

人脸口罩检测

自动化系 王圣杰 2019211242

摘要

如今新冠肺炎疫情在全球各地蔓延，研发各类防疫所需的设备是当下紧急且重要的任务，其中人脸口罩检测对提醒人们及时佩戴口罩和发现可能存在的隐患方面具有重要的意义。但是一般的目标检测模型精度和速度难以兼顾，所以不能在移动端使用。因此本文采用轻量化的 YOLOv3-tiny 以及 MobileNetv2-SSD 的轻量化模型对人脸口罩检测任务进行建模，最终经过参数调整后的 YOLOv3-tiny 模型实现了在测试集 mAP@[.5:.95]值达到了 0.52 的结果，且检测时间满足实时性。探究了网络超参数、数据增强、优化器、Anchor boxes 数量、NMS 类型、图片的尺寸对模型性能的影响，此外还分析了人脸不同特征区域对模型的影响以及不同模型在检测效果上的区别。最后针对模型的泛化性能，在人造佩戴口罩数据和艺术作品上验证了模型的性能；针对模型的鲁棒性，考虑真实相机可能遇到的干扰问题，通过加入椒盐噪声的数据增强方式使得在含有噪声的数据集上的性能仅损失 2%；针对模型的实时性，通过在 CPU 电脑上测试检测时间和模型大小，验证了模型具有迁移到 PC 端和移动端的能力。

关键词：目标检测、特征分析、模型鲁棒性、模型轻量化

1. 绪论

1.1 研究背景及现状

1.1.1 目标检测

人类对图像的感知超乎寻常，常常是一看到图片就知道物体的位置。人类的视觉系统快速和准确，帮助我们完成日常生活的各类事务。由此人们想到开发快速准确的目标检测算法替代人类完成如驾驶汽车等任务的系统是十分必要的。在深度学习流行之前，大部分的目标检测算法都是使用传统的机器学习算法，如 Cascade、HOG/DPM、Haar/SVM 以及上述方法的诸多改进、优化或者模型集成的方法。但在深度学习广泛应用于图像分类任务之后，目标检测任务也大量采用

深度学习的方式来实现，并取得了明显优于传统方法的结果。现有的目标检测任务主要分为两种思路，一种是 two-stage 模型，还有一种 one-stage 模型。其中 R-CNN 是最早提出的 two-stage 模型，其主要的贡献在于 1) 使用 CNN 网络用于区域的定位和分割物体，2) 用分类任务中训练好的模型作为基础网络训练检测任务，这两个做法深刻的影响了后续检测任务[1]。之后人们又提出了 Fast R-CNN 和 Faster R-CNN，主要从其耗时长的角度进行优化。Fast R-CNN 指明 R-CNN 网络耗时的原因在于其 CNN 网络在每个 proposal 区域没有共享计算，并对其进行改进[2]。而后的 Faster R-CNN 创新性更大，提出的 RPN 网络取代了之前的选择性搜索算法使得检测任务端到端的实现，这个方法后来被广泛使用[3]。One-stage 模型中非常著名的是 YOLO 算法，它的全称为“you look only once”，它的方法如同它的名字一样，将检测任务表述为一个统一的、端到端的回归问题，并且仅处理一次图片就得到位置和分类。它主要的特点有以下几个，速度快，全局的处理使得背景错误较少，在艺术作品检测上泛化性能强[4]。还有一类是 SSD 算法，其集成了 YOLO 算法的一些方法，但对比于 YOLO 算法采用多尺度的特征图，将不同的卷积段作为特征图输入回归器中提升小尺度物体检测精度。此外它还采用了更多的 Anchor boxes，并在 box 上预测概率而非划分的网格上[5]。之后针对两种模型的目标检测网络还有很多，在此不再继续展开讨论，但两者的方法都在不断借鉴中难以区分模型的属性[6]-[13]。

1. 1. 2 人脸口罩检测

针对现如今新冠肺炎的疫情需要，开发人脸口罩检测任务是我们科技行业的从业者应当为疫情防控任务所做的贡献。而对于人脸口罩识别其首先是一个二分类检测问题，主要分为戴口罩的人脸和未佩戴口罩的人脸两种情况，其次对于边框的预测是个回归问题。

在疫情发生以来有很多的公司开展了相关的业务，比如旷视、商汤、海康、百度都多家科技公司研发出了带有人脸检测算法的红外测温、口罩佩戴检测等设备，依图、阿里也研发出了通过深度学习来自动诊断新冠肺炎的医疗算法。其中腾讯优图针对戴口罩的人脸检测任务，开发了 DSFD 算法，在模型设计上进行局部特征增强，提升可见区域权重。同时针对口罩种类丰富、佩戴位置多样等问题，在数据增强方面设计相应策略，提升模型鲁棒性。目前，口罩场景下的人脸检测算法准确率超过 99%，召回率超过 98%。同时其还采用 FCN 网络做了口罩的属性识别，识别不正确的佩戴方式，识别准确率超过 98%[14]。此外，武汉大学也

开展了相关研究并开源了他们的数据集，其经过数据清洗和标注，有 525 人的 5000 张口罩人脸和 9 万正常人脸还有大量的模拟口罩人脸数据集[15]。

1.1.3 轻量化模型

针对我们所需要的人脸口罩检测任务，未来应该服务于各类防控一线，所以模型的运行效率和硬件成本是需要考虑的，模型的轻量化可以很好的解决实时性检测的问题。比如，Light Head R-CNN 就提出对检测头部进行轻量化的方法，在保持精度的前提下减少冗余的计算，实现精度和速度和平衡[16]。YOLOv2 网络也是进一步提高了其前身 YOLO 网络的速度，采用了比较多的技巧去处理，比如将 dropout 层改为 batch normalization，采用了更好的聚类来做先验生成 Anchor boxes，采用相比于 VGG 网络更高效的 Darknet-19，多尺度训练等等方法[17]。YOLOv3 网络进一步改进了前者，首先在损失函数中 softmax 损失改为 logistics 损失，将原先基础网络的 Darknet 和流行的 Resnet 结合，大量采用 3*3 和 1*1 卷积方式和 shortcut 连接，使得在小物体检测上又了很大的提高[18]。此外由此构建的 YOLOv3-tiny 网络减少了很多特征层的结构，在精度和速度上得到了权衡，这也是本文采用其作为人脸口罩检测任务的模型的原因。此外，基于计算量更少的基础网络结构也可以降低模型的复杂度，比如 MobileNet 的网络结构，对比于传统的 VGG 网络，计算量和参数量得到了极大的降低，并在多个分类、检测、分割的数据集上获得良好的结果[19]。此外，还有不少针对基础的 YOLO 算法做的模型轻量化设计[20]。

1.2 主要贡献

针对上述要求，本文主要使用 YOLOv3-tiny 模型以及 MobileNetv2-SSD 模型对人脸口罩检测任务进行建模，最终 YOLOv3-tiny 模型实现了在测试集上 mAP@[.5:.95] 值最大达到了 0.52 的结果。并且探究了超参数、数据增强、优化器、Anchor boxes 数量、NMS 类型、图片的尺寸对模型性能的影响，此外还分析了人脸不同特征区域对模型的影响以及不同模型在检测效果上的区别。最后针对模型的泛化性能，在人造佩戴口罩数据和艺术作品上进行验证。针对模型的鲁棒性，通过加入椒盐噪声的数据增强方式使得在含有噪声的数据集上的性能仅损失 2%。针对模型的实时性，通过在 CPU 电脑上测试检测时间和模型大小，验证了模型具有迁移到 PC 端和移动端的能力。

2. 数据整理

2.1 数据来源及内容

人脸口罩检测数据集使用来自于 AIZOO 网站所公开的口罩检测数据集。该作者开源了 7959 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。并且在其对数据集做了训练集和验证集的分配，6120 张图像为训练集，1839 位验证集。数据清洗工作由于开源数据集的作者已经做了相关工作，本文不再对数据进行清洗工作。

