**BỘ GIÁO DỤC VÀ ĐÀO TẠO**

**ĐẠI HỌC KINH TẾ TP. HỒ CHÍ MINH**



**TÊN MÔN HỌC: TRÍ TUỆ NHÂN TẠO**

**CHỦ ĐỀ:**

**NHẬN DIỆN NGÔN NGỮ KÝ HIỆU(SIGN LANGUAGE DETECTION)**

**TÊN GIẢNG VIÊN: NGUYỄN TRƯỜNG THỊNH**

**MÃ HỌC PHẦN: 24D1INT54702201**

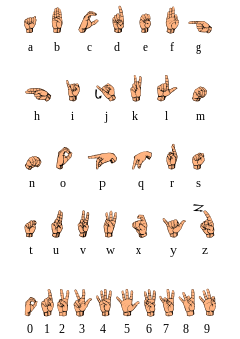
**NGƯỜI THỰC HIỆN:ĐỖ DƯƠNG TRÍ THUẬN\_MSSV: 31231024253**

TP. Hồ Chí Minh – 2024

1. Phần giới thiệu

-Trong thế kỷ 21, trí tuệ nhân tạo (AI) đã trở thành một trong những lĩnh vực công nghệ quan trọng nhất và được chú ý nhất trên toàn thế giới. AI nghiên cứu và phát triển các hệ thống và thuật toán có khả năng tự học và tự thích nghi từ dữ liệu, từ đó giải quyết các vấn đề phức tạp mà trước đây chỉ có con người mới có thể làm được.Các ứng dụng của AI rất đa dạng, từ tự lái xe, dự đoán thị trường tài chính, đến chẩn đoán y tế và dịch ngôn ngữ. Trong công nghiệp, AI cũng được áp dụng để tối ưu hóa quy trình sản xuất và giảm thiểu sự cố.Có nhiều phương pháp và kỹ thuật khác nhau trong lĩnh vực AI, bao gồm học máy (machine learning), học sâu (deep learning), xử lý ngôn ngữ tự nhiên (NLP), thị giác máy tính (computer vision), và nhiều hơn nữa.Mặc dù AI mang lại nhiều tiềm năng lớn cho sự tiến bộ của con người, nhưng cũng đi kèm với những thách thức, bao gồm đạo đức và an ninh thông tin. Do đó, việc nghiên cứu và triển khai AI cần được tiếp tục theo dõi và quản lý một cách cẩn thận và có trách nhiệm.

-Bài báo cáo này nhấn mạnh vào ứng dụng của trí tuệ nhân tạo (AI) trong việc phân loại và nhận biết ngôn ngữ ký hiệu (American Sign Language - ASL). Việc sử dụng kỹ thuật Convolutional Neural Networks (CNN) để huấn luyện AI nhận biết các ngôn ngữ ký hiệu là một bước tiến quan trọng trong việc tạo ra các hệ thống hỗ trợ cho người khiếm thính hoặc người nói ngôn ngữ ký hiệu.



Bảng ngôn ngữ ký hiệu (nguồn Wikipedia)

Cụ thể, CNN là một loại mạng nơ-ron nhân tạo được thiết kế để xử lý dữ liệu có cấu trúc lưới như hình ảnh. Với dữ liệu hình ảnh về ngôn ngữ ký hiệu, CNN có thể học các đặc điểm đặc trưng của các biểu hiện ngôn ngữ ký hiệu và sử dụng chúng để phân loại các biểu hiện khác nhau.

Quy trình có thể bao gồm các bước sau:

1. Thu thập dữ liệu: Tập hợp các hình ảnh về ngôn ngữ ký hiệu từ nguồn đáng tin cậy và đa dạng để sử dụng cho việc huấn luyện và kiểm tra mô hình AI.

2. Tiền xử lý dữ liệu: Chuẩn hóa và xử lý dữ liệu hình ảnh để chuẩn bị cho việc đào tạo mô hình. Điều này có thể bao gồm chuyển đổi hình ảnh thành định dạng phù hợp và thực hiện các phương pháp tiền xử lý như cắt tỉa, phân đoạn và cân bằng màu sắc.

3. Xây dựng và huấn luyện mô hình CNN: Xây dựng một mạng CNN có thể nhận diện và phân loại các biểu hiện ngôn ngữ ký hiệu từ dữ liệu hình ảnh. Sau đó, huấn luyện mô hình bằng cách sử dụng tập dữ liệu đã được chuẩn bị.

4. Kiểm tra và điều chỉnh: Kiểm tra hiệu suất của mô hình trên tập dữ liệu kiểm tra để đảm bảo rằng nó có khả năng nhận diện các biểu hiện ngôn ngữ ký hiệu một cách chính xác. Nếu cần, điều chỉnh mô hình và lặp lại quá trình này cho đến khi đạt được hiệu suất mong muốn.

