

**TRƯỜNG ĐẠI HỌC HỌC VĂN LANG
KHOA CÔNG NGHỆ THÔNG TIN**



**VAN LANG
UNIVERSITY**



**BÁO CÁO ĐỒ ÁN MÔN HỌC HK242
NHẬP MÔN PHÂN TÍCH DỮ LIỆU VÀ HỌC SÂU**

**XÂY DỰNG HỆ THỐNG NHẬN DẠNG KHUÔN
MẶT TRONG VIDEO VÀ HÌNH ẢNH**

Nhóm sinh viên thực hiện (Họ tên - Mã SV):

1. NGUYỄN CÔNG HUY – 2274802010310 (Trưởng nhóm)

2. TRẦN VĂN QUÂN – 2274802010727

3. NGUYỄN TĂNG TIẾN – 2274802010889

GVHD: TS. HUỖNH THÁI HỌC

TP. Hồ Chí Minh – năm 2025

LỜI MỞ ĐẦU

Để hoàn thành đồ án này, em xin gửi lời cảm ơn đến các Quý Thầy cô Khoa Công Nghệ Thông Tin, Trường đại học Văn Lang đã tạo cơ hội cho em được học tập, rèn luyện và tích lũy kiến thức, kỹ năng để thực hiện đồ án.

Đặc biệt, em xin gửi lời cảm ơn đến Giảng viên hướng dẫn Huỳnh Thái Học đã tận tình chỉ dẫn, theo dõi và đưa ra những lời khuyên bổ ích giúp em giải quyết được các vấn đề gặp phải trong quá trình nghiên cứu và hoàn thành đề tài một cách tốt nhất.

Do kiến thức của bản thân còn hạn chế và thiếu kinh nghiệm thực tiễn nên nội dung đồ án khó tránh những thiếu sót. Em rất mong nhận sự góp ý, chỉ dạy thêm từ Quý Thầy cô.

Cuối cùng, em xin chúc Quý Thầy Cô luôn thật nhiều sức khỏe và đạt được nhiều thành công trong công việc.

Chúng em xin chân thành cảm ơn!

TP.Hồ Chí Minh, ngày tháng năm 2025

Bảng Phân Công:

STT	Công việc	Thành viên
1	Thu thập dữ liệu hình ảnh khuôn mặt	Nguyễn Tăng Tiến
2	Xây dựng mô hình tổng quát	Trần Văn Quân
3	Tiền xử lý dữ liệu	Nguyễn Tăng Tiến
4	Train model nhận dạng khuôn mặt	Nguyễn Công Huy
5	Soạn báo cáo chương I	Nguyễn Tăng Tiến
6	Soạn báo cáo chương II	Trần Văn Quân
7	Soạn báo cáo chương III & IV	Nguyễn Công Huy

Mục lục

CHƯƠNG 1. GIỚI THIỆU ĐỀ TÀI	7
1. Lý do chọn Chủ đề	7
2. Đối tượng, phạm vi tìm hiểu:	7
2.1. Lịch sử và phát triển	7
2.2. Phạm vi tìm hiểu	7
3. Phương pháp thực hiện	7
CHƯƠNG 2. CƠ SỞ LÝ THUYẾT	8
1. Tổng quan lý thuyết về học sâu	8
2. Quy trình nhận dạng một khuôn mặt	8
3. MTCNN (Multi-task Cascaded Convolutional Networks)	9
3.1. Khái niệm	9
3.2. Kiến trúc	9
4. FACENET	13
4.1. Khái niệm	13
4.2. Khái quát thuật toán	13
4.3. Ứng dụng:	13
5. Yolo (You Only Look Once)	14
5.1. Khái niệm	14
5.2. Ứng dụng của Yolo	14
6. SVM (Support Vector Machine)	14
6.1. Khái niệm	14
6.2. Ứng dụng	16
7. Phương pháp nghiên cứu	16
8. Quy trình xây dựng mô hình	17
9. Quy trình nhận dạng khuôn mặt	19
CHƯƠNG 3. KẾT QUẢ THỰC HIỆN ĐỒ ÁN	20

1. Kết quả thực nghiệm trên hình ảnh.....	20
2. Kết quả thực nghiệm trên videos	22
CHƯƠNG 4. TỔNG KẾT.....	25
1. Hạn chế	25
2. Kết luận	25

Mục lục hình ảnh

Hình 1: Quy trình nhận dạng khuôn mặt	8
Hình 2: Kiến trúc MTCNN.....	10
Hình 3: P-Net.....	11
Hình 4: Cách chọn box cho khuôn mặt	11
Hình 5: R-Net	12
Hình 6: O-Net	12
Hình 7: SVM	15
Hình 8: Quá trình training mô hình	17
Hình 9: Quy trình nhận dạng khuôn mặt	19
Hình 10: Độ chính xác.....	20
Hình 11: Code nhận diện hình ảnh	21
Hình 12: Kết quả trên hình ảnh	22
Hình 13: Thử nghiệm trên video	22
Hình 14: kết quả trên video	23
Hình 15: Kết quả dưới 40%	24

CHƯƠNG 1. GIỚI THIỆU ĐỀ TÀI

1. Lý do chọn Chủ đề

Nhận diện khuôn mặt là một trong những công nghệ quan trọng trong lĩnh vực trí tuệ nhân tạo (AI) và thị giác máy tính (Computer Vision), với nhiều ứng dụng thực tiễn trong an ninh, tài chính, giáo dục, y tế, thương mại điện tử, và nhiều ngành nghề khác.

Trong bối cảnh an ninh và bảo mật ngày càng được chú trọng, công nghệ này giúp xác thực danh tính, kiểm soát ra vào và giám sát an toàn hiệu quả. Ngoài ra, nhận diện khuôn mặt còn đóng vai trò quan trọng trong tự động hóa quy trình, như điểm danh sinh viên, xác thực giao dịch ngân hàng, và cá nhân hóa trải nghiệm người dùng trong thương mại.

Với sự phát triển mạnh mẽ của học sâu (Deep Learning), các mô hình như YOLO, MTCNN, FaceNet, SVM đã cải thiện đáng kể độ chính xác và tốc độ nhận diện, mở ra nhiều cơ hội nghiên cứu và ứng dụng. Do đó, việc nghiên cứu và triển khai hệ thống nhận diện khuôn mặt không chỉ giúp nâng cao bảo mật mà còn tối ưu hóa nhiều quy trình trong thực tế.

2. Đối tượng, phạm vi tìm hiểu:

2.1. Lịch sử và phát triển

Dạng khuôn mặt có một lịch sử phát triển lâu dài, bắt đầu từ những năm 1960 với các phương pháp đơn giản dựa trên đặc trưng hình học. Tuy nhiên, với sự xuất hiện của học sâu (deep learning), các mô hình như MTCNN, FaceNet và SVM đã được phát triển và trở thành tiêu chuẩn trong ngành, mang lại độ chính xác cao và khả năng xử lý dữ liệu lớn.

