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*

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 đã được các Thầy Cô trường Đại học Công nghiệp Thực phẩm Thành phố Hồ Chí Minh đã tạo mọi điều kiện thuận lợi, 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 - Trưởng chuyên nghành Hệ thống thông tin là người thầy mà tôi muốn bày tỏ lòng biết ơn sâu sắc nhất.

Xin chân thành cảm ơn Ban giám hiệu, PGS.TS Nguyễn Xuân Hoàn Hiệu trưởng nhà trường và 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 iii](#_Toc88422172)

[LỜI CẢM ƠN iv](#_Toc88422173)

[MỤC LỤC v](#_Toc88422174)

[DANH MỤC HÌNH ẢNH viii](#_Toc88422175)

[DANH MỤC BẢNG BIỂU x](#_Toc88422176)

[MỞ ĐẦU xi](#_Toc88422177)

[CHƯƠNG 1 TỔNG QUAN ĐỂ TÀI 13](#_Toc88422178)

[1.1 Hiện trạng 13](#_Toc88422179)

[1.2 Mục tiêu đề tài 13](#_Toc88422180)

[1.3 Nội dung nghiên cứu 13](#_Toc88422181)

[1.4 Phạm vi đề tài 14](#_Toc88422182)

[1.5 Phương pháp nghiên cứu 14](#_Toc88422183)

[1.6 Công nghệ sử dụng 14](#_Toc88422184)

[CHƯƠNG 2 CƠ SỞ LÝ THUYẾT 15](#_Toc88422185)

[2.1 Giới thiệu về ảnh số 15](#_Toc88422186)

[2.1.1 Khái niệm 15](#_Toc88422187)

[2.1.2 Kích thước ảnh 17](#_Toc88422188)

[2.1.3 Ảnh xám: 18](#_Toc88422189)

[2.1.4 Ảnh màu: 18](#_Toc88422190)

[2.2 Phát hiện khuôn mặt (Face Detection) 19](#_Toc88422191)

[2.2.1 Khái niệm 19](#_Toc88422192)

[2.2.2 Phương pháp MTCNN 19](#_Toc88422193)

[2.2.3 Phương pháp Haar-like - Adaboost 21](#_Toc88422194)

[2.3 Trích xuất đặc trưng (Feature Extraction) 24](#_Toc88422195)

[2.3.1 Khái niệm 24](#_Toc88422196)

[2.3.2 Phương pháp 25](#_Toc88422197)

[2.4 Nhận dạng khuôn mặt (Face Recognition) 31](#_Toc88422198)

[2.4.1 Khái niệm 31](#_Toc88422199)

[2.4.2 SVM (Support Vector Machine) 32](#_Toc88422200)

[2.4.3 KNN (K Nearest Neighbors) 33](#_Toc88422201)

[2.4.4 Cây quyết định (Decision Tree) 34](#_Toc88422202)

[2.5 Thiết kế cơ sở dữ liệu 37](#_Toc88422203)

[CHƯƠNG 3 PHƯƠNG PHÁP NHẬN DẠNG SINH VIÊN VÀ TRA CỨU KẾT QUẢ 41](#_Toc88422204)

[3.1 Mô hình tổng quan 41](#_Toc88422205)

[3.2 Các bước nhận dạng 42](#_Toc88422206)

[3.2.1 Đầu vào, huấn luyện 42](#_Toc88422207)

[3.2.2 Thực thi 42](#_Toc88422208)

[3.3 Chi tiết phương pháp nhận dạng 42](#_Toc88422209)

[3.3.1 Tiền xử lý 42](#_Toc88422210)

[3.3.2 Cách thức hoạt động của SVM 46](#_Toc88422211)

[CHƯƠNG 4 THỰC NGHIỆM VÀ ĐÁNH GIÁ 47](#_Toc88422212)

[4.1 Giao diện chương trình 47](#_Toc88422213)

[4.2 Kết quả thực nghiệm 49](#_Toc88422214)

[4.2.1 Dữ liệu thực nghiệm 49](#_Toc88422215)

[4.2.2 Phương pháp thực nghiệm 49](#_Toc88422216)

[4.3 So sánh phương pháp 50](#_Toc88422217)

[4.3.1 Độ chính xác 50](#_Toc88422218)

[4.3.2 Tốc độ hai phương pháp 53](#_Toc88422219)

[4.3.3 So sánh HOG+SVM với PCA 53](#_Toc88422220)

[4.4 Cài đặt trên bộ ảnh sinh viên 54](#_Toc88422221)

[4.5 Kết luận 54](#_Toc88422222)

[TÀI LIỆU THAM KHẢO 55](#_Toc88422223)

# DANH MỤC HÌNH ẢNH

[Hình 2‑1 Ảnh raster 17](#_Toc89120978)

[Hình 2‑2 Ảnh vector 18](#_Toc89120979)

[Hình 2‑3 Kích thước ảnh 19](#_Toc89120980)

[Hình 2‑4 Ảnh xám 19](#_Toc89120981)

[Hình 2‑5 Ảnh màu 20](#_Toc89120982)

[Hình 2‑6 Cấu trúc của MTCNN 21](#_Toc89120983)

[Hình 2‑7 Đặc trưng haar - like 23](#_Toc89120984)

[Hình 2‑8 Đặc trưng Haar-like cơ bản để xác định khuôn mặt 23](#_Toc89120985)

[Hình 2‑9 Đặc trưng Haar – like mở rộng 24](#_Toc89120986)

[Hình 2‑10 68 điểm landmark 32](#_Toc89120987)

[Hình 2‑11 Phân loại hình thoi 35](#_Toc89120988)

[Hình 2‑12 Cơ sở dữ liệu 38](#_Toc89120989)

[Hình 4‑1 Tập ảnh YaleFaceB 48](#_Toc89120990)

[Hình 4‑2 Kết quả train đặc trưng HOG bằng SVM trên tập ảnh YaleFacesB 49](#_Toc89120991)

[Hình 4‑3 Kết quả train đặc trưng Facial landmark bằng SVM trên tập ảnh YaleFacesB 50](#_Toc89120992)

[Hình 4‑4 Biểu đồ thể hiện độ chính xác của HOG và Landmark 50](#_Toc89120993)

[Hình 4‑5 Thời gian xử lý rút trích đặc trưng HOG 51](#_Toc89120994)

[Hình 4‑6 Thời gian xử lý rút trích đặc trưng Facial landmark 51](#_Toc89120995)

[Hình 4‑7 Biểu đồ thể hiện thời gian xử lý 51](#_Toc89120996)

[Hình 4‑8 Hình ảnh so sánh giữa HOG+SVM và PCA 52](#_Toc89120997)

[Hình 4‑9 Tập ảnh sinh viên khoa Công nghệ thông tin 53](#_Toc89120998)

[Hình 4‑10 Độ chính xác khi train và validate model với Kfold = 10 53](#_Toc89120999)

[Hình 4‑11 Độ chính xác của model khi kiểm chứng với tập test 54](#_Toc89121000)

[Hình 4‑12 Giao diện nhận diện hình ảnh. 55](#_Toc89121001)

[Hình 4‑13 Giao diện kết quả nhận diện hình ảnh. 56](#_Toc89121002)

[Hình 4‑14 Giao diện xem điểm sinh viên. 56](#_Toc89121003)

[Hình 4‑15 Giao diện Website trang nhận diện gương mặt 57](#_Toc89121004)

[Hình 4‑16 Giao diện Website sau khi nhận diện được gương mặt 57](#_Toc89121005)

[Hình 4‑17 Giao diện Website xem kết quả học tập của sinh viên 58](#_Toc89121006)

# DANH MỤC BẢNG BIỂU

[Bảng 2.5‑1 Bảng BACDAOTAO (bậc đào tạo) 37](#_Toc88421837)

[Bảng 2.5‑2 Bảng LOAIHINHDAOTAO (loại hình đào tạo) 37](#_Toc88421838)

[Bảng 2.5‑3 Bảng KHOA (khoa) 37](#_Toc88421839)

[Bảng 2.5‑4 Bảng NGANH (ngành) 38](#_Toc88421840)

[Bảng 2.5‑5 Bảng LOP (lớp) 38](#_Toc88421841)

[Bảng 2.5‑6 Bảng NIENKHOA (niên khóa) 38](#_Toc88421842)

[Bảng 2.5‑7 Bảng HOCKY (học kỳ) 38](#_Toc88421843)

[Bảng 2.5‑8 Bảng MONHOC (môn học) 39](#_Toc88421844)

[Bảng 2.5‑9 Bảng SINHVIEN (sinh viên) 39](#_Toc88421845)

[Bảng 2.5‑10 Bảng DIEM (điểm) 40](#_Toc88421846)

# MỞ ĐẦU

Trong những năm gần đây, các hệ thống sinh trắc học như nhận dạng vân tay (Fingerprint 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 Hệ thống 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 ngành 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:

* Năm 2018, một nhóm sinh viên Đại học FPT, chuyên ngành Kỹ thuật phần mềm đã phát triển giải pháp hỗ trợ thông qua đồ án Attendance System using face recognition (hệ thống điểm danh sinh viên dùng nhận dạng khuôn mặt) [19].
* Ngày 5/9/2019 tại trường THCS Lê Quý Đôn TP.HCM, lần đầu tiên công nghệ nhận diện khuôn mặt điểm danh học sinh được tập đoàn NEC lắp đặt tại Việt Nam dựa trên nền trí tuệ nhân tạo. Công nghệ nhận diện gương mặt KAOATO được đưa vào điểm danh học sinh giúp tối đa hóa việc điểm danh, khi học sinh vừa bước vào lớp, ngay lập tức phụ huynh sẽ nhận được thông báo con đã tới lớp [20].
* Vào tháng 4/2019 Đại học Thăng Long đang thử nghiệm hệ thống điểm danh thông minh, tự động nhận diện khuôn mặt sinh viên đến lớp. Công nghệ nhận diện khuôn mặt "TLnet" của Đại học Thăng Long dù chỉ đang thử nghiệm nhưng độ chính xác và khả năng realtime rất cao, đạt 99,78%. Dự án do 1 giáo viên và 3 sinh viên chuyên ngành IT thực hiện, nhằm chống học hộ, điểm danh hộ, giám sát hành vi của sinh viên trên trường theo thời gian thực (realtime)...[21]
* Và mới đây nhất ngày 24/11 công nghệ nhận diện khuôn mặt của Tổng công ty Giải pháp doanh nghiệp Viettel (Viettel Solutions) đã vào top 30 bài kiểm tra đo lường và đánh giá chất lượng toàn cầu của Viện Tiêu chuẩn và Công nghệ quốc gia Mỹ (NIST). Kết quả thử nghiệm ấn tượng của Viettel Solutions. Được thế giới công nhận, bài đo của NIST là "dấu tích xanh" cho giá trị của các giải pháp công nghệ. "Việc vượt qua bài đánh giá còn góp phần khẳng định năng lực công nghệ của người Việt đủ khả năng cạnh tranh với giải pháp từ các công ty, tập đoàn hàng đầu thế giới", đại diện Viettel Solutions chia sẻ [22].

