**BỘ CÔNG THƯƠNG**

**TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP HÀ NỘI**

****

**ĐỒ ÁN MẠNG NƠ – RON NHÂN TẠO**

**Ngành Điện Tử Viễn Thông**

**Thiết kế mô hình nhận diện thủ ngữ - ngôn ngữ ký hiệu tay - để giao tiếp với người khuyết tật.**

**GVHD: TS. Nguyễn Thị Thu**

**Nhóm 5**

**Hà Nội, Năm 2023**

**BỘ CÔNG THƯƠNG**

**TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP HÀ NỘI**

****

**ĐỒ ÁN MẠNG NƠ – RON NHÂN TẠO**

**Ngành Điện Tử Viễn Thông**

**Thiết kế mô hình nhận diện thủ ngữ - ngôn ngữ ký hiệu tay - để giao tiếp với người khuyết tật.**

**GVHD: TS. Nguyễn Thị Thu**

**Nhóm 5**

1. Vũ Thị Lan Anh MSV: 2020603356
2. Vũ Minh Đức MSV: 2020603670
3. Đào Đức Phú MSV: 2020608066
4. Nguyễn Thị Vân MSV: 2020608011
5. Nguyễn Thị Khánh Vi MSV: 2020603294

**Hà Nội, Năm 2023**

Mục Lục

[Chương 1 : Tổng Quan Về Đề Tài 5](#_Toc143029963)

[1.1 Lí do chọn dề tài. 5](#_Toc143029964)

[1.1.1 Giới thiệu chung về nhận diện thủ ngữ. 5](#_Toc143029965)

[1.1.2 Lý do lựa chọn dề tài. 6](#_Toc143029966)

[1.2 Giới thiệu chung về mạng nơ-ron nhân tạo. 7](#_Toc143029967)

[1.2.1 Giới thiệu chung về mạng nơ-ron nhân tạo. 7](#_Toc143029968)

[1.2.2 Cấu trúc cơ bản của mạng nơ – ron nhân tạo 7](#_Toc143029969)

[1.2.3 Ứng dụng của mạng nơ - ron nhận tạo 9](#_Toc143029970)

[Chương 2 : Thiết kế mô hình mạng nơ ron nhân tạo nhận diện thủ ngữ sử dụng Python. 9](#_Toc143029971)

[2.1 Giới thiệu bài toán 9](#_Toc143029972)

[2.1.1 Thủ ngữ - ngôn ngữ ký hiệu tay là gì ? 9](#_Toc143029973)

[2.1.2 Mô tả thuật toán nhận diện thủ ngữ. 10](#_Toc143029974)

[2.2 Xây dựng mô hình. 10](#_Toc143029975)

[2.2.1 Mô hình mạng noron VGG16. 10](#_Toc143029976)

[2.2.2 Thuật toán 11](#_Toc143029977)

[2.3 Huấn Luyện 12](#_Toc143029978)

[2.4 Chương trình 12](#_Toc143029979)

[2.4.1 Chương trình train\_modle.py 12](#_Toc143029980)

[2.4.2 Chương trình nhận diện. 17](#_Toc143029981)

[Chương 3 : Thực Nghiệm. 22](#_Toc143029982)

[3.1 Thực nghiệm huấn luyện. 22](#_Toc143029983)

[3.1.1 Thực hiện chạy chương trình huấn luyện 22](#_Toc143029984)

[3.1.2 Đánh giá kết quả. 22](#_Toc143029985)

[3.2 Thực nghiệm nhận diện. 22](#_Toc143029986)

[3.2.1 Thực hiện chạy chương trình nhận diện 22](#_Toc143029987)

