

ĐẠI HỌC QUỐC GIA HÀ NỘI  
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ

---

BÁO CÁO MÔN CHUYÊN ĐỀ CÔNG NGHỆ  
NĂM HỌC 2020-2021

TÊN DỰ ÁN:

**NHẬN DẠNG KHUÔN MẶT HỖ TRỢ CAMERA  
AN NINH GIA ĐÌNH**

Nhóm 22: Nguyễn Xuân Trường

Nguyễn Đình Ngọc

HÀ NỘI – 2021

# Mục lục

1. TÓM TẮT.....	2
2. ĐẶT VẤN ĐỀ.....	2
2.1 Động lực .....	2
2.2 Mô tả dự án.....	3
2.3 Đóng góp chính.....	4
3. ỨNG DỤNG “NHẬN DIỆN KHUÔN MẶT HỖ TRỢ CAMERA AN NINH GIA ĐÌNH” 4	
3.1 Mô tả ứng dụng .....	4
3.1.1 Thiết kế, xây dựng cơ sở dữ liệu .....	4
3.1.2 Thiết kế giao diện .....	6
3.2 Công nghệ cốt lõi được sử dụng.....	10
3.2.1. Nhận diện người.....	10
3.2.2. Nhận diện khuôn mặt (Face Detection).....	11
3.2.3. Nhận dạng khuôn mặt (Face Recognition) .....	13
3.3 Thử nghiệm trên tập dữ liệu lớn.....	17
3.3.1 Tổng quan về bộ dữ liệu .....	17
3.3.2 Xử lý dữ liệu .....	18
3.3.3 Kết quả thu được.....	20
4. KẾT LUẬN.....	21

# 1. TÓM TẮT

Ngày nay, công nghệ nhận dạng khuôn mặt đã và đang được áp dụng rộng rãi hiện nay, và rất có thể trong tương lai, công nghệ này sẽ tiếp tục được phát triển và ứng dụng nhiều hơn nữa trong cuộc sống. Trên thế giới, công nghệ nhận diện khuôn mặt được ứng dụng rất rộng rãi cho các mục đích bảo mật, xác thực danh tính, tìm người mất tích,... Ở Việt Nam, đa số các ứng dụng của công nghệ nhận dạng khuôn mặt dành cho những mục đích như xác thực thanh toán, hỗ trợ bảo mật cho các tổ chức, cho các thiết bị cá nhân, chấm công, điểm danh,... Tuy nhiên, chưa có một ứng dụng cụ thể nào cho camera an ninh gia đình. Thực tế cho thấy, các hệ thống camera an ninh gia đình hiện tại chỉ có chức năng ghi hình. Hệ quả là sau mỗi vụ trộm cắp tài sản, người ta chỉ có thể sử dụng camera để truy tìm thủ phạm, mà không hề có biện pháp cảnh báo, phòng ngừa. Nhận thấy được vấn đề trên, nhóm quyết định nghiên cứu, xây dựng ứng dụng “Nhận diện khuôn mặt hỗ trợ camera an ninh gia đình”, với các chức năng chính là nhận diện người có mặt thông qua hình ảnh từ camera thứ nhất, sau đó nhận diện, xác minh danh tính người đó thông qua hình ảnh từ camera thứ 2, từ đó đưa ra các quyết định như xác nhận người đó là người lạ hay quen, gửi thông báo đến cho chủ nhà. Phần cốt lõi của ứng dụng, nhóm dựa vào thuật toán Histograms of Oriented Gradients (HoG) để nhận diện người, kết hợp thuật toán Local Binary Patterns Histograms (LBPH) và mô hình DNN để nhận dạng khuôn mặt. Phần giao diện được xây dựng với PyQt5, thông tin của người dùng được quản lý bằng cơ sở dữ liệu Sqlite. Ngoài ra, để kiểm tra độ chính xác, nhóm đã thử nghiệm trên tập dữ liệu lớn Labeled Faces in the Wild-a.

## 2. ĐẶT VẤN ĐỀ

### 2.1 Động lực

Công nghệ nhận diện khuôn mặt đang được áp dụng trong rất nhiều các lĩnh vực, đặc biệt là lĩnh vực bảo mật nhờ các ưu điểm vượt trội của nó như: là kỹ thuật sinh trắc học không cần tiếp xúc trực tiếp với thiết bị để xác thực, yêu cầu xử lý ít hơn, độ chính xác được cải thiện theo thời gian, dễ dàng tích hợp với các tính năng bảo mật hiện có,...

Trên thế giới, các nước phát triển đang đầu tư nhiều nguồn lực vào công nghệ nhận diện khuôn mặt. Chính phủ Mỹ đã quyết định tăng cường an ninh sân bay bằng hệ thống nhận dạng khuôn mặt để nhận dạng và đăng ký của du khách. Hoa Kỳ có một số tiểu bang đã cho phép cơ quan thực thi pháp luật thực hiện các cuộc tìm kiếm trong cơ sở dữ liệu – những cuộc tìm kiếm này bao gồm các chi tiết về giấy phép lái xe và ảnh ID. Kỹ thuật nhận dạng khuôn mặt và kết quả tìm kiếm cũng có thể được sử dụng trong các cuộc kiểm tra của cảnh sát. Ở Ấn Độ, các ngân hàng đang sử dụng công nghệ nhận dạng khuôn mặt này để ngăn chặn gian lận tại ATM. Nó cũng được sử dụng để báo cáo cử tri trùng lặp, xác minh hộ chiếu và thị thực, giấy phép lái xe,... Không chỉ vậy, các tập đoàn công nghệ đa quốc gia như Facebook, Google,... cũng phát triển công nghệ nhận diện khuôn mặt cho riêng mình.

Ở Việt Nam, công nghệ nhận dạng khuôn mặt được ứng dụng vào các mục đích như xác thực danh tính, xác thực thanh toán, chấm công, điểm danh, hỗ trợ bảo mật cho các tổ chức,... Tuy nhiên việc áp dụng để hỗ trợ an ninh gia đình thì chưa có. Ngày nay, nhu cầu về bảo mật, an ninh gia đình ngày càng trở nên quan trọng. Một trong các phương pháp để nâng cao vấn đề này là sử dụng camera an ninh gia đình. Tuy nhiên, thực tế khảo sát cho thấy, các camera hiện tại chỉ có chức năng chủ yếu là ghi lại hình ảnh. Khi có các vụ trộm cắp tài sản xảy ra, người dùng chỉ có thể trích xuất dữ liệu từ camera để truy lùng ra thủ phạm. Việc này vô cùng khó khăn và nhiều khi không hiệu quả. Bởi vậy, một phương pháp để cảnh báo cho người dùng khi camera phát hiện ra những hành vi bất thường là điều vô cùng cần thiết để phòng chống sớm các vụ đột nhập, trộm cắp tài sản.

Lý do trên chính là động lực để nhóm thực hiện dự án nghiên cứu, xây dựng ứng dụng “Nhận diện khuôn mặt hỗ trợ camera an ninh gia đình”. Về cơ bản, ứng dụng trích xuất hình ảnh thu được từ các camera, sử dụng công nghệ nhận diện người ở camera thứ nhất để kiểm tra sự có mặt của người tại vị trí phía trước camera thứ hai, sử dụng công nghệ nhận dạng khuôn mặt để xác minh danh tính của người đứng trước camera thứ 2, từ đó đưa ra thông báo cho người dùng (chủ nhân của ngôi nhà).

## 2.2 Mô tả dự án

Khi xác định được vấn đề của bài toán, nhóm nghiên cứu đã bàn bạc và thực hiện dự án bắt đầu từ tháng 3/2021. Dự án bao gồm những 4 thành phần như sau:

- Cơ sở dữ liệu: Sử dụng cơ sở dữ liệu Sqlite 3, lưu trữ dữ liệu của người dùng.
- Phần nhận diện người: Nhận diện người xuất hiện trước camera thứ nhất.
- Phần nhận diện khuôn mặt: Nhận dạng khuôn mặt của người xuất hiện trước camera thứ hai.
- Phần giao diện: Tạo ứng dụng máy tính bằng thư viện để tạo giao diện Pyqt5.
- Phần thử nghiệm trên bộ dữ liệu lớn: Thực hiện đánh giá trên bộ dữ liệu Labeled Faces in the Wild-a (LFW-a).

