**BỘ GIÁO DỤC VÀ ĐÀO TẠO**

**TRƯỜNG ĐẠI HỌC GIAO THÔNG VẬN TẢI PHÂN HIỆU TẠI TP. HỒ CHÍ MINH**

**BỘ MÔN CÔNG NGHỆ THÔNG TIN**

**BÁO CÁO TỐT NGHIỆP**

**ĐỀ TÀI: ÁP DỤNG MẠNG WPOD NHẬN DIỆN BIỂN SỐ XE ỨNG DỤNG TRONG VIỆC QUẢN LÝ XE**

|  |  |
| --- | --- |
| Giảng viên hướng dẫn | : ThS. TRẦN PHONG NHÃ |
| Sinh viên thực hiện | : ĐÀO ĐỨC ĐẠT |
| Lớp | : CÔNG NGHỆ THÔNG TIN |
| Khóa | : 58 |

**Thành Phố Hồ Chí Minh, tháng 7 năm 2021**

TRƯỜNG ĐẠI HỌC GIAO THÔNG VẬN TẢI **CỘNG HÒA XÃ HỘI CHỦ NGHIÃ VIỆT NAM**

**PHÂN HIỆU TẠI THÀNH PHỐ HỒ CHÍ MINH** Độc lập – Tự do – Hạnh phúc

# NHIỆM VỤ THIẾT KẾ TỐT NGHIỆP

BỘ MÔN: **CÔNG NGHỆ THÔNG TIN**

-------\*\*\*-------

**Mã sinh viên:** 5851071020 **Họ tên SV:** Đào Đức Đạt

**Khóa:** 58 **Lớp:** Công nghệ thông tin

1. **Tên đề tài**

***NGHIÊN CỨU NHẬN DIỆN BIỂN SỐ XE ỨNG DỤNG TRONG VIỆC QUẢN LÝ XE TRONG BÃI ĐỖ XE.***

1. **Mục đích, yêu cầu**
2. **Mục đích**

Xây dựng tính năng nhận diện được biển số xe qua ảnh.

Xây dựng tính năng xây dựng hệ thống ghi nhận xe ra vào.

Xây dựng tính năng tính tiền cho xe ra vào.

1. **Yêu cầu**

Tìm hiểu về xử lý ảnh.

Tìm hiểu về phát hiện biển số xe.

Tìm hiểu về nhận diện ký tự.

1. **Nội dung và phạm vi đề tài**
2. **Nội dung đề tài**

Giới thiệu các bước về nhận diện biển số

Giới thiệu thuật toán dùng để phát hiện biển số xe.

Giới thiệu thuật toán dùng để nhận diện ký tự.

Xây dựng phần mềm thử nghiệm.

1. **Phạm vi nghiên cứu**

Nghiên cứu thuật toán CNN dùng để phát hiện biển số xe và thuật toán SVM để nhận diện ký tự

Ứng dụng các thuật toán và các công cụ nhỏ gọn của ngôn ngữ cung cấp để xây dựng bản thử nghiệm đơn giản.

1. **Công nghệ, công cụ, và ngôn ngữ lập trình**

Công nghệ: Python, SQLite3.

Công cụ: Một số thư viện mã nguồn mở của Python: numpy, opencv, keras, tkinter.

Ngôn ngữ sử dụng: Python, SQL

1. **Các kết quả cơ bản đạt được**

Hoàn chỉnh cuốn báo cáo đề tài.

Hiểu về các thức hoạt động của của thuật toán phát hiện biển số xe và thuật toán nhận dạng ký thự.

Xây dựng phần mềm ứng dụng trong việc nhận diện

1. **Giáo viên hướng dẫn**

Họ tên: Trần Phong Nhã

Đơn vị công tác: Bộ môn Công Nghệ Thông Tin – Trường Đại học Giao Thông Vận Tải phân hiệu TPHCM

Điện thoại: 0906761014 Email: [tpnha@utc2.edu.vn](mailto:tpnha@utc2.edu.vn)

|  |  |
| --- | --- |
| Ngày tháng 7 năm 2021  Trưởng BM Công Nghệ Thông Tin | Đã giao nhiệm vụ TKTN  Giáo viên hướng dẫn |
| ThS.Trần Phong Nhã |  |

Đã nhận nhiệm vụ TKTN

Sinh viên: Đào Đức Đạt Ký tên:

Điện thoại: 0965025225 Email: 5851071020@st.utc2.edu.vn

LỜI CẢM ƠN

Lời nói đầu tiên, em xin kính gửi lời cảm ơn chân thành nhất tới Quý thầy cô trong Bộ môn Công Nghệ Thông Tin, cũng như Ban Giám Hiệu Trường Đại học Giao thông Vận tải phân hiệu tại Thành phố Hồ Chí Minh, đã cho phép em thực hiện đề tài tốt nghiệp: ***NGHIÊN CỨU NHẬN DIỆN BIẾN SỐ ỨNG DỤNG TRONG VIỆC QUẢN LÝ XE.***

Để hoàn thành nhiệm vụ được giao này, ngoài sự nỗ lực học hỏi không ngừng của bản thân còn có sự hướng dẫn tận tình của thầy giáo **Trần Phong Nhã**, người đã hướng dẫn cho em những hướng đi, truyền đạt cho em những kiến thức, kỹ năng để em có thể hoàn thành đề tài tốt nghiệp này.

Mặc dù đã cố gắng hết sức để hoàn thành đề tài, nhưng chắc chắn rằng sẽ khó tránh khỏi những thiếu sót. Em rất mong nhận được những sự đánh giá, góp ý của Quý thầy cô để em có thể rút ra cho mình những bài học, kinh nghiệm quý báu.

Sau cùng, em cũng không biết nói gì hơn ngoài kính chúc Quý thầy cô trong Bộ môn Công Nghệ Thông Tin và đặc biệt là thầy giáo **Trần Phong Nhã** thật dồi dào sức khỏe và ngày càng gặt hái được nhiều thành công hơn nữa trong cuộc sống cũng như trong sự nghiệp giảng dạy của mình.

Em xin chân thành cảm ơn !

|  |
| --- |
| *Tp. Hồ Chí Minh, ngày tháng năm 2021*  *Sinh viên thực hiện*  Đào Đức Đạt |

NHẬN XÉT CỦA GIÁO VIÊN

|  |
| --- |
| ***Tp. Hồ Chí Minh, ngày tháng năm 2021***  ***Sinh viên thực hiện***  **Đào Đức Đạt** |

MỤC LỤC

[NHIỆM VỤ THIẾT KẾ TỐT NGHIỆP 2](#_Toc77199202)

[GIỚI THIỆU ĐỀ TÀI 1](#_Toc77199203)

[Đặt vấn đề 1](#_Toc77199204)

[Mục đích đề tài 1](#_Toc77199205)

[Đối tượng nghiên cứu 1](#_Toc77199206)

[Phạm vi nghiên cứu 1](#_Toc77199207)

[CHƯƠNG 1: CƠ SỞ LÝ THUYẾT 3](#_Toc77199208)

[1.1 Tìm hiểu thuật toán phát hiện biển số 3](#_Toc77199209)

[1.1.1 Giới thiệu CNN 3](#_Toc77199210)

[1.1.2 Mạng noron tích chập (mạng CNN) 4](#_Toc77199211)

[1.1.3 Giới thiệu mạng WPOD 6](#_Toc77199212)

[1.1.4 Kiến trúc Mạng WPOD-NET 6](#_Toc77199213)

[1.1.5 Hàm loss của WPOD-NET 7](#_Toc77199214)

[1.2 Thuật toán SVM 8](#_Toc77199215)

[1.2.1 Giới thiệu thuật toán SVM 8](#_Toc77199216)

[1.2.2 SVM Tuyến tính 9](#_Toc77199217)

[1.2.3 Soft margin và 10](#_Toc77199218)

[1.3 Các công cụ xây dựng đồ án 11](#_Toc77199219)

[**1.3.1** **Ngôn ngữ lập trình** 11](#_Toc77199220)

[**1.3.2** **Thư viện Tensorflow** 11](#_Toc77199221)

