HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG

Nguyễn Tất Hậu

NGHIÊN CỨU XÂY DỰNG MÔ HÌNH PHÂN LOẠI GIỚI TÍNH VÀ VÙNG MIỀN CHO TIẾNG NÓI TIẾNG VIỆT DỰA TRÊN ÂM THANH

LUẬN VĂN THẠC SĨ KỸ THUẬT

(***Theo định hướng ứng dụng***)

HÀ NỘI - NĂM 2021

HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG

Nguyễn Tất Hậu

NGHIÊN CỨU XÂY DỰNG MÔ HÌNH PHÂN LOẠI GIỚI TÍNH VÀ VÙNG MIỀN CHO TIẾNG NÓI TIẾNG VIỆT DỰA TRÊN ÂM THANH

Chuyên ngành: Khoa học máy tính

Mã số: 8.48.01.01

LUẬN VĂN THẠC SĨ KỸ THUẬT

(***Theo định hướng ứng dụng***)

NGƯỜI HƯỚNG DẪN KHOA HỌC: TS. NGUYỄN NGỌC ĐIỆP

HÀ NỘI - NĂM 2021

LỜI CAM ĐOAN

Tôi cam đoan đây là công trình nghiên cứu của riêng tôi. Nội dung của luận văn có tham khảo và sử dụng các tài liệu, thông tin được đăng tải trên những tạp chí khoa học và các trang web được liệt kê trong danh mục tài liệu tham khảo. Tất cả các tài liệu tham khảo đều có xuất xứ rõ ràng và được trích dẫn hợp pháp.

Tôi xin hoàn toàn chịu trách nhiệm và chịu mọi hình thức kỷ luật theo quy định cho lời cam đoan của mình.

Hà nội, ngày … tháng … năm 2021

Tác giả luận văn

Nguyễn Tất Hậu

MỤC LỤC

DANH MỤC CÁC THUẬT NGỮ TẮT

Viết tắt	Tiếng Anh	Tiếng Việt
SVM	Support vector machine	Máy véc tơ hỗ trợ
AI	Artificial Intelligence	Trí tuệ nhân tạo
RNN	Recurrent Neural Network	Mạng nơ-ron hồi quy
CNN	Convolutional Neural Network	Mạng nơ-ron tích chập
FC	Fully Connected Layer	Lớp kết nối đầy đủ
STFT	Short-time Fourier Transform	Phép biến đổi Fourier thời gian ngắn
DL	Deep Learning	Học sâu
MFCC	Mel-Frequency Cepstral Coefficients	
SIFT	Scale Invariant Feature Transform	
SURF	Speeded-Up Robust Features	
HOG	Histogram of Oriented Gradients	

DANH MỤC CÁC BẢNG

DANH MỤC CÁC HÌNH

MỞ ĐẦU

1. Tính cấp thiết của đề tài

Trong những năm gần đây, các bước tiến trong lĩnh vực Học sâu (Học sâu), nhất là về Thị giác Máy tính đã giải quyết rất nhiều vấn đề từ các lĩnh vực khác nhau và đóng góp vào sự cải thiện đời sống hàng ngày của con người.

Ngày nay, để tăng tính bảo mật và xác thực, nhiều kĩ thuật liên quan đến nhận diện giọng nói, giới tính đang áp dụng trong các ngân hàng, cơ quan và tổ chức doanh nghiệp, riêng trong viễn thông vẫn đề về gian lận cước đang gây ra những tổn hại về doanh thu rất lớn cho các nhà mạng, tập đoàn viễn thông. Việc tìm bắt những thuê bao lậu cước là một thử thách rất lớn vì các hệ thống lậu được lập trình một cách rất tinh vi để hành vi của chúng trở nên vô cùng khó phân biệt với các thuê bao của người dùng thông thường. Các tập đoàn viễn thông đang thử nghiệm theo dõi các yếu tố mà các nhóm làm lậu cước khó tác động vào nhằm phát hiện ra các bất thường, một trong số đó giọng nói thu được từ các cuộc gọi. Vì các thuê bao lậu cước là các thuê bao mà từ đó nhiều người dùng gọi đi, nên giọng nói sẽ thay đổi theo từng người, sự thay đổi này có thể được thấy trong giới tính và giọng vùng miền của người nói. Như vậy, nếu như có một cách để tự động chỉ ra giới tính và giọng vùng miền của người nói thì chúng ta có thể phần nào phát hiện ra các bất thường.

Hiện nay, lĩnh vực xử lý âm thanh – mà chủ yếu là các bài toán phân loại âm thanh đã tận dụng nhiều từ các kĩ thuật mà được sử dụng nhiều trong lĩnh vực Thị giác Máy tính và xử lý hình ảnh.

Xuất phát từ thực tế và mục tiêu như trên, học viên với sự giúp đỡ của TS. Nguyễn Ngọc Điệp học viên lựa chọn thực hiện đề tài luận văn tốt nghiệp chương trình đào tạo thạc sĩ có tên “Nghiên cứu xây dựng mô hình phân loại giới tính và vùng miền cho tiếng nói tiếng Việt dựa trên âm thanh”.

2. Tổng quan vấn đề cần nghiên cứu

Hiện nay đã có rất nhiều bài toán phân loại dữ liệu âm thanh sử dụng các mô hình học sâu, từ các cuộc thi Khoa học dữ liệu của Kaggle như "TensorFlow Speech Recoginition Challenge" (2017), "Freesound Tagging" (2018, 2019) trong đó người tham gia tận dụng việc đưa dữ liệu âm thanh về dạng biểu diễn tần số (được coi là "ảnh của âm thanh") và áp dụng các kĩ thuật về xử lý ảnh, huấn luyện mạng nơ-ron để giải quyết bài toán, trong đó dữ liệu âm thanh sẽ được chuyển sang bài toán phân loại ảnh phổ.

Một số kỹ thuật phân loại âm thanh khác đã được nghiên cứu và chứng minh tính khả thi và độ chính xác cao bằng cách sử dụng mô hình GMM (Gaussian mixture models), kết hợp với việc xây dựng tập thuộc tính âm thanh (Mel-frequency cepstrum coefficients)

Vào tháng 8/2018, ZALO Inc. cũng đã tổ chức một cuộc thi phân loại giới tính và vùng miền cho tiếng nói tiếng Việt với một bộ dữ liệu khá lớn và đa dạng. Lời giải của đội chiến thắng cũng là sử dụng phương pháp kết hợp trích xuất thuộc tính của âm thanh và ứng dụng các thuật toán Học sâu. Vì vậy nội dung của luận văn này sẽ tập trung nghiên cứu giải pháp để xây dựng được một bộ phân loại với độ chính xác cao hơn và tốc độ xử lý nhanh hơn. Để làm được việc đó, ngoài việc tận dụng các mạng nơ-ron tiên tiến có khả năng phân loại, luận văn sẽ nghiên cứu thêm về các kĩ thuật sử dụng trong từng phần của của xử lý phân loại, ví dụ như kĩ thuật xử lý và đưa âm thanh về miền tần số hay các kiến trúc mạng nơ-ron khác nhau, v.v nhằm nâng cao hơn nữa độ chính xác. Ngoài ra, luận văn cũng sẽ sử dụng thêm bộ data giọng nói VIVOS để huấn luyện mô hình chứ không chỉ bộ data của ZALO, vì bộ VIVOS này được thu trong điều kiện lý tưởng hơn, do đó dễ dàng cho mô hình hơn trong giai đoạn đầu của việc học.

3. Mục đích nghiên cứu

- Tìm hiểu về các phương pháp xử lý dữ liệu giọng nói;

- Nghiên cứu phương pháp xây dựng mô hình học sâu phân loại giới tính và vùng miền của giọng nói;
- Rèn luyện phương pháp và khả năng nghiên cứu.

4. Đối tượng và phạm vi nghiên cứu

- Đối tượng nghiên cứu: Tiếng nói Tiếng Việt được thu trong môi trường có tạp âm, đa dạng về độ tuổi, cảm xúc của người nói.
- Phạm vi nghiên cứu: Mô hình phân loại giới tính (nam – nữ) và vùng miền (bắc – trung - nam) từ giọng nói.

5. Phương pháp nghiên cứu

Dựa trên cơ sở lý thuyết của xử lý dữ liệu giọng nói và các phương pháp huấn luyện mô hình học sâu phân loại âm thanh nói chung và giới tính - vùng miền của giọng nói nói riêng.

Cấu trúc nội dung luận văn gồm 3 chương với các nội dung như sau:

Chương 1: Tổng quan về các phương pháp phân loại âm thanh

Nội dung chương 1 của luận văn sẽ trình bày tổng quan về các phương pháp phân loại âm thanh ứng dụng các mô hình học từ dữ liệu, từ phương pháp dùng mô hình học máy truyền thống đến các mô hình học sâu. Ưu nhược điểm của từng phương pháp sẽ được phân tích để lựa chọn ra phương pháp phù hợp cho bài toán đang cần giải quyết.

Chương 2: Một số phương pháp về phân loại âm thanh

Nội dung chương 2 của luận văn sẽ trình bày các cơ sở lý thuyết liên quan đến phương pháp sử dụng mô hình học sâu CNN trên dữ liệu dạng biểu diễn tần số của âm thanh. Cụ thể, phần đầu tiên của chương này sẽ tóm lược các một vài phương pháp chuyển dữ liệu âm thanh từ miền thời gian sang miền tần số để thu được "ảnh của âm thanh". Tiếp đến, một số mô hình học sâu quan trọng, thường dùng trong các bài toán phân loại hình ảnh sẽ được trình bày.

Chương 3: Thực nghiệm và đánh giá

Nội dung chương 3 của luận văn sẽ trình bày các bước triển khai tiền xử lý dữ liệu và xây dựng, huấn luyện mô hình cũng như các bước hậu xử lý, sau cùng là đánh giá độ chính xác của mô hình trên dữ liệu mới.

Kết luận.

CHƯƠNG 1: TỔNG QUAN VỀ CÁC PHƯƠNG PHÁP PHÂN LOẠI ÂM THANH

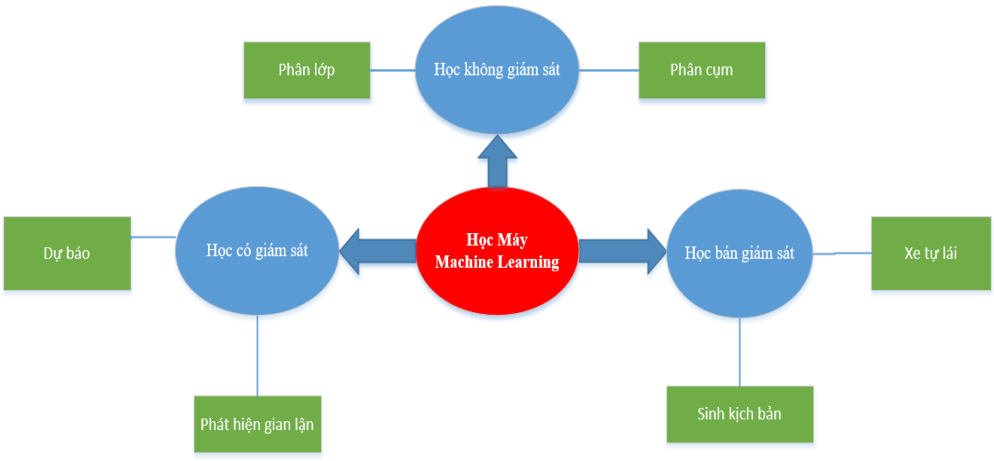
Tóm tắt chương: Chương 1 trình bày tổng quan về các phương pháp phân loại âm thanh ứng dụng các mô hình học từ dữ liệu, từ phương pháp dùng mô hình học máy truyền thống đến các mô hình học sâu. Ưu nhược điểm của từng phương pháp sẽ được phân tích để lựa chọn ra phương pháp phù hợp cho bài toán đang cần giải quyết.

1.1. Mô hình học máy truyền thống

1.1.1. Giới thiệu về học máy và các mô hình học máy truyền thống

Một trong những khác biệt chính giữa con người và máy tính là con người học hỏi từ những kinh nghiệm trong quá khứ, nhưng với máy tính hoặc máy móc cần được phải được thực hiện theo một quy trình có sẵn. Máy tính là những máy logic nghiêm ngặt với ý nghĩa thông thường. Điều đó có nghĩa là nếu chúng ta muốn máy làm điều gì đó, chúng ta phải cung cấp cho nó những quy trình và các hướng dẫn chi tiết, từng bước về chính xác những việc cần làm.

Vì vậy, con người đã viết nên các kịch bản và lập trình để máy tính làm theo các hướng dẫn và có khả năng tự học hỏi. Đó là cái cách mà học máy ra đời. Khái niệm máy học chính xác là việc máy tính học hỏi từ dữ liệu trong quá khứ và rút kinh nghiệm qua thời gian.



Hình 1.1: Phân loại các thuật toán học máy

❖ **Phân loại học máy**

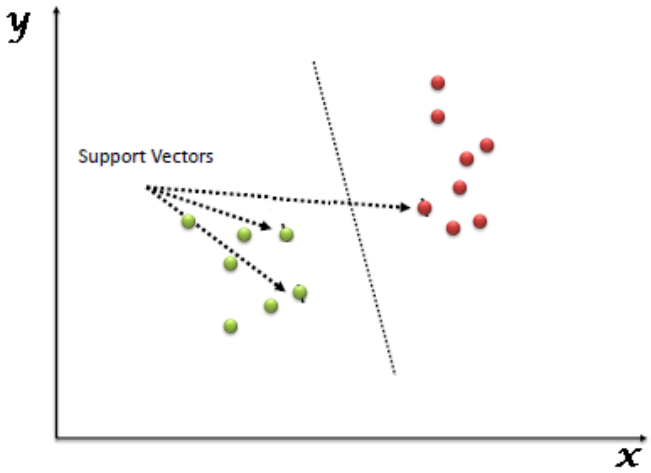
- Học máy được giám sát (*Supervised*): đây là các thuật toán áp dụng những gì đã được học trong quá khứ vào dữ liệu mới bằng cách sử dụng các ví dụ được gắn nhãn để dự đoán các sự kiện trong tương lai. Bắt đầu từ việc phân tích một tập dữ liệu huấn luyện đã biết, thuật toán học tạo ra một hàm được suy ra để đưa ra dự đoán về các giá trị đầu ra.
- Học máy không giám sát (*Unsupervised*): ngược lại, thuật toán học máy không giám sát được sử dụng khi thông tin được sử dụng để đào tạo không được phân loại cũng không được dán nhãn. Nghiên cứu học tập không giám sát làm thế nào các hệ thống có thể suy ra một chức năng để mô tả một cấu trúc ẩn từ dữ liệu không được gắn nhãn.
- Học máy bán giám sát (*Reinfocement*): các thuật toán học máy được giám sát bán nằm ở đâu đó giữa học tập có giám sát và không giám sát, vì chúng sử dụng cả dữ liệu được gắn nhãn và không nhãn cho đào tạo - thường là một lượng nhỏ dữ liệu được gắn nhãn và một lượng lớn dữ liệu không được gắn nhãn. Các hệ thống sử dụng phương pháp này có thể cải thiện đáng kể độ chính xác trong học tập.

1.1.2. Giới thiệu một số thuật toán học máy có giám sát

1.1.2.1. Phương pháp học có giám sát sử dụng SVM (SVM- Support vector machine)

Support Vector Machine (SVM) là một thuật toán thuộc nhóm Supervised Learning (học có giám sát) dùng để phân chia dữ liệu (Classification) thành các nhóm riêng biệt.

Trong thuật toán này, chúng ta vẽ đồ thị dữ liệu là các điểm trong n chiều (ở đây n là số lượng các tính năng bạn có) với giá trị của mỗi tính năng sẽ là một phần liên kết. Sau đó chúng ta thực hiện tìm "đường bay" (*hyper-plane*) phân chia các lớp. Hyper-plane nó chỉ hiểu đơn giản là 1 đường thẳng có thể phân chia các lớp ra thành hai phần riêng biệt.



Hình 1.2: Mô hình phân loại SVM
(Nguồn: Tìm hiểu về Support Vector Machine (SVM) – Internet)

Support Vectors hiểu một cách đơn giản là các đối tượng trên đồ thị tọa độ quan sát, *Support Vector Machine* là một biên giới để chia hai lớp tốt nhất.

1.1.2.2. Phương pháp học có giám sát sử dụng cây quyết định (Decision Tree)

Decision Tree- cây quyết định là một mô hình được đánh giá cao trong việc phân lớp dữ liệu, nó bao gồm những ưu điểm như: xây dựng tương đối nhanh; đơn giản, dễ hiểu. Hơn nữa các cây có thể dễ dàng được chuyển đổi sang các câu lệnh SQL để có thể được sử dụng để truy nhập cơ sở dữ liệu một cách hiệu quả. Cuối cùng, việc phân lớp dựa trên cây quyết định đạt được sự tương tự và đôi khi là chính xác hơn so với các phương pháp phân lớp khác.

