

ĐẠI HỌC QUỐC GIA TPHCM

TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN

KHOA CÔNG NGHỆ THÔNG TIN

Báo cáo tổng kết

Đề tài: Nghiên cứu, phân tích và chạy thử nghiệm mô hình AST trong bài báo "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 12 tháng 12 năm 2025



Mục lục

1 Giới thiệu	1
2 Các công trình liên quan	2
2.1 Phương pháp thống kê và mô hình GMM-HMM	2
2.2 Mạng Nơ-ron Tích chập (CNN) và Kiến trúc Lai ghép (Hybrid Models)	2
2.3 Kiến trúc Transformer và Vision Transformer (ViT)	3
2.4 Audio Spectrogram Transformer (AST)	4
3 Cài đặt mô hình	5
4 Mô tả tập dữ liệu	7
5 Cài đặt thử nghiệm của tác giả [6]	8
5.1 Môi trường tối ưu hoá và huấn luyện	8
5.2 Xử lý dữ liệu và tăng cường (Augmentation)	8
5.3 Cấu hình chi tiết cho từng bộ dữ liệu	8
6 Thủ nghiệm của nhóm	10
6.1 Môi trường thực nghiệm	10
6.2 Tiền xử lý tín hiệu âm thanh	10
6.3 Cấu hình mô hình Audio Spectrogram Transformer	11
6.3.1 Cấu hình AST-S (AST-ImageNet)	11
6.3.2 Cấu hình AST-P (AST-ImageNet, AudioSet)	11
6.4 Chiến lược huấn luyện và đánh giá	12
7 Đánh giá và phân tích kết quả	13
7.1 Kết quả định lượng trên UrbanSound8K	13
7.2 Phân tích confusion matrix	13
7.3 So sánh AST-S và AST-P	15
8 So sánh với bài báo gốc và thảo luận	16
8.1 So sánh định lượng với bài báo gốc	16

8.2	So sánh định lượng với mô hình CNN trên UrbanSound8K	16
8.3	Điểm mạnh và điểm yếu của mô hình thực nghiệm	17
8.4	Đề xuất cải tiến và hướng phát triển	18
9	Mô tả ứng dụng	19
10	Kết luận	20
11	Tài liệu tham khảo	21

Danh sách bảng

1	Bảng tham số thực nghiệm cho các bộ dữ liệu AudioSet, SpeechCommands và ESC-50	9
2	Kết quả 10-fold cross-validation trên UrbanSound8K cho hai cấu hình AST.	13
3	Confusion matrix tổng hợp của cấu hình AST-P trên UrbanSound8K.	14
4	Confusion matrix tổng hợp của cấu hình AST-S trên UrbanSound8K.	14

Danh sách hình vẽ

1 Giới thiệu

Lĩnh vực phân loại âm thanh (Audio Classification) đóng vai trò then chốt trong sự phát triển của các hệ thống trí tuệ nhân tạo hiện đại. Trong nhiều thập kỷ, các phương pháp tiếp cận từ mô hình thống kê GMM-HMM đến mạng nơ-ron tích chập (CNN) và các biến thể lai ghép (CNN-Attention Hybrid [1]) đã đặt nền móng vững chắc cho việc trích xuất đặc trưng từ tín hiệu âm thanh. Tuy nhiên, các kiến trúc dựa trên tích chập vẫn đối mặt với thách thức cố hữu trong việc nắm bắt các mối quan hệ ngữ cảnh toàn cục do giới hạn về vùng tiếp nhận cục bộ (local receptive field), hạn chế khả năng biểu diễn các chuỗi tín hiệu phức tạp.

Sự xuất hiện của kiến trúc Transformer, đặc biệt là thành công của mô hình Vision Transformer (ViT) [2] trong thị giác máy tính, đã mở ra một hướng đi đột phá: xử lý dữ liệu không gian dưới dạng chuỗi các mảnh ghép (patches). Kế thừa tư duy này, mô hình **Audio Spectrogram Transformer (AST)** [3] đã được đề xuất như một giải pháp tiên phong loại bỏ hoàn toàn sự phụ thuộc vào tích chập (convolution-free). Bằng cơ chế Tự chú ý (Self-Attention), AST không chỉ giải quyết bài toán mô hình hóa sự phụ thuộc thời gian - tần số ở phạm vi toàn cục mà còn tận dụng hiệu quả tri thức chuyển giao (Transfer Learning) từ các bộ dữ liệu quy mô lớn như ImageNet và AudioSet để đạt hiệu suất vượt trội.

Trong phạm vi báo cáo này, chúng tôi tập trung nghiên cứu cơ chế hoạt động của kiến trúc AST và thực hiện quy trình thực nghiệm tinh chỉnh (fine-tuning) mô hình AST – vốn được huấn luyện trước trên AudioSet – để giải quyết bài toán phân loại âm thanh môi trường trên tập dữ liệu UrbanSound8K. Nghiên cứu nhằm mục đích kiểm chứng khả năng tổng quát hóa và hiệu quả của phương pháp học chuyển giao khi áp dụng các mô hình Transformer tiên tiến vào các bài toán cụ thể với tài nguyên dữ liệu giới hạn.

2 Các công trình liên quan

Lịch sử phát triển của bài toán phân loại âm thanh và nhận dạng sự kiện âm thanh (Audio Event Detection - AED) là một quá trình tiến hóa liên tục nhằm tìm kiếm các biểu diễn đặc trưng (feature representations) hiệu quả hơn. Quá trình này có thể được chia thành ba giai đoạn chính: từ các mô hình thống kê dựa trên đặc trưng thủ công, đến các mạng nơ-ron tích chập (CNN) và các biến thể lai ghép, và gần đây nhất là sự trỗi dậy của kiến trúc Transformer thuần túy.

2.1 Phương pháp thống kê và mô hình GMM-HMM

Trong giai đoạn đầu, các hệ thống nhận dạng âm thanh chủ yếu dựa vào các đặc trưng được thiết kế thủ công (hand-crafted features) như Mel-Frequency Cepstral Coefficients (MFCCs) hay Filterbanks. Mô hình tiêu biểu nhất trong giai đoạn này là sự kết hợp giữa **Mô hình Hỗn hợp Gaussian (Gaussian Mixture Models - GMM)** và **Mô hình Markov Ẩn (Hidden Markov Models - HMM)** [4].

