文章编号：2095-2163(2022)10-0130-07 中图分类号：TP391.41 文献标志码：A

# 基于 CNN 的 FaceNet 算法人脸图像识别研究

郝林倩
（福建船政交通职业学院 信息与智慧交通学院，福州 350007）

**摘 要：**当今已是全球网络信息化时代，网络信息安全显得尤为重要。利用人脸这一不可复制的生物特征，用以维护信息安全。采用 CNN 算法，基于 Pytorch 深度学习库，构建 Mobilenetv1 网络模型，在此基础上通过 FaceNet 预测显示的 *Distance* 值与事先设置的阈值对比情况，验证人脸图像，达到人脸识别的目的。实验结果表明，该算法在人脸识别方面取得了较好的效果。

**关键词：**卷积神经网络；Pytorch；Mobilenetv1；FaceNet

## Research on face recognition using FaceNet algorithm based on CNN

HAO Linqian
（School of Information and Intelligent Transportation, Fujian Chuanzheng Communications College, Fuzhou 350007, China）

**【Abstract】** In the era of network informationization network information security is particularly important. Currently, the unrepeatable biometric feature of the human face is used to maintain information security. The CNN algorithm is used to build the Mobilenetv1 network model based on the deep learning library of Pytorch. And the face image is verified by comparing the *Distance* value predicted and displayed by FaceNet with the threshold value set in advance to achieve the purpose of face recognition. Experimental results show that the algorithm has achieved good results in face recognition.

**【Key words】** CNN; Pytorch; Mobilenetv1; FaceNet

## 0 引 言

人脸识别（Face Recognition），是指对人脸图像进行辨识提取，通过分析比较而获得的人脸视觉特征信息数据，并以此来对个人身份特征进行比对或鉴别的计算机技术。分析可知，该技术属于生物学特性识别算法，迄今为止也已然成为现阶段国内外学者的研究热点，即能够根据人们对自然界中某一种特定生命体（一般特指人）的自身所存在的生物学特性，来实现特征识别以区分特定生物体的个体。

卷积神经网络（Convolutional Neural Networks，CNN）算法是目前人脸图像识别训练中具有可观应用前景的一种深度学习方法。深度学习方法的主要优势就在于能够分析训练大量的人脸数据，并能够快速学到人脸训练中的数据所表现出的人脸特征变化，进而对其他未知情况做出准确可靠的人脸表征识别。这种训练方法通常不需要预先设计出对不同身体类型的类域内的差异因素（比如光照、姿势、面部表情、年龄等）所稳健具备的各种特定运动特征，而是完全可以通过从训练结果数据中学习来得到。

## 1 传统人脸图像识别算法

### 1.1 PCA 算法

主成分分析方法（Principal Component Analysis，PCA），是一种使用最广泛的数据降维算法。PCA 的主要思想是将 $n$ 维特征映射到 $k$ 维上，这个 $k$ 维是全新的正交特征、也被称为主成分，是在原有 $n$ 维特征的基础上重新构造出来的 $k$ 维特征[1]。将 2 个数据轴各由某一个原本已经给定坐标的坐标系轴中转置至多个新坐标的坐标系轴中，第一个坐标的选择是原始数据中方差最大的，第二个坐标轴选择的是和第一个坐标轴正交、且具有最大方差的方向，重复该过程，重复的次数则为原始数据的特征数。因此大部分方差都在前几个新坐标中，把后面的坐标忽略，如此就完成了数据降维。

PCA 在本质上主要是要将方差为最大的正交方向数据作为主要的统计特征[2]，并且要在其各个主要正交方向上将数据写“离相关”，即尽量让数据彼此间在其不同主要正交方向数据上完全没有任何相关性，原理示意如图 1 所示。

作者简介：郝林倩（1983-），女，硕士，副教授，主要研究方向：数据挖掘、数据分析、机器学习算法。
通讯作者：郝林倩 Email：378802718@qq.com

