MỤC LỤC

[I/ Giới thiệu đề tài 4](#_Toc60194136)

[II. Linh kiện sử dụng 5](#_Toc60194140)

[1.Danh sách linh kiện 5](#_Toc60194141)

[2.Giới thiệu từng linh kiện: 5](#_Toc60194142)

[3. Mô hình lắp đặt 9](#_Toc60194143)

[4.  Sơ đồ mạch 10](#_Toc60194144)

[III. Các hệ thống triển khai: 10](#_Toc60194145)

[1. Chụp ảnh biển số xe 10](#_Toc60194146)

[2. Quy trình nhận diện biển số xe 10](#_Toc60194147)

[3.Tách biển số xe 12](#_Toc60194148)

[4. Phân đoạn kí tự 13](#_Toc60194149)

[5  Nhận diện ký tự 17](#_Toc60194150)

[6. Tỉ lệ chính xác 20](#_Toc60194153)

[IV. Ảnh demo thực tế: 20](#_Toc60194154)

[V. Website quản lý dữ liệu xe vào 22](#_Toc60194155)

[VI. Kết quả đạt được 24](#_Toc60194156)

[VII. Đánh giá và phát triển 24](#_Toc60194157)

[VIII.Tài liệu tham khảo 25](#_Toc60194158)

# Bảng phân công công việc

|  |  |  |
| --- | --- | --- |
| **STT** | **Họ và tên** | **Nội dung thực hiện** |
| 1 | Lê Nhật Tân  (Trưởng nhóm) | * Thiết kế cơ sở dữ liệu. * Xây dựng website hiển thị dữ liệu xe vào. * Hỗ trợ viết thuật toán. * Viết báo cáo. * Làm slide thuyết trình |
| 2 | Trần Thanh Như | * Thiết kế phần cứng cho hệ thống. * Thiết lập camera chụp hình. * Làm mô hình xe điều khiển thông qua di động * Thiết kế trang trí mô hình demo. * Viết báo cáo. |
| 3 | Nguyễn Diệu Quỳnh | * Thiết lập kết nối giữa phần cứng, thuật toán nhận diện và kiểm tra tính hợp lệ của biển số. * Bóc tách biển số xe. * Bóc tách kí tự. * Tổng hợp báo cáo. |
|  | | |

**LỜI CẢM ƠN**

Ngày nay, với sự tiến bộ của khoa học kĩ thuật và sự nâng cao về mức sống ở các thành phố, việc sỡ hữu các phương tiện giao thông như xe máy, xe hơi không còn xa lạ với mỗi người. Chính thực tiễn này đã dẫn đến việc nảy sinh nhiều vấn đề trong việc kiểm soát cũng như quản lý các phương tiện. Điều này đã giúp chúng em thực hiện “Mô hình bãi đỗ xe ô tô thông minh” như một giải pháp cơ bản cho việc tự động hóa quy trình đỗ xe. Với đề tài này, chúng em lựa chọn Arduno Uno và thuật toán nhận diện biển số xe.

Để hoàn thành đồ án này, chúng em xin chân thành cảm ơn cô Bùi Thị Thanh Thanh, là giáo viên hướng dẫn đã nhiệt tình chỉ bảo chúng em trong suốt quá trình hoàn thành đề tài.

# I/ Giới thiệu đề tài

1. **Lý do chọn đề tài :**

* Trong thời đại trí tuệ nhân tạo đang dần chiếm ưu thế, máy móc đang ngày càng “thông minh” hơn,việc tự động hóa các thiết bị xung quanh đang dần trở thành sự ưu tiên của hầu hết mọi người.
* Đơn giản như công việc gửi xe, chỉ vài năm trước đây khi đi đến bất kì bãi gửi xe nào chúng ta cũng thấy tất cả xe đều phải được nhân viên coi xe ghi bằng giấy, việc này là vô cũng bất tiện và nhiều rủi ro như giấy rất dễ bị rách hoặc dính nước…không những vậy việc ghi giấy không thể tránh khỏi sai sót khi người nhân viên coi xe một thoáng mất tập trung và ghi sai, vì vậy rất mất thời gian.
* Đặc biệt đối với các bãi đỗ xe nội bộ cho công nhân viên, bảo vệ phải quan sát qua camera để xác nhận người trong nội bộ cũng như điều khiển thanh chắn mỗi lần có xe vào, dẫn đến tốn thời gian, công sức.
* Nhưng giờ đây đó không còn là vấn đề khi chúng ta có công nghệ nhận diện biển số xe.
* Trong đề tài này chúng em sẽ trình bày phương pháp làm sao để có thể nhận diện được biển số xe đó.

1. **Mục tiêu của đề tài**:

Trong quá trình đề tài nhóm đề ra 2 mục tiêu chính.

-    Có kiến thức cơ bản về lập trình Python, hiểu được khái quát quy trình xây dựng một thuật toán học máy. Vì đây là ngôn ngữ vô cùng phổ biến và quan trọng, nhất là trong lĩnh vực AI và Machine Learning.

-    Tìm hiểu thư viện OpenCV qua đó ứng dụng nó nhận diện biển số xe một cách tương đối chính xác.

- Áp dụng được các kiến thức đã học về lập trình hệ thống nhúng để xây dựng các hệ thống tự động nhỏ.

1. **Giới hạn đề tài:**

Tạo dựng một hệ thống bãi giữ xe ô tô, có khả năng tự nhận dạng biển số xe và xác định có cho phép xe ra vào hay không.