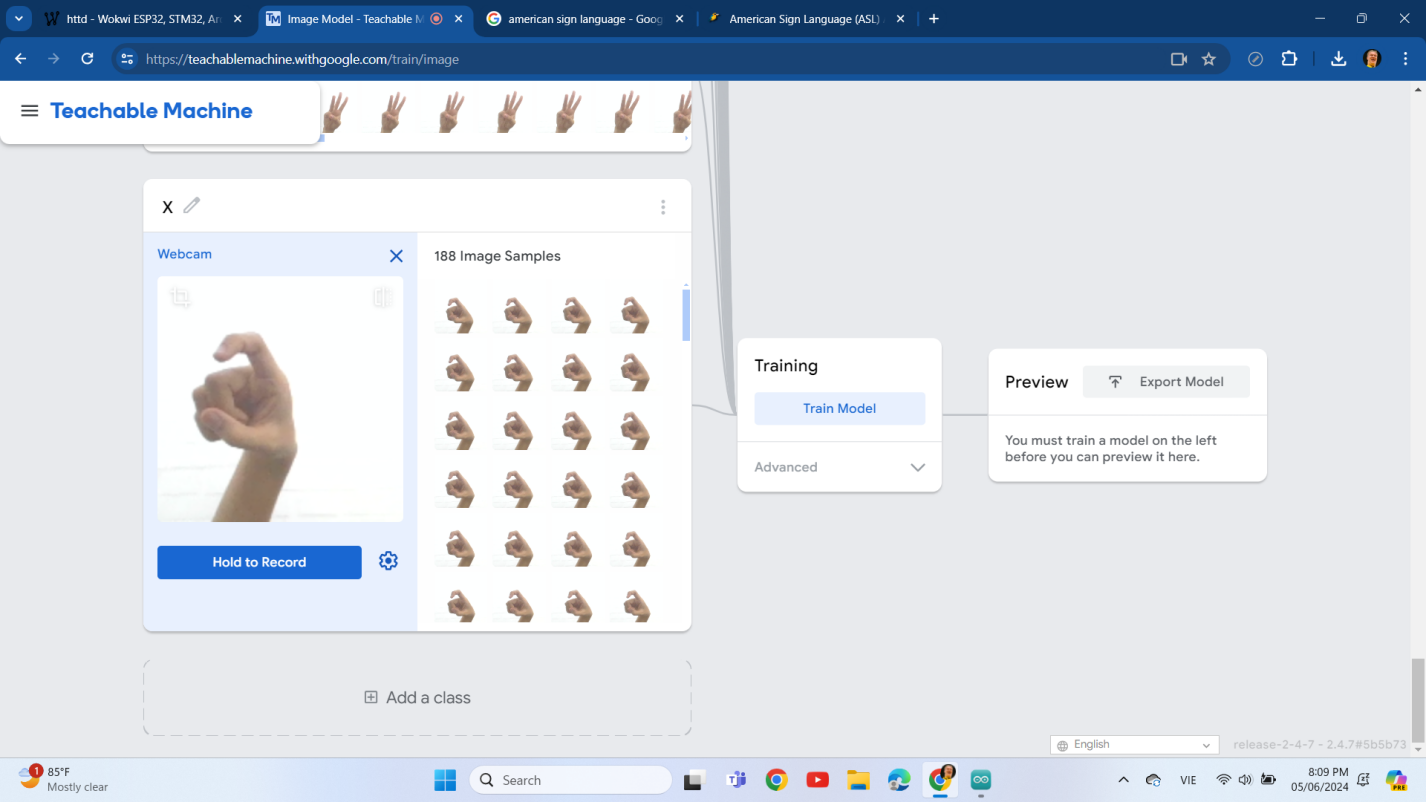
5. Triển khai và ứng dụng: Sau khi mô hình đã được huấn luyện và kiểm tra, triển khai nó vào các ứng dụng thực tế để hỗ trợ người dùng khiếm thính trong việc giao tiếp và tương tác.

Bằng cách này, việc sử dụng AI và CNN trong việc nhận biết ngôn ngữ ký hiệu có thể cải thiện sự tiếp cận và hỗ trợ cho cộng đồng khiếm thính, tạo ra một môi trường giao tiếp hòa nhập và công bằng hơn.

