



TRƯỜNG ĐẠI HỌC BÁCH KHOA
KHOA CÔNG NGHỆ THÔNG TIN



BÁO CÁO
PBL5 - ĐỒ ÁN KỸ THUẬT MÁY TÍNH

HỆ THỐNG NHẬN DẠNG NGƯỜI ĐEO
KHẨU TRANG & KHUÔN MẶT

Giảng viên hướng dẫn: TS. Ninh Khánh Duy

STT NHÓM: 12	LỚP HỌC
HỌ VÀ TÊN SINH VIÊN	PHẦN ĐỒ ÁN
Nguyễn Hưng	20N11A
Bùi Phước Huy	20N11A
Nguyễn Hữu Huy	20N11A
Trương Hoàng Nhật Huy	20N11A

ĐÀ NẴNG, 06/2023

TÓM TẮT ĐỀ ÁN

Đề tài "Hệ thống phát hiện người đeo khẩu trang và nhận diện người không đeo khẩu trang" tập trung vào việc giải quyết vấn đề của việc kiểm soát và quản lý việc đeo khẩu trang trong môi trường công cộng. Phương pháp giải quyết của đề tài là xây dựng một hệ thống sử dụng công nghệ học máy và trí tuệ nhân tạo để phân loại và phát hiện người đeo khẩu trang và nhận diện người không đeo khẩu trang. Kết quả đạt được dự kiến là một hệ thống có khả năng phát hiện và nhận diện chính xác người đeo và không đeo khẩu trang, từ đó hỗ trợ trong việc giám sát và quản lý tuân thủ các biện pháp đeo khẩu trang và đảm bảo an toàn cộng đồng.

BẢNG PHÂN CÔNG NHIỆM VỤ

Sinh viên thực hiện	Nhiệm vụ	Tự đánh giá
Trương Hoàng Nhật Huy	<ul style="list-style-type: none"> - Quản lý tiến độ - Làm việc với phần cứng - Viết báo cáo 	<ul style="list-style-type: none"> - Đã hoàn thành - Đã hoàn thành - Đã hoàn thành
Bùi Phước Huy	<ul style="list-style-type: none"> - Thiết kế server - Thiết kế web - Thiết kế database - Viết báo cáo 	<ul style="list-style-type: none"> - Đã hoàn thành - Đã hoàn thành - Đã hoàn thành - Đã hoàn thành
Nguyễn Hưng	<ul style="list-style-type: none"> - Tìm hiểu AI, học máy - Xây dựng mô hình AI - Viết ứng dụng - Viết báo cáo 	<ul style="list-style-type: none"> - Đã hoàn thành - Đã hoàn thành - Đã hoàn thành - Đã hoàn thành
Nguyễn Hữu Huy	<ul style="list-style-type: none"> - Tìm hiểu AI, học máy - Xây dựng mô hình AI - Viết báo cáo 	<ul style="list-style-type: none"> - Đã hoàn thành - Đã hoàn thành - Đã hoàn thành

MỤC LỤC

1. Giới thiệu đề tài	6
2. Giải pháp.....	6
2.1. Giải pháp về phần cứng và truyền thông	7
2.2. Giải pháp về truyền thông	8
2.3. Giải pháp về phần mềm.....	9
3. Quá trình triển khai.....	10
3.1. Thu thập dữ liệu.....	10
3.1.1. Bộ dữ liệu khẩu trang	10
3.1.2. Bộ dữ liệu khuôn mặt	11
3.2. Công cụ và thư viện.....	12
3.3. Điều kiện tiến hành thực nghiệm.....	13
3.4. Các độ đo (metrics) để đánh giá hiệu suất.....	13
3.5. Quy trình và điều kiện kiểm thử hệ thống.....	13
4. Thuật toán triển khai.....	14
4.1. Thuật toán phát hiện khuôn mặt và nhận diện khẩu trang.....	14
4.1.1. YOLO	15
4.1.2. SSD	15
4.2. So sánh các thuật toán trích xuất đặc trưng khuôn mặt.....	16
4.2.1. Facenet.....	16
4.2.2. LBPH.....	17
4.2.3. Eigenface	17
4.2.4. So sánh kết quả và đánh giá	18
4.3. Thuật toán nhận diện phân lớp khuôn mặt	18
4.3.1. Support Vector Machines (SVM).....	19
4.3.2. Euclidean Distance	20
4.3.3. So sánh kết quả.....	22
5. Kết quả đạt được.....	23
6. Kết luận.....	24
DANH MỤC TÀI LIỆU THAM KHẢO	26

DANH SÁCH HÌNH VẼ

Hình 1. Sơ đồ khối thể hiện tính kết nối của hệ thống	7
Hình 2. Sơ đồ thuật toán của hệ thống	9
Hình 3. Biểu đồ số lượng mẫu của từng nhãn trong bộ dữ liệu khẩu trang	11
Hình 4. Sơ đồ thuật toán phát hiện khuôn mặt và nhận diện khẩu trang	14
Hình 5. Quá trình huấn luyện YOLO	15
Hình 6. Quá trình huấn luyện SSD	15
Hình 7. Sơ đồ khối thuật toán nhận dạng khuôn mặt	16
Hình 8. Ví dụ về LBPH	17
Hình 9. Ví dụ về Eigenface	17
Hình 10. Sơ đồ khối nhận dạng khuôn mặt sử dụng phân lớp SVM	19
Hình 11. Ma trận nhầm lẫn của thuật toán SVM	20
Hình 12. Sơ đồ khối nhận dạng khuôn mặt sử dụng thuật toán Euclidean Distance ..	21
Hình 13. Ma trận nhầm lẫn của thuật toán Euclidean Distance	21
Hình 14. Ứng dụng chạy trên Jetson Nano	23
Hình 15. Web Server nhận luồng dữ liệu từ Jetson Nano	23
Hình 16. Web Client hiển thị kết quả nhận diện	24
Hình 17. Firebase lưu trữ dữ liệu vector khuôn mặt và kết quả nhận diện	24

DANH SÁCH BẢNG BIỂU

Bảng 1. Danh sách thiết bị phần cứng	8
Bảng 2. So sánh kết quả các thuật toán trích xuất đặc trưng khuôn mặt	18
Bảng 3. So sánh kết quả các thuật toán phân lớp	22

1. Giới thiệu đề tài

Trong bối cảnh đại dịch COVID-19 và các biện pháp phòng ngừa lây nhiễm, việc đeo khẩu trang đã trở thành một biện pháp quan trọng để bảo vệ sức khỏe cộng đồng. Tuy nhiên, việc theo dõi và đảm bảo tuân thủ việc đeo khẩu trang đối với mọi người vẫn là một thách thức. Đề tài này tập trung vào nghiên cứu và phát triển hệ thống phát hiện người đeo khẩu trang và nhận diện người không đeo khẩu trang để hỗ trợ việc kiểm soát và quản lý dịch bệnh.

Mục tiêu chính của đề tài là xây dựng một hệ thống phát hiện người đeo khẩu trang và nhận diện người không đeo khẩu trang. Hệ thống này sẽ có khả năng phân loại và phát hiện người đeo khẩu trang đúng và nhận diện người không đeo khẩu trang một cách chính xác và hiệu quả, từ đó hỗ trợ việc giám sát và quản lý việc tuân thủ việc đeo khẩu trang trong các khu vực công cộng, như sân bay, bệnh viện, trường học, và các nơi công cộng khác.

Kết quả nghiên cứu của đề tài này dự kiến sẽ cung cấp một hệ thống phát hiện người đeo khẩu trang và nhận diện người không đeo khẩu trang có khả năng phân loại và nhận dạng chính xác. Hệ thống này có thể hỗ trợ việc giám sát và quản lý tuân thủ việc đeo khẩu trang trong các khu vực công cộng, đóng góp vào công cuộc phòng chống dịch bệnh và quản lý an toàn cộng đồng.

