

ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN



BÁO CÁO ĐỒ ÁN
NHẬP MÔN THỊ GIÁC MÁY TÍNH

Đề tài:

Tên tiếng Việt: Phân loại khuôn mặt đeo khẩu trang

Tên tiếng Anh: Face Mask Classification

Nhóm sinh viên thực hiện:

Nguyễn Công Phát - 23521143 - Nhóm trưởng

Nguyễn Lê Phong - 23521168

Vũ Việt Cương - 23520213

GVHD: TS. Mai Tiến Dũng

Lớp học phần: CS231.Q11

Ngày 19 tháng 1 năm 2026

MỤC LỤC

LỜI NÓI ĐẦU	6
THÔNG TIN THÀNH VIÊN	7
CHƯƠNG 1. TỔNG QUAN ĐỀ TÀI	8
1.1 Mô tả bài toán	8
1.2 Lý do thực hiện bài toán	8
1.3 Phát biểu bài toán	9
1.4 Quy trình thực hiện tổng quát (Workflow)	9
1.5 Mục tiêu của đề tài	10
CHƯƠNG 2. PHƯƠNG PHÁP THỰC HIỆN BÀI TOÁN	12
2.1 Tiền xử lý ảnh	12
2.1.1 Chuẩn hóa kích thước ảnh	12
2.1.2 Chuẩn hóa giá trị điểm ảnh	12
2.1.3 Chuyển đổi không gian màu	13
2.2 Trích xuất đặc trưng	13
2.2.1 Đặc trưng HOG (Histogram of Oriented Gradients)	13
2.2.1.1 Định nghĩa	13
2.2.1.2 Nguyên lý hoạt động	13
2.2.1.3 Tại sao lựa chọn đặc trưng HOG?	15
2.2.1.4 Cài đặt trong bài toán	15
2.2.1.5 Nhận xét về HOG	16
2.2.2 Đặc trưng LBP (Local Binary Patterns)	16
2.2.2.1 Định nghĩa	16
2.2.2.2 Nguyên lý hoạt động	17
2.2.2.3 Cài đặt trong bài toán	18
2.2.2.4 Nhận xét về đặc trưng LBP	19
2.2.3 Tổng hợp đặc trưng	20
2.3 Các mô hình phân loại	20
2.3.1 Mô hình Support Vector Machine (SVM)	20
2.3.1.1 Nguồn gốc	20
2.3.1.2 Nguyên lý hoạt động	21
2.3.1.3 Cài đặt trong bài toán	21
2.3.2 Mô hình K-Nearest Neighbors (KNN)	21
2.3.2.1 Nguồn gốc	21
2.3.2.2 Nguyên lý hoạt động	22
2.3.2.3 Cài đặt trong bài toán	22

2.3.3	Mô hình Random Forest (RF)	22
2.3.3.1	Nguồn gốc	22
2.3.3.2	Nguyên lý hoạt động	22
2.3.3.3	Cài đặt trong bài toán	23
2.3.4	Mô hình Mạng Neural Tích chập (CNN)	23
2.3.4.1	Nguồn gốc	23
2.3.4.2	Nguyên lý hoạt động	23
2.3.4.3	Cài đặt trong bài toán	24
2.4	Phương pháp tối ưu siêu tham số (Optuna & Keras Tuner)	25
2.4.1	Optuna	25
2.4.1.1	Nguyên lý hoạt động	25
2.4.1.2	Nhận xét chung về Optuna	25
2.4.1.3	Ứng dụng	26
2.4.1.4	Cài đặt Optuna cho SVM	26
2.4.1.5	Cài đặt Optuna cho Random Forest	27
2.4.1.6	Cài đặt Optuna cho KNN	27
2.4.2	Keras Tuner	28
2.4.2.1	Nguyên lý hoạt động	28
2.4.2.2	Nhận xét chung về Keras Tuner	29
2.4.2.3	Ứng dụng	29
2.4.2.4	Cấu hình Keras Tuner cho CNN	29
CHƯƠNG 3. THỰC NGHIỆM VÀ KẾT QUẢ		32
3.1	Bộ dữ liệu (Dataset)	32
3.1.1	Tổng quan bộ dữ liệu	32
3.1.2	Cấu trúc bộ dữ liệu	33
3.2	Độ đo đánh giá hiệu năng	33
3.3	Thiết lập thực nghiệm	35
3.4	Kết quả thực nghiệm	36
3.4.1	Mô hình Support Vector Machine (SVM)	36
3.4.1.1	Đặc trưng HOG 8x2	36
3.4.1.2	Đặc trưng HOG 6x3	37
3.4.1.3	Đặc trưng LBP	37
3.4.1.4	Tổng hợp kết quả mô hình SVM	38
3.4.2	Mô hình K-Nearest Neighbors (KNN)	39
3.4.2.1	Đặc trưng HOG 8x2	40
3.4.2.2	Đặc trưng HOG 6x3	40
3.4.2.3	Đặc trưng LBP	41
3.4.2.4	Tổng hợp kết quả mô hình KNN	42
3.4.3	Mô hình Random Forest (RF)	44
3.4.3.1	Đặc trưng HOG 8x2	44
3.4.3.2	Đặc trưng HOG 6x3	44
3.4.3.3	Đặc trưng LBP	45
3.4.3.4	Tổng hợp kết quả mô hình Random Forest	46
3.4.4	Mô hình Convolutional Neural Network (CNN)	46
3.4.4.1	Cấu hình và Kết quả thực nghiệm	46
3.4.4.2	Ma trận nhầm lẫn (Confusion Matrix)	47

3.4.4.3	Phân tích dự đoán sai	47
3.4.5	Tổng kết kết quả thực nghiệm	48
CHƯƠNG 4. DEMO VÀ ĐÁNH GIÁ HỆ THỐNG		51
4.1	Ứng dụng Web phân loại ảnh (Flask Framework)	51
4.2	Hệ thống nhận diện thời gian thực qua Webcam	52
4.3	Đánh giá khả năng vận hành	52
4.4	Tổng kết	53
CHƯƠNG 5. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN		54
5.1	Kết luận	54
5.2	Hướng phát triển	54
TÀI LIỆU THAM KHẢO		56

DANH SÁCH HÌNH ẢNH

1.1	Quy trình thực hiện tổng quát của hệ thống nhận diện khuôn mặt đeo khẩu trang	10
2.1	Minh họa ảnh Gradient: G_x (trái), G_y (giữa) và Magnitude (phải)	14
2.2	Trực quan hóa hướng Gradient bên trong một ô (cell) 8×8	14
2.3	Trực quan hóa bộ mô tả HOG trích xuất trên khuôn mặt	15
2.4	Trực quan hóa HOG cấu hình 6×3	16
2.5	Trực quan hóa HOG cấu hình 8×2	16
2.6	Minh họa kết quả trích xuất HOG trên khuôn mặt mẫu	16
2.7	Quy trình ngưỡng hóa và tạo mã nhị phân LBP từ vùng lân cận 3×3	17
2.8	Minh họa Histogram của các mẫu LBP trích xuất từ hình ảnh	18
2.9	Cài đặt thuật toán LBP và trích xuất Histogram trong Notebook	18
2.10	Trực quan hóa đặc trưng LBP trên khuôn mặt đeo khẩu trang (kết cấu bề mặt được làm nổi bật)	19
2.11	Minh họa siêu phẳng phân tách với lề cực đại (Maximum Margin) trong SVM	21
2.12	Quy trình phân loại dựa trên láng giềng gần nhất trong KNN	22
2.13	Cấu trúc mô hình Random Forest với sự kết hợp của nhiều cây quyết định	23
2.14	Kiến trúc tổng quát của mạng CNN từ giai đoạn trích xuất đặc trưng đến phân loại	24
2.15	Quy trình lựa chọn mô hình và tối ưu siêu tham số trong giai đoạn thử nghiệm 3	25
2.16	Code snippet minh họa quá trình cài đặt Optuna cho HOG + SVM	26
2.17	Code snippet minh họa quá trình tối ưu có điều kiện cho LBP + SVM	26
2.18	Code snippet minh họa quá trình tối ưu Random Forest với Pruning cho đặc trưng HOG	27
2.19	Code snippet minh họa quá trình tối ưu Random Forest tiêu chuẩn cho đặc trưng LBP	27
2.20	Code snippet minh họa quá trình tối ưu KNN trên đặc trưng HOG	28
2.21	Code snippet minh họa quá trình tối ưu KNN trên đặc trưng LBP	28
3.1	Các mẫu hình ảnh thực tế trong bộ dữ liệu Face Mask 12k Images	32
3.2	Tỷ lệ phân chia dữ liệu giữa các tập Train, Validation và Test	33
3.3	Cấu trúc Ma trận nhầm lẫn dùng để tính toán các độ đo hiệu năng	34
3.4	Các mẫu dự đoán sai tiêu biểu (False Positive và False Negative) của SVM + HOG 8×2	36
3.5	Mẫu dự đoán sai tiêu biểu của mô hình SVM với đặc trưng HOG 6×3	37
3.6	Mẫu dự đoán sai tiêu biểu của mô hình SVM với đặc trưng LBP	38
3.7	Ma trận nhầm lẫn SVM + HOG 6×3	39
3.8	Ma trận nhầm lẫn SVM + HOG 8×2	39
3.9	Các mẫu dự đoán sai tiêu biểu của mô hình KNN với đặc trưng HOG 8×2	40
3.10	Các mẫu dự đoán sai tiêu biểu của mô hình KNN với đặc trưng HOG 6×3	41

3.11	Các mẫu dự đoán sai tiêu biểu của mô hình KNN với đặc trưng LBP	42
3.12	Ma trận nhầm lẫn KNN + HOG 6x3	43
3.13	Ma trận nhầm lẫn KNN + HOG 8x2	43
3.14	Ma trận nhầm lẫn KNN + LBP	43
3.15	Các mẫu dự đoán sai tiêu biểu của mô hình RF với đặc trưng HOG 8x2	44
3.16	Các mẫu dự đoán sai tiêu biểu của mô hình RF với đặc trưng HOG 6x3	45
3.17	Các mẫu dự đoán sai tiêu biểu của mô hình RF với đặc trưng LBP	45
3.18	Biểu đồ Loss và Accuracy của mô hình CNN trong quá trình huấn luyện	47
3.19	Ma trận nhầm lẫn của mô hình CNN trên tập Test	47
3.20	Các mẫu dự đoán sai tiêu biểu của mô hình CNN	48
4.1	Giao diện ứng dụng Web Flask thực hiện phân loại ảnh tĩnh	51

DANH SÁCH BẢNG

0.1	Phân chia công việc và mức đóng góp của các thành viên trong nhóm	7
2.1	Tổng hợp các tập đặc trưng phục vụ thực nghiệm	20
2.2	Kiến trúc mạng CNN chi tiết dựa trên thực nghiệm	24
2.3	Kiến trúc và thông số tham số chi tiết của mô hình CNN tối ưu	31
3.1	Thống kê chi tiết số lượng hình ảnh trong bộ dữ liệu	33
3.2	Bảng tổng hợp thiết lập thông số và phương pháp tối ưu cho các mô hình	35
3.3	Kết quả mô hình SVM với các đặc trưng khác nhau	39
3.4	Kết quả mô hình KNN với các đặc trưng khác nhau	43
3.5	Kết quả mô hình Random Forest với các đặc trưng khác nhau	46
3.6	Bảng tổng kết độ chính xác của các mô hình thực nghiệm	49

LỜI NÓI ĐẦU

Lời đầu tiên, nhóm chúng tôi xin bày tỏ lòng biết ơn sâu sắc đến thầy **TS. Mai Tiến Dũng**, người đã tận tình hướng dẫn, định hướng và truyền đạt những kiến thức quý báu về lĩnh vực Thị giác máy tính trong suốt quá trình thực hiện đồ án này. Sự chỉ bảo tận tâm, những ý kiến đóng góp chuyên môn sắc sảo và sự khích lệ kịp thời của thầy đã là nguồn động lực to lớn, giúp chúng tôi vượt qua những khó khăn trong quá trình huấn luyện mô hình và hoàn thành báo cáo này một cách tốt nhất.

Chúng tôi cũng xin chân thành cảm ơn các bạn sinh viên lớp CS231.Q11 đã nhiệt tình chia sẻ kinh nghiệm học tập, hỗ trợ thảo luận về các kỹ thuật trích xuất đặc trưng và tối ưu hóa mô hình, cũng như đóng góp những ý kiến quý giá trong suốt quá trình thực hiện dự án.

Cuối cùng, nhóm chúng tôi xin gửi lời cảm ơn chân thành đến tất cả những ai đã đồng hành và hỗ trợ chúng tôi trong suốt quá trình học tập và nghiên cứu. Sự hỗ trợ của quý thầy cô và các bạn là yếu tố then chốt, quyết định sự thành công của đồ án này.

Chúng tôi vô cùng trân trọng và biết ơn những tình cảm và sự giúp đỡ quý báu đó! Rất mong nhận được sự góp ý từ quý thầy cô để nhóm có thể tiếp tục hoàn thiện và phát triển các ứng dụng thị giác máy tính tốt hơn trong tương lai.

Thành phố Hồ Chí Minh, Ngày 19 tháng 1 năm 2026

THÔNG TIN THÀNH VIÊN

Dưới đây là bảng thống kê chi tiết về vai trò, nhiệm vụ cụ thể và mức độ đóng góp của các thành viên trong nhóm đối với đồ án "Phân loại khuôn mặt đeo khẩu trang":

Họ và tên	MSSV	Đóng góp	Vai trò	Phần việc đảm nhận
Nguyễn Công Phát	23521143	35%	Nhóm trưởng	Thực hiện EDA, Trích xuất đặc trưng (HOG, LBP), thiết kế kiến trúc và tối ưu hóa hai mô hình SVM và KNN bằng Optuna, viết báo cáo
Nguyễn Lê Phong	23521168	35%	Thành viên	Tiền xử lý dữ liệu, xây dựng mô hình CNN, thực hiện đánh giá và so sánh kết quả các mô hình.
Vũ Việt Cương	23520213	30%	Thành viên	Thu thập dữ liệu, tối ưu hóa mô hình Random Forest bằng Optuna, thiết kế Slide.
<i>Tổng cộng: 100%.</i>				

Bảng 0.1: Phân chia công việc và mức đóng góp của các thành viên trong nhóm

Mức độ đóng góp trên đã được các thành viên trong nhóm thảo luận, thống nhất và đánh giá một cách khách quan dựa trên khối lượng công việc thực tế đã hoàn thành.

Chương 1

TỔNG QUAN ĐỀ TÀI

1.1 Mô tả bài toán

Bài toán đặt ra là xây dựng một hệ thống thị giác máy tính có khả năng nhận diện trạng thái đeo khẩu trang của con người thông qua hình ảnh hoặc video. Cụ thể, hệ thống sẽ tiếp nhận dữ liệu đầu vào là các khung hình khuôn mặt người, sau đó sử dụng các mô hình học máy và học sâu để phân tích, trích xuất đặc trưng và đưa ra kết luận về việc người đó có đang tuân thủ việc đeo khẩu trang hay không.

Về mặt kỹ thuật, đây là một bài toán **Phân loại nhị phân (Binary Classification)** trong lĩnh vực Thị giác máy tính. Dữ liệu đầu vào sau khi được xử lý sẽ được gán vào một trong hai nhãn chính:

- **With Mask:** Khuôn mặt có đeo khẩu trang hợp lệ.
- **Without Mask:** Khuôn mặt không đeo khẩu trang.

1.2 Lý do thực hiện bài toán

Việc thực hiện đề tài "Phân loại khuôn mặt đeo khẩu trang" xuất phát từ những nhu cầu thực tiễn và cấp bách trong bối cảnh hiện nay:

- **An toàn lao động và Sức khỏe cộng đồng:** Tại các môi trường đặc thù như nhà máy, xưởng sản xuất, bệnh viện hoặc trong các giai đoạn bùng phát dịch bệnh truyền nhiễm qua đường hô hấp, việc đeo khẩu trang là giải pháp cơ bản, tiết kiệm và hiệu quả nhất để bảo vệ sức khỏe con người.
- **Hạn chế của phương pháp thủ công:** Việc giám sát trực tiếp bằng con người tại các khu vực đông đúc thường gây tổn kém về nhân lực, dễ xảy ra sai sót do mệt mỏi và thiếu tính liên tục 24/7.
- **Tính ứng dụng cao của AI:** Với sự phát triển của Trí tuệ nhân tạo (AI), việc tích hợp các mô hình nhận diện vào hệ thống camera giám sát hiện có sẽ giúp tự động hóa quy trình kiểm soát, đưa ra cảnh báo kịp thời và nâng cao ý thức cộng đồng mà không cần sự can thiệp quá nhiều của con người.
- **Nghiên cứu khoa học:** Đề tài cho phép nhóm nghiên cứu sâu hơn về sự khác biệt giữa các phương pháp trích xuất đặc trưng truyền thống (như HOG, LBP) và các kiến trúc hiện đại (CNN), từ đó đánh giá được sự cân bằng giữa độ chính xác và tài nguyên tính toán.