Sau khi thống nhất về các công việc cần triển khai thì nhóm đã phân công cụ thể cho từng thành viên phụ trách từng phần trong dự án phù hợp với điều kiện và năng lực của từng thành viên như sau:

STT	Thành viên	Nhiệm vụ
1	Nguyễn Xuân Trường	Thiết kế, xây dựng cơ sở dữ liệu Nghiên cứu áp dụng công nghệ nhận dạng khuôn mặt Thử nghiệm trên bộ dữ liệu lớn
2	Nguyễn Đình Ngọc	Thiết kế giao diện Nghiên cứu, áp dụng công nghệ nhận diện người Thử nghiệm trên bộ dữ liệu lớn

Bảng 1: Phân chia công việc

## 2.3 Đóng góp chính

Sau khi hoàn thành dự án nghiên cứu, xây dựng ứng dụng "*Nhận dạng khuôn mặt hỗ trợ camera an ninh gia đình*", nhóm nghiên cứu đã có những đóng góp như sau:

- Về dữ liệu, nhóm đã xử lý bộ dữ liệu (...) sao cho phù hợp với dự án, thử nghiệm trên bộ dữ liệu lớn để đánh giá kết quả đã đạt được.
- Về mặt kỹ thuật, các đóng góp có thể chia nhỏ thành những điểm sau:
  - Bài toán nhận dạng khuôn mặt là bài toán kinh điển, đã có nhiều ứng dụng. Tuy nhiên, ứng dụng công nghệ nhận dạng khuôn mặt, kết hợp nhận diện người để hỗ trợ camera an ninh gia đình là điều chưa được nghiên cứu hay ứng dụng cụ thể tại Việt Nam. Bên cạnh đó, có rất nhiều thuật toán, mô hình khác nhau có thể áp dụng, nhóm đã tìm hiểu, đánh giá điểm mạnh yếu và lựa chọn thuật toán phù hợp.
  - Nhóm đã sử dụng PyQt5 để xây dựng một ứng dụng với giao diện thân thiện với người dùng. Với điều kiện hiện tại, ứng dụng đã chạy thành công trong điều kiện mô phỏng với một camera từ máy tính và một camera IP từ điện thoại.
  - Nhóm sử dụng Sqlite để quản lý thông tin (họ tên, id, tuổi, giới tính, ảnh) của người dùng.
  - Nhóm đã thử nghiệm và đánh giá kết quả đạt được qua bộ dữ liệu lớn Labeled Faces in the Wild-a (LFW-a).




## 3. ỨNG DỤNG “NHẬN DIỆN KHUÔN MẶT HỖ TRỢ CAMERA AN NINH GIA ĐÌNH”

### 3.1 Mô tả ứng dụng

#### 3.1.1 Thiết kế, xây dựng cơ sở dữ liệu

Trong dự án này, nhóm thiết kế một cơ sở dữ liệu đơn giản chỉ gồm một bảng, trong đó có các trường thông tin như sau:

- ID: id là duy nhất để phân biệt các người dùng khác nhau.
- Name: Tên của người dùng
- Age: Tuổi của người dùng
- Gender: Giới tính của người dùng

Table name: <input type="text" value="People"/>		<input type="checkbox"/> WITHOUT ROWID							
	Name	Data type	Primary Key	Foreign Key	Unique	Check	Not NULL	Collate	Default value
1	ID	INTEGER							NULL
2	NAME	STRING							NULL
3	Age	INTEGER							NULL
4	Gender	TEXT							NULL

Hình 1: Bảng cơ sở dữ liệu cho người dùng

Trong quá trình xây dựng cơ sở dữ liệu, nhóm sử dụng công cụ SQLiteStudio để thêm thủ công dữ liệu người dùng, ví dụ như:

	ID	NAME	Age	Gender
1	<input type="text" value="1"/>	John Wick	53	Male

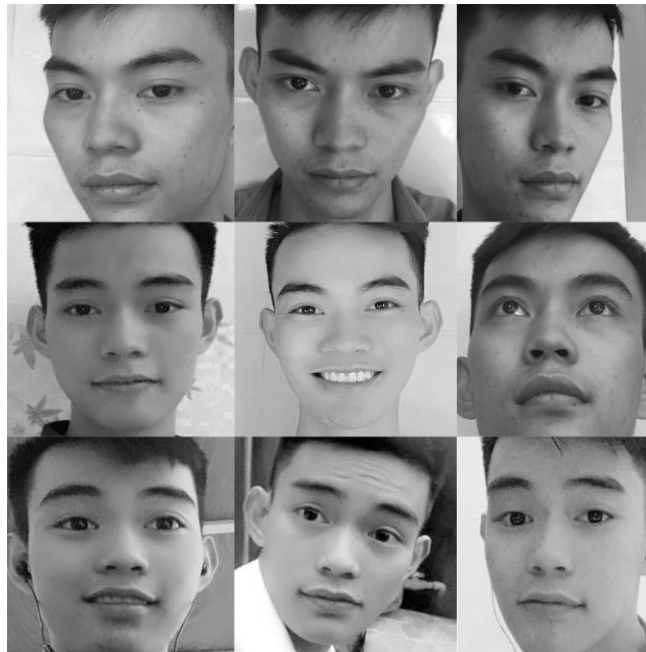
Hình 2: Ví dụ dữ liệu của một người dùng

Bên cạnh cách thêm dữ liệu thủ công, nhóm đã xây dựng một module để thêm một người dùng mới thông qua việc chọn “Đăng ký mới” trên màn hình giao diện.

```
def insertOrUpdate(id, name, age, gender):
    db = connectToDatabase()
    query = "SELECT * FROM People WHERE ID = " + str(id)
    cursor = db.execute(query)
    isRecordExist = 0
    for row in cursor:
        isRecordExist = 1
    if isRecordExist == 0:
        query = "Insert into People(ID, Name, Age, Gender) values(" + str(id) + "," + str(name) + \
            + "," + str(age) + "," + str(gender) + ")"
    else:
        query = "Update People set Name = '" + str(name) + "', Age = " + str(age) + ", Gender = '" + \
            + str(gender) + "' Where ID = " + str(id)
    db.execute(query)
    db.commit()
    db.close()
```

Hình 3: Module thêm người dùng mới hoặc cập nhật thông tin người dùng cũ

Khi đăng ký mới, người dùng thực hiện chụp các tấm hình khuôn mặt của mình để sử dụng làm tập dữ liệu huấn luyện mới. Chi tiết về cách thức hoạt động sẽ được đề cập ở phần sau. Bộ dữ liệu này bao gồm 20 tấm ảnh chụp các khuôn mặt với biểu cảm và góc cạnh khác nhau.



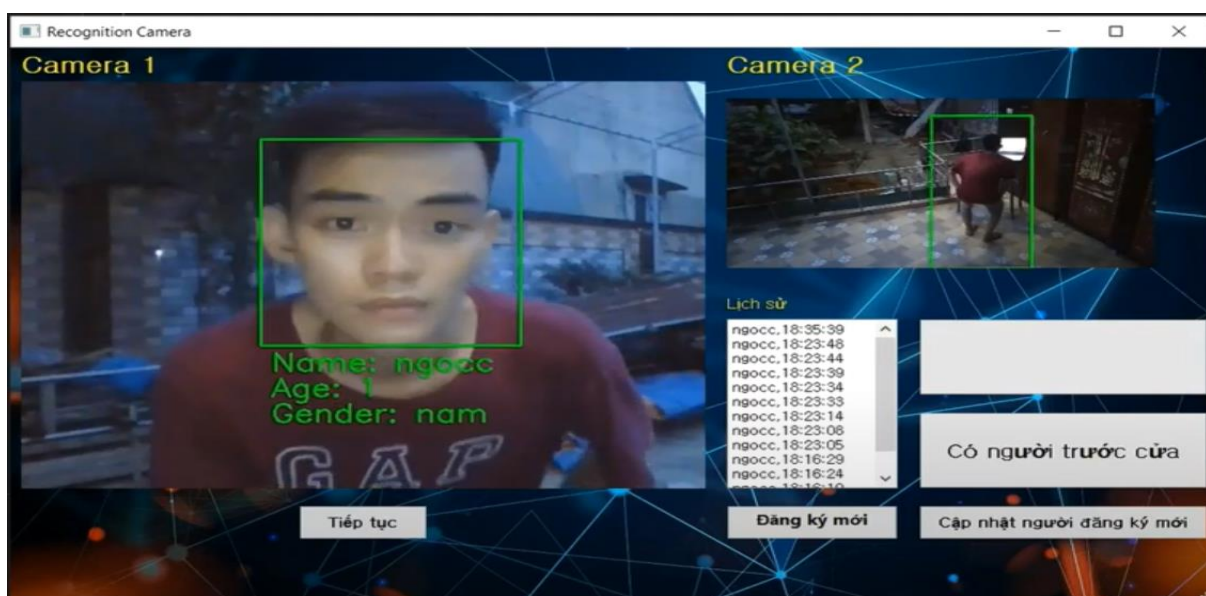
Hình 4: Một số hình ảnh đăng ký mới

### 3.1.2 Thiết kế giao diện

Dự án sử dụng thư viện PyQt. PyQt là một thư viện Python để tạo các ứng dụng GUI bằng bộ công cụ Qt. Được tạo bởi Riverbank Computing, PyQt là phần mềm miễn phí (được cấp phép GPL) và được phát triển từ năm 1999. Có hai phiên bản chính hiện đang được sử dụng: PyQt5 dựa trên Qt5 và PyQt6 dựa trên Qt6. PyQt6 có cải tiến hơn ở việc hỗ trợ cho một số mô-đun nâng cao và nó cũng thực hiện một số thay đổi đối với không gian tên và các hoạt động.

Trong dự án này thì nhóm quyết định sử dụng PyQt5 vì nó đã có đủ những thứ mà nhóm cần, hơn nữa việc sử dụng PyQt5 là điều mà chúng em đã có kinh nghiệm.

