

# DLP第三次实验报告

## ——TinaFace[3]复现及改进

韩广意PB20030835

### 1 Tinaface复现

#### 1.1 网络结构

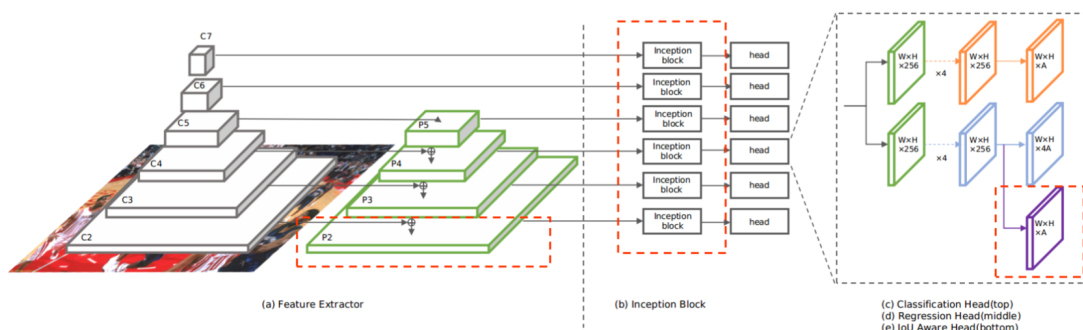


Figure 1: TinaFace的模型架构。(a)特征提取器:ResNet-50和6级特征金字塔网络,P4、P5使用了DCN网络,提取输入图像的多尺度特征。(b)Inception模块增强接受野。(c)分类头:5层FCN用于锚框的分类。(d)回归头:5层FCN,用于锚框回归到地真框。(e) IoU感知头:用于IoU预测的单个卷积层。

#### 1.2 实现细节

##### 1.2.1 原论文实验条件

作者在 `mmdetection` 库的基础上将工程结构进行了一定修改<sup>1</sup>, 使其适应TensorRT。作者论文的实验部分提到了以下对复现结果有影响的配置,我根据所能使用的硬件条件进行了自己的选择。

##### 归一化

作者考虑到批量归一化(BN)随着批处理规模的减小性能会下降,当批处理规模小于4时,由于批统计估计不准确而导致模型性能下降。同时考虑到大显存的gpu并不广泛使用,提供了将所有BN层网络用Group Normalization(GN)替代的方案。在复现过程中也使用GN方法。

##### 训练配置

作者在3个GeForce GTX 1080 Ti上使用批量为3\*4的SGD优化器(动量0.9,重量衰减 $5e-4$ )对模型进行训练。学习速率的过程是在630个epoch中每30个epoch用余弦衰减规则从 $3.75e-3$ 退火到 $3.75e-5$ 。在前500次迭代中,学习率从 $3.75e-4$ 线性升到 $3.75e-3$ 。

<sup>1</sup>代码仓库地址:<https://github.com/Media-Smart/vedadet/tree/main/configs/trainval/tinaface>

## 测试设置

作者在评估环节为了提高评分,最后的评估结果是使用了测试时增强(Test Time Augmentation, TTA)的,但是考虑到TTA只是提高评分的tricks,不能说明模型设计的优越性且花费更多的计算资源,我在复现评估时没有使用TTA方法,只将baseline的评估结果与作者对应的结果进行对比。

## 环境配置

Linux;Python 3.7+;PyTorch 1.6.0 or higher;CUDA 10.2 or higher

## 1.3 复现过程

### 环境配置

最终在Bitahub上使用的环境为 Ubuntu;Python 3.6;PyTorch 1.6.0;CUDA 11.0

因为需要对C++和Cuda进行编译,所以对torch和cudatoolkit的版本要求并不像作者所说的那么宽松。torch的版本不能过高,在高版本中的torch中移除了THC/THC.h文件(替换为ATen/ATen.h);cuda的版本不能低,不然也无法编译。因为不懂dockerfile,长时间无法构建出一个可以使用合适版本的nvcc编译的镜像,后来在助教的帮助下解决。

### 数据集加载

在本地运行时数据集是以小文件存储读取的,但是上传到Bitahub时管理员告知需要对图片进行聚合,所以将数据集聚合成lmdb数据集,但是似乎自己写的lmdb数据集的读取过于简单粗暴了,在本地上训练一个epoch需要大概20分钟,但是到Bitahub变成30分钟了,观察GPU的Memory情况,发现波动较大,初步推测是数据加载成为瓶颈,但是也不排除其它硬件条件差异和版本差异影响。

