# 基于 YOLOv5 的口罩佩戴检测系统设计与实现

**20121802 严昕宇**

**摘　要：**面对当前疫情防控的实际需求，自动化检测可以减少管理人员与他人接触感染的风险且能使疫情防控管理更加高效。针对人脸佩戴口罩识别问题，使用 YOLOv5 目标检测算法训练网络模型，实现对佩戴口罩和未佩戴口罩的检测；通过 PyQt5 设计并实现了可视化检测系统，提高了检测效率。

**关键词：**深度学习；YOLOv5；口罩佩戴检测

## 1. 项目背景和意义

2019 年底，新冠疫情开始在全球范围蔓延。而口罩能有效避免人们呼吸时吸入空气中的新冠病毒，从而降低被病毒感染的风险，因此规范佩戴口罩是防控病毒传播的重要手段。但随着全国疫情防控进入常态化阶段，人们也间歇性放松自我防护，经常会在公共场所不佩戴口罩，这一行为增加了感染病毒的风险。同时，国内正面临新一轮疫情冲击，为疫情防控增添了更多的不确定风险。

目前在医院等关键场合对人员是否佩戴口罩都是人工进行检测，但是人工检测存在工作强度大、效率低、覆盖面窄、时效性差等问题。针对此问题，本文基于 YOLOv5s 目标检测算法实现了口罩佩戴检测系统。此系统可以提高检测效率，并在一定程度上规避上述弊端。

## 2. 目标检测算法 YOLO

### 2.1 YOLO 概述

近年来，目标检测任务成为大众焦点，它实现了在不同的场景中标记出需要检测的目标并且确定它们的位置和类别。21 世纪初，目标检测在卷积神经网络的大量应用中得到了快速发展。在本系统中口罩佩戴检测本质是在人脸的基础上对是否佩戴口罩进行识别，因此需要使用目标检测类的神经网络[1]。现阶段基于深度学习目标检测算法主要分为两大类：一类是以 R-CNN 为代表的双阶段检测算法，另一类是以 YOLO 为代表的单阶段检测算法。双阶段检测算法的优势主要体现在可扩展性和高准确率性方面，而单阶段检测算法的主要优势是识别速度快，更适合用于需要实时检测的场景。

YOLO 算法是 2015 年 Redmon 等人提出的一种使用神经网络提供实时对象检测的算法。它将单个卷积神经网络应用于整个图像，将图像分成网格，并预测每个网格的类概率和边界框。该算法因其速度快和准确性高而广受欢迎，已广泛应用于检测交通信号、人员、停车计时器和动物中。

对 YOLO 系列算法、SSD 算法[2]和 R-CNN 系列算法在 VOC2007、VOC2012 和 COCO 数据集上的测试结果进行对比，如表 2.1 所示。其中 mAP 指平均精度均值，是横量模型效果的综合指标，“-”表示没有相关数据。相较于其他网络模型，YOLOv5 在检测速度与精度上均有一定的优势。

**表 2.1 算法性能对比**

| 算法 | 骨干网络 | 检测速度(FPS) | mAP/% | VOC2007 | VOC2012 | COCO |
|---|---|---|---|---|---|---|
| YOLO | VGG-16 | 45.0 | | 63.4 | 57.9 | - |
| YOLOv4 | CSPDarknet-53 | 23.0 | | - | - | 43.5 |
| YOLOv5 | Modified CSP v5 | 62.5 | | - | - | 68.8 |
| SSD | VGG-16 | 19.3 | | 79.8 | 78.5 | 28.8 |
| R-CNN | VGG-16 | 0.5 | | 66.0 | - | - |
| Faster R-CNN | ResNet-101 | 5 | | 76.4 | 73.8 | 39.8 |

### 2.2 YOLOv5

YOLOv5 模型主要由 Backbone、Neck 和 Head 三部分组成，网络模型见图。其中，Backbone 主要负责对输入图像进行特征提取，Neck 负责对特征图进行多尺度特征融合，并把这些特征传递给预测层，Head 进行最终的回归预测[3]。

Backbone 骨干网络是指用来提取图像特征的网络，它的主要作用是将原始的输入图像转化为多层特征图，以便后续的目标检测任务使用。在 Yolov5 中，使用的是 CSPDarknet53 或 ResNet 骨干网络，这两个网络都是相对轻量级的，能够在保证较高检测精度的同时，尽可能地减少计算量和内存占用。Backbone 中的主要结构有 Conv 模块、C3 模块、SPPF 模块。

