

重庆工商大学

人工智能编程语言

课程设计说明书

设计题目： 基于 opencv 的人脸识别系统

学 院： 计算机科学与信息工程学院

专业班级： 电子信息工程 2021 级二班

学生姓名： 刘银杰 学号： 2021413080

同组姓名： 王科文 学号： 2021413094

同组姓名： 姚 双 学号： 2021413062

同组姓名： 张 平 学号： 2021413022

指导教师： 陈明杰 职称： 副教授

设计地点： 重庆工商大学

日 期： 2023 年 11 月 23 日---12 月 26 日

摘要

OpenCv 人脸识别是一种广泛应用于计算机视觉领域的识别技术，其核心是利用 OpenCv 库中的算法对人脸图像进行特征提取和比对，从而实现人脸的自动识别。

OpenCv 人脸识别技术具有以下几个关键步骤：人脸检测、特征提取和比对。在人脸检测阶段，通过一系列的算法和技术，将输入的图像中的人脸位置进行定位和识别。这一步骤的关键在于算法的准确性和速度，以保证在实际应用中能够实时地对人脸进行检测和定位。

在特征提取阶段，利用 OpenCv 中的算法和技术，对检测到的人脸图像进行特征提取，以获取人脸的特征信息。这一步骤的目的是为了将人脸图像转化为可进行比对的数字特征向量，从而方便后续的比对和识别。在比对阶段，将提取出的特征向量与预先存储的特征向量进行比对，以实现人脸的识别。这一步骤的关键在于比对算法的准确性和速度，以保证在实际应用中能够实时地完成人脸的识别任务。最后，在团队不懈努力下，我们的人脸识别系统不论单张人脸识别还是多张人脸识别，准确率达都到了 80%，并且实现了中文显示姓名并进行半加密的功能，这样对于用户来说能起到一定的保护隐私的作用。

OpenCv 人脸识别技术的应用非常广泛，例如在安全监控、人机交互、智能门禁等领域都有广泛的应用。随着人工智能技术的不断发展，OpenCv 人脸识别技术也将不断地优化和改进，实现更高准确率和更快速的人脸识别任务。

关键词：OpenCv 人脸识别技术 人工智能 特征提取

目 录

一、研究背景..... 1

二、方案选择..... 1

三、整体框架..... 3

四、主要模块介绍..... 4

五、测试过程..... 5

六、结果与分析..... 7

七、总结..... 7

八、致谢..... 8

九、心得与体验..... 8

参考文献..... 9

附录 1 完整设计框图

附录 2 完整源程序

附录 3 设计碰到的问题及解决办法/易错问题及解决办法

一、研究背景

在中国，基于 OpenCV 的人脸识别的研究主要涉及到以下几个方面：

人脸检测与识别：国内研究者基于 OpenCV 开发了一些人脸检测和识别系统，通过使用 OpenCV 中的人脸检测算法和识别算法，实现了人脸的自动识别和验证功能。

人脸表情识别：国内的研究者利用 OpenCV 中的图像处理和分类算法，开展了人脸表情识别的研究。他们利用 OpenCV 提供的人脸检测和关键点定位功能，结合机器学习算法，能够准确地识别出人脸表情。

人脸年龄和性别识别：国内的研究者利用 OpenCV 的人脸检测和关键点定位功能，结合机器学习算法，实现了人脸年龄和性别的准确识别。这种技术在视频监控、人脸身份验证等领域有着广泛的应用。

人脸活体检测：国内的研究者基于 OpenCV 开发了一些人脸活体检测系统，通过分析人脸的纹理、表情以及眼球运动等特征，判断人脸是否为真实的活体，从而提高人脸识别的安全性。

总而言之，人脸识别技术作为一种生物识别技术，具有非接触、非侵扰、直观、用户友好等优点，在身份认证、安全监控、社交网络等领域得到了广泛应用。近年来，随着深度学习技术的发展，人脸识别技术取得了显著的进步。国内外研究者提出了多种基于深度学习的人脸识别算法，并取得了很好的效果。然而，在实际应用中，仍存在一些挑战和问题需要解决，如光照变化、表情变化等。因此，本课程设计报告旨在探讨基于 OpenCV 的人脸识别系统的实现过程和性能评估。

二、方案选择

在方案选择方面，本课程设计报告采用了基于 OpenCV 的人脸识别系统。OpenCV 是一个开源的计算机视觉库，提供了丰富的人脸识别算法和工具。通过 OpenCV 库的 API 接口，可以方便地实现人脸检测、对齐、特征提取和比对等功能。此外，OpenCV 还提供了多种图像处理和计算机视觉算法，可以满足不同应用场景的需求。当然，在设计时除了可以使用 OpenCV，还可以使用其他的人脸识别库和工具，例如：

Dlib：Dlib 是一个强大的 C++ 库，提供了人脸检测和人脸关键点定位的功能，可以用于人脸识别任务。它的优点是准确性较高，但对于图像质量和角度变化比较敏感。