Hệ thống sẽ được áp dụng tại những nơi yêu cầu việc đeo khẩu trang để ra vào:

- Sân bay hoặc cửa khẩu
- Các cơ sở y tế
- Công ty hoặc văn phòng

2. Giải pháp

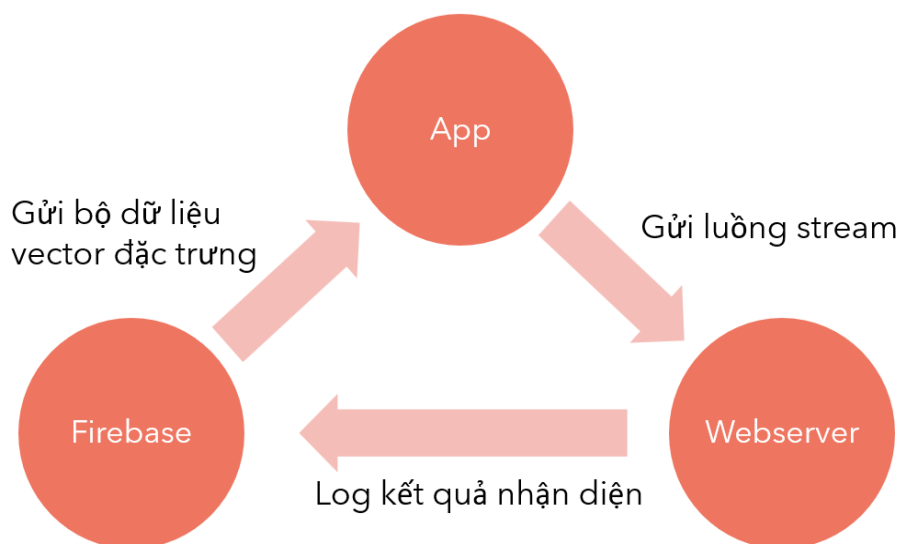
Hệ thống phát hiện người đeo khẩu trang và nhận diện người không đeo khẩu trang là một ứng dụng AI được phát triển để giúp xác định xem một người có đang đeo khẩu trang hay không thông qua dữ liệu được lấy trực tiếp từ camera. Dưới đây là phân tích các chức năng của hệ thống và các giải pháp để thực hiện chúng:

- Phát hiện người trong hình ảnh/video: Để bắt đầu quá trình nhận diện khẩu trang, hệ thống cần có khả năng phát hiện và xác định vị trí của người trong hình ảnh

hoặc video. Điều này có thể được thực hiện bằng các thuật toán phát hiện đối tượng như Haar cascades, HOG (Histogram of Oriented Gradients) hoặc các mô hình deep learning như YOLO (You Only Look Once) hoặc SSD (Single Shot MultiBox Detector).

- Phát hiện khẩu trang: Sau khi nhận diện được người trong hình ảnh hoặc video, hệ thống cần phân biệt xem người đó có đeo khẩu trang hay không. Có thể sử dụng các mô hình deep learning như CNN (Convolutional Neural Network) để phân loại khẩu trang và khuôn mặt của người đó. Mô hình này có thể được huấn luyện trên tập dữ liệu lớn chứa các hình ảnh của người đeo khẩu trang và người không đeo khẩu trang.
- Nhận diện người không đeo khẩu trang: Khi hệ thống đã xác định được người không đeo khẩu trang, tiếp theo sẽ tiến hành nhận diện khuôn mặt của người không đeo khẩu trang. Nhờ đó hệ thống sẽ hỗ trợ đảm bảo việc tuân thủ quy định đeo khẩu trang tại các nơi công cộng. Điều này có thể được thực hiện bằng cách gắn kết hệ thống với các cảm biến hoặc thiết bị ngoại vi như loa để phát ra thông báo hoặc hệ thống cảnh báo tự động để nhắc nhở người không đeo khẩu trang.

2.1. Giải pháp về phần cứng và truyền thông



Hình 1. Sơ đồ khối thể hiện tính kết nối của hệ thống

Nguyên lý hoạt động:

- Ứng dụng đã triển khai trên phần cứng Jetson Nano, nhận hình ảnh từ camera và xử lý nhận dạng, dán nhãn phân biệt.

- Ứng dụng sẽ gửi luồng stream đến webserver
- Webserver tiếp nhận và log kết quả nhận diện vào Firebase.
- Firebase sẽ cập nhật và gửi bộ dữ liệu vector đặc trưng lên ứng dụng để tiếp tục quá trình xử lý.

Chi phí triển khai phần cứng hệ thống:

Thiết bị	Chi phí (đồng)
Jeston Nano	5.000.000
Camera	500.000
Thẻ nhớ 64GB	150.000
USB Wifi	150.000
Tổng	5.800.000

Bảng 1. Danh sách thiết bị phần cứng

Hệ thống được cài đặt trên môi trường thực thi sau:

- Jetson Nano, OS Ubuntu 20.04
- CUDA 10.3
- Python 3.8
- OpenCV 4.6
- PyTorch 1.12 & TorchVision 0.13

2.2. Giải pháp về truyền thông

- Xây dựng Web – Server để nhận kết quả xử lý từ Jeston và hiển thị dữ liệu
- Web Client:

Framework sử dụng: ReactJS

Các chức năng chính:

- Đăng nhập, đăng ký, thông tin người dùng
- Xem các kết quả phát hiện khẩu trang và nhận diện khuôn mặt

- Xem video stream trực tiếp từ Jeston nano

- Server:

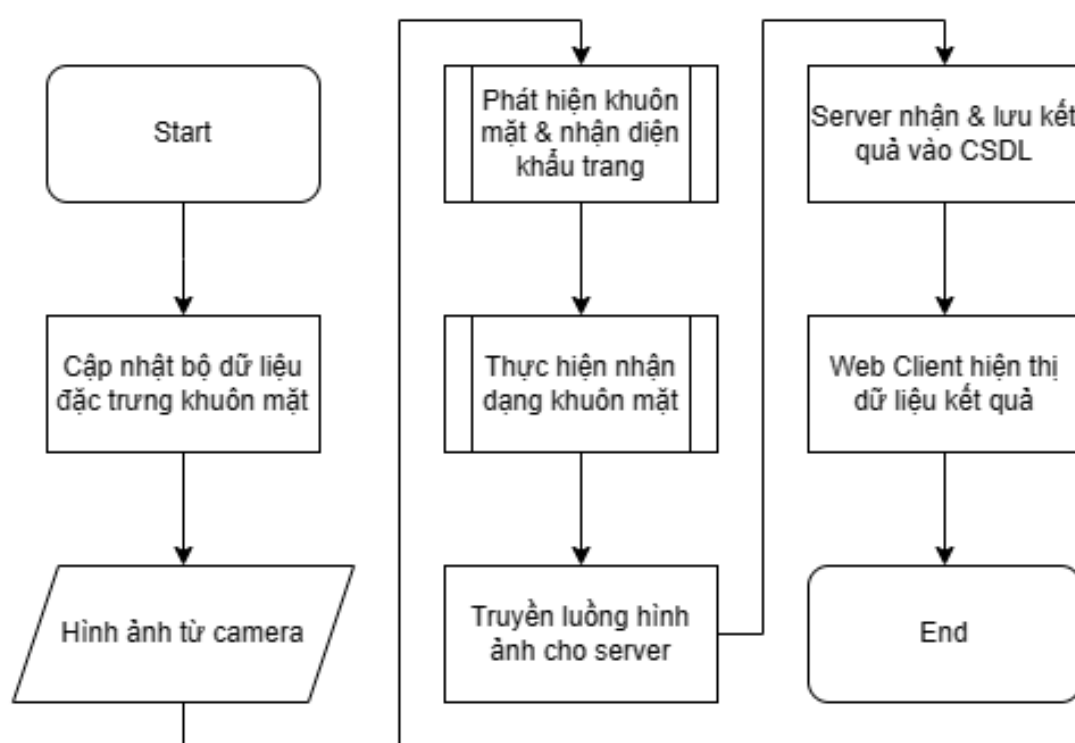
Sử dụng Flask – một framework của Python để xây dựng server

Các chức năng chính:

- Nhận dữ liệu hình ảnh từ Jeston Nano
- Nhận kết quả xử lý từ Jeston Nano
- Lưu trữ kết quả xử lý vào cơ sở dữ liệu

- Cơ sở dữ liệu: sử dụng dịch vụ Firebase do Google cung cấp

2.3. Giải pháp về phần mềm



Hình 2. Sơ đồ thuật toán của hệ thống

Nguyên lý hoạt động:

- Khi khởi động, hệ thống sẽ cập nhật bộ dữ liệu đặc trưng khuôn mặt mới nhất từ cơ sở dữ liệu.
- Dữ liệu đầu vào của hệ thống sẽ là hình ảnh được lấy trực tiếp từ camera theo thời gian thực.

