**TRƯỜNG ĐẠI HỌC GIAO THÔNG VẬN TẢI**



**BÁO CÁO TỔNG KẾT**

**ĐỀ TÀI NGHIÊN CỨU KHOA HỌC CỦA SINH VIÊN**

**NĂM 2023**

**Ứng dụng xử lý ảnh nhận dạng cử chỉ bàn tay**

**điều khiển robot chuyển động**

**Sinh viên thực hiện**

**Lê Đức Tuấn – 201404129**

Trần Thái Hưng - 201403981

Nguyễn Tiến Hải - 201403932

Nguyễn Minh Tuấn - 201414132

Nguyễn Ngọc Thắng – 201404094

Lớp: KTĐT-THCN2-K61 Khoa: Điện-Điện tử

**Người hướng dẫn:** TS.Nguyễn Thúy Bình

**MUC LỤC**

[**Danh mục những từ viết tắt** 3](#_Toc136449474)

[**Danh mục bảng biểu** 4](#_Toc136449475)

[**MỞ ĐẦU** 5](#_Toc136449476)

[**1.** **Đặt vấn đề .** 5](#_Toc136449477)

[**2. Lý do chọn đề tài .** 5](#_Toc136449478)

[**3. Mục tiêu nghiên cứu .** 5](#_Toc136449479)

[**4. Đối tượng nghiên cứu** 6](#_Toc136449480)

[**5. Phạm vi nghiên cứu của đề tài .** 6](#_Toc136449481)

[**CHƯƠNG I . CƠ SỞ LÝ THUYẾT.** 7](#_Toc136449482)

[**1.1. Tổng quan về ngôn ngữ Python.** 7](#_Toc136449483)

[**1.1.1.Python là gì?** 7](#_Toc136449484)

[**1.1.2.Ứng dụng của Python** 7](#_Toc136449485)

[**1.1.3.Truyền thông nối tiếp ở Python.** 8](#_Toc136449486)

[**1.2.Tổng quan về nhận dạng đối tượng.** 9](#_Toc136449487)

[**1.2.1.Nhận dạng đối tượng là gì ?** 9](#_Toc136449488)

[**1.2.2.Ứng dụng của nhận dạng đối tượng .** 9](#_Toc136449489)

[**1.3.Tổng quan về mediapipe.** 10](#_Toc136449490)

[**1.3.1.Mediapipe là gì?** 10](#_Toc136449491)

[**1.3.2.MediaPipe Hand Landmarker.** 13](#_Toc136449492)

[**1.4.Tổng quan giao thức SPI.** 14](#_Toc136449493)

[**CHƯƠNG II . TỔNG QUAN BÀI TOÁN NHẬN DẠNG BÀN TAY.** 16](#_Toc136449494)

[**2.1. Những vấn đề cơ bản trong xử lý nhận diện bàn tay .** 16](#_Toc136449495)

[**2.1.1. Xử lý ảnh đầu vào.** 16](#_Toc136449496)

[**2.1.2. Đánh dấu các điểm trên bàn tay.** 17](#_Toc136449497)

[**2.2. Bài toán nhận dạng bàn tay .** 17](#_Toc136449498)

[**2.2.1. Chi tiết nhiệm vụ .** 17](#_Toc136449499)

[**2.2.2. Thực hiện nhiệm vụ .** 18](#_Toc136449500)

[**CHƯƠNG III . XÂY DỰNG BÀI TOÁN NHẬN DẠNG BÀN TAY ĐIỀU KHIỂN ROBOT.** 19](#_Toc136449501)

[**3.1. Mô tả bài toán.** 19](#_Toc136449502)

[**3.2. Thiêt kế chức năng nhận dạng và xử lý .** 20](#_Toc136449503)

[**3.2.1.Sơ đồ nhận dạng bàn tay.** 20](#_Toc136449504)

[**3.2.2. Chương trình.** 20](#_Toc136449505)

[**3.3. Thiết kế mô hình truyền dữ liệu .** 23](#_Toc136449506)

[**3.3.1 . Linh kiện sử dụng .** 24](#_Toc136449507)

[**3.3.2 . Chương trình .** 25](#_Toc136449508)

[**3.4. Thiết kế mô hình nhận dữ liệu và chấp hành lệnh .** 27](#_Toc136449509)

[**3.4.1. Linh kiện sử dụng .** 27](#_Toc136449510)

[**3.4.2. Chương trình .** 29](#_Toc136449511)

[**CHƯƠNG IV. THỬ NGHIỆM VÀ KẾT QUẢ .** 31](#_Toc136449512)

[**4.1 . khối xử lý ảnh và truyền tín hiệu** 31](#_Toc136449513)

[**4.2. Khối Phát tín hiệu .** 32](#_Toc136449514)

[**4.3. Mô hình nhận tín hiệu và chấp hành lệnh .** 32](#_Toc136449515)

[**KẾT LUẬN VÀ KIẾN NGHỊ .** 33](#_Toc136449516)

[**Kết luận** 33](#_Toc136449517)

[**Kiến nghị** 34](#_Toc136449518)

[**Tải liệu tham khảo** 35](#_Toc136449519)

# **Danh mục những từ viết tắt**

|  |  |
| --- | --- |
| ML | Machine Learning |
| OPENCV | Open source computer vision library |
| HSV | Hue Saturation Value |
| HSB | Hue Saturation Bright |
| RGB | Red, Green, Blue |
| AI | Artificial Intelligence . |
| I/O | Input/Output |
| IDE | Integrated Development Environment |
| FTDI | Future Technology Devices International |
| PHP | Hypertext Preprocessor |
| UART | Universal Asynchronous Receiver / Transmitter |
| I2C | Inter – Integrated Circuit |
| SPI | Serial Peripheral Interface |
| SCK | Serial clock |
| MISO | Master Input Slave Output |
| MOSI | Master Output Slave Input |
| SS | Slave Select |
| CS | Chip Select |
| GPIO | General Purpose Input Output |
| WC | Website camera |
| MP | Megapixel |
| PWM | Pulse Width Modulation |

# **Danh mục bảng biểu**

[Hình 1 Một số hàm thông dụng 8](#_Toc136449933)

[Hình 2 Sơ đồ khối nhận dạng đối tượng . 9](#_Toc136449934)

[Hình 3 Mô hình xử lý tromg MediaPipe 11](#_Toc136449935)