1. Các bước thực hiện

Bước 1: Thu thập dữ liệu

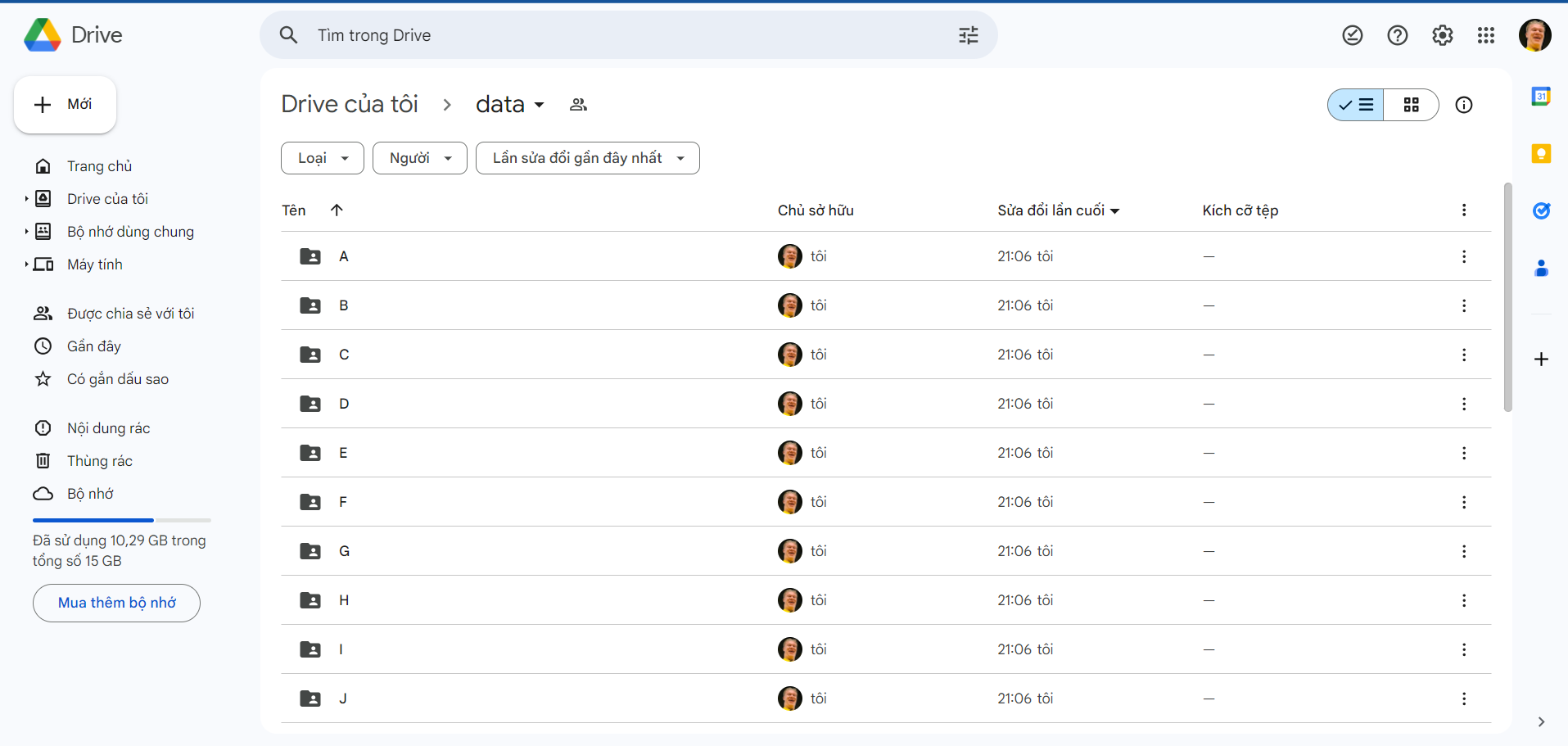
Bước đầu tiên trong quá trình thu thập dữ liệu ký hiệu tay bằng Teachable Machine là lựa chọn các biểu hiện ngôn ngữ ký hiệu mà muốn mô hình nhận biết và phân loại. Có thể chọn từ một loạt các ký hiệu tay phổ biến hoặc tùy chỉnh các biểu hiện riêng của mình. Sau đó, cần thu thập dữ liệu hình ảnh của các biểu hiện đó từ nhiều góc độ và điều kiện ánh sáng khác nhau để đảm bảo rằng mô hình sẽ hoạt động tốt trong các tình huống thực tế, các hình ảnh thu thập bằng Teachable Machine có kích thước 300x300 pixel và mỗi một chữ cái sẽ có khoảng hơn 100 bức ảnh liên quan. Đây là một quá trình quan trọng và tốn thời gian, nhưng đảm bảo rằng dữ liệu thu thập là đa dạng và đại diện cho các biểu hiện muốn mô hình nhận biết. Trong Teachable Machine, có thể sử dụng webcam hoặc tải lên các tập tin hình ảnh đã được thu thập trước đó. Sau khi thu thập đủ dữ liệu, cần phân loại chúng thành các nhóm tương ứng với các biểu hiện khác nhau. Cuối cùng, khi dữ liệu đã được thu thập và phân loại, cần kiểm tra và làm sạch dữ liệu để loại bỏ các hình ảnh không chính xác hoặc nhiễu. Điều này sẽ giúp mô hình học được từ dữ liệu chất lượng cao và cải thiện khả năng nhận biết và phân loại các biểu hiện ngôn ngữ ký hiệu.