### 训练过程

在两块1080Ti上训练,每块batchsize = 4,其余设置未更改。因为数据集较大,训练一个epoch大致要30min,如果真的要训练满630个epoch消耗算力过多,为了不占用其他同学的资源,我训练了105个epoch,消耗算力70点。

## 1.4 复现结果分析

模型	epochs	AP(easy)	AP(middle)	AP(hard)
作者(有TTA)	630	0.970	0.963	0.934
作者(无TTA)	630	0.963	0.957	0.930
hgy(无TTA)	105	0.940	0.931	0.899

### 1.4.1 loss曲线

见左下 Figure 2,可以发现在epoch = 105时,损失函数下降趋势变缓但是仍在逐渐下降,反映到训练效果上应该是:较容易学习的特征已经学习完成,但是较难学习的内容仍在学习中。推测WIDER FACE 数据集中测试结果应该是easy、middle、hard部分距离作者最终成品的差距越大,验证结果确实如此,见上。



Figure 2: loss

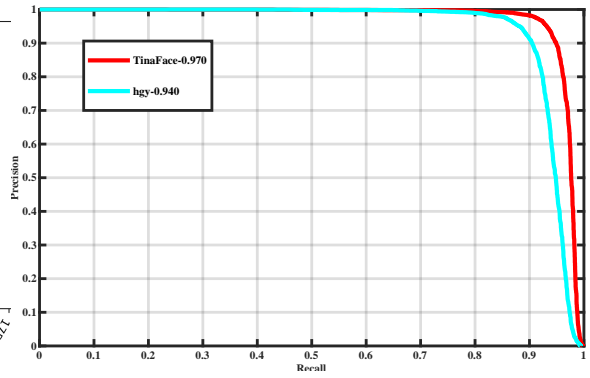


Figure 3: easy

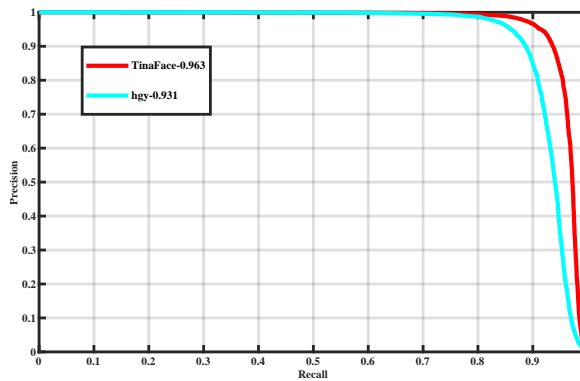


Figure 4: middle

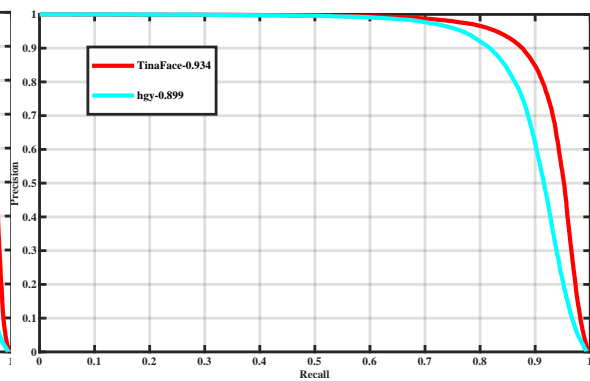


Figure 5: hard

#### 1.4.2 PR曲线

easy、middle、hard三个验证集的子部分的PR曲线见上

## 2 网络结构改进

### 2.1 思路来源

Table 1: AP performance on WIDER FACE validation subset

Baseline	DIoU	Inception	IoU-aware	DCN	TTA	Easy	Medium	Hard
✓	-	-	-	-	-	0.959	0.952	0.924
✓	✓	-	-	-	-	0.959	0.952	0.927
✓	✓	✓	-	-	-	0.958	0.952	0.928
✓	✓	✓	✓	-	-	0.963	0.955	0.929
✓	✓	✓	✓	✓	-	0.963	0.957	0.930
✓	✓	✓	✓	✓	✓	<b>0.970</b>	<b>0.963</b>	<b>0.934</b>

Figure 6: Ablation experience

观察上面作者所做的实验,可以发现在加入 IOU-aware 分支后 AP 提升较大,作者做出的改进主要是在head部分的,boneback和neck部分的提升并不大。如果继续改进 head 的话,看了看最近的方向是 anchor-free 的,但是如果将头部换成anchor-free的,似乎就是在复现 FCOS[2] 之类的网络。所以想保留作者的改进,来对neck部分进行改进,这里使用了 EfficientDet[1] 的思路,将普通