Về mặt lý thuyết, GMM được sử dụng để mô hình hóa phân phối xác suất của các đặc trưng quang phổ tại mỗi khung thời gian (acoustic modeling), trong khi HMM chịu trách nhiệm mô hình hóa sự phụ thuộc và chuyển đổi trạng thái theo trình tự thời gian (temporal modeling). Mặc dù GMM-HMM đã đặt nền móng vững chắc cho ngành xử lý tiếng nói, chúng tồn tại những hạn chế cố hữu:

- **Phụ thuộc vào trích xuất đặc trưng:** Hiệu suất mô hình bị giới hạn bởi chất lượng của các đặc trưng thủ công, vốn không thể nắm bắt hết sự phức tạp của tín hiệu âm thanh thực tế.
- **Giả định độc lập:** HMM dựa trên giả định Markov, cho rằng trạng thái hiện tại chỉ phụ thuộc vào trạng thái liền trước, do đó gặp khó khăn trong việc nắm bắt các phụ thuộc dài hạn (long-term dependencies) trong chuỗi tín hiệu.

2.2 Mạng Nơ-ron Tích chập (CNN) và Kiến trúc Lai ghép (Hybrid Models)

Sự ra đời của AlexNet [5] vào năm 2012 đã đánh dấu sự chuyển dịch sang kỷ nguyên học sâu (Deep Learning). Các nhà nghiên cứu bắt đầu tiếp cận âm thanh dưới dạng hình ảnh thông qua biểu đồ

phổ (Spectrogram), cho phép áp dụng các kiến trúc Mạng nơ-ron tích chập (CNN) mạnh mẽ như VGG hay ResNet để tự động học các đặc trưng từ dữ liệu thay vì thiết kế thủ công.

CNN sở hữu thiên kiến quy nạp (inductive bias) mạnh mẽ, bao gồm tính cục bộ (locality) và tính bất biến tịnh tiến (translation equivariance), giúp chúng rất hiệu quả trong việc phát hiện các mẫu hình tần số cục bộ. Tuy nhiên, CNN thuần túy gặp hạn chế trong việc mô hình hóa chuỗi thời gian toàn cục do giới hạn của vùng tiếp nhận (receptive field). Để khắc phục, các kiến trúc Lai ghép (Hybrid Models) đã được đề xuất:

- **CRNN (Convolutional Recurrent Neural Networks):** Kết hợp CNN (để trích xuất đặc trưng không gian) với RNN hoặc LSTM (để mô hình hóa chuỗi thời gian).
- **CNN-Attention:** Thay thế hoặc bổ sung cơ chế RNN bằng cơ chế Chú ý (Attention Mechanism) để cho phép mô hình tập trung vào các đoạn âm thanh quan trọng và tổng hợp thông tin tốt hơn.

Mặc dù các mô hình lai ghép này đã đạt được những kết quả vượt trội, chúng vẫn chịu ràng buộc bởi cấu trúc tích chập: khả năng nắm bắt thông tin toàn cục chỉ đạt được ở các lớp rất sâu hoặc thông qua các phép toán gộp (pooling) làm mất mát thông tin chi tiết.

2.3 Kiến trúc Transformer và Vision Transformer (ViT)

Kiến trúc Transformer, ban đầu được giới thiệu cho các tác vụ xử lý ngôn ngữ tự nhiên, dựa hoàn toàn vào cơ chế Tự chú ý (Self-Attention) để mô hình hóa các mối quan hệ toàn cục trong chuỗi dữ liệu. Dosovitskiy et al. (2020) đã đề xuất Vision Transformer (ViT) [2], minh chứng rằng một kiến trúc Transformer thuần túy có thể đạt hiệu suất vượt trội trong thị giác máy tính bằng cách chia hình ảnh thành chuỗi các mảnh ghép (patches) cố định (ví dụ: 16×16) và xử lý chúng tương tự như các từ trong câu.

Khác biệt cốt lõi của ViT so với CNN nằm ở Vùng tiếp nhận toàn cục (Global Receptive Field). Ngay từ lớp đầu tiên, cơ chế Self-Attention cho phép mỗi mảnh ghép tham chiếu thông tin từ tất cả các mảnh ghép khác, giúp mô hình nắm bắt cấu trúc tổng thể của dữ liệu ngay lập tức mà không cần trải qua nhiều lớp tích chập phân cấp. Hơn nữa, ViT có thiên kiến quy nạp yếu hơn, cho phép mô hình linh hoạt hơn trong việc học các mối quan hệ phức tạp từ các tập dữ liệu quy mô lớn.

2.4 Audio Spectrogram Transformer (AST)

Kế thừa thành tựu của ViT, Gong et al. (2021) đã đề xuất mô hình Audio Spectrogram Transformer (AST) [3]. Đây là kiến trúc đầu tiên áp dụng cơ chế Transformer không tích chập (convolution-free) cho bài toán phân loại âm thanh.

AST xử lý biểu đồ phổ âm thanh đầu vào (kích thước $128 \times 100t$) bằng cách chia nhỏ thành chuỗi các mảnh 16×16 có sự chồng lấn (overlap). Các mảnh này được chiếu tuyến tính thành các vector nhúng (embeddings), cộng với thông tin vị trí (positional embedding) và đưa qua bộ mã hóa Transformer chuẩn. Điểm đột phá của AST nằm ở khả năng tận dụng tri thức chuyển giao (Transfer Learning) từ miền hình ảnh (ImageNet) sang miền âm thanh. Các tác giả đã đề xuất các kỹ thuật thích ứng hiệu quả như trung bình hóa trọng số kênh và nội suy vị trí để giải quyết sự khác biệt về chiều dữ liệu giữa hai miền.

Các thực nghiệm trên AudioSet và ESC-50 cho thấy AST không chỉ vượt qua các mô hình CNN-Attention lai ghép tốt nhất (SOTA) mà còn chứng minh khả năng hội tụ nhanh và hiệu quả dữ liệu vượt trội nhờ vào cơ chế chú ý toàn cục và chiến lược khởi tạo trọng số thông minh.

