**BỘ GIÁO DỤC VÀ ĐÀO TẠO**

**TRƯỜNG ĐẠI HỌC TÂY ĐÔ**

**KHOA KỸ THUẬT – CÔNG NGHỆ**

🙠🙠 🕮 🙢🙢

**TIỂU LUẬN TỐT NGHIỆP ĐẠI HỌC**

**NHẬN DIỆN CẢM XÚC MẮT**

**CHUYÊN NGÀNH CÔNG NGHỆ THÔNG TIN**

***Giảng viên hướng dẫn: Sinh viên thực hiện:***

ThS. Bùi Xuân Tùng Trần Thành Tiến 1752060038

**Cần Thơ, Tháng 5/2021**

**BỘ GIÁO DỤC VÀ ĐÀO TẠO**

**TRƯỜNG ĐẠI HỌC TÂY ĐÔ**

**KHOA KỸ THUẬT – CÔNG NGHỆ**

🙠🙠 🕮 🙢🙢

**TIỂU LUẬN TỐT NGHIỆP ĐẠI HỌC**

**NHẬN DIỆN CẢM XÚC MẮT**

**CHUYÊN NGÀNH CÔNG NGHỆ THÔNG TIN**

***Giảng viên hướng dẫn: Sinh viên thực hiện:***

ThS. Bùi Xuân Tùng Trần Thành Tiến 1752060038

**Cần Thơ, Tháng 5/2021**

**LỜI CẢM ƠN**

Trên thực tế không có sự thành công nào mà không gắn liền với những sự hỗ trợ, giúp đỡ dù ít hay nhiều, dù trực tiếp hay gián tiếp của người khác. Trong suốt thời gian làm tiểu luận tốt nghiệp, em đã tìm hiểu, học hỏi và nhận được rất nhiều sự hỗ trợ từ thầy và các anh chị khóa trước. Với lòng biết ơn sâu sắc, em xin gửi lời cảm ơn đến thầy ThS. Bùi Xuân Tùng đã truyền đạt vốn kiến thức quý báu cho em cũng như các bạn cùng khoá, đã tận tâm hướng dẫn em qua từng buổi hướng dẫn để hoàn thành tiểu luận tốt nghiệp. Nếu không có những lời hướng dẫn, dạy bảo của thầy thì em nghĩ tiểu luận tốt nghiệp của em khó có thể hoàn thiện được. Một lần nữa em xin chân thành cảm ơn thầy. Qua đây em cũng xin cảm ơn các anh chị khóa trước đã đóng góp ý kiến giúp đỡ em để hoàn thành đồ án tốt nghiệp này.

Cần Thơ, ngày …. tháng …. năm 2021

**LỜI CAM KẾT**

Em xin cam kết bài tiểu luận này được hoàn thành dựa trên các kết quả nghiên cứu của em và các kết quả nghiên cứu này chưa được dùng cho bất cứ tiểu luận nào cùng cấp khác.

Cần Thơ, ngày …. tháng …. năm 2021

Sinh viên thực hiện

Trần Thành Tiến

**MỤC LỤC**

Contents

[LỜI CẢM ƠN 1](#_Toc71981518)

[LỜI CAM KẾT 2](#_Toc71981519)

[MỤC LỤC 3](#_Toc71981520)

[MỤC LỤC HÌNH ẢNH 4](#_Toc71981521)

[CHƯƠNG 1: TỔNG QUAN ĐỀ TÀI 5](#_Toc71981522)

[**1.1. Đặt vấn đề** 6](#_Toc71981523)

[**1.2. Phương pháp thực hiện** 7](#_Toc71981524)

[**1.3. Kế hoạch thực hiện** 8](#_Toc71981525)

**CHƯƠNG 2: CƠ SỞ LÝ LUẬN……………………………………………………7.**

**2.1. Cơ sở lý luận…………………………………………………………………..7.**

[**2.2. Nhận Diện Cảm Xúc MắT…………………………………………………….7**](#_Toc71981526)

[**2.3. Xử lý ngôn ngữ tự nhiên** 8](#_Toc71981527)

[**2.3.1. Khái niệm** 8](#_Toc71981528)

**2.4.Nhận Diện……………………………………………………………………..**9

[**2.4.1. Nhận Diện Khuôn mặt và mắt** 9](#_Toc71981530)

[**2.4.2. Khó Khăn** ……………………………9](#_Toc71981531)

**2.4.3. Giao tiếp giữa người và máy dựa trên NLP…………………………….10**

**2.5. Ứng Dụng**……………………………………………………………………....11

[**2.5.1. Công cụ lập trình** 11](#_Toc71981532)

