**人脸识别**

**22121883 刘远航 22121347 叶思清 22121926 林升华**

**摘 要：**本文旨在探讨主成分分析（PCA）与K最近邻（KNN）算法在人脸识别领域的应用。通过对data数据集进行实验，我们展示了使用PCA降维和KNN分类器相结合的方法在人脸识别中的有效性。实验结果表明，该方法在人脸识别任务中取得了良好的准确率和性能表现。

**关键词：**人脸识别 PCA降维 KNN算法

**1. 项目背景和意义**

人脸识别技术在当今社会中具有广泛的应用前景，涵盖了安防监控、人机交互、身份验证等领域。随着计算机视觉和机器学习技术的发展，基于人脸图像的识别技术也取得了长足的进步。PCA和KNN作为经典的机器学习算法，在人脸识别任务中具有重要的作用。

本项目旨在利用PCA对人脸图像进行降维处理，然后使用KNN算法对降维后的特征进行分类，从而实现人脸识别任务。通过这种方法，可以有效地提取人脸图像的重要特征，并利用KNN算法进行快速准确的识别，具有较高的实用价值和研究意义。

**2. 项目实现过程**

**2.1 数据预处理**

我们组在数据预处理的过程中，首先根据文件形式读取数据集，使用Python的os模块读取data数据集中的人脸图像数据。每个子文件夹代表一个不同的人脸。

然后进行图像处理，对每张图像进行灰度转换和大小调整。利用PIL库中的Image.open()函数读取图像，然后使用convert('L')方法将其转换为灰度图像，并通过resize()方法将图像大小调整为32x32像素。

**2.2 特征提取**

我们小组使用主成分分析（PCA）算法对灰度图像进行降维处理。将灰度图像转换为二维数组表示，并将其展平为一维数组，然后利用PCA类从sklearn库拟合PCA模型，并使用transform()方法将训练集和测试集的特征数据进行降维。

下面是具体实现过程：

①计算主成分方差占比：首先进行PCA分析，得到每个主成分（特征向量）对应的方差值。然后计算每个主成分方差占总方差的比例，即主成分方差除以所有主成分方差之和。  
 ②计算累计方差：将每个主成分的方差占比依次累加，得到累积方差值。这反映了每个主成分对数据方差的贡献程度，以及前n个主成分能保留的数据信息量。  
 ③选择合适的维数：根据累积方差的曲线图，选择一个合适的累积方差阈值（通常为80%至95%之间）。找到曲线上第一个超过阈值的点，该点对应的主成分数量即为保留的维数。  
 ④降维处理：根据选择的维数，保留对应的主成分，进行数据降维处理。这样可以在尽可能保留原始数据信息的前提下，减少数据维度，提高计算效率和降低噪声影响。

**2.3 模型训练**

提取完特征之后，我们利用K最近邻（KNN）算法进行分类器的训练。通过KNeighborsClassifier类从sklearn库建立KNN模型，使用降维后的训练集数据对模型进行拟合，这一步将使用训练数据来训练模型，使其学习到数据的特征和标签之间的关系。我们调节KNN算法的k值以及其他参数，观察各个参数对实验准确率的影响。

**2.4 模型评估**

我们组最后使用测试集对训练好的模型进行性能评估，训练集占总数据集的20%，计算准确准以评估模型在人脸识别任务中的表现。

**3. 实验结果及分析**

本次实验所用的平台工具情况如下：

编程语言和环境： Python3.11 PyCharm

库和框架： NumPy、Pandas、scikit-learn等用于数据处理和机器学习的库，以及PIL（Python Imaging Library）用于图像处理。

实验数据： data数据集，训练集比测试集为8:2。

**3.1计算PCA降维最终降到几维--累计方差法**

我们通过调节阈值得到了不同的PCA降维的维数，进而得到不同的准确率。表1是我们得到的实验数据。

表 1 实验结果

|  |  |  |
| --- | --- | --- |
| 阈值 | PCA降维 | 准确率 |
| 0.8 | 21 | 0.8125 |
| 0.85 | 32 | 0.825 |
| 0.9 | 51 | 0.8357 |
| 0.95 | 91 | 0.8375 |

对实验结果进行处理，得到下图1。

图1实验结果折线图

分析：当阈值在0.9以上时，PCA降维得到的人脸识别模型的准确率均较高。但是，对比阈值为0.9和0.95两种情况，我们可以发现：虽然将阈值从0.9提升到0.95时，准确率熊0.8357提高到了0.8375，但是PCA降维的维数却从51维提升到了91维度。因此综合分析，我们将累计方差阈值设定为0.9（也即选取的特征向量能够代表原图片90%的特征）。  
 通过累积方差法计算PCA降维的维数，可以根据数据的特点和需求来选择合适的维数，从而实现有效的数据降维处理。

