**TRƯỜNG ĐẠI HỌC ĐIỆN LỰC**

**KHOA CÔNG NGHỆ THÔNG TIN**



**BÁO CÁO CHUYÊN ĐỀ HỌC PHẦN**

**NGÔN NGỮ LẬP TRÌNH PYTHON**

**ĐỀ TÀI: NHẬN DIỆN KHUÔN MẶT SỬ DỤNG THUẬT**

**TOÁN SVM**

|  |  |
| --- | --- |
|  |  |
| |  |  | | --- | --- | | **Sinh viên thực hiện** | **: NGUYỄN VĂN NAM**  **HÀ QUÝ ĐỨC** | | **Giảng viên hướng dẫn** | **: PHẠM THỊ KIM DUNG** | | | **Ngành** | **: CÔNG NGHỆ THÔNG TIN** | | | **Chuyên ngành** | **: CÔNG NGHỆ PHẦN MỀM** | | | **Lớp** | **: D13CNPM5** | | | **Khóa** | **: 2018 - 2023** | | |  |

**Hà Nội, tháng 01 năm 2022**

**PHIẾU CHẤM ĐIỂM**

**Sinh viên thực hiện:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Mã Sinh Viên** | **Họ tên sinh viên** | **Nhiệm vụ** | **Điểm** | **Chữ ký** |
| 18810310428 | Nguyễn Văn Nam | Phân tích + code + báo cáo |  |  |
| 18810310435 | Hà Quý Đức | Phân tích + code |  |  |

**Giản viên chấm điểm:**

|  |  |  |
| --- | --- | --- |
| **Họ và tên** | **Nhận Xét** | **Chữ Ký** |
| Giảng Viên 1 |  |  |
| Giảng Viên 2 |  |  |

**MỤC LỤC**

Contents

[CHƯƠNG 1: TỔNG QUAN VỀ NGÔN NGỮ PYTHON VÀ THƯ VIỆN TKINTER 1](#_Toc91966804)

[1.1. Tổng quan về ngôn ngữ lập trình Python 1](#_Toc91966805)

[1.2. Các tính năng nổi bật 1](#_Toc91966806)

[1.3. Mục đích của việc sử dụng ngôn ngữ lập trình Python 2](#_Toc91966807)

[1.4. Tổng quan về Tkinter 3](#_Toc91966808)

[1.4.1. Giới thiệu 3](#_Toc91966809)

[1.4.2. Các widget cơ bản 3](#_Toc91966810)

[1.4.3. Bố cục trong Python Tkinter 4](#_Toc91966811)

[2.1. Giới thiệu 5](#_Toc91966812)

[2.2. Support vector classifier phân lớp nhị phân với SVC 6](#_Toc91966813)

[2.3. Vấn đề dữ liệu không phân tách tuyến tính 8](#_Toc91966814)

[CHƯƠNG 3: TÌM HIỂU BÀI TOÁN NHẬN DIỆN KHUÔN MẶT 11](#_Toc91966815)

[SỬ DỤNG THUẬT TOÁN SVM 11](#_Toc91966816)

[3.1. Tổng quan công nghệ sinh trắc học, bài toán nhận dạng khuôn mặt và ứng dụng của nhận dạng khuôn mặt 11](#_Toc91966817)

[3.1.1. Tổng quan sinh trắc học 11](#_Toc91966818)

[3.1.2. Bài toán nhân dạng khuôn mặt 12](#_Toc91966819)

[3.1.3. Ứng dụng cho hệ thống nhận dạng khuôn mặt 14](#_Toc91966820)

[3.2. Kỹ thuật nhận dạng khuôn mặt 17](#_Toc91966821)

[3.2.1. Finding all the Faces 18](#_Toc91966822)

[3.2.2. Posing and Projecting Faces 20](#_Toc91966823)

[3.2.3. Encoding Faces 21](#_Toc91966824)

[3.2.4. Finding the person’s name from the encoding 23](#_Toc91966825)

[3.3. Đánh giá thực nghiệm 24](#_Toc91966826)

[3.3.1. Độ đo đánh giá hiệu quả 24](#_Toc91966827)

[3.3.2. Môi tường thực nghiệm 24](#_Toc91966828)

[3.3.3. Dữ liệu thực nghiệm 26](#_Toc91966829)

[3.4. Kết quả thực nghiệm 27](#_Toc91966830)

[3.4.1. Độ chính xác 27](#_Toc91966831)

[3.4.2. Thời gian 27](#_Toc91966832)

[3.4.3. Giao diện thể hiện kết quả thực nghiệm 28](#_Toc91966833)

[TÀI LIỆU THAM KHẢO. 30](#_Toc91966834)

**DANH MỤC HÌNH ẢNH**

[Hình 2.1: Minh hoạ phân tách tuyến tính 6](#_Toc67170632)

[Hình 2.2: Margin trong SVM 7](#_Toc67170633)

[Hình 2.3: Soft margin SVM khi có nhiễu 8](#_Toc67170634)

[Hình 3.1: Các đặc tính sinh trắc học của con người 11](#_Toc67170635)

[Hình 3.2: Hệ thống xác minh nhân thân của công dân nhập cảnh 16](#_Toc67170636)

[Hình 3.3: Hình ảnh gốc được chuyển thành hình ảnh đại diện HOG 19](#_Toc67170637)

[Hình 3.4: Tìm phần của ảnh giống nhất với mẫu HOG đã biết 19](#_Toc67170638)

[Hình 3.5: kết quả detect face 19](#_Toc67170639)

[Hình 3.6: 68 điểm trên khuôn mặt 20](#_Toc67170640)

Hình 3.7: Xác định mốc mặt cho khuôn mặt 21

[Hình 3.8: Huấn luyện mạng nơ ron tự động trích rút đặc trưng 22](#_Toc67170642)

[Hình 3.9: Kết quả của việc trích chọn đặc trưng khuôn mặt 23](#_Toc67170643)

[Hình 3.10: Phân lớp khuôn mặt 23](#_Toc67170644)

[Hình 3.11: Ngôn ngữ sử dụng python 24](#_Toc67170645)

[Hình 3.12: Tập dữ liệu thực nghiệm 26](#_Toc67170646)

[Hình 3.13: Kết quả thực nghiệm 27](#_Toc67170647)

[Hình 3.14: Kết quả thử nghiệm với ảnh có một người 28](#_Toc67170648)

[Hình 3.15: Kết quả thử nghiệm với ảnh có hai người 28](#_Toc67170649)

**DANH MỤC BẢNG**

[Bảng 3.1: Độ chính xác 27](#_Toc67170632)

[Hình 3.2: Thời gian thực hiện detect một ảnh và train một ảnh 27](#_Toc67170633)

**LỜI MỞ ĐẦU**

Chúng ta đã biết với thế giới ngày nay cùng sự phát triển mạnh mẽ của kỹ thuật số và mạng toàn cầu thì vấn đề đảm bảo an toàn thông tin, vật chật cũng ngày càng trở nên quan trọng. Hiện nay, những công nghệ hiện đại đã cho phép xác thực hay nhận dạng một cá nhân dựa vào các đặc trưng sinh lý học của người đó như đặc điểm về dấu vân tay, đặc điểm về khuôn mặt, giọng nói, và nhiều đặc điểm khác.

Nhận diện khuôn mặt hay tiếng anh là Face Recognition là một trong số ít các phương pháp nhận dạng dựa vào các đặc trưng sinh lý cho kết quả độ chính xác cao và thuận tiện áp dụng vào thực tế để sử dụng như áp dụng nhận dạng khuôn mặt để nhận dạng tội phạm, điểm danh sinh viên của một trường đại học, quản lý nhân viên trong các công ty, giải pháp bảo mật bổ sung cho các giao dịch rút tiền tự đông ATM, … Chính vì những nhu cầu thực tiễn đó nên nhóm em đã quyết định lựa chọn nghiên cứu để tài “ Nhận dạng khuôn mặt sử dụng thuật toán SVM”.

Qua quá trình nghiên cứu, tham khảo một số chương trình nhận dạng cũng như được sự giúp đỡ nhiệt tình của giảng viên **Phạm Thị Kim Dung** đã giúp cho nhóm em nâng cao hơn được kỹ thuật lập trình bằng ngôn ngữ Python, kỹ năng nghiên cứu, tìm hiểu tài liệu liên quan đến đề tài. Tuy nhiên trong quá trình làm thì không thể tránh khỏi những thiếu xót mong các thầy cô có thể góp ý để cho đề tài của chúng em được hoàn thiện đầy đủ hơn.

**LỜI CẢM ƠN**

Chúng em xin chân thành cảm ơn giảng viên **Phạm Thị Kim Dung** đã truyền đạt một cách tận tình những kiến thức cần thiết, bổ ích về học phần **“Ngôn ngữ lập trình Python”** trong những buổi học trên lớp cũng như qua quá trình trao để có thể hoàn thành tốt bài báo cáo này, nhóm em đã tìm hiểu rất nhiều thông tin về nhận dạng khuôn mặt trên các trang mạng như google, các group trên facebook, … để chọn lọc những phương pháp chính xác cũng như để hoàn thành tốt đề tài đã chọn với tất cả sự nỗ lực. Tuy nhiên, với sự hiểu biết còn hạn hẹp chưa được đầy đủ nên không thể tránh khỏi những thiếu xót. Chúng em rất mong được sự quan tâm, giúp đỡ cũng như những đóng góp quý báu của thầy cô để có thể hoàn thiện bài báo cáo cũng như hoàn thành tốt đề tài đã chọn.