#### 3.1.2.1 Mô tả giao diện



Hình 5: Giao diện ứng dụng

- Hai màn hình thể hiện hình ảnh đã xử lý về ô khoanh vùng nhận diện từ hình ảnh đầu vào trực tiếp theo thời gian thực của hai camera chính
- Màn hình lịch sử: màn hình này hiển thị thông tin về người ra vào được camera nhận diện, bao gồm thông tin về tên và thời gian ra vào. Dữ liệu hiển thị được lấy từ file Attendance.csv
- Hai màn hình thông báo: Hai dòng chữ “Có người lạ” và “Có người trước cửa” sẽ xuất hiện trên màn hình khi hệ thống phát hiện được các dấu hiệu tương ứng trên
- 3 nút bấm với các chức năng được mô tả trong phần cách thức hoạt động của hệ thống.

### 3.1.2.2 Cách thức hoạt động

#### 1. Lấy hình ảnh trực tiếp từ hai camera

Trong đó một camera được đặt thấp cận cảnh để thu được hình ảnh khuôn mặt nhận diện chính xác nhất và camera còn lại được bố trí đặt trên cao để thu được khung hình toàn cảnh khu vực cần được theo dõi và nhận diện bởi hệ thống. Mỗi hình ảnh sẽ được xử lý ở mục 2.

#### 2. Đối với hình ảnh ở Camera nhận dạng khuôn mặt

Hệ thống xử lý để tìm kiếm khuôn mặt người và sau đó nhận dạng:

- Với những khuôn mặt được nhận dạng: thực hiện đánh dấu vùng khuôn mặt xuất hiện kèm thông tin đã được đăng ký. Cùng lúc đó thì thông tin ra vào kèm thời gian của người đấy cũng sẽ được ghi lại ở file Attendance.csv.
- Với những khuôn mặt không thể nhận diện: người đó được coi là người lạ và lưu hình ảnh người đó lại và gửi qua Email cho chủ nhà. Ý tưởng là như vậy, nhưng khi áp dụng, để tránh trường hợp những người đã được đăng ký mà hệ thống không thể nhận diện thành công (do khoảng cách, góc nhìn, độ chính xác của thuật toán nhận diện...) thì quy tắc để đánh dấu sự xuất hiện của một người lạ mà bọn em đang quyết định dùng là như sau: xét các khoảng thời gian con 0.5s, nếu tồn tại liên tiếp 6 khoảng thời gian con như thế mà hệ thống phát hiện được khuôn mặt người mà không thể nhận diện thì hệ thống mới xem người đó là người lạ và thực hiện các hành vi thông báo và cảnh báo.

```
def checkNumSecondsHasStranger(self):
    if self.hasStranger == True:
        self.numSecondsHasStranger = self.numSecondsHasStranger + 1
        self.hasStranger = False
    else:
        self.numSecondsHasStranger = 0
        self.notifyHasStranger = False
    if self.numSecondsHasStranger > 5:
        self.notifyHasStranger = True
        self.setAlertHasStranger()
```

Hình 6: Module đánh dấu sự xuất hiện của một người lạ



### 3. Đối với hình ảnh từ Camera nhận diện người

Ý tưởng xử lý là khi có người xuất hiện thì hệ thống sẽ chờ Camera nhận dạng khuôn mặt nhận dạng người đó, nếu nhận dạng được thì không có chuyện gì xảy ra, ngược lại thì hệ thống sẽ thực hiện các hành vi thông báo và cảnh báo qua Email. Cụ thể như sau:

- Với những người đi vào và được nhận dạng: hệ thống sẽ cho người đó 1 khoảng thời gian là 3 phút để thực hiện các hành vi của mình nếu người ấy tiếp tục ở lại khu vực đó để làm việc bất kỳ trước khi yêu cầu nhận dạng lại sau khoảng thời gian đấy. (Độ dài của khoảng thời gian này tùy thuộc vào vị trí đặt camera giám sát, đề xuất của nhóm trong dự án ứng với camera trước cửa ra vào là từ 30s ~ 1 phút)
- Với những người đi vào mà không thể nhận dạng: hệ thống sẽ làm việc như sau: để đánh dấu sự xuất hiện chính thức của một người, tránh nhầm lẫn với những người đi ngang qua thì giải pháp cài đặt hiện tại của hệ thống là xét các khoảng thời gian con 3s, nếu có 3 khoảng thời gian con liên tục như thế có hệ thống phát hiện có người xuất hiện trong khung hình thì người đó sẽ được coi là là xuất hiện chính thức trước cửa. Khi đó, hình ảnh người đó sẽ được lưu lại cùng thời gian xuất hiện để hệ thống thực hiện gửi Email thông báo.

```
def makeRequestRecognition(self):
    if self.hasHuman == True:
        self.hasHuman = False
        print("nhìn vào camera nhận dạng")
        self.setAlertPeopleAtDoor()
        self.numSecondsHasHuman += 3
    else:
        self.numSecondsHasHuman = 0
def checkNumSecondsHasHuman(self):
    if self.numSecondsHasHuman == 9 and self.IsAcquaintance == False:
        print("nguoi la khong the nhan dien, can gui mail thong bao")
        engine.say("Please look at camera 1 for identification")
        engine.runAndWait()
    if self.numSecondsHasHuman > 8 and self.IsAcquaintance == False:
        # self.notifyHasHumanUnidentified = True
        if self.sentEmailHasHumanUnidentified == False:
            self.sentEmailHasHumanUnidentified = True
            print("dang gui email")
            self.AlertShow()
            self.timerForSendEmailHasHumanUnidentified.stop()
            sendEmail.sendEmailNotify("database/infoStranger/human.jpg")
            print("da gui email")
            self.AlertClose()
            self.timerForSendEmailHasHumanUnidentified.start(300000)
```

Hình 7: Module thực hiện nhận diện, nhận dạng và gửi email cảnh báo đến chủ nhà

#### 4. Lịch sử ra vào

Như đã đề cập tới ở mục (2a), lịch sử xuất hiện của người ra vào sẽ nằm được lưu ở file Attendance.csv. Thông tin lưu sẽ bao gồm tên và thời gian nhận dạng. Mục đích lưu ở đây là để cho phép chủ nhà có thể xem lại lịch sử ra vào khi cần thiết và hiển thị phân lịch sử trên giao diện.

#### 5. Đăng ký mới

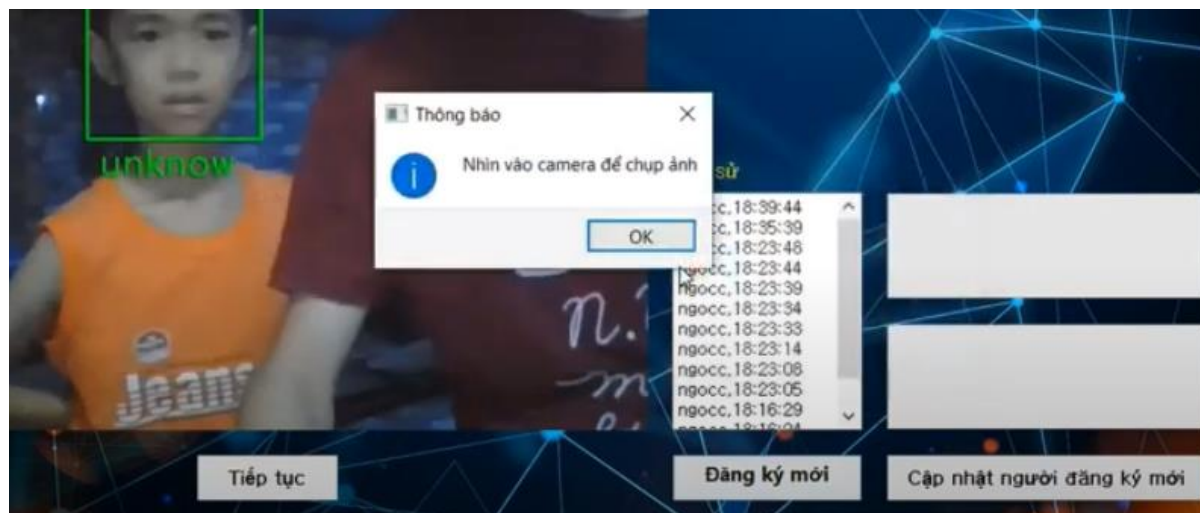
Nút bấm này kích hoạt chức năng đăng ký người dùng mới vào hệ thống.



Hình 8: Giao diện đăng ký mới

Hệ thống yêu cầu người dùng nhập đầy đủ các thông tin cần thiết của người dùng mới, có một điều lưu ý đó là khi ID của người dùng mới trùng với ID của một người dùng khác đã có trong Cơ sở dữ liệu thì hệ thống sẽ thay thế thông tin cũ bằng thông tin mới này.

Sau khi đã điền đầy đủ thông tin và ấn nút Lưu, hệ thống sẽ thông báo yêu cầu người mới nhìn vào Camera 1 để chụp ảnh



Hình 9: Quá trình thu thập ảnh để đăng ký người dùng mới

Ấn tiếp tục thì Camera sẽ xuất hiện và biến mất sau khi lấy đủ 20 khung hình có sự xuất hiện của khuôn mặt người mới. Như vậy là thông tin người dùng mới hệ thống đã có đầy đủ.

