鹰眼 AI 技术文档（后端）

SEU-Monash EES Group

一、环境配置

1. Python == 3.6.13
2. Pytorch == 1.10.0
3. Opencv == 3.4.2
4. Cudatoolkit == 11.3.0
5. Numpy == 1.17.0

二、模型训练

（一）数据预处理

1. 对采集到的数据集进行标注，使用软件 LabelImg 即可（教程：https://blog.csdn.net/python_pycharm/article/details/85338801）
2. 将 Img 放入 ./data/Images 文件夹中，将同名 xml 文件放入 ./data/xml 文件夹中
3. 根据想要识别的类别修改 ./data 文件夹中 .data 与 .names 文件内容
4. 使用 ./data 内的脚本 makeTxt.py 划分数据集（train/valid/test）
5. 使用 ./data 内的脚本 voc_label.py 生成数据集对应的 txt 文件（yolo 需求）
6. 根据类别数修改 yolov3-tiny-3cls.cfg 的配置，主要修改 classes = 类别数，filter=（5+classes）*3

（二）模型训练

在 train.py 文件中根据服务器配置修改 batch_size, epoch, numbers of worlers,

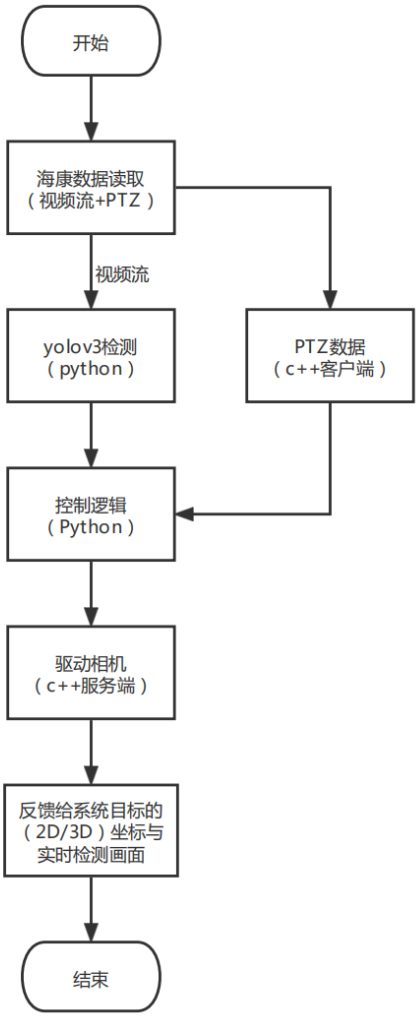
cpu/gpu 等内容，修改后可直接运行 train.py。

（三）测试集比对

模型在训练之后生成 best.pt 与 last.pt 两种权重，根据模型训练情况选择权重（一般选 best.pt），在 test.py 修改相关路径并运行，查看模型训练效果。

三、鹰眼 AI 流程

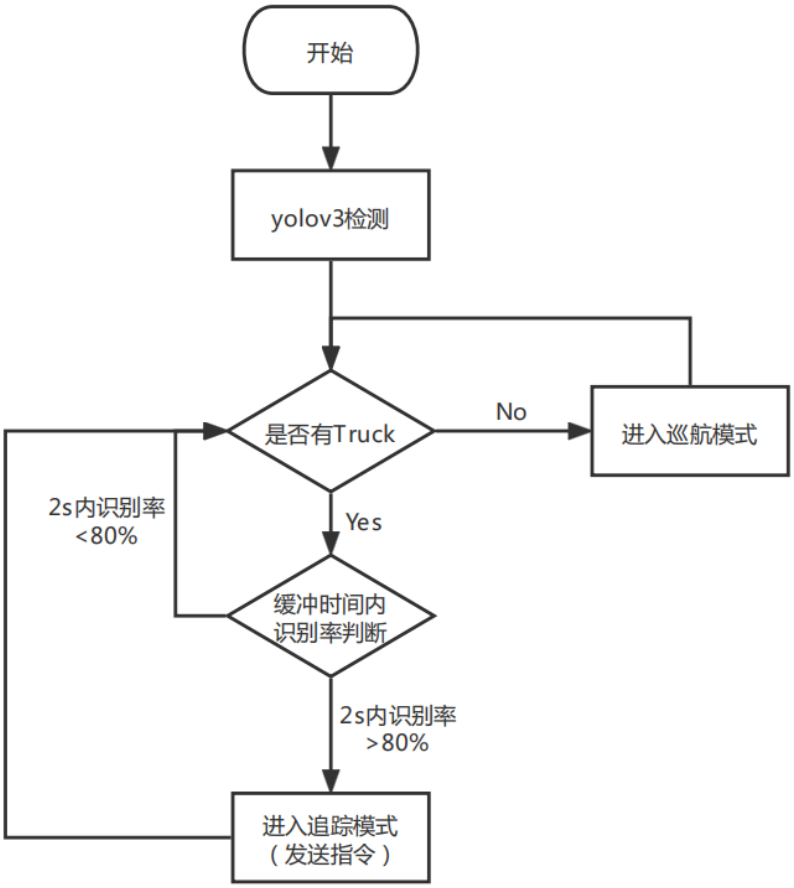
（一）项目整体流程



该系统主体程序在 detect.py 中，为了达到实时性的要求，该系统分为三个线程同时运行：

1. Thread1: yolo 目标识别，跟踪算法，2D/3D 坐标解算
2. Thread2: python 作为 TCP 服务端，接收相机的 PTZ 数据
3. Thread3: python 作为 TCP 客户端，向相机发送实时的控制指令

（二）主体控制逻辑



该部分为 Thread1 的内容主要分为目标识别，跟踪算法，2D/3D 坐标解算

1. **目标识别**：该模型使用 yolov3 模型，实时性较好，应用于项目中识别效率可达每秒 25 帧。具体识别原理与模型结构请参考 yolov3 的论文（本模型使用 tiny 版），结合论文并进行代码仿真可快速理解该部分代码。

2. **跟踪算法**：该算法代码位于 Track_and_position.py 内的 Track 函数，该函数作用为当检测到需求目标（卡车）时，解算出目标相对于图像中心的位置差，从而生成对相机的控制指令，使目标始终保持在图像中心，方便坐标解算。

3. **2D/3D 坐标解算**：
   2D 坐标：解算仅根据相机返回的 PTZ 值与相机的世界坐标系，应用三角函数原理解算出目标的平面坐标系。该部分代码位于 Track_and_position.py 内的 position_2D 函数。
   3D 坐标：首先通过小孔成像原理计算得到目标相对于相机的深度，然后再结合 PTZ 值与相机的世界坐标系，应用三角函数原理解算出目标的世界坐标系。

该部分代码位于 Track_and_position.py 内的 position_3D 函数。