[**1.3.3** **Database** 12](#_Toc77199222)

[**1.3.4** **Xây dựng GUI** 12](#_Toc77199223)

[CHƯƠNG 2: BÀI TOÁN NHẬN DIỆN BIỂN SỐ XE 14](#_Toc77199224)

[2.1 Tìm hiểu về biển số xe 14](#_Toc77199225)

[2.2 Khái quát chung về nhận dạng biển số xe 16](#_Toc77199226)

[2.3 Phát hiện biển số 16](#_Toc77199227)

[2.3.1 Sử dụng phương pháp biến đổi hình thái học 16](#_Toc77199228)

[2.3.2 Sử dụng mạng WPOD 20](#_Toc77199229)

[2.4 Tách ký tự 21](#_Toc77199230)

[2.5 Nhận diện ký tự 23](#_Toc77199231)

[3.5.1 Hitogram insector kernel 23](#_Toc77199232)

[3.5.2 Huấn luyện thuật toán SVM 24](#_Toc77199233)

[CHƯƠNG 3: ỨNG DỤNG 25](#_Toc77199234)

[3.1 Yêu cầu bài toán 25](#_Toc77199235)

[3.2 Thiết kế cơ sở dữ liệu 25](#_Toc77199236)

[3.3 Sơ đồ UseCase 25](#_Toc77199237)

[3.4 Sơ đồ hoạt động 26](#_Toc77199238)

[3.5 Giao diện phần mềm 27](#_Toc77199239)

[Tổng kết 29](#_Toc77199240)

[Kết quả đạt được 29](#_Toc77199241)

[Hạn chế 29](#_Toc77199242)

[Hướng phát triển 29](#_Toc77199243)

[PHỤ LỤC 30](#_Toc77199244)

[Hướng dẫn cài đặt 30](#_Toc77199245)

[Hướng dẫn sử dụng 30](#_Toc77199246)

[Tài liệu tham khảo 31](#_Toc77199247)

**DANH MỤC THUẬT NGỮ**

|  |  |  |  |
| --- | --- | --- | --- |
| STT | THUẬT NGỮ | Ý NGHĨA TIẾNG VIỆT | TỪ VIẾT TẮT |
| 1 | Convolutional Neural Network | Mạng nơ-ron tích chập | CNN |
| 2 | Support Vector Machine |  | SVM |
| 3 | License-plate | Biển số xe | LP |
| 4 | Warped Planar Object Netwwork | Mạng phát hiện đối tượng mặt phẳng cong vênh | WPOD |
| 5 | User Interface | Giao diện người dùng | UI |
| 6 | Graphical User Interface | Giao diện đồ họa người dùng | GUI |
| 7 | Optical Character Recognition | Nhận dạng ký tự | OCR |
| 8 | Artificical Neural Network | Mạng nơ-ron nhân tạo | ANN |
| 9 | Single Shot Detector | Tên 1 mạng phát hiện đối tượng | SSD |
| 10 | You only look once | Tên 1 mạng phát hiện đối tượng | YOLO |

**Danh sách hình ảnh**

[Hình 1.1 Hoạt động tích chập của CNN 3](#_Toc77199051)

[Hình 1.2 Kiến trúc mạng WPOD 6](#_Toc77199052)

[Hình 1.3 Thuật toán SVM 9](#_Toc77199053)

[Hình 1.4 Cách thức hoạt động của thuật toán SVM 10](#_Toc77199054)

[Hình 2.1 Biển số nền trắng, chữ đen 14](#_Toc77199055)

[Hình 2.2 Biển số nền xanh dương, chữ trắng 14](#_Toc77199056)

[Hình 2.3 Biển số nền đỏ, chữ trắng 15](#_Toc77199057)

[Hình 2.4 Biển số nền vàng, chữ đen 15](#_Toc77199058)

[Hình 2.5 Sơ đồ nhận diện biển số 16](#_Toc77199059)

[Hình 2.6 Hình biển số xe sau khi làm mịn và cân bằng sáng 17](#_Toc77199060)

[Hình 2.7 Hình biển số xe sau khi nhị phân 17](#_Toc77199061)

[Hình 2.8 Đường bao của hình biển số xe 18](#_Toc77199062)

[Hình 2.9Biển số xe bị mất khi tìm đường bao 19](#_Toc77199063)

[Hình 2.10 Sơ đồ khối phát hiện biển số xe 20](#_Toc77199064)

[Hình 2.11 Ảnh biển số được nhị phân để tách ký tự 21](#_Toc77199065)

[Hình 2.12 Sơ đồ khối tách ký tự 22](#_Toc77199066)

[Hình 2.13 Kết quả sau khi thực hiện tách ký tự 23](#_Toc77199067)

[Hình 2.14 Các kí tự để huyến luyện thuật toán SVM 24](#_Toc77199068)

[Hình 3.1 Sơ đồ use case hệ thống 26](#_Toc77199069)

[Hình 3.2 Sơ đồ hoạt động khi xe vào 26](#_Toc77199070)

[Hình 3.3 Sơ đồ hoạt động khi xe ra 27](#_Toc77199071)

[Hình 3.4 Giao diện nhận diện biển số xe vào- ra 27](#_Toc77199072)

[Hình 3.5 Màn hình đăng nhập khi muốn thay đổi giá 28](#_Toc77199073)

[Hình 3.6 Màn hình cho phép điều chỉnh giá và lịch sử ra vào 28](#_Toc77199074)

**LỜI GIỚI THIỆU**

Xử lý và nhận dạng là 1 lĩnh vực được nhiều người quan tâm. Nó đã được ứng dụng và nhiều lĩnh vực như:

Trong y học, cải thiện thiện ảnh Xquang và nhận dạng đường biên mạch máu từ ảnh chụp tia X, ứng dụng vào các xét nghiệm lâm sàng như phát hiện u não, nội soi cắt lớp…

Trong thiên văn, hệ thống chụp hình gắn ở các tàu vũ trụ hạn chế về mặt kích thước và trọng lượng, do đó chất lượng ảnh nhận được bị giảm đi. Các hình ảnh sẽ được xử lý bằng máy vi tính.

Trong lĩnh vực công nghiệp, người máy được sử dụng trong những công việc nguy hiểm, đòi hỏi cả về tốc độ và độ chính xác cao. Người máy sẽ trở nên tinh vi hơn và thị giác máy tính đóng vai trò quan trọng hơn. Người ta sẽ không chỉ đòi hỏi người máy phát hiện và nhận dạng các bộ phận công nghiệp mà còn phải hiểu những gì chúng thấy. Xử lý ảnh sẽ tác động đến thị giác máy tính.

Ngoài ra, xử lý ảnh và nhận dạng còn được ứng dụng trong lĩnh vực khác ít được nói đến hơn. Công an giao thông thường chụp ảnh trong môi trường không thuận lợi, ảnh thường bị chéo nên cần xử lý và nhận dạng

# GIỚI THIỆU ĐỀ TÀI

## Đặt vấn đề

Nhu cầu sở hữu riêng phương tiện cá nhân tăng cao, từ đó dẫn đến nhu cầu gửi xe cũng đã tăng cao. Do đó nhu cầu cần hệ thống tự động. Một trong những hệ thống đó là hệ thống nhận dạng biển số xe. Đó là hệ thống có thể nhận biết được biển số xe và có thể đọc được những ký tự.

Trong bài luận văn, người thực hiện sẽ xây dựng một hệ thống có thể có thể đọc được biển số xe đưa ra thành ký tự. Do có quá nhiều bất cập trong quá trình gửi xe:

Tốn nhiều nhân công.

Thẻ xe vẫn có thể mất.

Giá cả có thể chưa được chính xác.