3 Cài đặt mô hình

Mô hình Audio Spectrogram Transformer (AST) được xây dựng dựa trên nền tảng kiến trúc DeiT (Data-efficient Image Transformers) để tận dụng các trọng số pre-trained hiệu quả từ ImageNet. Chi tiết cài đặt như sau: [3]

Đầu vào và tiền xử lý:

- **Đặc trưng:** Tín hiệu âm thanh (waveform) được chuyển đổi thành log-Mel spectrogram 128 chiều.
- **Tham số STFT:** Cửa sổ Hamming 25 ms, bước trượt 10 ms; chuẩn hoá đặc trưng trước khi đưa vào mạng.
- **Kích thước:** Với đoạn âm thanh dài t giây, số khung thời gian xấp xỉ $100t$, tạo ra spectrogram kích thước $128 \times 100t$.

Phân mảnh (Patching):

- Spectrogram được chia thành các patch kích thước 16×16 .
- Áp dụng cơ chế chồng lấn (overlap) với stride = 10 trên cả hai trực thời gian và tần số (tương đương vùng chồng lấn 6 đơn vị).
- Số patch theo trực tần số cố định là 12; theo trực thời gian là $\left\lfloor \frac{100t-16}{10} \right\rfloor + 1$.
- Tổng số patch:

$$N = 12 \times \left(\left\lfloor \frac{100t-16}{10} \right\rfloor + 1 \right).$$

Kiến trúc Transformer và Token đặc biệt: Mô hình sử dụng backbone là ViT (biến thể DeiT-Base) với cấu hình: embedding $d = 768$, 12 lớp encoder, 12 đầu attention.

- **Token đặc biệt:** Do sử dụng kiến trúc DeiT, chuỗi đầu vào được bổ sung hai token đặc biệt ở đầu: Token phân loại ([CLS]) và Token chưng cất ([DIST]).
- **Chuỗi đưa vào Encoder:**

$$Input = [\text{CLS}, \text{DIST}, \text{Patch}_1, \text{Patch}_2, \dots, \text{Patch}_N] + \text{PosEmbed}$$

- **Mã hóa vị trí:** Sử dụng positional embedding có thể học được. Trong code gốc, vector này được nội suy (interpolate) động để phù hợp với chiều dài thay đổi của âm thanh đầu vào.

Đầu ra và huấn luyện:

- Lấy vector của [CLS] làm biểu diễn và đưa qua đầu phân loại tuyến tính.
- Bài toán gốc (AudioSet): đa nhãn, sử dụng sigmoid ở đầu ra và Binary Cross-Entropy (BCE) làm hàm mất mát.
- Pretrain: Khởi tạo từ ViT đã được pretrain trên ImageNet hoặc Imagenet + AudioSet; sau đó fine-tune trên tập target.

Đầu ra và huấn luyện:

- **Biểu diễn đầu ra (Pooling):** Đặc trưng cuối cùng của đoạn âm thanh không chỉ dùng token [CLS] mà là trung bình cộng của token [CLS] và [DIST]:

$$h_{final} = \frac{h_{[CLS]} + h_{[DIST]}}{2}$$

- **Lớp phân loại:** Vector h_{final} được chuẩn hóa (LayerNorm) trước khi đưa qua lớp tuyến tính (Linear Head) để dự đoán nhãn.
- **Hàm mất mát:** Sử dụng Binary Cross-Entropy (BCE) cho bài toán đa nhãn (AudioSet).

4 Mô tả tập dữ liệu

Trong nghiên cứu này, nhóm sử dụng tập dữ liệu UrbanSound8K, một bộ dữ liệu âm thanh môi trường đô thị được xây dựng để phục vụ các bài toán phân loại các đoạn âm thanh ngắn trong đời sống hằng ngày. UrbanSound8K bao gồm 8.732 đoạn âm thanh (audio clips) đã được gán nhãn, với thời lượng không quá 4 giây cho mỗi đoạn. Các đoạn âm thanh này được trích xuất từ các bản ghi dài hơn và được chú thích cẩn thận.

Toàn bộ dữ liệu được chia thành 10 fold (fold1 đến fold10), tuân theo chiến lược 10-fold cross-validation được đề xuất sẵn bởi tác giả bộ dữ liệu. Cách chia này được thiết kế nhằm đảm bảo người dùng có thể đánh giá mô hình một cách công bằng và nhất quán, đồng thời hạn chế hiện tượng data leakage (rò rỉ dữ liệu giữa tập huấn luyện và tập kiểm thử). Trong mỗi fold, các đoạn âm thanh thuộc nhiều lớp khác nhau được phân bố một cách tương đối cân bằng.

UrbanSound8K được gán nhãn theo 10 loại âm thanh (10 classes) thường gặp trong môi trường đô thị, ví dụ như tiếng ô tô (car horn), tiếng chó sủa (dog bark), tiếng khoan (drilling), tiếng còi cứu hỏa (siren), tiếng máy nổ (engine idling), v.v. Mỗi bản ghi được mô tả bởi các thông tin meta kèm theo như: tên file, fold, mã lớp (class ID) và tên lớp (class name). Các thông tin này được lưu trong file metadata UrbanSound8K.csv, giúp việc truy vấn, lọc và xây dựng tập train/test trở nên thuận tiện.

Về mặt tổ chức thư mục, UrbanSound8K được cấu trúc như sau:

- Thư mục audio/ chứa 10 thư mục con: fold1/, fold2/, ..., fold10/.
- Mỗi thư mục foldX/ chứa các file âm thanh định dạng .wav tương ứng với fold đó.
- Thư mục metadata/ chứa file UrbanSound8K.csv, trong đó mỗi dòng tương ứng với một đoạn âm thanh kèm nhãn và thông tin mô tả.

