**TRƯỜNG ĐẠI HỌC PHENIKAA**

**KHOA CÔNG NGHỆ THÔNG TIN**



**ĐỒ ÁN CƠ SỞ**

TÌM HIỂU VÀ THỰC HIỆN NHẬN DIỆN BIỂN SỐ XE

**Giảng viên hướng dẫn: LƯƠNG VĂN THIỆN**

**Sinh viên thực hiện: Tống Minh Khang(20010744)**

**Khoá: K14 – 2021 - 2024**

**Ngành/ chuyên ngành: Công nghệ thông tin**

Hà nội, tháng 11 năm 2023

**LỜI CẢM ƠN**

Chúng tôi xin cảm ơn TS. Lương Văn Thiện đã tận tình hướng dẫn, định hướng và chia sẻ các nguồn tài nguyên cũng như công cụ để thực hiện dự án, giúp đỡ tôi hoàn thành đồ án liên ngành nhận diện biển số xe.

# LỜI CAM ĐOAN

Tôi cam đoan Đồ án liên ngành là sản phẩm trí tuệ của tôi. Mọi thông tin, dữ liệu, hình ảnh, etc. được sử dụng từ các nguồn khác đều được trích dẫn đầy đủ và có thể tìm thấy các tài liệu liên quan thông qua mục tài liệu tham khảo.

Tôi xin chịu trách nhiệm hoàn toàn về nội dung của Đồ án liên ngành mà tôi đã nộp.

Hà nội, ngày … tháng … năm

Sinh viên thực hiện

(ký tên)



| TRƯỜNG ĐẠI HỌC PHENIKAA | **KỲ THI KẾT THÚC HỌC PHẦN**  **HỌC KỲ I NĂM HỌC 2023 - 2024** |
| --- | --- |

**PHIẾU CHẤM THI TIỂU LUẬN/ĐỒ ÁN**

Môn học: Đồ án liên ngành

Sinh viên thực hiện: Tống Minh Khang (20010744)

Điểm:

Ngày thi:

Phòng thi:

| Giảng viên chấm thi 1  (Ký và ghi rõ họ tên) | Giảng viên chấm thi 2  (Ký và ghi rõ họ tên) |
| --- | --- |
|  |  |
|  |  |

**MỤC LỤC**

[**LỜI CAM ĐOAN 2**](#_heading=h.gjdgxs)

[**DANH MỤC HÌNH VẼ 5**](#_heading=h.3znysh7)

[**LỜI MỞ ĐẦU 6**](#_heading=h.ezvf9ylfqgn)

[**CHƯƠNG 1. GIỚI THIỆU BÀI TOÁN 7**](#_heading=h.tyjcwt)

[1.1 Tổng quan 7](#_heading=h.1t3h5sf)

[1.2 Nhiệm vụ đề tài 7](#_heading=h.661wob3ca9oy)

[1.3 Tổng quan bài toán nhận diện biển số xe 7](#_heading=h.2s8eyo1)

[1.3.1 Khái niệm biển số xe 7](#_heading=h.f2b2bpvwc986)

[1.3.2 Tìm hiểu xử lý ảnh và OpenCV 8](#_heading=h.ppk0hf6tuqpb)

[**CHƯƠNG 2. TIỀN XỬ LÝ ẢNH VÀ TÁCH BIỂN SỐ 9**](#_heading=h.26in1rg)

[2.3 Hướng giải quyết 9](#_heading=h.lnxbz9)

[2.3.1 Chuyển ảnh xám 9](#_heading=h.35nkun2)

[2.3.2 Giảm nhiễu bằng bộ lọc Gauss 10](#_heading=h.h13x3lvyx0eh)

[2.3.3 Nhị phân hóa hình ảnh 12](#_heading=h.rgx86wq59crf)

[2.3.4 Tiến hành vẽ đường biên 13](#_heading=h.1ksv4uv)

[2.3.5 Giữ lại 30 đường biên lớn nhất 14](#_heading=h.2jxsxqh)

[2.3.6 Crop vùng ảnh chứa biển số 15](#_heading=h.novpt72ynxme)

[2.4 Nhận diện biển text trong biển số 16](#_heading=h.z337ya)

[2.4.1 Train mô hình nhận diện text 16](#_heading=h.vw2p5c7aa9q9)

