**TRƯỜNG ĐẠI HỌC TÀI NGUYÊN VÀ MÔI TRƯỜNG**

**THÀNH PHỐ HỒ CHÍ MINH**

**KHOA HỆ THỐNG THÔNG TIN VÀ VIỄN THÁM**

****

**BÁO CÁO CUỐI KÌ**

**NGHIÊN CỨU MÔ HÌNH XÁC THỰC SINH TRẮC HỌC ĐA NHÂN TỐ**

**Giảng viên hướng dẫn : Ths.Phạm Trọng Huynh**

**Sinh viên thực hiện : Nguyễn Trần Hưng 0950080122**

**Trần Thanh Lâm 0950080127**

**Nguyễn Hoàng Minh Tuấn 0950080080**

**Lớp 09\_ĐH\_CNPM3**

**Khóa 2020 - 2024**

*Tp. Hồ Chí Minh, ngày 10 tháng 4 năm 2024*

BẢN NHẬN XÉT CỦA GIÁO VIÊN HƯỚNG DẪN

|  |  |  |
| --- | --- | --- |
| Stt | Nội dung thực hiện | Nhận xét |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

Nhận xét tổng quát:

**mục lục**

[**CÁC TỪ VIẾT TẮT** iii](#_Toc164726736)

[Chương 1: Tổng quan 1](#_Toc164726737)

[1.1. giới thiệu 1](#_Toc164726738)

[1.2. mục tiêu đề tài 1](#_Toc164726739)

[1.3. GIỚI HẠN ĐỀ TÀI 1](#_Toc164726740)

[1.4. PHƯƠNG PHÁP NGHIÊN CỨU 1](#_Toc164726741)

[1.5. ĐỐI TƯỢNG VÀ PHẠM VI NGHIÊN CỨU 2](#_Toc164726742)

[1.6. BỐ CỤC QUYỂN BÁO CÁO 2](#_Toc164726743)

[Chương 2: CƠ SỞ LÝ THUYẾT 3](#_Toc164726744)

[2.1. Giới thiệu về sinh trắc học 3](#_Toc164726745)

[2.2. Hệ thống sinh trắc học 3](#_Toc164726746)

[2.2.1. Khái quát về hệ thống sinh trắc học 3](#_Toc164726747)

[2.2.2. Các đặc điểm của hệ thống sinh trắc học 5](#_Toc164726748)

[2.3. Đánh giá hiệu năng và chất lượng hoạt động của hệ sinh trắc học 9](#_Toc164726749)

[2.3.1. Vấn đề lỗi trong hoạt động của hệ sinh trắc 9](#_Toc164726750)

[2.3.2. Các tham số đánh giá chất lượng 9](#_Toc164726751)

[2.4 Nghiên cứu các mô hình sinh trắc 10](#_Toc164726752)

[2.4.1 Nhận dạng khuôn mặt 10](#_Toc164726753)

[2.4.2 Nhận dạng vân tay 12](#_Toc164726754)

[2.5 Nghiên cứu các mô hình sinh trắc học đa nhân tố 14](#_Toc164726755)

[2.5.1 Mô hình xác thực sinh trắc học đa nhân tố kết hợp mức đặc trưng 14](#_Toc164726756)

[2.5.2 Mô hình xác thực sinh trắc học đa nhân tố kết hợp mức điểm số 15](#_Toc164726757)

[2.5.3 Mô hình xác thực sinh trắc học đa nhân tố kết hợp mức quyết định 15](#_Toc164726758)

[2.5.4 Ưu điểm mô hình xác thực sinh trắc học đa nhân tố 16](#_Toc164726759)

[Chương 3: xây dựng chương trình 18](#_Toc164726760)

[3.1. xây dựng mô hình 18](#_Toc164726761)

[3.2. Các bước thực hiện 18](#_Toc164726762)

[3.2.1. Tạo tập dữ liệu 18](#_Toc164726763)

[3.2.2. Huấn luyện dữ liệu đưa vào 25](#_Toc164726764)

[3.2.3. Hiển thị dữ liệu đầu ra 30](#_Toc164726765)

[Chương 4: Kết quả thí nghiệm 32](#_Toc164726766)

[4.1. Kết quả sau khi huấn luyện 32](#_Toc164726767)

[4.2. Sử dụng phần mềm 33](#_Toc164726768)

[Chương 5: KẾT LUẬN 36](#_Toc164726769)

[PHỤ LỤC 37](#_Toc164726770)

[TÀI LIỆU THAM KHẢO 43](#_Toc164726771)

CÁC TỪ VIẾT TẮT

|  |  |
| --- | --- |
| ReLu | Rectified Linear Unit |
| CNN | Convolutional Neural Network |
| OpenCV | Open Source Computer Vision |
| CV | Computer Vision |
| MLL | Machine Learning Library |
| CONV | Convolution |

# Tổng quan

## giới thiệu

Sinh trắc học là độ đo các đặc điểm hành vi và vật lý độc nhất của con người như vân tay, giọng nói, và mống mắt, từ đó xác định và phân biệt cá nhân. Công nghệ này giúp tránh rủi ro như mất mật khẩu bằng cách sử dụng các thiết bị tự động. Hiện nay, các công nghệ như nhận diện khuôn mặt, vân tay, và mống mắt đang được sử dụng, cùng với việc phát triển các phương pháp như gân mu bàn tay, DNA, và dáng điệu. Mục tiêu là kết hợp nhiều công nghệ sinh trắc với nhau và tích hợp chúng vào hệ thống PKI để tạo ra hệ thống BioPKI đảm bảo tính bảo mật và xác thực trong hoạt động mạng.

## mục tiêu đề tài

Nghiên cứu các mô hình sinh trắc học có thể áp dụng trong tương lai.

- Hệ thống an ninh

- Phân tích dữ liệu

## GIỚI HẠN ĐỀ TÀI

Phạm vi giới hạn của đề tài:

Phạm vi của đề tài là sử dụng Python và các mô hình máy học để xây dựng hệ thống xác thực sinh trắc học đa nhân tố, chủ yếu tập trung vào việc xử lý hình ảnh và dữ liệu biometric từ các nguồn giới hạn. Đối tượng là tích hợp các phương pháp này vào các hệ thống lớn như PKI để đảm bảo tính bảo mật.

## PHƯƠNG PHÁP NGHIÊN CỨU

Trong phần này, chúng tôi thiết kế mô hình nhận dạng khuôn mặt dựa trên mạng CNN. Dựa trên các phương pháp nghiên cứu chính như phương pháp phân tích, tham khảo tài liệu, phương pháp tổng hợp tài liệu lý thuyết…

## ĐỐI TƯỢNG VÀ PHẠM VI NGHIÊN CỨU

Các đối tượng cần nghiên cứu và phạm vi nghiên cứu có thể giải quyết được đề tài:

- *Đối tượng nghiên cứu:* Trí tuệ nhân tạo, lập trình python.

- *Phạm vi nghiên cứu:* Mạng CNN, các module trong python: opencv, numpy, os, tensorflow, image.

## BỐ CỤC QUYỂN BÁO CÁO

Nội dung báo cáo:

CHƯƠNG 1: GIỚI THIỆU

CHƯƠNG 2: CƠ SỞ LÝ THUYẾT

CHƯƠNG 3: XÂY DỤNG CHƯƠNG TRÌNH

CHƯƠNG 4: KẾT QUẢ THÍ NGHIỆM

# CƠ SỞ LÝ THUYẾT

## Giới thiệu về sinh trắc học

Sinh trắc học đo các đặc điểm hành vi và vật lý độc nhất của con người như vân tay, giọng nói, và mống mắt, từ đó xác định và phân biệt cá nhân. Công nghệ này giúp tránh rủi ro như mất mật khẩu bằng cách sử dụng các thiết bị tự động. Hiện nay, các công nghệ như nhận diện khuôn mặt, vân tay, và mống mắt đang được sử dụng, cùng với việc phát triển các phương pháp như gân mu bàn tay, DNA, và dáng điệu. Mục tiêu là kết hợp nhiều công nghệ sinh trắc với nhau và tích hợp chúng vào hệ thống PKI để tạo ra hệ thống BioPKI đảm bảo tính bảo mật và xác thực trong hoạt động mạng.

## Hệ thống sinh trắc học

### Khái quát về hệ thống sinh trắc học

Hệ thống sinh trắc học (Biometric System) thực chất là một hệ nhận dạng dựa trên các đặc điểm về hành vi hay thuộc tính vật lý của người cần nhận dạng [9]. Hệ thống sinh trắc học được phân ra thành hai loại chính [13]:

• Hệ thẩm định (Verification): Hệ thống thực hiện đối sánh 1-1 giữa mẫu sinh trắc học thu

nhận được (Biometric sample) với mẫu dạng sinh trắc học (biometric template) đã có trong hệ thống từ trước. Kết quả trả lời câu hỏi mẫu sinh trắc thu nhận có liên quan tới mẫu dạng sinh trắc hay không, thông thường trong hệ thẩm định kết hợp với thông tin định danh chủ thể thực hiện chức năng xác thực thẩm định sinh trắc (Authentication).

Trong hệ xác thưc thẩm định đòi hỏi cao về độ chính xác để kết quả trả lời câu hỏi “sinh trắc học sống thu nhận được (biometric sample) có phải là sinh trắc của chủ thể đã lưu trong hệ thống không?”

• Nhận dạng (Identification, Recognition): Hệ thống thực hiện chức năng tìm kiếm (1-n)

từ một cơ sở dữ liệu đề tìm một mẫu sinh trắc cụ thể trong các mẫu khuôn dạng sinh trắc thu thập từ trước và sau đó thực hiện đối sánh xấp xỉ để nhận dạng phân lớp

(Classification) hoặc nhận dạng đồng nhất (Identification), ví dụ như việc tìm mẫu vân tay

tội phạm trong hồ sơ các vân tay, từ đó xác định danh tính của chủ sở hữu vân tay.

• Sơ đồ khối chức năng của 2 loại hệ thống sinh trắc được minh họa trong Hình Các thành phần chức năng chủ yếu của hệ thống sinh trắc học [13]:

- Thu nhận (Sensor, Capture): thu nhập mẫu sinh trắc học và biểu diễn dưới dạng số hóa.

- Xử lý và trích chọn đặc trưng (Feature Extraction): Thực hiện các phép xử lý phân tích và

trích chọn các đặc trưng từ mẫu sinh trắc học.

- Đối sánh (Matching): thực hiện so sánh các đặc trưng vừa trích chọn với khuôn mẫu sinh

trắc đã có trước.

- Ra quyết định (Decision): dựa trên kết quả đối sánh sẽ khẳng định danh tính người dùng

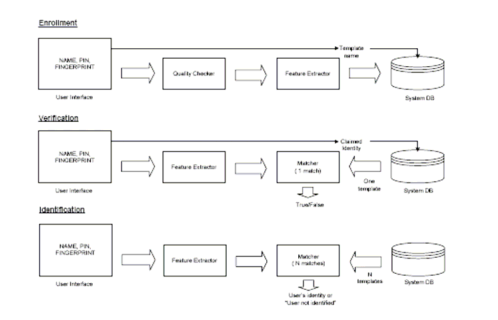
(với hệ nhận dạng) hoặc là một câu trả lời đúng hoặc sai về mẫu sinh trắc học so với

khuôn mẫu sinh trăc có từ trước (với hệ thẩm định).