- Đầu tiên, hệ thống xác định các khuôn mặt có trong hình ảnh, kiểm tra trên từng khuôn mặt xem có mang khẩu trang hay không, nếu mang khẩu trang thì log kết quả nhận diện rằng người đó có mang khẩu trang, nếu không thì sẽ chuyển đến bước tiếp theo.
- Ở bước này, hệ thống sẽ tiến hành nhận diện khuôn mặt. Trước hết, hệ thống thực hiện trích xuất đặc trưng khuôn mặt. Sau đó, sẽ thực hiện phân lớp nhận diện khuôn mặt.
- Nếu sau quá trình xử lý, hệ thống nhận diện được người này có trong cơ sở dữ liệu, hệ thống sẽ xuất kết quả ra màn hình. Ngược lại, nếu dữ liệu sau quá trình xử lý không khớp với bất kỳ khuôn mặt nào trong cơ sở dữ liệu, hệ thống sẽ xuất ra kết quả là ‘unknown’.
- Kết quả được gửi về server, lưu vào cơ sở dữ liệu. Người dùng có thể xem dữ liệu kết quả thông qua Web Client.

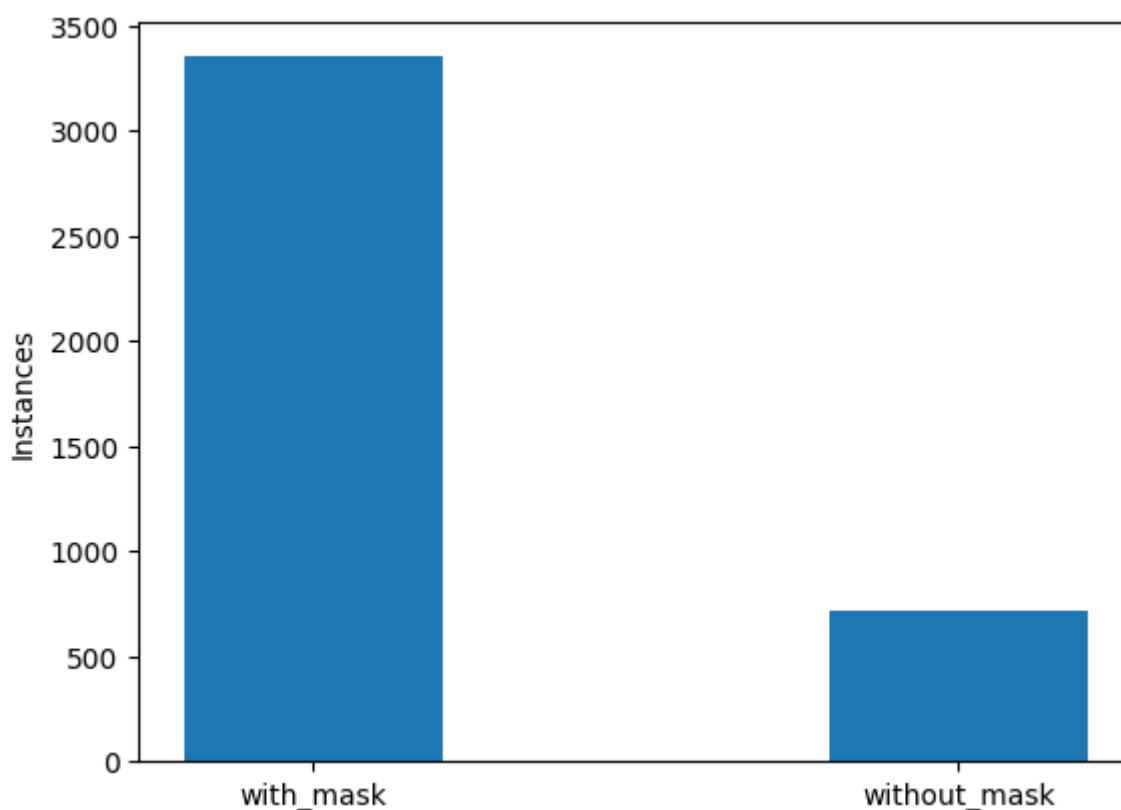
3. Quá trình triển khai

3.1. Thu thập dữ liệu

3.1.1. Bộ dữ liệu khẩu trang

Nhóm sử dụng bộ dữ liệu có sẵn từ Kaggle, kết hợp với những dữ liệu mà nhóm tự thu thập thêm. Bộ dữ liệu gồm 853 ảnh chia thành 2 lớp:

- Đeo khẩu trang
- Không đeo khẩu trang



Hình 3. Biểu đồ số lượng mẫu của từng nhân trong bộ dữ liệu khẩu trang

Để huấn luyện và kiểm thử mô hình, bộ dữ liệu được chia thành 3 tập train, validate, test với tỉ lệ tương ứng là 68%, 12% và 20%.

3.1.2. Bộ dữ liệu khuôn mặt

Nhóm chúng em đã tự xây dựng một tập dữ liệu riêng về hình ảnh bằng cách thu thập và cắt trích trực tiếp từ video của các thành viên trong nhóm và bạn bè trong lớp. Mục tiêu của chúng em là thu thập dữ liệu đa dạng và đầy đủ, bao gồm mọi góc mặt của từng cá nhân trong đoạn video. Với mỗi thành viên, chúng em yêu cầu họ quay một đoạn video kéo dài khoảng 30 giây, đảm bảo rằng toàn bộ khuôn mặt của họ được ghi lại trong quá trình quay.

Sau khi thu thập được video, chúng em đã triển khai một hàm Python để tự động cắt video thành nhiều hình ảnh khác nhau và lưu chúng vào cơ sở dữ liệu theo từng người.

Sau khi tiến hành xử lý tập dữ liệu, nhóm thu được bộ dữ liệu gồm 356 ảnh khuôn mặt từ các góc khác nhau của nhiều người, chia nó thành 2 phần train/test theo tỉ lệ xác định nhằm đảm bảo tính khách quan và đáng tin cậy của quá trình huấn luyện và đánh giá mô hình. Cụ thể, tỉ lệ được xác định như sau:

- Phần train (huấn luyện): Đây là phần dữ liệu được sử dụng để huấn luyện mô hình. Chúng em đã chọn một tỉ lệ phù hợp để đảm bảo mô hình có đủ dữ liệu

để học và hội tụ. Tỷ lệ của phần train được thiết lập ở mức 80% của toàn bộ tập dữ liệu.

- Phần test (kiểm tra): Phần dữ liệu kiểm tra được sử dụng để đánh giá hiệu suất của mô hình sau khi hoàn thành quá trình đào tạo và đánh giá. Đây là phần dữ liệu không được sử dụng trong quá trình huấn luyện và đánh giá mô hình. Tỷ lệ của phần test được thiết lập ở mức 20% của toàn bộ tập dữ liệu.

3.2. Công cụ và thư viện

- OpenCV: Sử dụng để xử lý ảnh, phát hiện khuôn mặt và trích xuất đặc trưng.
- PyTorch: Sử dụng để xây dựng mô hình nhận dạng khuôn mặt và huấn luyện mạng neural.
- Scikit-learn: Sử dụng để huấn luyện các thuật toán phân loại: SVM hoặc KNN.
- NumPy: Sử dụng để xử lý dữ liệu số học và tính toán.

3.3. Điều kiện tiến hành thực nghiệm

- Ánh sáng: Cố gắng sử dụng ánh sáng đồng đều và không quá chói hoặc không quá tối.
- Thời tiết: Không có yêu cầu đặc biệt về thời tiết, tuy nhiên, trong môi trường ngoài trời, các yếu tố như mưa, sương mù có thể ảnh hưởng đến chất lượng của ảnh và hiệu suất của hệ thống.

