

ĐẠI HỌC HUẾ  
TRƯỜNG ĐẠI HỌC KHOA HỌC  
KHOA CÔNG NGHỆ THÔNG TIN  
\*\*\*



**TIỂU LUẬN  
KHAI PHÁ DỮ LIỆU**

**XÂY DỰNG ỨNG DỤNG ĐIỂM DANH DỰA  
TRÊN MÔ HÌNH MTCNN VÀ FACENET**

Nhóm sinh viên thực hiện:

- 1. NGUYỄN LUÔN MONG ĐỔ (NT)**
- 2. VÕ ĐẠT VĂN**
- 3. NGUYỄN TIỀN NHẬT**
- 4. NGUYỄN HOÀI NAM**
- 5. LÝ NHẬT PHƯƠNG**
- 6. NGUYỄN NGỌC QUANG HUY**

Tên học phần: **KHAI PHÁ DỮ LIỆU - NHÓM 1**

Giáo viên hướng dẫn: **T.S NGUYỄN NGỌC THUỶ**

Huế, 12 - 2023

# Mục lục

<b>LỜI MỞ ĐẦU</b>	<b>1</b>
<b>1 GIỚI THIỆU</b>	<b>2</b>
1.1 Mô tả bài toán . . . . .	2
1.2 Đầu vào của bài toán . . . . .	3
1.3 Đầu ra của bài toán . . . . .	3
1.4 Các thách thức . . . . .	4
1.4.1 Thách thức chung . . . . .	4
1.4.2 Thách thức của nhận diện khuôn mặt MTCNN . . . . .	5
1.4.3 Thách thức của nhận dạng khuôn mặt FaceNet . . . . .	7
1.5 Mục tiêu của tiểu luận . . . . .	7
1.6 Cấu trúc của tiểu luận . . . . .	8
<b>2 CÁC CÁCH TIẾP CẬN</b>	<b>9</b>
2.1 Sử dụng đặc trưng SIFT và BoVW . . . . .	9
2.2 Sử dụng thuật toán DeepFace . . . . .	11
<b>3 PHƯƠNG PHÁP THỰC HIỆN</b>	<b>13</b>
3.1 Giới thiệu Multi-task Cascaded Convolutional Networks (MTCNN) .	13
3.1.1 Proposal Network (P-Net) . . . . .	15
3.1.2 Refine Network (R-Net) . . . . .	17
3.1.3 Output Network (O-Net) . . . . .	18
3.2 Giới thiệu FaceNet . . . . .	19
3.2.1 FaceNet . . . . .	19
3.2.2 Triplet Loss . . . . .	20

<b>4 THỦ NGHIỆM VÀ ĐÁNH GIÁ</b>	<b>21</b>
4.1 Tổng quan các bước thử nghiệm . . . . .	21
4.2 Dữ liệu đánh giá . . . . .	21
4.2.1 Cấu trúc dữ liệu đầu vào . . . . .	21
4.2.2 Thông tin dữ liệu . . . . .	22
4.2.3 Quy trình tạo dữ liệu . . . . .	24
4.3 Kết quả thí nghiệm và thảo luận . . . . .	24
4.3.1 Cách đánh giá . . . . .	24
4.3.2 Kết quả . . . . .	24
4.3.3 Thảo luận . . . . .	26
<b>5 XÂY DỰNG ỨNG DỤNG</b>	<b>28</b>
5.1 Giao diện ứng dụng . . . . .	28
5.1.1 Trang chủ . . . . .	28
5.1.2 Thông kê . . . . .	28
5.1.3 Giao diện điểm danh . . . . .	30
<b>6 KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN</b>	<b>31</b>
6.1 Kết luận . . . . .	31
6.2 Hướng phát triển . . . . .	32

## Danh sách hình vẽ

1.1	Minh họa đầu vào của bài toán. Ảnh sinh viên Trần Công Sơn. . . . .	3
1.2	Minh họa đầu ra của bài toán. Ảnh sinh viên Trần Công Sơn với Bounding Box trên khuôn mặt và nhãn CongSon. . . . .	4
1.3	Minh họa việc giả mạo.(Nguồn: Mì AI) . . . . .	5
1.4	Minh họa thay đổi sinh học (sau khi giảm cân).(Nguồn: random.com.vn)	5
1.5	Minh họa việc ảnh đối tượng có góc độ xấu làm mất đi đặc trưng khuôn mặt của đối tượng (không nhận dạng được). . . . .	6
2.1	Minh họa thuật toán BoVW. Với hình ảnh cô giá bên trái, ta có các đặc trưng là mắt, mũi, miệng, ... sau đó cho vào một cái "túi"đặc trưng như hình bên phải. Nguồn [14]. . . . .	10
2.2	Minh họa sơ đồ rút trích đặc trưng. Nguồn [10]. . . . .	10
2.3	Ảnh cấu trúc huấn luyện của DeepFace, từ ảnh vào, lấy khuôn mặt, sau đó chỉnh chính diện vào mô hình 3D, tiếp theo là các lớp tích chập C1- lớp Pooling (M2) - C3, sau đó là 3 lớp Local Connected (L4-L6), cuối cùng là 2 lớp Fully Connected (F7-F8). Nguồn [1]. . . . .	12
3.1	Hình ảnh các mạng tích chập sâu đa tác vụ ba giai đoạn. Thứ nhất, cửa sổ ứng viên được tạo thông qua Mạng đề xuất nhanh (P-Net). Sau đó, mạng tinh chỉnh những ứng cử viên này trong giai đoạn tiếp theo thông qua Mạng sàng lọc (RNet). Trong giai đoạn thứ ba, Mạng đầu ra (O-Net) tạo ra hộp giới hạn cuối cùng. Nguồn: [5] . . . . .	14
3.2	Image pyramid. Nguồn: Zhang et al [16] . . . . .	15
3.3	P-Net ("Conv"là lớp tích chập; "MP"là lớp gộp (Pooling Layer). Nguồn: Zhang et al [16] . . . . .	15

3.4	Hình minh họa cách tính đầu ra sau khi qua lớp Tích chập. Nguồn: Stanford . . . . .	16
3.5	Kết quả sử dụng Non-Maximum Suppression cho P-Net. Nguồn: Zhang et al [16] . . . . .	17
3.6	R-Net. Nguồn: Zhang et al [16] . . . . .	18
3.7	Kết quả sử dụng Non-Maximum Suppression cho R-Net . . . . .	18
3.8	O-Net. Nguồn: Zhang et al [16] . . . . .	18
3.9	Kết quả sử dụng R-Net cùng với Non-Maximum Suppression. Nguồn: Zhang et al [16] . . . . .	19
3.10	Kiến trúc của FaceNet. Nguồn: Schoff et at 2015 [7] . . . . .	19
3.11	Bộ ba sai số tối thiểu hoá khoảng cách giữa ảnh vào (Anchor) và ảnh cùng loại với ảnh vào (Positive) và tối đa hoá khoảng cách giữa ảnh vào và ảnh khác loại với ảnh vào (Negative). Nguồn: Schoff et at 2015 [7] . . . . .	20
4.1	Mô tả cấu trúc dữ liệu đầu vào. . . . .	22
4.2	Một số hình ảnh của các tập dữ liệu thử nghiệm. . . . .	23
4.3	Ma trận nhầm lẫn (thực nghiệm lần 1). . . . .	25
4.4	Ma trận nhầm lẫn (thực nghiệm lần 2). . . . .	26
4.5	Một số hình ảnh nhận diện sai của đối tượng Nguyễn Ngọc Quang Huy.	27
4.6	Một số hình ảnh nhận diện sai của đối tượng Nguyễn Văn Tiến. . . . .	27
5.1	Giao diện trang chủ . . . . .	28
5.2	Thông kê chung về tình trạng lớp học . . . . .	29
5.3	Danh sách sinh viên của lớp học phần . . . . .	29
5.4	Danh sách sinh viên tham gia buổi học . . . . .	30
5.5	Danh sách sinh viên vắng mặt trong buổi học . . . . .	30
5.6	Giao diện điểm danh . . . . .	30

# LỜI MỞ ĐẦU

Học sâu là một lĩnh vực học máy mới nổi đã phát triển nhanh chóng và áp dụng cho nhiều lĩnh vực với tần suất thành công cao bao gồm xử lý hình ảnh, nhận dạng giọng nói và xử lý văn bản. Đặc biệt, hiệu suất nhận dạng khuôn mặt đã cải thiện nhanh chóng nhờ kỹ thuật học sâu gần đây đang phát triển và tích lũy tập dữ liệu đào tạo lớn. Tuy nhiên, hình ảnh khuôn mặt ngoài tự nhiên tồn tại nhiều biến thể lớn ở mỗi cá nhân, chẳng hạn như tư thế, độ sáng, độ che khuất và độ phân giải thấp, gây ra những thách thức lớn cho các ứng dụng liên quan đến khuôn mặt.

Nhận thấy vì lí do trên, song song với lí do thiếu thiết bị chuyên dụng, trong nhà trường Đại học Khoa Học, Đại học Huế nói riêng, và nhiều trường Đại học trên toàn quốc nói chung, hệ thống điểm danh sinh viên bằng phương pháp nhận dạng khuôn mặt vẫn đang còn gặp nhiều hạn chế, chưa thể đưa vào ứng dụng.

Với mong muốn giải quyết được những thách thức trên, trong bài tiểu luận của mình, chúng tôi xây dựng hệ thống nhận dạng khuôn mặt dựa trên hai biến thể của mô hình CNN là Multi-task Cascaded Convolutional Networks (MTCNN) và FaceNet, sau đó, ứng dụng hệ thống để xây dựng một ứng dụng điểm danh sinh viên. Với ứng dụng điểm danh sinh viên, nhà trường có thể đánh giá sự tương tác của người học với bộ môn được giảng dạy một cách tốt hơn.

Về quy trình, trước tiên chúng tôi sử dụng MTCNN [16] để nhận diện khuôn mặt. Sau đó sử dụng kết quả của MTCNN làm đầu vào của FaceNet để thực hiện nhận dạng khuôn mặt [6]. Mạng MTCNN, là mạng phát hiện mục tiêu chính thống với độ chính xác phát hiện cao, nhẹ và thời gian thực. Phương pháp FaceNet, nó trực tiếp học cách ánh xạ từ hình ảnh khuôn mặt sang không gian Euclide nhỏ gọn, trong đó khoảng cách tương ứng trực tiếp với độ giống nhau của khuôn mặt. Cách tiếp cận này làm giảm đáng kể sự khác biệt giữa các cá nhân, đồng thời duy trì tính phân biệt giữa các cá nhân.

# Chương 1

## GIỚI THIỆU

### 1.1 Mô tả bài toán

Với sự phát triển nhanh chóng của trí tuệ nhân tạo trong những năm gần đây, nhận dạng khuôn mặt ngày càng được chú ý nhiều hơn. So với nhận dạng thẻ truyền thống, nhận dạng vân tay và nhận dạng mống mắt, nhận dạng khuôn mặt có nhiều ưu điểm, bao gồm nhưng hạn chế ở mức không tiếp xúc, tính đồng thời cao và thân thiện với người dùng. Nó có tiềm năng cao để được sử dụng trong chính phủ, cơ sở công cộng, an ninh, thương mại điện tử, bán lẻ, giáo dục và nhiều lĩnh vực khác.

