

第十一届中国大学生服务外包大赛

多一份安全，
少一份牵挂

项目开发详细文档

目录

第一章 需求分析	1
1.1 项目背景	1
1.2 目标用户	2
1.3 竞争产品	3
第二章 概要设计	5
2.1 系统架构	5
2.2 总体功能模块设计	6
2.3 用户信息采集模块	6
2.4 用户信息管理模块	7
2.5 人脸识别模块	7
2.6 安全帽检测模块	8
2.7 签到签退模块	8
第三章 详细设计	9
3.1 关键技术	9
3.2 yolo v3 网络	9
3.3 LBPH 算法	12
3.4 界面设计	14
3.5 数据库设计	14
第四章 测试报告	16
4.1 测试速度	16
4.2 准确度	16
第五章 使用流程	17
第六章 项目总结	21
6.1 项目协调	21
6.2 水平提升	21
6.3 升级演进与商业推广	22

第一章 需求分析

1.1 项目背景

工地用安全帽是工作场所中最常使用、最便宜且最有效的个人防护用具,曾拯救过无数劳工的生命以 1995 年美国琳毛业伤害统计资料显示,在 6210 名的劳工死亡人数中有 1498 名是因头部伤害致死的, 约占全部死亡人数 1/4, 可见头部防护的重要性。

安全帽检测是实现安全生产的重要环节。在施工场景中, 作业环境的复杂性和工人的不安全行为易引发施工事故威胁生命安全, 其中施工人员头部受伤最易致命, 是造成施工事故的主要原因。安全帽的佩戴是行为规范的基本要求, 但在实际作业中时常存在未佩戴安全帽的行为。在施工场景中, 对人员佩戴安全帽的情况进行图像描述的研究, 能够降低作业风险, 从而排除安全隐患以保证人身安全。

施工现场的安全帽佩戴检测主要包括查看监控图像、人工巡检等方式。人工巡检费时费力, 实时监控图像需要监督人员长时间紧盯屏幕, 都会出现漏判情况。如果在施工现场没有佩戴安全帽, 将会造成严重的后果。使用深度学习方法实时检测现场工人安全帽佩戴情况, 具有重要的工程实用价值。

近年来随着视频监控技术的日益成熟, 视频监控系统已经开始向智能化的方向发展。安全帽作为一种个人头部防护用品, 在工厂的施工作业中, 是明令要求每个工人必须佩戴的。但目前对于工人安全帽的佩戴主要依靠人工监察, 效率十分低下。所以如果能够利用智能视频监控系统来进行安全帽佩戴检测, 进而实现报警和联动控制, 将对工厂的安全生产具有重要意义。

随着物联网、云计算、大数据和人工智能等信息技术的飞速发展, 全球化工业革命开始提上日程, 工业转型开始进入实质阶段。智能工厂作为工业智能化发展的重要实践模式, 已经引发行业的广泛关注。所谓的智能工厂就是利用物联网技术和监控技术加强信息管理和服务, 提高生产过程可控性、减少生产线人工干预, 以及智能化管理工厂安全生产。集智能手段和智能系统等新兴技术于一体, 构建智能、高效、节能、舒适的人性化工厂。

近几年，视频监控技术得到了蓬勃发展，以高清化、智能化为特征的第三代网络视频监控系统日益普及。目前工厂中的重要场景下，普遍安装了大量网络摄像头，所以利用这些网络摄像头来实现工厂的智能化视频监控，已经成为智能工厂发展道路上的关键环节。智能视频监控是利用计算机视觉技术对场景中感兴趣的目标进行检测、分类、跟踪和识别，能够有效地识别、追踪、理解监控场景，并且在出现异常事件时进行主动报警及记录，摆脱了传统的人工监察的方式。目前智能化视频监控主要用于工厂中的安全管理和经营管理。

在工厂的安全管理中，对于工人安全帽佩戴的监管一直是必不可少的。安全帽作为一种个人头部防护用品，在建筑施工作业中，是明令要求每个工人必须佩戴的。安全帽不仅可以承受和分散落物的冲击力，而且也可保护或减轻人员在高处坠落头部先着地时的撞击伤害程度，甚至可以挽救一个人的生命。所以我们可以看到工厂的入口都有“进入施工场地必须佩戴安全帽”之类的提示标语，安全规程上也有“施工人员不佩戴安全帽要被罚款”的规定。

但目前工厂中对工人安全帽佩戴情况的监管大多依赖人工监察，而且施工场地普遍存在作业面广、施工人员管理困难，安监人员难以及时准确了解工人的分布和作业情况，加之工地中各单位安全责任划分不明确，通常导致安全监督检查力度不够，所以这种人工监察的方式的效率非常低下。而且尽管工人在安全作业中离不开它，但大多工人安全意识不强，对佩戴安全帽的重要性意识不到位。据有关部门统计，工厂中坠落物致伤致死事故中的人员大多没有佩戴安全帽。

在这种背景下，如果能设计一套智能化的安全帽佩戴检测系统，既可以大大提升对于工人安全帽佩戴情况的监管效率，也迎合了当前工业 4.0 背景下智能工厂的发展前景，实现了智能化的安全管理。智能视频监控的关键技术主要包括实时图像获取、目标检测、目标跟踪以及行为分析等，结合安全帽佩戴检测的实际应用场景，我们设计并实现一套智能视频监控下的安全帽佩戴检测系统，该系统能够实时检测和跟踪到工人，捕捉工人的安全帽佩戴情况，及时发现没有佩戴安全帽的工人并发出警报，为工人的安全做出了一定的保障，对工厂的安全生产具有重大意义。

1.2 目标用户

此产品主要针对管理者对工地或者生产车间的工人进行安全帽的检测，并且

在出入口处能够对工人进行人脸识别完成签到签退的功能,方便管理者对工人们进行管理,提高生产效率。

越来越多的应用于 1.建筑工地 2.钻井平台 3.矿区 4.化工机械等高危区域,因为在这些区域物体掉落、倒塌造成的安全事故时有发生,根据《JGJ59-99 建筑施工安全检查标准》,进入施工现场必须戴安全帽,各行业的作业规范也对佩戴安全帽做了严格的规定。安全帽检测系统采用视频监控和 AI 智能分析,对高危区域实时监控,发现人员没有佩戴安全帽立即报警,有效的提升了现场管理水平。