Biểu đồ phát triển hình cây của cây quyết định gồm:

- Gốc: là node trên cùng của cây;
- Node trong: biểu diễn một kiểm tra trên một thuộc tính đơn;
- Nhánh: biểu diễn các kết quả của kiểm tra trên node trong;
- Node lá: biểu diễn lớp.

Để phân lớp mẫu dữ liệu chưa biết, giá trị các thuộc tính của mẫu được đưa vào kiểm tra trên cây quyết định. Mỗi mẫu tương ứng có một đường đi từ gốc đến lá và lá biểu diễn dự đoán giá trị phân lớp của mẫu đó.

1.1.2.3. Phương pháp học có giám sát sử dụng rừng ngẫu nhiên (Random Forest)

Random Forests – rừng ngẫu nhiên là thuật toán học có giám sát (supervised learning). Nó có thể được sử dụng cho cả phân lớp và hồi quy. Nó cũng là thuật toán linh hoạt và dễ sử dụng nhất. Một khu rừng bao gồm cây cối. Càng có nhiều cây thì rừng càng mạnh. Random forests tạo ra cây quyết định trên các mẫu dữ liệu được chọn ngẫu nhiên, được dự đoán từ mỗi cây và chọn giải pháp tốt nhất bằng cách bỏ phiếu. Nó cũng cung cấp một chỉ báo khá tốt về tầm quan trọng của tính năng.

Random forests có nhiều ứng dụng, chẳng hạn như công cụ đề xuất, phân loại hình ảnh và lựa chọn tính năng. Nó có thể được sử dụng để phân loại các ứng viên cho vay trung thành, xác định hoạt động gian lận và dự đoán các bệnh. Nó nằm ở cơ sở của thuật toán Boruta, chọn các tính năng quan trọng trong tập dữ liệu.

Ưu điểm: Random forests được coi là một phương pháp chính xác và mạnh mẽ vì số cây quyết định tham gia vào quá trình này. Nó không bị vấn đề overfitting. Lý do chính là nó mất trung bình của tất cả các dự đoán, trong đó hủy bỏ những thành kiến. Thuật toán có thể được sử dụng trong cả hai vấn đề phân loại và hồi quy. Random forests cũng có thể xử lý các giá trị còn thiếu. Có hai cách để xử lý các giá trị này: sử dụng các giá trị trung bình để thay thế các biến liên tục và tính toán mức trung bình gần kề của các giá trị bị thiếu. Bạn có thể nhận được tầm quan trọng của tính năng tương đối, giúp chọn các tính năng đóng góp nhiều nhất cho trình phân loại.

Nhược điểm: Random forests chậm tạo dự đoán bởi vì nó có nhiều cây quyết định. Bất cứ khi nào nó đưa ra dự đoán, tất cả các cây trong rừng phải đưa ra dự đoán cho cùng một đầu vào cho trước và sau đó thực hiện bỏ phiếu trên đó. Toàn bộ quá trình này tốn thời gian.

1.1.3. Giới thiệu về các đặc trưng thủ công

Trước năm 2012, hầu hết mọi mô hình AI đều được chia thành 2 bước độc lập tách rời nhau:

- Feature Engineering (Trích chọn đặc trưng): là quá trình dựa trên những hiểu biết của con người về lĩnh vực cần giải quyết (domain knowledge) để từ đó rút ra những đặc trưng (features) của dataset mà có thể giúp ích cho việc giải quyết vấn đề. Do đó các features này được gọi là các đặc trưng thủ công (làm thủ công). Feature extractor là một phần của model dùng để trích xuất ra features nói chung. Các phương pháp thường được sử dụng cho ảnh là SIFT (Scale Invariant Feature Transform), SURF (Speeded-Up Robust Features), HOG (Histogram of Oriented Gradients), LBP (Local Binary Pattern),...
- Classifier/Regressor (Phân loại): dùng các thuật toán Machine Learning để học và dự đoán các kết quả từ những features được tạo ra ở bước trên. Các Classifier thường được sử dụng là multi-class SVM, Softmax Regression, Discriminative Dictionary Learning, Random Forest,...

Các phương pháp này cho kết quả khá ấn tượng trong một số trường hợp. Tuy nhiên, chúng vẫn còn nhiều hạn chế vì quá trình tìm ra các features và các classifier phù hợp vẫn là riêng biệt.

❖ **Một số các đặc trưng thủ công thông dụng**

- SIFT (Scale Invariant Feature Transform)

SIFT là một feature descriptor được sử dụng trong computer vision và xử lý hình ảnh được dùng để nhận dạng đối tượng, khớp hình ảnh, hay áp dụng cho các bài toán phân loại…

Với đầu vào là một hình ảnh, sử dụng SIFT ta thu được các tập các đặc trưng. Mỗi đối tượng trong hình ảnh sẽ cho ra các đặc trưng khác nhau. Ứng với mỗi keypoint ta sẽ thu được: toạ độ keypoint, scale và orientation của keypoint, descriptor. Ta phân biệt các keypoint này với nhau thông qua một vector 128 chiều hay còn gọi là descriptor. Các descriptor này được sử dụng để nhận dạng đối tượng hay sử dụng cho các bài toán phân loại.

- SURF (Speeded-Up Robust Features)

Cũng gồm các bước tương tự như SIFT nhưng ở từng bước, SURF có những cải thiện đáng kể để cải thiện tốc độ tính toán mà vẫn đảm bảo độ chính xác. SURF có tốc độ xử lý nhanh gấp nhiều lần so với SIFT.

- HOG (Histogram of Oriented Gradients)

HOG là một feature descriptor được sử dụng trong computer vision và xử lý hình ảnh, dùng để nhận diện và mô tả hình dạng một đối tượng.

HOG tương tự như các biểu đồ edge orientation, scale-invariant feature transform descriptors (như sift, surf, ...), shape contexts nhưng hog được tính toán trên một lưới dày đặc các cell và chuẩn hóa sự tương phản giữa các block để nâng cao độ chính xác.

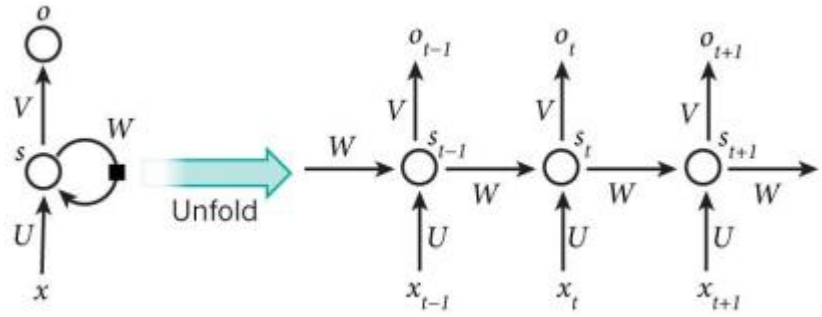
1.2. Các mô hình Học sâu: RNN và CNN

1.2.1. RNN với dữ liệu tín hiệu trên miền thời gian

Mạng nơ-ron hồi quy RNN (Recurrent Neural Network) được giới thiệu bởi John Hopfield năm 1982, là một trong những mô hình học sâu. Recurren có nghĩa là thực hiện lặp lại cùng một tác vụ cho mỗi thành phần trong chuỗi. Trong đó, kết quả đầu ra tại thời điểm hiện tại phụ thuộc vào kết quả tính toán của các thành phần ở những thời điểm trước đó.

RNN là một mô hình có trí nhớ (memory), có khả năng nhớ được thông tin đã tính toán trước đó. Không như các mô hình Neural Network truyền thống trước đó là thông tin đầu vào (input) hoàn toàn độc lập với thông tin đầu ra (output).

Hầu hết RNN được thiết kế như là một chuỗi các module được lặp đi lặp lại, các môdun này thường có cấu trúc đơn giản chỉ có một lớp mạng tanh. Huấn luyện RNN tương tự như huấn luyện ANN truyền thống. Giá trị tại mỗi output không chỉ phụ thuộc vào kết quả tính toán của bước hiện tại mà còn phụ thuộc vào kết quả tính toán của các bước trước đó.



Hình 1.3: Quá trình xử lý thông tin trong mạng RNN[8]

RNN có khả năng biểu diễn mối quan hệ phụ thuộc giữa các thành phần trong chuỗi (nếu chuỗi đầu vào có 6 từ thì RNN sẽ dàn ra thành 6 layer, mỗi layer ứng với mỗi từ, chỉ số mỗi từ được đánh từ 0 đến 5. Trong Hình 1.1 ở trên, x_t là input tại thời điểm thứ t, s_t là hidden state (memory) tại thời điểm thứ t, được tính dựa trên các hidden state trước đó kết hợp với input của thời điểm hiện tại với công thức:

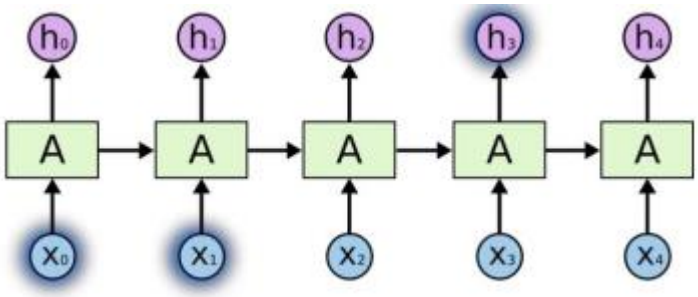
$$S_t = tanh\ (U_{x_t} + W_{s_{t-1}})$$

s_{t-1} là hidden state được khởi tạo là 1 vector 0. O_t là output tại thời điểm thứ t, là một vector chứa xác xuất của toàn bộ các từ trong từ điển.

$$O_t = softmax\ (V_{s_t})$$

Không như ANN truyền thống, tại mỗi layer cần phải sử dụng một tham số khác, RNNs chỉ sử dụng một bộ parameters (U, V, W) cho toàn bộ các bước.

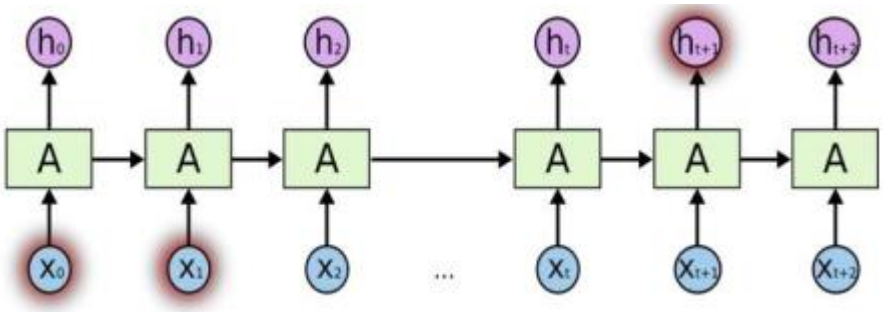
Ý tưởng ban đầu của RNN là kết nối những thông tin trước đó nhằm hỗ trợ cho các xử lý hiện tại. Nhưng đôi khi, chỉ cần dựa vào một số thông tin gần nhất để thực hiện tác vụ hiện tại. Ví dụ, chúng ta dự đoán từ cuối cùng trong câu "chuồn_chuồn bay thấp thì mưa", thì chúng ta không cần truy tìm quá nhiều từ trước đó, ta có thể đoán ngay từ tiếp theo sẽ là "mưa". Trong trường hợp này, khoảng cách tới thông tin liên quan được rút ngắn lại, mạng RNN có thể học và sử dụng các thông tin quá khứ.



Hình 1.4: RNN phụ thuộc short-term
(Nguồn: Tìm hiểu về Recurrent Neural Network – Internet)

Trường hợp có nhiều thông tin hơn trong một câu, nghĩa là phụ thuộc vào ngữ cảnh. Ví dụ nhưng khi dự đoán từ cuối cùng trong đoạn văn bản **"Tôi sinh ra và lớn lên ở Việt_Nam… Tôi có_thể nói thuần_thục Tiếng_Việt."** Từ thông tin gần nhất cho thấy rằng từ tiếp theo là tên một ngôn ngữ, nhưng khi chúng ta muốn biết cụ thể ngôn ngữ nào, thì cần quay về quá khứ xa hơn, để tìm được ngữ cảnh **Việt_Nam**. Và như vậy, RNN có thể phải tìm những thông tin có liên quan và số lượng các điểm đó trở nên rất lớn.

Không được như mong đợi, RNN không thể học để kết nối các thông tin lại với nhau.



Hình 1.5: RNN phụ thuộc long-term
(Nguồn: Tìm hiểu về Recurrent Neural Network – Internet)

Về lý thuyết, RNN có thể nhớ được thông tin của chuỗi có chiều dài bất kì, nhưng trong thực tế mô hình này chỉ nhớ được thông tin ở vài bước trước đó.

RNN có các phiên bản mở rộng như: Bidirectional RNN **(**RNN hai chiều**),** Deep (Bidirectional) RNN**,** Long short-term memory networks (LSTM).

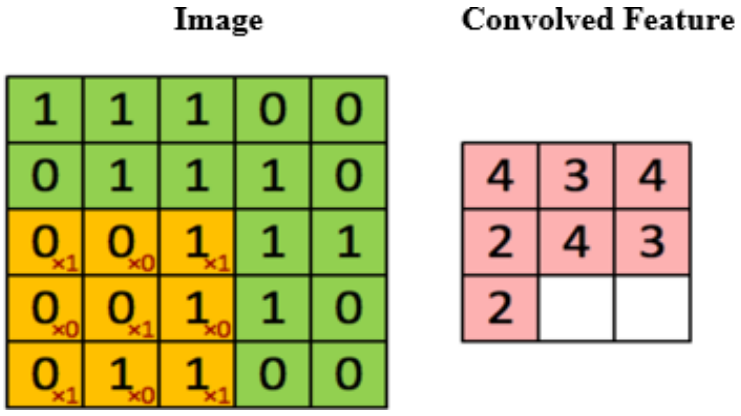
1.2.2. CNN với dữ liệu "ảnh của âm thanh" (dạng biểu diễn tần số của âm thanh)

Những năm gần đây, chúng ta đã chứng kiến được nhiều thành tựu vượt bậc trong ngành thị giác máy tính (Computer Vision). Các hệ thống xử lý ảnh lớn như Facebook, Google hay Amazon đã đưa vào sản phẩm của mình những chức năng thông minh như nhận diện khuôn mặt người dùng, phát triển xe hơi tự lái hay drone giao hàng tự động, Convolutional Neural Network (CNN – Mạng nơ-ron tích chập) là một trong những mô hình Học sâu tiên tiến giúp cho chúng ta xây dựng được những hệ thống thông minh với độ chính xác cao như hiện nay. Tên gọi của mô hình được dựa trên phép tính quan trọng được sử dụng trong mạng đó là Tích chập. Đây là mô hình Học sâu tiên tiếp giúp ta xây dựng được hệ thống thông minh tiên tiến với độ chính xác cao.

❖ **Convolution (Tích chập)**

Phép tích chập được sử dụng đầu tiên trong xử lý tín hiệu số (Signal processing). Nhờ vào nguyên lý biến đổi thông tin có thể áp dụng kỹ thuật này vào xử lý ảnh và video số.

Để dễ hình dung, ta có thể xem tích chập như một cửa số trượt (sliding window) áp đặt lên một ma trận.



Hình 1.6: Minh họa phép tích chập
(Nguồn: Mạng nơ-ron tích chập - Internet)

Ma trận bên trái là một bức ảnh đen trắng. Mỗi giá trị của ma trận tương đương với một điểm ảnh (pixel), 0 là màu đen, 1 là màu trắng (nếu là ảnh grayscale thì giá trị biến thiên từ 0 đến 255).