3.4. Các độ đo (metrics) để đánh giá hiệu suất

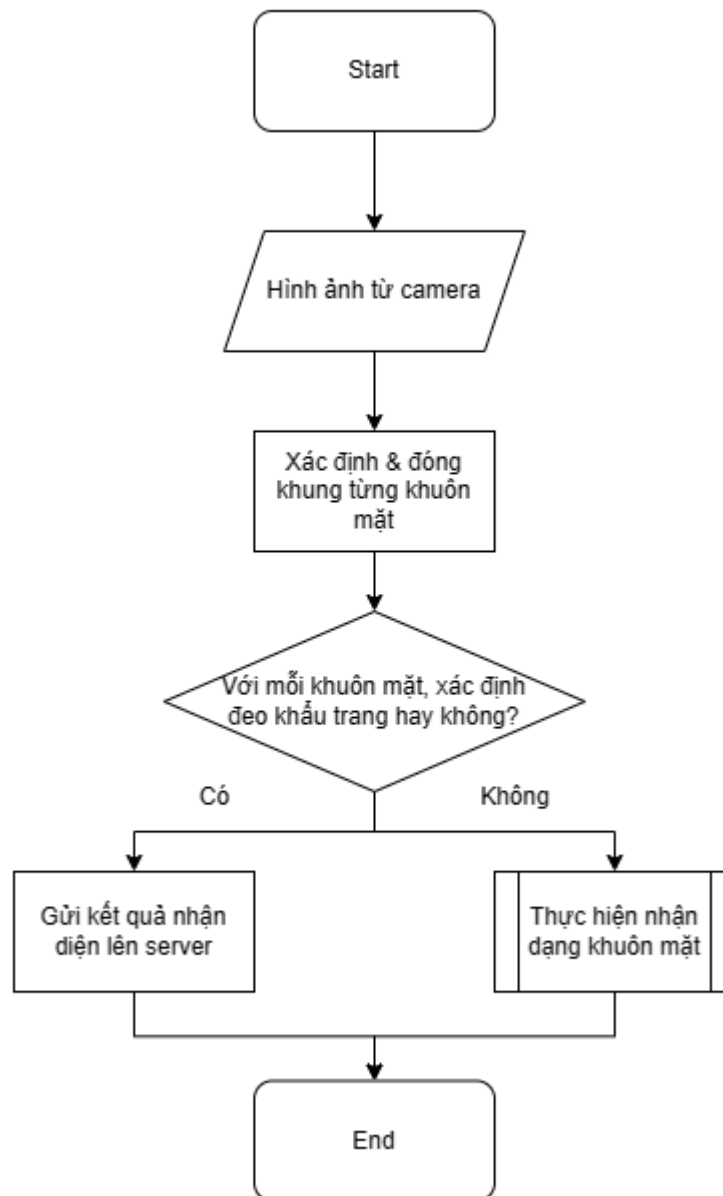
- Độ chính xác (Accuracy): Đo lường tỉ lệ dự đoán chính xác đối với các trường hợp người đeo khẩu trang và người không đeo khẩu trang.
- Tốc độ thực thi (FPS - Frames Per Second): Đo lường số lượng khung hình xử lý được trong một giây, thể hiện tốc độ xử lý của hệ thống.
- Độ ổn định: Đánh giá khả năng của hệ thống duy trì hiệu suất và độ chính xác ổn định trong thời gian dài và trường hợp dữ liệu đa dạng.
- Tính bảo mật: Đánh giá khả năng bảo vệ dữ liệu và thông tin người dùng, đảm bảo không có sự xâm nhập hay vi phạm quyền riêng tư.

3.5. Quy trình và điều kiện kiểm thử hệ thống

- Thu thập dữ liệu ảnh chứa người đeo khẩu trang và người không đeo khẩu trang.
- Tiền xử lý dữ liệu: Phát hiện khuôn mặt, trích xuất đặc trưng và chuẩn bị dữ liệu huấn luyện và kiểm thử.
- Huấn luyện mô hình: Sử dụng dữ liệu huấn luyện để huấn luyện mô hình phân loại.
- Đánh giá hiệu suất: Sử dụng dữ liệu kiểm thử để đánh giá độ chính xác, tốc độ thực thi và các độ đo khác.
- Tinh chỉnh mô hình: Nếu kết quả đánh giá không đạt yêu cầu, tiến hành điều chỉnh các tham số và kiến trúc mô hình để cải thiện hiệu suất.
- Lặp lại quy trình kiểm thử và đánh giá cho đến khi đạt được kết quả tốt nhất.

4. Thuật toán triển khai

4.1. Thuật toán phát hiện khuôn mặt và nhận diện khẩu trang

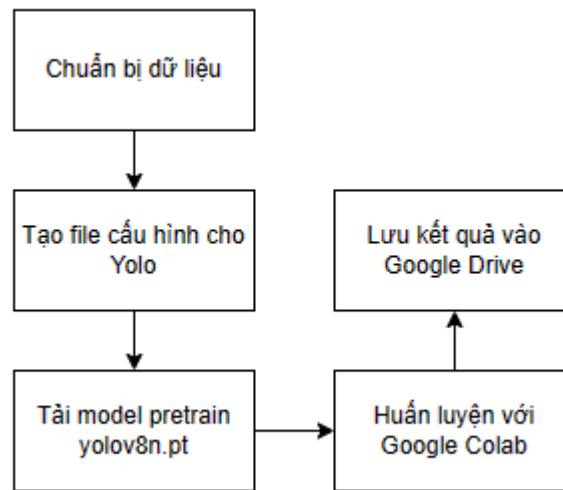


Hình 4. Sơ đồ thuật toán phát hiện khuôn mặt và nhận diện khẩu trang

Việc phát hiện khuôn mặt và nhận diện khẩu trang sẽ được thực hiện bằng mô hình Yolo hoặc SSD.

4.1.1. YOLO

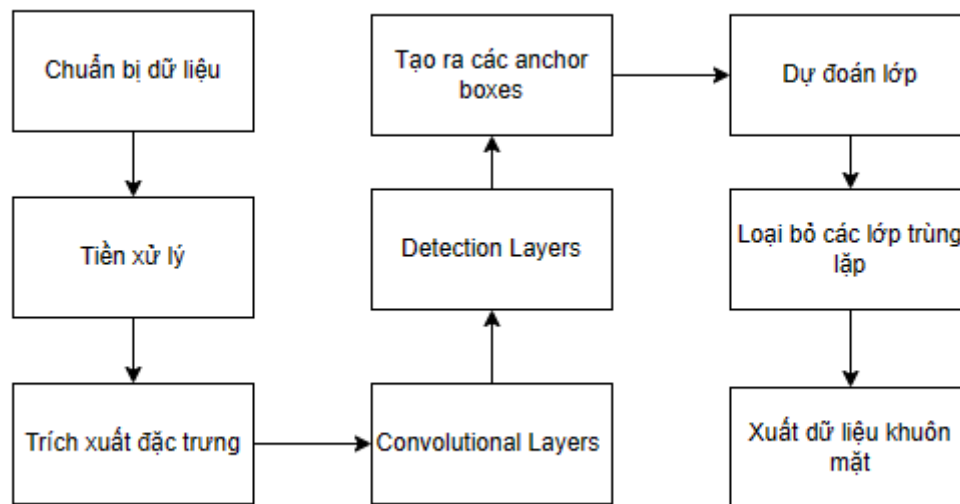
Quá trình huấn luyện mô hình YOLO sử dụng bộ dữ liệu khẩu trang.



Hình 5. Quá trình huấn luyện YOLO

Về mô hình YOLO V8N: là mô hình nhận dạng vật thể được huấn luyện sẵn trên COCO Dataset

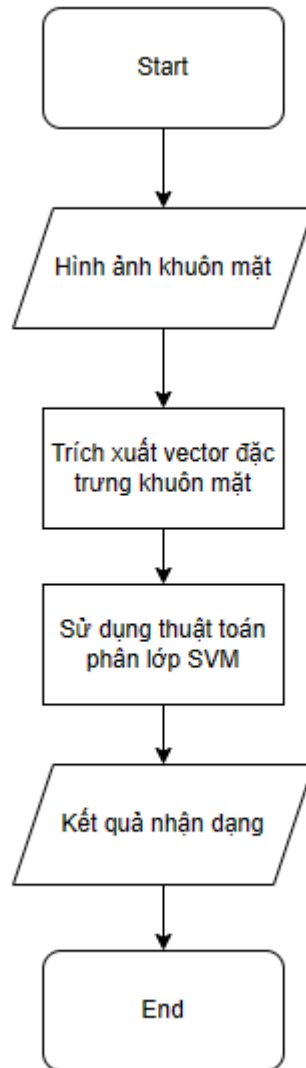
4.1.2. SSD



Hình 6. Quá trình huấn luyện SSD

Tương tự như YOLO, mô hình SSD được huấn luyện với bộ dữ liệu khẩu trang.

