ĐỀ TÀI:

PHÂN LOẠI VÙNG MIỀN VÀ GIỚI TÍNH DỰA TRÊN GIỌNG NÓI

Giảng viên hướng dẫn:
Cao Trần Trưởng
Nguyễn Quang Uy

Thành viên:
1. Lưu Minh Quân
2. Trần Hoàng Vũ
3. Trần Tấn Thành

Hanoi, 1-2021

MỤC LỤC

CHƯƠNG 1

GIỚI THIỆU

1.1. Giới thiệu vấn đề

Trong các hệ thống nhận diện giọng nói, việc phân loại giới tính và vùng miền dựa trên tín hiệu âm thanh rất cần thiết cho các hệ thống thông minh như chatbot hội thoại, hệ thống khuyến nghị, nhà thông minh.

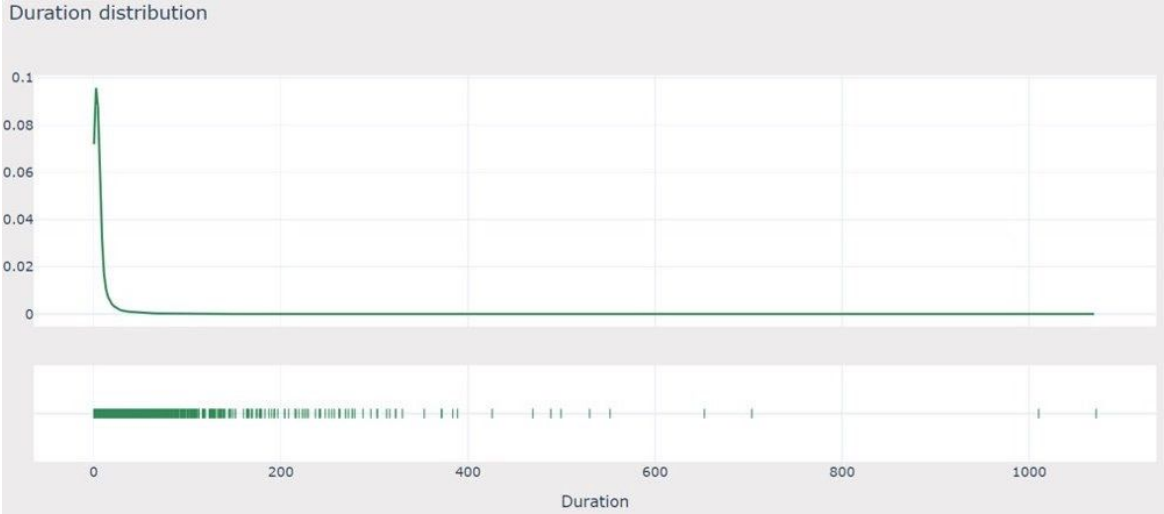
1.2. Mục tiêu

Mục tiêu chính của đề tài là nhận diện giới tính cũng như vùng miền của giọng nói người Việt Nam áp dụng các kĩ thuật các phương pháp tiền xử lý tín hiệu âm thanh và các phương pháp học sâu. Cụ thể, đề tài sẽ cố gắng:

- Xây dựng các model phân loại dựa trên bộ data gồm các file âm thanh giọng nói người Việt
- So sánh một số phương pháp học sâu để chọn ra mô hình phù hợp nhất với bộ dữ liệu
- Xây dựng trang web để người dùng có thể sử dụng mô hình được đào tạo nhằm phân loại giọng nói

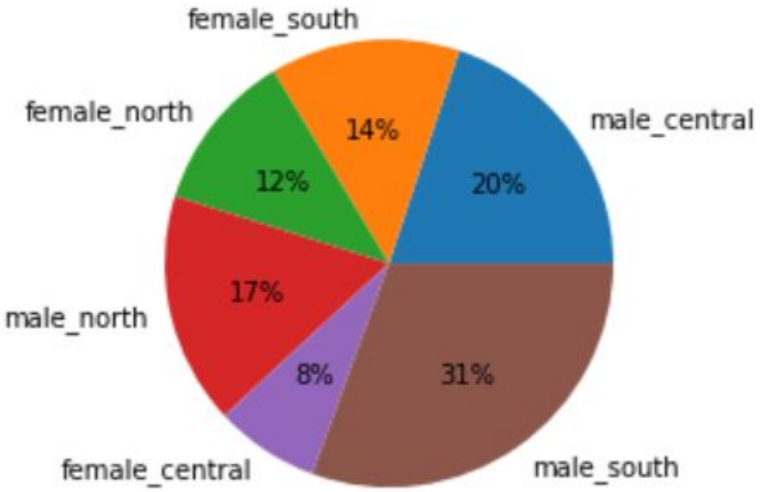
1.3. Dataset

Bộ dữ liệu được sử dụng trong đề tài gồm hơn 25,000 file âm thanh giọng nói người Việt, trong đó có khoảng 20,000 trên cho tập Training và Validation, trong khi có hơn 5000 file cho tập Testing. Mỗi file âm thanh có độ dài trung bình khoảng ba giây. Nhãn của bộ dataset được chia thành hai lớp bao gồm lớp giới tính: Nam/ Nữ và lớp vùng miền: Bắc/ Trung/ Nam.



Hình 1. Phân bố của độ dài các file âm thanh

Đa phần các file âm thanh có độ dài ngắn chỉ khoảng vài giây; tuy nhiên, có những file kéo dài đến hơn 10 phút.



