

目录

1. 研究内容综述	1
1.1 研究背景	1
1.2 任务分解与计划	1
2. 研究方案与成果	3
2.2 数据预处理	3
2.2.1 颜色空间模型	3
2.2.2 图像平滑降噪算法	4
2.2.3 图像增强	6
2.2.4 数据集标注	8
2.3 基于霍夫圆的传统检测	10
2.3.1 边缘提取	10
2.3.2 霍夫圆检测	11
2.4 HOG+SVM 算法	13
2.4.1 数据标注与处理	13
2.4.2 HOG 特征提取	14
2.4.3 SVM 训练	15
2.4.4 滑动窗口算法或 NMS 算法	15
2.5 YOLOv5 目标检测算法	16
2.5.1 YOLOv5 简介	16
2.5.2 数据集制作与预处理	17
2.5.3 训练步骤	18
2.5.4 部署与使用方法	19
2.5.5 评判指标	20
2.6 追踪器设计	21
3. 研究总结	23
4. 个人感悟与建议	25

1. 研究内容综述

1.1 研究背景

视频监控已经深入到国民生活的方方面面，对视频监控画面的处理可以极大的提升监控系统的应用价值。客流量作为公交车站、博物馆、图书馆及购物中心等等公共场所一个极为重要的统计参数，对于管理决策等都具有很重要的意义。而这些公共场所一般都具有非常完善的视频监控系统，因此，通过对视频监控画面的处理，利用计算机视觉技术，综合使用图像处理、模式识别等智能视频分析方法，智能处理监控画面，以实现各项自动监测处理功能。

目前，基于监控的视频处理最常用的领域便是客流量检测。客流量统计的本质是人数统计，本次大作业便是基于公交车摄像头视频的客流量统计。

1.2 任务分解与计划

依据给定大作业任务，我们将本次大作业基本任务分为以下四个步骤，逐步展开。

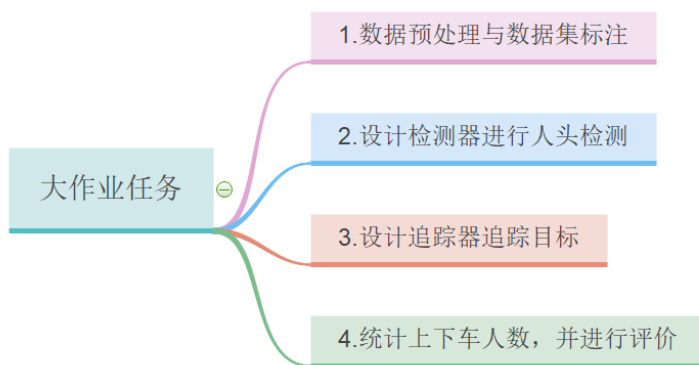


图 1-1 任务分解图

同时，在讨论之后，我们对本次作业提出以下基本要求与展望

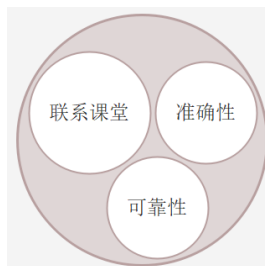


图 1-2 设计要求

结合组内成员的实际情况及时间分配状况，我们小组基本分工如下所示。



图 1-3 小组分工图

计划时间安排流程基本如下所示，而在实际完成过程中，由于期中考试周及其它种种原因，我们在中期进展比实际较晚了一周，但最终依据完成了设计的全部任务。

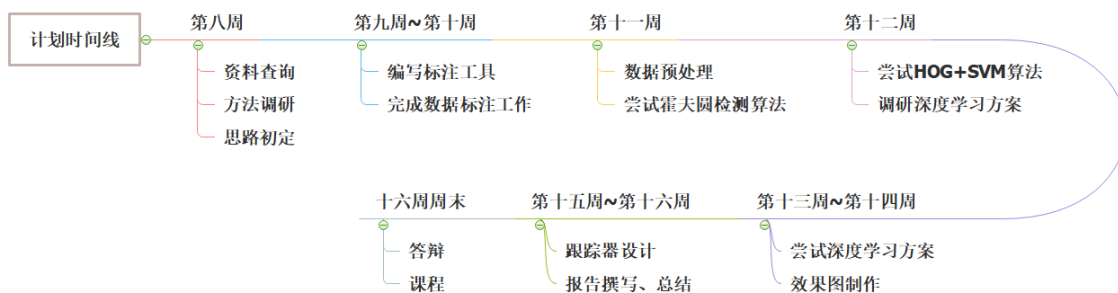


图 1-4 任务进度图

2. 研究方案与成果

在本次数字图像处理大作业一公交车上下客人流量检测任务中，我们小组以设计了基于霍夫圆的传统检测、HOG+SVM 人头检测算法、2. 目标检测网络 Yolo5 三种检测器为主要探究方向，并以此为基础展开，完成了本次大作业。

下面我们将从数据预处理、基于霍夫检测的传统方法、HOG 与 SVM 结合的检测算法、Yolo5 深度学习目标检测网络、追踪器与最终效果这五个部分展开叙述。

2.2 数据预处理

2.2.1 颜色空间模型

颜色模型指的是某个三维颜色空间中的一个可见光子集，它包含某个色彩域的所有色彩。一般而言，任何一个色彩域都只是可见光的子集，任何一个颜色模型都无法包含所有的可见光。我们课堂上主要提到的颜色模型为 RGB，HSV 等。

RGB 颜色空间是图像处理中最基础的颜色模型，它是在配色实验基础上建立的。RGB 颜色空间的主要观点是人的眼睛有红、绿、蓝 3 种色感细胞，它们的最大感光灵敏度分别落在红色、蓝色和绿色区域，其合成的光谱响应就是视觉曲线，由此可推论出任何彩色都可以用红、绿、蓝 3 种基色来配制。

HSI 颜色模型是孟赛尔颜色系统中的一种，以人眼的视觉特征为基础，利用三个相对独立、容易预测的颜色心理属性：色度、光强度和饱和度来表示颜色反映了人的视觉系统观察彩色的格式。

我们选取了两帧图像对其进行变换如下图所示。可以看到，RGB 颜色空间也是视频原始的颜色区间，HIS 颜色空间可以作为辅助的颜色判断或者进行亮度、对比度等通道的增强。之后是直接进行变换的灰度图，一些数字图像处理算法在灰度图上可以更好的得到运用，但灰度图单通道信息的存储也丧失了许多图像信息。最后则是直接对图像进行简单的二值化，可以看到二值化可以协助提取图像信息，但效果一般。

