TRƯỜNG ĐẠI HỌC THỦ DẦU MỘT

VIỆN KỸ THUẬT CÔNG NGHỆ



**ĐỒ ÁN MÔN HỌC**

**NHẬP MÔN TRÍ TUỆ NHÂN TẠO**

Đề tài

**XÂY DỰNG ỨNG DỤNG**

**NHẬN DIỆN BIỂN BÁO GIAO THÔNG**

GVHD: Th.S NGUYỄN LÊ HIỀN DUYÊN

SVTH: Nguyễn Minh Hùng – 1824801030141

Phạm Ngọc Yến Nhi – 1824801030195

Bình Dương, 4/2021

**VIỆN KỸ THUẬT – CÔNG NGHỆ**

**NHẬN XÉT VÀ CHẤM ĐIỂM CỦA GIẢNG VIÊN**

Họ và tên giảng viên: **Nguyễn Lê Hiền Duyên**

Tên đề tài: **XÂY DỰNG ỨNG DỤNG NHẬN DIỆN BIỂN BÁO GIAO THÔNG**

Nội dung nhận xét:

**Điểm:**

Bằng số:

Bằng chữ:

|  |  |
| --- | --- |
|  | **GIẢNG VIÊN CHẤM**  *(Ký, ghi rõ họ tên)*  **NGUYỄN LÊ HIỀN DUYÊN** |
|  |  |

MỤC LỤC

[LỜI MỞ ĐẦU 3](#_Toc69332952)

[CHƯƠNG 1. TỔNG QUAN VỀ ĐỀ TÀI 4](#_Toc69332953)

[1. Ý tưởng dự án 4](#_Toc69332954)

[2. Định nghĩa bài toán 4](#_Toc69332955)

[3. Phương pháp thực hiện 4](#_Toc69332956)

[4. Giới thiệu công cụ 4](#_Toc69332957)

[4.1. Python 4](#_Toc69332958)

[4.2. TensorFlow 5](#_Toc69332959)

[4.3. Keras 5](#_Toc69332960)

[4.4. PIL 6](#_Toc69332961)

[4.5. Convolutional Neural Networks 6](#_Toc69332962)

[CHƯƠNG 2. PHÂN TÍCH THIẾT KẾ HỆ THỐNG 8](#_Toc69332963)

[1. Đối tượng dữ liệu 8](#_Toc69332964)

[2. Giải thuật thực hiện 8](#_Toc69332965)

[3. Phương pháp hoạt động 8](#_Toc69332966)

[3.1. Cách thức file train\_bb.py hoạt động 8](#_Toc69332967)

[3.2. Cách thức file GUI.py hoạt động 13](#_Toc69332968)

[4. Các bước thực hiện 16](#_Toc69332969)

[4.1. Chuẩn bị 16](#_Toc69332970)

[4.2. Tạo dữ liệu 17](#_Toc69332971)

[4.3. Training 17](#_Toc69332972)

[4.4. Sử dụng 17](#_Toc69332973)

[CHƯƠNG 3. VẬN HÀNH VÀ THỰC NGHIỆM 19](#_Toc69332974)

[1. Vận hành 19](#_Toc69332975)

[1.1. Quá trình vận hành 19](#_Toc69332976)

[1.2. Kết quả vận hành 19](#_Toc69332977)

[1.3. Nguyên nhân 20](#_Toc69332978)

[2. Thực nghiệm 20](#_Toc69332979)

[CHƯƠNG 4. KẾT LUẬN 21](#_Toc69332980)

[1. Kết quả đạt được 21](#_Toc69332981)

[2. Việc chưa làm được 21](#_Toc69332982)

[3. Hướng phát triển của đề tài 21](#_Toc69332983)

[TÀI LIỆU THAM KHẢO 22](#_Toc69332984)

# LỜI MỞ ĐẦU

Chắc hẳn bạn đã từng nghe về xe ô tô tự lái mà hành khách hoàn toàn có thể tin tưởng vào chiếc xe để di chuyển. Nhưng để đạt được mức độ chính xác và an toàn thì các phương tiện cần phải hiểu và tuân theo các quy tắc giao thông. Trong thế giới của trí tuệ nhân tạo và sự tiến bộ trong công nghệ, nhiều nhà nghiên cứu và các công ty lớn như Tesla, Uber, Google, Mercedes-Benz, Toyota, Ford, Audi, v.v. đang nghiên cứu về xe tự hành và xe tự lái. Để đạt được độ chính xác trong công nghệ này, các phương tiện cần phải giải thích được các biển báo giao thông và đưa ra quyết định sao cho phù hợp.

Với nhu cầu đó, nhóm chúng em quyết định chọn đề tài “Nhận diện biển báo giao thông”. Xây dựng một mô hình có thể phân loại các biển báo giao thông có trong hình ảnh thành các loại khác nhau. Có rất nhiều loại biển báo giao thông khác nhau như biển báo giới hạn tốc độ, biển cấm đi vào, tín hiệu giao thông, rẽ trái hoặc phải, trẻ em băng qua đường, cấm xe nặng vượt qua, v.v.

Khi chọn đề tài này, nhóm chúng em mong muốn sau khi hoàn thành có thể giúp các phương tiện đọc và hiểu các biển báo giao thông, một nhiệm vụ rất quan trọng đối với tất cả các phương tiện tự hành. Được sự giúp đỡ tận tình của cô Nguyễn Lê Hiền Duyên nhưng vẫn không thể tránh khỏi những sai xót, mong nhận được sự đóng góp và cảm thông của cô và các bạn.

