# 人脸口罩检测

张斐然

**摘要**

人脸口罩检测，作为目标检测的一个应用，在当下疫情的防控环节具有重要作用。深度学习的快速发展使得目标检测的方法不断能够突破现有准确率。本项目基于现有的主流模型，实现了较高准确率，且实时速度的口罩检测，同时分析了不同模型的特点，对比了不同数据处理和训练参数对检测准确率的影响。

**关键词**

目标检测；人脸检测；口罩检测；深度学习；卷积神经网络

## 一、简介

目标检测，作为计算机视觉中最基本的任务之一，其目的是得到某些物体（如人、动物、交通工具等）出现在图像中的位置信息，主要应用场景包括自动驾驶、人脸识别、机器人智能、视频监控等。目标检测与其他计算机视觉任务关联紧密，包括图像分类、目标分割、图像标注等。本项目的任务，则是在当前疫情背景下，对人们是否佩戴口罩进行检测。更具体地，本文中的任务是一个两类检测问题，即检测戴口罩的人脸和不戴口罩的人脸；要求能够输入一张图像，输出若干个五元组：检测到的每个人脸的位置坐标，以及其是否佩戴口罩。

目标检测的方法大部分属于机器学习的范畴，即首先通过一系列标注好的图像（已知各个物体坐标以及类别信息）来训练模型，然后用训练好的模型作为检测器实现目标检测任务。在深度学习兴起前的二十多年中，由于算力的不足以及缺乏图像特征的有效表示，目标检测任务的准确率始终不高。而在几年前，在基于卷积神经网络（CNN）的深度学习方法发展起来后，目标检测准确率飞速提高，各种方法如雨后春笋般出现，使得目标检测成为热门领域。

本项目利用 7000 多张标注好的人脸图片，通过 PyTorch 实现了三种主流的深度学习目标检测模型：Faster RCNN[1]、SSD[2]，和 RetinaNet[3]来完成人脸口罩的检测任务，以及一种基于 RetinaNet 的针对遮挡人脸的进一步改进模型 Facial Attention Network[4]，在测试集上均达到了较高的准确率；同时分析了不同模型的特点，对比了不同参数对检测准确率的影响。

## 二、数据处理

本项目使用了 AIZOO 开源的人脸口罩数据集，共包含来自于 Wider Face 和 MAFA 数据集（各占一半）的 7959 张人脸图像，其中前者主要包含各种场景下的无口罩人脸图像；而后者则基本上都是真实口罩人脸图像。AIZOO 作者在其基础上进一步校验修改了数据标注（如将 MAFA 中脸部非口罩遮挡的图像重新标注为无口罩）。本项目则在使用该数据集的过程中，删去了其中两张无脸部标注的图像，并在输入模型时将部分灰度图像，以及 RGBA 通道的图像统一转变为 RGB 通道的图像。