• Hoạt động của hệ thống sinh trắc bao gồm 2 giai đoạn cơ bản:

- Đăng ký (Enrollment): Đăng ký mẫu sinh trắc vào hệ thống

- Thẩm định hoặc nhận dạng (Verification/ Identification)



*Hình 2.1. Sơ đồ khối chức năng của 2 loại hệ thống sinh trắc.*

### Các đặc điểm của hệ thống sinh trắc học

1. Các vấn đề về thu nhận và biểu diễn mẫu sinh trắc như sau:

Xác thực bằng mật khẩu truyền thống dùng Password không cần sử dụng các phương pháp nhận dạng mẫu phức tạp, mà chỉ cần đối sánh trực tiếp mật khẩu. Cơ chế này cho phép xây dựng hệ xác thức mật khẩu đảm bảo tính chính xác, ổn định, hiệu quả đúng như thiết kế. Tuy nhiên vấn đề không an toàn và điểm yếu nhất của hệ thống là thông thường mật khẩu chỉ gồm 6-8 ký tự, mật khẩu này dẽ dàng bị đánh cắp, bị quên hay bị mạo danh, khi xảy ra mất an toàn mật khẩu, toàn bộ hệ thống an toàn của hệ thống sẽ sụp đổ. Đối với sinh trắc học, mẫu sinh trắc có tính bền vững cao, khó giả mạo dịnh danh và chp phép đảm bảo an toàn cho hệ thông. Mặt khác khi thu nhận các mẫu sinh trắc sống và xử lý biểu diễn trích chọn đặc trưng, các kết quả này phụ thuộc rất nhiều vào yếu tố như phương pháp lấy mẫu,

môi trường lấy mẫu, trạng thái tương tác của người lấy mẫu với thiết bị và tùy theo loại sinh

trắc thu nhận [7,10].

• Thu nhận mẫu sinh trắc không ổn định

Như đã nói, tín hiệu sinh trắc học thu nhận được phụ thuộc vào đặc trưng sinh lý, hành vi tương tác của người dùng… Ví dụ như với thu nhận mẫu vân tay từ máy quét (trường hợp thu nhận mẫu được coi là lý tưởng nhất), sự khác nhau về lực ấn của ngón tay lên thiết bị quét, vị trí ấn ngón tay lên mặt phẳng quét đều ảnh hưởng tới kết quả thu nhận ảnh vân tay. Vì các ngón tay không phải là đối tượng cố định và quá trình chiếu bề mặt đầu ngón tay lên mặt phẳng quét không tuyệt đối chính xác, nên với lực ấn khác nhau, các phần khác nhau của vân tay sẽ được quét như ví dụ ở hình dưới đây:



*Hình 2.2. Thu nhận mẫu sinh trắc không ổn định*

Đối với nhận dạng khuôn mặt, do góc chụp hình khuôn mặt không thể tuyệt đối giống nhau ở

mọi lần lấy mẫu, nên kết quả lấy mẫu phụ thuộc vào vị trí chụp hình khuôn mặt. Vì thế các

mẫu thu được đều có sự khác với nhau.

• Thay đổi của đặc trưng sinh trắc

Ngoài việc khó khăn về quá trình thu nhận, đặc sinh trắc học còn bị ảnh hưởng bởi

ngoại cảnh bên ngoài. Với vân tay, các hoạt động làm việc, tai nạn lao động… đều tác động

tới chất lượng hình ảnh trên đầu ngón tay. Kết quả thu nhận còn thay đổi khi người dùng có

đeo đồ trang sức, ví dụ như nhẫn khi nhận dạng hình dáng bàn tay. Nhận dạng khuôn mặt có

thể gặp khó khăn sau một khoảng thời gian vì độ dài và kiểu tóc, râu người dùng thay đổi,

hoặc bị tai nạn ảnh hưởng tới khuôn mặt… Tất cả các tác động ngoại cảnh đều thay đổi lớn

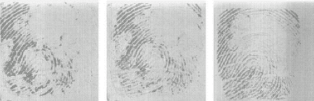
tới kết quả thu nhận mẫu.

• Tác động của môi trường

Các tác động của môi trường tại thời điểm thu nhận cũng ảnh hưởng tới kết quả mẫu

sinh trắc. Ví dụ như độ ẩm, độ sạch của da, ảnh hưởng của tuổi tác, bệnh tật về da… ảnh

hưởng tới mẫu vân tay (Hình 2-3).



*Hình 2.3. Ảnh hưởng của môi trường lên mẫu vân tay*

Ngoài ra, các thuật toán phân tách đặc trưng sinh trắc học từ mẫu thu nhận cũng không hoàn

hảo và có một độ lỗi nhất định. Kết quả là để đối sánh hai mẫu sinh trắc học có giống nhau

hay không là quá trình nhận dạng mẫu và ra quyết định khá phức tạp

b/ Đối sánh sinh trắc học

Do các nguyên nhân ảnh hưởng nêu trên, đối sánh sinh trắc học không thể thực hiện

một cách tuyệt đối như với mật khẩu truyền thống. Thông thường, đối sánh sinh trắc học

thường dùng cách đối sánh tương đối giữa hai mẫu, sự giống nhau của từng thành phần nhỏ

được đánh giá bằng cho điểm (matching score). Khi số điểm đối sánh đủ lớn vượt ngưỡng định trước, có thể coi là hai mẫu sinh trắc gần tương tự nhau.

Ví dụ với nhận dạng vân tay, các thành phần nhỏ được so sánh là điểm kết thúc (ridge

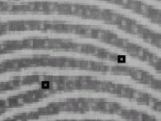
ending) và điểm rẽ nhánh (ridge bifurcation), gọi chung là điểm đặc trưng cục bộ (minutiae).

Các điểm này được tách ra bằng thuật toán trích chọn đặc trưng vân tay. Các điểm đặc trưng

cục bộ được định vị bằng ba tham số (x, y, θ) với (x, y) biểu diễn tọa độ tương đối của điểm

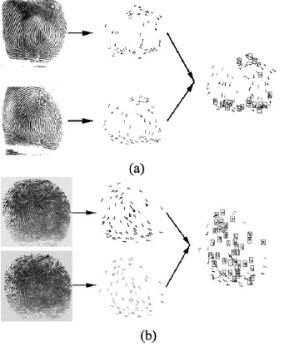
và θ biểu diễn hướng của đỉnh tại điểm đó. Thông thường, một mẫu vân tay tốt có từ 20-70

điểm đặc trưng cục bộ.



*Hình 2.4. Điểm đặc trưng cục bộ của vân tay*

Quá trình đối sánh với một mẫu vân tay khác thực hiện bằng cách so sánh vị trí tương đối giữa các điểm đặc trưng cục bộ với nhau qua thuật toán đối sánh. Kết quả thuật toán trả về là tỷ số điểm đối sánh được chập nhận (matching score):



*Hình 2.5. Đối sánh vân tay*

Kết quả minh họa trong hình 2-5(a), hai vân tay khác nhau cho ra điểm đối sánh là 4, trong

hình 2-5(b), hai vân tay giống nhau cho ra điểm đối sánh là 49. Giá trị tối đa của điểm đối

sánh là 100.

## Đánh giá hiệu năng và chất lượng hoạt động của hệ sinh trắc học

### Vấn đề lỗi trong hoạt động của hệ sinh trắc

Khi hoạt động một hệ sinh trắc học thường gặp hai vấn đề về lỗi sau đây:

- Lỗi khi đối sánh mẫu sinh trắc của hai người khác nhau nhưng cho kết quả là của cùng một

người. Lỗi này được gọi là loại bỏ sai (false reject hay false match).

- Lỗi khi đối sánh hai mẫu sinh trắc của cùng một người nhưng cho kết quả sai, vì cho rằng là

của hai người khác nhau. Lỗi này được gọi là chấp nhận sai (false accept hay false

nonmatch).

### Các tham số đánh giá chất lượng

Để đo lường mức độ lỗi của hệ thông, các độ đo thường dùng được định nghĩa như sau

• FMR (False Match Rate): còn gọi là FAR (False Accept Ratio)- Tỷ số chấp nhận sai :

cho biết tỉ lệ trả lời là đúng đối với dữ liệu vào là sai

• FNMR (False Nonmatch Rate): còn gọi là FRR (False Rejection Ratio) - Tỷ số từ chối

sai: cho biết tỉ lệ trả lời là sai đối với dữ liệu vào là đúng. 25 Hai độ đo này có ràng buộc với nhau: nếu FMR cao thì FNMR sẽ giảm tương đối và ngược lại. Mức độ chấp nhận được của FMR và FNMR tùy thuộc vào từng hệ xác thực sinh trắc cụ thể. Với hệ yêu cầu tính bảo mật cao, và đặt nặng vấn đề an toàn của xác thực hơn sự tiện dụng của người dùng, thì FMR sẽ nhỏ và FNMR sẽ cao. Ngoài hai độ đo trên, người ta còn sử dụng độ đo FTC (Failure To Capture - thu nhận mẫu thất bại) và FTE (Failure to Enroll - chấp nhận mẫu thất bại) để đánh giá hiệu năng của hệ xác thực sinh trắc học.

2.4 **Nghiên cứu các mô hình sinh trắc**

2.4.1 Nhận dạng khuôn mặt

* Xây dựng mô hình:

Mô hình nhận dạng được chia thành 3 bước chính (Hình 2.1), bao gồm:

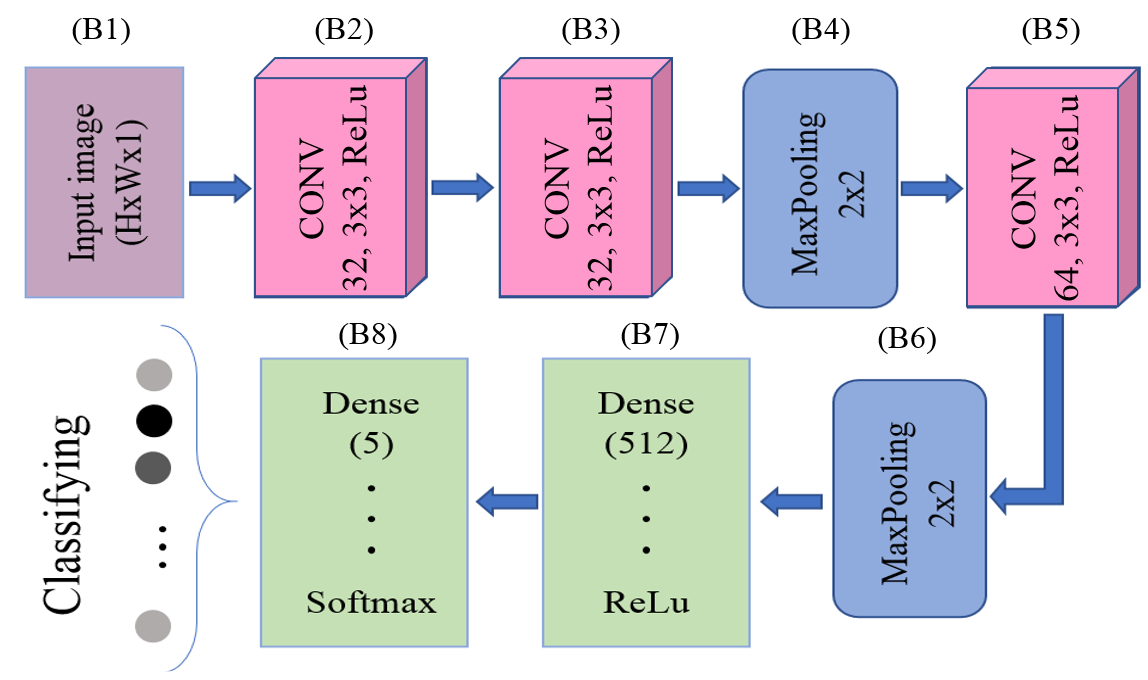
Bước 1: Tạo tập dữ liệu.