[Hình 4 Giải pháp 12](#_Toc136449936)

[Hình 5 Mediapipe model maker 12](#_Toc136449937)

[Hình 6 Điểm trên bàn tay 13](#_Toc136449938)

[Hình 7 Mô hình SPI 14](#_Toc136449939)

[Hình 8 Chu trình xử lý ảnh. 15](#_Toc136449940)

[Hình 9 Lưu đồ thuật toán 18](#_Toc136449941)

[Hình 10 Mô tả bài toán 18](#_Toc136449942)

[Hình 11 Sơ đồ. 19](#_Toc136449943)

[Hình 12 Thực hiện đào tạo ở model maker. 19](#_Toc136449944)

[Hình 13 Ví dụ về kết quả thu được trên bàn tay 22](#_Toc136449945)

[Hình 14 Mô hình truyền dữ liệu 23](#_Toc136449946)

[Hình 15 Arduino UNO 23](#_Toc136449947)

[Hình 16 Module NRF24L01 24](#_Toc136449948)

[Hình 17 Led 24](#_Toc136449949)

[Hình 18 Điện trở 24](#_Toc136449950)

[Hình 19 Mô hình nhận dữ liệu 26](#_Toc136449951)

[Hình 20 Module L298N 27](#_Toc136449952)

[Hình 21 Cấu tạo chân L298N 28](#_Toc136449953)

[Hình 22 Động cơ giảm tốc 28](#_Toc136449954)

[Hình 23 Khối xử ký ảnh và truyền tín hiệu 30](#_Toc136449955)

[Hình 24 Khối phát tín hiệu 31](#_Toc136449956)

[Hình 25 Khối thu tín hiệu chấp hành lệnh 31](#_Toc136449957)

# **MỞ ĐẦU**

## **Đặt vấn đề .**

Cùng với sự phát triển của thị giác máy tính và kỹ thuật học sâu, phân tích nội dung hình ảnh/video ngày càng thu hút được sự chú ý của cộng đồng nghiên cứu. Xã hội con người ngày càng phát triển theo đó công nghệ cũng tiến bộ theo từng ngày. Kéo theo các hệ thống tự động hóa sử dụng công nghệ nhận diện đối tượng ngày càng nhiều và được ứng dụng trên mọi lĩnh vực trong đời sống đặc biệt là trong điều khiển Robot .Với phương pháp này, chúng ta có thể thu nhận được nhiều thông tin từ đối tượng mà không cần tác động nhiều đến đối tượng nghiên cứu. Sự phát triển của khoa học máy tính tạo môi trường thuận lợi cho bài toán nhận dạng đối tượng từ video, hình ảnh. Các hệ thống nhận dạng đã ra đời và có độ tin cậy cao đáp ứng được các nhu cầu của nền công nghiệp 4.0 và đang tiến tới nền công nghiệp 5.0.

## **2. Lý do chọn đề tài .**

Hiện nay việc nhận diện đối tượng đã trở nên cần thiết đối với nhiều ngành nghề, lĩnh vực khác nhau. Trong đó ta có thể áp dụng xử lý nhận diện đối tượng trong việc điều khiển Robot để qua đó có thể đưa vào các hệ thống tự động trong công nghiệp, nông nghiệp, y tế, giao thông,.... Vì vậy nhóm nghiên cứu chúng em đã lựa chọn đề tài:

**“Ứng dụng xử lý ảnh nhận dạng cử chỉ bàn tay điều khiển robot chuyển động**”

## **3. Mục tiêu nghiên cứu .**

* Xây dựng chương trình nhân diện bàn tay người, gửi dữ liệu qua vi xử lý và điều khiển cơ cấu chấp hành .
* Nhiệm vụ của mục tiêu như sau :

+ Tìm hiểu bài toán nhận dạng đối tượng .

+ Xây dụng chương trình nhận diện bàn tay .

+ Thiết kế mô hình truyền nhận dữ liệu và điều khiển Robot.

## **4. Đối tượng nghiên cứu**

* Bài toán nhận dạng đối tượng.
* Nhận dạng cử chỉ bàn tay để điều khiển hoạt động của Robot .

## **5. Phạm vi nghiên cứu của đề tài .**

* Tập trung vào 1 số hoạt động của robot tương ứng với cử chỉ của tay .
* Các cử chỉ này có thể điều khiển các hoạt động bao gồm: di chuyển, dừng lại, quay trái, quay phải, tăng tốc, giảm tốc, các thao tác khác.

# **CHƯƠNG I . CƠ SỞ LÝ THUYẾT.**

## **Tổng quan về ngôn ngữ Python.**

* + 1. **Python là gì?**

Python là ngôn ngữ lập trình máy tính thường được sử dụng để xây dựng trang web và phần mềm, tự động hóa các tác vụ và tiến hành phân tích dữ liệu. Python là ngôn ngữ có cú pháp đơn giản giúp xây dựng các dự án nhanh và cải thiện chúng nhanh hơn. Đặc biệt, python có một cộng đồng lớn và tích cực có nguồn tài nguyên cực kì lớn . Bởi tính linh hoạt này, đã khiến nó trở thành một trong những ngôn ngữ lập trình được sử dụng nhiều.

* + 1. **Ứng dụng của Python**
* Ngày nay Python được sử dụng với các mục đích sau:
* Lập trình ứng dụng web: Bạn có thể tạo web app có khả năng mở rộng (scalable) được bằng cách sử dụng framework và CMS (Hệ thống quản trị nội dung) được tích hợp trong Python.
* Khoa học và tính toán: Có nhiều thư viện trong Python cho khoa học và tính toán số liệu, như SciPy và NumPy, được sử dụng cho những mục đích chung chung trong tính toán. Và, có những thư viện cụ thể như: EarthPy cho khoa học trái đất, AstroPy cho Thiên văn học,... Ngoài ra, Python còn được sử dụng nhiều trong machine learning, khai thác dữ liệu và deep learning.
* Tạo nguyên mẫu phần mềm: Python chậm hơn khi so sánh với các ngôn ngữ được biên dịch như C++ và Java. Nó có thể không phải là lựa chọn tốt nếu nguồn lực bị giới hạn và yêu cầu về hiệu quả là bắt buộc.
* Ngôn ngữ tốt để dạy lập trình: Python được nhiều công ty, trường học sử dụng để dạy lập trình cho trẻ em và những người mới lần đầu học lập trình. Bên cạnh những tính năng và khả năng tuyệt vời thì cú pháp đơn giản và dễ sử dụng của nó là lý do chính cho việc này.
  + 1. **Truyền thông nối tiếp ở Python.**
* PySerial (Giao tiếp nối tiếp Python) là một công cụ mạnh mẽ để các nhà phát triển tạo ra các ứng dụng tương tác với phần cứng . Nó là một thư viện cho phép các nhà phát triển truy cập và kiểm soát các cổng nối tiếp từ bên trong các ứng dụng Python của họ . với nó ta có thể dễ dàng tạo ra các ứng dụng có thể giao tiếp với các thiết bị phần cứng như vi điều khiển và các hệ thống nhúng khác .
* Giao tiếp nối tiếp Python hoạt động bằng cách truy cập và kiểm soát các cổng nối tiếp từ bên trong ứng dụng .
* Để giao tiếp nối tiếp python ta cần có
* Môi trường để lập trình như : Python , Arduino IDE , Visual studio code ,….
* Thư viện : PySerial .
* Một số hàm được sử dụng trong thư viện :

