLO algorithm is not widely studied in academia, it plays an important role in industry. So that it has attracted all kinds of subsequent improvements on YOLO algorithm. With a variety of new versions of YOLO algorithm[10,11,12,13]gradually proposed, YOLO algorithm also began to be used in face recognition tasks[14].

This paper focuses on the role of tracking in identity recognition. Compared with one-stage algorithms such as SSD and YOLO, the two-stage algorithm that distinguishes the process of detection and classification can better intersperse tracking experiments. Therefore, this paper selects a single detection network to realize it. Multi-task Cascaded Convolutional Networks is a face detection algorithm was proposed by K. Zhang et al. in 2016, we usually call it MTCNN[15]. It is divided into three cascaded networks,each network adopts the idea of candidate box and classifier. In this paper, we use MTCNN to detect and local faces. We also adjusted some parameters of MTCNN which weakened the applicability of the algorithm for some small face recognition, because some small faces are harmful to identity recognition.

2.2. **Deep Convolutional Neural Networks for face.**

Compared with ordinary convolutional neural network, deep convolutional neural network, as its name implies, has deeper levels and much larger parameters. In the classification task, this network will also be more accurate than before. The principle of individual face recognition (without identity recognition) is basically the same as that of target detection and recognition. It can directly train the network to detect and recognize faces by relying on the supervised training data set. The faces here can be regarded as a class, so this task can also be called face classification.

Different from simple face classification, face recognition does not need to obtain the final classification results, but to obtain the embedding vector extracted by CNN. Therefore, for the detected face, we need to find an excellent feature extraction network to obtain the embedding vector of the face.The performance of deep convolution neural network is obviously better than the traditional convolution neural network, so it is more competent to extract face embedding vectors. VGG[16]is a better classification network which often appears as the backbone layer in the network of many other visual tasks.It won the second place in the 2014 ILSVRC. The residual structure Resnet [17] proposed by K. He et al. in 2015 has improved the accuracy of classification network to a new level and won the first place in ILSVRC & CoCo 2015.The combination of inception architecture[6] and residual connection may improve the performance of feature extraction network[18]. Experiments show that the residual connection significantly accelerates the training of perception network. Inception-resnet-v1 is actually obtained by adding the residual module to Inception v3. In this paper, we use the trained model for feature extraction.

2.3. **Object Tracking.** For the face that has appeared in the picture, if its identity has been determined, it is a waste of resources to continue the continuous feature extraction and embedding vector similarity calculation of its identity. Therefore, if the algorithm can screen the recognized face, cancel the feature extraction process and replace it with tracking, It can greatly improve the frame rate of video, and has better adaptability to multi face targets.

The tracking algorithm often adopts the centroid tracking algorithm or calculates the IOU coincidence degree to realize the tracking function.The centroid tracking algorithm is introduced to achieve the purpose of face tracking. However, due to the need to take into account the feature extraction of the face and confirm the identity of the person at the same time, this paper proposes a method called Face-tracking Feature Extraction which uses both the tracking algorithm and the feature extraction algorithm. It can not only track the known face, but also introduce the tracking function to reduce the computation of the computer, Make real-time performance higher. Experiments show that this method can improve the number of frames while recognizing face.

3. **Methods.** The algorithm of face ID recognition in this paper is shown in Fig 1. The detect module is the first part of the algorithm. We use MTCNN algorithm to detect the face in this frame of image collected by the camera. In the feature extraction module, the detection result of the original image obtained by MTCNN is sent to the feature extraction network Inception-resnet-v1, and finally an embedding vector will be obtained. This embedding vector can be compared with the embedding vectors in the database to determine the ID of the face. The two modules of feature extraction and identification are conventional face identification processes. In order to improve the speed and efficiency in practical use, the identified face can record the face position through face tracking method, and avoid unnecessary feature extraction and feature comparison in the subsequent processing process.

3.1. **MTCNN.** We use MTCNN to detect faces in images.The structure of MTCNN algorithm is shown in Fig.2. After resizing the original image several times, an image pyramid will be generated. The function of the image pyramid is to ensure that the subsequent network can find the face with the corresponding recognition size. The full name of p-net network is proposed network. The network contains three convolution layers and uses the idea of classifier to judge whether the area is a face. If it is a face, it will continue to be corrected by regression coefficient. P-net will finally generate a series of candidate boxes. R-Net (refine network) resizes all boxes output by p-net, and then, similar to p-net, it will continue to filter and correct candidate boxes. After R-Net, the number of candidate boxes will be reduced. O-net (output network) will adopt more supervised methods to recognize faces. In this network, the facial feature points of the face will be returned and output, and the network will eventually output a detection image containing face frame and face key points.