Việc nhận dạng khuôn mặt là một bài toán phức tạp đòi hỏi xử lý một lượng lớn thông tin và đặc trưng về các đối tượng có trong hình ảnh hoặc video. Để giải quyết bài toán này, có sự đa dạng về các phương pháp và thuật toán được áp dụng, từ các phương pháp truyền thống cho đến các phương pháp sử dụng trí tuệ nhân tạo như học máy và học sâu. Các phương pháp này đòi hỏi các kỹ thuật sâu rộng, kinh nghiệm phong phú trong việc tiếp cận và xử lý dữ liệu, đặc biệt là khi đối mặt với các tình huống khó khăn như ánh sáng kém, góc chụp khó khăn, hay nhiễu trong dữ liệu.

Các phương pháp nhận dạng khuôn mặt truyền thống sử dụng các toán tử đặc trưng để lập mô hình khuôn mặt, phương pháp này đơn giản và dễ thực hiện. Tuy nhiên, với những nghiên cứu sâu hơn, các thuật toán này có thể cho thấy hiệu quả mạnh mẽ rong việc tìm kiếm các cấu trúc tuyến tính, nhưng khi đối mặt với các cấu trúc phi tuyến tính, chúng thường đạt được kết quả nhận dạng không đạt yêu cầu.

Như vậy, quá trình nhận dạng khuôn mặt của chúng tôi chủ yếu được chia thành hai bước: nhận diện khuôn mặt và nhận dạng khuôn mặt. Đầu tiên, MTCNN được sử

dụng để nhận diện khuôn mặt để có được tọa độ khuôn mặt chính xác. Dựa trên kết quả của bước trước, FaceNet được sử dụng để nhận dạng khuôn mặt.

## 1.2 Đầu vào của bài toán

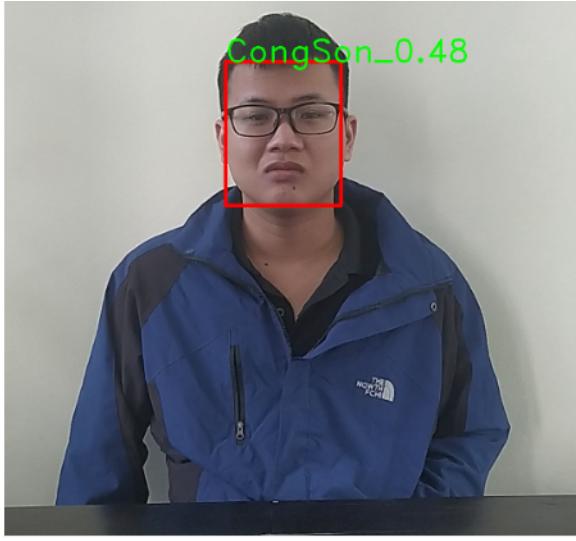
Là hình ảnh 2D lấy từ video hoặc camera của một đối tượng như hình 1.1.



Hình 1.1: Minh họa đầu vào của bài toán. Ảnh sinh viên Trần Công Sơn.

## 1.3 Đầu ra của bài toán

Kết quả quá nhận dạng khuôn mặt là một hình ảnh tương tự như hình ảnh đầu vào, tuy nhiên, có thêm một hộp dự đoán (Bounding Box) trên khuôn mặt của đối tượng, trên đó có tên (nhãn) của đối tượng như hình 1.2.



Hình 1.2: Minh họa đầu ra của bài toán. Ảnh sinh viên Trần Công Sơn với Bounding Box trên khuôn mặt và nhãn CongSon.

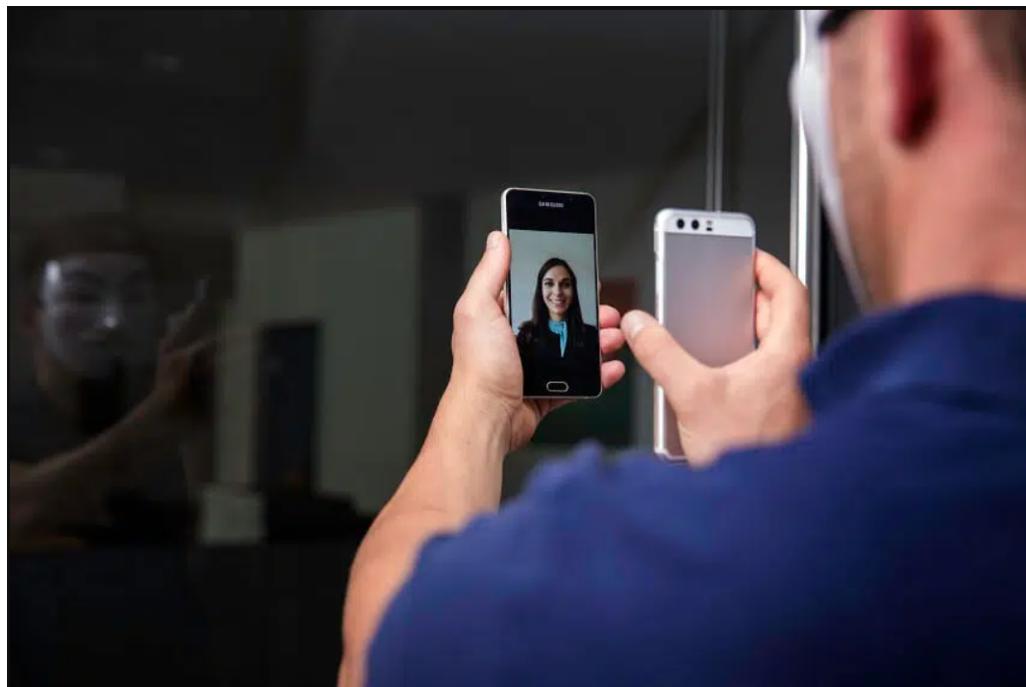
## 1.4 Các thách thức

### 1.4.1 Thách thức chung

Bài toán nhận dạng khuôn mặt để điểm danh sinh viên mang lại nhiều lợi ích về tính tiện lợi và tự động hóa quy trình quản lý lớp học. Tuy nhiên, cũng có những thách thức cần đối mặt khi triển khai hệ thống như vậy:

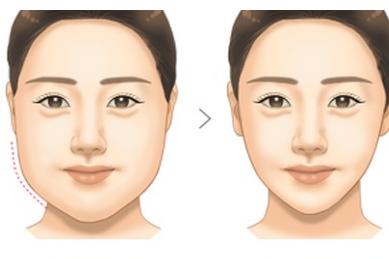
- Chất lượng Ảnh và Ánh Sáng: Nếu ảnh chất lượng thấp hoặc mờ, có thể ảnh hưởng đến độ chính xác của hệ thống; Điều kiện ánh sáng yếu hoặc đặc biệt có thể làm giảm hiệu suất của hệ thống.
- Quản lý lớp học lớn: Đối với các lớp học có số lượng sinh viên lớn, việc xử lý ảnh và nhận dạng một lượng lớn người dùng có thể đặt ra thách thức về hiệu suất.
- Quyền Riêng Tư: Việc sử dụng hệ thống nhận dạng khuôn mặt trong môi trường giáo dục có thể đối mặt với lo ngại về quyền riêng tư của sinh viên. Cần phải có chính sách và biện pháp bảo vệ quyền riêng tư chặt chẽ.
- Đồng Bộ Hóa Dữ Liệu: Đối với các hệ thống lớn, việc đồng bộ dữ liệu sinh viên và hình ảnh khuôn mặt đôi khi có thể là một thách thức, đặc biệt là khi cập nhật thông tin sinh viên.
- Giả Mạo: Có nguy cơ người khác có thể giả mạo hình ảnh khuôn mặt để thay thế cho sinh viên thực sự (bằng cách sử dụng thẻ sinh viên, cẩn cước của sinh viên,...)

bất cứ thứ gì có khuôn mặt của sinh viên đó như hình 1.3).



Hình 1.3: Minh họa việc giả mạo.(Nguồn: Mì AI)

- **Điều Kiện Sinh Học và Thay Đổi Khuôn Mặt:** Sinh viên có thể thay đổi khuôn mặt do nhiều lý do như việc để tóc, đeo kính, hoặc thậm chí phẫu thuật thẩm mỹ như hình 1.4.



Hình 1.4: Minh họa thay đổi sinh học (sau khi giảm cân).(Nguồn: random.com.vn)

- Tuân thủ pháp luật: Việc triển khai hệ thống cần phải tuân thủ các quy định pháp luật về quyền riêng tư và sử dụng dữ liệu cá nhân.

#### 1.4.2 Thách thức của nhận diện khuôn mặt MTCNN

MTCNN (Multi-task Cascaded Convolutional Networks) là một mô hình sử dụng trong bài toán nhận diện khuôn mặt, chủ yếu là để xác định vị trí và các điểm chính trên khuôn mặt. Tuy nhiên, như mọi mô hình, MTCNN cũng có những hạn chế. Dưới đây là một số điểm hạn chế của MTCNN:

- Hiệu suất trên ảnh có chất lượng thấp: MTCNN có thể gặp khó khăn khi đối mặt với ảnh có chất lượng thấp, mờ, hoặc nhiễu, do đó có thể dẫn đến việc nhận diện không chính xác hoặc thất bại.
- Xử lý chậm trên ảnh lớn: Việc xử lý ảnh lớn có thể là một thách thức đối với MTCNN. Do mô hình có cấu trúc nhiều tầng và đặc trưng, nó có thể đòi hỏi nhiều tài nguyên tính toán.
- Mặc dù MTCNN có thể hoạt động tốt trên nhiều góc độ và định dạng khuôn mặt, nhưng vẫn có những trường hợp nơi nó gặp khó khăn, đặc biệt là khi khuôn mặt nghiêng nhiều hoặc có định dạng khác biệt như hình 1.5.



Hình 1.5: Minh họa việc ảnh đối tượng có góc độ xấu làm mất đi đặc trưng khuôn mặt của đối tượng (không nhận dạng được).

- Cần đủ dữ liệu đào tạo đa dạng: Như mọi mô hình máy học, MTCNN yêu cầu một lượng lớn dữ liệu đào tạo đa dạng để đảm bảo hiệu suất tốt trên nhiều trường hợp sử dụng. Việc thiếu dữ liệu đào tạo có thể dẫn đến overfitting hoặc hiệu suất thấp trên dữ liệu mới.