Hình 2. Phân bố của các lớp trong tập Training

Ta thấy giọng của giới tính Nam cùng với miền Nam chiếm tỉ lệ cao nhất, lên tới 31%, trong khi đó giọng nữ miền Trung chiếm tỉ lệ 8%

CHƯƠNG 2

CƠ SỞ LÝ THUYẾT

2.1. Giới thiệu

Nhằm xây dựng mô hình cho việc phân loại âm thanh, bước đầu tiên cần phải xác định những biểu diễn nào cho bộ dữ liệu này. Chúng ta có thể xây dựng những mô hình dựa trên những đặc trưng cấp thấp như đặc trưng miền thời gian: Root-mean-square energy, Zero-crossing rate, Amplitude envelope; đặc trưng miền tần số: Spectral Frequency, Spectral Envelope với bộ phân lớp học máy truyền thống như ID3; hoặc những đặc trưng cấp cao mang cả thông tin về thời gian và tần số như Spectrogram, Mel-spectrogram, MFCCs với các mạng tích chập; hoặc sử dụng mô hình end-to-end để học những đặc trưng từ waveform.

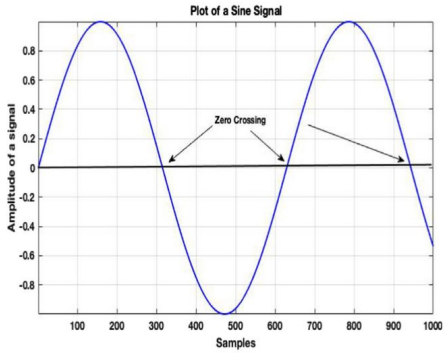
2.2. Phương pháp sử dụng học máy

2.2.1. Trích xuất đặc trưng

Xây dựng những đặc trưng cấp thấp trong miền thời gian và trong miền tần số.

2.2.1.1. Đặc trưng trong miền thời gian

a. Root-mean-square: Hay còn gọi là Volume được định nghĩa là độ lớn của tín hiệu trong một frame
b. Zero-crossing-rate: ZCR được định nghĩa là số lần giao động của tín hiệu trong một frame



Hình 2. Zero - crossing trong tín hiệu

$$Z(i) = \frac{1}{2N} \sum_{n-1}^{N} [sgn[x_i(n)] - sgn[x_i(n-1)]]$$

c. Amplitude envelope - ADSR: Là phương pháp mang lại thông tin về Amplitude của tín hiệu

2.2.1.2. Đặc trưng trong miền tần số

Spectral: Sử dụng phép biến đổi Fourier để chuyển miền thời gian sang miền tần, DFT được thực hiện trong tín hiệu rời rạc.
Một số spectral rất hữu ít cho bài toán phân loại âm thanh như: Spectral Centroid, Spectral Center, Spectral roll - off , Spectral Spread, Spectral Skewness, Spectral Kurtosis, Spectral Entropy.
Một số đặc trưng khác: MFCC, Mel - Spectrogram cũng rất hữu ích cho bài toán phân loại âm thanh.

2.2.2. Bộ phân loại

Xây dựng bộ những phân loại Decision Tree (ID3) và Gradient Boosting để tạo ra bộ phân loại mạnh. Thuật toán Gradient Boosting bắt đầu bằng cách huấn luyện một Decision Tree trong tập huấn luyện với mỗi mẫu quan sát có trọng số như nhau. Sau khi đánh giá trong lần đầu tiên với Decision Tree. Ta tăng trọng số đối với những quan sát khó phân loại và giảm trọng số với những quan sát dễ phân loại. Đào tạo bộ phân loại Decision Tree mới trên tập dữ liệu này. Các Decision Tree sau sẽ giúp phân loại những quan sát mà Decision Tree trước khó phân loại. Và bộ dự đoán cuối cùng là trung bình cộng của các Decision Tree.

2.3. Phương pháp sử dụng mạng tích chập

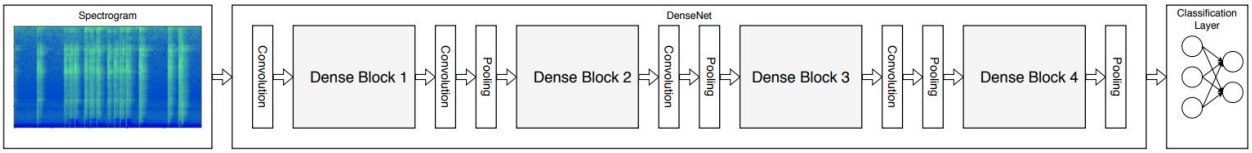
2.3.1. Trích xuất đặc trưng

Nhằm xây dựng mô hình cho việc phân loại âm thanh với mạng tích chập, chúng ta có thể xây dựng mô hình dựa trên âm thanh thô dạng waveform (Lee and Park) sử dụng những mạng CNN (VGG16-1D, ResNet-1D) (Soloyyev and Vakhrushev) để học những đặc trưng , hoặc biểu diễn âm thanh 2D như Spectrograms, Mel-spectrograms. Mel-spectrograms đã trở nên ngày càng phổ biến bởi chúng hiệu quả với các mạng tích chập (CNN) (Choi and Fazekas), (Choi and Fazekas). Tuy nhiên, CNN được xây dựng dựa trên hình ảnh và 2D Spectrograms khác biệt so với hình ảnh thông thường vì ảnh thông thường chứa cả thông tin về không gian và thời gian. Còn, Spectrograms chứa cả chiều thời gian và tần số. Do đó, những kiến trúc mạng CNN có thể được thay đổi để phù hợp hơn với spectrograms. Năm 2014, (Gwardys and Grzywczak) chứng minh rằng ta có thể xem những Mel-spectrograms như hình ảnh và sử dụng các mạng CNN như AlexNet được pretrained trên tập ImageNet cho việc phân loại âm thanh. Trong phương pháp này, ta chứng minh bằng việc sử dụng các mạng CNN như DenseNet (Huang and Liu #) và một đặc trưng âm nhanh như Mel-Spectrograms, ta có thể đạt kết quả tốt trên bộ dữ liệu được sử dụng trong đề tài.

