ĐẠI HỌC QUỐC GIA TPHCM

TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN

KHOA CÔNG NGHỆ THÔNG TIN

Phân tích bài báo

Đề tài: AST: Audio Spectrogram Transformer

Môn học: Nhập môn học máy

Sinh viên thực hiện:

Tô Hữu Danh (23127336)

Nguyễn Đăng Hưng (23127050)

Lê Phú Cường (23127164)

Nguyễn Bá Đăng Khoa (23127392)

Giáo viên hướng dẫn:

Thầy Võ Nhật Tân

Ngày 29 tháng 11 năm 2025

Mục lục

Danh sách bảng

Danh sách hình vẽ

1 Bài toán giải quyết. Đầu vào, đầu ra

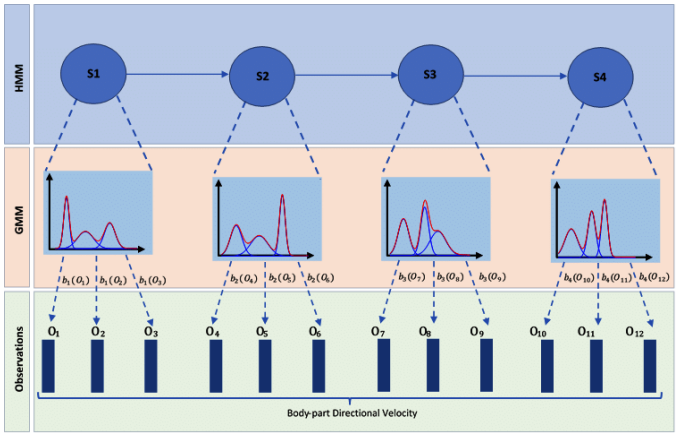
2 Công trình liên quan, quá trình phát triển

2.1 Hidden Markov Models và Gaussian Mixture Models [1]

Là phương pháp phổ biến nhất, mạnh nhất những năm 1980 đến tận trước 2010. Người ta tiền xử lý dữ liệu waveform của audio thủ công bằng các thuật toán như Fourier Transform, Cepstral Analysis để tạo ra các thuộc tính như MFCC (*Mel-Frequency Cepstral Coefficient*).

Về MFCC, đây là một loại đặc trưng âm thanh được sử dụng làm đầu vào cho mô hình GMM/HMM, nó chuyển đổi dữ liệu waveform thành một tập hợp các con số (thường là 12-40 hệ số) mà máy tính có thể phân tích, nhưng được lọc để gắn với cách tai người cảm nhận âm thanh hơn. Quá trình gồm tách khung, biến đổi Fourier (để xem tần số), lọc bằng bộ lọc Mel (làm nổi bật các tần số thấp mà tai người nhạy cảm), biến đổi (logarit, sau đó biến đổi cosin rời rạc (DCT)) sau cùng tạo ra các hệ số MFCC.

GMM (*Gaussian Mixture Model*) là mô hình dùng để gán một xác suất cho một âm thanh ví dụ một đoạn âm thanh nghe giống "A" hay "B". HMM chia đoạn âm thanh đó thành nhiều state (số lượng state do người cài đặt), ứng với mỗi state GMM sẽ có một hàm b tương ứng để tính toán xác suất mà một vector MFCC thuộc về state đó (ví dụ b1(O1) là xác suất mà vector O1 thuộc về s1). HMM (*Hidden Markov Model*) là thành phần gom các vector về các state để tổ chức trình tự cho chúng bằng cách tính toán xác suất chuyển đổi từ trạng thái này sang trạng thái khác. Sau đó HMM trả về một bộ xác suất để giúp GMM đoán được chính xác âm thanh kế tiếp. Ví dụ kế âm "M" thường sẽ là âm "ơ", "a", "e" chứ không thể là âm "n", "k", "m".