在矿山生产中,安全帽是对矿工生命的保障,能够有效地减少或防止外来危险对矿工头部的伤害。安全帽的佩戴,可降低矿工工作时的作业风险。本文主要使用深度学习研究方法,针对矿工身份识别并对矿工不安全行为做出违规判识,这种不安全行为主要是安全帽的佩戴问题。

安全帽检测已经成为构建生产安全视频监控的一项重要技术,在煤矿、变电站、建筑工地等实际场景中需求广泛。

安全帽的使用单位分布在各行各业的建筑施工企业(建筑工地),主要有住建部门管理的建筑公司、电力建设公司,水利建设公司,公路桥梁建设公司,铁路建设公司,化工企业,工贸企业,矿山企业等使用单位,分布范围广,监管难度大。

1.3 竞争产品

目前市面上在生产车间的安全帽检测系统并不多,但工人刷脸进入工地相对比较普遍,我们产品的优势是可以记录工人的上下班时间,方便管理者查看考勤记录,并且当出现陌生人闯入或者工人未佩戴安全帽的情况会出发警报系统。同时开发成本较低,具有明显的价格优势。

1.蓝讯智能科技(BCI)推出的安全帽佩戴识别系统。

该系统基于普通 IPC 视频,对监控区域人员是否佩戴安全帽进行实时识别、跟踪和报警,实现了作业人员安全佩戴状况的全程自动监督,极大的提高了安全监督的时效性,降低了管理成本。系统采用先进计算机视觉和人工智能技术,适用于人员重叠或局部遮挡各种复杂背景,复杂场景准确率大于 90%,简单场景准确率大于 99%。

该系统具有以下功能及特性：

实时识别与告警：基于监控视频对工地或生产区域人员的佩戴安全帽情况进行实时检测和识别，对没有佩戴安全帽人员实时告警，告警信号可显示在监控客户端界面，也可以与扬声器关联以音频告警；告警功能可设置为开或关。

告警存储功能：将没有佩戴安全帽的人员的信息与快照实时存储到服务器数据库中，包括时间、地点、快照照片等。

历史记录查询：能够根据时间段、监控地点对没佩戴安全帽的行为进行历史记录查询，以列表形式显示查询结果，每条记录包括时间、地点、没有佩戴安全帽人员的现场照片等。

适应复杂场景：对人员较多时的人员重叠和部分遮挡、人员静止或运动、人员的各种姿态和角度均能够有很高的识别准确率，基于安全帽形状而不是安全帽颜色进行识别，提高对安全帽颜色识别的适应性。

高准确率：复杂场景识别准确率大于 90%，非复杂场景准确率大于 99%。

高实时性：对监控视频的识别率可达 30 帧/秒，实现实时检测和报警，准确率/实时性处于国际先进水平

可一次同时识别多人是否佩戴安全帽，触发报警并录入数据库。

2.北京舞水科技有限公司：1 万/套

产品亮点：

深度优化：针对动态视频，实时识别，深度优化。

不同环境适应性强：识别、跟踪精度高，对光线，阴天等不同环境适应性强；

不受人员外貌等影响，不受人员眼镜，胡须，发型、表情等遮挡影响；

不受人员姿态影响：不受人员正面、背面、侧面、跑动、低头等不同姿态影响。

识别参数：

识别内容：红色、黄色、蓝色、白色、橘色安全帽；

识别精确率：90%以上；

识别速率：大于 5fps；

识别范围：人员头部大小最低 30 像素可识别。

第二章 概要设计

2.1 系统架构

本系统由布署在施工现场、出入口闸道的安全帽检测监控设备以及服务器 3 部分组成，如图 2-1 所示。

在施工现场的安全帽检测监控设备通过无线连接的方式连接到局域网，设备通过摄像头取得现场实时数据，并将数据发送到服务器，进行数据处理，如果发现未佩戴安全帽者，系统将会发出警报。此外管理员还可以定期通过终端进行查看。

在出入口闸道处，通过一个带有摄像头的设备对出入的员工进行人脸识别，安全帽检测以及考勤。数据处理的方式和前者一样，如果发现未佩戴安全帽者或者发现外来人员，系统将会发出警报。管理员可以定期对员工出入生产现场的情况进行查看。

