

ĐẠI HỌC KHOA HỌC TỰ NHIÊN, ĐHQG-HCM
KHOA CÔNG NGHỆ THÔNG TIN
BỘ MÔN KHOA HỌC MÁY TÍNH



BÁO CÁO SEMINAR CUỐI KỲ
NHẬN ĐẶNG
SINH TRẮC HỌC GIỌNG NÓI

Giảng viên lý thuyết
PGS. TS. Lê Hoàng Thái

Giảng viên hướng dẫn

Lê Thanh Phong, Nguyễn Ngọc Thảo

Sinh viên thực hiện

Nguyễn Hoàng Đức, Lê Nhựt Nam, Nguyễn Viết Dũng

Lời cảm ơn

Trong quá trình thực hiện đồ án này, chúng em đã nhận được rất nhiều sự giúp đỡ cũng như hỗ trợ từ các thầy cô Trường Đại học Khoa học Tự nhiên, ĐHQG-HCM và các bạn bè trong lớp Nhân dạng. Chúng em xin bày tỏ lòng cảm ơn chân thành đến mọi người vì đã giúp đỡ hướng dẫn, chỉ bảo rất tận tình.

Đặc biệt, chúng em xin bày tỏ lòng biết ơn sâu sắc đến các thầy cô khoa Công nghệ Thông tin, cụ thể hơn là thầy Hoàng Thái, thầy Thanh Phong và cô Ngọc Thảo, đã đã giảng dạy rất nhiệt, cung cấp nhiều slides, tài nguyên học tập cần thiết, tạo điều kiện tốt nhất để chúng em có thể hoàn thành được đồ án này.

Trong lúc viết báo cáo này, chúng em không thể tránh khỏi nhiều thiếu sót, hy vọng mong nhận được góp ý từ thầy để chúng em tiếp tục hoàn thiện hơn đối với đồ án này, cũng như rút kinh nghiệm cho những đồ án, những báo cáo kế tiếp.

Đại học Khoa học Tự nhiên, ĐHQG-HCM.

Nguyễn Hoàng Đức, Lê Nhựt Nam, Nguyễn Viết Dũng

Tháng 5 năm 2021

Mục lục

Lời cảm ơn	i
A. Thông tin nhóm	1
B. Thông tin đề tài	1
C. Nội dung phân công	1
D. Nội dung báo cáo	2
D.1 Nội dung tìm sách Handbook of Biometrics	2
1. Giới thiệu	2
1.1 Các ứng dụng	2
1.2 Công nghệ	2
2. Thông tin nhận dạng trong tín hiệu giọng nói	2
2.1 Đặc trưng Idiolectal	3
2.2 Đặc trưng ngữ âm	3
2.3 Đặc trưng ngữ điệu	3
2.3 Đặc trưng phổ	4
3. Rút trích đặc trưng và Tokenization	4
3.1 Phân tích cửa sổ (Short-term Analysis)	4
3.2 Tham số hóa (Parameterization)	4
3.3 Phân tích ngữ âm và tách từ (Phonetic and word tokenization)	6
3.4 Phân tách ngữ điệu (Prosodic Tokenization)	6
4. Nhận dạng giọng nói phụ thuộc văn bản	6
4.1 Phân loại các hệ thống và kỹ thuật	7
4.2 Kho ngữ liệu	7
4.3 Công trình điển hình	7
5. Nhận dạng giọng nói độc lập với văn bản	8
5.1 Short-term spectral systems - Hệ thống cửa sổ phổ âm	8
5.2 Idiolectal systems - Hệ thống Idiolectal	9
5.3 Phonotactic systems - Hệ thống ngữ âm	9
5.4 Prosodic systems - Hệ thống ngữ điệu	11
5.5 Kho ngữ liệu và benchmarks	12
5.6 Công trình điển hình	12
D.2 Các phương pháp SOTA - State of The Art gần đây về Voice Biometrics	14
D.2.1 Giới thiệu	14
D.2.2 Động lực nghiên cứu	14
D.2.3 Phát biểu bài toán	14
D.2.4 Một số kho ngữ liệu	15
D.2.5 Kiểm định mô hình nhận dạng giọng nói	16
D.2.5 Các độ đo trong nhận dạng giọng nói	16
D.2.5 Học Sâu trong tác vụ rút trích đặc trưng giọng nói	18
D.2.5.1 Về d-vectors	18
D.2.5.2 Về j-vectors	21
D.2.5.3 Về x-vectors	25
D.2.5.4 So sánh d-vectors, j-vectors và x-vectors	29
D.2.5 Học Sâu trong tác vụ phân lớp giọng nói	29
D.2.5.1 Multi-domain features	29
D.2.5.1 SincNet	33
E. Thực nghiệm của nhóm	42
E.1 Phương pháp	42
E.2 Kho ngữ liệu	43
E.3 Thực nghiệm	43

E.3.1 Thực nghiệm trên tập TIMIT	43
E.3.2 Thực nghiệm trên tập Librispeech	44
E.3.2 Thực nghiệm trên tập Son et al. Dataset	45
E.4 Dánh giá mô hình	45
E.5 Các kết quả	46
E.5.1 Kết quả thực nghiệm trên tập TIMIT	46
E.5.2 Kết quả thực nghiệm trên tập Librispeech	48
E.5.3 Kết quả thực nghiệm trên tập Son et al. Dataset	50
E.5.4 Suy luận từ mô hình nhận dạng giọng nói tiếng Việt	52
E.6 Những nhận xét	54

A. Thông tin nhóm

STT	MSSV	Họ tên đầy đủ	Email liên lạc
1	18120018	Nguyễn Hoàng Đức	18120018@student.hcmus.edu.vn
2	18120061	Lê Nhựt Nam	18120061@student.hcmus.edu.vn
3	18120167	Nguyễn Viết Dũng	18120167@student.hcmus.edu.vn

B. Thông tin đề tài

Tên đề tài:

- Tên đề tài (Tiếng Việt): Sinh trắc học giọng nói
- Tên đề tài (Tiếng Anh): Voice Biometrics

Nguồn:

- Sách: Chương 8: Voice Biometrics, Handbook of Biometric, Anil K. Jain, Patrick Flynn, Arun A. Ross
- Github - Nguồn Source Code
- Paper with Code - Nguồn paper kèm theo source cài đặt chính thức của nhóm tác, cộng đồng

Từ khóa:

- Tên từ khóa (Tiếng Việt): Nhận dạng sinh trắc học, Nhận dạng giọng nói, Nhận dạng người nói, Mạng Neural Tích chập, Mẫu thô, Xác minh người nói, Định danh người nói
- Tên từ khóa (Tiếng Anh): Biometric Recognition, Voice Recognition, Speaker Recognition, Convolutional Neural Networks, Raw Samples, Speaker Verification, Speaker Identification