2.2.2. Bộ phân loại

2.2.2.1. Dựa trên tích chập 2D

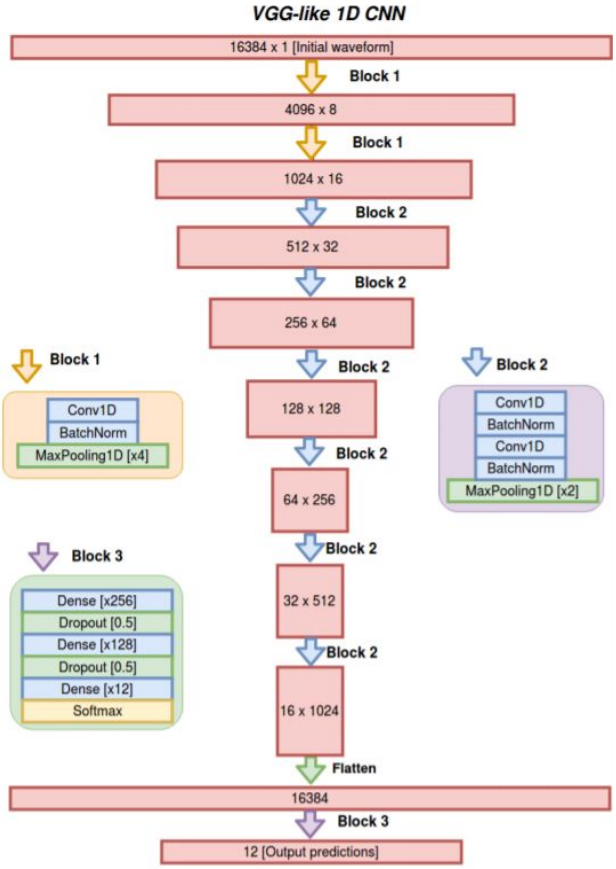
DenseNet được sử dụng để huấn luyện cho đặc trưng Mel-Spectrogram được trích ra từ âm thanh. DenseNet kết nối mỗi lớp với lớp tiếp theo theo mô hình feed-forward. Với mỗi lớp, ma trận đặc trưng của tất cả lớp phía trước được sử dụng như đầu vào của những lớp tiếp theo



Hình 3. Kiến trúc DenseNet: Mỗi lớp Dense Block chứa số các lớp tích chập nhất định mà đầu vào của nó chứa các đặc trưng từ các lớp trước

2.2.2.2. Dựa trên tích chập 1D

Mạng tích chập 1D có khả năng nhận diện những đặc trưng địa phương của tín hiệu âm thanh. Ta xem xét loại tích chập 1D phổ biến cho phân loại âm thanh: VGG-16 có đầu vào là tín hiệu âm thanh waveform



Hình 4. Biểu diễn của VGG16-1D

CHƯƠNG 3

PHƯƠNG PHÁP THỰC HIỆN VÀ KẾT QUẢ

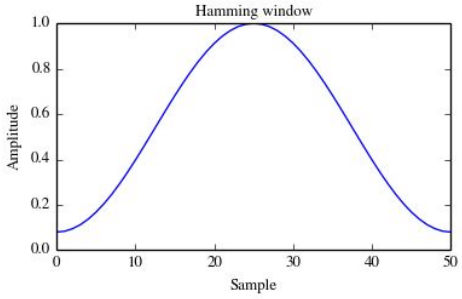
3.1. Tiền xử lý

Vì các file âm thanh đều ở ba định dạng âm thanh phổ biến như wav, mp3, amr, và ở các mức lấy tần số lấy mẫu khác nhau.Dựa trên thực nghiệm, ta chuyển đổi các file âm thanh này thành wav với kênh âm thanh là mono cùng tần số lấy mẫu là 16000 Hz và Pulse-code modulation (PCM)

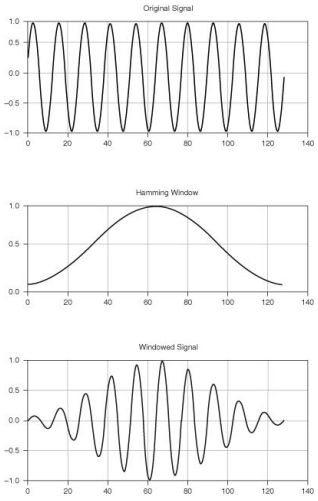
3.1.1. Discrete Fourier Transform

3.1.1.1. Một vài định nghĩa trong xử lý số:

- Frames: Được định nghĩa là một đoạn tín hiệu nhỏ trong tín hiệu chứa một số mẫu của tín hiệu rời rạc tương ứng với kích thước frame.
- Windowing: Áp dụng một hàm window cho mỗi frame. Mục đích loại bỏ những samples ở đầu và cuối frame để tránh spectral leakage. Có một số window được sử dụng như Hann window, Blackman window, Hamming window



Hình 5. Hamming window



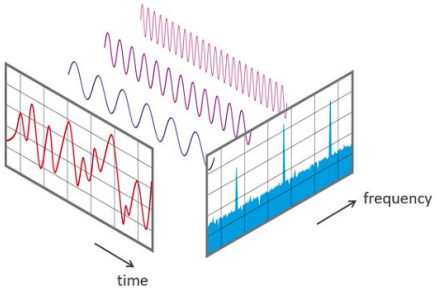
Hình 6. Sử dụng Hamming window trong một frame

Hop length: Việc sử dụng các window vào tín hiệu thì đã loại bỏ đi thông tin tín hiệu ở đầu và cuối mỗi frame điều này sẽ không tốt cho việc biến đổi fourier. Giải quyết vấn để này bằng cách chồng các frame với hop length.