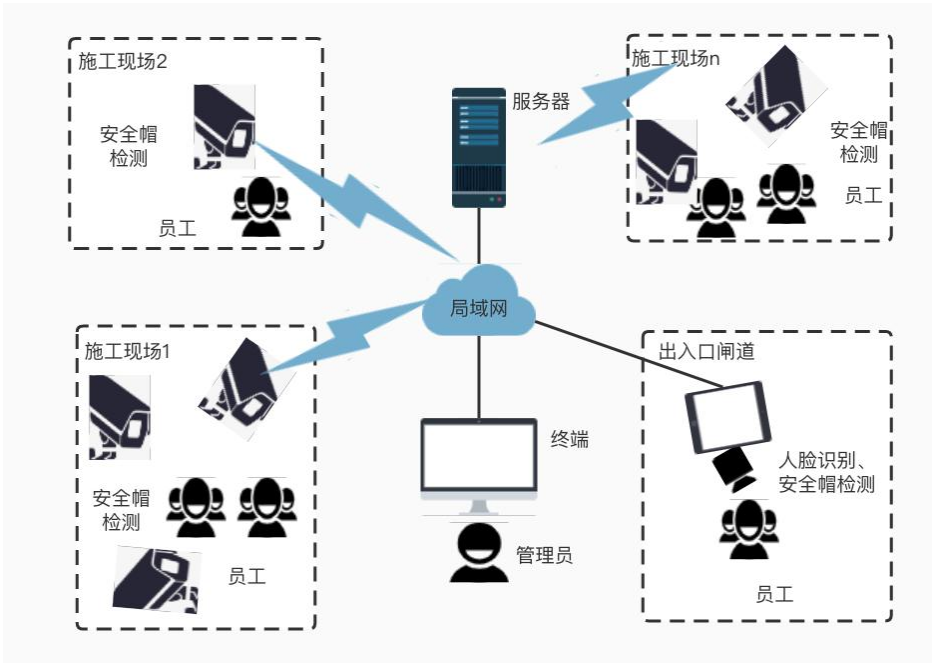


图 2-1 系统架构图

2.2 总体功能模块设计

在管理平台下面，分别有数据库管理、用户管理和系统配置 3 个部分，如图 2-2 所示。数据库部分记录了用户上下班的时间、用户第一次注册时的用户信息以及上班时间安全帽的佩戴情况，方便管理者来管理用户。在用户管理部分，管理员可以根据需求来添加、查询和删除某个用户的信息，便于管理。在系统配置部分，可在程序运行时对程序的一些设定进行配置，比如处理图像时可以直方图均衡化，在开始检测时可以开启调试模式等。

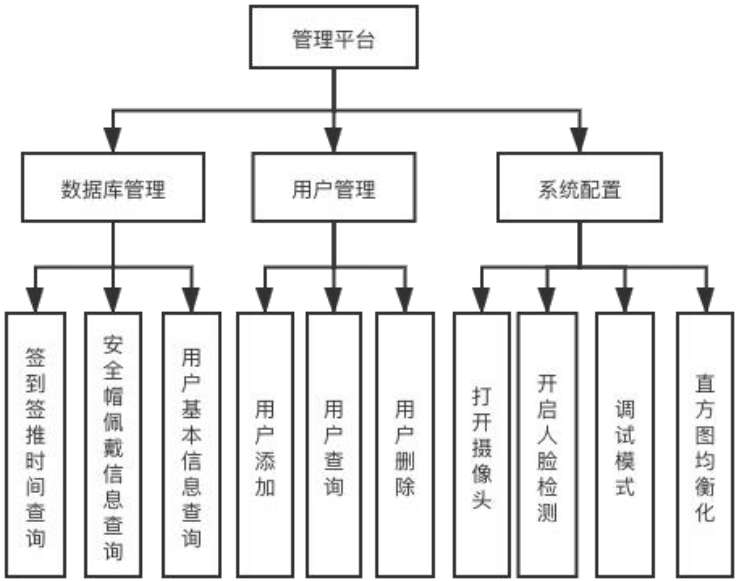


图 2-2 总体功能模块图

2.3 用户信息采集模块

此模块主要功能是注册用户，一个新的员工入职时，需要对员工的数据进行采集，包括员工的人脸数据以及姓名，工号等基本信息。采集用户人脸数据时，截取用户的人脸照片(≥ 100)，当然采集的照片越多，后期识别的准确度就越高。用户截取完自己的人脸图片之后，保存到项目目录下的 dataset 文件夹里面，同时，用户的信息保存到对用的数据库里面，如图 2-3 所示。

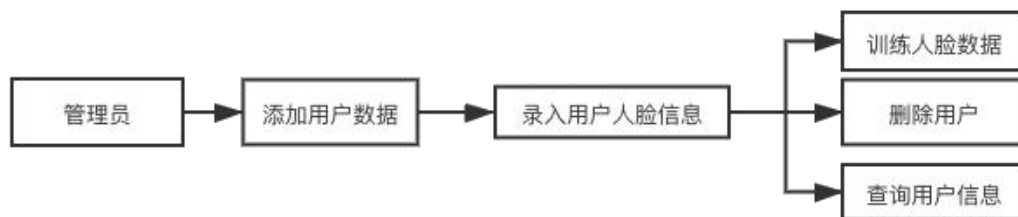


图 2-3 信息采集模块图

2.4 用户信息管理模块

此模块主要是对用户的信息进行管理，管理员可以训练刚才截取的用户的人脸的照片，并在项目目录下的 recognizer 文件夹下生成一个 yml 训练文件，当进行人脸识别时，就会加载这个文件。此模块还可以通过工号来查询用户的信息，当找到相应的用户时，可以对用户信息进行删除操作。当没有找到用户的信息时，将会提示错误，未找到用户的信息，如图 2-4 所示。



图 2-4 信息管理模块图

2.5 人脸识别模块

此模块负责人脸识别以及线程之间的通信。先通过获取一帧图像，用 haar 检测图像的人脸数据，并将此人脸截图交给安全帽检测模块进行检测，等获取到安全帽模块返回的结果后，将人脸信息以及安全帽的佩戴情况显示出来，如图 2-5 所示。

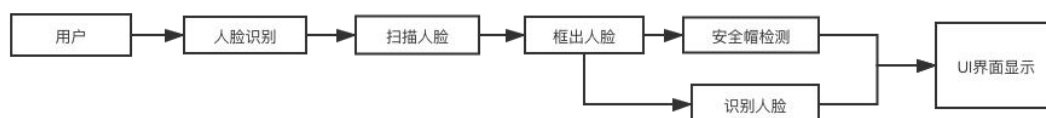


图 2-5 人脸识别模块图

2.6 安全帽检测模块

此模块主要负责接受人脸识别模块传过来的人脸照片，然后放入 yolov3 网络进行安全帽检测，最后将检测的结果返回给人脸识别模块，如图 2-6 所示。



图 2-6 安全帽检测模块图

2.7 签到签退模块

此模块主要负责在出入口对员工进行签退签到，在上班时间段，用户进入施工场地或者车间的时候，将进行人脸识别和安全帽检测，在人脸识别的同时，系统会记录下此时的时间，作为用户的上班時間，并保存在数据库之中。同理，用户下班时也是这样的流程，如图 2-7 所示。