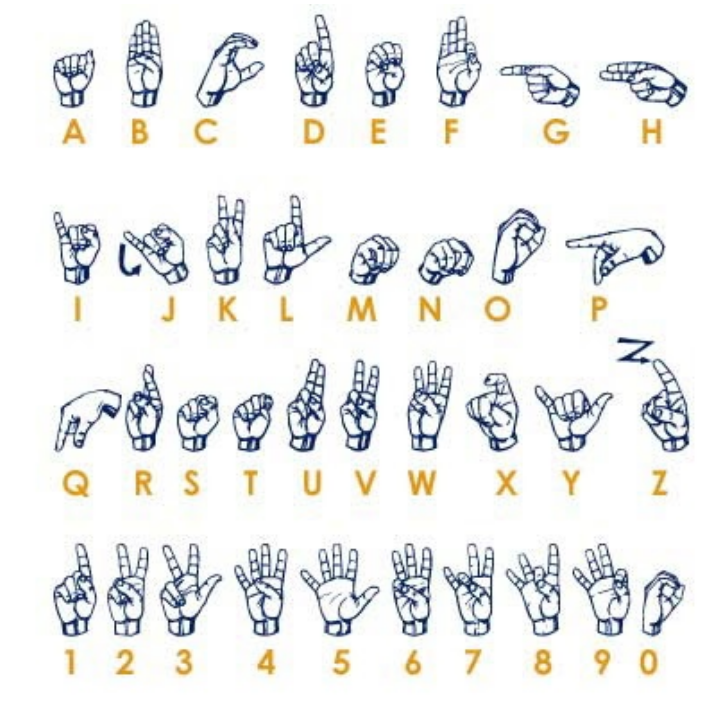
[3.2.2 Đánh giá kết quả nhận diện và so sánh với chương trình huấn luyện. 22](#_Toc143029988)

# Tổng Quan Về Đề Tài

## Lí do chọn dề tài.

### Giới thiệu chung về nhận diện thủ ngữ.

* **Thủ ngữ là gì?** Là ngôn ngữ ký hiệu hay ngôn ngữ dấu hiệu, thủ ngữ là ngôn ngữ dùng những biểu hiện của bàn tay thay cho âm thanh của tiếng nói. Ngôn ngữ ký hiệu do người khiếm thính tạo ra nhằm giúp họ có thể giao tiếp với nhau trong cộng đồng của mình và tiếp thu tri thức xã hội.
* **Nhận diện thủ ngữ:** Được sử dụng để thể hiện ý nghĩa tinh vi, mô tả hình ảnh. Chúng tạo tính sáng tạo và mang lại màu sắc ngôn ngữ, nó cũng đóng góp phần tạo nên các văn bản cổ ngữ, câu chuyện…
* **Bảng chữ cái thủ ngữ :**



