|  |  |  |
| --- | --- | --- |
|  | **TRƯỜNG ĐẠI HỌC BÁCH KHOA**  **KHOA CÔNG NGHỆ THÔNG TIN** |  |

BÁO CÁO

PBL5 – DỰ ÁN KỸ THUẬT MÁY TÍNH

**HỆ THỐNG CẢNH BÁO AN TOÀN KHI LÁI XE**

Giảng viên hướng dẫn: Huỳnh Hữu Trung

|  |  |
| --- | --- |
| NHÓM SINH VIÊN THỰC HIỆN | LỚP HỌC PHẦN |
| Trịnh Huy Nam | 21N14 |
| Phạm Hồng Phúc | 21N14 |
| Nguyễn Huy Nhân | 21N14 |
| Nguyễn Viết Nhật Long | 21N14 |

ĐÀ NẴNG, 06/2024

**TÓM TẮT ĐỒ ÁN**

Trong những năm gần đây, an toàn giao thông đã trở thành một vấn đề cấp bách trên toàn thế giới. Các tai nạn giao thông thường do nhiều nguyên nhân khác nhau, trong đó sai lệch làn đường và buồn ngủ khi lái xe là hai yếu tố đáng lo ngại nhất. Theo thống kê của Tổ chức Y tế Thế giới (WHO), hàng năm có hàng triệu người tử vong và bị thương do tai nạn giao thông, mà một phần lớn trong số đó liên quan đến sự mất tập trung của tài xế hoặc tình trạng buồn ngủ.

Hệ thống cảnh báo sai lệch làn và buồn ngủ cho tài xế ôtô được phát triển nhằm giảm thiểu các tai nạn giao thông gây ra bởi hai yếu tố này. Hệ thống này sử dụng các công nghệ tiên tiến như xử lý hình ảnh, trí tuệ nhân tạo (AI) và cảm biến để phát hiện và cảnh báo kịp thời cho tài xế khi họ có dấu hiệu lệch làn hoặc buồn ngủ.

Mục tiêu chính của dự án này là phát triển và ứng dụng một hệ thống cảnh báo hiệu quả, giúp nâng cao an toàn cho người lái và hành khách.

Dự án này sẽ tập trung vào hai khía cạnh chính:

Cảnh báo sai lệch làn đường: Sử dụng các camera và cảm biến để giám sát vị trí của xe trên làn đường và phát hiện khi xe bắt đầu lệch làn mà không có tín hiệu báo trước.

Cảnh báo buồn ngủ: Sử dụng các công nghệ nhận diện khuôn mặt và theo dõi chuyển động mắt để phát hiện các dấu hiệu mệt mỏi và buồn ngủ của tài xế, từ đó đưa ra cảnh báo sớm.

Cuối cùng xin chân thành cảm ơn thầy đã nhiệt tình hướng dẫn, gợi ý những giải pháp tuyệt vời để chúng em có thể hoàn thành được dự án này.

**BẢNG PHÂN CÔNG NHIỆM VỤ**

|  |  |  |
| --- | --- | --- |
| Sinh viên thực hiện | Các nhiệm vụ | (Đã hoàn thành/Chưa hoàn thành/Không triển khai) |
| Nguyễn Viết Nhật Long | - Viết API, code Server, truyền nhận dữ liệu.  - Code module phần cứng, lắp vi mạch.  - Viết báo cáo. | - Đã hoàn thành tất cả nhiệm vụ |
| Trịnh Huy Nam | - Tìm dữ liệu để huấn luyện model nhận diện đi lệch làn đường.  - Huấn luyện model nhận diện đi lệch làn đường.  - Viết báo cáo. | - Đã hoàn thành tất cả nhiệm vụ |
| Nguyễn Huy Nhân | - Xây dựng pipeline xử lý ảnh làn đường.  - Huấn luyện model nhận diện đi lệch làn đường.  - Viết báo cáo. | - Đã hoàn thành tất cả nhiệm vụ |
| Phạm Hồng Phúc | - Nghiên cứu mô hình nhận diện buồn ngủ.  - Huấn luyện model nhận diện ngủ gật.  - Viết báo cáo. | - Đã hoàn thành tất cả nhiệm vụ |

**MỤC LỤC**

[1. Giới thiệu 7](#_Toc169250062)

[1.1 Thực trạng sản phẩm 7](#_Toc169250063)

[1.2. Các vấn đề cần giải quyết 7](#_Toc169250064)

[1.3. Đề xuất giải pháp tổng quan 8](#_Toc169250065)

[2. Giải pháp 9](#_Toc169250066)

[2.1. Giải pháp phần cứng và truyền thông 9](#_Toc169250067)

[2.1.1 Sơ đồ tổng quan hệ thống 9](#_Toc169250068)

[2.1.2 Sơ đồ hoạt động của hệ thống 9](#_Toc169250069)

[2.1.3. Phần cứng 11](#_Toc169250070)

[2.1.4. Truyền thông 14](#_Toc169250071)

[2.2. Giải pháp TTNT/KHDL 16](#_Toc169250072)

[2.2.1. Mô hình nhận diện xe đi lệch làn đường 16](#_Toc169250073)

[2.2.2. Mô hình nhận diện buồn ngủ 20](#_Toc169250074)

[2.3. Giải pháp phần mềm 20](#_Toc169250075)

[2.3.1. Luồng xử lý hệ thống 20](#_Toc169250076)

[2.3.2. Giải pháp nhận diện xe đi lệch làn đường 21](#_Toc169250077)

[2.3.3. Giải pháp nhận diện buồn ngủ 21](#_Toc169250078)

[2.3.4. Server 22](#_Toc169250079)

[3. Kết quả 24](#_Toc169250080)

[3.1. Nhận diện lệch làn đường 24](#_Toc169250081)

[3.1.1. Tập dữ liệu 24](#_Toc169250082)

[3.1.2. Huấn luyện 24](#_Toc169250083)

[3.2. Nhận diện buồn ngủ 26](#_Toc169250084)

[3.3. Đánh giá 26](#_Toc169250085)

[3.3.1. Kết quả deploy 26](#_Toc169250086)

[3.3.2. Kết quả phần cứng hệ thống 26](#_Toc169250087)

[3.3.3. Kết quả vận hành server 26](#_Toc169250088)

[3.3.4. Kết quả nhận diện làn đường 27](#_Toc169250089)