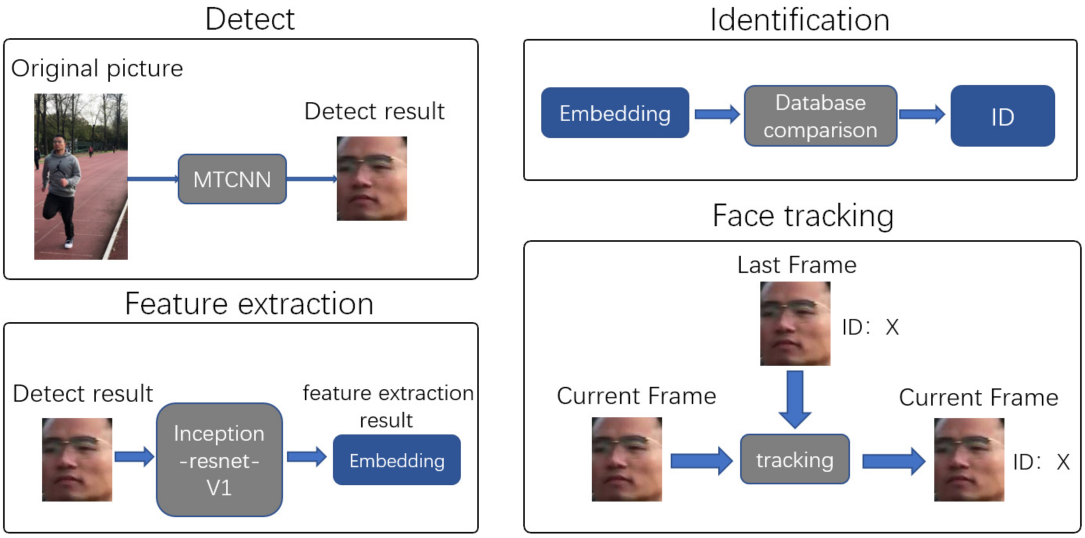


FIGURE 1. Algorithm structure

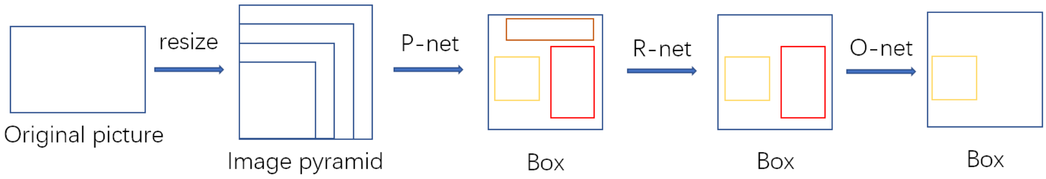


FIGURE 2. MTCNN structure

3.2. **Inception-resnet-V1.** The structure of Inception-resnet-v1 network used in this paper is shown in Fig.3. The left side is the overall structure of Inception-resnet-v1 network, and the right side introduces the internal structure of stem module and the internal structure of normalize module in detail. The input we use is a 160 × 160 × 3 tensor vector, which is generated by MTCNN. The stem module contains several 3 × 3 convolution layers and one 1 × 1 convolution layer. We will finally get an output with a length of 512. This output is the embedding vector we need for comparison with the face embedding vectors of the database.

3.3. **Face Identification.** The process of face recognition is based on the detection of MTCNN and the generation of face embedding vector by Inception-resnet-v1. The similarity between the currently obtained face embedding vector and the database face embedding vectors is calculated. This similarity is often expressed by the Euclidean distance $d(a, b)$ of the embedding vector. The smaller the value of $d(a, b)$, the higher the similarity between the two faces. Therefore, the interference items can be filtered by setting the threshold. If the two faces are determined to be the same person, then this $d(a, b)$ must be far less than other distances.

$$d(a,b) = \sqrt{(a_1 - b_1)^2 + (a_2 - b_2)^2 + \cdots + (a_n - b_n)^2} \tag{1}$$

