

# 项目名称：基于NVIDIA-GPU的 深度学习和代码加速计算

---

## 项目名称：基于NVIDIA-GPU的深度学习和代码加速计算

项目描述

项目许可证

项目人员和分工

项目开发流程

项目源码

邮件列表

缺陷追踪系统

项目计划（版本、用户、推广）

项目配置

主要代码

1. [clean fer2013.py](#)
2. [EmotionRecongnition.py](#)
3. [FER\\_test.py](#)
4. [Parse fer2013.py](#)
5. [real\\_time\\_video\\_me.py](#)
6. [runMain.py](#)
7. [train\\_emotion\\_classifier.py](#)
8. [fer2013\\_clean2.csv](#)

程序算法测试

项目改进

运行结果展示

## 项目描述

---

1. 基于Python 3实现表情识别模型训练及测试；
2. 基于Python 2实现NAO机器人利用已训练好的模型进行表情识别；

本项目涉及使用NVIDIA GPU加速深度学习训练过程，既有的GPU串行深度学习的源码，在此基础上做GPU并行加速。原理是基于CNN卷积神经网络算法训练模型，在TensorFlow学习框架下对人的表情进行识别，预测人表情所表达的情绪。

# 项目许可证

---

MIT License

Copyright (c) 2020 dajiao918

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

## 项目人员和分工

---

姓名	分工
陈浩	代码分析/撰写文档
殷绍轩	代码分析/撰写文档
姜明宇	代码分析
孙涛	撰写文档
余兴广	撰写文档

## 项目开发流程

---

第一轮，全部成员并行分工通读源码，并且借助互联网搜索引擎写对应源码的解释文档。文档中进行不明晰语句的注释和函数原理的阐述。

第二轮，成员之间进行圆桌讨论，交流代码内容，把各分支代码汇总，最终融合为贴近源码作者逻辑的开发思想。讨论中进行代码的释义。

第三轮，针对代码运行现有的bug报错信息进行调试溯源，修复源码漏洞。

第四轮，轮换几个CNN算法，改进关于opencv和tensorflow-keras函数，比对运行时间。

第五轮，收集现有成果，撰写项目报告。

## 项目源码

---

[https://github.com/dajiao918/homework/tree/main/Emotion\\_Recognition](https://github.com/dajiao918/homework/tree/main/Emotion_Recognition)

## 邮件列表

---

人员	邮箱	学号	班级
陈浩	<a href="mailto:ccc147569@gmail.com">ccc147569@gmail.com</a>	2019010132	计科 1905
余兴广	<a href="mailto:3070261303@qq.com">3070261303@qq.com</a>	2018010029	计科 1906
殷绍轩	<a href="mailto:djmax96945147@outlook.com">djmax96945147@outlook.com</a>	2018010636	计科 1905
姜明宇	<a href="mailto:1617350667@qq.com">1617350667@qq.com</a>	2019011254	计科 1905
孙涛	<a href="mailto:2179020062@qq.com">2179020062@qq.com</a>	2019011170	计科 1905

## 缺陷追踪系统

---

Bugzilla 2.3.0

## 项目计划（版本、用户、推广）

---

版本：1.0

用户：学习使用Tensorflow框架-Python实现深度学习学习的人群

推广：线上推广

在此次为期不长的项目旅途中，收获更多的是对于陌生项目源码的快速学习能力和分析能力。这一点至关重要，很荣幸我们这一点受到了充分的锻炼，受制于知识储备不够。小组成员辛勤奉献，虽然有所进展，但是遇到了难以跨越的瓶颈，依然希望能够看到未来更大的进步。

本项目后续2021年初应该会有后续进展与更新，更多请关注本仓库。

## 项目配置

---

1. 安装python3.6.8

2. 安装IDE：pycharm
3. 安装cuda，在NVIDIA控制面板中查询自己所需要的的cuda版本，然后再根据cuda版本下载对应的cudnn
4. 安装Anaconda3.6
5. Anaconda安装完后，在anaconda自带命令行敲入以下命令行

```
1 | pip install -i http://pypi.douban.com/simple ... (需要  
    安装的库)
```

6. 根据上面的命令行依次安装以下库和TensorFlow

1. tensorflow-gpu
2. keras 2.2.4
3. numpy 1.16.6
4. scikit-learn 0.21.2
5. matplotlib 3.2.1
6. pandas 0.24.2
7. opencv-python 4.1.0.25
8. Pillow 7.0.0 （用于NAO机器人进行表情识别）
9. PyQt5 5.13.2
10. imutils 0.5.2