Hình : bảng chữ cái thủ ngữ

### Lý do lựa chọn dề tài.

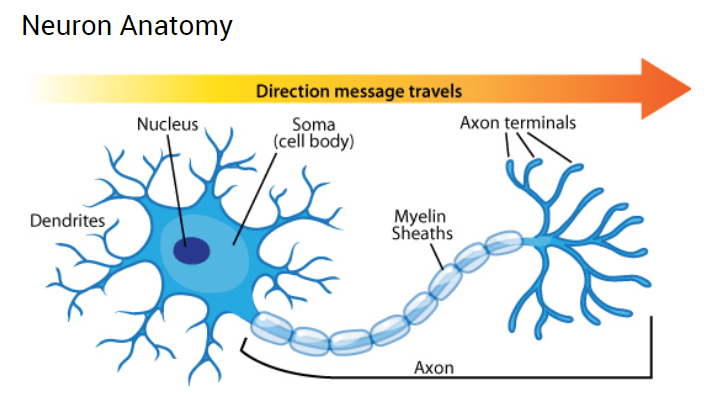
* ***Phương diện lý thuyết***: Giúp cho người khuyết tật có thể giao tiếp, sử dụng máy tính và các công cụ công nghệ cao một cách thuận tiện và dễ dàng mà không cần thông qua người phiên dịch, thuận tiện cho người khuyết tật đặc biệt là những người khiếm thính.
* ***Phương diện thực tiễn***: Người dùng chỉ cần sử dụng các động tác tay trong bảng chữ cái của thủ ngữ thì chương trình có thể nhận diện và xác định được thủ ngữ người dùng muốn là gì và đưa ra chữ dưới dạng bảng chữ cái lating ra màn hình cho người khác hiểu. Người dùng có thể tự mình học tập và sử dụng các thiết bị điện tử cũng như giao tiếp với người khác thông qua phần mềm mà không tốn quá nhiều công sức.
* ***Đối tượng nghiên cứu***: Thiết kế mô hình nhận diện thủ ngữ - ngôn ngữ kí hiệu tay dùng để giao tiếp với người khuyết tật để giúp họ có thể giao tiếp với mọi người và sử dụng các thiết bị điện tử dễ dàng hơn.
* ***Phạm vi nghiên cứu***: Tập chung vào nhận diện thủ ngữ cơ bản và không đi sâu vào các thủ ngữ phức tạp. Khả năng hiệu quả của mô hình có thể bị giới hạn về dữ liệu hoặc khả năng tính toán.
* ***Mục tiêu nghiên cứu:***
* **Mục tiêu tổng quát**: Xây dựng một ứng dụng nhận diện cử chỉ tay của người dùng, sau đó từ cử chỉ nhận dậng xuất ra màn hình chữ được nhận diện.
* **Mục tiêu cụ thể**: Học và hiểu cơ bản bảng chữ cái thủ ngữ, về phương diện chuyên môn cần phải nắm rõ các kiến thức về máy học, nắm rõ cấu trúc thuật toán sử dụng, tìm kiếm thông tin về bảng chữ cái thủ ngữ dưới dạng hình ảnh. Thiết lập cấu hình giao diện máy tính để máy tính nhận diện qua camera thiết bị. Thể hiện rõ hình ảnh, nhận diện chính xác từ được sử dụng của người dùng.
* ***Phương pháp nghiên cứu***:
* Tìm hiểu về bảng chữ cái thủ ngữ cho người khuyết tật ở đây tôi dùng bảng chữ cái ASL.
* Nghiên cứu các đề tài hoặc bài toán tương tự trên internet.
* Chọn mô hình thực hiện: CNN, VGG,…. Ở đây tôi chọn VGG16
* Nghiên cứu và sử dụng Visial Studio code bắt đầu thiết kế giao diện người dùng

## Giới thiệu chung về mạng nơ-ron nhân tạo.

### Giới thiệu chung về mạng nơ-ron nhân tạo.

***Mạng nơ-ron nhân tạo*** là một chuỗi những thuật toán được đưa ra để tìm kiếm các mối quan hệ cơ bản trong tập hợp các dữ liệu. Thông qua việc bắt trước cách thức hoạt động từ não bộ con người. Nói cách khác, mạng nơ ron nhân tạo được xem là hệ thống của các tế bào thần kinh nhân tạo. Đây thường có thể là hữu cơ hoặc nhân tạo về bản chất.

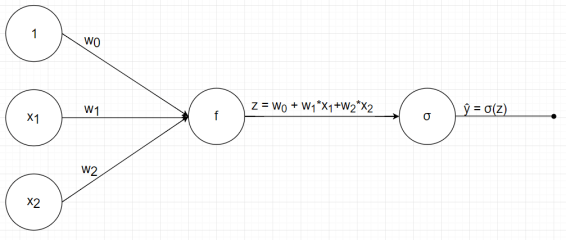
***Hoạt động của nơ-ron:***



Hình : Hoạt động của mạng noron

### Cấu trúc cơ bản của mạng nơ – ron nhân tạo

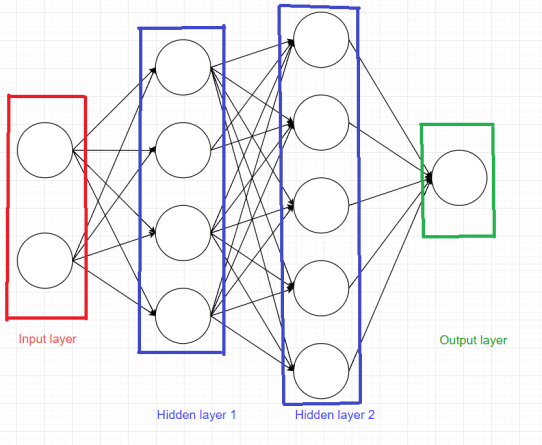
* Logistic regression là mô hình neural network đơn giản nhất chỉ với input layer và output layer.
* Mô hình của logistic regression là: y^= *σ* (*w*0+ *w*1∗ *x*1+ *w*2∗ *x*2) Có 2 bước:
* Tính tổng linear:  *z*=1∗*w*0​+*x*1​∗*w*1​+*x*2​∗*w*2​
* Áp dụng sigmoid function: *y*^​=*σ*(*z*)



Hình : Mô hình logistic regression.