收稿日期：2022-08-19

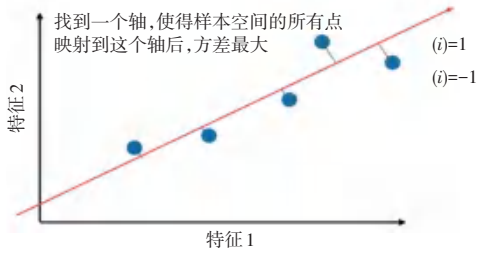


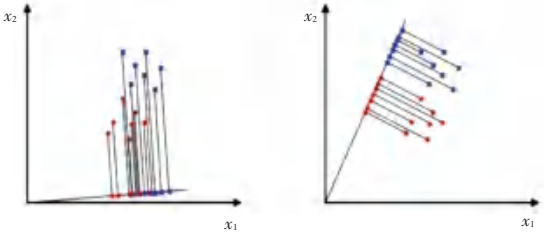

**图 1　PCA 算法原理**

**Fig. 1　Principle of PCA algorithm**

### 1.2　LDA 算法

线性判别分析（Linear Discriminant Analysis，LDA）算法的思路与 PCA 类似，都是对图像的整体分析。不同之处在于，PCA 是通过确定一组正交基对数据进行降维，而 LDA 是通过确定一组投影向量使得数据集不同类的数据投影的差别较大、同一类的数据经过投影更加聚合。在统计形式处理上，PCA 方法与传统 LDA 算法的 2 个最大显著区别即在于，PCA 方法中最终能求得结果的特征向量通常是完全正交的，而在 LDA 方法中求得的结果特征向量就不可能一定全部正交。

利用 LDA 寻找一个投影向量，使得不同类型的数据点在投影向量所在直线上投影能较好地做出区分，如图 2 所示。




（a）不同类别的投影点交织在一起　　（b）不同类别的投影点位于不同区域

**图 2　LDA 算法向量投影**

**Fig. 2　Vector projection of LDA algorithm**

### 1.3　LBPH 算法

局部二进制模式直方图（Local Binary Pattern Histograms，LBPH）人脸识别方法中的技术核心之一就是 LBP 算子。LBP 算子主要是指可以用来分析计算与描述图像局部纹理特征信息关系的一种算子，其最终所能反映到的抽象内容则往往仅仅是描述图像内每个纹理特征像素间与该图像及其周围所有纹理像素间信息的关系。

原始的 LBP 算子定义为在 3 * 3 的窗口内，以窗口中心像素为阈值，将相邻的 8 个像素的灰度值与其进行比较，若周围像素值大于或等于中心像素值，则该像素点的位置被标记为 1，否则为 0。这样，3 * 3 邻域内的 8 个点经比较可产生 8 位二进制数（通常转换为十进制数、即 LBP 码，共 256 种），就得到该窗口中心像素点的 LBP 值，并用这个值来反映该区域的纹理特征，也就是图像区域图像中包含的所有图像纹理信息。这里给出的 LBP 算子计算方式如图 3 所示。

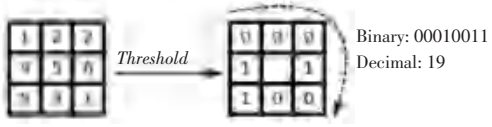



**图 3　LBP 算子计算方式**

**Fig. 3　LBP Operator calculation method**

## 2　深度学习算法

### 2.1　卷积神经网络介绍和卷积神经网络的网络模型

卷积神经网络（CNN）的优点有[3]：能够将大图片数据量下的图片有效降维成更小图片数据量；可有效保留图片特征，符合图片处理的原则。

多层网络结构及组成用到了神经网络，如图 4 所示。由图 4 可知，这是一个常见的多层卷积网络，首先是输入层、然后是隐含层，隐藏层中包括了：卷积层、池化层、全连接层，最后是输出层。对此拟做研究论述如下。