3.1.1.2. Discrete Fourier Transform

Là một phép biến đổi fourier cho tín hiệu rời rạc. Giúp chuyển tín hiệu trong miền thời gian sang miền tần.

$$X(k) = \sum_{n=0}^{N-1} x(n) e^{-j2\pi kn/N}$$

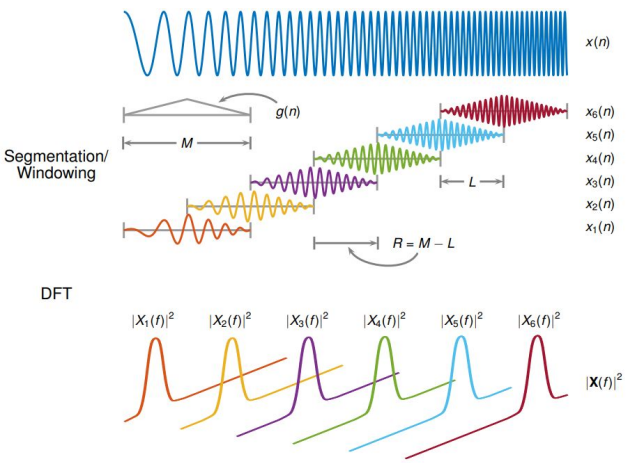


Hình 7. Biến đổi fourier transform miền thời gian sang miền tần số

3.1.1.3 Short - Time Fourier Transform

Là phép biến đổi DFT trên từng frame.

$$STFT = X(m, \omega) = \sum_{n=-\infty}^{\infty} x_n w_{n-m} e^{-i\omega t_n}$$



Hình 8. Biến đổi short time fourier transform

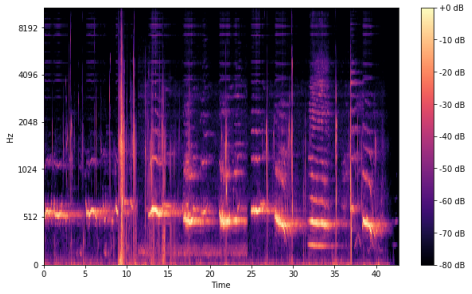
Việc sử dụng kích thước frame khác nhau sẽ có được những thông tin về tần số khác nhau (độ phân giải về miền tần số trong một frame)

$$No.\ of\ frequence\ bins\ = \frac{frame\ size}{2} + 1$$

3.1.2. Mel - Spectrogram

3.1.2.1. Spectrogram

Là biểu diễn phổ tần số của tín hiệu khi nó thay đổi theo gian. Sau khi STFT của tín hiệu ta xây dựng spectrogram của tín hiệu với trục x là miền thời gian, trục y là miền tần số. Với giá trị là năng lượng của phổ tần số.



Hình 9. Spectrogram của tín hiệu

3.1.2.2. Mel Spectrogram

Mel - scale: Là một phép biến đổi phi tuyến của thang tần số. Mel scale được xây dựng sao cho từ dải tần số 0 - 1000Hz là tuyến tính và từ 1000Hz là phí tuyến.

$$Mel(f) = 2595\ log(1 + \frac{f}{700})$$

3.1.3. Tính đánh đổi thông tin trong miền thời và miền tần số.

Như đã đề cập ở 3.1.1.3 STFT: việc chọn những kích thước frame khác nhau sẽ có được thông tin về tần số khác nhau.

$$No.\ of\ frequence\ bins\ = \frac{frame\ size}{2} + 1$$

Nếu kích thước frame lớn, chúng ta có được nhiều sample trong một frame. Việc biến đổi STFT trên nhiều sample sẽ cho ta được nhiều thông tin về tần số (tương tự như một bộ lọc thông thấp) nhưng điều này cũng đánh đổi đi thông tin về thời gian khi không biết chính xác được những phổ tần số này xuất hiện trong thời gian. Ngược lại nếu kích thước frame nhỏ, sẽ có ít sample và điều này dẫn tới có ít thông tin về miền tần số nhưng sẽ có nhiều thông tin về miền thời gian.

3.2. Phương pháp sử dụng CNNs

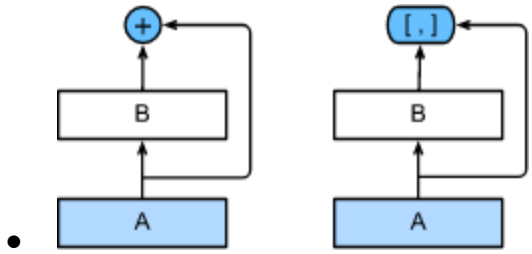
3.2.1. Hướng tiếp cận 2D - Convolution neural network

Các phương pháp dựa trên CNN đều yêu cầu đầu vào gồm ba kênh. Do đó, ta cần chuyển đặc trưng Mel-Spectrogram mà ta đã trích xuất thành đầu vào gồm ba kênh. Ba kênh đầu vào của Mel-Spectrogram được tính toán bằng cách sử dụng các windows size và các hop lengths khác

nhau bao gồm các cặp [25 ms, 10 ms], [50 ms, 25 ms], [100 ms, 50 ms] ở mỗi kênh. Bằng cách này, ta có thể đảm bảo rằng CNN sẽ có những đầu vào gồm nhiều cấp độ thông tin tần số và thời gian trên mỗi kênh.