Thu thập dữ liệu hình ảnh bằng web Teachable Machine

Bước 2:Tải dữ liệu lên Google Drive

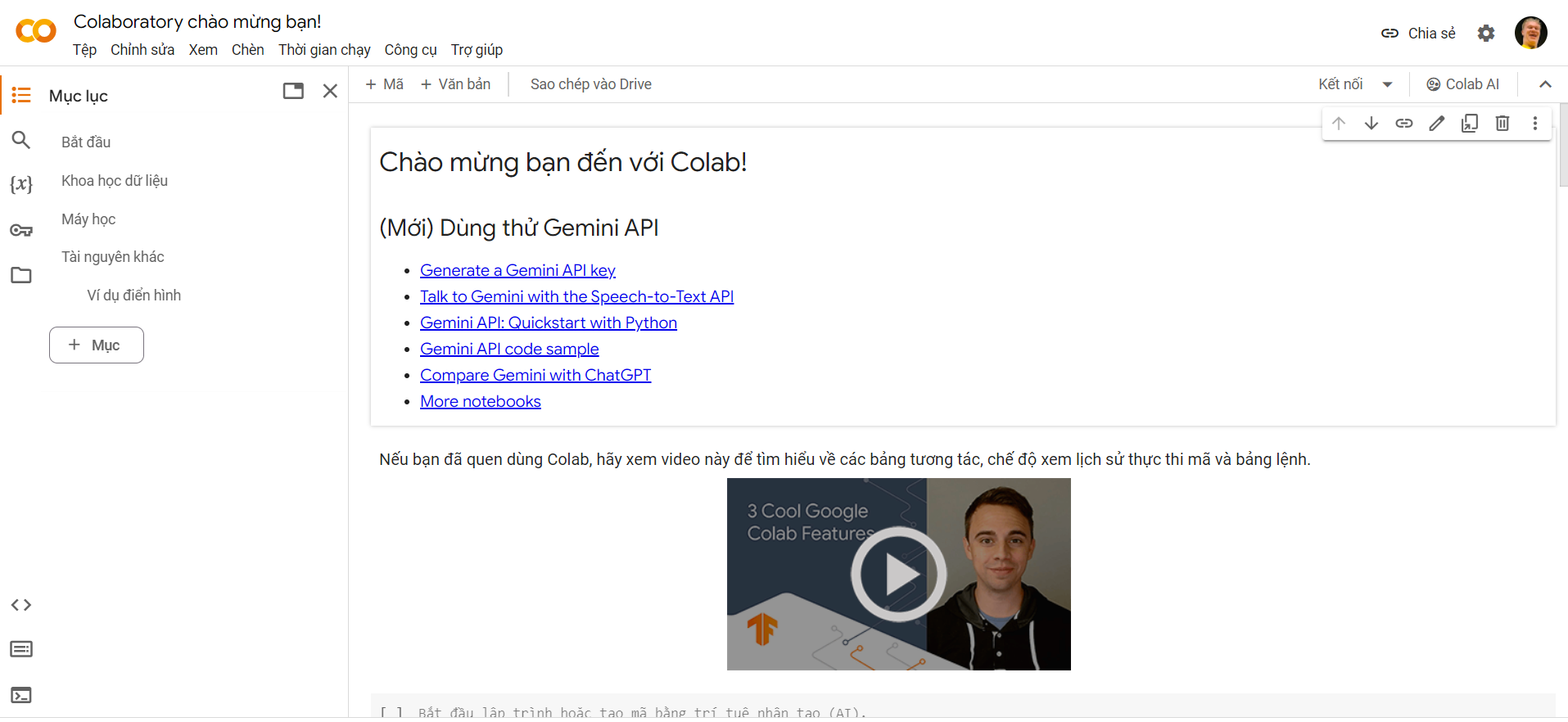
Sau khi hoàn thành quá trình thu thập dữ liệu trên Teachable Machine, việc tiếp theo là tải dữ liệu lên Google Drive để lưu trữ và sử dụng cho việc huấn luyện mô hình hoặc các mục đích khác. Đầu tiên, sau khi đã thu thập và phân loại dữ liệu trên Teachable Machine,cần tải dữ liệu xuống máy tính của mình dưới dạng các tệp hình ảnh. Sau đó, sẽ tạo một thư mục trên Google Drive để lưu trữ các tệp dữ liệu này. Khi thư mục đã được tạo, có thể kéo và thả các tệp hình ảnh đã tải xuống từ Teachable Machine vào thư mục trên Google Drive. Đảm bảo đã tổ chức các tệp theo cách phù hợp và gắn nhãn chúng để dễ dàng quản lý và truy cập sau này.Sau khi dữ liệu đã được tải lên Google Drive, có thể chia sẻ thư mục dữ liệu đó cho dự án AI. Điều này giúp tạo điều kiện thuận lợi cho việc làm việc nhóm và sử dụng dữ liệu một cách hiệu quả trong các quy trình tiếp theo của dự án.



