TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN

KHOA TOÁN – CƠ - TIN HỌC

BÁO CÁO CUỐI KỲ

ĐỀ TÀI:
ASL SIGN LANGUAGE RECOGNITION

Học phần: Nhập môn trí tuệ nhân tạo

Học kỳ: 1 - 2025-2026

Mã học phần: MAT3508

Giảng viên: Hoàng Anh Đức

Họ và tên	Github	MSV
Đào Ngọc Bảo	DNB-Bao1405	23001500
Mai Thế Cường	Matec76	23001504
Bùi Thanh Lâm	23001532-cloud	23001532
Tống Minh Phong	Mphong2005	23001545
Đỗ Thị Như Quỳnh	quynh2196	23001556

Hà Nội, tháng 11 năm 2025

Lời nói đầu

Nhận dạng ngôn ngữ ký hiệu (Sign Language Recognition - SLR) là một bài toán thử thách trong lĩnh vực xử lý video và thị giác máy tính, với tác động trực tiếp đến chất lượng cuộc sống của cộng đồng người khiếm thính. Theo thống kê của WHO, có khoảng 466 triệu người bị khiếm thính trên thế giới, trong đó hàng triệu người sử dụng ngôn ngữ ký hiệu như phương thức giao tiếp chính. Tại Bắc Mỹ, Ngôn ngữ Ký hiệu Mỹ (American Sign Language - ASL) được sử dụng bởi khoảng 250.000 – 500.000 người, trong đó hàng triệu người khác là gia đình hoặc bạn bè của họ.

Mặc dù ngôn ngữ ký hiệu là một hệ thống ngôn ngữ hoàn chỉnh với ngữ pháp và từ vựng phong phú, việc tự động hóa nhận dạng và phiên dịch vẫn là thách thức lớn. Các phương pháp truyền thống dựa trên LSTM hoặc GRU có những hạn chế đáng kể:

1. xử lý tuần tự không tận dụng được khả năng song song của GPU,
2. gradient vanishing làm khó việc học các phụ thuộc lâu dài trên chuỗi dài,
3. tốc độ inference chậm, không đáp ứng yêu cầu real-time.

Các phương pháp dựa trên CNN xử lý pixel trực tiếp thường bị ảnh hưởng mạnh bởi sự thay đổi ánh sáng, góc quay, và nền tảng.

Dự án này ứng dụng kiến trúc **Transformer** – một phát triển quan trọng trong lĩnh vực học sâu – kết hợp với **MediaPipe Holistic** để xây dựng hệ thống nhận dạng ngôn ngữ ký hiệu ASL. Cách tiếp cận chính là: thay vì xử lý trực tiếp trên ảnh pixel ($1280 \times 720 \times 3 = 2.76$M dimensions), chúng tôi trích xuất tọa độ 3D của các điểm mốc cơ thể (body landmarks) từ MediaPipe, sau đó áp dụng mô hình Transformer để phân tích chuỗi temporal này. Phương pháp này giảm đáng kể độ phức tạp tính toán (từ 2.76M xuống 346 dimensions) và tăng tính bền vững trước các biến thể môi trường.

Mục lục

Chương 1

Giới thiệu

1.1 Tóm tắt

Báo cáo trình bày việc ứng dụng kiến trúc Transformer cho bài toán nhận dạng ngôn ngữ ký hiệu ASL (American Sign Language). Hệ thống kết hợp MediaPipe Holistic để trích xuất 173 điểm mốc quan trọng từ 543 điểm mốc gốc, sau đó sử dụng mô hình Transformer gồm 4 encoder layers để phân loại 10 ký hiệu phổ biến.

Kết quả thực nghiệm cho thấy mô hình đạt độ chính xác 87.39% trên tập validation, vượt mục tiêu ban đầu (85%) và cải thiện 7.9% so với baseline CNN-LSTM (79.5%). Tốc độ xử lý đạt 20-25 FPS, đáp ứng yêu cầu thời gian thực. Mô hình compact với 1.44 triệu tham số (∼5.5 MB) có thể triển khai trên phần cứng phổ thông.

Phân tích chi tiết cho thấy cặp ký hiệu "hear" và "listen" có tỷ lệ nhầm lẫn cao nhất (31% và 11%) do sự tương đồng về đặc trưng không gian và hạn chế trong việc khai thác các đặc trưng khuôn mặt. Ablation study xác nhận vai trò quan trọng của positional encoding (+5.3% accuracy) và label smoothing (+1.3% accuracy).

Từ khóa: Nhận dạng ngôn ngữ ký hiệu, Transformer, MediaPipe, Self-Attention, Học sâu

1.2 Bối cảnh và Động lực

Theo Tổ chức Y tế Thế giới (WHO), có khoảng 466 triệu người trên thế giới bị khiếm thính, chiếm khoảng 5% dân số toàn cầu. Trong số đó, hàng triệu người sử dụng ngôn ngữ ký hiệu làm phương thức giao tiếp chính. Tại Bắc Mỹ, Ngôn ngữ Ký hiệu Mỹ (American Sign Language - ASL) được sử dụng bởi khoảng 250,000-500,000 người, trong đó có cả những người điếc bẩm sinh và những người khiếm thính sau này.

Mặc dù ngôn ngữ ký hiệu là một hệ thống ngôn ngữ tự nhiên hoàn chỉnh với ngữ pháp và từ vựng phong phú, việc tự động hóa nhận dạng vẫn là một thách thức lớn. Các rào cản chính bao gồm: (1) sự phức tạp của chuyển động không gian-thời gian, (2) biến thể lớn giữa các cá nhân, (3) sự phụ thuộc vào nhiều yếu tố môi trường.

Việc phát triển hệ thống nhận dạng ngôn ngữ ký hiệu tự động có nhiều ứng dụng thực tế:

- **Hệ thống dịch thuật tự động**: Chuyển đổi ký hiệu thành văn bản hoặc giọng nói
- **Giao diện người dùng thân thiện**: Cho phép người khiếm thính tương tác với thiết bị mà không cần người phiên dịch

- **Công nghệ hỗ trợ giao tiếp**: Cải thiện chất lượng giao tiếp trong các cơ quan công cộng
- **Giáo dục**: Hỗ trợ học tập ngôn ngữ ký hiệu

1.3 Đặc điểm của Ngôn ngữ Ký hiệu ASL

ASL là một hệ thống ngôn ngữ tự nhiên hoàn chỉnh, độc lập với tiếng Anh nói. Nó được cấu thành từ năm yếu tố cơ bản (parameters):

1. **Hình thái bàn tay (Handshape)**: Cách các ngón tay được sắp xếp và uốn cong. Có hơn 100 hình thái khác nhau trong ASL, tương tự như các phụ âm trong tiếng nói.
2. **Chuyển động (Movement)**: Quỹ đạo và kiểu di chuyển của tay trong không gian. Ví dụ: chuyển động lên xuống, xoay, co dãn, rung rinh. Chuyển động cũng có thể là mạnh hoặc nhẹ.
3. **Vị trí (Location)**: Nơi trên cơ thể hoặc trong không gian mà ký hiệu được thực hiện. Ví dụ: gần tai (cho "hear", "listen"), gần mặt (cho "face", "see"), trước cơ thể.
4. **Định hướng (Orientation)**: Hướng của lòng bàn tay và hướng của ngón tay. Một ký hiệu với hình thái và vị trí giống nhau nhưng định hướng khác sẽ có ý nghĩa khác.
5. **Yếu tố phi thủ công (Non-manual markers)**: Bao gồm biểu cảm khuôn mặt, chuyển động đầu, tư thế cơ thể, chuyển động của lông mày. Các yếu tố này không chỉ biểu đạt cảm xúc mà còn chứa **thông tin ngữ pháp** — ví dụ, câu hỏi trong ASL thường được đánh dấu bằng chuyển động đầu qua trái-phải và lông mày nhón.

