Vietnamese Sign Language detection

Nguyễn Công Nguyên, Trương Lê Minh Hiếu

Trường Đại học Công nghệ Thông tin, Thành phố Hồ Chí Minh, Việt Nam

Đại học Quốc gia Hồ Chí Minh, VNU-HCM

{21521200, 21522079}@gm.uit.edu.vn

22/1/2023

Summarize

Nhóm quyết định thực hiện chủ đề Sign Language Detection với dataset custom mà nhóm thực hiện, dataset gồm 500 từ ngữ, chứa hơn 3000 video được lấy từ 2 nguồn chính: Youtube và tự quay, mỗi folder từ ngữ phải có tối thiểu 5 video.

Đề tài được thực hiện dựa trên paper "Word-level Deep Sign Language Recognition from Video: A New Large-scale Dataset and Methods Comparison" (Dongxu Li, 2020), model được thực hiện dựa trên một trong 4 cách thức mà paper thực hiện và phần thực hiện code nhóm tham khảo từ video "Sign Language Detection using ACTION RECOGNITION with Python | LSTM Deep Learning Model" (Renotte, 2021), với ý tưởng chủ đạo gồm:

1. Tăng cường video dùng để train bằng cách phương pháp shear, translate, rotate (Tung, 2022)
2. Thực hiện remove background bằng cách crop video sao cho chỉ lấy phần người thực hiện là chủ yếu để tăng độ chính xác của model, nhóm sử dụng YOLOv5 để lấy bounding box của người thực hiện rồi crop video dựa vào bounding box đó
3. Các video đã remove background được đưa vào các model MediaPipe của Google (MediaPipe BlazePose GHUM 3D, n.d.), (Model Card Hand Tracking (Lite/Full) with Fairness Oct 2021, n.d.) để trích xuất các keypoint và lưu lại
4. Đưa thông tin các keypoint vào một mạng LSTM xếp chồng 2 lớp để train
5. Sử dụng top-K để đánh giá model và nhận xét

Link:

https://github.com/CongNguyenGitHub/CS114.O11-21521200/tree/main/Final%20Project

Chương 0

1. So với trong (Dongxu Li, 2020), nhóm sử dụng LSTM thay vì GRU. Hiện nay nhiều mô hình chuộng việc sử dụng LSTM hơn so với GRU do LSTM thường cho kết quả cao hơn. Trong đề tài của nhóm, việc sử dụng LSTM được kết quả tốt hơn so với dùng GRU tuy sự cải thiện ấy là không quá đáng kể.

Kết quả khi thực hiện với GRUs:

Method	Top-1	Top-5	Top-10
Custom model	0.075	0.18	0.236

2. Nhóm đã thực hiện việc chỉnh sửa và bổ sung thêm các trích dẫn các bài báo (nguồn tham khảo) cụ thể vào từng phần của bài báo cáo, như đã được yêu cầu bởi giáo viên.

Introduction:

Không phải ai sinh ra cũng có thể lành lặn như mọi người, nói chuyện để giao tiếp có thể là một điều hiển nhiên với người này nhưng cũng có thể là một điều bất khả thi với một người kém may mắn. Với những người khiếm thính, họ phải học một ngôn ngữ kí hiệu để có thể giao tiếp được với những người khác, tuy nhiên nếu muốn hiểu họ đang nói gì thì người giao tiếp với họ cũng phải học ngôn ngữ kí hiệu, và do rất ít khi tiếp xúc với những người khiếm thính nên hầu hết mọi người đều không có tìm hiểu về ngôn ngữ kí hiệu.

Ngôn ngữ ký hiệu (NNKH) là một hình thức ngôn ngữ tư duy, được sử dụng để giao tiếp trong cộng đồng người điếc, có nhiều biến thể tuỳ thuộc vào từng quốc gia, vùng miền. NNKH cho tiếng Việt được coi là một ngôn ngữ riêng biệt, có nhiều biểu hiện riêng biệt nhưng cũng chia sẻ những nguyên tắc riêng. Hơn nữa, nhu cầu hỗ trợ cho việc giao tiếp của người điếc ở Việt Nam là rất lớn. Theo kết quả từ cuộc điều tra tổng điều tra dân số và nhà ở Việt Nam năm 2009, nước ta có khoảng 6,7 triệu người khuyết tật, chiếm khoảng 7,8% tổng số dân số, trong đó có 2,5 triệu người điếc. Đề tài của bọn em sẽ vận dụng kiến thức machine learning hay cụ thể hơn là deep learning để giải quyết một phần trong giao tiếp giữa những người khiếm thính và những người không biết ngôn ngữ kí hiệu – Bài toán Sign Language Detection (SLD).

1. Bọn em đã xây dựng một bộ dữ liệu riêng cho bài toán (Những yêu cầu khi xây dựng bộ dữ liệu được miêu tả chi tiết ở phần tiếp theo)
2. Bọn em sử dụng những kĩ thuật tiền xử lý raw data tự thu được để tăng cường bộ dữ liệu của mình cũng như để có thể thao tác dễ dàng hơn
3. Bọn em kết hợp nhiều phương pháp trong machine learning cũng như deep learning và cả computer vision để thực hiện bài toán.

Những lợi ích của đề tài:

- Giao tiếp: Giúp giao tiếp với người khiếm thính kể cả khi không biết ngôn ngữ kí hiệu
- Hỗ trợ giáo dục: Công nghệ này có thể áp dụng vào giáo dục để hỗ trợ thầy cô giảng dạy với những học sinh khiếm thính. Nó còn giúp học sinh khiếm thính và giáo viên tương tác với nhau hiệu quả hơn
- Hoà nhập: Giúp hoà nhập người khiếm thính với cộng đồng hơn
- Tiện ích: Giống như chức năng nhận diện giọng nói để tra cứu tiện hơn của Google, có thể mở rộng một ứng dụng nhận diện ngôn ngữ kí hiệu để tiện lợi hơn, không phải gõ chữ…
- Phát triển ứng dụng AI: Việc phát triển các thuật toán và mô hình AI để nhận diện ngôn ngữ ký hiệu có thể đóng góp vào việc nghiên cứu và phát triển hệ thống trí tuệ nhân tạo, mở ra cánh cửa cho các ứng dụng AI khác

Fundamental of Sign Language Detection on Custom data

Introduction to SLD task

Bài toán này được thực hiện nhằm xác định được ý nghĩa của động tác mà người thực hiện muốn truyền đạt. Mô tả bài toán cụ thể như sau:

Input:

- Dữ liệu đầu vào là hình ảnh từ camera, có thể chứa người, tay, khuôn mặt, và các yếu tố môi trường khác.
- Các video, frame ảnh này có thể có độ phân giải, kích thước, và tỉ lệ khung hình khác nhau.
- Phải được quay chính diện, khoảng cách từ 30-50 cm
- Có ánh sáng tốt, đến mức nhìn thấy được cử chỉ của bàn tay
- Độ phân giải tối thiểu 480p

