

#### A. Thông tin cá nhân

Họ và tên: Vũ Hoàng Giang

Ngày tháng năm sinh: 05/11/2003

Địa chỉ: Cụm 9, xã Ngọc Hồi, TP Hà Nội

Trường: Đại học Xây dựng Hà Nội

Số điện thoại: 0963320223

Email: [gvu0326@gmail.com](mailto:gvu0326@gmail.com)

#### B. Trả lời câu hỏi

- GITHUB: <https://github.com/hjiang511/Ai4li-a.git>

### Phần 1: Xử lý ảnh/video

#### Câu 1:

1.1 Chương trình trích xuất tọa độ các điểm khớp của bàn tay qua từng frame và lưu thành file JSON

Refer: <https://github.com/Hzzone/pytorch-openpose/tree/master>

Chương trình sử dụng 2 model: recognize\_hand và recognize\_body để nhận diện hand bone. Qua quá trình test, chương trình chỉ chạy tốt nếu chạy cả 2 model.

#### 🔥 Lớp Hand – Bộ xử lý phát hiện bàn tay

##### 1 Nhận ảnh gốc (BGR)

- Resize ảnh sang nhiều tỉ lệ (scale\_search), chuẩn hóa về kích thước chuẩn (boxsize = 368), và pad cho chia hết stride (8).

##### 2 Chạy model handpose\_model

- Model sinh ra heatmap có 22 kênh (21 kênh cho từng keypoint + 1 background).

##### 3 Xử lý heatmap

- Áp dụng Gaussian filter để làm mượt.
- Với mỗi keypoint (0→20):
  - Lấy kênh heatmap tương ứng.
  - Xác định điểm cực đại (pixel có giá trị cao nhất).
  - Nếu điểm đó < ngưỡng thre → bỏ qua, gán [0,0].

##### 4 Tạo mảng kết quả

- Kết quả cuối cùng: mảng 21 điểm (x,y).

- Mỗi bàn tay có **21 keypoint** theo cấu trúc:
  - **0:** cổ tay
  - **1-4:** ngón cái
  - **5-8:** ngón trỏ
  - **9-12:** ngón giữa
  - **13-16:** ngón áp út
  - **17-20:** ngón út

### Lý do 21 điểm?

👉 Vì bàn tay có 1 cổ tay + 5 ngón × 4 khớp = **21 điểm**.

Mô hình handpose\_model học theo cấu trúc chuẩn này (giống OpenPose Hand).

### Lớp Body – Bộ xử lý phát hiện cơ thể (Body Pose Estimation)

### Pipeline hoạt động:

#### 1 Nhận ảnh gốc (BGR)

- Ảnh được resize theo nhiều tỉ lệ (scale\_search) và pad để phù hợp với kích thước chuẩn (boxsize = 368) và stride (8).

#### 2 Chạy model bodypose\_model

- Model sinh ra:
  - **Heatmap (19 kênh):**
    - 18 kênh cho 18 keypoint cơ thể (mắt, tai, vai, khuỷu tay, cổ tay, hông, gối, mắt cá, ...).
    - 1 kênh background.
  - **PAF (Part Affinity Fields – 38 kênh):** biểu diễn hướng và kết nối giữa các cặp khớp.

#### 3 Xử lý heatmap

- Làm mượt bằng Gaussian filter.
- Tìm điểm cực đại trên từng heatmap → lấy **toạ độ + điểm số** cho từng keypoint.

#### 4 Phân tích PAF (Part Affinity Fields)

- Dựa trên PAF để “nối” các điểm keypoint thành bộ xương hoàn chỉnh.
- **LimbSeq** định nghĩa các cặp điểm phải nối (VD: vai – khuỷu tay, khuỷu tay – cổ tay).
- Tính toán vector hướng trên PAF để chọn kết nối có xác suất cao nhất.

