**CÔNG TY CỔ PHẦN CÔNG NGHỆ TK25**

🙡★🙣



**BÁO CÁO ĐỀ TÀI**

**BÃI GIỮ XE THÔNG MINH**

**SỬ DỤNG NHẬN DIỆN BIỂN SỐ XE**

**Người thực hiện:** Tô Quang Huấn

Trương Bảo Nguyên

**TP. Hồ Chí Minh – tháng 10 năm 2022**

# MỤC LỤC

[MỤC LỤC i](#_Toc116397672)

[DANH MỤC HÌNH iii](#_Toc116397673)

[DANH MỤC BẢNG iv](#_Toc116397674)

[Chương 1. TỔNG QUAN ĐỀ TÀI 1](#_Toc116397675)

[1.1. Giới thiệu chung 1](#_Toc116397676)

[1.2. Hiện trạng 1](#_Toc116397677)

[1.3. Mục tiêu 1](#_Toc116397678)

[1.4. Phương pháp thực hiện 2](#_Toc116397679)

[1.4.1. Đối tượng 2](#_Toc116397680)

[1.4.2. Phạm vi nghiên cứu 2](#_Toc116397681)

[Chương 2. PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG 3](#_Toc116397682)

[2.1. Sơ đồ hệ thống 3](#_Toc116397683)

[2.1.1. Sơ đồ thiết bị 3](#_Toc116397684)

[2.1.2. Sơ đồ hoạt động 4](#_Toc116397685)

[Chương 3. HIỆN THỰC VÀ KẾT QUẢ THỰC NGHIỆM 10](#_Toc116397686)

[3.1. Thuật toán nhận diện biển số xe 10](#_Toc116397687)

[3.1.1. Các bước xử lý 10](#_Toc116397688)

[3.1.2. Định vị vùng chứa biển số xe 11](#_Toc116397689)

[3.1.3. Nhận dạng ký tự 15](#_Toc116397690)

[3.1.4. Xử lý ký tự nhầm lẫn 19](#_Toc116397691)

[3.2. Xây dựng Server 20](#_Toc116397692)

[3.2.1. Cấu trúc gói tin lưu trữ lên Server 20](#_Toc116397693)

[3.2.2. Gửi gói tin lên Server 20](#_Toc116397694)

[3.3. Kết quả thực nghiệm 22](#_Toc116397695)

[Chương 4. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN 23](#_Toc116397696)

[4.1. Kết luận 23](#_Toc116397697)

[4.1.1. Ưu điểm 23](#_Toc116397698)

[4.1.2. Nhược điểm 23](#_Toc116397699)

[4.2. Hướng phát triển 23](#_Toc116397700)

[Chương 5. TÀI LIỆU THAM KHẢO 24](#_Toc116397701)

# DANH MỤC HÌNH

[Hình 2.1: Sơ đồ tổng quát các thiết bị của hệ thống 3](#_Toc116392373)

[Hình 2.2: Lưu đồ hoạt động của hệ thống 4](#_Toc116392374)

[Hình 2.3: Lưu đồ quá trình khởi tạo và kiểm tra hệ thống 5](#_Toc116392375)

[Hình 2.4: Lưu đồ quy trình các bước trích xuất biển số xe 6](#_Toc116392376)

[Hình 2.5: Lưu đồ quy trình hình thành data package 7](#_Toc116392377)

[Hình 2.6: Lưu đồ quy trình khởi tạo container 8](#_Toc116392378)

[Hình 2.7: Lưu đồ quy trình upload file lên server 9](#_Toc116392379)

[Hình 3.1: Các bước xử lý bài toán nhận diện biển số xe 10](#_Toc116392380)

[Hình 3.2: Detect biển số xe của YOLO 12](#_Toc116392381)

[Hình 3.3: Detect biển số xe của WPOD-NET 12](#_Toc116392382)

[Hình 3.4: Quy trình phát hiện biển số bằng OpenCV 13](#_Toc116392383)

[Hình 3.5: Các kết quả của giai đoạn tiền xử lý ảnh 14](#_Toc116392384)

[Hình 3.6: Biển số xe đã được cắt 15](#_Toc116392385)

[Hình 3.7: Tiền xử lý ảnh biển số xe 15](#_Toc116392386)

[Hình 3.8: Các ký tự đã được xác định 16](#_Toc116392387)

[Hình 3.9: Các ký tự đã được cắt ra và chuyển về dạng Binary 16](#_Toc116392388)

[Hình 3.10: Kết quả dự đoán các ký tự 16](#_Toc116392389)

[Hình 3.11: Một số ví dụ kết quả đọc ký tự sử dụng EasyOCR 17](#_Toc116392390)

[Hình 3.12: File json được lưu trữ trong Azure Blob Storage 21](#_Toc116392391)

[Hình 3.13: File jpg được lưu trữ trong Azure Blob Storage 21](#_Toc116392392)

[Hình 3.14: Nội dung file json được lưu trữ trên Azure Blob Storage 21](#_Toc116392393)

[Hình 3.15: Hình ảnh được lưu trữ trên Azure Blob Storage 22](#_Toc116392394)

# DANH MỤC BẢNG

[Bảng 1: Bảng so sánh giữa 3 giải thuật OCR 18](#_Toc115767062)

[Bảng 2: Bảng biểu thị tần số nhầm lẫn giữa các số và ký tự 19](#_Toc115767063)

# TỔNG QUAN ĐỀ TÀI

## Giới thiệu chung

Bãi giữ xe thông minh là hệ thống tích hợp nhiều thiết bị riêng lẻ kiểm soát an ninh vào ra lại với nhau tạo thành một hệ thống giữ xe thông minh giúp quản lý chặt chẽ, an toàn và tiện lợi. Với sự phát triển của xã hội, con người ngày càng ứng dụng các hệ thống quản lý thông minh, tự động hóa vào công việc để phục vụ, hỗ trợ con người làm việc tốt hơn, hiệu quả hơn mà chi phí lại thấp hơn. Vì vậy, hệ thống bãi giữ xe thông minh ngày càng được sử dụng rộng rãi.

## Hiện trạng

Trong những năm gần đây cùng với sự phát triển của nền kinh tế, thu nhập đời sống của người dân được nâng lên cùng với đó là sự gia tăng chóng mặt của số lượng phương tiện giao thông, đặc biệt là trong các thành phố lớn.

Trong khi đó, theo thống kê hiện nay số lượng các bãi đỗ xe có giấy phép ở các thành phố chỉ đáp ứng được 8-10% nhu cầu người dân, dẫn tới tình trạng thiếu bãi đỗ xe nghiêm trọng. Đồng thời điều đó gây ra tình trạng đỗ, dừng xe không đúng nơi quy định. Các điểm đỗ xe được tận dụng ở mọi chỗ, mọi nơi như trên vỉa hè, lòng lề đường, các công viên... Điều đó ảnh hưởng không nhỏ đến giao thông nội thành và mỹ quan đô thị.

Hiện nay có rất nhiều hệ thống bãi giữ xe thông minh trên thị trường, đa số trong đó là các hệ thống sử dụng thẻ từ RFID kết hợp Camera nhận diện biển số xe dựa trên phần cứng là máy tính có sức mạnh xử lý cao.