|  |  |  |
| --- | --- | --- |
| **Tên Hàm** | **Tác dụng** | **Ghi chú** |
| Import serial | Thêm thư viện serial |  |
| Ser=Serial.serial(‘COM1’,9600) | Tạo đối tượng kết nối | COM1 là cổng kết nối  9600 là tốc độ baudrate |
| Ser.open() | Mở đối tượng |  |
| Ser.isOpen() | Mở đối tượng |  |
| Ser.write() | Ghi dữ liệu |  |
| Ser.read() | Đọc dữ liệu |  |
| Ser.close() | Đóng kết nối |  |

Hình Một số hàm thông dụng

## **Tổng quan về nhận dạng đối tượng.**

* + 1. **Nhận dạng đối tượng là gì ?**
* Nhận dạng đối tượng (Object Detection) được coi là một trong những lĩnh vực quan trọng nhất trong phát triển của học sâu (Deep Learning) và xử lý hình ảnh (Image Processing). Các mô hình phát hiện đối tượng thường được đào tạo để phát hiện sự hiện diện của các đối tượng cụ thể trong hình ảnh, video.
* Phương pháp được đề xuất cho bài toán phát hiện đối tượng. Đa phần các kỹ thuật phát hiện đối tượng được phân loại theo hai hướng tiếp cận chính: dựa vào các đặc trưng thiết kế (hand-designed features) và dựa vào các đặc trưng học sâu (Deep- Learned Features).

**Ảnh có chứa văn bản, Phông chữ, Giấy nhớ, thiết kế

Mô tả được tạo tự động**

Hình :Sơ đồ khối nhận dạng đối tượng .

* + - Tiền xử lý dữ liệu (Pre-Processing): Cần phải xử lý trước để làm sạch dữ liệu hình ảnh cho đầu vào của mô hình. Tiền xử lý hình ảnh cũng có thể làm giảm thời gian huấn luyện mô hình và tăng tốc độ suy luận mô hình.
* Phát hiện đối tượng (Object detection): Dự đoán các vùng không gian chứa đối tượng và phân lớp.
  + 1. **Ứng dụng của nhận dạng đối tượng .**
* Một số ứng dụng chính của công nghệ nhận dạng đối tượng :

+ Nhận dạng biển số – sử dụng cả công nghệ phát hiện đối tượng và nhận dạng ký tự quang học để nhận dạng các ký tự chữ và số trên biển số xe. Tính năng phát hiện đối tượng được sử dụng thông qua việc lưu giữ lại hình ảnh và phát hiện các đối tượng cụ thể như xe cộ, phương tiện đi lại trên bức ảnh đó. Sau khi đã xác định được tọa độ của phương tiện trên ảnh, chúng ta có thể sử dụng các mô hình Object Detection khác để phát hiện biển số xe và các ký tự trên biển số. Từ đó, áp dụng các mô hình OCR hoặc image classification để nhận dạng ký tự trên hình, ánh xạ từ dạng dữ liệu hình ảnh sang dạng chữ (text).

+ Phát hiện và nhận dạng khuôn mặt – một trong những ứng dụng chính của phát hiện đối tượng là nhận diện và phát hiện khuôn mặt. Với sự trợ giúp của các thuật toán hiện đại, chúng ta có thể phát hiện khuôn mặt người trong một hình ảnh hoặc video.

+ Theo dõi đối tượng - Công nghệ này có thể ứng dụng để theo dõi chuyển động của một đối tượng hay đồ vật cụ thể. Chẳng hạn, trong khi xem một trận bóng chày hoặc cricket, quả bóng có thể bị đập ra xa. Trong những tình huống này, thuật toán giúp chúng ta phát hiện, nhận dạng và theo dõi (tracking) vị trí và đường bay của quả bóng hiện tại.

+ Ô tô tự lái - đối với ô tô tự lái, điều quan trọng là phải nghiên cứu các yếu tố môi trường xung quanh ô tô khi lái xe. Một mô hình phát hiện đối tượng được đào tạo trên nhiều đối tượng để phát hiện các thực thể khác nhau cản trở việc lái xe nhằm đảo bảo an toàn cho người lái.

+ Người máy – Hiện nay thuật toán phát hiện đối tượng thường được ứng dụng trong robot để nhận diện và di chuyển các vật nặng, giảm thiểu những công việc về thể chất cho con người và tự động hóa công việc.