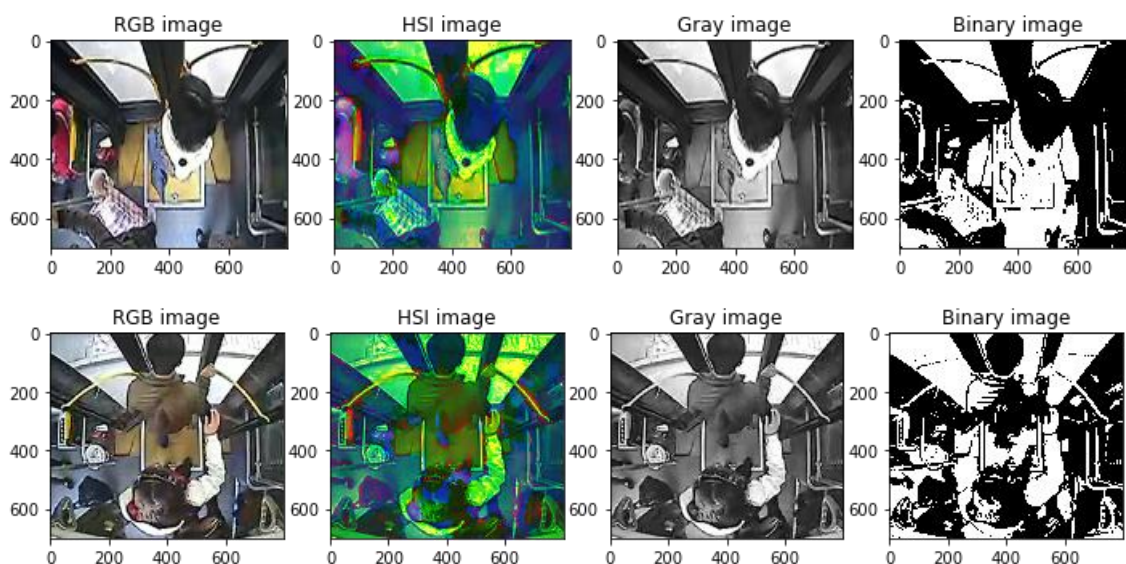


图 2-1 颜色空间模型图

2.2.2 图像平滑降噪算法

在数字图像处理领域，常采用滤波器对图像进行平滑处理，以达到降噪的效果，在本次实践中，我们主要尝试采用了中值滤波、高斯滤波、双边滤波、开运算、闭运算等方法来对图像进行噪点去除。

中值滤波是一种非线性滤波算法，其原理很简单，就是用像素点邻域灰度值的中值来代替该像素点的灰度。由于中值滤波是基于排序统计理论，所以其对脉冲噪声、椒盐噪声会有很好的滤波效果，并能

保留图像的边缘细节。

高斯滤波是线性低通滤波中的一种，基本思想是：图像上的每一个像素点的值，都由其本身和邻域内其他像素点的值经过加权平均后得到。具体操作是，用一个核(又称为卷积核、掩模、矩阵)扫描图像中每一个像素点，将邻域内各个像素值与对应位置的权值相称并求和。从数学的角度来看，高斯滤波的过程是图像与高斯正态分布做卷积操作。

双边滤波是一种非线性的滤波方法，是结合图像的空间邻近度和像素值相似度的一种折衷处理，同时考虑空域信息和灰度相似性，达到保边去噪的目的。双边滤波器的好处是可以做边缘保存，它比高斯滤波多了一个高斯方差 σ_d ，它是基于空间分布的高斯滤波函数，所以在边缘附近，离的较远的像素不会太多影响到边缘上的像素值，这样就保证了边缘附近像素值的保存。

开运算与闭运算，也能平滑图像的轮廓，这两种运算基于腐蚀膨胀。开运算可以平滑图像轮廓，削弱狭窄的部分，去掉细长的突出、毛刺和孤立斑点，而闭运算可以平滑图像轮廓，融合窄的缺口和细长的弯口，填补裂缝及破洞。

我们使用以上五种方法对选取的两帧图像进行变换，效果图如下所示，可以看到各效果图均有不同程度的平滑效果。



图 2-2 平滑降噪图

2.2.3 图像增强

本次实验中，存在许多不同光照条件下的视频环境，光照或黑暗的环境对于统一的算法提出了极大的挑战，因此，我们必须对图像的亮度进行调节。在这里，我们尝试了在灰度空间进行灰度直方图均衡化与 HSV 空间对亮度进行均衡化两种方法。

直方图均衡化是一种简单有效的图像增强技术，通过改变图像的直方图来改变图像像素点，主要用于增强动态范围偏小的图像的对比度。

对灰度图直方图均衡化的基本原理是：对在图像中像素个数多的灰度值（即对画面起主要作用的灰度值）进行展宽，而对像素个数少的灰度值（即对画面不起主要作用的灰度值）进行归并，从而增大对比度，使图像清晰，达到增强的目的。

灰度均衡亮度增强得到的效果如下图所示，图像的亮度与原有图像相比有了较好的增强，从灰度直方图也可以发现，灰度的分布有了一些改进。

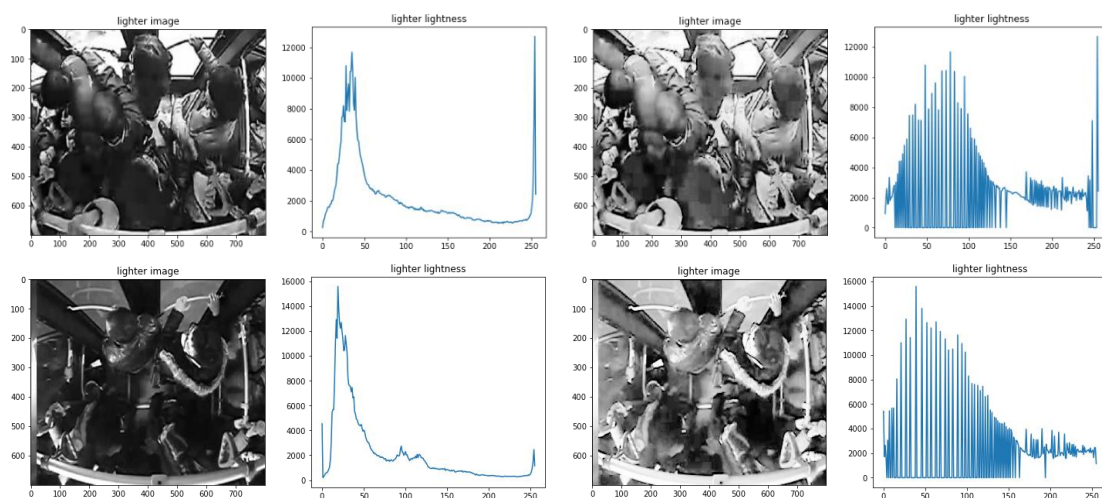


图 2-3 灰度直方图均衡化

我们也尝试了在 HSV 空间对图像亮度单通道，进行直方图均衡化。具体操作与灰度图的方法类似，只不过将处理对象变为亮度通道，处理的结果如下所示，从亮度通道的直方图可以看出亮度分布有了明显改善，缺点是对单通道运算可能会对颜色造成畸变。