2.2. Phạm vi tìm hiểu

Báo cáo này tập trung vào việc sử dụng YOLO, MTCNN để phát hiện khuôn mặt trong video, FaceNet để trích xuất đặc trưng khuôn mặt, và SVM để phân loại khuôn mặt. Phạm vi nghiên cứu bao gồm các kỹ thuật tiền xử lý ảnh, cách thu thập và tiền xử lý dữ liệu, cũng như việc đánh giá hiệu suất của mô hình trên các tập dữ liệu đa dạng.

3. Phương pháp thực hiện

Phương pháp thực hiện bao gồm thu thập dữ liệu video từ nhiều nguồn, tiền xử lý ảnh để cải thiện chất lượng, sử dụng MTCNN và YOLO để phát hiện khuôn mặt, FaceNet để trích xuất đặc trưng, và SVM để phân loại khuôn mặt. Hiệu suất của mô hình được đánh giá bằng các chỉ số như độ chính xác, độ nhớ và tốc độ nhận diện.

CHƯƠNG 2. CƠ SỞ LÝ THUYẾT

1. Tổng quan lý thuyết về học sâu

Học sâu (deep learning) là một tập hợp con của học máy (machine learning), tập trung vào việc xây dựng và huấn luyện mạng nơ-ron nhiều lớp, được gọi là mạng nơ-ron sâu (DNN – Deep neural networks) để chúng có thể tự động học, hiểu dữ liệu, mô phỏng khả năng ra quyết định phức tạp của bộ não con người.

Mô hình học sâu có thể nhận diện nhiều hình mẫu phức tạp trong hình ảnh, văn bản, âm thanh và các dữ liệu khác để tạo ra thông tin chuyên sâu và dự đoán chính xác. Bạn có thể sử dụng các phương pháp học sâu để tự động hóa các tác vụ thường đòi hỏi trí tuệ con người, chẳng hạn như phân loại hình ảnh hoặc chép lời một tập tin âm thanh.

2. Quy trình nhận dạng một khuôn mặt

Nhận dạng khuôn mặt (Face Recognition): là quá trình so sánh vector đặc trưng của khuôn mặt cần nhận diện với các vector đặc trưng trong cơ sở dữ liệu để xác định danh tính. Mục tiêu chính là tìm ra người tương ứng trong cơ sở dữ liệu hoặc xác định rằng khuôn mặt đó không thuộc bất kỳ ai đã được đăng ký. Các phương pháp phổ biến bao gồm so sánh khoảng cách, trong đó khoảng cách Euclidean hoặc cosine được tính toán giữa các vector đặc trưng để đo lường sự tương đồng; và phân loại, sử dụng các thuật toán như SVM hoặc k-NN để gán nhãn danh tính dựa trên vector đặc trưng. Kết quả cuối cùng là danh tính của người được nhận diện, hoặc thông báo “*không nhận diện được*” nếu không tìm thấy sự trùng khớp.

Quy trình nhận dạng một khuôn mặt (Face Recognition) có thể được chia thành nhiều bước như hình bên dưới:



Hình 1: Quy trình nhận dạng khuôn mặt

Phát hiện khuôn mặt (Face Detection): là quá trình xác định vị trí và kích thước của khuôn mặt trong ảnh hoặc video. Mục tiêu chính là tìm ra các khuôn mặt và đánh dấu chúng bằng khung chữ nhật. Các thuật toán phổ biến được sử dụng bao gồm YOLO (You Only Look Once) với ưu điểm tốc độ nhanh, hiệu quả cho việc phát hiện theo thời gian thực;

MTCNN (Multi-task Cascaded Convolutional Networks) có độ chính xác cao, đặc biệt trong các điều kiện khó khăn và Haar Cascades, một thuật toán cổ điển, nhanh nhưng độ chính xác hạn chế. Kết quả đầu ra của quá trình này là tọa độ và kích thước của khung chữ nhật bao quanh khuôn mặt được phát hiện.

Trích xuất đặc trưng (*Feature Extraction*): là quá trình hệ thống sẽ nhận diện và trích xuất các đặc điểm đặc trưng của hình ảnh như các cạnh, góc, kết cấu, hình dạng hoặc phân bố màu sắc. Việc này giúp đơn giản hóa dữ liệu, giữ lại các thông tin quan trọng và loại bỏ những chi tiết không cần thiết, từ đó tạo ra đầu vào hiệu quả cho các bước phân tích và phân loại sau.

Phân loại khuôn mặt (*Face Classification*): là quá trình xác định danh tính hoặc thuộc tính của một khuôn mặt đã được phát hiện. Đây là bước quan trọng trong hệ thống nhận dạng khuôn mặt, thường diễn ra sau khi khuôn mặt đã được phát hiện và xác định vị trí.

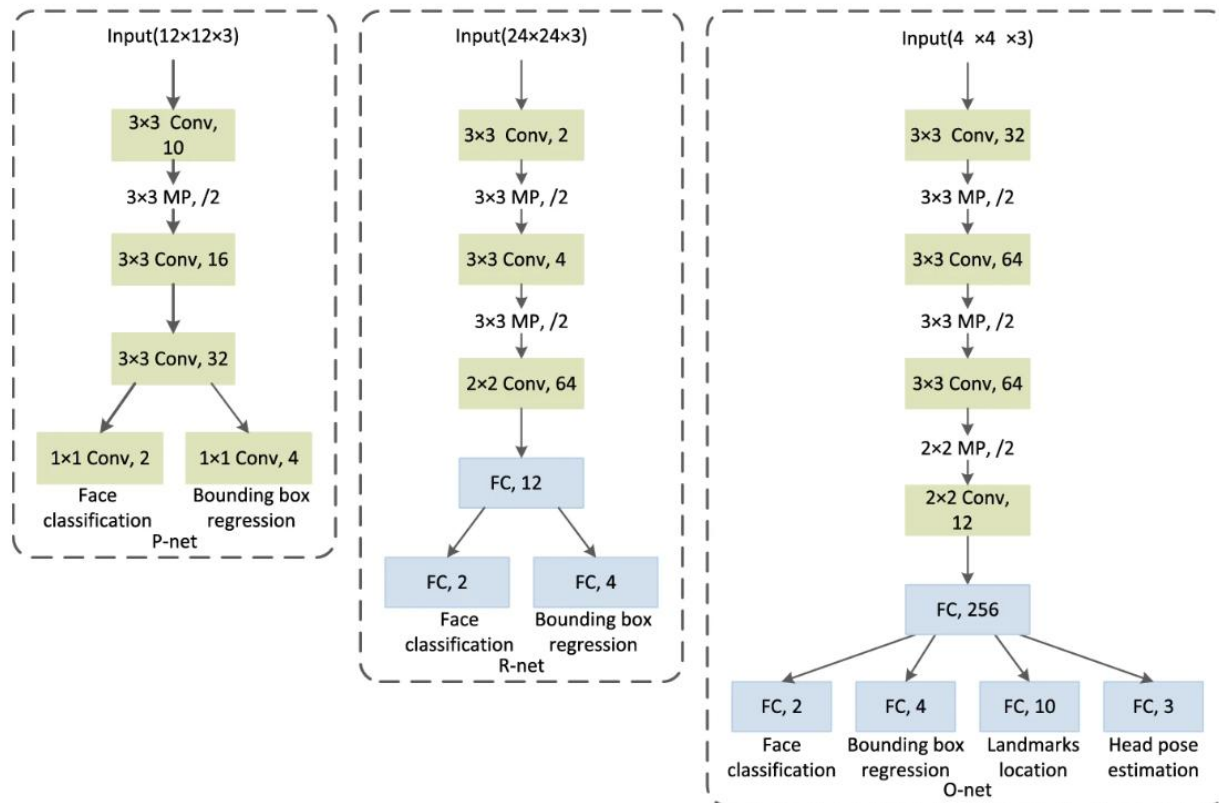
3. MTCNN (Multi-task Cascaded Convolutional Networks)