2.2 数据可视化

我们选择了几张数据集中的几张图片作为样本展示，如下图所示。可以看到在每张图片中有时有一个人脸图像，有时有多个人脸，这就对模型的细粒度提出要求，小物体的检测历来是目标检测的难点。从图片中也可以看到数据集中也包含了一些用手遮住口鼻的图片，其真实标签应该为未带口罩。这类数据会使得训练难度加大，但也防止了模型认为遮住口鼻就是带口罩的误判。这在后续实验中可以加一验证。

表 1 数据集分布

训练集中未佩戴口罩人脸个数	10588
训练集中佩戴口罩人脸个数	2985
测试集中未佩戴口罩人脸个数	1898
测试集中佩戴口罩人脸个数	1041

Face



Face_mask



图 1 数据集可视化

3. 模型设计

3.1 YOLOv3-tiny 模型

本文中我们采用的 YOLOv3 模型, 这里简要介绍下 YOLO 模型的构建过程。整个网络采用了卷积层、池化层和全连接层, 在网络结构上近似于分类或者回归的 CNN 网络。但是在最后输出层用线性函数做激活函数, 因为需要预测 bounding box 的位置, 而不仅仅是对象的概率。我们以 20 个类别 2 个 bounding box 为例, 整个算法首先将图像修整到 448*448 的尺寸大小, 经过网络预测得到 7*7*30 的图像块。7*7*30 矩阵中每个 1*30 的向量对应原始图像中 7*7 的网格, 代表每个网格中的预测结果。1*30 的向量分为 20 个对象的概率, 2 个 bounding box 的置信度和每个 bounding box 的四个角点位置。其中预测的置信度计算公式为:

$$Confidence = \text{Pr}(\text{Object}) * IOU_{\text{pred}}^{\text{truth}} \quad (1)$$

网络的整体损失函数设计为如下公式:

$$\begin{aligned}
& \lambda_{\text{coord}} \sum_{i=0}^{S^2} \sum_{j=0}^B \mathbb{1}_{ij}^{\text{obj}} \left[(x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2 \right] \\
& + \lambda_{\text{coord}} \sum_{i=0}^{S^2} \sum_{j=0}^B \mathbb{1}_{ij}^{\text{obj}} \left[(\sqrt{w_i} - \sqrt{\hat{w}_i})^2 + (\sqrt{h_i} - \sqrt{\hat{h}_i})^2 \right] \\
& + \sum_{i=0}^{S^2} \sum_{j=0}^B \mathbb{1}_{ij}^{\text{obj}} (C_i - \hat{C}_i)^2 \\
& + \lambda_{\text{noobj}} \sum_{i=0}^{S^2} \sum_{j=0}^B \mathbb{1}_{ij}^{\text{noobj}} (C_i - \hat{C}_i)^2 \\
& + \sum_{i=0}^{S^2} \mathbb{1}_i^{\text{obj}} \sum_{c \in \text{classes}} (p_i(c) - \hat{p}_i(c))^2
\end{aligned} \tag{2}$$

可以看出整体的损失函数分为分类误差，bounding box 的置信度误差和 bounding box 的位置误差。在预测结束后，还会对所有的 bounding box 做非极大抑制，其核心思想是：选择得分最高的作为输出，与该输出重叠的去掉，不断重复这一过程直到所有备选处理完。YOLO 网络的 NMS 的计算方法为，在每一个网格中，对象 C_i 位于第 j 个 bounding box 的得分为：

$$\text{Score}_{ij} = P(C_i|\text{Object}) * \text{Confidence}_j \tag{3}$$

由此遍历每个对象，选择最高的 Score 的对象和 bounding box，然后用 IOU 阈值的方法去掉重叠的候选对象。由此实现最后的定位和输出。

相比于 YOLOv1，YOLOv2 主要在以下几点做出改进，比如采用了 BN 层来提高泛化能力，并且支持了高分辨率的图像输入。此外，在预测框的选定上，Anchor boxes 的使用借鉴了 R-CNN 网络的结构，使得同时得到 1000 以上的预测框。在选择 anchor boxes 的种类是也是用 k-means 聚类来自动学习 Anchor boxes 的种类，度量方式采用了 IoU（交并比）。对于 Anchor boxes 坐标的预测，不再采用 R-CNN 中的坐标偏移量，而是采用将网格横纵坐标控制在 0-1 之间，用 logistic 预测坐标实现较小偏移，训练更为稳定。其他所采用的方法还有细粒度检测、多尺度训练还有基础模型的更改。

而相比于 YOLOv2，YOLOv3 网络在其上的主要改进在于基础网络 darknet-53 加入 Resnet 思想、多尺度特征借鉴 SSD 思想，实现了更为快速和准确的小物体检测任务，在此不多赘述。而由其模型构建的 tiny 模型由于简化了很多网络层，在精度和速度上做到了更好的平衡。由于我们的口罩检测实时性要求较高，

所以采用的是 YOLOv3-tiny 网络来实现整个模型。下图为 YOLOv3-tiny 的网络结构图。

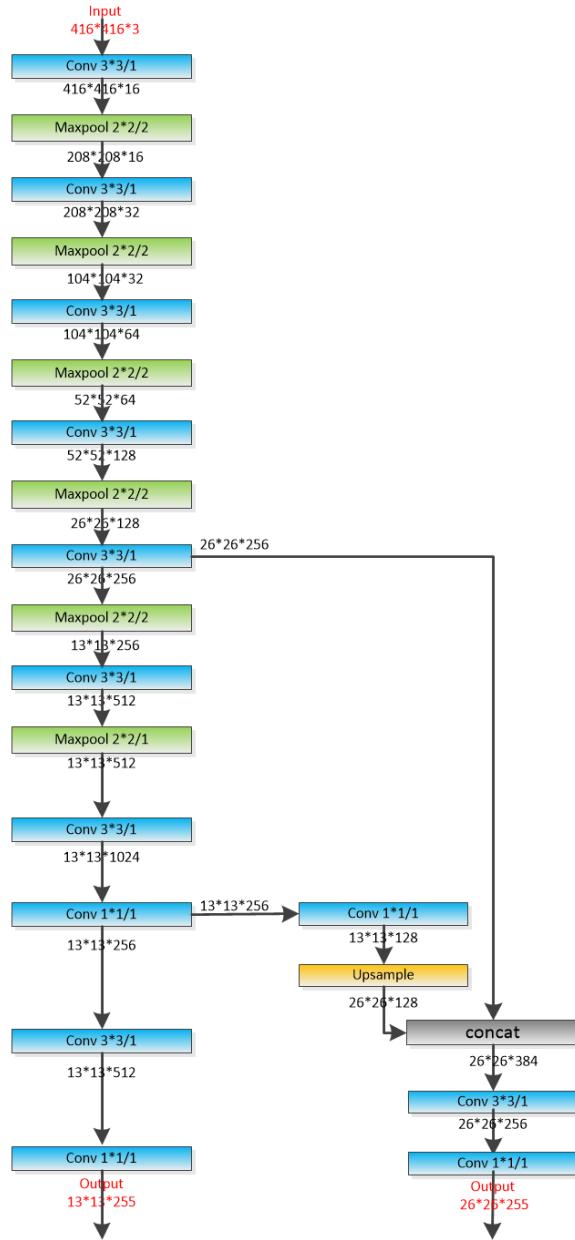


图 2 YOLOv3-tiny 模型

3.2 MobileNetv2-SSD 模型

SSD 是一种单阶段的目标检测网络，其主要的特点在于将边界框的输出空间离散化为一组默认框，该默认框在每个特征图位置有着不同的宽高比和尺寸。其实现了比 YOLO 检测器更快的速度和接近 Faster R-CNN 的检测准确率。SSD 方法在基础网络的基础上添加了辅助结构。1) 多尺度特征图检测：将卷积特征层添加到基础网络的末尾，得到多个尺度检测的特征图，在其上预测实现多尺度预

测。2) 卷积预测器：在添加的特征层上使用一组卷积滤波器预测每个类别的分数和产生相对于默认框的坐标偏移。3) 默认框及宽高比：默认框以卷积的形式平铺特征映射，以便每个框相对于其对应单元的位置是固定的，并且在每个位置的 k 个框采用不同的宽高比。

