

单阶段 vs 二阶段——人脸口罩的目标检测

郑家瀚 2019280627 医学院 生物医学工程 zheng-jh19@mails.tsinghua.edu.cn

姚非凡 2019312571 医学院 生物医学工程 yff19@mails.tsinghua.edu.cn

摘要 本文利用目标检测深度学习技术对人脸口罩进行识别检测，构建了 3 种不同的模型，分别是一阶段的 SSD 以及二阶段的 Faster RCNN 以及 Focal Loss，同时也针对结果最优秀的 Faster RCNN 做了一个电脑摄像头的实时检测。我们发现，各个模型都有它们倾向专注的地方，也各有自己的盲点，一阶段模型如 SSD 虽然很快，但存在一些结构性问题，导致它在 mAP 上无法突破；而 Focal Loss 的出现虽后于 Faster RCNN，但是在经过改进以后，其整体效能上更优于 Focal Loss 模型。三个模型都很好地完成了目标检测的基本功能，是能够在真实世界里派得上用场的目标检测模型。

Keywords—目标检测 SSD Focal Loss Faster RCNN

I. 简介

在 2020 年即将到来之时，新冠肺炎的病毒入侵了中国的武汉，随之席卷了全中国乃至全球。在抗击疫情的过程中，人脸口罩检测 (Face Mask Detection) 是其中一项必要的工作，通过对口罩的检测，可以加快后续的人脸识别操作，也是对医疗卫生、公共安全的保障。

口罩的识别任务主要依据的是目标检测 (Object Detection) 技术。目标检测关注是在图片中特定的物体目标，要求同时获得单个目标或多个目标的类别信息和位置信息。目标检测给出的是对图片前景和背景的理解，需要从背景中分离出感兴趣的目标，并确定这一目标的描述（类别和位置），因此检测模型的输出是一个列表，列表的每一项使用一个数据组给出检出目标的类别和位置（常用矩形检测框的坐标表示）。而我们则是需要在佩戴口罩的图片中识别出口罩是否存在；存在遮挡物的话，是否是口罩；存在的话，给出对应的位置。

A. 单阶段 (1-stage) 检测模型

单阶段模型没有中间的区域检出过程，直接从图片获得预测结果，也被成为 Region-free 方法。

1) YOLO

YOLO[1] 是单阶段方法的开山之作。它将检测任务表述成一个统一的、端到端的回归问题，并且以只处理一次图片同时得到位置和分类而得名。YOLO 将图片缩放，划分为等分的网格，每个网格按跟 Ground Truth 的 IoU 分配到所要预测的样本，其卷积网络由 GoogLeNet 更改而来，每个网格对每个类别预测一个条件概率值，并在网格基础上生成 B 个 box，每个 box

预测五个回归值，四个表征位置，第五个表征这个 box 含有物体（注意不是某一类物体）的概率和位置的准确程度（由 IoU 表示）。测试时，分数如下计算：

$$Pr(Class_i|Object) * Pr(Object) * IOU_{pred}^{truth} = Pr(Class_i) * IOU_{pred}^{truth}$$

等式左边第一项由网格预测，后两项由每个 box 预测，以条件概率的方式得到每个 box 含有不同类别物体的分数。因而，卷积网络共输出的预测值个数为 $S \times S \times (B \times 5 + C)$ ，其中 S 为网格数，B 为每个网格生成 box 个数，C 为类别数。在后处理上，YOLO 使用 NMS (Non-Maximum Suppression，非极大抑制) 过滤得到最后的预测框。损失函数被分为三部分：坐标误差、物体误差、类别误差。为了平衡类别不均衡和大小物体等带来的影响，损失函数中添加了权重并将长宽取根号。

YOLO 的主要优点是快；全局处理使得背景错误相对少，相比基于局部（区域）的方法，泛化性能好。

2) SSD: Single Shot Multibox Detector

SSD 算法 [2] 在传统的基础网络（比如 VGG）后添加了 5 个特征图尺寸依次减小的卷积层，对 5 个特征图的输入分别采用 2 个不同的 3×3 的卷积核进行卷积，一个输出分类用给的 confidence，每个 default box (default box，是指在 feature map 的每个小格 (cell) 上都有一系列固定大小的 box) 生成 21 个类别的 confidence；一个输出回归用的 localization，每个 default box 生成 4 个坐标值，最后将 5 个特征图上的结果合并 (Contact)，送入 loss 层。

SSD 在基础网络 (VGG) 后添加了辅助性的层进行多尺度卷积图的预测结果融合，提出了类似 Anchor 的

Default boxes，解决了输入图像目标大小尺寸不同的问题，同时提高了精度，可以理解为一种特征金字塔;SSD提出了一个彻底的 end to end 的训练网络，保证了精度的同时大幅度提高了检测速度，且对低分辨率的输入图像的效果很好。

B. 两阶段 (*2-stage*) 检测模型

两阶段模型因其对图片的两阶段处理得名，也称为基于区域 (Region-based) 的方法。

1) R-CNN

R-CNN[3] 将检测抽象为两个过程，一是基于图片提出若干可能包含物体的区域（即图片的局部裁剪，被称为 Region Proposal），文中使用的是 Selective Search 算法 CNN 对输入图像的大小有限制，所以在将候选区域输入 CNN 网络之前，要将候选区域进行固定尺寸的缩放，缩放分为两大类：各向同性缩放，长宽放缩相同的倍数与各向异性缩放，长宽放缩的倍数不同；二是在提出的这些区域上运行当时表现最好的分类网络 (AlexNet)，对 CNN 输出的特征用 SVM 进行打分，得到每个区域内物体的类别，针对每个类，通过计算 IoU 指标，采取非极大性抑制，以最高分的区域为基础，剔除掉那些重叠位置的区域，并将 CNN 对候选区域提取出的特征输入训练好的线形回归器中，得到更为精确的位置定位，实现时加入了 log/exp 变换来使损失保持在合理的量级上，可以看做一种标准化 (Normalization) 操作。

