

**ĐẠI HỌC QUỐC GIA TP.HỒ CHÍ MINH**  
**TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN**



**TRƯỜNG VĂN KHẢI – 21520274**

**HOÀNG TIẾN ĐẠT – 21520696**

**PHÂN LOẠI THỂ LOẠI ÂM NHẠC TRUYỀN  
THỐNG VIỆT NAM**  
**VIETNAMESE TRADITIONAL MUSIC  
CLASSIFICATION**

**ĐỒ ÁN CUỐI KỲ**

**TP.HỒ CHÍ MINH – 2023**

**ĐẠI HỌC QUỐC GIA TP.HỒ CHÍ MINH**  
**TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN**



**TRƯỜNG VĂN KHẢI – 21520274**

**HOÀNG TIẾN ĐẠT – 21520696**

**PHÂN LOẠI THỂ LOẠI ÂM NHẠC TRUYỀN  
THỐNG VIỆT NAM**  
**VIETNAMESE TRADITIONAL MUSIC  
CLASSIFICATION**

Ngành: Khoa Học Máy Tính

Chuyên ngành: Trí Tuệ Nhân Tạo

**ĐỒ ÁN CUỐI KỲ**

**TP.HỒ CHÍ MINH – 2023**

## **LỜI CAM ĐOAN**

Chúng tôi xin cam đoan nội dung được trình bày trong báo bài cáo “Phân loại thể loại âm nhạc truyền thống Việt Nam” là do chúng tôi nghiên cứu, tìm hiểu và phát triển dưới sự dẫn dắt của ThS. Trịnh Quốc Sơn. Bài báo cáo không sao chép từ các tài liệu, công trình nghiên cứu của người khác mà không ghi rõ trong tài liệu tham khảo. Tất cả những tham khảo từ các nghiên cứu liên quan đều được nêu nguồn gốc một cách rõ ràng từ danh mục tài liệu tham khảo trong đồ án. Chúng tôi xin chịu trách nhiệm về lời cam đoan này.

Hồ Chí Minh, ngày tháng năm 2023

## **LỜI CẢM ƠN**

Chúng tôi xin chân thành cảm ơn thầy giáo, ThS. Trịnh Quốc Sơn, người đã định hướng, giúp đỡ, trực tiếp hướng dẫn và tận tình chỉ bảo chúng tôi trong suốt quá trình nghiên cứu, xây dựng và hoàn thiện bài báo cáo này.

Chúng tôi cũng xin được cảm ơn tới gia đình, những người thân, bạn bè thường xuyên quan tâm, động viên, chia sẻ kinh nghiệm, cung cấp các tài liệu hữu ích trong thời gian học tập, nghiên cứu cũng như trong suốt quá trình thực đồ án cuối kỳ.

Hồ Chí Minh, ngày tháng năm 2023

## MỤC LỤC

LỜI CAM ĐOAN.....	1
LỜI CẢM ƠN .....	2
MỤC LỤC.....	3
<b>DANH MỤC HÌNH VẼ VÀ ĐỒ THỊ .....</b>	<b>6</b>
<b>MỞ ĐẦU.....</b>	<b>8</b>
1. Động lực nghiên cứu .....	8
2. Mục tiêu đồ án.....	8
3. Cấu trúc đồ án .....	9
<b>CHƯƠNG 1: TỔNG QUAN BÀI TOÁN “PHÂN LOẠI THỂ LOẠI ÂM NHẠC TRUYỀN THỐNG VIỆT NAM” .....</b>	<b>10</b>
1.1. Phát biểu bài toán.....	10
1.2. Kỹ thuật rút trích đặc trưng âm thanh .....	11
1.2.1. Chi tiết kỹ thuật tiền xử lý dữ liệu âm thanh .....	12
1.2.2. Đánh giá mô hình phân lớp.....	16
1.2.3. Mạng nơ-ron tích chập - CNN .....	18
<b>CHƯƠNG 2: XÂY DỰNG MÔ HÌNH, THỰC NGHIỆM VÀ ĐÁNH GIÁ.....</b>	<b>23</b>
2.1. Khảo sát dữ liệu .....	23
2.2. Bộ dữ liệu.....	24
2.3. Các thí nghiệm .....	25
2.2.1. Thí nghiệm 1: Tiến hành thực nghiệm trên mô hình CNN tự xây dựng thứ 1 .....	25
2.2.2. Thí nghiệm 2: Tiến hành thực nghiệm trên mô hình CNN tự xây dựng thứ 2 .....	28
2.2.3. Thí nghiệm 3: Tiến hành thực nghiệm trên mô hình CNN tự xây dựng thứ 3 .....	31
2.2.4. Thí nghiệm 4: Tiến hành thực nghiệm trên mô hình kết hợp .....	34
<b>CHƯƠNG 3: XÂY DỰNG HỆ THỐNG PHÂN LOẠI ÂM NHẠC.....</b>	<b>38</b>
3.1. StreamLit.....	38
3.2. Phân tích thiết kế hệ thống.....	39
3.2.1. Phân tích khảo sát .....	39
3.2.2. Thiết kế giao diện.....	42
3.3. Kiểm thử hệ thống.....	44
3.3.1. Môi trường thực nghiệm .....	44
3.3.2. Kết quả thực nghiệm .....	44
<b>KẾT LUẬN .....</b>	<b>46</b>

TÀI LIỆU THAM KHẢO .....	49
--------------------------	----

## DANH MỤC KÝ HIỆU VÀ CÁC CHỮ VIẾT TẮT

Từ viết tắt	Từ chuẩn	Diễn giải
STFT	Short-time Fourier Transform	Phép biến đổi một hàm số hoặc một tín hiệu theo miền thời gian sang miền tần số
CNN	Convolution Neural Network	Mạng Nơron tích chập
TFR	Time Frequency Representation	Tần số đại diện thời gian
ANN	Artificial Neuron Network	Mạng nơ-ron nhân tạo
FC	Fully Connected	Lớp kết nối toàn phần

## DANH MỤC HÌNH VẼ VÀ ĐỒ THỊ

Hình 1.1 Ảnh minh họa Phân loại âm nhạc .....	11
Hình 1.2 Các bước tiền xử lý dữ liệu Audio .....	12
Hình 1.3 Định dạng âm thanh được lưu trong máy tính.....	12
Hình 1.4 Biểu đồ Audio Signal cho 1 file âm thanh .wav dài 2.5s .....	13
Hình 1.5 Công thức biến đổi STFT. ....	13
Hình 1.6 Biểu đồ biến đổi STFT .....	14
Hình 1.7 Spectrogram biểu diễn cho một file .wav trong bộ dataset .....	15
Hình 1.8 Công thức chuyển đổi từ thang đo Hz sang Mel .....	15
Hình 1.9 Mel - Spectrogram cho một file .wav trong bộ dataset. ....	16
Hình 1.10 Ví dụ về confusion matrix .....	17
Hình 1.11 Cấu tạo của một mạng nơ-ron chập tích cơ bản .....	18
Hình 1.12 Các lớp phổ biến của mô hình CNN.....	19
Hình 1.13 Hình ảnh minh họa đầu vào cuat lớp tích chập .....	20
Hình 1.14 Kỹ thuật zero padding.....	20
Hình 1.15 Các hàm kích hoạt phổ biến. ....	21
Hình 1.16 Hai hàm Pooling thông dụng. ....	21
Hình 2.1 Bộ dataset trên kaggle mà nhóm sử dụng.....	24
Hình 2.2 Số lượng bài hát trong từng thể loại .....	24
Hình 2.3 Số lượng bài hát trong từng tập dữ liệu.....	25
Hình 2.4 Tổng quan cấu trúc các lớp mô hình CNN thứ 1.....	26
Hình 2.5 Training history mô hình CNN thứ 1 .....	27
Hình 2.6a Bảng hiệu suất mô hình CNN thứ 1.....	27
Hình 2.6b Bảng Confusion Matrix mô hình CNN thứ 1 .....	28
Hình 2.7 Tổng quan cấu trúc các lớp mô hình CNN thứ 2.....	29
Hình 2.8 Training history mô hình CNN thứ 2 .....	30
Hình 2.9a Bảng hiệu suất mô hình CNN thứ 2.....	30
Hình 2.9b Bảng Confusion matrix của mô hình thứ 2 .....	31
Hình 2.10 Tổng quan cấu trúc các lớp mô hình CNN thứ 3.....	32



Hình 2.11 Training history mô hình CNN thứ 3 .....	33
Hình 2.12a Bảng hiệu suất mô hình CNN thứ 3.....	33
Hình 2.12b Bảng Confusion Matrix của mô hình CNN thứ 3.....	34
Hình 2.13 Kết quả của cả 3 model trên dữ liệu mới.....	35
Hình 2.14 Ví dụ về cách tính toán của mô hình kết hợp .....	36
Hình 2.15 Kết quả mô hình kết hợp trên dữ liệu mới.....	36
Hình 2.16 Confusion matrix của mô hình kết hợp trên bộ dữ liệu mới .....	37
Hình 3.1 So sánh giữa hai framework .....	38
Hình 3.2 Framework StreamLit.....	39
Hình 3.3 Sơ đồ Use-case hệ thống.....	40
Hình 3.4 Pipeline hệ thống .....	40
Hình 3.5 Activity Diagram hệ thống .....	41
Hình 3.6 Màn hình giao diện 1 .....	42
Hình 3.7 Màn hình giao diện 2 .....	43
Hình 3.8 Quản lý hệ thống .....	43
Hình 3.9 Quản lý một số dữ liệu về các thể loại nhạc truyền thống tự thu thập .....	44
Hình 3.10 Bảng kết quả thực nghiệm.....	46

## MỞ ĐẦU

Theo dòng chảy của cuộc cách mạng 4.0, trí tuệ nhân tạo ngày càng được phổ biến và ứng dụng rộng rãi trong mọi lĩnh vực của cuộc sống. Hiện nay, các kho nhạc số Việt Nam đang sắp xếp các bài nhạc theo tên ca sĩ hoặc tên bài hát, trong khi đó người dùng cũng muốn tìm kiếm các bài nhạc theo thể loại và nội dung. Điều này dẫn đến nhu cầu cầu phân loại nhạc theo thể loại trong các kho lưu trữ âm nhạc số để cho phép người dùng tìm các bài hát theo thể loại.

Sự phát triển của tự động hóa thúc đẩy cuộc cách mạng hóa toàn cầu. Hiện nay, phân loại thể loại là một nhiệm vụ quan trọng với nhiều ứng dụng thực tế. Trong khi lượng âm nhạc được phát hành hàng ngày tiếp tục tăng vọt, theo thống kê số liệu thống kê từ Music Business Worldwide<sup>1</sup> cho biết mỗi ngày có hơn 100.000 bài hát được tải lên các đơn vị âm nhạc trực tuyến. Việc có khả năng phân loại ngay lập tức các bài hát trong bất kỳ danh sách phát hoặc thư viện nào theo thể loại là một chức năng quan trọng đối với bất kỳ dịch vụ phát nhạc/mua nhạc nào, và khả năng phân tích thống kê mà đánh nhãn đúng và đầy đủ về âm nhạc và âm thanh mang lại là vô cùng hữu ích.

### 1. Động lực nghiên cứu

Từ lâu, việc nhận diện và phân loại thể loại âm nhạc không đơn thuần dừng lại ở một bài tập về nhà mà còn là một trong những nhiệm vụ quan trọng trong lĩnh vực xử lý âm thanh. Đặc biệt hơn, “Phân loại âm nhạc truyền thống Việt Nam” không chỉ là một nhiệm vụ nghiên cứu mà còn là một hành trình khám phá, tôn vinh và bảo tồn di sản văn hóa. Việc xây dựng một hệ thống phân loại âm nhạc truyền thống không chỉ giúp chúng ta hiểu rõ hơn về sự đa dạng của âm nhạc Việt Nam mà còn tạo ra cơ hội để tạo ra những sản phẩm công nghệ có thể giúp mọi người kết nối và hiểu biết về nền văn hóa Việt Nam nói riêng và văn hóa thế giới nói chung.

Triển vọng của đề tài này không chỉ giới hạn trong phạm vi nghiên cứu mà còn mở ra những cánh cửa cho việc phát triển ứng dụng giáo dục, du lịch văn hóa và tạo ra trải nghiệm âm nhạc độc đáo. Thông qua việc ứng dụng công nghệ, chúng ta có thể làm cho âm nhạc truyền thống không chỉ là một phần quan trọng của quá khứ mà còn là một phần sống động, đầy sức sống trong hiện tại và tương lai.

### 2. Mục tiêu đề án

Mục tiêu của đề án "Phân loại âm nhạc truyền thống Việt Nam" là xây dựng một hệ thống phân loại âm nhạc truyền thống Việt Nam thông qua sự kết hợp giữa nghệ thuật và công nghệ. Dưới đây là một số mục tiêu cụ thể của đề án:

1. Phân loại chính xác: Xây dựng một mô hình máy học có khả năng phân loại các thể loại âm nhạc truyền thống Việt Nam với độ chính xác cao.

2. Bảo tồn di sản âm nhạc: Đề án nhằm giữ gìn và bảo tồn di sản âm nhạc truyền thống Việt Nam, giúp nâng cao nhận thức và sự hiểu biết về giá trị văn hóa của các thể loại âm nhạc này.

3. Tạo ra sản phẩm ứng dụng: Phát triển ứng dụng hoặc giao diện trực tuyến để người dùng có thể trải nghiệm, khám phá, và học hỏi về âm nhạc truyền thống Việt Nam một cách thuận tiện và thú vị.

4. Khuyến khích nghiên cứu và phát triển tiếp theo: Đề án có thể tạo động lực cho các nghiên cứu và dự án tương tự trong tương lai, đóng góp vào sự phát triển tiếp theo của lĩnh vực này.

Đề án này sẽ tập trung nghiên cứu các kỹ thuật xây dựng một hệ thống phân loại thể loại nhạc truyền thống Việt Nam, bao gồm: các kỹ thuật tiền xử lý âm thanh, kiến trúc các mô hình phân loại, ... Sử dụng nền tảng StreamLit để phát triển, xây dựng giao diện ứng dụng dựa trên công nghệ học máy để cung cấp cho người dùng trải nghiệm chính xác và nhanh nhạy.

### **3. Cấu trúc đề án**

Đề án có cấu trúc như sau:

**MỞ ĐẦU:** Giới thiệu và đưa ra hướng nghiên cứu bài toán Phân loại âm nhạc theo chủ đề.

**CHƯƠNG 1:** Tổng quan bài toán "Phân loại thể loại âm nhạc truyền thống Việt Nam"

**CHƯƠNG 2:** Xây dựng mô hình, thực nghiệm và đánh giá

**CHƯƠNG 3:** Xây dựng hệ thống phân loại thể loại âm nhạc

**KẾT LUẬN:** Đưa ra những kết luận, đánh giá và định hướng nghiên cứu tiếp theo

**TÀI LIỆU THAM KHẢO:** Tài liệu tham khảo và phụ lục

## CHƯƠNG 1: TỔNG QUAN BÀI TOÁN “PHÂN LOẠI THỂ LOẠI ÂM NHẠC TRUYỀN THỐNG VIỆT NAM”

Trong chương này sẽ giới thiệu tổng quan về đề án, kiến trúc và các kỹ thuật xử lý được sử dụng trong bài toán.

- Bài toán phân loại âm thanh: đưa ra định nghĩa thế nào là một bài toán phân loại âm thanh; trình bày các kỹ thuật được sử dụng để trích xuất đặc trưng âm thanh và phương pháp đánh giá độ hiệu quả của một mô hình phân lớp.
- Mạng nơ-ron tích chập CNN: giới thiệu tổng quát về mạng nơron tích chập và các lớp phổ biến của nó; trình bày kiến trúc cho một vài bộ CNN nhiều tầng nổi bật.