## Mục tiêu

Ở đề tài này, nhóm sẽ hướng đến xây dựng hệ thống bãi giữ xe thông minh sử dụng UHF RFID tầm xa để quét thẻ Etag đang được nhà nước áp dụng cho trạm thu phí không dừng, kết hợp camera nhận diện biển số xe dựa trên phần cứng là Raspberry Pi 4. Ưu điểm của hướng tiếp cận này so với những hệ thống trên thị trường là tận dụng được thẻ Etag được dán ở trên mọi xe hơi ở Việt Nam, không cần người dùng phải giữ thẻ vật lý, chờ đợi việc quét thẻ và đặc biệt là chi phí của hệ thống này thấp vì chỉ cần Raspberry Pi để xử lý, truyền dữ liệu thay vì máy tính có khả năng xử lý mạnh.

Đặc biệt, đề tài hướng đến xây dựng giải pháp cho các bãi giữ xe quy mô lớn, có nhiều bãi như sân bay, siêu thị, trường học, với mỗi bãi được lắp đặt mỗi node và các node có thể giao tiếp, đồng bộ dữ liệu với nhau từ đó xây dựng được hệ thống giữ xe thông minh có khả năng thông báo và hướng dẫn người dùng đỗ xe ở những bãi đỗ còn chỗ trống cũng như thuận tiện cho việc tính phí, thống kê, quản lý lịch sử, doanh thu của điểm giữ xe.

Đồng thời nhóm cũng thiết kế một hệ thống các thiết bị nhỏ gọn, đơn giản, có khả năng cài đặt và tháo dỡ dễ dàng. Điều này làm cho hệ thống phù hợp với mọi nhu cầu sử dụng, từ quy mô lớn đến nhỏ. Hệ thống bãi giữ xe thông minh này còn có thể được lắp đặt với quy mô hộ gia đình, giúp người dùng tận dụng những không gian trống để xây dựng được một bãi giữ xe, giảm thiểu các tình trạng kẹt xe, dừng đỗ xe bừa bãi,...

## Phương pháp thực hiện

### Đối tượng

* Raspberry Pi 4.
* Board UHF Reader (M6e Nano kết hợp ESP32).
* Các phương pháp, thuật toán phục vụ cho việc phát hiện và nhận dạng biển số xe.
* Bộ thư viện xử lý ảnh OpenCV, thư viện nhận dạng ký tự Pytesseract.
* Azure Storage, Blob container.

### Phạm vi nghiên cứu

* Tìm hiểu nhận dạng biển số xe
* Tìm hiểu việc đọc dữ liệu từ tag sử dụng board UHF Reader
* Thu thập hình ảnh từ Camera với Raspberry Pi
* Truyền nhận dữ liệu qua UART giữa Raspberry Pi với board UHF Reader
* Xây dựng Server với Azure Storage
* Tìm hiểu cảm biến phát hiện vật thể, cảm biến an toàn
* Nhận dạng biển số xe, cập nhật Server và mở cổng

# PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG

## Sơ đồ hệ thống

### Sơ đồ thiết bị

****

Hình .: Sơ đồ tổng quát các thiết bị của hệ thống

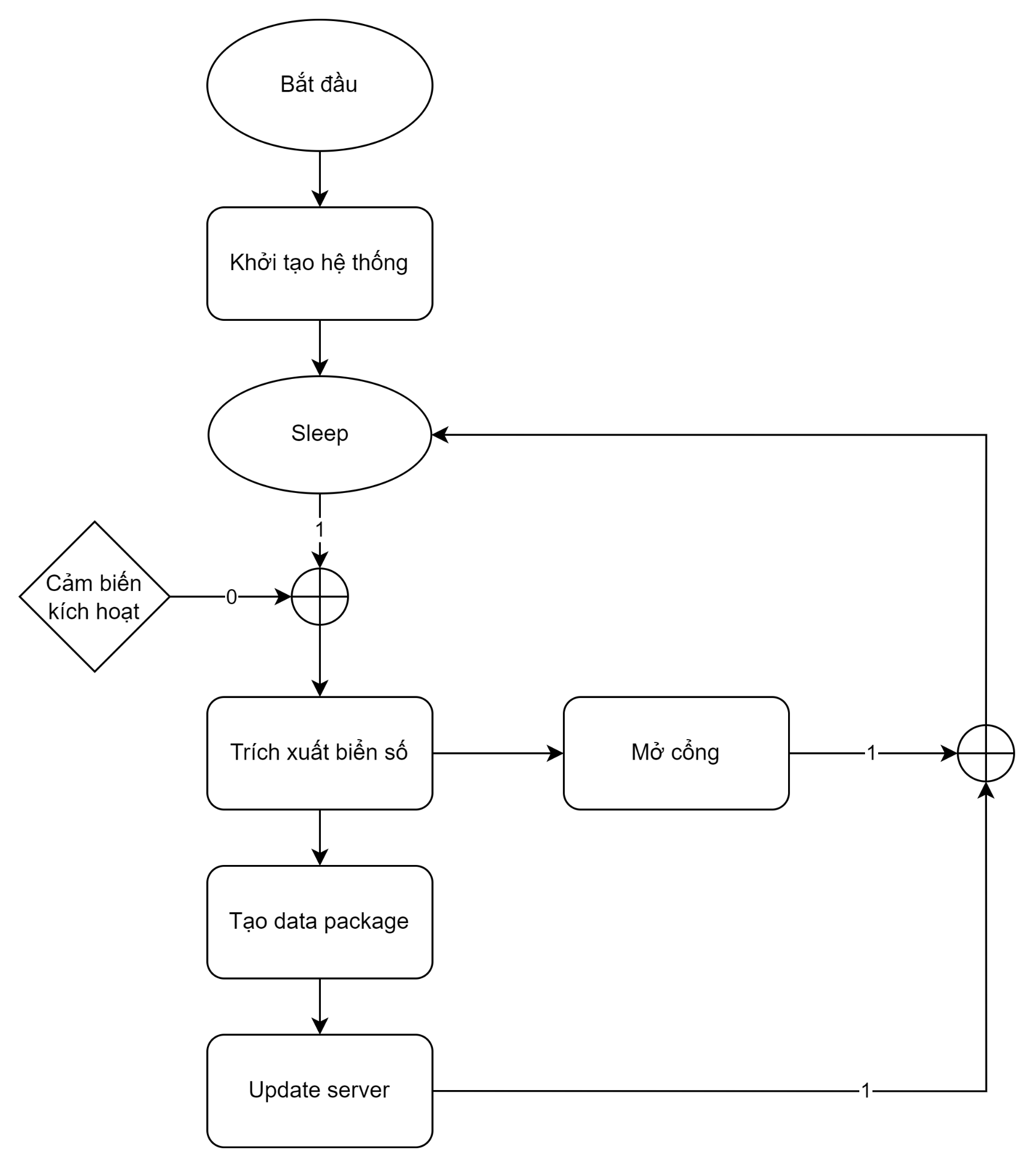
Về tổng quan hệ thống sẽ có hai khu vực xử lý chính là phía node và phía cloud. Node sẽ được lắp đặt ở các cổng ra vào của bãi xe, gồm có các thành phần như:

* Raspberry Pi: Là một máy tính nhúng thực hiện các xử lý chính của node.
* Camera: Có chức năng đọc hình ảnh trực tiếp, là dữ liệu đầu vào của hệ thống.
* Cảm biến tiệm cận: Sử dụng để phát hiện xe đang tới gần, đồng thời cũng là tín hiệu giúp hệ thống biết khi nào cần hoạt động.
* Cảm biến an toàn: Được lắp đặt trong hệ thống barrier, đảm bảo an toàn cho người dùng khi mở cổng bị lỗi.
* UHF Reader: Đầu đọc RFID ở dải tần số cao, có nhiệm vụ đọc thông tin các thẻ được gắn trên xe người dùng.