由于物体在图像中的大小和位置是不确定的，因此需要一种机制来处理不同尺度和大小的目标。特征金字塔是一种用于处理多尺度目标检测的技术，它可以通过在骨干网络上添加不同尺度的特征层来实现。在 YOLOv5 中，使用了 PANet 作为 Neck 模块。其通过自顶向下部分和自下向上部分的特征图进行融合，得到最终的特征图，用于目标检测

Head 目标检测头是用来对特征金字塔进行目标检测的部分，它包括了一些卷积层、池化层和全连接层等。在 YOLOv5 模型中，检测头模块主要负责对骨干网络提取的特征图进行多尺度目标检测。该模块主要包括三个部分，此外，YOLOv5 还使用了一些技巧来进一步提升检测精度，比如 GIoU loss、Mish 激活函数和多尺度训练等。

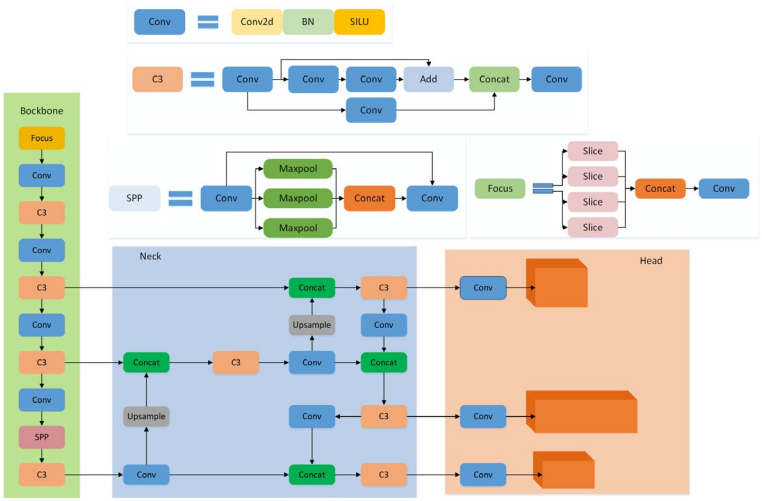



**图 2.1 YOLOv5 网络模型**

## 3. 模型训练

### 3.1 实验环境及数据集

本实验使用 Ubuntu 操作系统，基于 PyTorch 框架，使用 4 块 RTX 2080 Ti 显卡进行训练。

```
SoundLab                        Tue May 16 00:17:28 2023  470.182.03
[0] NVIDIA GeForce RTX 2080 Ti | 20°C,   0 % |   172 / 11019 MB | gdm(18M) gdm(17M) lfy(99M) lfy(16M) lfy(9M)
[1] NVIDIA GeForce RTX 2080 Ti | 22°C,   0 % |  6366 / 11019 MB | jj(6351M) gdm(4M) lfy(4M)
[2] NVIDIA GeForce RTX 2080 Ti | 22°C,   0 % |  7116 / 11019 MB | jj(7101M) gdm(4M) lfy(4M)
[3] NVIDIA GeForce RTX 2080 Ti | 22°C,   0 % |  7116 / 11019 MB | jj(7101M) gdm(4M) lfy(4M)
[4] NVIDIA GeForce RTX 2080 Ti | 23°C,   0 % |  7116 / 11019 MB | jj(7101M) gdm(4M) lfy(4M)
[5] NVIDIA GeForce RTX 2080 Ti | 22°C,   0 % |    10 / 11019 MB | gdm(4M) lfy(4M)
[6] NVIDIA GeForce RTX 2080 Ti | 21°C,   0 % |    10 / 11019 MB | gdm(4M) lfy(4M)
[7] NVIDIA GeForce RTX 2080 Ti | 22°C,   0 % |    10 / 11019 MB | gdm(4M) lfy(4M)
```

**图 3.1 实验环境**

本次数据集从 WIDER Face、MAFA 和 VOC 等公开数据集和 CSDN 等网络资源中共获取了 1600 张图片，包括人员佩戴口罩和未佩戴口罩两种情况。将采集的数据按照训练数据与测试数据之比为 3:1 进行分组，其中训练数据图片有 1200 张，测试数据图片有 400 张。

### 3.2 实验环境

PyTorch 是 Facebook 人工智能研究院(FAIR)团队开发的一个开源的深度学习框架，是目前主流的深度学习框架之一。因此，本系统所用目标检测模型以 PyTorch 为训练框架，模型训练基本流程如图 5 所示。具体如下：(1)将输入模型的图片调整为 640×640 大小，并将图片进行旋转、平移等数据增强操作。(2)将图像数据输入到 YOLOv5s 网络模型中(每个神经网络对输入数据进行加权累加再输入到激活函数作为该神经元的输出值)通过前向传播，得到得分值。(3)将得分值输入到损失函数中，与真实值相差较大再计算损失值，通过损失值判断识别程度的好坏。(4)通过反向传播(反向求导，损失函数和网络模型中的每个激活函数都要求，最终目的是使误差最小)来确定梯度向量。(5)最后通过梯度向量来调整每一个权值，使误差趋于 0 或者模型趋于最优解。(6)重复上述过程直到设定的次数。