### 1.1. Phát biểu bài toán

Bài toán phân loại âm thanh là bài toán phân loại có đối tượng phân loại là âm thanh. Mục tiêu chính của bài toán này là thực hiện việc gán nhãn một cách tự động cho một đoạn âm thanh đầu vào dựa trên một tập nhãn hữu hạn của một nhóm dữ liệu âm thanh cho trước.

Có nhiều cách định nghĩa bài toán này dựa trên cách tiếp cận của chúng ta, tuy nhiên với phương pháp tiếp cận “học sâu”, ta có thể định nghĩa bài toán này như sau: Giả sử ta có một tập âm thanh  $D$  có  $N$  đoạn âm thanh và có tập nhãn là  $Y$ . Mỗi đoạn âm thanh  $x$  có độ dài  $s$  (giây) và được gán nhãn  $y$ . Khi đó ta có thể ký hiệu  $D = (x_i, y_i)$  với  $i \in [1, 2, \dots, N]$ ,  $x_i \in Y$ . Vậy quá trình phân loại âm thanh thực chất là một ánh xạ từ  $R$  vào  $Y$ , ký hiệu:  $f: R \rightarrow Y$  và bài toán phân loại bản chất là đi tìm hàm  $f$  sao cho kết quả giống với nhãn thực tế nhất.



Hình 1.1 Ảnh minh họa Phân loại âm nhạc

Tuy nhiên, việc đánh giá sự giống nhau một cách trực tiếp là không thể các nhãn là độc lập và việc làm này không có ý nghĩa. Vì thế, thay vì tính toán sự giống nhau đó, người ta thường đánh giá khả năng phân loại chính xác của một hàm  $f$  trên  $D$ , với mong muốn đó, nếu  $f$  có độ chính xác cao trên  $D$  thì  $f$  cũng sẽ có khả năng phân loại với độ chính xác cao trên một tập dữ liệu  $D'$  mới.

Tuy nhiên, việc đánh giá này có thể dẫn đến trường hợp hàm  $f$  quá phù hợp với  $D$  nhưng lại không phù hợp với một  $D'$  mới. Chính vì thế trong bài toán phân loại âm thanh, tập  $D$  thường được chia tối thiểu thành hai giai đoạn: Giai đoạn huấn luyện và giai đoạn kiểm nghiệm. Giai đoạn huấn luyện  $f$  sao cho độ chính xác trên tập huấn luyện cao nhất. Giai đoạn kiểm- nghiệm đánh giá lại khả năng phân loại của  $f$  trên tập thử nghiệm. Kết quả của giai đoạn thử nghiệm chính là kết quả cuối cùng của hàm  $f$ .

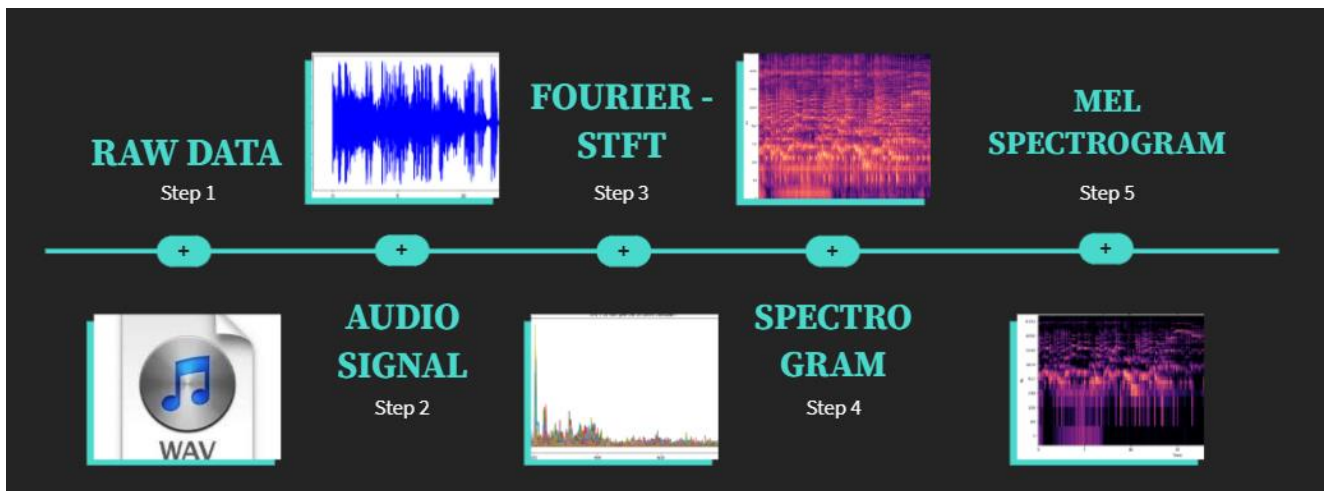
## 1.2. Kỹ thuật rút trích đặc trưng âm thanh

Trong quá trình giải bài toán phân loại âm thanh, trước khi bước vào hai giai đoạn huấn luyện và thử nghiệm, các tấm ảnh sẽ trải qua một giai đoạn tiền xử lý âm thanh để chuẩn hóa, tái tạo lại cấu trúc dữ liệu và lấy ra các thông tin hữu ích nhất để tối ưu hiệu suất cho mô hình. Trích xuất đặc trưng là một trong những kỹ thuật quan trọng và được sử dụng phổ biến trong giai đoạn này. Chi tiết về giai đoạn này sẽ được chúng tôi trình bày ở các bước sau.

Để đơn giản hơn, chúng tôi có thể tóm tắt đơn giản ý nghĩa thực chất giai đoạn này như sau: Vì CNN là mô hình thường được sử dụng đối với dữ liệu dạng hình ảnh, nên bước này của chúng tôi được áp dụng để biến đổi từ một đoạn âm thanh, qua các bước rút trích đặc trưng thành một tệp hình ảnh (ở dạng Mel - Spectrogram) sau đó mới đưa vào mô hình để xử lý với các bước tiếp theo.

### 1.2.1. Chi tiết kỹ thuật tiền xử lý dữ liệu âm thanh

Trong phần này, chúng tôi sẽ trình bày chi tiết cách xử lý dữ liệu Audio để biến nó phù hợp với dữ liệu đầu vào của mô hình CNN. Các bước xử lý sẽ được trình bày trong hình sau đây:



Hình 1.2 Các bước tiền xử lý dữ liệu Audio

#### 1.2.1.1. Tín hiệu âm thanh (Audio Signal)

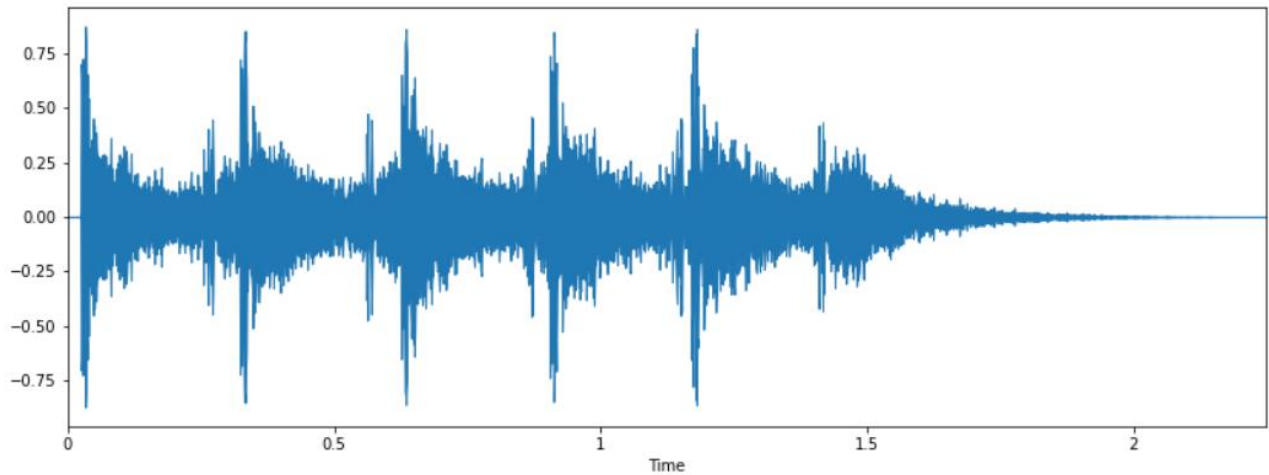
Âm thanh chúng ta nghe được thực tế ở dạng tín hiệu tương tự, nó được số hóa và lưu lại theo các định dạng khác nhau: .wav, .mp3, .wma, ...