Mô hình logistic regresion

* ***Mô hình tổng quát:***



Hình : Kiến trúc mạng noron nhân tạo.

Mô hình neural network

* Layer đầu tiên là input layer, các layer ở giữa được gọi là hidden layer, layer cuối cùng được gọi là output layer. Các hình tròn được gọi là node.
* Mỗi mô hình luôn có 1 input layer, 1 output layer, có thể có hoặc không các hidden layer. Tổng số layer trong mô hình được quy ước là số layer – 1 (Không tính input layer).
* Ví dụ như ở hình trên có 1 input layer, 2 hidden layer và 1 output layer. Số lượng layer của mô hình là 3 layer.

Mỗi node trong hidden layer và output layer :

* Liên kết với tất cả các node ở layer trước đó với các hệ số w riêng.
* Mỗi node có 1 hệ số bias b riêng.
* Diễn ra 2 bước: tính tổng linear và áp dụng activation function.

### Ứng dụng của mạng nơ - ron nhận tạo

* Nhận diện khuôn mặt
* Dự đoán thị trường chứng khoán
* Mạng xã hội
* Quốc phòng
* Chăm sóc sức khỏe

# Thiết kế mô hình mạng nơ ron nhân tạo nhận diện thủ ngữ sử dụng Python.

## Giới thiệu bài toán

### Thủ ngữ - ngôn ngữ ký hiệu tay là gì ?

* Thủ ngữ là một hệ thống giao tiếp bằng cách sử dụng các ngôn ngữ kí hiệu tay, cử chỉ và biểu hiện mặt để truyền đạt ý nghĩa. Nó được dùng bởi người điếc và người khiếm thính như một phương thức để giao tiếp.
* Ngôn ngữ ký hiệu tay:
* Là hình thức ngôn ngữ tự nhiên, phát triển và sử dụng bởi cộng đồng người khiếm thính. Nó có các quy tắc ngữ pháp, từ vựng riêng. Ngôn ngữ ký hiệu tay quốc tế (International Sign Language) để giao tiếp giữa người khiếm thính từ các quốc gia khác nhau.
* Sử dụng các cử chỉ tay, vị trí, hình dạng của các điểm trong không gian để biểu thị các từ. Ngoài ra cử chỉ cơ thể, cử chỉ miệng, … cũng sử dụng để truyền đạt cảm xúc. Nó đóng vai trò quan trọng giúp người khiếm thính có thể giao tiếp, ... và tham gia cộng đồng một cách tự nhiên và hiệu quả.

### Mô tả thuật toán nhận diện thủ ngữ.

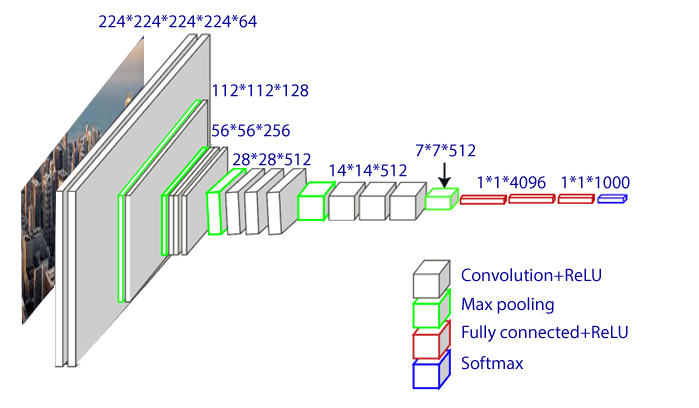
* **Train**:
  + Ta chuẩn bị sẵn các ảnh đầu vào là các ảnh bàn tay đang thực hiện các ký tự nói trên và lưu trong thư mục Data. Các ảnh được gán nhãn đầy đủ bằng tên file.
  + Ta thực hiện chuyển các ảnh đó về size 224 x224 và đưa qua mạng VGG16 đã được thêm 1 vài layer Dense (Fully Connect) và cuối cùng là 1 lớp softmax để dự đoán kết quả đầu ra.
  + Ta thực hiện train mạng của chúng ta với khoảng hơn 2000 ảnh train và gần 300 ảnh test.
* **Chạy thử:**
  + Ta thực hiện mở Webcam và khoanh 1 vùng sẽ phát hiện bàn tay ở trong vùng đó. Gọi là detection region.
  + Người dùng bỏ tay ra khỏi vùng detection region và nhấn phím B để máy thu nhận nền của vùng. Nếu muốn thu nhận lại nền, người dùng bấm phím R.
  + Người dùng đưa tay vào trong vùng detection (sau khi đã capture nền) và tạo thành các hình ký tự.
  + Model sẽ hiển thị ký tự ra màn hình.
  + Người dùng bấm Q để thoát.

