

所在组别	2021 年中国高校大数据挑战赛	参赛编号
研究生		bdc210702

人脸口罩佩戴检测研究

摘要

新冠疫情在全球的蔓延，危害着全球 70 多亿人口的生命财产安全，虽然我国的疫情已经总体得到遏制，但国外疫情依旧严重，由于境外输入导致的小规模新冠疫情在我国时常发生。目前我国要求人员在乘坐火车、地铁等公共交通工具，和出入商场、医院等人员聚集场所时必须佩戴口罩。针对人员是否佩戴口罩，现在采用的主要方式是人工查看，在固定的出入口配备检查人员，这种方式不具有全天性，且在人群大流量流通区域容易出现漏检的情况，这样给防疫安全工作带来了很大的隐患，因此实现人脸口罩的自动化识别就能解决这些问题，有效的检测人群佩戴口罩的情况，给我国的防疫工作智能化做出贡献，抑制新冠疫情在我国的传播，保护人民群众的生命安全。

对于问题一，通过分析标签文件，理解标签文件中包含的对应图片尺寸、人脸坐标信息，以及人脸佩戴口罩情况，将标签信息在对应的图片上可视化。选用 Python 语言结合 OpenCV 包编写程序，对于“佩戴口罩、未佩戴口罩、未正确佩戴口罩”三种情况，分别选用黄、红、绿三种目标框进行框选，并且在方框的下方显示相应的人脸佩戴口罩情况“with_mask、without_mask、mask_wearred_incorrect”标注。

对于问题二，结合成熟的深度学习目标检测算法，和本题的实际应用背景，考虑人脸口罩检测的精确性和实时性，选用最新的 YOLOv5 算法作为本题的人脸口罩检测算法的主体框架；在对训练集数据进行分析时，发现样本数据较小而且类别数据不均衡，于是对 YOLOv5 算法采用 MSFE 损失函数进行改进，尽量消除因为数据不平衡带来的检测精度下降问题；在模型训练阶段，对训练集样本进行 9:1 划分（9 份训练集，1 份验证集），在训练过程中结合检测框和类别损失函数等随迭代次数的变化曲线，最终确定 epoch=1000 作为最终迭代次数，模型训练完成之后，进行模型的测试。本模型的准确度 acc 值高达 96.1%，precision 达到 86.2%，召回率 97.1%，mAP 值达到 79.1% (IOU=0.45)。

对于问题三，需要利用问题二设计的算法，对测试集样本进行人脸口罩佩戴检测，将检测的原图片经过算法进行口罩佩戴情况标注后，输出出来，然后将检测到的每张图片包含的人脸数目、正确佩戴口罩数目、未佩戴口罩数目、未正确佩戴口罩数目输出并且填写到结果表格中。

关键词：YOLOv5；MSFE 损失；人脸口罩检测；IoU

一、问题重述

1.1 问题背景

新冠疫情的爆发对人类生命安全及全球经济发展造成了重大影响。虽然现在国内疫情基本得到有效遏制，但日常防控仍不可松懈。戴口罩是预防新冠肺炎最便捷、最有效的措施和方法。人脸佩戴口罩的自动化识别可以有效检测人群佩戴口罩情况，是抑制疾病在客流量大的公共场合快速传播和保护身体健康的重要手段。

1.2 问题重述

问题 1：通过分析题目要求得出，本题要解决的问题就是将标签文件（xml 格式）的信息，`<object>` 表示一张人脸，其中包含 `<name>` 表示人脸处于 “`with_mask`、`without_mask`、`mask_wearred_incorrect`” 三种状态的其中一种，`<bndbox>` 中记录了人脸口罩检测目标框的左上及右下角的坐标，我们就是要将这些信息在标签文件对应的原图片中对人脸进行框选，并且利用不同的颜色框将人脸状态标识出来，是对原图片进行标识的问题。

问题 2：通过分析和理解题目得出，本题需要我们设计一个人脸口罩识别的算法，并且设定 IoU 大于 0.45 时，才认为人脸口罩的佩戴情况被正确的分类，我们还需要对设计的人脸口罩识别算法进行识别效果衡量，量化的指标为 $\frac{\text{被正确分类的人脸的数量}}{\text{标签文件中包含的所有人脸数量}}$ ，也就是整体的 acc 值；该题是一个设计人脸口罩识别算法以及衡量算法效果的问题，我们需要陈述人脸口罩识别算法的设计过程及展示算法的识别性能。

问题 3：本题是利用问题 2 设计的人脸口罩识别算法，对 `test_images` 中的图片进行人脸口罩识别和检测，我们需要输出每一张图片中检测到的人脸的数量、正确佩戴口罩人的数量、未佩戴口罩人的数量、以及未正确佩戴口罩人的数量，并且将这些结果填写到相应的表格中。

二、模型假设与符号说明

2.1 模型假设

根据题目中所给信息以及要求，本文做出如下假设：

- (1) 假设训练样本的标签文件（.xml 格式）与训练集图片一一对应；
- (2) 假设训练样本的标签文件对人脸佩戴口罩类别的标注正确；
- (3) 假设训练样本的标前文件对人脸的框选位置合理；
- (4) 假设训练集标样本的标签文件不存在对图片人脸口罩检测漏标的情况。

2.2 符号说明

表 1 符号说明

序号	符号	含义
1	Acc	分类的准确度 (Accuracy)
2	P	类别的精确率 (Precision)
3	R	类别的召回率 (Recall)
4	F1	精确率和召回率的 F1-Score
5	PR	表示精确率和召回率曲线
6	mAP	均值平均精度

三、标签信息可视化

3.1 问题一的分析及思路

通过分析题目要求，问题一是一个将附件 1 训练样本中标签信息（人脸口罩佩戴类别、人脸检测框），显示到对应达到图片中的问题。

基于以上问题，我们首先读取标签文件（.xml）与对应的图片，然后根据数字图像处理的知识，通过 PIL 包中的 Image 的 Rectangle 函数在对应的图片中画出检测框，然后在根据口罩佩戴情况类别的不同显示不同的颜色框，并直观显示标签文件中“with_mask、without_mask、mask_wearred_incorrect”的分类状态，然后利用循环程序一次性标记出一张图片中的所有人脸标签信息。

3.2 数字图像处理相关知识

3.2.1 像素坐标系

在 OpenCV 中像素坐标系的坐标原点在图像的左上角，这种坐标系在结构体 Mat、Rect、Point 中都是适用的，在 Point (x,y) 和 Rect (x,y) 中，第一个参数 x 代表的是元素所在图像的列数，第二个参数 y 代表的是元素所在图像的行数。

像素坐标系^[1]如图 1 所示，以图像左上角为原点建立以像素为单位的坐标系 u-v。像素的横坐标 u 与纵坐标 v 分别是在其图像数组中所在的列数与所在行数。（在 OpenCV 中 u 对应 x，v 对应 y）；坐标体系的 x 轴为图像矩形中的那条水平线，从左往右为正方

向；y 轴为图像矩形中的那条垂直线，从上往下为正方向。

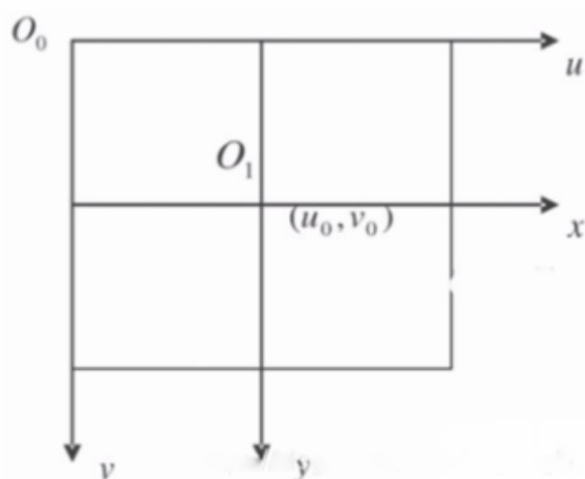


图 1 像素坐标系

如图 2 所示更好的展示了像素坐标系，点 (3, 4) 表示的是从原点起，向下数四行，向右数三列。相对位置都是以 0 为索引开始计算，即图像坐标系先按宽，再按高进行索引，即 (W,H)