## Mục đích đề tài

Hệ thống nhận dạng biển số xe là một ứng dụng dựa trên kỹ thuật xử lý ảnh số. Mục đích nhận dạng biển số xe là thực hiện các bước xử lý để từ một ảnh đầu vào, máy tính có thể nhận ra chính xác biến số xe trên ảnh. Nhận dạng biển số xe trở thành một ứng dụng hữu ích, được đưa vào trong lĩnh vực như: quản lý gia thông, kiểm tra an ninh, thu phí giao thông, trạm gác cổng, quản lý các bãi giữ xe một cách tự động

## Đối tượng nghiên cứu

Đây là một trong những ứng dụng xử lý ảnh nhận được nhiều sự quan tâm nghiên cứu về cả hai mặt lý thuyết và thực hiện chương trình. Đề tài của em bao gồm các quá trình xử lý: phân vùng biển số, tách các ký tự, nhận biết các ký tự, phần mềm quản lý cơ sở dữ liệu biển số xe. Mục đích của phần mềm quản lý cơ sở dữ liệu biển số xe là thu nhận chuỗi biển số xe sau khi đã nhận dạng được, lưu vào cơ sở dữ liệu để người dùng dễ dàng quản lý và theo dõi thông qua các báo cáo chi tiết các lượt xe vào-ra. Mục đích của phần mềm quản lý cơ sở dữ liệu là theo dõi lượt ra vào, tính tiền lưu bãi.

## Phạm vi nghiên cứu

Chủ yếu dựa vào các tài liệu và chạy mô phỏng nhận dạng biển số xe trên trên máy tính bằng phần mềm xây dựng bằng Python, chưa có mô hình thực tế.

Do thời gian thực hiện đề tài không cho phép nên người thực hiện giới hạn các biển số xe và điều kiện:

* Biển số xe của Việt Nam, nền trắng, chữ đen.
* Biển số xe nguyên vẹn, không bị tróc sơn hay rỉ sét, không bị che khuất.
* Hình chụp biển số không bị mờ, ký tự biển số còn phân biệt, nhận dạng bằng trực quan
* Không bị nhiễu bởi ánh sáng làm ảnh chụp bị chói.

# CƠ SỞ LÝ THUYẾT

## Tìm hiểu thuật toán phát hiện biển số

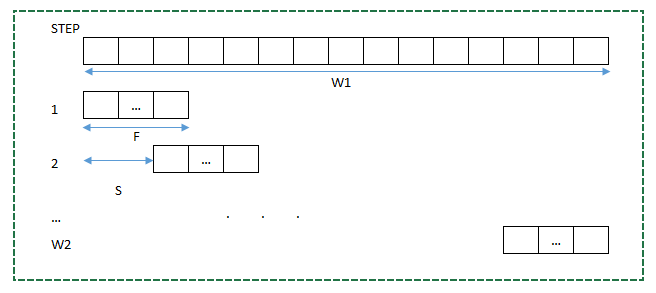
### Giới thiệu CNN

Tích chập là một khái niệm trong xử lý tín hiệu số nhằm biến đổi thông tin đầu vào thông qua một phép tích chập với bộ lọc để trả về đầu ra là một tín hiệu mới. Tín hiệu này sẽ làm giảm những đặc trưng mà bộ lọc không quan tâm và chỉ giữ những đặc trưng chính.

Tích chập thông dụng nhất là tích chập 2 chiều được áp dụng trên ma trận đầu vào và ma trận bộ lọc 2 chiều. Phép tích chập của một ma trận  với một bộ lọc (receptive field)  là một ma trận  sẽ trả qua những bước sau:

Tính tích chập tại 1 điểm: Tại vị trí đầu tiên trên cùng của ma trận đầu vào ta sẽ lọc ra một ma trận con  có kích thước bằng với kích thước của bộ lọc. Giá trị  tương ứng trên Y là tích chập của  với F được tính như sau:

Tiến hành trượt dọc ma trận theo chiều từ trái qua phải, từ trên xuống dưới với bước nhảy (stride) S ta sẽ tính được các giá trị  tiếp theo. Sau khi quá trình này kết thúc ta thu được trọn vẹn ma trận đầu ra Y.



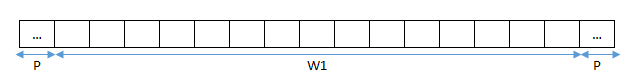
Hình 1.1 Hoạt động tích chập của CNN

Quá trình trượt theo chiều rộng . Mỗi dòng tương ứng với một bước. Mỗi bước dịch sang phải một khoảng  đơn vị cho tới khi đi hết  ô. Nếu bước cuối cùng bị dư thì sẽ lát (padding) thêm để mở rộng ma trận sao cho quá trình tích chập không bị dư ô.

Giả sử quá trình này sẽ dừng sau  bước. Tại bước đầu tiên ta đi được đến vị trí thứ . Sau mỗi bước liền sau sẽ tăng so với vị trí liền trước là . Như vậy đến bước thứ i quá trình trượt sẽ đi đến vị trí . Suy ra tại bước cuối cùng  ma trận sẽ đi đến vị trí . Đây là vị trí lớn nhất gần với vị trí cuối cùng là . Trong trường hợp lý tưởng thì . Từ đó ta suy ra:

Khi vị trí cuối cùng không trùng với thì số bước sẽ lấy được phần nguyên:

Luôn có thể tạo ra đẳng thức (1) nhờ thêm phần đường viền (padding) tại các cạnh của ảnh với độ rộng viền là P sao cho phép chia cho S là chia hết. Khi đó:



Hoàn toàn tương tự ta cũng có công thức ứng với chiều cao:

### Mạng noron tích chập (mạng CNN)

Tích chập được ứng dụng phổ biến trong lĩnh vực thị giác máy tính. Thông qua các phép tích chập, các đặc trưng chính từ ảnh được trích xuất và truyền vào các tầng tích chập (layer convolution). Mỗi một tầng tích chập sẽ bao gồm nhiều đơn vị mà kết quả ở mỗi đơn vị là một phép biến đổi tích chập từ layer trước đó thông qua phép nhân tích chập với bộ lọc.

Về cơ bản thiết kế của một mạng nơ ron tích chập 2 chiều có dạng như sau:

INPUT -> [[CONV -> RELU]\*N -> POOL?]\*M -> [FC -> RELU]\*K -> FC

* Input: Đầu vào
* CONV: Tầng tích chập
* RELU: Tầng kích hoạt. Thông qua hàm kích hoạt (activation function), thường là ReLU hoặc LeakyReLU để kích hoạt phi tuyến
* POOL: Tầng tổng hợp, thông thường là Max pooling hoặc có thể là Average pooling dùng để giảm chiều của ma trận đầu vào.
* FC: Tầng kết nối hoàn toàn. Thông thường tầng này nằm ở sau cùng và kết nối với các đơn vị đại diện cho nhóm phân loại.

Như vậy ta có thể thấy một mạng nơ ron tích chập về cơ bản có 3 quá trình khác nhau:

Quá trình tích chập (convolution): Thông qua các tích chập giữa ma trận đầu vào với bộ lọc để tạo thành các đơn vị trong một tầng mới. Quá trình này có thể diễn ra liên tục ở phần đầu của mạng và thường sử dụng kèm với hàm kích hoạt ReLU. Mục tiêu của tầng này là trích suất đặc trưng hai chiều.

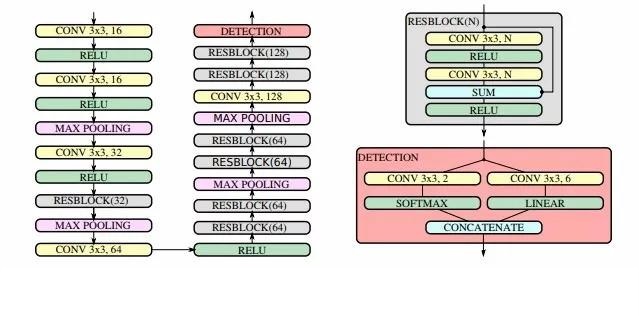
Quá trình tổng hợp (max pooling): Các tầng càng về sau khi trích xuất đặc trưng sẽ cần số lượng tham số lớn do chiều sâu được qui định bởi số lượng các kênh ở các tầng sau thường tăng tiến theo cấp số nhân. Điều đó làm tăng số lượng tham số và khối lượng tính toán trong mạng nơ ron. Do đó để giảm tải tính toán sẽ cần giảm kích thước các chiều của khối ma trận đầu vào hoặc giảm số đơn vị của tầng. Vì mỗi một đơn vị sẽ là kết quả đại diện của việc áp dụng 1 bộ lọc để tìm ra một đặc trưng cụ thể nên việc giảm số đơn vị sẽ không khả thi. Giảm kích thước khối ma trận đầu vào thông qua việc tìm ra 1 giá trị đại diện cho mỗi một vùng không gian mà bộ lọc đi qua sẽ không làm thay đổi các đường nét chính của bức ảnh nhưng lại giảm được kích thước của ảnh. Do đó quá trình giảm chiều ma trận được áp dụng. Quá trình này gọi là tổng hợp nhằm mục đích giảm kích thước dài, rộng.