## Xây dựng mô hình.

### Mô hình mạng noron VGG16.

Mạng VGG16 là một kiến trúc mạng noron sâu bao gồm 16 lớp tham gia vào mạng, trong đó có 13 lớp tích chập (Convolutional Layers) và 3 lớp kết nối đầy đủ (Fully Connected Layers).

Mạng này nổi bật bởi kiến trúc đơn giản và đồng nhất với các lớp tích chập có kích thước nhỏ(3x3) và các lớp kết nối đầy đủ ở cuối để thực hiện phân loại.



Hình : Cấu trúc mô hình mạng noron VGG16

Cấu trúc của mạng VGG16:

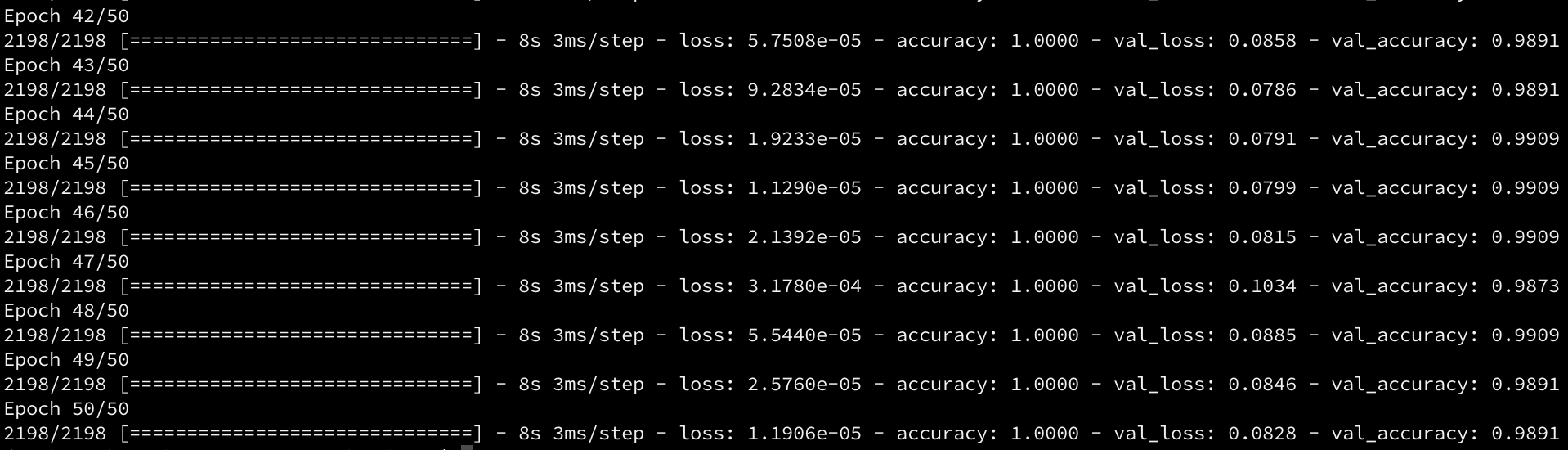
* **Input Layer:**
  + Kích thước: 224x224x3(màu RGB).
  + Lớp tích chập với 64 bộ lọc 3x3 hà hàm kích hoạt ReLU.
* **Chuỗi các lớp tích chập:** lớp này bao gồm chuỗi các lớp tích chập với 64,128,512 bộ lọc 3x3 và hàm kích hoạt ReLU cùng với lớp MaxPooling với bước nhảy 2x2.
* **Lớp Fully Connected:** 
  + Lớp Flatten: Chuyển đổi đầu ra từ lớp trước thành vector 1D.
  + Lớp Dense (Fully connected) với 4096 đơn vị và hàm kích hoạt ReLU.
  + Lớp Dropout với tỉ lệ Dropout là 0.5
* **Lớp Fully Connected tiếp theo:**
  + Lớp Dense với 4096 đơn vị và hàm kích hoạt ReLU.
  + Lớp Dropout với tỉ lệ dropout là 0.5
* **Lớp Output:**
  + Lớp Dense cuối cùng với 1000 đơn vị (tương ứng với số lớp classes trong tập dữ liệu ImageNet) và hàm kích hoạt softmax để tính toán xác suất dự đoán cho từng lớp.

