Vietnam National University, Ho Chi Minh City
University of Technology
Faculty of Computer Science and Engineering

ARTIFICIAL INTELLIGENCE PROJECT

IMPROVE ASR IN NOISY ENVIRONMENT

Advisor: Võ Thanh Hùng

Student: Nguyễn Anh Tấn – 2114741

Ho Chi Minh City, 3/2024

Contents

List of Figures

Lịch trình

Bảng dưới đây mô tả lịch trình thực hiện project từ ngày 17/03 đến 19/05.

Week	Time	Describe
1	17/03 - 23/03	Tìm hiểu project
2	24/03 - 30/03	Học Tensorflow
3	31/03 - 06/04	Hiện thực quy trình chuyển file âm thanh sang spectogram
4	07/04 - 13/04	Code training cho Convolutional Neural Network
5	14/04 - 20/04	Code training cho Convolutional Neural Network
6	21/04 - 27/04	Code training cho Convolutional Neural Network
7	28/04 - 04/05	Code training cho Convolutional Neural Network
8	05/05 - 11/05	Hoàn thiện báo cáo
9	12/05 - 19/05	Tuần dự trữ, điều chỉnh những thứ chưa hoàn thiện

Danh mục các từ viết tắt

Word	Describe
CNN	Convolutional Neural Network
SNR	Signal to Noise Ratio

1 Tổng quan

1.1 Giới thiệu

Hiện nay, lĩnh vực trí tuệ nhân tạo đang phát triển mạnh mẽ và nhận dạng giọng nói là một lĩnh vực nghiên cứu đầy tiềm năng và ứng dụng rộng rãi.

Nhận dạng giọng nói có thể được áp dụng trong nhiều lĩnh vực khác nhau, từ đời sống hàng ngày đến công việc chuyên môn, công nghệ này giúp tạo ra các trợ lý ảo như Siri, Google Assistant, và Alexa, giúp người dùng thực hiện các tác vụ hàng ngày một cách tiện lợi chỉ bằng giọng nói, hỗ trợ bệnh nhân và người cao tuổi trong việc sử dụng các thiết bị y tế và nhắc nhở uống thuốc, cung cấp các công cụ hỗ trợ học tập ngôn ngữ, kiểm tra phát âm, và giảng dạy từ xa hiệu quả hơn, cải thiện hiệu suất và chất lượng dịch vụ trong các trung tâm chăm sóc khách hàng thông qua hệ thống trả lời tự động và tổng đài thông minh, nhận dạng giọng nói có thể được sử dụng như một phương thức xác thực sinh trắc học để tăng cường an ninh cho các hệ thống...

Những tiến bộ gần đây trong trí tuệ nhân tạo (AI) và học sâu (deep learning) đã mang lại những bước đột phá lớn trong lĩnh vực nhận dạng giọng nói. Các mô hình như mạng nơ-ron hồi tiếp (RNN), mạng nơ-ron tích chập (CNN), và Transformers đã giúp cải thiện đáng kể độ chính xác và hiệu suất của các hệ thống nhận dạng giọng nói, sự sẵn có của lượng dữ liệu âm thanh khổng lồ đã cung cấp cơ sở dữ liệu phong phú để huấn luyện các mô hình nhận dạng giọng nói, việc xử lý và phân tích dữ liệu âm thanh trên đám mây đã giúp mở rộng khả năng và quy mô của các ứng dụng nhận dạng giọng nói.

Bên cạnh đó lĩnh vực nhận dạng giọng nói vẫn còn nhiều thách thức cần được giải quyết, mang lại nhiều cơ hội cho nghiên cứu và phát triển, ví dụ như cải thiện độ chính xác và khả năng hiểu ngữ cảnh, đặc biệt là trong các môi trường ồn ào và với các ngôn ngữ và phương ngữ khác nhau, nghiên cứu các phương pháp tối ưu hóa mô hình để giảm bớt tài nguyên tính toán và thời gian xử lý, đảm bảo rằng các hệ thống nhận dạng giọng nói tuân thủ các tiêu chuẩn bảo mật và quyền riêng tư cao.

Trong bài báo cáo này tác giả sẽ nghiên cứu và xây dựng một mô hình đơn giản **"Nhận dạng giọng nói trong môi trường nhiễu"**

1.2 Mục tiêu

Phát triển một ứng dụng có khả năng nhận dạng được một số âm thanh với độ chính xác tương đối được tính toán thông qua tập dữ liệu giọng nói Speech Command Musan dataset.

1.3 Mục tiêu

Ứng dụng được thiết kế và xây dựng cơ bản để nhận dạng âm thanh ngắn khoảng 1 giây với 35 từ thông dụng.

1.4 Phương pháp nghiên cứu

Các phương pháp nghiên cứu được áp dụng trong quá trình hoàn thành đề tài bao gồm phương pháp thu thập số liệu, phương pháp phân tích và tổng hợp lý thuyết, phương pháp phân loại và hệ thống hoá lý thuyết, phương pháp thực nghiệm khoa học, phương pháp phân tích và tổng kết kinh nghiệm, phương pháp chuyên giá, phương pháp giả thuyết...

1.5 Đối tượng và phạm vi nghiên cứu

Đề tài sẽ tập trung nghiên cứu trọng tâm vào lý thuyết mô hình mạng neural tích chập CNN, lý thuyết cơ bản đặc trưng của tín hiệu âm thanh, tập dữ liệu nhận dạng giọng nói Speech Command Musan dataset, Từ đó ứng dụng các lý thuyết nghiên cứu để xây dựng một ứng dụng nhận dạng giọng nói đơn giản từ mô hình mạng CNN bằng ngôn ngữ python.

1.6 Bố cục

Bố cục của bài báo cáo gồm 6 chương:

- 1. Tổng quan
- 2. Cơ sở lý thuyết
- 3. Mô hình
- 4. Hiện thực
- 5. Kết quả
- 6. Kết luận

2 Cơ sở lý thuyết

2.1 Mạng neural tích chập

Mạng neural tích chập (Convolutional Neural Network) là một loại mạng neural nhân tạo (Artificial Neural Networks), đã được chứng minh là có hiệu suất cao trên các tác vụ hình ảnh khác nhau, bao gồm phân loại hình ảnh, phân đoạn hình ảnh, truy xuất hình ảnh, phát hiện đối tượng, chú thích hình ảnh, nhận dạng khuôn mặt, ước lượng tư thế, nhận dạng dấu hiệu, xử lý giọng nói, ...