## **Tổng quan về mediapipe.**

* + 1. **Mediapipe là gì?**
* Mediapipe là một khung mã nguồn mở để “xây dựng các giải pháp máy học” của Google. Nó đã được mở và có thể sẽ được phát triển lâu hơn nữa. MediaPipe là một cross-platform để xây dựng các đường các mô hình trên nhiều nền tảng khác nhau, có thể áp dụng cho nhiều cách thức xử lý như: video, hình ảnh,...
* Các mô hình ML models nổi bật: Face detection, Face mesh, Multi-hand Tracking, Object detection – tracking, AutoFlip (Automatic video cropping pipeline).
* MediaPipe Solutions cung cấp một bộ thư viện và công cụ để bạn nhanh chóng áp dụng các kỹ thuật trí tuệ nhân tạo (AI) và máy học (ML) trong các ứng dụng của mình. Bạn có thể sử dụng các giải pháp này vào ứng dụng của mình ngay lập tức, tùy chỉnh chúng theo nhu cầu của bạn và sử dụng chúng trên nhiều nền tảng phát triển. Vì vậy bạn có thể tùy chỉnh thêm mã giải pháp để đáp ứng nhu cầu ứng dụng của mình. Bộ giải pháp MediaPipe bao gồm những điều sau đây:

Ảnh có chứa văn bản, ảnh chụp màn hình, biểu đồ, hàng

Mô tả được tạo tự động

Hình Mô hình xử lý tromg MediaPipe

* Các thư viện và tài nguyên này cung cấp chức năng cốt lõi cho từng Giải pháp MediaPipe:

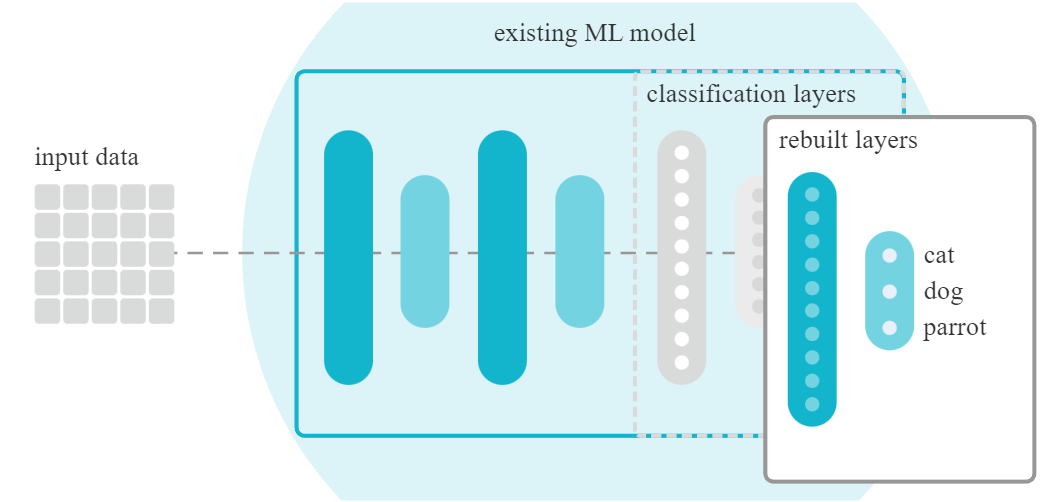
**+ MediaPipe Tasks** : cung cấp giao diện lập trình cốt lõi của bộ Giải pháp MediaPipe, bao gồm một bộ thư viện để triển khai các giải pháp ML đổi mới trên các thiết bị. Nó hỗ trợ nhiều nền tảng, bao gồm Android, Web / JavaScript, Python và hỗ trợ cho nền tảng IOS.

**+ MediaPipe models** : Các mô hình được đào tạo trước, sẵn sàng chạy để sử dụng với từng giải pháp cụ thể sau :

|  |  |  |  |
| --- | --- | --- | --- |
| **Giải pháp** | **Android** | **Web** | **Python** |
| Phát hiện đối tượng | ✓ | ✓ | ✓ |
| Phân đoạn hình ảnh | ✓ | ✓ | ✓ |
| Phát hiện mốc bàn tay | ✓ | ✓ | ✓ |
| Nhận dạng cử chỉ | ✓ | ✓ | ✓ |
| Nhúng hình ảnh | ✓ | ✓ | ✓ |
| Phát hiện khuân mặt | ✓ | ✓ | ✓ |
| Phân loại văn bản | ✓ | ✓ | ✓ |
| Nhúng văn bản | ✓ | ✓ | ✓ |
| Phát hiện ngôn ngữ | ✓ | ✓ | ✓ |
| Phân loại âm thanh | ✓ | ✓ | ✓ |

Hình Giải pháp

* **MediaPipe Model Maker** : là một công cụ để tùy chỉnh các mô hình máy học (ML) hiện có để hoạt động với dữ liệu và ứng dụng của bạn. Bạn có thể sử dụng công cụ này như một giải pháp thay thế nhanh hơn để xây dựng và huấn luyện một mô hình ML mới .



Hình Mediapipe model maker

**+ MediaPipe Studio** : là một ứng dụng dựa trên web để đánh giá và tùy chỉnh các mô hình và quy trình ML trên thiết bị cho các ứng dụng của bạn. Ứng dụng cho phép bạn nhanh chóng kiểm tra các giải pháp MediaPipe trong trình duyệt của mình bằng dữ liệu của riêng bạn và các mô hình ML tùy chỉnh của riêng bạn.

### **MediaPipe Hand Landmarker.**

Tác vụ MediaPipe Hand Landmarker cho phép bạn phát hiện các mốc của bàn tay trong một hình ảnh. Bạn có thể sử dụng Tác vụ này để đánh dấu vị trí các điểm chính của bàn tay và hiển thị hiệu ứng hình ảnh trên bàn tay. Tác vụ này hoạt động trên dữ liệu hình ảnh với mô hình máy học (ML) dưới dạng dữ liệu tĩnh hoặc luồng liên tục và xuất ra các điểm của bàn tay trong tọa độ hình ảnh. Mô hình bàn tay tổng hợp chất lượng cao cũng được hiển thị trên nhiều nền khác nhau và ánh xạ nó tới các tọa độ 3D tương ứng.

Ảnh có chứa biểu đồ, hàng

Mô tả được tạo tự động

Hình Điểm trên bàn tay

* 1. **Tổng quan giao thức SPI.**
* SPI – Serial Peripheral Interface hay còn gọi là giao diện ngoại vi nối tiếp . Là giao tiếp đồng bộ vơi chế độ truyền song công toàn phần ( full duplex ) cho phép tại một thời điểm có thể truyền và nhận .
* SPI thường được sử dụng để giao tiếp với các loại cảm biến , bộ nhớ EPROM , đồng hồ thời gian thực và các vi điều khiển khác.

Ảnh có chứa văn bản, Phông chữ, ảnh chụp màn hình, hàng

Mô tả được tạo tự động

Hình Mô hình SPI

* Hoạt động của giao tiếp SPI sử dụng 4 đường dây giao tiếp :

+ SCK(Serial clock ) : Thiết bị Master tạo xung tín hiệu SCK và cung cấp cho Slave. Xung này có chức năng giữ nhịp cho giao tiếp SPI. Mỗi nhịp trên chân SCK báo 1 bit dữ liệu đến hoặc đi.

