

本项目根据 MediaPipe 和 Openpose 具备单独提取能力和速度快的特点,进行动/静态的骨架识别与肢体定位。

# 5.1 总体设计

本部分包括整体框架和系统流程。

## 5.1.1 整体框架

整体框架如图 5-1 所示。



图 5-1 整体框架

## 5.1.2 系统流程

系统流程如图 5-2 所示。



# 5.2 运行环境

本部分包括 Python 环境、Openpose 环境和 PyQt6 环境。

### 5.2.1 Python 环境

在 Windows 环境下下载 Anaconda,完成 Python 3.9 以上版本的环境配置,如图 1-3 所示,也可以下载虚拟机在 Linux 环境下运行代码。

#### 5.2.2 Openpose 环境

(1) 下载开源项目,如图 5-3 所示。

(2) 进入.. \openpose-master \models 目录,运行 getModels. bat。

(3) 等待加载模型 pose\_iter\_584000. caffemodel。时间较长,如果中途中断,重新双击运行 getModels. bat 即可,完整加载模型大小是 100MB。

(4) 保存位置如下:

..\openpose-master\models\pose/mpi/pose\_iter\_160000. caffemodelOpenPose

Product 🗸 Solutions	✓ Open Source ✓ Pricing	Q Search or jump to	7 Sign in Sign up	
GMU-Perceptual-Co	mputing-Lab / openpose Public	☐ Notifications ♀	Fork 7.7k 🛱 Star 27.8k 👻	
<> Code   Issues 256	11 Pull requests 16 🕟 Actions 🗄 Projects ① Securi	ty 🗠 Insights		
₽ master → ₽ 3 branch	es 🖏 15 tags	Code - About		
auto-matic Fixed typo in	README (#2212) 8	00000000000000000000000000000000000000	se: Real-time multi-person It detection library for body, face,	
.github	Fixed several doc typos (#2068)	last year	nands, and root estimation	
3rdparty	Eigen updated to 3.3.8 and working on Windows	3 years ago		
Cmake	Fixed several doc typos (#2068)	last year caffe		
doc	Add RealSense2OpenPose3D project link (#2216)	2 months ago cpp		
examples	Fixed several doc typos (#2068)	last year		
include	Fixed several doc typos (#2068)	last year		
models	Added Travis build for CPU/CUDA/U16/U14 (#1003)	5 years ago keypoin		
python	Update openpose_python.cpp (#2105)	last year hand-es		
scripts	Fixed several doc typos (#2068)	last year 🛛 🛱 Read		
src src	Fix typos. (#2086)	last year গ্রুম View		
doc_autogeneration.dox	/gen Doxygen to 1.9; doc reordered, renamed & modified to	adapt to GH 2 years ago		
gitignore	Improved GitHub pages & fixed some typos in there	2 years ago 💿 918		
🗋 .gitmodules	Python Pybind11 wrapper (#1014)	Python Pybind11 wrapper (#1014) 5 years ago 😵 7.7k forks		
CMakeLists.txt	Fixed several doc typos (#2068)	last year	Report repository	
	Updated Caffe version license 1.0.0rc5 to 1.0.0	6 years ago		
「門 README.md 尊待 github.com	Fixed typo in README (#2212)	last month	Releases 15	

图 5-3 开源项目界面

(5) 使用 pip 安装所需的包。

pip install opencv2
pip install matplotlib
pip install mediapipe

## 5.2.3 PyQt6 环境

(1) 在 Anaconda 中如果没有 PyQt6,需要使用 pip 进行安装,步骤如下: 打开 Anaconda Prompt (Anaconda3)控制台,切换至相应的 conda 环境,执行如下命令:

```
pip install sip
pip install PyQt6
pip install PyQt6 - tools
```

(2) 可以在使用 pip 时后面加上参数 i, 指定 pip 源。

# 5.3 模块实现

本部分包括静态识别、动态识别和模块展示,下面分别给出各模块的功能介绍及相关 代码。

#### 5.3.1 静态识别

静态识别结构如图 5-4 所示。



图 5-4 静态识别结构

通过 Openpose 中带有 pose 的识别数据集,可以在同一张图像中对脸、手和骨架进行识别,还能输出对应的关键点热力图,相关代码见"代码文件 5-1"。

#### 5.3.2 动态识别

动态识别使用 mediapipe 更加轻量化,在安装过程中并没有预安装数据集的过程,因此 实现起来接近 API 的调用,相关代码见"代码文件 5-2"。

#### 5.3.3 模块展示

界面设计包括四个按键:选择图像、运行静态识别输出结果、开启摄像头并进行动态识别和关闭摄像头。moving.py负责动态识别、single.py负责静态识别、main.py负责对界面数据槽关系进行绑定,相关代码见"代码文件 5-3"。