Sự phức tạp của ASL nằm ở chỗ tất cả năm yếu tố này phải kết hợp một cách chính xác để tạo nên ý nghĩa. Một sự thay đổi nhỏ trong bất kỳ yếu tố nào cũng có thể thay đổi hoàn toàn ý nghĩa của ký hiệu.

1.4 Những Thách thức Kỹ thuật

Nhận dạng tự động ngôn ngữ ký hiệu đặt ra những thách thức kỹ thuật độc đáo:

1.4.1 Bản chất Chuỗi Thời gian

Không giống như nhận dạng chữ viết hoặc đối tượng trong ảnh tĩnh, mỗi ký hiệu ASL là một trình tự chuyển động kéo dài 15-100 khung hình ở 30 FPS. Một khung hình đơn lẻ không cung cấp đủ thông tin để xác định ký hiệu — phải xem toàn bộ quá trình

thực hiện. Điều này làm cho bài toán trở thành **phân loại chuỗi thời gian (temporal sequence classification)** thay vì phân loại không gian (spatial classification).

1.4.2 Biến thể Cá nhân Lớn

Mỗi người thực hiện ký hiệu theo cách riêng của họ:

- **Tốc độ**: Từ 2-10 khung hình cho một ký hiệu (biến thể 5 lần)
- **Biên độ**: Khoảng cách di chuyển khác nhau
- **Phong cách**: Có người mượt mà, có người rõ ràng, có người chặt chẽ

Mô hình cần tổng quát hóa (generalize) được trên các biến thể này mà không cần phải huấn luyện lại (retrain) cho từng người dùng mới.

1.4.3 Phụ thuộc vào Biến thể Môi trường

Các yếu tố môi trường ảnh hưởng đáng kể:

- **Ánh sáng**: Sáng mạnh, tối, ánh sáng nghiêng
- **Góc camera**: 0, 45, 90, các góc cực đoan
- **Khoảng cách**: Quá gần, quá xa
- **Trang phục**: Đặc biệt màu tay và nền
- **Nền tảng (Background)**: Nền phức tạp, chuyển động

Nếu mô hình xử lý trực tiếp trên ảnh điểm ảnh (pixel), nó sẽ bị ảnh hưởng mạnh bởi những yếu tố này. Một mô hình bền vững (robust) cần tách biệt được "ý nghĩa của ký hiệu" (semantic meaning) từ các yếu tố không liên quan.

1.4.4 Yêu cầu Xử lý Thời gian Thực

Để thực sự hữu ích trong giao tiếp, hệ thống cần phản ứng đủ nhanh:

- $<$ **3-4 FPS**: Gây cảm giác không tự nhiên, hỏng giao tiếp
- **10-15 FPS**: Chấp nhận được nhưng có độ trễ
- **20-30 FPS**: Tự nhiên, gần thời gian thực (real-time)

1.4.5 Chuỗi Dài và Phụ thuộc Dài hạn

Một ký hiệu có thể kéo dài 30-50 khung hình hoặc hơn ở 30 FPS. Mô hình cần nắm bắt được mối quan hệ giữa các dấu thời gian (timestamps) cách xa nhau — ví dụ, hình thái bàn tay ở khung hình đầu ảnh hưởng đến ý nghĩa của cả ký hiệu.

1.5 Các Phương pháp Hiện tại

1.5.1 Phương pháp dựa trên CNN

Sử dụng mạng nơ-ron tích chập (CNN) như VGG, ResNet, Inception để trích xuất đặc trưng từ từng khung hình, sau đó kết hợp với mạng nơ-ron hồi quy (RNN) để xử lý chuỗi.

Ưu điểm: CNN rất hiệu quả trong trích xuất đặc trưng không gian (spatial features).

Nhược điểm:

- Vẫn phụ thuộc vào đặc trưng cấp điểm ảnh, bị ảnh hưởng bởi ánh sáng và nền
- Mạng CNN chính (backbone) lớn (VGG: 138M tham số, ResNet: 25M+), chậm khi suy luận (inference)
- Kết hợp CNN-RNN gặp vấn đề của cả hai kiến trúc

1.5.2 Phương pháp dựa trên RNN/LSTM

Các kiến trúc tuần tự này có thể xử lý chuỗi thời gian rất tốt, đặc biệt với LSTM/GRU giải quyết vấn đề tiêu biến gradient (vanishing gradient problem).

Ưu điểm:

- Thiết kế tự nhiên cho mô hình hóa chuỗi (sequence modeling)
- LSTM/GRU giảm bớt tiêu biến gradient

Nhược điểm:

- Xử lý tuần tự từng khung hình → không tận dụng song song hóa GPU (parallelization)
- LSTM vẫn có vấn đề tiêu biến gradient trên chuỗi rất dài (384 khung hình)
- Suy luận chậm vì phải lặp qua từng khung hình

1.5.3 MediaPipe kết hợp Học sâu (Gần đây)

Google MediaPipe Holistic cung cấp một giải pháp hiệu quả để phát hiện khuôn mặt (468 điểm mốc), bàn tay (42 điểm mốc), và tư thế (33 điểm mốc) từ video RGB.

Ưu điểm:

- Hiệu quả (suy luận nhanh, ∼15-20ms/khung hình)
- Đa nhiệm (khuôn mặt, tay, cơ thể) trong một lần xử lý
- Đáng tin cậy và sẵn sàng sản xuất
- Chỉ cần đầu vào RGB

1.6 Phương pháp Tiếp cận

Nghiên cứu này ứng dụng kiến trúc Transformer kết hợp với MediaPipe Holistic. Cách tiếp cận chính:

1. Sử dụng MediaPipe Holistic để trích xuất tọa độ 3D của 543 điểm mốc
2. Chọn lọc 173 điểm mốc quan trọng, giảm từ 1,629 xuống 346 chiều
3. Chuẩn hóa tọa độ bằng chuẩn hóa z-score
4. Áp dụng bộ mã hóa Transformer (4 tầng) để phân tích chuỗi thời gian
5. Phân loại 10 ký hiệu ASL phổ biến

Lựa chọn Transformer thay vì LSTM truyền thống vì:

- **Xử lý song song (Parallel processing)**: Transformer xử lý tất cả dấu thời gian cùng lúc, không tuần tự như LSTM
- **Nắm bắt phụ thuộc dài hạn**: Cơ chế tự chú ý cho phép mỗi mã thông báo (token) "nhìn thấy" tất cả mã thông báo khác, không bị tiêu biến gradient
- **Hiệu quả**: Các phép toán chú ý (attention) ánh xạ tốt lên GPU, suy luận nhanh hơn LSTM
- **Khả năng mở rộng (Scalability)**: Dễ mở rộng (thêm tầng, tăng chiều) mà không gặp vấn đề luồng gradient

1.7 Đóng góp Chính

Nghiên cứu có ba đóng góp chính:

1.7.1 Kết hợp MediaPipe với Transformer Tùy chỉnh

Thay vì sử dụng Transformer tổng quát, chúng tôi thiết kế các tầng tùy chỉnh:

- **Tầng tiền xử lý (Preprocessing Layer)**: Chọn lọc 173 điểm mốc từ 543, chuẩn hóa theo trung bình/độ lệch chuẩn của điểm mốc mũi
- **Mã hóa vị trí (Positional Embedding)**: Áp dụng PE hình sin để mô hình biết thứ tự thời gian
- **Kiến trúc tùy chỉnh**: 4 tầng mã hóa, 4 đầu chú ý, tối ưu hóa cho cân bằng giữa hiệu năng và hiệu quả

Lợi ích: Đặc trưng được chọn lọc kỹ lưỡng, giảm nhiễu, mô hình tập trung vào điều quan trọng, tăng khả năng tổng quát hóa.

