

**BỘ GIÁO DỤC VÀ ĐÀO TẠO**

**TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT TP.HCM**

**KHOA ĐÀO TẠO CHẤT LƯỢNG CAO**

**🙥🙤🙥🙤🕮🙦🙧🙦🙧**

****

**Trí Tuệ Nhân Tạo**

**NHẬN DẠNG NGÔN NGỮ TIẾNG VIỆT**

**QUA VĂN BẢN WEB,APP**

**GVHD: PGS.TS NGUYỄN TRƯỜNG THỊNH**

**SVTH: NGUYỄN NHẬT TIẾN MSSV:19146273**

**Ngày 20 tháng 6 năm 2022, TP. Hồ Chí Minh**

**BẢNG PHÂN CÔNG NHIỆM VỤ**

**ĐỀ TÀI: Nhận dạng ngôn ngữ tiếng việt qua văn bản web,app**

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **HỌ TÊN** | **MSSV** | **HOÀN THÀNH** |
| **1** | Nguyễn Nhật Tiến | 19146273 | 100% |

**Nhận xét của giảng viên**

**KÝ TÊN**

**LỜI CẢM ƠN**

Lời đầu tiên cho tôi xin gửi lời cảm ơn đến cha mẹ đã sinh thành và nuôi dưỡng tôisuốt thời gian qua để hôm nay được ngồi trên giảng đường đại học và thực hiện ước mơ của mỗi bản thân.

Tiếp đến, tô ixin gửi lời cảm ơn chân thành tới trường Đại học Sư phạm Kỹ thuật Thành phố Hồ Chí Minh đã tạo điều kiện vật chất và cả tinh thần để nhóm được học hỏi, giao lưu tìm tòi và sáng tạo.

Trong quá trình học tập và tìm hiểu đề tài lần này, tôi đã được nhận sự hướng dẫn, chỉ bảo tận tình và đầy tâm huyết của thầy Nguyễn Trường Thinh. Thầy đã mang đến cho tôi nhiều kiến thức chuyên môn trong mảng trí tuệ nhân tạo, máy học,….

Do còn thiếu kinh nghiệm cũng như những hạn chế về kiến thức, trong bài mô phỏng và báo cáo sẽ không tránh khỏi những thiếu sót. Rất mong nhận được sự đánh giá, ý kiến đóng góp, phê bình từ phía Thầy một cách trực quan để cải thiện hơn cho sau này.

Cuối lời, kính chúc Thầy có một sức khỏe dồi dào, thành công, hạnh phúc hơn trong quãng đường sắp tới và cùng nhau bước qua cơn đại dịch mang tên Covid.

**MỤC LỤC**

[**PHỤ LỤC HÌNH ẢNH 6**](#_Toc106789384)

[**CHƯƠNG 1: GIỚI THIỆU TỔNG QUAN 1**](#_Toc106789385)

[**1. Đặt vấn đề 1**](#_Toc106789386)

[**2. Ứng dụng thực tiễn 1**](#_Toc106789387)

[**3. Giới hạn đề tài 1**](#_Toc106789388)

[**CHƯƠNG 2: XÂY DỰNG MÔ HÌNH 2**](#_Toc106789389)

[**2.1 Chuẩn bị dữ liệu 2**](#_Toc106789390)

[**2.2 Tiền xử lý dữ liệu 3**](#_Toc106789391)

[**2.2.1 Bộ dữ liệu văn bản tiếng việt 3**](#_Toc106789392)

[**2.2.1.1 Thư viện được sử dụng 3**](#_Toc106789393)

[**2.2.1.2 Dataframe 4**](#_Toc106789394)

[**2.2.1.3 Tiền xử lý dữ liệu 5**](#_Toc106789395)

[**2.2.2 Bộ dữ liệu hình ảnh văn bản 7**](#_Toc106789396)

[**2.2.2.1 Thư viện được sử dụng 7**](#_Toc106789397)

[**2.2.2.2 Tạo dữ liệu hình ảnh 8**](#_Toc106789398)

[**2.3 Xây dựng mô hình 10**](#_Toc106789399)

[**2.3.1 Mô hình dự đoán văn bản ngôn ngữ tiếng việt 11**](#_Toc106789400)

[**2.3.2 Mô hình nhận diện ngôn ngữ tiếng việt bằng hình ảnh 12**](#_Toc106789401)

[**CHƯƠNG 3: MÔ PHỎNG VÀ ĐÁNH GIÁ CHẤT LƯỢNG MÔ HÌNH 19**](#_Toc106789402)

[**3.1 Mô hình dự đoán văn bản ngôn ngữ tiếng việt 19**](#_Toc106789403)

[**3.2 Mô hình nhận diện văn bản tiếng việt qua hình ảnh 20**](#_Toc106789404)

[**3.3 Real time detection 24**](#_Toc106789405)

[**3.4 Đánh giá mô hình 25**](#_Toc106789406)

[**CHƯƠNG 4 : THIẾT KẾ GIAO DIỆN WEB 26**](#_Toc106789407)

[**4.1 HTML 26**](#_Toc106789408)

[**4.2 Flask 26**](#_Toc106789409)

[**4.3 Giao diện Website 27**](#_Toc106789410)

[**CHƯƠNG 5: KẾT LUẬN VÀ ĐỊNH HƯỚNG 30**](#_Toc106789411)

[**5.1 Kết luận 30**](#_Toc106789412)

[**5.2 Định hướng 30**](#_Toc106789413)

[**TÀI LIỆU THAM KHẢO 31**](#_Toc106789414)

# **PHỤ LỤC HÌNH ẢNH**

[Hình 2. 1 Load dữ liệu file csv vào dataframe 2](#_Toc106648929)

[Hình 2. 2 File chứa hình ảnh văn bản 2](#_Toc106648930)

[Hình 2. 3 Thư viện cần thiết cho mô hình nhận dạng 3](#_Toc106648931)

[Hình 2. 4 Kiểm tra dữ liệu trong dataframe 4](#_Toc106648932)

[Hình 2. 5 Tách hai cột dữ liệu vào hai biến độc lập 4](#_Toc106648933)

[Hình 2. 6 Biến chữ in hoa thành chữ thường 5](#_Toc106648934)

[Hình 2. 7 Hàm loại bỏ các kí tự đặc biệt ra khỏi văn bản 6](#_Toc106648935)

[Hình 2. 8 Áp dụng hàm loại bỏ kí tự đặc biệt ra khỏi văn bản vào dữ liệu đã tạo 6](#_Toc106648936)

[Hình 2. 9 Phân chia dữ liệu để huấn luyện mô hình 6](#_Toc106648937)

[Hình 2. 10 Xử lý dữ liệu với TfidVectorizer 7](#_Toc106648938)

[Hình 2. 11 Thư viện cần thiết cho mô hình phân loại 7](#_Toc106648939)

[Hình 2. 12 Dạng văn bản dùng để huấn luyện 9](#_Toc106648940)

[Hình 2. 13 Tạo sự đa dạng cho hình ảnh với ImageDataGenerator 10](#_Toc106648941)

[Hình 2. 14 Số class của mô hình 10](#_Toc106648942)

[Hình 2. 15 Tạo pipline và bắt đầu việc huấn luyện mô hình 11](#_Toc106648943)

[Hình 2. 16 Độ chính xác mô hình sau khi huấn luyện 11](#_Toc106648944)

[Hình 2. 17 Sơ đồ của việc xử lý ảnh qua mạng nơron CNN 12](#_Toc106648945)

[Hình 2. 18 Ví dụ về lớp Convolution-relu 13](file:///C:\Users\Tien1\Desktop\LỜI%20CẢM%20ƠN.docx#_Toc106648946)

[Hình 2. 19 Ví dụ về lớp Pooling 14](#_Toc106648947)

[Hình 2. 20 Thiết lập mạng thần kinh nơ ron 14](#_Toc106648948)

[Hình 2. 21 Mạng nơ ron được xây dựng cho mô hình phân loại văn bản tiếng việt 15](#_Toc106648949)

[Hình 2. 22 Bảng tổng kết của mô hình mạng thần kinh nơ ron 16](#_Toc106648950)

[Hình 2. 23 Thuật toán tối ưu và hàm mất mát cho mô hình 18](#_Toc106648951)

[Hình 2. 24 Huấn luyện mô hình 18](#_Toc106648952)

[Hình 2. 25 Độ chính xác sau 10 lần học 18](#_Toc106648953)

[Hình 3. 1 Hàm dự đoán ngôn ngữ qua việc nhập 19](#_Toc106789425)

[Hình 3. 2 Kết quả dự đoán của hàm 19](#_Toc106789426)

[Hình 3. 3 Hình ảnh dùng cho việc dự đoán 20](#_Toc106789427)

[Hình 3. 4 Code cho việc dự đoán 21](#_Toc106789428)

[Hình 3. 5 Dự đoán với hình ảnh vân bản không phải tiếng việt 22](#_Toc106789429)

[Hình 3. 6 Dự đoán với hình ảnh vân bản tiếng việt 22](#_Toc106789430)

[Hình 3. 7 Những hình ảnh dự đoán sai của mô hình phân loại 23](#_Toc106789431)

[HÌnh 3. 8 Thử nghiệm camera của laptop 24](#_Toc106789432)

[HÌnh 3. 9 Với hình ảnh văn bản tiếng việt 24](file:///C:\Users\Tien1\Desktop\NguyenNhatTien_19146273_original.docx#_Toc106789433)

[HÌnh 3. 10 Với hình ảnh văn bản ngôn ngữ khác 25](#_Toc106789434)

[Hình 4. 1 Flask Framework 26](#_Toc106648961)

[Hình 4. 2 Giao diện cơ bản của website 26](#_Toc106648962)

[Hình 4. 3 Lựa chọn hình ảnh từ máy 27](#_Toc106648963)

[Hình 4. 4 Tiến hành dự đoán 27](#_Toc106648964)

[Hình 4. 5 Bài báo được chọn để trích văn bản 28](#_Toc106648965)

[Hình 4. 6 Kết quả dự đoán sau khi add link bài báo 28](#_Toc106648966)

# **CHƯƠNG 1: GIỚI THIỆU TỔNG QUAN**

1. **Đặt vấn đề**

Với nhu cầu muốn tìm hiểu rõ hơn về trí tuệ nhân tạo và đặc biệt là về NLP (Neuro – Linguistic – Programming) thì tôi đã chọn đề tài nhận dạng văn bản tiếng việt sử dụng machine learning + NLP với những kiến thức đã được học ở lớp và tự tìm hiểu khi ở nhà nhầm tạo ra một model có thể dự đoán ngôn ngữ tiếng việt thông qua web hoặc hình ảnh văn bản.