**2.5.2. Ngôn ngử lập trình………………………………………………………**11

**2.5.3. SQL Lite………………………………………………………………….11**

[CHƯƠNG 3: PHƯƠNG PHÁP NGHIÊN CỨUI 13](#_Toc71981522)

**3.1. Thuật toán sử dụng trong Nhận diện mắt…………………………………..13**

**3.1.1. Thuật toán Haar Cade…………………………………………………...13**

**3.1.2: Thuật toán OpenCV……………………………………………………..16**

[CHƯƠNG 4: KẾT QUẢ NGHIÊN CỨU…………………………………………..18](#_Toc71981522)

[**4.1. Phát triển bài toán** 18](#_Toc71981534)

[4.1.1 Tại sao nhận dạng khuôn mặt là một việc khó ? 18](#_Toc71981535)

**4.2. Chương trình thực nghiệm…………………………………………………19**

**4.2.1 Nhận xét và kết quả …………………………………………………………20 CHƯƠNG 5: KẾT LUẬN………………………………………………………….21**

**MỤC LỤC HÌNH ẢNH**

**Hình 2.2: Sử dụng nhận diện Khuôn mặt và Mắt………………………….10**

**Hình 2.4.1. Camera Nhận Diện……………………………………………...11**

**Hình 3.1.1, Cửa sổ Haar……………………………………………………...16**

**Hình 3.1.2: Thuật Toán OpenCV……………………………………………17**

**CHƯƠNG 1: TỔNG QUAN ĐỀ TÀI**

**1.1. Đặt vấn đề**

Ngày nay, Eye detection luôn là một chủ đề thú vị, chủ yếu là vì kết quả rất trực quan - và nó không cần nhiều trí tưởng tượng để hiểu cách face detection có thể được sử dụng trong các ứng dụng trong thế giới thực. Từ hệ thống an ninh gia đình, đến giám sát, hoặc chỉ đơn giản là điều khiển TV của bạn, nhận diện khuôn mặt là một phần quan trọng của thị giác máy tính. Và hôm nay, chúng ta có thể khám phá cách thức mà nó thực hiện trong OpenCV.

**1.2. Phương pháp thực hiện**

* Nghiên cứu tài liệu về ngôn ngữ tự nhiên (natural language processing - NLP)
* Tìm hiểu về Nhận Diện Mắt
* Tìm hiểu cài đặt python và các thư viện có liên quan
* Phân tích thông tin, thiết kế các chức năng, lập trình xử lí chương trình
* Các thuật toán liên quan

**1.3. Kế hoạch thực hiện**

Đề tài dự kiến hoàn thành trong 8 tuần, cụ thể như sau:

* Tuần 1: Tìm hiểu vấn đề
* Tuần 2: Viết chi tiết đặt tả yêu cầu cho sản phẩm
* Tuần 3: Thiết kế
* Tuần 4+5+6: Lập trình
* Tuần 7: Kiểm duyệt, hiệu chỉnh
* Tuần 8: Viết tài liệu và hướng dẫn sử dụng

**CHƯƠNG 2: CƠ SỞ LÝ LUẬN**

**2.1.** **Cơ Sở Lý Luận**

Trong những năm gần đây các hệ thống nhận dạng ngày càng phát triển. Một trong  
những ứng dụng rất thực tế đó là nhận dạng trạng thái của mắt có thể được áp dụng  
để để kiểm tra tài xế ngủ gật và đưa ra cảnh báo. Theo dõi chuyển động của mắt để  
điều khiển các thiết bị công nghệ, xe lăn cho người tàn tật.  
Ngủ gật là một trong những nguyên nhân dẫn đến các vụ tai nạn giao thông. Báo  
cáo về "Rối loạn giấc ngủ và tai nạn giao thông" tại hội nghị khoa học thường niên  
Hội Hô hấp Việt Nam và Chương trình đào tạo y khoa liên tục 2015, giáo sư Telfilo  
Lee Chiong (Trung tâm Nationnal Jewish Health, Mỹ) cho biết thiếu ngủ là một  
trong những nguyên nhân chính gây tai nạn giao thông trên thế giới. Ước tính  
khoảng 10-15% tai nạn xe có liên quan đến thiếu ngủ. Nghiên cứu về giấc ngủ ở các  
tài xế 19 quốc gia châu Âu cho thấy tỷ lệ buồn ngủ khi lái xe cao, trung bình 17%.  
Trong đó 10,8% người buồn ngủ khi lái xe ít nhất một lần trong tháng, 7% từng gây  
tai nạn giao thông do buồn ngủ, 18% suýt xảy ra tai nạn do buồn ngủ [4]. Do đó,  
cảnh báo tài xế khi có hiện tượng buồn ngủ là một vấn đề thiết thực góp phần làm  
giảm tỷ lệ tai nạn giao thông đang ngày càng gia tăng hiện nay.  
Từ những vấn đề nêu trên tôi chọn đề tài “nhận diện trạng thái mắt” dùng ngôn ngữ  
Python mô phỏng, thực hiện nhận dạng trạng thái của mắt bằng bộ phân loại  
AdaBoost rút trích đặc trưng Haar like nhận hình ảnh thu được từ camera.