图 2-7 签到签退模块图

第三章 详细设计

3.1 关键技术

- (1) 基于 yolo v3 网络实现安全帽检测
- (2) 基于 OpenCV 中的 Haar 特征和级联分类器与 dlib 库进行人脸检测及实时跟踪, 应用 LBPH 算法实现了人脸识别
- (3) 采用 sqlite3 进行序列化数据存储
- (4) 基于 PyQt5 设计 GUI 界面
- (5) 人脸识别和安全帽检测之间采用队列通信

3.2 yolo v3 网络

1. 网络模型

图 3-1 所示为 yolo v3 的网络模型。为了便于说明, 我们采用了 Levio 的模型。下面对图 3-1 一些内容作出补充解释:

DBL: 如图 3-1 左下角所示, 也就是代码中的 Darknetconv2d_BN_Leaky, 是 yolo v3 的基本组件。就是卷积+BN+Leaky relu。对于 v3 来说, BN 和 leaky relu 已经是和卷积层不可分离的部分了(最后一层卷积除外), 共同构成了最小组件。

resn: n 代表数字, 有 res1,res2,⋯,res8 等等, 表示这个 res_block 里面含有多少个 res_unit。这是 yolo v3 的大组件, yolo v3 借鉴了 ResNet 的残差结构, 使用这种结构可以让网络结构更深(从 v2 的 darknet-19 上升到 v3 的 darknet-53, 前者没有残差结构)。对于 res_block 的解释, 可以在图 1 的右下角直观看到, 其基本组件也是 DBL。

concat: 张量拼接。将 darknet 中间层和后面的某一层的上采样进行拼接。拼接的操作和残差层 add 的操作是不一样的, 拼接会扩充张量的维度, 而 add 只是直接相加不会导致张量维度的改变。

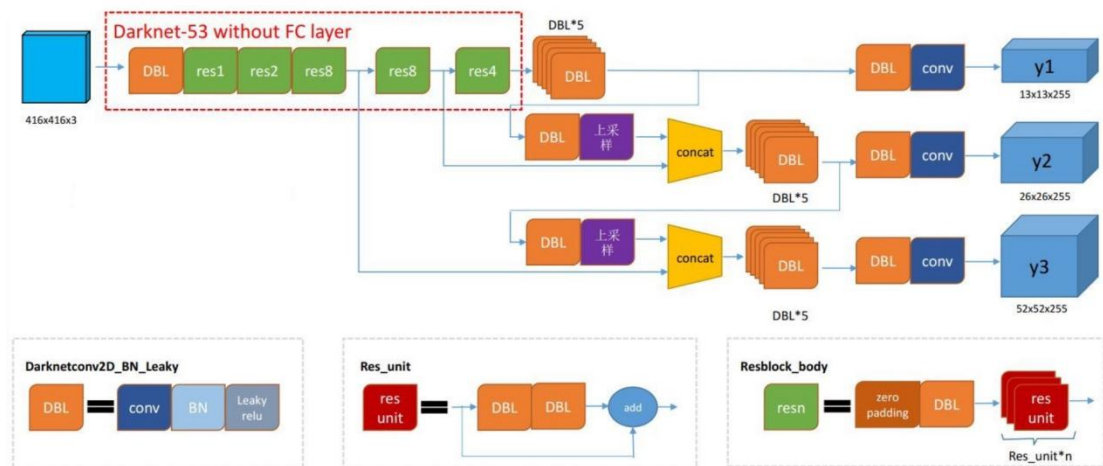


图 3-1 YOLOv3 结构图

2. darknet-53

整个 v3 结构里面，是没有池化层和全连接层的。前向传播过程中，张量的尺寸变换是通过改变卷积核的步长来实现的，比如 $\text{stride}=(2, 2)$ ，这就等于将图像边长缩小了一半(即面积缩小到原来的 $1/4$)。在 yolo v2 中，要经历 5 次缩小，会将特征图缩小到原输入尺寸的 $1/25$ ，即 $1/32$ 。输入为 416×416 ，则输出为 $13 \times 13 (416/32=13)$ 。

yolo v3 也和 v2 一样，backbone 都会将输出特征图缩小到输入的 $1/32$ 。所以，通常都要求输入图片是 32 的倍数。

为了达到更好的分类效果，作者自己设计训练了 darknet-53，相对于 ResNet-152 和 ResNet-101，darknet-53 不仅在分类精度上差不多，计算速度还比 ResNet-152 和 ResNet-101 强多了，网络层数也比他们少。

yolo v3 使用了 darknet-53 的前面的 52 层（没有全连接层），是一个全卷积网络，同时使用了大量残差的跳层连接，并且为了降低池化带来的梯度负面效果，作者直接摒弃了 POOLing，用 conv 的 stride 来实现降采样。

为了加强算法对小目标检测的精确度，yolo v3 中采用类似 FPN 的 upsample 和融合做法（最后融合了 3 个 scale，其他两个 scale 的大小分别是 26×26 和 52×52 ），在多个 scale 的 feature map 上做检测。

3. 输出

yolo v3 输出了 3 个不同尺度的 feature map，如图 3-2 所示的 y1, y2, y3。这也是 v3 论文中提到的为数不多的改进点：predictions across scales

这个借鉴了 FPN(feature pyramid networks), 采用多尺度来对不同 size 的目标进行检测, 越精细的 grid cell 就可以检测出越精细的物体。

y1,y2 和 y3 的深度都是 255, 边长的规律是 13:26:52

对于 COCO 类别而言, 有 80 个种类, 所以每个 box 应该对每个种类都输出一个概率。

yolo v3 设定的是每个网格单元预测 3 个 box, 所以每个 box 需要有(x, y, w, h, confidence)五个基本参数, 然后还要有 80 个类别的概率。所以 $3 \times (5 + 80) = 255$ 。

v3 用上采样的方法来实现这种多尺度的 feature map, 可以结合图 3-1 和图 3-1 右边来看, 图 3-1 中 concat 连接的两个张量是具有一样尺度的(两处拼接分别是 26x26 尺度拼接和 52x52 尺度拼接, 通过(2, 2)上采样来保证 concat 拼接的张量尺度相同)。

