Scientific Research Publishing

International Journal of Intelligence Science, 2021, 11, 1-16
https://www.scirp.org/journal/ijis
ISSN Online: 2163-0356
ISSN Print: 2163-0283

Lightweight FaceNet Based on MobileNet

Xinzheng Xu[1,2*], Meng Du[1], Huanxiu Guo[1], Jianying Chang[1], Xiaoyang Zhao[1]

[1]School of Computer Science and Technology, China University of Mining and Technology, Xuzhou, China
[2]Engineering Research Center of Mining Digital Ministry of Education, China University of Mining and Technology, Xuzhou, China
Email: *xuxinzh@163.com

How to cite this paper: Xu, X.Z., Du, M., Guo, H.X., Chang, J.Y. and Zhao, X.Y. (2021) Lightweight FaceNet Based on MobileNet. *International Journal of Intelligence Science*, **11**, 1-16.
https://doi.org/10.4236/ijis.2021.111001

Received: October 29, 2020
Accepted: November 28, 2020
Published: December 2, 2020

Abstract

Face recognition is a kind of biometric technology that recognizes identities through human faces. At first, the speed of machine recognition of human faces was slow and the accuracy was lower than manual recognition. With the rapid development of deep learning and the application of Convolutional Neural Network (CNN) in the field of face recognition, the accuracy of face recognition has greatly improved. FaceNet is a deep learning framework commonly used in face recognition in recent years. FaceNet uses the deep learning model GoogLeNet, which has a high accuracy in face recognition. However, its network structure is too large, which causes the FaceNet to run at a low speed. Therefore, to improve the running speed without affecting the recognition accuracy of FaceNet, this paper proposes a lightweight FaceNet model based on MobileNet. This article mainly does the following works: Based on the analysis of the low running speed of FaceNet and the principle of MobileNet, a lightweight FaceNet model based on MobileNet is proposed. The model would reduce the overall calculation of the network by using deep separable convolutions. In this paper, the model is trained on the CASIA-WebFace and VGGFace2 datasets, and tested on the LFW dataset. Experimental results show that the model reduces the network parameters to a large extent while ensuring the accuracy and hence an increase in system computing speed. The model can also perform face recognition on a specific person in the video.

Keywords

Face Recognition, Deep Learning, FaceNet, MobileNet

1. Introduction

With the advent of the era of big data, artificial intelligence has been developing more rapidly. Artificial intelligence involves many fields, including deep learn-

ing [1], reinforcement learning [2], cluster analysis [3] and support vector machine [4] and other branches. As an emerging field of artificial intelligence in recent years, deep learning aims to use computers to simulate the human brain to think and learn. The rapid development of deep learning in the fields of computer vision [5], natural language processing [6], data mining and robotics in recent years has opened a new chapter in the history of human science of artificial intelligence.

Computer vision is an application of machine learning in the field of vision and an important part of the field of artificial intelligence. The purpose of computer vision is to collect pictures or videos, analyze the pictures or videos, and accordingly obtain the required information. Computer vision is widely used nowadays: video surveillance, automatic drive, medical treatment, face punching, and consumption are all supported by computer vision. Studying computer vision can start from the perspective of object vision and space vision. The purpose of object vision is to determine the type of object, while space vision is to determine the position and shape of the object. At present, there are some tasks such as image classification [7], face recognition [8] [9] [10] and object detection [11] in the field of computer vision. Among them, face recognition has been always a work of great significance.

With the development of the times, face recognition has gradually evolved from artificial recognition to machine recognition, and the accuracy of machine recognition has long surpassed that of human beings. Face recognition is a kind of biometrics technology that recognizes identities through facial features. Compared with other biometrics, face recognition has the advantages of naturalness, uniqueness, and inconsistency. Other biometric methods such as fingerprint recognition and iris recognition are not natural, and require pressure sensors and other equipments. Face recognition is not only used in the field of video surveillance and finance, but also shows its broad application space in many scenarios, such as transportation, education, medical care, and e-commerce. The rapid development of deep learning has made deep learning models widely used in face recognition. Since its introduction, the deep neural network model has been widely used in computer vision tasks such as image classification and face detection [12], and has achieved very good results. Convolutional Neural Network (CNN) [13], the appearance of deep learning models such as neural networks based on probabilistic decision-making, greatly improved the accuracy of face recognition. With the fierce development of deep learning, face recognition technology continues to reach new heights, and the proposal of FaceNet [14] [15] has increased the face recognition rate in the LFW dataset to more than 99%. Face recognition problems can generally be divided into face detection and face recognition. The so-called face detection is not only to detect whether there is a face in the photo, but also to remove the unrelated parts of the picture. In the early days, face detection and face recognition could only be achieved separately through different algorithm frameworks. To realize face detection and face rec-

ognition, it is necessary to train two neural networks at the same time. Until 2015, Google's FaceNet once completely solved this problem, for the first time unified the two into the same framework. FaceNet is a general face recognition system that maps images to Euclidean space through deep neural networks. The spatial distance is related to the similarity of pictures. The distance between different images of the same person in Euclidean space is small, and the distance between images of different people in Euclidean space is large, which makes FaceNet can be used for face detection, recognition and clustering [16].

