

TRƯỜNG ĐẠI HỌC MỎ - ĐỊA CHẤT

KHOA CÔNG NGHỆ THÔNG TIN



Học phần : Trí tuệ nhân tạo (7080122)

BÀI THU HOẠCH :

DỰ ÁN MÁY HỌC TẬP

Họ và tên: Nguyễn Văn Cao

Mã sinh viên: 1921050095

Nhóm: 01

Chuyên ngành: Tin địa học

Giảng viên:

Trần Trường Giang

HÀ NỘI - 2022

LỜI MỞ ĐẦU

Trí tuệ nhân tạo là môn học với mục đích chính cung cấp lời giải cho các vấn đề cuộc sống thực tế. Ngày nay, các ứng dụng trí tuệ nhân tạo rất đa dạng và phong phú, áp dụng cho nhiều lĩnh vực như điều khiển tự động, các loại robot, các hệ dịch tự động cho các ngôn ngữ tự nhiên, các nhận dạng, trò chơi điện tử...

Ngày 08 tháng 6 năm 2022 em được nghe giới thiệu về dự án **Máy Học Tập**. Trong quá trình nghe trình bày dự án em rất chú ý đến một vài vấn đề sau: mạng nơ-ron tính chập (Convolution neural network), các bộ thư viện tensorflow, keras, môi trường Anaconda, quá trình tạo dữ liệu huấn luyện, quá trình huấn luyện, quá trình chạy mô hình đã được huấn luyện. Sau đây em xin được trình bày các vấn đề trên.

I. Mô tả về dự án:

1.1. Dự án Máy Học Tập:

- Tên dự án: Máy Học Tập
- Mục đích huấn luyện mô hình học máy: phát triển một ứng dụng có thể nhận diện từ ảnh các cử chỉ tay và chuyển thành các số.

1.2. Xây dựng mô hình Máy Học Tập:

- Mô hình Máy học tập dựa trên mạng nơ-ron tích chập (Convolution Neural Network)
- Mạng nơ-ron tích chập (CNN hay ConvNet) là mạng nơ-ron phổ biến nhất được dùng cho dữ liệu ảnh. Bên cạnh các lớp liên kết đầy đủ (FC layers), CNN còn đi cùng với các lớp ẩn đặc biệt giúp phát hiện và trích xuất những đặc trưng - chi tiết (patterns) xuất hiện trong ảnh gọi là Lớp Tích chập (Convolutional Layers). Chính những lớp tích chập này làm CNN trở nên khác biệt so với mạng nơ-ron truyền thống và hoạt động cực kỳ hiệu quả trong bài toán phân tích ảnh.
- Lớp tích chập (Convolutional Layers)
 - Lớp tích chập được dùng để phát hiện và trích xuất đặc trưng - chi tiết của ảnh.
 - Giống như các lớp ẩn khác, lớp tích chập lấy dữ liệu đầu vào, thực hiện các phép chuyển đổi để tạo ra dữ liệu đầu vào cho lớp kế tiếp (đầu ra của lớp này là đầu vào của lớp sau). Phép biến đổi được sử dụng là phép tính tích chập. Mỗi lớp tích chập chứa một hoặc nhiều bộ lọc - bộ phát hiện đặc trưng (filter - feature detector) cho phép phát hiện và trích xuất những đặc trưng khác nhau của ảnh.
 - Bộ lọc ở lớp tích chập càng sâu thì phát hiện các đặc trưng càng phức tạp.
 - Độ phức tạp của đặc trưng được phát hiện bởi bộ lọc tỉ lệ thuận với độ sâu của lớp tích chập mà nó thuộc về. Trong mạng CNN, những lớp tích chập đầu tiên sử dụng bộ lọc hình

học (geometric filters) để phát hiện những đặc trưng đơn giản như cạnh ngang, dọc, chéo của bức ảnh. Những lớp tích chập sau đó được dùng để phát hiện đối tượng nhỏ, bán hoàn chỉnh như mắt, mũi, tóc, v.v. Những lớp tích chập sâu nhất dùng để phát hiện đối tượng hoàn hình như: chó, mèo, chim, ô tô, đèn giao thông, v.v. Để hiểu cách thức hoạt động của lớp tích chập cũng như phép tính tích chập, hãy cùng xem ví dụ về bộ lọc phát hiện cạnh (edge filters/ detectors) dưới đây.