R-CNN 将检测任务转化为区域上的分类任务，是深度学习方法在检测任务上的试水。模型本身存在的问题也很多，如需要训练三个不同的模型 (proposal, classification, regression)、重复计算过多导致的性能问题等。

2) Fast R-CNN

Fast R-CNN[4] 指出 R-CNN 耗时的原因是 CNN 是在每一个 Proposal 上单独进行的，没有共享计算，便提出将基础网络在图片整体上运行完毕后，再传入 R-CNN 子网络，共享了大部分计算，故有 Fast 之名。

图片经过 feature extractor 得到 feature map，同时在原图上运行 Selective Search 算法并将 RoI (Region of Interest, 实为坐标组，可与 Region Proposal 混用) 映射到 feature map 上，再对每个 RoI 进行 RoI Pooling 操作便得到等长的 feature vector，将这些得到的 feature vector 进行正负样本的整理（保持一定的正负样本比

例），分 batch 传入并行的 R-CNN 子网络，同时进行分类和回归，并将两者的损失统一起来。Fast R-CNN 将 Proposal, Feature Extractor, Object Classification 和 Localization 统一在一个整体的结构中，并通过共享卷积计算提高特征利用效率。

Faster R-CNN 是 2-stage 方法的奠基性工作，提出的 RPN 网络取代 Selective Search 算法使得检测任务可以由神经网络端到端地完成。RPN 网络将 Proposal 这一任务建模为二分类 (是否为物体) 的问题。第一步是在一个滑动窗口上生成不同大小和长宽比例的 anchor box，取定 IoU 的阈值，按 Ground Truth 标定这些 anchor box 的正负。于是，传入 RPN 网络的样本数据被整理为 anchor box (坐标) 和每个 anchor box 是否有物体 (二分类标签)。RPN 网络将每个样本映射为一个概率值和四个坐标值，概率值反应这个 anchor box 有物体的概率，四个坐标值用于回归定义物体的位置。最后将二分类和坐标回归的损失统一起来，作为 RPN 网络的目标训练。由 RPN 得到 Region Proposal 在根据概率值筛选后经过类似的标记过程，被传入 R-CNN 子网络，进行多分类和坐标回归，同样用多任务损失将二者的损失联合。

3) Focal Loss

Focal Loss 的模型，Facebook 团队称之为 Retinanet，是由 Facebook 团队研发出来的，其中来自广州的作者——何恺明，便曾经参与了 Faster-RCNN 模型的开发，所以也不难猜到这两个模型是有其相似性的，而实际上 Focal Loss 模型便是基于 Faster-RCNN 改进开发出来的。

迄今为止，最高精度的目标检测模型是基于 R-CNN 普及的两阶段方法，它们同样是可以使用在高度稀疏的图像检测上的。相反的，如果应用在重复性高的采样，一阶段检测器可能会变得更快更简单，但到目前为止，它的精度已经落后于二阶段检测器。这是因为，在检测器密集的训练过程中，遇到了极端前景/背景类别的失衡问题。而 Retinanet 通过重塑标准交叉熵来解决此类不平衡问题，从而降低把权重分配给容易分类的示例的可能性。这是个在当时很富开创性的工作，它让模型能够把重点放在稀疏、困难的示例上，并防止在培训过程中，受到太多 easy negative 的影响。它的效能当时是超越了 Faster-RCNN 的算法模型的，而且运算速度可以跟一阶段检测模型比拟，其准确度也比它们要高。

