

第十五届中国研究生电子设计竞赛

技术论文

论文题目：

口罩佩戴检测与人脸识别系统

Mask Wearing Detection And Face Recognition
System

参赛单位：上海大学

队伍名称：奇迹再现

指导老师：朱秋煜

参赛队员：郑国辉、栾剑兵

完成时间：2020 年 7 月 19 日

摘要

2020 年初，突如其来的新冠疫情打破了人们的平静生活，口罩成为了人们生活的必需品，佩戴口罩既是保护自己的生命健康安全也是对于他人的安全负责，特别是在公共场所，佩戴口罩已经变成每一位公民的责任与义务。因此，需要在各公共场所限定未佩戴口罩人群进入一系列公共场所。最为简单的一种方式就是采取人工监督的方式，但是大部分公共场所需要监督的区域可能比较多样化，在公共场所采用人工监督的方式可能会消耗较多的人力资源，并且一旦发生特殊情况，很难在最短的时间内循迹到源头。随着国内疫情趋势走向平稳，全国性复工复产复学工作的稳步推进，如何督促人们在进入公众场合时佩戴好口罩是一个需要解决的问题。据此，我们队伍设计并实现一款用于检测人脸是否佩戴口罩，并提醒未佩戴口罩人员佩戴口罩的软件，应用场景集中于学校公司等进出人员确定的场景，对每一位进入此场景的人员核实其是否佩戴口罩，并对未佩戴口罩人员做人脸识别，提醒其佩戴口罩。本软件系统基于深度神经网络实现人脸识别及口罩佩戴检测。

本次设计的口罩佩戴检测与人脸识别系统使用 VS2015 集成开发环境，使用 C++ 编程语言实现，系统界面使用 Qt 组件设计，图像和模型数据加载使用计算机视觉库 OpenCV 实现，核心算法口罩佩戴检测模块使用自设计的深度神经网络训练得到，使用的是 Caffe 深度学习框架实现，人脸识别基于开源人脸识别库 SeetaFace 改进设计实现，基于 MySQL 设计数据库用以记录检测与识别的结果信息。该口罩佩戴检测与识别系统设计能够实现检测多人场景下的人群佩戴口罩状况，拟应用于人场景如学校、机关单位和公司等系列场所入口，实时监测口罩佩戴情况并将信息显示到主界面上，条件允许的情况下可外配语音模块进行报警，此外，系统还具备有统计显示过去具体一段时间内的报警次数信息以便于后续开发中的数据处理等。

关键字：神经网络; SeetaFace; miniSSD; 口罩佩戴检测; 多人场景

Abstract

At the beginning of 2020, the sudden new crown epidemic broke people's peaceful lives. Masks have become a necessity for people's lives. Wearing masks is not only to protect your own life, health and safety, but also to be responsible for the safety of others. Especially in public places, wearing masks has become a necessity. The responsibilities and duties of a citizen. Therefore, it is necessary to restrict people who do not wear masks to a series of public places in various public places. The easiest way is to adopt manual supervision, but most public places need to be supervised in areas that may be diversified. Using manual supervision in public places may consume more human resources, and once special circumstances occur, It is difficult to trace to the source in the shortest time. As the domestic epidemic trend stabilizes, and the national resumption of work, production and school is progressing steadily, how to urge people to wear masks when entering public places is a problem that needs to be solved. Based on this, our team designed and implemented a software that detects whether a face is wearing a mask and reminds those who are not wearing a mask. The application scenarios are concentrated in the scene determined by the school company and other personnel entering and exiting. The personnel verify whether they are wearing masks, and perform face recognition for those who do not wear masks, and remind them to wear masks. This software system is based on a deep neural network to realize face recognition and mask wearing detection.

The mask wearing detection and face recognition system designed this time uses the VS2015 integrated development environment and the C++ programming language. The system interface is designed with Qt components. The image and model data are loaded using the computer vision library OpenCV. The core algorithm mask wearing detection module is used. The self-designed deep neural network is trained and implemented using the Caffe deep learning framework. The face recognition is based on the improved design and implementation of the open source face recognition library SeetaFace, and the database is designed based on MySQL to record the result information of detection and recognition. The mask wearing detection and recognition system is designed to detect the status of people wearing masks in multi-person scenarios. It is intended to be applied to the entrances of a series of places such as schools, institutions and companies in human scenarios to monitor the mask wearing status in real time and display the information on the main

interface. Above, when conditions permit, an external voice module can be equipped to give an alarm. In addition, the system also has statistics showing the number of alarms in a specific period of time in the past to facilitate data processing in subsequent development.

Key words: Neural network; SeetaFace; miniSSD; mask wearing detection; multiplayer scene

目录

第 1 章 简介	1
1.1 引言	1
1.2 系统架构介绍	1
第 2 章 方案设计	3
2.1 系统方案设计	3
2.1.1 系统开发平台	3
2.1.2 口罩佩戴检测方案设计	3
2.1.3 人脸识别方案设计	3
第 3 章 核心检测算法原理分析	5
3.1 口罩佩戴检测原理分析	5
3.1.1 核心人脸检测算法 MINISSD	5
3.1.2 实验效果	7
3.2 人脸识别原理分析	8
3.2.1 人脸检测	8
3.2.2 面部关键点定位	9
3.2.3 人脸特征提取与比对	10
第 4 章 软件设计与流程	12
4.1 软件界面设计与功能介绍	12
4.1.1 主界面	12
4.1.2 注册子界面	13
4.1.3 信息查询子界面	13
4.2 软件运作流程介绍	14
第 5 章 系统测试与误差分析	16
5.1 系统测试与误差分析	16
5.1.1 系统测试	16
5.1.2 人脸识别测试	16

第 6 章 总结.....	18
参考文献.....	19

第 1 章 简介

1.1 引言

疫情特殊时期在某些公共场合比如火车站和药店等，又比如公司工作集聚地点，在这些人员密集的公共场所，佩戴口罩不仅是保护自己的身体安全，也是为他人的健康负责，采用人工监督是一种可行的方式，但是当人流量达到一定程度之后，人工监督的方式就可能显得效率较低，为了提高效率那么就需要投入大量的人力资源，并且一旦发生异常情况，想要找到引起异常情况的源头目标，效率就更为低下。在这样一个情况下，使用配备有摄像头的计算机去代替人工执行此类任务不失为一种良性的解决方案。