CNN được thiết kế để xử lý dữ liệu dưới dạng nhiều mảng, ví dụ, một hình ảnh màu được tạo ra từ ba mảng 2D chứa cường độ pixel trong các kênh ba màu. Sử dụng các ống kính tích tụ của chúng để trích xuất thông tin từ hình ảnh, các lớp trước đó phát hiện các cạnh, các lớp sau có thể phát hiện một phần của đối tượng, sau đó thậm chí các lớp sau có thể phát hiện các đối tượng hoàn chỉnh, chẳng hạn như khuôn mặt hoặc các hình dạng hình học phức tạp khác. CNN được cấu tạo bởi một tập hợp các lớp có thể được nhóm lại theo chức năng của chúng, ba loại lớp chính là: convolutional layer (lớp tích chập), pooling layer(lớp gộp lại) và fully-connected layer (lớp kết nối đầy đủ).

2.1.1 Convolutional Layer

Tích chập là một trong những nền tảng cơ bản của mạng CNN. Các tham số của convolutional layer bao gồm một tập hợp bộ lọc có thể học được (kernel). Mọi bộ lọc đều nhỏ về mặt không gian (dọc theo chiều rộng và chiều cao), nhưng mở rộng toàn bộ độ sâu của khối lượng đầu vào. Kích thước bộ lọc điển hình có thể là 3x3, 5x5, 7x7. Kích thước thứ ba của bộ lọc tương ứng với số kênh của đầu vào. Các độ sâu hình ảnh xám là 1 và hình ảnh màu có 3 (RGB) kênh màu.

Hình 1 bên dưới thể hiện một ví dụ cơ bản về tích chập trong lớp convolution layer, ta có một ảnh với kích thước 5x5

$$\begin{bmatrix} 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 \end{bmatrix}$$

Được tích chập với filter có kích thước 3x3

$$\begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix}$$

Hình trên cùng bên trái sẽ thực hiện việc nhân từng phần tử tương ứng nhau của filter và ma trận con 3x3 (ba cột đầu tiên 0, 1, 2 và ba hàng đầu tiên 0, 1, 2) của ảnh đầu vào và sau đó tính tổng kết quả. Tương đương với phép tính 1x1 + 1x0 + 1x1 + 0x0 + 1x1 + 1x0 + 0x1 + 0x0 + 1x1 do đó thu được kết quả là 4. Tương tự vậy thực hiện tiếp 8 phép tích chập filter với từng ma trận con khi dịch 1 cột hoặc 1 hàng so với ma trận liền kề trước nó thu được ảnh đầu ra là hình dưới cùng bên phải.

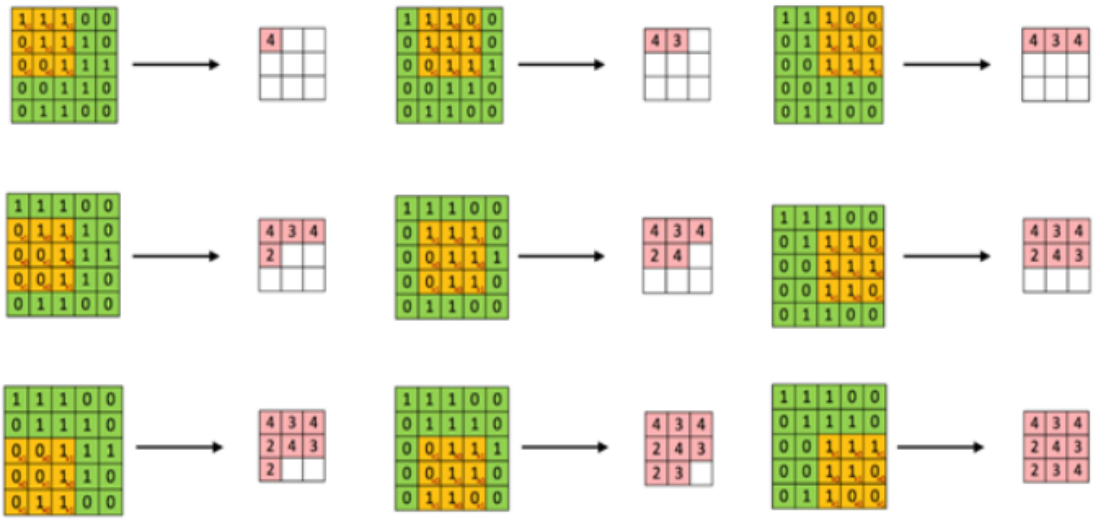

Figure 1: Tính toán tích chập của Convolution layer (nguồn: topdev.vn)

Trong lan truyền thẳng, mỗi bộ lọc thực hiện tích chập trên khối đầu vào theo chiều rộng, chiều cao và tính các kết quả tích chập của bộ lọc và đầu vào ở mọi vị trí. Việc tính toán này được theo sau bởi một hàm kích hoạt phi tuyến (sigmoid, tanh, ReLU, v.v.), kết quả đầu ra được gọi là bản đồ đặc trưng (featuremap). Khối đầu ra phụ thuộc vào ba tham số: độ sâu (depth), stride và padding.

- Độ sâu của khối đầu ra đại diện cho số lượng bộ lọc được sử dụngtrong tính toán tích chập. Mỗi bộ lọc đang học một cái gì đó khác nhau về đầu vào, cạnh, đốm màu, màu sắc.
- Stride là số bước trượt bộ lọc trên đầu vào. Khi stride là 1 thì sẽ di chuyển các bộ lọc từng pixel một. Khi stride là 2 thì các bộ lọc nhảy 2 pixel tại một thời điểm. Stride bằng 2 sẽ làm giảm kích thước không gian của khối đầu ra.
- Padding cho phép kiểm soát kích thước đầu ra. Áp dụng tích chập cho một đầu vào nếu giảm kích thước đầu ra dễ dẫn đến mất thông tin. Để tránh điều đó, thông thường sẽ đệm khối đầu vào bằng các số không xung quanh biên. Hai lựa chọn phổ biến là valid convolution và same convolution. Trong đó, valid convolution có nghĩa là không có padding, còn khi sử dụng same convolution thì kích thước đầu ra vẫn giống như kích thước đầu vào.

