操作系统课程设计

学 号：\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

姓 名：\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**实验目的：**

1. 以90%的精度学习人脸图像，从20个人中识别出目标人物
2. 讨论隐藏节点数量对结果的影响
3. 讨论学习速率对结果的影响

**实验过程：**

1. 实现人脸识别

先进行降维处理

图形用户界面, 应用程序

描述已自动生成

将960维的属性，下降为67维。

图形用户界面

描述已自动生成

可以看出，可以正确识别每个人的身份信息

1. 实现姿势识别：

图片包含 应用程序

描述已自动生成

可以看出，可以正确识别每个人的姿势信息

1. 讨论隐藏节点数量对结果的影响

在学习速率为0.3的情况下进行测试：

|  |  |
| --- | --- |
| 隐藏的节点数 | 正确率 |
| a ((attributes + classes) / 2) | 99.5192 % |
| 1 | 19.391 % |
| 2 | 64.5833 % |
| 3 | 93.75 % |
| 4 | 97.5962 % |
| 5 | 98.8782 % |
| 6 | 98.3974 % |
| 7 | 99.0385 % |
| 8 | 99.5192 % |
| 9 | 99.1987 % |
| 10 | 99.5192 % |

从上表可以看出随着隐藏节点数的增加，正确率会收敛到通用符a的正确率

1. 讨论学习速率对结果的影响

使用隐藏节点数为2的情况下进行测试

|  |  |
| --- | --- |
| 学习比率 | 正确率 |
| 0.1 | 44.391 % |
| 0.2 | 61.859 % |
| 0.3 | 64.5833 % |
| 0.4 | 62.0192 % |
| 0.5 | 62.0192 % |
| 0.6 | 67.3077 % |
| 0.7 | 61.2179 % |
| 0.8 | 65.2244 % |
| 0.9 | 55.2885 % |

可以看到学习比率较高和较低时都会导致机器学习无法正确识别

1. 高级任务：

使用不同的学习速率和隐藏节点数：

|  |  |  |  |
| --- | --- | --- | --- |
| 学习速率 | 隐藏节点数 | 正确率 | 加入pose的正确率 |
| 0.3 | a | 99.36% | 99.52% |
| 0.5 | a | 99.52% | 99.52% |
| 0.7 | a | 99.52% | 99.52% |
| 0.9 | a | 99.52% | 99.20% |
| 0.3 | 3 | 93.75% | 95.67% |
| 0.5 | 3 | 93.27% | 92.31% |
| 0.7 | 3 | 92.47% | 91.83% |
| 0.9 | 3 | 90.22% | 88.14% |
| 0.3 | 6 | 98.40% | 99.04% |
| 0.5 | 6 | 98.88% | 99.04% |
| 0.7 | 6 | 98.72% | 99.20% |
| 0.9 | 6 | 98.24% | 99.36% |
| 0.3 | 9 | 99.20% | 99.52% |
| 0.5 | 9 | 99.04% | 99.68% |
| 0.7 | 9 | 99.36% | 99.68% |
| 0.9 | 9 | 99.52% | 99.68% |

但是，从表格无法看出有任何关系

但大多数情况会提高一些识别的正确率

**实验心得：**

1. **如何在weka中导入图像数据？**:

将pgm文件导出为矩阵数组，将其转化为一维数组后加入<userid>类标签导入excel表格中，将xlsx文件另存为csv文件，然后用weka的arffviewer打开，另存为arff格式，并稍加修饰。

1. **数据中的pgm文件如何打开？**:

*# 使用原始字符串表示路径*image = Image.open(r'C:\code\pythonProject\jupy\_test\机器学习实验\faces\at33\at33\_left\_angry\_open.pgm')  
  
*# 显示图像*image.show()

1. 如何实现降维?

fliters -> unsupervised -> attribute -> PrincipalComponents

点击Apply

**实验代码：**

from PIL import Image

import os

import cv2

import pandas as pd

查看pgm图片

目标：可以观看

*# 打开pgm照片查看*

*# 使用原始字符串表示路径*

image = Image.open(r'C:\code\pythonProject\jupy\_test\机器学习实验\faces\at33\at33\_left\_angry\_open.pgm')

*# 显示图像*

image.show()

遍历一下，查看所有的照片

目的：已可以遍历所有文件

*# 打开所有照片查看*

*# 指定要搜索的文件夹路径*

folder\_path = r'C:\code\pythonProject\jupy\_test\机器学习实验\faces\an2i'