计算机接收到摄像头所捕获的每一帧图像之后，对其中的人脸区域进行实时检测判别，判断该对象是否有佩戴口罩的行为，同时判断该对象是否是系统中已注册的用户，如果存在于已注册系统信息中，那么同时判断返回的还包括该用户的姓名信息，否则判断返回为陌生人对象，检测判断完毕后保存每一次的检测信息到本地，或者通过通信的方式上传到另一台计算机进行结果及信息存储，有了这些信息之后，也会方便后续统计或者发生任何异常情况，能够快速找到该事情发生的源头在何处，从而针对目标拟定采取一系列的有效措施。

1.2 系统架构介绍

本次设计实现的口罩检测与识别系统，主要是通过对摄像头捕获的视频流进行处理，对捕获到的每一帧图像检测人脸区域后进行口罩检测和人脸识别操作，口罩检测部分主要检测对象佩戴口罩的状况，人脸识别部分主要判断该对象是否是当前系统中已登记注册人员。

本系统采用 C/S 架构实现的模式，强调实时性和可靠性，使用 C++ 与 Python 语言进行开发。软件系统总共包括主界面、注册子界面、信息查询子界面三个界面以及 MySQL 数据库和用于存储人脸特征的 CSV 格式文件，其系统架构示意如图 1-1 所示。

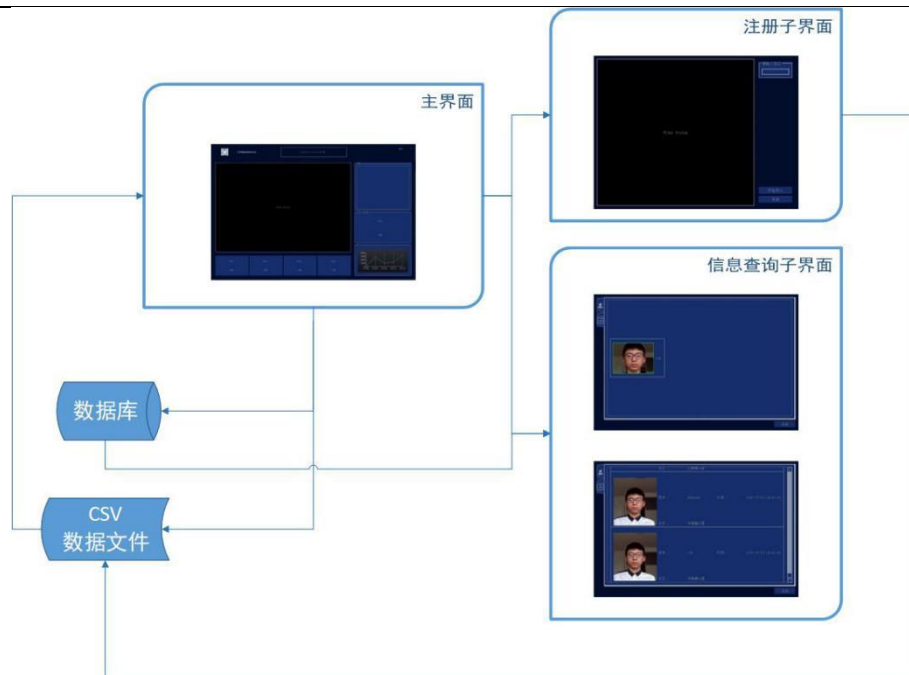


图 1-1 系统架构示意图

关于主界面、注册子界面以及信息查询子界面功能介绍将在第 4 章——软件设计与流程中具体介绍，这里不再过多赘述，由上述系统架构图可知，主界面与注册子界面以及信息查询子界面构成本系统的主要组成部分，信息主要存储在 MySQL 数据库以及 CSV 格式文件中，各界面通过访问 MySQL 数据库或者 CSV 格式文件数据记录进行查询操作或者取出特征及数据进行处理。

第2章 方案设计

2.1 系统方案设计

2.1.1 系统开发平台

本系统在 Windows 10 操作系统的环境下运行，以 Microsoft Visual Studio 为开发平台，开发语言采用 C++，数据库采用开源 MySQL 数据库，软件系统与数据库进行通信，存储以及修改数据。Visual Studio 是一种基于 Windows 平台应用程序集成开发环境，是微软公司的开发工具包系列产品。

2.1.2 口罩佩戴检测方案设计

深度学习是实现人工智能的方式之一，深度学习应用场景之一为目标检测，口罩佩戴检测可以看为一个目标检测问题，即检测佩戴口罩和未佩戴口罩两类。设计一个有能力检测图中人物是否佩戴口罩的深度神经网络并可以用于工程实际中是本系统的难点及创新点。

基于深度神经网络的目标检测算法有 RCNN, faster-RCNN, SSD 和 YOLO 等，这些神经网络模型都可以通过学习完成目标检测，在检测效果上 SSD 在速度上要优于其算法，这是由于 SSD 模型在结构上的优势获得的。随着深度学习在科研界中蓬勃发展，它也渐渐的得到了工业界的青睐，但是将学术界中研究的深度学习模型运用到工业界中，需要解决的一大难点就是实时性问题，工业界要去的更多的是实用性，如目标检测若达不到实时性的检测速度，那么在工业界中就很难得到大范围的应用。这也是为什么会出现越来越多的轻量化的深度神经网络，本系统实现口罩佩戴检测的神经网络是在 SSD 模型的基础上简化设计，设计了一款用于检测人脸是否佩戴口罩的轻量化的神经网络模型。

在算法实现上，我们先使用 PyTorch 搭建网络，由于后续需要通过 C++调用算法，所以我们将使用 PyTorch 搭建并训练好的模型转化为 Caffe 模型，这样 C++可以利用 OpenCV 来调用 Caffe 模型，这样做的一大好处是方便了模型搭建和训练的过程，并且 Caffe 在速度上优于 PyTorch，这也是为什么工程实际中最常用的是 Caffe。

2.1.3 人脸识别方案设计

人脸识别在日常生活中一直有着广泛的应用场景，优秀的算法模型可以很好地提升在实际应用场景中人脸识别的速度以及降低人脸识别的错误率，本系统中采用的是 SeetaFace 人脸识别库来完成人脸识别的操作。