3.2.2. DenseNet

ResNet đã làm thay đổi đáng kể quan điểm về cách tham số hóa các hàm số trong mạng nơ-ron sâu. Ở một mức độ nào đó, DenseNet có thể được coi là phiên bản mở rộng hợp lý của ResNet. Cụ thể ResNet tách một hàm số thành số hạng tuyến tính đơn giản và một số hạng phi tuyến phức tạp hơn. DenseNet giúp ta tách ra nhiều hơn hai số hạng

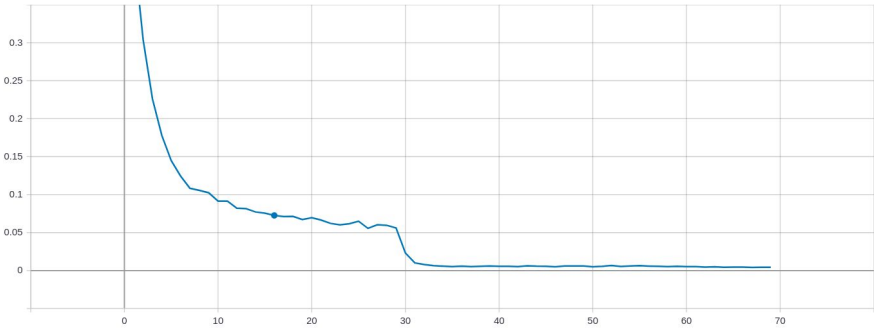


Hình 10. Sự khác biệt chính giữa ResNet (bên trái) và DenseNet (bên phải)

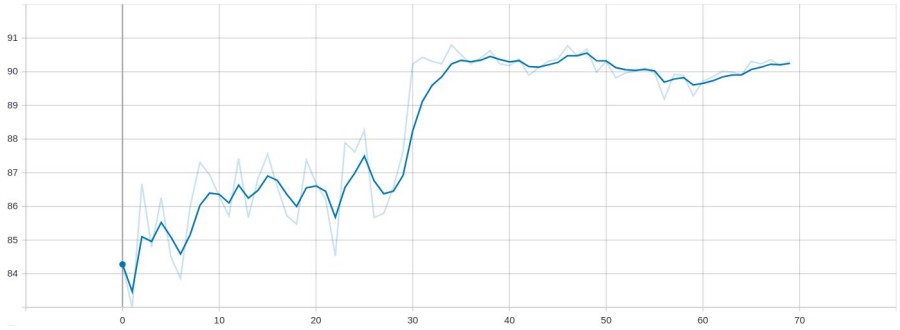
Cụ thể DenseNet nối đầu ra lại với nhau thay vì cộng lại như ở ResNet. Cuối cùng tất cả các tham số này này được kết hợp lại qua Dense Block để giảm lượng tham số một lần nữa.

3.2.3. Kết quả thực hiện

Qua thực nghiệm, ta tìm ra rằng tốc độ học là 1e-4, và weight decay là 1e-3 là những giá trị tốt nhất cho việc huấn luyện mô hình. Cùng với đó, ta sử dụng Adam optimizer với batch size là 32. Mô hình được huấn luyện trên GPU Nvidia RTX 2070. Mô hình Densenet được pretrained trên tập ImageNet được huấn luyện trên tập dữ liệu âm thanh được chuẩn hóa và trích xuất đặc trưng Mel-Spectrogram cho khoảng 70 epochs. Cùng với đó learning rate sẽ được giảm bội số của 10 mỗi 30



Hình 11. Loss trên tập Training với chiều x là số epoch và chiều y là loss trên tập Training



Hình 12. Accuracy trên tập Validation với chiều x là số epoch và chiều y là accuracy trên tập Validation

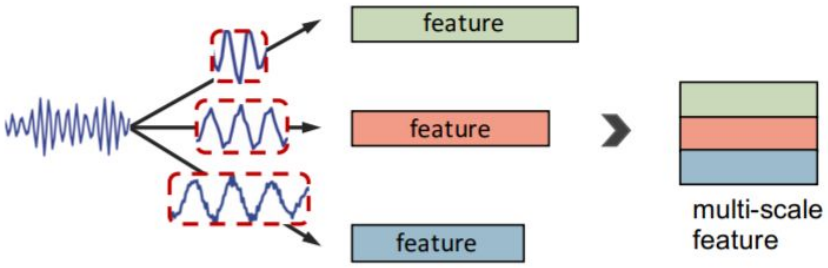
Ta thấy rằng model đã hội tụ tại khoảng epoch 35 và cũng đạt accuracy trên tập Validation cao nhất tại epoch này. Đánh giá model trên tập Testing, ta thu được độ chính xác là 77.6%

Model	Train accuracy	Valid accuracy	Test accuracy
DenseNet (pretrained)	99.89%	90.8%	78%

