

VIETNAMESE TRADITIONAL MUSIC CLASSIFICATION

Nhóm 10:

- Trương Văn Khải - 21520274
- Hoàng Tiến Đạt - 21520696

Start

Phụ Lục

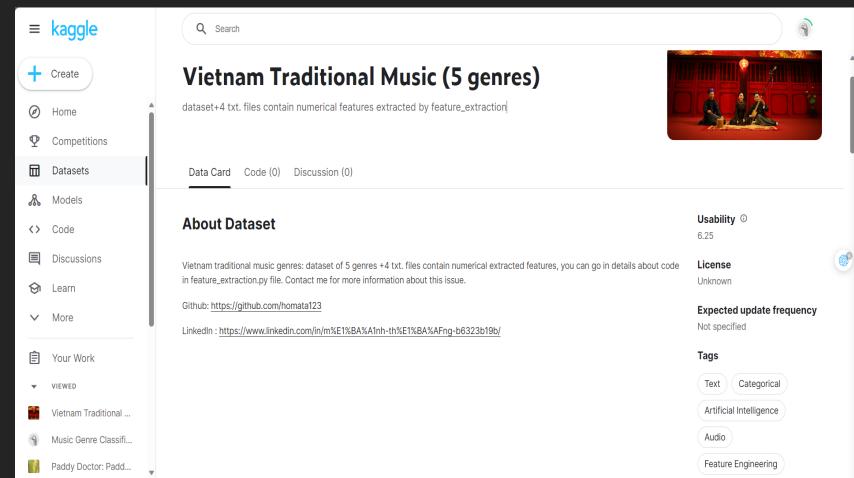
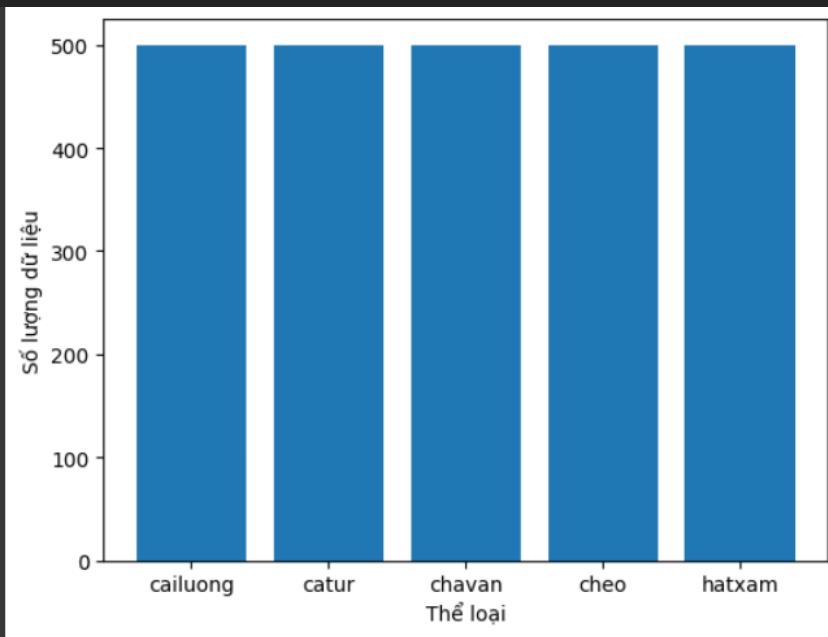
- 01. Giới thiệu bộ dữ liệu** →
- 02. Mục tiêu đồ án** →
- 03. Các tài liệu liên quan** →
- 04. Hướng tiếp cận** →
 - 04.01. Tiền xử lý dữ liệu âm thanh**
 - 04.02. Mô hình CNN**
- 05. Kết quả sơ bộ** →
- 06. Demo chương trình/ứng dụng** →
- 07. Tóm tắt kết quả đạt được** →
- 08. Hạn chế và các điểm cần cải tiến** →

01

Giới thiệu bộ dữ liệu

01. Giới thiệu bộ dữ liệu

Trong bài báo cáo này, chúng tôi lựa chọn bộ data set "Vietnam Traditional Music (5 genres)" trên Kaggle¹. Bộ dataset này được lấy từ hơn 20 giờ record, bao gồm 2500 file .wav được chia đều thành 5 thể loại nhạc: cai luong, catru, chau van, cheo, hat xam. Mỗi file gồm khoảng 30 giây và được chia thành 500 file mỗi thể loại.



[1]: <https://www.kaggle.com/datasets/homata123/vntm-for-building-model-5-genres/data?select=VNTM3>

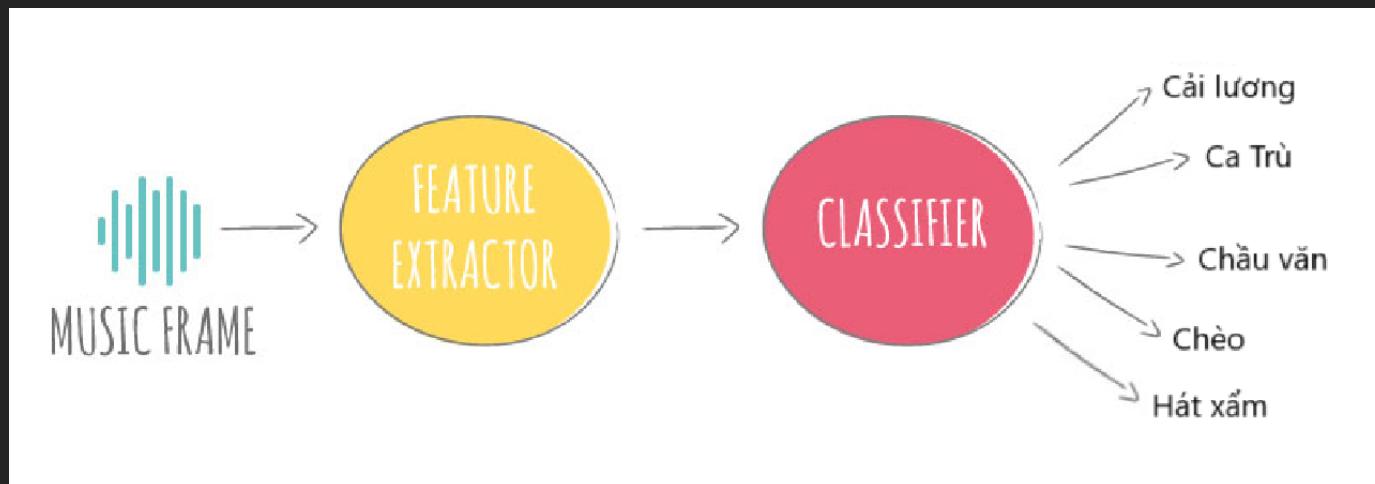
02

Mục tiêu đồ án

02. Mục tiêu chung

Mục Tiêu của bài toán:

- Input: Một đoạn nhạc tiếng việt tầm 30s (ở định dạng .wav, .mp3, ...).
- Output: Thể loại của bài nhạc đó thuộc vào 5 output như: cailuong, catru, chauvan, cheo, hatxam.



Cuối cùng, chúng tôi sẽ cố gắng tìm hiểu để xây dựng thêm 1 phương pháp mới để so sánh hiệu suất của cả hai phương pháp trên cùng 1 bộ dữ liệu.

03

Các tài liệu liên quan

03. Tổng hợp tài liệu

Nguồn tài liệu được tổng hợp trên
paperswithcode.com

[¹] Explaining Deep Convolutional Neural Networks on Music Classification - Keunwoo Choi -
<https://arxiv.org/pdf/1607.02444v1.pdf>

[²] Receptive-Field Regularized CNNs for Music Classification and Tagginger - Khaled Koutini -
<https://arxiv.org/pdf/2007.13503v1.pdf>

[³] Convolutional Recurrent Neural Networks for Music Classification - Keunwoo Choi -
<https://arxiv.org/pdf/1609.04243v3.pdf>

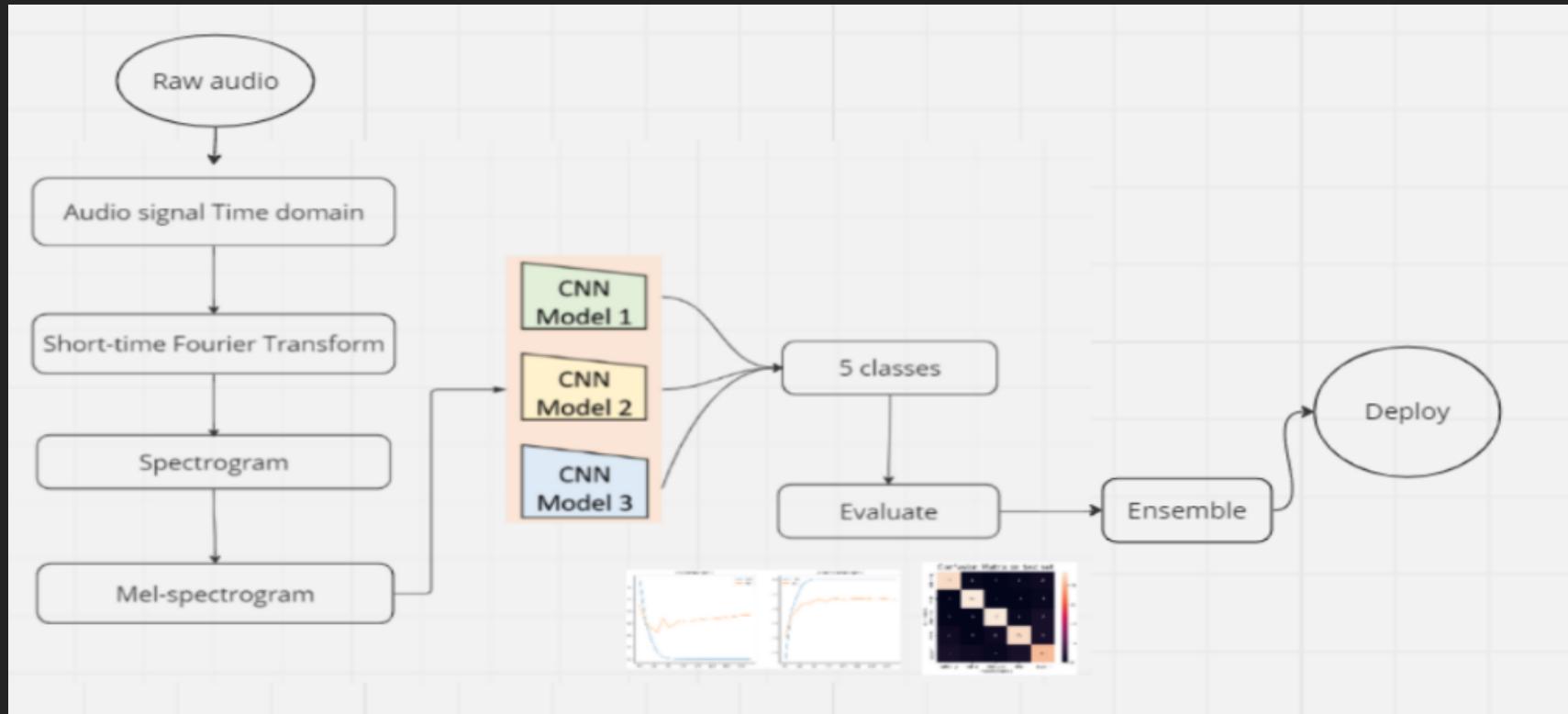
[⁴] Multi-Level and Multi-Scale Feature Aggregation Using Sample-level Deep Convolutional Neural Networks for Music Classification - Jongpil Lee - <https://arxiv.org/pdf/1706.06810v1.pdf>

[⁵] Preprocess Audio Data with the Signal Envelope - Preprocess Audio Data with the Signal Envelope -
<https://towardsdatascience.com/preprocess-audio-data-with-the-signal-envelope-499e6072108>

04

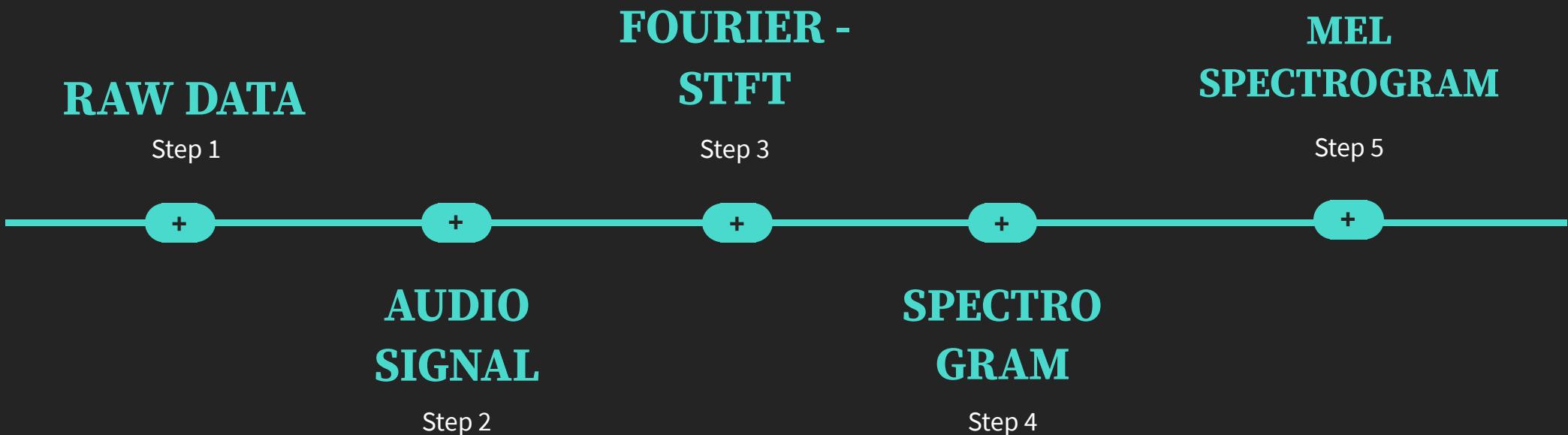
Hướng tiếp cận

04. Hướng tiếp cận



Pipeline cho Audio Classification

04.01. Tiền xử lý dữ liệu âm thanh



A. Rút trích đặt trưng âm thanh

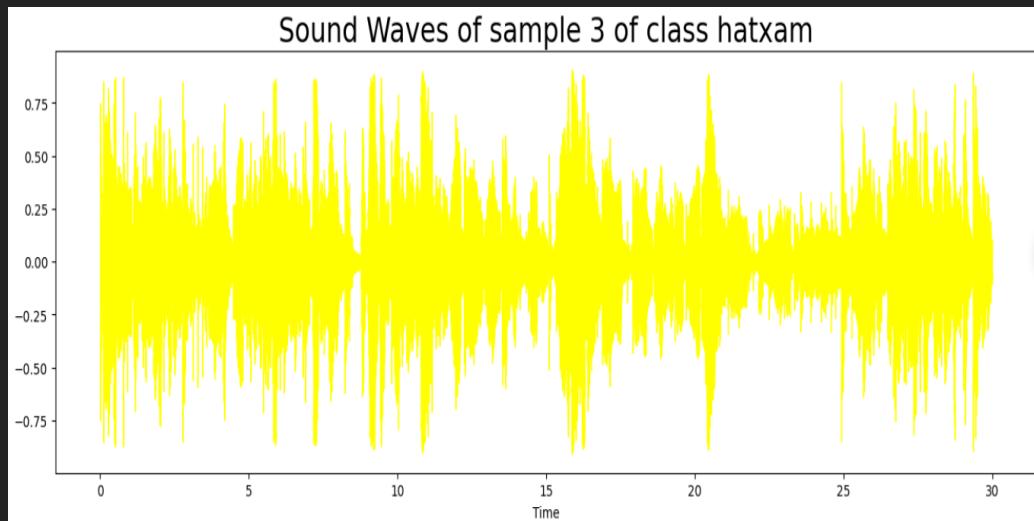


Fig1: Biểu diễn dạng biểu đồ đường cho Audio Signal
cho 1 file âm thanh .wav dài 30s.

Âm thanh sẽ được định dạng nén ở 1 tệp .wav, khi dùng hàm để tải lên nó sẽ được giải nén và chuyển đổi thành 1 mảng numpy.

```
[ 0.02862549  0.02334595  0.0133667   ... -0.00460815 -0.00561523  
-0.00732422]
```

Mỗi phần tử trong mảng này đại diện cho biên độ của sóng âm thanh ở 1/sample_rate khoảng thời gian của giây.