Bước 2: Huấn luyện dữ liệu đưa vào và trích chọn các đặc trưng.

Bước 3: Phân loại ảnh khuôn mặt dựa trên đặc trưng được trích chọn và đưa ra kết quả.

* Huấn luyện dữ liệu đưa vào

Mô hình CNN được thiết kế gồm hai phần chức năng là trích chọn đặc trưng của ảnh khuôn mặt và phân lớp đối tượng dựa trên đặc trưng đã chọn. Mô hình CNN bao gồm nhiều lớp, số lớp nơron và độ lớn (số nơron) của mỗi lớp ảnh hưởng đến chất lượng cũng như độ phức tạp trong tính toán của mạng nơron. Các nghiên cứu thường điều chỉnh hai yếu tố này tuỳ theo bài toán ứng dụng để đạt được chất lượng mong muốn và đồng thời đảm bảo sự phức tạp tính toán chấp nhận được.



*Hình 2.2 Kiến trúc dạng khối của mô hình CNN*

Mỗi lớp nơron trong mô hình CNN lấy một mảng nhiều chiều gồm các số làm đầu vào và tạo ra một mảng số nhiều chiều khác ở đầu ra (sau đó trở thành đầu vào của lớp tiếp theo). Khi phân loại hình ảnh khuôn mặt, đầu vào của lớp nơron đầu tiên là kích thước hình ảnh đầu vào. Kích thước đầu ra của lớp cuối cùng là tập hợp các khả năng của các lớp khác nhau được phân loại cho mỗi ảnh đầu vào. Cả ba loại lớp nơron để xây dựng kiến trúc của CNN bao gồm: 3 lớp tích chập (CONV), 2 lớp nơron Maxpooling và 2 lớp nơron kết nối đầy đủ để phân loại (gọi là lớp Dense). Mỗi lớp CONV được kết nối theo sau nó bởi một lớp Maxpooling, áp dụng cơ chế kích hoạt ReLu (Rectified Linear Unit, mặc định là max(x,0)) sau mỗi lớp CONV để đảm bảo đầu vào không âm cho lớp nơron kế tiếp. Theo nguyên tắc xếp chồng các lớp nơron và giảm không gian mẫu (downsampling) tại các kết quả đầu ra của chúng, CNN thực hiện trích xuất các đặc trưng ngày càng trừu tượng và phức tạp hơn, đồng thời, là bất biến đối với các phép biến dạng và chuyển đổi. Mạng tích chập CNN trong mô hình này được chia thành 8 khối (Hình 2.2).

• Khối B1 là ảnh đầu vào có kích thước H×W×1 (cao × rộng × sâu). Để giảm bớt không gian và bộ nhớ của quá trình tính toán mạng nơron nên chúng tôi sử dụng ảnh đầu vào đa cấp xám (số chiều thứ 3 (độ sâu) trong kích thước ảnh bằng.

• Khối B2, B3 là lớp nơron tích chập có 2 bộ lọc đặc trưng với kích thước cửa sổ hàm nhân là 3×3. Hàm kích hoạt ReLu được sử dụng trong lớp nơron này. Hiệu ứng này nhằm cải thiện các đặc trưng thưa của toàn mạng và tránh sự phụ thuộc vào việc truyền tham số giữa các nơron.

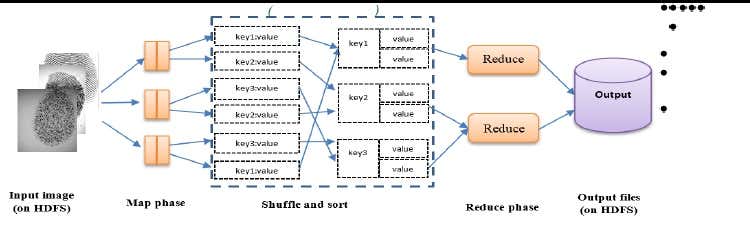
• Khối B4 là lớp Maxpooling, cửa sổ xử lý có kích thước 2×2 được sử dụng. Lớp giảm không gian mẫu (downsampling) này sử dụng phương pháp MaxPooling với việc có thể giữ lại thông tin hữu ích và cắt giảm lượng dữ liệu cần xử lý ở bước tiếp theo.

• Khối B5 tương tự khối B3 nhưng số các bộ lọc (filter) tăng lên 4, sau đó khối này kết nối ngay theo sau khối tích chập CONV để thực hiện cắt giảm không gian mẫu.

2.4.2 Nhận dạng vân tay

* Huấn luyện ảnh vân tay với mô hình MapReduce trong môi trường Spark

Giai đoạn huấn luyện hàng loạt ảnh vân tay được mô tả trong Hình 4.1. Gọi ( Label, Fea\_vector ) tương ứng với nhãn và vector đặc trưng của tập vân tay huấn luyện và ( Label, Fea\_vector\_new ) tương ứng với nhãn và vector đặc trưng đầu ra của tập vân tay huấn luyện được lưu trữ trên HDFS.

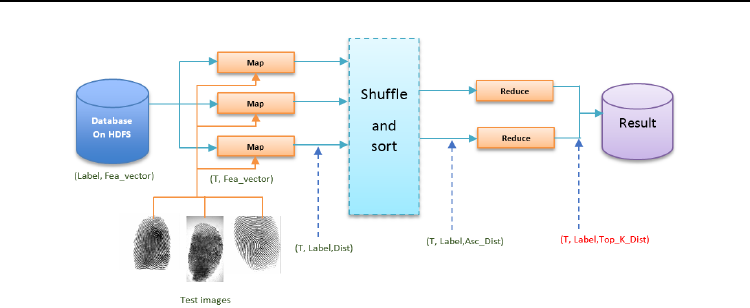


*Hình 3.1 Huấn luyện ảnh vân tay với mô hình MapReduce trong môi trường Spark*

Quá trình thực hiện chi tiết được mô tả như sau:

* Giai đoạn Map:
* Đầu vào hàm Map: tạo danh sách chứa nhân (Label) và giá trị đặc trưng (Fea\_vector).
* Thủ tục Map: lấy tất cả các ảnh từ hệ thống tập tin phân tán HDFS - sử dụng bộ lọc Gabor tăng cường chất lượng ảnh nhị phân hoá - làm móng ảnh trích xuất đặc trưng trộn và sắp xếp → nhân tương ứng với các giá trị đặc trưng.
* Đầu ra hàm Map: vector đặc trưng của tập ảnh vân tay trên HDFS.
* Giai đoạn Reduce:
* Đầu vào hàm Reduce: danh sách chứa nhân và đặc trưng của tập ảnh vân tay.
* Thủ tục Reduce: duyệt qua từng vector đặc trưng sử dụng phương pháp PCA để giảm số chiều các vector đặc trưng cập nhật lại danh sách nhân và giá trị đặc trưng tương ứng.
* Đầu ra hàm Reduce: danh sách đầu ra với nhãn (Label) và giá trị đặc trưng (Fea\_vector\_new).
* Kiểm tra vân tay với mô hình MapReduce trong môi trường Spark

Sau quá trình huấn luyện ảnh vân tay, bước tiếp theo chúng tôi thực hiện kiểm tra ảnh vân tay để đánh giá độ chính xác của hệ thống. Mô hình tổng quát được thể hiện trong Hình 3.2. Gọi (Label, Fea\_vector) tương ứng với nhân và vector đặc trưng của tập vân tay huấn luyện được lưu trữ trên hệ thống tập tin phân tán HDFS và (T. Fea\_vector) tương tứng với nhân và vector đặc trưng của tập vân tay kiểm tra. Quá trình kiểm tra trên hai thuật toán KNN và matching bao gồm các giai đoạn:

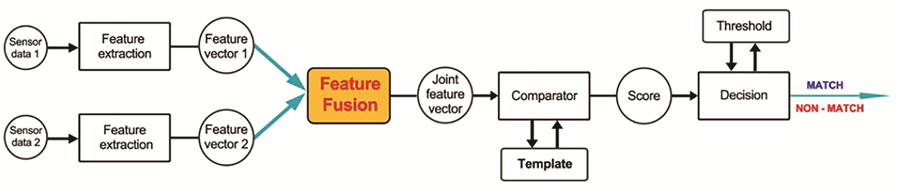


Hình 3.2. Kiểm tra ảnh vân tay trên thuật toán KNN với mô hình MapReduce trong môi trường Spark

## 2.5 Nghiên cứu các mô hình sinh trắc học đa nhân tố

### 2.5.1 Mô hình xác thực sinh trắc học đa nhân tố kết hợp mức đặc trưng

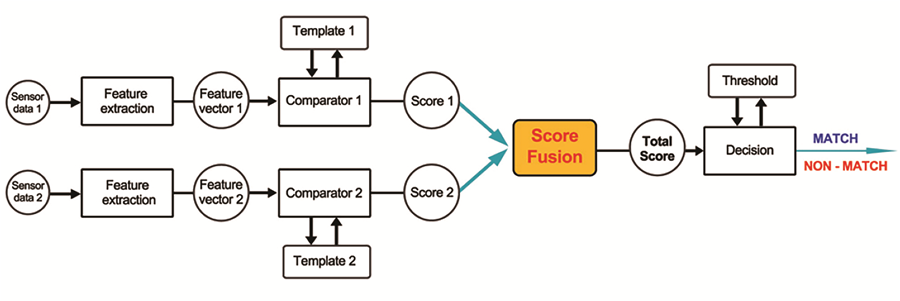
Kết hợp các đặc trưng (Feature Fusion) khác nhau để tạo ra một tập hợp đặc trưng mới. Ví dụ, về kết hợp mức đặc trưng có thể thực hiện với người dùng được thực hiện trích xuất bằng nhiều cảm biến. Khi các vectơ đặc trưng là đồng nhất, chẳng hạn như nhiều hình ảnh vân tay, giá trị trung bình có trọng số của các đặc trưng riêng lẻ có thể được tính toán tạo ra một vectơ đặc trưng (Feature vector) duy nhất (Hình 5).