In face recognition, posture and lighting have always been a long-standing problem. The traditional face recognition method based on convolutional neural network is to use CNN's twin network [17] to extract face features, and then use Support Vector Machine (Support Vector Machine, SVM) and other methods for classification. However, FaceNet directly learns the mapping of images to points on the Euclidean space and judge whether the two images which the distance between the features of the two images in the Euclidean space directly corresponds to are similar. The Euclidean distance between image features is shown in **Figure 1**. The numbers correspond to the Euclidean distance between this set of image features. The Euclidean distance of 1.1 is used as the threshold. When the Euclidean distance is greater than 1.1, the two The faces in the images are determined to be from different people, and when the Euclidean distance is less than 1.1, the faces in the two images are determined to be from the same person.

FaceNet has two different deep network structures, both of which are deep convolutional networks. The first structure is based on the Zeiler & Fergus model [18], which consists of multiple intersecting layers such as convolutional layers, nonlinear activation layers, local response normalization layers, and maximum pooling layers. The second structure is based on the Inception model of

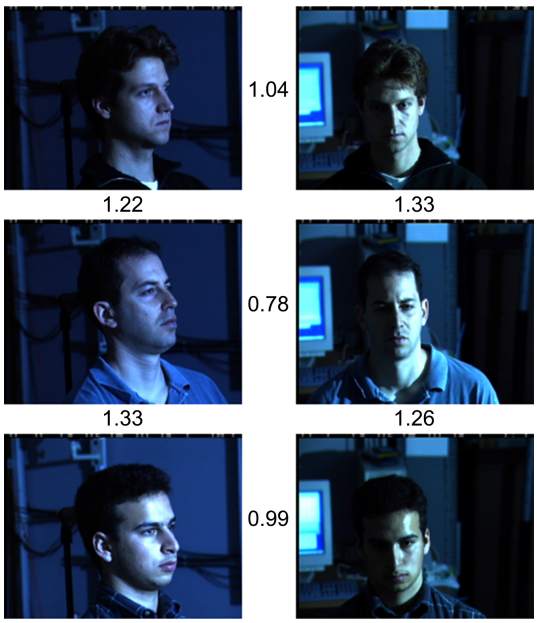

Figure 1. Euclidean distance between image features.

Szegedy *et al.*, which uses mixed layers that run several different convolutional and pooling layers in parallel and concatenate their responses. These models can reduce the number of FLOPS to achieve better performance. Zhenyao *et al.* used a deep network to "distort" human faces into a canonical frontal view, and then learned to classify each human face as a known CNN. For facial verification, the principal component analysis on the network output is used in combination with a set of SVMs. Taigman *et al.* proposed a multi-stage method to align the face with a general 3D model. And they trained a multi-category network that can perform facial recognition on more than 4,000 identities. The authors also conducted experiments on the proposed twin network, in which they optimized the L1 distance between two facial features. Their best performance on LFW comes from the collection of three networks using different arrangements and color channels, using nonlinear SVM to combine the prediction distances of these networks (nonlinear SVM prediction based on $\chi 2$ kernel), through semantic and visual similarity Ranking images. The new generation of FaceNet uses the Inception-ResNet-v2 network, which combines Microsoft's ResNet idea of residual network on the basis of the original Google's Inception series network [19] [20] [21]. Among them, the residual connection [22] can train deeper neural networks, while also significantly simplifying the Inception block. Regarding FaceNet, the current research focus is mainly on proposing a more efficient and concise network structure. With the rapid development of lightweight models in recent years, it is bound to provide new ideas for the innovation of FaceNet network structure.

The main contributions of this article are as follows:

1) A Fast-FaceNet model based on MobileNet is proposed to reduce the overall calculation of the network.

2) Fast-FaceNet was applied to video face recognition to improve the recognition rate while ensuring a certain recognition accuracy rate.

This paper is divided into five parts: Section 1 introduces the relevant background, the related work in recent years, and summarizes the main work and organizational structure of this paper. Section 2 introduces the relevant basic theory. Sections 3 and 4 are the core of this paper, model architecture and the analysis of experimental results. The final part summarizes the entire article.

2. Basic Theory

2.1. FaceNet Basic Structure

The FaceNet system can directly map face images to a compact Euclidean space,

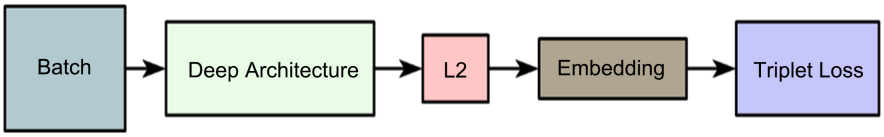

Figure 2. FaceNet architecture.

where the length of the spatial distance directly corresponds to the measure of face similarity. Once this space is generated, you can use standard techniques with FaceNet embedding as feature vectors to easily perform tasks such as face recognition, verification, and clustering. The advantage of this model is that only a small amount of processing on the image can be used as input. At the same time, the accuracy of the model is very high in the data set. Facenet can be widely used in face recognition in mobile termial. Its network structure is shown in Figure 2.

The FaceNet network consists of a batch input layer and a deep convolutional network, and then L2 normalization, which leads to face embedding, and finally calculates the triplet loss to make the distance between the same objects. As small as possible, the distance between different objects is as large as possible. It uses a deep convolutional neural network to learn the Euclidean embedding method of each image, and trains the network so that the squared L2 distance in the embedding space directly corresponds to the face similarity. FaceNet directly uses the Loss function of Triplets-based LMNN (Maximum Boundary Nearest Neighbor Classification) to train the neural network, and the network output is a 128-dimensional vector space. The selected Triplets contain two matching face thumbnails and a non-matching face thumbnail. The Loss function target distinguishes positive and negative classes by distance boundaries.