Ví dụ:

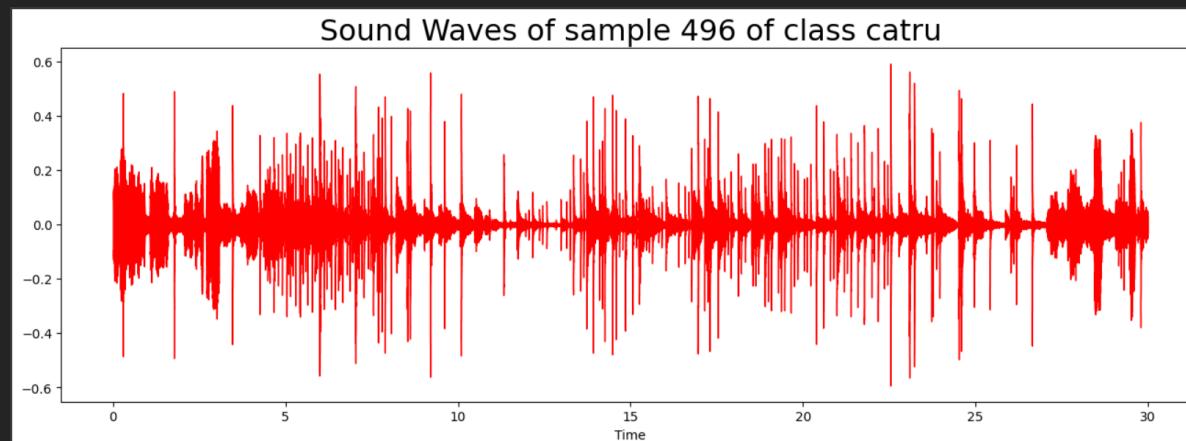
- Ví dụ với file âm thanh ở trên dài 30s với sample rate là 16,000hz thì số lượng samples của file sẽ là * 16,000 = 480,000
- Biên độ của tần số ở giây thứ nhất sẽ là:

```
print(samples[16000])
```

```
→ -0.0060424805
```

A. Rút trích đặc trưng âm thanh - Cài đặt tham số

Các bản các file âm thanh được lấy mẫu lại thành tần số 22050 Hz.



Sau đó, sử dụng thuật toán Short Time Fourier Transform với cấu hình kích thước cửa sổ Hann tương ứng là: n_fft = 2048, hop_length = 512. Trong đó:

- n_fft: kích thước Time-Window, hay chiều dài mỗi Time-Section.
- hop_length: Có thể hiểu nó tương tự như Stride trong CNN, tức là số bước trượt/nhảy tính theo đơn vị Hop của Time-Window

B. Short-time Fourier Transform (STFT)

Theo Wikipedia, biến đổi Fourier hay chuyển hóa Fourier, được đặt tên theo nhà toán học người Pháp Joseph Fourier, là **phép biến đổi một hàm số hoặc một tín hiệu theo miền thời gian sang miền tần số**. Chẳng hạn như một bản nhạc có thể được phân tích dựa trên tần số của nó.²

Thì STFT là 1 sơ đồ phân tích cục bộ cho tần số đại diện thời gian (time-frequency representation -TFR) :

- Nó sẽ phân đoạn tín hiệu theo từng đoạn thời gian hẹp (đủ hẹp để có thể được coi là đứng yên).
- Sau đó sẽ lấy phép biến đổi Fourier theo từng đoạn.

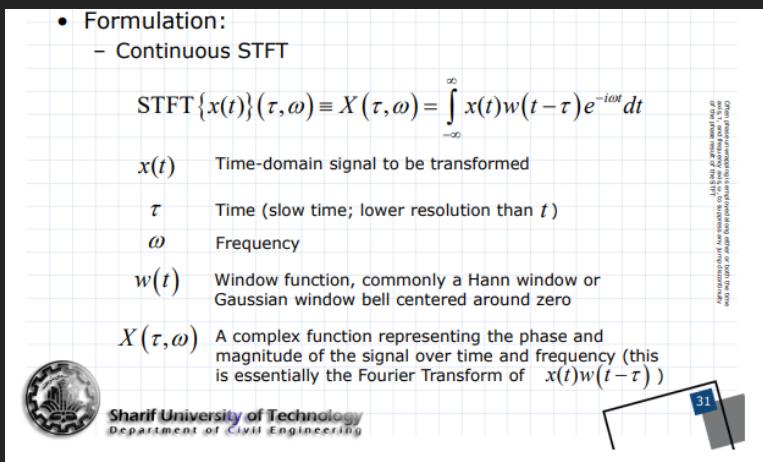


Fig2: Công thức biến đổi STFT ³

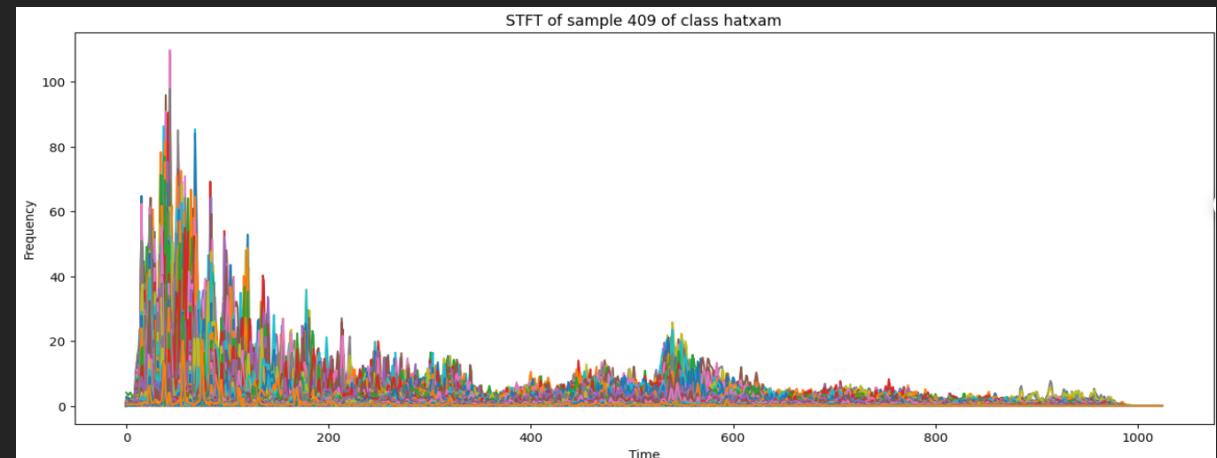


Fig3: Biểu đồ biến đổi STFT của 1 file .wav trong bộ dataset

Như vậy, phép biến đổi STFT sẽ huyển đổi một tín hiệu từ miền thời gian sang miền tần số, đồng thời chúng còn huyển đổi một tín hiệu từ miền thời gian sang miền tần số. Tuy nhiên, hạn chế của biểu diễn miền tần số là không có thông tin về thời gian.

C. Spectrogram

Trong phần trước, chúng ta đã biểu tín hiệu thành các giá trị tần số của nó, chúng sẽ đóng vai trò là features cho mạng nơ ron nhận dạng giọng nói. Nhưng khi áp dụng STFT thì chúng chỉ cung cấp các giá trị tần số và chúng ta bị mất dấu thông tin thời gian. Do đó, chúng ta cần tìm một cách khác để tính toán các features sao cho các giá trị tần số và thời gian đều được quan sát. Spectrogram có thể giải quyết được vấn đề này.

Biểu diễn trực quan các tần số của một tín hiệu nhất định với thời gian được gọi là Spectrogram. Trong biểu đồ spectrogram, một trục biểu thị thời gian, một trục biểu thị tần số và màu sắc sẽ biểu thị biên độ của tần số được quan sát tại 1 thời điểm. Màu sắc càng sáng sẽ biểu thị tần số mạnh, và ngược lại.

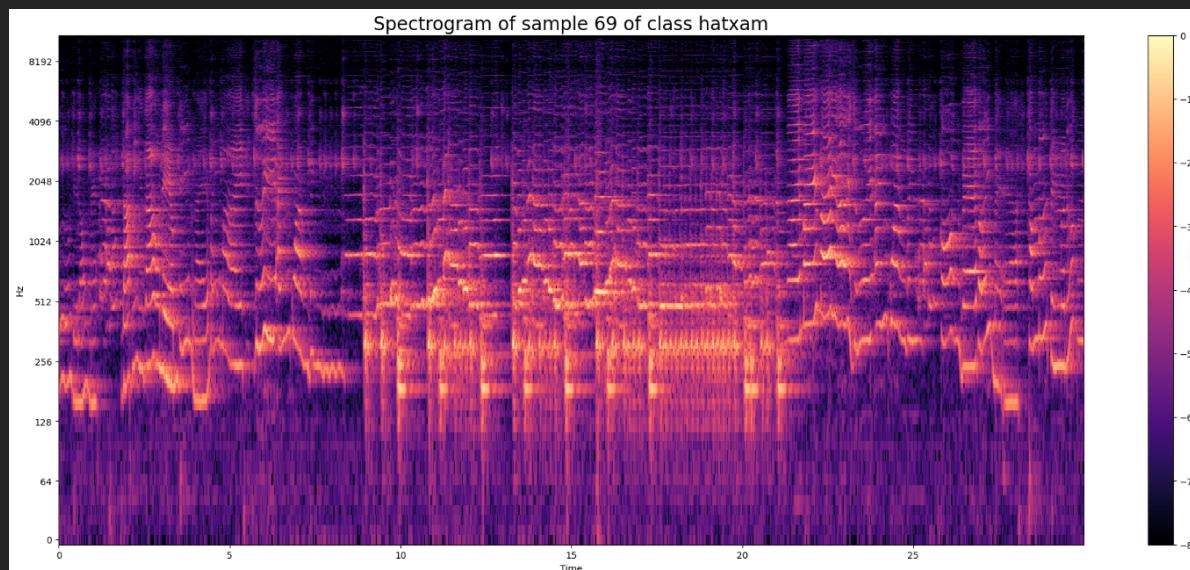
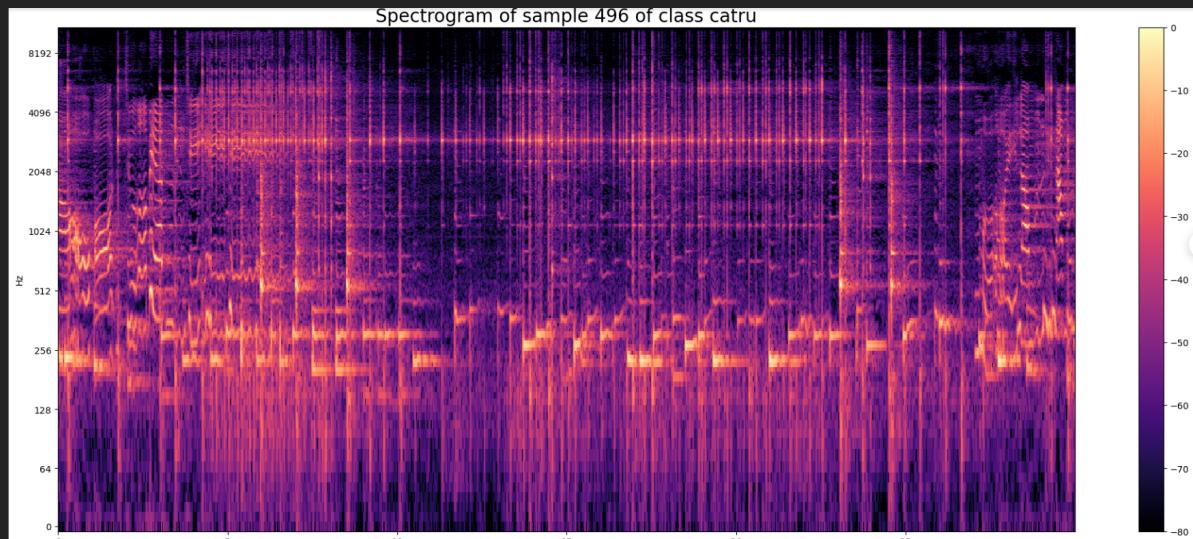


Fig4: Spectrogram biểu diễn cho một file .wav trong bộ dataset

C. Spectrogram - Cài đặt tham số

Với kết quả của phép biến đổi STFT, chúng ta sẽ sử dụng lại và biến đổi thành các biểu đồ Spectrogram bằng việc sử dụng thư viện Librosa. HOP_LENGTH ở bước biến đổi này tương ứng sẽ là 512.



D. Mel - Spectrogram

MEL SCALE

Con người không thể cảm nhận được các tần số trên thang đo tuyến tính. Ví dụ, chúng ta rất dễ cảm nhận được sự khác biệt giữa âm thanh 100Hz và 200Hz, tuy nhiên chúng ta khó có thể nhận ra sự khác biệt giữa âm thanh 10000Hz và 10100Hz, mặc dù khoảng cách giữa 2 bộ âm thanh này là như nhau.

Đây là cách con người chúng ta cảm nhận tần số, chúng ta nghe thấy âm thanh ở thang đo logarit chứ không phải thang đo tuyến tính. Sự chuyển đổi từ thang đo Hertz sang thang đo Mel như sau:

$$m = 1127 \cdot \log\left(1 + \frac{f}{700}\right)$$

Fig5: Công thức chuyển đổi từ thang đo Hz sang Mel

D. Mel - Spectrogram

DECIBLE SCALE

Trên thang đo này, 0 dB là hoàn toàn im lặng. Từ đó, đơn vị đo tăng theo cấp số nhân. 10 dB to hơn 10 lần so với 0 dB, 20 dB to hơn 100 lần và 30 dB to hơn 1000 lần. Ở thang âm này, âm thanh trên 100 dB bắt đầu trở nên to đến mức không thể chịu nổi.

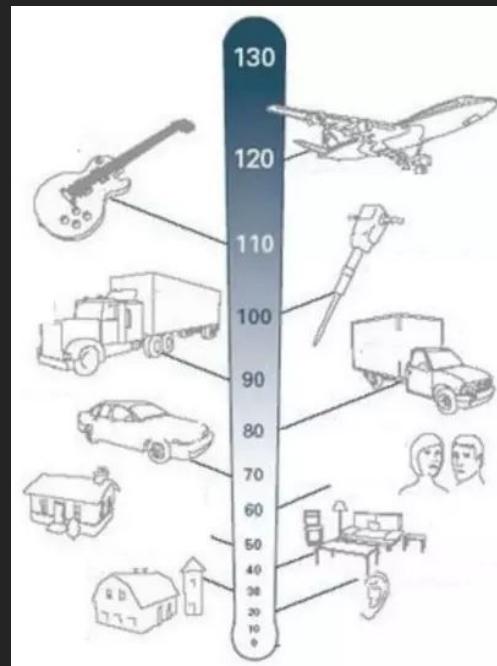


Fig6: Thang đo Decible trên các phương tiện thường ngày

D. Mel - Spectrogram

Để xử lý âm thanh một cách chân thực và gần giống con người nhất, cách xử lý của Mel Spectrogram như sau:

1. Tần số (trục y) được thay thế bằng giá trị Logarithmic của nó, gọi là Mel Scale.
2. Biên độ được thay thế bằng giá trị Logarithmic của nó, gọi là Decibel Scale để chỉ ra màu sắc.

Vì vậy, chúng ta cùng thử sử dụng thang đo Decible Scale thay vì biên độ để biểu diễn biểu đồ Mel - Spectrogram:

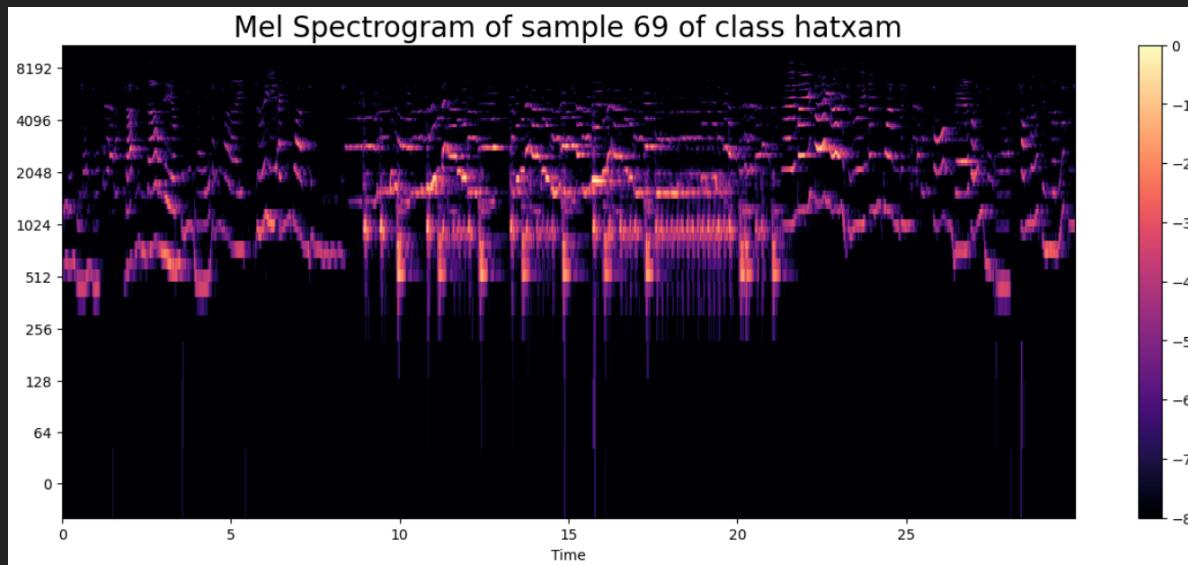


Fig7: Mel - Spectrogram cho một file .wav trong bộ dataset

D. Mel - Spectrogram - Cài đặt tham số

Cuối cùng, tạo ra một biểu đồ log-Mel spectrogram có kích thước $128 \times 1290 \times 3$ từ một đoạn âm thanh kéo dài 30 giây.