## 1、 数据分布可视化

首先统计来自于两个数据集中图像的人脸个数如下。可见 Wider Face 的平均人脸更多，有些甚至多达几十上百个；相比 MAFA 中则相对较少。

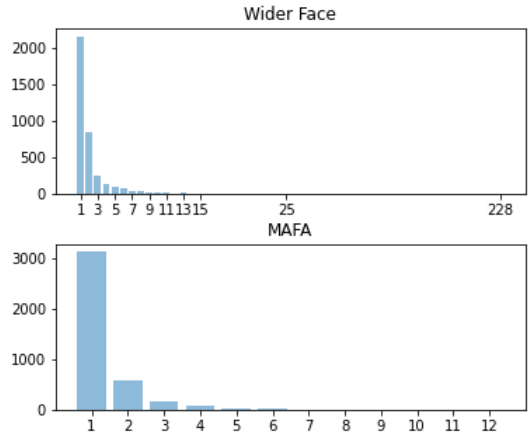


图 1 图像中人脸个数分布

其次，统计每张图像的尺寸，发现 Wider Face 中的图像宽度被归一到了 1024，而 MAFA 中图像尺寸则相对更小。

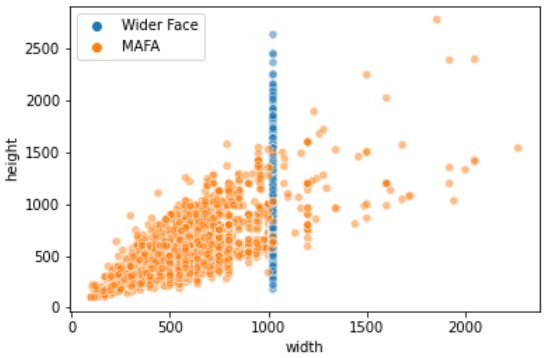


图 2 图像尺寸分布

最后，统计人脸的高宽比，发现两个数据集中均主要分布在 1~1.5 之间。

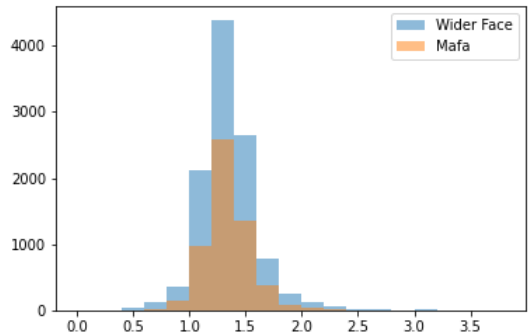

图 3 人脸高宽比分布

# 三、模型设计

总体上，基于深度学习目标检测模型大致可分为两步（two-stage）检测模型和一步（one-stage）检测模型，其中前者首先得到检测框的候选（proposal），再进一步得到更精细的候选框；而后者则只用一个网络直接得到检测框。因此，two-stage 的方法相对准确率更高，但速度更慢；one-stage 方法速度相对更快，但准确率偏低，尤其是对小物体的检测。

而从实现角度，目前最新的深度学习目标检测模型由三个部分构成：躯干（backbone）、脖子（neck），和头部（head）。其中 backbone 负责从原始图像中提取特征，主要来自于主流的 CNN 模型网络的卷积层部分，同时一般会通过在图像分类数据集上预训练得到权重来实现迁移学习，以此来加快模型收敛速度。Neck 特指 2017 年提出的 Feature Pyramid Network（FPN[5]），作用是在 backbone 中的不同层中加入自顶向下的侧向连接到 head 中，而非只在 backbone 的最后一层。FPN 的思想是 backbone 中更深层提取到的图像特征会更全局，不利于小物体的识别；而多个中间层中的结果直接连入 head 中则有利于识别多个有着不同尺度的物体。而 head 则是不同目标检测模型的主要区别所在，一般来说，one-stage 方法只有一个 head 用于输出检测坐标和类别；而 two-stage 方法则有两个 head 来得到更精细的结果。

本项目共实现四种模型，其中第一个 Faster RCNN（带 FPN）属于 two-stage 方法，而后的 SSD 和 RetinaNet（带 FPN）及其改进 FAN 则属于 one-stage 方法。各模型的具体介绍如下。

**1、 Faster RCNN**

该模型于 2015 年在 RCNN 和 Fast RCNN 的基础上提出，其中 2014 年的 RCNN 为最早的基于深度卷积神经网络的检测模型。Faster RCNN 的主要贡献是提出了 Region Proposal Network（RPN），来以很低的成本生成检测框的候选，同时将之前两个模型中的各个相对的独立部分均整合到了一个网络里，包括候选检测、特征提取、检测框回归，以及分类等，实现了端到端的一步训练，同时检测速度也达到了几乎实时的。

**2、 Single Shot MultiBox Detector (SSD)**

SSD 同样在 2015 年被提出，是继 YOLO 后的第二个深度学习时代的 one-stage 的模型。SSD 是第一个提出在多个尺度上（也即之后 FPN 的思路），和多个长宽比上（即多个预设的“anchor”）同时进行物体检测的模型，因而显著提升了 one-stage 方法的精度，尤其是对小物体。SSD 方法的诸多改进版，如 DSSD、RBFNet 等，都在更好的解决多尺度的特征的融合问题，来实现不同尺度下物体检测的更高准确率。