### 1.4.3 Thách thức của nhận dạng khuôn mặt FaceNet

Facenet là một mô hình nhận dạng khuôn mặt sử dụng kỹ thuật học sâu để sinh ra các vectơ biểu diễn khuôn mặt chất lượng cao. Tuy nhiên, cũng giống như các mô hình khác, Facenet cũng đối mặt với một số hạn chế:

- Yêu Cầu lượng dữ liệu lớn: Facenet đòi hỏi một lượng lớn dữ liệu đào tạo để tạo ra các biểu diễn chất lượng cao. Điều này có thể là một thách thức đối với các tổ chức hoặc ứng dụng có lượng dữ liệu hạn chế.
- Cần Tài Nguyên Tính Toán Lớn: Việc triển khai và duy trì Facenet đòi hỏi tài nguyên tính toán lớn, đặc biệt là khi áp dụng mô hình trên các thiết bị máy tính cá nhân.
- Nhạy cảm với Sự Thay Đổi Ánh Sáng và Góc Độ: Facenet có thể trở nên nhạy cảm với sự thay đổi về điều kiện ánh sáng và góc độ của khuôn mặt, gây ra sự giảm chất lượng của các biểu diễn và làm tăng sai số nhận dạng.
- Khả Năng Đối Mặt với Sự Biến Đổi của Khuôn Mặt: Facenet có thể không hiệu quả đối với các biến đổi lớn của khuôn mặt như việc thay đổi hình dạng khuôn mặt (chẳng hạn, khi cười), làm tăng khả năng xảy ra sai sót.

## 1.5 Mục tiêu của tiểu luận

Các mục tiêu chính của bài tiểu luận bao gồm:

- Hiểu biết về các phương pháp nhận diện khuôn mặt: Cung cấp một cái nhìn tổng quan về các phương pháp nhận diện khuôn mặt truyền thống và hiện đại, qua đó đề cập đến lợi ích và hạn chế của mỗi phương pháp.
- Triển khai mô hình: Trình bày rõ ý tưởng triển khai mô hình MTCNN và FaceNet trong bài toán điểm danh sinh viên.
- Phân tích và đánh giá kết quả: Thực hiện phân tích sâu sắc và đánh giá kết quả

của hệ thống, bao gồm cả độ chính xác và hiệu suất thời gian thực.

- **Ưu và nhược điểm:** Biết được ưu nhược điểm, những mặt còn hạn chế của mô hình MTCNN và FaceNet trong bài toán điểm danh để có phương pháp cải tiến hợp lý.

## 1.6 Cấu trúc của tiểu luận

Sau phần giới thiệu, các phần còn lại của đồ án được tổ chức như sau:

**Phần 2** Trình bày tóm tắt các cách tiếp cận bài toán nhận dạng khuôn mặt từ các tài liệu, bài báo trước đây.

**Phần 3** Trình bày tổng quan về hai mô hình MTCNN và FaceNet. Trong đó, chúng tôi mô tả một cách khái quát các thuật toán, kỹ thuật sử dụng trong các thuật toán và đánh giá các thuật toán của tác giả.

**Phần 4** Thủ nghiệm, đánh giá, thảo luận với 2 phương pháp. Chúng tôi thực nghiệm hệ thống nhận dạng trên tập dữ liệu sinh viên tự xây dựng và tiến hành đánh giá, cải tiến kết quả.

**Phần 5** Xây dựng ứng dụng điểm danh bằng StreamLit.

**Phần 6** Kết luận chung về hai mô hình MTCNN và FaceNet đối với bài toán và đưa ra phương hướng phát triển tiếp theo cho tiểu luận.

## Chương 2

# CÁC CÁCH TIẾP CẬN

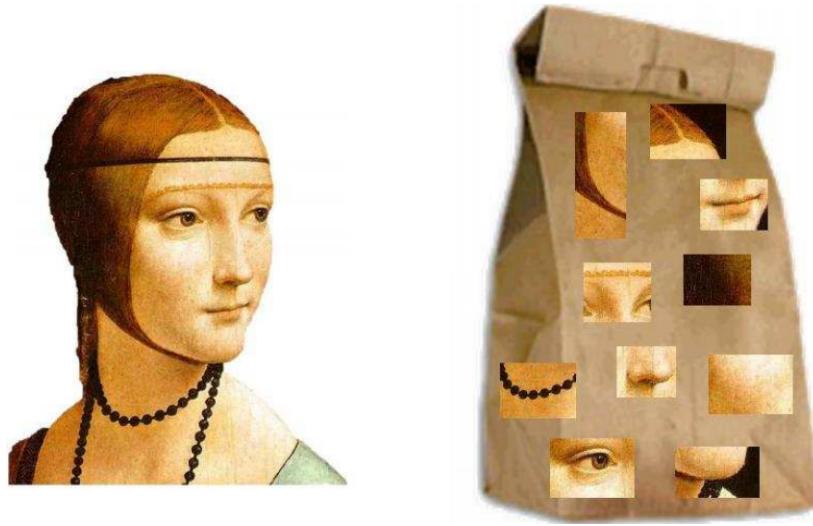
Bộ phát hiện khuôn mặt theo tầng do Viola và Jones [11] đề xuất sử dụng các tính năng Haar-Like và AdaBoost để huấn luyện các bộ phân loại xếp tầng, đạt được hiệu suất tốt với hiệu quả theo thời gian thực. Tuy nhiên, khá nhiều công trình [13] [3] chỉ ra rằng phương pháp này có thể suy giảm đáng kể trong các ứng dụng trong thế giới thực với các biến thể hình ảnh lớn hơn của khuôn mặt con người ngay cả với các tính năng và phân loại tiên tiến hơn. Gần đây, mạng thần kinh tích chập (CNN) đã đạt được những tiến bộ đáng chú ý trong nhiều nhiệm vụ thị giác máy tính, chẳng hạn như phân loại hình ảnh và nhận dạng khuôn mặt [8] [2], các tác giả này đã huấn luyện mạng lưới thần kinh tích chập sâu để nhận dạng thuộc tính khuôn mặt nhằm đạt được phản hồi cao ở các vùng khuôn mặt, từ đó mang lại nhiều cửa sổ khuôn mặt ứng cử viên hơn. Tuy nhiên, do cấu trúc CNN phức tạp nên phương pháp này tốn nhiều thời gian trong thực tế.

Ngoài các cách tiếp cận trên, còn có một số cách tiếp cận sau:

### 2.1 Sử dụng đặc trưng SIFT và BoVW

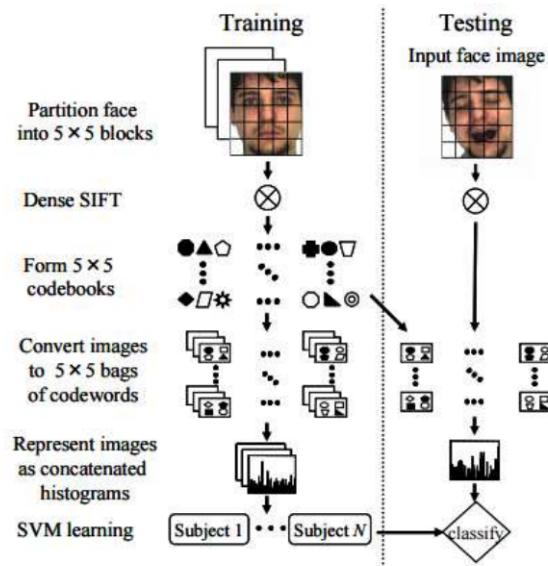
Nhóm tác giả [14] đề xuất một thuật toán Bag of Visual Words (BoVW) để nhận dạng bằng cách chia khuôn mặt thành nhiều khối đặc trưng Scale Invariant Feature Transform (SIFT), từ đó tính toán và lượng tử hóa vector thành các codeword khác nhau. Cuối cùng, ở mỗi khối ta tính tần số phân phối của mỗi codeword, sau đó nối dài các tần số từ các khối để biểu diễn khuôn mặt.

Ý tưởng của thuật toán này bắt nguồn từ BoW ở lĩnh vực xử lý ngôn ngữ tự nhiên (NLP) (Hình 2.1).



Hình 2.1: Minh họa thuật toán BoVW. Với hình ảnh cô gái bên trái, ta có các đặc trưng là mắt, mũi, miệng, ... sau đó cho vào một cái "túi" đặc trưng như hình bên phải. Nguồn [14].

Nhóm tác giả [10] đánh giá rằng các ảnh khuôn mặt đều cùng một khuôn mẫu, cho nên, nếu ta trích xuất đặc trưng khuôn mặt thành tập các phần nhỏ thì không thể đảm bảo thông tin khuôn mặt. Do đó, nhóm tác giả này đã đề xuất thuật toán rút trích đặc trưng khuôn mặt như hình 2.2.



Hình 2.2: Minh họa sơ đồ rút trích đặc trưng. Nguồn [10].

Về cơ bản, ta chia ảnh thành các khối  $5 \times 5$  và xem mỗi khối nhỏ là vùng quan tâm (ROI — Region of Interest). Với mỗi ROI, ta tính đặc trưng SIFT đặc trên mỗi đoạn lây mẫu dài 2 điểm ảnh, thu được vector SIFT 128 chiều, từ đó, mỗi khối ta thu được một tập các vector SIFT. Ở bước huấn luyện, sử dụng thuật toán k-means chuyển

đổi vector SIFT ở mỗi ROI thành các codeword. Ở trong một ROI, ta phân vùng các đặc trưng SIFT ở mỗi đoạn thành K cụm, khi đó ta định nghĩa codeword là tâm của cụm. Một codebook bao gồm K codeword của cùng một ROI và từ dữ liệu huấn luyện, ta được  $5 \times 5$  codebook. Cuối cùng, ta đổi chiều mỗi vector SIFT của mỗi đoạn ở mỗi ROI với codebook tương ứng, sử dụng biểu đồ tần số của các codeword khác nhau và dùng biểu đồ này làm đặc trưng của ROI, sau đó ta nối dài  $5 \times 5$  biểu đồ để thu được một vector biểu diễn ảnh khuôn mặt. Sử dụng SVM tuyến tính để huấn luyện biểu đồ của từng người.

Thuật toán này cho kết quả nhận dạng cao, tuy nhiên, lại mang nhược điểm lớn là chỉ hiệu quả khi ảnh **không bị che khuất** quá nhiều. Vì lý do này, BoVW không hiệu quả khi nhận dạng một phần mặt.

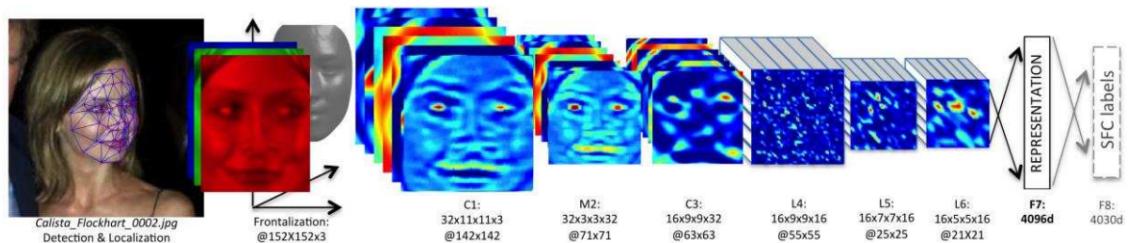
## 2.2 Sử dụng thuật toán DeepFace

Nhóm tác giả [1] từ Trung tâm nghiên cứu Facebook và trường Đại học Tel Aviv, Israel đề xuất một thuật toán có tên là DeepFace, sử dụng nguồn ảnh do người dùng đăng tải lên Facebook làm bộ dữ liệu. Về cơ bản, thuật toán này trải qua 4 bước:

- Bước 1, xác định khuôn mặt.
- Bước 2, canh chỉnh khuôn mặt.
- Bước 3, biểu diễn khuôn mặt.
- Bước 4, phân loại khuôn mặt.