C. 深度学习目标检测的发展

深度学习对目标检测的研究上不断发展，诞生了大量的工作。

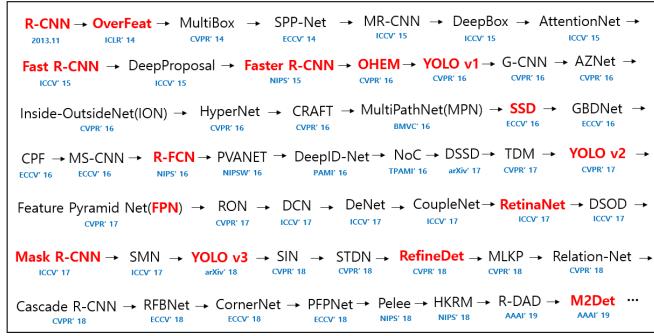


图 1. 2013 至 2019 深度学习目标检测发展

在这个基础上，一系列技术也得到了进步 [5]：

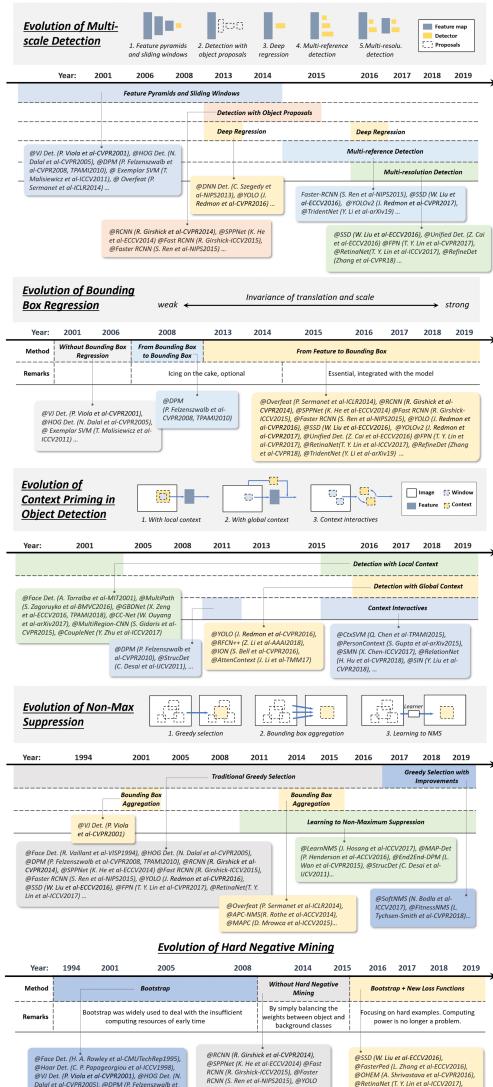


图 2. Evolution of techniques in object detection from 2001 to 2019

D. 本文贡献

本文利用目标检测深度学习技术对人脸口罩进行识别检测。本文构建了 3 种不同的模型，分别是一阶段的 SSD 以及二阶段的 Faster RCNN 以及 Focal Loss，来做口罩目标的检测。同时也针对结果最优秀的 Faster RCNN 做了一个电脑摄像头的实时检测。

II. 数据整理

我们对根据助教所提供的线索，即三大类(AIZOO、RMFD、Chan Chi Choi) 来源的数据，进行了整理分类，如下：

表 I
数据集整理

Dataset	Name	Syn	Mask	NoMask	Pos	label	P.format	L.format
AIZOO	Trainset	✗	3006	3114	✓	✓	.jpg	.xml
	Testset	✗	1059	780	✓	✓	.jpg .png	.xml
RMFD	RWMFD _part _1	✗	1000	0	✓	✗	.jpg	✗
	single2-0	✗	336	0	✓	✗	.jpg	✗
	single2-0 -1	✗	188	0	✓	✗	.jpg .webp .png .jpeg .jfif	✗
	self-built -masked -face -recognition -dataset	✗	2203	90000	✗	✗	.jpg	✗
CASIA -WebFace _masked		✓	500000	0	✗	✗	.jpg	✗
	Ifw _masked	✓		0	✗	✗	.jpg	✗
Chan Chi Choi	MALF	✗	0	5250	✓	✓	.jpg	.txt
	FDDB	✗	0	2845	✓	✓	.jpg	.txt
	Wider Face	✗	0	33203	✓	✗	.jpg	✗

AIZOO 的 FaceMaskDetection 数据集 (<https://github.com/AIZOOTech/FaceMaskDetection>) 开源了人脸口罩检测的主流框架的相应模型，并提供了相应的推理代码。该作者开源了如表格所示的 7,959 张人脸标注图片，数据集来自于 WIDER Face 和 MAFA 数据集，并重新修改了标注和校验。

Real-World Masked Face Dataset(RMFD) 数据集 (<https://github.com/X-zhangyang/Real-World-Masked-Face-Dataset>) 为武汉大学国家多媒体软件工程技术研发中心收集和标注的口罩数据集。包含从网络爬取样本，经过整理、清洗和标注后，含 525 人的 5 千张口罩人脸、9 万正常人脸。以及通过公开数据集中的人脸戴上口罩，得到 1 万人、50 万张人脸的模拟口罩人脸数据集。

Chan Chi Choi 是在 github 上的介绍的数据集，我们也一并下载进行了分析。首先我们先得搞清楚要训练的模型是需要什么样的数据，他们的标签需求又是如何的？简言之，在这个目标检测的问题中，神经网络要学习的大体有二，一是“在哪里”的问题，二是“这是什么”的问题，意即我们的数据集必须要能提供给模型一些位置检测的讯息以及对应此位置是否有戴口罩的讯息，以便让模型能够进行反馈训练。

综上所述，AIZOO 以及 FDDB 都比较适合我们来使用，毕竟他们的标签讯息比较完善，而且数据集的设计便是用来解决“在哪里”以及“这是什么”的问题。其他数据集像是 RMFD 的有的照片只能用来解决“这是什么”的问题，而缺乏了“在哪里”的问题。为了保证数据不平衡问题不会发生，我们便采用了有戴口罩、没戴口罩数量比较平均的 AIZOO 数据集。

我们使用的 AIZOO 数据集，可以发现该数据集有两类的数据，分别为口罩和非口罩人像，且数据相对平衡，数据集包含对每张照片的注释，注释信息包含图片的类别、目标的位置，该数据集适合作为训练与测试，该数据分布如下：

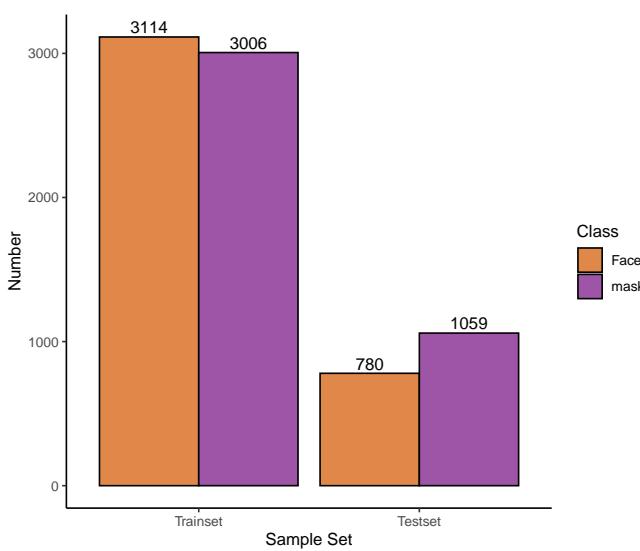


图 3. AIZOO 数据集分布可视化

在把数据应用在模型训练之前，我们务必要搞清楚要使用的标签类别与格式。主要的标签类别有 Bounding Boxes、Polygonal Segmentation、Semantic Segmentation 以及 3D cuboids，在这一模型我们想当然用的就是用得最广泛的 Bounding boxes 的格式，而刚好 AIZOO 的数据类型也是属于这一类。在数据标签格式方面的话，AIZOO 是属于 Pascal VOC 的标签格