FaceX：FaceX 是一个基于深度学习的人脸识别库，提供了人脸检测、人脸特征提取和人脸比对等功能。它的优点是准确性较高，但可能需要较多的训练数据和计算资源。

TensorFlow：TensorFlow 是一个流行的深度学习框架，它提供了多种人脸识别模型和算法，可以用于构建自定义的人脸识别系统。它的优点是可以用于构建自定义的人脸识别系统，但需要一定的深度学习知识和技能。

PyTorch：PyTorch 是另一个流行的深度学习框架，也提供了多种人脸识别模型和算法，适用于构建人脸识别应用。它的优点是可以用于构建自定义的人脸识别系统，但需要一定的深度学习知识和技能。

MTCNN：MTCNN 是一种多任务级联卷积神经网络，可以实现人脸检测和关键点定位等任务，是一种常用的人脸识别方法。它的优点是准确性较高且速度较快，但可能对于小尺寸人脸检测效果不佳。

而 OpenCV 相较于上述几种库函数，它具有更好的应用性。OpenCV 是一个开源的计算机视觉库，提供了丰富的图像处理和计算机视觉算法。他的优点如下：

1、广泛的功能：OpenCV 提供了丰富的图像处理和计算机视觉算法，包括图像滤波、边缘检测、图像分割、特征提取等。它还支持人脸检测、目标跟踪、光流估计等高级功能。

2、跨平台性：OpenCV 可以在多个平台上运行，包括 Windows、Linux、macOS 和 Android 等。这使得开发者可以在不同的环境下使用相同的代码。

3、大量的社区支持：OpenCV 拥有庞大的开发者社区，提供了丰富的文档、示例代码和技术支持。开发者可以从社区中获取帮助和交流经验。

4、高效的图像处理：OpenCV 使用优化的算法和数据结构，能够高效地处理图像和视频数据。它还支持并行计算和硬件加速，可以在多核处理器和 GPU 上加速计算。

但是它也有一定的缺点：

1、学习曲线较陡：OpenCV 是一个功能强大的库，但也比较复杂。对于初学者来说，需要一定

的学习曲线和实践经验才能熟练使用。

2、文档不够完善：尽管 OpenCV 有大量的文档和示例代码，但有时候可能会遇到文档不够详细或不准确的情况。开发者可能需要花费额外的时间来解决问题。

综上所述，OpenCV 是一个功能强大的计算机视觉库，具有广泛的应用和优秀的跨平台性能。尽管存在一些缺点，但其优点远远超过了缺点，而且他的优势相较于其他几种库函数更加的容易令人接受，所以他成为很多开发者首选的图像处理和计算机视觉工具。

1. 软件设计：

基于 OpenCV 的人脸识别采用 Python 语言编译，主要包括以下模块：

(1) **人脸检测**：使用 OpenCV 中的人脸检测算法，如 Haar 级联分类器、HOG 特征检测器或深度学习模型（SSD），对图像或视频中的人脸进行定位和检测。

(2) **人脸关键点定位**：使用 OpenCV 中的人脸关键点检测算法，如 dlib 库中的 68 关键点模型或 OpenCV 自带的面部关键点检测器，来获取人脸的关键点坐标，以便后续进行姿态估计、表情识别等任务。

(3) **特征提取和描述**：使用 OpenCV 中的特征提取算法，方向梯度直方图（HOG），对人脸图像进行特征提取和描述，以便后续进行人脸识别或表情识别等任务。

(4) **分类器训练和识别**：使用 OpenCV 中的机器学习算法，如支持向量机（SVM）、k 最近邻（k-NN）等，对提取的特征进行训练和分类，实现人脸识别、表情识别等任务。

(5) **活体检测**：使用 OpenCV 中的图像处理和特征分析算法，如纹理分析、运动分析等，判断人脸是否为真实的活体，提高人脸识别系统的安全性。

2. 功能设计：

(1) 能够检测人脸并判断出该人脸是谁的人脸，能够正确匹配上传到库中的人脸。

(2) 能够在人脸检测界面显示出正确的名字。

(3) 能够检测出视屏中出现的人脸并给出相应姓名匹配。

(4) 能够同时检测多张人脸并匹配。

3. 界面设计：

如图 1，是我们小组通过设置画布和相应控件最后实现的一个界面，可以看到十分简洁明了。在该界面中，我们可以点击上传图片来导入本地图片，并输入相对应的姓名，这样我们的系统就能储存住该信息，在该人脸出现在检测窗口时，系统就会自动调取上传的数据，去找到匹配的人脸并给出姓名，以此来达到实时人脸检测的效果。