Quá trình kết nối hoàn toàn (fully connected): Sau khi đã giảm kích thước đến một mức độ hợp lý, ma trận cần được trải phẳng (flatten) thành một vector và sử dụng các kết nối hoàn toàn giữa các tầng. Quá trình này sẽ diễn ra cuối mạng CNN và sử dụng hàm kích hoạt là ReLU. Tầng kết nối hoàn toàn cuối cùng (fully connected layer) sẽ có số lượng đơn vị bằng với số classes và áp dụng hàm kích hoạt là softmax nhằm mục đích tính phân phối xác xuất.

Biển số xe về bản chất là vật thể hình chữ nhật và phẳng được gắn vào xe nhằm mục đích nhận dạng. Để tận dụng lợi thế của hình dạng của nó, em đề xuất một mạng CNN có tên là có tên là Warped Planar Object Detection Network (WPOD-NET). Mạng này học cách phát hiện biển số xe ở nhiều dạng khác nhau và hồi quy các hệ số của một phép biến đổi affine làm phẳng biển số xe bị méo trở thành một hình chữ nhật giống như nhìn trực tiếp.

WPOD-NET được phát triển bằng cách sử dụng chi tiết từ YOLO và SSD và Spatial Tranformer Networks (STN). YOLO và SSD phát hiện nhanh và nhiều đối tượng cùng một lúc nhưng chúng không thực hiện phép biến đổi không gian, chỉ tạo ra được một hình chữ nhật bao quanh. Ngược lại STN có thể có thể sử dụng để phát hiện vùng không phải hình chữ nhật tuy nhiên nó không thể xử lý nhiều chuyển đổi cùng một lúc chỉ thực hiện 1 chuyển đổi.

### Kiến trúc Mạng WPOD-NET



Hình 1.2 Kiến trúc mạng WPOD

Kiến trúc được đề xuất có tổng cộng 21 lớp chập, trong đó 14 lớp nằm bên trong các khối còn lại. Kích thước của tất cả phức tạp bộ lọc được cố định trong 3 × 3. Kích hoạt ReLU được sử dụng trong toàn bộ mạng, ngoại trừ trong khối phát hiện. Có 4 lớp gộp tối đa kích thước 2 × 2 và bước 2 làm giảm kích thước đầu vào đi một hệ số là 16. Cuối cùng, khối phát hiện có hai lớp chập song song: (i) một lớp để suy ra xác suất, được kích hoạt bởi một hàm softmax và một xác suất khác để hồi quy các tham số affine, mà không cần kích hoạt (hoặc, tương đương, sử dụng danh tính F (x) = x là hàm kích hoạt).

### Hàm loss của WPOD-NET

Đặt , i=1,2,3,4 biểu thị bốn góc của biển số xe theo chiều kim đồng hồ bắt đầu từ góc trên bên trái. Đặt , , , biểu thị tương ứng các đỉnh của 1 hình vuông đơn vị chính tắc có tâm tại điểm gốc.

Bức ảnh đầu vào có chiều cao là H và chiều rộng là W và bước nhảy là đầu ra của mạng là trong đó và . Đối với mỗi điểm trong ảnh có có 8 giá trị được ước tính: 2 giá trị đầu tiên (và ) là xác xuất có phải là đối tượng, sáu giá trị cuối cùng (từ đến ) được xây dựng để phép biến đổi afine được định nghĩa bởi:

Hàm mắc được sử dụng cho và để đảm bảo rằng đường chéo là dương (tránh phản chiều không mong muốn hoặc xoay quá nhiều). Để phù hợp với độ phân giải đầu ra, các điểm được chia lại tỷ lệ lại bằng nghịch đảo của bước sóng và căn giữa lại theo từng điểm (m,n) trong bản đồ đặc trưng. Điều này được thực hiện bằng một chức năng chuẩn hóa

Với là hằng số tỷ lệ đại diện cho cạnh của hình vuông giả tưởng. Đặt , điều này có nghĩa là điển trung bình tối đa và tối thiểu trong dữ liệu đào tạo tăng cường chia cho bước mạng.

Giả sử rằng có một đối tượng (biển số) tại ô (m,n) phần đầu tiên của sự mất mát hàm xem lỗi giữa một phiên bản cong vênh của hình vuông chuẩn và các điểm được chú thích chuẩn hóa của biển số, được cung cấp bởi

Phần thứ hai của hàm mất mát đó là xử lý xác xuất có/không của một đối tượng tại ô (m,n)

Với là chức năng chỉ điểm đối tượng trả về 1 nếu có điểm (m,n) đó là biển số hoặc 0 nếu điểm (m,n) đó không là biển số), và . Một đối tượng đượ cho là biển số trong ô (m, n) nếu hình hộp chữ nhật có IOU lớn hơn ngưỡng

Hàm mất mát cuối cùng là:

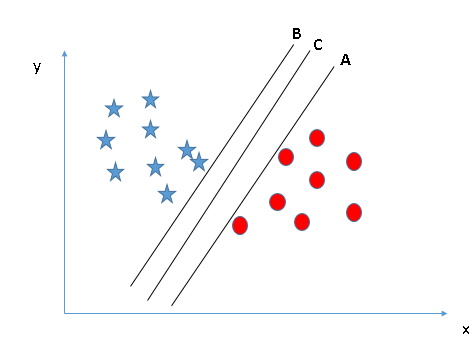
Các bước để cắt biển số xe

Ảnh sẽ được xử lý tiền xử lý sau đó sẽ đi qua mạng WPOD và sau đó ảnh biển số xe sẽ được cắt ra

## Thuật toán SVM

### Giới thiệu thuật toán SVM

SVM là một thuật toán học giám sát, nó có thể sử dụng cho cả việc phân loại hoặc đệ quy. Thường thì nó được sử dụng trong việc phân loại. Về cơ bản, SVM tìm thấy một mặt phẳng tạo ranh giới giữa các loại



Hình 1.3 Thuật toán SVM

Ta có tập mẫu:

Với mang giá trị 1 hoặc -1, xác định lớp của điểm . Mỗi là 1 véc-tơ p chiều. Ta cần tìm siêu phẳng để tách các điểm y = 1 và các điểm có y = -1.

Mục tiêu là tìm một mặt phẳng:

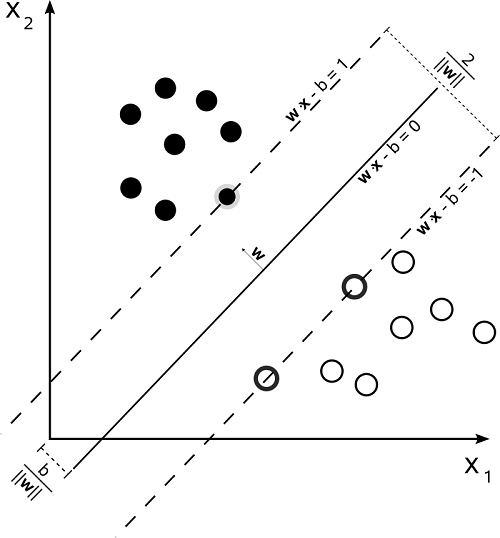
H:

Và 2 mặt siêu phắng song song:

Trong đó w là vector trọng số, b là độ dịch.

Sao cho không có bất cứ điểm nào nằm giữa H1 và H2 đồng thời khoảng cách (còn gọi là lề) giữa H1 và H2 là lớn nhất.

