TRƯỜNG ĐẠI HỌC VĂN LANG

KHOA CÔNG NGHỆ THÔNG TIN



TIỂU LUẬN

NHẬP MÔN XỬ LÝ ẢNH

ĐỀ TÀI : NGHIÊN CỨU XÂY DỰNG ỨNG DỤNG XỬ LÝ HÌNH ẢNH VÀ NHẬN DIỆN KHUÔN MẶT

|  |  |
| --- | --- |
| ****Giảng viên hướng dẫn:****  Sinh viên thực hiện: | TS Đỗ Hữu Quân  **1. Trần Minh Tâm - 2174802010640**  **2. Trần Văn Quốc Thiện – 2174802010618**  **3. Trần Văn Tú -**2174802010126  4. Vũ Hoàng Minh Khoa -2174902010319  5. Chế Ngọc Linh -2174802010363 |

TP. Hồ Chí Minh – 2025

**LỜI CẢM ƠN**

Lời nói đầu tiên nhóm chúng em xin cảm ơn sâu sắc đến Ban giám hiệu trường Đại học Văn Lang, quý thầy cô trong khoa Công Nghệ Thông Tin đặc biệt là thầy Đỗ Hữu Quân về sự hỗ trợ và giúp đỡ trong quá trình thực hiện báo cáo cuối kì đề tài về xử lý ảnh.

Quý thầy đã cung cấp cho chúng em kiến thức và kinh nghiệm quý giá trong việc tìm hiểu và xử lý hình ảnh trước khi được sử dụng cho các mục đích sau.

Với sự hỗ trợ và đóng góp quý báu của quý thầy, chúng em một chương trình giúp nhận diện khuôn mặt. Chương trình này cũng được thiết kế để người dùng có thể sử dụng như một công cụ lưu trữ hình ảnh đồng thời giúp máy tính học các khuôn mặt đã được lưu trữ, phục vụ mục đích nhận diện.

Chúng em rất biết ơn sự giúp đỡ của hai thầy và mong rằng những kiến thức và kinh nghiệm quý báu của quý thầy sẽ tiếp tục hỗ trợ và giúp đỡ chúng em trong các hoạt động nghiên cứu và phát triển tiếp theo.

Trân trọng cảm ơn.

**MỤC LỤC**

[**BẢNG PHÂN CÔNG NHIỆM VỤ THỰC HIỆN ĐỐI VỚI ĐỀ TÀI 4**](#_Toc32004)

[**LỜI NÓI ĐẦU 5**](#_Toc32437)

[**Chương 1 Giới thiệu tổng quan 6**](#_Toc20046)

[1.1. Tổng hợp về nhận diện khuôn mặt 6](#_Toc5175)

[1.2. Phân tích bài toán 6](#_Toc4370)

[1.3. Các công cụ 6](#_Toc10415)

[1.4 Phương pháp giải quyết 9](#_Toc14747)

[1.5 Cấu trúc báo cáo 10](#_Toc5470)

[**CHƯƠNG 2 XỬ LÝ ẢNH 11**](#_Toc21156)

[2.1 Giới thiệu về hệ thống xử lí ảnh: 11](#_Toc10838)

[2.2 Những vấn đề cơ bản trong hệ thống xử lý ảnh: 13](#_Toc3601)

[2.3 Những vấn đề khác: 16](#_Toc19346)

[**CHƯƠNG 3 THUẬT TOÁN NHẬN DẠNG 17**](#_Toc15426)

[3.1 Phát hiện khuôn mặt 17](#_Toc21452)

[3.2 Đổi góc độ và chiều cho khuôn mặt 21](#_Toc8895)

[3.3 Mã hóa khuôn mặt 22](#_Toc24514)

[**PHẦN CODE DEMO 24**](#_Toc2735)

[**TÀI LIỆU THAM KHẢO 29**](#_Toc28883)

**BẢNG PHÂN CÔNG NHIỆM VỤ THỰC HIỆN ĐỐI VỚI ĐỀ TÀI**

|  |  |  |
| --- | --- | --- |
| **TT** | **Họ và tên** | **Nội dung thực hiện** |
| 1 | Trần Minh Tâm | Thực hiện nội dung báo cáo |
| 2 | Trần Văn Quốc Thiện | Viết chương trình code main |
| 3 | Trần Văn Tú | Viết chương trình code train |
| 4 | Vũ Hoàng Đăng Khoa | Thiết kế PowerPoint |
| 5 | Chế Ngọc Linh | Thiết kế PowerPoint |

**LỜI NÓI ĐẦU**

Khuôn mặt đóng vai trò quan trọng trong quá trình giao tiếp giữa con người với con người , và mang một lượng thông tin giàu có , chẳng hạn có thể xác định giới tính ,tuổi tác ,ngoại hình , trạng thái cảm xúc của người đó,....hơn nữa khảo sát chuyển động các đường nét khuôn mặt có thể biết được người đó đang muốn nói gì .Trong hệ thống nhận dạng khuôn mặt được đánh giá là bước khó khăn và quan trong nhất so với các bước còn lại của hệ thống . Do đó, nhận diện được khuôn mặt là điều quan trọng và thiết yếu được đề cập đến.

Nhận dạng khuôn mặt người là một công nghệ được ứng dụng rộng rãi trong đời sống hàng ngày của con người như các hệ thống giám sát, quản lí vào ra, tìm kiếm thông tin người nổi tiếng.... có rất nhiều phương pháp nhận dạng khuôn mặt để nâng cao hiệu suất tuy nhiên dù ít hay nhiều những phương pháp này đang vấp phải những thử thách về độ sáng, hướng nghiên, kích thước ảnh, hay ảnh hưởng của tham số môi trường.