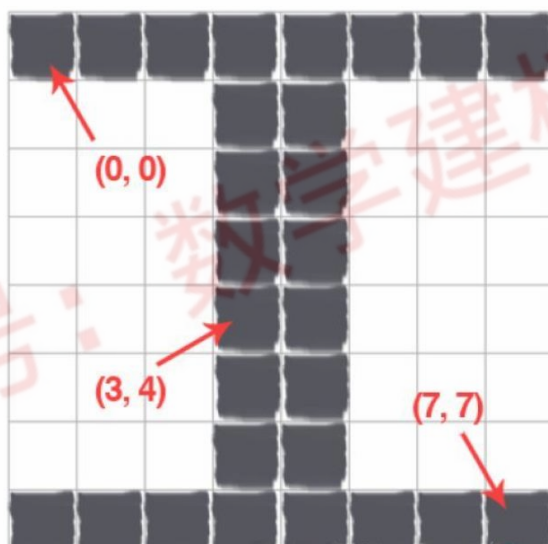


图 2 像素点的坐标

3.3 标签信息可视化流程图

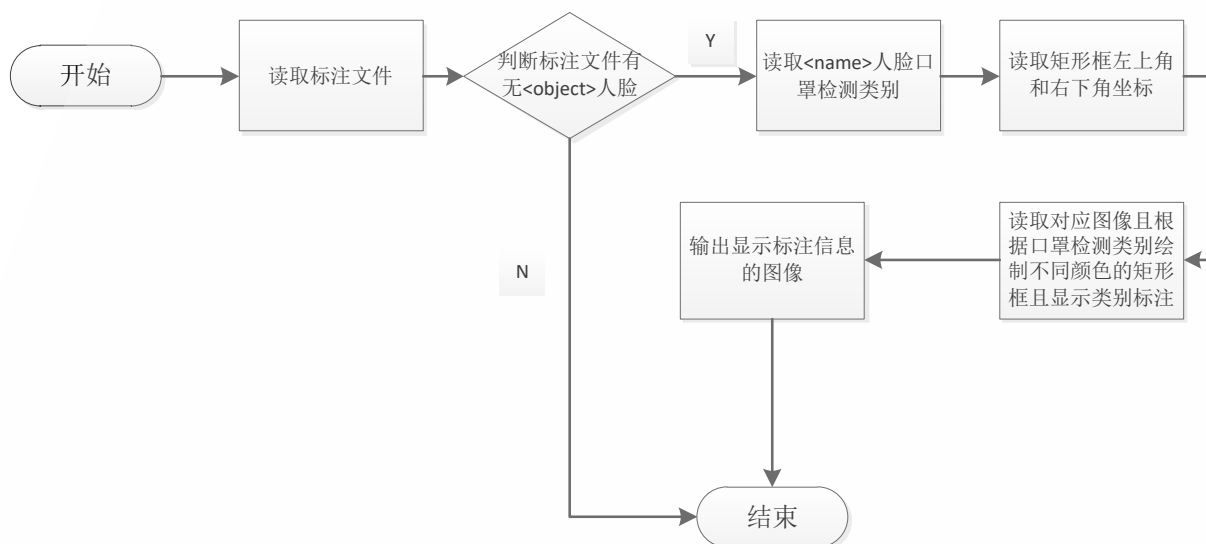


图 3 标签信息可视化流程

3.4 标签信息可视化结果

根据问题 1 的要求，我们重点展示 250.png 和 477.png 这两张图的标签信息可视化图片。图片 250.png 结合标签信息的可视化结果如图 4 所示。



图 4 图片 250.png 标签信息可视化结果
图片 477.png 的标签信息可视化图片如图 5 所示。



图 5 图 477.png 标签信息可视化结果
(该题代码可详见附录中，tagging.ipynb)

四、人脸口罩检测算法设计

4.1 问题二的分析及思路

通过分析问题二的要求，需要设计一个人脸口罩检测算法，首先第一步是检测人脸，定位人脸的位置，然后依据人脸佩戴口罩的状态，分为“戴口罩、未戴口罩、未正确佩戴口罩”三类。

人脸口罩检测首先是一个目标检测的问题，考虑到各类成熟目标检测算法的特点，以及本题的实际背景，我们不仅需要检测精度高而且需要做到实时检测，只有这样的模型才能符合本题背景的需求，于是最终选定基于深度学习的 YOLOv5 模型来解决这个问题。

4.2 训练样本数据分析及处理

对训练样本数据进行分析时，会发现“戴口罩、未戴口罩、未正确佩戴口罩”三类人脸的数据非常不平衡，三类训练标签数值如图 6 所示。当训练的样本不平衡时，类别间的样本例数目相差过多，会对模型的训练过程造成一定的影响，从而导致分类模型的准确度有所下降。

训练样本各类人脸数

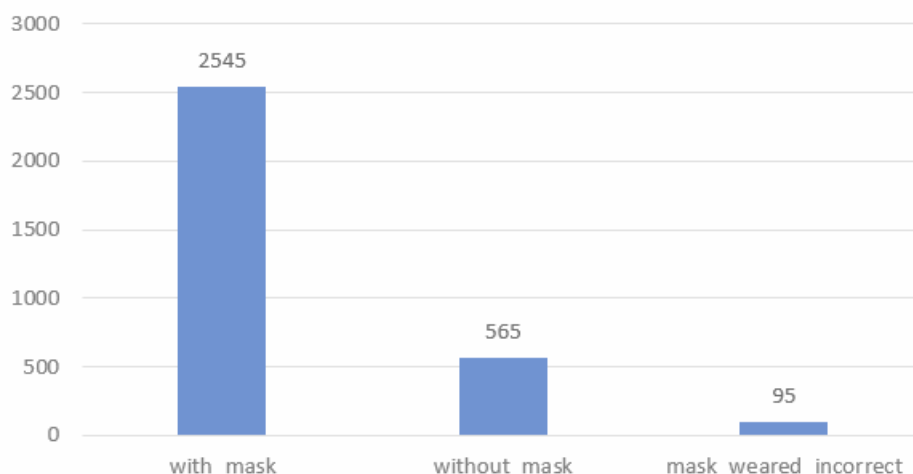


图 6 训练样本各类人脸数

由于一张图片中，包含多张人脸，而且未正确佩戴口罩的人脸图片很少，而且没有单纯未正确佩戴口罩的样本，所以我们没有考虑提升样本的方法，没有将分类样本少的数据集使用复制的方法进行扩充，我们从算法的角度考虑，通过对现有的 YOLOv5 算法的损失函数进行改进，消除类别间训练样本不平衡带来的分类影响^[2]。我们使用 mean squared false error (MSFE) loss^[3]，具体改进的损失函数公式如下所示。

$$FPE = \frac{1}{N} \sum_{i=1}^n \sum_n \frac{1}{2} (d_n^{(i)} - y_n^{(i)})^2$$

$$FNE = \frac{1}{P} \sum_{i=1}^P \sum_n \frac{1}{2} (d_n^{(i)} - y_n^{(i)})^2$$

$$MFE = FPE + FNE$$

$$MSFE = FPE^2 + FNE^2$$

MSFE loss 能够很好地平衡正反例之间的关系，从而实现更好的优化结果。

4.3 YOLOv5 网络模型介绍

YOLOv5 (You only look Once) 是 YOLO 系列的最新目标检测网络^[4]，其图像推理速度最快达 0.007s, 即每秒可处理 140 帧，满足视频图像实时检测的需求，同时结构更为小巧，YOLOv5 版本的权重数据文件为 YOLOv4 的九分之一，大小为 27M，YOLOv5 的网络模型结构如图 7 所示