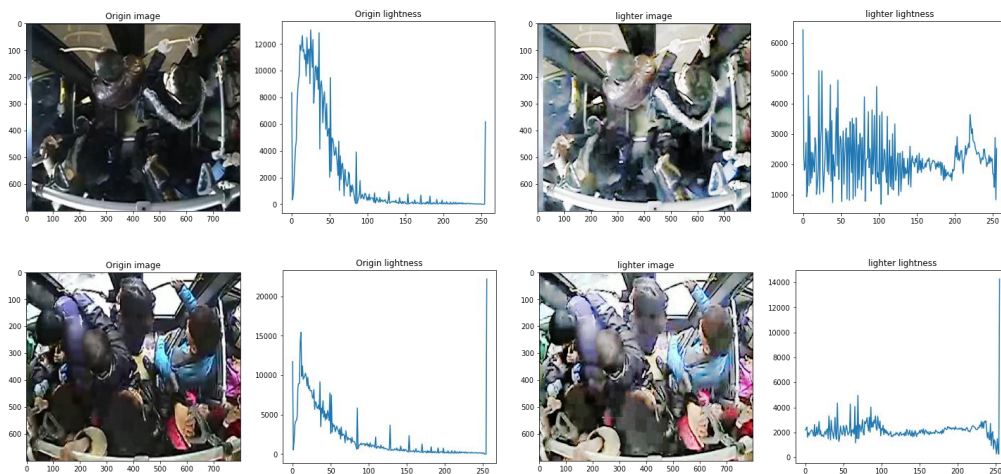


图 2-4 亮度直方图均衡化

2.2.4 数据集标注

在实际的图像处理算法中，无论是 SVM 亦或者深度学习网络，我们都需要进行样本的标注，考虑到公交车上可能存在的拥挤状况，因而我们主要对人头进行了标注

考虑到是要对视频进行标注，有一个跟踪器会很大程度上提高标注的效率，所以我们先尝试自主编写了一个基于 OpenCV 图像跟踪器的视频标注工具，在选取一个 roi 区域作为目标之后，使用图像跟踪器自动标注后续图像帧。当发现跟踪器跟踪失效，或跟踪偏差过大后，通过人工的方式，手动重置跟踪区域，这样可以较大地提升了视频数据标注的速度，自制的标注工具 github 开源地址为：
<https://github.com/xinyang-go/LabelVideo>，实际标注效果图如图 2-5 所示。

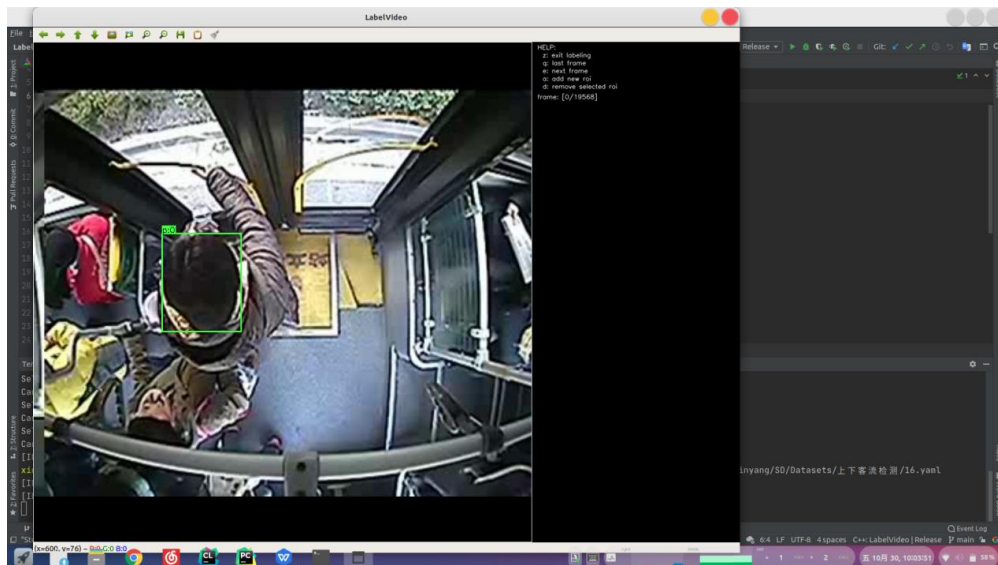


图 2-5 自制标注工具

之后，在标注过程中，我们发现使用简单的追踪器标注这些视频在效率上仍然达不到我们的预期。所以，我们最后选用了开源项目 DarkLabel 这一工具，与我们编写的标注工具相比，DarkLabel 的追踪器较优，可以帮助我们提高标注效率。DarkLabel 的标注界面如下图所示。



图 2-6 DarkLabel 标注工具

在得到合适的标注工具后，我们在六个实验视频中选取了四个视频进行逐帧的标注，共标注了 8w 帧的照片，考虑到相邻帧图像的相似性，其中实际的有效数据数要低于这一数据。

2.3 基于霍夫圆的传统检测

使用霍夫圆进行检测这一策略较为简单，其基本步骤为：图像预处理、边缘提取、霍夫圆检测

但简单的策略与步骤也代表着这一方法的检测效果不会太好，我们尝试这一方法主要是对课堂上学到的一些传统图像处理算法进行实践与进一步的理解。

由于图像预处理在之前已单独展开叙述，此节不再加以赘述。

2.3.1 边缘提取

边缘是指周围像素灰度有阶跃变化或屋顶变化的那些像素的集合边缘是图像的最重要的特征之一。边缘检测主要是灰度变化的度量、检测和定位。目前有很多种不同的边缘检测算子，一阶的有 Roberts 算子、Sobel 算子、Prewitt 算子等，二阶算子有 Laplacian 算子、Canny 算子、Marr-Hildreth 等。

本次实验中，我们主要尝试了 Canny 算子、Laplacian 算子、Sobel 算子等等。

Laplacian 算法采用 Laplacian 算子，拉普拉斯算子是一个二阶的微分是最简单的各向同性微分算子，它具有旋转不变性，在边缘检测算法中有着广泛的应用。类似的，Sobel 算子是一个一阶差分算子，

常常用于对单独 X 或 Y 方向进行提取或加强，本次实验是采用了两个方向的结合。

Canny 算法，是目前最为经典的边缘检测算法，相比与其它的算法，他采用了基于边缘梯度方向的非极大值抑制与双阈值的滞后阈值处理的策略。他的基本步骤是对输入图像进行高斯平滑，再计算梯度幅度和方向来估计每一点处的边缘强度与方向，并根据梯度方向，对梯度幅值进行非极大值抑制，最后使用双阈值处理和连接边缘。

使用以上三种算法并采用滤波去噪、灰度截断等一系列处理算法后的效果图如下所示，可以看到 Canny 算法具有较好的边缘提取效果，Laplacian 算法也基本可以提取得到所需的边缘，而一阶 Sobel 算子效果相比而言较差。因而，我们最终主要采用了 Laplacian 算法与 Canny 算法进行之后的探究。

