

**ĐỀ TÀI:**

**ĐỒ ÁN MÔN HỌC 2**

**THIẾT KẾ VÀ THI CÔNG HỆ THỐNG BẢO MẬT ỨNG DỤNG XỬ LÝ ẢNH**

**GVHD: TS. Phạm Văn Khoa**

**SVTH: Nguyễn Phúc Kha 18119085**

**Hấu Trung Đạt 18119064**

**Nguyễn Thanh Nhân 18119103**

**NGÀNH CÔNG NGHỆ KỸ THUẬT MÁY TÍNH**

Hồ Chí Minh, tháng 2 năm 2022



**GVHD: TS. Phạm Văn Khoa**

**SVTH: Nguyễn Phúc Kha 18119085**

**Hấu Trung Đạt 18119064**

**Nguyễn Thanh Nhân 18119103**

**THIẾT KẾ VÀ THI CÔNG HỆ THỐNG BẢO MẬT ỨNG DỤNG XỬ LÝ ẢNH**

**ĐỀ TÀI:**

**NGÀNH CÔNG NGHỆ KỸ THUẬT MÁY TÍNH**

**ĐỒ ÁN MÔN HỌC 2**

Hồ Chí Minh, tháng 2 năm 2022

**NHẬN XÉT CỦA GIẢNG VIÊN HƯỚNG DẪN**

*Ký tên*

TS. Phạm Văn Khoa

**LỜI CÁM ƠN**

Nhóm em xin gửi lời cám ơn sâu sắc đến Thầy Phạm Văn Khoa – Giảng viên ngành Công nghệ kỹ thuật Máy tính đã tận tình giúp đỡ chúng em trong lựa chọn đề tài cũng như trong quá trình thực hiện đề tài. Trong quá trình thực hiện đồ án cũng đã xảy ra nhiều khó khăn, thiếu sót nhưng được sự hỗ trợ và góp ý của Thầy nên nhóm đã hoàn thành được đề tài. Một lần nữa nhóm em xin chân thành cám ơn Thầy!

Nhóm thực hiện đề tài

Nguyễn Phúc Kha

Hấu Trung Đạt

Nguyễn Thanh Nhân

**MỤC LỤC**

[**CHƯƠNG 1: GIỚI THIỆU 7**](#_Toc95685040)

[1.1. Giới thiệu. 7](#_Toc95685041)

[1.2. Mục tiêu đề tài. 7](#_Toc95685042)

[1.3. Phương pháp nghiên cứu. 8](#_Toc95685043)

[1.4. Đối tượng và phạm vi nguyên cứu. 8](#_Toc95685044)

[1.5. Bố cục báo cáo. 8](#_Toc95685045)

[**CHƯƠNG 2: CƠ SỞ LÝ THUYẾT 10**](#_Toc95685046)

[2.1. Giới thiệu về hệ thống xử lý ảnh 10](#_Toc95685047)

[2.1.1. Phần thu nhận ảnh (Image Acquisition) 11](#_Toc95685048)

[2.1.2. Tiền xử lý (Image Processing) 11](#_Toc95685049)

[2.1.3. Phân đoạn (Segmentation) hay phân vùng ảnh 11](#_Toc95685050)

[2.1.4. Biểu diễn ảnh (Image Representation) 12](#_Toc95685051)

[2.1.5. Nhận dạng và nôi suy ảnh (Image Recognition and Interpretation) 12](#_Toc95685052)

[2.1.6. Cơ sơ tri thức 12](#_Toc95685053)

[2.1.7. Các thành phần cơ bản của hệ thống xử lý ảnh 12](#_Toc95685054)

[2.2. Những vấn đề cơ bản trong xử lý ảnh 13](#_Toc95685055)

[2.2.1. Điểm ảnh (Picture Element) 13](#_Toc95685056)

[2.2.2. Độ phân giải của ảnh 13](#_Toc95685057)

[2.2.3. Mức xám của ảnh 14](#_Toc95685058)

[2.2.4. Định nghĩa ảnh số 14](#_Toc95685059)

[2.3. Giới thiệu về thuật toán Local Binary Patterns (LBPs) 15](#_Toc95685060)

[2.4. Giới thiệu về các linh kiện sử dụng trong thiết kế, thi công đề tài 16](#_Toc95685061)

[2.4.1. Board Arduino Uno R3 ATmega328 16](#_Toc95685062)

[2.4.2. Tổng quan về động cơ Servo 17](#_Toc95685063)

[2.4.3. Tổng quan về Buzzer 18](#_Toc95685064)

[**CHƯƠNG 3: THIẾT KẾ HỆ THỐNG 19**](#_Toc95685065)

[3.1. Sơ đồ và thiết kế từng khối 19](#_Toc95685066)

[3.1.1. Sơ đồ khối 19](#_Toc95685067)

[3.1.2. Thiết kế từng khối 19](#_Toc95685068)

[3.1.3. Sơ đồ nguyên lý 21](#_Toc95685069)

[3.2. Lưu đồ hoạt động. 22](#_Toc95685070)

[3.2.1. Lưu đồ hoạt động của hệ thống nhận diện khuôn mặt. 22](#_Toc95685071)

[3.2.2. Lưu đồ hoạt động của hệ thống huấn luyện 23](#_Toc95685072)

[3.2.3. Lưu đồ hoạt động của toàn hệ thống bảo mật 24](#_Toc95685073)

[**CHƯƠNG 4: KẾT QUẢ - ĐÁNH GIÁ 25**](#_Toc95685074)

[4.1. Kết quả mô hình thi công. 25](#_Toc95685075)

[4.2. Đánh giá. 28](#_Toc95685076)

[**CHƯƠNG 5: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN 29**](#_Toc95685077)

[5.1. Kết luận. 29](#_Toc95685078)

[5.2. Hướng phát triển. 29](#_Toc95685079)

[**TÀI LIỆU THAM KHẢO 31**](#_Toc95685080)

[**PHỤ LỤC 32**](#_Toc95685081)

# **CHƯƠNG 1: GIỚI THIỆU**

* 1. **Giới thiệu.**

Ngày nay nhiều thiết bị thông minh ra đời nó có thể thay cho một chìa khoá để mở cửa hoặc có thể thay thế cho con người để điều khiển vật dụng thông qua một vài cử chỉ, giọng nói và đặc biệt là từ hình ảnh của con người. Hệ thống bảo mật thông tin cũng đang là một trong những sản phẩm phổ biến và được giới công nghệ rất quan tâm. Các hệ thống bảo mật càng ngày được tích hợp thêm nhiều các thiết bị điện tử để tăng cường độ bảo mật tối đa và tạo nên sự tin tưởng cho người dùng. Vì vậy, các hệ thống bảo mật dần được ứng dụng nhiều hơn vào thực tiễn đời sống với mục đích đơn giản nhất chỉ là để bảo vệ các tài sản như tiền bạc hoặc các vật dụng có giá trị cao…

Với xu thế áp dụng mô hình tự động hoá ngày càng nhiều tại Việt Nam, việc áp dụng mô hình khoá cửa bảo mật sẽ có nhiều ưu điểm hơn so với mô hình bảo mật truyền thống như:

* Tiết kiệm chi phí thuê nhân công
* Dễ dàng quản lý tài sản
* Gây khó khăn hơn trong việc cố ý bẻ khoá của những kẻ xấu

Thấy được những lợi ích mà hệ thống bảo mật đem lại nên nhóm sinh viên chúng em quyết định thực hiện đề tài “Thiết kế và thi công hệ thống bảo mật ứng dụng xử lý ảnh”. Nhóm thực hiện đề tài mong muốn tạo ra một hệ thống bảo mật sử dụng xử lý ảnh để nhận dạng khuôn mặt cùng với board Arduino để điều khiển việc đóng mở cửa.