Kích thước khối ngỏ ra sẽ được tính toán theo công thức:

$$\frac{n - f + 2p}{s} + 1$$

Trong đó, n là kích thước khối đầu vào, f là kích thước bộ lọc, p là padding và stride là s.
Sử dụng công thức để tính toán kích thước ảnh đầu ra của hình 1 phía trên, ta có kích thước ảnh đầu vào là n = 5, kích thước filter f = 3, padding p = 0 và stride s = 1; kết quả là 3, đối chiếu với hình 1, ta có kích thước trong hình và kết quả tính toán tương đương nhau.

2.1.2 Pooling Layer

CNN thường sử dụng lớp pooling sau các lớp tích chập, chức năng của nó là giảm kích thước của khối. Các tham số của lớp pooling bao gồm kích thước bộ lọc và Stride. Pooling layer được sử dụng phổ biến nhất là với bộ lọc có kích thước 2x2 và có stride là 2. Hai dạng pooling layer phổ biến là max pooling layer và averge pooling layer, trong đó giá trị lớn nhất và giá trị trung bình được lấy tương ứng. Max pooling layer được sử dụng phổ biến hơn averge pooling layer.

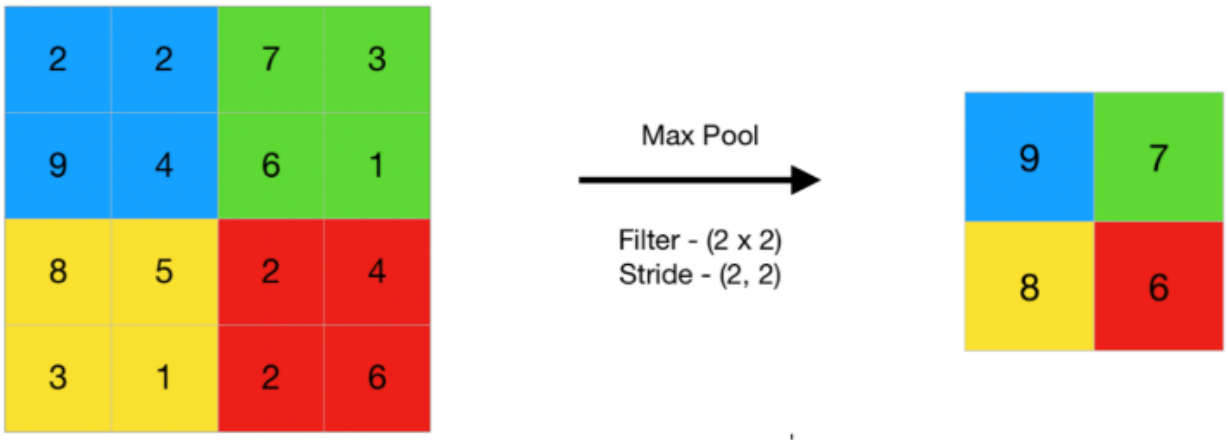

Figure 2: Max pooling layer (nguồn: www.geeksforgeeks.org)

Theo như hình 2, có thể dễ dàng nhận thấy rằng max pooling layer sẽ làm giảm kích thước của khối ngõ vào bằng cách lấy một giá trị lớn nhất tượng trưng cho toàn bộ lọc: 9 là giá trị lớn nhất của tập {2; 2; 9; 4}, 7 là giá trị lớn nhất của tập {7, 3, 6, 1}, 8 là giá trị lớn nhất của tập {8; 5; 3; 1} và 6 là giá trị lớn nhất của tập {2; 4; 2; 6}. Do bước trượt Stride bằng với kích thước bộ lọc nên các filter không chồng lấn lên nhau. Kích thước ngõ ra cũng có thể xác định bằng công thức tính kích thước ngõ ra của lớp tích chập.

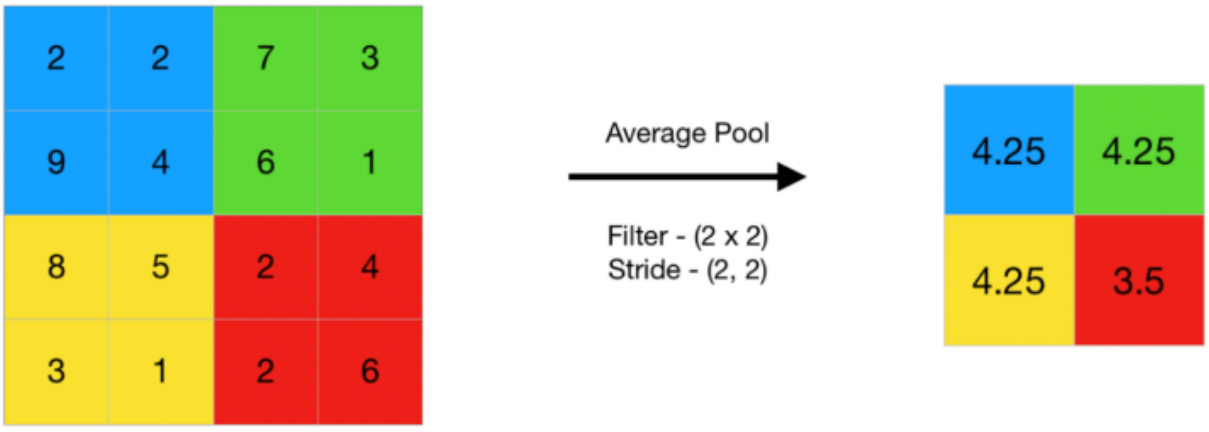

Figure 3: Average pooling layer (nguồn: www.geeksforgeeks.org)

Hình 3 thể hiện ví dụ một lớp pooling trung bình khi giá trị không lấy lớn nhất như ở hình 2 mà sẽ được tính trung bình cộng tất cả các số trong mỗi filter tương ứng. Và sử dụng chúng để đại diện cho cả bộ lọc đó. Với 4.25 là giá trị trung bình của ba cặp bốn số 2, 2, 9, 4; 7, 3, 6, 1 và 8, 5, 3, 1; 3.5 sẽ là giá trị trung bình của bốn số 2, 4, 2, 6.

2.1.3 Fully Connected Layer

Sau một số lớp tích chập và pooling, CNN thường kết thúc với một số lớp kết nối đầy đủ. Các lớp kết nối đầy đủ thường là một vài lớp cuối cùng của kiến trúc và kỹ thuật chính quy hoá dropout có thể được áp dụng trong các được kết nối đầy đủ để ngăn chặn overfitting. Lớp kết nối đầy đủ cuối cùng trong kiến trúcchứa cùng số lượng neural đầu ra như số lớp được công nhận.

