

**BỘ GIÁO DỤC VÀ ĐÀO TẠO  
TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT  
THÀNH PHỐ HỒ CHÍ MINH**



**LUẬN VĂN THẠC SĨ  
TRẦN QUANG**

**PHÁT HIỆN VÀ CẢNH BÁO BUỒN NGỦ CỦA  
TÀI XẾ LÁI XE DÙNG THỊ GIÁC MÁY TÍNH**

**NGÀNH: KỸ THUẬT ĐIỀU KHIỂN VÀ  
TỰ ĐỘNG HÓA - 8520216**



**Tp. Hồ Chí Minh, tháng 09/2019**

**BỘ GIÁO DỤC VÀ ĐÀO TẠO  
TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT  
THÀNH PHỐ HỒ CHÍ MINH**

**LUẬN VĂN THẠC SĨ**

**PHÁT HIỆN VÀ CẢNH BÁO BUỒN NGỦ CỦA TÀI XẾ LÁI XE  
DÙNG THỊ GIÁC MÁY TÍNH**

**NGÀNH: KỸ THUẬT ĐIỀU KHIỂN VÀ TỰ ĐỘNG HÓA - 8520216**

**GIẢNG VIÊN HƯỚNG DẪN: TS. LÊ MỸ HÀ  
HỌC VIÊN: TRẦN QUANG**

**Tp. Hồ Chí Minh, tháng 09 năm 2019**

# LÝ LỊCH KHOA HỌC

(Dùng cho nghiên cứu sinh & học viên cao học)

## I. LÝ LỊCH SƠ LƯỢC:

Họ & tên: TRẦN QUANG

Giới tính: Nam

Ngày, tháng, năm sinh: 19/05/1990

Nơi sinh: Khánh Hòa

Quê quán: Khánh Hòa

Dân tộc: Kinh

Chỗ ở riêng hoặc địa chỉ liên lạc: 554/1 Lê Văn Việt, Quận 9, TP.HCM

Điện thoại cơ quan:

Điện thoại nhà riêng:

Fax:

E-mail: quangtran190590@gmail.com

## II. QUÁ TRÌNH ĐÀO TẠO:

### 1. Trung học chuyên nghiệp:

Hệ đào tạo:

Thời gian đào tạo từ ...../..... đến ...../.....

Nơi học (trường, thành phố):

Ngành học:

### 2. Đại học:

Hệ đào tạo: Chính Quy

Thời gian đào tạo từ 11/2015 đến 06/2017

Nơi học (trường, thành phố): Đại học Công Nghiệp Thực Phẩm TP.HCM

Ngành học: Công nghệ kỹ thuật điện - điện tử

Tên đề án, luận án hoặc môn thi tốt nghiệp:

**Hệ thống mạch điều khiển bơm nước cho nhà cao tầng**

Ngày & nơi bảo vệ đề án, luận án hoặc thi tốt nghiệp: **06/2017 tại trường Đại học Công Nghiệp Thực Phẩm TP.HCM**

Người hướng dẫn: Thạc sỹ Phạm Công Thành

### 3. Thạc sĩ:

Hệ đào tạo: Chính qui

Thời gian đào tạo từ 10/2017 đến 10/2019

Nơi học (trường, thành phố): Đại Học Sư Phạm Kỹ Thuật TP.HCM

Ngành học: Kỹ thuật Điều khiển và Tự động hóa

Tên luận văn:

Ngày & nơi bảo vệ luận văn: **tháng 10/2019 tại trường Đại Học Sư Phạm Kỹ Thuật TPHCM**

Người hướng dẫn: TS. Lê Mỹ Hà

**4. Tiến sĩ:**

Hệ đào tạo:

Thời gian đào tạo từ ...../..... đến ...../...

Tại (trường, viện, nước):

Tên luận án:

Người hướng dẫn:

Ngày &amp; nơi bảo vệ:

**5. Trình độ ngoại ngữ** (biết ngoại ngữ gì, mức độ): Tiếng Anh – B1**6. Học vị, học hàm, chức vụ kỹ thuật được chính thức cấp; số bằng, ngày & nơi cấp:****III. QUÁ TRÌNH CÔNG TÁC CHUYÊN MÔN KỂ TỪ KHI TỐT NGHIỆP ĐẠI HỌC:**

Thời gian	Nơi công tác	Công việc đảm nhiệm
10/2017 – 10/2019	Công ty KONE Việt Nam	Kỹ sư trưởng công trình

**IV. CÁC CÔNG TRÌNH KHOA HỌC ĐÃ CÔNG BỐ:**

**XÁC NHẬN CỦA CƠ QUAN hoặc ĐỊA PHƯƠNG** Ngày 12 tháng 09 năm 2019  
(Ký tên, đóng dấu) **Người khai ký tên**

## **LỜI CAM ĐOAN**

Tôi cam đoan đây là công trình nghiên cứu của tôi.

Các số liệu, kết quả nêu trong luận văn là trung thực và chưa từng được ai công bố trong bất kỳ công trình nào khác.

*Tp. Hồ Chí Minh, ngày 12 tháng 09 năm 2019*

(Ký tên và ghi rõ họ tên)

**Trần Quang**

## LỜI CẢM ƠN

Trong suốt quá trình thực hiện đề tài, với sự giúp đỡ nhiệt tình của quý thầy cô, hướng dẫn về mọi mặt từ thiết kế phần cứng đến phần mềm điều khiển là những yếu tố quyết định đến sự thành công của đề tài ngày hôm nay. Tôi xin chân thành gửi lời cảm ơn đến:

Giảng viên hướng dẫn **TS. Lê Mỹ Hà** đã định hướng, nhiệt tình giúp đỡ, chỉ bảo tận tình và tạo điều kiện tốt nhất cho tôi khi làm việc trong suốt quá trình thực hiện đề tài.

Tôi cũng xin gửi lời cảm ơn đến toàn bộ quý thầy cô trong Khoa Điện – Điện tử và Bộ môn Tự Động Điều Khiển đã giúp đỡ tôi rất nhiều trong quá trình thực hiện đề tài cũng như đóng góp ý kiến và tạo điều kiện thuận lợi giúp đề tài hoàn thiện hơn.

Tôi xin gửi lời cảm ơn đến tập thể lớp cao học TĐH17B, trong quá trình làm đề tài các anh chị đã có những ý kiến thiết thực và giúp đỡ tôi trong việc thực hiện đề tài.

Cuối cùng, tôi xin gửi lời cảm ơn chân thành tới nhà trường, thầy cô, cha mẹ và bạn bè đã động viên và giúp đỡ trong suốt quá trình thực hiện đề tài.

Tôi xin chân thành cảm ơn!

# TÓM TẮT

Cuộc cách mạng công nghiệp lần thứ 4 đã và đang diễn ra mạnh mẽ trên thế giới với những đột phá chưa từng có về công nghệ như trí tuệ nhân tạo (AI), máy học (Machine Learning), thị giác máy tính (Computer Vision), Internet vạn vật (IoT), điện toán đám mây, phương tiện tự hành, công nghệ Nano, in 3D và thực tế ảo,... Cuộc cách mạng này được dự đoán sẽ tác động mạnh mẽ đến tất cả các quốc gia, chính phủ, doanh nghiệp và người dân toàn cầu bởi khả năng khai thác vô cùng lớn từ nó. Những công nghệ với nhiều ưu thế cùng khả năng kết nối, tương tác, độ chính xác và tốc độ ngày càng tốt hơn, ổn định hơn trước.

Xử lý ảnh là một phân ngành trong xử lý tín hiệu số với tín hiệu xử lý là ảnh. Xử lý ảnh gồm 4 lĩnh vực chính: xử lý nâng cao chất lượng ảnh, nhận dạng ảnh, nén ảnh và truy vấn ảnh. Xử lý ảnh ngày càng thể hiện khả năng hữu dụng của mình trên nhiều phương diện, được sử dụng trong nhiều lĩnh vực, ứng dụng vào mọi mặt của đời sống, dần thay thế các công nghệ cũ đặc biệt trong mảng nhận dạng ảnh. Nó sẽ dần thay thế con người trong các công việc liên quan đến việc nhận dạng đối tượng như nhận dạng thuộc tính của đối tượng, phân loại đối tượng, phát hiện đối tượng,...

Trong luận văn này, chúng tôi thiết kế và thi công thiết bị ứng dụng công nghệ xử lý hình ảnh kết hợp mạng nơ-ron học sâu với đề tài “Phát hiện và cảnh báo buồn ngủ của tài xế lái xe dùng thị giác máy tính”. Thiết bị này sử dụng bộ xử lý trung tâm là máy tính nhúng Raspberry Pi, hình ảnh sau khi thu được từ camera thông qua kỹ thuật xử lý ảnh sẽ phát hiện và định vị 68 điểm đặc trưng (mắt, mũi, miệng,...) trên khuôn mặt, trích xuất vùng mắt và tỉ lệ nhắm mở mắt trong một chu kỳ định định trước. Sau đó, mạng học sâu được huấn luyện để dự đoán, phát hiện dấu hiệu buồn ngủ và phát ra các cảnh báo cần thiết cho người lái xe. Kết quả thực nghiệm cho thấy tính hiệu quả và sự mạnh mẽ của mạng nơ-ron học sâu trong việc phát hiện trạng thái buồn ngủ trong thời gian thực. Thiết bị hoạt động ổn định và cảnh báo chính xác dấu hiệu buồn ngủ trong nhiều điều kiện lái khác nhau.

# MỤC LỤC

MỤC LỤC.....	i
DANH MỤC CÁC CHỮ VIẾT TẮT .....	iii
DANH MỤC CÁC HÌNH ẢNH VÀ BIỂU ĐỒ .....	iv
Chương 1: TỔNG QUAN.....	1
1.1. Giới thiệu.....	1
1.1.1. Đặt vấn đề .....	1
1.1.2. Tính cấp thiết của đề tài.....	3
1.2. Mục tiêu đề tài .....	5
1.2.1. Mục tiêu của đề đề tài .....	5
1.2.2. Nhiệm vụ đề tài.....	6
1.3. Giới hạn đề tài .....	6
1.4. Phương pháp nghiên cứu .....	6
1.5. Nội dung đề tài .....	7
Chương 2: CƠ SỞ LÝ THUYẾT .....	9
2.1. Tổng quan về xử lý ảnh.....	9
2.1.1. Giới thiệu về xử lý ảnh .....	9
2.1.2. Các thành phần cơ bản của hệ thống ảnh.....	11
2.1.3. Các vấn đề cơ bản trong xử lý ảnh .....	12
2.2. Tổng quan về bài toán nhận dạng mặt người .....	14
2.2.1. Giới thiệu. ....	14
2.2.2. Thách thức trong nhận dạng mặt người.....	15
2.2.3. Các cách tiếp cận cho bài toán nhận dạng mặt người.....	16
2.3. Phương pháp mô tả đặc trưng Histogram of Oriented Gradient (HOG). ....	19
2.4. Support vector machine (SVM).....	22
2.5. Phân tích điểm đặc trưng trên khuôn mặt người - Facial Landmark.....	24
2.6. Tỷ lệ nhắm mở mắt – Eyes Aspect Ratio (EAR) .....	26
2.7. Mạng neural nhân tạo .....	28
2.7.1. Khái niệm chung.....	28
2.7.2. Mạng nơron và mạng nơron lan truyền ngược .....	29
2.7.3. Một số kiểu mạng nơron.....	32



2.7.4. Các phương pháp huấn luyện mạng thần kinh nhân tạo.....	34
2.8. Ngôn ngữ lập trình Python. ....	35
2.9. Thư viện OpenCV và dlib.....	37
2.9.1. Thư viện OpenCV.....	37
2.9.2. Thư viện dlib.....	41
Chương 3: GIẢI THUẬT .....	42
3.1. Sơ đồ khối hệ thống.....	42
3.2. Sơ đồ tiến trình hệ thống phát hiện dấu hiệu buồn ngủ.....	43
3.3. Lưu đồ chương trình phát hiện dấu hiệu buồn ngủ. ....	44
3.4. Lưu đồ chương trình trên Arduino. ....	45
Chương 4: KẾT QUẢ THỰC NGHIỆM.....	46
4.1. Xây dựng phần cứng.....	46
4.1.1. Sơ đồ nối dây phần cứng .....	46
4.1.2. Board xử lý trung tâm.....	46
4.1.3. Board điều khiển – Board Arduino Nano. ....	50
4.1.4. Camera Logitech C170.....	52
4.2. Kết quả thực nghiệm.....	52
4.2.1. Mô hình phần cứng thiết bị.....	52
4.2.2. Thu thập dữ liệu huấn luyện .....	53
4.2.3. Cấu trúc mạng và tối ưu các thông số.....	55
4.2.4. Kết quả huấn luyện mạng.....	56
4.2.5. Kết quả thực nghiệm trên hệ thống.....	57
Chương 5: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN.....	62
5.1. Kết luận: .....	62
5.2. Hướng phát triển:.....	62
TÀI LIỆU THAM KHẢO.....	63

## **DANH MỤC CÁC CHỮ VIẾT TẮT**

1. AI: Artificial Intelligence
2. ANN: Artificial Neural Network
3. EAR: Eyes Aspect Ratio
4. PCA: Principal Components Analysis
5. LDA: Linear Discriminant Analysis
6. LBP: Local Binary Pattern
7. k-NN: k Nearest Neighbors
8. HOG: Histogram of Gradient
9. SVM: Support Vector Machine

## DANH MỤC CÁC HÌNH ẢNH VÀ BIỂU ĐỒ

Hình 2.1 Các bước cơ bản trong xử lý ảnh .....	9
Hình 2.2 Các thành phần cơ bản của hệ thống xử lý ảnh.....	11
Hình 2.3 Độ phân giải của ảnh.....	12
Hình 2.4 Sự khác nhau giữa ảnh màu và ảnh xám.....	13
Hình 2.5 Các thông tin có trong ảnh mặt người.....	15
Hình 2.6 Các bước trong một hệ thống nhận dạng mặt. ....	18
Hình 2.7 Ảnh input và hai đạo hàm của nó.....	20
Hình 2.8 Một ảnh sau khi trích chọn đặc trưng HOG.....	22
Hình 2.9 SVM tìm mặt phẳng phân chia dữ liệu ứng với margin lớn nhất .....	23
Hình 2.10 Ảnh input và kết quả phát hiện và căn chỉnh ảnh mặt với HOG và SVM. .....	24
Hình 2.11 68 điểm đặc trưng trên khuôn mặt người.....	25
Hình 2.12 Xác định mặt người trong tấm ảnh.....	25
Hình 2.13 Xác định 68 điểm đặc trưng trên khuôn mặt dùng OpenCV .....	26
Hình 2.14 Các điểm xác định vị trí mắt .....	26
Hình 2.15 Xác định vị trí vùng mắt trên khuôn mặt và đồ thị thời gian nhắm mở mắt.....	27
Hình 2.16: Mạng thần kinh nhân tạo.....	30
Hình 2.17 Một số hàm tác động.....	32
Hình 2.18 Mạng truyền thẳng một lớp.....	32
Hình 2.19 Mạng truyền thẳng nhiều lớp. ....	32
Hình 2.20 Mạng hồi quy một lớp.....	33
Hình 2.21 Mạng hồi quy nhiều lớp. ....	33
Hình 2.22 Phân loại kiểu học. ....	34
Hình 2.23 Học có giám sát.....	34
Hình 2.24 Học củng cố.....	35
Hình 2.25 Học không giám sát.....	35

Hình 2.26 Cấu trúc của thư viện OpenCV .....	38
Hình 2.27 Cấu trúc của thư viện dlib. ....	41
Hình 3.1 Sơ đồ khối hệ thống .....	42
Hình 3.2 Sơ đồ tiến trình hệ thống .....	43
Hình 3.3 Lưu đồ giải thuật cho chương trình Raspberry .....	44
Hình 3.4 Lưu đồ giải thuật cho chương trình Arduino .....	45
Hình 4.1 Sơ đồ nối dây phần cứng .....	46
Hình 4.2 Board Raspberry Pi 3 Model B+ .....	47
Hình 4.3 Hệ điều hành Raspian trên trang chủ Raspberry Pi .....	49
Hình 4.4 Phần mềm Win32 Disk Imager .....	49
Hình 4.5 Sơ đồ Arduino Nano .....	50
Hình 4.6 Camera Logitech C170 .....	52
Hình 4.7 Mặt trước của thiết bị cảnh báo buồn ngủ.....	52
Hình 4.8 Thành phần bên trong của thiết bị cảnh báo buồn ngủ .....	53
Hình 4.9 Dây EAR liên tục khi thu dữ liệu.....	54
Hình 4.10 Dây EAR trong 90 khung hình liên tục.....	54
Hình 4.11 Phân bố dữ liệu huấn luyện.....	55
Hình 4.12 Cấu trúc mạng nơron phát hiện buồn ngủ.....	55
Hình 4.13 Thông số của mạng nơron phát hiện buồn ngủ .....	56
Hình 4.14 Confusion Matrix .....	57
Hình 4.15: Thực nghiệm khi tài xế ngồi gần trong trạng thái tỉnh táo.....	58
Hình 4.16: Thực nghiệm khi tài xế ngồi xa trong trạng thái tỉnh táo.....	59
Hình 4.17: Thực nghiệm khi tài xế ngồi gần trong trạng thái buồn ngủ.....	60
Hình 4.18: Thực nghiệm khi tài xế ngồi xa trong trạng thái buồn ngủ.....	61



# **Chương 1: TỔNG QUAN**

## **1.1. Giới thiệu**

### **1.1.1. Đặt vấn đề**

Tai nạn giao thông từ lâu nay luôn là vấn đề nhức nhối ở Việt Nam. Thống kê từ Ủy ban An toàn Giao thông Quốc gia cho thấy trong quý I năm 2018, cả nước xảy ra gần 5.000 vụ tai nạn giao thông, làm chết 2.149 người, bị thương 3.627 người. Năm 2017, cả nước có hơn 20.000 vụ tai nạn giao thông, khiến 8.279 người chết và hơn 17.000 người bị thương. So với cách đây 5 năm, lượng người chết đã giảm đi, nhưng chưa bao giờ thấp hơn con số 8.000. Thậm chí năm 2016, số người chết vì tai nạn giao thông còn tăng lên 14 người so với năm trước đó. “Có thể nói chúng ta sống trong thời bình nhưng số người chết và bị thương do tai nạn giao thông giống như những năm chiến tranh”, theo Bộ trưởng Bộ Giao thông Vận tải Nguyễn Văn Thể, tại lễ trao giải cuộc thi báo chí viết về an toàn giao thông 2017.

Một trong những tác nhân mà không ai có thể kiểm tra, ngăn chặn được, đó là việc ngủ gật của tài xế, đây chính là nguyên nhân gây ra tai nạn giao thông hàng đầu trên thế giới. Điển hình như vụ tai nạn xe đi rước dâu thảm khốc tại tỉnh Quảng Nam vừa qua (30/7/2018), khiến 13 người tử vong. Nguyên nhân của việc ngủ gật nói trên là do tài xế này đã lái xe 12 giờ đồng hồ liên tục.

Lái xe là một nghề khá nguy hiểm, nhất là đối với những lái xe đi đường dài như xe khách, xe tải, xe container. Chỉ cần tài xế trong tình trạng không tỉnh táo, khả năng kiểm soát kém, quan sát và phản ứng khi cơn buồn ngủ ập đến sẽ dễ gây ra tai nạn. Hiện nay, trên cả nước mỗi năm có hàng trăm vụ tai nạn xảy ra, chỉ vì tài xế ngủ gật trên vô lăng. Theo Đại úy Huỳnh Phước Huy, Đội trưởng Đội Tuyên truyền - xử lý, phòng Cảnh sát giao thông đường bộ - đường sắt, Công an tỉnh cho biết: "Chỉ cần vài giây "lơ mơ" trong lúc cầm lái thì điều đáng tiếc có thể đã xảy ra. Bất kỳ người điều khiển xe thô sơ hay cơ giới đều cũng gặp nguy hiểm khi tài xế trong tình trạng buồn ngủ. Nếu trường hợp xe thô sơ, hậu quả có thể đơn lẻ, nhưng với ô tô, một cái chớp mắt có thể gây ra tai nạn liên hoàn, nhiều khi là thảm khốc”.

Qua 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 National Jewish Health, Mỹ), cho biết 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. Cũng theo một cuộc thăm dò tiến hành bởi Tổ chức chăm sóc giấc ngủ quốc gia Mỹ (National Sleep Foundation) cho thấy 60 % người Mỹ ngủ gật khi lái xe, 37 % thú nhận không tỉnh táo sau vô lăng trong năm qua. Đây là điều hết sức nguy hiểm bởi theo Cục An toàn giao thông đường bộ Mỹ (NHTSA), ngủ gật khi lái xe là nguyên nhân dẫn tới hơn 100.000 vụ đụng xe và khiến trên 1.500 người tử vong mỗi năm. Ai cũng biết ngủ gật khi lái xe là một việc hết sức nguy hiểm. Theo giáo sư, tiến sĩ Ngô Quý Châu, Chủ tịch Hội Hô hấp Việt Nam khuyến nghị để giảm thiểu tai nạn giao thông vốn là vấn đề nhức nhối mang tính thời đại cần có sự vào cuộc của nhiều cơ quan, tổ chức, đặc biệt là ý thức người tham gia giao thông. Những khảo sát cho thấy, ngủ gật sau vô lăng là nguyên nhân dẫn tới 30% số tai nạn xảy ra, đặc biệt với những chuyến đi đường dài.

### **Các nguyên nhân gây ra trạng thái buồn ngủ cho tài xế:**

- Trong thực tế, di chuyển xe đường dài hoặc vào ban đêm là lúc người lái xe dễ rơi vào trạng thái buồn ngủ nhất, phản xạ của người lái sẽ chậm lại, độ tỉnh táo giảm, rất dễ căng thẳng và khả năng xử lý tình huống thiếu chính xác.

- Không những vậy, để tăng chuyên, nhiều tài xế ô tô đường dài vắt kiệt sức liên tục quay vòng dẫn đến tình trạng buồn ngủ.

- Hầu hết tài xế gây tai nạn đều có thời gian thức kéo dài, ngủ ít hơn 5 giờ vào đêm trước khi gặp nạn do các nhà xe khách hiện nay đều tận dụng lái xe ít nhất từ 12 - 14 giờ/ngày.

- Đặc biệt, việc sử dụng rượu bia và rối loạn giấc ngủ có tác dụng cộng gộp làm suy giảm hiệu suất của các cơ quan trong cơ thể, dẫn đến nhiều sai sót nghiêm trọng.

- Chưa có thiết bị giúp các tay lái nhận ra mình đang trong tình trạng báo động để kiểm soát cơn buồn ngủ sau vô lăng .