* 1. **Mục tiêu đề tài.**

Tạo được hệ thống bảo mật bằng phương pháp ứng dụng xử lý ảnh để nhận diện khuôn mặt bằng ngôn ngữ Python với thư viện chính là OpenCV cùng những mục tiêu được đặt ra trong đề tài:

* Thiết kế mô hình cửa tự động
* Thiết kế cơ sở dữ liệu để lưu hình ảnh và huấn luyện
* Nhận diện khuôn mặt bằng camera
* So sánh khuôn mặt với tập ảnh có sẵn trong cơ sở dữ liệu
  1. **Phương pháp nghiên cứu.**

Để giải quyết các vấn đề, nhóm thực hiện các phương pháp nghiên cứu như sau:

* Tìm hiểu phương pháp rút trích đặc trưng khuôn mặt Local Binary Pattern (LBP).
* Tìm hiểu các lý thuyết có liên quan như ngôn ngữ Python và thư viện OpenCV.
* Phương pháp thực nghiệm: xây dựng chương trình nhận diện khuôn mặt để điều khiển việc đóng mở cửa và hiển thị kết quả lên màn hình
  1. **Đối tượng và phạm vi nguyên cứu.**
* Đối tượng nguyên cứu
* Các phương pháp, giải thuật phát hiện và nhận dạng khuôn mặt người trên ảnh.
* Ngôn ngữ lập trình Python, thư viện OpenCV
* Phạm vi nguyên cứu

Trong đề tài này, nhóm nghiên cứu tập trung vào giải thuật nhận diện khuôn mặt để đóng mở khóa cửa. Ban đầu tìm và cắt các khuôn mặt có trong ảnh thu về từ camera. Sau đó nhận diện xem khuôn mặt đó là ai trong cơ sở dữ liệu đã được huấn luyện. Sau khi nhận diện xong sẽ xuất tín hiệu điều khiển Servo đóng mở cửa

Do các khó khăn của bài toán nhận diện khuôn mặt như: tư thế góc chụp, các thành phần xuất hiện trên khuôn mặt, độ sáng của ảnh, … Vì thế trong luận văn này, nhóm nghiên cứu đưa ra các ràng buộc và giả định sau để làm giảm độ phức tạp của bài toán nhận diện khuôn mặt:

* Người được nhận diện phải nhìn thằng vào camera hoặc góc nhìn so với camera không đáng kể (nhỏ hơn 15o)
* Camera có độ phân giải kém sẽ ảnh hưởng đến kết quả xử lý không được tốt nhất có thể
* Khuôn mặt không bị che khuất
* Khoảng cách người nhận diện đến camera trong khoảng 40 – 60 cm
  1. **Bố cục báo cáo.**

Báo cáo đề tài gồm 5 chương:

* Chương 1: Giới thiệu sơ lược về lý do chọn đề tài, mục tiêu đề ra, phương pháp và phạm vi nguyên cứu.
* Chương 2: Giới thiệu về lý thuyết liên quan tới các vấn đề cần giải quyết trong đề tài.
* Chương 3: Thiết kế hệ thống (giới thiệu về ý tưởng thiết kế, thiết kế phần cứng và phần mềm).
* Chương 4: Kết quả đạt được của đề tài
* Chương 5: Kết luận tổng kết lại các vấn đề đã giải quyết được của đề tài và hướng phát triển xa hơn cho đề tài.

# **CHƯƠNG 2: CƠ SỞ LÝ THUYẾT**

1. **Giới thiệu về hệ thống xử lý ảnh**

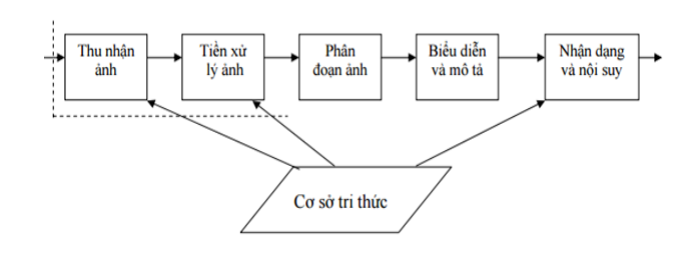
Xử lý ảnh là một lĩnh vực mang tính khoa học và công nghệ. Nó là một ngành khoa học mới mẻ so với nhiều ngành khoa học khác nhưng tốc đó phát triển của nó rất nhanh, kích thích các trung tâm nghiên cứu, ứng dụng, đặc biệt là máy tính chuyên dụng riêng cho nó.

Xử lý ảnh là kỹ thuật áp dụng trong việc tăng cường và xử lý các ảnh thu nhận từ các thiết bị như camera, webcam… Do đó, xử lý ảnh đã được ứng dụng và phát triển trong rất nhiều lĩnh vực quan trọng như:

* Trong lĩnh vực quân sự: xử lý và nhận dạng ảnh quân sự.
* Trong lĩnh vực giao tiếp người máy: nhận dạng ảnh, xử lý âm thanh, đồ họa.
* Trong lĩnh vực an, bảo mật: nhận diện khuôn mặt người, nhận diện vân tay, mẫu mắt, …
* Trong lĩnh vực giải trí: trò chơi điện tử.
* Trong lĩnh vực y tế: Xử lý ảnh y sinh, chụp X quang, MRI,…

Các phương pháp xử lý ảnh bắt đầu từ các ứng dụng chính: nâng cao chất lượng và phân tích ảnh. Ứng dụng đầu tiên được biết đến là nâng cao chất lượng ảnh báo được truyền từ London đến New York từ những năm 1920. Vấn đề nâng cao chất lượng ảnh có liên quan tới phân bố mức sáng và độ phân giải của ảnh. Việc nâng cao chất lượng ảnh được phát triển vào khoảng những năm 1955. Điều này có thể giải thích được vì sau thế chiến thứ hai, máy tính phát triển nhanh tạo điều kiện cho quá trình xử lý ảnh số được thuận lợi hơn. Năm 1964, máy tính đã có khả năng xử lý và nâng cao chất lượng ảnh từ mặt trăng và vệ tinh Ranger 7 của Mỹ bao gồm: làm nổi đường biên, lưu ảnh. Từ năm 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 đâị và cải tiến, các công cụ nén ảnh ngày càng được áp dụng rộng rãi và thu được nhiều kết quả khả quan hơn.

Sau đây, ta sẽ xét các bước cần thiết trong quá trình xử lý ảnh. Đầu tiên, ảnh tự nhiên từ thế giới bên ngoài được thu nhận qua các thiết bị thu (như Camera, máy chụp ảnh). Trước đây, ảnh thu qua Camera là các ảnh tương tự (loại Camera ống kiểu CCIR). Gần đây với sự phát triển của công nghệ, ảnh màu hoặc đen trắng được lấy ra từ Camera, sau đó nó được chuyển trực tiếp thành ảnh số tạo thuận lợi cho xử lý tiếp theo. Mặt khác ảnh có thể được quét từ vệ tinh chụp trực tiếp bằng máy quét ảnh.