1.3 Phát biểu bài toán

Bài toán phân loại khuôn mặt đeo khẩu trang được mô hình hóa dưới dạng bài toán học máy có giám sát (Supervised Learning) với các thành phần định nghĩa như sau:

Input (Đầu vào):

- Tập dữ liệu huấn luyện gồm N mẫu dữ liệu đã được gán nhãn:

$$\mathcal{D} = \{(x_i, y_i)\}_{i=1}^N$$

- Trong đó, mỗi mẫu dữ liệu x_i là một ảnh khuôn mặt đã được chuẩn hóa về kích thước 128×128 pixel. Tùy theo kiến trúc mô hình, không gian biểu diễn của x_i được quy định:
 - $x_i \in R^{128 \times 128}$ đối với các mô hình học máy truyền thống (SVM, KNN, Random Forest) sau khi chuyển đổi sang hệ ảnh xám (Grayscale).
 - $x_{i(CNN)} \in R^{128 \times 128 \times 1}$ đối với mô hình CNN, trong đó chiều cuối cùng đại diện cho kênh đơn sắc (Single channel), giúp mô hình tập trung vào đặc trưng hình thái thay vì thông tin màu sắc.
- Tập nhãn phân loại nhị phân: $\mathcal{Y} = \{0, 1\}$, với định nghĩa:
 - $y = 0$: *Without Mask* (Không đeo khẩu trang).
 - $y = 1$: *With Mask* (Có đeo khẩu trang).

Output (Đầu ra): Nhãn dự đoán $\hat{y} \in \mathcal{Y}$ tương ứng với một ảnh đầu vào x mới chưa xuất hiện trong tập huấn luyện.

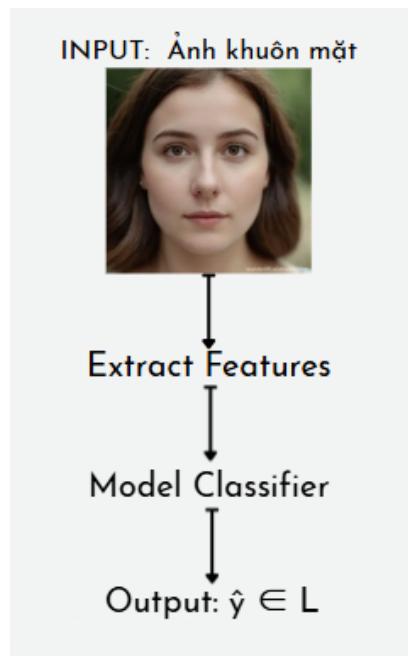
Mục tiêu kỹ thuật: Xây dựng hàm ánh xạ $f : \mathcal{X} \rightarrow \mathcal{Y}$ (với \mathcal{X} là không gian hình ảnh đầu vào) có bộ tham số θ . Quá trình huấn luyện thực hiện tối ưu hóa bộ tham số θ^* bằng cách cực tiểu hóa hàm mất mát (Loss function) trên toàn bộ tập dữ liệu:

$$\theta^* = \arg \min_{\theta} \frac{1}{N} \sum_{i=1}^N \mathcal{L}(f(x_i; \theta), y_i) \quad (1.1)$$

Trong đó \mathcal{L} là hàm mất mát (ví dụ: Binary Cross-Entropy cho CNN hoặc Hinge Loss cho SVM) đo lường sự sai khác giữa giá trị dự báo và nhãn thực tế.

1.4 Quy trình thực hiện tổng quát (Workflow)

Để giải quyết bài toán, nhóm đã xây dựng một quy trình thực hiện (Pipeline) khép kín từ khâu thu thập dữ liệu đến khâu đánh giá mô hình. Sơ đồ dưới đây mô tả các giai đoạn chính trong quá trình thực hiện đồ án:



Hình 1.1: Quy trình thực hiện tổng quát của hệ thống nhận diện khuôn mặt đeo khẩu trang

Dựa trên sơ đồ 1.1, quy trình bao gồm 4 giai đoạn chính: Tiền xử lý dữ liệu, Trích xuất đặc trưng (HOG, LBP hoặc tự động qua CNN), Huấn luyện các bộ phân loại và cuối cùng là Đánh giá hiệu năng dựa trên tập Test.

1.5 Mục tiêu của đề tài

Dựa trên những nhu cầu thực tiễn đã nêu, đồ án tập trung vào việc nghiên cứu và xây dựng một hệ thống nhận diện khuôn mặt đeo khẩu trang ổn định và chính xác. Các mục tiêu cụ thể bao gồm:

- **Xây dựng mô hình phân loại:** Nghiên cứu và triển khai mô hình có khả năng phân loại khuôn mặt người giữa hai trạng thái (Đeo khẩu trang và Không đeo khẩu trang). Mô hình này hướng tới việc tích hợp vào các hệ thống camera giám sát tại nhà máy, xưởng sản xuất để tự động phát hiện vi phạm về an toàn lao động.
- **Đánh giá và so sánh các kỹ thuật trích xuất đặc trưng:** Thực hiện khảo sát chi tiết hiệu quả của các kỹ thuật trích xuất đặc trưng thủ công truyền thống bao gồm:
 - **HOG (Histogram of Oriented Gradients):** Đánh giá khả năng bắt đặc trưng hình học và biên cạnh của khuôn mặt.
 - **LBP (Local Binary Patterns):** Đánh giá khả năng mô tả cấu trúc bề mặt và kết cấu của ảnh.
- **Khảo sát đa dạng mô hình học máy:** Thực hiện huấn luyện và so sánh hiệu năng của nhiều thuật toán khác nhau như Support Vector Machine (SVM), K-Nearest Neighbors (KNN), Random Forest kết hợp với tối ưu hóa siêu tham số bằng Optuna.

- **Nghiên cứu mô hình Deep Learning:** Xây dựng mạng Neural tích chập (CNN) để so sánh sức mạnh giữa việc tự động học đặc trưng (Deep Learning) và đặc trưng thủ công (Hand-crafted features).

Chương 2

PHƯƠNG PHÁP THỰC HIỆN BÀI TOÁN

Chương này trình bày chi tiết quy trình các phương pháp thực hiện mà nhóm áp dụng để giải quyết bài toán phân loại ảnh khuôn mặt đeo khẩu trang.

Cụ thể, nội dung chương tập trung vào các thành phần chính sau:

- **Tiền xử lý ảnh:** chuẩn hóa kích thước ảnh, chuẩn hóa giá trị điểm ảnh và chuyển đổi không gian màu phù hợp cho từng nhóm mô hình.
- **Trích xuất đặc trưng ảnh:** sử dụng các phương pháp trích xuất đặc trưng thủ công bao gồm Histogram of Oriented Gradients (HOG) và Local Binary Patterns (LBP).
- **Các mô hình phân loại:** triển khai và so sánh các mô hình học máy truyền thống như Support Vector Machine (SVM), K-Nearest Neighbors (KNN), Random Forest (RF), cùng với mô hình học sâu Convolutional Neural Network (CNN).
- **Tối ưu siêu tham số:** áp dụng các chiến lược tìm kiếm siêu tham số bằng công cụ Optuna cho các mô hình truyền thống và Keras Tuner cho mô hình CNN nhằm cải thiện hiệu năng phân loại.

2.1 Tiền xử lý ảnh

Nhằm đảm bảo tính nhất quán của dữ liệu đầu vào và phù hợp với đặc điểm của từng nhóm mô hình, nhóm áp dụng các bước tiền xử lý ảnh như sau.

2.1.1 Chuẩn hóa kích thước ảnh

Toàn bộ ảnh đầu vào được thay đổi kích thước về 128×128 pixel. Việc chuẩn hóa kích thước ảnh giúp đảm bảo tính đồng nhất của dữ liệu, đồng thời giảm chi phí tính toán trong quá trình trích xuất đặc trưng và huấn luyện mô hình.

2.1.2 Chuẩn hóa giá trị điểm ảnh

Sau khi thay đổi kích thước, giá trị các điểm ảnh được chuẩn hóa về khoảng $[0, 1]$ bằng cách chia cho giá trị cực đại 255. Bước này giúp cải thiện độ ổn định số học và hỗ trợ quá trình hội tụ của các thuật toán học máy và học sâu.

2.1.3 Chuyển đổi không gian màu

Trong nghiên cứu này, nhóm quyết định chuyển đổi toàn bộ dữ liệu hình ảnh sang không gian ảnh xám (Grayscale) cho tất cả các mô hình thực nghiệm, bao gồm cả nhóm học máy truyền thống và học sâu (CNN). Việc đồng nhất không gian màu này mang lại các lợi ích sau:

- **Giảm chi phí tính toán:** Chuyển từ ảnh RGB ($128 \times 128 \times 3$) sang ảnh Grayscale ($128 \times 128 \times 1$) giúp giảm lượng dữ liệu đầu vào xuống 3 lần, từ đó tối ưu hóa tốc độ huấn luyện và dự báo.
- **Tập trung vào đặc trưng hình thái:** Việc loại bỏ thông tin màu sắc giúp các mô hình (đặc biệt là HOG và CNN) tập trung trích xuất các đặc trưng về biên cạnh, độ tương phản cục bộ và cấu trúc hình học của khâu trang. Điều này giúp hệ thống hoạt động ổn định hơn trong các điều kiện ánh sáng khác nhau hoặc với các loại khâu trang có màu sắc đa dạng.
- **Tính nhất quán trong đánh giá:** Việc sử dụng cùng một loại dữ liệu đầu vào (ảnh xám) tạo ra một môi trường thử nghiệm công bằng để so sánh trực tiếp hiệu năng giữa các phương pháp trích xuất đặc trưng thủ công (HOG, LBP) và trích xuất đặc trưng tự động (CNN).

Các bước tiền xử lý trên được áp dụng nhất quán cho toàn bộ tập dữ liệu (Train, Validation, Test) trước khi đưa vào giai đoạn trích xuất đặc trưng và huấn luyện.

2.2 Trích xuất đặc trưng

Trong thị giác máy tính, trích xuất đặc trưng là bước chuyển đổi dữ liệu ảnh thành các vector số học đại diện cho các thông tin quan trọng như hình dáng, kết cấu hay biên cạnh. Nhóm đã thử nghiệm hai phương pháp trích xuất đặc trưng thủ công (Hand-crafted features) phổ biến nhất là HOG và LBP.

2.2.1 Đặc trưng HOG (Histogram of Oriented Gradients)

2.2.1.1 Định nghĩa

HOG là một bộ mô tả đặc trưng (feature descriptor) dùng để mô tả hình thái và cấu trúc (shape) của đối tượng thông qua sự phân bố của các hướng gradient. Phương pháp này cực kỳ hiệu quả trong việc nhận diện con người và khuôn mặt vì nó tập trung vào các đường biên và góc cạnh của đối tượng.

2.2.1.2 Nguyên lý hoạt động

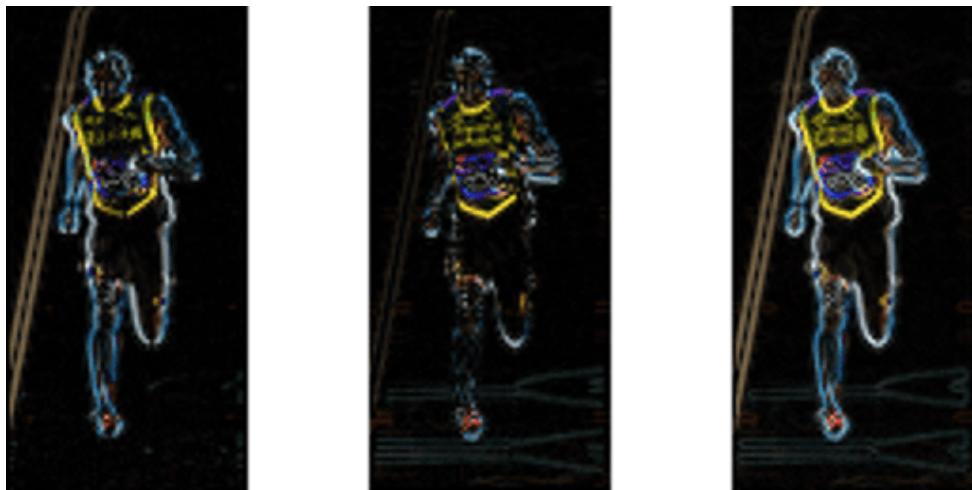
Quy trình tính toán HOG bao gồm các bước tuần tự nhằm chuyển đổi từ giá trị cường độ pixel sang một vector đặc trưng có tính phân biệt cao, tập trung vào cấu trúc hình học của khuôn mặt:

1. **Tiền xử lý (Preprocessing):** Ảnh đầu vào được chuẩn hóa về kích thước cố định là 128×128 pixel. Việc duy trì tỉ lệ khung hình (aspect ratio) ổn định giúp bộ mô tả HOG hoạt động đồng nhất, đảm bảo các đặc trưng hình học như mắt, mũi, miệng nằm ở các vị trí tương đồng nhất trên không gian vector.

2. **Tính toán Gradient:** Bước này loại bỏ các thông tin không thiết yếu và làm nổi bật các đường nét. Nhóm sử dụng toán tử Sobel để tính đạo hàm theo phương ngang (G_x) và phương dọc (G_y):

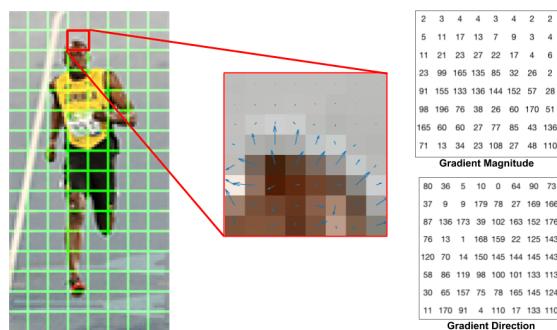
$$M(x, y) = \sqrt{G_x^2 + G_y^2}, \quad \theta(x, y) = \arctan\left(\frac{G_y}{G_x}\right) \quad (2.1)$$

Trong đó, M đại diện cho độ lớn gradient và θ là hướng của gradient (thường được giới hạn trong khoảng $0^\circ - 180^\circ$ - unsigned gradient).



Hình 2.1: Minh họa ảnh Gradient: G_x (trái), G_y (giữa) và Magnitude (phải)

3. **Tạo ô (Cell) và Histogram hướng:** Ảnh được chia thành các ô nhỏ 8×8 pixel. Trong mỗi ô, một histogram 9 bin (tương ứng các góc $0, 20, \dots, 160$ độ) được xây dựng. Mỗi pixel đóng góp vào bin tương ứng dựa trên độ lớn M . Việc gom nhóm theo ô giúp hệ thống mô tả được cấu trúc cục bộ và tăng tính bền vững trước nhiễu.

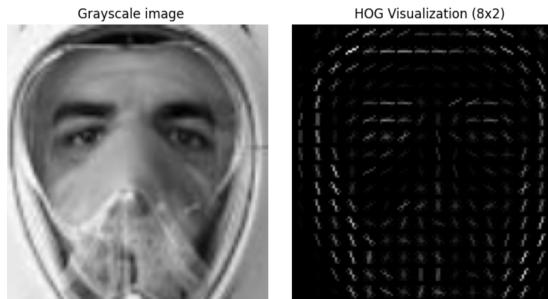


Hình 2.2: Trực quan hóa hướng Gradient bên trong một ô (cell) 8×8

4. **Chuẩn hóa khối (Block Normalization):** Để giảm thiểu ảnh hưởng của sự thay đổi ánh sáng, nhóm thực hiện chuẩn hóa trên các khối 16×16 (gồm 2×2 ô). Vector đặc trưng v của khối được chuẩn hóa theo chuẩn L_2 -norm:

$$v_{norm} = \frac{v}{\sqrt{\|v\|_2^2 + \epsilon^2}} \quad (2.2)$$

5. **Tạo vector đặc trưng cuối cùng:** Tất cả các vector đã chuẩn hóa được nối lại (concatenate) tạo thành một vector phẳng duy nhất. Hình 2.3 minh họa đặc trưng HOG trích xuất được trên khuôn mặt người.



Hình 2.3: Trực quan hóa bộ mô tả HOG trích xuất trên khuôn mặt

2.2.1.3 Tại sao lựa chọn đặc trưng HOG?

Trong bài toán phân loại khuôn mặt đeo khẩu trang, HOG mang lại những lợi thế quan trọng:

- **Mô tả biên dạng tối ưu:** Khẩu trang có cấu trúc hình học rõ ràng (viền cong, hình bầu dục). HOG tập trung vào các cạnh và hướng, do đó mô tả cực kỳ hiệu quả sự khác biệt giữa khuôn mặt trần và khuôn mặt có khẩu trang.
- **Bất biến với thay đổi ánh sáng:** HOG dựa trên hướng gradient (sự thay đổi cường độ tương đối) thay vì mức xám tuyệt đối, giúp mô hình hoạt động ổn định trong các môi trường chiếu sáng khác nhau.
- **Hiệu quả về không gian đặc trưng:** Thay vì sử dụng toàn bộ pixel, HOG nén thông tin vào các histogram, giúp giảm chi phí tính toán cho các bộ phân lớp phía sau như SVM hay KNN.

2.2.1.4 Cài đặt trong bài toán

Dựa trên các thực nghiệm thực tế, nhóm đã triển khai thuật toán HOG thông qua thư viện skimage.feature.hog. Điểm đặc biệt là nhóm tiến hành triển khai song song hai cấu hình đặc trưng để khảo sát sự tương quan giữa độ chi tiết của các ô (cells) và hiệu quả phân loại:

- **Cấu hình HOG 6x3:** Thiết lập $pixels_per_cell = (6, 3)$. Cấu hình này tăng cường độ phân giải theo chiều ngang, giúp bắt được các đặc trưng dẹt và kéo dài của khẩu trang một cách chi tiết hơn.
- **Cấu hình HOG 8x2:** Thiết lập $pixels_per_cell = (8, 2)$. Việc thay đổi tỉ lệ ô này nhằm khảo sát khả năng bao phủ các đường biên khuôn mặt khi không gian đặc trưng bị nén lại theo chiều dọc.

