TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP HÀ NỘI

KHOA CÔNG NGHỆ THÔNG TIN

–––––––––––––––––––––––



BÁO CÁO KIỂM TRA THƯỜNG XUYÊN

HỌC PHẦN MỘT SỐ CÔNG NGHỆ PHÁT TRIỂN PHẦN MỀM

TRAFFIC SIGNS RECOGNITION USING CNN & KERAS IN PYTHON (OPENCV 2)

|  |  |
| --- | --- |
| GVHD: | TS Hà Mạnh Đào |
| Lớp: | 20224IT6024001 |
| Sinh viên: | Vũ Hồng Phương – 2020601638 |
|  |  |
|  |  |

Hà Nội - Năm 2023

**MỤC LỤC**

[1. Giới thiệu 3](#_Toc143886362)

[2. Công dụng của chương trình 3](#_Toc143886363)

[3. Phân tích cấu trúc chương trình 3](#_Toc143886364)

[4. Công nghệ sử dụng 6](#_Toc143886365)

[5. Kết quả chạy chương trình 6](#_Toc143886366)

[6. Giải thích kết quả 7](#_Toc143886367)

[TÀI LIỆU THAM KHẢO 8](#_Toc143886368)

# Giới thiệu

Báo cáo này trình bày về một ứng dụng sử dụng mạng neural tích hợp (Convolutional Neural Network - CNN) và thư viện Keras trong Python để nhận diện biển báo giao thông. Chương trình được phát triển nhằm mục tiêu tự động phân loại các biển báo giao thông từ ảnh chụp và đưa ra dự đoán về loại biển báo cụ thể.

# Công dụng của chương trình

Chương trình này có các ứng dụng quan trọng trong lĩnh vực giao thông và an toàn đường. Thay vì việc thủ công kiểm tra và phân loại các biển báo giao thông, ứng dụng này giúp tự động hoá quá trình này, tiết kiệm thời gian và giúp giảm nguy cơ sai sót từ con người.

