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

**BỘ MÔN CÔNG NGHỆ THÔNG TIN**

**BÁO CÁO ĐỒ ÁN TỐT NGHIỆP**

**ĐỀ TÀI: NGHIÊN CỨU MẠNG WPOD VÀ XÂY DỰNG PHẦN MỀM NHẬN DẠNG BIỂN SỐ 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, năm 2021**

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

**BỘ MÔN CÔNG NGHỆ THÔNG TIN**

**BÁO CÁO ĐỒ ÁN TỐT NGHIỆP**

**ĐỀ TÀI: NGHIÊN CỨU MẠNG WPOD VÀ XÂY DỰNG PHẦN MỀM NHẬN DẠNG BIỂN SỐ 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, năm 2021**

TRƯỜNG ĐẠI HỌC GIAO THÔNG VẬN TẢI **CỘNG HÒA XÃ HỘI CHỦ NGHĨA 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 MẠNG WPOD VÀ XÂY DỰNG PHẦN MỀM NHẬN DẠNG BIỂN SỐ XE.**

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

* Xây dựng tính năng nhận dạng 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 dạng ký tự.

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

* Giới thiệu các bước về nhận dạng biển số
* Giới thiệu thuật toán dùng để phát hiện biển số xe.
* Giới thiệu thuật toán dùng để nhận dạng 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 dạng 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.
2. **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 dạng.

1. **Giảng 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 3 năm 2021**  **Trưởng BM Công Nghệ Thông Tin** | **Đã giao nhiệm vụ TKTN**  **Giảng viên hướng dẫn** |
| **ThS.Trần Phong Nhã** | **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 MẠNG WPOD VÀ XÂY DỰNG PHẦN MỀM NHẬN DẠNG BIỂN SỐ 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 **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 **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!

NHẬN XÉT CỦA GIẢNG VIÊN

|  |
| --- |
| **Tp. Hồ Chí Minh, ngày tháng năm 2021**  **Giảng viên hướng dẫn**  **ThS. Trần Phong Nhã** |

MỤC LỤC

[LỜI CẢM ƠN i](#_Toc79672763)

[NHẬN XÉT CỦA GIẢNG VIÊN ii](#_Toc79672764)

[MỤC LỤC iii](#_Toc79672765)

[DANH MỤC CHỮ VIẾT TẮT vi](#_Toc79672766)

[BẢNG BIỂU, SƠ ĐỒ, HÌNH VẼ vii](#_Toc79672767)

[LỜI GIỚI THIỆU ix](#_Toc79672768)

[GIỚI THIỆU ĐỀ TÀI 1](#_Toc79672769)

[Đặt vấn đề 1](#_Toc79672770)

[Mục đích đề tài 1](#_Toc79672771)

[Đối tượng nghiên cứu 1](#_Toc79672772)

[Phạm vi nghiên cứu 1](#_Toc79672773)

[CHƯƠNG 1: CƠ SỞ LÝ THUYẾT 3](#_Toc79672774)

[1.1 Machine Learning 3](#_Toc79672775)

[1.1.1 Giới thiệu về Machine Learning 3](#_Toc79672776)

[1.1.2 Phân nhóm các thuật toán Machine Learning 6](#_Toc79672777)

[1.1.3 Các bước thực hiện Machine Learning 9](#_Toc79672778)

[1.1.4 Các ứng dụng 11](#_Toc79672779)

[1.1.5 Các loại giải thuật 12](#_Toc79672780)

[1.2 Tự động nhận dạng biển số xe 13](#_Toc79672781)

[1.2.1 Khái niệm 13](#_Toc79672782)

[1.2.2 Khó khăn 13](#_Toc79672783)

[1.3 Tìm hiểu mạng nơ-ron tích chập 14](#_Toc79672784)

[1.3.1 Giới thiệu mạng nơ ron nhân tạo 14](#_Toc79672785)

[1.3.2 Một số kiểu mạng nơ-ron 15](#_Toc79672786)

[1.3.3 Giới thiệu mạng nơ-ron tích chập 16](#_Toc79672787)

[1.3.4 Cấu tạo mạng nơ-ron tích chập (CNN) 17](#_Toc79672788)

[1.4 Tìm hiểu mạng WPOD 19](#_Toc79672789)

[1.4.1 Giới thiệu mạng WPOD 19](#_Toc79672790)

[1.4.2 Kiến trúc Mạng WPOD-NET 20](#_Toc79672791)

[1.5 Giới thiệu thuật toán SVM 20](#_Toc79672792)

[1.6 Các công cụ xây dựng đồ án 21](#_Toc79672793)

[1.6.1 Ngôn ngữ lập trình Python 21](#_Toc79672794)

[1.6.2 Thư viện Tensorflow 22](#_Toc79672795)

[1.6.3 Hệ quản trị cơ sở dữ liệu 22](#_Toc79672796)

[1.6.4 Xây dựng GUI 23](#_Toc79672797)

[1.6.5 Opencv 24](#_Toc79672798)

[CHƯƠNG 2: BÀI TOÁN NHẬN DẠNG BIỂN SỐ XE 25](#_Toc79672799)

[2.1 Tìm hiểu về biển số xe 25](#_Toc79672800)

[2.2 Khái quát chung về nhận dạng biển số xe 27](#_Toc79672801)

[2.3 Phát hiện biển số 27](#_Toc79672802)

[2.3.1 Sử dụng phương pháp biến đổi hình thái học 27](#_Toc79672803)

[2.3.2 Huấn luyện mạng WPOD 30](#_Toc79672804)

[2.3.3 Sử dụng mạng WPOD 35](#_Toc79672805)

[2.4 Tách ký tự 36](#_Toc79672806)

[2.5 Nhận dạng ký tự 39](#_Toc79672807)

[2.5.1 SVM tuyến tính (linear) 39](#_Toc79672808)

[2.5.2 SVM tuyến tính lề mềm (Soft margin) và phi tuyến tính (non-linear) 41](#_Toc79672809)

[2.5.3 Hitogram Intersection Kernel 42](#_Toc79672810)

[2.5.4 Huấn luyện thuật toán SVM 43](#_Toc79672811)

[CHƯƠNG 3: PHẦN MỀM THỬ NGHIỆM 44](#_Toc79672812)

[3.1 Yêu cầu bài toán 44](#_Toc79672813)

[3.1.1 Yêu cầu chức năng 44](#_Toc79672814)

[3.1.2 Yêu cầu phi chức năng 44](#_Toc79672815)

[3.2 Chức năng hệ thống 45](#_Toc79672816)

[3.2.1 Sơ đồ use case 45](#_Toc79672817)

[3.2.2 Phân tích module phần mềm 46](#_Toc79672818)

[3.3 Sơ đồ hoạt động 46](#_Toc79672819)

[3.4 Thiết kế cơ sở dữ liệu 47](#_Toc79672820)

[3.4.1 Tìm hiểu về SQLite3 47](#_Toc79672821)

[3.4.2 Các bảng phục vụ phần mềm 49](#_Toc79672822)

[3.5 Thiết kế giao diện phần mềm 50](#_Toc79672823)

[3.5.1 Các thành phần cơ bản trong Tkinter 50](#_Toc79672824)

[3.5.2 Giao diện phần mềm 53](#_Toc79672825)

[KẾT LUẬN 58](#_Toc79672826)

[Kết quả đạt được 58](#_Toc79672827)

[Hạn chế 58](#_Toc79672828)

[Hướng phát triển 59](#_Toc79672829)

[PHỤ LỤC 60](#_Toc79672830)

[Hướng dẫn cài đặt 60](#_Toc79672831)

[Hướng dẫn sử dụng 60](#_Toc79672832)

[TÀI LIỆU THAM KHẢO 61](#_Toc79672833)

# DANH MỤC CHỮ VIẾT TẮT

|  |  |  |  |
| --- | --- | --- | --- |
| 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 | Graphical User Interface | Giao diện đồ họa người dùng | GUI |
| 6 | Optical Character Recognition | Nhận dạng ký tự | OCR |
| 7 | Machine Learning | Máy học | ML |
| 8 | Deep Learning | Học sâu | DL |
| 9 | Automatic License Plate Recognition | Tự động nhận diện biển số xe | ALPR |

**BẢNG BIỂU, SƠ ĐỒ, HÌNH VẼ**

[Hình 1.1 Mối quan hệ giữa AI, Machine Learning và Deep Learning 3](#_Toc79703815)

[Hình 1.2 Mối quan hệ giữa AI, Machine Learning và Deep Learning. 4](#_Toc79703816)

[Hình 1.3 Lựa chọn thuật toán phù hợp trong Machine Learning. 10](#_Toc79703817)

[Hình 1.4 Cấu tạo mạng nơ-ron nhân tạo 14](#_Toc79703818)

[Hình 1.5 Hoạt động tích chập của CNN 16](#_Toc79703819)

[Hình 1.6 Thêm vào đường viền để chia hết 17](#_Toc79703820)

[Hình 1.7 Hoạt động của mạng WPOD 19](#_Toc79703821)

[Hình 1.8 Kiến trúc mạng WPOD 20](#_Toc79703822)

[Hình 1.9 Hoạt động thuật toán SVM 21](#_Toc79703823)

[Hình 2.1 Biển số nền trắng, chữ đen 25](#_Toc79703824)

[Hình 2.2 Biển số nền xanh dương, chữ trắng 25](#_Toc79703825)

[Hình 2.3 Biển số nền đỏ, chữ trắng 26](#_Toc79703826)

[Hình 2.4 Biển số nền vàng, chữ đen 26](#_Toc79703827)

[Hình 2.5 Sơ đồ nhận dạng biển số 27](#_Toc79703828)

[Hình 2.6 Hình biển số xe sau khi làm mịn và cân bằng sáng 28](#_Toc79703829)

[Hình 2.7 Hình biển số xe sau khi nhị phân 28](#_Toc79703830)

[Hình 2.8 Đường bao của hình biển số xe 29](#_Toc79703831)

[Hình 2.9 Biển số xe bị mất khi tìm đường bao 30](#_Toc79703832)

[Hình 2.10 Sơ đồ khối phát hiện biển số xe 35](#_Toc79703833)

[Hình 2.11 Ảnh biển số được nhị phân để tách ký tự 36](#_Toc79703834)

[Hình 2.12 Minh họa cho phương thức connected-compoment 37](#_Toc79703835)

[Hình 2.13 Sơ đồ khối tách ký tự 38](#_Toc79703836)

[Hình 2.14 Kết quả sau khi thực hiện tách ký tự 39](#_Toc79703837)

[Hình 2.15 Phương pháp cực đại hóa lề (margin) 40](#_Toc79703838)

[Hình 2.16 Các kí tự để huấn luyện thuật toán SVM 43](#_Toc79703839)

[Hình 3.1 Sơ đồ Use Case hệ thống 45](#_Toc79703840)

[Hình 3.2 Sơ đồ hoạt động khi xe vào 47](#_Toc79703841)

[Hình 3.3 Sơ đồ hoạt động khi xe ra 47](#_Toc79703842)

[Hình 3.4 Hệ quản trị cơ sở dữ liệu 48](#_Toc79703843)

[Hình 3.5 Giao diện phần mềm DB Broswer for SQLite 49](#_Toc79703844)

[Hình 3.6 Giao diện để kiểm tra Tkinker đã có trong máy 52](#_Toc79703845)

[Hình 3.7 Giao diện khi bắt đầu khởi động chương trình 53](#_Toc79703846)

[Hình 3.8 Giao diện nhận dạng biển số xe vào - ra 54](#_Toc79703847)

[Hình 3.9 Đã nhận dạng được xe vào bãi đỗ xe 55](#_Toc79703848)

[Hình 3.10 Xe ra thành công và được tính tiền đỗ xe 56](#_Toc79703849)

[Hình 3.11 Màn hình đăng nhập khi muốn thay đổi giá 56](#_Toc79703850)

[Hình 3.12 Màn hình cho phép điều chỉnh giá và lịch sử ra vào 57](#_Toc79703851)

# LỜI GIỚI THIỆU

Xử lý và nhận dạng là 1 lĩnh vực được nhiều người quan tâm. Xử lý và nhận dạng 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

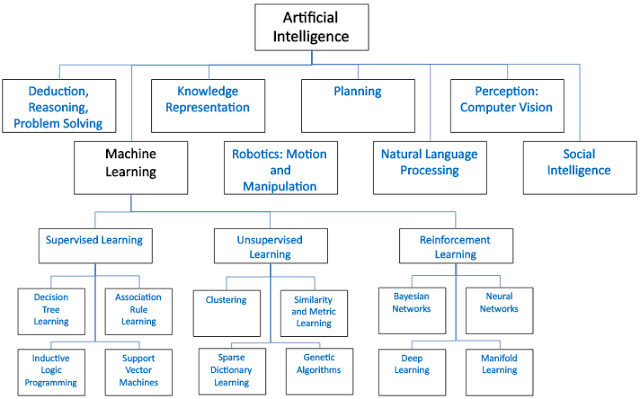
## Machine Learning

### Giới thiệu về Machine Learning

Những năm gần đây, AI – Trí tuệ nhân tạo nổi lên như một bằng chứng của cuộc cách mạng công nghiệp 4.0. Càng ngày, AI càng len lỏi vào mọi lĩnh vực trong đời sống mà có thể chúng ta không nhận ra.

Xe tự lái của Google và Tesla, hệ thống tự nhận dạng khuôn mặt trong ảnh của Facebook, trợ lý ảo Siri của Apple, hệ thống gợi ý sản phẩm của Amazon, hệ thống gợi ý phim của Netflix, máy chơi cờ vây AlphaGo của Google DeepMind, … chỉ là một vài trong vô vàn những ứng dụng của trí tuệ nhân tạo.

Và Machine Learning chính là một tập con trong trí tuệ nhân tạo. ML là một lĩnh vực nhỏ trong ngành khoa học máy tính, có khả năng tự học hỏi dựa tên dữ liệu được đưa vào mà không cần phải lập trình cụ thể.