1. **Ứng dụng thực tiễn**

Xử lý văn bản được lấy từ một link website bất kì hoặc xử lý hình ảnh có chứa văn bản từ người dùng nhập vào rồi từ đó đi so sánh với mô hình đã được huấn luyện sẵn. Tiếp đến mô hình sẽ xử lý là dự đoán là dữ liệu người dùng nhập vào có phải là tiếng việt hay không rồi sau đó trả ra kết quả.

Tất cả sẽ được tương tác trên một website tự tạo.

1. **Giới hạn đề tài**

Đề tài chỉ dừng lại ở việc nhận biết ngôn ngữ tiếng việt chứ không phải các dạng ngôn ngữ khác.

Đề tài sẽ chỉ sử dụng CNN (mạng nơron tích chập) và NLP cơ bản để xử lý nên độ chính xác sẽ không cao như khi dùng các model đã được pretrain như YOLO,VGG16,…

Với việc không có sẵn dữ liệu để huấn luyện mô hình nên dữ liệu tự thêm có thể không được hiệu quả và tối ưu cho mô hình nhận diện

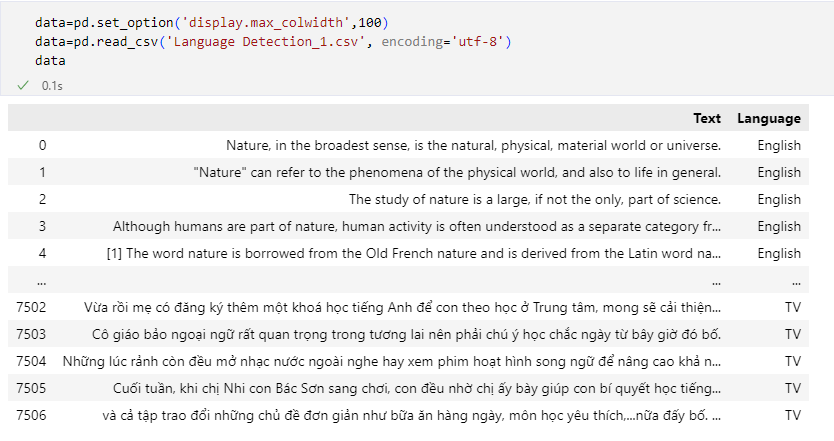
Chưa thể dùng để làm một ứng dụng cho thực tế.

# **CHƯƠNG 2: XÂY DỰNG MÔ HÌNH**

## **2.1 Chuẩn bị dữ liệu**

Ban đầu ta sẽ có 2 bộ dữ liệu thô gồm một bộ dữ liệu cho việc xử lý văn bản tiếng việt và một bộ xử lý dữ liệu dành cho việc nhận dạng văn bản tiếng việt thông qua hình ảnh.

Với dữ liệu sẽ được chọn lọc từ trên internet đối với bộ dữ liệu xử lý văn bản thì dữ liệu chủ yếu sẽ được lấy từ các bài báo, văn bản mẫu, công văn,… và được tổng hợp vào một file csv gồm 7506 dòng văn bản.



Hình 2. 1 Load dữ liệu file csv vào dataframe

Đối với dữ liệu hình ảnh để nhận diện là hình ảnh được cắt trực tiếp từ các bài báo như vn express, báo mới, new york times , bcc ,……..

Và dữ liệu hình ảnh sẽ được chia thành 2 tập với một tập là TV và một tập là NTV

Với TV là 335 tấm hình văn bản tiếng việt

Với NTV là 270 tấm hình chứa các văn bản ngôn ngữ khác như tiếng anh, pháp, ý,….



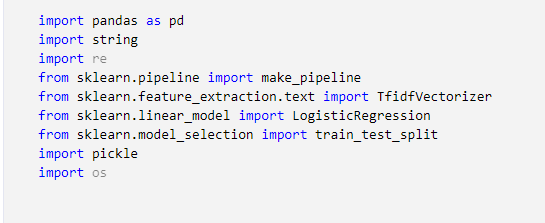
Hình 2. 2 File chứa hình ảnh văn bản

## **2.2 Tiền xử lý dữ liệu**

Sử dụng Visual Studio code để tiến hành xử lý dữ liệu cũng như dựng model với ngôn ngữ lập trình khá phổ biến của hiện tại là Python.

### 2.2.1 Bộ dữ liệu văn bản tiếng việt

### 2.2.1.1 Thư viện được sử dụng



Hình 2. 3 Thư viện cần thiết cho mô hình nhận dạng

+ Thư viện pandas : Được sử dụng để đưa dữ liệu vào dataframe để tiến hành tiền xử lý

+ Thư viện string : Được dùng để xử lý chuyên sâu về chuỗi

+ Thư viện re : Thư viện chuỗi các kí tự được sử dụng để tìm kiếm một mẫu trong chuỗi (Regular Expression)

+ Thư viện make\_pipeline của sklearn: Thư viện make\_pipeline dùng để xử lý các thuật toán học máy theo một chuỗi

+ Thư viện TfidVectorizer: Thư viện dùng để đánh giá tầm quan trọng của các từ trong văn bản dựa theo tần suất xuất hiện

+ Thư viện thuật toán Hồi Quy Logistic từ sklearn

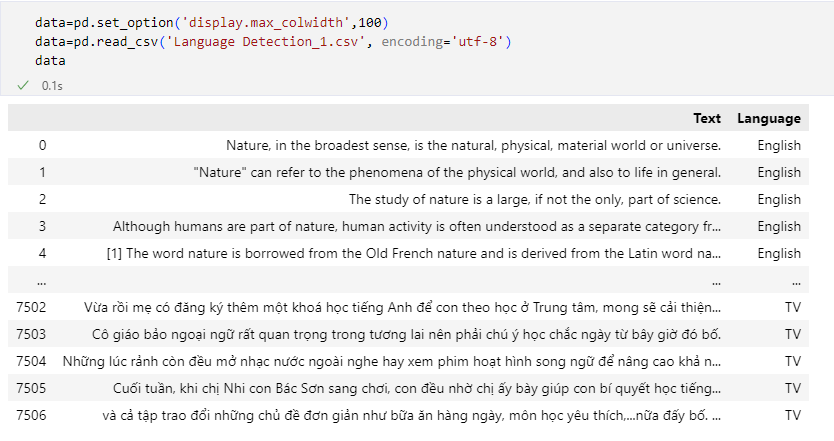
+ Thư viện phân chia tập dữ liệu (train\_test\_split) của sklearn dùng để phục vụ việc chia dữ liệu ra để huấn luyện mô hình

+ Thư viện pickle dùng để lưu model sau khi train

+ Module OS để có thể tương tác với hệ điều hành

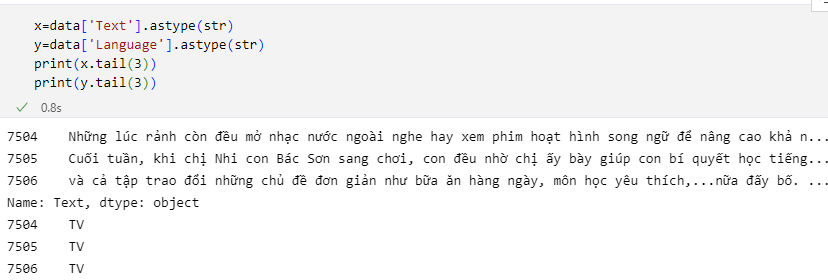
### 2.2.1.2 Dataframe

Sử dụng thư viện pandas đọc dữ liệu từ file Language Detection\_1.csv đã được chuẩn bị vào Dataframe để tiến hành phục vụ cho việc tiền xử lý dữ liệu .



Hình 2. 4 Kiểm tra dữ liệu trong dataframe

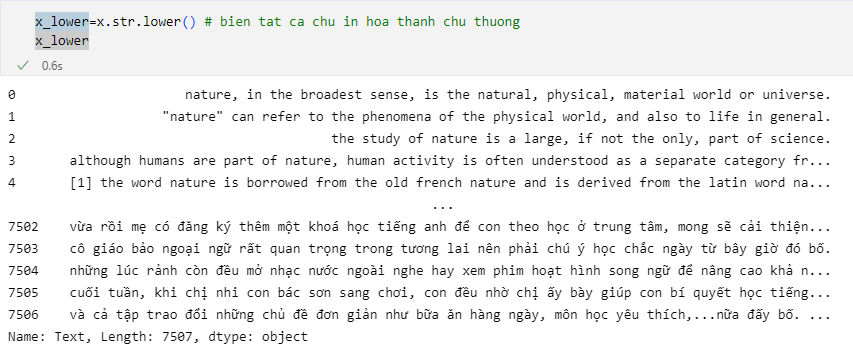
Ta chia dữ liệu của từng cột vào hai biến x và y để có thể tiến hành chia tập dữ liệu để huấn luyện thông qua thư viện train\_test\_split



Hình 2. 5 Tách hai cột dữ liệu vào hai biến độc lập

### 2.2.1.3 Tiền xử lý dữ liệu

Việc đầu tiên khi chúng ta xử lý văn bản là biến tất cả chữ in hoa thành chữ thường thông qua thư viện xử lý chuỗi string.