C. Nội dung phân công

STT	MSSV	Họ tên	Nội dung công việc	Hoàn thiện (%)
1	18120018	Nguyễn Hoàng Đức	Thu thập dữ liệu, đọc hiểu source, báo cáo Related Work Paper	100%
2	18120061	Lê Nhựt Nam	Thu thập dữ liệu, đọc hiểu source, báo cáo, SincNet Architecture, Slides thuyết trình, Midterm Report	100%
3	18120167	Nguyễn Viết Dũng	Thu thập dữ liệu, đọc hiểu source, báo cáo experimental setup Paper	100%

29/03 - 04/04	• Tiếp nhận đồ án, phân công công việc
05/04 - 11/04	• Đọc source code mẫu từ Github, trình bày những gì tìm hiểu được
12/04 - 18/04	• Chạy thử source code mẫu từ Github, trình bày những gì tìm hiểu được
19/04 - 25/04	• Trình bày những gì tìm hiểu được, phân công làm slides, trình bày đồ án
26/04 - 02/05	• Thu thập dữ liệu, tiền xử lý dữ liệu
03/05 - 09/05	• Tinh chỉnh source code, chạy thử mô hình
10/05 - 16/05	• Huấn luyện mô hình
17/05 - 23/05	• Phân công làm slides, trình bày đồ án
24/05 - 30/05	• *Dự phòng
31/05 - 06/06	• *Dự phòng

D. Nội dung báo cáo

D.1 Nội dung tìm sách Handbook of Biometrics

1. Giới thiệu

Trong thời gian gần đây, dữ liệu về người dùng điện thoại di động trên toàn cầu, số lượng điện thoại cố định đang trong tình trạng hoạt động, việc triển khai VoIP (Voice over IP networks - Mạng hội thoại IP), cho thấy rằng giọng nói (voice) là một đặc điểm sinh trắc học (nhân trắc học) dễ dàng tiếp cận nhất mà không cần phải có thêm thiết bị thu nhận và hệ thống truyền dẫn.

Những dữ kiện trên cho thấy giọng nói một lợi thế so với những đặc điểm sinh trắc học (nhân trắc học) khác, đặc biệt là khi người dùng hoặc các hệ thống điều khiển từ xa được tính đến. Tuy nhiên, đặc điểm giọng nói không chỉ liên đến các đặc trưng cá thể mà còn liên quan với môi trường xung quanh và vấn đề xã hội, do vậy việc tạo thành giọng nói là một kết quả của một quá trình hết sức phức tạp.

Vì thế, việc truyền giọng nói sẽ làm mất đi nhiều đặc trưng của giọng nói tùy thuộc vào đặc điểm của giọng nói và sẽ bị ảnh hưởng bởi ngữ cảnh, gây khó khăn trong việc xử lý. May mắn thay, các công nghệ và ứng dụng tiên tiến có thể khắc phục những vấn đề trên, cho phép cải thiện độ hiệu quả và tin cậy cho việc xác nhận từ xa hoặc nhận diện giọng nói chỉ dựa vào tín hiệu giọng nói truyền qua sóng điện thoại.

1.1 Các ứng dụng

Do tính phổ biến của tín hiệu giọng nói, phạm vi ứng dụng có thể có của sinh trắc học (nhân trắc học) giọng nói rộng hơn so với các đặc điểm sinh trắc học thông thường khác. Chúng ta có thể phân biệt ba loại ứng dụng chính tận dụng thông tin sinh trắc học có trong tín hiệu giọng nói như sau:

- **Voice authentication** (Xác nhận giọng nói) (Access control - Điều khiển truy cập, thường là điều khiển từ xa bằng điện thoại) và back-ground recognition (Nhận dạng lý lịch) (natural voice checking - kiểm tra bằng giọng nói tự nhiên)
- **Speaker detection** (Nhận diện người nói) (ví dụ như: phát hiện danh sách đen trong các tổng đài điện thoại hoặc trong nghe lén và giám sát, ...), hay còn được gọi là speaker spotting
- **Forensic speaker recognition** (Nhận dạng người nói trong Pháp y) (sử dụng giọng nói làm bằng chứng trước tòa án hoặc làm thông tin tình báo trong các cuộc điều tra của cảnh sát hình sự)

1.2 Công nghệ

1.2.1 Text-dependent Đầu tiên, công nghệ **text-dependent**, trong đó user được yêu cầu phải nói ra một từ khóa (Ví dụ: "Open, Sesame", "Vìtng ơi! Mở ra!") hoặc chuỗi (Ví dụ: "12-34-56"), đã trở thành chủ đề chính của nhân trắc học điều khiển truy cập và ứng dụng xác minh giọng nói.

1.2.1 Text-independent Loại thứ hai của công nghệ nhận dạng người nói được biết như là **text-independent**. Nó là nhân tố thúc đẩy sự phát triển của hai loại ứng dụng còn lại, cụ thể là **Speaker detection** (Nhận diện người nói) và **Forensic speaker recognition** (Nhận dạng người nói trong Pháp y). Vì nội dung ngôn ngữ là nguồn thông tin chính được mã hóa trong lời thoại, tính độc lập với văn bản đã là một thách thức lớn và là chủ đề nghiên cứu chính của cộng đồng Nhận dạng người nói trong suốt hai thập kỷ qua. NISTSRE (**Speaker Recognition Evaluations**) được tiến hành hàng năm kể từ năm 1996 đã thúc đẩy sự xuất sắc trong nghiên cứu trong lĩnh vực này, với tiến bộ thường đạt được qua từng năm dựa trên đánh giá mờ với các cơ sở dữ liệu và giao thức chung, và đặc biệt là việc chia sẻ thông tin giữa những người tham gia hội thảo tiếp theo sau mỗi lần phát triển.

2. Thông tin nhận dạng trong tín hiệu giọng nói

Sự tạo thành giọng nói là quá trình cực kỳ phức tạp mà kết quả của nó phụ thuộc vào nhiều tham biến ở nhiều cấp độ khác nhau:

- Các yếu tố xã hội học (Ví dụ như: trình độ học vấn, ngữ cảnh nôn ngữ, sự khác biệt vùng miền)
- Các vấn đề sinh lý (Ví dụ như: chiều dài đường thanh âm, hình dạng và mô và cấu trúc động của các cơ quan cấu âm)
- ...

Những ảnh hưởng này sẽ xuất hiện đồng thời trong mỗi hành động lời nói và một số hoặc tất cả chúng sẽ chứa đựng những đặc điểm cụ thể của giọng nói. Vì lý do đó, chúng ta cần làm rõ và phân biệt rõ ràng các cấp độ và nguồn thông tin giọng nói khác nhau mà chúng ta có thể trích xuất để mô hình hóa tính đặc biệt của giọng nói.