Hình 5. Mô hình xác thực sinh trắc học đa nhân tố kết hợp mức đặc trưng

### 2.5.2 Mô hình xác thực sinh trắc học đa nhân tố kết hợp mức điểm số

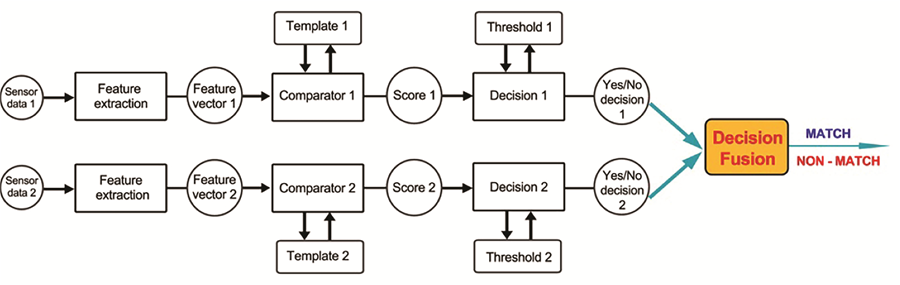
Hệ thống xác thực sinh trắc học đa nhân tố kết hợp mức điểm số (Score fusion) giữa các điểm số riêng biệt của các nhân tố sinh trắc khác nhau đưa ra điểm số tổng (Total score) để so sánh với ngưỡng (Threshold) cho trước trong việc đưa ra quyết định khớp/không khớp (Hình 6). Do đó, tổng hợp điểm so sánh nói chung là cách tiếp cận được ưu tiên để tích hợp dữ liệu. Điểm số tổng được kết hợp bằng nhiều kỹ thuật khác nhau để tạo ra một điểm số mới để so sánh với ngưỡng. Ngày nay, có hai cách tiếp cận chính được sử dụng để tạo ra điểm số so sánh là phân lớp và tổ hợp.

Trong cách tiếp cận phân lớp, mô hình có thể xây dựng một vectơ đặc trưng với các điểm số riêng lẻ và sau đó phân thành lớp với nhãn chấp nhận hoặc loại bỏ. Cách tiếp cận phân lớp có thể sử dụng thuật toán cây quyết định, SVM hoặc LDA để phân lớp vectơ đặc trưng xác định người dùng là giả mạo hoặc thật. Với phương pháp tổ hợp, mô hình tổ hợp các điểm số riêng lẻ để tạo ra một điểm số vô hướng duy nhất để đưa ra quyết định cuối cùng. Phương pháp tổ hợp tạo ra điểm số tổng hợp tạo ra hiệu suất vượt trội so với mức kết hợp khác.

Hình 6. Mô hình xác thực sinh trắc học đa nhân tố kết hợp mức điểm số

### 2.5.3 Mô hình xác thực sinh trắc học đa nhân tố kết hợp mức quyết định

Ở kết hợp mức quyết định (Decision Fusion), mỗi mô hình xác thực sinh trắc học riêng lẻ đưa ra một quyết định xác thực và sau đó những quyết định đó được kết hợp bằng cách sử dụng phương pháp tương tự như bỏ phiếu chọn đa số. Hai quá trình xác thực riêng được kết hợp mức quyết định với đâu ra là quyết định có/không. Kết hợp ở mức độ quyết định thường được sử dụng nhưng được coi là không mềm dẻo và khá đơn giản do thông tin quyết định có sẵn rất hạn chế. Tuy nhiên, đối với một số mô hình triển khai nhất định, việc sử dụng mô hình này có tính khả thi cao (Hình 7).

Hình 7. Mô hình xác thực sinh trắc học đa nhân tố kết hợp mức quyết định

### 2.5.4 Ưu điểm mô hình xác thực sinh trắc học đa nhân tố

Nhìn chung, hệ thống xác thực sinh trắc học đa nhân tố cung cấp hiệu suất nhận dạng vượt trội so với hệ thống xác thực sinh trắc học đơn phương thức. Hệ thống đa phương thức cung cấp một loạt các ưu điểm:

– Giảm số lần chấp nhận sai và từ chối sai, do đó cải thiện đáng kể độ chính xác đối sánh và hiệu suất tổng thể của hệ thống xác thực sinh trắc học, điều này giúp giảm đáng kể tỷ lệ lỗi.

– Ngăn chặn tốt hơn những tấn công hệ thống xác thực sinh trắc học vì rất khó để giả mạo đồng thời nhiều đặc điểm.

– Mở rộng phạm vi khả năng hoạt động của hệ thống xác thực trong điều kiện môi trường khắc nghiệt có thể chấp nhận được (ví dụ: môi trường cần phải giảm tiếng ồn) để xác thực hoặc định danh có thể thực hiện.

– Cung cấp thêm giải pháp bổ trợ cho việc ghi danh, xác minh và nhận dạng, tăng tính khả dụng của hệ thống sinh trắc học, mở rộng đối tượng sử dụng và giảm thiểu tác động của sự khác biệt giữa các đối tượng sử dụng.

Hệ thống xác thực sinh trắc học đa nhân tố còn một số nhược điểm như chi phí cao, phức tạp và thời gian xử lý lâu hơn. Song, những nhược điểm này được bù đắp lại bằng ưu điểm của mô hình mang lại.

# xây dựng chương trình

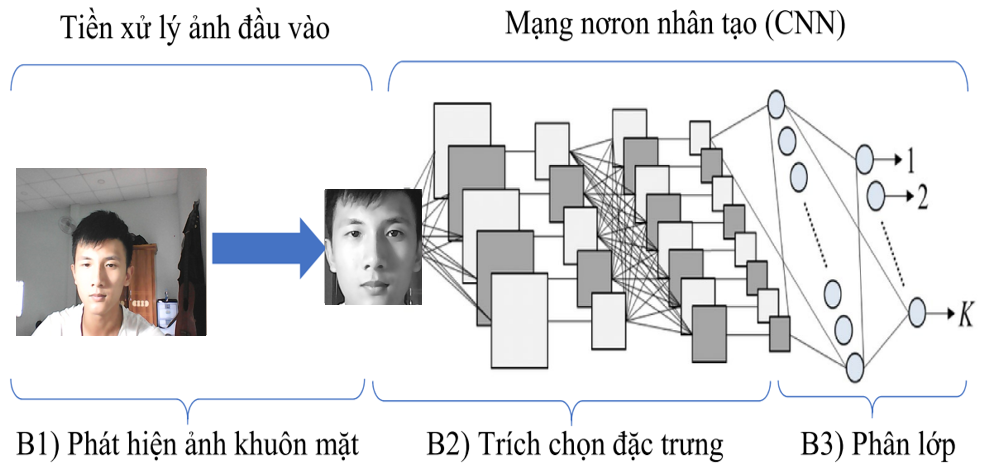
## xây dựng mô hình

Mô hình nhận dạng được chia thành 3 bước chính (Hình 3.1), bao gồm:

Bước 1: Tạo tập dữ liệu.

Bước 2: Huấn luyện dữ liệu đưa vào và trích chọn các đặc trưng.

Bước 3: Phân loại ảnh khuôn mặt dựa trên đặc trưng được trích chọn và đưa ra kết quả.



Hình 3.1 Sơ đồ quy trình của mô hình nhận dạng khuôn mặt

## Các bước thực hiện

### Tạo tập dữ liệu

Các dữ liệu dùng để huấn luyện có thể lấy từ ảnh có sẵn hoặc lấy từ webcam. Các bước phát hiện khuôn mặt và lấy dữ liệu là giống nhau.

Bước 1: Tiền xử lý

- Phương pháp thực hiện trên ảnh xám (gray image). Mỗi điểm ảnh (pixel) sẽ có giá trị mức xám từ 0 đến 255(không gian màu 8 bit). Như vậy phương pháp sẽ không khai thác những đặc điểm về màu sắc khuôn mặt để nhận dạng song vẫn rất hiệu quả. Ảnh màu sẽ được chuyển về ảnh xám để nhận dạng, việc chuyển đổi này khá đơn giản, thực hiện bằng một hàm chuyển đổi và sử dụng chỉ với một câu lệnh trong OpenCV nên báo cáo chưa đề cập tới. Sau khi chuyển thành ảnh xám, ảnh lại tiếp tục được chuyển thành “ảnh tích hợp” và trong bước đầu tiên của quá trình nhận dạng.



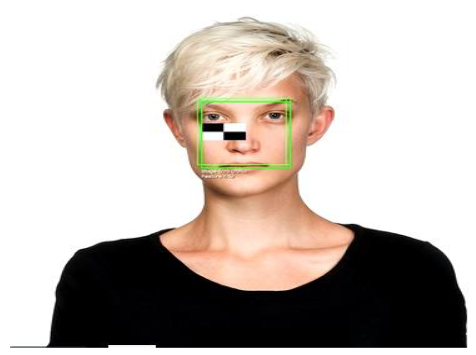
Hình 3.2 Chuyển về ảnh xám

Bước 2: Dò tìm khuôn mặt

Integal Image là mảng 2 chiều với kích thước ảnh cần tính đặc trưng. Bắt đầu từ vị trí trên bên trái đến vị trí dưới, bên phải của ảnh. Sau khi đã tính được Integral Image, việc tính tổng điểm ảnh của một vùng bất kỳ nào đó trên ảnh thực hiện sẽ đơn giản hơn.

Để phát hiện khuôn mặt, hệ thống sẽ cho một cửa sổ con (sub-window) có kích thước cố định quét lên toàn bộ ảnh đầu vào. Như vậy sẽ có rất nhiều ảnh con ứng với từng cửa sổ con, các đặc trưng sẽ được đặt lên các cửa sổ con này để từ đó tính ra giá trị của đặc trưng. Sau đó các giá trị này được bộ huấn luyện xác nhận xem khung hình đó có phải khuôn mặt hay không.

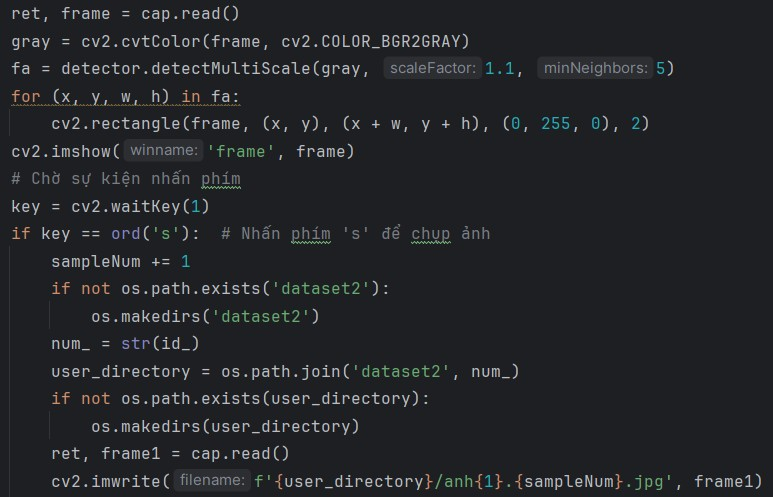
Hình dưới là một ví dụ: khung màu xanh là cửa sổ con, một đặc trưng Haar-like với kích thước và vị trí đặt như trong hình sau:



Hình 3.3 Khung xác định khuôn mặt

Bước 3: Hậu xử lí

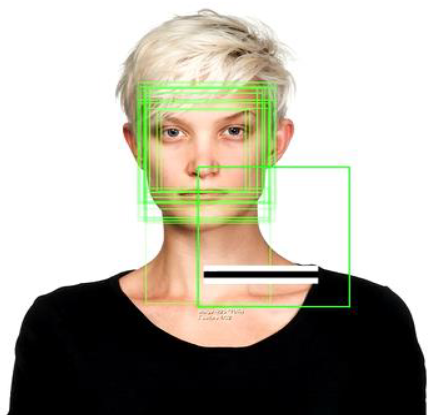
Sau khi quét các cửa sổ con khắp bức ảnh. Đặc trưng sẽ so sánh với bộ huấn luyện mẫu và sẽ đánh dấu những nơi có thể là khuôn mặt trong ảnh.