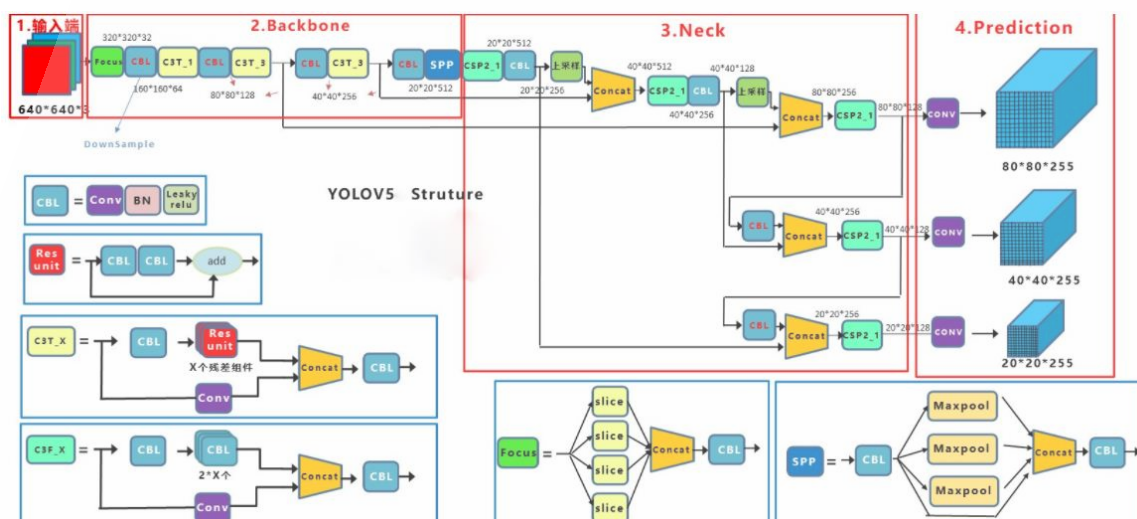


图 7 YOLOv5 的网络结构

YOLOv5 仍然是分为四个部分：

- (A) 输入端：Mosaic 数据增强、自适应锚框计算、自适应图片缩放
- (B) Backbone: Focus 结构，CSP 结构
- (C) Neck: FPN+PAN 结构
- (D) Prediction: GIoU_Loss

目前 YOLOv5 一共有 4 个版本，分别是 YOLOv5s、YOLOv5m、YOLOv5l、YOLOv5x。其中，YOLOv5s 网络是 YOLOv5 系列中深度最小，特征图的宽度最小的网络，后面三种都是在此基础上不断加深、不断加宽。

1) Mosaic 数据增强

YOLOv5 的输入端采用了和 YOLOv4 一样的 Mosaic 数据增强的方式，Mosaic 参考的 2019CutMix 数据增强的方式，但 CutMix 只使用了两张图片进行拼接，而 Mosaic 数据增强则采用了 4 张图片，随机缩放、随机裁剪、随机排布的方式进行拼接，如图 8 所示。



图 8 Mosaic 数据增强

采用了 Mosaic 数据增强的方式，其优点主要有：

- (A) 丰富数据集：随机使用 4 张图片，随机缩放，再随机分布进行拼接，大大丰富了检测数据集，特别是随机缩放增加了很多小目标，让网络的鲁棒性更好。
- (B) 减少 GPU：作者考虑到很多人可能只有一个 GPU，因此 Mosaic 增强训练时，可以直接计算 4 张图片的数据，使得 Mini-batch 大小并不需要很大，一个 GPU 就可以达到比较好的效果。

这种方式对小目标检测效果比较理想，符合本文人脸口罩检测需求。

2) 自适应锚框计算

在 YOLO 算法中，针对不同的数据集，都会有初始设定长宽的锚框。在网络训练中，网络在初始锚框的基础上输出预测框，进而和真实框 `groundtruth` 进行比对，计算两者差距，再反向更新，迭代网络参数。在 YOLOv3、YOLOv4 中，训练不同的数据集时，计算初始锚框的值是通过单独的程序运行的。

但 YOLOv5 中将此功能嵌入到代码中，每次训练时，自适应的计算不同训练集中的最佳锚框值。

3) 自适应图片缩放

在常用的目标检测算法中，不同的图片长宽都不相同，因此常用的方式是将原始图片统一缩放到一个标准尺寸，再送入检测网络中。但 YOLOv5 代码中对此进行了改进，也是 YOLOv5 推理速度能够很快的一个不错的 `trick`。

作者认为，在项目实际使用时，很多图片的长宽比不同，因此缩放填充后，两端的黑边大小都不同，而如果填充的比较多，则存在信息冗余，影响推理速度。

因此在 YOLOv5 的代码中 `datasets.py` 的 `letterbox` 函数中进行了修改，对原始图像自适应的添加最少的黑边。图像高度上两端的黑边变少了，在推理时，计算量也会减少，即目标检测速度会得到提升。

4) Focus 结构

Focus 是 YOLOv5 新增的操作，在 YOLOv3&YOLOv4 中并没有这个结构，以 YOLOv5s 的结构为例，原始 $608 * 608 * 3$ 的图像输入 Focus 结构，采用切片操作，先变成 $304 * 304 * 12$ 的特征图，再经过一次 32 个卷积核的卷积操作，最终变成 $304 * 304 * 32$ 的特征图，其结构如图 6 所示。

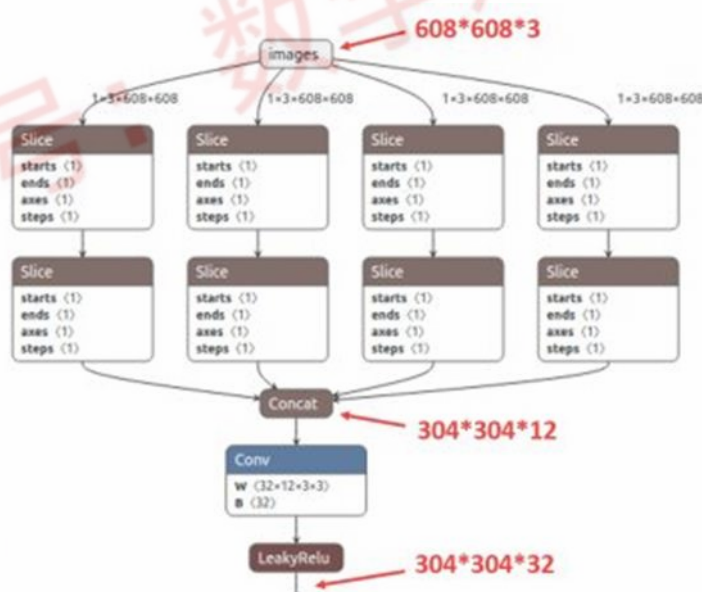


图 9 Focus 结构图

其中比较关键的是切片操作，如图 10 所示：

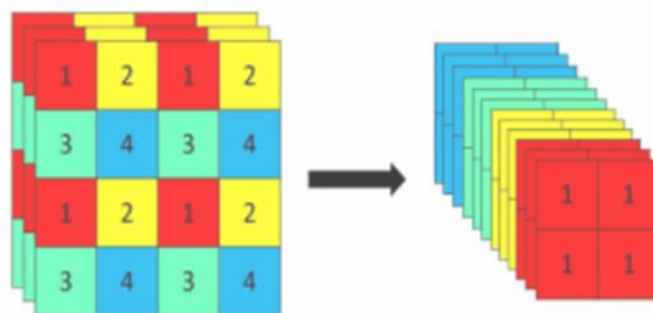


图 10 切片操作

4) CSP 结构

YOLOv4 网络结构中,借鉴了 CSPNet 的设计思路,在主干网络中设计了 CSP 结构。YOLOv5 与 YOLOv4 不同点在于,YOLOv4 中只有主干网络使用了 CSP 结构。而 YOLOv5 中设计了两种 CSP 结构,以 YOLOv5s 网络为例, CSP1_X 结构应用于 Backbone 主干网络,另一种 CSP2_X 结构则应用于 Neck 中。

