**TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN**

**KHOA KHOA HỌC MÁY TÍNH**

****

**BÁO CÁO MÔN TÍNH TOÁN ĐA PHƯƠNG TIỆN**

**CHỦ ĐỀ: NHẬN DIỆN BẢNG SỐ XE**

**Lớp: CS332.J11**

**Sinh viên thực hiện: Lý Bảo Khang - 15520343**

**Mai Quốc Kiệt – 15520400**

**Lê Thiện Duy – 15520158**

Mục lục

[Chương 1: Giới thiệu về đồ án 3](#_Toc531867192)

[1. Mục tiêu bài toán: 3](#_Toc531867193)

[2. Ý tưởng bài toán: 3](#_Toc531867194)

[Chương 2: Phát hiện bảng số xe 4](#_Toc531867195)

[1. Tiền xử lý ảnh: 4](#_Toc531867196)

[2. Tìm vị trí có thể là ký tự trong ảnh: 4](#_Toc531867197)

[3. Gom nhóm các contour: 5](#_Toc531867198)

[4. Trích xuất bảng số xe: 6](#_Toc531867199)

[5. Cắt ảnh bảng số đã xác định: 7](#_Toc531867200)

[Chương 3: Nhận diện ký tự trong bảng số xe 8](#_Toc531867201)

[1. Huấn luyện (training): 8](#_Toc531867202)

[2. Nhận diện (detect): 8](#_Toc531867203)

[Chương 4: Kết quả thực hiện và nhận xét 9](#_Toc531867204)

[1. Kết quả thực hiện: 9](#_Toc531867205)

[2. Nhận xét: 12](#_Toc531867206)

[3. Hướng phát triền: 12](#_Toc531867207)

# Chương 1: Giới thiệu về đồ án

1. Mục tiêu bài toán:

Input: một tấm ảnh có bảng số xe hơi phía trước ở Việt Nam, có nền trắng chữ đen

Output: một tấm ảnh chỉ chứa bảng số xe đã cắt và các ký tự trên bảng số đó

1. Ý tưởng bài toán:

Bài toán sẽ được thực hiện thông qua hai bước chính:

Phát hiện bảng số xe trong bức ảnh (License Plate Detection)

Nhận diện ký tự trong bảng số xe (Letters and Digits Recognition) bằng việc áp dụng các phương pháp máy học như KNN, SVM

# Chương 2: Phát hiện bảng số xe

Mục tiêu của phát hiện bảng số xe là nhận diện một khung hình chữ nhật có chứa các ký tự trên bảng số xe, để sau đó thực hiện bước tiếp theo là nhận diện các ký tự trên bảng số xe đã được lấy ra. Phần này được thực hiện thông qua các bước: tiền xử lý ảnh, tìm vị trí có thể là ký tự trong ảnh, gom nhóm các contours, trích xuất bảng số xe và cắt ảnh bảng số đã xác định.

1. Tiền xử lý ảnh:

Trước khi thực hiện các thao tác trên ảnh, ta cần phải xử lý ảnh qua các bước:

Đưa ảnh về ảnh xám (grayscale)

Tăng độ tương phản (contrast) lên tối đa

Áp dụng bộ lọc Gaussian Blur để làm mịn ảnh

Đặt một phân ngưỡng ảnh để chuyển về thành ảnh trắng đen thông qua hàm cv2.adaptiveThreshold()

Kết quả của quá trình này sẽ cho ra một ảnh grayscale và đã được phân ngưỡng:

A car parked in a parking lot

Description automatically generated A close up of a logo

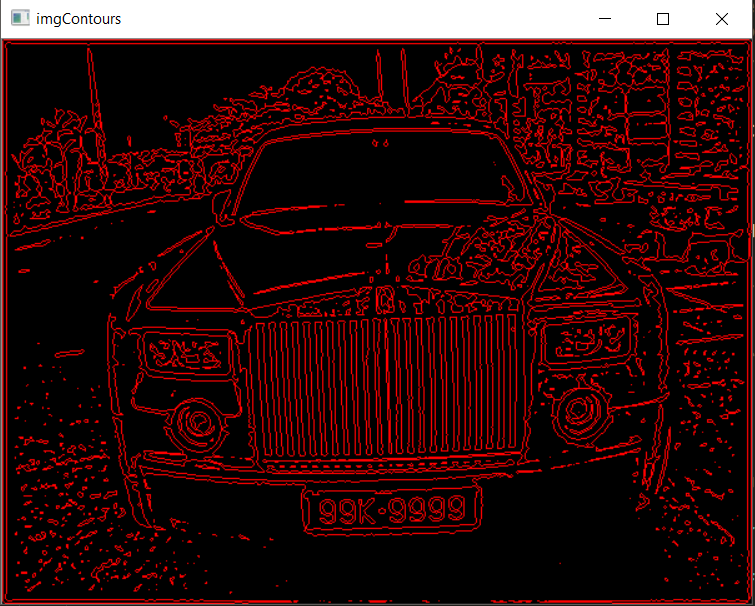
Description automatically generated

*Ảnh trước và sau khi tiền xử lý ảnh*

1. Tìm vị trí có thể là ký tự trong ảnh:

Ta thực hiện thông qua các bước như sau:

Lấy tất cả contours có trong ảnh đã qua tiền xử lý, contour là viền của đối tượng trong ảnh.



*Kết quả sau khi lấy tất cả các contour*

Xét từng contour trong tất cả các contour đã lấy để xem nó có phải là ký tự hay không. Contour sẽ được xét thông qua các thông số tự đặt như: chiểu cao (height), chiều rộng (width), thỉ lệ width /height (từ 0.25 cho chữ I đến 1 cho chữ O), diện tích nhỏ nhất.

Sau đó ta append những contour có khả năng là ký tự vào một list, mỗi một contour sẽ được lưu dưới dạng một kiểu dữ liệu tự định nghĩa PossibleChar.

Class PossibleCharbao gồm các đối tượng sau:

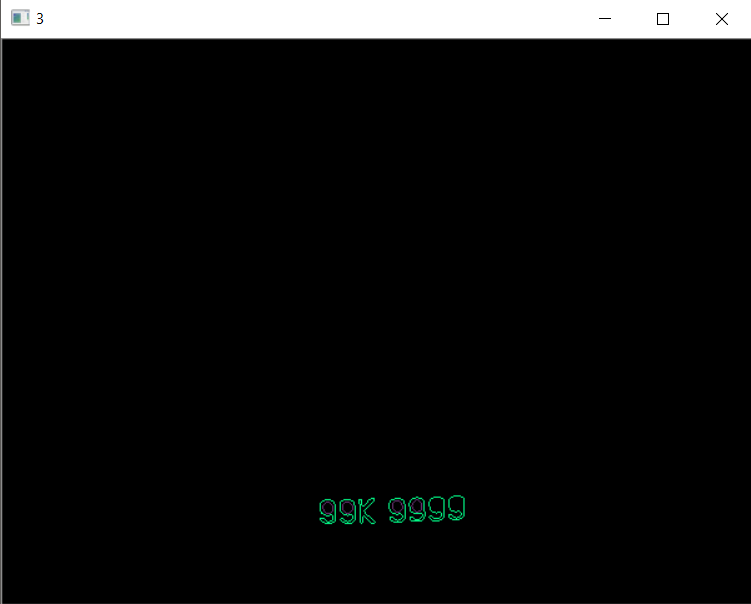
Contour đang xét

Một hình chữ nhật bao quanh contour đó (tọa độ x, y của điểm trên cùng bên trái, width, height, tọa độ Center của hình chữ nhật đó, đường chéo và tỉ lệ giữa width và height của hình chữ nhật)

1. Gom nhóm các contour:

Dựa vào list chứa các đối tượng đó, ta duyệt và gom nhóm các contour bằng cách: so sánh một contour với từng contour còn lại trong list.