Em xin chân thành cảm ơn.

Báo cáo bao gồm các phần:

**Chương 1. Tổng quan về đề tài**

**Chương 2. Phân tích thiết kế hệ thống**

**Chương 3. Vận hành và thử nghiệm**

**Chương 4. Kết quả đạt được**

# CHƯƠNG 1. TỔNG QUAN VỀ ĐỀ TÀI

## Ý tưởng dự án

* Xây dựng một ứng dụng có khả năng nhận dạng được các loại biển báo giao thông.

## Định nghĩa bài toán

* Tên dự án: Ứng dụng nhận diện biển báo giao thông.
* Sử dụng dữ liệu có sẵn để giúp máy tính có thể học và đưa ra kết quả đúng nhất có thể.
* Input: Hình ảnh biển báo giao thông.
* Output: Tên của biển báo giao thông ở input.

## Phương pháp thực hiện

* Dự án được phát triển bởi mô hình Machine Learning với công nghệ Norton Networks
* Sử dụng dữ liệu đầu vào tại Kaggle (https://www.kaggle.com/meowmeowmeowmeowmeow/gtsrb-german-traffic-sign)
* Sử dụng PIL, TensorFlow và Keras để phân tích, xây dựng, tính toán các dữ liệu.
* Thiết bị khởi chạy: Windows 10

## Giới thiệu công cụ

### Python

Python là một [ngôn ngữ lập trình](https://vi.wikipedia.org/wiki/Ng%C3%B4n_ng%E1%BB%AF_l%E1%BA%ADp_tr%C3%ACnh) bậc cao cho các mục đích lập trình đa năng, do [Guido van Rossum](https://vi.wikipedia.org/w/index.php?title=Guido_van_Rossum&action=edit&redlink=1) tạo ra và lần đầu ra mắt vào năm [1991](https://vi.wikipedia.org/wiki/1990). Python được thiết kế với ưu điểm mạnh là dễ đọc, dễ học và dễ nhớ. Python là ngôn ngữ có hình thức rất sáng sủa, cấu trúc rõ ràng, thuận tiện cho người mới học lập trình. Cấu trúc của Python còn cho phép người sử dụng viết mã lệnh với số lần gõ phím tối thiểu. Vào tháng 7 năm 2018, Van Rossum đã từ chức Leader trong cộng đồng ngôn ngữ Python sau 30 năm lãnh đạo.

Python hoàn toàn [tạo kiểu động](https://vi.wikipedia.org/w/index.php?title=T%E1%BA%A1o_ki%E1%BB%83u_%C4%91%E1%BB%99ng&action=edit&redlink=1) và dùng cơ chế [cấp phát bộ nhớ tự động](https://vi.wikipedia.org/wiki/Qu%E1%BA%A3n_l%C3%BD_b%E1%BB%99_nh%E1%BB%9B); do vậy nó tương tự như [Perl](https://vi.wikipedia.org/wiki/Perl), [Ruby](https://vi.wikipedia.org/wiki/Ruby_(ng%C3%B4n_ng%E1%BB%AF_l%E1%BA%ADp_tr%C3%ACnh)), [Scheme](https://vi.wikipedia.org/wiki/Scheme), [Smalltalk](https://vi.wikipedia.org/wiki/Smalltalk), và [Tcl](https://vi.wikipedia.org/wiki/Tcl). Python được phát triển trong một dự án mã mở, do tổ chức phi lợi nhuận Python Software Foundation quản lý.

Ban đầu, Python được phát triển để chạy trên nền [Unix](https://vi.wikipedia.org/wiki/Unix). Nhưng rồi theo thời gian, Python dần mở rộng sang mọi [hệ điều hành](https://vi.wikipedia.org/wiki/H%E1%BB%87_%C4%91i%E1%BB%81u_h%C3%A0nh) từ [MS-DOS](https://vi.wikipedia.org/wiki/MS-DOS) đến [Mac OS](https://vi.wikipedia.org/wiki/Mac_OS), OS/2, [Windows](https://vi.wikipedia.org/wiki/Microsoft_Windows), [Linux](https://vi.wikipedia.org/wiki/Linux) và [các hệ điều hành khác thuộc họ Unix](https://vi.wikipedia.org/wiki/T%C6%B0%C6%A1ng_t%E1%BB%B1_Unix). Mặc dù sự phát triển của Python có sự đóng góp của rất nhiều cá nhân, nhưng Guido van Rossum hiện nay vẫn là tác giả chủ yếu của Python. Ông giữ vai trò chủ chốt trong việc quyết định hướng phát triển của Python.

### TensorFlow

[TensorFlow](https://www.tensorflow.org/) chính là thư viện mã nguồn mở cho machine learning nổi tiếng nhất thế giới, được phát triển bởi các nhà nghiên cứu từ Google. Việc hỗ trợ mạnh mẽ các phép toán học để tính toán trong machine learning và deep learning đã giúp việc tiếp cận các bài toán trở nên đơn giản, nhanh chóng và tiện lợi hơn nhiều.

Các hàm được dựng sẵn trong thư viện cho từng bài toán cho phép TensorFlow xây dựng được nhiều neural network. Nó còn cho phép bạn tính toán song song trên nhiều máy tính khác nhau, thậm chí trên nhiều CPU, GPU trong cùng 1 máy hay tạo ra các dataflow graph – đồ thị luồng dữ liệu để dựng nên các model. Nếu bạn muốn chọn con đường sự nghiệp trong lĩnh vực A.I. này, nắm rõ những điều cơ bản của TensorFlow thực sự rất quan trọng.