Tải hình ảnh thu thập lên Google Drive và phân loại

Bước 3:Huấn luyện AI

Việc huấn luyện AI trên Google Colab là điều cần thiết cho việc huấn luyện. Google Colab cung cấp một môi trường miễn phí cho việc chạy mã Python trên các máy chủ có GPU, giúp tăng tốc quá trình huấn luyện mô hình AI. Việc sử dụng Colab rất đơn giản, chỉ cần truy cập trình duyệt web và có thể bắt đầu làm việc ngay lập tức mà không cần cài đặt bất kỳ phần mềm nào. Một trong những lợi ích lớn nhất của việc sử dụng Colab là sức mạnh của GPU miễn phí. Điều này cho phép huấn luyện các mô hình phức tạp nhanh chóng và hiệu quả hơn so với việc sử dụng CPU thông thường. Cũng như có thể tận dụng các thư viện Python phổ biến như TensorFlow, PyTorch và Keras mà không cần lo lắng về việc cài đặt chúng trên máy tính cá nhân. Tuy nhiên, có một số hạn chế khi sử dụng Colab, bao gồm giới hạn về thời gian chạy và dung lượng lưu trữ.

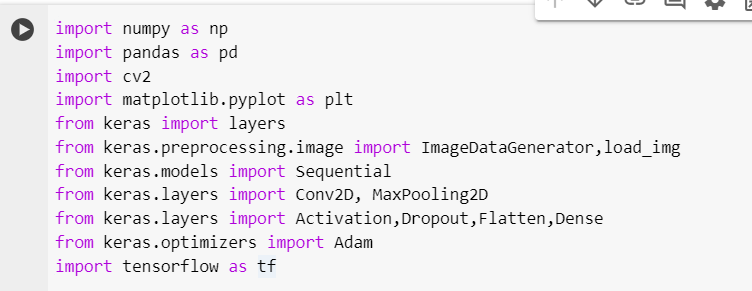


Màn hình giao diện chào mừng khi truy cập vào Google Colab

1. Khai báo các thư viện cần dùng

Khai báo thư viện Python là một phần quan trọng trong quá trình phát triển phần mềm và thực hiện các dự án lập trình. Dưới đây là một số tác dụng chính của việc khai báo thư viện Python: mở rộng chức năng, tăng tốc độ phát triển, cụ thể hóa vấn đề,....

Dưới đây là một số thư viện được sử dụng để huấn luyện mô hình AI



Bao gồm:+Numpy là một thư viện Python cung cấp hỗ trợ cho các phép toán ma trận và mảng nhiều chiều. Bằng cách import NumPy và đặt tên viết tắt là np, có thể sử dụng các hàm và công cụ của NumPy một cách tiện lợi.

+Pandas là một thư viện dành cho việc xử lý và phân tích dữ liệu. Bằng cách import Pandas và đặt tên viết tắt là pd, có thể sử dụng các cấu trúc dữ liệu như DataFrame và Series để thao tác và phân tích dữ liệu một cách thuận tiện.

+OpenCV là một thư viện mã nguồn mở cho xử lý hình ảnh và computer vision. Bằng cách import cv2, có thể sử dụng các công cụ và chức năng của OpenCV để xử lý, phân tích và nhận dạng hình ảnh.

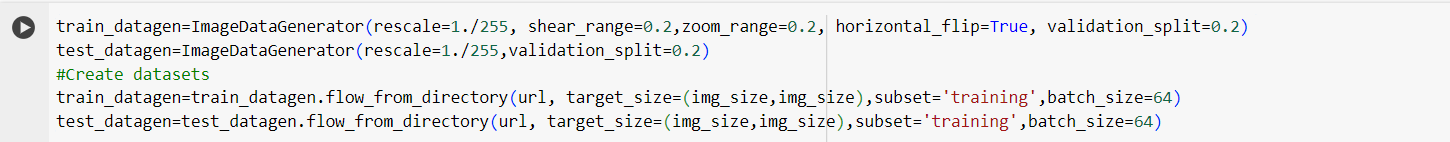
+Matplotlib là một thư viện cho việc tạo ra các biểu đồ và hình vẽ trong Python. Bằng cách import matplotlib.pyplot và đặt tên viết tắt là plt, có thể tạo ra các biểu đồ đẹp mắt để trực quan hóa dữ liệu.

+TensorFlow là một thư viện mã nguồn mở cho việc xây dựng và huấn luyện mạng neural. Bằng cách import tensorflow và đặt tên viết tắt là tf, có thể sử dụng các công cụ và chức năng của TensorFlow để xây dựng và huấn luyện mạng neural.