04.02. Mô hình CNN

MODEL 1

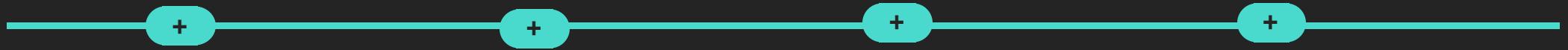


TABLE I: Model 1 Architecture

Layers	Output
Conv[3x3] @ 16 - Relu - MP [2x2]	(63, 645, 16)
Conv[3x3] @ 32 - Relu - MP [2x2]	(30, 321, 32)
Conv[3x3] @ 64 - Relu - MP [2x2]	(14, 159, 64)
Conv[3x3] @ 64 - Relu - MP [2x2]	(6, 78, 64)
Flatten-Dropout(0.2)	29952
FC	7667968
FC	256
FC	128
FC-Softmax	5

MODEL 2

**COMBINED
MODEL**

TABLE III: Model 3 Architecture

Layers	Output
Conv[5x5] @ 16 - Relu - MP [2x2]	(62, 644, 16)
Conv[3x3] @ 32 - Relu - MP [2x2]	(30, 321, 32)
Conv[3x3] @ 64 - Relu - MP [2x2]	(14, 159, 64)
Conv[3x3] @ 64 - Relu - MP [2x2]	(6, 78, 64)
Flatten-Dropout(0.2)	29252
FC-Dropout(0.2)-BN	128
FC-Dropout(0.2)-BN	64
FC-Softmax	5

TABLE II: Model 2 Architecture

Layers	Output
Conv[5x5] @ 32 - Relu - MP [2x2]	(64, 646, 32)
Conv[3x3] @ 32 - Relu - MP [2x2]	(32, 323, 32)
Conv[3x3] @ 64 - Relu - MP [2x2]	(16, 161, 64)
Flatten-Dropout(0.2)	164864
FC	128
FC	64
FC-Softmax	5

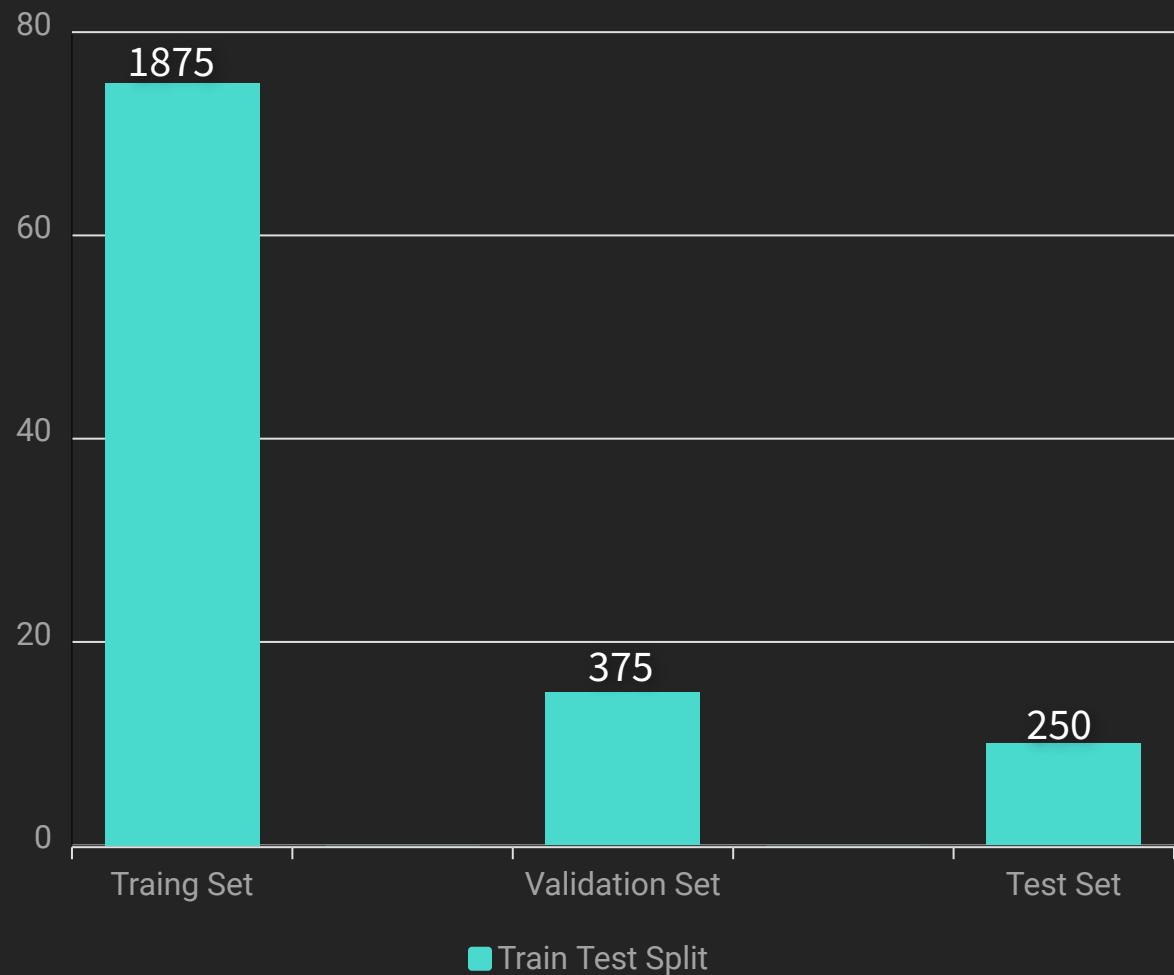
MODEL 3

A. Train-Validation-Test Split

Sau khi đã cho tất cả các file âm thanh chạy qua phép biến đổi thanh Mel-Spectrogram, chúng tôi bắt đầu chia toàn bộ 2500 file âm thanh thành 3 bộ Train-Val-Test.

Tỷ lệ tập Train, tập Validation và tập Test lần lượt là 0.75, 0.15, 0.10. Kết quả thu được sẽ là: 1875 ảnh ở tập train, 375 ảnh ở tập val và 250 ảnh ở tập test.

Chúng tôi thực hiện huấn luyện tuần tự từng Model với số epochs, batch_size, validation_batch_size lần lượt là: 50, 32, 32.



B. Kiến trúc Model 1

Với model đầu tiên, cấu trúc của mạng Convolution Neuron Network được thiết kế với 4 lớp Convolution và Max Pooling để chiết tách các đặc trưng của Mel-spectrogram với kích thước mỗi lớp filter lần lượt là 3x3, 2x2, số lượng filter lần lượt là: 16,32,64 và 64. Sau đó được Flatten với dropout rate là 0.2 để truyền vào các lớp fully connected. Lớp FC đầu tiên có output size lớn hơn output size của lớp Flatten có mục đích tăng khả năng phân tích các đặc trưng phức tạp của mô hình. Các lớp FC còn lại dùng để giảm chiều và phân lớp.

Dưới đây là sơ đồ biểu diễn thông số các lớp trong mô hình 1:

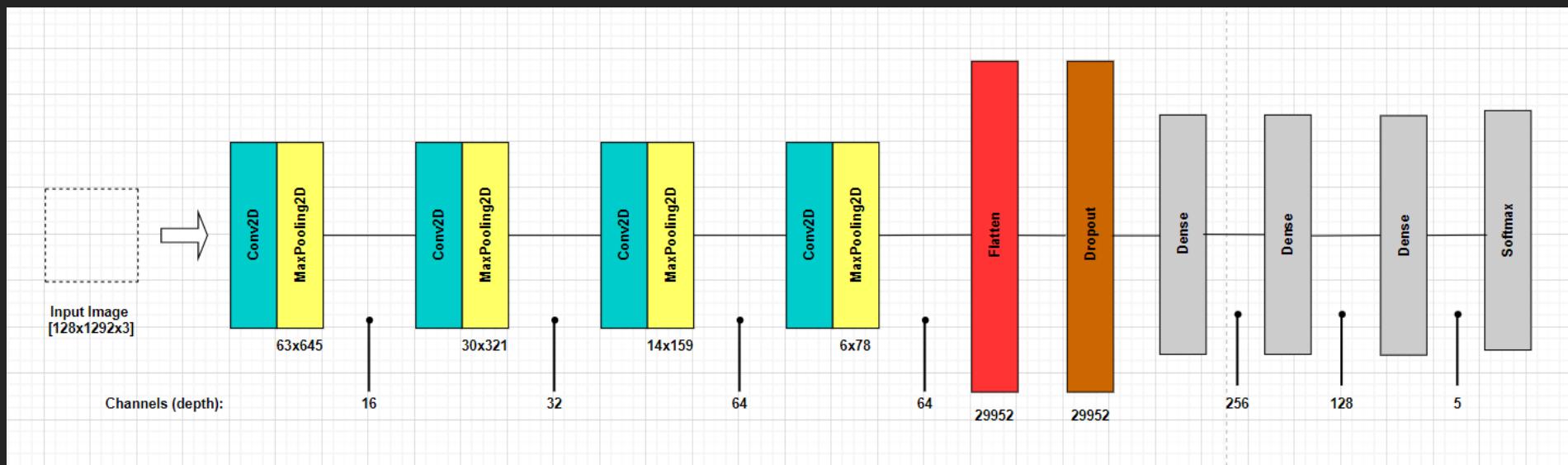


Fig8: Tổng quan cấu trúc các lớp mô hình 1

C. Kiến trúc Model 2

Khác với model 1, model 2 có sự thay đổi về cả số lớp convolution, kích cỡ của filter và số lượng filter trong lớp convolution đầu tiên. Do đó, tổng quan model 2 sẽ được thu nhỏ lại và số lớp Fully Connected cũng giảm xuống còn 3 do không còn lớp FC đầu tiên của model 1.

Dưới đây là sơ đồ biểu diễn thông số các lớp trong mô hình 2:

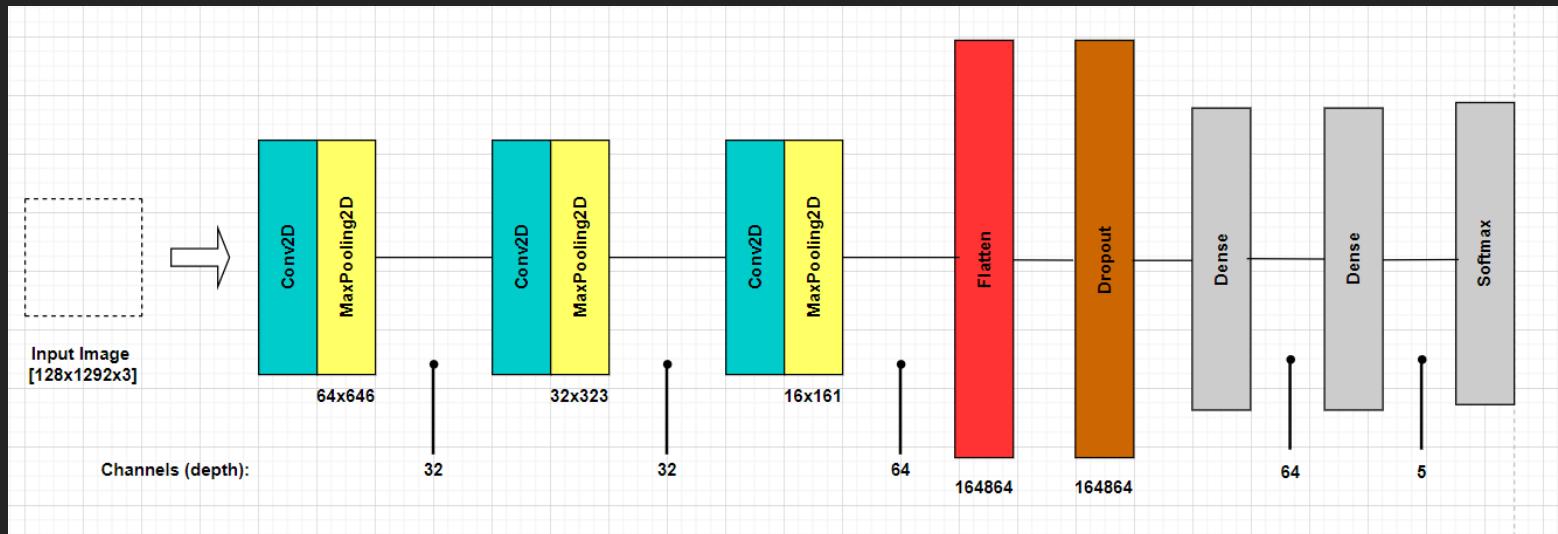


Fig9: Tổng quan cấu trúc các lớp mô hình 2

D. Kiến trúc Model 3

Với model 3, số lượng các lớp Convolution và số lượng filter của mỗi lớp không thay đổi so với model 1 tuy nhiên kích cỡ filter của lớp đầu tiên được thay đổi cho giống model 2 (5x5). Ngoài ra, model 3 còn có Dropout và Batch Normalization ở mỗi lớp Fully Connected để tương đồng với số lớp của model 2. Do đó, model 3 có số lớp convolution của model 1 và số lớp Fully Connected của model 2

Dưới đây là sơ đồ biểu diễn thông số các lớp trong mô hình 3:

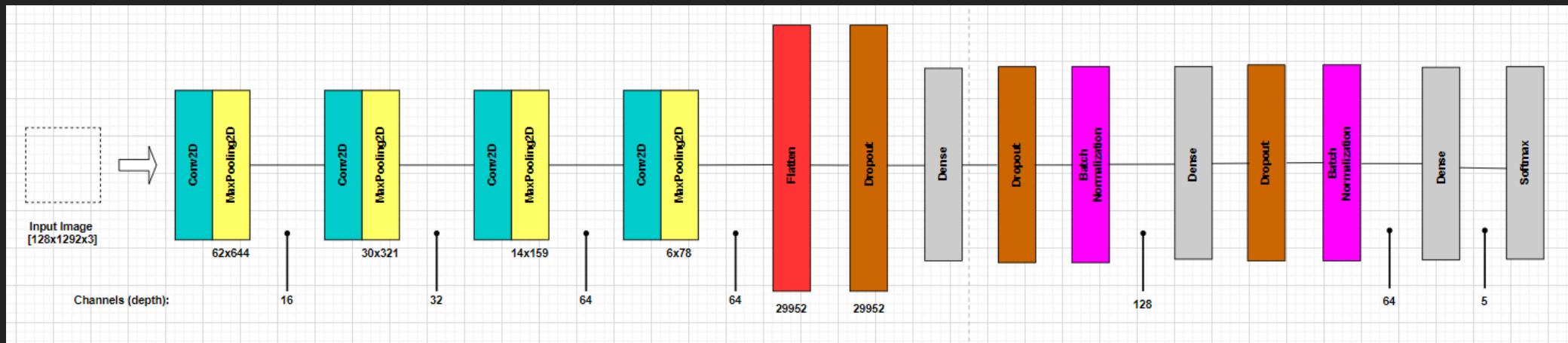


Fig10: Tổng quan cấu trúc các lớp mô hình 3

E. Combined model - PROD (Mô hình kết hợp)

Từ kết quả khá tốt của cả 3 kiến trúc mạng trên, đồng thời để tăng hơn nữa hiệu suất của mô hình, chúng tôi đề xuất phương pháp tổ hợp tổng hợp để kết hợp các vector xác suất dự đoán của cả 3 mô hình. Với mỗi model đều có cấu trúc mạng conv khác nhau, chúng có thể tìm ra những đặc điểm khác nhau trong một sample, do đó combined model sẽ có được sự tổng hợp của kết quả dự đoán từ những đặc trưng tìm được của 3 model.