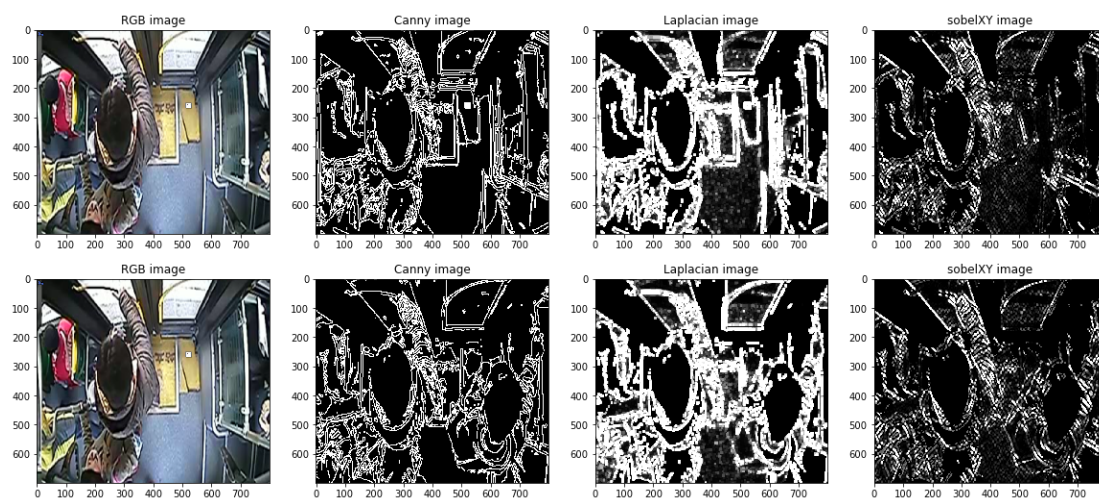


图 2-7 边缘检测算法效果图

2.3.2 霍夫圆检测

霍夫圆检测基于霍夫变换，采用投票的思想。其基本思路是每一个非零像素点都有可能是一个潜在的圆上的一点，通过投票，生成累

积坐标平面，设置一个累积权重来定位圆。

二维图像空间中一个圆转换为该圆半径、圆心横纵坐标所确定的三维参数空间中一个点的过程。因此，圆周上任意三点所确定的圆，经 Hough 变换后在三维参数空间应对应一点。该过程类似于选举投票过程，圆周上任意三个点为一选举人，而这三个点所确定的圆则为一候选圆。遍历圆周上所有点，任意三个点所确定的候选圆进行投票。遍历结束后，得票数最高点所确定的圆为该圆周上绝大多数点所确定的圆。

使用霍夫变换的效果图如下所示，可以看出，霍夫变换具有一定的完成任务要求的能力，但其效果较差，存在许多漏检、乱检问题，且极为依赖相关参数的调节，不具有普适性，很难对于视频进行处理，也无法满足多种环境下公交车客流量检测的任务，因而很难应用到实际应用中。

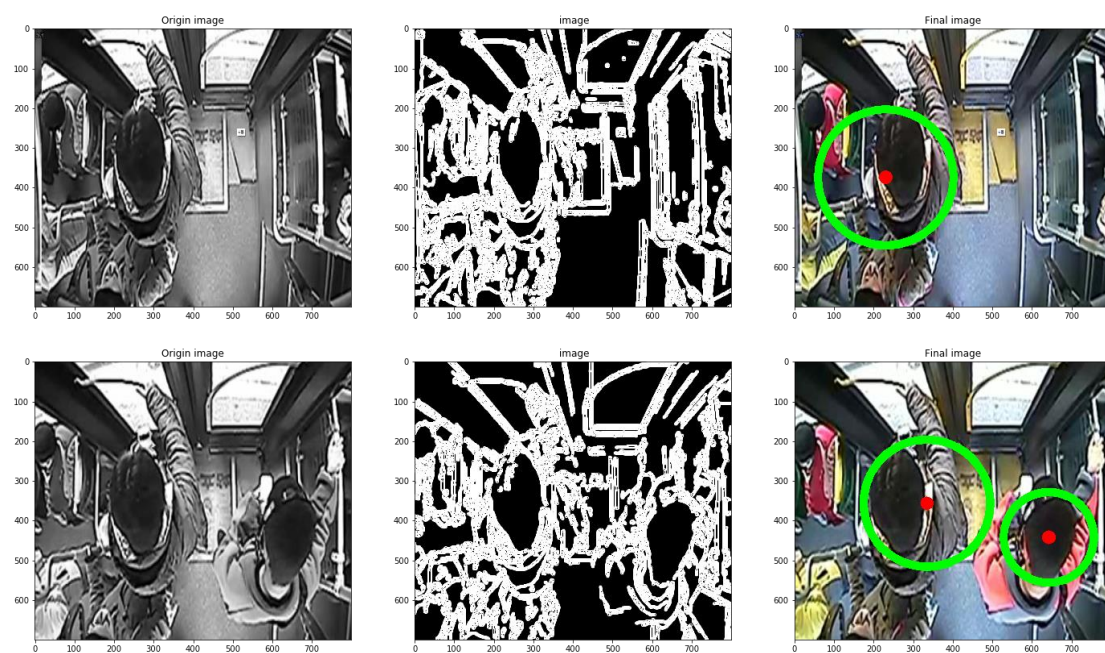


图 2-8 霍夫圆检测算法效果图

2.4 HOG+SVM 算法

在 2005 年 CVPR 上，两位研究人员提出利用 Hog 进行特征提取，利用线性 SVM 作为分类器，从而实现人的检测，此后 HOG 与 SVM 结合的方法开始在行人检测上有了较多的应用。这一方法的基本步骤为：

1. 数据标注与预处理
2. HOG 运算
3. SVM 训练
4. 滑动窗口或 NMS 算法

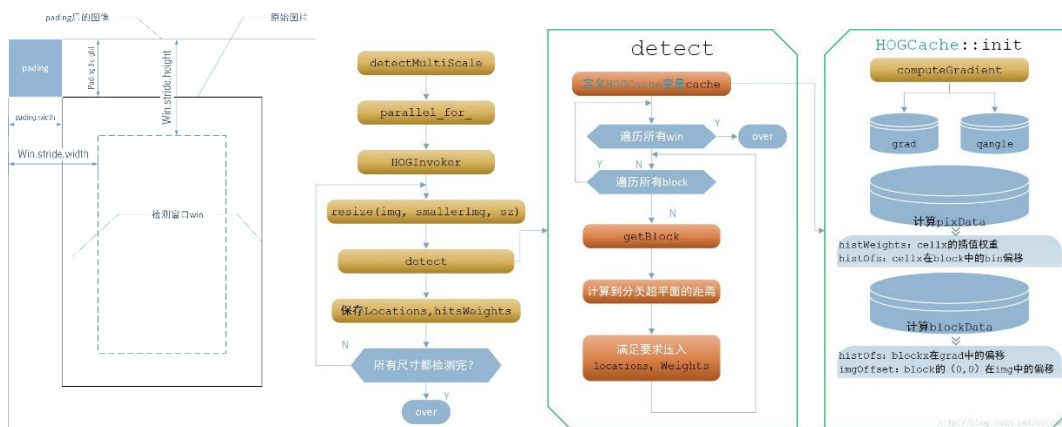


图 2-9 HOG+SVM 算法流程图

2.4.1 数据标注与处理

我们基于标注好的数据从视频中采集了人头正样本，并采用随机采样的方法获取了负样本。并对这些数据进行了简单的图像预处理，然后将它们统一归一化到 64*64 的大小，正负样本的部分截图如下所示。