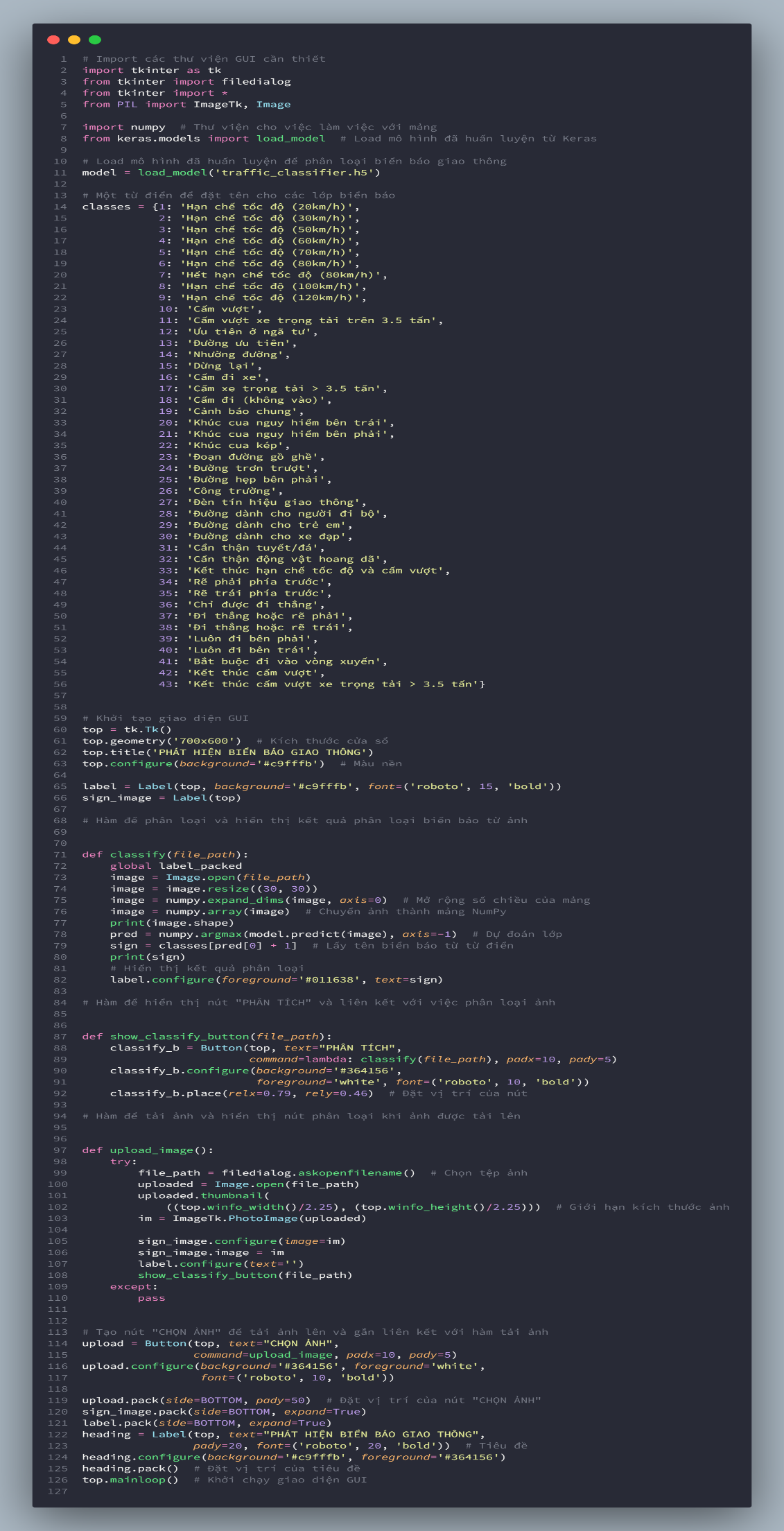
# Phân tích cấu trúc chương trình

Chương trình bao gồm hai phần chính: ***main.py*** để tạo giao diện người dùng và ***traffic\_sign.py*** để huấn luyện mô hình CNN.

**3.1. main.py**

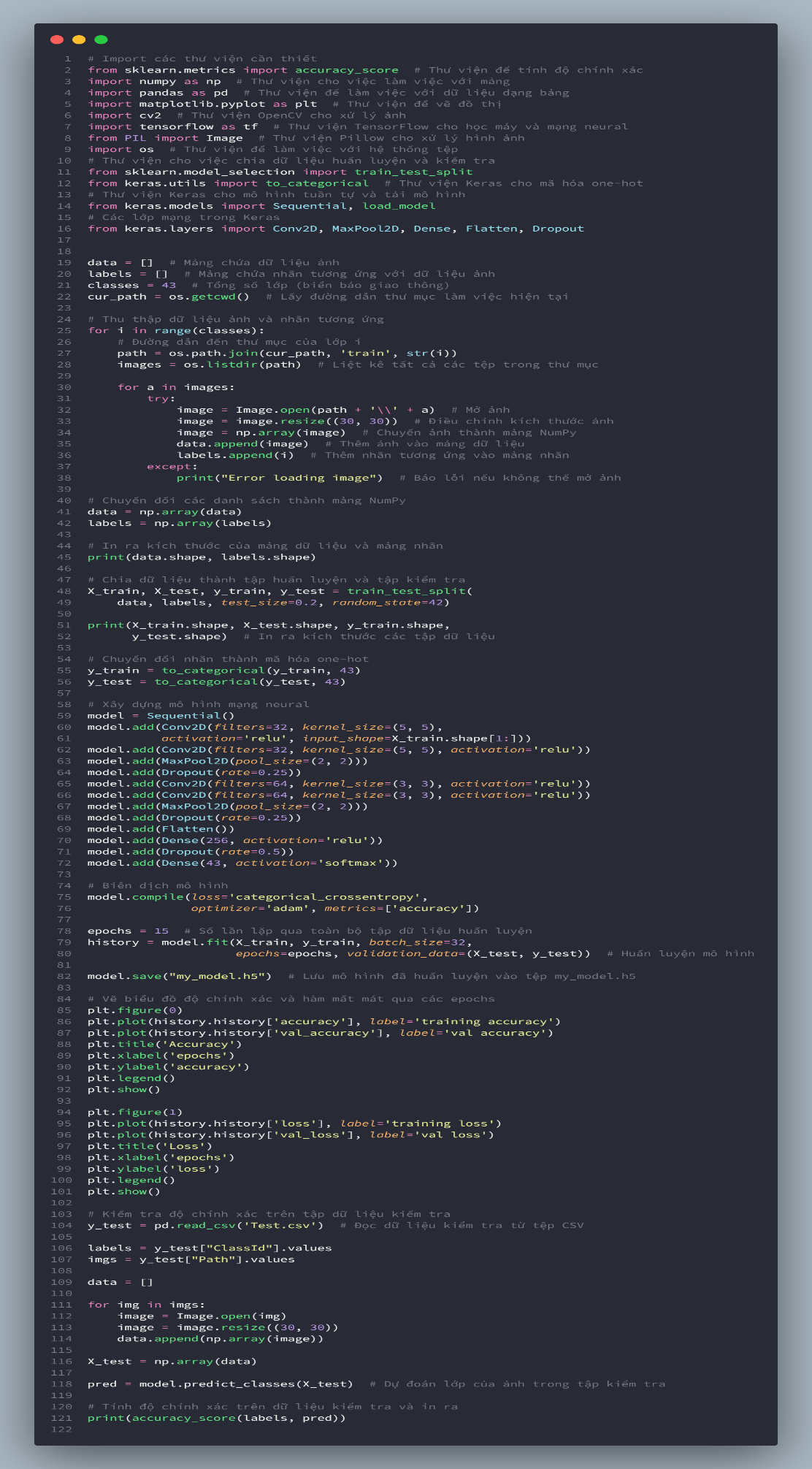
* **Import Thư Viện**: Đầu tiên, chương trình import các thư viện cần thiết như Tkinter để tạo giao diện người dùng, PIL để làm việc với hình ảnh, numpy để xử lý mảng và keras để load mô hình đã huấn luyện.
* **Load Mô Hình**: Chương trình load mô hình CNN đã huấn luyện từ tệp traffic\_classifier.h5.
* **Tạo Giao Diện GUI**: Sử dụng Tkinter, chương trình tạo một giao diện đơn giản để tải ảnh và hiển thị kết quả phân loại.
* **Phân Loại và Hiển Thị Kết Quả**: Khi người dùng chọn ảnh, chương trình thực hiện phân loại ảnh bằng cách sử dụng mô hình CNN và hiển thị kết quả lên giao diện.