Cùng với sự phát triển của xã hội, vấn đề an ninh, bảo mật đang được yêu cầu khắt khe tại mọi quốc gia trên thế giới. Các hệ thống nhận dạng con người, đồ vật... được ra đời và phát triển với độ tin cậy ngày càng cao. Với cách tiếp cận đối tượng nhận dạng theo phương pháp này, chúng ta có thể thu nhập được nhiều thông tin từ đối tượng hơn, mà không cần tác động nhiều đến đối tượng cũng vấn đảm bảo tính chính xác, an toàn, thuận tiện. Trong phạm vi bài báo cáo này chúng em xin được trình bài quá trình thực hiện điểm danh qua nhận diện khuôn mặt qua thư viện opencv. Cuối cùng, chủ đề của nhóm chúng em chọn đó chính là “Nhận diện khuôn mặt”.Vì chủ đề khá khó nên nội dung đồ án này không thể tránh khỏi những thiếu sót, rất mong được sự chỉ bảo , và góp ý của thầy cô và các bạn .

**Chương 1 Giới thiệu tổng quan**

**1.1. Tổng hợp về nhận diện khuôn mặt**

Hơn một thập kỉ qua có rất nhiều công trình nghiên cứu về bài toán xác định khuôn mặt người từ ảnh đen trắng, xám đến ảnh màu như ngày hôm nay. Các nhà nghiên cứu đi từ bài toán đơn giản, mỗi ảnh chỉ có một mặt người nhìn thẳng vào thiết bị thu hình và đầu ở tư thế thẳng đứng trong ảnh đen trắng. Cho đến ngày hôm nay bài toán mở rộng cho ảnh màu, có nhiều khuôn mặt trong cùng một ảnh, có nhiều tư thế thay đổi trong ảnh. Không những thế mà còn mở rộng cả phạm vi từ môi trường xung quanh khá đơn giản cho đến môi trường xung quanh rất phức tạp nhằm đáp ứng nhu cầu của con người.

Trong những năm gần đây các ứng dụng về trí tuệ nhân tạo ngày càng phát triển và được đánh giá cao. Một lĩnh vực đang được quan tâm của trí tuệ nhân tạo nhằm tạo ra các ứng dụng thông minh, có tính người đó là nhận dạng. Trong đề tài này em chọn đối tượng là khuôn mặt.

**1.2. Phân tích bài toán**

Bài toán Nhận Diện Khuôn mặt(Face Recognition) bao gồm các bài toán khác nhau như Phát hiện khuôn mặt (Face detection), đánh dấu(facial landmarking), trích chọn(rút) đặc trưng(feature extration), gán nhãn, phân lớp(classification).

**1.3. Các công cụ**

Để giải quyết bài toán khuôn mặt trong đồ án này chúng em sử dụng thư viện OpenCV và ngôn ngữ python trên phần mền Visual Studio Code

**1.3.1. Python**

Python là một ngôn ngữ lập trình thông dịch (interpreted), hướng đối tượng (object-oriented), và là một ngôn ngữ bậc cao (high-level) ngữ nghĩa động (dynamic semantics). Python hỗ trợ các module và gói (packages), khuyến khích chương trình module hóa và tái sử dụng mã. Trình thông dịch Python và thư viện chuẩn mở rộng có sẵn dưới dạng mã nguồn hoặc dạng nhị phân miễn phí cho tất cả các nền tảng chính và có thể được phân phối tự do.

Các đặc điểm của Python:

* Ngữ pháp đơn giản, dễ đọc.
* Vừa hướng thủ tục (procedural-oriented), vừa hướng đối tượng (object-oriented)
* Hỗ trợ module và hỗ trợ gói (package)
* Xử lý lỗi bằng ngoại lệ (Exception)
* Kiểu dữ liệu động ở mức cao.
* Có các bộ thư viện chuẩn và các model ngoài , đáp ứng tất cả các nhu cầu lập trình
* Có khả năng tương tác với các module khác viết trên C/C++(Hoặc Java cho Jython, hoặc .Net cho Iron Python).
* Có thể nhúng vào ứng dụng như một giao tiếp kịch bản (sripting interface)

**1.3.2. Thư viện Open CV**

Opencv (Open Computer Vision library) do Intel phát triển, được giới thiệu năm 1999 và hoàn thiện thành phiên bản 1.0 năm 2006. Thư viện opencv – gồm khoảng 500 hàm – được viết bằng ngôn ngữ lập trình C và tương thích với các hệ điều hành Windows, Linux, Mac OS... đóng vai trò xác lập chuẩn giao tiếp, dữ liệu, thuật toán cho lính vực CV và tọa điều kiện cho mọi người tham gia nghiên cứu và phát triển ứng dụng.

Trước Opencv không có một công cụ chuẩn nào cho lĩnh vực xử lí ảnh. Các đoạn code đơn lẻ do các nhà nghiên cứu tự viết thường không thống nhất và không ổn định. Các bộ công cụ thương mại như Matlab, Simulink,..v.v.. lại có giá cao chỉ thích hợp cho các công ty phát triển các ứng dụng lớn. Ngoài ra còn có các giải pháp kèm theo thiết bị phần cứng mà phần lớn là mã đóng và được thiết kế riêng cho tứng thiết bị, rất khó khan cho việc mở rộng ứng dụng. OpenCV là công cụ hữu ích cho những người bước đầu làm quen với xử lí ảnh số vì cá ưu điểm sau :

* OpenCV là công cụ chuyên dụng: được Intel phát triển theo hướng tối ưu hóa cho các ứng dụng xử lí và phân tích ảnh, với cấu trúc dữ liệu hợp lí, thư viện tạo giao diện, truy xuất thiết bị phần cứng được tích hợp sẵn. OpenCV thích hợp để phát triển nhanh ứng dụng
* OpenCV là công cụ mã nguồn mở: Không chỉ là công cụ miễn phí, việc được xây dựng trên mã nguồn mở giúp OpenCV trở thành công cụ thích hợp cho nghiên cứu và phát triển, với khả năng thay đổi và mở rộng các mô hình, thuật toán
* OpenCV đã được sử dụng rộng rãi: Từ năm 1999 đến nay, OpenCV đã thu hút được một lượng lớn người dung, trong đó có các công ty lớn như Microsoft, IBM, Sony, Siemens, Google và các nhóm nghiên cứu ở Standford, MIT, CMU, Cambridge,... Nhiều forum hỗ trợ và cộng đồng người dung đã được thành lập, tạo nên kênh thông 4 tin rộng lớn, hữu ích cho việc tham khảo, tra cứu