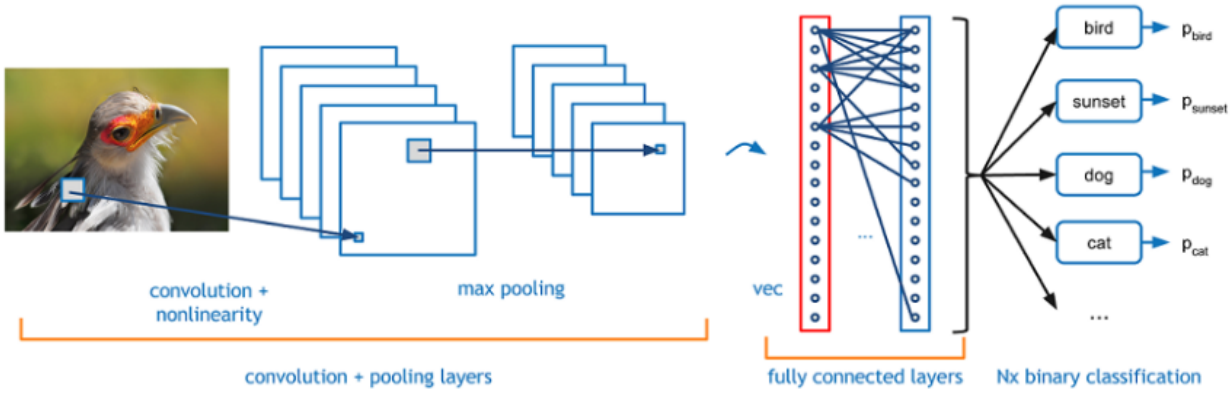
Sliding window còn có tên gọi là kernel, filter hay feature detector. Ở đây, ta dùng một ma trận filter 3×3 nhân từng thành phần tương ứng (element-wise) với ma trận ảnh bên trái. Giá trị đầu ra do tích của các thành phần này cộng lại. Kết quả của tích chập là một ma trận (convoled feature) sinh ra từ việc trượt ma trận filter và thực hiện tích chập cùng lúc lên toàn bộ ma trận ảnh bên trái. Ví dụ của phép toán tích chập như:

- Ta có thể làm mờ bức ảnh ban đầu bằng cách lấy giá trị trung bình của các điểm ảnh xung quanh cho vị trí điểm ảnh trung tâm.
- Ngoài ra, ta có thể phát hiện biên cạnh bằng cách tính vi phân (độ dị biệt) giữa các điểm ảnh lân cận.

Bây giờ, chúng ta đã biết thế nào là tích chập. Tiếp đến chúng ta sẽ tìm hiểu về CNNs, CNNs chỉ đơn giản gồm những layer của convolution kết hợp với các hàm kích hoạt phi tuyến (nonlinear activation function) như ReLU, SoftMax hay Tanh để tạo ra thông tin trừu tượng hơn cho các layer tiếp theo.

Trong mô hình này, các layer được liên kết với nhau thông qua cơ chế convolution. Layer tiếp theo là kết quả convolution từ layer trước đó, nhờ vậy mà ta có được các kết nối cục bộ.

Mỗi layer như vậy được áp đặt các filter khác nhau, thông thường có vài trăm đến vài nghìn filter như vậy. Một số layer khác như pooling/subsampling layer dùng để chắt lọc lại các thông tin hữu ích hơn (loại bỏ các thông tin nhiễu). Trong suốt quá trình huấn luyện, CNNs sẽ tự động học được các thông số cho các filter. Ví dụ trong tác vụ phân lớp ảnh, CNNs sẽ cố gắng tìm ra thông số tối ưu cho các filter tương ứng theo thứ tự raw pixel > edges > shapes > facial > high-level features. Layer cuối cùng được dùng để phân lớp ảnh.



Hình 1.7: Mô tả quá trình phân loại ảnh
(Nguồn: Mạng nơ-ron tích chập - Internet)

Việc xây dựng lớp tích chập dựa vào các ý tưởng như sau:

- *Kết nối cục bộ*

Trong quá trình xử lý dữ liệu đầu vào có chiều rộng, chiều cao như hình ảnh, việc kết nối các nơ-ron với tất cả các nơ-ron trong lớp trước đó là không thực tế vì kiến trúc mạng không tính đến cấu trúc không gian của dữ liệu. thay vào đó, chúng ta sẽ thực thi một kiểu kết nối cục bộ thưa thớt giữa các nơ-ron của các lớp liền kề: mỗi nơ-ron chỉ được kết nối với một vùng nhỏ của tín hiệu đầu vào.

- *Sắp xếp không gian*

Trong phần này đồ án sẽ giải thích rõ hơn về số lương nơ ron tín hiệu đầu ra và cách chúng được sắp xếp. ba siêu đường kính (heperparameters) kiểm soát kích thước tín hiệu đầu ra là: depth , stride và zero-padding.

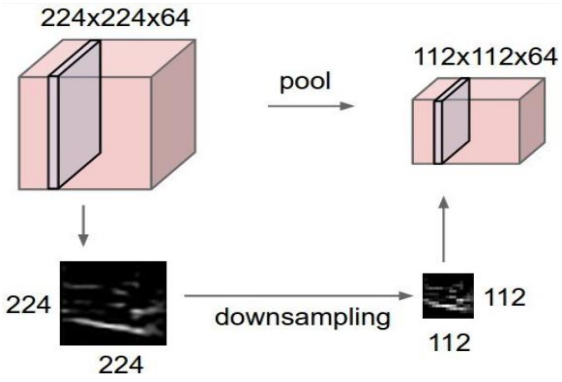
Đầu tiên, giá trị của depth là độ sâu của tín hiệu đầu ra. Đây là một siêu tham số, nó tương ứng với số lượng bộ lọc muốn sử dụng. Depth (độ sâu) kiểm soát số lượng nơ ron trong một lớp kết nối với một vùng của tín hiệu đầu vào. Nhưng nơ ron này học các kích hoạt cho các đặc trưng khác nhau trong tín hiệu đầu vào.

Tiếp theo, chúng ta phải xác định stride – tức là bước tiến chúng ta trượt filter (bộ lọc). Chẳng hạn với stride bằng 1, chúng ta di chuyển các bộ lọc một pixel mỗi lần.

Để kiểm soát kích thước không gian đầu ra, chúng ta sử dụng Zero-padding tức là sẽ thêm các số 0 vào các đường biên của ma trận hình ảnh.

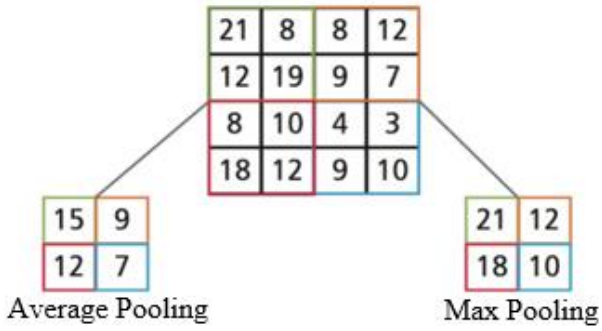
- *Lớp tổng hợp (Pooling Layer)*

Bài báo [7] định nghĩa lớp Pooling sử dụng một cửa sổ trượt quét qua toàn bộ ma trận dữ liệu theo một bước trượt cho trước để tiến hành lấy mẫu. Cụ thể là, Pooling layer thực hiện chức năng làm giảm chiều không gian của đầu ra một lớp tích chập trước khi cho nó đi vào một lớp tích chập khác. Gọi pooling size kích thước K*K. Input của pooling layer có kích thước H*W*D, ta tách ra làm D ma trận kích thước H*W. Với mỗi ma trận, trên vùng kích thước K*K trên ma trận ta tìm maximum hoặc average của dữ liệu rồi viết vào ma trận kết quả. Quy tắc về stride và padding áp dụng như phép tính convolution trên ảnh đã được trình bày phía trên.



Hình 1.8: Mô tả lớp Pooling Layer
(Nguồn: Tìm hiểu về Convolutional Neural Network - Internet)

Có hai loại Pooling layer phổ biến đó là max pooling và average pooling.



Hình 1.9: Hình ảnh mô tả hai loại Pooling
(Nguồn: Tìm hiểu về Convolutional Neural Network - Internet)

- *Lớp kết nối đầy đủ (Fully Connected Layer – FC)*

Bài báo [7] định nghĩa, lớp kết nối đầy đủ là một lớp giống như mạng nơ ron truyền thẳng các giá trị được tính toán được ở các lớp trước sẽ được liên kết đầy đủ với các nơ ron ở lớp tiếp theo.

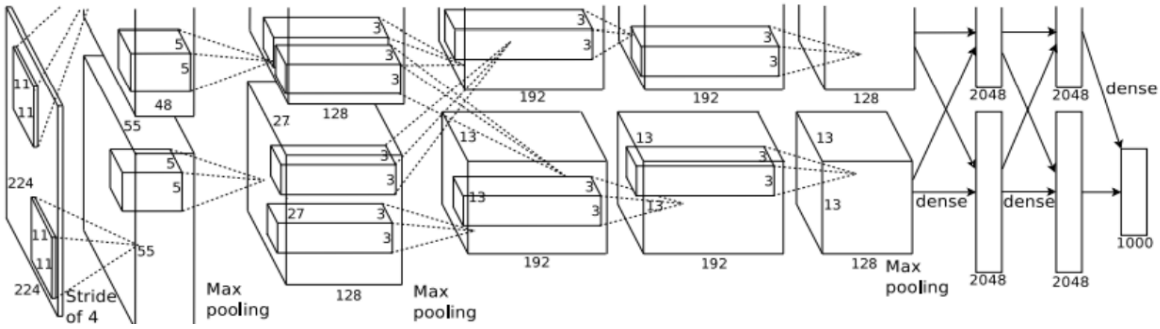
Tại lớp mạng này, mỗi nơ ron của layer này sẽ liên kết với mọi nơ ron của lớp khác. Để đưa ảnh từ các layer trước vào mạng này, buộc phải flatten (làm phẳng) bức ảnh thành một vector thay vì mảng nhiều chiều. Tại layer cuối cùng sẽ sử dụng hàm kích hoạt phù hợp với yêu cầu của bài toán.

1.3. Các mô hình mô hình học sâu cho phân loại hình ảnh

1.3.1. Các mô hình học sâu tiên tiến

Có một số kiến trúc mạng nơ ron tích chập nổi tiếng. Một số thử nghiệm cho thấy chúng có hiệu suất tốt hơn. Vì vậy, đôi khi nhiều người sử dụng mạng được thiết kế sẵn thay vì tự thiết kế mạng. Ở các phần sau luận văn sẽ giới thiệu một vài mạng tích chập nổi tiếng và thông dụng hiện nay.

❖ ***AlexNet***



Hình 2.8: Kiến trúc mạng AlexNet [13]

Alex đã phát triển mạng này vào năm 2012. Cho tới thời điểm hiện tại, AlexNet vẫn còn đang được sử dụng phổ biến và rộng rãi. Mạng AlexNet có năm lớp chập và ba lớp kết nối đầy đủ. Cấu trúc trong AlexNet được chia thành hai khối. Nguyên nhân vì tác giả đã sử dụng hai GPU để huấn luyện dữ liệu song song. Mạng này được sử dụng trong phân loại đối tượng quy mô lớn. Lớp đầu ra có một nghìn nơ

ron. Đó là bởi vì kiến trúc ban đầu được thiết kế để phân loại một nghìn nhãn. Thông thường, những người áp dụng kiến trúc mạng nơ ron AlexNet sẽ thay thế lớp cuối cùng, phụ thuộc vào mục đích của họ. Tác giả của mạng này đã làm nhiều thử nghiệm để có thể ra được mô hình này có kết quả tốt nhất. Vì vậy, hiệu suất của cấu trúc này rất ổn định và mạng này được sử dụng rộng rãi trong nhiều ứng dụng

Dưới đây là các điểm quan trọng về AlexNet:

- Huấn luyện mạn với bộ dữ liệu ImageNet. Bộ dữ liệu bao gồm 15 triệu ảnh được chú giải từ 22000 lớp.
- Sử dụng ReLU cho hàm nonlinearity (để làm giảm thời gian huấn luyện, vì ReLU nhanh hơn một vài lần so với hàm tanh).
- Sử dụng kỹ thuật tăng dữ liệu bao gồm dịch ảnh, phản xạ ngang và chiết xuất bản vá.
- Sử dụng lớp dropout để xử lý vấn đề overfitting cho dữ liệu huấn luyện.
- Huấn luyện mô hunhf sử dụng batch stochastic gradient descent.
- Huấn luyện trên 2 GPU GTX 580 trong 5-6 ngày.

❖ ***ZF Net***

Với sự nổi bật của AlexNet năm 2012, đã có rất nhiều mô hình CNN được gửi đến cuộc thi ILSVRC 2013. Mô hình chiến thắng cuộc thi năm đó là mạng được xây dựng bởi Matthew Zeiler và Rob Fergus có tên là ZF Net với 11.2% tỉ lệ lỗi. Kiến trúc này được tối ưu tốt hơn so với kiến trúc của AlexNet như tăng độ chính xác và cải thiện hiệu suất.

Dưới đây là một số đặc điểm về mô hình ZF Net:

- Kiến trúc giống với AlexNet, ngoại trừ một vài sửa đổi nhỏ
- AlexNet huấn luyện trên 15 triệu hình ảnh, trong khi ZF Net chỉ huấn luyện trên 1,3 triệu hình ảnh.
- Thay vì sử dụng các bộ lọc có kích thước 11x11 trong lớp đầu tiên (theo AlexNet), ZF Net đã sử dụng các bộ lọc có kích thước 7x7. Lý do đằng sau sự thay đổi này là bộ lọc có độ lớn nhỏ hơn ở lớp conv đầu tiên sẽ giúp giữ

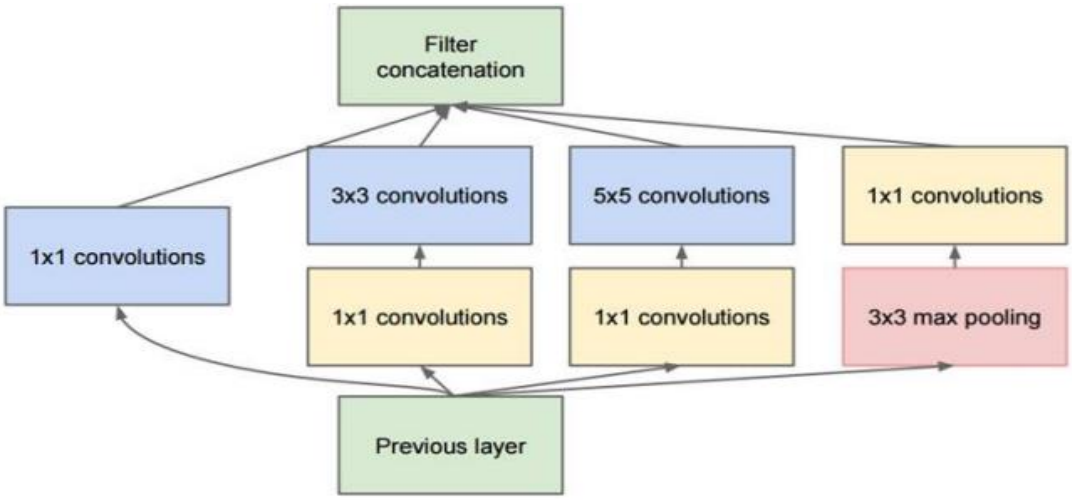
lại nhiều điểm ảnh của đầu vào hơn. Bộ lọc với độ lớn 11*11 đã được chứng minh là bỏ qua rất nhiều những thông tin, đặc biệt là trong lớp conv đầu tiên

- Khi mạng phát triển, số lượng bộ lọc được sử dụng tăng lên.
- Được huấn luyện trên GPU GTX 580 trong mười hai ngày.

ZF Net không chỉ giành chiến thắng trong cuộc thi năm 2013, phương pháp này còn cung cấp kiến thức để làm việc với CNNs và mô tả một vài cách để tăng cường hiệu suất. Cách mô tả mới không chỉ giúp ta giải thích rõ hơn bên trong của mạng CNN, mà nó còn cung cấp cái nhìn sâu sắc để cải tiến kiến trúc mạng.

❖ ***GoogleNet (2015)***

GoogleNet là một mạng nơ-ron tích chập 22 lớp, chiến thắng cuộc thi ILSVRC 2014 với top 5 tỉ lệ lỗi là 6.7%. Đây là một trong những kiến trúc mạng CNN đi ra khỏi mục đích thông thường của việc chồng các lớp conv và pooling.



Hình 2.9: Kiến trúc mạng GoogleNet [13]

Thông thường, tại mỗi lớp ConvNet truyền thống, ta phải chọn giữa lớp pooling hoặc lớp conv. Mô hình Inception cho phép ta thực hiện tất cả các bước trên song song nhau. Kiến trúc mô hình chứa một mạng conv, một lớp conv độ lớn trung bình, một lớp conv lớn, và một lớp pooling. Mạng conv có khả năng tách ra các chi tiết nhỏ, mạng 5*5 có khả năng phủ các vùng rộng hơn. Ta còn sử dụng lớp pooling

giúp giảm độ lớn dữ liệu và xử lý overfitting. Thêm vào đó, ta thêm vào sau mỗi lớn conv một lớp ReLUs, giúp cải tiến tính chất phi tuyến của mạng. Về cơ bản, mạng có khả năng thực hiện các chức năng với các bước tính toán trên mà vẫn giữ được khả năng tính toán tốt.

Dưới đây là một số đặc điểm về mô hình GoogleNet:

- Sử dụng 9 mô hình Inception trong kiến trúc, với tổng hơn 100 lớp mạng.
- Kiến trúc này sử dụng một kỹ thuật tên Average Pooling để lấy trung bình các lớp thuộc tính
- Sử dụng ít tham số hơn mạng AlexNet đến 12 lần.
- GoogleNet là một trong những mô hình đầu tiên giới thiệu ý tưởng các lớp CNN không cần phải chồng lên nhau một cách tuần tự.