Được viết bằng C++ và thao tác interface bằng Python nên phần performance của TensorFlow cực kỳ tốt. Đối tượng sử dụng nó cũng đa dạng không kém: từ các nhà nghiên cứu, nhà khoa học dữ liệu và dĩ nhiên không thể thiếu các lập trình viên.

### Keras

Keras là một library được phát triển vào năm 2015 bởi François Chollet, là một kỹ sư nghiên cứu deep learning tại google. Nó là một open source cho neural network được viết bởi ngôn ngữ python. keras là một API bậc cao có thể sử dụng chung với các thư viện deep learning nổi tiếng như tensorflow(được phát triển bởi gg), CNTK(được phát triển bởi microsoft), theano(người phát triển chính Yoshua Bengio). Keras có một số ưu điểm như:

* Dễ sử dụng, xây dựng model nhanh.
* Có thể run trên cả cpu và gpu
* Hỗ trợ xây dựng CNN , RNN và có thể kết hợp cả 2.

### PIL

Python cho phép giải quyết vấn đề các về xử lý hình ảnh thông qua thư viện **Imaging (PIL)**. Thư viện này hỗ trợ nhiều định dạng tập tin, và cung cấp khả năng xử lý hình ảnh và đồ hoạ mạnh mẽ.

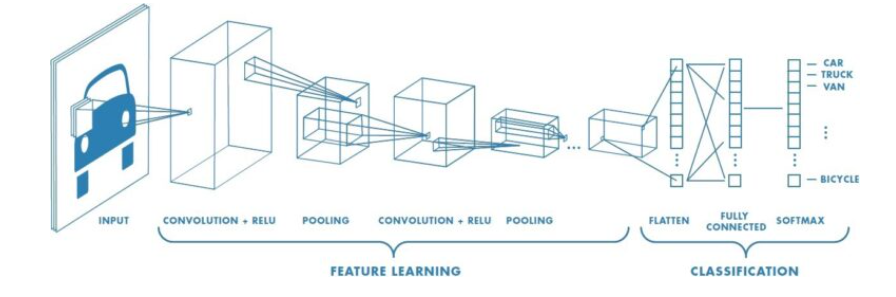
Thư viện xử lý ảnh của Python (Python Imaging Library) thêm vào cho trình thông dịch của python khả năng xử lý ảnh. Pillow là tập hợp của 3 chức năng: lưu trữ, trình chiếu, và xử lý thông tin

### Convolutional Neural Networks

Khác với mạng Neural Networks thông thường, mạng CNN gồm nhiều tầng convolution (lớp tích chập), non-linear activation layer (ReLU, sigmoid), pooling (lớp gộp) và cuối cùng là tầng fully connected.

Thông thường, một ảnh sẽ được lan truyền qua tầng convolution và non-linear activation đầu tiên, sau đó các giá trị tính toán được sẽ lan truyền qua pooling layer. Bộ ba convolution layer + nonlinear layer + pooling layer có thể được lặp lại nhiều lần trong network. Cuối cùng, tầng fully connected layer và softmax được dùng để tính xác suất ảnh đó thuộc lớp nào.

Hình sau minh hoạ một kiến trúc của CNN.



1. Kiến trúc của CNN

Cấu trúc cơ bản của CNN gồm: tầng convolution, tầng activation , tầng pooling, và tầng fully connected.

Trong tầng convolution và pooling, một phần của neural đầu vào được thu hẹp và liên kết với các tầng tiếp theo. Cùng với các filter giúp nhận diện được các đặc tính nào đó của ảnh. Nói cách khác, mỗi neural ở layer tiếp theo được sinh ra từ việc đặt filter lên 1 vùng ảnh cục bộ của neural layer trước.

# CHƯƠNG 2. PHÂN TÍCH THIẾT KẾ HỆ THỐNG

## Đối tượng dữ liệu

* Đối tượng dữ liệu là các hình ảnh về các biển báo giao thông
* Training sẽ tạo ra final\_model.h5, đây là nơi lưu trữ các dữ liệu đã được train.
* Dữ liệu Input là các hình ảnh cần nhận dạng.
* Dữ liệu Output là tên của biển báo giao thông ở input.
* Học máy sẽ làm nhiệm vụ học tập các trường hợp chính xác và gần giống của dữ liệu mẫu.
* Các đối tượng dữ liệu được phân chia theo các nhóm. Gồm 43 nhóm là 43 loại biển báo giao thông.

## Giải thuật thực hiện

* Giải thuật được xử lí dữ liệu bao gồm 5 bước:

+ Khám phá tập dữ liệu

+ Xây dựng mô hình CNN

+ Training và xác thực

+ Kiểm tra mô hình với bộ dữ liệu test set

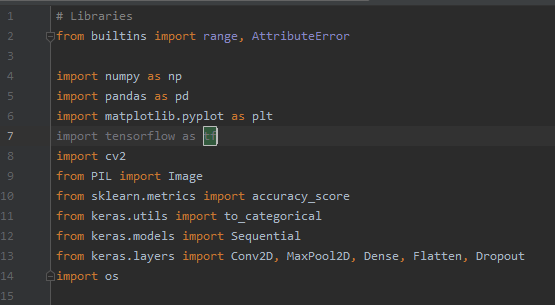
+ Xây dựng ứng dụng nhận dạng trên model đã training

## Phương pháp hoạt động

* Train\_bb.py là file chạy training nhận dạng biển báo giao thông.
* GUI.py là file giao diện của ứng dụng nhận dạng biển báo giao thông.