3.1. Khái niệm

MTCNN là một mô hình phức hợp bao gồm ba mạng nơ-ron tích chập, được thiết kế để phát hiện khuôn mặt và các điểm chính trên khuôn mặt. MTCNN hoạt động theo ba bước: phát hiện khuôn mặt, xác định các điểm chính, và thực hiện sửa đổi hình ảnh để cải thiện độ chính xác. Mô hình này nổi bật với khả năng phát hiện khuôn mặt trong các điều kiện ánh sáng khác nhau và với nhiều góc độ.

3.2. Kiến trúc

Kiến trúc của MTCNN gồm 3 mạng NN lay P - net, R - net và O – net.

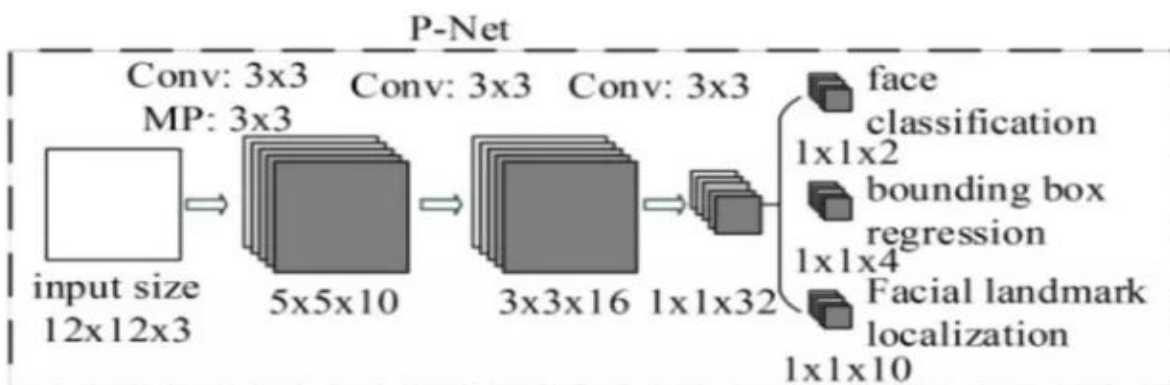


Hình 2: Kiến trúc MTCNN

The proposal network (P-Net)

Ở bước đầu tiên sẽ sử dụng mạng FCN (fully convolutional network). Mạng FCN khác mạng CNN ở chỗ mạng FCN không sử dụng lớp Dense layer. P-Net được sử dụng để có được các windows tiềm năng và bounding box regression vectors của chúng (tọa độ).

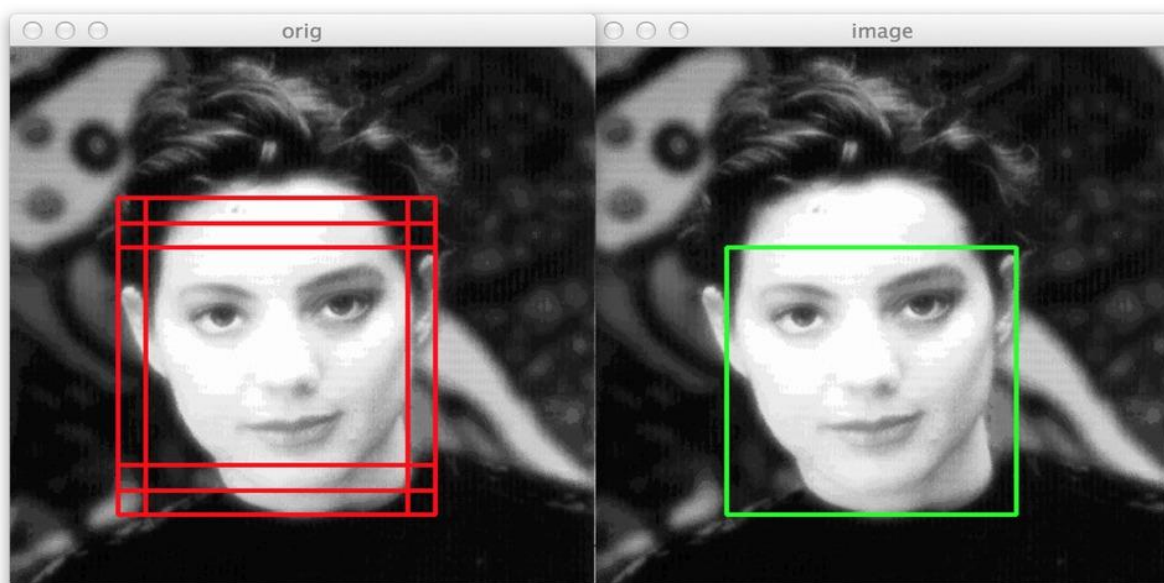
box regression là kỹ thuật để dự đoán vị trí của bounding box khi chúng ta cần phát hiện đối tượng (ở đây là khuôn mặt). Sau khi có được tọa độ của bounding boxes một vài tính chỉnh được thực hiện để loại bỏ một số bounding boxes overlap với nhau. Đầu ra của bước này là tất cả bounding boxes sau khi đã thực hiện sàng lọc.



P-Net (from MTCNN paper)

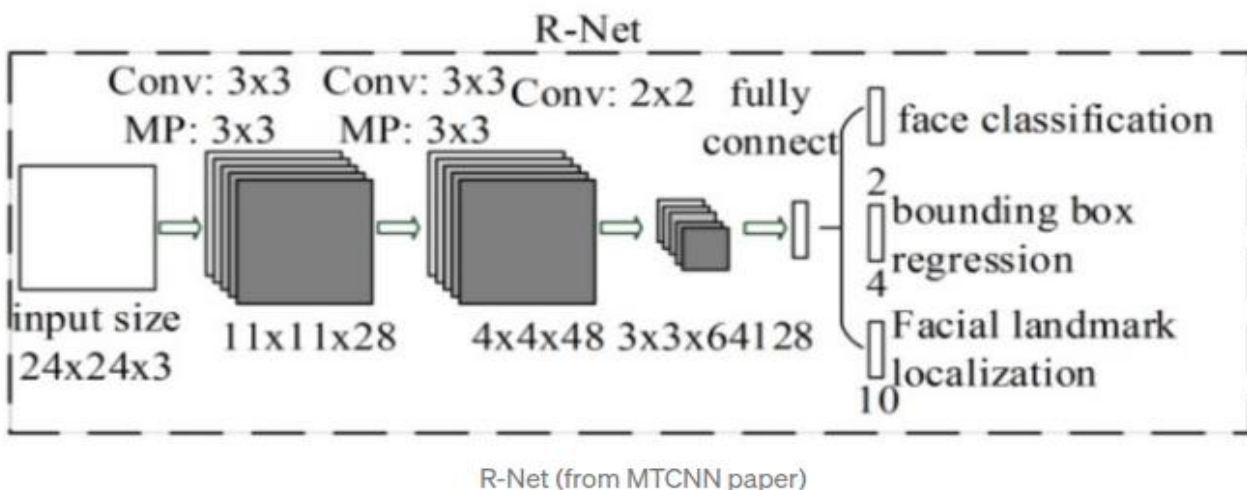
Hình 3: P-Net

NMS (Non-Maximum Suppression) để xóa các box có tỷ lệ trùng nhau (Intersection Over Union) vượt qua 1 mức threshold tự đặt nào đó. Hình ảnh dưới đây là minh họa cho phép NMS, những box bị trùng nhau sẽ bị loại bỏ và giữ lại 1 box có mức confident cao nhất.