Fully-connected layer tương tự như cách sắp xếp các nơ-ron trong một mạng nơ-ron truyền thống. Do đó, mỗi nút trong một lớp fully-connected layer được kết nối trực tiếp với mọi nút trong cả lớp trước và trong lớp tiếp theo.

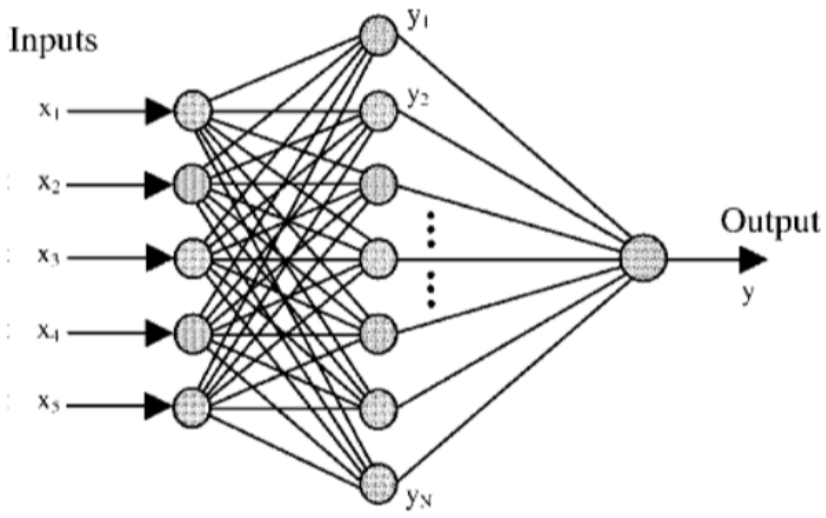

Figure 4: Fully-connected layer (nguồn www.researchgate.net)

Hình 4 thể hiện một lớp fully-connected đơn giản với tất cả các nút đều được liên kết trực tiếp với các nút ở tầng liền kề trước và sau nó.

Hạn chế chính của lớp kết nối đầy đủ là nó bao gồm rất nhiều tham số cần tính toán phức tạp trong các ví dụ huấn luyện. Do đó, cần cố gắng loại bỏ số lượng các nút và kết nối.

2.2 Speech Command-Musan dataset

2.2.1 Tổng quan

Tập dữ liệu Speech Command dataset là một tập dữ liệu tiêu chuẩn để huấn luyện và ước lượng cho những tác vụ nhận diện giọng nói đơn giản. Mục tiêu chính là cung cấp một cách để xây dựng và đánh giá một mô hình nhỏ để phát hiện khi một từ đơn giản được phát ra từ một tập hợp của một vài các từ đích (target) với ít lỗi sai.

Tập dữ liệu này được kết hợp với tập dữ liệu MUSAN (Music, Speech, and Noise). MUSAN chứa các mẫu âm thanh của âm nhạc, tiếng nói và tiếng ồn từ nhiều nguồn khác nhau. Khi kết hợp với Speech Commands, người ta có thể tạo ra các dữ liệu phức tạp hơn, giúp mô hình học được từ nhiều điều kiện môi trường khác nhau.

Để tiếp cận nhiều đối tượng hơn bao gồm các nhà nghiên cứu và nhà phát triển, bộ dữ liệu này đã được phát hành theo giấy phép Creative Commons BY 4.0. Điều này cho phép nó dễ dàng được kết hợp trong các hướng dẫn và các tập lệnh khác, nơi nó có thể được tải xuống và sử dụng mà không cần bất kỳ sự can thiệp nào của người dùng (ví dụ: đăng ký trên một trang web hoặc gửi email cho quản trị viên để xin phép)

2.2.2 Đặc điểm

Bộ dữ liệu Speech Commands Dataset gồm các đoạn âm thanh ngắn (khoảng 1 giây) của 35 từ khóa khác nhau, chẳng hạn như "yes", "no", "up", "down", "left", "right", "on", "off", "stop", "go"... của hàng nghìn

người khác nhau, do các thành viên của công chúng đóng góp.

Bộ dữ liệu MUSAN dataset bao gồm khoảng 109 giờ dữ liệu âm thanh, chia thành ba loại chính: nhạc, giọng nói, và nhiễu. Nhạc gồm các thể loại như cổ điển, pop, và nhiều thể loại khác. Giọng nói gồm các bản ghi âm tiếng nói trong 12 ngôn ngữ, cả hội thoại và phát thanh. Nhiễu bao gồm các bản ghi âm nhiễu từ môi trường như tiếng nói chuyện, tiếng ồn từ các thiết bị gia dụng...

Bộ dữ liệu Speech Command-Musan dataset được tạo ra bằng cách thêm nhiễu từ tập dữ liệu MUSAN vào các đoạn âm thanh từ tập Speech Command. Speech Command-Musan dataset gồm 77034 mẫu âm thanh được chia thành 7 phần với mức độ SNR khác nhau -12.5,-10,0,10,20,30 và 40 decibel (dB). Mỗi mẫu được gắn nhãn với từ khóa tương ứng (35 nhãn tương ứng với 35 từ), các mẫu ở định dạng WAV với tần số 16kHZ.

2.3 Đặc trưng của tín hiệu âm thanh

Con người có thể nghe thấy âm thanh có tần số từ 20 Hz đến 20 kHz dựa trên áp lực tác dụng lên màng nhĩ. Một nhân tố nữa của âm thanh là cường độ, được đo bằng dB (decibel). Âm thanh thấp nhất mà con người có thể phát hiện được là 0 dB và hầu hết sẽ cảm thấy khó chịu ở 120 dB. Tín hiệu âm thanh thay đổi theo thời gian. Chúng có thể được biểu diễn trong miền thời gian hoặc trong miền tần số. Mỗi cảm giác âm thanh của chúng ta có liên quan đến một hoặc nhiều thuộc tính phổ hoặc thời gian của âm thanh. Do đó, các đặc điểm của cả hai miền thời gian và miền tần số đều được yêu cầu chung để biểu diễn âm thanh. Trọng tâm của phân loại tín hiệu âm thanh là tìm kiếm được các đặc trưng riêng biệt để việc phân tách chúng thành các lớp khác nhau trở nên dễ dàng. Thông thường, có hai phạm trù rộng của tín hiệu âm thanh bao gồm những đặc trưng vật lý (physical features) và những đặc trưng cảm quan (perceptual features)

2.3.1 Đặc trưng vật lí

Đặc trưng vật lý là các đặc trưng cấp thấp, có thể được đo trực tiếp từ biểu diễn tần số hoặc thời gian của tín hiệu âm thanh. Chúng là vật lý vì chúng có thể được tính toán trực tiếp từ các giá trị biên độ hoặc các giá trị phổ khác của tín hiệu.Phần này sẽ đề cập đến các đặc trưng vật lý điển hình nhất bao gồm Zero Cross Rate, Short-Time Energy, Spectral Centroid, Spectral Roll-off, Spectral Flux, Fundamental Frequency, Mel-Frequency Cepstral Coefficient.