Output:

- Nếu xác định được từ ngữ của frame ảnh đó, ứng dụng sẽ trả về từ ngữ của frame ảnh đó
- Một cử chỉ có thể có ý nghĩa của một từ (vd: cá, chim) nhưng cũng có những từ cần nhiều cử chỉ để biểu hiện (vd: thầy giáo, hiệu trưởng,…)

Related works:

Đây không phải là một chủ đề mới lạ, trước đó đã có nhiều tiền nhân thực hiện với những phương pháp trước nhằm giải quyết bài toán. Tuy nhiên, việc giải quyết trên ngôn ngữ Tiếng Việt vẫn còn nhiều khiêm tốn, với chủ yếu xây dựng dựa trên các tập dữ liệu Tiếng Anh nổi tiếng như Purdue RVL-SLLL ASL Database, Boston ASLLVD, RWTH-BOSTON-50.

"Real-Time Sign Language Detection Using CNN" (Saiful, 2022) là cách giải quyết cơ bản nhất với bài toán SLD. Tuy nhiên, phương pháp này đã trở nên lỗi thời và độ chính xác khi dự đoán cũng không cao. Trong điều kiện công nghệ và nhu cầu hiện nay thì hướng tiếp cận sử dụng mạng Convolutional Neural Network tuy vẫn khả thi nhưng không được đánh giá cao.

Lấy cảm hứng từ paper "Word-level Deep Sign Language Recognition from Video: A New Large-scale Dataset and Methods Comparison" (Dongxu Li, 2020), bọn em đã cải tiến và sử dụng những công cụ mới hơn hiện nay như MediaPipe nhằm tìm hiểu sâu hơn về bài toán SLD trong bộ dữ liệu Tiếng Việt mà bọn em xây dựng.

Dataset creation

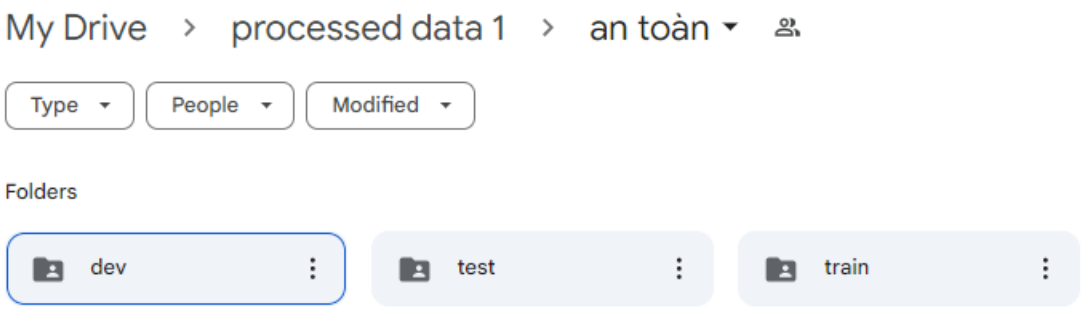
Data preparation

Bộ dataset mà nhóm tự xây dựng có 2 nguồn chính: Youtube và tự quay. Với mục tiêu là 5 video mỗi từ vựng, bọn em cắt ra từ Youtube những video mà đạt yêu cầu kể trên, thông thường

một từ vựng kiếm thì kiếm được từ Youtube 3 video, phần còn lại chia ra nhóm tự quay thoả những yêu cầu trên. Những video kiếm được đều từ những nguồn, những lớp dạy ngôn ngữ kí hiệu Tiếng Việt uy tín, có học viên tham gia đầy đủ. Những video từ ngữ bọn em thu thập được đều là những từ ngữ gần guỗi, quen thuộc với đời sống hằng ngày, phủ đều nhiều chủ đề khác nhau, từ gia đình, công việc, nhà trường cho đến quê hương đất nước,…

Annotation guidelines

Để tránh việc trùng lặp cũng như là dễ dàng hơn trong việc tìm kiếm và sửa chữa sai sót trong bộ dữ liệu, bọn em quyết định mỗi từ ngữ là một nhãn riêng, một folder riêng. Bên trong folder từ ngữ sẽ gồm 3 label chính: train, test, dev. Những từ ngữ không có đủ 5 video hoặc hơn sẽ bị loại bỏ. Những từ ngữ đó thường là những từ ngữ không thường xuyên dùng trong sinh hoạt hằng ngày, nên việc loại bỏ đi cũng không ảnh hưởng nhiều đến hiệu suất thật tế.



Hình 1: Những folder con bên trong folder ‘an toàn’

Dataset criteria

Mỗi video trong dataset phải thoả mãn được những điều kiện sau:

- Video thu được phải được quay từ hướng chính diện

- Tuỳ thuộc vào góc quay của camera mà khoảng cách giữa người thực hiện động tác với camera dao động từ 30-50 cm
- Background trống, chỉ có một người trong mỗi khung hình
- Độ sáng khi quay ISO > 100
- Với những từ ngữ có nhiều động tác biểu thị khác nhau theo từng vùng miền, nhóm thống nhất chỉ chọn một vùng phổ biến nhất để bộ dataset được thống nhất