+ MISO (Master Input Slave Output): Tín hiệu tạo bởi thiết bị Slave và nhận bởi thiết bị Master. Đường MISO phải được kết nối giữa thiết bị Master và Slave.

+ MOSI (Master Output Slave Input): Tín hiệu tạo bởi thiết bị Master và nhận bởi thiết bị Slave. Đường MOSI phải được kết nối giữa thiết bị Master và Slave.

+ SS (Slave Select): Chọn thiết bị Slave cụ thể để giao tiếp. Để chọn Slave giao tiếp thiết bị Master chủ động kéo đường SS tương ứng xuống mức 0 (Low). Chân này đôi khi còn được gọi là CS (Chip Select). Chân SS của vi điều khiển (Master) có thể được người dùng tạo bằng cách cấu hình 1 chân GPIO bất kỳ chế độ Output.

# **CHƯƠNG II . TỔNG QUAN BÀI TOÁN NHẬN DẠNG BÀN TAY.**

## **2.1. Những vấn đề cơ bản trong xử lý nhận diện bàn tay .**

### **2.1.1. Xử lý ảnh đầu vào.**

Xử lý ảnh đầu vào là quá trình cần thiết để chuẩn bị một bức ảnh cho việc phân tích hoặc điều khiển bởi chương trình máy tính. Trước khi một hệ thống máy học có thể sử dụng 1 bức ảnh để thực hiện các nhiệm vụ như nhận dạng hình ảnh, thì ảnh đầu vào cần được tiền sử lý để dễ dàng cho thuật toán nhận diện và xác định .

Quá trình xử lý ảnh được mô tả như sau :

Ảnh có chứa biểu đồ, hàng, văn bản, Kế hoạch

Mô tả được tạo tự động

Hình . Chu trình xử lý ảnh.

* Thu nhận ảnh (Image acquisition): Ảnh có thể nhận qua camera màu hoặc đen trắng. Thường ảnh nhận qua camera là ảnh tương tự, cũng có loại camera số hòa là loại Photodiot tạo cường độ sáng tại mỗi điểm ảnh. Camera thường dùng là loại quét dòng, ảnh tạo ra có dạng hai chiều. Chất lượng một ảnh thu nhận được phụ thuộc vào thiết bị thu, môi trường (ánh sáng, phong cảnh).
* Tiền xử lý (Image processing): Sau khi thu nhận ảnh, ảnh có thể bị nhiễu, hoặc có độ tương phản thấp nên cần đưa vào bộ tiền xử lý ảnh để nâng cao chất lượng. Chức năng chính của bộ tiền xử lý là lọc nhiễu và năng cao độ tương phản làm cho ảnh rõ hơn và nét hơn.
* Biển diễn ảnh (Image representation): Đầu ra của ảnh 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 thành dạng thích hợp là cần thiết cho xử lý tiếp theo bằng máy tính. Việc chọn các tính chất để thể hiện 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 các đối tượng khác trong phạm vi ảnh nhận được.
* Nhận dạng và nội suy ảnh: Là quá trình xác định ảnh. Quá trình này thường thu được bằng cách so sánh với mẫu chuẩn đã được học và nội suy, phán đoán theo ý nghĩa dựa trên cơ sở nhận dạng. Có nhiều cách phân loại ảnh khác nhau.
* 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 nhiễu. 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 con người nhận thức.

### **2.1.2. Đánh dấu các điểm trên bàn tay.**

Mô hình mốc bàn tay phát hiện và đánh dấu các điểm chính của 21 tọa độ khớp ngón tay trong các vùng bàn tay được phát hiện. Mô hình được đào tạo trên khoảng 30 nghìn hình ảnh trong thế giới thực, cũng như một số mô hình bàn tay tổng hợp được dựa trên các nền khác nhau.

## **2.2. Bài toán nhận dạng bàn tay .**

### **2.2.1 . Chi tiết nhiệm vụ .**

* Đầu vào của nhiệm vụ :

+ Ảnh tĩnh.

+ Video.

+ Video trực tiếp .

* Đầu ra của nhiệm vụ :

+ Bàn tay được phát hiện .

+ Các điểm đánh dấu của 21 điểm trên bàn tay .

### **2.2.2 . Thực hiện nhiệm vụ .**

* Sử dụng môi trường PyCharm để thực hiện và chạy chương trình .
* Sử dụng các thư viện OpenCV , mediapipe để lập trình xử lý nhận dạng bàn tay .
* Sử dụng camera laptop để thu nhận hình ảnh .
* Các bước xây dựng chương trình phát hiện và đánh điểm trên bàn tay .

+Bước 1 : Lấy dữ liệu từ camera

+Bước 2 : Chuyển đổi không gian màu để phù hợp với định dạng của mediapipe.

+Bước 3 : Tạo biến và đối tượng để sử dụng các tính năng của mediapipe.

+Bước 4 : Đánh các điểm trên bàn tay sử dụng giải pháp MediaPipe Hand Landmarks.

+Bước 5 : So sánh tọa độ các điểm đã được đánh dấu để xác đinh số ngón tay.

+Bước 6 : Đưa kết quả sang nền tảng khác để xử lý thông tin.

Ảnh có chứa văn bản, hàng, ảnh chụp màn hình, biểu đồ

Mô tả được tạo tự động

Hình Lưu đồ thuật toán

# **CHƯƠNG III . XÂY DỰNG BÀI TOÁN NHẬN DẠNG BÀN TAY ĐIỀU KHIỂN ROBOT.**

## **3.1. Mô tả bài toán.**

Ảnh có chứa văn bản, ảnh chụp màn hình, biểu đồ, Hình chữ nhật

Mô tả được tạo tự động

Hình Mô tả bài toán

Trong mô hình của chúng ta, video sẽ thu được qua camera của laptop , sau đó được đưa vào môi trường Pycharm để thực hiện việc nhận dạng và đếm số lượng ngón tay. Tiếp theo, dữ liệu thu được từ xử lý nhận diện sẽ được truyền qua Arduino (khối phát) thông qua giao thức truyền thông nối tiếp UART và kết hợp với module NRF24L01 . Bên Arduino (Khối thu) được kết nối với module NRF24L01 sẽ tiếp nhận tính hiệu của khối phát . Cuối cùng sẽ thực hiện điều khiển Robot thông qua mạch điều khiển động cơ .