3.2.3. Hướng tiếp cận 1D - Convolution neural network

Xu hướng hiện tại tập trung vào hướng tiếp cận End - to - End. Học đặc trưng trực tiếp từ waveform thay vì học trên những đặc trưng cấp thấp hoặc cấp cao cần những kiến thức từ chuyên gia. Cách tiếp cận cho bài toán này sử dụng convolution 1D trên miền thời gian để trích xuất thông tin từ waveform cho bài toán phân loại.
Như đã đề cập ở 3.1.3, luôn luôn có tính đánh đổi thông tin trong miền thời gian và miền tần số cho việc chọn kích thước cửa sổ convolution. Cửa sổ với kích thước lớn cho ta độ phân giải về miền tần số tốt nhưng cũng giảm đi độ phân giải của miền thời gian và ngược lại.
Đề xuất sử dụng nhiều bộ lọc convolution với các kích thước khác nhau học được những đặc trưng về cả miền thời gian và tần số. Cân bằng việc đánh đổi thông tin của hai miền này.

3.2.4. Multi scale convolutional neural network



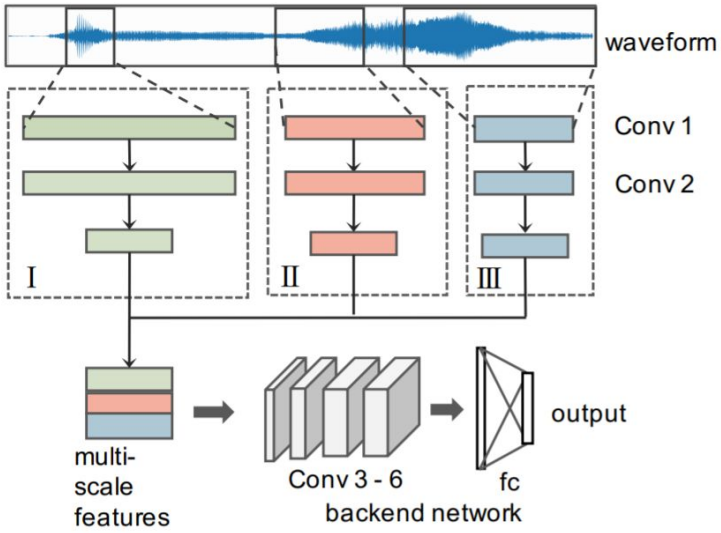
Hình 13. Multi scale feature

Đề xuất multi-scale convolution để học những đặc trưng khác nhau của cả miền thời gian và miền tần số, điều mà những phép convolution với kích thước bộ lọc cố định không đạt được.

$$x_j^{(s)}(n) = f\left(\sum_{i=0}^{N-1} w(i) h_j^{(s)}(n-i) + b_j^{(s)}\right)$$

Với f: hàm kích hoạt
N: là chiều dài của w(i) hay kích thước bộ lọc
Sử dụng 3 scale (s)

Với đầu vào waveform qua 3 bộ lọc với kích thước bộ lọc lần lượt [1 x 11], [1 x 51], [1 x 101] sau đó qua một vài lớp max - pooling và conv [1x11] để có được một vector [32, 320]. Kết hợp 3 kênh tương ứng với 3 scale convolution có được một vector đặc trưng của waveform [96, 320]. Sử dụng phép Conv2D và Fully - Connected để phân loại vector đặc trưng cho từng lớp.

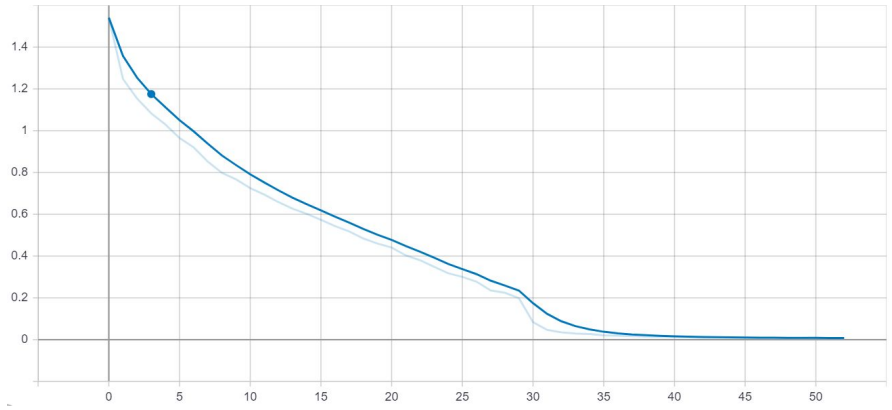


Hình 14. Cấu trúc mô hình Wavemsnet.

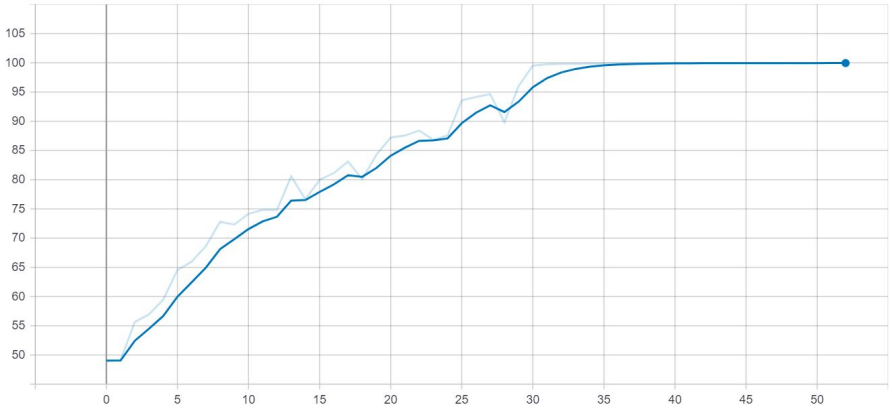
3.2.5. Kết quả thực hiện

Qua thực nghiệm, sử dụng Stochastic Gradient Descent with momentum để huấn luyện với learning rate: 0.001, momentum: 0.9 và weight decay: 0.0005 là giá trị tốt nhất cho huấn luyện model. Model được huấn luyện trên Tesla K80 GPU của Google Colab.