Hình 2: Động tác ‘an toàn’ được thực hiện qua các frame ảnh

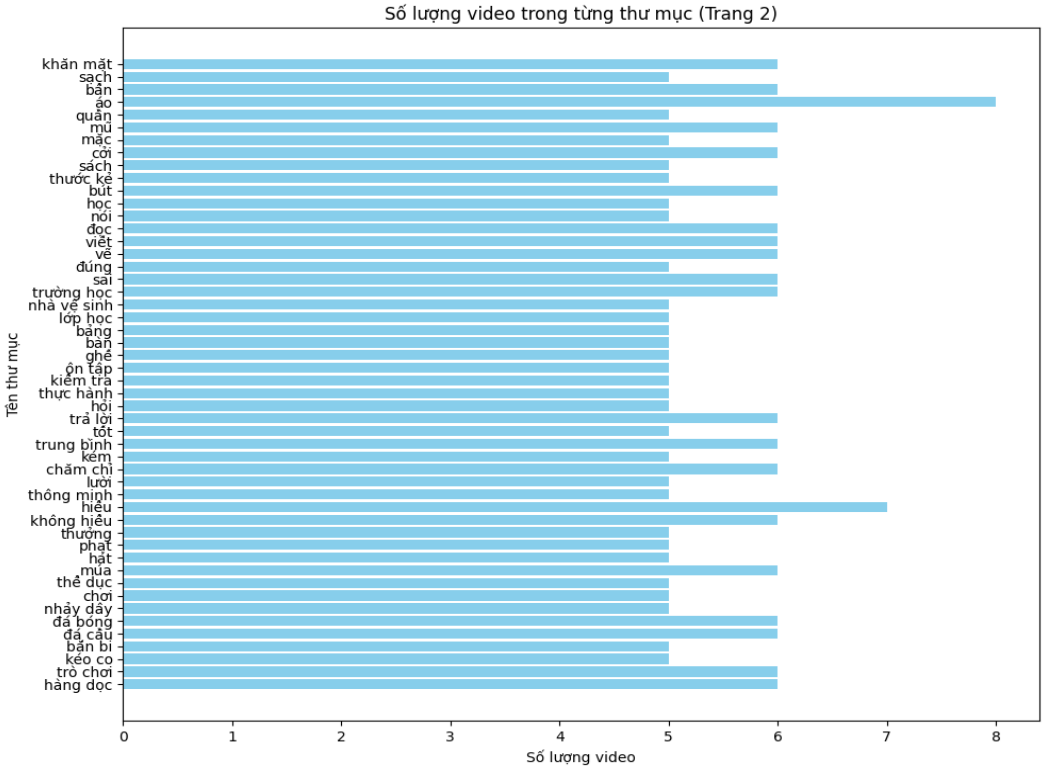
Dataset arrangement

Mỗi từ ngữ là một folder, mỗi folder sau khi được lọc có đến 5 hoặc hơn video, mỗi video đều thoả mãn những tiêu chí trên, thời lượng chưa đến 5 giây, đều thực hiện động tác như tên folder (vd: folder ‘hiệu trưởng’ có 4 video khác nhau từ 4 người khác nhau thực hiện động tác ‘hiệu trưởng’) => Tất cả các video đều được label thủ công trước khi bắt đầu thao tác. Mỗi folder

từ ngữ (ví dụ: folder “Hiệu trưởng”) sẽ được chia thành 3 folder con gồm ‘train’, ‘test’, ‘dev’, với tập test và dev đều chứa 1 video, phần còn lại chuyển vào tập train.

Dataset overview

Dataset	Gloves	Videos	Type	Language
Custom	500	3193	RGB	Vietnamese



Hình 3: Thống kê số lượng video trong 50 folder ngẫu nhiên

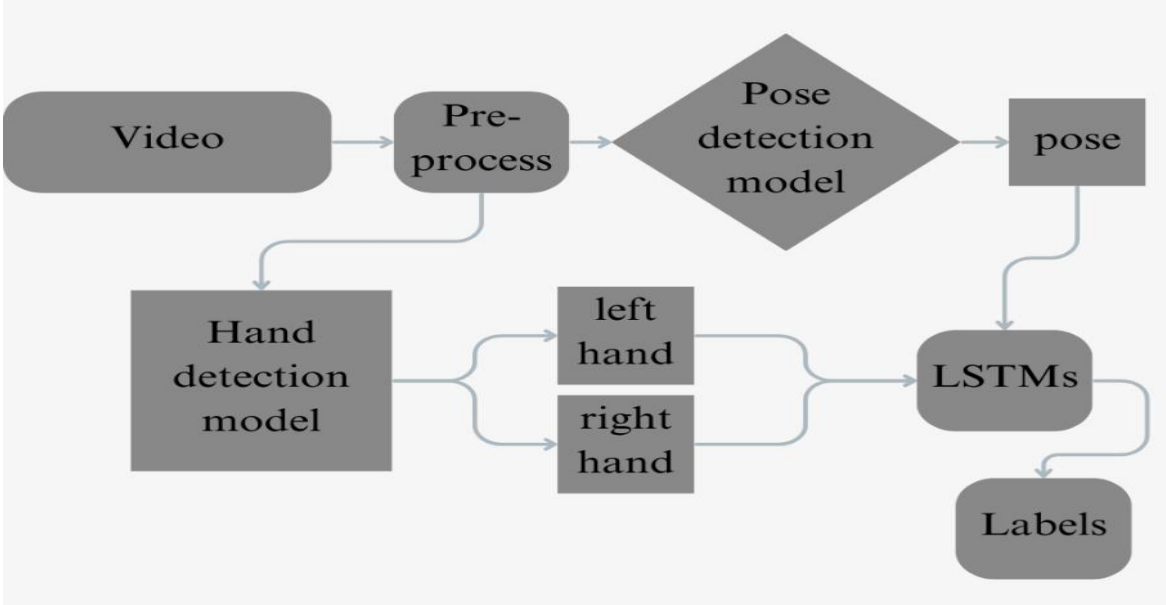
Tổng cộng có 3193 video, được chia thành 500 folder con tượng trưng cho 500 từ ngữ Việt Nam, các từ đa dạng về số lượng âm tiết, đa dạng về đề tài (từ trong cuộc sống hằng ngày đến trong công việc)

Các từ ngữ là những từ dùng thường xuyên trong sinh hoạt hằng ngày, cũng là những từ dùng thường xuyên trong các chủ đề mà chúng ta hay dùng để giao tiếp.

Model

Pipeline:

Từ ý tưởng và phương pháp trong (Dongxu Li, 2020), pipeline của model nhóm được miêu tả như sau: Dữ liệu thô đã được chuẩn bị từ trước -> Tiền xử lý trước khi đưa vào để trích xuất các keypoint (1) -> Đưa dữ liệu đã xử lý vào các model mediapipe để trích xuất các keypoint (pose, left hand, right hand) (2) -> flatten output của (2) trước khi đưa vào mạng LSTMs để training (3) -> dùng hàm softmax để đưa ra labels có tỉ lệ dự đoán cao nhất (4)



Hình 4: Pipeline của model

Đầu tiên, để dễ dàng xác định được kí hiệu, bọn em muốn video tập trung hoàn toàn vào người đang thực hiện động tác. Từ các video thô ban đầu, bọn em sử dụng YOLOv5 để xác định các bounding box của phần người, từ đó crop các frame theo các bounding box đó.