**2.2. Nhận Diện Cảm Xúc Mắt**

Phát hiện khuôn mặt và mắt là một kỹ thuật máy tính để xác định vị trí của mặt người trong ảnh. Việc phát hiện mắt thực hiện trên khuôn mặt đã được phát hiện đó. Có bốn  
hướng tiếp cận để phát hiện khuôn mặt và mắt người:  
- Hướng tiếp cận dựa trên tri thức: dựa vào các thuật toán, mã hoá các đặc trưng và  
quan hệ giữa các đặc trưng của khuôn mặt và mắt thành các luật.  
- Hướng tiếp cận dựa trên đặc trưng không thay đổi: xây dựng thuật toán để tìm ra  
các đặc trưng của khuôn mặt và mắt, các đặc trưng này không thay đổi khi tư thế  
khuôn mặt hay vị trí camera thay đổi.  
- Hướng tiếp cận dựa trên so sánh khớp mẫu: dùng các mẫu chuẩn của khuôn mặt  
và mắt để mô tả. Mẫu này được sử dụng để phát hiện khuôn mặt và mắt bằng cách  
quét nó qua ảnh và tính toánh giá trị tương đồng cho mỗi vị trí. Việc xuất hiện một  
khuôn mặt tại một vị trí nào đó trong ảnh phụ thuộc vào giá trị tương đồng của điểm đó so với mẫu chuẩn.  
- Hướng tiếp cận dựa trên diện mạo: trái với hướng tiếp cận dựa trên khớp mẫu, các  
mô hình sẽ được học từ một tập ảnh huấn luyện mà thể hiện tính chất tiêu biểu của  
sự xuất hiện mặt người và mắt trong ảnh. Sau đó hệ thống sẽ xác định mặt người và  
mắt. Phương pháp này còn được gọi bằng phương pháp học máy.  
Việc tiếp cận theo mỗi hướng cần những thuật toán riêng biệt để mô tả những điểm  
đặc biệt..

Từ trên chúng ta có thể hiểu đơn giản trợ lý ảo là phần mềm, công cụ được xây dựng dựa trên nền tảng trí thông minh nhân tạo (AI) và được các nhà phát triển hệ điều hành, hay các hãng công nghệ tích hợp sâu vào trong hệ điều hành với mục đích chính là hỗ trợ người dùng thiết bị dễ dàng hơn bằng chính thói quen mà người dùng thường làm trên thiết bị.



Hình 2.2. Sử Dụng Nhận Diện Mắt

**2.3. Xử Lý Ngôn Ngữ Tự Nhiên**

**2.3.1. Khái Niệm**

Xử lý ngôn ngữ tự nhiên là một nhánh của Trí tuệ nhân tạo, tập trung vào việc nghiên cứu sự tương tác giữa máy tính và ngôn ngữ tự nhiên của con người, dưới dạng tiếng nói (speech) hoặc văn bản (text). Mục tiêu của lĩnh vực này là giúp máy tính hiểu và thực hiện hiệu quả những nhiệm vụ liên quan đến ngôn ngữ của con người như: tương tác giữa người và máy, cải thiện hiệu quả giao tiếp giữa con người với con người, hoặc đơn giản là nâng cao hiệu quả xử lý văn bản và lời nói.

Xử lý ngôn ngữ tự nhiên ra đời từ những năm 40 của thế kỷ 20, trải qua các giai đoạn phát triển với nhiều phương pháp và mô hình xử lý khác nhau. Có thể kể tới các phương pháp sử dụng ô-tô-mát và mô hình xác suất (những năm 50), các phương pháp dựa trên ký hiệu, các phương pháp ngẫu nhiên (những năm 70), các phương pháp sử dụng học máy truyền thống (những năm đầu thế kỷ 21), và đặc biệt là sự bùng nổ của học sâu trong thập kỷ vừa qua.

**2.4.Nhận Diện Mắt**

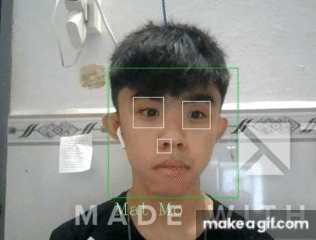
**2.4.1. Nhận Diện Khuôn mặt và mắt**

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.