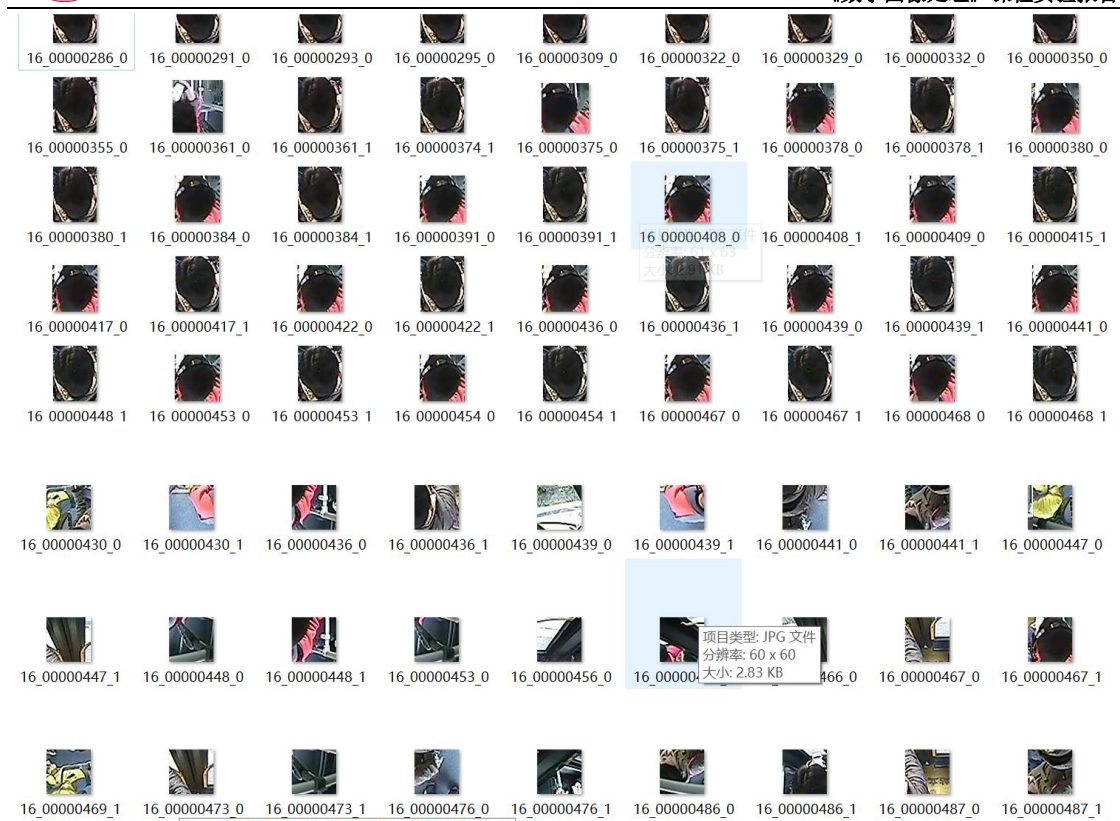


图 2-10 正负样本图

2.4.2 HOG 特征提取

HOG 特征提取是常用于人的特征提取方法，其核心思想是所检测的局部物体外形能够被光强梯度或边缘方向的分布所描述。

通过将整幅图像分割成小的连接区域，每个连接区域生成一个方向梯度直方图，这些直方图的组合可表示出目标的描述子。为改善准确率，局部直方图可以通过计算图像中一个较大区域的光强被对比标准化，然后用这个值归一化这个较大区域中的所有连接区域，这个归一化过程完成了更好的照射/阴影不变性。与其他描述子相比，HOG 得到的描述子保持了几何和光学转化不变性。因此 HOG 描述子尤其适合人的检测。

在实际使用中，我们对标注好的图片样本进行了 HOG 特征提取，

以便之后的 SVM 分类训练。HOG 特征提取后的效果图如下所示。

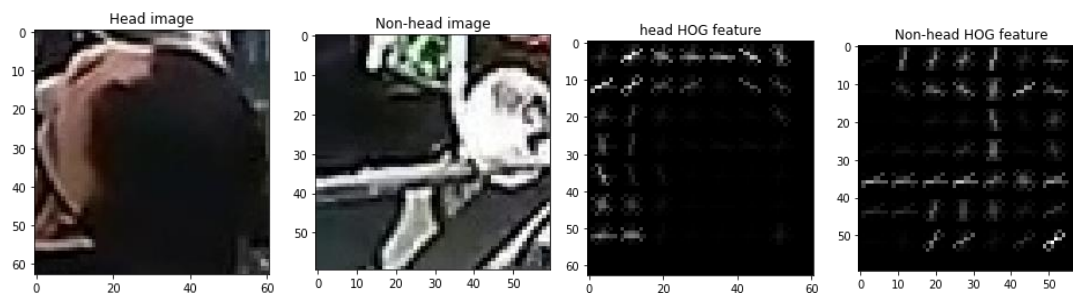


图 2-11 HOG 特征图

2.4.3 SVM 训练

支持向量机（SVM）是在所有知名的数据挖掘算法中最健壮，最准确的方法之一，它属于二分类算法，属于有监督的学习，可以支持线性和非线性的分类。

我们使用 sklearn 中 svm 包对 HOG 提取之后的特征向量进行分类，选取 75%的数据作为测试集，25%的数据作为训练集进行训练。训练得到样本的准确率为 81.27%，在测试数据集上有较好的效果。

2.4.4 滑动窗口算法或 NMS 算法

滑动窗口算法，顾名思义，使用一个窗口在被检测图像进行逐步检测，可以理解为这一算法是对图像的量化分块检测。

我们主要使用滑动窗口通过设置一定的窗口重叠率，以确保使用 SVM 分类器时不会丢失所检测图像，并检测为正的样本数绘制在热力图中，达到抑制重复检测的效果，同时也保证了检测的准确率。

类似的这一部分也可以通过 NMS（非极大值抑制）算法实现，在此不加赘述。

使用这一算法最终效果图如下，展示的效果图分别为使用滑动窗口后的热力图与基于连通域对热力图标签化的结果图。可以看到，这一算法有一定的分类与识别能力，但在人头簇拥的区域，如图二的左下角，容易识别连片，也有一定的几率产生误识别与漏识别。同时，这种方法过于依赖数据，对未训练的图像数据效果一般。当然，这一算法作为经典的行人识别算法，还有着较大的可取性与很大的优化空间。