#### 6. Cập nhật người đăng ký mới

Nút bấm này làm cho hệ thống chuyển những thông tin hình ảnh cùng ID của người dùng mới vừa được lấy ở chức năng đăng ký mới thành thông tin nhận dạng và lưu ở file trainingData.yml

#### 7. Tiếp tục

Nút này cho phép bạn khởi động lại Camera 1 sau khi nó được tạm dừng để làm nhiệm vụ thu thập hình ảnh người dùng mới ở một giao diện khác.

## 3.2 Công nghệ cốt lõi được sử dụng

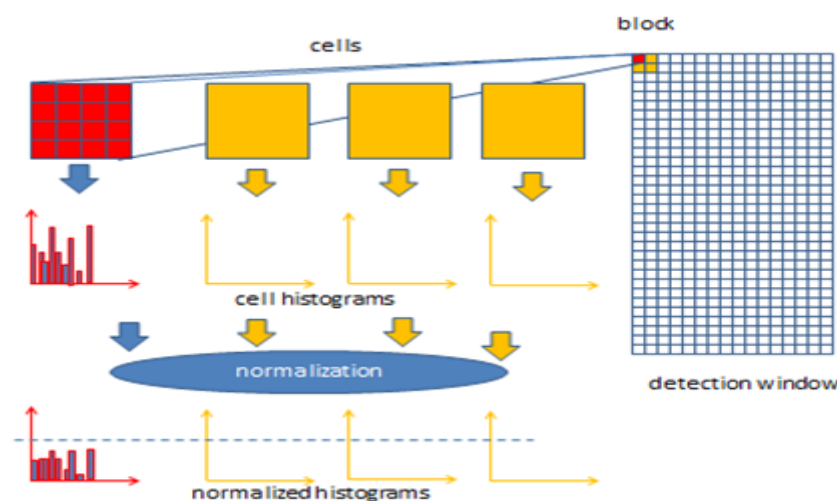
Công nghệ cốt lõi của dự án này là công nghệ nhận diện người và công nghệ nhận dạng khuôn mặt. Có rất nhiều thuật toán, mô hình để giải quyết bài toán này. Nhóm đã nghiên cứu, tìm hiểu, đánh giá và lựa chọn các thuật toán, mô hình phù hợp với điều kiện hiện tại của nhóm. Việc áp dụng vào việc xây dựng ứng dụng cũng được đề cập bên dưới.

### 3.2.1. Nhận diện người

#### 3.2.1.1 Tổng quan

Bộ mô tả HOG có một vài lợi thế so với các bộ mô tả khác. Vì nó hoạt động trên các ô cục bộ, nó bất biến đối với các phép biến đổi hình học, thay đổi độ sáng. Hơn nữa, Dalal và Triggs đã phát hiện ra, khi sử dụng phép chuẩn hóa trên vùng cục bộ sẽ cho phép chuyển động cơ thể của người đi bộ được loại bỏ miễn là họ duy trì được tư thế đứng thẳng. Do đó, bộ mô tả HOG đặc biệt phù hợp để phát hiện con người trong hình ảnh.

#### 3.2.1.2 Thuật toán Histograms of Oriented Gradients (HoG)



Hình 10: Sơ đồ các bước thuật toán HoG

- Chia hình ảnh thành các vùng nhỏ được nối với nhau được gọi là các ô và với mỗi ô, tính toán một biểu đồ về hướng gradient hoặc hướng cạnh cho các pixel trong ô.
- Rời từng ô thành các thùng góc cạnh theo hướng gradient.
- Mỗi pixel của ô đóng góp gradient có trọng số vào thùng góc tương ứng của nó.
- Các nhóm ô liên kề được coi là vùng không gian gọi là khối. Việc nhóm các ô thành một khối là cơ sở để nhóm và chuẩn hóa biểu đồ.
- Nhóm biểu đồ chuẩn hóa đại diện cho biểu đồ khối. Tập hợp các biểu đồ khối này đại diện cho bộ mô tả.

### 3.2.1.3 Áp dụng

Trong dự án này, nhóm sử dụng mô hình pretrained - SVM dự báo dựa trên đầu vào là đặc trưng của ảnh được trích xuất từ thuật toán HOG. Các mô hình đã được tích hợp sẵn vào opencv nên khá đơn giản để áp dụng.

```
HOGCV = cv2.HOGDescriptor()
HOGCV.setSVMDetector(cv2.HOGDescriptor_getDefaultPeopleDetector())

def detect(frame):
    bounding_box_coordinates, weights = HOGCV.detectMultiScale(frame, winStride = (4, 4),
                                                                padding = (8, 8), scale = 1.03)

    person = 1
    for x,y,w,h in bounding_box_coordinates:
        cv2.rectangle(frame, (x,y), (x+w,y+h), (0,255,0), 2)
        person += 1
    hasHuman = False
    if person > 1: hasHuman = True
    return frame, hasHuman
```

Hình 11: Module nhận diện người

Hàm HOGDescriptor() sẽ khởi tạo một bộ mô tả đặc trưng theo thuật toán HOG. Sau đó hàm setSVMDetector() để thiết lập mô hình pretrained dựa trên thuật toán SVM. Cuối cùng ta thu được một mô hình phát hiện người trên các bức ảnh. Hình ảnh người nhận diện được sẽ được bao quanh bởi hộp giới hạn màu xanh.

## 3.2.2. Nhận diện khuôn mặt (Face Detection)

### 3.2.2.1 Tổng quan

Mục tiêu của nhận diện khuôn mặt (Face Detection) đó là tìm kiếm các khuôn mặt (vị trí và kích thước) trong một hình ảnh và có thể trích xuất chúng để sử dụng thuật toán nhận dạng khuôn mặt (Face Recognition).

Dựa vào kết quả trong [bài viết của Vikas Gupta](#) về kết quả thu được sau khi thử nghiệm đối với các phương pháp khác nhau: Haar Cascade Face Detector của OpenCV, Deep Learning based Face Detector của OpenCV, HoG Face Detector của Dlib và Deep Learning based Face Detector của Dlib. Nhóm đã thống kê, so sánh lại thành bảng với các tiêu chí sau:

	Haar Cascade	DNN	HoG	CNN
Độ chính xác	Thấp	Cao	Cao	Cao
Tốc độ	Trung bình	Real-time trên CPU	Nhanh nhất trên CPU	Chậm trên CPU Nhanh trên GPU
Hoạt động đối với khuôn mặt không trực diện	Không	Có	Ít	Không
Hoạt động đối với khuôn mặt bị che khuất	Không	Có	Ít	Không
Kích thước yêu cầu	Tất cả	Tất cả	Trung bình và lớn	Trung bình và lớn

*Bảng 2: Bảng so sánh các phương pháp Face Detector*

Có thể thấy rằng Haar Cascade đứng cuối trong hầu hết các tiêu chí, điều này là dễ hiểu vì đây là thuật toán lâu đời nhất và đơn giản nhất. DNN khắc phục tất cả các điểm yếu của Haar Cascade, nó dựa trên Haar Cascade và mang đầy đủ các ưu điểm của phương pháp này. HoG và CNN chạy nhanh hơn tuy nhiên không làm tốt với các khuôn mặt bị che khuất hoặc không trực diện. Với điều kiện cơ sở vật chất hiện tại, nhóm nhận thấy DNN là phương pháp phù hợp nhất để sử dụng.

### 3.2.2.2 Mô hình DNN Face Detector

Mô hình DNN Face Detector đã được đưa vào OpenCV từ phiên bản 3.3. Nó dựa trên máy dò Single-Shot-Multibox và sử dụng Kiến trúc ResNet-10 làm xương sống. Mô hình đã được đào tạo bằng cách sử dụng các hình ảnh có sẵn trên web, tuy nhiên nguồn gốc của những tấm ảnh này không được tiết lộ. OpenCV cung cấp 2 phiên bản:

- Phiên bản 16. của triển khai caffe gốc (5,4 MB)
- Phiên bản lượng tử hóa 8 bit sử dụng Tensorflow (2,7 MB)

### 3.2.2.3 Áp dụng

Đoạn mã dưới đây để tải mô hình. Nhóm ưu tiên sử dụng phiên bản mô hình dấu chấm động của Caffe, sử dụng các tệp caffemodel và prototxt. Bên cạnh đó, có thể sử dụng mô hình lượng tử hóa Tensorflow.

```

DNN = "TF"
if DNN == "CAFFE":
    modelFile = "res10_300x300_ssd_iter_140000_fp16.caffemodel"
    configFile = "deploy.prototxt"
    net = cv2.dnn.readNetFromCaffe(configFile, modelFile)
else:
    modelFile = "opencv_face_detector_uint8.pb"
    configFile = "opencv_face_detector.pbtxt"
    net = cv2.dnn.readNetFromTensorflow(modelFile, configFile)

```

Hình 12: Module tải mô hình

Hình ảnh gương mặt sau khi được detect sẽ được mô tả trong một hộp giới hạn màu xanh lục:

```

profile = detectFace.getProfile(id)
#load info to camera
if (profile != None):
    self.isnew = Attendance.markAttendance(str(profile[1]), self.isnew)
    cv2.putText(img, "Name: " + str(profile[1]), (x + 10, y + h + 30), fontFace, 1, (0, 255, 0), 2)
    cv2.putText(img, "Age: " + str(profile[2]), (x + 10, y + h + 60), fontFace, 1, (0, 255, 0), 2)
    cv2.putText(img, "Gender: " + str(profile[3]), (x + 10, y + h + 90), fontFace, 1, (0, 255, 0), 2)

```

Hình 13: Đoạn code Detect Face

Trong dự án thực tế, đây chỉ là bước trung gian trong phần nhận diện khuôn mặt. Kết quả của bước này là tiền đề để xác minh danh tính ở bước sau đó. Bởi vậy, phần code trên có thêm các thông tin của người dùng được hiển thị bên dưới hộp giới hạn.