3.4. **Our Face-tracking Feature Extraction.** The tracking algorithm often adopts the centroid tracking algorithm or calculates the IOU coincidence degree to realize the tracking function.We use centroid tracking algorithm to realize the basic tracking function.The principle of the algorithm is shown in the figure. The last frame and this frame represent the faces collected in the current frame and the previous frame respectively. We calculate

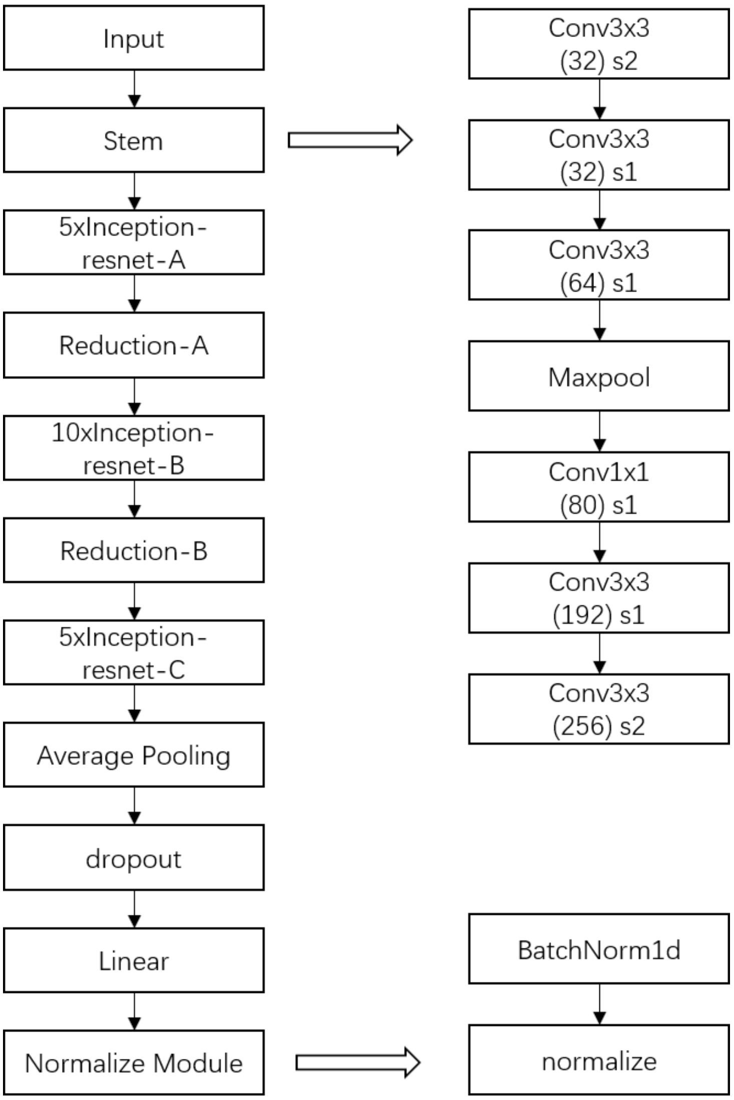


Figure 3. Inception-resnet-v1 structure

the distance $L(c_1, c_2)$ between the two faces. If this $L(c_1, c_2)$ is less than a threshold set by us, we can think that the two faces are actually the same face. The distance $L(c_1, c_2)$ can be expressed by the centroid distance of two faces. The calculation method of centroid distance is shown in Fig.4

$$L(c_1, c_2) = \sqrt{(x_1 - y_1)^2 + (x_2 - y_2)^2} \tag{2}$$

Fig.5 shows the process of Face-tracking Feature Extraction. For the current frame, after the face is extracted by using MTCNN detection algorithm, the ID of the face in the current frame is unknown. Instead of directly sending it to the Inception-resnet-v1 network for feature extraction, we directly perform face tracking operation on it. Because the face identity information in the previous frame is confirmed, We can confirm whether the ID of the face in the current frame exists by tracking. If so, we can directly omit the process of feature extraction and output its ID in the current frame to reduce the amount of calculation; If the tracking result determines that the ID does not exist, the normal process is used to extract the feature of a single face and obtain its ID. By the way, if there are many faces in the picture, our algorithm can also judge and track the ID of each