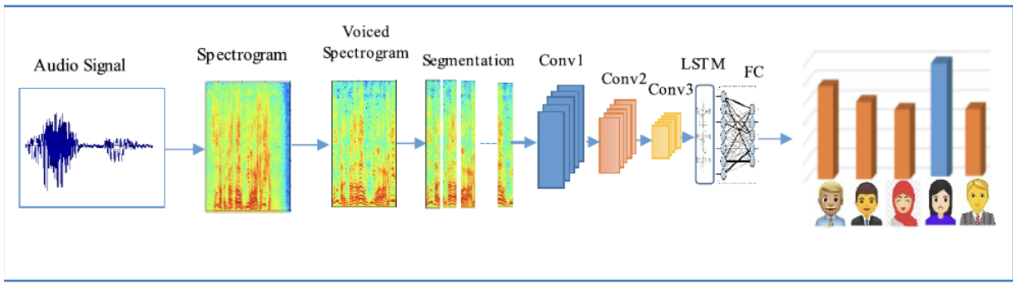
Hình 1: Ví dụ minh hoạ cho GMM-HMM. Bài toán nhận diện hành động của con người. [2]

2.2 Mô hình Hybrid LST-CNN Attention

Nền tảng của CNN (*Convolutional Neural Network*) xuất hiện vào năm 1989 trong bài báo [3], tại đây nhóm tác giả đã huấn luyện thành công các lớp tích chập đầu tiên bằng thuật toán lan truyền ngược, dùng cho bài toán nhận diện chữ viết. Đến năm 1988, kiến trúc LeNet-5 được hoàn thiện và trở thành tiêu chuẩn cho CNN vào thời điểm đó.

Đến năm 2012, AlexNet ra đời và lần đầu tiên triển khai ý tưởng biến âm thanh thành hình ảnh thông qua kỹ thuật Spetrogram rồi sử dụng CNN để xử lý. Mô hình AlexNet với kiến trúc sâu hơn hay nhiều lớp tích chập hơn, sử dụng hàm kích hoạt ReLU thay cho các hàm kích hoạt phổ biến, đồng thời kết hợp thêm cơ chế Dropout để chống overfitting. Từ đó AlexNet cải thiện được đáng kể và đạt tỉ lệ lỗi cực thấp 15.3% trong cuộc thi ILSVRC (*ImageNet Large Scale Visual Recognition Challenge*).

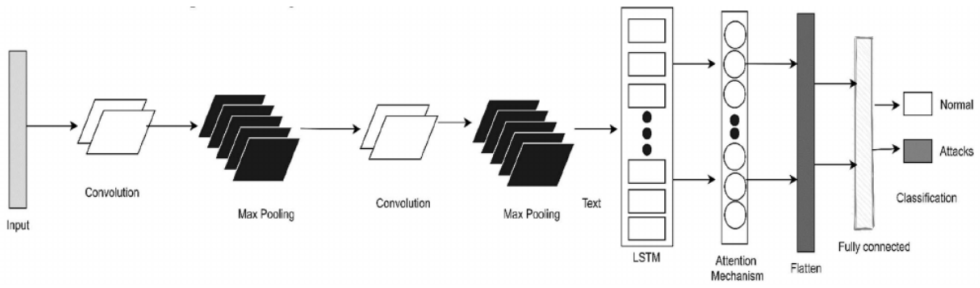
Sau thành công của AlexNet, có rất nhiều nghiên cứu về việc ứng dụng CNN trong bài toán liên quan đến xử lý âm thanh bằng Spectrogram đã được tiến hành và nổi bật nhất phải kể đến CRNN (*Convolutional Recurrent Neural Network*). Về cơ bản, mô hình CRNN chỉ đơn giản thêm một lớp RNN (*Recurrent Neural Network*) vào sau các lớp CNN để có thể xử lý được chuỗi thời gian, quản lý thứ tự của các token - điều vốn rất quan trọng với âm thanh nhưng các bài toán xử lý hình ảnh thuần tuý thường xem nhẹ. RNN được sử dụng nhiều nhất là biến thể LSTM (*Long Short Term Memory*). Vì có LSTM nên mô hình này có thể hiểu được thứ tự và ngữ cảnh của đoạn âm thanh, từ đó có thể tăng được hiệu suất. Tuy nhiên, LSTM cũng chỉ có bộ nhớ giới hạn nên mô hình cơ bản vẫn không nắm được toàn bộ ngữ cảnh của đoạn âm thanh.