Các thông số kỹ thuật bổ trợ được thiết lập đồng nhất để đảm bảo tính khách quan:

- **Orientations = 9:** Chia không gian hướng thành 9 bins giúp bắt gọn các góc cạnh phức tạp.

- **Block Norm (L2-Hys):** Nhóm sử dụng phương pháp chuẩn hóa L2-Hys (L2-norm followed by clipping and renormalization). Kỹ thuật này giúp vector đặc trưng trở nên bền vững hơn trước các thay đổi về độ tương phản và nhiễu ánh sáng.

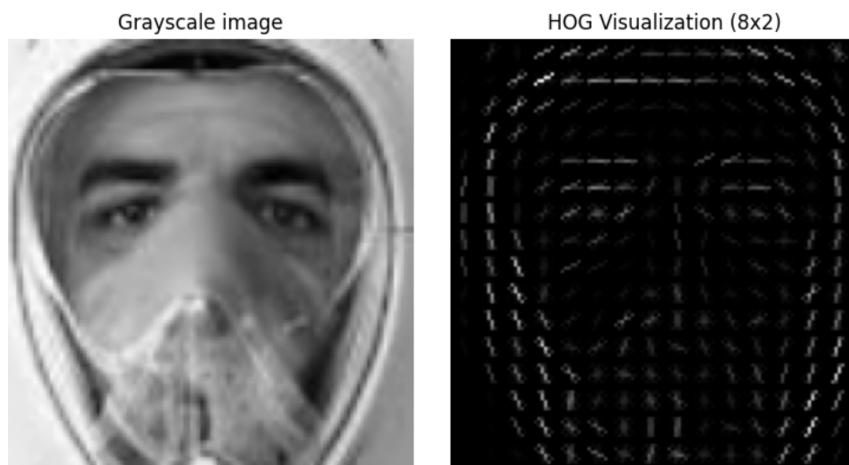
Quy trình trích xuất đặc trưng thực tế của hai cấu hình được minh họa như sau:

```
# HOG parameters (6x3)
HOG_PARAMS = {
    "orientations": 9,
    "pixels_per_cell": (6, 6),
    "cells_per_block": (3, 3),
    "block_norm": "L2-Hys"
}
```

Hình 2.4: Trực quan hóa HOG cấu hình 6x3

```
# HOG parameters (8x2)
HOG_PARAMS = {
    "orientations": 9,
    "pixels_per_cell": (8, 8),
    "cells_per_block": (2, 2),
    "block_norm": "L2-Hys"
}
```

Hình 2.5: Trực quan hóa HOG cấu hình 8x2



Hình 2.6: Minh họa kết quả trích xuất HOG trên khuôn mặt mẫu

Việc so sánh giữa hai cấu hình này là tiền đề để nhóm đánh giá độ nhạy của các bộ phân lớp (SVM, KNN, RF), từ đó xác định không gian đặc trưng tối ưu cho bài toán nhận diện khuôn mặt đeo khẩu trang.

2.2.1.5 Nhận xét về HOG

- **Ưu điểm:** Rất mạnh mẽ trong việc mô tả hình dáng (shape) và cấu trúc của khuôn mặt; bất biến đối với các phép biến đổi hình học nhỏ và điều kiện chiếu sáng sau khi đã chuẩn hóa khối.
- **Nhược điểm:** Chi phí tính toán khá lớn do phải tính toán gradient trên từng pixel; hiệu quả giảm sút nếu đối tượng bị che khuất quá nhiều hoặc xoay góc quá lớn.

2.2.2 Đặc trưng LBP (Local Binary Patterns)

2.2.2.1 Định nghĩa

Local Binary Patterns (LBP) là một bộ mô tả đặc trưng mạnh mẽ được sử dụng để phân tích kết cấu (texture) của bề mặt hình ảnh. Khác với HOG tập trung vào hình dáng (shape), LBP tập

trung vào mối quan hệ cường độ giữa một pixel và các lân cận của nó, giúp nhận diện các chi tiết nhỏ trên bề mặt khuôn mặt và khẩu trang.

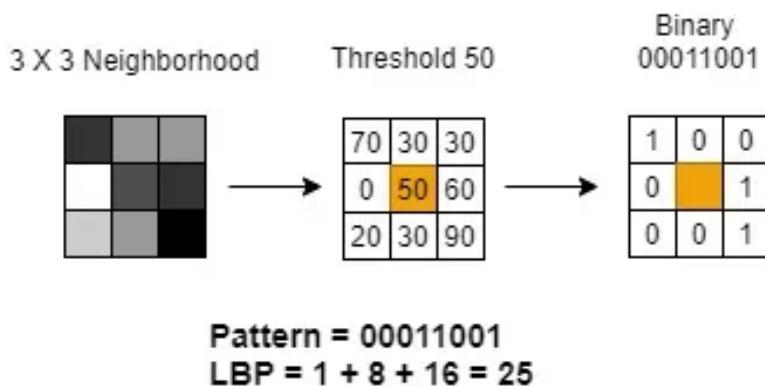
2.2.2.2 Nguyên lý hoạt động

Thuật toán LBP hoạt động dựa trên việc mô tả mối quan hệ cường độ sáng cục bộ. Quy trình tính toán mã LBP cho từng pixel và xây dựng vector đặc trưng được thực hiện qua các bước sau:

- Xác định vùng lân cận:** Với mỗi pixel trung tâm có cường độ g_c , ta xác định P điểm lân cận g_p nằm trên đường tròn có bán kính R . Việc sử dụng bán kính R cho phép thuật toán bắt được các đặc trưng kết cấu ở các quy mô (scales) khác nhau.
- Ngưỡng hóa cục bộ (Thresholding):** Thực hiện so sánh cường độ của pixel trung tâm với các pixel lân cận. Giá trị nhị phân $s(x)$ được xác định như sau:

$$s(g_p - g_c) = \begin{cases} 1 & \text{nếu } g_p \geq g_c \\ 0 & \text{nếu } g_p < g_c \end{cases} \quad (2.3)$$

Bước này giúp LBP bất biến với các thay đổi đơn sắc về độ sáng (gray-scale invariance).



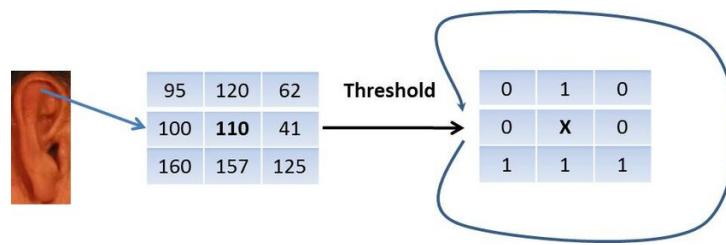
Hình 2.7: Quy trình ngưỡng hóa và tạo mã nhị phân LBP từ vùng lân cận 3×3

- Tạo mã thập phân LBP:** Chuỗi bit nhị phân thu được từ bước trên được chuyển đổi sang giá trị thập phân bằng cách nhân với trọng số 2^p :

$$LBP_{P,R} = \sum_{p=0}^{P-1} s(g_p - g_c) 2^p \quad (2.4)$$

Giá trị này đại diện cho loại kết cấu tại vị trí pixel đó (ví dụ: cạnh, góc, hoặc vùng phẳng).

- Xây dựng Histogram đặc trưng:** Sau khi tính toán mã LBP cho toàn bộ các pixel, nhóm xây dựng biểu đồ tần suất (Histogram) của các giá trị này. Histogram cuối cùng chính là vector đặc trưng đại diện cho kết cấu bề mặt của khuôn mặt, được sử dụng làm đầu vào cho mô hình phân loại.

**Hình 2.8:** Minh họa Histogram của các mẫu LBP trích xuất từ hình ảnh

2.2.2.3 Cài đặt trong bài toán

Trong đồ án này, nhóm triển khai thuật toán LBP thông qua thư viện skimage.feature với các tham số được tối ưu hóa cho bài toán nhận diện khuôn mặt đeo khẩu trang:

- **Cấu hình tham số:** Nhóm thiết lập bán kính $R = 1$ và số điểm lân cận $P = 8$. Đây là cấu hình chuẩn giúp mô tả chi tiết các biến đổi kết cấu nhỏ nhất trên bề mặt da và sợi vải của khẩu trang.
- **Phương pháp Uniform LBP:** Nhóm sử dụng biến thể "uniform" để trích xuất đặc trưng.
 - *Nguyên lý:* Chỉ những mẫu nhị phân có tối đa 2 lần chuyển đổi bit (0 sang 1 hoặc ngược lại) mới được giữ lại.
 - *Hiệu quả:* Kỹ thuật này giúp nén không gian đặc trưng một cách mạnh mẽ, giảm số chiều của histogram từ 256 xuống còn **59 bins**. Điều này không chỉ giúp mô hình học tập nhanh hơn mà còn loại bỏ được các nhiễu tần số cao, giữ lại các đặc trưng bền vững như cạnh, góc và các vùng phẳng.

Dưới đây là minh họa việc cài đặt mã nguồn và kết quả trích xuất đặc trưng LBP thực tế trong bài toán:

```

IMAGE_SIZE = (128, 128)

LBP_RADIUS = 1
LBP_POINTS = 8 * LBP_RADIUS
LBP_METHOD = "uniform"

```

Hình 2.9: Cài đặt thuật toán LBP và trích xuất Histogram trong Notebook



Hình 2.10: Trực quan hóa đặc trưng LBP trên khuôn mặt đeo khẩu trang (kết cấu bề mặt được làm nổi bật)

Việc sử dụng Uniform LBP giúp nhóm xây dựng được một bộ vector đặc trưng ổn định, là tiền đề quan trọng để so sánh hiệu năng với phương pháp HOG ở các bước tiếp theo.

2.2.2.4 Nhận xét về đặc trưng LBP

Thông qua quá trình thực nghiệm và đối chiếu với lý thuyết, nhóm đưa ra các đánh giá về ưu điểm và nhược điểm của bộ mô tả đặc trưng LBP như sau:

Ưu điểm:

- **Bền vững với biến đổi ánh sáng (Illumination Invariance):** Đây là ưu điểm lớn nhất của LBP. Do thuật toán chỉ dựa trên sự so sánh cường độ tương đối giữa các pixel lân cận, LBP có khả năng trích xuất đặc trưng kết cấu ổn định ngay cả khi ảnh có điều kiện chiếu sáng khác nhau.
- **Hiệu suất tính toán cao:** LBP có cấu trúc tính toán đơn giản, cho phép xử lý các tập dữ liệu lớn và đáp ứng tốt cho các ứng dụng nhận diện thời gian thực (real-time).
- **Tính phân biệt cao:** LBP cực kỳ hiệu quả trong việc mô tả các chi tiết bề mặt nhỏ, giúp phân biệt tốt các cấu trúc da mặt hoặc bề mặt khẩu trang khác nhau.
- **Bất biến với phép quay và tỉ lệ:** Trong các phiên bản cải tiến, LBP có khả năng giữ nguyên giá trị đặc trưng khi ảnh bị xoay hoặc thay đổi kích thước nhẹ.

Nhược điểm:

- **Nhạy cảm với nhiễu (Sensitivity to Noise):** Vì LBP so sánh trực tiếp cường độ pixel, các nhiễu hạt (noise) trong ảnh có thể làm sai lệch các giá trị nhị phân, dẫn đến sai sót trong histogram đặc trưng cuối cùng.
- **Thiếu thông tin toàn cục:** LBP chỉ tập trung vào các vùng lân cận cực nhỏ xung quanh pixel, do đó nó không thể bắt được các thông tin về hình dáng tổng thể (global structure) tốt như bộ mô tả HOG.
- **Hạn chế về hướng:** Mặc dù bất biến với phép quay, LBP không mã hóa được các thông tin về hướng của mẫu kết cấu, làm giảm khả năng phân biệt giữa các mẫu có cấu trúc giống nhau nhưng khác hướng.
- **Mất mát thông tin màu sắc:** LBP thường chỉ hoạt động trên ảnh mức xám, do đó các thông tin quan trọng về màu sắc của khẩu trang bị loại bỏ hoàn toàn.

2.2.3 Tổng hợp đặc trưng

Sau khi hoàn tất quá trình trích xuất bằng hai phương pháp HOG và LBP, nhóm tiến hành tổng hợp dữ liệu thành các bộ đặc trưng riêng biệt. Thay vì kết hợp (concatenate) các đặc trưng lại với nhau, nhóm lựa chọn hướng tiếp cận so sánh độc lập để đánh giá khách quan sức mạnh của từng loại đặc trưng đối với bài toán nhận diện khuôn mặt đeo khẩu trang.

Quy trình thực hiện cụ thể như sau:

- Xây dựng không gian đặc trưng HOG:** Nhóm tạo ra hai tập dữ liệu huấn luyện tương ứng với hai cấu hình 6×3 và 8×2 . Mỗi hình ảnh được chuyển đổi thành một vector phẳng (flattened vector) có số chiều phụ thuộc vào cấu hình cell và block. Đây là đầu vào chính cho các bộ phân lớp SVM, KNN và Random Forest.
- Xây dựng không gian đặc trưng LBP:** Toàn bộ tập dữ liệu được trích xuất sang dạng histogram Uniform LBP với 59 chiều. Tập dữ liệu này được dùng để đối soát hiệu năng với đặc trưng HOG, nhằm kiểm chứng xem kết cấu bề mặt (LBP) hay hình dáng biên cạnh (HOG) mang lại độ chính xác cao hơn.
- Đánh giá sự đánh đổi (Trade-off):** Nhóm thực hiện so sánh song song giữa các kích thước khối (6×3 và 8×2) không chỉ về độ chính xác (Accuracy) mà còn về thời gian trích xuất và kích thước vector đặc trưng. Mục tiêu là tìm ra điểm cân bằng tối ưu giữa việc giữ lại độ chi tiết của ảnh và tốc độ xử lý của hệ thống.

Bảng 2.1 tóm tắt các bộ đặc trưng đã được nhóm tổng hợp để đưa vào giai đoạn huấn luyện mô hình:

Bộ đặc trưng	Cấu hình chi tiết	Mục tiêu so sánh
HOG 6x3	Cell: (6, 3), Block: (2, 2), Ori: 9	Độ chi tiết theo chiều ngang
HOG 8x2	Cell: (8, 2), Block: (2, 2), Ori: 9	Độ nén đặc trưng theo chiều dọc
LBP Uniform	$P = 8, R = 1, 59$ bins	Đặc trưng kết cấu bề mặt

Bảng 2.1: Tổng hợp các tập đặc trưng phục vụ thực nghiệm

Việc chuẩn bị dữ liệu theo các nhánh độc lập như trên giúp nhóm dễ dàng thực hiện tối ưu hóa siêu tham số (Hyperparameter Tuning) cho từng mô hình cụ thể trong các bước tiếp theo.

2.3 Các mô hình phân loại

Sau khi trích xuất các bộ đặc trưng HOG và LBP, nhóm tiến hành thực nghiệm trên các mô hình học máy truyền thống (SVM, KNN, Random Forest) và mô hình học sâu hiện đại (CNN). Việc đưa vào mô hình CNN nhằm mục đích so sánh hiệu năng giữa phương pháp tự động học đặc trưng (End-to-End Learning) so với phương pháp trích xuất đặc trưng thủ công truyền thống (Hand-crafted features).

2.3.1 Mô hình Support Vector Machine (SVM)

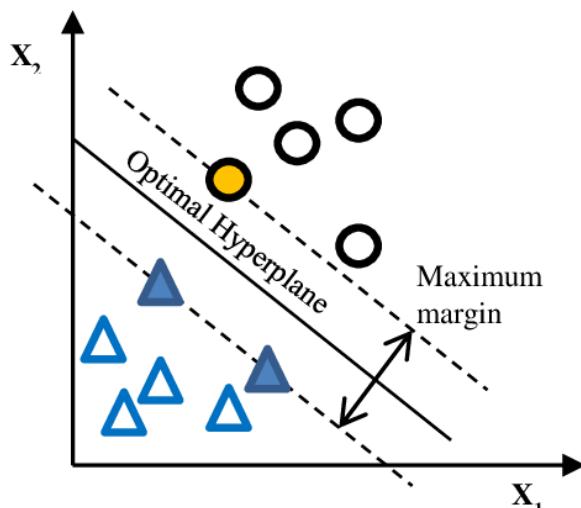
2.3.1.1 Nguồn gốc

SVM được giới thiệu lần đầu bởi Vladimir Vapnik và các đồng nghiệp vào năm 1992. Đây là một trong những thuật toán học máy có giám sát mạnh mẽ nhất, đặc biệt hiệu quả trong các bài

toán phân loại nhị phân trên không gian dữ liệu nhiều chiều.

2.3.1.2 Nguyên lý hoạt động

- **Tìm siêu phẳng tối ưu:** SVM tìm kiếm một siêu phẳng (hyperplane) trong không gian đặc trưng nhằm phân tách hai lớp dữ liệu sao cho khoảng cách (margin) từ siêu phẳng đến các điểm dữ liệu gần nhất là lớn nhất.
- **Hàm mục tiêu:** SVM tối ưu hóa bài toán cực tiểu hóa trọng số lề để đạt được siêu phẳng phân tách cực đại.
- **Kernel Trick:** Sử dụng các hàm nhân (Kernel) như RBF hoặc Linear giúp xử lý hiệu quả các tập dữ liệu phi tuyến bằng cách ánh xạ chúng sang không gian cao chiều hơn.
- **Dự đoán nhãn:** Nhãn dự đoán được xác định dựa trên vị trí của mẫu dữ liệu so với siêu phẳng tối ưu đã tìm được.