Hình 4: Cách chọn box cho khuôn mặt

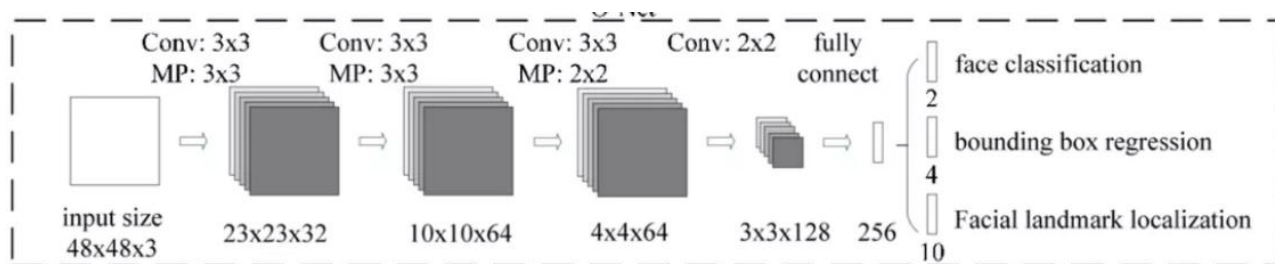
The Refine Network (R-Net)



Hình 5: R-Net

Mạng R (Refine Network) thực hiện các bước như mạng P. Tuy nhiên, mạng còn 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 24x24, được coi như 1 kernel và feed vào mạng R. Kết quả sau cũng là những tọa độ mới của các box còn lại và được đưa vào mạng tiếp theo, mạng O.

The Output Network (O-Net)



Hình 6: O-Net

Cuối cùng là mạng O (Output Network), mạng cũng thực hiện tương tự như việc trong mạng R, thay đổi kích thước thành 48x48. 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 (out[0]), tọa độ 5 điểm landmark trên mặt, bao gồm 2 mắt, 1 mũi, 2 bên cánh môi (out[1]) và điểm confident của mỗi box (out[2]). Tất cả sẽ được lưu vào thành 1 dictionary với 3 keys kể trên.

4. FACENET

4.1. Khái niệm

FaceNet là một mô hình học sâu được phát triển bởi Google, nhằm trích xuất đặc trưng khuôn mặt. FaceNet sử dụng một phương pháp được gọi là "triplet loss" để tối ưu hóa khoảng cách giữa các đặc trưng của khuôn mặt giống nhau và tăng khoảng cách giữa các khuôn mặt khác nhau. Kết quả là một vector đặc trưng có thể được sử dụng cho các nhiệm vụ nhận diện và xác thực.

4.2 Khái quát thuật toán

Hầu hết các thuật toán nhận diện khuôn mặt trước facenet đều tìm cách biểu diễn khuôn mặt bằng một véc tơ embedding thông qua một layer bottle neck có tác dụng giảm chiều dữ liệu.

Tuy nhiên hạn chế của các thuật toán này đó là số lượng chiều embedding tương đối lớn (thường ≥ 1000) và ảnh hưởng tới tốc độ của thuật toán. Thường chúng ta phải áp dụng thêm thuật toán PCA để giảm chiều dữ liệu để tăng tốc độ tính toán.

Hàm loss function chỉ đo lường khoảng cách giữa 2 bức ảnh. Như vậy trong một đầu vào huấn luyện chỉ học được một trong hai khả năng là sự giống nhau nếu chúng cùng 1 class hoặc sự khác nhau nếu chúng khác class mà không học được cùng lúc sự giống nhau và khác nhau trên cùng một lượt huấn luyện.

Facenet đã giải quyết cả 2 vấn đề trên bằng các hiệu chỉnh nhỏ nhưng mang lại hiệu quả lớn:

Base network áp dụng một mạng convolutional neural network và giảm chiều dữ liệu xuống chỉ còn 128 chiều. Do đó quá trình suy diễn và dự báo nhanh hơn và đồng thời độ chính xác vẫn được đảm bảo

Sử dụng loss function là hàm triplet loss có khả năng học được đồng thời sự giống nhau giữa 2 bức ảnh cùng nhóm và phân biệt các bức ảnh không cùng nhóm. Do đó hiệu quả hơn rất nhiều so với các phương pháp trước đây.

4.3. Ứng dụng:

Nhận diện khuôn mặt: FaceNet có thể xác định danh tính của một người bằng cách so sánh khuôn mặt của họ với cơ sở dữ liệu khuôn mặt đã biết.

Xác thực sinh trắc học: Sử dụng FaceNet để xác minh danh tính người dùng trong các hệ thống bảo mật, như mở khóa điện thoại hoặc truy cập vào các ứng dụng nhạy cảm.

Phân loại khuôn mặt: FaceNet có thể phân loại khuôn mặt theo các nhóm khác nhau, chẳng hạn như theo độ tuổi, giới tính hoặc cảm xúc.

Theo dõi khuôn mặt: Trong video hoặc ứng dụng trực tiếp, FaceNet có thể theo dõi chuyển động của khuôn mặt theo thời gian.

Tìm kiếm khuôn mặt: FaceNet cho phép tìm kiếm và nhận diện khuôn mặt trong một Gợi ý bạn bè trên mạng xã hội: FaceNet có thể được sử dụng để gợi ý người dùng kết bạn dựa trên sự tương đồng về khuôn mặt.

Ngoài ra FaceNet có thể được ứng dụng trong các lĩnh vực như kiểm tra chất lượng sản phẩm trong các nhà máy, truy tìm người thân, tìm kiếm tội phạm.

5. Yolo (You Only Look Once)

5.1. Khái niệm

YOLO là một trong những mô hình phát hiện đối tượng nổi bật, cho phép phát hiện khuôn mặt một cách nhanh chóng và hiệu quả. Khác với các phương pháp truyền thống, YOLO sử dụng một mạng nơ-ron tích chập (CNN) để thực hiện phát hiện đối tượng trong một lần duy nhất. Điều này giúp giảm thời gian xử lý và tăng tốc độ nhận diện, đặc biệt trong các ứng dụng video thời gian thực.

5.2. Ứng dụng của Yolo

Phát hiện vị trí khuôn mặt trong ảnh và video.

Đặc biệt hiệu quả trong việc xử lý video với tốc độ khung hình cao.

