

项目名称：基于云机器人系统的物体识别

项目简介：

随着计算机性能的不断提升，计算机应用领域也越来越广泛。在计算机识别领域，不断涌现出越来越多优秀的识别算法，配合强大的硬件支持，物体识别已经真正走进了我们的生活。比如在图片识别方面，可以用于车型识别、商品识别等；在视频识别方面，可用于对道路通行车辆分析，行人分析跟踪、人群密度客流分析等。由此观之，物体识别技术在当前乃至未来一段时间内，仍然是比较主流的研究方向之一。另外，机器人开发作为当前热门领域之一，也是社会各界特别关注的热点。本项目将这两个领域结合在一起，以机器人开发平台为载体，在其基础上实现了对静态图片的识别以及动态视频中的物体识别。在识别准确率上达到基本应用要求，具有一定的实际意义。

项目报告

Github 链接：<https://github.com/Fyz01/objectDetectionRobo>

团队成员：

范耀中 15331071 —— 组长

范永业 15331072 —— 组员

傅城钢 15331084 —— 组员

团队目标：

学习 ROS 基本原理 ,基本掌握在 ROS 机器人软件平台上进行简单开发的能力。
在其基础上 ,学习开源的物体识别框架 API(Tensorflow Object Detection API) ,
在 ROS 上实现一个具有物体识别能力的云机器人系统。系统结构主要分为三部分：

1. 获取识别图像或视频的 Web 前端
2. 对识别信息进行处理与显示的机器人客户端
3. 进行物体识别信息计算的云服务器端

项目设计：

1. 前端

通过摄像头拍摄图片/视频，然后上传至机器人节点；
返回识别结果；

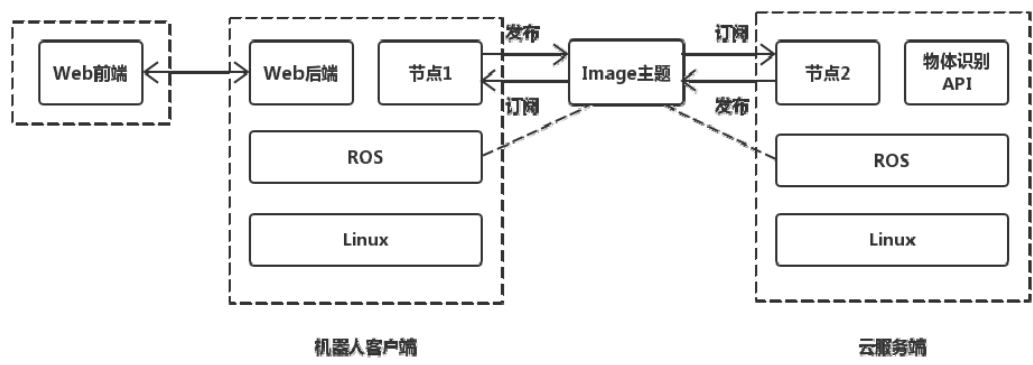
2. 机器人节点

接收来自前端的图片/视频，将收集到的数据发布到一个主题（topic）
上，让其他订阅该主题的主题可以接收到数据；
接收来自云服务器节点的识别结果；
将识别结果传递到前端；

3. 云服务器节点

通过订阅主题获得来自机器人节点的数据，对数据进行加工处理，然后
调用已有的物体识别 API 进行识别，将识别结果发布主题上。

系统原理图：



开发相关：

系统平台	Ubuntu-16.04、ROS-kinetic
Web 框架	Flask
物体识别 API	Tensorflow Object Detection
图像处理库	Opencv、Mapplotlib