# 

# DANH MỤC HÌNH VẼ

[Hình 1: Hướng giải quyết bài toán 9](#_heading=h.vcow1ckdm1cz)

[Hình 2: Chuyển ảnh xám từ ảnh màu 10](#_heading=h.dyl5cxbysk2d)

[Hình 3: Ảnh nhiễu chưa xử lý 10](#_heading=h.sm2vvf7sl8jb)

[Hình 4: Ma trận lọc Gauss 11](#_heading=h.j34e2m5wyff3)

[Hình 5: Ảnh sau khi giảm nhiễu 11](#_heading=h.ajc75vfdbuq4)

[Hình 6: Nhị phân hóa hình ảnh 12](#_heading=h.ewtxm3mo9hke)

[Hình 7: Vẽ đường biên 14](#_heading=h.x65f55ialosx)

[Hình 8: Giữ lại 30 đường biên lớn nhất 15](#_heading=h.ixfz06xizg8f)

[Hình 9: Ảnh crop vùng chứa biển số 16](#_heading=h.bv3ei2n16uuu)

[Hình 10: Khởi tạo mô hình với 4 tầng Conv2D 17](#_heading=h.olyj64msjrp4)

[Hình 11: Kiểm tra độ chính xác của mô hình 18](#_heading=h.jqf3166vin1f)

# LỜI MỞ ĐẦU

Hiện nay, số lượng xe cộ tham gia giao thông trên đường là rất lớn dẫn đến tiêu tốn rất nhiều nhân lực và vật lực cho việc quản lý phương tiện cá nhân trong bãi gửi xe. Nếu không có một công cụ thuận tiện thì việc quản lý phương tiện cá nhân rất mất thời gian, dễ gây nhầm lẫn, thiệt hay cho người sử dụng dịch vụ tại các bãi đỗ xe. Để giảm tải cho các công việc như thu tiền, bảo hiểm xe, tìm xe cộ trong bãi đỗ xe, trên thế giới đã phát triển công nghệ giám sát tự động đối với các phương tiện giao thông, chính nhờ tính cá nhân của biển số xe mà nó đã trở thành đối tượng chính được sử dụng để nghiên cứu, phát triển trong công nghệ này.Do đó em muốn chọn đề tài này như bước căn bản trong việc tìm hiểu các công cụ giám sát mạnh hơn như kiểm soát xe lưu thông trên đường hay nhận dạng khuôn mặt ... đang được thế giới rất chú trọng lúc này.

## GIỚI THIỆU BÀI TOÁN

## Tổng quan

* Tìm hiểu về biển số xe và hệ thống nhận dạng biển số xe
* Phát biểu bài toán và hướng giải quyết

## Nhiệm vụ đề tài

Từ nội dung nêu trên, đề tài của em sẽ bao gồm các nhiệm vụ sau:

* Tìm hiểu khái quát về xử lý ảnh và bài toán nhận dạng biển số xe.
* Tìm hiểu các công đoạn chính của bài toán nhận dạng biển số xe gồm 3 khâu chính:
* Tiền xử lý ảnh
* Phát hiện vị trí và tách biển số xe
* Nhận dạng ký tự- Cài đặt thử nghiệm.

## Tổng quan bài toán nhận diện biển số xe

### Khái niệm biển số xe

Ở Việt Nam, biển kiểm soát xe cơ giới (hay còn gọi tắt là biển kiểm soát, biển số xe)là tấm biển gắn trên mỗi xe cơ giới, được cơ quan công an cấp (đối với xe quân sự do Bộ Quốc phòng cấp) khi mua xe mới hoặc chuyển nhượng xe. Biển số xe được làm bằng hợp kim nhôm sắt, có dạng hình chữ nhật hoặc hơi vuông, trên đó có in số và chữ (biển xe dân sự không dùng các chữ cái I, J, O, Q, W. Chữ R chỉ dùng cho xe rơ-moóc, sơ-mi rơ-moóc) chobiết: Vùng và địa phương quản lý, các con số cụ thể khi tra trên máy tính còn cho biết danh tính người chủ hay đơn vị đã mua nó, thời gian mua nó phục vụ cho công tác an ninh, đặc biệt trên đó còn có hình Quốc huy Việt Nam dập nổi.