Hình 1.1 Mối quan hệ giữa AI, Machine Learning và Deep Learning

Thực tế gần đây, khi mà khả năng tính toán của các máy tính ngày càng được nâng lên một tầm cao mới, song song với đó là sự bùng nổ của Big Data, ML đã tiến thêm một bước dài và một lĩnh vực mới được ra đời gọi là Deep Learning (DL).

DL đã giúp máy tính thực thi những việc tưởng chừng như không thể vào 10 năm trước như: phân loại cả ngàn vật thể khác nhau trong các bức ảnh, tự tạo chú thích cho ảnh, bắt chước giọng nói và chữ viết của con người, giao tiếp với con người, hay thậm chí cả sáng tác văn hay âm nhạc, ...



Hình 1.2 Mối quan hệ giữa AI, Machine Learning và Deep Learning.

1. **Định nghĩa Machine Learning**

Máy học (Machine Learning) là nghiên cứu các thuật toán máy tính cải tiến tự động thông qua kinh nghiệm và bằng cách sử dụng dữ liệu. ML được xem như một phần của trí tuệ nhân tạo. Các thuật toán học máy xây dựng một mô hình dựa trên dữ liệu mẫu, được gọi là dữ liệu đào tạ, để đưa ra dự đoán hoặc quyết định mà không cần được lập trình rõ ràng để làm như vậy. [1]

Ví dụ 1: Giả sử như bạn muốn máy tính xác định một tin nhắn có phải là SPAM hay không?

* Tác vụ T: Xác định 1 tin nhắn có phải SPAM hay không?
* Kinh nghiệm E: Xem lại những tin nhắn đánh dấu là SPAM xem có những đặc tính gì để có thể xác định đó là SPAM.
* Độ đo P: Là phần trăm số tin nhắn SPAM được phân loại đúng.

Ví dụ 2: Chương trình nhận dạng số (số từ 0 -> 9)

* T: Là nhận dạng được ảnh chứa ký tự số.
* E: Đặc trưng để phân loại ký tự số từ tập dữ liệu số cho trước.
* P: Độ chính xác của quá trình nhận dạng.

1. **Sự hữu ích của Machine Learning**

Từ lâu đã có nhiều thuật toán ML nổi tiếng nhưng khả năng tự động áp dụng các phép tính phức tạp vào Big Data, lặp đi lặp lại với tốc độ nhanh hơn, chỉ mới phát triển gần đây.

Các ứng dụng của ML đã trở nên quá quen thuộc như:

* Xe tự lái, giảm thiểu tai nạn của Google? Chính là bản chất của ML.
* Các ưu đãi Recommendation Online như của Amazon & Netflix? Ứng dụng của Machine Learning trong cuộc sống hằng ngày.
* Muốn biết người dùng nói gì về bạn trên Twitter? ML kết hợp với sự sáng tạo của quy tắc ngôn ngữ.
* Nhận dạng lừa đảo? Một trong những nhu cầu sử dụng hiển nhiên ngày nay.

1. **Đối tượng sử dụng**

Hầu hết mọi ngành công nghiệp đang làm việc với hàm lượng lớn dữ liệu đều nhận ra tầm quan trọng của công nghệ ML. Những cái nhìn sâu sắc từ nguồn dữ liệu này, sẽ giúp các tổ chức vận hành hiệu quả hơn hoặc tạo được lợi thế cạnh tranh so với các đối thủ.

* Các dịch vụ tài chính

Ngân hàng và những doanh nghiệp hoạt động trong lĩnh vực tài chính sử dụng công nghệ ML với 2 mục đích chính: xác định insights trong dữ liệu và ngăn chặn lừa đảo. Insights sẽ biết được các cơ hội đầu tư hoặc thông báo đến nhà đầu tư thời điểm giao dịch hợp lý. Khai phá dữ liệu cũng có thể tìm được những khách hàng đang có hồ sơ rủi ro cao hoặc sử dụng giám sát mạng để chỉ rõ những tín hiệu lừa đảo.

* Chính phủ

Các tổ chức chính phủ hoạt động về an ninh cộng đồng hoặc tiện ích xã hội sở hữu rất nhiều nguồn dữ liệu có thể khai thác insights. Ví dụ, khi phân tích dữ liệu cảm biến, chính phủ sẽ tăng mức độ hiệu quả của dịch vụ và tiết kiệm chi phí. ML còn hỗ trợ phát hiện gian lận và giảm thiểu khả năng trộm cắp danh tính.

* Chăm sóc sức khỏe

ML là 1 xu hướng phát triển nhanh chóng trong ngành chăm sóc sức khỏe, nhờ vào sự ra đời của các thiết bị và máy cảm ứng đeo được sử dụng dữ liệu để đánh giá tình hình sức khỏe của bệnh nhân trong thời gian thực. Công nghệ ML còn giúp các chuyên gia y tế xác định những xu hướng hoặc tín hiệu để cải thiện khả năng điều trị, chẩn đoán bệnh.

* Marketing và sales

Dựa trên hành vi mua hàng trước đây, các trang website sử dụng ML phân tích lịch sử mua hàng, từ đó giới thiệu những vật dụng mà bạn có thể sẽ quan tâm và yêu thích. Khả năng tiếp nhận dữ liệu, phân tích và sử dụng những dữ liệu đó để cá nhân hóa trải nghiệm mua sắm hoặc thực hiện chiến dịch Marketing chính là tương tai của ngành bán lẻ.

* Dầu khí

Tìm kiếm những nguồn nguyên liệu mới. Phân tích các mỏ dầu dưới đất. Dự đoán tình trạng thất bại của bộ cảm biến lọc dầu. Sắp xếp các kênh phân phối để đạt hiệu quả và tiết kiệm chi phí. Có thể nói, số lượng các trường hợp sử dụng ML trong ngành công nghiệp này cực kì lớn và vẫn ngày càng mở rộng.

* Vận tải

Phân tích dữ liệu để xác định mô hình và các xu hướng là trọng tâm trong ngành vận tải vì đây là ngành phụ thuộc vào khả năng tận dụng hiệu quả trên mỗi tuyến đường và dự đoán các vấn đề tiềm tàng để gia tăng lợi nhuận. Các chức năng phân tích dữ liệu của ML đóng vai trò quan trọng với các doanh nghiệp vận chuyển, vận tải công cộng và các tổ chức vận chuyển khác*.*

### Phân nhóm các thuật toán Machine Learning

Các thuật toán ML thường được chia làm 4 nhóm:

- Supervise Learning (Học có giảm sát)

- Unsupervised Learning (Học không giám sát)

- Semi-supervised Lerning (Học bán giám sát)

- Reinforcement Learning (Học củng cố)

1. **Supervised Learning (Học có giám sát)**

Supervised Learning là thuật toán dự đoán đầu ra (outcome) của một dữ liệu mới dựa trên các cặp (input, outcome) đã biết từ trước. Cặp dữ liệu này còn được gọi là (data, label), tức (dữ liệu, nhãn). Supervised Learning là nhóm phổ biến nhất trong các thuật toán Machine Learning. [1]

Một cách toán học, Supervised Learning là khi chúng ra có một tập hợp biến đầu vào X={x1,x2,…,xN} và một tập hợp nhãn tương ứng Y={y1,y2,…,yN} trong đó xi,yi là các vector. Các cặp dữ liệu biết trước (xi,yi)∈X×Y được gọi là tập training data (dữ liệu huấn luyện). Từ tập traing data này, chúng ta cần tạo ra một hàm số ánh xạ mỗi phần tử từ tập X sang một phần tử (xấp xỉ) tương ứng của tập Y: yi≈f(xi), ∀i=1, 2, …,N Mục đích là xấp xỉ hàm số f thật tốt để khi có một dữ liệu xx mới, chúng ta có thể tính được nhãn tương ứng của nó y=f(x).

Ví dụ**:** Trong nhận dạng chữ viết tay, ta có ảnh của hàng nghìn ví dụ của mỗi chữ số được viết bởi nhiều người khác nhau. Chúng ta đưa các bức ảnh này vào trong một thuật toán và chỉ cho nó biết mỗi bức ảnh tương ứng với chữ số nào. Sau khi thuật toán tạo ra một mô hình, tức một hàm số mà đầu vào là một bức ảnh và đầu ra là một chữ số, khi nhận được một bức ảnh mới mà mô hình chưa nhìn thấy bao giờ, nó sẽ dự đoán bức ảnh đó chứa chữ số nào.

Ví dụ này khá giống với cách học của con người khi còn nhỏ. Ta đưa bảng chữ cái cho một đứa trẻ và chỉ cho chúng đây là chữ A, đây là chữ B. Sau một vài lần được dạy thì trẻ có thể nhận biết được đâu là chữ A, đâu là chữ B trong một cuốn sách mà chúng chưa nhìn thấy bao giờ.

Thuật toán Supervised Learning còn được tiếp tục chia nhỏ ra thành hai loại chính:

* Classification (Phân loại)

Một bài toán được gọi là Classification nếu các labelcủainput data được chia thành một số hữu hạn nhóm.

Ví dụ: Gmail xác định xem một email có phải là spam hay không; các hãng tín dụng xác định xem một khách hàng có khả năng thanh toán nợ hay không. Ba ví dụ phía trên được chia vào loại này.

* Regression (Hồi quy)

Nếu label không được chia thành các nhóm mà là một giá trị thực cụ thể.

Ví dụ: Một căn nhà rộng x m2, có y phòng ngủ và cách trung tâm thành phố z km sẽ có giá là bao nhiêu?

1. **Unsupervised Learning (Học không giám sát)**

Trong thuật toán này, chúng ta không biết được outcome hay nhãn mà chỉ có dữ liệu đầu vào. Thuật toán Unsupervised Learning sẽ dựa vào cấu trúc của dữ liệu để thực hiện một công việc nào đó, ví dụ như phân nhóm hoặc giảm số chiều của dữ liệu để thuận tiện trong việc lưu trữ và tính toán.[1]

Một cách toán học, Unsupervised Learning là khi chúng ta chỉ có dữ liệu vào X mà không biết nhãn Y tương ứng.

Những thuật toán loại này được gọi là Unsupervised Learning vì không giống như Supervised Learning, chúng ta không biết câu trả lời chính xác cho mỗi dữ liệu đầu vào. Giống như khi ta học, không có thầy cô giáo nào chỉ cho ta biết đó là chữ A hay chữ B. Cụm không giám sát được đặt tên theo nghĩa này.

Các bài toán Unsupervised Learning được tiếp tục chia nhỏ thành hai loại:

* Clustering (phân nhóm)

Một bài toán phân nhóm toàn bộ dữ liệu XX thành các nhóm nhỏ dựa trên sự liên quan giữa các dữ liệu trong mỗi nhóm.

Ví dụ: Phân nhóm khách hàng dựa trên hành vi mua hàng. Điều này cũng giống như việc ta đưa cho một đứa trẻ rất nhiều mảnh ghép với các hình thù và màu sắc khác nhau, ví dụ tam giác, vuông, tròn với màu xanh và đỏ, sau đó yêu cầu trẻ phân chúng thành từng nhóm. Mặc dù không cho trẻ biết mảnh nào tương ứng với hình nào hoặc màu nào, nhiều khả năng chúng vẫn có thể phân loại các mảnh ghép theo màu hoặc hình dạng.

* Association Rule (Luật kết hợp)

Là bài toán khi chúng ta muốn khám phá ra một quy luật dựa trên nhiều dữ liệu cho trước.

Ví dụ: Những khách hàng nam mua quần áo thường có xu hướng mua thêm đồng hồ hoặc thắt lưng; những khán giả xem phim Spider Man thường có xu hướng xem thêm phim Bat Man, dựa vào đó tạo ra một hệ thống gợi ý khách hàng, thúc đẩy nhu cầu mua sắm.

1. **Semi-Supervised Learning (Học bán giám sát)**

Các bài toán khi chúng ta có một lượng lớn dữ liệu XX nhưng chỉ một phần trong chúng được gán nhãn được gọi là Semi-Supervised Learning. Những bài toán thuộc nhóm này nằm giữa hai nhóm được nêu bên trên. [1]

Một ví dụ điển hình của nhóm này là chỉ có một phần ảnh hoặc văn bản được gán nhãn (ví dụ bức ảnh về người, động vật hoặc các văn bản khoa học, chính trị) và phần lớn các bức ảnh, văn bản khác chưa được gán nhãn được thu thập từ internet. Thực tế cho thấy rất nhiều các bài toán ML thuộc vào nhóm này vì việc thu thập dữ liệu có nhãn tốn rất nhiều thời gian và có chi phí cao. Rất nhiều loại dữ liệu thậm chí cần phải có chuyên gia mới gán nhãn được. Ngược lại, dữ liệu chưa có nhãn có thể được thu thập với chi phí thấp từ Internet.

1. **Reinforcement Learning (Học Củng Cố)**

Reinforcement Learning là các bài toán giúp cho một hệ thống tự động xác định hành vi dựa trên hoàn cảnh để đạt được lợi ích cao. Hiện tại, Reinforcement Learning chủ yếu được áp dụng vào Lý Thuyết Trò Chơi, các thuật toán cần xác định nước đi tiếp theo để đạt được điểm số cao nhất. [1]

**Ví dụ:** [AlphaGo gần đây nổi tiếng với việc chơi cờ vây thắng cả con người](https://gogameguru.com/tag/deepmind-alphago-lee-sedol/). [Cờ vây được xem là có độ phức tạp cực kỳ cao](https://www.tastehit.com/blog/google-deepmind-alphago-how-it-works/) với tổng số nước đi là xấp xỉ 1076110761, so với cờ vua là 1012010120 và tổng số nguyên tử trong toàn vũ trụ là khoảng 10801080.

Vì vậy, thuật toán phải chọn ra 1 nước đi tối ưu trong số hàng nhiều tỉ tỉ lựa chọn, và tất nhiên, không thể áp dụng thuật toán tương tự như IBM Deep.