# II. Linh kiện sử dụng

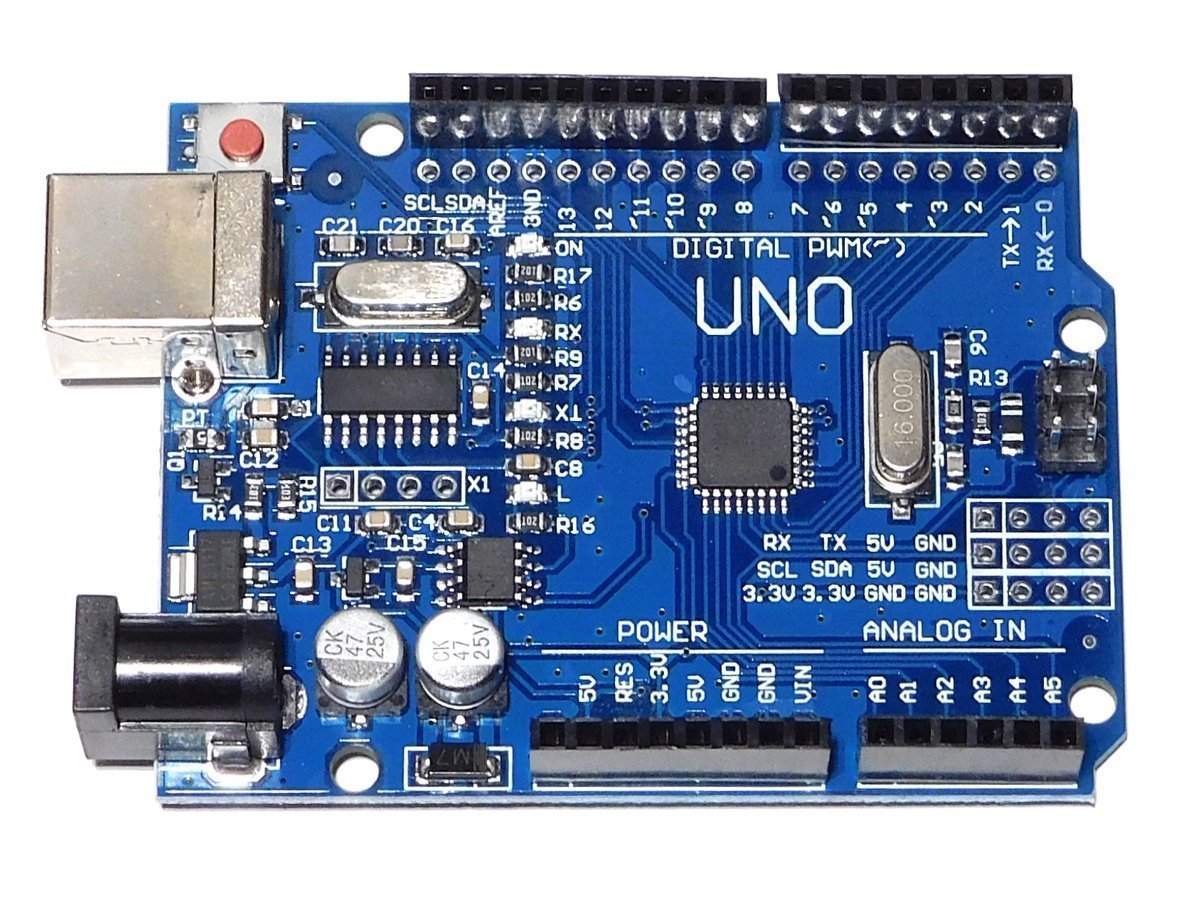
## 1.Danh sách linh kiện

|  |  |
| --- | --- |
| **STT** | **Tên linh kiện** |
| **1** | Arduino Uno R3 |
| **2** | Servo SG90 |
| **3** | Cảm biến khoảng cách HC-SR04 |
| **4** | Màn hình LCD + I2C |
| **5** | Webcam điện thoại |

## 2.Giới thiệu từng linh kiện:

**2.1 Arduino Uno R3:**

**Arduino Uno** là một bảng mạch vi điều khiển nguồn mở dựa trên vi điều khiển Microchip ATmega328. Bảng mạch được trang bị các bộ chân đầu vào/đầu ra Digital và Analog có thể giao tiếp với các bảng mạch mở rộng khác nhau. Dựa trên nền tảng mở do Arduino.cc cung cấp để ta dễ dàng dùng Arduino Uno xây dựng cho mình những dự án như lập trình Robot, xe tự hành, điều khiển bật tắt led,…



Bảng 1- Các thông số kĩ thuật của Arduino U no R3

|  |  |
| --- | --- |
| Chip điều khiển | ATmega328P |
| Điện áp hoạt động | 5V |
| Số chân Digital | 14 chân |
| Số chân Analog | 6 chân |
| Chiều dài  Chiều rộng | 68mm  53,4mm |
| SRAM | 2KB (ATmega328P) |
| Flash Memory | 32KB |
| EEPROM | 1KB(ATmega328P) |

**2.2 Servo SG90**

Là một dạng động cơ điện đặc biệt. Không giống như động cơ thông thường cứ cắm điện vào là quay liên tục, servo chỉ quay khi được điều khiển  với góc quay nằm trong khoảng bất kì từ 0o - 180o

- Điện áp hoạt động: 3-6 V

- Kết nối 3 chân với 2 chân cấp nguồn (VCC và GND) và 1 chân nhận tín hiệu vào

- Lực kéo: 1.6KG.CM, góc quay 0-1800

- Kích thước: 21x12x22mm



**2.3 Cảm biến khoảng cách HC-SR04**

- Điện áp hoạt động 5V

- Phạm vi phát hiện vật cản 2 - 300 cm.