### Cách thức file train\_bb.py hoạt động

* Các thư viện sử dụng: numpy, pandas, cv2, matplotlib, tensorflow, PIL, keras, sklearn



1. Khai báo thư viện

* Khám phá tập dữ liệu:

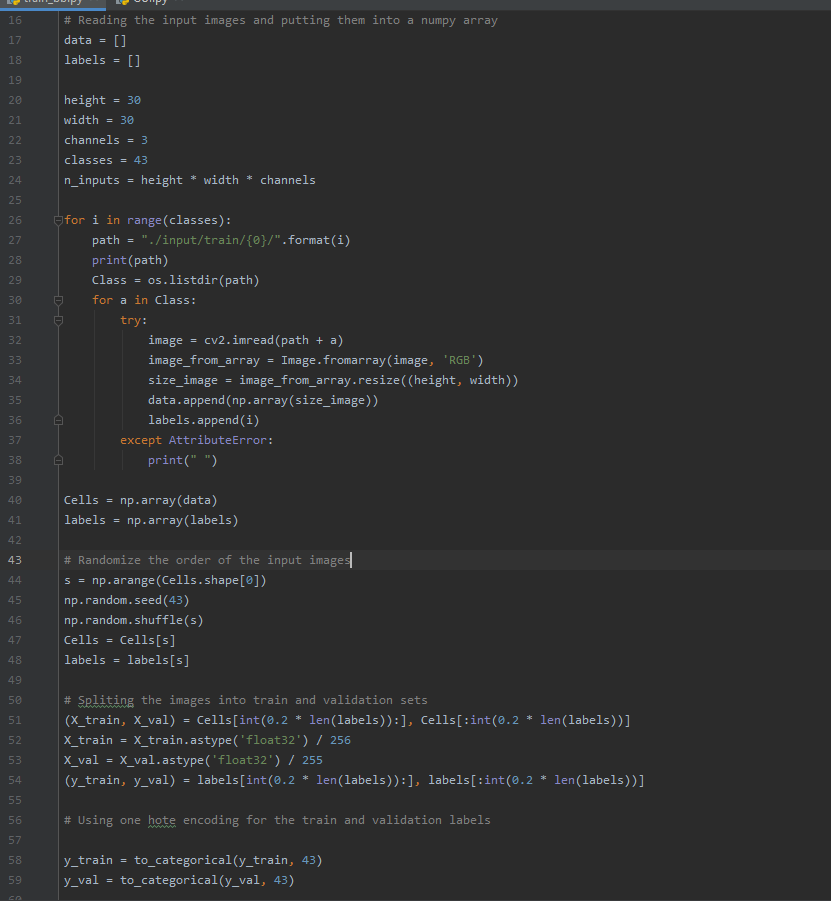
Thư mục train có chưa 43 thư mục con mỗi thư mục đại diện cho một lớp khác nhau được đánh số thứ tự từ 0 đến 42. Sử dụng module OS ta sẽ lặp lại trên tất cả các lớp để kết nối các hình ảnh với nhãn tương ứng của chúng.

Sau đó ta sẽ lưu tất cả các hình ảnh và nhãn của chúng vào lists data và labels.

Chúng ta chuyển các lists thành các numpy array để làm đầu vào cho model.

Kích thước của dữ liệu là  (39209, 30, 30, 3), có nghĩa là có 39209 hình ảnh, mỗi ảnh có kích thước 30×30 pixel, số 3 ở cuối thể hiện dữ liệu là hình ảnh màu (giá trị RGB).

Sử dụng phương thức to\_categorical của keras.utils để xử lý các nhãn của y\_train và y\_test bằng one-hot encoding.

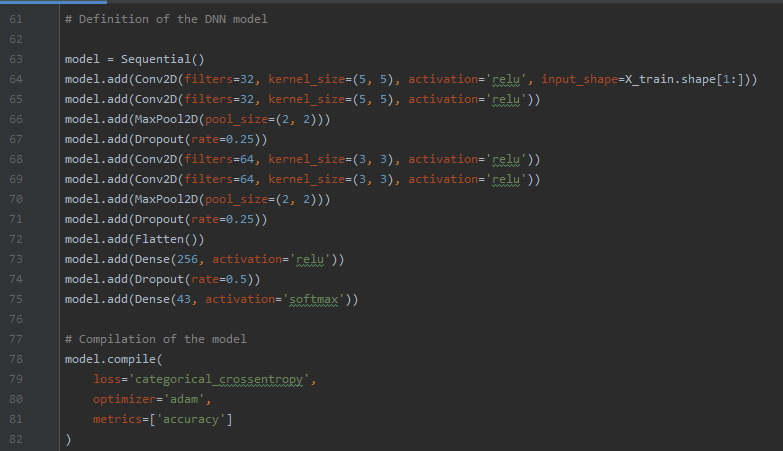


* Xây dựng mô hình CNN:

Để phân loại các hình ảnh thành các danh mục tương ứng, chúng ta sẽ xây dựng mô hình CNN (***Convolutional Neural Network***). CNN là mô hình tốt nhất cho mục đích phân loại hình ảnh.