Về cơ bản, AlphaGo bao gồm các thuật toán thuộc cả Supervised learning và Reinforcement Learning. Trong phần Supervised Learning, dữ liệu từ các ván cờ do con người chơi với nhau được đưa vào để huấn luyện. Tuy nhiên, mục đích cuối cùng của AlphaGo không phải là chơi như con người mà phải thậm chí thắng cả con người.

Vì vậy, sau khi học xong các ván cờ của con người, AlphaGo tự chơi với chính nó với hàng triệu ván chơi để tìm ra các nước đi mới tối ưu hơn. Thuật toán trong phần tự chơi này được xếp vào loại Reinforcement Learning.

### **Các bước thực hiện Machine Learning**

Thực hiện Machine Learning bao gồm các bước như sau:

* **Thu thập và chuẩn bị dữ liệu**

Yếu tố ban đầu cần thiết để thực hiện ML là cần có dữ liệu. Dữ liệu có thể được lấy từ nhiều nguồn khác nhau, có thể ít, có thể nhiều, có thể sạch, có thể nhiều dữ liệu lỗi…Sau khi thu thập, dữ liệu cần được làm sạch.

* **Chọn thành phần**

Với mỗi tập dữ liệu có thể có rất nhiều thành phần, nhưng không phải thành phần nào cũng liên quan tới bài toán mà ta cần giải, việc lựa chọn thành phần (loại bỏ các thành phần không cần thiết) giúp cho việc học của ta trở nên nhanh và hiệu quả hơn. Tuy nhiên, việc lựa chọn đòi hỏi sự thấu hiểu về dữ liệu và bài toán, chủ yếu làm bằng tay và sức người. Sau khi chọn được thành phần, nhiều khi ta quay lại bước 1, tiến hành loại bỏ các dữ liệu không liên quan để thu nhỏ tập dữ liệu.

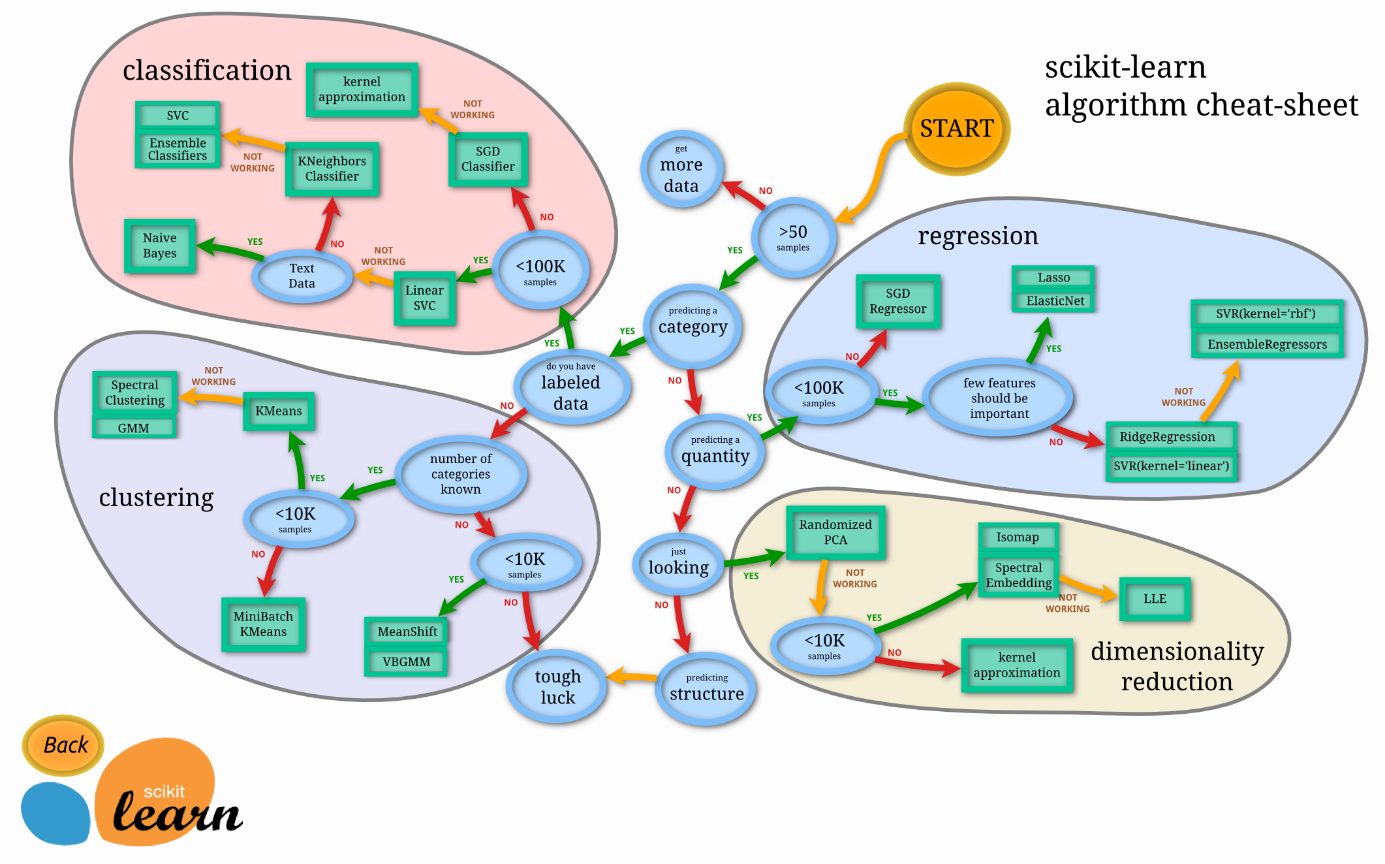
Việc thu nhỏ tập dữ liệu cũng khiến cho việc học của ta tốn ít thời gian hơn, tuy nhiên dữ liệu ít quá cũng khiến việc học đạt độ chính xác không cao, cần cân đối giữa các yếu tố này.

* **Chuẩn hoá dữ liệu**

Nhiều khi dữ liệu của từng thành phần có định dạng, kích thước khác biệt lớn, ví dụ thành phần 1 có dữ liệu trong khoảng [0, 1], thành phần 2 có dữ liệu trong khoảng [-1000, 1000], thành phần 3 có dữ liệu là hình ảnh… Để tăng tốc độ và hiệu quả của việc học, ta cần chuẩn hoá dữ liệu, đưa dữ liệu của tất cả các thành phần về cùng một định dạng (số hoá), và cùng khoảng biến thiên.

* **Chọn thuật toán**

Tuỳ vào dữ liệu, bài toán mà ta lựa chọn thuật toán ML tương ứng. Đôi khi lựa chọn thuật toán cũng cần dựa trên kinh nghiệm. Một số gợi ý cho việc lựa chọn thuật toán là tham khảo của Scikit-learn.



Hình 1.3 Lựa chọn thuật toán phù hợp trong Machine Learning.

* **Chọn parameter cho thuật toán**

Tuỳ mỗi thuật toán mà có nhiều các cách cài đặt khác nhau. Hơn thế nữa các tham số tính toán cũng quyết định không nhỏ tới kết quả tính toán. Vì thế khi sau khi chọn thuật toán thì việc chọn parameter phù hợp với dữ liệu cũng khá quan trọng. Việc chọn Parameter chủ yếu dựa trên kinh nghiệm, nhiều khi có thể sử dụng các thư viện support.

* **Huấn luyện và Đánh giá**

Sau khi chọn được thuật toán (một hoặc nhiều thuật toán) và parameter tương ứng ta cho dữ liệu vào train. Tiến hành cross-validation để điều chỉnh model.

Sau khi train được model, ta đưa data test vào kiểm tra và đánh giá độ chính xác của model vừa train được.

* Phân chia dữ liệu

Thông thường với tập data cho trước ta thường chia làm 3 tập để sử dụng với mục đích khác nhau:

* **Tập huấn luyện** (Training set): dùng để huấn luyện model.
* **Tập kiểm chứng** (Validation set): dùng để đánh giá, điều chỉnh model. Ví dụ như ta dùng tập huấn luyện cho nhiều thuật toán khác nhau rồi dùng tập kiểm chứng để chọn thuật toán phù hợp nhất. Hoặc tìm parameter phù hợp nhất cho một thuật toán cụ thể.
* **Tập kiểm tra** (Test set): dùng để đánh giá model sau khi huấn luyện. Mô hình đã được đánh giá bằng tập test phải là mô hình cuối cùng, không được thay đổi nữa.
* Việc phân chia tỉ lệ giữa 3 tập này được khuyên là 60% - 20% - 20%.
* Nếu model cuối cùng thu được sau khi huấn luyện và kiểm chứng cần phải được đánh giá bằng tập kiểm tra. Nếu kết quả không tốt, cần thực hiện huấn luyện lại từ đầu.

### Các ứng dụng

Có rất nhiều ứng dụng mang tính thực tế cao của máy học mà khó có thể kể hết được. Những ứng dụng dưới đây là những ứng dụng phổ biến và được chọn lọc theo góc nhìn cá nhân.

* Nhận dạng và phát hiện khuôn mặt: Nhận dạng và phát hiện khuôn mặt là ứng dụng khá thú vị của máy học và được áp dụng khá nhiều vào đời sống. Tiêu biểu là tính năng phát hiện khuôn mặt ở máy chụp ảnh. Ứng dụng được phát triển thêm thành phát hiện chớp mắt, phát hiện cười....
* Xe tự lái: Xe tự lái mặc dù phát triển từ đầu thập niên 90 nhưng cho tới nay vẫn còn là vấn đề được nhiều người quan tâm. Các hãng lớn như Google, NVDIA đang nỗ lực để tạo ra một cỗ máy có thể hoàn toàn tự động lái xe và giảm thiểu tai nạn cho con người.
* Phân lớp ảnh: Tìm kiếm ảnh trên Google hiện rất quen thuộc với nhiều người. Ứng dụng của phân lớp ảnh giúp người dùng sử dụng ảnh làm từ khóa tìm kiếm thay thế cho việc tìm kiếm truyền thống trên Google. Bạn upload một ảnh lên và Google sẽ giúp bạn tìm kiếm những thông tin liên quan đến bức ảnh đó.
* Nhận dạng giọng nói: Các trợ lý ảo như Siri, Conrtana hay Google Now là ví dụ điển hình cho nhận dạng giọng nói. Một ví dụ khác nữa là tính năng dịch thuật trực tuyến của Youtube. Với ứng dụng của DL, khả năng dịch thuật chín xác ngôn ngữ từ các video Youtube đang ngày một phát triển vượt bậc.
* Anti-virus: Có thể nhiều người không nghĩ rằng các phần mềm diệt virus lại áp dụng máy học. Tuy nhiên, áp dụng máy học vào để phân tích và dự đoán xu hướng các loại virus sẽ giúp ích rất nhiều trong việc bảo vệ dữ liệu máy tính.

### **Các loại giải thuật**

Các thuật toán học máy được phân loại theo kết quả mong muốn của thuật toán. Các loại thuật toán thường dùng bao gồm:

* Học có giám sát: trong đó, thuật toán tạo ra một hàm ánh xạ dữ liệu vào tới kết quả mong muốn. Một phát biểu chuẩn về một việc học có giám sát là bài toán phân loại: chương trình cần học (cách xấp xỉ biểu hiện của) một hàm ánh xạ một vector [X1, X2, ..., Xn] tới một vài lớp bằng cách xem xét một số mẫu dữ\_liệu - kết\_quả của hàm đó.
* Học không giám sát: mô hình hóa một tập dữ liệu, không có sẵn các ví dụ đã được gắn nhãn.
* Học nửa giám sát: kết hợp các ví dụ có gắn nhãn và không gắn nhãn để sinh một hàm hoặc một bộ phân loại thích hợp.
* Học tăng cường: trong đó, thuật toán học một chính sách hành động tùy theo các quan sát về thế giới. Mỗi hành động đều có tác động tới môi trường, và môi trường cung cấp thông tin phản hồi để hướng dẫn cho thuật toán của quá trình học.
* Chuyển đổi: tương tự học có giám sát nhưng không xây dựng hàm một cách rõ ràng. Thay vì thế, cố gắng đoán kết quả mới dựa vào các dữ liệu huấn luyện, kết quả huấn luyện, và dữ liệu thử nghiệm có sẵn trong quá trình huấn luyện.
* Học cách học: trong đó thuật toán học thiên kiến quy nạp của chính mình, dựa theo các kinh nghiệm đã gặp.

Phân tích hiệu quả các thuật toán học máy là một nhánh của ngành thống kê, được biết với tên lý thuyết học điện toán.

## Tự động nhận dạng biển số xe

### Khái niệm

Nhận dạng biển số tự động (ANPR/ALPR) là một hệ thống có khả năng đọc biển số xe mà không cần sự can thiệp của con người. Nó hoạt động thông qua việc sử dụng tính năng chụp ảnh tốc độ cao, phát hiện các ký tự trong hình ảnh được cung cấp, nhận dạng ký tự để chuyển đổi hình ảnh sang văn bản. Vì vậy, công nghệ này cho kết quả với một tập hợp siêu dữ liệu xác định hình ảnh có chứa biển số xe và văn bản được giải mã liên quan của biển số đó. [2]

ALPR được phát minh vào năm 1976 tại Chi nhánh Phát triển Khoa học Cảnh sát ở Anh. Các hệ thống nguyên mẫu bắt đầu hoạt động vào năm 1979 và các hợp đồng đã được trao để sản xuất các hệ thống công nghiệp, đầu tiên là tại EMI Electronics, và sau đó là Computer Recognition Systems (CRS, nay là một phần của Jenoptik) ở Wokingham, Vương quốc Anh. Các hệ thống thử nghiệm ban đầu đã được triển khai trên đường A1 và tại Đường hầm Dartford. Vụ bắt giữ đầu tiên thông qua việc phát hiện một chiếc xe bị đánh cắp được thực hiện vào năm 1981. Tuy nhiên, ANPR đã không được sử dụng rộng rãi cho đến khi những phát triển mới về phần mềm rẻ hơn và dễ sử dụng hơn được tiên phong trong suốt những năm 1990.