$$P_{prod} = [p_1, p_2, p_c] \text{ where } P_i = \frac{1}{S} \prod_{s=1}^S P_{si} \text{ for } 1 \leq i \leq C$$

$$\hat{Y} = \text{argmax}(P_{prod}) = \text{argmax}(p_1, p_2, \dots, p_c)$$

Fig11: Công thức dự đoán của combined model với S là số phân lớp, C là số model

E. Combined model - PROD (Mô hình kết hợp)

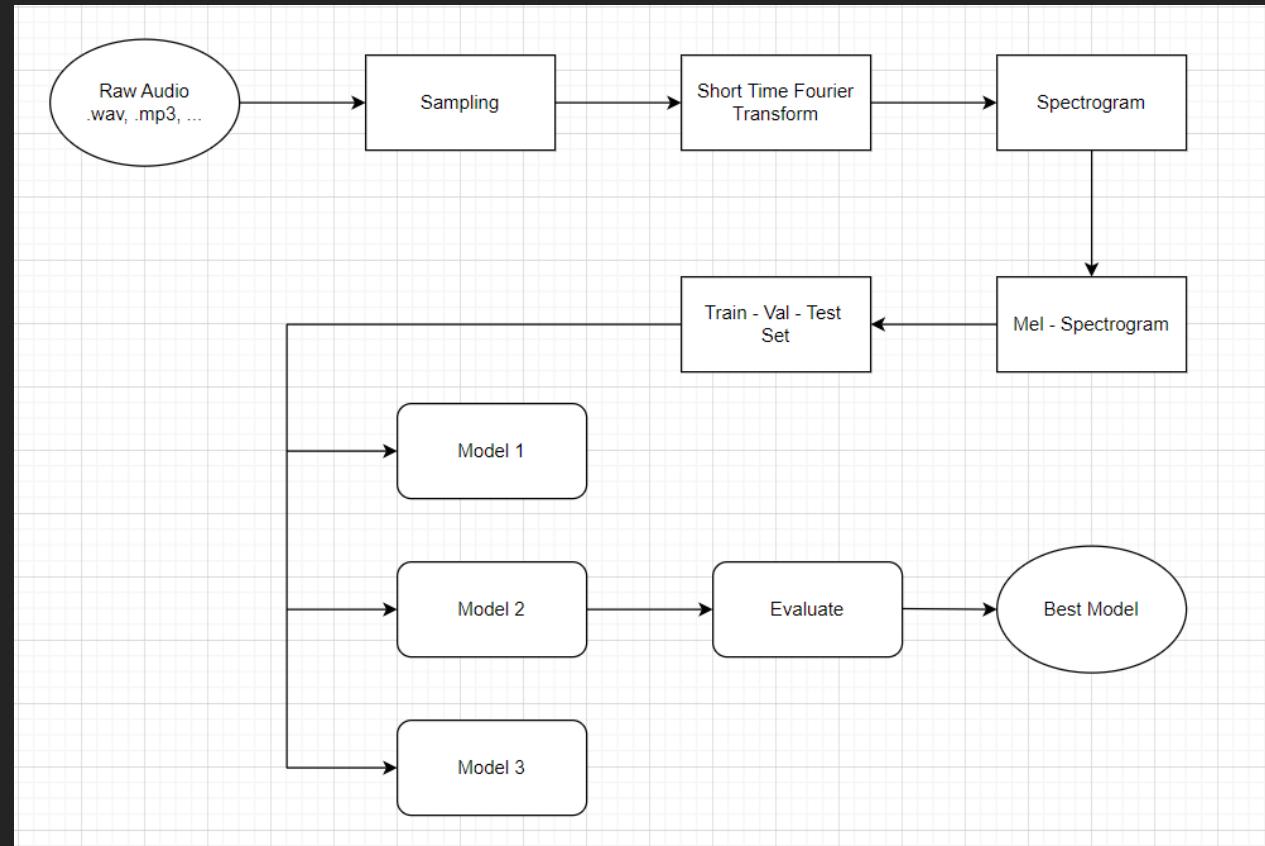
Từ kết quả khá tốt của cả 3 kiến trúc mạng trên, đồng thời để tăng hơn nữa hiệu suất của mô hình, chúng tôi đề xuất phương pháp tổ hợp tổng hợp để kết hợp các vector xác suất dự đoán của cả 3 mô hình. Với mỗi model đều có cấu trúc mạng conv khác nhau, chúng có thể tìm ra những đặc điểm khác nhau trong một sample, do đó combined model sẽ có được sự tổng hợp của kết quả dự đoán từ những đặc trưng tìm được của 3 model.

$$P_{prod} = [p_1, p_2, p_c] \text{ where } P_i = \frac{1}{S} \prod_{s=1}^S P_{si} \text{ for } 1 \leq i \leq C$$

$$\hat{Y} = \text{argmax}(P_{prod}) = \text{argmax}(p_1, p_2, \dots, p_c)$$

Fig11: Công thức dự đoán của combined model với S là số phân lớp, C là số model

F. Tóm tắt phương pháp tiếp cận của nhóm



Pipeline cho Audio Classification

05

Phương Pháp Đánh Giá

05. Phương pháp đánh giá

Về các chỉ số đánh giá được sử dụng cho một nhiệm vụ phân loại đa lớp điển hình, trong báo cáo này, chúng tôi áp dụng các phương pháp đánh giá sau:

1. Accuracy: Giả sử C là số lượng mẫu kiểm tra âm thanh/hình ảnh được phân loại đúng và tổng số mẫu kiểm tra âm thanh/hình ảnh là T , độ chính xác phân loại (Acc. (%)) được tính bằng tỷ lệ C chia cho T .
2. Precision: Độ chính xác cho một lớp cụ thể trong phân loại đa lớp là tỷ lệ số trường hợp được phân loại đúng là thuộc về một lớp cụ thể (True Positive: TP) trên tổng số trường hợp mà mô hình dự đoán thuộc về lớp đó (True Positive + False Positive: TP + FP).
3. Recall: Độ phủ sóng trong phân loại đa lớp là tỷ lệ số trường hợp trong một lớp mà mô hình phân loại đúng (True Positive: TP) trên tổng số trường hợp trong lớp đó (True Positive + False Negative: TP + FN).
4. Confusion Matrix: Ma trận nhầm lẫn là một bảng mô tả hiệu suất của một mô hình phân loại trên một tập dữ liệu mà giá trị thật sự đã được biết trước. Nó được sử dụng để trực quan hóa hiệu suất của một thuật toán phân loại bằng cách hiển thị số lượng phân loại đúng và sai mà thuật toán đã thực hiện.

A. Model 1

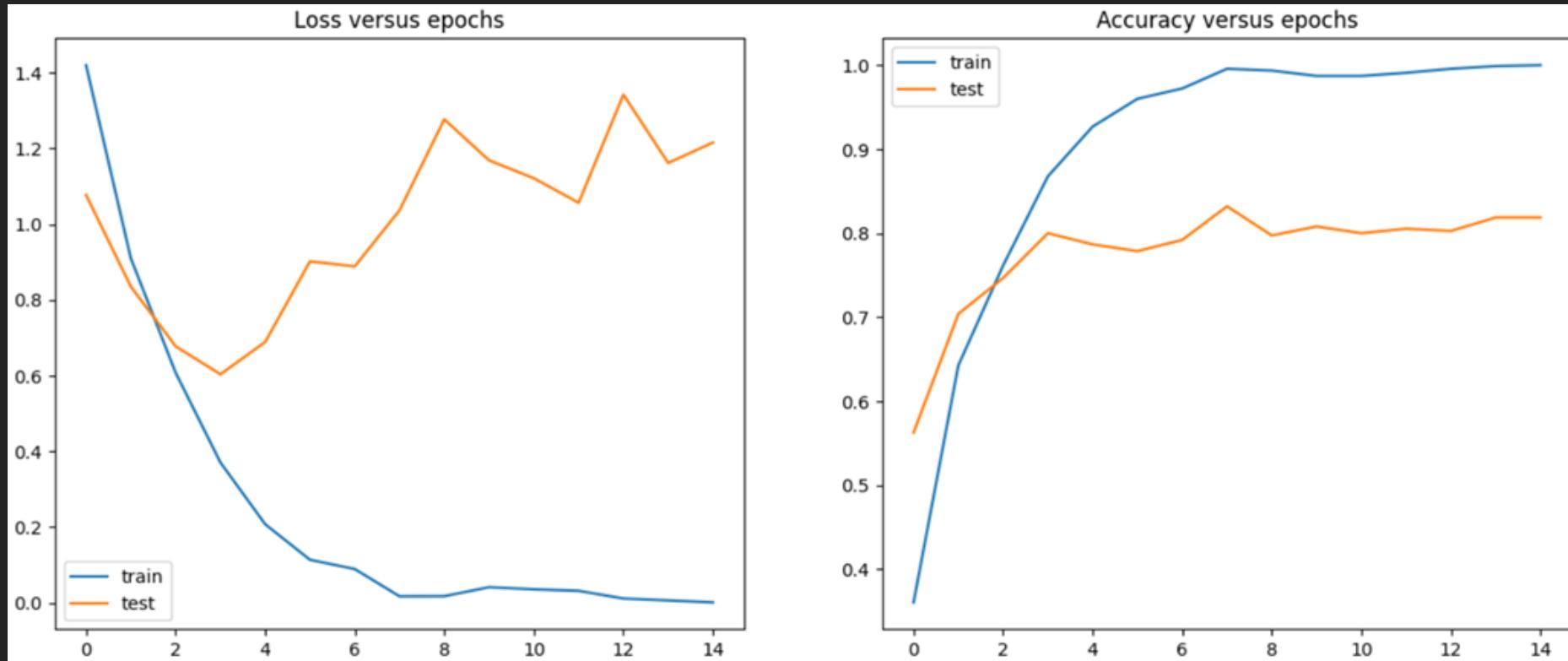
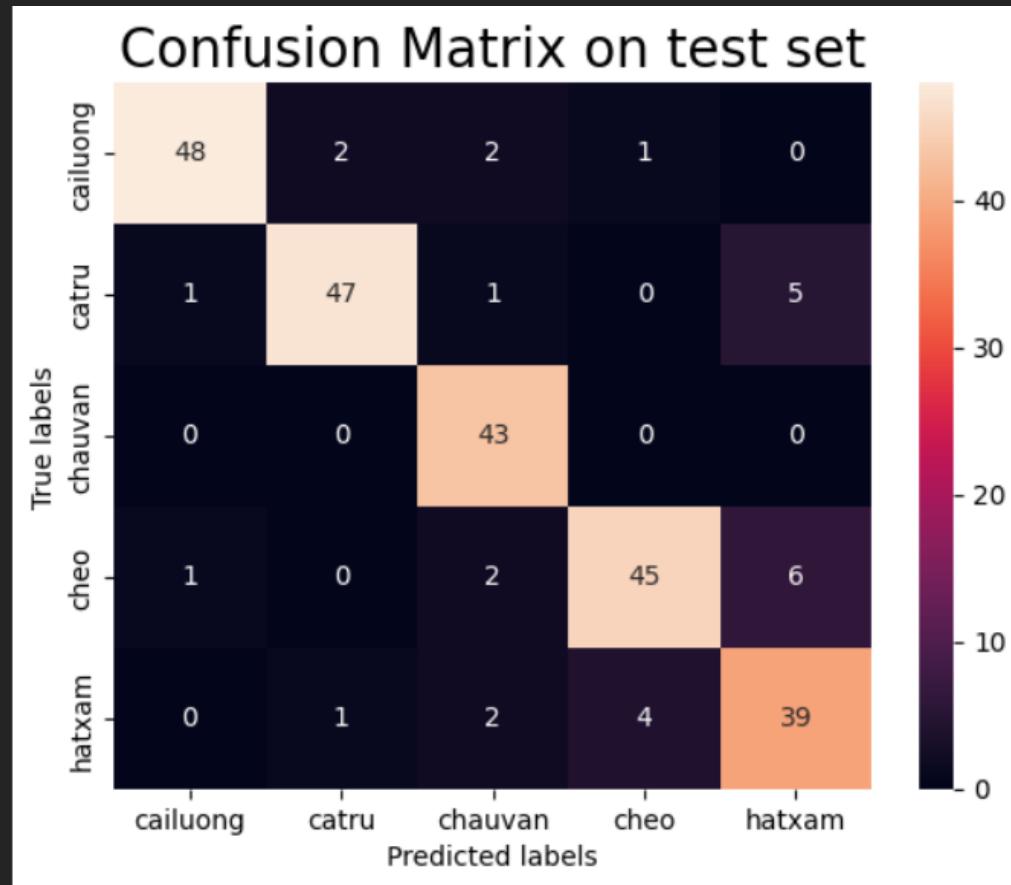


Fig12: Training history của Model 1

A. Model 1



Đánh giá hiệu suất của Model 1:

	precision	recall	f1-score	support
Cai Luong	0.96	0.91	0.93	53
Ca tru	0.94	0.87	0.90	54
Chau Van	0.86	1.00	0.92	43
Cheo	0.90	0.83	0.87	54
Hat Xam	0.78	0.85	0.81	46
accuracy			0.89	250
macro avg	0.89	0.89	0.89	250
weighted avg	0.89	0.89	0.89	250

Fig13: Confusion Matrix của Model 1

B. Model 2

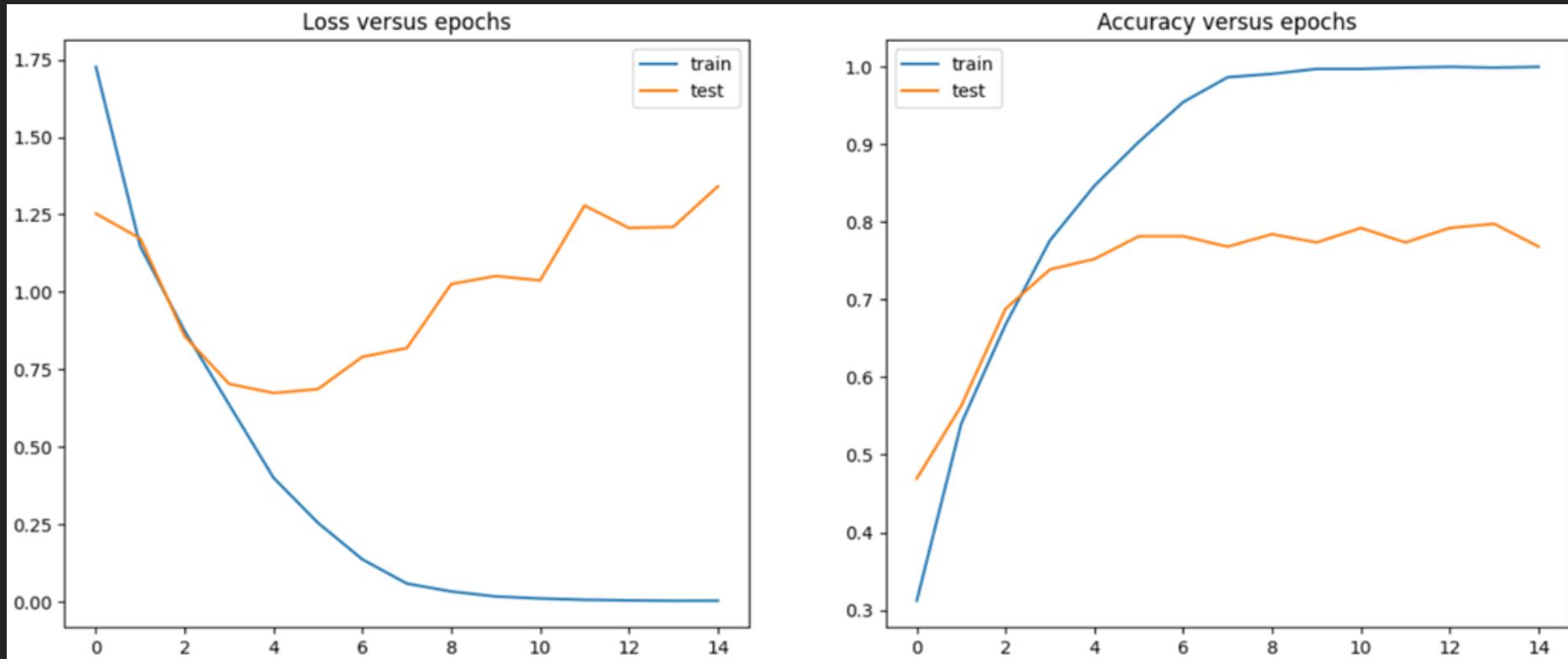
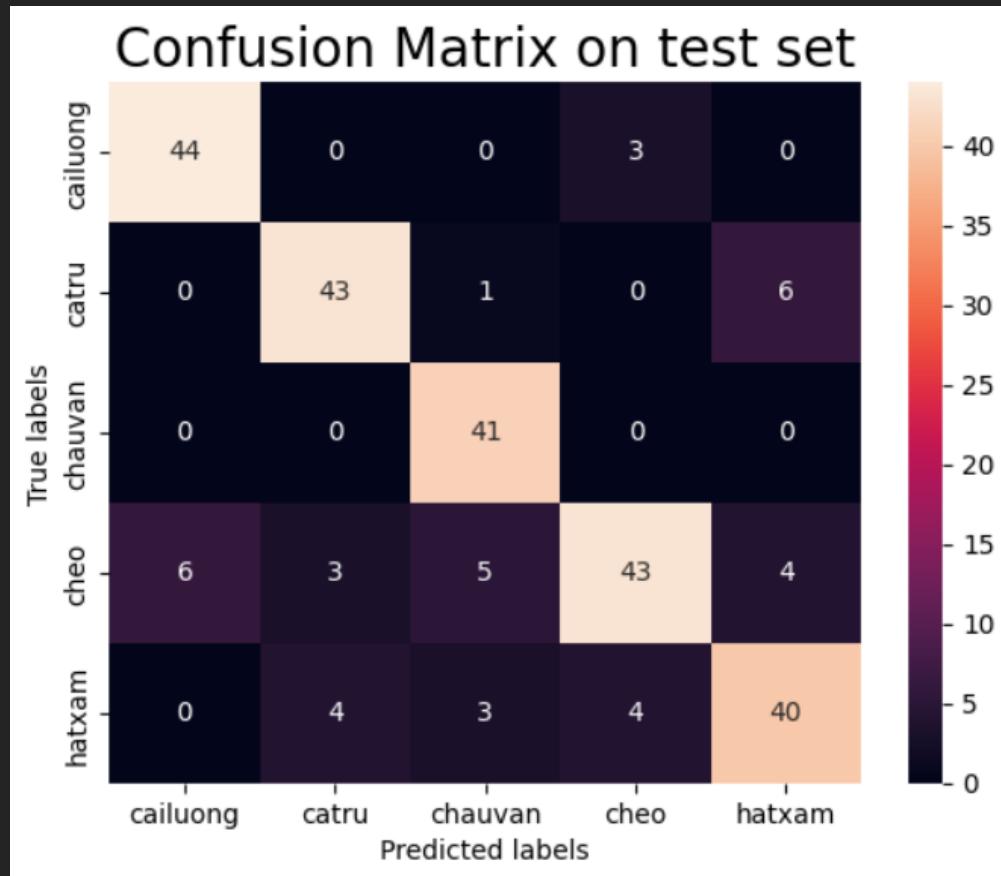