2.2. GoogLeNet

The deep neural network in the classic FaceNet system is GoogLeNet which uses the Inception module, so it is also called the Inception network.

The original Inception module contains several convolutions of different sizes, namely 1 × 1 convolution, 3 × 3 convolution and 5 × 5 convolution, and also includes a 3 × 3 maximum pooling layer. The features obtained by these convolutional layers and pooling layers are aggregated together as the final output, which is also the input of the next module. The original Inception module is shown in Figure 3.

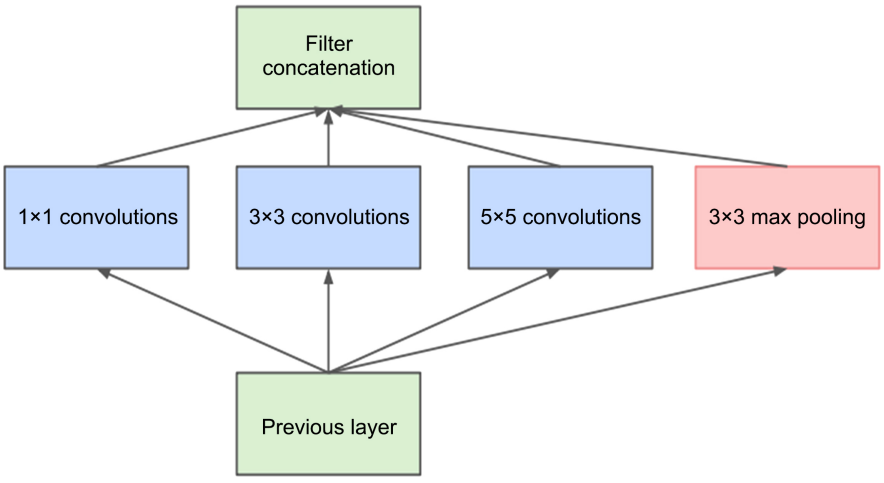

Figure 3. Original inception module.

However, a larger convolution kernel is used in the original Inception module, and the calculation complexity is larger, which can only limit the number of feature channels. So GoogLeNet uses 1 × 1 convolution to optimize, that is, firstly use 1 × 1 convolution to perform up-down dimension, and secondly perform convolution and aggregation on multiple sizes at the same time. The size reduction Inception module is shown in Figure 4.

The entire GoogLeNet network is formed by stacking Inception modules. The entire network has a total of 22 layers. The specific network and parameter configuration are shown in Table 1.

The modular structure (Inception structure) adopted by GoogLeNet is easy to add and modify. At the end of the network, the average pooling is used to replace the fully connected layer, which can improve the accuracy. However, GoogLeNet's network model is relatively large, and the calculation speed is also slow.

Table 1. GoogLeNet architecture.

Type	Size-in	Size-out	Kernel (stride)
Conv	224 × 224 × 3	112 × 112 × 64	7 × 7 × 3 (2)
Max pool	112 × 112 × 64	56 × 56 × 64	3 × 3 × 64 (2)
Inception (2)	56 × 56 × 64	56 × 56 × 192	3 × 3 × 192 (1)
Max pool	56 × 56 × 192	28 × 28 × 192	3 × 3 × 192 (2)
Inception (3a)	28 × 28 × 192	28 × 28 × 256	1 × 1 × 64 (1), 3 × 3 × 128 (1), 5 × 5 × 32 (1), 1 × 1 × 32(1)
Inception (3b)	28 × 28 × 256	28 × 28 × 320	1 × 1 × 64 (1), 3 × 3 × 128 (1), 5 × 5 × 64 (1), 1 × 1 × 64(1)
Inception (3c)	28 × 28 × 320	14 × 14 × 640	3 × 3 × 256 (2), 5 × 5 × 64 (2)
Inception (4a)	14 × 14 × 640	14 × 14 × 640	1 × 1 × 256 (1), 3 × 3 × 192 (1), 5 × 5 × 64 (1), 1 × 1 × 128(1)
Inception (4b)	14 × 14 × 640	14 × 14 × 640	1 × 1 × 224 (1), 3 × 3 × 224 (1), 5 × 5 × 64 (1), 1 × 1 × 128(1)
Inception (4c)	14 × 14 × 640	14 × 14 × 640	1 × 1 × 192 (1), 3 × 3 × 256 (1), 5 × 5 × 64 (1), 1 × 1 × 128(1)
Inception (4d)	14 × 14 × 640	14 × 14 × 640	1 × 1 × 160 (1), 3 × 3 × 288 (1), 5 × 5 × 64 (1), 1 × 1 × 128(1)
Inception (4e)	14 × 14 × 640	7 × 7 × 1024	3 × 3 × 256 (2), 5 × 5 × 128 (2)
Inception (5a)	7 × 7 × 1024	7 × 7 × 1024	1 × 1 × 384 (1), 3 × 3 × 384 (1), 5 × 5 × 128 (1), 1 × 1 × 128(1)
Inception (5b)	7 × 7 × 1024	7 × 7 × 1024	1 × 1 × 384 (1), 3 × 3 × 384 (1), 5 × 5 × 128 (1), 1 × 1 × 128(1)
Avg Pool	7 × 7 × 1024	1 × 1 × 1024	7 × 7 × 3 (1)
FC	1 × 1 × 1024	1 × 1 × 128	1024 × 128 (1)