4.2. So sánh các thuật toán trích xuất đặc trưng khuôn mặt



Hình 7. Sơ đồ khối thuật toán nhận dạng khuôn mặt

Để thực hiện trích xuất đặc trưng khuôn mặt, nhóm đã chọn 3 thuật toán trích xuất đặc trưng khuôn mặt: FaceNet, LBPH & Eigenface; kết hợp cùng với thuật toán SVM để phân lớp.

Dữ liệu cho quá trình huấn luyện bao gồm 280 mẫu, mỗi mẫu có kích thước 160x160 pixel và 3 kênh màu RGB.

Dữ liệu cho quá trình kiểm thử bao gồm 76 mẫu, mỗi mẫu có kích thước 160x160 pixel và 3 kênh màu RGB.

4.2.1. Facenet

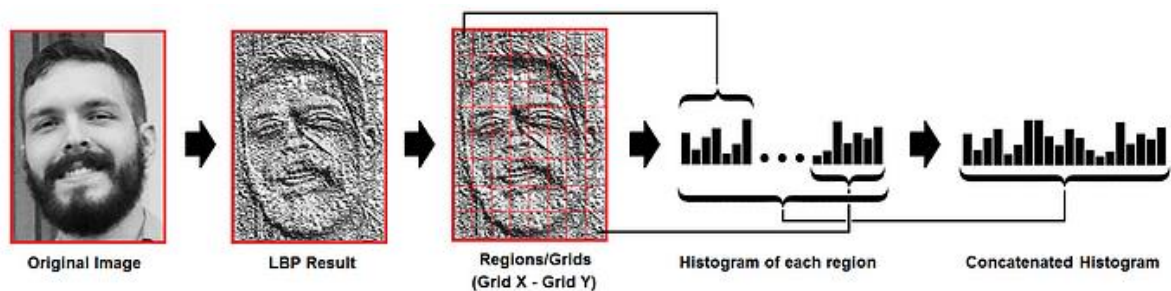
FaceNet là một thuật toán nhận diện khuôn mặt sử dụng deep learning. Dưới đây là các bước chính trong quá trình hoạt động của FaceNet:

- Trích xuất đặc trưng
- Normalization
- Triplet Loss
- Siamese Network
- Xác định đối tượng

Đối với FaceNet, vector đặc trưng được sử dụng có số chiều là 512.

4.2.2. LBPH

Thuật toán LBPH (Local Binary Patterns Histograms) là một phương pháp nhận diện khuôn mặt dựa trên mô hình không gian màu và các mẫu nhị phân cục bộ.

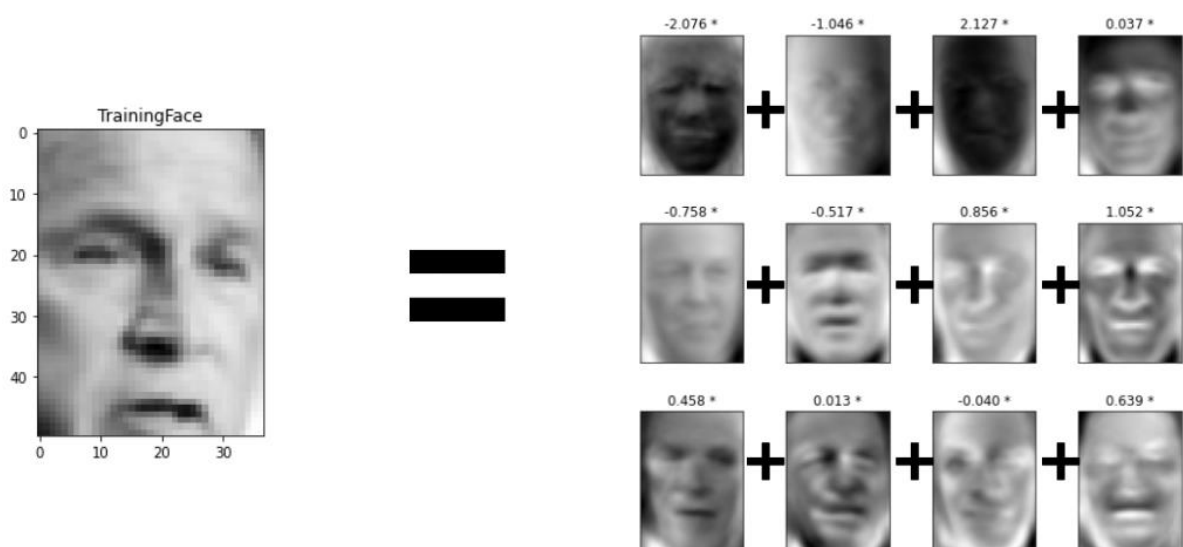


Hình 8. Ví dụ về LBPH

Số chiều của vector đặc trưng đối với thuật toán LBPH là 512.

4.2.3. Eigenface

Nguyên lý hoạt động của thuật toán Eigenface dựa trên phân tích thành phần chính (Principal Component Analysis - PCA) để trích xuất và biểu diễn khuôn mặt dưới dạng các "eigenfaces".



Hình 9. Ví dụ về Eigenface

Nhóm đã sử dụng PCA để giảm chiều của vector đặc trưng nhằm loại bỏ các thành phần không quan trọng, giúp giảm kích thước và độ phức tạp. Số chiều sau của vector đặc trưng sau khi giảm là 100.

4.2.4. So sánh kết quả và đánh giá

Sau khi triển khai ba thuật toán trích xuất đặc trưng khuôn mặt, kết hợp với thuật toán phân lớp SVM, nhóm thu được kết quả như sau:

Thuật toán	FaceNet + SVM	LBPH + SVM	Eigenface + SVM
Accuracy score	98.68%	97.37%	92.11%

Bảng 2. So sánh kết quả các thuật toán trích xuất đặc trưng khuôn mặt

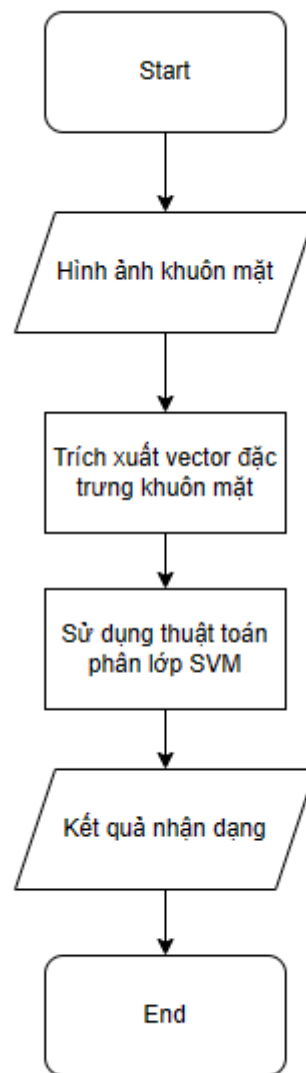
Dựa trên kết quả trên, nhóm có đánh giá như sau:

- FaceNet và LBPH đạt độ chính xác tương đối cao và không có sự khác biệt lớn trong độ chính xác giữa hai thuật toán này. Cả hai đều sử dụng số chiều đặc trưng lớn (512 chiều) và có khả năng trích xuất thông tin chi tiết và phức tạp từ khuôn mặt. FaceNet và LBPH đều là lựa chọn tốt trong việc nhận dạng khuôn mặt.
 - Eigenface đạt độ chính xác thấp hơn so với FaceNet và LBPH. Mặc dù sử dụng số chiều đặc trưng thấp (100 chiều), Eigenface không thể trích xuất và biểu diễn đủ thông tin để phân biệt giữa các khuôn mặt với độ chính xác cao như hai thuật toán kia.
- Nhóm chọn FaceNet để thực hiện việc trích xuất đặc trưng khuôn mặt bởi vì độ chính xác cao mà FaceNet đem lại.

4.3. Thuật toán nhận diện phân lớp khuôn mặt

Để thực hiện phân lớp nhận diện khuôn mặt, nhóm đã chọn SVM và Euclidean Distance.