[4. Kết luận 29](#_Toc169250090)

[4.1. Kết luận: 29](#_Toc169250091)

[5. Hướng phát triển 30](#_Toc169250092)

[5. Danh mục tài liệu tham khảo 30](#_Toc169250093)

**MỤC LỤC HÌNH ẢNH**

[Hình 1. Sơ đồ tổng quan hệ thống 10](#_Toc169254710)

[Hình 2 . Sơ đồ IoT 11](#_Toc169254711)

**MỤC LỤC BẢNG BIỂU**

[Bảng 1. Đề xuất giải pháp. 9](#_Toc169254825)

[Bảng 2. Danh sách phần cứng. 12](#_Toc169254826)

[Bảng 3. Chi phí linh kiện 14](#_Toc169254827)

**CÁC YÊU CẦU VIẾT BÁO CÁO**

**VỀ HÌNH THỨC:**

* Định dạng: căn chỉnh lề kiểu Justify, font Times new roman, cỡ chữ 12, giãn dòng 1.5, cách trước 6pt; hình ảnh phải đánh số, có chú thích bên dưới hình; bảng phải đánh số, có chú thích bên trên bảng.
* Báo cáo có độ dài từ 20 đến 40 trang và không đóng bìa gương (chỉ cần bìa mềm có màu) gồm: trang bìa (trang 1 của file hướng dẫn này), tóm tắt, bảng phân công nhiệm vụ,mục lục, danh sách hình vẽ, danh sách bảng, và các trang nội dung.

**VỀ NỘI DUNG:**

Khi viết báo cáo cần chú ý:

* Báo cáo để người khác đọc hiểu được logic vấn đề và giải pháp, và sau khi đọc xong thì người đọc có thể làm lại được sản phẩm đồ án theo nội dung báo cáo.
* Báo cáo không phải để kể lể các việc đã triển khai và/hoặc để liệt kê mã nguồn.

Báo cáo gồm 04 phần sau:

# Giới thiệu

## 1.1 Thực trạng sản phẩm

Hiện nay, các hệ thống cảnh báo sai lệch làn và buồn ngủ cho tài xế đã và đang được triển khai và áp dụng trên nhiều dòng xe ôtô hiện đại. Các nhà sản xuất xe hơi lớn như BMW, Mercedes-Benz, Toyota, và Volvo đã tích hợp những hệ thống này vào các mẫu xe cao cấp của họ, với những công nghệ tiên tiến nhằm nâng cao an toàn cho người lái và hành khách. Tuy nhiên, giá thành lại rất đắt và hầu hết được tích hợp sẵn vào xe, Do đó, chúng em tiến hành thử nghiệm với đề tài này nhằm tìm ra giải pháp tốt với chi phí thấp hơn.

## 1.2. Các vấn đề cần giải quyết

* Cần có các thiết bị phần cứng để thu nhận dữ liệu.
* Phát hiện được làn đường và từ đó kiểm tra xem xe có đang đi lệch hay không.
* Nhận diện được khuôn mặt tài xế đang trong trạng thái buồn ngủ hay tỉnh táo
* Hệ thống xử lý theo thời gian thực và đảm bảo độ trễ thấp nhất có thể

## Đề xuất giải pháp tổng quan

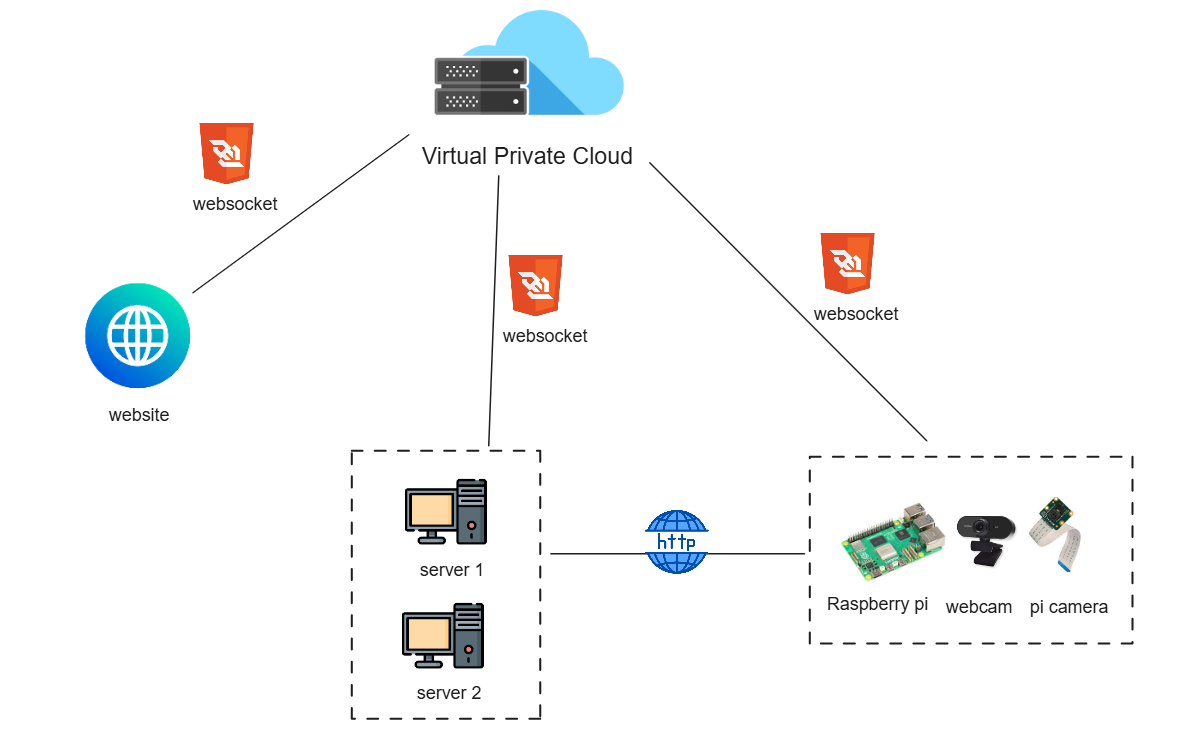
Bảng 1. Đề xuất giải pháp.