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 có một ưu điểm đáng kể so với các phương pháp 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 đấy tiện lợi hơn khi muốn xem bạn 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 chúng ta.

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) với phương pháp PCA thì phương pháp do nhóm đề xuất SVM kết hợp với đặc trưng HOG trong nhận dạng khuôn mặt cho kết quả tốt nhất [2]. 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 trang, 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 về tra cứu kết quả học tập dựa trên ảnh, tìm hiểu một số phương pháp trích chọn đặc trưng ảnh và tìm hiểu một số hệ thống tra cứu ảnh sẵn có.

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

Chương này tìm hiểu một số kỹ thuật phản hồi liên quan trong tra cứu học tập kết quả dựa trên ảnh, tìm hiểu các kỹ thuật kết hợp các đặc trưng hình ảnh trong SVM.

CHƯƠNG 4: THỰC NGHIỆM VÀ ĐÁNH GIÁ

Xây dựng chương trình thực nghiệm tra cứu kết quả học tập dựa trên ảnh kết hợp nhiều đặc trưng với phản hồi liên quan, đánh giá hiệu năng và một số kết quả đạt được.

# TỔNG QUAN ĐỂ TÀI

## 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, bạn 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, chúng tôi 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.

## Mục tiêu đề tài

* Xây dựng được bộ nhận diện tốt nhất cho phần mềm tra cứu kết quả học tập.

## Nội dung nghiên cứu

* Xây dựng một mô hình hoàn thiện trong việc tìm kiếm bằng hình ảnh và truy xuất nội dung tìm kiếm được.
* Xây dựng một bộ face detection (phát hiện khuôn mặt) từ hình ảnh:
* MTCNN (Multi-task Cascaded Convolutional Networks)
* Các thuật toán sử dụng cho việc trích xuất đặc trưng ảnh:
* HOG (của Histogram of Oriented Gradient)

- Xây dựng một model facerecognizer (nhận dạng khuôn mặt).

* SVM (Support Vector Machine)

Xây dựng database kết quả sinh viên.

* Xây dựng ứng dụng chụp hình khuôn mặt sinh viên và trả về kết quả học tập của sinh viên đó.

## Phạm vi đề tài

* Đề tài nghiên cứu về bộ dữ liệu hình ảnh và nội dung của bộ ảnh sinh viên khoa CNTT.
* Đề tài sử dụng HOG để rút trích đặc trưng của hình ảnh.
* Đề tài thực hiện xây dựng model học máy SVM để phân loại sinh viên.

## Phương pháp nghiên cứu

Nghiên cứu lý thuyết: Nghiên cứu cở sở lý thuyết về các đối tượng trong đề tài từ đó áp dụng vào việc xây dưng mô hình thực nghiệm.

Nghiên cứu thực nghiệm: Tiến hành nghiên cứu thực nghiệm trên bộ dữ liệu ảnh sinh viên khoa CNTT, qua đó rút ra được phương pháp tối ưu.

## 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: thư viện xử lý ảnh mã nguồn mở.
* Flask: thư viện xây dựng api của Python.
* Scikit-learning là một thư viện máy học phần mềm miễn phí cho ngôn ngữ lập trình Python.
* TensorFlow là một thư viện phần mềm mã nguồn mở dành cho máy học trong nhiều loại hình tác vụ nhận thức và hiểu ngôn ngữ.

# CƠ SỞ LÝ THUYẾT

## Giới thiệu về ảnh số

### Khái niệm

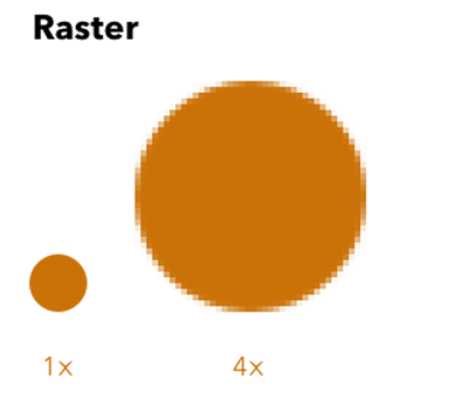
Ảnh số (digital image) là một thành phần biểu diễn hình ảnh trong hầu hết các thiết bị điện tử hiện nay như máy ảnh, điện thoại, máy tính, các công cụ hỗ trợ có sử dụng ảnh. Để bắt đầu với xử lý ảnh, chúng ta phải nắm được các kiến thức cơ bản về ảnh số và cách biểu diễn ảnh số trong máy tính. Bài viết này sẽ khái quát sơ lược về ảnh số và các không gian màu để biểu diễn ảnh số.

Ảnh số thực tế là biểu diễn số học của hình ảnh trong máy tính, thường là biểu diễn nhị phân. Có thể phân ảnh số thành 2 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.



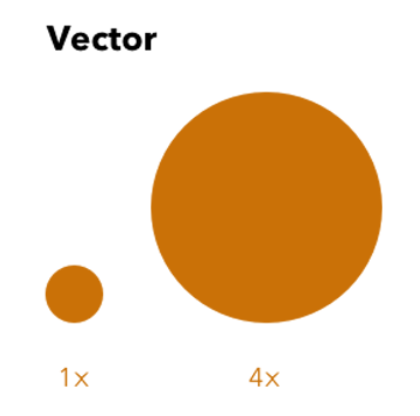
Hình 2‑1 Ảnh raster

#### 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 dãn, 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.



Hình 2‑2 Ảnh vector

### 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

### 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 1 ảnh xám cần có phải có giá trị kênh màu Red(x, y) = Green(x, y) = Blue(x, y) (với x, y lần lượt là tọa độ của điểm ảnh).

A close up of a cat with green eyes

Description automatically generated

Hình 2‑4 Ảnh xám

### 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.

A close up of a cat with green eyes

Description automatically generated

Hình 2‑5 Ảnh màu

## Phát hiện khuôn mặt (Face Detection)

### Khái niệm

Phát diện 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 diện 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 diện 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 diện 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?

### Phương pháp MTCNN

#### MTCNN là gì?

MTCNN là một phương pháp nhận diện khuôn mặt tuyệt vời được đề xuất bởi Zhang và cộng sự vào năm 2016. Phương pháp này có thể đạt được sự tích cực thực sự tỷ lệ 95,3% trên FDDB.

MTCNN là viết tắt của Multi-task Cascaded Convolutional Networks. Bao gồm 3 mạng CNN xếp chồng và đồng thời hoạt động khi detect khuôn mặt. Mỗi mạng có cấu trúc khác nhau và đảm nhiệm vai trò khác nhau trong task. Đầu ra của MTCNN là vị trí khuôn mặt và các điểm trên mặt như: mắt, mũi, miệng…

MTCNN - một thư viện python (pip) được viết bởi [người dùng Github ipacz](https://github.com/ipazc/mtcnn), triển khai [bài báo Zhang, Kaipeng et al. “Nhận diện khuôn mặt chung và căn chỉnh bằng cách sử dụng mạng kết nối đa nhiệm xếp tầng.” Thư xử lý tín hiệu IEEE 23.10 (2016): 1499–1503. Crossref. Web](https://arxiv.org/abs/1604.02878%5d(https:/arxiv.org/abs/1604.02878) . Trong bài báo này, họ đề xuất một khuôn khổ đa nhiệm vụ phân tầng sâu sử dụng các tính năng khác nhau của “mô hình con” để mỗi mô hình tăng cường các điểm mạnh tương quan của chúng.

MTCNN hoạt động khá nhanh trên CPU, mặc dù S3FD vẫn chạy nhanh hơn trên GPU - nhưng đó là một chủ đề cho một bài đăng khác.

#### Cách thức hoạt động 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, chúng tôi thường 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.

Ảnh có chứa văn bản, ảnh chụp màn hình

Mô tả được tạo tự động

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).

* Resize ảnh, để tạo một loạt các bản copy từ ảnh gốc với kích cỡ khác nhau, từ to đến nhỏ, tạo thành 1 Image Pyramid. Với mỗi một phiên bản copy-resize của ảnh gốc, ta sử dụng kernel 12x12 pixel và stride = 2 để đi qua toàn bộ bức ảnh, dò tìm khuôn mặt.
* Kết quả của mạng cho ra một loạt các bounding boxes nằm trong mỗi kernel, mỗi bounding boxes sẽ chứa tọa độ 4 góc để xác định vị trí trong kernel chứa nó (đã được normalize về khoảng từ (0,1)) và điểm confident (Điểm tự tin) tương ứng.
* Để loại trừ bớt các bounding boxes trên các bức ảnh và các kernels, ta sử dụng 2 phương pháp chính là lập mức Threshold confident - nhằm xóa đi các box có mức confident thấp và sử dụng NMS (Non-Maximum Suppression) để xóa các box có tỷ lệ trùng nhau. Những box bị trùng nhau sẽ bị loại bỏ và giữ lại 1 box có mức confident cao nhất.
* Kết quả là những tọa độ mới của các box.