Nhóm tác giả này biểu diễn khuôn mặt theo mô hình 3D, nhằm áp dụng biến đổi affine từng phần, từ đó biểu diễn khuôn mặt từ 9 lớp Mạng Neuron Sâu (Deep Neural Network - DNN), mạng này có hơn 120 ngàn tham số sử dụng một số lớp liên thông mà không chia sẻ trọng số.

Hình 2.3 dưới đây thể hiện quy trình huấn luyện của phương pháp DeepFace.



Hình 2.3: Ảnh cấu trúc huấn luyện của DeepFace, từ ảnh vào, lấy khuôn mặt, sau đó chỉnh chính diện vào mô hình 3D, tiếp theo là các lớp tích chập C1- lớp Pooling (M2) - C3, sau đó là 3 lớp Local Connected (L4-L6), cuối cùng là 2 lớp Fully Connected (F7-F8). Nguồn [1].

Về cơ bản, đầu tiên đưa ảnh vào 3D đã canh chỉnh với 3 kênh màu RGB có kích thước  $152 \times 152 \times 3$ . Sau đó, đưa ảnh tiếp vào các lớp tích chập C1- lớp Pooling M2 - C3. Mục đích của 3 lớp này là trích xuất các đặc trưng ở mức thấp như các cạnh hay kết cấu ảnh. Các lớp sau đó là L4-L6, có tác dụng áp dụng băng lọc. Cuối cùng, hai lớp F7 và F8 có tác dụng bắt được mối quan hệ đặc trưng giữa các phần xa trong khuôn mặt.

Thuật toán này là một trong những thuật toán nhận dạng khuôn mặt có độ chính xác thuộc "top performing". Tương tự với thuật toán DeepFace, là thuật toán FaceNet, nhưng với lượng dữ liệu tự xây dựng khá nhỏ, nên chúng tôi đã quyết định chọn phương pháp FaceNet để thực hiện việc nhận dạng.

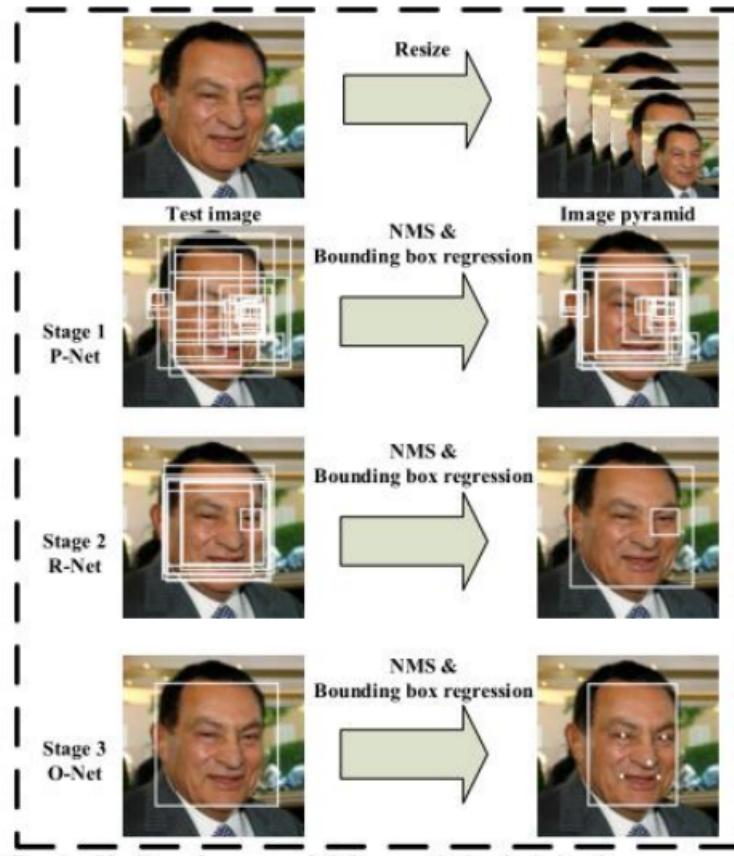
## Chương 3

# PHƯƠNG PHÁP THỰC HIỆN

### 3.1 Giới thiệu Multi-task Cascaded Convolutional Networks (MTCNN)

Multi-task Cascaded Convolutional Networks (MTCNN) là một framework được sử dụng để phát hiện khuôn mặt người sử dụng kiến trúc nhiều tầng và ba bộ phận riêng biệt kết nối với nhau [16] (Hình 3.1).

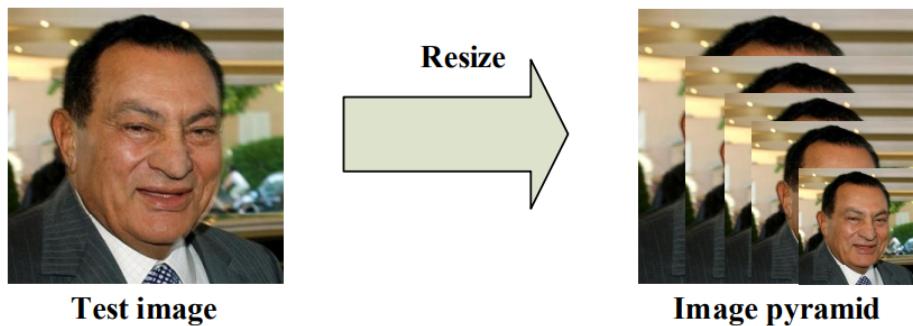
MTCNN khai thác mối tương quan vốn có giữa phát hiện và căn chỉnh để tăng hiệu suất của chúng. Khung MTCNN tận dụng kiến trúc xếp tầng với ba giai đoạn của mạng tích chập sâu được thiết kế cẩn thận để dự đoán vị trí khuôn mặt và điểm mốc theo cách từ thô đến tinh.



Hình 3.1: Hình ảnh các mạng tích chập sâu đa tác vụ ba giai đoạn. Thứ nhất, cửa sổ ứng viên được tạo thông qua Mạng đề xuất nhanh (P-Net). Sau đó, mạng tinh chỉnh những ứng cử viên này trong giai đoạn tiếp theo thông qua Mạng sàng lọc (RNet). Trong giai đoạn thứ ba, Mạng đầu ra (O-Net) tạo ra hộp giới hạn cuối cùng. Nguồn: [5]

Như vậy, cho một hình ảnh ban đầu, ta tiến hành thay đổi kích thước của nó thành các tỷ lệ khác nhau để xây dựng một kim tự tháp hình ảnh, sau đó ta đưa vào mô hình MTCNN, mô hình này sẽ thực hiện bao gồm 3 giai đoạn:

- Giai đoạn 1: Proposal Network (P-Net)
- Giai đoạn 2: Refine Network (R-Net)
- Giai đoạn 3: Output Network (O-Net)



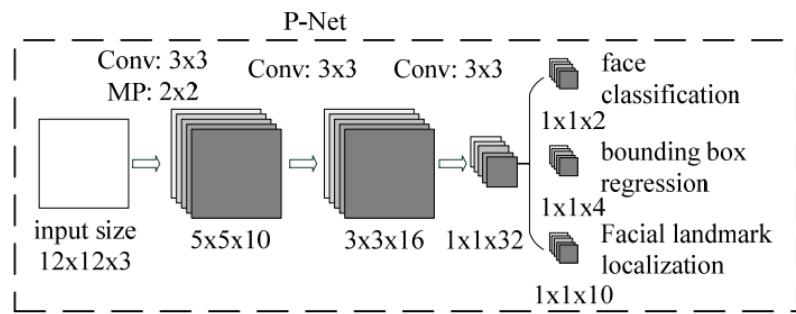
Hình 3.2: Image pyramid. Nguồn: Zhang et al [16]

### 3.1.1 Proposal Network (P-Net)

Giai đoạn đầu tiên trong MTCNN là sử dụng P-Net nhằm đề xuất ra các ứng viên là vùng chứa khuôn mặt trong ảnh và các vector bounding box regression.

Trước hết, một bức ảnh thường sẽ có nhiều hơn một người – một khuôn mặt. Ngoài ra, những khuôn mặt thường sẽ có kích thước khác nhau. Ta cần một phương thức để có thể nhận dạng toàn bộ số khuôn mặt đó, ở các kích thước khác nhau. MTCNN đưa cho chúng ta một giải pháp, bằng cách sử dụng phép Resize ảnh, để tạo một loạt các bản copy từ ảnh gốc với kích cỡ khác nhau, từ to đến nhỏ, tạo thành 1 Image Pyramid.

Image pyramid là phương pháp thu phóng hình ảnh để tạo ra hàng loạt bản sao của một hình ảnh ban đầu với kích cỡ khác nhau từ to đến nhỏ dần (Hình 3.2).



Hình 3.3: P-Net ("Conv" là lớp tích chập; "MP" là lớp gộp (Pooling Layer)). Nguồn: Zhang et al [16]

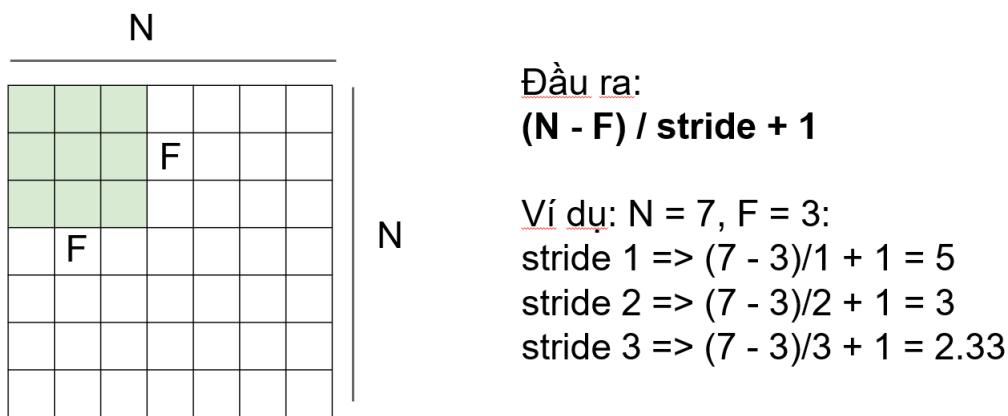
Đầu tiên, nhìn vào hình 3.3, ta cần biết ba khái niệm:

- face classification: Mục tiêu của face classification là xác định xem mỗi bounding box ứng viên có chứa khuôn mặt hay không. Cho nên đây là một bài toán nhị phân giữa hai lớp: có khuôn mặt (face) hoặc không có khuôn mặt (non-face).
- bounding box regression: Mục tiêu của bounding box regression là dự đoán

những thay đổi cần thiết để điều chỉnh vị trí của các bounding boxes ứng viên. Cụ thể, kernel 1x1x4 sẽ tạo ra bốn giá trị, mỗi giá trị đại diện cho thay đổi về tọa độ x, y, chiều rộng, và chiều cao của bounding box.