Quá trình mà con người có thể xây dựng một thông điệp được mã hóa bằng ngôn ngữ đã là một chủ đề nghiên cứu trong nhiều năm trong lĩnh vực Ngôn ngữ học tâm lý (Psycholinguistics). Nhưng một khi thông điệp đã được mã hóa trong não người, vẫn cần một quá trình sinh lý và ngữ âm khớp phức tạp để cuối cùng tạo ra một dạng sóng lời nói (giọng nói) chứa thông điệp ngôn ngữ (cũng như nhiều nguồn thông tin khác, một trong số đó là nhận dạng người nói) được mã hóa như một sự kết hợp của các đặc điểm phổ thời gian.

Quá trình hình thành giọng nói là chủ đề nghiên cứu của các nhà ngữ âm học và một số lĩnh vực liên quan đến phân tích giọng nói khác (kỹ sư, bác sĩ, v.v.). Trong cả hai giai đoạn của quá trình tạo giọng nói (tạo ngôn ngữ và tạo giọng nói), các đặc điểm cụ thể của người nói đều được giới thiệu. Trong lĩnh vực sinh trắc học giọng nói - còn được gọi là nhận dạng người nói - hai thành phần này tương ứng với nó thường được gọi là đặc điểm cấp cao (ngôn ngữ) và cấp thấp (âm thanh).

2.1 Đặc trưng Idiolectal

Idiolectal characteristics - Đặc trưng Idiolectal Idiolectal = idio (personal, private) + (dia)lect

- Thói quen nói đặc biệt của một người cụ thể.
- Một "Idiolectal" là lời nói đặc biệt của một cá nhân, một ngôn ngữ được coi là duy nhất trong số những người nói ngôn ngữ hoặc phương ngữ của một người. Nhưng nó thậm chí còn hẹp hơn nhiều so với tất cả những người nói một phương ngữ cụ thể.
- Ngôn ngữ đa dạng duy nhất cho một người nói một ngôn ngữ được gọi là idiolect. idiolect của bạn bao gồm các từ vựng phù hợp với các sở thích và hoạt động khác nhau của bạn, cách phát âm phản ánh khu vực bạn đang sống hoặc đã sống và các phong cách nói khác nhau thay đổi một cách tinh vi tùy thuộc vào người bạn đang nói đến

2.2 Đặc trưng ngữ âm

Phonotactics characteristics - Đặc trưng ngữ âm Mô tả cách sử dụng của người nói của các đơn vị ngữ âm và khả năng sử dụng có thể khác nhau giữa những người nói.

Ngữ âm là rất cần thiết cho việc sử dụng đúng đắn về một ngôn ngữ, và một chìa khóa trong việc học ngoại ngữ, nhưng khi chúng ta nhìn vào nét đặc trưng phonotactic người nói, chúng ta có thể tìm thấy một số kiểu sử dụng khác biệt với những người dùng khác.

2.3 Đặc trưng ngữ điệu

Prosody characteristics - Đặc trưng ngữ điệu Prosody (ngữ điệu), là sự kết hợp của năng lượng tức thời, âm điệu, tốc độ nói và thời lượng đơn vị để cung cấp cho lời nói sự tự nhiên, đầy đủ ý nghĩa và giọng điệu cảm xúc.

Prosody (ngữ điệu) xác định các đối tượng âm điệu ở mức độ cụm và đoạn đàm thoại, và định nghĩa nhanh chóng hành động để phù hợp với các đối tượng trên. Nó giúp làm rõ thông điệp ("nine hundred twenty seven" có thể phân biệt "927" hoặc "900 27" bởi ý nghĩa của ngữ điệu), loại thông điệp (trần thuật, nghi vấn, mệnh lệnh), hoặc trạng thái suy nghĩ của người nói.

Những theo cách mà mỗi người nói sử dụng các thành phần ngữ điệu khác nhau, nhiều đặc trưng của người bị trùng lặp, ví dụ như các đường đặc trưng bị trùng nhau ở điểm đầu và điểm của của một cụm hoặc nhóm giọng nói.

2.3 Đặc trưng phô

Spectral characteristics - Đặc trưng phô Đặc trưng phô rời rạc của tín hiệu giọng nói, liên quan trực tiếp với hành động phát âm đơn lẻ, quan hệ với tạo ra ngữ âm và cũng liên quan đến sinh lý cá nhân của quá trình phát sinh giọng nói.

Thông tin về phô là trọng tâm của sự phân biệt giọng nói, thường được dùng trong các ứng dụng, và là nghiên cứu trọng tâm trong hàng 20 năm qua. Thông tin phô chú trọng vào rút trích các đặc điểm trong giọng nói và động lực phát âm tương ứng của người nói.

Hai loại của thông tin cấp thấp thường được dùng:

- Thông tin tĩnh có quan hệ với phân tích từng frame
- Thông tin động có quan hệ với cách mà thông tin phát triển trong các frame liền kề, có tính đến hiện tượng nối âm, phụ thuộc nhiều vào người nói, quá trình mà một cá nhân tự động di chuyển từ vị trí phát âm này sang vị trí phát âm tiếp theo.

3. Rút trích đặc trưng và Tokenization

Bước đầu trong việc xây dựng một hệ thống nhận dạng người nói tự động là rút trích các đặc trưng đáng tin cậy và những tokens có chứa thông tin nhận dạng.

3.1 Phân tích cửa sổ (Short-term Analysis)

Để thực hiện phân tích phô đáng tin cậy, các tín hiệu phải thể hiện các đặc tính tĩnh, điều này không dễ quan sát trong các tín hiệu giọng nói thay đổi liên tục.

Để giải quyết vấn đề trên, người ta thường giới hạn cửa sổ phân tích, thường dùng cửa sổ dạng cosine như hamming hoặc hanning, với độ dài ngắn từ 20 đến 40 mili giây, hệ thống ngữ âm của chúng ta không thể thay đổi đáng kể trong một khung thời gian ngắn như vậy, những gì thu được thường được gọi là tín hiệu giả tĩnh trên mỗi khung. Quá trình này được mô tả trong hình 8.1 sách Handbook of Biometrics.

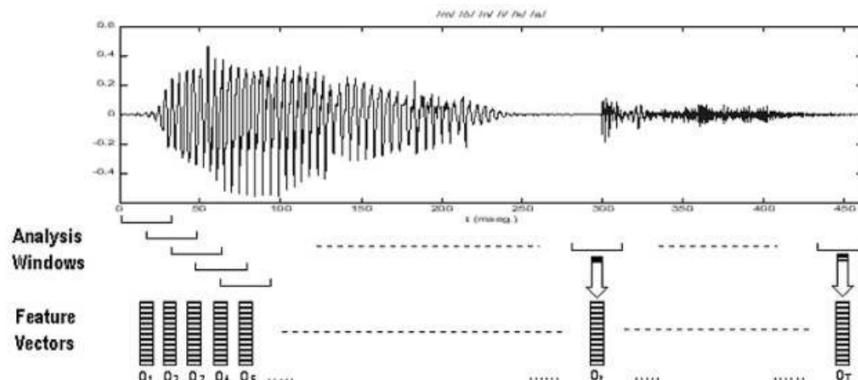


Fig. 8.1. Short-term analysis and parameterization of a speech signal.