在训练过程中，对于真实标签框，需要从默认框中选择重叠和置信度最高的默认框为真实匹配默认框。训练的误差分为位置损失 (loc) 和置信损失 (conf) 的加权和：

$$L(x, c, l, g) = \frac{1}{N} (L_{\text{conf}}(x, c) + \alpha L_{\text{loc}}(x, l, g)) \quad (4)$$

其中 N 是匹配的默认框的数量，x 是预测的类别，l 为预测框，g 为真实标签框， α 为权重系数。对于位置损失，采用的是平滑的 L1 损失，公式如下所示：

$$L_{\text{loc}}(x, l, g) = \sum_{i \in \text{Pos}}^N \sum_{m \in \{cx, cy, w, h\}} x_{ij}^k \text{smooth}_{L1}(l_i^m - \hat{g}_j^m) \quad (5)$$

$$\hat{g}_j^{cx} = (g_j^{cx} - d_i^{cx}) / d_i^w \quad \hat{g}_j^{cy} = (g_j^{cy} - d_i^{cy}) / d_i^h \quad (6)$$

$$\hat{g}_j^w = \log(\frac{g_j^w}{d_i^w}) \quad \hat{g}_j^h = \log(\frac{g_j^h}{d_i^h}) \quad (7)$$

其中 d 代表默认框，cx 和 cy 代表框的中心坐标，w 和 h 代表宽度和高度。

置信损失是 softmax 损失对多类别置信和权重系数 α 设置为 1 的交叉验证。下式为具体的求解方法：

$$L_{\text{conf}}(x, c) = - \sum_{i \in \text{Pos}}^N x_{ij}^p \log(\hat{c}_i^p) - \sum_{i \in \text{Neg}} \log(\hat{c}_i^0) \quad (8)$$

$$\hat{c}_i^p = \frac{\exp(c_i^p)}{\sum_p \exp(c_i^p)} \quad (9)$$

此外，在不同的特征层还使用了不同的默认框比例来适应不同层次特征图的感受野的大小，在训练中由于大多数的默认框均为负样本，为了平衡训练的正负样本比例，使用默认框的最高置信度对负样本排序，挑选较高置信度的负样本。

应用于移动端架构的 MobileNetv2 所提出的倒置残差结构使得在运算时间和分类、检测、分割等任务的准确率得到了权衡。这种网络结构不同于 Resnet 的先升维再降维，而是先对输入进行低维压缩表示然后扩展到高维，使用更为轻量级深度卷积，然后使用线性映射投影到低维。其中值得一提的是 Deep-wise 卷积，这种卷积结构代替传统的 3D 卷积方式可以减少卷积核的冗余，计算和参数量得到极大的降低，对比与 VGG-16，其网络运算量和参数数量下降了 30 倍左右。并且文章中也对目标检测任务进行实验，发现使用 VGG 基础网络的 SSD 算法和 MobileNet 网络相比，较好的保持了精度，但极大的降低了运算量。所以本文考虑采用基于 MobileNetv2 基础网络架构的 SSD 算法实现人脸口罩识别任务。

3.3 评估指标

为了评估各种模型训练的算法性能，我们使用到的评估指标有查全率（recall）、查准率（precision）和均值平均精确度（mAP）等，具体的计算公式为：

$$\begin{aligned} \text{Precision} &= \frac{TP}{TP+FP} \\ \text{Recall} &= \frac{TP}{TP+FN} \\ mAP &= \frac{1}{|Q_R|} \sum_{q \in Q_R} AP(q) \end{aligned} \quad (10)$$

其中，TP 代表真实存在目标且被检测出的实例个数；FP 代表不存在目标但被检测出来的实例个数；FN 为实际存在目标但未被检测出的实例个数。综上而言，查全率（recall）表示图像中被正确检测到的数量占所有目标数量的比例；查准率（precision）表示被正确检测到的数量占所有检测出的数量的比例。由于单独使用上述两者都无法评估模型的有效性，所以人们才用了平均精确度（AP），其代表 Precision-Recall 曲线下的面积。而其中的 mAP 表示所有类别的平均精度求和除以类别均值。

除此之外，在分类问题中 F1-Score 的指标也是衡量多分类任务的重要指标，是查全率和查准率的调和平均数。对于单个类别的 F1-Score 的计算公式如下：

$$F1 = 2 \frac{\text{Recall} * \text{Precision}}{\text{Recall} + \text{Precision}} \quad (11)$$

4. 实验设计及结果

4.1 数据集划分

对数据集的数据进行分析，AIZOO 公开的人脸口罩识别已经对训练集和测试集做了划分，有 6120 张图片为训练集，1839 张图片为测试集。除此之外，在测试集中取出一半作为验证集在训练过程中寻优。

4.2 模型结果对比

下表格为各个模型在人脸和带口罩人脸两个类别上的 mAP 值，并且分别计算了不同 IoU 阈值下的 mAP。从图中可以看出不同的模型虽然有些区别，但整体上对于该问题的解决都给出了较好的结果。从整体的 mAP 上看，YOLOv3-tiny 模型的结果最好，获得 0.497 的 mAP@[.5:.95]。单纯看人脸的检测效果，YOLOv3-

tiny 模型结果更优，相比于 MobileNetv2-SSD 模型，其在 mAP@[.5:.95]上得到了 17%的提升；单纯看带口罩人脸的检测效果，YOLOv3-tiny 模型结果更优，相比于 MobileNetv2-SSD 模型，其在 mAP@[.5:.95]上得到了 2.6%的提升。综上所述，YOLOv3-tiny 模型的综合结果更优。但我们也注意到两个模型对于人脸的识别的效果均弱于佩戴口罩的人脸检测，分析首先在于数据集中的人脸数量和佩戴口罩人脸数量的不均衡导致。YOLOv3-tiny 模型对比 MobileNetv2-SSD 模型有更好的表现，也和其运用的技巧更多所致。

表 2 模型性能对比

模型	类别	mAP@.5	mAP@.7	mAP@.9	mAP@[.5:.95]
YOLOv3-tiny	Face	0.866	0.67	0.019	0.471
	Face_mask	0.899	0.726	0.027	0.523
	all	0.883	0.698	0.023	0.497
MobileNetv2-SSD	Face	0.749	0.512	0.016	0.403
	Face_mask	0.869	0.728	0.028	0.537
	all	0.810	0.619	0.022	0.471

下图为绘制不同模型在不同 IoU 阈值下每类的 Precision-Recall 曲线，分别为 IoU 阈值 0.5, 0.7 和 0.9。通过上述曲线可以找到不同阈值下的最优的查全率和查准率。在这个指标下，也同样可以看出 YOLOv3-tiny 模型相比于 MobileNetv2-SSD 模型效果更好。