5) Neck-路径聚合网络

YOLO V5 和 V4 都使用 PANET 作为 Neck 来聚合特征。Neck 主要用于生成特征金字塔,增强模型对于不同缩放尺度对象的检测,从而能够识别不同大小和尺度的同一个物体^[5]。在 PANET 之前,一直使用 FPN(特征金字塔)作为对象检测框架的特征聚合层, PANET 在借鉴 Mask R-CNN 和 FPN 框架的基础上,加强了信息传播。Neck 采用 FPN+PAN 结构如图 11 所示。

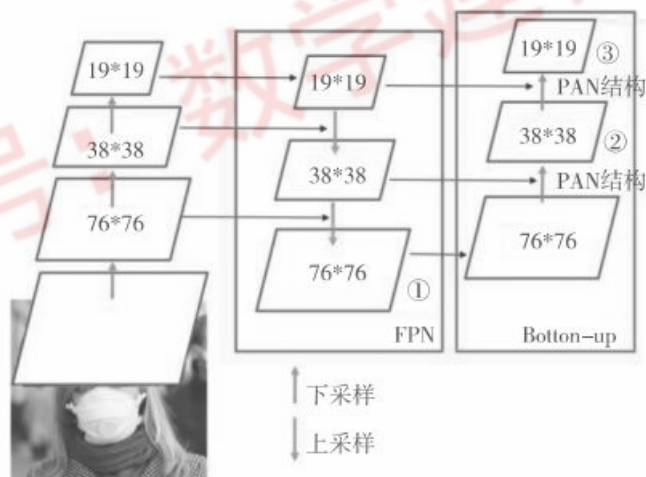


图 11 FPN+PAN 结构

PANET 基于 Mask R-CNN 和 FPN 框架,同时加强了信息传播。该网络的特征提取器采用了一种新的增强自下向上路径的 FPN 结构,改善了低层特征的传播。第三条通路的每个阶段都将前一阶段的特征映射作为输入,并用 3x3 卷积层处理它们。输出通过横向连接被添加到自上而下通路的同一阶段特征图中,这些特征图为下一阶段提供信息。同时使用自适应特征池化(Adaptive feature pooling)恢复每个候选区域和所有特征层次之间被破坏的信息路径,聚合每个特征层次上的每个候选区域,避免被任意分配。

6) Prediction

Prediction 包括 Bounding box 损失函数和非极大值抑制(NMS)^[6]。YOLOv5 中使用 GIOUS_Loss 作为损失函数,有效解决了边界框不重合时问题。在目标检测预测结果处理阶段,针对出现的众多目标框的筛选,采用加权 NMS 操作,获得最优目标框。

4.4 人脸口罩检测模型评估指标介绍

(1) 混淆矩阵

混淆矩阵^[7]也称误差矩阵，是表示精度评价的一种标准格式，用 n 行 n 列的矩阵形式来表示。具体评价指标有总体精度、制图精度、用户精度等，这些精度指标从不同的侧面反映了图像分类的精度。在人工智能中，混淆矩阵（confusion matrix）是可视化工具，特别用于监督学习，在无监督学习一般叫做匹配矩阵。在图像精度评价中，主要用于比较分类结果和实际测得值，可以把分类结果的精度显示在一个混淆矩阵里面。混淆矩阵是通过将每个实测像元的位置和分类与分类图像中的相应位置和分类相比较计算的。

混淆矩阵中的横轴是模型预测的类别数量统计，纵轴是数据真实标签的数量统计。对角线表示模型预测和数据标签一致的数目，所以对角线之和除以测试集总数就是准确率。对角线上数字越大越好，在可视化结果中颜色越深，说明模型在该类的预测准确率越高。如果按行来看，每行不在对角线位置的就是错误预测的类别。总的来说，我们希望对角线越高越好，非对角线越低越好。

以二分类模型为例，我们的模型最终需要判断样本的结果是 0 还是 1，或者说是 positive 还是 negative。

我们通过样本的采集，能够直接知道真实情况下，哪些数据结果是 positive，哪些结果是 negative。同时，我们通过用样本数据跑出分类型模型的结果，也可以知道模型认为这些数据哪些是 positive，哪些是 negative。

因此，我们就能得到这样四个基础指标：

真实值是 positive，模型认为是 positive 的数量（True Positive=TP）

真实值是 positive，模型认为是 negative 的数量（False Negative=FN）：这就是统计学上的第二类错误（Type II Error）

真实值是 negative，模型认为是 positive 的数量（False Positive=FP）：这就是统计学上的第一类错误（Type I Error）

真实值是 negative，模型认为是 negative 的数量（True Negative=TN）

二分类问题的混淆矩阵如表 2 所示：

表 2 混淆矩阵

混淆矩阵		真实值	
		Positive	Negative
预测值	Positive	TP	FP
	Negative	FN	TN

(2) 准确率（Accuracy）

分类模型所有判断正确的结果占总观测值的比重，公式如下：

$$Precision = \frac{TP + TN}{TP + TN + FP + FN}$$

(3) 精确率（Precision）

在模型中是 Positive 的所有结果中，模型预测的比重，公式如下：

$$Precision = \frac{TP}{TP + FP}$$

（4）召回率（Recall）

Recall 是测试集中所有正样本样例中，被正确识别为正样本的比例，公式如下：

$$Recall = \frac{TP}{TP + FN}$$

（5）F1-Score

F1 指标是衡量精确率和找回率的指标，计算公式如下：

$$F1 = \frac{2PR}{P + R}$$

（6）平均精度（AP）与 mean Average Precision（mAP）

AP 就是 Precision-Recall 曲线下方的面积，通常来说一个越好的分类器，AP 值越高。

mAP 是多个类别 AP 的平均值。这个 mean 的意思是对每个类的 AP 再求平均，得到的就是 mAP 的值，mAP 的大小一定在[0,1]区间，越大越好。该指标是目标检测算法中最重要的一个。

4.5 人脸口罩检测模型建立

通过附件 1 中的训练样本进行 9: 1（9 份作为训练集，1 份作为验证集）数据集划分之后，开始训练 YOLOv5 模型，得到最适合人脸口罩检测的 YOLOv5 网络权重数据。然后输入一张新的图片到模型中，输出模型的测试结果，检测流程图如图 12 所示：

