

# 说明文档

## 1. 项目概述

### 1.1 项目背景

情绪是对一系列主观认知经验的通称,是人对客观事物的态度体验以及相应的行为反应,一般认为,情绪是以个体愿望和需要为中介的一种心理活动。微反应体系将人类的情绪按照情绪与刺激源力度之间的相关性,分为:惊讶、厌恶、愤怒、恐惧、悲伤和快乐等。

### 1.2 需求描述

设计出一种基于 Python 的人脸表情识别和情绪分类模型,在 MindSpore 框架下,使用其内部高级 API 搭建训练模型,并利用 OpenCV 库中的相关函数共同实现人脸识别、人脸表情识别、情绪分类的目的。该模型能对传入的图片进行降噪和灰度处理等操作后对人脸进行检测,然后将检测到的人脸裁剪后完成表情分析工作。

### 1.3 项目介绍

本项目是一个基于 Python 的人脸表情识别和情绪分类模型。该项目基于 mindspore 框架搭建了一个 LeNet5 网络模型,选取 fer2013Plus 数据集对模型进行训练。在模型应用的过程中,首先使用 CascadeClassifier 进行人脸检测,识别出人脸区域,再用训练好的模型进一步识别出人脸表情,最终实现人脸的表情分类。

**数据集介绍:** fer2013Plus 人脸表情数据集由 35585 张人脸表情图片组成,每张图片是由大小固定为  $48 \times 48$  的灰度图像组成,该数据被分为训练集和测试集,每个数据集文件下共有 8 种表情文件,分别为 anger (生气)、contempt (蔑视)、disgust (憎恶)、fear (害怕)、happiness (高兴)、neutral (正常)、sadness (伤心)、surprise (惊讶)

**数据集下载地址:** [Fer2013Plus](#) | [Kaggle](#)

## 2. 模型搭建

### 2.1 运行环境

#### (1) 硬件环境

Intel(R) Core(TM) i5-1035G1 CPU @ 1.00GHz    1.19 GHz

## (2) 软件环境

Windows 10 操作系统 64 位

python 3.7

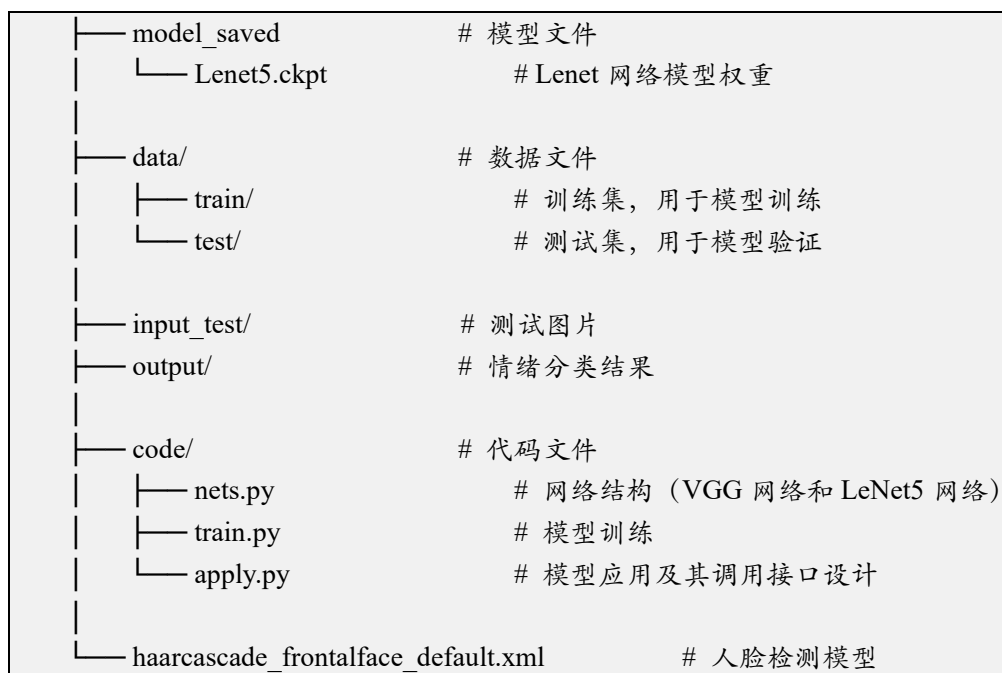
## (3) 项目相关依赖包

mindspore 1.7

cv2 4.6.0

numpy 1.21.6

## 2.2 目录结构



## 2.3 模型搭建流程

### (1) 构建模型的网络结构（nets.py）

定义了两个网络模型进行测试训练，最后选择效果最优的模型进行应用。

➤ VGG 网络

➤ Lenet5 网络

### (2) 模型训练及优化（train.py）

➤ 构造模型输入：读取数据集文件，使用图像处理算子进行数据增强，然后调整数据集的输入格式，进行批处理操作，获得最终的模型输入

➤ 获取网络结构：从 nets.py 中导入定义好的模型结构，并将其实例化得到预训练模型

➤ 编译、训练模型：模型的损失函数选择 SoftmaxCrossEntropyWithLogits（Softmax 交叉熵损失函数），优化器选择 SGD（随机梯度下降法），然后对模型进行训练