Hình 1: Phân tích theo từng đoạn ngắn và tham số hóa tín hiệu giọng nói

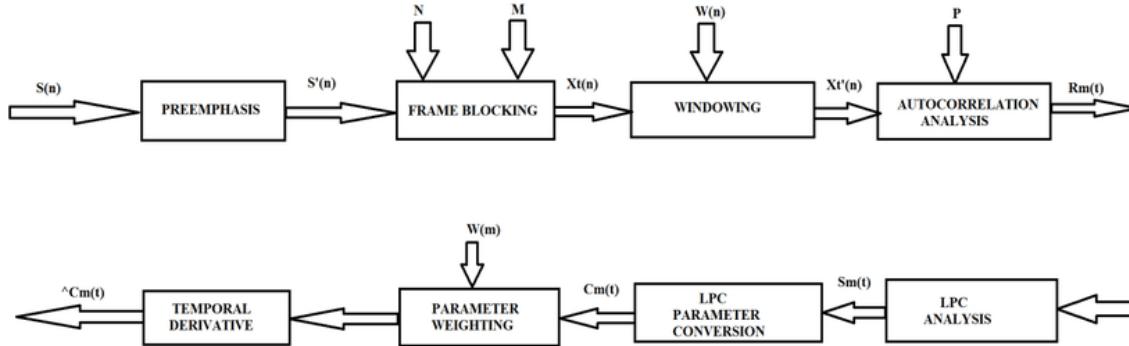
3.2 Tham số hóa (Parameterization)

Tín hiệu cửa sổ hamming/ hanning trong thời gian ngắn này có tất cả thông tin thời gian/ phô mong muốn, mặc dù ở tốc độ bit cao (ví dụ: số hóa giọng nói điện thoại với tần số lấy mẫu 8 kHz trong một cửa sổ 32 ms. Có nghĩa là $256 \text{ mẫu} \times 16 \text{ bit} / \text{mẫu} = 4096 \text{ bits} = 512 \text{ bytes} \text{ mỗi khung}$).

Phương pháp tham số hóa

- Linear Predictive Coding (LPC)
- Mel-Frequency based Cepstral Coefficients (MFCC)

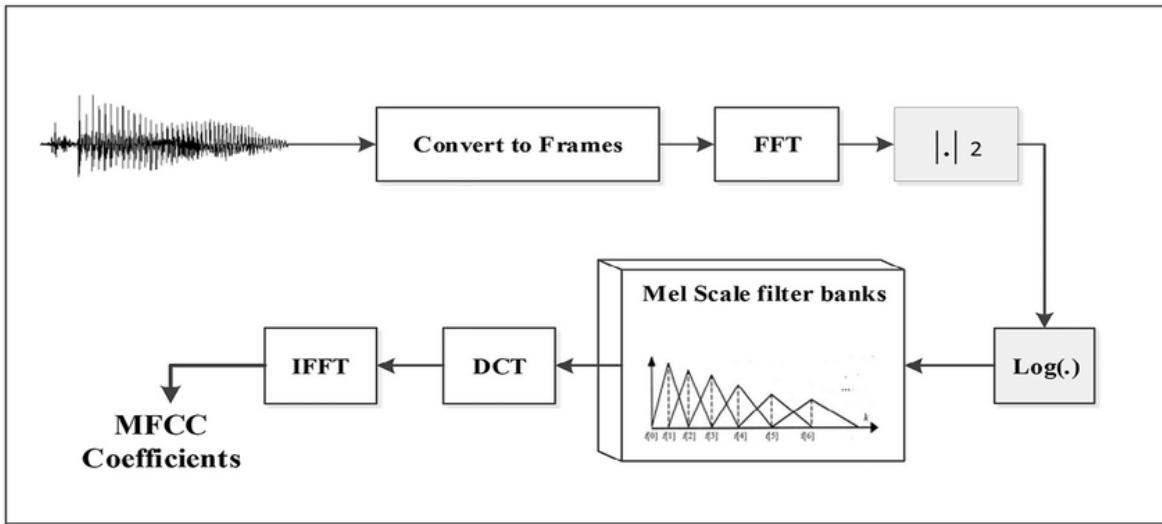
Linear Predictive Coding (LPC)



Hình 2: Linear Predictive Coding (LPC) Diagram

Linear Predictive Coding là một phương pháp sử dụng chủ yếu trong xử lý tín hiệu âm thanh và lời nói cho kết quả là đường bao phổ (đường cong đường bao của phổ biên độ - mô tả một điểm trong thời gian) của một số tín hiệu giọng nói dưới dạng nén, sử dụng thông tin của một mô hình dự đoán tuyến tính.

Mel-Frequency based Cepstral Coefficients (MFCC)



Hình 3: Mel-Frequency based Cepstral Coefficients (MFCC) Diagram

Mel-Frequency based Cepstral Coefficients là một phương pháp để rút trích các đặc trưng (feature extraction) giọng nói thường được sử dụng trong các mô hình nhận dạng giọng nói tự động (Automatic Speech Recognition) hay phân lớp giọng nói (Speech Classification). Nó cho ra các kết quả là các hệ số (Coefficients) của cepstral từ Mel Filter trên phổ từ dữ liệu âm thanh giọng nói.

Nguyên lý hoạt động của MFCC: Dữ liệu giọng nói thường được biểu diễn dưới dạng speech signal (tín hiệu giọng nói), hai chiều (x, y) với x là thời gian, y là biên độ. Speech signal sẽ được biến đổi

thành phổ âm (spectrum) bằng cách sử dụng phép biến đổi Fourier (Fourier Transformation, để thực hiện nhanh, người ta thường dùng Fast Fourier Transform Algorithm), kết quả là spectrum được biểu diễn lại ở hai chiều mới (x', y'), với x' là tần số và y' là cường độ.

Trong spectrum thu được, gọi spectrum này là $X[k]$, có hai thành phần là spectral envelopes $H[k]$ và spectral details $E[k]$. Người ta dùng logarithm để tách spectral envelopes $H[k]$ và lấy phần ở tần số thấp

$$X[k] = H[k] * E[k] \Leftrightarrow \log(X[k]) = \log(H[k]) + \log(E[k])$$

Thay vì phải tập trung vào tất cả thông tin, tai con người hoạt động như một bộ lọc (filters), vì ý tưởng này, người ta chỉ cần tập trung vào một phần spectral envelopes. Sau khi dùng bộ lọc, người ta sử dụng phép biến đổi Fourier Ngược (Inverse Fourier Transformation, thông thường để đạt hiệu suất cao người ta sử dụng Fast Inverse Fourier Transform Algorithm)

$$IFFT(\log(X[k])) = IFFT(\log(H[k]) + \log(E[k])) \Leftrightarrow x[k] = h[k] + e[k]$$

Khi đó, $x[k]$ được gọi là Cepstrum, nghịch đảo của Spectrum

Chúng ta có một pipeline nguyên lý hoạt động của MFCC như sau: speech signal \Rightarrow spectrum \Rightarrow mel-freq filter \Rightarrow cepstral

3.3 Phân tích ngữ âm và tách từ (Phonetic and word tokenization)

Mô hình Markov ẩn (HMM - Hidden Markov Models) là công cụ thành công nhất và được sử dụng rộng rãi (ngoại trừ một số kiến trúc ANN) để mã hóa ngữ âm, âm tiết và từ, nghĩa là dịch từ lời nói được lấy mẫu thành một căn chỉnh thời gian dãy các đơn vị ngôn ngữ.