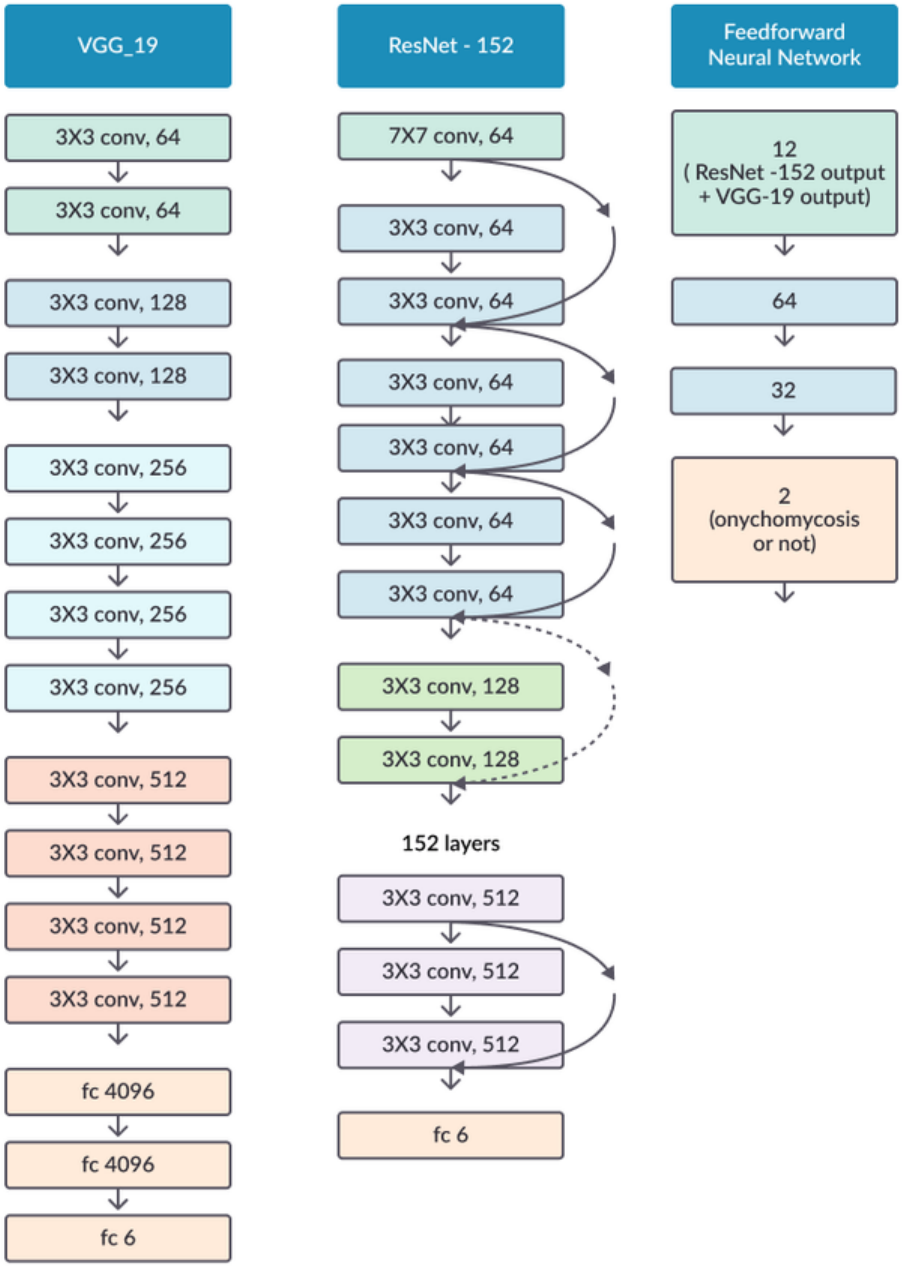
❖ ***ResNet (2015)***

ResNet (Residual Network) được giới thiệu đến công chúng vào năm 2015 và đã giành được vị trí thứ 1 trong cuộc thi ILSVRC 2015 với tỉ lệ lỗi chỉ 3.57%. Mạng ResNet (R) là một mạng CNN được thiết kế để làm việc với hàng trăm hoặc hàng nghìn lớp chập.

ResNet đưa ra là sử dụng kết nối "tắt" đồng nhất để xuyên qua một hay nhiều lớp.

- ResNet gần như tương tự với các mạng gồm có convolution, pooling, activation và fully connected. Ảnh bên trên hiển thị khối dư được sử dụng trong mạng. Xuất hiện một mũi tên cong xuất phát từ đầu và kết thúc tại cuối khối dư. Với H(x) là giá trị dự đoán, F(x) là giá trị thật (nhãn), chúng

ta muốn H(x) bằng hoặc xấp xỉ F(x). Việc F(x) có được từ x như sau: **X->weight1-> ReLU -> weight2**.



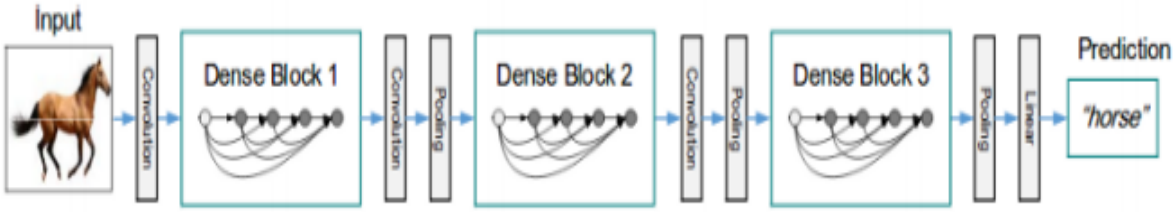
Hình 2.10: Kiến trúc các lớp của mạng ResNet [13]

- ResNet sử dụng các kết nối tắt (kết nối trực tiếp đầu vào của lớp (n) với (n+x) được hiển thị dạng mũi tên cong. Qua mô hình nó chứng minh được có thể cải thiện hiệu suất trong quá trình training model khi mô hình có hơn 20 lớp.

❖ ***DenseNet (2016)***

DenseNet được phát minh bởi Gao Huang, Zhuang Liu, Laurens van der Maaten và Kilian Q. Weinberger. DenseNet rất giống với ResNet nhưng có hai sự thay đổi quan trọng:

- Thay vì cộng vào các đặc trưng như trong ResNet, họ nối chúng với nhau.
- Thay vì chỉ cộng với đầu vào của một lớp trước, DenseNet cộng với đầu vào của nhiều lớp trước nó. Nói rõ hơn, trong cấu trúc mạng ResNet, tại lớp thứ l, ta cộng lớp thứ (l-1) và lớp thứ nhất. Trong mạng DenseNet, tại lớp thứ l, ta nối tất cả các đặc trưng từ lớp thứ nhất tới lớp thứ (l-1).
- DenseNet121 chỉ với 8 triệu tham số nhưng có độ chính xác cao hơn so với ResNet50 với gần 26 triệu tham số trên bộ dữ liệu ImageNet
- Áp dụng BatchNormalization trước khi thực hiện tích chập ở các tầng chuyển tiếp nên giảm được triệt tiêu đạo hàm



Hình 2.11: Quá trình nhận diện hình ảnh sử dụng kiến trúc mạng DenseNet [13]

❖ ***Phương pháp học chuyển giao mạng nơ-ron***

Học chuyển tiếp là một phương pháp rất nổi tiếng trong lĩnh vực xử lý thị giác máy tính vì với học chuyển tiếp, thay vì phải bắt đầu quá trình học từ đầu, ta có thể bắt đầu từ các mẫu đã được học khi giải quyết các vấn đề khác.

Trong lĩnh vực thị giác máy tính, học chuyển tiếp thường sử dụng các mô hình được huấn luyện sẵn. Mô hình huấn luyện sẵn là mô hình được huấn luyện trên một tập dữ liệu lớn để giải quyết một vấn đề tương tự như cái mà ta muốn giải quyết. Thông thường, ta sẽ sử dụng các mô hình từ các bài nghiên cứu được công khai (ví dụ VGG, Inception, …).

Một vài mô hình huấn luyện sẵn được sử dụng trong học chuyển tiếp được dựa trên các mạng nơ-ron tích chập lớn, như đã bàn luận ở phần trước. Kiến trúc CNN thông thường có hai phần:

- Phần trích xuất đặc trưng, bao gồm các lớp conv và pooling chồng lên nhau. Mục đích chính là để xuất ra các thuộc tính từ hình ảnh.
- Phần phân loại, được tạo thành từ các lớp liên kết đầy đủ. Mục đích chính của phần phân loại là để phân loại hình ảnh theo các đặc trưng đã được trích xuất. Lớp liên kết đầy đủ là lớp các nơ-ron liên kết hoàn toàn với các lớp đằng trước.

Một mặt quan trọng của mô hình học sâu là nó có thể tự động học thứ bậc cấu trúc của các đặc trưng. Điều này có nghĩa là, đặc trưng được tính toán bởi các lớp đầu tiên là rất tổng quát, có thể sử dụng lại ở các mô hình khác, trong khi đặc trưng được tạo ra bởi lớp cuối cùng là rất riêng và phụ thuộc vào bộ dữ liệu và nhiệm vụ.

Khi sử dụng mô hình huấn luyện sẵn cho nhiệm vụ mới, ta có thể loại bỏ lớp phân loại cũ và thay vào lớp mới, phù hợp với mục đích của ta hơn, và cuối cùng cần căn chỉnh lại mô hình theo một tổng hai phương pháp sau:

- Huấn luyện lại toàn bộ mô hình. Trong trường hợp này, ta sử dụng kiến trúc của mô hình có sẵn và huấn luyện nó với bộ dữ liệu mới. Cách này cần bộ dữ liệu lớn và khả năng tính toán cao.
- Chỉ huấn luyện một vài lớp. Nhắc lại rằng, các lớp cấp thấp thường là các đặc trưng chung chung (không phụ thuộc vào vấn đề), còn các lớp cấp cao hơn cho ra các đặc trưng rất riêng (phụ thuộc vào vấn đề cần giải quyết). Thông thường, nếu chúng ta có một tập dữ liệu nhỏ và một số lượng tham số lớn, nhiều lớp nên được đóng băng để tránh tình trạng bị overfit. Ngược lại, nếu tập dữ liệu lớn và số lượng tham số nhỏ, mô hình có thể được cải thiện bằng cách huấn luyện nhiều lớp hơn cho nhiệm vụ mới vì overfit không phải là vấn đề.

- Trong học máy và học sâu, thì transfer learning là kỹ thuật cho phép tận dụng những gì mình học được từ tập dữ liệu/ ứng dụng/ kiến trúc này sang tập dữ

liệu/ứng dụng/kiến trúc khác. Khác với học máy thông thường, từng nhiệm vụ sẽ có một hệ thống học riêng như ở hình ví dụ trên.

Tại sao nên dùng Transfer Learning trong Học sâu?

- **Không đủ dữ liệu**: DL cần rất nhiều dữ liệu, và ngốn rất nhiều tài nguyên để học trên tập dữ liệu và ứng dụng đó và việc huấn luyện DL trên tập dữ liệu ít phần nhiều là không hiệu quả. Vậy ngoài kỹ thuật data augmentation (kĩ thuật sử dụng để làm tăng số lượng dữ liệu trong tập dữ liệu huấn luyện).
- **Không đủ tài nguyên**: Việc học trên tập dữ liệu lớn rất ngốn nhiều tài nguyên. Transfer learning sẽ góp phần giảm phần nào thời lượng training.
- **Cải thiện chất lượng**. Rất nhiều trường hợp transfer learning cải thiện chất lượng dự đoán của Target Task so với việc train lại từ đầu. Lý do có thể do Source Network được train với dữ liệu lớn và học được tính khái quát hóa tốt hơn.

Transfer learning giúp giải quyết các vấn đề mà không có dữ liệu, các vấn đề mới chưa được học trước đó.

1.3. Kết luận chương 1

Trong chương 1, luận văn đã tìm hiểu được tổng quan về học máy, giới thiệu một số thuật toán học máy truyền thống cũng như tìm hiểu về các mô hình, ứng dụng của các mạng Học sâu như RNN&CNN. Trong chương tiếp theo, luận văn sẽ nghiên cứu cơ sở lý thuyết về âm thanh, phương pháp chuyển đổi tín hiệu âm thanh từ miền thời gian sang miền tần số từ đó đưa ra được "ảnh phổ của âm thanh" và một số phương pháp phân loại âm thanh thường được sử dụng.

CHƯƠNG 2: MỘT SỐ PHƯƠNG PHÁP PHÂN LOẠI ÂM THANH

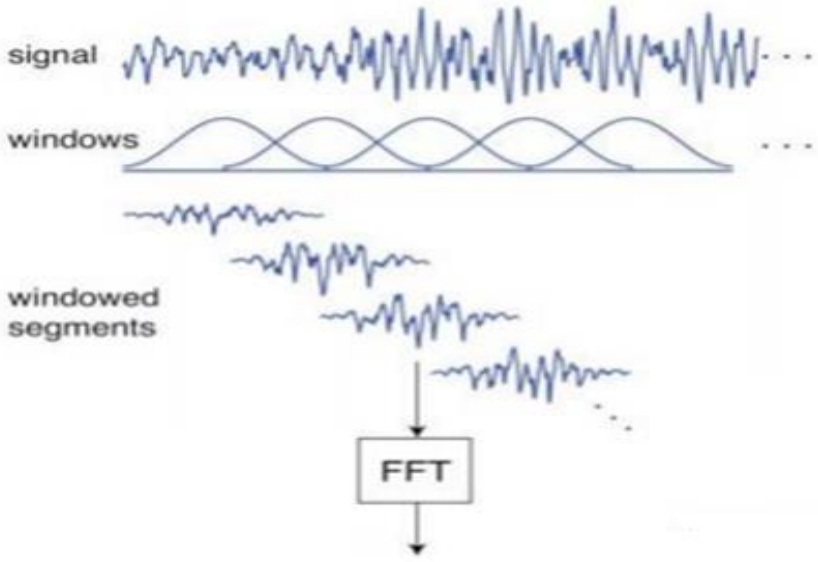
Tóm tắt chương: Trong chương 2, luận văn trình bày các cơ sở lý thuyết liên quan đến phương pháp sử dụng mô hình học sâu CNN trên dữ liệu dạng biểu diễn tần số của âm thanh. Cụ thể, phần đầu tiên của chương này sẽ tóm lược các một vài phương pháp chuyển dữ liệu âm thanh từ miền thời gian sang miền tần số để thu được "ảnh của âm thanh". Tiếp đến, một số mô hình học sâu quan trọng, thường dùng trong các bài toán phân loại hình ảnh sẽ được trình bày.

2.1. Phương pháp tiền xử lý dữ liệu âm thanh

2.1.1. Short-time Fourier Transform

Short-time Fourier Transform là phép biến đổi Fourier thời gian ngắn (STFT), nói một các đơn giản, STFT thực hiện các nội dung sau:

- Định nghĩa một cửa sổ để phân tích (ví dụ: hẹp 30ms, rộng 5ms);
- Định nghĩa một lượng trùng lên nhau giữa các cửa sổ (ví dụ: 30%);
- Định nghĩa hàm cửa sổ (ví dụ: Hann, Gaussian);
- Tạo ra phân đoạn cửa sổ (nhân tín hiệu với hàm cửa sổ);
- Áp dụng biến đổi Fourier thời gian ngắn với mỗi phân đoạn cửa sổ.