### Khó khăn

Độ phân giải tệp kém, thường là do tấm quá xa nhưng đôi khi do sử dụng máy ảnh chất lượng thấp

Một vật thể che khuất (một phần của) tấm, thường là thanh kéo hoặc bụi bẩn trên tấm

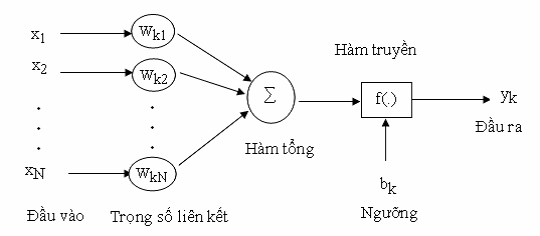
Hình ảnh mờ do chuyển động, do độ phân giải của thiết bị kém, những tác nhân thời tiết bên ngoài không được tốt.

Người sử dụng xe có thể gian lận bằng cách sử dụng tấm phủ hay bình xịt để làm tăng độ tương phản để không đọc được biển số

## Tìm hiểu mạng nơ-ron tích chập

### Giới thiệu mạng nơ ron nhân tạo

Mạng nơ-ron nhân tạo, Artificical Neural Network (ANN) là một mô hình xử lý thông tin phỏng theo cách xử lý thông tin của các hệ nơ-ron sinh học. Nó được tạo nên từ một sô lượng lớn các phần tử (nơ-ron) kết nối với nhau thông qua các liên kết (trọng số liên kết) làm việc như một thể thống nhất để giải quyết một vấn đề cụ thể nào đó. Một mạng nơ-ron nhân tạo được cấu hình cho một ứng dụng cụ thể nào đó (nhận dạng mẫu, phân loại dữ liệu, …) thông qua một quá trình học từ tập các mẫu huấn luyện. Về bản chất quá trình học chính là quá trình hiệu chỉnh trọng số liên kết giữa các nơ-ron. [3]



Hình 1.4 Cấu tạo mạng nơ-ron nhân tạo

Các thành phần cơ bản của một mạng nơ-ron:

Tập đầu vào: Là các tín hiệu vào (input signals) của nơ-ron, các tín hiệu này thường được đưa vào dưới dạng một vector N chiều.

Tập các liên kết: Mỗi liên kết được thể hiện bởi một trọng số liên kết – Synaptic weight. Trọng số liên kết giữa tín hiệu vào thứ j với nơ-ron k thường được ký hiệu . Thông thường, các trọng số này được khởi tạo một cách ngẫu nhiên ở thời điểm khởi tạo mạng và được được cập nhật liên tục trong quá trình học mạng.

Bộ tổng (Summing function): Thường dùng để tính tổng của tích các đầu vào với trọng số liên kết của nó.

Ngưỡng (còn gọi là một độ lệch-bias): Ngưỡng này thường được đưa vào một thành phần của hàm truyền.

Hàm truyền (Tranfer function): Hàm này được dùng để giới hạn phạm vi đầu vào của mỗi nơ-ron. Nó nhận đầu vào là kết quả của hàm tổng và ngưỡng.

Đầu ra: Là tín hiệu đầu ra của một nơ-ron, với mỗi nơ-ron sẽ có tối đa là một đầu ra.

Xét về mặt toán học, cấu trúc của một nơ-ron k, được mô tả bằng biểu thức như sau: và

Trong đó: : là các tín hiệu vào; ( là các trọng số liên kết của nơ-ron thứ k; là hàm tổng; là hàm tổng; f là hàm truyền và là tín hiệu đầu ra của nơ-ron

Như vậy nơ-ron nhân tạo nhận các tín hiệu đầu vào, xử lý (nhân các tín hiệu này với trọng số liên kết, tính tổng các tích thu được rồi gửi kết quả tới hàm truyền), và cho một tín hiệu đầu ra (là kết quả của hàm truyền).

### Một số kiểu mạng nơ-ron

Cách thức kết nối các nơ-ron trong mạng xác định kiến trúc (topology) của mạng. Các nơ-ron trong mạng có thể kết nối đầy đủ (fully connected) tức là mỗi nơ-ron khác, hoặc kết nối cục bộ (partially connected) chẳng hạn chỉ kết nối giữa các nơ-ron trong các tầng khác nhau. Có 2 loại kiến trúc chính:

Tự kết hợp(autoassociative): là mạng có các nơ-ron đầu vào cũng là các nơ-ron đầu ra.

Kết hợp khác kiểu (heteroassociative): là mạng có tập nơ-ron đầu vào và đầu ra riêng biệt. Perceptron, các mạng Perceptron nhiều tầng (MLP: MultiLayer Percepton), …

Ngoài ra tùy thuộc vào mạng có các kết nối có các kết nối ngược (feedback connection) từ các nơ-ron đầu ra tới các nơ-ron đầu vào hay không, Mạng nơ-ron được chia làm 2 loại

Kiến trúc truyền thẳng (feedforward architechture): là kiến trúc mạng không có các kết nối ngược trở lại từ các nơ-ron đầu ra veef các nơ-ron đầu vào; mạng không lưu lại các giá trị output trước và các trạng thái kích hoạt của nơ-ron. Các mạng truyền thẳng cho phép tín hiệu di chuyển theo đường duy nhất; từ đầu vào tới đầu ra, đầu ra của một tầng bất kì sẽ không ảnh hưởng tới tầng đó. Các mạng kiểu Perceptron là mạng truyền thẳng.

Kiến trúc phản hồi (Feedback architecture): là kiểu kiến trúc mạng có các kết nối từ nơ-ron đầu ra tới nơ-ron đầu vào. Mạng lưu lại các trạng thái trước đó, và trạng thái tiếp theo không chỉ phụ thuộc vào các tín hiệu đầu vào mà còn phụ thuộc vào các trạng thái trước đó của mạng. Mạng Hopfield thuộc loại này.

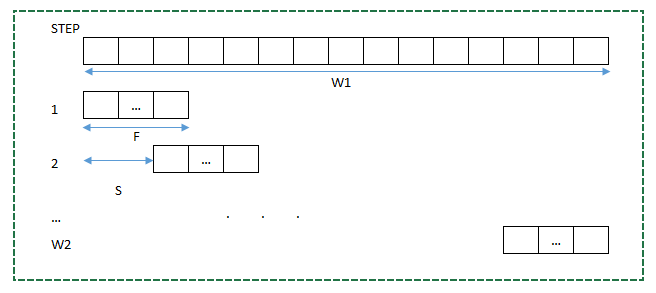
### Giới thiệu mạng nơ-ron tích chập

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. [4]

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.



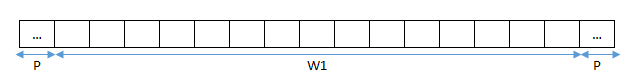
Hình 1.5 Hoạt động tích chập của CNN

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 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 :



Hình 1.6 Thêm vào đường viền để chia hết

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

### Cấu tạo mạng nơ-ron tích chập (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.

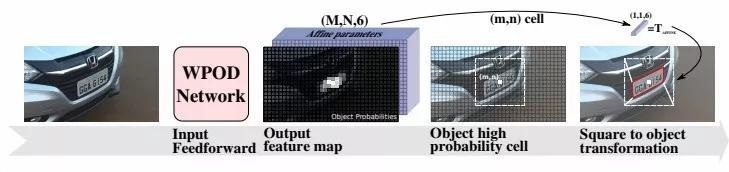
Quá trình tổng hợp (max pooling): Các tầng càng về sau khi trích xuất đặc trưng sẽ cần số lượng tham số lớn do chiều sâu được qui định bởi số lượng các kênh ở các tầng sau thường tăng tiến theo cấp số nhân. Điều đó làm tăng số lượng tham số và khối lượng tính toán trong mạng nơ-ron. Do đó để giảm tải 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.

## Tìm hiểu mạng WPOD

### Giới thiệu mạng WPOD

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. [5]

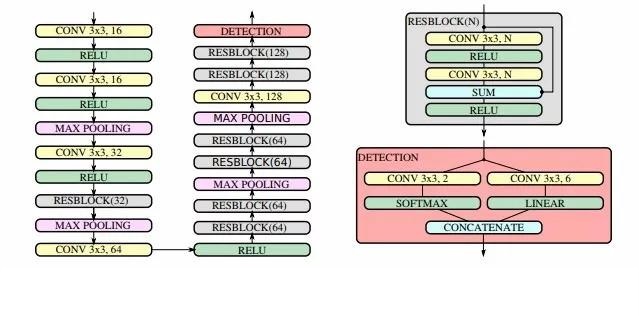


Hình 1.7 Hoạt động của mạng WPOD

Hình bên trên thể hiện cách thức hoạt động của mạng WPOD. Ta có thể thấy từ 1 ảnh đầu vào thông qua quá trình lan truyền tiến qua mạng ta thu được output features map gồm 8 channels trong đó 2 channels đầu tiên là xác suất có/không có biển số xe và 6 channels còn lại là những thông số để tính toán ma trận transform (là ma trận được sử dụng để chuyển đổi góc nhìn của biển số xe).

Để trích xuất ra được vùng biển số xe thì nhóm tác giả đã xét một hình vuông với kích thước cố định (phần hình vuông viền trắng trên hình) xung quanh từng ô trên output features map. Nếu xác suất có đối tượng của ô đó lớn hơn ngưỡng quy định thì những giá trị của 6 channels còn lại của ô đó sẽ được sử dụng để tính toán ma trận tranform từ vùng hình vuông về vùng biển số xe (đa giác viền đỏ trên hình). Và ta cũng sẽ sử dụng ma trận này để đưa biển số xe về hướng nhìn chính diện.

### Kiến trúc Mạng WPOD-NET

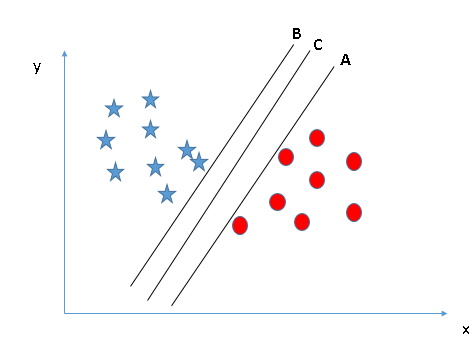


Hình 1.8 Kiến trúc mạng WPOD

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 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). [5]

## 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 độ dài lớn nhất của lề sẽ có một mặt phẳng tạo ranh giới giữa các loại. [6]



Hình 1.9 Hoạt động thuật toán SVM

## Các công cụ xây dựng đồ án

### Ngôn ngữ lập trình Python

1. **Giới thiệu về ngôn ngữ lập trình 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. [7]

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.

Với Python, việc phát triển ứng dụng và debug trở nên nhanh hơn bởi vì không cần đến bước biên dịch và chu trình edit-test-debug của Python là rất nhanh.

1. **Đặc điểm của Python**

Dưới đây là một số đặc điểm chính của Python:

* Dễ dàng để sử dụng: Python là một ngôn ngữ bậc cao rất dễ dàng để sử dụng. Python có một số lượng từ khóa ít hơn, cấu trúc của Python đơn giản hơn và cú pháp của Python được định nghĩa khá rõ ràng, … Tất cả các điều này là Python thực sự trở thành một ngôn ngữ thân thiện với lập trình viên.
* Bạn có thể đọc code của Python khá dễ dàng. Phần code của Python được định nghĩa khá rõ ràng và rành mạch.
* Python có một thư viện chuẩn khá rộng lớn. Thư viện này dễ dàng tương thích và tích hợp với UNIX, Windows, và Mac.
* Python là một ngôn ngữ thông dịch. Trình thông dịch thực thi code theo từng dòng, điều này giúp cho quá trình debug trở nên dễ dàng hơn và đây cũng là yếu tố khá quan trọng giúp Python thu hút được nhiều người học và trở nên khá phổ biến.
* Python cũng là một ngôn ngữ lập trình hướng đối tượng. Ngoài ra, Python còn hỗ trợ các phương thức lập trình theo hàm và theo cấu trúc.
* Ngoài các đặc điểm trên, Python còn khá nhiều đặc điểm khác như hỗ trợ lập trình GUI, mã nguồn mở, có thể tích hợp với các ngôn ngữ lập trình khác, …

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. [8]

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. [9]

1. **Hệ quản trị cơ sở dữ liệu**

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. [10]

SQLite là một công cụ cơ sở dữ liệu SQL được nhúng. Không giống như hầu hết các cơ sở dữ liệu SQL khác, SQLite không có một quy trình máy chủ riêng biệt. SQLite đọc và ghi trực tiếp vào các tệp đĩa thông thường. Một cơ sở dữ liệu SQL hoàn chỉnh với nhiều bảng, chỉ mục, trình kích hoạt và dạng xem, được chứa trong một tệp đĩa duy nhất. Định dạng tệp cơ sở dữ liệu là đa nền tảng - bạn có thể thoải mái sao chép cơ sở dữ liệu giữa các hệ thống 32bit và 64bit hoặc giữa các kiến ​​trúc big-endian và little-endian. Các tính năng này làm cho SQLite trở thành một lựa chọn phổ biến làm Định dạng Tệp Ứng dụng. Tệp cơ sở dữ liệu SQLite là định dạng lưu trữ được đề xuất bởi Thư viện Quốc hội Hoa Kỳ. Hãy nghĩ về SQLite không phải là một sự thay thế cho Oracle mà là một sự thay thế cho fopen().