Sau cùng, chúng em xin kính chúc các thầy cô trong Khoa **Công Nghệ Thông Tin** dồi dào sức khoẻ và mọi điều tốt đẹp để có thể tiếp tục thực hiện sứ mệnh cao đẹp của mình, truyền đạt được nhiều kiến thức hay – bổ ích cho thế hệ trẻ mai sau.

Một lần nữa, chúng em xin chân thành cảm ơn!

Nhóm thực hiện

Nguyễn Văn Nam

Hà Quý Đức

# 

# **CHƯƠNG 1: TỔNG QUAN VỀ NGÔN NGỮ PYTHON VÀ THƯ VIỆN TKINTER**

* 1. **Tổng quan về ngôn ngữ lập trình Python**

Python là ngôn ngữ lập trình hướng đối tượng thông dụng dùng để viết các tiện ích hệ thống. Nó cũng được sử dụng như ngôn ngữ kết dính đóng vai trò tích hợp C và C++.

Python là ngôn ngữ lập trình bậc cao, phục vụ cho mục đích đa năng. Ưu điểm nổi bật nhất đó chính là dễ đọc, dễ nhớ, dễ học. Ngoài ra, Python là ngôn ngữ có cấu trức tương đối rõ ràng thuận tiện cho người mới bắt đầu nó còn cho phép người dùng viết mã lệnh với số lần gõ tối thiểu.

Được tạo ra bởi Guido van Rossum tại Amsterdam năm 1990, Python hoàn toàn tạo kiểu động và dùng cơ chế cấp phát bộ nhớ tự động. Python được pháp triển trong một dự án mở do tổ chức phi lợi nhuận Python Software Foundation quản lý.

Ban đầu Python được phát triển để chạy trên nền Unix nhưng rồi theo thời gian nó được mở rộng ra mọi hệ điều hành từ MS-DOC đến MAC OS, OS/2, Windows, Linux và các hệ điều hành khác thuộc họ Unix.

Trải qua hơn 30 năm hình thành và phát triển, Python hiện có 2 version phổ biến chính là python 2 và python 3.

* 1. **Các tính năng nổi bật**

Ngôn ngữ lập trình Python có những tính năng nổi bật như sau:

**Miễn phí, mã nguồn mở:** Người dùng có thể thỏa mái sử dụng và phân phối Python thậm chí là có thể sử dụng chúng để phục vụ cho mục đích thương mại. Bởi chúng là mã nguồn mở, người dùng không chỉ sử dụng các phần mềm chương trình uược viết trong Python mà còn có thể thay đổi mã nguồn. Python có một cộng đồng lớn, thường xuyên được cập nhật và không ngừng cải tiến.

**Ngôn ngữ lập trình đơn giản, dễ đọc**: Python có cấu trúc ngữ pháp đơn giản, rõ ràng. Nó dễ đọc và viết đơn giản nhiều hơn so với các ngôn ngữ như lập trình Java, C#, Python làm cho việc lập trình trở nên thú vị giúp cho các lập trình viên tập trung vào những giải pháp chứ không phải là cú pháp.

**Khả năng di chuyển**: Các chương trình trên Python có thể di chuyển từ nền tảng này đến nền tảng khác mà không gặp phỉa bất kỳ thay đổi nào khi chạy. Nó chạy liền mạch trên các nền tảng như Mac, Windows, Linux.

**Khả năng mở rộng và có thể nhúng**: Nếu một ứng dụng đòi hỏi sự phức tạp lớn, người dùng có thể kết hợp dễ dàng với các phần code bằng C, C++ và những ngôn ngữ khác vào code python. Bởi vậy sẽ giúp ứng dụng của bạn có những tính năng tốt hơn, khả năng scripting mà các ngôn ngữ lập trình khác khó có thể làm được.

**Ngôn ngữ thông dịch cấp cao**: với python bạn không cần lo lắng về việc quản lý bộ nhớ, dọn dẹp những dữ liệu vô nghĩa, … khi chạy ngôn ngữ python sẽ tự chuyển đổi code sang ngôn ngữ để máy tính có thể hiểu và người dùng cũng không cần lo lắng về bất kỳ hoạt động ở cấp thấp nào.

**Hướng đối tượng**: Mọi thứ của python đều là hướng đối tượng. Lập trình đối tượng sẽ giúp giải quyết các vấn đề một cách trực quan nhất. Với lập rình đối tượng, bạn có thể phân chia nhiều phức tạp thành các tập nhỏ hơn bằng cách tạo ra các đối tượng.

**Thư viện chuẩn lớn**: Ngôn ngữ python có thư viện lớn giúp cho việc lập trình trở lên dễ dàng hơn vì không phải tự viết tất cả code. Những thư viện này được kiểm tra kỹ lưỡng nên chắc chắn nó không làm hỏng code hay ứng dụng nào của bạn.

* 1. **Mục đích của việc sử dụng ngôn ngữ lập trình Python**

**Viết các ứng dụng web:** Lập trình viên có thể tạo ra các web app có khả năng mở rộng bằng cách sử dụng framwork và CMS (hệ thống quản trị nội dung) có trong Python. Các nền tảng ứng dụng web được sử dụng phổ biến như Django, Flask, Pyramid.

**Khoa học, phân tích số liệu**: Trong có python có nhiều thư viện khoa học và tính toán số liệu. Hiếm có một ngôn ngữ nào có được sức mạnh như Python khi nó có thể sự dụng trong Data Analytics, Machine Learning và Computing Science. Từ số lượng thư viện chuẩn lớn và các plugin python hoàn toàn đáp ứng được công việc phân tích và tính toán số liệu, ngoài ra còn có rât nhiều ứng dụng khác.

* 1. **Tổng quan về Tkinter**
     1. **Giới thiệu**

Tkinter là một gói trong Python có chứa module Tk hỗ trợ cho việc lập tình giao diện bằng Python. Tk ban đầu được viết cho ngôn ngữ TCL. Sau đó Tkinter được viết ra để sử dụng Tk bằng trình thông dịch TCL trên nền Python.

Giao diện người dùng đồ họa (GUI) là một ứng dụng máy tính để bàn giúp cho người dùng tương tác với máy tính. Chúng được sử dụng thực hiện các tác vụ khác nhau trong máy tính để bản, máy tính xách tay và các thiết bị điện tử khác.

Các ứng dụng GUI như trình chỉnh sửa văn bản được sử dụng để tạo, đọc cập nhật và xóa các loại tệp khác nhau. Các ứng dụng GUI nuhw Sudoku, Chess là những trò chơi mà người dùng có thể chơi. Các ứng dụng GUI như Google Chrome, Firefox, Microsoft Edge được sử dụng để duyêt qua Internet.

* + 1. **Các widget cơ bản**

Tkinter Widgets: các widget giống như các phần tử HTML. Người dùng sẽ tìm thấy các loại vật dụng khác nhau với các loại phần tử khác nhau trong Tkinter.

* Canvas: được sử dụng dùng dể vẽ các hình dạng trong GUI.
* Button: Tiện ích nút được sử dụng để đặt các nút trong Tkinter
* CheckButton: được sử dụng để tạo ra các nút kiểm tra trong ứng dụng của bạn. Lưu ý bạn có thể chọn nhiều tùy chọn cùng một lúc.
* Entry: được sử dụng dùng để làm thùng chứa trong tkinter.
* Lable: được dùng để tạo ra các tiện ích con một dòng văn bản, hình ảnh.
* Menu: dùng để tạo ra menu trong GUI.

Tiện ích nhãn (Lable trong Tkinter): nhãn được sử dụng để tạo ra các văn bản và hình ảnh và tất cả những thứ đó nhưng quan trọng cần lưu ý là nó chỉ đinh nghĩa một dòng duy nhất.

Tiện ích Button: sử dụng để tạo ra các button trong ứng dụng Python. Các button này có thể hiển thị văn bản hoặc hình ảnh để thể hiện mục đích của chúng. Bạn có thể gán một hàm hoặc một phương thức vào một button hoặc phương thức này sẽ được tự động gọi khi click vào button đó.

Message: dùng để hiển thị hộp tin nhắn cho người dùng. MessageBox dùng để hiển thị các thông báo trong các ứng dụng desktop.

* + 1. **Bố cục trong Python Tkinter**

Python Tkinter cung cấp các phương thức để bố cục các widget như sau:

**Phương thức pack():** được sử dụng để tổ chức các widget theo khối. Vị trí các widget được thêm vào ứng dụng python bằng phương thức pack() có thể được kiểm soát bằng cách sử dụng các tùy chọn khác nhau được chỉ định trong lệnh gọi phương thức.

**Phương thức gird():** được dùng dể sắp xếp các widget ở dạng bảng. Chúng ta có thẻ chỉ định các hàng và cột. Chúng ta cũng có thể chỉ định khoảng cột chiều rộng hoặc chiều dài hàng của widget con. Một số phương thức gird() như column: số cột của widget con, cột ngoài cùng bên trái được biểu thị bằng số 0; row: số hàng mà widget con được đặt. Hàng trên cùng được biểu thị bằng 0.

**Phương thức place():** phương thức được sử dụng trong báo cáo này, nó sắp xếp các widget theo các tọa độ x và y. dưới đây là một số danh sách các tùy chọn như là x và y là đề cập đến độ lệch ngang và dọc theo pixel; height và width là đề cập tới chiều cao và chiều rộng của widget tính bằng pixel.

**CHƯƠNG 2: TÌM HIỂU THUẬT TOÁN SVM**

# **2.1. Giới thiệu**

Support Vector Machine (SVM) là kỹ thuật mới đối với việc phân lớp dữ liệu, là phương pháp học sử dụng không gian giả thuyết các hàm tuyến tính trên không gian giả thuyết các hàm tuyến tính trên không gian đặc trưng nhiều chiều, dựa trên lý thuyết tối ưu và lý thuyết thống kê.