* Dùng Online Camera để nhận diện



Hình 2.4.1. Camera Nhận Diện

**2.4.2. Khó khăn**

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 2.4.2: Hình Dung Các Khuôn Mặt

#### 

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

Ở góc độ ngữ pháp, câu này có thể được giải thích theo hai cây cú pháp như trên Hình 2.4. Những cấu trúc khác nhau dẫn đến những cách hiểu khác nhau: “a computer understands you like your mother does” hoặc “a computer understands that you like your mother”. Hiện tượng này gây khó khăn cho cả hai bài toán là phân tích cú pháp và phân tích ngữ nghĩa.

**2.4.3. Giao tiếp giữa người và máy dựa trên NLP**

Chúng ta có thể kỳ vọng ngày càng nhiều hơn các doanh nghiệp áp dụng cá nhân hóa trải nghiệm vào dịch vụ của họ. Công nghệ nhận diện khuôn mặt và trí tuệ nhân tạo sẽ giúp điều này trở nên dễ dàng hơn bao giờ hết. Bằng việc thu thập và phân tích độ tuổi, giới tính, sở thích, lịch sử mua sắm hệ thống sẽ tự động đưa ra những nội dung và dịch vụ phù hợp với khác hàng giúp tăng trải nghiệm khách hàng tốt hơn.

Mới đây hãng hàng không Delta đã triển khai công nghệ xác thực khuôn mặt tại sân bay Atlanta. Dự kiến vào cuối năm 2020 Delta sẽ triển khai nhận diện khuôn mặt ở hơn 20 sân bay để cải thiện hành trình của khách hàng.

Khách hàng sẽ có thể sử dụng khuôn mặt của mình để làm thủ tục, kiểm tra hành lý, vào phòng chờ và lên máy bay. Mặc dù những lo ngại đã được đặt ra về việc sử dụng nhận diện khuôn mặt để xác định danh tính tại sân bay, tuy nhiên, 73% khách hàng nói rằng họ sẽ cảm thấy thoải mái khi sử dụng công nghệ này sau khi đi qua hệ thống nhận diện khuôn mặt của Delta.

Công nghệ xử lý ngôn ngữ tự nhiên ngày càng có nhiều ứng dụng tốt, dần phổ biến, và gần gũi với cuộc sống hàng ngày. Với sự tiến bộ không ngừng về mặt kỹ thuật, tốc độ xử lý, độ chính xác được cải thiện, xử lý ngôn ngữ tự nhiên đã từng bước trở thành công nghệ giúp máy tính hiểu và giao tiếp với con người thông qua ngôn ngữ của chính chúng ta.

**2.4.4. Công Cụ và Ngôn Ngữ Lập Trình**

**2.5. Ứng Dụng**

**2.5.1. Công Cụ Lập Trình**

[Sublime Text](https://dizibrand.com/sublime-text-la-gi/) là một phần mềm lập trình với đầy đủ tính năng để chỉnh sửa các tệp cục bộ hoặc đoạn code. Nó bao gồm các tính năng khác nhau để chỉnh sửa code giúp các nhà lập trình theo dõi các thay đổi này. Các tính năng khác nhau được [Sublime Text](https://www.sublimetext.com/) hỗ trợ như sau :

- Syntax Highlight (Tô sáng cú pháp)

- Auto Indentation (Tự động thụt lề)

- File Type Recognition (Nhận dạng loại tệp)

- Sidebar with files of mentioned directory (Thanh bên với các tập tin của thư mục được đề cập)

- Macros

- Plug-in and Packages

được sử dụng [Integrated Development Editor](https://en.wikipedia.org/wiki/Integrated_development_environment) (IDE) chương trình chỉnh sửa phát triển tích hợp giống như mã [Visual Studio](https://visualstudio.microsoft.com/) và [NetBeans](https://netbeans.org/). Phiên bản hiện tại của trình soạn thảo Sublime Text là 3.0 và tương thích với các hệ điều hành khác nhau như Windows, Linux và MacOS.