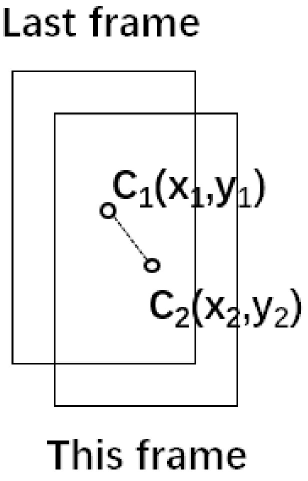


Figure 4. Centroid tracking

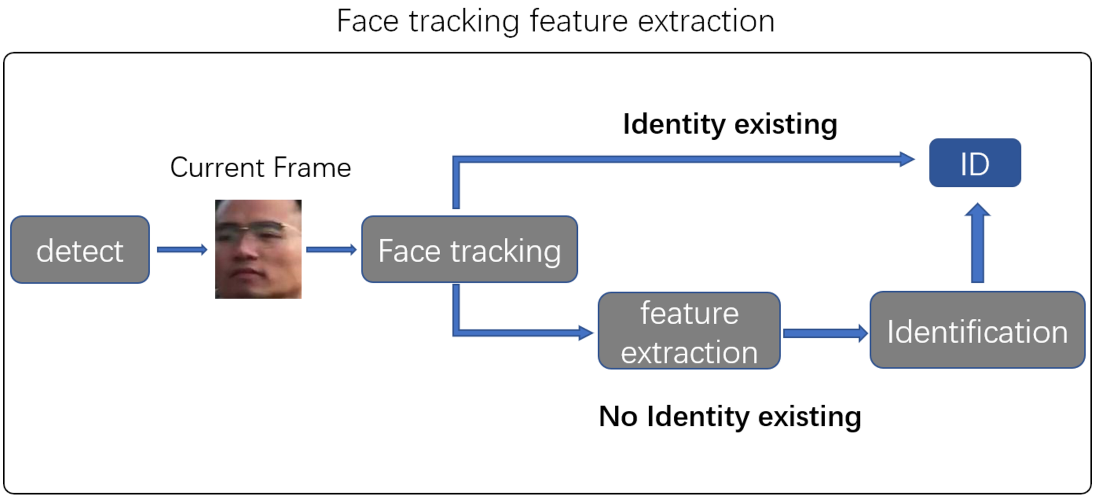


Figure 5. Face-tracking Feature Extraction

face separately. In any case, it can ensure that if the ID exists, the feature extraction process will be cancelled.

4. **Experiments.** This part introduces the experimental results of our data set, model accuracy test and algorithm comparison.We use the tensorflow model used in Schroff et al.'s paper[19]. There are two latest models, one is trained on vggface2, and the other is trained on CASIA-webface. For our convenience, we transformed it into a pytorch model and verified its accuracy again on the LFW data set.I would like to thank the relevant work of this paper.

We compared the Face-tracking Feature Extraction with the previous algorithm, the difference is whether tracing is introduced. All the experiments are tested and run on a desktop equipped with i7-9700k CPU, 16GB ram and NVIDIA rtx3060 GPU.The environment used by the algorithm is Python 3.8,torch1.9.0 and cuda11.1.

4.1. **Datasets.** We use the existing trained MTCNN weight files and the weight files of Inception-resnet-v1 trained and fine tuned on vggface2 data set, CASIA-webface data set.In order to test the performance of the model used, we also collected the LFW data set and tested the accuracy of the model on the LFW data set.

We have prepared videos for the algorithm test. The resolution of these videos is 864×480 or 480×864, and the content of these videos contains several faces. There may be a single face or multiple faces in the picture of the same frame. Use this dataset to test the frame rate of Face-tracking Feature Extraction in actual use, and make the average frame rate of the whole video, compare with the results without Face-tracking Feature Extraction.

Table 1. Dataset

Dataset	Data Number	ID Number	Availability	Purpose
vggface2	3.31M	9131	*Public*	*Pretrained*
CASIA webface	494414	10,575	*Public*	*Pretrained*
LFW	13233	5,749	*Public*	*Testing*
Test video	33	—	*Private*	*Testing*

Table.1 shows the data sets we use or use indirectly. Data number refers to the total amount of the data set. ID number refers to the number of different identity names contained in the data set. Availability indicates that the data set is the source of the data set, and purpose indicates the role of the data set in this article.