Support Vector Machine (SVM) là phương pháp mạnh và chính xác nhất trong số các thuật toán nổi bật ở lĩnh vực khai phá dữ liệu. SVM bao gồm support vector classifier (SVC), bộ phân lớp dựa theo hỗ trợ và support vector regressor (SRV), bộ hồi quy dựa theo vector hỗ trợ.

Được phát triển đầu tiên bởi Vapnik vào những năm 1990, SVM có nền tảng lý thuyết được xây dựng trên nền móng lý thuyết xác suất thống kê. Nó yêu cầu số lượng mẫu huấn luyện không nhiều và thường không nhạy cảm với số chiều của dữ liệu. Trong nhiều thập niên qua, SVM đã phát triển nhanh chóng cả về mặt lý thuyết lẫn thực nghiệm.

SVM là một phương pháp dựa trên nền tảng của lý thuyết thống kê nên có một nền tảng toán học chặt chẽ để đảm bảo rằng kết quả tìm được là chính xác.

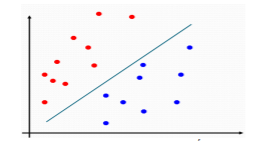
SVM Là một thuật toán học có giám sát (supervied learning) được sử dụng cho phân lớp dữ liệu.

SVM Là một phương pháp thử nghiệm, đưa ra một trong những phương pháp mạnh và chính xác nhát trong số các thuật toán nổi tiếng về phân lớp dữ liệu.

SVM là một phương pháp có tính tổng quát cao nên có thể áp dụng cho nhiều loại bài toán nhận dạng và phân loại.

# **2.2. Support vector classifier phân lớp nhị phân với SVC**

Xét một ví dụ của bài toán phân lớp như hình vẽ, ở đó ta phải tìm một đường thẳng sao cho phía bên trái toàn các điểm màu đỏ, bên phải nó toàn các điểm màu xanh. Bài toán mà dùng được thẳng để phân chia này được gọi là phân lớp tuyến tính (linear classification)



Hình 2.1: Minh hoạ phân tách tuyến tính

Hàm tuyến tính phân biệt bởi hai lớp như sau: yx = wT x+ b

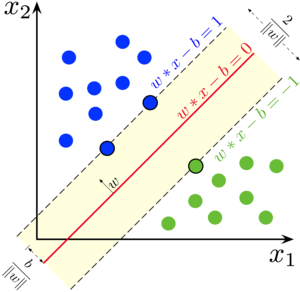
Trong đó:

* w ∊ Rm là vector trọng số hay vector chuẩn của siêu phẳng phân cách.
* T là kí hiệu chuyển vị.
* b ∊ R là độ lệch.

Lưu ý: với không gian hai chiều thì đường thẳng phân cách là đường thẳng, nhưng đối với không gian đa chiều thì đường thẳng phân cách gọi là siêu phẳng.

Tập dữ liệu đầu vào gồm N mẫu input vector {x1, x2, …, xn}, với các giá trị nhãn tương ứng là {t1, t2, …, tn} trong đó t ∊ {-1, 1}. Giả sử tập dữ liệu có thể phân tách tuyến tính hoàn toàn (nghĩa là các mẫu đều được phân tách đúng lớp bởi đường thẳng phân cách) khi đó, giá trị tham số w và b theo yx = wT x+ b luôn tồn tại và thoả mãn y(x1) >0 cho nhưng điểm có nhãn là t= +1 và y(x1) < 0 cho những điểm có t= -1, vì thế mà tiy(xi)>0 cho mọi điểm dữ liệu huấn luyện.

Để tìm đường phân cách, SVC thông qua khái niệm gọi là lề, đường biên (margin). Lề là khoảng cách nhỏ nhất giữa điểm dữ liệu gần nhất đến một điểm bất kỳ trên đường phân cách



Hình 2.2: Margin trong SVM

Theo SVC, đường phân cách tốt nhất là đường có margin lớn nhất. điều này có nghĩa là tồn tại rất nhiều phương pháp xoay theo các phương khác nhau và khi đó phương pháp sẽ chọn ra đường phân cách mà có margin lớn nhất tức là cần tìm ra một mặt phẳng H0: y = wx+b =0 và 2 siêu phẳng H+, H- hỗ trợ song song với H0 và có cùng khoảng cách đến H0. Với điều kiện không có phần tử nào nằm giữa H+, H- khi đó:

H+: wTx+b >=1 với t = +1

H-: wTx+b >=-1 với t = -1

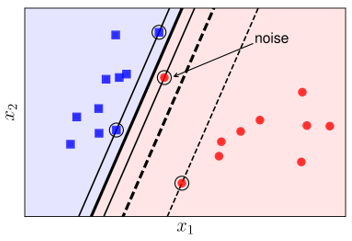
Kết hợp 2 điều kiện trên có y(wT+b)>=1 Khoảng cách từ H+ và H- đến H0 là . Cần tìm siêu phẳng với lề lớn nhất, là giải bài toán tối ưu tìm cho w và b sao cho đạt cực đại với ràng buộc yi(wTxi+b)>=1. Tương đương với bài toán cực tiểu hoá với điều kiện yi(wTxi+b)>=1 với mọi i thuộc từ 1 đến n.

Lời giải cho bài toán tối ưu này là cực tiểu hoá hàm Lagrange:

L (w, b, a) = ||w. w|| -

Trong đó, a là hệ số Lagrange, a>=0.

# **2.3. Vấn đề dữ liệu không phân tách tuyến tính**



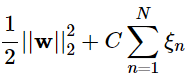
Hình 2.3: Soft margin SVM khi có nhiễu

Việc yêu cầu dữ liệu phân tách tuyến tính hoàn toàn là nghiêm ngặt hơn và không phù hợp với các bài toán thực tế, đặc biết là các trường hợp phân lớp phi tuyến tính phức tạp. Trong khi đó, các mẫu không phân tách tuyến tính hoàn toàn dẫn đến việc không thể giải quyết các bài toán tối ưu để tìm w và b tương ứng. Để giải quyết vấn đề này có hai cách tiếp cận chính: soft-margin và thủ thuật kernel.

**Soft-Margin:**

Để có một margin lớn hơn trong soft margin SVM chúng ta cần hi sinh một vài điểm dữ liệu bằng cách chấp nhận cho chúng rơi vào vùng không an toàn. Tất nhiên, chúng ta phải hạn chế việc hy sinh này nếu không chúng ta có thể tạo ra một biên cực lớn hơn bằng cách hy sinh hầu hết các điểm. với hàm mục tiêu là một sự kết hợp để tối đa margin và tối thiểu sự hy sinh.

Với mỗi điểm xn trong tập dữ liệu huấn luyện, thêm một biến đo sự hy sinh ξn tương ứng. Biến này còn được gọi là slack variable. Với những điểm xn nằm trong vùng an toàn ξn = 0. Với mỗi điểm nằm trong vùng không an toàn như x1, x2 hay x3 ta có ξi > 0. Nhận thấy rằng nếu yi = +1 hay yi = -1 là nhãn của xi trong vùng không an toàn thì ξi = | wTxi+b-yi|. khi đó hàm mục tiêu sẽ có dạng:



Trong đó C là một hằng số dương và ξ = [ξ1, ξ2, …, ξn].

Hằng số C được dùng để điều chỉnh tầm quan trọng giữa margin và sự hy sinh.

Điều kiện ràng buộc sẽ làm thay đổi. Với mỗi cặp dữ liệu (xn, yn) thày vì ràng buộc yn (wTxn+b) >=1, chúng ta sẽ có thêm ràng buộc:

yn (wTxn+b) >1- ξn với mọi n = 1, 2, …, n, ξn > 0

Nếu C nhỏ, việc sự hy sinh cao hay thấp không gây ảnh hưởng nhiều tới giá trị của hàm mục tiêu, thuật toán sẽ chỉnh cho margin là lớn nhất điều này dẫn tới sẽ lớn theo.

Ngược lại, nếu C quá lớn thì để hàm mục tiêu đạt giá trị nhỏ nhất thuật toán sẽ tập trung vào làm giảm . Trong tược hợp C rất lớn và hai lớp là linearly separable ta sẽ thu được .

**Thủ thuật Kernel:**

Nói một cách ngắn gọn, Kernel SVM là việc đi tìm một hàm biến đổi dữ liệu x từ không gian feature ban đầu dữ liệu trong một không gian mới bằng hàm số ɸ(x). Hàm ɸ () đơn giản là giới thiệu thêm một chiều dữ liệu mới (một feature mới) là một hàm số của các feature đã biết.

Hàm này thoả mãn mục đích của chúng ta: trong không gian mới dữ liệu giữa 2 classes là phân biệt tuyến tính hoặc gần như phân biệt tuyến tính. Khi đó, ta có thể dùng các bộ lớp phân lớp tuyến tính thông thường như Hard/ Soft Margin SVM, …

Các hàm ɸ () thường tạo dữ liệu mới có số chiều cao hơn số chiều của dữ liệu ban đầu, thậm chí là vô hạn chiều. Nếu tính toán các hàm trực tiếp chắc chắn gặp các vấn đề về bộ nhớ và hiệu năng tính toán.

**Một số hàm kernel thông dụng:**

**Linear:** đây là trường hợp đơn giản với kernel chính tích vô hướng của hai vector:

k (x, z) = xTz

Khi sử dụng sklearn. svm. SVC, kernel này được chọn bằng cách đặt kernel = ‘linear’.

**Polynomial:**

k (x, z) = (r + ℽxTz) d

với d là một số dương chỉ bậc của đa thức. d có thể không là số tự nhiên vì mục đích chính của ta không phải là bậc của đa thức mà là cách tính kernel. Polynomial kernel có thể dùng để mô tả hầu hết các đa thức có bậc không vượt quá d nếu d là một số tự nhiên.