Zero Cross Rate (ZCR) là thước đo tần suất tín hiệu âm thanh chuyển từ dương sang âm hoặc ngược lại. ZCR cao tương ứng với tần số cao, ZCR thấp tương ứng với tần số thấp. ZCR là một tính năng quan trọng đối với nhiều phân loại như tách tín hiệu thành phân loại có âm thanh tần số cao và âm thanh tần số thấp hoặc không có âm thanh. ZCR là một tính năng phổ biến vì nó rất dễ đo bằng cách sữ dụng máy đo ZCR. ZCR được tính bằng công thức sau:

$$\text{ZCR} = \frac{1}{N-1}\sum_{n=1}^{N-1} \mathbf{1}\left(x[n] \cdot x[n-1] < 0\right)$$

Trong đó:

- N là số mẫu trong đoạn tín hiệu.
- $x[n]$ là giá trị của tín hiệu tại mẫu thứ n.

Short-Time Energy (STE) là một đặc trưng quan trọng trong xử lý tín hiệu âm thanh, được sử dụng để đo lường năng lượng của tín hiệu âm thanh trong các khoảng thời gian nhỏ (cửa sổ thời gian). Được sử dụng để phát hiện các sự kiện âm thanh ví dụ như tiếng nói, tiếng nổ hoặc tiếng vang, nơi có sự thay đổi đột ngột về STE. Công thức của STE được tính bằng công thức sau:

$$\text{STE} = \sum_{n=0}^{N-1} x[n]^2$$

Trong đó:

- $x[n]$ là giá trị của tín hiệu tại mẫu thứ n.
- N là số mẫu trong khoảng thời gian ngắn.

Đặc trưng Spectral Centroid được sử dụng để mô tả trọng tâm tần số của âm thanh thường được sử dụng để phân loại âm thanh hoặc phát hiện các đặc điểm âm thanh đặc biệt. Spectral Centroid được tính toán bằng

cách tính trung bình có trọng số của tất cả các tần số trong phổ của mẫu âm thanh, trong đó trọng số được xác định bởi mức độ năng lượng ở mỗi tần số. Spectral Centroid được tính bằng công thức:

$$SC = \frac{\sum_{f=0}^{N-1} f \cdot X(f)}{\sum_{f=0}^{N-1} X(f)}$$

Trong đó:

- f là tần số.
- $X(f)$ là năng lượng của tín hiệu âm thanh tại tần số f.
- N là số lượng tần số trong phổ.

Đặc trưng Spectral Roll-off để tìm ra ngưỡng tần số chiếm hầu hết năng lượng của âm thanh. Nó cung cấp thông tin về phân phối của năng lượng âm thanh. Spectral Roll-off được tính bằng công thức:

$$\text{Spectral Roll-off} = \min \left\{ f : \sum_{f=0}^{f} |X(f)| \geq \alpha \sum_{f=0}^{N-1} |X(f)| \right\}$$

- α là phần trăm năng lượng phổ mà chúng ta quan tâm, thường được chọn trong khoảng từ 0.85 đến 0.95.
- $X(f)$ là biểu diễn phổ của tín hiệu âm thanh tại tần số f.
- N là số lượng tần số trong phổ.

Đặc trưng Spectral Flux được sử dụng để mô tả sự biến đổi của năng lượng tần số của âm thanh qua thời gian. Spectral Flux đo lường sự khác biệt hoặc sự thay đổi năng lượng giữa các khung thời gian liên tiếp trong một tín hiệu âm thanh. Giá trị cao Spectral Flux cho thấy sự thay đổi đột ngột về độ lớn năng lượng. Spectral Flux được tính bằng công thức:

$$SF(t) = \sum_{t=1}^{N-1} |X(t) - X(t-1)|^2$$

Trong đó:

- $X(t)$ là năng lượng tại khung thời gian t.
- N là số lượng khung thời gian.

Đặc trưng Fundamental Frequency (F0) là tần số cơ bản của một âm thanh, tức là tần số của hành trình dao động cơ bản của âm thanh. Trong ngữ cảnh của giọng nói, F0 thường tương ứng với tần số của âm thanh cơ bản của dải tiếng nói, hay còn gọi là "âm sắc" của giọng nói. Trong âm nhạc, F0 thường tương ứng với tần số của nốt nhạc cơ bản.

Đặc trưng Mel-Frequency Cepstral Coefficients (MFCCs) là một phương pháp phổ biến trong xử lý tín hiệu âm thanh, đặc biệt là trong nhận dạng giọng nói và xử lý ngôn ngữ tự nhiên. MFCCs thường được sử dụng để biểu diễn đặc tính tần số và thời gian của tín hiệu âm thanh theo cách mà con người nghe thấy và xử lý âm thanh. Nó thường được thực hiện theo quy trình như sau:

- Chia tính hiệu thành các khung (frame) thường khoảng vài mili giây đến vài chục mili giây. Mỗi khung sau đó được xử lý độc lập.
- Sử dụng phép biến đổi Fourier (FFT) hoặc các phương pháp tương tự để tính toán phổ năng lượng của mỗi khung âm thanh.
- Sử dụng bộ lọc thông qua cấu trúc bộ lọc Mel để biến đổi phổ năng lượng từ thang tần số tuyến tính sang thang đo Mel, giúp tương thích hơn với cách con người nghe và xử lý âm thanh.
- Logarithm hóa phổ năng lượng để giảm độ nhạy cảm về biến thiên tần số và giảm độ lớn của các giá trị.
- Sử dụng biến đổi cosin rời rạc (Discrete Cosine Transform - DCT) để tính toán các hệ số từ phổ Mel đã được logarithm hóa.

MFCCs cung cấp một biểu diễn hiệu quả của tín hiệu âm thanh, giúp giảm chiều dài của dữ liệu và tăng cường tính chất phân loại và nhận dạng của tín hiệu âm thanh trong nhiều ứng dụng khác nhau.