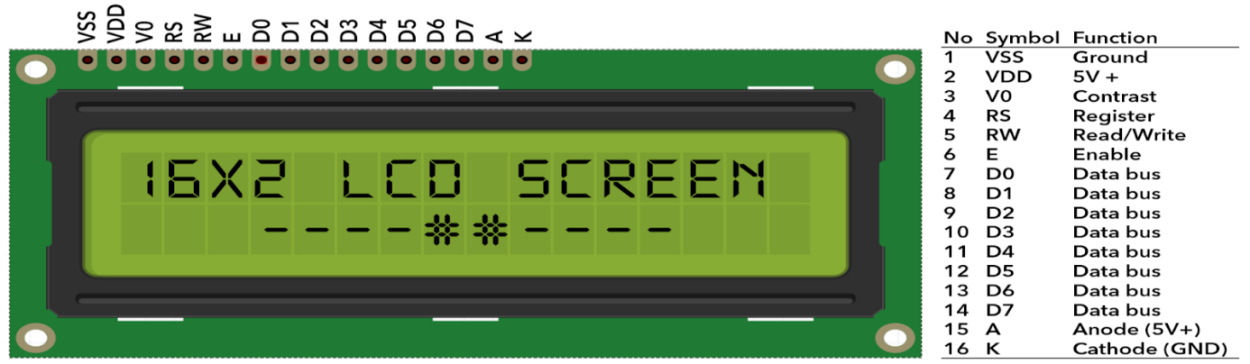
- Kết nối 4 chân với 2 chân cấp nguồn (VCC và GND) và 1 chân tín hiệu ngõ vào Trig và 1 chân tín hiệu ngõ ra echo.



**2.4 Màn hình LCD**

**Thông số kỹ thuật**

* Điện áp hoạt động: 5V
* Kích thước: 80 x 36 x 12.5 mm
* Chữ trắng, nền xanh
* Khoảng cách giữa hai chân kết nối là 0.1 inch tiện dụng khi kết nối với Breadboard
* Đèn led nền có thể dùng biến trở hoặc PWM điều chỉnh độ sáng thích hợp

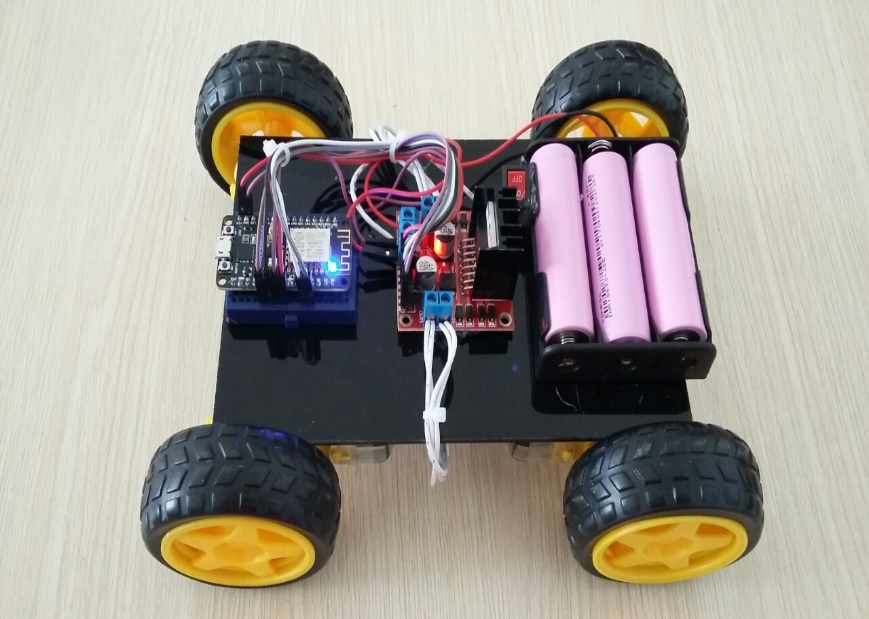


**2.5. Webcam**

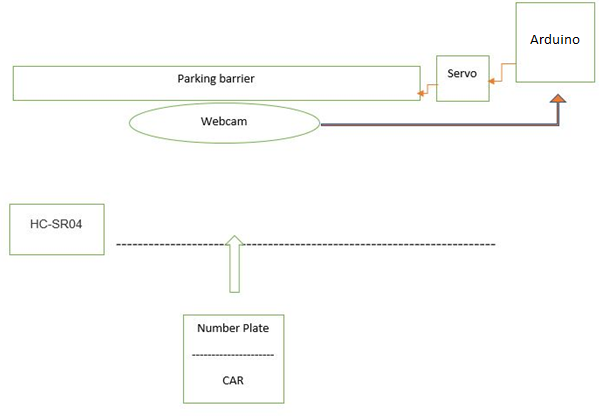
Bài báo cáo này ta dùng webcam có sẵn trên điện thoại kết hợp với phần mềm IP Webcam để truyền hình ảnh, video stream thông qua internet đến máy tính để xử lý.

**2.6 Mô hình xe**

Mô hình xe này sử dụng module Wifi esp 8266 dùng để kết nối với điện thoại để thực hiện chức năng điều khiển từ xa. Mạch sử dụng nguồn cấp là nguồn rời 9v. Mô hình sử dụng mạch L298N để điều khiển động cơ DC đóng vai trò là các bánh xe.