式，而我们在 SSD 使用便是这一类型；在 Faster RCNN 以及 Focal Loss 则是采用了 COCO 的标签格式。之所以会采用两种不同的数据格式，这是因为考虑到原模型使用了这样的标签格式，为了避免成果相差太大，我们采用了符合原模型的数据类型。

在数据整理的主要工作，便是要把 Pascal VOC 转换到 COCO。Pascal VOC 格式是一个照片对应一个同名的 xml 文件，而 COCO 则是所有照片对应一个 json 文件。在转换过程中也有发现一些错字，比如 face_mask 打成了 face_nask。在格式转换过程中，最大的难点在于原文件的数据格式并不一致，有者少了 pascal voc 的 path 的数据，有者则是没有图像大小的数据，而且数字序号并不统一，这些都是必须要解决的。毕竟 COCO 格式是在假定名称后面的数字序号是不重复的，因此可以直接通过序号来搜索资料。在转换过程中，统一把 path 的资料只留下图像的名称，序号也重新计算，格式转换是通过 python 完成，也一并附在参考代码里了。

III. 模型设计

A. SSD

模型使用了 SSD 类型的架构，本模型输入大小为 260x260，主干网络只有 8 个卷积层，加上定位和分类层，一共只有 24 层（每层的通道数目基本都是 32、64、128），模型只有 101.5 万参数。八个卷积层是主干网络，也就是特征提取层，20 层是定位和分类层。训练目标检测模型，最重要的合理的设置 anchor 的大小和宽高比，笔通过统计数据集的目标物体的宽高比和大小来设置 anchor 的大小和宽高比，因为人脸的一般是长方形的，而很多图片是比较宽的，人脸的宽度和高度归一化后，有很多图片的高度是宽度的 2 倍甚至更大。从上图也可以看出，归一化后的人脸高宽比集中在 12.5 之间。根据数据的分布，我们将五个定位层的 anchor 的宽高比统一设置为 1, 0.62, 0.42。（转换为高宽比，也就是约 1, 1.6:1, 2.4:1）。

为了避免使用手挡住嘴巴就会欺骗部分口罩检测系统的情况，在数据集中加入了部分嘴巴被手捂住的数据，另外在训练的过程中，随机的往嘴巴部分粘贴一些其他物体的图片，从而避免模型认为只要露出嘴巴的就是没戴口罩，没露出嘴巴的就是带口罩这个问题，通过这两个规避方法，解决了非口罩遮挡物被当作口罩的误判。后处理部分主要就是非最大抑制（NMS），我们使

用了单类的 NMS，也就是戴口罩人脸和不戴口罩人脸两个类别一起做 NMS，从而提高速度。

迭代下模型 Loss 如下：

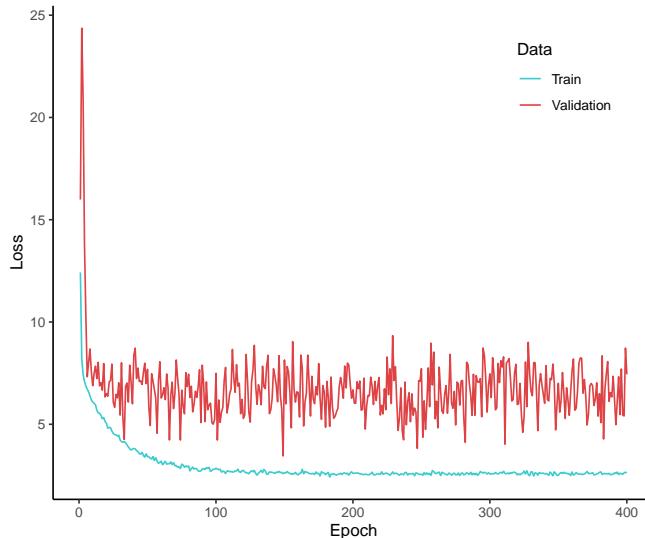


图 4. Total Loss of SSD while training

B. Faster RCNN

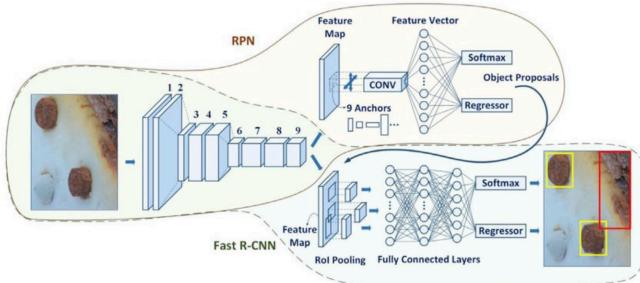


图 5. Faster RCNN 的基本示意图

在前面有提过这个是二阶段的目标检测模型，前一阶段是 RPN 也就是寻找位置的模型，它会提出一些可能的格子提供给下一阶段的模型进行分类判断，此处是全连接。从前一阶段到后一阶段是循序渐进，前后可谓泾渭分明的。这是一种 Single Feature Map 的方法。

在经过调研后发现，比较多人开始使用 Feature Pyramid Network 的方法，这是后期 Facebook 团队在开发 Retinanet 时使用的方法。这是个很富开创性的工作，它不仅使得训练变得更加快速，也使得模型准确率更高。这是因为它让这二阶段的方法，在第一阶段进行特征提取的时候，也往第二阶段的 predict 层做连接，这也变相使得，二阶段的方法不再那么泾渭分明了。