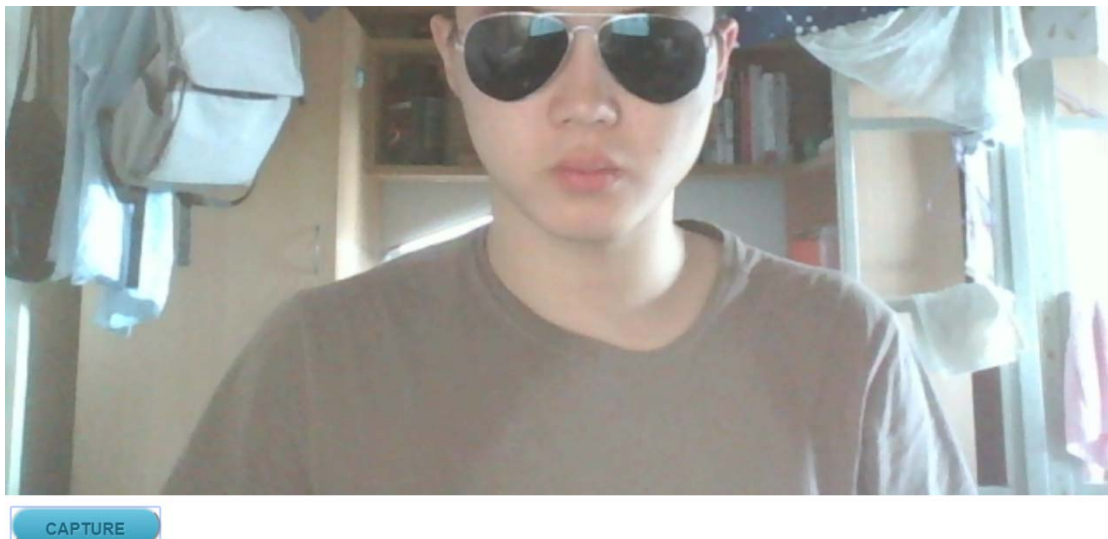
项目进度：

时间	工作内容
1~2 周	团队组建
3~4 周	ROS 基本原理学习、ROS 环境搭建（虚拟机）

5~7 周	项目主题确定
8~9 周	ROBOEYES 样例原理学习及重现
10~11 周	物体识别框架 API 学习、ROS 节点代码设计
12 周	中期汇报
13~14 周	实现云机器人的图像物体识别
15~16 周	实现云机器人视频（动态图像）识别
16~	总结

项目成果：

Web 前端界面



运行云服务器节点

```
This call to matplotlib.use() has no effect because the backend has already
been chosen; matplotlib.use() must be called *before* pylab, matplotlib.pyplot,
or matplotlib.backends is imported for the first time.

The backend was *originally* set to 'TkAgg' by the following code:
  File "cloud_server.py", line 45, in <module>
    import objectdetect
  File "/home/fan/catkin_ws/src/video_detection/scripts/objectdetect.py", line 1
5, in <module>
    from matplotlib import pyplot as plt
  File "/home/fan/.local/lib/python3.5/site-packages/matplotlib/pyplot.py", line
71, in <module>
    from matplotlib.backends import pylab_setup
  File "/home/fan/.local/lib/python3.5/site-packages/matplotlib/backends/__init_
.py", line 16, in <module>
    line for line in traceback.format_stack()

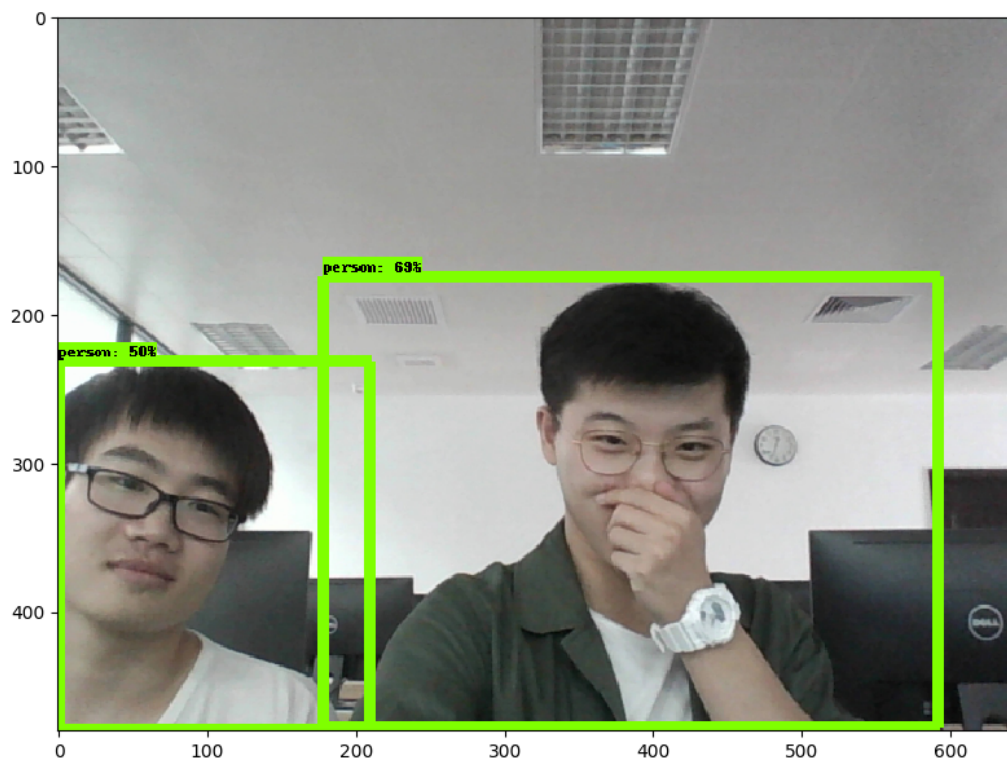
import matplotlib; matplotlib.use('Agg') # pylint: disable=multiple-statement
s
video detection start
[MoviePy] >>> Building video my_test_videos/video1_out.mp4
[MoviePy] Writing video my_test_videos/video1_out.mp4
59%|██████████          | 52/88 [06:04<04:12, 7.02s/it]
```