的FPN换成BiFPN。

1. 删除那些只有一条输入边的节点。它对旨在融合不同特征的特征网络的贡献很小,删除它网络影响不大,同时简化了双向网络
2. 如果原始输入节点和输出节点处于同一层,在原始输入节点和输出节点之间添加一条额外的边
3. 带权特征融合: 学习不同输入特征的重要性,对不同输入特征有区分的融合。设计思路: 传统的特征融合往往只是简单的特征图叠加/相加等合并操作,而不同的输入特征图具有不同的分辨率,对融合后特征图的贡献也是不同的,因此简单的对他们进行相加或叠加处理并不是最佳的操作。可以使用一种简单而高效的加权特融合的机制,  $\omega$  是学习到的参数,用于区分特征融合过程中不同特征的重要程度

$$O = \sum_i \frac{\omega_i}{\epsilon + \sum_j \omega_j} I_i \quad (1)$$

BiFPN = 新型加强版的PANet(重复双向跨尺度连接) + 带权重的特征融合机制

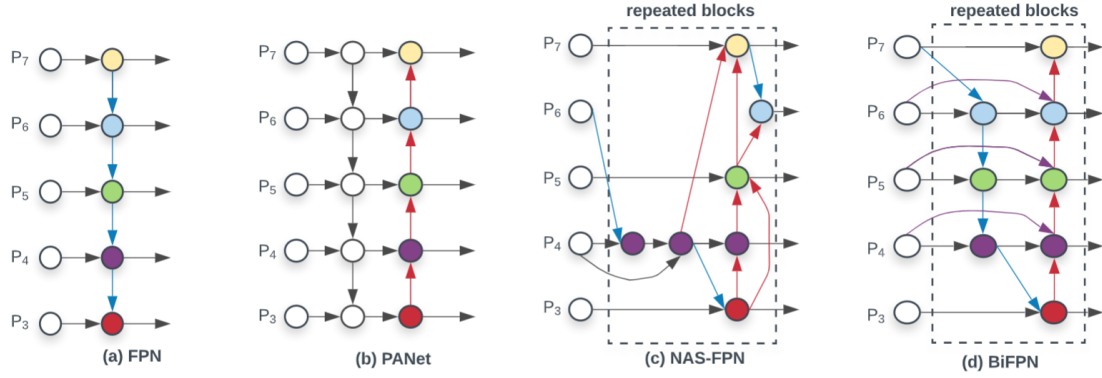


Figure 7: FPN及其变体

## 2.2 实现细节

将 neck 部分原来的 FPN 部分替换为 BiFPN，可能是因为结点数、连接数以及要学习参数的增多，按照原来的batchsize训练开始爆显存，开始时以为是新加的Bifpn.py有bug，后来发现简单的减小batchsize就可以解决问题。最后使用batchsize = 2 在两块1080Ti上训练了75轮(同样考虑算力问题)

## 2.3 实验结果分析

模型	epochs	AP(easy)	AP(middle)	AP(hard)
TinaFace(fpn)	630	0.963	0.957	0.930
hgy(fpn)	105	0.940	0.931	0.899
repro(fpn)	75	0.942	0.934	0.0.891
bifpn(bifpn)	105	0.955	0.946	0.914