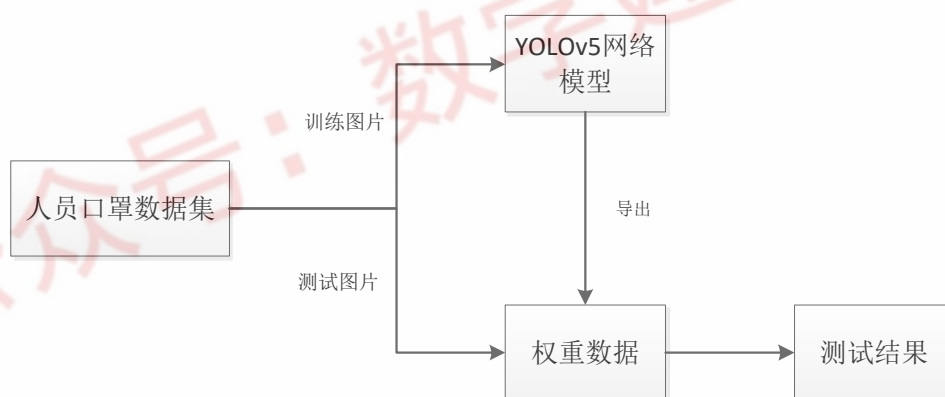


图 12 检测流程

在模型的训练过程中，出现了训练集中一些图片无法训练的问题，原因是训练集中的一些图片格式不是标准的 PNG 格式，原始图片是如 jpg 格式等，错误的采用了 png 为后缀导致的错误，而新版本的 libpng 对关于 iCCP 采用了更严苛的约束。我们删除图片内嵌 profile，可先进入 Image > Mode > Assign Color Profile 并设置为 RGB workspace(sRGB built-in)，然后 File > Overwrite add_green.png 覆盖原来的 png 文件。至此每张训练集中的图片都能得到训练。

根据问题二的要求我们设定，仅当 IoU 大于 0.45 时，分类器才对人脸口罩检测类别正确分类。

损失函数是一个用来衡量模型的预测值 $f(x)$ 与真实值 Y 的不一致程度，它是一个非负值函数，我们认为损失函数越小，模型的鲁棒性就越好。因此在模型的训练过程中，根据模型训练的 epoch，我们输出框的损失函数如图 13 所示、人脸口罩类别的损失函数如图 14 所示，以及检测到人脸的损失函数如图 15 所示。