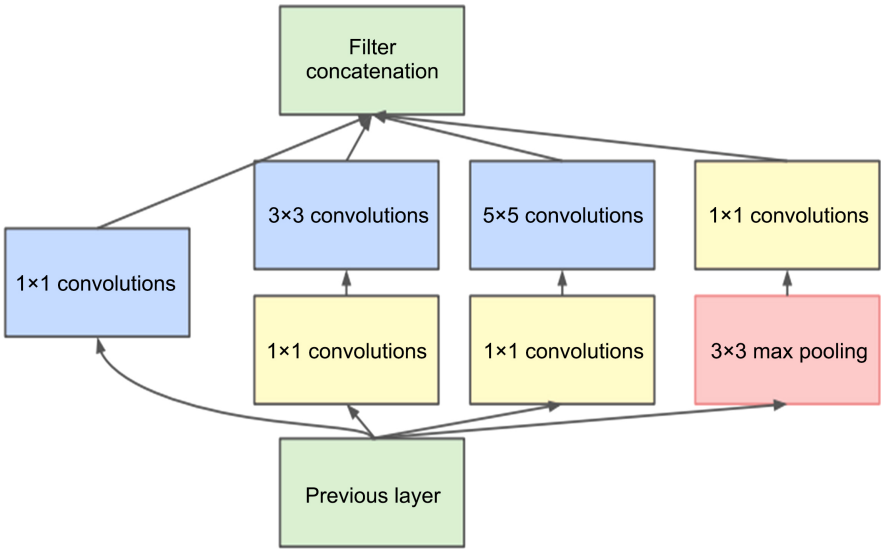

Figure 4. Inception module with reduced size.

2.3. MobileNet

MobileNet [23] [24] [25] is a lightweight deep neural network which is based on streamline architecture and built by using deep separable convolution. When FaceNet performs face recognition, in order to achieve a certain degree of accuracy, the network is relatively complex. Therefore, these complex networks will affect the size and speed of the model. For example, when the model is used in automatic driving and criminal detection, the real-time nature of visual tasks and other factors need to be considered by reason of the limitations of the platform's calculation. MobileNet proposes a high-performance architecture with hyperparameters, which can make the model smaller and the calculation speed faster. And it is very practical for face recognition systems.

The core layer built by MobileNet is a deep separable filter. Deep separable convolution is a form of deconvolution. The standard convolution operation directly extracts the features from the input and combines them into a series of outputs. The depth separable convolution divides this process into two layers: one layer is the depth convolution, which is used to extract each channel of the input separately Features; One layer is a point-by-point convolution, which uses a 1 × 1 convolution to combine the output of the previous step. This decomposition has the effect of significantly reducing the calculation and model size. **Figure 5**, **Figure 6** and **Figure 7** show the process of decomposing the standard convolution integral into deep convolution and point-by-point convolution.

Suppose the size of the input feature map is $D_F \times D_F \times M$, M is the number of input channels, N is the number of output channels, the parameters of a standard convolutional layer are $D_K \times D_K \times M \times N$, and D_K is the size of the convolution kernel. If the space size of the output feature map remains unchanged, the calculation cost of standard convolution is shown in Equation (1):

$$D_K \times D_K \times M \times N \times D_F \times D_F \tag{1}$$

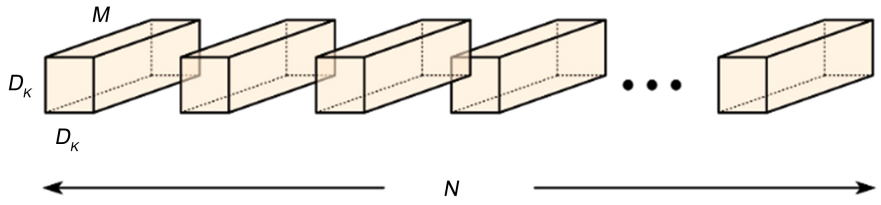

Figure 5. Standard convolution.

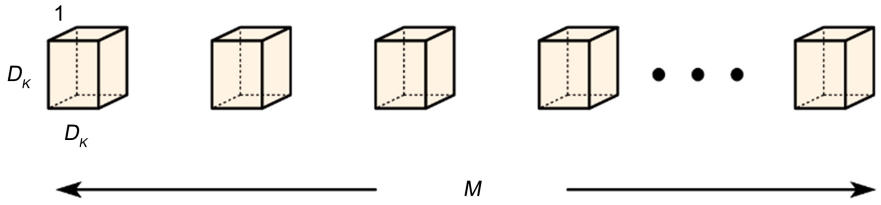

Figure 6. Depthwise convolution.

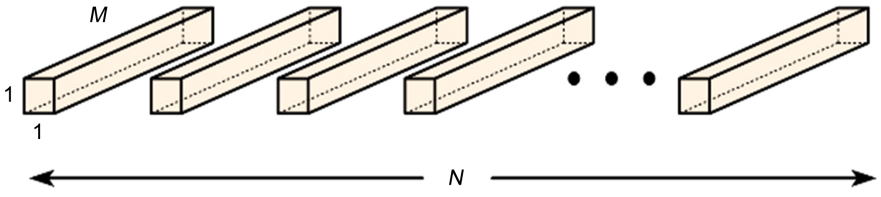

Figure 7. Pointwise convolution.