SQLite là một thư viện nhỏ gọn. Với tất cả các tính năng được bật, kích thước thư viện có thể nhỏ hơn 600KiB, tùy thuộc vào nền tảng mục tiêu và cài đặt tối ưu hóa trình biên dịch. Có sự cân bằng giữa việc sử dụng bộ nhớ và tốc độ. SQLite thường chạy nhanh hơn khi bạn cung cấp nhiều bộ nhớ hơn. Tuy nhiên, hiệu suất thường khá tốt ngay cả trong môi trường bộ nhớ thấp. Tùy thuộc vào cách nó được sử dụng, SQLite có thể nhanh hơn I / O hệ thống tệp trực tiếp.

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. [11]

Gói tkinter (“giao diện Tk”) là giao diện Python tiêu chuẩn cho bộ công cụ Tk GUI. Cả Tk và tkinter đều có sẵn trên hầu hết các nền tảng Unix, cũng như trên các hệ thống Windows. (Bản thân Tk không phải là một phần của Python; nó được duy trì tại ActiveState).

Tcl (Tool Command Language) là một ngôn ngữ lập trình động rất mạnh mẽ nhưng dễ học, phù hợp với nhiều mục đích sử dụng, bao gồm các ứng dụng web và máy tính để bàn, mạng, quản trị, thử nghiệm và nhiều hơn nữa. Mã nguồn mở và thân thiện với doanh nghiệp, Tcl là một ngôn ngữ đã trưởng thành nhưng đang phát triển, thực sự là một nền tảng đa nền tảng, dễ dàng triển khai và có khả năng mở rộng cao.

Tk là một bộ công cụ giao diện người dùng đồ họa đưa việc phát triển các ứng dụng máy tính để bàn lên một cấp độ cao hơn so với các cách tiếp cận thông thường. Tk là GUI tiêu chuẩn không chỉ cho Tcl mà còn cho nhiều ngôn ngữ động khác và có thể tạo ra các ứng dụng gốc, phong phú chạy không thay đổi trên Windows, Mac OS X, Linux.

1. **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. [12]

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 DẠNG 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 theo 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

* Biển nền vàng chữ đen là loại biển dành cho những xe thuộc công ty vận tải hay kinh doanh dịch vụ vận tải.

## 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 dạng 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 diện.

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 dạng 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 dạng. 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

Khi một bức ảnh bình thường được đưa vào xử lý bức ảnh đó là bức ảnh có màu và điều này là không cần thiết nên ta sẽ đưa ảnh màu về ảnh xám để giảm những thông tin không cần chú ý đến.

Đầu tiên, sử dụng phương thức cv2.cvtColor và tham số truyền là cv2.COLOR\_BGR2GRAY để chuyển từ ảnh màu sang ảnh xám

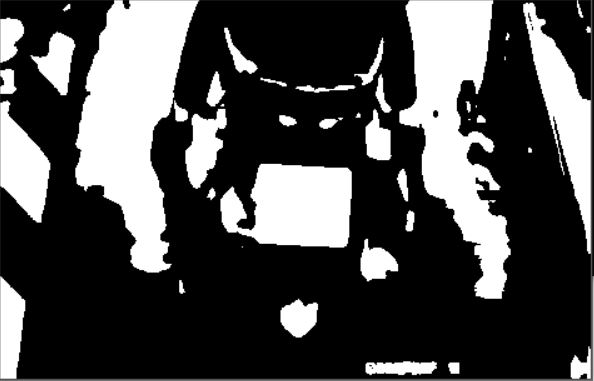
Bước tiếp theo cần, ảnh cần được làm mịn để giữ lại những đường biên của ảnh ta cần dùng phương thức cv2.bilateralFilter và sử dụng cv2.equalizeHist để cân bằng sáng cho hình biển số xe.



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. Phân ngưỡng là hoạt động nếu một pixel trong ảnh xám có giá trị lớn hơn ngưỡng thì sẽ được gán giá trị 1 và ngược lại.



Hình 2.7 Hình biển số xe sau khi nhị phân

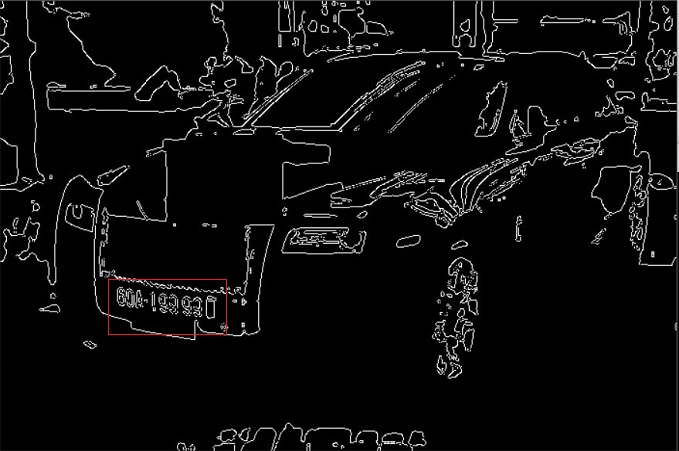
Tiếp đến cần vẽ ra những đương bao xung quanh ảnh đã được nhị phân hóa vừa rồi. Ta sẽ dùng hàm cv2.Canny. Nếu các đường thẳng chưa rõ nét ta dùng phương thức cv2.dilate để làm nở rộng những đường bao vừa tìm được.



Hình 2.8 Đường bao của hình biển số xe

Từ ảnh đường bao trên ta sẽ tìm contour cho ảnh. Contour là đường bao quanh được tạo bởi các điểm liên tục có cùng màu hoặc cùng cường độ. Và sử dụng cv2.approxPolyDP tìm ra contour nào có 4 cạnh thì có thể 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. Từ đó không cắt được biển số xe.



Hình 2.9 Biển 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 có độ tương phản cao 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 dạng biển số xe.

### Huấn luyện mạng WPOD

Đánh nhãn biển số xe gồm 10 thông số. Đó là tọa độ các góc của biển số xe từ trái qua phải theo chiều kim đồng hồ. Thông số thể thể hiện trục hoành(x) chia tỷ lệ cho chiều dài của bức ảnh. Thông số thể hiện trục tung(y) chia tỷ lệ theo chiều cao của bức ảnh.

Tạo 1 file txt cùng tên với file ảnh muốn huấn luyện và ghi lại các thông số: N, tlx, trx, brx, blx, tly, try, bry, bly, LABEL.

* Với N là số góc của biển số xe mặc định là 4.
* Tlx = x góc trái trên /chiều rộng của ảnh.
* Trx = x góc phải trên /chiều rộng của ảnh.
* Brx = x góc phải dưới /chiều rộng của ảnh.
* Blx = x góc trái dưới /chiều rộng của ảnh.
* Tly = y góc trái trên /chiều cao của ảnh.
* Try = y góc phải trên /chiều cao của ảnh.
* Bry = y góc dưới phải /chiều cao của ảnh.
* Bly = y góc dưới trái /chiều cao của ảnh.
* Label là tên đối tượng muốn huấn luyện, ở đây là biển số xe (LP).
* Ví dụ về cách gán nhãn của WPOD



Độ lớn của ảnh là: 4032 x 2268 pixel

Tọa độ 4 góc của biển số xe: 1657 x 1281, 2477 x 1213, 2541 x 1797, 1729 x 1861

tlx = 1657/4032 = 0.410962, trx = 2477/4032 = 0.614335, brx = 2541/4032 = 0.630208, blx=1729/4032=0.428819,

tly = 1281/2268= 0.564815, try = 1213/2268 = 0.534832, bry = 1797/2268= 0.792328, bry = 1861/2268= 0.820547

Thông tin ghi trong file txt được tạo cùng tên với tên file ảnh là 4, 0.410962, 0.614335, 0.630208, 0.428819, 0.564815, 0.534832, 0.792328, 0.820547, LP,

Sử dụng lệnh git clone: <https://github.com/sergiomsilva/alpr-unconstrained.git> để tải tool huấn luyện.

Vào thư mục vừa tải về được sử dụng lệnh: mkdir models để tạo folder models.

Tiếp tục sử dụng lệnh: python create-model.py eccv models/eccv-model-scracth để tạo ra một model trước.

Cuối cùng sử dụng lệnh: $ python train-detector.py --model models/eccv-model-scracth --name my-trained-model --train-dir samples/train-detector --output-dir models/my-trained-model/ -op Adam -lr .001 -its 300000 -bs 64

Những tham số cần lưu ý tham số --model là đường dẫn file model đã tạo ở bước trên, --name là tên model sẽ được tạo ra sau khi huấn luyện, --train-dir đường dẫn tới nơi chứa file huấn luyện, -its là số vòng muốn huấn luyện -bs là số ảnh được lấy trong lần huấn luyện số lượng được lấy sẽ phụ thuộc vào dung lượng của RAM. RAM càng có dung lượng lớn số bức ảnh được huấn luyện càng nhiều.

Với số lượng ảnh đã được gán nhãn trong tập dữ liệu đào tạo, sử dụng tăng dữ liệu rất quan trọng. Các biến đổi nâng cao sau đây sẽ được sử dụng

* Chỉnh lưu: Toàn bộ hình ảnh được chỉnh sửa dựa trên chú thích LP, giả sử rằng LP nằm trên một mặt phẳng.
* Tỷ lệ khung hình: Tỷ lệ khung hình LP được đặt ngẫu nhiên trong khoảng [2, 4] đến phù hợp với kích thước từ các khu vực khác nhau.
* Định tâm: Trung tâm LP trở thành trung tâm hình ảnh.
* Chia tỷ lệ: LP được chia tỷ lệ để chiều rộng của nó khớp với giá trị từ 40px đến 208px.
* Xoay: Xoay 3D với các góc được chọn ngẫu nhiên được thực hiện, để giải thích cho nhiều loại thiết lập máy ảnh.
* Chuyển đổi: Dịch ngẫu nhiên để di chuyển LP từ trung tâm của hình ảnh, giới hạn trong một hình vuông có kích thước 208 × 208pixel xung quanh tâm.
* Cắt xén: Xem xét trung tâm LP trước khi dịch, chúng tôi cắt xén 208 ×208 khu vực xung quanh nó.
* Không gian màu: Sửa đổi nhỏ trong không gian màu HSV.
* Chú thích: Vị trí của bốn góc LP được điều chỉnh bằng cách áp dụng các phép biến đổi không gian tương tự được sử dụng để tăng cường hình ảnh đầu vào.

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 max 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 hàm mất mát 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:

Ví dụ tính hàm loss: Sau khi hoàn thành 1 vòng huấn luyện mạng sẽ đưa ra 1 output feature map 0.9987205, 0.00127955, 0.66043425, 0.00179346, -0.01054392, 0.12250977, 0.15289468, 0.06239167

Phần đầu tiên cần tính đó là xây dựng phép biến đổi affine tại ô (15,8):

Với q1 = [-0.5, -0.5]

Với q2 = [0.5, -0.5]

Với q3 = [0.5, 0.5]

Với q4 = [-0.5, 0.5]

Phần tiếp theo là tính 4 điểm góc thực tế trên ảnh input về các toạ độ tương ứng trên output features map tại ô [15,8]

Với p1 = [114,291]

Với p2 = [214,309]

Với p3 = [114,291]

Với p4 = [114,314]

Tính faffine (15,8):

faffine (15,8) = 0.538+0.756+0.86+0.357 = 2.511

Xử lý xác xuất có/không của một đối tượng tại ô (15, 8)

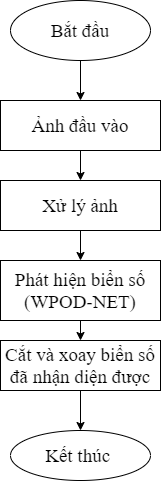
fprobs (15,8)=

=

Loss của ô (15,8) là:

loss (15,8) = faffine (15,8) + fprobs (15,8) = 2.514

### 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 dạng bằng opencv thì nhận dạng 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 dạng 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 dạng 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.

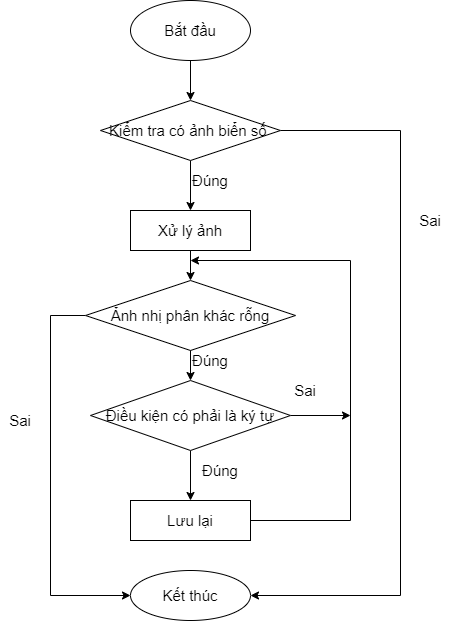


Hình 2.12 Minh họa cho phương thức connected-compoment

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.

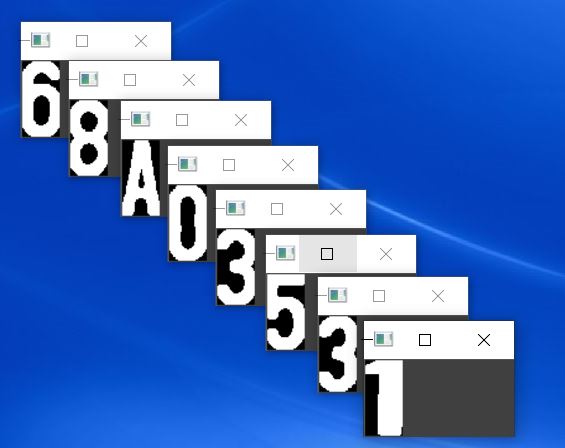
Từ ảnh nhị phân ta thấy rằng số lượng pixel cho mỗi ký tự rơi vào một phạm vi nhất định. Tử lệ cạnh chiều dài/ rộng <= 8.6 và >0.4 và tỉ lệ điểm trắng > 0.35 và <0.85.

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 2.13 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 2.14 Kết quả sau khi thực hiện tách ký tự

## Nhận dạng ký tự

### SVM tuyến tính (linear)

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.