Fig14: Training history của Model 2

B. Model 2



Đánh giá hiệu suất của Model 2:

	precision	recall	f1-score	support
Cai Luong	0.88	0.94	0.91	47
Ca tru	0.86	0.86	0.86	50
Chau Van	0.82	1.00	0.90	41
Cheo	0.86	0.70	0.77	61
Hat Xam	0.80	0.78	0.79	51
accuracy			0.84	250
macro avg	0.84	0.86	0.85	250
weighted avg	0.84	0.84	0.84	250

Fig15: Confusion Matrix của Model 2

C. Model 3

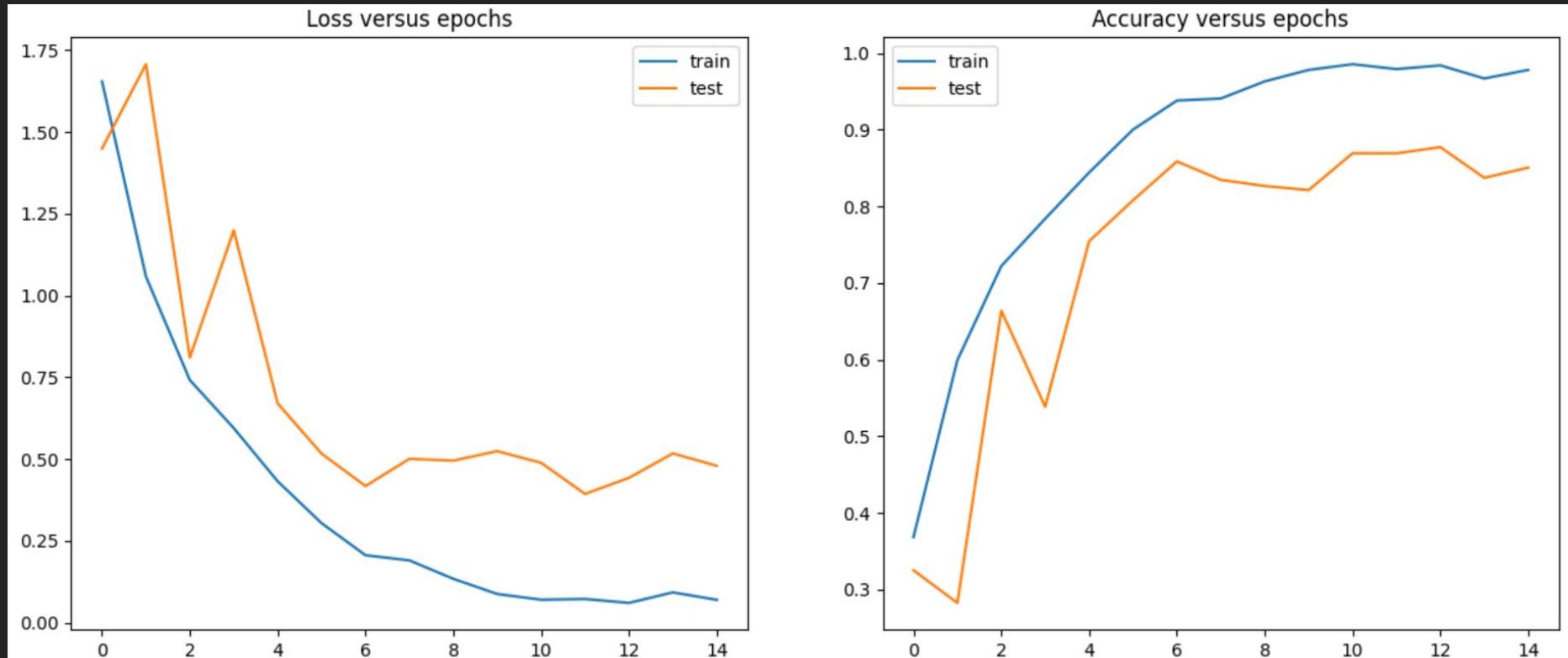
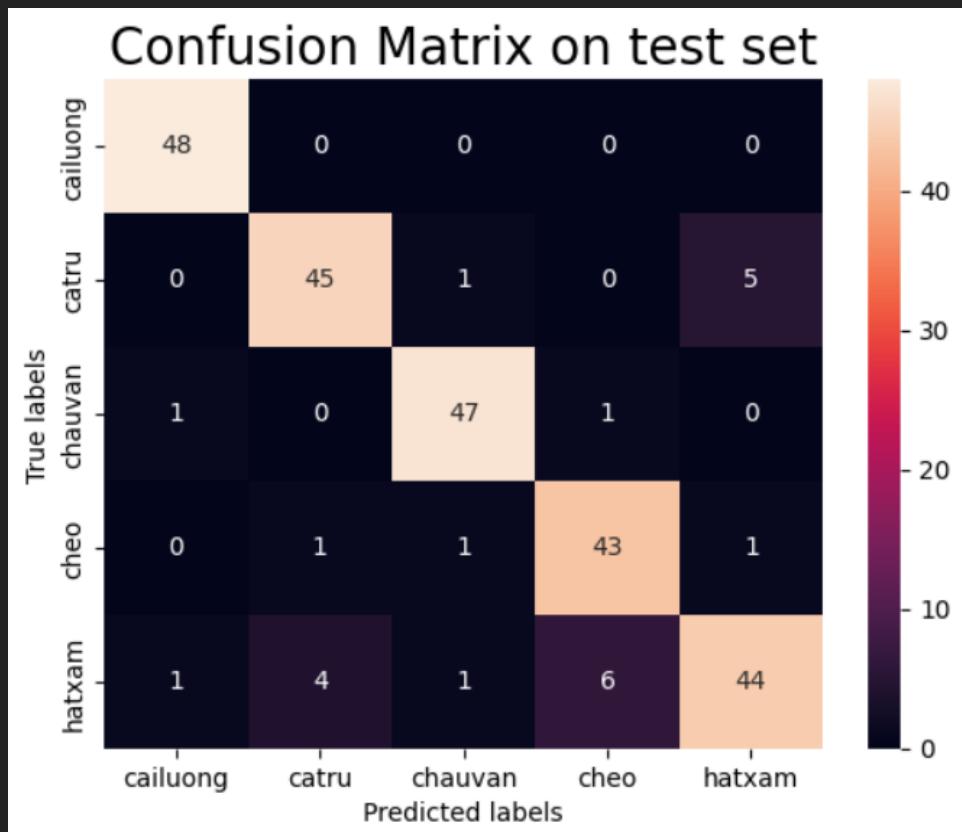


Fig16: Training history của Model 3

C. Model 3



Đánh giá hiệu suất của Model 3:

	precision	recall	f1-score	support
Cai Luong	0.96	1.00	0.98	48
Ca tru	0.90	0.88	0.89	51
Chau Van	0.94	0.96	0.95	49
Cheo	0.86	0.93	0.90	46
Hat Xam	0.88	0.79	0.83	56
accuracy			0.91	250
macro avg	0.91	0.91	0.91	250
weighted avg	0.91	0.91	0.91	250

Fig17: Confusion Matrix của Model 3

D. Tổng kết sơ bộ kết quả đạt được

Nhìn chung, kết quả của cả 3 mô hình đều khá tốt trên cả tập Val và tập Test. Ngoài ra, chúng tôi không xây dựng những mô hình CNN khá sâu để tránh 'gradient explosion' hoặc 'gradient vanishing' nhưng vẫn đảm bảo độ chính xác tốt.

Trong cả hai bộ, mô hình 3 tốt hơn mô hình 2 và mô hình 1 ở hầu hết các số liệu đánh giá. Accuracy, precision, recall đạt được trên tập test là 0.91, 0.91, 0.91.

Cuối cùng, bài báo cáo này đã trình bày một quy trình cho bài toán Phân loại âm thanh và kết quả của nhiều mô hình CNN dựa trên bài toán này. Chúng tôi đã trình bày phương pháp cho Vietnamese Traditional Classification (5 thể loại) và đạt được thành tích tốt về độ chính xác, và hy vọng nó có thể đóng góp phần nào đó vào một số nhiệm vụ thực tế khác.

E. Inference

Chúng tôi thực hiện test với toàn bộ dữ liệu mới mà chúng tôi tự tìm trên mạng. Mỗi sample là một bài hát hoàn chỉnh có độ dài từ 3 - 8 phút

```
✓ 0 giây [86] chauvan_test = ["/content/drive/MyDrive/DATA/test_audio/CoChin-ThanhNgoanKhacTu-CHAU VAN.mp3",
  "/content/drive/MyDrive/DATA/test_audio/CauBeDoiNgang-VanChuong-CHAU VAN.mp3",
  "/content/drive/MyDrive/DATA/test_audio/ThinhMauVaQuanDeNhat-CHAU VAN.mp3",
  "/content/drive/MyDrive/DATA/test_audio/BaChuaThac-ChauVan-ThanhNgoan-CHAU VAN.mp3",
  "/content/drive/MyDrive/DATA/test_audio/CoSau-VanChuong-CHAU VAN.mp3"]

● predict_new(chauvan_test, best_model1, "/content/drive/MyDrive/DATA/test_images") # Bad
([4, 4, 4, 4, 4], ['hatxam', 'hatxam', 'hatxam', 'hatxam', 'hatxam'])

[88] predict_new(chauvan_test, best_model2, "/content/drive/MyDrive/DATA/test_images") # Not good
([0, 2, 1, 4, 2], ['cailuong', 'chauvan', 'catru', 'hatxam', 'chauvan'])

✓ 35 giây [89] predict_new(chauvan_test, best_model3, "/content/drive/MyDrive/DATA/test_images") # Quite good
([4, 2, 1, 2, 2], ['hatxam', 'chauvan', 'catru', 'chauvan', 'chauvan'])
```

Như kết quả trên hình, ở cùng 1 tập dữ liệu mới thuộc lớp "Chầu Văn", lần lượt từng model 1, model 2, model 3 cho ra kết quả dự đoán 0/5, 2/5, 3/5. Vì vậy, chúng tôi đã đề xuất và sử dụng phương pháp PROD để kết quả dự đoán của cả 3 model với mong muốn sẽ cho ra kết quả tốt hơn.

E. Inference

Với lý do đã trình bày, chúng tôi áp dụng phương pháp hợp nhất PROD được đề cập trong phần 2 để kết hợp các vectơ xác suất dự đoán từ ba mô hình. Trong giai đoạn suy luận, hệ thống nhận đầu vào là một tệp âm thanh và thực hiện quy trình: xử lý dữ liệu, trích xuất đặc trưng và giai đoạn suy luận. Ba mô hình mạng thần kinh được đặt để huấn luyện trên dữ liệu âm thanh có thời lượng 30 giây. Do đó, khi nhận được âm thanh có độ dài lớn hơn 30 giây, âm thanh này sẽ được chia thành các đoạn mẫu có độ dài 30 giây bằng nhau, sau đó chúng sẽ được đưa vào ba bộ phân loại. Nhãn của mỗi đoạn âm thanh 30 giây được xác định bằng cách áp dụng phương pháp PROD. Nhãn cuối cùng của âm thanh đó được xác định bằng cách sử dụng nguyên tắc bỏ phiếu giữa các nhãn của các đoạn mẫu đã dự đoán.

	Model 1 Pred	Model 2 Pred	Model 3 Pred	Prod Fusion	
$\frac{1}{3} \times$	0.1 0.3 0.2 0.1 0.3	0.3 0.2 0.2 0.1 0.2	0.1 0.3 0.4 0.1 0.1	0.001 0.006 0.0053333 0.0003333 0.002	$= \text{argmax}() = 1 \Rightarrow \text{Ca Trù}$

$P_{prod} = [p_1, p_2, p_c]$ where $P_i = \frac{1}{S} \prod_{s=1}^S P_{si}$ for $1 \leq i \leq C$ — $\text{argmax}(P_{prod}) = \text{argmax}(p_1, p_2, \dots, p_c)$.

E. Inference

Như vậy sau khi kết hợp cả 3 vector xác suất đầu ra của 3 model bằng việc sử dụng PROD fusion. kết quả dự đoán đã cho thấy hiệu quả khi dự đoán trên tập test giả định (như trong hình).

```
chauvan_test = ["/content/drive/MyDrive/DATA/test_audio/CoChin-ThanhNgoanKhacTu-CHAU VAN.mp3",
                 "/content/drive/MyDrive/DATA/test_audio/CauBeDoiNgang-Van Chuong-CHAU VAN.mp3",
                 "/content/drive/MyDrive/DATA/test_audio/ThinhMauVaQuanDeNhat-CHAU VAN.mp3",
                 "/content/drive/MyDrive/DATA/test_audio/BaChuaThac-ChauVan-ThanhNgoan-CHAU VAN.,
                 "/content/drive/MyDrive/DATA/test_audio/CoSau-Van Chuong-CHAU VAN.mp3"]

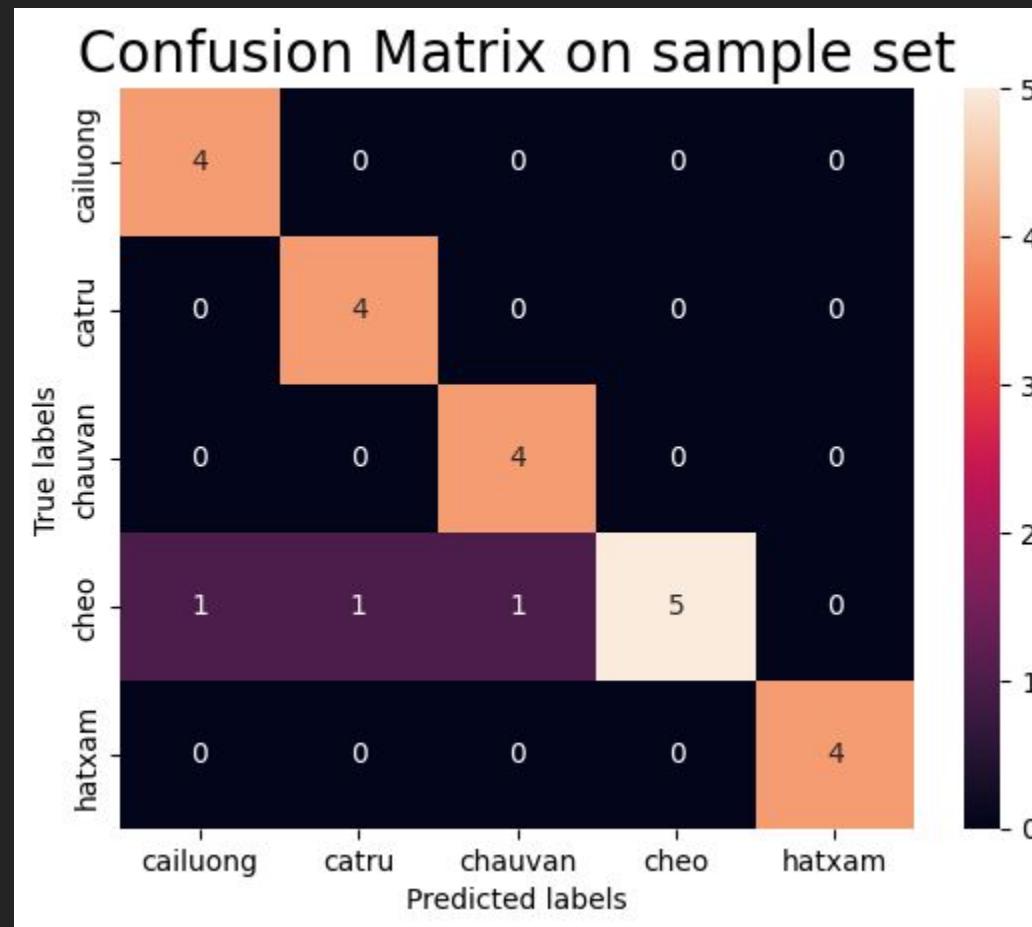
prod_pred, prod_class = PROD_predict(chauvan_test, "/content/drive/MyDrive/DATA/test_images",
                                         best_model1, best_model2, best_model3)

prod_pred, prod_class

([2, 2, 2, 2, 2], ['chauvan', 'chauvan', 'chauvan', 'chauvan', 'chauvan'])
```

E. Inference

Như vậy sau khi kết hợp cả 3 vector xác suất đầu ra của 3 model bằng việc sử dụng PROD fusion. kết quả dự đoán đã cho thấy hiệu quả khi dự đoán trên tập test giả định (như trong hình).