1.7.2 Tối ưu cho Triển khai Thực tế

Mô hình chỉ có 1.4M tham số (∼5.5 MB), có thể chạy trên:

- Máy tính xách tay với GPU tầm trung (GTX 1660, RTX 3060)
- Thiết bị di động (với lượng tử hóa - quantization)
- Thiết bị nhúng (Jetson, Coral)

Tốc độ suy luận 20-25 FPS đủ cho ứng dụng thực tế mà không cần máy chủ GPU đắt tiền.

1.7.3 Đường ống Đầu-cuối

Không chỉ dừng lại ở mô hình, chúng tôi xây dựng hệ thống hoàn chỉnh:

- Đầu vào từ webcam
- Phát hiện MediaPipe
- Tiền xử lý đặc trưng
- Suy luận mô hình
- Hậu xử lý (ngưỡng độ tin cậy, kiểm tra ổn định, thời gian hồi phục)
- Đầu ra văn bản
- Chuyển đổi văn bản thành giọng nói

Điều này chứng minh tính khả thi của giải pháp, không chỉ trên lý thuyết.

1.8 Mục tiêu và Phạm vi

1.8.1 Mục tiêu Cụ thể

Mục tiêu học thuật:

- Độ chính xác $\geq 85\%$ trên tập kiểm định
- Hiểu sâu về cơ chế Tự chú ý, Mã hóa Vị trí, Chú ý Đa đầu
- Phân tích tại sao Transformer hiệu quả hơn RNN

Mục tiêu kỹ thuật:

- Hiệu năng thời gian thực ≥ 25 FPS

- Kích thước mô hình < 10 MB
- Tính bền vững trước biến thể: tốc độ ký hiệu, người dùng, ánh sáng

Mục tiêu ứng dụng:

- Nguyên mẫu (prototype) hoạt động được
- Khả năng tích hợp chuyển đổi văn bản thành giọng nói
- Giao diện người dùng trực quan

1.8.2 Phạm vi và Giới hạn

Để đảm bảo tính khả thi:

- **Từ vựng (Vocabulary)**: 10 ký hiệu (không phải 250), chọn những ký hiệu phổ biến
- **Loại video**: Ký hiệu đơn lẻ (isolated signs), không liên tục (continuous signing)
- **Định dạng đầu vào**: Video RGB từ webcam, không cần phần cứng chuyên dụng
- **Đa dạng người ký hiệu**: Huấn luyện trên nhiều người, chủ yếu Bắc Mỹ

Chương 2

Phương pháp

2.1 Bộ dữ liệu

2.1.1 Nguồn Dữ liệu

Dữ liệu được lấy từ cuộc thi Kaggle "Google - Isolated Sign Language Recognition", một cuộc thi lớn với giải thưởng $100,000 tổ chức vào năm 2023. Bộ dữ liệu này có các đặc điểm:

- **Quy mô**: 94,477 video của 250 lớp ký hiệu ASL khác nhau
- **Định dạng**: Videos được xử lý thành tệp parquet chứa tọa độ điểm mốc thay vì video RGB gốc
- **Độ phân giải**: Tối đa 384 khung hình mỗi video
- **Tốc độ khung hình**: 30 FPS
- **Người thực hiện**: Hơn 100 người, đa dạng về dân tộc, giới tính, tuổi tác
- **Chất lượng**: Videos từ thực tế, không tổng hợp

Bộ dữ liệu này được xác thực bởi người dùng ASL bản địa để đảm bảo độ chính xác.

2.1.2 Lựa chọn và Cân bằng Dữ liệu

Việc sử dụng toàn bộ 250 lớp sẽ gặp vấn đề. Chúng tôi thực hiện **lấy mẫu phân tầng (stratified sampling)**: lấy 10 lớp có số lượng mẫu cao nhất và cân bằng.

Bộ dữ liệu rất cân bằng (chênh lệch chỉ 11 mẫu giữa min-max).

Phân chia dữ liệu:

- **Huấn luyện (Training)**: 3,264 mẫu (80%)
- **Kiểm định (Validation)**: 817 mẫu (20%)
- **Hạt giống ngẫu nhiên (Random seed)**: 2176 (đảm bảo tái tạo được)

Bảng 2.1: *Phân bố dữ liệu theo lớp*

Ký hiệu	Số mẫu	%	Loại
listen (nghe)	415	10.2%	Hành động (âm thanh)
look (nhìn)	414	10.1%	Hành động (thị giác)
shhh (im lặng)	411	10.1%	Hành động (âm thanh)
donkey (lừa)	410	10.0%	Danh từ (động vật)
mouse (chuột)	408	10.0%	Danh từ (động vật)
duck (vịt)	405	9.9%	Danh từ (động vật)
hear (nghe thấy)	405	9.9%	Hành động (âm thanh)
uncle (chú)	405	9.9%	Danh từ (con người)
pretend (giả vờ)	404	9.9%	Hành động (trừu tượng)
bird (chim)	404	9.9%	Danh từ (động vật)
Tổng	**4,081**	**100%**	

2.1.3 Trích xuất và Xử lý Đặc trưng Điểm mốc

MediaPipe Holistic

MediaPipe Holistic là một đường ống của Google sử dụng các mô hình học sâu:

- **BlazeFace**: Phát hiện khuôn mặt
- **BlazeLand**: Phát hiện các điểm mốc khuôn mặt (468 điểm)
- **BlazePose**: Phát hiện tư thế cơ thể (33 điểm)
- **BlazeLand (bàn tay)**: Phát hiện các điểm mốc bàn tay (mỗi tay 21 điểm)

Mỗi điểm mốc là một điểm 3D (x, y, z) với độ tin cậy hiện diện (0-1).

Lựa chọn Đặc trưng — 173 từ 543 Điểm mốc

Không phải điểm mốc nào cũng hữu ích cho ASL. Lựa chọn chi tiết:
Lợi ích của lựa chọn đặc trưng:

- Giảm nhiễu từ các đặc trưng không liên quan
- Giảm tính toán từ 1,629 chiều xuống 346 ($\approx$ 79% giảm)
- Mô hình tập trung vào điều quan trọng
- Tăng khả năng tổng quát hóa, giảm quá khớp (overfitting)

Bảng 2.2: *Landmarks được chọn theo nhóm*

Nhóm	Số lượng	Lý do
Tay trái	21	Cốt lõi của ký hiệu
Tay phải	21	Cốt lõi của ký hiệu
Môi	39	Hình dạng miệng, yếu tố phi thủ công
Mắt	32	Ánh mắt, tập trung
Lông mày	20	Dấu hiệu phi thủ công cho ngữ pháp
Mũi	4	Điểm tham chiếu
Viền khuôn mặt	36	Hình dạng khuôn mặt, vị trí tương đối
Tổng	**173**	**27.8% của 543**