Hình 2.1 Các bước cơ bản trong xử lý ảnh

1. Phần thu nhận ảnh (Image Acquisition)

Ảnh có thể nhận qua camera mà hoặc trắng đen. Thường ảnh nhận qua camera là ảnh tương tự (loại camera ống chuẩn CCIR với tần số 1/25, mỗi ảnh 25 dòng), cũng có loại camera đã số hóa (như loại CCD – Change Coupled Device) là loại photodiot tạo cường độ sáng tại mỗi điểm ảnh.

Camera thường dùng là loại quét dùng: ảnh tạo ra có dạng hai chiều. Chất lượng ảnh thu nhận được phụ thuộc vào thiết bị thu, vào môi trường (ánh sáng, phong cảnh).

1. Tiền xử lý (Image Processing)

Sau bộ thu nhận, ảnh có thể nhiễu độ 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 độ tương phản để làm ảnh rõ nét hơn.

1. Phân đoạn (Segmentation) hay phân vùng ảnh

Phân vừng ảnh là tách một ảnh ban đầu vào thành các vùng thành phần để biểu diễn phân tích, nhận diện ảnh. Ví dụ: để nhận dạng chữ (hoặc mã vạch) trên phong bì thư cho mục đích phân loại bưu phẩm, cần chia các câu chữ về địa chỉ hoặc tên người thành các từ, các chữ, các số (hoặc các vạch) riêng biệt để nhận dạng. Đây là phần phức tạp khó khăn nhất trong xử lý ảnh và cũng dễ gây lỗi, làm mất độ chính xác của ảnh. Kết quả nhận dạng ảnh phụ thuộc rất nhiều vào công đoạn này.

1. Biểu diễn ảnh (Image Representation)

Đầu ra ảnh sau phân đoạn chứa các điểm ảnh của vùng ảnh (ảnh đã phân đoạn) cộng với mã liên kết ở 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. Việc chọn các tính chất để thể hiện ảnh gọi là trích chọn đặc trưng( Feature Extration) gắn với việc tách các đặc tính của ảnh dưới dạng các thông tin định lượng hoặc làm cơ sở để phân biệt lớp đối tượng này với đối tượng khác trong phạm vi ảnh nhận được. Ví dụ: trong nhận dạng ký tự trên phong bì thư, chúng ta miêu tả các đặc trưng của từng ký tự giúp phận biệt ký tự này với ký tự khác.

1. Nhận dạng và nôi suy ảnh (Image Recognition and Interpretation)

Nhận dạng ảnh là quá trình xác định ảnh. Quá trình này thường thu được bằng cách so sánh với mẫu chuẩn đã được học (hoặc lưu) từ trước. Nội suy là phán đoán theo ý nghĩa trên cơ sở nhận dạng. Ví dụ: một loạt chữ số và nét gạch ngang trên phong bì thư có thể nội suy thành mã điện thoại. Có nhiều cách phân loại ảnh khác nhau về ảnh. Theo lý thuyết về nhận dạng, các mô hình toán học về ảnh được phân theo hai loại nhận dạng ảnh cơ bản:

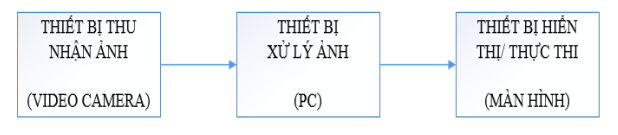
* Nhận dạng theo tham số.
* Nhận dạng theo cấu trúc.

Một số đối tượng nhận dạng khá phổ biến hiện nay được áp dụng trong khoa học và công nghệ là: nhận dạng ký tự (chữ in, chữ viết tay, chữ ký điện tử), nhận dạng văn bản (Text), nhận dạng vân tay, nhận dạng mã vạch, nhận dạng mặt người,…

1. Cơ sơ tri thức

Ảnh là một dối tượng khá phức tạp về đường nét, độ sáng tối, dung lượng điểm ảnh, môi trường để thu ảnh phong phú kéo theo nhiều. Trong nhiều khâu xử lý và phân tích ảnh ngoài việc đơn giản hóa các phương pháp toán học đảm bảo tiện lợi cho xử lý, người ta mong muốn bắt chước quy trình tiếp nhận và xử lý ảnh theo cách của con người. Trong các bước xử ly đó, nhiều khâu hiện nay đã xử lý theo phương pháp trí tuệ con người.

1. Các thành phần cơ bản của hệ thống xử lý ảnh



Hình 2.2 Các thành phần chính của hệ thống xử lý ảnh

Một hệ thống xử lý ảnh cơ bản có thể gồm:máy tính các nhân kèm theo vi mạch chuyển đổi đồ họa VGA hoặc SVGA. Nếu điều kiện cho phép, nên có một hệ thống như hình 2.2, bao gồm một máy tính PC kem theo thiết bị xử lý ảnh, nối với cổng vào của thiết bị thu nhận ảnh là một video camera và cổng ra nối với một màn hình.

Khi hệ thống này thực hiện một quá trình xử lý ảnh, đầu tiên, ảnh sẽ được nhận qua thiết bị thu nhận ảnh. Ảnh này có thể là ảnh đen trắng hoặc ảnh màu, chất lượng ảnh tùy thuộc vào chất lượng của camera cũng như các yếu tố của môi trường. Sau đó, ảnh sẽ được đưa qua thiết bị xử lý ảnh, ảnh sau khi được xử lý sẽ được đưa ra thiết bị hiển thị/ thực thi để xuất kết quả cuối cùng (hiển thị kết quả ra màn hình hoặc thực thi chức năng của thiết bị: mở cửa, cảnh báo,…).

1. **Những vấn đề cơ bản trong xử lý ảnh**
2. Điểm ảnh (Picture Element)

Gốc của ảnh (ảnh tự nhiên) là ảnh liên tục về không gian và độ sáng. Đề xử lý bằng máy tính (số), ảnh cần phải được số hóa. Số hóa ảnh là sự biến đổi gần đúng một ảnh liên tục thành một tập điểm phù hợp với ảnh thật về vị trí (không gian) và độ sáng (mức sáng). Khoảng cách giữa các diểm ảnh đó được thiết lập sao cho mắt người không phân biệt được ranh giới giữa chúng. Mỗi một điểm như vậy được gọi là điểm ảnh (PEL: Picture Element) hay gọi tắt là Pixel. Trong khuôn khổ ảnh hai chiều, mỗi pixel ứng với cặp tọa độ (x,y).

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 giữa các điểm ảnh đó được chọn thích hợp sao cho mắt người cảm nhận sự liên tục về không gian và mức xám (hoặc màu) của ảnh số gần như ảnh thật. Mỗi phần tử trong ma trận được gọi là một phần tử ảnh.

1. Độ phân giải của ảnh

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

Theo định nghĩa, khoảng cách giữa các điểm ảnh phải được chọn sao cho mắt người vẫn thấy được sự liên tục của ảnh. Việc lựa chọn khoảng cách thích hợp tạo nên một mật độ phân bổ, đó chính là độ phân giải và được phân bố theo trục x và y trong không gian hai chiều.

Ví dụ: Độ phân giải của ảnh trên màn hình CGA (Color Graphic Adaptor) là một lưới điểm theo chiều ngang màn hình: 320 điểm chiều dọc nhân với 200 điểm ảnh (320 x 200). Rõ ràng cùng màn hình CGA 12 inch ta nhận thấy mịn hơn màn hình CGA 17 inch độ phân giải 320 x 200. Lý do: cùng một mật độ (độ phân giải) nhưng diện tích màn hình rộng hơn thì độ mịn (liên tục của các điểm) kém hơn.