Hình 2. 6 Biến chữ in hoa thành chữ thường

Tiếp đến ta sẽ loại bỏ các kí tự đặc biệt trong văn bản để khiến cho văn bản được đưa vào trở nên sạch hơn

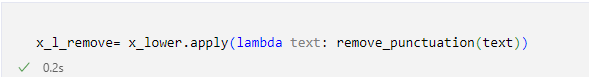
Ta tạo một function trả về một bản sao của chuỗi ban đầu trong đó các ký tự đã được thay thế sử dụng dictionary hoặc mapping table (được xây dựng với hàm maketrans()).

Cụ thể với những kí tự đặc biệt trong văn bản sẽ được thay thế thành khoảng trắng



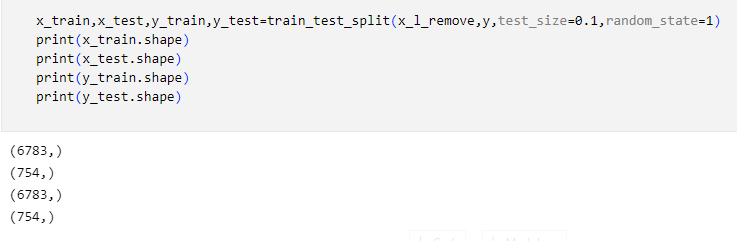
Hình 2. 7 Hàm loại bỏ các kí tự đặc biệt ra khỏi văn bản

Ta áp dụng function loại bỏ kí tự đặc biệt vào tập dữ liệu đã tạo và sau đó tiến hành chia tập dữ liệu để huấn luyện mô hình



Hình 2. 8 Áp dụng hàm loại bỏ kí tự đặc biệt ra khỏi văn bản vào dữ liệu đã tạo

Với x\_train, y\_train là dữ liệu dùng để huấn luyện và x\_test, y\_test dùng để tiên đoán sau khi mô hình huấn luyện xong



Hình 2. 9 Phân chia dữ liệu để huấn luyện mô hình

Tfid (viết tắt của Term Frequency- Inverse Document Frequency là một phương thức thống kê dùng để truy xuất thông tin (information retrieval) và khai phá dữ liêu văn bản (text mining) dùng để đánh giá mức độ quan trọng của các cụm từ trong một tập dữ liệu.Tfid được sử dụng nhiều trong lĩnh vực ngôn ngữ học (linguistics) và cấu trúc thông tin (information architecture) nhờ vào khả năng hỗ trợ xử lý nhiều tài liệu với số lượng lớn trong thời gian ngắn.

Đối với đề tài thì hàm TfidfVectorizer() sẽ được dùng để phân thích các cụm từ có độ quan trọng trong một đoạn văn bản để xác định loại ngôn ngữ và biến nó thành một vector đặc trưng được sử dụng để huấn luyện mô hình dự đoán ngôn ngữ

Ta sẽ lấy một cụm gồm 3 từ có mức độ quan trọng cao trong từng đoạn văn bản trong tập dữ liệu và biến nó thành một vector đặc trưng và sẽ lấy vector đó là dữ liệu đầu vào để huấn luyện mô hình nhận dạng văn bản tiếng việt.



Hình 2. 10 Xử lý dữ liệu với TfidVectorizer

### 2.2.2 Bộ dữ liệu hình ảnh văn bản

### 2.2.2.1 Thư viện được sử dụng



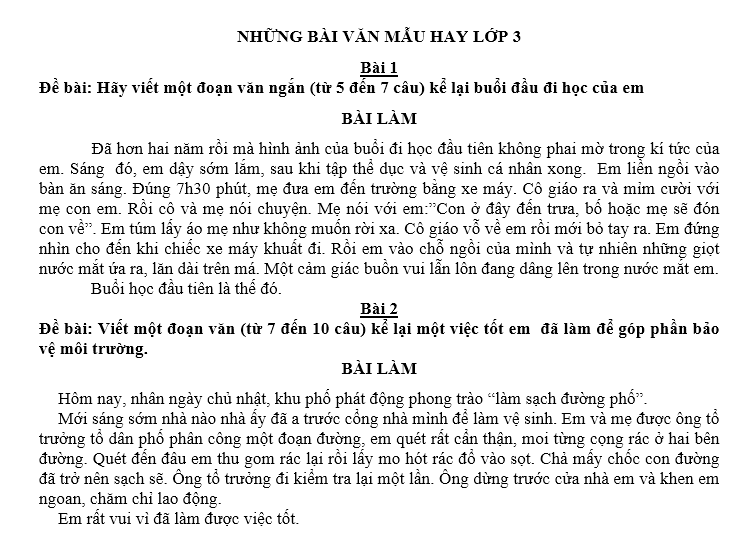
Hình 2. 11 Thư viện cần thiết cho mô hình phân loại

* import numpy as np : Sử dụng thư viện numpy
* from keras.models import load\_model : Thư viện dùng để load model sau khi huấn luyện xong (đã save)
* from keras.utils import load\_img,img\_to\_array, to\_categorical: Thư viện dùng để tải lên hình ảnh, biến ảnh thành mảng và biết ảnh là thành dữ liệu phân loại
* from keras.optimizers import SGD,Adam,RMSprop : Thư viện thuật toán tối ưu gồm SGD(Stochastic gradient descent),Adam,RMSprop (Root mean square)
* from keras.preprocessing.image import ImageDataGenerator : Thư viện tạo dữ liệu hình ảnh
* from keras.models import Sequential : Thư viện mô hình tuần tự
* from keras.layers import Dense,Activation,Dropout,LSTM,BatchNormalization : Thư viện xử lý bài toán liên quan đến mạng thần kinh nơ ron
* from keras.layers import Flatten : Thư viện dùng để làm phẳng một đặc trưng
* from keras.layers.convolutional import MaxPooling2D,Conv2D : Thư viện xử lý cho bài toán CNN (tích chập mạng nơron)
* from sklearn.model\_selection import train\_test\_split: Hỗ trợ chia tập dữ liệu để huấn luyện mô hình

### 2.2.2.2 Tạo dữ liệu hình ảnh

Việc quan trọng trong việc xử lý hình ảnh là cân bằng màu sắc ảnh vì mỗi một điểm ảnh (pixel) của một bức ảnh được cấu tạo từ một ma trận. Mỗi bức ảnh sẽ được chia thành ba lớp màu là đỏ, xanh lá, xanh dương và mức độ đậm nhạt của từng màu sẽ khác nhau từ mức độ 0 đến 255. Nếu chúng ta để mặc định màu từ 0 đến 255 thì mô hình sẽ thiên vị cho các màu sắc có giá trị lớn hơn với những sự thay đổi màu sắc nhỏ sẽ không làm ảnh hưởng dến mô hình làm giảm hiệu năng của mô hình. Vì thế điều đầu tiên khi xử lý hình ảnh là đưa về khoảng giá trị 0 đến 255 về 0 đến 1 bằng cách chia tất cả dữ liệu cho 255.

Đối về đề tài thì do là hình ảnh văn bản nền trắng mực đen nên việc chia cho 255 cũng gọi là không thực sự cần thiết.



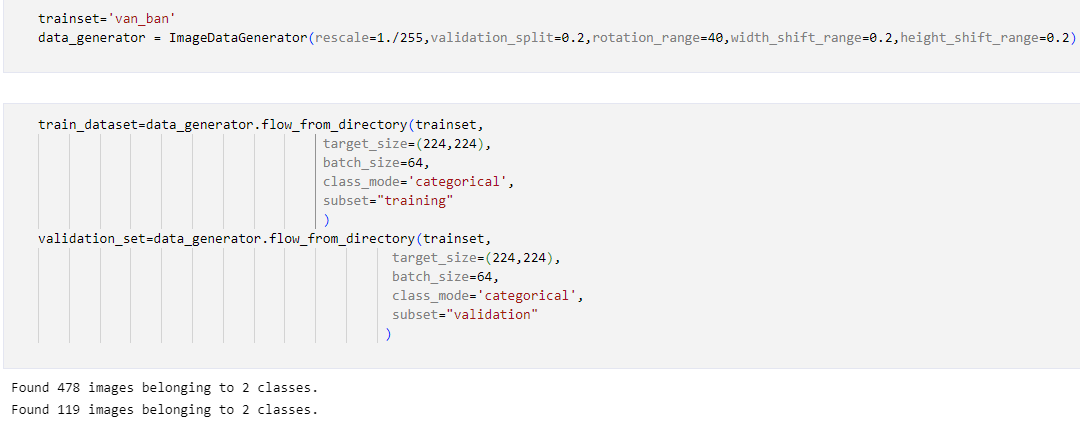
Hình 2. 12 Dạng văn bản dùng để huấn luyện

Tiếp đến ta sẽ tạo độ đa dạng cho hình ảnh bằng cách thay đổi hình ảnh như xoay một góc 40 độ thay dổi kích thước hình ảnh cho lật ngang lật dọc phóng to ảnh,…. Ta sẽ dùng hàm ImageDataGenerator để xử lý việc này.