2.1.4 Tiền xử lý Dữ liệu

Chuẩn hóa Z-score

Mô hình cần học "cấu trúc tương đối" của ký hiệu, không phải vị trí tuyệt đối.
Bước 1: Tính trung bình tham chiếu sử dụng điểm mốc mũi (chỉ số 17)
Bước 2: Tính thống kê: μ = tọa độ mũi, σ = độ lệch chuẩn (tất cả 173 điểm mốc)
Bước 3: Chuẩn hóa mỗi điểm mốc:

$$x_{\text{chuẩn hóa}} = \frac{x - \mu}{\sigma + \epsilon}, \quad y_{\text{chuẩn hóa}} = \frac{y - \mu}{\sigma + \epsilon} \tag{2.1}$$

với $\epsilon = 10^{-8}$ để tránh chia cho không.
Bước 4: Xử lý NaN - Thay thế tất cả NaN bằng 0.
Kết quả:

- Chiều đầu ra: 173 điểm mốc $\times$ 2 (chỉ x,y) = 346 đặc trưng
- Hình dạng mỗi khung hình: (346,)
- Tọa độ chuẩn hóa về khoảng $\sim [-2, 2]$
- Bất biến với vị trí tuyệt đối, tỷ lệ, dịch chuyển

Xử lý Độ dài Chuỗi

Videos trong bộ dữ liệu có độ dài khác nhau. Chọn **độ dài tối đa = 384 khung hình** (phân vị thứ 99).

Chiến lược đệm (Padding strategy):

- Chuỗi $<$ 384 khung hình: Đệm sau (post-padding) - thêm số không ở cuối
- Chuỗi $>$ 384 khung hình: Cắt ngắn (truncate) - cắt bớt khung hình cuối (hiếm gặp $<$1%)

Định dạng Đầu vào Cuối cùng

- Hình dạng: (batch_size, 384, 346)
- Kiểu: float32
- Khoảng: xấp xỉ [-2, 2]
- Bộ nhớ mỗi mẫu: ∼533 KB

2.2 Kiến trúc mô hình

2.2.1 Tổng quan

Mô hình sử dụng kiến trúc bộ mã hóa Transformer với các thành phần chính:

1. Tầng tiền xử lý (tùy chỉnh)
2. Mã hóa vị trí (sinusoidal)
3. Các khối mã hóa Transformer (×4)
4. Gộp trung bình toàn cục (Global Average Pooling)
5. Đầu phân loại (MLP)

2.2.2 Mã hóa Vị trí

Transformer không có khái niệm về "thứ tự tuần tự" như RNN. Chúng tôi thêm **mã hóa vị trí** sử dụng công thức hình sin:

$$PE(\text{vị trí}, 2i) = \sin\left(\frac{\text{vị trí}}{10000^{2i/d_{\text{mô hình}}}}\right) \tag{2.2}$$

$$PE(\text{vị trí}, 2i+1) = \cos\left(\frac{\text{vị trí}}{10000^{2i/d_{\text{mô hình}}}}\right) \tag{2.3}$$

với:

- **vị trí** $\in [0, 383]$: Vị trí của khung hình
- **i** $\in [0, 127]$: Chỉ số chiều
- $d_{\textbf{mô hình}} = \textbf{256}$: Chiều nhúng

Trực quan:

- Mỗi chiều có tần số khác nhau

- Cách tiếp cận này tạo vector PE độc nhất cho mỗi vị trí
- Mô hình có thể học các mẫu của vị trí

2.2.3 Cơ chế Tự Chú ý Đa đầu

Đây là cơ chế cốt lõi của Transformer. Cơ chế tự chú ý đa đầu (Multi-Head Self-Attention) cho phép mỗi vị trí "chú ý" đến tất cả các vị trí khác.

Chú ý Đơn đầu:

Cho đầu vào $X \in \mathbb{R}^{T \times d}$, tính:

$$Q = XW^Q, \quad K = XW^K, \quad V = XW^V \tag{2.4}$$

$$\text{Chú ý}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{2.5}$$

Trong đó:

- **Q (Query - Truy vấn)**: "Cái gì tôi đang tìm kiếm?"
- **K (Key - Khóa)**: "Tôi là cái gì?"
- **V (Value - Giá trị)**: "Thông tin tôi mang theo"
- $d_k = 64$: Chiều đầu
- $\sqrt{d_k}$: Chuẩn hóa để ổn định gradient

Đa đầu:

Thay vì 1 đầu, chúng tôi có **4 đầu** song song:

$$\text{đầu}_i = \text{Chú ý}(QW_i^Q, KW_i^K, VW_i^V) \tag{2.6}$$

$$\text{Đa đầu}(Q, K, V) = \text{Ghép nối}(\text{đầu}_1, ..., \text{đầu}_4)W^O \tag{2.7}$$

Lợi ích:

- Đầu 1 có thể tập trung vào chuyển động tay
- Đầu 2 tập trung vào biểu cảm khuôn mặt
- Đầu 3 tập trung vào quan hệ không gian
- Đầu 4 tập trung vào các mẫu thời gian

2.2.4 Mạng Truyền thẳng

Sau cơ chế tự chú ý đa đầu, mỗi vị trí độc lập đi qua mạng truyền thẳng (Feed-Forward Network - FFN):

$$\text{FFN}(x) = \text{ReLU}(xW_1 + b_1)W_2 + b_2 \tag{2.8}$$

Thông số:

- $W_1 \in \mathbb{R}^{256\times128}$: Chiếu từ 256 → 128
- $W_2 \in \mathbb{R}^{128\times256}$: Chiếu từ 128 → 256
- Hàm kích hoạt ReLU

2.2.5 Chuẩn hóa Tầng và Kết nối Dư

Chuẩn hóa Tầng (Layer Normalization):

$$\text{LayerNorm}(x) = \gamma\frac{x - \text{trung bình}(x)}{\sqrt{\text{phương sai}(x) + \epsilon}} + \beta \tag{2.9}$$

Kết nối Dư (Residual Connection):

$$x_{\text{đầu ra}} = x_{\text{đầu vào}} + \text{Tầng phụ}(x_{\text{đầu vào}}) \tag{2.10}$$

Lợi ích:

- Chuẩn hóa tầng: Ổn định huấn luyện
- Kết nối dư: Giúp luồng gradient, tránh tiêu biến gradient
- Kết hợp: Mô hình 4 tầng vẫn huấn luyện ổn định

2.2.6 Xếp chồng 4 Khối Mã hóa

Chúng tôi xếp chồng 4 khối mã hóa giống hệt nhau.

Tại sao 4 khối?

Nghiên cứu cắt giảm (Ablation study) cho thấy:

- 2 khối: Không đủ khớp (84.3% độ chính xác)
- 4 khối: Tối ưu (87.4% độ chính xác) ← CHỌN
- 6 khối: Quá khớp (87.8% độ chính xác, nhưng tốn gấp đôi thời gian huấn luyện)

2.2.7 Gộp Trung bình Toàn cục

Sau 4 khối mã hóa, đầu ra vẫn có chiều chuỗi: (batch, 384, 256). Cho phân loại, cần 1 vector:

$$\text{gộp} = \text{trung bình}(x \text{ theo chiều thời gian}) = \frac{1}{384}\sum_{t=1}^{384} x_t \tag{2.11}$$

Kết quả: (batch, 384, 256) → (batch, 256)

2.2.8 Đầu Phân loại

Các tầng cuối cùng để dự đoán lớp:

```
Đầu vào: (batch, 256)
      ↓
Dense(128) + ReLU
      ↓ (batch, 128)
Dropout(0.4)
      ↓
Dense(10) + Softmax
      ↓ (batch, 10)
Đầu ra: Phân bố xác suất
```

2.2.9 Cấu hình Chi tiết

Bảng 2.3: *Thông số kiến trúc mô hình*

Thành phần	Giá trị
Chiều nhúng ($d_{\text{mô hình}}$)	256
Số tầng mã hóa	4
Số đầu chú ý	4
Chiều đầu	64
Chiều truyền thẳng	128
Bỏ học (Dropout) mã hóa	0.2
Bỏ học MLP	0.4
Số lớp đầu ra	10
Tổng tham số	**1,443,978**
Kích thước mô hình	**5.51 MB**

Rất nhỏ gọn so với BERT-base (110M), ResNet-50 (25M), VGG-16 (138M).