1. Mức xám của ảnh

Một điểm ảnh (pixel) có hai đặc trưng cơ bản là vị trí (x,y) của điểm ảnh và độ xám của nó. Dưới đây chúng ta xem xét một số khái niệm và thuật ngữ thường được dùng trong xử lý ảnh.

Mức xám của điểm ảnh là cường độ sáng của nó được gán bằng giá trị số tại điểm đó.

* Ảnh trắng đen: là ảnh có hai màu đen, trắng (không chứa màu khác) với mức xám ở các điểm ảnh có thể khác nhau.
* Ảnh nhị phân ảnh: chỉ có hai mức đen trắng phân biệt tức dùng 1 bit mô ta 2 mức khác nhau. Nói cách khác: mỗi điểm ảnh của ảnh nhị phân chỉ có thể là 0 hoặc 1.
* Ảnh màu: trong khuôn khổ lý thuyết ba màu (Red, Blue, Green) để tạo nên thế giới đa màu, người ta thường dùng 3 byte để mô tả mức màu, khi đó các giá trị màu: 256 x 256 x 256 hay 28 x 3 = 224 = 16,7 triệu màu. Vì khi biểu diễn dưới dạng số, các giá trị RGB thường được ghi bằng 3 cặp số nguyên giữa 0 và 255, mỗi số đại diện cho cường độ của 3 màu: Red (255, 0, 0) Blue (0, 255, 0) Green (0, 0, 255).

1. Định nghĩa ảnh số

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

1. **Giới thiệu về thuật toán Local Binary Patterns (LBPs)**

Phương pháp LBPs for face recognition được giới thiệu vào năm 2006 và vẫn đang được sử dụng rất rộng rãi.

Đầu tiên, các bức ảnh sẽ được chia thành 7 x 7 = 49 vùng bằng nhau, tại mỗi vùng này chúng ta sẽ tính được một LBP histogram.



Hình 2.3 Bức ảnh được chia thành 7 x 7 = 49 vùng

Bằng cách chia bức ảnh thành nhiều vùng và tính LBP histogram cho mỗi vùng, chúng ta đã có được thông tin về vị trí mắt, mũi, miệng trên khuôn mặt (locality and spatial infomation). Các thông tin này còn cho phép chúng ta áp dụng trọng số lên histogram của các vùng chứa các đặc trưng quan trọng cho phép phân biệt giữa các khuôn mặt, mô tả như hình dưới.



Hình 2.4 Các vùng quan trọng trên khuôn mặt

Như ở hình trên, histogram của vùng hình ảnh chứa đôi mắt và miệng sẽ được nhân với trọng số lớn hơn các vùng có màu tối hơn. Cụ thể, ảnh có 49 grid cells, mỗi cell sẽ có trọng số riêng của nó khi tính feature chung của khuôn mặt:

* LBP histogram cho cell trắng (như mắt) có trọng số lớn hơn 4 lần các cell khác. Đơn giản chỉ cần nhân 4 lần LBP histogram của cell trắng (có tính đến scaling và normalization.
* Cell xám nhạt (ligh gray cell) - vùng tai, miệng có trọng số 2
* Cell xám đậm (dark gray cell) - vùng má trong và trán có trọng số 1
* Các cell đen còn lại như mũi, vùng má ngoài… bị bỏ qua, không đóng góp gì vào feeature chung.

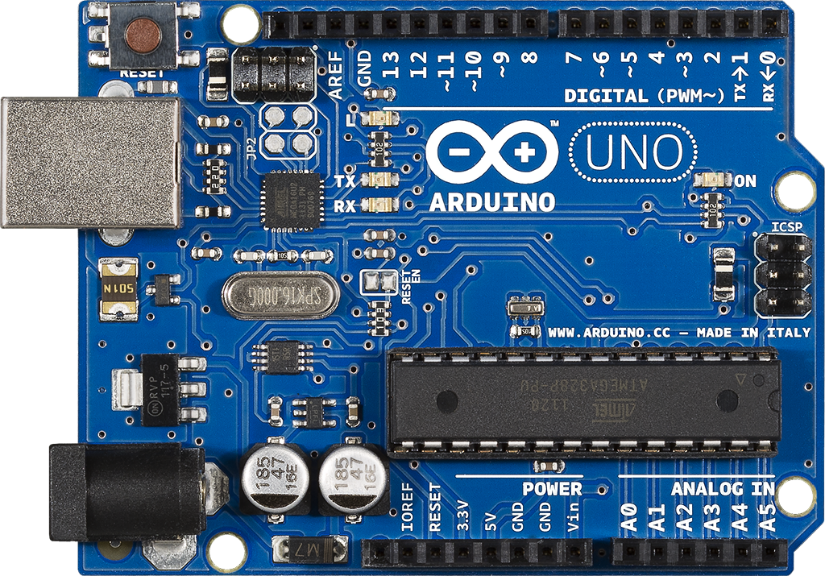
Cuối cùng, histogram của 49 vùng hình ảnh sẽ được ghép lại với nhau thành một Feature Vector duy nhất. Lúc này chúng ta có thể sử dụng k-NN classifier (sử dụng χ2 distance) với k=1 để phân loại các khuôn mặt.

1. **Giới thiệu về các linh kiện sử dụng trong thiết kế, thi công đề tài**
2. Board Arduino Uno R3 ATmega328

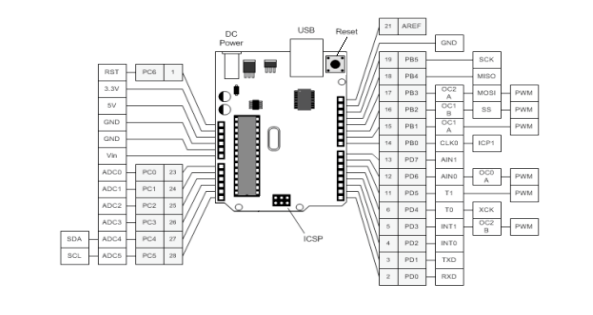
Arduino Uno R3 - ATmega328 là dòng Arduino thế hệ thứ 3, cũng giống như các phiên bản trước đây giúp người dùng dễ dàng tiếp cận với lập trình để tạo ra phần cứng có những tính năng mong muốn một cách nhanh chóng với chi phí hợp lý.

Nó là một bo mạch vi điều khiển dựa trên ATmega328, có 14 chân đầu vào / đầu ra kỹ thuật số (trong đó 6 chân có thể được sử dụng làm đầu ra PWM), 6 đầu vào analog, bộ cộng hưởng 16 MHz, kết nối USB, giắc cắm nguồn, đầu cắm ICSP và nút reset. Mạch chứa mọi thứ cần thiết để hỗ trợ vi điều khiển, chỉ cần kết nối thiết bị với máy tính bằng cáp USB hoặc cấp nguồn từ 7 đến 12VDC để bắt đầu.

Được giới thiệu vào năm 2005, Những nhà thiết kế của Arduino cố gắng mang đến một phương thức dễ dàng, không tốn kém cho những người yêu thích, sinh viên và giới chuyên nghiệp để tạo ra những nhiết bị có khả năng tương tác với môi trường thông qua các cảm biến và các cơ cấu chấp hành. Những ví dụ phổ biến cho những người yêu thích mới bắt đầu bao gồm các robot đơn giản, điều khiển nhiệt độ và phát hiện chuyển động.