Sau khi hoàn tất các bước xử lý tại node, các thông tin như biển số xe, thời gian, địa điểm, hình ảnh chụp biển số, mã số thẻ... được đóng gói và gửi lên cloud. Cloud nhóm sử dụng với mục đích lưu trữ dữ liệu theo từng ngày cụ thể để phục vụ cho các công việc khác sau này.

### Sơ đồ hoạt động

#### Sơ đồ hoạt động chính

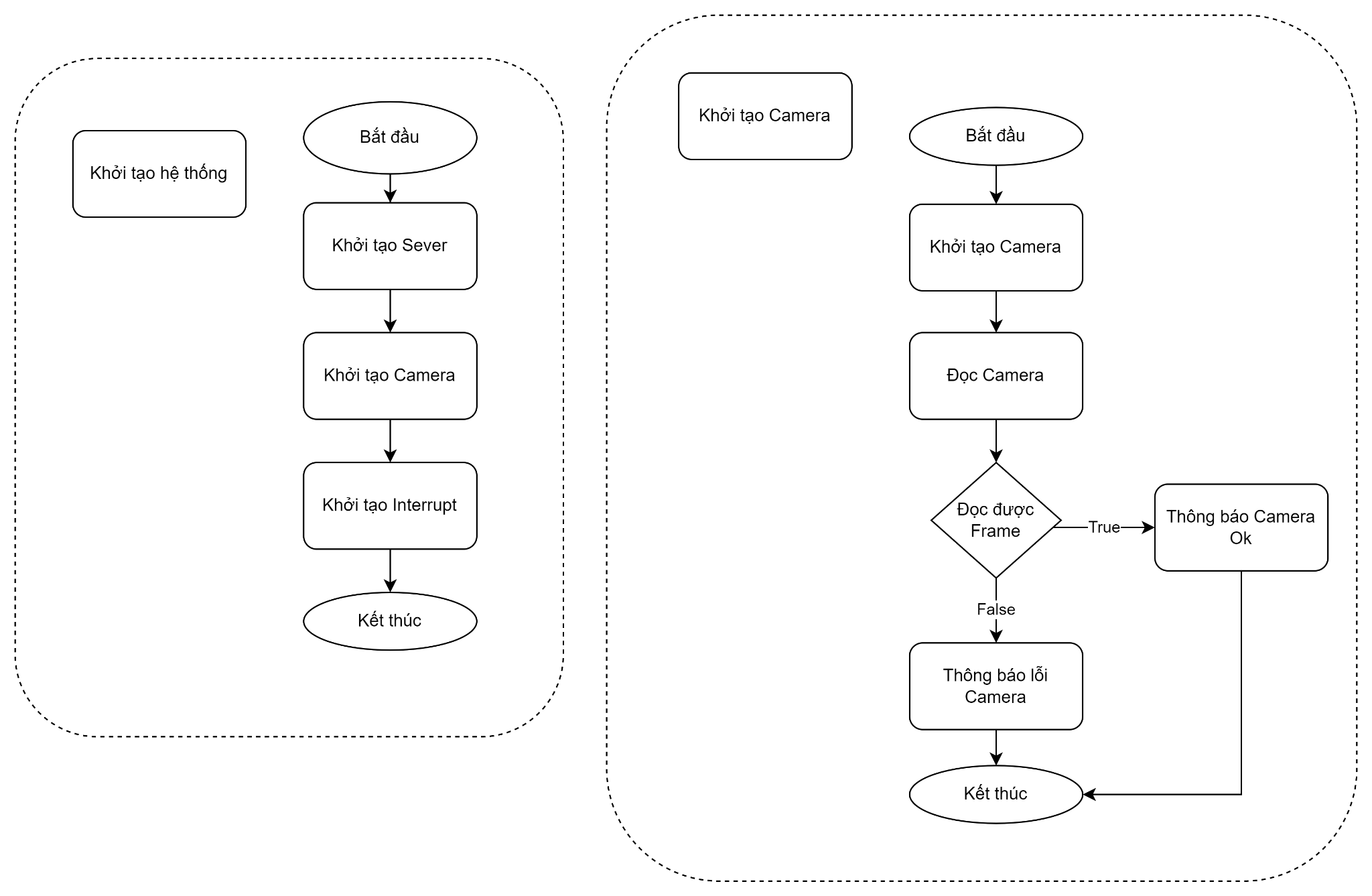
****

Hình .: Lưu đồ hoạt động của hệ thống

Sơ đồ bên trên mô tả quy trình hoạt động của hệ thống bãi giữ xe thông minh sử dụng nhận diện biển số xe.

Đầu tiên, khi hệ thống được khởi động, chương trình sẽ thực hiện các bước khởi tạo và kiểm tra các module cần thiết. Sau đó sẽ được đưa về trạng thái ngủ để tiết kiệm năng lượng. Trong trạng thái ngủ, khi có tín hiệu phát hiện xe đến từ cảm biến, hệ thống sẽ thức dậy và thực hiện trích xuất biển số xe từ hình ảnh được chụp từ camera. Sau đó sẽ đóng gói những thông tin cần thiết và gửi lên server. Khi hoàn tất, hệ thống sẽ kiểm tra nếu cập nhật thông tin lên server và tín hiệu mở cổng bật lên thì sẽ kết thúc 1 chu kỳ hoạt động, đưa hệ thống về trạng thái sleep và tiếp tục chờ interrupt.

#### Khởi tạo hệ thống

****

Hình .: Lưu đồ quá trình khởi tạo và kiểm tra hệ thống

Module khởi tạo hệ thống có nhiệm vụ khởi tạo các module như là Server, Camera và Interrupt. Đây đều là những module rất cần thiết của hệ thống, nếu thiếu một trong các module này thì hệ thống sẽ không hoạt động được. Khi khởi tạo các module này, chương trình sẽ thực hiện kiểm tra xem các thiết bị có còn hoạt động bình thường hay không, nếu có lỗi thì sẽ thông báo để người dùng có thể thực hiện kiểm tra sửa chữa.

Ví dụ như trong module khởi tạo camera được trình bày trong sơ đồ bên trên, hệ thống kiểm tra xem có đọc được frame ảnh từ camera hay không để thông báo thông tin cho người dùng.