Tiêu chuẩn về kích thước: Ở mỗi nước thường có tiêu chuẩn về kích thước nhất định,còn riêng Việt Nam tỉ lệ kích thước giữa các biển số là gần như giống nhau. Biển số xe có 2 loại, kích thước như sau: Loại biển số dài có chiều cao 110 mm, chiều dài 470 mm; loại biển số ngắn có chiều cao 200mm, chiều dài 280mm nên ta sẽ giới hạn tỉ lệ cao/rộng là 3.5 ≤ cao/rộng ≤ 6.5 (biển một hàng) và 0.8 ≤ cao/rộng ≤ 1.5 (biển hai hàng).

Số lượng ký tự trong biển số xe nằm trong khoảng [7,9]. Chiều cao của chữ và số:80mm, chiều rộng của chữ và số: 40mm.

Từ những đặc điểm trên ta có thể thiết lập những thông số, điều khiển để lọc chọn những đối tượng phù hợp mà ta cần.

### Tìm hiểu xử lý ảnh và OpenCV

Xử lý ảnh là một phân ngành trong xử lý số tín hiệu với tín hiệu xử lý là ảnh. Đây là một phân ngành khoa học mới rất phát triển trong những năm gần đây. Xử lý ảnh gồm 4 lĩnh vực chính: xử lý nâng cao chất lượng ảnh, nhận dạng ảnh, nén ảnh và truy vấn ảnh. Sự phát triển của xử lý ảnh đem lại rất nhiều lợi ích cho cuộc sống của con người. Ngày nay xử lý ảnh đã được áp dụng rất rộng rãi trong đời sống như: photoshop, nén ảnh, nén video, nhận dạng biển số xe, nhận dạng khuôn mặt, nhận dạng chữ viết, xử lý ảnh thiên văn, ảnh y tế,....

OpenCV (Open Computer Vision) là một thư viện mã nguồn mở hàng đầu cho xử lý về thị giác máy tính, machine learning, xử lý ảnh. OpenCV được viết bằng C/C++, vì vậy có tốc độ tính toán rất nhanh, có thể sử dụng với các ứng dụng liên quan đến thời gian thực. Opencv Có các interface cho C/C++, Python Java vì vậy hỗ trợ được cho Window, Linux, Mac Os lẫnAndroid, iOS OpenCV có cộng đồng hơn 47 nghìn người dùng và số lượng download vượt quá 6 triệu lần. Opencv có rất nhiều ứng dụng như:Nhận dạng ảnh, Xử lý hình ảnh, Phục hồi hình ảnh/video, Thực tế ảo, Các ứng dụng khác, …

# 

## TIỀN XỬ LÝ ẢNH VÀ TÁCH BIỂN SỐ

## Hướng giải quyết



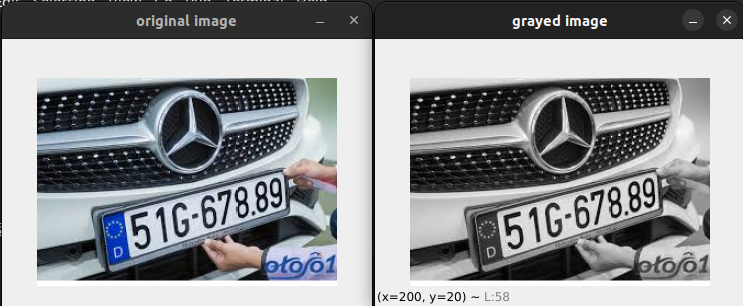
###### Hình 1: Hướng giải quyết bài toán

Ở phạm vi đồ án này, ý tưởng chủ yếu là nhận diện được biển số từ sự thay đổi đột ngột về cường độ ánh sáng giữa biển số và môi trường xung quanh nên ta sẽ loại bỏ các dữ liệu màu sắc RGB bằng cách chuyển sang ảnh xám. Sau đó, ta giảm nhiễu bằng bộ lọc Gauss để loại bỏ những chi tiết nhiễu có thể gây ảnh hưởng đến quá trình nhận diện, đồng thời làm tăng tốc độ xử lý

Việc lấy ngưỡng sẽ giúp ta tách được thông tin biển số và thông tin nền, ở đây em chọn lấy ngưỡng động (Adaptive Threshold). Tiếp đó ta sử dụng thuật toán phát hiện cạnh Canny để trích xuất những chi tiết cạnh của biển số. Trong quá trình xử lý máy tính có thể nhầm lẫn biển số với những chi tiết nhiễu, việc lọc lần cuối bằng các tỉ lệ cao/rộng hay diện tích của biển số sẽ giúp xác định được đúng biển số. Cuối cùng, ta sẽ xác định vị trí của biển số trong ảnh bằng cách vẽ Contour bao quanh