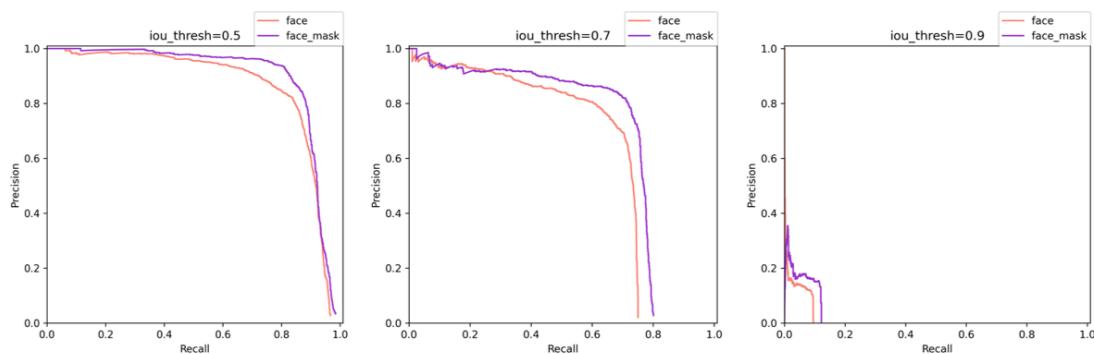


图 3 YOLOv3-tiny 的各个阈值下的 Precision-Recall 曲线

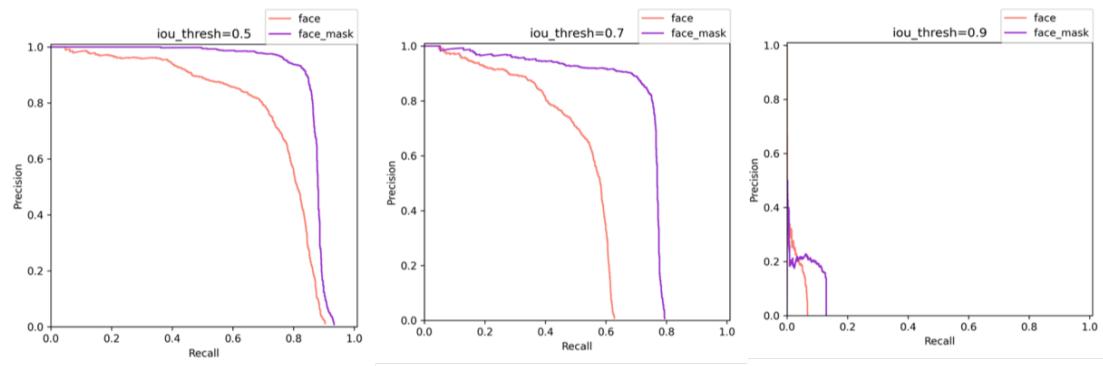


图 4 MobileNetv2-SSD 模型的各个阈值下的 Precision-Recall 曲线

由此也可以得到不同置信度阈值下的查全率和查准率，上图的计算结果，也可以看出随着置信度阈值的增加，查准率逐渐升高，而查全率逐渐降低，因此需要选择一个均衡点，使得查全率和查准率都具有比较好的表现。在具体选择上，可以根据 Precision-Recall 曲线上的选择查全率和查准率的折中点所对应的置信度阈值。

4. 3 超参数表现

对于 YOLOv3-tiny 模型来说，主要的超参数有坐标误差中的 IoU 的阈值 λ_{iou_t} ，分类误差的权重 λ_{cls} ，网格中物体置信度误差权重 λ_{obj} ，以及边框误差中的权重 λ_{giou} ，我们研究了在不同超参数下的模型 mAP、查全率、查准率和 F1-score 值变化，得到了下图所示曲线。由此可以寻找到最优超参数下的模型。从图中可以发现，和一般的神经网络一样，其超参数的选择根据不同的问题有不一样的结果，通过上述的选择，可以看出当 iou_t 选择在 0.2-0.4 之间效果较好，当 λ_{cls} 和 λ_{obj} 选择为 20 时，mAP 值可以得到相对较高的值，而对于 λ_{giou} ，则是取 5 时得到相对最优的性能。

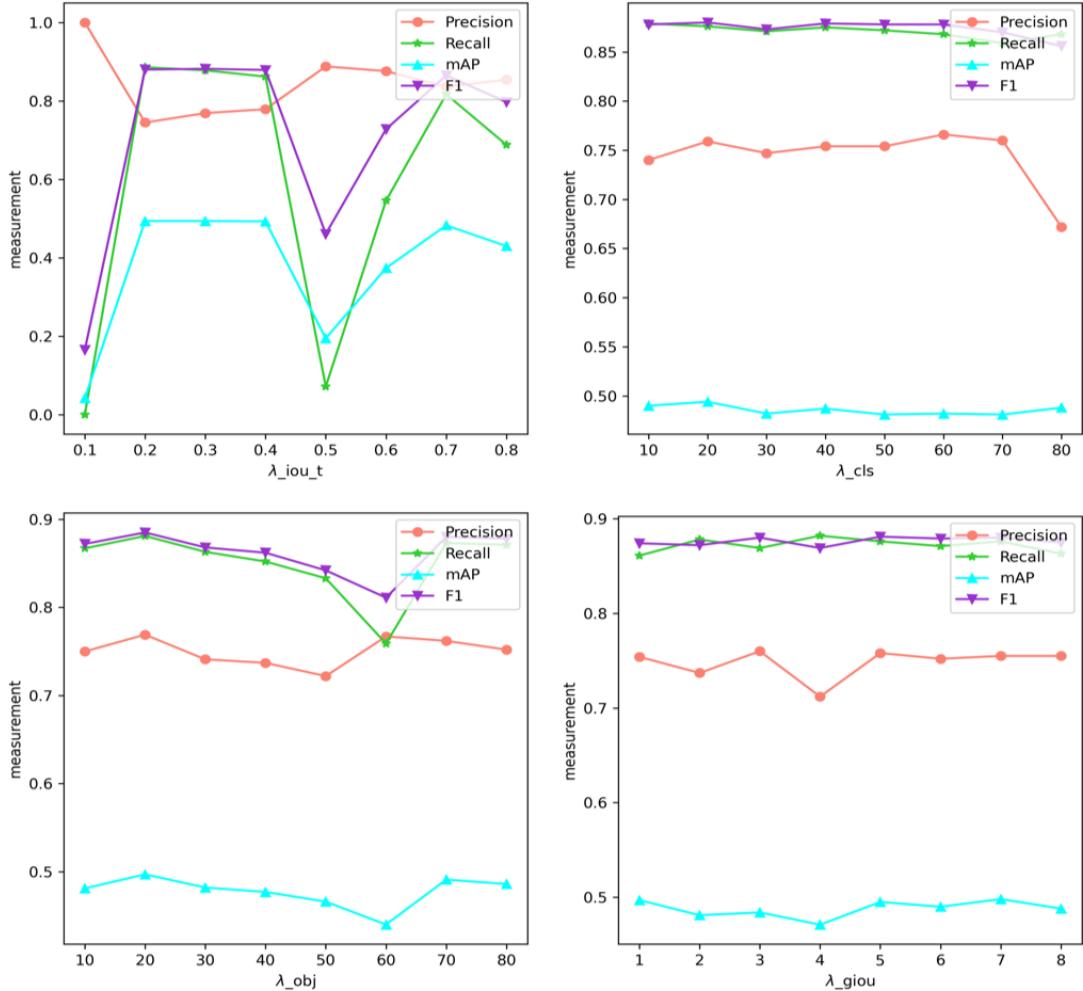


图 5 YOLOv3-tiny 模型的超参数和最终测试集结果的参数实验

对于 MobileNetv2-SSD 模型，其超参数较少，主要是坐标误差的权重 α ，坐标误差中的 IoU 的阈值 λ_{iou_t} 和所采取的正负样本比例系数 γ 。可以看出对于我们的问题， α 的系数较小时，在验证集上的分类和回归的误差更小，从原理上也可以理解，针对我们这个二分类问题，人脸和带口罩人脸时较为相似的，所以分类误差更为重要。而对于 iou_t ，则基本没有什么影响，说明其对该参数不敏感。此外，对于正负样本比例 γ ，则有所不同，原文 MobileNetv2-SSD 中采用的比例时 1:3 的正负样本比例获得较好的结果，但实验说明正负样本均衡可能效果更优，当然这和我们两分类的问题也有很大关系。本文中并未对学习率的超参数进行选取，而是都使用了现今比较流行的余弦学习率衰竭从而让其对该超参数不敏感。

