

※

**项目名称** 基于keras框架的yolo4的多视频检测： 含有目标识别记录，目标追踪电子门阀统计

### 项目安装

参考链接：<https://github.com/LeonLok/Multi-Camera-Live-Object-Tracking>

1. 在anaconda环境中创建新的python环境： `conda create -n`

`yolo4tf114 python=3.6`

2. 安装yolo4运行需要的包：

Tensorflow-GPU 1.14

Keras 2.3.1

opencv-python 4.2.0 --- 4.4

ImageZMQ

numpy 1.18.2 --1.19.0

Flask 1.1.1

pillow

`conda install matplotlib`

`pip install sklearn`

`pip install imutils`

如果有报错sklearn,则升级:

`pip install scikit-learn==0.22.2 -i`

<https://pypi.tuna.tsinghua.edu.cn/simple/>

3. 测试tensorflow是否可以正常调用GPU:

python:

`import tensorflow as tf`

`print("Num GPUs Available: ",`

`len(tf.config.experimental.list_physical_devices('GPU')))`

返回Num GPUs Available:1, 则说明tensorflow使用GPU正常

## 项目测试过程 LEARNINGS

### 1. 测试代码的过程:

直接在object\_counting 和traffic\_counting下直接flask run, 发现没有视频

测试opencv是否正常, 网上找个使用opencv的例子即可,发现正常

测试videoStream是否正常, 网上找个使用videoStream的例子即可,发现正常

跟踪错误发现没有yolo4.h5, 下载推荐的<https://github.com/Ma-Dan/keras-yolo4>, 并下载 里面的yolo4.weight, 利用convert.py生成yolo4.h5,

将h5拷贝到model\_data 下

测试ImageZMQ是否正常, 发现需要修改camera\_client\_0.py和camera\_client\_1.py, 启动 python camera\_client\_0.py

测试flask 是否正常

测试ImageZMQ转发的frame是否可以正常取到

最后, 发现 keras-yolo4,转换出来的模型和当前使用的模型的参数不同,需要把 self.model\_image\_size = (608,608), 从416 修改为608

### 2. 最终使用:

启动摄像头客户端, 发送视频到端口5555,5566 (此步骤是启动视频读取, 然后发送到指定的端口, 可以将此部署到内网机器上, 读取后发送到云服务器上, 目前这步骤在本项目中不需要)

```
conda activate yolo4new
```

```
cd D:\Work-yun\yolo-project\Multi-Camera-Live-Object-Tracking-master\Multi-Camera-Live-Object-Tracking-master\object_traffic_counting
```

```
python camera_client_0.py
```

```
python camera_client_1.py
```

启动网页端服务,接受视频,并检测视频

```
conda activate yolo4tf114
```

```
cd D:\Work-yun\yolo-project\Multi-Camera-Live-Object-Tracking-
```

master\Multi-Camera-Live-Object-Tracking-

master\object\_traffic\_counting

改变端口号启动

flask run -h 0.0.0.0 -p 5001

object\_counting 为计算视频中人的数量, 在此目录下启动则为检测人数

traffic\_counting 为检测通过视频线物体的数量的,在此目录下启动为检测方向,分别计数.

html中的摄像头配置:

```
<div class="img-container">
```

```
  <p align="center">Camera 2</p>
```

```
  
```

```
</div>
```

```
<div class="img-container">
```

```
  <p align="center">Camera 2 Counting</p>
```

```
  
```

```
</div>
```

```
<div class="img-container">
```

```
  <p align="center">Camera 2 Tracking</p>
```

```
  
```

```
</div>
```

#### 简化后的启动方式：

1. `conda activate yolo4tf114`
2. `cd` 到 `object_traffic_counting` 目录
3. 启动服务：`flask run -h 0.0.0.0 -p 5001`
4. 预测功能需要参考fbprophet的文档启动预测的后台服务。

#### 需要后期优化的地方

本项目基于yolo4，在window上跑起来基于GPU的识别速度一般，看视频有卡顿。最好能修改代码支持到yolo5。

macOS上使用此项目，如果是英伟达的显卡没有问题，可以使用GPU来识别视频，但如果是AMD的显卡，则需要安装基于keras的plaidml-keras框架，参考下面链接

<https://blog.csdn.net/renzibei/article/details/106174239>

<https://rustyonrampage.github.io/deep-learning/2018/10/18/tensorflow-amd.html>

但是我安装后发现，还是默认使用了cpu，需要继续研究怎么使用gpu或者升级代码支持yolo5，来解决在mac上的识别速度。