|  |  |
| --- | --- |
| **Vấn đề** | **Giải pháp đề xuất** |
| Phần cứng | Raspberry Pi 4 4GB RAM  02 camera (USB Webcam & Pi Camera)  Máy chủ ảo  Quạt tản nhiệt |
| Phát hiện buồn ngủ | Xây dựng mô hình phát hiện khuôn mặt, mắt và trạng thái của mắt  Huấn luyện trên Google Colab, Kaggle |
| Phát hiện sai lệch làn | Xây dựng mô hình phát hiện sai lệch làn đường, sử dụng các thuật toán xử lý như Hough.  Huấn luyện trên Google Colab, Kaggle |
| Server | Sử dụng websocket để chuyển tiếp các yêu cầu giữa các client để giải quyết vấn đề truyền thông giữa các mạng |
| Website | Viết bằng framework VueJs  Có chức năng giám sát video lái của tài xế  Bật tắt được hệ thống cảnh báo |

# 2. Giải pháp

## 2.1. Giải pháp phần cứng và truyền thông

### 2.1.1 Sơ đồ tổng quan hệ thống



Hình 1. Sơ đồ tổng quan hệ thống

Hệ thống bao gồm Raspberry Pi 4 và hai webcam dùng để stream video, hai server dùng để dự đoán kết quả, một server sẽ dự đoán trạng thái buồn ngủ và một server sẽ dự đoán sai lệch làn đường, hai server này sẽ giao tiếp với raspberry thông qua giao thức HTTP. Một server được deploy lên VPS (Virtual Private Cloud) dùng để chuyển tiếp trạng thái và video để giải quyết vấn đề truyền thông giữa các mạng, website dùng để giám sát trạng thái hoạt động của raspberry cũng như hiển thị video đã được hai server xử lý.

### 2.1.2 Sơ đồ hoạt động của hệ thống

A diagram of a diagram

Description automatically generated

Hình 2 . Sơ đồ IoT

Hệ thống bao gồm hai thành phần hệ thống, một thành phần nằm trong mạng nội bộ và một thành phần nằm trong mạng công khai

Thành phần nằm trong mạng nội bộ bao gồm: raspberry pi 4, sử dụng hai camera, một nút nhấn, 2 LED vàng để mô phỏng xinhan và gồm hai server dùng để xử lý hai module riêng biệt là nhận diện sai lệch làn đường và nhận diện buồn ngủ

Thành phần nằm trong mạng công khai bao gồm: một VPS (Virtual Private Server) dùng để deploy server python có nhiệm vụ chuyển tiếp các gói tin hình ảnh ở dạng buffer nhận được từ hai server AI, một website sử dụng framework Vuejs để hiển thị video đã xử lý và điều khiển raspberry

Đầu tiên Raspberry sẽ nhận dữ liệu hình ảnh từ camera USB, sau đó sẽ khởi tạo một luồng stream thông qua plugin mjpeg-streamer ở port 8080, tiếp đó sẽ đọc dữ liệu ở Pi camera sau đó cũng sẽ khởi tạo một luồng stream qua giao thức HTTP ở port 8000. Server AI nhận diện buồn ngủ sẽ đọc luồng stream của camera USB để xử lý và Server AI nhận diện làn đường sẽ đọc luồng stream của Pi camera để xử lý, cả hai sẽ xử lý theo thời gian thực sau đó sẽ trả kết quả về cho raspberry, raspberry sẽ đọc kết quả sau đó sẽ truyền tín hiệu cho 2 đèn LED sáng. Đồng thời khi 2 server AI xử lý dự đoán thì sẽ gửi các frame đã dự đoán cho server python để hiển thị video thông qua kết nối websocket, nhờ vậy sẽ đạt được fps cao hơn và độ trễ thấp hơn so với sử dụng giao thức HTTP.

Khi người dùng bấm nút bật/ tắt chế độ cảnh báo trên website, website sẽ gửi yêu cầu cho server python, sau đó server python chuyển tiếp yêu cầu cho raspberry để xử lý

### 2.1.3. Phần cứng

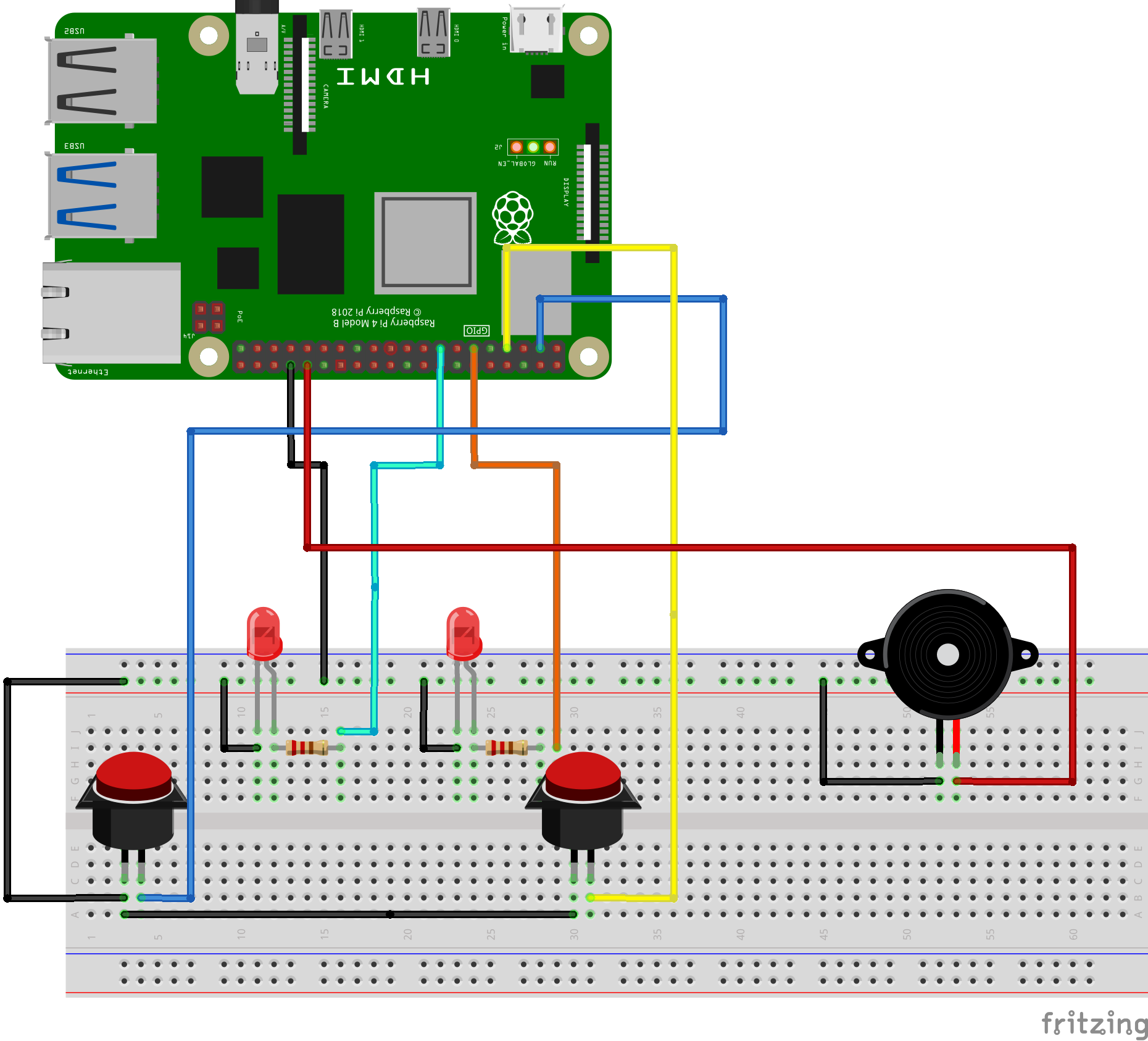
Bảng 2. Danh sách phần cứng.