+Keras là một thư viện cao cấp cho việc xây dựng và huấn luyện mạng neural, tạo ra dữ liệu hình ảnh mở rộng và tăng cường (data

augmentation) trong quá trình huấn luyện mạng neural, xây dựng các tích chập và cũng như các hàm tối ưu hóa,...

1. Xử lý dữ liệu thô



Các dòng code xử lý dữ liệu thô

Code đã cung cấp thiết lập các đối tượng tạo dữ liệu hình ảnh cho tập huấn luyện và kiểm tra. Dưới đây là phân tích chi tiết về mỗi phần của mã:

1. Thiết lập ImageDataGenerator:

- train\_datagen và test\_datagen là các thể hiện của lớp ImageDataGenerator, được sử dụng để tăng cường dữ liệu thời gian thực trong quá trình huấn luyện và chuẩn hóa dữ liệu trong quá trình kiểm tra.

- rescale=1./255: Điều này chia giá trị pixel của các hình ảnh để nằm trong khoảng [0,1].

- shear\_range=0.2: Ngẫu nhiên áp dụng các biến đổi cắt.

- zoom\_range=0.2: Ngẫu nhiên áp dụng các biến đổi phóng to.

- horizontal\_flip=True: Ngẫu nhiên lật hình ảnh theo chiều ngang.

- validation\_split=0.2: Phân chia dữ liệu thành tập huấn luyện và tập xác thực, với 80% dữ liệu được sử dụng cho việc huấn luyện và 20% cho việc xác thực.

2. Tạo Bộ dữ liệu:

- train\_datagen.flow\_from\_directory() và

test\_datagen.flow\_from\_directory() tạo các bộ lặp thư mục cho tập huấn luyện và tập kiểm tra, tương ứng.

- url là đường dẫn đến thư mục chứa dữ liệu hình ảnh.

- target\_size=(img\_size, img\_size): Thay đổi kích thước hình ảnh thành kích thước được chỉ định.

- subset=training: Xác định rằng các bộ lặp này sẽ được sử dụng cho dữ liệu huấn luyện.

- batch\_size=64: Đặt kích thước cho quá trình huấn luyện và kiểm tra.

Code này thiết lập các bộ tạo dữ liệu sẽ tải hình ảnh từ thư mục chỉ định, áp dụng các biến đổi đa dạng (cho tập huấn luyện), và chuẩn bị dữ liệu để huấn luyện và kiểm tra một mô hình học sâu. Phương pháp này thường được sử dụng trong các ứng dụng học sâu cho các nhiệm vụ phân loại hình ảnh.

1. Tạo mô hình mạng nơ-ron sâu



Tạo mô hình mạng nơ-ron sâu

Mã trên định nghĩa một mô hình mạng nơ-ron sâu (deep neural network) bằng cách sử dụng API Sequential của TensorFlow Keras. Dưới đây là giải thích từng phần của mã:

1. Convolutional Layers:

- tf.keras.layers.Conv2D: Đây là lớp convolutional trong mạng nơ-ron sâu, được sử dụng để trích xuất đặc trưng của hình ảnh. Mỗi lớp này có một số filters để học các đặc trưng cụ thể.

- Số 16, 32 và 64 là số filters được sử dụng trong các lớp convolutional. Các filters này sẽ quét qua hình ảnh để tạo ra các feature maps.

2. Activation Functions:

- activation=relu: Hàm kích hoạt ReLU (Rectified Linear Activation) được sử dụng sau mỗi lớp convolutional để thêm tính phi tuyến tính vào mô hình và giúp giảm thiểu hiện tượng mất mát đạo hàm (vanishing gradient).

3. Pooling Layers:

- tf.keras.layers.MaxPool2D: Lớp pooling được sử dụng để giảm kích thước của feature maps bằng cách chọn giá trị lớn nhất từ mỗi vùng của feature maps. Việc này giúp giảm chiều dài và chiều rộng của feature maps, giúp giảm bớt số lượng tham số và tính toán.

4. Flatten Layer:

- tf.keras.layers.Flatten(): Lớp này được sử dụng để chuyển đổi các feature maps thành một vectơ 1D. Việc này là bước chuẩn bị cho các lớp fully connected (hoàn toàn kết nối) tiếp theo.

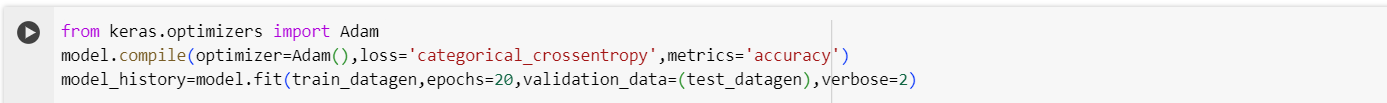
5. Dense Layers:

- tf.keras.layers.Dense: Đây là các lớp fully connected, trong đó mỗi nơ-ron trong lớp trước được kết nối với tất cả các nơ-ron trong lớp tiếp theo.

