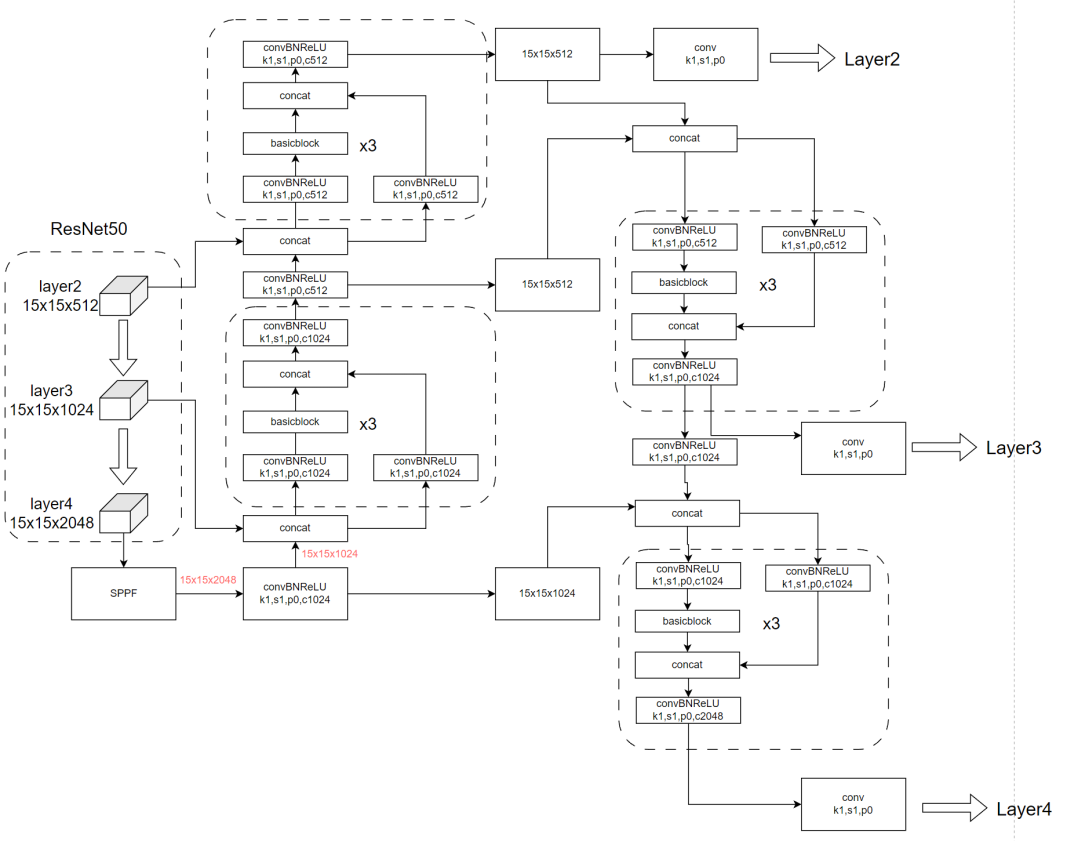
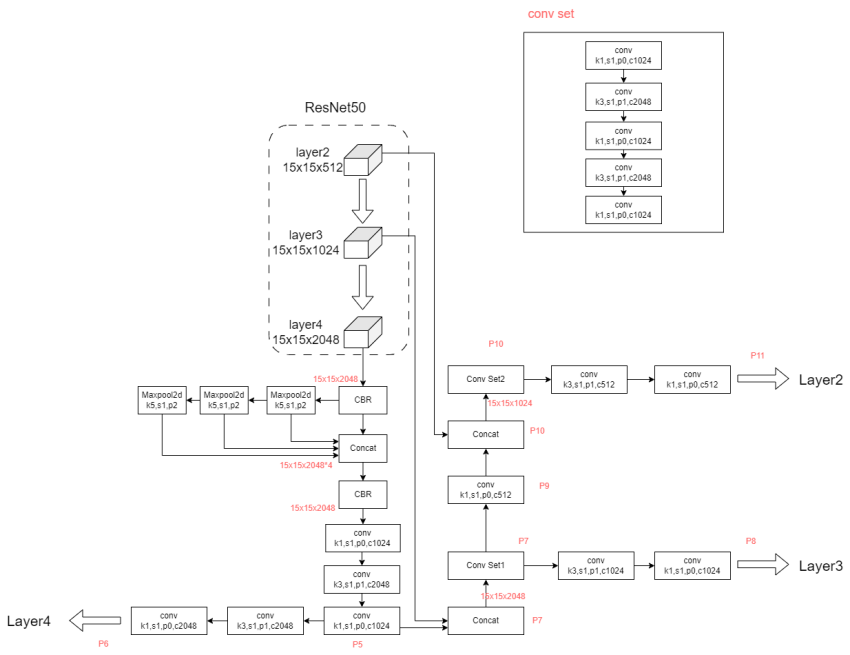
张云鑫实习工作总结