### Thuật toán

* Các bước xây dựng chương trình training:
  + - **Bước 1:** Giai đoạn tiền xử lý dữ liệu training. (đọc và mở hình ảnh từ đường dẫn sau đó chuyển đổi kích thước ảnh rồi chuyển đổi hình ảnh thành mảng numpy)
    - **Bước 2:** Xử lý dữ liệu
    - **Bước 3:** Duyệt qua tất cả dữ liệu hình ảnh, xử lý và chuyển đổi dữ liệu hình ảnh. Trả về dữ liệu huấn luyện
    - **Bước 4:** Tải dữ liệu Huấn Luyện và phân chia thành tập train và test.
    - **Bước 5:** Xây dựng mô hình Training.
    - **Bước 6:** Huấn Luyện Mô Hình.
    - **Bước 7:** Lưu mô hình đã huấn luyện ra file
* Các bước xây dựng chương trình nhận diện.
  + - **Bước 1:** tải mô hình huấn luyện.
    - **Bước 2:** xử lý hình ảnh
    - **Bước 3:** xử lý nhận dạng
    - **Bước 4:** Xử lý phím bấm và hiển thị kết quả.
    - **Bước 5:** giải phóng tài nguyên và kết thúc chương trình.

## Huấn Luyện

* Project bao gồm dữ liệu input, file train\_model.py và dection.py. hãy chạy file train\_model.py để bắt đầu huấn luyện.



Hình : Quá trình trainning model.

* Sau khi train xong, trong thư mục models sẽ xuất hiện file mymodel.h5 chính là file lưu model.

## Chương trình

### Chương trình train\_modle.py

**Bắt đầu:**

* Import các thư viện mà module cần thiết:

A screenshot of a computer program

Description automatically generated

Hình : Import thư viện cần thiết.

* Khởi tạo các biến:
  + - “gestures”,”gestures\_map”,”gestures\_name” : là những dictionary để ánh xạ giữa cử chỉ và các giá trị số.
    - “image\_path”: Đường dẫn thư mục chưa dữ liệu hình ảnh.
    - “models\_path”: Đường dẫn lưu mô hình đã huấn luyện.
    - “rgb” : Biến kiểm soát chế độ xử lý RGB hay không.
    - “imageSize”: Biến kích thước ảnh đầu vào khi được xử lý.

A screenshot of a computer program

Description automatically generated

Hình : Khởi tạo các biến cần thiết trong quá trình trainning.

**Thân chương trình:**

* + - Hàm “process\_image(path)”:
      * Đọc và mở hình ảnh từ “path”.
      * Thay đổi kích thước hình ảnh thành “imageSize x imageSize“ (224x224).
      * A screen shot of a computer program

        Description automatically generatedChuyển đổi hình ảnh thành mảng numpy.