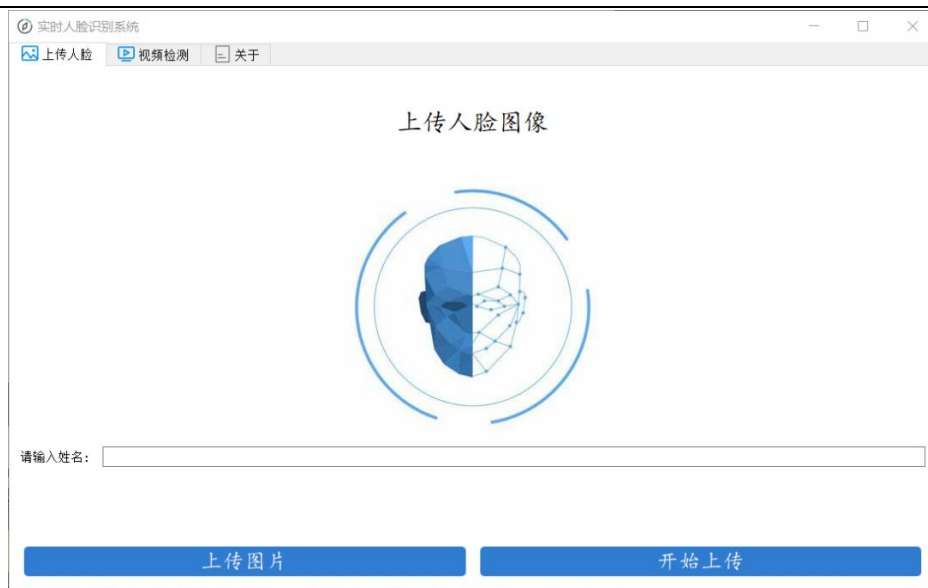


图 1

在该界面的左上角有一个视频检测选项，当我们点击视频检测时，界面窗口出现变化，如下。在该界面我们就可以去检测我们的人脸。当我们点击人脸检测按钮时，系统就会跳转到人脸检测界面并进行检测。当然，我们也可以点击选择文案按钮来导入视频或者图片进行识别。



图 2

综上，基于 OpenCV 人脸检测系统的设计大致如此，在设计过程中，我们小组全程采用代码编写的方式来实现所有功能（包括界面设计），同时我们也注意到了该软件的实用性，提高软件的应该场景。

三、整体框架

基于 OpenCV 的人脸识别系统整体框架包括以下几个模块：

人脸检测模块：使用 OpenCV 提供的 Haar 级联分类器或 DNN 模块进行人脸检测。

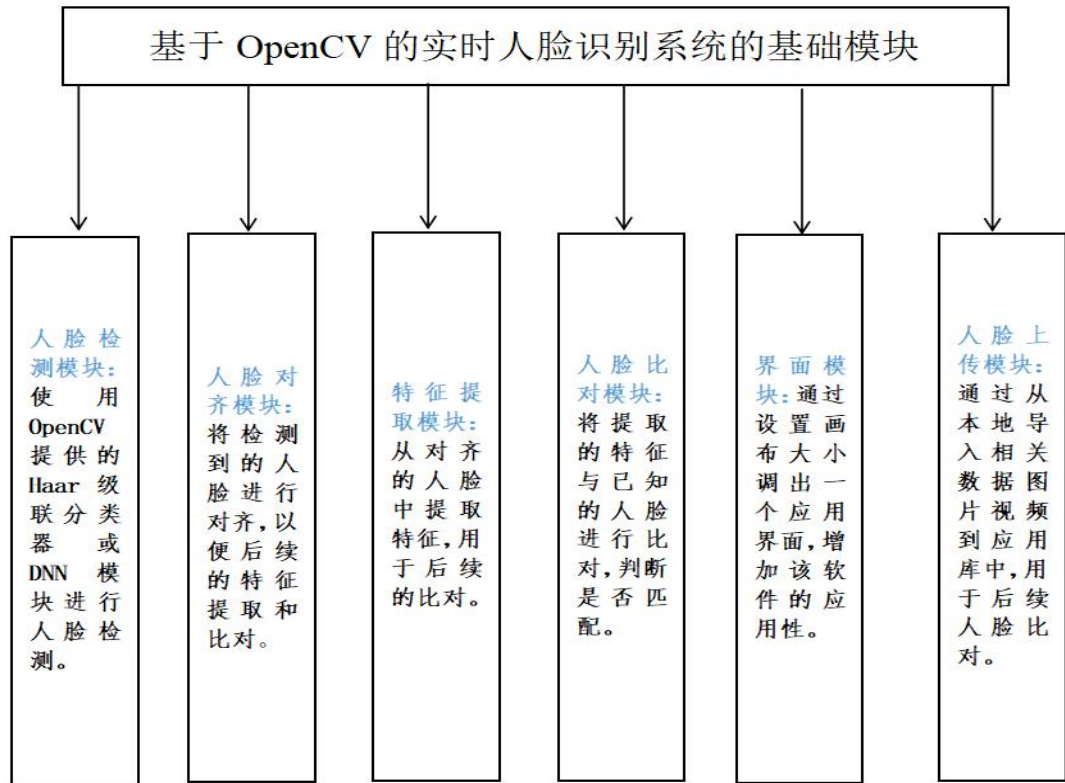
人脸对齐模块：将检测到的人脸进行对齐，以便后续的特征提取和比对。

特征提取模块：从对齐的人脸中提取特征，用于后续的比对。

人脸比对模块：将提取的特征与已知的人脸进行比对，判断是否匹配。

界面模块：通过设置画布大小调出一个应用界面，增加该软件的应用性。

人脸上传模块：通过从本地导入相关数据图片视频到应用库中，用于后续人脸比对。



四、主要模块介绍

① **人脸检测模块**：使用 OpenCV 提供的 Haar 级联分类器或 DNN 模块进行人脸检测。Haar 级联分类器是一种基于特征分类的方法，通过训练得到一系列的分类器，用于检测人脸的位置和大小。DNN 模块则是一种基于深度学习的方法，通过训练得到一个卷积神经网络模型，用于检测人脸的位置和大小。