1. 工作摘要

* 调研了近几年单目标跟踪的文献，运行了其中几个算法demo
* 解决了121服务器多卡训练失效的问题
* 复现了siamrpn++算法，并与卡尔曼滤波算法相结合
* 实现了视频流中实时切换跟踪目标的demo
* 算法优化：
* 为提升对车辆目标的跟踪效果，从vid数据集的val中选择73个车辆视频训练siamrpn++网络
* 设计了新的backbone，添加SPPF和PAFPN模块进行特征融合，并在vid数据集上进行了训练



* 设计了resnet\_cbam的backbone，并在vid数据集上进行了训练
* 在resnet\_cbam基础上添加sppf和fpn



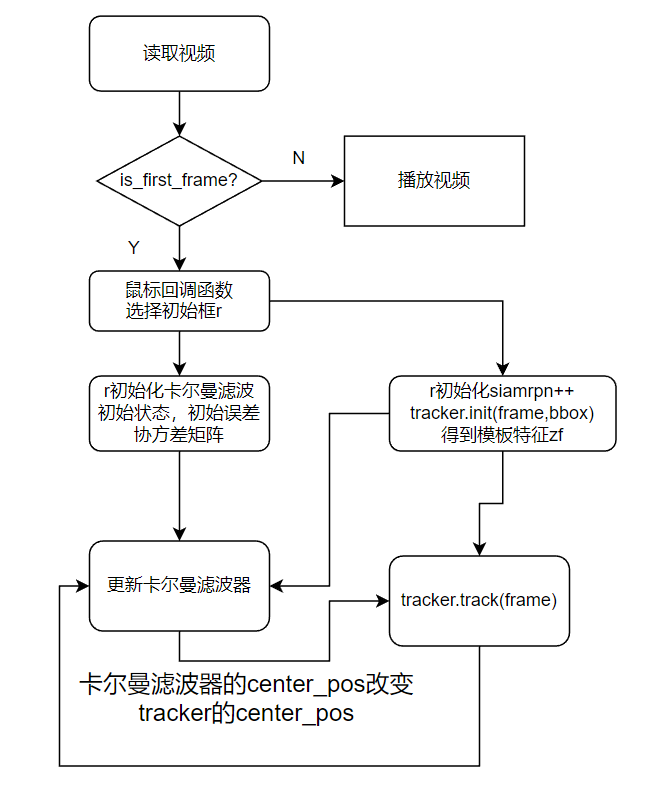
1. 技术细节

* 121服务器多卡训练问题：

使用分布式时，在梯度汇总求平均的过程中，各主机之间需要进行通信，Pytorch提供的多卡分布式训练，torch.distributed支持NCCL,GLOO,MPI三种通信后端。英伟达显卡默认为NCCL，但是121服务器的通信后端为GLOO。在./pysot/utils/distributed中修改dist\_init函数。