Tập dữ liệu UrbanSound8K có dung lượng khoảng 6 GB, bao gồm cả các file âm thanh và metadata. Với quy mô vừa phải nhưng đa dạng về loại âm thanh và điều kiện thu, UrbanSound8K là lựa chọn phù hợp cho các thí nghiệm *fine-tuning* mô hình Audio Spectrogram Transformer (AST) trong bối cảnh phân loại âm thanh môi trường. Đồng thời, đây cũng là một bộ dữ liệu phổ biến trong cộng đồng, cho phép so sánh kết quả với nhiều mô hình khác (CNN, CRNN, hay các mô hình transformer khác) đã được công bố trước đó.

5 Cài đặt thử nghiệm của tác giả [6]

Để đánh giá hiệu quả của mô hình Audio Spectrogram Transformer (AST), tác giả thiết lập quy trình huấn luyện và kiểm thử với các cấu hình chi tiết như sau.

5.1 Môi trường tối ưu hóa và huấn luyện

Mô hình được huấn luyện theo phương pháp học giám sát (supervised learning) sử dụng thuật toán tối ưu hóa Adam. Các tham số của bộ tối ưu hóa được thiết lập như sau:

- Trọng số suy giảm (Weight decay): 5×10^{-7} .
- Các hệ số Beta: $\beta_1 = 0.95$, $\beta_2 = 0.999$.

Chiến lược khởi tạo trọng số (Initialization) tận dụng mô hình đã được pre-trained trên ImageNet (đối với tất cả các tập dữ liệu) để tăng cường khả năng trích xuất đặc trưng và hội tụ nhanh hơn.

5.2 Xử lý dữ liệu và tăng cường (Augmentation)

Dữ liệu đầu vào là Log-Mel Spectrogram được chuẩn hóa (Normalization) bằng cách trừ đi giá trị trung bình (Mean) và chia cho độ lệch chuẩn (Std) của toàn bộ tập dữ liệu.

Để giảm thiểu hiện tượng quá khớp (overfitting), tác giả áp dụng các kỹ thuật tăng cường dữ liệu mạnh mẽ ngay trong quá trình tải dữ liệu (online augmentation):

- **SpecAugment:** Áp dụng mặt nạ tần số (Frequency Masking) và mặt nạ thời gian (Time Masking) với kích thước tối đa tùy thuộc vào từng tập dữ liệu.
- **Mixup:** Trộn lẫn hai mẫu dữ liệu ngẫu nhiên với hệ số λ được lấy mẫu từ phân phối Beta hoặc Uniform, giúp mô hình học được các đặc trưng lai giữa các lớp.
- **Noise Augmentation:** Thêm nhiễu ngẫu nhiên vào spectrogram (chỉ áp dụng cho tập Speech-Commands).

5.3 Cấu hình chi tiết cho từng bộ dữ liệu

Các siêu tham số (Hyperparameters) cụ thể cho từng bài toán thực nghiệm được thiết lập dựa trên các kịch bản chuẩn (`run.sh`, `run_sc.sh`, `run_esc.sh`) như bảng dưới đây:

Tham số	AudioSet (Full)	SpeechCommands (v2)	ESC-50
Kích thước đầu vào	1024 frames (~10s)	128 frames (~1s)	512 frames (~5s)
Batch Size	12	128	48
Learning Rate (LR)	1×10^{-5}	2.5×10^{-4}	1×10^{-5}
Số Epochs	5	30	25
LR Scheduler	MultiStepLR (Giảm 0.5 mỗi epoch từ epoch 2)	MultiStepLR (Giảm 0.85 mỗi epoch từ epoch 5)	MultiStepLR (Giảm 0.85 mỗi epoch từ epoch 5)
Warmup	Có	Không	Không
SpecAugment	Freq Mask: 48, Time Mask: 192	Freq Mask: 48, Time Mask: 48	Freq Mask: 24, Time Mask: 96
Mixup (α)	0.5	0.6	0
Hàm mất mát	BCEWithLogitsLoss	BCEWithLogitsLoss	CrossEntropyLoss
Độ đo chính	mAP	Accuracy	Accuracy

Bảng 1: Bảng tham số thực nghiệm cho các bộ dữ liệu AudioSet, SpeechCommands và ESC-50

Lưu ý đặc biệt cho AudioSet: Đối với tập dữ liệu AudioSet (cấu hình Full), tác giả sử dụng kỹ thuật **Weight Averaging** (trung bình trọng số mô hình) từ epoch 1 đến epoch 5 để tạo ra mô hình cuối cùng ổn định hơn.

Thông số chuẩn hóa: Các giá trị trung bình, độ lệch chuẩn được sử dụng cho chuẩn hóa đầu vào:

- AudioSet: Mean - 4.268, Std 4.569.
- SpeechCommands: Mean - 6.846, Std 5.565.
- ESC-50: Mean - 6.627, Std 5.358.

6 Thủ nghiệm của nhóm

6.1 Môi trường thực nghiệm

Các thí nghiệm được triển khai trong môi trường notebook trực tuyến có hỗ trợ GPU với nền tảng CUDA (Kaggle Notebook). Mô hình được hiện thực bằng thư viện PyTorch (phiên bản 2.x), kết hợp cùng các thư viện torchaudio, timm, librosa và scikit-learn để xử lý tín hiệu và xây dựng pipeline huấn luyện. Các phép tính trên GPU được tăng tốc bằng cơ chế mixed precision training thông qua mô-đun torch.amp.autocast và GradScaler, giúp giảm dung lượng bộ nhớ và rút ngắn thời gian huấn luyện.

Dữ liệu UrbanSound8K được mount sẵn tại thư mục /kaggle/input và toàn bộ mã nguồn, checkpoint của mô hình được lưu tại /kaggle/working.