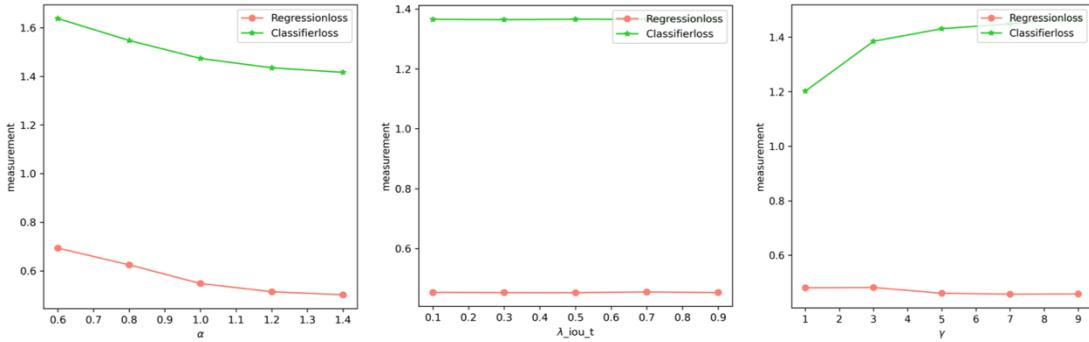


图 6 MobileNetv2-SSD 模型的超参数和最终测试集各部分 loss 值结果的参数实验

5. 实验结果分析

5.1 消融实验

5.1.2 数据增强

在面对如此小样本学习下，如何提高其泛化性能是我们所需要关注的，这样才能在实际使用中获得较好的性能。一般所采用的提高泛化能力的方式为数据增强来丰富样本，从而增强目标检测算法的准确性和鲁棒性。其中几种最可能在真实情况下出现的图像变化为亮度和对比度的变化。由于真实环境光照等外部环境的变化，亮度和对比度的变化是经常发生的。此外，我们知道图像的 HSV 空间包含色调、饱和度和亮度这三个信息。通过三者的微小调节，可以实现对亮度和对比度等信息的调整。

下表格 YOLOv3-tiny 模型为不做 HSV 空间变化和变化后的 mAP@[.5:.95]、F1-score 值的对比。此外，还对比了经典的图像预处理操作，如随机的旋转、平移、尺度变换等处理方式。可以看出经过预处理操作后的模型具有更好的泛化性能，在 mAP 等值上得到了提升。但是对于 HSV 空间的变化，模型并在该数据集上表现更好，分析可能是数据集图片较少，并不会带来大的泛化性能提升。

表 3 YOLOv3-tiny 性能对比

指标	YOLOv3-tiny+HSV 变换	YOLOv3-tiny	YOLOv3-tiny + 数 据增强
mAP@[.5:.95]	0.488	0.481	0.49
F1-score	0.877	0.876	0.877

此外，我们也探究了 MobileNetv2-SSD 模型在是否进行图片预处理的效果对结果的影响，可以得到图像的预处理工作对其模型的结果有一定的影响，mAP@[.5:.95]值从 0.464 值变化到 0.471 值。这也充分说明在训练中对数据的预处理十分重要。

表 4 MobileNetv2-SSD 性能对比

模型	Face	Face_mask	All
MobileNetv2-SSD+数据增强	0.403	0.537	0.471
MobileNetv2-SSD	0.381	0.546	0.464

5. 1. 3 优化器选择

对于模型优化器的选择，深度学习经过多年的发展，在基础的 SGD 优化器的基础上研究了多种的优化器，比如现在在多种深度学习任务上获得较好结果的 Adam 优化器。本节我们修改了优化器对 YOLOv3-tiny 模型训练，将原始的 SGD 优化器改进为 Adam 优化器，可以得到如下的对比结果。可以看出 Adam 优化器，mAP 的结果获得明显的提升，提升了 2%，说明在该类问题中优化器的选择也会进一步提升效果。

表 5 YOLOv3-tiny 性能对比

指标	YOLOv3-tiny+Adam	YOLOv3-tiny+SGD
mAP@[.5:.95]	0.498	0.487
F1-score	0.87	0.874

对于 MobileNetv2-SSD 模型，我们也对其采用 Adam 优化器进行训练，对比于传统的 SGD 优化器，结果并未发生较大的变化，但是 mAP 值也得到了提升。下表为使用 Adam 和 SGD 的每类 mAP 值变化对比表格。

表 6 MobileNetv2-SSD 性能对比

模型	Face	Face_mask	all
MobileNetv2-SSD+SGD	0.403	0.537	0.471
MobileNetv2-SSD+Adam	0.399	0.547	0.473

5.1.4 NMS 类型

一般在检测模型的后处理操作中均会包含 NMS (Non-Maximum Suppression 非极大抑制)，其基本的作用是去除重合度 (IoU) 较高的预测框，保留预测分数最高的预测框作为检测输出。Soft NMS 提出在传统的 NMS 中，最高预测分数预测框重合度超过一定的阈值的预测框会被直接舍弃，但这样不利于相邻物体的检测。所以其提出的改进方式为根据 IoU 将预测框的预测分数进行惩罚，然后再根据分数过滤。根据这一思想，我们将传统的 NMS 方式进行改进，对预测分数利用高斯函数进行加权，由于引入了高斯函数的方差 σ 的超参数，所以我们对其进行研究，对于 YOLOv3-tiny 模型，可以发现 Soft NMS 运算方式对结果的影响不大，此外，较小的方差对于我们的问题更为适用。

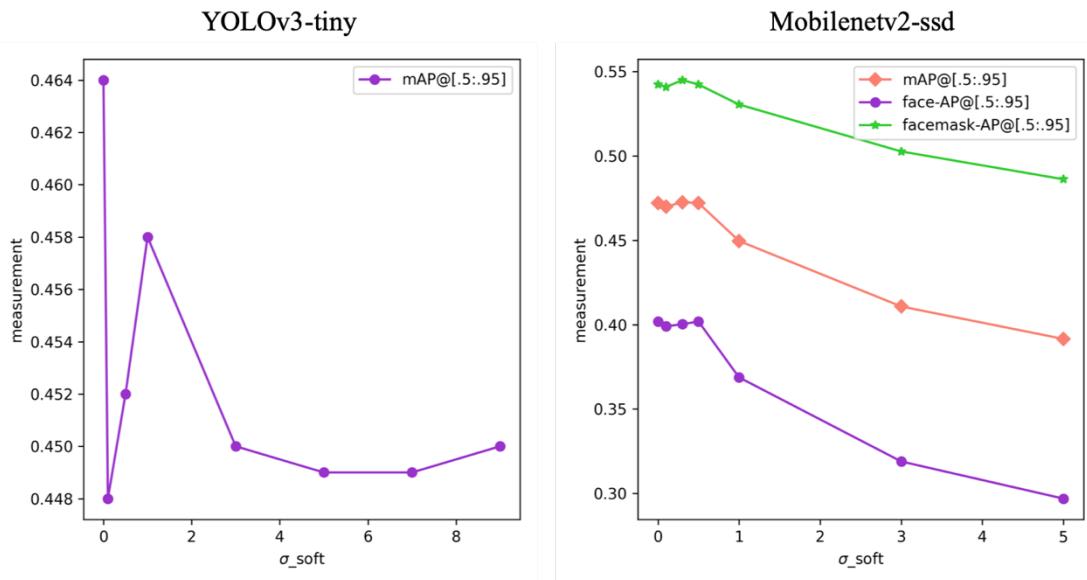


图 7 YOLOv3-tiny 和 MobileNetv2-SSD 模型使用 Soft NMS 下关于方差 σ 的 mAP 值变化

而对于 MobileNetv2-SSD 模型，我们使用 Soft NMS 对比于基础的 NMS 算法，可以发现同 YOLOv3-tiny 模型类似的结果，但其在方差超参数为 0.5 时，略微好于基础算法，但如果超参数 σ 过大，结果会迅速以线性方式降低。

此外，在后续看到在训练中对于 NMS 的处理上，使用了 Merge NMS 来用其余重叠边框的预测分数的均值作为权重对边框的位置进行修正，使得框的位置更为精准。对比结果可以发现这种 NMS 处理方法极大提高了 mAP 值。但 F1-score 的值却发生了降低，这可能和查全率的增大有关。

表 7 YOLOv3-tiny 性能对比

指标	YOLOv3-tiny+Merge NMS	YOLOv3-tiny
mAP@[.5:.95]	0.498	0.464
F1-score	0.87	0.881