Nếu contour đang xét và contour đang kiểm tra nhỏ hơn threshold được đưa ra, có khoảng cách gần nhau, có góc không vượt quá 12 độ (góc được tính bằng góc giữa hai center của hai contour thông qua công thức arctan2) và diện tích hình chữ nhật bao quanh tương tự nhau ta sẽ gom lại thành một nhóm.

*Kết quả sau khi gom nhóm các contour*

1. Trích xuất bảng số xe:

Trong mỗi nhóm đã có ở bước trước, ta thực hiện như sau:

Sắp xếp theo x của từng PossibleChar (x là hoành độ điểm trên cùng bên trái của hình chữ nhật bao quanh contour đó).

Xóa bỏ các contour bị chồng lên nhau.

Ta tính center của từng nhóm bằng cách lấy tọa độ (x,y) của phần tử đầu + (x,y) của phần tử cuối của nhóm, sau đó chia cho 2.

Tính width của của từng nhóm controur: [hoành độ x của phần tử cuối cùng trong nhóm) + độ rộng của phần tử cuối cùng trong nhóm – hoành độ x phần tử đầu tiên trong nhóm ]\* 1.1 (tăng width lên 10%).

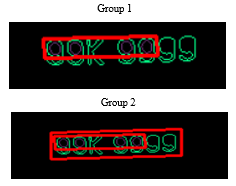
Tính height của từng nhóm contour: trung bình cộng độ cao của từng phần tử \* 1.1 (tăng height lên 10%).

Ta tăng width, height lên 10% để tránh làm mất các pixel ở ngoài cùng khi cắt bảng số.

Trước khi cắt, ta cần kiểm tra xem bảng số xe có nằm ngang theo trục Ox hay không để khi cắt cho kết quả chính xác. Nếu biển số xe không nằm ngang, ta sẽ xoay lại cho đúng theo chiều ngang. Bước này được thực hiện bằng cách:

Tính góc của đường thẳng được tạo bởi center của phần từ đầu và center của phần tử cuối với trục Ox,

Xoay bảng số theo góc đã tính nếu bảng số không nằm ngang



*Kết quả sau khi trích xuất bảng số xe*

1. Cắt ảnh bảng số đã xác định:

Sau khi đã có các nhóm ở bước trên, ta tiến hình cắt ảnh theo các nhóm đó:

A close up of a logo

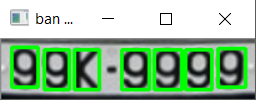
Description automatically generated A black sign with white text

Description automatically generated

*Cắt ảnh theo các nhóm*

Sau khi cắt, ta sẽ xét số lượng các contour trong nhóm có đúng với số lượng ký tự trong bảng số ở Việt Nam hay không (7 hoặc 8 ký tự) và loại bỏ các nhóm có số lượng không đúng. Kết quả là nhóm thứ nhất bị loại bỏ.

Cuối cùng ta cắt từng contour bên trong biển số đã phát hiện để chuẩn bị cho nước nhận diện ký tự trên bảng số:



*Cắt từng contour*

# Chương 3: Nhận diện ký tự trong bảng số xe

1. Huấn luyện (training):

Chuẩn bị dữ liệu huấn luyện:

Sử dụng dataset với 36 ký tự (26 chữ cái và 10 chữ số latin), mỗi ký tự 1016 ảnh với kích thước 60 x 60

Để tránh nhầm lẫn giữa chữ và số, nhóm train chữ và số riêng biệt.

Sử dụng phương pháp SVM và thư viện hỗ trợ sklearn.

Tiến hành huấn luyện:

Đọc từng ảnh, sau đó ta chuyển về ảnh xám và đưa vào một list train.

