

BỘ CÔNG THƯƠNG
TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP THỰC PHẨM TP HCM
KHOA CÔNG NGHỆ THÔNG TIN



KHÓA LUẬN TỐT NGHIỆP

ĐỀ TÀI:

**TRA CỨU KẾT QUẢ HỌC TẬP SINH VIÊN HUFIT QUA
HÌNH ẢNH**

GVHD: NGUYỄN VĂN THỊNH

SVTH:

ĐẶNG HOÀNG CẨM MY 2001180476

NGUYỄN NGỌC HẢI 2001181090

TRƯƠNG HỒNG PHI 2001180002

TP.HCM, tháng 12 năm 2021

BỘ CÔNG THƯƠNG
TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP THỰC PHẨM TP HCM
KHOA CÔNG NGHỆ THÔNG TIN



KHÓA LUẬN TỐT NGHIỆP

ĐỀ TÀI:

**TRA CỨU KẾT QUẢ HỌC TẬP SINH VIÊN HUFI QUA
HÌNH ẢNH**

GVHD: NGUYỄN VĂN THỊNH

SVTH:

ĐẶNG HOÀNG CẨM MY 2001180476

NGUYỄN NGỌC HẢI 2001181090

TRƯƠNG HỒNG PHI 2001180002

TP.HCM, tháng 12 năm 2021

NHẬN XÉT CỦA GVHD

.....

.....

.....

.....

.....

.....

.....

.....

.....

Ngày tháng năm 2021

GVHD

(Ký ghi rõ họ tên)

NHẬN XÉT CỦA GV CHẤM

.....

.....

.....

.....

.....

.....

.....

.....

.....

Ngày tháng năm 2021

GV chấm

(Ký ghi rõ họ tên)

LỜI CAM ĐOAN

Chúng tôi xin cam đoan rằng báo cáo khóa luận tốt nghiệp này là do chính bản thân thực hiện dưới sự hướng dẫn của Thầy Nguyễn Văn Thịnh, các kết quả nghiên cứu và các kết luận trong bài là trung thực, không sao chép từ bất cứ đề tài nghiên cứu khoa học nào. Việc tham khảo các nguồn tài liệu đã được thực hiện trích dẫn và ghi nguồn tài liệu tham khảo theo đúng yêu cầu.

Chúng tôi xin chịu hoàn toàn trách nhiệm với những nội dung viết trong khóa luận này!

TP.HCM, tháng 12 năm 2021

SVTH1

(Ký và ghi rõ họ tên)

SVTH2

(Ký và ghi rõ họ tên)

SVTH3

(Ký và ghi rõ họ tên)

LỜI CẢM ƠN

Trong quá trình học tập và thực hiện khóa luận, chúng tôi được các Thầy Cô trường Đại học Công nghiệp Thực phẩm Thành phố Hồ Chí Minh tạo mọi điều kiện thuận lợi cũng như đồng nghiệp và bạn bè đã thường xuyên động viên. Chúng tôi xin bày tỏ sự cảm ơn chân thành với những sự hỗ trợ và giúp đỡ này.

Khóa luận sẽ không thể hoàn thành nếu không có sự hướng dẫn tận tình của Thầy hướng dẫn Ths Nguyễn Văn Thịnh là người mà tôi muốn bày tỏ lòng biết ơn sâu sắc nhất.

Xin chân thành cảm ơn tập thể Thầy Cô trong khoa Công nghệ Thông tin - Trường Đại học Công nghiệp Thực phẩm Thành phố Hồ Chí Minh đã quan tâm tạo môi trường thuận lợi để học tập và nghiên cứu chuyên sâu về lĩnh vực Công nghệ thông tin.

Cuối cùng chúng tôi cảm ơn tất cả những sự giúp đỡ của đồng nghiệp, bạn bè đã đóng góp ý kiến, động viên để chúng tôi hoàn thành được khóa luận này.

MỤC LỤC

LỜI CAM ĐOAN	v
LỜI CẢM ƠN	vi
MỤC LỤC	vii
DANH MỤC HÌNH ẢNH	xi
DANH MỤC BẢNG BIỂU	xiii
MỞ ĐẦU	xiv
CHƯƠNG 1 TỔNG QUAN ĐỀ TÀI	17
1.1 Hiện trạng	17
1.2 Mục tiêu và phạm vi đề tài	17
1.3 Nội dung nghiên cứu	18
1.4 Phương pháp nghiên cứu	18
1.5 Công nghệ sử dụng	18
CHƯƠNG 2 CƠ SỞ LÝ THUYẾT	19
2.1 Giới thiệu về ảnh số	19
2.1.1 Khái niệm	19
2.1.2 Kích thước ảnh	21
2.1.3 Ảnh xám	21
2.1.4 Ảnh màu	22
2.2 Phát hiện khuôn mặt (Face Detection)	23
2.2.1 Khái niệm	23
2.2.2 Phương pháp MTCNN	23
2.2.3 Phương pháp Haar-like - Adaboost	26

2.3	Trích xuất đặc trưng (Feature Extraction).....	29
2.3.1	Khái niệm	29
2.3.2	Phương pháp HOG	29
2.3.3	Phương pháp Facial landmark.....	33
2.4	Nhận dạng khuôn mặt (Face Recognition)	35
2.4.1	Khái niệm	35
2.4.2	SVM (Support Vector Machine)	36
2.4.3	KNN (K Nearest Neighbors)	38
2.4.4	Cây quyết định (Decision Tree)	39
2.5	Giới thiệu về API	40
2.5.1	API là gì?.....	40
2.5.2	Web API là gì?	40
2.6	Giới thiệu về Flask	40
2.6.1	Flask là gì?.....	40
2.6.2	Tính năng của Flask Framework.....	41
CHƯƠNG 3 PHƯƠNG PHÁP NHẬN DẠNG SINH VIÊN VÀ TRA CỨU KẾT QUẢ.....		42
3.1	Phương pháp nhận dạng.....	42
3.1.1	Huấn luyện mô hình	42
3.1.2	Các bước nhận dạng	42
3.2	Chi tiết phương pháp nhận dạng	43
3.2.1	Phát hiện khuôn mặt.....	43
3.2.2	Rút trích đặc trưng HOG	44

3.2.3	Huấn luyện mô hình phân loại SVM.....	44
3.3	Thiết kế cơ sở dữ liệu.....	45
CHƯƠNG 4 THỰC NGHIỆM VÀ ĐÁNH GIÁ		49
4.1	Mô hình thực nghiệm	49
4.2	Dữ liệu thực nghiệm.....	50
4.3	Phương pháp thực nghiệm	52
4.3.1	Rút trích đặc trưng HOG	52
4.3.2	Rút trích đặc trưng Land-mark.....	53
4.3.3	Xây dựng mô hình phân loại SVM	53
4.4	Kết quả thực nghiệm	54
4.4.1	Tốc độ xử lý rút trích đặc trưng	54
4.4.2	Độ chính xác.....	55
4.4.3	So sánh với các phương pháp khác	55
4.5	Kiến trúc ứng dụng.....	56
4.6	Một số xử lý chính.....	57
4.6.1	Phát hiện khuôn mặt.....	57
4.6.2	Rút trích đặc trưng.....	57
4.6.3	Nhận dạng khuôn mặt.....	58
4.6.4	Xây dựng web API lấy dữ liệu hình ảnh và nhận dạng.....	58
4.6.5	Truyền dữ liệu từ ứng dụng và website lên web API.....	59
4.7	Giao diện chương trình	61
4.7.1	Giao diện ứng dụng	61
4.7.2	Giao diện website	63

Kết luận.....	65
TÀI LIỆU THAM KHẢO	66

DANH MỤC HÌNH ẢNH

Hình 2-1 Ảnh raster	19
Hình 2-2 Ảnh vector	20
Hình 2-3 Kích thước ảnh	21
Hình 2-4 Ảnh xám	22
Hình 2-5 Ảnh màu	22
Hình 2-6 Cấu trúc của MTCNN	24
Hình 2-7 Cấu trúc mạng P-Net	25
Hình 2-8 Cấu trúc mạng R-Net.....	25
Hình 2-9 Cấu trúc mạng O-Net	26
Hình 2-10 Đặc trưng Haar – like	27
Hình 2-11 Đặc trưng Haar-like cơ bản để xác định khuôn mặt.....	27
Hình 2-12 Đặc trưng Haar – like mở rộng.....	27
Hình 2-13 Resize kích thước tất cả các hình ảnh	30
Hình 2-14 Phép nhân chập ảnh gốc với 2 chiều	30
Hình 2-15 Chia hình ảnh thành các block	31
Hình 2-16 Tính toán đặc trưng HOG tại mỗi cell.....	32
Hình 2-17 Xây dựng một biểu đồ cường độ gradient.....	32
Hình 2-18 68 điểm landmark	35
Hình 2-19 Ví dụ, trong không gian 3 chiều, siêu phẳng chính là mặt phẳng 2 chiều. Trong không gian 2 chiều, siêu phẳng là đường thẳng 1 chiều.	36
Hình 2-20 Ảnh minh họa tìm biên lớn nhất.....	37
Hình 2-21 Cách tính khoảng cách	39

Hình 2-22 Web API liên kết các ứng dụng với nhau.....	40
Hình 2-23 Tìm hiểu về Flask	41
Hình 2-24 Ví dụ về cách triển khai web API của Flask	41
Hình 3-1 Mô hình xây dựng model phân loại.....	42
Hình 3-2 Mô hình nhận dạng ảnh khuôn mặt	42
Hình 3-3 Phát hiện khuôn mặt	43
Hình 3-4 Cơ sở dữ liệu	45
Hình 4-1 Mô hình hệ thống tra cứu kết quả học tập sinh viên qua hình ảnh.....	49
Hình 4-2 Tập ảnh YaleFace	51
Hình 4-3 Tập ảnh YaleFaceB	51
Hình 4-4 Tập ảnh sinh viên khoa Công nghệ thông tin.....	52
Hình 4-5 Kiến trúc ứng dụng.....	56
Hình 4-6 Giao diện nhận diện hình ảnh.....	61
Hình 4-7 Giao diện kết quả nhận diện hình ảnh.	62
Hình 4-8 Giao diện xem điểm sinh viên.....	62
Hình 4-9 Giao diện Website trang nhận diện gương mặt	63
Hình 4-10 Giao diện Website sau khi nhận diện được gương mặt.....	63
Hình 4-11 Giao diện Website xem kết quả học tập của sinh viên.....	64

DANH MỤC BẢNG BIỂU

Bảng 3-1 Bảng BACDAOTAO (bậc đào tạo)	45
Bảng 3-2 Bảng LOAIHINHDAOTAO (loại hình đào tạo)	45
Bảng 3-3 Bảng KHOA (khoa)	46
Bảng 3-4 Bảng NGANH (ngành)	46
Bảng 3-5 Bảng LOP (lớp).....	46
Bảng 3-6 Bảng NIENKHOA (niên khóa).....	46
Bảng 3-7 Bảng HOCKY (học kỳ)	47
Bảng 3-8 Bảng MONHOC (môn học).....	47
Bảng 3-9 Bảng SINHVIEN (sinh viên).....	47
Bảng 3-10 Bảng DIEM (điểm)	48
Bảng 4-1 Bảng hiệu xuất độ chính xác của tập dữ liệu học khi kiểm thử với Kfold trên bộ dữ liệu sinh viên.....	53
Bảng 4-2 Bảng hiệu suất nhận dạng ảnh khuôn mặt của phương pháp được đề xuất trên các bộ dữ liệu thực nghiệm.....	54
Bảng 4-3 Bảng so sánh tốc độ xử lý khi rút trích đặc trưng trên hai phương pháp thực nghiệm.....	54
Bảng 4-4 Bảng so sánh độ chính xác nhận dạng trung bình trên hai phương pháp thực nghiệm.....	55
Bảng 4-5 So sánh các phương pháp trên bộ dữ liệu YaleFace	55
Bảng 4-6 So sánh các phương pháp trên bộ dữ liệu YaleFaceB	55

MỞ ĐẦU

Trong những năm gần đây, các công nghệ sinh trắc học như nhận dạng vân tay (Finger recognition) hay nhận dạng mống mắt (Iris recognition) đã được áp dụng rộng rãi trong cuộc sống, với độ tin cậy cao [1]. Một ví dụ cụ thể là từ năm 2011, Google đã áp dụng công nghệ xác minh mống mắt để kiểm soát truy cập tại các trung tâm dữ liệu của tập đoàn này. Nhận dạng khuôn mặt (Face Recognition) là một hướng nghiên cứu quan trọng của công nghệ sinh trắc học (Biometrics) và thị giác máy tính (Computer Vision). Đã có nhiều công trình nghiên cứu về bài toán xác định khuôn mặt và nhận diện khuôn mặt người từ ảnh đen trắng, ảnh xám, đến ảnh màu [4, 5, 6]. Sau đây là một vài công trình nghiên cứu liên quan đến nhận dạng khuôn mặt:

Trên tạp chí Khoa học và Công nghệ tập 57 số 3 (tháng 6 năm 2021) có bài “NHẬN DIỆN KHUÔN MẶT SỬ DỤNG MẠNG NƠON TÍCH CHẬP XẾP CHỒNG VÀ MÔ HÌNH FACENET” của tác giả Trần Hồng Việt, Đỗ Đình Tiến, Nguyễn Thị Trà và Trần Lâm Quân cho thấy thuật toán nhận dạng khuôn mặt được đề xuất có độ chính xác nhận dạng cao sử dụng phương pháp nhận dạng khuôn mặt dựa trên học sâu. Kết quả 10 lần chạy thử nghiệm. Kết quả trên tập dữ liệu đều cho kết quả độ chính xác phân lớp (accuracy) tốt cụ thể tỉ lệ train model bằng phương pháp CNN là 95% bằng MTCNN, FaceNet sử dụng SVM là 95,1% và MTCNN, FaceNet sử dụng VGG16 là 97% [31]. Tuy nhiên phương pháp học sâu có hạn chế cần dữ liệu khá nhiều để train và rất tốn chi phí.

Tháng 5 năm 2008 AYDIN KAYACIK của Đại học Halmstad đã sử dụng phương pháp phát hiện khuôn mặt dựa trên phân loại bằng cách sử dụng các đặc trưng của bộ lọc Gabor. Vector đặc trưng dựa trên bộ lọc Gabor được sử dụng làm đầu vào của bộ phân loại khuôn mặt và được rút gọn được trích xuất bằng cách sử dụng PCA để giảm chiều, và sử dụng SVM để xây dựng bộ phân loại. Kết quả thử nghiệm đạt độ chính xác 78% cho thấy hiệu suất đầy hứa hẹn, đặc biệt là trên các hình ảnh khuôn mặt đơn lẻ [32].

Năm 2001 P. Quintiliano và các cộng sự của mình đã đề xuất một cách tiếp cận dựa trên các eigenfaces và phân tích thành phần chính đạt tỉ lệ chính xác là 72% khi làm việc với

các hình ảnh có vấn đề về ánh sáng. Có thể thấy hạn chế ở đây là các ảnh có vấn đề về ánh sáng chưa được nhận diện tốt [33].

Năm 2012 Kỹ sư Nguyễn Thị Nông đã nghiên cứu ứng dụng của lý thuyết Compressive sensing (CS) vào bài toán nhận dạng khuôn mặt nói trên nhằm nâng cao hiệu suất nhận dạng và tối ưu hóa hệ thống. Kết quả cho thấy tỉ lệ nhận dạng khi thay đổi không gian đặc tính qua các trường hợp đặc tính khác nhau: với ngưỡng $D = 30, 56, 120, 504$ lần lượt là 98,26%; 98,16%; 99,65%; 100%. Về tốc độ xử lý qua các lần thử nghiệm cho ra các kết quả: 1,09; 1,16; 2,07; 7,07 (giây). Có thể thấy dù có kết quả chính xác cao nhưng đòi lại thời gian xử lý khá chậm làm hao tổn tài nguyên và chi phí [35].

Cho đến nay vẫn còn nhiều thách thức và vấn đề cần giải quyết trong bài toán nhận dạng khuôn mặt, làm hạn chế khả năng triển khai của hệ thống nhận dạng mặt người trong thực tế. Vì vậy, lĩnh vực nghiên cứu nhận dạng mặt người vẫn đang nhận được sự quan tâm lớn của các nhà khoa học. Nhận dạng khuôn mặt có một ưu điểm đáng kể so với các công nghệ sinh trắc học khác là ảnh có thể được thu nhận từ xa cũng như không cần sự đồng ý cộng tác từ đối tượng.