|  |  |  |
| --- | --- | --- |
| Linh kiện | Hình ảnh | Thông số, hoạt động |
| Webcam Aukey PC-W3 |  | - Độ phân giải: HD 1080p  - Góc camera: 55 độ  - Kích thước:7.3cm x3.2cm x 6.7cm  - Khối lượng: 75g |
| Raspberry PI 4 |  | - Bộ vi xử lý 64-bit quad-core ARM Cortex-A72  - Kết nối không dây: Wi-Fi IEEE 802.11ac 2.4 GHz và 5.0 GHz, Bluetooth 5.0, BLE  - Bộ nhớ LPDDR4 2GB  - Nguồn 2.5A  - 1 cổng Gigabit Ethernet  - 2 cổng HDMI  - 1 Jack cắm  - USB 3.0 |
| LED |  | - Điện áp: 1.9 - 3.6V  - Dòng điện: 20mA  - Ánh sáng phủ màu đỏ  - Kích thước: 5mm |
| Nút nhấn |  | - Phân loại: Nút nhấn DIP  - Kiểu chân: DIP ( chân cắm )  - Số chân: 2 chân  - Kích thước: 6 x 6 x 5cm  - Màu sắc: Đen |
| Còi báo |  | - Nguồn : 3.5V - 5.5V  - Dòng điện tiêu thụ: <25mA  - Tần số cộng hưởng: 2300Hz ± 500Hz  - Biên độ âm thanh: >80dB  - Nhiệt độ hoạt động:-20°C đến 70°C  - Kích thước : Đường kính 12mm, cao 9,7mm |

Bảng 3. Chi phí linh kiện

|  |  |  |
| --- | --- | --- |
| **Tên linh kiện** | **Đơn giá (VNĐ)** | **Ghi chú** |
| Raspberry Pi model B 4GB ram | 1.800.000 | Mua |
| 1 x VPS | 200.000 | Mua |
| 1 x Thẻ nhớ 64GB | 80.000 | Mua |
| 1 x Webcam | 100.000 | Mua |
| 1 x Pi Camera | 100.000 | Mua |
| Các linh kiện khác | 200.000 | Mua |
| Thành tiền: 2.000.000 VNĐ | | |

Sơ đồ lắp mạch:



Hình 3. Sơ đồ lắp mạch.

Bảng 4. Nối mạch đèn LED với Raspberry PI 4

|  |  |
| --- | --- |
| LED | Raspberry PI 4 |
| Anode (Nối với điện trở 220 Ohm) | GPIO22 (LED trái), GPIO17 (LED phải) |
| Canode | GND |

Bảng 5. Nối mạch nút nhấn với Raspberry PI 4

|  |  |
| --- | --- |
| Nút nhấn | Raspberry PI 4 |
| Leg 1 | GPIO02 (nút trái), GPIO04 (nút phải) |
| Leg 0 | GND |

Bảng 6. Nối mạch còi báo với Raspberry PI 4

|  |  |
| --- | --- |
| Còi báo | Raspberry PI 4 |
| Pin 1 | GPIO12 |
| Pin 2 | GND |