The MobileNet model uses deep separable convolutions to break the interaction between the number of output channels and the size of the kernel to greatly reduce the computational cost. The calculation cost of deep convolution is shown in Equation (2):

$$D_K \times D_K \times M \times D_F \times D_F \tag{2}$$

Although deep convolution is much more efficient than standard convolution, it only filters the input channels and does not combine them to generate new features. An additional 1 × 1 convolution is required to combine the features obtained by these filters to form a New multi-channel features. The calculation cost of the final depth separable convolution is the sum of depth convolution and point-by-point convolution, as shown in Equation (3):

$$D_K \times D_K \times M \times D_F \times D_F + M \times N \times D_F \times D_F \tag{3}$$

By decomposing the standard convolution integral into deep convolution and point-by-point convolution, the calculation amount is reduced as shown in Equation (4):

$$\frac{D_K \times D_K \times M \times D_F \times D_F + M \times N \times D_F \times D_F}{D_K \times D_K \times M \times N \times D_F \times D_F} = \frac{1}{N} + \frac{1}{D_K^2} \tag{4}$$

MobileNet which uses deep separable convolution and 8 - 9 times less computation than standard convolution can greatly improve the operation rate. Therefore, this article uses MobileNet to replace the deep learning model in FaceNet.

3. Lightweight FaceNet based on MobileNet

3.1. Network model design

The original FaceNet network is relatively complex. However, these complex networks will affect the size and speed of the model. In order to be better deployed on the mobile terminal without affecting the accuracy of face recognition. This paper uses MobileNet to replace GoogLeNet, and proposes a Fast-FaceNet model based on MobileNet in order to improve the practicality of FaceNet. Its network structure is shown in Figure 8.

In Figure 8, Batch refers to the input face image samples that have been detected by face detection and cropped to a fixed size, and then feature extraction through the lightweight model MobileNet, then L2 feature normalization. Finally, classify through the Triplet loss function so that the feature distance between the same identities should be as small as possible and the feature distance between different identities should be as large as possible.

The percentage of the total parameters and the total calculation amount of each operation of MobileNet in Fast-FaceNet is shown in Table 2.

It can be seen from Table 2 that MobileNet spends 95% of its computing time in the 1 × 1 convolution. The 1 × 1 convolution also contains 75% of the parameters, and almost all other parameters are located in the fully connected layer. The 1 × 1 convolution does not need to be reordered in memory, and can be implemented directly using general matrix multiplication, therefore it improves the operation rate.

The results of comparing the parameters of MobileNet and GoogLeNet with the amount of calculation are shown in Table 3.

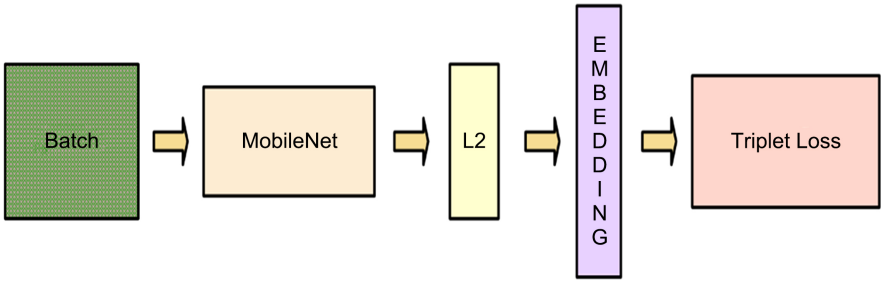

Figure 8. Fast-FaceNet network architecture.

Table 2. Each operation of MobileNet accounts for the percentage of total parameters and total calculations.

Type	Calculation	Parameter
Conv 1 × 1	94.86%	74.59%
Conv DW 3 × 3	3.06%	1.06%
Conv 3 × 3	1.19%	0.02%
Fully Connected	0.18%	24.33%

After comparison, it can be found that MobileNet is smaller than GoogleNet in size, less in parameters, and the amount of calculation is reduced by more than 2.5 times. So it is effective that this article uses MobileNet to improve FaceNet.

The parameter configuration of each network layer of Fast-FaceNet is shown in Table 4.

Table 3. Comparison of parameters and calculations between MobileNet and GoogleNet.

Model	Calculation	Parameter
MobileNet	569	4.2
GoogLeNet	1600	7.5

Table 4. Fast-FaceNet paramter configuration.