## **3.2. Thiêt kế chức năng nhận dạng và xử lý .**

### **3.2.1.Sơ đồ nhận dạng bàn tay.**

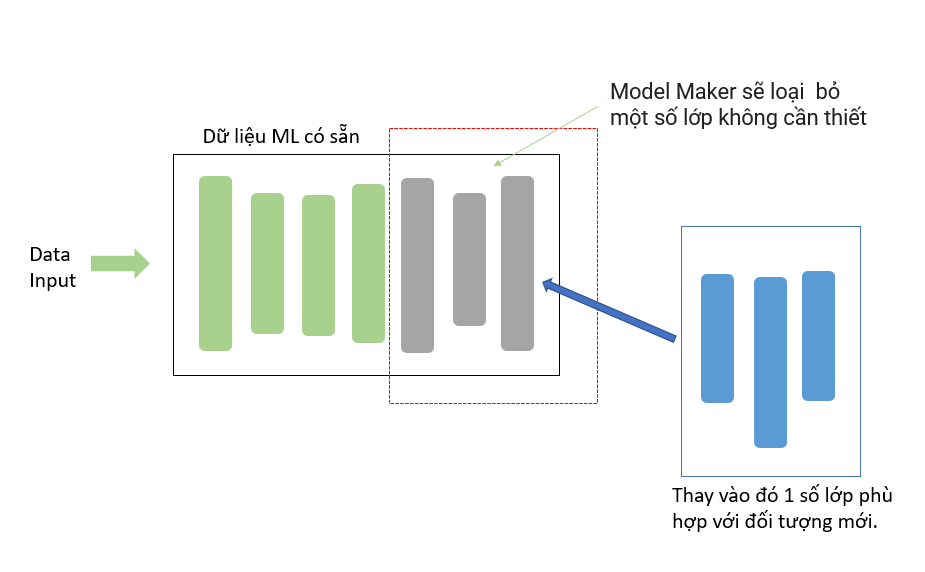
Ảnh có chứa văn bản, biểu đồ, hàng, Phông chữ

Mô tả được tạo tự động

Hình : Sơ đồ.

Khối chuyển đổi hình ảnh thành các điểm thông qua mediapipe dựa trên mô hình **MediaPipe Model Maker** (hình 3) mô hình này sẽ thu thập các dữ liệu của các mô hình ML cũ và dựa vào đó để đào tạo các mô hình mới. Kỹ thuật này sử dụng lại một phần đáng kể logic mô hình hiện có, nghĩa là việc đào tạo mất ít thời gian hơn so với đào tạo một mô hình mới và có thể được thực hiện với ít dữ liệu hơn.

**MediaPipe Model Maker** thực hiện đào tạo như sau :



Hình . Thực hiện đào tạo ở model maker.

### **3.2.2. Chương trình.**

* Bước 1 : Lấy dữ liệu từ camera .

*cap = cv2.VideoCapture(0)*

+ Đối tượng cap được khởi tạo để lấy dữ liệu từ camera . Biến 0 được truyền vào để chỉ định rằng chương trình sẽ sử dụng camera được kết nối mới máy tính.

* Bước 2 : Chuyển đổi không gian màu để phù hợp với định dạng của mediapipe.

*img = cv2.cvtColor(img, cv2.COLOR\_BGR2RGB)*

*result = hands.process(img)*

*img = cv2.cvtColor(img, cv2.COLOR\_RGB2BGR)*

* Bước 3 : Tạo biến và đối tượng để sử dụng các tính năng của mediapipe.

+ Trước khi bắt đầu nhận dạng bàn tay, chương trình cần khởi tạo một số biến và đối tượng để sử dụng các tính năng của Mediapipe .

*mp\_drawing\_util = mp.solutions.drawing\_utils*

*mp\_drawing\_style = mp.solutions.drawing\_styles*

+ Được sử dụng để vẽ các điểm đỉnh trên bàn tay trên hình ảnh đầu ra của chương trình.

*mp\_hand = mp.solutions.hands*

+Chứa các phương thức để xử lý bàn tay, trong đó hands là một đối tượng được khởi tạo để nhận diện và theo dõi các điểm đỉnh trên bàn tay.

*hands = mp\_hand.Hands*

*(*

*model\_complexity=0,*

*min\_detection\_confidence=0.5,*

*min\_tracking\_confidence=0.5*

*)*

*+ model\_complexity :* sử dụng mô hình đơn giản nhất để giảm thiểu tài nguyên tính toán.

*+ min\_detection\_confidence :* ngưỡng nhận dạng tối thiểu cho việc phát hiện bàn tay.

*+ min\_tracking\_confidence :*  là ngưỡng tối thiểu cho việc theo dõi các điểm đỉnh trên bàn tay.

* Bước 4 : Đánh các điểm trên bàn tay sử dụng giải pháp MediaPipe Hand Landmarks.

*myHand = []*

*count = 0*

*for idx, hand in enumerate(result.multi\_hand\_landmarks):*

*for id, lm in enumerate(hand.landmark):*

*h, w, \_ = img.shape*

*myHand.append([int(lm.x \* w), int(lm.y \* h)])*

* Bước 5 : So sánh tọa độ các điểm đã được đánh dấu để xác đinh số ngón tay và đưa lên màn hình hiển thị .

+So sánh :

*if myHand[8][1] < myHand[6][1]:  
 count = count + 1  
if myHand[12][1] < myHand[10][1]:  
 count = count + 1  
  
if myHand[16][1] < myHand[14][1]:  
 count = count + 1  
  
if myHand[20][1] < myHand[19][1]:  
 count = count + 1  
  
if myHand[4][0] > myHand[2][0]:  
 count = count + 1*

+Hiển thị lên màn hình :

*cv2.putText(img, str(count), (50, 50), cv2.FONT\_HERSHEY\_SIMPLEX, 1, (0, 0, 255), 2, cv2.LINE\_AA)*

*cv2.imshow("Nhan dang ban tay", img)*

* Bước 6 : Đưa kết quả sang nền tảng khác để xử lý thông tin.