### 2.1.4. Truyền thông

**Kết nối có dây:**

* Camera kết nối với raspberry thông qua cổng USB

- Pi camera kết nối với raspberry thông qua cáp FFC

**Kết nối không dây:**

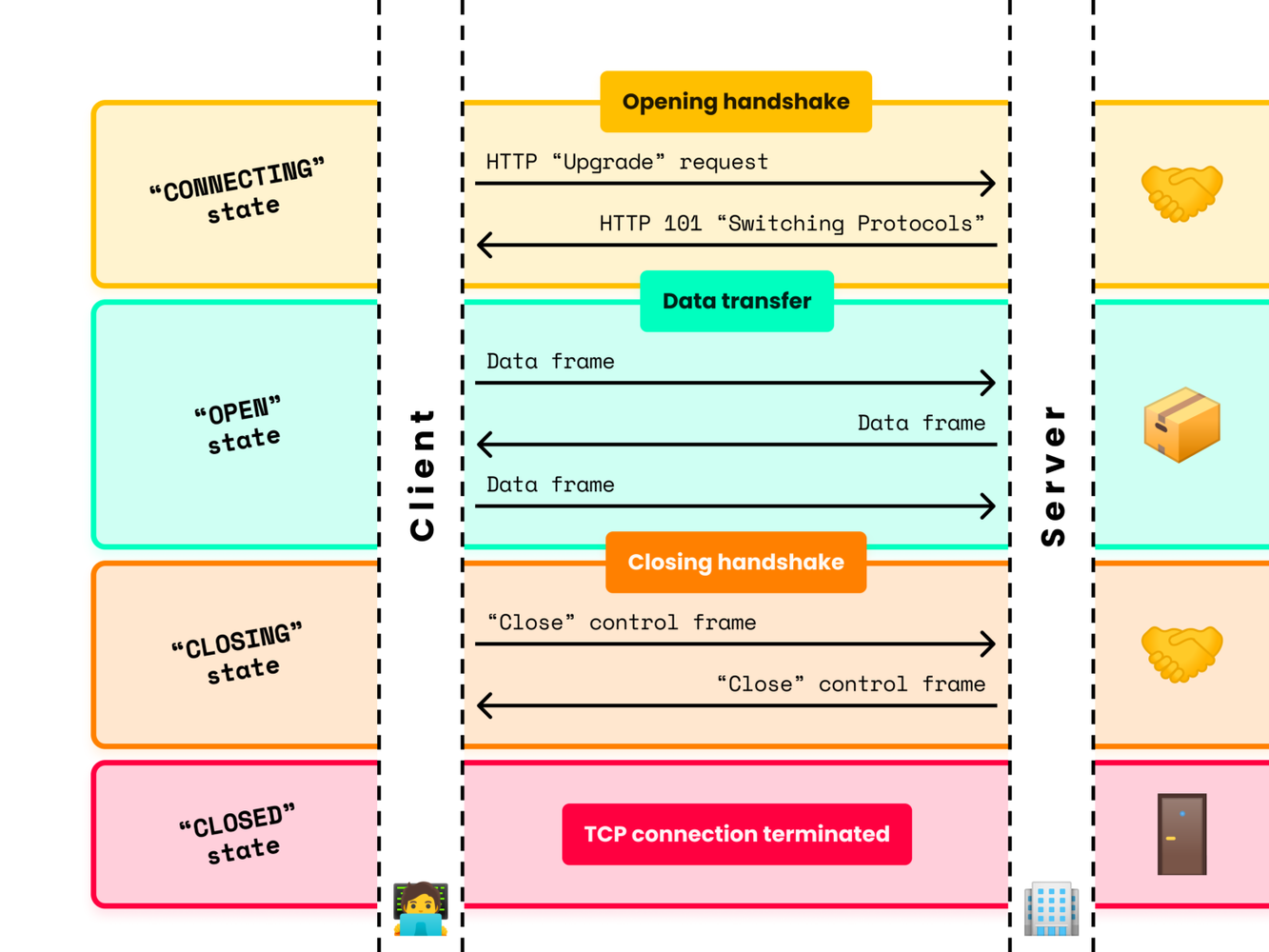
**HTTP** (Hypertext Transfer Protocol): là giao thức truyền tải siêu văn bản. Đây là giao thức tiêu chuẩn cho World Wide Web (www) để truyền tải dữ liệu dưới dạng văn bản, âm thanh, hình ảnh, video từ Web Server tới trình duyệt web của người dùng và ngược lại. HTTP hoạt động theo mô hình Client (máy khách) – Server (máy chủ). Việc truy cập website được tiến hành dựa trên các giao tiếp giữa 2 đối tượng trên. Khi truy cập một trang web qua giao thức HTTP, trình duyệt sẽ thực hiện các phiên kết nối đến server của trang web đó thông qua địa chỉ IP do hệ thống phân giải tên miền DNS cung cấp. Máy chủ sau khi nhận lệnh, sẽ trả về lệnh tương ứng giúp hiển thị website, bao gồm các nội dung như: văn bản, ảnh, video, âm thanh,…

A close-up of a diagram

Description automatically generated

Hình 4. Minh họa giao thức HTTP

**Websocket:** Giao thức WebSocket là một tiêu chuẩn mở được hỗ trợ rộng rãi để phát triển các ứng dụng thời gian thực. Các phương pháp trước đây để mô phỏng kết nối song công hoàn toàn dựa trên Polling, một phương pháp đồng bộ trong đó client đưa ra yêu cầu tới server để xem liệu có bất kỳ thông tin nào có sẵn hay không. WebSocket là một giao thức truyền thông máy tính (computer communication protocol), cung cấp các kênh liên lạc dạng full-duplex (song công) qua một kết nối TCP. Giao thức WebSocket hiện nay được bao hàm ngay ở mục Connectivity trong đặc tả của HTML5.

****

Hình 5. Minh họa Websocket

## 2.2. Giải pháp TTNT/KHDL

### 2.2.1. Mô hình nhận diện xe đi lệch làn đường

**Hough Transform:** là một kỹ thuật phổ biến để phát hiện bất kỳ hình dạng nào nếu có thể biểu diễn hình dạng đó dưới dạng toán học. Có thể phát hiện hình dạng ngay cả khi nó bị hỏng hoặc bị bóp méo một chút. Ý tưởng chung của việc phát hiện đường thẳng trong thuật toán này là tạo mapping từ không gian ảnh (A) sang một không gian mới (B) mà mỗi đường thẳng trong không gian (A) sẽ ứng với một điểm trong không gian (B).

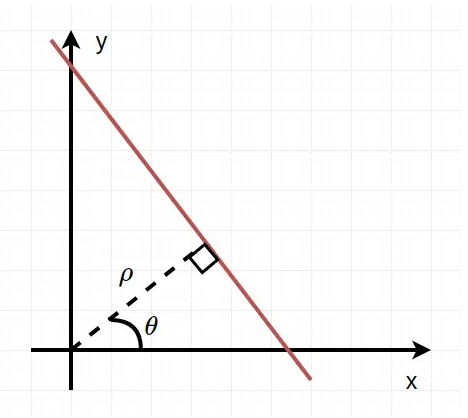
**Phương trình đường thẳng trong không gian ảnh (A):** Phương trình đường thẳng cơ bản sẽ được biểu diễn theo 2 tham số a và b như sau



Tuy nhiên, với cách biểu diễn này, giá trị của góc nghiêng 𝑎 trải dài từ –∞ đến +∞. Có thể lấy ví dụ, để có được phương trình đường Oy (x=0) thì 𝑎 phải tiến tới ∞. Thuật toán Hough Transform yêu cầu các giá trị a, b nằm trong một khoảng xác định (hay bị chặn trên dưới), ta phải sử dụng hệ tọa độ cực để biểu diễn phương trình đường thẳng.

ρ=xcos(θ)+ysin(θ).

Xét thấy trong phương trình tọa độ cực, giá trị của góc θ có thể bị chặn lại trong khoảng [0, π). Trên thực tế, không gian ảnh là không gian hữu hạn (bị chặn lại bởi các cạnh của ảnh), do vậy giá trị 𝜌 cũng bị chặn.