SeetaFace 人脸识别算法是由中国科学院计算技术研究所视觉信息处理与学习(VIPL)组开源的一款商用级人脸识别算法,不依赖于任何第三方库,其中包括人脸检测模块 Face Detector、面部关键点定位模块 Face Landmarker 以及人脸特征提取与比对模块 Face Recognizer,未来将会陆续开源人脸跟踪、闭眼检测等辅助模块。

SeetaFace Detection 实现了用于实时多视图面部检测的 FuSt^[1]级联架构,从而在检测精度和速度之间有了良好的折中; SeetaFace Alignment 将一些堆叠式自动编码器网络(SAN)层叠起来,以逐步逼近面部轮廓的准确位置; SeetaFace Identification 是 AlexNet CNN 的改进版,用于面部特征识别,在 LFW 上的平均准确率为 97.1%^①。

^① <https://github.com/seetaface/SeetaFaceEngine/tree/master/FaceIdentification>

第3章 核心检测算法原理分析

3.1 口罩佩戴检测原理分析

根据 SSD 目标检测的原理，自设计了一款轻量化的神经网络模型。依据为 SSD 网络层数深，可以学习到图片中非常多的特征信息用于多种类的目标检测，而本系统仅仅需要检测人脸及佩戴口罩的人脸两个类别，所以一款轻量级的网络同样可以完成，在准确度上不差于更复杂的网络，同时在检测速度上更加具有优势，下面用 miniSSD 代指此网络。

3.1.1 核心人脸检测算法 miniSSD

SSD 算法核心观点就是通过不同的大小深度的特征图来检测不同大小的目标，深度较浅的特征图用于检测较小的目标，深度较深的特征图用于检测较大的目标，这使得 SSD 在检测小目标时候优于其他的算法，同时 SSD 借鉴 Faster-RCNN 算法设置默认检测框，通过深度学习的结果对默认检测框做回归分析，这使得 SSD 具备了速度上的优势。

本系统的核心算法之一即为目标检测算法，通过目标检测算法检测人脸是否佩戴有口罩。通过对比不同的目标检测算法，最终我们组选择了基于 SSD 算法，并做出一些改变实现人脸佩戴口罩检测。我们对 SSD 做的改变主要是在网络层数和复杂度上，由于 SSD 网络层数深，复杂度高，这对运行此深度学习算法的机器提出了很高的要求，为了降低硬件成本提高算法的通用性，我们组对 SSD 算法做了一些简化，提出了一种轻量级的目标检测算法，称之为 miniSSD。

我们的 miniSSD 的主干网络只有 8 层用于提取图片中的特征，有 5 个定位和分类层，其中 5 个用于定位和分类的特征图大小为 $38*38, 19*19, 10*10, 5*5, 3*3$ ，输入的图片大小为 $300*300$ ，整个模型的总参数数量不超过 150 万个，是一款轻量级的神经网络模型。模型的详细结构如图 3-1 所示。使用的每一个卷积层的卷积通道数在 32、64 和 128 之间，5 个用于分类和定位的特征图的每一个点配有 4 个默认检测框，共有 7756 个默认检测框。通过与 SSD 模型对比，我们的 miniSSD 在特征提取网络上比前者使用 VGG 结构要简化很多，默认检测框数量也比 SSD 少许多，所以不论从模型复杂度上还是简约性上来看，我们 miniSSD 都是有优势的。当然轻量级网络在算法稳定性上存在弱点，不过考虑到我们的 miniSSD 算法仅仅用于检测人脸和佩戴口罩的人脸两类，所以轻量级的网络在性能上不会弱于重量级的网络太多，在后面的

实验中也将会证明此点。

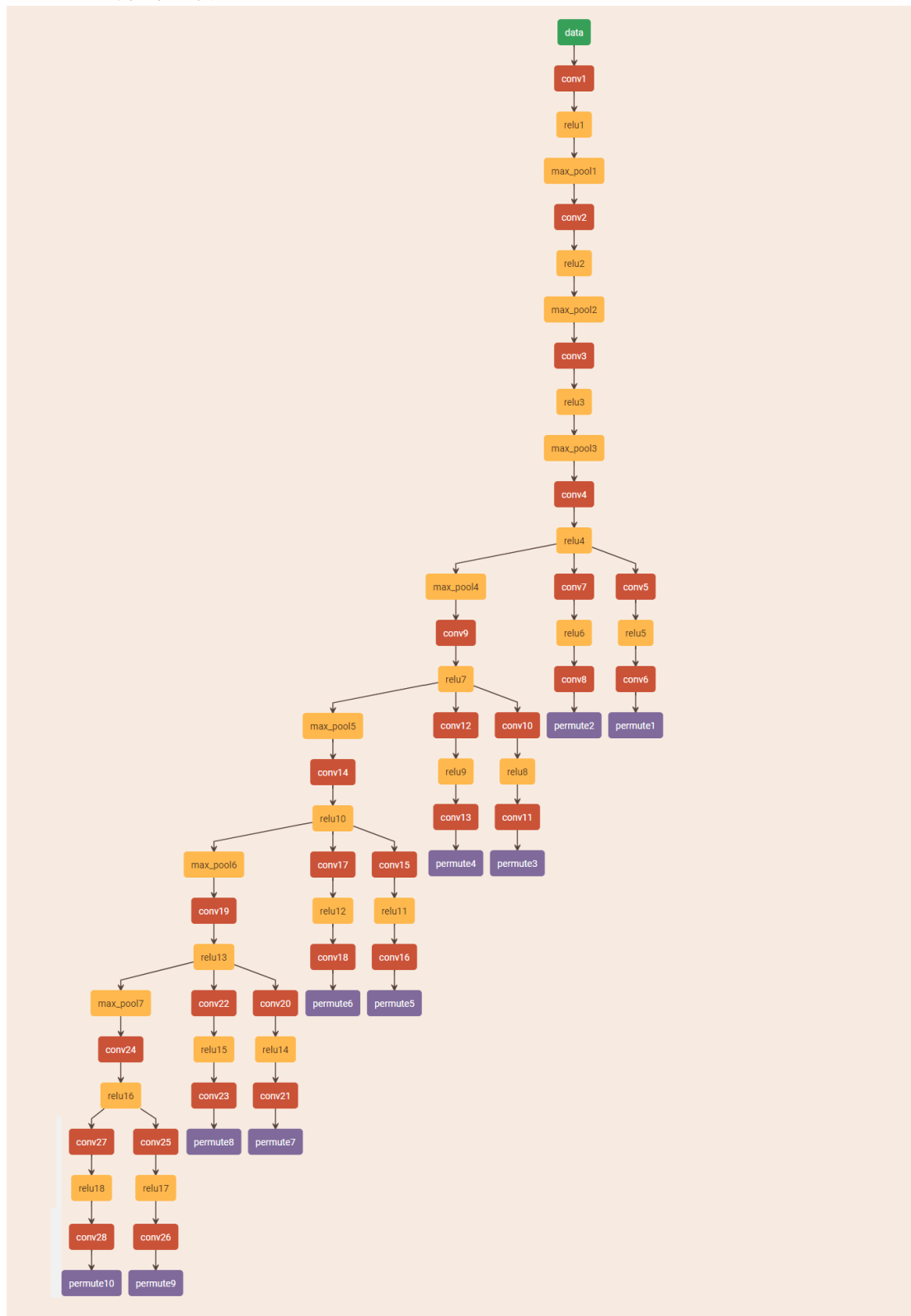


图 3-1 miniSSD 网络结构图

3.1.2 实验效果

首先是选择合适的数据集，人脸检测数据集非常多，其中最常用的莫过于 WIDER Face 数据集，我们从中选择了 3894 张图片，并进行了校验，主要是将部分戴口罩的人脸标注为戴口罩。对于戴口罩的人脸，我们使用了中科院信工所葛仕明老师开源的 MAFA 数据集，该数据集本是一个遮挡人脸的数据集，其中有各种被遮挡的人脸图片，大多数都是被口罩遮挡的人脸图片，我们从中选择了 4064 张人脸戴口罩的图片。我们利用 PyTorch 搭建神经网络，通过对比 SSD 和本系统简化得到 miniSSD 在人脸检测和口罩佩戴检测的结果来证明我们 miniSSD 在性能上不差于 SSD 算法，我们将 PyTorch 搭建得 SSD 和 miniSSD 在同样条件下训练，其中唯一的差别在于训练 SSD 时候我们设置的 epoch 较大，这是因为 SSD 网络层数较深，需要训练久一些。下面通过几张具有代表性的图片比较二者。



图 3-2 miniSSD 与 SSD 检测效果对比（左边为 SSD 右边为 miniSSD）

图 3-2 选取了几张具有代表性的 miniSSD 和 SSD 检测效果对比图，可以看到在同样的条件下，miniSSD 检测效果不差于 SSD，二者的差别仅仅是 SSD 检

测精度高一些，但是这种差别对实际使用不会造成什么影响。相反，在检测速度上 miniSSD 的速度要绝对快于 SSD 算法，在我们组自己电脑上运行时，miniSSD 检测一张图片的平均时间为 200ms 左右，而 SSD 则需要将近 2s。而将 PyTorch 模型转化为 Caffe 之后，用 C++调用 miniSSD 算法，检测速度会进一步缩小到 30ms 左右，说明我们的 miniSSD 模型是一款对硬件要求低，实用性广的模型。

3.2 人脸识别原理分析

本口罩佩戴检测与识别系统中使用到了人脸识别的功能部分，由中国科学院计算技术研究所视觉信息处理与学习(VIPL)组开源的一款商用级人脸识别算法 SeetaFace 人脸识别库，该人脸识别库现已开源的三个模块包括人脸检测、面部关键点定位和人脸特征提取与比对。

3.2.1 人脸检测

人脸检测模型使用的是 Cascade CNN^[5]，其测试流程如图 3-3 所示，该模型将人脸检测分为三个阶段，即进行三次检测。第一级 12-net 通过多尺度滑窗暴力扫描，过滤大量窗口，剩下的窗口通过 12-calibration-net 来进一步调整 face bounding box 的位置和大小，并使用 NMS(非极大值抑制)消除 IOU 高度重合的 bounding box；第二级 24-net 输入为第一级检测样本，进一步过滤后再接 24-calibration-net+NMS 进一步校准消除重合的 bounding box；第三级输入为第二级的检测样本，作进一步人脸筛选，这一步之后优先使用 NMS 来消除重合 bounding box，再通过 48-calibration-net 校准，得出结果。

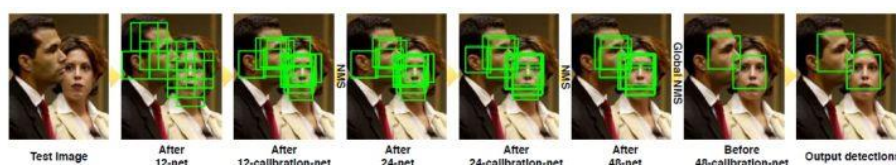


图 3-3 Cascade CNN 测试流程

SeetaFace 人脸检测模块基于一种结合经典级联结构和多层神经网络的人脸检测方法，采用漏斗型级联结构(Funnel-Structured Cascade, FuSt)，针对于多姿态人脸检测而设计，兼顾速度与精度平衡，如图 3-4 所示。

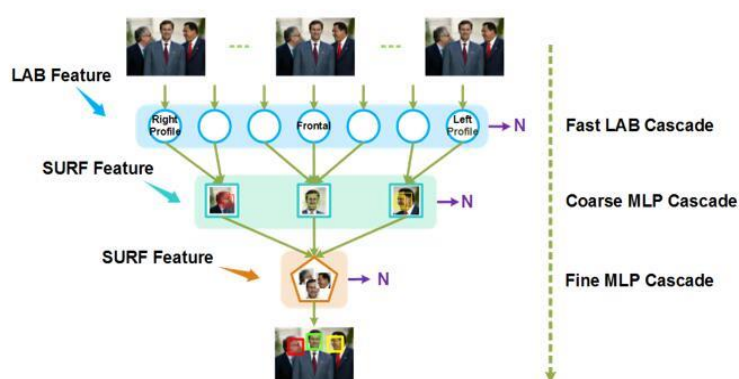


图 3-4 FuSt 漏斗型级联结构

由上图可见，该级联结构首先顶层为针对不同姿态的快速 LAB 级联分类器^[4]，接下来是基于 SURF 特征的 MLP 级联结构，最后由一个统一的 MLP 级联结构处理得到候选窗口。

SeetaFace 人脸检测相比于其他某些算法具有一定速度优势，表 3-1 是该检测模块与 Cascade CNN 和 FacenessNet 在某些条件下的速度对比。

表 3-1 SeetaFace 人脸检测算法速度对比^②

Detector	Min Face Size	Speed on CPU	Speed on GPU
SeetaFace Detector	40x40	16.0 FPS	n/a
SeetaFace Detector	80x80	55.0 FPS	n/a
Cascade CNN	80x80	9.1 FPS	100 FPS
FacenessNet	80x80	n/a	20 FPS

3.2.2 面部关键点定位

人脸关键点定位即人脸对齐在人脸识别、表情识别等人脸分析情况下扮演着非常重要的角色，这是由于抓拍图像会有姿态、光照和遮挡等因素的影响，因此在真实应用场景下必须进行面部关键点定位即人脸对齐操作。SeetaFace Alignment 人脸对齐采用的是一种由粗到精的自编码器网络(Coarse-to-Fine Auto-encoder Networks, CFAN)^[2]来实现这个过程，CFAN 级联了多级栈式自编码器网络，其中的每一级都刻画从人脸表观到人脸形状的部分非线性映射，如图 3-5 所示。

^② <https://github.com/seetaface/SeetaFaceEngine/tree/master/FaceDetection>

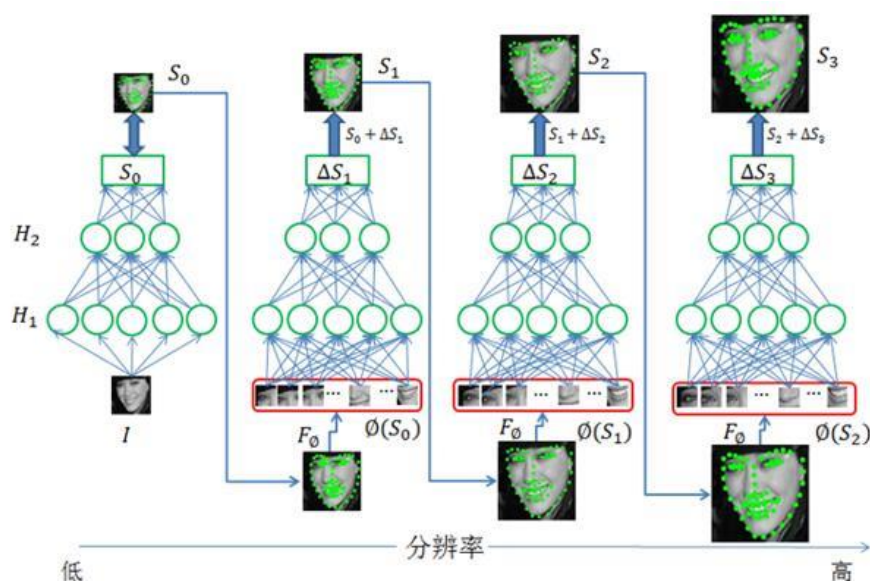


图 3-5 CFAN 人脸对齐方法

本口罩佩戴检测与识别系统采用的是 SeetaFace 基于 CFAN 实现面部 5 个关键特征点精确定位，即在人脸定位双眼、鼻子及嘴角这 5 个关键点，此外，在保持精度可观的情况下，SeetaFace 人脸对齐将 CFAN 级联数目减少到两级，能够在 CPU 上达到每个人脸 5ms 的速度^③。

3.2.3 人脸特征提取与比对

人脸识别的本质即计算两幅图像中人脸的相似程度，通常包括人脸特征提取与比对两个大块，前提是经过图像目标人脸检测与人脸特征点提取处理后。特征提取即将图像的人脸区域输入特征提取器，通过某种方式计算编码得到一个特征向量，本系统中即提取出 2048D 特征向量，继而以某种方式将两幅图像的特征向量进行对比，比如计算二者之间的距离或者计算二者之间的余弦相似度等，当这个距离或者相似度大于某个阈值或者小于某个阈值的时候就判定这两张图像所代表的目标对象是同一个人。

本系统中使用 SeetaFace 开源的人脸识别特征提取模块 SeetaFace Identification 也是基于深度卷积神经网络 DCNN，是 VIPLFaceNet^[3]的具体实现，VIPLFaceNet 由 7 个卷积层和 2 个完全链接层组成，可以说是 AlexNet 的改进版，与 2012 年提出的 AlexNet 相比，VIPLFaceNet 将 5x5 的卷积拆分为两层 3x3 的卷积操作，增加了该网络深度但是并没有增加计算量，此外，VIPLFaceNet 减少了每个卷积层的核的数目并引入 FNL，加速了该网络的收敛

^③ <https://github.com/seetaface/SeetaFaceEngine/tree/master/FaceAlignment>

速度，提高了模型的泛化能力。如图 3-6 所示为两个网络的结构对比图。

AlexNet	VIPLFaceNetFull	VIPLFaceNet
Conv1: 96x11x11, S:4, Pad:0	Conv1: 96x9x9, S:4, Pad:0	Conv1: 48x9x9, S:4, Pad:0
LRN	—	—
Pool1: 3x3,S:2	Pool1: 3x3,S:2	Pool1: 3x3,S:2
Conv2: 256x5x5, G:2, S:1, Pad:2	Conv2: 192x3x3,S:1, Pad:1	Conv2: 128x3x3,S:1, Pad:1
LRN	—	—
—	Conv3: 192x3x3,S:1, Pad:1	Conv3: 128x3x3,S:1, Pad:1
Pool2: 3x3,S:2	Pool2: 3x3,S:2	Pool2: 3x3,S:2
Conv3: 384x3x3, S:1, Pad:1	Conv4: 384x3x3,S:1, Pad:1	Conv4: 256x3x3,S:1, Pad:1
Conv4: 384x3x3, G:2, S:1, Pad:1	Conv5: 256x3x3,S:1, Pad:1	Conv5: 192x3x3,S:1, Pad:1
—	Conv6: 256x3x3,S:1, Pad:1	Conv6: 192x3x3,S:1, Pad:1
Conv5: 256x3x3, G:2, S:1, Pad:1	Conv7: 192x3x3,S:1, Pad:1	Conv7: 128x3x3,S:1, Pad:1
Pool3: 3x3,S:2	Pool3: 3x3,S:2	Pool3: 3x3,S:2
FC1, 4,096	FC1, 4,096	FC1, 4,096
Dropout1: dropout_ratio:0.5	Dropout1: dropout_ratio:0.5	Dropout1: dropout_ratio:0.5
FC2, 4,096	FC2, 2,048	FC2, 2,048
Dropout2: dropout_ratio:0.5	Dropout2: dropout_ratio:0.5	Dropout2: dropout_ratio:0.5
FC3, 10,575	FC3, 10,575	FC3, 10,575