Khi sử dụng thư viện sklearn, kernel này được chọn bằng cách đặt kernel = ‘poly’.

**Radial Basic Function:**

Radial Basic Function kernel hay Gausian kernel được sử dụng nhiều nhất trong thực tế là lựa chọn mặc định trong sklearn. Nó được định nghĩa bởi:



Khi sử dụng thư viện sklearn, kernel này được chọn bằng cách sử dụng kernel = ‘rbf’.

**Sigmoid:**

Sigmoid function cũng được sử dụng làm kernel:

k (x, z) = tanh (r + ℽxTz)

khi sử dụng thư viện sklearn, kernel này được chọn bằng cách sử dụng kernel = ‘sigmoid’.

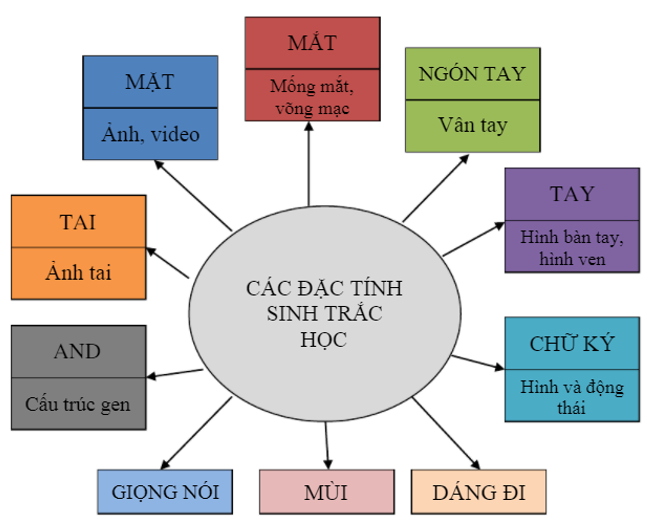
# **CHƯƠNG 3: TÌM HIỂU BÀI TOÁN NHẬN DIỆN KHUÔN MẶT**

# **SỬ DỤNG THUẬT TOÁN SVM**

**3.1. Tổng quan công nghệ sinh trắc học, bài toán nhận dạng khuôn mặt và ứng dụng của nhận dạng khuôn mặt**

## **3.1.1. Tổng quan sinh trắc học**

Có rất nhiều phương pháp được sử dụng để nhận dạng người, sinh trắc học là một phương pháp hiệu quả trong việc nhận dạng người. Sinh trắc học hay công nghệ sinh trắc học (tiếng anh: Biometric) là công nghệ sử dụng tính chất vật lý, đặc điểm sinh học riêng của mỗi cá nhân như vân tay, mống mắt, khuôn mặt, giọng nói, … để nhận diện. Đây là công cụ được coi là công cụ xác thực nhân thân hữu hiệu và phổ biến nhất hiện nay do mỗi người có những đặc điểm sinh trắc học duy nhất.



Hình 3.1: Các đặc tính sinh trắc học của con người

Những năm gần đây, kỹ thuật sinh trắc học trong nhận dạng cá nhân nổi lên một cách đầy hứa hẹn., thay vì chứng thực người và cho phép họ truy cập vào các hệ thống dựa trên các phương thức như thông qua mật khẩu, thẻ thông minh, … thì phương pháp kiểm tra một đặc tính sinh lý cá nhân hoặc hành vi để xác định danh tính người dùng. Mật khẩu thì nó khó nhớ và có thể bị đánh cắp, bị lộ, bị đoán ra. Thẻ thông minh và chìa khoá và những cái tương tự có thể bị làm giả, bị đánh cắp hay bị mất. Tuy nhiên đặc điểm sinh trắc học nhận dạng dựa trên đặc điểm sinh lý như khuôn mặt, dấu vân tay, hình ngón tay, hình học mặt, mống mắt, võng mạc, giọng nói hay đặc điểm hành vi như dáng đi, chữ ký, …

Tuy nhiên, hầu hết các phương pháp sinh trắc học yêu cầu phải có một hay nhiều hành động chủ quan của người sử dụng, ví dụ người sử dụng cần đặt tay lên thiết bị quét vân tay để lấy được mẫu vân tay cần được phát hiện hoặc cần phải đứng ở một vị trí cố định ở phía trước của một máy ảnh cho võng mạc, mống mắt được xác định. Bên cạnh đó việc thu thập dữ liệu cho hệ thống nhận dạng bằng vân tay có thể trở lên vô dụng khi mô biểu bì bị hư hỏng bởi một số lý do như bị thương, bị khuyết định, việc thu dữ liệu cho hệ thống nhận dạng bằng mống mắt, võng mạc có thể không thực hiện đuọc nếu đối tượng nhận diện chuyển động.

## **3.1.2. Bài toán nhân dạng khuôn mặt**

Nhận dạng khuôn mặt (Face Recognition) là một phương pháp sinh trắc để xác định hoặc xác minh một cá nhân nào đó bằng cách so sánh những dữ liệu hình ảnh chụp trực tiếp hoặc hình ảnh kỹ thuật số với bản ghi được lưu trữ cho người đó. Nó được xem là một lĩnh vực nghiên cứu của ngành Biometrics (tương tự nhận dạng vân tay- Fingerprint Recognition, hay nhận diện mống mắt- Iris Recognition). Xét về nguyên tắc chung, nhận dạng khuôn mặt có sự tương đồng rất lớn với nhận dạng vân tay và nhận dạng mống mắt nhưng sự khác biết nằm ở bước trích chọn đặt trưng (feature extraction) của mỗi lính vực.

Trong khi nhận dạng vân tay và nhận dạng mống mắt đã đạt tới độ chín tức là có thể áp dụng thực tế một cách rộng rãi thì nhận dạng khuôn mặt vẫn còn rất nhiều thách thức và vẫn là một lĩnh vực nghiên cứu đầy thú vị thu hút rất nhiều người ở Việt Nam cũng như ở thế giới.

So với nhận dạng vân tay và nhận dạng mống mắt thì nhận dạng khuôn mặt có nhiều nguồn dữ liệu phong phú hơn, chúng ta có thể nhìn thấy mặt người ở bất cứ nơi đâu như các tấm ảnh, đoạn video, … và ít đòi hỏi sự tương tác kiểm soát hơn (để thực hiện nhận dạng vân tay và nhận dạng mống mắt thì dữ liệu input được lấy từ con người đòi hỏi phải có sự hợp tác trong môi trường kiểm soát).

Các hệ thống nhận dạng thường được sử dụng cho những mục đích an ninh như kiểm soát an ninh của toà nhà, sân bay, máy ATM, tra cứu thông tin tội phạm, phát hiện tội phạm ở nơi công cộng, … và ngày càng được áp dụng rộng rãi trong đời sống sinh hoạt, ….

Bên cạnh những thành công đã được ghi nhận thì nhận dạng khuôn mặt cũng còn gặp rất nhiều khó khăn như về độ sáng, hướng nghiêng, kích thước hình ảnh, diện mạo, biểu hiện cảm xúc của khuôn mặt ảnh hưởng của tham số môi trường.

Để xây dựng một hệ thống nhận dạng khuôn mặt có đầu vào là một hình ảnh kỹ thuật số hay một khung video từ một đoạn video, clip, … Đầu ra là xác định hoặc xác minh xem người ở trong bức hình đó hoặc người ở trong đoạn video clip đó là ai. Hướng tới mục tiêu đó nhận dạng khuôn mặt thường được chia làm 3 bước: phát hiện khuôn mặt, trích rút đặc trưng và nhận dạng khuôn mặt.

Face Recognition

Feature Extraction

Face Detection

Image

Phát hiện khuôn mặt (Face Detection): chức năng chính của bước này là phát hiện khuôn mặt xem khuôn mặt đó có xuất hiện trong bước hình hay đoạn video hay không. Tỉ lệ phát hiện khuôn mặt phụ thuộc nhiều vào khuôn mặt như điều kiện về độ sáng, hướng nghiêng của khuôn mặt, biểu hiện cảm xúc trên khuôn mặt hay các yếu tố môi trường khác. Để hệ thống nhận dạng hoạt động đạt hiểu quả cao thì hình ảnh khuôn mặt sau khi được phát hiện cần chuẩn hoá về kích thước hình ảnh, độ sáng.

Trích rút đặc trưng (Feature Extraction): sau khi phát hiện được khuôn mặt trong bức ảnh chúng ta tiến hành trích rút những đặc trưng của khuôn mặt. Bước này sẽ trích rút ra một vector đặc trưng đại diện cho một khuôn mặt. Nó phải đảm bảo được tính duy nhất của khuôn mặt.

Nhận dạng khuôn mặt (Face Recognition): với hình ảnh đầu vào và sau khi đã phát hiện khuôn mặt, trích rút các đặc trưng của khuôn mặt, rút trích xuất các đặc trưng của khuôn mặt và đem so sánh các đặc trưng này với cơ sở dữ liệu khuôn mặt.

Bài toán nhận dạng khuôn mặt được ứng dụng trong nhiều lĩnh vực đời sống đặc biệt ở những lĩnh vực công nghệ cao, yêu cầu về an ninh, bảo mật. Do đó để hệ thống nhận diện khuôn mặt hoạt động mạnh mẽ với tốc độ và độ tin cậy thì có rất nhiều các phương pháp về nhận dạng khuôn mặt được đưa ra. Trong đó có phương pháp nhận dạng dựa vào tiêu chí đầu vào với dữ liệu là ảnh tĩnh 2D, và phổ biến hơn là đầu vào là ảnh 3D, …

## **3.1.3. Ứng dụng cho hệ thống nhận dạng khuôn mặt**

xác định các nhân (one to one mathing): khi đưa ra một khuôn mặt của cá nhân và yêu cầu nêu ra danh tính của họ, đòi hỏi hệ thống trả về kết quả họ là ai.

xác đinh (one to many mathing): khi đưa ra hình ảnh của một cá nhân không rõ danh tính, xác định danh tính người đó bằng cách có thể so sánh với một cơ sở dữ liệu hình ảnh của các cá nhân đã biết.