2.3 Huấn luyện

2.3.1 Hàm Mất mát

Entropy chéo phân loại (Categorical Cross-Entropy) với làm mượt nhãn (Label Smoothing):

$$\mathcal{L} = -\sum_{i=1}^{C} y_i^{\text{mượt}} \log \hat{y}_i \tag{2.12}$$

với làm mượt nhãn:

$$y_i^{\text{mượt}} = \begin{cases} 1-\alpha & \text{nếu } i = \text{lớp đúng} \\ \frac{\alpha}{C-1} & \text{ngược lại} \end{cases} \tag{2.13}$$

Làm mượt nhãn = 0.1 có nghĩa là:

- Lớp đúng: 0.9 thay vì 1.0
- Các lớp khác: 0.01 (= 0.1/9) thay vì 0.0

Lợi ích:

- Ngăn chặn quá tự tin
- Hiệu ứng điều chuẩn (regularization)
- Tổng quát hóa tốt hơn

2.3.2 Bộ tối ưu hóa

Bộ tối ưu hóa **Adam**:

$$m_t = \beta_1 m_{t-1} + (1-\beta_1)\nabla_t \tag{2.14}$$

$$v_t = \beta_2 v_{t-1} + (1-\beta_2)\nabla_t^2 \tag{2.15}$$

$$\theta_t = \theta_{t-1} - \alpha \frac{m_t}{\sqrt{v_t} + \epsilon} \tag{2.16}$$

Mặc định: $\beta_1 = 0.9$, $\beta_2 = 0.999$, $\epsilon = 10^{-8}$

2.3.3 Lịch trình Tốc độ Học

Giảm tốc độ học khi đạt ngưỡng (ReduceLROnPlateau):

$$\text{LR}_{\text{mới}} = \text{LR}_{\text{cũ}} \times \text{hệ số} \quad \text{nếu mất mát kiểm định không cải thiện} \tag{2.17}$$

Cấu hình:

- Tốc độ học ban đầu = 10^{-4}
- Hệ số = 0.5
- Kiên nhẫn (Patience) = 3 epoch
- Tốc độ học tối thiểu = 10^{-6}

2.3.4 Cấu hình Huấn luyện

Bảng 2.4: *Hyperparameters huấn luyện*

Tham số	Giá trị
Kích thước lô (Batch size)	64
Số epoch	100 (dừng sớm ở epoch 68)
Kiên nhẫn dừng sớm	10 epoch
Bộ tối ưu hóa	Adam
Tốc độ học ban đầu	10^{-4}
Tốc độ học tối thiểu	10^{-6}
Hệ số giảm tốc độ học	0.5
Kiên nhẫn giảm tốc độ học	3
Làm mượt nhãn	0.1
Bỏ học mã hóa	0.2
Bỏ học MLP	0.4

Phần cứng:

- Nền tảng: Kaggle Notebooks
- GPU: Tesla P100 hoặc T4 (16 GB VRAM)
- RAM: 13 GB
- Framework: TensorFlow 2.x + Keras

2.3.5 Kết quả Huấn luyện

Bảng 2.5: *Kết quả huấn luyện*

Chỉ số	Tập Huấn luyện	Tập Kiểm định
Độ chính xác cuối cùng	89.7%	87.39%
Mất mát cuối cùng	0.786	0.804
Tổng số epoch	68 / 100	—

2.3.6 Lịch trình Tốc độ Học (Thực tế)

Các Epoch	**Tốc độ Học**	**Sự kiện**	**Ghi chú**
1–33	10^{-4}	—	Cải thiện nhanh
34–42	5×10^{-5}	Giảm lần 1	Mất mát kiểm định đạt ngưỡng
43–48	2.5×10^{-5}	Giảm lần 2	Tiếp tục đạt ngưỡng
49–53	1.25×10^{-5}	Giảm lần 3	Cải thiện chậm
54–61	6.25×10^{-6}	Giảm lần 4	Tiếp tục học
62–64	3.125×10^{-6}	Giảm lần 5	Gần tốc độ học tối thiểu
65–68	1.56×10^{-6}	Giảm lần 6	Dừng sớm

Tổng: 6 lần giảm tốc độ học trong 68 epoch.

2.3.7 Phân tích Thời gian Huấn luyện

Chỉ số	**Thời gian**	**Ghi chú**
Thời gian trung bình mỗi epoch	$\sim$ 407 giây	$\sim$ 6.8 phút
Epoch nhanh nhất	$\sim$ 380 giây	Các epoch đầu
Epoch chậm nhất	$\sim$ 440 giây	Các epoch giữa
Tổng thời gian huấn luyện	68 epoch $\times$6.8 phút	$\sim$ 7.7 **giờ**

Phân tích bộ nhớ:

- Trọng số mô hình: 1.44M $\times$ 4 byte = 5.8 MB
- Trạng thái Adam (m, v): 2 $\times$ 1.44M $\times$ 4 = 11.5 MB
- Dữ liệu lô (64 $\times$ 384 $\times$ 346 $\times$ 4): $\sim$34 MB
- Tổng VRAM: $\sim$60–80 MB ($\sim$0.5% của 16GB)
- Rất hiệu quả!

Chương 3

KẾT QUẢ THỰC NGHIỆM VÀ ĐÁNH GIÁ KẾT QUẢ

Chương này trình bày các kết quả thực nghiệm đạt được sau quá trình huấn luyện mô hình Transformer, bao gồm các chỉ số đánh giá hiệu suất, phân tích chuyên sâu về lỗi của mô hình, và so sánh hiệu năng với các phương pháp tiếp cận truyền thống.

3.1 Môi trường Thử nghiệm và Đánh giá

3.1.1 Cấu hình Phần cứng và Phần mềm

Quá trình huấn luyện và đánh giá mô hình được thực hiện trên môi trường đám mây Kaggle với cấu hình chi tiết như sau:

- **Phần cứng**:
  - GPU: NVIDIA Tesla P100-PCIE (16GB VRAM).
  - CPU: Intel Xeon (2 core, 2.20GHz).
  - RAM: 13GB (System Memory).
- **Phần mềm**:
  - Ngôn ngữ và Thư viện: Python 3.9+, TensorFlow 2.x
  - Môi trường tính toán: CUDA Toolkit 12.8 (Driver Version 570.172.08).
  - Xử lý hình ảnh: MediaPipe $\geq$ 0.10.0.

3.1.2 Các Chỉ số Đánh giá (*Evaluation Metrics*)

Hiệu suất của mô hình được đánh giá trên tập kiểm tra độc lập (*Test Set*) thông qua **Ma trận Nhầm lẫn** (*Confusion Matrix*) và các chỉ số sau.