Hình 3.4 Tạo khung khuôn mặt

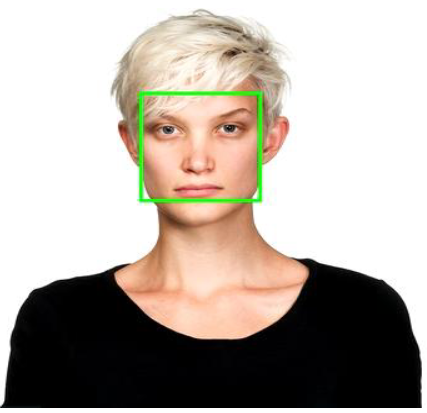
Hàm detectMultiScale là phần tìm kiếm khuôn mặt, hàm (hay phương thức) này thuộc lớp CascadeClassifier (lớp phục vụ tìm kiếm đối tượng của OpenCV)

Hàm detectMultiScale sau khi tìm kiếm xong sẽ trả về bộ giá trị gồm tọa độ gốc của khung chứa khuôn mặt x, y; chiều dài, rộng của khung w, h. Các giá trị này nằm trong mảng faces. Cấu trúc for...sẽ duyệt qua toàn bộ các bộ giá trị này,với mỗi bộ giá trị ta dùng hàm rectangle để vẽ một hình chữ nhật lên ảnh ban đầu với tọa độ 2 điểm trái trên và phải dưới: (x,y),(x+w,y+h). (0,0,255) là màu sẽ vẽ hình chữ nhật.



*Hình 3.5 Các đặc trưng được đánh dấu những nơi có thể là khuôn mặt*

Sau khi đã đánh dấu hết những nơi có thể là khuôn mặt thì sẽ gộp lại những hình chữ nhật chứa cùng một khuôn mặt để chỉ cho ra một hình chữ nhật cho 1 khuôn mặt.

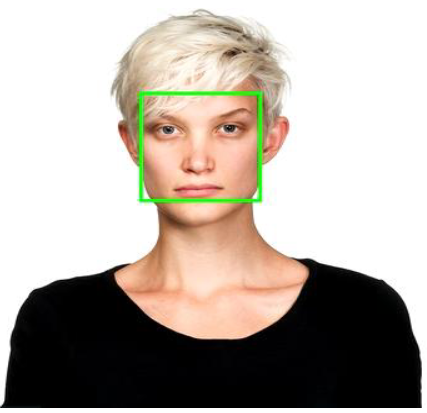


*Hình 3.6 Bức ảnh đã được xác định khuôn mặt*

*- Kết quả của quá trình tạo tập dữ liệu:*



*Hình 3.7 Đưa bức ảnh cần nhận diện vào chương trình*

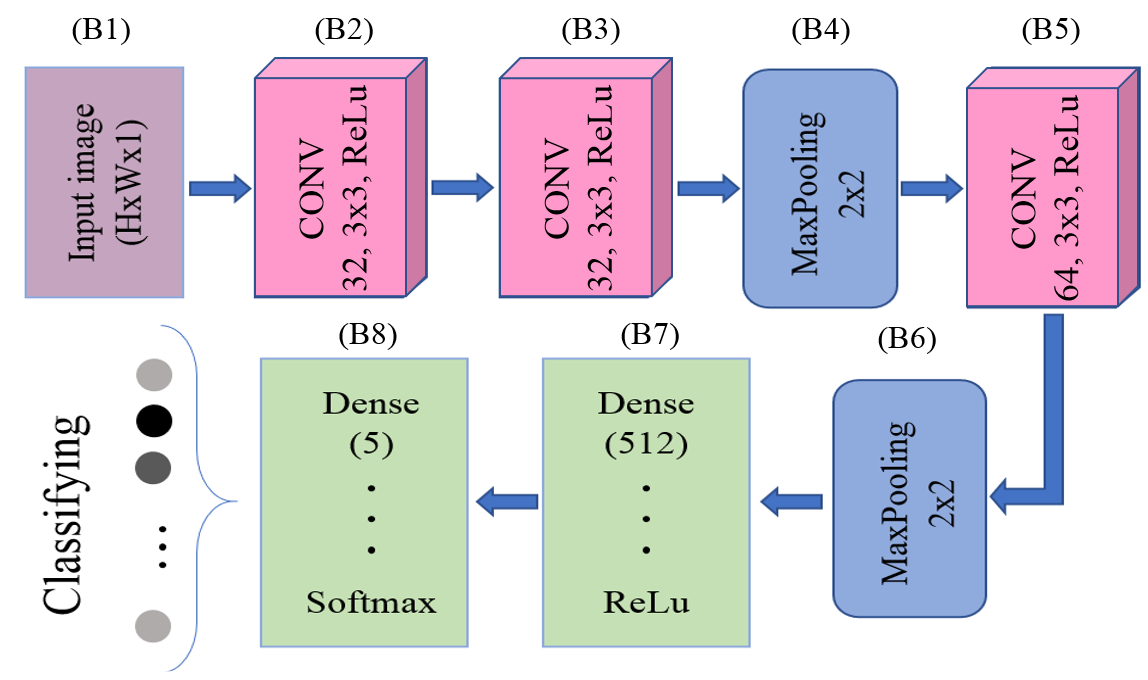


*Hình 3.8 Sau khi phát hiện khuôn mặt*



*Hình 3.9 Dữ liệu được lưu*

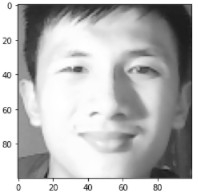
### Huấn luyện dữ liệu đưa vào

Mô hình CNN được thiết kế gồm hai phần chức năng là trích chọn đặc trưng của ảnh khuôn mặt và phân lớp đối tượng dựa trên đặc trưng đã chọn. Mô hình CNN bao gồm nhiều lớp, số lớp nơron và độ lớn (số nơron) của mỗi lớp ảnh hưởng đến chất lượng cũng như độ phức tạp trong tính toán của mạng nơron. Các nghiên cứu thường điều chỉnh hai yếu tố này tuỳ theo bài toán ứng dụng để đạt được chất lượng mong muốn và đồng thời đảm bảo sự phức tạp tính toán chấp nhận được. 

*Hình 3.10 Kiến trúc dạng khối của mô hình CNN*

Mỗi lớp nơron trong mô hình CNN lấy một mảng nhiều chiều gồm các số làm đầu vào và tạo ra một mảng số nhiều chiều khác ở đầu ra (sau đó trở thành đầu vào của lớp tiếp theo). Khi phân loại hình ảnh khuôn mặt, đầu vào của lớp nơron đầu tiên là kích thước hình ảnh đầu vào. Kích thước đầu ra của lớp cuối cùng là tập hợp các khả năng của các lớp khác nhau được phân loại cho mỗi ảnh đầu vào. Cả ba loại lớp nơron để xây dựng kiến trúc của CNN bao gồm: 3 lớp tích chập (CONV), 2 lớp nơron Maxpooling và 2 lớp nơron kết nối đầy đủ để phân loại (gọi là lớp Dense). Mỗi lớp CONV được kết nối theo sau nó bởi một lớp Maxpooling, áp dụng cơ chế kích hoạt ReLu (Rectified Linear Unit, mặc định là max(x,0)) sau mỗi lớp CONV để đảm bảo đầu vào không âm cho lớp nơron kế tiếp. Theo nguyên tắc xếp chồng các lớp nơron và giảm không gian mẫu (downsampling) tại các kết quả đầu ra của chúng, CNN thực hiện trích xuất các đặc trưng ngày càng trừu tượng và phức tạp hơn, đồng thời, là bất biến đối với các phép biến dạng và chuyển đổi. Mạng tích chập CNN trong mô hình này được chia thành 8 khối (Hình 3.10).

• Khối B1 là ảnh đầu vào có kích thước H×W×1 (cao × rộng × sâu). Để giảm bớt không gian và bộ nhớ của quá trình tính toán mạng nơron nên chúng tôi sử dụng ảnh đầu vào đa cấp xám (số chiều thứ 3 (độ sâu) trong kích thước ảnh bằng 1). Hình vẽ sau minh hoạ cho một ảnh đầu vào:



*Hình 3.11 Một ảnh đầu vào kích thước 100x100x1 (đa cấp xám)*

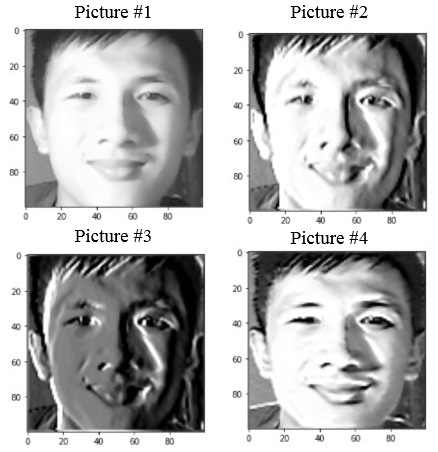
• Khối B2, B3 là lớp nơron tích chập có 2 bộ lọc đặc trưng với kích thước cửa sổ hàm nhân là 3×3. Hàm kích hoạt ReLu được sử dụng trong lớp nơron này. Hiệu ứng này nhằm cải thiện các đặc trưng thưa của toàn mạng và tránh sự phụ thuộc vào việc truyền tham số giữa các nơron.

• Khối B4 là lớp Maxpooling, cửa sổ xử lý có kích thước 2×2 được sử dụng. Lớp giảm không gian mẫu (downsampling) này sử dụng phương pháp MaxPooling với việc có thể giữ lại thông tin hữu ích và cắt giảm lượng dữ liệu cần xử lý ở bước tiếp theo.

Hình 3.11 minh hoạ kết quả xử lý của lớp nơron tích chập CONV ở khối B2 và lớp MaxPooling ở khối B3.

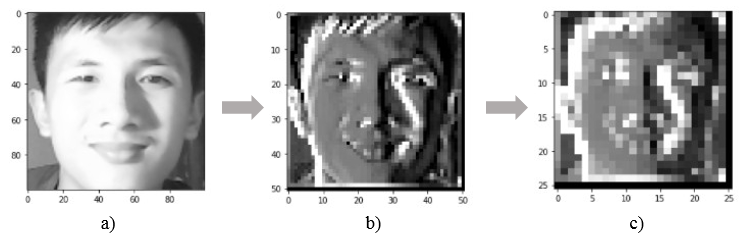
Khối B5 tương tự khối B3 nhưng số các bộ lọc (filter) tăng lên 4, sau đó khối này kết nối ngay theo sau khối tích chập CONV để thực hiện cắt giảm không gian mẫu.