- Facial landmark localization: Mục tiêu của facial landmark localization là dự đoán vị trí của các điểm mốc trên khuôn mặt, như mắt, mũi, và miệng. MTCNN thường dự đoán vị trí của 5 điểm mốc trên khuôn mặt, bao gồm mắt trái, mắt phải, mũi, và hai miệng. Với mỗi điểm mốc, cần hai giá trị để xác định tọa độ (x, y), và do đó, cần 10 giá trị tổng cộng ( $2 \times 5$ ).

Ngoài ra, chúng ta cần biết cách tính toán ở các bước (Hình 3.4):



Hình 3.4: Hình minh họa cách tính đầu ra sau khi qua lớp Tích chập. Nguồn: Stanford

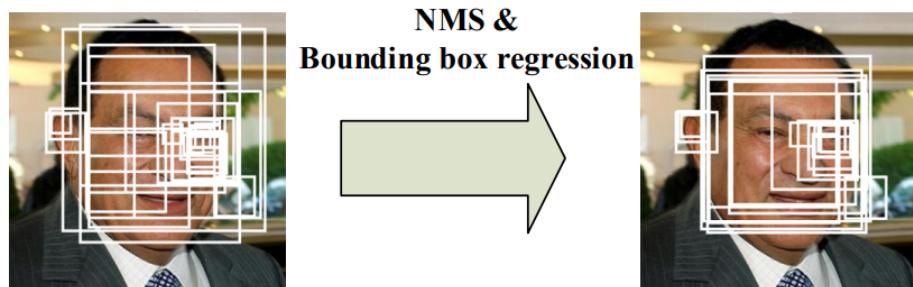
- Với đầu vào là input size  $12 \times 12 \times 3$ , sau khi đi qua lớp Conv với 10 bộ lọc  $3 \times 3$ , bước nhảy (stride) là 1 thì ta có đầu ra là  $(10 \times 10 \times 10)$  từ phép tính  $((12-3)/1 + 1)$ . Tiếp theo, sau khi qua lớp gộp (MP) thì kích thước của đầu ra giảm đi một nửa. Như vậy, kết quả sẽ là  $5 \times 5 \times 10$ .
- Tương tự như cách tính ở bước trước, với đầu vào  $5 \times 5 \times 10$ , sau khi đi qua lớp Conv với 16 bộ lọc  $3 \times 3$ , bước nhảy là 1 thì đầu ra sẽ là  $3 \times 3 \times 16$  từ phép tính  $((5-3)/1+1)$ .
- Cuối cùng, đầu ra sẽ là  $1 \times 1 \times 32$  từ phép tính  $((3-3)/1 + 1)$ .

Nhìn vào hình 3.3, ta thấy với mỗi một phiên bản copy-resize của ảnh gốc, ta sử dụng bộ lọc (kernel)  $12 \times 12$  pixel để đi qua toàn bộ bức ảnh, dò tìm khuôn mặt. Vì các bản sao của ảnh gốc có kích thước khác nhau, cho nên mạng có thể dễ dàng nhận biết được các khuôn mặt với kích thước khác nhau, mặc dù chỉ dùng 1 kernel với kích

thuộc cố định (Ảnh to hơn, mặt to hơn; Ảnh nhỏ hơn, mặt nhỏ hơn).

Sau đó, ta sẽ đưa những kernels được cắt ra từ trên và truyền qua mạng P-Net (Proposal Network). Kết quả của mạng cho ra một loạt các bounding boxes nằm trong mỗi kernel, mỗi bounding boxes sẽ chứa tọa độ 4 góc để xác định vị trí trong kernel chứa nó (đã được normalize về khoảng từ (0,1)) và điểm confident (Điểm tự tin) tương ứng.

Kết quả thu được là hình ảnh với nhiều hộp dự đoán (bounding box). Để loại trừ bớt các bounding box trên các bức ảnh và các kernels, ta sử dụng 2 phương pháp chính là lập mức ngưỡng tự tin (confident threshold) nhằm xóa đi các bounding box có mức confident thấp (tức là khả năng nó chứa khuôn mặt thấp). Cuối cùng, sử dụng Non-Maximum Suppression (NMS) để xóa các bounding box có tỷ lệ trùng nhau vượt qua ngưỡng cho trước (Hình 3.5).



Hình 3.5: Kết quả sử dụng Non-Maximum Suppression cho P-Net. Nguồn: Zhang et al [16]

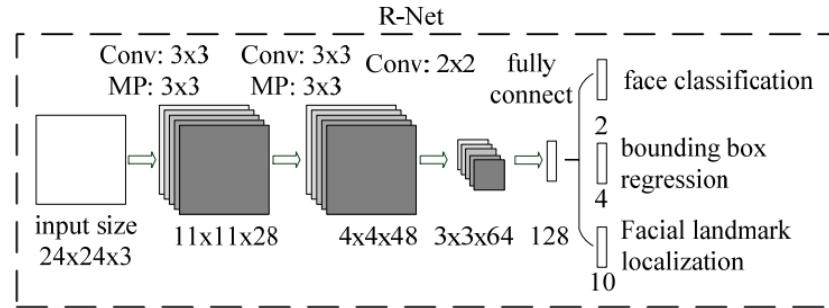
### 3.1.2 Refine Network (R-Net)

R-Net nhận các bounding box khuôn mặt được tạo ra bởi P-Net, đây là boxes có thể chứa khuôn mặt dựa trên việc phát hiện các đặc trưng của khuôn mặt.

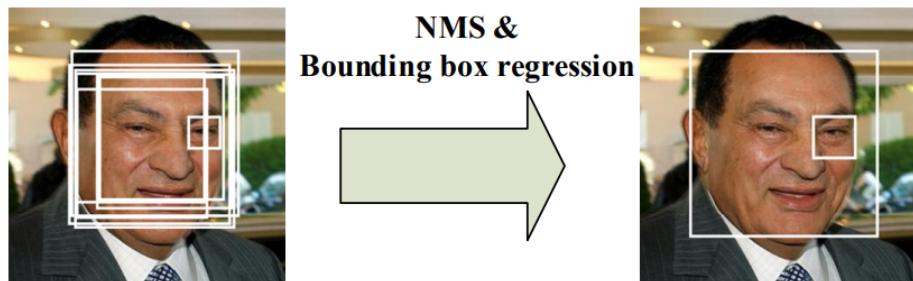
Mạng R-Net (Refine Network - Hình 3.6) thực hiện các bước như mạng P-Net. Tuy nhiên, ở mạng này, có sử dụng một phương pháp tên là padding, nhằm thực hiện việc chèn thêm các zero-pixels vào các phần thiếu của bounding box nếu bounding box bị vượt quá biên của ảnh. Tất cả các bounding box lúc này sẽ được resize về kích thước  $24 \times 24$ , được coi như 1 kernel và đưa vào mạng R-Net. Kết quả sau cũng là những tọa độ mới của các box còn lại (hình 3.7) và được đưa vào mạng tiếp theo, mạng O-Net.

Sau khi trích xuất đặc trưng, R-Net sử dụng các lớp kết nối đầy đủ (fully connected) để dự đoán các tham số cần thiết để cân chỉnh bounding box xung quanh

khuôn mặt. Điều này giúp cải thiện độ chính xác của việc xác định vị trí và kích thước của khuôn mặt.

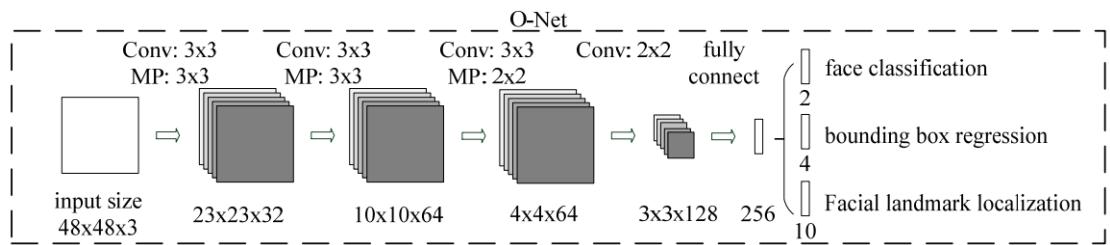


Hình 3.6: R-Net. Nguồn: Zhang et al [16]



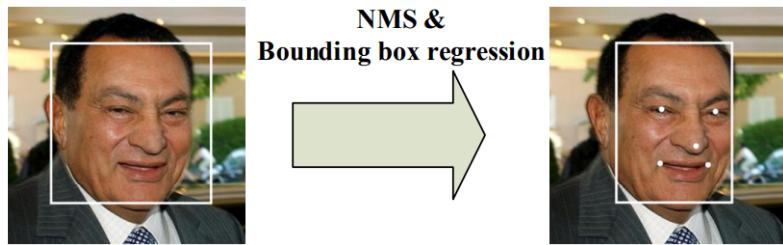
Hình 3.7: Kết quả sử dụng Non-Maximum Suppression cho R-Net

### 3.1.3 Output Network (O-Net)



Hình 3.8: O-Net. Nguồn: Zhang et al [16]

Cuối cùng, O-Net (Hình 3.8) cũng thực hiện tương tự như việc trong mạng R, thay đổi kích thước thành  $48 \times 48$ . Tuy nhiên, kết quả đầu ra của mạng lúc này không còn chỉ là các tọa độ của các box nữa, mà trả về 3 giá trị bao gồm: 4 tọa độ của bounding box, tọa độ 5 điểm landmark trên mặt, bao gồm 2 mắt, 1 mũi, 2 bên cánh môi và điểm confident của mỗi box (Hình 3.9).



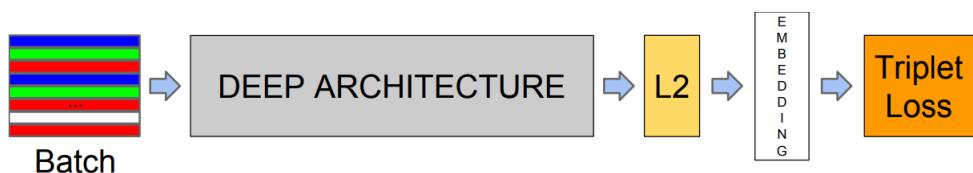
Hình 3.9: Kết quả sử dụng R-Net cùng với Non-Maximum Suppression. Nguồn: Zhang et al [16]

## 3.2 Giới thiệu FaceNet

Nhóm tác giả [4] từ Google đề xuất một thuật toán có tên là FaceNet sẽ học cách ánh xạ từ ảnh khuôn mặt vào không gian Euclidean compact với khoảng cách đo được tương ứng với độ tương đồng khuôn mặt. Thuật toán này có thể tạo ra vector đặc trưng và nhúng vào bài toán nhận dạng khuôn mặt. Nhóm tác giả này sử dụng Mạng Tích Chập Sâu (Deep Convolution Network - DNN) được huấn luyện để tự tối ưu hóa bài toán. Mạng được huấn luyện sao cho khoảng cách  $L_2$  bình phương trong không gian nhúng tương ứng với mức độ tương đồng của khuôn mặt: Mặt cùng người sẽ có khoảng cách nhỏ, mặt khác người sẽ có khoảng cách lớn.