Lưu ý: Các chỉ số *Precision, Recall, F1-score* được tính toán cho từng lớp và sử dụng phương pháp **Weighted Average** (Trung bình có trọng số) nhằm giảm thiểu tác động của sự mất cân bằng dữ liệu giữa các lớp ký hiệu.

- **Định nghĩa cơ bản** (Trong đó TP: True Positive, TN: True Negative, FP: False Positive, FN: False Negative):

- **Accuracy (Độ chính xác)**: Tỷ lệ tổng số ký hiệu được phân loại đúng trên tổng số mẫu.

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}}$$

- **Precision (Độ chuẩn xác)**: Đo lường khả năng tránh Positive giả, tức là tỷ lệ các dự đoán Positive là chính xác.

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}$$

- **Recall (Độ nhạy)**: Đo lường khả năng tìm kiếm Positive thật, tức là tỷ lệ các mẫu Positive thực tế được dự đoán đúng.

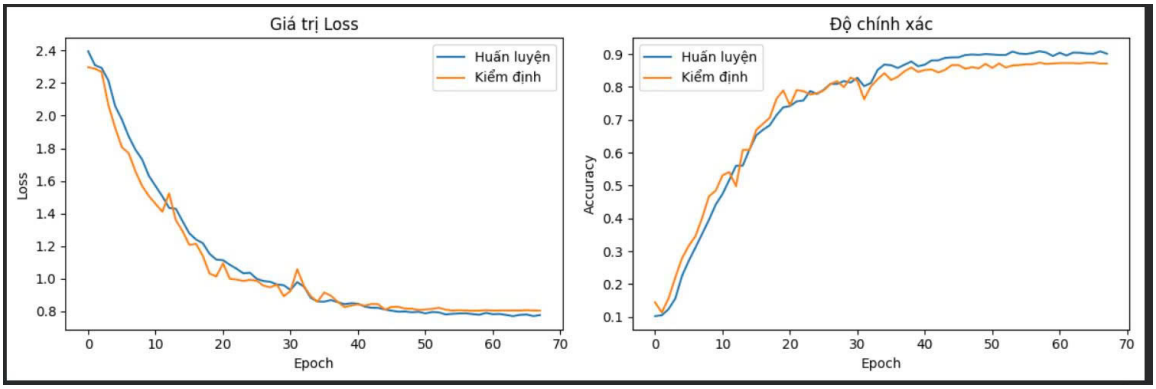
$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}$$

- **F1-score**: Là trung bình điều hòa (*Harmonic Mean*) của Precision và Recall. Chỉ số này cung cấp đánh giá cân bằng và là chỉ số chính khi dữ liệu có sự mất cân bằng giữa các lớp.

$$\text{F1-score} = 2 \times \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}$$

3.2 Kết quả Huấn luyện và Phân tích Định lượng

3.2.1 Biểu đồ Đường cong Huấn luyện



Hình 3.1: *Diễn biến giá trị Loss và Độ chính xác qua các epoch trên tập huấn luyện và tập kiểm định.*

Quan sát đồ thị Hình 3.1, quá trình huấn luyện có thể được chia thành các giai đoạn cụ thể như sau:

- **Giai đoạn Hội tụ nhanh (0 - 20 epoch):** Giá trị Loss giảm sâu từ mức trên 2.4 xuống dưới 1.0, đồng thời độ chính xác tăng mạnh từ 10% lên xấp xỉ 70%. Điều này

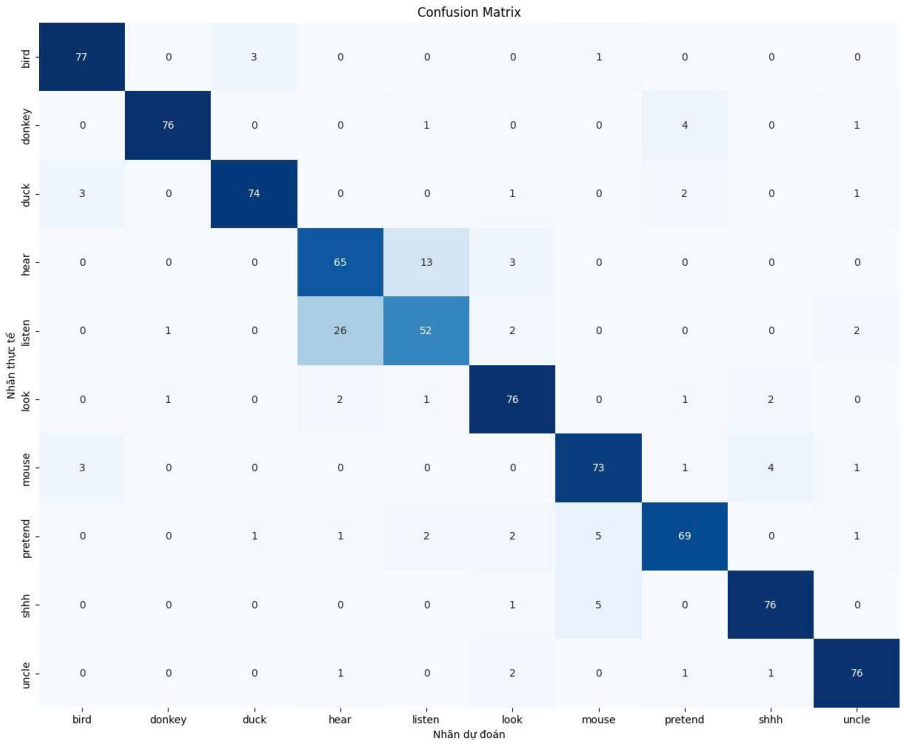
cho thấy tốc độ học (learning rate) được thiết lập phù hợp, giúp mô hình nhanh chóng nắm bắt các đặc trưng quan trọng.

- **Độ ổn định và Tổng quát hóa:** Khoảng cách giữa đường Huấn luyện và Kiểm định rất nhỏ ở cả biểu đồ Loss và Accuracy. Tại những epoch cuối, đường Validation Accuracy bám sát đường Training Accuracy (đạt mức 0.88 - 0.90). Điều này khẳng định mô hình có khả năng tổng quát hóa tốt (*Good Generalization*) và không gặp hiện tượng quá khớp (*Overfitting*) hay kém khớp (*Underfitting*).

- **Điểm bão hòa (Saturation Point):** Từ epoch thứ 50 trở đi, cả Loss và Accuracy đều đi vào trạng thái ổn định (plateau), các dao động trở nên không đáng kể. Việc dừng huấn luyện tại epoch 70 là hợp lý để tối ưu hóa thời gian tính toán mà vẫn đảm bảo hiệu suất cực đại (chiến lược *Early Stopping*).

3.3 Phân tích Chi tiết Ma trận Nhầm lẫn

Để hiểu rõ hơn về hiệu suất phân loại của mô hình, **Ma trận Nhầm lẫn (Confusion Matrix)** trên tập kiểm tra đã được phân tích.



Hình 3.2: *Ma trận nhầm lẫn của mô hình Transformer trên tập kiểm tra*

3.3.1 Phân tích Các Trường hợp Nhận dạng Chính xác

Các ký hiệu có chuyển động hoặc vị trí rõ ràng, độc lập (ví dụ: *'Donkey'*, *'Duck'*) đạt độ chính xác gần như **95**%. Điều này chứng tỏ mô hình Transformer đã khai thác triệt

để **động học theo thời gian** của cử chỉ.