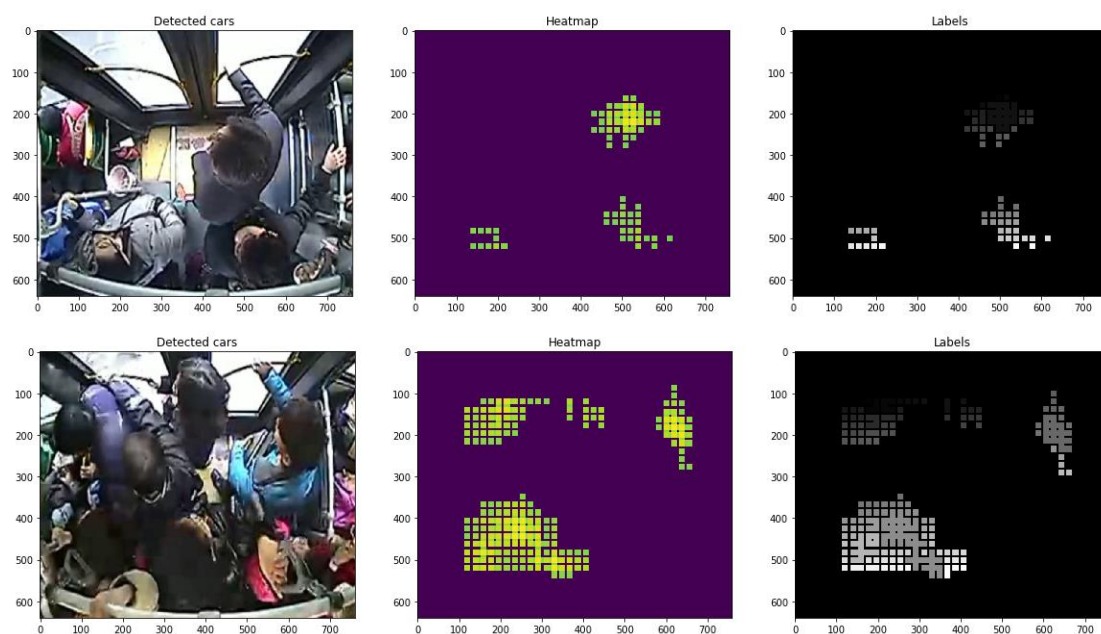


图 2-12 HOG+SVM 效果图

2.5 YOLOv5 目标检测算法

YOLOv5 是在 PyTorch 中实现的，它受益于成熟的 PyTorch 生态系统，支持更简单，部署更容易，相对于 YOLOv4，YOLOv5 具有以下优点：速度快、精度高、体积小等优点。

2.5.1 YOLOv5 简介

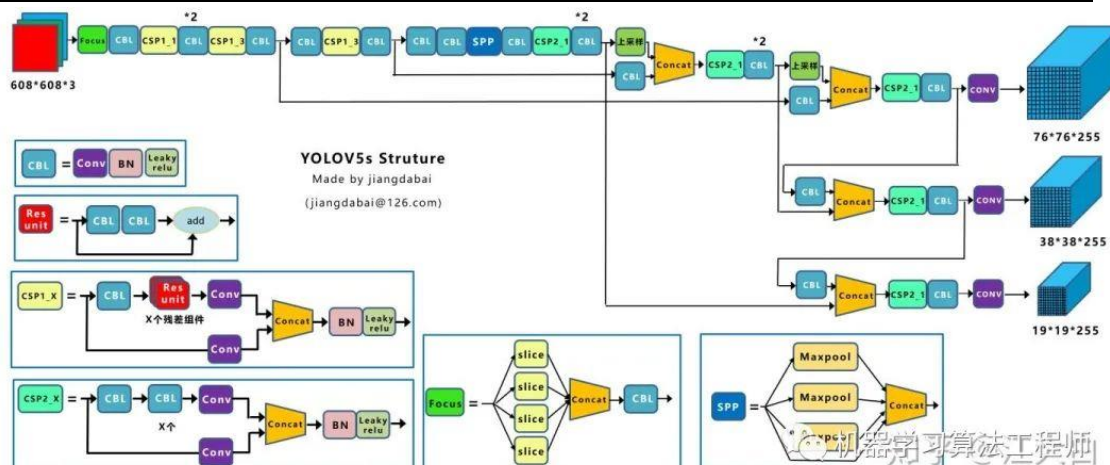


图 2-13 YOLOv5 结构图

YOLO 是一个轻量级单阶段目标检测网络，从 YOLOv1 到 YOLOv5 也经过了多次改良和升级。其主打的效果就是快速，同时检测精度也并不低。同时结构简单，没有特殊网络结构，使得它容易部署，从出现开始，便受到了工业界的广泛使用。因此网络上和 YOLO 网络相关的博客文章也十分丰富，极大的方便了人们的使用。因此我们选择了 YOLO 系列中的最新版本 YOLOv5 作为我们和目标检测算法。

YOLOv5 的特点主要包括：

1. 特征金字塔的检测方式，覆盖各种尺度的目标。
2. 全卷积网络，可以处理不同分辨率的输入图像。
3. mosaic 数据增强，拼接图像，减少训练时的样本不平衡现象。
4. Mish 激活函数、Dropblock、CSP 结构、Focus 结构。（融合一些较为有用的 tricks）

2.5.2 数据集制作与预处理

YOLO 需要的训练数据标签文件的格式是[类别 id] [目标中心 x 坐标] [目标中心 y 坐标] [目标宽度] [目标长度]。由于我们想要检测的目标仅为人头，所以将所有类别 id 都设为 0。上文提到的数据

标定软件，可以直接生成 YOLO 格式的标签文件。

训练过程中考虑到有些识别场景环境十分昏暗，而这样昏暗的数据在数据集中占比极少，为了平衡数据，我们尝试了对训练数据进行直方图归一化。同样在测试识别时，也将测试图片先直方图均衡，再进行识别。然而最终这样做并没有什么效果，所以被我们抛弃了。

考虑到明暗数据量的不平衡，我们对较暗数据进行重采样，而较亮数据进行降采样，以此一定程度上平衡数据量，这个方法起到了的了一定的作用，但并不算是明显。

2.5.3 训练步骤

在准备好数据之后，训练数据不算麻烦。首先从 github 上下载下来官方的 YOLOv5 代码（基于 Pytorch1.6），然后选择一个具体的模型。这里我们选择了模型大小适中的 YOLOv5m 版本。然后修改配置文件中的目标类别数，因为我们只识别人头，所以我们的目标类别数是 1。

```
# parameters
nc: 1 # number of classes
depth_multiple: 0.67 # model depth multiple
width_multiple: 0.75 # layer channel multiple

# anchors
anchors:
  - [10,13, 16,30, 33,23] # P3/8
  - [30,61, 62,45, 59,119] # P4/16
  - [116,90, 156,198, 373,326] # P5/32
```