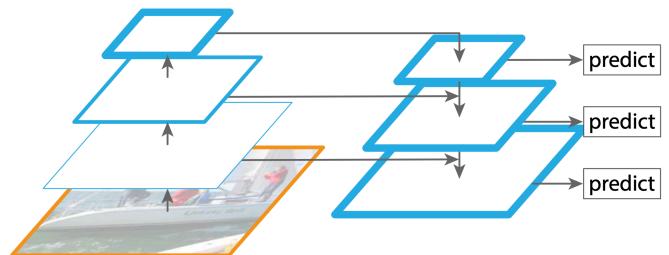


图 6. Feature Pyramid Network

我们在此次模型设计上便是采用了这样的 FPN 模型——Faster RCNN R50 FPN 3x，意即在 RPN 阶段，只会提供 50 个 proposals，这将使得此模型的运算速度快三倍。

在训练的过程中，我们依照 TotalLoss 以及正确率来判断模型的训练是否已经足够收敛：

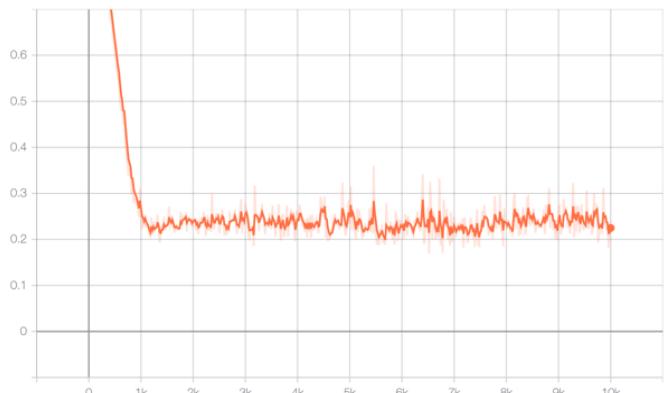


图 7. Total Loss of Faster-RCNN while training

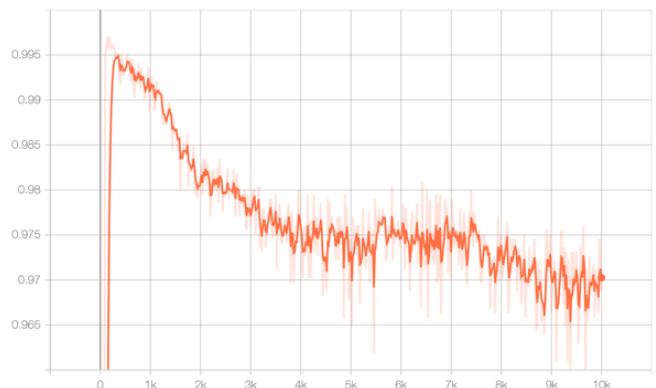


图 8. Class Accuracy of Faster-RCNN while training

从上图我们可以观察到，在训练了 10000 次迭代以后，TotalLoss 已经不再下降，分类准确率也已经渐趋平衡，因此我们可以判断说 10000 次的迭代就已经很足够。

经过一些测试，我们判断当 ROI_HEADS 的 SCORE_THRESH_TEST 的阈值设定为 0.7 的时候，效果就足够好，连用花盖着脸的女孩的图像都能很好地检测到：



图 9. An Example

C. Focal Loss

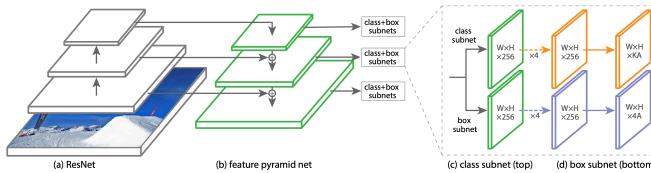


图 10. The One-stage RetinaNet Network Architecture

Focal Loss 的方法跟前面的 Faster RCNN 是很像的，也是第一次比起 Faster RCNN 首先使用 FPN 方法的模型。我们在这里使用的是 Retinanet R 50 FPN 1x，同样的我们认为，对于我们这一次作业的要求，50 个 Proposals 就已经非常足够。毕竟照片的人脸数量比较常见的都是一两个，最多也就十几个，因此把模型训练好的话，我们就可以很好的达标了。

表 II
不同参数下的 AP

γ	α	AP	AP_{50}	AP_{75}
0	0.75	31.1	49.4	30.0
0.1	0.75	31.4	49.9	30.1
0.2	0.75	31.9	50.7	33.4
0.5	0.5	32.9	51.7	35.2
1	0.25	33.7	52.0	36.2
2	0.25	34.0	52.5	36.5
5	0.25	32.2	49.6	34.8

我们模型的参数是使用了论文里最优秀的参数，也就是 gamma 为 2.0, alpha 为 0.25。FocalLoss 的损失函数曲线会比较特别，毕竟它有对交叉熵作了次方的乘积，当它遇到比较难分类的问题时，交叉熵就会比 easynegative 会重得多，变相让 BP 的权重分配得到更多的回馈。下图显示损失函数的曲线：

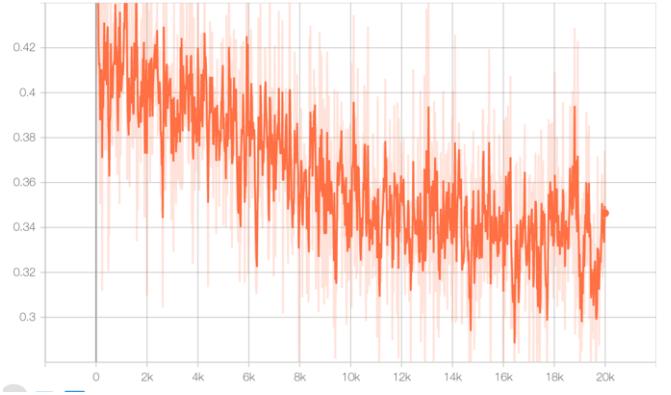


图 11. Total Loss of Focal Loss

可以看到在 20000 的训练回圈里，尽管熵函数是很不稳定，急剧上上下下的，但总得来说还是有其下降趋势的，因此我们可以借此判断是否已经足够收敛。但这个收敛的趋势比起前面的 Faster RCNN 来得慢得多的，因此我们也唯有让它作更多次训练的迭代，这里总共花了我们三个半小时，才完成了 20000 次迭代的次数，因此我们可以说，要训练这一模型是挺不容易的。经过一些测试以后，我们判断 SCORE_THRESH_TEST 使用 0.5 就效果最好。