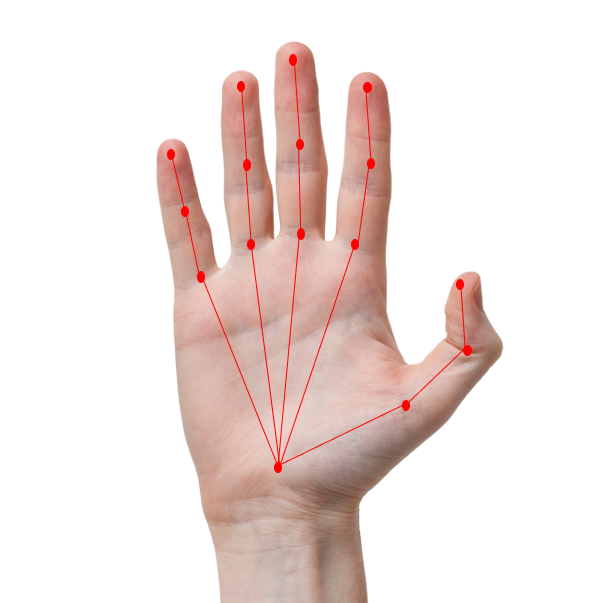
+ Khởi tạo thư viện kết nối serial.

*import serial*

+ Gửi dữ liệu qua cổng Arduino qua truyền thông nối tiếp

if ser.isOpen():  
ser.write(str(count).encode())

**3.2.2. Kết quả .**



Hình Ví dụ về kết quả thu được trên bàn tay

## **3.3. Thiết kế mô hình truyền dữ liệu .**

Ảnh có chứa văn bản, biểu đồ, Kế hoạch, Bản vẽ kỹ thuật

Mô tả được tạo tự động

Hình Mô hình truyền dữ liệu

### **3.3.1 . Linh kiện sử dụng .**

* Arduino.

Arduino đóng vai trò là vi xử lý chính .

Ảnh có chứa đồ điện tử, Kỹ thuật điện, Linh kiện điện, Thành phần mạch điện

Mô tả được tạo tự động

Hình Arduino UNO

* Module NRF24L01.

Ảnh có chứa đồ điện tử, Kỹ thuật điện, Linh kiện điện, Thành phần mạch điện

Mô tả được tạo tự động

Hình Module NRF24L01

+ Module NRF24L01 Có thể truyền và nhận dữ liệu bằng một module duy nhất. Thu phát sóng tín hiệu không phải là khả năng duy nhất, module có thể giao tiếp với tổng cộng 6 module NRF24L01 khác cùng một lúc. Thiết bị có thể giao tiếp với Arduino tạo ra nhiều ứng dụng điều khiển từ xa .

+ Trong mô hình module được kết nói với arduino ( bên phát và bên thu ) thông qua giao thức SPI để truyền lệnh từ bên phát đến bên thu .

* Led.

Ảnh có chứa công cụ

Mô tả được tạo tự động

Hình Led

* Điện trở .



Hình Điện trở

### **3.3.2 . Chương trình .**

* Được thực hiện trên môi trường Arduino IDE.
* Thư viện sử dụng SPI.h , RF24.h
* Một số hàm chính .

+Sử dụng chân 9 , 10 cho chân CE , CSN

*RF24 radio(9, 10);*

+Khởi tạo

*radio.setPALevel(RF24\_PA\_MIN);*

*radio.openWritingPipe(0xE8E8F0F0E1LL);*

*radio.stopListening();*

+Kiêm tra có dữ liệu tới không

Serial.available()

+ Truyền dữ liệu mong muốn

*radio.write(&numberToSend, sizeof(numberToSend));*

* Chương trình chính :

*void loop() {*

*int numberToSend ;*

*if (Serial.available() > 0)*

*{*

*val = Serial.read() - '0';*

*if (val > 0 && val < 6 ) {*

*if(val==1){*

*numberToSend = 1 ;*

*}*

*else if(val==2){*

*numberToSend = 2 ;*

*}*

*else if(val==3){*

*numberToSend = 3 ;*

*}*

*else if(val==4){*

*numberToSend = 4 ;*

*digitalWrite(led4, HIGH); // Bật LED*

*}*

*else if(val==5){*

*numberToSend = 5 ;*

*digitalWrite(led5, HIGH); // Bật LED*

*}*

*} else {*

*numberToSend = 0;*

*}*

*radio.write(&numberToSend, sizeof(numberToSend));*

*Serial.println(numberToSend);*

*}*

*}*

## **3.4. Thiết kế mô hình nhận dữ liệu và chấp hành lệnh .**

Ảnh có chứa biểu đồ, Kế hoạch, văn bản, Bản vẽ kỹ thuật

Mô tả được tạo tự động

Hình Mô hình nhận dữ liệu

### **3.4.1. Linh kiện sử dụng .**

* Arduino .(Hình ).
* Module NRF24L01.(Hình ).
* Module điều khiển động cơ L298N.

Ảnh có chứa Thành phần mạch điện, đồ điện tử, Linh kiện điện, Cấu phần mạch bị động

Mô tả được tạo tự động

Hình Module L298N

+ L298N là module điều khiển động cơ trong các xe DC và động cơ bước. Module có một IC điều khiển động cơ L298 và một bộ điều chỉnh điện áp 5V 78M05. Module L298N có thể điều khiển tối đa 4 động cơ DC hoặc 2 động cơ DC với khả năng điều khiển hướng và tốc độ.

+ Các linh kiện khác: IC điều khiển động cơ LM298, Bộ điều chỉnh điện áp 78M05, Tụ điện, Điện trở, Tản nhiệt

+ Cấu tạo các chân của L298N :

|  |  |
| --- | --- |
| Tên chân | Mô tả |
| IN1 & IN2 | Các chân đầu vào điều khiển hướng quay động cơ A |
| IN3 & IN4 | Các chân đầu vào điều khiển hướng quay của Động cơ B |
| ENA | Kích hoạt tín hiệu PWM cho Động cơ A |
| ENB | Kích hoạt tín hiệu PWM cho Động cơ B |
| OUT1 & OUT2 | Chân đầu ra cho động cơ A |
| OUT3 & OUT4 | Chân đầu ra cho Động cơ B |
| 12V | Đầu vào cấp nguồn 12 |
| 5V | Cấp nguồn cho mạch logic bên trong IC L298N |
| GND | Chân nối đất |

Hình Cấu tạo chân L298N

* Động cơ giảm tốc.