**3、 RetinaNet**

RetinaNet 于 2017 年提出，试图进一步解决 one-stage 方法准确度较低的问题。其主要思路是解决模型在训练时前景（也即包含物体的检测框）和背景两个类的严重不均衡问题，方法则是提出了新的损失函数：focal loss 来代替原本的交叉熵损失函数，可以让模型在训练时“聚焦”在更难的错误分类样本，从而让被错分的检测框得到更充分的训练，提高检测的准确率。RetinaNet 使得 one-stage 方法达到了与 two-stage 方法相当的准确率，同时有着非常快的检测速度。

### 4、 Facial Attention Network (FAN)

FAN 在 2017 年基于 RetinaNet 模型，提出了更适合人脸检测的 anchor 参数；并进一步加入了在不同层的 anchor-level attention 的机制，使模型在人脸被部分遮挡时更注意没被遮挡的部分，来提高模型在不同大小下的部分遮挡的人脸上的准确率。FAN 在 Wider Face 和 MAFA 数据集下的困难样本上得到了较大的准确率提升。

## 四、实验设计及结果

### 1、 实验设计

数据集划分：数据集沿用了 AIZOO 中的划分，并将原本验证集中的 1839 张随机（numpy 的 seed 设置为 0）选择出 839 张作为验证集，剩余 1000 张作为测试集。

数据增广：在训练样本中，每张图像首先会以 0.5 的概率水平翻转；其次图像亮度、对比度、饱和度各会以 0.5 的概率和随机的顺序进行调整。最后图像被重新调整为适合各个模型的大小，并归一化为 ImageNet 中图像的均值和方差。

训练参数：经过简单调参后，使用如下参数训练：SSD 的 batch 大小为 8，其余三种模型 batch 大小为 2；Faster RCNN 和 SSD 使用 SGD 作为优化器，学习率分别为 5e-3 和 2e-3，momentum 为 0.9，权重衰减为 5e-3；RetinaNet 和 FAN 使用 Adam 作为优化器，学习率为 1e-4，其余参数为默认。SSD 的 epoch 次数为 12，其中学习率在第 8 和 11 轮结束后减为 1/10；其余 epoch 次数为 7，在第 3 和第 6 轮结束后减为原本的 1/10。

### 2、 实验结果汇总

表 1 每种（验证集上）最好的模型（测试集上）的结果

| | | Faster RCNN | SSD | RetinaNet | FAN |
|---|---|---|---|---|---|
| 无口罩 | AP@.5 | **0.8995** | 0.7992 | 0.8092 | 0.8792 |
| | AP@.7 | 0.7908 | 0.6683 | **0.7927** | 0.7787 |
| | AP@.9 | 0.0694 | 0.1151 | **0.1350** | 0.0940 |
| | AP@[.5:.95] | **0.5977** | 0.5101 | 0.5789 | 0.5860 |
| 有口罩 | AP@.5 | **0.9036** | 0.9024 | 0.9013 | 0.9008 |
| | AP@.7 | **0.7946** | 0.7867 | 0.7888 | 0.7822 |
| | AP@.9 | 0.1077 | 0.1313 | 0.1309 | **0.1619** |
| | AP@[.5:.95] | **0.6274** | 0.5983 | 0.6179 | 0.6166 |
| mAP | mAP@.5 | **0.9015** | 0.8508 | 0.8552 | 0.8900 |
| | mAP@.7 | **0.7927** | 0.7275 | 0.7908 | 0.7804 |

| | | | | | |
|---|---|---|---|---|---|
| | mAP@.9 | 0.0885 | 0.1232 | **0.1329** | 0.1279 |
| | mAP@[.5:.95] | **0.6126** | 0.5542 | 0.5984 | 0.6013 |
| Detection FPS | | 24.85 | **62.18** | 26.37 | 24.62 |