Hình 6. Đường thẳng trong hệ tọa độ cực

**Mapping giữa không gian ảnh (A) và không gian Hough (B)**

* Từ một đường thẳng trong không gian ảnh (A) với 2 tham số ρ và θ, chúng ta sẽ map sang không gian Hough (B) thành một điểm.

A diagram of a graph

Description automatically generated with medium confidence

Hình 7. Mapping một đường thẳng từ không gian ảnh sang không gian Hough

* Từ một điểm trong không gian ảnh, chúng ta lại có được một hình sin trong không gian Hough:

A graph of a function

Description automatically generated

Hình Mapping một điểm từ không gian ảnh sang không gian Hough

Các điểm nằm trên cùng một đường thẳng lại có biểu diễn là các hình sin giao nhau tại một điểm trong không gian Hough. Đây là nơi xuất phát ý tưởng của thuật toán Hough Transform. Chúng ta sẽ dựa vào các điểm giao nhau này để suy ngược lại phương trình đường thẳng trong không gian ảnh.

A diagram of a line and a graph

Description automatically generated

Hình: Mapping nhiều điểm thẳng hàng từ không gian ảnh sang không gian Hough

Mỗi đường thẳng khác nhau sẽ tạo thành một điểm sáng (nơi giao nhau của nhiều hình sin) trên không gian Hough. Dưới đây là sự biểu diễn 2 đường thẳng trong không gian Hough.

A black square with green arrows

Description automatically generated

Hình: Biểu diễn 2 đường thẳng trong không gian Hough

**- Tiền xử lý**

Hough Transform yêu cầu đầu vào là một ảnh nhị phân. Trên thực tế ảnh sẽ được đưa về dạng ảnh xám, áp dụng các thuật toán lọc biên để xác định các đường biên trong ảnh. Ở đây chúng ta sẽ sử dụng thuật toán Canny để lọc biên.

A road with trees on the side

Description automatically generated

### 2.2.2. Mô hình nhận diện buồn ngủ

## 2.3. Giải pháp phần mềm

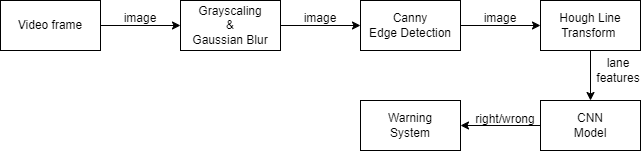
### 2.3.1. Luồng xử lý hệ thống

A diagram of a raspberry

Description automatically generated

Đầu tiên raspberry sẽ nhận video thu được từ camera 1 và camera 2 và lần lượt gửi đến cho server 1 và server 2 để lần lượt dự đoán trạng thái khuôn mặt và trạng thái làn đường, sau khi dự đoán xong thì sẽ trả kết quả về lại cho raspberry, raspberry sẽ xử lý logic và tính toán để quyết định trạng thái của còi và hệ thống đèn LED

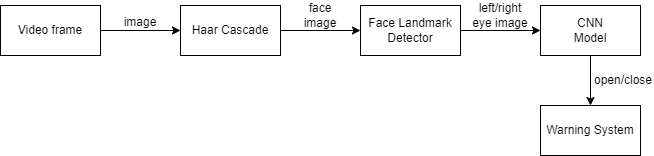
### 2.3.2. Giải pháp nhận diện xe đi lệch làn đường



Hình 3. Giải pháp nhận diện xe đi lệch làn đường.

* Thiết lập một camera để liên tục theo dõi khuôn mặt của người lái xe.
* Chuyển ảnh màu sang ảnh xám và sử dụng bộ lọc Gaussian để làm mờ ảnh, giúp giảm nhiễu trong ảnh.
* Sử dụng bộ lọc Canny để trích xuất cạnh trong ảnh đã làm mờ.
* Trích vùng ảnh làn đường cần thiết và sử dụng phép biến đổi Probabilistic Hough Line để xác định các tọa độ của hai cạnh bên trái và bên phải của làn đường.
* Dùng mô hình CNN để nhận diện xe đang ở trong làn đường hay đang rời khỏi làn đường.
* Đưa ra cảnh báo tài xế đi lệch làn đường khi phát hiện xe rời khỏi làn mà xi nhan không được bật.

### 2.3.3. Giải pháp nhận diện buồn ngủ



Hình 4. Giải pháp nhận diện tài xế ngủ gật.

* Thiết lập một camera để liên tục theo dõi khuôn mặt của người lái xe.
* Sử dụng mô hình “Haar cascades frontal face default” để xác định vùng khuôn mặt.
* Đối với mỗi khuôn mặt xác định được, sử dụng mô hình “Shape predictor 68 face landmarks” để xác định tọa độ các điểm đặc trưng của mắt.
* Trích xuất vùng mắt từ khuôn mặt sử dụng các điểm đặc trưng đã tìm được.
* Dùng mô hình CNN để phân biệt mắt nhắm hay mở.
* Đưa ra cảnh báo tài xế đang ngủ gật nếu phát hiện mắt nhắm trong nhiều khung hình video liên tiếp nhau..

## 2.3.4. Server

Server được deploy lên VPS (Virtual Private Server), sử dụng cloudflare để map domain unlockscan.site đến VPS

A diagram of a cloud computing network

Description automatically generated

Nhóm mua domain ở website <namecheap.com>, để có thể map được domain mua từ một nhà cung cấp bất kỳ đến được VPS thì trước hết phải truy cập trang quản trị nơi mua domain. Sau đó tìm đến mục nameserver, chọn custom DNS (nếu có) và chỉnh sửa nameserver1 và nameserver2 lần lượt đến gail.ns.cloudflare.com và noah.ns.cloudflare.com sau đó lưu lại.