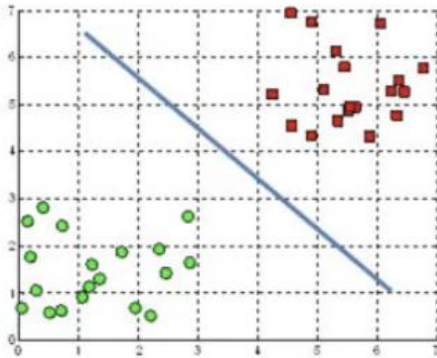
6. SVM (Support Vector Machine)

6.1. Khái niệm

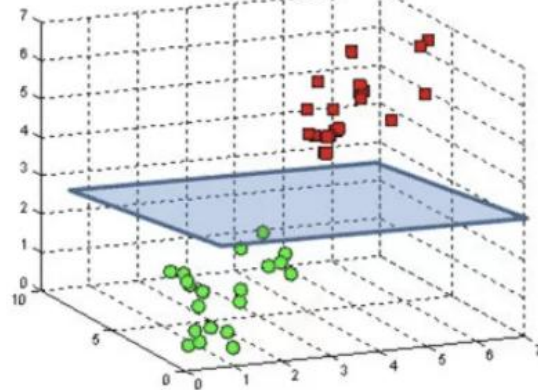
Support Vector Machines (SVM) là một thuật toán học máy phổ biến, được sử dụng để phân loại dữ liệu bằng cách tìm một siêu phẳng phân cách trong không gian N chiều.

Mục tiêu của SVM là tìm ra một siêu phẳng trong không gian N chiều (ứng với N đặc trưng) chia dữ liệu thành hai phần tương ứng với lớp của chúng. Nói theo ngôn ngữ của đại số tuyến tính, siêu phẳng này phải có lề cực đại và phân chia hai bao lồi và cách đều chúng.

A hyperplane in \mathbb{R}^2 is a line



A hyperplane in \mathbb{R}^3 is a plane



Hình 7: SVM

6.2. Minh họa thuật toán SVM One-vs-Rest (OvR)

Phương pháp OvR huấn luyện một bộ phân loại SVM cho từng lớp, phân biệt lớp đó với tất cả lớp còn lại.

Mỗi mô hình SVM tạo ra một **decision score** cho lớp theo công thức:

$$F_i(x) = w_i \cdot x + b_i$$

Trong đó:

w_i là vector trọng số của lớp .

x là vector đặc trưng của mẫu cần phân loại.

b_i là hệ số bias của mô hình.

Lớp có decision score lớn nhất sẽ được chọn làm kết quả dự đoán.

Cách tính decision score:

- Với mỗi lớp i , tính giá trị $F_i(x)$ theo công thức trên.
- Giá trị này thể hiện khoảng cách của điểm đến siêu phẳng phân chia lớp đó với phần còn lại.
- Lớp có $F_i(x)$ giá trị lớn nhất sẽ là lớp dự đoán cuối cùng.

Ví dụ: Giả sử mô hình có 3 lớp (A, B, C), đầu ra của `decision_function(X_test)` là:

$$F_A(x) = -0.5, F_B(x) = 1.3, F_C(x) = 0.2$$

Vì là lớn nhất, nên mẫu thuộc lớp **B**.

6.3. Ứng dụng

Y tế:

- Chẩn đoán bệnh: SVM được sử dụng để phân tích dữ liệu hình ảnh y tế (như X-quang, MRI) để chẩn đoán các bệnh như ung thư, bệnh tim mạch, và các bệnh khác.
- Dự đoán tiên lượng bệnh: SVM giúp dự đoán tiên lượng bệnh dựa trên dữ liệu lâm sàng và các yếu tố nguy cơ.

Nhận dạng hình ảnh và video:

- Nhận dạng khuôn mặt: SVM được sử dụng trong các hệ thống nhận dạng khuôn mặt để xác định danh tính của người trong hình ảnh hoặc video.
- Phân loại hình ảnh: SVM giúp phân loại hình ảnh thành các danh mục khác nhau, như trong các ứng dụng phân loại ảnh trên mạng xã hội.

Xử lý ngôn ngữ tự nhiên (NLP):

- Phân loại văn bản: SVM được sử dụng để phân loại văn bản thành các danh mục như tin tức, email thư rác, hoặc các loại văn bản khác.
- Phân tích cảm xúc: SVM giúp phân tích cảm xúc trong văn bản, như trong các đánh giá sản phẩm hoặc bình luận trên mạng xã hội.

7. Phương pháp nghiên cứu

Phương pháp nghiên cứu trong đề tài này bao gồm các bước sau:

- **Thu Thập Dữ Liệu:**

Nguồn Dữ Liệu Dùng bộ dữ liệu Labelled Faces in the Wild (LFW) gồm 13,233 ảnh của 5,749 người trên web.

- **Tiền Xử Lý Dữ Liệu:**

Chọn 100 người ngẫu nhiên trong tập dataset với số lượng ảnh mỗi người lớn hơn hoặc bằng 5 ảnh.

- **Phát Hiện Khuôn Mặt:**

Sử Dụng MTCNN để detect khuôn mặt và khoanh vùng xuất hiện khuôn mặt trong ảnh và cắt ảnh được khoanh vùng.

- **Trích Xuất Đặc Trưng Khuôn Mặt:**

Bức ảnh cắt sẽ Sử Dụng FaceNet để trích xuất đặc trưng của người trong ảnh và lưu lại thành một vector với 512 chiều.

- **Phân Loại Khuôn Mặt:**

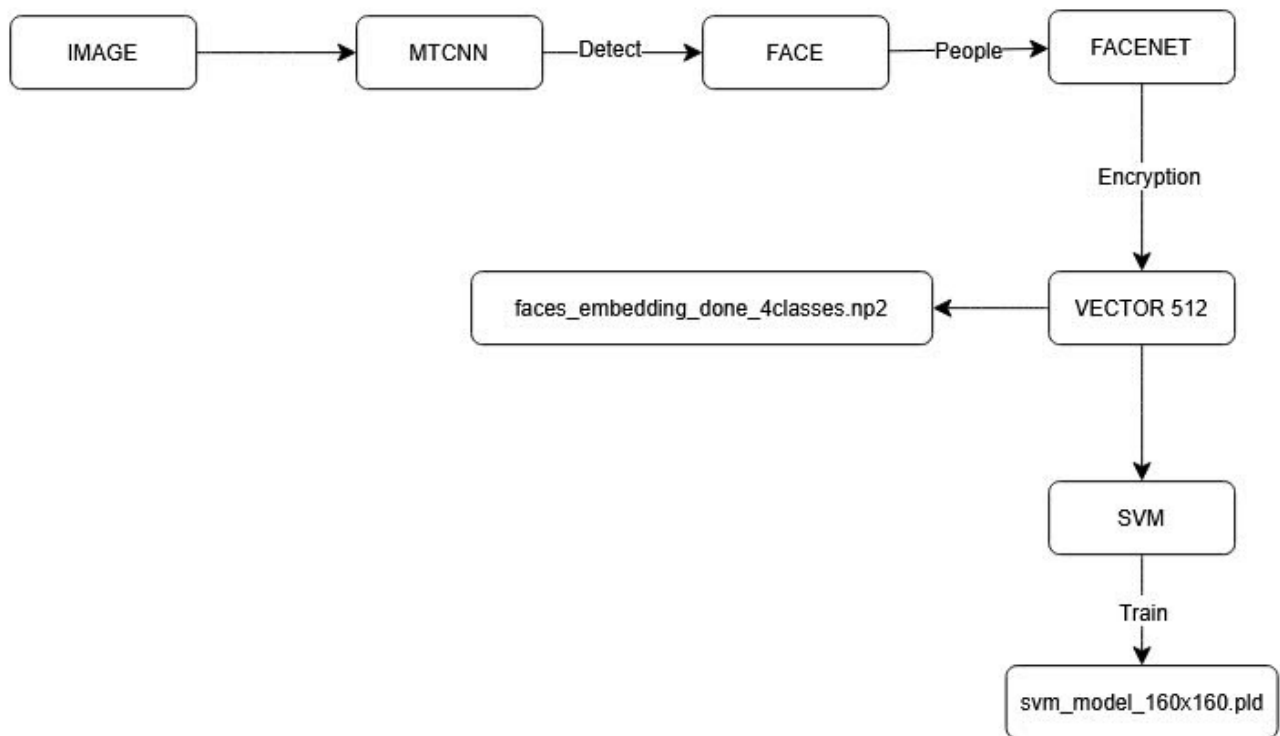
Các vector sẽ đi qua SVM để phân loại ra từng người.