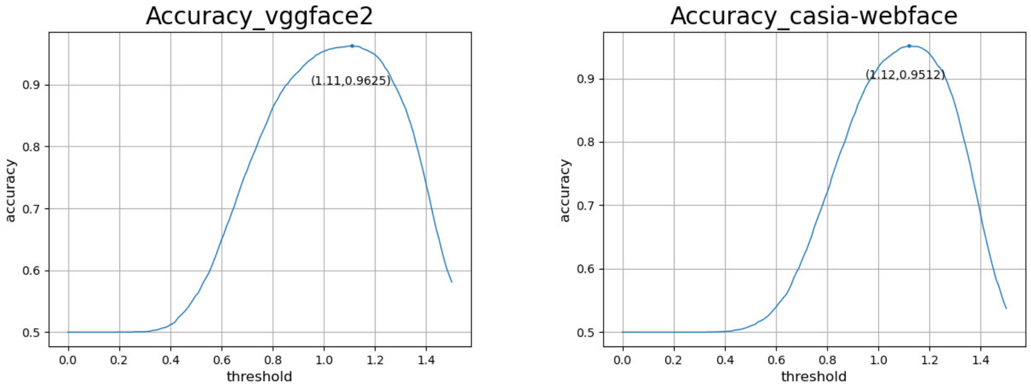
4.2. **Models test on LFW.** In order to better complete our experiment, we selected the model with better known effect for our experimental work. However, since most of the earliest face recognition models were run in the environment of the early old version of tensorflow, in order to facilitate our experiment, we transformed the model into PTH format that pytorch can use. We tested the accuracy of the two pytorch model files on the LFW data set, and drew a graph of accuracy and threshold. In Fig.6, (A) represents the model in the vggface2 data set, and (B) represents the model in the CASIA webface data set.For the minimum accuracy of threshold, we set it to 0.01.In order to explain more concisely, we use model A and model B to represent the two models.

We put the two pictures together to better explain the meaning of the chart. In Fig.7, the blue line represents the threshold accuracy curve of model a, and the red line represents the threshold accuracy curve of model B. We can see that when the threshold is 1.11, the accuracy of model a reaches the maximum value of 0.9625, and for model B, when the threshold is 1.12, the accuracy is the highest value of 0.9512.

By comparing the data with the line chart, we can see that the accuracy of model A will be slightly higher than that of model B. therefore, in the subsequent algorithm comparison experiment, we will use model A for related work.

4.3. **Algorithm comparison.** We tested the accuracy of the proposed Face-tracking Feature Extraction algorithm. The essence of the Face-tracking Feature Extraction algorithm is to add the tracking function to the original algorithm, so it will not affect the accuracy in theory. However, we still tested the accuracy of the algorithm on the LFW data set and compared it with the original algorithm. It can be seen from the Table.2, when the threshold is 1.11, the accuracy of the algorithm has not changed before and after improvement.

Of course, the purpose of our algorithm is to improve the frame rate of real-time detection and recognition, but the video data set for face recognition is not common, so we collected some video files for the comparison test of the frame rate of the algorithm.



(A) vggface2

(B) Casia-webface

Figure 6. These two figures respectively correspond to the test results of the two models on LFW data set. The data set of LFW test contains 3000 pairs of faces with the same identity and 3000 pairs of faces with different identities. The ordinate accuracy of the two images represents the accuracy of face recognition by the model, and the maximum value is 1. The abscissa threshold represents the set threshold. The lower the threshold, the more strict the model is for face recognition, and vice versa. The distance obtained by Equation.1 will be compared with the threshold to obtain the judgment results of the model, all results will be compared with the correct answers to finally obtain the accuracy. The Fig.7 shows the comparison of the accuracy of the two models