6.2 Tiền xử lý tín hiệu âm thanh

Mỗi đoạn âm thanh trong UrbanSound8K đều được chuyển về dạng log Mel filterbank (fbank) trước khi đưa vào mô hình Audio Spectrogram Transformer (AST). Quy trình tiền xử lý gồm các bước:

- Tải tín hiệu âm thanh bằng torchaudio với dạng tensor (channels, time).
- Nếu tín hiệu có nhiều kênh (stereo), chuyển về mono bằng cách lấy trung bình trên trực kinh.
- Chuẩn hóa tần số lấy mẫu về 16 kHz bằng phép nội suy lại (resampling).
- Cắt hoặc pad tín hiệu về đúng 10 giây, tương ứng với một số mẫu cố định.
- Trích xuất đặc trưng log Mel filterbank với 128 băng tần Mel, window 25 ms, bước trượt 10 ms, thu được khoảng 1024 khung thời gian.
- Nếu số khung thời gian nhỏ hơn 1024 thì pad thêm khung 0; nếu lớn hơn thì cắt bớt cho vừa 1024 khung.
- Chuẩn hóa đặc trưng theo công thức $(x - \mu)/(2\sigma)$ với $\mu = -4.2677$ và $\sigma = 4.5690$, nhằm đưa về miền giá trị phù hợp với mô hình AST gốc.

Kết quả mỗi đoạn âm thanh là một tensor kích thước $(T, F) = (1024, 128)$, sau đó được đưa vào mô hình dưới dạng batch.

6.3 Cấu hình mô hình Audio Spectrogram Transformer

Trong tất cả các thí nghiệm, nhóm sử dụng cùng một kiến trúc Audio Spectrogram Transformer (AST) với cấu hình base384, stride thời gian và tần số đều bằng 10. Với đầu vào 128 băng tần Mel và 1024 khung thời gian, mô hình tạo ra 1212 patch spectrogram. Phần đầu ra của backbone là một vector đặc trưng có chiều 768, sau đó được đưa qua một lớp phân loại tuyến tính (mlp head) ánh xạ về 10 lớp tương ứng với 10 loại âm thanh của UrbanSound8K.

Trên cơ sở kiến trúc chung này, nhóm xây dựng hai cấu hình tiền huấn luyện khác nhau:

6.3.1 Cấu hình AST-S (AST-ImageNet)

Trong kịch bản thứ nhất, mô hình AST được khởi tạo từ các trọng số đã được huấn luyện trước trên ImageNet thông qua tham số `imagenet_pretrain = True` và `audioset_pretrain = False` trong hàm khởi tạo `ASTModel`. Toàn bộ backbone vì vậy kế thừa khả năng trích xuất đặc trưng từ ảnh (được ánh xạ sang miền spectrogram), nhưng không sử dụng thêm bất kỳ checkpoint nào từ AudioSet.

Lớp phân loại gốc của AST (với 527 nút đầu ra) được thay thế bằng một lớp tuyến tính mới có 10 nút, khởi tạo ngẫu nhiên, để phù hợp với bài toán phân loại 10 lớp trên UrbanSound8K. Cấu hình này được ký hiệu là AST-S.

6.3.2 Cấu hình AST-P (AST-ImageNet, AudioSet)

Trong kịch bản thứ hai, mô hình AST được khởi tạo sao cho tương thích với checkpoint đã fine-tune trên AudioSet (`audioset_0.4593.pth`). Cụ thể, mô hình được tạo với `label_dim = 527`, `input_tdim = 1024`, `input_fdim = 128`, và cả hai tham số `imagenet_pretrain`, `audioset_pretrain` đều đặt `False` để nhóm chủ động nạp trọng số từ checkpoint AudioSet bên ngoài.

Sau khi tải file trọng số `audioset_0.4593.pth`, toàn bộ `state_dict` được nạp vào mô hình, giúp backbone kế thừa trực tiếp khả năng phân biệt âm thanh từ tập AudioSet. Tương tự như cấu hình trước, lớp phân loại 527 chiều được thay thế bằng một lớp tuyến tính 10 chiều, khởi tạo ngẫu nhiên cho UrbanSound8K. Cấu hình này được ký hiệu là AST-P.

Hai cấu hình trên sử dụng chung toàn bộ pipeline tiền xử lý, kiến trúc backbone, hàm mất mát, optimizer và chiến lược huấn luyện; điểm khác biệt duy nhất nằm ở nguồn trọng số tiền huấn luyện được sử dụng cho backbone.

6.4 Chiến lược huấn luyện và đánh giá

Để đánh giá mô hình một cách công bằng, nhóm áp dụng chiến lược 10-fold cross-validation theo đúng cách chia fold có sẵn trong UrbanSound8K:

- Ở mỗi vòng lặp, chọn một fold (từ 1 đến 10) làm tập kiểm thử.
- Chín fold còn lại được gộp lại tạo thành tập huấn luyện.
- Tuyệt đối không trộn mẫu giữa các fold, đảm bảo không xảy ra rò rỉ dữ liệu.

Với mỗi cấu hình mô hình (AST-S và AST-P), nhóm sử dụng cùng một bộ siêu tham số:

- Hàm mất mát: Cross-entropy cho bài toán phân loại đơn nhãn.
- Optimizer: AdamW với learning rate cố định là 1×10^{-5} .
- Kích thước batch: 8 (được lựa chọn để phù hợp với dung lượng bộ nhớ GPU).
- Số epoch: từ 3 đến 5 epoch cho mỗi fold, tùy theo giới hạn tính toán; trong cả hai cấu hình, mô hình hội tụ khá nhanh nhờ sử dụng trọng số tiền huấn luyện.
- Huấn luyện sử dụng mixed precision (autocast và GradScaler) để giảm chi phí tính toán.

Sau mỗi epoch, mô hình được đánh giá trên tập kiểm thử của fold tương ứng. Nhóm lưu lại checkpoint có độ chính xác cao nhất trên tập kiểm thử (best test accuracy) cho mỗi fold. Các checkpoint này được ghi vào thư mục `ast_us8k_checkpoints`.

Cuối cùng, để tổng hợp kết quả, nhóm tính toán độ chính xác trung bình và độ lệch chuẩn trên toàn bộ 10 fold, theo công thức:

$$\text{Mean Accuracy} = \frac{1}{10} \sum_{k=1}^{10} \text{Acc}_k, \quad \text{Std} = \sqrt{\frac{1}{10} \sum_{k=1}^{10} (\text{Acc}_k - \text{Mean Accuracy})^2}.$$

Việc sử dụng cùng một thiết lập huấn luyện và cùng chiến lược cross-validation cho cả AST-S và AST-P giúp việc so sánh trực tiếp tác động của tiền huấn luyện trên AudioSet đối với hiệu năng phân loại âm thanh môi trường trên UrbanSound8K.