Truy cập trang quản trị của Cloudflare ở đường dẫn <https://dash.cloudflare.com/>, thêm domain và chuyển đến trang DNS và thêm các record cho phù hợp với VPS

A screenshot of a computer

Description automatically generated

Kết hợp Cloudflare với Nginx cũng là một hình thức để tăng tính bảo mật cho server, giúp ẩn giấu IP thực sự của server ở phía sau Cloudflare

A diagram of a cloudflare

Description automatically generated

Trong private network nhóm chúng em có 3 server, server chạy ở port 5000 sẽ dùng để chuyển tiếp các gói tin hình ảnh liên quan đến nhận diện khuôn mặt, vì vậy khi client thiết lập một kết nối đến wss://face.unlockscan.site thì đầu tiên các gói tin sẽ đi đến Cloudflare và được Cloudflare chuyển tiếp đến server thông qua public IP, khi các gói tin đến server chúng sẽ được Nginx xử lý, Nginx sẽ kiểm tra các thông tin của gói tin, trong đó bao gồm các nơi chúng xuất phát (ví dụ face.unlockscan.site), tùy vào config của Nginx mà gói tin đó sẽ được chuyển đến server nào trong private network của VPS, ở đây nhóm chúng em đã config để các gói tin đến từ face.unlockscan.site sẽ đi đến server chạy port 5000, các gói tin từ lane.unlockscan.site sẽ đi dến server chạy port 5001, các gói tin đi đến từ unlockscan.site sẽ đi đến server chạy port 5002.

# 3. Kết quả

### 3.1. Nhận diện lệch làn đường

### 3.1.1. Tập dữ liệu

Trong phần huấn luyện mô hình nhận diện làn đường, nhóm sử dụng tập dữ liệu CULane. Thông tin chi tiết về tập dữ liệu:

Nguồn: <https://xingangpan.github.io/projects/CULane>

Mô tả: CULane là bộ dữ liệu quy mô lớn dành cho nghiên cứu học thuật về phát hiện làn đường giao thông. Được thu thập bởi các camera gắn trên sáu phương tiện khác nhau do những người lái xe khác nhau ở Bắc Kinh điều khiển. Hơn 55 giờ video đã được thu thập và 133.235 khung hình được trích xuất. Vì mục tiêu đề tài là nhận diện đúng làn và sai làn nên nhóm đã lọc thủ công ra 3000 ảnh xe trong trạng thái chuyển làn và 3000 ảnh trong trạng thái đi đúng làn.

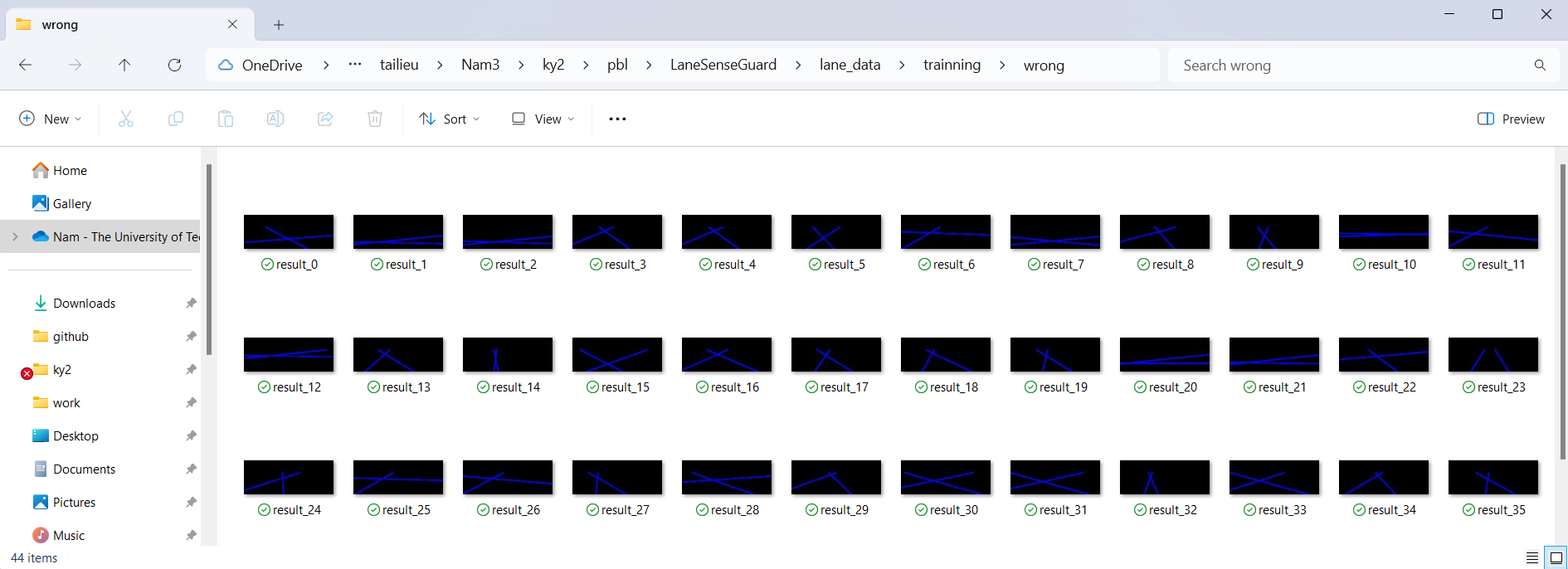
Phân chia dữ liệu: Train : 4200 (70%), Validation: 900, Test: 900

### 3.1.2. Huấn luyện

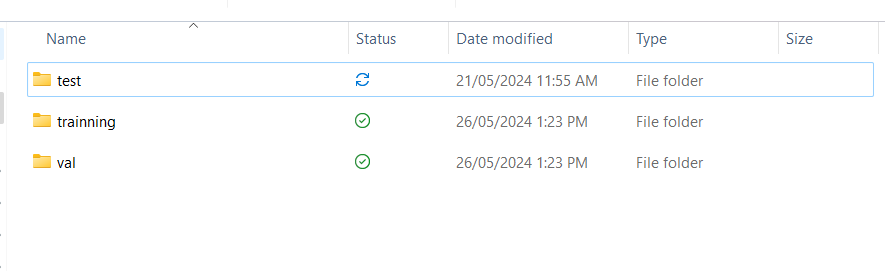
Sau khi đã chuẩn bị và phân chia dữ liệu theo tỉ lệ phù hợp. Nhóm đã tiến hành sử dụng phương pháp Hough Transform để tiền xử lý dữ liệu trước khi đưa vào huấn luyện mô hình

A screenshot of a computer

Description automatically generated



Nhóm xử lý và phân chia data theo các folder, gồm ba folder: test, tran, val



Sau khi đã có data nhóm tiến hành upload data lên Google Drive để lưu trữ và đưa model lên Google Colab để tiến hành huấn luyện. Sau khi huấn luyện mô hình với 300 epoch đã cho ra mô hình nhận diện lệch làn. Trong quá trình training nhóm đã lưu lại các kết quả train ở từng epoch để tiện cho việc vẽ đồ thị.

