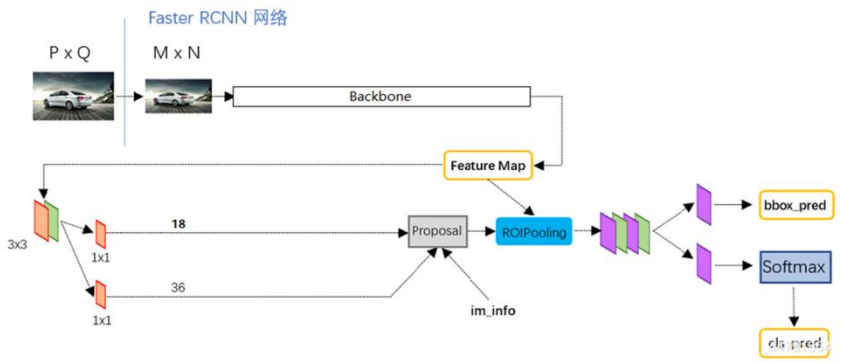
目标检测 CV-Object Detection

项目研究报告

一、原理介绍

Two-stage



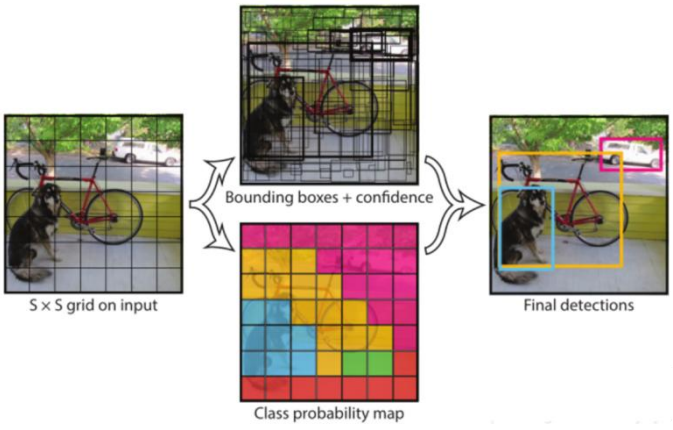
Faster-RCNN 作为一种 two-stage 的算法， Faster RCNN 整体框架包括 4 部分：

1. 使用 VGG16 或者其他成熟的图片分类模型提取图片特征(feature map)． 2. 将图片特征喂入 RPN(Region Proposal Network)网络得到 proposals.

3. 将上两步的结果:图片特征和 proposals 喂入 RoI Pooling 层得到综合的 proposals 特征.

4. 4. 根据 poposals 特征预测物体的 bounding box 和物体的类别.

One-stage



Two stage 的检测网络，相当于在 One stage 的密集检测上增加了一个稀疏的预测器。对于 YOLO v4 而言，其整个网络结构可以分为三个部分。分别是：

1. 主干特征提取网络 Backbone，对应图像上的 CSPdarknet53.
2. 加强特征提取网络，对应图像上的 SPP 和 PANet.
3. 预测网络 YOLOHead，利用获得到的特征进行预测.

（第一部分主干特征提取网络的功能是进行初步的特征提取，利用主干特征提取网络，我们可以获得三个初步的有效特征层。第二部分加强特征提取网络的功能是进行加强的特征提取，利用加强特征提取网络，我们可以对三个初步的有效特征层进行特征融合，提取出更好的特征，获得三个更有效的有效特征层。第三部分预测网络的功能是利用更有效的有效特整层获得预测结果。）