- **Đánh Giá Hiệu Suất:**

Sử dụng các chỉ số như độ chính xác để đánh giá mô hình.

Sử dụng các tập dữ liệu test 20% để đánh giá.

8. Quy trình xây dựng mô hình



Hình 8: Quá trình training mô hình

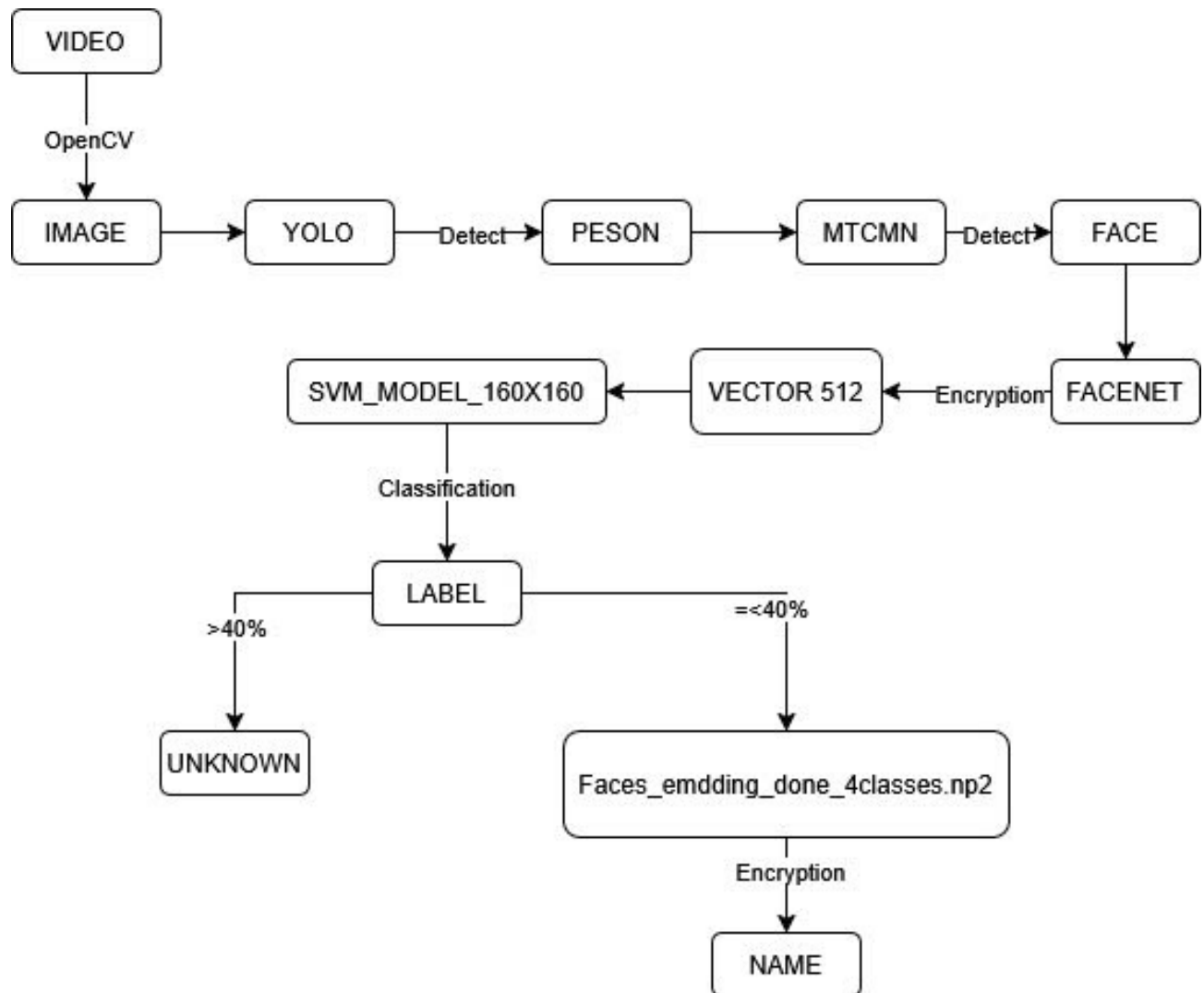
Sau khi các ảnh đã được xử lý bởi mô hình MTCNN để phát hiện khuôn mặt và được đánh dấu bằng các bounding box, các khuôn mặt được phân tích bởi mạng neuron

Facenet để tạo ra các vector embedding 512 chiều. Các vector này được lưu trữ trong tệp **faces_embedding_done_4classes.npz**, chứa thông tin mã hóa của khuôn mặt.

Các vector embedding được sử dụng làm dữ liệu đầu vào cho thuật toán Support Vector Machine (SVM), một phương pháp phân loại được sử dụng rộng rãi trong các nhiệm vụ nhận diện và phân loại. SVM được huấn luyện để tìm ra đường phân cách tối ưu giữa các lớp khuôn mặt khác nhau, từ đó xây dựng mô hình nhận diện khuôn mặt.

Quá trình huấn luyện SVM kết thúc với việc tạo ra một mô hình đã được đào tạo, được lưu trữ trong tệp **svm_model_160x160.pld**. Mô hình này có khả năng nhận diện và phân loại khuôn mặt trong các ảnh mới bằng cách so sánh khoảng cách với đường ranh các vector embedding của khuôn mặt mới với đường ranh của các vector đã được huấn luyện trong mô hình SVM.

9. Quy trình nhận dạng khuôn mặt



Hình 9: Quy trình nhận dạng khuôn mặt

Quy trình nhận dạng khuôn mặt sẽ bao gồm các bước chính như sau:

- **Thu thập Dữ Liệu:**

Dữ liệu đầu vào được lấy từ video, được xử lý bằng thư viện OpenCV để đọc và chuyển đổi thành các khung hình.