1.3. Các bộ thư viện :

Mô hình Máy Học Tập sử dụng hai bộ thư viện chính là: tensorflow và keras

- Thư viện tensorflow: Thư viện TensorFlow là thư viện mã nguồn mở dùng cho tính toán số học sử dụng đồ thị luồng dữ liệu. Thư viện TensorFlow tích hợp sẵn rất nhiều các thư viện machine learning. Đồng thời TensorFlow có khả năng tương thích và mở rộng tốt. Điều này được Google phát triển cho học máy phục vụ cả nguyên cứu lẫn xây dựng các ứng dụng thực tế. Kiến trúc của Tensorflow hoạt động được chia thành 3 phần. Đó là tiền xử lý dữ liệu, dựng model, train và ước tính model.
 - Tensor : Tensor là được đưa ra trực tiếp nhờ vào framework cốt lõi của TensorFlow. Trong TensorFlow, tất cả các tính toán đều liên quan đến các tensor. Một tensor là một vector hay ma trận của n-chiều không gian đại diện cho tất cả loại dữ liệu. Tensor có 3 thuộc tính cơ bản là rank, shape và type.
 - Rank: Rank là số bậc của tensor. Việc phân rank này khá quan trọng vì nó giúp phân loại dữ liệu của tensor. Khi các rank đặc biệt cụ thể, tensor sẽ có những tên gọi riêng như:
 - Scalar: Khi Tensor có rank bằng 0.
 - Vector: Vector là một Tensor rank 1.
 - Matrix: Đây là một Tensor rank 2 hay mảng hai chiều theo khái niệm của Python.

- N-Tensor: Khi rank của Tensor tăng lên lớn hơn 2, chúng được gọi chung là N-Tensor.
- Shape: Shape của dữ liệu chính là chiều của ma trận hay mảng. Tất cả các giá trị trong một tensor chứa đựng loại dữ liệu giống hệt nhau với 1 shape đã biết hoặc đã biết một phần.
- Type: Type là kiểu dữ liệu của các elements trong Tensor. Một Tensor chỉ có duy nhất một thuộc tính Type. Do đó, chỉ có một kiểu Type duy nhất cho toàn bộ các elements có trong Tensor hiện tại.
- Thư viện 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. Một số ưu điểm của Keras như:
 - Dễ sử dụng, dùng đơn giản hơn Tensor, xây dựng model nhanh.
 - Run được trên cả CPU và GPU.
 - Hỗ trợ xây dựng CNN , RNN hoặc cả hai. Với những người mới tiếp cận đến Deep như mình thì mình chọn sử dụng Keras để build model vì nó đơn giản, dễ nắm bắt hơn các thư viện khác. Dưới đây mình xin giới thiệu một chút về API này.

1.4. Môi trường Anaconda:

Anaconda là một nền tảng mã nguồn (open source platform) về khoa học dữ liệu (data science). Anaconda bao gồm các gói (packages), các thư viện (libraries) mã nguồn mở viết bằng Python và R. Đây là bộ công cụ dành cho các lập trình viên, chuyên gia phân tích dữ liệu, thậm chí là chuyên viên kinh tế - tài chính với công việc phân tích dữ liệu.

Các chương trình phân tích dữ liệu phát triển bởi Python Anaconda có tính linh hoạt rất cao. Do sử dụng Python nên có thể chạy trên bất cứ nền tảng nào: Windows, Linux, Mac OS. Thêm vào đó, do tận dụng được thế mạnh Virtual Environment của ngôn ngữ python, các môi trường phân tích dữ liệu có thể hoạt động đồng thời mà không ảnh hưởng lẫn nhau.