Sao cho không có bất cứ điểm nào nằm giữa H1 và H2 đồng thời khoảng cách (còn gọi là lề ) giữa H1 và H2 là lớn nhất.



Hình 1.4 Cách thức hoạt động của thuật toán SVM

Khoảng cách giữa H và H1 là suy ra khoảng cách giữa H1 và H2 là

Vì vậy để lề cực đại cần tìm min ||w|| = . Với điều kiện không có điểm nằm giữa H1 và H2 ta có:

với y = 1

với y = -1

Như vậy ta có bài toán:

### Soft margin và non-linear

Trong thực tế dữ liệu sẽ bị nhiễu không thể tách tuyến tính nên không thể tìm được H1 và H2. Thêm tỉ lệ sai sót vào, ta có công thức:

Với điều kiện

Cách thứ 2, sử dụng một ánh xạ phi tuyến để ánh xạ các điểm đầu vào sang một không gian khác có số chiều cao hơn. Trong không gian này, các điểm dữ liệu trở thành khả năng tách tuyến tính, hoặc có khả năng phân tách ít lỗi hơn so với trường hợp sử dụng không gian ban đầu.

Một cách tổng quát nếu ta gọi và là một nhân K() thì ta có công thức:

## Các công cụ xây dựng đồ án

1. **Ngôn ngữ lập trình**

Ngôn ngữ lập trình được sử dụng trong đây bài đồ án này là Python.

Python là ngôn ngữ lập trình bậc cao cho các mục đích lập trình đa năng do Guido van Rossum tạo ra và lần đầu ra mắt vào năm 1991. Python được thiết kế với ưu điểm mạnh là dễ đọc, dễ học, dễ nhớ.

Python hoàn toàn tạo kiểu động và dùng cơ chế cấp phát bộ nhớ động. Python sử dụng hệ thống kiểu *duck typing*, còn gọi là *latent typing* (tự động xác định kiểu). Có nghĩa là, Python không kiểm tra các ràng buộc về kiểu dữ liệu tại thời điểm dịch, mà là tại thời điểm thực thi. Khi thực thi, nếu một thao tác trên một đối tượng bị thất bại, thì có nghĩa là đối tượng đó không sử dụng một kiểu thích hợp.

Python được phát triển trong một dự án mở, do tổ chức phi lợi nhuận Python Software Foundation.

1. **Thư viện Tensorflow**

Là một thư viện mã nguồn mở dành cho máy học. Tensorflow có thể sử dụng được trong những nhiệm vụ đặc biệt tập trung vào đào tạo và suy luận các các mạng nơ-ron

Tensorflow được phát triển bởi nhóm Google Brain để sử dụng nội bộ trong Google. Và thư viện đó được cấp giấy chứng nhận Apache License 2.0 vào năm 2015.

Apache License là một giấy phép phần mềm miễn phí được cấp bởi Apache Software Foundation. Giấy phép đó cho phép người dùng sử dụng phần mềm với bất kỳ mục đích nào, phân phối, sửa đổi và phân phối các phiên bản đã sửa đổi của phần mềm theo các điều khoản mà không cần quan tâm đến tiền bản quyền

Keras là một API học sâu được viết bằng Python, chạy trên nền tảng máy học Tensorflow. Keras được phát triển với trọng tâm là cho phép thử nghiệm nhanh. Có thể đi từ ý tưởng đến kết quả nhanh nhất có thể.

1. **Database**

SQLite là một thư viện ngôn ngữ C triển khai một công cụ cơ sở dữ liệu SQL nhỏ, nhanh, khép kín, có độ tin cậy cao, đầy đủ tính năng. SQLite là công cụ cơ sở dữ liệu được sử dụng nhiều nhất trên thế giới. SQLite được tích hợp vào tất cả điện thoại di động và hầu hết các máy tính và đi kèm trong vô số ứng dụng khác mà mọi người sử dụng hàng ngày.

1. **Xây dựng GUI**

Tkinter là mô-đun python có sẵn được sử dụng để tạo các ứng dụng GUI. Đây là một trong những mô-đun được sử dụng phổ biến nhất để tạo các ứng dụng GUI bằng Python vì nó đơn giản và dễ làm việc. Bạn không cần phải lo lắng về việc cài đặt riêng mô-đun Tkinter vì nó đã đi kèm với Python. Nó cung cấp giao diện hướng đối tượng cho bộ công cụ Tk GUI.

Opencv

OpenCV (Open Source Computer Vision Library) là một thư viện phần mềm máy tính và thị giác máy tính mã nguồn mở. OpenCV được xây dựng để cung cấp một cơ sở hạ tầng chung cho các ứng dụng thị giác máy tính và để tăng tốc việc sử dụng cảm giác máy trong các sản phẩm thương mại.

Opencv có hơn 2500 thuật toán được tối ưu hóa, trong đó bao gồm một bộ toàn diện các thuật toán máy học và thị giác máy tính cổ điển và hiện đại. Các thuật toán này có thể được sử dụng để phát hiện và nhận dạng khuôn mặt, xác định đối tượng, phân loại hành động của con người trong video, theo dõi chuyển động của camera, theo dõi đối tượng chuyển động, trích xuất mô hình 3D của đối tượng, tạo ra các đám mây điểm 3D từ camera âm thanh nổi, ghép các hình ảnh lại với nhau để tạo ra độ phân giải cao hình ảnh của toàn bộ cảnh, tìm các hình ảnh tương tự từ cơ sở dữ liệu hình ảnh, loại bỏ mắt đỏ khỏi hình ảnh được chụp bằng đèn flash, theo dõi chuyển động của mắt, nhận dạng phong cảnh và thiết lập các điểm đánh dấu để phủ lên nó bằng thực tế tăng cường,

OpenCV có hơn 47 nghìn người dùng cộng đồng và ước tính số lượt tải xuống vượt quá 18 triệu. Thư viện được sử dụng rộng rãi trong các công ty, nhóm nghiên cứu và các cơ quan chính phủ.

# BÀI TOÁN NHẬN DIỆN BIỂN SỐ XE

## Tìm hiểu về biển số xe

Ở Việt Nam, biển kiểm soát xe cơ giới hay còn gọi biển số là tấm biển gắn trên mỗi xe cơ giới, được cơ quan công an cấp ki mua xe mới hoặc chuyển nhượng xe.

Biển số xe được làm bằng kim loại, có dạng hình chữ nhật trên đó có những con số, chữ cho biết: vùng và địa phương quản lý

Phân loại biển số xe:

Màu sắc

Nền biển trắng, chữ màu đen là thuộc sở hữu cá nhân và xe của các doanh nghiệp.



Hình 2.1 Biển số nền trắng, chữ đen

Nền biển xanh, chữ màu trắng là biển số xe các cơ quan hành chính công dân sự như các cơ quan của Đảng, văn phòng chủ tịch nước,…



Hình 2.2 Biển số nền xanh dương, chữ trắng

Nền đỏ chữ trắng sẽ dành cho xe đơn vị quân đội, xe của các doanh nghiệp quân đội



Hình 2.3 Biển số nền đỏ, chữ trắng

Biển nền vàng chữ đen là biển số xe dành cho các doanh nghiệp kinh doanh vận tại.



Hình 2.4 Biển số nền vàng, chữ đen

## Khái quát chung về nhận dạng biển số xe

Hệ thống sẽ nhận dạng hình ảnh và phần mềm sẽ phân tích ảnh đó để lấy ra các ký tự trên biển số xe.



Hình 2.5 Sơ đồ nhận diện biển số

Tách biển số: Khối chức năng này sẽ tách biển số ra từ ảnh bằng mạng WPOD. Kết quả của khối này đó chính là ảnh biển số đã được điều chỉnh theo phương nhìn trực diên.