Nhiều thuật toán nhận dạng khuôn mặt sử dụng DNN trước đây sử dụng lớp phân loại đã qua huấn luyện trên toàn bộ ảnh đã biết nhãn, sau đó lấy lớp thắt cổ chai trung bình (medium bottleneck layer) để biểu diễn tông quát cho tập huấn luyện. Tuy nhiên, mặt tiêu cực là lớp thắt cổ chai đôi khi không rõ và không tiện lợi do lớp thắt cổ chai này thường rất lớn (khoảng 1000 chiều). Để khắc phục điều này, FaceNet huấn luyện output thành nhúng compact 128 chiều sử dụng hàm bộ ba sai số dựa trên LMNN [12], mẫu bộ ba này gồm 2 ảnh cùng loại và 1 ảnh khác loại và hàm sai số có nhiệm vụ tách ảnh đúng ra khỏi ảnh sai dựa vào biên khoảng cách. Nhóm tác giả sử dụng 2 kiến trúc Mạng Tích Chập Sâu, một mạng dựa theo mô hình của Zeiler và Fergus [15], mạng còn lại sử dụng mô hình Inception từ GoogLeNet [9].

### 3.2.1 FaceNet



Hình 3.10: Kiến trúc của FaceNet. Nguồn: Schaff et al 2015 [7]

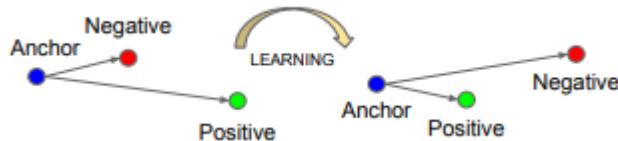
FaceNet sử dụng DNN, giả sử cấu trúc mô hình là một khối lớp (Hình 3.10), sau khi sử dụng cấu trúc CNN, vẫn đề quan trọng nằm ở kết quả sau khi huấn luyện. Do đó, nhóm tác giả đã sử dụng đến bộ ba sai số có thể giúp kiểm tra, nhận dạng và phân cụm khuôn mặt. Giả sử ta có ảnh  $x$ , đưa qua hàm nhúng  $f(x)$  vào không gian đặc trưng  $R^d$  sao cho khoảng cách bình phương của tất cả khuôn mặt cùng loại phải nhỏ hơn khoảng cách bình phương với mặt khác loại.

### 3.2.2 Triplet Loss

Mục đích của việc sử dụng Triplet loss trong FaceNet, sau khi đã có được embedding của hình ảnh dưới dạng vector, ta áp dụng loss function như Triplet loss nhằm phân cụm các khuôn mặt giống nhau gần nhau lại trong vector không gian, và tách khuôn mặt khác nhau ra xa

Hàm Triplet loss gồm có ba thành phần chính:

- Anchor: Ảnh gốc của một người, được sử dụng làm mốc để xác định khoảng cách đến các ảnh được gọi là Positive và Negative của một người.
- Positive: Ảnh cùng một người với Anchor, được sử dụng làm để xác định khoảng cách từ Anchor đến Positive.
- Negative: Ảnh của một người khác với Anchor, được sử dụng làm để xác định khoảng cách từ Anchor đến Negative.



Hình 3.11: Bộ ba sai số tối thiểu hoá khoảng cách giữa ảnh vào (Anchor) và ảnh cùng loại với ảnh vào (Positive) và tối đa hoá khoảng cách giữa ảnh vào và ảnh khác loại với ảnh vào (Negative). Nguồn: Schoff et al 2015 [7]

Như vậy, hình 3.11 cho thấy quy trình huấn luyện.

Công thức của Triplet loss được tính như sau:

$$L(A, P, N) = \sum_{i=0}^n ||f(A_i) - f(P_i)||_2^2 - ||f(A_i) - f(N_i)||_2^2 + \alpha \quad (3.1)$$

## Chương 4

# THỬ NGHIỆM VÀ ĐÁNH GIÁ

Trong phần này, chúng tôi thử nghiệm hai mô hình MTCNN và FaceNet trên tập dữ liệu chính chúng tôi tạo ra để đánh giá hai thuật toán này.

## 4.1 Tổng quan các bước thử nghiệm

Quá trình thử nghiệm hai thuật toán trên cùng một thiết bị máy tính trên một tập dữ liệu. Chúng tôi thực hiện so sánh và đánh giá dựa trên kết quả của phương pháp trên tập dữ liệu.

Đầu tiên, chúng tôi quay 15 video dữ liệu đầu vào. Sau đó, thực hiện chuyển dữ liệu là các video thành tập ảnh bằng phương pháp MTCNN (Tổng cộng có 15 tập dữ liệu của 15 đối tượng).

Sau đó, sử dụng các tập dữ liệu này để đưa vào mô hình FaceNet.

## 4.2 Dữ liệu đánh giá

### 4.2.1 Cấu trúc dữ liệu đầu vào

Dữ liệu đầu vào bao gồm 2 phần chính (Hình 4.1):

- Thư mục chứa ảnh của đối tượng: các hình ảnh được cắt từ video (từ nhiều góc độ khác nhau của đối tượng).
- Thư mục chứa video của đối tượng.

```

|-- Dataset_name
    |-- videos
        |-- MongDo.mp4
        .....  

        |-- ThuThao.mp4
    |-- images
        |-- MongDo:  

            |-- image_0.jpg
            .....  

            |-- image_N.jpg
        .....  

        |-- ThuThao:  

            |-- image_0.jpg
            .....  

            |-- image_N.jpg

```

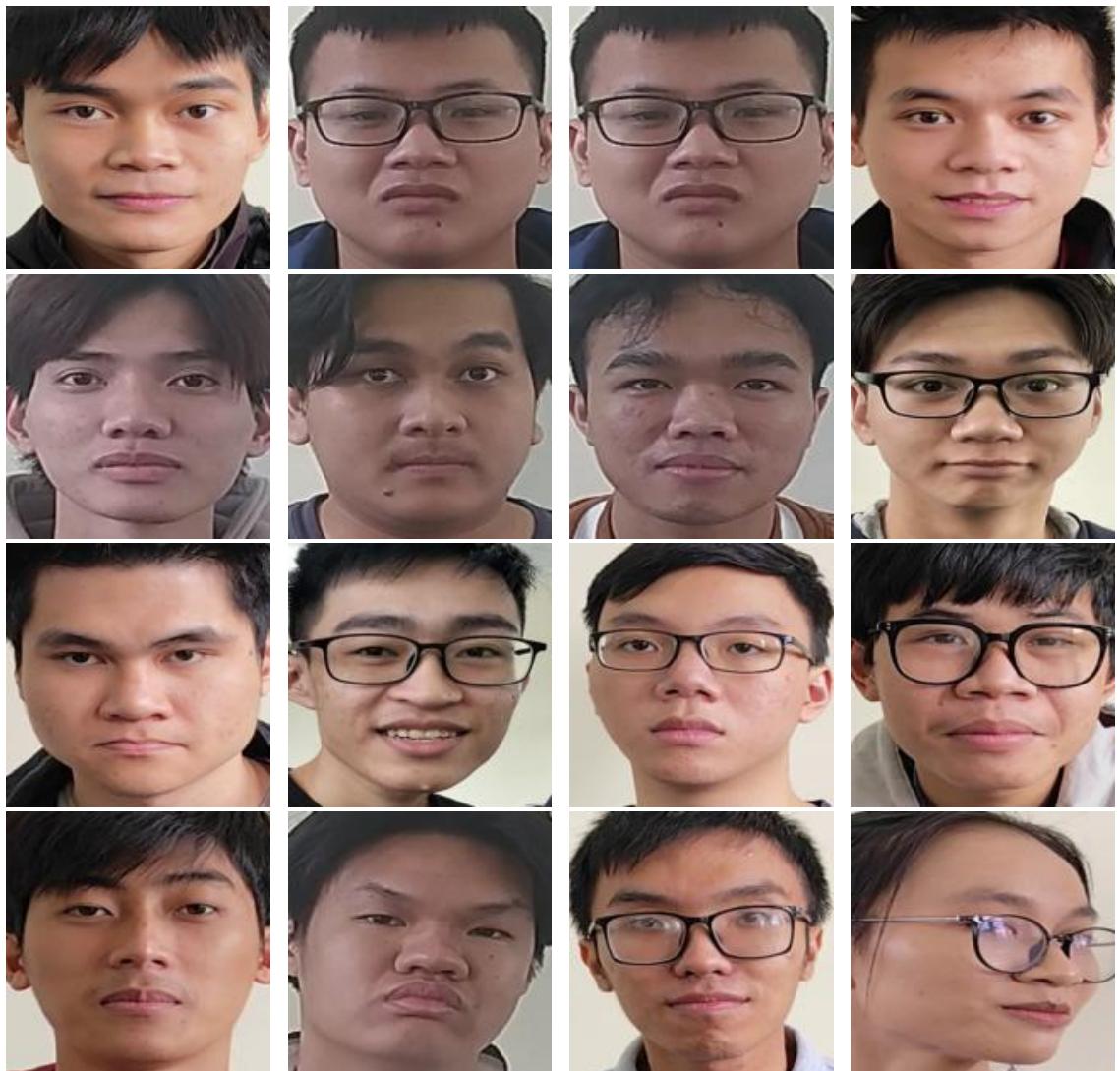
Hình 4.1: Mô tả cấu trúc dữ liệu đầu vào.

### 4.2.2 Thông tin dữ liệu

Thông tin 15 tập dữ liệu do chính chúng tôi tự tạo được để cập ở bảng 4.1 bên dưới.

Bảng 4.1: Thông tin bộ dữ liệu thực nghiệm.

STT	Tên dữ liệu	Số lượng ảnh	Kích thước ảnh	Môi trường
1	Nguyễn Luôn Mong Đổ	377	160x160	Trong nhà
2	Trần Công Sơn	433	160x160	Trong nhà
3	Võ Đạt Văn	350	160x160	Trong nhà
4	Nguyễn Khánh Dương	627	160x160	Trong nhà
5	Lê Kỳ Nam	570	160x160	Trong nhà
6	Lê Công Lương	574	160x160	Trong nhà
7	Huỳnh Văn Nguyên Bảo	448	160x160	Trong nhà
8	Lý Nhật Phương	350	160x160	Trong nhà
9	Võ Phúc Duy	341	160x160	Trong nhà
10	Nguyễn Ngọc Quang Huy	580	160x160	Trong nhà
11	Nguyễn Văn Tiến 591	429	160x160	Trong nhà
12	Nguyễn Tiến Nhật	575	160x160	Trong nhà
13	Trần Tuấn Anh	557	160x160	Trong nhà
14	Nguyễn Văn Tiến 585	458	160x160	Trong nhà
15	Ngô Thị Thu Thảo	65	160x160	Trong nhà



Hình 4.2: Một số hình ảnh của các tập dữ liệu thử nghiệm.

### **4.2.3 Quy trình tạo dữ liệu**

Để tạo dữ liệu cho đầu vào cho mô hình, chúng tôi thực hiện quay video đối tượng cần nhận diện (chúng tôi dùng thiết bị điện thoại di động thông minh Redmi Note 8).