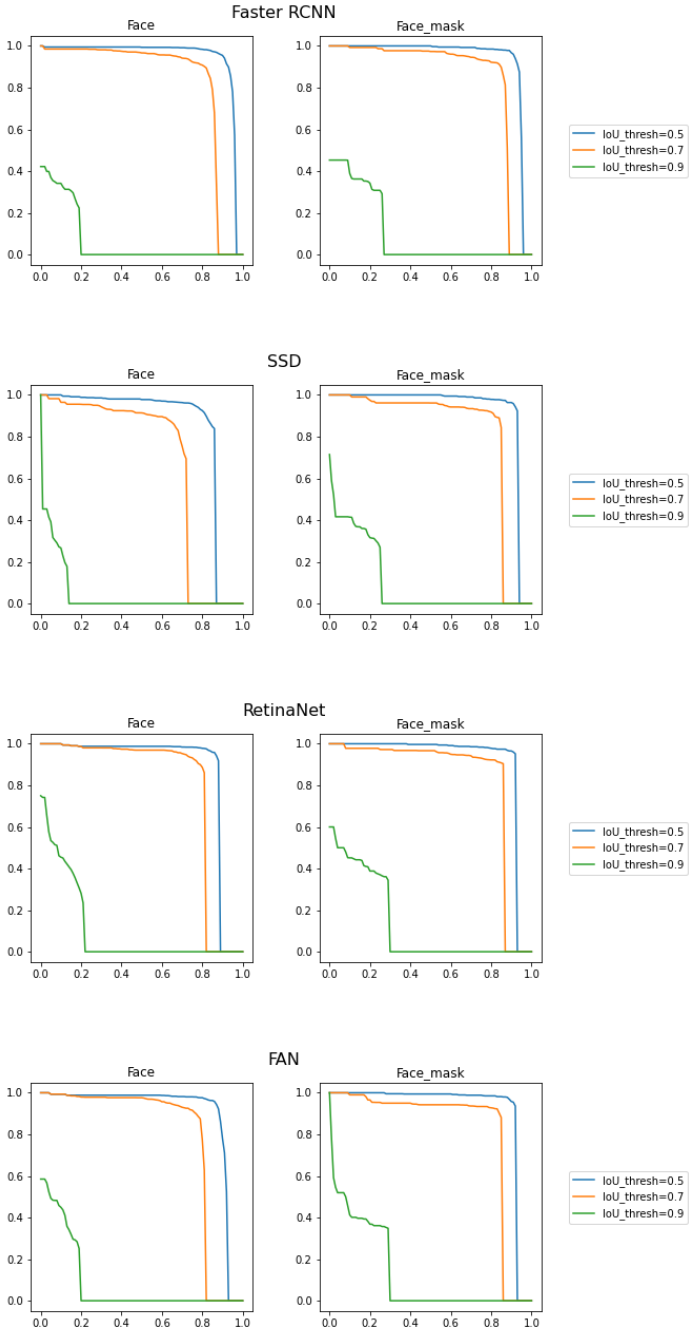


图 4 Precision-Recall 曲线

### 3、 基于不同预训练对 Faster RCNN 结果的影响

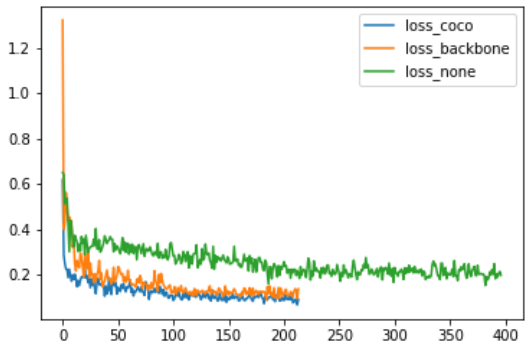


图 5 不同预训练下的 loss 曲线

表 2 不同预训练下 Faster RCNN（验证集上）的 mAP@[.5:.95]

| | COCO | Backbone (ImageNet) | None |
|---|---|---|---|
| mAP@[.5:.95] | 0.6157 | 0.5896 | 0.4681 |