06

Demo chương trình / ứng dụng

06. Demo Ứng Dụng

Tên đề tài: VIETNAMESE TRADITIONAL MUSIC CLASSIFICATION

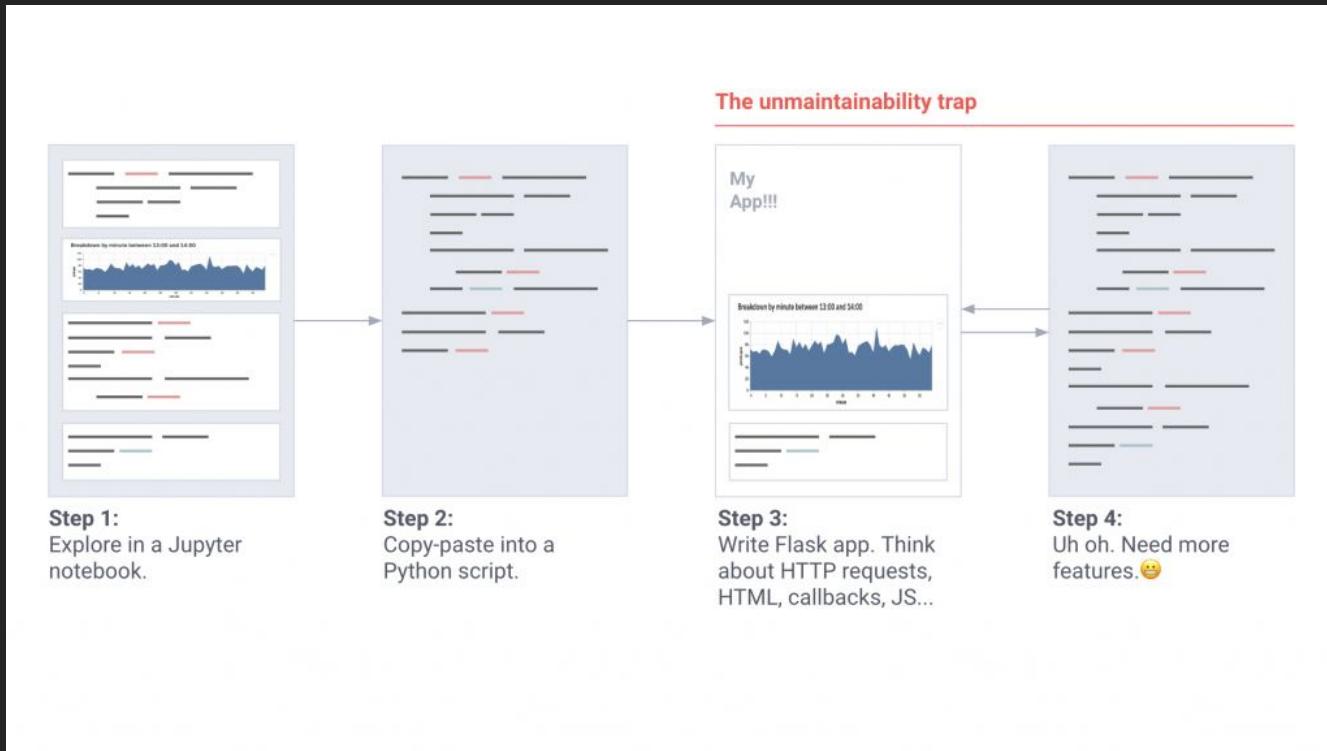
Đặt vấn đề?

- Sau khi đã train cả 3 model CNN trên bộ dataset Vietnam Traditional Music (5 genres) và thu được file pre-trained của cả ba mô hình, thì bây giờ chúng tôi muốn xây 1 web-app demo để người dùng tiếp cận dễ dàng hơn khi muốn xem thử thể loại của bài hát truyền thống đó thuộc vào thể loại nào
- Khi đã có 1 file âm thanh cần dùng để phân loại bài hát, thì người dùng chỉ cần tương tác trực tiếp với web app demo để nhận được kết quả mong muốn, thay vì phải chạy những dòng code khô khan.

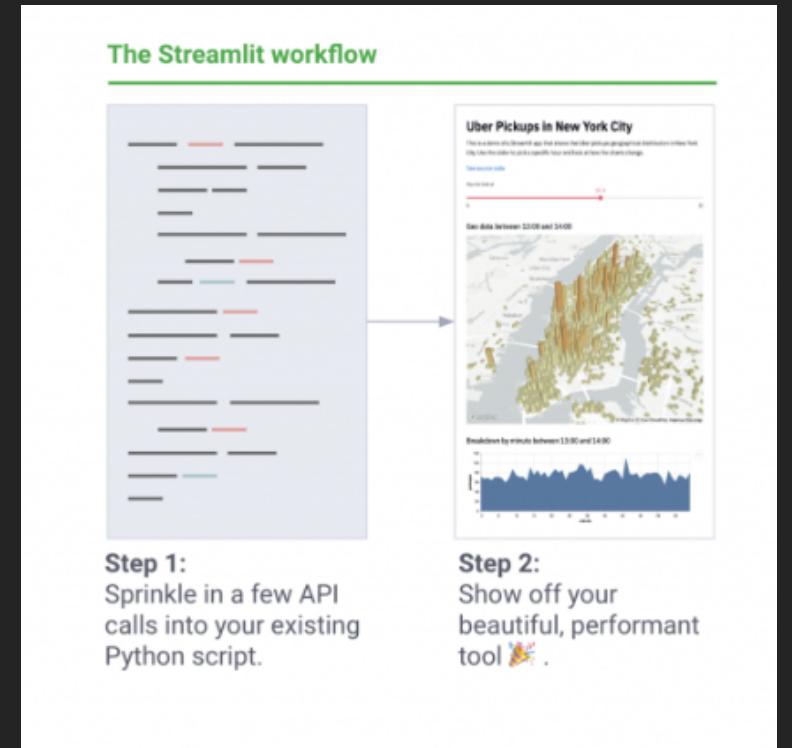


06. Demo Ứng Dụng

Có rất nhiều framework có thể được sử dụng để deploy model lên web như: Streamlit, Flask, Django, ... Tuy nhiên, ta có thể nhìn sơ qua workflows của Flask và Streamlit:



Flask workflow

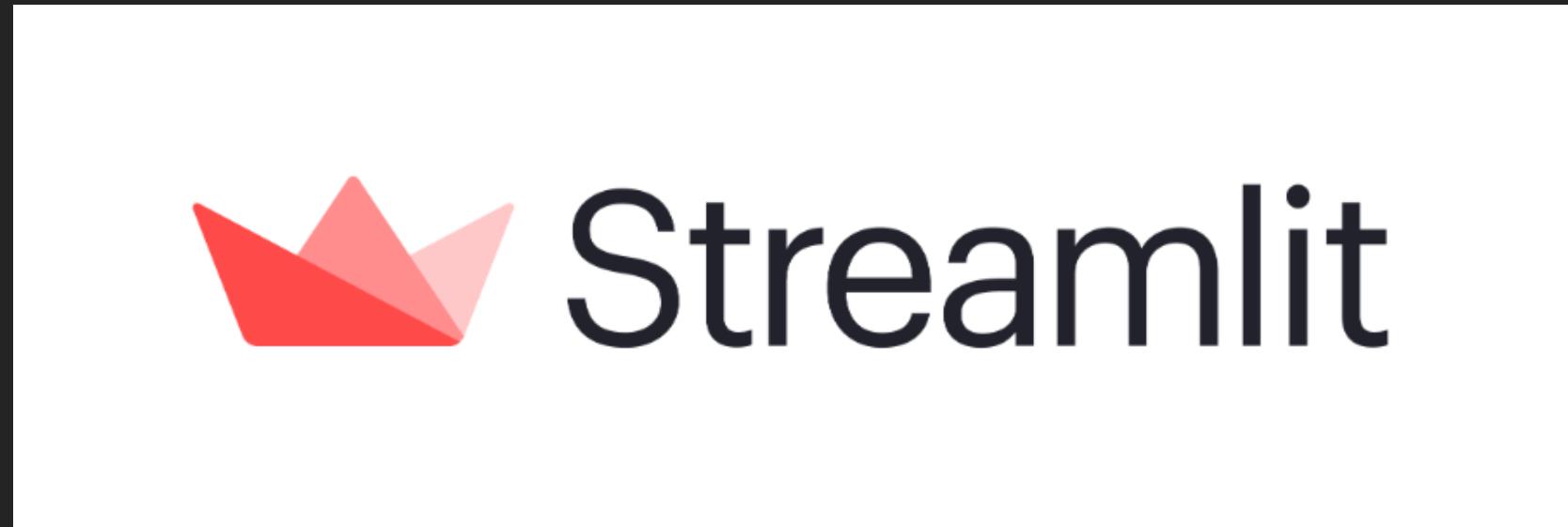


Streamlit workflow

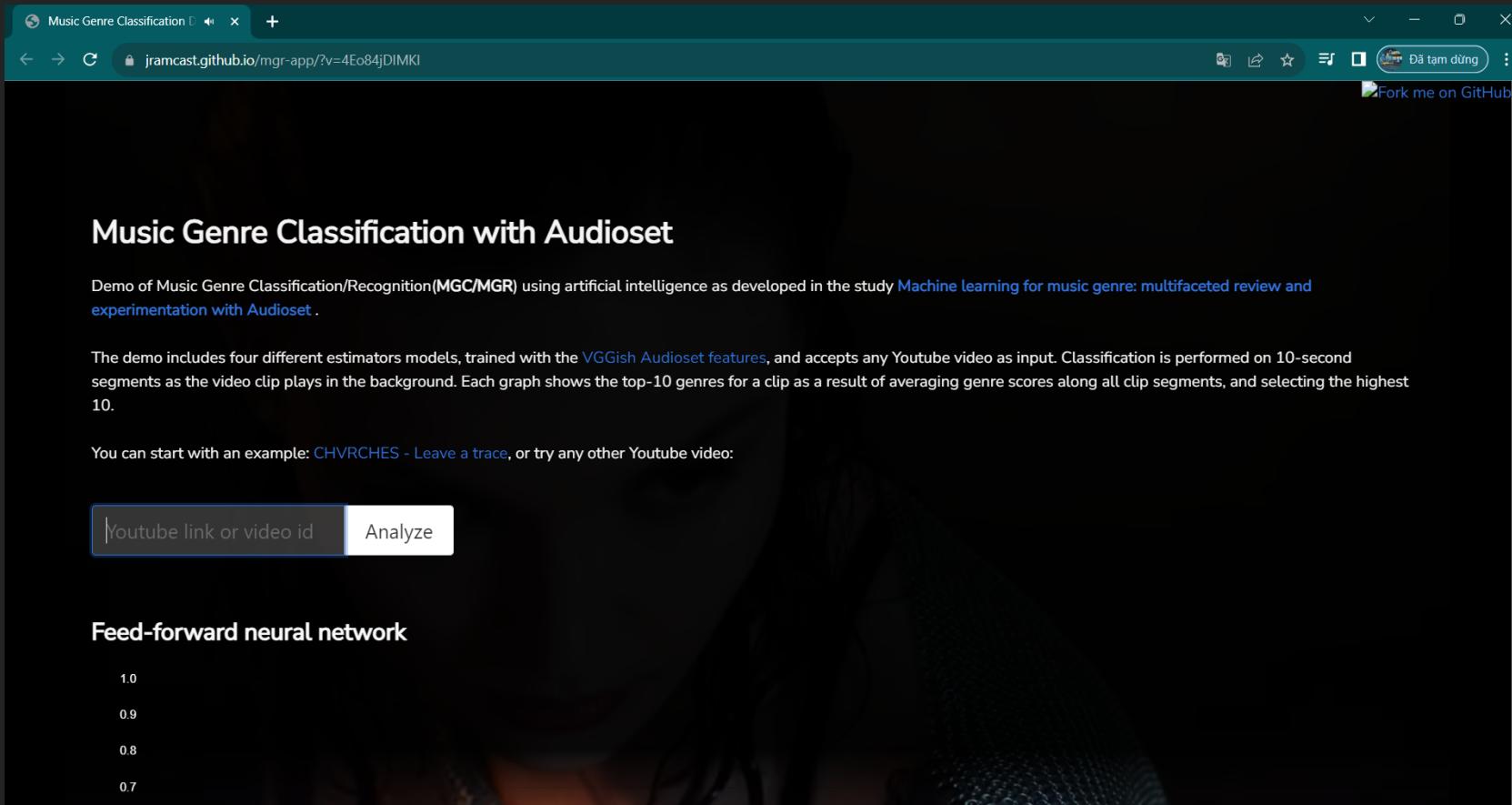
06. Demo Ứng Dụng

Vì vậy, với phạm vi nghiên cứu và quy mô dự án không quá lớn. Chúng tôi quyết định sử dụng framework Streamlit để code demo web app cho tác vụ: Phân loại âm nhạc truyền thống Việt Nam.

Streamlit là một open-source Python lib, nó giúp ta dễ dàng tạo một web app cho Machine Learning. Ưu điểm của Streamlit là Build & Deploy nhanh.