Huấn luyện HMM thường được thực hiện thông qua ước lượng Baum-Welch, trong khi giải mã và căn chỉnh thời gian thường được thực hiện thông qua giải mã Viterbi. Hiệu suất của các phổ HMM đó được cải thiện bằng cách sử dụng các mô hình ngôn ngữ, mô hình này áp đặt một số ràng buộc về ngôn ngữ hoặc ngữ pháp đối với sự kết hợp gần như vô hạn của tất cả các đơn vị có thể. Để cho phép tăng hiệu quả, việc lược bỏ tìm kiếm chùm tia cũng là một cơ chế tổng quát để đẩy nhanh đáng kể quá trình nhận dạng mà không có hoặc ít suy giảm hiệu suất.

3.4 Phân tách ngữ điệu (Prosodic Tokenization)

Các đặc trưng ngữ điệu cơ sở như cao độ và năng lượng cũng có được ở mức frame. Năng lượng cửa sổ thu được rất dễ dàng thông qua định lý Parseval, ở dạng thời gian hoặc dạng phổ, và cao độ tức thời có thể được xác định bằng, ví dụ như, phương pháp tự động tương quan hoặc dựa trên phân rã cepstral, thường được làm mịn bằng một số lọc thời gian.

Các đặc điểm thuận âm quan trọng khác là những đặc điểm liên quan đến thời lượng của các đơn vị ngôn ngữ, tốc độ nói và tất cả những đặc điểm liên quan đến trọng âm.

Trong tất cả những trường hợp đó, cần phải phân đoạn chính xác, đánh dấu các vị trí âm tiết, đường nét năng lượng và cao độ để phát hiện các vị trí trọng âm và dấu chuyển cụm từ hoặc giọng nói.

Việc phân đoạn ngữ âm và âm tiết của lời nói là một vấn đề phức tạp còn lâu mới giải quyết được và mặc dù nó có thể hữu ích cho việc nhận dạng giọng nói, các hệ thống âm tiết không phải lúc nào cũng yêu cầu phân đoạn chi tiết như vậy.

4. Nhận dạng giọng nói phụ thuộc văn bản

Hệ thống nhận dạng giọng nói phụ thuộc văn bản, sử dụng nội dung từ vựng của giọng nói phát ra để nhận dạng giọng nói, ứng dụng chính của hệ thống này trong các hệ thống tương tác, nơi cần có sự hợp tác từ người dùng để xác thực danh tính của họ.

Ví dụ điển hình của các ứng dụng này là xác thực bằng giọng nói qua điện thoại cho các hệ thống phản hồi giọng nói tương tác yêu cầu một số mức độ bảo mật như các ứng dụng ngân hàng hoặc đặt lại mật khẩu.

Tương tự như các phương thức sinh trắc học khác, việc sử dụng hệ thống nhận dạng giọng nói phụ thuộc vào văn bản yêu cầu giai đoạn đăng ký trong đó người dùng cung cấp một số mẫu để xây dựng mô hình người dùng và giai đoạn nhận dạng trong đó mẫu giọng nói mới được so khớp với mô hình người dùng.

4.1 Phân loại các hệ thống và kỹ thuật

Chúng ta có thể phân loại hệ thống nhận dạng người nói phụ thuộc vào văn bản theo quan điểm ứng dụng thành hai loại

- **Hệ thống văn bản tĩnh:** nội dung từ vựng trong ghi danh và các mẫu nhận dạng luôn giống nhau. Trong các **hệ thống văn bản động**, nội dung từ vựng trong mẫu nhận dạng là khác nhau trong mọi thử nghiệm truy cập với nội dung từ vựng của các mẫu đăng ký.
- **Hệ thống văn bản động:** tạo ra một lời nhắc mật khẩu được tạo ngẫu nhiên khác nhau mỗi khi người dùng được xác minh (hệ thống nhắc bằng văn bản), do đó hầu như không thể sử dụng bản ghi nên linh hoạt hơn và mạnh mẽ hơn trước các cuộc tấn công sử dụng bản ghi âm từ người dùng hoặc bắt chuốc sau khi nghe người nói thực sự nói đúng mật khẩu

Tuy nhiên, thông tin được sử dụng rộng rãi nhất là thông tin **phổ của tín hiệu tiếng nói**, được xác định bởi cấu hình vật lý và động lực của đường thanh quang. Thông tin này thường được tóm tắt dưới dạng chuỗi thời gian của các vector MFCC, mỗi vector trong số đó đại diện cho một thời lượng nói từ 20-40 mili giây. Bằng cách này, vấn đề nhận dạng người nói phụ thuộc vào văn bản được giảm xuống thành vấn đề so sánh chuỗi các vector MFCC với mô hình của người dùng.

Có hai phương pháp đã được sử dụng rộng rãi:

- **Phương pháp dựa trên khuôn mẫu:** bao gồm một số chuỗi vector tương ứng với lời nói đăng ký và việc nhận dạng được thực hiện bằng cách so sánh lời nói xác minh với lời nói đăng ký. So sánh này được thực hiện bằng cách sử dụng Dynamic Time Warping (DTW) như một cách hiệu quả để cải thiện sai lệch thời gian giữa các cách phát âm khác nhau.
- **Phương pháp thống kê:** nổi bật nhất là **mô hình Markov ẩn (HMM)**, có xu hướng được sử dụng thường xuyên hơn các mô hình dựa trên khuôn mẫu. HMMs cung cấp tính linh hoạt hơn, cho phép chọn đơn vị tiếng nói từ đơn vị âm vị phụ đến từ và cho phép thiết kế hệ thống nhắc văn bản.

4.2 Kho ngữ liệu

- YOHO Speaker Verification
- MIT Mobile Device Speaker Verification Corpus
- BIOSEC Baseline Corpus

4.3 Công trình điển hình

Tên công trình Text-dependent speaker recognition with HMM speaker adaptation and HMM reestimation

Hệ thống nhận dạng giọng nói phụ thuộc văn bản được thử nghiệm trên cơ sở dữ liệu chuẩn YOHO, hai hệ thống nhận dạng người nói phụ thuộc vào văn bản do các tác giả phát triển. Các hệ thống này mô phỏng một hệ thống được nhắc bằng văn bản dựa trên một tập hợp các HMM ngữ âm không phụ thuộc vào người nói và không phụ thuộc vào ngữ cảnh được đào tạo trên TIMIT.

