

基于卷积神经网络的人脸识别

王荣平

(广州华立学院 511325)

摘要：随着人工智能技术不断发展，人脸识别技术应用愈加广泛。本文首先介绍了卷积神经网络结构，对MTCNN算法进行简要阐述，选用LFW人脸数据集开展实验，讨论卷积神经网络模型设计，根据模型设计卷积神经网络的训练和优化方法，包括梯度下降、反向传播、正则化等。经过实验和对比分析，证明了基于卷积神经网络的人脸识别技术在准确率和鲁棒性上的优势，本文对于理解和应用基于卷积神经网络的人脸识别技术具有一定意义。

关键词：卷积神经；人脸识别；模型

一、引言

随着人工智能技术的不断发展，人脸识别技术已成为当今研究的热点之一。人脸识别技术可以应用于安防、身份验证、社交媒体等多个领域，具有广泛的应用前景和市场需求。然而，在实际应用中，人脸识别技术面临着许多挑战，如光照、姿态、表情、遮挡等问题，这些问题会导致人脸识别的准确率和鲁棒性大幅降低。为了提高人脸识别技术的准确率和鲁棒性，近年来，基于卷积神经网络的人脸识别技术成了研究的热点之一。卷积神经网络具有自动学习特征、适应性强等特点，在图像识别、语音识别等方面都取得了很好的效果。因此，将卷积神经网络应用于人脸识别技术中，可以提高识别准确率和鲁棒性，具有重要的研究和应用价值^[1]。

二、卷积神经网络结构

卷积神经网络(CNN)是一种专门用于处理图像、视频、自然语言处理等高级数据的深度学习模型。它的基本原理是对输入数据进行卷积运算，从而提取出不同层次的特征，再通过池化操作降低特征的维度，最终使用全连接层进行分类或回归等任务。CNN的基本组成包含卷积层、池化层、激活函数、全连接层等。如图1，这是卷积神经网络的构造。

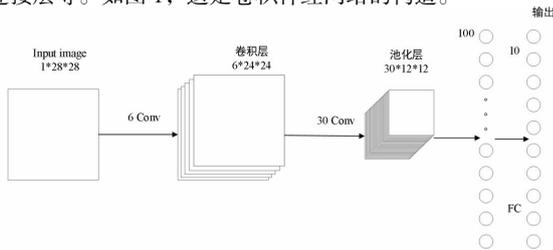


图1 卷积神经网络结构

2.1 CNN 基本结构

卷积层主要负责提取图像中的局部特征。卷积操作类似一种滑动窗口的操作，通过将卷积核与输入数据进行卷积运算，提取出局部的特征信息。在卷积运算的过程中，卷积核的参数会被训练得到，从而实现对不同特征的识别。

池化层基本操作用于降低特征的维度，减少网络中的参数数量，同时还可以增强模型的鲁棒性。常见的池化操作包括最大池化和平均池化。最大池化操作选取卷积核中的最大值作为池化后的特征，而平均池化则是将卷积核中的特征值求平均。

激活函数用于对神经网络中的输出进行非线性映射，从而增强网络的表达能力。常见的激活函数包括 Sigmoid、tanh、ReLU 函数等^[2]。相比于 sigmoid 和 tanh 等函数，ReLU 函数具有稀疏性、计算速度快、有效避免梯度消失等特点，使神经网络训练速度更快。它的表达式如下：

$$f(x) = \max(0, x) \tag{1}$$

全连接层位于卷积神经网络隐层的最后一部分，通过激励函数将其输出的特征图进行展开成一维向量，然后输入到全连接层进行分类预测或回归等任务。由于全连接层具有很强的拟合能力，容易出现过拟合现象，因此通过引入 Dropout 技术随机抑制一部分神经元。

CNN 通过多个卷积层、池化层、激活函数、全连接层的组合，实现了对输入数据的特征提取和分类。其中，卷积层和池化层的堆叠可以提取出更加丰富的特征信息，从而提高网络的准确率和鲁棒性。