Hình 2.1 Phép biến đổi Fourier từ miền thời gian sang miền tần số [9]

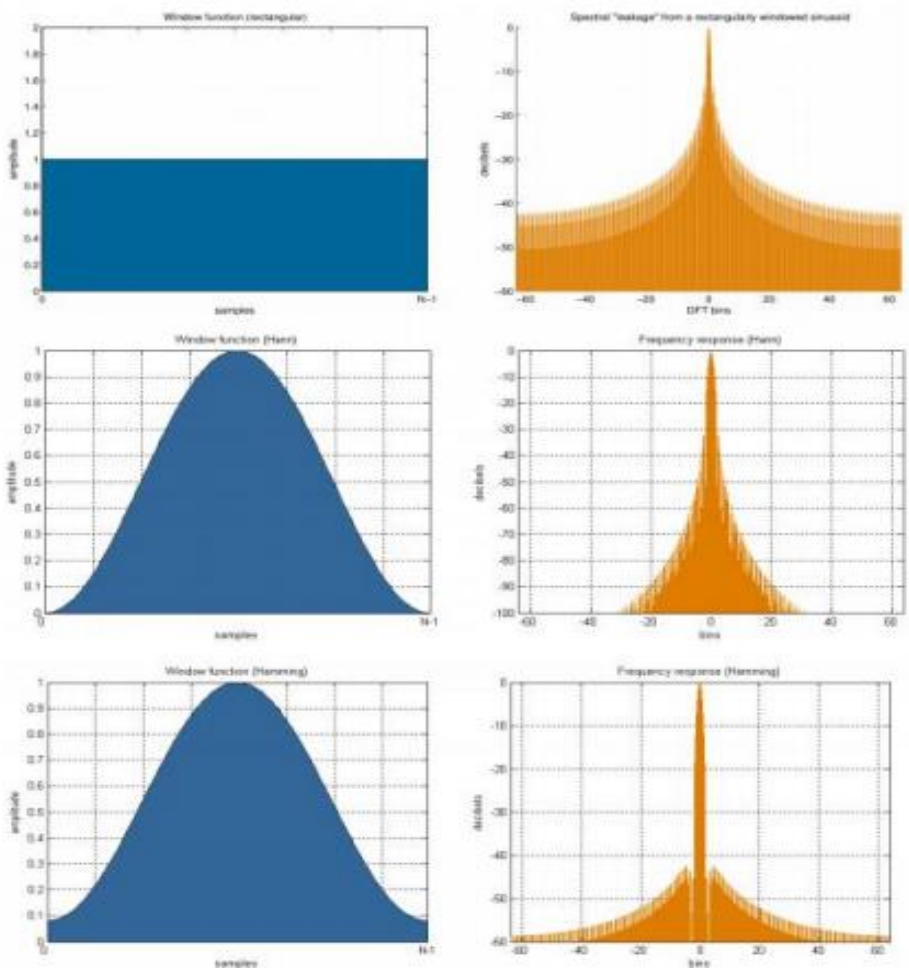
Trong xử lý tín hiệu và thống kê, hàm cửa sổ là một hàm toán học có giá trị bằng không, ngoài một số khoảng đã chọn, đối xứng xung quanh tâm khoảng thời gian và gần mức tối đa ở giữa và giảm dần từ giữa.

- Hàm cửa sổ: trong các ứng dụng điển hình, hàm cửa sổ được sử dụng là các đường cong "hình chuông" không âm, mượt mà. Hình chữ nhật, hình tam giác và các chức năng khác cũng có thể được sử dụng. Cửa sổ được chọn để đánh đổi chiều rộng thùy chính của nó so với sự suy giảm của các thùy bên của nó.

Rectangular

Hann

Hamming



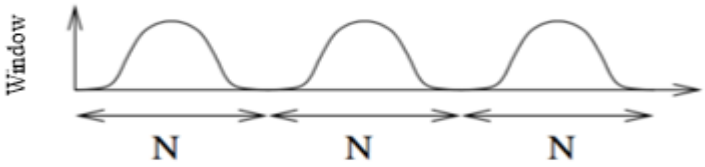
Hình 2.2. Hàm cửa sổ [10]

Hàm cửa sổ Hann và Hamming:

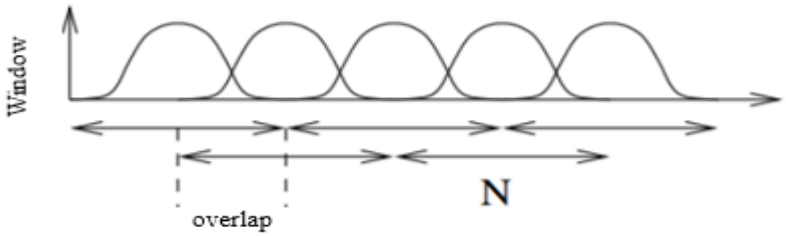
$$w[n,\tau] = 0.54 - 0.4cos\frac{2\pi(n-\tau)}{N_w - 1}$$

$$w[n,\tau] = 0.5(1 - cos\frac{2\pi(n-\tau)}{N-1}))$$

Lý do cho việc sử dụng hàm cửa sổ là để làm giảm ảnh hưởng của rò rỉ phổ và ta dùng bất kỳ hàm làm thon nào để làm giảm các biên tới không.



Vì vậy, chúng ta thường mất một số dữ liệu, để khắc phục, ta sẽ chồng 50% cửa sổ khi xử lý.



Phép biến đổi Fourier thời gian ngắn:

- Phép biến đổi Fourier thời gian (DTFT): biểu thức của biến đổi Fourier

$$X(j\omega) = \int_{-\infty}^{\infty} x(t)e^{-j\omega}dt$$

DTFT có thể được suy ra bởi lấy tích phân:

$$X(e^{j\hat{\omega}}) = \sum_{-\infty}^{\infty} x[n]e^{-j\,\hat{\omega}n} \quad với\ x[n] = x(nT_S)\ và\ \hat{\omega} = 2\pi F/F_S$$

- Biến đổi Fourier phân tán (DFT): DFT thu được khi lấy mẫu DTFT với N tần số rời rạc $\omega_k = 2\pi F/N$ cái mà cho ra biến đổi:

$$X[k] = \sum_{n=0}^{N-1} x[n]e^{-j\frac{2\pi}{N}kn}$$

- Biến đổi Fourier phân tán thời gian ngắn: Biến đổi Fourier dạng sóng giọng nói cửa sổ được định nghĩa là:

$$X(n,\omega) = \sum_{m=-\infty}^{\infty} x[m]w[n-m]e^{-j\omega n}$$

Với $f_n[m] = x[m]w[n-w]$ là phần thời gian ngắn của tín hiệu giọng nói x[m] tại thời điểm n.

- STFT rời rạc: tương đồng với DTFT, STFT được định nghĩa bởi

$$X(n,k) = X(n,\omega)|_{\omega=\frac{2\pi}{N}k}$$

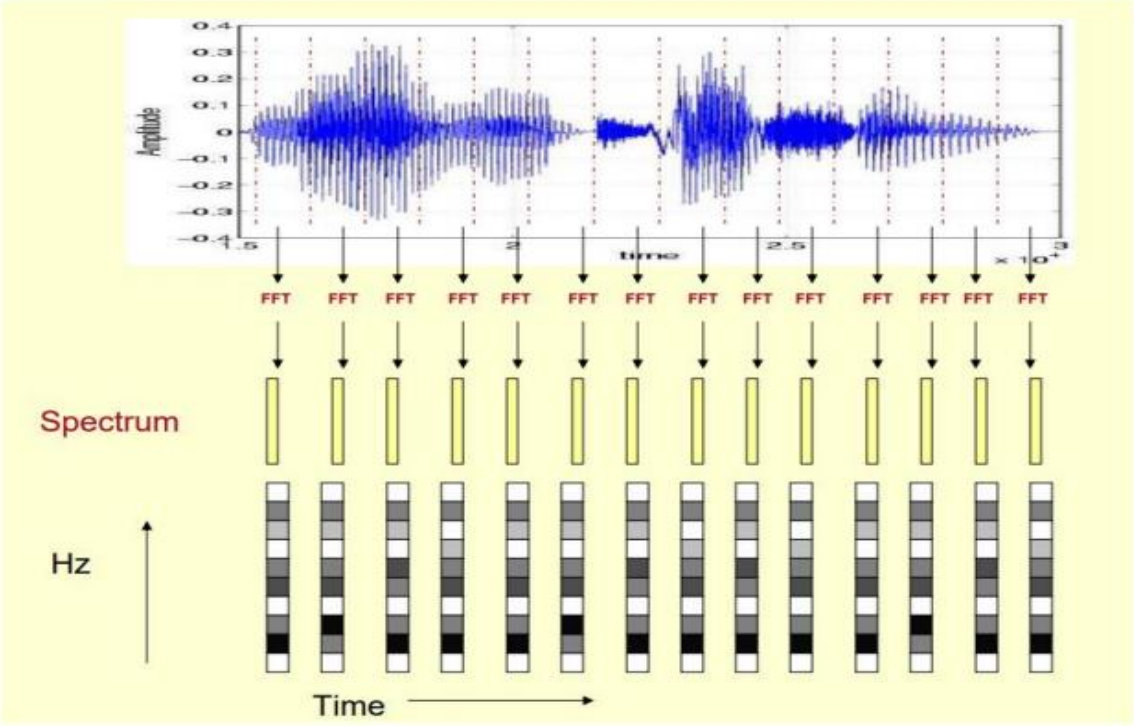
2.1.2. Spectrogram

Có thể liên tưởng spectrogram là "bức tranh của âm thanh", với một cột biểu diễn những tần số tạo nên âm thanh, từ thấp đến cao trong khi cột còn lại biểu diễn sự thay đổi theo thời gian. Nói đơn giản, spectrogram là những tín hiệu âm thanh được biểu diễn bởi một chuỗi các vector quang phổ.

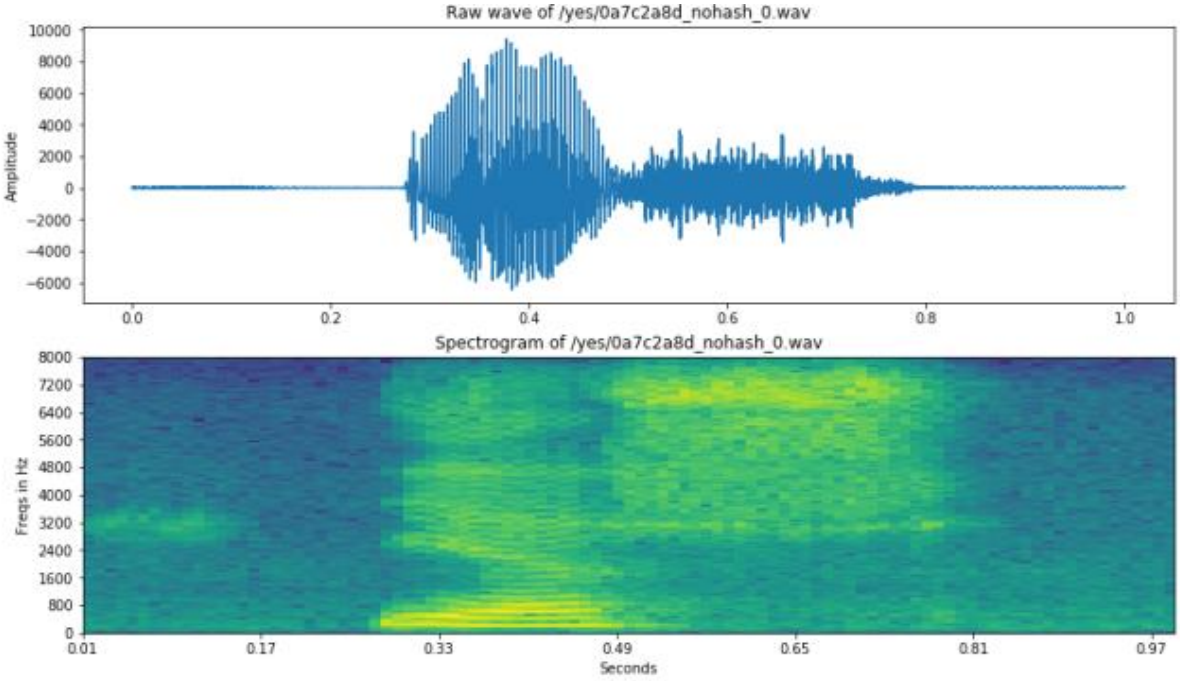
Spectrogram được tạo nên bởi các bước sau:

- Chia tín hiệu thành các đoạn có độ dài bằng nhau. Các phân đoạn phải đủ ngắn để đảm bảo tần số của tín hiệu không thay đổi đáng kể trong đoạn. Đoạn có thể trùng lặp hoặc không.
- Nhân chập mỗi đoạn với hàm cửa sổ để tính STFT.

Trục x là trục thời gian (tương ứng với thứ tự các frame), trục y thể hiện dải tần số từ 0 đến 10000 Hz, giá trị cường độ tại từng tần số được thể hiện bằng màu sắc.



Hình 2.3: Spectrogram của âm thanh
(Nguồn: Tìm hiểu về xử lý ngôn ngữ tư nhiên Speeech To Text - Internet)



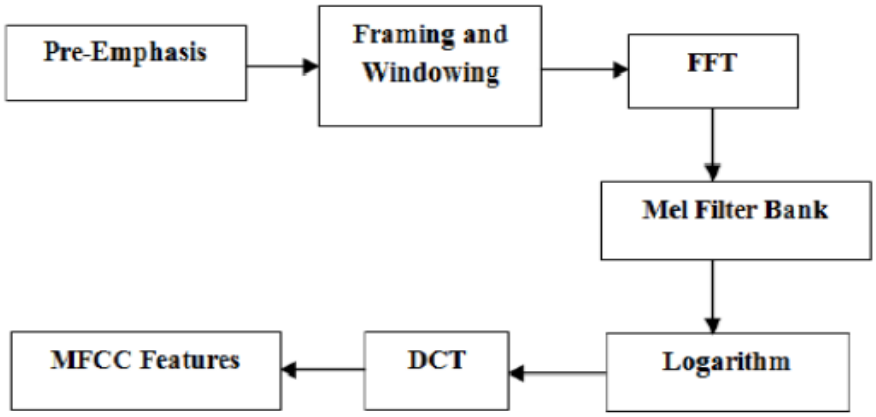
Hình 2.4: Spectrogram của âm thanh Yes/No

(Nguồn: Tìm hiểu về xử lý ngôn ngữ tư nhiên Speeech To Text - Internet)

2.1.3. Ngân hàng bộ lọc và Mel-Frequency Cepstral Coefficients (MFCC)

MFCCs là một thuộc tính được sử dụng rộng rãi trong nhận dạng giọng nói và nói tự động. Thuộc tính này được giới thiệu bởi Davis và Mermelstein trong những năm 1980 và nó vẫn rất tốt từ đó đến nay.Trước khi MFCCs được giới thiệu, hệ số Linear Prediction và hệ số Linear Prediction Cepstral là những thuộc tính chính cho việc nhận diện giọng đọc tự động.

Tính toán ngân hàng bộ lọc và MFCCs về cơ bản là một quy trình, trong cả hai trường hợp, các ngân hàng bộ lọc được tính toán và với một vài bước bổ sung, có thể thu được MFCC. Nói đơn giản, tín hiệu đi qua một bộ lọc pre-emphasis, rồi được cắt thành các đoạn (đè lên nhau) và áp dụng hàm cửa sổ cho từng khung. Sau đó, chúng ta sử dụng biến đổi Fourier cho từng khung và tính quang phổ cường độ và ngân hàng bộ lọc. Để thu được MFCCs, biến đổi cosin rời rạc được áp dụng cho ngân hàng bộ lọc, chỉ giữ lại một số hệ số kết quả trong khi phần còn lại bị loại bỏ. Bước cuối cùng đều là chuẩn hóa.