**R-Net:** Refine network sử dụng đầu ra của P-Net để loại bỏ các vùng không phải là khuôn mặt.

* Tất cả các bounding box lúc này sẽ được resize về kích thước 24x24.
* Thực hiện các bước như mạng P-Net.
* Sử dụng phương pháp padding, nhằm thực hiện việc chèn thêm các zero-pixels vào các phần thiếu của bounding box nếu bounding box bị vượt quá biên của ảnh.
* Kết quả sau cùng là những tọa độ mới của các box còn lại.

**O-Net:** Output network sử dụng đầu ra của R-Net để đưa ra kết quả cuối cùng.

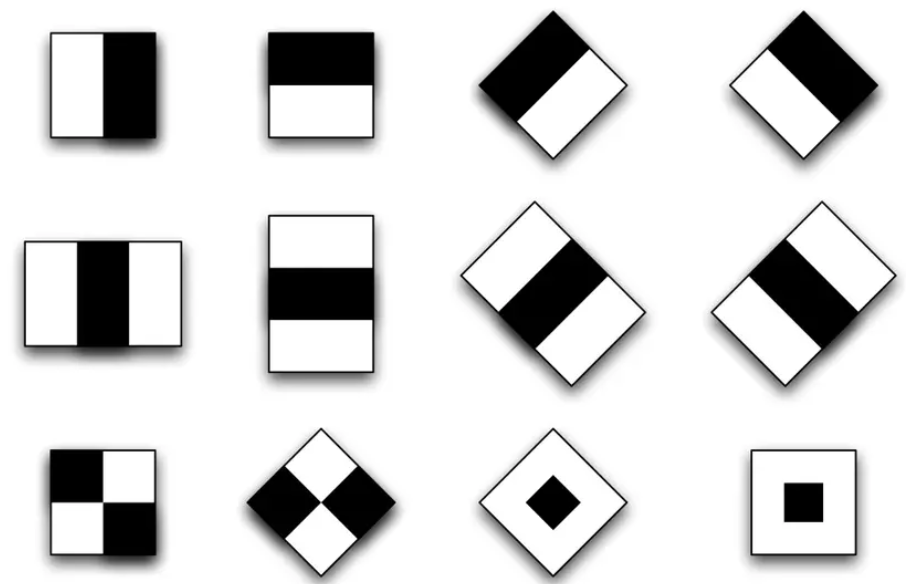
* Thay đổi kích thước thành 48x48.
* Thực hiện tương tự như việc trong mạng R-Net.
* Kết quả đầu ra 4 tọa độ của bounding box (out[0]), tọa độ 5 điểm landmark trên mặt, bao gồm 2 mắt, 1 mũi, 2 bên cánh môi (out[1]) và điểm confident của mỗi box (out[2]).

### 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:

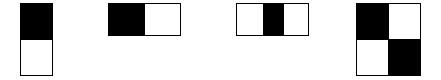
#### Đặc trưng haar-like

Các đặc trưng Haar-like là [các](https://en.wikipedia.org/wiki/Digital_image) [tính năng](https://en.wikipedia.org/wiki/Feature_(computer_vision)) [hình ảnh kỹ thuật số](https://en.wikipedia.org/wiki/Digital_image) được sử dụng trong [nhận dạng đối tượng](https://en.wikipedia.org/wiki/Object_recognition) . Chúng có tên gọi tương đồng trực quan với [các wavelet Haar](https://en.wikipedia.org/wiki/Haar_wavelet) 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.



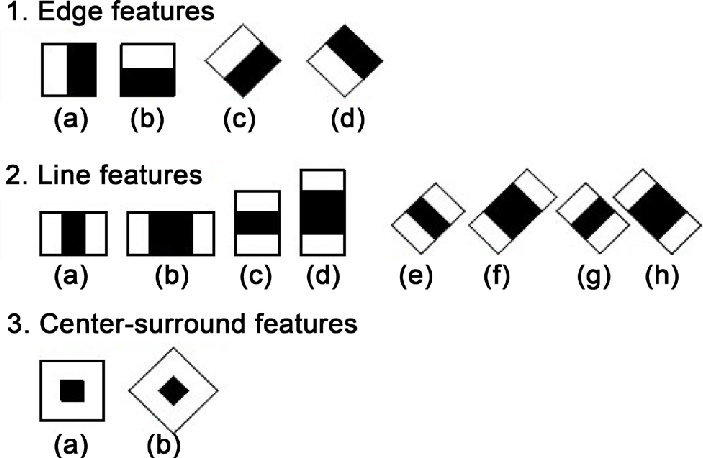
Hình 2‑7 Đặc trưng haar - like

Ưu điểm chính của tính năng Haar-like so với hầu hết các tính năng khác là tốc độ tính toán của nó. Do việc sử dụng [hình ảnh tích phân](https://en.wikipedia.org/wiki/Integral_image) (integral image), đặc điểm 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 2‑8 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:

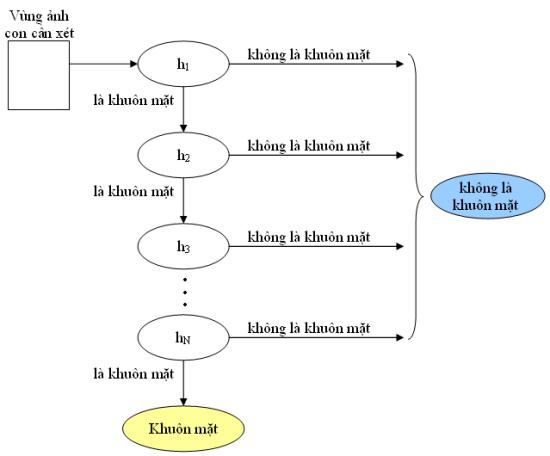


Hình 2‑9 Đặc trưng Haar – like mở rộng

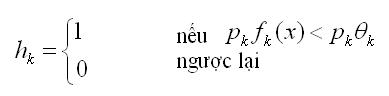
#### AdaBoost (Adaptive [Boosting](https://en.wikipedia.org/wiki/Boosting_(meta-algorithm)))

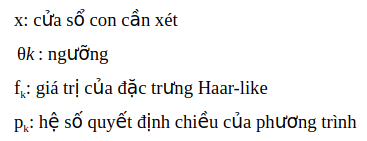
AdaBoostlà một [thuật toán tổng hợp](https://en.wikipedia.org/wiki/Meta-algorithm) [phân loại thống kê được](https://en.wikipedia.org/wiki/Statistical_classification) xây dựng bởi [Yoav Freund](https://en.wikipedia.org/wiki/Yoav_Freund) và [Robert Schapire](https://en.wikipedia.org/wiki/Robert_Schapire). Nó có thể được sử dụng kết hợp với nhiều loại thuật toán học tập khác để cải thiện hiệu suất. Adaboost cũng hoạt động trên nguyên tắc kết hợp tuyến tính các weak classifiers để hình thành một trong các classifiers.

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:

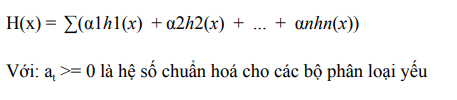


Trong đó, hk là các bộ phân loại yếu, được biểu diễn như sau:





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



## Trích xuất đặc trưng (Feature Extraction)

### 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. 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.

### Phương pháp

#### HOG

HOGlà viết tắt của Histogram of Oriented Gradient - một loại “feature descriptor”. Mục đích của “feature descriptor” 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.

HOGlà viết tắt của Histogram of Oriented Gradient - một loại “feature descriptor”. Mục đích của “feature descriptor” 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.

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 vectơ 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 HOG

##### Tiền xử lý

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

Ảnh có chứa văn bản, khác nhau

Mô tả được tạo tự động

Hình 3-3 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:

Ảnh có chứa văn bản

Mô tả được tạo tự động

Hình 3-4 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.

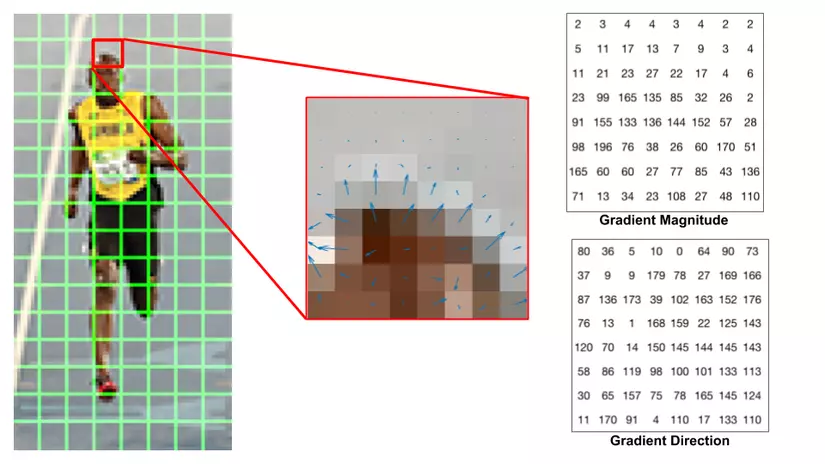
Giá trị bin được định lượng bởi tổng cường độ biến thiên của các pixels thuộc về bin đó. Sau khi tính toán đặc trưng ô, ta sẽ nối các vector đặc trưng ô để thu được vector đặc trưng khối.



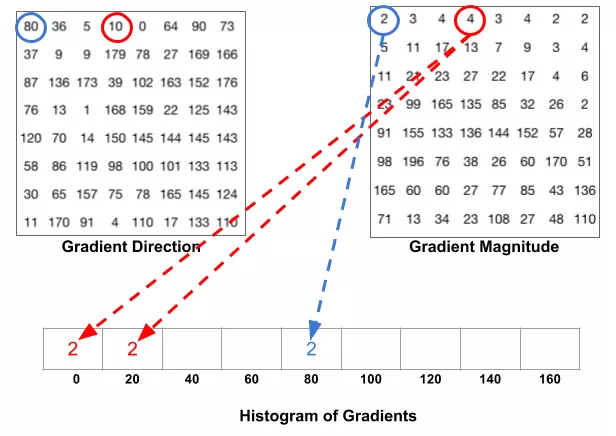
Hình 3-5 Chia hình ảnh thành các block

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 3-6 Tính toán đặc trưng HOG tại mỗi cell



Hình 3-7 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 64x128, chia thành các block 16x16 chồng nhau, sẽ có 7 block ngang và 15 block dọc, nên sẽ có 7x15 = 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 36x1.
* 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 105x36x1 = 3780x1.