YOLOv5

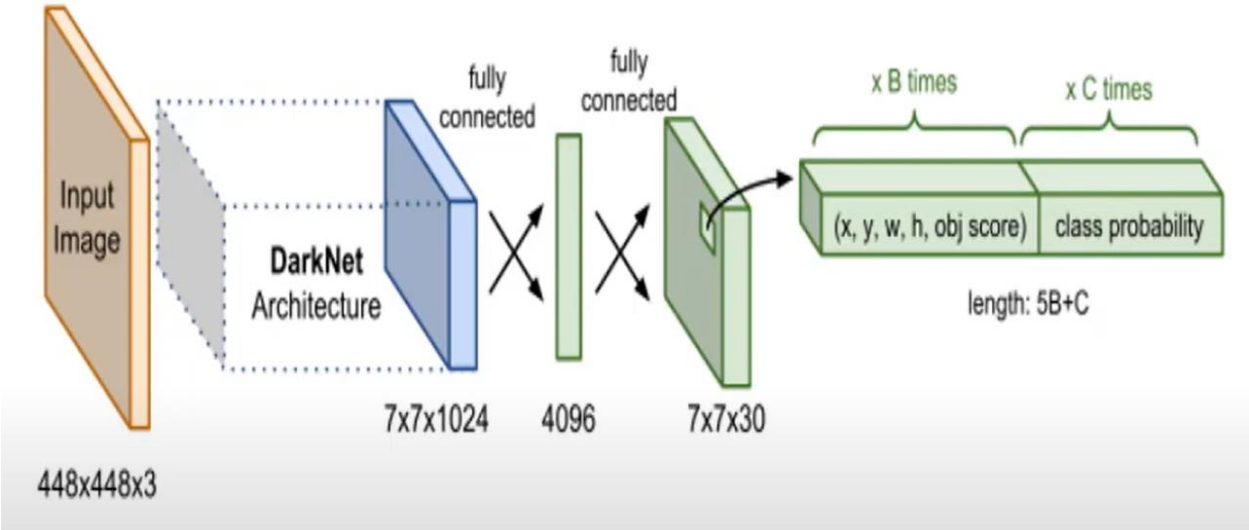
YOLO là thuật toán object detection được sử dụng rộng rãi nhất, được sử dụng chủ yếu để nhận diện vật thể trong ảnh và video. Hiện nay, YOLO vẫn là một trong những model để xây dựng state-of-the-art objects detector tốt nhất.

YOLOv5 là một mô hình được phát triển bởi nhóm Ultralytics và công bố dưới dạng mã nguồn mở trên GitHub, không thông qua bài báo khoa học đánh giá chính thức. Mô hình này được giới thiệu lần đầu vào năm 2020 và nhanh chóng thu hút sự chú ý lớn từ cộng đồng học máy và thị giác máy tính.

Những cải tiến của YOLOv5 so với YOLOv4:

- Cải thiện về cấu hình huấn luyện
- Cải thiện về tốc độ huấn luyện
- Tốc độ phát hiện nhanh
- YOLOv5 có nhiều phiên bản phù hợp với từng đối tượng sử dụng => Tính thực thi cao hơn. Tuy nhiên về độ chính xác thì v4 tốt hơn

Tổng quan về kiến trúc của YOLOv5:



Hình 5: Kiến trúc của YOLOv5

- Backbone: YOLOv5 sử dụng một mạng neural network chính, thường là một biến thể của CSPDarknet53 hoặc EfficientNet. Đây là phần dùng để trích xuất các đặc trưng của ảnh đầu vào
- Neck and Head: Đầu ra của backbone sẽ được đưa vào neck and head. Phần neck sẽ giúp khái quát hoá các object có quy mô và kích thước khác nhau. Từ đó phần head sẽ xác định các bounding box và xác suất tương ứng cho các lớp đối tượng. Thông thường, YOLOv5 sử dụng anchor boxes để dự đoán vị trí và kích thước của các đối tượng.

Sử dụng YOLOv5:

Do trong mỗi video đã chắc chắn chỉ có một người, output của YOLOv5 chỉ có một số thông tin đáng quan tâm như:

- (x_min; y_min): Toạ độ của góc dưới trái của vùng cần cắt
- (x_max; y_max): Toạ độ của góc trên phải của vùng cần cắt

Do có nhiều bounding box người được tìm thấy trong mỗi frame, nếu chọn bounding box có diện tích lớn nhất thì khi đó sẽ chắc chắn rằng không bỏ sót bất kì chi tiết người nào, từ đó dễ dàng xác định phần ảnh cần giữ lại khi crop.

Do bọn em sử dụng YOLOv5 chỉ để xác định phần người trong các frame ảnh, không cần phải huấn luyện lại YOLOv5 với dataset riêng của bọn em, nên bọn em dùng model YOLOv5 đã được huấn luyện sẵn với mục tiêu là human detection, giúp tiết kiệm thời gian và tài nguyên.

Đánh giá YOLOv5 trong điều kiện bài toán:

Ưu điểm:

- Tốc độ xử lý rất nhanh
- Đang được sử dụng rộng rãi trong nhiều lĩnh vực hiện nay
- Có nhiều phiên bản custom riêng, phù hợp với nhu cầu của các bài toán khác nhau
- Có nhiều model được train trước với dataset khổng lồ, tin cậy hơn, không cần phải train lại

Nhược điểm:

- Kích thước một model lớn cũng như đòi hỏi tài nguyên tính toán cao để triển khai, nhất là khi máy bọn em có giới hạn về tài nguyên

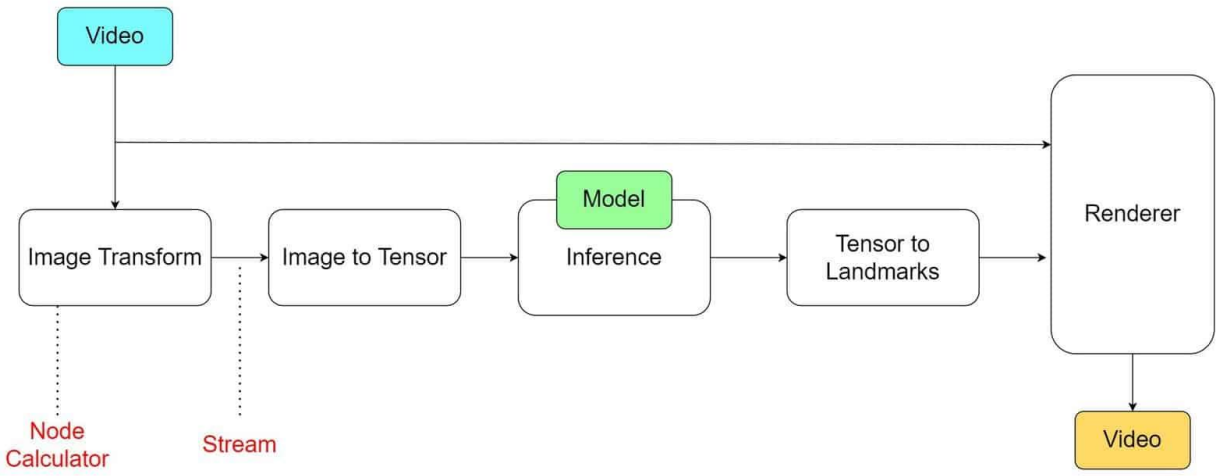
MediaPipe

Do các hành động của con người khá khó để xác định trong duy nhất một frame ảnh, việc sử dụng một model CNN thông thường để xác định hành động của con người (ví dụ: frame một người đang chạy bộ (jogging) và một người đang chạy (running) gần như là giống hoàn toàn), vì thế không thể dùng CNN classification được.