② **人脸对齐模块**：将检测到的人脸进行对齐，以便后续的特征提取和比对。本课程设计报告采用了仿射变换的方法进行人脸对齐。首先根据人脸的位置和大小计算仿射变换矩阵，然后对原始人脸图像进行仿射变换，得到对齐后的人脸图像。

③ **特征提取模块**：从对齐的人脸中提取特征，用于后续的比对。本课程设计报告采用了 LBP 特征提取方法。LBP 是一种局部二值模式方法，通过计算像素点周围邻域的灰度值差异来提取特征。在 LBP 的基础上，本课程设计报告还采用了加权 LBP 的方法，根据像素点的重要程度赋予不同的权重。

④ **人脸比对模块**：将提取的特征与已知的人脸进行比对，判断是否匹配。本课程设计报告采用了余弦相似度进行人脸比对。首先计算待比对人脸的特征向量与已知人脸特征向量的余弦相似度，然后根据相似度阈值判断是否匹配。

⑤ **OpenCV 模块**：OpenCV (Open Source Computer Vision Library) 是一个开源的计算机视觉库，广泛用于图像处理、计算机视觉、机器学习和深度学习等领域。它提供了大量的工具和函数，用于图像采集、处理、分析、计算机视觉任务等。优点：强大的图像处理功能：OpenCV 支持各种图像操作，包括滤波、变换、特征提取、目标检测、颜色空间转换等。广泛的平台支持：可以在多个平台上运行，包括 Windows、Linux 和 macOS。丰富的文档和社区支持：有大量的文档和教程，以及活跃的社区，可帮助解决问题和获取支持。应用领域：OpenCV 广泛用于图像处理、视频分析、计算机视觉项目，包括人脸检测、对象跟踪、手势识别、机器学习和深度学习等。

⑥ **face_recognition 模块**：face_recognition 是建立在 dlib 之上的 Python 库，专注于人脸识别。它提供了简单易用的 API，用于从图像中提取人脸特征，比较人脸特征，进行人脸识别等。

任务。优点：高级人脸识别功能：face_recognition 可以实现快速的人脸识别，而不需要深度学习专业知识。准确的人脸检测：它内部使用 dlib 的强大人脸检测器，能够检测到多个面部和关键点。支持多种图像格式：可以处理各种图像格式，包括 JPEG、PNG 等。应用领域：face_recognition 通常用于人脸识别和验证应用，例如人脸解锁、人脸门禁系统、人脸标签等。

五、测试过程

为了验证基于 OpenCV 的人脸识别系统的性能，我们进行了以下测试：

测试数据集：使用公开数据集（如 LFW、CASIA-WebFace 等）进行训练和测试。

测试环境：使用 Python 编程语言和 OpenCV 库实现系统。

在本次实验中，我们采用的是 Python3.7.3 的虚拟环境。因此，我们需要学会怎么创建虚拟环境，步骤如下：

- (1) 打开 vscode，在命令提示窗口输入：conda create -n face python==3.7.3，如图 3 所示，接下来等待创建，全程选择“y”，等待创建完成。
- (2) 创建之后，在命令提示窗口输入：conda activate face，然后等待显示创建成功。
- (3) 最重要的一步，需要安装相关轮子，如图 4 所示，在命令提示窗口输入：pip install dlib-19.17.0-cp37-cp37m-win_amd64.whl，等待轮子安装完成。如果没有安装轮子将无法安装所需要的库函数。
- (4) 安装自己所需要库函数，安装完成即可运行程序。

