

# UNIVERSITY OF INFORMATION TECHNOLOGY

MÔN HỌC: MÁY HỌC - CS114.M11

---

## Nhận diện ngôn ngữ ký hiệu

---

*Student:*

- Nhóm trưởng: Nguyễn  
Đỗ Quang  
MSSV: 20520720  
Lớp:  
CS114.M11.KHCL
- Thành viên: Trần  
Nguyễn Quỳnh Anh  
MSSV: 19521217  
Lớp: CS114.M11
- Thành viên: Đinh  
Hoàng Linh Đan  
MSSV: 19521309  
Lớp: CS114.M11

*Lecturer:*

Lê Đình Duy  
Phạm Nguyễn Trường An



## Contents

|           |  |           |
|-----------|--|-----------|
| <b>1</b>  | <b>GIẢI TRÌNH CHỈNH SỬA SAU VẤN ĐÁP:</b>   | <b>2</b>  |
| 1.1       | Mô tả lại quá trình làm dữ liệu: . . . . . | 2         |
| 1.2       | Giới thiệu thêm về Model: . . . . .        | 2         |
| 1.3       | Đánh giá kết quả: . . . . .                | 4         |
| 1.4       | Một số từ dễ bị nhận nhầm: . . . . .       | 8         |
| <b>2</b>  | <b>Mô tả bài toán:</b>                     | <b>8</b>  |
| <b>3</b>  | <b>Bài toán:</b>                           | <b>9</b>  |
| <b>4</b>  | <b>Mô tả bộ dữ liệu:</b>                   | <b>9</b>  |
| <b>5</b>  | <b>Phương pháp tiếp cận bài toán:</b>      | <b>10</b> |
| <b>6</b>  | <b>Model Training:</b>                     | <b>13</b> |
| <b>7</b>  | <b>Đánh giá model:</b>                     | <b>13</b> |
| <b>8</b>  | <b>Ứng dụng thực tế:</b>                   | <b>14</b> |
| <b>9</b>  | <b>Hướng phát triển của đề án:</b>         | <b>15</b> |
| <b>10</b> | <b>Tài liệu tham khảo</b>                  | <b>15</b> |

# 1 GIẢI TRÌNH CHỈNH SỬA SAU VẤN ĐÁP:

## 1.1 Mô tả lại quá trình làm dữ liệu:

Đầu tiên, xây dựng code để thu thập dữ liệu.

Mỗi từ sẽ được người thực hiện ngôn ngữ kí hiệu thể hiện 30 lần trước Webcam của laptop.

Mỗi lần thực hiện ngôn ngữ kí hiệu sẽ được lưu lại với 30 file .npp trong đó chứa tọa độ của những keypoint từ mặt, tay và cử chỉ của người thực hiện ngôn ngữ kí hiệu.

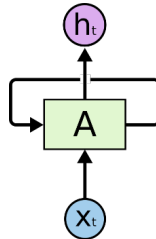
## 1.2 Giới thiệu thêm về Model:

- RNN:

*Giới thiệu:*

Ý tưởng chính của **RNN (Recurrent Neural Network)** là sử dụng chuỗi các thông tin. RNN thực hiện cùng một tác vụ cho tất cả các phần tử của một chuỗi với đầu ra phụ thuộc vào cả các phép tính trước đó. Nói cách khác, RNN có khả năng nhớ các thông tin được tính toán trước đó.

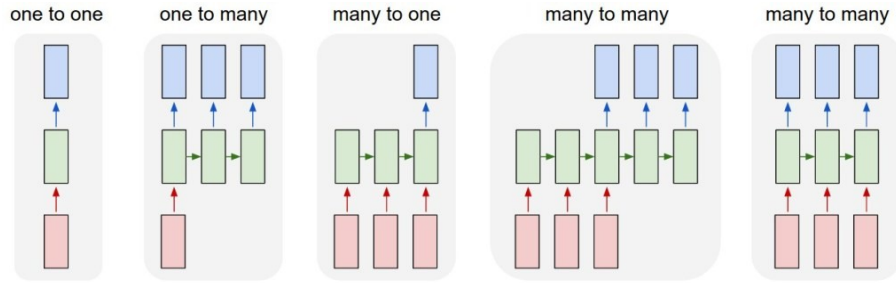
RNN chứa các vòng lặp bên trong cho phép thông tin có thể lưu lại được.