Công nghệ thông tin đang được ứng dụng trong mọi lĩnh vực của cuộc sống. Với một hệ thống máy tính, chúng ta có thể làm được rất nhiều việc, tiết kiệm thời gian và công sức. Điển hình như công việc xem kết quả học tập của học sinh, sinh viên. Ngày xưa muốn xem kết quả học tập thì phải chờ thầy cô phát sổ liên lạc về mới xem được. Ngày nay công việc này tiện lợi hơn khi muốn xem học sinh, sinh viên chỉ cần đăng nhập vào phần mềm hay trang web tra cứu là có thể xem kết quả học tập. Trong trường hợp này, một ứng dụng tra cứu kết quả học tập bằng cách nhận dạng khuôn mặt học sinh, sinh viên cho phép đưa ra định danh và kết quả học tập cá nhân từ ảnh đầu vào sẽ giúp ích rất nhiều cho học sinh, sinh viên.

Đó cũng là lý do mà chúng tôi chọn đề tài “Tra cứu kết quả học tập sinh viên Hufi qua hình ảnh”. Để có thể cung cấp cho học sinh, sinh viên một phần mềm tra cứu tiện lợi nhất. Trong khoá luận này, nhóm đề xuất sử dụng phương pháp học máy SVM để phân loại hình ảnh, một phương pháp được sử dụng trong nhiều nghiên cứu về thị giác máy tính. Trên cơ sở

đánh giá trên bộ dữ liệu chuẩn YaleFace và YaleFace B (của Yale University) so với các phương pháp trên thì phương pháp do nhóm đề xuất SVM kết hợp với đặc trưng HOG trong nhận dạng khuôn mặt cho kết quả có phần vượt trội hơn. Khóa luận được trình bày trong 4 chương với bố cục như sau:

CHƯƠNG 1: TỔNG QUAN ĐỀ TÀI

Chương này tổng quan về đề tài đang thực hiện. Như hiện trạng, mục tiêu đề tài, nội dung nghiên cứu, phạm vi đề tài, phương pháp nghiên cứu và công nghệ sử dụng.

CHƯƠNG 2: CƠ SỞ LÝ THUYẾT

Chương này trình bày khái quát lý thuyết cơ bản, tìm hiểu một số phương pháp trích chọn đặc trưng ảnh và tìm hiểu một số phương pháp phát hiện và nhận dạng khuôn mặt.

CHƯƠNG 3: PHƯƠNG PHÁP NHẬN DẠNG SINH VIÊN VÀ TRA CỨU KẾT QUẢ

Chương này tìm hiểu chi tiết về phương pháp nhận dạng và tra cứu học tập kết quả dựa trên ảnh sinh viên, tìm hiểu các kỹ thuật validate model SVM.

CHƯƠNG 4: THỰC NGHIỆM VÀ ĐÁNH GIÁ

Xây dựng chương trình thực nghiệm tra cứu kết quả học tập dựa trên ảnh, đánh giá hiệu năng và một số kết quả đạt được.

CHƯƠNG 1 TỔNG QUAN ĐỀ TÀI

1.1 Hiện trạng

Theo số liệu của Bộ GD-ĐT cung cấp, quy mô học sinh cả nước năm học 2020-2021 là 23 triệu học sinh, trong đó học sinh mầm non là gần 5,4 triệu cháu; học sinh phổ thông là gần 17,6 triệu em (trong đó tiểu học trên 8,7 triệu; THCS trên 6 triệu; THPT trên 2,8 triệu). Tăng khoảng 1 triệu em so với năm học 2019-2020 [3]. Như vậy có thể thấy việc tra cứu kết quả học tập của học sinh, sinh viên rất là cần thiết, mà các trang tra cứu thông thường phải cần có mật khẩu mã số của học sinh, sinh viên để có thể đăng nhập vào xem kết quả học tập. Việc phải ghi nhớ những con số hay chuỗi kí tự dài ngoằn để có thể xem được kết quả học tập là rất đáng quan ngại vì nếu quên thì không thể tra cứu được kết quả. Vì vậy định danh theo hình ảnh mang tính trực quan sẽ dễ dàng cho người tra cứu thông tin đi kèm ảnh học sinh, sinh viên sẽ chỉ cần có một bức ảnh của bản thân mình để xem kết quả học tập điều này sẽ không gây khó khăn. Do đó, nhu cầu thật sự đòi hỏi chúng ta phải có một công cụ hỗ trợ cho việc tra cứu kết quả học tập bằng hình ảnh càng sớm càng tốt.

Phương pháp để giải bài toán đó là: Phương pháp dựa trên những đặc trưng của hình ảnh như màu sắc, kết cấu, hình dạng, Dựa trên các đặc trưng đó, phần mềm sẽ tiến hành tìm ra học sinh, sinh viên theo hình ảnh dựa trên thuật toán phân loại SVM. Sau khi tìm ra được học sinh, sinh viên sẽ tiến hành tra cứu kết quả học tập của học sinh, sinh viên đó.

1.2 Mục tiêu và phạm vi đề tài

- Xây dựng phần mềm tra cứu kết quả học tập bằng hình ảnh cho sinh viên Khoa Công nghệ Thông tin trường Đại học Công nghiệp Thực phẩm TP. Hồ Chí Minh.
- Nghiên cứu nhận dạng sinh viên và tra cứu kết quả học tập qua hình ảnh. Các đặc trưng hình ảnh được trích xuất và phân lớp nhằm định danh sinh viên. Sau đó, việc tra cứu kết quả học tập được thực hiện trên một cơ sở dữ liệu đã xây dựng.

1.3 Nội dung nghiên cứu

- Xây dựng một bộ phát hiện khuôn mặt từ hình ảnh bằng MTCNN (Multi-task Cascaded Convolutional Networks).
- Trích xuất đặc trưng ảnh bằng HOG (Histogram of Oriented Gradient).
- Xây dựng một model phân loại bằng SVM (Support Vector Machine).
- Xây dựng cơ sở dữ liệu quản lý kết quả học tập sinh viên.
- Xây dựng web api truyền dữ liệu qua lại giữa C# và Python.
- Xây dựng ứng dụng nhận diện khuôn mặt sinh viên và trả về kết quả học tập của sinh viên đó.

1.4 Phương pháp nghiên cứu

- Nghiên cứu các phương pháp phát hiện khuôn mặt: MTCNN, Dlib,
- Nghiên cứu các phương pháp rút trích đặc trưng hình ảnh: HOG, Landmark,
- Nghiên cứu các phương pháp phân loại: SVM, KNN,
- Nghiên cứu xây dựng api bằng Flask và Flask-cors.

1.5 Công nghệ sử dụng

- Ngôn ngữ lập trình: C#, và Python.
- Môi trường lập trình: MS Visual Studio và Pycharm.
- Các thư viện sử dụng: OpenCV, Flask, Flask-cors, Scikit-learning, TensorFlow, Kfold,

CHƯƠNG 2 CƠ SỞ LÝ THUYẾT

2.1 Giới thiệu về ảnh số

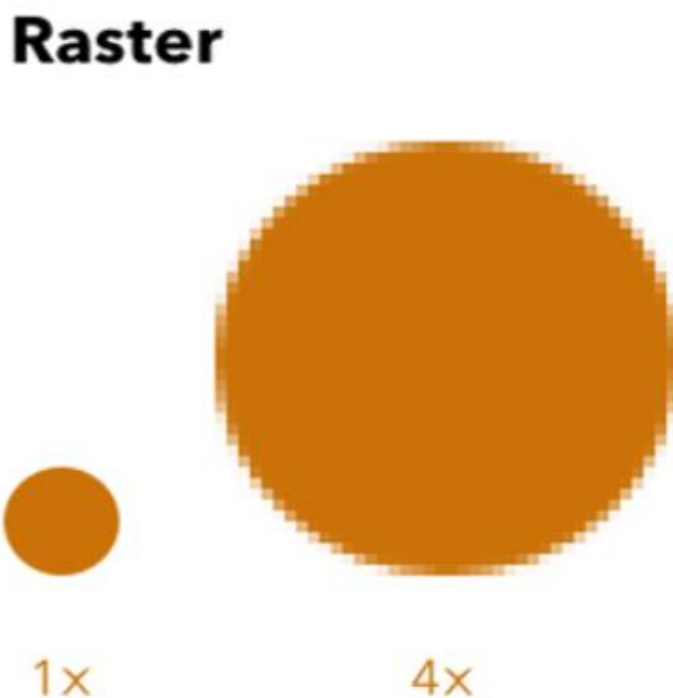
2.1.1 Khái niệm

Ảnh số (digital image) là tập hợp hữu hạn các điểm ảnh có thể được biểu diễn dưới dạng một ma trận 2 chiều, mỗi phần tử của ma trận tương ứng với một điểm ảnh. Mỗi phần tử này được gọi là một pixel [23]. Ảnh số có hai loại: ảnh raster và ảnh vector.

❖ Ảnh raster

Ảnh Raster là một tập hợp hữu hạn các giá trị số, gọi là điểm ảnh (pixel - picture element). Thông thường một hình ảnh được chia thành các hàng và cột chứa điểm ảnh. Điểm ảnh là thành phần bé nhất biểu diễn ảnh, có giá trị số biểu diễn màu sắc, độ sáng, ... của một thành phần trong bức ảnh.

Ảnh raster thường được thu từ camera, các máy chiếu, chụp, quét, ... và chính là đối tượng chính của xử lý ảnh và thị giác máy tính [24].



Hình 2-1 Ảnh raster

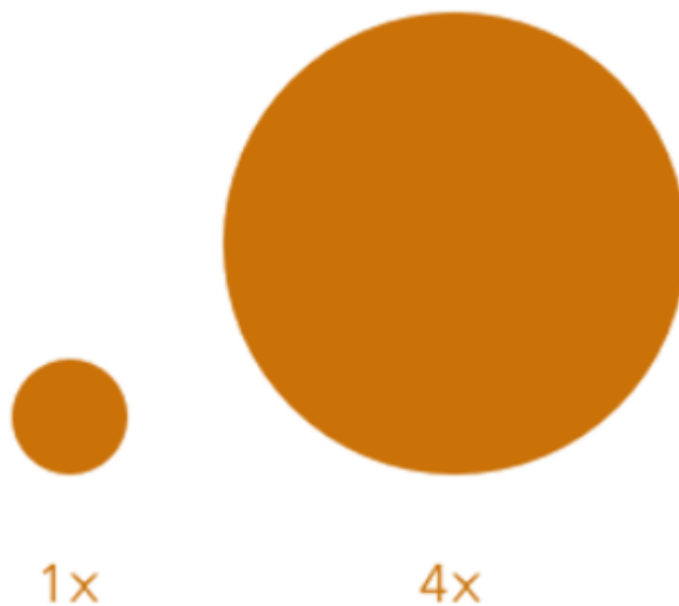
❖ Ảnh vector

Ảnh vector là loại ảnh tạo thành từ các thành phần đơn giản của hình học như điểm, đường thẳng, hình khối, Thay vì được lưu lại thành các ma trận điểm ảnh như ảnh raster, ảnh vector được biểu diễn dưới dạng tọa độ của các thành phần trong ảnh.

Chính điều này đã tạo nên sự đặc biệt của ảnh vector, khiến nó có thể được kéo giãn, thu nhỏ tùy ý mà không bị vỡ, không xuất hiện răng cưa như ảnh raster. Dữ liệu trong ảnh vector nhỏ, do vậy thường tiết kiệm dung lượng lưu trữ hơn ảnh raster. Tuy thế, màu sắc trong ảnh vector nhìn không thật, sắc độ ít tinh tế hơn ảnh raster.

Thông thường người ta sử dụng ảnh vector trong thiết kế các logo, banner, giao diện đồ họa, Loại ảnh này gần như không xuất hiện khi đề cập đến xử lý ảnh/thị giác máy tính [24].

Vector



Hình 2-2 Ảnh vector

2.1.2 Kích thước ảnh

Kích thước hình ảnh hay độ phân giải là thuật ngữ được đưa ra để mô tả chiều cao và chiều rộng của hình ảnh tính bằng pixel. Kích thước hình ảnh tối đa được xác định bằng số megapixel của một máy ảnh nhất định.



Hình 2-3 Kích thước ảnh

2.1.3 Ảnh xám

Ảnh xám (Gray image) hay còn gọi là ảnh đơn sắc (Monochromatic), mỗi giá trị điểm ảnh (Pixel) trong ma trận điểm ảnh mang giá trị từ 0 đến 255. Trong không gian màu RGB, để có một ảnh xám cần có phải có giá trị kênh màu Red $(x, y) = \text{Green}(x, y) = \text{Blue}(x, y)$ (với x, y lần lượt là tọa độ của điểm ảnh).



Hình 2-4 Ảnh xám

2.1.4 Ảnh màu

Ảnh màu là một ma trận các pixel mà mỗi pixel biểu diễn một điểm màu. Mỗi điểm màu được biểu diễn bằng bộ 3 số (r, g, b). Để tiện cho việc xử lý ảnh thì sẽ tách ma trận pixel ra 3 channel red, green, blue.



Hình 2-5 Ảnh màu

2.2 Phát hiện khuôn mặt (Face Detection)

2.2.1 Khái niệm

Phát hiện khuôn mặt là một công nghệ máy tính đang được sử dụng trong nhiều ứng dụng khác nhau để xác định khuôn mặt người trong hình ảnh kỹ thuật số. Phát hiện khuôn mặt cũng đề cập đến quá trình tâm lý mà con người xác định vị trí và quan sát khuôn mặt trong một cảnh trực quan.

Phát hiện khuôn mặt có thể được coi là một trường hợp cụ thể của phát hiện lớp đối tượng (Object detection) đối tượng cụ thể ở đây là khuôn mặt. Phát hiện khuôn mặt chỉ trả lời được hai câu hỏi, có khuôn mặt nào trong hình ảnh hoặc video được thu thập không? Và nó đã được đặt ở đâu?

2.2.2 Phương pháp MTCNN

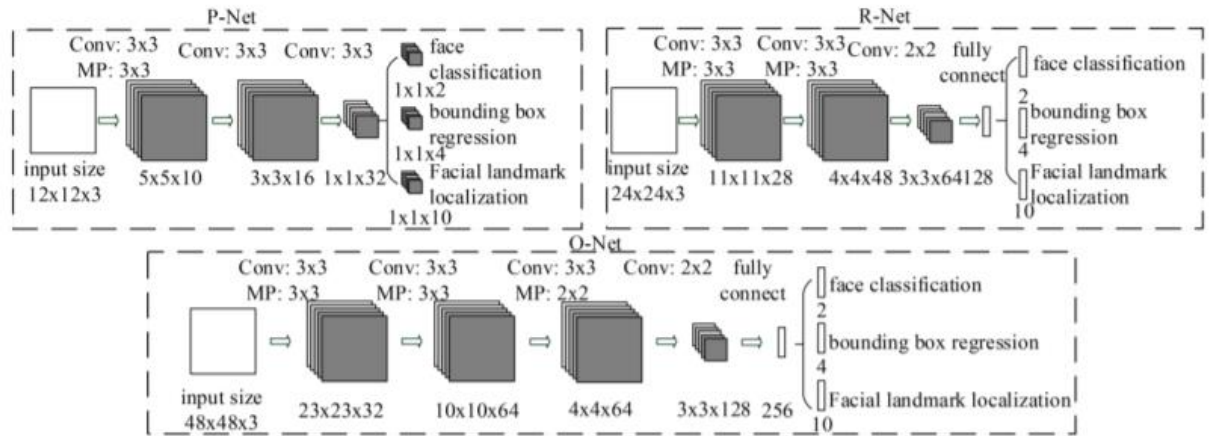
❖ MTCNN là gì?

MTCNN (Multi-task Cascaded Convolutional Networks) là một phương pháp phát hiện khuôn mặt được đề xuất bởi Zhang và cộng sự vào năm 2016. Bao gồm 3 mạng CNN xếp chồng và đồng thời hoạt động khi phát hiện khuôn mặt. Mỗi mạng có cấu trúc khác nhau và đảm nhiệm vai trò khác nhau trong xử lý. Đầu ra của MTCNN là vị trí khuôn mặt và các điểm trên mặt như: mắt, mũi, miệng. Phương pháp này có thể đạt được tỷ lệ chính xác 95,4% trên FDDB [25].