### Chuyển ảnh xám

Ảnh xám (Gray Scale) đơn giản là một hình ảnh trong đó các màu là các sắc thái của màu xám với 256 cấp độ xám biến thiên từ màu đen đến màu trắng, nằm trong giải giá trị từ 0 đến 255, nghĩa là cần 8 bits hay 1 byte để biểu diễn mỗi điểm ảnh này. Lý do cần phải phân biệt giữa ảnh xám và các ảnh khác nằm ở việc ảnh xám cung cấp ít thông tin hơn cho mỗi pixel. Với ảnh thông thường thì mỗi pixel thường được cung cấp 3 trường thông tin trong khi với ảnh xám chỉ có 1 trường thông tin, việc giảm khối lượng thông tin giúp tăng tốc độ xử lý, đơn giản hóa giải thuật nhưng vẫn đảm bảo các tác vụ cần thiết



###### Hình 2: Chuyển ảnh xám từ ảnh màu

### Giảm nhiễu bằng bộ lọc Gauss

### 

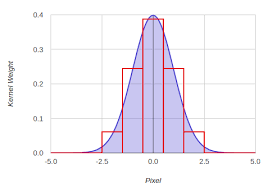
###### Hình 3: Ảnh nhiễu chưa xử lý

Noise được hiểu cơ bản là các dạng chấm hạt nhỏ phân bố trên hình ảnh. Noise có thể làm biến dạng các chi tiết trong ảnh khiến cho chất lượng ảnh thấp. Trên thực tế có nhiều loại nhiễu, nhưng người ta thường chia làm ba loại: nhiễu cộng,nhiễu nhân và nhiễu xung. Bản chất của nhiễu thường tương ứng với tần số cao và cơ sở lý thuyết của bộ lọc là chỉ cho những tín hiệu có tần số nhất định đi qua, nên người ta thường sử dụng bộ lọc thông thấp hay trung bình

#### Bộ lọc Gauss

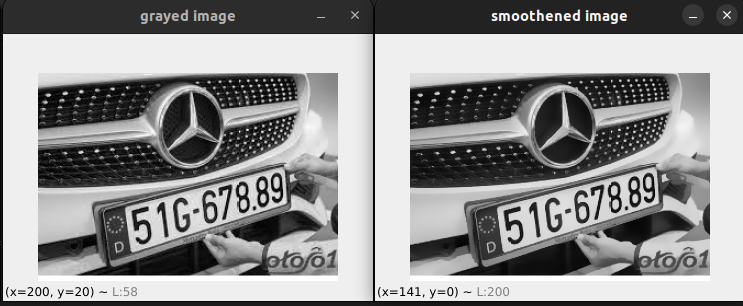
Bộ lọc Gauss được cho là bộ lọc hữu ích nhất, được thực hiện bằng cách nhân chập ảnh đầu vào với một ma trận lọc Gauss sau đó cộng chúng lại để tạo thành ảnh đầu ra.

Ý tưởng chung là giá trị mỗi điểm ảnh sẽ phụ thuộc nhiều vào các điểm ảnh ở gần hơn là các điểm ảnh ở xa. Trọng số của sự phụ thuộc được lấy theo hàm Gauss (cũng được sử dụng trong quy luật phân phối chuẩn)



###### Hình 4: Ma trận lọc Gauss

Giả sử ảnh là một chiều. Điểm ảnh ở trung tâm sẽ có trọng số lớn nhất. Các điểm ảnh ở càng xa trung tâm sẽ có trọng số giảm dần khi khoảng cách từ chúng tới điểm trung tâm tăng lên. Như vậy điểm càng gần trung tâm sẽ càng đóng góp nhiều hơn vào giá trị điểm trung tâm.



###### Hình 5: Ảnh sau khi giảm nhiễu

Và trong dự án này tôi đã sử dụng hàm bilateralFilter là một hàm trong xử lý ảnh, thường được sử dụng để làm mờ (filter) ảnh trong cả không gian màu (color space) và không gian độ sáng (intensity space), giữ lại các cạnh và đồng thời giảm nhiễu.