5.1.5 Anchor boxes 数量

在 YOLOv3-tiny 实验中, 对于 Anchor boxes 的选择我们并未采用数据集标注的大小做 k-means 的评估, 而是直接采用了 COCO 数据集中的初始化边框大小。下表格为对比的在采用 k-means 初始化边框个数和 coco 数据集的初始化 6 个边框的指标对比。可以看出对于 Anchor boxes 的个数选择, 过多和过少都会让检测效果降低, 此外可以发现使用更大的数据集如 COCO 上产生的 Anchor boxes, 因为这样泛化的性能更好, 所以在测试集上获得了更高的 mAP 值。

表 8 YOLOv3-tiny 性能对比

指标	YOLOv3-tiny+4 anchor	YOLOv3-tiny+6 anchor	YOLOv3-tiny+8 anchor	YOLOv3-tiny+6 anchor (COCO)
mAP@[.5:.95]	0.452	0.481	0.471	0.49
F1-score	0.857	0.871	0.872	0.877

5.1.6 多尺度图片

由于输入图像是千差万别的, 探究输入模型的图片的大小对预测结果的影响, 可以更好的让我们知道模型或者数据的改进方式, 有利于我们的进一步提升性能, 所以我们探究了 YOLOv3-tiny 模型, 由于其采用了多尺度训练后, 但多尺度图片的表现由于任务不同最优的参数并不一定相同。下图是采用不同的输入尺寸的测试集得到的结果, 可以看出当输入尺寸为 512 时, 得到了最高的 mAP 值 0.497, 这说明图片的大小在这个范围的测试集得到检测效果更优, 在实际的测试环境可以使用这个尺寸作为实时运行的输入。但其中值得注意的一点在于, 每类的 mAP 的最大值不是在同一个输入尺寸达到, 而且当输入尺寸最大时, 两类的 mAP 值最为接近。

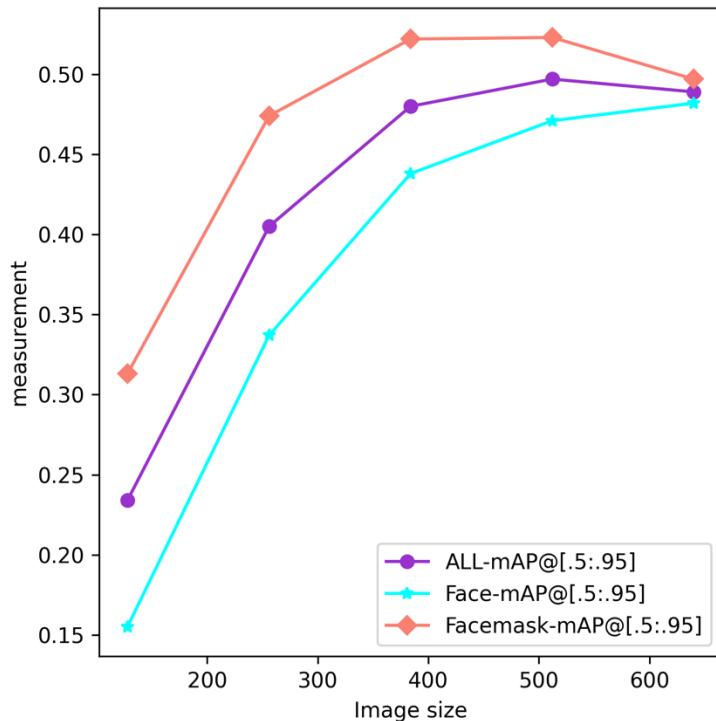


图 8 YOLOv3-tiny 模型检测效果随输入图片尺寸大小的变化

5.2 特征的重要性分析

通过遮挡人面部的各个区域来分析最终结果的不同，可以有助于我们对模型学习到的特征进行分析，在下述实验中，我们通过对人的面部重点区域进行遮挡，如人眼、鼻子、嘴巴和眉毛等，在两个模型预测后所给出的结果举例如下。可以看出人眼的遮挡并不会导致模型预测失败，仅单独遮挡鼻子、嘴巴和眉毛也不会出现明显的错误，但置信度的降低不同。遮挡眉毛的置信度下降最低，其次分别为眼睛、鼻子和嘴巴。其中遮挡嘴巴所下降最多，达到了 24%。这也反映了模型学习到的脸部特征的重要程度不同，嘴巴部位的重要性最大。但如果同时遮挡住鼻子和嘴巴模型检测为戴口罩人脸。



图 9 遮挡人脸部分区域实验

此外，我们还研究了不同遮挡物的颜色对模型的影响，发现如果采用接近人类脸部的肤色作为遮挡物，那么模型不会产生太大的波动，在预测上降低有限。但如果采用黑色或者蓝色作为遮挡物，就会预测出存在戴口罩人脸。而且发现蓝色的遮挡物比黑色、棕色和橙色等颜色检测的置信度更高，这也反映在实际过程中蓝色口罩的数量要多于其余颜色，同样符合实际。



图 10 不同颜色遮挡物区域实验

此外我们探究了数据集中包含的用手捂脸的图片的重要性。在实际的检测中，可以想象到的是如果用手捂住口鼻即可通过测试，那么说明我们的口罩检测系统存在不合理的漏洞。通过将这样的数据加入检测数据集，可以实现对捂住口鼻的人脸进行检测。下图为未加入遮挡口鼻数据和加入后训练结果图的对比。由图可以看出加入了遮挡口鼻的训练集后可以检测出仅仅遮挡口鼻是无效的口罩佩戴，符合实际使用的要求。（对于无口鼻的数据集采用了 B 站 up 主 HamlinZheng 的数据集）

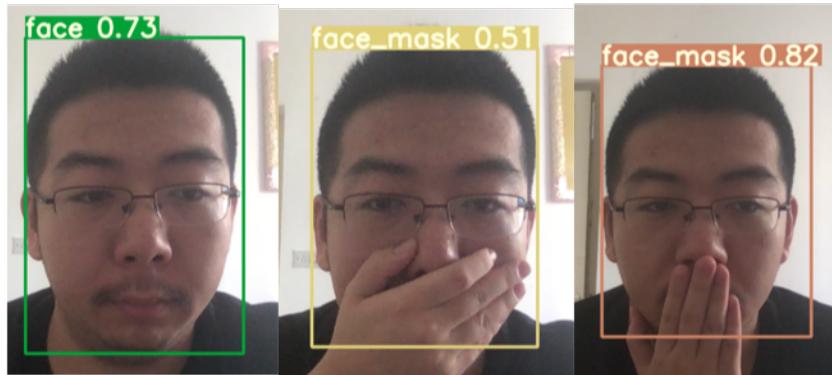


图 11 未加入手部遮挡面部数据的测试效果

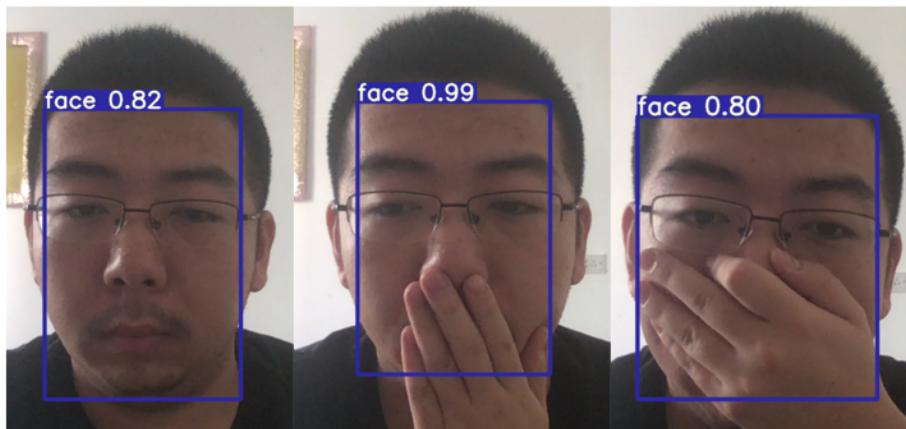


图 12 加入手部遮挡面部数据的测试效果

5.3 模型及结果可视化

在模型结果的可视化方面我们以案例分析的方式来具体观察不同情况下的不同模型的结果。

5.3.1 人脸口罩佩戴检测

下面举测试数据集中的几个比较典型的人脸和口罩图片的识别效果展示。下图为两个模型下的预测框展示，可以看到基本框选到了人脸的大部分区域，但有时当图片中存在多个小物体时，YOLOv3-tiny 模型预测的效果要比 MobileNetv2-SSD 模型预测效果好，找到了更多的人脸。但对于重叠度高的物体的图片，MobileNetv2-SSD 模型能够实现更好的预测效果，而 YOLOv3-tiny 模型却无法做出正确的预测。其原因可能是 YOLOv3-tiny 模型在训练中多尺度训练以及降采样的技巧使得网络在预测小物体上有更好的效果。而 MobileNetv2-SSD 模型由于采用了更为简单的 MobileNet 网络，其中的 Deep-wise 卷积结构，在对于不同位置的重叠的图片识别效果更好。