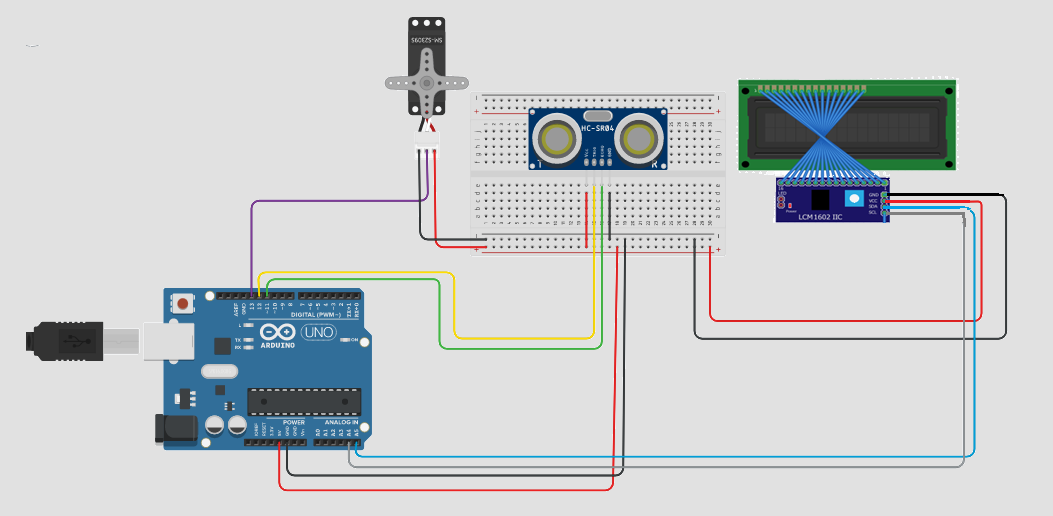


## Mô hình lắp đặt



*Hình 1 Mô hình lắp đặt*

## Sơ đồ mạch



*Hình 2 Sơ đồ mạch*

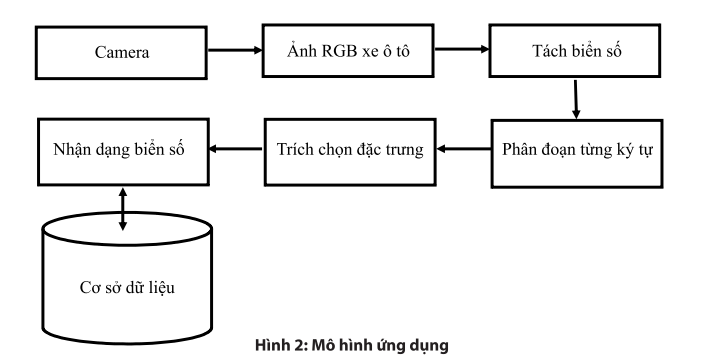
# III. Các hệ thống triển khai:

## 1. Chụp ảnh biển số xe

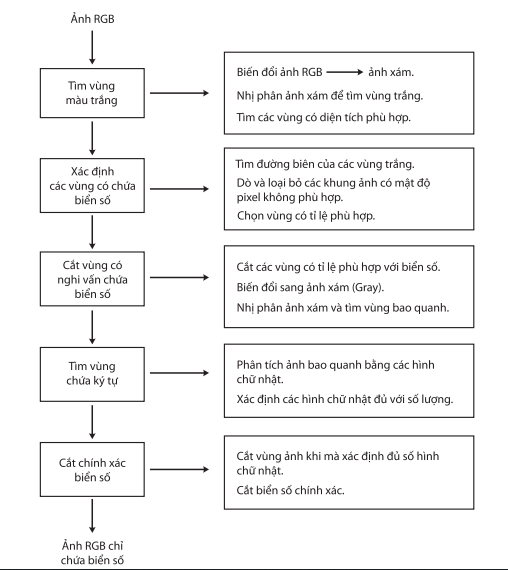
Setup như ảnh demo bên dưới.

Sử dụng cảm biến khoảng cách HC-SR04 để nhận biết có xe vào bãi, nếu có xe vào, gủi tín hiệu thông qua cổng Serial của Arduino đến máy tính kích hoạt điều khiển webcam để chụp lại ảnh. Sau đó tiền hành xử lí, gửi trả lại dữ liệu cho Arduino để điều khiển servo và cho ra kết quả trên màn hình LCD.

## 2. Quy trình nhận diện biển số xe

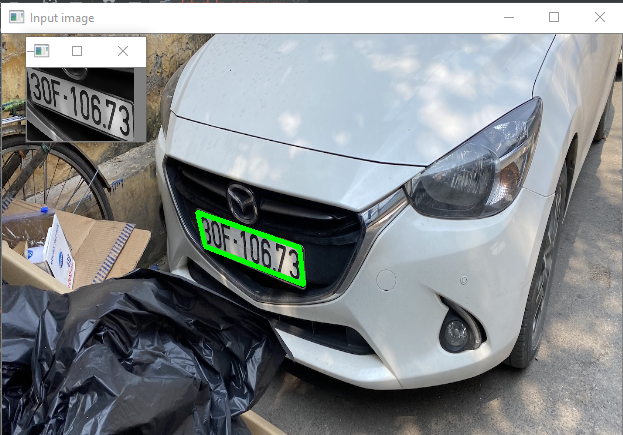
* Bước 0: Chuẩn bị dữ liệu chữ số và train một model SVM để nhận diện ký tự sau đó lưu ra file.
* Bước 1: Đọc file ảnh đầu vào. Ảnh này gồm toàn cảnh chiếc xe.
* Bước 2: Dùng WPOD để tách biển số ra khỏi ảnh và thực hiện skew ảnh để dễ xử lý tiếp.
* Bước 3: Thực hiện threshold để phân tách trắng đen và đồng thời khử nhiễu.
* Bước 4. Tìm các contour trong ảnh đã threshold và giữ lại các contour có khả năng là số.
* Bước 5. Nhận diện các contour đó bằng SVM đã train ở bước 0 và in kết quả biển số ra màn mình.
* 

## 3.Tách biển số xe



**3.1 Tiền xử lý:**

* + *Bước 1.Định nghĩa vài thông số cho bài toán như: Kích thước biển số lớn nhất và kích thước biển số nhỏ nhất. Chúng ta tùy vào hiện trường thực tế mà tinh chỉnh cho phù hợp.*
* *Bước 2. Load ảnh và thực hiện reszie về kích thước mong muốn.*
* *Bước 3. Chuyển về đen trắng và thực hiện thuật toán tìm cạnh băng Canny*
* *Bước 4. Tìm các contour có trong hình.*
* *Bước 5. Với các contour tìm được ở bước 4, chúng ta lặp qua 1 lượt và chọn contour lớn nhất hội đủ các điều kiện:*
* *Có 4 cạnh (sau khi đã approx): vì biển số là đối tượng có 4 cạnh.*
* *Có kích thước nằm trong vùng kích thước đã định nghĩa ở bước 1.*
* *Contour đó sẽ là biển số chúng ta cần tìm.*
* *Bước 6. Tiến hành tìm các điểm góc của contour nói trên và trích hình ảnh biển số ra khỏi hình ảnh cả xe.*



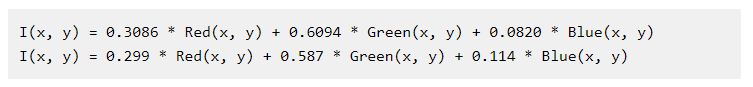
## 4. Phân đoạn kí tự

**4.1 Tiền xử lý:**

* Ta thực hiện chuyển đổi ảnh về ảnh thang độ xám.