Để đưa vào máy tính xử lý, 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]
```

Hình 1.3 Định dạng âm thanh được lưu trong máy tính

Trong bộ nhớ, Audio có thể coi là một chuỗi các giá trị của biên độ theo thời gian. Ví dụ, nếu tần số lấy mẫu là 16800Hz thì cứ 1s Audio sẽ có 16800 giá trị biên độ. Khi đọc tệp âm thanh thì nó sẽ được thể hiện trên đồ thị như sau:



Hình 1.4 Biểu đồ Audio Signal cho 1 file âm thanh .wav dài 2.5s.

#### 1.2.1.2. 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.


Thì STFT là một 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.


- Formulation:
  - Continuous STFT

$$\text{STFT}\{x(t)\}(\tau, \omega) = X(\tau, \omega) = \int_{-\infty}^{\infty} x(t)w(t-\tau)e^{-i\omega t} dt$$

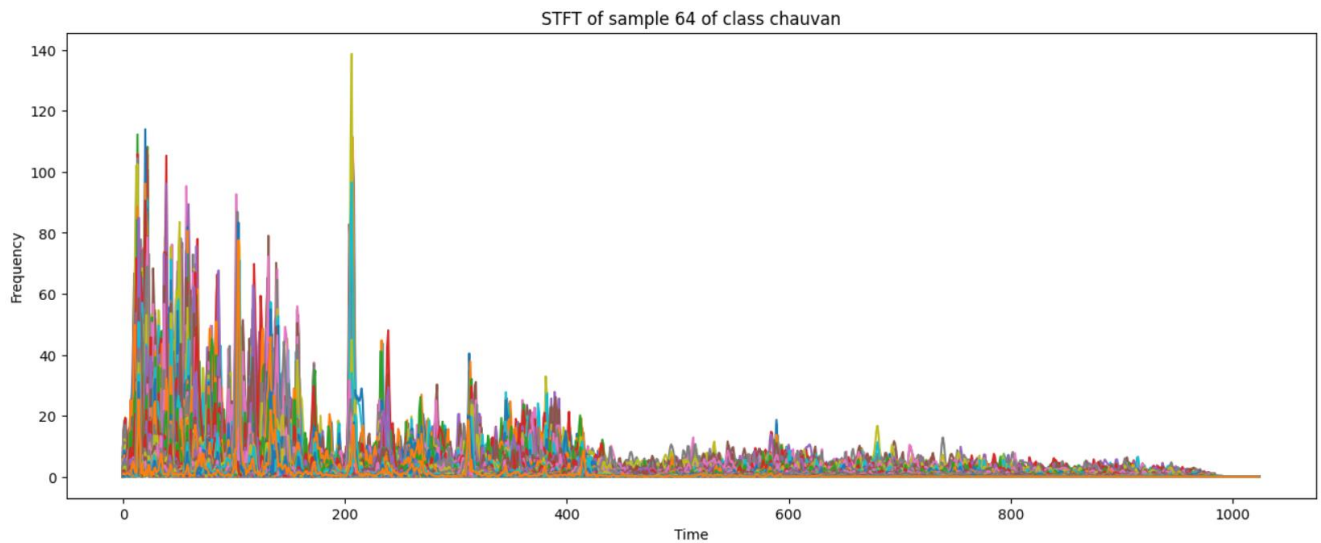
$x(t)$	Time-domain signal to be transformed
$\tau$	Time (slow time; lower resolution than $t$ )
$\omega$	Frequency
$w(t)$	Window function, commonly a Hann window or Gaussian window bell centered around zero
$X(\tau, \omega)$	A complex function representing the phase and magnitude of the signal over time and frequency (this is essentially the Fourier Transform of $x(t)w(t-\tau)$ )



**Sharif University of Technology**  
Department of Civil Engineering



Hình 1.5 Công thức biến đổi STFT.



Hình 1.6 Biểu đồ biến đổi STFT.

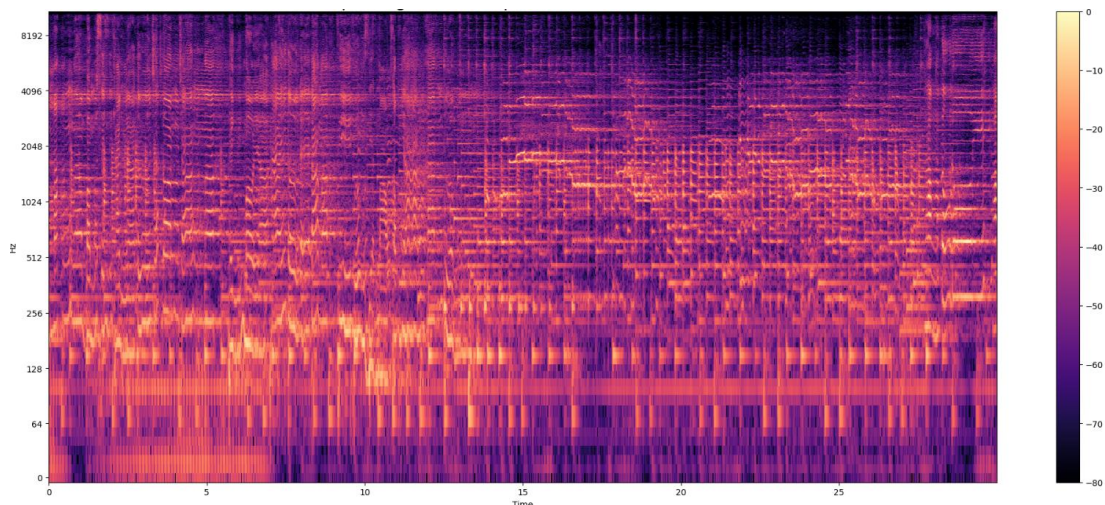
Như vậy, phép biến đổi STFT sẽ chuyển đổi một tín hiệu từ miền thời gian sang miền tần số, đồng thời chúng còn chuyển đổi một tín hiệu từ miền thời gian sang miền tần số. Tuy nhiên, hạn chế của biểu diễn miền tần số là không có thông tin về thời gian.

### 1.2.1.3. Spectrogram

Trong phần trước, chúng ta đã biểu diễn 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.





Hình 1.7 Spectrogram biểu diễn cho một file .wav trong bộ dataset.

#### 1.2.1.4. Mel – Spectrogram

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:

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

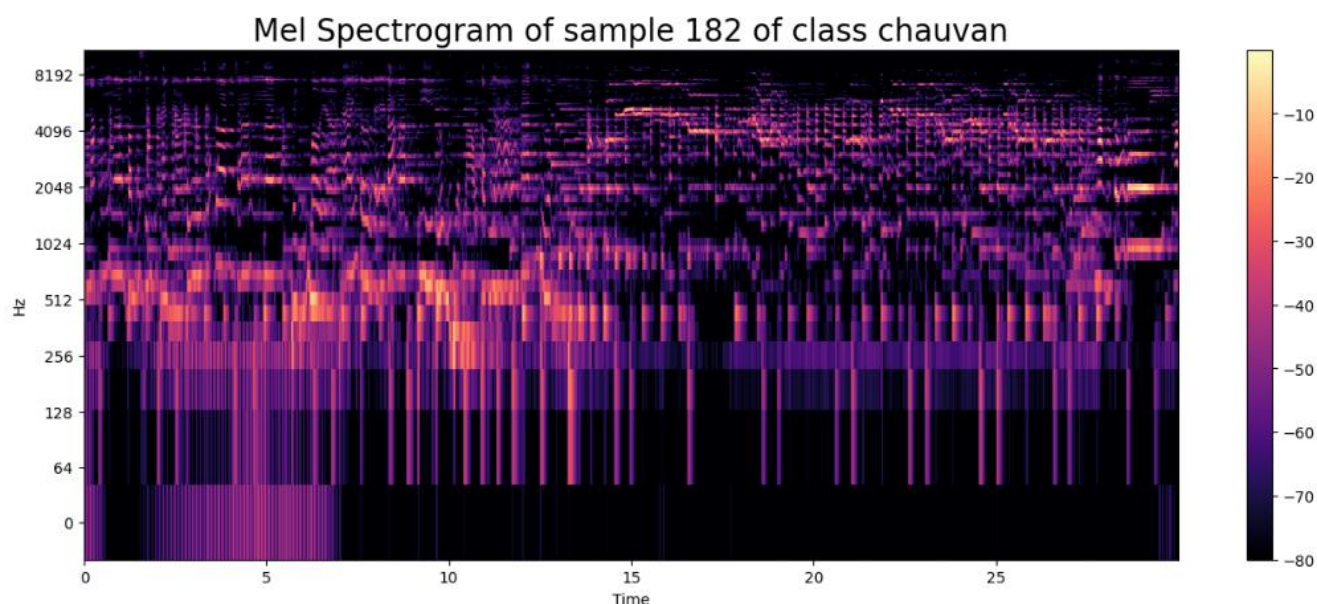
Hình 1.8 Công thức chuyển đổi từ thang đo Hz sang Mel.

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 Decibel Scale thay vì biên độ để biểu diễn

biểu đồ Mel - Spectrogram:



Hình 1.9 Mel - Spectrogram cho một file .wav trong bộ dataset.

### 1.2.2. Đánh giá mô hình phân lớp

Ở phần này chúng tôi sẽ đề cập đến các độ đo cho mô hình phân lớp. Các độ đo sẽ cho ta các nhìn tổng quát về hiệu quả của mô hình. Một độ đo tốt có thể cho ta cái nhìn tổng thể hơn về mô hình và đánh giá mô hình một cách khách quan hơn. Từ kết quả mà độ đo mang lại, ta có thể đưa ra các phương pháp cải tiến và nâng cao độ hiệu quả của mô hình. Trong bài toán phân loại thể loại âm nhạc, có 4 độ đo thông dụng như sau: Accuracy, Precision, Recall, F1\_score.

Confusion matrix: là một ma trận lưu lại kết quả phân loại của mô hình theo các nhãn thực và dự đoán. Confusion matrix cho bài toán phân loại nhị phân có dạng như sau:

		True Class	
		Positive	Negative
Predicted Class	Positive	TP	FP
	Negative	FN	TN

Hình 1.10 Ví dụ về confusion matrix.

Trong đó:

- TP: là số lần nhãn dự đoán là Positive và trùng với nhãn thực tế
- TN: là số lần nhãn dự đoán là Negative và trùng với nhãn thực tế
- FP: là số lần nhãn dự đoán là Positive và không trùng với nhãn thực tế
- FN: là số lần nhãn dự đoán là Negative và không trùng với nhãn thực tế

Độ đo Accuracy: Là độ đo cho thấy khả năng dự đoán đúng nhãn của mô hình.

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

Độ đo Precision: là tỷ lệ bao nhiêu cái đúng được lấy ra, cân nhắc trên tập dữ liệu kiểm soát xem có bao nhiêu dữ liệu được mô hình phán đoán đúng.

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

Độ đo Recall: là tỷ lệ bao nhiêu cái được lấy ra là đúng, chỉ số này còn được gọi là độ bao phủ, tức là xét xem mô hình tìm được có khả năng tổng quát hóa như nào.

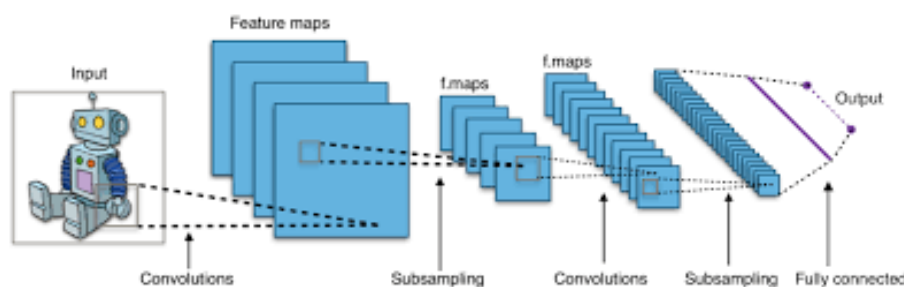
$$Recall = \frac{TP}{TP + FN}$$

Độ đo F1\_score: Là độ đo trung bình điều hòa giữa giữa 2 độ đo Precision và Recall. Giá trị của F1\_score càng cao, tức là càng gần 1 thì hiệu suất của mô hình càng cao.

$$F1_{score} = 2 * \frac{precision * recall}{precision + recall}$$

### 1.2.3. Mạng nơ-ron tích chập - CNN

Mạng nơ-ron tích chập có tên tiếng anh là Convolutional Neural Network (hay còn viết tắt là CNN). Là một trong những mô hình học sâu khá lâu đời và nổi tiếng nhất trong giới Khoa Học Máy Tính, được sử dụng rộng rãi trong lĩnh vực Thị Giác Máy Tính. Lấy cảm hứng từ mạng nơ-ron nhân tạo (ANN), chỉ có điểm khác là thêm một bộ trích xuất đặc trưng ở lớp trước. CNN hoạt động theo nguyên lý “đầu cuối” (end to end), tức là ngay sau khi đưa dữ liệu vào, mô hình sẽ tự động học và cho ra kết quả rất tốt mà không cần thêm bước xử lý nào.



Hình 1.11 Cấu tạo của một mạng nơ-ron chập tích cơ bản.

Một mạng CNN cơ bản sẽ bao gồm hai thành phần chính là: “trích xuất đặc trưng” và phần “liên kết toàn phần” (fully-connected hay FC).

Phần “trích xuất đặc trưng” về cơ bản sẽ là các lớp tích chập được đặt chồng lên nhau. Mỗi lớp “tích chập” sẽ có 2 thành phần chính:

- Thứ nhất là một tập hợp các “bộ lọc” (filter) hay còn được gọi là “nhân” (kernel).
- Thứ hai là hàm kích hoạt (activation function).

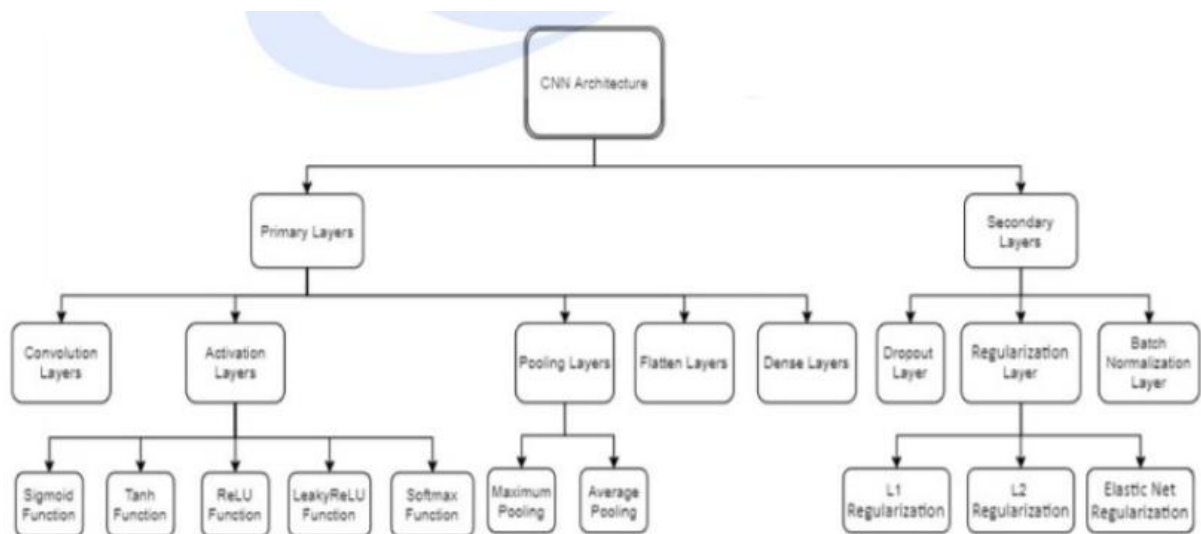
Các lớp “tích chập” có nhiệm vụ nhân tích chập dữ liệu đầu vào, sau đó đưa kết quả thu được qua hàm kích hoạt để xác định các thuộc tính quan trọng cần được giữ lại, và tạo

thành một “bản đồ kích hoạt” (activation map). Trừ lớp tích chập đầu tiên nhận dữ liệu đầu vào, các lớp “tích chập” sau đó sẽ nhận các “bản đồ tích chập” do các lớp tích chập trước nó tạo thành. Tuy nhiên, việc đặt các lớp “tích chập” vô tội vạ, liên tiếp nhau sẽ các thuộc tính bị lặp đi lặp lại nhiều lần, gây ra hiện tượng tốn tài nguyên, việc học cũng không được cải thiện được mấy. Thế nên ta nên ta sẽ một lớp “pooling” để tránh hiện tượng kể trên, cũng như để mô hình trở nên tinh gọn và đơn giản hơn.

Ngoài ra, phần liên kết toàn phần (fully-connected) có cấu trúc giống một mạng nơ-ron nhân tạo ANN. Đầu vào của lớp này sẽ là một “bản đồ tích chập” là đầu ra của khối “tích chập” đã được kéo dãn ra “flatten”. Đầu ra của khối này sẽ là vector đại diện cho kết quả của bài toán. Ở giữa sẽ là các “lớp ẩn” (hidden layer). Mỗi lớp ẩn sẽ có một số lượng nơ-ron riêng và mỗi nơ-ron này sẽ liên kết với toàn bộ nơ-ron ở lớp kế tiếp.

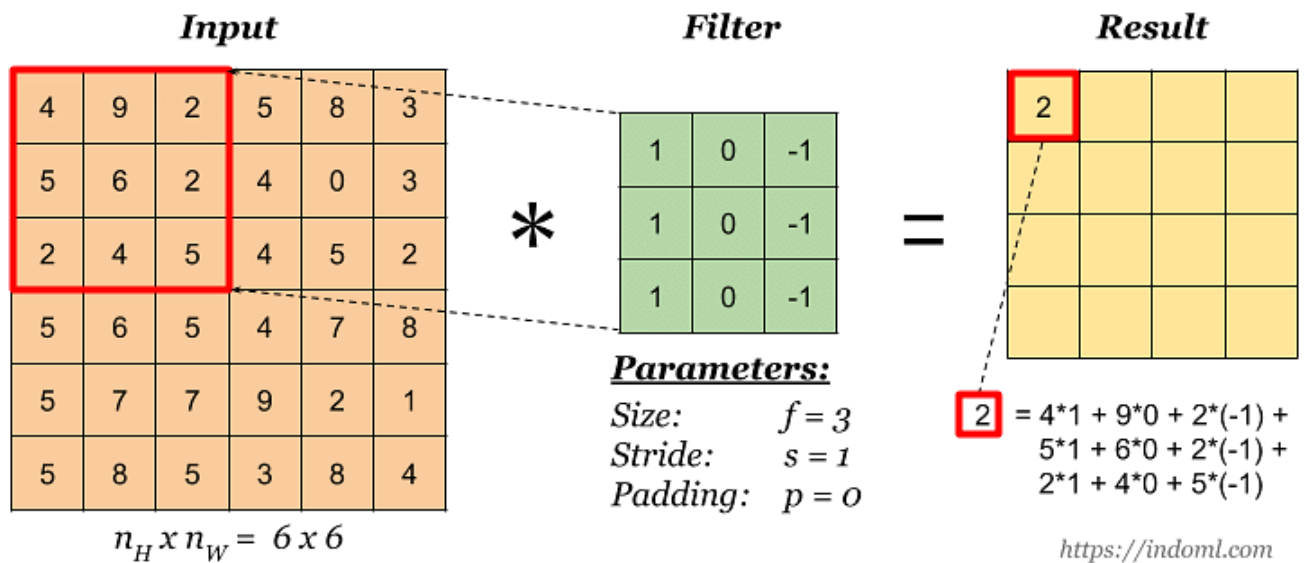
### 1.2.3.1. Mạng nơ-ron tích chập - CNN

Ở phần trước chúng tôi đã nói sơ bộ và kiến trúc của một mạng CNN, phần này ta sẽ đi qua chi tiết của mỗi lớp. Sau đây là chi tiết về các lớp của mạng CNN.



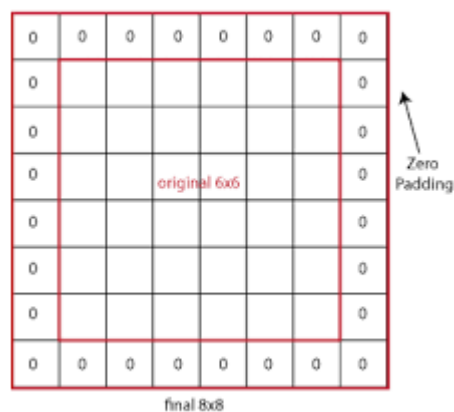
Hình 1.12 Các lớp phổ biến của mô hình CNN.

Lớp “tích chập” (Convolutional Layers): Lớp này sẽ có nhiệm vụ rút trích đặc trưng từ ảnh đầu vào hay “bản đồ kích hoạt” trước đó và tạo ra một “bản đồ kích hoạt” mới.



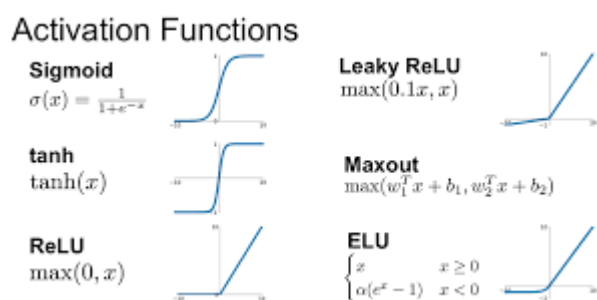
Hình 1.13 Hình ảnh minh họa đầu vào của lớp tích chập.

Kích thước đầu vào và đầu ra của lớp “tích chập” không giống nhau. Kích thước đầu ra sẽ phụ thuộc kích thước của “bộ lọc” (filters) và “tốc độ trượt” (stride). Nếu muốn khắc phục sự khác nhau về kích thước của ảnh đầu vào, một kỹ thuật được đề xuất gọi là “padding”.