4.3.1. Support Vector Machines (SVM)



Hình 10. Sơ đồ khối nhận dạng khuôn mặt sử dụng phân lớp SVM

Các siêu tham số của mô hình SVM:

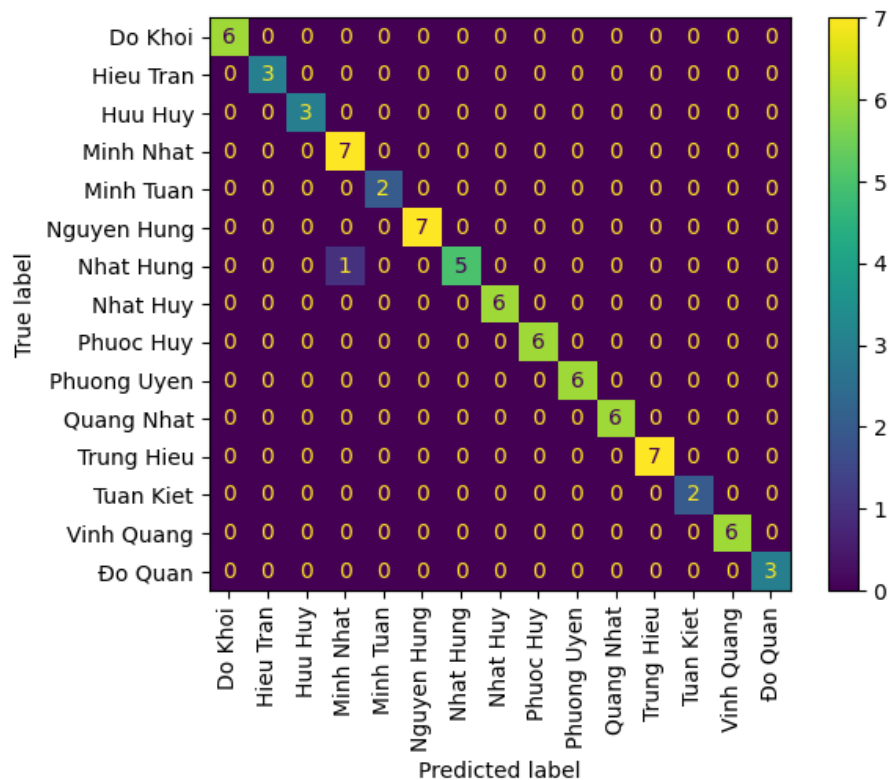
- Kernel: SVM có thể sử dụng một số loại kernel khác nhau để ánh xạ dữ liệu vào không gian cao hơn. Các kernel phổ biến bao gồm Linear Kernel, Polynomial Kernel, Radial Basis Function (RBF) Kernel, và Sigmoid Kernel.
- C: Tham số C điều chỉnh sự đánh đổi giữa việc tối thiểu hóa sai số phân loại và việc tạo ra một ranh giới quyết định phân loại tốt hơn.
- Gamma: quyết định mức độ tác động của một mẫu dữ liệu đào tạo đến việc tạo ra đường biên quyết định. Giá trị gamma càng cao, mức độ ảnh hưởng càng nhỏ và đường biên quyết định càng mượt mà. Giá trị gamma thấp có thể dẫn đến overfitting.

- Degree: Tham số degree chỉ áp dụng cho kernel đa thức (Polynomial Kernel). Nó xác định bậc của đa thức được sử dụng để ánh xạ dữ liệu.
- Coefficient: Tham số coefficient cũng chỉ áp dụng cho kernel đa thức (Polynomial Kernel). Nó quyết định độ lớn của hệ số đa thức.
- Shrinking: Nếu shrinking được bật, SVM sẽ sử dụng kỹ thuật thu nhỏ đặc biệt để tìm kiếm vector hỗ trợ.

Bộ siêu tham số mà nhóm đã sử dụng trong quá trình huấn luyện:

- Kernel: Linear Kernel
- C: 100
- Gamma: 0.001

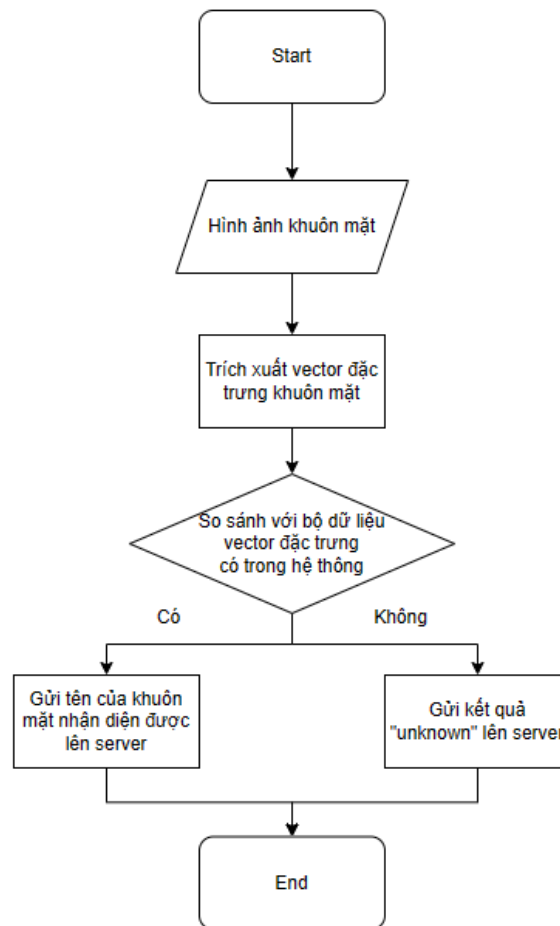
Dưới đây là kết quả của ma trận nhầm lẫn sau khi sử dụng thuật toán SVM:



Hình 11. Ma trận nhầm lẫn của thuật toán SVM

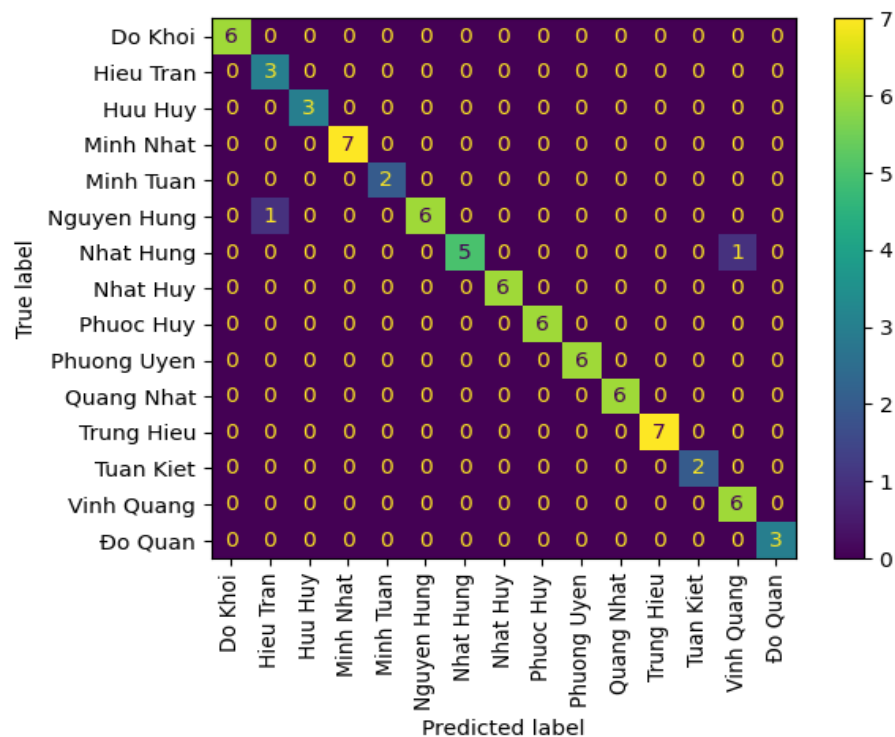
4.3.2. Euclidean Distance

Quá trình nhận dạng khuôn mặt sử dụng thuật toán trích xuất đặc trưng FaceNet và phân lớp Euclidean Distance được thực hiện theo sơ đồ khối sau:



Hình 12. Sơ đồ khối nhận dạng khuôn mặt sử dụng thuật toán Euclidean Distance

Dưới đây là kết quả của ma trận nhầm lẫn sau khi sử dụng thuật toán Euclidean Distance:



Hình 13. Ma trận nhầm lẫn của thuật toán Euclidean Distance

4.3.3. So sánh kết quả

Khi kiểm thử thuật toán trích xuất đặc trưng FaceNet với 2 thuật toán phân lớp, nhóm thu được kết quả như sau:

Thuật toán	SVM	Euclidean Distance
Accuracy score	98.68%	97.37%

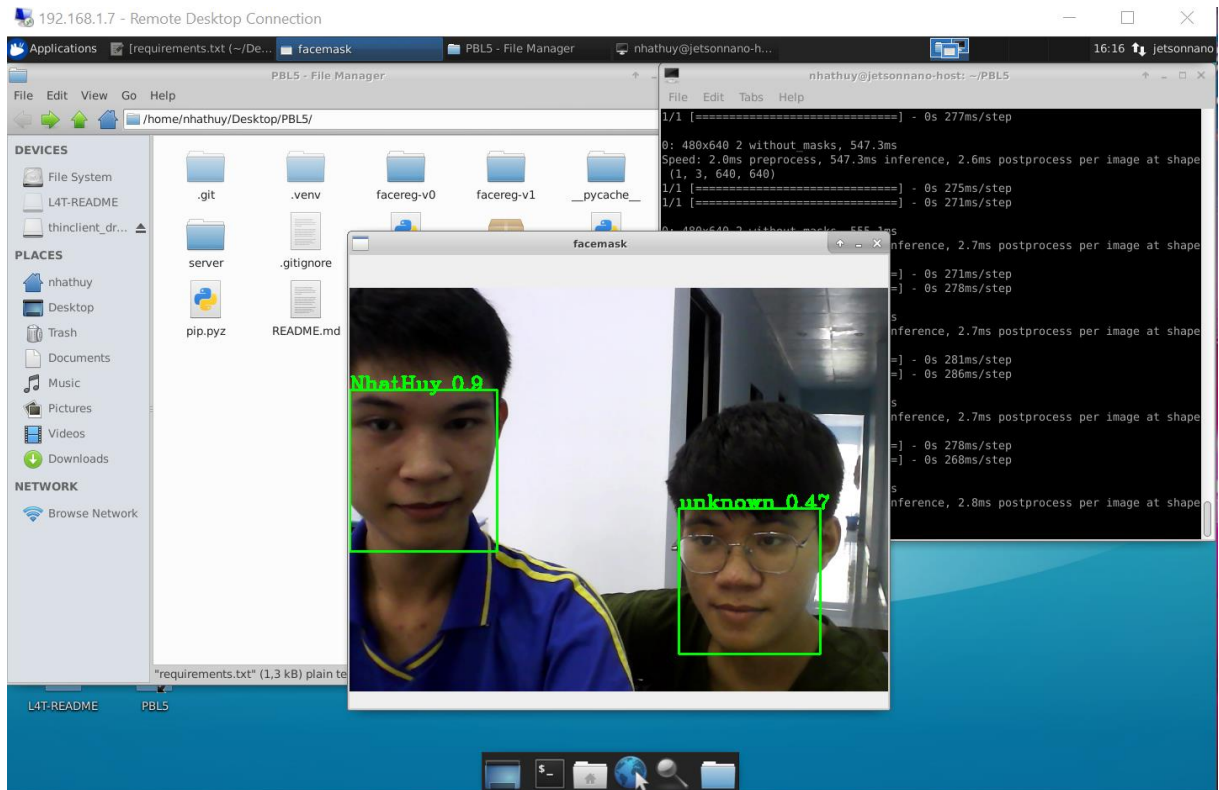
Bảng 3. So sánh kết quả các thuật toán phân lớp

- Kết quả thu được giữa 2 thuật toán không có sự chênh lệch lớn.
- Thực tế khi triển khai, nếu cần thêm khuôn mặt mới vào dataset, SVM yêu cầu phải huấn luyện lại toàn bộ dataset, Euclidean Distance thì không.
- Nhóm chọn Euclidean Distance để phân lớp nhận diện khuôn mặt.

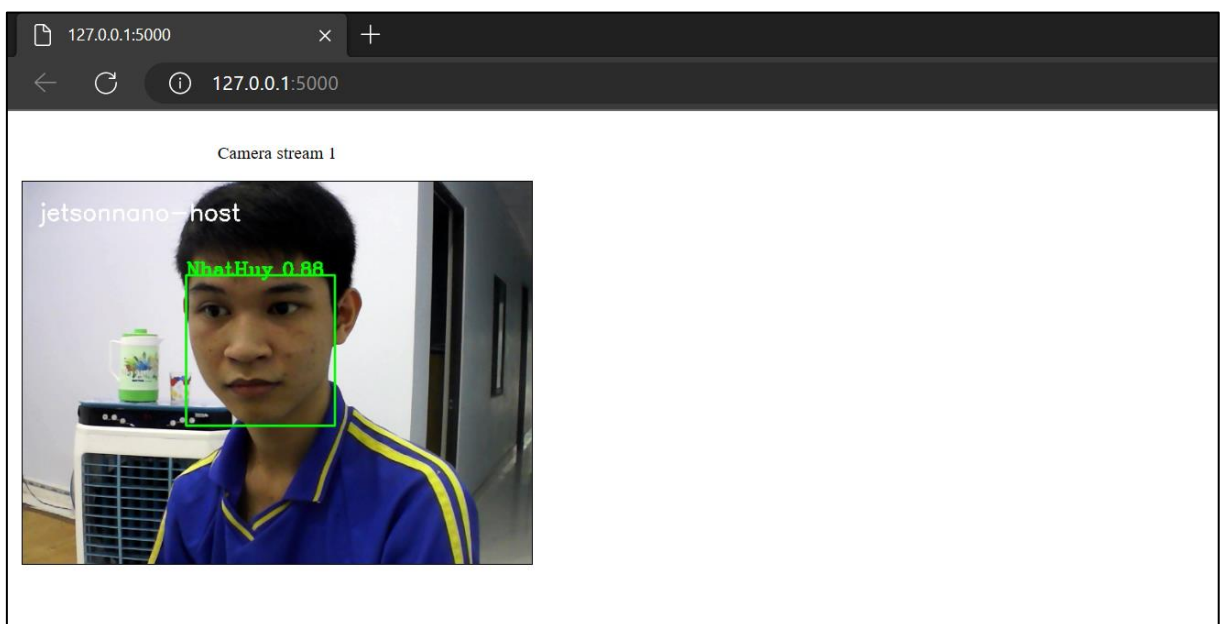
5. Kết quả đạt được

Sau khi triển khai hệ thống, nhóm thu được các kết quả sau:

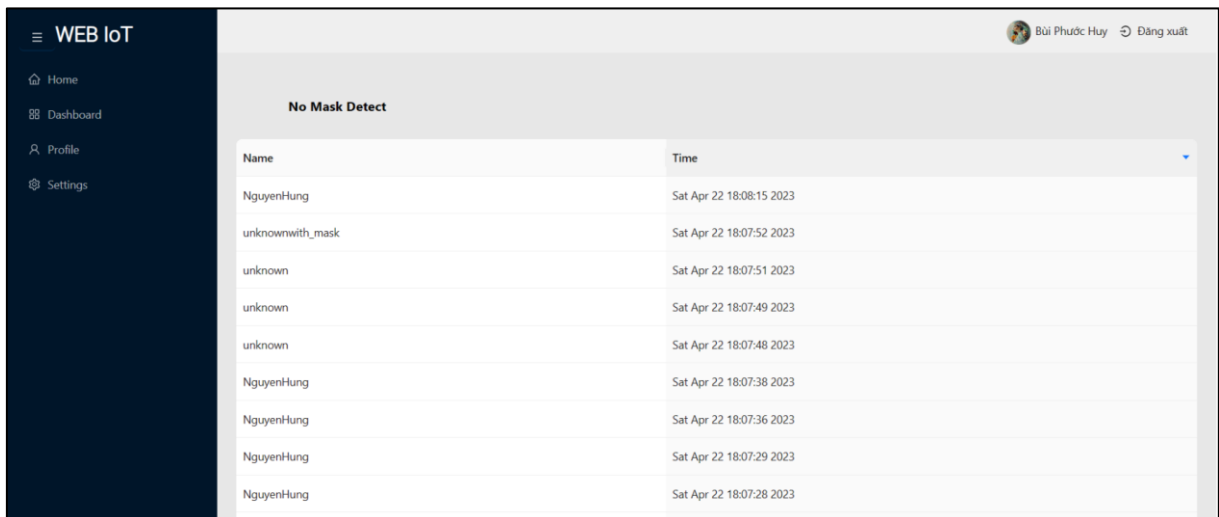
- Kết quả nhận diện chính xác, lên đến gần 90%.
- Tốc độ xử lý nhanh, đạt được 10-15 FPS.
- Độ trễ thấp, dao động trong khoảng 100-150 ms.
- Có thể hoạt động ổn định trong thời gian dài.
- Đảm bảo được tính bảo mật thông tin.