Có rất nhiều lĩnh vực ứng dụng của hệ thống nhận dạng khuôn mặt, trong đó nhận dạng khuôn mặt có thể được khai thác cho hai mục đích trên, một vài ứng dụng tiêu biểu được triển khai ở một số quốc gia như:

**Các ứng dụng chuyên biệt cho ngành hàng không**:

Thời gian gần đây, vấn đề an toàn hàng không đang được cả thế giới quan tâm là một đề tài nóng bỏng của cả xã hội. Có rất nhiều vụ tấn công khủng bố được thực hiện đối với ngành hàng không do việc đảm bảo an ninh, an toàn hàng không có những thiếu xót. Một số nước trên thế giới đã ứng dụng công nghệ nhận dạng khuôn mặt trong lĩnh vực hàng không như:

Một chương trình tên US visits (United States Visiors and Immigrant Status Indicator Techology) áp dụng cho công dân du lịch tới Mỹ hoặc xin nhập cư vào Mỹ. chương trình này được bắt đầu thực hiện từ tháng 5 năm 2004 tại các lãnh sự quán của Mỹ ở nước ngoài. Khi một công dânn đến lãnh sự quán xin cấp visa thì họ sẽ thu thập được thông tin sinh trắc học, ảnh khuôn mặt để kiểm tra đối chiếu với một cơ sở dữ liệu của bọn tội phạm nổi tiếng và những nghi can khủng bố. Nếu quá trình kiểm tra đối chiếu được thông qua thì công dân này được cấp visa tới Mỹ. Khi khách tới cửa khẩu, sân bay thì các thông tin sinh trắc học của họ sẽ được sử dụng để xác minh xem họ có phải chủ sở hữu của visa không, có phải là tội phạm nguy hiểm hay nghi can khủng bố hay không. Chương trình US visits tăng cường sự an toàn của công dân Mỹ và du khách bằng xã minh nhận dạng của khách có visa, đồng thời nó tạo điều kiện đi lại và thương mai hợp pháp bằng cách tận dụng công nghệ và phát triển sử dụng sinh trắc học để tiến hành giám sát ngay tại đường biên giới trước khi nhận cảnh về Mỹ.

Một chương trình khác cũng được sử dụng tại Úc là SmartGate. Nó được thử nghiệm bởi hải quan sân bay quốc tế Sydney từ năm 2002. Mục tiêu chính của thử nghiệm này là phát triển và giới thiệu một hệ thống tự xử lý sử dụng công nghệ sinh trắc học cho công dân nhập cảnh, Mất khoảng 17 giây để SmartGate thực hiện so sánh một hình ảnh thực của công dân được hệ thống tự động chụp lại tại điểm giao dịch SmartGate với một hoặc nhiều hình ảnh của họ đã được lưu trữ. Nếu những bức ảnh này cho kết quả là trùng khớp thì họ được chấp nhận nhập cảnh vào Úc.

|  |  |
| --- | --- |
|  |  |

Hình 3.2: Hệ thống xác minh nhân thân của công dân nhập cảnh

**Nhận dạng khuôn mặt được sử dụng kèm với thẻ truy cập:** Tại các nước phát triển, hầu như mọi người dân đều biết sử dụng thẻ tín dụng để mua bán, rút tiền, trao đổi hàng hoá. Trong tương lai gần, Việt Nam chúng ta có lẽ cũng được áp dụng hình thức thanh toán này. Điều này rất nguy hiểm khi thẻ truy cập này nhặt được hay biết được mật khẩu của người chủ sở hữu thẻ này. Để đảm bảo an toàn hơn cho việc phổ cập sử dụng thẻ trong toàn xã hội là chúng ta hãy dùng thêm một phương pháp nữa để xác minh tính xác thực của người sử dụng thẻ song song với mật khẩu, đó là sử dụng khuôn mặt như là một mật khẩu thứ hai để truy cập vào hệ thống cùng với thông tin từ thẻ truy cập. Để rút được tiền, người dùng cần thực hiện các bước sau:

* Đưa thẻ vào hệ thống.
* Đứng trước camera để nhận dạng.
* Xác minh người này có phải chủ sở hữu của thẻ hay không.

Nếu thông tin về người sử dụng thẻ khớp với thông tin của chủ thẻ thì cho thực hiện việc rút tiền ngược lại thì không cho rút tiền và tự động báo tới nhà chức trách.

**Lần dâu vết đi tìm kẻ tội phạm:** Từ những bức ảnh hay những đoạn video đã được ghi lại tự động tại hiện trường trước khi vụ khủng bố xảy ra, lực lượng an ninh có thể tìm được những kẻ có khả nghi xuất hiện trong những bức ảnh hay đoạn video này. Nếu trong cơ sở dữ liệu của hệ thống có thông tin về những người xuất hiện trong đó, chúng ta có thể dễ dàng hơn trong biệc tìm ra họ vì chúng ta biết họ là ai.

Tháng 2 năm 2012 FBI trển khai trương trình NGI Facial Recognition Pilot là tiến hình tìm kiếm nhận dạng khuôn mặt dựa trên hình ảnh của kho lưu trữ quốc gia của FBI (có khoảng 12,8 triệu bức ảnh được lưu trữ) và cung cấp danh sách những đối tượng khả nghi cho cơ quan điều tra. ảnh truy vấn là các hình ảnh được thu thập tại hiện trường, từ các trang mạng và yêu cầu truy vấn được hệ thống xử lý kín (không có sự can thiệp của con người) và kết quả trả về cho cơ quan điều tra như là một gợi ý về đối tượng tình nghi.

**Hệ thống giám sát công nhân và chấm công tự động:**

Hiện nay trong các khu công nghiệp hay các công ty sản xuất lớn có hàng ngàn công nhân ra vào mỗi ngày nên việc giám sát kẻ gian vào công ty cũng như công việc chấm công rất phức tạp. Vậy làm thế nào để nhận ra từng nhân viên của công ty? Người ta xây dựng hệ thống có thể nhận dạng và chấm công tự động cho công nhân khi họ đi qua khu vực làm việc của hệ thống chấm công tự động.

# **3.2. Kỹ thuật nhận dạng khuôn mặt**

Xét về nguyên tắc chung, nhận dạng khuôn mặt có các vấn đề liên quan như sau:

Vấn đề 1: Để nhận dạng được khuôn mặt thì đầu vào của chúng ta là ảnh chứa các mặt người, khi ấy bước đầu tiên trong nhận dạng khuôn mặt là hãy nhìn vào bức ảnh đó và tìm ra tất cả các khuôn mặt có trong hình ảnh đó.

Vấn đề 2: Hãy tập trung vào từng khuôn mặt và có thể hiểu rằng ngay cả khi khuôn mặt bị quay theo một hướng nào khác hoặc điều kiện môi trường như điều kiện ánh sáng xấu thì vẫn phải nhận dạng được đó chính là người đó.

Vấn đề 3: khi nhìn vào một khuôn mặt thì hãy chọn ra các điểm đặc trưng trên khuôn mặt đó mà nhờ điểm đặc trưng đó mà người ta có thể sử dụng nó để phân biệt với người khác. Ví dụ như mắt to như nào, khuôn mặt dài bao nhiêu, …

vấn đề 4: so sánh điểm đặc trưng đó với tất cả các khuôn mặt đã được biết để tìm ra tên của người cần được nhận dạng.

Là một con người, bộ não rất phát triển thì tất cả các vấn đề này được giải quyết một cách rất nhanh chóng, tự động. Trên thực tế, thì máy tính không có khả năng tổng quát đến mức độ cao này. Vì vậy mà con người cần phải dạy cho máy tính từng bước trong quy trình để nhận dạng được ra một người mà tên nhận dạng được chính xác là người đó.

Ý tưởng cơ bản cho bài toán được trình bày như sau: Từ một hay nhiều hình ảnh đầu vào, mỗi hình ảnh chứa một khuôn mặt. Sau đó tiến hành phát hiện (face detection) và xác định vị trí các khuôn mặt có trong bức ảnh (nếu có). Các bức ảnh ảnh sau khi được detection face sẽ được đưa vào cho mô hình tự động học và trích rút ra các đặc trưng quan trọng đại diện cho mỗi khuôn mặt. Cuối cùng, các mô hình SVM sẽ đợc áp dụng để phân lớp (nhận dạng) khuôn mặt đầu vào dựa trên các đặc trưng đã được trích rút.

## **3.2.1. Finding all the Faces**

Phương pháp phát hiện khuôn mặt ở đây được đề xuất sử dụng các đặc trưng HOG. Và bộ phân lớp tuyến tính SVM.

Vậy HOG (histogram of oriented gradients) là gì? Nó là một feature descriptor được sử dụng trong computer vision và xử lý ảnh. Dùng để detect một đối tượng. Các khái niệm Hog được nêu ra từ năm 1986 tuy nhiên cho đến năm 2005 HOG mới được sử dụng rộng rãi sau khi Navneet Dalal và Bill Triggs công bố những bổ sung về HOG.