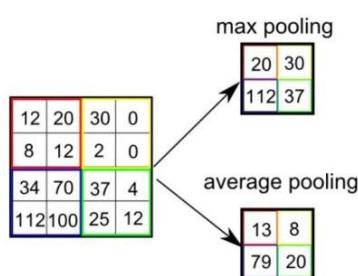
Hình 1.14 Kỹ thuật zero padding.

Lớp “kích hoạt” (Activation function): Lớp này thường đi liền sau lớp “tích chập”. Nó đảm nhận vai trò cho phép hay không cho phép nơ-ron nào đi qua. Các hàm kích hoạt thường được sử dụng là: Sigmoid, Tanh, Relu, ...



Hình 1.15 Các hàm kích hoạt phổ biến.

Lớp “lấy mẫu con” (Pooling Layer): Lớp này có nhiệm vụ giảm kích thước của mô hình xuống trong khi vẫn muốn giữ các đặc trưng cần thiết. Các hai kiểu lấy mẫu thông dụng là “Max Pooling” và “Average Pooling”.



Hình 1.16 Hai hàm Pooling thông dụng.

Lớp Flattening Layer: Lớp này đảm nhận việc kéo dẫn input đầu vào về dạng vector 1 chiều. Trong mô hình thì lớp này thường là lớp duy nhất, và được đặt ngay trước lớp FC.

Lớp Dense layer: Lớp này còn được gọi là lớp FC. Mỗi nơ-ron trong lớp này thường được liên kết đến với mỗi nơ-ron trong lớp tiếp theo hay lớp kế trước nó.

Lớp Drop-out: Lớp này có thể được áp dụng cho mọi lớp trong mô hình với ý nghĩa: Trong quá trình huấn luyện, mô hình sẽ chủ động bỏ X nơ-ron nào đó để tránh over-fitting.

Batch Normalization (BN): Lớp này giúp cải thiện quá trình huấn luyện bằng cách chuẩn hóa các đầu ra tầng trung gian trong mỗi mini-batch. BN giúp ổn định và tăng

tốc quá trình học bằng cách đảm bảo rằng đầu vào cho mỗi tầng có phân phối trung bình xấp xỉ 0 và độ lệch tiêu chuẩn xấp xỉ 1.



## CHƯƠNG 2: XÂY DỰNG MÔ HÌNH, THỰC NGHIỆM VÀ ĐÁNH GIÁ

Trong chương này sẽ mô tả chi tiết về bộ dữ liệu chúng tôi sử dụng, quá trình, kết quả của các thí nghiệm mà chúng tôi thực nghiệm. Nội dung chương này sẽ có 4 phần:

- Bộ dữ liệu: Giới thiệu về bộ dữ liệu mà chúng tôi sử dụng để tiến hành thực nghiệm
- Các thí nghiệm
- Xây dựng và hoàn thiện mô hình

### 2.1. Khảo sát dữ liệu

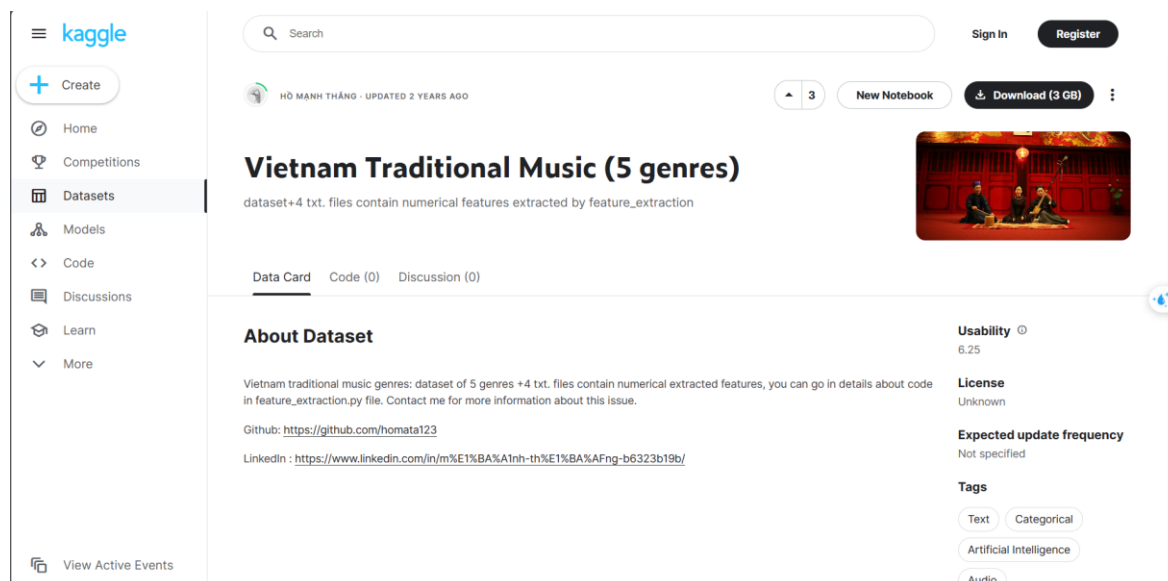
Đây là bài toán nhận được sự quan tâm của cộng đồng các nhà nghiên cứu trong lĩnh vực Xử lý Âm thanh nói chung. Các bộ dữ liệu trong đề tài Phân loại âm nhạc khá nhiều, các bộ dữ liệu nổi tiếng có thể được kể đến như:

1. Bộ dữ liệu GTZAN: Đây là bộ dữ liệu về Phân loại thể loại âm nhạc khá lớn, được phát hành vào năm 2002. Trong bộ dữ liệu có 10 thể loại chính và bao gồm 1000 file WAV chia đều cho cả 10 thể loại.
2. Bộ dữ liệu MagnaTagATune: Đây là bộ dữ liệu được công bố vào năm 2009 và phát hành như một phần của cuộc thi "MIREX Audio Music Similarity and Retrieval". Trong bộ dữ liệu có khoảng 25000 mẫu âm thanh từ nhiều nghệ sĩ và thể loại khác nhau.
3. Bộ dữ liệu Million Song Dataset: là một tập dữ liệu âm nhạc lớn, được phát triển để hỗ trợ nghiên cứu trong lĩnh vực xử lý âm thanh và khoa học dữ liệu âm nhạc. Phát hành vào năm 2011, bộ dữ liệu này chứa thông tin về một triệu bài hát, bao gồm metadata như nghệ sĩ, tựa đề, thể loại, và các đặc trưng âm thanh như tempo, key, và energy.
4. Bộ dữ liệu MTG-Jamendo: Đây cũng là bộ dữ liệu mới nhất về chủ đề Phân loại âm nhạc mà chúng tôi tìm được. Trong bộ dữ liệu có khoảng 55000 file âm thanh và được chia thành 195 thể loại khác nhau.

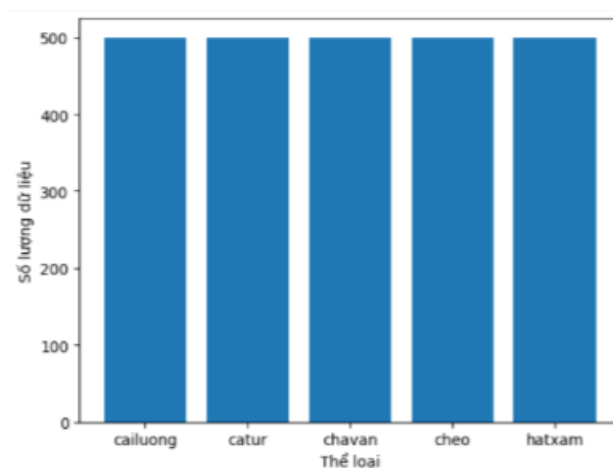
Các bộ dữ liệu về cùng thể loại này còn khá nhiều nhưng chúng tôi chỉ kể đến những bộ dữ liệu tiêu biểu nhất. Ngoài ra, các bộ dữ liệu trên đều là những bộ dữ liệu trên ngôn ngữ khác, không phải trên Tiếng Việt nên chúng tôi sẽ không thực nghiệm trên những bộ dữ liệu này.

## 2.2. Bộ dữ liệu

Trong đồ án này, chúng tôi lựa chọn bộ dataset "Vietnam Traditional Music (5 genres)" trên Kaggle. Bộ dataset này được lấy từ hơn 20 giờ record thuộc 200 bài hát, bao gồm 2500 file .wav được chia đều thành 5 thể loại nhạc: cailuong, catru, chauvan, cheo, hatxam. Mỗi bài hát được tách ra thành các file data có độ dài 30 giây và chia đều 500 file ở mỗi thể loại nhạc. Mỗi file data có thể thuộc cùng bài hát (độ dài mỗi bài khoảng 4 phút).

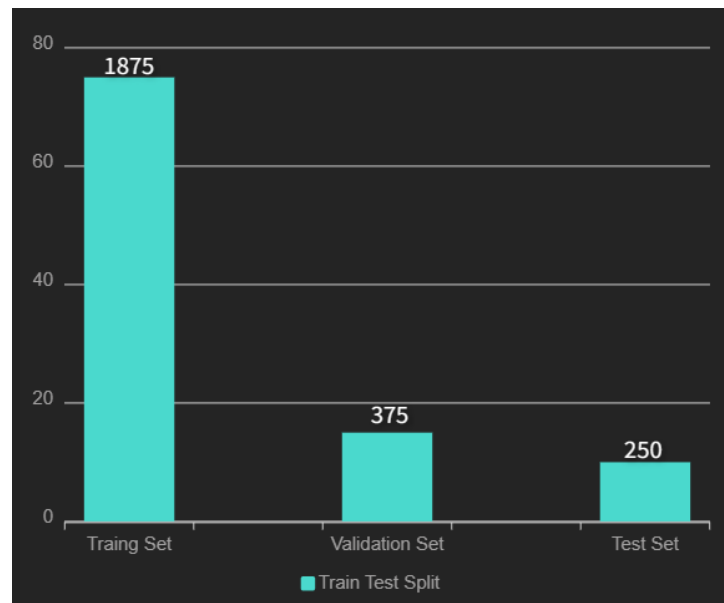


Hình 2.1 Bộ dataset trên kaggle mà nhóm sử dụng



Hình 2.2 Số lượng bài hát trong từng thể loại

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



Hình 2.3 Số lượng bài hát trong từng tập dữ liệu

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.

## 2.3. Các thí nghiệm

Tất cả các thí nghiệm trong phần này đều được thực nghiệm trên bộ dữ liệu chúng tôi đã nêu ở trên. Việc đánh giá và thực nghiệm được chúng tôi tiến hành trên Google Colab phiên bản thường.

Cấu hình máy thực nghiệm:

- Google Colab
- GPU: Tesla T4
- Ram: 21GB

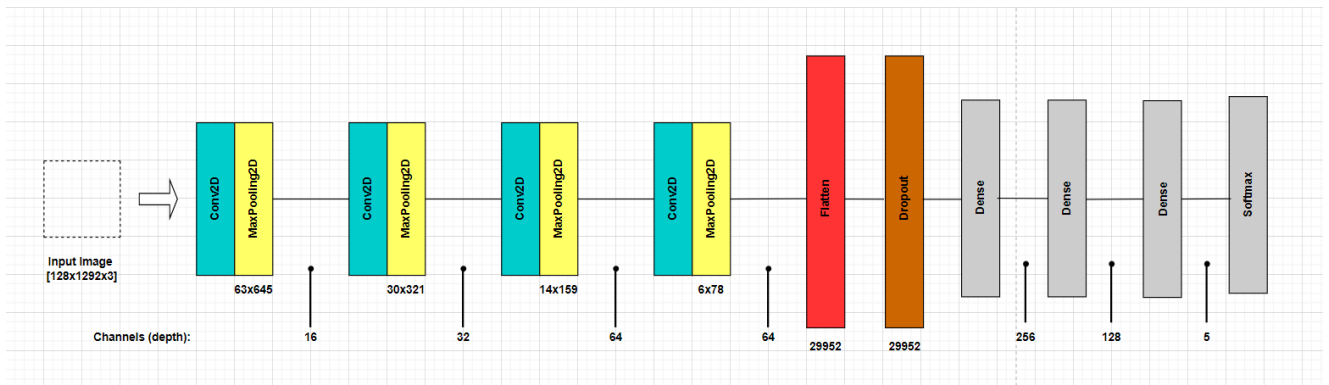
### 2.2.1. Thí nghiệm 1: Tiến hành thực nghiệm trên mô hình CNN tự xây dựng thứ 1

#### 2.2.1.1. Kiến trúc mô hình cơ sở

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 trong mô hình CNN đầu tiên:



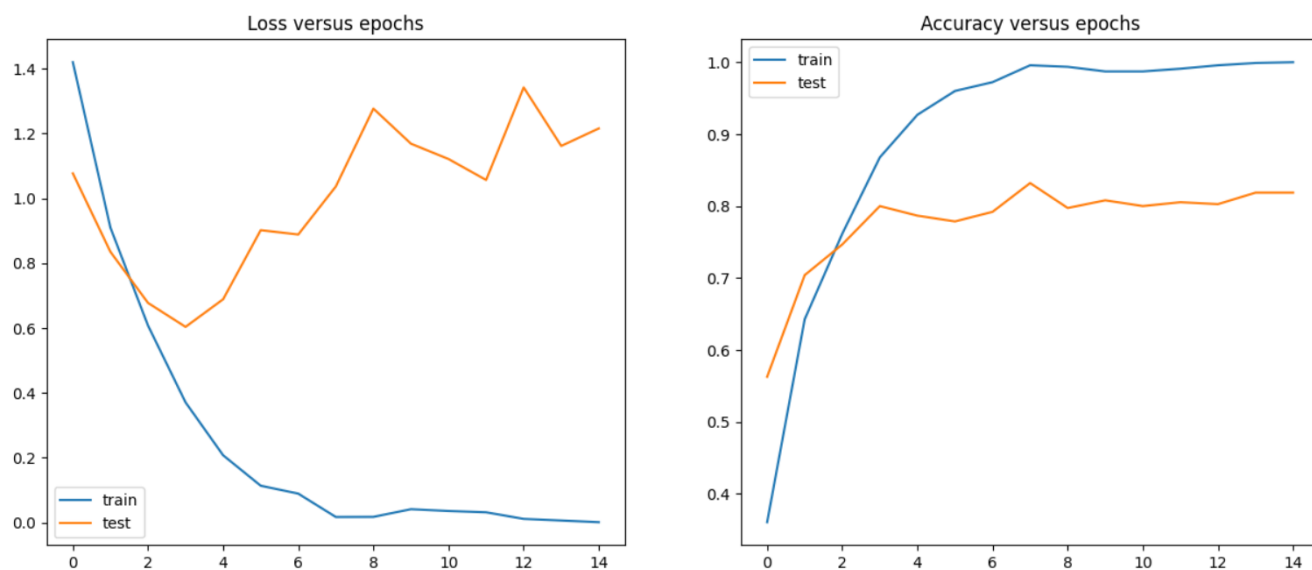
Hình 2.4 Tổng quan cấu trúc các lớp mô hình CNN thứ 1

Với kiến trúc trên, ban đầu ảnh đầu vào sẽ có kích thước ( $W \times H \times C$ ) lần lượt là ( $128 \times 129 \times 3$ ), sau đó sẽ đi qua khối Chập Tích đầu tiên với 16 kernel với  $kernel\_size$  là ( $3 \times 3$ ) và sử dụng hàm kích hoạt là “relu”. Sau đó, một bản đồ đặc trưng được tạo ra với kích thước ( $W \times H \times C$ ) là ( $63 \times 645 \times 16$ ), sau đó, lại tiếp tục đi qua một lớp MaxPooling2D với  $pool\_size$  là ( $2 \times 2$ ) để bảo toàn được tính nguyên vẹn của các đặc trưng đã phát hiện được trên ảnh đồng thời cũng giảm được kích thước của bản đồ đặc trưng. Sau đó lại cho các bản đồ đặc trưng lại được tiếp tục tuần tự đi qua bốn khối chập tích có thông số tương tự về  $kernel\_size$  ( $3 \times 3$ ), padding (‘valid’), stride (1), hàm kích hoạt là “ReLU” và MaxPooling2D với  $pool\_size$  ( $2 \times 2$ ). Tuy nhiên, ở mỗi khối Chập Tích sẽ có sự khác nhau về chiều channels lần lượt ở khối thứ nhất, thứ hai, thứ ba, thứ tư là: 16, 32, 64, 64.

Sau đó, dữ liệu sẽ được làm phẳng ra theo chiều dọc bằng việc sử dụng một lớp Flatten, đi kèm với đó sẽ là một lớp Dropout với tỷ lệ là 0.2 để tránh được hiện tượng OverFitting. Cuối cùng, dữ liệu sẽ đi vào 2 lớp Fully Connected với số node lần lượt là 256 và 128. Và để model có thể dự đoán được đoạn âm thanh thuộc vào thể loại nào, đầu ra của lớp FC cuối cùng sẽ đi vào lớp Dense cuối với số node là 5, bằng với số lớp của bộ dữ liệu, sử dụng hàm kích hoạt là Softmax để đảm bảo phân phối về tỷ lệ của các lớp dự đoán.

### 2.2.1.2. Kết quả thực nghiệm

Chúng tôi thực hiện huấn luyện tuần tự từng Model với số epochs,  $batch\_size$ , validation  $batch\_size$  lần lượt là: 50, 32, 32. Kết quả thực nghiệm thu được như sau:

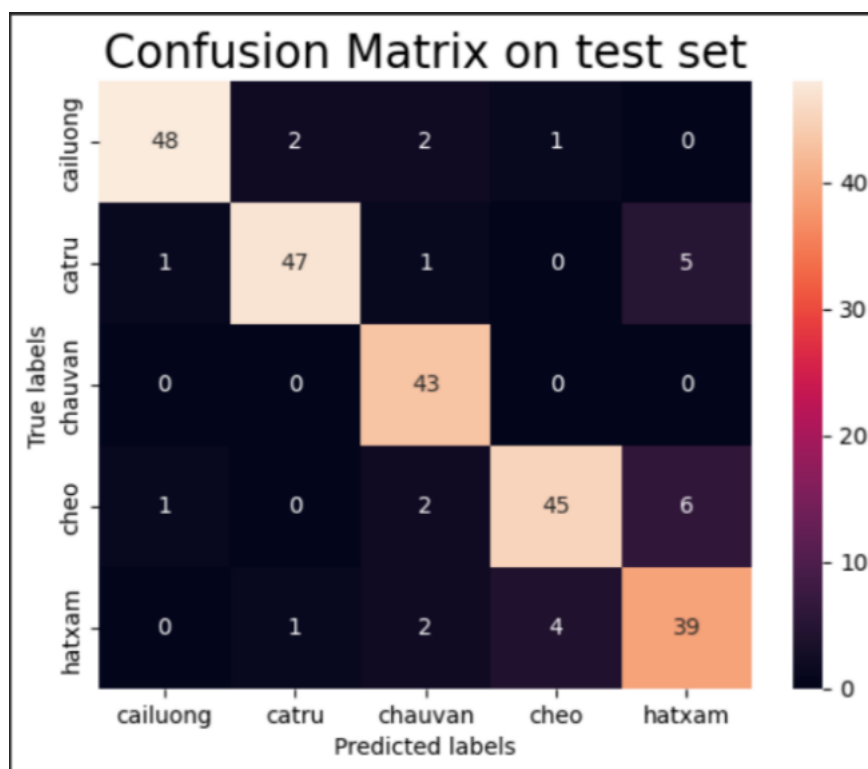


Hình 2.5 Training history mô hình CNN thứ 1

Training history của mô hình thứ 1 có phần ổn định khi giá trị Loss giảm đều qua các epoch của tập train tương tự như Accuracy của từng epoch. Tuy nhiên, khi thực hiện thử nghiệm ở tập dữ liệu test thì giá trị Loss tăng trở lại ở epoch thứ 3 do đó Accuracy có giá trị thấp hơn so với tập train

	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

Hình 2.6a Bảng hiệu suất mô hình CNN thứ 1



Hình 2.6b Bảng Confusion Matrix mô hình CNN thứ 1

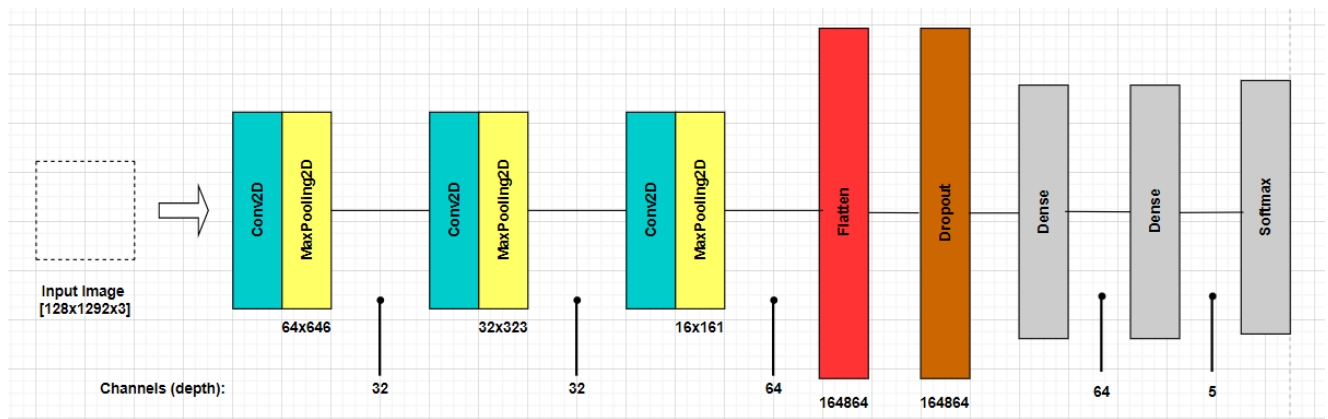