Hình 1: Hình vẽ trên mô tả một đoạn của mạng nơ-ron hồi quy **A** với đầu vào là  $x_t$  và đầu ra là  $h_t$

Một vòng lặp cho phép thông tin có thể được truyền từ bước này qua bước này qua bước khác của mạng nơ-ron.

*Phân loại:*



Hình 2: Những phân loại của model RNN

**One to one:** mẫu bài toán cho Neural Network (NN) và Convolutional Neural Network (CNN), 1 input và 1 output, ví dụ với CNN input là ảnh và output là ảnh được segment.

**One to many:** bài toán có 1 input nhưng nhiều output, ví dụ: bài toán ghi chú cho ảnh, input là 1 ảnh nhưng output là nhiều chữ mô tả cho ảnh đấy, dưới dạng một câu.

**Many to one:** bài toán có nhiều input nhưng chỉ có 1 output, ví dụ bài toán phân loại hành động trong video, input là nhiều ảnh (frame) tách ra từ video, output là hành động trong video.

**Many to many:** bài toán có nhiều input và nhiều output, ví dụ bài toán dịch từ tiếng anh sang tiếng việt, input là 1 câu gồm nhiều chữ: "I love Vietnam" và output cũng là 1 câu gồm nhiều chữ "Tôi yêu Việt Nam".

- **LSTM:**

Nhưng khoảng cách càng lớn dần thì RNN bắt đầu không thể nhớ và học được nữa.

LSTM (Long Short Term Memory networks) là một dạng đặc biệt của RNN. LSTM được giới thiệu bởi Hochreiter Schmidhuber (1997), và sau đó đã được cải tiến và phổ biến bởi rất nhiều người trong ngành.

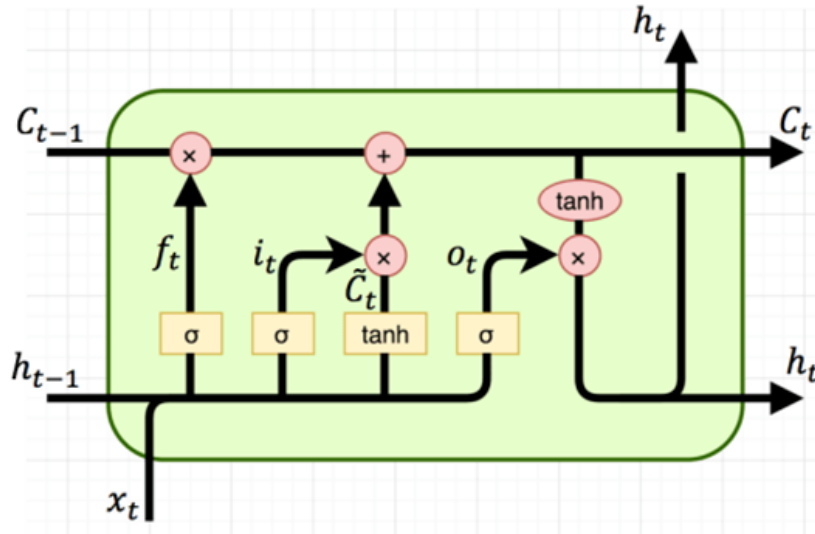
LSTM được thiết kế để tránh được vấn đề phụ thuộc xa (long-term dependency). Việc nhớ thông tin trong suốt thời gian dài là đặc tính mặc định của chúng, chứ ta không cần phải huấn luyện nó để có thể nhớ được. Tức là ngay nội tại của nó đã có thể ghi nhớ được mà không cần bất kì can thiệp nào.

Ở giai đoạn thứ  $t$  của mô hình LSTM:

+ Output:  $c_t, h_t$  ta gọi  $c$  là cell state,  $h$  là hidden state.