Chuyển đổi từ ảnh 24 bits RGB sang ảnh 8 bits Grayscale.

***Công thức tính cường độ sáng tại 1 điểm ảnh từ ảnh RGB:***



I(x, y): cường độ sáng tại điểm ảnh (x, y) của ảnh xám.

Red(x, y): giá trị của kênh màu Red (Đỏ) tại điểm ảnh (x, y) của ảnh màu (RGB).

Green(x, y): giá trị của kênh màu Green (Xanh lá cây) tại điểm ảnh (x, y) của ảnh màu (RGB).

Blue(x, y): giá trị của kênh màu Blue (Xanh lơ) tại điểm ảnh (x, y) của ảnh màu (RGB).





Ảnh thang độ xám

* Nhị phân hóa ảnh với ngưỡng động

***Nhị phân hóa:*** Là quá trình biến đổi 1 ảnh xám thành ảnh nhị phân.

Gọi giá trị cường độ sáng tại 1 điểm ảnh là I(x,y) .

INP(x,y) là cường độ sáng của điểm ảnh trên ảnh nhị phân .

(Với 0 < x < image.width) và (0 < y < image.height).

Để biến đổi ảnh xám thành ảnh nhị phân. So sánh giá trị cường độ sáng của điểm ảnh với 1 ngưỡng nhị phân T.

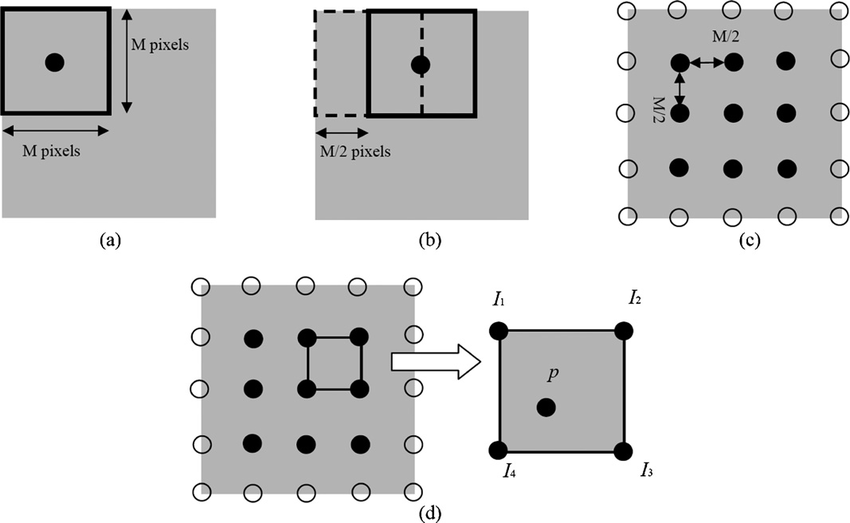
Nếu I(x,y) > T thì INP(x, y) = 0 (0).

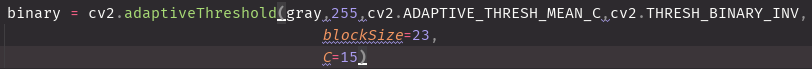
Nếu I(x,y) > T thì INP(x, y) = 255 (1).

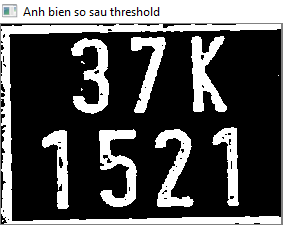


***Nhị phân hóa ngưỡng động***

* Chia ảnh thành nhiều khu vực, cửa sổ khác nhau (Region).
* Dùng 1 thuật toán để tìm 1 giá trị T phù hợp với từng khu vực, cửa sổ (Region).
* Áp dụng phương pháp nhị phân hóa cho từng khu vực, cửa sổ (Region) với T phù hợp.







Ngưỡng ảnh.

* Segment kí tự



****

**4.2 Tìm các contour có khả năng là số:**

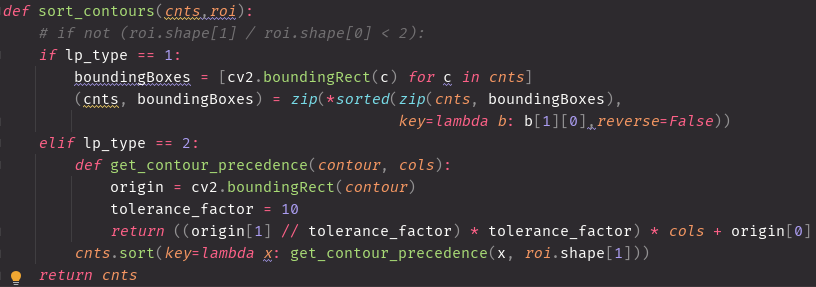


Về phần tìm contour là số: duyệt hết các contour mà kiểm tra các bounding box của các contour đó, nếu có 1 tỷ lệ nhất định giữa width và height thì nó là số. Trong bài nhóm để ratio này nằm trong khoảng 1.3 < ratio < 3.5. Ngoài ra chọn các contour có chiều cao trong khoảng 90% và 60% chiều cao biển số trở lên và trong khoảng 50% và 35 % đối với biển số vuống 2 dòng.

* Cấu hình tham số cho model SVM

digit\_w = 30 *# Kich thuoc ki tu*digit\_h = 60 *# Kich thuoc ki tu*model\_svm = cv2.ml.SVM\_load('trainSVM/svm.xml')

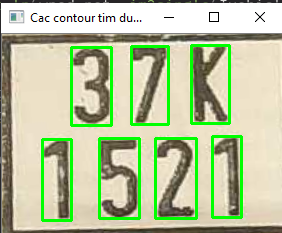
* Trong mỗi danh sách contour phù hợp, sắp xếp các contour từ trái sang phải và từ trên xuống dưới đối với biển số 2 dòng.