### **1.1.2. Tính cấp thiết của đề tài**

Tình hình nghiên cứu trong và ngoài nước:

*\* Trong nước:*

- Đề tài “Nhận diện chớp mắt và cảnh báo buồn ngủ khi lái xe trên nền Windows và Android” của sinh viên Lê Thế Hải và Lê Thanh Hòa (ngành Viễn thông, Khoa Điện - Điện tử, niên khóa 2007 – 2012, Trường Đại học Bách Khoa TP.HCM). Đây là chương trình phần mềm trên hệ điều hành Android cho điện thoại và máy tính bảng nhận biết được dấu hiệu buồn ngủ thông qua trạng thái đóng mở mắt và phát âm thanh. Tuy nhiên, đề tài cũng thể hiện một số nhược điểm như: không theo dõi được khuôn mặt trong một số trường hợp quay trái, quay phải một góc lớn hơn 40 độ hoặc cúi đầu lớn hơn 10 độ - 15 độ/ngẩng đầu lên quá 30 độ... cũng như chưa có phần cứng thiết bị riêng và camera điện thoại chỉ xử lý được điều kiện ánh sáng giới hạn.

- Sản phẩm máy chống buồn ngủ của Nguyễn Ngọc Đức, học sinh lớp 12C1 trường THPT Nguyễn Xuân Nguyên (Thanh Hóa), đã giành giải Nhì cấp Quốc gia tại cuộc thi sáng tạo Khoa học kỹ thuật dành cho học sinh THPT năm 2016, được thiết kế theo mô hình của một chiếc máy tính thu nhỏ. Thiết bị dùng phần cứng là một máy tính mini, phần mềm được lập trình theo ngôn ngữ C# có sử dụng mã nguồn mở và một số thư viện về nhận dạng mắt và khuôn mặt. Nguyên lý hoạt động của máy chống buồn ngủ là khi camera ghi lại hình ảnh của lái xe, thiết bị sẽ nhận diện ra điểm khác biệt trên khuôn mặt. Từ đó, phần mềm sẽ phân tích để tìm ra được dấu hiệu buồn ngủ qua ánh mắt bằng biểu đồ ánh sáng để phân biệt độ sáng, tối của mắt. Sau khi thiết bị nhận dạng xong ánh mắt sẽ tiếp tục đưa ra các phương án để xử lý, cảnh báo bằng giọng nói và còi báo động. Tuy nhiên, đây là một sản



phẩm mô hình thử nghiệm chỉ dừng lại ở việc cảnh báo mà chưa có biện pháp hạn chế tai nạn cho lái xe cũng như chưa được ứng dụng vào thực tế.

*\*Ngoài nước:*

- Tại triển lãm công nghệ CES đầu năm 2018, Hyundai giới thiệu hệ thống chống ngủ gật cho tài xế có tên DDREM. Hệ thống DDREM sử dụng một camera hồng ngoại để theo dõi và phân tích các chuyển động của mắt cũng như khuôn mặt để phát hiện dấu hiệu buồn ngủ của tài xế. Hệ thống này cũng giám sát hệ thống hỗ trợ lái xe để xem xe có đang đi chệch làn đường không và điều chỉnh sao cho xe đi đúng làn đường trở lại. Nếu hệ thống xác nhận người lái đã ngủ và không phản hồi gì thì DDREM sẽ chuyển xe sang chế độ lái tự động cấp độ 4 để có thể tự tìm một nơi an toàn và dừng xe lại. Công nghệ này tương tự chức năng cảnh báo của hệ thống tự lái cấp độ 3 trên chiếc Audi A8 cũng dùng camera phân tích vị trí và chuyển động của đầu và mắt để tạo ra dữ liệu cho robot AI phân tích. Nếu mắt của lái xe vẫn nhắm trong một khoảng thời gian dài, hệ thống sẽ nhắc người lái xe tiếp tục công việc lái xe, tránh trường hợp lái xe ngủ gục rất nguy hiểm.

Theo như bài báo “Driver Drowsiness Detection System and Techniques: A Review” của tác giả Vandna Saini và Rekha Saini đăng trên tạp chí (IJCSIT) International Journal of Computer Science and Information Technologies, Vol. 5 (3) năm 2014 tại Ấn Độ trình bày về các kỹ thuật phát hiện ngủ gật thì việc phát hiện trạng thái buồn ngủ của tài xế có thể chia ra làm 3 hướng tiếp cận:

(1) Thông số phương tiện dựa trên quãng đường đã đi, độ lệch làn đường, số vòng quay của vô lăng, tốc độ đạp chân ga,... Các thông số này được theo dõi liên tục, nếu có bất kỳ thay đổi nào trong số đó vượt qua một ngưỡng xác định cho thấy khả năng người lái xe có dấu hiệu buồn ngủ.

(2) Biểu hiện trên khuôn mặt của tài xế bao gồm việc ngáp, thời gian nhắm mắt, số lần chớp mắt, hướng nhìn,...

(3) Thông số về sinh lý dựa trên mối tương quan giữa tín hiệu điện tâm đồ (ECG) và điện nhãn đồ (EOG), trạng thái buồn ngủ được xác định trên thông tin về nhịp tim, huyết áp và sóng não.

Theo đó, tác giả nhận thấy việc giám sát các thông số trên phương tiện là điều rất khó thực hiện khi phải tác động trực tiếp vào cả phần cứng và phần mềm của xe, việc thiết kế bộ giám sát các thông số của xe và đưa ra cảnh báo nên được nghiên cứu bởi các nhà sản xuất và hoàn thiện trước khi xe xuất xưởng. Trong khi đó, việc nghiên cứu chế tạo thiết bị theo dõi các thông số sinh học của tài xế khi lái xe với độ ổn định và chính xác cao là một nhiệm vụ rất lớn, đòi hỏi về thời gian, kinh phí và khối lượng rất lớn kiến thức chuyên môn đa ngành bao gồm cả y học và kỹ thuật. Chính vì những lý do trên, tác giả chọn hướng tiếp cận vấn đề thứ (2) – phát hiện dấu hiệu buồn ngủ dựa trên biểu hiện khuôn mặt của tài xế, cụ thể ở đây là độ nhắm mở mắt và thời gian nhắm mắt, phương pháp này đã được rất nhiều các hãng xe lớn như BMW, Mercedes-Benz, Audi, Hyundai, Cadillac trong hệ thống nhận cảnh báo mất tập trung cho tài xế tên các dòng xe sang của họ.

Từ những thực tiễn trình bày ở trên có thể nhận thấy rằng việc nghiên cứu, thiết kế và chế tạo thiết bị cảnh báo ngủ gật cho tài xế có độ chính xác cao là việc cấp bách và rất cần thiết để giảm tỷ lệ tai nạn giao thông hiện nay. Vì thế, tác giả lựa chọn và thực hiện đề tài: “***Phát hiện và cảnh báo buồn ngủ của tài xế lái xe dùng thị giác máy tính***”.

## **1.2. Mục tiêu đề tài**

### **1.2.1. Mục tiêu của đề đề tài**

Nghiên cứu, thiết kế và chế tạo thiết bị phát hiện và cảnh báo buồn ngủ cho tài xế dùng thị giác máy tính và mạng nơ-ron học sâu hoạt động ổn định, có thể sử dụng trên các phương tiện giao thông như xe ô tô, xe tải, xe khách và xe container... thực hiện các nhiệm vụ sau:

- Ứng dụng công nghệ xử lý hình ảnh nhận diện khuôn mặt, trích xuất vùng mắt, theo dõi trạng thái nhắm mở mắt của tài xế trong quá trình điều khiển phương tiện với điều kiện ánh sáng ban ngày.

- Thu thập mẫu về tần số nhắm mở mắt để huấn luyện bộ nhận dạng trạng thái buồn ngủ của tài xế sử dụng mạng nơ-ron học sâu.

- Xây dựng và nhúng hệ thống vào board mạch Raspberry Pi 3 Model B+ và Arduino.

- Khi có dấu hiệu buồn ngủ hệ thống sẽ cảnh báo cho tài xế bằng giọng nói, âm thanh và đèn báo động.

### **1.2.2. Nhiệm vụ đề tài**

- Nghiên cứu các bài toán xử lý ảnh: phát hiện mặt người dùng HOG và SVM, trích xuất 68 điểm đặc trưng trên khuôn mặt, trích vùng mắt và tính toán độ nhắm mở mắt (EAR).

- Thu thập dữ liệu nhắm mở mắt, xây dựng và huấn luyện mạng nơron học sâu dùng xác định trạng thái buồn ngủ của tài xế.

- Viết chương trình Python chạy trên board mạch Raspberry Pi 3 Model B+ để thiết bị có thể phát hiện và cảnh báo trạng thái buồn ngủ cho tài xế trong thời gian thực.

### **1.3. Giới hạn đề tài**

- Ứng dụng công nghệ xử lý hình ảnh và mạng nơron học sâu trong việc theo dõi và cảnh báo dấu hiệu buồn ngủ của tài xế trong điều kiện ánh sáng ban ngày.

- Điều kiện ánh sáng không quá sáng chói hoặc quá tối làm mất hình dạng khuôn mặt trên camera.

- Nghiên cứu và thực nghiệm sản phẩm trên mô hình, sau khi hiệu chỉnh hoàn thiện thì có thể áp dụng cho việc cảnh báo dấu hiệu ngủ cho gạt các tài xế lái xe trên đường.

- Khi phát hiện tài xế có dấu hiệu ngủ gật, hệ thống sẽ cảnh báo bằng giọng nói và âm thanh và đèn báo.

- Thiết bị được đặt cách tài xế với khoảng cách nhất định, nhỏ hơn 100cm để đảm bảo camera nhận dạng rõ khuôn mặt.

### **1.4. Phương pháp nghiên cứu**

- Tham khảo, nghiên cứu nền tảng lý thuyết, bài báo khoa học và nhu cầu thực tiễn, nội dung, yêu cầu và đối tượng nghiên cứu được cụ thể hóa. Sau đó, việc

nghiên cứu lý thuyết được tiến hành (phương pháp nhận dạng khuôn mặt, trích vùng khuôn mặt và 2 mắt, xác định độ mở của mắt bằng công nghệ xử lý hình ảnh).

- Lựa chọn board mạch và các linh kiện cũng như tìm hiểu nguyên lý hoạt động, cách kết nối chúng được tiến hành. Thông qua quá trình nghiên cứu thiết kế, lựa chọn các thông số của thiết bị, bắt đầu gia công và lắp ráp phần cứng sản phẩm. Thiết bị được kiểm tra sơ bộ, điều chỉnh và được hoàn thiện theo các yêu cầu về thiết kế.

- Từ kiến thức về lý thuyết, xây dựng thuật toán, viết chương trình cảnh báo ngủ gật từ hình ảnh thu về bằng Camera, cảnh báo bằng giọng nói và gửi tin nhắn cho nhà điều hành xe bằng ngôn ngữ lập trình Python và Arduino, hiệu chỉnh phần mềm sao cho ổn định và có độ chính xác cao.

- Thiết bị sau khi hoàn thiện được tiến hành chạy thử nghiệm trong môi trường ánh sáng ban ngày và đánh giá hiệu năng, thiết bị cảnh báo ngủ gật tiếp tục được nghiên cứu, điều chỉnh và hoàn thiện.

## 1.5. Nội dung đề tài

Đề tài “Phát hiện và cảnh báo buồn ngủ của tài xế lái xe dùng thị giác máy tính” bao gồm các chương sau:

**Chương I: Tổng quan:** Chương này trình bày tổng quan sơ bộ về các yêu cầu của bài báo cáo như đặt vấn đề, mục tiêu, giới hạn và nội dung đề tài, phương pháp nghiên cứu.

**Chương II: Cơ sở lý thuyết:** Chương này trình bày về lý thuyết phương pháp nhận dạng mặt người dùng HOG và SVM, xác định 68 điểm đặc trưng trên khuôn mặt người, trích xuất vùng mắt và tính toán độ nhắm mở mắt, lý thuyết mạng nơron học sâu và ứng dụng trong bài toán phân loại.

**Chương III: Giải thuật:** Chương này trình bày về các giải thuật, lưu đồ huấn luyện, cấu trúc mạng nơron, phát hiện dấu hiệu buồn ngủ với mạng đã được huấn luyện, sơ đồ khối và lưu đồ phần mềm trên Raspberry Pi và Arduino.

**Chương V: *Kết quả thực nghiệm:*** Chương này trình bày về thiết kế phần cứng, lựa chọn board mạch cho thiết bị và kết quả đạt được sau khi thực hiện. Đánh giá, nhận xét những gì đã đạt và chưa đạt so với mục tiêu đề ra.

**Chương VI: *Kết luận và hướng phát triển:*** Kết luận chung về ưu điểm và giới hạn của đề tài. Đồng thời cũng đưa ra những đóng góp và đề xuất, định hướng phát triển của đề tài.

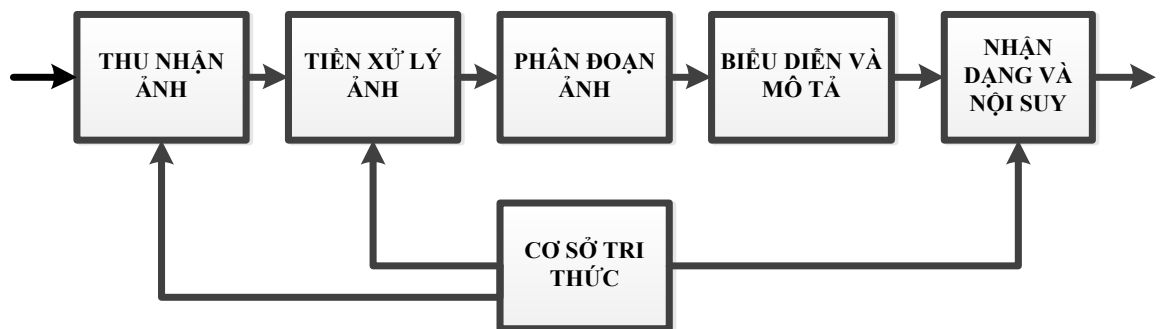
## Chương 2: CƠ SỞ LÝ THUYẾT

### 2.1. Tổng quan về xử lý ảnh

#### 2.1.1. Giới thiệu về xử lý ảnh

Xử lý ảnh (XLA) là đối tượng nghiên cứu của lĩnh vực thị giác máy, là quá trình biến đổi từ một ảnh ban đầu sang một ảnh mới với các đặc tính và tuân theo ý muốn của người sử dụng. Xử lý ảnh có thể gồm quá trình phân tích, phân lớp các đối tượng, làm tăng chất lượng, phân đoạn và tách cạnh, gán nhãn cho vùng hay quá trình biên dịch các thông tin hình ảnh của ảnh.

Trước đây, hình ảnh được thu từ camera là các ảnh tương tự. Gần đây, với sự phát triển không ngừng của công nghệ, ảnh màu hoặc ảnh đen trắng được lấy từ camera, sau đó được chuyển trực tiếp qua ảnh số để dễ dàng cho các bước xử lý tiếp theo. Dưới đây sẽ mô tả về các bước trong xử lý ảnh.



Hình 2.1 Các bước cơ bản trong xử lý ảnh

**Thu nhận ảnh:** Ảnh được nhận qua camera màu hoặc trắng đen. Thông thường ảnh được nhận qua camera, video, máy scan.v.v.

**Mục đích:** biến đổi thông tin hình ảnh về các cấu trúc được lưu trữ trong máy tính, có thể hiển thị ra các thiết bị ngoại vi như là máy in, màn hình...

**Gồm hai tiến trình:**

- Biến đổi năng lượng quang học thành năng lượng điện.
- Tổng hợp năng lượng điện thành ảnh hoặc ma trận số.

**Tiền xử lý:** Sau bộ thu nhận ảnh, hình ảnh có thể có độ tương phản thấp nên cần đưa vào bộ tiền xử lý để nâng cao chất lượng hình ảnh. Bộ tiền xử lý có chức năng lọc nhiễu, nâng độ tương phản để làm cho ảnh rõ hơn và sắc nét hơn.

*Mục đích:*

- Điều chỉnh độ chiếu sáng để khắc phục hậu quả của việc chiếu sáng không đều.
- Giảm nhỏ thành phần nhiễu của ảnh tức là các đối tượng xuất hiện ngoài ý muốn.
- Hiệu chỉnh giá trị độ sáng giữa nền và đối tượng.
- Chuẩn hoá độ lớn, màu, dạng của ảnh.
- Điều chỉnh bộ lọc để khuếch đại và nén các tần số.

**Phân đoạn ảnh:** Là quá trình phân chia nội dung các đối tượng cần khảo sát ra khỏi ảnh, phân chia các đối tượng tiếp giáp nhau, phân tách các đối tượng riêng biệt thành các đối tượng con.

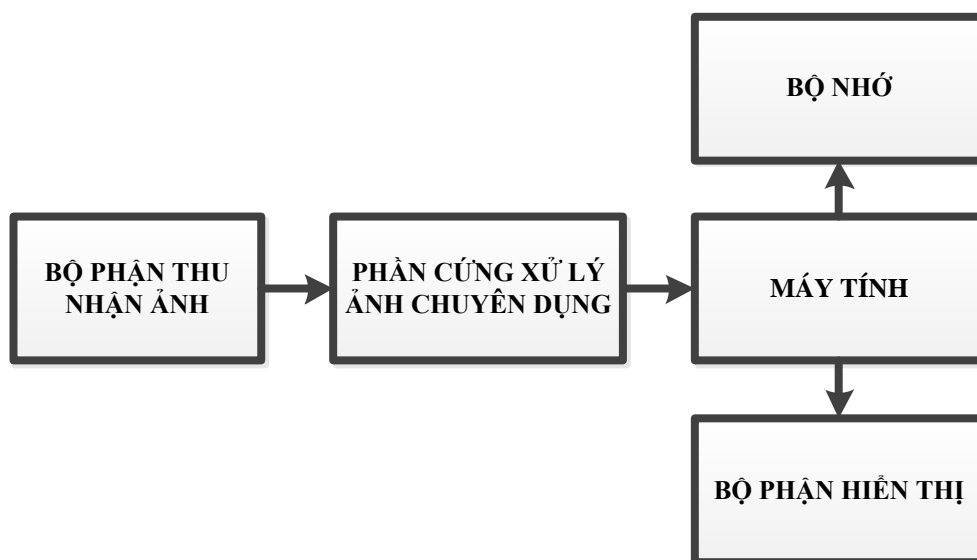
**Biểu diễn ảnh và mô tả ảnh:** Đầu ra ảnh sau phân đoạn chứa các điểm ảnh của vùng ảnh (ảnh đã phân đoạn) cộng với mã liên kết với các vùng lân cận. Việc biến đổi các số liệu này thành dạng thích hợp là cần thiết cho xử lý tiếp theo bằng máy tính. Việc chọn các tính chất để thể hiện ảnh gọi là trích chọn đặc trưng (Feature Selection) 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.

**Nhận dạng và nội suy ảnh:** Nhận dạng ảnh là quá trình xác định ảnh. Bằng cách so sánh mẫu với mẫu chuẩn đã được lưu trữ từ trước. Nội suy là phán đoán theo ý nghĩa trên cơ sở nhận dạng. Các mô hình toán học về ảnh được phân loại với hai dạng cơ bản:

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

**Cơ sở tri 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 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 lý đó, nhiều khâu hiện nay đã xử lý theo các phương pháp trí tuệ con người. Vì vậy, ở đây các cơ sở tri thức được phát huy.

### 2.1.2. Các thành phần cơ bản của hệ thống ảnh



Hình 2.2 Các thành phần cơ bản của hệ thống xử lý ảnh

**Bộ phận thu nhận ảnh:** Máy quay (Camera), máy quét (scanners) chuyên dụng, các bộ cảm biến ảnh.

**Phần cứng xử lý ảnh chuyên dụng:** Bộ số hóa (chuyển đổi ảnh truyền thống từ bên ngoài thành dạng dữ liệu số mà máy tính có thể hiểu được). Phần cứng thực hiện các thao tác cơ bản để nâng cao tốc độ xử lý ảnh.

**Máy tính:** Thiết bị thông thường hoặc chuyên dụng.

**Bộ nhớ:** Bắt buộc phải có. Lưu trữ tạm thời để phục vụ và sử dụng cho quá trình xử lý hiện tại. Lưu trữ vĩnh viễn là lưu trữ dữ liệu, truy cập không thường xuyên.

**Bộ phận hiển thị:** Màn hình máy tính...

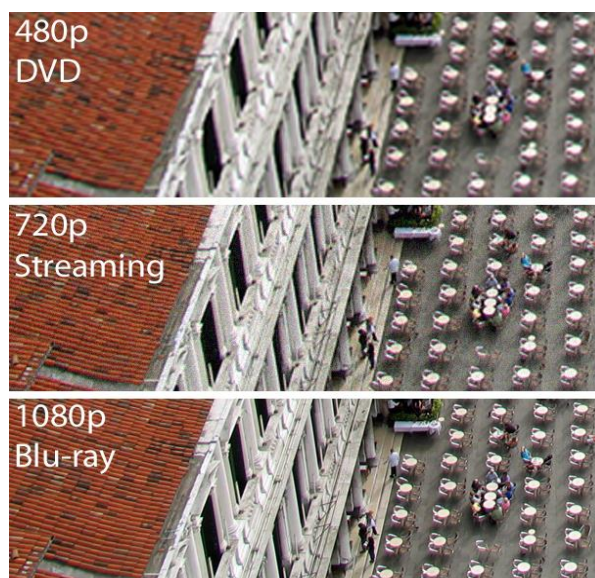


### 2.1.3. Các vấn đề cơ bản trong xử lý ảnh

*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 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.

Pixel có kích thước vào khoảng  $0,26 \times 0,35$  mm, pixel là đơn vị cơ bản nhất để tạo nên bức ảnh kỹ thuật số.

*Độ phân giải của ảnh:* là mật độ điểm ảnh được ấn định trên một ảnh số được hiện thị. Khoảng cách giữa các điểm ảnh sao cho mắt người vẫn thấy được sự liên tục của ảnh. Độ phân giải được phân bố theo trục x và y trong không gian hai chiều. Với cùng một ảnh, độ phân giải càng cao thì ảnh càng chứa nhiều thông tin và sắc nét hơn. Ví dụ như hình bên dưới:

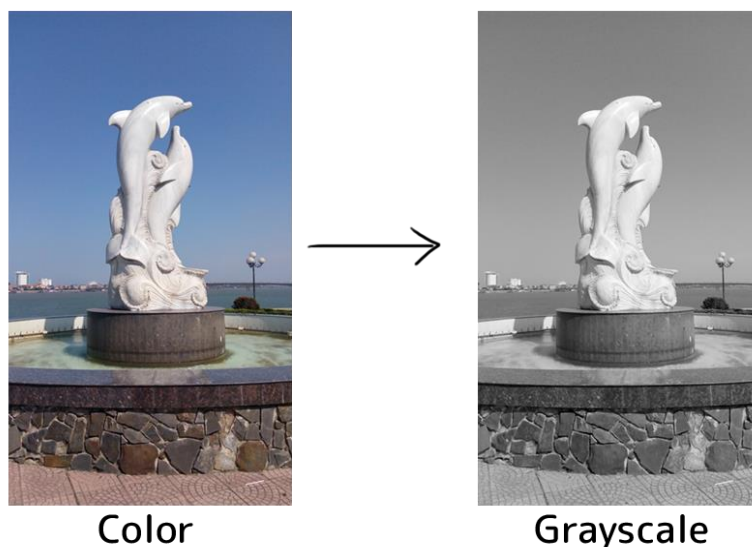


Hình 2.3 Độ phân giải của ảnh

Hình trên cùng có độ phân giải là 480p cho ra chất lượng ảnh mờ và không sắc nét bằng hình có độ phân giải là 720p và 1080p. Với độ phân giải càng cao sẽ cho ra những tấm ảnh rõ nét.