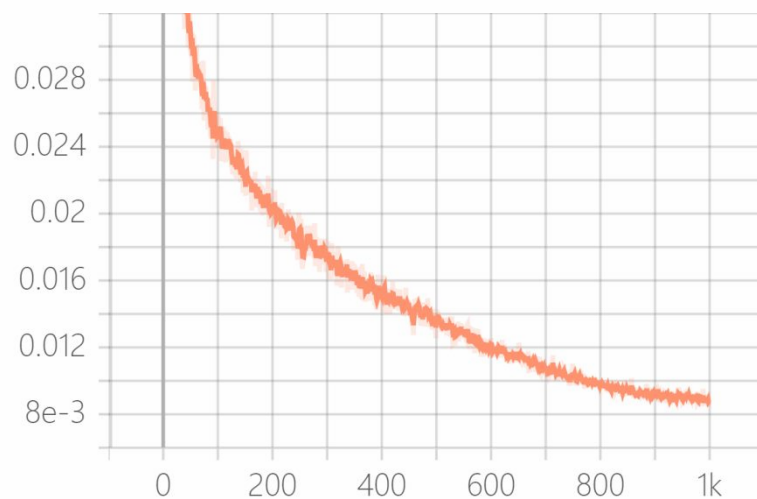


图 13 训练集框的损失函数

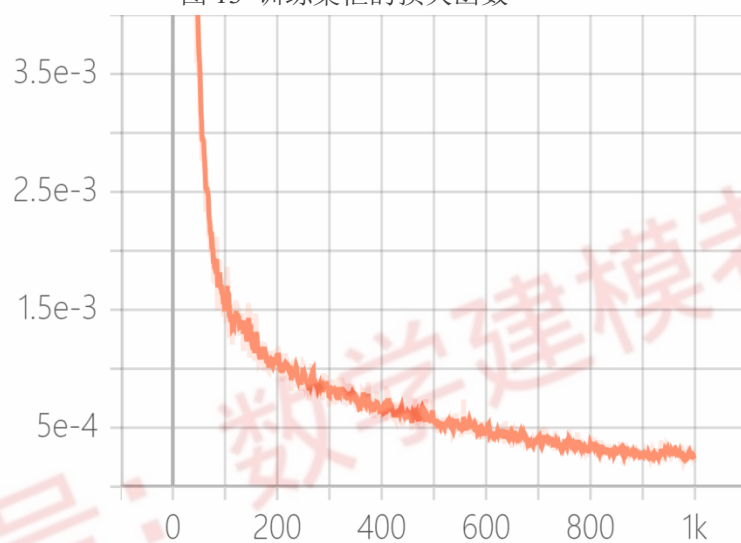


图 14 训练集人脸口罩检测类别损失函数

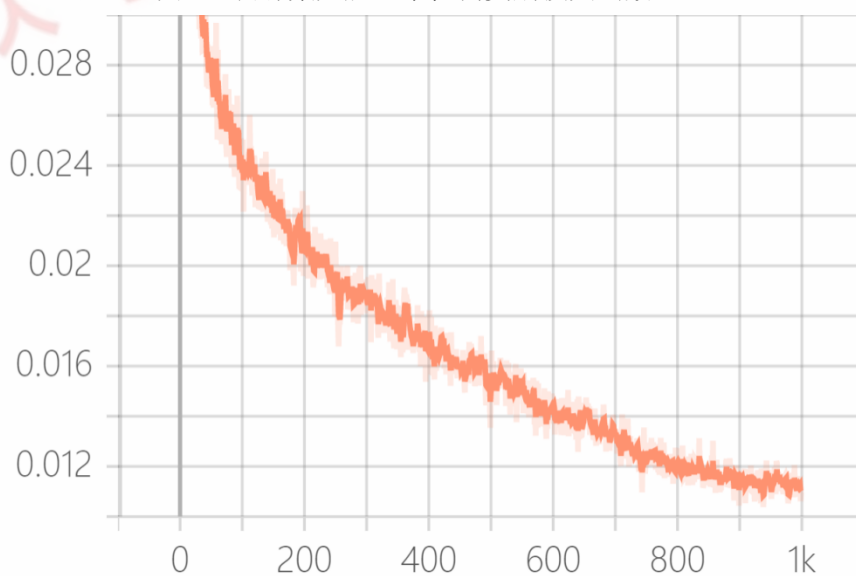


图 15 训练集检测人脸的损失函数

从上面的图片中我们可知，随着模型迭代次数的增加，框、类别以及人脸检测的损失函数逐渐减小，且迭代到 1000 代左右模型损失函数基本收敛，已经有非常好的检测

效果了。结合精确率、召回率、mAP 等随迭代次数增长的曲线如图 16 所示，我们最终选定 epoch=1000 作为模型的迭代次数。

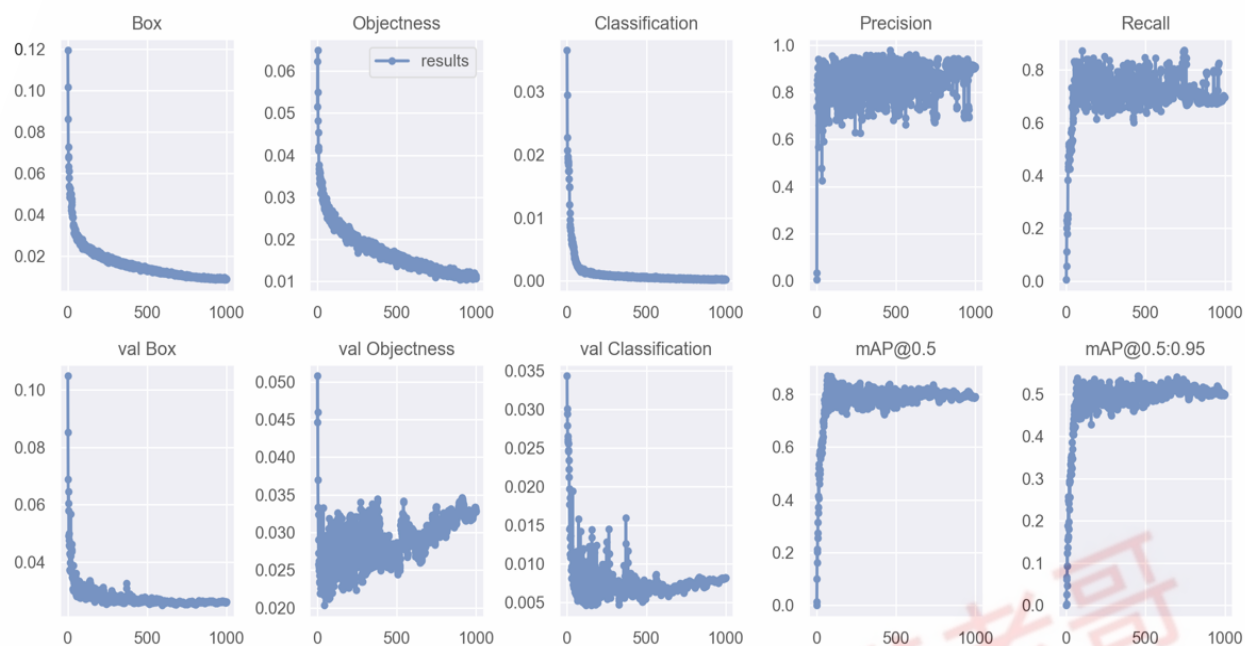


图 16 随迭代次数增加的各类指标

4.6 改进 YOLOv5 模型人脸口罩检测结果

运用我们训练好的模型，我们选择测试集中的 101.png 图片，输入到模型中进行人脸口罩检测，输出的结果如图 17 所示；在原图的基础上框选出了人脸，而且给出了对应的标签以及检测的置信度。

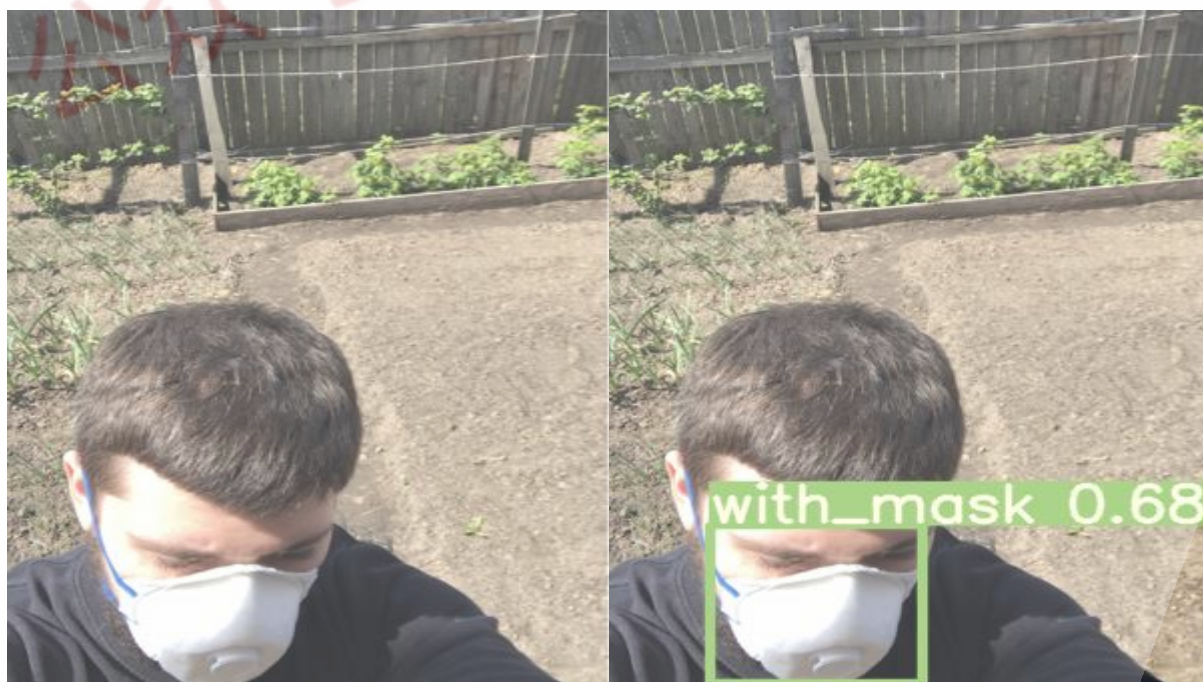


图 17 人脸口罩检测结果

4.6 人脸口罩检测模型评估

在模型的训练阶段对训练样本进行 9:1 划分, 1 份用于验证集 66 张图片, 我们统计其中的人脸佩戴口罩各类别的数量, 结果如表 3 所示

表 3 验证集标签文件各类人脸的数量

类别	人脸数量
with_mask	230
without_mask	62
mask_wearred_incorrect	15

基于已经训练好的改进 YOLOv5 的人脸口罩检测模型, 我们需要对模型的检测效果进行评估。混淆矩阵是一种常用的分类问题模型的评价指标, 是准确度 (Accuracy)、精确度 (Precision)、召回率 (Recall) 等评价指标的基础。于是我们通程序计算出在训练样本的验证集中的混淆矩阵, 结果如表 4 所示。

表 4 人脸口罩检测混淆矩阵

混淆矩阵		真实值		
		with_mask	without_mask	mask_wearred_incorrect
预测值	with_mask	221	1	0
	without_mask	2	59	0
	mask_wearred_incorrect	7	2	15

根据上图混淆矩阵的结果, 可知 with_mask、without_mask、mask_wearred_incorrect 三分类中, 混淆矩阵对角线上的数值很大, 对角线之外的值很小, 说明我们的模型的检测效果很好。

根据问题 2 的要求, 我们首先进行准确度 ($Acc = \frac{\text{被正确分类人脸的数量}}{\text{标签文件中所有人脸的数量}}$) 的评估。针对于标签文件中每一类别人脸检测的数量结果如表 4 所示。

结合混淆矩阵的结果, 我们通过公式计算出 Acc 指标:

$$Acc = \frac{221+59+15}{230+62+15} = 0.961$$

所以针对于问题二的量化指标准确度 (Acc), 我们的模型在样本数据集比较小的情况下, 对于人脸口罩检测的问题效果很好, 准确率高达 0.961, 足以见得我们设计的改进 YOLOv5 的人脸口罩检测模型十分优异。

在目标检测领域对于模型的评价, 通常认为只评价准确度是不够充分的, 还需要精确率 (P), 召回率 (R)、F1-Score 指标以及 PR、mAP 曲线的综合评价。对此我们接

着对模型进行了更加充分检验。

人脸口罩检测各个类别的精确率如表 5 所示：

表 5 各类别精确率

类别	精确率 (P)
with_mask	0.995
without_mask	0.967
mask_wearred_incorrect	0.625

通过精确率的计算，得知预测正确占预测总数的比例，精确率越高说明真实为某一类被正确预测效果越好，从表 5 可知戴口罩和未佩戴口罩的预测效果很好，由于未正确佩戴口罩验证集数据太少容易出现偶然性，导致未正确佩戴口罩的精确率不高，但精确率对于模型的衡量需要结合召回率才更有说服力，避免极端情况，精确率很高但是召回率很低，模型的效果也理想，所以我们得出各类的召回率 (R) 如表 6 所示。

表 6 各类别召回率

类别	召回率 (R)
with_mask	0.961
without_mask	0.952
mask_wearred_incorrect	1

召回率能够说明模型中预测正确占该类总数的比例，召回率越大说明在某一类的预测中真实类被正确分类的概率越大即很少将真实某一类错分，从表中可知模型预测器对未正确佩戴口罩的效果极好。