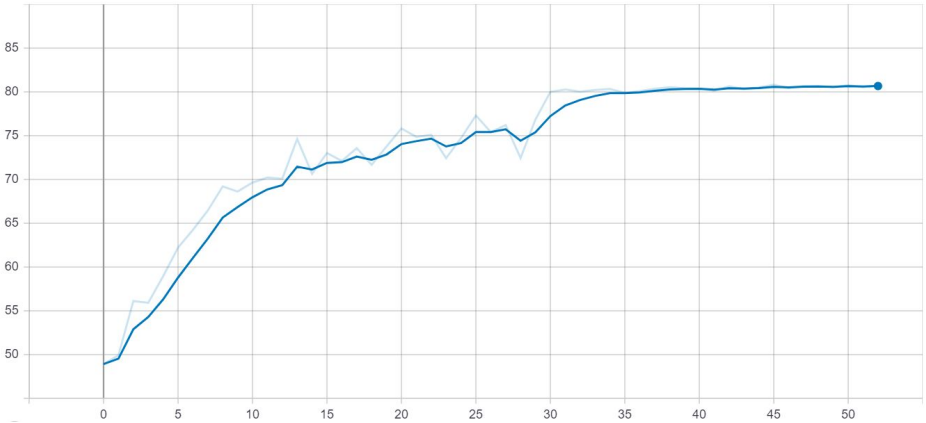
Model	Train accuracy	Valid accuracy	Test accuracy
WavemsNet	99.9%	80.8%	70%



Hình 15. Loss trên tập Training với chiều x là số epoch và chiều y là loss trên tập Training

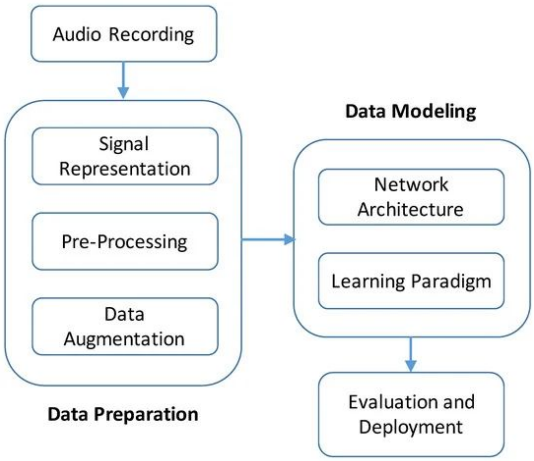


Hình 16. Accuracy trên tập Training với chiều x là số epoch và chiều y là accuracy trên tập Training



Hình 17. Accuracy trên tập Validation với chiều x là số epoch và chiều y là accuracy trên tập Validation

3.3. Pipeline bài toán



Hình 18. Pipeline bài toán phân loại âm thanh.

Sau khi âm thanh được ghi âm, tín hiệu sẽ được trích xuất, tiền xử lý và được sinh nhiều dữ liệu. Tiếp theo, dữ liệu được học và dự đoán ở bước Modeling trước khi được đánh giá và đưa lên hệ thống.

3.4. Nhận xét

Hầu hết các phương pháp bị quá khớp với dữ liệu training set:

- Traditional machine learning (98% acc training set, 76% acc validation set)
- DesenNet 2D Conv (99.89% acc training set, 90,8% acc validation set)
- WavemsNet (99.9% acc training set, 80.8% acc validation set)

Việc sử dụng Augmentation data trong DesenNet 2D Conv đã cải thiện đáng kể model bị quá khớp với dữ liệu training.

Và kết quả trên tập public test thấp. Nguyên nhân là do mismatch data. Phân bố dữ liệu của tập public test rất khác so với tập training set và validation set.