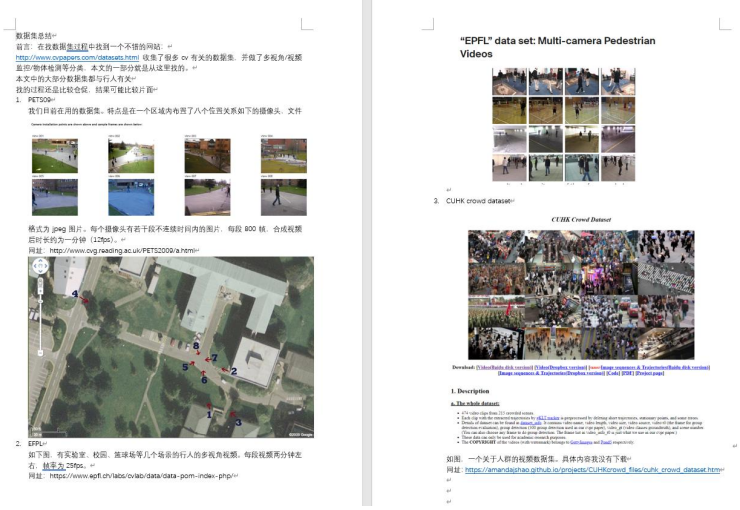
移动设备因硬件资源和算力的限制。因此更轻量的神经网络被考虑。Mobilenet 系列网络可用于进行分类，其主干部分的作用是进行特征提取. 我们可以使用 Mobilenet 系列网络代替 YOLO V4 当中的 CSPdarknet53 进行特征提取.

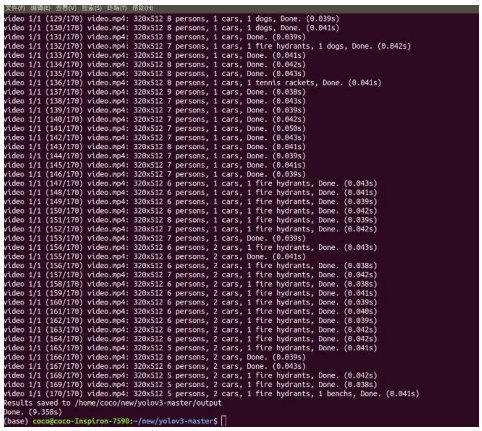
二、研究情况

我们大创团队尝试了如下探索实验：

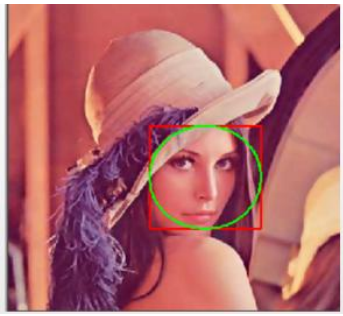
- 搜集了网上多个开源的图片、视频数据集，如 PETS 2009，EPFL 等，作为系统设计的输入，用来为之后测试系统框架的性能做准备。
- 利用 Socket 搭建了一个完备的通信框架，实现了服务器与客户端间的文件传输功能。
- 初步尝试了目标检测算法，尝试了多种目标检测网络，对视频进行逐帧的图像识别
- 通过压缩图片质量、更改线程数控制算力等方式，研究了多个框架在不同环境下的的运行效率。
- 在图像识别后通过剪裁无关部分的方式来达到压缩图片的效果，并通过实验测试了此做法的压缩效率。