Giai đoạn đăng ký sử dụng một số câu của người nói để điều chỉnh HMM cho người đó. Ta so sánh hai cách thức thực hiện sự thích ứng này: Baum-Welch Reestimation và MLLR (maximum likelihood linear regression). Trong đó, Baum-Welch Reestimation là cách tiếp cận thông thường nhất nhưng yêu cầu sử dụng HMM rất đơn giản (chỉ một hoặc một vài Gaussian cho mỗi trạng thái) còn MLLR thì mới hơn và cho phép sử dụng các HMM phức tạp hơn.

Một vấn đề quan trọng trong việc phát triển hệ thống nhận dạng giọng nói phụ thuộc văn bản là số lượng tài nguyên huấn luyện. YOHO chứa 4 phần với 24 câu nói mỗi phần, đây là một lượng tài nguyên đáng kể để huấn luyện một cách hiệu quả.

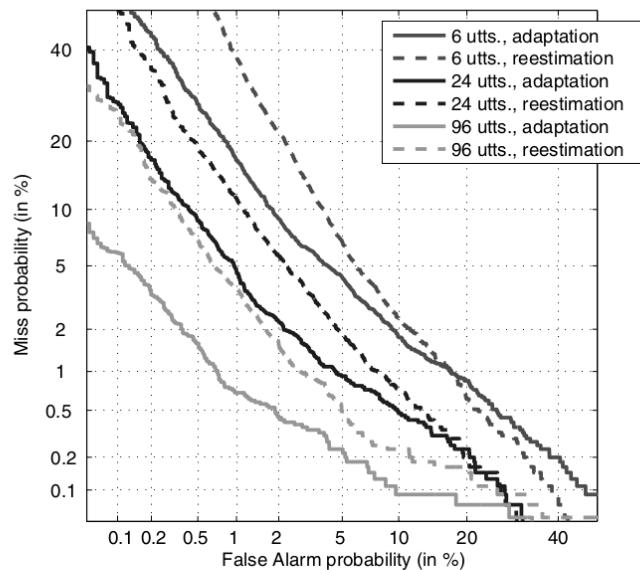


Fig. 8.2. Example results on YOHO of two text-dependent speaker recognition systems based on speaker-independent phonetic HMMs and MLLR speaker-adaptation and Baum-Welch re-estimation for different amounts of enrollment speech.

Hình 4: Text-dependent speaker recognition with HMM speaker adaptation and HMM reestimation

5. Nhận dạng giọng nói độc lập với văn bản

Hệ thống nhận dạng giọng nói độc lập với văn bản có gắng giảm thiểu ảnh hưởng của nội dung từ vựng vốn được coi là không xác định đối với khả năng nhận dạng của giọng nói, điều này trái ngược với hệ thống nhận dạng giọng nói phụ thuộc văn bản, đương nhiên việc nghiên cứu và phát triển nó sẽ khó khăn hơn.

5.1 Short-term spectral systems - Hệ thống cửa sổ phô âm

Khi phân tích phô trong khoảng thời gian ngắn được sử dụng để mô hình các đặc trưng người nói, chúng ta đang mô hình hóa các “âm thanh” khác nhau mà một người có thể tạo ra, nhờ vào đường âm và các cơ quan cấu âm của họ. Khi con người cần nhiều âm thanh (hoặc các ký hiệu âm học khác nhau) để nói bằng bất kỳ ngôn ngữ chung nào, chúng ta rõ ràng đang đối mặt với một **không gian đa lớp gồm nhiều đặc trưng**.

Kỹ thuật Lượng tử hóa Vector (**Vector Quantization techniques**) hiệu quả trong các bài toán đa lớp như vậy và đã được sử dụng để xác định người nói, điển hình là một mô hình VQ cụ thể cho một người nói, tính toán khoảng cách từ bất kỳ câu nói nào đến bất kỳ mô hình nào dưới dạng tổng trong số của khoảng cách tối thiểu trên mỗi khung đến codevector gần của codebook. Việc sử dụng các giới hạn và các điểm trung tâm thay vì dùng mật độ xác suất làm hiệu suất của VQ kém hơn so với mô hình Markov ẩn với mật độ liên tục và liên thông hay còn gọi là Ergodic HMMs.

Tuy nhiên, yếu tố quan trọng trong **E-HMM** là tích số trạng thái với hàm Gaussian mỗi trạng thái, điều này loại bỏ triệt để ảnh hưởng của quá trình chuyển đổi trong mô hình liên thông. Sau đó, một hệ thống HMM với 5 trạng thái - 4 Gaussian cho mỗi trạng thái sẽ hoạt động tương tự như 4-trạng

thái 5-Gaussian, 2-trạng thái 10-Gaussian, hoặc thậm chí là 1-trạng thái 20-Gaussian, mà một cách hiểu tổng quát là GMM (**Gaussian Mixture Model**). Những one-state **E-HMMs** hoặc **GMMs** này có lợi thế lớn, tránh được cả ước lượng Baum-Welch cho việc huấn luyện, không cần sự liên kết giữa lời nói và trạng thái (tất cả lời nói đều được tinh chỉnh với cùng một trạng thái duy nhất) và dùng Viterbi decoding cho việc kiểm thử (không cần tinh chỉnh thời gian), giúp tăng tốc thời gian tính toán mà không ảnh hưởng đến hiệu suất.

GMM là một kỹ thuật tổng hợp trong đó một hỗn hợp các hàm Gaussians Da chiều có gắng mô hình hóa phân phối thống kê chưa rõ của dữ liệu người nói. GMM trở thành một kỹ thuật hiện đại vào những năm 1990, cả khi **Maximum likelihood** ((Cực đại triển vọng) **Expectation-Maximization**, EM) hoặc huấn luyện phân loại (**Maximum Mutual Information**, MMI) còn được dùng. Tuy nhiên, việc sử dụng MAP tương thích với hầu hết phương tiện từ một **Universal Background Model** (UBM) đã mang lại cho GMMs một lợi thế lớn so với các kỹ thuật khác, đặc biệt khi được sử dụng với các kỹ thuật chuẩn hóa như **Z-standard** (chuẩn hóa điểm giả), **T-norm** (chuẩn hóa âm thanh), **H-norm** (chuẩn Z phụ thuộc vào thiết bị cầm tay), **HT-norm** (H + T-norm) hoặc **Feature Mapping** (xác định và chuẩn hóa kênh).