- **Tiền Xử Lý:**

Các khung hình được phân tích bằng thuật toán YOLO để phát hiện sự xuất hiện của con người trong từng khung hình.

- **Phát Hiện Khuôn Mặt:**

Mô hình MTCNN được áp dụng sau đó để phát hiện và định vị chính xác khuôn mặt trong các khung hình đã xác định sự hiện diện của con người.

- **Trích Xuất Đặc Trưng Khuôn Mặt:**

Mạng neuron FaceNet được sử dụng để tạo ra các vector đặc trưng 512 chiều cho mỗi khuôn mặt đã được phát hiện.

- **Phân Loại Khuôn Mặt:**

Mô hình SVM đã được huấn luyện trước (**SVM_MODEL_160X160**) được sử dụng để phân loại khuôn mặt dựa trên các vector đặc trưng đã trích xuất.

Sau khi đi qua SVM nó sẽ tính toán xem khoảng cách đó với vector đó với các SVM hay là decision score sau đó sẽ đi qua hàm sigmoid biến thành xác suất từ 0 đến 1 và SVM sẽ lấy lable của class có xác suất lớn nhất làm lable của khuôn mặt đó

Nếu độ chính xác của phân loại vượt quá 40%, kết quả sẽ được gán nhãn với tên của cá nhân đó. Ngược lại, nếu độ chính xác không vượt quá 40%, kết quả sẽ được gán nhãn là "UNKNOWN".

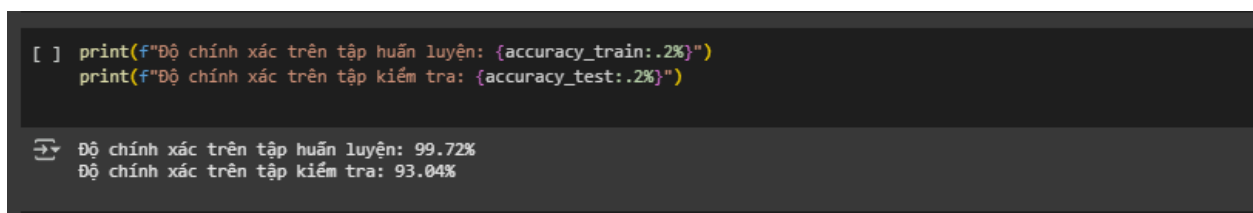
- **Kết Quả:**

Hiển thị tên của cá nhân được nhận diện cùng với độ chính xác của phân loại.

Nếu độ chính xác không vượt quá 40%, hiển thị "UNKNOWN".

CHƯƠNG 3. KẾT QUẢ THỰC HIỆN ĐỒ ÁN

1. Kết quả thực nghiệm trên hình ảnh



```
[ ] print(f"Độ chính xác trên tập huấn luyện: {accuracy_train:.2%}")
    print(f"Độ chính xác trên tập kiểm tra: {accuracy_test:.2%}")

Độ chính xác trên tập huấn luyện: 99.72%
Độ chính xác trên tập kiểm tra: 93.04%
```

Hình 10: Độ chính xác

Độ chính xác trên tập test là 93% và train là 99%

```
[ ] # Load model SVM
model_path = "/content/svm_model_160x160.pkl"
svm_model = pickle.load(open(model_path, 'rb'))

# Load embeddings & encoder
data_path = "/content/faces_embeddings_done_4classes.npz"
faces_embeddings = np.load(data_path)
Y = faces_embeddings['arr_1']

encoder = LabelEncoder()
encoder.fit(Y)

# Khởi tạo MTCNN và FaceNet
detector = MTCNN()
facenet = FaceNet()
```

```
# Load ảnh kiểm tra
image_path = "/content/Fiona Gallagher Icon.jpg" # Thay bằng đường dẫn ảnh của bạn
t_im = cv.imread(image_path)
t_im = cv.cvtColor(t_im, cv.COLOR_BGR2RGB)

# Phát hiện khuôn mặt
faces = detector.detect_faces(t_im)

if len(faces) == 0:
    print("Không tìm thấy khuôn mặt nào trong ảnh.")
else:
    x, y, w, h = faces[0]['box']
    x, y = max(0, x), max(0, y) # Đảm bảo tọa độ không âm

    # Cắt và resize khuôn mặt
    face_img = t_im[y:y+h, x:x+w]
    face_img = cv.resize(face_img, (160, 160))
    face_img = np.expand_dims(face_img, axis=0)

    # Trích xuất đặc trưng
    test_embedding = facenet.embeddings(face_img)

    # Dự đoán bằng SVM
    ypreds = svm_model.predict(test_embedding)
    predicted_name = encoder.inverse_transform(ypreds)[0]

    # Tính độ chính xác
    proba = svm_model.predict_proba(test_embedding)
    confidence = max(proba[0])

    print(f"Dự đoán: {predicted_name} (Độ chính xác: {confidence:.2f})")
    if confidence < 0.5:
        print("=> Kết quả không chắc chắn, có thể là Unknown.")
```

Hình 11: Code nhận diện hình ảnh

Code trên dùng để nhận dạng người trong bức ảnh, chỉ cần thay image_path thành ảnh của người đã train thì sẽ chạy ra kết quả là tên và độ chính xác

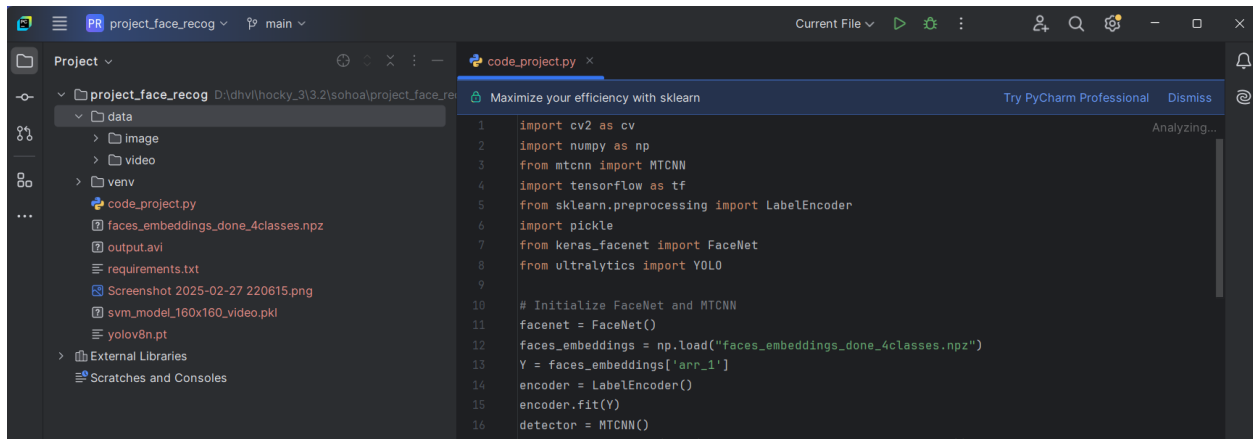
```
1/1 ————— 6s 6s/step
Dự đoán: Emmy_Rossum (Độ chính xác: 0.06)
=> Kết quả không chắc chắn, có thể là Unknown.
```

Hình 12: Kết quả trên hình ảnh

2. Kết quả thực nghiệm trên videos

Code trên báo cáo chạy trên pycharm để chạy được cần cài đặt pycharm.