## 5 Tạo kết quả cuối cùng

- **Candidate:** Danh sách tất cả các keypoint phát hiện được [x, y, score, id].
- **Subset:** Ma trận biểu diễn từng người trong ảnh:
  - Cột 0–17: index keypoint (tương ứng với Candidate).
  - Cột 18: tổng điểm tin cậy.
  - Cột 19: số keypoint phát hiện.

### ✓ Lý do 18 điểm?

- 0-1: Mũi
- 2-3: Mắt trái, mắt phải
- 4-5: Tai trái, tai phải
- 6-7: Vai trái, vai phải
- 8-9: Khuỷu tay trái, khuỷu tay phải
- 10-11: Cổ tay trái, cổ tay phải
- 12-13: Hông trái, hông phải
- 14-15: Gối trái, gối phải
- 16-17: Mắt cá trái, mắt cá phải

### 🔥 Demo:

Input: Video .mp4

Output: file json, lưu lại tọa độ của khớp tay qua từng frame.

Example:



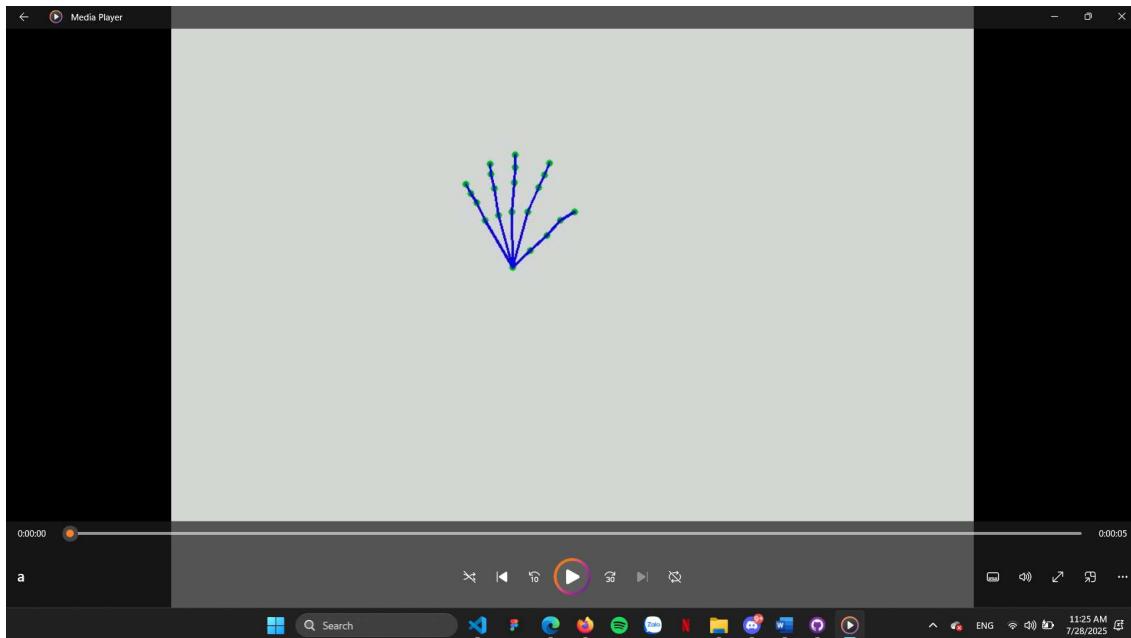
Orginal.mp4



Output.mp4

## 1.2 Mô phỏng lại cử động từ file JSON

convert.py: Lấy tọa độ từ 21 khớp tay trong file json để vẽ lại theo từng frame



*Convert.mp4*

Câu 2: Xây dựng model machine learning để phân loại 5-10 cử chỉ cơ bản

### 2.1 Dataset:

- Kích thước dataset: (631, 44)
- Cột: ['frame\_index', 'x0', 'y0', 'x2', 'y1', 'x4', 'y2', 'x6', 'y3', 'x8', 'y4', 'x10', 'y5', 'x12', 'y6', 'x14', 'y7', 'x16', 'y8', 'x18', 'y9', 'x20', 'y10', 'x22', 'y11', 'x24', 'y12', 'x26', 'y13', 'x28', 'y14', 'x30', 'y15', 'x32', 'y16', 'x34', 'y17', 'x36', 'y18', 'x38', 'y19', 'x40', 'y20', 'label']
- Label : q (103), r (91), t (80), a (78), e (71), c (61), d (58), b (53), đ (36)