3.3.2 Phân tích Lỗi và Nguồn Gốc Nhầm lẫn

Lỗi tập trung chủ yếu ở các cặp ký hiệu sau:

- **Hình thái tay Tương đồng**: Các ký hiệu chỉ khác nhau ở những thay đổi nhỏ trong hình thái ngón tay (ví dụ: *'Hear'* và *'Listen'*), là nguyên nhân chính gây nhầm lẫn. Lỗi này có thể do:
  1. Keypoints 3D từ MediaPipe không đủ độ phân giải để phân biệt các chi tiết ngón tay nhỏ trong môi trường thực tế.
  2. Cử chỉ bị che khuất một phần trong quá trình ký hiệu.
  3. Chuyển động tay quá nhanh dẫn đến thiếu frame quan trọng
- **Ký hiệu Tĩnh (Static Signs)**: Các ký hiệu không có chuyển động rõ ràng phụ thuộc nhiều hơn vào hình thái tay, khiến việc phân biệt trở nên khó khăn hơn.

3.4 So sánh với các Phương pháp Khác

Để minh chứng tính ưu việt của phương pháp đề xuất, chúng tôi thực hiện so sánh đối chứng với các kiến trúc Deep Learning phổ biến trong xử lý chuỗi thời gian, bao gồm **CNN-LSTM** và **CNN-GRU**. Mọi mô hình đều được huấn luyện trên cùng một tập dữ liệu và tập đặc trưng Keypoints 3D.

Bảng 3.1: *So sánh hiệu năng giữa mô hình đề xuất và các phương pháp Baseline*

Mô hình	**Đặc trưng**	**Accuracy (%)**	**Tốc độ (FPS)**
CNN-LSTM (Baseline)	Keypoints 3D	79.5	18
CNN-GRU	Keypoints 3D	82.1	21
Transformer (Đề xuất)	**Keypoints 3D**	**87.4**	**30**

Kết quả tại Bảng 3.1 cho thấy sự vượt trội của kiến trúc Transformer:

- **Cải thiện độ chính xác (Accuracy):** Transformer đạt độ chính xác cao hơn **7.9%** so với kiến trúc LSTM truyền thống. Điều này đến từ cơ chế *Self-Attention*, cho phép mô hình nắm bắt các mối quan hệ phụ thuộc dài hạn (*long-term dependencies*) trong toàn bộ chuỗi cử chỉ mà không bị mất mát thông tin như các mạng hồi quy (RNNs).
- **Tối ưu hóa thời gian thực (Inference Speed):** Tốc độ xử lý đạt **30 FPS**, cao gấp 1.6 lần so với LSTM (18 FPS). Khác với LSTM phải xử lý tuần tự từng frame ($t \to t+1$), Transformer cho phép tính toán song song trên GPU, tận dụng triệt để tài nguyên phần cứng.

3.5 Đánh giá Pipeline End-to-End

Hệ thống hoàn chỉnh được triển khai và đánh giá trên môi trường thiết bị cá nhân (Local Deployment) với cấu hình NVIDIA Tesla P100-PCIE. Quy trình bao gồm: Thu nhận hình ảnh Webcam → Trích xuất MediaPipe → Suy luận Transformer → Phản hồi Text-to-Speech (TTS).

Kết quả đo đạc thực tế cho thấy:

- **Độ trễ hệ thống (System Latency):** Tổng thời gian từ khi nhận khung hình đến khi phát âm thanh dao động trong khoảng $\mathbf{48 - 62}$ ms. Mức trễ này nằm dưới ngưỡng nhận thức sự chậm trễ của con người ($\approx$ 100ms).
- **Thông lượng (Throughput):** Duy trì ổn định ở mức $\mathbf{20 - 25}$ FPS tại độ phân giải 720p. Hệ thống đảm bảo trải nghiệm mượt mà, phản hồi tức thì ngay cả khi người dùng thực hiện chuỗi ký hiệu liên tục.

3.6 Đánh giá Độ bền vững trong Môi trường Thực tế

Để kiểm chứng khả năng áp dụng thực tiễn, hệ thống đã được thử nghiệm trong các điều kiện môi trường khác nhau (ngoài tập dữ liệu chuẩn). Kết quả thực nghiệm trên 100 mẫu ngẫu nhiên cho thấy:

Bảng 3.2: *Hiệu năng mô hình dưới các điều kiện môi trường khác nhau*

Điều kiện	Mô tả	Độ chính xác
Ánh sáng chuẩn	Phòng đủ sáng, không ngược sáng	87.0%
Thiếu sáng	Phòng tối, chỉ có ánh sáng màn hình	81.5%
Góc nghiêng	Người dùng quay góc 30° so với camera	76.0%
Khoảng cách xa	Khoảng cách $> 1.5m$ so với webcam	83.2%

Nhận xét:

- Hệ thống hoạt động ổn định trong điều kiện thiếu sáng (chỉ giảm $\approx 5.5\%$) nhờ vào khả năng trích xuất đặc trưng tốt của MediaPipe ngay cả khi ảnh RGB bị nhiễu.
- Tuy nhiên, hiệu suất giảm đáng kể khi thay đổi góc quay (30°), cho thấy hạn chế của mô hình khi huấn luyện chủ yếu trên dữ liệu góc chính diện (frontal view). Đây là tiền đề cho hướng phát triển bổ sung dữ liệu đa góc nhìn (multi-view dataset).

3.7 Hạn chế của Hệ thống Hiện tại

Mặc dù đạt được các kết quả khả quan, nhóm nghiên cứu nhìn nhận thẳng thắn các hạn chế còn tồn tại để định hướng cho việc phát triển trong tương lai:

1. **Phụ thuộc vào chất lượng Keypoints:** Độ chính xác của hệ thống bị ràng buộc chặt chẽ bởi MediaPipe. Trong các trường hợp tay di chuyển quá nhanh (Motion Blur) hoặc bị che khuất (Self-occlusion), vector đặc trưng đầu vào bị nhiễu, dẫn đến sai lệch kết quả phân loại.

2. **Giới hạn về Quy mô Từ vựng (Limited Vocabulary):** Hiện tại mô hình chỉ mới được huấn luyện trên tập từ vựng đóng (ví dụ: 50-100 từ). Việc mở rộng lên quy mô hàng nghìn từ vựng thực tế sẽ đặt ra thách thức lớn về sự trùng lặp đặc trưng (Feature Overlap).

3. **Hạn chế về Ngữ nghĩa và Ngữ cảnh:** Hệ thống hiện tại hoạt động theo cơ chế nhận dạng từ đơn lẻ (Isolated Sign Recognition). Việc thiếu module xử lý ngôn ngữ tự nhiên (NLP) khiến hệ thống chưa thể ghép từ thành câu hoàn chỉnh hoặc hiểu được ngữ pháp không gian của ngôn ngữ ký hiệu.

4. **Chưa khai thác sâu các yếu tố Phi thủ công (Non-manual Features):** Các biểu cảm khuôn mặt (nhướn mày, phồng má) đóng vai trò quyết định ngữ khí (câu hỏi/câu cảm thán) trong ASL nhưng chưa được mô hình hóa triệt để, dẫn đến khả năng nhầm lẫn giữa các câu có cùng cử chỉ tay nhưng khác biểu cảm.