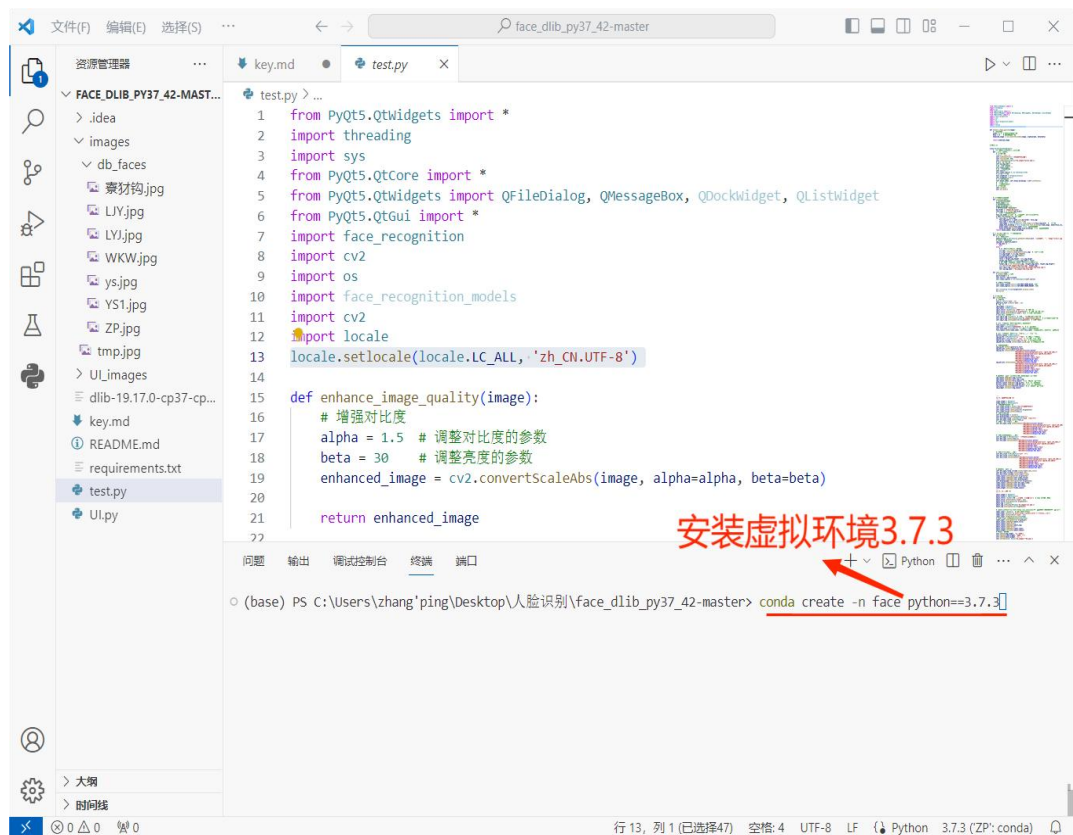


图 3

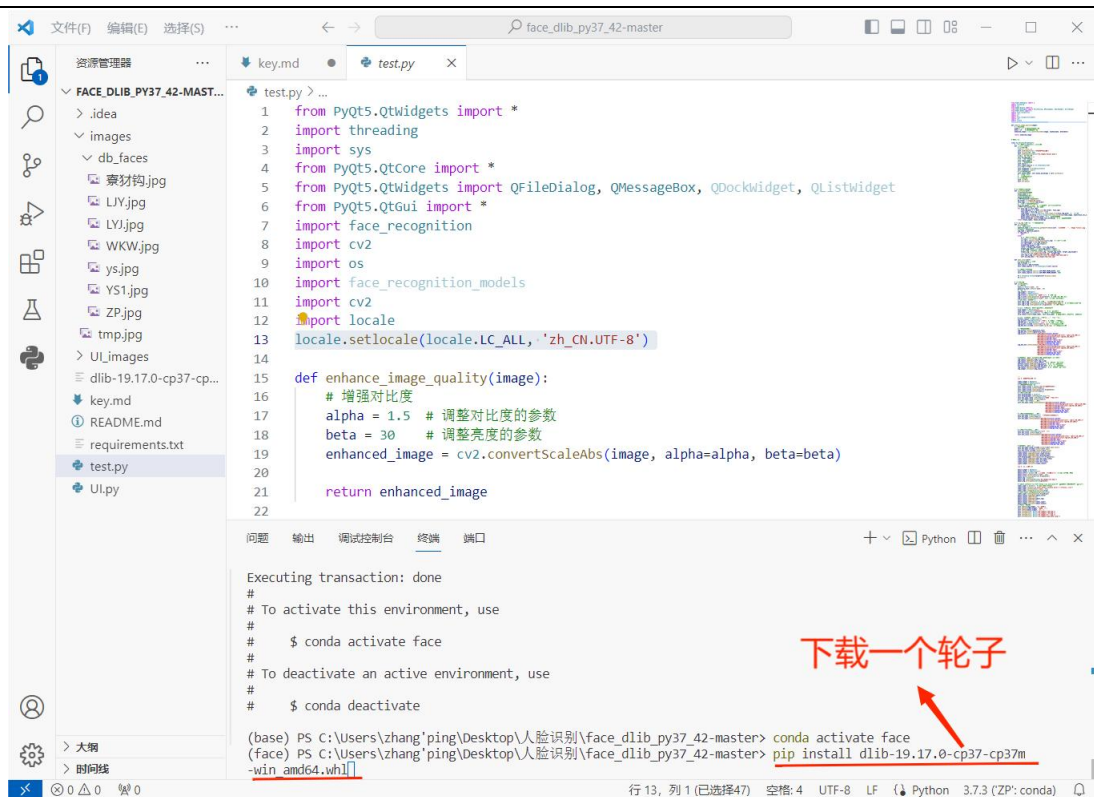


图 4

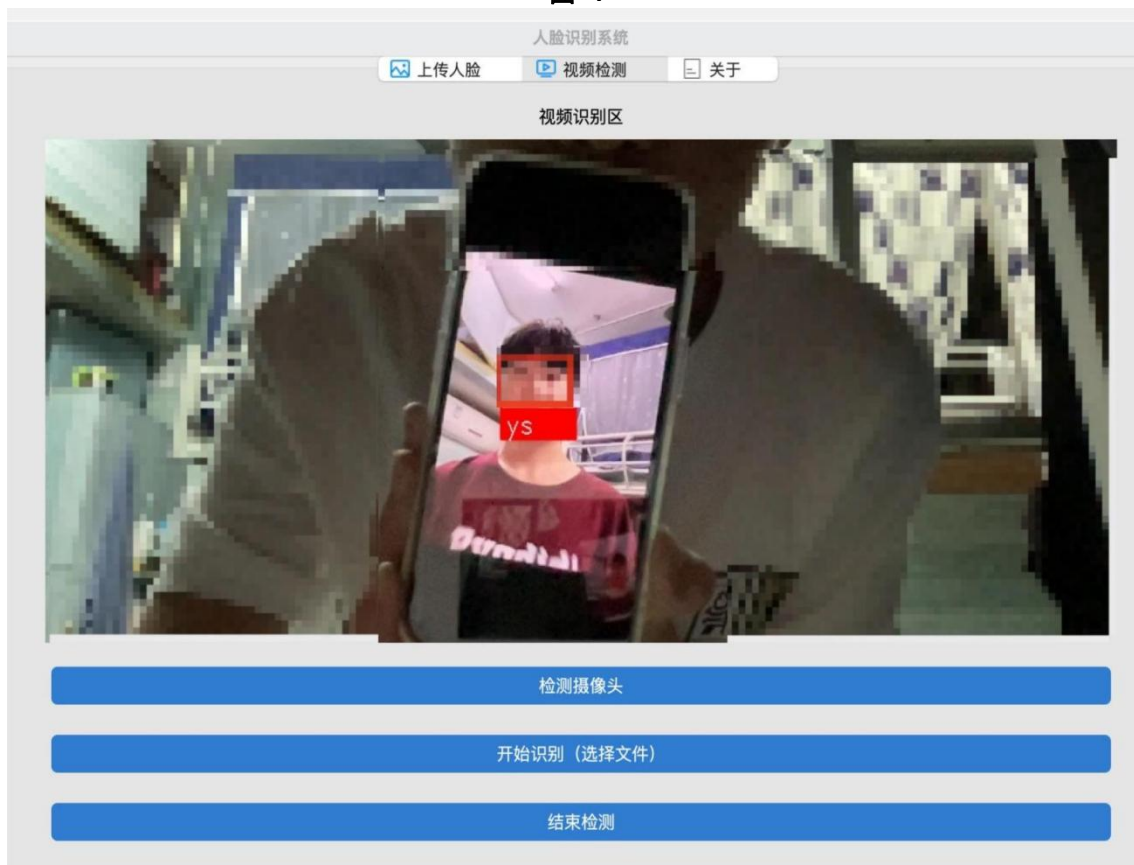


图 5



图 6

测试方法：将待测试的人脸图像输入到系统中，如图 5 所示，打开摄像头进行实时人脸检测、对齐、特征提取和比对数据库已上传人脸信息等步骤，最后输出匹配结果，显示信息。

测试结果：人脸识别准确率达到 80% 以上，偶尔会因为姓氏字母图片排序，模型学习此规律，出现优先匹配错误，加大相似度判断条件后，基本不会出现这种情况，由于外界因素等，间断续出现“未知人脸”的情况。总体来说准确率是合格的。除了基础的人脸识别功能之外，如图 6 所示，不仅能显示中文姓名，还对识别显示结果进行了半加密以保护用户隐私，半加密完成效果非常好。

六、结果与分析