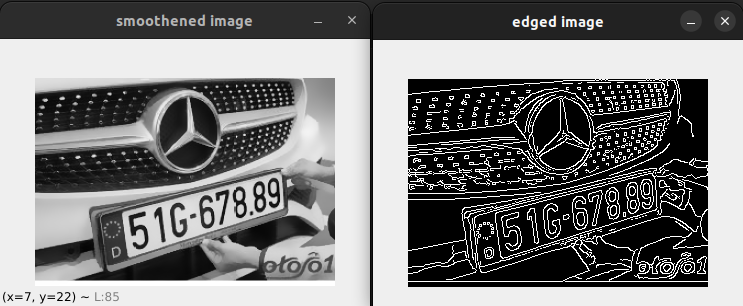
Thuật toán của hàm bilateralFilter kết hợp giữa thông tin về không gian màu và không gian độ sáng của ảnh. Nó sử dụng một hàm cơ bản dựa trên Gaussian để tạo ra một hàm trọng số cho việc lọc. Trọng số này sẽ giảm khi khoảng cách màu hoặc khoảng cách độ sáng giữa các điểm ảnh tăng lên. Điều này giúp giữ lại các đặc trưng quan trọng như cạnh trong ảnh.

### Nhị phân hóa hình ảnh

Edge detector (còn được gọi là bộ phát hiện cạnh) là một thuật toán trong xử lý ảnh được sử dụng để tìm ra các cạnh hoặc biên của các vật thể trong ảnh. Các cạnh thường thể hiện sự thay đổi đột ngột trong độ sáng hoặc màu sắc của điểm ảnh trong không gian ảnh.

Các thuật toán edge detector hoạt động bằng cách phân tích độ dốc hoặc độ biến đổi của đặc trưng của ảnh (như độ sáng hoặc màu sắc) để xác định các điểm ảnh có thể là điểm cạnh. Kết quả của quá trình này thường là một ảnh nhị phân, trong đó các điểm ảnh thuộc về cạnh được gán giá trị 1 và các điểm ảnh khác được gán giá trị 0.

Các kỹ thuật edge detector thường được sử dụng trong nhiều ứng dụng xử lý ảnh, bao gồm nhận diện vật thể, phân tích hình dạng, nhận diện biên bản, và nhiều ứng dụng khác. Một số thuật toán edge detector phổ biến bao gồm Canny edge detector, Sobel operator, Laplacian of Gaussian (LoG), và Prewitt operator



###### Hình 6: Nhị phân hóa hình ảnh

### Tiến hành vẽ đường biên

Việc tìm và vẽ đường viền (contours) trong hình ảnh sau khi đã thực hiện bước nhị phân hóa có nhiều ứng dụng quan trọng trong xử lý hình ảnh và phân tích hình ảnh. Dưới đây là một số lý do:

**Phát hiện và nhận diện các hình dạng**: Việc tìm ra các contours giúp bạn xác định được các hình dạng, đường biên của các vật thể trong hình ảnh. Điều này có thể hữu ích trong việc nhận diện các đối tượng cụ thể hoặc đặc trưng của hình ảnh.

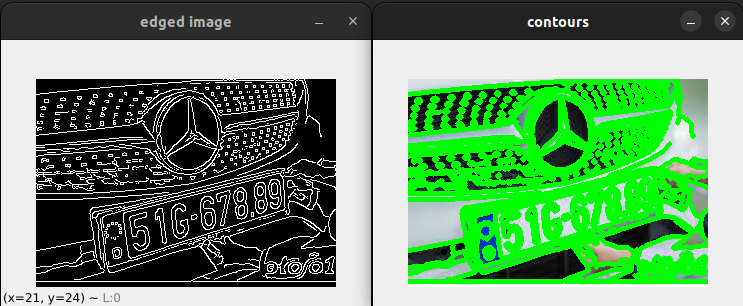
**Trích xuất thông tin về đối tượng**: Khi bạn đã có các contours, bạn có thể tính toán các thông số như diện tích, chu vi, tọa độ các điểm biên của đối tượng. Điều này hữu ích để phân tích và trích xuất thông tin từ hình ảnh.

**Phân tách các vùng khác nhau**: Các contours giúp phân tách các vùng khác nhau trong hình ảnh. Điều này hữu ích khi bạn muốn tách riêng các vật thể hoặc phần của hình ảnh để xử lý riêng biệt.