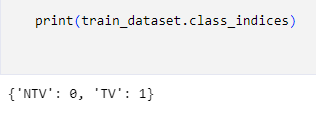
Với các thông số:

* Rescale hình ảnh về khoảng 0 đến 1
* Chia dữ liệu để đánh giá là 0.2
* Xoay ảnh một góc 40 độ
* Dịch ảnh sang trái sang phải 0.2
* Dịch ảnh lên trên xuống dưới 0.2
* Scale size ảnh về 224x224
* Số lượng mẫu trong một lần lấy mẫu là 64
* Ảnh được dùng train cho việc phân loại (categorical)

Và số ảnh sẽ được dùng cho việc train là 478 ảnh và 119 ảnh còn lại sẽ dùng cho việc validation



Hình 2. 13 Tạo sự đa dạng cho hình ảnh với ImageDataGenerator



Hình 2. 14 Số class của mô hình

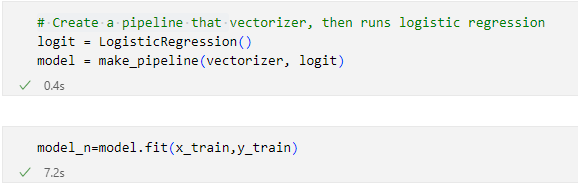
## **2.3 Xây dựng mô hình**

### 2.3.1 Mô hình dự đoán văn bản ngôn ngữ tiếng việt

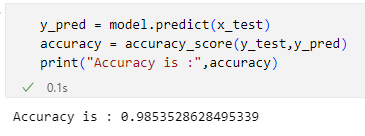
Đối với mô hình này sẽ sử dụng pipeline trong machine learning

Pipeline là một phần không thể thiếu trong một model machine learning vì trong một hệ thống machine learning thường sẽ có rất nhiều thành phần như xử lý dữ liệu, áp dụng thuật toán huấn luyện mô hình, gán nhãn dữ liệu, đánh giá mô hình,v.v. Nếu ta không xây dựng một pipeline hoàn chỉnh với từng phần xử lý tách biệt hoàn toàn thì có nhiều vấn đề có thể xảy ra nên việc tách chúng ra và ghép lại thành một pipeline hoàn chỉnh giúp ta dễ dàng hơn khi huấn luyện mô hình cũng như tìm lỗi cho nó

Sử dụng hàm make\_pipeline() của sklearn để tạo nên một quy trình là xử lý dữ liệu với TfidVectorize là lấy một cụm gồm 3 từ có mức độ quan trọng cao trong từng đoạn văn bản trong tập dữ liệu và biến nó thành một vector đặc trưng và sẽ lấy vector đó là dữ liệu đầu vào để huấn luyện mô hình nhận dạng. Sau đó sẽ áp dụng thuật toán hồi quy Logisic để huấn luyện mô hình



Hình 2. 15 Tạo pipline và bắt đầu việc huấn luyện mô hình



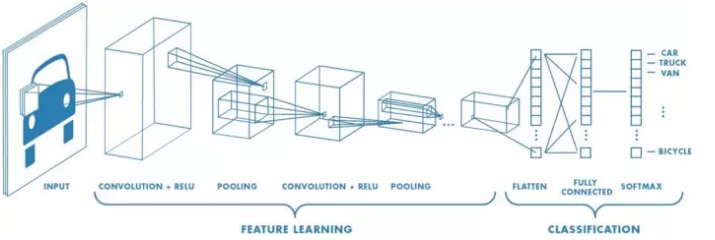
Hình 2. 16 Độ chính xác mô hình sau khi huấn luyện

### 2.3.2 Mô hình nhận diện ngôn ngữ tiếng việt bằng hình ảnh

Mạng nơ-ron nhân tạo (Neural Network - NN) là một mô hình lập trình mô phỏng cách thức hoạt động của mạng nơ-ron thần kinh. Kết hợp với các kĩ thuật học sâu (Deep Learning - DL), mạng nơ-ron nhân tạo đang trở thành một công cụ rất mạnh mẽ mang lại hiệu quả tốt nhất cho nhiều bài toán khó như nhận dạng ảnh, giọng nói hay xử lý ngôn ngữ tự nhiên.

Lịch sử phát triển của mạng nơ-ron nhân tạo bắt đầu khi Warren McCulloch và Walter Pitts đã tạo ra một mô hình tính toán cho mạng nơ-ron dựa trên các thuật toán gọi là logic ngưỡng vào năm 1943.

Về mặt kỹ thuật, trong mô hình học sâu CNN, mô hình ảnh đầu vào sẽ chuyển nó qua một loạt các lớp tích chập với các bộ lọc, sau đó đến lớp Pooling, rồi tiếp theo là các lớp được kết nối đầy đủ (FC — fully connected layers) và cuối cùng áp dụng hàm softmax để phân loại một đối tượng dựa trên giá trị xác suất trong khoản từ 0 đến 1.



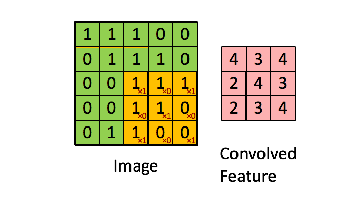
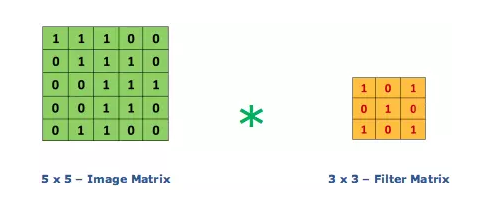
Hình 2. 17 Sơ đồ của việc xử lý ảnh qua mạng nơron CNN

Convolution (lớp tích chập) là lớp đầu tiên trích xuất các đặc tính từ hình ảnh. Tham số lớp này bao gồm một tập hợp các bộ lọc có thể học được. Các bộ lọc đều nhỏ thường có kích cỡ hai chiều đầu tiên khoảng 3x3 hoặc 5x5, .... và có độ sâu bằng với độ sâu của đầu vào đầu vào. Bằng cách trượt dần bộ lọc theo chiều ngang và dọc trên ảnh, chúng thu được một Feature Map chứa các đặc trưng được trích xuất từ trên hình ảnh đầu vào.

Quá trình trượt các bộ lọc thường có các giá trị được quy định bao gồm:

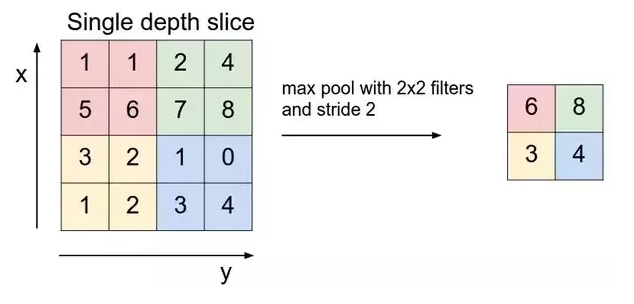
* padding: quy định bộ đệm của bộ lọc hay chính là phần màu xám được thêm vào ảnh
* stride: quy định bước nhảy trong quá trình thực hiện.

Hình 2. 18 Ví dụ về lớp Convolution-relu



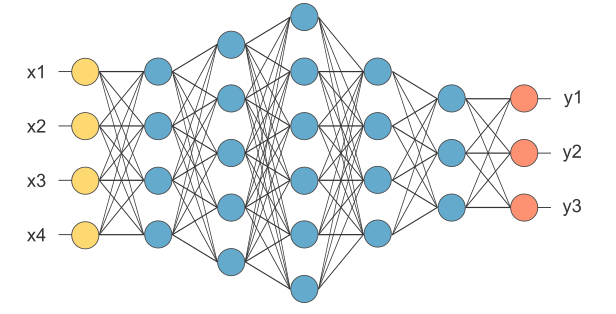
Pooling layer thường được dùng giữa các convolutional layer, để giảm kích thước dữ liệu nhưng vẫn giữ được các thuộc tính quan trọng. Kích thước dữ liệu giảm giúp giảm việc tính toán trong model. Trong quá trình này, quy tắc về stride và padding áp dụng như phép tính convolution trên ảnh.

Ví dụ như hình dưới với kích thước ảnh ban đầu là 4x4 sau khi dùng pooling để giảm kích thước hình ảnh còn 2x2 nhưng vẫn giữ được phẩn tử lớn nhất trong từng map



Hình 2. 19 Ví dụ về lớp Pooling

Fully connected layer: Sau khi ảnh đã truyền qua các lớp tích chập và pooling để xử lý thì lúc này mô hình đã được học tương đối các đặc điểm của ảnh thì output của layer cuối cùng sẽ được làm phẳng (Flatten) từ một ma trận thành một vector để đưa vào một lớp được kết nối như một mạng nơron, cuối cùng dùng softmax hoặc sigmoid để phân loại đầu ra.



Hình 2. 20 Thiết lập mạng thần kinh nơ ron

Thực hiện bài toán phân loại văn bản với CNN trong Visual Studio Code:

Với ảnh đầu vào là 224x224x3 ta sẽ sử dụng 64 kernel với bộ lọc ma trận là 3x3 để trượt qua từng ảnh gốc làm nổi bật lên các đặc trưng đồng thời sử dụng thêm các lớp padding để giữ lại kích thước gốc ban đầu của ảnh sau đó ta sẽ thêm một activision (relu) phi tuyến để có thể sấp sỉ được các phân bố phức tạp và không làm thay đổi kích thước ban đầu của ảnh.Tiếp đến ta thêm các lớp MaxPooling 2D, lớp này dùng để chọn ra những đặc trưng nổi bật làm tăng độ chính xác cũng như làm giảm kích thước ảnh.

Tiếp đến là lớp DropOut để qua một 1 nút mạng trong quá trình đạo tạo một cách ngẫu nhiên giúp việc học trở nên mạnh mẽ, tăng gấp đôi số eposch cần để hội tụ,…

Với các Block tiếp theo sẽ tương tự nhưng tăng lượng kernel lên là 128 và 384.

Ở Block cuối cùng để tiến hành phân loại đầu vào thì ta cần biến output của ảnh lúc này là một ma trận về một vector để có thể làm điều này từ ta sẽ cho qua một lớp Flatten. Sau đó thu lại số lớp phù hợp cho số lượng cần dự đoán là 2 tương ứng với hai lớp tiếng việt và không phải tiếng việt.Dùng softmax để tiến hành phân loại đầu ra.