7. 在pycharm中配置anconda环境

- 选择File->Setting->Project Interpreter，点击右上角齿轮->add，选择Conda Environment->Existing enviroment，选择anconda的安装路径，目录加载之后，会自动加载该python环境下的包，点击apply即可

## 主要代码

---

### 1. clean fer2013.py

- 初始化：定义emotion选项，并读取指定文档中的图片数据，检测共读取了多少张照片，并采用级联分级器设定人脸框的范围 (line:59-61).
  - `open(name[, mode[, buffering]])` 函数，
    - name : 文件名

- mode : mode 决定了打开文件的模式：只读，写入，追加等。默认文件访问模式为只读(r)。
  - buffering : 如果 buffering 的值被设为 0，就不会有寄存。如果 buffering 的值取 1，访问文件时会寄存行。如果将 buffering 的值设为大于 1 的整数，表明了这就是的寄存区的缓冲大小。如果取负值，寄存区的缓冲大小则为系统默认。
- xml文件--数据集文件
  - stageType级联类型，目前只能取BOOST
  - featureType训练使用的特征类型，目前支持的特征有Haar，LBP和HOG
  - maxWeakCount每级强分类器中弱分类器的最大个数，当FA降不到指定的maxFalseAlarm时可以通过指定最大弱分类器个数停止单个强分类器
- cv2.CascadeClassifier()是opencv的级联分级器
- data.loc是根据索引来读取该索引下的数据，来自于pandas
- list(image\_data.split())：通过指定分隔符对字符串进行[切片]，如果参数 num 有指定值，则分隔 num+1 个子字符串。e.g.print str.split(';')#以空格为分隔符，包含换行符
- np.asarray(\*a\*, dtype=None, order=None)[Convert the input to an array.](#)
- python路径拼接os.path.join()函数
- [Image.Fromarray\(img.astype\('uint 8'\), mode='rgb'\)](#)返回灰度图像
- cv2.imread():读取图片文件
  - imread函数有两个参数，第一个参数是图片路径，第二个参数表示读取图片的形式，有2种：
    - cv2.IMREAD\_COLOR：加载彩色图片，这个是默认参数，可以直接写1。  
LE：以灰度模式加载图片，可以直接写0。
    - cv2.IMREAD\_UNCHANGED：包括alpha，可以直接写-1

- ```

1 faces =
  face_detection.detectMultiScale(img, scaleFactor=1.1,
2
3         minNeighbors=5,
4         minSize=(30, 30),
5
6         flags=cv2.CASCADE_SCALE_IMAGE)

```

- image表示的是要检测的输入图像
- objects表示检测到的人脸目标序列
- scaleFactor表示每次图像尺寸减小的比例
- minNeighbors表示每一个目标至少要被检测到3次才算是真的目标 (因为周围的像素和不同的窗口大小都可以检测到人脸),
- minSize为目标的最小尺寸
- minSize为目标的最大尺寸

**总结：**这个文件功能是读取数据`array`，然后转换成图像，在图像上框出人脸

## 2. EmotionRecongnition.py

- 用Qt设计一个界面，并检测摄像头功能是否正常工作 (line:378)（采用定时刷新摄像头捕获的画面），并且读取rgb图片类型将其转换为bgr (line:486)（因后续有OpenCV处理需求）。
- `getcwd()` 函数 返回当前工作目录。
- `retranslate()` 函数 ch传入参数为窗口对象，重新刷新UI。
- `setMovie()` 函数 传入QMovie对象，设置UI界面显示内容。
- `cv2.VideoCapture()` 函数 打开摄像头，可传入0，打开内置摄像头。无参数传入默认打开内置摄像头。
- `line 486 imdecode()` 函数
  - `np.fromfile(filePath, dtype=np.uint8)`，调用numpy库的fromfile方法，以指定类型读取文件。
  - filePath:读取文件的路径。
  - dtype=np.uint8：以uint8类型读取文件

- 注：uint8类型是专门用于存储各种图像的（包括RGB，灰度图像等），范围是从0–255。
- `line375 timer_camera` 计时器与 `show_camera` 函数绑定，`line397` 设置定时器为30毫秒，每隔30 毫秒触发槽函数，执行 `show_camera` 方法，刷新摄像头画面。经测试，调整计时器间隔对识别用时有一定影响。

### 3. FER\_test.py

- 实现简易的Qt界面，用于测试，并将bgr图片类型转为灰度图  
`(line:173)`（减少图像原始数据量，减少计算量，优化GPU对图像的处理），对摄像头所读取的数据进行处理，显示概率柱状图  
`(line:177).`