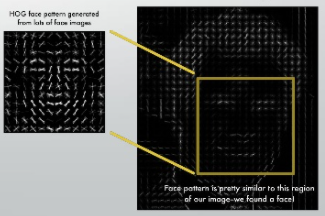
HOG tương tự như các biểu đồ edge orientation, scale-invariant, … nhưng HOG được tính toán trên một lưới dày đặt các cell và chuẩn hoá sự tương phản giữa các block để nâng cao độ chính xác. HOG được sử dụng chủ yếu để mô hình nhận dạng.

HOG thể hiện được sự thay đổi về sự sáng tối trong bức ảnh. Tại các nét trên khuôn mặt có sự thay đổi sáng tối rõ rệt. HOG thể hiện được đưofng nét của khuôn mặt.

|  |  |
| --- | --- |
|  |  |

Hình 3.3: Hình ảnh gốc được chuyển thành hình ảnh đại diện HOG

Sau khi huấn luyện với nhiều loại khuôn mặt sẽ thu được pattern của khuôn mặt. sau đó so sánh pattern với các vùng trong bức ảnh để phát hiện mặt người



Hình 3.4: Tìm phần của ảnh giống nhất với mẫu HOG đã biết

Và kết quả cuối cùng là thu được một bức ảnh đã được detect như hình ảnh phía dưới: khuôn mặt được detect sẽ được thể hiện bằng hình chữ nhật có các cạnh màu đỏ xung quanh.

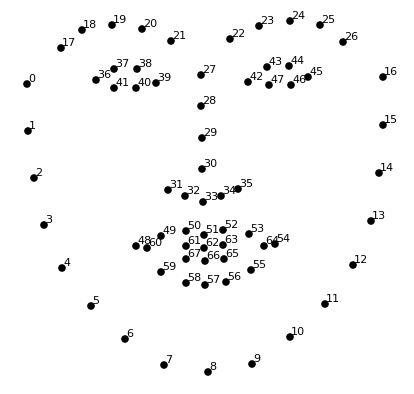
|  |  |
| --- | --- |
|  |  |

Hình 3.5: kết quả detect face

## **3.2.2. Posing and Projecting Faces**

Để giải quyết vấn đề 2 đã được nêu ở phía trên: giải quyết vấn đề nếu như khuôn mặt quay theo hướng khác nhau thì vẫn phải xác định được là người đó. Và để giải quyết vấn đề này chúng tôi sử dụng thuật toán face landmark estimation.

Ý tưởng cơ bản là chúng ta sẽ đưa ra 68 điểm cụ thể gọi là điểm mốc tồn tại trên khuôn mặt như cằm, mắt, mép trong của lông mày, … sau đó chúng ta sẽ đào tào khiến cho máy có thể tìm thấy được 68 điểm cụ thể này trên bất kỳ khuôn mặt nào.



Hình 3.6: 68 điểm trên khuôn mặt

Nó sẽ detection 62 điểm trên khuôn mặt cụ thể như sau:

"chin": points [0:17],

"left\_eyebrow": points [17:22],

"right\_eyebrow": points [22:27],

"nose\_bridge": points [27:31],

"nose\_tip": points [31:36],

"left\_eye": points [36:42],

"right\_eye": points [42:48],

"top\_lip": points [48:55] + [points [64]] + [points [63]] + [points [62]] + [points [61]] + [points [60]],

"bottom\_lip": points [54:60] + [points [48]] + [points [60]] + [points [67]] + [points [66]] + [points [65]] + [points [64]]

Và kết quả khi sử dụng thuật toán face landmark estimation chúng ta thu được kết quả như sau:

|  |  |
| --- | --- |
|  |  |

Hình 3.7: Xác định mốc mặt cho khuôn mặt

## **3.2.3. Encoding Faces**

Phương pháp trích chọn đặc trưng ở đây được đề xuất dựa trên ý tưởng xây dựng các mạng nơ-ron học sâu FaceNet được Florian Schroff và cộng sự đề xuất năm 2015.đây là mô hình có khả năng học từ tập mẫ thu thập từ các môi trường thực tế để tự động phát hiện ra các đặc trưng quan trọng nhất để nhận dạng đối tượng.

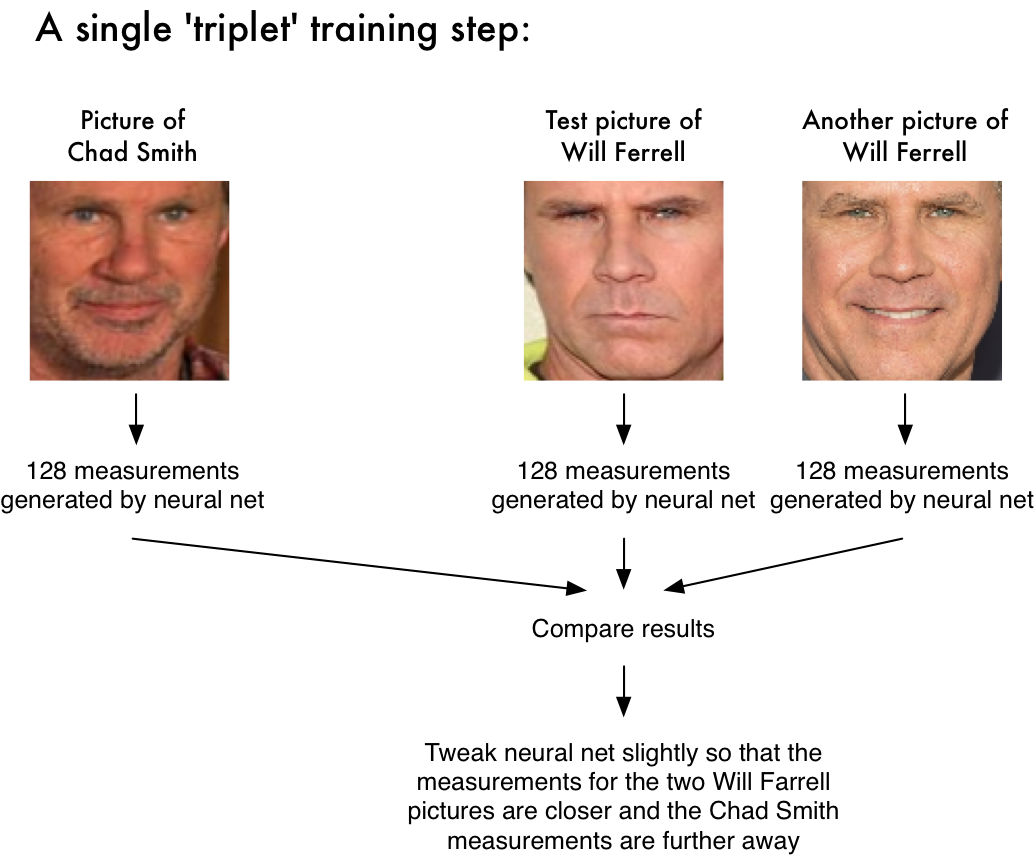
Ý tưởng chính của phương pháp dựa trên việc học một không gian Euclidean nhúng trong mối hình ảnh hay hiểu một cách đơn giản nhất để nhận dạng khuôn mặt là so sánh trực tiếp các khuôn mặt không xác định tức là ảnh input đầu vào cần được nhận diện mà chúng ta đã tìm được. Kiểm tra xem khuôn mặt được đã được xác định với khuôn mặt chưa được xác định có cùng là một người hay không.

Những gì chúng ta cần là trích xuất ra một vector từ mỗi khuôn mặt, sau đó chúng ta có thể đo khuôn khuôn mặt chưa biết theo một cách tương tự và tìm kiếm khuôn mặt với số đo gần nhất. ví dụ là chúng ta có thể đo kích thước của tau, khoảng cách giữa 2 mắt, chiều dài của mũi, … và cách tiếp cận nhanh nhất là chúng ta sẽ huấn luyện nó để tạo ra một vector có 128 chiều cho mỗi khuôn mặt.

Quá trình đào tạo hoạt động bằng cách nhìn vào 3 hình ảnh khuôn mặt cùng một lúc.

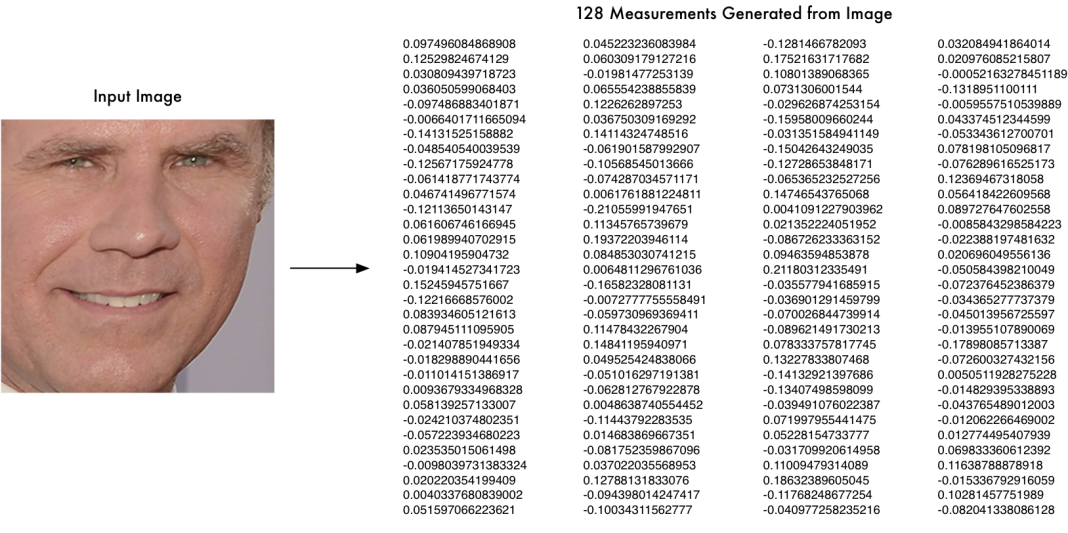
* Hình ảnh khuôn mặt đào tạo của một người đã biết.
* Hình ảnh khác của cùng một người đã biết.
* Hình ảnh của một người hoàn toàn khác.