IV. 实验设计及结果

数据集将训练集分为训练样本和验证样本，并对测试集中的样本进行测试。

AP(Average Precision) 是指平均精准度，是对 PR 曲线上的 Precision 值求均值。对于 pr 曲线来说，我们使用积分来进行计算。在实际应用中，我们并不直接对该 PR 曲线进行计算，而是对 PR 曲线进行平滑处理。即对 PR 曲线上的每个点，Precision 的值取该点右侧最大的 Precision 的值。我们使用 PASCAL VOC 风格的 11 点插值法进行计算 AP。Pascal VOC 2008 中设置 IoU 的阈值为 0.5，如果一个目标被重复检测，则置信度最高的为正样本，另一个为负样本。在平滑处理的 PR 曲线上，取横轴 0-1 的 10 等分点（包括断点共 11 个点）的 Precision 的值，计算其平均值为最终 AP 的值。

下表中展示各个模型两个类别的在不同 IoU 取值下的平均 AP，即 mAP@.5, mAP@.7, mAP@.9, mAP@[.5:.95]。

表 III
不同模型下的 MAP

Model	Class	mAP @.5	mAP @.7	mAP @.9	mAP @[.5 : .95]
SSD	Mask	0.86	0.78	0.22	0.61
	Face	0.81	0.78	0.29	0.63
Focal Loss	Mask	0.90	0.78	0.14	0.59
	Face	0.87	0.84	0.34	0.69
Faster RCNN	Mask	0.91	0.82	0.25	0.65
	Face	0.91	0.89	0.47	0.75

Precision 其实就是在识别出来的图片中，True positives 所占的比率：

$$precision = \frac{tp}{tp + fp} = \frac{tp}{n}$$

，其中的 n 代表的是 (True positives + False positives)，也就是系统一共识别出来多少照片。Recall 是被正确识别出来的飞机个数与测试集中所有飞机的个数的比值：

$$recall = \frac{tp}{tp + fn}$$

，通常 tp+fn 在目标检测中指 groundTruth 中的真实目标数量。Precision 和 Recall 最早是信息检索中的概念，用来评价一个信息检索系统的优劣。而在目标检测领域，假设我们有一组图片，里面有若干待检测的目标，Precision 就代表我们模型检测出来的目标有多打比例是真正的目标物体，Recall 就代表所有真实的目标有多大比例被我们的模型检测出来了。

我们当然希望检测的结果 P 越高越好，R 也越高越好，但事实上这两者在某些情况下是矛盾的。比如极端情况下，我们只检测出了一个结果，且是准确的，那么 Precision 就是 100%，但是 Recall 就很低；而如果我们把所有结果都返回，那么必然 Recall 必然很大，但是 Precision 很低。

因此在不同的场合中需要自己判断希望 P 比较高还是 R 比较高。如果是做实验研究，可以绘制 Precision-Recall 曲线来帮助分析。以 Recall 值为横轴，Precision 值为纵轴，我们就可以得到 PR 曲线。我们会发现，Precision 与 Recall 的值呈现负相关，在局部区域会上下波动。

SSD 模型下 IoU 阈值取 0.5, 0.7, 0.9 时每类的 Precision-Recall 曲线如下：

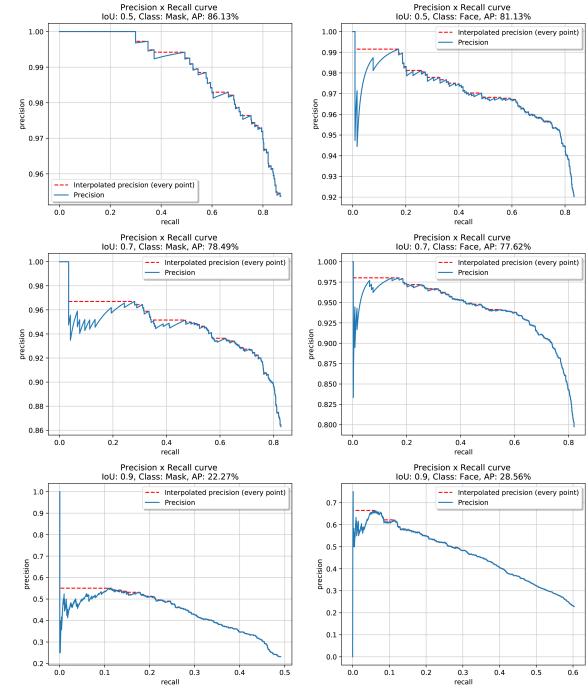


图 12. SSD 模型的 Precision-Recall 曲线（左侧从上到下分别是口罩目标检测在 IoU=0.5、0.7、0.9 时的 Precision-Recall 曲线；右侧为脸目标检测。）