数据集：主要基于 VOCdevkit 数据。同时，我们搜集了网上多个图片、视频数据集，用于后面进一步测试框架的性能。

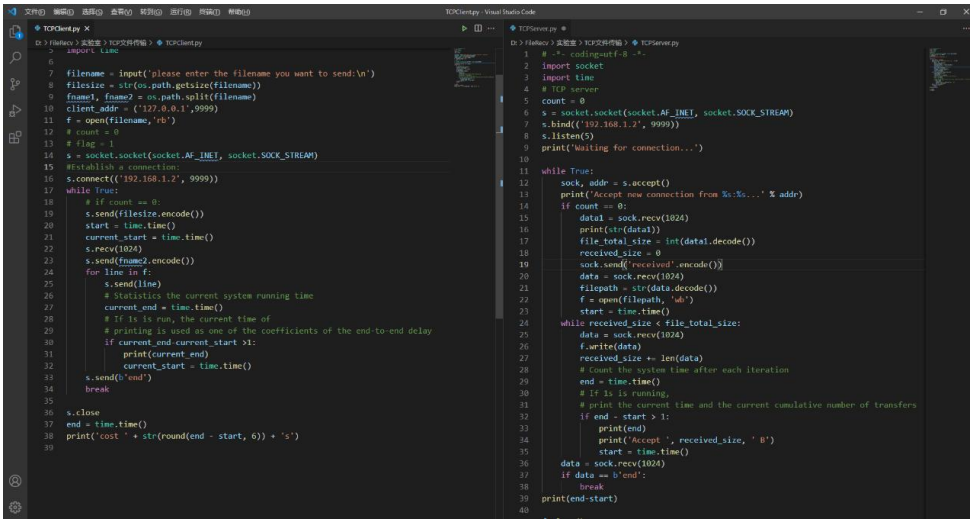




初步尝试：我们用普通 OpenCV 分类器和 YOLO V4 测试了对图片及视频逐帧进行物体识别的效果，为后面进一步的优化做铺垫。

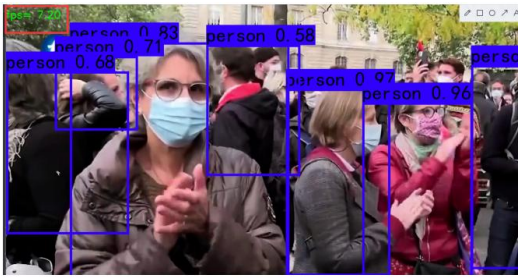


通信框架：我们利用 python 的 Socket 模块搭建了基于 TCP 协议和 UDP 协议的通信框架，实现了客户端与服务器端间的连接建立与文件传输功能，并能记录这些操作所花费的时间。

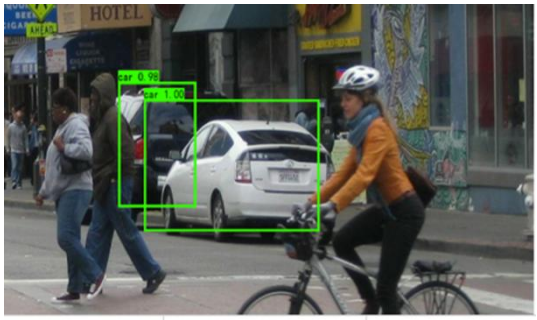


不同阶段算法对比：我们尝试在电脑上运行了 Mobilenet（基于 YOLO V4）与 Faster RCNN 两种算法框架，对视频进行逐帧的图像识别并比较了其识别准确度与时间消耗。

Faster Rcnn FPS



Mobilenet FPS



Resolution	Accuracy (the first car)	Accuracy (the second car)	Executing time (s)
176x144	0.99	0.68	1.4447
320x240	1.00	0.71	1.4506
640x480	1.00	0.95	1.4590
1024x768	1.00	0.99	1.4644
2400x1800	1.00	0.98	1.4750