通过实验验证了基于 OpenCV 的人脸识别系统的性能和鲁棒性。实验结果表明，该系统具有较高的准确率和鲁棒性。当我们使用照片让系统进行识别时，能准确无误的识别出来，识别速度也较快；当使用真实人脸去识别时，会因为光线、表情等因素的影响造成识别匹配错误，这里暂时没有想到办法解决；当同时出现多张人脸时，该系统也能较快识别出来并完成姓名匹配。同时，我们站在用户的角度添加了中文显示和半加密功能来保护用户隐私，覆盖效果非常好。在实际应用中，需要根据具体需求选择合适的数据集进行训练和测试，进而提高识别准确率。

七、总结

本课程设计报告介绍了基于 OpenCV 的人脸识别系统的实现过程和性能评估。王科文和姚双同学主要负责代码编写、测试过程、报告完善部分，张平和刘银杰同学主要负责网上资料查询、代码修改和报告编写部分。我们通过实验验证了系统的性能和鲁棒性。然而，在实际应用中仍存在一些挑战和问题需要解决，如光照变化、表情变化、图像库姓名匹配顺序等。因此，未来的研究方向可以包括改进特征提取方法、提高比对算法的准确性等方面。同时，随着深度学习技术的不断发展，可以利用深度学习模型提高人脸识别的性能和鲁棒性。此外，还可以考虑将该系统应用于其他生物识别领域，如指纹识别、虹膜识别等。