Focal Loss 模型下 IoU 阈值取 0.5, 0.7, 0.9 时每类的 Precision-Recall 曲线如下：

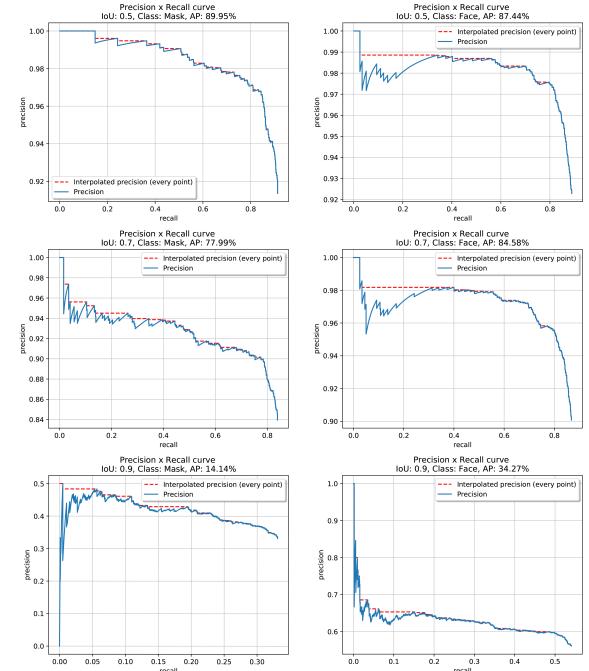


图 13. Focal Loss 模型的 Precision-Recall 曲线（左侧从上到下分别是口罩目标检测在 IoU=0.5、0.7、0.9 时的 Precision-Recall 曲线；右侧为脸目标检测。）

Faster RCNN 模型下 IoU 阈值取 0.5, 0.7, 0.9 时每类的 Precision-Recall 曲线如下：

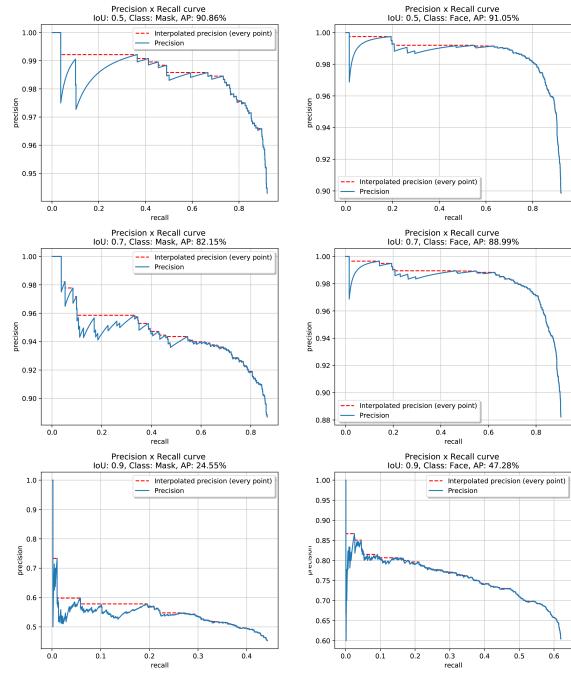


图 14. Faster RCNN 模型的 Precision-Recall 曲线（左侧从上到下分别是口罩目标检测在 IoU=0.5、0.7、0.9 时的 Precision-Recall 曲线；右侧为脸目标检测。）

不同模型的检测时间不同，对测试集的所有图片进行测试，再进行平均可以发现 SSD 模型的检测时间最短，其次是 Focal Loss，最长的是 Faster RCNN：

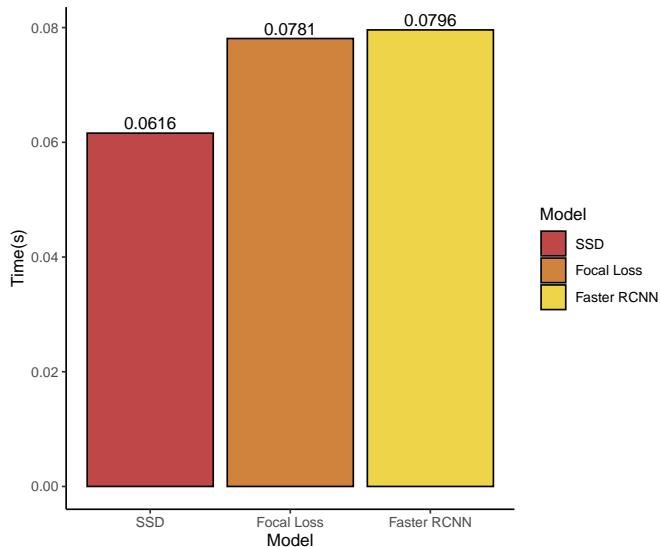


图 15. 三种模型检测的时间

V. 实验结果分析

综合前面的结果，要回答哪一个是最好的模型这一问题的话，我们就要看看模型的平均运算速度，以及它们的各种准确度。我们可以看到 SSD 是目标检测最快的模型，这正符合了我们对单阶段模型的预测，属于二阶段的 Faster RCNN 则拥有最高的整体准确度，而同属二阶段的 Focal Loss，却只是在平均准确度略胜于 SSD。因此最快的是 SSD，而最准确的则是 Faster RCNN，在回答“哪一个是最好的模型”这一问题以前，我们先来深入分析一下前面的结果。

首先，我们来看看各个模型、各个判别、各个 IoU 阈值的 Precision-Recall curve，可以看到 SSD 比较其他的模型，在 Mask 的检测上，在开头（左边）的时候都有比较尖锐的下坠，这其实可以说明一件事，就是 SSD 在最有信心的时候会犯错，而这想当然的不是一个好现象。我们知道，在作 PR curve 的时候，会先把判别的 confidence——信心，进行排序，最有信心的结果会先在 PR curve 的左边出现，从左到右画完，所以在最左方的就是最有信心的，而 SSD 在最有信心的部分却屡屡犯错，导致前面有尖锐的下坠。我们找到如下面的这个例子，SSD 很有信心地把用花盖脸的女孩判断成了戴口罩的女孩，而且信心度达 0.96，如下：