**图 4　CNN 的网络结构**

**Fig. 4　Network structure of CNN**

卷积网络模型中的图像输入层通常都可以做到只需接收到一组二维对象，卷积网络模型上的一些图像的输入层特征也大多只要预先进行一次图像输入标准化设计和处理即可，例如输入层的输入数据定义为像素，将分布到每个[0，255]像素区间上的所有原始图像的每个像素值都必须归一化地紧致分布在同一个[0，1]像素的区间。输入特征的标准化也更有利于显著提升在卷积网络学习中的学习对象的学习效率水平和学习表现。

卷积网络模型中的隐藏层中一般也包含有卷积层、池化层和全连接层等这 3 类网络最广泛常见层的构筑[4]，相较于其他任何类型网络，卷积层和池化层结构多是为卷积神经网络模型中所特有。

卷积图层法就是用某一个卷积核函数来遍历出任意的一张图片，而对于卷积图层的一种基本的计

算图的输出过程就是要用函数输入每一张图片信息,并把与其中的卷积核函数所对应出的所有元素信息进行相乘处理后,再做求和,这样经过计算输出后的结果就得到了一张较新的计算图(feature map),因此选择一个合适的卷积核,可以突显不同的图像特征。该过程示意如图 5 所示。

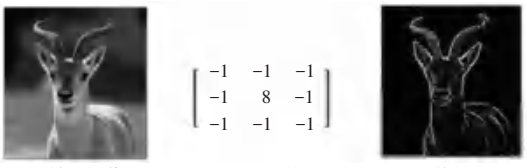



**图 5 利用卷积核提取特征图**

**Fig. 5 Extracting feature maps using convolution kernels**

在处理图像时,可以选择多个卷积核,生成不同的图像。这些不同图像可以理解为不同通道,对此可称为多核卷积。

同卷积层结构一样,池化层就是对每次计算输入元素的数据都通过一个池化的窗口元素来完成计算输入元素或输出,如图 6 所示。不同于一般卷积核层结构的特点是,卷积核层只计算输入元素与卷积核层的互相关性,池化核层可直接完成关于一个池化窗口元素输出的最大值或最小平均值的计算。

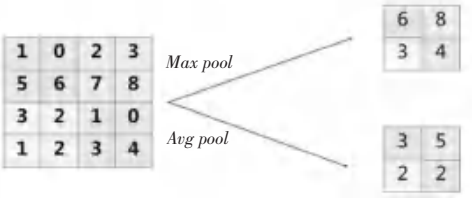



**图 6 池化层计算方法**

**Fig. 6 Pooling layer calculation method**

全连接层中的所有其他神经元都会与上一层中的所有其他神经元进行全连接,具体如图 7 所示。同时为了提高网络的整体性能,全连接层中的所有其他神经元通常只会被一个 *Relu* 激励函数激活。

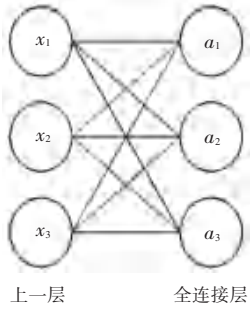



**图 7 全连接网络层的模型**

**Fig. 7 Model of fully connected network layer**

卷积神经网络图中使用的图像分类输出层中的图像上层一般都是有一个全连接的输出层,对于图像上的分类的标签问题,输出层本身也可以考虑通过使用输出层本身用到的逻辑函数或归一化指数函数(*Softmax function*),而后再输出图像的分类上的标签,如图 8 所示。