- Lớp cuối cùng có 26 nơ-ron với hàm kích hoạt softmax, cho ra xác suất dự đoán của 26 lớp đối với đầu ra mong muốn.

Mô hình này được thiết kế để phù hợp với bài toán phân loại hình ảnh, trong đó đầu vào là hình ảnh với kích thước (img\_size, img\_size, 3) và đầu ra là một trong 26 lớp.

1. Pre-train



Những dòng code trên có tác dụng:

1. Compile Model:

- model.compile: Hàm này được sử dụng để biên dịch mô hình trước khi huấn luyện. Trong đó:

- optimizer=Adam(): Chọn thuật toán tối ưu hóa Adam để điều chỉnh các trọng số của mạng nơ-ron dựa trên đạo hàm của hàm mất mát.

- loss='categorical\_crossentropy': Chọn hàm mất mát là categorical crossentropy, phù hợp với bài toán phân loại nhiều lớp.

- metrics='accuracy': Chọn độ đo để đánh giá hiệu suất mô hình trong quá trình huấn luyện là accuracy (độ chính xác).

2. Training Model:

- model.fit: Hàm này được sử dụng để huấn luyện mô hình trên dữ liệu đào tạo. Trong đó:

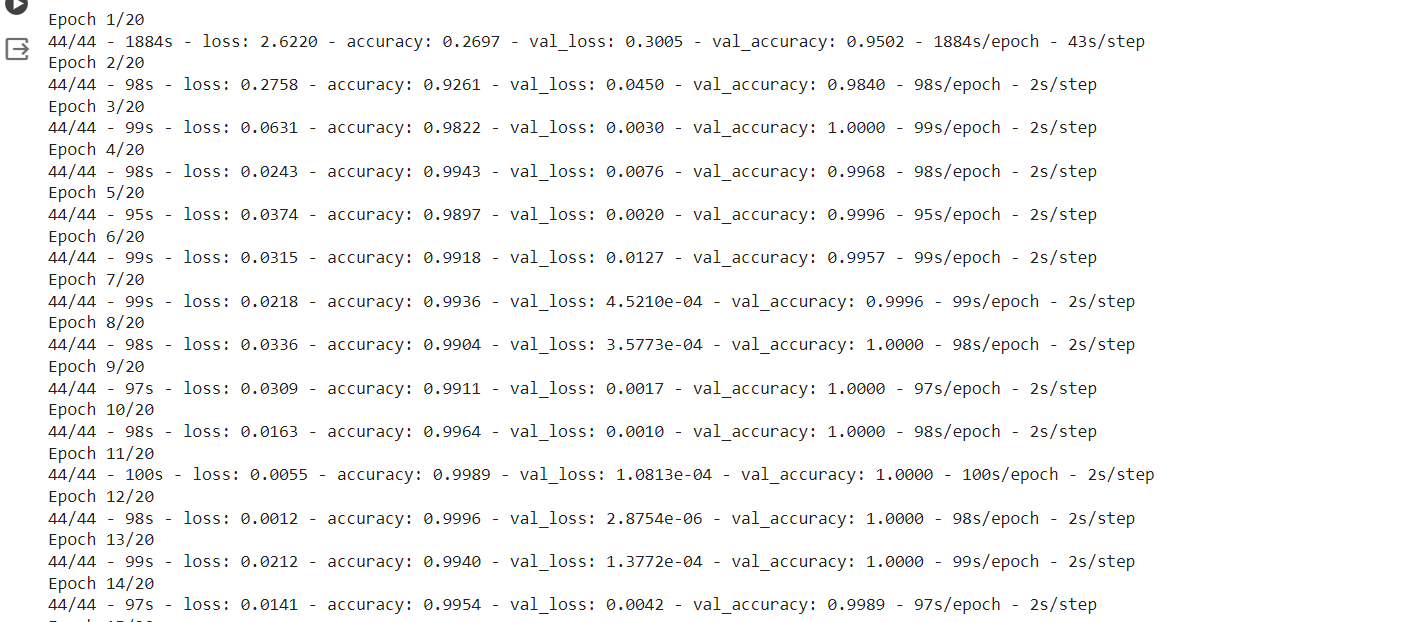
- train\_datagen: Dữ liệu đào tạo được chuyển đến hàm fit để huấn luyện mô hình.

- epochs=20: Số lần lặp lại qua toàn bộ dữ liệu đào tạo.