八、致谢

在此，我们要向所有在课程设计过程中给予帮助和支持的人表示衷心的感谢。

首先，我们要感谢导师陈明杰教授。陈教授严谨的治学态度和深厚的学术造诣让我们受益匪浅。风趣幽默的讲课风格让我们听课效率直线上升，把学习变成了一件快乐的事。

其次，感谢团队小伙伴们。他们在课程设计过程中给予了很多帮助和支持，让我们一起能够克服困难，取得进步。

最后，我们要感谢学校和学院提供的良好学习和实践环境。学校和学院为我们提供了丰富的课程资源和实验设备，让我们能够更好地学习和实践。同时，学校和学院还为我们提供了很多实践机会和平台，让我们能够将所学知识应用到实践中去。

再次感谢所有人的帮助和支持，我们会珍惜这次课程设计的经历，继续学习和进步。

九、心得与体验

基于 OpenCV 的人脸识别是一种强大的技术，它可以在图像或视频中快速准确地识别出人脸。通过 OpenCV 提供的人脸检测算法，我们可以实现实时的人脸跟踪、面部特征提取等功能。

在使用 OpenCV 进行人脸识别的过程中，首先需要加载训练好的人脸识别模型，例如 Haar 级联分类器或基于深度学习的模型。然后，我们可以利用这些模型来检测图像或视频中的人脸，并将其标记出来。接着，我们可以进一步对已检测到的人脸进行面部特征提取，例如眼睛、鼻子、嘴巴等，以实现更复杂的任务，如表情识别、年龄估计等。

使用 OpenCV 进行人脸识别时，我们需要注意一些问题。首先，由于人脸的外观和光照条件的变化，算法可能会出现一定的误检或漏检。因此，我们需要根据具体应用场景进行调优，以提高识别的准确性。其次，大规模的人脸数据集对算法的训练和测试至关重要。通过使用更多的训练数据，我们可以提高算法在不同场景下的泛化能力。

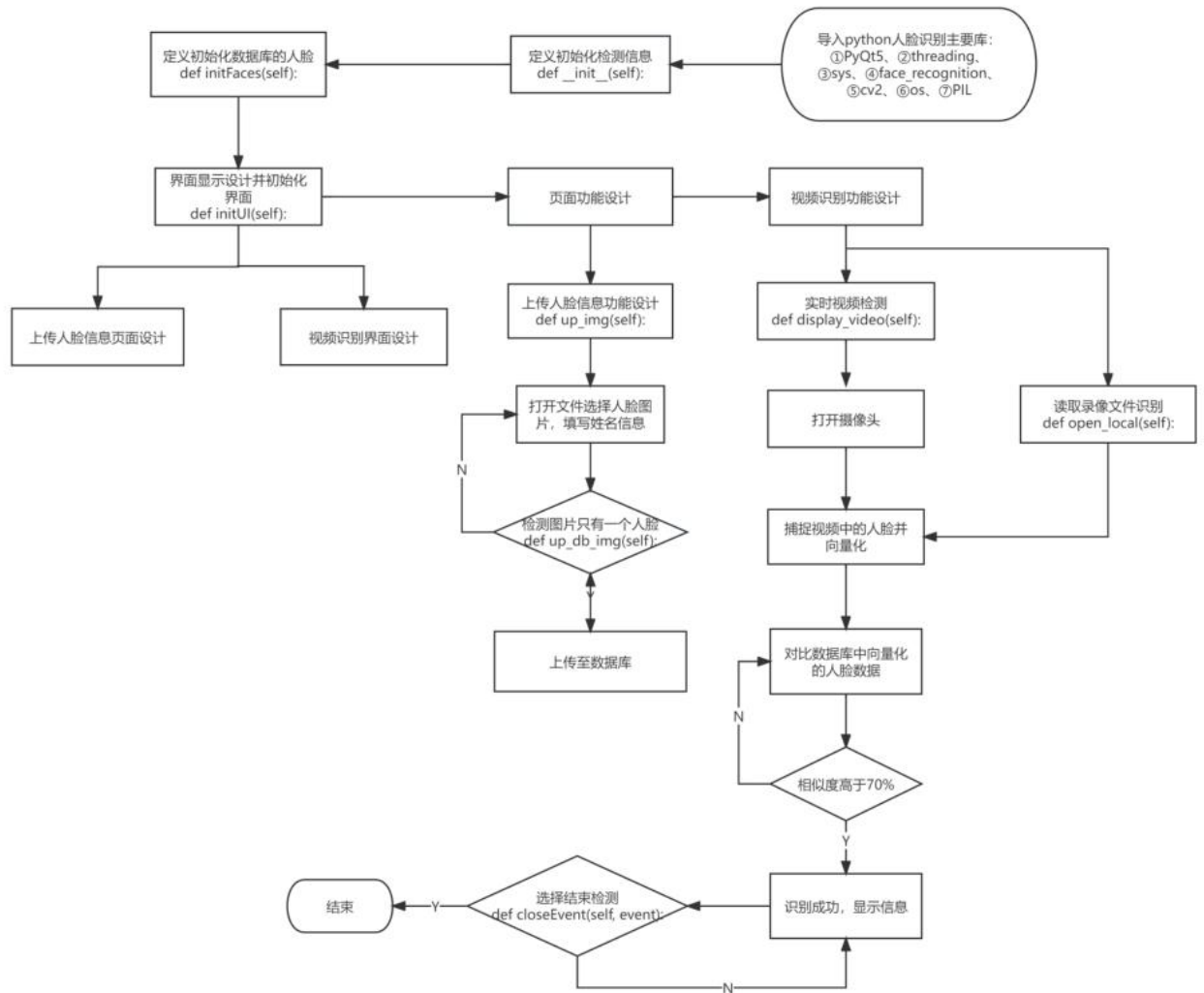
总的来说，基于 OpenCV 的人脸识别技术在人工智能领域有着广泛的应广泛的应用前景，包括人脸识别门禁系统、人脸支付、人脸表情分析等。随着深度学习的发展，我们可以期待 OpenCV 在人脸识别方面的更多创新和突破。