#### Trích xuất biển số xe

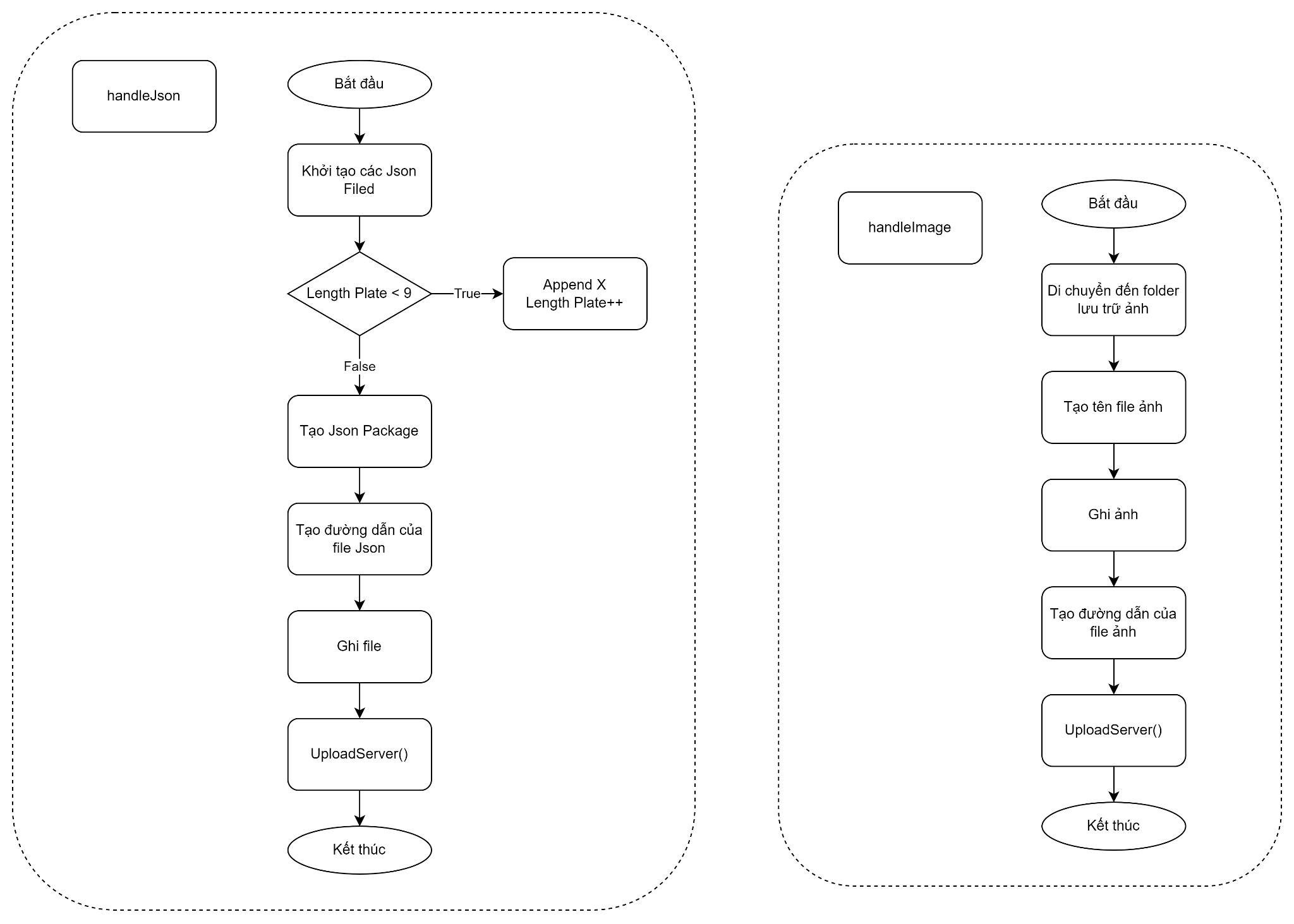


Hình .: Lưu đồ quy trình các bước trích xuất biển số xe

Module trích xuất biển số thực hiện nhiệm vụ xác định và trích xuất biển số từ hình ảnh được chụp từ camera. Đầu vào sẽ là hình ảnh chụp biển số và đầu ra sẽ là thông tin của biển số và hình ảnh chụp biển số xe. Thực hiện liên tiếp các thao tác như đọc ảnh, khử nhiễu, tìm cạnh vật thể từ đó xác định được vị trí của biển số xe trong bức ảnh. Từ khu vực biển số xe đã xác định được, ta chuyển ảnh về dạng binary và thực hiện trích xuất văn bản bằng OCR, từ đó ta có được chuỗi giá trị của biển số xe.

#### Tạo data package

Công đoạn này có nhiệm vụ tổng hợp các thông tín, định nghĩa gói tin, lưu chúng thành file để gửi lên cloud storage. Dữ liệu chính gồm có file hình ảnh chụp biển số xe và file json chứa các thông tin như biển số xe, thời gian, vị trí, hình ảnh chụp biển số, mã số thẻ,...



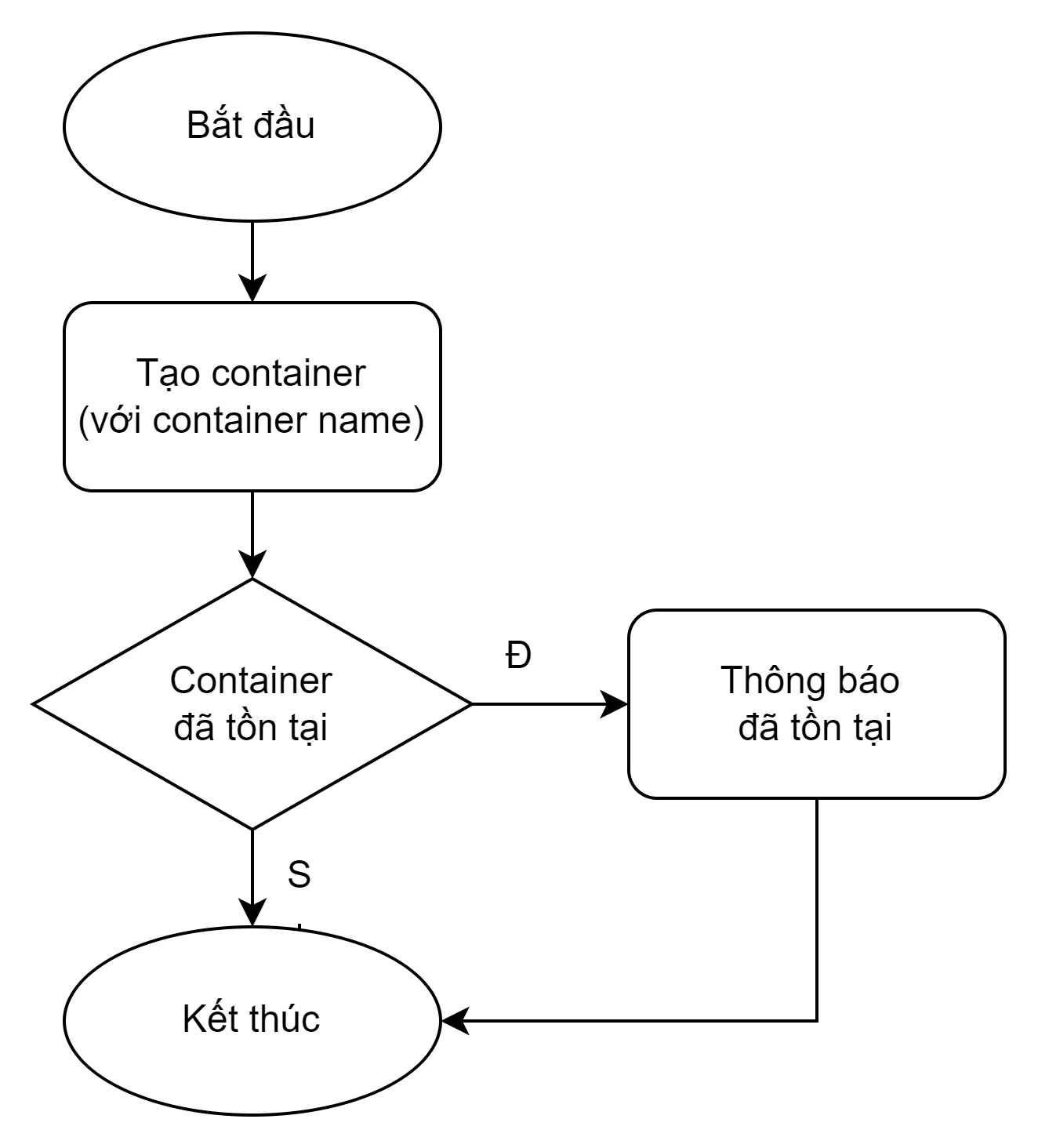
Hình .: Lưu đồ quy trình hình thành data package

Đối với file json, ta thực hiện khởi tạo các trường trong file tương ứng với các thông tin cần thiết. Tiếp đó, vì chiều dài ký tự của biển số xe là không cố định tùy theo từng loại phương tiện và có tối đa là 9 ký tự nên đối với những biển số ngắn hơn sẽ được thêm những ký tự X vào sau cho đủ 9 ký tự. Sau đó các thông tin này sẽ được ghi vào file và dùng file đó để gửi lên server.