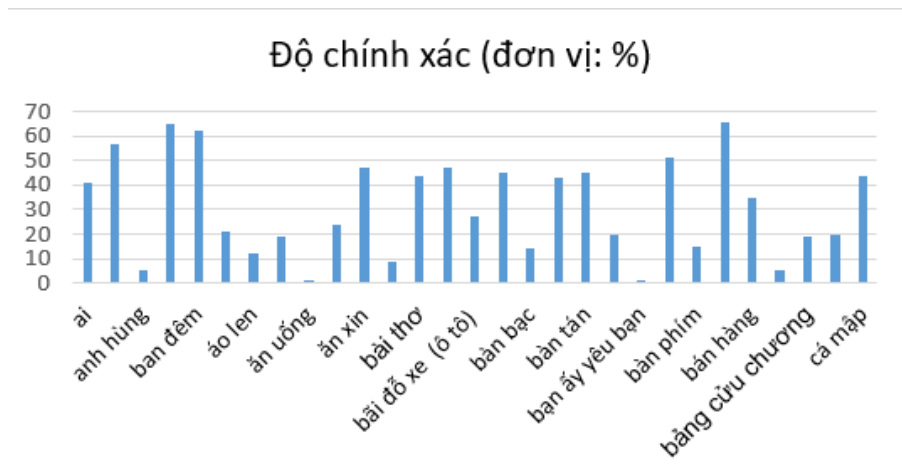
+ Input:  $c(t-1), h(t-1), x_t$  trong đó  $x_t$  là input ở giai đoạn thứ  $t$  của model.

$c_{(t-1)}, h_{(t-1)}$  là output của layer trước.



Hình 3: Model LSTM

### 1.3 Đánh giá kết quả:



Hình 4: Biểu đồ minh họa thể hiện độ chính xác của 30 từ đầu tiên trong bộ dữ liệu

Độ chính xác (accuracy score) khi được tính theo confusion matrix

```
> confusion_matrix(ytrue,yhat)
[26] ✓ 0.4s
... array([[1, 0, 0, ..., 0, 0, 0],
          [0, 1, 0, ..., 0, 0, 0],
          [0, 0, 2, ..., 0, 0, 0],
          ...,
          [0, 0, 0, ..., 2, 0, 0],
          [0, 0, 0, ..., 0, 1, 0],
          [0, 0, 0, ..., 0, 0, 0]], dtype=int64)
```

Hình 5: Confusion Matrix của tất cả các từ

```
> accuracy_score(ytrue,yhat)
[27] ✓ 0.2s
... 0.7808219178082192
```

Hình 6: Độ chính xác của tất cả các từ

Độ chính xác khi test trên thời gian thực.

| STT | Tên từ           | Độ chính xác (đơn vị %) |
|-----|------------------|-------------------------|
| 1   | ai               | 41                      |
| 2   | ác mộng          | 57                      |
| 3   | anh hùng         | 5                       |
| 4   | ác độc           | 65                      |
| 5   | bán đêm          | 62                      |
| 6   | áo               | 21                      |
| 7   | áo len           | 12                      |
| 8   | ăn hối lộ        | 19                      |
| 9   | ăn uống          | 1                       |
| 10  | ăn vụng          | 24                      |
| 11  | ăn xin           | 47                      |
| 12  | bài báo          | 9                       |
| 13  | bài thơ          | 44                      |
| 14  | bãi cát          | 47                      |
| 15  | bãi đỗ xe (ô tô) | 27                      |
| 16  | bán ngày         | 45                      |
| 17  | bàn bạc          | 14                      |

|    |                       |    |
|----|-----------------------|----|
| 18 | bàn chân              | 43 |
| 19 | bàn tán               | 45 |
| 20 | bàn chải đánh răng    | 20 |
| 21 | bạn ấy yêu bạn        | 1  |
| 22 | bảng chữ cái ngón tay | 51 |
| 23 | bàn phím              | 15 |
| 24 | bàn tay               | 66 |
| 25 | bán hàng              | 35 |
| 26 | bạn ấy yêu tôi        | 5  |
| 27 | bảng cửu chương       | 19 |
| 28 | bán kính              | 20 |
| 29 | cá mập                | 44 |
| 30 | cá voi                | 35 |
| 31 | cách ly               | 56 |
| 32 | cãi nhau              | 67 |
| 33 | cái gương             | 60 |
| 34 | cái túi               | 66 |
| 35 | cảnh giác             | 22 |
| 36 | kèn                   | 8  |
| 37 | cái kéo               | 65 |
| 38 | kẹo                   | 26 |
| 39 | kể chuyện             | 66 |
| 40 | kết quả               | 27 |
| 41 | khả năng              | 48 |
| 42 | khác nhau             | 41 |
| 43 | khách                 | 48 |
| 44 | khách sạn             | 70 |
| 45 | khai mạc              | 20 |
| 46 | kháng sinh            | 57 |
| 47 | khăn mặt              | 63 |
| 48 | khâu vá               | 13 |
| 49 | khẩu trang            | 63 |
| 50 | la hét                | 55 |
| 51 | là gì?                | 28 |
| 52 | lá cây                | 48 |
| 53 | lá thư                | 24 |
| 54 | lạc quan              | 43 |
| 55 | lái tàu hỏa           | 68 |
| 56 | làm bài tập           | 64 |
| 57 | làm được              | 29 |
| 58 | làm ruộng             | 3  |
| 59 | làm việc              | 67 |