Mô hình thứ 1 có kết quả tương đối khi giá trị Loss và Accuracy trong quá trình huấn luyện khá ổn định, tuy nhiên khi thực hiện kiểm thử thì giá trị Loss bị biến động nhưng trên tổng thể giá trị Loss vẫn giảm và accuracy tăng đều (Hình 2.4). Mô hình đạt 0.89 trên 3 chỉ số precision, recall và f1 (Hình 2.5a). Thể loại “hát xẩm” là thể loại mô hình dự đoán sai nhiều nhất khi chỉ đoán đúng 39 sample, còn lại dự đoán là ”chèo” (4 sample), châu văn và ca trù (Hình 2.5b).

## 2.2.2. Thí nghiệm 2: Tiến hành thực nghiệm trên mô hình CNN tự xây dựng thứ 2

### 2.2.2.1. Kiến trúc mô hình cơ sở

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 trong mô hình CNN thứ hai:



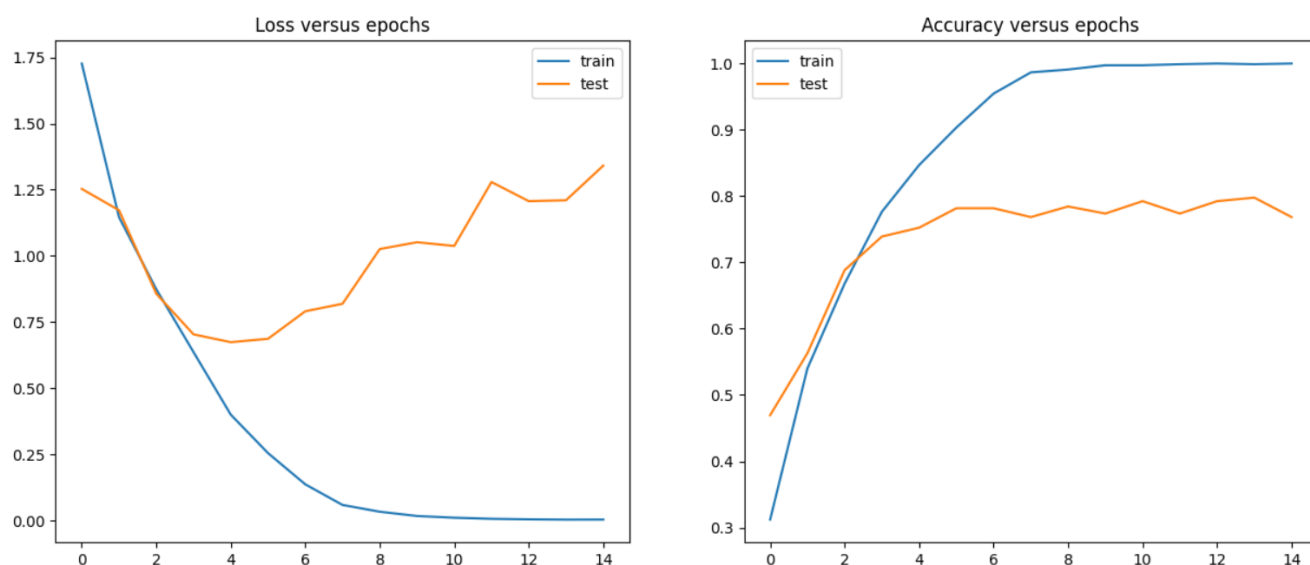
Hình 2.7 Tổng quan cấu trúc các lớp mô hình CNN thứ 2

Với kiến trúc trên, ban đầu ảnh đầu vào sẽ có kích thước (WxHxC) lần lượt là (128x1292x3), sau đó sẽ đi qua khối Chập Tích đầu tiên với với 32 kernel với kernel\_size là (5x5) và sử dụng hàm kích hoạt là “relu”. Sau đó, một bản đồ đặc trưng được tạo ra với kích thước (WxHxC) là (64x64x6x16), sau đó, lại tiếp tục đi qua một lớp MaxPooling2D với pool\_size là (1x1) để bảo toàn được tính nguyên vẹn của các đặc trưng đã phát hiện được trên ảnh. Sau đó lại cho các bản đồ đặc trưng lại được tiếp tục tuần tự đi qua ba khối chập tích có thông số tương tự về kernel\_size (3x3), padding (‘same’), stride (1), hàm kích hoạt là “ReLU” và MaxPooling2D với pool\_size (2x2). Tuy nhiên, ở mỗi khối Chập Tích sẽ có sự khác nhau về chiều channels lần lượt ở khối thứ nhất, thứ hai, thứ ba, thứ tư là: 32, 32, 64. Sở dĩ, chúng tôi cài đặt kiến trúc model 2 như trên là bởi vì chúng tôi muốn so sánh sự khác nhau về hiệu suất như thế nào khi cho model 1 và model có sự khác biệt về: số khối Chập Tích, số lượng kernel và kể cả kích thước của kernel.

Sau đó, dữ liệu sẽ được làm phẳng ra theo chiều dọc bằng việc sử dụng một lớp Flatten, đi kèm với đó sẽ là một lớp Dropout với tỷ lệ là 0.3 (khác biệt so với model 1) để tránh được hiện tượng OverFitting. Cuối cùng, dữ liệu sẽ đi vào 2 lớp Fully Connected với số node lần lượt là 64 và 5 đồng thời sử dụng hàm kích hoạt là Softmax để rút ra được kết quả dự đoán.

### 2.2.2.2. Kết quả thực nghiệm

Chúng tôi thực hiện huấn luyện tuần tự từng Model với số epochs, batch\_size, validation batch\_size lần lượt là: 50, 32, 32. Kết quả thực nghiệm thu được như sau:



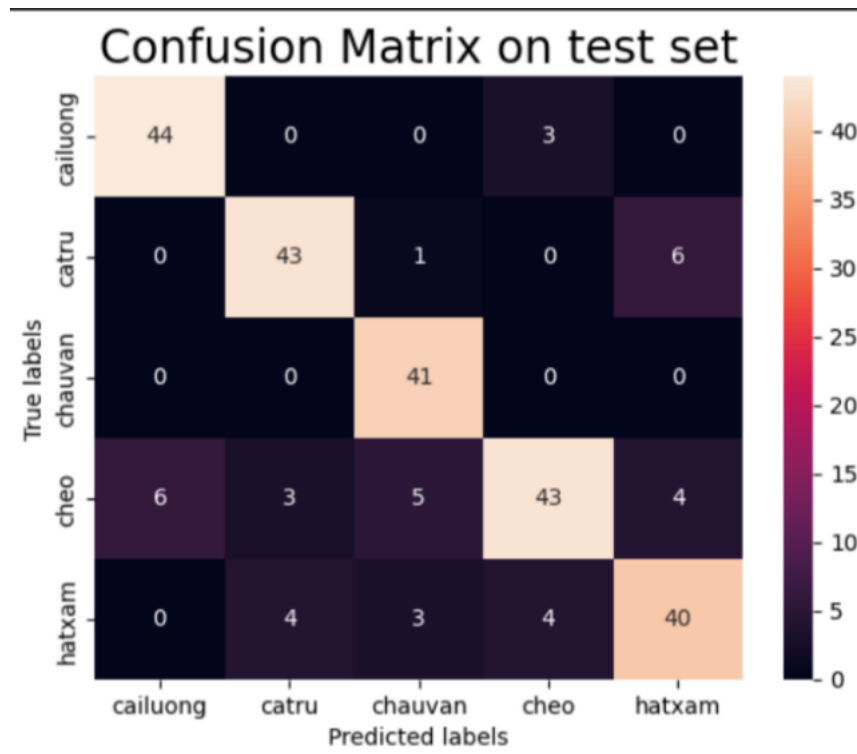
Hình 2.8 Training history mô hình CNN thứ 2

Với mô hình thứ 2, giá trị Loss có phần giảm nhanh hơn với mô hình 1 trong quá trình huấn luyện, đồng thời Accuracy trên tập test đạt giá trị rất cao. Ngược lại, trường hợp của mô hình trước bị lặp lại khi Loss tăng lại rõ rệt ở khoảng epoch 6 và Accuracy cũng giảm rất nhiều.

	precision	recall	f1-score	support
Cải Lương	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

Hình 2.9a Bảng hiệu suất mô hình CNN thứ 2





Hình 2.9b Bảng Confusion matrix của mô hình thứ 2

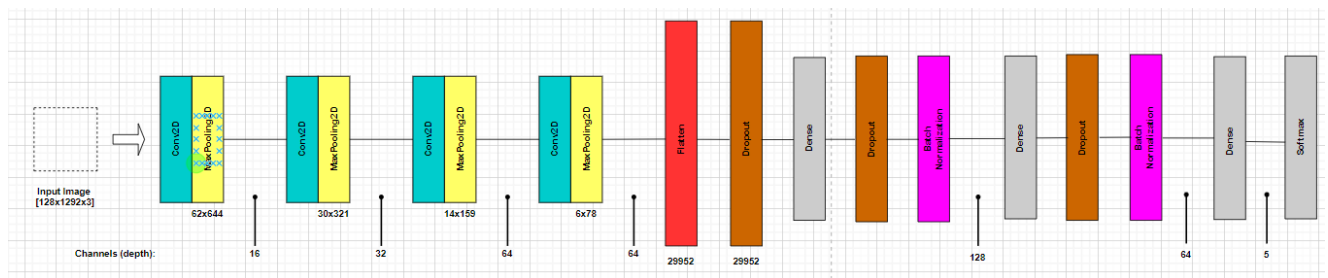
Với mô hình CNN thứ 2, các chỉ số có phần ổn định và tốt hơn mô hình 1, Với hiệu suất training khá tốt và biểu đồ Loss cũng ít biến động hơn, tuy nhiên mô hình 2 vẫn có Loss tăng trở lại khi thực hiện thử nghiệm (Hình 2.7). Các chỉ số đánh giá của mô hình 2 đều đạt 0.84 theo weighted average (Hình 2.8a) và “Chèo”, ”Hát xẩm” là thể loại nhạc bị dự đoán sai nhiều nhất với lần lượt là 14 và 11 sample bị dự đoán sai (Hình 2.8b)

### 2.2.3. Thí nghiệm 3: Tiến hành thực nghiệm trên mô hình CNN tự xây dựng thứ 3

#### 2.2.3.1. Kiến trúc mô hình cơ sở

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 trong mô hình CNN thứ ba:



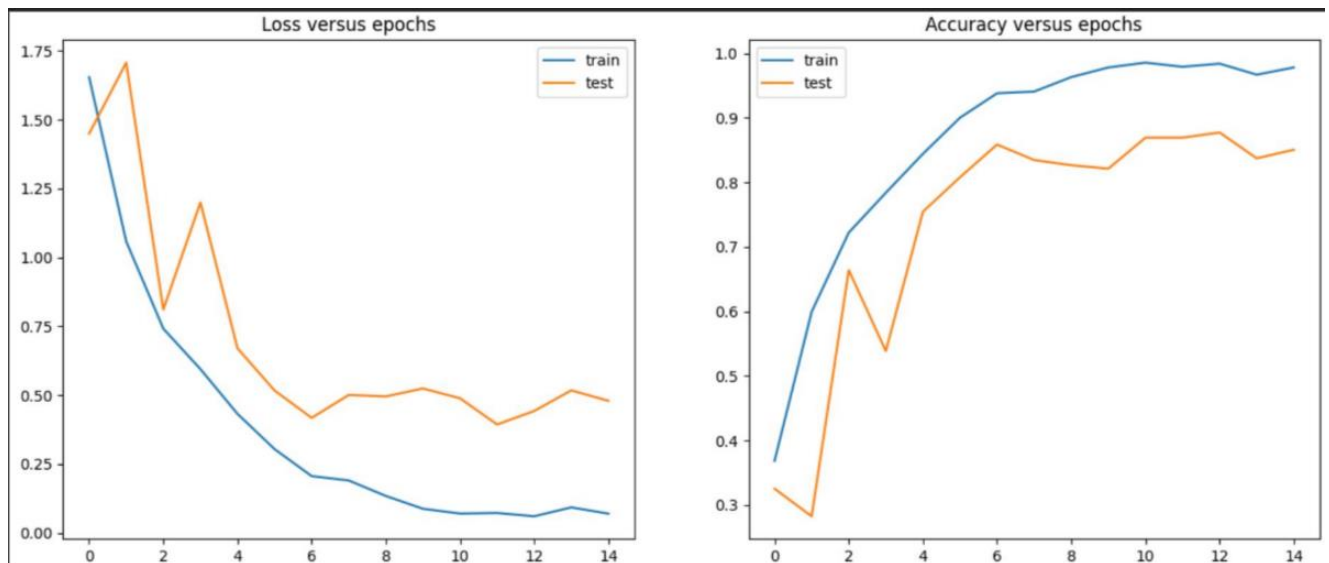
Hình 2.10 Tổng quan cấu trúc các lớp mô hình CNN thứ 3

Với kiến trúc trên, ban đầu ảnh đầu vào sẽ có kích thước (WxHxC) lần lượt là (128x1292x3), sau đó sẽ đi qua khối Chập Tích đầu tiên với 16 kernel với kernel\_size là (5x5) và sử dụng hàm kích hoạt là “relu”. Sau đó, một bản đồ đặc trưng được tạo ra với kích thước (WxHxC) là (64x646x16), sau đó, lại tiếp tục đi qua một lớp MaxPooling2D với pool\_size là (2x2) để bảo toàn được tính nguyên vẹn của các đặc trưng đã phát hiện được trên ảnh và giảm kích thước dữ liệu. Sau đó lại cho các bản đồ đặc trưng lại được tiếp tục tuần tự đi qua bốn khối chập tích tương tự như model 1 với số out channels lần lượt là: 16, 32, 64, 64.

Sau đó, dữ liệu sẽ được làm phẳng ra theo chiều dọc bằng việc sử dụng một lớp Flatten, đi kèm với đó sẽ là một lớp Dropout với tỷ lệ là 0.2 (giống với model 1) để tránh được hiện tượng OverFitting. Cuối cùng, dữ liệu sẽ đi vào 1 lớp Fully Connected với số node lần lượt là 128 và sử dụng hàm kích hoạt là “ReLU”. Và lại tiếp tục gắn một lớp Dropout với tỷ lệ là 0.2. Sự khác biệt của model 3 với model 1 và model 2 là có sử dụng một lớp BatchNorm làm cho quá trình huấn luyện dữ liệu nhanh và ổn định hơn. Sau đó vẫn sẽ dùng một lớp Fully Connected với số node là 64 và hàm kích hoạt là ReLu và vẫn sử dụng một lớp Dropout(0.2) và một lớp BatchNorm theo sau để ổn định quá trình huấn luyện. Đầu ra của lớp FC cuối cùng sẽ đi vào lớp Dense cuối với số node là 5, bằng với số lớp của bộ dữ liệu, sử dụng hàm kích hoạt là Softmax để đảm bảo phân phối về tỷ lệ của các lớp dự đoán.

### 2.2.3.2. Kết quả thực nghiệm

Chúng tôi thực hiện huấn luyện tuần tự từng Model với số epochs, batch\_size, validation batch\_size lần lượt là: 50, 32, 32. Kết quả thực nghiệm thu được như sau:

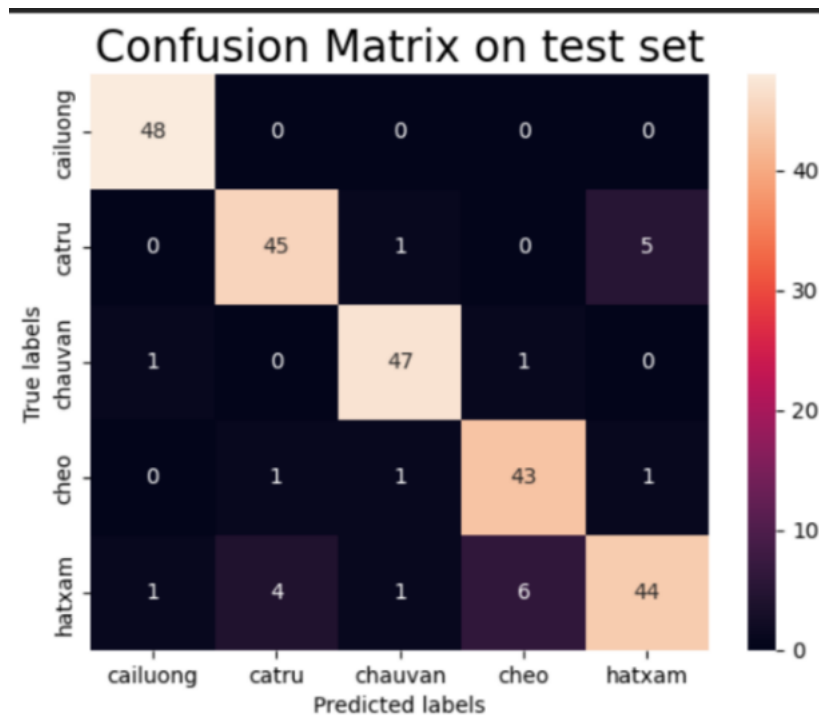


Hình 2.11 Training history mô hình CNN thứ 3

Đến với quá trình huấn luyện và kiểm thử của mô hình thứ 3, kết quả được cho ra cải thiện rõ rệt so với hai mô hình trước tuy hội tụ Loss chậm hơn, Accuracy tăng chậm hơn. Các giá trị kiểm thử đã thể hiện rõ ràng sự khác biệt này khi có sự tương đồng giữa 2 đường của tập train và tập test trong xuyên suốt các epoch.

	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

Hình 2.12a Bảng hiệu suất mô hình CNN thứ 3



Hình 2.12b Bảng Confusion Matrix của mô hình CNN thứ 3

Mô hình CNN thứ 3 có kết quả tốt nhất trên tất cả các mô hình. Biểu đồ Loss có phần ổn định và đạt giá trị thấp nhất trên cả 3 mô hình, tương tự với Accuracy (Hình 2.10). Các chỉ số đánh giá của mô hình 3 đạt 0.91 và là tốt nhất trong cả 3 mô hình CNN đã được thử nghiệm. Biểu đồ Confusion Matrix khá tương đồng với mô hình 1 khi “Hát xẩm” vẫn là lớp bị đánh nhãn sai nhiều nhất (Hình 2.11a và 2.11b).

## 2.2.4. Thí nghiệm 4: Tiến hành thực nghiệm trên mô hình kết hợp

### 2.2.4.1. Kiến trúc mô hình cơ sở

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.

Mô hình trên được chúng tôi gọi là Combined Model, với công thức được biểu diễn hóa như sau:

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

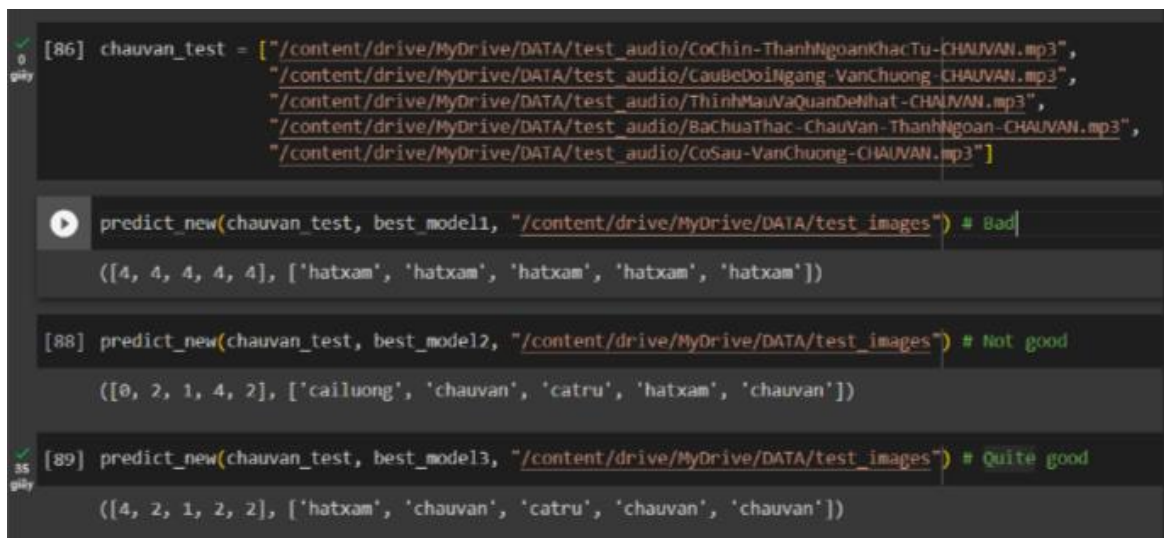
$$\bar{y} = \operatorname{argmax}(P_{\text{prod}}) = \operatorname{argmax}(p_1, p_2, \dots, p_c)$$

Trong đó:

- S: là số lớp cần phân loại
- C: là số model kết hợp lại

#### 2.2.4.2. Inferences

Tại sao cần sử dụng mô hình kết hợp? Lý do là bởi vì trong quá trình Inference, chúng tôi đã kiểm chứng với 1 bộ dữ liệu mới hoàn toàn, gồm 24 bài nhạc chia đều cho mỗi thể loại. Khi thực nghiệm kiểm tra trên mỗi 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.