Hình 2.11: Minh họa siêu phẳng phân tách với lề cực đại (Maximum Margin) trong SVM

2.3.1.3 Cài đặt trong bài toán

Nhóm sử dụng thư viện `sklearn.svm.SVC` kết hợp với Kernel RBF. Các tham số trọng yếu như C và γ được tối ưu hóa tự động thông qua khung thử nghiệm Optuna nhằm tìm ra cấu hình phù hợp nhất cho các đặc trưng HOG và LBP đã trích xuất.

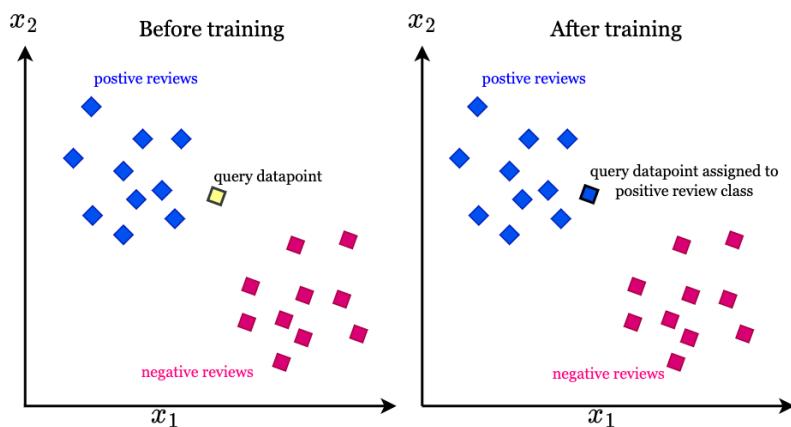
2.3.2 Mô hình K-Nearest Neighbors (KNN)

2.3.2.1 Nguồn gốc

KNN là một trong những thuật toán học máy đơn giản nhất, thuộc nhóm học lười (lazy learning), dựa trên các nghiên cứu về thống kê phi tham số.

2.3.2.2 Nguyên lý hoạt động

- **Tìm khoảng cách:** Thuật toán tính toán khoảng cách (phổ biến là Euclidean) từ điểm cần dự đoán đến tất cả các điểm trong tập huấn luyện.
- **Tìm K điểm gần nhất:** Xác định tập hợp k điểm lân cận có khoảng cách ngắn nhất trong không gian đặc trưng.
- **Dự đoán nhãn:** Kết quả cuối cùng được quyết định dựa trên cơ chế biểu quyết theo đa số (majority voting) từ các láng giềng đã chọn.



Hình 2.12: Quy trình phân loại dựa trên láng giềng gần nhất trong KNN

2.3.2.3 Cài đặt trong bài toán

Tham số số lượng láng giềng k được nhóm khảo sát trong khoảng từ 1 đến 31 và thực hiện tối ưu hóa bằng Optuna. Để tăng cường khả năng phân loại, nhóm áp dụng kỹ thuật trọng số theo khoảng cách (*distance weighting*), giúp các mẫu lân cận gần hơn có tầm ảnh hưởng lớn hơn đến kết quả dự báo.

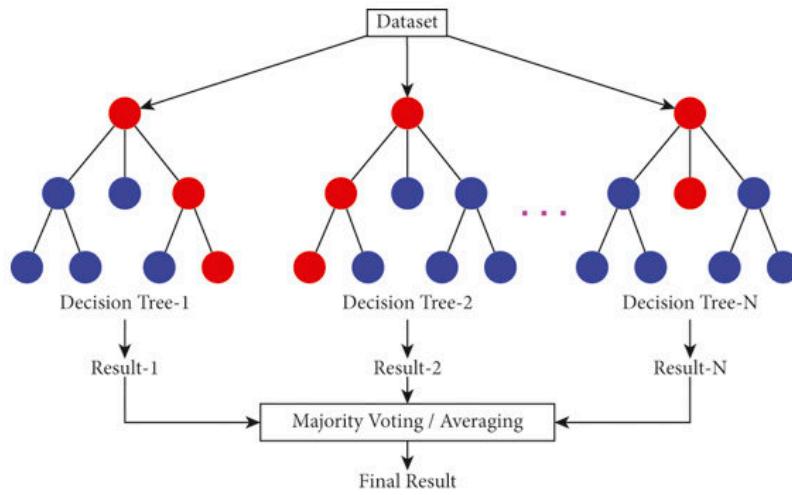
2.3.3 Mô hình Random Forest (RF)

2.3.3.1 Nguồn gốc

Random Forest được đề xuất bởi Leo Breiman vào năm 2001, hoạt động dựa trên nguyên lý kết hợp (Ensemble Learning) và kỹ thuật lấy mẫu Bagging.

2.3.3.2 Nguyên lý hoạt động

Mô hình xây dựng một hệ thống gồm nhiều cây quyết định độc lập. Trong quá trình dự báo, mỗi cây thành phần sẽ đưa ra một kết quả riêng lẻ, và nhãn cuối cùng của mô hình là giá trị chiếm đa số (Majority Voting). Hướng tiếp cận này giúp cải thiện tính ổn định và giảm thiểu hiện tượng quá khớp (Overfitting) thường gặp ở các cây quyết định đơn lẻ.



Hình 2.13: Cấu trúc mô hình Random Forest với sự kết hợp của nhiều cây quyết định

2.3.3.3 Cài đặt trong bài toán

Nhóm thiết lập phạm vi số lượng cây (*n_estimators*) từ 50 đến 300. Quá trình tinh chỉnh các tham số cấu trúc như độ sâu tối đa của cây (*max_depth*) và điều kiện tách nút (*min_samples_split*) được thực hiện thông qua Optuna để tối ưu khả năng học đặc trưng.

2.3.4 Mô hình Mạng Neural Tích chập (CNN)

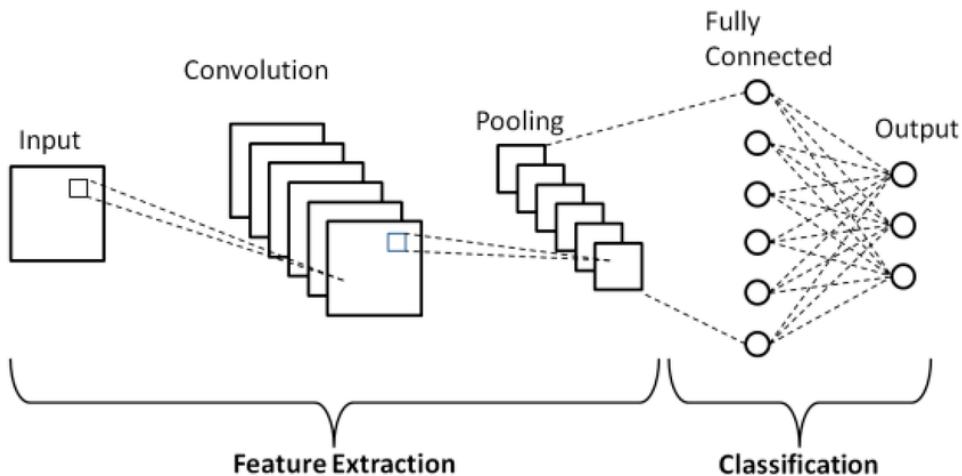
2.3.4.1 Nguồn gốc

CNN là một kiến trúc mạng nơ-ron học sâu được thiết kế chuyên biệt để xử lý dữ liệu có cấu trúc lưới như hình ảnh, lấy cảm hứng từ cơ chế hoạt động của vỏ não thị giác ở động vật.

2.3.4.2 Nguyên lý hoạt động

Mô hình CNN thực hiện việc học đặc trưng tự động thông qua các tầng chức năng:

- **Lớp Tích chập (Convolutional):** Áp dụng các bộ lọc để trích xuất thông tin không gian (từ các cạnh biên đến các kết cấu phức tạp).
- **Lớp Gộp (Pooling):** Giảm độ phân giải không gian của các bản đồ đặc trưng, giúp tăng tính bất biến với các phép dịch chuyển nhỏ và giảm khối lượng tính toán.
- **Lớp Kết nối đầy đủ (Fully Connected):** Phẳng hóa các đặc trưng thu được để đưa qua mạng nơ-ron truyền thống, thực hiện tính toán xác suất cho các nhãn lớp đầu ra.



Hình 2.14: Kiến trúc tổng quát của mạng CNN từ giai đoạn trích xuất đặc trưng đến phân loại

2.3.4.3 Cài đặt trong bài toán

Nhóm thiết kế kiến trúc CNN gồm 3 lớp Convolutional với số lượng filters là (32, 64, 64) kết hợp cùng các tầng Max Pooling tương ứng. Nhóm sử dụng công cụ Keras Tuner với thuật toán Hyperband để tìm kiếm cấu hình tối ưu về số lượng nơ-ron tại tầng ẩn và tỷ lệ Dropout. Đặc biệt, mô hình sử dụng đầu vào là ảnh xám (1 kênh) để tối ưu hiệu năng tính toán.

Chi tiết kiến trúc và số lượng tham số của mô hình CNN (với cấu hình tiêu biểu) được trình bày trong Bảng 2.2.

Layer (Type)	Output Shape	Kernel	Parameters
Input (Grayscale)	$128 \times 128 \times 1$	-	0
Conv2D + ReLU	$126 \times 126 \times 32$	3×3	320
MaxPooling2D	$63 \times 63 \times 32$	2×2	0
Conv2D + ReLU	$61 \times 61 \times 64$	3×3	18,496
MaxPooling2D	$30 \times 30 \times 64$	2×2	0
Conv2D + ReLU	$28 \times 28 \times 64$	3×3	36,928
MaxPooling2D	$14 \times 14 \times 64$	2×2	0
Flatten	12544	-	0
Dense + ReLU	128	-	1,605,760
Dropout (0.5)	128	-	0
Dense (Output)	2	-	258
Total Params	1,661,762 (6.34 MB)		

Bảng 2.2: Kiến trúc mạng CNN chi tiết dựa trên thực nghiệm

Có thể thấy rằng phần lớn tham số tập trung ở tầng Dense (kết nối toàn bộ), chiếm hơn 96% tổng số tham số. Các tầng Convolutional đóng vai trò trích xuất đặc trưng không gian nhưng có lượng tham số rất nhỏ, giúp mô hình giữ được sự gọn nhẹ và giảm thiểu Overfitting trên tập dữ liệu ảnh Grayscale.

2.4 Phương pháp tối ưu siêu tham số (Optuna & Keras Tuner)

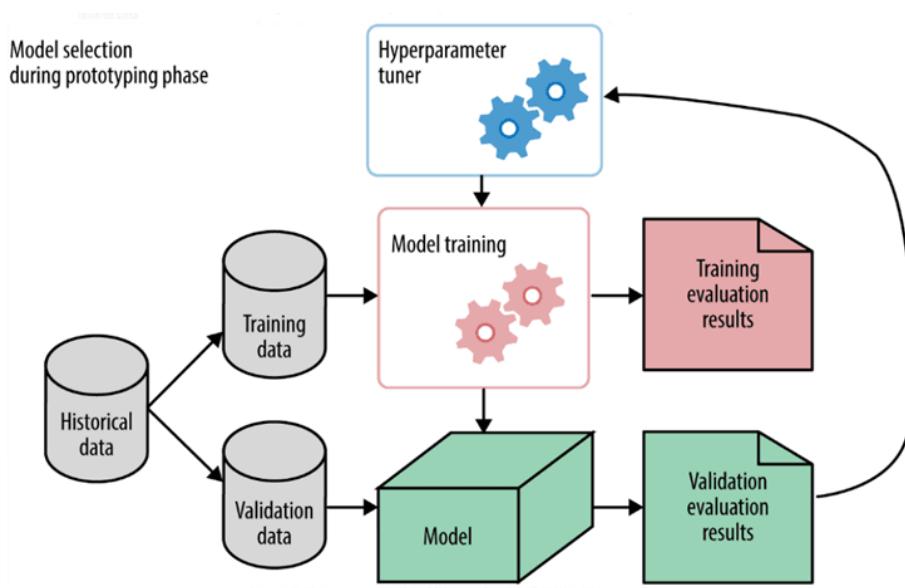
Để đảm bảo các mô hình đạt được hiệu suất phân loại tốt nhất, nhóm không lựa chọn các tham số mô hình một cách thủ công mà sử dụng các công cụ tối ưu hóa tự động. Cách tiếp cận này giúp tìm ra sự kết hợp tối ưu giữa các đặc trưng (HOG, LBP) và các bộ phân loại.

2.4.1 Optuna

2.4.1.1 Nguyên lý hoạt động

Optuna là một framework tối ưu hóa siêu tham số tự động mã nguồn mở. Cơ chế cốt lõi của Optuna dựa trên:

- **Bayesian Optimization:** Sử dụng sampler TPE (Tree-structured Parzen Estimator) để dự đoán vùng không gian tham số có khả năng mang lại kết quả tốt nhất dựa trên lịch sử các lần thử trước.
- **Cơ chế Pruning:** Áp dụng Median Pruner để dừng sớm các thử nghiệm (trials) không tiềm năng, giúp tiết kiệm đáng kể thời gian huấn luyện.



Hình 2.15: Quy trình lựa chọn mô hình và tối ưu siêu tham số trong giai đoạn thử nghiệm 3

2.4.1.2 Nhận xét chung về Optuna

- **Ưu điểm:** Tốc độ tìm kiếm nhanh, hiệu quả vượt trội so với Grid Search truyền thống, có khả năng khám phá không gian tham số phức tạp.
- **Nhược điểm:** Việc cài đặt hàm mục tiêu (*objective function*) đòi hỏi kiến thức lập trình và hiểu biết về mô hình.

2.4.1.3 Ứng dụng

Nhóm sử dụng Optuna để tinh chỉnh siêu tham số cho các mô hình SVM, Random Forest và KNN trên cả hai tập đặc trưng HOG và LBP.

2.4.1.4 Cài đặt Optuna cho SVM

Nhóm thiết lập không gian tìm kiếm cho tham số C (từ 10^{-2} đến 10^2), loại kernel (Linear, RBF, Poly) và tham số γ .

- **Đối với đặc trưng HOG:** Tập trung tối ưu các tham số chính nhằm giảm thời gian huấn luyện.
- **Đối với đặc trưng LBP:** Áp dụng tối ưu có điều kiện (Conditional tuning), chỉ tìm tham số $degree$ khi kernel là *poly*.

```
def objective(trial):
    params = {
        "C": trial.suggest_float("C", 1e-2, 1e2, log=True),
        "kernel": trial.suggest_categorical("kernel", ["linear", "rbf"]),
        "gamma": trial.suggest_categorical("gamma", ["scale", "auto"]),
        "probability": False, # Set probability for speed-up
        "random_state": 42,
    }

    clf = SVC(**params)
    clf.fit(X_train, y_train_enc)
    pred = clf.predict(X_val)
    return accuracy_score(y_val_enc, pred)

study = optuna.create_study(
    direction="maximize",
    pruner=MedianPruner(n_startup_trials=5),
    sampler=TPESampler(multivariate=True),
)

study.optimize(objective, n_trials=30, n_jobs=1)

print("Best validation accuracy:", study.best_value)
print("Best params:", study.best_params)
```

Hình 2.16: Code snippet minh họa quá trình cài đặt Optuna cho HOG + SVM

```
def objective(trial):
    kernel = trial.suggest_categorical("kernel", ["linear", "poly", "rbf"])
    C = trial.suggest_categorical("C", [1, 5, 10])

    if kernel == "poly":
        gamma = trial.suggest_float("gamma_poly", 0.01, 1.0)
        degree = trial.suggest_int("degree", 3, 15)
    else:
        gamma = trial.suggest_categorical("gamma", ["scale", "auto"])

    model = SVC(kernel=kernel, C=C, gamma=gamma)
    model.fit(X_train, y_train)
    pred = model.predict(X_val)
    return accuracy_score(y_val, pred)

study = optuna.create_study(direction="maximize")
study.optimize(objective, n_trials=50)
```

Hình 2.17: Code snippet minh họa quá trình tối ưu có điều kiện cho LBP + SVM

2.4.1.5 Cài đặt Optuna cho Random Forest

- **Với đặc trưng HOG:** Sử dụng kỹ thuật *Warm Start* kết hợp *Pruning* lặp lại qua các mức *n_estimators* (50, 100, 150, 200) để loại bỏ sớm các cấu hình kém.
- **Với đặc trưng LBP:** Do vector đặc trưng có kích thước nhỏ, nhóm sử dụng không gian tìm kiếm tiêu chuẩn cho *max_depth* và *min_samples_split*.

```

def objective(trial):
    params = {
        'max_depth': trial.suggest_int('max_depth', 5, 50),
        'min_samples_split': trial.suggest_int('min_samples_split', 2, 20),
        'min_samples_leaf': trial.suggest_int('min_samples_leaf', 1, 10),
        'max_features': trial.suggest_categorical('max_features', ['sqrt','log2']),
        'warm_start': True, 'n_jobs': 1, 'random_state': 42
    }
    clf = RandomForestClassifier(**params)
    for n in [50, 100, 150, 200]:
        clf.set_params(n_estimators=n)
        clf.fit(X_train, y_train)
        acc = accuracy_score(y_val, clf.predict(X_val))
        trial.report(acc, n)
        if trial.should_prune(): raise optuna.TrialPruned()
    return acc

```

Hình 2.18: Code snippet minh họa quá trình tối ưu Random Forest với Pruning cho đặc trưng HOG

```

def objective(trial):
    params = {
        'n_estimators': trial.suggest_int('n_estimators', 50, 300),
        'max_depth': trial.suggest_int('max_depth', 5, 50),
        'min_samples_split': trial.suggest_int('min_samples_split', 2, 20),
        'min_samples_leaf': trial.suggest_int('min_samples_leaf', 1, 10),
        'max_features': trial.suggest_categorical('max_features',[ 'sqrt','log2',None]),
        'n_jobs': -1, 'random_state': 42
    }
    clf = RandomForestClassifier(**params)
    clf.fit(X_train, y_train)
    return accuracy_score(y_val, clf.predict(X_val))

```

Hình 2.19: Code snippet minh họa quá trình tối ưu Random Forest tiêu chuẩn cho đặc trưng LBP

2.4.1.6 Cài đặt Optuna cho KNN

Không gian tham số cho KNN bao gồm *n_neighbors* (từ 1 đến 31, bước nhảy 2) và tham số *weights* (uniform hoặc distance).