Thông thường, chúng ta càng có nhiều bước sử dụng phép tích chập thì cơ hội trích chọn càng nhiều đặc trưng phức tạp hơn, qua đó kỳ vọng mô hình đề xuất có thể học để nhận biết đối tượng ở mức tốt hơn. Chẳng hạn, trong phân loại hình ảnh, mô hình CNN có thể học để phát hiện các đặc trưng cạnh từ các pixel thô trong lớp CONV đầu tiên, sau đó sử dụng các đặc trưng cạnh này để phát hiện các đặc trưng hình dạng đơn giản trong lớp CONV thứ hai, sau đó sử dụng các đặc trưng hình dạng này để phát hiện các đặc trưng ở mức cao hơn, chẳng hạn như hình dạng khuôn mặt ở các lớp cao hơn.



*Hình 3.12 Hình ảnh sau khi xử lý của khối B2*

Trong Hình 3.13/a/b/c dưới đây minh hoạ kết quả xử lý ở bộ lọc đầu tiên của các khối B3, B5 cùng với lớp nơron MaxPooling ngay sau chúng (tương ứng ở B4, B6), do đó kích thước của hình ảnh sau mỗi bước xử lý giảm dần với hệ số ½ (sau B4 là 50x50, sau B6 là 25x25). Kết quả minh hoạ trực quan cho thấy càng về sau hình ảnh càng “mờ”, thể hiện khả năng trừu tượng hoá và biểu diễn các đặc trưng chung nhất của khuôn mặt một cá nhân, dù được chụp dưới góc độ nào. Hay có thể nói rằng các đặc trưng này của cá nhân có độ bất biến cao nhất đối với bất kỳ hình ảnh khác nhau nào của họ dù dưới các dạng thức, độ sáng tối, màu sắc, kích thước khác nhau.



*Hình 3.13 Hình ảnh kết quả xử lý sau B4 và B6*

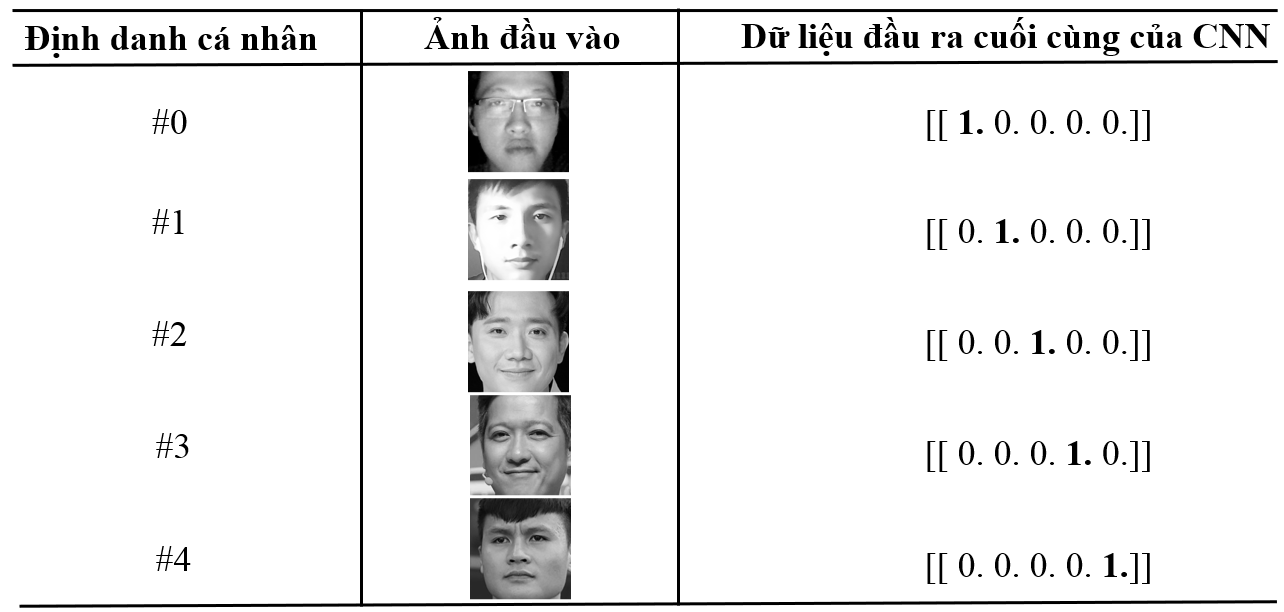
•Khối B7 là lớp nơron kết nối đầy đủ (fully connected layers). Lớp này cùng với lớp trong khối B8 nhằm mục tiêu phân lớp các đặc trưng được trích chọn ở các lớp trước, do đó chúng tôi thiết kế số nơron đủ lớn. Hàm kích hoạt ReLu cũng được áp dụng. Dữ liệu sau đây là đầu ra của 512 nơron đầu tiên trong khối B7, các giá trị ở đây được xem như biểu diễn dạng số của các đặc trưng khuôn mặt tương ứng với ảnh đầu vào.

• Khối B8 là lớp nơron đầu ra cuối cùng, đây chính là một phân bố cho phân loại của các lớp đối tượng khác nhau với tính năng của hàm kích hoạt Softmax. Áp dụng phương pháp hồi quy Softmax ở lớp đầu ra của mạng CNN nhằm thẩm định cho quá trình huấn luyện mạng. Cơ chế thẩm định này đảm bảo mô hình CNN không bị quá khớp (overfiting) dữ liệu học và có khả năng dự đoán tốt hơn. Cơ chế này được thực hiện thông qua việc chia dữ liệu học thành hai phần, một phần để tính toán cập nhật và điều chỉnh trọng số mạng, một phần để tính toán sai số và cũng đưa vào pha cập nhật thay đổi trọng số mạng. Đầu ra phân lớp cuối cùng của mô hình được xác định dựa trên giá trị cực đại hàm Softmax của nơron tương ứng, với mô hình CNN có lớp (tức là có nơron ở lớp ra) thì ta có công thức xác định như sau:

outputclassified = argmaxCk{Ok­­: k = 1, …, |Ok|} (3.1)

Trong đó, Ok là đầu ra của nơron thứ ở lớp nơron cuối cùng và tương ứng với nó là lớp đối tượng Ck.

Bảng 3.1 minh hoạ dữ liệu gồm 5 giá trị sau là kết quả đầu ra của lớp nơron cuối cùng (khối B8), tương ứng với kết quả phân lớp của ảnh đầu vào. Kết quả đầu ra của mô hình CNN tương ứng với mỗi ảnh đầu vào gồm 5 giá trị (ở đây chỉ minh hoạ với 5 lớp). Giá trị cao nhất ở vị trí nào (tính theo chỉ số từ 0 và từ trái sang phải) trong bộ 5 giá trị đầu ra chính là định danh cá nhân (vị trí tương ứng giá trị và số in đậm) của dữ liệu ảnh đầu vào.

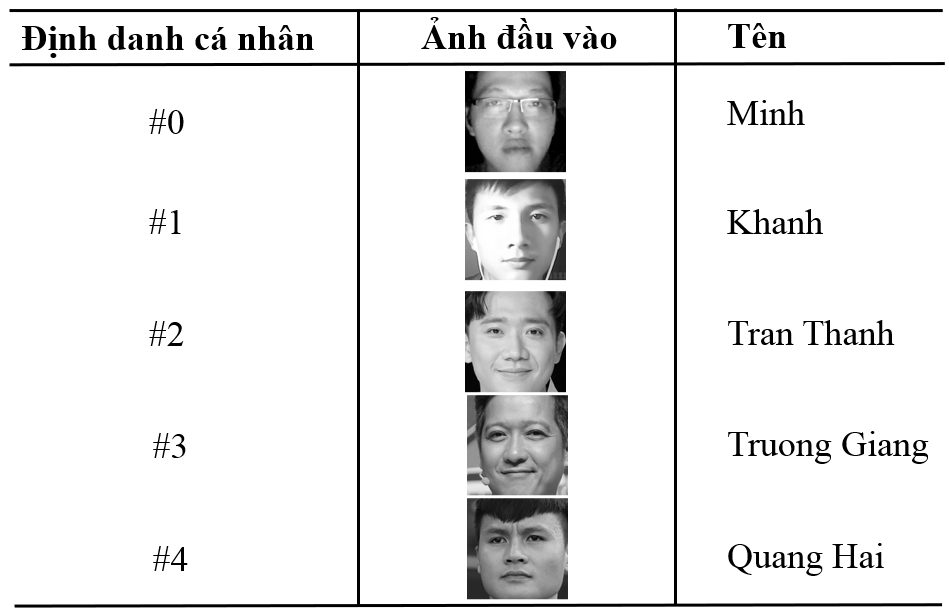


*Hình 3.14 Dữ liệu phân lớp của mô hình CNN*

### Hiển thị dữ liệu đầu ra

Sau khi mô hình được huấn luyên, lưu module vào file “khuonmat.h5”, sau đó dùng file này để nhận diện và dự đoán khuôn mặt của hình ảnh đưa vào.

Từ dữ liệu đầu ra của module, đặt tên cho khuôn mặt được nhận diện riêng biệt. Từ đó sử dụng hàm có sẵn trong OpenCV để hiển thị chữ phía dưới hình vuông nhận diện khuôn mặt. Cách đặt tên tương ứng với hình ảnh như hình 3.15.