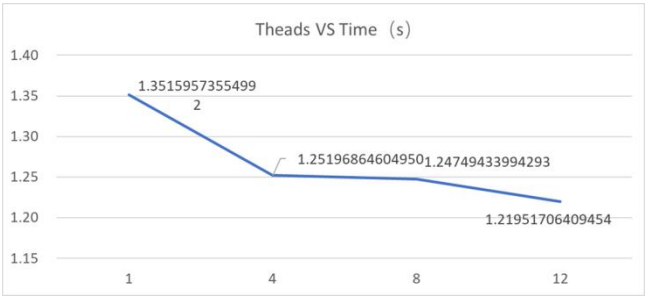
Faster RCNN

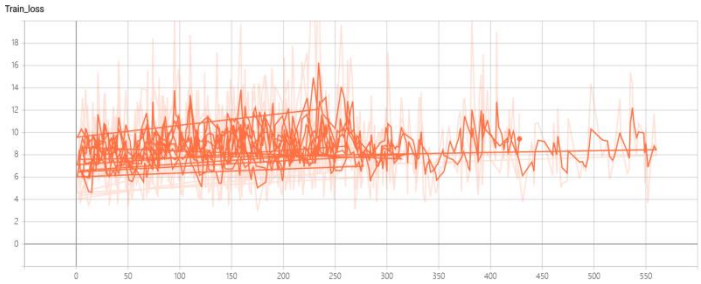
Resolution	Accuracy (the first car)	Executing time (s)
176x144	0.81	0.9227
320x240	0.95	0.9433
640x480	0.98	0.9787
1024x768	0.99	0.9824
2400x1800	0.99	0.9921

Mobilenet-YOLOV4

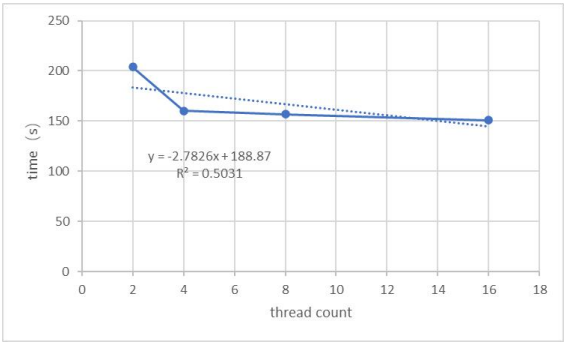
通过实验，我们对比 One-stage 和 Two-stage 的目标检测算法得到一系列结论。首先 Faster RCNN 在不同分辨率下检测准确度均高于 Mobilenet-YOLO V4。同时还能检测到被部分遮挡的第二辆汽车。但是值得一提的是它在低分辨率时，还将其他物体误检测为汽车。同时检测时间也是 Mobilenet-YOLO V4 的 1.5 倍。

算力控制：我们通过更改电脑线程数控制算力等方式，研究了在不同情况下 Mobilenet 及 Faster RCNN 两个框架的识别准确度及运行时间，得到了多组数据。





可以看出，整体上随着线程数增加，该框架检测用时减少，在线程数为 1 时变化较显著，4 与 8 之间则较不明显。由此可见算力的贡献存在边际效应。



thread count	time (s)
2	204
4	160
8	157
16	151

当计算机的 CPU 线程受到约束时，每个训练阶段都呈现出不同的时间变化。然而，需要注意的一点是，即使线程数量相同，训练一个时代所需的时间也可能会有所不同。当电脑一开始开始训练时，速度会变慢，性能没有完全释放，需要运行一段时间。

图片提取：我们尝试在图像识别后通过剪裁无关部分的方式来达到压缩图片的效果，并通过实验测试了此做法的压缩效率。

Size of frame before compressing (Byte)	Size of frame before compressing (Byte)	Reduced proportion
66,720	10,437	84.36%
447,571	16,474	96.32%
36176	15,783	56.37%

在图像识别后，为了进一步减少不必要的数据量的传输。我们考虑将目标提取出来。通过编程提取并裁剪图片，我们发现可以显著起到压缩的效果。

三、总结

我们主要聚焦对比多种不同的目标检测模型，包括更精确的二阶段检测算法、速度更快的单次目标检测器近来，人们开始着重研究 Anchor-Free 的单次目标检测器，试图在不提取 ROI 的情况下得到更高的检测准确率，未来，研究将继续朝着将二者统一的方向发展，以求达到准确率与速度的平衡。

参考文献

1.Ren S, He K, Girshick R, et al. Faster R- CNN: Towards real-time object detection with region proposal networks[J]. Advances in neural information processing systems, 2015, 28.
2.. Howard A G, Zhu M, Chen B, et al. Mobilenets: Efficient convolutional neural networks for mobile vision applications[J]. arXiv:1704.04861, 2017.
3. Bochkovskiy A, Wang C Y, Liao H Y M. YOLOv4: Optimal speed and accuracy of object detection[J]. arXiv:2004.10934, 2020.