- **Với HOG:** Điều chỉnh thêm chuẩn khoảng cách p để phù hợp với không gian đặc trưng nhiều chiều.
- **Với LBP:** Tập trung tìm số lảng giềng tối ưu nhằm phân biệt kết cấu da mặt và khẩu trang.

```

def objective(trial):
    # Khởi tạo tham số cho KNN
    params = {
        # Số láng giềng, từ 1 đến 31 để tránh tie
        "n_neighbors": trial.suggest_int("n_neighbors", 1, 31, step=2),
        # uniform: mô hình xem xét nhau, distance: căn bậc hai của tổng số điểm
        "weights": trial.suggest_categorical("weights", ["uniform", "distance"]),
        # p = 1: Manhattan, p = 2: Euclidean
        "p": trial.suggest_int("p", 1, 2),
        # metric là métric minkowski (đồng chung với p)
        "metric": "minkowski",
    }

    clf = KNeighborsClassifier(**params)
    clf.fit(X_train, y_train_enc)

    y_val_pred = clf.predict(X_val)
    acc = accuracy_score(y_val_enc, y_val_pred)
    return acc

study_knn = optuna.create_study(
    direction="maximize",
    pruner=MedianPruner(n_startup_trials=5, n_warmup_steps=1),
    sampler=TPESampler(multivariate=True),
)

study_knn.optimize(objective, n_trials=30, n_jobs=1)

print("[KNN] Best validation accuracy:", study_knn.best_value)
print("[KNN] Best params:", study_knn.best_params)

```

Hình 2.20: Code snippet minh họa quá trình tối ưu KNN trên đặc trưng HOG

```

def objective(trial):
    # Khởi tạo tham số cho KNN
    params = {
        # Số láng giềng 1 đến 31 để tránh tie
        "n_neighbors": trial.suggest_int("n_neighbors", 1, 31, step=2),
        # uniform: mô hình xem xét nhau, distance: căn bậc hai của tổng số điểm
        "weights": trial.suggest_categorical("weights", ["uniform", "distance"]),
        # p = 1: Manhattan, p = 2: Euclidean
        "p": trial.suggest_int("p", 1, 2),
        # metric là métric minkowski (đồng chung với p)
        "metric": "minkowski",
    }

    clf = KNeighborsClassifier(**params)
    clf.fit(X_train, y_train_enc)

    y_val_pred = clf.predict(X_val)
    acc = accuracy_score(y_val_enc, y_val_pred)
    return acc

study_knn = optuna.create_study(
    direction="maximize",
    pruner=MedianPruner(n_startup_trials=5, n_warmup_steps=1),
    sampler=TPESampler(multivariate=True, seed=42),
)

study_knn.optimize(objective, n_trials=30, n_jobs=1)

```

Hình 2.21: Code snippet minh họa quá trình tối ưu KNN trên đặc trưng LBP

2.4.2 Keras Tuner

2.4.2.1 Nguyên lý hoạt động

Keras Tuner là thư viện chuyên dụng để tìm kiếm cấu hình mạng Neural tối ưu. Nhóm sử dụng thuật toán **Hyperband**, một phương pháp dựa trên cơ chế dừng sớm (Early Stopping) nhằm nhanh chóng hội tụ về các kiến trúc tiềm năng.

2.4.2.2 Nhận xét chung về Keras Tuner

- **Ưu điểm:** Tự động hóa việc thiết kế kiến trúc mạng (số filter, số lớp ẩn), đặc biệt hiệu quả đối với các mô hình Deep Learning.
- **Nhược điểm:** Yêu cầu tài nguyên tính toán lớn (GPU/RAM) khi huấn luyện đồng thời nhiều mô hình CNN.

2.4.2.3 Ứng dụng

Keras Tuner được áp dụng để tìm cấu hình CNN tối ưu bằng cách thay đổi số lượng filter, số neuron ở lớp Dense, tỉ lệ Dropout và Learning rate.

2.4.2.4 Cấu hình Keras Tuner cho CNN

Để tìm ra kiến trúc tối ưu nhất cho bài toán phân loại ảnh xám, nhóm sử dụng thư viện Keras Tuner với thuật toán tìm kiếm Hyperband. Quá trình tối ưu hóa tập trung vào việc tinh chỉnh số lượng bộ lọc (filters), số lượng nơ-ron lớp ẩn và tốc độ học (learning rate).

Dưới đây là cấu trúc logic của hàm xây dựng mô hình (*build_model*) được sử dụng trong quá trình tìm kiếm:

Listing 2.1: Pseudocode xây dựng kiến trúc CNN và tối ưu siêu tham số bằng Keras Tuner

```
def build_model(hp):
    model = Sequential()

    # Lop dau vao: Anh xam Grayscale 128x128x1
    model.add(Input(shape=(128, 128, 1)))

    # Lop Convolution thu nhat: toi uu so filters
    hp_filters = hp.Int(
        name='filters_1',
        min_value=32,
        max_value=96,
        step=32
    )
    model.add(Conv2D(hp_filters, (3, 3), activation='relu'))
    model.add(MaxPooling2D(pool_size=(2, 2)))

    # Lop Convolution thu hai
    model.add(Conv2D(64, (3, 3), activation='relu'))
    model.add(MaxPooling2D(pool_size=(2, 2)))

    # Lop Convolution thu ba
    model.add(Conv2D(64, (3, 3), activation='relu'))
    model.add(MaxPooling2D(pool_size=(2, 2)))

    # Chuyen doi dac trung sang vector 1 chieu
    model.add(Flatten())
```

```

# Lop Fully Connected:toi uu so neuron
hp_units = hp.Int(
    name='dense_units',
    min_value=64,
    max_value=256,
    step=64
)
model.add(Dense(hp_units, activation='relu'))

# Toi uu Dropout de giam overfitting
hp_dropout = hp.Float(
    name='dropout',
    min_value=0.2,
    max_value=0.5,
    step=0.1
)
model.add(Dropout(hp_dropout))

# Lop dau ra cho bai toan phan loai nhi phan
model.add(Dense(2, activation='softmax'))

# Toi uu learning rate
hp_lr = hp.Choice(
    name='learning_rate',
    values=[1e-2, 1e-3, 1e-4]
)

model.compile(
    optimizer=Adam(learning_rate=hp_lr),
    loss='sparse_categorical_crossentropy',
    metrics=['accuracy']
)

return model

```

Đoạn mã trên mang tính minh họa quy trình xây dựng kiến trúc và tối ưu siêu tham số, không phải mã nguồn huấn luyện đầy đủ được sử dụng trong thực nghiệm.

Sau quá trình thực nghiệm với 30 thử nghiệm (trials), bộ siêu tham số tốt nhất đã được xác định. Bảng 2.3 trình bày chi tiết kiến trúc của mô hình CNN cuối cùng (với giả định cấu hình tối ưu đạt được):

Lớp (Loại)	Kích thước đầu ra	Kích thước nhân	Tham số
InputLayer	(128, 128, 1)	-	0
Conv2D (ReLU)	(126, 126, 32)	3×3	320
MaxPooling2D	(63, 63, 32)	2×2	0
Conv2D (ReLU)	(61, 61, 64)	3×3	18,496
MaxPooling2D	(30, 30, 64)	2×2	0
Conv2D (ReLU)	(28, 28, 64)	3×3	36,928
MaxPooling2D	(14, 14, 64)	2×2	0
Flatten	12544	-	0
Dense (ReLU)	128	-	1,605,760
Dropout	128	-	0
Dense (Softmax)	2	-	258
Tổng số tham số (Total Params)			1,661,762
Tham số có thể huấn luyện (Trainable Params)			1,661,762

Bảng 2.3: Kiến trúc và thông số tham số chi tiết của mô hình CNN tối ưu

Nhận xét: Với việc chuyển sang ảnh xám, mô hình giảm thiểu được số lượng tham số đáng kể ở lớp tích chập đầu tiên (chỉ 320 tham số thay vì gần 1,000 như ảnh màu). Phần lớn dung lượng mô hình tập trung ở lớp Dense đầu tiên, chiếm khoảng 1.6 triệu tham số, giúp mạng có khả năng ghi nhớ các đặc trưng phức tạp để phân loại chính xác giữa hai lớp có và không có khẩu trang.

Chương 3

THỰC NGHIỆM VÀ KẾT QUẢ

Chương này trình bày chi tiết về quá trình thực nghiệm, các thông số thiết lập hệ thống, các độ đo đánh giá và phân tích kết quả thu được từ tổng hợp các phương pháp đã triển khai.

3.1 Bộ dữ liệu (Dataset)

3.1.1 Tổng quan bộ dữ liệu

Trong đồ án này, nhóm sử dụng bộ dữ liệu [Face Mask 12k Images Dataset](#) được cung cấp trên nền tảng Kaggle. Đây là một bộ dữ liệu lớn, có độ đa dạng cao về đặc điểm khuôn mặt, góc chụp và điều kiện môi trường, giúp mô hình đạt được khả năng tổng quát hóa tốt.



Hình 3.1: Các mẫu hình ảnh thực tế trong bộ dữ liệu Face Mask 12k Images

3.1.2 Cấu trúc bộ dữ liệu

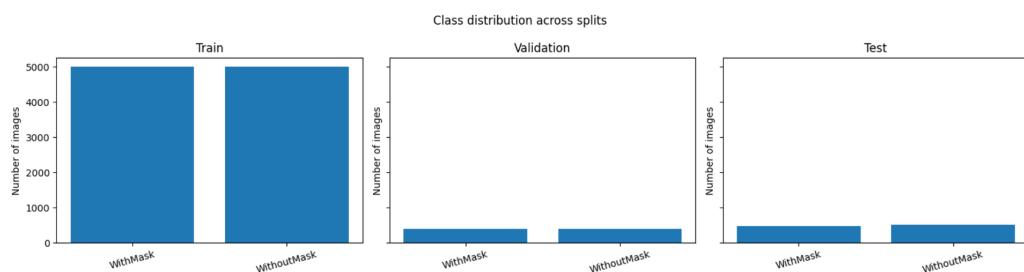
Bộ dữ liệu bao gồm tổng cộng gần 12,000 hình ảnh màu (RGB) đã được phân loại sẵn vào hai nhãn: *With Mask* và *Without Mask*. Nhóm tuân thủ cấu trúc chia dữ liệu có sẵn của bộ dataset để đảm bảo tính khách quan khi so sánh hiệu năng:

- **Train set (10,000 ảnh):** Được sử dụng để huấn luyện trọng số cho các mô hình học máy và học sâu.
- **Validation set (800 ảnh):** Được sử dụng trong quá trình tối ưu hóa siêu tham số bằng Optuna và Keras Tuner để tránh hiện tượng quá khớp (Overfitting).
- **Test set (992 ảnh):** Được sử dụng để đánh giá hiệu năng cuối cùng của các mô hình trên dữ liệu chưa từng xuất hiện trong quá trình huấn luyện.

Chi tiết về số lượng mẫu tại tập Test được trình bày trong bảng 3.1.

Nhãn (Label)	Tập Train	Tập Validation	Tập Test
With Mask	5,000	400	483
Without Mask	5,000	400	509
Tổng cộng	10,000	800	992

Bảng 3.1: Thống kê chi tiết số lượng hình ảnh trong bộ dữ liệu



Hình 3.2: Tỷ lệ phân chia dữ liệu giữa các tập Train, Validation và Test

3.2 Độ đo đánh giá hiệu năng

Để đánh giá mô hình một cách toàn diện và khách quan, nhóm sử dụng các độ đo dựa trên Ma trận nhầm lẫn (Confusion Matrix). Các giá trị cơ bản bao gồm:

- **TP (True Positive):** Số mẫu có khẩu trang được dự đoán đúng là có khẩu trang.
- **TN (True Negative):** Số mẫu không khẩu trang được dự đoán đúng là không khẩu trang.
- **FP (False Positive):** Số mẫu không khẩu trang bị dự đoán sai là có khẩu trang.
- **FN (False Negative):** Số mẫu có khẩu trang bị dự đoán sai là không khẩu trang.

Các công thức tính toán cụ thể cho từng độ đo như sau:

- **Accuracy (Độ chính xác tổng quát):** Tỷ lệ giữa tổng số điểm dữ liệu được dự đoán đúng trên tổng số dữ liệu của tập kiểm tra.

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \quad (3.1)$$

- **Precision (Độ chính xác trên lớp dự đoán):** Đại diện cho khả năng mô hình dự đoán chính xác các trường hợp thực sự là dương tính (đeo khẩu trang) trong số tất cả các trường hợp mà mô hình đã gán nhãn là dương tính.

$$\text{Precision} = \frac{TP}{TP + FP} \quad (3.2)$$

- **Recall (Độ nhạy):** Tỷ lệ các trường hợp thực sự đeo khẩu trang mà mô hình đã phát hiện đúng, tránh bỏ sót các trường hợp vi phạm.

$$\text{Recall} = \frac{TP}{TP + FN} \quad (3.3)$$

- **F1-score:** Là giá trị trung bình điều hòa (Harmonic mean) giữa Precision và Recall. Chỉ số này cực kỳ quan trọng khi tập dữ liệu có sự mất cân bằng giữa hai nhãn.

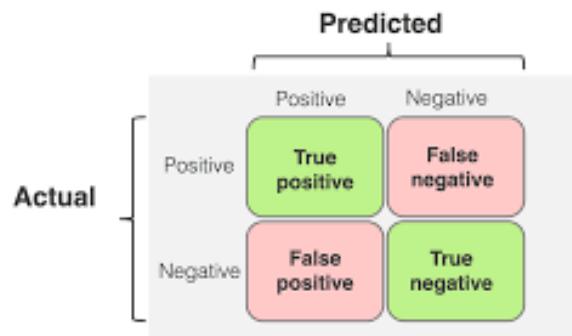
$$\text{F1-score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \quad (3.4)$$

- **Macro Average:** Trung bình cộng đơn giản của các chỉ số (Precision, Recall hoặc F1) tính riêng cho từng lớp. Độ đo này coi trọng vai trò của nhãn *With Mask* và *Without Mask* là ngang nhau.

$$\text{Macro Avg} = \frac{1}{N} \sum_{i=1}^N \text{Score}_i \quad (3.5)$$

- **Weighted Average:** Trung bình cộng các chỉ số có tính đến trọng số là số lượng mẫu (support) của mỗi lớp. Chỉ số này phản ánh hiệu năng thực tế dựa trên phân bố của dữ liệu tập Test.

$$\text{Weighted Avg} = \frac{\sum (\text{Score}_i \times \text{Support}_i)}{\sum \text{Support}_i} \quad (3.6)$$



Hình 3.3: Cấu trúc Ma trận nhầm lẫn dùng để tính toán các độ đo hiệu năng

3.3 Thiết lập thực nghiệm

Để đảm bảo tính công bằng và khả năng tái lập kết quả, nhóm thống nhất các thiết lập thực nghiệm chung bao gồm:

- Môi trường:** Nhóm thực hiện triển khai song song trên nền tảng Kaggle Kernel (tận dụng GPU Cloud) và môi trường Local (Sử dụng máy tính cá nhân) với giao diện Jupyter Notebook, sử dụng ngôn ngữ lập trình Python và các thư viện chuyên dụng như Scikit-learn, TensorFlow/Keras.
- Random seed:** Thiết lập cố định giá trị 42 để đảm bảo các phép chia dữ liệu và khởi tạo trọng số không thay đổi qua các lần thực nghiệm.
- Lưu trữ:** Sau mỗi quá trình huấn luyện, mô hình có kết quả tốt nhất trên tập Validation được lưu tự động vào thư mục `models/` dưới định dạng `.joblib` (SVM, KNN, RF) hoặc `.keras` (CNN).

Bảng 3.2 trình bày chi tiết các thông số kỹ thuật được thiết lập cho từng cặp đặc trưng và mô hình trong đồ án:

Mô hình	Đặc trưng	Phương pháp tối ưu	Thông số thiết lập chi tiết
SVM	HOG (6x3, 8x2)	Optuna (TPE)	Kernel: 'rbf', C: [0.1, 100], Gamma: ['scale', 'auto'], Block norm: L2-Hys, orientations: 9.
	LBP		Kernel: 'rbf', C: [0.1, 100], Gamma: 'scale', LBP Method: 'uniform', Radius: 1, Points: 8.
KNN	HOG (6x3, 8x2)	Optuna (TPE)	k: [1, 31], Weights: ['uniform', 'distance'], Metric: 'minkowski' (p: [1, 2]).
	LBP		k: [1, 31], Weights: ['uniform', 'distance'], Metric: 'minkowski' (p: [1, 2]).
Random Forest	HOG (6x3, 8x2)	Optuna (TPE)	n_estimators: [50, 200], Max_depth: [5, 50], Min_samples_split: [2, 20], Warm_start: True.
	LBP		n_estimators: [50, 300], Max_depth: [5, 50], Min_samples_leaf: [1, 10], Max_features: ['sqrt', 'log2', None].
CNN	(Tự học)	Keras Tuner	Cấu trúc: 3 lớp Conv2D (filters: [32, 96]), Hidden: [64, 256], Dropout: [0.2, 0.5], Optimizer: Adam (lr: [1e-2, 1e-4]).