Kiến trúc mô hình CNN của chúng ta như sau:

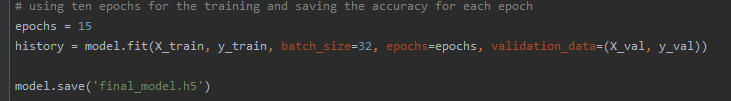
* 2 Conv2D layer (filter=32, kernel\_size=(5,5), activation=”relu”)
* MaxPool2D layer ( pool\_size=(2,2))
* Dropout layer (rate=0.25)
* 2 Conv2D layer (filter=64, kernel\_size=(3,3), activation=”relu”)
* MaxPool2D layer ( pool\_size=(2,2))
* Dropout layer (rate=0.25)
* Flatten layer to squeeze the layers into 1 dimension
* Dense Fully connected layer (256 nodes, activation=”relu”)
* Dropout layer (rate=0.5)
* Dense layer (43 nodes, activation=”softmax”)

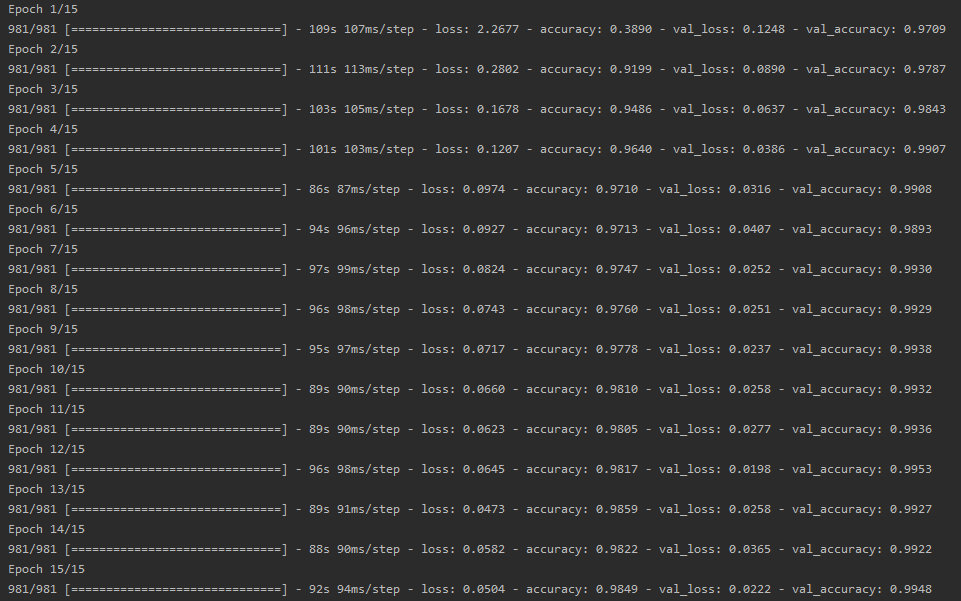


* Training và xác thực

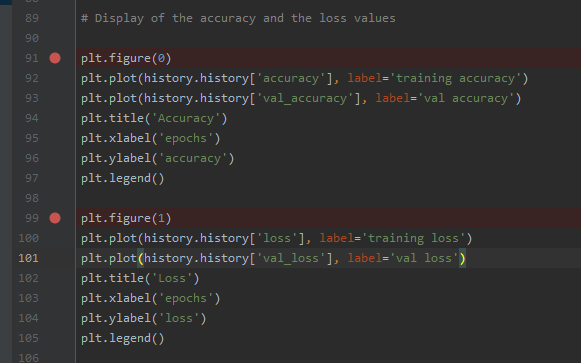
Sau khi xây dựng kiến ​​trúc mô hình, chúng ta sẽ huấn luyện mô hình bằng cách sử dụng model.fit ().

Sau đó chúng ta lưu model lại thành file final\_model.h5 để sử dụng.





Mô hình của chúng ta đã đạt đến độ chính xác ~ 97% trên tập training set. Sử dụng matplotlib để vẽ biểu đồ tương quan giữa độ chính xác và sự mất mát.



|  |  |
| --- | --- |
|  |  |
| Sơ đồ độ sai lệch | Sơ đồ độ chính sát |

* Quá trình training sẽ kết thúc cho đến khi training đủ 15 lần.
* Sau khi training, ta kiểm tra với test set được kết quả là độ chính xác đạt khoảng 97%



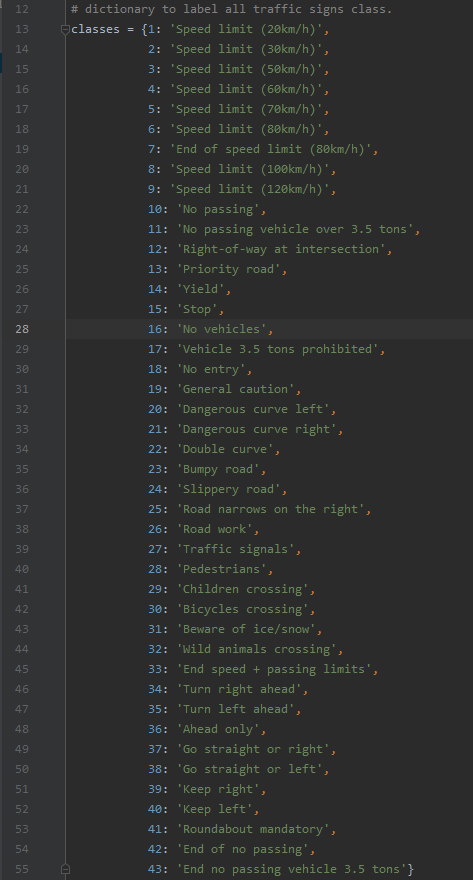
Tập dữ liệu của chúng ta có chứa 1 thư mục test và 1 file test.csv. Chúng ta cần phải trích xuất hình ảnh và nhãn tương ứng bằng cách sử dụng pandas. Sau đó, để dự đoán mô hình, chúng ta phải thay đổi kích thước hình ảnh thành 30 × 30 pixel và tạo một mảng numpy chứa tất cả dữ liệu hình ảnh. Sử dụng precision\_score của sklearn.metrics để dự đoán các nhãn của bộ test. Có thể thấy ta đã đạt được độ chính xác là 97% trong mô hình này.