Còn đối với file chứa hình ảnh chụp biển số xe thì hình ảnh sẽ được lưu lại ngay khi quá trình detect hoàn tất. Ảnh sẽ được lưu dưới dạng \*.jpg và tên ảnh sẽ là giá trị biển số và thời gian checkin.

#### Update Server

##### Tạo container

****

Hình .: Lưu đồ quy trình khởi tạo container

Để lưu trữ dữ liệu lên Azure Blob, server cần kết nối với Azure Blob Service, sau khi kết nối, ta sẽ nhận được một client để thao tác trên Blob Service. Tiếp đến là tạo các blob container để lưu trữ dữ liệu. Khi thực hiện tạo blob container, ta sẽ nhận lại được một client để thao tác trên container.

##### Upload file lên Server

****

Hình .: Lưu đồ quy trình upload file lên server

Sau khi đã khởi tạo thành công một container, ta có thể thực hiện lưu trữ file trên Azure Blob. Khi chúng ta gửi dữ kiệu kiểu file từ máy local lên server, server sẽ tự động tạo các thư mục tương ứng với đường dẫn chúng ta yêu cầu, từ đó hình thành thư mục lưu thông tin cụ thể.

# HIỆN THỰC VÀ KẾT QUẢ THỰC NGHIỆM

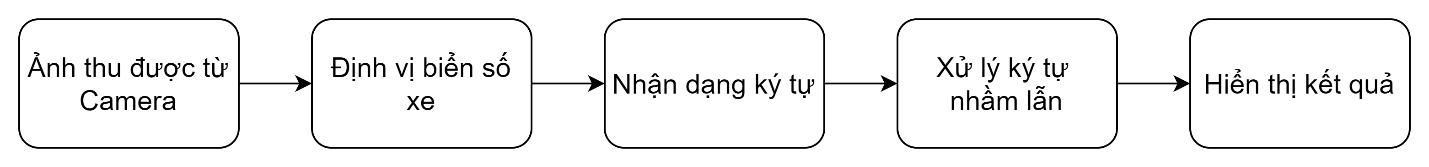
## Thuật toán nhận diện biển số xe

Với bài toán đầu vào là xây dựng hệ thống giữ xe thông minh thì công việc tất yếu ban đầu là nhận diện được biển số xe. Với mục đích giám sát cũng như thuận tiện cho việc truy xuất và tìm ra biển số xe nhanh chóng khi có sự cố xảy ra. Nên nhận diện biển số xe là phần không thể thiếu trong hệ thống giám sát xe. Công nghệ này thông qua hình ảnh để đọc biển số xe và tạo ra dữ liệu quản lý xe. Bằng cách dùng camera giám sát hoặc camera được thiết kế riêng để chụp lại hình ảnh biển số xe.

Vai trò của việc nhận diện biển số xe trong hệ thống:

* Bằng công nghệ kỹ thuật số hiện đại hệ thống nhận dạng biển số giúp thu thập và liên kết dữ liệu nhanh chóng. Hệ thống có thể tự động ghi lại thông tin về thời gian, tốc độ, hướng đi của xe. Thông qua đó cũng có thể biết đến xuất xứ của xe hay thông tin người lái xe.
* Giúp quản lý lượng xe ra vào nhanh chóng.
* Ứng dụng cao hơn của hệ thống này là tạo ra bãi giữ xe tự động. Cho phép xe tự tìm chỗ trống và đậu mà không cần vé xe hay phải mất thời gian chờ đợi trước rào chắn.

### Các bước xử lý

****

Hình .: Các bước xử lý bài toán nhận diện biển số xe

Các bước thực hiện:

* Bước 1: Thu thập dữ liệu đầu vào. Từ video camera được đặt cố định ta thực hiện đọc từng frame ảnh để làm input.
* Bước 2: Xác định vùng chứa biển số xe. Sau khi nhận được ảnh đầu vào, thực hiện một số thao tác xử lý ảnh để cắt được biển số xe
* Bước 3: Thực hiện đọc giá trị các ký tự trên biển số xe. Bước này ứng dụng trí tuệ nhân tạo để xử lý. Cụ thể là các thuật toán máy học (Machine Learning) để phân lớp dữ liệu và nhận dạng ký tự.
* Bước 4: Xử lý ký tự nhầm lẫn. Vì có thể các ký tự sau khi được chụp và định hình không rõ ràng gây nên sự nhầm lẫn. Ví dụ các ký tự dễ bị nhầm lẫn như 6 – G, 8 – B, 1 – I, ...
* Bước 5: Cuối cùng hiển thị kết quả theo định dạng biển số xe.

Nhìn vào các bước trên ta có thể thấy để giải quyết được bài toán nhận diện biển số xe thì chúng ta phải đi giải quyết ba bài toán con.

* **Bài toán thứ nhất:** Định vị vùng chứa biển số xe và tách ra được biển số.
* **Bài toán thứ hai:** Nhận dạng ký tự.
* **Bài toán thứ ba:** Xử lý ký tự nhầm lẫn.

### Định vị vùng chứa biển số xe

Hiện nay có rất nhiều cách tiếp cận để giải quyết bài toán định vị vùng chứa biển số xe từ xử lý ảnh thông thường đến sử dụng các giải thuật máy học. Mỗi cách tiếp cận đều có những ưu điểm và nhược điểm riêng. Sau đây các phương pháp mà nhóm đã thực hiện và sẽ so sánh các kết quả của mỗi phương pháp với nhau, từ đó rút ra kết luận sẽ sử dụng mô hình tính toán nào để giải quyết bài toán này.

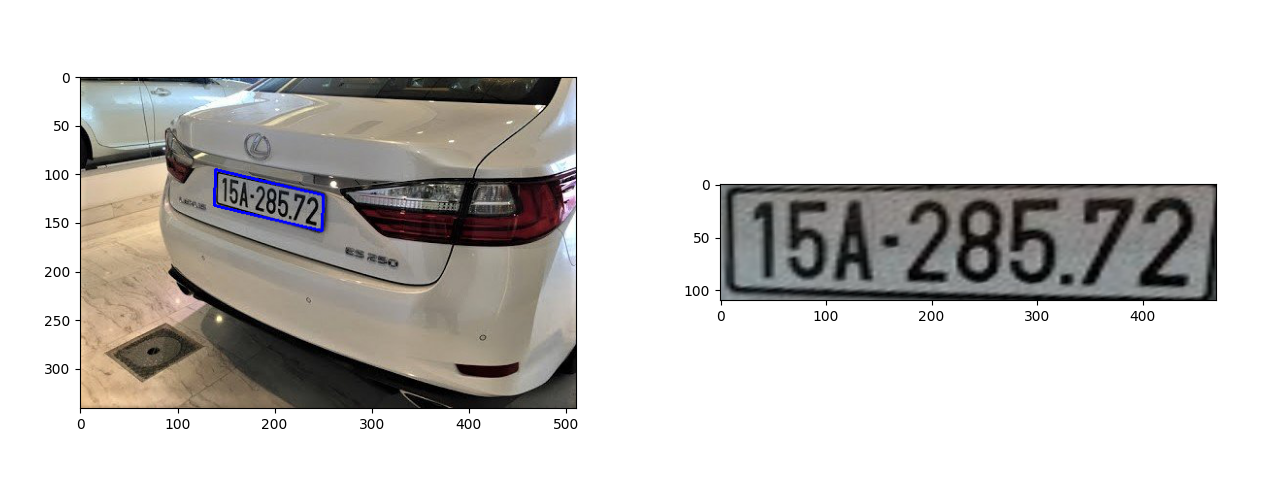
#### Sử dụng pre-trained model WPOD-NET

WPOD - Wraped Planer Object Detection Network [1]. Là mạng được thiết kế dựa trên ý tưởng của YOLO, SSD và STN (Spatial Transformer Network). Trong khi các mạng YOLO và SSD chỉ trả về một hình chữ nhật bao quanh biển số xe (bounding box) mà không quan tâm đến không gian xung quanh biển số xe là như thế nào thì WPOD có thể trả về một vùng tứ giác bao quanh toàn bộ biển số xe và đưa biển số về hướng nhìn chính diện.