Sau đó thuật toán xem xét các phép đo mà nó hiện đang đào tạo cho từng hình ảnh trong 3 hình ảnh đó.



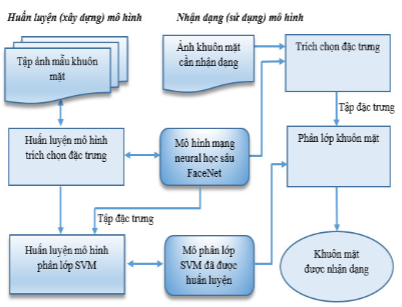
Hình 3.8: Huấn luyện mạng nơ ron tự động trích rút đặc trưng

Mạng được huấn luyện một cách trực tiếp để đầu ra của nó trở thành một vector đặc trưng 128 chiều sử dụng hàm chí phí bộ ba (tripletbased loss function). Một bộ ba (triplet) được định nghĩa bao gồm 2 khuôn mặt của cùng một người (positive) và một khuôn mặt của người khác- negative. Mục tiêu chính của hàm chi phí là phân tách cặp khuoon mặt pisitive ra khỏi khuôn mặt negative sử dụng một lề khoảng cách distance margin. Từ các độ đo thu được thuật toán sẽ ước lượng giá trị các hàm chi phí dựa trên việc so sánh khoảng cách giữa 2 tập đặc trưng trong đó có 2 tập đặc trung được sinh ra từ 2 ảnh khuôn mặt khác nhau của một người (người thứ nhất) và tập đặc trưng thứ 3 được sinh ra từ ảnh khuôn mặt của người khác (người thứ hai)



Hình 3.9: Kết quả của việc trích chọn đặc trưng khuôn mặt

## **3.2.4. Finding the person’s name from the encoding**



Hình 3.10: Phân lớp khuôn mặt người

Bước cuối cùng thực sự đơn giản. điều cần làm ở bước này là tìm người trong cơ sở dữ liệu về những người đã biết có số đo gần nhất với hình ảnh thử nghiệm. Đối với bài toán này, thuật toán phân loại học máy được sử dụng là bộ phân loại SVM tuyến tính. Tất cả điều cần làm là đào tạo ra bộ phân loại có thể thực hiện các phép đo từ hình ảnh thử nghiệm mới và cho biết người đó phù hợp với đối tượng nào đã biết.

# **3.3. Đánh giá thực nghiệm**

## **3.3.1. Độ đo đánh giá hiệu quả**

Quá trình đánh giá thực nghiệm được chia thành hai công đoạn: đánh giá hiệu quả của mô hình phát hiện khuôn mặt và đánh giá độ chính xác nhận dạng trên số ảnh đã phát hiện được khuôn mặt (không tính độ chính xác cho các ảnh chưa phát hiện được khuôn mặt).

Hiệu quả của mô hình phát hiện khuôn mặt được đánh giá dựa trên các độ đo tự được định nghĩa cụ thể như sau:

* **Độ đo chính xác phát hiện khuôn mặt**: được tính bằng số khuôn mặt phát hiện đúng trên tổng số ảnh cần phát hiện.
* **Độ đo nhận dạng chính xác:** được tính bằng số người nhận dạng đúng trên tổng số người được phát hiện khuôn mặt.

## **3.3.2. Môi tường thực nghiệm**



Hình 3.11: Ngôn ngữ sử dụng python

Chương trình thực nghiệm được cài đặt trong môi trường python. Python là một ngôn ngữ lập trình bậc cao cho các mục đích lập trình đa năng, do Guido van Rossum tạo ra và lần đầu được ra mắt vào năm 1991. Python được thiết kế với ưu điểm mạnh là dễ đọc, dễ học và dễ nhớ. Python là ngôn ngữ có hình thức rất sáng sủa, cấu trúc rõ ràng, rất thuận tiện cho người mới học lập trình. Python còn cho phép người dùng sử dụng viết mã lệnh với số lần gõ phím tối thiểu. Tuy nhiên, python có tốc độ thực hiện chậm hơn nhiều lần so với ngôn ngữ biên dịch khác như C, …

Sử dụng các thư viện numpy cho việc biểu diễn, lưu trữ thao tác dự liệ, thư viện sklearn cho việc thử nghiệp mô hình học máy cụ thể là mô hình SVM.

Ngoài ra, còn sử dụng thư viện hỗ trợ cho bài toán nhận diện khuôn mặt là face\_recognition. Bài toán sử dụng các thư viện có sẵn trong thư viện như face\_encoding để trích chọn đặc trưng cho khuôn mặt tạo ra vector đặc trưng có 128 chiều, …

**Cài đặt thư viện face\_recognition:** (chỉ hướng dẫn trên hệ điều hành windows):

* Bước 1: download và install 2 thư viện scipy và numpy+mkl phù hợp với phiên bản python theo link sau: [**https://www.lfd.uci.edu/~gohlke/pythonlibs/**](https://www.lfd.uci.edu/~gohlke/pythonlibs/)**.**
* Bước 2: Download Boost mã nguồn mở hoặc bản phát hành nhị phân cho MSVC hiện tại tại đường link: [**https://sourceforge.net/projects/boost/files/**](https://sourceforge.net/projects/boost/files/)**.**
* Bước 3: Nếu tải bản nhị phân, hãy bỏ qua bước này ngược lại làm theo các bước sau để tự biên dịch và xây dựng Boost.
  + Bước 3.1: Hãy giải nén thư mục Boost vừa tải xuống ở bước 2.
  + Bước 3.2: Tiếp theo tạo một biến hệ thống với các tham số sau:

Name: VS140COMNTOOLS

Value: C:\Program Files(x86) \Microsoft Visual Studio\2017\Community\Common7\ Tools\

* Bước 3.3: Open Developer Command Prompt for Visual Studio với quyền admin rồi chạy lệnh boostrap.bat tại thư mục Boost vừa giải nén để sinh ra file b2. exe. Sau đó chạy tiếp lệnh b2. exe -a --with-python address-model=64 toolset=msvc runtime-link=static. Khi chạy xong nó sẽ sinh ra các tệp đã biên dịch trong thư mục stage.
* Bước 4: Nếu bạn đã biên dịch Boost, bỏ qua bước này. Nếu tải xuống bản nhị phân chỉ cần giải nén nó.
* Bước 5: Tải dlib phiên bản mới nhất từ: <https://github.com/davisking/dlib>.
* Bước 6: Đi tới thư mục dlib vừa tải mở command line để xây dựng dlib

set BOOST\_ROOT = đường dẫn đến thư mục Boost đã giải nén ở bước 2.

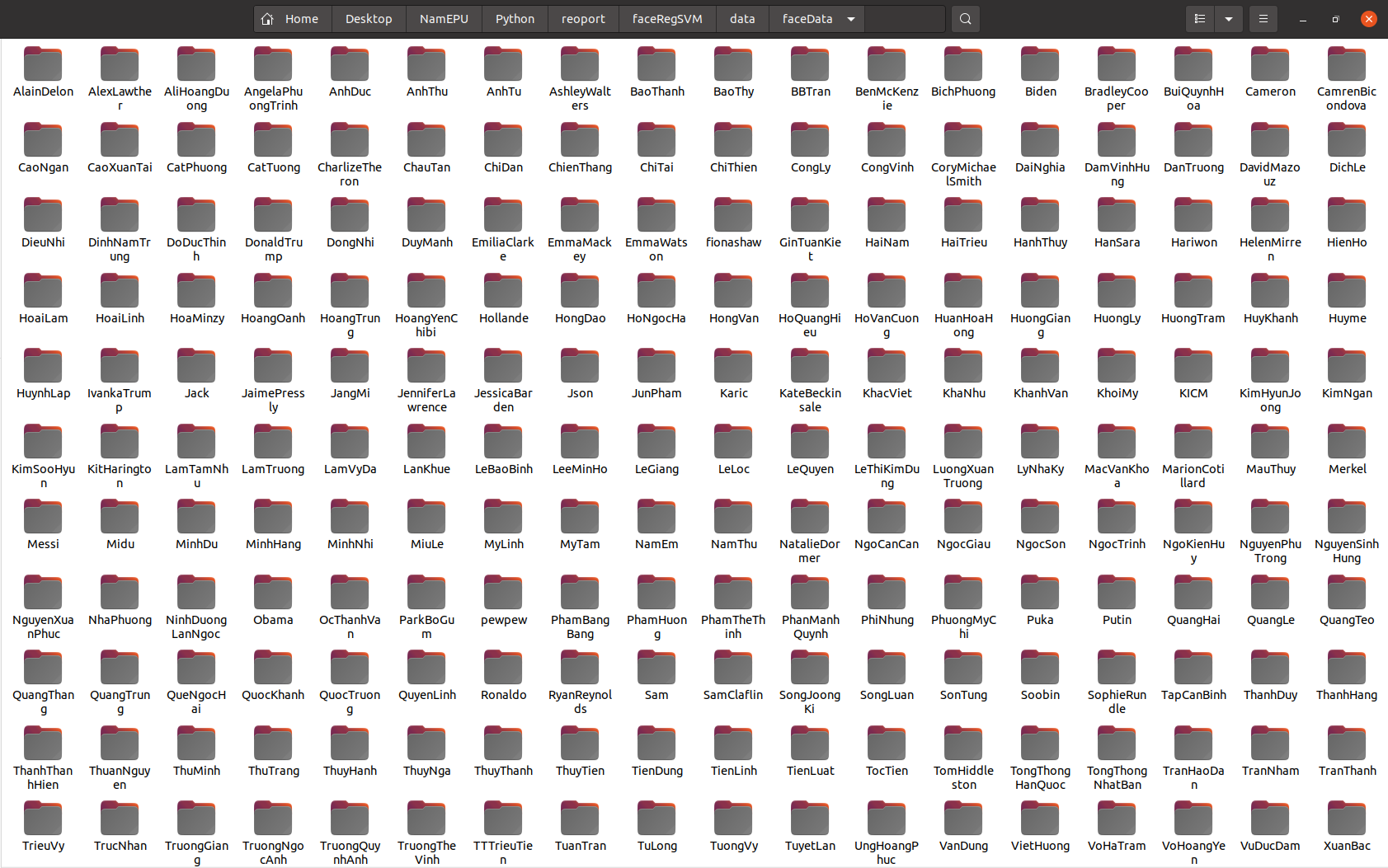
set BOOST\_LIBRARYDIR = đường dẫn tới thư mục stage\lib đã build được bước 3.