### Cách thức file GUI.py hoạt động

* Các thư viện được sử dụng: tkinter, PIL, numpy, keras.
* Load file dữ liệu đã được train

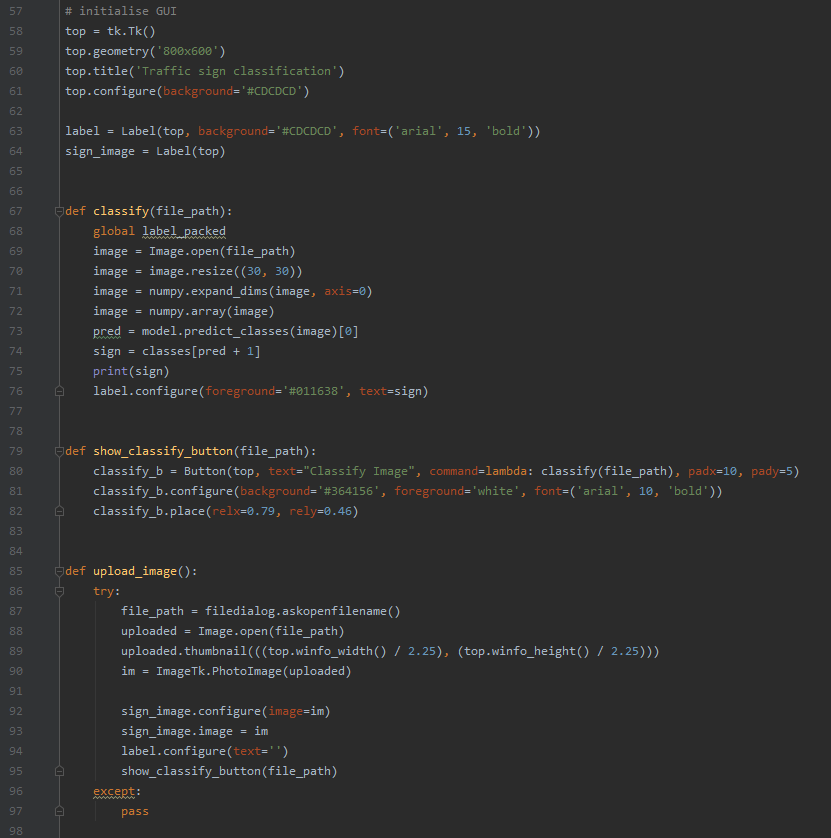


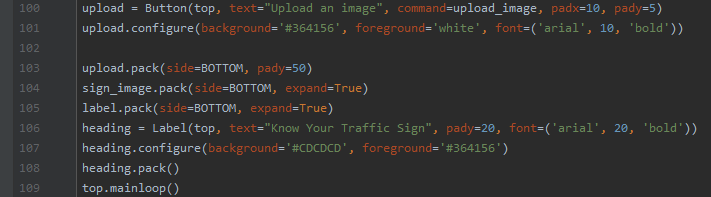
* Tạo thư viện chứa các id và tên của biển báo



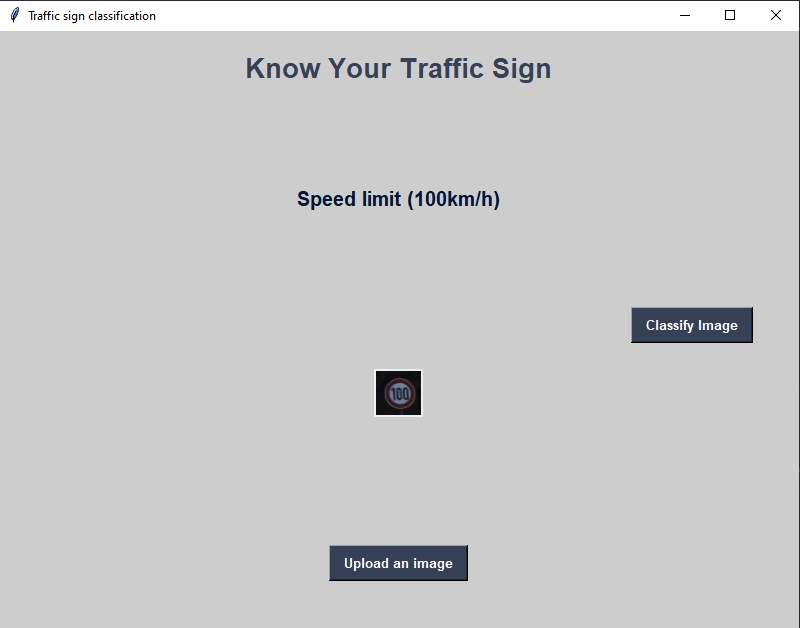
1. Thư viện chứa các id và tên của biển báo