Hình 2.5 Board Arduino Uno R3 – Atmega323



Hình 2.6 Sơ đồ chân kết nối Arduino Uno R3 – Atmega323

1. Tổng quan về động cơ Servo

Servo là một dạng động cơ điện đặc biệt. Không giống như động cơ thông thường cứ cắm điện vào là quay liên tục, servo chỉ quay khi được điều khiển (bằng xung PPM) với góc quay nằm trong khoảng bất kì từ 0o – 180o . Mỗi loại servo có kích thước, khối lượng và cấu tạo khác nhau. Có loại thì nặng chỉ 9g (chủ yếu dùng trên máy bay mô mình), có loại thì sở hữu một momen lực bá đạo (vài chục Newton/m), hoặc có loại thì khỏe và nhông sắc chắc chắn,...



Hình 2.7 Động cơ Servo

1. Tổng quan về Buzzer

Buzzer còn gọi là loa mini hay còi báo. Nó dùng để phát ra các âm thanh. Nó gồm 2 chân: Chân GND nối GND arduino và chân tín hiệu nối pin arduino. Ta sẽ lập trình để nó phát ra cao độ hay cường độ hay âm sắc gì đó thay đổi (Mù âm nhạc) từ đó tạo ra những bản nhạc thú vị mang phong cách arduino.



Hình 2.8 Buzzer

# **CHƯƠNG 3: THIẾT KẾ HỆ THỐNG**

1. **Sơ đồ và thiết kế từng khối**
   1. Sơ đồ khối

**KHỐI NGUỒN**

**KHỐI XỬ LÝ TRUNG TÂM**

**KHỐI THIẾT BỊ NGOẠI VI**

Khối nguồn: Nguồn từ laptop cung cấp điện cho toàn hệ thống

Khối xử lý trung tâm: Xử lý dữ liệu từ việc nhận diện hình ảnh người dùng ở máy tính gửi về để điều khiển khối ngoại vi

Khối thiết bị ngoại vi: Nhận tín hiệu từ khối xử lý trung tâm để hoạt động

* 1. Thiết kế từng khối

Khối xử lý trung tâm

Hiện nay trên thị trường có rất nhiều dòng vi điều khiển khác nhau như PIC, AVR, 8051, Raspberry, Arduino...Tất cả đều có thể đáp ứng được yêu cầu đặt ra nhưng nhóm chọn Arduino Uno R3 ATmega328 vì nó có những ưu điểm sau:

* Giá thành rẻ, dễ sử dụng, là module hoàn chỉnh sử dụng vi điều khiển AVR.
* Kích thước nhỏ gọn.
* Là dòng vi điều khiển mã nguồn mở, có nhiều thư viện hổ trợ cho các module chức năng khác nhau, trình biên dịch đơn giản, dễ sử dụng.

Thông số kỹ thuật:

* Vi điều khiển chính: ATmega328
* Điện áp hoạt động:5VDC
* Điện áp vào: 7~12VDC
* Điện áp vào giới hạn: 6~20VDC
* Số chân Digital: 14 (6 chân PWM)
* Số chân vào Analog: 6
* Dòng DC trên mỗi chân: 40mA
* Dòng DC trên chân 3.3V: 50mA
* Bộ nhớ Flash: 32 KB (0.5KB dùng cho bootloader)
* SRAM: 2 KB - EEPROM: 1KB
* Tần số xung clock: 16 MHz

Khối thiết bị ngoại vi

* Động cơ servo MG90S

Động cơ MG90S thường được sử dụng trong các mô hình nhỏ hoặc các cơ cấu kéo không cần đến lực nặng, động cơ có tích hợp sẵn Driver điều khiển động cơ bên trong nên có thể dễ dàng điều khiển góc quay bằng phương pháp điều độ rộng xung PWM.

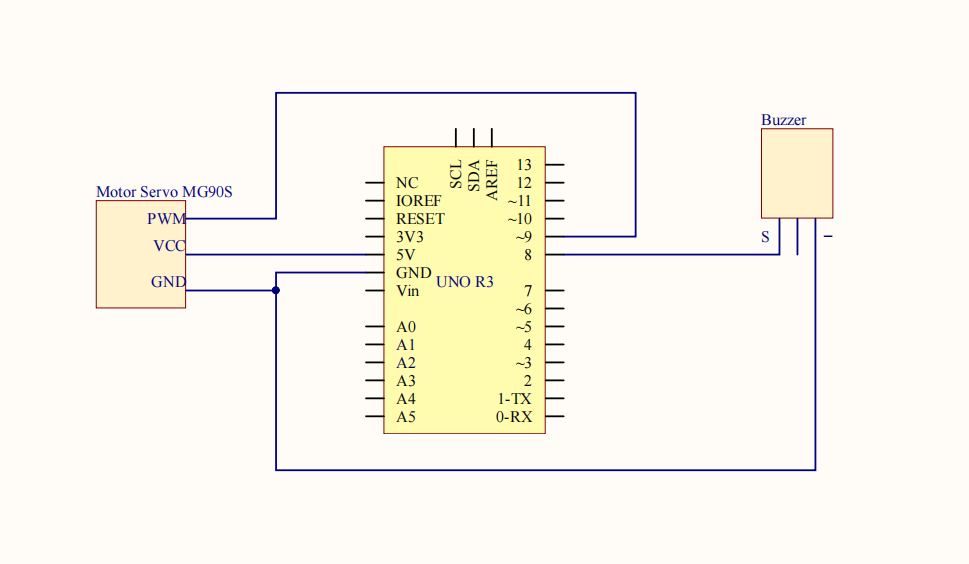
Thông số kỹ thuật:

* Kích thước 22.8 x 12.2 x 28.5 mm
* Stall Torque 1.8 kg.cm (4.8V); 2.2 kg.cm (6V)
* Nhiệt độ hoạt động 0 - 55°C
* Tốc độ 0.1 giây / 60° (4.8V), 0,08 giây / 60° (6V)
* Điện áp hoạt động 4.8 → 6 V dc
* Buzzer KY-012

KY-012 là một mạch buzzer lý tưởng để tăng thêm âm thanh cho dự án và tương thích khi hoạt động với các bộ vi điều khiển như Arduino. Mạch buzzer này bao gồm một bộ rung áp điện hoạt động, nó tạo ra âm thanh có tần số gần 2.5kHz khi tín hiệu ở mức cao.

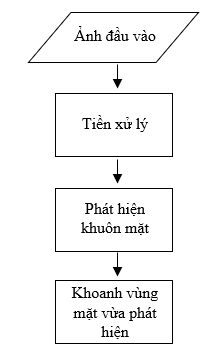
Thông số kỹ thuật:

* Điện áp hoạt động : 3.5V ~ 5.5V
* Dòng hoạt động: 30mA / 5VDC
* Tần số cộng hưởng 2500Hz ± 300Hz
* Đầu ra âm thanh tối thiểu 85Db
* Nhiệt độ làm việc: -20°C ~ 70°C
  1. Sơ đồ nguyên lý



Hình 3.1 Sơ đồ nguyên lý kết nối Arduino với servo và buzzer

1. **Lưu đồ hoạt động.**
2. Lưu đồ hoạt động của hệ thống nhận diện khuôn mặt.