图 2-14 修改后的参数文件的一部分

随后我们就可以开始训练了，为了提升训练速度，我们使用了学生创新中心的 GPU 池进行训练。大约 3-5 小时后，可以得到较好的识

别效果，训练的 Loss 图如下所示，可以看到训练效果较好。

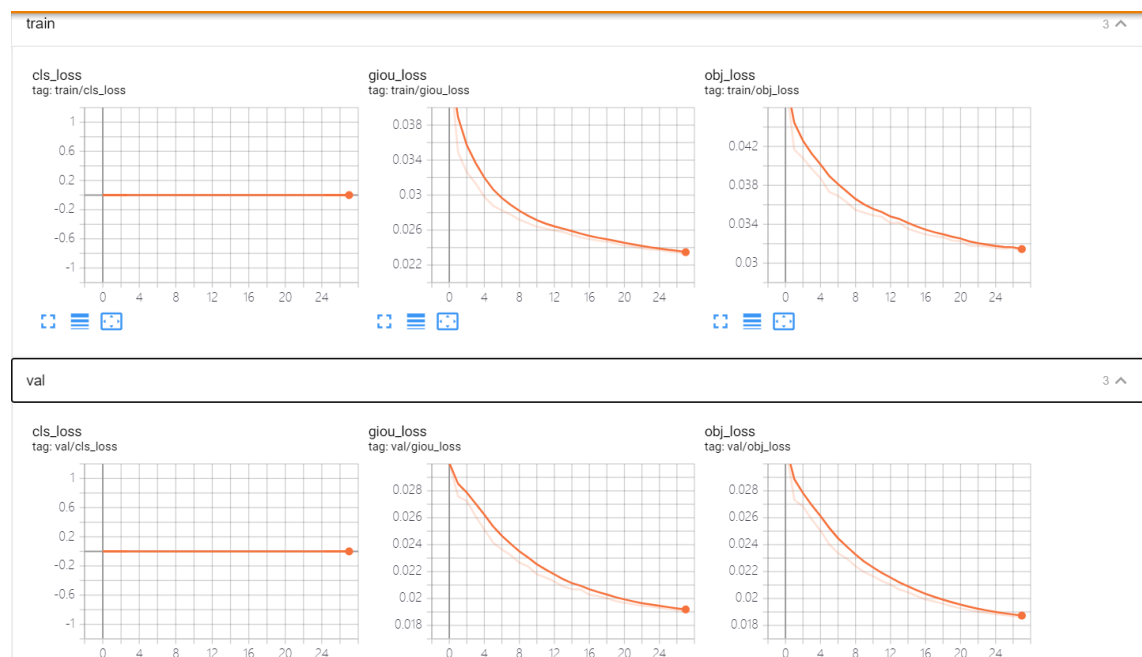


图 2-15 训练 Loss 图

2.5.4 部署与使用方法

为了正确使用 YOLOv5，不发生由于输入数据和训练时不一致导致无法正常识别。我们直接在官方提供的识别代码中进行修改，添加跟踪器和统计上下车人数的代码。由于读取视频和预处理视频图像帧的步骤都是由已有代码完成，所以这样一定不会出错。

最终检测效果如下所示。



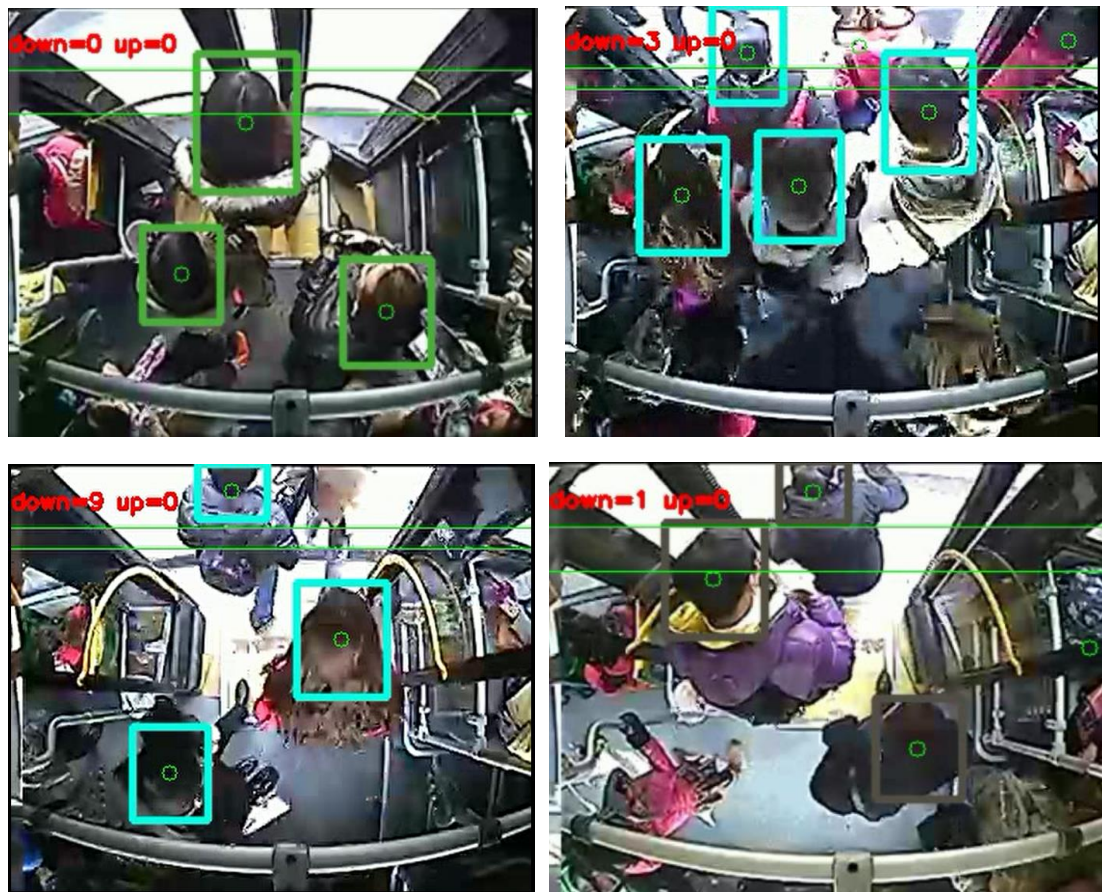


图 2-16 检测效果图

2.5.5 评判指标

对于实际的评价指标,我们选用了目标检测领域常用的mAP(mean average precision)、查准率 (Precision)、查全率 (Recall), 其各个效果图如下所示。