*Mức xám của ảnh:* là kết quả của sự biến đổi tương ứng 1 giá trị độ sáng của 1 điểm ảnh với 1 giá trị nguyên dương. Thông thường nó xác định trong  $[0, 255]$  tùy thuộc vào giá trị mà mỗi điểm ảnh được biểu diễn. Các thang giá trị mức

xám thông thường: 16, 32, 64, 128, 256. Mức 256 là mức thông dụng vì kỹ thuật máy tính dùng 1-byte (8 bit) để biểu diễn mức xám. Mức xám dùng 1-byte biểu diễn:  $2^8 = 256$  mức, tức là từ 0 đến 255.



Hình 2.4 Sự khác nhau giữa ảnh màu và ảnh xám

*Biến đổi ảnh:* trong xử lý ảnh do số điểm ảnh lớn hơn các tính toán nhiều (độ phức tạp tính toán cao) đòi hỏi dung lượng bộ nhớ lớn, thời gian tính toán lâu. Các phương pháp khoa học kinh điển áp dụng cho xử lý ảnh hầu như khó khả thi. Người ta sử dụng các phép toán tương đương hoặc biến đổi sang miền xử lý khác để dễ tính toán, sau khi đã xử lý dễ dàng, dùng biến đổi ngược để đưa về miền xác định ban đầu, các biến đổi thường gặp trong xử lý ảnh bao gồm:

- Biến đổi Fourier, Cosin, Sin.
- Biến đổi (mô tả) ảnh bằng tích chập, tích Kronecker.
- Các biến đổi khác như KL (Karhunen Loeve), Hadamard.

#### **Các phân loại ảnh:**

- *Ảnh nhị phân* : Giá trị xám của tất cả các điểm ảnh chỉ nhận giá trị 1 hoặc 0 như vậy mỗi điểm ảnh trong ảnh nhị phân được biểu diễn bởi 1 bit.
- *Ảnh xám*: Giá trị xám nằm trong  $[0, 255]$  như vậy mỗi điểm ảnh trong ảnh nhị phân được biểu diễn bởi 1 byte.

- *Ảnh màu:*

➤ *Hệ màu RGB:*

Một pixel được biểu diễn bằng 3 giá trị (R, G, B) trong đó R, G, B là một giá trị xám và được biểu diễn bằng 1 byte. Khi đó ta có một ảnh 24 bits.

$$P(x, y) = (R, G, B)$$

➤ *Hệ màu CMY:* là phần bù của hệ màu RGB

$$(C, M, Y) = (1, 1, 1) - (R, G, B)$$

$$\text{Hay } C+R=M+G=Y+B=1$$

Hệ màu này thường được dùng trong máy in.

➤ *Hệ màu CMYK:* trong đó K là độ đậm nhạt của màu  $K = \min(C, M, Y)$

$$P(x, y) = (C-K, M-K, V-K, K).$$

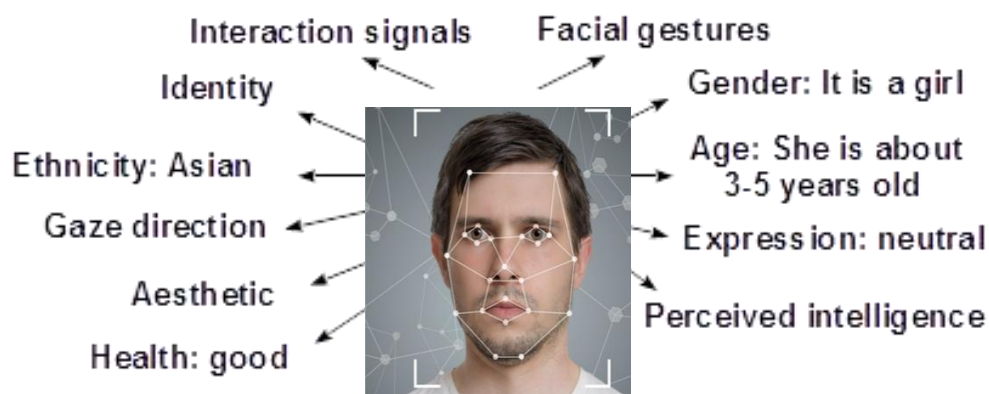
## 2.2. Tổng quan về bài toán nhận dạng mặt người

### 2.2.1. Giới thiệu.

Nhận dạng mặt người (Face recognition) là một chủ đề nghiên cứu thuộc lĩnh vực thị giác máy tính (Computer Vision) đã được phát triển từ đầu những năm 90 của thế kỷ trước [1]. Cho tới hiện nay, đây vẫn là một chủ đề nghiên cứu mở nhận được sự quan tâm của nhiều nhà nghiên cứu từ nhiều lĩnh vực nghiên cứu khác nhau như nhận dạng mẫu (Pattern Recognition), học máy (Machine Learning), thống kê (Statistics), sinh trắc học (Biometrics). Điều này là do có rất nhiều ứng dụng thực tế cần tới một hệ thống nhận dạng mặt, từ các hệ thống quản lý đăng nhập đơn giản cho tới các ứng dụng giám sát tại các địa điểm công cộng (public areas surveillance) hoặc quản lý dân số (population management) và pháp lý (forensics). Bên cạnh đó, so với các hệ thống nhận dạng dựa trên các đặc điểm sinh trắc học khác của con người, như nhận dạng móng mắt và vân tay (fingerprint and iris recognitions), dáng đi (gait recognition), nhận dạng mặt có nhiều ưu điểm:

+ Một hệ thống nhận dạng mặt không đòi hỏi có sự tương tác trực tiếp giữa đối tượng được nhận dạng và hệ thống.

+ Việc thu nhận dữ liệu (ảnh mặt) cho quá trình nhận dạng một con người dễ thực hiện hơn so với thu nhận các đặc điểm sinh trắc học khác (như thu nhận dấu vân tay và mống mắt).



Hình 2.5 Các thông tin có trong ảnh mặt người

+ Dữ liệu về mặt người phổ biến hơn so với các đặc trưng khác do sự bùng nổ các mạng xã hội (facebook, twitter ...), các dịch vụ chia sẻ dữ liệu đa phương tiện (youtube, vimeo ...) và sự phát triển mạnh mẽ của các thiết bị thu nhận hình ảnh.

+ Từ ảnh khuôn mặt của một người ta có thể khai thác nhiều thông tin liên quan chứ không chỉ là danh tính, chẳng hạn như giới tính (gender), màu da (skincolor), hướng nhìn (gaze direction), chủng tộc, hành vi, sức khỏe, độ tuổi, cảm xúc và mức độ thông minh ... như trong hình 2.1.

### 2.2.2. Thách thức trong nhận dạng mặt người

Tuy nhiên việc xây dựng một hệ thống nhận dạng mặt hoàn toàn tự động với khả năng nhận dạng chính xác cao thực sự là một thách thức đối với các nhà nghiên cứu. Điều này là do các yếu tố (chủ quan và khách quan) ảnh hưởng tới quá trình thu nhận ảnh và tạo ra các bức ảnh có độ khác biệt rất lớn của cùng một khuôn mặt. Có thể liệt kê ra đây các yếu tố chủ yếu ảnh hưởng tới độ chính xác của một hệ thống nhận dạng mặt [2]:

+ Ánh sáng (light conditions). Các bức ảnh mặt thu nhận ở các điều kiện sáng khác nhau sẽ rất khác nhau và làm giảm sự chính xác trong quá trình nhận dạng.

+ Thay đổi về tuổi (aging changes). Khuôn mặt người có các thay đổi lớn khi tuổi thay đổi và khó nhận dạng hơn ngay cả đối với hệ thống thị giác của con người.

+ Các vấn đề về hướng (pose variations). Việc nhận dạng với các ảnh có góc chụp thẳng (frontal) có kết quả tốt hơn rất nhiều so với các ảnh được chụp ở góc nghiêng lớn hơn  $45^\circ$ . Giải pháp thường thấy đối với các ảnh có hướng chụp lớn là sử dụng các thuật toán nội suy để cố gắng bù đắp phần khuôn mặt bị che khuất.

+ Cảm xúc (facial expression variations). Ở các trạng thái cảm xúc khác nhau, các đặc điểm quan trọng cho nhận dạng mặt (như mắt, mũi, mồm) có thể bị biến dạng (deformed) và dẫn tới các kết quả nhận dạng sai.

+ Che khuất (occlusions). Các ảnh mặt có thể bị che khuất bởi các yếu tố khách quan như vật chắn ở trước mặt hoặc chủ quan như các phụ kiện trên khuôn mặt (khăn, kính mắt) và làm cho quá trình nhận dạng bị sai.

Các hệ thống nhận dạng mặt được chia thành hai loại: xác định danh tính (face identification) và xác thực (face verification). Bài toán xác định danh tính là bài toán dạng 1-N trong đó hệ thống sẽ đưa ra kết quả là danh tính của ảnh được nhận dạng dựa trên sự tương đồng của ảnh input với một danh sách N ảnh đã biết danh tính chính xác. Trong khi đó, ở bài toán xác thực danh tính, hệ thống sẽ đưa ra câu trả lời đúng hoặc sai dựa vào việc xác định xem 2 bức ảnh có thuộc về cùng một người hay không. Trong phạm vi của đề tài này, chúng tôi chỉ tập trung vào bài toán xác định danh tính.

### **2.2.3. Các cách tiếp cận cho bài toán nhận dạng mặt người**

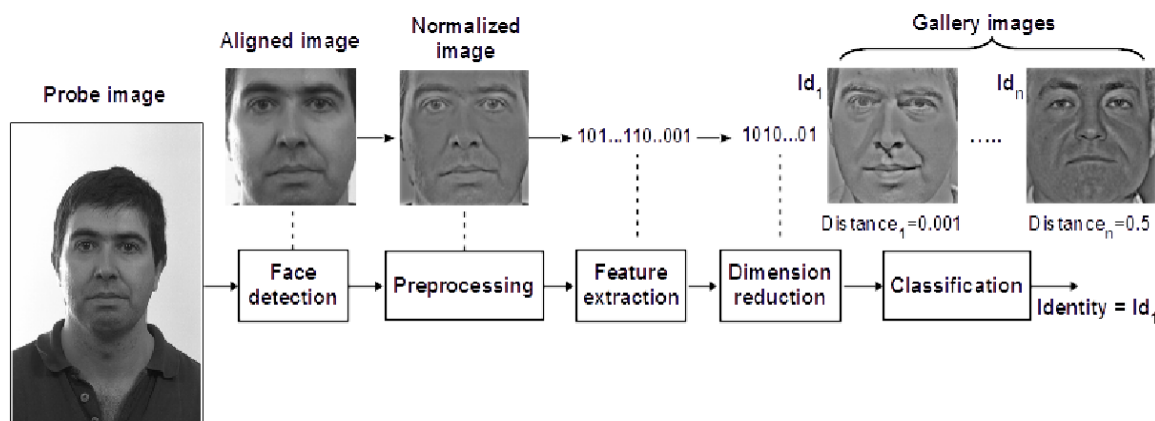
Dựa vào việc sử dụng các đặc trưng của khuôn mặt trong khi nhận dạng, các hệ thống nhận dạng mặt được chia thành hai hướng tiếp cận chính: hướng tiếp cận toàn cục (global approaches) và hướng tiếp cận cục bộ (local features-based approaches) [2]. Các phương pháp theo hướng tiếp cận toàn cục sẽ sử dụng các đặc điểm toàn cục của khuôn mặt (hình dạng, màu sắc, các đường nét chính ...) trong khi các hệ thống theo cách tiếp cận dựa vào các đặc trưng cục bộ sử dụng các đặc điểm cục bộ của khuôn mặt (các điểm ảnh, các chi tiết như mắt, mũi, mồm, lông mày ...) để nhận dạng. Trong số các hệ thống nhận dạng mặt dựa trên các đặc điểm toàn cục Eigengaces [3] và Fisherfaces [4] là các hệ thống tiêu biểu nhất. Eigenfaces sử dụng phép phân tích thành phần chính PCA (Principal Components

Analysis) để biểu diễn mỗi ảnh mặt như là một tổ hợp tuyến tính của các vector riêng thu được từ phép phân giải ma trận hiệp phương sai được tính từ các ảnh mặt sau khi đã chuẩn hóa. Do PCA là một thuật toán học không có giám sát nên không tận dụng được các thông tin về lớp khi tập ảnh huấn luyện có nhiều hơn 1 mẫu cho mỗi lớp nên trong Fisherfaces phương pháp phân tích tuyến tính khác biệt LDA (Linear Discriminant Analysis) đã được sử dụng nhằm khai thác tốt hơn các thông tin này. Về các hệ thống nhận dạng mặt dựa trên các đặc điểm cục bộ thì phương pháp mẫu nhị phân cục bộ LBP (Local Binary Pattern) [5] và biến đổi sóng nhỏ Gabor (Gabor wavelets) [6] là các kỹ thuật điển hình dùng để trích xuất các đặc điểm cục bộ từ ảnh mặt.

Các nghiên cứu cho thấy các hệ thống dựa trên các đặc điểm cục bộ cho kết quả tốt hơn các hệ thống theo cách tiếp cận toàn cục, đặc biệt là khi làm việc với các ảnh bị ảnh hưởng bởi các điều kiện như đã đề cập ở trên. Một cách tiếp cận khác là kết hợp cả hai đặc trưng cục bộ và toàn cục để nhận được một hệ thống lai ghép (hybrid system) nhằm đạt được hiệu quả cao hơn. Hệ thống được xây dựng trong tài liệu này dựa trên các đặc trưng cục bộ.

Về cơ bản, một hệ thống nhận dạng mặt người làm việc giống như hệ thống thị giác của con người khi cần nhận dạng (nhận ra) một ai đó khi nhìn vào một bức ảnh. Ở bước đầu tiên hệ thống cần xác định xem có mặt người trong bức ảnh input hay không, sau đó nếu có thì chỉ vùng ảnh đó sẽ được quan tâm, xử lý, bước này gọi là phát hiện khuôn mặt (face detection). Tiếp đến, các đặc điểm là đặc trưng nhất của khuôn mặt (giúp phân biệt ảnh mặt người này với người khác và các ảnh thuộc về cùng một người) sẽ được trích chọn (extract) để nhận dạng, bước này gọi là trích chọn đặc trưng (feature extraction). Trong bước tiếp theo, gọi là nhận dạng (recognition), hệ thống sẽ đối sánh các đặc trưng vừa nhận được với một kho dữ liệu các đặc trưng của những người đã biết rõ danh tính để xác định xem các đặc trưng đó là của khuôn mặt này và đưa ra kết luận về danh tính tương ứng với khuôn mặt có tỉ lệ tương đồng lớn nhất. Tỉ lệ các ảnh nhận dạng đúng danh tính (so với dữ liệu thực tế) trên tổng số ảnh cần nhận dạng gọi là tỉ lệ nhận

dạng (recognition rate) và được dùng để đánh giá hiệu năng (độ chính xác) của một hệ thống nhận dạng.



Hình 2.6 Các bước trong một hệ thống nhận dạng mặt.

Trên thực tế, việc xây dựng một hệ thống nhận dạng mặt sẽ gồm nhiều bước chi tiết hơn như được chỉ ra trong hình 2.2. Trước bước trích chọn đặc trưng, một thuật toán tiền xử lý (preprocessing) sẽ được sử dụng để thực hiện lọc nhiễu và chuẩn hóa các điều kiện ánh sáng của các bức ảnh sao cho chúng có cùng điều kiện ánh sáng nhất có thể. Tiếp đó, sau bước trích chọn đặc trưng, mỗi ảnh mặt sẽ được biểu diễn dưới dạng một vector đặc trưng có số chiều lớn (từ vài nghìn cho tới hàng trăm nghìn) và chứa nhiều thông tin dư thừa, không cần thiết cho quá trình nhận dạng. Do đó, một thuật toán học sẽ được sử dụng để loại bỏ các đặc điểm dư thừa và đồng thời làm tăng tính phân biệt (discriminative) của các thuộc tính được giữ lại nhằm làm tăng hiệu năng của hệ thống về cả tốc độ xử lý (do chỉ cần làm việc với các vector có số chiều nhỏ hơn rất nhiều) và độ chính xác. Cuối cùng, ở bước nhận dạng (classification stage), có nhiều thuật toán có thể sử dụng, chẳng hạn như bộ phân lớp k-láng giềng gần nhất (k Nearest Neighbors – k-NN), SVM (Support Vector Machine) hoặc mạng nơron (neural network).

Do một hệ thống nhận dạng mặt gồm nhiều bước nên các nghiên cứu về nhận dạng mặt cũng chia thành các lĩnh vực nhỏ hơn chỉ tập trung vào các phương pháp để giải quyết từng bước trong cả hệ thống. Vì thế, mặc dù đã có nhiều nghiên

cứu về nhận dạng mặt, song việc xây dựng hoàn chỉnh một hệ thống hoàn toàn tự động từ bước đầu tiên tới cuối cùng vẫn còn chưa nhiều.

### 2.3. Phương pháp mô tả đặc trưng Histogram of Oriented Gradient (HOG).

Các đặc trưng HOG [7] ban đầu được thiết kế để phát hiện người (human detection) trong dữ liệu ảnh nhưng sau đó được mở rộng và được sử dụng rộng rãi cho các bài toán phát hiện đối tượng nói chung (object detection). Phương pháp HOG dựa trên việc đếm số lần xuất hiện của các hướng đạo hàm (gradient orientation) trong các vùng cục bộ của ảnh.

Bản chất của phương pháp HOG là các thông tin về hình dáng và vẻ bề ngoài của các đối tượng cục bộ trong ảnh có thể được mô tả bằng cách sử dụng thông tin về sự phân bố của các cường độ gradient (intensity gradients) hoặc của các hướng biên (edge directions). Các toán tử HOG được cài đặt bằng cách chia nhỏ một bức ảnh thành các vùng con được gọi là các “tế bào” (cells) và với mỗi cell đó một histogram về các hướng của gradients sẽ được tính cho các điểm nằm trong cell. Ghép các histogram này lại ta sẽ có một biểu diễn cho bức ảnh ban đầu. Để tăng cường hiệu năng nhận dạng, các histogram cục bộ có thể được chuẩn hóa về độ tương phản bằng cách tính một ngưỡng cường độ trong một vùng lớn hơn của ảnh, gọi là các khối (blocks) và sử dụng giá trị ngưỡng đó để chuẩn hóa tất cả các cell trong khối. Kết quả của bước chuẩn hóa này là vector đặc trưng sẽ có tính bất biến cao hơn đối với các thay đổi về điều kiện ánh sáng.

Các bước cơ bản của việc tính một vector HOG cho một ảnh gồm có: Tính gradient, gán hướng vào các vùng giá trị khác nhau, tính histogram của các khối và cuối cùng là chuẩn hóa các khối.

**Tính gradient:** Đây là bước đầu tiên trong phương pháp HOG và thường được thực hiện bằng hai phép nhân chập ảnh gốc với 2 nhân 1 chiều tương ứng với các toán tử lấy đạo hàm ảnh theo hai hướng Ox và Oy, cụ thể là:

$$D_x = [-1 \ 0 \ 1] \text{ và } D_y = [1 \ 0 \ -1]^T,$$



trong đó  $T$  là ký hiệu của phép chuyển vị ma trận.

Với một ảnh input  $I$ , ta sẽ có 2 ảnh đạo hàm riêng theo 2 hướng được tính

bằng các công thức:  $I_x = I * D_x$  và  $I_y = I * D_y$ .

Thành phần cường độ (magnitude) và hướng (orientation) sẽ được tính theo các công thức:  $|G| = \sqrt{I_x^2 + I_y^2}$  và  $\theta = \arctan(I_y, I_x)$ . Ví dụ về kết quả của phép lấy đạo hàm có thể xem ở hình 2.3.

**Gán hướng:** Sau khi tính đạo hàm, ta có thành phần hướng ở các điểm ảnh trong mỗi cell của ảnh input. Các giá trị hướng này được chia thành các vùng hướng đều nhau trong khoảng từ 0 tới  $360^\circ$ . Theo các tác giả của HOG, thì 9 vùng chia cho kết quả tốt nhất đối với các bài toán nhận dạng. Có nghĩa là các giá trị hướng nằm trong khoảng  $[0, 360/9)$  sẽ được gán bằng 1 (coi là như nhau), các giá trị



Hình 2.7 Ảnh input và hai đạo hàm của nó.

hướng trong khoảng  $[360/9, 2*360/9)$  sẽ được gán bằng 2, vân vân. Sau đó histogram của mỗi cell sẽ được tính độc lập.

**Tính histogram của các khối:** Để đáp ứng tốt hơn với các thay đổi về điều kiện ánh sáng và độ tương phản, các giá trị gradient cần phải được chuẩn hóa cục bộ, điều này đòi hỏi việc nhóm các cell thành các vùng liên kết không gian với nhau gọi là các khối. Vector HOG do đó sẽ là tổng hợp các histogram của các cell đã được chuẩn hóa từ tất cả các khối. Các khối sẽ được chia và không tách biệt

nhau, nói cách khác mỗi cell sẽ đóng góp thông tin về các hướng (tại các điểm ảnh của nó) nhiều hơn 1 lần.

Để tính toán vector đặc trưng cho từng ô (cell), chúng ta cần chia hình ảnh thành các block, mỗi block lại chia đều thành các cell. Để xác định được số block, chúng ta sẽ sử dụng công thức sau:

$$n_{block\_image} = \left( \frac{W_{image} - W_{block} * W_{cell}}{W_{cell}} \right) * \left( \frac{H_{image} - H_{block} * H_{cell}}{H_{cell}} + 1 \right) \quad (2.4)$$

$W_{image}, W_{block}, W_{cell}$ : lần lượt là chiều rộng của ảnh, block và cell

$H_{image}, H_{block}, H_{cell}$ : lần lượt là chiều dài của ảnh, block và cell

Sau khi xác định số block và kích thước mỗi block, cell, để tính toán vector đặc trưng cho từng cell, chúng ta cần:

- Chia không gian hướng thành p bin (số chiều vector đặc trưng của ô)
- Rời rạc hóa góc hướng nghiêng tại mỗi điểm ảnh vào trong các bin.