Hình 2: Ví dụ minh hoạ cho mô hình CRNN. [4]

Sau cùng, cơ chế Attention đã được thêm vào mô hình CRNN để tạo thành mô hình Hybrid LST-CNN Attention. Cơ chế Attention giúp mô hình có thể tập trung vào những phần quan trọng của

đoạn âm thanh, bỏ qua những phần không cần thiết. Nhờ vậy mà mô hình có thể nắm bắt được ngữ cảnh toàn diện hơn, từ đó cải thiện hiệu suất nhận diện.



Hình 3: Ví dụ minh hoạ cho mô hình LSTM-CNN Attention. [5]

2.3 Mô hình Vision Transformer [6]

Áp dụng Transformer (vốn được xài cho xử lý ngôn ngữ) trực tiếp lên hình ảnh với ít sự thay đổi nhất có thể. Xử lý hình ảnh thành chuỗi các patches, từ đó biến một hình ảnh thành chuỗi các visual words

Quá trình xử lý:

- Nhận vào một hình ảnh, cắt nó thành tập hợp các ô có kích thước cố định 16x16 pixels.
- Duỗi nó thành vector 1D (16x16x3 (kênh màu) = 768).
- Vector đó được nhân với một ma trận trọng số để chuyển thành các dense vector có kích thước cố định (768) (gọi là các patch embeddings). Những embeddings này biểu diễn cho các visual words.
- Mượn ý tưởng từ BERT language model, tác giả không nghĩ việc lấy trung bình các vector output là một cách tốt để nhận về phân loại cho cả image. Thay vào đó, họ xài một vector CLS để prepend vào phần đầu của patch sequence.
- Sau đó, cộng dãy các vector (gồm cả CLS) với vector positional để nhận được đầu vào cuối cùng cho lớp encoder. Vector CLS tương tác với toàn bộ vector còn lại theo cơ chế self-attention của transformer. Xài đầu ra tương ứng của token này (qua lớp encoder) để đưa qua lớp phân loại cuối cùng.

Với cấu trúc như vậy, ViT giải quyết được các vấn đề của CNN gồm:

- Inductive Bias: ViT model xử lý tất cả các patch như các thực thể độc lập nhau, không có mối liên hệ đến các patch khác từ đó giảm được inductive bias so với CNN.
- Receptive field: CNN chỉ xử lý được cục bộ vì các lớp chỉ nhìn thấy một phần nhỏ cục bộ của tấm ảnh, còn ViT xử lý toàn cục vì mỗi patch được liên kết lại với nhau thông qua các lớp đầu.
- Data Hunger: CNN chỉ hoạt động tốt với tập dữ liệu nhỏ vì cách học của nó đã được xây dựng trước, còn cách học của ViT sẽ được xây dựng khi nó phân tích dữ liệu nên nó sẽ học được trên tập dữ liệu lớn hơn.

3 Mô hình đề xuất của tác giả

Mô hình được đề xuất kế thừa kiến trúc *Vision Transformer (ViT)* đã được pretrain trên ImageNet, sau đó được điều chỉnh để phù hợp với bài toán phân loại âm thanh. Ý tưởng chính là chuyển đổi tín hiệu âm thanh từ miền thời gian sang miền tần số để tạo ra biểu diễn dạng ảnh (spectrogram), từ đó tận dụng khả năng học đặc trưng mạnh mẽ của Transformer.

3.1 Kiến trúc tổng thể

Audio Spectrogram Transformer (AST) được xây dựng theo kiến trúc **pure Transformer**, hoàn toàn không sử dụng bất kỳ lớp Convolutional Neural Network (CNN) nào. Mô hình khai thác cơ chế *multi-head self-attention* để học các quan hệ phụ thuộc dài hạn trong spectrogram.

3.2 Biểu diễn đầu vào

Đầu vào của mô hình là ảnh **Log-Mel Spectrogram** được trích xuất từ tín hiệu waveform. Spectrogram được tính theo quy trình:

- Windowing: Hamming window 25 ms,
- Hop size: 10 ms,
- Số lượng filterbank Mel: 128.

Với đoạn âm thanh dài t giây, spectrogram thu được có kích thước:

$$128 \times 100t.$$

Spectrogram sau đó được chuẩn hóa và điều chỉnh kích thước để phù hợp với đầu vào ViT.