Hình : hàm xử lý hình ảnh sang dạng mảng numpy

* + - Hàm “process\_data(X\_data,Y\_data)”:
      * Chuyển đổi dữ liệu hình ảnh và nhãn thành định dạng phù hợp để đưa vào mii hình.
      * Chuẩn hóa giá trị của giữ liệu hình ảnh về khoảng [0,1].
      * One-hot encode nhãn.
    - A screen shot of a computer program

      Description automatically generatedhàm “walk\_file\_tree(image\_path)”:

Hình : hàm xử lý dữ liệu numpy

* + - * Duyệt qua tất cả các tệp hình ảnh thuộc thư mục “image\_path”
      * Xử lý chuyển đổi dữ liệu hình ảnh và nhãn
      * A computer screen shot of a program

        Description automatically generatedTrả về dữ liệu huấn luyện

Hình : Hàm Duyệt tệp ảnh.

* + - Mô hình training:
      * Gọi hàm “walk\_file\_tree(image\_path)” để tải dữ liệu hình ảnh và nhãn.
      * Sử dụng “train\_test\_split” để phân chia tập train và test
      * Đặt các checkpoint để lưu lại model tốt nhất.
      * Gọi mô hình VGG16.
      * Thêm các lớp fully connected layers để thực hiện phân loại tùy chỉnh.
      * Đóng băng các lớp dưới của mô hình VGG16 để chỉ huấn luyện các lớp thêm vào.

A screen shot of a computer program

Description automatically generated**Kết thúc chương trình**

Hình : xử lý VGG16 và thêm các lớp bên trên

* + - Biên dịch và huấn luyện mô hình sau đó lưu vào file.

A screen shot of a computer code

Description automatically generated

Hình : Biên dịch và lưu lại model.

### Chương trình nhận diện.

* Bắt đầu:
  + A screen shot of a computer program

    Description automatically generatedImport những thư viện cần thiết.

Hình : Thêm các thư viện cần thiết cho chương trình nhận diện

* + Khai báo biến:
    - “prediction”: Biến lưu trữ dự đoán kí tự.
    - “score”: Biến lưu điểm dự đoán.
    - A screen shot of a computer program

      Description automatically generated“bgModel”” Biến lưu trữ mô hình nền.

Hình : Khai báo các biến cần thiết cho chương trình nhận diện

Thân chương trình:

* + Tải mô hình đã huấn luyện:
    - A computer screen shot of a program

      Description automatically generatedSử dụng “load\_model” để tải mô hình đã huấn luyện từ tệp mymodels.h5.

Hình : Tải lên mô hình huấn luyện từ tệp mymodel.h5

* + Hàm “predict\_rgb\_image\_vgg(image)”:
    - Chuyển đổi hình ảnh đầu vào thành mảng numpy và chuẩn hóa giá trị về khoảng [0,1].
    - Sử dụng mô hình để dự đoán lớp của hình ảnh.
    - Tính toán xác suất dự đoán cao nhất và ánh xạ kết quả bằng dictionary “gesture\_name”.
    - A computer screen shot of a program code

      Description automatically generatedTrả về kết quả dự đoán và điểm xác suất.

Hình : Hàm nhận diện và đánh giá xác suất.

* + Hàm “remove\_background(frame)”:
    - Sử dụng mô hình nền (“bgModels”) để tách nền ra khỏi hình ảnh.
    - Áp dụng phép AND trên hình ban đầu và nền đã được tách.

A screen shot of a computer code

Description automatically generated

Hình : Hàm xóa nền ảnh

* + Các tham số và biến cho quá trình nhận dạng.