Sau khi cài đặt mở thư mục đã giải nén tên project_face_recog.



Hình 13: Thử nghiệm trên video

Đây là giao diện khi mở thư mục bằng pycharm trong thư mục project có thư mục video là nơi để lưu video để nhận sau khi lưu vào thay video_path thành địa chỉ video bạn muốn nhận dạng.

Để chạy được code_project.py cần cài các thư viện đã được gán sẵn vào tệp requirement.txt

Chạy 2 lệnh sau:

Mở môi trường ảo: `venv\Scripts\activate`

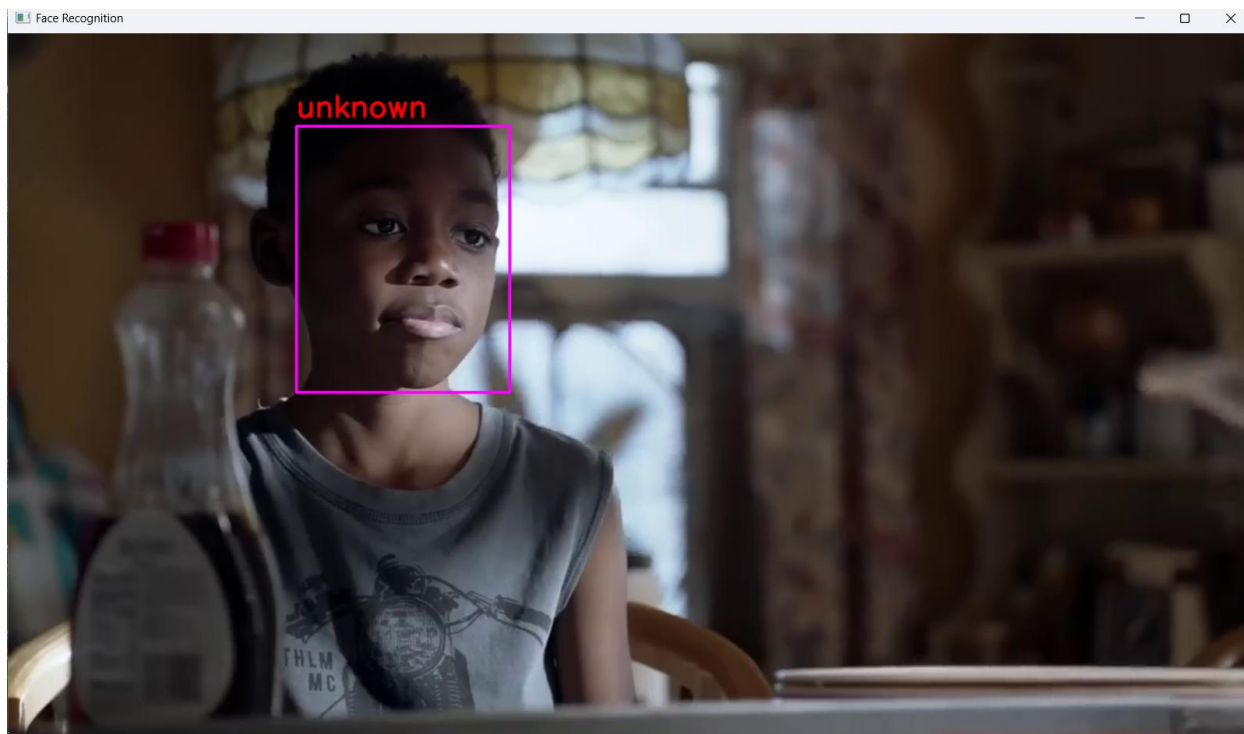
Cài thư viện: `pip install -r requirements.txt`

Sau đó chạy `python code_project.py`

Dưới đây là hình ảnh minh họa quá trình nhận diện trong video:



Hình 14: kết quả trên video



Hình 15: Kết quả dưới 40%

Nếu độ chính xác $> 40\%$ thì sẽ hiện tên và độ chính xác nếu nhỏ hơn thì hiện unknown.

CHƯƠNG 4. TỔNG KẾT

1. Hạn chế

Mặc dù hệ thống nhận diện khuôn mặt được xây dựng đã đạt được kết quả nhất định, nhưng vẫn tồn tại một số hạn chế cần khắc phục:

- Độ chính xác: Mô hình có thể gặp khó khăn khi nhận diện khuôn mặt trong điều kiện ánh sáng yếu hoặc góc quay không thuận lợi. Việc cải thiện dữ liệu huấn luyện có thể giúp tăng độ chính xác.
- Hiệu suất thời gian thực: Việc sử dụng YOLO, MTCNN và FaceNet mang lại hiệu quả cao nhưng vẫn có độ trễ khi xử lý các video có độ phân giải cao. Tối ưu hóa thuật toán hoặc sử dụng phần cứng mạnh hơn có thể cải thiện tốc độ.
- Tính đa dạng của dữ liệu: Hệ thống chưa được thử nghiệm trên nhiều tập dữ liệu với sự đa dạng về chủng tộc, độ tuổi, và giới tính. Điều này có thể ảnh hưởng đến tính tổng quát của mô hình.
- Khả năng chống giả mạo: Hệ thống chưa được tích hợp các phương pháp chống giả mạo khuôn mặt (Face Anti-Spoofing), khiến nó dễ bị lừa bởi hình ảnh hoặc video giả mạo.

2. Kết luận

Nhận diện khuôn mặt là một lĩnh vực quan trọng trong trí tuệ nhân tạo và thị giác máy tính, với nhiều ứng dụng thực tiễn trong bảo mật, giáo dục và thương mại. Đề tài đã tập trung nghiên cứu và triển khai một hệ thống nhận diện khuôn mặt sử dụng YOLO, MTCNN, FaceNet và SVM, giúp tối ưu hóa độ chính xác và hiệu suất nhận diện.

Kết quả thu được cho thấy hệ thống hoạt động khá hiệu quả trong các điều kiện tiêu chuẩn, nhưng vẫn cần cải thiện để nâng cao khả năng nhận diện trong môi trường thực tế. Trong tương lai, việc mở rộng tập dữ liệu huấn luyện, tối ưu thuật toán và tích hợp các phương pháp chống giả mạo sẽ là những hướng phát triển quan trọng để hoàn thiện hệ thống.

Cuối cùng, đề tài này là một bước tiến trong việc ứng dụng công nghệ nhận diện khuôn mặt vào thực tế, mở ra nhiều cơ hội phát triển và ứng dụng trong các lĩnh vực khác nhau.

DANH MỤC THAM KHẢO

<https://phamdinhhkhanh.github.io/2020/03/12/faceNetAlgorithm.html>

<https://huytranvan2010.github.io/FaceNet/>

<https://medium.com/@iselagradilla94/multi-task-cascaded-convolutional-networks-mtcnn-for-face-detection-and-facial-landmark-alignment-7c21e8007923>