图 3-6 VIPLFaceNet 与 AlexNet 网络结构对比^④

本口罩佩戴检测与识别系统中，包含一个用户注册功能，用户将人脸信息通过注册界面来将其人脸部分检测首先进行特征提取并保存至指定的 CSV 文件中。这样，在系统运行的之后，可以通过事先读取已经存储的特征向量这种方式，与实时转换的特征向量去比对，同时限定阈值，那么就可以快速比对出当前目标对象的人脸信息。

^④ Liu X , Kan M , Wanglong W U , et al. VIPLFaceNet: an open source deep face recognition SDK[J]. Frontiers of Computer ence, 2017, 11(002):208-218.

第4章 软件设计与流程

4.1 软件界面设计与功能介绍

本软件界面整体来说总共包括三个部分，分别是主界面、注册子界面以及信息查询子界面，通过主界面菜单按键实现与各个子界面的切换调用。

4.1.1 主界面

主界面主要实现对摄像头获取的实时图像处理结果以及信息显示，如图 4-1 所示。

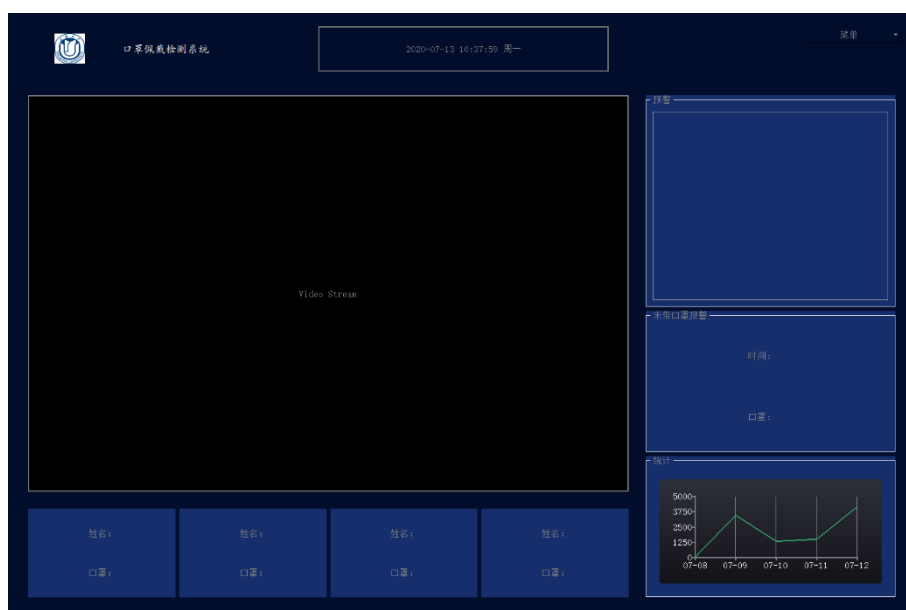


图 4-1 主界面

主界面正中央为实时获取的视频流，经过处理之后负责显示实时处理结果，主要包括图像中人群口罩佩戴检测显示以及人脸识别效果显示。

主界面正下方 4 个预置结果显示空位负责显示已注册人群的检测识别结果，包括抓拍头像、姓名和佩戴口罩结果。

主界面右侧预警负责显示监测过程中所有未佩戴口罩的情况，下方用于显示在监测过程中存在的陌生人未佩戴口罩的情况。

主界面右下角用于统计显示五天内未佩戴口罩的人数。

主界面菜单键负责启动本系统与关闭，并且提供有用户注册与信息查询两种功能动作，如图 4-2 所示。

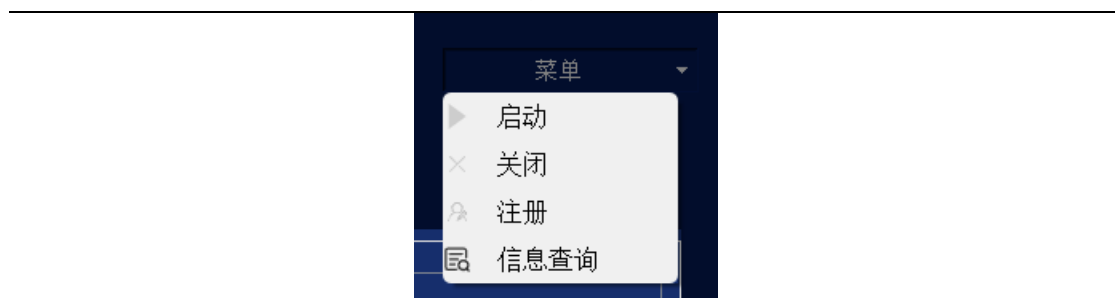


图 4-2 菜单功能显示

4.1.2 注册子界面