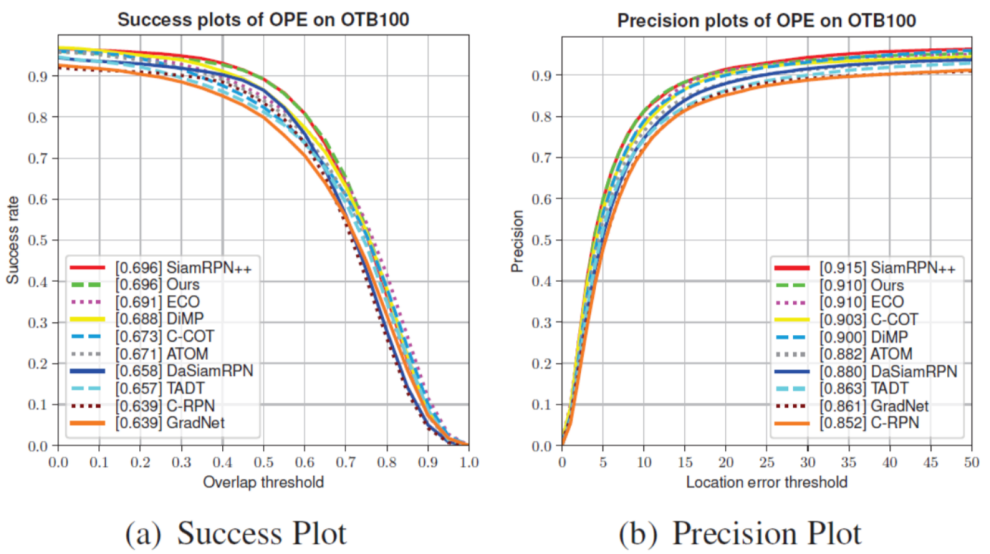
* 整体流程：

整体流程如下图，通过opencv读取本地视频，两个测试视频分别为行人和车辆（768x576/1280x720）。Siamrpn++的运算在GPU进行，卡尔曼滤波计算在CPU上进行。一帧处理时间平均为36ms。



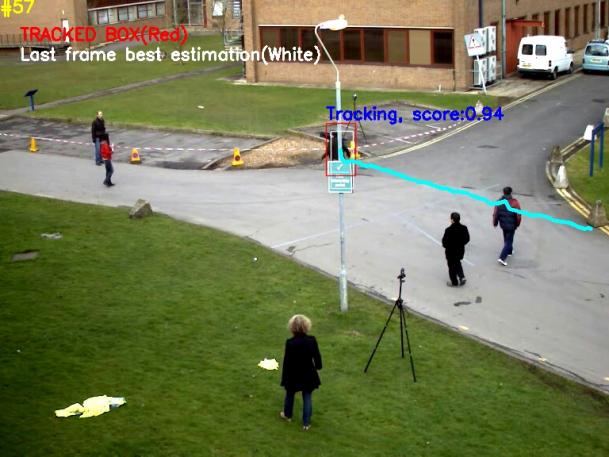
* 为什么选择siamrpn++?

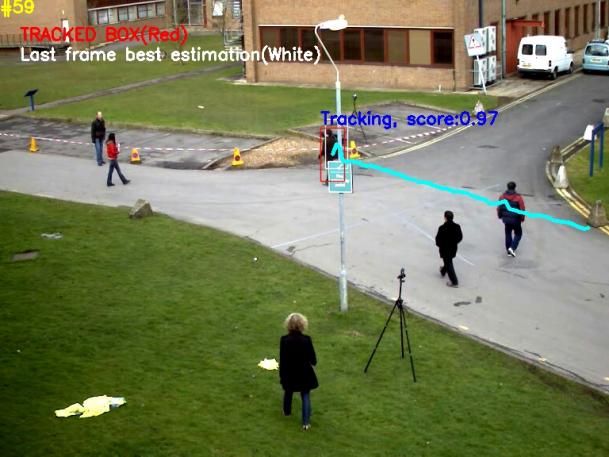
1. 之前的算法都是用浅层网络提取特征，导致特征表征性不强，最后的跟踪效果不好。Siamrpn++解决了深层网络的padding导致的偏移问题，并使用resnet50的layer2,layer3,layer4输出特征分别进入rpn模块，最后在输出上使用加权求和，提升了跟踪效果。之后的算法基本借鉴siamrpn++的思路，大多在head部分进行改进，且在siamBAN(2020CVPR)论文中，siamrpn++在OTB100数据集上的表现良好。
2. Transformer系列的网络需要大量的数据进行训练才能超越CNN的效果，训练难度大；自身对transformer的了解不够多，不好做改进；从demo的效果上来看跟踪效果不如siamrpn++。



* 为什么要结合卡尔曼滤波

卡尔曼滤波通过目标的运动信息进行轨迹预测，siamrpn++通过目标的外观信息进行轨迹预测，两者结合可以提高跟踪任务的性能和鲁棒性。通过siamrpn++输出的目标框作为卡尔曼滤波器的观测值，更新卡尔曼滤波的参数；同时卡尔曼滤波输出的目标框中心点作为siamrpn++搜索区域（Xf）的中心点，更新siamrpn++的跟踪框。【红框为siamrpn++结果，白框为kalman结果】