**Tính toán và vẽ các đặc trưng hình học**: Dựa trên các contours, bạn có thể tính toán các đặc trưng hình học như diện tích, chu vi, trung tâm hình ảnh, hướng chung của đối tượng. Các thông tin này có thể được sử dụng để xác định các thuộc tính quan trọng của đối tượng.

**Phát hiện biên ngưỡng giữa các vùng khác nhau**: Các contours có thể được sử dụng để xác định các biên ngưỡng (boundary) giữa các vùng khác nhau trong hình ảnh. Điều này hữu ích trong việc nhận diện các đối tượng hoặc phần của hình ảnh.

Kết luận: việc tìm và vẽ contours là một bước quan trọng trong xử lý hình ảnh và giúp trích xuất thông tin quan trọng từ hình ảnh để có thể sử dụng cho các mục đích phân tích và nhận diện đối tượng.



###### Hình 7: Vẽ đường biên

### Giữ lại 30 đường biên lớn nhất

Việc giảm số lượng contours sau khi đã xác định tất cả các contours ban đầu có thể có một số lý do:

**Loại bỏ các contours không quan trọng**: Trong một số trường hợp, không tất cả các contours đều mang lại thông tin quan trọng hoặc không cần thiết cho ứng dụng cụ thể. Việc giảm số lượng contours có thể giúp loại bỏ những thông tin không cần thiết này.

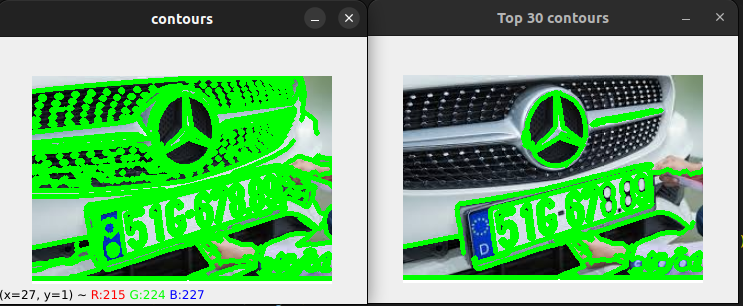
**Tăng hiệu suất tính toán**: Việc xử lý một lượng lớn các contours có thể tốn thời gian và tài nguyên tính toán. Bằng cách giảm số lượng contours, bạn có thể cải thiện hiệu suất của ứng dụng.

**Nhóm các contours liên quan**: Có thể có trường hợp nhiều contours đang đại diện cho cùng một đối tượng hoặc khu vực. Việc giảm số lượng contours có thể giúp gom nhóm và đại diện cho các vùng đối tượng một cách hiệu quả hơn.

**Tách biệt các vật thể**: Khi có nhiều contours gần nhau, việc giảm số lượng contours có thể giúp tách biệt các vật thể một cách rõ ràng hơn.

**Loại bỏ nhiễu**: Trong một số trường hợp, có thể có các contours nhỏ không quan trọng được tạo ra do nhiễu hoặc sự không chính xác trong quá trình xử lý hình ảnh. Việc giảm số lượng contours có thể giúp loại bỏ những contours không cần thiết này.

Kết luận: việc giảm số lượng contours sau khi đã xác định tất cả các contours ban đầu có thể giúp cải thiện hiệu suất tính toán và loại bỏ thông tin không cần thiết hoặc nhiễu từ hình ảnh.



###### Hình 8: Giữ lại 30 đường biên lớn nhất

### Crop vùng ảnh chứa biển số

Sau khi đã nhị phân hóa ảnh và tiến hành xác định và lọc bỏ các contours không cần thiết. Chúng ta tiến hành duyệt vòng lặp for các phần tử contours trong danh sách các contours đã xác định được giữ lại và kiểm tra các điều kiện để tìm được vùng ảnh có khả năng chứa biển số cao nhất.

Đầu tiên chúng ta tiến hành tính chu vi contour giúp định lượng độ dài đường viền của contour đó. Điều này sau đó có thể được sử dụng để xác định xem contour đó có đặc trưng gần giống với một hình chữ nhật hay không. Nếu chu vi của contour xấp xỉ với chu vi của một hình chữ nhật, có thể rằng contour đó tương ứng với một tấm biển.