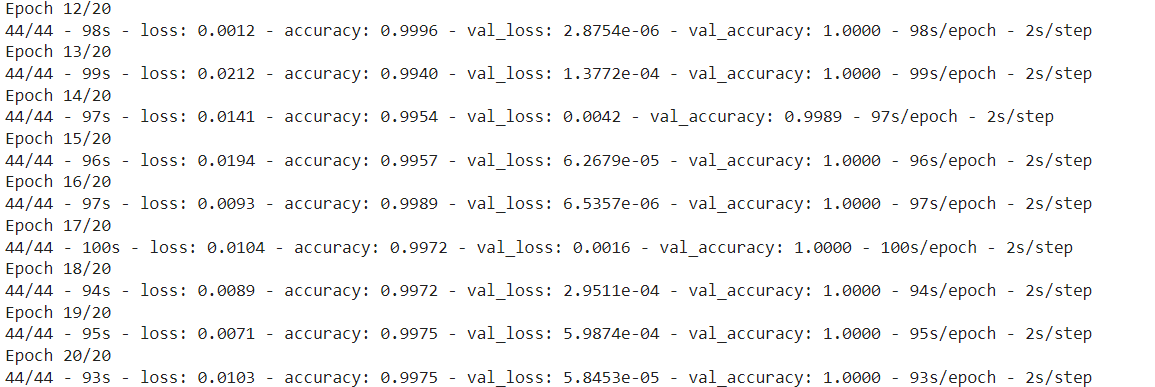
- validation\_data=(test\_datagen): Dữ liệu kiểm tra được sử dụng để đánh giá hiệu suất của mô hình sau mỗi epoch.

- verbose=2: Cấp độ chi tiết của quá trình huấn luyện, trong đó verbose=2 chỉ in ra thông tin mỗi epoch.

1. Huấn luyện



Với 20 lần huấn luyện và thời gian trung bình mỗi lần huấn luyện là khoảng 97-100s và đạt kết quả tốt khi train

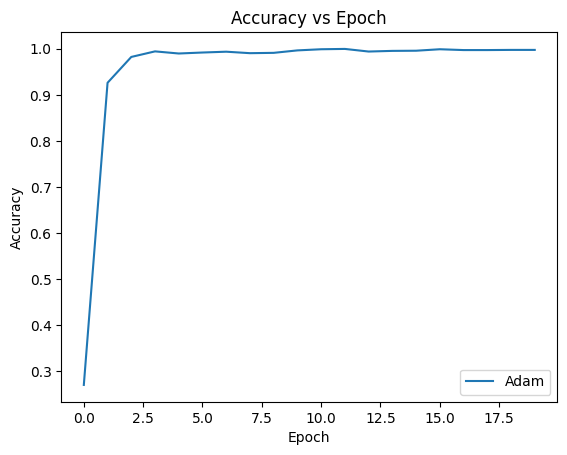


Sau 20 lần train thì độ chính xác đạt 99%

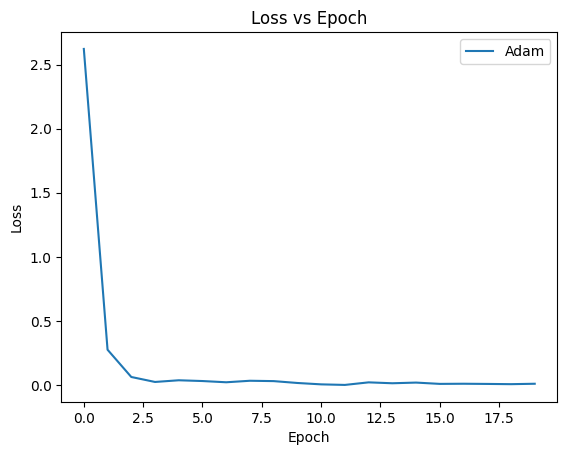
1. Kiểm tra



Kiểm tra độ chính xác và sai lệch



Độ chính xác mỗi epoch



Độ sai lệch mỗi epoch

1. Thử nghiệm mô hình

Mô hình sẽ được thử nghiệm trực tiếp trên colab bằng cách sử dụng webcam có sẵn của colab. Mô hình webcam có sẵn của colab sử dụng JavaScript và HTML để tạo các lớp hình và lưu trên Google Drive



1. Báo cáo kết quả

Mô hình AI hoạt động khá tốt với việc nhận diện các ký hiệu ngôn ngữ nhưng đôi khi vẫn xảy ra tình trạng dự đoán sai cần phải tối ưu hóa độ chính xác bằng các nâng cao chất lượng dữ liệu hình ảnh, số lượng dữ liệu hình ảnh từ nhiều góc độ, ánh sáng,....

Việc ứng dụng AI vào các hoạt động xã hội là việc làm cần được nghiên cứu và đẩy mạnh nhằm mục đích giải quyết các vấn đề xã hội cũng như tạo môi trường hòa nhập cho người giữa người cụ thể ở đây xóa bỏ khoảng cách giữa người câm và mọi người.