## 5.4 系统测试

静态识别可以在输入图像、选择模型后显示识别结果和特征点,如图 5-5 所示;动态识 别通过 cap 输入图像流,然后可以在图像上绘出关键点,达到即时识别的结果,如图 5-6 所 示;界面识别结果如图 5-7 所示。

#### 46 🚽 深度学习应用开发实践——文本音频图像处理30例



图 5-5 静态识别结果和热力图



图 5-6 动态识别结果



图 5-7 界面识别结果



本项目基于循环神经网络,构建多层 RNN 模型,通过数据集的训练,生成藏头诗与歌词。

# 6.1 总体设计

本部分包括整体框架和系统流程。

## 6.1.1 整体框架

古诗生成整体框架如图 6-1 所示,歌词生成整体框架如图 6-2 所示。



图 6-1 古诗生成整体框架

#### 48 🚽 深度学习应用开发实践——文本音频图像处理30例



图 6-2 歌词生成整体框架

# 6.1.2 系统流程

古诗生成系统流程如图 6-3 所示,歌词生成系统流程如图 6-4 所示。



图 6-3 古诗生成系统流程



图 6-4 歌词生成系统流程

# 6.2 运行环境

本部分包括 Python 环境、TensorFlow 环境和 PyCharm 环境。

#### 6.2.1 Python 环境

在 Windows 环境下下载 Anaconda,完成 Python 3.7.3 及以上版本的环境配置,如图 1-3 所示,在 PyCharm 环境下进行开发。

## 6.2.2 TensorFlow 环境

(1) 打开 Anaconda Prompt,输入清华仓库镜像。

```
conda config -- add channels
https://mirrors.tuna.tsinghua.edu.cn/anaconda/pkgs/free/
conda config - set show_channel_urls yes
```

(2) 创建 Python 3.7 环境,名称为 TensorFlow,此时 Python 版本和后面 TensorFlow 的版本如有匹配问题,此步选择 Python 3.x。

```
conda create - n tensorflow python = 3.7
```

- (3) 有需要确认的地方都输入 y。
- (4) 在 Anaconda Prompt 中激活 TensorFlow 环境:

activate tensorflow

(5) 安装 CPU 版本的 TensorFlow:

pip install - upgrade -- ignore - installed tensorflow

## 6.2.3 PyCharm 环境

PyCharm 版本为 PyCharm 2018。

## 6.3 模块实现

本部分包括数据准备、模型构建、模型训练及保存、生成歌词,下面分别给出各模块的功 能介绍及相关代码。

### 6.3.1 数据准备

本部分包括加载所需要的库与数据、数据处理和整理训练数据,如图 6-5 所示。

Product × Solutions × Open Source × Pricing	Q Search or jump to	Sign in Sign up
Chinese-poetry/chinese-poetry	D Notifications	: 💱 Fork 9.1k 🛱 Star 44.7k ≠
<> Code ⊙ Issues 100 I <sup>↑</sup> Pull requests 3 ☑ Discussions ⊙ Actions □ Wiki ⑦ Se	ecurity 🗠 Insights	
P master - P 1 branch Q1 tag	Code 👻	About
jackeyGao Update README.md 66fc88c or	The most comprehensive database of Chinese poetry Q 最全中华古诗词数据库, 唐宋两朝近14000位古诗人,接近5.5万首 唐诗加26万宋词,两宋时期1564位词人, 21050首词。	
Failed to load latest commit information.		
images		${\cal O}$ shici.store
loader		json ci poetry chinese tangshi
ank rank	chinese-poetry	
strains		ស្ថារ MIT license
五代诗词		- Activity
■ 元曲		☆ 44.7k stars
全部诗	<ul> <li>1.2k watching</li> <li>9.1k forks</li> <li>Report repository</li> </ul>	
四书五经		
■ 宋词		
■ 歯芬影	Releases	
御定全唐詩		
曹操诗集		· · · · · · · · · · · · · · · · · · ·
楚辞	Sponsor this project	
水墨唐诗		

图 6-5 诗词数据集

#### 1. 加载库与数据

相关代码如下。

import tensorflow as tf
import numpy as np
import glob
import json
from collections import Counter
from tqdm import tqdm
from snownlp import SnowNLP #主要实现繁体字转简体字
poets = []
paths = glob.glob('chinese-poetry/json/poet. \*.json') #加载数据

#### 训练古诗数据的具体样式如图 6-6 所示。



图 6-6 训练古诗数据样式

#### 2. 数据处理

相关代码如下。

```
for path in paths:
    data = open(path, 'r').read()
```