II. Mô tả về quá trình tạo dữ liệu huấn luyện

– Cách tạo thư mục chứa dữ liệu huấn luyện:

- Tạo thư mục **data** để chứa tất cả dữ liệu huấn luyện:

op > BTL_Project

Name	Date modified	Type	Size
data	10/06/2022 11:55 SA	File folder	
BTL_CNN_model.h5	16/06/2022 9:17 SA	H5 File	1.657 KB
gesture_data	10/06/2022 12:04 CH	Python File	5 KB
gesture_model	16/06/2022 9:20 SA	Python File	4 KB
gesture_training	10/06/2022 12:03 CH	Python File	6 KB

- Trong thư mục **data** tạo hai thư mục **train** và **test**:

- Thư mục train lưu dữ liệu huấn luyện
- Thư mục test lưu dữ liệu để kiểm tra

Name	Date modified	Type	Size
test	10/06/2022 11:55 SA	File folder	
train	10/06/2022 11:55 SA	File folder	

- Trong hai thư mục tạo các thư mục từ 1 → 10 lưu các dữ liệu cử chỉ tay tương ứng

1	10/06/2022 11:55 SA	File folder
2	10/06/2022 11:55 SA	File folder
3	10/06/2022 11:55 SA	File folder
4	10/06/2022 11:55 SA	File folder
5	10/06/2022 4:53 SA	File folder
6	10/06/2022 4:53 SA	File folder
7	10/06/2022 4:53 SA	File folder
8	10/06/2022 4:53 SA	File folder
9	10/06/2022 4:53 SA	File folder
10	10/06/2022 12:42 CH	File folder

– Cách chạy file huấn luyện:

- Bước 1: khởi động **Anaconda**

- Bước 2: chọn **Spyder** (lý do chọn: **Spyder** là một môi trường do Anaconda cung cấp được cài sẵn các thư viện của Python và có giao diện dễ nhìn giúp ta có thể chạy trực tiếp code trên **Spyder**)
- Bước 3: Chọn file `gesture_data.py` → bấm Run.

III. Mô tả quá trình huấn luyện

3.1. Kí hiệu cử chỉ của các số:

Số 1:



Số 2:



Số 3:



Số 4:



Số 5:



Số 6:



Số 7:



Số 8:



Số 9:



Số 10:



3.2. Các bước huấn luyện

- Bước 1: Vào **Anaconda** → vào **Spyder** → chọn file **gesture_data.py**
- Bước 2: Chỉnh sửa các thông số:
 - Điều chỉnh số tương ứng với chỉ tay mà bạn muốn huấn luyện

```
cam = cv2.VideoCapture(0)

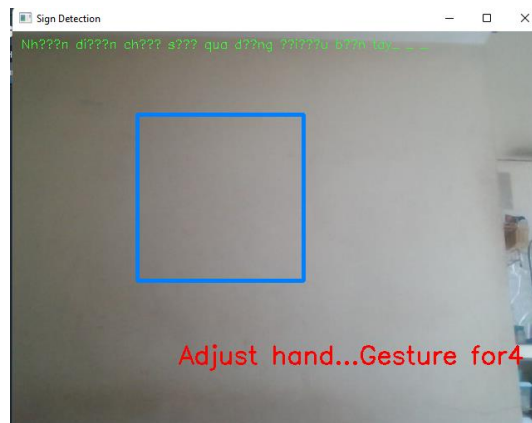
num_frames = 0
element = 4  ##--> chữ số tương ứng với dáng điệu tay
num_imgs_taken = 0

while True:
    ret, frame = cam.read()
```