3.5. Hướng phát triển.

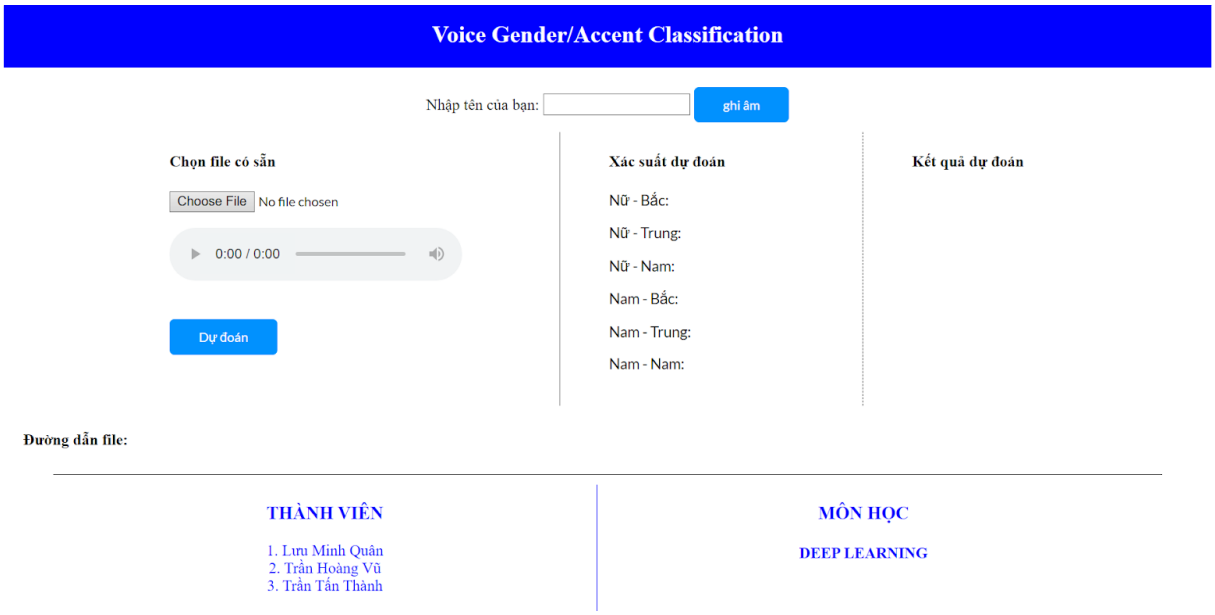
Cải thiện hệ thống phân loại giới tính và vùng miền dựa trên âm thanh bằng việc bổ sung dữ liệu cho tập training. Vừa giúp model khỏi bị quá khớp với giữ liệu training vừa giải quyết vấn đề mismatch data.

Tích hợp hệ thống nhận diện giọng nói (Automatic Speech Recognition)

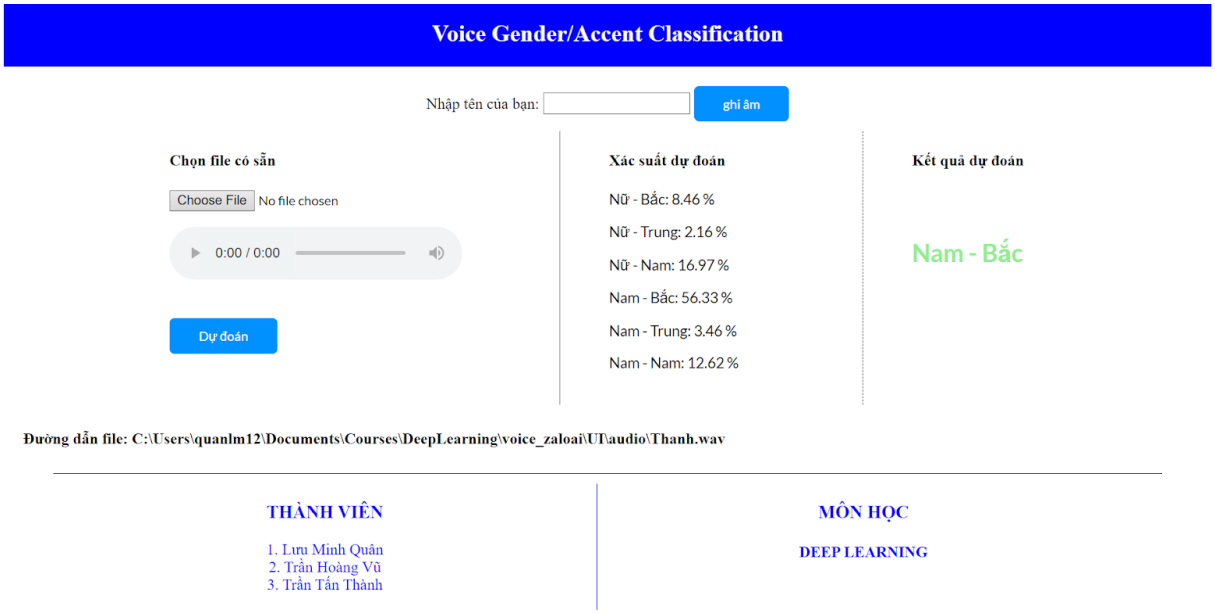
CHƯƠNG 4

DEMO

Model sau khi được huấn luyện được đưa lên một dịch vụ web để người dùng có thể dễ dàng nhập giọng nói của mình và sử dụng hệ thống phân loại



Người dùng có thể hoặc tải lên file âm thanh chưa giọng nói của mình hoặc điền tên và ghi âm trực tiếp trên hệ thống



Sau đó hệ thống sẽ cho ra kết quả phân loại cùng với xác suất các lớp

Bibliography

Choi, K., and G. Fazekas. *Automatic tagging using deep convolutional neural networks*. arXiv preprint arXiv:1606.00298, 2016.

Dieleman, S., and B. Brakel. *Audio-based music classification with a pretrained convolutional network*. ISMIR, 2011.

Gwardys, G., and D. M. Grzywczak. “Deep image features in music information retrieval.” *International Journal of Electronics and Telecommunications*, vol. 60, no. 4, 2014, pp. 321-326.

Huang, G., and Z. Liu. *Densely connected convolutional networks*. Proceedings of the IEEE conference on computer vision and pattern recognition, 2017.

Lee, J., and J. Park. *“Sample-level deep convolutional neural networks for music auto-tagging using raw waveforms*. arXiv preprint arXiv:1703.01789, 2017.

Soloyyev, R. A., and M. Vakhrushev. *Deep Learning Approaches for Understanding Simple Speech Commands*. IEEE 40th, 2020.