Hình 2. 21 Mạng nơ ron được xây dựng cho mô hình phân loại văn bản tiếng việt

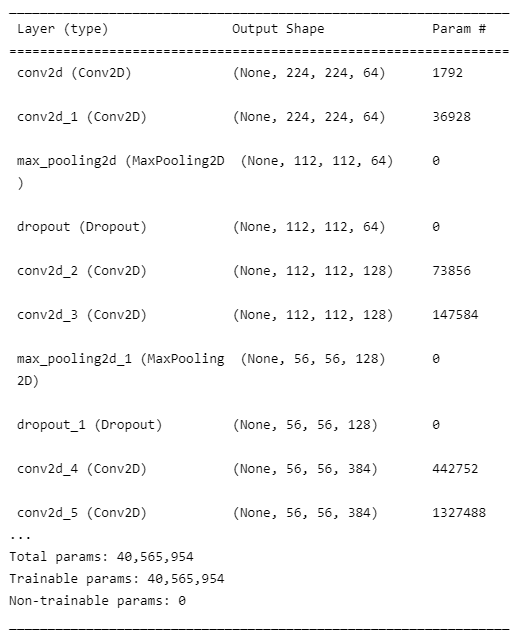
Ta dùng hàm summary() của tensorflow để cho chúng ta thấy mạng nơron ta tạo gồm những thành phần gì:

Tên và loại các lớp trong model

Kích thước đầu ra của từng lớp

Tham số trọng lượng của từng lớp

Số lượng phần tử huấn luyện được và không huấn luyện được trong mô hình



Hình 2. 22 Bảng tổng kết của mô hình mạng thần kinh nơ ron

Tiếp đến chính là quá trình xây dựng cho mô hình các hàm loss và thuật toán tối ưu

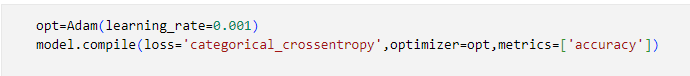
Hàm mất mát là một phương pháp đánh giá độ hiệu quả của thuật toán “học” cho mô hình trên tập dữ liệu được sử dụng. Hàm mất mát trả về một số thực không âm thể hiện sự chênh lệch giữa hai đại lượng: a, nhãn được dự đoán và y, nhãn đúng. Hàm mất mát, bản thân chính là một cơ chế thưởng-phạt, mô hình sẽ phải đóng phạt mỗi lần dự đoán sai và mức phạt tỉ lệ thuận với độ lớn sai sót. Trong mọi bài toán học có giám sát, mục tiêu luôn bao gồm giảm tổng mức phạt phải đóng. Trong trường hợp lý tưởng a = y, loss function sẽ trả về giá trị cực tiểu bằng 0 . Hai hàm mất mát thường xuyên được sử dụng trong mạng nơ-ron: MSE (Mean Squared Error) và Cross Entropy.

Trong thuật toán học máy nói chung và kĩ thuật học sâu nói riêng, thuật toán tối ưu hóa là một khâu quan trọng không thể thiếu. Quá trình tối ưu hóa thực hiện xác định hàm mất mát (loss function) và sau đó tối thiểu hóa hàm trên bằng cách sử dụng hàm tối ưu. Cụ thể, thông qua việc cập nhật các tham số của mô hình (w, b) và đánh giá lại hàm mất mát với một tỉ lệ học (learning rate) xác định, quá trình tối ưu giúp mô hình tương thích tốt hơn với tập dữ liệu được đào tạo.

Trong mô hình sẽ sử dụng thuật toán Adam (Adaptive Moment Estimation) là thuật toán tối ưu. Adam được xtôinhư là sự kết hợp của RMSprop và Stochastic Gradient Descent với động lượng. Adam là một phương pháp tỉ lệ học thích ứng, nó tính toán tỉ lệ học tập cá nhân cho các tham số khác nhau. Adam sử dụng ước tính của khoảng thời gian thứ nhất và thứ hai của độ dốc để điều chỉnh tỉ lệ học cho từng trọng số của mạng nơ-ron.

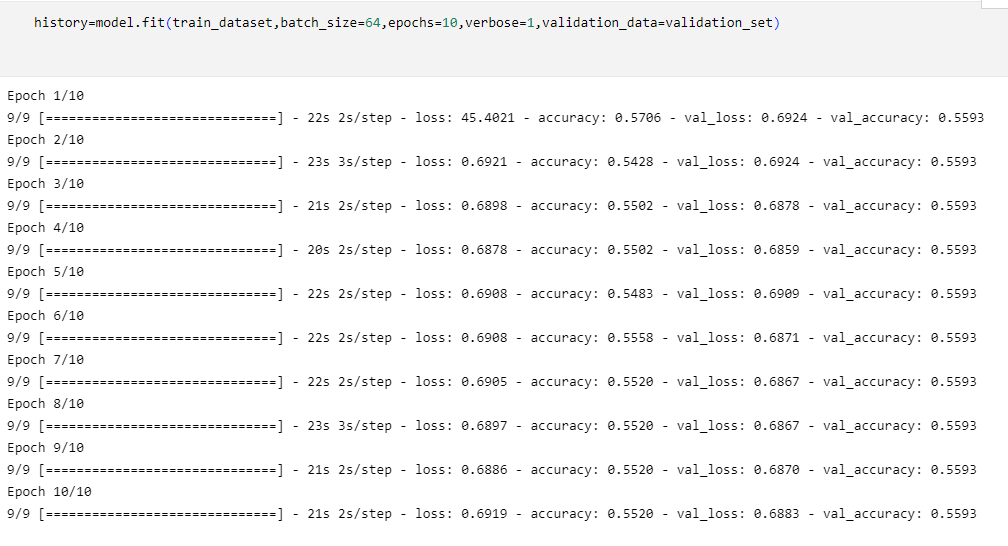
Với tỉ lệ học (learning\_rate) sẽ là 0.001 để tối ưu trọng số và tham số ngưỡng – bias trong quá trình học để giúp mô hình ổn định.

Hàm mất mát sẽ là hàm được thường xuyên trong mạng nơ ron cho bài toán phân loại là categorical\_crossentropy với thước đo đánh giá là accuracy

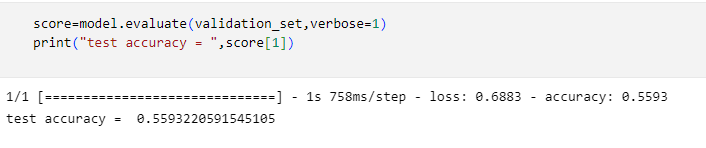


Hình 2. 23 Thuật toán tối ưu và hàm mất mát cho mô hình

Bước cuối cùng chính là huấn luyện mô hình với dữ liệu là đầu vào là hình ảnh văn bản đã được tạo ra từ ImageDataGenerator ở phần trước với batch\_size (số lần lấy mẫu trong một lần học) là 64 với số lần học (espoch) là 10 .



Hình 2. 24 Huấn luyện mô hình

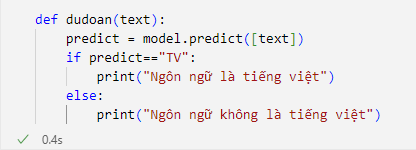


Hình 2. 25 Độ chính xác sau 10 lần học

# **CHƯƠNG 3: MÔ PHỎNG VÀ ĐÁNH GIÁ CHẤT LƯỢNG MÔ HÌNH**

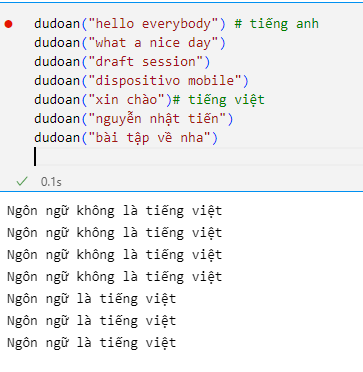
* 1. **Mô hình dự đoán văn bản ngôn ngữ tiếng việt**

Ta sẽ tạo một function mang tên là dudoan với tham số text để dự đoán văn bản mà người dùng sẽ nhập vào



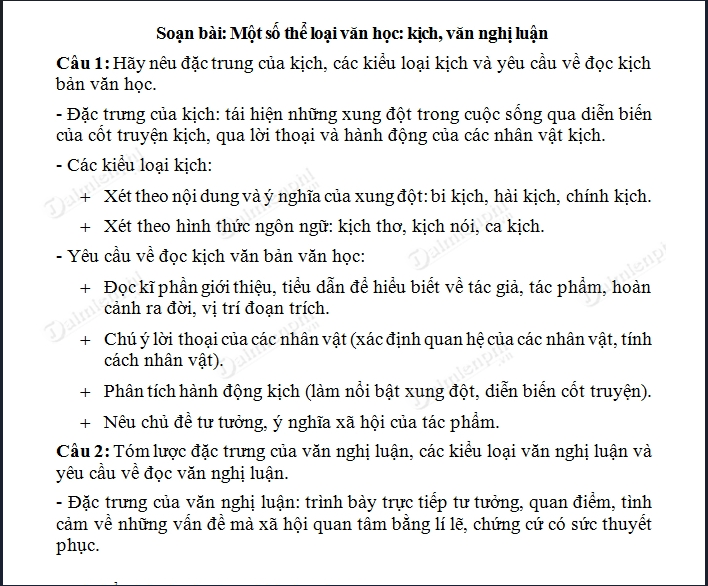
Hình 3. 1 Hàm dự đoán ngôn ngữ qua việc nhập

Tiến hành dự đoán một loạt input văn bản



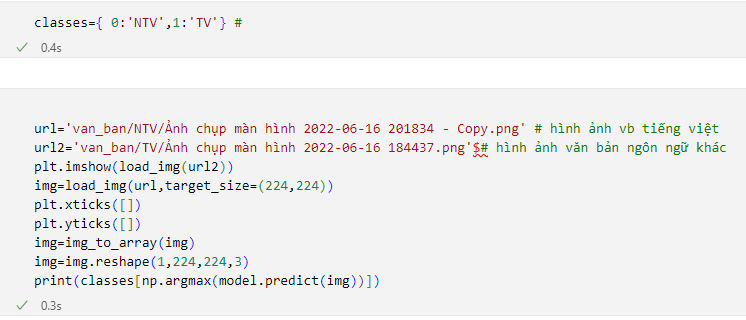
Hình 3. 2 Kết quả dự đoán của hàm

* 1. **Mô hình nhận diện văn bản tiếng việt qua hình ảnh**





Hình 3. 3 Hình ảnh dùng cho việc dự đoán

****

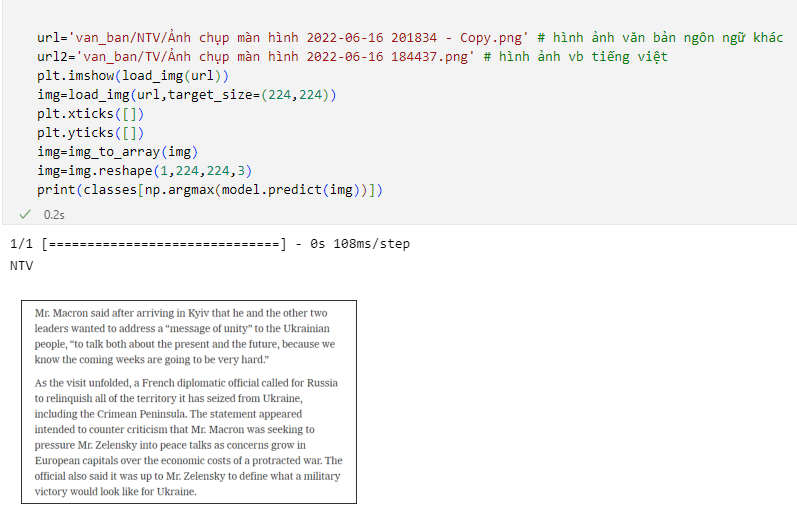
Hình 3. 4 Code cho việc dự đoán

Đầu tiên ta sẽ tạo mapping gồm 2 class là NTV và TV để phục vụ cho việc gắn nhãn dự đoán.