|     |                     |    |
|-----|---------------------|----|
| 60  | lang tang           | 40 |
| 61  | lãng phí            | 67 |
| 62  | lãnh đạo            | 10 |
| 63  | lạnh lùng           | 48 |
| 64  | người già (đàn ông) | 14 |
| 65  | lau chùi            | 58 |
| 66  | lau miệng           | 51 |
| 67  | lắc đầu             | 39 |
| 68  | lắng nghe           | 16 |
| 69  | im lặng             | 8  |
| 70  | lắp ráp             | 45 |
| 71  | lẫn lộn             | 49 |
| 72  | lâu                 | 0  |
| 73  | ma sát              | 28 |
| 74  | ma túy              | 34 |
| 75  | má hồng             | 70 |
| 76  | mái vòm             | 2  |
| 77  | màu đen             | 32 |
| 78  | màu nâu             | 22 |
| 79  | màu trắng           | 40 |
| 80  | màu xanh da trời    | 2  |
| 81  | xa cách             | 64 |
| 82  | xà phòng            | 68 |
| 83  | xả rác              | 45 |
| 84  | xào                 | 66 |
| 85  | xảy ra              | 1  |
| 86  | xã hội              | 39 |
| 87  | xác chết            | 7  |
| 88  | xác định            | 9  |
| 89  | xách nặng           | 32 |
| 90  | xăm                 | 68 |
| 91  | xăng                | 30 |
| 92  | xâm lược            | 35 |
| 93  | xây                 | 30 |
| 94  | xe đạp              | 8  |
| 95  | xem                 | 35 |
| 96  | va chạm             | 44 |
| 97  | vận tốc             | 25 |
| 98  | tài năng            | 56 |
| 99  | tạm biệt            | 6  |
| 100 | sách                | 68 |



Một số từ có độ chính xác thấp từ có độ chính xác thấp:

| STT | Tên từ           | Độ chính xác (đơn vị %) |
|-----|------------------|-------------------------|
| 1   | ăn uống          | 1                       |
| 2   | xảy ra           | 1                       |
| 3   | xác chết         | 7                       |
| 4   | màu xanh da trời | 2                       |
| 5   | im lặng          | 8                       |
| 6   | làm ruộng        | 3                       |

#### 1.4 Một số từ dễ bị nhận nhầm:

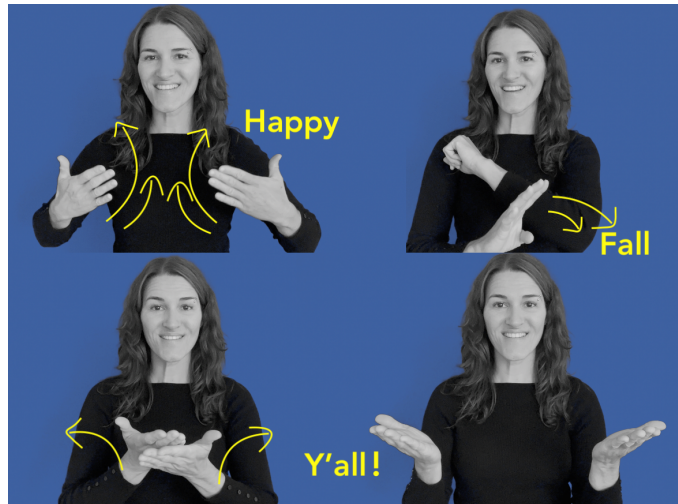
| STT | Tên từ    | Số lần nhận diện sai |
|-----|-----------|----------------------|
| 1   | Xác chết  | 10/15 lần            |
| 2   | Xách nặng | 9/15 lần             |
| 3   | Lãnh đạo  | 9/15 lần             |
| 4   | Đàn ông   | 11/15 lần            |

## 2 Mô tả bài toán:

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 của xã hội.

Nhận dạng ngôn ngữ ký hiệu là một nhiệm vụ tính toán liên quan đến việc nhận dạng các hành động từ ngôn ngữ ký hiệu. Đây là một vấn đề thiết yếu, đặc biệt là trong thế giới kỹ thuật số để thu hẹp khoảng cách giao tiếp mà những người khiếm thính đang phải đối mặt.