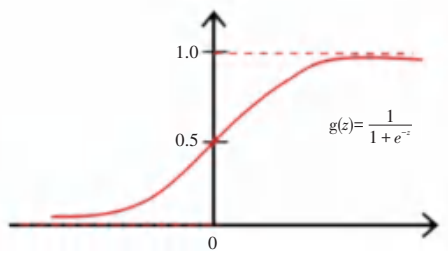



**图 8 *Softmax* 激活函数**

**Fig. 8 *Softmax* activation function**

卷积神经网络图层会在每一个隐含层的神经元中分别提取出这些图像表面上的某些局部特征,并再将其各自映射到下一个平面,技术上是通过使用 *Relu* 激励函数来使得所有这些局部特征中的每个映射点都是具有位移性质的完全不变性。每一个神经元都要与局部神经感受野相联系、并建立连接。这样的特征提取过程使得网络对输入的样本有较强的容忍能力,对图像处理也具有非常好的鲁棒性。

## 2.2 深度可分离卷积

深度可分离卷积(Depthwise separable convolution)的过程可分为 2 步。分别是:逐通道卷积(Depthwise Convolution,Dw),逐点卷积(Pointwise Convolution,Pw)。对此可做分析概述如下。

Dw 中的一个卷积核负责一个输出通道,一个输入通道也只会由其中的一个卷积核来做卷积,由此得到的计算图(feature map)输出通道需要与输入通道完全一致,即如图 9 所示。

Pw 与常规卷积 d 方法颇为相似,所要求的卷积核面积一般为 $1 \times 1 \times M$, 为最上一层的下一层的卷积通道,其中的卷积核数目与计算图(feature map)的数目相同,即如图 10 所示。

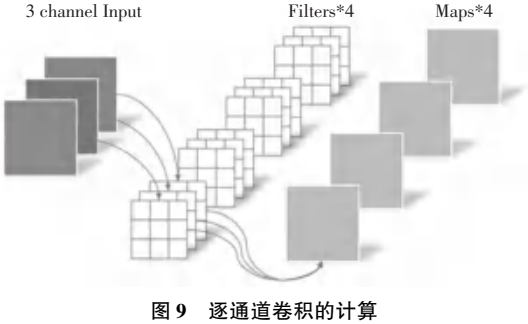



**图 9 逐通道卷积的计算**

**Fig. 9 Calculation of depthwise convolution**

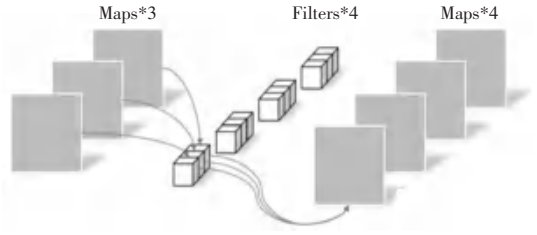



**图 10　逐点卷积的计算**

**Fig. 10　Calculation of pointwise convolution**

深度可分卷积的基本假设，是卷积神经网络中特征图的空间维和通道（深度）维是可以解耦（decouple）的。标准的卷积计算使用权重矩阵实现了空间维和通道维特征的联合映射（joint mapping），这样做的代价则是提升了计算复杂度、增加了内存开销，并引入了大量的权重系数计算。理论上，深度可分卷积通过对空间维和通道维分别进行映射、并将结果进行组合，在基本保留卷积核的表征学习（representation learning）能力的同时减少了权重系数的个数。考虑输入和输出通道数的差异，深度可分卷积的权重数约为标准卷积权重数的 10%～25%。使用深度可分卷积搭建的一些卷积神经网络、例如 Xception，在 ImageNet 数据集的图像识别任务中的表现要优于隐含层权重相同、但使用标准卷积和 Inception 模块的 Inception v3，因此研究认为深度可分卷积提升了卷积核参数的使用效率。

### 2.3　Mobilenet v1 网络模型

Mobilenet V1 是一种流水线的结构，是由深度可分离卷积构建的十分轻量的神经网络，往往会应用在移动的设备端上，因此可以通过超参数对网络模型大小做出限制，使得开发人员能够更好地使用模型。

Mobilenet 中的网络结构，则如前面所提到过的卷积层结构中除了第一层为标准卷积层结构外，其他层都是深度可分离卷积结构（Conv dw + Conv/s1），卷积结构的后一层连接了一个 7 * 7 的平均池化层，此后再通过一个全连接层，最末端在利用 *Softmax* 激活函数运算后是将全连接层的输出归一化到 0～1 中的任意一个概率值，根据概率值的高低可以得到图像的分类情况。对于 Mobile 的超参数，这里可做剖析详述如下。