2.2 L2 范数标准化

自 Wright 提出 SRC 方法，稀疏表示方法在人脸识别中越来越受到关注。同时，由于 L₀ 范数和 L₁ 范数最小化技术不断发展，稀疏表示方法在人脸识别领域取得了良好的效果，很多学者对其做了大量研究并获得一定成就，提出了一种更加准确、快速、鲁棒性的 L₂ 范数方法，Ortiz 等人应用 L₂ 范数最小化方法减少字典原子数量，通过处理后的字典求解 L₂ 范数最小化，进而在保证 SRC 识别率的基础上，提高了运行速率 200 倍，L₂ 范数的计算公式如(2)所示：

$$\|x\|_2 = \sqrt{\sum_i^n (x_i)^2} \tag{2}$$

2.3 损失函数

损失函数在 CNN 训练过程中通过测试，并根据预测的结果，进而判断模型预测能力的性能。经过大量的研究，很多研究人员提出了各种不同的损失函数，如 Softmax loss、Center Loss 和 Triplet Loss 等^[2]。

2.3.1 Softmax Loss

当二分类问题扩展为多分类问题时，Softmax 将 CNN 网络获得的多个值进行回归处理，使得函数输出既概率总和为 1，各值介于 0~1 之间，然后把输出值最大的神经元对应的类别作为最终的识别结果。Softmax 回归的数学表达式如(3)所示。

$$P(Y_i = j|X_i, \beta) = \frac{\exp(X_i^T \beta_j)}{\sum_{k=1}^K \exp(X_i^T \beta_k)} \tag{3}$$

2.3.2 交叉熵损失函数

交叉熵损失函数广泛应用于多分类问题的优化中，用于衡量和优化输出概率分布与目标概率分布之间差异。在分类任务中，经常采用 softmax 激活函数+交叉熵损失函数，因为交叉熵描述了两个概率分布的差异，然而神经网络输出的是向量，并不是概率分布的形式，所以需要 softmax 激活函数将一个向量进行“归一化”成概率分布的形式，再采用交叉熵损失函数计算 loss。交叉熵损失函数的数学表达式如(4)所示。

$$\text{Loss}(x, \text{label}) = -\log \frac{e^{x^{\text{label}}}}{\sum_{j=1}^N e^{x^j}} \tag{4}$$

一、相关算法运用

MTCNN 是一种用于多尺度对象检测的卷积神经网络算法^[3]，常用于人脸识别、人脸检测等。它由三个级联的卷积神经网络(P-Net、R-Net、O-Net)组成，每个网络负责人脸/非人脸检测、边框回归、面部特征点位置。具体训练包括以下：

1. 人脸检测分类、2.边框回归、3.人脸面部特征点坐标、4.多源训练、5.在线挖掘困难样本、6.训练数据。

由于MTCNN算法采用了级联式结构以及多尺度检测策略，其检测速度比传统的检测方法更快，检测效果也更加准确。

MTCNN 还适应多种复杂的场景,如遮挡、表情变化、尺度变化、光照不均等,因此 MTCNN 在实际应用中有着广泛的应用。

三、实验以及分析

3.1 数据集介绍及选择

在人脸识别中,通过训练好的模型可以更准确提取从未出现的人脸特征信息,因此数据集的质量和数量对于人脸识别的效果至关重要。目前最常用的人脸识别数据集包括 LFW、YTF、CelebA、MegaFace2 等。这些数据集包含了大量的人脸图像和标签信息,可以用于训练和测试卷积神经网络。在这些常用的人脸数据集中,LFW 图片可以通过网络获得,能很好地评价人脸识别的性能。

3.2 实验模型

在本实验中,我们使用 LFW 数据集进行人脸识别任务。该数据集包含 13,233 张人脸图像和 5,749 个人的标签,图像来自互联网,存在光照、姿态、年龄等方面的变化。

在数据预处理阶段,我们将图像像素值缩放到 0~1 之间,并对原始图像进行人脸检测和对齐操作。我们使用 MTCNN 人脸检测算法对图像进行人脸检测,并使用 dlib 库中的人脸对齐算法将人脸对齐此外,我们还对图像进行数据增强操作,包括随机旋转、平移、缩放和水平翻转等操作,以扩充训练数据集大小。其中 MTCNN 人脸检测和数据增强主要代码:

```
# MTCNN 人脸检测
def detect_faces(image):
    detector = MTCNN()
    faces = detector.detect_faces(image)
    return faces
# 数据增强
def data_augmentation():
    datagen = ImageDataGenerator(
        rotation_range=20,
        width_shift_range=0.2,
        height_shift_range=0.2, #随机垂直平移
        zoom_range=0.2, # 随机缩放
        horizontal_flip=True #随机水平翻转
    )
    return datagen
```