**1.3.3 Visual Studio Code**

Visual Studio Code là một trình biên tập mã được phát triển bởi Microsoft dành cho Windows, Linux và macOS. Nó hỗ trợ chức năng debug, đi kèm với Git, có syntax highlighting, tự hoàn thành mã thông minh, snippets, và cải tiến mã nguồn. Visual Studio Code là một trình biên tập mã. Nó hỗ trợ nhiều ngôn ngữ và chức năng tùy vào ngôn ngữ sử dụng theo như trong bảng sau. Nhiều chức năng của Visual Studio Code không hiển thị ra trong các menu tùy chọn hay giao diện người dùng. Thay vào đó, chúng được gọi thông qua khung nhập lệnh hoặc qua một tập tin .json (ví dụ như tập tin tùy chỉnh của người dùng). Khung nhập lệnh là một giao diện theo dòng lệnh. Tuy nhiên, nó biến mất khi người dùng nhấp bất cứ nơi nào khác, hoặc nhấn tổ hợp phím để tương tác với một cái gì đó ở bên ngoài đó. Tương tự như vậy với những dòng lệnh tốn nhiều thời gian để xử lý. Khi thực hiện những điều trên thì quá trình xử lý dòng lệnh đó sẽ bị hủy.

**1.3.4 Thư viện Tkinter**

Tkinter là một gói trong Python có chứa module **Tk** hỗ trợ cho việc lập trình GUI. **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. Ngoài Tkinter ra còn có một số công cụ khác giúp tạo một ứng dụng GUI viết bằng Python như wxPython, PyQt, và PyGTK.

**1.3.5 Mysql**

MySQL là hệ quản trị cơ sở dữ liệu tự do nguồn mở phổ biến nhất thế giới và được các nhà phát triển rất ưa chuộng trong quá trình phát triển ứng dụng. Vì MySQL là hệ quản trị cơ sở dữ liệu tốc độ cao, ổn định và dễ sử dụng, có tính khả chuyển, hoạt động trên nhiều hệ điều hành cung cấp một hệ thống lớn các hàm tiện ích rất mạnh. Với tốc độ và tính bảo mật cao, MySQL rất thích hợp cho các ứng dụng có truy cập CSDL trên internet. Người dùng có thể tải về MySQL miễn phí từ trang chủ. MySQL có nhiều phiên bản cho các hệ điều hành khác nhau: phiên bản Win32 cho các hệ điều hành dòng Windows, Linux, Mac OS X, Unix, FreeBSD, NetBSD, Novell, NetWare, SGI Irix, Solaris, SunOS,…

MySQL là một trong những ví dụ rất cơ bản về Hệ Quản trị Cơ sở dữ liệu quan hệ sử dụng Ngôn ngữ truy vấn có cấu trúc (SQL).

MySQL được sử dụng cho việc bổ trợ NodeJs, PHP, Perl, và nhiều ngôn ngữ khác, làm nơi lưu trữ những thông tin trên các trang web viết bằng NodeJs, PHP hay Perl,...

**1.4 Phương pháp giải quyết**

Những vấn đề cần giải quyết :

* Đầu tiên, nhìn vào hình và tìm tất cả các khuôn mặt có trong đó
* Thứ 2, tập trung vào một khuôn mặt của một người và nhận diện ngay cả khi khuôn mặt quay đi hướng khác, hoặc trong môi trường thiếu ánh sáng
* Thứ 3 chọn ra những đặc điểm đặc trưng của khuôn mặt mà bạn sử dụng để phân biệt với khuôn mặt của người khác. Vd: mắt lớn bao nhiêu, mặt dài bao nhiêu mũi cao hay không, …
* Cuối cùng, đối chiếu những đặc điểm đặc trưng đó với những người bạn đã biết và xác định được tên người đó

**1.5 Cấu trúc báo cáo**

* Chương 1: giới thiệu tổng quan
* Nội dung chương này giới thiệu bài toán nhận dạng khuôn mặt, cách triển khai giải quyết bài toán và các công cụ.
* Chương 2: Thuật toán nhận dạng
* Nội dung chương này diễn giải những hiểu biết cá nhân về thuật toán nhận diện khuôn mặt được sử dụng trong chưa trình thông qua các hàm của thư viện.
* Chương 3: Triển khai
* Nội dung chưa này thuật lại quá trình viết code cũng như các kết quá đạt được trong từng bước thông qua code và hình ảnh minh họa
* Chương 4: Kết luận và hướng phát triển
* Nội dung chương này sẽ kết luận, đánh giá quá trình nghiên cứu và hướng phát triển cho chương trình sau này.

**CHƯƠNG 2 XỬ LÝ ẢNH**

**2.1 Giới thiệu về hệ thống xử lí ảnh:**

Xử lí ảnh là một lĩnh vực mang tính khoa học và công nghệ. Nó là một ngành khóa học mớI mẻ so vớI nhiều ngành khoa học khác nhưng vớI tốc độ phát triển của nó rất nhanh.

Xử lí ảnh được đưa vào giảng dạy ở bậc đại học ở nước khoảng chục năm nay. Nó là một môn học liên qua nđến nhiều lĩnh vực và cần nhiều kiến thức cơ bản khác. Đầu tiên phải kể đến các biến đổi Fourier, biến đổi Laplace, các bộ lọc hữu hạn…. Thứ hai, công cụ toán như Đại số tuyến tính, xác xuất, thống kê…

Các phương pháp xử lý ảnh bắt đầu từ các ứng dụng chính: nâng cao chất lượng ảnh và phân tích ảnh. Ứng dụng đầu tiên được biết đến là nâng cao chất lượng ảnh truyền qua cáp từ những năm 1920 và được phát triển vào khoảng những năm 1955. Từ 1964 đến nay, các phương tiện xử lý, nâng cao chất lượng, nhận dạng ảnh phát triển không ngừng. Các phương pháp tri thức nhân tạo như mạng nơ ron nhân tạo, các thuật toán xử lý hiện đại và cải tiến.

Trước đây, ảnh được thu qua Camera ( kiểu CCIR). Gần đây, với sự phát triển của công nghệ, ảnh màu hoặc đen trắng được lấy ra từ Camera, sau được được chuyển trực tiếp thành ảnh số tạo thuận lợi cho xử lý tiếp theo. Ảnh còn có thể tiếp nhận từ vệ tinh.