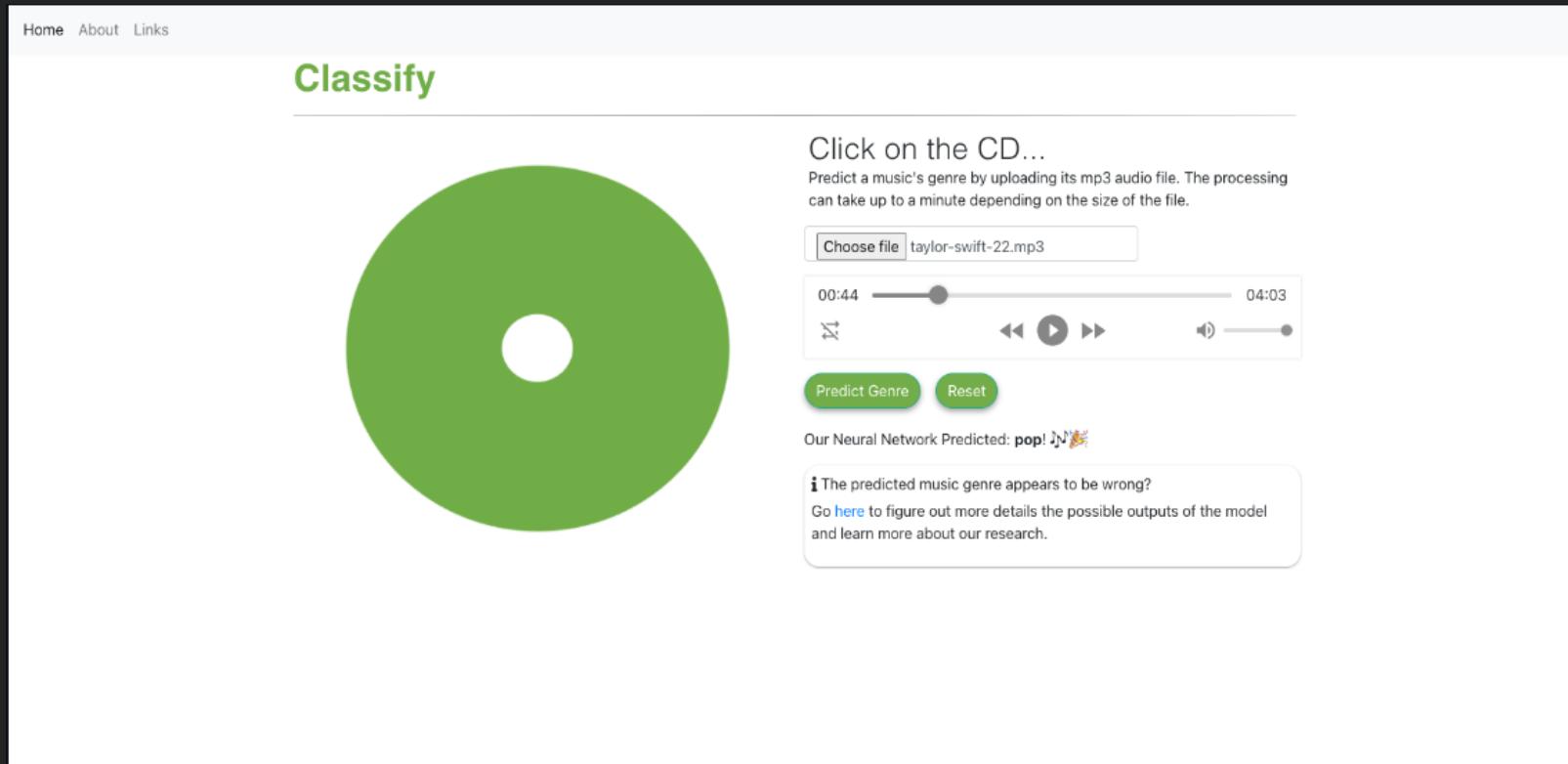


06. Demo Ứng Dụng - Phân tích khảo sát



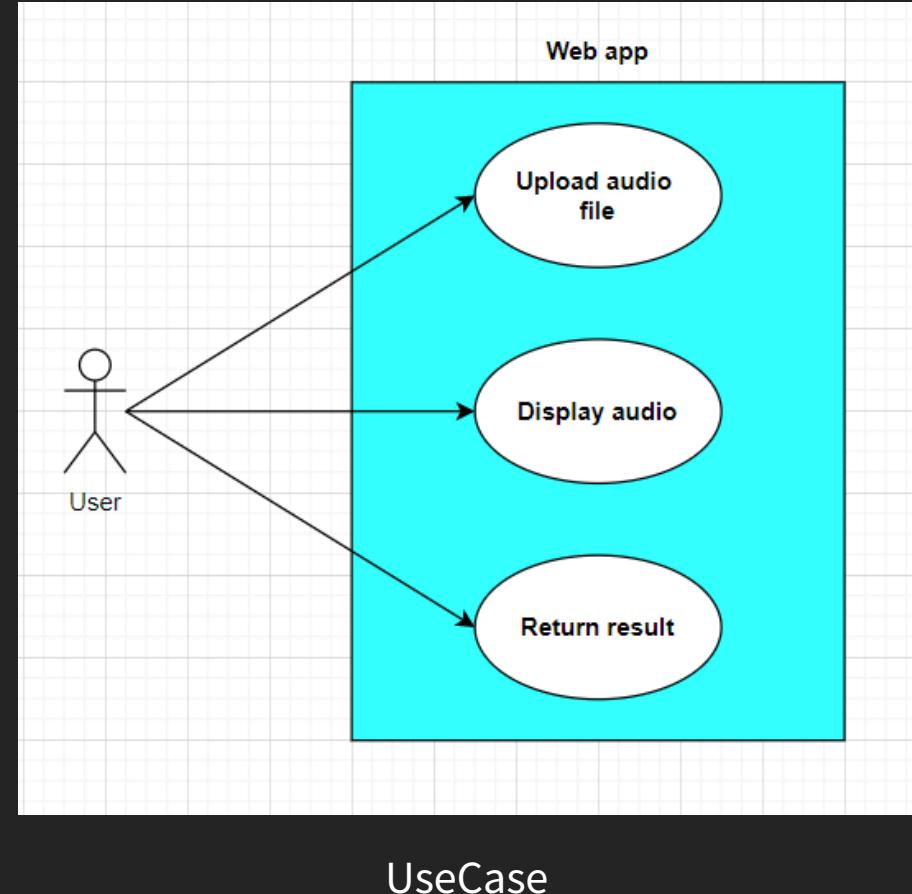
<https://jramcast.github.io/mgr-app/?v=4Eo84jDIMKI> - Phân loại âm nhạc với link Youtube

06. Demo Ứng Dụng - Phân tích khảo sát

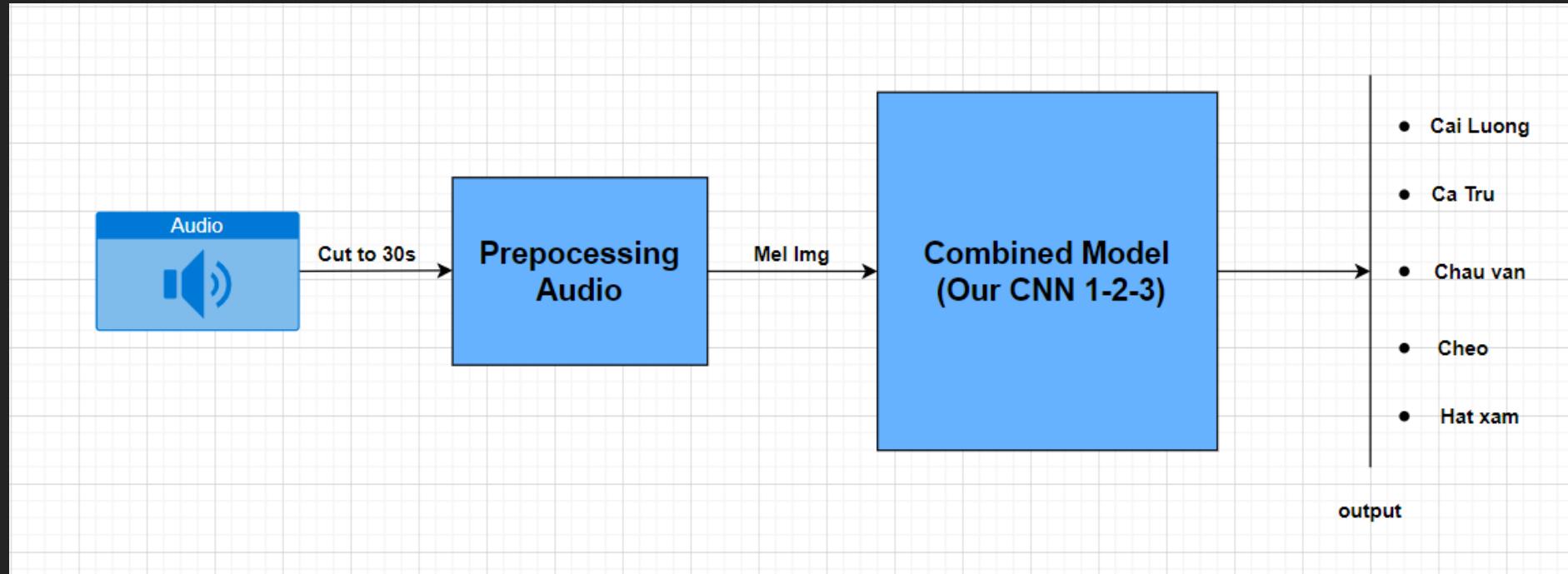


<https://classify.k8s.pouretadev.com/> - Phân loại thể loại âm nhạc bằng upload file

06. Demo Ứng Dụng - Rút ra yêu cầu người dùng



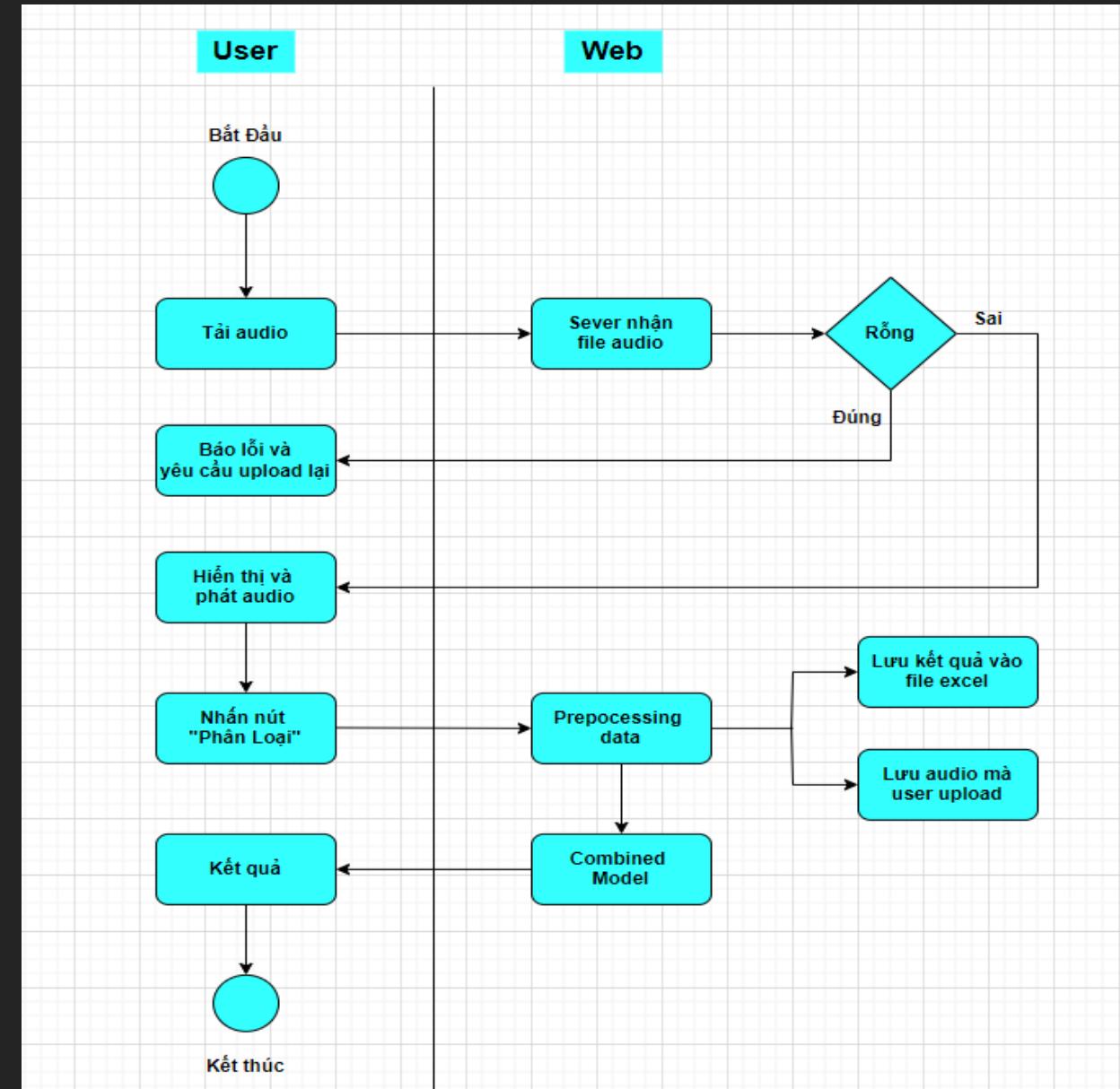
06. Demo Ứng Dụng - Pipeline



Pipeline

06. Demo Ứng Dụng

- Workflow các bước xử của Web app demo có thể được thể hiện rõ qua Activity Diagramme như hình bên
- Ưu điểm của kiến workflow như hình bên:
 - Dễ dàng tiếp cận
 - Trình tự xử lý rõ ràng
 - Không phụ thuộc môi trường, nền tảng xây dựng
 - Chức năng riêng biệt, dễ dàng quản lý, cải tiến



06. Demo Ứng Dụng - Thiết kế giao diện

Vietnamese Traditional Music Classifier



Cải lương



Ca trù



Chầu văn



Chèo



Hát xẩm



Upload your audio below. (.wav, .mp3)

Kéo thả hoặc upload file từ thiết bị của user

Hiển thị tên những file audio đã upload thành công

Drag and drop files here
Limit 200MB per file • MP3, WAV

CoChin-ThanhNgoanKhacTu-CHAU VAN.mp3 9.4MB

y2mate.com - Ca Trù Tây Hồ hoài cổ.mp3 6.0MB

DaoLieu-VanChuongNSUT-CHEO.mp3 2.8MB

Xóa file đã upload

06. Demo Ứng Dụng - Thiết kế giao diện

The screenshot shows a user interface for a music classification application. At the top, there is a list of uploaded files:

- CoChin-ThanhNgoanKhacTu-CHAU VAN.mp3 9.4MB
- y2mate.com - Ca Trù Tây Hồ hoài cổ.mp3 6.0MB
- DaoLieu-VanChuongNSUT-CHEO.mp3 2.8MB

Below this list, three audio files are shown in a grid format:

- File uploaded 0: DaoLieu-VanChuongNSUT-CHEO.mp3
Duration: 0:00 / 3:05
- File uploaded 1: y2mate.com - Ca Trù Tây Hồ hoài cổ.mp3
Duration: 0:00 / 6:31
- File uploaded 2: CoChin-ThanhNgoanKhacTu-CHAU VAN.mp3
Duration: 0:00 / 10:16

A yellow callout points from the text "Hiển thị audio về thời lượng và cho phép phát nhạc để nghe thử file audio" to the duration and play controls of the first file.

An orange callout points from the text "Trả về kết quả dựa đoán ứng với từng file" to the predicted results for each file.

A blue callout points from the text "Nhấn nút để 'Phân Loại'" to the "Classify" button.

The predicted results are displayed in green boxes:

- File uploaded 0: DaoLieu-VanChuongNSUT-CHEO.mp3 --> PREDICT: Chèo
- File uploaded 1: y2mate.com - Ca Trù Tây Hồ hoài cổ.mp3 --> PREDICT: Ca trù
- File uploaded 2: CoChin-ThanhNgoanKhacTu-CHAU VAN.mp3 --> PREDICT: Chầu văn

06. Demo Ứng Dụng

The screenshot shows a code editor interface with several windows open. On the left, the file explorer displays a project structure under 'APP'. A green box highlights the 'audio_from_user' folder, which contains various audio files (mp3 and wav formats) related to different songs and artists. A vertical green line points from this folder to a text annotation at the bottom left: 'Hệ thống lưu lại các audio khác nhập mà user đã upload' (The system saves other audio files uploaded by the user).

In the center, a spreadsheet window titled 'user_input.xlsx' is shown. An orange box highlights the file tab. The spreadsheet has three columns: A, B, and C. Column A contains row numbers from 7 to 34. Column B lists various audio file names, such as 'Haxam-NhoInay-HaThiCau-HATXAM.mp3', 'QuanDeNhat-VanChuong-2815994.mp3', etc. Column C lists corresponding categories like 'Hát xẩm', 'Chầu văn', 'Chèo', etc. An orange arrow points from the 'user_input.xlsx' tab to another text annotation on the right: 'Hệ thống ghi lại lịch sử sử dụng của user bao gồm: STT, Tên bài hát, Thể loại' (The system records the user's usage history including: Line number, Song title, and Category).