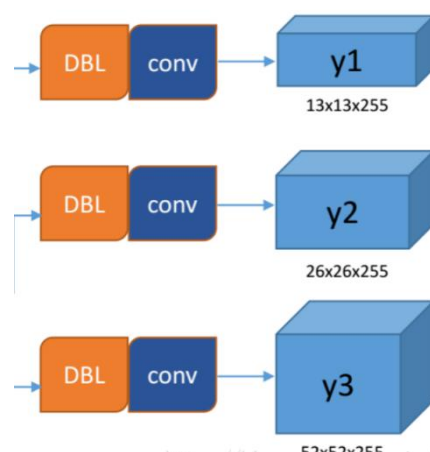


图 3-2 输出结构

4. 损失函数

在 yolo 系列论文里面只有 yolo v1 明确提出了损失函数的公式。v1 中使用了一种叫 sum-square error 的损失计算方法, 就是简单的差方相加。在目标检测任务里, 有几个关键信息是需要确定的:

(x,y),(w,h),class,confidence(x, y), (w, h), class, confidence

(x,y),(w,h),class,confidence

根据关键信息的特点可以分为上述四类, 损失函数应该由各自特点确定。最后加到一起就可以组成最终的 loss_function 了, 从而通过 loss_function 来解决端到端的训练。从代码可以分析出 v3 的损失函数, 相比于 v1 中简单的总方误差,

作出了一些调整:

```
xy_loss = object_mask * box_loss_scale * K.binary_crossentropy(raw_true_xy,
raw_pred[...,0:2], from_logits=True)

wh_loss = object_mask * box_loss_scale * 0.5 * K.square(raw_true_wh -
raw_pred[..., 2:4])

confidence_loss = object_mask * K.binary_crossentropy(object_mask,
raw_pred[...,4:5],from_logits=True)+(1-object_mask)*K.binary_crossentropy(object_ma
sk, raw_pred[..., 4:5],from_logits=True) * ignore_mask

class_loss = object_mask * K.binary_crossentropy(true_class_probs, raw_pred[...,
5:], from_logits=True)

xy_loss = K.sum(xy_loss) / mf
wh_loss = K.sum(wh_loss) / mf
confidence_loss = K.sum(confidence_loss) / mf
class_loss = K.sum(class_loss) / mf

loss += xy_loss + wh_loss + confidence_loss + class_loss
```

以上是一段 keras 框架描述的 yolo v3 的 loss_function 代码。忽略恒定系数不看, 可以从上述代码看出: 除了 w, h 的损失函数依然采用总方误差之外, 其他部分的损失函数用的是二值交叉熵。

3.3 LBPH 算法

1. 圆形 LBP 算子

基本的 LBP 算子的最大缺陷在于它只覆盖了一个固定半径范围内的小区域, 这显然不能满足不同尺寸和频率纹理的需要。为了适应不同尺度的纹理特征, Ojala 等对 LBP 算子进行了改进, 将 3×3 邻域扩展到任意邻域, 并用圆形邻域代替了正方形邻域, 改进后的 LBP 算子允许在半径为 R 的圆形邻域内有任意多个像素点, 从而得到了诸如半径为 R 的圆形区域内含有 P 个采样点的 LBP 算子, OpenCV 中正是使用圆形 LBP 算子。

2. 旋转不变模式

从 LBP 的定义可以看出, LBP 算子是灰度不变的, 但却不是旋转不变的, 图

像的旋转就会得到不同的 LBP 值。Maenpaa 等人又将 LBP 算子进行了扩展，提出了具有旋转不变性的 LBP 算子，即不断旋转圆形邻域得到一系列初始定义的 LBP 值，取其最小值作为该邻域的 LBP 值。下图给出了求取旋转不变 LBP 的过程示意图，图中算子下方的数字表示该算子对应的 LBP 值，图中所示的 8 种 LBP 模式，经过旋转不变的处理，最终得到的具有旋转不变性的 LBP 值为 15。

3. 等价模式

一个 LBP 算子可以产生不同的二进制模式，对于半径为 R 的圆形区域内含有 P 个采样点的 LBP 算子将会产生 2^P 种模式。很显然，随着邻域集内采样点数的增加，二进制模式的种类是急剧增加的。例如： 5×5 邻域内 20 个采样点，有 $2^{20} = 1,048,576$ 种二进制模式。如此多的二值模式无论对于纹理的提取还是对于纹理的识别、分类及信息的存取都是不利的。为了解决二进制模式过多的问题，提高统计性，Ojala 提出了采用一种“等价模式”（Uniform Pattern）来对 LBP 算子的模式种类进行降维。Ojala 等认为，在实际图像中，绝大多数 LBP 模式最多只包含两次从 1 到 0 或从 0 到 1 的跳变。因此，Ojala 将“等价模式”定义为：当某个局部二进制模式所对应的循环二进制数从 0 到 1 或从 1 到 0 最多有两次跳变时，该局部二进制模式所对应的二进制就成为一个等价模式类。如 00000000（0 次跳变），00000111（含一次从 0 到 1 的跳变和一次 1 到 0 的跳变），10001111（先由 1 跳到 0，再由 0 跳到 1，共两次跳变）都是等价模式类。除等价模式类以外的模式都归为另一类，称为混合模式类，例如 10010111（共四次跳变）。通过这样的改进，二进制模式的种类大大减少，模式数量由原来的 2^P 种减少为 $P(P-1)+2+1$ 种，其中 P 表示邻域集内的采样点数，等价模式类包含 $P(P-1)+2$ 种模式，混合模式类只有 1 种模式。对于 3×3 邻域内 8 个采样点来说，二进制模式由原始的 256 种减少为 59 种，这使得特征向量的维数更少，并且可以减少高频噪声带来的影响。