Tiếp tục viền khung xanh từng ký tự để xử lý tiếp bước tiếp theo là nhận diện ký tự đó là ký tự gì.





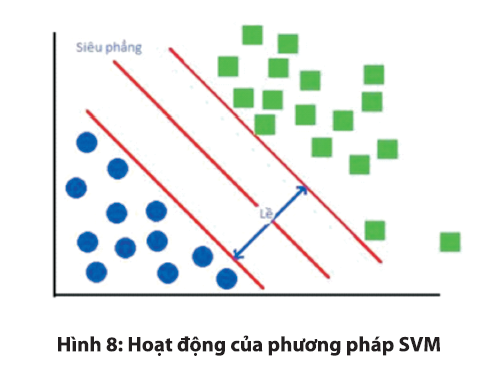


Tách và viền khung xanh từng ký tự.

## 5  Nhận diện ký tự

Bài toán xử lý ảnh để nhận diện biển số xe sử dụng thuật toán chính là SVM(Support Vector Machine).

* 1. **. Cơ sơ lí thuyết:**
  + Phương pháp SVM là một mô hình máy học giám sát được dùng trong việc phân tích, phân lớp dữ liệu dựa vào các siêu phẳng. Giả sử ta có một tập dữ liệu hai chiều như hình dưới, khi đó ta có thể phân lớp dữ liệu này thành 2 phần nhờ một siêu phẳng. Siêu phẳng trong mặt phẳng là một đường thẳng, trong không gian 3 chiều là một mặt phẳng và tổng quát trong không gian n chiều là một không gian n-1 chiều. Trong trường hợp dữ liệu là không tuyến tính, ta cần ánh xạ dữ liệu đó lên một không gian có số chiều lớn hơn để thuận tiện cho việc phân loại dữ liệu. Nhiệm vụ là cần phải tìm siêu phẳng sao cho khoảng cách tới các biên của dữ liệu là lớn nhất. Hiểu 1 cách đơn giản về phương pháp này là: cho 1 tập các mẫu huấn luyện, với mỗi mẫu được gắn nhãn, quá trình huấn luyện SVM sẽ xây dựng một mô hình cho phép dự đoán một tập dữ liệu khác thuộc về nhãn nào, tức phân loại tập dữ liệu đó thuộc vào lớp nào. [3]



Nhận dạng ký tự trải qua 2 giai đoạn sau:

a. Chuẩn bị dữ liệu huấn luyện

Cần chuẩn bị cơ sở dữ liệu là tập hợp của các ký tự trong biển số. Có 33 ký tự thường gắp trong biển số, do đó ta cần phân lọai 33 lớp này, trong trường hợp giả sử với mỗi lớp, tức là mỗi ký tự có 10 ảnh, ta sẽ lưu các ảnh này vào các folder, chẳng hạn như folder 0 có chứa 10 ảnh của ký tự 0, folder 1 chứa 10 ảnh của ký tự 1,..Ta cần đánh tên folder theo số thứ tự, vì số thứ tự cũng là nhãn tương ứng đưa vào việc nhận dạng. Ta sẽ tính toán đặc trưng của từng ký tự và lưu tất cả các đặc trưng này vào 1 ma trận để phục vụ việc huấn luyện. Hàm tính toán đặc trưng tronh một ảnh sẽ dựa trên ý tưởng tổng các điểm đen trong 1 khung hình, tuy nhiên nó chưa được chuẩn hóa bằng cách chaia cho tổng các điểm đen của ký tự.

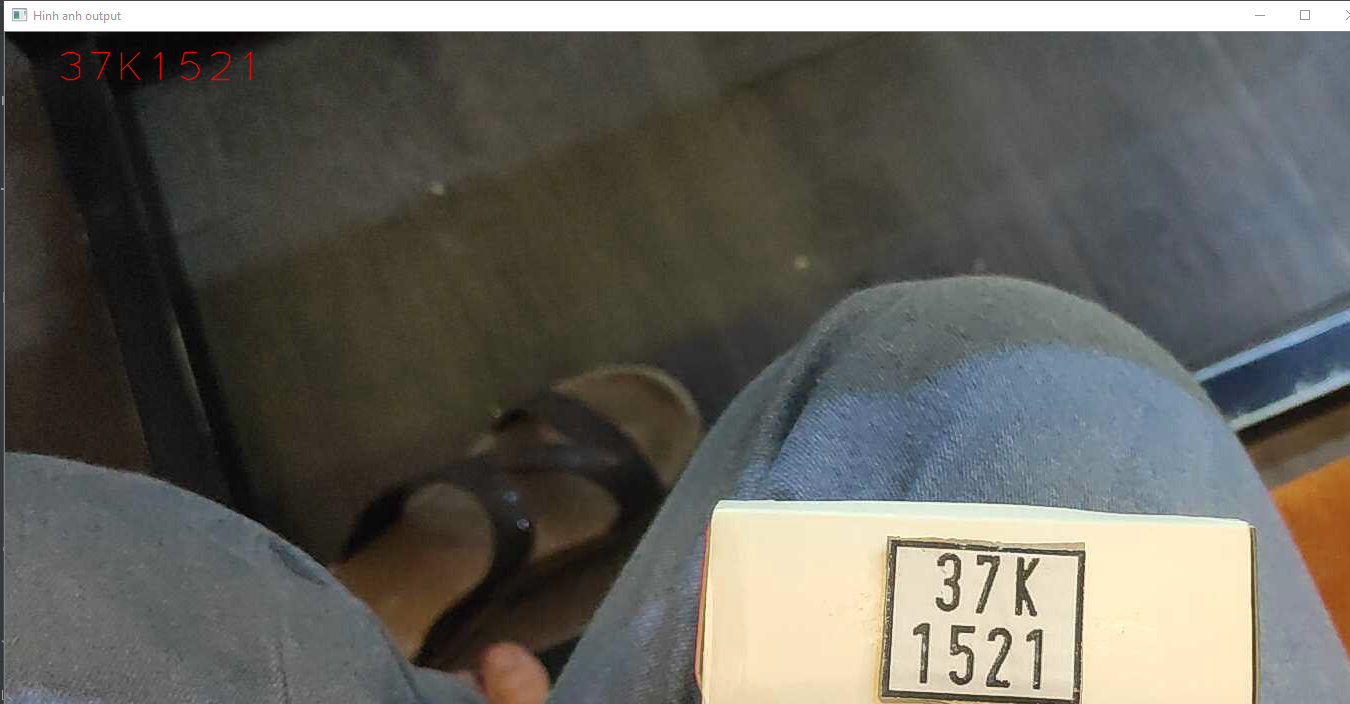
b. Nhận dạng ký tự dựa trên phương pháp SVM