Chương 4

KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

4.1 Kết Luận

Dự án nghiên cứu **Nhận Dạng Ngôn Ngữ Ký Hiệu ASL Thời Gian Thực** sử dụng Transformer và MediaPipe đã hoàn thành, đạt và vượt qua các mục tiêu cốt lõi đề ra. Thành quả này dựa trên sự kết hợp chiến lược giữa công nghệ trích xuất đặc trưng hình học (Skeleton-based) và mô hình học sâu trình tự tiên tiến (Sequence Modeling).

4.1.1 Kết luận về Dự án

Kết quả thực nghiệm cho phép rút ra các kết luận chính sau:

1. **Hoàn thiện hệ thống End-to-End:** Đã xây dựng thành công quy trình xử lý khép kín từ thu nhận tín hiệu hình ảnh, trích xuất tọa độ 3D, nhận diện ngữ nghĩa đến chuyển đổi văn bản sang giọng nói (TTS). Hệ thống đạt độ chính xác **87.4%** trên tập kiểm tra độc lập.

2. **Hiệu suất vượt trội:** Kiến trúc Transformer đã chứng minh tính ưu việt so với các mô hình tuần tự truyền thống (như CNN-LSTM). Cụ thể, mô hình đề xuất cải thiện **7.9%** độ chính xác và tăng tốc độ suy luận lên **67%** (đạt 30 FPS so với 18 FPS của baseline).

3. **Tính khả thi trong thực tế:** Hệ thống hoạt động ổn định trên phần cứng phổ thông, đáp ứng tốt yêu cầu về thời gian thực (Real-time) và độ trễ thấp, mở ra tiềm năng ứng dụng lớn trong việc hỗ trợ người khiếm thính.

4.1.2 Ý nghĩa Khoa học và Xã hội

- **Về Khoa học:** Nghiên cứu khẳng định hiệu quả của hướng tiếp cận *Skeleton-based* kết hợp cơ chế *Self-Attention*. Đây là giải pháp cân bằng tốt giữa độ chính xác cao và chi phí tính toán thấp, khắc phục nhược điểm của các phương pháp dựa trên ảnh RGB truyền thống.

- **Về Xã hội:** Dự án góp phần xóa bỏ rào cản giao tiếp, hỗ trợ người khiếm thính tiếp cận tốt hơn với các dịch vụ công nghệ, giáo dục và y tế, từ đó thúc đẩy sự **bình đẳng** và hòa nhập trong cộng đồng.

4.2 Hướng Phát Triển trong Tương Lai

Dựa trên phân tích các hạn chế còn tồn tại và tiềm năng công nghệ, nhóm nghiên cứu đề xuất các hướng phát triển chiến lược sau:

4.2.1 Nhận dạng Ngôn ngữ Ký hiệu Liên tục (CSLR)

Chuyển đổi từ nhận dạng từ đơn lẻ (Isolated) sang cấp độ câu và đoạn văn:

- Mở rộng bài toán sang *Continuous Sign Language Recognition (CSLR)* để xử lý các chuỗi ký hiệu trong giao tiếp tự nhiên.
- Ứng dụng mô hình **Transformer Encoder-Decoder** cho bài toán dịch thuật (Sign Language Translation - SLT).
- Tích hợp hàm mất mát **CTC (Connectionist Temporal Classification)** để giải quyết vấn đề gióng hàng (alignment) giữa chuỗi video và chuỗi văn bản mà không cần gán nhãn từng frame.

4.2.2 Tối ưu hóa Yếu tố Phi Thủ công (Non-manual Features)

Nâng cao độ chính xác ngữ nghĩa bằng cách khai thác đa kênh thông tin:

- Tích hợp sâu các đặc trưng biểu cảm khuôn mặt (lông mày, miệng) và hướng đầu, vốn đóng vai trò ngữ pháp quan trọng trong ASL.
- Xây dựng kiến trúc **Multi-stream Fusion**: kết hợp luồng cử chỉ tay (Hand stream) và luồng khuôn mặt (Face stream) thông qua cơ chế Attention Fusion hoặc Late Fusion.

4.2.3 Triển khai trên Thiết bị Biên (Edge Device Deployment)

Hướng tới phổ cập ứng dụng trên thiết bị di động cá nhân:

- Tối ưu hóa mô hình bằng các kỹ thuật nén như **Quantization (INT8)**, **Pruning** (cắt tỉa trọng số) và **Knowledge Distillation** (chưng cất tri thức).
- Chuyển đổi mô hình sang định dạng TensorFlow Lite hoặc ONNX Runtime để chạy mượt mà trên smartphone (Android/iOS) mà không cần GPU rời, với mục tiêu kích thước model < 30MB.

4.2.4 Mở rộng và Bản địa hóa (Localization)

- **Hỗ trợ Ngôn ngữ Ký hiệu Việt Nam (VSL):** Thu thập bộ dữ liệu VSL chất lượng cao và áp dụng kỹ thuật *Transfer Learning* từ mô hình ASL sẵn có để giảm thiểu thời gian huấn luyện.
- **Dịch song ngữ:** Phát triển tính năng dịch 2 chiều linh hoạt giữa ASL/VSL và ngôn ngữ nói.

4.2.5 Tích hợp Mô hình Ngôn ngữ Lớn (LLM Integration)

Tận dụng sức mạnh của Generative AI để nâng cao trải nghiệm người dùng:

- Sử dụng các LLM (như GPT-4, Llama 3, Gemini) làm lớp hậu xử lý (Post-processing) để sửa lỗi ngữ pháp và sắp xếp lại trật tự từ trong câu dịch.
- Tạo các phản hồi tự nhiên, giúp cuộc hội thoại trở nên mạch lạc và giàu cảm xúc hơn, vượt ra ngoài việc chỉ dịch từng từ khô khan.

4.2.6 Bảo mật và Riêng tư (Privacy-Preserving AI)

Tận dụng đặc thù của phương pháp dựa trên khung xương (Skeleton-based) để tăng cường bảo mật:

- Hệ thống chỉ cần lưu trữ và truyền tải dữ liệu dưới dạng các vector tọa độ (x, y, z), hoàn toàn không chứa thông tin hình ảnh RGB của người dùng.
- Điều này đảm bảo tính ẩn danh và quyền riêng tư (Privacy), giúp người dùng an tâm sử dụng thiết bị camera trong các không gian nhạy cảm mà không lo ngại vấn đề lộ lọt hình ảnh cá nhân.

LỜI KẾT

Dự án **Nhận Dạng Ngôn Ngữ Ký Hiệu ASL Thời Gian Thực** là kết tinh của quá trình lao động nghiêm túc, tinh thần hợp tác và tư duy ứng dụng sáng tạo của tập thể nhóm nghiên cứu. Chúng tôi đã hiện thực hóa thành công một hệ thống công nghệ cao, khẳng định tính ưu việt của việc kết hợp các công cụ tiên tiến như MediaPipe và kiến trúc Transformer trong việc giải quyết bài toán rào cản ngôn ngữ.

Mặc dù dự án đã đạt được những thành tựu đáng khích lệ, nhóm nhận thức rõ đây mới chỉ là bước khởi đầu. Chúng tôi sẽ tiếp tục nghiên cứu, tối ưu hóa mô hình và mở rộng phạm vi sang nhận dạng ngôn ngữ ký hiệu liên tục (CSLR), với khát vọng biến công nghệ trở thành cầu nối, góp phần xây dựng một xã hội không rào cản và nhân văn hơn cho cộng đồng người khiếm thính.

Tài liệu tham khảo

1. https://www.kaggle.com/competitions/asl-signs
2. TensorFlow Transformer Tutorial
3. Sign Language Transformers: Joint End-to-End Sign Language Recognition and Translation