```
[86] chauvan_test = ["/content/drive/MyDrive/DATA/test_audio/CoChin-ThanhNgoanKhacTu-CHAUVAN.mp3",
"/content/drive/MyDrive/DATA/test_audio/CauBeDoiNgang-VanChuong-CHAUVAN.mp3",
"/content/drive/MyDrive/DATA/test_audio/ThinhMauVaQuanDeNhat-CHAUVAN.mp3",
"/content/drive/MyDrive/DATA/test_audio/BaChuaThac-ChauVan-ThanhNgoan-CHAUVAN.mp3",
"/content/drive/MyDrive/DATA/test_audio/CoSau-VanChuong-CHAUVAN.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'])

[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'])
```

Hình 2.13 Kết quả của cả 3 model trên dữ liệu mới

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

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

Hình 2.14 Ví dụ về cách tính toán của mô hình kết hợp

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). Với kiến trúc mô hình kết hợp với nguyên tắc bỏ phiếu của ba mô hình 1,2 và 3. Kết quả cho ra cho thấy với 5 file âm thanh thể loại “Châu Văn”, mô hình kết hợp đã dự đoán đúng hoàn toàn.

```
chauvan_test = ["/content/drive/MyDrive/DATA/test_audio/CoChin-ThanhNgaoKhacTu-CHAUVAN.mp3",
                 "/content/drive/MyDrive/DATA/test_audio/CauBeDoiNgang-VanChuong-CHAUVAN.mp3",
                 "/content/drive/MyDrive/DATA/test_audio/ThinhMauVaQuanDeNhat-CHAUVAN.mp3",
                 "/content/drive/MyDrive/DATA/test_audio/BaChuaThac-ChauVan-ThanhNgao-CHAUVAN.",
                 "/content/drive/MyDrive/DATA/test_audio/CoSau-VanChuong-CHAUVAN.mp3"]

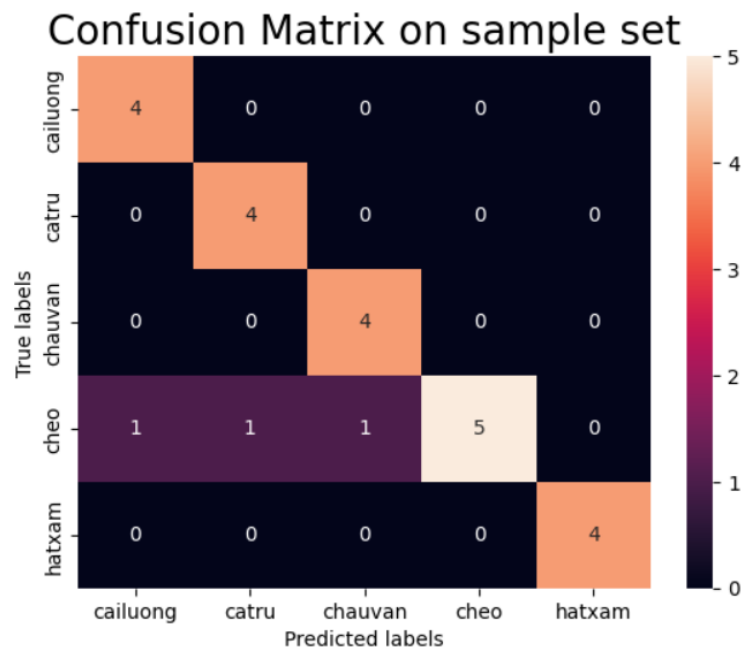
prod_pred, prod_class = PROD_predict(chauvan_test, "/content/drive/MyDrive/DATA/test_images",
                                     best_model1, best_model2, best_model3)

prod_pred, prod_class

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

Hình 2.15 Kết quả mô hình kết hợp trên dữ liệu mới

Confusion matrix của mô hình kết hợp trên bộ dữ liệu mới:



Hình 2.16 Confusion matrix của mô hình kết hợp trên bộ dữ liệu mới

Biểu đồ Confusion Matrix của mô hình kết hợp cho ra kết quả rất tốt trên 24 bài nhạc mới. Chỉ có kết quả dự đoán sai ba bài (1 bài Cải Lương dự đoán thành Chèo, 1 bài Ca Trù dự đoán thành Chèo, 1 bài Châu Văn dự đoán thành chèo).