2.3.2 Đặc trưng cảm quan

Con người nhận biết âm thanh dựa trên các thuộc tính tri giác của âm thanh. Mô hình tâm lý học cũng đã được đề xuất để đo lường các đặc điểm cảm nhận của âm thanh để phân loại. Hầu hết các nhận thức về âm thanh được đo bằng độ to, cao độ và âm sắc. Âm sắc chủ yếu được sử dụng để phân biệt giữa các âm thanh có cùng độ lớn và cao độ.

Loudness (độ to) cho biết cảm giác về cường độ của tín hiệu. Độ to có thể liên quan đến cường độ như sau:

$$L = k.I^{\alpha}$$

Giá trị được chứng minh bằng 0.23 trong trường hợp có nhiễu. Độ to cũng phụ thuộc vào tần số. Xét rằng, một số mô hình được xác định ở đó để tính toán độ to.

Pitch (cao độ) là thuộc tính cảm nhận của âm thanh cho phép sắp xếp thứ tự của chúng trên thang đo liên quan đến tần số, hoặc phổ biến hơn, cao độ là chất lượng giúp bạn có thể đánh giá âm thanh là "cao hơn" và "thấp hơn" theo nghĩa liên quan đến âm nhạc. Pitch được định nghĩa là tần số cơ bản của nguồn kích thích.

3 Mô hình

3.1 Mô hình tổng quan

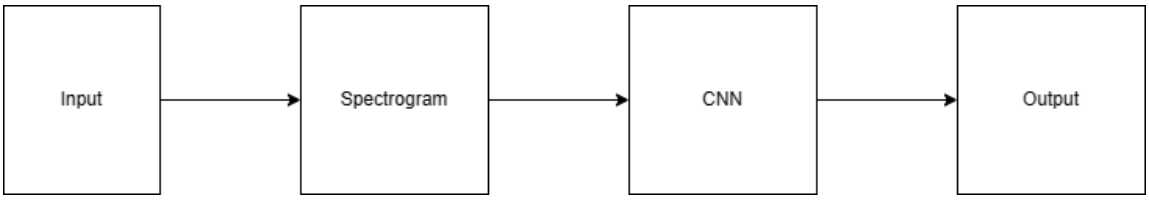

Figure 5: Overview

Mô hình ban đầu nhận vào 1 file âm thanh định dạng WAV, sau đó chuyển âm thanh sang spectrogram để đưa vào mô hình mạng CNN đã được huấn luyện để dự đoán ra được từ được nói trong file âm thanh.

3.2 Mô hình Convolutional Neural Network

Mô hình mạng neural network được sử dụng trong thiết kế này là mạng CNN được thể hiện như hình 6 bên dưới:

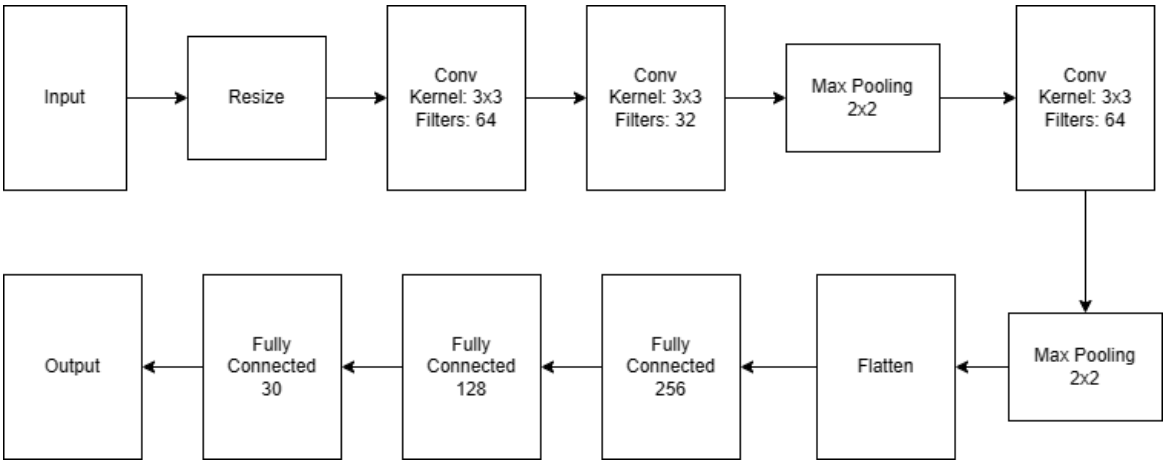

Figure 6: CNN

Đầu vào spectrogram của file âm thanh có dạng hình ảnh 2D, kích thước hình khá lớn do đó để tiết kiệm tài nguyên cần biến đổi về dạng hình ảnh nhỏ (64x64). Tiếp theo, sử dụng preprocessing để chuẩn hóa dữ liệu.Sau đó đi qua các lớp tích chập.

- Lớp tích chập đầu tiên có số filter là 64, kernel size là 3x3 và sử dụng hàm kích hoạt ReLU. Kích thước đầu ra là 62x62x64.
- Lớp tích chập thứ 2 có số filter là 32, kernel size là 3x3 và hàm kích hoạt ReLU. Kích thước đầu ra là 60x60x32.
- Lớp tiếp theo là MaxPooling 2x2 nhằm giảm chiều dữ liệu đi một nữa. Kích thước đầu ra là 30x30x32.
- Lớp tích chập thứ 3 có số filter là 64, kernel size là 3x3 và hàm kích hoạt ReLU. Kích thước đầu ra là 28x28x64.
- Lớp tiếp theo là MaxPooling 2x2 nhằm giảm chiều dữ liệu đi một nữa. Kích thước đầu ra là 14x14x64.
- Tiếp theo, mô hình có lớp làm phẳng để chuyển ngõ ra dạng 2D của các mạng tích chập sang dạng vector để làm ngõ vào cho các lớp Fully Connected. Đầu ra là 18,496 entry.
- Lớp fully connected đầu tiên có số lượng Neural là 256 neural và sử dụng hàm kích hoạt ReLU.
- Lớp fully connected tiếp theo có số lượng neural là 128 neural và sử dụng hàm kích hoạt ReLU.
- Lớp cuối cùng cũng là lớp ngõ ra, có số lượng Neural đúng bằng với số ngõ ra là 35 neural và hàm kích hoạt được sử dụng là Softmax nhằm chuyền đổi các ngõ ra thành dạng phân bố xác suất.