Type	Size-in	Size-out	Kernel (stride)
Conv	224 × 224 × 3	112 × 112 × 32	3 × 3 × 3 × 32 (2)
Conv dw	112 × 112 × 32	112 × 112 × 32	3 × 3 × 32 dw (1)
Conv	112 × 112 × 32	112 × 112 × 64	1 × 1 × 32 × 64 (1)
Conv dw	112 × 112 × 64	56 × 56 × 64	3 × 3 × 64 dw (2)
Conv	56 × 56 × 64	56 × 56 × 128	1 × 1 × 64 × 128 (1)
Conv dw	56 × 56 × 128	56 × 56 × 128	3 × 3 × 128 dw (1)
Conv	56 × 56 × 128	56 × 56 × 128	1 × 1 × 128 × 128 (1)
Conv dw	56 × 56 × 128	28 × 28 × 128	3 × 3 × 128 dw (2)
Conv	28 × 28 × 128	28 × 28 × 256	1 × 1 × 128 × 256 (1)
Conv dw	28 × 28 × 256	28 × 28 × 256	3 × 3 × 256 dw (1)
Conv	28 × 28 × 256	28 × 28 × 256	1 × 1 × 256 × 256 (1)
Conv dw	28 × 28 × 256	14 × 14 × 256	3 × 3 × 256 dw (2)
Conv	14 × 14 × 256	14 × 14 × 512 14 × 14 × 512	1 × 1 × 256 × 512 (1)
5× Conv dw Conv	14 × 14 × 512 14 × 14 × 512	14 × 14 × 512	3 × 3 × 512 dw (1) 1 × 1 × 512 × 512 (1)
Conv dw	14 × 14 × 512	7 × 7 × 512	3 × 3 × 512 dw (2)
Conv	7 × 7 × 512	7 × 7 × 1024	1 × 1 × 512 × 1024 (1)
Conv dw	7 × 7 × 1024	7 × 7 × 1024	3 × 3 × 1024 dw (1)
Conv	7 × 7 × 1024	7 × 7 × 1024	1 × 1 × 1024 × 1024 (1)
Avg Pool	7 × 7 × 1024	1 × 1 × 1024	7 × 7 × 3 (1)
FC	1 × 1 × 1024	1 × 1 × 1024	1024 × 1000 (1)

3.2. Selection of Loss Function

This paper uses the loss function based on Triplets' maximum boundary nearest neighbor classification algorithm to train the neural network. The network directly outputs a 128-dimensional vector space.

Triplets means triples, that is, the loss function is calculated by three parameters: Anchor, Negative, and Positive. Anchor refers to the benchmark image, Positive refers to the image under the same category as Anchor, and Negative refers to the category different from Anchor picture.

The loss function makes the feature distance between the same identities as small as possible, while the feature distance between different identities is as large as possible. Therefore, the distance of the points in the Euclidean space of the features corresponding to the two images directly corresponds to the two Whether the images are similar. The process of Triplet Loss is shown in Figure 9.

As shown in Figure 9, the purpose of Triplet Loss is to embed the face image X into the Euclidean space $f(x) \in R^d$ of the D dimension, ensuring the distance when the image of a specific person x_i^a (reference picture) is compared with its own other images x_i^p (positive values), Which is closer than when the person's images x_i^n (negative values) are compared with other people's images.

$$L = \sum_i^N \left[\left\| f\left(x_i^a\right) - f\left(x_i^p\right) \right\|_2^2 - \left\| f\left(x_i^a\right) - f\left(x_i^n\right) \right\|_2^2 + \propto \right]_+ \tag{5}$$

where the $L2$ on the left is the intra-class distances, and the $L2$ on the right is the inter-class distances. α is a constant. The meaning of formula (5) is to optimize the triplets that do not meet the conditions; for the triplets that meet the conditions, set aside and ignore. In the optimization process, the gradient descent method is used to make the loss function decrease continuously, that is, the intra-class distances decreases and inter-class distances increases continuously.

The choice of Triples is crucial to the convergence of the model. In actual training, it is unrealistic to calculate the maximum and minimum distances between images across all training samples, and it is also difficult to converge due to incorrectly labeled images. Therefore, this article sets every 64 samples as a Mini-Batch, and uses online generation to select Triplets in each Mini-Batch. In each Mini-Batch, two face pictures are selected as positive samples for a single individual, and other face pictures are randomly selected as negative samples. In order to avoid premature training convergence caused by improper selection of negative samples, this paper uses Equation (6) to filter negative samples:

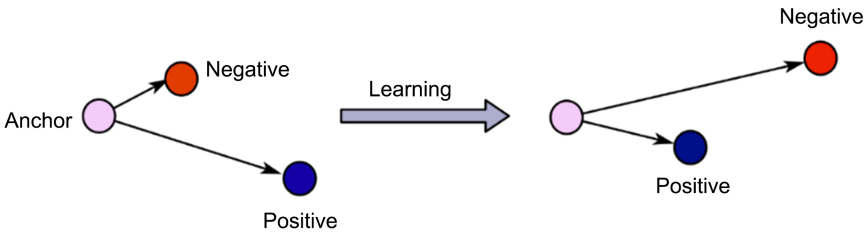

Figure 9. Triplet loss process.

$$\left\| f\left(x_i^a\right) - f\left(x_i^p\right) \right\|_2^2 < \left\| f\left(x_i^a\right) - f\left(x_i^n\right) \right\|_2^2 \tag{6}$$

4. Experimental Results and Analysis

In order to verify the model proposed in this paper, the CASIA WebFace dataset and the VGGFace2 dataset are used to train the proposed Fast-FaceNet model, and the trained model is tested with the LFW dataset. All the experimentally verified platforms in this article use Google open source deep learning platform Tensorflow, which is an artificial intelligence-oriented learning system that and uses tensorflow to calculate logarithmic graphs. The platform mainly analyzes and processes neural network models in artificial intelligence, which is easy to use.