3.4 界面设计

1. 人脸采集界面

人脸采集界面有打开摄像头、开启人脸检测、初始化数据库、增加用户资料、开始采集人脸数据等按钮。打开摄像头负责获取视频，开启人脸检测后会框出人

脸的位置，初始化数据库会搜索数据库中有没有已保存的人脸的信息，增加用户当前信息会把用户的信息保存到数据库中，然后通过采集当前捕获帧，不停的采集当前人脸照片，最后存放到项目下的 dataset 文件中。这样人脸的数据就会保存好了。

2.人脸数据训练及信息管理界面

这个界面可以显示出当前数据库中人脸的数据信息，并对人脸数据进行训练，还可以选择图像的预处理，还可以从数据库中删除用户以及查询用户。

3.人脸识别和安全帽检测界面

这个界面主要负责的是检测功能，通过打开摄像头获取视频，初始化数据库来获取训练的 yml 文件，开启人脸检测后会对人脸进行检测，并且安全帽检测的线程也会启动，对传给安全帽线程的图片进行检测，然后返回给人脸线程，最后在 UI 界面的显示出来。

3.5 数据库设计

1.数据库需求分析

(1)在录入人脸数据之前，需要添加用户信息，此时我们需要一个表用来存储用户的基本信息，如工号(worker_id)，人脸 id(face_id)，姓名(cn_name)，姓名汉语拼音(en_name)，添加用户时间(created_time)。

(2)由于该系统需要记录用户的签到签退时间，所以第二个表用来记录相关数据，以便管理者可以轻松查询员工的考勤。记录到数据库的信息有姓名(name)，签到时间(in_time)，签退时间(out_time)，以及员工佩戴安全帽的状态(status)。

2.ER 图

对 ER 图中的关系解释如下：

- (1) 员工基本信息包含工号、训练状态、姓名、姓名汉语拼音信息。
- (2) 员工考勤信息包含姓名、安全帽的佩戴情况、上下班时间。
- (3) 员工基本信息和员工考勤信息之间相互联系。

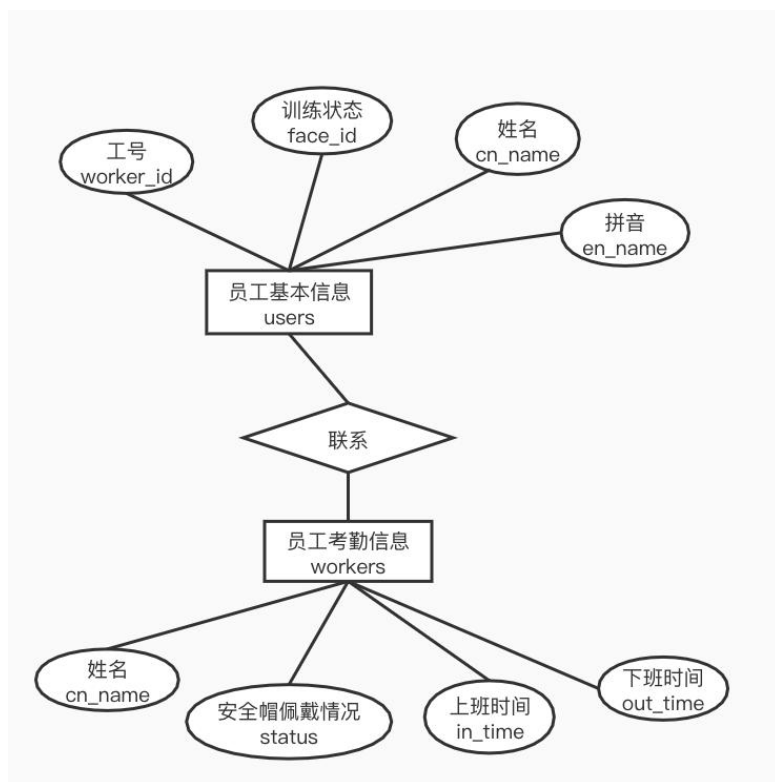


图 3-3 数据库 ER 图

3.数据库实现

sqlite3 是开源的 SQL 数据库管理系统，开源，意味着任何人都可以使用和修改该软件，任何人都可以使用和下载 sqlite3 而不需要支付任何费用，可以节省项目的开发经费，因此，sqlite3 目前是很受欢迎的一类数据库管理系统。主要用于数据量不是很大的小项目的开发。并且 sqlite3 具有多线程、快速的、多用户和健壮的特点，是受到很多开发人员喜爱的 SQL 数据库服务器。

(1) users 表，用来存储用户的信息记录，用户填好用户信息之后点击保存到数据库就是保存到此数据库中，如表 1。

表 1 users 表

序号	字段	类型	大小	必填字段
1	worker_id	VARCHAR	12	是
2	face_id	INTEGER	12	是
3	cn_name	VARCHAR	10	是
4	en_name	VARVHAR	16	是
5	created_time	DATE	12	是

(2) workers 表，用来存放用户签到以及签退时的信息以及佩戴安全帽的状态，如表 2。

表 2 workers 表

序号	字段	类型	大小	必填字段
1	name	integer	100	是
2	in_time	integer	100	是
3	out_name	integer	100	是
4	status	test	100	是

第四章 测试报告

4.1 测试速度

在 macnook pro 13inch(7 代 i5)的电脑上：人脸识别可以实时显示，速度比较快。但是由于缺少 GPU 的加持，在进行安全帽检测的时候速度 FPS 并不理想，实验最小的训练权重文化进行检测，检测一张图片一个人脸最快可以达到 0.5s。使用最大的权重文件的话，一张照片识别的速度大概在 3-5 秒之间。相信后期有 GPU 加速的话，使用较大的训练权重文件可以达到实时显示的效果。

4.2 准确度

在 macnook pro 13inch(7 代 i5)的电脑上：使用最大的权重文件，通过测试几千张的照片识别的准确度在 90%以上。图片中人脸或者安全帽有遮挡的话准确度在 70%以上。最差的情况，照片模糊，遮挡严重的情况下准确度维持在 30%左右。