A	B	C
7	Haxam-NhoInay-HaThiCau-HATXAM.mp3	Hát xẩm
8	QuanDeNhat-VanChuong-2815994.mp3	Chầu văn
9	NonThungQuaiThao-CHEO.mp3	Chèo
10	ThinhMauVaQuanDeNhat-CHAU VAN.mp3	Chầu văn
11	TuongTienTuu-QuachThiHo-CATRU.wav	Ca trù
12	TinhyeuVagiotnuocmat-NguyenKha-CAILUONG.mp3	Cải lương
13	ThinhMauVaQuanDeNhat-CHAU VAN.mp3	Chầu văn
14	TuongTienTuu-QuachThiHo-CATRU.wav	Ca trù
15	CauBeDoiNgang-VanChuong-CHAU VAN.mp3	Chèo
16	CauBeDoiNgang-VanChuong-CHAU VAN.mp3	Chèo
17	NonThungQuaiThao-CHEO.mp3	Chèo
18	Ioithenonnuc_catru.mp3	Ca trù
19	TinhThuHaVi-QuocPhong-CHEO.mp3	Chèo
20	TinhyeuVagiotnuocmat-NguyenKha-CAILUONG.mp3	Cải lương
21	LenhTruyNa-VuongLinh-CAILUONG.mp3	Cải lương
22	VanChauLuc-VanChung_CHAU VAN.mp3	Chầu văn
23	BaChuaThac-ChauVan-ThanhNgoan-CHAU VAN.mp3	Chầu văn
24	y2mate.com - Ca Trù Tây Hồ hoài cổ.mp3	Ca trù
25	y2mate.com - Ca Trù Tây Hồ hoài cổ.mp3	Ca trù
26	y2mate.com - Ca Trù Tây Hồ hoài cổ.mp3	Ca trù
27	CoChin-ThanhNgoanKhacTu-CHAU VAN.mp3	Chầu văn
28	y2mate.com - Ca Trù Tây Hồ hoài cổ.mp3	Ca trù
29	y2mate.com - Ca Trù Tây Hồ hoài cổ.mp3	Ca trù
30	y2mate.com - Ca Trù Tây Hồ hoài cổ.mp3	Ca trù
31	CoChin-ThanhNgoanKhacTu-CHAU VAN.mp3	Chầu văn
32	DaoLieu-VanChuongNSUT-CHEO.mp3	Chèo
33	y2mate.com - Ca Trù Tây Hồ hoài cổ.mp3	Ca trù
34	CoChin-ThanhNgoanKhacTu-CHAU VAN.mp3	Chầu văn

File Explorer:

- APP
 - pycache
 - audio_from_user
 - CoChin-ThanhNgoanKhacTu-CHAU VAN.mp3
 - CoChin-ThanhNgoanKhacTu-CHAU VAN.wav
 - Daolieu-VanChuongNSUT-CHEO.mp3
 - Daolieu-VanChuongNSUT-CHEO.wav
 - y2mate.com - Ca Trù Tây Hồ hoài cổ.mp3
 - y2mate.com - Ca Trù Tây Hồ hoài cổ.wav
 - checkpoint
 - ffmpeg
 - mel-images
 - app_helpers.py
 - app.py
 - background.jpg
 - blur.jpg
 - cailuong.jpg
 - catru.jpg
 - chauvan.jpg
 - cheo.jpg
 - config.py
 - hatxam.jpg
 - screen1.png
 - screen2.png
 - user_input.xlsx
 - utils.py
- OUTLINE
- TIMELINE
- OPENXML EXPLORER

Bottom status bar:

- master
- 0 △ 1
- 0 0
- Live Share
- 1 file to analyze
- Reconnect to Discord
- Go Live

07

Tóm tắt kết quả đạt được

07

Tóm tắt kết quả đạt được

07. Kết quả đạt được

Ở bài báo cáo này, chúng tôi đã tìm hiểu và trình bày một quy trình cụ thể cho bài toán Phân loại âm thanh và kết quả của nhiều mô hình CNN dựa trên bài toán này. Chúng tôi đã trình bày phương pháp cho Vietnamese Traditional Classification (5 thể loại) và đạt được thành tích tốt về độ chính xác (đặc biệt là với kiến trúc PROD - combined model).

Trong quá trình tiến hành thực nghiệm, mô hình 3 tốt hơn mô hình 2 và mô hình 1 ở hầu hết các số liệu đánh giá. Accuracy, precision, recall đạt được trên tập test là 0.91, 0.91, 0.91.

Model	Test set			Accuracy
	Precision	Recall	F1-score	
model 1	0.89	0.89	0.89	0.89
model 2	0.84	0.86	0.85	0.84
model 3	0.91	0.91	0.91	0.91

07. Kết quả đạt được

Để kiểm chứng kết quả dự đoán của Combined Model, tụi em tiến hành xây dựng 1 bộ dữ liệu nhỏ gồm 50 samples (10 samples mỗi thể loại) và sau đó tiến hành kiểm tra trực tiếp bằng việc tương tác với web demo.

29	BaChuaThac-ChauVan-ThanhNgoan-CHAU VAN.w	Chầu văn
30	cheo(1).mp3	Chèo
31	cheo(10).mp3	Chèo
32	cheo(2).mp3	Chèo
33	cheo(4).mp3	Chèo
34	cheo(11).mp3	Chèo
35	cheo(3).mp3	Chèo
36	cheo(8).mp3	Chèo
37	cheo(5).mp3	Chèo
38	cheo(9).mp3	Chèo
39	cheo(6).mp3	Chèo
40	y2mate.com - Xẩm Thập Ân Nghệ Nhân Hà Thị C	Hát xẩm
41	hatxam_hathicau_2.mp3	Hát xẩm
42	hatxam_hathicau_3.mp3	Hát xẩm
43	y2mate.com - Giải nhất hát xẩm Mục hạ vô nhân	Ca trù
44	hatxam_hathicau_1.mp3	Hát xẩm
45	hatxam_hathicau_4.mp3	Hát xẩm
46	hatxam_hathicau_5.mp3	Hát xẩm
47	hatxam_hathicau_6.mp3	Hát xẩm
48	hatxam_xuanhoach_2.mp3	Ca trù
49	hatxam_xuanhoach_1.mp3	Ca trù

	Filename	Genre
0	cailuong(8).mp3	Cải lương
1	cailuong(10).mp3	Cải lương
2	cailuong(2).mp3	Cải lương
3	cailuong(3).mp3	Cải lương
4	cailuong(9).mp3	Cải lương
5	cailuong(1).mp3	Cải lương
6	cailuong(6).mp3	Cải lương
7	cailuong(4).mp3	Cải lương
8	cailuong(7).mp3	Cải lương
9	cailuong(5).mp3	Cải lương
10	catru(1).mp3	Ca trù
11	catru(2).mp3	Ca trù
12	catru(5).mp3	Ca trù
13	catru(4).mp3	Ca trù
14	catru(3).mp3	Ca trù
15	catru(10).mp3	Ca trù
16	catru(6).mp3	Ca trù
17	catru(8).mp3	Ca trù
18	catru(7).mp3	Ca trù
19	catru(9).mp3	Ca trù
20	y2mate.com - Hát Văn Tình Cha NSƯT Đình Cươi	Chèo
21	CoSau-VanChuong-CHAU VAN.mp3	Chầu văn
22	CoChin-ThanhNgoanKhacTu-CHAU VAN (1).mp3	Chầu văn
23	y2mate.com - Vu lan báo hiếu hoài thanh dâng v	Ca trù
24	y2mate.com - Văn Hát Lễ Ông Hoàng Bảy.mp3	Ca trù
25	VanChauLuc-VanChung_CHAU VAN.mp3	Chầu văn
26	y2mate.com - Bạn sẽ phải tiếc nuối nếu không ng	Chầu văn
27	ThinhMauVaQuanDeNhat-CHAU VAN.mp3	Chầu văn
28	y2mate.com - Chầu Bé Bắc Lê Bản Đặc Biệt Hoài Chầu văn	

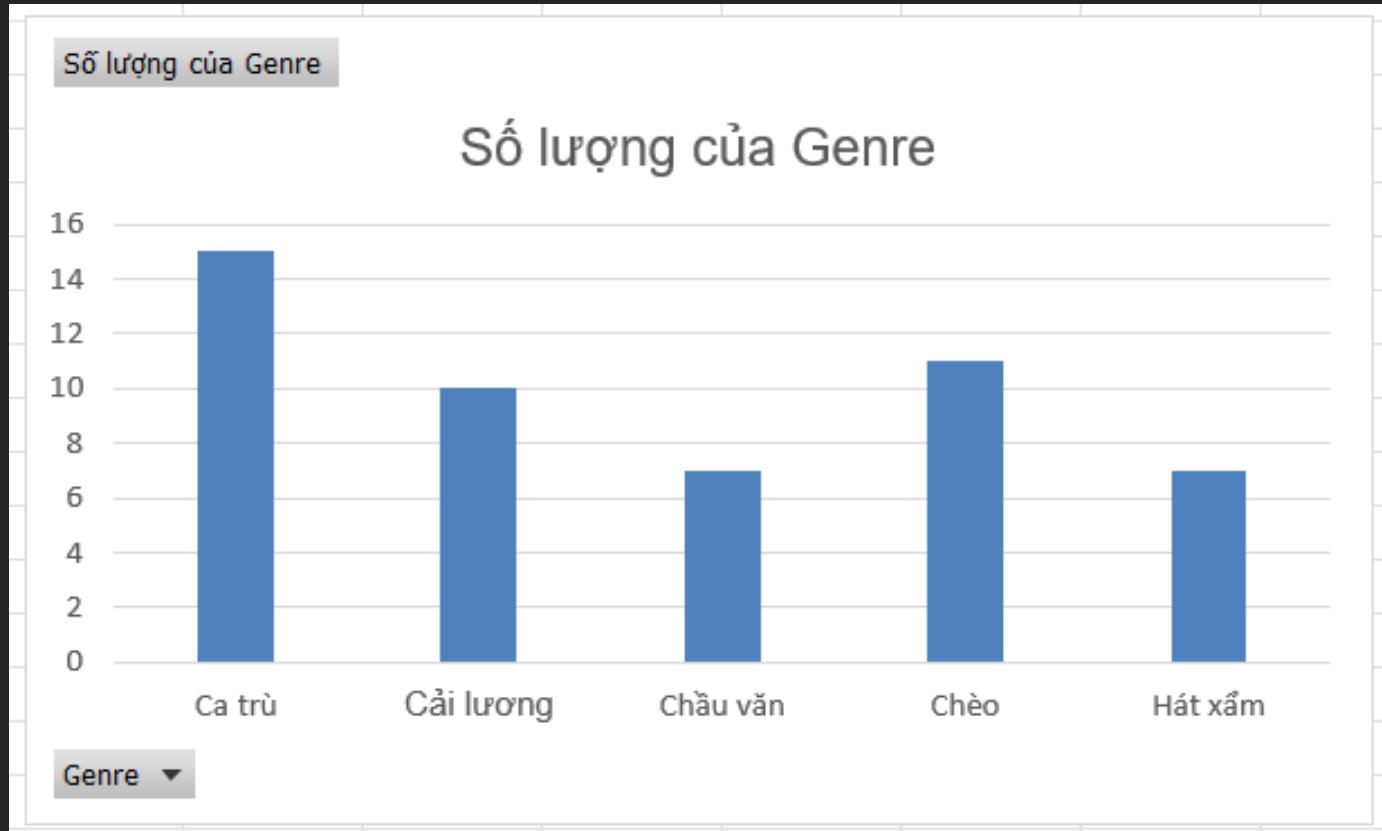
07. Kết quả đạt được

Kết quả thu được sau khi thực nghiệm trên 50 audio tự thu thập như sau:

Theo như quan sát kết quả dự đoán: 44/50

- Cải lương: 10/10
- Ca trù: 10/10
- Chầu văn: 7/10
 - 2 bài bị dự đoán thành: Ca trù
 - 1 bài bị dự đoán thành: Chèo
- Hát xẩm: 7/10
 - Có 3 bài bị dự đoán thành: Ca trù
- Chèo: 10/10

Genre	Số lượng của Genre
Ca trù	15
Cải lương	10
Chầu văn	7
Chèo	11
Hát xẩm	7



07. Kết quả đạt được

- Từ kết quả 44/50 từ dữ liệu chúng tôi tự thu thập: Chúng tôi nhận thấy model của chúng tôi dự đoán với tỷ lệ chính xác khá cao đối với bất kỳ bài hát nào thuộc 5 thể loại mà model đã được học
- Demo Web của chúng tôi được thiết kế với giao diện khá dễ nhìn, dễ thao tác và sử dụng. Tốc độ dự đoán cũng khá nhanh và có thể dự đoán một lúc không giới hạn về số lượng bài hát. Đồng thời Web demo cũng cho người dùng nghe thử bài nhạc mà họ upload lên.
- Về bản thân chúng tôi, chúng tôi đã được tìm hiểu và học được những điều như sau:
 - Pipeline cho nhiệm vụ Audio Classification
 - Dự đoán dựa trên kết hợp nhiều vector xác suất đầu ra của cả 3 mô hình CNN để gia tăng độ chính xác của kết quả dự đoán
 - Kinh nghiệm thiết kế 1 Web Demo đơn giản để demo model và tiến gần hơn với các ứng dụng ML thực tế.



08

Hạn chế và các điểm cần cài tiến

08.01 Hạn chế và các điểm cần cải thiện

1. Mô hình chưa hoàn thiện do chưa thể phân biệt được các đặc điểm phức tạp để phân biệt các thể loại nhạc (chầu văn và hát xẩm bị nhầm lẫn với ca trù)
2. Dữ liệu dùng để huấn luyện bị hạn chế do tính chất của các loại hình âm nhạc cổ truyền không thông dụng. Do đó, việc phát triển mô hình sẽ gặp khó khăn (nếu áp dụng các mạng neuron sâu hơn sẽ yêu cầu lượng dữ liệu lớn hơn để huấn luyện)
3. Các thể loại nhạc có sự tương đồng một phần trong việc sử dụng nhạc cụ, giai điệu (phần lớn với các thể loại có xuất xứ từ miền bắc).
4. Ngoài các yếu tố về thanh nhạc, các loại hình nghệ thuật còn có sự khác biệt về văn hoá (nội dung bài hát) và cách thức biểu diễn (ví dụ: hát xẩm thường được biểu diễn ngoài trời như các sự kiện cộng đồng, ca trù là thể loại nhạc truyền thống của đô thị thường được biểu diễn ở những khán phòng, hội quán).

08.02 Phương hướng cải tiến

1. Mở rộng dataset .
2. Có thể sử dụng CNN pretrained deep learning model để fine-tune cho dataset của đồ án .
3. Áp dụng speech-to-text model để phân tích nội dung bài hát, từ đó có thể cho ra hiệu suất tốt hơn.
4. Mở rộng thêm về số lượng thể loại.

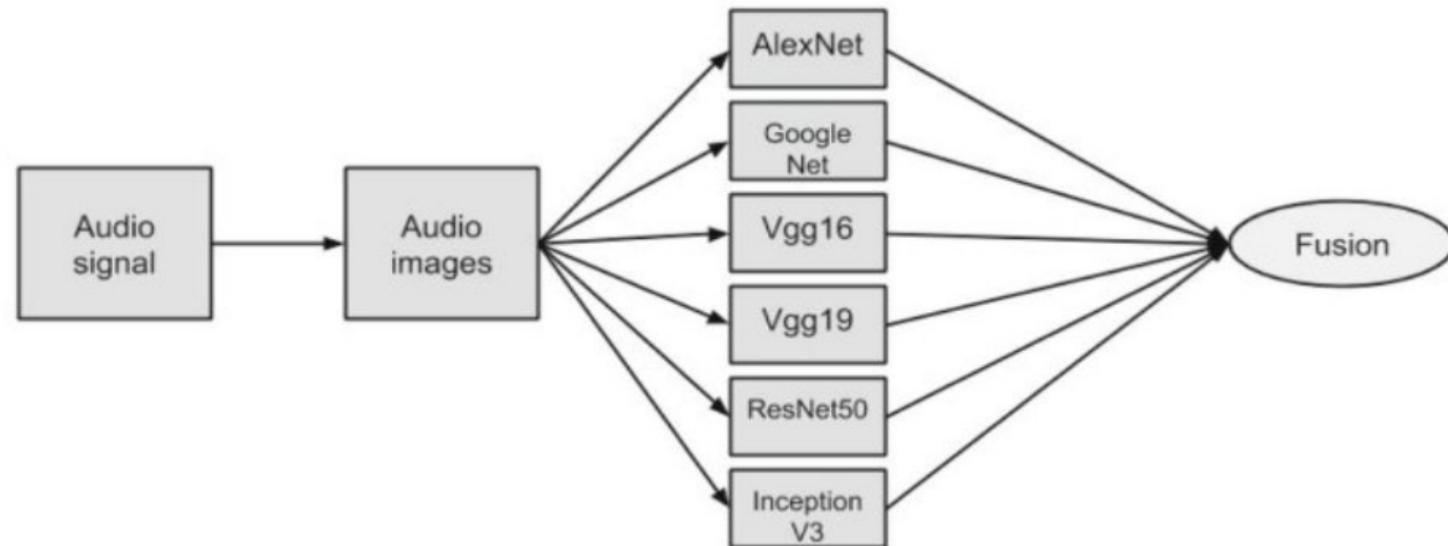


Fig. 5 Fusion of the sets of CNNs

Thanks for listening!





≡ CODE