模型分别选用 YOLOv5 中的 YOLOv5s、YOLOv5m、YOLOv5l 模型，图片尺度统一缩放为 640×640 的图片，批处理通道数(batch_size)设置为 64，训练代数(epoch)为 100。

### 3.3 模型训练与评估

通过分析训练数据，可以发现在前 20 个 epoch 训练中精度提升明显。cls_loss、box_loss 和 obj_loss 是在三个方面衡量模型训练效果的损失函数，其中 cls_loss 表示置信度的损失函数，box_loss 表示预测框位置的损失函数，obj_loss 表示检测目标的损失函数[4]。由于口罩检测数据集的数据规模不大，且模型较小，在 70 个 epoch 后损失函数“box”，“obj”，“cls”已经开始收敛，在 80 Epoch 后精度已经稳定在了较高水准，此时的模型在验证集上测试表现已较好。

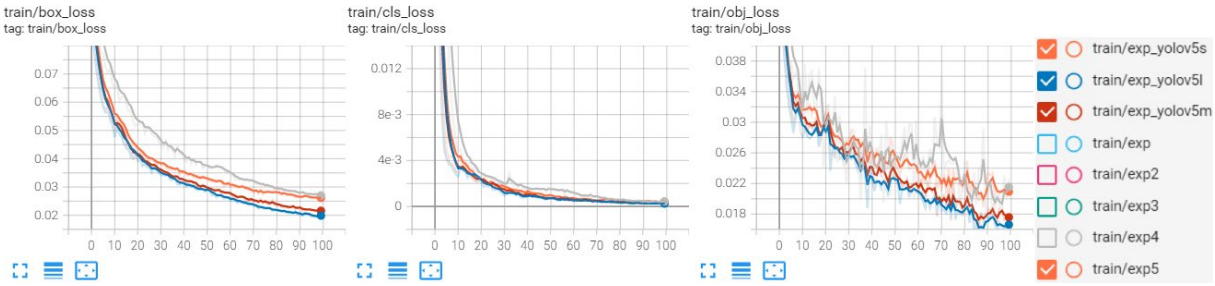



**图 3.2 Loss 损失函数**

训练好一个模型后，需要去评价这个模型的好坏，常用到的两个指标分别是召回率 Recall 和精确度 Precision。

召回率(Recall)，也被称为查全率，或者 True Positive Rate，反映了所有真正为正例的样本中被分类器判定出来为正例的比例。其计算公式如下，其中 *TP* 表示被模型判断为正类的正样本数，*FN* 表示被模型判断为负类的正样数[5]：

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

精确率(Precision)，或者叫做精度，反映了被分类器判定的正例中真正的正例样本的比例，其计算公式如下，其中 *FP* 表示被模型判断为正类的负样本数：

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

两个指标都是简单地从一个角度来判断模型的好坏，均是介于 0 到 1 之间的数值，其中接近于 1 表示模型的性能越好，接近于 0 表示模型的性能越差。

为了综合评价目标检测的性能，还需要采用 PR 曲线、F1 或均值平均密度 mAP 等参数来进一步评估模型，此处以模型的 mAP 为例进行分析。

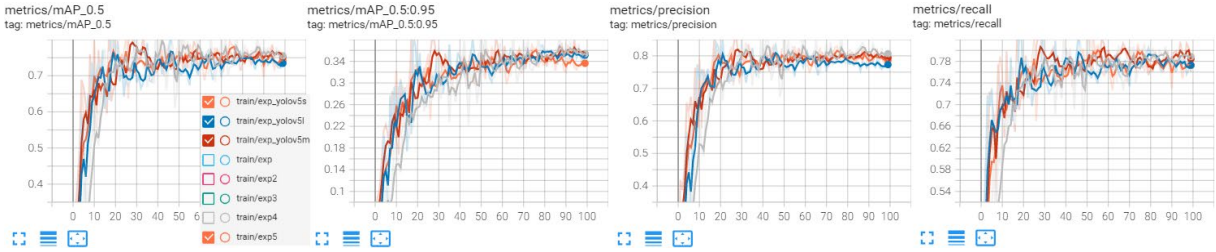



**图 3.3 召回率 Recall 与精确率 Precision**

mAP 即平均精度均值，是衡量模型训练效果的一个综合指标。图中的 mAP_0.5 指的是当损失函数 IoU 取 0.5 时的 mAP，mAP_0.5:0.95 指的是当 IoU 分别取 0.5～0.95 时(步进 0.05)mAP 的平均值[6]，其计算公式如下所示，其中 mAP_0.5 在模型迭代到 100 次时稳定在 0.75 左右。

$$AP=\sum_{i=1}^{n-1}(r_{i+1}-r_i)P_{\text{inter}}\ (r_i+1);mAP=\frac{\sum_{i=1}^{k}AP_i}{k}$$