Đã có một số tiến bộ trong công nghệ và rất nhiều nghiên cứu đã được thực hiện để giúp đỡ những người khiếm thính. Điều này rất hữu ích cho người khiếm thính trong việc giao tiếp với người khác vì biết ngôn ngữ ký hiệu không phải là điều phổ biến đối với tất cả mọi người, giúp người khiếm thính có thể dễ dàng giao tiếp chỉ bằng cử chỉ tay của họ.



Hình 7: Mô tả ngôn ngữ ký hiệu

### 3 Bài toán:

- **Input:**

Video stream được lấy từ webcam của máy tính của một người đang thực hiện ngôn ngữ ký hiệu.

- **Output:**

































Dòng chữ thể hiện từ ngữ đã được người thực hiện ngôn ngữ ký hiệu thực hiện.

### 4 Mô tả bộ dữ liệu:

Dữ liệu gồm hơn 100 từ ngữ được nhóm tự thu thập và lưu lại.

Mỗi từ ngữ bao gồm 30 video mà mỗi video là 30 frame lưu trữ những keypoint được lấy từ người thực hiện ngôn ngữ ký hiệu.

Nhóm sử dụng code để ghi lại trực tiếp những keypoint dưới dạng file **.npp** của 30 video sau đó file được lưu lại trên máy tính và up lên Google Drive để tiện sử dụng trong chương trình.

|   |   |   |
|---|---|---|
|  alert       |  0 |  0.npy  |
|  antibiotics |  1 |  1.npy  |
|  article     |  2 |  2.npy  |
|  assemble    |  3 |  3.npy  |
|  bag         |  4 |  4.npy  |
|  beggar      |  5 |  5.npy  |
|  bike        |  6 |  6.npy  |
|  black       |  7 |  7.npy  |

Hình 8: Bộ dữ liệu được lưu trên Google Drive

## 5 Phương pháp tiếp cận bài toán:

- Nhận diện mặt, tay và cử chỉ của người dùng:

### *MediaPipe:*

Về tổng quan, MediaPipe là tập hợp của một loạt các giải pháp Machine Learning đa nền tảng, có thể can thiệp được và cực kỳ lightweight. Một số ưu điểm có thể kể tới của giải pháp này bao gồm:

Cung cấp một giải pháp inference nhanh chóng: Google khẳng định rằng bộ công cụ này có thể chạy ổn định trên hầu hết các cấu hình phần cứng thông dụng.

Dễ dàng cài đặt và triển khai: Việc cài đặt cực kỳ dễ dàng và tiện lợi, có thể triển khai trên nhiều nền tảng khác nhau như Mobile (Android/iOS), Desktop/Cloud, Web và IoT devices.

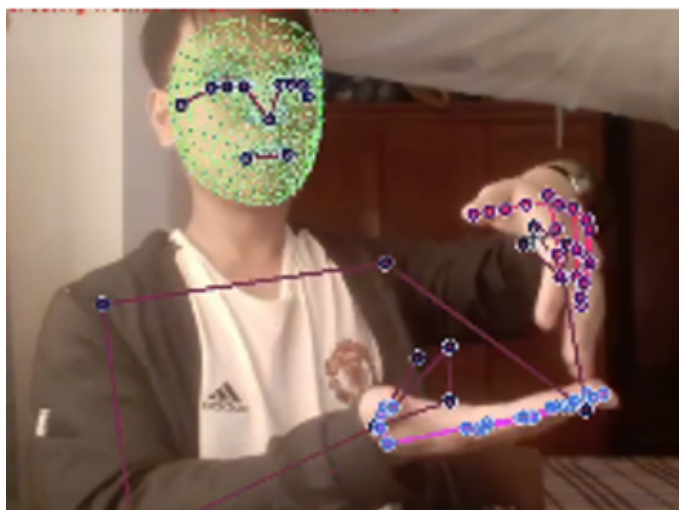
Mã nguồn mở và miễn phí: Toàn bộ source code được công khai trên MediaPipe, người dùng hoàn toàn có thể sử dụng và tùy chỉnh trực tiếp để phù hợp với bài toán của mình.

Bài toán này sử dụng face landmarks detection, hand landmarks detection, pose landmarks detection có sẵn trong MediaPipe để nhận diện mặt, tay, và cử chỉ.