Hình 2.5: Quá trình phép biến đổi MFCC

- Pre-emphasis: bước đầu tiên để áp dụng bộ lọc pre-emphasis là khuếch đại các tần số cao. Bộ lọc pre-emphasis hữu dụng trong một số trường hợp: (1) cân bằng quang phổ tần số bởi các tần sô cao thường có năng lượng nhỏ hơn các tần số thấp. (2) tránh được các lỗi tính toán khi dùng biến đổi Fourier và (3) có thể nâng cao tỉ lệ Signal-to-Noise (SNR). Bộ lọc pre-emphasis có thể được áp dụng cho tín hiệu x dùng với bộ lọc bậc một:

$$y(t) = x(t) - \alpha x(t-1)$$

Với hệ số bộ lọc a trong khoảng 0.95-0.97

- Chia khung: sau khi dùng bộ lọc pre-emphasis, chúng ta cần chia tín hiệu thành các phần bằng nhau. Lý do đằng sau việc đó là các tín hiệu thay đổi theo thời gian, do đó trong nhiều trường hợp sẽ không hợp lý khi dùng biến đổi Fourier trên toàn bộ tín hiệu vì nó sẽ gây mất dữ liệu. Để tránh việc đó, chúng ta có thể giả định một cách an toàn rằng các tần số trong tín hiệu là đứng yên trên khoảng thời gian rất ngắn. Do đó, khi sử dụng biến đổi Fourier trên khoảng thời gian này, chúng ta có thể có được một giá trị gần đúng của các đường viền tần số của tín hiệu bằng cách nối các khung liền kề. Độ lớn của khung thường trong khoảng 20ms-40ms với 50% (+/- 10%) đoạn trùng nhau giữa các khung liên tiếp.Thông thường sẽ chọn độ lớn 25ms cho khung và 10ms trùng nhau.
- Hàm cửa sổ: Sau khi cắt tín hiệu ra thành từng đoạn, ta áp dụng hàm cửa sổ cho từng đoạn, ví dụ như hàm Hamming. Hàm cửa sổ Hamming có dạng như sau:

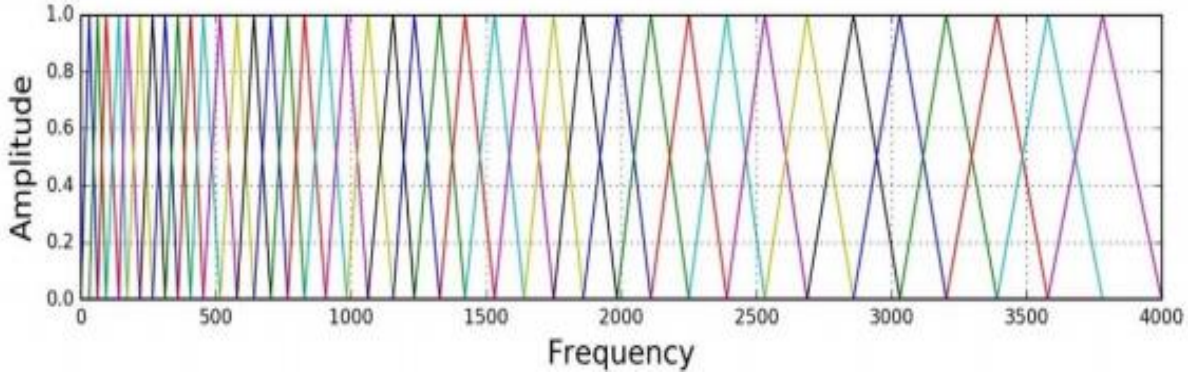
$$w[n] = 0.54 - 0.46cos\ (\frac{2\pi n}{N-1})$$

với 0 <= n <= N-1, N là độ dài cửa sổ

- Biến đổi Fourier và quang phổ công suất: chúng ta có thể áp dụng N-điểm FPT trên từng đoạn để tính quang phổ tần số, với N thường là 256 hoặc 512 và sau đó tính toán quang phổ công suất với phương trình sau:

$$m = 2595\left(1 + \frac{f}{700}\right)$$
$$f = 700\left(10^{\frac{m}{2595}} - 1\right)$$

Mỗi bộ lọc trong ngân hàng bộ lọc là một hình tam giác trả về 1 ở tần số trung tâm và giảm dần về 0 tới khi nó tiến đến tần số trung tâm của hai bộ lọc liền kề, nơi mà nó có giá trị 0. Ví dụ ở hình dưới

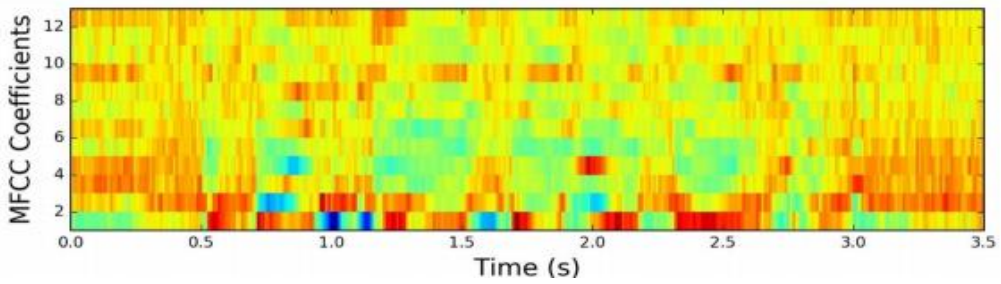


Hình 2.6: Minh họa cho ngân hàng bộ lọc MFCC [15]

Bộ lọc có thể được biểu diễn bởi hàm sau:

$$H_m(k) = \{\begin{matrix} 0, k < f(m-1) \\ \frac{k - f(m-1)}{f(m) - f(m-1)}, f(m-1) \le k \le f(m) \end{matrix} \quad 1, k$$
$$= f(m) \quad \begin{matrix} \frac{f(m+1) - k}{f(m+1) - f(m)}, f(m) < k \le f(m+1) \\ 0, k > f(m+1) \end{matrix}$$

- MFCC: chúng ta có thể áp dụng biến đổi Cosine rời rạc (DCT) để biểu diễn nén của bộ lọc ngân hàng. Thông thường, đối với nhận dạng giọng nói tự động (ASR), kết quả là hệ số cepstral 2-13 được giữ lại và phần còn lại bị loại bỏ.



Hình 2.7: Phép biến đổi Cosine rời rạc
(Nguồn:Tìm hiểu về phép biến đổi Cosine – Internet)

- Chuẩn hóa trung bình: Như đã nói ở trên, để cân bằng giữa quang phổ và tỉ lệ tín hiệu – nhiễu (Signal-to-Noise), chúng ta có thể trừ đi trung binh giữa các hệ số trong tất cả các khung.

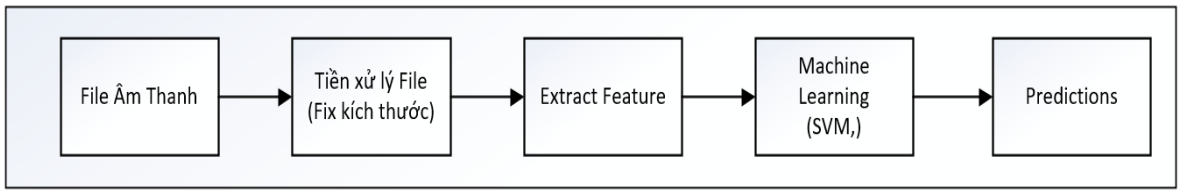
2.3. Giải pháp thường áp dụng để xây dựng mô hình phân loại âm thanh

2.3.1. Phương pháp sử dụng học máy truyền thống

❖ ***Mô hình***

Có khá nhiều thuật toán học máy truyền thống như: Logistic Regression, Support Vector Machine, K-Nearest Neighbor, ... được huấn luyện trên các đặc trưng được trích xuất từ tín hiệu âm thanh. Đặc trưng từ âm thanh sẽ được trích xuất thành các vector số khác nhau theo số chiều nhất định. Trích xuất đặc trưng là quá trình rất quan trọng, bởi mô hình không thể hiểu được dữ liệu trực tiếp từ các tệp âm thanh.

Quá trình thực thiện xây dựng mô hình phân loại âm thanh dựa trên học máy truyền thống được mô tả như hình dưới đây:



Có rất nhiều các đặc trưng có thể được sử dụng để mô tả dữ liệu âm thanh, tuy nhiên, trong phạm vi của bài luận này, các đặc trưng sẽ không được liệt kê và giải thích rõ ràng hết tất cả. Dưới đây là một số đặt trưng quan trọng được sử dụng rộng rãi:

- Các đặc trưng thống kê: tần số trung bình, độ lệch chuẩn của tần số, tần số trung vị, độ lệch, entropy của phổ, độ phẳng của phổ, mode của tần số, centroid của tần số, …
- Các đặc trưng về quang phổ: BarkBands, MelBands, ERBBands, MFCC, GFCC, LPC, HFC, tương phản phổ, …
- Mô tả tông của âm thanh: hàm cao độ, FFT âm độ, HPCP, tần số lên dây (Tuning Frequency), phát hiện hợp âm (Chords Detection), …
- Các đặc trưng miền thời gian: ZCR, độ to, khoảng thời gian hiệu quả (Effective Duration), …
- Đặc trưng về nhịp điệu: Beat Tracker Degara, Beat Tracker MultiFeature, Novelty Curve, Onset Detection, Onsets, …
- Đặc trưng SFX: LogAttackTime, MaxToTotal, MinToTotal.

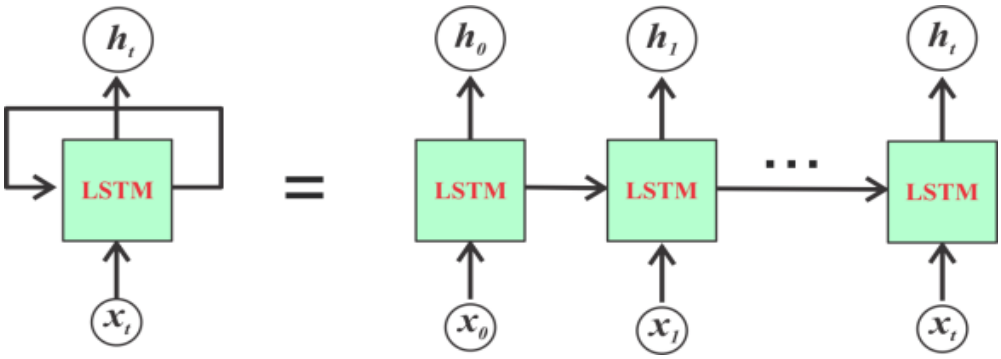
❖ ***Đánh giá***

Phương pháp tiếp cận bằng học máy cơ bản khó sử dụng, bởi các phương pháp này rất tốn tài nguyên cho bước trích xuất đặc trưng và các đặc trưng này rất nhạy cảm với độ nhiễu của dữ liệu âm thanh.

2.3.2. Phương pháp sử dụng bộ nhớ dài ngắn hạn (LSTM) với tín hiệu thô

❖ ***Mô hình***

Các kiến trúc LSTM thông thường khá tốt cho việc phát hiện ra các đặc điểm trong các tín hiệu, do đó phương pháp này có thể được dùng để tạo mô hình học máy cho tín hiệu âm thanh liên tục trên miền thời gian.



Hình 2.12: Mô hình hoạt động LSTM [11]

Một cách cụ thể, một tín hiệu âm thanh có thể được chia ra thành các phần trên miền thời gian, mỗi phần là mảng một chiều có độ dài cố đinh, có thể thêm phần đệm nếu không đủ độ dài. Giờ đây, mỗi tín hiệu âm thanh có thể được hiểu là một chuỗi các vectơ phân đoạn, các vectơ này sau đó được cấp cho LSTM để mô hình có thể xuất ra một vectơ tóm tắt cho toàn bộ âm thanh.

❖ ***Đánh giá***

LSTM trở nên phổ biến vì có thể giải quyết vấn đề mất mát gradient. Nhưng thực tế, LSTM không thể xử lý vấn đề này một cách hoàn hảo. Vấn đề nằm ở chỗ, dữ liệu đầu vào vẫn phải di chuyển từ cell này sang cell khác trong quá trình tính toán. Hơn nữa, mô hình sẽ trở nên khá phức tạp với việc bổ sung thêm các đặc trưng (chẳng hạn như cổng quên – forget gate).

LSTM đòi hỏi nhiều tài nguyên, thời gian để được huấn luyện và trở nên sẵn sàng cho các ứng dụng trong thế giới thực. Về mặt kỹ thuật, LSTM cần băng thông bộ nhớ cao do các lớp tuyến tính hiện diện trong mỗi ô mà hệ thống thường không cung cấp được.

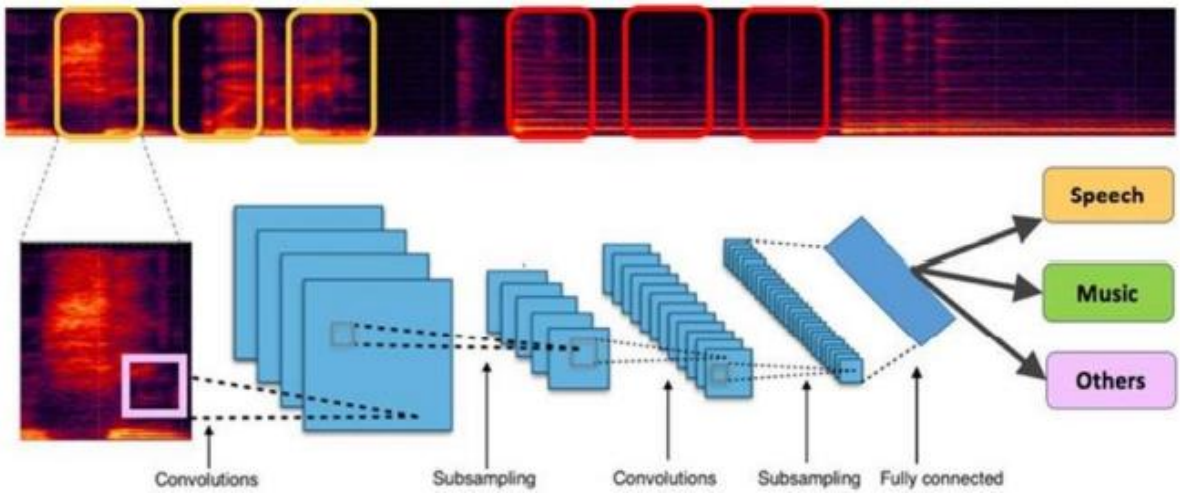
LSTM bị ảnh hưởng bởi các lần khởi tạo trọng số ngẫu nhiên khác nhau và do đó hoạt động khá giống với mạng nơ-ron truyền thẳng (feed – forward).

Các LSTM dễ bị overfitting và rất khó áp dụng thuật toán drop out để giải quyết vấn đề này.

2.3.3. Phương pháp sử dụng CNN với các đặc trưng về tần số

❖ ***Mô hình***

Theo cách tiếp cận này, dữ liệu âm thanh sẽ được trích xuất thành các đặc trưng dựa trên tần số như MFCC, biểu đồ phổ log-mel như đã thảo luận trong Chương 1. Sau đó, các đặc trưng này có thể được diễn giải dưới dạng hình ảnh và có thể được đưa vào các mô hình CNN.



Hình 2.13: Ảnh của âm thanh qua mô hình CNN [7]

Để được đưa vào các mô hình CNN, các đặc trưng dựa trên tần số có thể được giữ nguyên dưới dạng mảng 2 chiều, hoặc được chuyển đổi sang mảng 3 chiều để các mô hình CNN có thể xem chúng dưới dạng hình ảnh 3 kênh. Có một số cách phổ biến để hoàn thành nhiệm vụ này, cách thông thường là lặp lại từng phần tử, tiếp theo là các bước chuẩn hóa (các giá trị sẽ nằm trong khoảng [0, 255]); sử dụng thư viện của bên thứ ba (ví dụ: python matplotlib) để chuyển đổi và lưu chúng vào đĩa dưới dạng ảnh màu.

Về kiến trúc mạng sẽ được sử dụng, chúng ta có thể tạo một mạng CNN của riêng mình hoặc sử dụng kiến trúc CNN hiện đại như VGG, ResNet, Inception, v.v. Các mô hình này có thể được tinh chỉnh trọng số ImageNet để hội tụ nhanh hơn.

Nếu chuyển đổi các đặc trưng dựa trên tần số thành "hình ảnh màu", chúng ta có thể sử dụng các kỹ thuật nâng cao thường được áp dụng cho hình ảnh bình thường chẳng hạn như lật ngang hoặc lật dọc, cắt ngẫu nhiên, thêm nhiễu phụ gia trắng, biểu đồ chuẩn hóa đặc trưng, ...

❖ *Đánh giá*

Mạng nơ-ron tích chập thường được sử dụng để phân loại hình ảnh, chúng thường xử lý dữ liệu nhiều chiều (hình ảnh). Mặc dù cấu trúc của ConvNet nhằm mục

đích giảm thiểu sự overfitting, một lượng lớn dữ liệu vẫn là điều cần thiết để cho một mạng CNN hoạt động hiệu quả.

Vì vậy, luận văn lựa chọn phương pháp sử dụng CNN với các đặc trưng về tần số để xây dựng mô hình.

2.4. Kết luận chương 2

Trong chương 2, luận văn đã giới thiệu tổng quát cơ sở lý thuyết và các đặc trưng cơ bản của âm thanh trong các phương pháp phân loại âm thanh. Đưa ra được khái niệm "ảnh chụp của âm thanh" bằng các phép biến đổi từ miền âm thanh (liên tục) sang miền tần số (rời rạc). Đưa ra hướng giải quyết chuyển từ phân loại âm thanh sang phân loại hình ảnh ứng dụng các kiến trúc mạng học sâu tiến tiến hiện nay. Ngoài ra nội dung chương cũng đã nghiên cứu và đề cập phương pháp học chuyển tiếp, đưa ra nhưng điểm ưu việt của phương pháp này.

Nội dung cuối của chương, tác giả đã thực hiện so sánh các phương pháp tiếp cận khác nhau để giải quyết bài toán mô hình phân loại giới tính vùng miền. Trên cơ sở các kết quả đã đạt được của chương 2, trong chương tiếp theo luận văn sẽ tiến hành thực nghiệm xây dựng mô hình phân loại giới tính và vùng miền ứng dụng phương pháp học chuyển tiếp dựa trên kiến trúc DenseNet, ResNet.

CHƯƠNG 3: THỰC NGHIỆM VÀ ĐÁNH GIÁ

Tóm tắt chương: Trong chương 3, luận văn sẽ trình bày các bước triển khai tiền xử lý dữ liệu và xây dựng, huấn luyện mô hình cũng như các bước tinh chỉnh xử lý sau khi xây dựng mô hình, sau cùng là đánh giá độ chính xác của mô hình trên dữ liệu mới.