Giải pháp: thay vì trích xuất các đặc trưng từ frame ảnh, chúng ta tập trung vào các điểm chính, các khớp của con người (gọi tắt là keypoint). Từ các keypoint đó, chúng ta train để máy hiểu được ý nghĩa của kí hiệu đó thông qua vị trí của các keypoint, sự dịch chuyển của các keypoint,... (Dongxu Li, 2020)

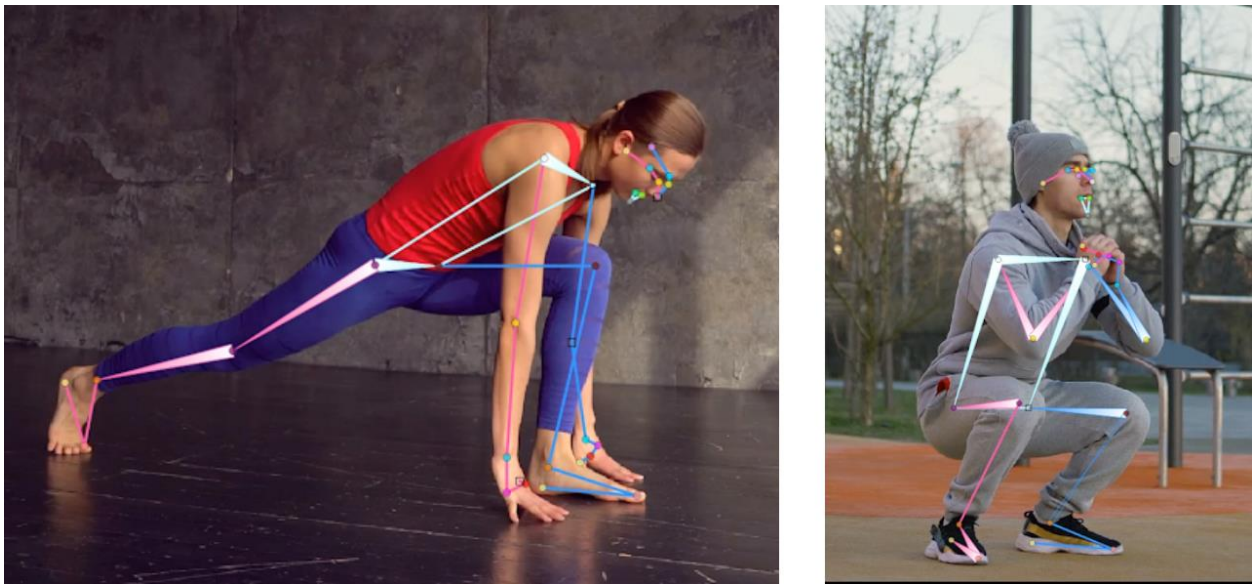
Có 3 phần chính của con người mà chúng ta cần quan tâm để xác định ý nghĩa kí hiệu đó: Tư thế (pose), bàn tay trái (left hand) và bàn tay phải (right hand). Phần khuôn mặt (dùng Face Mesh) đã được cân nhắc và bỏ qua do từ những data gốc, bọn em nhận thấy khuôn mặt không đóng vai trò quan trọng trong việc nhận diện ngôn ngữ kí hiệu.

MediaPipe là một nền tảng phần mềm của Google được công bố trong bài báo " (Camillo Lugaresi, 2019)", được sử dụng để xử lý dữ liệu thị giác máy tính và âm thanh trong thời gian thực. Nó cung cấp một bộ công cụ và thư viện để phát triển các ứng dụng liên quan đến nhận diện hình ảnh, phát hiện đối tượng, theo dõi điểm keypoint, và xử lý âm thanh. MediaPipe được thiết kế để hỗ trợ các ứng dụng trên nhiều nền tảng từ máy tính đến thiết bị di động.



Hình 6: Quy trình xử lý của MediaPipe

Bọn em sử dụng BlazePose GHUM 3D (MediaPipe BlazePose GHUM 3D, n.d.) để trích xuất các keypoint của tư thế người trong frame ảnh, và MediaPipe Hand Landmarker (Model Card Hand Tracking (Lite/Full) with Fairness Oct 2021, n.d.) để trích xuất các keypoint của bàn tay (cả trái và phải) trong frame ảnh.



Hình 7: Minh hoạ BlazePose GHUM 3D nhận diện keypoint

Model pose extraction sẽ trả về một array có kích thước 33x5 với 33 điểm keypoint, mỗi điểm gồm x, y, z, visibility, presence (MediaPipe BlazePose GHUM 3D, n.d.):

- x, y: Toạ độ của điểm chính, giá trị nằm trong khoảng [0,0; 255.0] và là vị trí của điểm đó trên mặt phẳng toạ độ 2D của hình ảnh
- z: Biểu thị cho độ sâu của điểm chính hoặc là khoảng cách của điểm chính so với mặt phẳng của hông người, có thể xem như hông người chính là gốc của trục z. Giá trị âm biểu thị rằng điểm đó đang nằm giữa hông người và camera, ngược lại giá trị dương biểu thị điểm đó đang nằm sau hông người.

- visibility: Cung cấp thông tin về khả năng nhìn thấy hoặc cụ thể hơn là xác suất mà điểm chính đang nằm trong khung hình và không bị che khuất bởi các bộ phận của cơ thể hoặc các đối tượng khác. Giá trị nằm trong khoảng [min_float; max_float] và khi được đưa qua hàm sigmoid thì sẽ thể hiện tỉ lệ nhìn thấy được điểm đó
- presence: tương tự như visibility, presence cũng có giá trị nằm trong khoảng [min_float; max_float] và khi được đưa qua hàm sigmoid thì sẽ thể hiện tỉ lệ điểm đó có nằm trong khung hình hay không

Model hand detection trả về một array kích thước 21x3 tượng trưng cho x, y, z (Model Card Hand Tracking (Lite/Full) with Fairness Oct 2021, n.d.):

- x, y: Toạ độ của điểm chính, giá trị nằm trong khoảng [0,0; 255.0] và là vị trí của điểm đó trên mặt phẳng toạ độ 2D của hình ảnh
- z: Biểu thị cho độ sâu của điểm chính hoặc là khoảng cách của điểm chính so với mặt phẳng của hông người, có thể xem như hông người chính là gốc của trục z. Giá trị âm biểu thị rằng điểm đó đang nằm giữa hông người và camera, ngược lại giá trị dương biểu thị điểm đó đang nằm sau hông người.