Hình 3.2 Lưu đồ hoạt động hệ thống nhận diện khuôn mặt

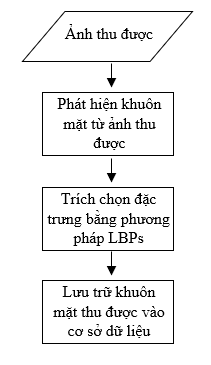
Khối ảnh đầu vào: Dữ liệu hình ảnh thu được từ camera

Khối tiền xử lý: Giúp cải thiện và nâng cao chất lượng của hình ảnh để hệ thống dễ xử lý hơn.

Khối nhận diện khuôn mặt người: phát hiện khuôn mặt người ban đầu với không gian RGB.

Khối hiển thị và khoanh vùng khuôn mặt vừa mới nhận diện được: Xuất kết quả đồng thời lưu lại kết quả khuôn mặt vừa thu được để tiến hành so sánh với cơ sở dữ liệu.

1. Lưu đồ hoạt động của hệ thống huấn luyện



Hình 3.3 Lưu đồ hoạt động của hệ thống huấn luyện

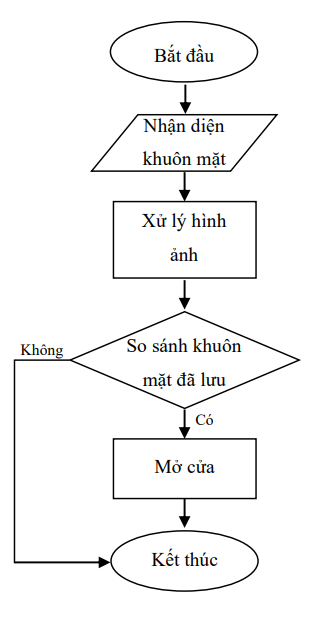
Khối ảnh thu được: Những ảnh được chọn làm tập mẫu.

Khối phát hiện khuôn mặt từ ảnh thu được: Phát hiện khuôn mặt người ban đầu với không gian RGB.

Khối trích chọc đặc trưng khuôn mặt bằng phương pháp LBPs: Dùng phương pháp LBPs để phát hiện khuôn mặt người đang ở phía trước camera.

Khối lưu trữ khuôn mặt thu được về cơ sở dữ liệu: Thiết lập các thống số lên khuôn mặt đã thu được và lưu lại về cơ sở dữ liệu (database).

1. Lưu đồ hoạt động của toàn hệ thống bảo mật



Hình 3.4 Lưu đồ hoạt động toàn hệ thống bảo mật

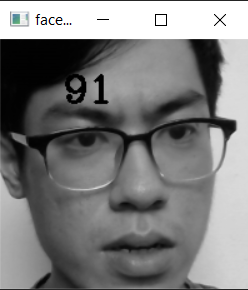
Khi hệ thống nhận diện hoạt động, người dùng đưa khuôn mặt cần quét trước camera, hệ thống nhận diện khuôn mặt sẽ thực hiện so sánh với cơ sở dữ liệu để đưa ra hướng giải quyết, nếu khuôn mặt có trong cơ sở dữ liệu thì servo hoạt động cửa mở ra.

# **CHƯƠNG 4: KẾT QUẢ - ĐÁNH GIÁ**

## **4.1. Kết quả mô hình thi công.**

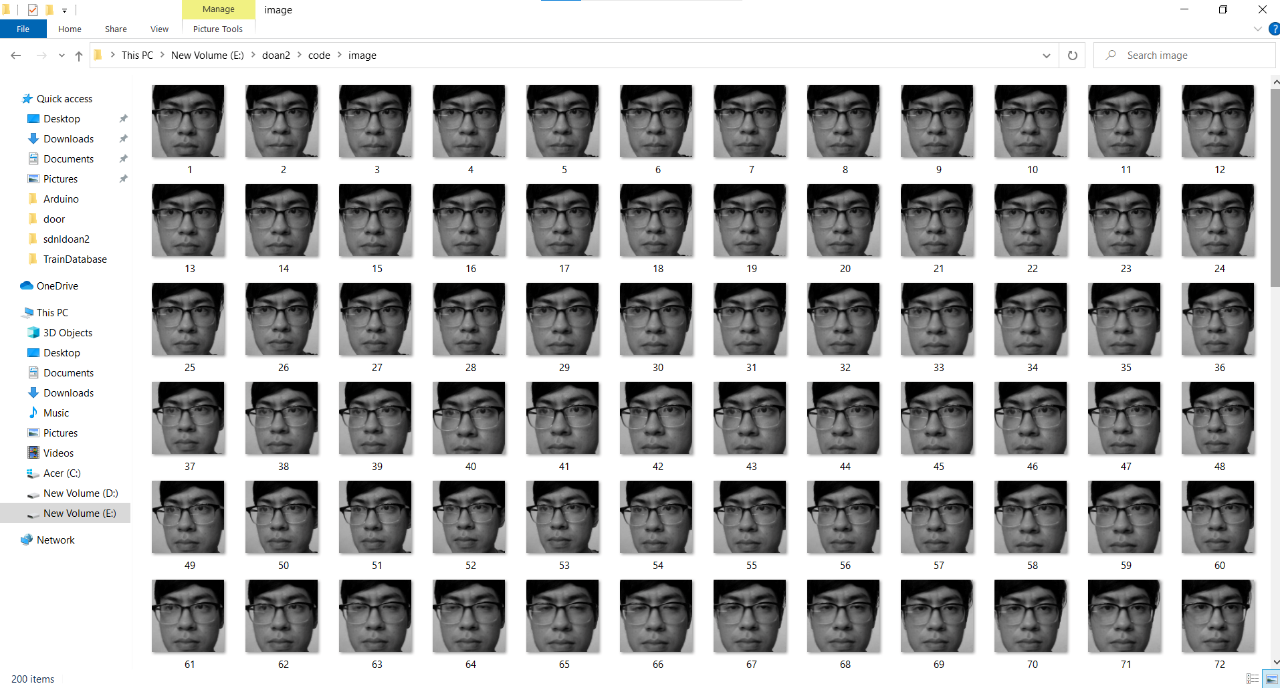
Sau quá trình tìm hiểu và thi công đề tài, nhóm đã hoàn thành và thực hiện được các kết quả sau:

Để bắt đầu sử dụng hệ thống bảo mật, camera sẽ tiến hành lấy mẫu ảnh lưu vào cơ sở dữ liệu (hình ảnh minh hoạ bên dưới là ảnh thứ 91 được lưu trong cơ sở dữ liệu sau khi lấy mẫu).