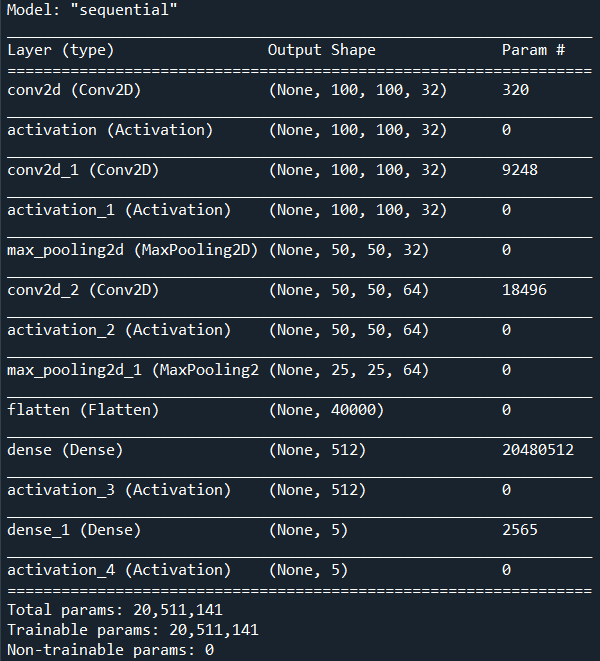


*Hình 3.15 Dữ liệu danh tính*

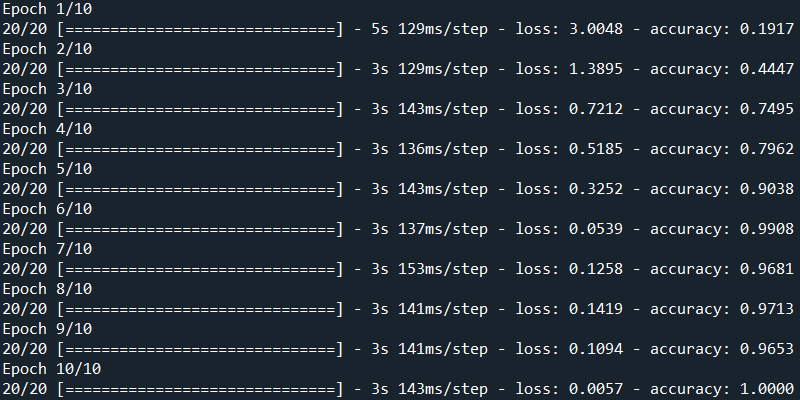
# Kết quả thí nghiệm

## Kết quả sau khi huấn luyện

Quá trình huấn luyện tập dữ liệu:



Hình 4.1 Hơn 20 triệu trọng số được huấn luyện



Hình 4.2 Độ chính và giá trị hàm lỗi theo từng epoch

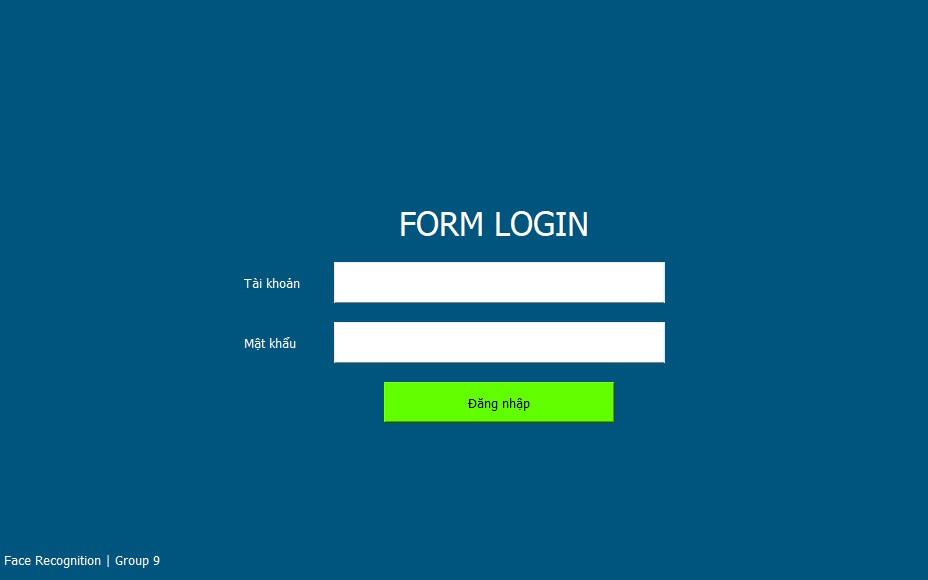
\*Nhận xét:

- Giá trị của hàm lỗi giảm dần theo từng epoch. Giá trị của hàm lỗi ở epoch 1 là 3.0048, giá trị hàm lỗi ở epoch 10 giảm xuống còn 0.0057.

- Độ chính xác tăng dần qua từng epoch, tăng lên tối đa 100% ở epoch thứ 10.

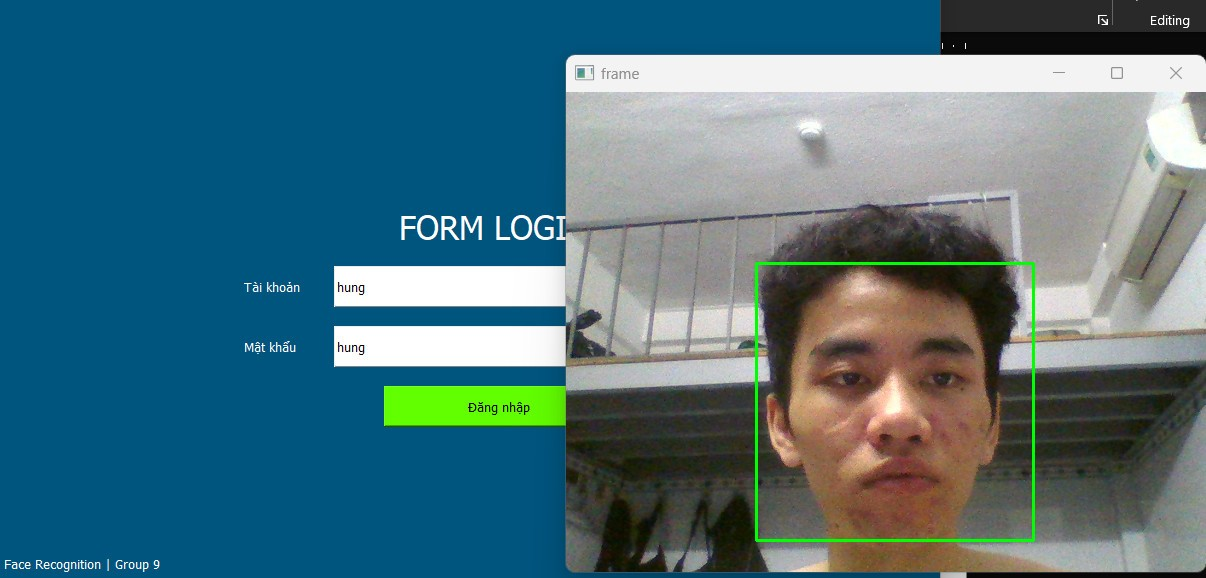
## Sử dụng phần mềm

Giao diện chính của phần mềm



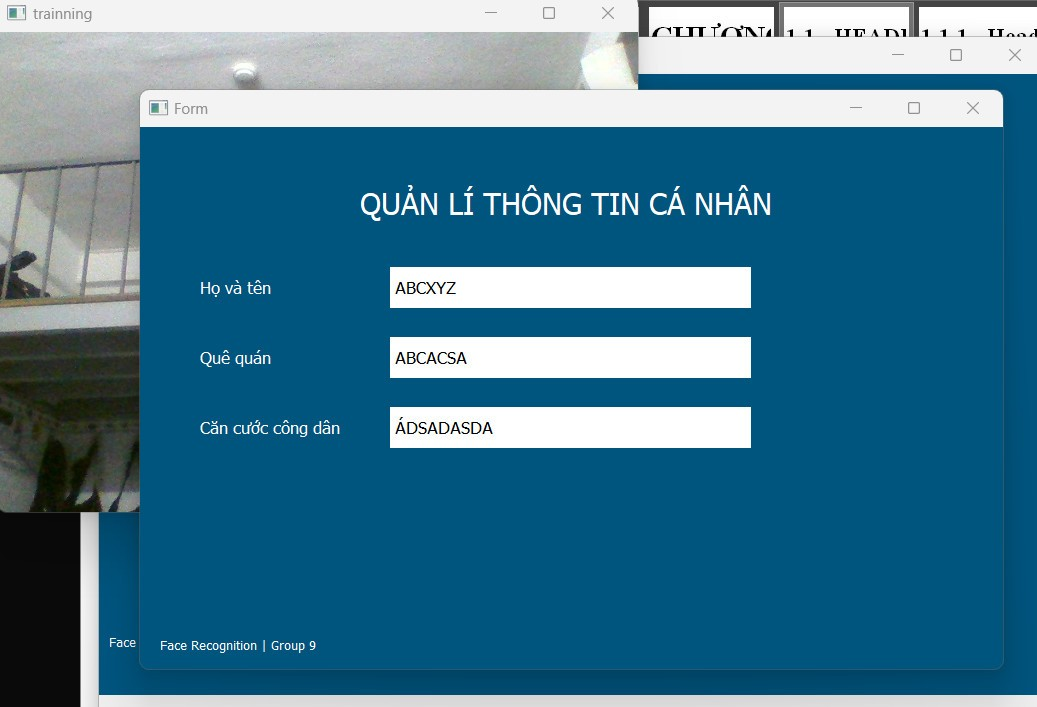
*Hình 4.3 Hình ảnh giao diện chính của phần mềm*

Hinh ảnh khi đăng nhập thành công sẽ hiện giao diện để chụp xác thực khuôn mặt



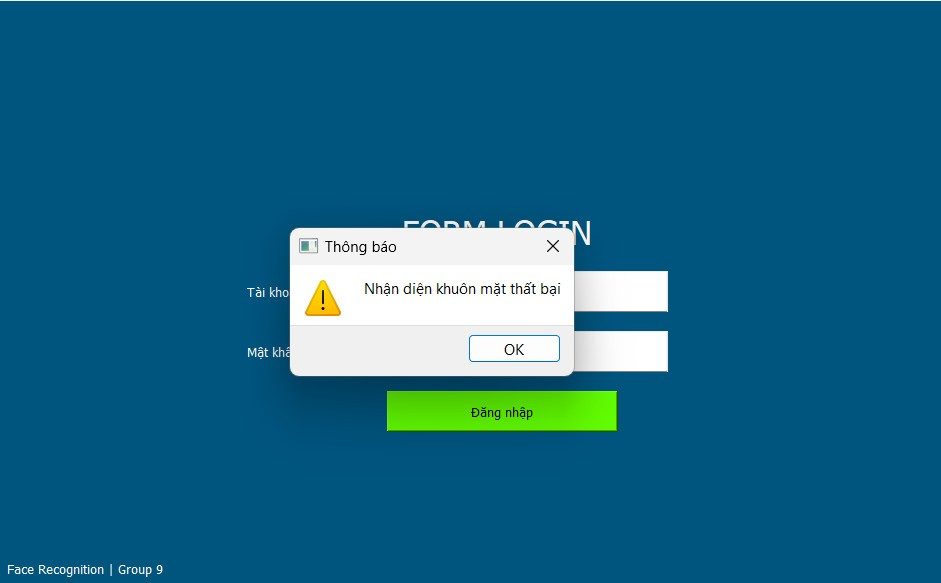
*Hình 4.4 Hình ảnh sau khi đăng nhập thành công*

Sau khi xác thực khuôn mặt thành công sẽ hiện ra 1 bảng thông tin cá nhân



*Hình 4.5 Hình ảnh sau khi xác thực thành công*

Còn nếu như xác thực thất bại sẽ hiện ra thông báo



*Hình 4.6 Hình ảnh xác thực khuôn mặt thất bại*

\* Nhận xét:

- Phần mềm nhận diện còn chưa được chính xác lắm do tập dữ liệu còn ít trong tương lai có thể mở rộng thêm.

- Có khả năng sử dụng trong thưc tiễn nhận diện ở cơ quan hay điểm danh sinh viên khi học online.

# KẾT LUẬN

Trong báo cáo này, chúng tôi đã đề xuất một mô hình dựa trên mạng nơron tích chập (CNN) để nhận dạng khuôn mặt con người. Mô hình này có 3 lớp nơron tích chập (Convolution) và 2 lớp nơron liên kết đầy đủ (Fully Connected), tổng số tham số là khoảng hơn 20 triệu.