Discriminative techniques - Kỹ thuật phân tách như **Artificial Neural Networks**, đã được sử dụng trong nhiều năm, nhưng hiệu suất của chúng chưa bao giờ đạt đến hiệu suất của **GMM**. Tuy nhiên, vào cuối những năm 90, **SVM - Support Vector Machine** được ví như một bộ phân loại hiệu năng cao được huấn luyện sẵn, mang lại cho **GMM** một đối thủ cạnh tranh vì hiệu suất gần như tương đương bằng việc sử dụng **SVM** trong không gian nhiều chiều hơn với các kernel tích hợp như **GLDS (Generalized Linear Discriminant Sequence Kernel)**

Gần đây, việc sử dụng **SuperVectors**, một kỹ thuật hỗn hợp **GMM-SVM** coi là công cụ của **GMM** cho mọi trường hợp (cả trong huấn luyện và kiểm tra) là các điểm trong không gian nhiều chiều (số chiều bằng với số hỗn hợp của **GMM** nhân với số chiều của vector được tham số hóa) bằng cách sử dụng **SVM** cho mỗi người nói để phân loại các cách phát âm chưa biết từ siêu phẳng giọng nói được huấn luyện sẵn.

5.2 Idiolectal systems - Hệ thống Idiolectal

Note: (Idiolectal = idio (personal, private) + (dia)lect)

Hầu hết hệ thống nhận dạng người nói không phụ thuộc văn bản đều dựa vào đặc trưng phổ ngắn cho đến khi công trình của Doddington được công bố, nó mở ra một thế giới mới khả năng cải thiện các hệ thống nhận dạng người nói không phụ thuộc vào văn bản.

Doddington đã nhận ra và chứng minh rằng lời nói của những người nói khác nhau không chỉ khác nhau về âm học, mà còn về các đặc điểm khác như cách sử dụng từ. Đặc biệt, trong công việc của mình, ông đã lập mô hình cách sử dụng từ của từng người nói cụ thể bằng cách sử dụng **n-gram** mô hình hóa các chuỗi từ và xác suất của chúng và chứng minh rằng việc sử dụng các mô hình đó có thể cải thiện hiệu suất của hệ thống GMM âm thanh/ phổ cơ bản. Quan trọng hơn kết quả cụ thể này là thực tế là công trình này đã thúc đẩy nghiên cứu trong việc sử dụng các cấp độ thông tin cao hơn (idiolectal, phonotactic, prosodic, v.v.) để **nhận dạng giọng nói độc lập với văn bản**.

Các phần tiếp theo mô tả hai trong số những hệ thống thành công nhất khai thác mức độ thông tin cao hơn: hệ thống âm vị, cố gắng mô hình hóa các đặc điểm phát âm và hệ thống thuận âm, mô hình hóa các mẫu âm thanh chuyên biệt dành cho người nói.

5.3 Phonotactic systems - Hệ thống ngữ âm

Một hệ thống nhận dạng giọng nói phonotactic điển hình bao gồm hai khía cạnh:

- Bộ giải mã ngữ âm (the phonetic decoders), chuyển đổi lời nói thành một chuỗi các nhãn ngữ âm.
- Giai đoạn mô hình hóa ngôn ngữ thống kê n-gram, mô hình hóa tần số của ngữ âm và chuỗi ngữ âm cho mỗi người nói cụ thể.

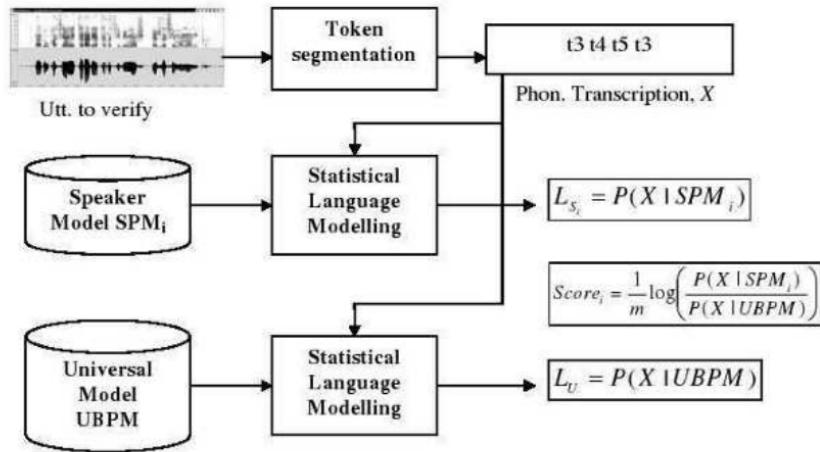


Fig. 8.3. Verification of an utterance against a speaker model in phonotactic speaker recognition

Hình 5: Quy trình của hệ thống ngữ âm

Bộ giải mã ngữ âm – thường dựa trên **Mô hình Markov ẩn (HMMs)** - có thể được lấy từ trình nhận dạng giọng nói có sẵn hoặc được huấn luyện đặc biệt. Đối với mục đích nhận dạng người nói, việc có bộ giải mã ngữ âm không quan trọng và thậm chí không quan trọng lắm phải có bộ giải mã ngữ âm trong ngôn ngữ của người nói để được nhận dạng. Thực tế có phần đáng ngạc nhiên này đã được phân tích cho thấy rằng các lỗi ngữ âm phụ thuộc vào giọng nói do bộ giải mã tạo ra dường như là của giọng nói cụ thể và do đó thông tin hữu ích cho việc nhận dạng giọng nói miễn là các lỗi này phù hợp với từng giọng nói cụ thể.

Sau khi có bộ giải mã ngữ âm, bản giải mã ngữ âm của nhiều câu từ nhiều giọng nói khác nhau có thể được sử dụng để huấn luyện **Mô hình Universal Background Phone (UBPM)** đại diện cho tất cả những giọng nói có thể có. Các **Mô hình Ngữ âm Giọng nói (SPMi - Speaker Phone Models)** được huấn luyện bằng cách sử dụng một số bộ giải mã ngữ âm của từng người nói cụ thể. Vì giọng nói có sẵn để huấn luyện một mô hình giọng nói thường bị hạn chế, các mô hình giọng nói được suy với **UBPM** để tăng tính mạnh mẽ trong ước tính tham số.

Sau khi các mô hình ngôn ngữ thống kê được huấn luyện, quy trình để xác minh cách phát âm một tập test so với mô hình giọng nói **SPMi** gồm các bước:

- Bước 01: Tạo ra giải mã ngữ âm của nó, **X**, giống như cách giải mã được sử dụng để huấn luyện **SPMi** và **UBPM**
- Bước 02: Giải mã ngữ âm của câu thử, **X** và các mô hình thống kê (**SPMi**, **UBPM**) được sử dụng để tính toán khả năng giải mã ngữ âm, **X**, dựa trên mô hình giọng nói **SPMi** và mô hình nền **UBPM**.