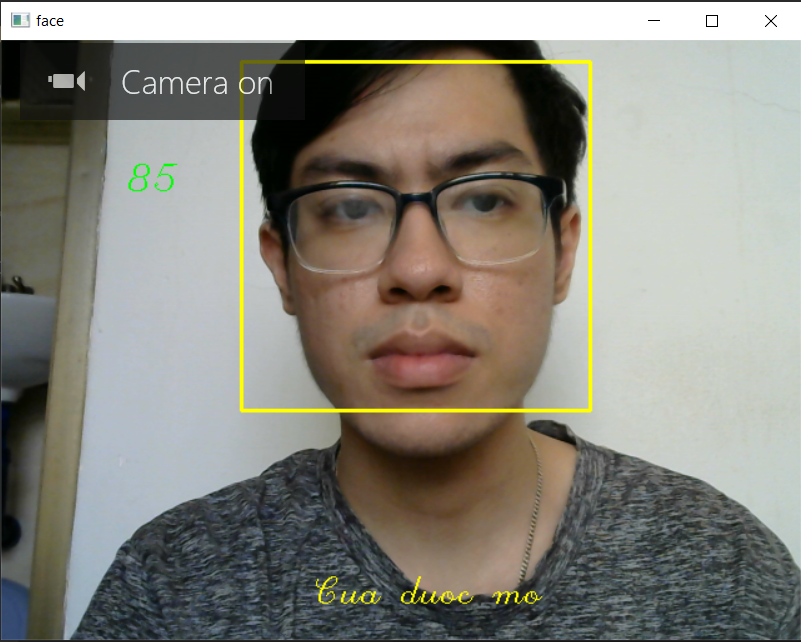
Hình 4.1 Ảnh chụp để làm dữ liệu cho hệ thống

Ảnh thu được từ camera sẽ được lưu lại trong cơ sở dữ liệu để tiến hành huấn luyện



Hình 4.2 Thư mục lưu trữ dữ liệu ảnh để huấn luyện

Khi hệ thống nhận diện hoạt động, khuôn mặt người dùng lúc này được chụp lại và hệ thống nhận diện khuôn mặt sẽ thực hiện so sánh với cơ sở dữ liệu để đưa ra hướng giải quyết, nếu khuôn mặt có trong cơ sở dữ liệu thì servo hoạt động mở cửa ra.

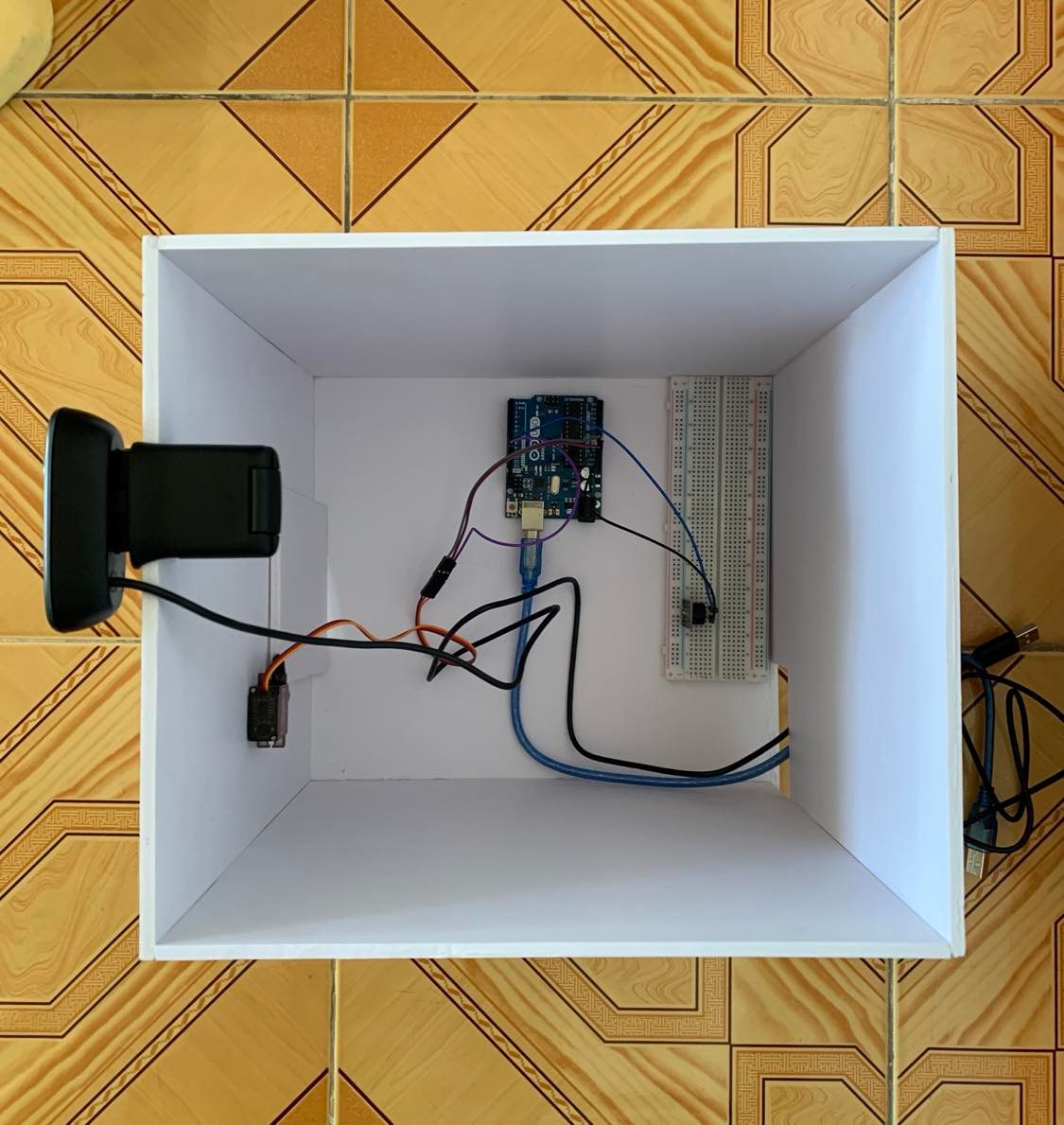


Hình 4.3 Hệ thống tiến hành nhận diện khuôn mặt

Hình ảnh thực tế mô hình sản phẩm:



Hình 4.4 Mô hình hệ thống bảo mật



Hình 4.5 Mô hình hệ thống bảo mật

## **4.2. Đánh giá.**

Sau khi vận hành hệ thống để tiến hành kiểm tra và đánh giá; nhìn chung, mô hình đã hoạt động ổn định, đạt yêu cầu so với mục tiêu đề ra, có thể làm việc liên tục. Hệ thống đảm bảo an toàn và bảo mật cho người dùng.

Tuy nhiên, để mô hình có thể áp dụng vào thực tế cần có sự cải tiến cần thiết về kích thức bộ xử lý cũng như sự chắc chắn của bộ điều khiển trước các hành động đột nhập, cố tình gây hư hỏng ngoài ý muốn.

# **CHƯƠNG 5: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN**

## **5.1. Kết luận.**

Sau thời gian tìm hiểu, nghiên cứu và thực hiện, nhóm đã hoàn thành đề tài “Thiết kế và thi công hệ thống bảo mật ứng dụng xử lý ảnh”. Qua quá trình thực hiện, nhóm đã đúc kết và học thêm được một số kỹ năng như:

* Biết cách viết chương trình của Python, kết hợp Python với môi trường OpenCV để đồng bộ cấu trúc chương trình
* Nâng cao được các kỹ năng bố trí, sắp xếp và hoàn thiện một mô hình
* Nâng cao khả năng làm việc nhóm cũng như các kỹ năng khác có liên quan (Word, Power Point, Altium…)

Kết quả cuối cùng mà nhóm thực hiện đã đáp ứng đầy đủ các nội dung mục tiêu ban đầu đề ra:

* Thiết kế mô hình cửa tự động
* Thiết kế cơ sở dữ liệu để lưu hình ảnh và huấn luyện
* Nhận diện khuôn mặt bằng camera
* So sánh khuôn mặt với tập ảnh có sẵn trong cơ sở dữ liệu thành công
* Giao tiếp và truyền dữ liệu thành công giữa Arduino Uno R3 với chuông và động cơ Servo.