Giữa những lớp Fully Connected sẽ bổ sung DropOut nhằm hạn chế được tình trạng overfitting

4 Hiện thực

4.1 chia tập dữ liệu

Dữ liệu có tổng số lượng mẫu là 77034 được chia thành 3 tập.

- train_files: chứa các mẫu được dùng để cập nhật trọng số cho mô hình
- val_files: chứa các mẫu để validation trong quá trình huấn luyện, nhằm giảm hiện tượng overfitting cho mô hình
- tes_files: chứa các mẫu được dùng cho quá trình kiểm thử mô hình

Với tỉ lệ là train_files 80%, val_files 10% và test_files 10%.

4.2 Xử lý dữ liệu

Tập dữ liệu Speech Command-Musan Dataset gồm các file âm thanh (*.wav) chứa trong các thư mục con có tên là nhãn của file âm thanh đó, tuy nhiên, đầu vào của mạng CNN có dạng là 1 hình ảnh, vì thế cần xử lý tập dữ liệu để chuyển đổi âm thanh thành dạng hình ảnh spectrogram.

- Hàm giải mã file âm thanh (*.wav): Hàm này có tác dụng giải mã file wav, giá trị đầu vào của hàm là biến nhị phân được đọc từ file âm thanh. Kết quả trả về của hàm là 1 vector biểu diễn dạng sóng của đoạn âm thanh đó theo thời gian.

```
def decode_audio(audio_binary)
```

- Hàm gán nhãn cho mẫu: Nhãn của file âm thanh là tên thư mục chứa file đó.

```
def get_label(file_path)
```

- Hàm thực hiện giải mã âm thanh và gán nhãn cho mẫu: Chức năng của hàm là thực hiện giải mã các file âm thanh (*.wav) thành dạng sóng theo thời gian đồng thời gán nhãn cho mẫu là dạng sóng của đoạn âm thanh đó.

```
def get_waveform_and_label(file_path)
```

- Hàm chuyển sang spectrogram: Chức năng của hàm biến đổi mẫu từ dạng waveform sang spectrogram. Kết quả trả về của hàm là 1 hình ảnh có kích thước m x n bằng phương thức tf.signal.stft của tensorflow. Với frame_length (chiều dài frame) là 255 và frame_step (bước nhảy frame) là 128 của hàm tf.signal.stft được lựa chọn để tạo thành hình ảnh có dạng gần hình vuông nhất để thuận tiện cho việc biến đổi kích thước trong quá trình huấn luyện.tất cả các mẫu trong tập dữ liệu sẽ cho kết quả đầu ra với kích thước là 124x129.

```
def get_spectrogram(waveform)
```

- Hàm chuyển sang spectrogram cho mẫu và chuyển nhãn của mẫu từ dạng chuỗi sang dạng số: Chức năng của hàm là biến đổi mẫu từ dạng waveform sang dạng spectrogram và chuyển nhãn của mẫu từ dạng chuỗi sang dạng số để phù hợp với đầu ra của mạng neural network.

```
def get_spectrogram_and_label_id(audio, label)
```

- Hàm map đường dẫn của file âm thanh thành các mẫu có các thuộc tính spectrogram và label: Chức năng của hàm là chuyển đổi các đường dẫn của file thành dataset gồm spectrogram và nhãn để làm đầu vào và ngõ ra mong đợi cho mạng neural.

```
def preprocess_dataset(files)
```

4.3 Xây dựng mô hình CNN

Mô hình được xây dựng bằng class Keras trong thư viện Tensorflow với cấu trúc mạng Neural Network như đã nói.

Layer (type)	Output Shape	Param #
resizing (Resizing)	(None, 64, 64, 1)	0
normalization (Normalization)	(None, 64, 64, 1)	3
conv2d (Conv2D)	(None, 62, 62, 64)	640
conv2d_1 (Conv2D)	(None, 60, 60, 32)	18,464
max_pooling2d (MaxPooling2D)	(None, 30, 30, 32)	0
conv2d_2 (Conv2D)	(None, 28, 28, 64)	18,496
max_pooling2d_1 (MaxPooling2D)	(None, 14, 14, 64)	0
dropout (Dropout)	(None, 14, 14, 64)	0
flatten (Flatten)	(None, 12544)	0
dense (Dense)	(None, 256)	3,211,520
dropout_1 (Dropout)	(None, 256)	0
dense_1 (Dense)	(None, 128)	32,896
dropout_2 (Dropout)	(None, 128)	0
dense_2 (Dense)	(None, 35)	4,515

Figure 7: CNN Layer

Sau khi tổng hợp xong mô hình, ta tiến hành compile cho mô hình với thuật toán tối ưu hóa 'Adam' và hàm mất mát sử dụng là 'Crossentropy', thông số đánh giá mô hình là 'accuracy'.

5 Kết quả

Sau khi huấn luyện mô hình, ta tiến hành đánh giá mô hình dựa vào tập dữ liệu test dataset.

- Đánh giá dựa trên độ chính xác của dự đoán trên tập test: Để đưa ra độ chính xác của mô hình trên tập test, ta dùng phương thức model.predict() và sau đó thống kê số lượng mẫu dự đoán đúng. Kết quả trên tập test có kết quả độ chính xác là 84%

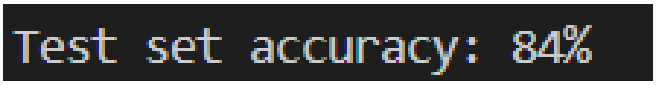

Figure 8: Accuracy

- Đánh giá qua confusion matrix: Confusion matrix cho ta cái nhìn trực quan hơn đối với từng nhãn riêng biệt, từ đó có thể nhận xét mô hình cho kết quả như thế nào đối với từng nhãn.Kết quả của đoạn chương trình trên cho ra kết quả là một ma trận confusion matrix như hình bên dưới. Hình ảnh confusion matrix cho thấy mô hình dự đoán gần như chính xác toàn bộ ở các nhãn. Các mẫu có nhãn là 'tree' dễ bị mô hình nhầm lẫn thành 'three', và tương tự nhãn 'no' cũng bị mô hình nhầm lẫn thành 'go' khá nhiều. Điều này có thể lý giải là vì âm điệu của 'no' và 'go' cũng như 'tree' và 'three' gần giống nhau.