python setup.py install

* Bước 7: bây giờ kiểm tra phiên bản hiện tại của dlib bằng lệnh pip show dlib để xem dlib đã cài đặt thành công hay chưa.
* Bước 8: thực hiện cài đặt thư viện face\_recognition bằng lệnh pip install face\_recognition.

## **3.3.3. Dữ liệu thực nghiệm**

Hiệu quả của mô hình nhận dạng được đánh giá trên bộ dữ liệu tự tìm kiếm. Bộ dữ liệu gồm 6202 khuôn mặt được thu thập từ web. Mỗi khuôn mặt được gán nhãn với tên của người đó.



Hình 3.12: Tập dữ liệu thực nghiệm

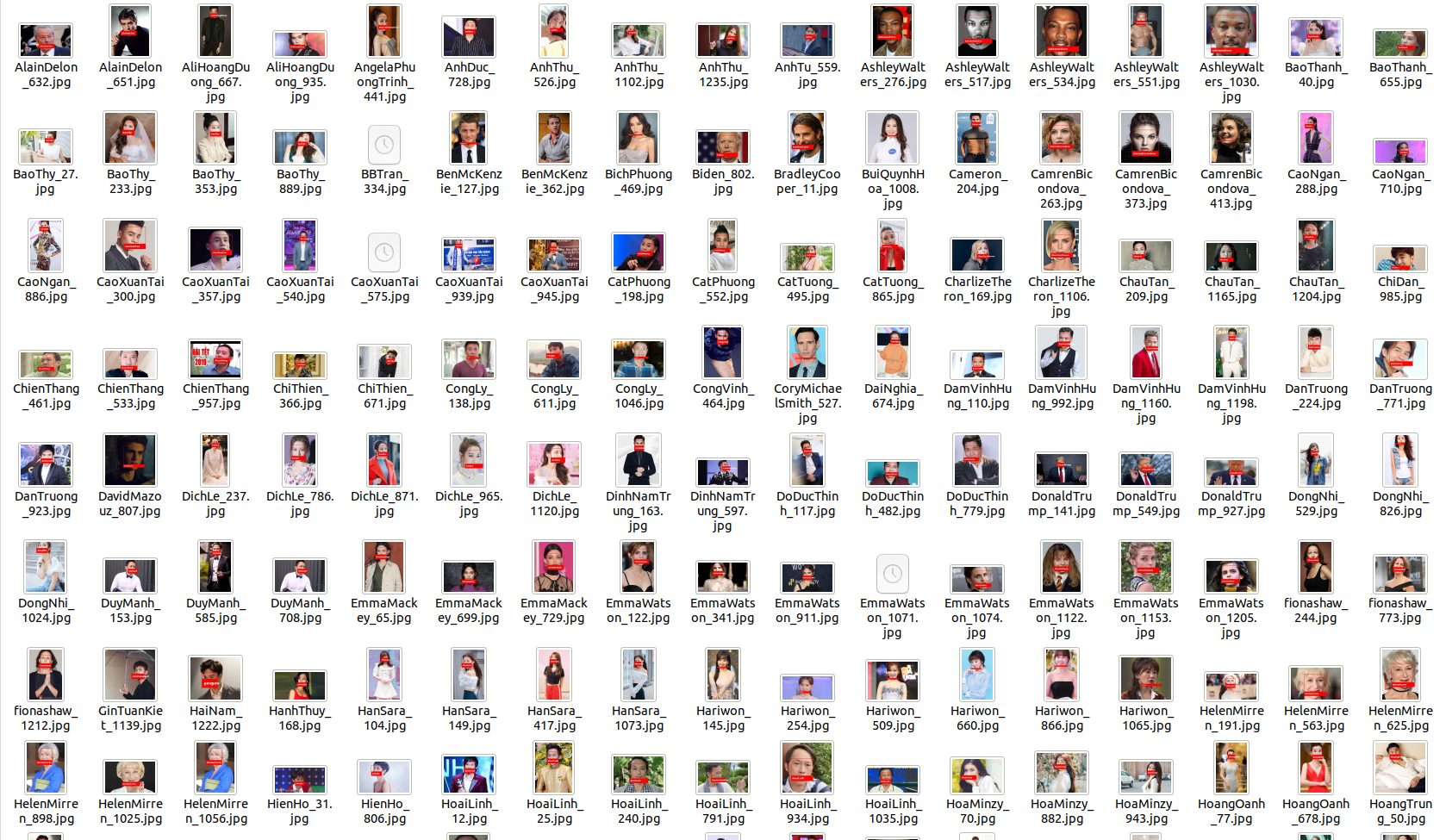
# **3.4. Kết quả thực nghiệm**

Tập dữ liệu được chia ngẫu nhiên bằng thư viện train\_test\_split của sklearn theo tỷ lệ 80/20 (80% số mẫu để huấn luyện mô hình và 20% số mẫu còn lại để kiểm thử).

## **3.4.1. Độ chính xác**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Dữ liệu thử nghiệm** | **Số mẫu** | **Số ảnh train** | **Số ảnh test** | **Độ chính xác detect face** | **Độ chính xác nhận dạng** |
| DataSetFace | 6202 | 4961 | 1241 | 98,15 % | 98,03 % |

Bảng 3.1: Độ chính xác

Hình 3.13: Kết quả thực nghiệm

## **3.4.2. Thời gian**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Dữ liệu** | **Số mẫu** | **Số ảnh train** | **Số ảnh test** | **Thời gian detect 1 ảnh** | **Thời gian train 1 ảnh** |
| DataSetFace | 6202 | 4961 | 1241 | 0,11 (s) | 0,23 (s) |

Bảng 3.2: Thời gian thực hiện detect một ảnh và train một ảnh

## **3.4.3. Giao diện thể hiện kết quả thực nghiệm**

****

Hình 3.14: Kết quả thử nghiệm với ảnh có một người

****

Hình 3.15: Kết quả thử nghiệm với ảnh có hai người

**KẾT LUẬN**

Qua việc thực hiện nghiên cứu đề tài “**Nhận dạng khuôn mặt sử dụng thuật toán SVM**”, nhóm chúng em đã được biết thêm rất nhiều kiến thức vềthuật toán cũng như bước đầu nắm bắt được ứng dụng của môn học “**Ngôn ngữ lập trình Python**”.

Thông qua thuât toán này đã giúp cho chúng ta tìm kiếm được dữ liệu và cách triển khai để sử dụng nó. Đồng thời, từ thuật toán này ta có thể ứng dụng bài toán vào các cách phân tích, dự báo, …. Kèm theo đó là áp dụng thêm các kiến thức được học trên trường lớp và kiến thức học hỏi được để có thể tạo ra một ứng dụng tốt hơn về mọi mặt trong thực tế.

Bên cạnh đó, việc làm nghiên cứu giúp chúng em đoàn kết hơn, rèn luyện cho chúng em kỹ năng làm việc nhóm. Trong quá trình thực hiện đề tài có rất nhiều ý tưởng hay và độc đáo. Nhưng do kiến thức của chúng em còn hạn hẹp và thời gian không cho phép nên chúng em chưa thể thực hiện được những ý tưởng đó. Tuy nhiên chúng em đã cố gắng để xây dựng một chương trình hoàn chỉnh và đẹp nhất để đưa tới Thầy(cô). Trong quá trình xây dựng chương trình nhóm chúng em khó tránh khỏi những sai xót còn tồn tại. Vì vậy chúng em rất mong được nhận lời góp ý và chỉnh sửa từ Thầy(cô) để có thể hoàn thành chương trình một cách hoàn chỉnh nhất.

Chúng em một lần nữa xin cảm ơn cô giáo **Phạm Thị Kim Dung** đã tận tình giảng dạy cũng như hướng dẫn chúng em làm sản phẩm kết thúc học phần trong môn học, cô đã giúp đỡ chúng em trong quá trình nghiên cứu đề tài và chia sẻ những tài liệu hay cũng như các kỹ năng lập trình cần thiết.

# **TÀI LIỆU THAM KHẢO**.

[1]. Vũ Hữu Tiệp; Machine Learning cơ bản; Last update: March8, 2018.

[2]. Nguyễn Thanh Tuấn; Deep Learning cơ bản; Last update: October 2019.

[3] <https://en.wikipedia.org/wiki/Support-vector_machine>

[4] Nâng cao độ chính xác nhận dạng khuôn mặt dựa trên mô hình CNN học sâu kết hợp với đặc trưng HOG và bộ phân lớp SVM – Nguyễn Thị Thanh Tân.