* **Phần thu nhận ảnh:**

Ảnh có thể nhận qua camera màu hoặc đen trắng. Thường ảnh nhận qua camera là ảnh tương tự ( CCIR ), hay loạI mã số hóa ( CCD ).

* **Tiền xử lý:**

Ảnh có đợ nhiễu tương phản thấp nên cần đưa vào bộ tiền xử lý để nâng cao chất lượng. Chức năng chính của bộ tiền xử lý là lọc nhiễu, nâng cao độ tương phản để làm ảnh rõ hơn, nét hơn.

* **Phân đoạn:**

Phân vùng ảnh là tách một ảnh đầu vào thành các vùng thành phần để biễu diễn phân tích, nhận dạng ảnh.

* **Biểu diễn ảnh:**

Đầu ra ảnh sau phân đoạn chứa các điểm ảnh của vùng ảnh cộng với mã liên kết với các vùng lân cận. Việc biến đổi các số liệu này thành dạng thích hợp là cần thiết cho xử lý tiếp theo bằng máy tính.

* **Nhận dạng và nội suy ảnh:**

Nhận dạng ảnh là quá trình xác định ảnh. Quá trình này thường thu được bằng cách so sánh với mẫu chuẩn đã được lưu từ trước. Nội suy là phán đoán theo ý nghĩa trên cơ sở nhận dạng. Ảnh được nhận dạng theo hai loạI nhận dạng cơ bản:

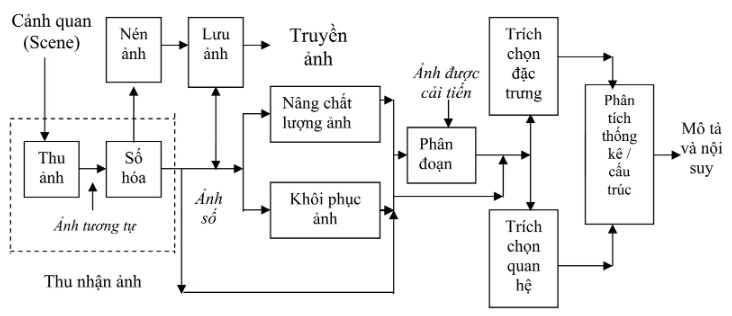
* Nhận dạng theo tham số
* Nhận dạng theo cấu trúc
* **Cơ sở trí thức:**

Ảnh là một đối tượng khá phức tạp về đường nét, độ sáng tối, dung lượng điểm ảnh, môi trường để thu ảnh phong phú kéo theo nhiễu. Trong nhiều khâu xử lý và phân tích ảnh ngoài việc đơn giản hóa phương pháp toán học đảm bảo tiện lợI cho xử lý.

* **Mô tả:**

Ảnh sau khi số hóa sẽ được lưu vào bộ nhớ, hoặc chuyển sang các khâu tiếp theo để phân tích. Nếu lưu trữ ảnh trực tiếp từ các ảnh thô, đòI hỏI dung lượng bộ nhớ cực lớn và không hiệu quả theo quan điểm ứng dụng và công nghệ.

* Biểu diễn bằng mã chạy
* Biểu diễn bằng mã xích
* Biểu diễn bằng mã tứ phân



**2.2 Những vấn đề cơ bản trong hệ thống xử lý ảnh:**

**2.2.1 Điểm ảnh**

Điểm ảnh ( Pixel ) là một phần tử của ảnh số tại tọa độ (x,y) với độ xám hoặc màu nhất định. Kích thước và khoảng cách giửa các điểm ảnh đó được chọn thích hợp sao cho mắt ngườI cảm nhận sự liên tục về không gian và mức xám của ảnh.

**2.2.2 Độ phân giải của ảnh**

Độ phân giải của ảnh là mật độ điểm ảnh được ấn định trên một ảnh số được hiển thị.

**2.2.3 Mức xám:**

* Mức xám của điểm ảnh là cường độ sáng của nó được gán bằng giá trị số tại điểm đó.
* Các thang giá trị mức xám thông thường: 16, 32, 64, 128, 256 ( Mức 256 là mức phổ dụng). 1 byte ( 8 bit ) để biểu diễn mức xám: Mức xám dùng 1 byte biểu diễm: wps = 256 mức, tức là từ 0 đến 255.
* Ảnh đen trắng: là ảnh có hai màu đen trắng với mức xám ở các điểm ảnh khác nhau.
* Ảnh nhị phân: ảnh chỉ có hai mức đen trắng phân biệt tức dùng 1 bit mô tả wps mức khác nhau. Nói cách khác là mỗi điểm ảnh của ảnh nhị phân chỉ có thể là 0 hoặc 1.
* Ảnh màu: Trong khuôn khổ lý thuyết ba màu (RGB) để tạo nên màu.

**2.2.4 Định nghĩa ảnh số**

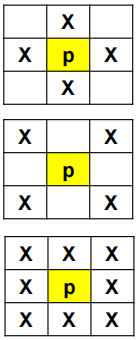
Là tập hợp các điểm ảnh với mức xám phù hợp dùng để mô tả ảnh gần với ảnh thật.

**2.2.5 Quan hệ giữa các điểm ảnh:**

* Tính lân cận:

Một điểm ảnh p tại tọa độ (x,y) có:

* 4 lân cận ngang-dọc của p: Ký hiệu là N4 (p): (x+1,y), (x-1,y), (x,y+1), (x,y-1)
* 4 lân cận chéo của p: Ký hiệu ND (p): (x+1,y+1), (x+1,y-1), (x- 1,y+1), (x-1,y-1)
* 8 lân cận của p: Ký hiệu N8 (p): là sự kết hợp của N4 (p) và ND (p): (x+1,y), (x-1,y), (x,y+1), (x,y-1), (x-1,y+1), (x-1,y-1)



* Tính liền kề:
* Cho V là tập các giá trị cấp xám, chúng ta có:

4 liền kề: Hai điểm ảnh p và q với các giá trị từ V là 4 liền kề nếu q thuộc tập N4 (p)

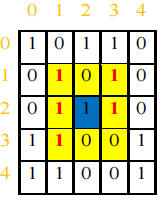
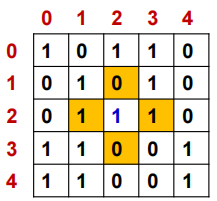
* Cho V là tập các giá trị cấp xám chúng ta có:

8 liền kề: Hai điểm ảnh p và q với các giá trị từ V là 8 liền kề nếu q thuộc tập N8 (p).