*# 列出文件夹中的所有文件*

files = os.listdir(folder\_path)

*# 遍历所有文件*

for file\_name in files:

full\_path = os.path.join(folder\_path, file\_name)

*# 检查是否是文件*

if os.path.isfile(full\_path):

image = Image.open(full\_path)

image.show()

测试将pgm图片转化为NumPy数组

目标：实现了将单一pgm文件转为NumPy数组

from PIL import Image

import numpy as np

*# 打开PGM图像*

image = Image.open(r'C:\code\pythonProject\jupy\_test\机器学习实验\faces\_4\an2i\an2i\_left\_angry\_open\_4.pgm')

*# 将图像转换为NumPy数组*

image\_array = np.array(image)

*# 打印矩阵*

print(image\_array)

将NumPy数组存储txt文件

目标：可以实现单一文件夹的所有pgm文件转为NumPy数组并存储到txt文件中

*# 打开所有照片查看*

import os

*# 指定要搜索的文件夹路径*

folder\_path = r'C:\code\pythonProject\jupy\_test\机器学习实验\faces\_4\an2i'

*# 列出文件夹中的所有文件*

files = os.listdir(folder\_path)

*# 把dataset中的数据先清空，方便后面输入数据*

file\_path = "C:\\code\\pythonProject\\jupy\_test\\机器学习实验\\dataset.txt"

*# 以写入模式打开文件（如果文件不存在将创建它）*

with open(file\_path, "w") as file:

*# 不需要写入任何内容，只是打开并立即关闭文件*

pass

*# 遍历所有文件*

for file\_name in files:

full\_path = os.path.join(folder\_path, file\_name)

*# for i in range(1):*

*# full\_path = r'C:\code\pythonProject\jupy\_test\机器学习实验\faces\_4\an2i\an2i\_left\_angry\_open\_4.pgm'*

*# 检查是否是文件*

if os.path.isfile(full\_path):

image = Image.open(full\_path)

*# 将图像转换为NumPy数组*

image\_array = np.array(image)

*# 将其打印到dataset数据集中*

with open("C:\code\pythonProject\jupy\_test\机器学习实验\dataset.txt", "a") as file:

*# 遍历图像数组的每一行*

for row in image\_array:

stri = ""

for i in row:

stri += str(i) + ","

file.write(stri)

path\_name = str(full\_path).split("\\")

files\_name = path\_name[-1].split("\_")

files\_id = files\_name[0] *# + "," + files\_name[1] + "," + files\_name[2] + "," + files\_name[3] + "\n"*

file.write(files\_id)

进度：过程出了点问题，应该导出到xlsx文件而不是txt文件

成功导入到dataset.xlsx文件中

*# 打开所有照片查看*

import os

*# 指定要搜索的文件夹路径*

folder\_path = r'C:\code\pythonProject\jupy\_test\机器学习实验\faces\_4'

excel\_path = r'C:\code\pythonProject\jupy\_test\机器学习实验\dataset.xlsx'

datasets = pd.DataFrame()

*# 列出文件夹中的所有文件*

files = os.listdir(folder\_path)

for root, dirs, files in os.walk(folder\_path):

for dir in dirs:

*# 继续遍历子文件夹*

folder\_path = os.path.join(root, dir)

files = os.listdir(folder\_path)

*# 遍历当前文件夹下的所有pgm文件*

for file\_name in files:

full\_path = os.path.join(folder\_path, file\_name)

print(full\_path)

*# 筛选出人名——userid*

path\_name = str(full\_path).split("\\")

files\_name = path\_name[-1].split("\_")

userid = files\_name[0]

*# 检查是否是文件*

if os.path.isfile(full\_path):

image = Image.open(full\_path)

*# 将图像转换为NumPy数组*

image\_array = np.array(image)

*# 将NumPy数组转为只有一行的数组*

image\_array = image\_array.reshape(1, -1)

*# 将 NumPy 数组转换为 Pandas DataFrame*

df = pd.DataFrame(image\_array)

*# 将人名添加到df中*

df['userid'] = userid

*# 将数据添加到数据集中*

datasets = pd.concat([datasets, df], ignore\_index=True)

*# 使用 Pandas 将 DataFrame 导出到 Excel 文件*

datasets.to\_excel(excel\_path, index=False)