Bảng 3.2: Bảng tổng hợp thiết lập thông số và phương pháp tối ưu cho các mô hình

3.4 Kết quả thực nghiệm

3.4.1 Mô hình Support Vector Machine (SVM)

Mô hình SVM được đánh giá là bộ phân lớp truyền thống mạnh mẽ nhất trong các thực nghiệm của nhóm, đặc biệt khi kết hợp với bộ mô tả đặc trưng HOG.

3.4.1.1 Đặc trưng HOG 8x2

Đây là cấu hình đạt hiệu suất cao nhất trong nhóm mô hình học máy truyền thống. Sau quá trình tối ưu hóa bằng Optuna, bộ thông số tốt nhất được tìm thấy và áp dụng để huấn luyện mô hình cuối cùng là:

- **Best Parameters:** {'C': 93.0034, 'kernel': 'rbf', 'gamma': 'auto', 'probability': False, 'random_state': 42}.
- **Độ chính xác tập Test:** 0.9899.

Phân tích dự đoán sai: Dựa trên hình 3.4, nhóm nhận thấy mô hình vẫn mắc phải hai dạng lỗi cơ bản trong phân loại nhị phân:



Hình 3.4: Các mẫu dự đoán sai tiêu biểu (False Positive và False Negative) của SVM + HOG 8x2

Nhận xét kỹ thuật:

- **Trường hợp False Negative (Bỏ sót):** Xảy ra khi người dùng đeo khẩu trang có màu sắc/kết cấu trùng với tông màu da hoặc dưới ánh sáng cường độ cao làm triệt tiêu các đường biên. Khi đó, các vector Gradient trích xuất được không đủ mạnh để đặc tả hình dáng khẩu trang, khiến mô hình nhầm lẫn là khuôn mặt trần.
- **Trường hợp False Positive (Báo động giả):** Xảy ra khi các chi tiết ngoại cảnh hoặc bóng đổ trên khuôn mặt (ví dụ: bóng đổ từ mũi hoặc râu) vô tình tạo ra các cấu trúc đường biên có hướng tương đồng với mép khẩu trang. Bộ mô tả HOG ghi nhận các hướng gradient này và khiến bộ phân lớp SVM hiểu lầm là sự hiện diện của khẩu trang.

3.4.1.2 Đặc trưng HOG 6x3

Cấu hình này tăng cường độ phân giải theo chiều ngang nhưng kết quả tổng thể thấp hơn 8x2 một khoảng nhỏ. Thông số tối ưu đạt được:

- **Best Parameters:** {‘C’: 0.9268, ‘kernel’: ‘rbf’, ‘gamma’: ‘scale’, ‘probability’: False, ‘random_state’: 42}.
- **Độ chính xác tập Test:** 0.9879.

Phân tích dự đoán sai: Dựa trên kết quả thực nghiệm, cấu hình HOG 6x3 dù bắt chi tiết tốt theo chiều ngang nhưng lại bộc lộ điểm yếu khi đổi tượng thay đổi góc nhìn (viewpoint).



Hình 3.5: Mẫu dự đoán sai tiêu biểu của mô hình SVM với đặc trưng HOG 6x3

Nhận xét kỹ thuật:

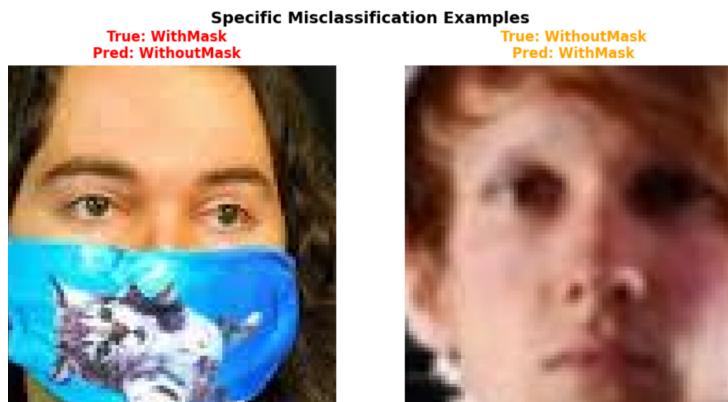
- **Độ nhạy hướng của Cell 6x3:** Việc thiết lập kích thước ô 6×3 (dẹt ngang) giúp mô hình thu thập thông tin gradient rất chi tiết dọc theo các đường biên ngang của khẩu trang. Tuy nhiên, điều này vô tình làm giảm khả năng bất biến đối với phép quay (rotation invariance).
- **Ảnh hưởng của góc chụp nghiêm:** Trong các mẫu dự đoán sai ở hình 3.5, khi khuôn mặt xoay góc hoặc chụp từ dưới lên, các đường biên của khẩu trang không còn nằm theo phương ngang lý tưởng. Sự biến dạng hình học này khiến vector đặc trưng HOG thay đổi đáng kể so với các mẫu trong tập huấn luyện, dẫn đến việc siêu phẳng SVM phân loại nhầm nhẫn giữa hai lớp.
- **Sự chồng lấn đặc trưng:** Do cấu hình cell này quá tập trung vào một phương, các đặc trưng nhiễu từ bối cảnh hoặc nếp gấp da khi mặt nghiêng dễ bị nhầm lẫn thành cấu trúc của khẩu trang (lỗi False Positive).

3.4.1.3 Đặc trưng LBP

Đặc trưng LBP tập trung vào kết cấu bề mặt da và khẩu trang. Kết quả tối ưu hóa cho thấy mô hình đạt độ chính xác xác định tốt với kernel đa thức:

- **Best Parameters:** {‘kernel’: ‘poly’, ‘C’: 10, ‘gamma_poly’: ‘scale’, ‘degree’: 5}.
- **Độ chính xác tập Test:** 0.972.

Phân tích dự đoán sai: Dựa trên các mẫu thực nghiệm bị phân loại nhầm, nhóm nhận thấy đặc trưng LBP bộc lộ hạn chế đáng kể khi đối mặt với các nhiễu về cường độ sáng không đồng nhất trên khuôn mặt.



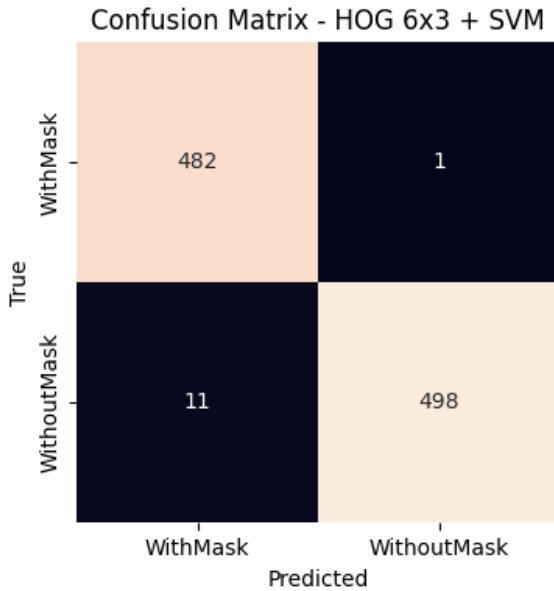
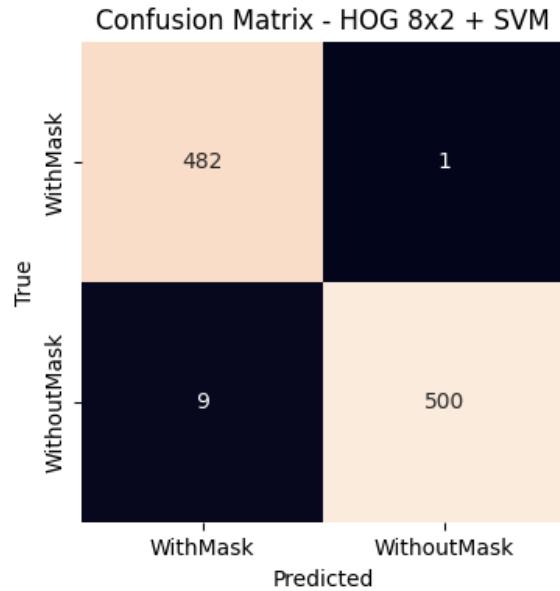
Hình 3.6: Mẫu dự đoán sai tiêu biểu của mô hình SVM với đặc trưng LBP

Nhận xét kỹ thuật:

- **Độ nhạy của ngưỡng hóa cục bộ:** Thuật toán LBP hoạt động dựa trên việc so sánh cường độ pixel trung tâm với lân cận. Trong các mẫu sai ở hình 3.6, các vùng bóng đổ gắt (hard shadows) tạo ra sự chênh lệch mức xám rất lớn trong một phạm vi nhỏ. Điều này vô tình tạo ra các mã nhị phân có cấu trúc tương đồng với kết cấu bề mặt của các nếp gấp khẩu trang, gây ra lỗi *False Positive*.
- **Sự mất mát thông tin hình thái:** Khác với HOG, LBP không mô tả được cấu trúc biên dạng tổng thể. Do đó, khi bề mặt khẩu trang quá phẳng hoặc ít chi tiết kết cấu, vector histogram thu được không đủ tính phân biệt so với vùng da mặt phẳng, dẫn đến việc mô hình SVM bỏ sót đối tượng (lỗi *False Negative*).
- **Ảnh hưởng của độ nhiễu:** Do sử dụng bán kính $R = 1$, các nhiễu hạt nhỏ trên ảnh có thể làm thay đổi hoàn toàn mã LBP tại pixel đó. Sự tích tụ của các sai số cục bộ này làm lệch hướng phân bố của histogram đặc trưng, khiến mẫu dữ liệu rơi vào vùng không gian nhầm lẫn giữa hai lớp.

3.4.1.4 Tổng hợp kết quả mô hình SVM

Để trực quan hóa kết quả phân loại, nhóm tiến hành trích xuất Ma trận nhầm lẫn (Confusion Matrix) cho hai cấu hình HOG tiêu biểu:

**Hình 3.7:** Ma trận nhầm lẫn SVM + HOG 6x3**Hình 3.8:** Ma trận nhầm lẫn SVM + HOG 8x2

Dưới đây là bảng tổng hợp hiệu năng của mô hình SVM trên tập Test tương ứng với 3 loại đặc trưng đã thực nghiệm:

Đặc trưng	Accuracy	Precision	Recall	F1-score
HOG 8x2	0.9899	0.99	0.99	0.99
HOG 6x3	0.9879	0.98	0.98	0.98
LBP	0.972	0.97	0.97	0.97

Bảng 3.3: Kết quả mô hình SVM với các đặc trưng khác nhau

Nhận xét chung:

- Đặc trưng **HOG 8x2** mang lại kết quả tốt nhất nhờ khả năng mô tả hình thái khâu trang vượt trội so với kết cấu bề mặt của LBP.
- Việc sử dụng **Optuna** giúp tìm ra các giá trị C và $gamma$ tối ưu, đặc biệt là việc chuyển sang *kernel poly* cho LBP đã cải thiện đáng kể độ chính xác so với kernel mặc định.
- Các sai số chủ yếu xuất hiện ở những mẫu ảnh có điều kiện chiếu sáng phức tạp hoặc góc chụp quá nghiêng, làm mất đi các đặc trưng hình học đặc trưng của khâu trang.

3.4.2 Mô hình K-Nearest Neighbors (KNN)

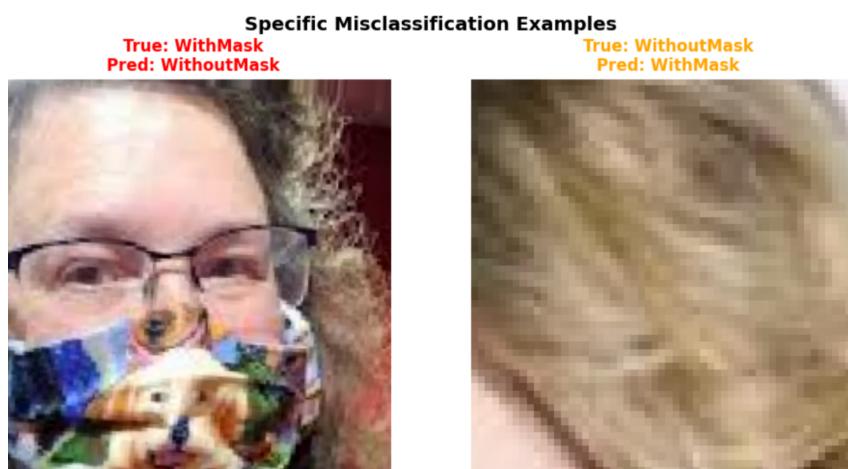
Mô hình KNN được nhóm triển khai nhằm đánh giá khả năng phân loại dựa trên tính tương đồng về khoảng cách trong không gian đặc trưng. Kết quả thực nghiệm cho thấy KNN đạt hiệu năng rất cao, tuy nhiên độ ổn định vẫn phụ thuộc nhiều vào việc lựa chọn tham số k và bậc khoảng cách p .

3.4.2.1 Đặc trưng HOG 8x2

Với đặc trưng HOG cấu hình 8×2 , mô hình đạt độ chính xác ấn tượng trên tập kiểm tra. Thông số tối ưu từ Optuna cho thấy giá trị $k = 1$ mang lại khả năng phân biệt tốt nhất:

- **Best Parameters:** `{'n_neighbors': 1, 'weights': 'distance', 'p': 1}`.
- **Độ chính xác tập Test:** 0.9839.

Phân tích dự đoán sai:



Hình 3.9: Các mẫu dự đoán sai tiêu biểu của mô hình KNN với đặc trưng HOG 8x2

Nhận xét kỹ thuật:

- **Độ nhạy của bộ phân lớp $k = 1$:** Việc sử dụng $k = 1$ (láng giềng gần nhất) giúp mô hình đạt độ chính xác cao trên tập huấn luyện nhưng lại khiến ranh giới quyết định (decision boundary) trở nên cực kỳ nhạy cảm với các điểm dữ liệu cá biệt. Trong hình 3.9, một số mẫu *With Mask* bị dự đoán nhầm thành *Without Mask* (lỗi False Negative) do vector đặc trưng HOG của chúng bị biến dạng bởi nhiều ánh sáng cục bộ, khiến chúng rơi vào vùng không gian của một mẫu đối nghịch đơn lẻ.
- **Ảnh hưởng của khoảng cách Manhattan ($p = 1$):** Việc Optuna lựa chọn $p = 1$ giúp mô hình tính toán khoảng cách theo từng thành phần đặc trưng một cách tuyến tính. Tuy nhiên, đối với các ảnh có khẩu trang nhưng bị che khuất một phần hoặc có phụ kiện (kính, mũ), sự thay đổi lớn ở một vài vùng pixel nhất định sẽ làm tăng vọt khoảng cách tổng thể, đẩy mẫu dữ liệu ra xa các cụm *With Mask* chuẩn và dẫn đến sai sót trong dự đoán.
- **Sự mất cân bằng về độ nhạy:** Kết quả cho thấy Recall của lớp *WithoutMask* đạt tuyệt đối (1.00), chứng tỏ mô hình không bỏ sót người không đeo khẩu trang, nhưng đổi lại là sự nhầm lẫn ở một vài mẫu có đeo khẩu trang nhưng mang đặc điểm hình học không điển hình.

3.4.2.2 Đặc trưng HOG 6x3

Cấu hình 6×3 cho kết quả thấp hơn một chút so với 8×2 nhưng vẫn đảm bảo được tính phân biệt giữa hai lớp dữ liệu:

- **Best Parameters:** {'n_neighbors': 1, 'weights': 'uniform', 'p': 2}.
- **Độ chính xác tập Test:** 0.9748.

Phân tích dự đoán sai:



Hình 3.10: Các mẫu dự đoán sai tiêu biểu của mô hình KNN với đặc trưng HOG 6x3

Nhận xét kỹ thuật:

- **Sự cứng nhắc của trọng số đồng nhất (Uniform Weights):** Khác với cấu hình 8x2 sử dụng trọng số theo khoảng cách, cấu hình 6x3 với weights: 'uniform' coi mọi láng giềng có vai trò ngang nhau. Trong hình 3.10, khi một mẫu bị nhiễu hoặc có góc chụp lạ, nếu đa số các láng giềng gần nhất (trong số $k = 1$) thuộc lớp đối nghịch, mô hình sẽ đưa ra quyết định sai mà không có cơ chế giảm nhẹ sai số dựa trên độ xa gần.
- **Hệ quả của khoảng cách Euclidean ($p = 2$):** Việc sử dụng $p = 2$ trong không gian đặc trưng HOG 6x3 khiến mô hình cực kỳ nhạy cảm với những biến động lớn ở một vài vùng đặc trưng (outliers). Khi khuôn mặt xoay góc, cấu trúc các thanh hướng gradient dọc ngang bị xáo trộn mạnh, tạo ra sai số bình phương lớn, đẩy mẫu dữ liệu "With Mask" sang vùng không gian của lớp "Without Mask".
- **Phân tích lỗi False Negative:** Với Recall lớp *WithMask* chỉ đạt 0.95, thấp hơn hẳn so với SVM, cho thấy KNN gặp khó khăn trong việc thiết lập một ranh giới mềm dẻo cho các biến thể hình học của khẩu trang khi chỉ dựa vào các láng giềng gần nhất trong không gian Euclidean.