### 4、 数据增广对 SSD 结果的影响

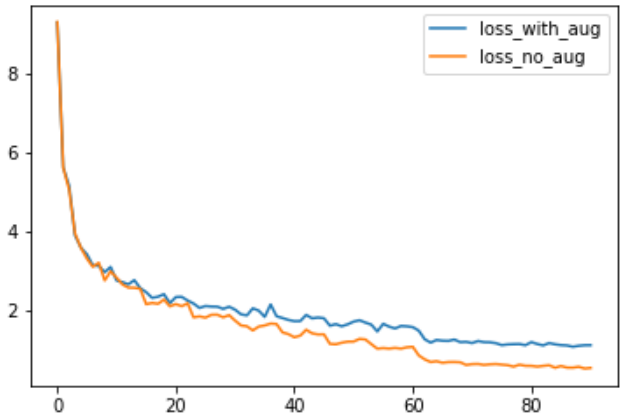


图 6 是否数据增广的 loss 曲线

表 3 是否数据增广 SSD（验证集上）的 mAP@[.5:.95]

| | 有数据增广 | 无数据增广 |
|---|---|---|
| mAP@[.5:.95] | 0.5614 | 0.5360 |

# 五、实验结果分析

## 1、 调参结果分析

Batch 大小：考虑到 batch 大小影响显存，除了 SSD 需要使用 300*300 的输入，可将 batch 大小设置为 8 之外，其余三个模型都无需限制输入图像尺寸，故设置为 512 或 600 后，较大的 batch 会超显存；故统一设置为 2。

图像预处理：考虑到图像长宽比最多不超过 1/2 或 2，故将输入图像 resize 到长宽一致以最大化利用信息，而非等比伸缩再用空白填充到长宽一致。因此，在设置 anchor 的 aspect ratio 时，就无法利用数据集中 box 的长宽比分布来确定，所以统一使用默认的[0.5, 1, 2]来作为 aspect ratio。

训练参数的选择：实验发现 Adam 在 RetinaNet 上收敛会更快一些，而对 Faster RCNN 和 SSD 无较大影响。此外加大 Epoch 会对模型结果有一定的提升，但并不明显，故维持不变。

## 2、 最好模型结果分析

首先，从每种模型最好的结果来看，四种模型相差并不大；Faster RCNN 相对更优一些，而 SSD 相对稍差一些。这与数据集本身相对简单有关，。其次，对比两类的结果，发现无口罩的人脸（基本来源于 Wider Face 数据集）普遍比带口罩人脸（基本全部来源于 MAFA 数据集）结果更差一些。这与两个数据集的难度也有关，直观来看 Wider Face 中的人脸图像来源于各种场景，人脸组成更加复杂一些；而 MAFA 中的则有很多的“大头照”，相对更容易检测出。此外如果可视化人脸的大小相对于整张图片的大小时（如下图），会发现 Wider Face 中的脸比 MAFA 中小很多，这也造成前者的检测更困难。

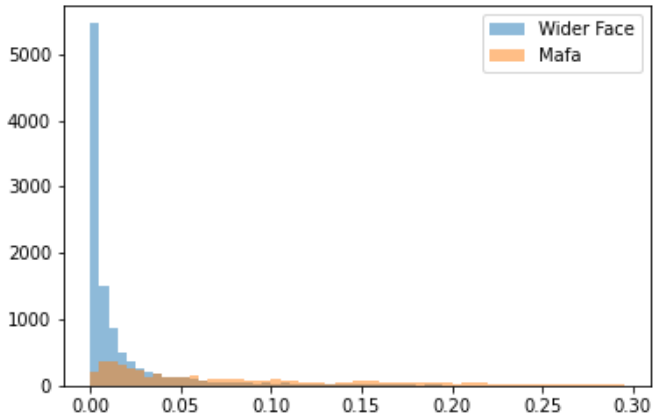


图 7 人脸相对大小分布

其次，考虑 FAN 相对于 RetinaNet 的改进，发现主要也在 Wider Face 上有一定的改进，这与文献中的实验结果，即 FAN 在 easy 数据集中没有改进，而在 hard 数据集中有较大改进相一致，说明引入注意力机制能够让模型更有能力检测较小的人脸。