Như vậy, có thể khẳng định mô hình của chúng tôi có độ phức tạp ở mức vừa phải, phù hợp với các hệ thống xử lý ở mức trung bình và đem lại tiềm năng khả thi trong ứng dụng thực tiễn.

Mặc dù độ phức tạp của mô hình ở mức thấp so với các mô hình khác, nhưng kết quả thử nghiệm cho thấy tính hiệu quả của phân lớp khá cao. Hiện nay do điều kiện tính toán nên chỉ áp dụng số lần huấn luyện còn thấp, nếu được huấn luyện ở mức độ sâu hơn thì kỳ vọng sẽ đem lại kết quả cao hơn nữa.

Để phát triển thêm cho mô hình, chúng tôi sẽ tìm hiểu và thiết kế một hệ thống thu thập dữ liệu hình ảnh để tạo bộ dữ liệu huấn luyện đa dạng cho mô hình, từ đó xây dựng một ứng dụng cho bài toán thực tiễn như hệ thống điểm danh sinh viên có mặt ở lớp học, hệ thống giám sát cán bộ vào/ra cổng cơ quan, hệ thống theo dõi và định danh liên tục quá trình học tập của người học trực tuyến.

# PHỤ LỤC

Chương trình được thực hiện trên môi trường Pycharm (Anaconda) với 3 chương trình như sau:

**+ Tạo tập dữ liệu:**

def create\_data(self, user\_, pass\_, id\_):

import os

import cv2

import pyodbc

id = 2

if id == 2:

cap = cv2.VideoCapture(0)

detector = cv2.CascadeClassifier(cv2.data.haarcascades + 'haarcascade\_frontalface\_default.xml')

sampleNum = 0

while True:

ret, frame = cap.read()

gray = cv2.cvtColor(frame, cv2.COLOR\_BGR2GRAY)

fa = detector.detectMultiScale(gray, 1.1, 5)

for (x, y, w, h) in fa:

cv2.rectangle(frame, (x, y), (x + w, y + h), (0, 255, 0), 2)

cv2.imshow('frame', frame)

# Chờ sự kiện nhấn phím

key = cv2.waitKey(1)

if key == ord('s'): # Nhấn phím 's' để chụp ảnh

sampleNum += 1

if not os.path.exists('dataset2'):

os.makedirs('dataset2')

num\_ = str(id\_)

user\_directory = os.path.join('dataset2', num\_)

if not os.path.exists(user\_directory):

os.makedirs(user\_directory)

ret, frame1 = cap.read()

cv2.imwrite(f'{user\_directory}/anh{1}.{sampleNum}.jpg', frame1)

print("Chụp ảnh số:", sampleNum)

if sampleNum > 19:

# Thiết lập thông tin kết nối

server = 'localhost'

database = 'cuoiki'

username = 'sa'

password = 'Tranlam1@!'

try:

# Tạo chuỗi kết nối

conn\_str = f'DRIVER={{SQL Server}};SERVER={server};DATABASE={database};UID={username};PWD={password}'

# Kết nối đến cơ sở dữ liệu

conn = pyodbc.connect(conn\_str)

# Tạo một đối tượng cursor để thực thi các truy vấn SQL

cursor = conn.cursor()

# Truy vấn SQL để kiểm tra thông tin đăng nhập

query = """

SELECT \*

FROM login

WHERE username = ? AND password = ?

"""

cursor.execute(query, (user\_, pass\_))

# Lấy dòng kết quả

row = cursor.fetchone()

if row:

# Cập nhật giá trị của dòng trong cơ sở dữ liệu

update\_query = """

UPDATE login

SET data = ?

WHERE username = ? AND password = ?

"""

cursor.execute(update\_query, (20, user\_, pass\_))

# Lưu thay đổi vào cơ sở dữ liệu

conn.commit()

print(id\_)

self.train(id\_)

cap.release()

cv2.destroyAllWindows()

return

except pyodbc.Error as ex:

# In ra thông báo lỗi nếu kết nối không thành công

print("Lỗi kết nối:", ex)

break

# Thoát khỏi vòng lặp nếu nhấn phím 'q'

elif key == ord('q'):

break

cap.release()

cv2.destroyAllWindows()

* Tạo dữ liệu bằng cách chụp 20 tấm hình bằng webcam

**+ Huấn luyện tập dữ liệu:**

#Huấn luyện

def train(self, id\_):

import cv2

import numpy as np

from sklearn.preprocessing import LabelBinarizer

from tensorflow.keras.models import Sequential

from tensorflow.keras.layers import Conv2D, MaxPooling2D, Flatten, Dense, Activation

from tensorflow.keras.optimizers import Adam

from sklearn.model\_selection import train\_test\_split

# ----convert-------

id\_int = int(id\_)

detector = cv2.CascadeClassifier(cv2.data.haarcascades + 'haarcascade\_frontalface\_default.xml')

if not os.path.exists('dataset\_end'):

os.makedirs('dataset\_end')

for j in range(1, id\_int + 1):

value\_ = str(j)

user\_directory = os.path.join('dataset\_end', value\_)

if not os.path.exists(user\_directory):

os.makedirs(user\_directory)

for i in range(1, 21):

filename = f'./dataset2/{j}/anh1.{i}.jpg'

frame = cv2.imread(filename)

gray = cv2.cvtColor(frame, cv2.COLOR\_BGR2GRAY)

fa = detector.detectMultiScale(gray, 1.1, 5)

for (x, y, w, h) in fa:

cv2.rectangle(frame, (x, y), (x + w, y + h), (0, 255, 0), 2)

cv2.imwrite(f'{user\_directory}/anh1.{i}.jpg', gray[y:y + h, x:x + w])

# #-----Training

data = []

label = []

for j in range(1, id\_int + 1):

for i in range(1, 21):

filename = f'./dataset\_end/{j}/anh1.{i}.jpg'

print(filename)

Img = cv2.imread(filename)

Img = cv2.cvtColor(Img, cv2.COLOR\_BGR2GRAY)

Img = cv2.resize(src=Img, dsize=(100, 100))

# Img = Img.reshape((100, 100, 1))

Img = np.array(Img)

data.append(Img)

label.append(j - 1)

print(filename)

data1 = np.array(data)

label = np.array(label) # nhãn tương ứng với mỗi ảnh

data1 = data1.reshape((id\_int \* 20, 100, 100, 1))

X\_train = data1 / 255

from sklearn.preprocessing import LabelBinarizer

lb = LabelBinarizer()

trainY = lb.fit\_transform(label)

# Sử dụng kiến trúc mạng CNN Sequential API của Keras.

# Bao gồm các lớp Convolutional, MaxPooling, Flatten và Dense.

Model = Sequential()

shape = (100, 100, 1)

Model.add(Conv2D(32, (3, 3), padding="same", input\_shape=shape))

Model.add(Activation("relu"))

Model.add(Conv2D(32, (3, 3), padding="same"))

Model.add(Activation("relu"))

Model.add(MaxPooling2D(pool\_size=(2, 2)))

Model.add(Conv2D(64, (3, 3), padding="same"))

Model.add(Activation("relu"))

Model.add(MaxPooling2D(pool\_size=(2, 2)))

Model.add(Flatten())

Model.add(Dense(512))

Model.add(Activation("relu"))

Model.add(Dense(id\_int))

Model.add(Activation("softmax"))

Model.summary()

Model.compile(loss='categorical\_crossentropy',

optimizer='adam',

metrics=['accuracy'])

print("start training")

Model.fit(X\_train, trainY, batch\_size=5, epochs=10)

Model.save("face-result.h5")

**+ Nhận diện khuôn mặt:**

//Kiểm tra

#kiem tra face sau khi login

def check(self, id\_, name\_):

main\_window2 = MainWindow2() # Tạo ra MainWindow2 ở đây

main\_window2.hide()

isTrue = False

cap = cv2.VideoCapture(cv2.CAP\_DSHOW)

detector = cv2.CascadeClassifier(cv2.data.haarcascades + 'haarcascade\_frontalface\_default.xml')

while True:

ret, frame = cap.read()

if not ret:

print("Failed to capture frame")

break

fa = detector.detectMultiScale(frame, 1.1, 5)

# Draw rectangles around faces only when displaying the frame

for (x, y, w, h) in fa:

cv2.rectangle(frame, (x, y), (x + w, y + h), (0, 255, 0), 2)

# Display the frame

cv2.imshow('frame', frame)

# Wait for key event

key = cv2.waitKey(1)

# Press 's' to capture the image

if key == ord('s'):

ret, frame1 = cap.read()

cv2.imwrite(f'test/test.jpg', frame1)

break

# Press 'q' to quit the application

if key == ord('q'):

break

cap.release()

cv2.destroyAllWindows()

##-------------

filename = 'test/test.jpg'

image = cv2.imread(filename)

face\_cascade = cv2.CascadeClassifier(cv2.data.haarcascades + 'haarcascade\_frontalface\_default.xml')

save\_model = tf.keras.models.load\_model("face-result.h5")

gray = cv2.cvtColor(image, cv2.COLOR\_BGR2GRAY)

fa = face\_cascade.detectMultiScale(gray, 1.1, 5)

fontface = cv2.FONT\_HERSHEY\_SIMPLEX

for (x, y, w, h) in fa:

cv2.rectangle(image, (x, y), (x + w, y + h), (0, 255, 0), 2)

roi\_gray = gray[y:y + h, x:x + w]

roi\_gray = cv2.resize(src=roi\_gray, dsize=(100, 100))

roi\_gray = roi\_gray.reshape((100, 100, 1))

roi\_gray = np.array(roi\_gray)

result = save\_model.predict(np.array([roi\_gray]))

final = np.argmax(result)

if(final + 1 == id\_):

msg = QMessageBox()

msg.setIcon(QMessageBox.Warning)

msg.setWindowTitle("Thông báo")

msg.setText("Nhận diện khuôn mặt thành công")

msg.exec\_()

QtCore.QTimer.singleShot(100, lambda: main\_window2.show())

cv2.imshow('trainning', image)

print(result)

cv2.waitKey(0)

else:

msg = QMessageBox()

msg.setIcon(QMessageBox.Warning)

msg.setWindowTitle("Thông báo")

msg.setText("Nhận diện khuôn mặt thất bại")

msg.exec\_()

cv2.imshow('trainning', image)

print(result)

TÀI LIỆU THAM KHẢO

[1] PGS. TS Trương Ngọc Sơn, *Trí tuệ nhân tạo cơ sở và ứng dụng,* Đại học quốc gia TPHCM, 2020.

[2] Trương Đức Nhật Minh, Phạm Đình Khanh, Tham khảo source -2021 https://drive.google.com/drive/folders/1GfejuNOH7LUUJTCEpgMBJnP2ciS20h3C

[3] Q. Zhang, M. Zhang, T. Chen, Z. Sun, Y. Ma, and B. Yu, “Recent advances in convolutional neural network acceleration,” *Neurocomputing*, vol. 323, pp. 0–38, 2019, doi: 10.1016/j.neucom.2018.09.038.