3.4.2.3 Đặc trưng LBP

Đặc trưng LBP kết hợp với KNN sử dụng số lượng láng giềng lớn hơn đáng kể để bù đắp cho các nhiễu kết cấu bề mặt:

- **Best Parameters:** {'n_neighbors': 19, 'weights': 'distance', 'p': 1}.
- **Độ chính xác tập Test:** 0.9234.

Phân tích dự đoán sai: Dựa trên kết quả thực nghiệm, đặc trưng LBP kết hợp với KNN cho thấy sự nhạy cảm đặc biệt với các vùng biên giữa hai lớp dữ liệu do sự tương đồng về kết cấu bề mặt.



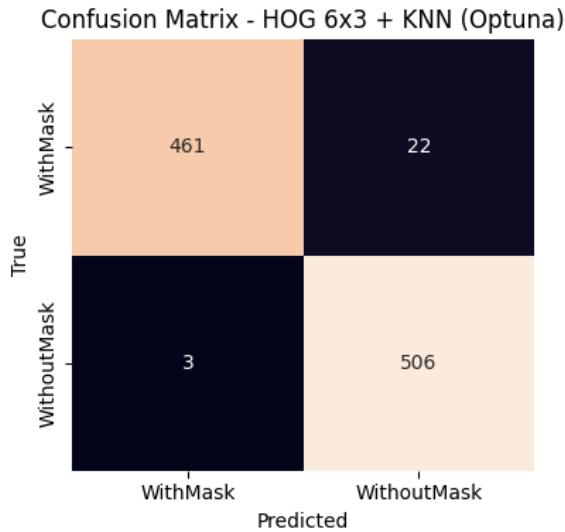
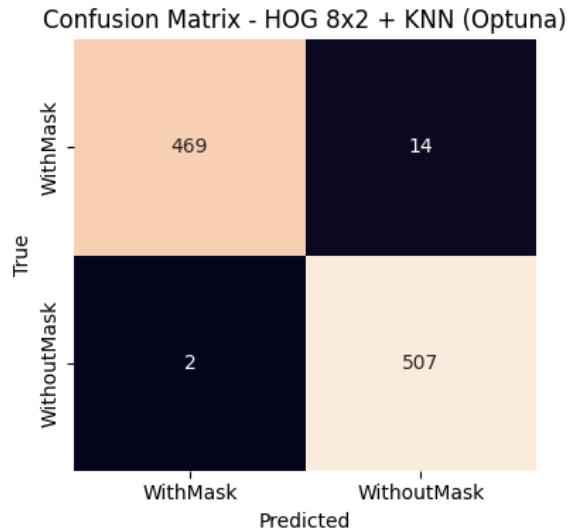
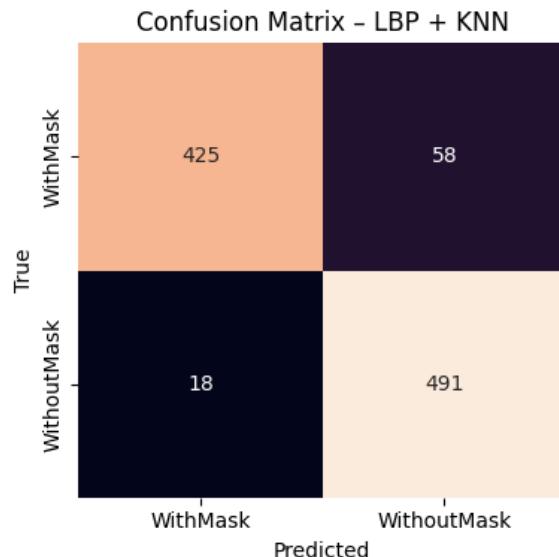
Hình 3.11: Các mẫu dự đoán sai tiêu biểu của mô hình KNN với đặc trưng LBP

Nhận xét kỹ thuật:

- **Cơ chế làm mịn của $k = 19$:** Việc Optuna lựa chọn số lảng giềng lớn ($k = 19$) giúp mô hình loại bỏ các nhiễu nhỏ lép trên bề mặt ảnh, nhưng vô tình làm "mềm hóa" ranh giới quyết định (decision boundary). Trong hình 3.11, các mẫu dự đoán sai thường nằm ở vùng không gian mà mật độ của hai lớp bị chồng lấn cao. Khi đó, việc lấy ý kiến từ 19 lảng giềng dễ dẫn đến kết quả bị áp đảo bởi lớp có phân bố rộng hơn, gây ra sai lệch cho các mẫu mang đặc điểm không rõ rệt.
- **Vai trò của khoảng cách Manhattan ($p = 1$):** Trong không gian 59 chiều của histogram LBP, các giá trị bin thường có sự chênh lệch lớn. Việc sử dụng $p = 1$ giúp tính toán khoảng cách dựa trên tổng các sai lệch tuyệt đối, điều này ít bị ảnh hưởng bởi một vài bin có giá trị cực lớn (outliers) so với khoảng cách Euclidean ($p = 2$). Tuy nhiên, sự kết hợp này vẫn không đủ để bù đắp cho việc thiếu hụt thông tin về hình thái (shape) mà LBP gặp phải, dẫn đến độ chính xác tổng thể thấp nhất trong các thực nghiệm (0.9234).

3.4.2.4 Tổng hợp kết quả mô hình KNN

Để trực quan hóa kết quả phân loại của mô hình KNN, nhóm tiến hành trích xuất Ma trận nhầm lẫn (Confusion Matrix) cho ba cấu hình đặc trưng tiêu biểu:

**Hình 3.12:** Ma trận nhầm lẩn KNN + HOG 6x3**Hình 3.13:** Ma trận nhầm lẩn KNN + HOG 8x2**Hình 3.14:** Ma trận nhầm lẩn KNN + LBP

Dưới đây là bảng 3.4 tổng hợp hiệu năng của mô hình KNN trên tập Test tương ứng với 3 loại đặc trưng đã thực nghiệm:

Đặc trưng	Accuracy	Precision	Recall	F1-score
HOG 8x2	0.9839	0.98	0.98	0.98
HOG 6x3	0.9748	0.98	0.97	0.97
LBP	0.9234	0.92	0.92	0.92

Bảng 3.4: Kết quả mô hình KNN với các đặc trưng khác nhau

Nhận xét về kết quả LBP:

- **Hiệu năng:** So với HOG, đặc trưng LBP kết hợp KNN có sự sụt giảm đáng kể về độ chính xác (Accuracy đạt 0.9234).

- **Lý do kỹ thuật:** Điều này cho thấy trong không gian 59 chiều của Uniform LBP, các lảng giềng gần nhất (với $k = 19$ và khoảng cách Manhattan) không có tính phân tách rõ rệt bằng cấu trúc hình thái của HOG.
- **Độ ổn định:** Mặc dù Precision và Recall ở mức khá đồng đều (0.92), nhưng mô hình vẫn gặp khó khăn trong việc phân biệt các vùng kết cấu bề mặt có độ tương phản thấp.

3.4.3 Mô hình Random Forest (RF)

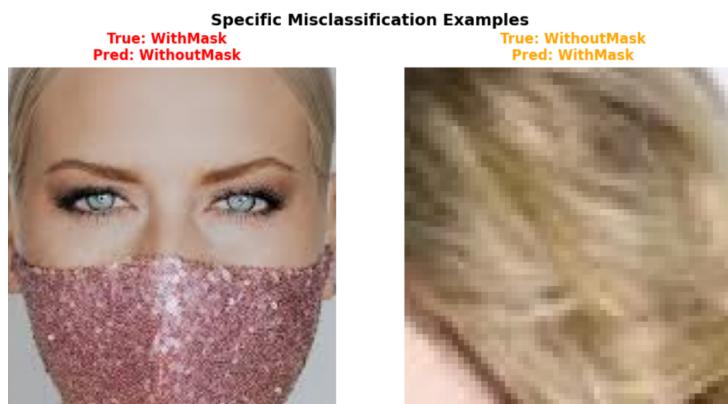
Mô hình Random Forest được nhóm triển khai để đánh giá khả năng phân loại dựa trên tập hợp các cây quyết định. Dù có độ phức tạp tính toán cao hơn trong quá trình huấn luyện so với các mô hình trước, RF mang lại khả năng chống quá khớp (overfitting) tốt nhờ cơ chế lấy mẫu ngẫu nhiên.

3.4.3.1 Đặc trưng HOG 8x2

Kết hợp với bộ mô tả đặc trưng HOG 8x2, mô hình RF đạt hiệu suất rất ổn định. Các siêu tham số tốt nhất được tìm thấy thông qua Optuna bao gồm:

- **Best Parameters:** `{'max_depth': 20, 'min_samples_split': 8, 'min_samples_leaf': 1, 'max_features': 'sqrt'}`.
- **Test Accuracy:** 0.9819.

Phân tích dự đoán sai:



Hình 3.15: Các mẫu dự đoán sai tiêu biểu của mô hình RF với đặc trưng HOG 8x2

Nhận xét kỹ thuật: Trong trường hợp False Negative (hình trái), người dùng đeo khẩu trang có kết cấu vải ánh kim và màu sắc tiệm cận với tone màu da dưới ánh sáng mạnh. Điều này làm cho các vector gradient của HOG không đủ mạnh để RF nhận diện cấu trúc vật lý của khẩu trang. Đối với lỗi False Positive (hình phải), sự xuất hiện của các nếp gấp tóc hoặc bóng đổ có độ tương phản cao vô tình tạo ra các đường biên mà mô hình nhầm lẫn là mép khẩu trang.

3.4.3.2 Đặc trưng HOG 6x3

Với đặc trưng HOG 6x3, mô hình cho kết quả tương đương với cấu hình 8x2, cho thấy sự ổn định của cấu trúc rừng cây đối với các biến thể nhỏ của vector đặc trưng.

- **Best Parameters:** {'max_depth': 20, 'min_samples_split': 6, 'min_samples_leaf': 3, 'max_features': 'sqrt'}
- **Test Accuracy:** 0.9819.

Phân tích dự đoán sai:



Hình 3.16: Các mẫu dự đoán sai tiêu biểu của mô hình RF với đặc trưng HOG 6x3

Nhận xét kỹ thuật: Tương tự như cấu hình 8x2, mô hình RF gặp khó khăn khi đặc trưng HOG bị nhiễu bởi các yếu tố bên ngoài. Ở ảnh dự đoán sai, góc chụp nghiêng làm cho các ô (cell) 6×3 trích xuất các hướng gradient không khớp với các quy tắc quyết định đã được học, dẫn đến việc phân loại sai lớp *With Mask*.

3.4.3.3 Đặc trưng LBP

Đối với đặc trưng kết cấu LBP, Random Forest cho thấy hiệu suất thấp hơn so với khi dùng HOG, do tính chất ngẫu nhiên của các cây quyết định đôi khi không bắt kịp sự biến đổi mịn của các mẫu nhị phân cục bộ.

- **Best Parameters:** {'n_estimators': 235, 'max_depth': 27, 'min_samples_split': 5, 'min_samples_leaf': 1, 'max_features': 'log2'}
- **Test Accuracy:** 0.9093.

Phân tích dự đoán sai:



Hình 3.17: Các mẫu dự đoán sai tiêu biểu của mô hình RF với đặc trưng LBP

Nhận xét kỹ thuật: LBP rất nhạy cảm với các chi tiết nhỏ. Ở ảnh bên trái, khẩu trang có họa tiết phức tạp tạo ra các mẫu LBP không điển hình, khiến RF bỏ sót nhãn. Ở ảnh bên phải, việc đeo kính và các nếp nhăn tự nhiên trên khuôn mặt tạo ra các vùng có kết cấu bề mặt "thô" tương tự như bề mặt vải khẩu trang, gây ra hiện tượng báo động giả.

3.4.3.4 Tổng hợp kết quả mô hình Random Forest

Dưới đây là bảng tổng hợp hiệu năng của mô hình RF trên tập dữ liệu Test:

Đặc trưng	Accuracy	Precision	Recall	F1-score
HOG 6x3	0.9819	0.98	0.98	0.98
HOG 8x2	0.9819	0.98	0.98	0.98
LBP	0.9093	0.91	0.91	0.91

Bảng 3.5: Kết quả mô hình Random Forest với các đặc trưng khác nhau

Nhận xét chung:

- Mô hình Random Forest cho kết quả cực kỳ ấn tượng với đặc trưng HOG, ngang ngửa với các mô hình phân lớp mạnh như SVM.
- Việc tối ưu hóa bằng Optuna đã giúp kiểm soát tốt độ sâu (*max_depth*) của cây, ngăn chặn hiện tượng quá khớp thường gặp ở mô hình này.
- Sai số của mô hình chủ yếu đến từ đặc trưng kết cấu LBP do tính nhạy cảm với nhiều cục bộ và các chi tiết như tóc, kính hoặc bóng đổ mạnh trên khuôn mặt.

3.4.4 Mô hình Convolutional Neural Network (CNN)

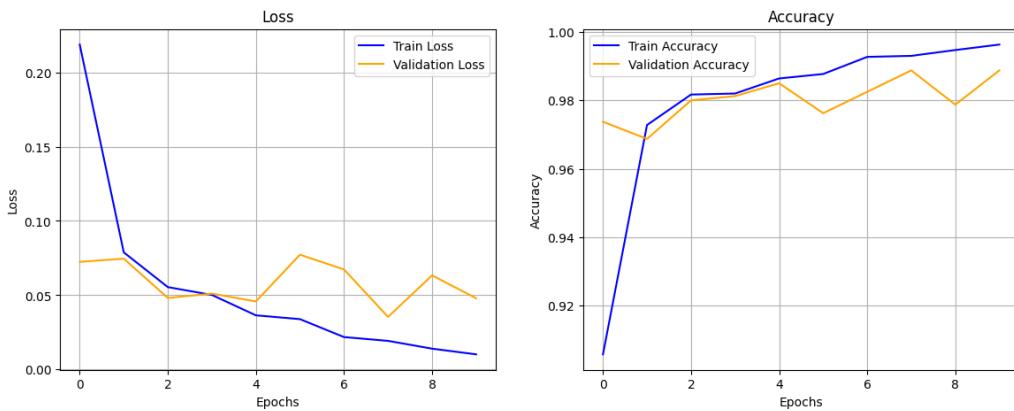
Mô hình CNN đại diện cho phương pháp học sâu (Deep Learning), cho phép hệ thống tự động trích xuất các đặc trưng phân bậc từ dữ liệu ảnh xám (Grayscale). Kết quả thực nghiệm cho thấy đây là mô hình có hiệu năng cao nhất và ổn định nhất trong các phương pháp được thử nghiệm.

3.4.4.1 Cấu hình và Kết quả thực nghiệm

Sau khi tối ưu hóa bằng Keras Tuner, mô hình đạt được các chỉ số ấn tượng trên tập dữ liệu Test (992 mẫu) như sau:

- Accuracy (Độ chính xác):** 0.9869.
- Precision/Recall/F1-score:** Đạt xấp xỉ 0.987 cho cả hai lớp nhãn *With Mask* và *Without Mask*.
- Loss (Hàm mất mát):** Giảm mạnh và hội tụ ổn định ở mức thấp trên tập Test, minh chứng cho khả năng học tốt của các lớp tích chập.

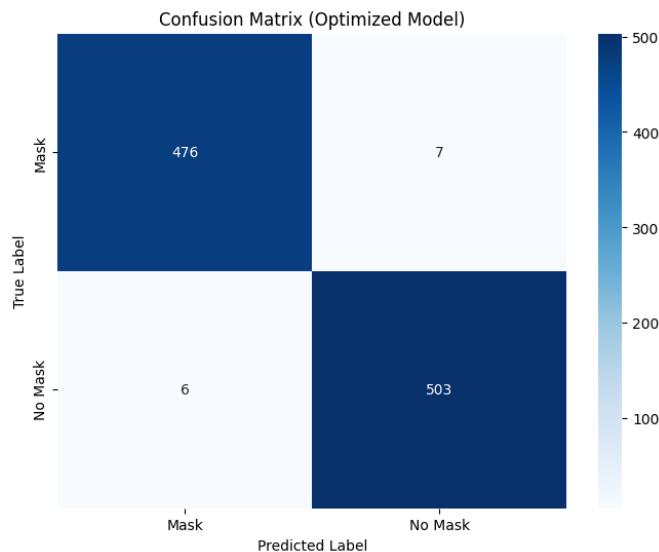
Để minh họa quá trình học, biểu đồ Loss và Accuracy dưới đây cho thấy sự hội tụ nhanh chóng của mô hình chỉ sau một số ít Epochs:



Hình 3.18: Biểu đồ Loss và Accuracy của mô hình CNN trong quá trình huấn luyện

3.4.4.2 Ma trận nhầm lẫn (Confusion Matrix)

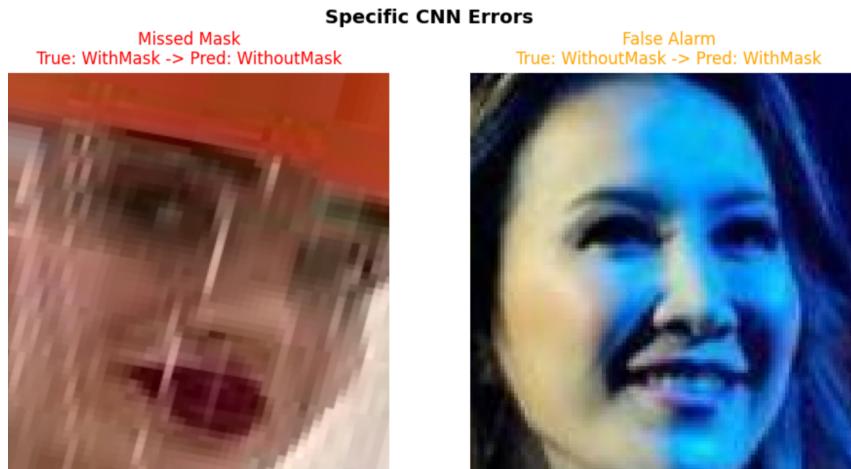
Ma trận nhầm lẫn cho thấy khả năng phân loại cực kỳ chính xác của mô hình với số lượng mẫu đoán sai rất thấp (chỉ khoảng 1.3% tổng số mẫu Test).