* Thiết kế giao diện





1. Thiết kế giao diện



1. Giao diện chương trình

## Các bước thực hiện

### Chuẩn bị

* Cài đặt các thư viện cần thiết

+ Cài TensorFlow:

pip install tensorflow

+ Cài Keras:

pip install keras

+ Cài Sklearn:

pip install matplotlib

+ Cài Pandas:

pip install pandas

+ Cài PIL:

pip install pil

### Tạo dữ liệu

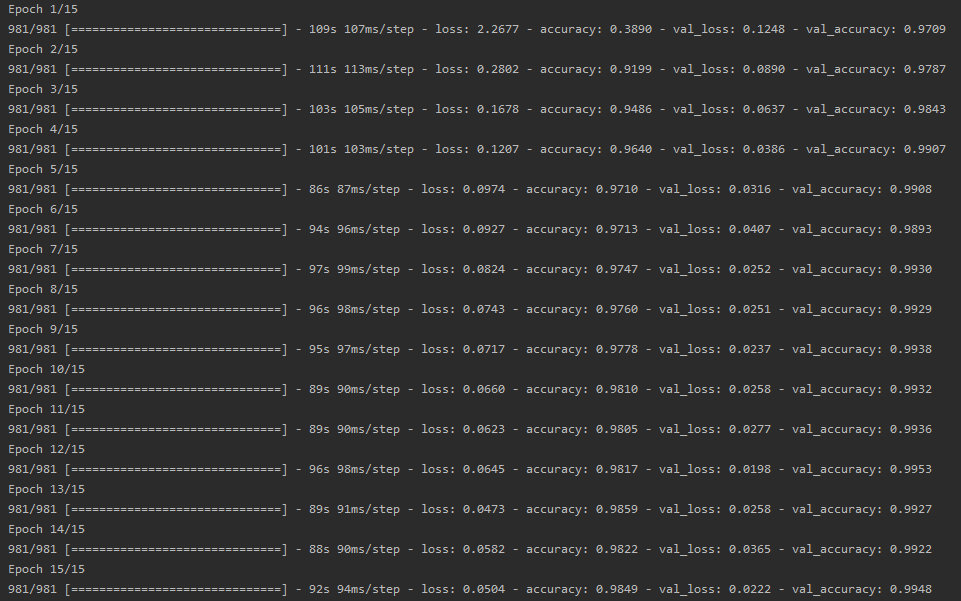
* Tải tập dữ liệu tại keggle (<https://www.kaggle.com/meowmeowmeowmeowmeow/gtsrb-german-traffic-sign>)

### Training

* Để training dữ liệu, ta dùng lệnh

python train\_bb.py

* Việc training sẽ được bắt đầu

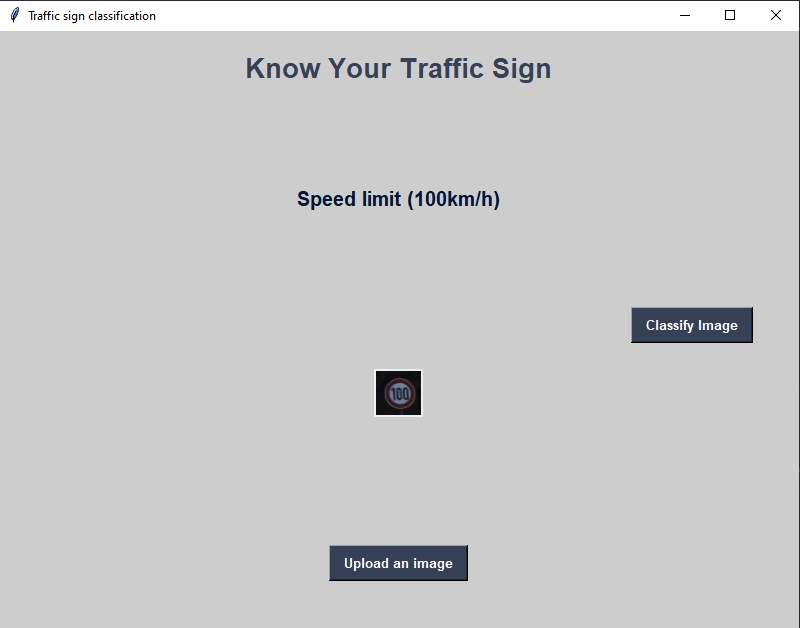


### Sử dụng

* Để có thể sử dụng được ta sẽ dùng lệnh:

python GUI.py

* Kết quả mong đợi sẽ là:



1. Giao diện kết quả

# CHƯƠNG 3. VẬN HÀNH VÀ THỰC NGHIỆM

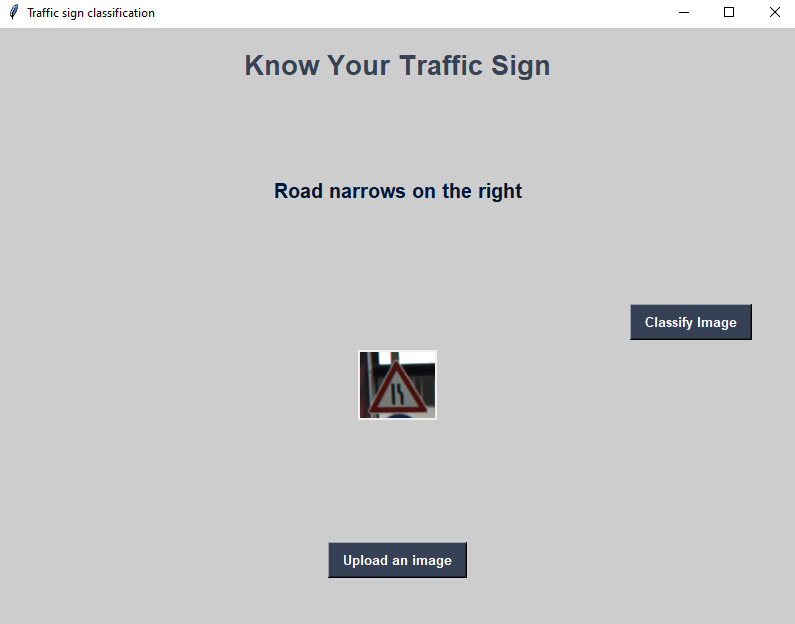
## Vận hành

### Quá trình vận hành

* Quá trình training là quá trình mất rất nhiều thời gian để đến được giá trị kì vọng.
* Tuy nhiên cũng vì mất nhiều thời gian nên khi training được ít dữ liệu, kết quả trả về sẽ bị sai.
* Khi dữ liệu càng được training nhiều độ chính sát sẽ tăng lên.

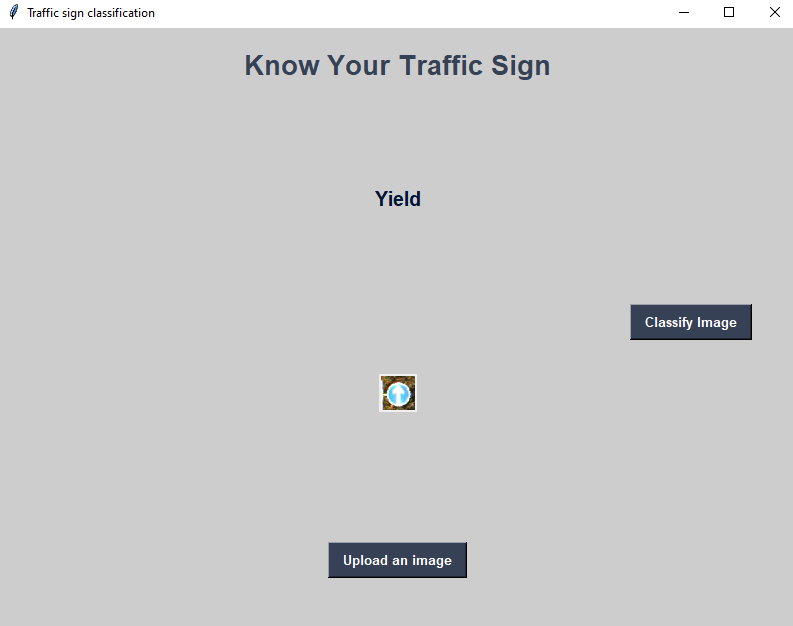
### Kết quả vận hành

* Kết quả thực tế khi training 10 lần, ta đạt được tỉ lệ chính xác khoảng 97%



1. Giao diện kết quả vận hành

* Vẫn có nhiều trường hợp kết quả bị sai



1. Giao diện khi kết quả bị sai

### Nguyên nhân

* Khi dữ liệu học ít và số lần train ít, hệ thống không được nhận biết nhiều dữ liệu mẫu nên câu trả lời bị sai.

## Thực nghiệm

* Vì độ chính xác đạt khoảng 97% nên nhận dạng đúng được hầu hết các biển báo được training.
* Trường hợp kết quả sai rất ít.

# CHƯƠNG 4. KẾT LUẬN

## Kết quả đạt được

* Biết cách xây dựng, thiết kế kiến trúc Machine Learning với kĩ thuật Norton Networks
* Đã tạo ra được một ứng dụng nhận dạng biển báo giao thông
* Thành thạo thao tác trên Python và các thư viện TensorFlow, PIL, Keras,...
* Thành thạo việc sửa lỗi, xây dựng, thay đổi kiến trúc Norton

## Việc chưa làm được

* Phần trăm sai sót và phần trăm chính sát chưa cao (Do chưa Training được nhiều dữ liệu)
* Giao diện còn sơ sài
* Chưa phát hiện và nhận diện trên ảnh, video, camera.

## Hướng phát triển của đề tài

* Trong tương lai, hệ thống sẽ được training nhiều dữ liệu hơn
* Tạo ra giọng nói cho AI
* Cho AI có khả năng lắng nghe người dùng
* Giảm độ phức tạp của giải thuật và tăng các giá trị, các dữ liệu lên
* Giao diện đẹp, thân thiện với người dùng, có nhiều chức năng và tương tác tốt hơn với người dùng.
* Thiết kế, xây dựng và lập trình bổ sung các tính năng còn thiếu hay chưa thật sự hoàn chỉnh trong quá trình sử dụng.

# TÀI LIỆU THAM KHẢO

* Các webside:

+ <https://www.tensorflow.org/?hl=ja>

+ <https://tek4.vn/nhan-dien-bien-bao-giao-thong-bang-cnn-keras/>

+ <https://aivietnam.ai/>

+ <https://thorpham.github.io/blog/2018/05/25/keras/>

+ <https://viblo.asia/p/huong-dan-su-dung-thu-vien-pillow-de-xu-ly-hinh-anh-trong-python-cho-nguoi-moi-bat-dau-3Q75wm4MZWb>

+ https://codelearn.io/sharing/scikit-learn-trong-python-la-gi