7 Đánh giá và phân tích kết quả

7.1 Kết quả định lượng trên UrbanSound8K

Sau khi huấn luyện và lưu checkpoint cho từng fold, nhóm tiến hành đánh giá lại toàn bộ 10 mô hình trên tập kiểm thử tương ứng của mỗi fold. Quá trình đánh giá được thực hiện riêng cho hai cấu hình:

- AST-P: backbone khởi tạo từ checkpoint đã fine-tune trên AudioSet, sau đó thay đầu ra 10 lớp và fine-tune trên UrbanSound8K.
- AST-S: backbone khởi tạo từ mô hình tiền huấn luyện trên ImageNet, chỉ fine-tune trực tiếp trên UrbanSound8K, không sử dụng checkpoint AudioSet.

Bảng dưới đây tóm tắt kết quả trung bình trên 10 fold:

Cấu hình mô hình	Mean accuracy (%)	Std (%)	Micro accuracy (%)
AST-S	79.75	3.76	79.70
AST-P	86.31	4.20	86.27

Bảng 2: Kết quả 10-fold cross-validation trên UrbanSound8K cho hai cấu hình AST.

Đối với cấu hình AST-P, độ chính xác trên từng fold dao động từ khoảng 76.86% (fold khó nhất) đến 92.12% (fold tốt nhất), với giá trị trung bình 86.31% và độ lệch chuẩn 4.20%. Micro accuracy tính trực tiếp từ confusion matrix tổng hợp trên toàn bộ dữ liệu là 86.27%, gần trùng với giá trị trung bình theo fold, cho thấy việc chia fold và tổng hợp là nhất quán.

Trong khi đó, cấu hình AST-S đạt độ chính xác trung bình 79.75% với độ lệch chuẩn 3.76%. Các fold dao động trong khoảng từ 72.65% đến 84.19%, micro accuracy tính từ confusion matrix tổng là 79.70%. Như vậy, so với AST-S, mô hình AST-P cải thiện khoảng 6.5 điểm phần trăm về độ chính xác trung bình trên UrbanSound8K.

7.2 Phân tích confusion matrix

Để quan sát chi tiết hơn cách mô hình nhầm lẫn giữa các lớp, nhóm tổng hợp confusion matrix trên toàn bộ 10 fold cho từng cấu hình. Thứ tự các lớp trong bảng là: air_conditioner, car_horn, children_playing, dog_bark, drilling, engine_idling, gun_shot, jackhammer, siren, street_music.

Báo cáo tổng kết

Thật \ Dự đoán	air_cond.	car_horn	child_play	dog_bark	drilling	eng_idle	gun_shot	jackhammer	siren	street_music
air_cond.	744	0	26	29	86	46	0	9	11	49
car_horn	0	399	3	2	1	1	0	13	0	10
child_play	2	0	951	6	11	0	0	0	6	24
dog_bark	18	2	9	958	3	0	0	0	5	5
drilling	27	11	8	4	856	14	0	58	4	18
eng_idle	102	3	5	0	40	715	0	118	4	13
gun_shot	0	0	0	4	1	0	368	1	0	0
jackhammer	22	0	1	0	171	10	0	782	0	14
siren	12	11	5	43	1	0	0	1	849	7
street_music	9	3	52	3	8	4	0	5	5	911

Bảng 3: Confusion matrix tổng hợp của cấu hình AST-P trên UrbanSound8K.

Thật \ Dự đoán	air_cond.	car_horn	child_play	dog_bark	drilling	eng_idle	gun_shot	jackhammer	siren	street_music
air_cond.	556	2	33	97	77	78	0	33	38	86
car_horn	1	392	2	0	7	0	0	10	0	17
child_play	8	0	879	48	6	3	0	1	6	49
dog_bark	10	2	31	923	8	6	3	0	7	10
drilling	36	16	12	18	774	20	2	70	28	24
eng_idle	114	1	17	10	14	680	2	121	28	13
gun_shot	0	0	0	5	0	0	369	0	0	0
jackhammer	55	3	4	0	182	53	4	688	0	11
siren	9	3	25	20	32	4	0	0	811	25
street_music	20	13	42	13	6	7	0	5	7	887

Bảng 4: Confusion matrix tổng hợp của cấu hình AST-S trên UrbanSound8K.

Từ hai confusion matrix trong Bảng 3 và Bảng 4, nhóm rút ra một số nhận xét định tính về hành vi của mô hình.

Các lớp dễ phân biệt. Các lớp có tính chất âm thanh rõ ràng, mang tính sự kiện ngắn và đặc trưng như gun_shot, car_horn, dog_bark, children_playing thường đạt độ chính xác rất cao ở cả hai cấu hình. Ví dụ, lớp gun_shot hầu như luôn được dự đoán đúng, số lượng mẫu bị nhầm sang lớp khác là rất ít. Điều này phù hợp với trực giác khi đây là những âm thanh có biên dạng thời gian và phổ tần tương đối đặc trưng.

Các lớp khó, dễ gây nhầm lẫn. Những lớp mang tính nền tiếng ồn đô thị và tiếng máy móc như air_conditioner, engine_idling, drilling, jackhammer và siren có xu hướng bị nhầm lẫn lẫn nhau. Trong confusion matrix của cả hai cấu hình, các hàng tương ứng với các lớp này có số lượng mẫu bị dự đoán sang nhau khá lớn. Điều này phản ánh việc đặc trưng log-Mel của các loại tiếng ồn công nghiệp có phổ tần chồng lấn và hình thái thời gian tương đối giống nhau.

So sánh hai confusion matrix cho thấy khi sử dụng pretraining AudioSet, số lượng nhầm lẫn giữa một số cặp lớp giảm xuống, đặc biệt là đối với các lớp có tính chất âm thanh gần với các nhãn trong AudioSet. Điều này gợi ý rằng kiến thức học được từ AudioSet giúp backbone AST phân tách tốt hơn các kiểu tiếng ồn phức tạp, dù vẫn còn một mức độ nhầm lẫn nhất định giữa các lớp có bẩn

chất rất giống nhau.