Hình .: Detect biển số xe của YOLO

Đây là một pre-trained model với đầu vào là hình ảnh chứa biển số xe, 2 thông số là Dmax và Dmin đại diện cho chiều cao và chiều rộng tối đa của biển số xe. Kết quả trả về sẽ là tọa độ đường bao xung quanh biển số xe hình ảnh của biển số xe đã xác định được trong tấm ảnh đầu vào. Bất kể góc chụp của tấm ảnh như thế nào thì kết quả tấm hình biển số được cắt đều được xử lý quay về hướng chính diện. Thuận tiện cho việc quan sát và nhận diện ký tự trên biển số.



Hình .: Detect biển số xe của WPOD-NET

**Ưu điểm:** Đây là một mô hình khá tốt vì kết quả của việc xác định vùng chứa biển số xe chính xác 93% với tập dữ liệu tiêu chuẩn gồm 112 hình ảnh (biển số nền trắng, chữ đen), tốc độ xử lý khá nhanh 1 giây xử lý được 3 ảnh.

**Nhược điểm:** Vì là sử dụng pre-train model, tập dữ liệu mẫu gồm các biển số xe nước ngoài, không bao gồm biển số vuông của Việt Nam nên việc sử dụng model này tương đối bất tiện. Thêm nữa, với việc sử dụng nhiều bộ nhớ khi hoạt động xấp xỉ 350MB, không phù hợp với các ứng dụng IoT, đặc biệt là hiện thực trên phần cứng với bộ nhớ thấp như là Raspberry Pi 4.

#### Sử dụng thuần OpenCV

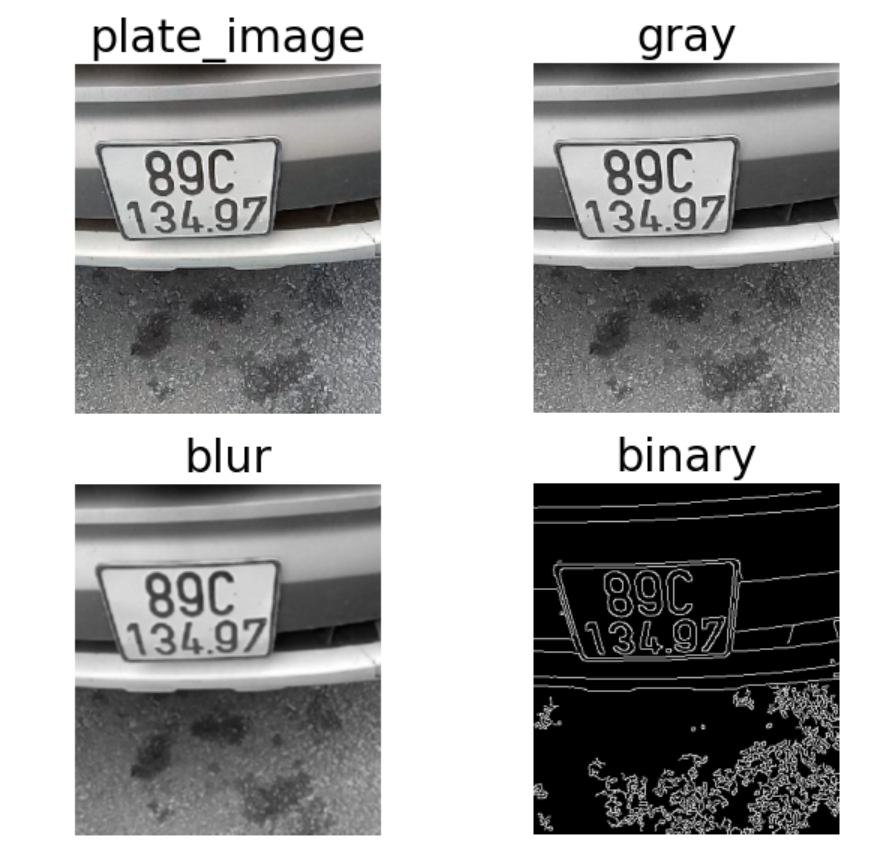
OpenCV là một thư viện được sử dụng rất phổ biến trong việc xử lý ảnh [2]. Mặc dù xử lý ảnh thông thường chịu ảnh hưởng rất nhiều từ các yếu tố bên ngoài môi trường như chất lượng của camera, độ sáng tối, biển số cũ mờ, ... Nhưng chúng ta có thể chủ động thiết lập được góc chụp của camera, tăng cường độ sáng bằng đèn led phụ nên việc sử dụng OpenCV để thực hiện phát hiện vùng chứa biển số xe là một cách tiếp cận khả thi.

Graphical user interface, text, application, chat or text message

Description automatically generated

Hình .: Quy trình phát hiện biển số bằng OpenCV

* + Bước 1: Chuyển ảnh đầu vào về ảnh ở mức xám. Mỗi pixel trong ảnh xám chỉ cần biểu diễn bằng một giá trị nguyên trong khoảng từ [0,255] thay vì (R, G, B) như trong ảnh màu. Do đó khi biểu diễn ảnh xám trong máy tính chỉ cần một ma trận là đủ. Điều này giảm dung lượng tính toán của bộ xử lý.
  + Bước 2: Giảm nhiễu bằng bộ lọc Gaussian. Sử dụng bộ lọc ảnh giúp khử nhiễu của ảnh, làm cho hình ảnh trông mượt mà hơn.
  + Bước 3: Phát hiện cạnh của vật thể. Sử dụng hàm bộ phát hiện cạnh Canny để tìm cạnh của vật thể. Đầu vào là hình ảnh đã được sử lý làm mờ khử nhiễu, đầu ra sẽ là hình ảnh nhị phân chứa cạnh của các vật thể trong hình.
  + Bước 4: Tìm và lọc vị trí bằng Contour. OpenCV cung cấp function giúp tìm các contour một cách dễ dàng. Với đặc tính nổi bật của biển số xe là hình chữ nhật có kích thước cụ thể cố định, ta dễ dàng thiết lập các thông số để tìm được đường bao của biển số xe.



Hình .: Các kết quả của giai đoạn tiền xử lý ảnh



Hình .: Biển số xe đã được cắt

### Nhận dạng ký tự

Sau khi phát hiện được biển số xe, công việc tiếp theo là thực hiện đọc các ký tự trong biển số. Cũng có nhiều phương pháp để thực hiện nhận dạng ký tự, hầu hết đều sử dụng các model đã được train sẵn với tập dữ liệu ký tự cụ thể. Vì đặc điểm của biển số xe Việt Nam khác với các biển số nước ngoài nên khi thực hiện detect sẽ gặp nhiều sự nhầm lẫn. Trong phạm vi dự án này, nhóm thực hiện nhận dạng ký tự với 3 phương pháp là sử dụng pre-train model MobileNet [3], thư viện mã nguồn mở EasyOCR [4] và Pytesseract [5, 6].