* Cho V là tập các giá trị cấp xám chúng ta có :

m liền kề (liền kề hỗn hợp): Hai điểm ảnh p và q với các giá trị từ V là m liền kề nếu:

* q thuộc N4 (p) hoặc
* q thuộc ND (p) và tập N4 (p)∩ N4 (q) rỗng-không có điểm ảnh nào có giá trị từ V. (Tập các điểm ảnh là 4 lân cận của p và q giao nhau không có giá trị nào từ V)



* Độ đo khoảng cách giữa các điểm ảnh:
* Cho các điểm ảnh p, q, z có tọa độ tương ứng là (x,y), (s,t), (u,v)
* D được gọi là hàm khoảng cách hay metric nếu:
* (a) D(p,q)≥0; D(p,q)=0 nếu và chỉ nếu p=q
* (b) D(p,q)=D(q,p)
* (c) D(p,z)≤D(p,q)+D(q,z)
* Khoảng cách Euclide giữa hai điểm p và q
* Đối với độ đo khoảng cách này các điểm ảnh có khoảng cách nhỏ hơn hoặc bằng giá trị r là các điểm được chứa trong đĩa có tâm (x,y)
* Khoảng cách khối D4(p,q) được gọi là khoảng cách khối đồ thị (CityBlock Distance)
* Các điểm ảnh với D4=1 là 4-lân cận của (x,y).

**2.3 Những vấn đề khác:**

**2.3.1 Biến đổi ảnh**

* Trong xử lý ảnh đòi hỏi dung lượng bộ nhớ lớn, thời gian tính toán lâu. Các phương pháp sau khi xử lý được thực hiện, dùng biến đổi ngược để đưa về miền xác định ban đầu.
* Biến đổI Furier, Cosin, Sin
* Biến đổi ảnh bằng tích chập, tích Kronecker

**2.3.2 Nén ảnh**

* Ảnh dù ở dạng nào vẫn chiếm không gian bộ nhớ rất lớn. khi mô tả ảnh, ngườI ta đưa kỹ thuật nén ảnh được chia ra thế hệ 1, 2.

**CHƯƠNG 3 THUẬT TOÁN NHẬN DẠNG**

**3.1 Phát hiện khuôn mặt**

* Phát hiện khuôn mặt (Face Detection) là một kĩ thuật máy tính để xác định được các vị trí và các kích thước của khuôn mặt người trong các ảnh bất kỳ. Kỹ thuật này nhận biết các đặc trưng của khuôn mặt và bỏ qua những thứ khác như: tòa nhà, cấy cối, cơ thể.

**3.1.1 Vấn đề**

* Như ở não của con người chúng ta có những sợi thần kinh để làm tất cả các việc đó một cách tự động và ngay lập tức. Trong thực tế, con người rất giỏi trong việc nhận diện khuôn mặt và tưởng tượng ra các khuôn mặt trong các vật thể hằng ngày



***Hình 3.1****: Hình dung các khuôn mặt từ vật thể*

* Máy tính thì lại không có khả năng bậc cao kiểu đó ít nhất là hiện tại (trong tương lai có thể có). Nên chúng ta cần dạy cho chúng cách để làm từng bước riêng biệt trong quá trình nhận dạng. Chúng ta cần xây dựng một quy trình(hệ thống) nơi chúng ta giải quyết từng bước của nhận diện khuôn mặt một cách riêng biệt, và chuyển kết quả hiện tại đó cho bước tiếp theo.