### Face landmarks detection

Với đầu vào là một ảnh hoặc một video, nhiệm vụ của chúng ta là tìm ra vị trí và đóng hộp (bounding box) những khuôn mặt con người xuất hiện trên đây, cũng như đánh dấu các điểm quan trọng (MediaPipe sử dụng 5-landmarks) trên khuôn mặt đó. MediaPipe Face Detection sử dụng mạng BlazeFace làm nền tảng nhưng thay đổi backbones. Ngoài ra, thuật toán NMS (non-maximum suppression) cũng được thay thế bởi một chiến thuật khác, giúp thời gian xử lý giảm đáng kể.

Face Mesh là một bài toán nhận diện một loạt các điểm trên khuôn mặt, từ đó tạo thành 1 lưới (mesh) của mặt. Công cụ trong MediaPipe sẽ sinh ra tổng cộng 468 điểm trên mặt và tạo thành lưới mà không đòi hỏi quá nhiều năng lực tính toán cũng như số camera (chỉ cần 1 camera chính diện - Webcam laptop).

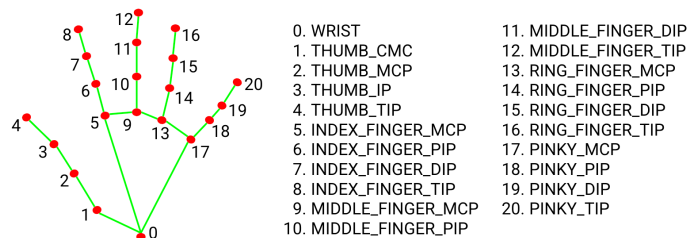


Hình 9: Mô tả Face landmarks detection

### Hand landmarks detection

Đầu ra của giải pháp này là một mô hình skeleton (khung xương) của bàn tay, gồm vị trí của các landmarks trên bàn tay và được nối với nhau thành một khung bàn tay hoàn chỉnh.

Ta xây dựng một mô hình skeleton tay với các điểm như sau:



Hình 10: Skeleton của tay gồm 21 điểm quan trọng trên bàn tay

### Pose landmarks detection

Mở rộng từ bài toán Hands Detection, Pose landmarks detection cung cấp một mô hình skeleton 3D cho cả cơ thể, với các khớp quan trọng được định nghĩa sẵn và được nối với nhau để tạo thành khung của người. Chiến thuật được đặt ra cho bài toán này tương tự như bài Hands landmarks detection và Face landmarks detection.

- **Tiền sử lý dữ liệu, gán nhãn:**

#### *Sklearn:*

Sử dụng **train\_test\_split** từ **Scikit-learn** để chia bộ dữ liệu thành 2 phần, 5% cho tập test và 95% cho tập train.

Scikit-learn ban đầu được đề xuất bởi David Cournapeau trong một dự án mùa hè của Google vào năm 2007.

Later Matthieu Brucher tham gia dự án trên và bắt đầu sử dụng nó làm một phần luận văn tiến sĩ của ông ấy. Vào năm 2010, INRIA bắt đầu tài trợ và phiên bản đầu tiên được xuất bản (v0.1 beta) vào cuối tháng 1 năm 2010.

Dự án vẫn đang được nghiên cứu bởi một đội ngũ hơn 30 nhà nghiên cứu đến từ các công ty lớn INRIA, Google, Tinyclues và Python Software Foundation.

Scikit-learn (Sklearn) là thư viện mạnh mẽ nhất dành cho các thuật toán học máy được viết trên ngôn ngữ Python. Thư viện cung cấp một tập các công cụ xử lý các bài toán machine learning và statistical modeling gồm: classification, regression, clustering, và dimensionality reduction.

#### *Keras:*

Keras là một open source cho Neural Network được viết bởi ngôn ngữ Python. Nó là một library được phát triển vào năm 2005 bởi Francois Chollet, là một kỹ sư nghiên cứu Deep Learning. Keras có thể sử dụng chung với các thư viện nổi tiếng như Tensorflow, CNTK, Theano. Bài toán sử dụng Keras chung với thư viện Tensorflow.

Keras cung cấp thư viện tiện ích numpy, cung cấp các hàm để thực hiện các hành động trên mảng numpy. Sử dụng phương thức `to_categorical`, một mảng numpy có các số nguyên đại diện cho các danh mục khác nhau, có thể được chuyển đổi thành một mảng numpy hoặc một ma trận có các giá trị nhị phân và có các cột bằng số danh mục trong dữ liệu.