Phân đoạn ký tự: Sau khi tách biển số ra khỏi ảnh, em bắt đầu phân đoạn ký tự. Khối này thực hiện tách từng ký tự, tạo thành 1 tập ảnh riêng biệt với mục đích sử dụng để nhận diện ký tự. Ảnh của mỗi ký tự sẽ là ảnh trắng đen

Nhận dạng ký tự: Sau khi tách biển số và tách ký tự tiếp đến là nhận dạng ký tự. Ở khối này, tất cả các ký tự sau khi phân đoạn sẽ được đưa vào để nhận diện. Kết quả cuối cùng là chuỗi ký tự.

Sau khi nhận dạng ký tự, hệ thống sẽ tự động ghi nhận ngày vào ngày ra của xe.

## Phát hiện biển số

### Sử dụng phương pháp biến đổi hình thái học

Đầu tiên, ảnh cần đưa về ảnh xám do để giảm thông tin không cần thiết.

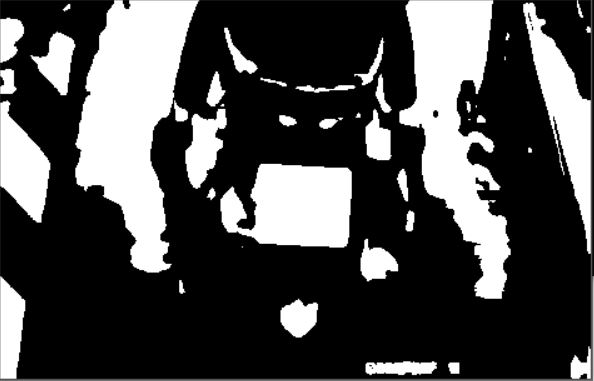
Bước tiếp theo cần, ảnh cần được làm mịn để giữ lại những đường biên của ảnh ta cần dùng phương thức cv2.bilateralFilter và sử dụng cv2.equalizeHist để cân bằng sáng cho hình biển số xe từ đây.



Hình 2.6 Hình biển số xe sau khi làm mịn và cân bằng sáng

Sau khi sử dụng những phương thức trên ảnh biển số xe đã được làm nổi bật những cạnh.

Tiếp đến ảnh cần được đưa về ảnh nhị phân tức là ảnh chỉ gồm màu đen trắng. Sử dụng hàm cv2.threshold để phân ngưỡng để nhị phân bức ảnh biển số xe



Hình 2.7 Hình biển số xe sau khi nhị phân

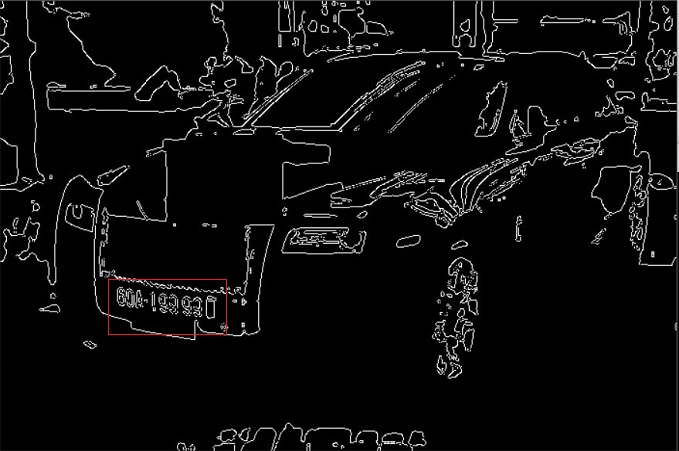
Tiếp đến cần vẽ ra những đương bao xung quanh ảnh đã được nhị phân hóa vừa rồi. Ta sẽ dùng hàm cv2.Canny. Nếu các đường thẳng chưa rõ nét ta dùng phương thức cv2.dilate để làm nở rộng những đường bao vừa tìm được.



Hình 2.8 Đường bao của hình biển số xe

Đưa ảnh đường bao về contour và tìm trên đó nếu contour đó là hình có bốn góc có thể cho là biển số xe.

Hạn chế: Nếu sử dụng cách này biển số xe phải nổi bật nếu không thì sẽ không tìm được hình chữ nhật.

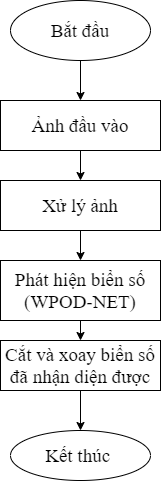


Hình 2.9Biển số xe bị mất khi tìm đường bao

Ngay tại vị trí ô vuông màu đỏ cần có hình chữ nhật ở đó nhưng do trong quá trình nhị phân thì tấm kim loại được làm nổi bật nên đã che viền hình chữ nhật của biển số xe.

Có thể thấy rằng việc phát hiện biển số xe bằng phương pháp biến đổi hình thái học sẽ phụ thuộc rất nhiều đến điều kiện xung quanh cho nên em đã quyết định tìm hiểu về mạng WPOD để nhận diện biển số xe

### Sử dụng mạng WPOD



Hình 2.10 Sơ đồ khối phát hiện biển số xe

Sau khi kết thúc em sẽ nhận được hình ảnh biển số xe và từ biển số xe này và từ biển số xe này em sẽ tiến hành tách ký tự.

Ưu điểm

So với nhận diện bằng opencv thì nhận diện bằng mạng WPOD cho kết quả chính xác hơn ít bị ảnh hưởng bởi các yếu tố xung quanh.

Khả năng cải thiện phát hiện của mạng WPOD biển số xe cũng sẽ tốt hơn.

Nhược điểm

Khó cài đặt, chi phí cao hơn nhận diện bằng opencv.

Tốc độ xử lý chậm hơn.

## Tách ký tự

Sau khi nhận được ảnh biển số xe cần tách riêng biệt từng ký tự với mục đích để nhận diện ký tự chính xác nhất. Trước khi tách ký tự em cần xử lý ảnh để có thể tách ký tự

Đầu tiên, em cần biến ảnh biển số xe từ ảnh màu thành ảnh xám.

Áp dụng tính năng làm mờ Gaussian và nhị phân để hiện thị các ký tự trên xe.



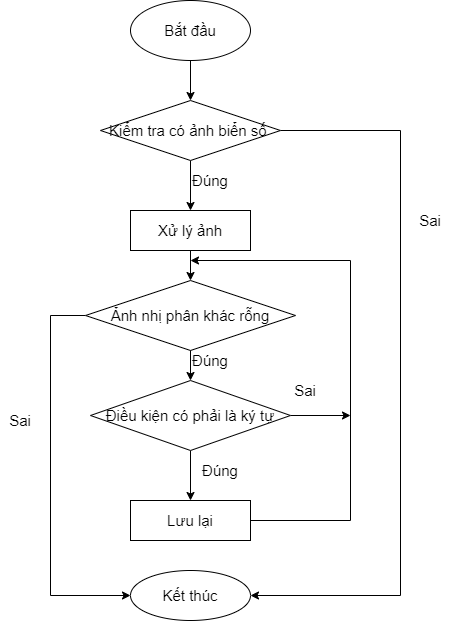
Hình 2.11 Ảnh biển số được nhị phân để tách ký tự

Có rất nhiều pixel trắng trong ảnh nhị phân. Em cần xác định những đốm trắng đó chính là ký tự trên biển số xe. Điều này có thể thực hiện được bằng cách áp dụng thuật toán connected-component.

Phương thức connected-compoment trả về một nhãn, một mảng numpy có cùng kích thước với ảnh tạo ngưỡng sẽ được tạo. Phần tử trả về 0 khi đó chỉ là nền và lớn hơn 0 khi nó là một thành phần kết nối. Mỗi thành phần kết nối tương ứng với một đốm trắng.