Giả sử góc hướng nghiêng tại pixel ở vị trí (x, y) có độ lớn là alpha (x, y):

- Trường hợp rời rạc hóa unsigned-HOG với  $p = 9$

$$B_{(x,y)} = \text{round} \left( \frac{\rho * \alpha(x,y)}{\pi} \right) \bmod p \quad (2.5)$$

- Trường hợp rời rạc hóa signed-HOG với  $p = 18$

$$B_{(x,y)} = \text{round} \left( \frac{\rho * \alpha(x,y)}{2\pi} \right) \bmod p \quad (2.6)$$

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

$$\text{size}_{block} = n * \text{size}_{cell} \quad (2.7)$$

$n$ : số ô (cell) trong một khối (block)

$\text{size}_{cell}$  là số chiều của vector đặc trưng ô ( $\text{size}_{cell} = 9$  nếu sử dụng unsigned-HOG và  $\text{size}_{cell} = 18$  nếu sử dụng signed-HOG)

**Chuẩn hóa khối:** Có nhiều phương pháp có thể được dùng cho việc chuẩn hóa khối. Gọi  $v$  là vector cần chuẩn hóa chứa tất cả các histogram của một khối,  $\|v_k\|$  là giá trị chuẩn của nó theo các chuẩn  $k=1, 2$  và  $e$  là một hằng số nhỏ, khi đó các giá trị chuẩn hóa có thể tính bằng 1 trong các công thức sau:

$$L2-norm: f = \frac{v}{\sqrt{\|v\|_2^2 + e^2}} \quad (2.8)$$

$$L1-norm: f = \frac{v}{\sqrt{\|v\|_1 + e}} \quad (2.9)$$

$$L1-sqrt: f = \sqrt{\frac{v}{\|v\|_1 + e}} \quad (2.10)$$



Hình 2.8 Một ảnh sau khi trích chọn đặc trưng HOG

#### 2.4. Support vector machine (SVM)

SVM là một phương pháp học có giám sát trong các mô hình nhận dạng mẫu. Nó không chỉ hoạt động tốt với các dữ liệu được phân tách tuyến tính mà còn tốt với cả dữ liệu phân tách phi tuyến. Với nhiều bài toán, SVM mang lại kết quả tốt như mạng nơ-ron với hiệu quả sử dụng tài nguyên tốt hơn hẳn.

Ý tưởng của SVM là đi tìm siêu mặt phẳng để phân tách các điểm dữ liệu.

Trong không gian nhiều chiều, khoảng cách từ một điểm (vector) có tọa độ  $x_0$  tới siêu mặt phẳng (hyperplane) có phương trình  $w^T x + b = 0$  được xác định

bởi:

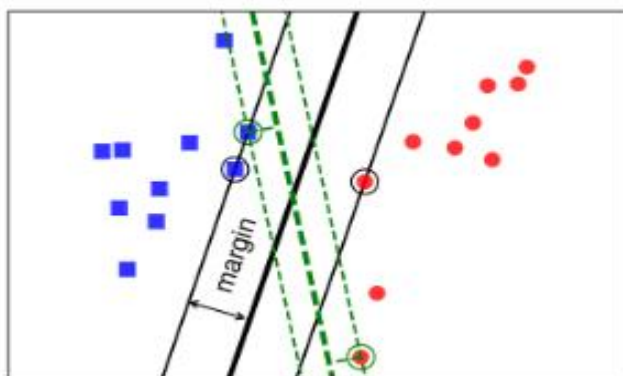
$$\frac{|w^T x_0 + b|}{\|w\|_2} \quad (2.11)$$

Với  $\|w\|_2$  là độ lớn của vector  $w$  tính theo norm 2.

Khi đó dấu của hàm ước lượng sẽ thể hiện được điểm dữ liệu  $x$  nằm ở cụm dữ liệu nào (H)

$$H = \{x \mapsto \text{sgn}(w^T x + b) ; w \in \mathbb{R}^N, b \in \mathbb{R}\} \quad (2.12)$$

SVM có vai trò đi tìm đường phân chia dữ liệu tốt nhất và chỉ duy nhất 1 đường thẳng thỏa mãn luật của SVM, là khoảng cách từ đường thẳng đó đến điểm gần nhất của mỗi lớp dữ liệu (margin) là như nhau và lớn nhất. Thuật toán đi tìm siêu mặt phẳng thông qua các bước tính toán phức tạp, mục đích tối ưu hóa khoảng cách mặt phân chia dữ liệu. Từ đó, tìm ra được một bộ phân lớp chính xác và tối ưu nhất.



Hình 2.9 SVM tìm mặt phẳng phân chia dữ liệu ứng với margin lớn nhất

**Phát hiện mặt người với các đặc trưng HOG và SVM:** để phát hiện mặt người với các đặc trưng HOG và SVM ta tiến hành các bước như sau:

**Bước 1:** Chuẩn bị  $P$  mẫu là ảnh mặt người và trích xuất các vector đặc trưng HOG từ các bức ảnh này.

**Bước 2:** Chuẩn bị  $N$  mẫu không phải là ảnh mặt người ( $N$  rất lớn so với  $P$ ) và trích xuất các vector HOG từ các ảnh này.

**Bước 3:** Sử dụng một bộ phân loại SVM tuyến tính để học với các vector của các mẫu tích cực (là ảnh mặt người) và tiêu cực (các ảnh không phải mặt người) đã chuẩn bị.

**Bước 4:** Đối với mỗi bức ảnh trong bộ ảnh tiêu cực, sử dụng một cửa sổ trượt di chuyển qua tất cả các vị trí có thể của ảnh input. Tại mỗi vị trí của cửa sổ trượt, tính vector HOG của cửa sổ và đưa vào bộ phân lớp. Nếu bộ phân lớp phân lớp sai một cửa sổ là ảnh mặt thì ghi lại vector tương ứng cùng với xác suất phân lớp.

**Bước 5:** Lấy các mẫu nhận dạng sai ở bước 4 và sắp xếp chúng theo mức xác suất nhận dạng sai và cho bộ phân lớp học lại sử dụng các mẫu sai này.

**Bước 6:** Áp dụng bộ phân lớp đã được học lại với các ảnh cần phát hiện mặt người.

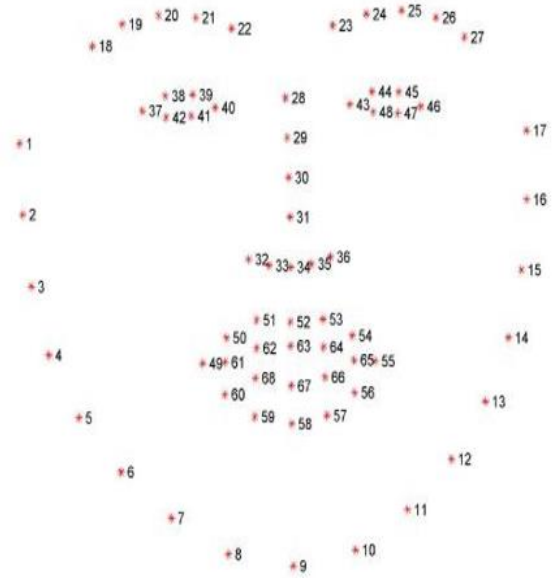
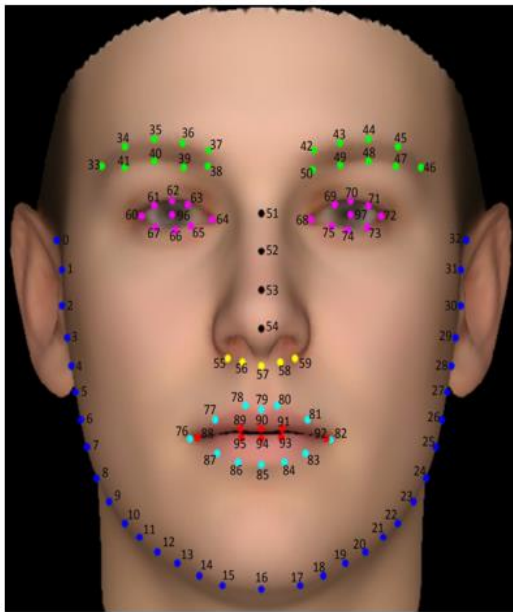


Hình 2.10 Ảnh input và kết quả phát hiện và căn chỉnh ảnh mặt với HOG và SVM.

Trong đề tài này, toàn bộ phần phát hiện mặt người sử dụng các dữ liệu phân lớp đã được học trước được cung cấp bởi thư viện dlib. Kết quả của bước này có thể xem ở các hình 2.10.

## 2.5. Phân tích điểm đặc trưng trên khuôn mặt người - Facial Landmark

Xác định facial landmark là một bài toán con của bài toán dự đoán hình dạng (shape prediction). Vậy bài toán dự đoán hình dạng là gì? Đó chính là việc chúng ta phải xác định được những điểm chính tạo nên hình dạng của đối tượng trong một bức ảnh. Trong bài toán xác định facial landmark, chúng ta sẽ phải xác định được những điểm chính trong bức ảnh tạo nên hình dạng khuôn mặt người. Facial landmark là đầu vào cho nhiều bài toán khác như dự đoán tư thế đầu, trao đổi khuôn mặt, phát hiện nháy mắt, xoay chỉnh lại khuôn mặt và điển hình là công nghệ nhận dạng khuôn mặt FaceID được Apple trang bị trên iPhone X.

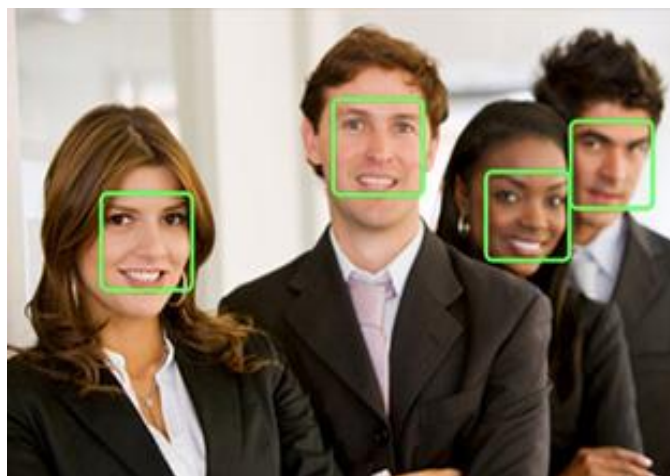


Hình 2.11 68 điểm đặc trưng trên khuôn mặt người

Việc xác định facial landmark gồm có hai bước:

**Bước 1:** Xác định được vị trí khuôn mặt trong bức ảnh

Việc xác định vị trí khuôn mặt có thể được thực hiện bằng nhiều cách từ đơn giản như thuật toán Haar cascades đến phức tạp như các thuật toán dựa trên deep-learning, mục đích cuối cùng là ta sẽ thu được một vùng xác định bởi tọa độ (x,y) bao quanh khuôn mặt trong bức ảnh.

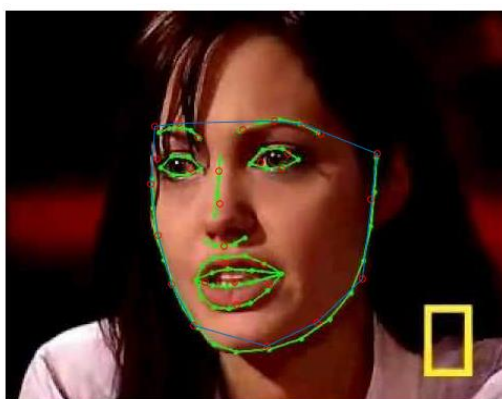


Hình 2.12 Xác định mặt người trong tấm ảnh

## Bước 2: Xác định facial landmark

Sau khi xác định được khuôn mặt trong bức ảnh, chúng ta sẽ xác định cấu trúc của khuôn mặt. Có rất nhiều kiểu cấu trúc khuôn mặt khác nhau nhưng về cơ bản, chúng ta sẽ phải xác định được những phần sau:

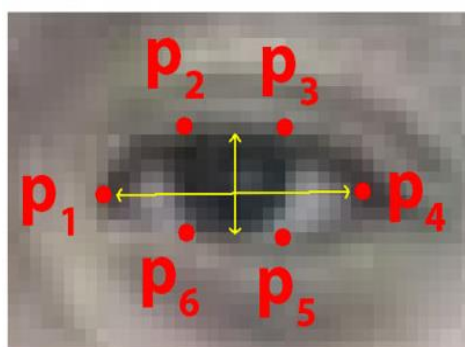
- Miệng
- Lông mày phải - Lông mày trái
- Mắt phải - Mắt trái
- Mũi
- Hàm



Hình 2.13 Xác định 68 điểm đặc trưng trên khuôn mặt dùng OpenCV

### 2.6. Tỷ lệ nhắm mở mắt – Eyes Aspect Ratio (EAR)

Mỗi mắt được biểu thị bằng 6 điểm  $(x, y)$  theo thứ tự bắt đầu từ góc trái của mắt và sắp xếp theo chiều kim đồng hồ thành các điểm  $p_1, p_2, p_3, p_4, p_5, p_6$  như hình bên dưới:



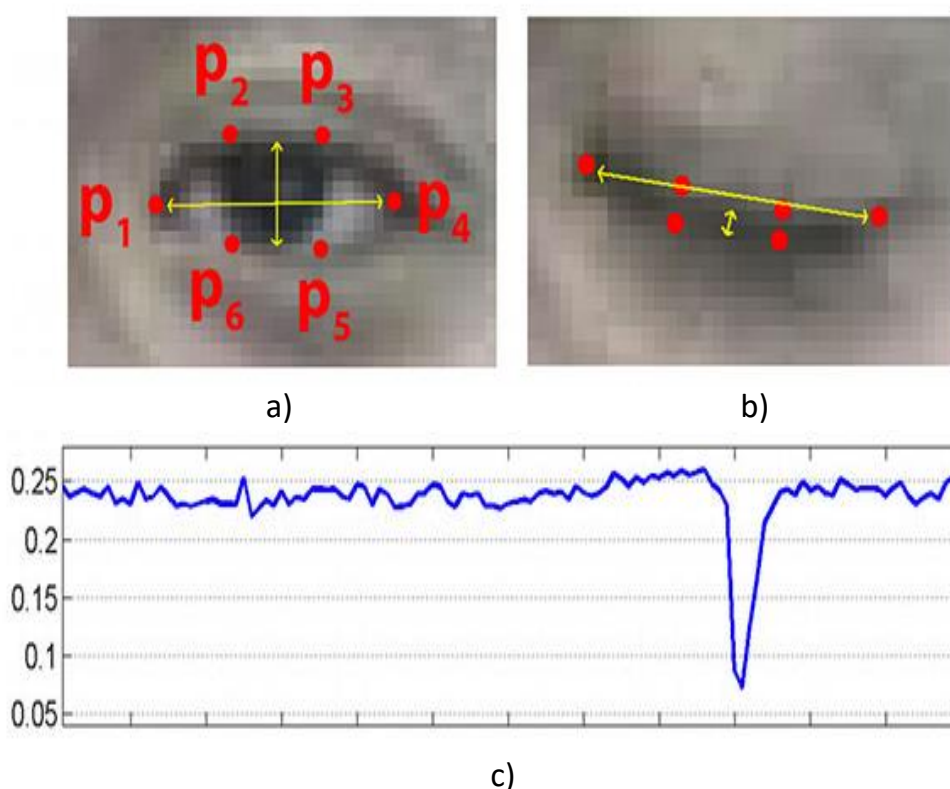
Hình 2.14 Các điểm xác định vị trí mắt



### Công thức tính tỉ lệ nhắm mở mắt:

$$EAR = \frac{\|p_2 - p_6\| + \|p_3 - p_5\|}{2\|p_1 - p_4\|} \quad (2.13)$$

Trong đó  $p_1, \dots, p_6$  là các vị trí mốc 2D trên khuôn mặt.



Hình 2.15 Xác định vị trí vùng mắt trên khuôn mặt và đồ thị thời gian nhắm mở mắt.

\* Khi mắt mở bình thường như hình a) thì theo công thức trên tỉ lệ nhắm mở mắt (EAR) sẽ lớn vì khoảng cách giữa  $p_2$ - $p_6$  cũng như  $p_3$ - $p_5$  lớn, trong khi khoảng cách giữa  $p_1$ - $p_4$  gần như không đổi với một mắt người cụ thể. Tuy nhiên, khi người đó chớp mắt (như trong hình b) thì tỷ lệ giảm đáng kể, gần như bằng không.

\* Đồ thị bên dưới biểu diễn tỉ lệ nhắm mở mắt theo thời gian trong 1 video clip. Ban đầu khi mở mắt, tỉ lệ này duy trì sấp xỉ là 0.25, sau đó nhanh chóng giảm xuống gần bằng không, rồi tăng trở lại, cho thấy một lần chớp mắt vừa diễn ra.

+ Tính toán độ mở của mắt tài xế bằng phương pháp Eyes Aspect Ratio (EAR), xác định thời gian nhắm mắt, đưa ra trạng thái ngủ gật.



## 2.7. Mạng neural nhân tạo

### 2.7.1. Khái niệm chung

Mạng neural nhân tạo, Artificial Neural Network (ANN) gọi tắt là mạng neural, neural network, là một mô hình xử lý thông tin phỏng theo cách thức xử lý thông tin của các hệ neural sinh học. Nó được tạo lên từ một số lượng lớn các phần tử (gọi là phần tử xử lý hay neural) kết nối với nhau thông qua các liên kết (gọi là trọng số liên kết) làm việc như một thể thống nhất để giải quyết một vấn đề cụ thể nào đó.

Một mạng neural nhân tạo được cấu hình cho một ứng dụng cụ thể (nhận dạng mẫu, phân loại dữ liệu, ...) thông qua một quá trình học từ tập các mẫu huấn luyện. Về bản chất học chính là quá trình hiệu chỉnh trọng số liên kết giữa các neural.

Các nghiên cứu về mạng thần kinh nhân tạo đã bắt đầu từ thập niên 1940. Năm 1944, McCulloch và Pitts công bố công trình nghiên cứu về liên kết của các tế bào thần kinh. Năm 1949, Hebb công bố về tính thích nghi của mạng thần kinh. Cuối những năm 1950, Rosenblatt đưa ra mạng Perceptron. Năm 1985 mạng Hopfield ra đời và sau đó là mạng lan truyền ngược.

Từ cuối thập niên 80, đầu thập niên 90 đến nay, mạng neural đã khẳng định được vị trí của mình trong nhiều ứng dụng khác nhau:

- *Lĩnh vực hàng không, vũ trụ*: ứng dụng trong kỹ thuật thiết kế máy bay không người lái, mô phỏng đường bay, hệ thống điều khiển máy bay, nâng cao khả năng bay tự động, tìm kiếm lỗi, hư hỏng trong máy bay, mô hình hóa các bộ phận của máy bay.

- *Ngân hàng*: kiểm tra thẻ tín dụng, định giá độ tin cậy của đầu tư

- *Trong hình sự*: phát hiện và so sánh dấu vân tay, nhận biết tiếng nói.

- *Quốc phòng*: điều khiển tên lửa, các thiết bị bay, hệ thống định vị như sonar, radar, các loại cảm biến mới.

- *Lĩnh vực điện*: dự báo phụ tải, chế tạo chip trong các mạch tích hợp, chip phân tích, phương pháp điều khiển động cơ.

- *Trong y học*: phân tích và phát hiện tế bào ung thư, lưu giữ thời gian nằm và ra viện của bệnh nhân.

- *Trong sản xuất*: kiểm tra theo dõi quá trình sản xuất, thử nghiệm phân tích, phân loại sản phẩm, dự báo, lên kế hoạch và điều khiển quá trình sản xuất.

#### **2.7.1.1. Ưu điểm**

- Có khả năng tự thích nghi, tự học cao bằng cách tự tổ chức lại mạng lưới.
- Tổng quát quá thành quy luật từ tập dữ liệu để xây dựng hệ thống luật dùng cho các phương pháp khác.
- Phù hợp với dạng dữ liệu số, có tính liên tục.

#### **2.7.1.2. Nhược điểm**

- Mạng chỉ có thể làm việc với những dữ liệu số.
- Để mạng đạt hiệu quả cao cần phải có dữ liệu mẫu đủ lớn (gồm tất cả các mẫu đặc trưng).
- Mạng chỉ có tính nội suy, khả năng ngoại suy kém.
- Mạng không thể đưa ra cơ chế giải thích.
- Các giải thuật học của mạng chưa đảm bảo hội tụ cần thiết.

### **2.7.2. Mạng nơron và mạng nơron lan truyền ngược**

#### **2.7.2.1. Giới thiệu về mạng Nơron**

Đặc điểm bộ não người : hệ xử lý thông tin phức tạp, phi tuyến và song song, có khả năng học, ghi nhớ, tổng quát hóa, xử lý lỗi và có khoảng  $10^{11}$  tế bào thần kinh.

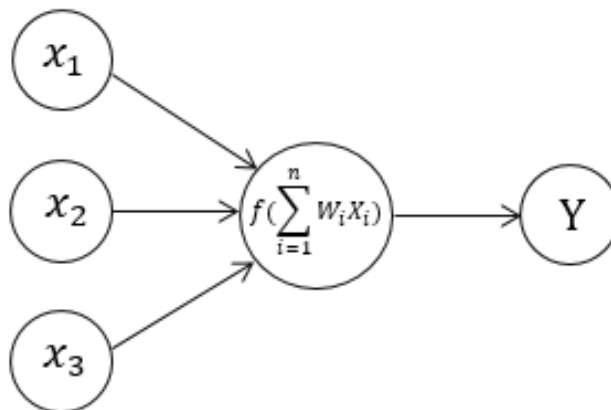
Tế bào thần kinh sinh học gồm : đầu dây thần kinh, thân tế bào, sợi thần kinh.

Tế bào thần kinh nhân tạo : được biết như là mô hình toán học đơn giản của bộ não con người.

Đặc điểm : Cần được huấn luyện trước khi sử dụng, các nghiên cứu về lĩnh vực này bắt từ 1940. Hiện nay có rất nhiều mạng và các thuật toán huấn luyện tương ứng được công bố để giải quyết các bài toán khác nhau.

Mạng thần kinh nhân tạo : Nó bao gồm các tế bào thần kinh kết nối với nhau bởi các liên kết, mỗi liên kết kèm theo một trọng số. Các vector tín hiệu ngõ vào và vector trọng số của tế bào thần kinh.

Ta có thể định nghĩa như sau : Mạng nơron (thần kinh) nhân tạo, Artificial Neural Network (ANN) là một mô hình xử lý thông tin phỏng theo cách thức xử lý thông tin của các hệ nơron sinh học. Nó được tạo nên từ một số lượng lớn các phần tử (nơron) kết nối với nhau thông qua các liên kết (trọng số liên kết) làm việc như một thể thống nhất để giải quyết một vấn đề cụ thể nào đó. Một mạng nơron nhân tạo được cấu hình cho một ứng dụng cụ thể (nhận dạng mẫu, phân loại dữ liệu, ...) thông qua một quá trình học từ tập các mẫu huấn luyện. Về bản chất học chính là quá trình hiệu chỉnh trọng số liên kết giữa các nơron.



Hình 2.16: Mạng thần kinh nhân tạo

**Các thành phần cơ bản của một nơron nhân tạo bao gồm:**

- Tập các đầu vào: là các tín hiệu vào của nơron, các tín hiệu này thường được đưa vào dưới dạng một vector N chiều.
- Tập các liên kết: mỗi liên kết được thể hiện bởi một trọng số liên kết – Synaptic weight. Trọng số liên kết giữa tín hiệu vào thứ j với nơron k thường được kí hiệu là  $W_{kj}$ . Thông thường, các trọng số này được khởi tạo một cách ngẫu nhiên ở thời điểm khởi tạo mạng và được cập nhật liên tục trong quá trình học mạng.