## **4.3 Kết quả thí nghiệm và thảo luận**

### **4.3.1 Cách đánh giá**

Chúng tôi thực hiện đánh giá dựa trên 2 tiêu chí:

- Độ chính xác.
- Ma trận nhầm lẫn. Đối với tiêu chí này, chúng tôi tiến hành nghiên cứu lại những đối tượng có số lượng dự đoán ít nhất để có thể tìm ra lí do. Đồng thời, chúng tôi cũng nghiên cứu đối tượng có số lượng dự đoán nhiều nhất để ra lí do.

Chúng tôi sẽ thực hiện 2 lần thực nghiệm với các tham số khác nhau.

### **4.3.2 Kết quả**

#### **Thiết lập tham số thực nghiệm lần 1:**

- MTCNN: margin = 20; threshold = [0.6, 0.7, 0.7]. Tham số 'margin' có ý nghĩa xác định khoảng cách giữa bounding box dự đoán và khuôn mặt thực tế. Tham số 'threshold' được sử dụng để quyết định độ tin cậy (confident score) tối thiểu.
- FaceNet: classify = False; pretrained = casia-webface. Tham số 'classify' cho biết rằng mô hình không sử dụng để phân loại. Tham số 'pretrained' cho biết rằng mô hình được khởi tạo trên bộ dữ liệu CASIA-WebFace.
- KNN: n-neighbors = 5. Tham số 'n-neighbors' xác định số lượng láng giềng gần nhất mà mô hình sẽ quyết định lớp có điểm dữ liệu mới.

## Kết quả thực nghiệm lần 1:

- Độ chính xác: 0.851.
- Ma trận nhầm lẫn 4.3

		Confusion Matrix																	
		True																	
		Predicted																	
CongLuong -	122	0 0 2 2 2 2 0 0 0 0 0 0 1 5 0 1																	
CongSon -	1	62 1 3 1 0 0 0 0 0 4 0 0 0 0 0 0																	
DatVan -	4	0 55 1 0 3 1 3 1 0 0 1 1 0 0 0 0 0																	
KhanhDuong -	1	0 0 134 0 2 0 0 0 0 0 0 0 0 3 0 0																	
KyNam -	4	3 3 4 93 2 1 0 0 0 0 0 0 1 0 0 1 0 1																	
MongDo -	1	1 3 4 1 55 0 5 1 1 0 2 0 0 0 0 0 0																	
NguyenBao -	2	1 0 1 2 1 80 1 2 1 0 0 0 0 0 1 0 0																	
NhatPhuong -	6	0 1 1 1 2 0 60 1 0 0 2 0 0 0 0 0 0 0																	
PhucDuy -	1	4 0 0 2 0 3 0 50 3 0 1 0 3 0 0 0 0 0																	
QuangHuy -	3	6 1 3 5 1 4 1 0 83 0 0 1 2 2 0 0 0 0																	
ThuThao -	0	0 0 0 0 0 0 0 0 0 12 0 0 0 0 0 0 0 0 0																	
TienNhat -	3	0 0 3 1 1 1 1 0 0 0 0 0 0 0 111 0 0 0																	
TuanAnh -	2	1 0 2 2 1 0 0 0 0 0 0 0 1 92 0 0 0 0																	
VanTien585 -	1	1 0 0 1 1 0 0 1 1 0 0 0 1 69 0 0 0 0																	
VanTien591 -	1	0 0 1 0 0 3 0 5 3 0 1 0 6 69 0 0 0 0																	

Hình 4.3: Ma trận nhầm lẫn (thực nghiệm lần 1).

## Thiết lập tham số thực nghiệm lần 2:

- MTCNN: margin = 20; threshold = [0.6, 0.7, 0.7]. Tham số 'margin' có ý nghĩa xác định khoảng cách giữa bounding box dự đoán và khuôn mặt thực tế. Tham số 'threshold' được sử dụng để quyết định độ tin cậy (confident score) tối thiểu.
- FaceNet: classify = False; pretrained = casia-webface. Tham số 'classify' cho biết rằng mô hình không sử dụng để phân loại. Tham số 'pretrained' cho biết rằng mô hình được khởi tạo trên bộ dữ liệu CASIA-WebFace.
- KNN: n-neighbors = 3; 'weights' = 'distance'. Tham số 'n-neighbors' xác định số lượng láng giềng gần nhất mà mô hình sẽ quyết định lớp có điểm dữ liệu mới. Tham số 'weights' xác định cách mà mô hình tính toán trọng số khi dự đoán nhãn.

## Kết quả thực nghiệm lần 2:

- Độ chính xác: 0.895.

- Ma trận nhầm lẫn 4.4

		Confusion Matrix																	
		CongLuong	CongSon	DatVan	KhanhDuong	KyNam	MongDo	NguyenBao	NhatPhuong	PhucDuy	QuangHuy	ThuThao	TienNhat	TuanAnh	VanTien585	VanTien591			
True	CongLuong	-127	0	0	1	2	0	0	0	0	0	0	0	0	4	0	1		
	CongSon	-1	66	1	1	0	0	0	0	1	2	0	0	0	0	0	0		
	DatVan	-5	0	58	1	0	0	1	3	0	0	0	1	0	0	1			
	KhanhDuong	-0	0	0	136	0	2	0	0	0	0	0	0	2	0	0			
	KyNam	-1	3	0	1	101	3	2	0	0	0	0	0	1	0	0			
	MongDo	-0	1	2	2	1	59	1	3	0	1	0	3	1	0	0			
	NguyenBao	-1	1	0	0	2	0	82	1	1	0	0	0	2	1	1			
	NhatPhuong	-1	0	0	1	1	3	0	64	1	0	0	2	1	0	0			
	PhucDuy	-0	3	0	0	1	0	2	0	53	4	0	1	0	3	0			
	QuangHuy	-1	2	1	2	3	1	3	1	2	95	0	0	1	0	0			
	ThuThao	-0	0	0	0	0	0	0	0	0	12	0	0	0	0	0			
	TienNhat	-2	0	0	1	1	1	1	0	0	0	0	113	1	1	0			
	TuanAnh	-2	0	0	1	1	1	1	0	0	0	0	0	95	0	0			
	VanTien585	-0	1	0	0	1	1	0	0	0	1	0	0	0	72	0			
	VanTien591	-1	0	0	1	0	0	3	0	4	2	0	1	0	4	73			
		CongLuong	CongSon	DatVan	KhanhDuong	KyNam	MongDo	NguyenBao	NhatPhuong	PhucDuy	QuangHuy	ThuThao	TienNhat	TuanAnh	VanTien585	VanTien591			

Hình 4.4: Ma trận nhầm lẫn (thực nghiệm lần 2).

### 4.3.3 Thảo luận

Từ các kết quả thực nghiệm trên các dữ liệu do chúng tôi tự tạo thì nhìn chung các kết quả thực nghiệm đều cho thông số khá tốt.

Sau khi thiết lập tham số lần 1, chúng tôi thực hiện kết hợp Grid Search để tìm ra tham số tối ưu cho mô hình KNN (n-neighbors = 3; weight = 'distance') và tiến hành thiết lập lại tham số lần 2.

Nhìn vào cả hai ma trận nhầm lẫn 4.3 và 4.4 ở hai lần thực nghiệm, chúng ta nhận thấy được đối tượng Nguyễn Ngọc Quang Huy đang có tỉ lệ dự đoán sai nhiều nhất (cũng như là dự đoán sai đến nhiều người khác nhau nhất). Vì vậy, chúng tôi tiến hành kiểm tra lại đầu vào của đối tượng.

Hình 4.5 trên là 4 trong số 15 hình ảnh nhận diện sai của đối tượng, chúng tôi nhận thấy rằng hầu hết những dự đoán sai này đều do đối tượng đã nhìn về phía ánh sáng cửa sổ, việc đối tượng có mang kính đã khiến cho ánh sáng bị biến đổi lớn, làm



Hình 4.5: Một số hình ảnh nhận diện sai của đối tượng Nguyễn Ngọc Quang Huy.

mô hình nhận diện sai.

Để kiểm tra cho quan sát trên, chúng tôi thực hiện nghiên cứu thêm đối tượng Nguyễn Văn Tiên (VanTien591). Kết quả là có những sự tương đồng về lí do nhận diện sai theo hình 4.6.



Hình 4.6: Một số hình ảnh nhận diện sai của đối tượng Nguyễn Văn Tiên.

# Chương 5

## Xây dựng ứng dụng

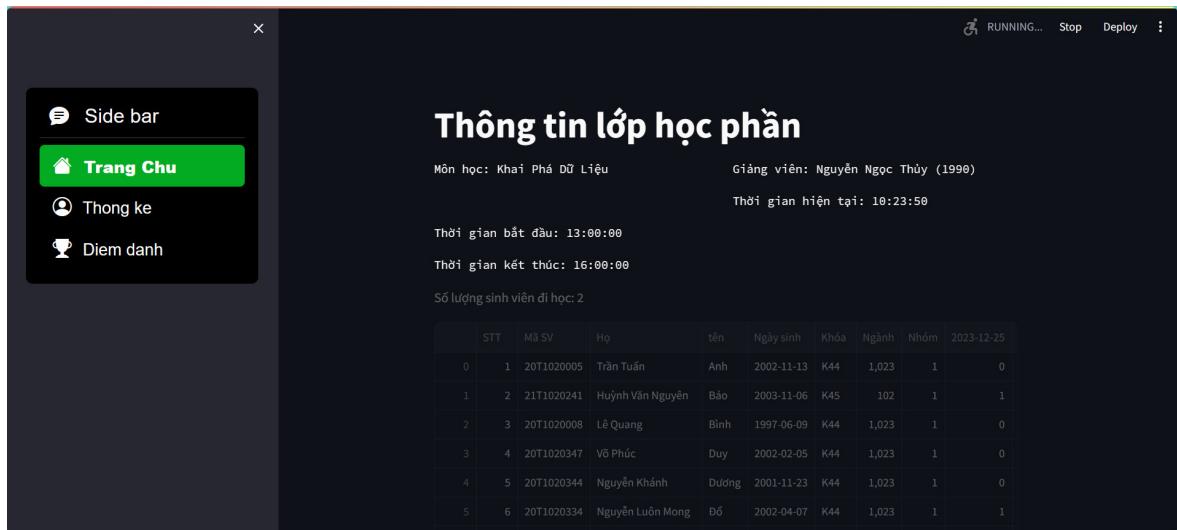
### 5.1 Giao diện ứng dụng

Ứng dụng được xây dựng và phát triển trên nền tảng web, với 2 chức năng chính:

- Thực hiện điểm danh sinh viên dựa vào camera được trang bị.
- Thống kê danh sách sinh viên đã đi học, vắng học trong buổi học ngày hôm đó.

Ứng dụng gồm có 3 trang chính bao gồm:

#### 5.1.1 Trang chủ



Hình 5.1: Giao diện trang chủ

#### 5.1.2 Thống kê

Trang thống kê có các chức năng chính:

- Hiển thị danh sách sinh viên đăng ký học phần.
- Hiển thị danh sách sinh viên đi học vào buổi học hiện tại.
- Hiển thị danh sách sinh viên nghỉ học tại buổi học hiện tại.