## 4. 系统实现

### 4.1 系统搭建

为了应用基于 YOLOv5 算法的人脸口罩佩戴检测模型，本系统采用 PyQt5 框架进行 GUI 界面搭建，将系统分为图像检测模块和视频检测模块。运用 Python 语言和 OpenCV 计算机视觉库对上传的图像数据、视频数据以及摄像头实时捕获的数据进行处理，将处理好的数据送入模型进行推理，识别出数据中的人脸是否佩戴口罩。系统结构如下图所示：


**图 4.1 系统结构图**

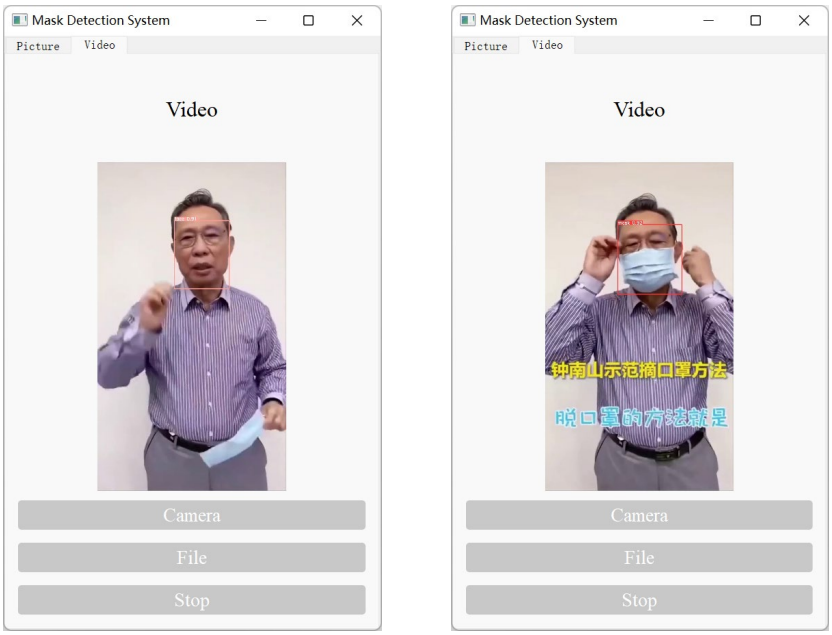
YOLOv5s 是 YOLOv5 系列模型中体积最小的一个模型，其便于部署、实时检测性能好。此处选择 YOLOv5s 作为系统检测的网络模型。

### 4.2 效果展示

本项目具有图像检测、视频检测和摄像头实时检测的功能，此处展示前两个功能。

**(a) 图像检测**


**(b) 视频检测**




通过观察结果，可以发现在图像与视频检测中，检测系统都可以根据人的佩戴情况变化，成功圈定出戴口罩与未戴口罩的人，并给出了其预计准确率。

## 5. 总结

针对疫情防控的实际需求，通过数据集获取、数据集标注、模型选择与训练、模型评估和系统搭建等一系列工作，设计并实现了基于 YOLOv5 的疫情防控口罩佩戴检测系统。该系统能有效快捷检测是否佩戴口罩，较好地满足了实际的应用需求。

## 参考文献

[1] 王迪聪，白晨帅，邬开俊. 基于深度学习的视频目标检测综述[J]. 计算机科学与探索, 2021.

[2] Leibe B , Matas J , Sebe N , et al. [Lecture Notes in Computer Science] Computer Vision – ECCV 2016 Volume 9905 || SSD: Single Shot MultiBox Detector[J]. 2016.

[3] 谭显东，彭辉. 改进 YOLOv5 的 SAR 图像舰船目标检测[J]. 计算机工程与应用, 2022, 58(4):8.

[4] 陈科圻，朱志亮，邓小明,等. 多尺度目标检测的深度学习研究综述[J]. 软件学报, 2021, 32(4):27.

[5] 谈世磊，别雄波，卢功林,等. 基于 YOLOv5 网络模型的人员口罩佩戴实时检测[J]. 激光杂志, 2021.

[6] 陈兆凡，赵春阳，李博. 一种改进 IoU 损失的边框回归损失函数[J]. 计算机应用研究.