注册子界面可离线注册人脸信息，即检测提取用户人脸信息特征用于个人信息保存，如图 4-3 所示。

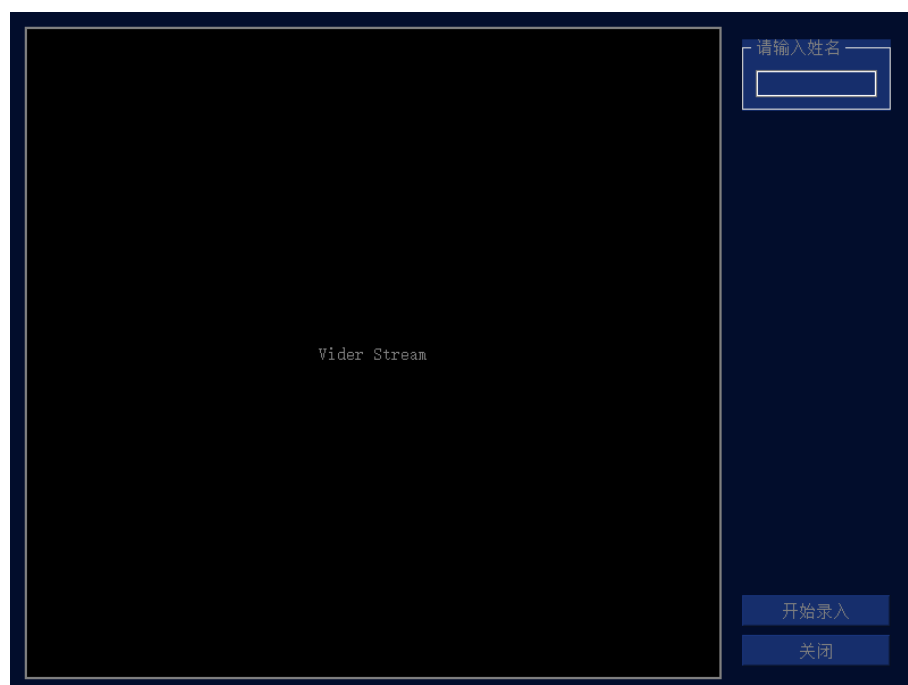


图 4-3 注册子界面

用户在输入姓名之后点击开始录入按键，开始录入人脸信息，提取人脸信息采取的是开源的 SeetaFace 识别引擎，将转换成的 2048 维特征向量以及姓名保存进入 CSV 格式文件中，用于在实时识别过程中的特征信息比较。

4.1.3 信息查询子界面

信息查询子界面主要是用于访问数据库中存储的监测信息记录及 CSV 文件中已注册用户信息并用于显示，图 4-4 所示为测试的用户注册信息查询显示界面效果，图 4-5 为测试的监测记录查询显示界面效果。

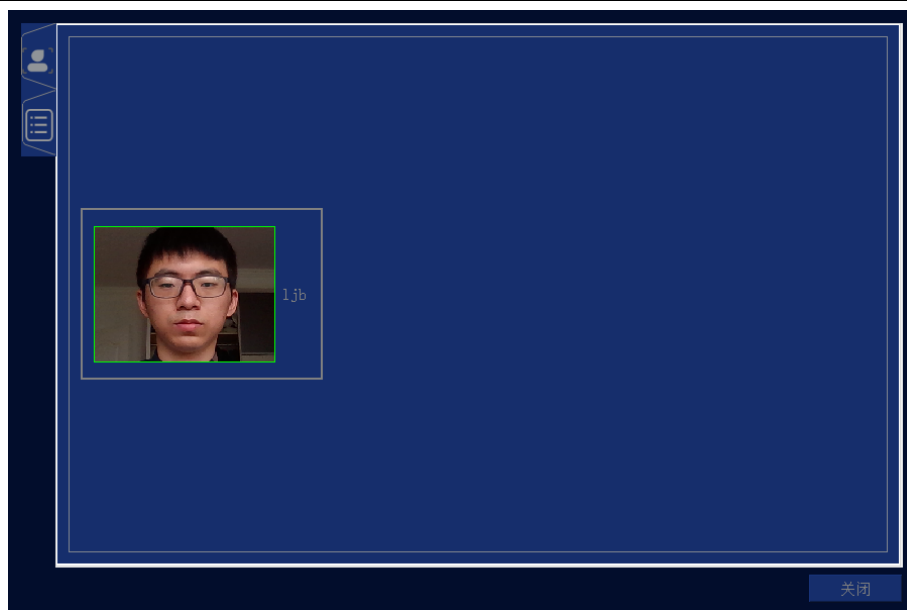


图 4-4 用户注册信息查询显示界面



图 4-5 监测记录查询显示界面

4.2 软件运作流程介绍

本次设计的口罩佩戴检测与识别系统基于 VS2015+Qt 实现，另通过 MySQL 数据库存储每一次的结果信息，存储的结果包括时间、姓名、佩戴口罩状况以及即时图像在本地保存地址，整个系统的软件运作逻辑流程示意如图 4-6 所示。

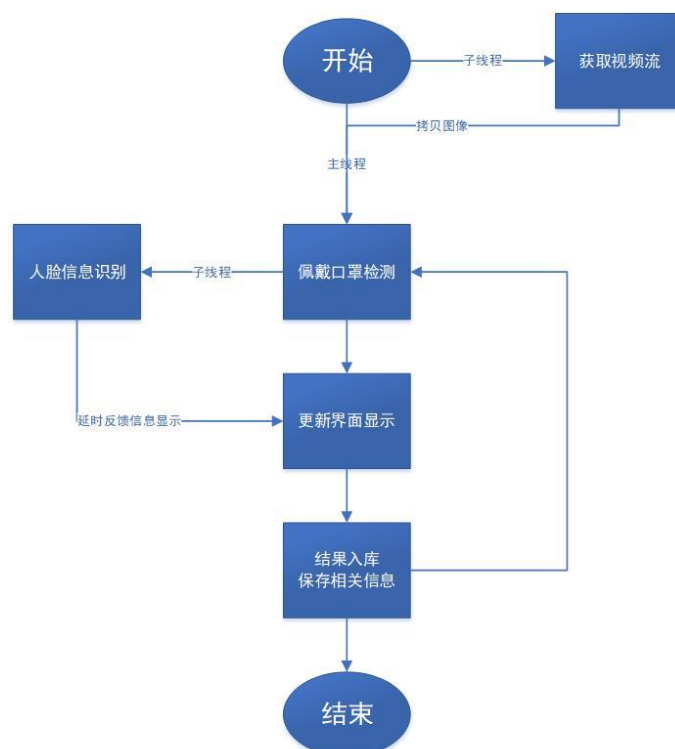


图 4-6 软件运作逻辑流程示意

由上图可见，系统开始运行存在两个线程，设定一子线程用于捕获视频流，主线程方面处理图像的时候从视频流中获取一副图像用作接下来的处理，对于该张图像，首先将其通过网络计算得到本次的识别区域坐标以及识别佩戴口罩状态，继而将该人脸区域送至 SeetaFace 进行人脸特征提取，与已注册过的人脸信息 CSV 文件中包含的特征进行比对，如果有则返回名称，否则不予返回姓名即无，在本段子操作过程中为了提高画面效果，故采用子流程非阻塞计算，因而在识别姓名方面会存在延迟，另外由于口罩遮挡也给识别带来了一些特征的丢失，所以在姓名比对方面准确率会降低。