Có rất nhiều cách để giải quyết nhiệm vụ phân loại âm thanh, chúng khác nhau ở cách tín hiệu âm thanh được diễn giải và các đặc trưng được trích xuất, các mô hình được huấn luyện từ các đặc trưng này, từ các mô hình học máy truyền thống đến các mô hình học sâu với các loại kiến trúc khác nhau. Trong phần này, trước tiên ta sẽ thảo luận về ba cách tiếp cận chính đối với các nhiệm vụ phân loại âm thanh, mỗi cách có ưu điểm và nhược điểm khác nhau khi được áp dụng. Sau đó, thực hiện phân tích khai thác dữ liệu để chỉ ra các thách thức gặp phải, từ đó đưa ra các phương hướng để giải quyết khi xây dựng mô hình nhận dạng âm thanh

3.1. Giới thiệu về bộ dữ liệu âm thanh

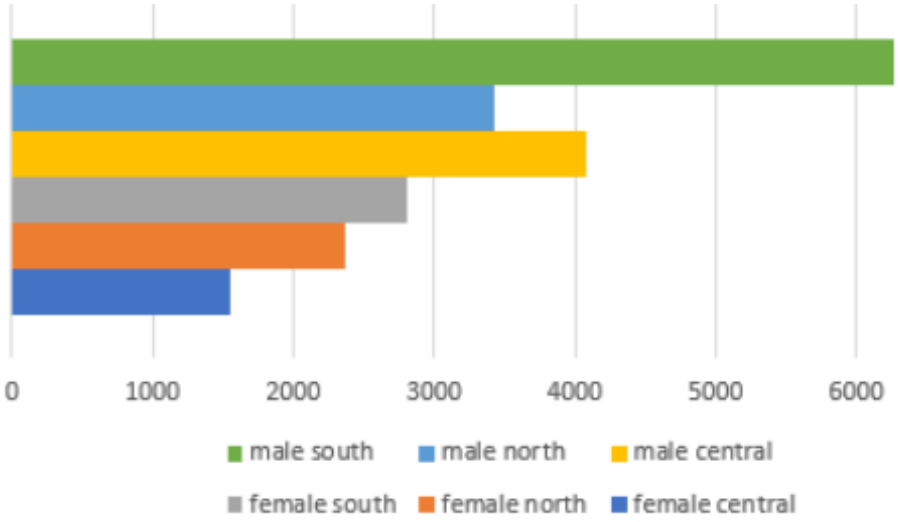
Để xây dựng một hệ thống có thể phân loại giới tính và giọng của người nói như đề xuất, ta sẽ sử dụng tập dữ liệu giọng nói từ Zalo AI Challenges. Tập dữ liệu này chứa 2 thư mục - "train", chứa 6 thư mục với các tệp âm thanh bên trong tương ứng với một trong 6 lớp “female_central”, “female_north”, …, “male_south”; và "test", cũng chứa các tập tin âm thanh. Tập dữ liệu cũng có thêm một tệp csv có 2 cột.

- “train”: chứa 6 thư mục “female_central”, “female_north”, …, "male_south"; mỗi tệp chứa các tệp âm thanh tương ứng với lớp.
- “public_test”: chứa các tệp âm thanh.
- “public_test_gt.csv”: chứa 2 cột, “id” - tên tệp trong public_test, “gender”: “0” (nữ) hoặc “1” (nam), “accent”: “0” (Bắc), “1” (Trung), “2” (Nam).

Dữ liệu trong tập “train” sẽ được sử dụng để huấn luyện mô hình và dữ liệu trong tập “public_test” sẽ được sử dụng để đánh giá mô hình.

Các tệp âm thanh trong tập dữ liệu có 3 định dạng: .wav, amr và .mp3 có thể thay đổi theo độ dài - các tệp trong ‘female_central’ chủ yếu dài 3 giây với tệp dài nhất là 19 giây. Trong khi đó, các tệp ở các lớp khác dao động từ dưới 1 giây đến hơn 200 giây hoặc thậm chí 500 giây như trong “female_south”.

Một vấn đề của tập dữ liệu này là nó rất mất cân bằng. Dữ liệu lớp ‘male_south’ lớn hơn gần 4 lần so với lớp các mẫu "female_central". Sự mất cân bằng này có thể được nhìn thấy trong biểu đồ dưới đây



Hình 3.1: Số lượng phân bố của dữ liệu trong tập mẫu

Các tệp âm thanh được ghi lại trong môi trường không được kiểm soát, tức là chúng có khả năng chứa tiếng ồn xung quanh và người nói khác nhau về độ tuổi, cảm xúc, v.v. Mỗi người nói có thể có nhiều tệp và theo quan sát, ID người nói có thể được suy ra từ tên tệp vì tên tệp chủ yếu ở dạng: [ID] _ [thứ tự khoảng thời gian], rất có thể một số tệp bị cắt (ngẫu nhiên) từ một cùng một âm thanh của một người.

Trong ví dụ dưới đây, 7 tệp đều là của người có ID ‘22da88321475487 8af2101120bbfb2ee__lWbAHVYAXI_160-181’.

Hình 3.2: Hình ảnh dữ liệu của cùng một người có trong "female_central"

Sau khi phân tích dữ liệu đầu vào, có thể thấy những vấn đề cần phải đối mặt khi sử dụng tập dữ liệu này:

- Đối với mỗi tệp, không thể sử dụng toàn bộ độ dài để chuyển đổi thành " hình ảnh của tần số" bởi vì chúng có độ dài rất khác nhau, và rõ ràng là không nên phân rã một tệp âm thanh dài mười phút từ miền thời gian thành các thành phần tần số của nó. Hơn nữa, nếu chỉ cắt ngẫu nhiên một khoảng thời gian có độ dài cố định từ một tệp thì sẽ thể không sử dụng tốt nhất tập dữ liệu.
- Sự mất cân bằng giữa các lớp dữ liêu có thể ảnh hưởng tới quá trình học và hiệu suất của mô hình.
- Mô hình được huấn luyện phải là mô hình không phụ thuộc người nói, tức là mô hình có thể dự đoán chính xác giọng nói của người mà mô hình chưa bao giờ được nghe trước đây. Việc tách ngẫu nhiên sẽ không hoạt động vì các mẫu từ cùng một người nói có thể xuất hiện trong cả tập huấn luyện và đánh giá, mô hình có xu hướng phát hiện ra các đặc trưng dành riêng cho người nói đó, trong khi mô hình cần được khái quát tốt hơn. Do đó, cách đánh giá mô hình phù hợp là chia theo id của người nói.
- Dữ liệu public_test là khá mất cân bằng lớp và dữ liệu xác thực cũng như vậy, nên thang đo độ chính xác có thể không có nhiều ý nghĩa. Điểm Micro F1 sẽ được sử dụng thay vì thang đo độ chính xác vì điểm Micro F1 có xét đến sự mất cân bằng lớp trong dữ liệu được sử dụng để đánh giá các mô hình.

3.2. Kịch bản xây dựng mô hình phân loại giới tính vùng miền

Điều đầu tiên cần làm là thiết lập một chiến lược đánh giá phù hợp. Đối với mỗi lớp, danh sách id người nói được lấy và chia theo tỷ lệ 8:2, đối với tập dữ liệu huấn luyện, kết quả sẽ xác thực bằng cách lấy các tập tương ứng với mỗi người nói. Làm như vậy, có thể đảm bảo rằng các tệp âm thanh của một người nói chỉ xuất hiện trong một trong các tập: huấn luyện hoặc xác thực. Hình sau trình bày tổng quan về các bước trong phương án vừa được đề xuất:

Hình 3.3: Quá trình xây dựng mô hình

Các tệp âm thanh gốc được chia thành nhiều phần khác nhau trên trục thời gian. Các kỹ thuật để tăng lượng dữ liệu trên miền thời gian như thay đổi tốc độ, trộn tiếng ồn cho các tệp âm thanh của các lớp có ít dữ liệu sẽ được áp dụng.

Các tệp âm thanh sau khi được phân mảnh sẽ được biến đổi dưới dạng quang phổ log-mel và được lưu lại thành ảnh màu. Các bước xử lý này sẽ được được áp dụng cho tập huấn luyện, tập kiểm thử và tập đánh giá.

Trước khi huấn luyện mô hình với các hình ảnh này, xử lý hậu kỳ được thực hiện trên các lớp sử dụng để huấn luyện dữ liệu, mục đích là để làm cân bằng dữ liệu. Đối với mỗi lớp, chỉ một phần của tập dữ liệu được lựa chọn. Điều này giúp cân bằng số lượng của các dữ liệu đã được xử lý ở mỗi lớp và có càng nhiều dữ liệu càng tốt.

Sau khi mô hình CNN được huấn luyện trên tập các hình ảnh của âm thanh, dự đoán của các phần bị phân mảnh sẽ được gộp lại thành dự đoán cho cả tệp âm thanh. Điểm micro F1 và độ chính xác được tính toán trên những dự đoán gộp này.

3.2.1. Tiền xử lý dữ liệu và trích xuất đặc trưng

Trong phần này, quy trình tiền xử lý được áp dụng cho dữ liệu huấn luyện, kiểm thử và đánh giá như đã đề cập ở trên sẽ được trình bày chi tiết.

- Tệp dữ liệu được chia thành các phần trên trục thời gian:
  - Từng tệp âm thanh với thời gian ngẫu nhiên được chia thành các phần với thời lượng 1.5 giây (phần sau đè lên 50% với phần trước cho lớp với ít dữ liệu “female_central”).
  - Với các lớp có ít dữ liệu, bắt đầu bằng việc chia ở các điểm ngẫu nhiên, ví dụ, khoảng giữa 0-2 giây và dừng lại ở điểm cuối trục thời gian. Tuy nhiên, đối với các lớp có nhiều dữ liệu nhất là “male_south”,chỉ sử dụng 3 giây để chia phần.
  - Dễ dàng nhận thấy sau khi thực hiện như trên, một lượng lớn các dữ liệu sẽ được thu thập, trong khi vẫn sử dụng được nhiều nhất dữ liệu sẵn có. Nếu tệp có độ dài hơn 1 phút, chỉ lấy mẫu ngẫu nhiên với khoảng thời gian 1.5 giây và sẽ làm mất rất nhiều thông tin. Nếu khoảng thời gian đó phần lớn là im lặng thì nó sẽ không mang được ý nghĩa.
- Tăng lượng dữ liệu trên miền thời gian:

  Bước này chỉ được áp dụng cho lớp “female_central” vì có ít dữ liệu nhất. Áp dụng thay đổi tốc độ (giảm hoặc tăng) của người nói và thêm tiếng ồn cho các phần này. Phần dữ liệu tăng thêm này không chỉ giúp với việc thăng thêm một lượng dữ liệu cho lớp (giúp giảm khoảng cách số lượng dữ liệu so với các lớp khác) mà nó còn giúp mô hình tổng quát hóa tốt hơn, giúp mô hình học được các đặc tính tổng quát như là mô hình sẽ không bị nhầm lẫn khi

người nói với tốc độ nhanh hoặc chậm hơn thông thường hoặc là có nhiễu ở môi trường.

- Biến đổi các phần âm thanh thành quang phổ log-mel với cấu hình như sau:

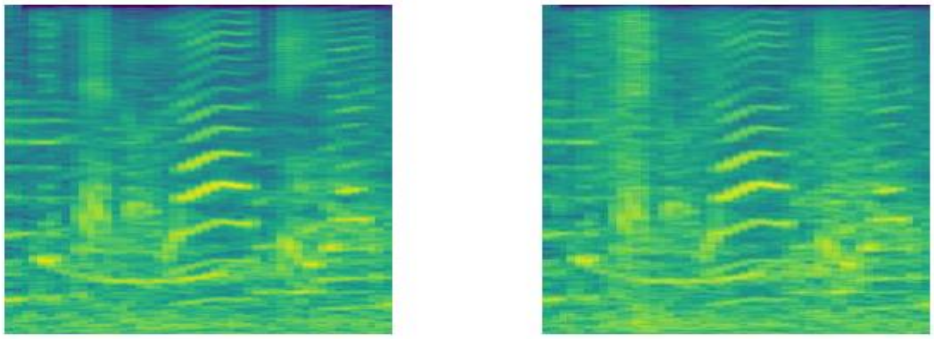
Bảng 3.1: Các thông số sử dụng trong phép biến đổi âm thanh

Sampling rate	8000 Hz
Manimum frequency	20 Hz
Maximum frequency	4000 Hz
Mel bands	224
Hop length	256
FFT window size	1024

Việc lấy mẫu cá file âm thanh ở tần số 8000 Hz là rất quan trọn bởi hệ thống của ta làm việc với tín hiệu GSM (được lấy mẫu ở tần số 8000 Hz). Ta không thể lấy mẫu tín hiệu GSM ở tần số cao hơn 8000 Hz, bởi vì nó sẽ làm thay đổi tần số thành phần và làm chúng thay đổi thành giọng khác hẳn.

Sau đó, các quang phổ log-mel được lưu lại thành các hình ảnh, làm như vậy, ta có quang phổ từ hình ảnh một kênh chuyển thành hình ảnh ba kênh.

Dưới đây là ví dụ cho quang phổ log-mel nhận được từ các phần âm thanh sau khi được áp dụng tăng lượng dữ liệu trên miền thời gian:



Hình 3.4: Ảnh chụp phổ của âm thanh gốc (bên trái) và ảnh của âm thanh sau khi thêm nhiễu (bên phải)

Sau quá trình xử lý hậu kỳ, số lượng các tệp hình ảnh của mỗi lớp như sau:

- “female_central”: 7914 tệp hình ảnh
- “female_north”: 8708 tệp hình ảnh
- “female_south”: 8331 tệp hình ảnh
- “male_central”: 6872 tệp hình ảnh
- “male_north”: 8589 tệp hình ảnh
- “male_south”: 8725 tệp hình ảnh

Có thể quan sát được rằng số lượng dữ liệu huấn luyện đã tăng lên 7 lần so với nguyên bản và lượng dữ liệu đã cân bằng hơn.

3.2.2. Làm giàu nguồn dữ liệu

Nói chung, trong các tác vụ phân loại hình ảnh, tăng cường dữ liệu trực tuyến được áp dụng trong quá trình huấn luyện, nó có thể mang lại lợi ích cho quá trình huấn luyện vì nếu được áp dụng đúng cách, nó sẽ làm cho dữ liệu huấn luyện hơi khác nhau từ vòng này qua vòng lặp khác, do đó, mô hình là bằng cách nào đó "buộc" phải học các mô hình tổng quát nhất cho mục tiêu, một cách hiệu quả giảm overfit. Vì bài toán bây giờ có thể được coi là nhiệm vụ phân loại hình ảnh, dữ liệu nên được tăng cường.

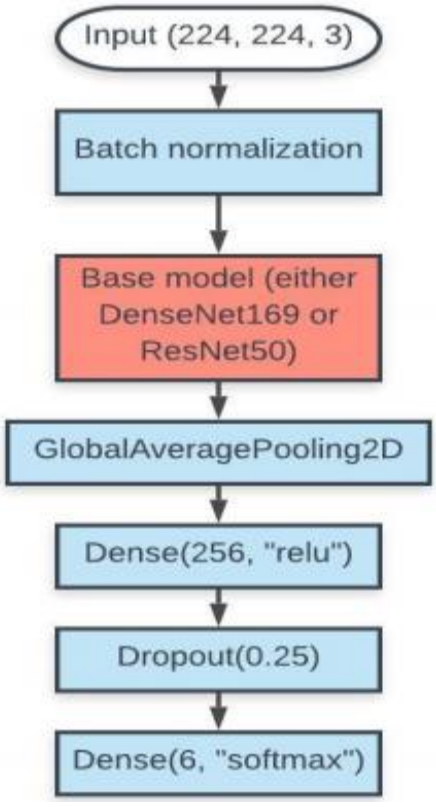
Trong thời gian huấn luyện, làm mờ Gaussian, chuẩn hóa độ tương phản, làm sắc nét đã được áp dụng cho dữ liệu huấn luyện. Trong mỗi vòng lặp, chỉ một trong các phương pháp gia tăng này được sử dụng. Trong thời gian suy luận, chỉ Gaussian Blur và Sharpen được áp dụng cho dữ liệu đầu vào.

3.2.3. Kiến trúc mạng áp dụng trong mô hình

Các kiến trúc mạng nơ-ron được trình bày trong hình 3.8. Lớp đầu vào với hình ảnh có kích thước (224, 224, 3) được theo sau bởi một lớp chuẩn hóa hàng loạt (batch normalization).

Batch normalization giúp tăng tốc độ huấn luyện, giảm số lượng epoch xuống gần một nửa. Batch normalization cũng giúp chuẩn hóa dữ liệu đầu vào, giảm lỗi của mô hình. Đầu vào của mỗi lớp có thể thay đổi sau những lần thay đổi trọng số. Trung bình và phương sai của đầu vào sẽ thay đổi trong quá trình mô hình học, vì thế sử dụng một lớp batch normalization ngay sau lớp đầu vào sẽ hữu dụng khi mà không cần thiết phải tính lại hai thông số này sau mỗi lô.