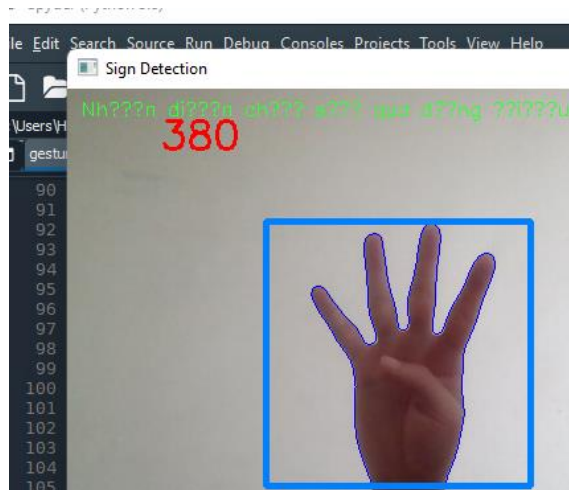
- Điều chỉnh đường dẫn đến thư mục chứa dữ liệu tương ứng

```
cv2.imwrite(r"C:\Users\HP\Desktop\BTL_Project\data\train\4"+"\" +
else:
    break
num_imgs_taken +=1
```

- Bước 3: Sau khi chỉnh sửa ta bắt đầu chạy file, cửa sổ camera được bật lên



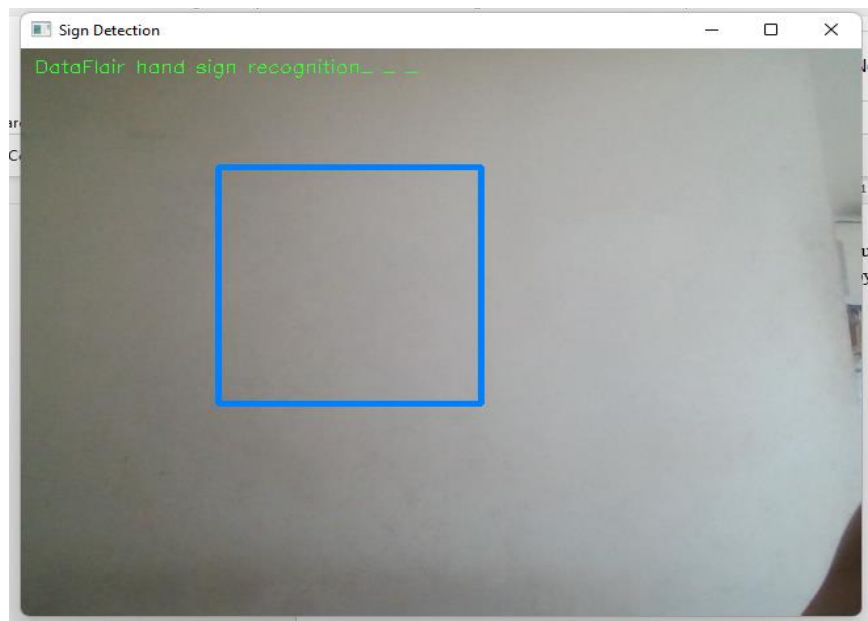
- Đưa cử chỉ tay tương ứng với số mà mình muốn huấn luyện vào trong ô vuông



- Điều chỉnh các góc độ tay để dễ quét. Sau khi quét xong 300 ảnh thì ứng dụng dừng lại và ta đã huấn luyện xong một cử chỉ. Các cử chỉ khác làm tương tự

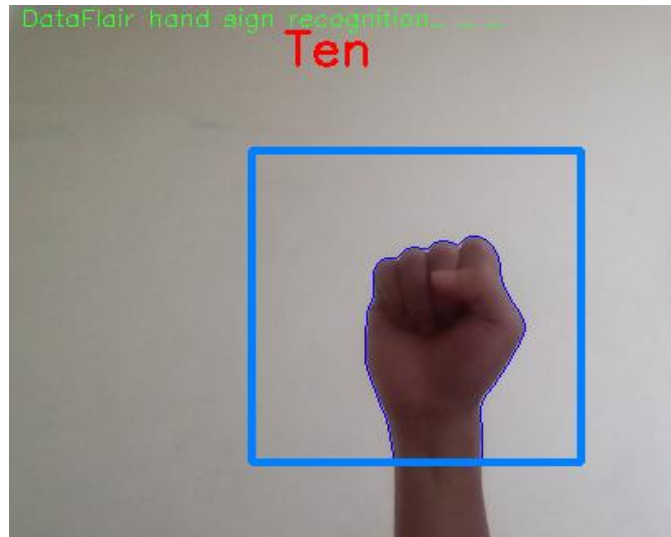
IV. Mô tả quá trình chạy mô hình đã được huấn luyện