**2.5.2. Ngôn Ngử Lập Trình**

Python là ngôn ngữ lập trình hướng đối tượng, cấp cao, mạnh mẽ, được tạo ra bởi Guido van Rossum. Nó dễ dàng để tìm hiểu và đang nổi lên như một trong những ngôn ngữ lập trình nhập môn tốt nhất cho người lần đầu tiếp xúc với ngôn ngữ lập trình. Python hoàn toàn tạo kiểu động và sử dụng cơ chế cấp phát bộ nhớ tự động. Python có cấu trúc dữ liệu cấp cao mạnh mẽ và cách tiếp cận đơn giản nhưng hiệu quả đối với lập trình hướng đối tượng. Cú pháp lệnh của Python là điểm cộng vô cùng lớn vì sự rõ ràng, dễ hiểu và cách gõ linh động làm cho nó nhanh chóng trở thành một ngôn ngữ lý tưởng để viết script và phát triển ứng dụng trong nhiều lĩnh vực, ở hầu hết các nền tảng.

Được ứng dụng rộng rãi trong nhiều lĩnh vực như: lập trình ứng dụng web, khoa học và tính toán, tạo nguyên mẫu phần mềm, sử dụng để dạy lập trình cho trẻ em và những người mới lần đầu học lập trình.

**2.5.3. Thư Viện SQL Lite**   
SQLite là một hệ quản trị cơ sở dữ liệu hay còn gọi là hệ thống cơ sở dữ liệu quan hệ nhỏ gọn, SQLite là một thư viện phần mềm mà triển khai một SQL Database Engine truyền thống, không cần mô hình client-server nên rất nhỏ gọn. SQLite được sử dụng vào rất nhiều chương trình từ desktop đến mobile hay là website.

Năm 2000: D.Richard Hipp đã thiết kế SQLite dưới dạng thư viện bằng ngôn ngữ lập trình c với mục đích không cần quản trị để điều hành một chương trình.  
Vào tháng 8, SQLite 1.0 được công bố với GNU Database Manager.  
Năm 2011: Hipp bổ sung UNQI Interface cho SQLite DB và để phát triển UNQLite (là một Document Oriented Database).

SQLite là mã nguồn mở, là một trong những Database Engine phát triển nhanh nhất, sự phát triển của nó là về mặt tính phổ biến. Ngoài những lý do trên thì không thể không kể đến những ưu điểm khi sử dụng SQLite, sau đây là phần ưu điểm của SQLite.

SQLite không cần mô hình client – server để hoạt động.

SQLite không cần phải cấu hình tức là bạn không cần phải cài đặt.

Với SQLite database được lưu trữ trên một tập tin duy nhất.

SQLite hỗ trợ hầu hết các tính năng của ngôn ngữ truy vấn SQL theo chuẩn SQL92.

SQLite rất nhỏ gọn bản đầy đủ các tính năng nhỏ hơn 500kb, và có thể nhỏ hơn nếu lược bớt một số tính năng.

Các thao tác dữ liệu trên SQLite chạy nhanh hơn so với các hệ quản trị cơ sở dữ liệu theo mô hình client – server.

SQLite rất đơn giản và dễ dàng sử dụng.

SQLite tuân thủ 4 tính chất ACID (là tính nguyên tổ (Atomic), tính nhất quán **(Consistent), tính cô lập (Isolated), và tính bền vững (Durable) ).**

Với đặc tính nhỏ gọn, truy xuất dữ liệu nhanh SQLite thường được sử dụng để nhúng vào các dự án.

**CHƯƠNG 3: PHƯƠNG PHÁP NGHIÊN CỨU**

**3.1. Thuật toán sử dụng trong Nhận diện mắt**

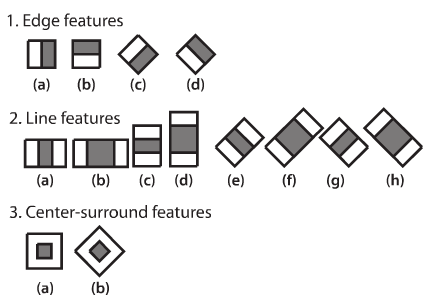
**3.1.1. Thuật toán Haar Cade**

Về cơ bản là sử dụng các đặc trưng loại Haar và sau đó sử dụng thật nhiều đặc trưng đó qua nhiều lượt (cascade) để tạo thành một cỗ máy nhận diện hoàn chỉnh. Vẫn khó hiểu phải không? Vậy chúng ta nhảy vào từng khái niệm một nhé.