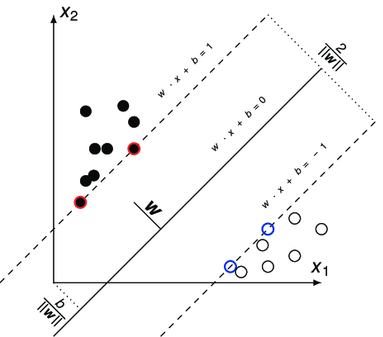
Mục tiêu là tìm một mặt phẳng:

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.



Hình 2.15 Phương pháp cực đại hóa lề (margin)

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: , sao cho .

Giải bằng phương pháp nhân tử Lagrange , , …, 0 đối với mỗi dữ liệu huấn luyện ta được:

Chúng ta cần chyển cực đại dựa vào với các ràng buộc gradient sẽ triệt tiêu biến w và b:

Thế vào L(w,b, α ) ta được công thức Lagrange đối ngẫu LD trong đó không còn các biến w và b :

Khi tìm được các chúng ta sẽ tìm lại được w nhờ công thức:

Và b sẽ là trung bình của y-wx (với các x nằm trong tập mẫu). Sau khi tìm w và b ta xây dựng hàm quyết định

### SVM tuyến tính lề mềm (Soft margin) và phi tuyến tính (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:

### Hitogram Intersection 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: [13]

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | Bin 1 | Bin 2 | Bin 3 | Bin 4 | Bin 5 | Bin 6 | Bin 7 | Bin 8 |
| H1 | 50 | 30 | 32 | 45 | 38 | 20 | 85 | 100 |
| H2 | 100 | 75 | 30 | 15 | 75 | 50 | 26 | 29 |
| Min (h1, h2) | 50 | 30 | 30 | 15 | 38 | 20 | 26 | 29 |

Ví dụ cách tính Histogram Intersection

Số pixel của 1 bức ảnh là: 400 pixels.

Histogram intersection = = 0.405

Ưu điểm

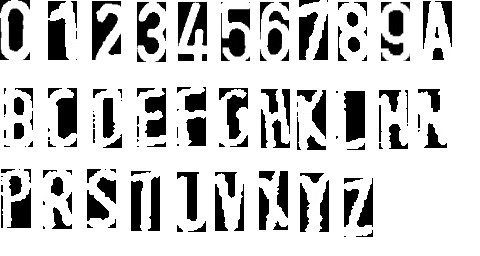
* Tốc độ nhanh.
* Đơn giản, dễ cài đặt.

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 2.16 Các kí tự để huấn luyện thuật toán SVM

# PHẦN MỀM THỬ NGHIỆM

## Yêu cầu bài toán

### Yêu cầu chức năng

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. Khi vào chỉnh sửa giá hay muốn biết lịch sử ra vào của một xe trong bãi chỉ có người được giao trách nhiệm mới có thể thực hiện những công việc đó.

### Yêu cầu phi chức năng

Giao diện phải được thiết kế đơn giản, dễ hiểu, dễ sử dụng, thể hiện đầy đủ thông tin, cũng như tính nhất quán của giao diện.

Dễ dàng nâng cấp phần mềm, phần mềm phải ít lỗi.

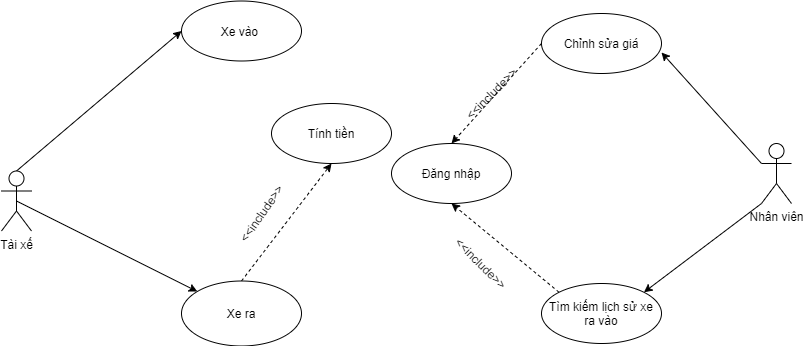
Hoạt động tốt trên mọi hệ điều hành chạy trên máy tính phổ biến hiện nay đặc biệt là máy tính chạy hệ điều hành Window.

Tốc độ xử lý không cần quá nhanh nhưng cũng không được quá chậm tốc độ đề nghị là 2 đến 3 giây trong mỗi lần nhận dạng.

Chú ý tới yêu cầu dễ nâng cấp của phần mềm và tốc độ xử lý của phần mềm.

## Chức năng hệ thống

### Sơ đồ use case



Hình 3.1 Sơ đồ Use Case hệ thống

Sơ đồ Use Case biểu diễn chức năng hệ thống. Từ các yêu cầu của hệ thống, sơ đồ Use Case 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.

Chức năng xe vào: Chức năng sẽ nhận dạng khi xe ở cửa vào của bãi đỗ xe và tiến hành ghi nhận vào cơ sở dữ liệu.

Chức năng xe ra: Chức năng sẽ nhận biển số xe khi xe ở cổng ra của bãi đỗ xe và tiến hành tính chi phí đỗ xe và ghi nhận vào cơ sở dữ liệu.

Chức năng tính tiền: Sẽ hoạt đồng thời cùng với chức năng xe 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.

Chức năng điều chỉnh phí đỗ xe: Giá sẽ thay đổi theo thời gian khi cần thay đổi phí đỗ xe chức năng này sẽ sửa phí đỗ xe đang hiện hành.

Chức năng xem lịch sử xe: Chức năng cho phép người quản lý có thể lịch sử vào ra bãi đỗ xe của một chiếc xe.

Chức năng đăng nhập: Nếu muốn sử dụng một trong hai chức năng cần qua chức năng đăng nhập.

### Phân tích module phần mềm

Module nhận dạng: Mỗi khi có ảnh chứa biển số xe thì module này được sử dụng cần một mạng đã được huấn luyện để có thể nhận dạng được biển số xe, cần một file đã được huần luyện về nhận dạng ký tự. Kết quả của module nhận dạng sẽ là một chuỗi ký tự của biển số xe.

Module lưu vào bảng xe vào: Module này sẽ được thực thi khi module nhận dạng trả ra một chuỗi ký tự. Công việc của module này là nhận chuỗi ký tự được truyền vào, cùng với tạo ghi nhận lại thời gian hiện tại. Cuối cùng là thực thi câu lệnh sql để lưu chuỗi và thời gian được ghi nhận vào cơ sở dữ liệu.

Module lưu vào bảng xe ra: Module này cũng sẽ nhận một chuỗi từ module nhận dạng và thực thì câu lệnh sql để lấy thông tin của chuỗi trong bảng xe vào. Xóa thông tin của chuỗi đó trong bảng xe vào ghi nhận lại thời gian hiện tại và lưu vào bảng xe ra. Đồng thời module này sẽ gọi luôn module tính phí.

Module tính phí: Sẽ nhận được thời gian vào và thời gian ra của xe, thực thi câu lệnh để lấy phí đỗ xe của một giờ. Từ đó tính toán thời gian xe ở lại trong bãi đỗ xe và tính tiền phí. Trả ra sẽ là một chuỗi số là phí đỗ xe và hiển thị lên cho người dùng.

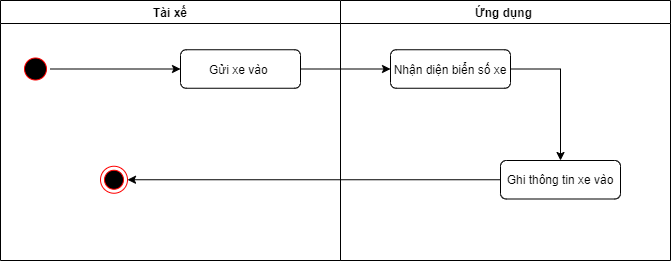
Module đăng nhập: Module này sẽ nhận từ giao diện hai chuỗi. Một chuỗi là tên đăng nhập, một là mật khẩu. Nếu đúng thì sẽ cho vào giao diện quản lý.

Module thay đổi phí: Module này sẽ nhận một chuỗi là giá tiền cần thay đổi từ giao diện và thực thi một câu lệnh để thêm giá mới vào cơ sở dữ liệu.

Module tìm kiếm lịch sử ra vào: Module là này sẽ nhận từ giao diện một chuỗi là biển số của một xe và tìm kiếm trong bảng xe ra thời gian ra vào và số tiền phải trả của xe đó.

## Sơ đồ hoạt động

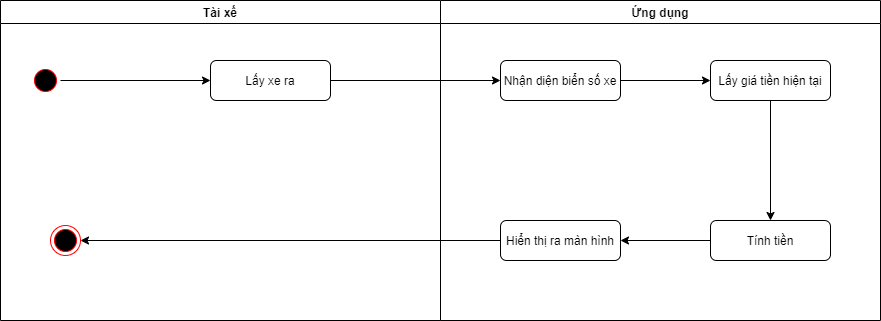
Sơ đồ hoạt động xe vào



Hình 3.2 Sơ đồ hoạt động khi xe vào

Khi đưa xe vào ảnh có chứa biển số xe sẽ được đưa vào chức năng nhận dạng biển số sẽ đưa ảnh chứa biển số xe ra thành văn bản. Sau đó văn bản chứa các ký tự của biển số xe sẽ được lưu vào trong cơ sở dữ liệu ở bảng xe vào. Cùng lúc đó là ghi những thông tin liên qua như ngày giờ xe vào.

Sơ đồ hoạt động xe ra



Hình 3.3 Sơ đồ hoạt động khi xe ra

Khi lấy xe ra, ảnh chứa biển số xe cũng được đưa vào nhận dạng đưa thành văn bản, cùng lúc đó sẽ lấy trong giá tiền đã được niêm yết trước đó và ngày giờ chiếc xe ấy đi vào từ đó có thể tính ra được số tiếng chiếc xe ở trong bãi và tính tiền.

## Thiết kế cơ sở dữ liệu

### Tìm hiểu về SQLite3



Hình 3.4 Hệ quản trị cơ sở dữ liệu

Hệ quản trị cơ sở dữ liệu được sử dụng ở đây là SQLite3. SQLite3 truy vấn dữ liệu nhanh hơn cả khi đọc ghi dữ liệu ra file của Python, được Python hỗ trợ. SQLite 3 cũng có những ưu nhược điểm cần lưu ý tới:

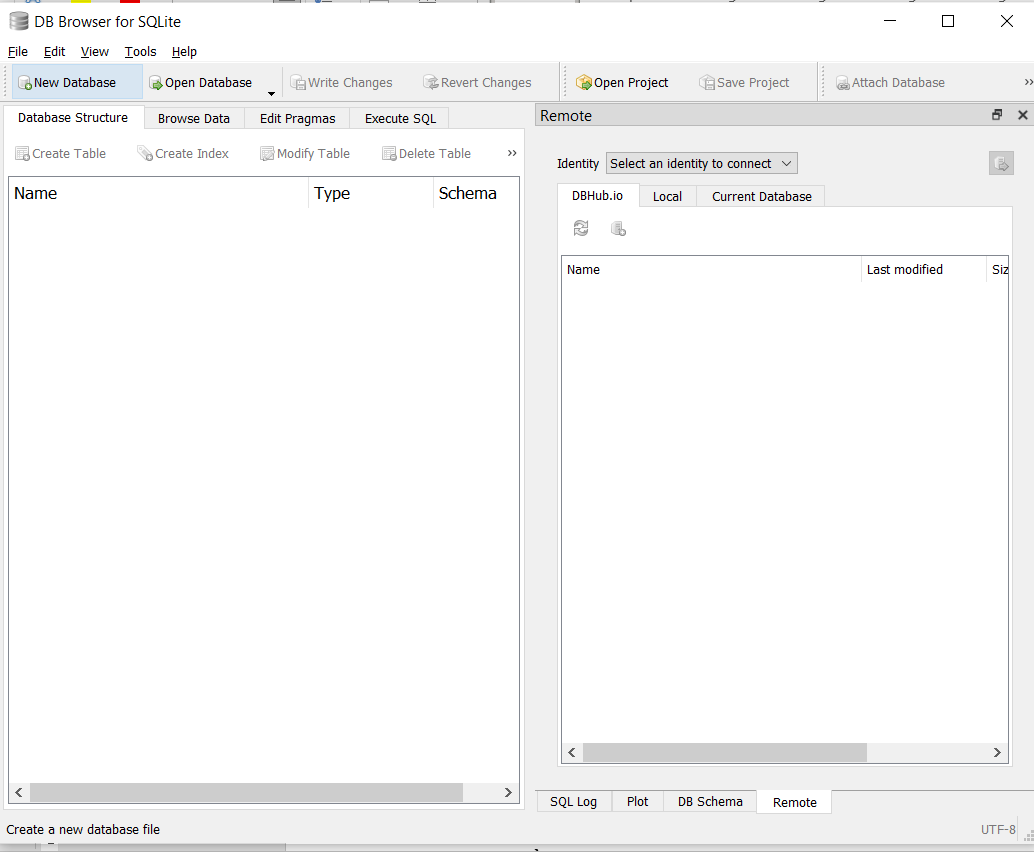
Ưu điểm:

* + SQLite 3 là một hệ quản trị cơ sở dữ liệu nhỏ gọn.
  + SQLite 3 được nhiều ngườ sử dụng xây dựng ứng dụng. Đặc biệt là ứng dụng điện thoại.