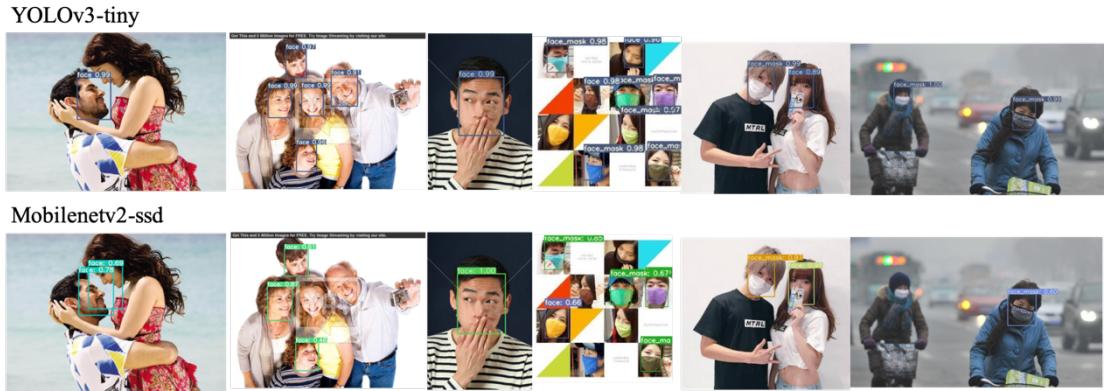


图 13 两个模型的对比可视化图

5.4 模型泛化性分析

5.4.1 人造口罩检测

虽然我们没有在数据集中加入虚拟口罩的图片，但在最终测试中我们制作了佩戴人脸虚拟口罩的脚本，实现对人脸的虚拟口罩佩戴。下图为佩戴虚拟口罩的识别效果展示。可以看出模型对比佩戴前，都精准地识别到了佩戴口罩的人脸。说明了模型对于口罩的检测泛化性能较好，学习到了口罩的位置和颜色等信息。两种模型效果一致，此处不一一展示。



图 14 人造口罩实验

5.4.2 艺术作品中人脸识别

对于真实环境的人脸已经可以实现正确的识别，但在原始的 YOLO 论文中探究了艺术作品下对物体的识别，并且实现了较好的识别效果。所以考虑到真实环境中复杂多变的情况，我们对艺术作品的人脸进行检测，发现其也可以实现对人脸的检测，而且对于 MobileNetv2-SSD 模型，取得非常不错的效果，在艺术作品上的泛化能力很强。但是在一些图片的预测中 YOLOv3-tiny 模型表现一般，但由于是艺术作品，我们也很难说那个模型的表现就差。但可以说明的是两者在对识

别口罩的特征的选择上有着不同的理解。这部分需要后续重新制作该类艺术作品数据集进行测试，才能更为详细的讨论性能。

YOLOv3-tiny



Mobilenetv2-ssd



图 15 艺术作品上的测试结果图

5.5 模型的鲁棒性

在神经网络的鲁棒性研究中，研究者曾经探究了在图片中叠加一定的噪声后，经过训练后的神经网络后，图片分类结果出现了明显错误，从而极大引起大家对神经网络鲁棒性的研究。在这个问题上，我们考虑到在实际部署在户外情况下摄像头难免会收到干扰，拍摄的图像可能出现各类噪声，其中以椒盐噪声最为常见，所以对其可以抵抗的椒盐噪声的强度进行分析有利于我们实际部署到户外情况。下表格为 YOLOv3-tiny 模型和 MobileNetv2-SSD 模型采取随机 50%的不同强度椒盐噪声的各类 AP 值和 mAP@[.5:.95]值的评估。可以看出随着椒盐噪声强度的增加，评估的指标在不断下降，其中 yolov3-tiny 模型随着 SNR 的降低，带口罩人脸的 mAP 值迅速下降，在 SNR 取 94%，性能下降 24%左右，而 MobileNetv2-SSD 模型性能下降 19%。总的来说，在抗椒盐噪声干扰中，MobileNetv2-SSD 模型的表现要优于 YOLOv3-tiny 模型。这同样可能和 MobileNet 的 Deep-wise 卷积方式有关。

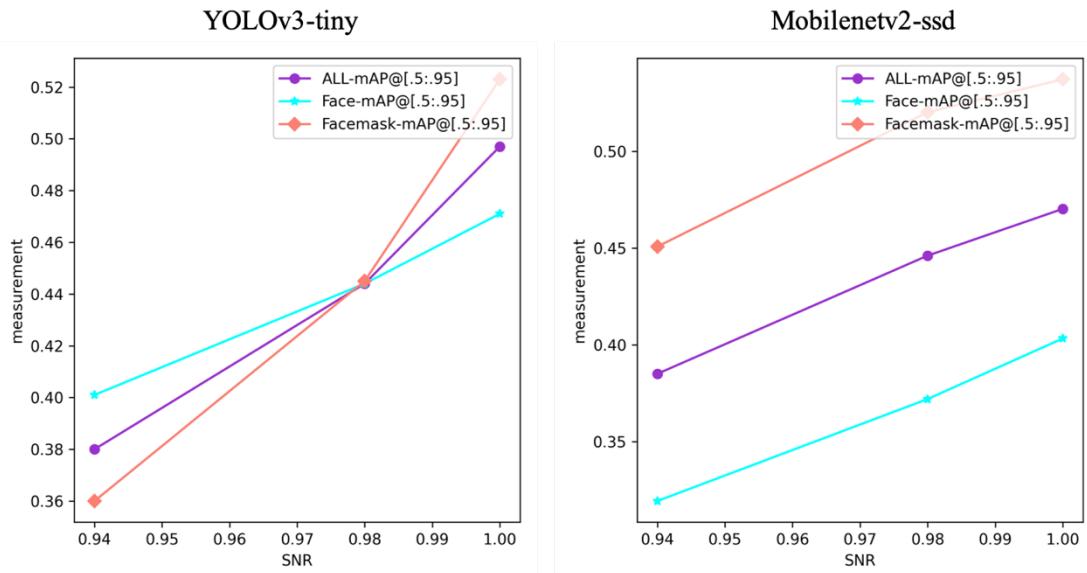


图 16 YOLOv3-tiny 模型和 MobileNetv2-SSD 模型随 SNR 的变化 mAP 值的变化

但是单纯知道了模型抵抗椒盐噪声的能力较弱还不够，但这启发我们在训练中加入这部分的数据增强方式来抵抗椒盐噪声，通过随机的加入一定范围的噪声，实现了在 SNR 为 0.94 和 1 的 mAP@[.5:.95]值接近一致的效果。可以看出 YOLOv3-tiny 模型从刚开始的下降 24%，到数据增强后仅下降了 2%。可以看出在数据增强中加入一定的噪声使得模型的鲁棒性有了很大的提高。

表 9 噪声实验性能对比

模型	SNR	Face	Face_mask	All
MobileNetv2- SSD+randomnoise	1	0.336	0.511	0.423
YOLOv3- tiny+randomnoise	0.94	0.330	0.517	0.424
YOLOv3- tiny+randomnoise	1	0.471	0.54	0.505
YOLOv3- tiny+randomnoise	0.94	0.46	0.529	0.495