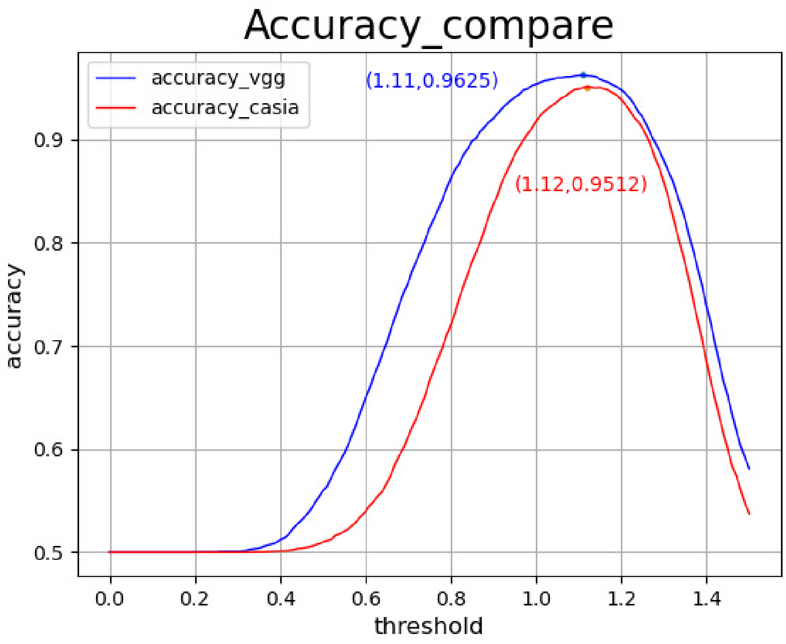


Figure 7. Accuracy compare

The Fig.8 and Table.3 are examples of partial frames of a test video.The Fig.8 shows the movement process of a single person for 9 consecutive frames, there is no undetected or

Table 2. Algorithm accuracy on LFW

Algorithm	threshold	accuracy
Original	1.11	0.9625
Face-tracking Feature Extraction	1.11	0.9625

lost face identity in each frame. The data in the Table.3 is the real-time frame rate of these 9 frames, which can be seen to be significantly improved.