由于精确率和召回率是一对矛盾的评价指标，不可能做到两者都好，因此我们需要平衡精确率和召回率指标，我们计算模型的 F1-Score 指标，结果如表 7 所示。

表 7 各类 F1 得分

类别	F1-Score
with_mask	0.978
without_mask	0.959
mask_wearred_incorrect	0.769

通过对 F1-Score 指标的分析，我们得知对于戴口罩以及未戴口罩的得分很高说明模型对于这两个类的检测效果很好。

最后我们进行目标检测领域，最重要的性能评价指标分析，如图 18 所示我们对戴口罩、未戴口罩、以及未正确佩戴口罩三类进行了 PR 曲线绘制，以及所有类的 mAP 曲线绘制。

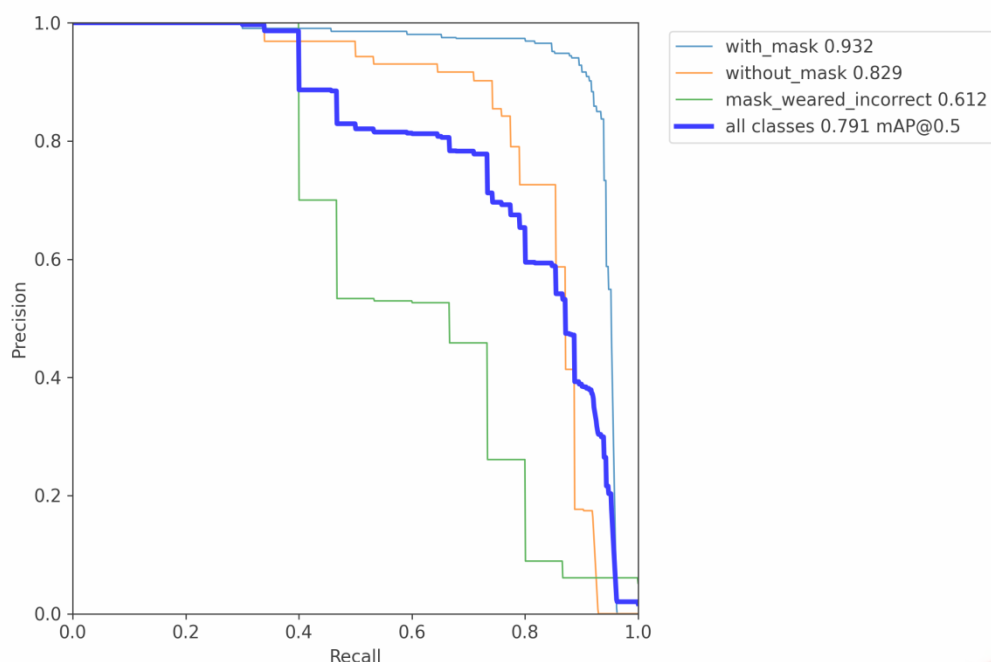


图 18 各类的 PR 曲线以及 mAP 曲线

从 PR 曲线以及 mAP 曲线的走势分析，戴口罩和未戴口罩的分类效果很好，未正确佩戴口罩的分类效果相较前两种略微逊色。mAP 曲线的下方的面积很大，总体说明模型的效果很好。

经过训练后的模型性能和检测的精度都较高，其模型 ACC 值为 96.1%，precision 值 86.2%，召回率 97.1%，F1-score=0.902，mAP(IOU=0.45)为 79.1%，总体来说性能十分优异。(模型代码见项目源文件中 model 文件夹)。

五、人脸口罩检测

5.1 问题三的分析及思路

针对问题三，我们需要利用问题二得到的人脸口罩检测模型对测试集的图片进行人脸口罩检测，查看“问题 B 提交结果.xlsx”文件，我们需要统计模型检测到每张图片中人脸的数目、正确佩戴数量、未佩戴数量、未正确佩戴数量；所以我们只需要将测试集中的 200 张图片输入到我们训练的模型中，然后输出以上数据，并填入相应的表格中即完成本题的任务要求。

完成问题三的操作流程图如图 19 所示

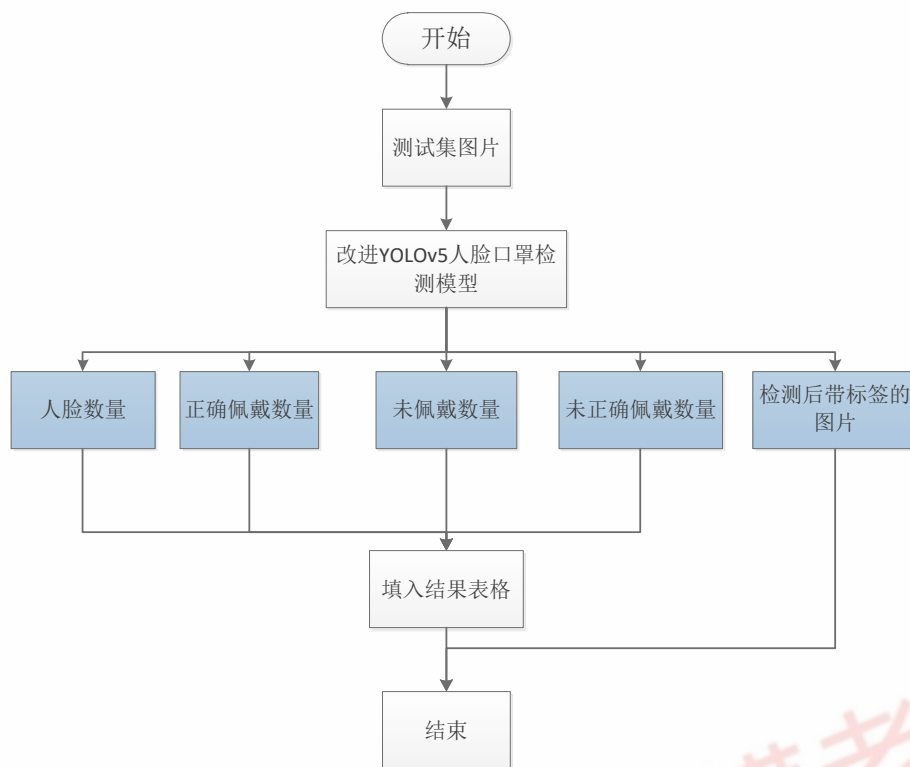


图 19 测试集检测流程图

5.2 改进 YOLOv5 人脸口罩检测结果

使用我们改进的 YOLOv5 模型，将测试集的图片输入到模型中去，输出人脸位置框选的图片 and 佩戴口罩的情况，输出的检测效果图放在支撑材料中 detect 文件夹中，部分检测结果如图 20 所示。