### 4. Parse fer2013.py

- 实现 `clean fer2013.py` 中的初始化，并且将 `csv` 文件中的一维像素灰度数据，通过切片转换为 `48x48` 的矩阵，以恢复为图片文件。

### 5. real\_time\_video\_me.py

- 实现 FER\_test.py 的相同功能，并直接用于 runMain.py 中。
- `fX, fY, fW, fH`: 刚开始的两个变量表示的是候选框x和y的位置，接下来两个变量的值表示的是候选框的宽度和高度
- `cv2.cvtColor(frame, COLOR_BGR2GRAY)` 将图片从BGR转成灰度图像

### 6. runMain.py

- 实现调出设计完整的Qt界面显示以及人脸识别系统。

### 7. train\_emotion\_classifier.py

- 人脸表情识别训练库  
 构建模型，并在模型中使用了交叉熵损失函数 `(line:35)`（输出一个概率值，反映预测为正类的可能性），混淆矩阵 `(line:84)`（观察模型在不同类别的表现，并计算模型在对应类别中的准确率，使各个类别具有区分性）。



具体操作为改动参数来实现不同模型的创建，而从中创建的模型可以被导入 `runMain.py` 中实现人脸识别的概率分析。

- `batch_size = 32` (mini-batch gradient decent - 小批梯度下降)

1次迭代所使用的样本梯度数。

每训练一次batch中的样本梯度都会下降，进而使目标函数优化

- `num_epochs = 10000`

1遍训练集中的所有样本数。

即训练过程中数据将被“轮询”的总次数。

- `input_shape = (48, 48, 1)`

定义：`input_shape=(heights,widths,channels)`。

代表为：具有1阶张量（应该为层数为1）的48\*48的数据域。

- `input_shape = (48, 48, 1)`

定义：`input_shape=(heights,widths,channels)`。

代表为：具有1阶张量（应该为层数为1）的48\*48的数据域。

- `validation_split = .2`

0~1之间的浮点数，用于在没有提供验证集的时候，按一定比例从训练集中取出一部分作为验证集。

主要用于在epochs结束后，测试损失函数和精确度。

- `verbose = 1`

日志显示

0为不在标准输出流输出日志信息

1为输出进度条记录

2为每个epoch输出一行记录

- `num_classes = 7`

含义：最后一个全连接层所分出的类个数为7

- `patience = 40`

当monitor不再有改善的时候就会停止训练，这个可以通过patience看出来

- `base_path = 'models/'`

训练模型数据上传位置

## 8. fer2013\_clean2.csv

通过像素点数据集来算出情绪类别，每个数据--向量有2304(48\*48)个值，数据被分为四个用途：`Training, CK+, Public Test, Private Test`

## 程序算法测试

实现步骤：

1. 运行 `FER_test.py` 做简单的表情识别测试，从摄像头获取人脸并识别表情。模型路径在代码中设置
2. 运行 `runMain.py` 可在功能更全的界面进行测试，可从界面选择模型和测试图片及摄像头。

## 项目改进

- 在 `FER_test.py` (line:189-190) 和 `real_time_video_me.py` (line:55-56) 中添加了一行代码：

```
1 | if roi.shape[0] < 48 or roi.shape[1] < 48:  
2 |     return
```

- **实现：**人脸被捕捉后不在摄像头捕捉范围内，像素点数量无法到达 `48x48` 个, 这个时候若强行生成 `48x48` 的矩阵，一定是会报错导致程序意外退出的。此语句保证了当像素点数量不足时，不触发表情识别。
- **效果：**运行文件 `FER_test.py` 和 `runMain.py` 时候不会触发库函数调用失败的错误。

## 运行结果展示

# 表情识别系统

使用说明：首先选择模型，然后选择  
摄像头或图片文件进行识别



D:/程序\_计科1602\_2016011230\_康路\_基



实时摄像已关闭



C:/Users/陈浩/OneDrive/桌面/开源软件



用时: 0.448



识别结果: happy

angry: 4.40%

disgust: 1.20%

fear: 15.66%

happy: 68.88%

sad: 5.99%

surprise: 1.58%

neutral: 2.29%

# 表情识别系统

使用说明：首先选择模型，然后选择  
摄像头或图片文件进行识别



D:/程序\_计科1602\_2016011230\_康路\_基



实时摄像已关闭



C:/Users/陈浩/OneDrive/桌面/开源软件



用时: 0.454



识别结果: angry

angry: 58.31%

disgust: 4.98%

fear: 2.20%

happy: 20.13%

sad: 2.99%

surprise: 2.13%

neutral: 9.25%