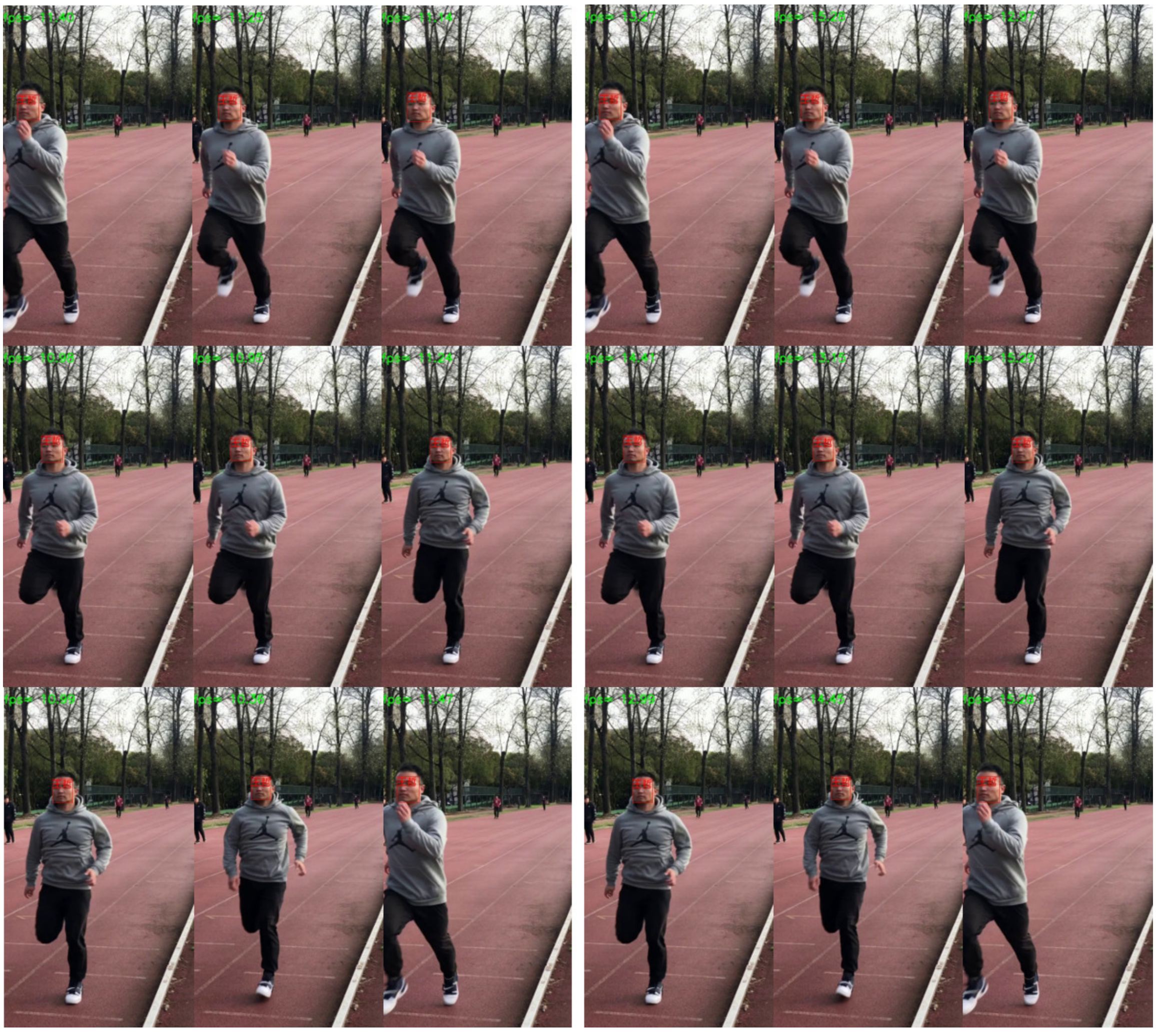


Figure 8. These two frame to frame pictures are from a part of the test video, which shows the running process of athletes on the track. We can see the comparison between the real-time frame rate without Face-tracking Feature Extraction (left) and the real-time frame rate with Face-tracking Feature Extraction (right) in Table. 3.

4.4. **Results.** The scene of the test video mainly involves the movement, appearance and disappearance of the face. These videos contain 1-5 known faces and some unknown faces, ranging from about 5 seconds to more than 1 minutes. It can be seen that after face tracking, the process of face feature extraction is tracked and replaced, which can greatly improve the frame rate.

Table 3. Real time FPS comparison

Algorithm	Frame1	Frame2	Frame3	Frame4	Frame5
No tracking	11.40	11.25	11.14	10.98	10.95
Face-tracking	13.27	15.28	12.79	14.41	13.15
Algorithm	Frame6	Frame7	Frame8	Frame9	—
No tracking	11.24	10.99	10.36	11.47	—
Face-tracking	15.29	12.99	14.45	15.28	—

The Table.4 shows the information of the six videos we selected. This information includes the length of the video (Time), the number of input faces(Number of known faces) and the total number of faces(Total number). We did not completely prohibit the appearance of uknown face, because the appearance of unknown face is also the result of detection and feature extraction, which does not affect the effect of tracking algorithm.

Table 4. Information of multiple videos

Information	Video1	Video2	Video3	Video4	Video5	Video6
Time(s)	8	13	21	13	93	5
Number of known faces	1	3	2	2	4	1
Total number	2	9	2	2	4	2

Table 5. Rescults

Algorithm	Video1	Video2	Video3	Video4	Video5	Video6
No tracking	14.83	5.93	6.93	7.17	4.99	5.02
Face-tracking	25.89	10.30	17.06	17.67	11.97	12.05
Promotion rate(%)	74.58	73.69	146.17	146.44	139.88	140.04

The data shown in Table.5 is the comparison of the average frame rate of the whole process of the six randomly selected test videos.We can see that the percentage of frame rate increase is very large. The reason is that the original algorithm needs three processes: MTCNN calculation, Inception-resnet-v1 calculation and face embedding vector comparison. The process of Inception-resnet-v1 calculation and face embedding vector comparison is very time-consuming (if the amount of data in the database is huge, the embedding vector comparison will also occupy a large amount of calculation), our algorithm uses the tracking algorithm to replace the calculation process of partial frame Inception-resnet-v1, and avoids frequent access to the database, so it can greatly improve the frame rate.

From the above experimental results, we can determine that our algorithm does not lag behind in the accuracy of face recognition after the test of LFW data set. At the same time, through the frame rate test of actual scene video, the experiment shows that for the real-time detection of multiple faces, the speed of our algorithm has been greatly improved, and the improvement of multiple faces is greater than that of a single face. Therefore, compared with the original algorithm, our algorithm is more likely to be applied to the scene that needs real-time detection in the industrial field, such as intelligent runway detection.