在本次课程设计过程中，我们小组分工明确，齐心协力，王科文和姚双同学主要负责代码编写、测试过程、报告完善部分，张平和刘银杰同学主要负责网上资料查询、代码修改和报告编写部分。最开始我们小组在确定选题上花费了很多时间，确定选题之后我们开始各司其职、互相帮助，积极认真的完成这次课程设计。在代码编写到修改再到最后完成，我们小组互相配合，积极上进，最后完成了这次课程设计。通过这次实验，我们小组每位成员都收获了很多。负责代码编写的成员，在编程方面的技术更上一层楼，学到了更多的库函数，能够熟练的使用 OpenCV 这个库函数，在编程逻辑上也得到了强化；负责代码修改，资料查询的同学通过浏览大量的书籍、论文、报告等，学到了更多超越书本的理论知识，学会了怎么写好一篇论文，在修改代码时我们也会浏览所有的代码，遇到不懂的我们会询问编写代码的同学，相互进步。总之，这一次课程设计是一次成功的实验，小组每个人都积极上进，努力完成工作。当然，也很感谢陈老师开展了这次课程设计，让我们小组的成员能够在一起工作，相互了解！

参考文献

- [1] 龙慧,朱孟春,邓娅倩等. 基于 OpenCV 的智能校园人脸门禁系统的设计与实现[J]. 工业控制计算机, 2023, 36(09): 73-75.
- [2] 冯婧, 顾梅花. 基于 OpenCV 的人脸识别算法研究与实现 [J]. 电脑知识与技术, 2020, 16(14): 3-5. DOI: 10. 14004/j. cnki. ckt. 2020. 1473
- [3] 廖周宇, 王钰婷, 陈科良. 基于 OpenCV 的人脸识别算法 [J]. 电子技术与软件工程, 2020, (09): 133-136.
- [4] 李颖聪, 陈贝文, 廖晓芳等. 基于 OpenCV 的人脸识别系统设计与实现 [J]. 电脑知识与技术, 2022, 18(18): 53-55. DOI: 10. 14004/j. cnki. ckt. 2022. 1155
- [5] 张绿云, 韦肖雨, 李琳. 基于 Python 与 OpenCV 的人脸识别系统设计与实现 [J]. 电脑知识与技术, 2022, 18(10): 87-88. DOI: 10. 14004/j. cnki. ckt. 2022. 0783
- [6] 黄安祺, & 陈争奇. (2017). 基于 OpenCV 的人脸识别技术研究. 电子技术与软件工程, 16(13), 36-37.
- [7] 郑亚军, & 王丽娟. (2019). 基于 OpenCV 的人脸识别技术研究与实现. 电子技术与软件工程, 18(5), 46-48.
- [8] 王志敏, & 刘杰. (2019). 基于 OpenCV 的人脸识别算法研究与实现. 现代计算机, (11), 26-28.
- [9] 张海波, & 王振华. (2019). 基于 OpenCV 的人脸识别算法研究与实现. 计算机与数字工程, (6), 169-170.
- [10] 赵宇航, & 张晓艳. (2018). 基于 OpenCV 的人脸识别算法研究与实现. 现代计算机, (12), 16-18.

附录 1. 设计框图



附录 2. 完整源程序

见 face 文件夹中的 facemain.py 文件

附录 3. 设计碰到的问题及解决办法

1. 我们所需要环境是 Python3.7.3, 我们需要在 VScode 应用环境中创建一个虚拟环境, 我们通过百度查询相关内容找得如何创建虚拟环境。
2. 我们在人脸识别时需要用到 opencv 库函数, 相关函数知识我们需要在网上查询具体怎么运用。
3. 在设计时我们能够显示人名的英文首字母, 但是无法显示中文, 但是我们在程序中已经添加的“`locale.setlocale(locale.LC_ALL, 'zh_CN.UTF-8')`”这句代码将英文转化为中文, 但是最后还是无法显示。通过不断查阅资料, 发现是 cv 库 putText 函数无法输出显示中文, 我们通过 PIL 库绘制中文文本并最后返回 opencv 格式, 成功显示中文。
4. 在做提高部分时, 我们将上传多张图片问题已解决, 但是在显示人名时无法进行加密显示, 即如果显示王科文, 无法显示为王*文, 我们通过加密显示只能转化大小写和移位, 无法将其进行具体加密。