- Bộ tổng (Summing function): thường dùng để tính tổng của tích các đầu vào với trọng số liên kết của nó.
- Ngưỡng (còn gọi là một độ lệch - bias): ngưỡng này thường được đưa vào như một thành phần của hàm truyền.
- Hàm truyền (Transfer function): hàm này được dùng để giới hạn phạm vi đầu ra của mỗi nơron. Nó nhận đầu vào là kết quả của hàm tổng và ngưỡng.
- Đầu ra: là tín hiệu đầu ra của một nơron, với mỗi nơron sẽ có tối đa là một đầu ra.
- Xét về mặt toán học, cấu trúc của một nơron k, được mô tả bằng biểu thức sau:

Hàm xử lý ngõ vào:

- Hàm tuyến tính:

$$f = net_i = \left( \sum_{j=1}^n w_{ij} x_j \right) - \theta_i = W^T X - \theta \quad (26)$$

- Hàm toàn phương:

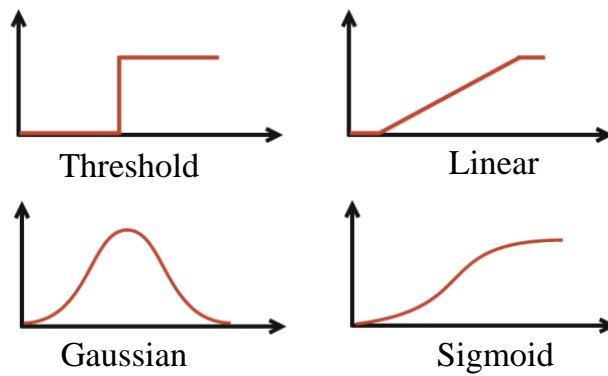
$$f = net = \left( \sum_{j=1}^n w_{ij} x_{ij}^2 \right) - \theta_i \quad (27)$$

- Hàm cầu:

$$f = net = \left( \rho^{-2} \sum_{j=1}^n (w_{ij} - x_{ij})^2 \right) - \theta_i = \rho^{-2} (X - W)^T (X - W) - \theta \quad (28)$$

Hàm xử lý ngõ ra \_ Hàm kích hoạt:  $Y = a(f)$ , bao gồm:

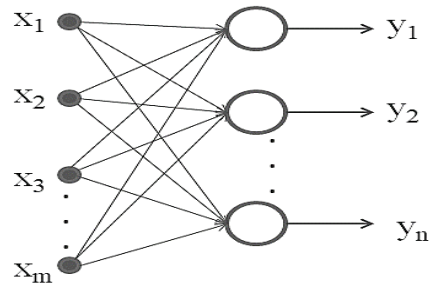
- Hàm nấc.
- Hàm dấu.
- Hàm tuyến tính.
- Hàm dốc bão hòa.
- Hàm tuyến tính bão hòa.
- Hàm dạng S.



Hình 2.17 Một số hàm tác động.

### 2.7.3. Một số kiểu mạng nơron

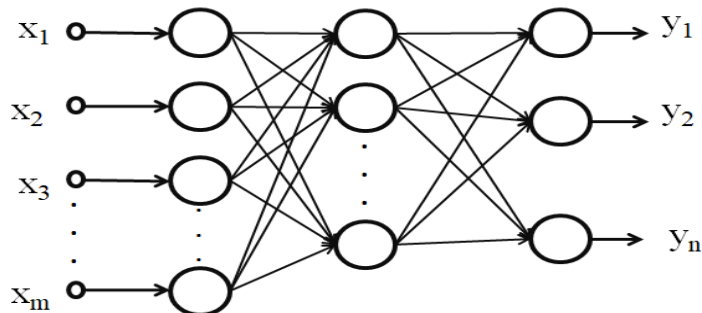
#### 2.7.3.1. Mạng truyền thẳng một lớp



Hình 2.18 Mạng truyền thẳng một lớp.

Là mạng chỉ gồm một lớp tế bào thần kinh xử lý. Tín hiệu chỉ truyền theo một chiều từ ngõ vào đến ngõ ra.

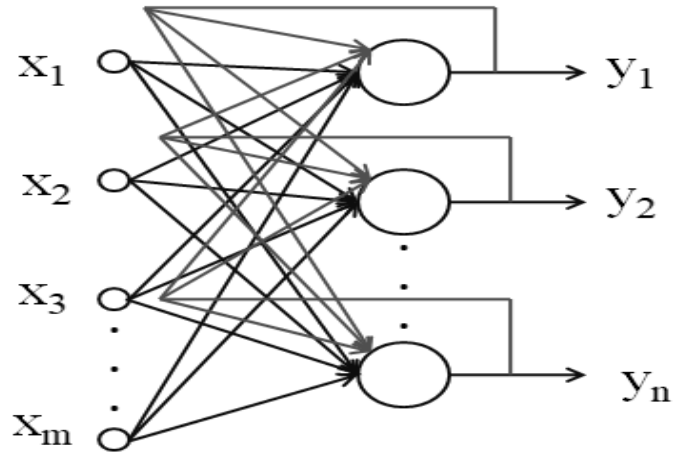
#### 2.7.3.2. Mạng truyền thẳng nhiều lớp



Hình 2.19 Mạng truyền thẳng nhiều lớp.

Là mạng gồm nhiều lớp tế bào thần kinh xử lý. Tín hiệu chỉ truyền theo một chiều từ ngõ vào đến ngõ ra.

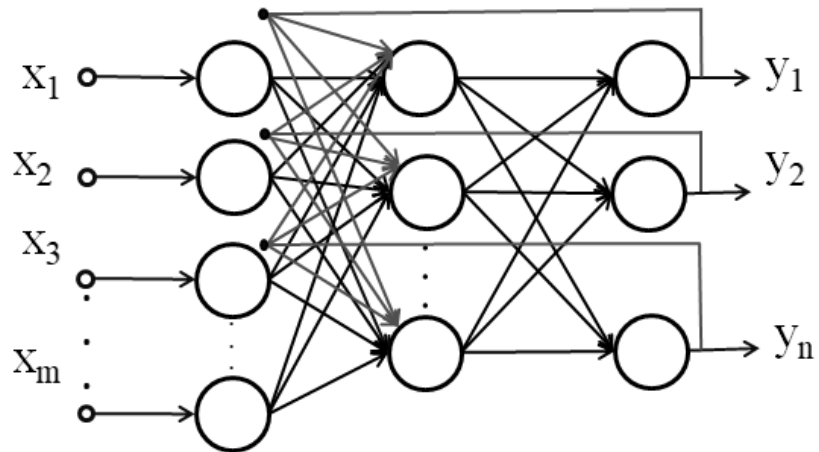
### 2.7.3.3. Mạng hồi quy một lớp



Hình 2.20 Mạng hồi quy một lớp.

Là mạng chỉ gồm một lớp trong đó có tín hiệu hồi tiếp từ ngõ ra trở về ngõ vào.

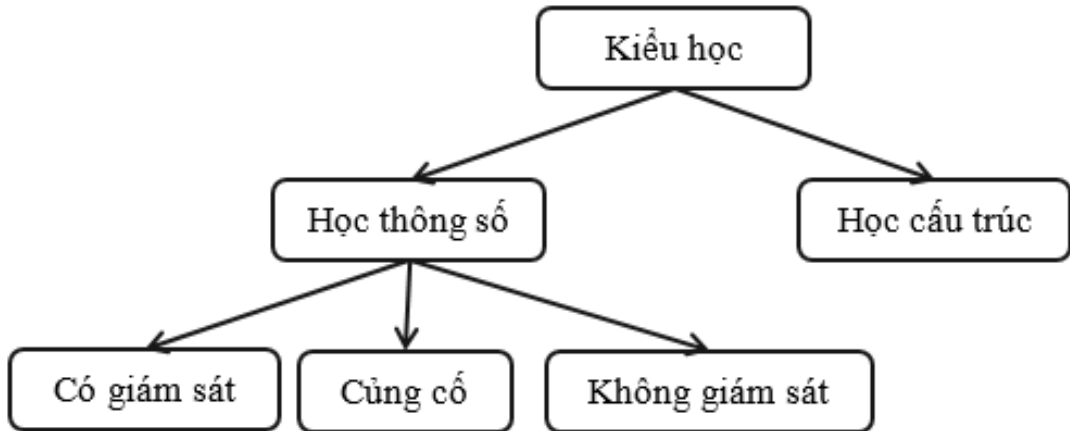
### 2.7.3.4. Mạng hồi quy nhiều lớp



Hình 2.21 Mạng hồi quy nhiều lớp.

Là mạng gồm nhiều lớp tế bào thần kinh xử lý. Trong đó có tín hiệu hồi tiếp từ ngõ ra trở về ngõ vào.

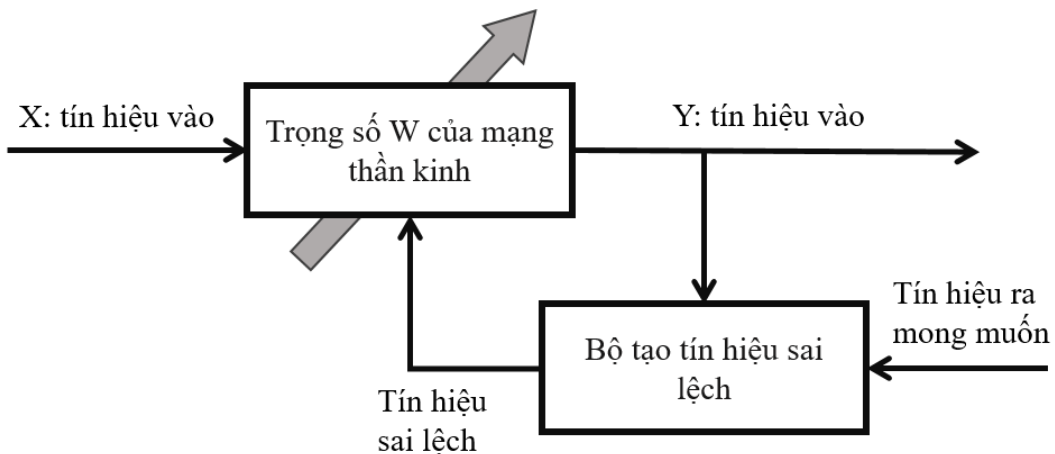
#### 2.7.4. Các phương pháp huấn luyện mạng thần kinh nhân tạo



Hình 2.22 Phân loại kiểu học.

##### 2.7.4.1. Học có giám sát

Học có giám sát: Là quá trình học có sự tham gia giám sát của một “thầy giáo”. Cũng giống như việc ta dạy một em nhỏ các chữ cái. Ta đưa ra một chữ “a” và bảo với em đó rằng đây là chữ “a”. Việc này được thực hiện trên tất cả các mẫu chữ cái. Sau đó khi kiểm tra ta sẽ đưa ra một chữ cái bất kì (có thể viết hơi khác đi) và hỏi em đó đây là chữ gì?

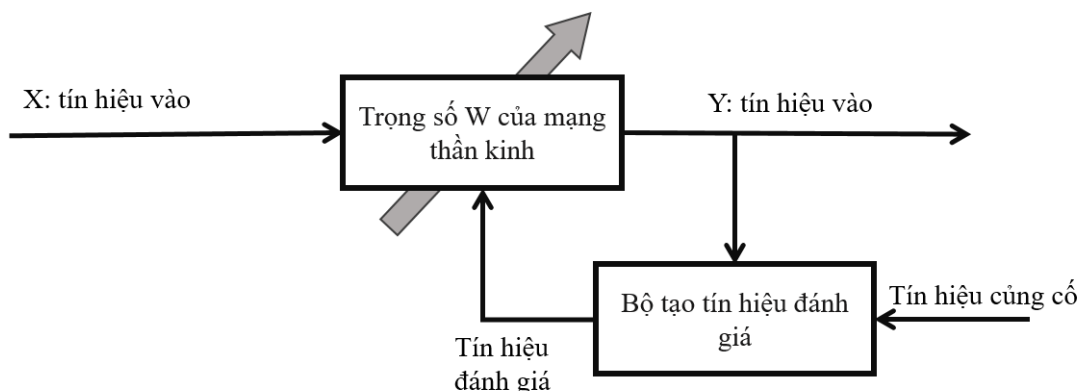


Hình 2.23 Học có giám sát

Như vậy với học có giám sát, số lớp cần phân loại đã được biết trước. Nhiệm vụ của thuật toán là phải xác định được một cách thức phân lớp sao cho với mỗi vector đầu vào sẽ được phân loại chính xác vào lớp của nó.

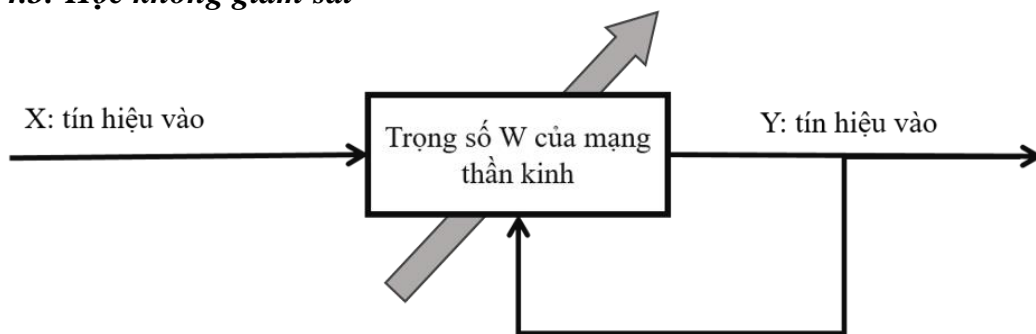
### 2.7.4.2. Học củng cố

Học củng cố: đôi khi còn được gọi là học thưởng-phạt (rewardpenalty learning), là sự tổ hợp của cả hai mô hình trên. Phương pháp này cụ thể như sau: với vector đầu vào, quan sát vector đầu ra do mạng tính được. Nếu kết quả được xem là “tốt” thì mạng sẽ được thưởng theo nghĩa tăng các trọng số kết nối lên; ngược lại mạng sẽ bị phạt, các trọng số kết nối không thích hợp sẽ được giảm xuống.



Hình 2.24 Học củng cố

### 2.7.4.3. Học không giám sát



Hình 2.25 Học không giám sát

## 2.8. Ngôn ngữ lập trình 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.

**Python** đã được hình thành vào cuối những năm 1980, và việc thực hiện nó vào tháng 12 năm 1989 bởi **Guido van Rossum** tại **Centrum Wiskunde & Informatica (CWI)** ở **Hà Lan** như là một kế thừa cho ngôn ngữ **ABC** (tự lấy cảm hứng từ SETL) có khả năng xử lý ngoại lệ và giao tiếp với Hệ điều hành Amoeba. **Van Rossum** là tác giả chính của **Python**, và vai trò trung tâm của ông trong việc quyết định hướng phát triển của **Python**.

**Python 2.0** đã được phát hành vào ngày 16 tháng 10 năm 2000 và có nhiều tính năng mới, bao gồm bộ thu gom rác theo chu kỳ (cycle-detecting garbage) và hỗ trợ Unicode. Với việc phát hành này quá trình phát triển đã được thay đổi và trở nên minh bạch hơn và cộng đồng hậu thuẫn

**Python 3.0** được phát hành năm 2008, sau một thời gian dài thử nghiệm.

Cho tới năm 2017, Python đang có phiên bản 3.7

**Sau đây là 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 module 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 IronPython).

Có thể nhúng vào ứng dụng như một giao tiếp kịch bản (scripting interface).

*Python dễ dàng kết nối với các thành phần khác:*

Python có thể kết nối với các đối tượng COM, .NET (Ironpython, Python for .net), và CORBA, Java... Python cũng được hỗ trợ bởi Internet Communications Engine (ICE) và nhiều công nghệ kết nối khác.

Có thể viết các thư viện trên C/C++ để nhúng vào Python và ngược lại.

*Python là ngôn ngữ có khả năng chạy trên nhiều nền tảng.*

Python có cho mọi hệ điều hành: Windows, Linux/Unix, OS/2, Mac, Amiga, và những hệ điều hành khác. Thậm chí có cả những phiên bản chạy trên .NET, máy ảo Java, và điện thoại di động (Nokia Series 60). Với cùng một mã nguồn sẽ chạy giống nhau trên mọi nền tảng.

*Python rất đơn giản và dễ học*

Python có cộng đồng lập trình rất lớn, hệ thống thư viện chuẩn, và cả các thư viện mã nguồn mở được chia sẻ trên mạng.

*Python là ngôn ngữ mã nguồn mở*

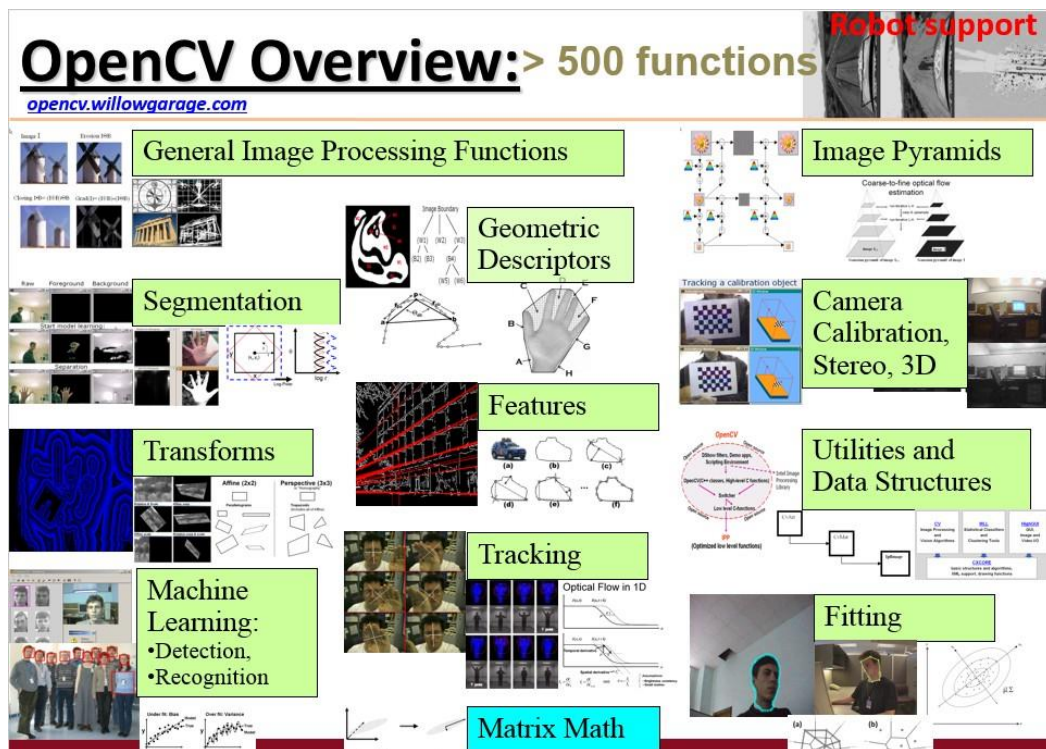
Cài đặt Python dùng giấy phép nguồn mở nên được sử dụng và phân phối tự do, ngay cả trong việc thương mại. Giấy phép Python được quản lý bởi Python Software Foundation.

## **2.9. Thư viện OpenCV và dlib**

Để cài đặt hệ thống, trong đề tài này chúng tôi lựa chọn ngôn ngữ C++ với bộ công cụ Visual Studio 2015 Update 2 và sử dụng hai thư viện mã nguồn mở là OpenCV và dlib.

### **2.9.1. Thư viện OpenCV**

OpenCV (<http://opencv.org>) là một thư viện mã nguồn mở cài đặt các thuật toán xử lý ảnh, thị giác máy tính và học máy. OpenCV được viết bằng ngôn ngữ C/C++ và có thể chạy trên hầu hết các nền tảng phần cứng (Intel, Mac, Raspberry Pi, Arduino...) và phần mềm (Windows, Mac OS, Linux, iOS, Android). Các ngôn ngữ và công cụ lập trình phổ biến như Python, Java, Ruby, Matlab, Objective C, cũng có thể dễ dàng sử dụng OpenCV trong việc viết và triển khai các ứng dụng thực tế.



Hình 2.26 Cấu trúc của thư viện OpenCV

Ban đầu thư viện OpenCV do hãng Intel phát triển và bảo trợ (từ những năm cuối của thế kỷ trước – bản alpha được phát hành chính thức vào tháng 2 năm 1999). Sau đó Intel quyết định chuyển OpenCV thành một thư viện mã nguồn mở và từ bỏ dự án này. Ngày nay OpenCV được phát triển và hỗ trợ bởi đông đảo các nhà phát triển, các nhà nghiên cứu trên toàn thế giới. Mỗi ngày lại có một bản vá lỗi với các thuật toán mới được thêm vào. Phiên bản mới nhất của OpenCV là 3.1 và công ty hiện đang bảo trợ cho OpenCV là một công ty của Nga có tên là Itseez.

Mục đích thiết kế của OpenCV là nhắm tới các ứng dụng thời gian thực đòi hỏi các thao tác tính toán phải hết sức hiệu quả về tốc độ thực hiện nên các kỹ thuật tối ưu hóa được sử dụng ở tất cả các mức độ: từ việc tận dụng các kiến trúc đa nhân cho tới việc sử dụng các tập lệnh của CPU mới nhất như SSE 2, AVX, AVX 512 ... OpenCV cũng tận dụng các thư viện lập trình song song mới nhất như TBB (Thread Building Blocks) của Intel, OpenMP và nền tảng CUDA của Nvidia để có thể triệt để khai thác các tài nguyên của hệ thống mà nó chạy trên.

Một mục tiêu quan trọng của OpenCV là cung cấp một hạ tầng các thuật toán thị giác máy tính có giao diện đơn giản và dễ dàng sử dụng để có thể trợ giúp cho các lập trình viên, các nhà nghiên cứu nhanh chóng xây dựng được (hay hiện thực hóa được các ý tưởng của họ) các ứng dụng thực tế. Thư viện OpenCV hiện đã có tới hơn 500 hàm khác nhau trải dài từ các hàm xử lý ảnh cơ bản, giao diện, phân đoạn ảnh, trích chọn đặc trưng, học máy cho tới các hàm hỗ trợ an toàn bảo mật và robotics.

Trên thực tế có khá nhiều các thư viện thị giác máy tính và xử lý ảnh (mã nguồn mở hoặc thương mại) khác nhưng việc lựa chọn OpenCV trong đề tài là do các nguyên nhân sau:

- + OpenCV có hiệu năng cao. Do được tối ưu ở nhiều mức và sử dụng các thư viện tốt nhất (TBB và IPP) của hãng Intel nên các ứng dụng viết bằng OpenCV cho tốc độ thực hiện rất nhanh và tận dụng được hết các tài nguyên phần cứng của máy tính (chạy song song).

- + OpenCV có nhiều tài liệu hỗ trợ và một cộng đồng các nhà phát triển sẵn sàng hỗ trợ các lỗi debug ([answers.opencv.org](http://answers.opencv.org)). So với các thư viện khác, hiện nay số lượng các sách và tài liệu tham khảo viết về OpenCV chiếm ưu thế vượt trội về số lượng và chất lượng.