#### 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:

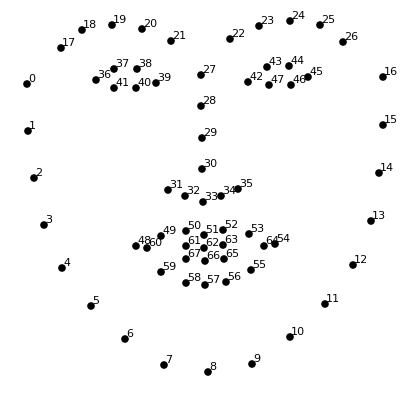
* Miệng
* Lông mày phải
* Lông mày trái
* Mắt phải
* Mắt trái
* Mũi
* Hàm

**4.5.2. 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](https://pdfs.semanticscholar.org/d78b/6a5b0dcaa81b1faea5fb0000045a62513567.pdf) của Kazemi và Sullivan (2014).

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 2‑10 68 điểm landmark

68 mốc sẽ xác định vị trí trên mọi khuôn mặt. Hình ảnh này được tạo bởi [Brandon Amos](http://bamos.github.io/) của CMU, người làm việc trên [OpenFace](https://github.com/cmusatyalab/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](https://ibug.doc.ic.ac.uk/resources/facial-point-annotations/)

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.

## Nhận dạng khuôn mặt (Face Recognition)

### Khái niệm

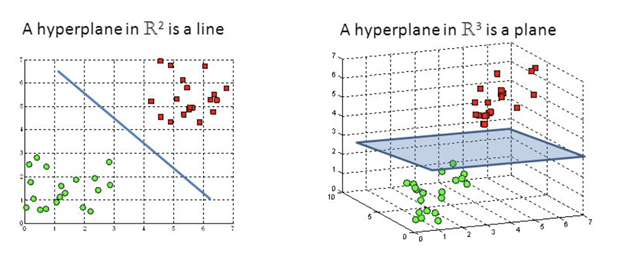
Hệ thống nhận dạng khuôn mặt là một ứng dụng máy tính tự động xác định hoặc nhận dạng một người nào đó từ một bức hình ảnh kỹ thuật số hoặc một khung hình video từ một nguồn video. Một trong những cách để thực hiện điều này là so sánh các đặc điểm khuôn mặt chọn trước từ hình ảnh và một cơ sở dữ liệu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

### 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.

#### Cách thức hoạt động của SVM

Ý tưởng hoạt động của SVM là tìm một siêu phẳng (hyper lane) 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 3-8 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.

Và sẽ có rất nhiều siêu phẳng chúng ta cần chọn ra cái nào để tối ưu nhất. Cách chọn siêu phẳng tối ưu trong điều kiện lý tưởng:

* Phân chia tốt nhất (tách biệt rõ ràng, lớp nào ra lớp đó)
* Khoảng cách lớn nhất (khoảng cách từ siêu phẳng đến từng lớp)
* Các hàm kernel thường dùng của SVM gồm LINEAR, POLY, RBF và SIGMOID. Các hàm Kernel khác nhau tạo ra các ánh xạ khác nhau cho tạo bề mặt phân cách phi tuyến tính.

#### Tham số của SVM

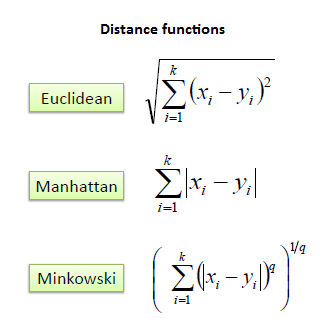
SVM cũng có một số siêu tham số (như giá trị C hoặc gamma để sử dụng) và việc tìm kiếm siêu tham số tối ưu là một nhiệm vụ rất khó giải quyết. Nhưng nó có thể được tìm thấy bằng cách thử tất cả các kết hợp và xem thông số nào hoạt động tốt nhất. Phương pháp này được gọi là Gridsearch (GridSearchCV).

### 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](https://en.wikipedia.org/wiki/Lazy_learning)), mọi tính toán được thực hiện khi nó cần dự đoán kết quả của dữ liệu mới.

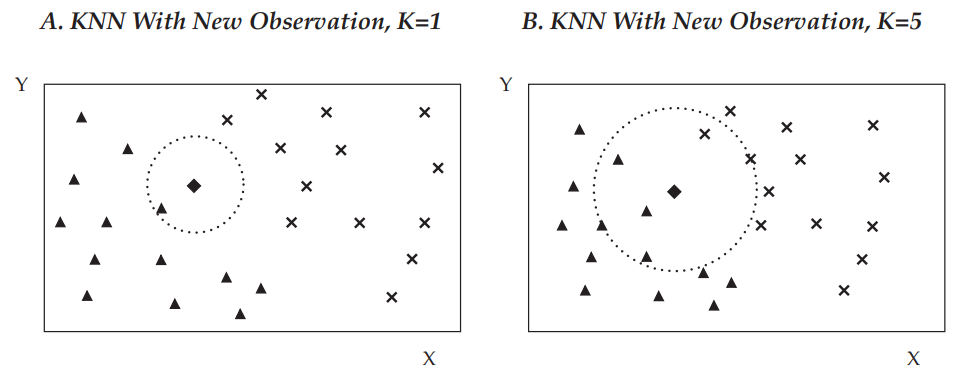
#### Cách thức hoạt động của KNN

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 củng 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:



Cách tính khoảng cách

#### Ví dụ cụ thể



Hình 2‑11 Phân loại hình thoi

Hình thoi trong Hình 2-11 trên đang cần được phân loại thuộc hình chữ thập hoặc hình tam giác.

* Nếu k = 1, hình thoi sẽ được phân loại vào cùng loại với điểm dữ liệu gần nhất của nó (tức là hình tam giác trong bảng bên trái - bảng A).
* Bảng bên phải (bảng B) thể hiện trường hợp k = 5, thuật toán sẽ xem xét 5 điểm dữ liệu gần hình thoi nhất, đó là 3 hình tam giác và 2 hình chữ thập. Qui tắc quyết định là chọn phân loại có số lượng lớn nhất trong 5 điểm dữ liệu được xem xét. Vì vậy, trong trường hợp này, hình thoi cũng được xếp vào phân loại tam giác.

### Cây quyết định (Decision Tree)

Cây quyết định ([Decision Tree](https://trituenhantao.io/tu-dien-thuat-ngu/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 tượngncó 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](https://trituenhantao.io/tu-dien-thuat-ngu/entropy/) và Information Gain để xây dựng một cây quyết định.

[Entropy](https://trituenhantao.io/tu-dien-thuat-ngu/entropy/) là thuật ngữ thuộc Nhiệt động lực học, là thước đo của sự biến đổi, hỗn loạn hoặc ngẫu nhiên. Năm 1948, Shannon đã mở rộng khái niệm [Entropy](https://trituenhantao.io/tu-dien-thuat-ngu/entropy/) sang lĩnh vực nghiên cứu, thống kê với công thức như sau:

Với một phân phối xác suất của một biến rời rạc x có thể nhận n giá trị khác nhau x1,x2,…,xn. Giả sử rằng xác suất để x nhận các giá trị này là pi=p(x=xi).

Ký hiệu phân phối này là p=(p1 ,p2 ,…,pn). [Entropy](https://trituenhantao.io/tu-dien-thuat-ngu/entropy/) của phân phối này được định nghĩa là:

H(p)=  – ∑nn=1 pi log(pi)

Information Gain dựa trên sự giảm của hàm [Entropy](https://trituenhantao.io/tu-dien-thuat-ngu/entropy/) khi tập dữ liệu được phân chia trên một thuộc tính. Để xây dựng một cây quyết định, ta phải tìm tất cả thuộc tính trả về Infomation gain cao nhất.

Để xác định các nút trong mô hình cây quyết định, ta thực hiện tính Infomation Gain tại mỗi nút theo trình tự sau:

**Bước 1**: Tính toán hệ số [Entropy](https://trituenhantao.io/tu-dien-thuat-ngu/entropy/) của biến mục tiêu S có N phần tử với Nc phần tử thuộc lớp c cho trước:

H(S)=  – ∑cc=1 (Nc/N) log(Nc/N)

**Bước 2**: Tính hàm số [Entropy](https://trituenhantao.io/tu-dien-thuat-ngu/entropy/) tại mỗi thuộc tính: với thuộc tính x, các điểm dữ liệu trong S được chia ra K child node S1, S2, …, SK với số điểm trong mỗi child node lần lượt là m1, m2 ,…, mK , ta có:

H(x, S) = ∑Kk=1 (mk / N) \* H(Sk)

**Bước 3**: Chỉ số Gain Information được tính bằng:

**G(x, S) = H(S) – H(x,S)**

#### Thuật toán C4.5

Thuật toán C4.5 là thuật toán cải tiến của ID3. Trong thuật toán ID3, Information Gain được sử dụng làm độ đo. Tuy nhiên, phương pháp này lại ưu tiên những thuộc tính có số lượng lớn các giá trị mà ít xét tới những giá trị nhỏ hơn. Do vậy, để khắc phục nhược điểm trên, ta sử dụng độ đo Gain Ratio (trong thuật toán C4.5) như sau:

Đầu tiên, ta chuẩn hoá information gain với trị thông tin phân tách (split information):

https://i1.wp.com/trituenhantao.io/wp-content/uploads/2020/04/Formula-10.jpg?w=810&ssl=1

Trong đó: Split Info được tính như sau:

https://i2.wp.com/trituenhantao.io/wp-content/uploads/2020/04/Formula-11.jpg?w=810&ssl=1