1. 性能对比

Success:边框重叠率大于给定阈值的视频帧的百分比曲线，值为曲线面积。

Presicion:中心位置误差小于给定阈值的视频帧的百分比曲线（20像素）

|  |  |  |
| --- | --- | --- |
| backbone name | Success | Precision |
| Resnet\_cbam\_e20 | 0.597 | 0.803 |
| Resnet\_origin\_e20 | 0.575 | 0.786 |
| Resnet\_cbam\_e36 | 0.569 | 0.774 |
| Resnet\_origin\_e36 | 0.557 | 0.771 |
| Resnet\_pafpn\_e39 | 0.521 | 0.701 |
| Resnet\_pafpn\_e20 | 0.497 | 0.660 |

通过表格可以看到resnet\_cbam的效果最好，而resnet\_pafpn的效果较差，原因可能是因为yolov5中的pafpn主要用于处理多尺寸输出，而siamrpn++中输出层的尺寸是相同的。

1. 运行结果







参考文献

1. Li B, Yan J, Wu W, et al. High performance visual tracking with siamese region proposal network[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2018: 8971-8980.
2. Li B, Wu W, Wang Q, et al. Siamrpn++: Evolution of siamese visual tracking with very deep networks[C]//Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2019: 4282-4291.
3. Wang Q, Zhang L, Bertinetto L, et al. Fast online object tracking and segmentation: A unifying approach[C]//Proceedings of the IEEE/CVF conference on Computer Vision and Pattern Recognition. 2019: 1328-1338.
4. Chen Z, Zhong B, Li G, et al. Siamese box adaptive network for visual tracking[C]//Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2020: 6668-6677.
5. Guo D, Wang J, Cui Y, et al. SiamCAR: Siamese fully convolutional classification and regression for visual tracking[C]//Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2020: 6269-6277.
6. Lin L, Fan H, Zhang Z, et al. Swintrack: A simple and strong baseline for transformer tracking[J]. Advances in Neural Information Processing Systems, 2022, 35: 16743-16754.
7. Cui Y, Jiang C, Wang L, et al. End-to-end tracking with iterative mixed attention[C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA. 2022: 18-24.
8. Dosovitskiy A , Beyer L , Kolesnikov A ,et al.An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale[J]. 2020.DOI:10.48550/arXiv.2010.11929.
9. Liu Z, Lin Y, Cao Y, et al. Swin transformer: Hierarchical vision transformer using shifted windows[C]//Proceedings of the IEEE/CVF international conference on computer vision. 2021: 10012-10022.
10. 卡尔曼滤波：https://github.com/liuchangji/kalman-filter-in-single-object-tracking
11. Yolo系列：<https://github.com/ultralytics>
12. Yolo讲解：<https://www.bilibili.com/video/BV1yi4y1g7ro>

附录

训练模型：

1. cd pysot/experiments/siamrpn\_r50\_l234\_dwxcorr/config.yaml

修改使用的backbone

1. 1)cd pysot/pysot/core/config.py

2)修改\_\_C.DATASET.NAMES参数，选择使用的训练集

[VID训练集路径：/mnt/e/zhangyunxin/training\_dataset/vid]

3)修改\_\_C.TRAIN.LOG\_DIR,\_\_C.TRAIN.SNAPSHOT\_DIR

4)cd pysot/experiments/siamrpn\_r50\_l234\_dwxcorr

终端输入训练脚本CUDA\_VISIBLE\_DEVICES=0,1,2 python -m torch.distributed.l aunch--nproc\_per\_node=3--master\_port=2333 ../../tools/train.py --cfg config.ya ml

测试模型：

1. cd pysot/experiments/siamrpn\_r50\_l234\_dwxcorr/config.yaml

修改使用的backbone

1. python pysot/tools/test.py

--snapshot：模型权重路径

--dataset：测试集名称

--config：config.yaml路径

画图：

1. python pysot-toolkit/bin/eval.py

--dataset：测试数据集(otb100)

--tracker\_result：存放测试结果txt的路径

--trackers：各跟踪器(tracker\_result文件夹)的名称

--num：线程数

--show\_video\_level：是否显示跟踪结果

--vis：是否画曲线图

视频跟踪：

1. cd pysot/experiments/siamrpn\_r50\_l234\_dwxcorr/config.yaml

修改使用的backbone

1. Python kalman-siamrpn/demo.py

--config：模型config文件

--checkpoint：模型权重

--video\_name：测试视频路径

--outpath：视频的保存路径

--imgpath：图片的保存路径