- + Cú pháp linh hoạt và đơn giản, dễ sử dụng. Ở các phiên bản đầu tiên, muốn sử dụng OpenCV, các nhà phát triển phải viết các đoạn mã C dài dòng và khó nhớ. Nhưng hiện tại, cú pháp của OpenCV đã được cải tiến rất nhiều (do học tập từ Matlab). Điều này làm giảm đáng kể công sức của các lập trình viên khi không cần phải nhớ các cú pháp phức tạp như trước.

Về mặt cấu trúc, OpenCV là một tập hợp các thư viện (xem thêm ở hình 3.1), sau đây là danh sách các thư viện cốt lõi:

- + core: thư viện con chứa tất cả các kiểu dữ liệu cơ sở và các hàm thực hiện các thao tác cơ sở.

- + imgproc: thư viện xử lý ảnh chứa các thao tác xử lý ảnh cơ bản, chẳng hạn như các phép nhân chập và các bộ lọc không gian.

+ `highgui`: thư viện chứa các hàm hỗ trợ giao diện đồ họa chẳng hạn như việc hiển thị các ảnh hoặc nhận các dữ liệu input đơn giản.

+ `video`: thư viện chứa các hàm thực hiện việc đọc dữ liệu, ghi dữ liệu đối với các file video.

+ `features2d`: các thuật toán phát hiện, mô tả và đối sánh các điểm bất biến (keypoints).

+ `objdetect`: thư viện chứa các thuật toán để phát hiện các đối tượng đặc thù chẳng hạn như mặt người hoặc người đi bộ. Chúng ta cũng có thể sử dụng thư viện này để xây dựng các bộ huấn luyện để phát hiện các đối tượng khác trong các ứng dụng cụ thể.

+ `ml`: thư viện các thuật toán học máy được cài đặt để có thể làm việc một cách linh hoạt và hiệu quả với các kiểu dữ liệu của OpenCV.

+ `gpu`: chứa các cài đặt sử dụng nền tảng CUDA của Nvidia để tận dụng sức mạnh của các chip đồ họa GPU nhằm tăng tốc độ thực hiện cho các thuật toán đòi hỏi số thao tác xử lý lớn.

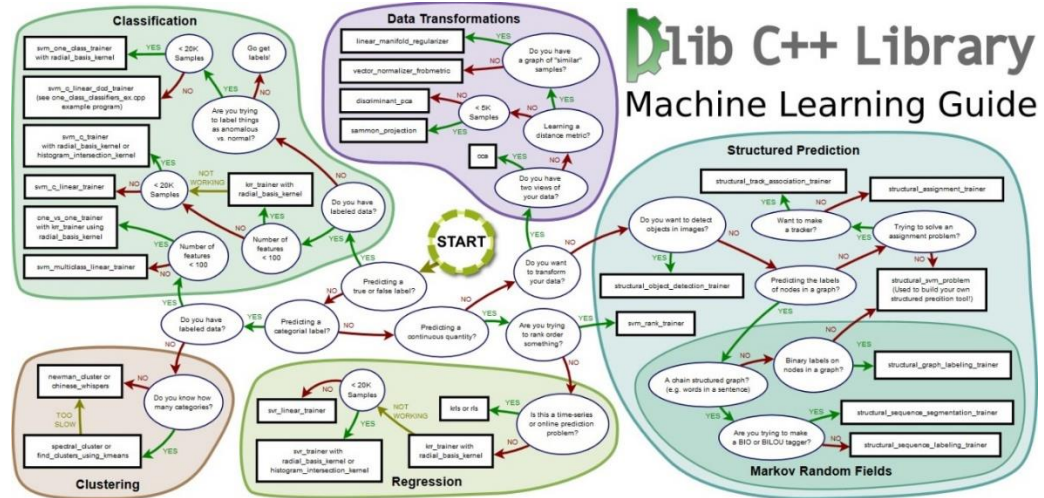
+ `nonfree`: chứa cài đặt của các thuật toán được bảo vệ bởi các bằng sáng chế hoặc có các điều kiện hạn chế trong các tình huống sử dụng cụ thể (miễn phí cho các ứng dụng giáo dục và nghiên cứu nhưng phải trả phí cho các ứng dụng mang tính thương mại, chẳng hạn như thuật toán SIFT hoặc SURF).

+ `contrib`: chứa các cài đặt của các thuật toán mới nhất, đang đưa xem xét để chính thức đưa vào OpenCV.

+ `ocl`: là một thư viện mới chứa cài đặt của các thuật toán sử dụng nền tảng OpenCL nhằm khai thác các kiến trúc không đồng nhất của hệ thống phần cứng trong các thao tác tính toán đòi hỏi số phép tính rất lớn.

Trong đề tài này, OpenCV được sử dụng cho các thao xử lý ảnh cơ bản được thực hiện trên các đối tượng ma trận điểm ảnh của lớp Mat (Matrix) như load thu mục ảnh, mở ảnh, load video, xem video, cắt hình ảnh, chuyển đổi ảnh và tham gia vào một số chức năng trong quá trình đào tạo mô hình nhận dạng.

## 2.9.2. Thư viện dlib



Hình 2.27 Cấu trúc của thư viện dlib.

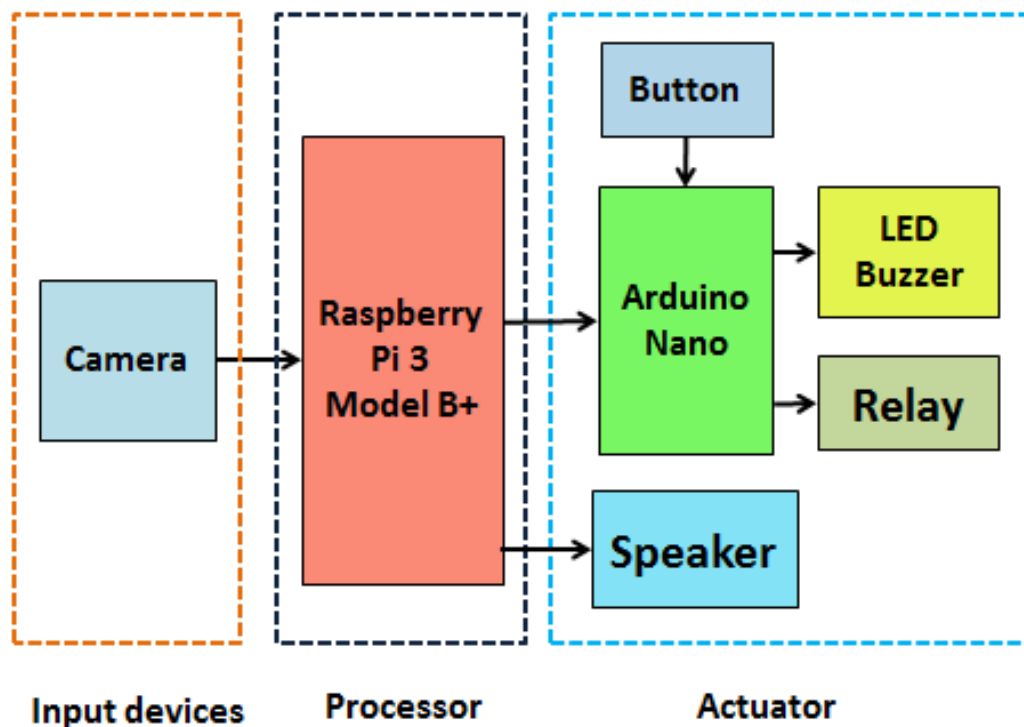
Ngoài thư viện OpenCV, chúng tôi còn sử dụng dlib (<http://dlib.net>), một thư viện mã nguồn mở khác cho việc cài đặt hệ thống. Khác với mục đích của OpenCV là cung cấp một hạ tầng thuật toán cho các ứng dụng xử lý ảnh và thị giác máy tính, dlib được thiết kế cho các ứng dụng học máy và trí tuệ nhân tạo với các thư viện con chính sau:

- + classification: các kỹ thuật phân lớp chủ yếu dựa trên hai phương pháp cơ sở là kNN và SVM.
- + data transformation: các thuật toán biến đổi dữ liệu nhằm giảm số chiều, loại bỏ các dữ liệu dư thừa và tăng cường tính khác biệt (discriminant) của các đặc điểm được giữ lại.
- + clustering: các kỹ thuật phân cụm.
- + regression: các kỹ thuật hồi qui
- + structured prediction: các thuật toán dự đoán có cấu trúc
- + Markov Random Fields: các thuật toán dựa trên các trường Markov ngẫu nhiên.

Cụ thể trong đề tài này, phần phát hiện và căn chỉnh ảnh mặt người một cách tự động sẽ được cài đặt bằng cách sử dụng thư viện dlib với dữ liệu huấn luyện đã được cung cấp trước như HOG bằng SVM, Facial Landmark 68...

## Chương 3: GIẢI THUẬT

### 3.1. Sơ đồ khối hệ thống.



Hình 3.1 Sơ đồ khối hệ thống

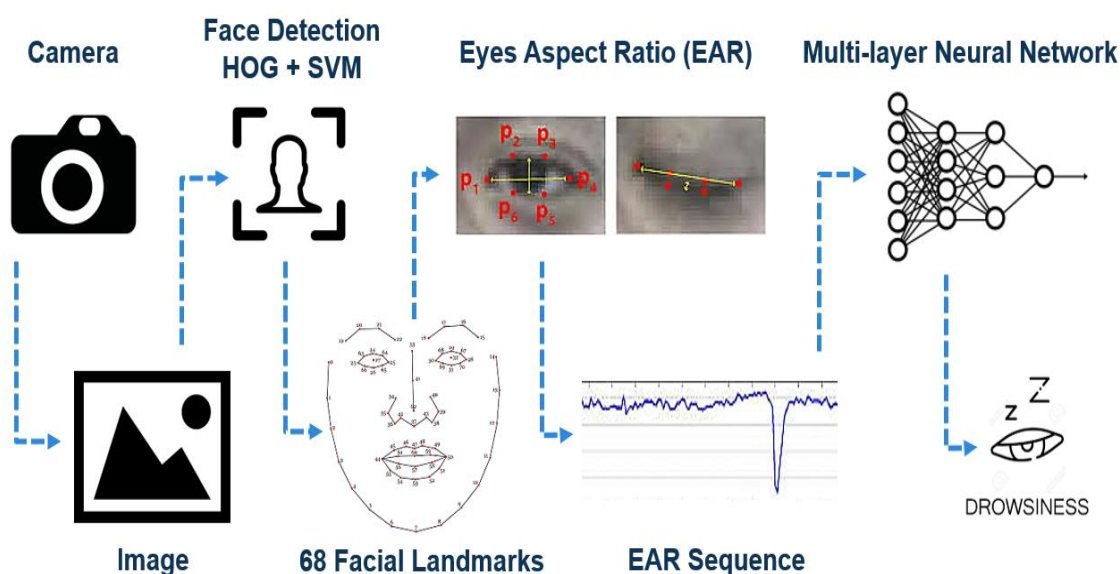
Theo như hình 3.1, thiết bị cảnh báo buồn ngủ gật được thiết kế với các board mạch và linh kiện điện tử sau: Raspberry Pi 3 Model B+, Arduino Nano, Camera Logitech C170, Mạch Relay SSR, LED, Button.

Chương trình Raspberry Pi 3 có nhiệm vụ thu thập tín hiệu hình ảnh từ Camera Logitech C170, sau đó tiến hành phát hiện và trích vùng khuôn mặt, phân tích 68 điểm đặc trưng trên khuôn mặt và trích xuất vùng mắt, tính toán độ mở mắt, lưu lại thông số EAR theo thời gian và kết luận trạng thái buồn ngủ bằng model đã được huấn luyện.

Nếu tài xế có dấu hiệu buồn ngủ, chương trình sẽ phát ra tín hiệu âm thanh qua loa, truyền dữ liệu cho board Arduino Nano thông qua cổng Serial.

Khi nhận được tín hiệu Serial, chương trình trên Arduino Nano điều khiển bật đèn cảnh báo, còi báo hiệu và điều khiển 2 Relay, ta có thể dùng Relay này cho động cơ rung ghế hoặc máy tạo độ ẩm cho tài xế.

### 3.2. Sơ đồ tiến trình hệ thống phát hiện dấu hiệu buồn ngủ.



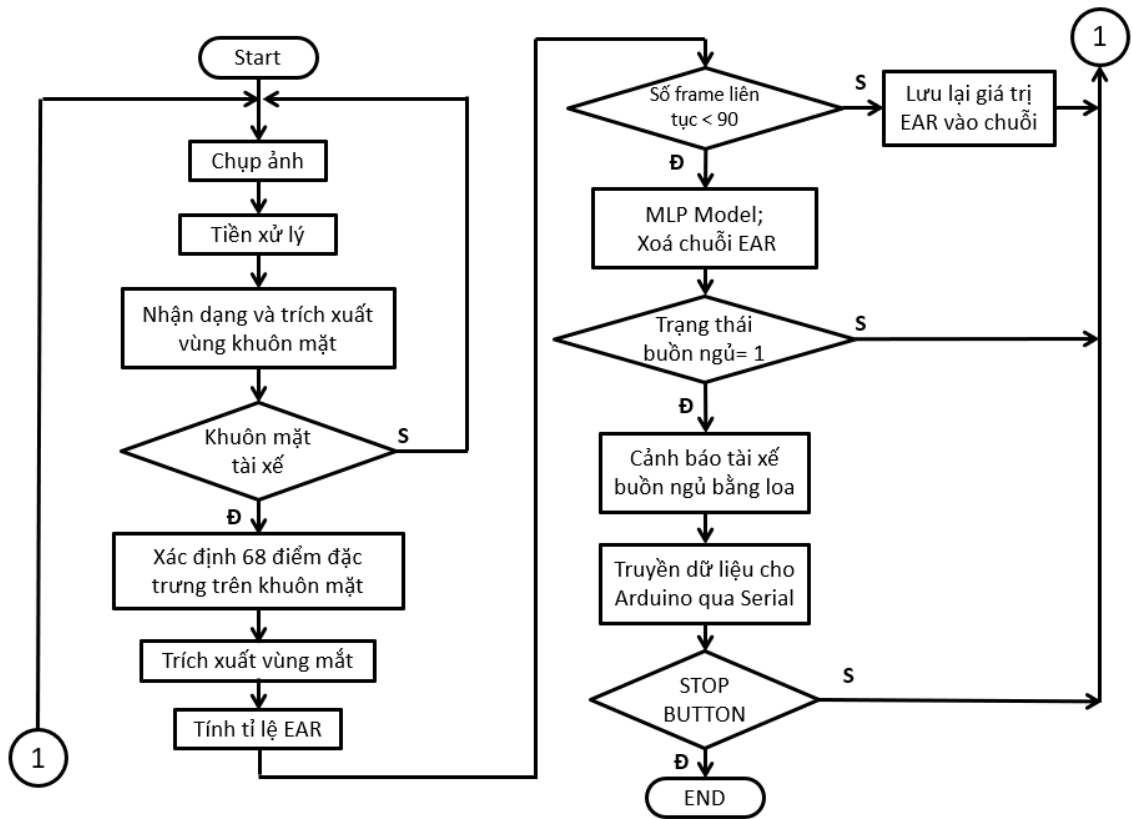
Hình 3.2 Sơ đồ tiến trình hệ thống

Theo như hình 3.2, tiến trình của hệ thống diễn ra theo các bước sau:

- Hình ảnh sau khi thu thập từ camera sẽ thông qua bộ nhận diện mặt người sử dụng đặc trưng HOG và SVM.
- Nếu xác định khuôn mặt có trong bức ảnh, vùng mặt sẽ được cắt ra và tiến hành phân tích 68 điểm đặc trưng trên khuôn mặt.
- Sau khi thu được các điểm đặc trưng, ta xác định được tọa độ của hai mắt trái / phải.
- Từ đó xác định được tỉ lệ nhắm mở mắt (EAR).
- Thu lại thông số EAR liên tục theo thời gian cho đến khi đủ số frame (khung hình) đặt trước, ta được dãy thông số EAR theo thời gian.
- Dãy thông số EAR này sẽ là giá trị đầu vào cho mạng nơron truyền thẳng nhiều lớp, giá trị đầu ra của mạng nơron này là trạng thái buồn ngủ của tài xế.
- Nếu tài xế có dấu hiệu buồn ngủ thì hệ thống sẽ cảnh báo.
- Nếu tài xế vẫn tỉnh táo thì dãy EAR được thu thập lại và tiến hành chu trình mới.



### 3.3. Lưu đồ chương trình phát hiện dấu hiệu buồn ngủ.



Hình 3.3 Lưu đồ giải thuật cho chương trình Raspberry

- Khi khởi động hệ thống chương trình Raspberry sẽ chụp ảnh từ Camera C170 rồi tiến hành tiền xử lý:

+ Chuyển từ ảnh màu RGB sang ảnh xám.

+ Giảm kích thước ảnh xuống độ phân giải 320x200 pixel để tốc độ tính toán nhanh hơn, phù hợp với việc chạy thời gian thực trên board mạch Raspberry Pi 3 Model B+.

+ Cân bằng histogram theo vùng 4x4 cho toàn bộ bức ảnh.

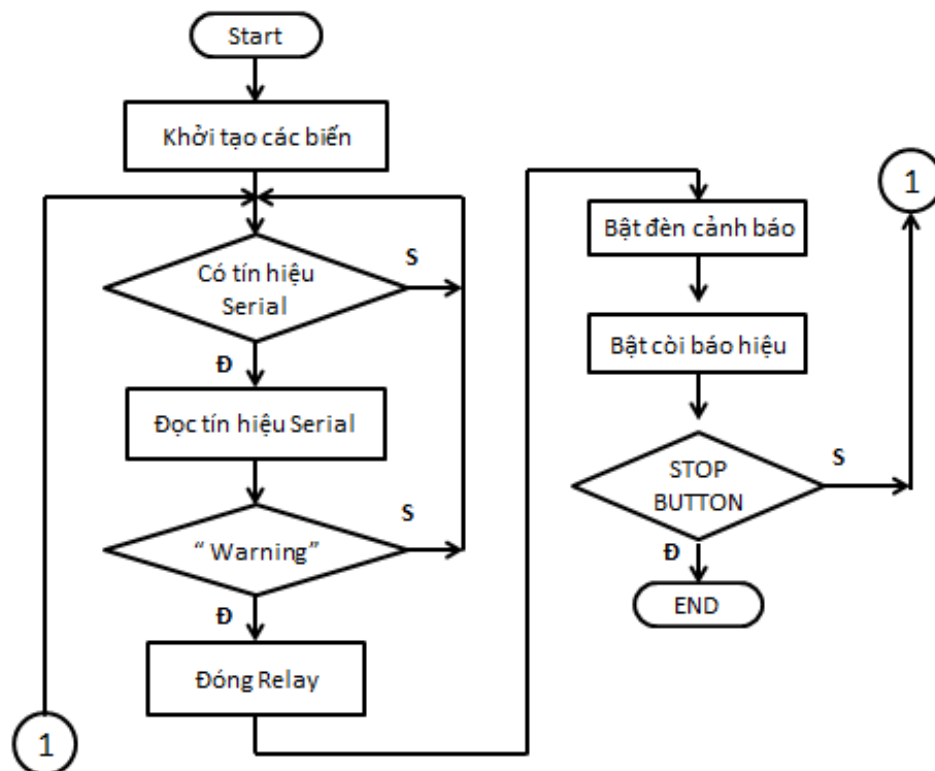
- Sau đó chương trình sẽ nhận dạng và trích vùng khuôn mặt dùng bộ nhận dạng được huấn luyện bằng đặc trưng HOG.

- Nếu phát hiện được khuôn mặt tài xế, chương trình sẽ xác định 68 điểm đặc trưng trên khuôn mặt. Từ đó trích xuất được 2 vùng mắt và tính độ nhắm mở mắt EAR rồi so sánh, nếu số frame (khung hình) liên tục < 90 thì lưu lại EAR vào chuỗi rồi quay lại chụp ảnh, cộng thêm 1 frame. Nếu số Frame > = 90 tiến hành dự đoán

trạng thái buồn ngủ của tài xế, nếu tài xế có dấu hiệu buồn ngủ thì cảnh báo đang buồn ngủ qua loa ngoài gắn vào Raspberry. Đồng thời là truyền dữ liệu qua Arduino Nano thông qua cổng giao tiếp Serial với tốc độ truyền là 115200 bit/s.

Chương trình được lập đi lập lại cho đến khi nhấn nút tắt hệ thống.

### 3.4. Lưu đồ chương trình trên Arduino.



Hình 3.4 Lưu đồ giải thuật cho chương trình Arduino

Khi cấp nguồn, chương trình trên Arduino Nano sẽ khởi tạo các biến.

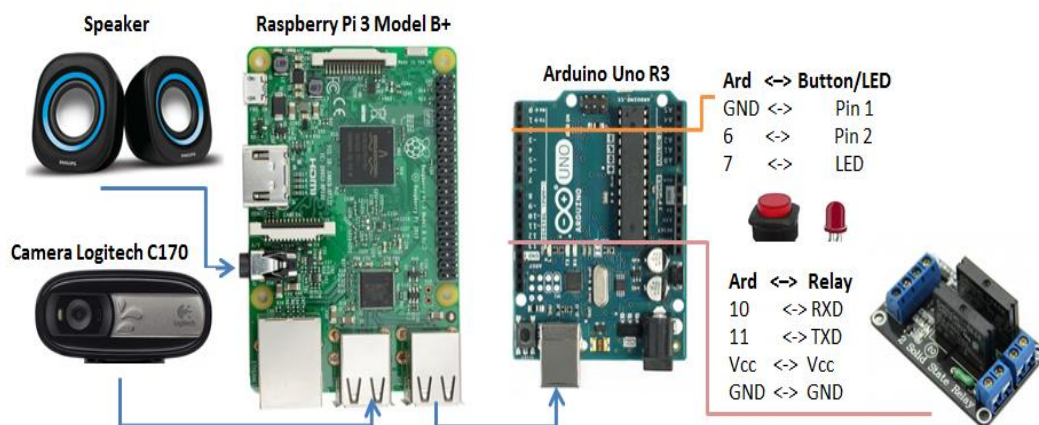
Nếu có tín hiệu truyền đến cổng Serial thì Arduino sẽ đọc và giải mã tín hiệu.

Nếu chuỗi đọc được bắt đầu bằng chuỗi “warning” thì sẽ đóng Relay , bật đèn cảnh báo và còi báo hiệu. Kết thúc chương trình này, Arduino sẽ quay lại và chờ nhận tín hiệu Serial.