Nhược điểm:

* SQLite3 chỉ thực hiện được một yêu cầu và phải thời gian đợi để thực hiện lệnh tiếp theo. SQLite3 không thể xử lý nhiều lệnh đồng thời như MySQL, SQLServer, … Đó chính là nhược điểm mà nếu muốn xây dựng thành ứng dụng web sẽ không sử dụng được SQLite3.

File lưu cơ sở dữ liệu của SQLite3 là đuôi .db nên vì vậy cần có một phần mềm để đọc và để dàng dàng cho người sử dụng. Phần mềm được sử dụng ở đây là DB Browser for SQLite.



Hình 3.5 Giao diện phần mềm DB Broswer for SQLite

Trước khi muốn thao tác với cơ sở dữ liệu ta cần tạp ra nó. Có thể bấm vào New Databasae xong sau đó phần mềm sẽ hỏi vị trí lưu cơ sở dữ liệu chỉ cần chọn vị trí muốn lưu, ghi tên cơ sở dữ liệu, chọn Ok.

Trước khi muốn thao tác với cơ sở dữ liệu, cơ sở dữ liệu cần được mở lên. Để có thể mở cơ sở dữ liệu có thể chọn vào Open Database và chọn cơ sở dữ liệu muốn mở hoặc có thể vào File -> Open Database và chọn cơ sở dữ liệu cần mở.

Sau khi mở được cơ sở dữ liệu lên ta có thể thao tác trực tiếp bằng câu lệnh sql trong thẻ Excute SQL.

Để có thể xem dữ liệu ta chọn Browse Data và chọn bảng cần xem.

### Các bảng phục vụ phần mềm

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.

Bảng tài khoản: Đây sẽ là nơi chứa thông tin đăng nhập như tên người sử dụng, mật mã đăng nhập, trong tương lai phần mềm được phát triển hơn nữa sẽ có sự phân quyền khi làm việc. Từ bảng này cũng có thể biết được ai đã truy cập vào hệ thống đã làm những gì đã thay đổi những gì.

## Thiết kế giao diện phần mềm

### Các thành phần cơ bản trong Tkinter

Tkinter cung cấp nhiều phần tử để xây dựng GUI phổ biến có thể kể đến như button (Nút), entry (trường nhập dữ liêu), label (nhãn hiển thị chứ thích) và còn nhiều thành phần như thế có thể tham khảo ở bảng dưới. Tất cả những thành phần đó được gọi là **Widgets.**

**Thông thường, Widget là những phần tử để xây dựng GUI và thể hiển thông hoặc cho người dùng tương tác với hệ điều hành. Trong Tkiner, Widget là những đối tượng; thể hiện của các lớp đại diện cho** Button (nút), Entry (trường nhập dữ liêu), Label (nhãn hiển thị chứ thích).

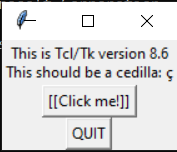
Mỗi Widget riêng biệt là một đối tượng Python. Khi tạo một Widget phải truyền cha của nó làm tham số. Root là một trường hợp ngoại lệ đó là cửa sổ cao cấp nhất chứa mọi thứ trong nó và khi tạo Root không cần cha.

Dưới đây là những widget mà Tkinter hỗ trợ.

|  |  |
| --- | --- |
| STT | Mô tả |
| 1 | Button: Tiện ích Button được sử dụng để hiển thị các nút trong ứng dụng |
| 2 | Canvas: Sử dụng để vẽ các hình dạng, chẳng hạn như đường thẳng, hình bầu dục, đa giác và hình chữ nhật, trong ứng dụng của bạn. |
| 3 | Checkbutton: Sử dụng để hiển thị một số tùy chọn dưới dạng hộp kiểm. Người dùng có thể chọn nhiều tùy chọn cùng một lúc. |
| 4 | Entry: Được sử dụng để hiển thị trường văn bản một dòng để chấp nhận các giá trị từ người dùng. |
| 5 | Frame: Được sử dụng như một widget vùng chứa để sắp xếp các widget khác. |
| 6 | Label: Được sử dụng để cung cấp chú thích một dòng cho các tiện ích con khác. Nó cũng có thể chứa hình ảnh. |
| 7 | Listbox: Được sử dụng để cung cấp danh sách các tùy chọn cho người dùng. |
| 8 | Menubutton: được sử dụng để hiển thị các menu trong ứng dụng của bạn. |
| 9 | Menu: Được sử dụng để cung cấp các lệnh khác nhau cho người dùng. Các lệnh này được chứa bên trong Menubutton. |
| 10 | Radiobutton: Được sử dụng để hiển thị một số tùy chọn dưới dạng các nút radio. Người dùng chỉ có thể chọn một tùy chọn tại một thời điểm. |
| 11 | Scale:  Được sử dụng để cung cấp tiện ích con trượt. |
| 12 | Scrollbar: Được sử dụng để thêm khả năng cuộn vào các tiện ích con khác nhau, chẳng hạn như hộp danh sách. |
| 13 | Text:  Được sử dụng để hiển thị văn bản trong nhiều dòng. |
| 14 | Toplevel: Được sử dụng để cung cấp một vùng chứa cửa sổ riêng biệt. |
| 15 | Spinbox: Tiện ích Spinbox là một biến thể của tiện ích Tkinter Entry tiêu chuẩn, có thể được sử dụng để chọn từ một số giá trị cố định. |
| 16 | LabelFrame: là một tiện ích chứa đơn giản. Mục đích chính của nó là hoạt động như một bộ đệm hoặc vùng chứa cho các bố cục cửa sổ phức tạp |
| 17 | tkMessageBox: Mô-đun này được sử dụng để hiển thị các hộp thông báo trong các ứng dụng của bạn. |
| 18 | Message: Được sử dụng để hiển thị các trường văn bản nhiều dòng để chấp nhận các giá trị từ người dùng. |

Để lập trình GUI bằng Tkinter thành thạo các khối hình cũng là yếu tố cần thiết và quan trọng. Các Widget Tkinter đều có quyền truy cập vào các phương pháp quản lý hình học cụ thể, có mục đích tổ chức các widget trong toàn bộ khu vực widget chính. Tkinter cũng đưa ra các lớp với trình quản lý hình học như: pack, grid và place:

* Phương thức pack(): Sắp xếp các tiện ích trong khối trước khi đặt chúng vào các parent widgets
* Phương thức grid(): Sắp xếp các tiện ích trong một cấu trúc bảng tương tự như trong tiện ích parents.
* Phương thức place(): Sắp xếp các tiện ích bằng cách đặt chúng vào các vị trí cụ thể trong tiện ích parents.



Hình 3.6 Giao diện để kiểm tra Tkinker đã có trong máy

Để có thể biết được Tkinter đã được cài đặt trong máy chưa. Ta có thể sử dụng câu lệnh python -m tkinter trong cửa sổ dòng lệnh để kiểm nếu ra một xuất hiện một giao diện đơn giản như hình tức là đã được cài đặt.

Giao diện phần mềm được xây dựng bằng Tkinter một thư viện mặc định của Python. Thư viện Tkinter có những ưu điểm và nhược điểm:

* Ưu điểm

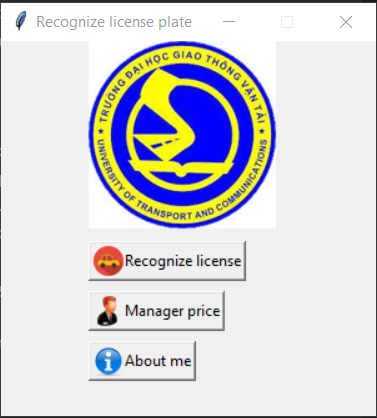
Thư viện Tkinter là một thư viện mặc định trong Python nên không cần cài đặt phức tạp.

Tkinter là một thư viện giúp xây dựng giao diện một cách đơn giản, trực quan nhất. Nhưng Tkinter vẫn có một công cụ cơ bản để xây dựng giao diện người dùng.

* Nhược điểm:

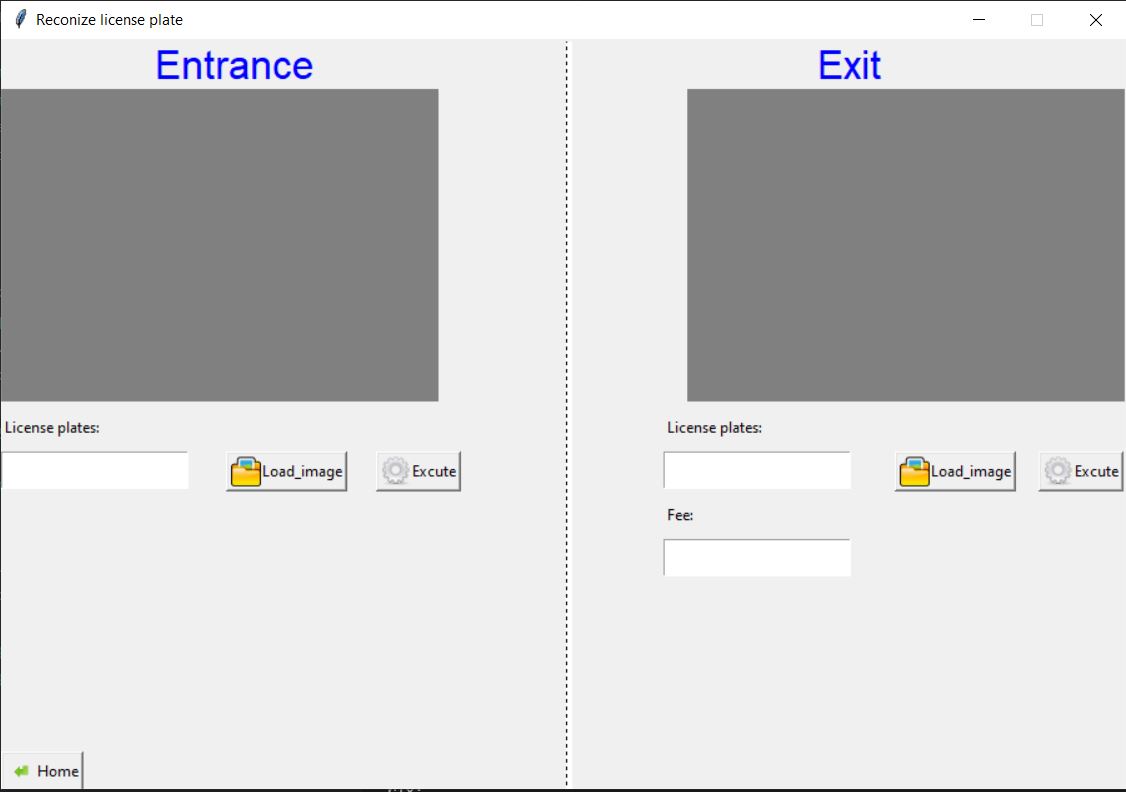
Do Tkinter là một thư viện đơn giản nên không phù hợp để xây dựng những ứng dụng có giao diện phức tạp.

### Giao diện phần mềm



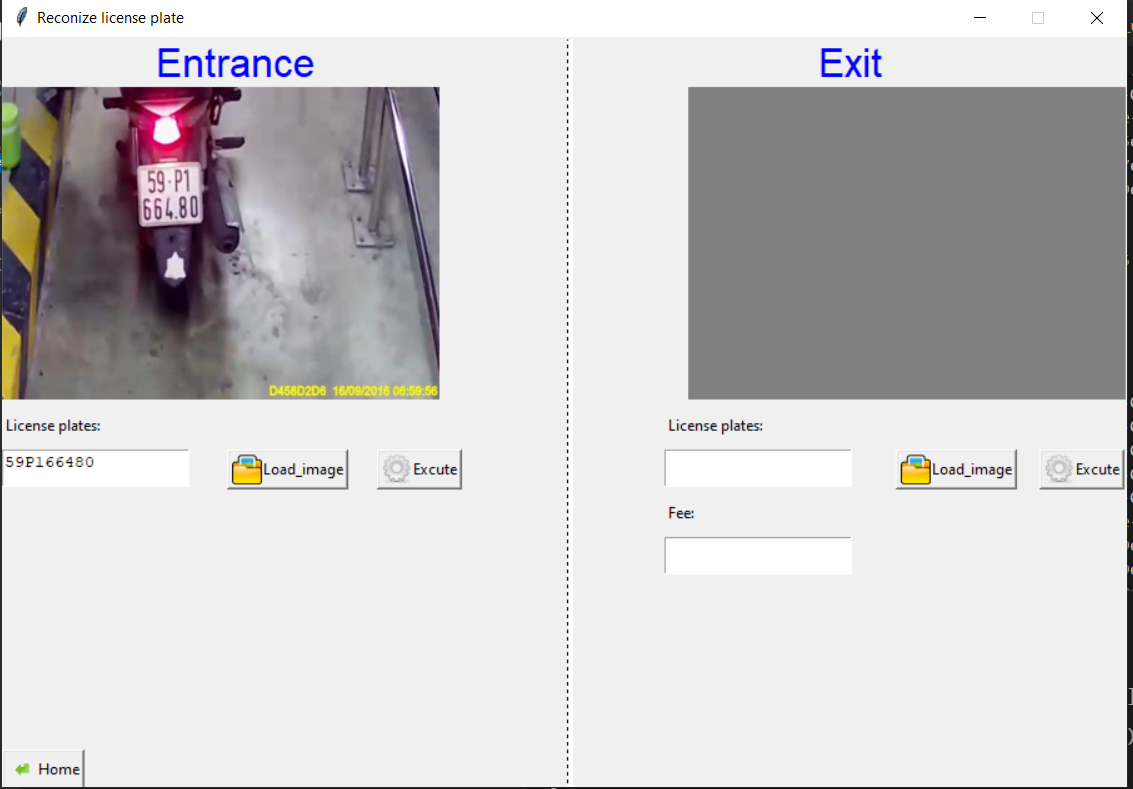
Hình 3.7 Giao diện khi bắt đầu khởi động chương trình

Giao diện cho phép người dùng có thể thấy rõ chức năng của phần mềm Reognize license là nhận dạng biển số xe. Manager price dùng để hiện giao diện cho người quản lý khi muốn điều chỉnh giá hay tìm kiếm lịch sử xe ra-vào.