## 3.2. Nhận diện buồn ngủ

## 3.3. Đánh giá

### 3.3.1. Kết quả deploy

- Server đã được deploy lên mạng thành công

- Các client có thể kết nối đến các websocket như: wss://lane.unlockscan.site, wss://face.unlockscan.site

### 3.3.2. Kết quả phần cứng hệ thống

- Hệ thống hoạt động ổn định, nhiệt độ của hệ thống giao động mức 45 độ C đến 47 độ C nếu bật hệ thống quạt tản nhiệt, trường hợp không bật quạt tản nhiệt thì nhiệt độ 60 độ C đến 65 độ C

- Camera thu đầy đủ hình ảnh, không có hiện tượng nóng lên, không có hiện tượng giật lag

### 3.3.3. Kết quả vận hành server

- Server hoạt động ổn định, không có hiện tượng giật lag.

- Tốc độ duy trì ở mức ổn định, nhiều client kết nối nhưng không có hiện tượng chậm trễ

- Server luôn duy trì trạng thái hoạt động 99,99%, không có hiện tượng downtime

A screenshot of a computer

Description automatically generated

- Server tối ưu RAM tốt, không bị tràn RAM khi chạy 3 server websocket

A black background with white text

Description automatically generated

### 3.3.4. Kết quả nhận diện làn đường

- Về cơ bản hệ thống đã nhận diện được làn đường, tuy nhiên trong một số điều kiện như vướng bóng cây, có vật cản thì hệ thống nhận diện có phần sai lệch

A screenshot of a computer program

Description automatically generated

A screenshot of a computer

Description automatically generated

# 4. Kết luận

## 4.1. Kết luận:

* Đối với module cảnh báo sai làn đường, ta phải nhận thấy rằng việc nhận diện chính xác các làn đường không phải lúc nào cũng đạt được. Có nhiều yếu tố gây ảnh hưởng đến quá trình nhận diện như bóng râm lớn che khuất làn đường, ánh nắng quá mạnh hay trời tối, và cả các kí hiệu lạ khác xuất hiện bên trong làn đường.
* Đối với module nhận diện tài xế mất tập trung, mô hình vẫn tồn tại những sai lệch. Ví dụ như việc phân biệt giữa tài xế mở hay nhắm mắt vẫn còn độ sai số chấp nhận được. Tuy nhiên, với việc tiếp tục phát triển và cải thiện mô hình, ta có thể giảm thiểu các sai lệch này để tăng cường tính hiệu quả và độ tin cậy của model.
* Đối với việc sử dụng API, thời gian truyền nhận vẫn còn khá lớn. Tuy nhiên, điều này có thể được giảm thiểu thông qua việc cải tiến và tối ưu hóa các thuật toán.
* Sản phẩm mang tính chất nghiên cứu, giáo dục.
* Đồ án đã thực hiện tối ưu hóa việc truyền nhận dữ liệu nhưng vẫn còn những giới hạn về phần cứng.

# 5. Hướng phát triển

**Phát hiện khuôn mặt:** Hiện tại mô hình nhận diện được khuôn mặt khá là chuẩn xác, tuy nhiên tùy vào một số điều kiện khách quan như độ phân giải camera kém, điều kiện thiếu ánh anh, hoặc bị cháy sáng thì mô hình nhận diện có phần sai lệch. Tương lai nhóm sẽ mở rộng và cải tiến mô hình thêm bằng cách huấn luyện thêm bằng các tập dự liệu ở nhiều điều kiện đa dạng hơn để gia tăng độ chính xác

**Nhận diện vùng mắt:** Hiện tại mô hình chưa thể nhận diện được các vùng mắt ở các điều kiện khó khăn như: mắt nhỏ, mắt híp, người có đeo kính. Nhóm sẽ tiếp tục mở rộng bộ dữ liệu để huấn luyện thêm, tích cực thu thập các dữ liệu về vùng mắt trong nhiều điều kiện khác nhau, đặc biệt là các trường hợp đeo kính.

**Server xử lý:** Hiện tại nhóm đang xử lý dự đoán mô hình ngay trực tiếp lên laptop cá nhân, điều này sẽ gây bất tiện trong việc xử lý. Để giải quyết vấn đề này buộc nhóm phải chi kinh phí để thuê một cloud server nhưng đây có lẽ sẽ là bài toán kinh tế lớn đối với nhóm.

**Xử lý stream video thông qua VPS:** Hiện tại nhóm đang xử lý việc stream video cho client thông qua một VPS, tuy nhiên nhóm đang mở 03 kết nối websocket để xử lý 3 tác vụ riêng biệt là: stream video làn đường, stream video khuôn mặt và xử lý quản lý hệ thống. Giải pháp này thực sự chưa tối ưu nên trong tương lai nhóm sẽ tìm giải pháp để kết hợp 3 tác vụ này lại chung một kết nối.

**Website:** Hiện tại website đã hoàn thiện tuy nhiên chưa có nhiều chức năng quản lý và điều khiển cũng như phân quyền. Tương lai nhóm sẽ phát triển thêm tính năng phân quyền để mỗi người có một tài khoản cũng như quyền hạn truy cập.

**Phần cứng:** Phần cứng hiện tại đã đủ đáp ứng tuy nhiên camera chất lượng còn thấp nên sẽ dẫn đến một số vấn đề không mong muốn trong việc dự đoán của mô hình. Đây cũng là một bài toán về kinh tế đối với nhóm tuy nhiên nhóm sẽ ưu tiên trước việc nâng cấp chất lượng của hệ thống camera.

# 5. Danh mục tài liệu tham khảo

[1] MDN Web Docs,

Nhóm SV liệt kê các TLTK đã trích dẫn (cite) trong báo cáo tại đây.

(Về quy định trích dẫn và lập danh mục TLTK, SV tham khảo hướng dẫn Số 30/HD-ĐHBK ban hành năm 2017 tại đây: <https://drive.google.com/file/d/1oWdUBoHOyAbK1tsax-kQnvuXXCMjmHlU/view?usp=sharing>)