A screenshot of a computer program

Description automatically generated

Hình : Các tham số cho quá trình nhận dạng

* + Xử lý hình ảnh từ camera.
    - Đọc hình ảnh cử camera,
    - Áp dụng xử lý hình ảnh như làm mờ và lật ngang hình ảnh.
    - Vẽ khung hình nhận dạng lên hình ảnh.
  + Xử lý và nhận dạng.
    - Tách nền khỏi hình ảnh và lấy vùng nhận dạng
    - Chuyển hình ảnh sang đen trắng và áp dụng bộ lọc Gaussian để làm mờ.
    - Tạo ngưỡng (threshold) bằng phương pháp Otsu để tách kí tự với nền.
    - Nếu tỉ lệ điểm đối với các điểm không phải là nền trên 1 ngưỡng, tiến hành dự đoán.
      * Chuẩn bị hình ảnh cho mạng VGG16.
      * Sử dụng hàm “predict\_rgb\_image\_vgg” để dự đoán kí tự và tính điểm.
      * Nếu điểm dự đoán đạt ngưỡng “preThreshold”, hiển thị kí tự dự đoán lên hình ảnh.
  + Xử lý phím bấm và đưa ra kết quả
    - Xử lý các phím bấm như “q” để thoát, “b” để lấy mẫu nền, “r” để đặt lại nền.
    - Hiển thị kết quả lên màn hình.
* Kết thúc chương trình:
  + Giải phóng tài nguyên và kết thúc chương trình.

# Thực Nghiệm.

## Thực nghiệm huấn luyện.

### Thực hiện chạy chương trình huấn luyện

Thực hiện chạy chương trình huấn luyện với file tran\_model.py với epochs = 30 và batch\_size = 50. Tỉ lệ train và test cho dữ liệu train là 80% và 20%.

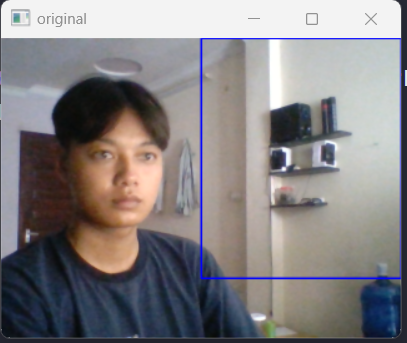
### Đánh giá kết quả.

## Thực nghiệm nhận diện.

### Thực hiện chạy chương trình nhận diện

Thực hiện chạy chương trình nhận diện với một số kí tự cơ bản. Các bước thực hiện:

* Đầu tiên chạy file dectection.py:



Hình : Cửa sổ nhận diện

* Căn chỉnh người sao cho vùng trống màu xanh không có vật gì chuyển động trong đó. Sau đó bấm phím B để bắt đầu nhận nền.

Screens screenshot of a computer

Description automatically generated

Hình : Cửa sổ xóa nền và nhận diện ẩn

* Sau đó nó hiện lên 2 cửa sổ. bên phải là cửa sổ nhận diện ẩn. bên trái là cửa sổ tách nền.
* A screenshot of a video call

  Description automatically generatedA computer screen shot of a number

  Description automatically generatedBắt đầu quá trình nhận diện:

Hình : hình ảnh quá trình nhận diện.

### Đánh giá kết quả nhận diện và so sánh với chương trình huấn luyện.

* Đánh giá kết quả nhận diện:
  + A screenshot of a video call

    Description automatically generated Đối với những kí tự có kí hiệu tay đơn giản như: C, Y, L,.. thì chương trình nhận dạng rất chính xác tỉ lệ 80-100%,

Hình : Nhận diện 2 chữ cái L và C

* A screenshot of a video call

  Description automatically generatedTuy nhiên, có một số kí tự vẫn bị nhầm lẫn với nhau như G với H; D với U,..

Hình : Nhận diện chữ D nhưng hiển thị lại là chữ Y

* Như hình trên ta thấy, mặc dù tay đang kí hiện chữ D nhưng máy nhận diện ra chữ y với tỉ lệ lên đến 99%

A screenshot of a computer program

Description automatically generated

Hình : cửa sổ terminal nhận diện sai chữ D sang chữ Y

Kết Luận

Theo đánh giá và tìm hiểu thì nhóm kết luận rằng hệ thống nhận diện của nhóm chưa ổn định với những kí tự khó. Lí do là chất lượng ảnh train còn thấp (60x60) và train với kích thước ảnh là (120x120) thêm vào đó, do tác động của ánh sáng môi trường xung quanh cũng làm ảnh hưởng tới kết quả của chương trình nhận diện.

Tài Liệu Tham Khảo

* Ebook Machine Learning Cơ Bản – Vũ Hữu Tiệp.
* Nguồn datasheet được lấy từ github.com