#### Pre-train model MobileNet

##### Tiền xử lý hình ảnh biển số

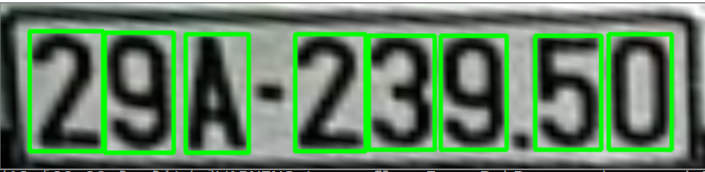
Với hình ảnh đầu vào là hình ảnh biển số xe đã tìm được, chúng ta thực hiện những thao tác tiền xử lý như làm mờ, khử nhiễu, chuyển ảnh sang dạng Binary.



Hình .: Tiền xử lý ảnh biển số xe

##### Định vị ký tự

Ta thực hiện tìm các contour của từng ký tự riêng lẻ trên ảnh binary, sau đó, từ những contour đó ta tính được chiều cao chiều rộng của ký tự và thực hiện cắt từng ký tự để làm đầu vào cho bước tiếp theo.



Hình .: Các ký tự đã được xác định



Hình .: Các ký tự đã được cắt ra và chuyển về dạng Binary

##### Kết quả

Logo

Description automatically generated

Hình .: Kết quả dự đoán các ký tự

Ta thấy rằng kết quả chỉ tương đối chính xác. Vẫn còn những ký tự gần giống nhau sẽ gây ra sai sót trong quá trình tính toán.

* + **Ưu điểm:** Ta có thể tự thiết kế lại bộ dữ liệu train cho phù hợp với hoàn cảnh hiện tại. Tốc độ tính toán nhanh, tiêu thụ bộ nhớ ít.
  + **Nhược điểm:** Khó cài đặt và triển khai, độ chính xác đạt mức trung bình và còn nhiều ký tự bị nhầm lẫn.

#### Thư viện easyocr

EasyOCR là một dự án OCR Python nguồn mở cho phép các nhà phát triển thị giác máy tính dễ dàng thực hiện Nhận dạng ký tự quang học, với hơn 80 ngôn ngữ, bao gồm tiếng Trung, Nhật, Hàn, Thái,... Việc cài đặt thư viện EasyOCR rất dễ dàng chỉ bằng một lệnh pip và nhập vào dự án bằng lệnh import.

Đầu vào sẽ là hình ảnh nhị phân chứa biển số xe, kết quả là một danh sách chứa giá trị bounding box của chuỗi ký tự, các ký tự trong biển số xe và độ chính xác tương ứng.

reader = easyocr.Reader(['en'])

result = reader.readtext(binary)

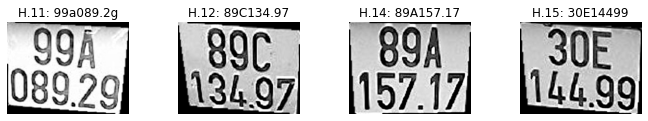
**Kết quả:** Với tập dữ liệu 112 tấm ảnh, ta có kết quả như sau:

Total: 112 Plate

Predict True: 37 Plate

Predict False: 75 Plate

Accuracy: 33.04%

****

Hình .: Một số ví dụ kết quả đọc ký tự sử dụng EasyOCR

* **Ưu điểm:** Kết quả dự đoán khá chính xác, dễ dàng cài đặt.
* **Nhược điểm:** Tiêu thụ bộ nhớ cao, thời gian thực thi lâu, khó ch ỉnh sửa theo nhu cầu của cá nhân.

#### Thư viện pytesseract

Tương tự như EasyOCR, Tesseract là thư viện OCR nổi tiếng do độ chính xác cao hơn hẳn các thư viên khác, cách cài đặt cũng rất đơn giản. Tesseract cung cấp nhiều options config giúp người dùng dễ dàng sử dụng với những tình huống cụ thể.

#Tesseract

text = pytesseract.image\_to\_string(binary, *config*='--psm 6 --oem 3 -c tessedit\_char\_whitelist=ABCDEFGHKLMNPSTUVXYZ0123456789')

text2 = ''.join(filter(str.isalnum, text))

**Kết quả:** Với tập dữ liệu 112 tấm ảnh tương tự như bên trên, ta có kết quả như sau:

Total: 112 Plate

Predict True: 79 Plate

Predict False: 33 Plate

Accuracy: 70.54%

* **Ưu điểm:** Kết quả dự đoán chính xác cao, thời gian thực thi nhanh, dễ dàng cài đặt, có nhiều phương thức chỉnh sửa cho giải thuật.
* **Nhược điểm:** Tiêu thụ bộ nhớ cao.

#### Kết luận

Sau khi hiện thực cả 3 giải thuật OCR trên, nhóm đã lập bảng so sánh các tiêu chí của giải thuật OCR trên cùng một tập dữ liệu mẫu gồm 112 ảnh như sau:

Bảng : Bảng so sánh giữa 3 giải thuật OCR

|  |  |  |  |
| --- | --- | --- | --- |
| **Tiêu chí so sánh** | **MobileNet** | **EasyOCR** | **PyTesseract** |
| Cài đặt thuật toán | Khó | Dễ | Dễ |
| Thời gian (s) | 85 | 109 | 78 |
| Bộ nhớ (Mb) | 1262.2 | 503.2 | 454.3 |
| Độ chính xác (%) | 12.8 | 33.04 | 70.54 |

### Xử lý ký tự nhầm lẫn

Đối với bài toán nhận dạng ký tự thì sự nhầm lẫn giữa những ký tự tương đồng nhau là điều không thể tránh khỏi. Biển số xe Việt Nam có quy luật sắp đặt các ký tự cụ thể nên việc xử lý những ký tự lỗi trở nên dễ dàng hơn. Ví dụ, với biển số xe 94H-400.95, thì 2 ký tự đầu luôn luôn là số, ký tự thứ 3 luôn luôn là chữ cái in hoa, ký tự thứ 4 có thể là chữ cái in hoa hoặc số, còn lại đều là số. Nếu những ký tự ở vị trí số mà kết quả nhận diện ra chữ thì chúng ta thực hiện chuyển đổi thành số tương đồng chúng. Tương tự với những ký tự còn lại.

Để xác định những ký tự tương đồng nhau dễ gây nhầm lẫn, nhóm đã thực hiện một bảng thống kê những sai số có thể xảy ra trong quá trình thực hiện như sau:

Bảng : Bảng biểu thị tần số nhầm lẫn giữa các số và ký tự

|  |  |  |
| --- | --- | --- |
| **Ký tự thực tế** | **Ký tự nhầm lẫn** | **Số lần nhầm lẫn** |
| G | 6 | 17 |
| A | 4 | 6 |
| E | 5 | 2 |
| Z | 7 | 2 |
| 8 | B | 2 |
| 2 | Y | 1 |
| 3 | B | 1 |
| 4 | G | 1 |
| 4 | L | 1 |
| 9 | G | 1 |
| Z | 2 | 1 |
| E | 6 | 1 |

Như vậy, ta thấy rằng số 6 và chữ G có sự dự đoán nhầm lẫn cao nhất, tiếp theo là số 4 với chữ A. Nên khi chúng ta phát hiện những nhầm lẫn khi nhận diện ký tự, ta sẽ dựa vào bảng trên để thực hiện chuyển đổi về đúng với dạng của chúng.

## Xây dựng Server

### Cấu trúc gói tin lưu trữ lên Server

Sau khi phát hiện và nhận dạng được biển số xe, Raspberry Pi tiến hành cấu trúc lại các dữ liệu, đóng gói thành json file và lưu dữ liệu lên Server.