### ***Tensorflow:***

TensorFlow 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. Tensorflow là 1 framework được Google xây dựng và ra mắt vào cuối năm 2015. Phiên bản TensorFlow ổn định cuối cùng cũng xuất hiện vào năm 2017. Là mã nguồn mở dưới sự cho phép của Apache Open Source.

## **6 Model Training:**

Xây dựng Sequential model.

Xây dựng LSTM Neural Network có thể xử lý thông tin dạng chuỗi (sequence). Để dự đoán hành động trong video, LSTM Neural Network có thể mang thông tin của frame ảnh từ giai đoạn trước tới các giai đoạn sau, ở giai đoạn cuối là sự kết hợp của tất cả các ảnh để dự đoán hành động trong video.

## **7 Đánh giá model:**

Sử dụng Confusion Matrix để đánh giá Model:

Confusion Matrix là một phương pháp đánh giá kết quả của những bài toán phân loại với việc xem xét cả những chỉ số về độ chính xác và độ bao quát của các dự đoán cho từng lớp. Một Confusion matrix gồm 4 chỉ số sau đối với mỗi lớp phân loại.

Confusion Matrix là một phương pháp đánh giá kết quả của những bài toán phân loại với việc xem xét cả những chỉ số về độ chính xác và độ bao quát của các dự đoán cho từng lớp. Một Confusion matrix gồm 4 chỉ số sau đối với mỗi lớp phân loại.

|                  |              | Actual Values |              |
|------------------|--------------|---------------|--------------|
|                  |              | Positive (1)  | Negative (0) |
| Predicted Values | Positive (1) | TP            | FP           |
|                  | Negative (0) | FN            | TN           |

Hình 11: Bảng Confusion Matrix

Chúng ta có thể suy ra ngay rằng tổng các phần tử trong toàn ma trận này chính là số điểm trong tập kiểm thử. Các phần tử trên đường chéo của ma trận là số điểm được phân loại đúng của mỗi lớp dữ liệu. Từ đây có thể suy ra accuracy chính bằng tổng các phần tử trên đường chéo chia cho tổng các phần tử của toàn ma trận.

Độ chính xác:

$$Acc = \frac{TP + TN}{P + N} = \frac{TP + TN}{TP + FP + TN + FN}$$

Hình 12: Công thức tính độ chính xác

## 8 Ứng dụng thực tế:

Giúp cho những người không biết ngôn ngữ ký hiệu vẫn có thể hiểu được những gì người khiếm thính muốn truyền đạt.

Bài toán này có thể giúp những người khiếm thính tự tin hơn khi họ có thể giao tiếp một cách bình thường với tất cả mọi người. Xóa bỏ khoảng cách giữa người khiếm thính với xã hội.

## 9 Hướng phát triển của đề án:

Tìm hiểu thêm về ngôn ngữ kí hiệu, khảo sát với những người khiếm thính và những người sử dụng ngôn ngữ kí hiệu để xây dựng ứng dụng phù hợp nhất với họ.

Thu thập thêm nhiều dữ liệu.

Tăng độ chính xác của model.

Phát triển thành một ứng dụng hoàn chỉnh có giao diện thân thiện với nhiều đối tượng người dùng để có thể sử dụng trong thực tế.

## 10 Tài liệu tham khảo

TopDev. 2022. Tensorflow là gì?

<https://topdev.vn/blog/tensorflow-la-gi/>

Quang Trần. 2021. MediaPipe: Live ML Solutions và ứng dụng vẽ bằng Hands Gestures.

<https://viblo.asia/p/mediapipe-live-ml-solutions-va-ung-dung-ve-bang-hands-gestures-gAm5ymC>

Nttuan8. 2019. Recurrent neural network

<https://nttuan8.com/bai-13-recurrent-neural-network/>

Nttuan8 . 2019 . Long short term memory (LSTM)

<https://nttuan8.com/bai-14-long-short-term-memory-lstm/>

To Duc Thang. 2020. Làm quen với Keras

<https://viblo.asia/p/lam-quen-voi-keras-gGJ59mxJ5X2>

Nicholas Renotte. 2021. Sign Language Detection using ACTION RECOGNITION with Python | LSTM Deep Learning Model

<https://www.youtube.com/watch?v=doDUihpj6ro>