## CHƯƠNG 3: XÂY DỰNG HỆ THỐNG PHÂN LOẠI ÂM NHẠC

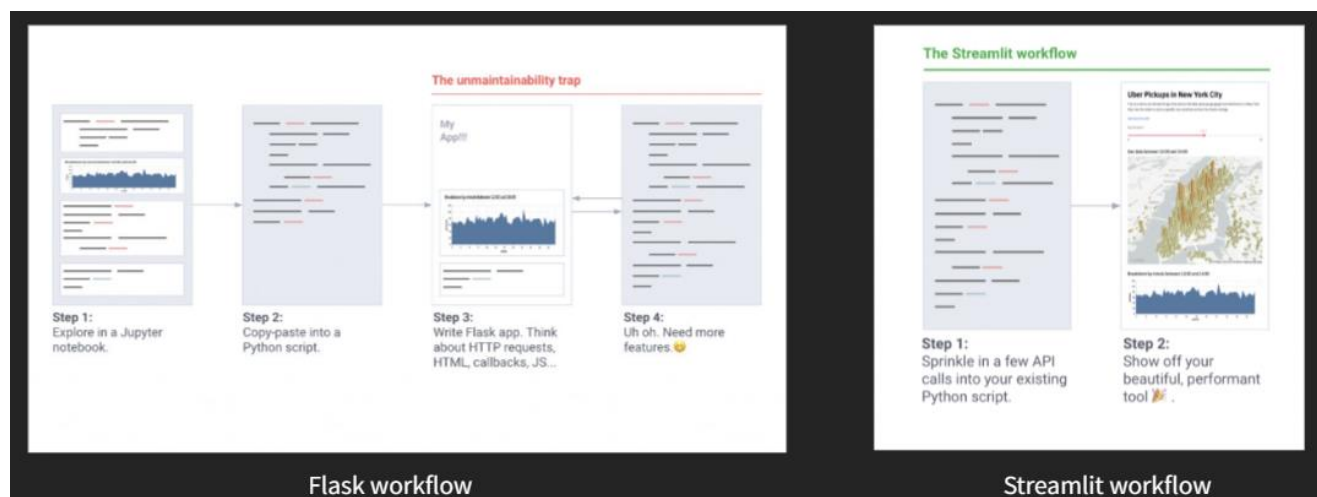
Để có một hệ thống phân loại âm nhạc theo thể loại hiệu quả và trải nghiệm người dùng thú vị, hệ thống sẽ là một Web App và phải được thiết kế để tạo các tương tác tự nhiên nhất có thể; và điều này yêu cầu các mô hình học máy có thể hiểu được âm nhạc.

Hiện nay có rất ít các website phân loại thể loại âm nhạc và đặc biệt hơn là thể loại âm nhạc truyền thống Việt Nam. Vì thế chúng tôi quyết định xây dựng hệ thống này mang lại nhiều ý nghĩa quan trọng về cả mặt bảo tồn và phát triển văn hóa. Ở chương này chúng tôi sẽ đề cập các bước chi tiết để xây dựng hệ thống Web App.

### 3.1. StreamLit

StreamLit là một framework trẻ nhưng lại nhận được rất nhiều quan tâm của cộng đồng Machine Learning. Lý do là vì tính nhanh chóng là tiện lợi của nó.

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:

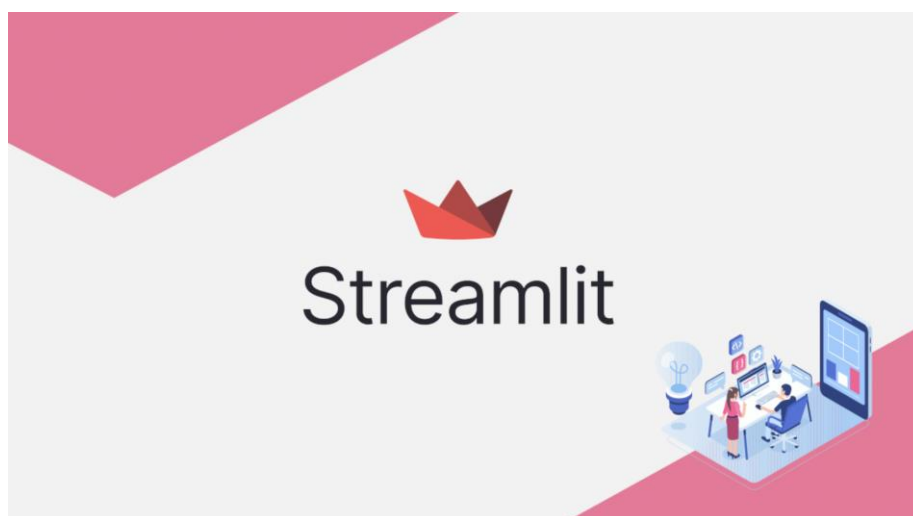


Hình 3.1 So sánh giữa hai framework

Quan sát hình trên, cùng với một bước để deploy model lên hệ thống, StreamLit tốn ít công đoạn hơn rất nhiều so với Flask. 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.





Hình 3.2 Framework StreamLit

### 3.2. Phân tích thiết kế hệ thống

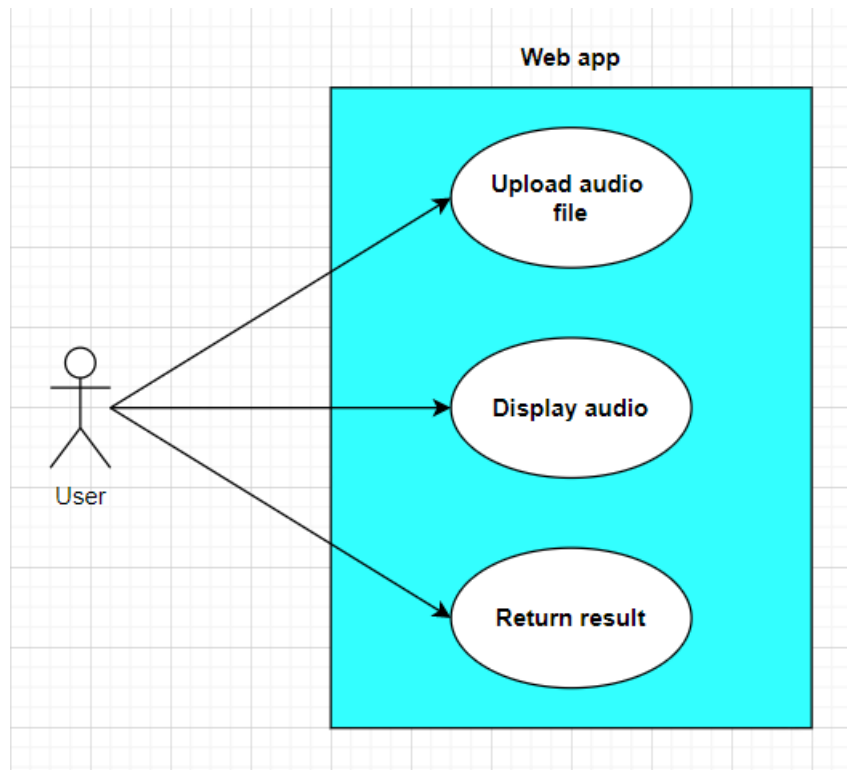
Sau khi đã train cả 3 model CNN trên bộ dataset Vietnam Traditional Music (5 genres) và thu được file pre-trained của cả ba mô hình, thì bây giờ chúng tôi muốn xây dựng một web-app demo để người dùng tiếp cận dễ dàng hơn khi muốn xem thử thể loại của bài hát truyền thống đó thuộc vào thể loại nào.

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.

#### 3.2.1. Phân tích khảo sát

Để xây dựng một Web App Demo hoàn thiện nhất, chúng tôi quyết định khảo sát các Web về “Phân loại âm nhạc về thể loại” khác họ làm như thế nào.

Sau khi phân tích khảo sát, chúng tôi rút được một số kết luận như sau: Có website cho phép người dùng dán đường dẫn vào thanh tìm kiếm để phân loại thể loại nhạc của bài hát trên youtube đó. Có website cho phép người dùng upload file bài hát để phân loại bài hát đó. Vì vậy, với những rút kết đó chúng tôi quyết định xây dựng Website hệ thống của chúng tôi như sau:

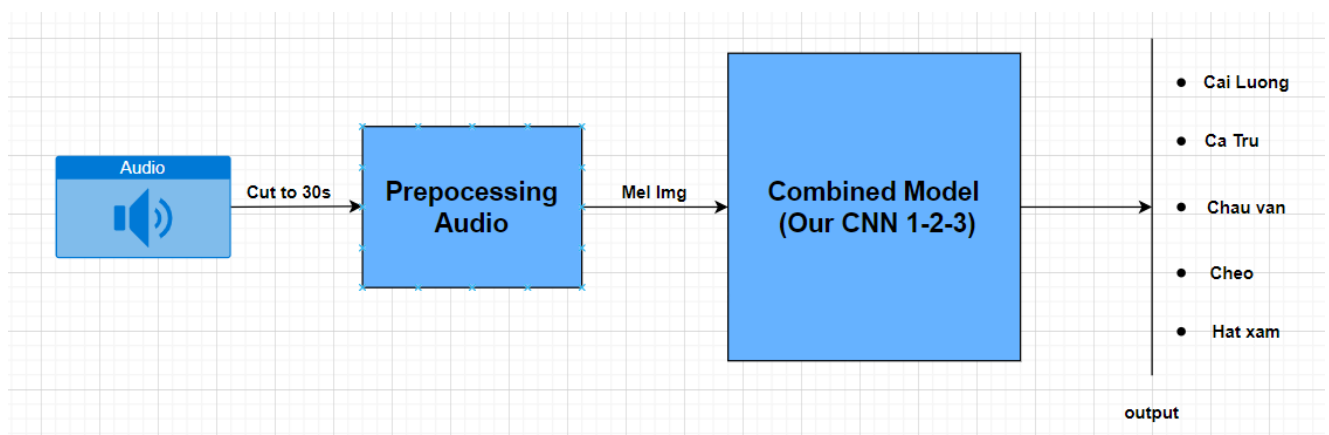


Hình 3.3 Sơ đồ Use-case hệ thống

Hệ thống bao gồm ba phạm vi kiến thức cung cấp chính như sau:

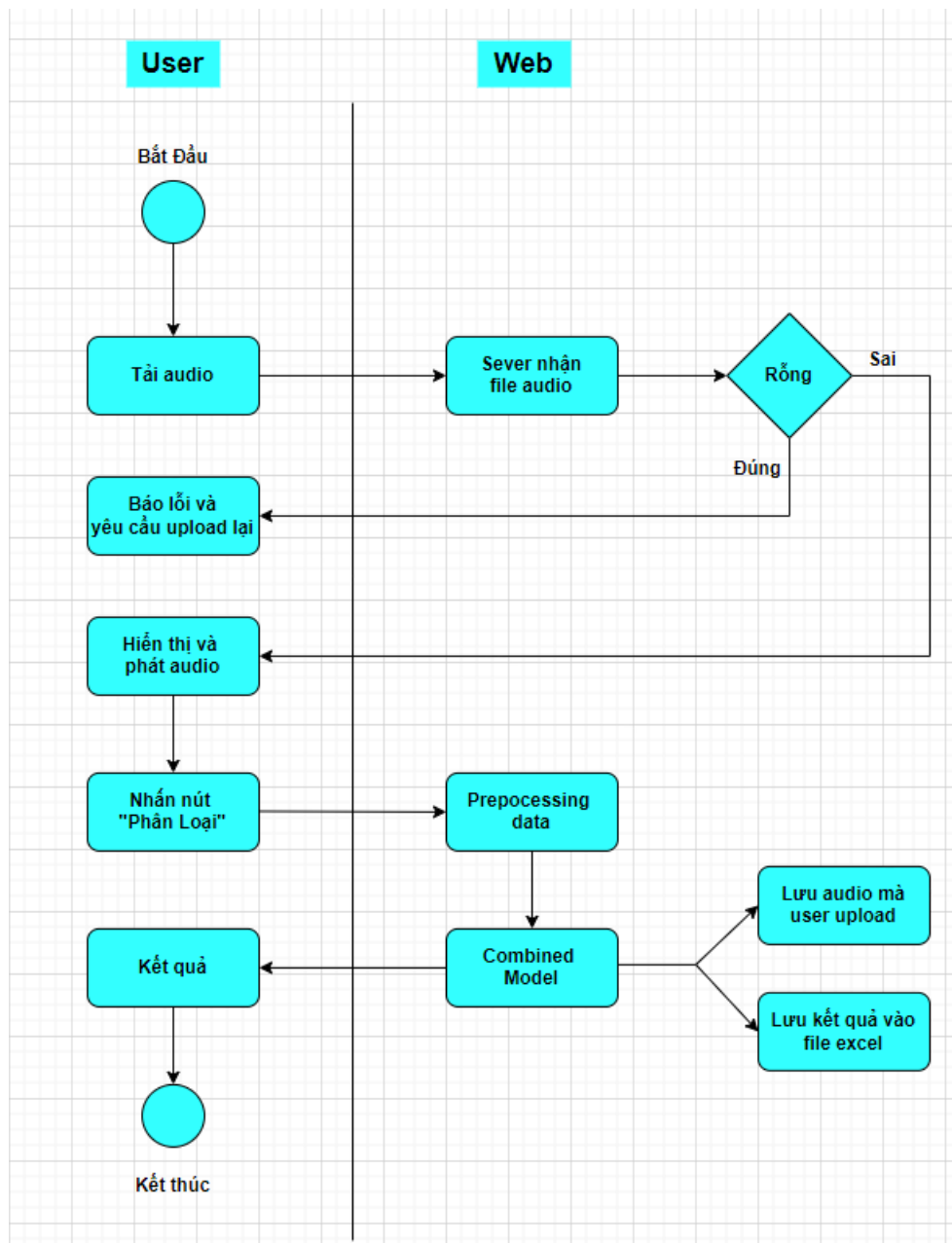
- Upload audio file: Cho phép người dùng tải một hoặc nhiều bài hát lên để phân loại
- Display audio: Cho phép người dùng nghe các bài nhạc mà họ đã tải lên
- Return result: Trả về kết quả phân loại thuộc các lớp có sẵn trong bộ dữ liệu

Từ đó, hệ thống của chúng tôi sẽ tuân theo tuần tự như sau:



Hình 3.4 Pipeline hệ thống

Workflow các bước xử của Web app demo có thể được thể hiện rõ qua Activity Diagram như hình sau:



Hình 3.5 Activity Diagram hệ thống

Ưu điểm của kiến workflow như hình trên:

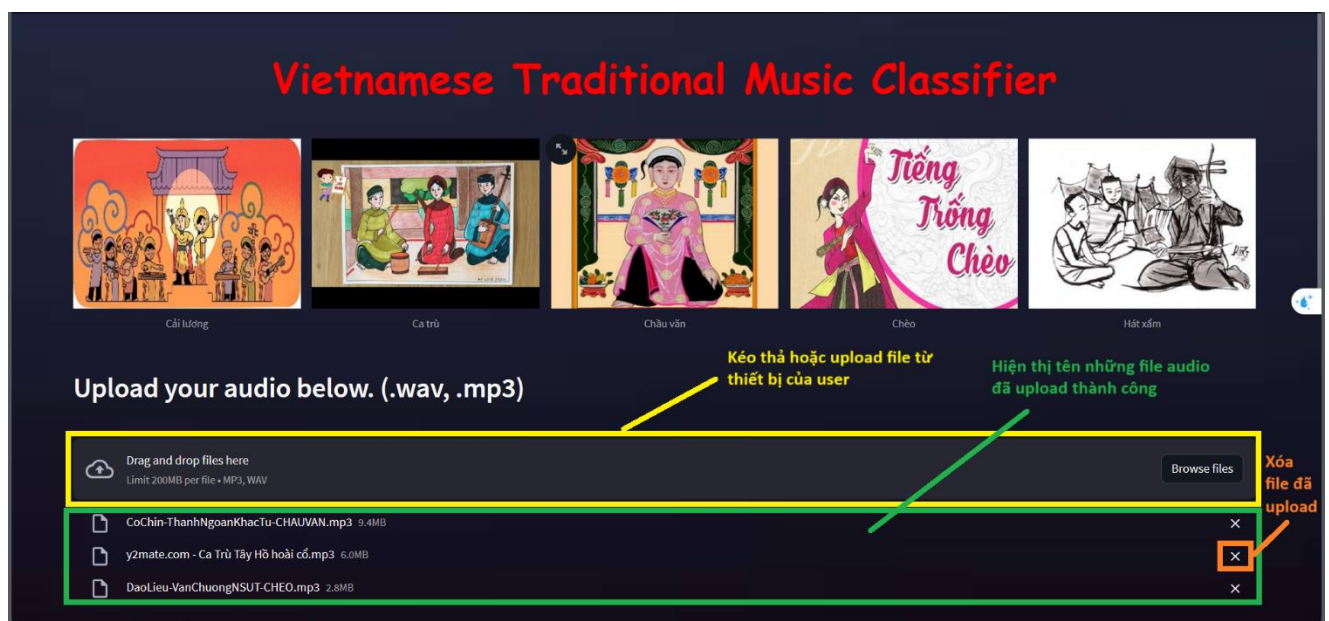
- Dễ dàng tiếp cận
- Trình tự xử lý rõ ràng