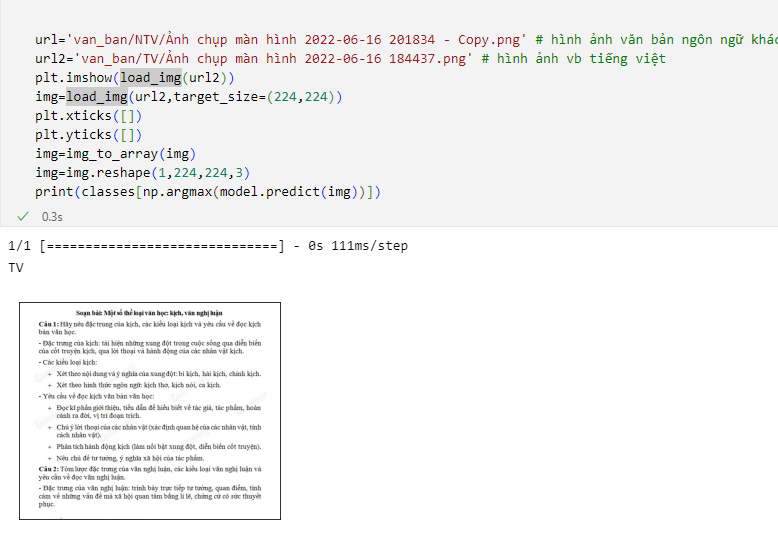
Sau đó chúng ta cho hiện ảnh mà ta chọn để dự đoán với scale với kích thước ảnh thành 224x224

Với việc ảnh là một ma trận điểm ảnh thì sau khi scale lai kích thước 224x 224 thì ta sẽ chuyển thành một mảng 224 hàng 224 cột 3 mã màu

Cuối cùng là việc in là kết quả dự đoán kết hợp giữa mapping mà ta đã tạo với function argmax ( trả về chỉ số của phần tử lớn nhất trong một trục cụ thể)



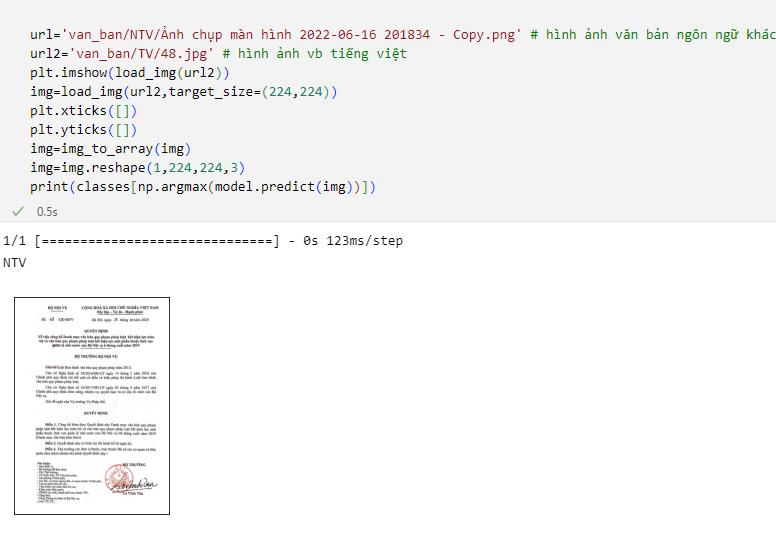
Hình 3. 5 Dự đoán với hình ảnh vân bản không phải tiếng việt



Hình 3. 6 Dự đoán với hình ảnh vân bản tiếng việt

Tuy nhiên mô hình vẫn còn dự đoán sai rất nhiều vì độ chính xác còn thấp cụ thể là 56%



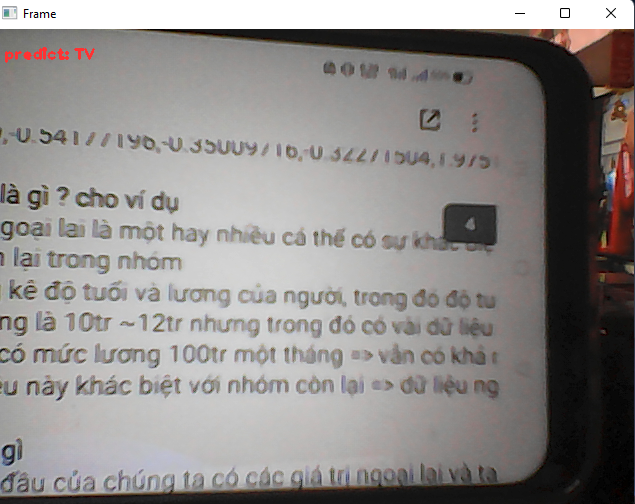


Hình 3. 7 Những hình ảnh dự đoán sai của mô hình phân loại

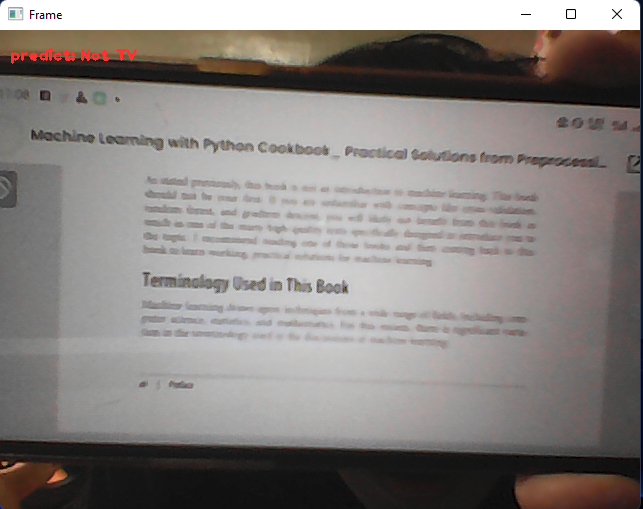
* 1. **Real time detection**



HÌnh 3. 8 Thử nghiệm camera của laptop



HÌnh 3. 9 Với hình ảnh văn bản tiếng việt



HÌnh 3. 10 Với hình ảnh văn bản ngôn ngữ khác

## **3.4 Đánh giá mô hình**

* Đối với mô hình nhận dạng ngôn ngữ tiếng việt qua văn bản thì hoạt động cực kì tốt với 98% độ chính xác nên ta không cần phải cải thiện thêm gì nhiều
* Còn với mô hình nhân dạng ngôn ngữ tiếng việt qua hình ảnh văn bản thì chưa tốt với độ chính xác 56% cho nên mô hình này chưa có thể đtôiđi áp dụng thực tế mà chỉ dùng cho việc thí nghiệm nhỏ lẻ bởi vì cần cải thiện hơn về dữ liệu, tiền xử lý dữ liệu và thuật toán phù hợp với bài toán nhận dạng này
* Việc xác nhận realtime tương đối khó khăn vì tình trạng máy móc giật lag thì chạy chương trình cộng với việc mở cam

# **CHƯƠNG 4 : THIẾT KẾ GIAO DIỆN WEB**

Sau khi huấn luyện được hai mô hình thì tôi bắt đầu đưa mô hình với một website cụ thể để người dùng có thể trực tiếp tương tác với nó.

Công cụ hỗ trợ để tạo website tương tác cho đề tài sẽ là Visual Studio Code, ngôn ngữ HTML và môi trường Flask

## **4.1 HTML**

HTML là viết tắt của từ Hypertext Markup Language một ngôn ngữ được thiết kế ra để tạo nên một trang web trên World Wide Web (www) với đầy đủ các phần như heading, titles,data,….. phục vụ cho việc tạo nên một giao diện tương tác cụ thể đẹp mắt cho người dùng trải nghiệm

Tuy nhiên HTML cũng chỉ là giao diện thiết kế còn về việc tương tác thông tin, tìm kiếm,v.v thì sẽ cần kết hợp thêm những ngôn ngữ kịch bản như java script,php, python,..

Và ta sẽ dùng python để tương tác với giao diện thiết kế trong đề tài lần này

## **4.2 Flask**

Flask là một web frameworks, nó thuộc loại micro-framework được xây dựng bằng ngôn ngữ lập trình Python. Flask cho phép bạn xây dựng các ứng dụng web từ đơn giản tới phức tạp. Nó có thể xây dựng các api nhỏ, ứng dụng web chẳng hạn như các trang web, blog, trang wiki hoặc một website dựa theo thời gian hay thậm chí là một trang web thương mại. Flask cung cấp cho bạn công cụ, các thư viện và các công nghệ hỗ trợ bạn làm những công việc trên.