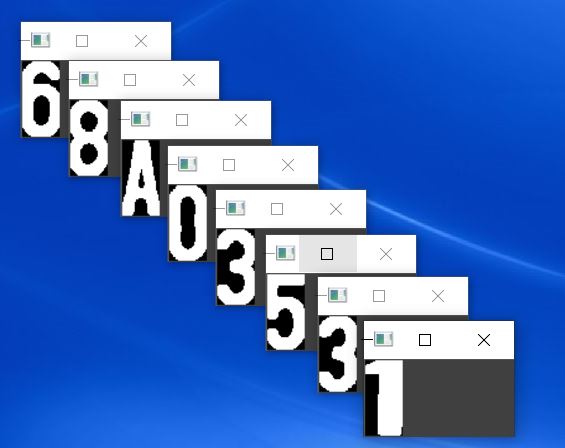
Từ ảnh nhị phân ta thấy rằng số lượng pixel cho mỗi ký tự rơi vào một phạm vi nhất định. Tử lệ cạnh chiều dài/ rộng <= 8.6 và >0.4 và tỉ lệ điểm trắng > 0.35 và <0.85

Sắp xếp theo thứ tự các ký tự vừa cắt được. Với điều kiện nếu độ chênh lệch tung độ > 20 thì tiến hành sắp xếp lại.



Hình 2.12 Sơ đồ khối tách ký tự

Sau khi xử lý ảnh và xét những điều kiện dựa trên thử nghiệm thực tế em sẽ có được kết quả như hình dưới đây:



Hình 2.13 Kết quả sau khi thực hiện tách ký tự

## Nhận diện ký tự

Tiếp theo đó em lấy hình ảnh những ký tự của hình 3.10 để tiến hành nhận diện ký tự.

Như ở phần đầu em đã giới thiệu về thuật toán dùng để nhận diện ký tự chính là SVM và Hitogram insector kernel sẽ được sử dụng để phân loại trong bài báo cáo này.

### Hitogram insector kernel

Giao điểm biểu đồ là một kỹ thuật được đề xuất để lập chỉ mục màu với ứng dụng để nhận dạng đối tượng. Được biết rằng giao điểm biểu đồ là một biểu diễn hiệu quả giúp có thể xây dựng nhận dạng dựa trên màu một cách hợp lý. Giao điểm biểu thị màu đo mức độ giống nhau giữa hai biểu đồ màu. Nó rất phù hợp để đối phó với các trường hợp thay đổi tỷ lệ. Chúng ta biểu thị với A và B các biểu đồ của hình ảnh và . Cả 2 biểu đồ gồm m bin và bin thứ i và i = 1,…,m được kí hiệu lần lượt là và . Giả sử và có cùng kích thước, khi đó ta có: = N và = N. Cuối cùng ta có:

Ưu điểm

Tốc độ nhanh.

Đơn giản, dễ cài đặt.

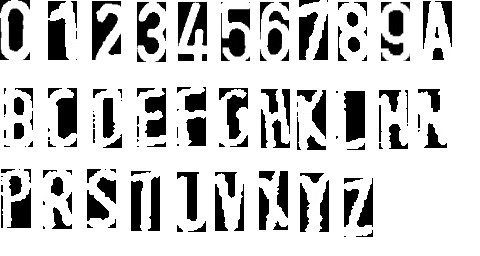
Khuyết điểm:

Độ chính xác không cao bằng những kernel khác.

Chỉ sử dụng cho hình ảnh

### Huấn luyện thuật toán SVM

Tập dữ liệu bao gồm các ảnh 30x60 pixel, tập train bao gồm số tù 0->9, tập chứ A->Z bỏ các ký tự I, J, O, Q nhãn sẽ được lấy theo tên folder chứa file ảnh và được đánh số theo bảng ASCII



Hình 2.14 Các kí tự để huyến luyện thuật toán SVM

# ỨNG DỤNG

## Yêu cầu bài toán

Một phần mềm cho phép đọc được biển số xe và từ đó cho biết xe nào vào bãi đỗ xe và ra khỏi bãi đỗ xe. Tiền phí sẽ được tính theo giờ. Tiền phí theo giờ có thể thay đổi được vì có những khác nhau về giá giữa ngày đặc biệt và những ngày bình thường. Sau khi xe ra, phần mềm phải ghi nhận được những thông tin như sau của một chiếc xe: Tiền phí đậu xe mà chiếc xe đó phải trả, giờ vào và giờ ra.

## Thiết kế cơ sở dữ liệu

Bảng xe vào: Đây sẽ là bảng chứa thông tin của xe vào như: Biển số xe, ngày giờ vào của xe mỗi biển số xe vào là duy nhất không được phép trùng.

Bảng xe ra: Đây sẽ là nơi chứa thông tin ra của xe ra: Biển số xe, ngày giờ vào, ngày giờ ra, tổng tiền phải trả. Sau ghi nhận lại ở bảng xe ra cần có câu lệnh xóa biển số xe đó ở bảng xe vào.

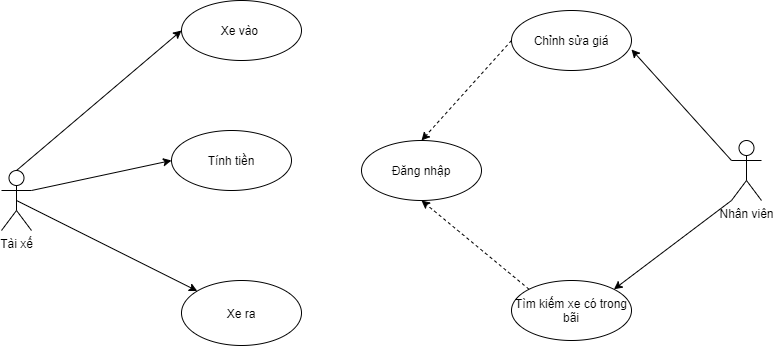
Bảng chi phí: Đây là nơi mà chứa thông tin phí của bãi giữ xe đang được áp dụng. Mỗi lần thay đổi đều có thể ghi nhận lại.

## Sơ đồ UseCase

Sơ đồ UseCase biểu diễn chức năng hệ thống. Từ các yêu cầu của hệ thống, biểu đồ usecase chỉ ra hệ thống cần thực hiện những điều gì để đáp ứng nhu cầu của người sử dụng hệ thống.

Tài xế là người sẽ lái xe qua cổng ra, cổng vào cửa bãi đỗ xe để hệ thống có thể ghi nhận xe đã vào hay ra.

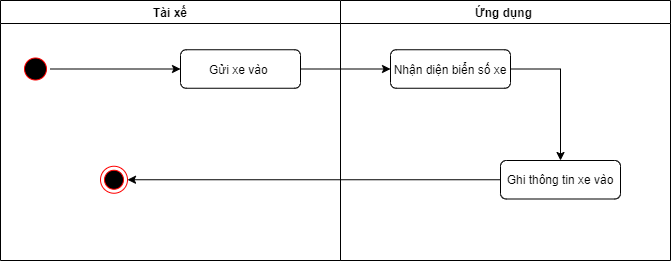
Nhân viên chính là người làm việc hay có thể chủ của bãi giữ xe, người này có thể điều chỉnh giá cả, xem thông tin của chiếc xe.