Một frame ảnh chỉ nhận diện 3 thứ: tư thế, tay trái, tay phải. Với mỗi điểm dự đoán được, chỉ nhận những điểm có độ tin cậy >= 0.5 (min_detection_confidence=0.5) và có độ tin cậy khi tracking >=0.5 (min_tracking_confidence=0.5). Những điểm không có trong frame ảnh sẽ được thay bằng vector 0, chứ không được bỏ trống (nghĩa là không xác định được những điểm đó chứ không phải không tồn tại). Nếu model không nhận diện được thì thay bằng vector 0 (ví dụ: Có một số từ ngữ chỉ thực hiện bằng một tay).

Sau khi đã có hết các dữ liệu, ta flatten ra thành một array chung để có thể thao tác tiếp. Như vậy, từ một video có x frame, chúng ta thu được một object có kích thước (x; 291) với mỗi frame có 291 parameters (33*5 + 21*3*2 = 291)

Đánh giá MediaPipe trong điều kiện bài toán:

Ưu điểm:

- Có thể tính toán thời gian thực
- Tuy không hoàn hảo nhưng độ chính xác khi nhận diện các điểm khung cao
- Dễ sử dụng, có thể sử dụng một model đã được train sẵn

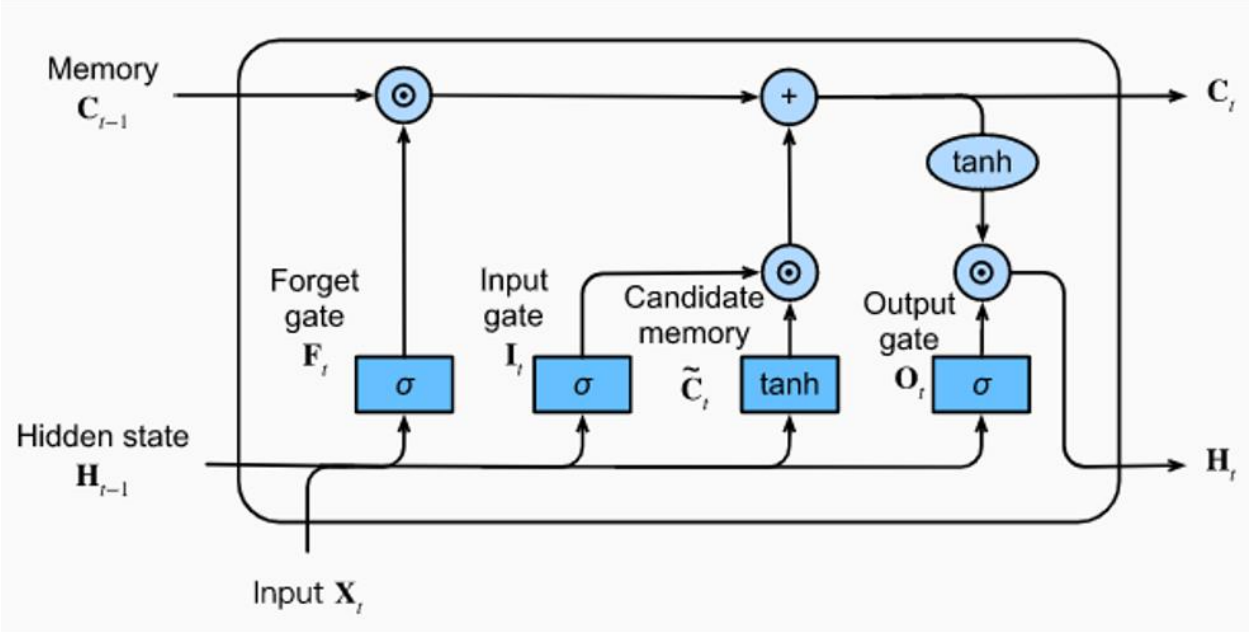
Nhược điểm:

- Yêu cầu tài nguyên khá cao
- Việc đánh giá model của MediaPipe là rất khó do việc label thủ công từng keypoint rất tốn thời gian
- Tuy dễ sử dụng nhưng việc tuỳ chỉnh hay cải tiến một model MediaPipe cho phù hợp với bài toán cũng gặp nhiều khó khăn

Long short-term memory (LSTM)

LSTM (Long Short-Term Memory) là một loại kiến trúc mạng nơ-ron thuộc dạng mạng nơ-ron hồi quy (RNN), được thiết kế để xử lý dữ liệu tuần tự có tính chất dài hạn. LSTM được sử dụng để mô hình hóa và dự đoán chuỗi dữ liệu, như dự đoán chuỗi thời gian, xử lý ngôn ngữ tự nhiên và nhiều ứng dụng khác đòi hỏi khả năng nhớ và hiểu cấu trúc dữ liệu tuần tự. LSTM được giới thiệu lần đầu vào năm 1997 bởi Sepp Hochreiter và Jürgen Schmidhuber trong một bài báo mang tên "Long Short-Term Memory" được công bố tại NIPS (Neural Information Processing

Systems). LSTM có khả năng học và lưu trữ thông tin trong một khoảng thời gian dài mà không bị ảnh hưởng bởi vấn đề biến mất hoặc triệt tiêu gradient như RNN thông thường. Điều này cho phép LSTM duy trì thông tin quan trọng và quyết định để đưa ra dự đoán hoặc xử lý các chuỗi dữ liệu tuần tự một cách hiệu quả



Hình 8: Cấu trúc của một LSTM

Cấu trúc của một LSTM:

1. Forget Gate:

Forget gate quyết định thông tin nào sẽ được quên hoàn toàn hoặc giữ lại từ cell trước. Nó sử dụng hàm sigmoid để đưa ra quyết định về việc loại bỏ thông tin nào là không cần thiết dựa trên input hiện tại và hidden state trước đó.

2. Input Gate:

Input gate quyết định thông tin mới nào sẽ được thêm vào cell. Nó sử dụng hàm sigmoid để quyết định thông tin nào sẽ được cập nhật. Cũng sử dụng hàm tanh để tạo ra vector các giá trị ứng cử (candidate values) có thể được thêm vào cell.

3. Cell State

Là bộ nhớ dài hạn của LSTM, nơi chứa thông tin từ các cell trước và thông tin được cập nhật qua các cổng. Cell state chịu ảnh hưởng của forget gate để loại bỏ hoặc giữ lại thông tin cũ, và nhận thông tin mới thông qua input gate.

4. Output Gate:

Output gate quyết định thông tin nào từ cell state sẽ được đưa ra làm output. Nó sử dụng hàm sigmoid để xác định phần nào của cell state sẽ được đưa ra, sau khi qua hàm tanh để chuẩn hóa giá trị.

5. Hidden State:

Là phiên bản được xử lý của cell state, nó chứa thông tin mà LSTM sử dụng để đưa ra output và quyết định cho các cell sau đó.