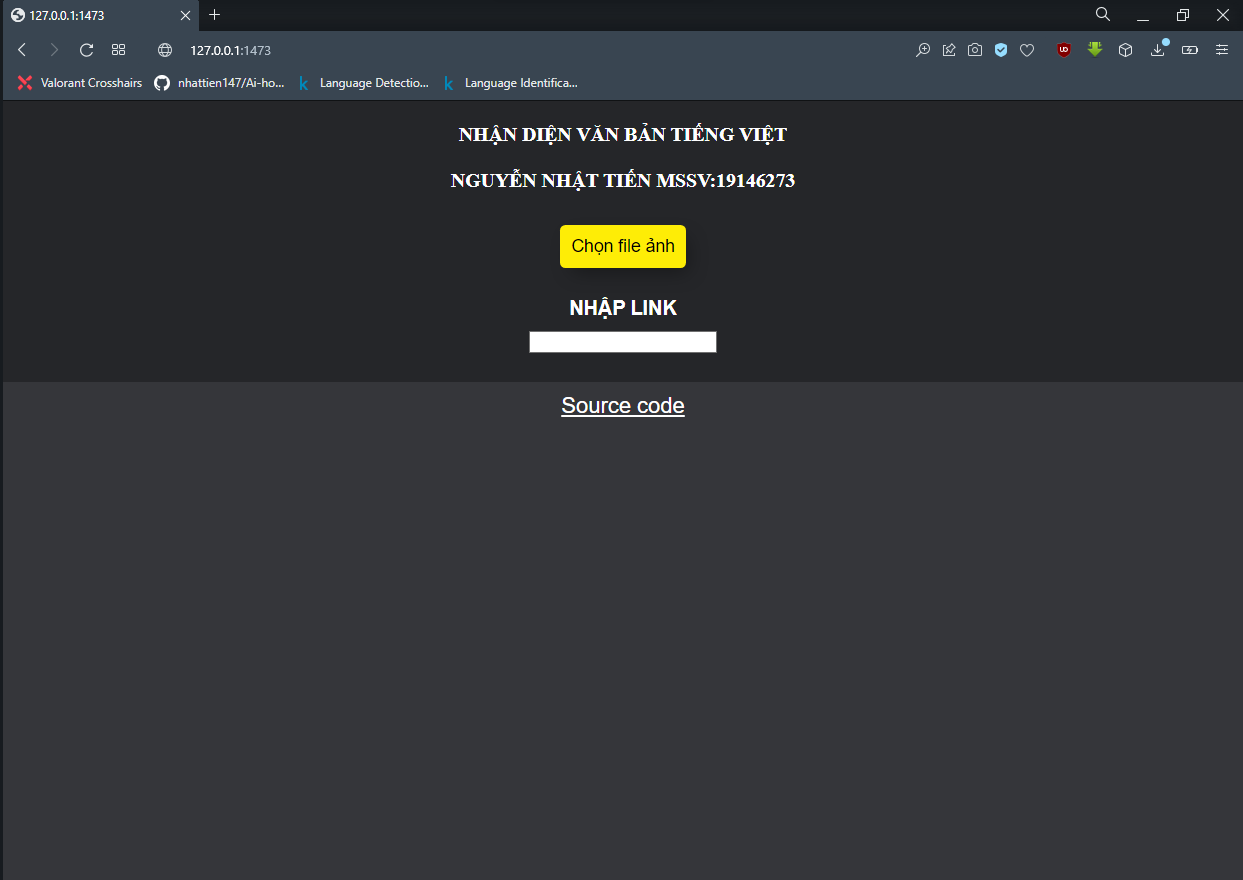
Flask là một micro-framework. Điều này có nghĩa Flask là một môi trường độc lập, ít sử dụng các thư viện khác bên ngoài. Do vậy, Flask có ưu điểm là nhẹ, có rất ít lỗi do ít bị phụ thuộc cũng như dễ dàng phát hiện và xử lý các lỗi bảo mật.



Hình 4. 1 Flask Framework

## **4.3 Giao diện Website**

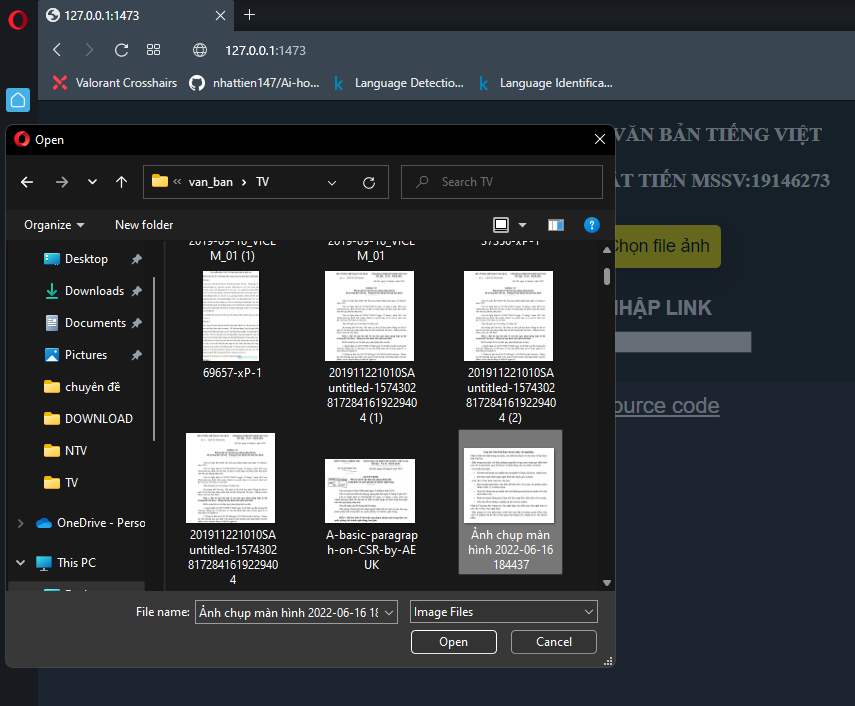
Phần code thiết kế HTML và code nhận gửi dữ liệu sẽ được để trong github



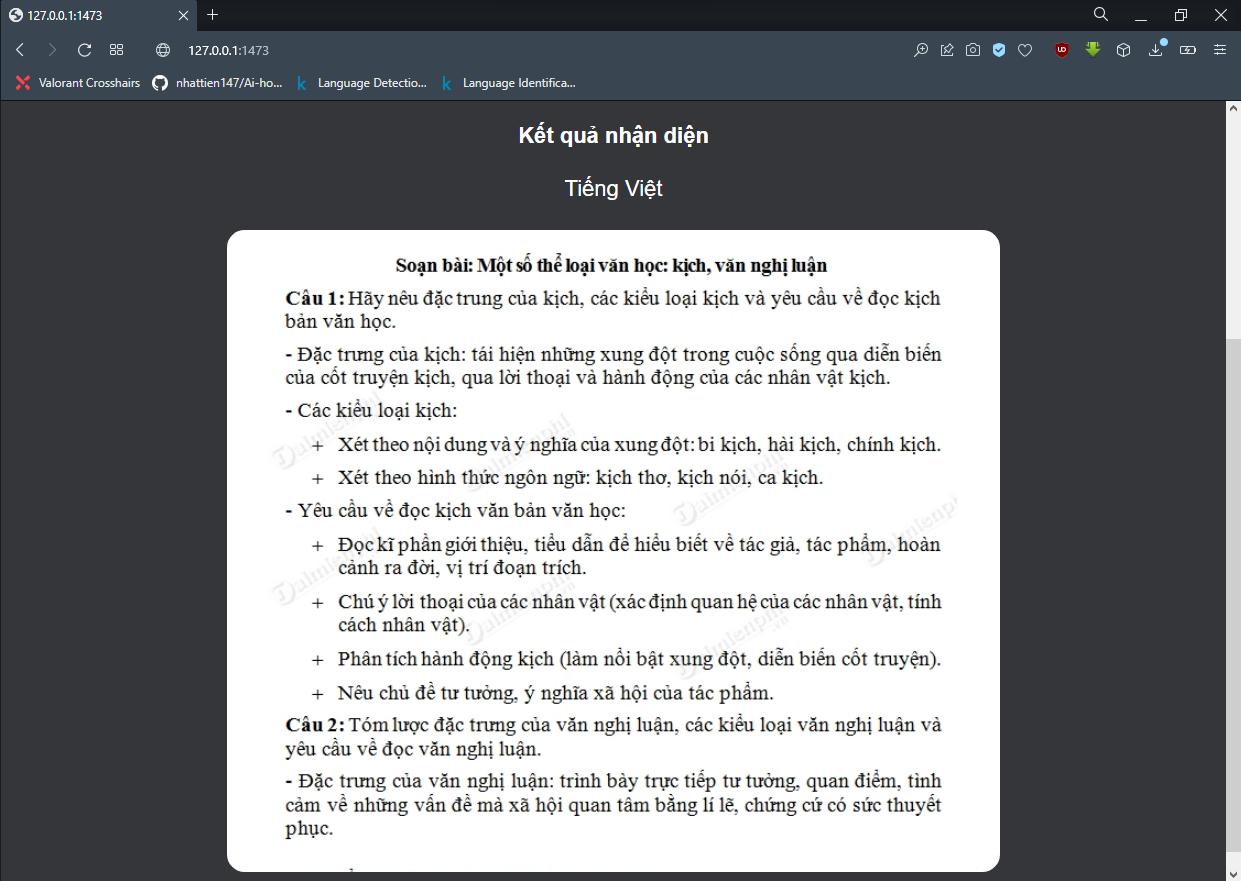
Hình 4. 2 Giao diện cơ bản của website

Tương tác giữa website của người dùng cũng rất đơn giản

Người dùng có thể chọn một tấm ảnh văn bản ngôn ngữ có trong máy tính của mình để upload lên và sau đó trang sẽ đưa ra dự đoán.