3.3 Patching

Spectrogram được chia thành các patch kích thước 16×16. AST sử dụng **stride = 10**, dẫn đến hiện tượng overlap 6 điểm theo cả trục tần số và trục thời gian.
Số lượng patch được tính bởi:

$$N = 12 \times \left\lceil \frac{100t - 16}{10} \right\rceil.$$

Trong đó 12 là số patch theo trục tần số. Mỗi patch được flatten để tạo thành vector 1 chiều.

3.4 Patch Embedding

Mỗi patch sau khi flatten được đưa qua một lớp Linear Projection để thu được embedding vector kích thước 768 (embedding dimension của ViT-Base).
Vì ViT gốc được pretrain trên ảnh RGB (3-channel), còn spectrogram chỉ có 1 channel, trọng số đầu vào được khởi tạo bằng cách **average-weighting** từ 3 channel của ViT sang 1 channel.

3.5 Positional Embedding

Để bảo toàn thông tin vị trí trên không gian thời gian–tần số, mỗi patch embedding được cộng thêm một vector positional embedding (learnable).
Do ViT gốc hoạt động trên ảnh vuông (ví dụ 384×384 với positional grid 24×24), còn spectrogram lại có dạng hình chữ nhật, nhóm tác giả đề xuất phương pháp **"crop-and-bilinear-interpolate"** đối với positional embedding:

- Crop số hàng từ 24 xuống 12 để phù hợp chiều frequency,
- Bilinear interpolate số cột từ 24 lên chiều thời gian tương ứng (ví dụ 100).

Cách này giúp positional embedding trở nên tương thích hoàn toàn với kích thước spectrogram.

3.6 Transformer Encoder

Chuỗi token đầu vào có dạng:

$$[\text{CLS}], x_1, x_2, \ldots, x_N.$$

Bộ *Transformer Encoder* trong AST được cấu hình như sau:

- 12 Transformer layers,
- 12 attention heads,

- Hidden size: 768.

Tất cả trọng số đều được khởi tạo từ ViT pretrained trên ImageNet để tận dụng khả năng nhận diện cấu trúc hình học đã học trước.

3.7 Classification Head

Sau Transformer Encoder, embedding của token `[CLS]` được dùng làm global representation của toàn bộ tín hiệu. Token này được truyền qua một lớp Linear mới (thay thế classification head của ViT) và một hàm kích hoạt Sigmoid để thực hiện phân loại đa nhãn.
Kết quả cuối cùng là vector xác suất tương ứng với từng loại âm thanh như tiếng chó, mèo, gà, chim, chuột, ...

4 Cài đặt thực nghiệm của tác giả

5 Cài đặt thực nghiệm của nhóm

6 Tài liệu tham khảo

[1] Jonathan Hui. *Speech Recognition — GMM, HMM*. Medium. Truy cập: 2025-11-29. Sept. 2019. URL: https://jonathan-hui.medium.com/speech-recognition-gmm-hmm-8bb5eff8b196.

[2] Sid Talha, Anthony Fleury, and Sebastien Ambellouis. "Human Action Recognition from Body-Part Directional Velocity Using Hidden Markov Models". In: *2017 16th IEEE International Conference on Machine Learning and Applications (ICMLA)*. IEEE, 2017, pp. 1123–1128. DOI: 10.1109/ICMLA.2017.00185. URL: https://ieeexplore.ieee.org/document/8260778.

[3] Y. LeCun et al. "Backpropagation Applied to Handwritten Zip Code Recognition". In: *Neural Computation* 1.4 (1989), pp. 541–551. DOI: 10.1162/neco.1989.1.4.541.

[4] Ali Hamid Meftah et al. "Speaker Identification in Different Emotional States in Arabic and English". In: *IEEE Access* 8 (2020), pp. 60070–60083. DOI: 10.1109/ACCESS.2020.2983029.

[5] Pendukeni Phalaagae et al. "A Hybrid CNN-LSTM Model With Attention Mechanism for Improved Intrusion Detection in Wireless IoT Sensor Networks". In: *IEEE Access* 13 (2025), pp. 57322–57341. DOI: 10.1109/ACCESS.2025.3555861.

[6] Alexey Dosovitskiy et al. "An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale". In: *International Conference on Learning Representations*. 2021. URL: https://openreview.net/forum?id=YicbFdNTTy.