Phương pháp học chuyển tiếp (transfer learning) sẽ được áp dụng. Huấn luyện một mô hình học sâu cần rất nhiều dữ liệu và tài nguyên. Vậy nên ngoài kĩ thuật tăng cường dữ liệu, học chuyển tiếp là một giải pháp cho bài toán này.Một số kiến trúc CNN hiện đại đã được thử nghiệm làm mô hình cơ sở, ví dụ như DenseNet169, ResNet50.



Hình 3.5: Kiến trúc mạng học sâu để phân loại âm thanh đề xuất

Một vấn đề xảy ra khi xây dựng mạng CNN với nhiều lớp chập sẽ xảy ra hiện tượng *Vanishing Gradient* dẫn tới quá trình học tập không tốt, mạng ResNet sẽ giúp giải quyết vấn đề này.

DenseNet cũng được lựa chọn trong quá trình thử nghiệm vì DenseNet sử dụng lượng tham số chỉ bằng một nửa so với Resnet nhưng có độ chính xác trên ImageNet tương đồng. DenseNet có khả năng tránh bị overfitting hiệu quả và giảm được *Vanishing Gradient*.

Ba lớp trên cùng của mạng cơ sở sẽ được thay thế lần lượt bằng các lớp Global Average Pooling 2D, lớp dense với ReLU làm hàm kích hoạt, lớp dropout với xác suất 0,25 và cuối cùng một lớp dense với hàm kích hoạt softmax.

Global Average Pooling 2D giống với cấu trúc tích chập. Global Average Pooling 2D cho phép mạng chấp nhận bất kỳ kích thước Tensor/hình ảnh nào, thay vì chỉ kích thước mà nó đã được đào tạo ban đầu. Một ưu điểm nữa là không có tham số nào để tối ưu hóa trong Global Average Pooling 2D do đó tránh việc bị overfitting tại lớp này. Lớp dropout được sử dụng để giảm tình trạng overfitting.

3.2.4. Mô hình huấn luyện

Mạng được huấn luyện bởi mô hình cơ sở đã được tinh chỉnh với trọng số ImageNet được huấn luyện sẵn. Cụ thể hơn, tất cả các lớp ngoại trừ 4 lớp cuối cùng đã bị "đóng băng", 4 lớp trên cùng đã được thêm và chỉ huấn luyện 4 lớp này trong vòng lặp. Sau đó, tất cả các lớp đều được "phá băng" và toàn bộ mạng được đào tạo.

Hàm cross-entropy được sử dụng như hàm mất mát và Adam được sử dụng như một giải thuật gradient descent.

Quá trình huấn luyện được giám sát với điểm micro F1 cho tập dữ liệu kiểm thử. Bộ trọng số của mô hình được lựa chọn để cập nhật ở vòng lặp tiếp theo là bộ cho ra ma trận điểm tốt nhất.

Về mặt dự đoán, luận văn sử dụng đầu vào của mạng nơ-ron được phân chia thành các phần tương ứng với các phần của tệp âm thanh nguyên gốc ban đầu. Dự đoán của các phần được hợp lại thành dự đoán của tệp âm thạnh gốc. Bước này được thực hiện bằng cách lấy trung bình nhân, vì phương pháp này ít ảnh hưởng bởi các dữ liệu nhiễu hơn là lấy trung bình số học.

3.3. Cài đặt mô hình phân loại

3.3.1. Một số yêu cầu về cài đặt

❖ *Yêu cầu chung cho cài đặt thử nghiệm*

- *Phần cứng:* bộ xử lý 32bit (x86) hoặc 64bit (x64) có tốc độ 2 gigahertz (GHz) hoặc nhanh hơn; RAM 16GB trở lên; GPU Nvidia GTX 1080 Ti 8GB; đĩa cứng có dung lượng trống 100 GB.

- *Phần mềm:* cài đặt trên hệ thống Windows/Linux (Centos 7.2); Cuda; công cụ lập trình: phần mềm Python 2.7 trở lên hoặc phần mềm Pycham Professional 2020.1.

- *Dữ liệu:* bộ dữ liệu âm thanh về giọng nói, vùng miền do Zalo cung cấp.

❖ **Thư viện Pytorch**

Pytorch là một thư viện học máy mã nguồn mở cho Python, được xây dựng trên ngôn ngữ lập trình Lua bởi Facebook. Pytorch có hai chức năng chính: tính toán hiệu năn cao với sự tăng tốc của GPU, là một nền tảng học sâu phục vụ cho nghiên cứu, xây dựng và huấn luyện các mạng nơ-ron, mang lại sự linh hoạt và tốc độ.

So với Tensorflow2, Pytorch mang lại khác nhiều ưu điểm. Đầu tiên, Pytorch mang lại khả năng debug (sửa lỗi) dễ dàng hơn theo hướng trực quan. Bên cạnh đó, nếu như ở Tensorflow, trước tiên ta cần xác định toàn bộ biểu đồ tính toán trước khi có thể chạy mô hình thì với Pytorch, nó cho phép ta xác định một biểu đồ tính toán động. Cuối cùng, Pytorch cũng hỗ trợ cả API cấp cao và API cấp thấp, với việc sử dụng tương đối dễ dàng.

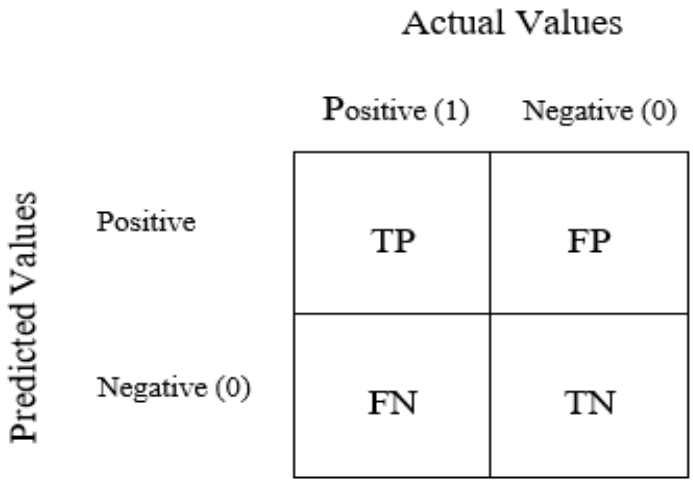
3.3.2. Phương pháp đánh giá

Trong luận văn này, để đánh giá độ chính xác của mô hình tác giả đã lựa chọn phương pháp đánh giá độ chính xác bằng cách sử dụng ma trận độ đo (confusion matrix) được mô tả như sau:

Confusion Matrix là một phương pháp đánh giá kết quả của những bài toán phân loại với việc xem xét cả những chỉ số về độ chính xác và độ bao quát của các

dự đoán cho từng lớp. Một confusion matrix gồm 4 chỉ số sau đối với mỗi lớp phân loại:

- **TP (True Positive)**: mẫu mang nhãn dương được phân lớp đúng vào lớp dương
- **TN (True Negative)**: mẫu mang nhãn âm được phân lớp đúng vào lớp âm.
- **FP (False Positive - Type 1 Error)**: mẫu mang nhãn âm bị phân lớp sai vào lớp dương.
- **FN (False Negative - Type 2 Error)**: mẫu mang nhãn dương bị phân lớp sai vào lớp âm.



Hình 3.6: Ma trận độ đo (Conusion matrix)

Ký hiệu TP là True Positive; TN là True Negative; FP là False Positive và FN là False Negative. Thực hiện phép đo Precision – Recall, trong đó, Precision là tỉ lệ số điểm TP trong những điểm được phân loại Positive, còn Recall là tỉ lệ số điểm TP trong số điểm thực sự là Positive. Công thức như sau:

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

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

Ta thấy rằng, Precision và Recall phủ càng cao thì càng tốt. Nhưng trong thực tế, hai giá này không thể đạt được cực đại cùng một lúc và thông thường phải tìm kiếm sự cân bằng. Thước đo $F1_{score}$ là trung bình hài hòa giữa Precision và Recall. Nó có xu hướng bằng không nếu hai giá trị này có xu hướng bằng không.

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

- **Micro - F1 Score**

Micro – F1 Score Tính toán các chỉ số toàn cục bằng cách đếm tổng số TP, FN và FP.

$$Micro - Average\ precision = \frac{\sum_{c=1}^{C} TP_C}{\sum_{c=1}^{C} (TP_C + FP_C)}$$

$$Micro - Average\ recall = \frac{\sum_{c=1}^{C} TP_C}{\sum_{c=1}^{C} (TP_C + FN_C)}$$

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

3.3.3. Kết quả của thử nghiệm

Dưới đây là kết quả ta thu được khi sử dụng mạng Densenet 169 và mạng ResNet50 sau 15 vòng huấn luyện cho mỗi mô hình.

Đánh giá dựa trên các dự đoán không được thống nhất (lựa chọn ngẫu nhiên dự đoán từ một phần của tập tin để dự đoán toàn bộ). Kết quả được sử dụng micro f1 để đánh giá vì trong tập dữ liệu test, số lượng tập mẫu ở các lớp lệch nhau rất nhiều, ví dụ như tập mẫu đo đươc của Nữ miền Trung ít hơn rất nhiều so với của Nam miền Nam. Chính vì vậy sử dụng độ đo micro f1 sẽ phản ánh đúng chất lượng của mô hình dự đoán thay vì dùng accuracy hoặc độ đo f1 từng lớp.

Bảng 3.2: Kết quả xây dựng mô hình dựa trên F1-Score

Dữ liệu	**Đánh giá**	**DenseNet169**	**ResNet50**
vKiểm thử	Độ chính xác tổng	0.78	0.76
	Độ chính xác giới tính	0.96	0.96
	Độ chính xác giọng	0.80	0.78
	Micro F1 tổng	0.78	0.76
	Micro F1 giới tính	0.93	0.93
	Micro F1 giọng	0.80	0.78
Đánh giá	Độ chính xác tổng	0.72	0.73
	Độ chính xác giới tính	0.95	0.96
	Độ chính xác giọng	0.75	0.75
	Micro F1 tổng	0.72	0.73
	Micro F1 giới tính	0.89	0.89
	Micro F1 giọng	0.75	0.75

Đánh giá dựa trên dự đoán được thống nhất bằng phương pháp trung bình nhân các kết quả của từng phần:

Bảng 3.3: Kết quả xây dựng mô hình dựa trên Micro - F1 Score

Dữ liệu	**Đánh giá**	**DenseNet169**	**ResNet50**
Kiểm thử	Độ chính xác tổng	0.81	0.80
	Độ chính xác giới tính	0.97	0.97
	Độ chính xác giọng	0.83	0.82
	Micro F1 tổng	0.81	0.80
	Micro F1 giới tính	0.95	0.95
	Micro F1 giọng	0.83	0.82
Đánh giá	Độ chính xác tổng	0.78	0.78
	Độ chính xác giới tính	0.97	0.97
	Độ chính xác giọng	0.80	0.80
	Micro F1 tổng	0.79	0.78
	Micro F1 giới tính	0.92	0.92
	Micro F1 giọng	0.80	0.80

Dựa vào kết quả của, ta có thể quan sát được rằng, với phương pháp tổng hợp các dự đoán, kết quả được cải thiện và DenseNet169 có độ chính xác cao hơn ResNet50. Cả 2 mô hình đều có điểm số cao với phân loại giới tính và điểm chấp nhận được với phân loại giọng vùng miền.

Kết quả chưa tốt ở mô hình phân loại vùng miền nguyên nhân chính trong dữ liệu có nhiều tập file vùng miền bị đánh nhãn sai, các tập file là giọng hát thay vì giọng nói bình thường, hoặc việc nhập nhằng giữa các vùng miền ví dụ như đánh nhãn giọng Thanh Hoá là giọng miền Trung nhưng giọng Thanh Hoá lại giống giọng Bắc hơn chứ ko nặng như Nghệ An, Hà Tĩnh. Để cải thiện mô hình tác giả đề xuất tập trung cải thiện chủ yếu ở phần dữ liệu: thu thập thêm nhiều dữ liệu, đánh nhãn lại những file bị đánh sai,....

3.4. Kết luận chương 3

Chương 3 đã trình bày quá trình thu thập, xử lý dữ liệu chuyển đổi từ âm thanh sang ảnh phục vụ cho quá trình huấn luyện mô hình. Từ bộ dữ liệu này, luận văn xây dựng và huấn luyện mô hình phân loại giới tính và vùng miền pretrain model là Resnet 50 và DenseNet169.

Việc sử dụng mô hình nhận dạng giới tính đã cho kết quả khả quan và có độ chính xác khá cao. Đối với nhận diện vùng miền kết quả chỉ ở mức chấp nhận được tuy nhiên các kết quả thử nghiệm ban đầu cho thấy giải pháp xây dựng mô hình phân loại giới tính và vùng miền cho tiếng việt dựa trên âm thanh đề xuất có tính khả thi cao và phù hợp với các yêu cầu đề ra.

KẾT LUẬN

1. Những đóng góp của luận văn

Với mục tiêu nghiên cứu các phương pháp xử lý dữ liệu giọng nói từ đó xây dựng mô hình học sâu phân loại giới tính và vùng miền cho tiếng việt, luận văn đã đi sâu nghiên cứu các vấn đề xung quanh đề tài nghiên cứu, từ các thuật toán học máy cổ điển đến các mô hình mạng nơ-ron tiến tiến hiện nay.

Những kết quả chính đã đạt được trong luận văn:

- Nghiên cứu về học máy cổ điển và các mô hình học sâu
- Tìm hiểu về các cơ sở lý thuyết về âm thanh, từ đó đưa ra phương pháp trích chọn thuộc tính từ giọng nói
- Lựa chọn và áp dụng thành công mô hình học chuyển tiếp ứng dụng các kiến trúc mạng nâng cao như ResNet, DenseNet
- Thực nghiệm cài đặt và xây dựng mô hình phân loại giọng nói vùng miền

2. Hướng phát triển của luận văn

Một số hướng phát triển tiếp theo của luận văn:

- Hiện tại tốc độ xử lý phân loại còn chưa nhanh, cần thử nghiệm thêm về các tham số trong mô hình nhằm giảm thời gian của việc phân loại
- Mặc dù ở bước tiền xử lý dữ liệu đã nêu ra được hướng xử lý âm thanh trong môi trường chứa nhiễu, tuy nhiên trong thực tế nếu ở môi trường nhiễu thì hệ thống hoạt động không ổn định.
- Kết quả của mô hình cho thấy việc phân loại vùng miền (Bắc-Trung-Nam) có độ chính xác chưa cao, cần thu thập, làm giàu thêm dữ liệu tập huấn luyện để tăng độ chính xác của mô hình.

DANH MỤC CÁC TÀI LIỆU THAM KHẢO

Tiếng Việt

[1] Vũ Hữu Tiệp (2016-2020) – "Machine Learning cơ bản".

Tiếng Anh

[2] Ratnadeep R. Deshmukh, "Comparative Study of Isolated Word Recognition System for Hindi Language", International Journal of Engineering and Technical Research, 2015.

[3] Geoffrey E. et al, "ImageNet Classification with Deep Convolutional Networks", Proceeding NIPS'12 Proceedings of the 25th International Conference on Neural Information Processing Systems, 2012, Volume 1, Pages 1097-1105.

[4] Gao Huang et al, "Densely Connected Convolutional Networks", IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017.

[5] Kaiming He et al, "Deep Residual Learning for Image Recognition", IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016.

[6] Karen Simonyan et al, "Very deep convolutional networks for large-scale image recognition", 3rd IAPR Asian Conference on Pattern Recognition (ACPR), 2015.

[7] Christian Szegedy et al, "Going deeper with convolutions", IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2015.

[8] Matthew D. Zeiler et al, "Visualizing and Understanding Convolutional Neural Networks", European Conference on Computer Vision, 2014, pp 818-833.

[9] Gröchenig K et al, "The Short-Time Fourier Transform", Foundations of Time-Frequency Analysis, 2001.

[10] K. M. M. Prabhu, Window Functions and their Applications in Signal Processing, 2018

Trang web

[11] http://research.cs.tamu.edu/prism/lectures/sp/l6.pdf

[12] https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53

[13] https://medium.com/@14prakash/understanding-and-implementing architectures-of-resnet-and-resnext-for-state-of-the-art-image-cf51669e1624

[14] https://github.com/davisking/dlib

[15] https://www.researchgate.net/publication/278685717_Speech_Feature_Extraction_Using_Mel-Frequency_Cepstral_Coefficient_MFCC

[16] https://github.com/SuperKogito/Voice-based-gender-recognition

[17] http://cs231n.github.io/neural-networks-2/