Tiếp theo sử dụng hàm cv2.approxPolyDP() để tính xấp xỉ đường viền của contour ban đầu bằng một đa giác có số cạnh ít hơn. Trong trường hợp này, chúng ta muốn xấp xỉ contour thành một đa giác có 4 cạnh (hình chữ nhật), vì vậy, độ dài của đoạn thẳng xấp xỉ này cần được xác định. Sau đó kiểm tra nếu đa giác xấp xỉ có 4 cạnh (có thể coi như một hình chữ nhật), thì đó là contour cần tìm.

Ta thực hiện cắt ra vùng hình ảnh con chứa contour tương ứng và lưu hình ảnh vùng con vào file PNG.

### 

###### Hình 9: Ảnh crop vùng chứa biển số

## Nhận diện biển text trong biển số

### Train mô hình nhận diện text

#### Chọn mô hình

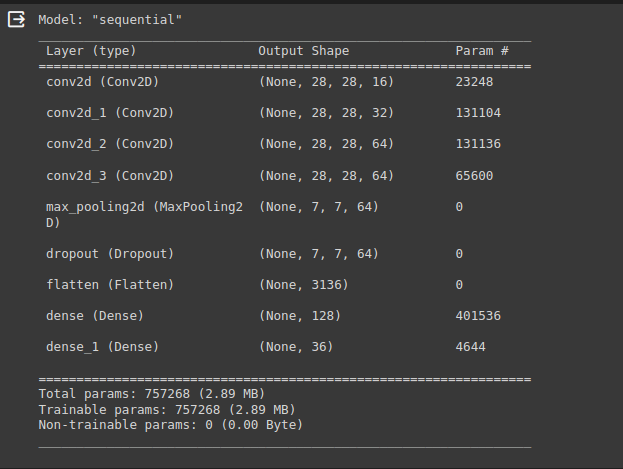
Lựa chọn mô hình huấn luyện rất quan trọng vì nó sẽ ảnh hưởng lớn đến hiệu suất của mô hình cuối cùng. Một số lý do quan trọng để lựa chọn mô hình huấn luyện bao gồm:

* **Phù hợp với bài toán cụ thể:** Mỗi bài toán có yêu cầu và đặc điểm riêng biệt. Có những mô hình được thiết kế đặc biệt cho việc giải quyết các loại vấn đề khác nhau như phân loại ảnh, nhận diện văn bản, dịch ngôn ngữ, v.v. Chọn mô hình phù hợp với bài toán sẽ giúp cải thiện hiệu suất.
* **Kích thước và độ phức tạp của mô hình**: Mô hình quá phức tạp có thể dẫn đến overfitting (mô hình quá tương tự với dữ liệu huấn luyện, nhưng không thể áp dụng cho dữ liệu mới). Ngược lại, mô hình quá đơn giản có thể dẫn đến underfitting (không thể học đúng cách từ dữ liệu huấn luyện). Cần cân nhắc kỹ lưỡng về kích thước và độ phức tạp của mô hình.
* **Dữ liệu huấn luyện có sẵn:** Nếu bạn có lượng dữ liệu huấn luyện lớn, bạn có thể sử dụng các mô hình phức tạp hơn như CNN (Convolutional Neural Network) hoặc Transformer. Tuy nhiên, nếu dữ liệu huấn luyện hạn chế, cần xem xét các mô hình nhỏ hơn để tránh overfitting
* **Tài nguyên tính toán có sẵn:** Một số mô hình yêu cầu tài nguyên tính toán lớn để huấn luyện và triển khai. Cần xem xét xem bạn có đủ tài nguyên (bộ nhớ, GPU, CPU) để huấn luyện và sử dụng mô hình hay không…vv

Và trong dự án này vì đã có sẵn tài nguyên để huấn luyện và cùng từng làm việc bằng mô hình CNN nên tôi quyết định lựa chọn mô hình CCN để tiến hình huấn luyện mô hình nhận diện ký tự.