（1）为了构造结构更小、且计算量更小的模型宽度因子（Width Multiplier）$\alpha$，对于深度可分离卷积层，输入的通道数 $M$ 乘上一个宽度因子 $\alpha$，变为 $\alpha M$，输出通道数变为 $\alpha N$，其中 $\alpha \in (0,1]$。此时深度可分离卷积的参数量为：$DK * DK * \alpha N + \alpha M * \alpha N = \alpha * \alpha(1/\alpha * DK * DK * N + M * N)$，计算量变为 $\alpha M * DF * DF * DK * DK + \alpha N * DF * DF * \alpha M = \alpha * \alpha(1/\alpha * M * DF * DF * DK * DK + N * DF * DF * M)$，所以参数量和计算量差不多都变为原来的 $\alpha * \alpha$ 倍。

（2）为了减少计算量，引入了第二个参数 $\rho$，称为分辨率因子。其作用是在每层特征图的大小乘以一定的比例分辨率因子（Resolution Multiplier）$\rho$，$\rho$ 改变了输入层的分辨率，所以深度可分离卷积的参数量不变，但计算量为 $M * \rho_{DF} * \rho_{DF} * DK * DK + N * \rho_{DF} * \rho_{DF} * M = \rho * \rho(M * DF * DF * DK * DK + N * DF * DF * M)$，即计算量变为原来的 $\rho * \rho$ 倍。

### 2.4　FaceNet 算法

FaceNet 是可以对人脸识别、验证、聚类等所有人脸问题方法进行系统解决的一个技术框架[5]，如图 11 所示。即能够把全部特征都放在同一个人脸特征空间里，研究人员只要致力于如何将人脸更好地映射到特征空间中即可。文中对此拟展开阐释分述如下。


**图 11　FaceNet 结构框架**

**Fig. 11　FaceNet structural framework**

（1）直接学习图像到欧式空间上点的映射，2 张图像对应特征的欧式空间上的点的距离直接表示着 2 张图像是否相似。

（2）总体网络结构：基于 GoogLeNet 或 Zeiler&Fergus 模型、CNN + Triplet 和 loss 方法，直接学习从人脸图像到紧凑的欧几里德空间的映射，提取的嵌入特征有助于实现人脸识别、验证和聚类分析等项目任务，训练的损失函数能够直接针对实际误差，端到端训练能得到更高的精度。

（3）嵌入：128 维特征，可通过直接使用网络训练优化嵌入本身。而不是像以前中间的瓶颈层的表示，是间接的分类网络。

（4）*Triplet Loss*：将正对与负对分开一个距离余量。

（5）*semi - hard*：半困难三元组样本选择方式。

（6）最小的对齐：面部周围紧密的裁剪。近期研究还尝试进行了相似性变换对齐，并没有注意到这一项设计实际上可以用来进行小幅性能提升，但却也会带来复杂性的额外略增[6]。而其他模型则需要复杂的 3D 对齐。

(7)人脸相似性验证:用来对 2 个人脸嵌入空间特征点之间的平方 $L_2$ 距离进行阈值比较,这 2 个嵌入特征空间点中人脸的平方 $L_2$ 距离值直接对应着人脸的相似性,同一人的 2 个人脸图像仅具有较小的距离值并且与不同人之间的一个人脸图像间具有相当大的距离。

(8)在 LFW 上有着 99.63%的正确率,YouTube Faces 上有 95.12%的正确率。

## 3 实验与分析

### 3.1 实验环境

本次实验中,使用的开发环境是:硬件选用了 RAM 16 G、CPU Intel(R) Core(TM) i7-1260P @ 2.1 GHz 4.7 GHZ,操作系统选用了 Windows10,开发测试软件选用了 pytorch_GPU + python3.8、pychram、Anaconda。