Thư viện MTCNN được viết bởi một người dùng github tên là ipazc, thư viện được triển khai dựa theo bài báo của Zhang.

❖ Quy trình xử lý của MTCNN

MTCNN sử dụng cấu trúc xếp tầng gồm 3 mạng CNN với ba giai đoạn (P-Net, R-Net và O-Net) mạng P-Net đầu tiên là một mạng dạng FCN (Fully convolutional network), và hai mạng sau R-Net và O-Net là CNN thông thường. Kích thước của hình ảnh đầu vào của MTCNN có thể có kích thước bất kỳ. Với một hình ảnh, có thể thay đổi kích thước nó thành các quy mô khác nhau để xây dựng một kim tự tháp hình ảnh (Image Pyramid) như là đầu vào của theo khuôn khổ xếp tầng ba giai đoạn.



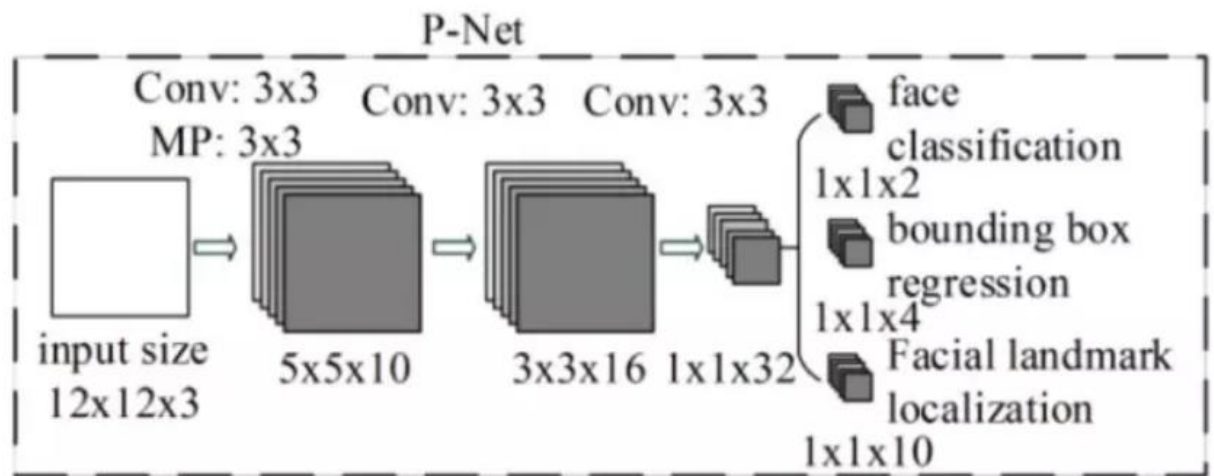
Hình 2-6 Cấu trúc của MTCNN

P-Net: Proposal network dự đoán các vùng trong bức ảnh có thể là khuôn mặt dự đoán khá nhanh nhưng lại thiếu chính xác (có nhiều vùng không phải là khuôn mặt). Nó là một mạng được kết nối đầy đủ đơn giản (FCN) có tên là Mạng đề xuất, một mạng để đề xuất diện tích khuôn mặt xử lý hình ảnh theo các bước sau:

Bước 1: Trích xuất các đặc điểm khuôn mặt ban đầu với FCN để tìm các Bounding-Box

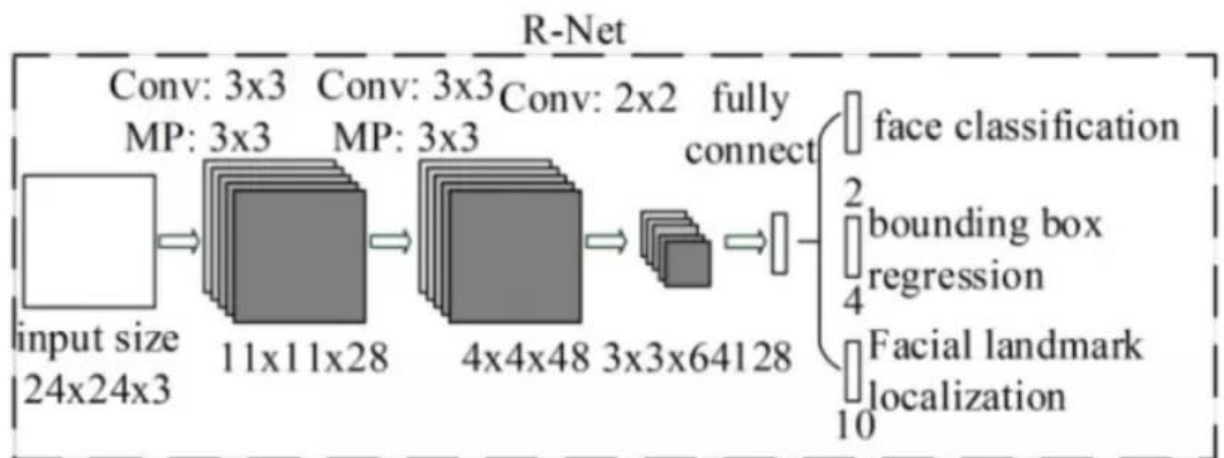
Bước 2: Nhập các tính năng vào ba lớp tích hợp để xác định xem đó có phải là khuôn mặt người hay không bằng cách sử dụng bộ phân loại khuôn mặt.

Bước 3: Lọc hầu hết các cửa sổ ứng cử viên với hồi quy Bounding-Box (Bounding-Box Regression) và NMS (Non-Maximum Suppression), có được các vị trí có thể có khuôn mặt bằng bộ định vị cho các đặc điểm khuôn mặt và tạo vùng đề xuất các khuôn mặt [26].



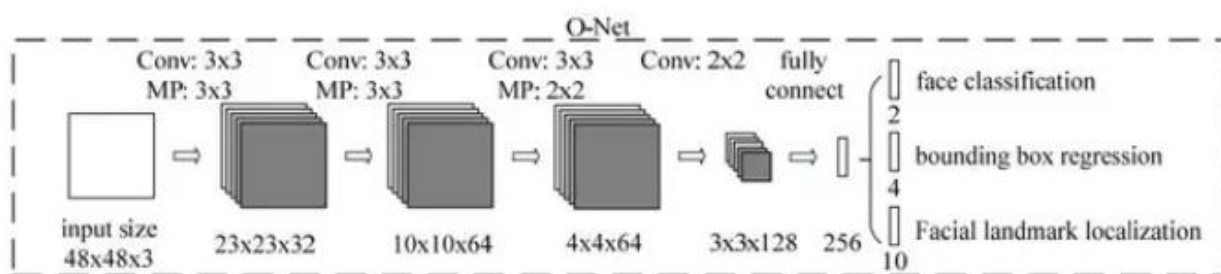
Hình 2-7 Cấu trúc mạng P-Net

R-Net: Refine network là một mạng tích chập phức tạp hơn có tên là Mạng tinh chỉnh, một mạng để lọc các cửa sổ khuôn mặt được đề xuất từ P-Net với độ chính xác cao và tối ưu hóa chúng. Vì nhận được các cửa sổ vùng mặt chính xác và đáng tin cậy hơn, R-Net bổ sung thêm 128 FCN sau lần chập cuối cùng để lấy được nhiều đặc trưng hình ảnh hơn các tính năng $1 \times 1 \times 32$ của P-Net. Nó sử dụng các quy tắc chặt chẽ hơn để chọn cẩn thận hơn và xóa các đề xuất khuôn mặt có đặc trưng không đủ tốt. Cuối cùng, R-Net cũng tối ưu hóa kết quả đầu ra với hồi quy Bounding-Box (Bounding-Box Regression) và NMS (Non-Maximum Suppression) [26].



Hình 2-8 Cấu trúc mạng R-Net

O-Net: Output network là một mạng tích chập tương đối phức tạp có tên là Mạng đầu ra, với nhiều hơn một lớp tích hợp so với R-Net, dùng để xuất ra năm đặc điểm cuối cùng trên khuôn mặt bằng cách giám sát các vùng mặt và chỉnh sửa các đường nét trên khuôn mặt. O-Net có nhiều đặc trưng trên khuôn mặt hơn R-Net và cuối cùng thêm 256 FCN để có thể giữ được nhiều đặc trưng hình ảnh hơn. Dựa trên tất cả những đặc điểm này để có độ chính xác cao hơn, O-Net đánh giá khuôn mặt, hồi quy Bounding-Box (Bounding-Box Regression) khuôn mặt và xác định vị trí các đặc điểm trên khuôn mặt một lần nữa. Sau cùng nó sẽ xuất ra tọa độ góc trên cùng bên trái và góc dưới bên phải cùng với năm đặc điểm trên khuôn mặt [26].



Hình 2-9 Cấu trúc mạng O-Net

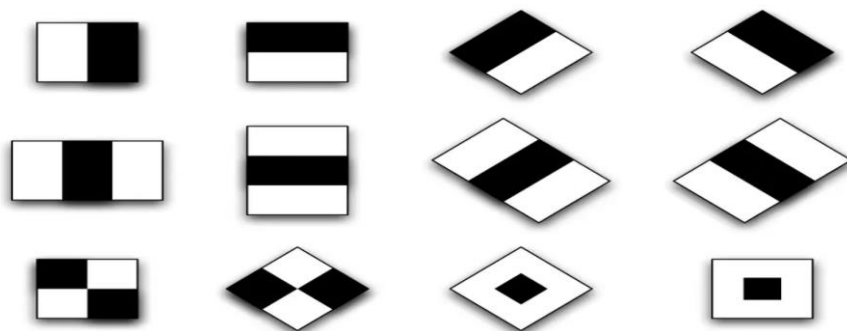
2.2.3 Phương pháp Haar-like - Adaboost

Haar-like - Adaboost (viết tắt HA) của hai tác giả Paul Viola và Michael J. Jones là phương pháp xác định mặt người dựa theo hướng tiếp cận trên diện mạo. Cụ thể hơn, hệ thống nhận dạng khuôn mặt sẽ được học các mẫu của khuôn mặt từ một tập ảnh mẫu. Sau khi quá trình học (training) thực hiện xong, hệ thống sẽ rút ra được những tham số để phục vụ cho quá trình nhận dạng. Vì vậy tiếp cận trên diện mạo còn được biết đến với tên gọi tiếp cận theo phương pháp học máy. Về tổng quan, phương pháp HA được xây dựng dựa trên sự kết hợp, lắp ghép của 2 thành phần, đó là:

❖ Đặc trưng Haar-like

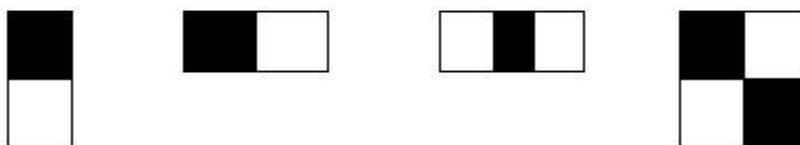
Các đặc trưng Haar-like là các đặc trưng hình ảnh kỹ thuật số được sử dụng trong nhận dạng đối tượng. Chúng có tên gọi tương đồng trực quan với các wavelet Haar [37]

và được sử dụng trong máy dò khuôn mặt thời gian thực đầu tiên. Các đặc trưng Haar-like là những hình chữ nhật được phân thành các vùng khác nhau.



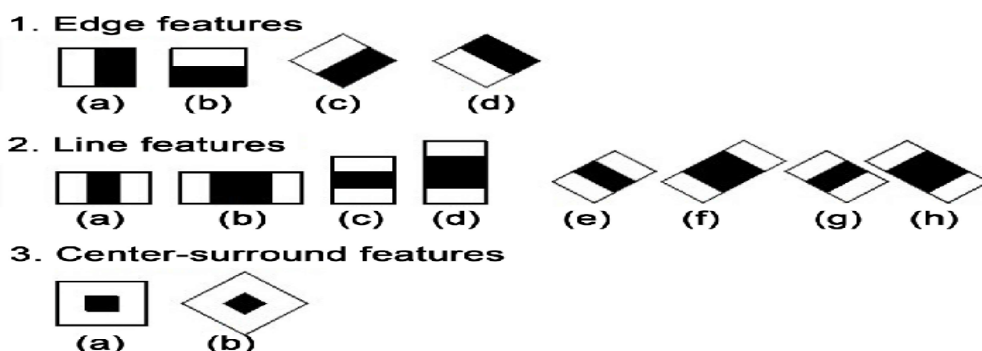
Hình 2-10 Đặc trưng Haar – like

Ưu điểm chính của đặc trưng Haar-like so với hầu hết các đặc trưng khác là tốc độ tính toán của nó. Do việc sử dụng hình ảnh tích phân (integral image), đặc trưng Haar-like ở bất kỳ kích thước nào có thể được tính toán trong thời gian không đổi (xấp xỉ 60 lệnh của bộ vi xử lý cho đặc trưng 2 hình chữ nhật). Có 4 đặc trưng cơ bản để xác định khuôn mặt người:



Hình 2-11 Đặc trưng Haar-like cơ bản để xác định khuôn mặt

Và các đặc trưng Haar-like này được mở rộng ra và được chia làm 3 tập đặc trưng như sau:

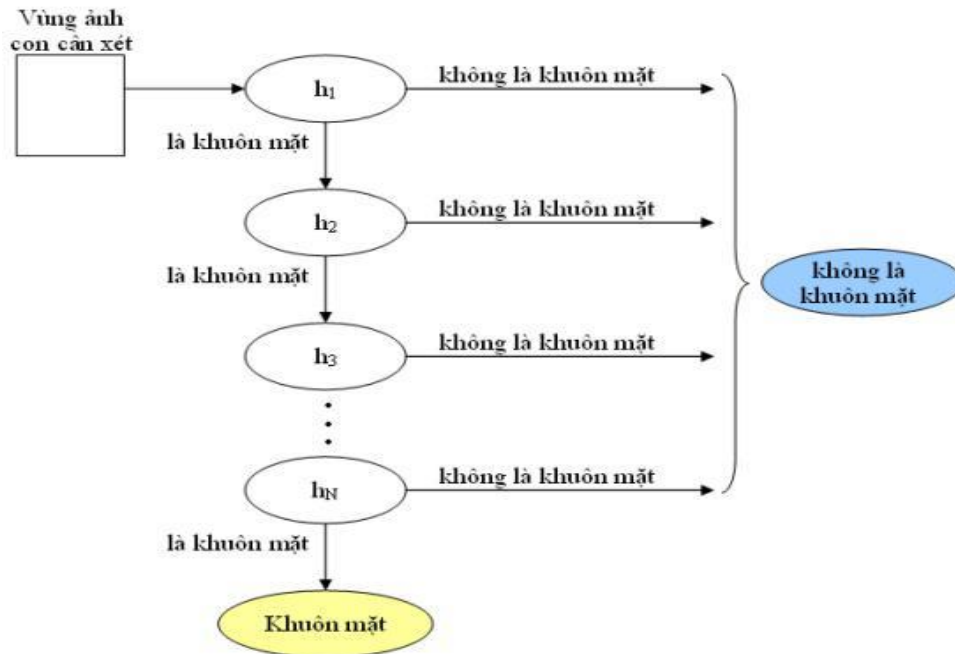


Hình 2-12 Đặc trưng Haar – like mở rộng

❖ AdaBoost (Adaptive Boosting)

AdaBoost là một bộ phân loại mạnh phi tuyến phức dựa trên hướng tiếp cận boosting được Freund và Schapire đưa ra vào năm 1995. Adaboost cũng hoạt động trên nguyên tắc kết hợp tuyến tính các bộ phân loại yếu để hình thành một trong các bộ phân loại mạnh.

AdaBoost lựa chọn những đặc trưng tốt nhất, huấn luyện bộ phân loại mạnh mẽ dựa vào tổ hợp tuyến tính của bộ phân loại yếu hơn. Viola và Jones dùng AdaBoost kết hợp các bộ phân loại yếu sử dụng các đặc trưng Haar-like theo mô hình phân tầng (cascade) như sau [17]:



Trong đó, h_k là các bộ phân loại yếu, được biểu diễn như sau:

$$h_k = \begin{cases} 1 & \text{nếu } P_k f_k(x) < P_k \theta_k \\ 0 & \end{cases}$$

x : cửa sổ con cần xét

θ_k : ngưỡng

f_k : giá trị đặc trưng của Haar-like

P_k : hệ số quyết định chiều của phương trình

AdaBoost sẽ kết hợp các bộ phân loại yếu thành bộ phân loại mạnh như sau:

$$H(x) = \sum (\alpha_1 h_1(x) + \alpha_2 h_2(x) + \dots + \alpha_n h_n(x))$$

Với: $\alpha \geq 0$ hệ số chuẩn hóa cho bộ phân loại yếu

2.3 Trích xuất đặc trưng (Feature Extraction)

2.3.1 Khái niệm

Trích xuất đặc trưng là một phép biến đổi để loại ra những dữ liệu nhiễu (noise), và để đưa dữ liệu thô với số chiều khác nhau về cùng một chuẩn (cùng là các vector hoặc ma trận). Dữ liệu chuẩn mới này phải đảm bảo giữ được những thông tin đặc trưng (features) cho dữ liệu thô ban đầu.