### 2.3.1 loss曲线

见左下 Figure 8,两者训练过程中loss的波动频率相似，但是neck更换bifpn的网络的波动幅度较大，可以达到更低的loss水平

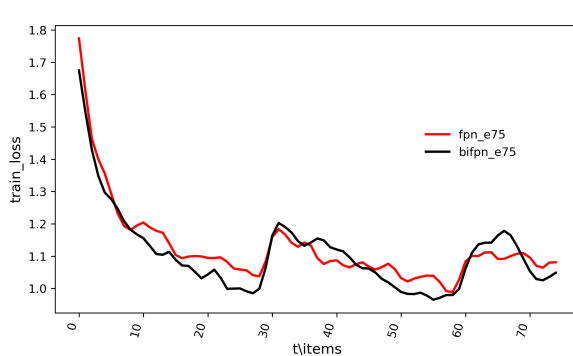


Figure 8: train loss

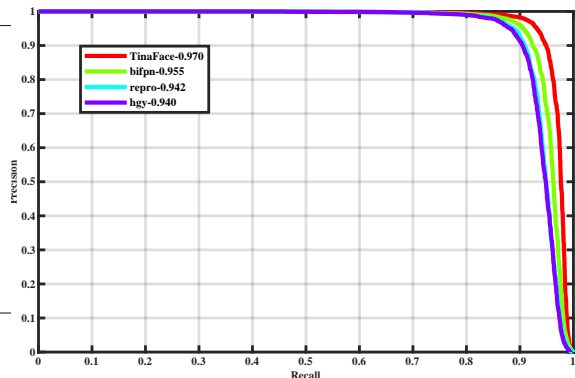


Figure 9: easy

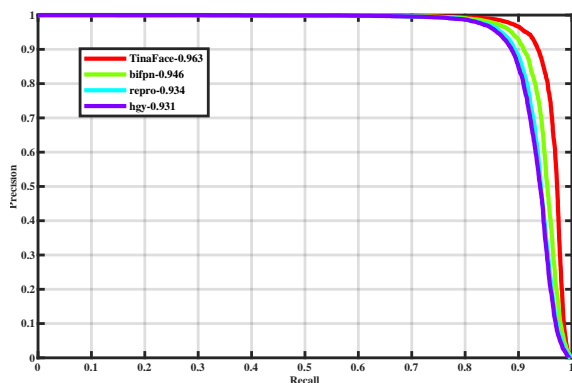


Figure 10: middle

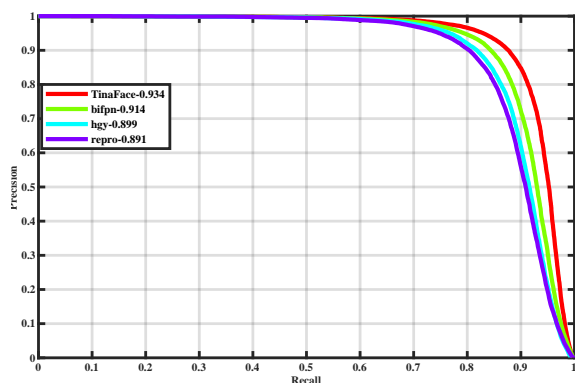


Figure 11: hard

### 2.3.2 PR曲线

easy、middle、hard三个验证集的子部分的PR曲线见上。标注TinaFace代表作者630轮neck为FPN，标注hgy代表105轮neck为FPN，标注repro代表75轮neck为FPN，标注bifpn代表105轮neck为BiFPN。

BiFPN为neck的模型在相同轮数(75 epochs)时优于原模型；有趣的是原模型在75轮时在easy、middle部分优于105轮时的情况，105轮时应该是为了学习hard部分，放弃了一部分的easy和middle的AP得分。

## 3 实验总结

1. 要多关注一些其它的研究方向，吸收前人的成果，站在巨人的肩膀上才能看的更远。
2. 大模型、大数据集是机器学习的潮流，找要复现的paper时发现几乎全部是大模型大数据集的，需要一定的资源才能继续深度学习方面的学习
3. 感觉深度学习已经变成了一门实验学科，怪玄学的，有些paper看的挺不明不白，参考网上的博客感觉也还是雾里看花，缺乏系统性的学习

## 4 工程结构<sup>2</sup>

1. report文件夹是与工程无关的，其中的log文件是训练时生成的loss文件，graph.ipynb是根据这些log文件生成可视化loss曲线的.jpg2lmdb.ipynb是转化数据集为lmdb格式时使用的。
2. eval\_tools文件夹是WIDERFace数据集评估PR曲线的matlab程序
3. 其余文件夹均与训练相关

<sup>2</sup>[https://github.com/guangyid/dlp\\_hw3](https://github.com/guangyid/dlp_hw3)

---

4. 我更改的大段代码会单独一行注释上 -hgy-

## 参考文献

- [1] Mingxing Tan, Ruoming Pang, and Quoc V. Le. Efficientdet: Scalable and efficient object detection. *CoRR*, abs/1911.09070, 2019.
- [2] Zhi Tian, Chunhua Shen, Hao Chen, and Tong He. FCOS: fully convolutional one-stage object detection. In *2019 IEEE/CVF International Conference on Computer Vision, ICCV 2019, Seoul, Korea (South), October 27 - November 2, 2019*, pages 9626–9635. IEEE, 2019.
- [3] Yanjia Zhu, Hongxiang Cai, Shuhan Zhang, Chenhao Wang, and Yichao Xiong. Tinaface: Strong but simple baseline for face detection. *CoRR*, abs/2011.13183, 2020.