1. 图像物体识别

待检测图片



识别结果

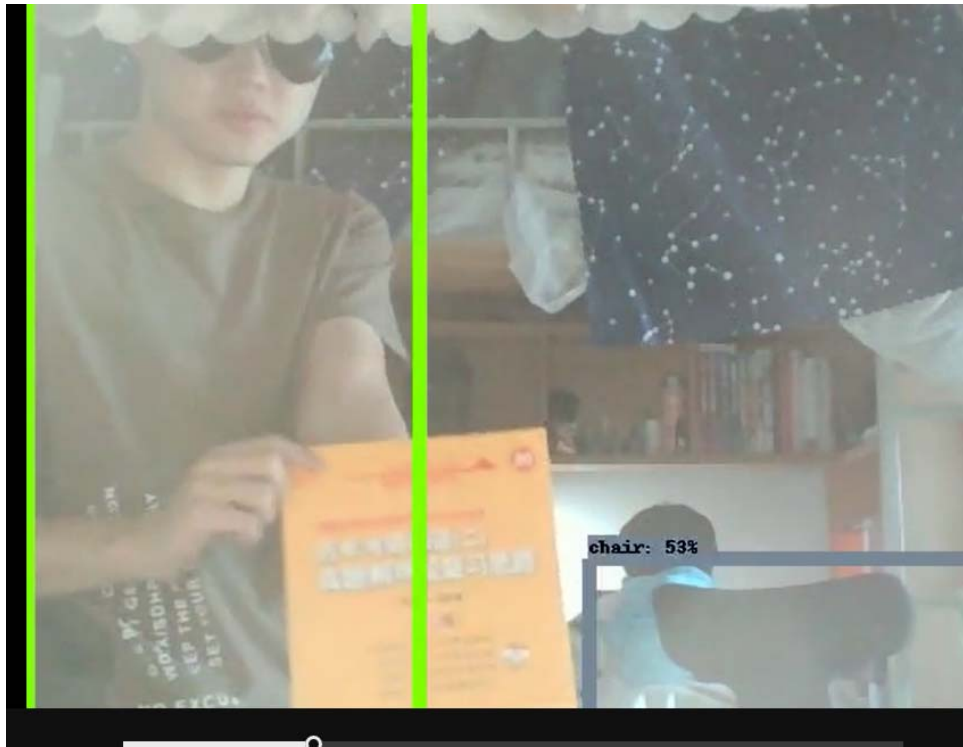


2. 视频物体识别

待检测视频



识别结果



项目总结：

1. 项目难点

- ROS 使用 python2.7 版本，Tensorflow 使用 python3.5 版本，如何将二者结合起来运行是一个较为麻烦的问题。
- 摄像头拍摄的图片需要经过加工处理，然后得到的数据才可以进行识别。所以处理图片的方式也是一个需要考虑的问题。
- 识别后的数据要返回到前端，确定返回路径，以及返回的数据格式，返回的方式，都花费了较长时间。

2. 待解决问题

- 模型识别速度较快，但是识别准确率低，如果想要进一步优化，可以考虑训练其他模型进行测试。

- 图片/视频数据的处理时间较长，在性能上仍有欠缺。
- 项目文件虽然正常打包，但是使用 rosrune 调用时却失败。所以只能通过 python 文件直接编译运行
- 当前系统机器人执行的节点操作较为简单，可在这一基础上尝试添加其他复杂功能。
- 缺少对异常情况的处理。
- 代码可扩展性较低。

团队分工：

范耀中	项目设计、代码编写、文档撰写	35%
范永业	项目设计、代码编写、测试、文档撰写	45%
傅城钢	项目设计、代码编写	20%