Trước hết ta cần nhận thấy SVM là 1 bộ máy phân loại dữ liệu, muốn sử dụng được nó ta cần phải có dữ liệu, dữ liệu đối với các ký tự mà ta cần nhận dạng ở dây là các đặc trưng trong ảnh của ký tự đó. Giả sử ta cần phân loại 33 lớp dữ liệu , với mỗi lớp dữ liệu, ta tính toán đưỡ 10 vector đặc trưng (10 mẫu), và mỗi vector đặc trưng tương ứng với các đặc trưng trong 1 ảnh. Khi đó ta sẽ đưa vào bộ huấn luyện SVM toàn bộ dữ liệu này, sau đó với 1 ảnh bất kỳ, ta sẽ tính toán 1 vector đặc trưng của ảnh đó, mô hình SVM sẽ xem xét dữ liệu này ( tức vector đặc trưng) thuộc vào lớp nào trong những lớp nó đã huấn luyện.

* 1. **Nhận diện ký tự trong khung:**
* Sử dụng phương pháp SVM training để train dữ liệu.
* Bộ dữ liệu là tập dữ liệu chữ số: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9 và tập chữ cái: A, B, C, D, E, F, G, H, I, K, L, M, N, O, P, Q, R, S, T, U, V, X, T, Y, W có kích thước 30X60.
* Dữ liệu đã được train và lưu vào file để máy học.
* Cuối cùng ta thu được chuỗi ký tự là mã số của biển số xe và in chuỗi ký tự này ra cửa sổ “ python shell “.



* Ngoài ra chúng ta cũng có thể viết chuỗi ký tự này lên ảnh gốc để quan sát và so sánh.



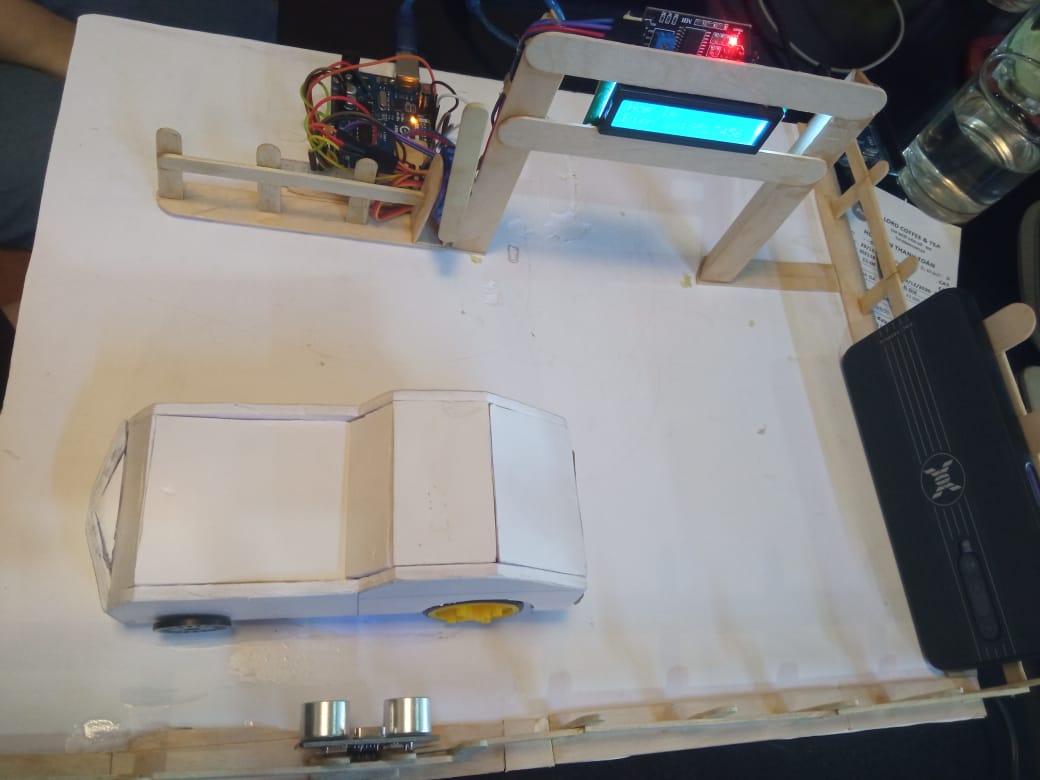
*Hình 9 Kết quả là biển số xe được nhận diện*

**6. Tỉ lệ chính xác**

Tiến hành thử nhiệm 2 lần, mỗi lần sử dụng 45 và 41 ảnh được chụp quanh khuôn viên trường và ngoài đường. Kết quả:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Số lần thử** | **Số hình ảnh thử demo** | **Số hình ảnh nhận dạng chính xác** | **Số hình ảnh nhận dạng không chính xác** | **Tỉ lệ tìm đúng** |
| 1 | 45 | 38 | 7 | 84.4 % |
| 2 | 41 | 34 | 7 | 82.9% |