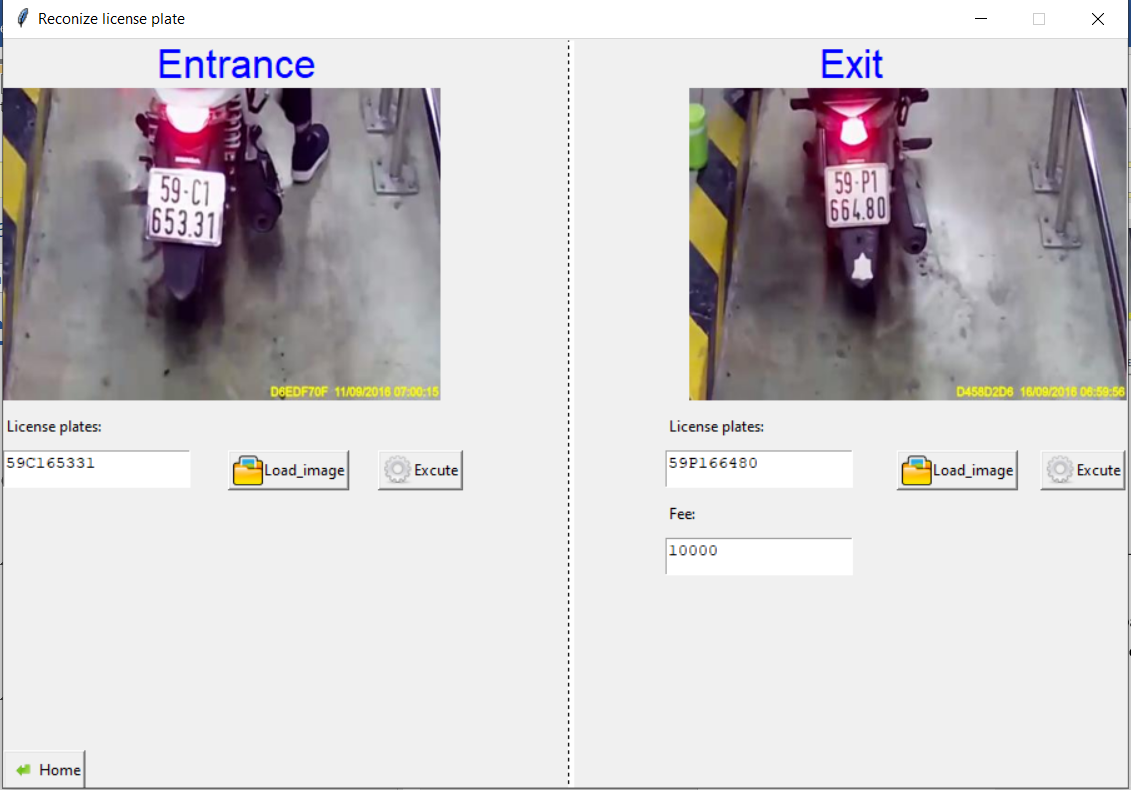
Hình 3.8 Giao diện nhận dạng biển số xe vào - ra

Giao diện được chia ra làm hai cổng Entrance chính làm cổng vào của bãi đỗ xe, Exit chính là cổng ra của bãi đỗ xe. Load image để tải ảnh lên và Excute là nút để tiến hành nhận biển số xe.



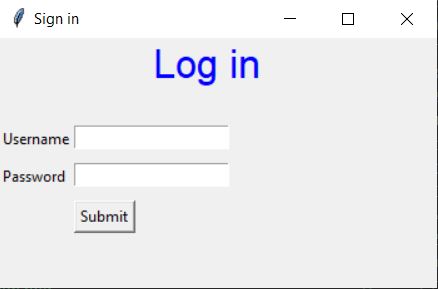
Hình 3.9 Đã nhận dạng được xe vào bãi đỗ xe

Hoạt động nhận dạng biển số xe đã hoạt động và nhận dạng thành công được biển số xe đó là 59P166480 khi xe vào bãi đỗ xe, sau khi nhận dạng thành công biển số xe sẽ được lưu vào cơ sở dữ liệu ở bảng xe vào và ghi nhận luôn cả ngày giờ xe vào bãi đỗ xe trong cơ sở dữ liệu.



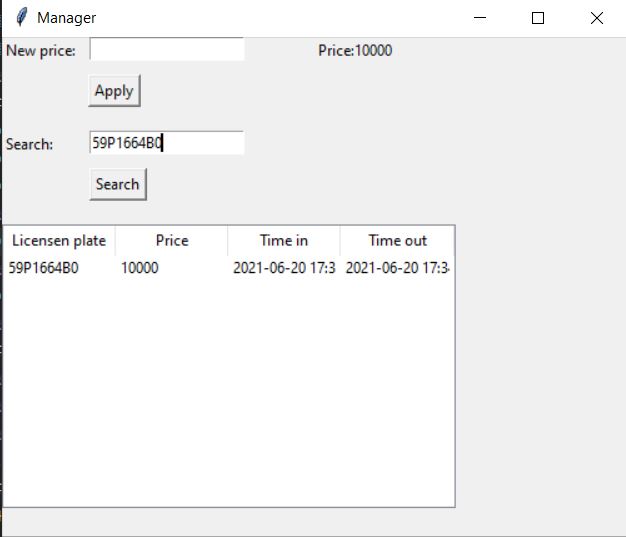
Hình 3.10 Xe ra thành công và được tính tiền đỗ xe

Xe mang biển kiểm soát 59P166480 đã ra khỏi bãi giữ xe hệ thống tiến hành nhận dạng biển số xe và tiến hành tính toán chi phí đỗ xe.



Hình 3.11 Màn hình đăng nhập khi muốn thay đổi giá

Để có thể vào chỉnh sửa giá cần đăng nhập để xác thực hệ thống sẽ cho phép vào trang quản lý để chỉnh sửa giá.



Hình 3.12 Màn hình cho phép điều chỉnh giá và lịch sử ra vào

Màn hình này cho phép người dùng có thể chỉnh sửa phí đậu xe sau khi chỉnh sửa phí sẽ được áp dụng ngay lập tức. Nên xe nào trước khi thay đổi giá nhưng ra sau khi thay đổi giá thì vẫn áp dụng phí mới cho xe đó. Ở đây, cũng được thiết kế để có thể tìm lịch sử xe ra vào trong bãi đỗ xe.

# KẾT LUẬN

## Kết quả đạt được

Về mặt học thuật, em đã bổ sung thêm kiến thức về machine learning cụ thể ở đây là bổ sung thêm kiến thức về thuật toán SVM, tìm hiểu được một phần của deep learning đó là mạng nơ-ron tích chập.

Em cũng đã bổ sung thêm một số kiến thức về những công cụ phục vụ cho machine learning như Tensorflow một nền tảng để xây dựng những ứng dụng về machine learning, cũng như là Keras một api hỗ trợ tối đa để xây dựng một mạng nơ-ron tích chập. Cuối cùng, em đã được biết thêm một công cụ quan trọng khác để huấn luyện mạng nơ-ron tích chập đó là google colab.

Trước đây trong quá trình học tập, em đểu được nghe tới những công cụ này nhưng chưa bao giờ có thể sử dụng nó và nhờ bài báo cáo này em đã hiểu thêm về những công cụ kia hơn.

Cũng từ bài báo cáo này, em đã cải thiện được kỹ năng làm báo cáo của chính mình qua những lần hỏi bạn, và những lần được hướng dẫn từ giảng viên hướng dẫn để có thể viết được một bài cáo cáo hoàn chỉnh.

Sau khi làm đồ án tốt nghiệp, em cũng đã xây dựng được một phần mềm nhận dạng biển số xe hoàn chỉnh tuy chỉ nằm ở mức thử nghiệm nhưng vẫn đáp ứng được những nhu cầu cơ bản của phần mềm đọc biển số xe quản lý xe ra - vào

## Hạn chế

Ở thực tế, có rất nhiều biển số xe có thể bị hắt sáng, bị dị dính trên mặt biển số xe nhưng những trường hợp đó phần mềm vẫn chưa thể phân biệt và nhận dạng ra được những ký tự bị hắt sang hay dị vật làm ảnh hưởng đến.

Phần mềm cũng bị giới hạn bởi kiến thức người làm nên chưa thể làm ra một phần mềm có tỷ lệ nhận dạng ra được ký tự có tỷ lệ đúng cao hơn.

Do ở phần nhận dạng chưa thể nhận dạng những ký tự bị ảnh hưởng từ ánh sáng, dị vật nên trong tập huấn luyện cũng không có những ký tự bị ảnh hưởng bởi ánh sáng và những dị vật.

Do chưa có mô hình thực tế và kết quả nhận dạng và điều kiện nhận dạng vẫn còn trong điều kiện lý tưởng nên phần mềm vẫn chưa có khả năng để triển khai ra thực tế.

Hiện nay trên thì trường cũng có nhiều phần mềm nhận dạng biển số nhưng vẫn chưa có kết quả so sánh giữa phần mềm thử nghiệm với những phần mềm thực tế ấy để có thể đánh giá phần mềm của bản thân tự làm.

## Hướng phát triển

Hướng phát triển ngắn hạn:

* Về phần cứng, phần mềm vẫn chưa kết nối được với vần cứng nào vì vậy kết nối với camera và chức năng nhận dạng hình ảnh sẽ chuyển qua để nhận dạng theo thời gian thực sẽ là một cập nhật đáng kể cho phần mềm.
* Về mặt nhận dạng ký tự sẽ tìm hiểu thêm những thuật toán cho kết quả đúng hơn để thử nghiệm, từ đó thu thập thêm dữ liệu về những ký tự bị ảnh hưởng bởi những yếu tố bên ngoài.
* Bổ sung thêm chức năng nhận dạng đó là loại xe gì trước mắt có thể sẽ phân biệt được xe máy hoặc xe ô-tô. Chức năng này mặc dù nằm ngoài đề tài và không thuộc đối tượng tìm hiểu tới những vẫn sẽ thêm vào. Bởi vì đây là một chức năng giúp hoàn thiện phần mềm, tính toán chi phí đỗ xe cho từng loại xe phù hợp hơn nhưng do hạn chế về kiến thức và thời gian nên chưa thể đưa vào phần mề.

Hướng phát triển dài hạn:

* Tìm hiểu thêm mô hình thực tế bên ngoài để có thể xây dựng một hệ thống quản lý xe ra – vào hoàn chỉnh.
* Từ những tìm hiểu về mô hình thực tế bên ngoài phát triển phần mềm lên thành ứng dụng trên internet. Như vậy, mỗi lần có sự thay đổi về phần cứng như máy tính, ổ cứng bị hư hại cần thay thế thì sẽ không ảnh hưởng đến dữ liệu, hoặc không tốn công cài đặt lại phần mềm

# PHỤ LỤC

## Hướng dẫn cài đặt

Cài đặt Python

Chạy câu lệnh: Pip install tensorflowđể cài đặt thư viện Tensorflow

Sử dụng câu lệnh git clone <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.pygiao diện phần mềm sẽ hiện ra:

Bấm vào nút Recognize license để hiện lên giao diện nhận dạng biển số xe.

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

Ở giao diện nhận dạng biển số xe:

* Nút Load Image để tải ảnh lên.
* Nút Excute để bắt đầu nhận dạng biển số xe.
* Ở ô License plates chính là kết quả sau khi nhận dạng 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] | "Machine learning," 24/6/2021. [Online]. Available: https://en.wikipedia.org/wiki /Machine\_learning. [Accessed 15/7/2021]. |
| [2] | "Automatic number-plate recognition," 29/7/2021. [Online]. Available: https://en. wikipedia.org/wiki/Automatic\_number-plate\_recognition. [Accessed 30/7/2021]. |
| [3] | "Artificial neural network," 1/6/2021. [Online]. Available: https://en.wikipedia .org/wiki/Artificial\_neural\_network. [Accessed 30/6/2021]. |
| [4] | P. Đ. Khánh, "Convolutional Neural Network," 22/8/2019. [Online]. Available: https://phamdinhkhanh.github.io/2019/08/22/convolutional-neural-network.html. [Accessed 30/6/2021]. |
| [5] | C. R. S´ergio Montazzolli Silva, "License Plate Detection and Recognition in Un- constrained Scenarios," in *ECCV 2018*, Munich, Germany, 2018. |
| [6] | M. S. V. K. PANG-NING TAN, Introduction To Data Mining, Boston: Pearson Education, 2006. |
| [7] | "Python (programming language)," 1/6/2021. [Online]. Available: https://en.wiki- pedia.org/wiki/Python\_(programming\_language). [Accessed 30/6/2021]. |
| [8] | "TensorFlow," 24/6/2021. [Online]. Available: https://en.wikipedia.org/wiki/Tensor- Flow. [Accessed 30/6/2021]. |
| [9] | "About Keras," [Online]. Available: https://keras.io/about/. [Accessed 30/6/2021]. |
| [10] | "About SQLite," [Online]. Available: https://www.sqlite.org/about.html. [Acc- essed 30/6/2021]. |
| [11] | "Python - GUI Programming (Tkinter)," [Online]. Available: https://www.tut- orialspoint.com/python/python\_gui\_programming.htm. [Accessed 15/7/2021]. |
| [12] | "About OpenCV," [Online]. Available: https://opencv.org/about/. [Accessed 15/7 /2021]. |
| [13] | F. O. A. V. Annalisa Barla, "Histogram Intersection Kernel For Image," Proceedings 2003 International Conference on Image Processing, Barcelona, Spain, 2003. |