**3.2 KNN参数的设定**

①weights（权值）：weights参数用于指定K个最近邻样本在预测时的权重分配方式

uniform：即均匀权重，表示所有最近邻样本的权重相同，不考虑距离远近。在这种情况下，每个最近邻样本对预测的贡献是相等的。  
 distance：即距离权重，表示最近邻样本的权重与其距离成反比。距离越近的样本权重越大，距离越远的样本权重越小。这种方式能够更好地反映样本之间的相似度。

显然distance优于unifrom。

②algorithm（计算最近邻居的算法）

brute：暴力搜索算法，即简单的线性扫描。它会计算所有样本点之间的距离，然后选择最近的K个邻居。虽然在小数据集上表现良好，但在大数据集上计算复杂度较高。  
 kd\_tree：KD树算法，通过构建KD树来实现快速最近邻搜索。KD树是一种二叉树数据结构，能够在高维空间中快速进行最近邻搜索，适用于维度较低的数据集。  
 ball\_tree：球树算法，通过构建球树来实现快速最近邻搜索。球树是一种多维空间的数据结构，能够有效地减少计算距离的次数，适用于高维数据集。  
 我们对比了三种算法，发现三种算法对应的人脸识别分类器模型的准确率相同。分析原因：由于数据集较小，且特征向量维数较小，因此kd\_tree和ball\_tree的优势没有充分体现。

③P值（计算距离的闵可夫斯基距离的指数参数）

当p=1时，闵可夫斯基距离等同于曼哈顿距离（L1范数），即各个坐标数值差的绝对值之和。当p=2时，闵可夫斯基距离等同于欧式距离（L2范数），即各个坐标数值差的平方和再开根号。  
 在scikit-learn库中，KNN算法的参数p可以通过设置来指定使用的距离度量。通常情况下，当p=1时，使用曼哈顿距离更适合处理特征维度较高的数据；当p=2时，使用欧式距离更适合处理特征维度较低的数据。最终通过测试我们选择采用欧氏距离。

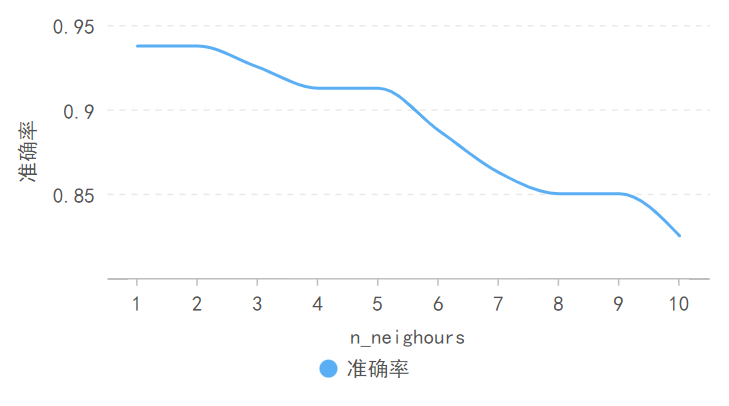
④n\_neighours(最近邻居的数量)

K最近邻（KNN）算法中的参数n\_neighbors表示用于进行分类或回归预测时考虑的最近邻居的数量。在KNN算法中，当需要对一个新样本进行预测时，算法会找出与该样本最近的n\_neighbors个训练样本，并根据它们的标签值（对于分类问题）或数值（对于回归问题）来进行预测。  
 增加n\_neighbors的值会增加模型的复杂度和计算成本，可能会导致模型在训练集上表现较好，但在测试集上过拟合的风险也会增加。相反，减少n\_neighbors的值可能会导致模型在训练集上表现较差，但在测试集上泛化能力可能更强。

下表2是我们更改n\_neighbors的值的得到的实验结果：

表 1 实验结果

|  |  |
| --- | --- |
| n\_neighbors | 准确率 |
| 1 | 0.9375 |
| 2 | 0.9375 |
| 3 | 0.925 |
| 4  5  6  7  8  9  10 | 0.9125  0.9125  0.8875  0.8625  0.85  0.85  0.825 |



图表 2 实验结果图

从上图2中我们可以发现随着n\_neighours的增大，模型的准确率逐渐降低。因此最终取n\_neighours的值为1.

最终我们模型的准确率为0.9500。