5. **Conclusion.** At present, deep learning has been applied in many fields and achieved good results[20,21]. In this paper, our contribution is mainly to insert the tracking algorithm into the two-stage target detection and recognition algorithm for multi-target scenes, which can significantly reduce the amount of real-time calculation and improve the number of video frames. Meanwhile, the algorithm can also be used as a practical deployment algorithm for engineering applications in the future. Of course, it still has some limitations. For example, it is mainly aimed at the two-stage algorithm and can not be applied to the one-stage algorithm. In addition, the tracking algorithm used is also relatively simple. In the future, for the problem of algorithm acceleration in multi-target scenarios, I think we can continue to combine the recognition algorithm with the tracking algorithm. We can learn from the ideas of other scholars and try to apply the tracking algorithm combined with some deep learning networks (such as the application of LSTM network[22]) to multi-target tasks to obtain better results

Acknowledgment. This research work is also partially supported by Ningbo Science and Technology innovation 2025 major project under Grants No. 2021Z010 and No. 2021Z063.

REFERENCES

[1] S. S. Farfade, M. J. Saberian, and L.-J. Li, "Multi-view face detection using deep convolutional neural networks," in *Proceedings of the 5th ACM on International Conference on Multimedia Retrieval*, 2015, pp. 643–650.

[2] J. Deng, J. Guo, E. Ververas, I. Kotsia, and S. Zafeiriou, "Retinaface: Single-shot multi-level face localisation in the wild," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 5203–5212.

[3] Z. Huang, E. Zhou, and Z. Cao, "Coarse-to-fine face alignment with multi-scale local patch regression," *arXiv preprint arXiv:1511.04901*, 2015.

[4] B. Browatzki and C. Wallraven, "3fabrec: Fast few-shot face alignment by reconstruction," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 6110–6120.

[5] W. Liu, Y. Wen, Z. Yu, M. Li, B. Raj, and L. Song, "Sphereface: Deep hypersphere embedding for face recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 212–220.

[6] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 1–9.

[7] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "Ssd: Single shot multibox detector," in *European conference on computer vision.* Springer, 2016, pp. 21–37.

[8] Y.-C. Chiu, C.-Y. Tsai, M.-D. Ruan, G.-Y. Shen, and T.-T. Lee, "Mobilenet-ssdv2: an improved object detection model for embedded systems," in *2020 International conference on system science and engineering (ICSSE).* IEEE, 2020, pp. 1–5.

[9] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 779–788.

[10] J. Redmon and A. Farhadi, "Yolo9000: better, faster, stronger," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 7263–7271.

[11] J. Redmon and A. Farhadi, "Yolov3: An incremental improvement," *arXiv preprint arXiv:1804.02767*, 2018.

[12] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao, "Yolov4: Optimal speed and accuracy of object detection," *arXiv preprint arXiv:2004.10934*, 2020.

[13] Z. Ge, S. Liu, F. Wang, Z. Li, and J. Sun, "Yolox: Exceeding yolo series in 2021," *arXiv preprint arXiv:2107.08430*, 2021.

[14] W. Yang and Z. Jiachun, "Real-time face detection based on yolo," in *2018 1st IEEE international conference on knowledge innovation and invention (ICKII).* IEEE, 2018, pp. 221–224.

[15] K. Zhang, Z. Zhang, Z. Li, and Y. Qiao, "Joint face detection and alignment using multitask cascaded convolutional networks," *IEEE signal processing letters*, vol. 23, no. 10, pp. 1499–1503, 2016.

[16] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[17] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[18] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. A. Alemi, "Inception-v4, inception-resnet and the impact of residual connections on learning," in *Thirty-first AAAI conference on artificial intelligence*, 2017.

[19] F. Schroff, D. Kalenichenko, and J. Philbin, "Facenet: A unified embedding for face recognition and clustering," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 815–823.

[20] M.-E. Wu, J.-H. Syu, and C.-M. Chen, "Kelly-based options trading strategies on settlement date via supervised learning algorithms," *Computational Economics*, pp. 1–18, 2022.

[21] J. M.-T. Wu, Q. Teng, S. Huda, Y.-C. Chen, and C.-M. Chen, "A privacy frequent itemsets mining framework for collaboration in iot using federated learning," *ACM Transactions on Sensor Networks (TOSN)*, 2022.

[22] S. Kumar, A. Damaraju, A. Kumar, S. Kumari, and C.-M. Chen, "Lstm network for transportation mode detection," *Journal of Internet Technology*, vol. 22, no. 4, pp. 891–902, 2021.