- Không phụ thuộc môi trường, nền tảng xây dựng
- Chức năng riêng biệt, dễ dàng quản lý, dễ cải tiến

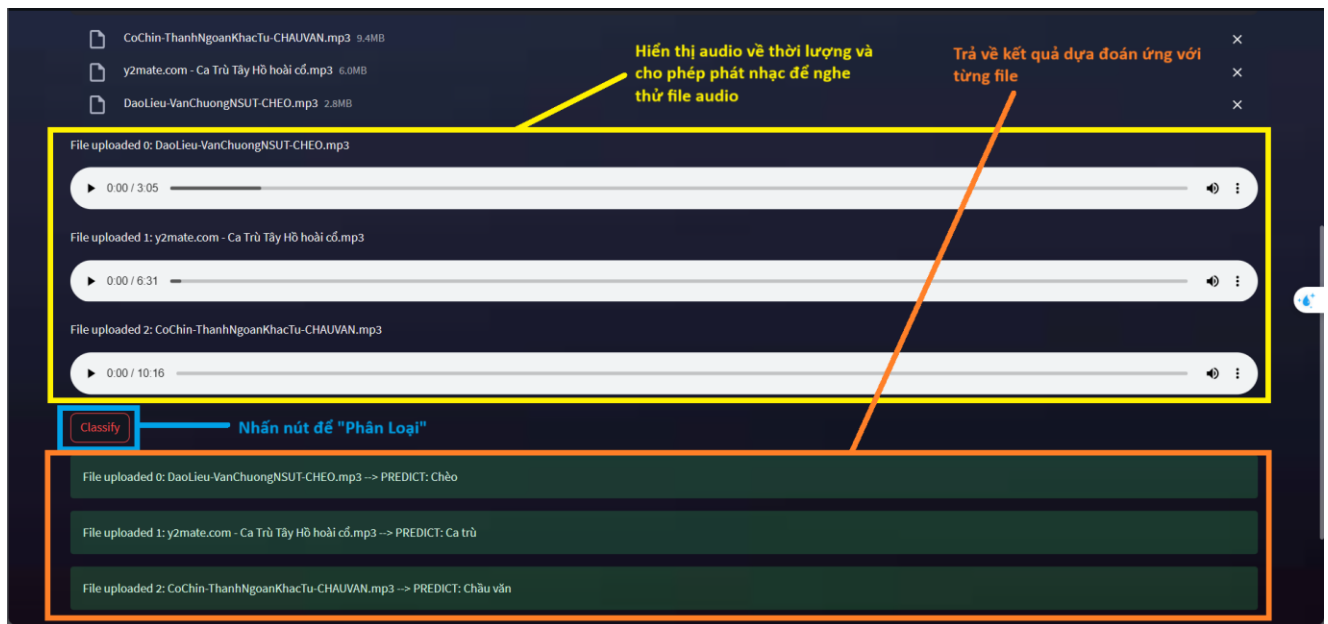
### 3.2.2. Thiết kế giao diện

Hệ thống bao gồm ba chức năng chính như sau:

- Upload audio file: Cho phép người dùng tải một hoặc nhiều bài hát lên để phân loại
- Display audio: Cho phép người dùng nghe các bài nhạc mà họ đã tải lên
- Return result: Trả về kết quả phân loại thuộc các lớp có sẵn trong bộ dữ liệu

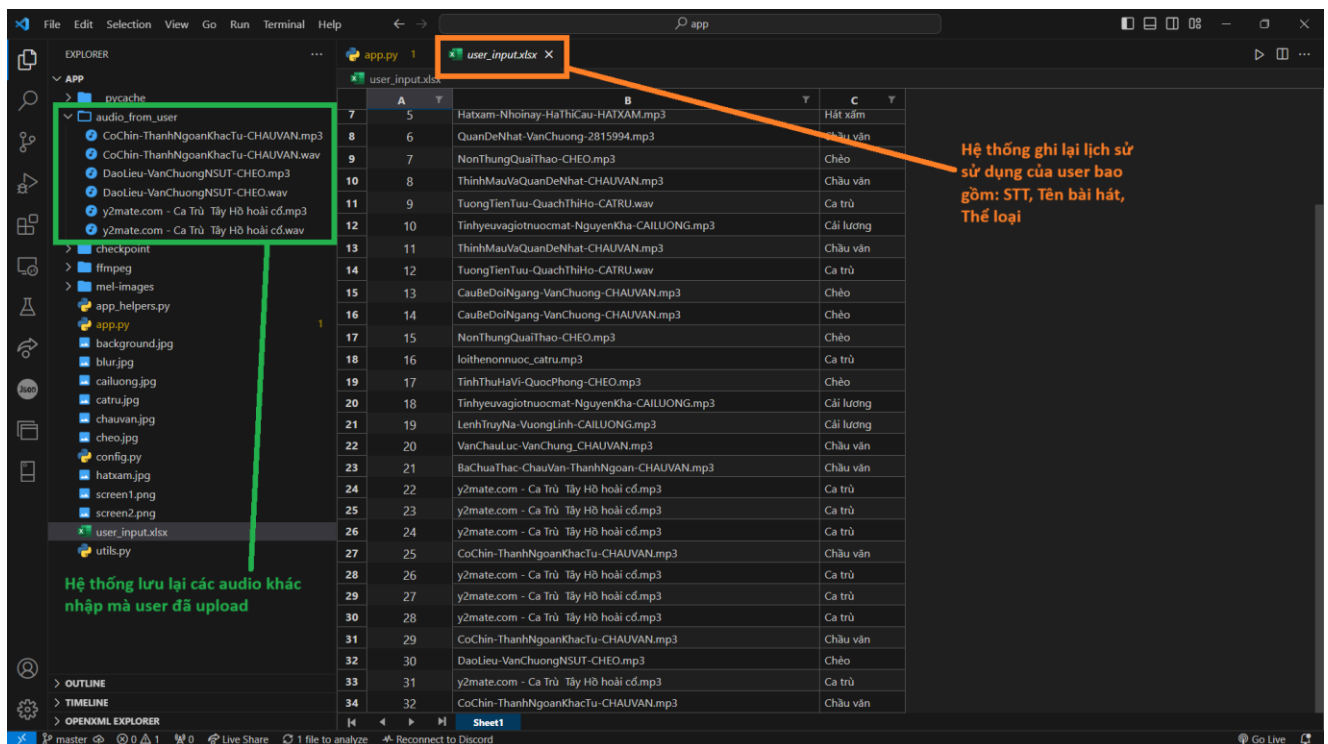


Hình 3.6 Màn hình giao diện 1



Hình 3.7 Màn hình giao diện 2

Ngoài ra, để lưu lại các bài hát mà user đã tải lên để mở rộng cơ sở dữ liệu các phục vụ cho các mục đích tương lai. Hay để ghi lại lịch sử hoạt động trên Website của user để dễ dàng quản lý, chúng đã thiết kế hệ thống của chúng tôi như sau:



Hình 3.8 Quản lý hệ thống

### 3.3. Kiểm thử hệ thống

Để tiến hành thử nghiệm ứng dụng ta cần xây dựng môi trường thực nghiệm và đánh giá kết quả thực nghiệm.

#### 3.3.1. Môi trường thực nghiệm

Môi trường tiến hành thử nghiệm như sau:

- Hệ điều hành: Windows 11, 64 bit, 8 Gb ram, RTX 3050
- Python: 3.7.0 và Pytorch 1.1.0+ hoặc Tensorflow 2.0+

#### 3.3.2. Kết quả thực nghiệm

Để kiểm chứng kết quả dự đoán của Combined Model, tui 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.

	A	B	C
1		Filename	Genre
2	0	cailuong(8).mp3	Cải lương
3	1	cailuong(10).mp3	Cải lương
4	2	cailuong(2).mp3	Cải lương
5	3	cailuong(3).mp3	Cải lương
6	4	cailuong(9).mp3	Cải lương
7	5	cailuong(1).mp3	Cải lương
8	6	cailuong(6).mp3	Cải lương
9	7	cailuong(4).mp3	Cải lương
10	8	cailuong(7).mp3	Cải lương
11	9	cailuong(5).mp3	Cải lương
12	10	catru(1).mp3	Ca trù
13	11	catru(2).mp3	Ca trù
14	12	catru(5).mp3	Ca trù
15	13	catru(4).mp3	Ca trù
16	14	catru(3).mp3	Ca trù
17	15	catru(10).mp3	Ca trù
18	16	catru(6).mp3	Ca trù
19	17	catru(8).mp3	Ca trù
20	18	catru(7).mp3	Ca trù
21	19	catru(9).mp3	Ca trù
22	20	y2mate.com - Hát Văn Tình Cha NSUT	Chèo
23	21	CoSau-VanChuong-CHAUVAN.mp3	Chầu văn
24	22	CoChin-ThanhNgoanKhacTu-CHAUVAN	Chầu văn
25	23	y2mate.com - Vu lan báo hiếu hoài th	Ca trù
26	24	y2mate.com - Văn Hát Lễ Ông Hoàng B	Ca trù
27	25	VanChauLuc-VanChung_CHAUVAN.mp	Chầu văn
28	26	y2mate.com - Bạn sẽ phải tiếc nuối n	Chầu văn
29	27	ThinhMauVaQuanDeNhat-CHAUVAN.n	Chầu văn
30	28	y2mate.com - Chầu Bé Bắc Lệ Bản Đặc	Chầu văn

Hình 3.9 Quản lý một số dữ liệu về các thể loại nhạc truyền thống tự thu thập

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
- Châu 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

## KẾT LUẬN

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 Phân loại âm nhạc truyền thống Việt Nam (5 thể loại) và đạt được thành tích tốt về độ chính xác (đặc biệt là với kiến trúc PROD - combined model). Độ chính xác của mô hình PROD này đạt được là việc kết sự bỏ phiếu giữa cả ba mô hình 1, mô hình 2 và mô hình 3 khi chúng đã được huấn luyện riêng lẻ. Điều này có nghĩa là nếu chúng ta kết hợp kết quả dự đoán của nhiều mô hình đã được huấn luyện trên tập dữ liệu rồi thì kết quả sẽ có xu hướng tăng cao và hiệu suất sẽ tốt hơn.

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.

Thuật toán	Precision	Recall	F1-Score	Accuracy
CNN 1	0.89	0.89	0.89	0.89
CNN 2	0.84	0.86	0.85	0.84
CNN 3	0.91	0.91	0.91	0.91

Hình 3.10 Bảng kết quả thực nghiệm

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.

Ngoài những kết quả khả quan đạt được, mô hình vẫn còn những hạn chế như sau:

- Điểm hạn chế đầu tiên có thể kể đến là do đến từ chính kiến trúc mô hình của chúng tôi. Các mô hình chỉ đơn thuần là rút trích đặc trưng của các bước ảnh về tần số âm thanh chứ chưa thực sự hiểu được “giai điệu” của âm thanh. Điều này có thể ảnh hưởng không nhỏ đến kết quả dự đoán của mô hình cho toàn bộ quá trình.



- Mô hình chưa hoàn thiện do chưa thể phân biệt được các đặc điểm phức tạp để phân biệt các thể loại nhạc (châu văn và hát xẩm bị nhầm lẫn với ca trù).
- 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) .
- Các thể loại nhạc có sự tương đồng một phần trong việc sử dụng nhạc cụ, giai điệu (phân lớn với các thể loại có xuất xứ từ miền bắc).
- 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).

Về nhóm chúng tôi, chúng tôi đã được tìm hiểu và học được những điều như sau:

- Các tiền xử lý và hậu xử lý dữ liệu âm thanh.
- 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.

### **Định hướng nghiên cứu tiếp theo:**

Từ hướng nghiên cứu đã trình bày rõ ràng ở trên, mô hình chúng tôi có thể phát triển theo các hướng sau:

Thứ nhất đó là có thể mở rộng bộ dữ liệu. Bộ dữ liệu càng lớn và với mô hình có kiến trúc tốt, càng nhiều dữ liệu cùng đồng nghĩa với việc hiệu suất và chất lượng của quá trình huấn luyện tăng lên. Tuy nhiên với bộ dữ liệu lớn thì cũng cần phải đảm bảo về tính cân bằng của toàn bộ dữ liệu nếu không thì ta sẽ không đạt được kết quả như mong được.

Có thể áp dụng các kỹ thuật khác như speech-to-text, hay giúp mô hình có thể hiểu được ý

nghe của đoạn âm thanh thì mô hình sẽ có nhiều cơ sở hơn để phân tích nội dung bài hát, từ đó có thể cho ra hiệu suất tốt hơn.

Cuối cùng là định hướng về phát triển ứng dụng, có thể sử dụng các kỹ thuật khác để làm nhẹ mô hình hơn và tích hợp vào điện thoại để làm một Mobile App.

## TÀI LIỆU THAM KHẢO

1. Dataset: Vietnam Traditional Music (5 genres),  
<https://www.kaggle.com/datasets/homata123/vntm-for-building-model-5-genres>.
2. TensorFlow, <https://www.tensorflow.org/>
3. StreamLit, <https://streamlit.io/>
4. K. Choi, G. Fazekas, and M. Sandler. Explaining deep convolutional neural networks on music classification. arXiv preprint arXiv:1607.02444, 2016.
5. Koutini, K., Eghbal-Zadeh, H., Haunschmid, V., Primus, P., Chowdhury, S., & Widmer, G. (2020). Receptive-Field Regularized CNNs for Music Classification and Tagging. arXiv preprint arXiv:2007.13503.
6. Choi, K., Fazekas, G., Sandler, M., & Cho, K. (2016). Convolutional Recurrent Neural Networks for Music Classification. arXiv preprint arXiv:1609.04243.
7. Lee, J., & Nam, J. (2017). Multi-Level and Multi-Scale Feature Aggregation Using Sample-level Deep Convolutional Neural Networks for Music Classification. arXiv preprint arXiv:1706.06810.
8. Ding, Y., & Lerch, A. (2023). Audio Embeddings as Teachers for Music Classification. arXiv preprint arXiv:2306.17424.
9. Wu, S., Yu, D., Tan, X., & Sun, M. (2023). CLaMP: Contrastive Language-Music Pre-training for Cross-Modal Symbolic Music Information Retrieval. arXiv preprint arXiv:2304.11029.
10. Hernandez-Olivan, C., Rubio Llamas, S., & Beltran, J. R. (2023). Symbolic Music Structure Analysis with Graph Representations and Change-point Detection Methods. arXiv preprint arXiv:2303.13881.
11. Doh, S., Won, M., Choi, K., & Nam, J. (2022). Toward Universal Text-to-Music Retrieval. arXiv preprint arXiv:2211.14558.

12. Spijkervet, J., & Burgoyne, J. A. (2021). Contrastive Learning of Musical Representations. arXiv preprint arXiv:2103.09410.
13. Won, M., Spijkervet, J., & Choi, K. (2021). Music Classification: Beyond Supervised Learning, Towards Real-world Applications. Zenodo. doi: 10.5281/ZENODO.5703779. Available at: <https://zenodo.org/record/5703779>.