在实验中使用经典模型 VGG16 建立人脸识别模型,去掉最后一层全连接层,添加新的全连接层和 softmax 层来进行人脸识别。此外,针对可能出现过拟合现象还使用 L2 正则化和 dropout 技术规避,添加新的全连接层和 softmax 层主要代码:

```
x = Dense(512, activation='relu', kernel_regularizer='l2')(x)
x = Dropout(0.5)(x)
predictions = Dense(2, activation='softmax')(x)
model = Model(inputs=base_model.input, outputs=predictions)
return model
```

在模型训练阶段,使用 Adam 优化器和交叉熵损失函数进行训练,并在每个 epoch 结束时进行模型保存。为了避免过拟合,运用了早停法和学习率调整技术。回调函数关键代码:

```
checkpoint=ModelCheckpoint('face_recognition_model.h5',
monitor='val_loss', save_best_only=True)
earlystop = EarlyStopping(monitor='val_loss', patience=10)
reduce_lr = ReduceLRonPlateau(monitor='val_loss', factor=0.1,
patience=5, min_lr=1e-7)
```

在模型评估阶段,加载训练好的模型,并在测试集上进行评估,通过计算测试集上的准确率和混淆矩阵来评估模型的性能。模型测试关键代码:

```
def evaluate_model(test_generator):
    model = load_model('face_recognition_model.h5')
    y_pred = model.predict(test_generator)
```

```
y_pred = np.argmax(y_pred, axis=1)
y_true = test_generator.classes
acc = accuracy_score(y_true, y_pred)
conf_matrix = confusion_matrix(y_true, y_pred)
return acc, conf_matrix
```

3.3 实验结果

本次实验模型采用了基于 LeNet-5 卷积神经网络改进型的人脸识别新卷积神经网络模型,表 1 中的模型是不同卷积核数的卷积神经网络模型,表 2 反映在学习速率为 0.05 时的各模型准确率。

表 1 不同卷积核数的卷积神经网络模型

模型	1 层卷积层卷积核数	2 层卷积层卷积核数
模型 1	20	30
模型 2	30	30
模型 3	40	30
模型 4	50	30
模型 5	60	30
模型 6	20	50
模型 7	30	50
模型 8	40	50
模型 9	50	50
模型 10	60	50

表 2 同卷积核数的卷积神经网络模型的准确率

模型	准确率
模型 1	93.5%
模型 2	82.6%
模型 3	93.0%
模型 4	86%
模型 5	88.5%
模型 6	92.5%
模型 7	95%
模型 8	95%
模型 9	95%
模型 10	95%

通过上述实验发现,当学习速率为 0.05,第一层卷积层卷积核数为 30,第二层卷积层卷积核数达到 50 时,能够构建一个针对 LFW 人脸数据库高识别率的新型卷积神经网络模型。

五、结语

本文基于卷积神经网络的人脸识别技术进行了深入的探讨和研究。通过实验研究,我们发现基于卷积神经网络的人脸识别技术具有很高的准确性和鲁棒性,但也存在一些问题和挑战,如数据质量问题、样本不均衡性、跨年龄和跨人种识别问题等。针对以上问题,提出一些改进方法和思路^[4],相信随着技术的不断发展和完善,基于卷积神经网络的人脸识别技术将在更多的应用场景中获得更广泛的应用和推广。

参考文献:

[1]Patricia M ,Daniela S ,Oscar C . Comparison of optimization algorithms based on swarm intelligence applied to convolutional neural networks for face recognition [J]. International Journal of Hybrid Intelligent Systems,2023,18(3-4).
 [2]李军,李明.改进多尺度卷积神经网络的人脸表情识别研究[J].重庆邮电大学学报(自然科学版),2022,34(02):201-207.
 [3]蓝雯飞,张盛兰,朱容波等.基于改进 MTCNN 的人脸检测算法[J].中南民族大学学报(自然科学版),2020,39(06):637-641.
 [4]常英丽,张雪华,刘强.改进的卷积神经网络模型在人脸识别中的应用[J].电子制作,2022,30(24):42-45.DOI:10.16589/j.cnki.cn11-3571/tn.2022.24.010.