**3.1.2 Tìm tất cả khuôn mặt có trong ảnh**

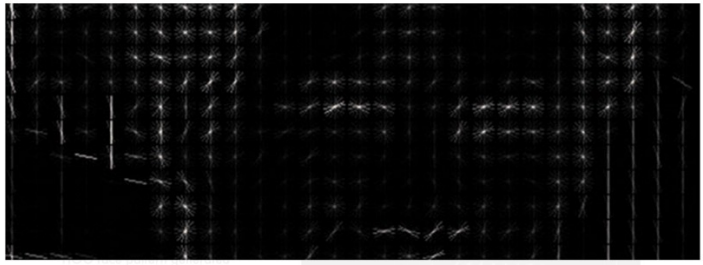


***Hình 3.2 :*** *Mục tiêu bước 1*

* Mục tiêu bước này là chúng ta sẽ tìm vị trí của khuôn mặt trong tấm hình trước khi chúng ta phân biệc chúng và qua bước tiếp theo.
* Nếu bạn đang sử dụng bất kì camera nào trong khoảng 5 năm trở lại đây thì bạn có thể thấy nhận diện khuôn mặt được tích hợp sẵn trong thiết bị.Nhận diện khuôn mặt là một tính năng tuyệt vời cho máy ảnh. Khi máy ảnh có thể tự động chọn ra các khuôn mặt, chúng có thể chắc chắn rằng tất cả các khuôn mặt được là nét trước khi nó ghi hình. Nhưng chúng ta lại sử dụng nó với mục đích khác, tìm kiếm vị trí khuôn mặt trong tấm hình và gửi kết quả đó cho bước tiếp theo trong quá trình.
* Nhận diện khuôn mặt đã trở thành xu hướng vào đầu những năm 2000 khi Paul Viola và Michael Jones phát minh ra cách để nhận diện khuôn mặt với tốc độ đủ để chạy trên con các dòng máy ảnh rẻ tiền. Tuy nhiên thì còn có các phương pháp đáng tin cậy cũng đã xuất hiện. Chúng ta sử dụng phương pháp được phát minh năm 2005 được gọi là “Histogram of Oriented Gradients” (rút gọn thành HOG).
* Để tìm những khuôn mặt trong một tấm hình, chúng ta bắt đầu với việc biến tấm ảnh của chúng ta thành tấm ảnh xảm, chỉ có đen và trắng vì chúng ta không cần màu sắc để tìm khuôn mặt. Sau đó chúng ta nhìn vào từng pixel trong tấm hình cùng một lúc. Với mỗi pixel một, chúng ta lại nhìn vào những pixel lân cận nó.
* Mục đích của chúng ta là tìm ra pixel hiện tại có màu tối như thế nào so với các pixel lân cận nó. Khi đó chúng ta sẽ vẽ một mũi tên theo chiều mà màu trở nên tối hơn.
* Nếu bạn lặp lại tiến trình đó với **mỗi pixel một trong trong** tấm ảnh, bạn sẽ kết thúc với mỗi pixel được thay thế bởi một mũi tên. Những mũi tên đó được gọi là “gradients”(vectơ độ dốc) và chúng chỉ ra dòng chảy(lưu lượng) từ những pixel sáng đến những pixel tối trên toàn bộ hình ảnh.
* Điều này có vẻ là một việc làm ngẫu nhiên, nhưng đó là một lí do tốt để thay thế các pixel đó thành *gradients*. Nếu chúng ta phân tích trực tiếp các pixel tối và các pixel sáng trong bức ảnh của cùng một người sẽ nhận được tổng các giá trị hoàn toàn khác nhau. Nhưng nếu xem xét hướng sáng thay đổi, cả hai hình ảnh tối và hình ảnh sáng sẽ cho kết quả với cùng một đại diện(con số) chính sác. Điều đó làm cho vấn đề dễ giải quyết hơn.
* Nhưng việc lưu gradient cho mỗi pixel đơn lẻ cho chúng ta quá nhiều chi tiết. Sẽ tốt hơn nếu chúng ta có thể nhìn thấy dòng chảy sáng / tối cơ bản ở mức cao hơn để chúng ta có thể thấy mô hình cơ bản của hình ảnh.
* Để làm điều này, chúng tôi sẽ chia hình ảnh thành các ô vuông nhỏ 16x16 pixel mỗi hình. Trong mỗi ô vuông, chúng tôi sẽ đếm xem có bao nhiêu độ dốc theo từng hướng chính (có bao nhiêu điểm hướng lên, hướng lên phải, điểm phải, v.v ...). Sau đó, chúng tôi sẽ thay thế hình vuông đó trong hình ảnh bằng các hướng mũi tên nổi bật nhất.
* Kết quả cuối cùng là chúng ta biến hình ảnh gốc thành một hình đại diện rất đơn giản, nắm bắt cấu trúc cơ bản của khuôn mặt một cách đơn giản hơn:

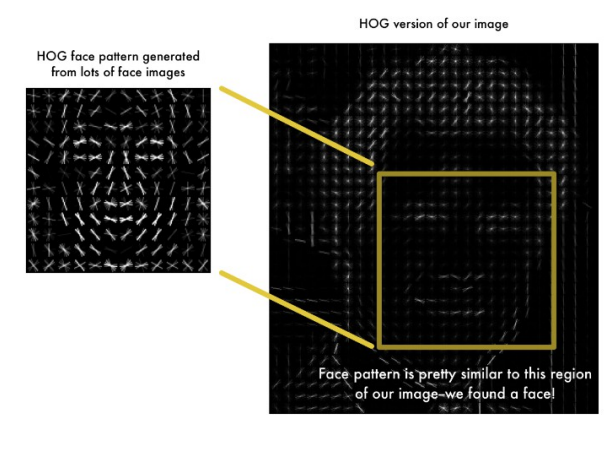
****

*Hình 3.3 : Ảnh input bước 1*



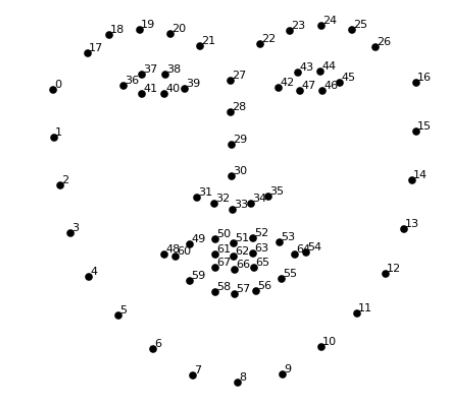
*Hình 3.4 Ảnh output bước 1*

* Để tìm các khuôn mặt trong hình ảnh HOG này, tất cả những gì chúng ta phải làm là tìm phần hình ảnh của chúng ta trông giống nhất với mẫu HOG đã biết được trích xuất từ một loạt các khuôn mặt đào tạo khác:



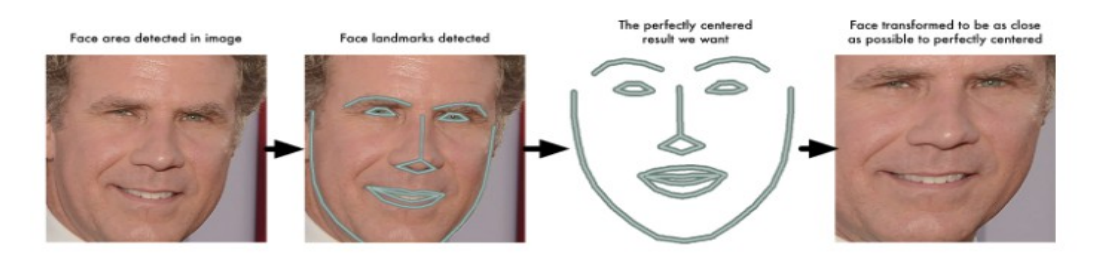
*Hình 3.5: Kết quả của thuật toán HOG*

**3.2 Đổi góc độ và chiều cho khuôn mặt**

* Để làm cho điều này, chúng ta sẽ cố gắng làm cong từng bức ảnh để mắt và môi luôn ở vị trí mẫu trong ảnh. Điều này sẽ giúp chúng ta dễ dàng hơn rất nhiều để so sánh khuôn mặt trong các bước tiếp theo.
* Để làm điều này, chúng ta sẽ sử dụng một thuật toán gọi là ước lượng mốc mặt. Có rất nhiều cách để làm điều này, nhưng chúng ta sẽ sử dụng phương pháp được phát minh vào năm 2014 bởi Vahid Kazemi và Josephine Sullivan.
* ****Ý tưởng cơ bản là chúng ta sẽ đưa ra 68 điểm cụ thể (được gọi là các mốc) tồn tại trên mỗi khuôn mặt - đỉnh cằm, cạnh ngoài của mỗi mắt, cạnh trong của mỗi lông mày, v.v. Sau đó, chúng ta sẽ huấn luyện một máy học thuật toán để có thể tìm thấy 68 điểm cụ thể này trên mọi mặt:

*Hình 3.6: 68 điểm mốc trên khuôn mặt*

* Giờ thì chúng ta đã biết được vị trí của mắt và miệng, chúng ta sẽ dùng các phép xoay đơn giản, chia cắt tỉ lệ hình ảnh sao cho mắt và miệng được căn giữa tốt nhất được gọi là biến đổi “affine”



* Bây giờ, bất kể khuôn mặt được xoay như thế nào, chúng ta có thể tập trung vào mắt và miệng ở cùng một vị trí trong ảnh. Điều này sẽ làm cho bước tiếp theo của chúng ta chính xác hơn rất nhiều

**3.3 Mã hóa khuôn mặt**

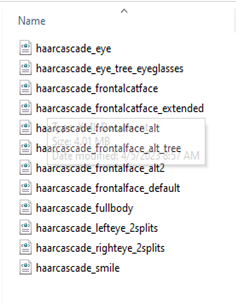
* Bây giờ chúng ta là phần cốt lõi của vấn đề - thực sự nói ra những khuôn mặt khác biệt. Cách tiếp cận đơn giản nhất để nhận dạng khuôn mặt là so sánh trực tiếp khuôn mặt chưa biết mà chúng tôi tìm thấy ở Bước 2 với tất cả các hình ảnh chúng tôi có của những người đã được gắn thẻ. Khi chúng ta tìm thấy một khuôn mặt được gắn thẻ trước đó trông rất giống với khuôn mặt chưa biết của chúng ta, nó phải là cùng một người. Có vẻ như là một ý tưởng khá tốt, phải không?
* Có một vấn đề rất lớn với cách tiếp cận đó. Một trang web như Facebook với hàng tỷ người dùng và hàng nghìn tỷ bức ảnh có thể vòng qua mọi khuôn mặt được gắn thẻ trước đó để so sánh nó với mọi bức ảnh mới được tải lên. Điều đó sẽ mất quá nhiều thời gian. Họ cần có khả năng nhận diện khuôn mặt tính bằng mili giây chứ không phải hàng giờ.
* Những gì chúng ta cần là một cách để trích xuất một vài phép đo cơ bản từ mỗi khuôn mặt. Sau đó, chúng ta có thể đo khuôn mặt chưa biết của mình theo cùng một cách và tìm ra khuôn mặt đã biết bằng các phép đo gần nhất. Ví dụ: chúng ta có thể đo kích thước của mỗi tai, khoảng cách giữa hai mắt, chiều dài của mũi, v.v
* Vậy cách đo khuôn mặt chính sác nhất là gì ? Kích thước tai? Mũi dài? Màu mắt? Thứ gì khác?
* Các nhà nghiên cứu đã phát hiện ra rằng phương pháp chính xác nhất là để máy tính tìm ra các phép đo để thu thập chính nó. Deep learning thực hiện công việc tốt hơn con người trong việc tìm ra phần nào của khuôn mặt là quan trọng để đo.
* Giải pháp là đào tạo một mạng lưới thần kinh .Nhưng thay vì đào tạo mạng để nhận ra các đối tượng hình ảnh như chúng ta đã làm lần trước, chúng ta sẽ đào tạo nó để tạo ra 128 phép đo cho mỗi khuôn mặt.
* Quá trình đào tạo hoạt động bằng cách nhìn vào 3 hình ảnh khuôn mặt cùng một lúc:
* Tải hình ảnh khuôn mặt đào tạo của một người đã biết (#1)
* Tải một hình ảnh khác của cùng một người được biết đến (#2)
* Tải hình ảnh của một người hoàn toàn khác (#3)
* Sau đó, thuật toán xem xét các phép đo mà nó hiện đang tạo cho mỗi trong số ba hình ảnh đó. Sau đó, nó tinh chỉnh mạng lưới thần kinh một chút để đảm bảo các phép đo mà nó tạo ra cho # 1 và # 2 gần hơn một chút trong khi đảm bảo các phép đo cho # 2 và # 3 cách xa nhau một chút

**PHẦN CODE DEMO**

**Sơ lược:**

A screenshot of a computer

AI-generated content may be incorrect.



Chương trình sẽ yêu cầu 1 số việc được làm trước khi bắt đầu viết code:

* - Chương trình yêu cầu người dùng cài đặt thư viện OpenCV.
* - Tạo 1 folder để chứa ảnh, ở đây là folder ‘image’.
* - Tạo 1 folder để chứa các khuôn mặt được cắt từ ảnh gốc, ở đây là folder ‘faces’.
* - Tạo 1 folder để chứa các bộ phân loại của haarcascade ( ở đây chỉ cần file ‘haarcascade\_frontalface\_default.xml’ ).

**Code:**

import cv2

#~ Đọc ảnh vào biến `img` từ file 'people.jpg' trong thư mục 'image'

img = cv2.imread('image/people.jpg')

#~ Tải bộ phận phân loại phát hiện khuôn mặt và lưu vào biến `face\_detector`

face\_detector = cv2.CascadeClassifier('haarcascades/haarcascade\_frontalface\_default.xml')

#~ Chuyển ảnh từ BGR sang ảnh xám để giảm bit và tăng tốc độ xử lý

gray = cv2.cvtColor(img , cv2.COLOR\_BGR2GRAY)

while True: #~ Sử dụng vòng lặp để lặp lại việc xử lý trên từng khung hình của ảnh

    #~ Sử dụng phương thức detectMultiScale để phát hiện khuôn mặt từ ảnh xám

    faces = face\_detector.detectMultiScale(gray, 1.3, 5)

    #~ Tạo biến đếm số lượng khuôn mặt được phát hiện

    count = 0

    #~ Sử dụng vòng lặp for để duyệt qua từng khuôn mặt đã phát hiện

    #~ Duyệt qua danh sách các hình chữ nhật và lấy ra tọa độ x, y, chiều rộng w, chiều cao h của mỗi hình chữ nhật

    for (x, y, w, h) in faces:

        #~ Vẽ hình chữ nhật màu đỏ quanh khuôn mặt được phát hiện

        cv2.rectangle(img, (x, y), (x + w, y + h), (0, 0, 255), 2)

        #~ Tăng biến đếm `count` lên để đánh số thứ tự

        count += 1

        #~ Cắt và thu nhỏ ảnh khuôn mặt để lưu trữ.