### 2.2 Model :

- Chuyển đổi dữ liệu từ dạng “frame rời rạc” thành “chuỗi nhiều frame liên tiếp, ở đây sequence\_length bằng 10, nghĩa là 10 frame mỗi chuỗi.
- Thông số model :

Layer (type)	Output Shape	Param #
lstm_3 (LSTM)	(None, 64)	27,392
dropout_3 (Dropout)	(None, 64)	0
dense_6 (Dense)	(None, 64)	4,160
dense_7 (Dense)	(None, 9)	585

- Đánh giá model :

Nhãn	Precision	Recall	F1-score	Ghi chú
a,b,c,e,q,t,đ	1.00	1.00	1.00	Hoàn hảo
d	1.00	0.90	0.95	Một vài frame nhãn “d” bị dự đoán nhầm
r	0.94	1.00	0.97	Precision hơi thấp do có vài dự đoán nhầm thành nhãn khác

- Accuracy tổng thể:** 99% – cực cao, mô hình LSTM học rất tốt từ dữ liệu.
- Macro avg & Weighted avg** đều đạt 0.99 → model cân bằng cho tất cả nhãn.

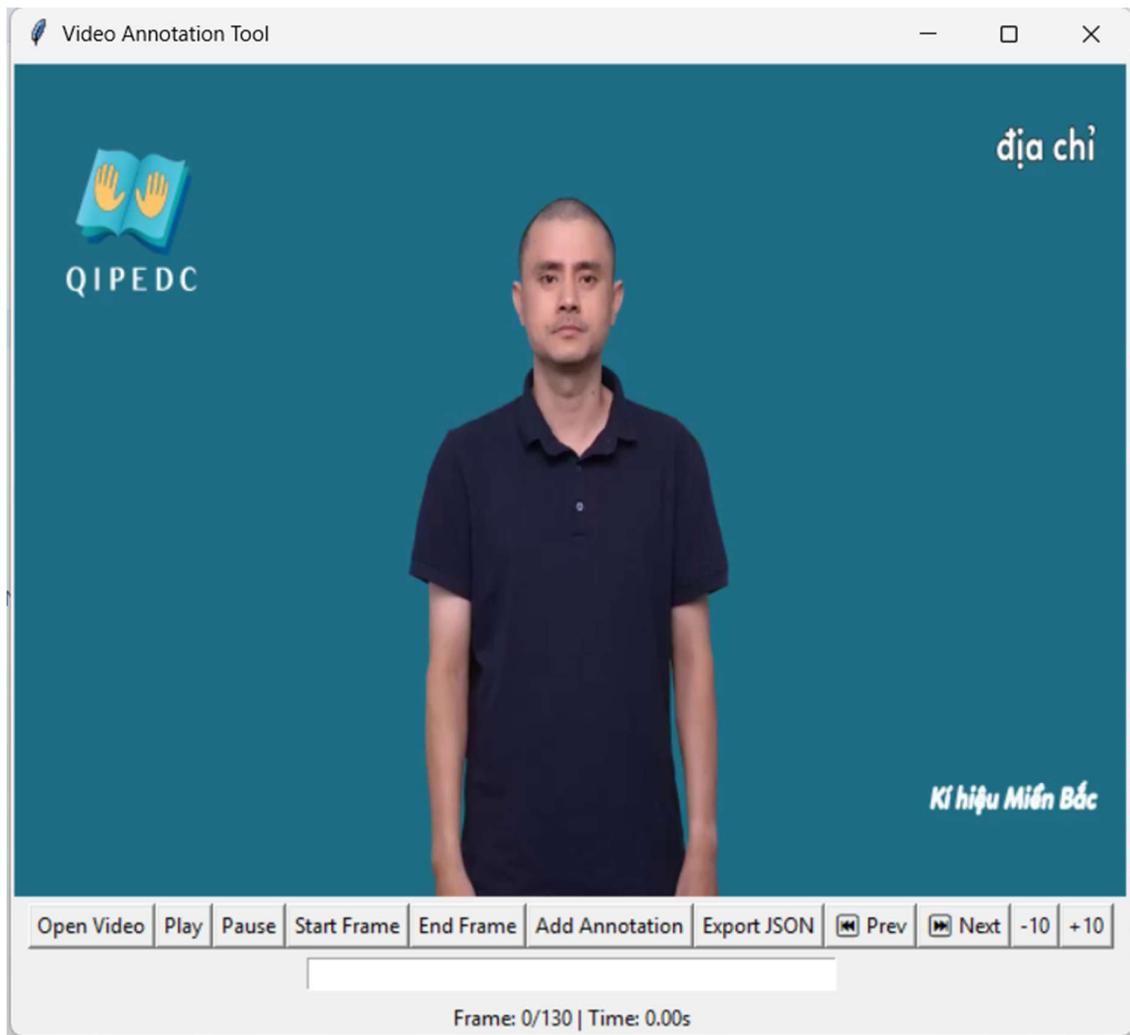
### Phần 3: Xây dựng và xử lý dữ liệu

Câu 1: Thu thập ít nhất 50 video ngôn ngữ ký hiệu tiếng Việt từ các sources (VTV, YouTube...) và tạo metadata chi tiết cho từng video.

- Data thu thập: video mp4
- Metadata:
  - Các trường: video\_id, file\_name, duration, frame\_count, fps, resolution, source\_url, signer\_gender, note
  - Số lượng: 51 video

Câu 2: Xây dựng một annotation tool (web hoặc desktop app) cho phép gán nhãn video ngôn ngữ ký hiệu, bao gồm tính năng play/pause, di chuyển frame, nhập text translation.

- Desktop app: PyQt5
- Chức năng:
  - Tải lên video
  - Play/Pause