Nếu bạn đã làm việc với xử lý ảnh hoặc Convolutional Neural Networks rồi thì chắc bạn cũng không còn lạ gì với các bộ lọc trong xử lý ảnh nữa cả — nếu không, bạn có thể đọc [phần này trong một bài của mình về CNN](https://viblo.asia/p/tai-sao-mang-tich-chap-lai-hoat-dong-hieu-qua-jvEla076Zkw#_cac-lop-tich-chap-co-ban-tu-xu-ly-anh-co-dien-9) để hiểu thêm. Các ví dụ bộ lọc được liệt kê ở dưới, trong đó a) là các bộ lọc bắt các cạnh trong ảnh, và b) bắt các đường thẳng trong ảnh, tương tự như các bộ lọc đã được mình nhắc tới trong bài trên. Ngoài ra, còn có các bộ lọc Haar khác, như ví dụ c) về đặc trưng 4-hình vuông dưới đây, 

hoặc đặc trưng nằm gọn trong trung tâm một vùng như

ví dụ 3. trong ảnh dưới đây:



Tuy nhiên, cách áp dụng các bộ lọc này khác một chút so với các cửa sổ bộ lọc bên CNN. Ở CNN, bộ lọc chiếm toàn bộ cửa sổ trượt, trong khi ở đặc trưng Haar, bộ lọc chỉ chiếm một phần trong cửa sổ trượt thôi. Điều đó được minh hoạ trên ảnh sau:



Hình 3.1.1. Cửa Sổ Haar

Trong hình trên, cửa sổ trượt được đặt ngay ngắn vừa gọn để nhìn được toàn bộ ảnh. Các bạn có thể nhận ra rằng bộ lọc đầu trong đó đang tìm một "cạnh" phân cách giữa mắt/lông mày với mũi, vì ở đoạn đọc có chênh lệch về màu đáng kể; và ở bộ lọc sau, mô hình đang tìm đường sống mũi, vì ở đó sẽ có màu sáng hơn so với 2 bên (vì nó cao hơn dễ bắt sáng). Và như đã nói trên, bộ lọc Haar chỉ nhìn cụ thể vào một vùng trong cửa sổ để tìm thôi: trong khuôn mặt thì mũi lúc nào cũng ở chính giữa chứ không ở các góc, nên không cần nhìn các góc để làm gì cả.

.

**3.1.2.Thuật Toán OpenCV**

OpenCV được sử dụng cho đa dạng nhiều mục đích và ứng dụng khác nhau bao gồm [1]:

Hình ảnh street view

Kiểm tra và giám sát tự động

Robot và xe hơi tự lái

Phân tích hình ảnh y học

Tìm kiếm và phục hồi hình ảnh/video

Phim – cấu trúc 3D từ chuyển động

Nghệ thuật sắp đặt tương tác

Tính năng và các module phổ biến của OpenCV

Theo tính năng và ứng dụng của OpenCV, có thể chia thư viện này thánh các nhóm tính năng và module tương ứng như sau:

Xử lý và hiển thị Hình ảnh/ Video/ I/O (*core, imgproc, highgui*)

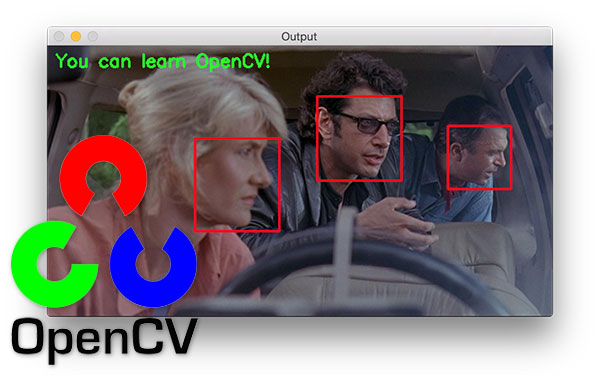
Phát hiện các vật thể (objdetect, features2d, nonfree)

Geometry-based monocular hoặc stereo computer vision (*calib3d, stitching, videostab*)

Computational photography (*photo, video, superres*)

Machine learning & clustering (*ml, flann*)

CUDA acceleration (*gpu*)



Hình 3.1.2. Thuật Toán OpenCV

OpenCV có cấu trúc module, nghĩa là gói bao gồm một số thư viện liên kết tĩnh (static libraries) hoặc thư viện liên kết động (shared libraries). Xin phép liệt kê một số định nghĩa chi tiết các module phổ biến có sẵn [2] như sau:

[**Core functionality**](https://docs.opencv.org/master/d0/de1/group__core.html) (core) – module nhỏ gọn để xác định cấu trúc dữ liệu cơ bản, bao gồm mảng đa chiều dày đặc và nhiều chức năng cơ bản được sử dụng bởi tất cả các module khác.

[**Image Processing**](https://docs.opencv.org/master/d7/dbd/group__imgproc.html) (imgproc) – module xử lý hình ảnh gồm cả lọc hình ảnh tuyến tính và phi tuyến (linear and non-linear image filtering), phép biến đổi hình học (chỉnh size, afin và warp phối cảnh, ánh xạ lại dựa trên bảng chung), chuyển đổi không gian màu, biểu đồ, và nhiều cái khác.