## Chương 4: KẾT QUẢ THỰC NGHIỆM

### 4.1. Xây dựng phần cứng.

#### 4.1.1. Sơ đồ nối dây phần cứng



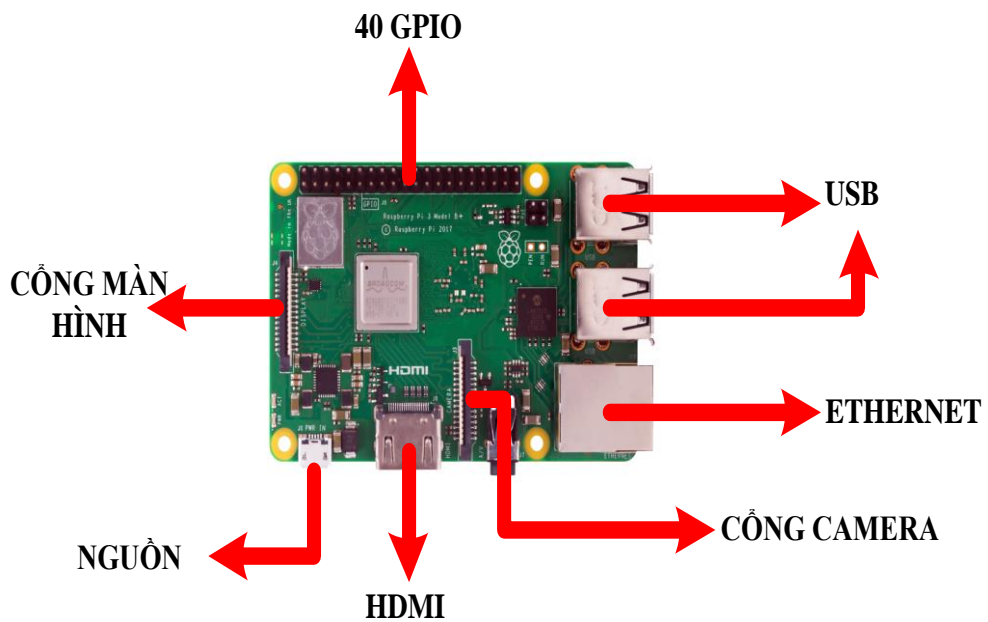
Hình 4.1 Sơ đồ nối dây phần cứng

Bảng 4.1: Nối dây phần cứng.

<b>Raspberry Pi 3</b>	
Audio (Jack 3.5)	Loa
USB	Camera Logitech C170
Serial Port USB	COM Arduino
<b>Arduino Uno R3</b>	
GND	Button (Pin 1)
6	Button (Pin 2)
7	LED
<b>SSR Relay</b>	
10	CH1
11	CH2
VCC	VCC
GND	GND

#### 4.1.2. Board xử lý trung tâm.

Raspberry Pi (H.4.4) [16] là chiếc máy tính kích thước nhỏ được tích hợp nhiều phần cứng mạnh mẽ đủ khả năng chạy hệ điều hành và cài đặt được nhiều ứng dụng trên nó, ngoài việc sử dụng như một máy tính bình thường chạy hệ điều hành Linux hoặc Windows 10 IoT, máy còn có khả năng xuất tín hiệu ra 40 chân GPIO giúp bạn có thể giao tiếp và điều khiển vô số các board mạch và ngoại vi bên ngoài để thực hiện vô số các ứng dụng khác nhau. Với giá chỉ vài chục USD, Raspberry hiện đang là mini computer nổi bật nhất hiện nay.



Hình 4.2 Board Raspberry Pi 3 Model B+

Ban đầu, tổ chức Raspberry Pi Foundation phát triển dự án Raspberry với mục tiêu chính là giảng dạy máy tính cho trẻ em và tạo ra một công cụ giá rẻ (chỉ vài chục USD) để sinh viên nghiên cứu học tập. Tuy nhiên, sau khi xuất hiện, Raspberry Pi được cộng đồng đánh giá cao về tính ứng dụng với phần cứng được hỗ trợ tốt, Pi đã nhanh chóng phát triển một cách rộng rãi. Pi phù hợp cho những ứng dụng cần khả năng xử lý mạnh mẽ, đa nhiệm hoặc giải trí và đặc biệt cần chi phí thấp. Hiện nay đã có hàng ngàn ứng dụng đa dạng được cài đặt trên Raspberry Pi.

Tổ chức Raspberry Pi Foundation được thành lập năm 2009, và cùng với sự giúp đỡ của 3 hãng cung cấp linh kiện điện tử lớn là EGOMAN, QSIDA, SONY phát triển lên Raspberry Pi. Trên thế giới, bạn có thể tìm mua tại những nhà phân phối của hãng như EGOMAN, ELEMENT14 hoặc RS Component.

Phiên bản Raspberry Pi đầu tiên được phát hành tháng 2 năm 2012, và tới nay đã có nhiều phiên bản khác nhau, với sự nâng cấp của phần cứng, cũng như hướng tới những mục tiêu khác nhau. Phiên bản theo thứ tự ra mắt là: Pi A → Pi A+ → Pi 1 B → Pi 1B+ → Pi 2B → Pi Zero → Pi 3B → Pi 3B+

Bảng 4.2: Thông số kỹ thuật chính của Raspberry Pi 3 Model B+

Raspberry	Pi 3 Model B+
Vi xử lý	Broadcom BCM2837, ARMv8 (64bit) quad-core
GPU	Broadcom VideoCore IV, OpenGL ES 2.0, OpenVG 1080p60 , 400 MHz
Tốc độ xử lý	1.4 Ghz
Power Ratings	800mA, 5V
RAM (chia sẻ với GPU)	1GB LPDDR2 (900Mhz)
Bộ nhớ	Micro SD
GPIO	40
Kết nối	4xUSB 2.0 10/100mb Ethernet wifi 802.11 n Bluetooth 4.1 CSI, DSI
Video & audio	1080p HDMI, stereo audio 3.5mm jack
Kích thước	85x56mm

### Cài hệ điều hành cho Raspberry:

Raspberry hoạt động như một máy tính thu nhỏ, do vậy cần cài đặt hệ điều hành và các thư viện cần thiết để Raspberry đáp ứng yêu cầu của hệ thống.

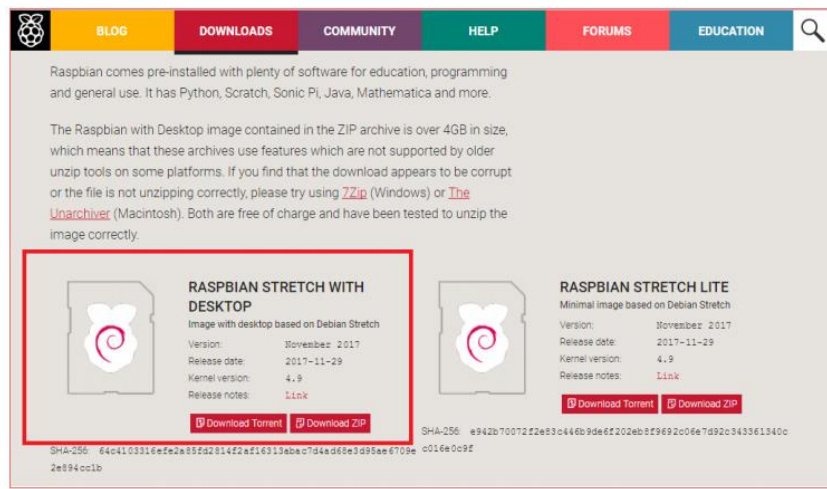
Thiết bị cần thiết để cài đặt:

- Thẻ SD dung lượng 32GB
- Máy tính cá nhân

- Cáp kết nối HDMI
- Bàn phím
- Màn hình kết nối HDMI

Các bước cài đặt hệ điều hành cho Raspberry Pi:

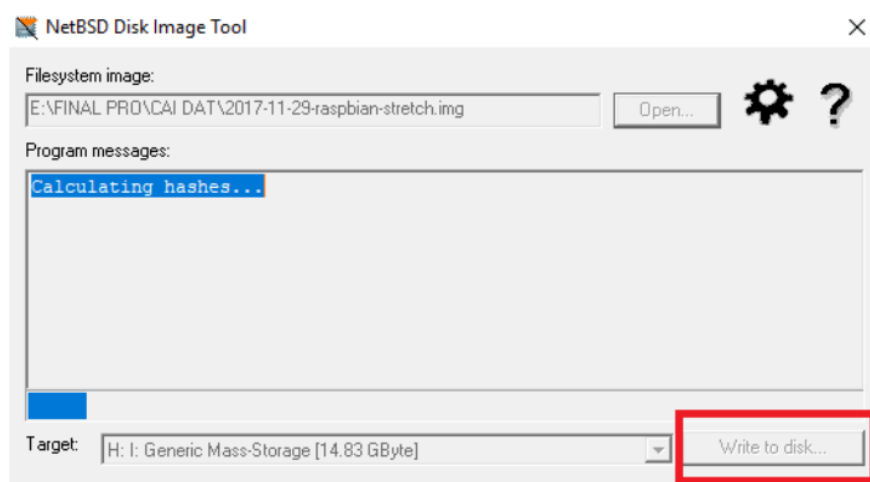
- Tải hệ điều hành trên máy tính cá nhân trên đường dẫn:  
<https://www.raspberrypi.org/downloads/raspbian/>



Hình 4.3 Hệ điều hành Raspbian trên trang chủ Raspberry Pi

Giải nén và cài hệ điều hành vào thẻ SD: Tải phần mềm Win32 Disk

Imager và cài đặt. Khi cài đặt xong. Chúng ta mở phần mềm, chọn đường dẫn đến file chứa hệ điều hành Raspbian Stretch và chọn Write. Chờ báo hoàn thành là đã xong bước cài hệ điều hành vào thẻ nhớ.



Hình 4.4 Phần mềm Win32 Disk Imager

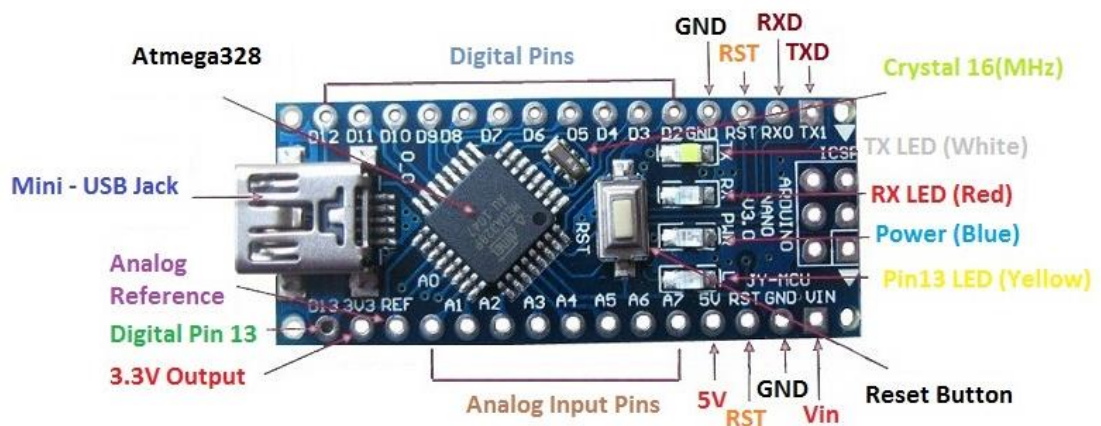
- Cắm thẻ nhớ vào khe thẻ nhớ của Raspberry.
- Cắm nguồn và kết nối HDMI Raspberry với màn hình.

Cài đặt các thư viện cần thiết cho Raspberry: Đối với đề tài, có một số thư viện cần thiết cho việc xử lý hình ảnh cũng như tính toán. 2 thư viện quan trọng sau:

- OpenCV: gõ lệnh `sudo apt-get install libopencv-dev python3-opencv` trên terminal của Raspberry Pi.
- Dlib: gõ lệnh `pip install dlib` trên terminal của Raspberry Pi.

#### 4.1.3. Board điều khiển – Board Arduino Nano.

Là một board mạch vi xử lý, phần cứng bao gồm một board mạch nguồn mở được thiết kế trên nền tảng vi xử lý AVR Atmel 8bit, hoặc ARM Atmel 32-bit. Những Model hiện tại được trang bị gồm 1 cổng giao tiếp USB, 6 chân đầu vào analog, 14 chân I/O kỹ thuật số tương thích với nhiều board mở rộng khác nhau. Cho phép người dùng viết các chương trình cho Aduino bằng ngôn ngữ C hoặc C++.



Hình 4.5 Sơ đồ Arduino Nano

Một mạch Arduino bao gồm một vi điều khiển AVR với nhiều linh kiện bổ sung giúp dễ dàng lập trình và có thể mở rộng với các mạch khác. Một khía cạnh quan trọng của Arduino là các kết nối tiêu chuẩn của nó, cho phép người dùng kết nối với CPU của board với các module thêm vào có thể dễ dàng chuyển đổi, được gọi là shield. Vài shield truyền thông với board Arduino trực tiếp thông qua các chân khác nhau, nhưng nhiều shield được định địa chỉ thông qua serial bus I<sup>2</sup>C-

nhều shield có thể được xếp chồng và sử dụng dưới dạng song song. Arduino chính thức thường sử dụng các dòng chip megaAVR, đặc biệt là ATmega8, ATmega168, ATmega328, ATmega1280, và ATmega2560. Một vài các bộ vi xử lý khác cũng được sử dụng bởi các mạch Aquino tương thích.

Hầu hết các mạch gồm một bộ điều chỉnh tuyến tính 5V và một thạch anh dao động 16 MHz (hoặc bộ cộng hưởng ceramic trong một vài biến thể); mặc dù một vài thiết kế như LilyPad chạy tại 8 MHz và bỏ qua bộ điều chỉnh điện áp onboard do hạn chế về kích cỡ thiết bị. Một vi điều khiển Arduino cũng có thể được lập trình sẵn với một boot loader cho phép đơn giản là upload chương trình vào bộ nhớ flash on-chip, so với các thiết bị khác thường phải cần một bộ nạp bên ngoài. Điều này giúp cho việc sử dụng Arduino được trực tiếp hơn bằng cách cho phép sử dụng 1 máy tính gốc như là một bộ nạp chương trình.

Theo nguyên tắc, khi sử dụng ngấn xếp phần mềm Arduino, tất cả các board được lập trình thông qua một kết nối RS-232, nhưng cách thức thực hiện lại tùy thuộc vào đời phần cứng. Các board Serial Arduino có chứa một mạch chuyển đổi giữa RS232 sang TTL. Các board Arduino hiện tại được lập trình thông qua cổng USB, thực hiện thông qua chip chuyển đổi USB-to-serial như là FTDI FT232. Vài biến thể, như Arduino Mini và Boarduino không chính thức, sử dụng một board adapter hoặc cáp nối USB-to-serial có thể tháo rời được, Bluetooth hoặc các phương thức khác. Board Arduino sẽ đưa ra hầu hết các chân I/O của vi điều khiển để sử dụng cho những mạch ngoài. Diecimila, Duemilanove, và bây giờ là Uno đưa ra 14 chân I/O kỹ thuật số, 6 trong số đó có thể tạo xung PWM (điều chế độ rộng xung) và 6 chân input analog, có thể được sử dụng như là 6 chân I/O số. Những chân này được thiết kế nằm phía trên mặt board, thông qua các header cái 0.10-inch (2.5 mm). Nhiều shield ứng dụng plug-in cũng được thương mại hóa. Các board Arduino Nano, và Arduino-compatible Bare Bones Board và Boarduino có thể cung cấp các chân header đực ở mặt trên của board dùng để cắm vào các breadboard.



#### 4.1.4. Camera Logitech C170.



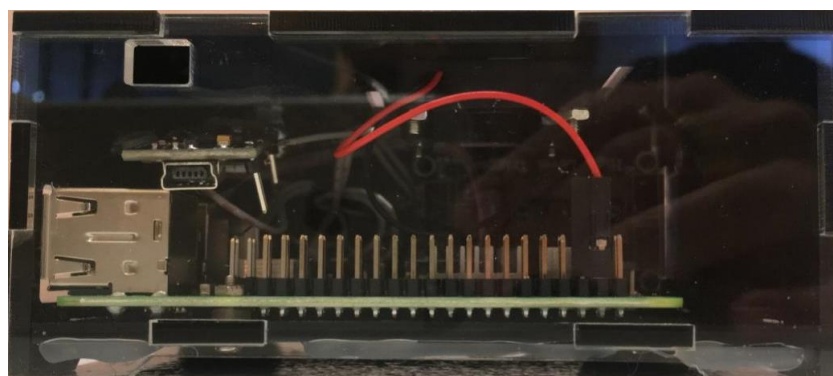
Hình 4.6 Camera Logitech C170

Bảng 4.3: Thông số kỹ thuật của Camera Logitech C170

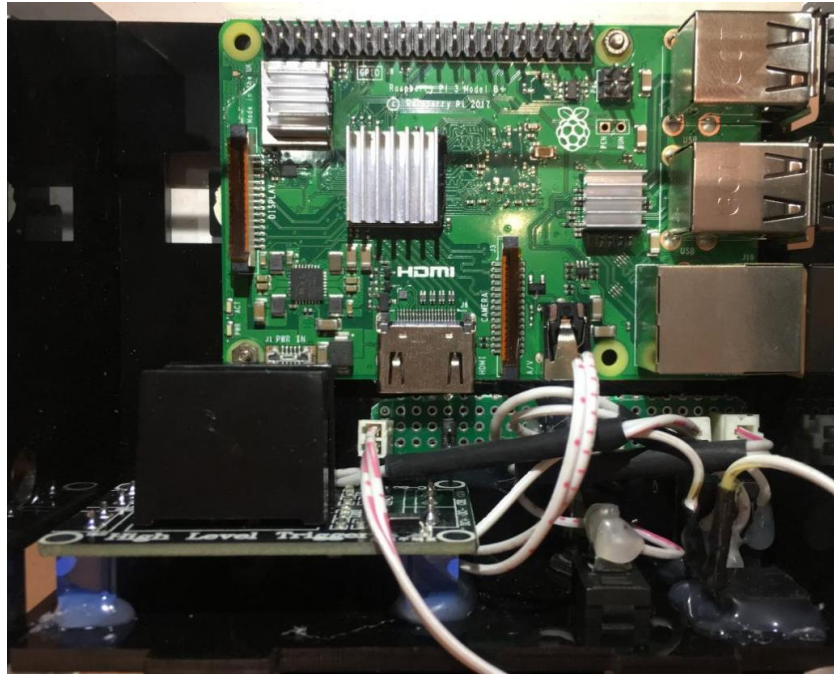
Độ phân giải tối đa	720p/30fps
Loại tiêu cự	Lấy nét cố định
Công nghệ thấu kính	Tiêu chuẩn
Micrô tích hợp	Đơn âm
Trường ngắm	58°
Độ dài dây cáp	0,95 m

#### 4.2. Kết quả thực nghiệm.

##### 4.2.1. Mô hình phần cứng thiết bị.



Hình 4.7 Mặt trước của thiết bị cảnh báo buồn ngủ



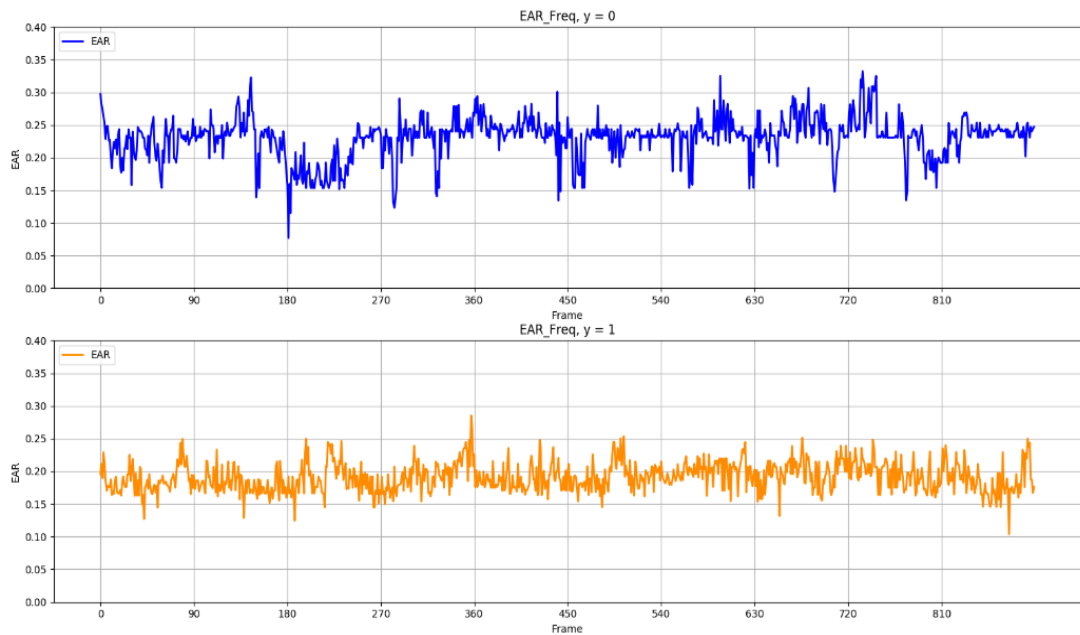
Hình 4.8 Thành phần bên trong của thiết bị cảnh báo buồn ngủ

Hình 4.8 là mô hình “Thiết bị phát hiện cảnh báo buồn ngủ cho tài xế” mà tác giả đã thực hiện được. Như trên hình, các phần cứng mà nhóm đã sử dụng bao gồm : Raspberry Pi 3 Model B+, Arduino Uno R3, Camera Logitech C170, Module SSR Relay, LED, Button. Với thiết kế như trên, ta có một thiết bị có thể gắn và thực nghiệm trực tiếp trên xe ô tô với kích thước nhỏ gọn chỉ 10cm x 8cm x 4.5cm ( Dài x Rộng x Cao) và dùng nguồn 5V 3A cấp từ bộ sạc xe hơi.

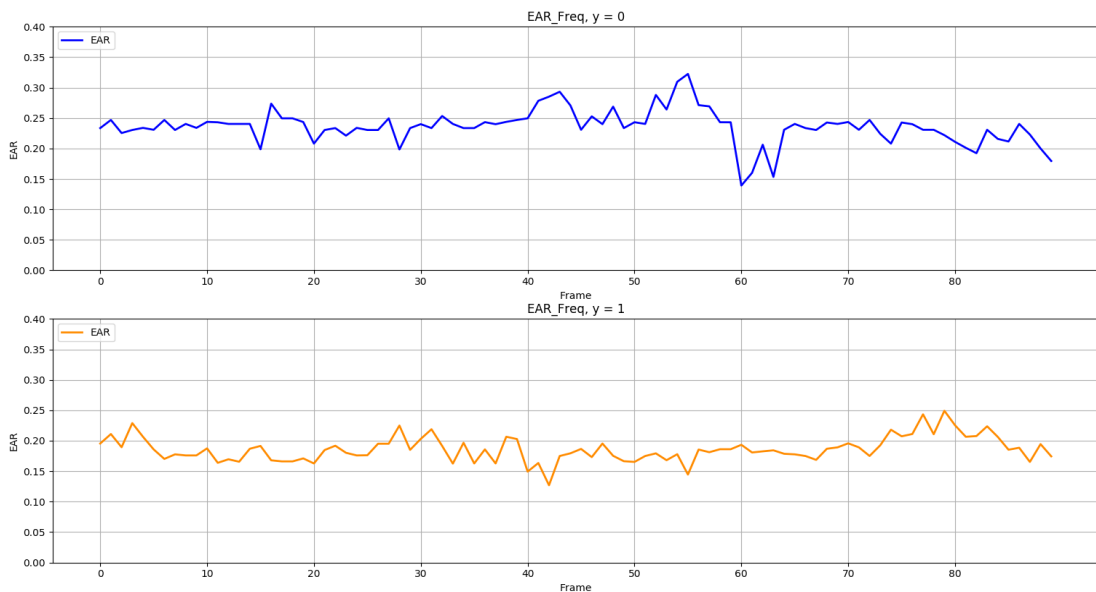
#### 4.2.2. Thu thập dữ liệu huấn luyện

Dữ liệu huấn luyện mạng nơron phát hiện dấu hiệu buồn ngủ được thu từ 25 người trong điều kiện ánh sáng ban ngày, bao gồm dãy EAR liên tục trong 90 khung hình lúc người đó tỉnh táo và lúc buồn ngủ kèm theo nhãn (label) cho từng trường hợp.