Hình 14.. Ứng dụng chạy trên Jetson Nano

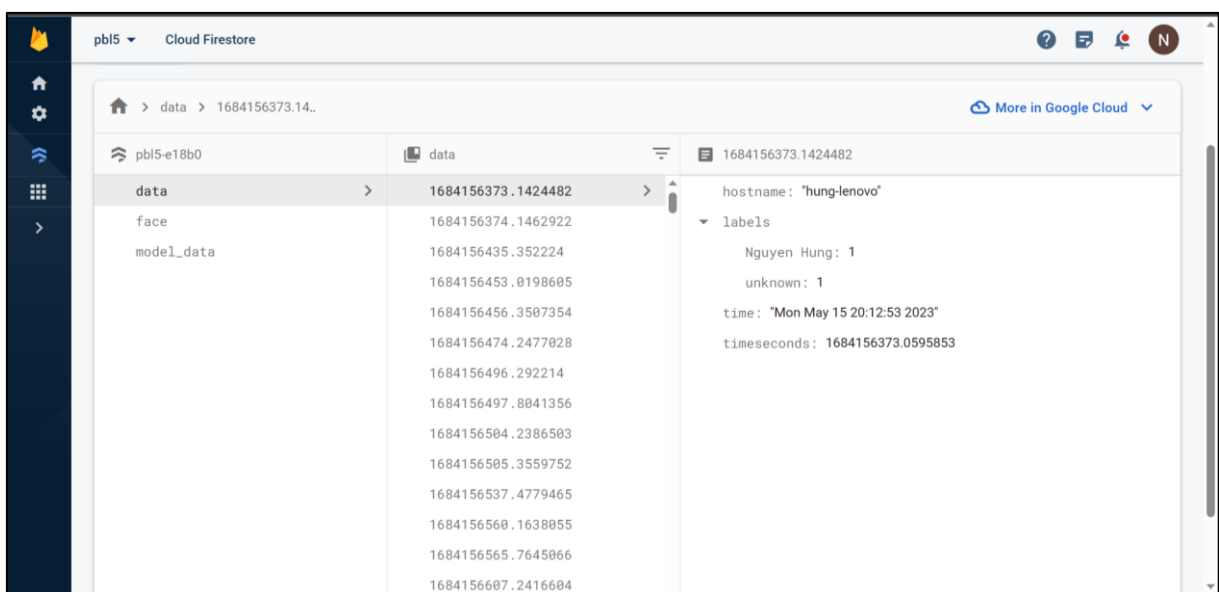


Hình 15. Web Server nhận luồng dữ liệu từ Jetson Nano



Name	Time
NguyenHung	Sat Apr 22 18:08:15 2023
unknownwith_mask	Sat Apr 22 18:07:52 2023
unknown	Sat Apr 22 18:07:51 2023
unknown	Sat Apr 22 18:07:49 2023
unknown	Sat Apr 22 18:07:48 2023
NguyenHung	Sat Apr 22 18:07:38 2023
NguyenHung	Sat Apr 22 18:07:36 2023
NguyenHung	Sat Apr 22 18:07:29 2023
NguyenHung	Sat Apr 22 18:07:28 2023

Hình 16. Web Client hiện thị kết quả nhận diện



data	1684156373.1424482
face	1684156374.1462922
model_data	1684156435.352224
	1684156453.0198605
	1684156456.3507354
	1684156474.2477028
	1684156496.292214
	1684156497.8041356
	1684156504.2386503
	1684156505.3559752
	1684156537.4779465
	1684156560.1638055
	1684156565.7645066
	1684156607.2416604

hostname: "hung-lenovo"
labels
Nguyen Hung: 1
unknown: 1
time: "Mon May 15 20:12:53 2023"
timeseconds: 1684156373.0595853

Hình 17. Firebase lưu trữ dữ liệu vector khuôn mặt và kết quả nhận diện

6. Kết luận

Sau thời gian hoàn thành đồ án về hệ thống phát hiện người đeo khẩu trang và nhận diện người không đeo khẩu trang, so sánh với yêu cầu đã đặt ra, nhóm rút ra kết luận về các kết quả đã đạt được:

- Đánh giá so với yêu cầu đặt ra:
 - Sản phẩm đồ án đã phát triển thành công một hệ thống có khả năng phát hiện người đeo khẩu trang từ hình ảnh. Điều này đáp ứng yêu cầu đặt ra về việc nhận diện khẩu trang trên khuôn mặt.

- Sản phẩm cũng đã đạt được yêu cầu đối với việc nhận diện người không đeo khẩu trang từ hình ảnh, góp phần xác định những người có nguy cơ tiềm tàng trong các tình huống cần thiết.
- Kết luận về kết quả đạt được:
 - Nhóm đã phát triển thành công một hệ thống có khả năng phát hiện và nhận diện khẩu trang, đáp ứng yêu cầu đối với vấn đề quan trọng trong việc kiểm soát an ninh và đảm bảo sức khỏe cộng đồng.
 - Hệ thống đã đạt được mức độ chính xác và đáng tin cậy đủ để áp dụng trong các tình huống thực tế, đảm bảo sự hiệu quả và tin cậy của quá trình phân loại.
- Hướng phát triển của dự án:
 - Nâng cao hiệu suất và chính xác của hệ thống: Cần tiếp tục tối ưu hóa và cải thiện các thuật toán và mô hình trong việc phát hiện và nhận diện khẩu trang, nhằm đảm bảo tính chính xác và hiệu quả trong các điều kiện ánh sáng khác nhau và góc nhìn đa dạng.
 - Mở rộng dữ liệu huấn luyện: Tiếp tục thu thập và bổ sung thêm dữ liệu huấn luyện từ nhiều nguồn và tình huống khác nhau, nhằm đảm bảo tính đa dạng và đại diện của tập dữ liệu.
 - Tích hợp hệ thống vào môi trường thực tế: Để triển khai hệ thống trên một quy mô lớn, cần xem xét các yếu tố như tương thích phần cứng, tối ưu hóa thời gian đáp ứng và tích hợp với các hệ thống an ninh hoặc giám sát hiện có.
 - Nghiên cứu và áp dụng các công nghệ tiên tiến: Có thể nghiên cứu và tích hợp các công nghệ như học sâu (deep learning), học tăng cường (reinforcement learning), hoặc tích hợp hệ thống vào các thiết bị di động để tăng tính di động và sử dụng rộng rãi của nó.

DANH MỤC TÀI LIỆU THAM KHẢO

- [1] Phạm Đình Khánh, *YOLO You Only Look Once*: [Khoa học dữ liệu \(phamdinhkhanh.github.io\)](https://phamdinhkhanh.github.io), 2020.
- [2] Nero Phung, *Tìm hiểu về SSD - Single Shot Detector trong bài toán Object Detection*: [Tìm hiểu về SSD - Single Shot Detector trong bài toán Object Detection - Nero's Blog \(nerophung.github.io\)](https://nerophung.github.io), 2019.
- [3] Larxel, *Face Mask Dataset*: [Face Mask Detection | Kaggle](https://www.kaggle.com/larxel/face-mask-dataset), 2020.
- [4] Muhammad-Yunus, *LBPH*: [Materi-Training/pertemuan 6.ipynb at main · Muhammad-Yunus/Materi-Training \(github.com\)](https://github.com/Muhammad-Yunus/Materi-Training), 2020.
- [5] Geeks for Geeks, *Face Recognition Using Eigenfaces (PCA Algorithm)*: [ML | Face Recognition Using Eigenfaces \(PCA Algorithm\) - GeeksforGeeks](https://www.geeksforgeeks.org/face-recognition-using-eigenfaces-pca-algorithm/), 2021.