In this paper, the AdaGrad optimizer is used to train the MobileNet model by a stochastic gradient descent method. The learning rate is 0.02. After 300 hours of training on the CPU cluster, the loss function drops significantly, and the boundary value α is set to 0.2. Since FaceNet only needs a small amount of processing on the image (only needs to crop the face area without additional preprocessing, such as 3D alignment, etc.), and then it can be used as the input of the model, in this article we first runs A face detector (implemented through MTCNN) on each image, and generate a tight bounding box around each face, and then adjust the size of these face thumbnails to 224 × 224 to input.

Although the basic MobileNet is already very small and the delay is very short, in order to test whether MobileNet can be further reduced and Fast-FaceNet's operation rate can be faster when using MobileNet to replace the original GoogLeNet in FaceNet, This article introduces a parameter called width multiplication Number θ, whose function is to make the network of each layer thinner evenly. For a given layer and width multiplier θ, the number of input channels becomes θM, and the number of output channels becomes θN, where $\theta \in (0,1]$. Take θ = 0.25, 0.5, 0.75, 1 to train Fast-FaceNet with different network widths and experiment on the LFW dataset. The results are shown in Table 5.

As shown in Table 5, with different width multipliers, the accuracy and rate of recognition of the entire FaceNet on the LFW data set have changed. The two factors of operation rate and recognition accuracy can be considered comprehensively. When the width multiplier is 0.75 and 1, the system performance is optimal. Therefore, comparing the Fast-FaceNet with the width multiplier of 0.75 and 1 to the original FaceNet system, the experimental results obtained on the LFW data set are shown in Table 6.

As can be seen in Table 6, Fast-FaceNet compared to the original FaceNet when the width multiplier is 1, although the accuracy of face recognition is slightly reduced, the calculation time is greatly reduced; when the width multiplier is 0.75, The recognition accuracy rate of Fast-FaceNet is reduced by 0.9% compared to the time when the width multiplier is 1, but the calculation rate has been greatly improved.

In order to test the effect of Fast-FaceNet on face recognition of video, a piece

of film and television video was intercepted on the network. For two objects respectively input as shown in Figure 10 the results after Fast-FaceNet recognition are shown in Figure 11.

As shown in Figure 11, the object on the left in Figure 10 has been successfully identified in the video by Fast-FaceNet and is marked by a red frame, and the object on the right has been marked by a yellow frame. Compare the results of FaceNet and Fast-FaceNet for video face detection, and use F1-score to evaluate the experimental results. The results are shown in Table 7.

As can be seen in Table 7, Fast-FaceNet compared to the original FaceNet, although the F1-score of face recognition is slightly reduced, there is a certain recognition accuracy rate.

Table 5. Experimental results on LFW dataset.

Model	Parameter	Calculation	Accuracy	CPU processing time for a picture
0.25Fast-FaceNet	0.9	62	79.53%	26
0.50Fast-FaceNet	1.6	181	88.02%	45
0.75Fast-FaceNet	3.1	392	97.74%	88
1.00Fast-FaceNet	4.8	684	98.63%	137

Table 6. Comparison of experimental results.

Model	Dataset	Accuracy	CPU processing time for a picture
FaceNet	CASIA-WebFace	99.05%	245
	VGGFace2	99.65%	
0.75Fast-FaceNet	CASIA-WebFace	97.25%	88
	VGGFace2	97.74%	
1.00Fast-FaceNet	CASIA-WebFace	98.15%	137
	VGGFace2	98.63%	

Figure 10. Picture of Fast-FaceNet input.

Figure 11. Fast-FaceNet recognition results on video.

Table 7. Experimental results in the video.

	FaceNet	0.75Fast-FaceNet	1.0Fast-FaceNet
Precision	0.8784	0.8314	0.8549
recall	0.7563	0.7095	0.7388
F1-score	0.8128	0.7656	0.7926

5. Conclusion

Based on the classic FaceNet, this paper introduced the lightweight model MobileNet and proposed a lightweight FaceNet based on MobileNet. Firstly the paper introduced the classic model FaceNet, then introduced MobileNet and proposed Fast-FaceNet. Fast-FaceNet was trained on the CASIA-WebFace and VGGFace2 datasets and tested on the LFW dataset. Finally, Fast-FaceNet was applied to video face recognition. It is proved by experiments that Fast-FaceNet greatly improves the recognition rate while ensuring a certain recognition accuracy rate.

Acknowledgements

This work was supported by the National Natural Science Foundation of China (61976217), the Fundamental Research Funds for the Central Universities (No. 2019XKQYMS87), and the Opening Foundation of Key Laboratory of Opto-technology and Intelligent Control, Ministry of Education (KFKT2020-3).

Conflicts of Interest

The authors declare no conflicts of interest regarding the publication of this paper.

References

[1] Deng, L. and Yu, D. (2013) Deep Learning: Methods and Applications. *Foundations and Trends® in Signal Processing*, **7**, 197-387. https://doi.org/10.1561/2000000039

[2] Sutton, R.S. and Barto, A.G. (1998) Reinforcement Learning. In: *A Bradford Book*, MIT Press, Cambridge, Vol. 15, 665-685.

[3] Brown, P.O. (1998) Cluster Analysis and Display of Genome-Wide Expression Patterns. *PNAS*, **95**, 14863-14868.

[4] Saunders, C., Stitson, M.O., Weston, J., *et al.* (2002) Support Vector Machine. *Computer Science*, **1**, 1-28.