2.3.2 Phương pháp HOG

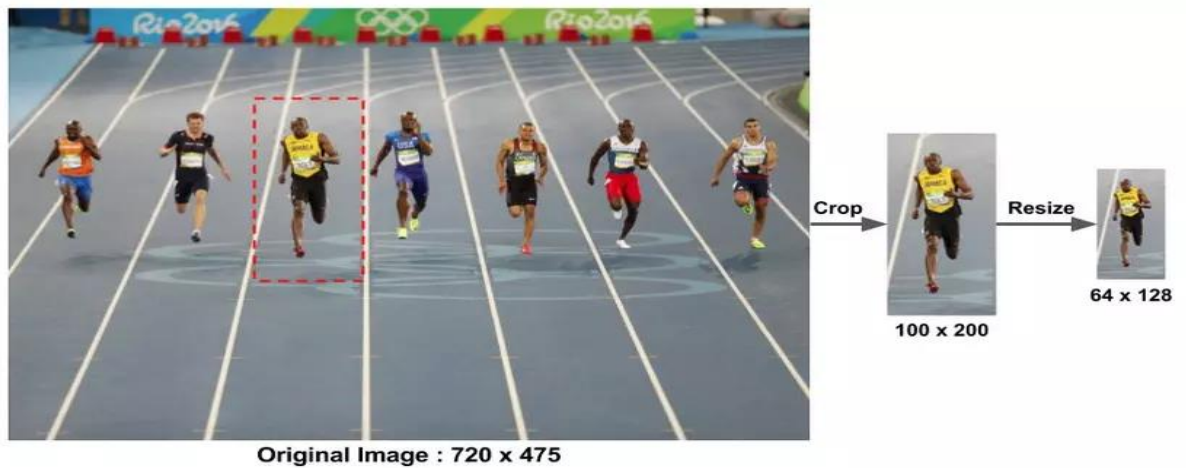
HOG là viết tắt của Histogram of Oriented Gradient - một loại bộ mô tả đặc trưng (feature descriptor). Mục đích của bộ mô tả đặc trưng là trừu tượng hóa đối tượng bằng cách trích xuất ra những đặc trưng của đối tượng đó và bỏ đi những thông tin không hữu ích. Vì vậy, HOG được sử dụng chủ yếu để mô tả hình dạng và sự xuất hiện của một đối tượng trong ảnh [14].

Có 5 bước cơ bản để xây dựng một vector HOG cho hình ảnh, bao gồm:

- Tiền xử lý
- Tính gradient (gradient của một trường vô hướng là một trường vector có chiều hướng về phía mức độ tăng lớn nhất của trường vô hướng, và có độ lớn là mức độ thay đổi lớn nhất.)
- Tính vector đặc trưng cho từng ô (cells)
- Chuẩn hóa khối (blocks)
- Tính toán vector đặc trưng HOG

❖ Tiền xử lý

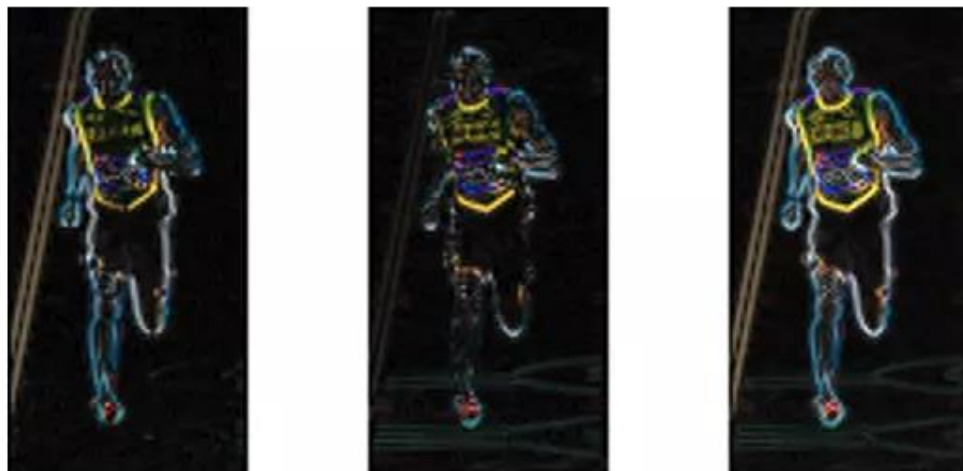
Resize kích thước tất cả các hình ảnh trong tập dữ liệu về một kích thước chung.



Hình 2-13 Resize kích thước tất cả các hình ảnh

❖ Tính gradient

Được thực hiện bằng hai phép nhân chập ảnh gốc với 2 chiều, tương ứng với các toán tử lấy đạo hàm theo hai hướng Ox và Oy. Đối với hình ảnh màu, gradient của ba kênh (red, green và blue) được đánh giá. Độ lớn của gradient tại một điểm ảnh là giá trị lớn nhất của cường độ gradient của ba kênh, và góc là góc tương ứng với gradient tối đa kết quả thu được sẽ là:



Left : Absolute value of x-gradient. Center : Absolute value of y-gradient.
Right : Magnitude of gradient.

Hình 2-14 Phép nhân chập ảnh gốc với 2 chiều

❖ Tính vector đặc trưng cho từng ô

Chia hình ảnh thành các block, mỗi block lại chia đều thành các cell.

Sau khi xác định số block và kích thước mỗi block, cell, để tính toán vector đặc trưng cho từng cell, chúng ta cần:

- Chia không gian hướng thành p bin (số chiều vector đặc trưng của ô).
- Rời rạc hóa góc hướng nghiêng tại mỗi điểm ảnh vào trong các bin.

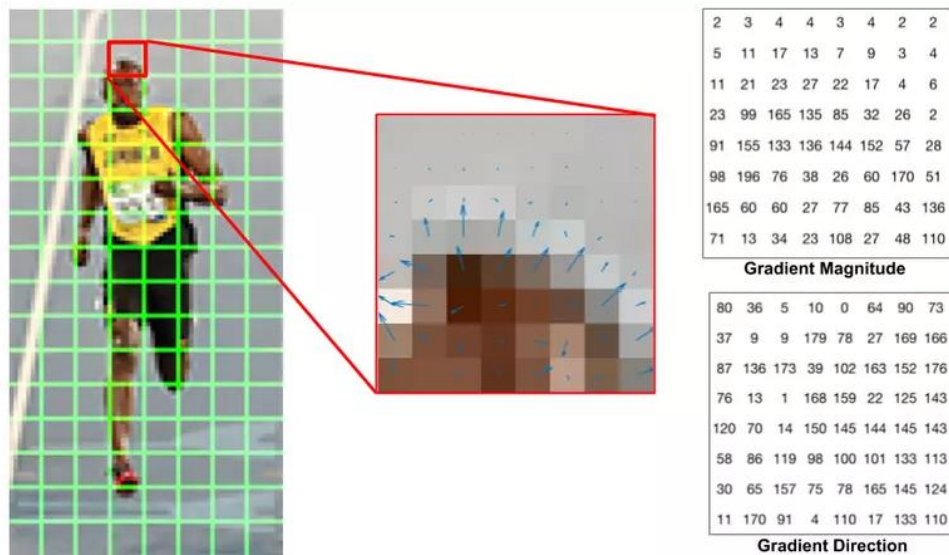
Giá trị bin được định lượng bởi tổng cường độ biến thiên của các pixels thuộc về bin đó. Sau khi tính toán đặc trưng ô, ta sẽ nối các vector đặc trưng ô để thu được vector đặc trưng khối.



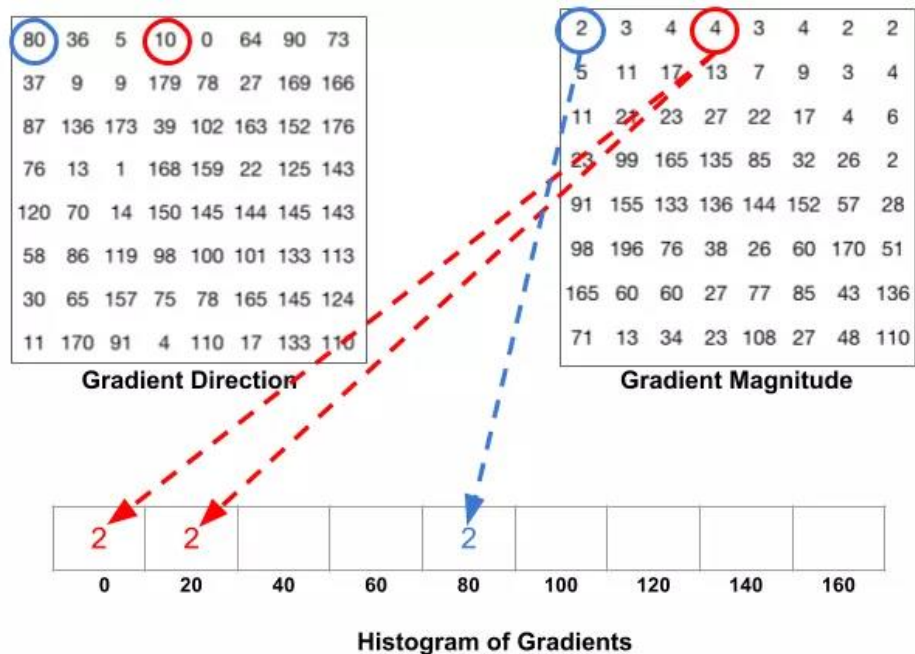
Hình 2-15 Chia hình ảnh thành các block

Tiếp theo, tiến hành tính toán đặc trưng HOG tại mỗi cell sử dụng không gian hướng 9bin, trường hợp “unsigned-HOG”. Hướng gradient sẽ chạy trong khoảng 0 độ đến 180 độ, trung bình 20 độ mỗi bin.

Tại mỗi cell, xây dựng một biểu đồ cường độ gradient bằng cách vote các pixel vào biểu đồ. Trọng số vote của mỗi pixel phụ thuộc hướng và cường độ gradient (được tính toán từ bước 2) của pixel đó. Ví dụ:



Hình 2-16 Tính toán đặc trưng HOG tại mỗi cell



Hình 2-17 Xây dựng một biểu đồ cường độ gradient

❖ Chuẩn hóa khối (blocks)

Để tăng cường hiệu năng nhận dạng, các histogram cục bộ sẽ được chuẩn hóa về độ tương phản bằng cách tính một ngưỡng cường độ trong một khối và sử dụng giá trị đó để chuẩn hóa tất cả các ô trong khối. Kết quả sau bước chuẩn hóa sẽ là một vector đặc trưng có tính bất biến cao hơn đối với các thay đổi về điều kiện ánh sáng.

Lưu ý cách hiển thị hình ảnh thứ ba tăng độ tương phản. Hiệu quả của phép nhân là các điểm ảnh sáng trở nên sáng hơn nhiều trong khi các điểm ảnh tối chỉ trở nên sáng hơn một chút, do đó làm tăng độ tương phản giữa phần sáng và phần tối của hình ảnh.

❖ **Tính toán vector đặc trưng HOG**

Với mỗi hình ảnh kích thước 64×128 , chia thành các block 16×16 chồng nhau, sẽ có 7 block ngang và 15 block dọc, nên sẽ có $7 \times 15 = 105$ blocks.

Mỗi block gồm 4 cell. Khi áp dụng biểu đồ 9-bin cho mỗi cell, mỗi block sẽ được đại diện bởi một vector có kích thước 36×1 .

Vì vậy, khi nối tất cả các vector trong một block lại với nhau, ta sẽ thu được vector đặc trưng HOG của ảnh có kích thước $105 \times 36 \times 1 = 3780 \times 1$.

2.3.3 Phương pháp Facial landmark

Facial landmark được sử dụng để xác định vị trí và thể hiện các vùng nổi bật trên khuôn mặt, chẳng hạn như mắt, lông mày, mũi, đường viền hàm, miệng, ...

Đây là một kỹ thuật đã được áp dụng cho các ứng dụng như căn chỉnh khuôn mặt, ước tính tư thế đầu, hoán đổi khuôn mặt, phát hiện chớp mắt, phát hiện buồn ngủ, ...

Việc xác định facial landmark gồm có hai bước:

- Bước 1: Xác định được vị trí khuôn mặt trong bức ảnh
- Bước 2: Xác định được các điểm tạo nên cấu trúc của khuôn mặt

Việc xác định vị trí khuôn mặt có thể được thực hiện bằng nhiều cách từ đơn giản như thuật toán Haar cascades đến phức tạp như các thuật toán dựa trên deep-learning. Tuy nhiên dù sử dụng thuật toán nào, mục đích cuối cùng là ta sẽ thu được một vùng (thường là hình vuông) được xác định bởi tọa độ (x, y) bao quanh khuôn mặt trong bức ảnh.

Sau khi xác định được khuôn mặt trong bức ảnh, chúng ta sẽ xác định cấu trúc của khuôn mặt. Có rất nhiều kiểu cấu trúc khuôn mặt khác nhau nhưng về cơ bản, chúng ta sẽ phải xác định được những phần sau:

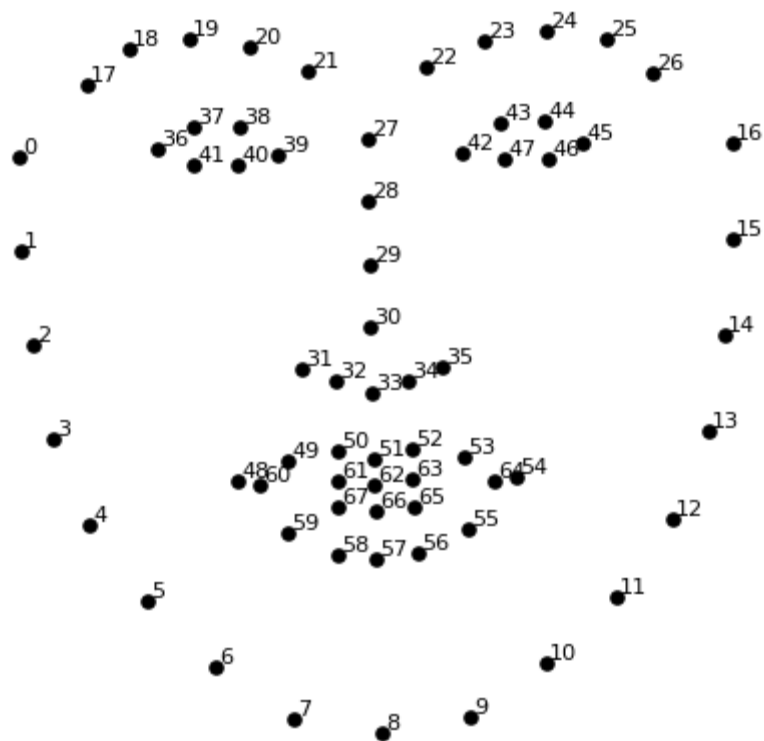
- Miệng
- Lông mày phải
- Lông mày trái
- Mắt phải
- Mắt trái
- Mũi
- Hàm

❖ **Tìm hiểu bộ xác định facial landmark của dlib**

Bộ xác định facial landmark của dlib là cài đặt của thuật toán được mô tả trong bài báo One Millisecond Face Alignment with an Ensemble of Regression Trees của Kazemi và Sullivan (2014) [16].

Bộ xác định facial landmark này sẽ xác định 68 điểm chính theo tọa độ (x, y) cấu tạo nên khuôn mặt người như hình bên dưới.

- Mouth point = 48-67
- Right brow point = 17-21
- Left brow point = 22-26
- Right eye point = 36-41
- Left eye point = 42-47
- Nose point = 27-35
- Jaw point = 0-16



Hình 2-18 68 điểm landmark

68 mốc sẽ xác định vị trí trên mọi khuôn mặt. Hình ảnh này được tạo bởi Brandon Amos của CMU, người làm việc trên OpenFace.