### 3.2 数据采集与处理

目前,互联网上已有不少采集完备的数据集,可以下载并进行数据清洗和标注,以此来获得本文研究所需的数据集。本次研究中使用的数据集为 CASIA-WebFace[7],该数据集源主要来自于 IMBb 网站,包含 1 w 个人的近 500 w 张图片。与此同时,又通过相似度聚类方法滤掉了其中的一部分噪声。CAISA-WebFace 的数据集源和 IMDb-Face 几乎是完全是一样的,唯一的不同点就是在数据集清洗后,CAISA-WebFace 的图片会相对少一些,而且噪声也相对较少,故适合用来作为训练的数据。具体步骤可做阐释如下。

(1)数据标注。通过以下代码,生成对应的文件夹下的图片标签,并写入 cls_train.txt 文件。

```
datasets_path = "datasets/datasets"
types_name = os.listdir(datasets_path)
// 获取该目录下的文件夹中的照片
types_name = sorted(types_name)
// 按照升序排序
list_file = open ('cls_train.txt','w')
for cls_id, type_name in enumerate(types_name):
photos_path = os.path.join(datasets_path,
type_name)
if not os.path.isdir(photos_path):
continue
photos_name = os.listdir(photos_path)
for photo_name in photos_name:
list_file.write(str(cls_id) +";" + '%s'%(os.
path.join(os.path.abspath(datasets_path),
type_name, photo_name)))
list_file.write('\n')
list_file.close()
```

(2)构建网络并训练。通过构建一个 Mobilenets 来进行模型训练,这可能是基于一种超轻量的深度级卷积神经网络,该网络核心部分为深度级可分离的卷积。使用的算法为 FaceNet 人脸特征识别算法,这个识别算法就是通过抽取人脸图像上的人脸某一层的特征,学习到一个从人脸图像到欧式空间图像的编码识别方法,再基于这个图像编码来做人脸特征识别。下面将给出具体的处理流程。

① 构建第一个网络结构。研究中用到的代码如下:

```
def conv_bn(inp, oup, stride = 1):
return nn.Sequential(
nn. Conv2d(inp, oup,3, stride,1, bias =
False),
nn.BatchNorm2d(oup),
nn.ReLU6()
)
```

该网络结构可用来加速网络模型的推理速度,通过 *Sequential* 堆叠网络,第一层为 *Conv*2*d* 卷积层、第二层为 *BN* 层、第三层为 *ReLU*6 激活层,分析可知 *Conv*2*d* 和 *BN* 都是线性运算,所以该网络结构融合后就减少了推理时间。

② 构建 Dw 深度可分离卷积网络。研究中用到的代码如下:

```
def conv_dw(inp, oup, stride = 1):// 深度可分离卷积<br>return nn. Sequential(
nn.Conv2d(inp, inp,3, stride,1, groups = inp,
bias = False),
nn.BatchNorm2d(inp),// 标准化
nn.ReLU6(),// 激活函数
//1×1 简单卷积
nn.Conv2d(inp, oup,1,1,0, bias = False),
nn.BatchNorm2d(oup),
nn.ReLU6(),
)
```

这个网络的构建就是深度可分离卷积层的构建方法,用 *group* 参数来选择。

③ 构建 MobilenetV1 模型。研究中用到的代码如下:

```
self.stage1 = nn.Sequential(
```

*conv_bn*(3,32,2),
*conv_dw*(32,64,1),
*conv_dw*(64,128,2),
*conv_dw*(128,128,1),
*conv_dw*(128,256,2),
*conv_dw*(256,256,1),)
*self.stage2* = *nn.Sequential*(
*conv_dw*(256,512,2),
*conv_dw*(512,512,1),
*conv_dw*(512,512,1),
*conv_dw*(512,512,1),
*conv_dw*(512,512,1),
*conv_dw*(512,512,1),)
*self.stage3* = *nn.Sequential*(
*conv_dw*(512,1 024,2),
*conv_dw*(1 024,1024,1),)