Hình 3.1 Sơ đồ use case hệ thống

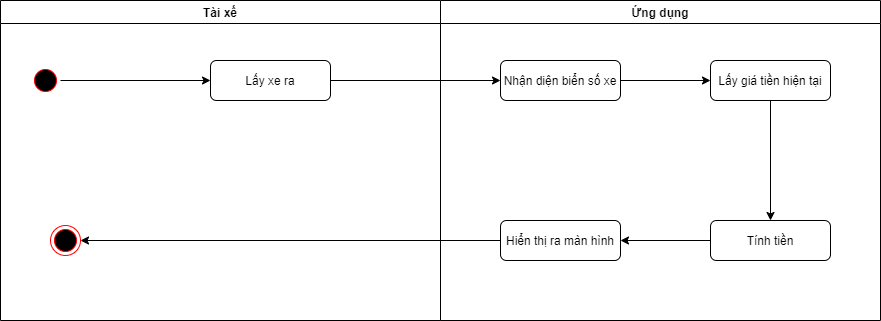
## Sơ đồ hoạt động

Sơ đồ hoạt động xe vào



Hình 3.2 Sơ đồ hoạt động khi xe vào

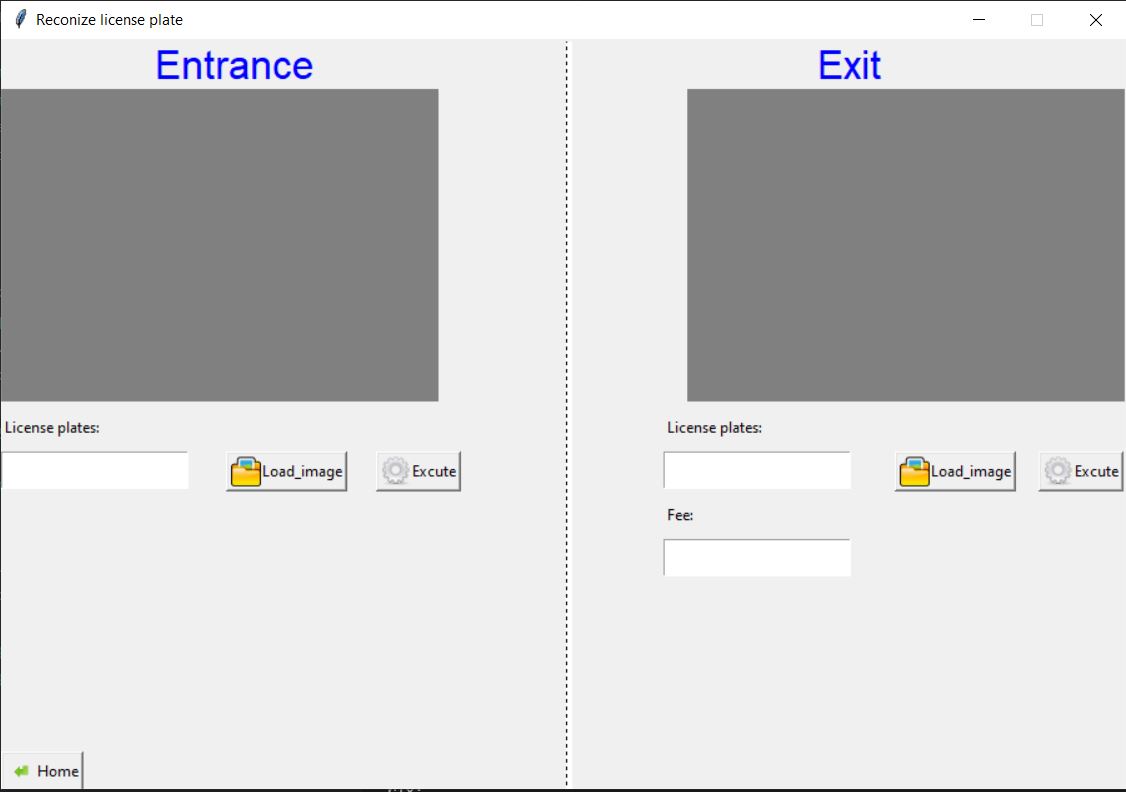
Sơ đồ hoạt động xe ra



Hình 3.3 Sơ đồ hoạt động khi xe ra

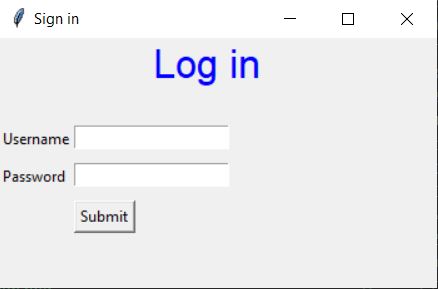
## Giao diện phần mềm

Màn hình kiểm soát xe của hệ thống



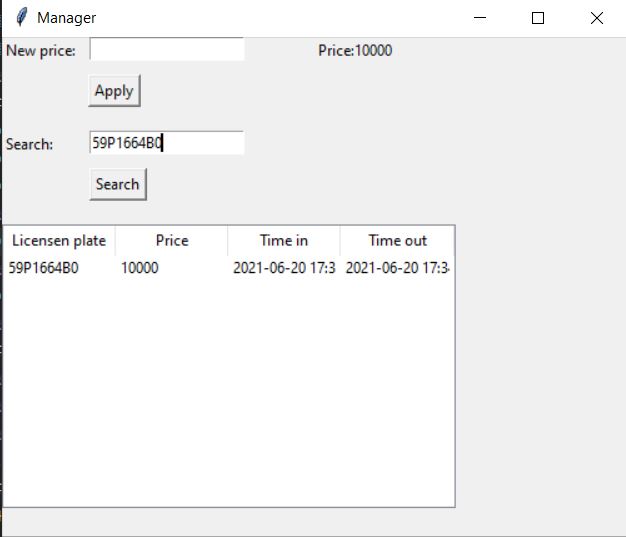
Hình 3.4 Giao diện nhận diện biển số xe vào- ra

Màn hình đăng nhập khi muốn chỉnh sửa giá hoặc tìm kiếm lịch sử ra vào của xe.



Hình 3.5 Màn hình đăng nhập khi muốn thay đổi giá

Màn hình chỉnh sửa giá đậu xe và lịch sử xe ra và



Hình 3.6 Màn hình cho phép điều chỉnh giá và lịch sử ra vào

# Tổng kết

## Kết quả đạt được

Hiểu thêm về deep learning, mạng nơ-ron tích chập, machine learning.

Biết thêm kiến thức về Tensorflow, Keras, Google Lab để huấn luyện mạng.

Cải thiện kỹ năng viết báo cáo.

Xây dựng được một số tính năng cho phần mềm: Đã nhận diện được biển số, ghi nhận được xe ra, xe vào, tính tiền theo giờ, cho phép chỉnh sửa giá theo giờ

## Hạn chế

Vẫn chưa thể tách ký tự khi biển số xe có hắt ánh sáng, bị dơ.

Tỷ lệ đúng chưa được cao

Tập dữ liệu nhận diện ký tự thiếu những trường hợp bị che khuất.

Chưa đưa ra được một hệ thống thực tiễn.

## Hướng phát triển

Khắc phục những hạn chế kể trên.

Thêm phần nhận diện loại xe để có thể phân loại tính toán chi phí một cách hợp lý hơn.

Chuyển từ ứng dụng local thành ứng dụng web.

# PHỤ LỤC

## Hướng dẫn cài đặt

Cài đặt python

Chạy câu lệnh: *pip install tensorflow* để cài đặt thư viện tensorflow

Sử dụng câu lệnh : *git clone* [*https://github.com/datdao999/graduation\_project.git*](https://github.com/datdao999/graduation_project.git)

Giải nén file vừa tải và vào folder vừa giải nén được

Chạy câu lệnh: *python gui.py*

## Hướng dẫn sử dụng

Sau khi chạy câu lệnh *python gui.py* giao diện phần mềm sẽ hiện ra:

Bấm vào nút ***recognize license*** để hiện lên giao diện nhận diện biển số xe.

Bấm vào nút ***Manager price*** để có thể điều chỉnh giá, tìm lịch sử xe ra vào.

Ở giao diện nhận diện biển số xe:

Nút Load Image để load ảnh lên.

Nút Excute để bắt đầu nhận diện ảnh.

Ở ô License plates chính là kết quả sau khi nhận diện biển số xe.

Khi nhấn vào nút Manager price, trước khi vào giao diện điều chỉnh giá thì cần phải qua bước đăng nhập. Cần đăng nhập đúng tài khoản mật khẩu đã cung cấp.

# Tài liệu tham khảo

|  |  |
| --- | --- |
| [1] | ECCV paper 2018 License Plate Detection and Recognition in Unconstrained Scenarios, S´ergio Montazzolli Silva, Cl´audio Rosito Jung |
| [2] | Detect and Recognize Vehicle’s License Plate with Machine Learning and Python, Quang Nguyen, 2020, https://medium.com/ |
| [3] | Introduce to Data Mining, Tan, Steinbach, Kumar,2006 |
| [4] | Histogram Intersection Kernel For Image Classification, Anna |
| [5] | Convolutional Neural Network, https://phamdinhkhanh.github.io/2019/08/22/convolutional-neural-network.html |
| [6] | Support Vector Machine, https://machinelearningcoban.com/2017/04/09/smv/ |
| [7] | Python - GUI Programming (Tkinter), https://www.tutorialspoint.com/python/python\_gui\_programming.htm |