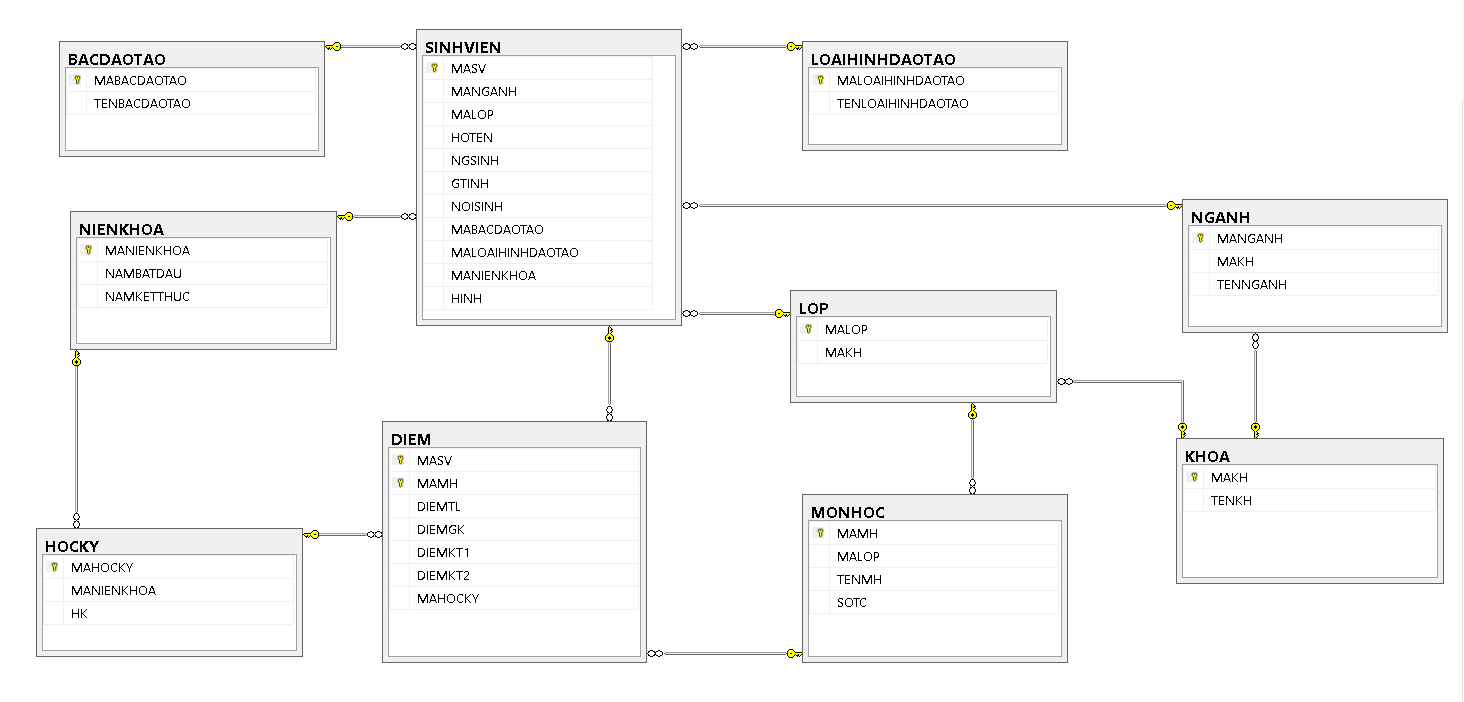
#### ****Tiêu chuẩn dừng****

Trong các thuật toán [Decision tree](https://trituenhantao.io/tu-dien-thuat-ngu/decision-tree/), với phương pháp chia trên, ta sẽ chia mãi các node nếu nó chưa tinh khiết. Như vậy, ta sẽ thu được một tree mà mọi điểm trong tập huấn luyện đều được dự đoán đúng (giả sử rằng không có hai input giống nhau nào cho output khác nhau). Khi đó, cây có thể sẽ rất phức tạp (nhiều node) với nhiều leaf node chỉ có một vài điểm dữ liệu. Như vậy, nhiều khả năng [overfitting](https://trituenhantao.io/tu-dien-thuat-ngu/overfitting/) sẽ xảy ra.

Để tránh trường họp này, ta có thể dừng cây theo một số phương pháp sau đây:

* Nếu node đó có [entropy](https://trituenhantao.io/tu-dien-thuat-ngu/entropy/) bằng 0, tức mọi điểm trong node đều thuộc một class.
* Nếu node đó có số phần tử nhỏ hơn một ngưỡng nào đó. Trong trường hợp này, ta chấp nhận có một số điểm bị phân lớp sai để tránh [overfitting](https://trituenhantao.io/tu-dien-thuat-ngu/overfitting/). Class cho leaf node này có thể được xác định dựa trên class chiếm đa số trong node.
* Nếu khoảng cách từ node đó đến root node đạt tới một giá trị nào đó. Việc hạn chế chiều sâu của tree này làm giảm độ phức tạp của tree và phần nào giúp tránh [overfitting](https://trituenhantao.io/tu-dien-thuat-ngu/overfitting/).
* Nếu tổng số leaf node vượt quá một ngưỡng nào đó.
* Nếu việc phân chia node đó không làm giảm [entropy](https://trituenhantao.io/tu-dien-thuat-ngu/entropy/) quá nhiều (information gain nhỏ hơn một ngưỡng nào đó).

## Thiết kế cơ sở dữ liệu



Hình 2‑12 Cơ sở dữ liệu

Bảng 2.5‑1 Bảng BACDAOTAO (bậc đào tạo)

|  |  |  |  |
| --- | --- | --- | --- |
| Tên cột | Kiểu dữ liệu | Mô tả | Ví dụ |
| **MABACDAOTAO** | VARCHAR(10) | Mã bậc đào tạo | DH, CD,… |
| TENBACDAOTAO | NVARCHAR(50) | Tên bậc đào tạo | Đại học, cao đẳng,… |

Bảng 2.5‑2 Bảng LOAIHINHDAOTAO (loại hình đào tạo)

|  |  |  |  |
| --- | --- | --- | --- |
| Tên cột | Kiểu dữ liệu | Mô tả | Ví dụ |
| **MALOAIHINHDAOTAO** | VARCHAR(10) | Mã loại hình đào tạo | CQ, VHVL,… |
| TENLOAIHINHDAOTAO | NVARCHAR(50) | Tên loại hình đào tạo | Chính quy, vừa học vừa làm,… |

Bảng 2.5‑3 Bảng KHOA (khoa)

|  |  |  |  |
| --- | --- | --- | --- |
| Tên cột | Kiểu dữ liệu | Mô tả | Ví dụ |
| **MAKH** | VARCHAR(10) | Mã khoa | CNTT, CK,… |
| TENKH | NVARCHAR(50) | Tên khoa | Khoa Công nghệ Thông tin,… |

Bảng 2.5‑4 Bảng NGANH (ngành)

|  |  |  |  |
| --- | --- | --- | --- |
| Tên cột | Kiểu dữ liệu | Mô tả | Ví dụ |
| **MANGANH** | VARCHAR(10) | Mã ngành | 7480103, 7480104 |
| MAKH | VARCHAR(10) | Mã khoa | CNTT, CK,… |
| TENNGANH | NVARCHAR(50) | Tên ngành | Hệ thống thông tin,… |

Bảng 2.5‑5 Bảng LOP (lớp)

|  |  |  |  |
| --- | --- | --- | --- |
| Tên cột | Kiểu dữ liệu | Mô tả | Ví dụ |
| **MALOP** | VARCHAR(10) | Mã lớp | 09DHTH5, 09DHTH8,… |
| MAKH | VARCHAR(10) | Mã khoa | CNTT, CK,… |

Bảng 2.5‑6 Bảng NIENKHOA (niên khóa)

|  |  |  |  |
| --- | --- | --- | --- |
| Tên cột | Kiểu dữ liệu | Mô tả | Ví dụ |
| **MANIENKHOA** | INT | Mã niên khoá | 08, 09,… |
| NAMBATDAU | INT | Năm bắt đầu | 2018 |
| NAMKETTHUC | INT | Năm kết thúc | 2022 |

Bảng 2.5‑7 Bảng HOCKY (học kỳ)

|  |  |  |  |
| --- | --- | --- | --- |
| Tên cột | Kiểu dữ liệu | Mô tả | Ví dụ |
| **MAHOCKY** | INT | Mã học kỳ | 1, 2, 3,… |
| MANIENKHOA | INT | Mã niên khoá | 08, 09,… |
| HK | VARCHAR(50) | Học kỳ | Học kỳ 1, Học kỳ 2,… |

Bảng 2.5‑8 Bảng MONHOC (môn học)

|  |  |  |  |
| --- | --- | --- | --- |
| Tên cột | Kiểu dữ liệu | Mô tả | Ví dụ |
| **MAMH** | VARCHAR(10) | Mã môn học | 0101003472, 0101006199,… |
| MALOP | 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 2.5‑9 Bảng SINHVIEN (sinh viên)

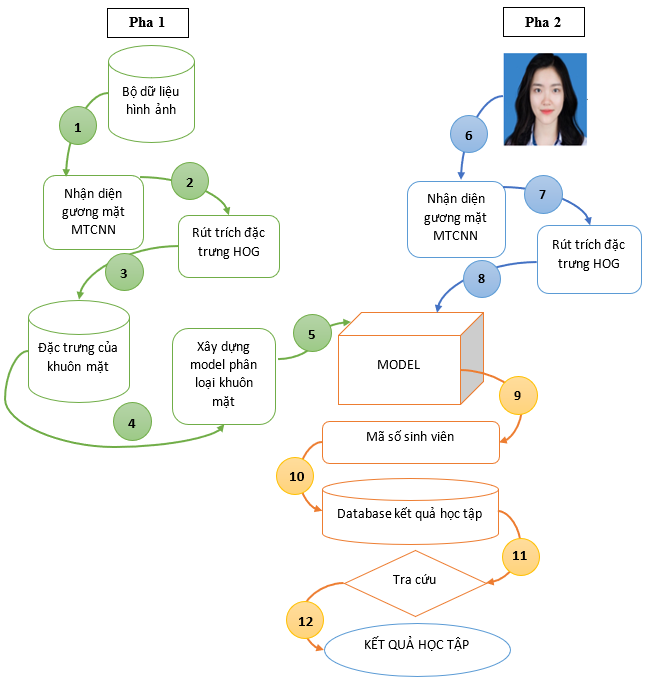
|  |  |  |  |
| --- | --- | --- | --- |
| Tên cột | Kiểu dữ liệu | Mô tả | Ví dụ |
| **MASV** | VARCHAR(10) | Mã sinh viên | 2001180476 |
| MANGANH | VARCHAR(10) | Mã ngành | 7480103 |
| MALOP | VARCHAR(10) | Mã lớp | 09DHTH5 |
| HOTEN | NVARCHAR(50) | Họ tên | Nguyễn Ngọc Hải |
| NGSINH | DATE | Ngành 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,… |
| MALOAIHINHDAOTAO | VARCHAR(10) | Mã loại hình đào tạo | CQ |
| MANIENKHOA | INT | Mã niên khoá | 09 |
| HINH | VARCHAR(50) | Hình | .jpg |