7.3 So sánh AST-S và AST-P

Kết quả trong Bảng 2 cho thấy:

- Pretraining trên AudioSet mang lại mức cải thiện đáng kể về hiệu năng: từ khoảng 79.75% (AST-S) lên 86.31% (AST-P), tương đương tăng khoảng 6.5 điểm phần trăm trên UrbanSound8K.
- Độ lệch chuẩn giữa các fold của hai cấu hình đều ở mức khoảng 4%, cho thấy hiệu năng ổn định trên các fold khác nhau, mặc dù một số fold vẫn khó hơn do phân bố lớp và điều kiện thu âm.
- Khoảng cách giữa độ chính xác huấn luyện (gần 100% ở các epoch cuối) và độ chính xác kiểm thử (khoảng 80–86%) cho thấy mô hình vẫn có hiện tượng overfitting ở cả hai cấu hình. Tuy nhiên, với cùng một pipeline huấn luyện, việc sử dụng trọng số tiền huấn luyện từ AudioSet giúp cải thiện đáng kể khả năng tổng quát hóa trên tập kiểm thử.

Nhìn chung, kết quả thực nghiệm trên UrbanSound8K phù hợp với kết luận của tác giả AST: pretraining trên một tập dữ liệu âm thanh lớn như AudioSet mang lại lợi ích rõ rệt cho các bài toán phân loại âm thanh môi trường trên những tập dữ liệu quy mô vừa và nhỏ. Trong bối cảnh bài toán này, AST-P không chỉ đạt độ chính xác cao hơn mà còn giữ được độ ổn định tốt trên 10 fold, thể hiện vai trò quan trọng của pretraining đúng miền dữ liệu so với chỉ pretraining trên ImageNet.

8 So sánh với bài báo gốc và thảo luận

8.1 So sánh định lượng với bài báo gốc

Do trong nghiên cứu gốc, tác giả không thực nghiệm trên tập dữ liệu UrbanSound8K nên phép so sánh được thực hiện thông qua việc đổi chiêu **mức độ cải thiện hiệu năng** (performance gain) khi áp dụng pretraining trên AudioSet đối với các tập dữ liệu âm thanh môi trường tương đồng (ESC-50, SpeedCommands V2).

Kết quả thực nghiệm từ [Bảng 2](#) cho thấy sự chuyển dịch từ cấu hình khởi tạo ImageNet (AST-S) sang ImageNet kết hợp AudioSet (AST-P) giúp độ chính xác trên UrbanSound8K tăng từ 79.75% lên 86.31%, tương đương mức tăng 6.56%. Con số này hoàn toàn tương thích với xu hướng được báo cáo trong nghiên cứu gốc, trên tập dữ liệu ESC-50 (có đặc tính gần với UrbanSound8K), việc sử dụng AudioSet pretraining giúp tăng độ chính xác từ 88.7% lên 95.6% (mức tăng +6.9% [\[3\]](#)). Điều này khẳng định rằng AST không tự động học tốt các đặc trưng âm thanh chỉ từ dữ liệu hình ảnh (ImageNet), mà cần sự hỗ trợ mạnh mẽ từ dữ liệu miền âm thanh (AudioSet) để đạt hiệu năng tối ưu trên bài toán phân loại âm thanh.

8.2 So sánh định lượng với mô hình CNN trên UrbanSound8K

So sánh trực tiếp trên cùng bộ dữ liệu UrbanSound8K, nhóm đổi chiêu kết quả AST với mô hình CNN tiêu biểu trong bài báo của Salamon và Bello (2016) [\[7\]](#). Trong nghiên cứu này, tác giả đề xuất kiến trúc SB-CNN và phân tích vai trò của data augmentation trong việc khắc phục hạn chế về quy mô dữ liệu gán nhãn cho bài toán phân loại âm thanh môi trường.

Theo phần Results của [\[7\]](#), mô hình SB-CNN khi huấn luyện không sử dụng augmentation đạt độ chính xác trung bình khoảng 0.73 (73%), tương đương với các baseline cùng thời điểm như SKM (0.74) và PiczakCNN (0.73). Khi kết hợp các kỹ thuật augmentation (tập “All” gồm Time Stretching, Pitch Shifting, Dynamic Range Compression và Background Noise), độ chính xác của SB-CNN tăng lên khoảng 0.79 (79%), tương ứng với mức cải thiện xấp xỉ 6 điểm phần trăm. Kết quả này cho thấy trong thiết lập CNN thuần túy, augmentation đóng vai trò quan trọng trong việc giúp mô hình khai thác tốt hơn tập dữ liệu UrbanSound8K có quy mô tương đối nhỏ.

So với các mốc trên, kết quả của nhóm trong [Bảng 2](#) cho thấy AST-S đạt độ chính xác trung bình 79.75%, gần tương đương với SB-CNN khi sử dụng augmentation. Điều này cho thấy ngay cả khi

chỉ khởi tạo từ ImageNet, kiến trúc Transformer vẫn có thể đạt mức hiệu năng tương đương với CNN được hỗ trợ mạnh bởi augmentation. Trong khi đó, cấu hình AST-P đạt độ chính xác 86.31%, cao hơn SB-CNN (aug) khoảng 7.3 điểm phần trăm. Mức chênh lệch này phản ánh lợi thế rõ rệt của việc pretrain trên dữ liệu âm thanh quy mô lớn (AudioSet) so với cách tiếp cận CNN vốn chủ yếu dựa vào augmentation để bù đắp sự thiếu hụt dữ liệu.

Bài báo [7] sử dụng đầu vào là log-Mel spectrogram và huấn luyện SB-CNN trên các TF-patch cố định dài 3 giây, sau đó tổng hợp dự đoán từ nhiều patch ở pha suy luận. Cách tiếp cận này khai thác hiệu quả các đặc trưng cục bộ theo thời gian và tần số, tuy nhiên vẫn bị giới hạn bởi khả năng học biểu diễn từ tập dữ liệu UrbanSound8K tương đối nhỏ. Ngược lại, AST-P được hưởng lợi từ các biểu diễn đã học trước trên AudioSet, giúp mô hình phân tách tốt hơn các lớp có đặc tính nhiễu nền và phổ tần chồng lấn mạnh như air_conditioner, engine_idling, drilling và jackhammer, phù hợp với các quan sát từ confusion matrix trong [Bảng 3](#) và [Bảng 4](#).