- Bước 1: vào **Anaconda** → mở **Spyder** → chọn file **gesture_mode.py** → bấm Run → cửa sổ camera hiện ra

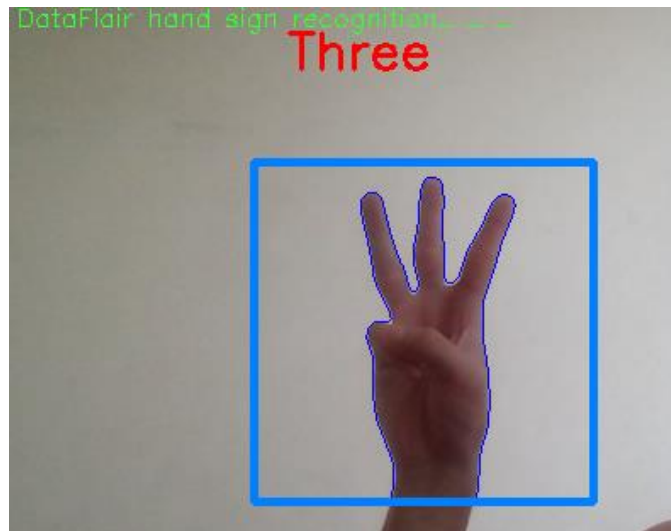


- Bước 2: Đưa cử chỉ tay vào khung ô vuông, chương trình quét và đưa ra số mà cử chỉ tay thể hiện

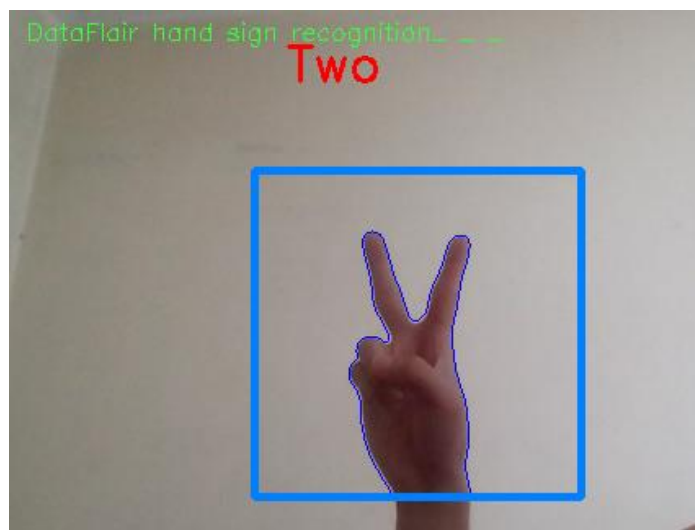
- Số 10:



- Số 3:



- Số 2:



V. Nhận xét của bản thân

Sau khi được nghe giới thiệu và chạy thử dự án Máy Học Tập em đã hiểu rõ hơn về một dự án trí tuệ nhân tạo được xây dựng và vận hành như thế nào, được củng cố kiến thức về mạng nơ-ron tích chập (Convolution Neural Network), làm quen với các bộ thư tensorflow, keras của python, môi trường Anaconda. Trong tương lai em sẽ luôn cố gắng trau dồi kiến thức để phát triển bản thân và có thể tạo ra một mô hình trí tuệ nhân tạo của riêng mình.