Ảnh có chứa nhựa, công cụ, màu vàng

Mô tả được tạo tự động

Hình Động cơ giảm tốc

Động cơ DC giảm tốc vàng là mẫu động cơ được sử dụng rộng rãi trong thiết kế robot, mô hình DIY, phục vụ việc nghiên cứu học tập chế tạo, với dải điện áp từ 3-9v.

* Một số linh kiện khác .

+ Công tắc.

+ Pin 18650.

+ Khay Pin .

+ Dây cắm .

+ Board cắm .

### **3.4.2. Chương trình .**

* Được thực hiện trên môi trường Arduino IDE.
* Thư viện sử dụng SPI.h , RF24.h .
* Một số hàm chính

+ Sử dụng chân 9 , 10 cho chân CE , CSN

*RF24 radio(9, 10);*

*+* Khởi tạo

*radio.setPALevel(RF24\_PA\_MIN);*

*radio.openReadingPipe(0,0xE8E8F0F0E1LL);*

*radio.startListening();*

+ Nhận dữ liệu

*radio.read(&numberReceived, sizeof(numberReceived));*

+ Chương trình chính bao gồm các hàm điều khiển 4 động cơ thông qua module L298N .

# **CHƯƠNG IV. THỬ NGHIỆM VÀ KẾT QUẢ .**

## **4.1 . khối xử lý ảnh và truyền tín hiệu**

Ảnh có chứa đồ điện tử, máy tính, văn bản, máy tính xách tay

Mô tả được tạo tự động

Hình Khối xử ký ảnh và truyền tín hiệu

## **4.2. Khối Phát tín hiệu .**

Ảnh có chứa Thành phần mạch điện, Linh kiện điện, Kỹ thuật điện, Cấu phần mạch bị động

Mô tả được tạo tự động

Hình Khối phát tín hiệu

## **4.3. Mô hình nhận tín hiệu và chấp hành lệnh .**

Ảnh có chứa đồ chơi, Phụ tùng ô tô

Mô tả được tạo tự động

Hình Khối thu tín hiệu chấp hành lệnh

# **KẾT LUẬN VÀ KIẾN NGHỊ .**

**Kết luận**

Trong nghiên cứu này, chúng em đã tìm hiểu về ứng dụng xử lý ảnh trong việc điều khiển xe chuyển động. Bằng cách áp dụng các thuật toán xử lý ảnh kết hợp với các cách thức và giao thức truyền thông dữ liệu và sử dụng vi xử lý để điều khiển tín hiệu cho 1 cơ cấu chấp hành .

Trước tiên, chúng em đã thu thập hình ảnh từ camera của laptop . Dữ liệu này được sử dụng cho việc xây dựng một mô hình học máy để xác định cử chỉ của bàn tay trong ảnh và đếm được số ngón tay .

Sau đó, chúng em đã xây dựng một chương trình để gửi tín hiệu sang vi xử lý thông qua giao thức UART . Vi xử lý bên phát tín hiệu sẽ được kết nối với module truyền không dây NRF24L01 và thông qua giao thức SPI truyền tới vi xử lý bên nhận . Cuối cùng vi xử lý bên nhận sẽ thông qua tín hiệu nhận được và điều khiển xe theo chuyển động mong muốn.

Kết quả nghiên cứu cho thấy

* Việc xử lý ảnh đạt được hiệu quả mong muốn .
* Hệ thống điều khiển xe đạt được mục đích ban đầu đề ra. Xe có thể di chuyển tiến, lùi, quay trái, quay phải, dừng .

Tuy nhiên, còn một số hạn chế cần được cải thiện trong tương lai. Ví dụ, hệ thống xe còn chưa di chuyển được linh hoạt, chưa đề cập đến việc nhiễu khi truyền tín hiệu, khả năng xử lý ảnh còn phụ thuộc vào điều kiện ánh sáng và chất lượng của camera ,...

**Kiến nghị**

Dựa trên nghiên cứu về xử lý ảnh điều khiển xe chúng em gợi ý các hướng phát triển và cải tiến trong tương lai để nâng cao hiệu suất và ứng dụng thực tế của hệ thống.

Nghiên cứu về điều kiện ánh sáng: Để cải thiện khả năng nhận dạng và phân loại đối tượng trong điều kiện ánh sáng yếu hoặc chói, cần tiếp tục nghiên cứu các thuật toán xử lý ảnh linh hoạt và khả năng thích ứng với biến đổi ánh sáng.

Mở rộng phạm vi đối tượng nhận dạng: Để xử lý hiệu quả trong các môi trường ,hệ thống cần có khả năng nhận dạng và phân loại nhiều loại đối tượng khác nhau, bao gồm ô tô, người đi bộ, xe đạp, biển báo giao thông, v.v.. Đồng thời, khám phá các phương pháp nhận dạng đối tượng mới và nâng cao khả năng tổng quát hóa cho các tình huống .

Tối ưu hóa tốc độ xử lý: Trong việc xử lý ảnh trực tiếp trong thời gian thực, tốc độ xử lý là yếu tố quan trọng.

Đánh giá hiệu suất và độ tin cậy: Để đảm bảo tính ứng dụng thực tế của hệ thống xử lý ảnh điều khiển xe, cần tiến hành đánh giá hiệu suất và độ tin cậy. Đề nghị thực hiện các thí nghiệm và kiểm tra đối với nhiều tình khác nhau để đánh giá khả năng nhận dạng đối tượng, độ chính xác trong việc phân loại đối tượng.

# **Tải liệu tham khảo**

* + - * Link video tham khảo :
      * <https://www.youtube.com/watch?v=UaPDbJsJ63Q&t=1452s>
      * https://www.youtube.com/watch?v=Egcw182f-WA&t=605s
      * Link các tải liệu tham khảo :
      * <http://arduino.vn/bai-viet/562-su-dung-module-nrf24l01>
      * https://vi.wikipedia.org/wiki/Arduino?fbclid=IwAR0cOERKL8dXVYvo4kr8cReQS32IyPoQIJcGtjlTzHbdyViGoLOwurD8zug
      * https://www.academia.edu/5753051/B%C3%A0i\_5\_Giao\_ti%E1%BA%BFp.UART
      * <https://huynhnhattung.com/nrf24l01-giao-tiep-arduino-thu-phat-rf-nrf24l01-2-4ghz-lcd1602/>
      * <https://developers.google.com/mediapipe/solutions>
      * https://www.analyticsvidhya.com/blog/2022/03/hand-landmarks-detection-on-an-image-using-mediapipe/