下面为图片的测试结果。可以看出经过一定的椒盐噪声的数据增强处理，对噪声具有一定的抵抗能力。

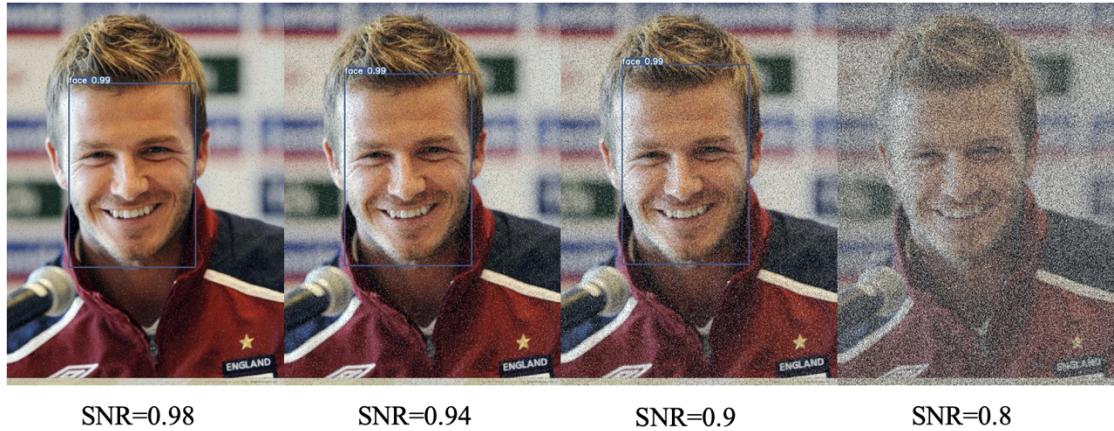


图 17 经过数据增强的不同信噪比下的检测效果

5.6 模型的实时性

由于我们所采用的两个检测模型最大的特点是实时性强，所以其具有部署普通 CPU 的 PC 电脑上的能力，我们采用 2.6GHz Intel Core i7 的 CPU 计算机上运行测试 python 代码。测试了两个模型在同一 CPU 上的检测速度和占据的内存大小，下表为平均的推断时间和模型大小。模型的存储方式均为 Pytorch 自带的后缀.pth 的格式。

表 10 实时性检测性能对比

指标	YOLOv3-tiny	MobileNetv2-SSD
平均检测时间	0.112s	0.091s
占据的内存空间	35.7MB	12.6MB

可以看出两者的推断时间差不多，但 MobileNetv2-SSD 模型的参数和运算更少，所以推断的时间也更短，占用的空间小使得它更适合在移动端部署。这也和它基础网络当时的定位十分符合。

6. 结论

本文在人脸口罩检测任务中使用 yolov3-tiny 模型以及 MobileNetv2-SSD 模型对任务进行建模，最终两类模型均实现了较好的对人脸和佩戴口罩人脸的检测功能。并且探究了超参数、数据增强、优化器、Anchor boxes 数量、NMS 类型、图片的尺寸对模型性能的影响，此外还分析了人脸不同特征区域对模型的影响以及不同模型在检测效果上的区别。最后针对模型的泛化性能，在人造佩戴口罩数

据和艺术作品上进行验证。针对模型的鲁棒性，考虑真实相机可能遇到的干扰问题，通过加入椒盐噪声的数据增强方式使得在含有噪声的数据集上的性能仅损失2%。针对模型的实时性，通过在CPU电脑上测试检测时间和模型大小，验证了模型具有迁移到PC端和移动端的能力。模型的整体效果具有迁移到配置要求较低的硬件中，这使得检测成本进一步降低，更有利于未来的落地实现，但也希望疫情可以早日结束，愿国泰民安，明天更好。

致谢

这项工作得到了清华大学模式识别课程的张长水老师和各位助教的大力支持。由于我校在冠状病毒流行期间入校审核严格，我们感谢清华大学导航、制导与控制实验室的师兄师姐协调的研究资源。部分计算是在清华大学导航、制导与控制实验室服务器（“彭于晏”）系统上完成的。

代码链接

本文所采用的 yolov3-tiny 算法和 mobilenetv2-ssd 算法的搭建上参考了公开的代码，具体的参考链接见 README.md 文件。其中 yolov3-tiny 参考部分为模型的构建 model.py 文件和必要的模块 utils 文件夹，其余的训练、测试代码进行改写来实现本实验的要求。数据读取处理、超参数寻优、可视化、其余本论文中实验的代码为个人编写。mobilenetv2-ssd 参考代码类似，参考部分为其提供的模块文件夹 vision，其余的训练、测试代码进行了改写来实现本实验要求。数据读取处理、超参数寻优、可视化、其余本论文设计实验的代码为个人编写。

参考文献

- [1] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Richfeature hierarchies for accurate object detection and semanticsegmentation [C]// In Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on, pages 580–587.
- [2] Ren S , He K , Girshick R , et al. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2015.
- [3] 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.

- [4] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 779–788, 2016.
- [5] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexan- der C Berg. Ssd: Single shot multibox detector. In European conference on computer vision, pages 21–37. Springer, 2016.
- [6] Dai J, Li Y, He K, et al. R-FCN: Object Detection via Region-based Fully Convolutional Networks[J]. arXiv: Computer Vision and Pattern Recognition, 2016.
- [7] Dai J, Qi H , Xiong Y , et al. Deformable Convolutional Networks[C]// 2017 IEEE International Conference on Computer Vision (ICCV). IEEE, 2017.
- [8] Lin T, Dollar P, Girshick R, et al. Feature Pyramid Networks for Object Detection[C]. computer vision and pattern recognition, 2017: 936-944.
- [9] Shrivastava A, Sukthankar R, Malik J, et al. Beyond Skip Connections: Top-Down Modulation for Object Detection[J]. arXiv: Computer Vision and Pattern Recognition, 2016.
- [10] Fu C Y , Liu W , Ranga A , et al. DSSD : Deconvolutional Single Shot Detector[J]. 2017.
- [11] Kong T , Sun F , Yao A , et al. RON: Reverse Connection with Objectness Prior Networks for Object Detection[J]. 2017.
- [12] Li Z, Zhou F. FSSD: Feature Fusion Single Shot Multibox Detector.[J]. arXiv: Computer Vision and Pattern Recognition, 2017.
- [13] Zhang S, Wen L, Bian X, et al. Single-Shot Refinement Neural Network for Object Detection[C]. computer vision and pattern recognition, 2018: 4203-4212.
- [14] 腾讯优图攻克口罩识别难题，口罩佩戴识别准确率超过 99%, [Online] <https://baijiahao.baidu.com/s?id=1659229413418885625&wfr=spider&for=pc>.
- [15] Zhongyuan Wang, Guangcheng Wang, Baojin Huang, Zhangyang Xiong, Qi Hong, Hao Wu, Peng Yi, Kui Jiang, Nanxi Wang, Yingjiao Pei, et al. Masked face recognition dataset and application. arXiv preprint arXiv:2003.09093, 2020.
- [16] Li Z, Peng C, Yu G, et al. Light-Head R-CNN: In Defense of Two-Stage Object Detector[J]. arXiv: Computer Vision and Pattern Recognition, 2017.
- [17] Joseph Redmon and Ali Farhadi. Yolo9000: better, faster, stronger. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 7263–7271, 2017.
- [18] Redmon J, Farhadi A. YOLOv3: An Incremental Improvement[J]. arXiv: Computer Vision and Pattern Recognition, 2018.
- [19] Rosenfeld A , Tsotsos J K . Incremental Learning Through Deep Adaptation[J]. IEEE Transactions on Pattern Analysis & Machine Intelligence, 2017.
- [20] Rachel Huang, Jonathan Pedoeem, and Cuixian Chen. Yolo-lite: a real-time object detection algorithm optimized for non-gpu computers. In 2018 IEEE International Conference on Big Data (Big Data), pages 2503–2510. IEEE, 2018.