Để có thể xác định được 68 điểm này trên khuôn mặt người, bộ xác định facial landmark của dlib được huấn luyện với bộ dữ liệu iBUG 300-W.

Xác định được càng nhiều điểm landmark thì khuôn mặt người càng cụ thể, rõ ràng và chính xác hơn. Ví dụ FaceID đã sử dụng bộ cảm biến hồng ngoại với 30 triệu điểm để dựng lên cấu trúc khuôn mặt 3D của người sử dụng điện thoại.

2.4 Nhận dạng khuôn mặt (Face Recognition)

2.4.1 Khái niệm

Hệ thống nhận dạng khuôn mặt là một ứng dụng máy tính tự động xác định hoặc nhận dạng một người nào đó từ một bức hình ảnh kỹ thuật số hoặc một khung hình video từ một nguồn video. Một trong những cách để thực hiện điều này là so sánh các đặc điểm khuôn mặt chọn trước từ hình ảnh và một cơ sở dữ liệu về khuôn mặt. Hệ thống này

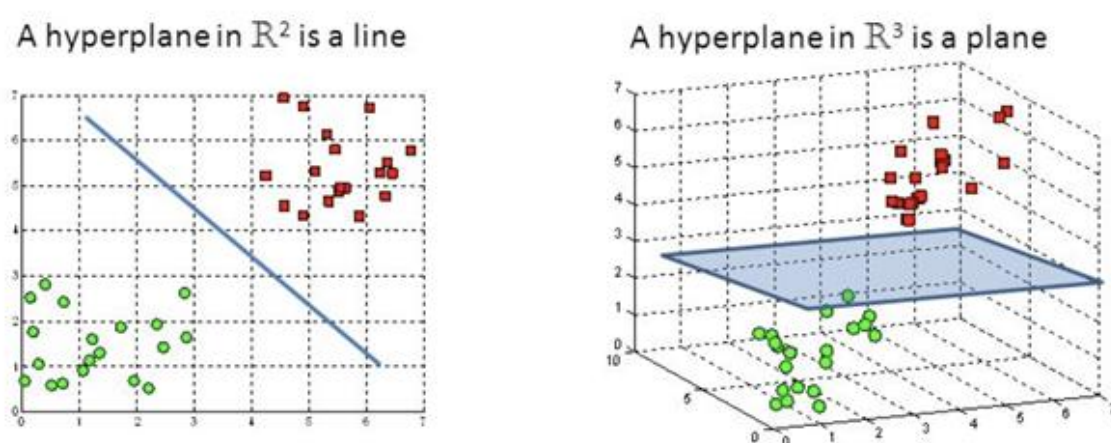
thường được sử dụng trong các hệ thống an ninh và có thể được so sánh với các dạng sinh trắc học khác như các hệ thống nhận dạng vân tay hay tròng mắt.

2.4.2 SVM (Support Vector Machine)

SVM (Support Vector Machine) là một thuật toán học máy có giám sát được sử dụng rất phổ biến ngày nay trong các bài toán phân lớp (classification) hay hồi quy (Regression). SVM được đề xuất bởi Vladimir N. Vapnik và các đồng nghiệp của ông vào năm 1963 tại Nga và sau đó trở nên phổ biến trong những năm 90 nhờ ứng dụng giải quyết các bài toán phi tuyến tính (nonlinear) bằng phương pháp Kernel Trick.

❖ Ý tưởng hoạt động của SVM

Ý tưởng hoạt động của SVM là tìm một siêu phẳng (hyper plane) để phân tách các điểm dữ liệu. Siêu phẳng này sẽ chia không gian thành các miền khác nhau và mỗi miền sẽ chứa một loại dữ liệu. Siêu phẳng của không gian n chiều là một không gian con $n-1$ chiều của nó. Một siêu phẳng trong không gian tách không gian đó thành hai nửa không gian.

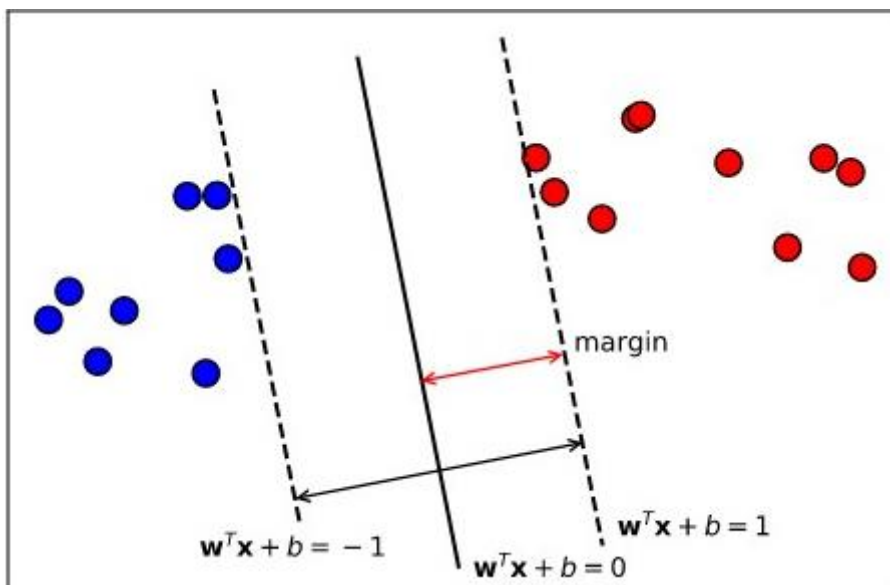


Hình 2-19 Ví dụ, trong không gian 3 chiều, siêu phẳng chính là mặt phẳng 2 chiều. Trong không gian 2 chiều, siêu phẳng là đường thẳng 1 chiều.

❖ SVM phân lớp tuyến tính

Phương pháp này tìm ra một siêu phẳng quyết định tốt nhất có thể chia các điểm trên không gian này thành hai lớp riêng biệt. Chất lượng của siêu phẳng này được quyết định bởi khoảng cách (gọi là biên) của điểm dữ liệu gần nhất của mỗi lớp đến mặt phẳng này.

Khi đó, khoảng cách biên càng lớn thì mặt phẳng quyết định càng tốt, đồng thời việc phân loại càng chính xác.



Hình 2-20 Ảnh minh họa tìm biên lớn nhất

Xét ví dụ phân lớp nhị phân với tập dữ liệu như sau:

$$D = \{(x_i, y_i) \mid x_i \in R^p, y_i \in \{-1, 1\}\}_{i=1}^n$$

với y_i mang giá trị 1 hoặc -1 , xác định lớp của điểm x_i . Mỗi x_i là một vector thực p -chiều. Ta cần tìm siêu phẳng có lề lớn nhất chia tách các điểm có $y_i = 1$ và các điểm có $y_i = -1$. Mỗi siêu phẳng đều có thể được viết dưới dạng một tập hợp các điểm x thỏa mãn

$$w \cdot x + b = 0$$

❖ Biên mềm

Năm 1995, Corinna Cortes và Vladimir N. Vapnik đề xuất một ý tưởng mới cho phép thuật toán gán nhãn sai cho một số ví dụ luyện tập. Nếu không tồn tại siêu phẳng nào phân tách được hai lớp dữ liệu, thì thuật toán lề mềm sẽ chọn một siêu phẳng phân tách các ví dụ luyện tập tốt nhất có thể, và đồng thời cực đại hóa khoảng cách giữa siêu

phẳng với các ví dụ được gán đúng nhãn. Phương pháp này sử dụng các biến bù ξ_i dùng để đo độ sai lệch của ví dụ x_i [29]

$$y_i(w \cdot x_i - b) \geq 1 - \xi_i \quad 1 \leq i \leq n$$

Trong đó $\xi_i \geq 0$ là các biến phụ không âm. $C \sum_{i=1}^n \xi_i$ được thêm vào hàm tối ưu hoá:

$$\min_{w, \xi, b} \left\{ \frac{1}{2} \|w\|^2 + C \sum_{i=1}^n \xi_i \right\}, \text{ với ràng buộc: } y_i(\langle w, x_i \rangle + b) \geq 1 - \xi_i, \xi_i \geq 0$$

Hằng số $C > 0$ thiết lập mức độ quan trọng của việc cực đại biên và giảm số lượng biến phụ ξ_i .

❖ SVM phân lớp phi tuyến

Trong trường hợp dữ liệu không thể tách rời tuyến tính (Letsche and Berry, 2001) có thể ánh xạ dữ liệu vào một không gian có số chiều lớn hơn bằng cách sử dụng một hàm nhân K (kernel). Một số hàm nhân thường dùng như:

- Hàm tuyến tính có dạng $K(x, y) = x \cdot y$
- Hàm đa thức có dạng $K(x, y) = (x \cdot y + 1)^d$
- Hàm RBF (Radial Basis Function) có dạng $K(x, y) = e^{\frac{-|x-y|^2}{2a^2}}$

❖ Tham số của SVM

SVM cũng có một số các siêu tham số (như giá trị C hoặc gamma để sử dụng) và việc tìm kiếm siêu tham số tối ưu là một nhiệm vụ rất khó giải quyết. Nhưng nó có thể được tìm thấy bằng cách thử tất cả các kết hợp và xem thông số nào hoạt động tốt nhất. Phương pháp này được gọi là Gridsearch (GridSearchCV).

2.4.3 KNN (*K Nearest Neighbors*)

KNN là một trong những thuật toán học có giám sát đơn giản nhất (mà hiệu quả trong một vài trường hợp) trong học máy. Khi training, thuật toán này không học một điều gì từ dữ liệu training (đây cũng là lý do thuật toán này được xếp vào loại lazy learning), mọi tính toán được thực hiện khi nó cần dự đoán kết quả của dữ liệu mới.

❖ Cách thức hoạt động của KNN

Thuật toán KNN cho rằng những dữ liệu tương tự nhau sẽ tồn tại gần nhau trong một không gian, từ đó công việc của chúng ta là sẽ tìm k điểm gần với dữ liệu cần kiểm tra nhất. Việc tìm khoảng cách giữa 2 điểm cũng có nhiều công thức có thể sử dụng, tùy trường hợp mà chúng ta lựa chọn cho phù hợp. Đây là 3 cách cơ bản để tính khoảng cách 2 điểm dữ liệu x, y có k thuộc tính:

Distance functions

Euclidean	$\sqrt{\sum_{i=1}^k (x_i - y_i)^2}$
Manhattan	$\sum_{i=1}^k x_i - y_i $
Minkowski	$\left(\sum_{i=1}^k (x_i - y_i)^q \right)^{1/q}$

Hình 2-21 Cách tính khoảng cách

2.4.4 Cây quyết định (Decision Tree)

Cây quyết định (Decision Tree) là một cây phân cấp có cấu trúc được dùng để phân lớp các đối tượng dựa vào dãy các luật. Các thuộc tính của đối tượng có thể thuộc các kiểu dữ liệu khác nhau như Nhị phân (Binary), Định danh (Nominal), Thứ tự (Ordinal), Số lượng (Quantitative) trong khi đó thuộc tính phân lớp phải có kiểu dữ liệu là Binary hoặc Ordinal.

❖ Thuật toán ID3

ID3 (J. R. Quinlan 1993) sử dụng phương pháp tham lam tìm kiếm từ trên xuống thông qua không gian của các nhánh có thể không có backtracking. ID3 sử dụng Entropy và Information Gain để xây dựng một cây quyết định.

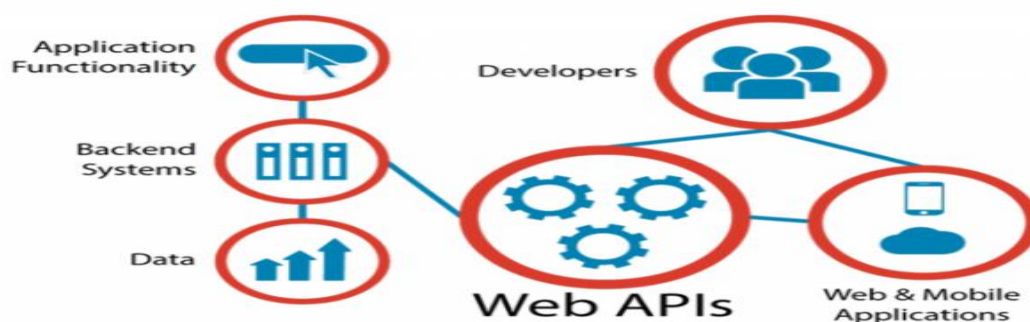
2.5 Giới thiệu về API

2.5.1 API là gì?

API là các phương thức, giao thức kết nối với các thư viện và ứng dụng khác. Nó là viết tắt của Application Programming Interface – giao diện lập trình ứng dụng. API cung cấp khả năng cung cấp khả năng truy xuất đến một tập các hàm hay dùng. Và từ đó có thể trao đổi dữ liệu giữa các ứng dụng [38].

2.5.2 Web API là gì?

Web API là một phương thức dùng để cho phép các ứng dụng khác nhau có thể giao tiếp, trao đổi dữ liệu qua lại với nhau. Dữ liệu được Web API trả lại thường ở dạng JSON hoặc XML thông qua giao thức HTTP hoặc HTTPS [38].



Hình 2-22 Web API liên kết các ứng dụng với nhau

2.6 Giới thiệu về Flask

2.6.1 Flask là gì?

Flask là một Web Framework rất nhẹ của Python, dễ dàng giúp người mới bắt đầu học Python có thể tạo ra website nhỏ. Flask cũng dễ mở rộng để xây dựng các ứng dụng web phức tạp. Flask có nền tảng là Werkzeug và Jinja2 và nó đã trở thành một trong những Web Framework phổ biến nhất của Python [39].



Flask là gì?

Hình 2-23 Tìm hiểu về Flask

2.6.2 Tính năng của Flask Framework

- RESTful request dispatch
- Hỗ trợ bảo mật cookie
- Full WSGI compliant
- Khả năng tương thích công cụ dựa trên ứng dụng Google
- Thiết kế gọn nhẹ
- Độ linh hoạt cao
- Cung cấp xử lý HTTP request
- API có độc đáo và mạch lạc
- Dễ dàng triển khai

```
from flask import Flask
app = Flask(__name__)

@app.route('/hello')
def hello_world():
    return 'Hello, World!'

if __name__ == '__main__':
    app.run(host="0.0.0.0", port='8080')
```

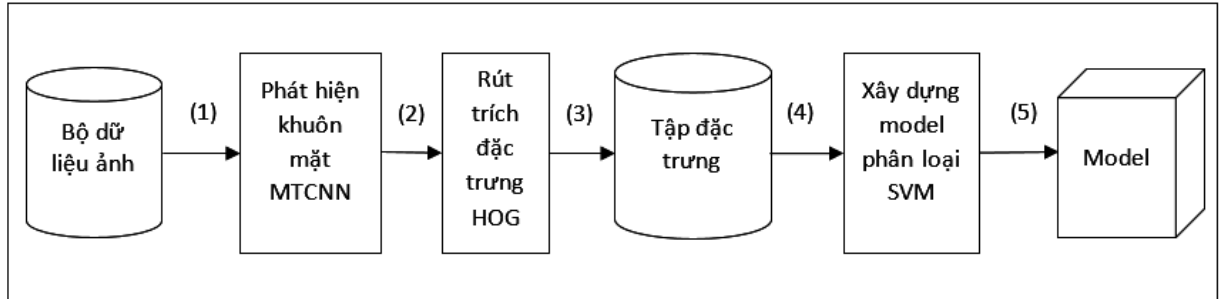
Hình 2-24 Ví dụ về cách triển khai web API của Flask

Chỉ với 9 dòng code đã có thể dựng lên một web API và nó hoàn toàn đáp ứng được yêu cầu của người lập trình, đó là trả về cho client chuỗi 'Hello, World!'.

CHƯƠNG 3 PHƯƠNG PHÁP NHẬN DẠNG SINH VIÊN VÀ TRA CỨU KẾT QUẢ

3.1 Phương pháp nhận dạng

3.1.1 Huấn luyện mô hình

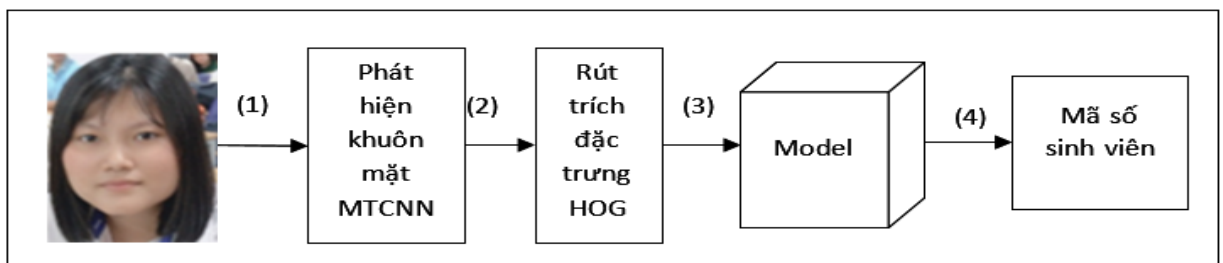


Hình 3-1 Mô hình xây dựng model phân loại

Để xây dựng bộ phân loại gồm có 3 phần chính: Phát hiện khuôn mặt, rút trích đặc trưng, huấn luyện model. Mô hình xây dựng Hình 3-1 gồm các bước sau:

- (1) Phát hiện khuôn mặt bằng mô hình MTCNN;
- (2) Rút trích đặc trưng HOG;
- (3) Lưu lại tập đặc trưng đã rút trích;
- (4) Huấn luyện thuật toán SVM để phân lớp các đặc trưng của dữ liệu ảnh đầu vào;
- (5) Lưu model đã huấn luyện lại.

3.1.2 Các bước nhận dạng



Hình 3-2 Mô hình nhận dạng ảnh khuôn mặt

Để nhận dạng khuôn mặt bao gồm 3 thành phần chính: Phát hiện khuôn mặt, rút trích đặc trưng và nhận dạng khuôn mặt. Mô hình nhận dạng Hình 3-2 gồm các bước sau:

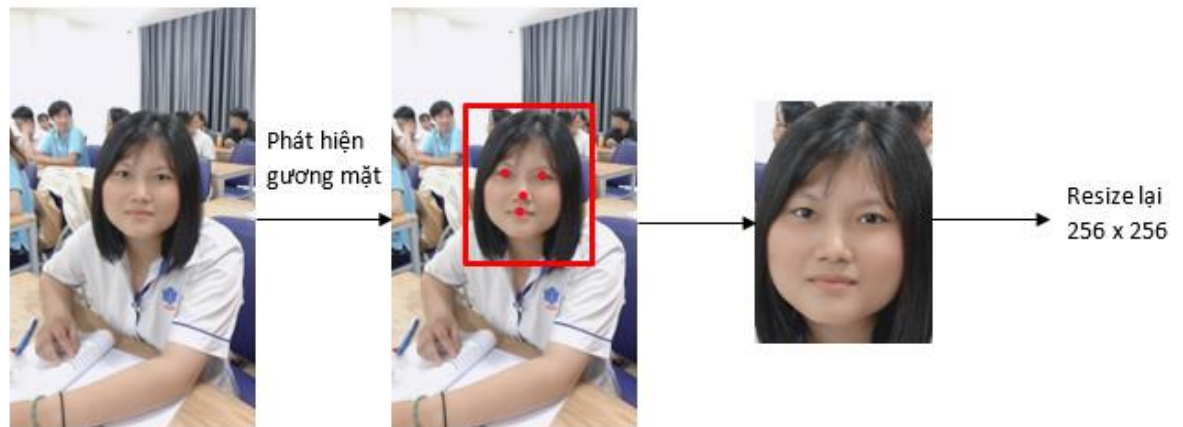
- (1) Phát hiện khuôn mặt bằng mô hình MTCNN;
- (2) Rút trích đặc trưng HOG;
- (3) Đưa qua model phân loại để nhận dạng khuôn mặt;
- (4) Kết quả trả về khi nhận dạng thành công.

3.2 Chi tiết phương pháp nhận dạng

3.2.1 Phát hiện khuôn mặt

Để có thể phát hiện được khuôn mặt thì có rất nhiều phương pháp và được gom thành 2 loại là phương pháp truyền thống và phương pháp học sâu, một số loại phương pháp truyền thống tuy nhanh và ít tốn kém nhưng đem lại hiệu quả không cao bằng các phương pháp học sâu.

MTCNN là một mạng học sâu gồm 3 tầng giúp đưa ra độ chính xác cao mà thời gian phát hiện cũng không quá lâu so với các phương pháp truyền thống. Cụ thể là MTCNN sẽ sử dụng 3 mạng CNN xếp chồng và đồng thời hoạt động khi phát hiện khuôn mặt. Và kết quả đầu ra của MTCNN là vị trí khuôn mặt và các điểm trên mặt như: mắt, mũi, miệng.



Hình 3-3 Phát hiện khuôn mặt

3.2.2 Rút trích đặc trưng HOG

Đầu vào của thuật toán là ảnh xám (gray scale), do đó bước xử lý đầu tiên là chuyển đổi ảnh sang ảnh xám, kích thước ảnh được chuẩn hóa về 256x256. Với mỗi hình ảnh kích thước 256x256, chia thành các block 4x4 chồng nhau, sẽ có 5 block ngang và 5 block dọc, nên sẽ có $5 \times 5 = 25$ blocks. Tiếp theo là tính gradient của ảnh bằng công thức:

$$|G| = \sqrt{F_x^2 + F_y^2} \text{ và } \theta = \tan^{-1}\left(\frac{F_y}{F_x}\right)$$

Với F_x là đạo hàm theo trục x, F_y là đạo hàm theo trục y được tính bởi công thức

$$F_x = F * k_x; F_y = F * k_y$$

Trong đó $k_x = [-1 \ 0 \ 1]$; $k_y = \begin{bmatrix} -1 \\ 0 \\ 1 \end{bmatrix}$

Mỗi block gồm 16 cell. Khi áp dụng biểu đồ 9-bin cho mỗi cell, mỗi block sẽ được đại diện bởi một vector có kích thước 144x1.

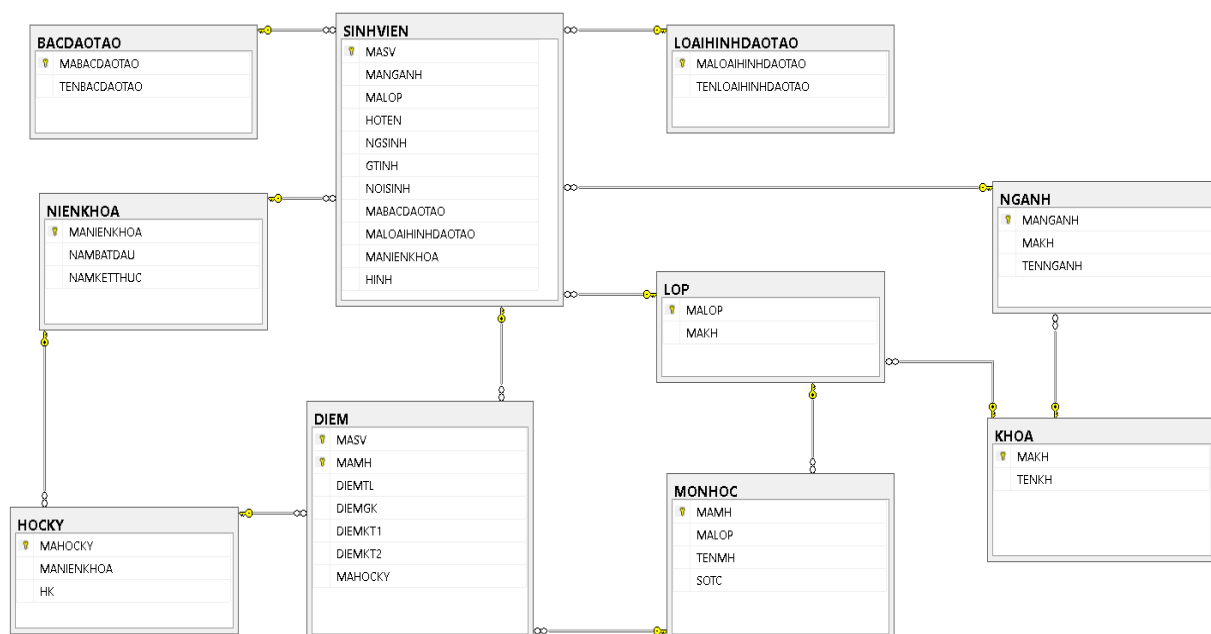
Vì vậy, khi nối tất cả các vector trong một block lại với nhau, thu được vector đặc trưng HOG của ảnh có kích thước $25 \times 144 \times 1 = 3600 \times 1$.

3.2.3 Huấn luyện mô hình phân loại SVM

Sau khi có được tập đặc trưng HOG tiến hành đem tập đặc trưng đi huấn luyện mô hình sử dụng các phương pháp Kfold, Ensemble, GridSearchCV,... để tìm các kernel và tham số cho kết quả tốt nhất đối với mô hình này.

Sau khi sử dụng các phương pháp trên lấy ra được kernel RBF, hệ số $C = 100$ và $\text{Gamma} = 0.01$, tập dữ liệu được chia làm 2 phần, 80% dùng cho huấn luyện và 20% dùng để kiểm thử.

3.3 Thiết kế cơ sở dữ liệu



Hình 3-4 Cơ sở dữ liệu

Bảng 3-1 Bảng BACDAOTAO (bậc đào tạo)

Tên cột	Kiểu dữ liệu	Mô tả	Ví dụ
<u>MABACDAOTAO</u>	VARCHAR(10)	Mã bậc đào tạo	DH, CD,...
TENBACDAOTAO	NVARCHAR(50)	Tên bậc đào tạo	Đại học, cao đẳng,...

Bảng 3-2 Bảng LOAIHINHDAOTAO (loại hình đào tạo)

Tên cột	Kiểu dữ liệu	Mô tả	Ví dụ
<u>MALOAIHINHDAOTAO</u>	VARCHAR(10)	Mã loại hình đào tạo	CQ, VHVL,...
TENLOAIHINHDAOTAO	NVARCHAR(50)	Tên loại hình đào tạo	Chính quy, vừa học vừa làm,...

Bảng 3-3 Bảng KHOA (khoa)

Tên cột	Kiểu dữ liệu	Mô tả	Ví dụ
<u>MAKH</u>	VARCHAR(10)	Mã khoa	CNTT, CK,...
TENKH	NVARCHAR(50)	Tên khoa	Khoa Công nghệ Thông tin,...

Bảng 3-4 Bảng NGANH (ngành)

Tên cột	Kiểu dữ liệu	Mô tả	Ví dụ
<u>MANGANH</u>	VARCHAR(10)	Mã ngành	7480103, 7480104
<u>MAKH</u>	VARCHAR(10)	Mã khoa	CNTT, CK,...
TENNGANH	NVARCHAR(50)	Tên ngành	Hệ thống thông tin,...

Bảng 3-5 Bảng LOP (lớp)

Tên cột	Kiểu dữ liệu	Mô tả	Ví dụ
<u>MALOP</u>	VARCHAR(10)	Mã lớp	09DHTH5, 09DHTH8,...
MAKH	VARCHAR(10)	Mã khoa	CNTT, CK,...

Bảng 3-6 Bảng NIENKHOA (niên khoá)

Tên cột	Kiểu dữ liệu	Mô tả	Ví dụ
<u>MANIENKHOA</u>	INT	Mã niên khoá	08, 09,...
NAMBATDAU	INT	Năm bắt đầu	2018
NAMKETTHUC	INT	Năm kết thúc	2022

Bảng 3-7 Bảng HOCKY (học kỳ)

Tên cột	Kiểu dữ liệu	Mô tả	Ví dụ
<u>MAHOCKY</u>	INT	Mã học kỳ	1, 2, 3,...
<u>MANIENKHOA</u>	INT	Mã niên khoá	08, 09,...
HK	VARCHAR(50)	Học kỳ	Học kỳ 1, Học kỳ 2,...

Bảng 3-8 Bảng MONHOC (môn học)

Tên cột	Kiểu dữ liệu	Mô tả	Ví dụ
<u>MAMH</u>	VARCHAR(10)	Mã môn học	0101003472, 0101006199,...
<u>MALOP</u>	VARCHAR(10)	Mã lớp	09DHTH5, 09DHTH8,...
TENMH	NVARCHAR(50)	Tên môn học	Nhập môn lập trình, Toán rời rạc,...
SOTC	INT	Số tín chỉ	1, 2, 3,...

Bảng 3-9 Bảng SINHVIEN (sinh viên)

Tên cột	Kiểu dữ liệu	Mô tả	Ví dụ
<u>MASV</u>	VARCHAR(10)	Mã sinh viên	2001180476
<u>MANGANH</u>	VARCHAR(10)	Mã ngành	7480103
<u>MALOP</u>	VARCHAR(10)	Mã lớp	09DHTH5
HOTEN	NVARCHAR(50)	Họ tên	Nguyễn Ngọc Hải
NGSINH	DATE	Ngày sinh	06/01/2000
GTINH	NVARCHAR(5)	Giới tính	Nam
NOISINH	NVARCHAR(50)	Nơi sinh	Quảng Nam
MABACDAOTAO	VARCHAR(10)	Mã bậc đào tạo	DH, CD,...

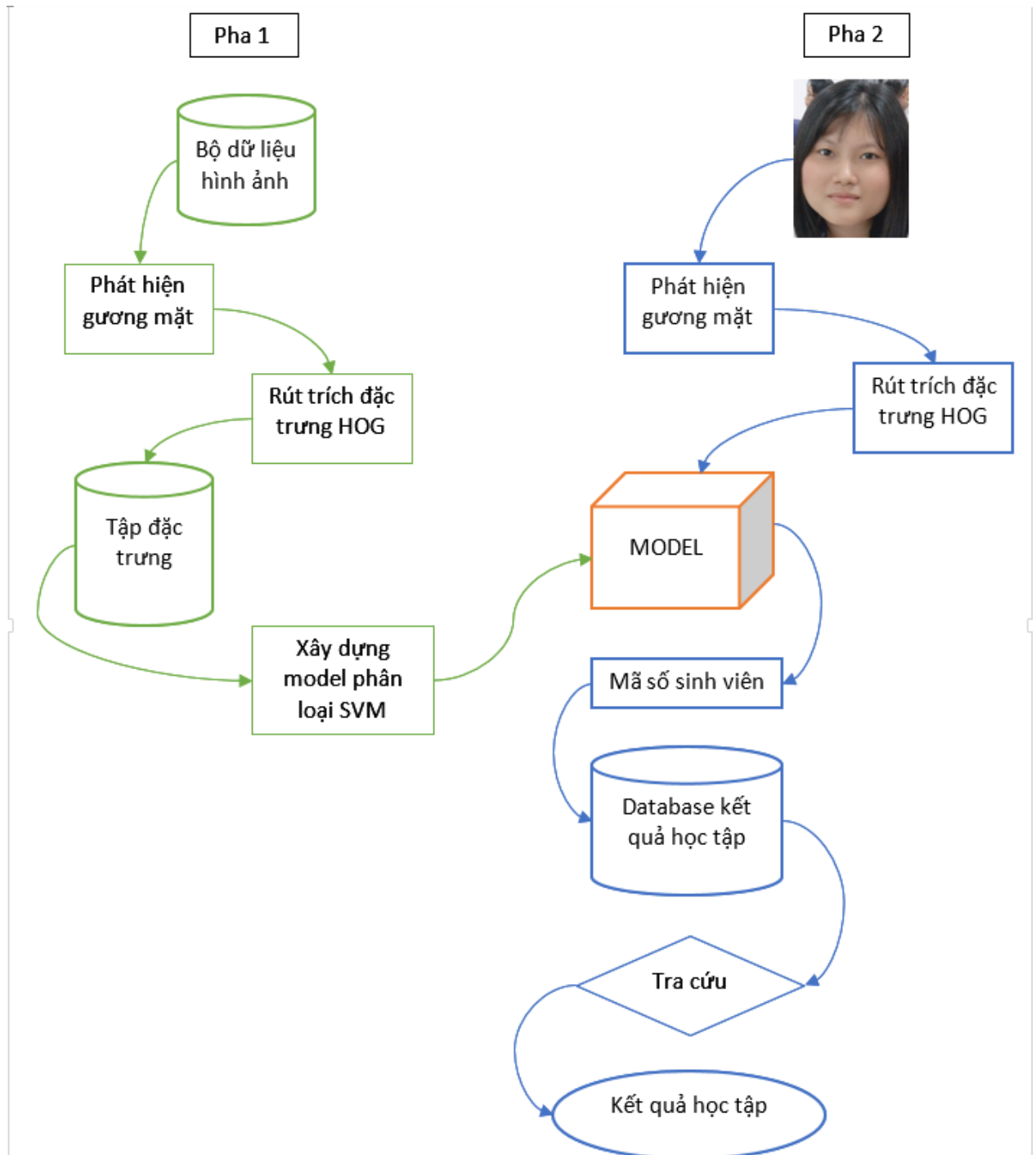
MALOAIIHINHDAOTAO	VARCHAR(10)	Mã loại hình đào tạo	CQ
MANIENKHOA	INT	Mã niên khoá	09
HINH	VARCHAR(50)	Hình	.jpg

Bảng 3-10 Bảng DIEM (điểm)

Tên cột	Kiểu dữ liệu	Mô tả	Ví dụ
<u>MASV</u>	VARCHAR(10)	Mã sinh viên	2001180476
<u>MAMH</u>	VARCHAR(10)	Mã môn học	0101003472
DIEMTL	FLOAT	Điểm tiểu luận	8, 9, 10,...
DIEMGK	FLOAT	Điểm giữa kỳ	10, 9, 8,...
DIEMKT1	FLOAT	Điểm kết thúc 1	8
DIEMKT2	FLOAT	Điểm kết thúc 2	9
<u>MAHK</u>	INT	Mã học kỳ	1

CHƯƠNG 4 THỰC NGHIỆM VÀ ĐÁNH GIÁ

4.1 Mô hình thực nghiệm



Hình 4-1 Mô hình hệ thống tra cứu kết quả học tập sinh viên qua hình ảnh

Mô hình hệ thống tra cứu kết quả học tập sinh viên qua hình ảnh gồm có 2 pha chính như sau:

❖ **Pha 1 dùng để huấn luyện mô hình phân loại khuôn mặt gồm các bước sau:**

Bước 1: Phát hiện khuôn mặt cho tập dữ liệu ảnh

Bước 2: Rút trích đặc trưng HOG

Bước 3: Xây dựng model phân loại SVM để phân loại khuôn mặt của ảnh đầu vào

❖ **Pha 2 dùng để nhận dạng và tra cứu kết quả học tập gồm các bước sau:**

Bước 1: Phát hiện khuôn mặt cho ảnh đầu vào

Bước 2: Rút trích đặc trưng HOG

Bước 3: Sử dụng model phân loại SVM đã huấn luyện để nhận dạng đặc trưng của ảnh đầu vào

Bước 4: Lấy kết quả nhận dạng được tra cứu với cơ sở dữ liệu để đưa ra kết quả học tập

4.2 Dữ liệu thực nghiệm

Tập ảnh YaleFace [28] (kích thước 6,4MB) chứa 165 hình ảnh thang độ xám ở định dạng GIF của 15 cá nhân. Có 11 hình ảnh cho mỗi chủ thể, một hình ảnh cho mỗi biểu hiện hoặc cấu hình khuôn mặt khác nhau: ánh sáng trung tâm, có đeo kính, hạnh phúc, ánh sáng trái, không đeo kính, bình thường, ánh sáng phải, buồn, buồn ngủ, ngạc nhiên và nháy mắt.



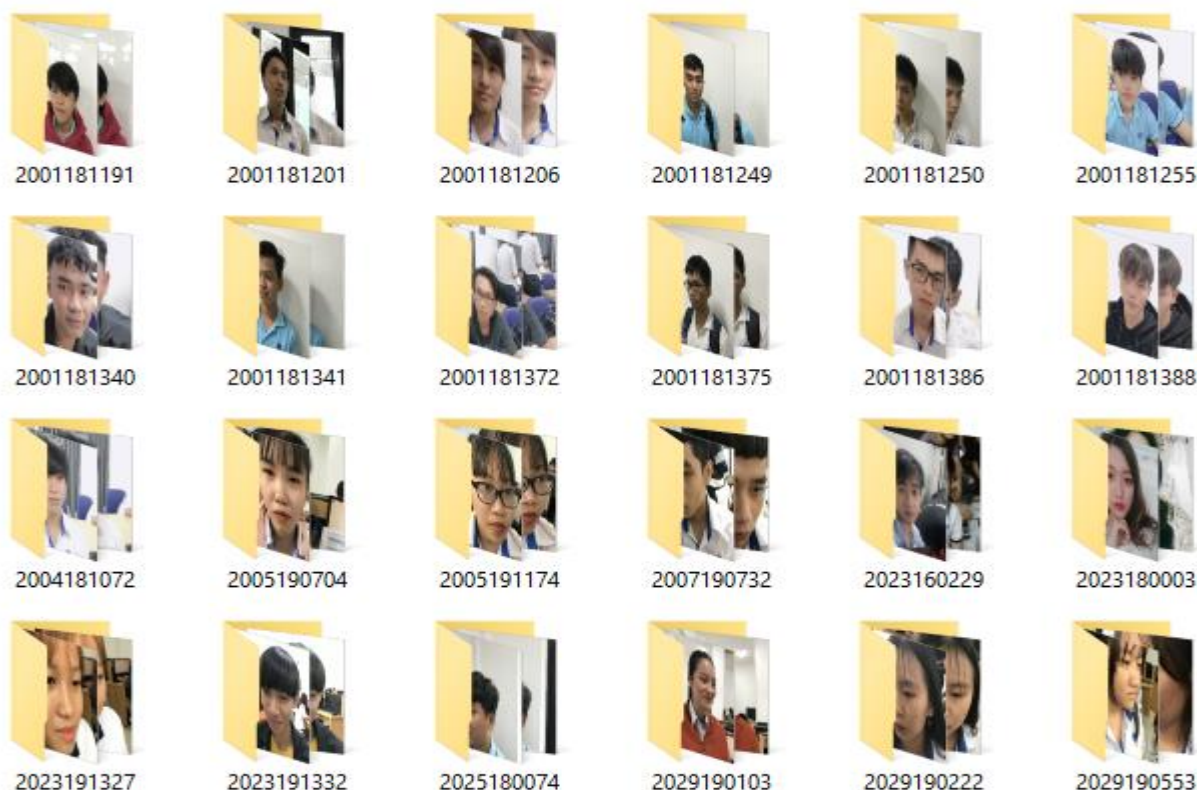
Hình 4-2 Tập ảnh YaleFace

Cơ sở dữ liệu khuôn mặt YaleFace B chứa 16128 hình ảnh của 28 người dưới 9 tư thế và 64 điều kiện chiếu sáng. Định dạng dữ liệu của cơ sở dữ liệu này giống như bộ ảnh YaleFace. Tất cả dữ liệu hình ảnh thử nghiệm được sử dụng trong các thí nghiệm được căn chỉnh thủ công, cắt xén và sau đó có kích thước lại thành hình ảnh 168x192, tham khảo bài báo PAMI2005.



Hình 4-3 Tập ảnh YaleFaceB

Tập ảnh sinh viên trường Đại học Công nghiệp Thực phẩm TP. Hồ Chí Minh khoa Công nghệ Thông tin gồm 80 bạn mỗi bạn từ 10-20 ảnh với các biểu cảm khác nhau: vui, buồn, hạnh phúc, ngạc nhiên,



Hình 4-4 Tập ảnh sinh viên khoa Công nghệ thông tin

4.3 Phương pháp thực nghiệm

4.3.1 Rút trích đặc trưng HOG

Với đầu vào là một bức ảnh kích thước 256x256, áp dụng thuật toán tính HOG với kích thước cells là 32x32 chúng ta sẽ thu được một lưới ô vuông với kích thước là $256/32 = 8$ ô theo chiều cao và $256/32 = 8$ ô theo chiều rộng. Tiếp tục khởi tạo các block kích thước 4x4 ô với stride là 128x128 pixels ta sẽ trải qua 5 bước theo chiều cao và 5 bước theo chiều rộng. Trên mỗi block ta có 4 vector histogram tương ứng với mỗi ô, mỗi vector gồm 9 chiều tương ứng với 9 bins. Như vậy vector HOG tổng hợp của bức ảnh sẽ có kích thước là $5 \times 5 \times 4 \times 4 \times 9 = 3600$ chiều.

4.3.2 Rút trích đặc trưng Land-mark

Sử dụng thư viện dlib và model có sẵn để tìm 68 điểm trên khuôn mặt. Sau khi tìm được tọa độ (x, y) của từng điểm ta chuyển nó về dạng numpy. Giờ ta có được 1 mảng 2 chiều gồm các tọa độ (x, y). Duỗi nó thành mảng 1 ta được 1 vector có **136 chiều**.

4.3.3 Xây dựng mô hình phân loại SVM

Sử dụng kernel RBF, hệ số $C = 100$ và $\text{Gamma} = 0.01$, tập dữ liệu được chia làm 2 phần, 80% dùng cho huấn luyện và 20% dùng để kiểm thử. Sử dụng phương pháp Kfold kiểm thử bộ dữ liệu học với 10 fold độ chính xác trung bình đạt 88,19%.

Bảng 4-1 Bảng hiệu suất độ chính xác của tập dữ liệu học khi kiểm thử với Kfold trên bộ dữ liệu sinh viên

Vòng lặp	Độ chính xác
1	93,02%
2	91,86%
3	89,53%
4	86,04%
5	87,2%
6	87,2%
7	87,05%
8	88,23%
9	80%
10	91,76%

4.4 Kết quả thực nghiệm

Thực nghiệm được thực thi trên máy PC CPU Intel Core i5 - 1035G1 CPU @ 1.00GHz - @1.19 GHz, 4.0GB RAM, hệ điều hành Windows 10 Home 64 bit và google colab sử dụng GPU, sử dụng ngôn ngữ lập trình python, C# và .NET Framework 4.6. Kết quả thực nghiệm được đánh giá trên bộ dữ liệu YaleFace, YaleFaceB và sinh viên HUFI như mô tả ở trên.

Bảng 4-2 Bảng hiệu suất nhận dạng ảnh khuôn mặt của phương pháp được đề xuất trên các bộ dữ liệu thực nghiệm

Tập ảnh	Tổng số ảnh	Số ảnh huấn luyện	Số ảnh kiểm thử	Độ chính xác trung bình (precision)	Độ phủ trung bình(recall)	Độ trung hoà trung bình (f1-score)
YaleFace	165	132	33	100%	100%	100%
YaleFaceB	16128	13618	2510	98,44%	98,46%	98,44%
Sinh viên HUFI	1040	832	208	86,2%	82,7%	81,8%

4.4.1 Tốc độ xử lý rút trích đặc trưng

Bảng 4-3 Bảng so sánh tốc độ xử lý khi rút trích đặc trưng trên hai phương pháp thực nghiệm

Phương pháp	Tốc độ
Land-mark	0.0024s/ảnh
HOG	0.0089s/ảnh

4.4.2 Độ chính xác

Bảng 4-4 Bảng so sánh độ chính xác nhận dạng trung bình trên hai phương pháp thực nghiệm

Phương pháp / Tập ảnh	YaleFace	YaleFaceB
Land-mark + SVM	70,07%	69,57%
HOG + SVM	100%	98,44%

4.4.3 So sánh với các phương pháp khác

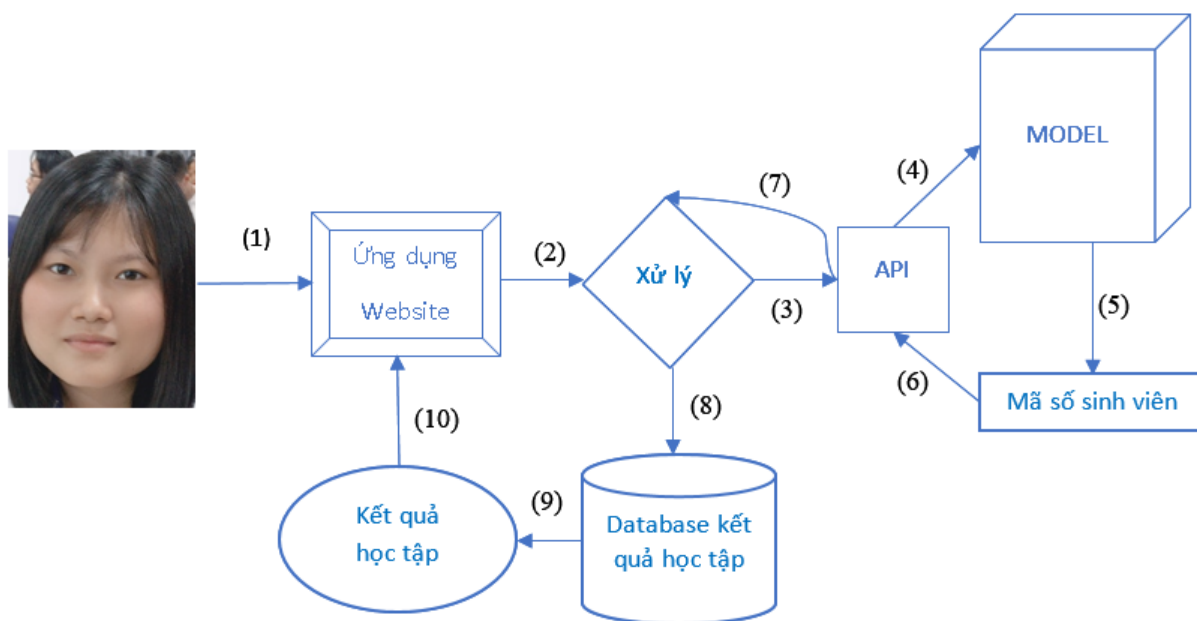
Bảng 4-5 So sánh các phương pháp trên bộ dữ liệu YaleFace

Phương pháp	Độ chính xác nhận dạng trung bình
Trần Hồng Việt và cộng sự, 2021 [31]	95%, 95,1%, 97%
P. Quintiliano và cộng sự, 2001 [33]	72%
Phạm Văn Trường, Trần Thị Thảo [34]	95%
AYDIN KAYACIK, 2008 [32]	78%
Đề xuất của nhóm	100%

Bảng 4-6 So sánh các phương pháp trên bộ dữ liệu YaleFaceB

Phương pháp	Độ chính xác nhận dạng trung bình
Nguyễn Thị Nông, 2012 [35]	98,26% ; 98,16% ; 99,65% ; 100%
T. Meenpal, 2018 [36]	91.7%
Đề xuất của nhóm	98,44%

4.5 Kiến trúc ứng dụng



Hình 4-5 Kiến trúc ứng dụng

Kiến trúc ứng dụng được thực thi theo các bước sau:

- (1) Lấy dữ liệu ảnh (load ảnh lên ứng dụng hoặc website)
- (2) Xử lý ảnh thành chuỗi và mã hoá chuỗi
- (3) Truyền chuỗi lên API đã dựng
- (4) Web API sẽ gọi hàm nhận dạng khuôn mặt (đưa qua model phân loại đã huấn luyện)
- (5) Trả về kết quả là mã số sinh viên khi nhận dạng thành công
- (6) Trả lại dữ liệu là kết quả đã nhận diện lên web API
- (7) Đưa qua lớp xử lý để tiến hành tra cứu kết quả học tập
- (8) Lấy mã số sinh viên truy vấn vào database kết quả học tập của sinh viên
- (9) Lấy được kết quả học tập
- (10) Trả về giao diện kết quả học tập trên ứng dụng hoặc website

4.6 Một số xử lý chính

4.6.1 Phát hiện khuôn mặt

```
def DetectFace(img):  
    detect = MTCNN()  
    results = detect.detect_faces(img)  
    if len(results) > 0:  
        result = results[0]  
        x1, y1, width, height = result['box']  
        x1, y1 = abs(x1), abs(y1)  
        x2 = x1 + width  
        y2 = y1 + height  
        face = img[y1:y2, x1:x2]  
        face = cv2.cvtColor(face, cv2.COLOR_BGR2GRAY)  
        face = cv2.resize(face, (256, 256))  
    return face
```

Xử lý phát hiện gương mặt gồm các bước:

- (1) Khởi tạo bộ phát hiện gương mặt MTCNN từ thư viện mtcnn
- (2) Phát hiện khuôn mặt
- (3) Lấy tọa độ x, y và chiều cao, chiều rộng của khuôn mặt đầu tiên và cắt khuôn mặt ra khỏi ảnh gốc
- (4) Chuẩn hóa ảnh khuôn mặt về ảnh xám và chỉnh kích thước ảnh về 256x256

4.6.2 Rút trích đặc trưng

```
def FeatureExtraction(face):  
    feature = []  
    hogImage = hog(face, orientations=9, pixels_per_cell=(32, 32),  
                   cells_per_block=(4, 4), transform_sqrt=True, block_norm="L2",  
                   visualize=False)  
    feature.append(hogImage)  
    return feature
```

Rút trích đặc trưng khuôn mặt gồm các bước sau:

- (1) Khai báo mảng để chứa các đặc trưng
- (2) Rút trích đặc trưng HOG bằng thư viện scikit-image

- orietations (số bin)
- pixels_per_cell (kích thước của một cell)
- cells_per_block (số cell trong một block)
- block_norm: Phương pháp chuẩn hóa block

(3) Thêm đặc trưng đã được rút trích vào mảng

4.6.3 Nhận dạng khuôn mặt

```
f = open("/content/drive/MyDrive/Python/KLTN/model.pkl", 'rb')
model = pickle.load(f)
f.close()
```

1

```
def RecognitionFace(feature):
    result = model.predict(feature)
    return result
```

2

Nhận dạng khuôn mặt gồm các bước sau:

- (1) Load model phân loại khuôn mặt sinh viên (SVM) đã huấn luyện
- (2) Nhận dạng và trả về kết quả

4.6.4 Xây dựng web API lấy dữ liệu hình ảnh và nhận dạng

```
app = Flask(__name__)
CORS(app)
```

1

```
@app.route('/detect', methods=['POST'])
def process():
    image_b64 = request.form.get('image')
    image = np.frombuffer(base64.b64decode(image_b64), dtype=np.uint8)
    image = cv2.imdecode(image, cv2.IMREAD_COLOR)
    face = DetectFace(image)
    feature = FeatureExtraction(face)
    predict = RecognitionFace(feature)
    print(predict)
    return str(predict[0])
```

2

```
if __name__ == "__main__":
    app.run(host="0.0.0.0", port='8080')
```

3

Xây dựng API gồm các bước sau:

- (1) Khởi tạo Flask server bằng thư viện flask
- (2) Xây dựng hàm xử lý khi nhận dữ liệu
 - Lấy dữ liệu request (chuỗi base 64 đã mã hoá)
 - Giải mã chuỗi
 - Chuyển chuỗi về hình ảnh
 - Phát hiện khuôn mặt
 - Rút trích đặc trưng
 - Nhận dạng khuôn mặt
- (3) Cấu hình host và port để khởi chạy web API

4.6.5 Truyền dữ liệu từ ứng dụng và website lên web API

```
public static string ConvertImageToBase64String(Image image) {  
    try  
    {  
        using (MemoryStream ms = new MemoryStream()) {  
            image.Save(ms, System.Drawing.Imaging.ImageFormat.Bmp);  
            return Convert.ToBase64String(ms.ToArray());  
        }  
    }  
    catch { return null; }  
}
```

1

```
private String EscapeData(String B64) {  
    int B64_length = B64.Length;  
    if (B64_length <= 32000) {  
        return Uri.EscapeDataString(B64);  
    }  
  
    int idx = 0;  
    StringBuilder builder = new StringBuilder();  
    String substr = B64.Substring(idx, 32000);  
    while (idx < B64_length) {  
        builder.Append(Uri.EscapeDataString(substr));  
        idx += 32000;  
  
        if (idx < B64_length){  
            substr = B64.Substring(idx, Math.Min(32000, B64_length - idx));  
        }  
    }  
    return builder.ToString();  
}
```

2

```

private String sendPOST(String url, String B64) {
    try
    {
        var request = (HttpWebRequest)WebRequest.Create(url);
        request.Timeout = 5000;
        var postData = "image=" + EscapeData(B64);
        var data = Encoding.ASCII.GetBytes(postData);
        request.Method = "POST";
        request.ContentType = "application/x-www-form-urlencoded";
        request.ContentLength = data.Length;

        using (var stream = request.GetRequestStream()) {
            stream.Write(data, 0, data.Length);
        }

        var response = (HttpWebResponse)request.GetResponse();

        var responseString = new StreamReader(response.GetResponseStream()).ReadToEnd();

        return responseString;
    }
    catch (Exception ex)
    {
        return "Exception" + ex.ToString();
    }
}

```

```

public String Detect(Image image) {
    String B64 = ConvertImageToBase64String(image);

    String server_ip = "192.168.1.3";
    String server_path = "http://" + server_ip + ":8080/detect";
    String retStr = sendPOST(server_path, B64);
    return retStr;
}

```

Truyền dữ liệu lên web API gồm các bước sau:

- (1) Hàm chuyển hình ảnh sang chuỗi base 64
- (2) Hàm mã hoá chuỗi base 64
- (3) Hàm gọi web API và truyền dữ liệu trả về kết quả
- (4) Hàm nhận dạng gọi lại các hàm trên và trả về kết quả
 - Server_ip là địa chỉ ip của web API
 - Server_path là đường dẫn kết hợp địa chỉ ip và port cùng với tên hàm cần gọi web API

4.7 Giao diện chương trình

Giao diện của chương trình xử lý hình ảnh của sinh viên khoa CNTT, trả về kết quả là kết quả học tập của sinh viên với hình ảnh tìm kiếm.

4.7.1 Giao diện ứng dụng

❖ Giao diện chọn ảnh sinh viên

- Số 1: button Load hình cần nhận diện.
- Số 2: button Nhận diện



Hình 4-6 Giao diện nhận diện hình ảnh.


❖ Giao diện kết quả nhận diện

- Số 1: Thông tin sinh viên đã nhận dạng được
- Số 2: button Xem điểm của sinh viên đó

Phần mềm tra cứu điểm sinh viên


Nhận diện khuôn mặt

Nhận diện sinh viên



Load hình Nhận diện

Thông tin sinh viên



Lớp học: 09DHTH8
 Khóa học: 2018
 Bắc đào tạo: Đại học
 Loại hình đào tạo: Chính quy
 Ngành: Khoa học phân tích dữ liệu
 MSSV: 2001180002
 Giới tính: Nam
 Họ tên: Trương Hồng Phi
 Ngày sinh: 06-9-2000

Xem điểm


Hình 4-7 Giao diện kết quả nhận diện hình ảnh.

❖ **Giao diện xem điểm của sinh viên**

- Số 1: Thông tin cá nhân
- Số 2: Thông tin học vụ
- Số 3: Chọn học kỳ muốn xem điểm
- Số 4: Bảng điểm theo học kỳ đã chọn

Phần mềm tra cứu điểm sinh viên

Nhận diện khuôn mặt



MSSV: 2001180002
 Họ tên: Trương Hồng Phi
 Giới tính: Nam
 Ngày sinh: 06-9-2000
 Nơi sinh: Tỉnh Bến Tre

Thông tin

Lớp học: 09DHTH8
 Khóa học: 2018
 Bắc đào tạo: Đại học
 Loại hình đào tạo: Chính quy
 Ngành: Khoa học phân tích dữ liệu

Kết quả học tập

HK1(2018-2019)

	Mã môn học	Tên môn học	Số tín chỉ	Lớp dự kiến	TL/BTL	Giữa kỳ	KT1	KT2	Đ. kết
▶	0101000094	Anh văn A1	3	09DHTH8			6.3		6.3
	0101003472	Nhập môn lập...	3	09DHTH8	9		6		7.5
	0101003473	Thực hành n...	2	09DHTH8			10		10
	0101003491	Những nguyê...	2	09DHTH8	5		6		5.5
	0101006144	Toán cao cấp...	3	09DHTH8	8		6.6		7.3
	0101007556	Anh văn A2	3	09DHTH8	6.6		8.4		7.5
	0101007557	Kỹ năng ứng ...	3	09DHTH8	8		9		8.5

Hình 4-8 Giao diện xem điểm sinh viên.

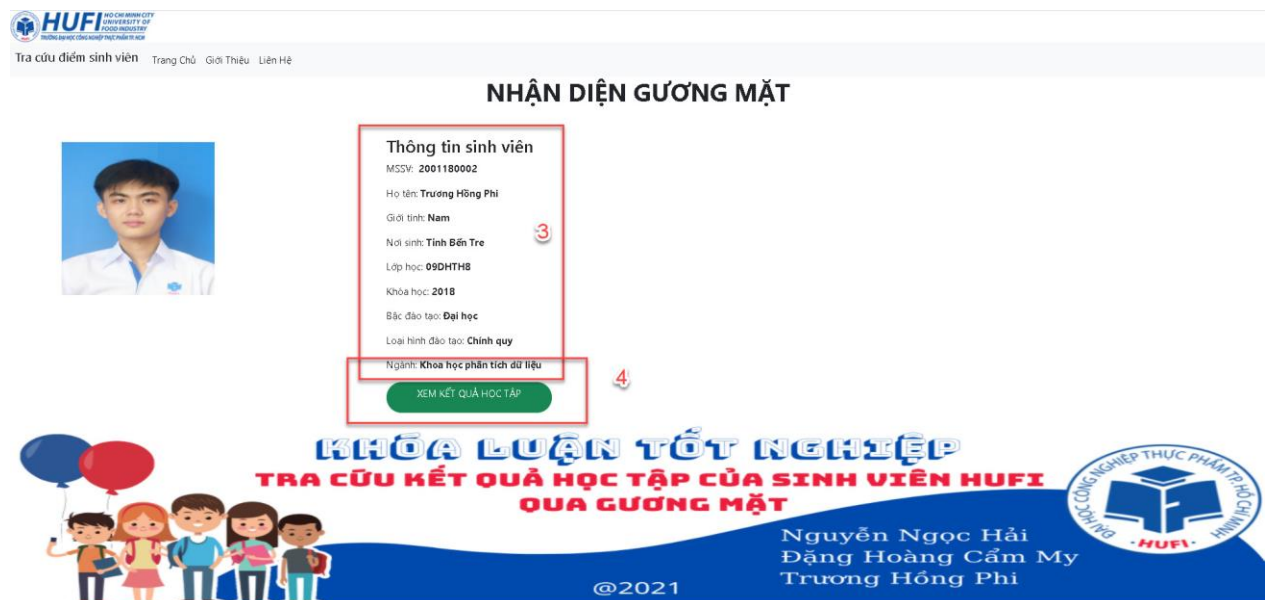
4.7.2 Giao diện website

- Số 1: Chọn hình sinh viên
- Số 2 : Nhấn vào nút nhận diện



Hình 4-9 Giao diện Website trang nhận diện gương mặt

- Số 3. Thông tin sinh viên được hiển thị
- Số 4. Nhấn vào nút xem kết quả học tập



Hình 4-10 Giao diện Website sau khi nhận diện được gương mặt

Kết Quả Học Tập

STT	Mã môn học	Tên môn học/học phần	Lớp dự kiến	Số tín chỉ	TL/BTL	Giữa kỳ	Kết thúc		Điểm tổng kết	Thang điểm 4	Điểm chữ	Xếp loại
							1	2				
HK1(2018-2019)												
1	0101000094	Anh văn A1	09CHTHB	3			6.3		6.3	2	C	Trung bình
2	0101003472	Nhập môn lập trình	09CHTHB	3	9		6		7.5	3	B	Khá
3	0101003473	Thực hành nhập môn lập trình	09CHTHB	2			10		10	4	A	Giỏi
4	0101003491	Những nguyên lý cơ bản của chủ nghĩa Mác-Lênin 1	09CHTHB	2	5		6		5.5	2	D+	Trung bình
5	0101006144	Toán cao cấp A1	09CHTHB	3	8		6.6		7.3	3	B	Khá
6	0101007556	Anh văn A2	09CHTHB	3	6.6		8.4		7.5	3	B	Khá
7	0101007557	Kỹ năng ứng dụng Công nghệ Thông tin	09CHTHB	3	8		9		8.5	4	A	Giỏi
HK2(2018-2019)												
1	0101003493	Những nguyên lý cơ bản của chủ nghĩa Mác-Lênin	09CHTHB	3	7		7		7	3	B	Khá
2	0101003671	Phương luật đại cương	09CHTHB	2	7		7.5		7.25	3	B	Khá
3	0101005281	Thực hành lập trình hướng đối tượng	09CHTHB	1			8.8		8.8	4	A	Giỏi
4	0101006199	Toán rời rạc	09CHTHB	3	9		4		6.5	2.5	C+	Trung bình

Hình 4-11 Giao diện Website xem kết quả học tập của sinh viên

KẾT LUẬN

Đề tài sử dụng các thuật toán phổ biến nhằm nhận diện khuôn mặt dựa trên các hình ảnh khuôn mặt sinh viên từ đó cho ra được kết quả học tập. Theo thực nghiệm cho thấy quá trình trích xuất bằng HOG mất khá nhiều thời gian, nhưng lại đem về hiệu quả chính xác cao hơn nhiều so với rút trích bằng Land-mark. Khi sử dụng Land-mark chỉ lấy được 68 điểm trên khuôn mặt nhưng nếu như khuôn mặt bị khuyết thì không thể tìm thấy được 68 điểm đó. Hướng phát triển tiếp theo là giảm thời gian phát hiện và rút trích đặc trưng của khuôn mặt nhưng vẫn không làm giảm đi độ chính xác.

TÀI LIỆU THAM KHẢO

- [1] <https://cyberlotus.com/cac-loai-sinh-trac-hoc-pho-bien>
- [2] <https://www.semanticscholar.org/paper/Recognition-Using-HOG-Features>
- [3] <https://biquyetxaynha.com/so-luong-hoc-sinh-sinh-vien-ca-nuoc-nam-2021>
- [4] <https://mca-journal.org/index.php/mca/article/view/28/12>
- [5] <http://112.137.129.28/handle/123456789/959>
- [6] <https://www.researchgate.net/profile/Nguyen-Khang-Pham/>
- [7] <https://viso.ai/deep-learning/face-detection-overview/>
- [8] <https://www.mygreatlearning.com/>
- [9] <https://taqadam.io/open-source-datasets/>
- [10] https://docs.opencv.org/3.4/db/d28/tutorial_cascade_classifier.html
- [11] <https://ietresearch.onlinelibrary.wiley.com/MTCNN>
- [12] <http://citeseerx.ist.psu.edu/viewdoc/SVM>
- [13] <https://cafedev.vn/sieu-tham-so-svm>
- [14] <https://www.learnopencv.com/histogram-of-oriented-gradients/>
- [15] <http://mccormickml.com/2013/05/09/hog-person-detector-tutorial/>
- [16] <https://blog.vietnamlab.vn/xac-dinh-facial-landmark>
- [17] <https://viblo.asia/p/nhan-dien-khuon-mat-cua-violas-john>
- [18] <https://machinelearningmastery.com/k-fold-cross-validation/>
- [19] <https://hanoi.fpt.edu.vn/sinh-vien-fpt-diem-danh-bang-khuon-mat>
- [20] <https://vietnamdigitalsignage.com/nhan-dien-khuon-mat-kaoato>
- [21] <https://www.facebook.com/172550369587828/posts/1250231841819670/>
- [22] <https://vnexpress.net/nhan-dien-khuon-mat-viettel-solutions>
- [23] <https://vtct.wordpress.com/2013/06/19/mot-vai-khai-niem-co-ban-ve-anh-so>
- [24] <https://lamchacancadoitay.vn/anh-so-va-cac-khong-gian-mau-trong-xu-ly-anh/>

- [25] https://kpzhang93.github.io/MTCNN_face_detection_alignment/paper/spl.pdf
- [26] <https://iopscience.iop.org/article/10.1088/1742-6596/1518/1/012066/pdf>
- [28] [Yale Face Database | vision.ucsd.edu](http://vision.ucsd.edu)
- [29] <https://vi.wikipedia.org/wiki/SVM>
- [30] <http://www1.vnua.edu.vn/tapchi/Upload/>
- [31] <https://khcn.hauu.edu.vn/media/30/uffile-upload-no-title30656.pdf>
- [32] <https://www.diva-portal.org/smash/get/diva2:239370/FULLTEXT01.pdf>
- [33] <http://ijofcs.org/quintiliano-papers-practical-proc-fr.PDF>
- [34] <https://mca-journal.org/index.php/mca/article/download/28/12>
- [35] <http://thuvien.vku.udn.vn/bitstream/123456789/21/1/full.pdf>
- [36] <https://www.sciencepubco.com/index.php/ijet/article/download/11826/4606>
- [37] https://en.wikipedia.org/wiki/Haar_wavelet
- [38] <https://topdev.vn/blog/api-la-gi/>
- [39] <https://niithanoi.edu.vn/flask-la-gi.html>