# IV. Ảnh demo thực tế:

****

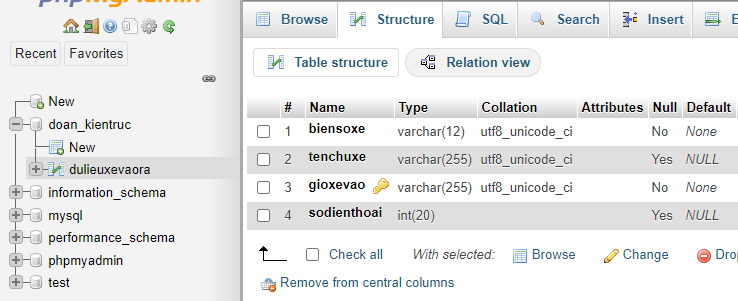
*Hình 10 Tổng quan mô hình thực tế.*



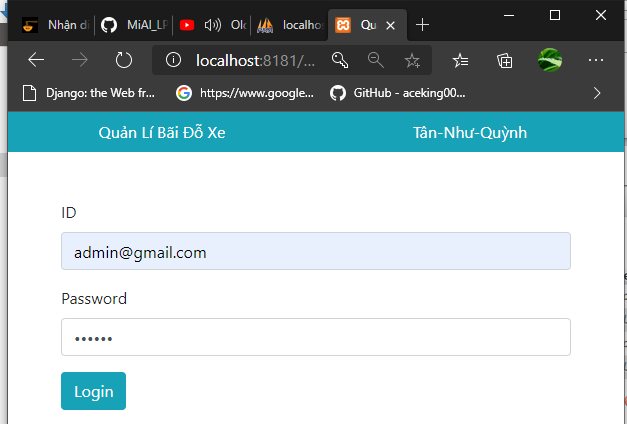
*Hình 11 Mô hình xe chứa biển số xe.*

# V. Website quản lý dữ liệu xe vào

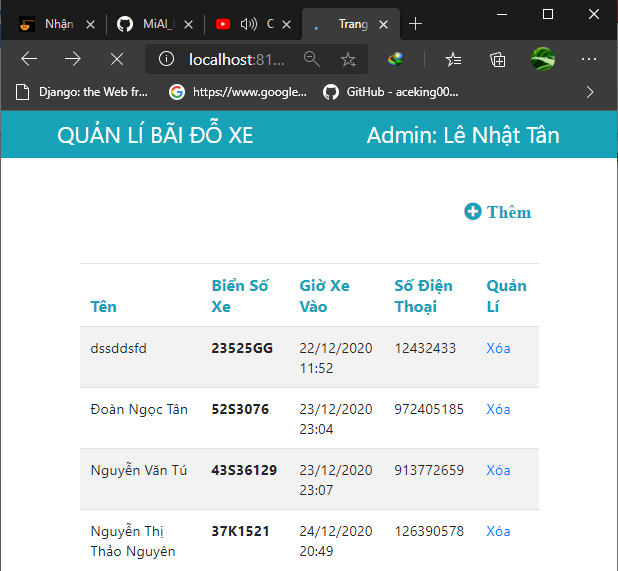
**1. Xây dựng cơ sở dữ liệu xe vào**

****

**2. Xây dựng trang đăng nhập**

****

**3. Xây dựng trang hiển thị dữ liệu xe vào**

****

**4. Xây dựng chức năng thêm dữ liệu**

****

**5. Xây dựng chức năng xóa**



# VI. Kết quả đạt được

- Nhìn chung nhóm đã đạt được những mục tiêu đề ra ban đầu là hoàn thành đề tài nhận diện biển số xe trong mô hình bãi giữ xe thông minh và tìm hiểu về ngôn ngữ lập trình Python, OpenCV ,dù còn tương đối nhiều hạn chế và thiếu sót nhưng hy vọng nhóm sẽ cải thiện hơn trong tương lai.

# VII. Đánh giá và phát triển

**1 Hạn chế của đề tài:**

Vẫn chưa xử lí realtime, mỗi lần nhận diện chỉ được một xe duy nhất. Sau đó phải chạy lại chương trình để tiếp tục xử lí xe khác.

**2 Hướng phát triển**

* Triển khai mô hình xử lí thời gian thực, để đạt được hiệu quả tối ưu.
* Từ đề tài nhận diện biển số xe trong bãi giữ xe thông minh, chúng ta có thể ứng dụng trực tiếp nó vào nhận diện biển số ở các bãi gửi xe trong trường Đại học Bách Khoa, trong toàn thành phố Đà Nẵng và các tỉnh thành cả nước.
* Nếu có thể phát triển lên có thể ứng dụng trong lĩnh vực trí tuệ nhân tạo như robot đọc chữ hoặc phát triển thành máy phát hiện ký tự hỗ trợ những người khiếm thính,mù lòa, ...

# VIII.Tài liệu tham khảo

1. Các bài báo cáo, nghiên cứu về nhận diện biển số xe.
2. Tài liệu về các linh kiện điện tử được sử dụng.
3. https://www.miai.vn/2019/12/05/nhan-dien-bien-so-xe-chuong-5-nhan-dien-bien-so-xe-bang-wpod-va-svm/
4. Xây dựng ứng dụng nhận dạng biển số xe ô tô sử dụng mã nguồn mở OpenCV, ThS. Nguyễn Duy Linh và Sinh viên Nguyễn Văn Nhân, Đặng Văn Đạt, Tạp chí thông tin khoa học và công nghệ Quảng Bình.
5. https://www.stdio.vn/computer-vision/ky-thuat-grayscale-va-nhi-phan-hoa-anh-adaptive-threshold-Fhjcg
6. https://opencv-python-tutroals.readthedocs.io/en/latest/py\_tutorials/py\_imgproc/py\_thresholding/py\_thresholding.html
7. https://giaphiep.com/blog/gioi-thieu-ve-support-vector-machine-svm-27209
8. https://www.miai.vn/2019/11/26/nhan-dien-bien-so-xe-chuong-3-phat-hien-bien-so-bang-opencv-thuan/