### 3.2.3. Nhận dạng khuôn mặt (Face Recognition)

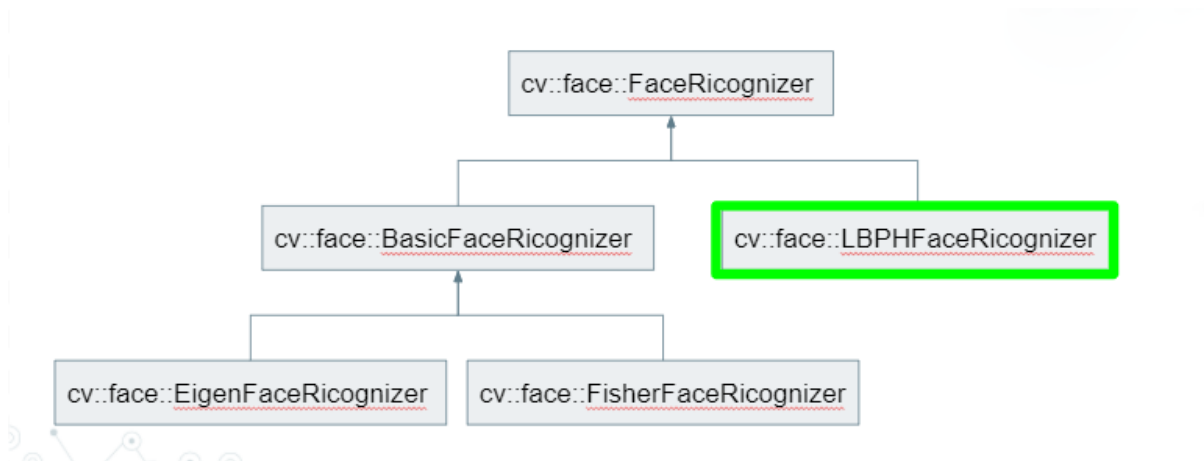
#### 3.2.3.1 Tổng quan

Nhận dạng khuôn mặt chịu trách nhiệm tìm kiếm các đặc điểm mô tả hình ảnh tốt nhất trên các hình ảnh có khuôn mặt đã được trích xuất, cắt, thay đổi kích thước và thường được chuyển đổi sang thang độ xám.

Hệ thống nhận dạng khuôn mặt có thể hoạt động về cơ bản ở hai chế độ:

- Xác minh hoặc xác thực hình ảnh khuôn mặt : nó so sánh hình ảnh khuôn mặt đầu vào với hình ảnh khuôn mặt liên quan đến người dùng đang yêu cầu xác thực. Về cơ bản nó là một so sánh 1vs1.
- Nhận dạng hoặc nhận dạng khuôn mặt : nó so sánh hình ảnh khuôn mặt đầu vào với tất cả các hình ảnh khuôn mặt từ một tập dữ liệu với mục đích tìm ra người dùng phù hợp với khuôn mặt đó. Về cơ bản nó là một so sánh 1vsN.

Có rất nhiều thuật toán khác nhau để nhận dạng khuôn mặt như: Eigenfaces (1991), Local Binary Patterns Histograms (LBPH) (1996), Fisherfaces (1997), Scale Invariant Feature Transform (SIFT) (1999), Speed Up Robust Features (SURF) (2006).



Hình 14: Các thuật toán nhận dạng khuôn mặt hỗ trợ trong OpenCV

Nhóm lựa chọn phương pháp lâu đời và dễ tiếp cận đó là thuật toán Local Binary Patterns Histograms (LBPH).

### 3.2.3.2 Thuật toán Local Binary Patterns Histograms (LBPH)

#### a. Tham số

Radius: bán kính được sử dụng để xây dựng mẫu nhị phân cục bộ hình tròn và đại diện cho bán kính xung quanh pixel trung tâm. Nó thường được đặt thành 1.

Vùng lân cận: số lượng điểm mẫu để xây dựng mẫu nhị phân cục bộ hình tròn. Càng nhiều điểm mẫu, chi phí tính toán càng cao, thường được đặt bằng 8.

Grid X: số ô theo chiều ngang. Càng nhiều ô, lưới càng mịn, thì số chiều của vector đặc trưng thu được càng cao, thường được đặt bằng 8.

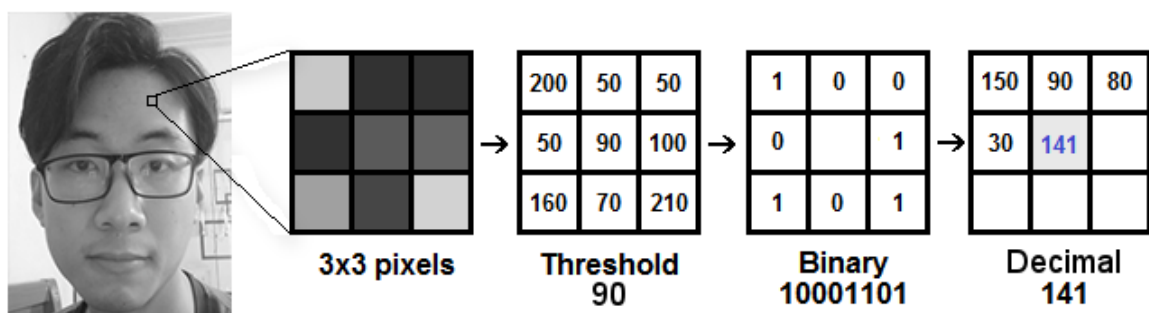
Grid Y: số ô theo chiều dọc. Càng nhiều ô, lưới càng mịn, thì số chiều của vector đặc trưng thu được càng cao, thường được đặt bằng 8.

#### b. Chuẩn bị tập huấn luyện

Đầu tiên, cần có một bộ dữ liệu với hình ảnh khuôn mặt của những người cần nhận dạng. Gán nhãn (ID hoặc tên người) cho mỗi hình ảnh.

#### c. Áp dụng LBP

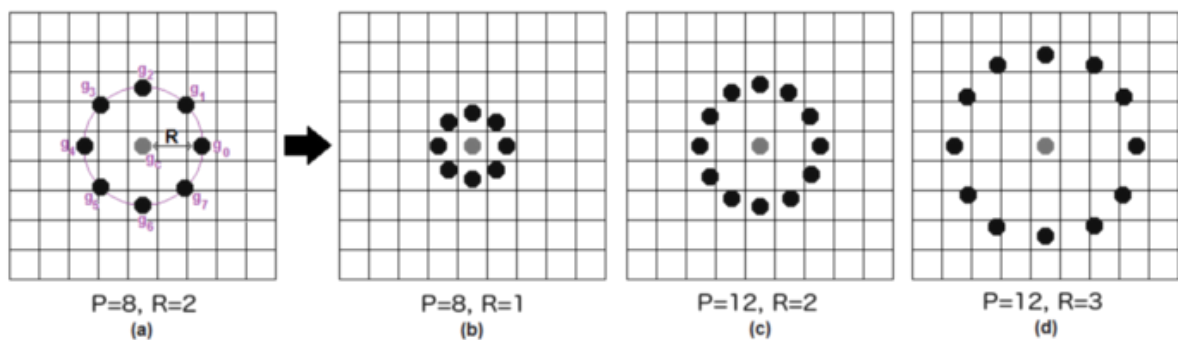
Bước tính toán đầu tiên của LBPH là tạo ra một hình ảnh trung gian mô tả hình ảnh gốc theo cách tốt hơn, bằng cách làm nổi bật các đặc điểm trên khuôn mặt. Để làm như vậy, thuật toán sử dụng khái niệm cửa sổ trượt, dựa trên bán kính tham số và các vùng lân cận.