Đánh giá về LSTM trong điều kiện bài toán:

Ưu điểm:

- LSTM rất phù hợp để thao tác với dữ liệu chuỗi, thứ tự trước sau
- Giảm được số lượng parameters đáng kể so với một layer fully connected thông thường

Nhược điểm:

- Đòi hỏi bộ dữ liệu huấn luyện lớn

- Yêu cầu tài nguyên tính toán khá cao, nhất là khi thiết bị của bọn em có giới hạn
- Độ chính xác của LSTM phụ thuộc nhiều vào độ đa dạng của dataset

Implimentation details

Ngoại trừ YOLOv5 dùng Torch, các model còn lại như MediaPipe, LSTM đều sử dụng Tensorflow để thực hiện. Kết hợp YOLOv5 pretrained, MediaPipe pretrained, và một mạng LSTM xếp chồng 2 lớp thành một model hoàn chỉnh, nhóm dùng thuật toán optimizer Adam. Model được train với batch size là 256, train đến tối đa 1000 epochs. Việc training sẽ tự động dừng lại khi validation accuracy dừng gia tăng và model được lưu lại. Để tăng cường tối đa cho việc training nên mỗi folder từ ngữ chỉ có 1 video cho tập 'test', 1 video cho tập 'dev', phần còn lại được đưa vào tập 'train'. Như thế sẽ đảm bảo mỗi từ ngữ đều có 1 video cho tập 'dev' và 'test', vừa tăng tối đa hiệu suất training.

Sau khi đã trích xuất 33 keypoint cơ thể người bằng model BlazePose GHUM 3D (MediaPipe BlazePose GHUM 3D, n.d.) và 21 keypoint ở mỗi bàn tay trái và phải bằng model MediaPipe Hands (Model Card Hand Tracking (Lite/Full) with Fairness Oct 2021, n.d.), ta concatenate các thông tin đã được trích xuất theo mỗi keypoint từ 75 keypoint và dùng nó làm feature đầu vào để train model LSTM xếp chồng 2 lớp, mỗi lớp có 128 đơn vị, tỉ lệ dropout là 0.5. Cross-entropy được sử dụng để làm loss function cho model này.

$$-\sum_{i=1}^{n} y_i \log \bar{y}_i$$

Trong đó, n là số lượng các label, log là hàm logarit tự nhiên, y là số nhị phân (0 hoặc 1) thể hiện phân phối xác suất thực tế của các lớp và thường được biểu diễn dưới dạng one-hot decoding, $\bar{y}$ thể hiện xác suất được dự đoán bởi model của các label.

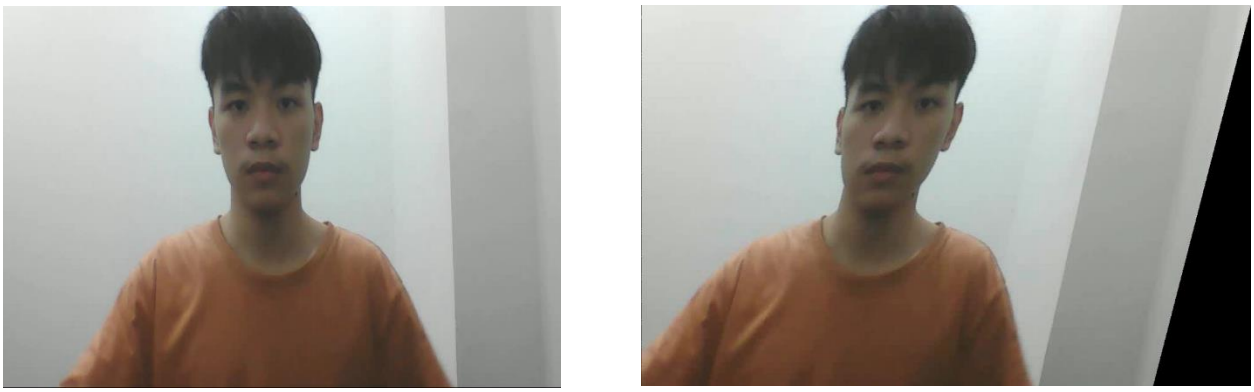
Data augmentation

Do mỗi từ ngữ chỉ có khoảng hơn 3 video để train, model khi đó dễ bị overfit và có độ chính xác thấp. Nhóm bọn em cũng không có đủ thời gian và tài nguyên để cải thiện bộ dữ liệu của mình. Vì thế bọn em quyết định sử dụng các thuật toán để tăng cường bộ dữ liệu (video augmentation). Tăng cường dữ liệu là một kỹ thuật được sử dụng để mở rộng quy mô và tính đa dạng của tập dữ liệu một cách giả tạo bằng cách tạo các phiên bản mới thông qua các phép biến đổi khác nhau.

Để cải thiện cũng như tăng độ đa dạng của dataset, nhóm đã quyết định chọn những những video sau khi tăng cường nên có sự dịch chuyển các keypoint khác so với video gốc, nên nhóm lựa chọn các phương pháp geometry based như shear, translate, rotate (Tung, 2022):

Shear:

Shear đùng để shear hình ảnh hay video tuỳ theo trục mà ta chọn. Những kiểu shear thông dụng nhất là shear theo trục x (shear_x) và shear theo trục y (shear_y)



Hình 9: Frame ảnh đầu của video trước và sau khi áp dụng shear x

Hình 10: Frame ảnh đầu của video trước và sau khi áp dụng shear y

Translate

Dịch chuyển ảnh hay video qua một khoảng ngẫu nhiên hoặc tự xác định trước. Có thể dịch chuyển theo trục x (translate_x) hoặc theo trục y (translate_y)

Hình 11: Frame ảnh đầu của video trước và sau khi áp dụng translate_x


Hình 12: Frame ảnh đầu của video trước và sau khi áp dụng translate_y

Rotate

Quay một frame hoặc video theo một hướng ngẫu nhiên hoặc tự xác định trước.


Hình 13: Frame ảnh đầu của video trước và sau khi áp dụng rotate

Lưu ý: Tất cả các frame trong video đều được thực hiện augment và khi augment thì các frame có liên kết với nhau chứ không augment cùng một tham số. Để dễ hình dung, khi áp dụng rotate vào video thì video sẽ trông như quay tròn một tấm ảnh cố định, chứ không phải chỉ nghiên cố định tấm ảnh.

Evaluation

Bọn em đánh giá model bằng phương pháp đánh giá độ chính xác phân loại top-K với K={1, 5, 10} với mọi label có trong dataset (Dongxu Li, 2020). Với việc có nhiều từ ngữ có các thao tác gần giống nhau, vì thế model sẽ có nhiều lỗi trong lúc đánh giá kết quả. Việc chỉ chấp nhận kết quả đúng có thể sẽ không đánh giá được model một cách tổng quan nhất. top-K sẽ chọn k label có tỉ lệ dự đoán cao nhất. Vì thế, việc sử dụng top-K sẽ giúp ta đánh giá model trực quan hơn và linh hoạt hơn. Áp dụng dự đoán top-K cho mọi label, sau đó tính trung bình số label được top-K dự đoán chính xác trong tổng số các label. Cụ thể hơn, top-1 dự đoán khả năng từ ngữ đó được model dự đoán chính xác, top-5 dự đoán khả năng từ ngữ đó nằm trong top 5 lựa chọn có xác suất cao nhất và top-10 dự đoán khả năng từ ngữ đó nằm trong top 10 lựa chọn có xác suất cao nhất.