Tuy nhiên, do sự hạn chế về kiến thức và thời gian thực hiện, nguồn tài liệu tham khảo chủ yếu thông qua internet nên đề tài không tránh khỏi sai sót và còn một số hạn chế:

* Vì điều kiện kinh phí và thời gian có hạn, nên nhóm chỉ thực hiện được mô hình tương đối hoàn chỉnh, tính thẩm mỹ không cao.
* Vẫn phải kết nối trực tiếp với laptop để thực hiện chương trình.
* Chương trình nhận diện khuôn mặt vẫn chưa thực sự hoàn chỉnh, còn nhiều sai sót khi vận hành.
* Hệ thống kết nối với ngoại vi vẫn chưa hoạt động ổn định.

## **5.2. Hướng phát triển.**

Có thể nâng cấp hệ thống bằng việc gửi dữ liệu lên internet để hệ thống có thể dễ dàng quản lý từ xa hơn.

Thiết kế thêm nguôn dự trữ dùng pin mặt trời và ác-quy để khi nguồn điện chính bị mất đi hệ thống vẫn có thể hoạt động bình thường.

Có thể nâng cao khả năng nhận dạng khuôn mặt bằng phương pháp nhận dạng bằng camera kép, một camera nhận dạng và một camera đo chiều sâu để nâng cao khả năng nhận dạng nhiều góc độ đồng thời tăng độ tin cậy của hệ thống.

# **TÀI LIỆU THAM KHẢO**

[1] T. Ahonen, A. Hadid, and M. Pietikäinen, “Face Recognition with Local Binary Patterns”, University of Oulu, May 2004.

[2] Mobile Computer Vision Team, “Facial Recognition using OpenCV”, Journal of Mobile, 2012.

[3] F. Parab, R. Shaikh, S. Sangle, N. Shaikh, “Optical Data Communication using Python and Arduino”, International Journal of Engineering Applied Sciences and Technology, March 2020.

# **PHỤ LỤC**

* **Mã nguồn chương trình Arduino:**

#include<Servo.h>

Servo myservo;

const int ledPin = 2;

char d;

int pos=0;

void setup()

{

Serial.begin(9600);

pinMode(5,OUTPUT);

myservo.attach(9);

myservo.write(0);

pinMode (ledPin, OUTPUT);

}

void loop()

{

if(Serial.available())

{

d=Serial.read();

}

if(d=='a')

{

Serial.print(d);

delay(300);

for(pos=0;pos<=90;pos+=3)

{

myservo.write(pos);

delay(20);

}

delay(5000);

for(pos=90;pos>=0;pos-=3)

{

myservo.write(pos);

delay(20);

}

}

if(d=='c')

{

tone(8,3000,3000);

delay (5000);

}

d="";

}

* **Mã nguồn lấy dữ liệu cho hệ thống**

import cv2

import numpy as np

face\_classifier = cv2.CascadeClassifier('haarcascade\_frontalface\_default.xml')

def face\_extractor(img):

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

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

if faces == ():

return None

for (x,y,w,h) in faces:

cropped\_face = img[y:y+h,x:x+w]

return cropped\_face

cap = cv2.VideoCapture(2)

count = 0

while True:

ret,frame = cap.read()

if face\_extractor(frame) is not None:

count += 1

face = cv2.resize(face\_extractor(frame),(200,200))

face = cv2.cvtColor(face,cv2.COLOR\_BGR2GRAY)

file\_name\_path = 'E:/doan2/code/image/'+str(count)+'.jpg'

cv2.imwrite(file\_name\_path,face)

cv2.putText(face,str(count),(50,50),cv2.FONT\_HERSHEY\_COMPLEX,1,(0,255,0),2)

cv2.imshow("face cropper",face)

else:

print('face not found')

pass

if cv2.waitKey(1)==13 or count==200:

break

cap.release()

cv2.destroyAllWindows()

print("collecting samples complete")

* **Mã nguồn huấn luyện và nhận diện khuôn mặt**

import cv2

import numpy as np

from os import listdir

from os.path import isfile,join

import serial

import time

import pyttsx3

from gtts import gTTS

from playsound import playsound

q=1

x=0

c=0

m=0

d=0

while q<=2:

data\_path = 'E:/doan2/code/image/'

onlyfiles = [f for f in listdir(data\_path) if isfile(join(data\_path,f))]

Training\_data, Lebels = [],[]

for i , files in enumerate(onlyfiles):

image\_path = data\_path + onlyfiles[i]

images = cv2.imread(image\_path,cv2.IMREAD\_GRAYSCALE)

Training\_data.append(np.asarray(images, dtype = np.uint8))

Lebels.append(i)

Lebels = np.asarray(Lebels, dtype = np.int32)

model = cv2.face.LBPHFaceRecognizer\_create()

model.train(np.asarray(Training\_data),np.asarray(Lebels))

print("training complete")

q+=1

face\_classifier = cv2.CascadeClassifier('haarcascade\_frontalface\_default.xml')

def texttospeech(text):

output = gTTS(text,lang="vi", slow=False)

output.save("output.mp3")

playsound('output.mp3', True)

def face\_detector(img, size= 0.5):

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

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

if faces == ():

return img,[]

for(x,y,w,h) in faces:

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

roi = img[y:y+h, x:x+w]

roi = cv2.resize(roi,(200,200))

return img,roi

cap = cv2.VideoCapture(2)

while True:

ret, frame = cap.read()

image, face = face\_detector(frame)

try:

face = cv2.cvtColor(face,cv2.COLOR\_BGR2GRAY)

result= model.predict(face)

if result[1]<500:

confidence = int((1-(result[1])/300)\*100)

display\_string = str(confidence)

cv2.putText(image, display\_string,(100,120),cv2.FONT\_HERSHEY\_SCRIPT\_COMPLEX,1,(0,255,0))

if confidence>=83:

cv2.putText(image,"Cua duoc mo",(250,450),cv2.FONT\_HERSHEY\_SCRIPT\_COMPLEX,1,(0,255,255))

cv2.imshow('face',image)

x+=1

else:

cv2.putText(image,"Cua khong duoc mo",(250,450),cv2.FONT\_HERSHEY\_SCRIPT\_COMPLEX,1,(0,255,255))

cv2.imshow('face',image)

c+=1

except:

cv2.putText(image,"Khong tim thay khuon mat",(250,450),cv2.FONT\_HERSHEY\_SCRIPT\_COMPLEX,1,(0,255,255))

cv2.imshow('face',image)

d+=1

pass

if cv2.waitKey(1)==13 or x==10 or c==30 or d==20:

break

cap.release()

cv2.destroyAllWindows()

if x>=5:

m=1

ard = serial.Serial('com3' ,9600)

time.sleep(2)

var = 'a'

b=var.encode()

texttospeech('Đã mở cửa')

ard.write(b)

time.sleep(4)

elif c==30:

texttospeech('Cửa sẽ không mở')

ard = serial.Serial('com3' ,9600)

time.sleep(2)

var = 'c'

d=var.encode()

ard.write(d)

time.sleep(4)

elif d==20:

text("Không tìm thấy khuôn mặt xin vui lòng thử lại ")

if m==1:

texttospeech("Cửa đã đóng")