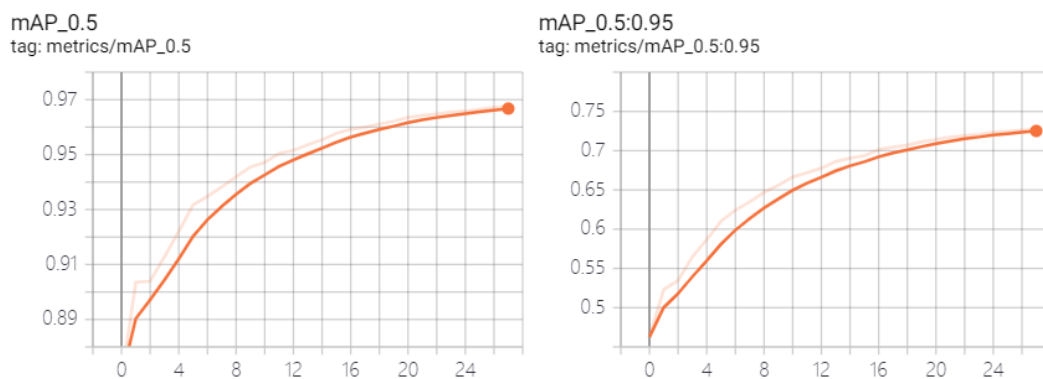


图 2-17 mAP 效果图

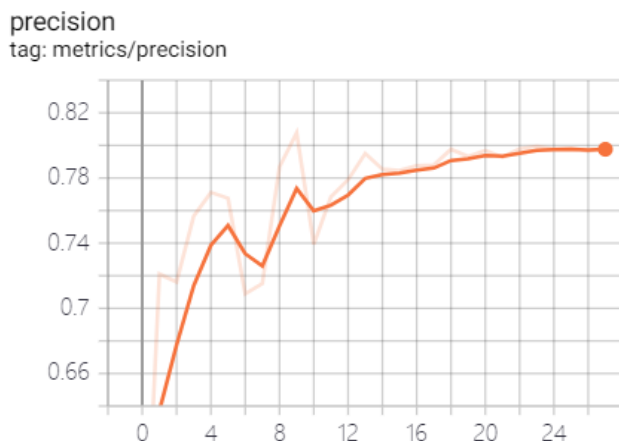


图 2-18 Precision 效果图

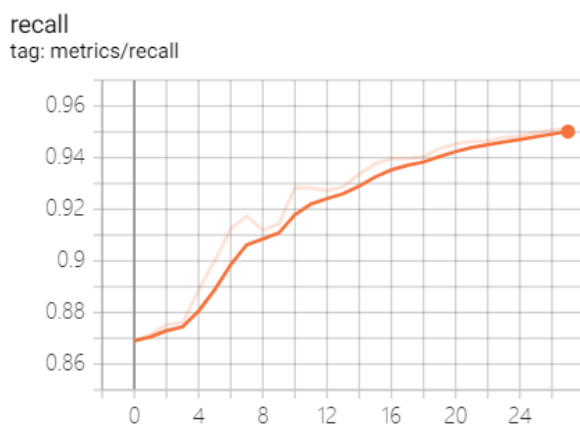


图 2-19 Recall 效果图

2.6 追踪器设计与评判

追踪器的设计需建立于精确的目标检测效果之上，优秀的追踪器可以辅助检测器进行滤波并准确的完成客流量的跟踪统计。

基于 YOLOv5 的目标检测网络，我们设计了以下的跟踪思路：当识别到一个目标后，对这个目标启用一个跟踪器。在后续帧中，如果连续 5 帧以内 YOLOv5 没有识别到这个目标，则使用跟踪器的结果，作为这个目标的检测。如果超过连续 5 帧，或跟踪失败，则视为改目标消失。

在尝试了一些基于 opencv 的追踪器后，我们最终选用了 KFC 这

一鉴别式追踪算法。KFC 全称为 Kernel Correlation Filter 核相关滤波算法,该跟踪器基于 MIL 跟踪器提出,根据 MIL 跟踪器在使用的多个正样本具有大的重叠区域,而这一重叠的数据可以形成一些良好的数学特性。利用这个跟踪器,跟踪更快,同时更准确。

一般而言,KFC 相对于其他的 tracking-by-detection 方法,速度得到了极大的提升,效果也相对较好,思想和实现十分简单。

由于跟踪器的效果展示需以视频流形式展出,无法置于 Word 中,因此请于附件视频中查看。

3. 研究总结

本次上下客流量检测大作业，我们小组以检测器的设计为主要研究目标，尝试了三种常用于人头检测的检测方法，完成了相关视频的数据集标注，完成了初步的数据预处理，并针对检测器与实际需求设计了相应的追踪器，最终完成了上下车客流量检测的基本任务。

本次实验的实际难点众多，多变的光照环境（如黑暗环境）及复杂的识别对象（如光头、小孩、遮挡等）使得精准识别的任务难度较大。在低流量密度的条件下，人群较为稀疏，遮挡问题并不严重，使用很多算法都可以进行检测识别；而在客流高峰时段，人流密度大，同时上下车的人数多，此时绝大多数的检测方法都将难以适用。

在追踪器这一难点问题上，我们首先尝试了基于传统霍夫变换的方法，发现该方法拟合效果极差，存在许多漏检、乱检问题，这是由于视频质量不高，且背景环境极为复杂造成的。部分参考文献指出，使用特殊设计的椭圆识别并进行相应的调参可以取得较好的效果，但我们考虑到设计要求的普适性与准确性要求，认为这一策略很难在实验任务中起到较大改进，难以达到预期，因此并未进行进一步的探究。

之后，在助教老师的推荐下，我们尝试了 HOG 特征提取与 SVM 训练的方式，这一方法与第一种方法相比，有了较大的效果改进，具备了一定应用的可能，但仍然存在一定的几率产生误识别与漏识别。同时，这种方法过于依赖数据，对未训练的图像数据效果一般。当然，这一算法作为经典的行人识别算法，也还有着较大的可取性与很大的

优化空间，由于时间紧迫，我们并未进行深度优化，而是尝试了基于 YOLO 网络的深度学习方法。

基于 YOLO v5 网络的深度学习方法，作为轻量级的单目标目标检测方法，结构简单且精度较高。我们进行简单的尝试后便发现这一目标检测网络针对本次任务有较好的表现效果，因而，我们再次基础上进行了一系列的优化与尝试，例如选用不同规模的数据集、采用不同的数据预处理算法等等进行训练，并最终选取了我们认为较优的最终模型，在正常光照条件下，这一模型的检测准确率极高，在极度黑暗的亮度条件下，识别效果有所降低，但仍在可接受范围内，在基于追踪器及视频流分析等处理下，亦可满足设计要求。

在追踪器的设计上，我们采用了以下基本思路：当识别到一个目标后，对这个目标启用一个跟踪器。在后续帧中，如果连续 5 帧以内 YOLOv5 没有识别识别到这个目标，则使用跟踪器的结果，作为这个目标的检测。如果超过连续 5 帧，或跟踪失败，则视为改目标消失。基于这一思想完成的追踪器，有着较优的实验效果。

当然，本次大作业的完成仍有着一些不足之处。首先，由于时间与精力的原因，我们在尝试两种传统方法后，并未进行深度的改进，未能触及这两种算法的上限。其次，最终设计的模型，极暗环境下拟合效果依旧存在一些小问题，我们也考虑过基于当前使用两种模式训练两个模型进行检测，但由于时间原因，未能尝试。