Các thông tin được lưu trữ lên Server bao gồm:

|  |  |  |  |
| --- | --- | --- | --- |
| **time**  **(20 bytes)** | **Type**  **(3 bytes)** | **DeviceID**  **(4 bytes)** | **BSX HEX String**  **(9 bytes)** |

Gói tin được lưu trữ lên Server có dạng json, cấu tạo của gói tin như sau:

{

"time": "dd/mm/yyyy, HH:MM:SS",

"type": "CAM"

"deviceID": "1234",

"data": "XXXXXXXXX"

}

*Trong đó:*

- Trường “time”: Là thời gian nhận diện được biển số xe với định dạng ngày giờ đầy đủ như trên.

- Trường “type”: Là kiểu định danh nguồn gửi gói tin (để phân biệt với UHF).

- Trường “deviceID”: Là mã định danh của camera tại mỗi cổng.

- Trường “data”: Là chuỗi biển số xe đã nhận diện được. Vì trường hợp biển số xe có số ký tự lớn nhất là 9 nên trường này sẽ có 9 bytes. Trường hợp nếu biển số xe nhận diện được có số ký tự bé hơn 9 thì được thêm ký tự X vào cuối dãy để tạo thành chuỗi 9 ký tự.

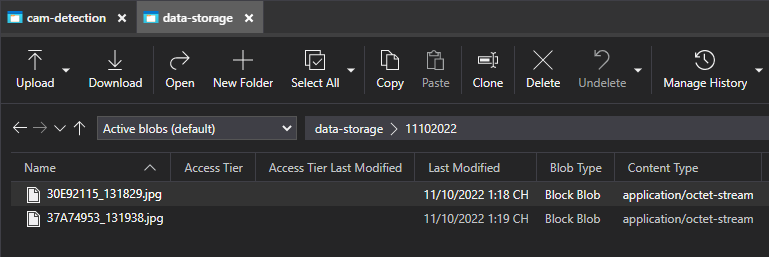
### Gửi gói tin lên Server

Gói tin json sau khi gửi lên server sẽ được lưu trữ trong blob container cam-detection, hình ảnh chụp vào thời điểm nhận diện được biển số xe sẽ được lưu trữ trong blob data-storage, trong thư mục đặt tên theo ngày nhận được. Tên file là biển số xe + thời gian xe vào bãi.

Ảnh có chứa văn bản, màn hình, đen, ảnh chụp màn hình

Mô tả được tạo tự động

Hình .: File json được lưu trữ trong Azure Blob Storage



Hình 3.: File jpg được lưu trữ trong Azure Blob Storage

Mở file được lưu trữ lên Server, ta có thể xem được nội dung file json, với các thông số như đã mô tả trong cấu trúc của gói tin cũng như xem được ảnh chụp khi nhận diện được biển số.

Ảnh có chứa văn bản

Mô tả được tạo tự động

Hình .: Nội dung file json được lưu trữ trên Azure Blob Storage

Ảnh có chứa văn bản, trong nhà

Mô tả được tạo tự động

Hình .: Hình ảnh được lưu trữ trên Azure Blob Storage

## Kết quả thực nghiệm

Bằng cách sử dụng phương pháp nhận diện biển số xe và cách thức lưu trữ dữ liệu trên server như đã trình bày ở trên, nhóm đã hiện thực và đạt được kết quả như mục tiêu đã đề ra ban đầu. Hệ thống có khả năng quan sát và nhận diện được biển số xe ở Việt Nam với tỉ lệ chính xác xấp xỉ 80%, tổ chức dữ liệu một cách trực quan với các thông tin cơ bản như biển số xe, thời gian ra vào cổng, vị trí của cổng và hình ảnh chụp được của biển số xe. Đồng thời cũng thực hiện được việc cập nhật và lưu trữ dữ liệu từ các thiết bị local lên trên máy server để phục vụ cho các công việc sau này cần dùng đến dữ liệu đó. Hệ thống hoạt động trơn tru, mượt mà, hoàn toàn có thể lắp đặt và sử dụng trong thực tế.

Video demo bên dưới mô tả các bước từ khi phát hiện xe vào bãi đỗ xe, mở camera nhận diện được biển số xe và lưu các dữ liệu cùng hình ảnh lên server.

Link video: [Video demo nhận dạng biển số xe sử dụng Raspberry Pi | TK25 Inc.](https://youtu.be/jvS0UlAhumA)

# KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

## Kết luận

### Ưu điểm

* Chi phí rẻ, không cần tốn thêm bất kỳ chi phí nào cho từng xe hoặc tài xế ví dụ như thẻ RFID.
* Tốc độ đáp ứng nhanh, phù hợp với yêu cầu hệ thống.
* Ít bị nhiễu từ môi trường vì hệ thống camera có thể được thiết kế để tránh nhiễu, phù hợp với môi trường thực tế.
* Dễ dàng sử dụng và lắp đặt hệ thống,

### Nhược điểm

* Một số biển số xe quá cũ, bẩn hoặc các nhân tố khác trên biển số làm biển số trở nên khác với chất lượng bình thường gây khó khăn cho việc nhận diện biển số.
* Đồng thời chất lượng của việc nhận diện biển số vẫn còn phụ thuộc vào các yếu tố bên ngoài môi trường như hướng chụp, góc chụp, độ sáng,...

## Hướng phát triển

* Định hướng tương lai sẽ phát triển hệ thống đầy đủ nhất, dần dần hoàn thiện hệ thống hiện tại và tìm hiểu, nghiên cứu để xây dựng những ứng dụng khác trên hệ thống. Các công việc sẽ thực hiện trong tương lai như tích hợp cảm biến phát hiện vật thể để phát hiện xe đang tiến vào để bắt đầu thực hiện việc nhận diện biển số, điều này giúp giảm thời gian hoạt động của hệ thống và tiết kiệm năng lượng.
* Tiếp đến là hiện thực hệ thống thực hiện việc điều hướng, thông báo cho tài xế biết những bãi đỗ xe nào còn trống và vị trí còn trống trong bãi đỗ xe.
* Sau này có thể sử dụng các thông tin mà chúng ta đã thu thập được để xây dựng các ứng dụng máy học, phục vụ cho việc phân tích và hỗ trợ tài xế đậu xe theo thói quen hoặc gợi ý nơi đậu xe thuận tiện nhất.

# TÀI LIỆU THAM KHẢO

|  |  |
| --- | --- |
| [1] | S´ergio Montazzolli Silva, Cl´audio Rosito Jung, "License Plate Detection and Recognition in Unconstrained Scenarios," Institute of Informatics - Federal University of Rio Grande, do Sul Porto Alegre, Brazil, 2018. |
| [2] | I Culjak, Abram, Pribanic, Dzapo and Cifrek, "A brief introduction to OpenCV," 21-25 May 2012. [Online]. Available: https://ieeexplore.ieee.org/document/6240859. [Accessed 22 8 2022]. |
| [3] | Q. Nguyen, "github.com," 29 Dec 2020. [Online]. Available: https://github.com/quangnhat185/Plate\_detect\_and\_recognize. [Accessed 22 8 2022]. |
| [4] | JaidedAI, "github.com," 2021. [Online]. Available: https://github.com/JaidedAI/EasyOCR. |
| [5] | "github.com," 04 Jun 2016. [Online]. Available: https://github.com/sirfz/tesserocr. [Accessed 1 9 2022]. |
| [6] | R. Smith, "An Overview of the Tesseract OCR Engine," Google Inc., 2012. |