Tạo list bao gồm các label tương ứng với từng ảnh trong list train ở trên

Khởi tạo SVM bằng hàm LinearSVC()

Train dữ liệu đã tạo bằng hàm fit(list train, list label)

1. Nhận diện (detect):

Sau khi đã huấn luyện xong, ta tiến hành detect cho từng contour đã được cắt:

Load model đã train bằng hàm load()

Resize các contour đã cắt về cùng kích thước với ảnh trong dataset (60x60)

Reshape ảnh thành ma trận (1, x)

Detect từng ký tự thông qua dataset đã train bằng hàm predict()

# Chương 4: Kết quả thực hiện và nhận xét

1. Kết quả thực hiện:

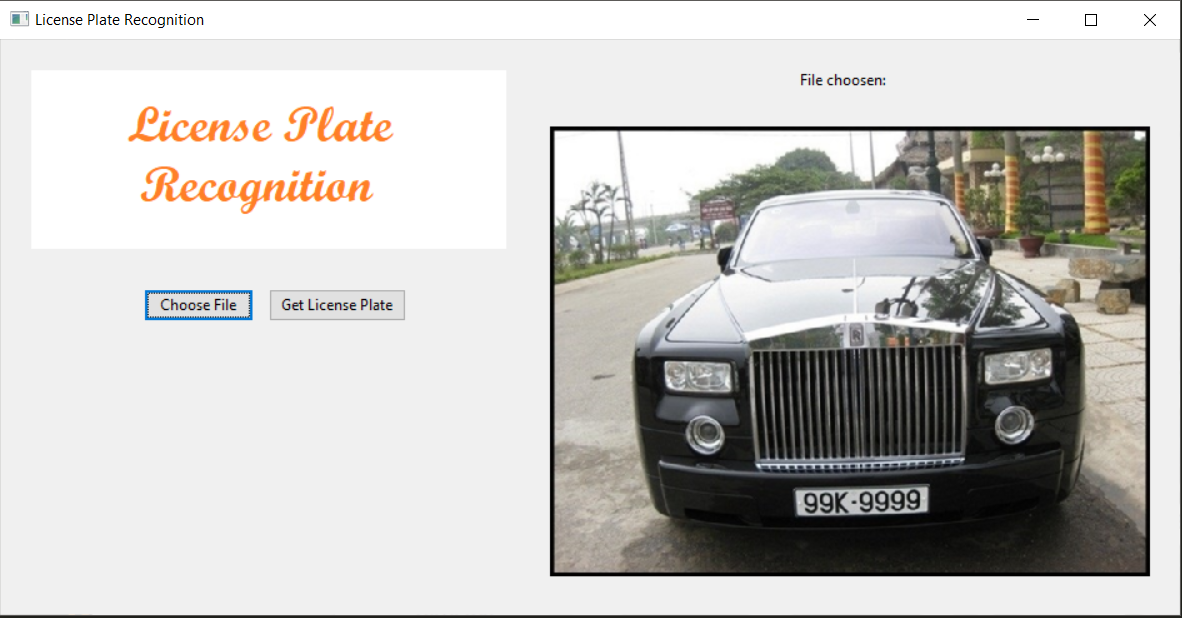
Giao diện của chương trình:



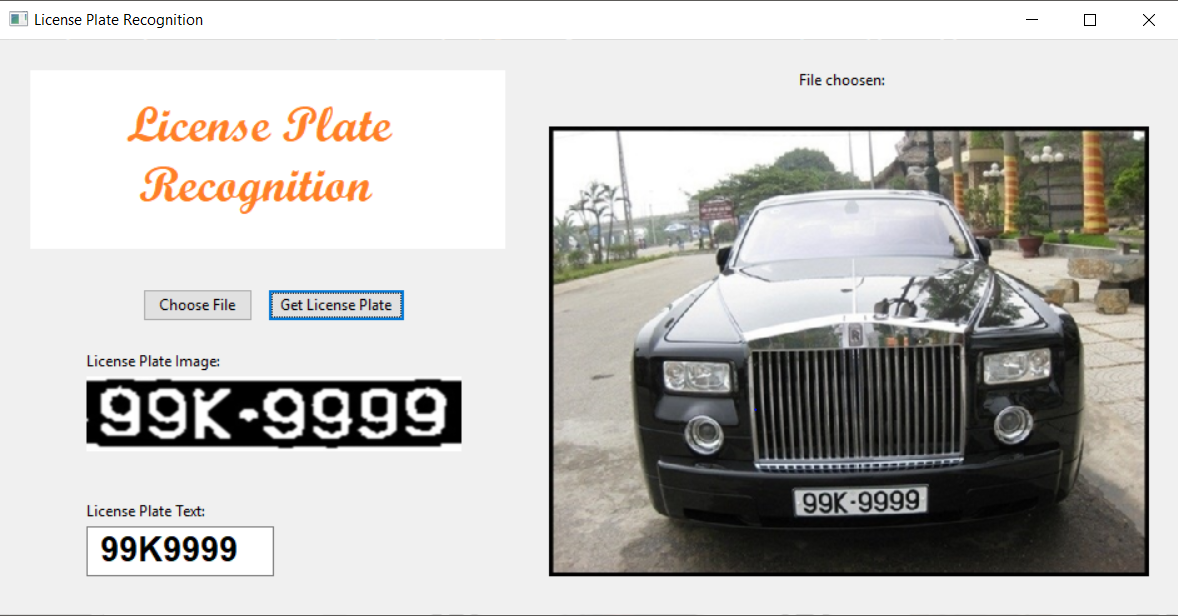
Hướng dẫn sử dụng:



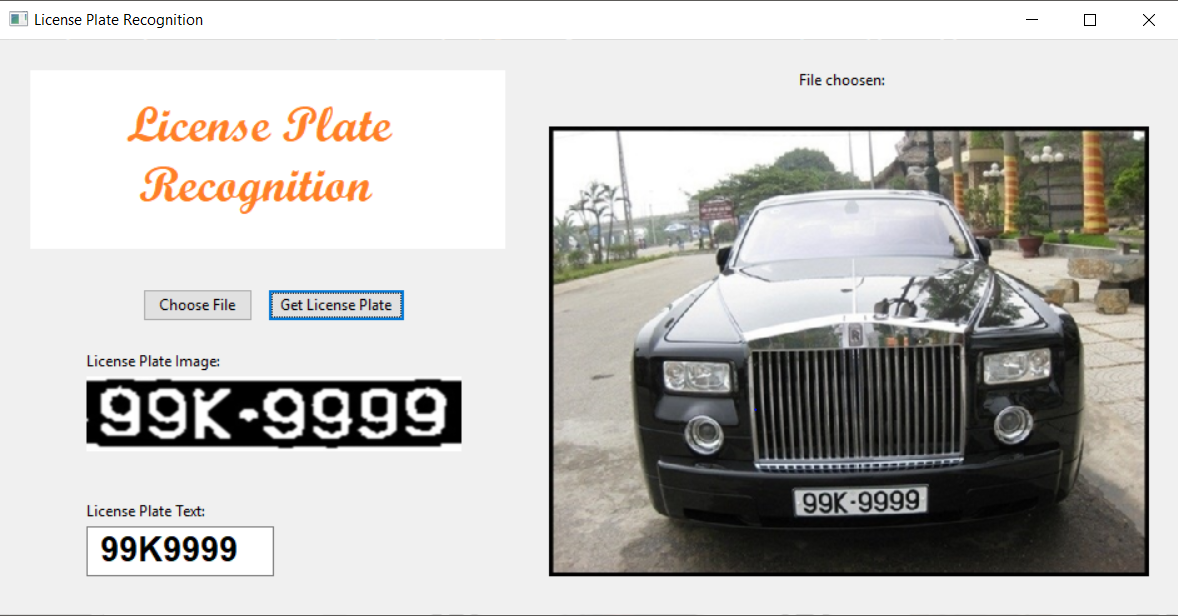
Ở giao diện chính của chương trình, ta chọn Choose File để chọn file hình cần nhận diện biển số:

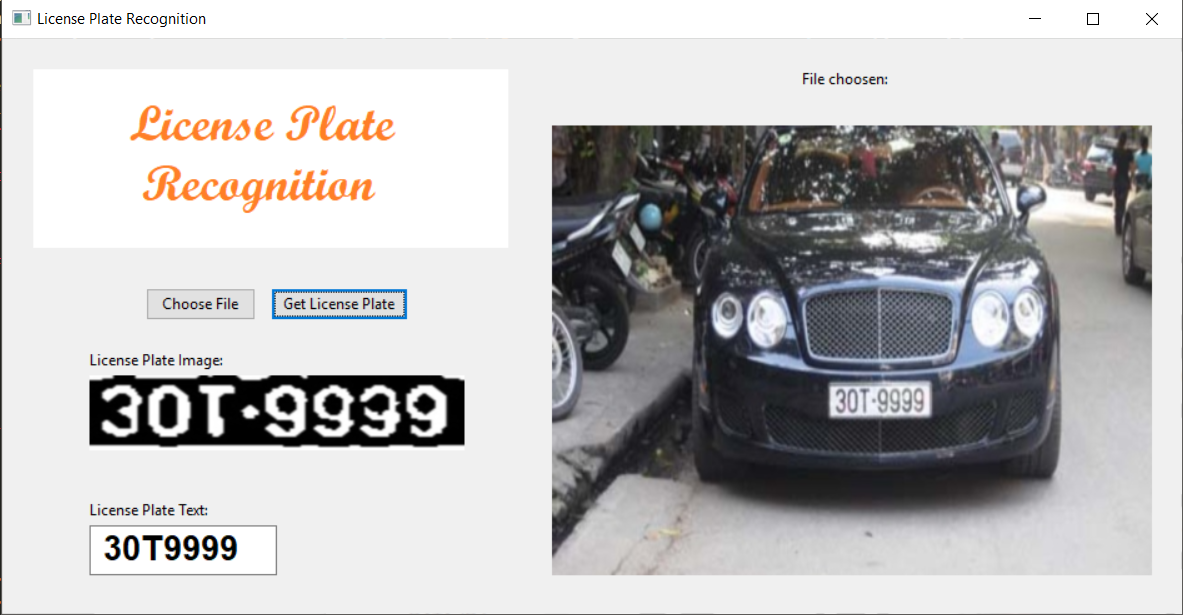


Sau đó ta bấm Get License Plate để thực hiện nhận diện bảng số xe:



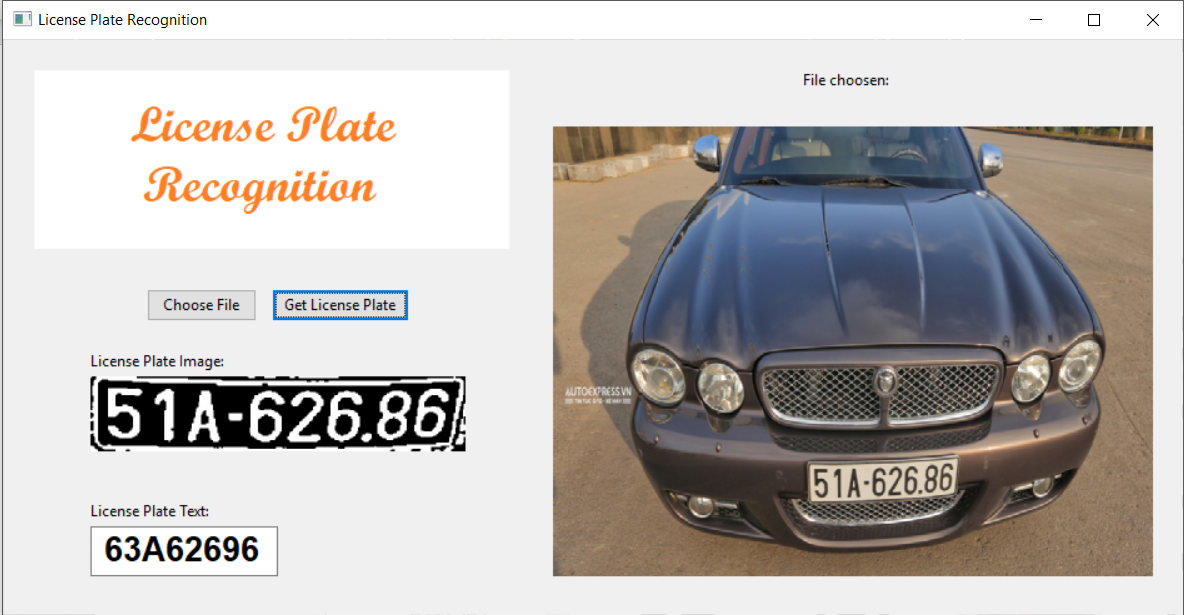
Các kết quả chính xác:



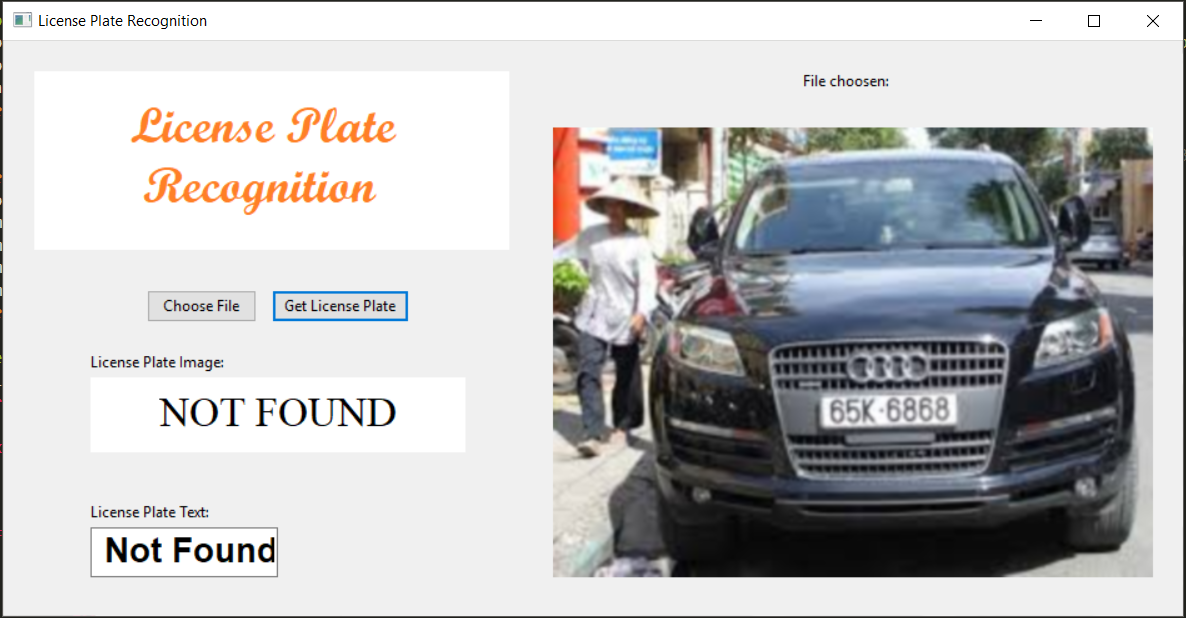


Các kết quả có sai một số ký tự:





Các kết quả mà không phát hiện được bảng số xe:



1. Nhận xét:

Tỉ lệ sai sót khá cao, do các contour được cắt ra có dạng hình chữ nhật, còn dataset có các hình ảnh có kích thước 60x60, nên khi reshape contour về giống như ảnh trong dataset sẽ dẫn đến sai sót.

Nhóm đã thử các phương pháp như SVM, KNN và rút trích đặc trưng bằng VGG-16 nhưng SVM cho kết quả tốt nhất.

1. Hướng phát triền:

Áp dụng phương pháp khác tốt hơn cho việc nhận diện ký tự trên bảng số xe.

Tự tạo một dataset có dạng hình chữ nhật để phù hợp với các contour được cắt, giảm tỉ lệ sai sót khi nhận diện.