Nói tóm lại, trong khi SB-CNN cho thấy việc kết hợp CNN với data augmentation có thể đạt kết quả tốt trên UrbanSound8K tại thời điểm nghiên cứu năm 2016, kết quả thực nghiệm của nhóm cho thấy hướng tiếp cận dựa trên Transformer kết hợp với pretrain trên dữ liệu âm thanh quy mô lớn mang lại mức cải thiện rõ rệt hơn, đặc biệt khi so sánh với các mô hình CNN phụ thuộc nhiều vào augmentation để nâng cao hiệu năng.

8.3 Điểm mạnh và điểm yếu của mô hình thực nghiệm

Dựa trên kết quả thực nghiệm và ma trận nhầm lẫn (confusion matrix), các ưu nhược điểm của mô hình được xác định như sau:

Điểm mạnh:

- Khả năng chuyển giao tri thức:** Mô hình chứng minh khả năng hội tụ nhanh và đạt độ chính xác cao chỉ với 3-5 epoch huấn luyện. Điều này cho thấy AST đã học được các biểu diễn đặc trưng rất mạnh từ quá trình pretraining, giúp giảm thiểu đáng kể chi phí tính toán cho các tác vụ phía sau.
- Độ ổn định cao:** Độ lệch chuẩn giữa các fold chỉ khoảng 4%, cho thấy mô hình không phụ thuộc quá nhiều vào cách chia dữ liệu và có khả năng tổng quát hóa tốt trên các phân phối mẫu khác nhau.

Điểm yếu:

- **Hiện tượng Overfitting:** Quan sát cho thấy độ chính xác trên tập huấn luyện tiêm cận 100% trong khi tập kiểm thử chỉ đạt khoảng 86%. Khoảng cách lớn này chỉ ra rằng các cơ chế điều chỉnh (regularization) hiện tại chưa đủ mạnh để ngăn chặn mô hình "học vẹt" trên tập dữ liệu quy mô nhỏ như UrbanSound8K.
- **Khó khăn với các lớp nhiễu nền:** Phân tích định tính cho thấy mô hình gặp khó khăn trong việc phân biệt các âm thanh có tính chất nhiễu (noise-like) như *air_conditioner*, *engine_idling* và *drilling*. Đây là hạn chế chung của việc sử dụng biểu diễn log-Mel spectrogram, nơi các đặc trưng tần số của các loại tiếng ồn này có sự chồng lấn lớn.

8.4 Đề xuất cải tiến và hướng phát triển

Để thu hẹp khoảng cách hiệu năng so với các kết quả SOTA trong bài báo gốc, một số cải tiến kỹ thuật được đề xuất:

1. **Tăng cường chiến lược Augmentation:** Bài báo gốc sử dụng Mixup với hệ số α cao và SpecAugment mạnh tay. Việc tăng cường độ của các kỹ thuật này có thể giúp giảm thiểu hiện tượng overfitting đang quan sát thấy trên UrbanSound8K.
2. **Sử dụng Model Ensemble:** Các kết quả tốt nhất trong bài báo gốc (trên AudioSet và ESC-50) đều đến từ việc kết hợp (ensemble) nhiều mô hình với các kích thước patch và stride khác nhau. Áp dụng kỹ thuật này (ví dụ: trung bình trọng số của các checkpoint từ các epoch khác nhau hoặc các fold khác nhau) hứa hẹn sẽ cải thiện đáng kể độ chính xác và độ ổn định.
3. **Tối ưu hóa Resolution:** Thử nghiệm với stride nhỏ hơn (ví dụ: thay vì 10 như hiện tại, giảm xuống 5) có thể giúp mô hình nắm bắt tốt hơn các đặc trưng thời gian ngắn của các lớp âm thanh khó như tiếng khoan (drilling) hay tiếng búa máy (jackhammer), mặc dù sẽ làm tăng chi phí tính toán.

9 Mô tả ứng dụng

10 Kết luận

11 Tài liệu tham khảo

- [1] Shawn Hershey et al. “CNN architectures for large-scale audio classification”. In: *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE. 2017, pp. 131–135. DOI: [10.1109/ICASSP.2017.7952132](https://doi.org/10.1109/ICASSP.2017.7952132).
- [2] Alexey Dosovitskiy et al. “An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale”. In: *arXiv preprint arXiv:2010.11929* (2020). URL: <https://arxiv.org/abs/2010.11929>.
- [3] Yuan Gong, Yu-An Chung, and James Glass. “AST: Audio Spectrogram Transformer”. In: *arXiv preprint arXiv:2104.01778* (2021). URL: <https://arxiv.org/abs/2104.01778>.
- [4] C.-H. Lee, C.-H. Lin, and B.-H. Juang. “A study on speaker adaptation of the parameters of continuous density hidden Markov models”. In: *IEEE Transactions on Signal Processing* 39.4 (1991), pp. 806–814. DOI: [10.1109/78.80902](https://doi.org/10.1109/78.80902).
- [5] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. “ImageNet Classification with Deep Convolutional Neural Networks”. In: *Advances in Neural Information Processing Systems*. Vol. 25. Curran Associates, Inc., 2012, pp. 1097–1105. URL: <https://proceedings.neurips.cc/paper/2012/file/c399862d3b9d6b76c8436e924a68c45b-Paper.pdf>.
- [6] YuanGongND. *GitHub - YuanGongND/ast: Code for the Interspeech 2021 paper “AST: Audio Spectrogram Transformer”*. en. URL: <https://github.com/YuanGongND/ast/tree/master>.
- [7] Justin Salamon and Juan Pablo Bello. *Deep Convolutional Neural Networks and Data Augmentation for Environmental Sound Classification*. 2016. arXiv: [1608.04363 \[cs.SD\]](https://arxiv.org/abs/1608.04363).