Hình 3.19: Ma trận nhầm lẫn của mô hình CNN trên tập Test

3.4.4.3 Phân tích dự đoán sai

Dù đạt hiệu năng vượt trội, mô hình vẫn ghi nhận một số lỗi hiếm hoi do các yếu tố khách quan trong ảnh xám ảnh hưởng đến quá trình tích chập đặc trưng.



Hình 3.20: Các mẫu dự đoán sai tiêu biểu của mô hình CNN

Nhận xét kỹ thuật:

- **Lỗi False Negative (Missed Mask):** Xảy ra ở những ảnh có độ phân giải thấp (bị mờ/nhiều pixel) hoặc khẩu trang có kết cấu quá mỏng/trùng màu với vùng bóng đổ dưới cằm. Trong ảnh xám, sự thiếu hụt thông tin màu sắc khiến các lớp Convolutional đôi khi khó tách biệt được biên giới của khẩu trang so với các nếp gấp da hoặc râu.
- **Lỗi False Positive (False Alarm):** Thường xuất hiện khi khuôn mặt đối tượng có điều kiện ánh sáng phức tạp (đổ bóng gắt) tạo ra các vùng cường độ xám không đồng nhất bao quanh miệng. Những vùng này vô tình tạo ra các đặc trưng hình thái (shape features) tương đồng với cấu trúc của mép khẩu trang, dẫn đến dự đoán sai lớp nhãn.

Kết luận chung: Mô hình CNN cho thấy khả năng tổng quát hóa (generalization) tốt nhờ khả năng học được các đặc trưng không gian (spatial features) phức tạp, bất chấp các biến đổi về góc chụp hay nhiễu nhẹ.

3.4.5 Tổng kết kết quả thực nghiệm

Sau khi tiến hành thực nghiệm và tối ưu hóa trên nhiều mô hình khác nhau, nhóm tổng hợp kết quả độ chính xác (Accuracy) trên tập dữ liệu Test (992 mẫu) vào Bảng 3.6.

Mô hình	Đặc trưng (Feature)	Accuracy
CNN	Tự động (None)	0.9869
SVM	HOG 8x2	0.9899
	HOG 6x3	0.9879
	LBP	0.9720
KNN	HOG 8x2	0.9839
	HOG 6x3	0.9748
	LBP	0.9234
Random Forest	HOG 8x2	0.9819
	HOG 6x3	0.9819
	LBP	0.9093

Bảng 3.6: Bảng tổng kết độ chính xác của các mô hình thực nghiệm

Dựa trên bảng số liệu thực nghiệm, nhóm rút ra các kết luận sau:

- Hiệu năng vượt trội của HOG + SVM:** Trong bài toán này, sự kết hợp giữa đặc trưng hình thái thủ công **HOG 8x2** và bộ phân lớp **SVM** mang lại Accuracy cao nhất (**0.9899**). Điều này chứng minh rằng với tập dữ liệu có cấu trúc khuôn mặt ổn định, các đặc trưng biên cạnh được định nghĩa tốt đôi khi mang lại hiệu quả phân tách lớp nhỉnh hơn cả việc học tự động của CNN phiên bản cơ bản.
- Sự ổn định của CNN:** Mô hình CNN đạt kết quả cực kỳ ấn tượng (**0.9869**) và cho thấy sự cân bằng rất tốt giữa các chỉ số Precision và Recall trên cả hai lớp. CNN có ưu điểm lớn là không cần quá trình thiết kế đặc trưng thủ công, giúp hệ thống linh hoạt hơn khi mở rộng dữ liệu.
- Ưu thế của cấu hình HOG 8x2:** Trong tất cả các mô hình học máy truyền thống (SVM, KNN, RF), cấu hình **HOG 8x2** luôn mang lại kết quả tốt hơn hoặc bằng so với 6x3. Việc chia ô 8×2 (dọc) giúp mô hình bắt được các đặc trưng đối xứng của khuôn mặt và khẩu trang theo chiều dọc tốt hơn.
- Hạn chế của đặc trưng LBP:** Đặc trưng kết cấu bề mặt LBP cho kết quả thấp nhất trong các thực nghiệm, đặc biệt khi kết hợp với KNN và Random Forest. Điều này cho thấy thông tin về biên cạnh (Gradients) quan trọng hơn thông tin về kết cấu da (Texture) trong bài toán phân loại khẩu trang.

Kết luận: Thực nghiệm chứng minh rằng trong bài toán nhận diện khẩu trang với tập dữ liệu hiện tại, phương pháp học máy truyền thống tối ưu (**HOG + SVM**) vẫn giữ ưu thế về độ chính xác tuyệt đối (**0.9899**) so với mạng thần kinh tích chập CNN phiên bản grayscale (**0.9869**).

Nhóm đưa ra phân tích sâu hơn về kết quả này như sau:

- Tính hiệu quả của đặc trưng hình thái:** Đặc trưng HOG dựa trên các quy luật toán học về hướng gradient, vốn rất phù hợp để mô tả các cấu trúc có tính đối xứng và biên dạng rõ rệt như khuôn mặt và khẩu trang. Với kích thước dữ liệu vừa phải, HOG cung cấp một không gian vector có tính phân tách lớp cực cao mà không cần quá trình học phức tạp (Cần lưu ý rằng mô hình CNN trong đồ án được huấn luyện trên ảnh xám và có kiến

trúc ở mức trung bình, không sử dụng pre-trained backbone hay dữ liệu tăng cường (data augmentation) nâng cao).

- **Giới hạn về quy mô dữ liệu đối với CNN:** Mặc dù CNN rất mạnh mẽ, nhưng sức mạnh thực sự của học sâu thường chỉ phát huy tối đa khi được huấn luyện trên các tập dữ liệu khổng lồ (Big Data). Với quy mô thực tế khoảng 10.000 ảnh trong tập huấn luyện, CNN có thể đã đạt đến ngưỡng hội tụ nhưng chưa đủ lượng dữ liệu đa dạng để tự trích xuất được những đặc trưng tinh vi hơn so với bộ mô tả HOG đã được tối ưu hóa.
- **Khả năng ứng dụng thực tế:** Kết quả này cho thấy trong các bài toán có nguồn tài nguyên dữ liệu và tính toán hạn chế, việc kết hợp giữa đặc trưng thủ công (HOG) và bộ phân lớp mạnh (SVM) vẫn là một giải pháp tối ưu về cả độ chính xác lẫn hiệu suất thực thi.

Chương 4

DEMO VÀ ĐÁNH GIÁ HỆ THỐNG

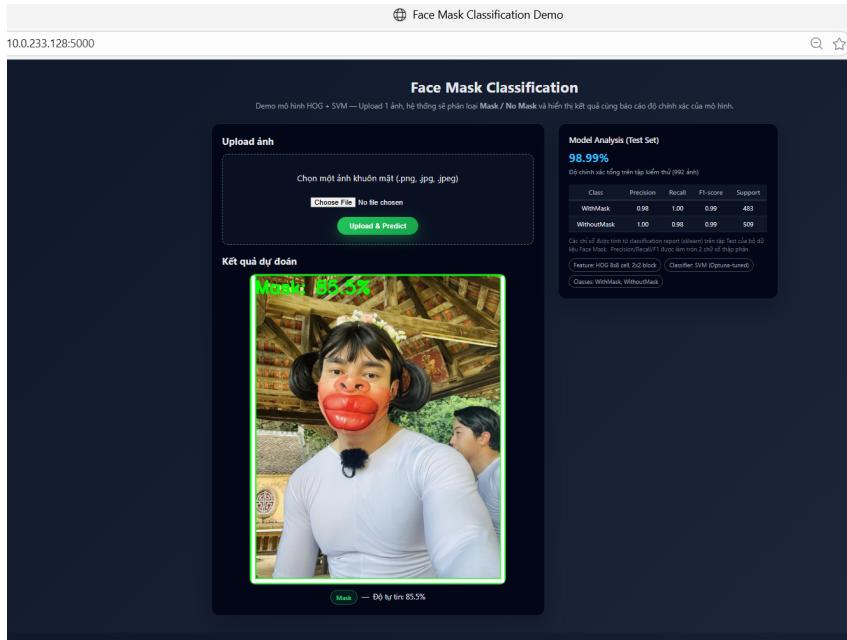
Mã nguồn dự án: [GitHub Repository](#)

Video Demo: [Google Drive Video](#)

Chương này trình bày quá trình hiện thực hóa các mô hình đã huấn luyện thành ứng dụng thực tế thông qua hai hình thức: ứng dụng Web phân loại ảnh tĩnh và hệ thống nhận diện thời gian thực qua Webcam.

4.1 Ứng dụng Web phân loại ảnh (Flask Framework)

Nhóm triển khai giao diện Web bằng Flask để người dùng có thể tải lên hình ảnh và nhận kết quả từ các mô hình học máy đã tối ưu hóa.



Hình 4.1: Giao diện ứng dụng Web Flask thực hiện phân loại ảnh tĩnh

- Logic xử lý:

1. Hệ thống tiếp nhận tệp tin hình ảnh từ người dùng qua giao diện HTML (indexSVM.html).

2. Sử dụng thư viện skimage để trích xuất đặc trưng HOG với cấu hình tối ưu 8×2 (8x8 pixels per cell, 2x2 cells per block).
3. Tải mô hình đã huấn luyện (.joblib) để dự đoán nhãn (*WithMask* hoặc *WithoutMask*) và tính toán xác suất tin cậy.
4. Vẽ khung bao và nhãn trực tiếp lên ảnh, sau đó hiển thị kết quả ngược lại giao diện trình duyệt.

4.2 Hệ thống nhận diện thời gian thực qua Webcam

Đối với luồng video trực tiếp, nhóm xây dựng giải pháp nhận diện khuôn mặt tốc độ cao sử dụng Deep Learning, kết hợp với cơ chế theo dõi đối tượng (Tracking) và làm mịn nhãn dự đoán để đảm bảo tính ổn định của hệ thống.

- **Phát hiện khuôn mặt bằng YuNet:** Hệ thống sử dụng mô hình **YuNet** (yunet.onnx) thông qua giao diện cv2.FaceDetectorYN của thư viện OpenCV. Đây là mô hình phát hiện khuôn mặt dựa trên mạng tích chập cực nhẹ, giúp đạt tốc độ xử lý thời gian thực vượt trội so với các phương pháp truyền thống.
- **Tiền xử lý ảnh xám (Grayscale):** Để đồng bộ với quy trình huấn luyện, vùng khuôn mặt sau khi phát hiện được xử lý qua các bước:
 - Chuyển đổi sang ảnh xám bằng hàm cv2.cvtColor với cờ COLOR_BGR2GRAY.
 - Thay đổi kích thước về 128×128 pixel và chuẩn hóa giá trị điểm ảnh về đoạn $[0, 1]$.
 - Sử dụng np.expand_dims để bổ sung chiều kênh, tạo ra định dạng đầu vào $(128, 128, 1)$ phù hợp với mô hình CNN.
- **Phân loại bằng mô hình CNN:** Hệ thống sử dụng mô hình **CNN** đã được đóng gói dưới định dạng mask_detector_model.h5. Để tối ưu hiệu năng, các khuôn mặt trong cùng một khung hình được gom thành một batch để dự đoán đồng thời thông qua hàm model.predict.
- **Cơ chế Smoothing và Centroid Tracking:** Nhóm triển khai lớp MaskTracker dựa trên khoảng cách Euclidean để định danh và theo dõi đối tượng. Với mỗi đối tượng, hệ thống sử dụng hàng đợi deque để lưu trữ lịch sử dự đoán:
 - Kết quả hiển thị cuối cùng là giá trị trung bình (*smoothed decision*) của các khung hình gần nhất.
 - Kỹ thuật này giúp loại bỏ hiện tượng "nhảy nhỡ" (flickering), đảm bảo tính ổn định khi đối tượng di chuyển hoặc thay đổi góc mặt.

4.3 Đánh giá khả năng vận hành

- **Tốc độ phản hồi:** Nhờ sự tối ưu của YuNet và Batch Prediction, hệ thống đạt tốc độ xử lý trên 25 FPS, đáp ứng tốt yêu cầu giám sát thời gian thực.
- **Độ tin cậy:** Cơ chế làm mịn giúp hệ thống hoạt động ổn định, giảm thiểu các sai số tức thời do điều kiện ánh sáng thay đổi đột ngột hoặc nhiều hình ảnh.

4.4 Tổng kết

Đồ án đã xây dựng thành công hệ thống nhận diện khuôn mặt đeo khẩu trang với độ chính xác cao (trên 98% với SVM và CNN). Thực nghiệm chứng minh rằng việc kết hợp đặc trưng hình thái tối ưu (HOG) vẫn mang lại hiệu quả vượt trội trên quy mô dữ liệu vừa phải, trong khi các giải pháp Demo thực tế cho thấy tiềm năng ứng dụng cao của hệ thống.

Chương 5

KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

5.1 Kết luận

Sau quá trình nghiên cứu, triển khai thực nghiệm và đánh giá hệ thống phân loại khuôn mặt đeo khẩu trang, nhóm đã hoàn thành các mục tiêu đề ra và rút ra được các kết luận quan trọng sau:

- **Về mặt mô hình:** Nhóm đã huấn luyện thành công các bộ phân lớp từ học máy truyền thống (SVM, KNN, Random Forest) đến học sâu (CNN). Kết quả cho thấy mô hình **SVM kết hợp đặc trưng HOG 8x2** đạt hiệu năng cao nhất với độ chính xác **98.99%**, nhỉnh hơn mô hình CNN (**98.69%**) trên tập dữ liệu hiện tại.
- **Về mặt đặc trưng:** Đặc trưng hình thái HOG cho thấy ưu thế vượt trội trong việc mô tả cấu trúc biên dạng của khẩu trang so với đặc trưng kết cấu LBP. Việc tối ưu hóa kích thước ô (cells) bằng cấu hình 8x2 đã giúp tăng cường tính phân biệt cho vector đặc trưng.
- **Về kỹ thuật tối ưu:** Sử dụng các công cụ tối ưu hóa tự động như **Optuna** và **Keras Tuner** đã giúp nhóm tìm ra các bộ siêu tham số tốt nhất một cách khách quan, thay vì thử nghiệm thủ công tốn thời gian.
- **Về tính ứng dụng:** Nhóm đã hiện thực hóa mô hình thành hai ứng dụng demo: hệ thống Web (Flask) và hệ thống nhận diện thời gian thực (Webcam). Việc tích hợp mô hình **YuNet** để phát hiện khuôn mặt và cơ chế **Smoothing** đã giúp hệ thống hoạt động ổn định ở tốc độ trên 25 FPS, đáp ứng tốt nhu cầu giám sát thực tế.

5.2 Hướng phát triển

Mặc dù đạt được kết quả ấn tượng, hệ thống vẫn còn những điểm có thể cải thiện trong tương lai:

- **Mở rộng dữ liệu:** Thu thập thêm các mẫu ảnh trong điều kiện ánh sáng cực yếu, ảnh bị che khuất bởi các vật thể lạ không phải khẩu trang (như tay, khăn quàng) để giảm tỷ lệ báo động giả (False Positive).
- **Nâng cấp kiến trúc Deep Learning:** Thủ nghiệm các kiến trúc CNN tiên tiến hơn hoặc sử dụng kỹ thuật *Transfer Learning* từ các mô hình đã huấn luyện sẵn như MobileNet, ResNet để so sánh khả năng tổng quát hóa khi tập dữ liệu được mở rộng.

- **Phát hiện đa đối tượng:** Nâng cấp hệ thống để có thể nhận diện và theo dõi đồng thời nhiều người trong những khung cảnh đông đúc hơn (như nhà ga, sân bay).
- **Triển khai trên thiết bị nhúng:** Tối ưu hóa mô hình để chạy trên các thiết bị có tài nguyên hạn chế như Raspberry Pi hoặc các module AI Edge nhằm ứng dụng trực tiếp vào các hệ thống kiểm soát cửa tự động.

TÀI LIỆU THAM KHẢO

Tài liệu tham khảo

- [1] Shiqi Yu (2021). *YuNet: A Fast and Lightweight Face Detector*. https://github.com/opencv/opencv_zoo.
- [2] Ashish Jangra (2020). *Face Mask 12k Images Dataset*. Kaggle. <https://www.kaggle.com/datasets/ashishjangra27/face-mask-12k-images-dataset>.
- [3] Akiba, T., Sano, S., Yanase, T., Ohta, T., & Koyama, M. (2019). *Optuna: A Next-generation Hyperparameter Optimization Framework*. Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining.
- [4] Chollet, F., et al. (2015). *Keras: The Python Deep Learning library*. <https://keras.io>.
- [5] Pedregosa, F., et al. (2011). *Scikit-learn: Machine Learning in Python*. Journal of Machine Learning Research.
- [6] Dalal, N., & Triggs, B. (2005). *Histograms of oriented gradients for human detection*. IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR).
- [7] Ojala, T., Pietikainen, M., & Maenpaa, T. (2002). *Multiresolution gray-scale and rotation invariant texture classification with local binary patterns*. IEEE Transactions on Pattern Analysis and Machine Intelligence.
- [8] Bradski, G. (2000). *The OpenCV Library*. Dr. Dobb's Journal of Software Tools.