Trung bình mỗi người thu được 1000 mẫu, tổng cộng ta có 25380 mẫu không có dấu hiệu buồn ngủ và 25020 mẫu có dấu hiệu buồn ngủ. Có thể thấy, bộ dữ liệu này đã cân bằng, đủ tin cậy để huấn luyện mạng nơron học sâu. Phân bố của bộ dữ liệu được thể hiện trong hình 4.11.



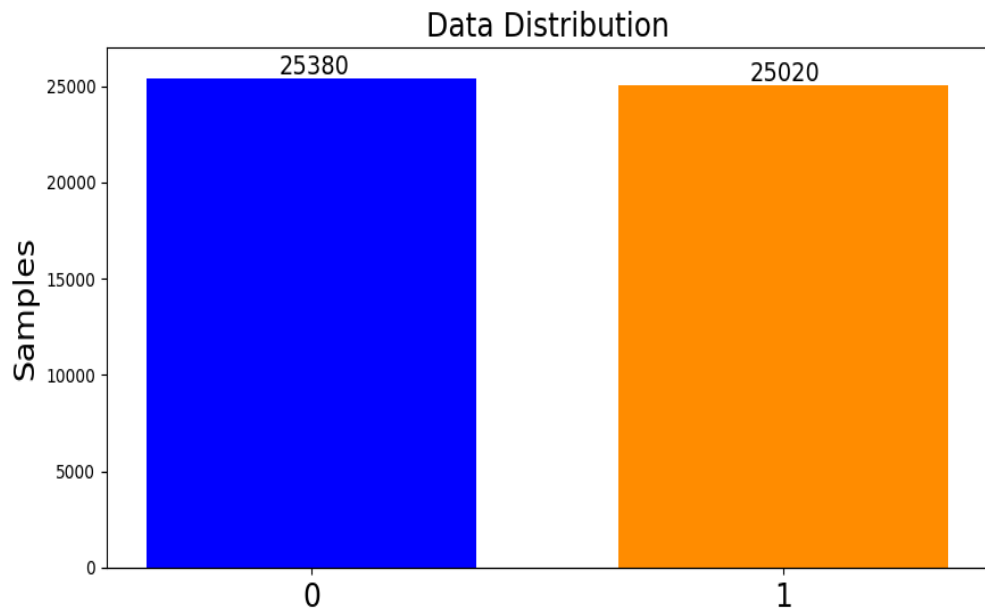
Hình 4.9 Dãy EAR liên tục khi thu dữ liệu



Hình 4.10 Dãy EAR trong 90 khung hình liên tục

Từ hình 4.10, ta thấy rằng:

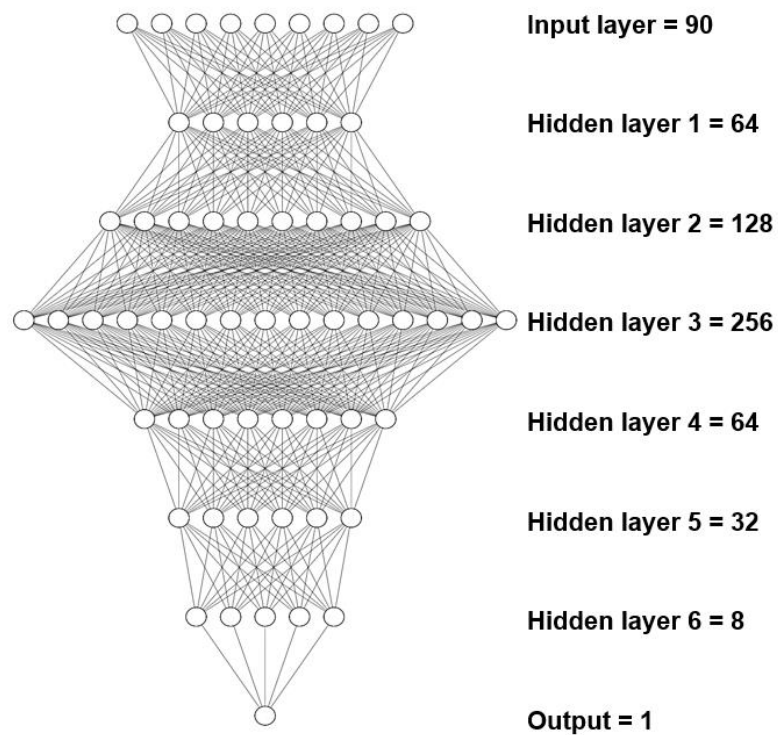
- Dãy EAR khi tài xế tỉnh táo (trên), giá trị của EAR dao động quanh 0.25 và giảm xuống thấp khi tài xế chớp mắt.
- Dãy EAR khi tài xế có dấu hiệu buồn ngủ (dưới), giá trị của EAR dao động quanh 0.2 và không thay đổi nhiều, chứng tỏ mắt có dấu hiệu lơ đãng.



Hình 4.11 Phân bố dữ liệu huấn luyện

Dữ liệu sau khi thu thập được chia ra làm 80% (40320 mẫu) để train và 20% (10080 mẫu) để test mạng nơron.

#### 4.2.3. Cấu trúc mạng và tối ưu các thông số



Hình 4.12 Cấu trúc mạng nơron phát hiện buồn ngủ

Hình 4.12 thể hiện cấu trúc mạng nơ-ron phát hiện dấu hiệu buồn ngủ gồm có:

- 1 lớp đầu vào bao gồm 90 nơ-ron
- 6 lớp ẩn với các kích thước lần lượt là 64, 128, 256, 64, 32, 8
- 2 lớp ẩn cuối cùng được dropout với tỉ lệ 0.2 nhằm đảm bảo cho mạng nơ-ron học được hiệu quả hơn, tránh overfitting trong quá trình huấn luyện.
- 1 lớp đầu ra mang giá trị trạng thái buồn ngủ là 0 và 1.

Layer (type)	Output Shape	Param #
dense_1 (Dense)	(None, 64)	5824
dense_2 (Dense)	(None, 128)	8320
dense_3 (Dense)	(None, 256)	33024
dense_4 (Dense)	(None, 64)	16448
dense_5 (Dense)	(None, 32)	2080
dropout_1 (Dropout)	(None, 32)	0
dense_6 (Dense)	(None, 8)	264
dropout_2 (Dropout)	(None, 8)	0
dense_7 (Dense)	(None, 1)	9
Total params: 65,969		
Trainable params: 65,969		
Non-trainable params: 0		

Hình 4.13 Thông số của mạng nơ-ron phát hiện buồn ngủ

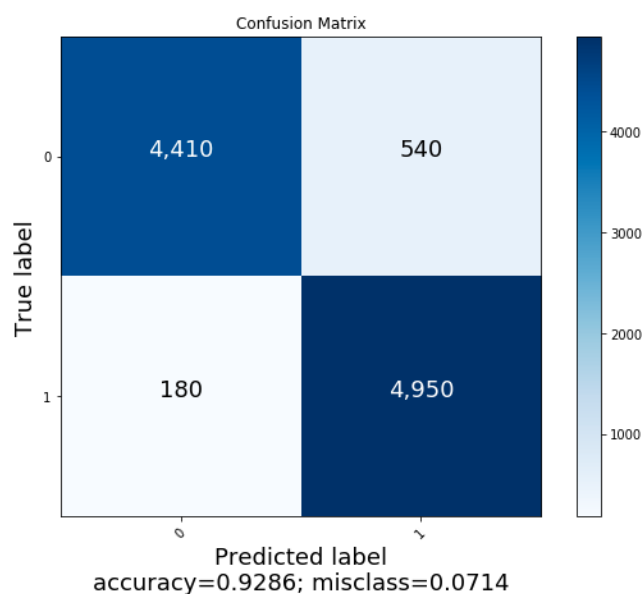
Mạng nơ-ron có 65,969 parameter được huấn luyện trên máy tính CPU Core i7, GPU NVIDIA GeForce GTX 1080Ti.

#### 4.2.4. Kết quả huấn luyện mạng.

Sau 4.5 giờ huấn luyện qua hơn 5000 epochs, training accuracy đạt được 98,4%, testing accuracy đạt được 92,8%.

Kết quả dự đoán trạng thái buồn ngủ trên 10080 mẫu testing data được trình bày trong Confusion Matrix hình 4.14.

Mạng sau khi được huấn luyện trên máy tính sẽ được nhúng sang thiết bị phát hiện buồn ngủ, cụ thể là nhúng vào máy tính Raspberry Pi 3 Model B+ để chạy thực nghiệm trong thời gian thực.



Hình 4.14 Confusion Matrix

#### 4.2.5. Kết quả thực nghiệm trên hệ thống.

Thiết lập hệ thống: Cố định camera, thay đổi khoảng cách từ camera đến ghế ngồi tài xế, thay đổi góc quét của camera.

Mô tả cách thức hoạt động của hệ thống:

+ Khi khởi động hệ thống, camera sẽ tiến hành ghi hình, nếu nhận ra có khuôn mặt tài xế trong 15 frames liên tục thì phát ra đoạn âm thanh: **“Welcom to the driver drowsiness detector”** và kích hoạt chức năng phát hiện và cảnh báo buồn ngủ cho tài xế.

+ Nếu tài xế có dấu hiệu buồn ngủ thì:

- Đèn cảnh báo hoạt động: đèn led màu đỏ sẽ sáng.
- Cảnh báo bằng giọng nói: **“Warning! The driver is sleepy!”**
- Còi báo hiệu sẽ báo để đánh thức tài xế.

+ Trong quá trình lái xe, nếu tài xế không nhìn về phía trước trong 15 frame liên tục thì hệ thống sẽ phát âm thanh: **“Please look ahead!”** nhắc nhở tài xế nhìn về phía trước khi lái xe.

+ Khi tài xế rời khỏi vị trí lái và hệ thống phát hiện không có khuôn mặt tài xế trong 150 frame liên tục thì sẽ tắt chức năng cảnh báo buồn ngủ và chờ phiên làm việc tiếp theo.



#### 4.2.5.1. Khi tài xế ở trạng thái tĩnh táo

Khi tài xế ngồi gần, khoảng cách từ tài xế đến camera là 60cm.



a) Nhìn trực diện



b) Nhìn lên trên



c) Nhìn xuống dưới



e) Nhìn sang trái 55°



d) Nhìn sang phải 55°

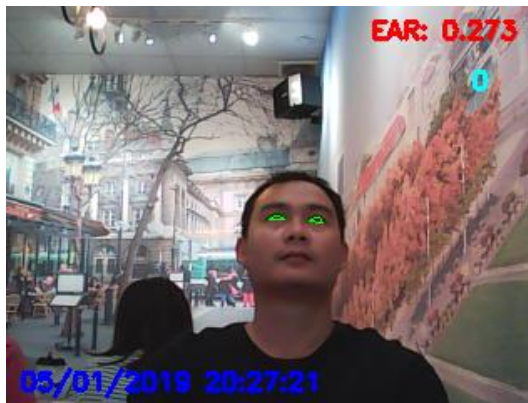
Hình 4.15: Thực nghiệm khi tài xế ngồi gần trong trạng thái tĩnh táo

Ở khoảng cách 60 cm, thay đổi góc quét camera có giá trị từ:  $-55^{\circ}$  đến  $55^{\circ}$  thì tỉ lệ mở mắt từ 0,234 đến 0,437.

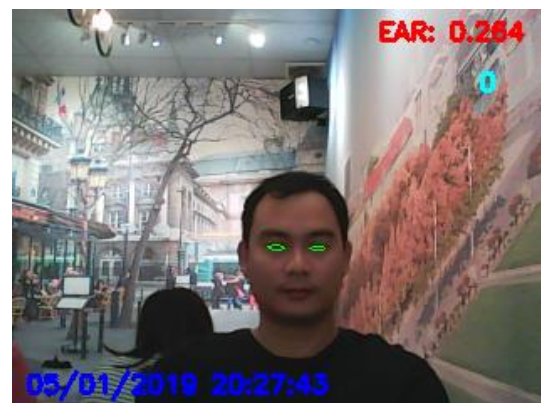
*Khi tài xế ngồi xa, khoảng cách giữa tài xế đến camera là 100cm.*



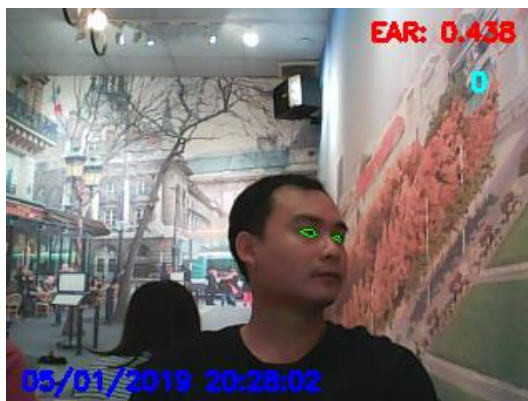
a) Nhìn trực diện



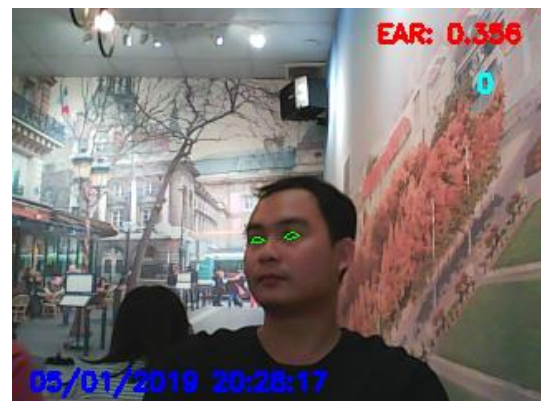
b) Nhìn lên trên



c) Nhìn xuống dưới



d) Nhìn sang trái 55°



e) Nhìn sang phải 55°

Hình 4.16: Thực nghiệm khi tài xế ngồi xa trong trạng thái tỉnh táo

Ở khoảng cách 100 cm, thay đổi góc quét camera có giá trị từ:  $-55^{\circ}$  đến  $55^{\circ}$  thì tỉ lệ mở mắt từ 0,264 đến 0,437. Như vậy, khi thay đổi khoảng cách từ camera đến vị trí ghế ngồi của tài xế thì sự nhận dạng và phân tích tỉ lệ mở mắt thay đổi không đáng kể.



#### 4.2.5.2. Khi tài xế có dấu hiệu buồn ngủ

Khi tài xế ngồi gần, khoảng cách từ tài xế đến camera là 60cm.



a) Nhìn trực diện



b) Nhìn lên trên



c) Nhìn xuống dưới



d) Nhìn sang trái 55°



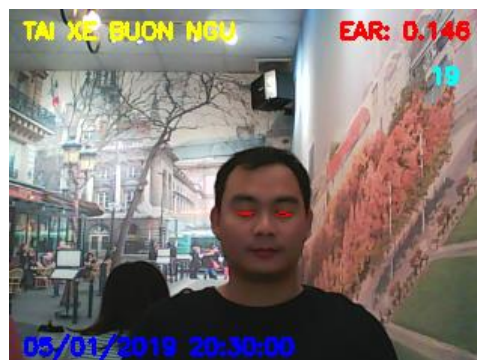
e) Nhìn sang phải 55°

Hình 4.17: Thực nghiệm khi tài xế ngồi gần trong trạng thái buồn ngủ

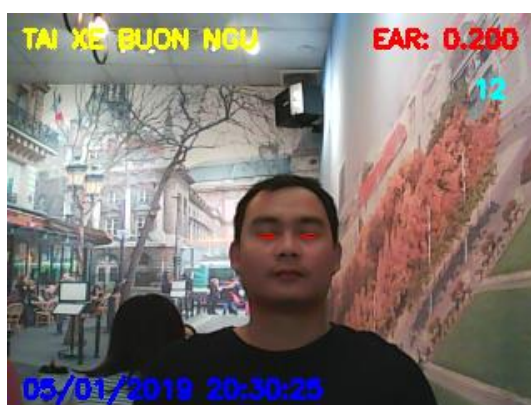
Ở khoảng cách 60 cm, thay đổi góc quét camera có giá trị từ:  $-55^{\circ}$  đến  $55^{\circ}$  thì tỉ lệ mở mắt từ 0,097 đến 0,209.

Khi tài xế ngồi xa, khoảng cách từ tài xế đến camera là 100cm.

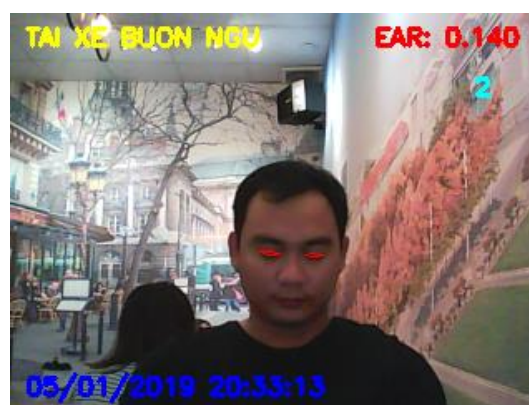
Thiết lập cho hệ thống, khi tỉ lệ mở mắt nhỏ hơn 0,22 và nhắm mắt liên tục 15 frame thì hệ thống xác định tài xế đang có dấu hiệu buồn ngủ.



a) Nhìn trực diện



b) Nhìn lên trên



c) Nhìn xuống dưới



e) Nhìn sang trái 55°



d) Nhìn sang phải 55°

Hình 4.18: Thực nghiệm khi tài xế ngồi xa trong trạng thái buồn ngủ

Ở khoảng cách 60 cm, thay đổi góc quét camera có giá trị từ:  $-55^{\circ}$  đến  $55^{\circ}$  thì tỉ lệ mở mắt từ 0,14 đến 0,2.

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

### **5.1. Kết luận:**

Sau khi nghiên cứu, tác giả đã hoàn thành đề tài “Phát hiện và cảnh báo buồn ngủ của tài xế lái xe dùng thị giác máy tính” với các thành quả đã đạt được:

- Hệ thống phát hiện và cảnh báo buồn ngủ hoạt động ổn định, ở điều kiện ban ngày và trong điều kiện ánh sáng thay đổi nhờ vào đặc trưng HOG và SVM.
- Mạng neural khi được train trên nhiều người giúp cho hệ thống phát hiện chính xác, khách quan hơn so với đặt số frames có định.
- Khi phát hiện tài xế rơi vào trạng thái buồn ngủ; hệ thống sẽ báo động bằng đèn led màu đỏ, cảnh báo bằng giọng nói.
- Thiết bị có thiết kế nhỏ gọn giúp có khả năng ứng dụng thử nghiệm trên xe ô tô.

Bên cạnh đó thì vẫn còn tồn tại một số nhược điểm:

- Chỉ nhận dạng được dấu hiệu buồn ngủ trong điều kiện ánh sáng ban ngày.
- Khi người lái quay mặt lớn hơn góc  $40^\circ$  thì hệ thống không phát hiện mặt.
- Phần cứng Raspberry Pi 3 Model B+ chưa thể đáp ứng cho việc chạy real-time các model sâu hơn.

### **5.2. Hướng phát triển:**

- Dùng mạng CNN để phát hiện khuôn mặt trong nhiều điều kiện ánh sáng phức tạp.
- Thử nghiệm thực tế trên buồng lái ô tô.
- Thu thập thêm dữ liệu để tăng độ khách quan hơn trong xác định dấu hiệu buồn ngủ.
- Thử nghiệm trên các máy tính nhúng mạnh hơn: Jeston Nano, Jeston TX2,..
- Thêm tính năng: báo rung điện thoại, cảnh báo tin nhắn...

## TÀI LIỆU THAM KHẢO

- [1] S. Z. Li and A. K. Jain, Eds., *Handbook of Face Recognition*, 2nd Edition. Springer, 2011.
- [2] W. Zhao, R. Chellappa, P. J. Phillips, and A. Rosenfeld, “Face recognition: A literature survey,” *Acm Comput. Surv. CSUR*, vol. 35, no. 4, pp. 399–458, 2003.
- [3] M. A. Turk and A. P. Pentland, “Face recognition using eigenfaces,” 1991, pp. 586–591.
- [4] P. N. Belhumeur, J. P. Hespanha, and D. J. Kriegman, “Eigenfaces vs. fisherfaces: Recognition using class specific linear projection,” *Pattern Anal. Mach. Intell. IEEE Trans. On*, vol. 19, no. 7, pp. 711–720, 1997.
- [5] T. Ahonen, A. Hadid, and M. Pietikainen, “Face description with local binary patterns: Application to face recognition,” *Pattern Anal. Mach. Intell. IEEE Trans. On*, vol. 28, no. 12, pp. 2037–2041, 2006.
- [6] J. G. Daugman, “Uncertainty relation for resolution in space, spatial frequency, and orientation optimized by two-dimensional visual cortical filters,” *Opt. Soc. Am. J. Opt. Image Sci.*, vol. 2, pp. 1160–1169, 1985.
- [7] N. Dalal and B. Triggs, “Histograms of oriented gradients for human detection,” in *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, 2005, vol. 1, pp. 886–893.
- [8] Saini, V., & Saini, R. (2014). Driver drowsiness detection system and techniques: a review. *International Journal of Computer Science and Information Technologies*, 5(3), 4245-4249.
- [9] Hjelmås, E., & Low, B. K. (2001). Face detection: A survey. *Computer vision and image understanding*, 83(3), 236-274.
- [10] Kazemi, V., & Sullivan, J. (2014). One millisecond face alignment with an ensemble of regression trees. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 1867-1874).
- [11] Tereza, S., & Jan, Č. (2016, February). Real-Time Eye Blink Detection

using Facial Landmarks. In Proc. Computer Vision Winter Workshop.

[12] Hewitt, R. (2007). Seeing with opencv. Servo. March.

[13] T. Ahonen, A. Hadid, and M. Pietikäinen, “Face recognition with local binary patterns,” *Comput. Vis.-ECCV 2004*, pp. 469–481, 2004.

[14] W. Deng, J. Hu, and J. Guo, “Gabor-Eigen-Whiten-Cosine: a robust scheme for face recognition,” *Anal. Model. Faces Gestures*, pp. 336–349, 2005.

[15] M. Yang, L. Zhang, S. C.-K. Shiu, and D. Zhang, “Robust Kernel Representation with Statistical Local Features for Face Recognition,” *IEEE Xplore*, 2013.