Hình 15: Quy trình LBP



- Chuyển ảnh về thang xám.
- Lấy một phần hình ảnh dưới dạng cửa sổ 3x3 pixels. Nó cũng có thể biểu diễn dưới dạng ma trận 3x3 chứa cường độ của mỗi pixels (0~255)
- Chọn giá trị trung tâm của ma trận làm ngưỡng để xác định các giá trị mới từ 8 lân cận.
- Với mỗi giá trị lân cận, đặt một giá trị nhị phân mới: 1 nếu giá trị bằng hoặc cao hơn giá trị của trung tâm và 0 nếu giá trị nhỏ hơn giá trị của trung tâm.
- Nối từng giá trị nhị phân từ mỗi vị trí từ dòng ma trận theo từng dòng để có một giá trị nhị phân mới (ví dụ: 10001101).
- Chuyển đổi giá trị nhị phân này thành giá trị thập phân và đặt nó vào trung tâm ma trận, thực chất là một pixels từ ảnh gốc.
- Cuối cùng, kết quả thu được là một hình ảnh mới thể hiện tốt hơn các đặc điểm của hình ảnh gốc



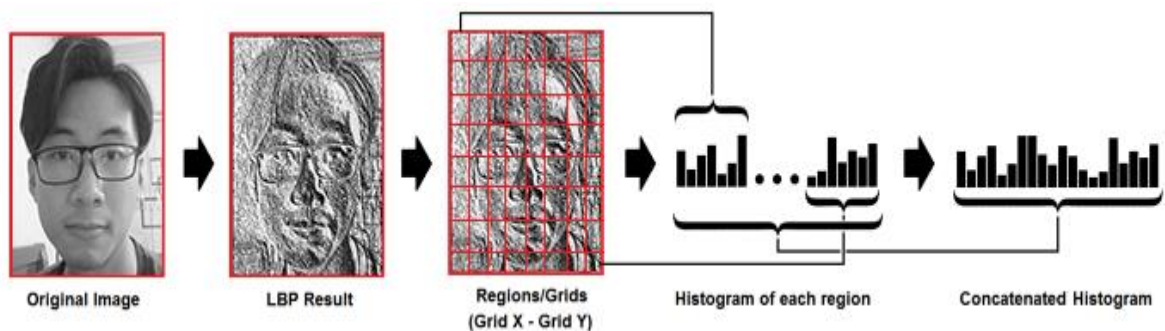
Hình 16: LBP mở rộng

Thủ tục LBP được mở rộng để sử dụng một số bán kính và lân cận khác nhau, nó được gọi là LBP tròn.

#### d. Trích xuất biểu đồ

Dựa trên hình ảnh trên, có thể trích xuất biểu đồ của từng vùng như sau:

- Khi chúng ta có một hình ảnh ở thang độ xám, mỗi biểu đồ (từ mỗi lưới) sẽ chỉ chứa 256 vị trí (0 ~ 255) đại diện cho các lần xuất hiện của mỗi cường độ pixel.
- Sau đó, nối từng biểu đồ để tạo một biểu đồ mới và lớn hơn. Giả sử có 8x8 lưới, chúng ta sẽ có  $8 \times 8 \times 256 = 16.384$  vị trí trong biểu đồ cuối cùng. Biểu đồ cuối cùng đại diện cho các đặc điểm của hình ảnh gốc.



Hình 17: Quá trình trích xuất biểu đồ



e. Thực hiện nhận dạng khuôn mặt

Mỗi biểu đồ được tạo được sử dụng để đại diện cho từng hình ảnh từ tập dữ liệu đào tạo. Với một hình ảnh đầu vào, thực hiện lại các bước cho hình ảnh mới này và tạo một biểu đồ đại diện cho hình ảnh.

Vì vậy, để tìm hình ảnh phù hợp với hình ảnh đầu vào, chỉ cần so sánh hai biểu đồ và trả về hình ảnh có biểu đồ gần nhất. Có thể sử dụng khoảng cách Euclid (đã được biết đến khá nhiều) theo công thức sau:

$$D = \sqrt{\sum_{i=1}^n (hist1_i - hist2_i)^2}$$

### 3.2.3.3 Áp dụng

Như đã nêu ở phần trên, kết quả của việc nhận diện khuôn mặt là tiền đề cho việc nhận dạng khuôn mặt dưới đây.

```
def getProfile(id):  
    db = sqlite3.connect("database/database.db")  
    query = "Select * from People Where ID= " + str(id)  
    cursor = db.execute(query)  
  
    profile = None  
    for row in cursor:  
        profile = row  
  
    db.close()  
    return profile
```

Hình 18: Module getProfile()

Hàm getProfile() dùng để kiểm tra gương mặt có nằm trong cơ sở dữ liệu hay không.

```
recognizer = cv2.face_LBPHFaceRecognizer.create()  
recognizer.read('recognizer/trainingData.yml')
```

Hình 19: face\_LBPHFaceRecognizer trong cv2

Nhận diện khuôn mặt trở nên đơn giản nhờ face\_LBPHFaceRecognizer đã được tích hợp trong thư viện OpenCV.

```

for (x, y, w, h) in faces:
    cv2.rectangle(img, (x,y), (x+w, y+h), (0,255,0), 2)
    id, configparser = recognizer.predict(gray[y:y+h, x:x+w])
    if configparser < 40:
        profile = detectFace.getProfile(id)
        #load info to camera
        if (profile != None):
            self.isnew = Attendance.markAttendance(str(profile[1]), self.isnew)
            cv2.putText(img, "Name: " + str(profile[1]), (x + 10, y + h + 30), fontFace, 1, (0, 255, 0), 2)
            cv2.putText(img, "Age: " + str(profile[2]), (x + 10, y + h + 60), fontFace, 1, (0, 255, 0), 2)
            cv2.putText(img, "Gender: " + str(profile[3]), (x + 10, y + h + 90), fontFace, 1, (0, 255, 0), 2)
        # Kích hoạt biến IsAcquaintance True
        self.IsAcquaintance = True
        self.timerForSetIsAcquaintanceFalse.stop()
        self.timerForSetIsAcquaintanceFalse.start(300000) #cho phép giữ trạng thái True trong 3 phút
    else:
        cv2.putText(img, "Unknown", (x + 10, y + h + 30), fontFace, 1, (0,255,0), 2)
        cv2.imwrite('database/infoStranger/Stranger.jpg', img)
        self.hasStranger = True

```

Hình 20: Module xử lý Face Recognition

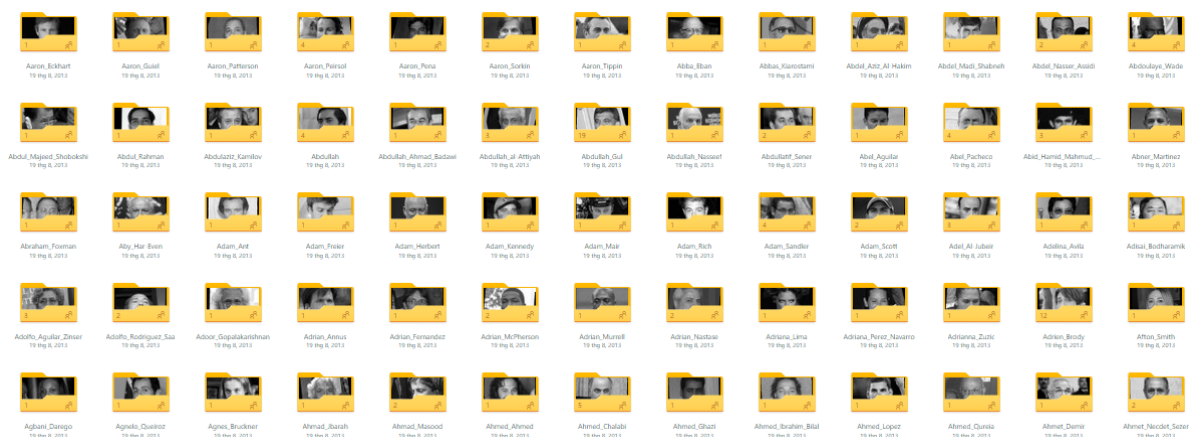
Nếu nhận diện thành công, khuôn mặt của người trước camera được bao quanh bởi một hộp giới hạn, kèm theo thông tin của người đó (tên, tuổi, giới tính). Biến IsAcquaintance được cập nhật lại, mang giá trị true.

Nếu nhận diện không thành công, khuôn mặt của người trước camera được bao quanh bởi hộp giới hạn kèm theo dòng chữ “Unknown”. Biến IsAcquaintance không được cập nhật, từ đó các hành động cảnh báo đến người dùng như gửi email chứa hình ảnh có người đứng trước cửa từ camera 1, hình ảnh người lạ không thể nhận dạng từ camera 2, thông báo đến người đứng trước camera việc nhận dạng không thành công bằng giọng nói.

### 3.3 Thử nghiệm trên tập dữ liệu lớn

### 3.3.1 Tổng quan về bộ dữ liệu

Bộ dữ liệu hình ảnh "Labeled Faces in the Wild-a (LFW-a)" là một cơ sở dữ liệu gồm các hình ảnh khuôn mặt được gán nhãn nhằm mục đích nghiên cứu tính năng Nhận dạng khuôn mặt.