Bảng 2.5‑10 Bảng DIEM (điểm)

|  |  |  |  |
| --- | --- | --- | --- |
| Tên cột | Kiểu dữ liệu | Mô tả | Ví dụ |
| **MASV** | VARCHAR(10) | Mã sinh viên | 2001180476 |
| **MAMH** | 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 |
| **MAHK** | INT | Mã học kỳ | 1 |

# PHƯƠNG PHÁP NHẬN DẠNG SINH VIÊN VÀ TRA CỨU KẾT QUẢ

## Mô hình tổng quan



Hình 3-1 Mô hình tổng quát của hệ thống tra cứu kết quả học tập

## Các bước nhận dạng

### Đầu vào, huấn luyện

* Dữ liệu đầu vào là hình ảnh
* Dữ liệu sẽ được đưa qua MTCNN để nhận diện khuôn mặt, resize kích thước và lưu lại
* Sau đó dùng các khuôn mặt đã nhận diện được, rút trích đặc trưng HOG
* Cuối cùng là đem các đặc trưng đi học và cho ra một model

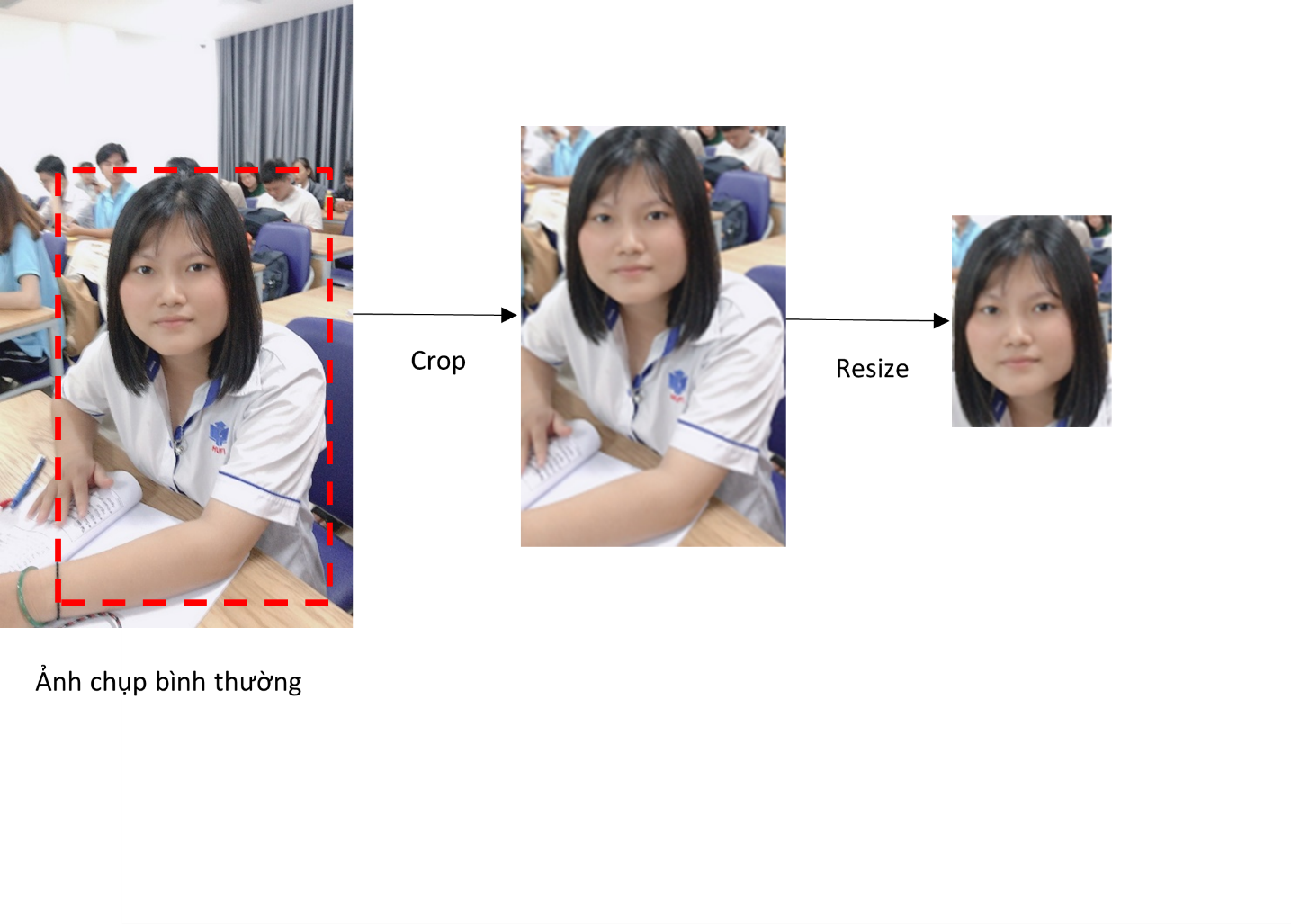
### Thực thi

* Đưa vào ảnh sinh viên
* Ảnh sẽ được đưa qua MTCNN để nhận diện khuôn mặt, resize kích thước
* Sau đó khuôn mặt nhận diện được sẽ đem đi rút trích đặc trưng HOG
* Đem đặc trưng đi nhận dạng bằng model đã huấn luyện và trả về kết quả là mã sinh viên
* Lấy mã sinh viên đi tra cứu trên database kết quả học tập sẽ tìm ra được kết quả học tập của sinh viên

## Chi tiết phương pháp nhận dạng

### Tiền xử lý

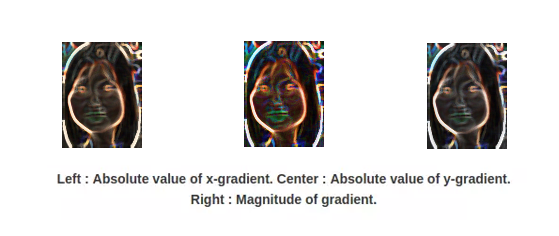
Resize kích thước tất cả các hình ảnh trong tập dữ liệu về một kích thước chung.



Hình 3-3 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:



Hình 3-4 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.

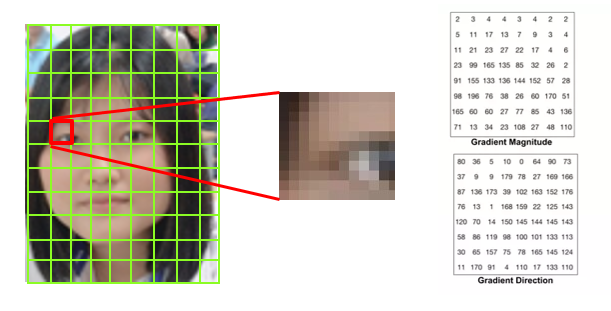
Ảnh có chứa động vật có vú, mờ

Mô tả được tạo tự động

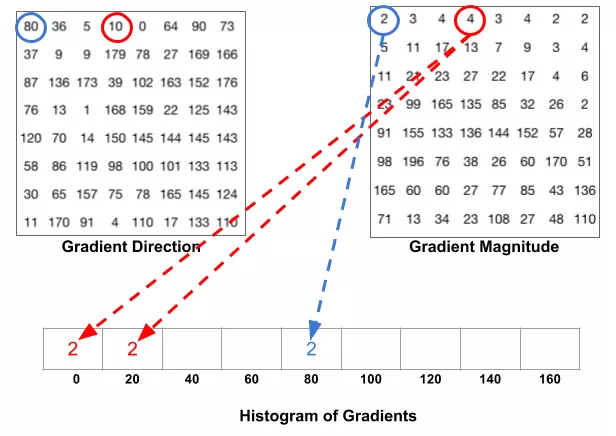
Hình 3-5 Chia hình ảnh thành các block

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 3-6 Tính toán đặc trưng HOG tại mỗi cell



Hình 3-7 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 256x256, chia thành các block 4x4 chồng nhau, sẽ có 5 block ngang và 5 block dọc, nên sẽ có 5x5 = 25 blocks.
* 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, ta sẽ thu được vector đặc trưng HOG của ảnh có kích thước 25x144x1 = 3600x1.

### Cách thức hoạt động của SVM

# THỰC NGHIỆM VÀ ĐÁNH GIÁ

## Kết quả thực nghiệm

### Dữ liệu thực nghiệm

Tập ảnh YaleFaceB của 28 người mỗi người 9 tư thế và 64 điều kiện ánh sáng.



Hình 4‑1 Tập ảnh YaleFaceB

### Phương pháp thực nghiệm

#### 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 x 5 x 4 x 4 x 9 = 3600 chiều**.

#### 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 toạ độ (x, y) của từng điểm ta chuyển nó về dạng numpy
* Giờ ta có được 1 mảng 2 chiều gồm các toạ độ (x, y)
* Ta duỗi nó thành mảng 1 ta được 1 vector có **136 chiều**

## So sánh phương pháp

Chúng tôi thực hiện cài đặt trên google colab và sử dụng GPU để tiến hành chạy thực nghiệm.