第五章 使用流程

1、首先运行 dataRecord.py 文件采集人脸数据

(1) 打开摄像头（也可使用外接摄像头）->开启人脸检测->初始化数据库->增加用户/修改用户资料，如图 5-1 所示。

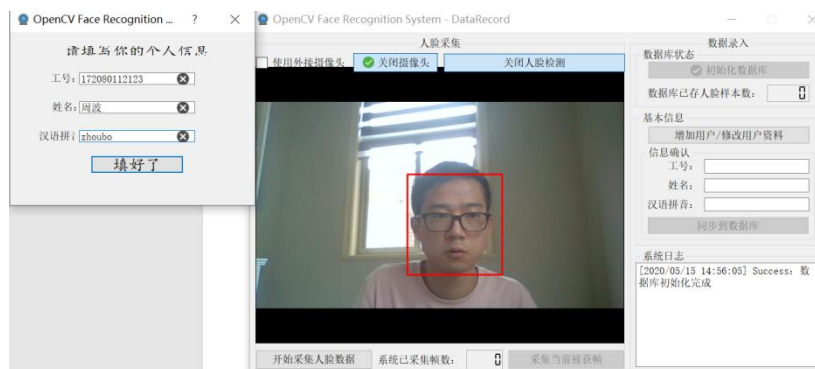


图 5-1 用户信息采集界面 1

(2) 开启当前人脸采集（右下角采集当前捕获帧越多，精度越高），如图 5-2 所示。



图 5-2 用户信息采集界面 2

(3) 结束当前人脸采集，并同步到数据库，如图 5-3 所示。

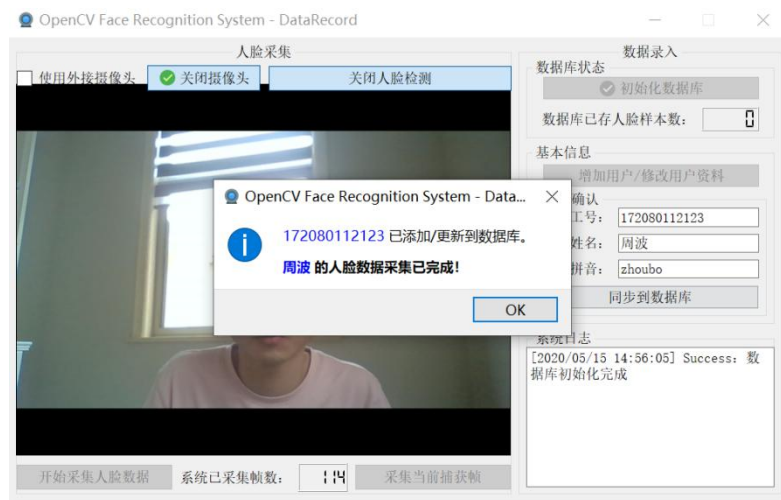


图 5-3 用户信息采集界面 3

2、接着运行 DataManage.py

(1) 训练人脸数据：（人脸训练前 Face ID 为-1，训练后 Face ID 为 1），如图 5-4 所示。

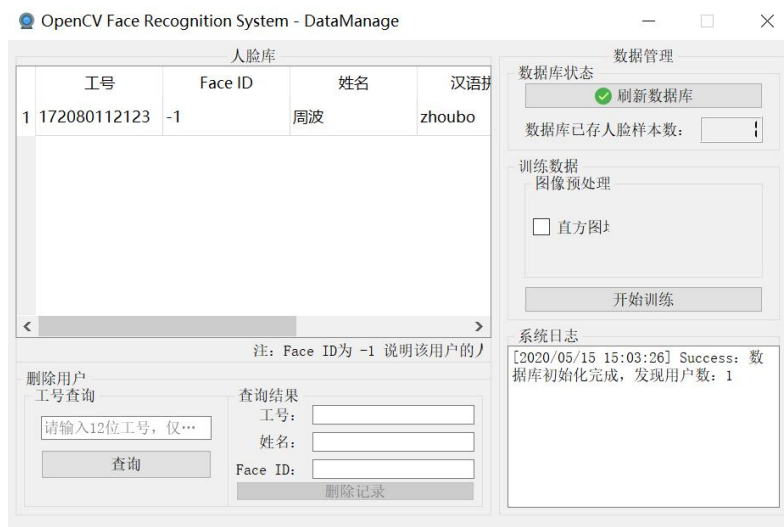


图 5-4 用户信息管理界面 1

(2) 点击直方图均衡化，然后点击开始训练，过一段时间后将弹出对话框提示人脸数据选练完成，如图 5-5 所示。

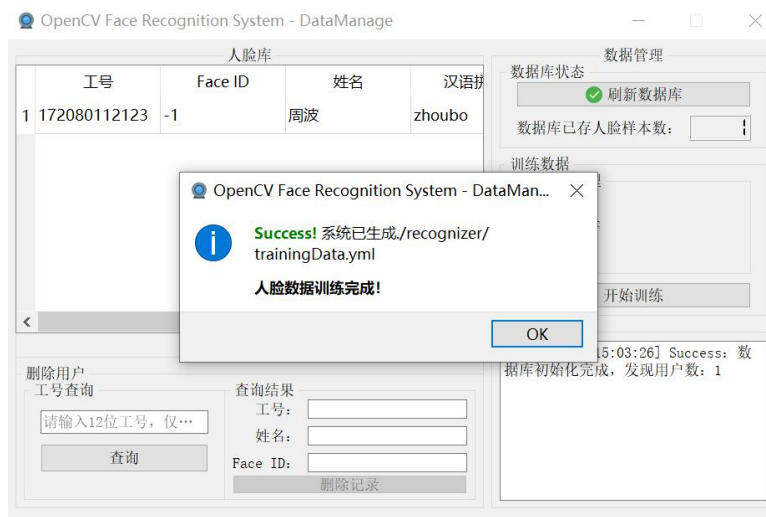


图 5-5 用户信息管理界面 2

(3) 在这个数据管理界面可以删除用户，先查询用户所对应的工号，然后点击 Yes 就可以删除记录，如图 5-6 所示。

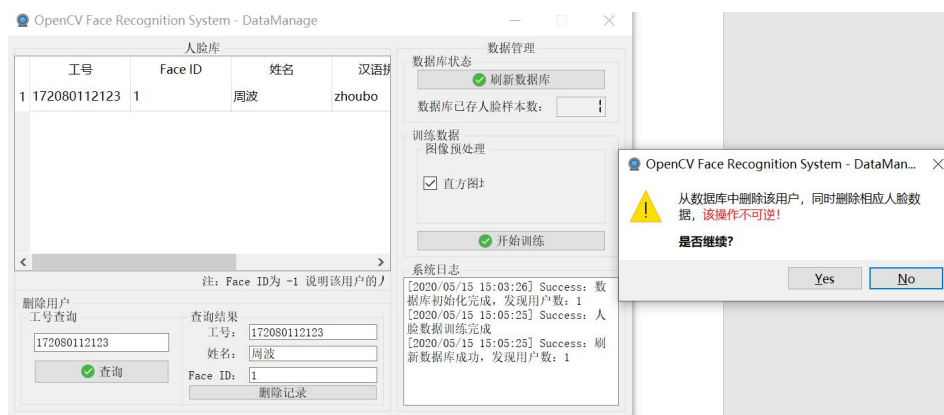


图 5-6 用户信息管理界面 3

3、运行 origin.py 文件即可进行人脸识别和安全帽检测，如图 5-7 所示。

- (1) 打开摄像头->初始化数据库->人脸识别
- (2) 将人的名字和佩戴安全帽的状态(person/hat)展示出来
- (3) 并且将签到签退、姓名、person/hat 等信息保存到数据库

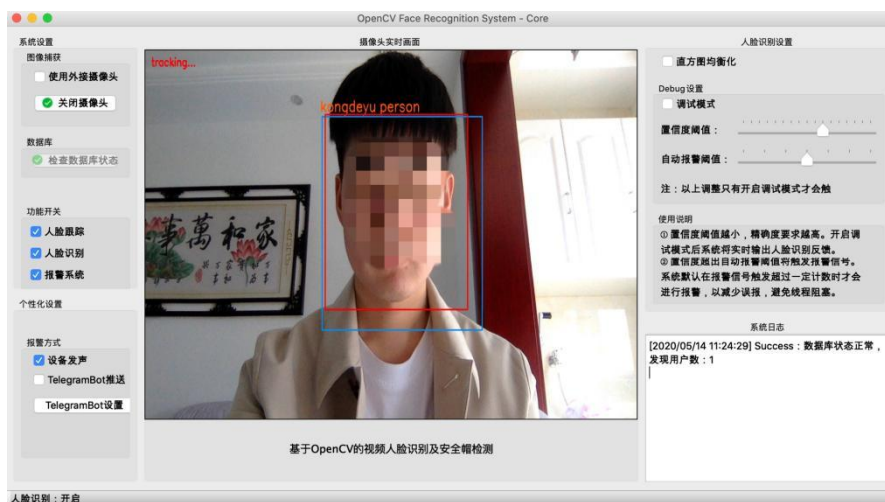


图 5-7 主界面

4、在车间内运行 workshop.py 文件，对车间内工人进行安全帽检测，此时无需进行人脸检测。

图 5-8 中可以看到，佩戴安全帽的员工没有框出来，框出来的是哪些未佩戴安全帽的员工，并且上面显示出了 person 关键字用来说明此员工没有佩戴安全帽。



图 5-8 安全帽检测界面