Hình 21: Một phần của bộ dữ liệu LFW-a

Bộ dữ liệu bao gồm 13233 tấm ảnh của 5749 người. Bộ dữ liệu lớn và phong phú, tuy nhiên số lượng ảnh của mỗi người là không đồng đều, cụ thể được thống kê dưới bảng sau:

Số tấm ảnh/ người	1	2	3	4	5	6	7	8	9	10	10-20	>20
Số người	4068	779	291	187	112	55	39	33	26	15	96	48

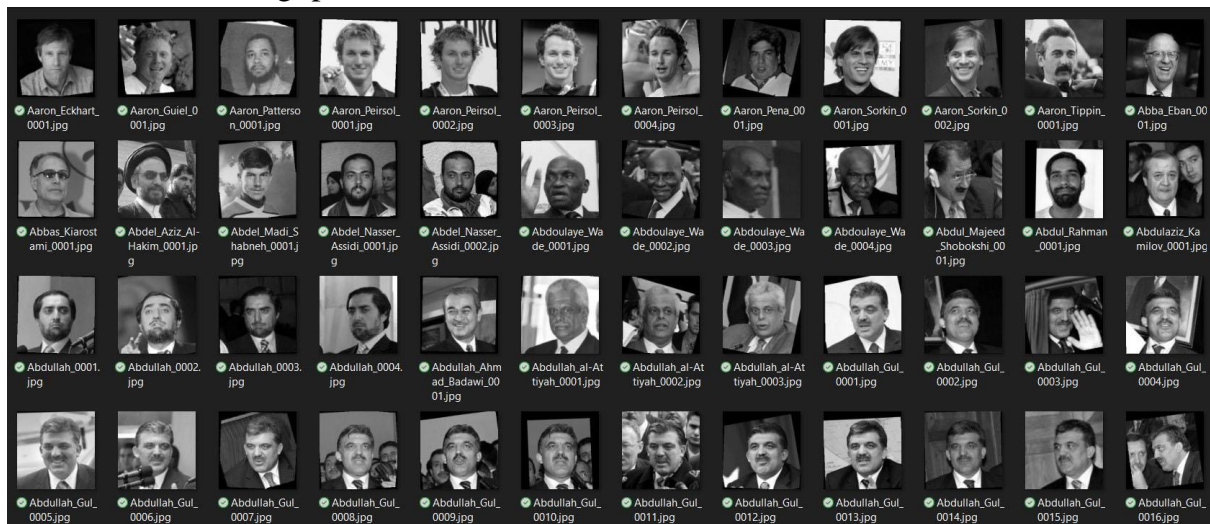
Bảng 22: Thống kê số lượng ảnh / người và số người

Nhóm nhận thấy bộ dữ liệu trên không đồng đều, bởi vậy nhóm chia thành 2 hướng để thử nghiệm:

- Thử nghiệm trên bộ dữ liệu hình ảnh ban đầu (bộ này đặt tên là Data\_NotFair)
- Lọc các dữ liệu hình ảnh của những người có số lượng ảnh nằm trong khoảng 10 đến 20 ảnh và thực hiện (bộ này đặt tên là Data\_Fair)

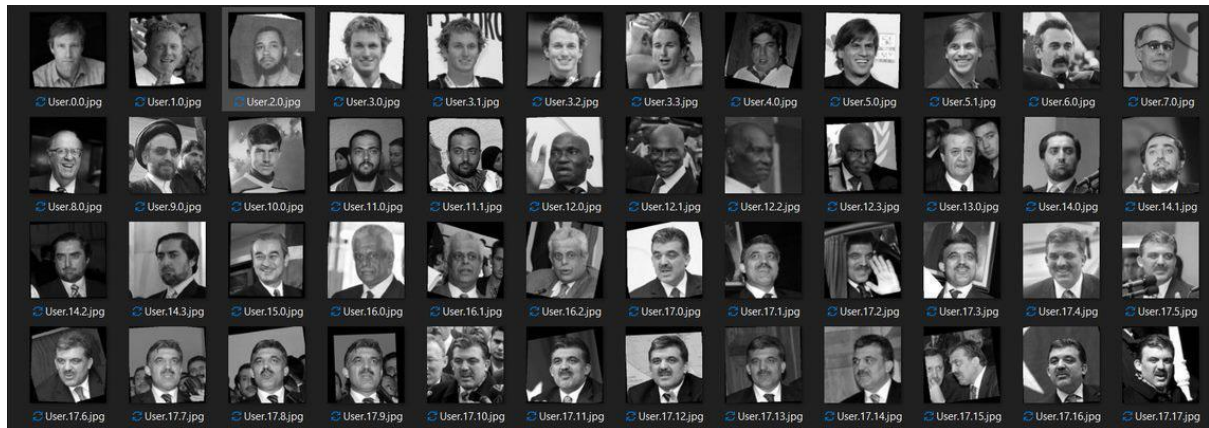
### 3.3.2 Xử lý dữ liệu

Đầu tiên dữ liệu được gộp lại vào một thư mục:



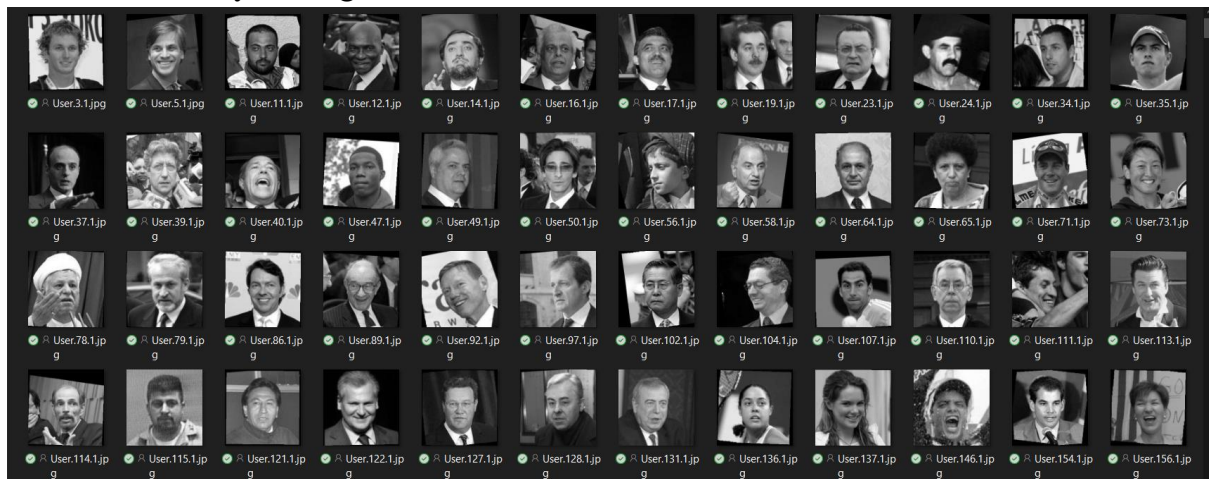
Hình 23: Một phần của thư mục chứa toàn bộ hình ảnh

Tiếp theo dữ liệu được đánh lại Id để phù hợp với mô hình huấn luyện:



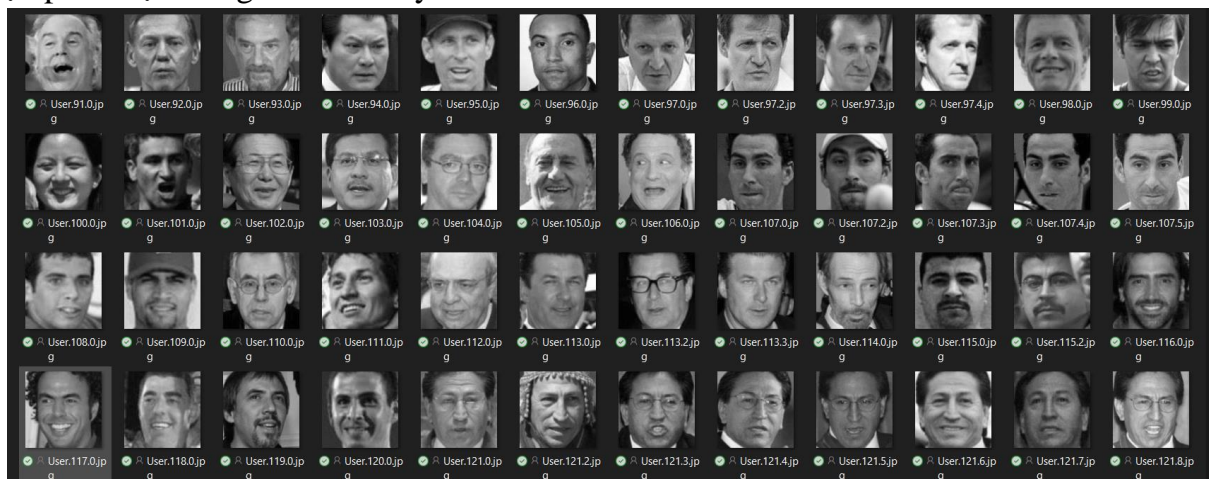
Hình 24: Các hình ảnh đã được đổi Id

Tiếp theo, chia các ảnh thành 2 tập train và test: ý tưởng là với những người có hơn hai bức ảnh thì bức ảnh có chỉ số là 1 sẽ được di chuyển sang thư mục Data\_Test, tất cả các hình ảnh còn lại sẽ được chuyển sang thư mục Data\_Train



Hình 25: Một phần dữ liệu tập test

Cuối cùng, thực hiện việc thay thế từng bức ảnh trong tập Train bằng hình ảnh khuôn mặt được phát hiện trong bức ảnh đấy.