### Độ chính xác

* Độ chính xác của HOG

Ảnh có chứa bàn

Mô tả được tạo tự động

Hình 4‑2 Kết quả train đặc trưng HOG bằng SVM trên tập ảnh YaleFacesB

* Độ chính xác của Land-mark

Ảnh có chứa bàn

Mô tả được tạo tự động

Hình 4‑3 Kết quả train đặc trưng Facial landmark bằng SVM trên tập ảnh YaleFacesB

#### Biểu đồ thể hiện độ chính xác

Hình 4‑4 Biểu đồ thể hiện độ chính xác của HOG và Landmark

* Vì có thể lấy được nhiều đặc trưng hơn nên độ chính xác của HOG tốt hơn của Land-mark

### Tốc độ hai phương pháp

#### Tốc độ xử lý

* Tốc độ xử lý của HOG

Ảnh có chứa văn bản

Mô tả được tạo tự động

Hình 4‑5 Thời gian xử lý rút trích đặc trưng HOG

* Tốc độ xử lý của Land-mark

Ảnh có chứa văn bản

Mô tả được tạo tự động

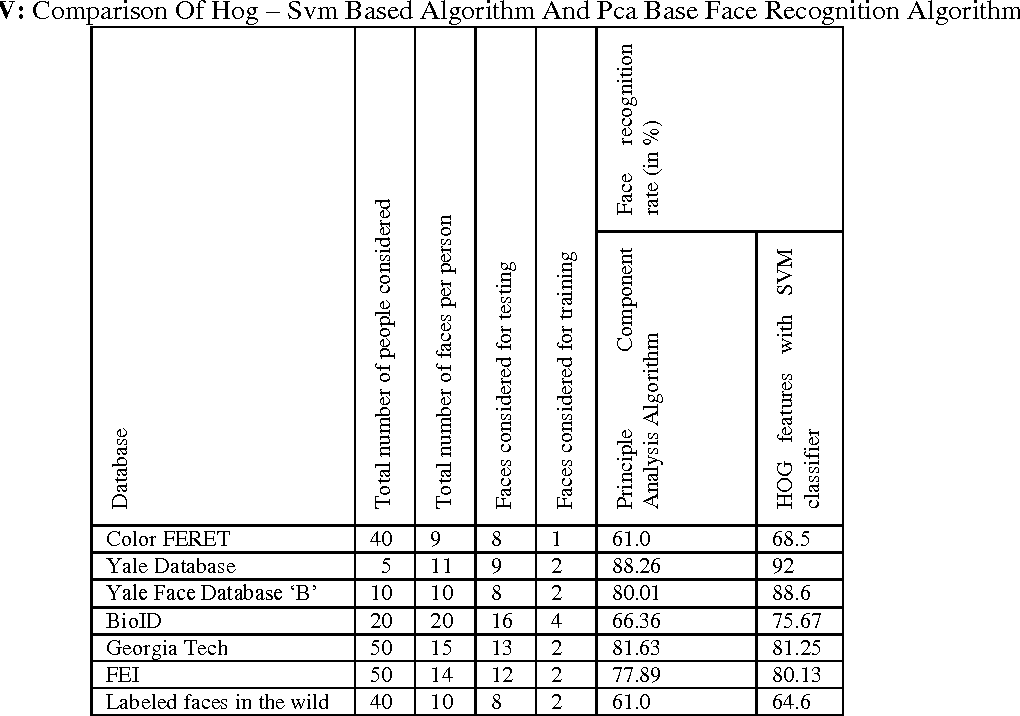
Hình 4‑6 Thời gian xử lý rút trích đặc trưng Facial landmark

#### Biểu đồ thể hiện thời gian xử lý

Hình 4‑7 Biểu đồ thể hiện thời gian xử lý

* Vì kích thước đặc trưng của Land-mark chỉ có 136 nên thời gian xử lý nhanh hơn HOG với kích thước đặc trưng lên đến 3600

### So sánh HOG+SVM với PCA



Hình 4‑8 Hình ảnh so sánh giữa HOG+SVM và PCA

## Cài đặt trên bộ ảnh sinh viên

Tập ảnh sinh viên khoa Công nghệ Thông tin gồm 80 bạn mỗi bạn từ 10-20 ảnh (trường Đại học Công nghiệp Thực phẩm TP. Hồ Chí Minh)

Ảnh có chứa văn bản

Mô tả được tạo tự động

Hình 4‑9 Tập ảnh sinh viên khoa Công nghệ thông tin

### Độ chính xác khi train và validate model bằng KFold

Ảnh có chứa văn bản

Mô tả được tạo tự động

Hình 4‑10 Độ chính xác khi train và validate model với Kfold = 10

### Độ chính xác của model khi kiểm định với tập test

Ảnh có chứa bàn

Mô tả được tạo tự động

Hình 4‑11 Độ chính xác của model khi kiểm chứng với tập test

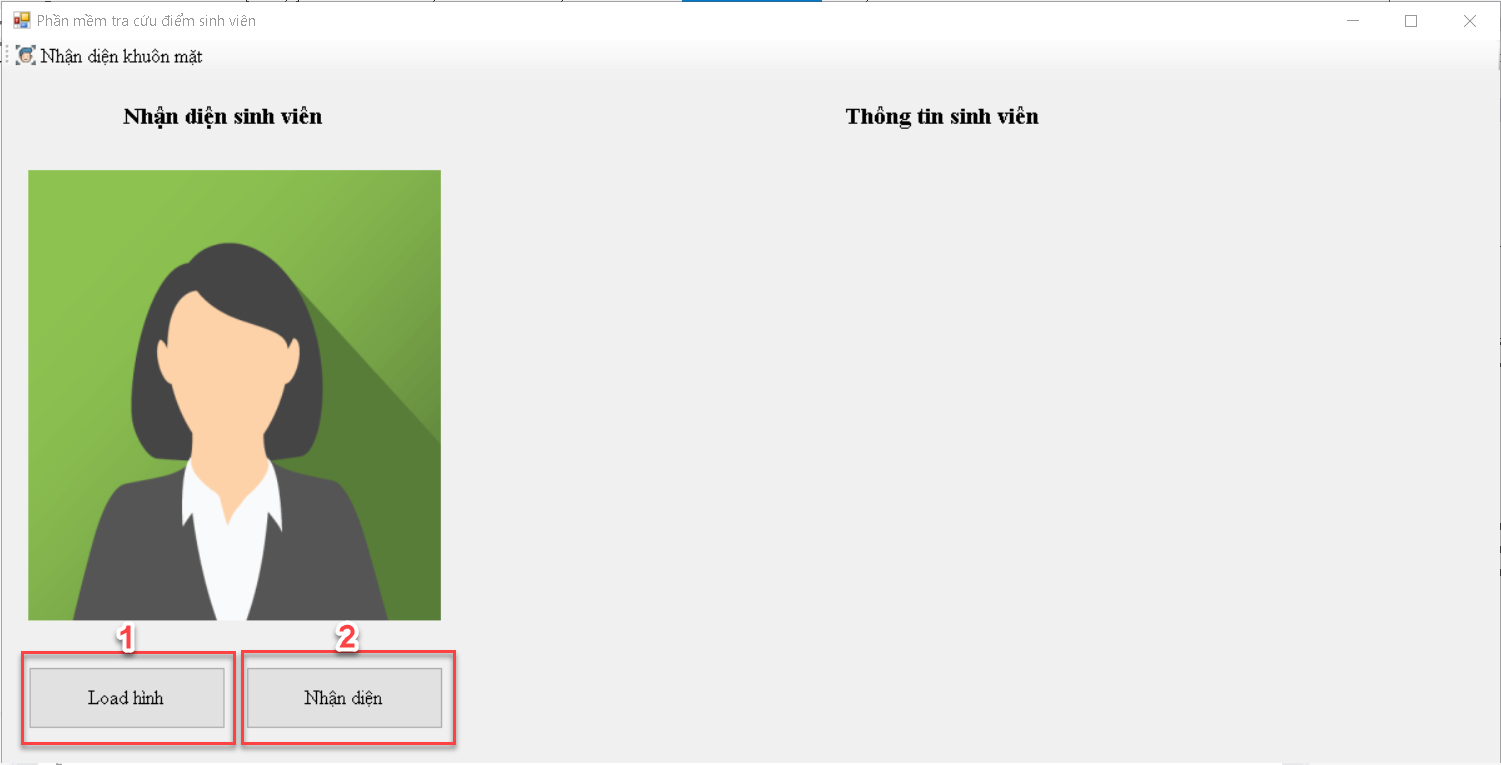
## Giao diện chương trình

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.