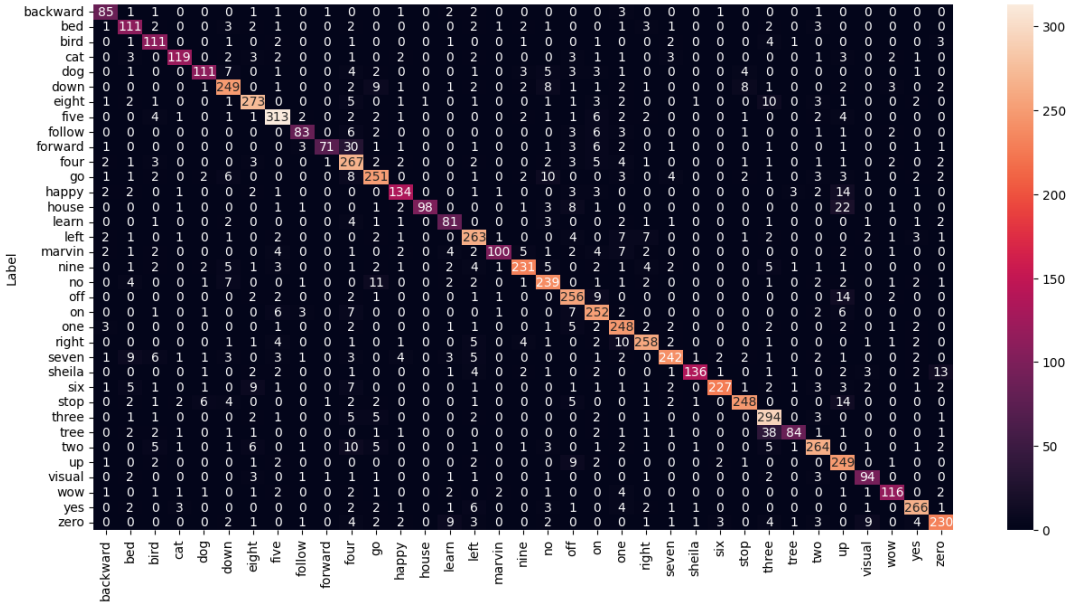

Figure 9: Prediction

6 Kết luận

Kết quả sau khi hoàn thành đã đáp ứng được mục tiêu đề ra là xây dựng một ứng dụng nhận dạng âm thanh cơ bản bằng mô hình mạng neural tích chập CNN . Sau quá trình hoàn thiện sản phẩm nghiên cứu học thuật trên, tác giả đã nhận thấy rằng ứng dụng được phát triển có các ưu điểm cũng như còn tồn động một vài hạn chế cần khắc phục cũng như nâng cấp. Mô hình mạng neural tích chập được xây dựng đã nhận dạng chính xác các âm thanh trong tập dữ liệu Speech Command-Musan dataset với độ chính xác hơn 84%, quá trình huấn luyện mô hình được thực hiện với thời gian khoảng 2.5 giờ với số lượng mẫu huấn luyện là 84% tổng số mẫu trong tập dữ liệu. Bên cạnh đó cũng còn các hạn chế của mô hình như chỉ nhận dạng được âm thanh có thời lượng ngắn (<1s), số lượng từ có khả năng nhận dạng chỉ là 35 từ, là những thứ cần được cải thiện thêm.

Tài liêu tham khảo

[1] Timea Bezdan, Nebojsa Bacanin, *Convolutional Neural Network Layers and Architectures*, Singidunum University, 2019.

[2] Saad ALBAWI, Tareq Abed MOHAMMED, *Understanding of a Convolutional Neural Network*, Istanbul Kemerburgaz University, Istanbul, Turkey, 2017.

[3] Nguyễn Thế Xuân Long, Mai Lam, Dương Quốc Hoàng Tú, *KỸ THUẬT NHẬN DẠNG GIỌNG NÓI SỬ DỤNG MÔ HÌNH MARKOV ẨN* , Trường Cao đẳng Công nghệ thông tin – Đại học Đà Nẵng.

[4] Karpagavalli S and Chandra E, *A Review on Automatic Speech Recognition Architecture and Approaches*, India.

[5] Ngô Minh Dũng, *NGHIÊN CỨU KỸ THUẬT NHẬN DẠNG NGƯỜI NÓI DỰA TRÊN TỪ KHÓA TIẾNG VIỆT* , Hà Nội.

[6] Jing Pan, Joshua Shapiro, Jeremy Wohlwend, Kyu J. Han, Tao Lei and Tao Ma, *ASAPP-ASR: Multistream CNN and Self-Attentive SRU for SOTA Speech Recognition*, ASAPP Inc.

[7] Andrew L. Maas, Quoc V. Le, Tyler M. O'Neil, Oriol Vinyals, Patrick Nguyen, Andrew Y. Ng, *Recurrent Neural Networks for Noise Reduction in Robust ASR*, USA.

[8] Amita C. Patil, Hyung Lee, *Speech to Text Translation Using Google SpeechCommands*, Stanford University, California.

[9] Madiha Jalil, Faran Awais Butt, Ahmed Malik, *Short-Time Energy, Magnitude, Zero Crossing Rate and Autocorrelation Measurement for Discriminating Voiced and Unvoiced segments of Speech Signals*, University of Management and Technology, Lahore, Pakistan, 2013.

[10] David Gerhard, *Audio Signal Classification: An Overview*, School of Computing Science Simon Fraser University Burnaby, BC, 2002

[11] Mittal C. Darji, *Audio Signal Processing: A Review of Audio Signal Classification Features*, Information Technology Department, G H Patel College of Engineering & Technology, India, 2017.

[12] Nguyễn Thị Mỹ Thanh, Phan Xuân Dũng, Nguyễn Ngọc Hay, Lê Ngọc Bích, Đào Xuân Quy, *ĐÁNH GIÁ CÁC HỆ THỐNG NHẬN DẠNG GIỌNG NÓI TIẾNG VIỆT (VAIS, VIETTEL, ZALO, FPT VÀ GOOGLE) TRONG BẢN TIN*, Trường Đại học Quốc tế Miền Đông, Việt Nam.

[13] Duong Trinh Anh, Sam Dang Van, Tuan Do Van, Vi Ngo Van, *Vietnamese Automatic Speech Recognition with Transformer*, VCCorp.

[14] Chadawan Ittichaichareon, Siwat Suksri and Thaweesak Yingthawornsuk, *Speech Recognition using MFCC*, Thailand.

[15] Ebru Arısoy, Tara N. Sainath, Brian Kingsbury, Bhuvana Ramabhadran, *Deep Neural Network Language Models*, USA.