**3.2. traffic\_sign.py**

* **Thu Thập Dữ Liệu**: Chương trình thu thập dữ liệu ảnh và nhãn tương ứng từ các thư mục tương ứng với các lớp biển báo giao thông.
* **Chia Dữ Liệu**: Sau khi thu thập dữ liệu, chương trình chia dữ liệu thành tập huấn luyện và tập kiểm tra.
* **Xây Dựng Mô Hình**: Chương trình xây dựng mô hình CNN sử dụng Keras. Mô hình gồm các lớp Convolutional, MaxPooling, Dropout và Fully Connected để học các đặc trưng từ ảnh.
* **Huấn Luyện Mô Hình**: Chương trình huấn luyện mô hình trên tập dữ liệu huấn luyện và kiểm tra.
* **Lưu Mô Hình**: Mô hình sau khi huấn luyện được lưu vào tệp traffic\_classifier.h5 để sử dụng sau này.



# Công nghệ sử dụng

Chương trình sử dụng thư viện Keras để xây dựng và huấn luyện mô hình CNN. Keras cung cấp một cách tiếp cận dễ dàng để tạo các kiến trúc mạng và huấn luyện mô hình với dữ liệu hình ảnh. Chương trình cũng sử dụng thư viện Tkinter để tạo giao diện người dùng đơn giản.

# Kết quả chạy chương trình

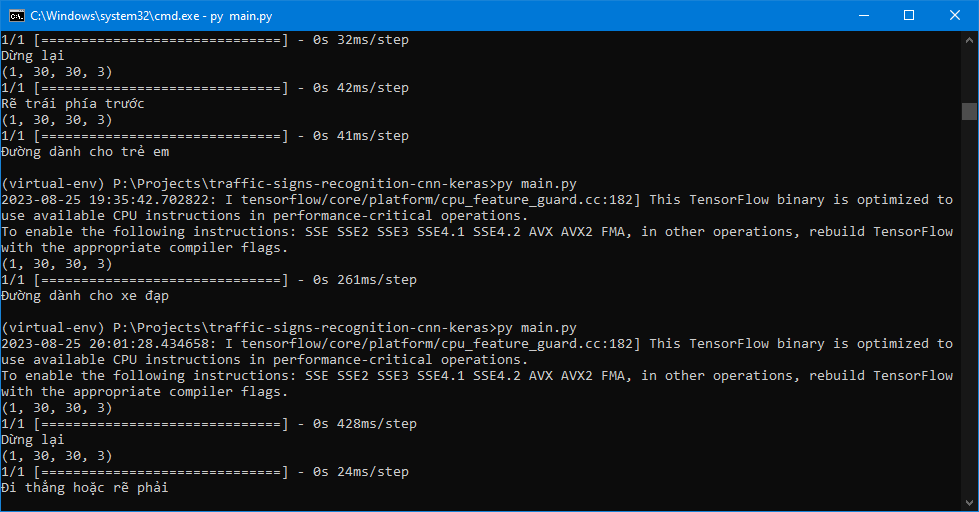
Sau khi mô hình được huấn luyện và giao diện người dùng được tạo, chương trình có khả năng tải ảnh, phân loại và hiển thị kết quả phân loại lên giao diện. Kết quả đạt được sau khi huấn luyện có thể được đánh giá bằng cách tính độ chính xác trên tập dữ liệu kiểm tra.





# Giải thích kết quả

Chương trình cho thấy khả năng nhận diện và phân loại biển báo giao thông từ hình ảnh chụp. Độ chính xác của mô hình có thể được đánh giá bằng cách so sánh các dự đoán với nhãn thực tế trên tập dữ liệu kiểm tra. Các biểu đồ độ chính xác và hàm mất mát qua các epoch cũng thể hiện quá trình huấn luyện và kiểm tra mô hình.



# TÀI LIỆU THAM KHẢO

<https://itsourcecode.com/free-projects/python-projects/traffic-signs-recognition-using-cnn-keras-in-python-with-source-code/?expand_article=1>