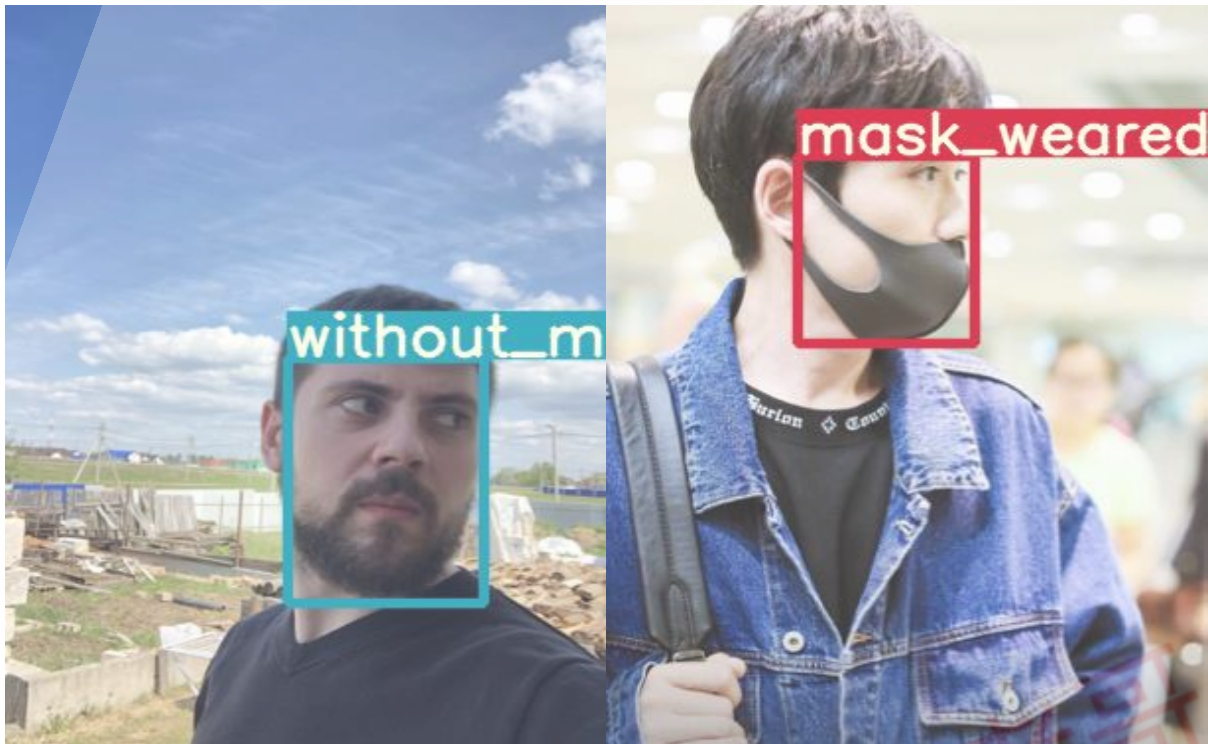


图 20 从左到右，从上到下分别为 0.png、6.png、14.png、132.png 检测结果

从上面的检测结果图可知人脸的位置被框选，而且方框的上方标注佩戴口罩的类别，以及属于该类别的置信度，检测人脸位置框选完整，而且类别标签正确。该模型的检测效果十分优异。

模型也会输出每张图片中人脸的数目、正确佩戴数量、未佩戴数量、未正确佩戴数量，对于 0.png 与 7.png 图片填入结果表格如表 8 所示。

表 8 检测结果表格

图片名称	人脸数量	正确佩戴数量	未佩戴数量	未正确佩戴数量
0.png	3	1	2	0
6.png	1	1	0	0
14.png	1	0	1	0
132.png	1	0	0	1

六、模型优缺点分析

6.1 模型的优点

1.本文在 YOLO 系列最新算法 YOLOv5 的基础上，对其损失函数做出改进，利用 MSFE 损失函数代替原来 YOLOv5 的损失函数，在一定程度上改善了本题训练集数据不平衡的问题，使得对数据量小的类别样本提高了预测的准确度，使得模型的鲁棒性增强。

2.利用准确度、精确率等评价指标对模型进行分析，本模型对人脸口罩检测的效果十分的优异，其总准确度高达 0.961，而且模型的检测速度非常快，吸收了 YOLO 系列算法检测速度快的优点，所以本模型是一个能实现高精度人脸口罩检测的实时检测模型，对于新冠疫情下运用于现实场景的人脸口罩检测，无论是在工业上还是个人使用上，都具有很高的利用价值。

6.2 模型的缺点

由于深度学习模型需要大量的训练数据作为训练集，但是本题提供的训练数据集比较小，而且人脸口罩检测类别的数据不平衡，所以削弱了我们的模型对小数据样本类的预测能力。

6.3 模型的改进和推广

1.本模型对于遮挡目标和极小目标的检测能力不足，存在缺检和漏检的情况，所以接下来可以在模型中增加注意力机制，提高模型的检测能力，以及对于工业部署，YOLOv5 的模型还是比较大，所以可以改善其主干网络，运用更轻量级的卷积神经网络主干进行优化，方便人脸口罩检测模型的工业化部署。

2.本模型不仅针对人脸口罩检测，而且还可以推广运用到其他目标检测的场景中去，提高社会的智能化层度，便利人们的生活。

七、参考文献

- [1] wwwlyj123321 , opencv 中的像素坐标系与操作像素的方法 , https://blog.csdn.net/wwwlyj123321/article/details/86770662?utm_term=%E5%83%8F%E7%B4%A0%E5%9D%90%E6%A0%87%E7%B3%BB&utm_medium=distribute.pc_aggpage_search_result.none-task-blog-2~all~sobaiduweb~default-0-86770662&spm=3001.4430 , 2021.10.30.
- [2] 专注于计算机视觉的 AndyJiang , 深度学习之如何处理样本不平衡 , <https://blog.csdn.net/andyjkt/article/details/107081858> , 2021.10.30
- [3] Wang S , Wei L , Jia W , et al. Training deep neural networks on imbalanced data sets[C]// International Joint Conference on Neural Networks. IEEE, 2016.
- [4] 江大白* , 深入浅出 Yolo 系列之 YOLOv5 核心基础知识完整讲解 , https://blog.csdn.net/nan355655600/article/details/107852353?ops_request_misc=%257B%2522request%255Fid%2522%253A%2522163565705916780265467191%2522%252C%2522scm%2522%253A%252220140713.130102334..%2522%257D&request_id=163565705916780265467191&biz_id=0&utm_medium=distribute.pc_search_result.none-task-blog-2~all~top_positive~default-1-107852353.pc_search_es_clickV2&utm_term=YOLOv5&spm=1018.2226.3001.4187 , 2021.10.30
- [5] libo-coder , 『 目标检测 』 YOLO V5(1) : 学习 笔记 , https://blog.csdn.net/libo1004/article/details/110928070?ops_request_misc=&request_id=&biz_id=102&utm_term=Neck%E4%B8%BB%E8%A6%81%E7%94%A8%E4%BA%8E%E7%94%9F%E6%88%90%E7%89%B9%E5%BE%81%E9%87%91%E5%AD%97%E5%A1%94%E5%EF%BC%8C%E5%A2%9E%E5%BC%BA%E6%A8%A1%E5%9E%8B%E5%AF%B9%E4%BA%8E%E4%B8%8D%E5%90%8C%E7%BC%A9%E6%94%BE%E5%B0%BA%E5%BA%A6%E5%AF%B9%E8%B1%A1&utm_medium=distribute.pc_search_result.none-task-blog-2~all~sobaiduweb~default-0-110928070.pc_search_es_clickV2&spm=1018.2226.3001.4187 , 2021.10.30
- [6] 江大白* , 深入浅出 Yolo 系列之 YOLOv3&YOLOv4&YOLOv5&YOLOx 核心基础知识完整讲解 , https://blog.csdn.net/nan355655600/article/details/106246625#2_YOLOv3_20 , 2021.10.30
- [7] 梅森上校 , [机器学习笔记] 混淆矩阵 (Confusion Matrix) , https://blog.csdn.net/seagal890/article/details/105059498?ops_request_misc=%257B%2522request%255Fid%2522%253A%2522163565787916780269815216%2522%252C%2522scm%2522%253A%252220140713.130102334..%2522%257D&request_id=163565787916780269815216&biz_id=0&utm_medium=distribute.pc_search_result.none-task-blog-2~all~top_click~default-2-105059498.pc_search_es_clickV2&utm_term=%E6%B7%B7%E6%B7%86%E7%9F%A9%E9%98%B5&spm=1018.2226.3001.4187 , 2021.10.30