**Tên Lớp Học Phân:**  
Khai phá dữ liệu - Nhóm 1

**Mã Lớp Học Phân:**  
2023-2024.1.TIN4103.001

**Giảng Viên:**  
Nguyễn Ngọc Thúy (1990)

**Home**

**Danh sách sinh viên của lớp học phần**

**Thống Kê Sinh Viên Ngày 27/12/2023**

Số lượng sinh viên nghỉ học: 18

Số lượng sinh viên đi học: 2

Hình 5.2: Thống kê chung về tình trạng lớp học

STT	Mã SV	Họ	tên	Ngày sinh	Khóa	Ngành	Nhóm	2023-12-25	
0	1	20T1020005	Trần Tuấn	Anh	2002-11-13	K44	1,023	1	0
1	2	21T1020241	Huỳnh Văn Nguyên	Bảo	2003-11-06	K45	102	1	1
2	3	20T1020008	Lê Quang	Bình	1997-06-09	K44	1,023	1	0
3	4	20T1020347	Võ Phúc	Duy	2002-02-05	K44	1,023	1	0
4	5	20T1020344	Nguyễn Khánh	Đương	2001-11-23	K44	1,023	1	0
5	6	20T1020334	Nguyễn Luôn Mong	Đỗ	2002-04-07	K44	1,023	1	1
6	7	20T1020397	Nguyễn Ngọc Quang	Huy	2002-05-21	K44	1,023	1	0
7	8	20T1020421	Nguyễn Văn	Khánh	2002-01-18	K44	1,023	1	0
8	9	20T1020455	Lê Công	Lương	2002-07-09	K44	1,023	1	0
9	10	20T1020211	Lê Kỳ	Nam	2002-06-16	K44	1,023	1	0

**Danh Sách Sinh Viên Nghỉ Học**

**Danh Sách Sinh Viên Di Học**

Hình 5.3: Danh sách sinh viên của lớp học phần

STT	Mã SV	Họ	tên	Ngày sinh	Khóa	Ngành
3	20T1020341	Võ Phúc	Duy	2002-02-05	K44	1,023
4	20T1020344	Nguyễn Khánh	Dương	2001-11-23	K44	1,023
5	20T1020334	Nguyễn Luân Mong	Đỗ	2002-04-07	K44	1,023
6	20T1020397	Nguyễn Ngọc Quang	Huy	2002-05-21	K44	1,023
7	20T1020421	Nguyễn Văn	Khánh	2002-01-18	K44	1,023
9	20T1020211	Lê Kỳ	Nam	2002-06-16	K44	1,023
10	20T1020469	Nguyễn Hoài	Nam	2002-07-22	K44	1,023
11	20T1020488	Nguyễn Tiến	Nhật	2002-07-23	K44	1,023
12	20T1020077	Lê Bá Tuấn	Phong	1998-03-05	K44	1,023
13	20T1020512	Lý Nhật	Phương	2002-07-28	K44	1,023
14	20T1020542	Trần Công	Sơn	2002-07-26	K44	1,023

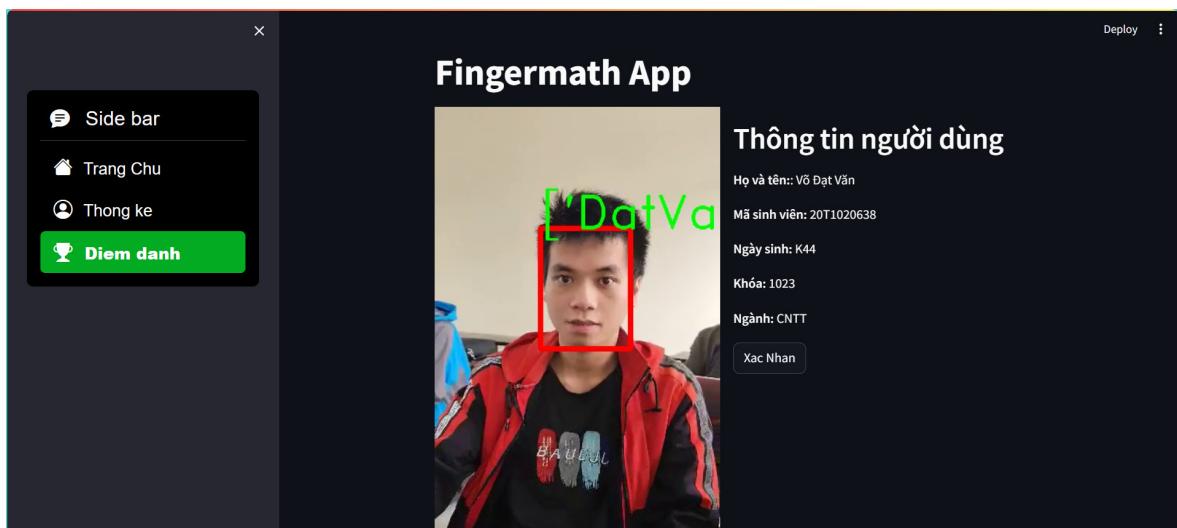
Hình 5.4: Danh sách sinh viên tham gia buổi học

STT	Mã SV	Họ	tên	Ngày sinh	Khóa	Ngành
8	20T1020455	Lê Công	Lương	2002-07-09	K44	1,023
18	20T1020638	Võ Đạt	Văn	2002-06-13	K44	1,023

Hình 5.5: Danh sách sinh viên vắng mặt trong buổi học

### 5.1.3 Giao diện điểm danh

Chức năng chính của Trang điểm danh cho phép sinh viên sử dụng camera hiện tại để điểm danh, xác nhận đã tham gia vào buổi học hiện tại.



Hình 5.6: Giao diện điểm danh

## Chương 6

# KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

### 6.1 Kết luận

Từ quá trình tìm hiểu các phương pháp MTCNN và FaceNet, chúng tôi nhận thấy rằng các phương pháp này đã mang lại những tiến bộ đáng kể trong việc nhận dạng khuôn mặt.

Qua quá trình nghiên cứu, chúng tôi đã tiếp cận, nắm được các nguyên lý cơ bản của mô hình MTCNN và mô hình FaceNet, cũng như cách chúng hoạt động và ứng dụng của chúng.

Trong phương pháp FaceNet, nó trực tiếp học cách ánh xạ từ hình ảnh khuôn mặt sang không gian Euclidean nhỏ gọn, trong đó khoảng cách tương ứng trực tiếp với thước đo độ giống nhau của khuôn mặt. Khi không gian này được tạo, nhận dạng khuôn mặt, xác thực và phân cụm có thể được thực hiện dễ dàng bằng cách sử dụng kỹ thuật nhúng FaceNet tiêu chuẩn làm vectơ đặc trưng. Cách tiếp cận này làm giảm đáng kể sự khác biệt giữa các cá nhân, đồng thời duy trì tính phân biệt đối xử giữa các cá nhân. Trong khi đó, luồng xử lý của MTCNN như sau: Trước hết, ảnh thử nghiệm được thay đổi kích thước liên tục để có được hình chéo hình ảnh. Sau đó, kim tự tháp hình ảnh được đưa vào P-Net để có được số lượng lớn ứng viên. Các hình ảnh ứng cử viên được P-Net sàng lọc được R-Net tinh chỉnh. Sau khi nhiều ứng cử viên bị R-Net loại bỏ, hình ảnh sẽ được đưa vào O-Net. Cuối cùng, tọa độ bbox chính xác sẽ được xuất ra.

So với DeepFace, FaceNet vẫn giữ nguyên tính năng căn chỉnh khuôn mặt, bỏ qua các bước trích xuất đặc trưng và trực tiếp sử dụng CNN để huấn luyện từ đầu đến

cuối sau khi căn chỉnh khuôn mặt. Hơn nữa, chúng tôi sử dụng phương pháp nhúng thống nhất để trực tiếp tìm hiểu cách nhúng vào không gian Euclidean để xác minh khuôn mặt. Điều này khiến phương pháp này trở nên khác biệt so với các phương pháp khác sử dụng lớp thắt cổ chai CNN hoặc yêu cầu xử lý hậu kỳ bổ sung chẳng hạn như ghép nối nhiều mô hình và PCA, cũng như phân loại SVM.

## 6.2 Hướng phát triển

Công việc trong tương lai sẽ tập trung vào việc hiểu rõ hơn về các trường hợp lỗi, cải thiện hơn nữa mô hình, đồng thời giảm kích thước mô hình và giảm yêu cầu CPU. Chúng tôi cũng sẽ xem xét các cách cải thiện thời gian đào tạo hiện tại rất dài.

Ngoài ra, đối với bài toán nhận dạng khuôn mặt, chúng tôi sẽ nghiên cứu các mô hình khác như DeepFace, ... với hi vọng sẽ có những kết quả tốt hơn.

# Tài liệu tham khảo

- [1] Y. Taigman; M. Yang; Ranzato; M. A. and L. Wolf. Deepface: Closing the gap to human-level performance in face verification. 2014.
- [2] Yang S.; Luo P.; Loy C. C.; and Tang X. From facial parts responses to face detection: A deep learning approach. 2016.
- [3] Pham M. T.; Gao Y.; Hoang V. D. D.; and Cham T. J. Fast polygonal integration and its application in extending haar-like features to improve object detection. 2010.
- [4] F. Schroff; D. Kalenichenko and J. Philbin. Facenet: A unified embedding for face recognition and clustering. 2015.
- [5] Rongrong Jin; Hao Li; Jing Pan; Wenxi Ma; and Jingyu Lin. Face recognition based on mtcnn and facenet. 2020.
- [6] Siyao Qi, Xinyu Zuo, Weijia Feng, and I G Naveen. Face recognition model based on mtcnn and facenet. In *2022 IEEE 2nd International Conference on Mobile Networks and Wireless Communications (ICMNWC)*, pages 1–5, 2022.
- [7] Florian Schroff, Dmitry Kalenichenko, and James Philbin. Facenet: A unified embedding for face recognition and clustering. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 815–823, 2015.
- [8] X.; Sun, Y.; Wang and X. Tang. Deep learning face representation by joint identification-verification. 2014.
- [9] C. Szegedy and e. al. "going deeper with convolutions. 2015.
- [10] C.-F. Tsai. Bag-of-words representation in image annotation: A review. 2012.
- [11] P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple

- features. In *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001*, volume 1, pages I–I, 2001.
- [12] K. Q. Weinberger and L. K. Saul. Distance metric learning for large margin nearest neighbor classification. 2009.
- [13] Yang B.; Yan J.; Lei Z.; and Li S. Z. Aggregate channel features for multi-view face detection. 2014.
- [14] J.-i. Imai Z. Li and M. Kaneko. Robust face recognition using block-based bag of words. 2010.
- [15] M. D. Zeiler and R. Fergus. Visualizing and understanding convolutional networks. 2014.
- [16] K. Zhang, Z. Zhang, Z. Li, and Y. Qiao. Joint face detection and alignment using multitask cascaded convolutional networks. *IEEE Signal Processing Letters*, 23(10):1499–1503, Oct 2016.