- 模型优化：通过观察模型训练过程中回调的损失值，不断调整超参数对模型进行优化

依次选择 DNN 网络和 LeNet5 网络进行上述的模型训练，最终发现 LeNet5 模型的测试效果更优，因此选择 LeNet5 模型作为我们情绪识别的最终模型。（保存模型文件：LeNet.ckpt）

### (3) 模型推理应用（apply.py）

- 加载相关模型：载入训练好的 LeNet5 模型和人脸检测器 CascadeClassifier
- 获取人脸数据：使用人脸分类器检测出图片中的人脸区域
- 调整输入数据的格式以满足模型输入要求：使用 opencv 库对获得的人脸图像进行灰度、降噪、裁剪处理，最后将其转化为 minspore 的 Tensor 格式作为模型输入
- 使用该 LeNet5 模型对输入数据进行预测，最终得到输入图像的的情绪识别结果。
- 应用接口设计：提供两种模式来选择待检测图像数据源，一种是调用摄像头直接获取图像数据，其情绪分类结果可实时显示在人脸框上方；另一种是从本地文件中导入图像数据，其情绪分类结果将保存在用户指定的文件路径中。

## 3. 模型应用

### 3.1 应用说明

该模型通过调用 apply.py 文件中的 get\_emotion 接口实现，即使用时需导入该应用模块，具体使用样例如下：

```
from apply import get_emotion
import os

# img_path = os.path.abspath('../input_test/1.jpg')
# res_path = os.path.abspath('../output')
# emotion = get_emotion(mode="file", img_path=img_path, res_path=res_path)

emotion = get_emotion(mode="video")
```

#### (1) 输入项：人脸数据源

get\_emotion 接口的默认输入如下所示：

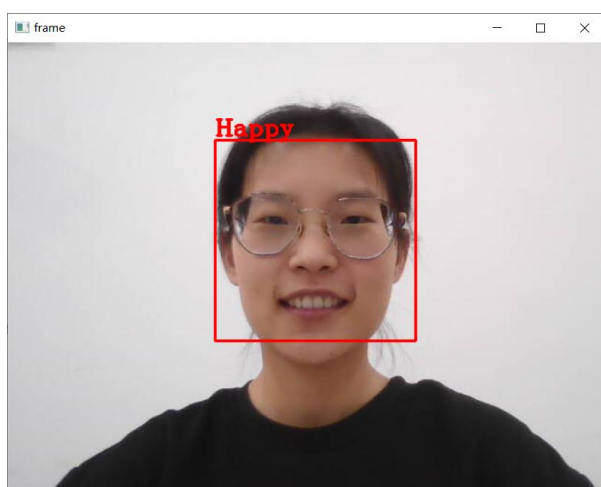
```
get_emotion(mode="video", img_path=None, res_path=None)
```

**mode 参数：**选择输入模式，即人脸数据的来源“video” or “file”，为必选参数

**img\_path 参数：**待检测的图片路径（若 mode=”file“，则必须填入此参数）

**res\_path 参数：**情绪分类结果的保存路径（若 mode=”file“，则必须填入此参数）

(2) 输出项：模型预测的情绪类别

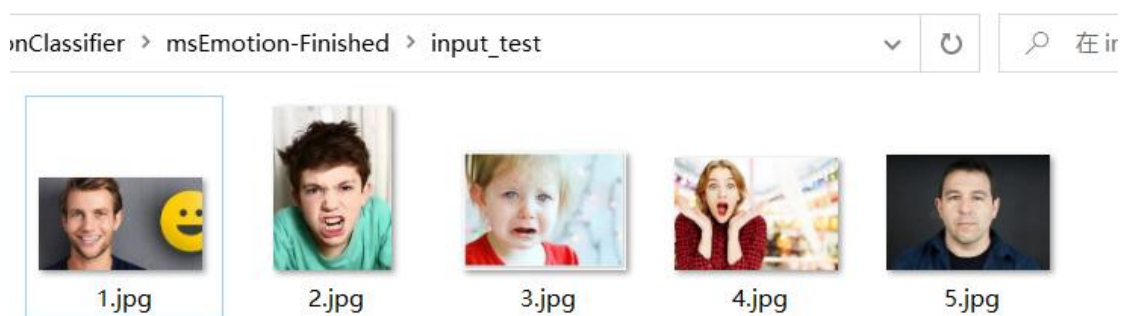


```
D:\ProgramFilesX86\anaconda3\envs\MindSpore\python.exe  
Emotion: Happy
```

```
进程已结束,退出代码0
```

### 3.2 模型效果测试：

(1) 测试数据：



(2) 输出结果:

```
1 import os
2 from apply import get_emotion
3
4 file_path = os.path.abspath('../input_test/')
5 res_path = os.path.abspath('../output')
6
7 img_list = os.listdir(file_path)
8 for img in img_list:
9     img_path = os.path.join(file_path, img)
10    emotion = get_emotion(mode="file", img_path=img_path, res_path=res_path)
```

Emotion: Happy  
Emotion: Angry  
Emotion: Sad  
Emotion: Surprise  
Emotion: Neutral

Classifier > msEmotion-Finished > output

在 output 中搜索