通过第一个 *conv_bn* 层提取特征，此后用的都是 *conv_dw* 深度卷积。

④ 用主干特征提取网络获得特征层。研究中用到的代码如下：

*self.avg* = *nn.AdaptiveAvgPool2d*((1,1))
*self.Dropout* = *nn.Dropout*(1 - *dropout_keep_prob*)
*self.Bottleneck* = *nn.Linear*(*flat_shape*, *embedding_size*,*bias* = *False*)
*self.last_bn* = *nn.BatchNorm1d*(*embedding_size*, *eps* = 0.001, *momentum* = 0.1, *affine* = *True*)
*x* = *self.backbone*(*x*)
*x* = *self.avg*(*x*)
*x* = *x.view*(*x.size*(0), -1)// *resize*
*x* = *self.Dropout*(*x*)// 丢弃部分神经元
*x* = *self.Bottleneck*(*x*)// 全连接
*x* = *self.last_bn*(*x*)

经过一个平均池化、一个 *reshape*、再经过一个 *Dropout* 后，又经过全连接，最后标准化输出一个 128 的特征向量。

⑤ $L_2$ 的标准化。在进行标准化运算前一般都会计算范数，对特征向量中的每个元素绝对值取平方和后、再去求其开方，$L_2$ 的标准化过程就是先计算出每个元素 /$L_2$ 的范数。此时用到的代码为：

*x* = *F.normalize*(*x*, *p* = 2, *dim* = 1)

代码中只需 *p* = 2，就表示使用 $L_2$ 标准化。

⑥ 分类器的构建。此时用到的代码为：

*self.classifier* = *nn.Linear*
(*embedding_size*, *num_classes*)

结合使用了 *Cross - Entropy Loss* 和 *Triplet Loss* 作为一种总体上的 *Loss*，单一使用 *Triplet* 的 *Loss* 可能会导致人脸网络的收敛困难，*Triplet* 的 *Loss* 则可同时用于 2 种不同类型人脸特征向量间在欧几里得的空间距离范围上的扩张，同一个人的 2 个人脸图像的特征向量间的距离由欧几里得缩小。*Cross - Entropy Loss* 用于人脸分类，加速了 *Triplet Loss* 的收敛。接下来，可开始进行模型训练，如图 12 所示。


**图 12　进行模型训练**

**Fig. 12　Conducting model training**

### 3.3　实验结果与分析

选取 2 张人脸图像，用训练好的模型来对其进行验证，如图 13 所示。FaceNet 的阈值设置为 1.1。

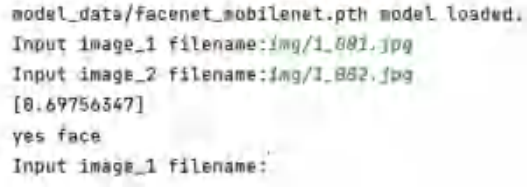



**图 13　进行图像验证**

**Fig. 13　Performing images verification**

通过 FaceNet 预测显示 *Distance* 为 0.698，比预测前设置的阈值小，所以是同一张人脸，如图 14 所示。

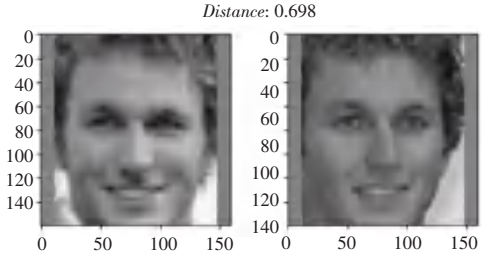



**图 14　相同人脸预测**

**Fig. 14　Same face prediction**

再选取 2 张人脸图像，通过 FaceNet 预测显示 *Distance* 为 1.337，超出预测前设置的阈值，所以是不同的人脸，如图 15 所示。

（下转第 143 页）