最后，从检测速度来看四种模型均能实现实时检测，其中 SSD 由于属于 one-stage 方法，同时输入图像更小，因此检测速度最快；而 RetinaNet 可能由于实现上的原因，检测速度和 two-stage 的 Faster RCNN 持平。

### 3、 其他实验结果分析

首先，在使用不同预训练参数对 Faster RCNN 的影响的实验中，结果显示越是预先充分训练过的参数在训练时收敛越快，同时验证集上表现越好，这应该主要由于数据集较小，同时 epoch 也相对较小导致。在已发表论文中的随机初始化不比预训练差的结果，如果能够增大数据集，同时引入更多数据增广（如 mosaic 等），并充分训练至完全收敛，应该能够复现。

其次，在数据增广对模型的影响实验中，结果显示数据增广会让训练时的 loss 降低更加缓慢，但却能提高验证集上的准确率，说明数据增广的确能够提高模型泛化能力，提高训练效果。

### 4、 后续改进

限于个人的时间，本项目没有进行更加深入的。如有可能，还可以有如下的改进：首先，测试发现用手捂住脸基本上不会被识别为口罩，但是其他物体挡住脸部，尤其是纯白、纯黑、或纯蓝等近似的口罩色，则很容易会被识别为口罩。所以，为了提高鲁棒性，需要收集相关近似物体的图像，并利用人脸关键点数据来人工合成一系列数据来增强模型鲁棒性。其次，目前所用模型均为 2~3 年前提出的模型，而由于目标识别领域发展迅速，最新的方法已经领先了之前模型好多，如 EfficientDet、D2Det 等。如果利用最新的模型，则应该能够得到更优的结果。最后，目前的实现还较为粗糙，没有考虑检测速度的优化。如果考虑一些更细致的优化，如自动 half float 等技术；同时参考一些成熟的实现，如 mmdetection 或 detectron2 等，则可以有更快的检测速度。

## 六、结论

本项目基于三种主流的深度学习目标检测模型和一种针对遮挡人脸的改进模型，实现了较高准确率，以及实时速度下的人脸口罩检测；同时分析了不同模型的特点，以及不同参数、不同数据处理对结果的影响。

## 七、参考代码

models/faster_rcnn.py：参考了 https://github.com/pytorch/vision

models/ssd.py：参考了 https://github.com/sgrvinod/a-PyTorch-Tutorial-to-Object-Detection

models/retinanet.py：参考了 https://github.com/yhenon/pytorch-retinanet

models/fan.py：参考了 https://github.com/rainofmine/Face_Attention_Network

dataset.py 中 transforms 相关函数：参考同 ssd.py

engine.py 中 train 函数：参考同 faster_rcnn.py

metric.py：参考同 ssd.py

utils.py：参考同 faster_rcnn.py

调参时参考了 https://github.com/open-mmlab/mmdetection 中的 configs

# 八、参考文献


[1] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, pages 91 – 99, 2015.

[2] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. Ssd: Single shot multibox detector. In European conference on computer vision, pages 21–37. Springer, 2016.

[3] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*, pages 2980 – 2988, 2017.

[4] Jianfeng Wang, Ye Yuan, and Gang Yu. Face Attention Network: An Effective Face Detector for the Occluded Faces. In *arXiv preprint arXiv:1711.07246*, 2017.

[5] T.-Y. Lin, P. Doll´ar, R. B. Girshick, K. He, B. Hariharan, and S. J. Belongie. Feature pyramid networks for object detection. In CVPR, vol. 1, no. 2, 2017, p. 4.