        #~ Hình ảnh được cắt bỏ 2 pixel ở mỗi cạnh để bỏ đi phần viền đỏ

        #~ Chỉnh sửa bằng `resize` để đồng bộ kích thước của mỗi hình ảnh thành (70, 70)

        img\_faces =cv2.resize(img[y+2: y+h-2, x+2: x+w-2], (70, 70))

        #~ Lưu ảnh khuôn mặt vào thư mục `faces`

        #~ Mỗi tệp được đặt tên và đánh theo số thứ tự

        cv2.imwrite('faces/people\_{}.jpg'.format(count), img\_faces)

    cv2.imshow('IMAGE', img)

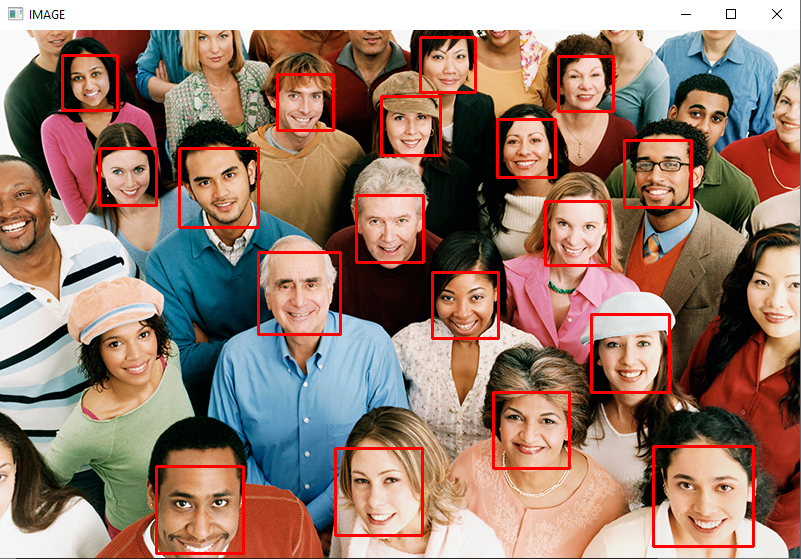
    k = cv2.waitKey(10) & 0xff #~ Bấm ESC để thoát

    if k == 27:

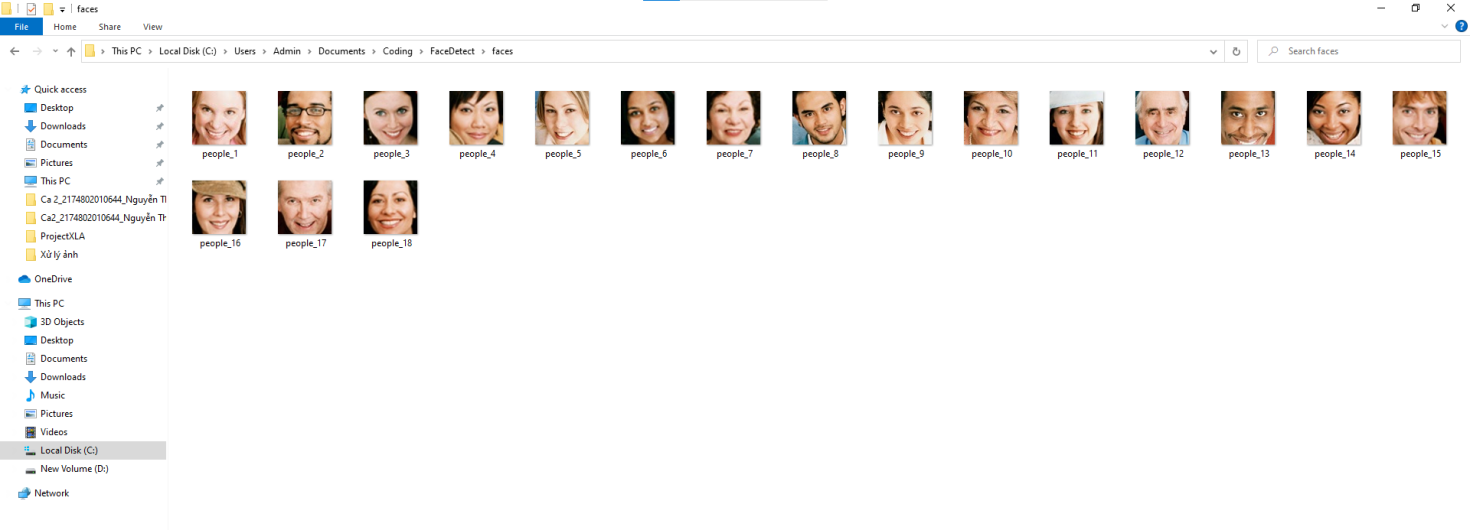
        break

cv2.destroyAllWindows()

**Kết quả:**



- Các khuôn mặt được cắt ra sẽ được lưu vào folder ‘faces’ để tiện cho các hoạt động sau này.



**TÀI LIỆU THAM KHẢO**

- Peter Dauvergne .*Identified, Tracked, and Profiled: The Politics of Resisting Facial Recognition Technology*, Edward Elgar Publishing.

- Dr. Awanit Kumar , Dr. Sheshang Degadwala and Dr. Darshana Pandya .

*Machine Learning for Emotion & Facial Recognition*, Scholars' Press.

- Zhang Zhi Yong Song Bin .*Facial Expression Recognition Technology for Human-Computer Interaction*, Science Press.