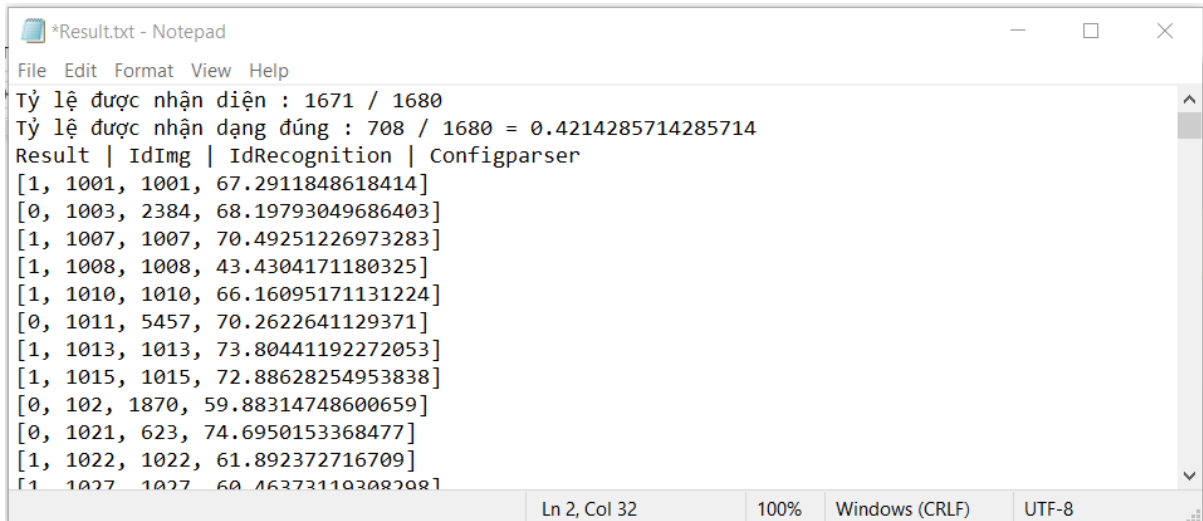
Hình 26: Một phần dữ liệu tập Train sau khi đã xử lý



Công việc trên được thực hiện đối với cả 2 hướng thử nghiệm. Sau khi xử lý, dữ liệu được huấn luyện và thử nghiệm.

### 3.3.3 Kết quả thu được

- Với hướng thử nghiệm trên tập dữ liệu nguyên thủy ban đầu, trong thư mục Data\_NotFair, kết quả thu được khá thấp chỉ là 42,14 %

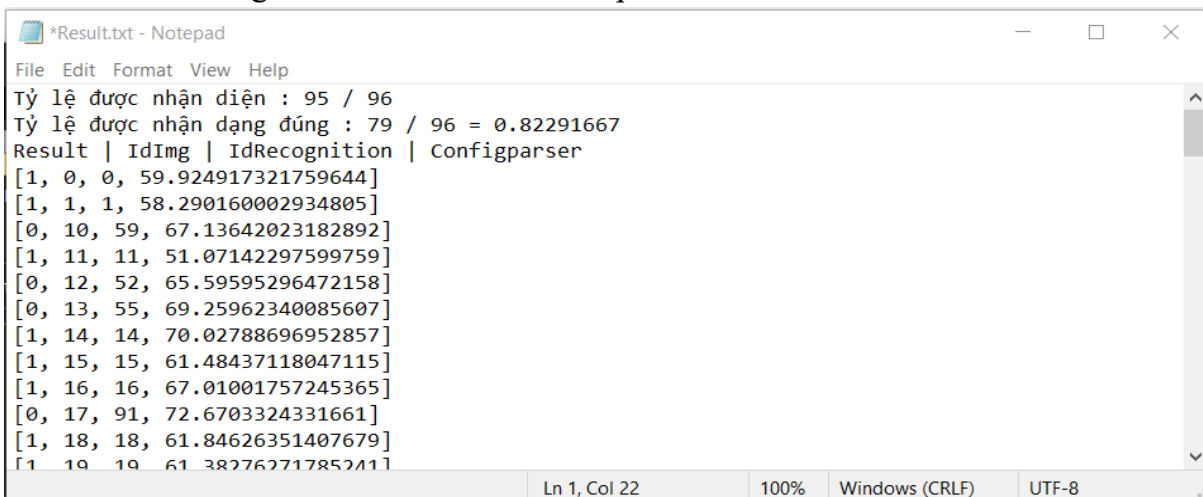


```
*Result.txt - Notepad
File Edit Format View Help
Tỷ lệ được nhận diện : 1671 / 1680
Tỷ lệ được nhận dạng đúng : 708 / 1680 = 0.4214285714285714
Result | IdImg | IdRecognition | Configparser
[1, 1001, 1001, 67.2911848618414]
[0, 1003, 2384, 68.19793049686403]
[1, 1007, 1007, 70.49251226973283]
[1, 1008, 1008, 43.4304171180325]
[1, 1010, 1010, 66.16095171131224]
[0, 1011, 5457, 70.2622641129371]
[1, 1013, 1013, 73.80441192272053]
[1, 1015, 1015, 72.88628254953838]
[0, 102, 1870, 59.88314748600659]
[0, 1021, 623, 74.6950153368477]
[1, 1022, 1022, 61.892372716709]
[1, 1027, 1027, 60.46373110308208]
```

Hình 27: Kết quả thử nghiệm trên tập Data\_NotFair

Đây là kết quả hơi thấp nhưng hoàn toàn dễ hiểu vì có lý do lớn nhất đó là dữ liệu quá mất cân bằng, có quá nhiều người chỉ chứa một bức ảnh, điều này gây thêm rất nhiều nhiễu cho việc nhận dạng những người còn lại.

- Với hướng thử nghiệm trên tập dữ liệu cân bằng hơn được trích lọc từ tập dữ liệu ban đầu, nằm trong thư mục Data\_Fair, kết quả thu được là 82.29 %



```
*Result.txt - Notepad
File Edit Format View Help
Tỷ lệ được nhận diện : 95 / 96
Tỷ lệ được nhận dạng đúng : 79 / 96 = 0.82291667
Result | IdImg | IdRecognition | Configparser
[1, 0, 0, 59.924917321759644]
[1, 1, 1, 58.290160002934805]
[0, 10, 59, 67.13642023182892]
[1, 11, 11, 51.07142297599759]
[0, 12, 52, 65.59595296472158]
[0, 13, 55, 69.25962340085607]
[1, 14, 14, 70.02788696952857]
[1, 15, 15, 61.48437118047115]
[1, 16, 16, 67.01001757245365]
[0, 17, 91, 72.6703324331661]
[1, 18, 18, 61.84626351407679]
[1, 19, 19, 61.382762717852411]
```

Hình 28: Kết quả thử nghiệm trên tập Data\_Fair

Có thể thấy kết quả đã được nâng cao hơn rất nhiều, hai lý do có thể hiểu được đó là hệ thống đã có nhiều bức ảnh hơn để training nhận dạng một người và lý do thứ hai là kích thước của tập data đã nhỏ hơn rất nhiều.

## 4. KẾT LUẬN

Qua quá trình hoàn thành dự án nghiên cứu, xây dựng ứng dụng "Nhận dạng khuôn mặt hỗ trợ camera an ninh gia đình", nhóm đã thu thập được nhiều kiến thức và kinh nghiệm quý báu trong việc nghiên cứu các công nghệ nhận diện người, nhận dạng khuôn mặt cũng như xây dựng được ứng dụng hỗ trợ cho camera an ninh gia đình. Cụ thể như sau:

- Nhóm đã nghiên cứu công nghệ nhận diện người và công nghệ nhận dạng khuôn mặt, đồng thời tìm hiểu, so sánh điểm mạnh, điểm yếu của các thuật toán, mô hình có thể giải quyết bài toán này. Từ đó, nhóm đã lựa chọn và áp dụng phương pháp phù hợp nhất với dự án.
- Từ những kiến thức về công nghệ đã tìm hiểu được, nhóm đã xây dựng thành công ứng dụng "Nhận dạng khuôn mặt hỗ trợ camera an ninh gia đình". Trong quá trình đó, nhóm đã nghiên cứu và sử dụng hệ cơ sở dữ liệu Sqlite 3 để xây dựng cơ sở dữ liệu đơn giản lưu trữ thông tin người dùng. Bên cạnh đó, nhóm đã nghiên cứu, sử dụng thư viện PyQt5 để tạo một ứng dụng với giao diện thân thiện, đầy đủ chức năng cần thiết.
- Ngoài ra, nhóm cũng đã thử nghiệm qua bộ dữ liệu lớn Labeled Faces in the Wild-a (LFW-a) và đạt được kết quả tương đối khả quan

Tuy nhiên hệ thống vẫn còn một số hạn chế, đồng thời cũng là kế hoạch nâng cấp trong tương lai, như sau:

- Về chất lượng của ứng dụng: Ứng dụng chạy mượt mà và trơn chu với điều kiện hiện tại. Tuy nhiên, để ứng dụng có thể áp dụng vào thực tế, cần có thời gian và điều kiện để nghiên cứu chi tiết hơn trong điều kiện thực tế. Các thuật toán, mô hình đang sử dụng hiện tại vì vậy cũng cần lựa chọn, xây dựng lại cho tốt hơn.
- Về kết quả sau khi thử nghiệm trên tập dữ liệu lớn: Tập dữ liệu còn nhiều thiếu sót khiến kết quả không cao, tuy nhiên đó là tập dữ liệu duy nhất chúng em tìm kiếm được trong quá trình thực hiện dự án. Kết quả tuy tương đối khả quan nhưng chưa đạt được kỳ vọng ban đầu của nhóm là mức 95%. Nếu trong điều kiện thực tế, con số này chắc chắn cần được nâng thêm để đáp ứng tính an toàn và bảo mật.