图 16. Another Example

模型在最有自信的时候应当是最准确的，而在 SSD 却成了最容易犯错的时候，这说明了 SSD 在学习过程中，有了错误的学习方向，而这可能与过学习有关。这很难说是学习时间不够久的原因，毕竟在最后损失函数也渐趋平衡，也已经所剩无几了。我们估计是模型对于极端数据较不敏感，再来是极端数据本来就占少数，在

SSD 的情况，可能需要更多的 Hard Negative 的极端数据才能有更好的训练结果。

尽管 SSD 会有这样的问题，但是它在人脸的识别上却是非常杰出的，在 $\text{IoU}=0.5$ 的时候，甚至是比其他两个还要好的，这说明了两个可能性，一是这一模型在掌握人脸的学习上是比较全面的，二则是这一模型的 classifier 更倾向于把物体判成人脸。综合在 Mask 的表现，我们推测二的可能性会更大。在 $\text{IoU}=0.5$ 的时候，整个准确率都倾向了人脸的部分，因此我们可以判断 SSD 的 classifier 的学习不比其他两个模型优秀，它是更加偏重了人脸的判别的。

延续我们前面的问题，关于“有信心的时候会犯错”的问题上，Focal Loss 以及 Faster RCNN 则是比较克制，它们不相伯仲，都很优秀。这因此可能是二阶段模型的一个特点，它把“物体在哪里”和“物体是什么”，两个问题前后分开，有两个阶段的损失函数来对他们进行修正，因此比较一阶段模型，会比较少出现这种“满招损，谦受益”的现象。

这里在说尽了 SSD 的坏话以后，要说点 SSD 的好话。在 IoU 阈值高达 0.9 的时候，二阶段的两个模型都表现了失衡的状态，Focal Loss 更倾向把数据判断成口罩，而 Faster RCNN 则更倾向把数据判断成无口罩，唯独 SSD 是比较平均的，这也说明 SSD 在“在哪里”的问题是比其他的两个模型学习得比较全面的。

Faster RCNN 在 $\text{IoU}=0.9$ 的严峻情况下，虽然有点不平衡，但是它的整体准确率是远高于其他两个模型的，这也说明了 Faster RCNN 在严酷的要求下，它的效能是更优于其他两个模型的。它在学习“在哪里”的问题上因此是比其他两个模型来得更优秀的，这不得不说这可真是 Facebook 团队的杰出之作，他们本来是用各方面性能都比较优秀的 Focal Loss 的 RetinaNet 打败了 Faster RCNN，却在随着研发的推进，把 Faster RCNN 改良成比原初的 RetinaNet 来得更加优秀。

此外，依照 Focal Loss 的论文，说是损失函数的 BP 会更多地着重在改善 Hard Negative 的分类，那具体我们可以从哪儿看到这个方向的改进呢？那我们可以看看在 $\text{IoU}=0.5$ 的时候，PR 图的右边方向的部分是更加饱满的，是不会掉得那么快的，这就说明了在模型自信比较低的时候，正确率还是有所保障的，上面当然说的是对比 SSD 的，SSD 在自信比较低的时候，准确率都掉得比较快。

综上所述，我们可以清楚地说明一个现象，就是二

阶段在整体准确率上是更胜于一阶段模型的，主要原因可能来自于模型结构性的限制。一阶段模型比起二阶段模型有的是更简洁明了的架构，而这样的模型当它面对真实世界的复杂性，可能太过于简化，而导致学习会面临到一些缺陷，这个缺陷在二阶段模型都可以比较好的解决。这里开头就问的一个问题：“哪个模型是最好的模型”，我想现在就可以回答了，Faster RCNN 是实至名归的，除了因为它在整体准确率上是更优于其他模型，它也可以更完善、更稳定地解决一些真实世界比较困难的问题。

VI. 小组成员贡献

郑家瀚：Focal Loss 模型、Faster RCNN 模型、撰写报告、数据分析、实时监测系统

姚非凡：SSD 模型、撰写报告、各模型的 mAP 运算、数据分析

VII. 参考代码

见附件中。

VIII. 结论

一阶段模型如 SSD 虽然很快，但存在一些结构性问题，导致它在 mAP 上无法突破。此外，虽然 Focal Loss 的出现后于 Faster RCNN，但是在经过改进以后，整体效能上更优于 Focal Loss 模型。我们也发现，各个模型都有它们倾向专注之处，也各有自己的盲点，其实三个模型都很好地完成了目标检测的基本功能，是能够在真实世界里派得上用场的目标检测模型。

经过这次的实验，我们也成了计算机视觉的半个内行人，在自动驾驶的战场上，我们看到有以 google 为首的 Waymo 公司与马斯克的特斯拉公司的战疫，前者是依靠多个传感器，比如测距、GPS、红外线等等回馈系统以求达到实时的自动驾驶，而后者则是为了兼顾美观以及科技感，摒除了所有这些传感器，只使用了 GPS 以及视觉来作实时的回馈，来做到自动驾驶。在经过这次的深入实验以后，我决定站在马斯克这一方，计算机视觉也许还有很长的路要走，但是绝对可行。

参考文献

- [1] Redmon, J. , Divvala, S. , Girshick, R. , & Farhadi, A. . (2015). You only look once: unified, real-time object detection.
- [2] Liu, W. , Anguelov, D. , Erhan, D. , Szegedy, C. , Reed, S. , & Fu, C. Y. , et al. (2016). Ssd: single shot multibox detector.
- [3] Girshick, R. , Donahue, J. , Darrell, T. , & Malik, J. . (2013). Rich feature hierarchies for accurate object detection and semantic segmentation.
- [4] Ren, S. , He, K. , Girshick, R. , & Sun, J. . (2015). Faster r-cnn: towards real-time object detection with region proposal networks. IEEE Transactions on Pattern Analysis and Machine Intelligence.
- [5] Zou, Z. , Shi, Z. , Guo, Y. , & Ye, J. . (2019). Object detection in 20 years: a survey.