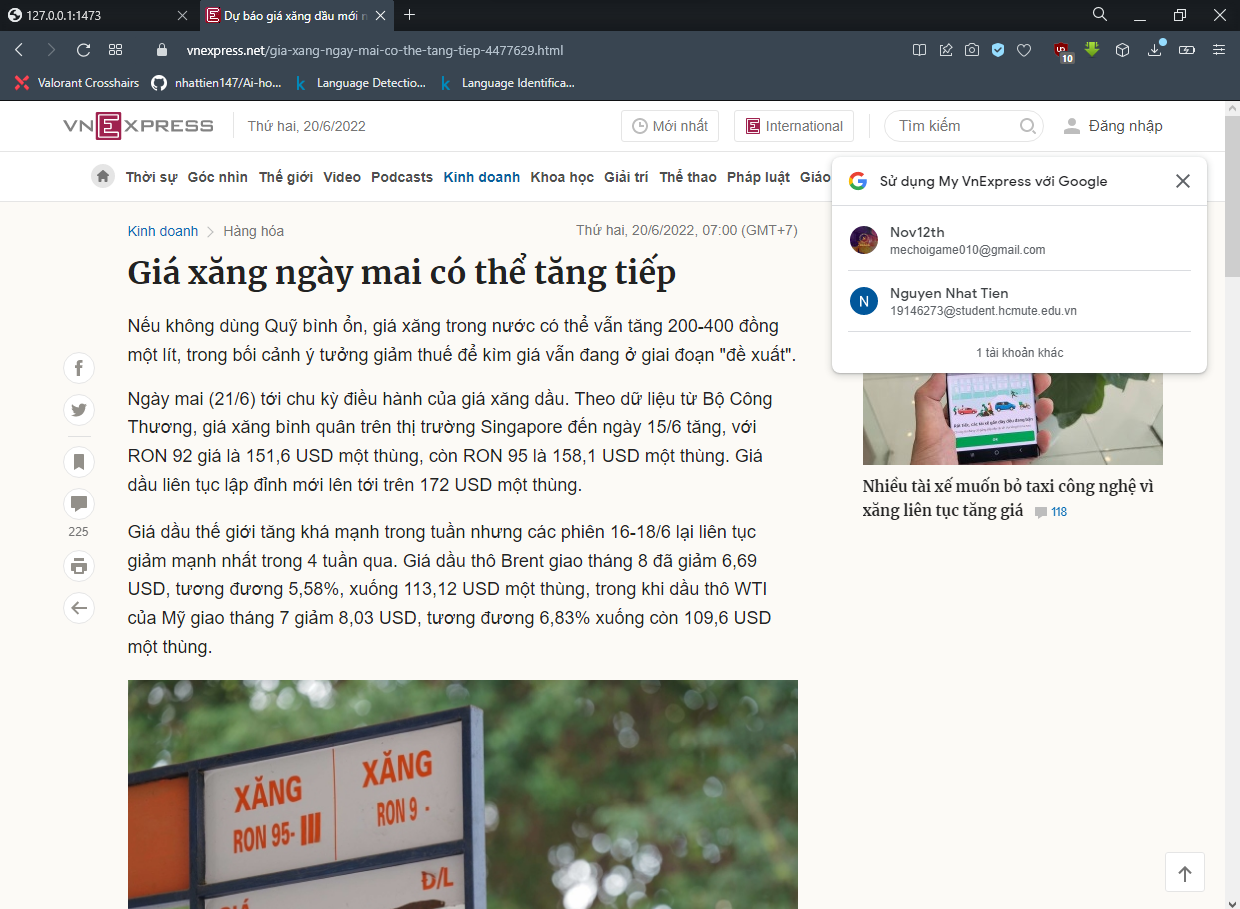


Hình 4. 3 Lựa chọn hình ảnh từ máy

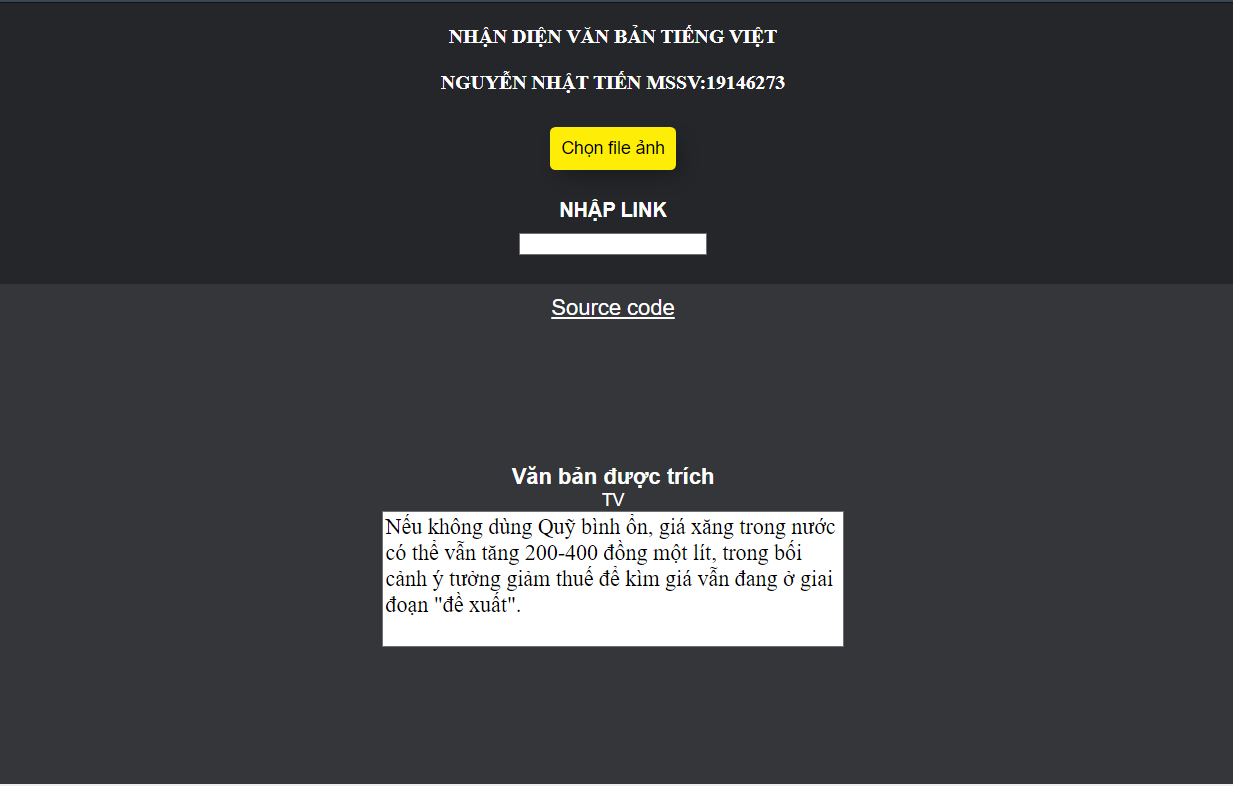


Hình 4. 4 Tiến hành dự đoán

Ngoài việc chọn hình ảnh thì người dùng có thể nhập link của một bài báo bất kì nào đó từ đó website sẽ trích một đoạn văn bản trong bài báo rồi đự doán và đưa ra kết quả



Hình 4. 5 Bài báo được chọn để trích văn bản



Hình 4. 6 Kết quả dự đoán sau khi add link bài báo

# **CHƯƠNG 5: KẾT LUẬN VÀ ĐỊNH HƯỚNG**

* 1. **Kết luận**

Sau một thời gian thực hiện thì tôiđã cơ bản dựng lên được hai mô hình dự đoán ngôn ngữ tiếng việt thông qua văn bản và hình ảnh văn bản

Những điều đã làm được:

* Áp dụng dược những kiến thức đã được học trên lớp trí tuệ nhân tạo kết hợp với internet để thực hiện đề tài
* Tìm hiểu được thêm các ngôn ngữ khác như HTML, python
* Tiếp xúc được với môi trường Flask framework
* Huấn luyện thành công tương đối hai mô hình nhận dạng văn bản ngôn ngữ tiếng việt và nhận dạng văn bản hình ảnh ngôn ngữ tiếng việt
* Thiết kế giao diện website tương tác người dùng cơ bản

Những điều chưa làm được:

* Mô hình nhận dạng ngôn ngữ tiếng việt bằng hình ảnh văn bản còn khá thấp chỉ đạt được 56%
* Dữ liệu còn ít 607 tấm với lại chưa đuọc chọn lọc kĩ càng
* Chưa làm được qua app
  1. **Định hướng**

Sẽ cải thiện mô hình bằng cách cải thiện dữ liệu đầu vào với nhiều hình ảnh và chọn lọc kĩ càng nhất có thể ngoài ra sẽ tính toán thiết kế lại mạng nơron để tối ưu hoá việc học cũng như dự đoán đạt được độ chính xác cao nhất

Ngoài ra sẽ tìm hiểu nhiều hơn nữa các tài liệu về trí tuệ nhận tạo các thuật toán để có sử dụng thuật toán tốt nhât cho mô hình

# **TÀI LIỆU THAM KHẢO**

**[1]** <https://viblo.asia/p/thuat-toan-toi-uu-adam-aWj53k8Q56m>

**[2]** <https://viblo.asia/p/logistic-regression-bai-toan-co-ban-trong-machine-learning-924lJ4rzKPM>

**[3]** <https://scikit-learn.org/stable/index.html>

**[4]** <https://viblo.asia/p/deep-learning-tim-hieu-ve-mang-tich-chap-cnn-maGK73bOKj2>

**[5]** <https://miai.vn>

**[6]** <https://www.tensorflow.org/resources/learn-ml?gclid=CjwKCAjwtcCVBhA0EiwAT1fY79io2mPKKn8ZpZA3pYua3eunulAVd8UoFA5ctRPY6ShETezMglRnhBoCkIwQAvD_BwE>

**[7]** <https://flask.palletsprojects.com/en/2.1.x/>

**[8]** <https://www.w3schools.com>