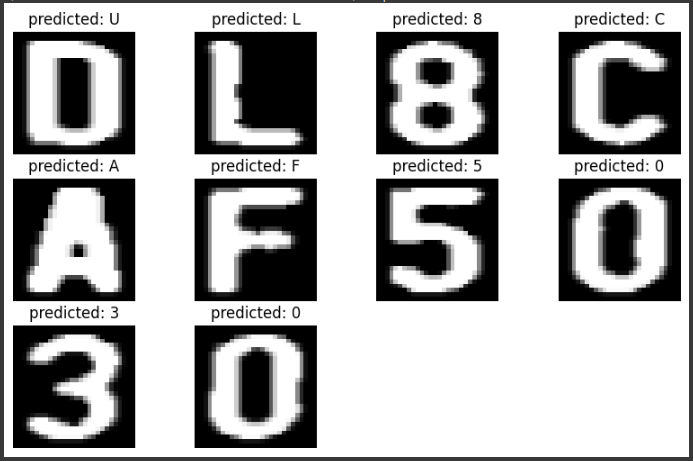
#### Huấn luyện mô hình:

* Đầu tiên chúng ta tiến hành import module tensorflow.keras.backend với mục đích sử dụng các hàm hỗ trợ của backend của Keras và load dữ liệu để đào tạo mô hình.
* Tiến hành xây dựng một mô hình mạng neural để nhận diện ký tự trong hình ảnh biển số xe.
* Tiến hành xây dựng một mạng neural convolutional (CNN) với nhiều tầng Conv2D, MaxPooling2D, Flatten và Dense. Các tầng Conv2D được sử dụng để học các đặc trưng cục bộ trong hình ảnh, trong khi các tầng MaxPooling2D giúp giảm kích thước của đầu ra và giảm số lượng tham số cần học.
* Tầng Conv2D đầu tiên sử dụng kernel size là (22,22) để học các đặc trưng lớn, sau đó các tầng tiếp theo sử dụng kernel size nhỏ hơn để học các đặc trưng nhỏ hơn.
* Dropout được áp dụng sau tầng MaxPooling2D để ngăn overfitting.
* Cuối cùng, thông qua các tầng Dense, mô hình học các mối quan hệ giữa các đặc trưng đã học để phân loại ký tự thành các lớp khác nhau.



###### Hình 10: Khởi tạo mô hình với 4 tầng Conv2D

* Thực hiện huấn luyện mô hình sử dụng dữ liệu từ train\_generator và đánh giá hiệu suất trên validation\_generator
* train\_generator: Đây là một ImageDataGenerator được sử dụng để tạo ra dữ liệu huấn luyện.
* steps\_per\_epoch: Số bước huấn luyện trong mỗi epoch. Trong trường hợp này, nó được tính bằng train\_generator.samples // batch\_size.
* validation\_data: Dữ liệu được sử dụng để đánh giá hiệu suất của mô hình sau mỗi epoch. Trong trường hợp này, đó là validation\_generator.
* epochs: Số lượng epochs bạn muốn huấn luyện mô hình.
* verbose: Có sử dụng log tiến trình không?



###### Hình 11: Kiểm tra độ chính xác của mô hình

KẾT LUẬN

Sau một thời gian tìm hiểu và nghiên cứu tôi đã thành công trong việc tiền xử lý ảnh và phát triển mô hình nhận diện biển số xe với thư viện pyocr. Ứng dụng đã có khả năng khoanh vùng và nhận diện được biển số. Với sự phát triển không ngừng của công nghệ và làn sóng công nghệ AI đang rất hot, chúng tôi đã nhận thấy được tiềm năng của ứng dụng nhận diện biển số bằng xử lý ảnh.

Tuy nhiên với sự hiểu biết có hạn về công nghệ khiến cho ứng dụng tôi tạo ra còn có khả năng nhận diện chưa tốt. Trong tương lai tôi cam kết tiếp tục nghiên cứu và phát triển để nâng cao tính chính xác của ứng dụng. Tôi hy vọng sản phẩm có thể đóng góp phần nào giá trị cho xã hội.

TÀI LIỆU THAM KHẢO

| [1] | Dữ liệu để huấn luyện mô hình nhận diện ký tự:  <https://www.kaggle.com/datasets/sarthakvajpayee/ai-indian-license-plate-recognition-data> |
| --- | --- |
| [2] | Đọc, hiển thị và ghi ảnh bằng OpenCV.  <https://aiots.vn/phan-2-doc-hien-thi-va-ghi-anh-bang-opencv/> |
| [3] | Tìm hiểu về mạng tích chập (CNN)  <https://viblo.asia/p/deep-learning-tim-hieu-ve-mang-tich-chap-cnn-maGK73bOKj2> |
| [4] | Tìm hiểu về thư viện PyOCR  <https://pypi.org/project/pyocr/> |