[**Video Analysis**](https://docs.opencv.org/master/d7/de9/group__video.html) (video) – module phân tích video bao gồm các tính năng ước tính chuyển động, tách nền, và các thuật toán theo dõi vật thể.

[**Camera Calibration and 3D Reconstruction**](https://docs.opencv.org/master/d9/d0c/group__calib3d.html) (calib3d) – thuật toán hình học đa chiều cơ bản, hiệu chuẩn máy ảnh single và stereo (single and stereo camera calibration), dự đoán kiểu dáng của đối tượng (object pose estimation), thuật toán thư tín âm thanh nổi (stereo correspondence algorithms) và các yếu tố tái tạo 3D.

[**2D Features Framework**](https://docs.opencv.org/master/da/d9b/group__features2d.html) (features2d) – phát hiện các đặc tính nổi bật của bộ nhận diện, bộ truy xuất thông số, thông số đối chọi.

[**Object Detection**](https://docs.opencv.org/master/d5/d54/group__objdetect.html) (objdetect) – phát hiện các đối tượng và mô phỏng của các hàm được định nghĩa sẵn – predefined classes (vd: khuôn mặt, mắt, cốc, con người, xe hơi,…).

[**High-level GUI**](https://docs.opencv.org/master/d7/dfc/group__highgui.html) (highgui) – giao diện dễ dùng để thực hiện việc giao tiếp UI đơn giản.

[**Video I/O**](https://docs.opencv.org/master/dd/de7/group__videoio.html) (videoio) – giao diện dễ dùng để thu và mã hóa video.

**GPU**– Các thuật toán tăng tốc GPU từ các modun OpenCV khác.

… và một số module hỗ trợ khác, ví dụ như FLANN và Google test wrapper, Python binding, v.v.

Sử dụng Machine Learning để vusualize customer prefernces

OpenCV hiện tại hỗ trợ nhiều ngôn ngữ, mỗi ngôn ngữ có thế mạnh riêng, vậy thì tùy theo nhu cầu mà chọn ngôn ngữ cho phù hợp.

C++: Đây là ngôn ngữ phổ biến nhất hiện tại vì nhanh, nhiều option, nếu bạn có IDE là Visual Studio quá tốt. Các thiết lập của nó rất hữu ích cho sản phẩm sau này, mặc dù ban đầu mới đụng tay khá phức tạp

Python: Ngôn ngữ được dùng nhiều để demo / test OpenCV do tính ngắn gọn, ít phải thiết lập. Bên cạnh đó, nếu dùng Python thì cũng có thể code được trên nhiều hệ điều hành.

Android: Hiện tích hợp sẵn camera, tiện lợi nên có thể sớm trở thành xu hướng của OpenCV

Java: Nhanh và đa nền tảng, tương tự C++

C#: Code tiện lợi, dễ dàng, có thư viện đa nền tảng là EmguCV hỗ trợ. Có một điểm trừ là EmguCV yêu cầu người dùng phải copy toàn bộ file \*.dll của lib đi kèm với ứng dụng nên làm ứng dụng sẽ khá nặng.

**CHƯƠNG 4: KẾT QUẢ NGHIÊN CỨU**

**4.1. Phát triển bài toán**

Nhận dạng khuôn mặt và mắt là một bài toán hấp dẫn, đã nhận được

rất nhiều sự quan tâm của các nhà nghiên cứu vì tính ứng dụng to lớn

trong thực tế. Báo cáo đã trình bày tổng quan về phương pháp nhận

dạng khuôn mặt người; phát hiện khuôn mặt; theo vết mắt; xây

dựng chương trình thử nghiệm sử dụng phương pháp nhận dạng

khuôn mặt người bằng thư viện opencv.

Nhưng nhận dạng khuôn mặt luôn là một bài toán khó

4.1.1 Tại sao nhận dạng khuôn mặt là một việc khó ?

* + **Ánh sáng:** Ảnh kỹ thuật số biểu diễn cường độ sáng của đối tượng, do đó khi ánh sáng thay đổi, thông tin về đối tượng sẽ bị ảnh hưởng.
  + **Cự ly của đối tượng so với camera:** khoảng cách đối tượng so với camera sẽ xác định số pixel ảnh quy định nên khuôn mặt.
  + **Cảm xúc biểu cảm trên khuôn mặt:** các nét biểu cảm cảm xúc trên khuôn mặt gây ra nhiễu, việc loại nhiễu này vẫn chưa có phương pháp hiệu quả.
  + **Tư thế đứng của đối tượng (nghiêng, xoay,…):** tư thế của đối tượng sẽ xác định thông tin của đối tượng đó. Việc tư thế thay đổi quá lớn sẽ làm thay phần lớn thông tin về đối tượng, dẫn đến kết quả nhận dạng sai.
  + **Trang phục của đối tượng:** Kết quả nhận dạng có thể bị ảnh hưởng lớn nếu như đối tượng có các trang phục khác biệt so với mẫu như đeo kính, đội mũ,…