第六章 项目总结

6.1 项目协调

组建队伍:	12月1日-12月15日(第一阶段)
需求分析环境搭配:	12月15日-1月5日(第二阶段)
学习所需知识:	1月15日-2月15日(第三阶段)
项目设计与实现:	2月15日-4月15日(第四阶段)
功能测试与改进:	4月15日-4月25日(第五阶段)
编写文档、录制视频:	4月25日-5月15日(第六阶段)

(1) 在整个开发过程中,当我们各自遇到一些问题,我们便会通过qq分享屏幕(因为疫情原因)的方式来互相帮忙debug或提出建设性的意见;而当遇到思考了很长时间无法解决的问题时,我们则会在qq群里通过分享屏幕的方式与指导老师进行探讨。

(2) 队员负责与队长的沟通,队长负责与指导老师的沟通并传达指导老师的修改意见,开发效率较高。

6.2 水平提升

(1) 通过这次作品的制作开发,我们团队逐步理解了软件开发的合作方式,掌握了软件开发的一些设计模式,编程能力和调试能力都有了较大的提高。

(2) 很多时候只有在实际开发过程中才会明白需要学习哪些技术栈,以及自己哪些方面的不足然后针对性地学习与改进。

(3) 在开发之前明确具体开发需求,有利于后期的开发;而如果开发之前不是很明确的话,会走很多弯路,开发效率低下。

(4) 熟悉并了数据库的 curd 操作以及实时人脸检测时对数据库的插入与更新。

(5) 在开发过程中,曾经遇到过线程之间通信的问题,当时我们对于线程的概念并没有了解,自己看了一些资料之后觉得还是有一点迷糊,于是请教了老

师，老师也很耐心的给我们写了一个例子，我们仔细研究了例子，最终在老师的指导下完成了线程之间的通信问题。

6.3 升级演进与商业推广

(1) 继续对本项目进行迭代优化，时机成熟将会放到 github 上，我们团队坚信在求学路上，对于程序员而言，最重要的精神是分享。互相分享，互相讨论才能更好地提高自己；我们也坚信将一个产品不断地迭代才能更好地吃透产品的本质，更加熟练地进行软件开发。

(2) 我们打算之后再加一些其他生产安全方面的目标检测，比如有的生产作业环境需要戴吊带，有的需要戴手套，我们可以通过多线程的方式进一步糅合这些生产作业环境所需要的安全检测，更好地为工人们和企业管理者们服务，保障大家的安全。

(3) 因为是在校学生，相比其他企业成本比较低，可以先通过价格优势打开市场。

(4) 通过为有不同需求的企业制定个性化解决方案，从而打开市场。