[5] Zhang, B. (2010) Computer Vision vs. Human Vision. *IEEE International Conference on Cognitive Informatics*, Beijing, 7-9 July 2010, 3. https://doi.org/10.1109/COGINF.2010.5599750

[6] Adam, L.B.T., Pietra, V.J.D. and Pietra, S.A.D. (2002) A Maximum Entropy Approach to Natural Language Processing. *Computational Linguistics*, **22**, 39-71.

[7] Nowak, E., Jurie, F. and Triggs, B. (2006) Sampling Strategies for Bag-of-Features Image Classification. *Computer Vision ECCV* 2006, 9*th European Conference on Computer Vision*, Graz, 7-13 May 2006, 490-503. https://doi.org/10.1007/11744085_38

[8] Lu, C. and Tang, X. (2015) Surpassing Human-Level Face Verification Performance on LFW with Gaussian Face. *Twenty-Ninth AAAI Conference on Artificial Intelligence*, Austin, Texas, June 2015, 3811-3819.

[9] Sim, T., Baker, S. and Bsat, M. (2002) The CMU Pose, Illumination, and Expression (PIE) Database. *Proceedings of Fifth IEEE International Conference on Automatic Face Gesture Recognition*, Washington DC, 21 May 2002, 53-58.

[10] Sun, Y., Wang, X. and Tang, X. (2015) Deeply Learned Face Representations Are Sparse, Selective, and Robust. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Boston, 7-12 June 2015, 2892-2900. https://doi.org/10.1109/CVPR.2015.7298907

[11] Viola (2001) Robust Real-Time Object Detection. *International Journal of Computer Vision*, **57**, 87. https://doi.org/10.1023/B:VISI.0000013087.49260.fb

[12] Dang, K. and Sharma, S. (2017) Review and Comparison of Face Detection Algorithms. 2017 7*th IEEE International Conference on Cloud Computing, Data Science & Engineering*, Noida, 12-13 January 2017, 629-633. https://doi.org/10.1109/CONFLUENCE.2017.7943228

[13] Chumerin, N. (2017) Convolutional Neural Network.

[14] William, I., Rachmawanto, E.H., Santoso, H.A., *et al.* (2019) Face Recognition Using FaceNet (Survey, Performance Test, and Comparison). 2019 *IEEE Fourth International Conference on Informatics and Computing* (*ICIC*), Semarang, 16-17 October 2019, 1-6. https://doi.org/10.1109/ICIC47613.2019.8985786

[15] Schroff, F., Kalenichenko, D. and Philbin, J. (2015) Facenet: A Unified Embedding for Face Recognition and Clustering. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Boston, 7-12 June 2015, 815-823. https://doi.org/10.1109/CVPR.2015.7298682

[16] Huang, D., Wang, C.D., Wu, J., *et al.* (2019) Ultra-Scalable Spectral Clustering and Ensemble Clustering. *IEEE Transactions on Knowledge and Data Engineering*, **32**, 1212-1226.

[17] Chang, S., Zhang, F., Huang, S., *et al.* (2019) Siamese Feature Pyramid Network for

Visual Tracking. 2019 *IEEE/CIC International Conference on Communications Workshops in China* (*ICCC Workshops*), Changchun, 11-13 August 2019, 164-168. https://doi.org/10.1109/ICCChinaW.2019.8849954

[18] Zeiler, M.D. and Fergus, R. (2014) Visualizing and Understanding Convolutional Networks. In: *European Conference on Computer Vision*, Springer, Cham, 818-833. https://doi.org/10.1007/978-3-319-10590-1_53

[19] Szegedy, C., Liu, W., Jia, Y., *et al.* (2015) Going Deeper with Convolutions. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Boston, 7-12 June 2015, 1-9. https://doi.org/10.1109/CVPR.2015.7298594

[20] Szegedy, C., Vanhoucke, V., Ioffe, S., *et al.* (2016) Rethinking the Inception Architecture for Computer Vision. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Las Vegas, 27-30 June 2016, 2818-2826. https://doi.org/10.1109/CVPR.2016.308

[21] Szegedy, C., Ioffe, S., Vanhoucke, V., *et al.* (2017) Inception-v4, Inception-Resnet and the Impact of Residual Connections on Learning. *Thirty-First AAAI Conference on Artificial Intelligence*, San Francisco, 4-9 February 2017, 4278-4284.

[22] Zhang, K., Sun, M., Han, T.X., *et al.* (2017) Residual Networks of Residual Networks: Multilevel Residual Networks. *IEEE Transactions on Circuits and Systems for Video Technology*, **28**, 1303-1314. https://doi.org/10.1109/TCSVT.2017.2654543

[23] Howard, A.G., Zhu, M., Chen, B., *et al.* (2017) MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications.

[24] Sandler, M., Howard, A., Zhu, M., *et al.* (2018) Mobilenetv2: Inverted Residuals and Linear Bottlenecks. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Salt Lake City, 18-23 June 2018, 4510-4520. https://doi.org/10.1109/CVPR.2018.00474

[25] Howard, A., Sandler, M., Chu, G., *et al.* (2019) Searching for Mobilenetv3. *Proceedings of the IEEE International Conference on Computer Vision*, Seoul, 27 October-2 November 2019, 1314-1324. https://doi.org/10.1109/ICCV.2019.00140