**4.2. Chương trình thực nghiệm**

Chúng ta có thể kỳ vọng ngày càng nhiều hơn các doanh nghiệp áp dụng cá nhân hóa trải nghiệm vào dịch vụ của họ. Công nghệ nhận diện khuôn mặt và trí tuệ nhân tạo sẽ giúp điều này trở nên dễ dàng hơn bao giờ hết. Bằng việc thu thập và phân tích độ tuổi, giới tính, sở thích, lịch sử mua sắm hệ thống sẽ tự động đưa ra những nội dung và dịch vụ phù hợp với khác hàng giúp tăng trải nghiệm khách hàng tốt hơn.

**4.2.1 Nhận xét và kết quả**

Sau khi thực hiện thử nghiệm Nhận diện mắt kết quả thu được khi đặt câu hỏi là tỷ lệ phản hồi nằm ở mức trung bình khá, tuy nghiên đã giải quyết được một số vấn đề:

- Vấn đề phát sinh không hiện cảm xúc như là mừng vui giận.

- vấn đề không có data.db trong sql

- với không nhận haar cade face và eye

**CHƯƠNG 5: KẾT LUẬN**

Hệ thống nhận diện khuôn mặt và cảm xúc mắt đang là lĩnh vực mà các công ty công nghệ tập trung đầu tư nghiên cứu mạnh mẽ và ngày càng được phát triển. Ứng dụng hệ thống trợ lý ảo đang dần được đưa vào đời sống, được tích hợp từ những hệ thống như y tế, dạy học... Qua tiểu luận, em đã đạt được những kết quả nhất định.

Đầu tiên là tìm hiểu về mô hình cảm xúc như là trạng thái mắt, về cấu trúc hệ thống. Từ mô hình hệ thống trợ lý ảo giúp em hiểu được cấu tạo, thuật toán ứng dụng trong từng mô hình. Từ đó giúp em có thể đi sâu nghiên cứu từng mô đun trong hệ thống và ứng dụng thực tiễn là xây dựng một hệ thống nhận diện mắt phù hợp với mục đích của mình.

Thứ hai là từ việc tạo ra nhận diện mắt cảm xúc đã giúp em có những kiến thức, độ hiểu biết sâu hơn về lập trình Python, cách xử dụng thư viện, khai thác dữ liệu. Từ đó đúc kết được kiến thức để có thể ứng dựng thực tế trong công việc và mục đích huấn luyện một hệ thống nhận diện.

Thứ ba là từ việc xây dựng nhận diện, em có thể xây dựng nhiều tính năng với nhiều mục đích khác nhau. Tuỳ theo yêu cầu và mục đích của thực tiễn có thể xây dựng được hệ thống nhận diện cái khác để đáp ứng yêu cầu người dùng.

Qua những kết quả đạt được ban đầu, em nhận thấy còn rất nhiều việc phải làm, cần phải tối ưu. Nhưng cách tiếp cận này ban đầu đã cho những kết quả rất tích cực và đúng đắn, có thể giải quyết được những vấn đề xây dựng và tính toán trong hệ thống nhận diện chính xác

Định hướng nghiên cứu tiếp theo, em sẽ tiếp tục làm mượt dữ liệu, để tạo ra các mô hình mới có khả năng trả lời sát với ngữ cảnh, đạt chất lượng cao hơn, giảm khả năng lảng tránh và đưa tính cá nhân vào trong đoạn camera để nhận diện. Từ đó áp dụng cho nhiều hệ thống nhận diện với những yêu cầu và mục đích xây dựng khác nhau.

**TÀI LIỆU THAM KHẢO**

* [1]Tkinter documentation] - <https://www.tutorialspoint.com/python/python_gui_programming.htm>
* [2]OpenCV documentation] - <https://opencv.org/>
* [3]PyMySQL documentation] - <https://pypi.org/project/PyMySQL/>
* [4]Python documentation] - <https://www.python.org/>
* [5]Python projects tutorial] - <https://www.youtube.com/channel/UCKJnF_GhwvmXqtGh9LePXpg>
* [6]DeepLearning\_by\_PhDScholar]-https://www.youtube.com/channel/UCUv49cJ3xwr1NXxl9qIJ7kA