  - StartFrame/EndFrame
  - Thêm chú thích
  - Lưu chú thích
  - Xuất ra file JSON
  - Di chuyển giữa các frame (1 hoặc 10)
- Giao diện:



- Lưu ý:
  - Chức năng di chuyển giữa các frame chỉ hoạt động khi pause video.
  - Cấu trúc Annotation: StartFrame, EndFrame, Text
  - Add Annotation là add vào file tạm, Export Json sẽ xuất file json với tất cả annotation.

#### Phần 4:

Câu 1: Xây dựng pipeline xử lý văn bản tiếng Việt bao gồm tokenization, POS tagging, và xử lý dấu thanh điệu. Test với các câu mẫu.

- Thư viện sử dụng: Underthesea
- Pipeline:
  - **Chuẩn hóa Unicode:** Bảo đảm văn bản không bị lỗi ký tự.
  - Tokenization: Dùng **underthesea** để tách từ.
  - POS Tagging: Gán **nhãn từ loại** cho từng từ.
  - Chuẩn hóa dấu thanh (demo)
- Example:

- Input: “Hôm nay trời đẹp và tôi đi dạo công viên.”
- Output:

**Bước 1: Normalize Unicode:**

Hôm nay trời đẹp và tôi đi dạo công viên.

**Bước 2: Tokenization:**

Hôm\_nay trời đẹp và tôi đi dạo công\_viên .

**Bước 3: POS Tagging:**

Hôm nay => N

trời => N

đẹp => A

và => C

tôi => P

đi => V

dạo => V

công viên => N

. => CH

**Bước 4: Chuẩn hoá dấu thanh (demo):**

Hôm nay trời đẹp và tôi đi dạo công viên.

Câu 2: Phân tích sự khác biệt về cấu trúc ngữ pháp giữa tiếng Việt và ngôn ngữ ký hiệu. Xây dựng hệ thống chuyển đổi câu tiếng Việt sang cấu trúc ngôn ngữ ký hiệu.