Sau khi đã đánh giá model, nhóm thu được kết quả sau (đã được làm tròn đến chỉ số thập phân thứ 3):

Method	Top-1	Top-5	Top-10
Custom model	0.082	0.196	0.248

Discussion

Difficulty

Do thao tác qua nhiều model khác nhau nên thời gian thực thi cao. Hầu hết mỗi kí hiệu chỉ có 2-3 video được đưa vào để train nên khó xác định được độ chính xác thấp là do thiếu dữ liệu hay

sai sót trong khâu xử lý của model. Với số lượng video hơn 3000, việc label 75 keypoint thủ công cho mỗi video gần như là một điều bất khả thi trong điều kiện thời gian và chi phí thiếu sót. Với những từ chỉ có một động tác thì dễ để dự đoán, tuy nhiên có nhiều từ phải thực hiện nhiều động tác cùng lúc, khó để máy có thể dự đoán. Do không đủ bộ nhớ nên tiền xử lý phải chia thành nhiều khâu nhỏ khác nhau. Do nhóm chỉ có 2 thành viên nên số lượng công việc khổng lồ này đòi hỏi nhiều thời gian thực hiện. Việc thao tác với lượng dữ liệu lớn đòi hỏi các thành viên phải thường xuyên dùng Google Drive, nhưng việc tải dữ liệu lên thường xuyên bị lỗi up thiếu dữ liệu, nhưng khi tải dữ liệu về thì ít khi bị thiếu. Những kiến thức áp dụng trong đề tài này không được giảng dạy nhiều trên lớp, hầu hết là biết trước hoặc tự tìm hiểu nên có thể có sai sót.

Improvement

Data:

Nên bổ sung thêm số lượng video trong mỗi từ vựng thay vì phụ thuộc nhiều vào tăng cường dữ liệu. Nếu có thể thì nên thêm số lượng từ vựng để bao hàm hơn về đề tài này. Ngoài ra, khi quay các động tác thực hiện, người thực hiện nên làm chậm rãi hơn, chú ý hơn về từng chi tiết trong mỗi động tác. Các động tác thực hiện nên rõ ràng hơn. Nên thêm số lượng người thực hiện các động tác để tránh overfit. Có thể đầu tư các thiết bị vật lý tốt hơn để chất lượng dataset được nâng cao. Nếu có thể thì nên lọc các từ có động tác khó nhằn, các từ nhiều động tác kết hợp ra khỏi dataset vì máy khó học.

Model:

Do độ chính xác không cao nên cân nhắc việc sử dụng pretrained model để nghiên cứu. Cải thiện dataset để giúp model hoạt động hiệu quả hơn. Ngoài ra, cũng còn một số biện pháp cải thiện model như điều chỉnh độ phức tạp lại của networks, thay đổi các tham số, ...

Application

Giống như chức năng tìm kiếm bằng giọng nói của Google, có thể thêm chức năng tìm kiếm bằng ngôn ngữ kí hiệu cho người khiếm thính

Có thể đảo ngược bài toán: Thay vì dự đoán ý nghĩa của ngôn ngữ kí hiệu, nếu phát triển dataset nhiều hơn về số lượng từ ngữ, kết hợp với các keypoint đã trích xuất và train, có thể làm một chương trình cho phép nhập văn bản cần thể hiện cho người khiếm thính, từ đó xuất ra hình ảnh 3D người đang thực hiện các động tác ngôn ngữ kí hiệu đó, khi đó có thể giải quyết giao tiếp cho cả 2 bên

References

Camillo Lugaresi, J. T. (2019). *MediaPipe: A Framework for Building Perception Pipelines.* Google Research.

Dongxu Li, C. R. (2020). *Word-level Deep Sign Language Recognition from Video: A New Large-scale Dataset and Methods Comparison.*

MediaPipe BlazePose GHUM 3D. (n.d.). Retrieved from https://drive.google.com/file/d/10WlcTvrQnR_R2TdTmKw0nkyRLqrwNkWU/preview?pli=1

Model Card Hand Tracking (Lite/Full) with Fairness Oct 2021. (n.d.). Retrieved from https://storage.googleapis.com/mediapipe-assets/Model%20Card%20Hand%20Tracking%20(Lite_Full)%20with%20Fairness%20Oct%202021.pdf#page=7&zoom=100,61,737

Renotte, N. (2021, June 19). *Sign Language Detection using ACTION RECOGNITION with Python | LSTM Deep Learning Model*. Retrieved from Youtube: https://www.youtube.com/watch?v=doDUihpj6ro&t=6827s&ab_channel=NicholasRenotte

Saiful, M. &. (2022). *Real-Time Sign Language Detection Using CNN.*

Starmer, S. w. (2022, November 7). *Long Short-Term Memory (LSTM), Clearly Explained*. Retrieved from Youtube: https://www.youtube.com/watch?v=YCzL96nL7j0&t=811s&ab_channel=StatQuestwithJoshStarmer

Starmer, S. w. (2022, July 11). *Recurrent Neural Networks (RNNs), Clearly Explained!!!* Retrieved from Youtube: https://www.youtube.com/watch?v=AsNTP8Kwu80&ab_channel=StatQuestwithJoshStarmer

Thắng, N. C. (2020, July 4). *Cơ bản về Object Detection với R-CNN, Fast R-CNN, Faster R-CNN và YOLO*. Retrieved from Mì AI: https://www.miai.vn/2020/07/04/co-ban-ve-object-detection-voi-r-cnn-fast-r-cnn-faster-r-cnn-va-yolo/

Thắng, N. C. (2020, July 4). *Cơ bản về Object Detection với R-CNN, Fast R-CNN, Faster R-CNN và YOLO. Retrieved from*. Retrieved from Mi AI web site: https://www.miai.vn/2020/07/04/co-ban-ve-object-detection-voi-r-cnn-fast-r-cnn-faster-r-cnn-va-yolo/

Tung, B. T. (2022, May 31). *[Paper Explain] Learning Temporally Invariant and Localizable Features via Data Augmentation for Video Recognition - Bàn luận 1 chút về video augmentation.* Retrieved from Viblo web site: https://viblo.asia/p/paper-explain-

learning-temporally-invariant-and-localizable-features-via-data-augmentation-for-video-recognition-ban-luan-1-chut-ve-video-augmentation-RnB5pAL6KPG