### Giao diện ứng dụng

#### Giao diện chọn ảnh sinh viên:

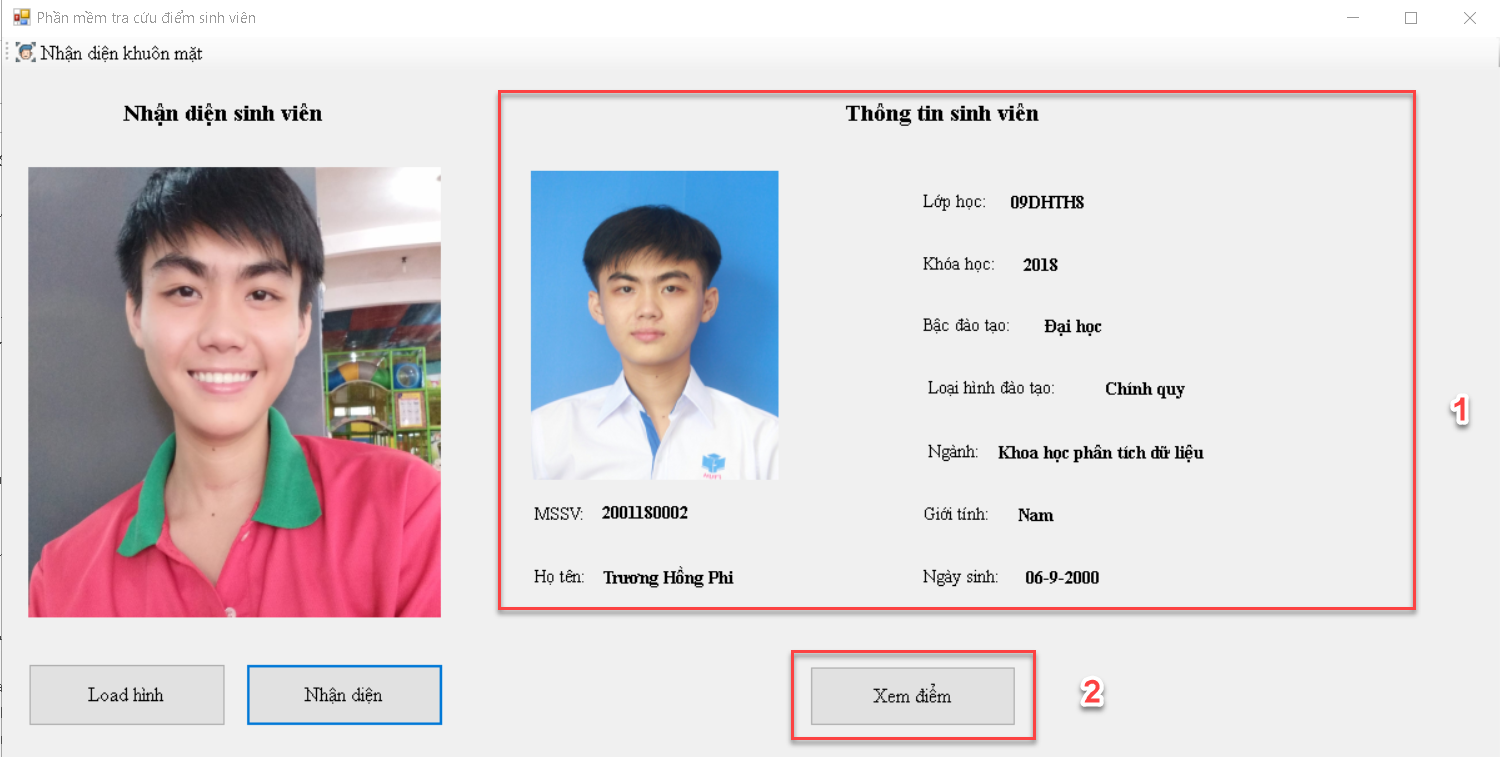
* Số 1: button Load hình cần nhận diện.
* Số 2: button Nhận diện



Hình 4‑12 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



Hình 4‑13 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

Ảnh có chứa bàn

Mô tả được tạo tự động

Hình 4‑14 Giao diện xem điểm sinh viên.

### Giao diện website

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

Ảnh có chứa văn bản

Mô tả được tạo tự động

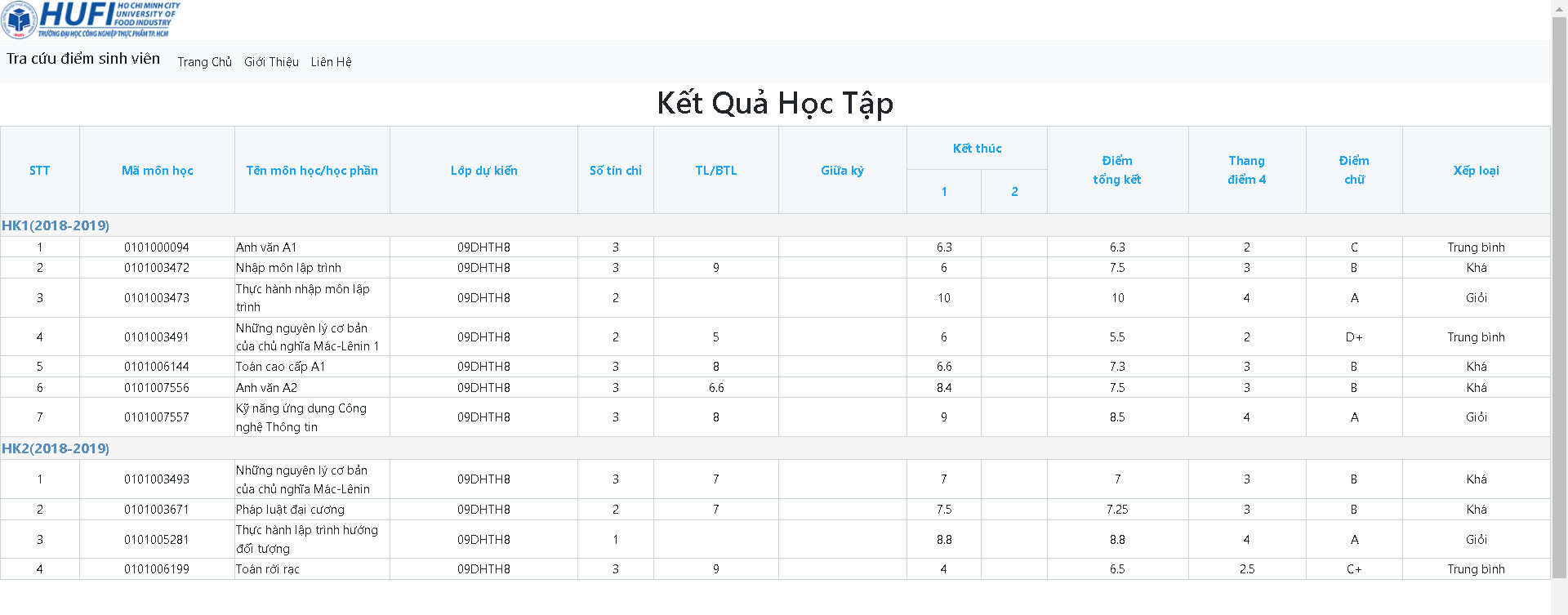
Hình 4‑15 Giao diện Website trang nhận diện gương mặt

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

Ảnh có chứa văn bản

Mô tả được tạo tự động

Hình 4‑16 Giao diện Website sau khi nhận diện được gương mặt



Hình 4‑17 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](https://cyberlotus.com/xac-thuc-sinh-trac-hoc-la-gi-cac-loai-sinh-trac-hoc-pho-bien.html)

[2] [https://www.semanticscholar.org/paper/Recognition-Using-HOG-Features](https://www.semanticscholar.org/paper/Improved-Face-Recognition-Rate-Using-HOG-Features-Dadi-Krishna/7fc5ab3743e6e9a2f4fe70152440e13a673e239b/figure/7)

[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/](https://www.researchgate.net/profile/Nguyen-Khang-Pham/publication/332198692_DIEM_DANH_BANG_MAT_NGUOI_VOI_DAC_TRUNG_GIST_VA_MAY_HOC_VECTO_HO_TRO/links/5f7c36c8458515b7cf6a31b7/DIEM-DANH-BANG-MAT-NGUOI-VOI-DAC-TRUNG-GIST-VA-MAY-HOC-VECTO-HO-TRO.pdf)

[7]<https://viso.ai/deep-learning/face-detection-overview/> - PP Face Recognizer

[8] [https://www.mygreatlearning.com/](https://www.mygreatlearning.com/blog/real-time-face-detection/) - Viola & Jones

[9]<https://taqadam.io/open-sourse-datasets/> - Datasets

[10] [https://docs.opencv.org](https://docs.opencv.org/3.4/db/d28/tutorial_cascade_classifier.html) – CascadeClassifier

[11] [https://ietresearch.onlinelibrary.wiley.com/MTCNN](https://ietresearch.onlinelibrary.wiley.com/doi/pdfdirect/10.1049/iet-ipr.2019.0141)

[12] [http://citeseerx.ist.psu.edu/viewdoc/SVM](http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.437.1036&rep=rep1&type=pdf)

[13] [https://cafedev.vn/sieu-tham-so-svm](https://cafedev.vn/tu-hoc-ml-dieu-chinh-sieu-tham-so-svm-bang-gridsearchcv-ml/)

[14][https://www.learnopencv.com/histogram-of-oriented-gradients/](https://www.learnopencv.com/histogram-of-oriented-gradients/?fbclid=IwAR18at-SxmVl7Uqb8_7wjpj5r6T0NzpSK-g9yc8i2QWd8WD_RGfKyVQZ-4U) - Histogram of Oriented Gradients

[15][http://mccormickml.com/2013/05/09/hog-person-detector-tutorial/](http://mccormickml.com/2013/05/09/hog-person-detector-tutorial/?fbclid=IwAR2d9GJgH8ZFgVGAwE0SUMndXUk2LN6kbb88ik1CSJLqyH3LBo4fMCSQXv4) - [HOG Person Detector Tuto](http://mccormickml.com/2013/05/09/hog-person-detector-tutorial/?fbclid=IwAR2d9GJgH8ZFgVGAwE0SUMndXUk2LN6kbb88ik1CSJLqyH3LBo4fMCSQXv4)rial

[16]<https://blog.vietnamlab.vn/dlib-phan-2-xac-dinh-facial-landmark-voi-dlib-va-python-2/> - Facial landmark

[17] <https://ichi.pro/vi/lop-phu-mat-na-voi-opencv-dlib-85300355583053> F.landmar

[18] <https://machinelearningmastery.com/k-fold-cross-validation/> K-Foldx

x

[19] [https://hanoi.fpt.edu.vn/sinh-vien-fpt-diem-danh-bang-khuon-mat](https://hanoi.fpt.edu.vn/sinh-vien-dh-fpt-diem-danh-bang-nhan-dien-khuon-mat.html?amp=1)

[20] [https://vietnamdigitalsignage.com/nhan-dien-khuon-mat-kaoato](https://vietnamdigitalsignage.com/detail/nhan-dien-khuon-mat-kaoato-tai-truong-thcs-le-quy-don-tphcm-501)

[21] <https://www.facebook.com/172550369587828/posts/1250231841819670/>

[22] [https://vnexpress.net/nhan-dien-khuon-mat-viettel-solutions-dat-tieu-chuan-my](https://vnexpress.net/giai-phap-nhan-dien-khuon-mat-viettel-solutions-dat-tieu-chuan-my-4393979.html)