在检测与识别之后更新到界面上显示检测结果，本次一个流程结束后将结果信息保存到数据库以及本地，便于后续其他查询操作。保存信息完毕之后将会开始一个新的计算流程，如果检测到停止动作的发出，则系统停止获取视频流并停止处理图像数据。

第 5 章 系统测试与误差分析

5.1 系统测试与误差分析

5.1.1 系统测试

本系统输入图片大小为 640×480 ，网络输入图像大小为 300×300 ，处理一幅图像时间约为 40ms，实际系统中采取每隔 10 帧图像处理一次来提高实时性，在 Release 环境下系统整体可以达到约 35FPS，图 5-1、图 5-2 为已注册用户系统测试示意图，图 5-3、图 5-4 为未注册用户系统测试示意图。

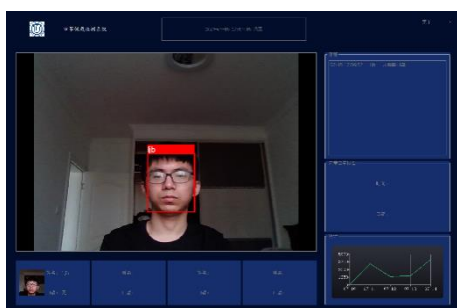


图 5-1 已注册未佩戴口罩系统测试

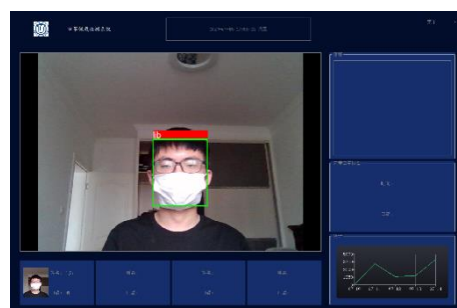


图 5-2 已注册佩戴口罩系统测试

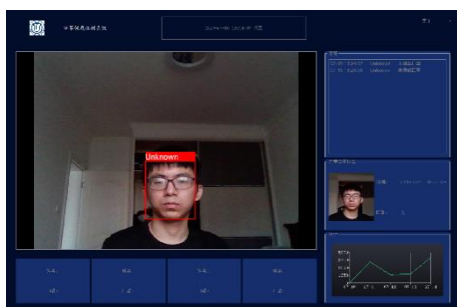


图 5-3 未注册未佩戴口罩系统测试

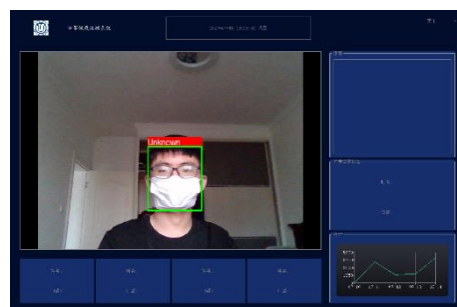


图 5-4 未注册佩戴口罩系统测试

5.1.2 人脸识别测试

本口罩佩戴检测与识别系统在进行人脸识别比对的时候计算出来的相似度较低，同一对象计算相似度约为 0.6，下表 5-1 为人脸识别功能部分在一定条件下实际测试准确率。

表 5-1 人脸识别功能部分实际测试

测试次数/次	测试准确率/百分比
100	78

针对上述情况，考虑到实际应用过程中，佩戴口罩会对人脸识别特征提取操作造成一定程度的影响，这也是因为所使用的 SeetaFace 人脸识别库版本中数

据集中并没有佩戴口罩的样本存在，由于时间问题，这个问题我们暂时并没有测试出很好的解决方法，同时，测试现场的环境如光照环境等也会对实际人脸特征提取造成一定的误差，这也是我们系统之后需要改进的部分。

第6章 总结

针对此次疫情风波，我们队伍期望能够针对监测公共场所内的人群佩戴口罩这一情况提出有效的解决方案，在条件有限的情况下，设计实现了一款基于深度学习的口罩佩戴检测与识别软件系统，我们对于其中主要功能口罩佩戴检测与识别采用了自训练的神经网络，佩戴口罩检测算法我们依据 SSD 模型提出了自己的 miniSSD，属于一种轻量级的算法，在实时性和精度上表现优良，人脸识别采用中国科学院计算技术研究所视觉信息处理与学习(VIPL)组开源的一款商用级人脸识别算法 SeetaFace，其余主要图像及数据处理采用 C++编程语言与 OpenCV 开源计算机视觉库实现。

该口罩佩戴检测与识别系统具备口罩佩戴检测、人脸识别、用户注册和信息查询等几个主要功能，能够满足离线脱机状况下的日常工作，不过依旧存在一部分问题有待解决，比如佩戴口罩情况下人脸识别准确率较低、识别不太理想的情况等。同时，该系统起初思想还可以额外配备语音插件及外设实现语音报警的功能以及实现基于热成像的非接触体温测量方式，不过由于条件所限暂时未实现，后续有机会在多种场景下不断实践改进，从而继续丰富完善额外功能以及提高系统的稳定性。

最主要的，我们队伍在此次软件系统搭建的过程中，提高了对于深度学习方法的认知、搜集资料、独立思考以及自我解决问题的能力，获得了综合素质上的提升，也深刻体会到了团队合作的重要性。

最后，感谢我们队伍指导老师朱秋煜老师的大力支持。

参考文献

- [1] Wu S , Kan M , He Z , et al. Funnel-Structured Cascade for Multi-View Face Detection with Alignment-Awareness[J]. 2016.
- [2] Zhang J , Shan S , Kan M , et al. Coarse-to-Fine Auto-Encoder Networks (CFAN) for Real-Time Face Alignment[C]// European Conference on Computer Vision. Springer, Cham, 2014.
- [3] Liu X , Kan M , Wanglong W U , et al. VIPLFaceNet: an open source deep face recognition SDK[J]. Frontiers of Computer ence, 2017, 11(002):208-218.
- [4] Shengye Yan, Shiguang Shan, Xilin Chen, Wen Gao. Locally Assembled Binary (LAB) Feature for Fast and Accurate Face Detection. IEEE Computer Society International Conference on Computer Vision and Pattern Recognition, CVPR2008, Anchorage, Alaska, U.S.A, Jun. 2008.
- [5] Li H , Lin Z , Shen X , et al. A convolutional neural network cascade for face detection[C]// Computer Vision & Pattern Recognition. IEEE, 2015.