#### REFERENCE:

[https://www.google.com/url?sa=t&source=web&rct=j&opi=89978449&url=https://www.hufofcw.org/Download/file/17875&ved=2ahUKEwiLg\\_k\\_9WOAxXImq8BHWCDMHEQFnoECB4QAQ&usg=AOvVaw1cetUDf7e\\_1gthRXKvIgvk](https://www.google.com/url?sa=t&source=web&rct=j&opi=89978449&url=https://www.hufofcw.org/Download/file/17875&ved=2ahUKEwiLg_k_9WOAxXImq8BHWCDMHEQFnoECB4QAQ&usg=AOvVaw1cetUDf7e_1gthRXKvIgvk)

### 2.1 Cấu trúc câu trong ngôn ngữ ký hiệu

- **Câu đơn:**

- Trong ngôn ngữ nói: **Chủ ngữ + Động từ + Bổ ngữ**
- Trong ngôn ngữ ký hiệu: **Chủ ngữ + Bổ ngữ + Động từ**

**Ví dụ:**

- Nói: “Tôi thích màu đỏ”
- Ký hiệu: Tôi / màu đỏ / thích

### 2.2 Cấu trúc cụm từ: Số từ + Danh từ

- Trong ngôn ngữ nói: **số từ + danh từ**
- Trong ngôn ngữ ký hiệu: **danh từ + số từ**

**Ví dụ:**

- Nói: “4 sinh viên”
- Ký hiệu: sinh viên / 4

### 2.3 Từ ghép

- Giống ngôn ngữ nói: phát âm **từng tiếng**.
- Nhưng **kí hiệu = lắp ghép** các ký hiệu thành phần.

**Ví dụ:**

- Hoa hồng → “hoa” + “hồng”
- Uống sữa → “uống” + “sữa”

#### 2.4 Câu hỏi

- **Từ nghi vấn** được đặt sau cùng.

**Ví dụ:**

- Nói: “Gia đình bạn có bao nhiêu người?”
- Ký hiệu: Bạn / gia đình / người / bao nhiêu?

#### 2.5 Câu phủ định

- Trong ngôn ngữ nói: **Từ phủ định + động từ**
- Trong ngôn ngữ ký hiệu: **Từ phủ định đặt cuối câu**

**Ví dụ:**

- Nói: “Tôi không thích màu đỏ”
- Ký hiệu: Tôi / màu đỏ / thích / không

#### 2.6 Tính từ & từ láy

- Chỉ giữ **nghĩa gốc**, bỏ phần láy.

**Ví dụ:**

- Nói: “vui vẻ” → trong ký hiệu chỉ ký hiệu “vui”.

#### 2.7 Kí hiệu hành động

- **Hướng tay xác định ai là chủ thể hành động:**

- Tay hướng ra ngoài → người ký hiệu là **chủ thể**.
- Tay hướng vào trong → người ký hiệu là **đối tượng chịu tác động**.

#### 2.8 Rút gọn ký hiệu

- **Bỏ bớt thành phần** không cần thiết trong câu.
- **Kết hợp ký hiệu** (tạo động tác liên tục).
- **Dùng cả hai tay** để tăng tốc độ truyền đạt.

2.9 Xây dựng hệ thống chuyển đổi câu tiếng Việt sang ngôn ngữ ký hiệu.

\* Input: Câu tiếng việt

\* Output: Câu dưới dạng ngôn ngữ ký hiệu

\* Pipeline:

(a) POS\_TAGS\_DESCRIPTION: Bảng giải thích các thẻ từ loại (POS tags) mà thư viện

**Underthesea** trả về.

(b) Hàm normalize\_tone(): Chuyển văn bản tiếng Việt sang dạng **không dấu** (bỏ hết thanh sắc, huyền, hỏi, ngã, nặng).

(c) Hàm convert\_to\_sign\_structure(): Nhận **POS Tag** (từ loại) cho từng từ; áp dụng **quy tắc chuyển đổi sang ký hiệu** dựa trên tài liệu HUFO

\* Example:

