****

课程设计报告书

**题目：基于PCA和BP神经网络的**

**人脸性别判断**

**学 院 自动化科学与工程学院**

**专 业 自动化**

**学生姓名 贺亦欣**

**学生学号 201730644216**

**指导教师 罗家祥**

**课程编号 046101351**

**课程学分**

**起始日期 2019.12**

|  |  |
| --- | --- |
| 教  师  评  语 | 教师签名：  日期： |
| 成  绩  评  定 |  |
| 备  注 |  |

**基于PCA和BP神经网络的人脸性别判断**

**一、选题背景**

人脸识别一直是模式识别技术实际应用的一个热门领域，计算机识别人脸代替人工识别，可以为生活带来极大的便利。本课题利用PCA对数据集中的人脸样本进行降维，然后训练BP神经网络对样本进行分类，不断优化网络结构，使得分类器对测试样本也有较好的分类效果。

**二、方案论证(设计理念)**

MATLAB作为一款强大的科学计算工具，集成了大量的工具，包括图像处理函数、机器学习工具箱、神经网络工具箱、GUI开发环境等，有利于快速求解和科学验证。本次课设利用MATLAB作为主要的编程环境，对Kaggle上下载的人脸数据集进行筛选、标注后训练性别分类器。同时设计了GUI界面对单张人脸进行识别，实现了更加直观的人机交互。

人脸性别判断的工作流程如下：

|  |
| --- |
| 图 1 人脸识别流程 |

一张照片的像素通常上万，若不进行降维，将会给计算机带来很大的运算量，同时过多的特征也不利于分类器的实现。因此十分有必要对样本进行特征提取。本课设采用的PCA（主成分分析法）常用于人脸识别，能够有效地降低特征维数。

对于分类器的设计，首先利用MATLAB的Classification Learner 工具箱，利用决策树、SVM以及KNN法分别对样本进行了分类，其中Medium Gaussian SVM的识别准确率最高，达到了87.6%，但由于其泛化性能不强，最终决定采用神经网络构建分类器。

BP神经网络是一个很强劲的分类算法，其具有很强的非线性能力和泛化的能力，在人脸识别的分类器中，是一款优势很大的分类器。但神经网络训练中容易陷入局部最优解，神经网络的结构和初始权值对分类效果有重要影响，为进一步提高识别准确率，我在神经网络的结构、参数设置以及权重初始化方面进行了一些探索，最终确定了一个分类结果较为满意的神经网络结构。

**三、过程论述**

1. 数据集预处理

原始数据集中存在部分样本会对分类器的训练带来不利影响，应当预先提出。除此之外还要进行图像的预处理，以便计算机的存储和运算。对数据集进行的预处理如下：

Step1:剔除不良样本个体，对样本重新标注。

|  |
| --- |
| 图 2 部分原始样本集 |

如图2所示，原始样本集中部分样本并不适合用于分类器的训练，例如帽子、墨镜、下巴附近的手，这些并不属于脸部本身的特征会对分类造成干扰，应当剔除。进行样本剔除与重标注后用于训练的样本男女人脸图像各有730张，如图3所示。

|  |
| --- |
| 图 3 筛选后的部分样本 |

Step2:读取图像，将RGB图像转换为灰度图。

Step3:将图像大小设为112\*92即10304个像素点，对应的特征维数为10304。

Step4:将图像信息以double类型的数值储存在1460\*10304的矩阵TrainFaceContainer中，每行对应一个样本。

（二）PCA人脸特征提取

* 1. PCA的基本原理

主成分分析法的主要原理是从一组特征中计算出一组按重要性从小到大排列的新特征，它们是原有特征的线性组合，并且相互之间是不相关的。PCA技术是将高维的数据通过投影的方式降至低维度的平面上，即乘以少量向量本身维数的基，来达到降维的效果。在PCA中，原样本特征构成的矩阵为A，由主成分特征向量构成的矩阵为V，降维后的样本维数为k，降维后的样本矩阵为pcaA。

* 1. 利用PCA对人脸特征降维

利用快速PCA算法对人脸进行特征提取的主要步骤如下：

Step1: 载入样本数据

Step2: 计算样本均值

Step3: 计算协方差矩阵的转置

Step4: 计算的前个特征值和特征向量

Step5: 计算协方差矩阵的特征向量

Step6: 特征向量归一化为单位特征向量

Step7: 原样本投影降维至维

经过多次实验，为使得分类尽量准确并且训练速度较快，最终选择作为降维后的特征维数。本征脸如图4所示。

|  |
| --- |
| 图 4 K=60时的本征脸 |

（三）BP神经网络分类器设计

2.1 网络结构的确定

首先，已知神经网络输入层节点数为60，输出层节点数为2。先从单层隐节点出发，利用经验公式计算得出隐节点数为。从11开始逐渐增加隐节点个数至20，每个不同的隐节点下训练五至十次以避免陷入局部最小值，保存训练效果最好的网络，对应的节点数为19。在第一隐层节点数为19的基础上再逐一地增加隐含层的数目，按照类似的方法确定每一层的节点个数。最终确定神经网络的结构为三个隐含层，各层节点数分别为19，6，6，如图5所示。

|  |
| --- |
| 图 5 BP神经网络结构 |

* 1. GA算法优化神经网络初始权值

利用MATLAB神经网络函数库进行网络的构建时，初始权重的选举是一个随机的过程，这个随机的权重对训练结果有较大的影响。如果初始化的权值不合适，就可能导致学习时间过长、陷入局部最优而无法逼近全局最优等一系列问题。遗传算法是模拟自然界种群行为的一种搜索算法，可以用于解决许多非线性的函数优化问题。而神经网络权值初始化可以看作是一个非线性函数的优化问题，可以采用遗传算法解决。遗传算法具有很强的全局寻优能力，可以使用遗传算法求解最优初始权值，使得训练结果逼近全局最优。GA算法流程如图6。

|  |
| --- |
| 图 6 GA算法流程 |

* 1. 网络其他参数的设置

网络中其他参数的设置主要有学习率、训练目标及训练函数等。由经验参数与多次实验，最终确定较为合适的参数设置学习率lr=0.01，训练目标为0.001，训练函数为trainrp。

（四）GUI交互界面设计

图形化的人机交互界面能够让人更加直观地进行单张照片的测试。本课题中利用MATLAB App Designer进行了GUI的设计。整体的UI界面如图7所示。

|  |
| --- |
| 图 7 测试UI界面 |

如图所示，整个界面由一个坐标区、一个按键区和一个结果显示区构成。当点击按键’Choose a photo’时，弹出文件夹界面，使用者在文件夹中选择一张图片，坐标区显示被选中的图片。点击’Test’按键时，将图片进行预处理、降维等操作，将降维后的信息输入神经网络，得出测试结果并在结果文本区中显示，完成整个识别过程。GUI工作流程如图8。

|  |
| --- |
| 图 8 GUI工作流程 |

**四、结果分析**

训练及测试效果如图9，10，11，12所示

|  |  |
| --- | --- |
| 图 9 训练过程误差曲线 | 图 10 训练过程梯度变化 |
| 图 11 预测输出与实际值的相关度 | 图 12 神经网络对280张测试照片预测的准确率 |

由图可知，训练集的准确率达到了88.216%，测试集的准确率为81.468%，两者相差不大，说明该网络的泛化性能较好。不足的地方在于识别准确率仍有待进一步提高。在这之前我尝试了用GA算法对输入节点为40的单层神经网络进行优化，但GA算法求解最优权值用时太长（大约耗时八小时），带来了一定的优化效果但不是非常显著，故最终的神经网络没有采用GA算法优化。若结合GA算法进行进一步的调试与实验应该会有更好的分类效果。由图12可看出该网络对女性的判断正确率较高，达到了92.14%，而对男性的判断正确率仅有67.86%，这可能是因为数据集中女性照片比男性的质量高，更能反应面部特征。除此之外，实验中用到的数据集表情、角度、光照程度各异，并且数据集大小有限，一定程度上给男女面部不同特征的区分造成了困难。

**五、课程设计总结**

通过本次课设，我掌握了利用PCA对图像进行特征提取的方法、BP神经网络的构建和调试、对输出结果的分析和改进，以及MATLAB GUI的设计与调试技巧。课设的关键问题在于分类器的设计和分类准确率的提升，但本次课设分类的准确率并不是很高，经过分析，我认为可以从以下几点着手改进：

1. **采用更加适合图像识别的分类器，如CNN等。**

目前图像识别采用的大多数是深度学习的方法，其中以卷积神经网络为代表的分类器对于图像、语音等具有空间、时间连续性的对象分类具有独特的优势。在已有的CNN框架上，再根据实际需求进行一些结构和超参数的调整，这样可以更快地构建准确率让人满意的分类器。

1. **精心选择设计训练样本集。**

采用更多的训练样本，同时使样本图像大小、光照条件等除面部特征以外的条件尽可能一致。这样进行PCA降维时才能更好地提取能够区分男女性别的面部特征。除此之外，本次课设的训练人脸大多数是欧美人和印度人，这对网络泛化性能的提升也有一定的限制。

1. **尝试采用其他优化算法对神经网络的初始权值进行优化。**

GA优化算法之所以难以运用再最终的神经网络上的一个重要原因是其求解速度过慢，难以实时对不同的网络结构进行优化并进行结果对比。若要对初始权值进行优化，应当先对比分析各种优化算法的复杂度、优化效果等，进行综合的权衡考虑，选择最为合适的优化算法进行实验。

**附录**

**ReadTrainFaces.m**

****

**ReadTestFaces.m**

****

**fastPCA.m******

**visualize\_PCA.m**

****

**GA.m**

****

**GAtotalAcu.m**

****

**network\_train.m**

****

**testresult.m**

****

**gender\_recognition.mlapp**

classdef gender\_recognition < matlab.apps.AppBase

% Properties that correspond to app components

properties (Access = public)

UIFigure matlab.ui.Figure

GenderRecognitionPanel matlab.ui.container.Panel

ChooseaphotoButton matlab.ui.control.Button

TestButton matlab.ui.control.Button

ResultTextAreaLabel matlab.ui.control.Label

ResultTextArea matlab.ui.control.TextArea

UIAxes matlab.ui.control.UIAxes

end

properties (Access = private)

testpath='dataset\test'

end

methods (Access = private)

function results = test(app,image)

%用于将图像特征输入神经网络，得出判断结果并输出在UI界面的文本区中

load PCA.mat;

load net27.mat;

testFace=zeros(1,10304);

testFace(:)=image(:)';

testpcaA=testFace\*V;

testpcaA=testpcaA';

testAn=sim(net,testpcaA);

out=find(testAn==max(testAn));

if out==1

results='male';

elseif out==2

results='female';

end

app.ResultTextArea.Value=results;

end

end

methods (Access = private)

% Callback function: ChooseaphotoButton

% 回调函数，用于用户在文件夹中选择一张照片

% 并对照片进行读取、重设大小、灰度处理和储存数值向量

function ChooseaphotoButtonPushed(app, event)

[imgfilename, imgpathname] = uigetfile('\*.jpg','Pick a photo to test');

if imgfilename

strPath=strcat(imgpathname,imgfilename);

imgdata=imread(strPath);

imgdata=imresize(imgdata,[112,92]);

imshow(imgdata,'Parent',app.UIAxes);

imgdata=rgb2gray(imgdata);

imgdata=im2double(imgdata);

save('img.mat','imgdata')

end

end

% Callback function: GenderRecognitionPanel, TestButton

% 回调函数，用于将图像特征输入训练好的神经网络并输出结果

function TestButtonPushed(app, event)

load img.mat;

test(app,imgdata);

end

end

% App initialization and construction

methods (Access = private)

% Create UIFigure and components

function createComponents(app)

% Create UIFigure

app.UIFigure = uifigure;

app.UIFigure.Position = [100 100 640 480];

app.UIFigure.Name = 'UI Figure';

% Create GenderRecognitionPanel

app.GenderRecognitionPanel = uipanel(app.UIFigure);

app.GenderRecognitionPanel.Title = 'Gender Recognition';

app.GenderRecognitionPanel.SizeChangedFcn = createCallbackFcn(app, @TestButtonPushed, true);

app.GenderRecognitionPanel.Position = [319 307 190 85];

% Create ChooseaphotoButton

app.ChooseaphotoButton = uibutton(app.GenderRecognitionPanel, 'push');

app.ChooseaphotoButton.ButtonPushedFcn = createCallbackFcn(app, @ChooseaphotoButtonPushed, true);

app.ChooseaphotoButton.Position = [54 38 100 22];

app.ChooseaphotoButton.Text = 'Choose a photo';

% Create TestButton

app.TestButton = uibutton(app.GenderRecognitionPanel, 'push');

app.TestButton.ButtonPushedFcn = createCallbackFcn(app, @TestButtonPushed, true);

app.TestButton.Position = [54 5 100 22];

app.TestButton.Text = 'Test';

% Create ResultTextAreaLabel

app.ResultTextAreaLabel = uilabel(app.UIFigure);

app.ResultTextAreaLabel.HorizontalAlignment = 'right';

app.ResultTextAreaLabel.Position = [319 273 40 15];

app.ResultTextAreaLabel.Text = 'Result';

% Create ResultTextArea

app.ResultTextArea = uitextarea(app.UIFigure);

app.ResultTextArea.Position = [374 230 121 60];

% Create UIAxes

app.UIAxes = uiaxes(app.UIFigure);

title(app.UIAxes, 'testing photo')

xlabel(app.UIAxes, 'X')

ylabel(app.UIAxes, 'Y')

app.UIAxes.DataAspectRatio = [1 1 1];

app.UIAxes.PlotBoxAspectRatio = [1 1 1];

app.UIAxes.XLim = [0 1];

app.UIAxes.YLim = [0 1];

app.UIAxes.ZLim = [0 1];

app.UIAxes.CLim = [0 1];

app.UIAxes.GridColor = [0.15 0.15 0.15];

app.UIAxes.MinorGridColor = [0.1 0.1 0.1];

app.UIAxes.XColor = [0.15 0.15 0.15];

app.UIAxes.XTick = [0 0.2 0.4 0.6 0.8 1];

app.UIAxes.YColor = [0.15 0.15 0.15];

app.UIAxes.YTick = [0 0.5 1];

app.UIAxes.ZColor = [0.15 0.15 0.15];

app.UIAxes.ZTick = [0 0.5 1];

app.UIAxes.CameraPosition = [0.5 0.5 9.16025403784439];

app.UIAxes.CameraTarget = [0.5 0.5 0.5];

app.UIAxes.CameraUpVector = [0 1 0];

app.UIAxes.Position = [1 218 300 185];

end

end

methods (Access = public)

% Construct app

function app = gender\_recognition

% Create and configure components

createComponents(app)

% Register the app with App Designer

registerApp(app, app.UIFigure)

if nargout == 0

clear app

end

end

% Code that executes before app deletion

function delete(app)

% Delete UIFigure when app is deleted

delete(app.UIFigure)

end

end

end