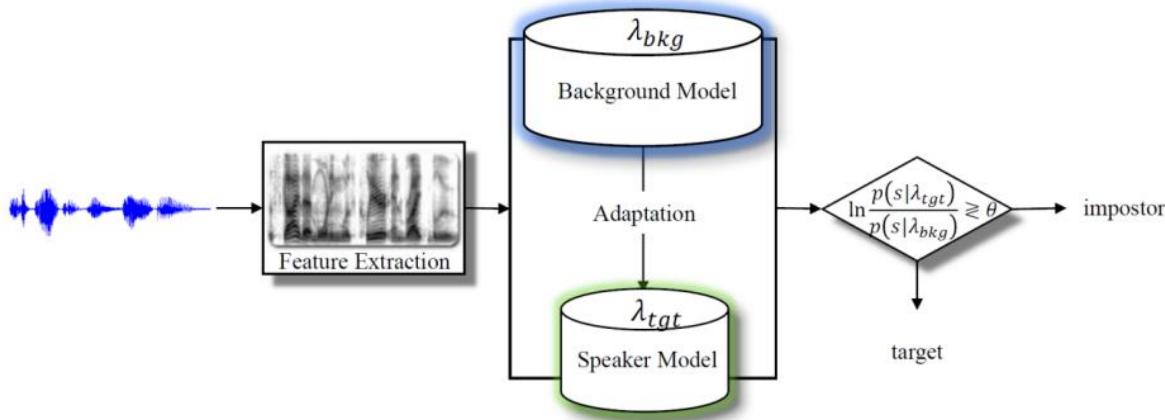
Điểm nhận dạng là logarithm tỷ lệ của cả hai khả năng xảy ra. Quá trình này, thường được mô tả là Nhận dạng ngữ âm, sau đó là Mô hình hóa ngôn ngữ (**PRLM**) có thể được lặp lại cho các bộ giải mã ngữ âm khác nhau (ví dụ: các ngôn ngữ hoặc độ phức tạp khác nhau) và các điểm nhận dạng khác nhau chỉ được thêm vào hoặc hợp nhất để có hiệu suất tốt hơn, mang lại một phương pháp được gọi là **PRLM** hoặc **PPRLM** song song.

Để cải thiện hiệu năng người ta sử dụng toàn bộ mạng lưới nhận dạng ngữ âm, nó là một đồ thị không chu trình có hướng chứa các giả thuyết có khả năng xảy ra nhất cùng với các xác suất của chúng. Có nhiều thông tin có nghĩa cho phép ước tính **n-gam** tốt hơn trên các tài nguyên giọng nói

hạn chế. Ngoài ra, hiệu năng còn có thể cải thiện bằng việc sử dụng **SVM** để phân lớp toàn bộ n-gram được huấn luyện với giả thuyết tốt nhất.

Tóm tắt các bước hoạt động của mô hình

- Bước 1: Đầu vào là giọng nói cần xác minh
- Bước 2 - Token segmentation: Tách thành các đoạn t3, t4, t5, t6
- Bước 3: Mô hình hóa ngôn ngữ thống kê n-gram
 - Huấn luyện mô hình Universal Background Phone (UBPM) $L_U = P(X|UBPM)$
 - Huấn luyện mô hình Speaker Phone Models (SPM_i): $L_{S_i} = P(X|SPM_i)$
- Bước 4: Tính recognition score: $Score_i = \frac{1}{m} \log \left(\frac{P(X|SPM_i)}{P(X|UBPM)} \right)$



Hình 6: Quy trình của hệ thống ngữ âm

5.4 Prosodic systems - Hệ thống ngữ điệu

Một trong những hệ thống ngữ điệu tiên phong và thành công nhất trong việc nhận dạng giọng nói không phụ thuộc vào văn bản là công trình của Adami. Hệ thống bao gồm hai khối xây dựng chính:

- Phân tách ngữ điệu, phân tích ưu điểm và biểu thị nó dưới dạng một chuỗi các nhãn hoặc token
- Mô hình hóa ngôn ngữ thống kê n-gram, mô hình hóa tần số của các tokens và trình tự của từng giọng nói cụ thể

Một số khả năng khác để mô hình hóa thông tin ngữ điệu cũng đã được chứng minh là khá thành công là việc sử dụng Non-uniform Extraction Region Features (NERFs) được phân định bằng khoảng dừng đủ dài hoặc NERF được xác định bởi cấu trúc âm tiết của câu (SNERFs).

Các tác giả đã triển khai một hệ thống ngữ điệu dựa trên công trình của Adami, trong đó khối thứ hai giống hệt nhau để nhận dạng ngữ âm và âm sắc chỉ với những điều chỉnh nhỏ để cải thiện hiệu suất. Quá trình tokenization bao gồm hai giai đoạn:

- Giai đoạn một, đối với mỗi đoạn giọng nói của đoạn thoại, quỹ đạo thời gian của các đặc điểm ngữ điệu (tần số cơ bản - hoặc cao độ- và năng lượng) được rút trích
- Giai đoạn hai, cả hai đường bao đều được phân đoạn và dán nhãn bằng định lượng trung bình độ dốc.

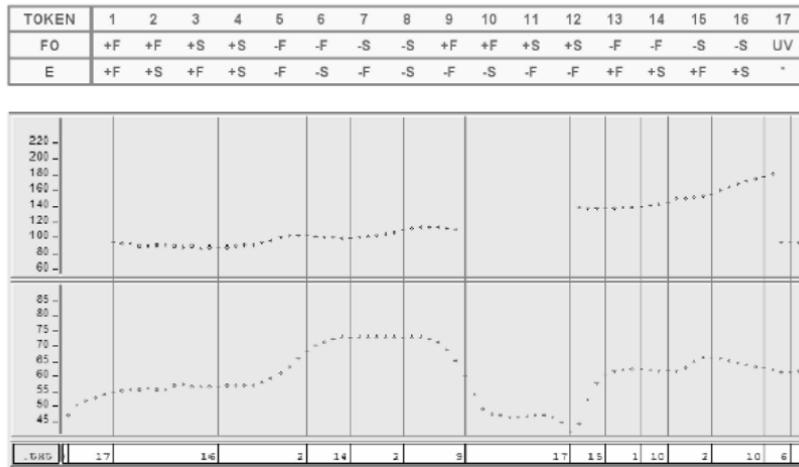


Fig. 8.4. Prosodic token alphabet (top table) and sample tokenization of pitch and energy contours (bottom figure).

Hình 7: Token ngữ điệu alphabet

Hình trên là một bảng chứa 17 token ngữ điệu. Một token đại diện cho các phân đoạn vô thanh, trong khi 16 token được sử dụng để đại diện cho các phân đoạn hữu thanh tùy thuộc vào độ dốc (phát nhanh, tăng chậm, giảm nhanh, giảm chậm) của năng lượng và cao độ.

5.5 Kho ngữ liệu và benchmarks

- Vào năm 1996, NIST bắt đầu Dánh giá Nhận dạng Giọng nói -Speaker Recognition Evaluations hàng năm, đây chắc chắn là động lực của những tiến bộ đáng kể.
- Các hội thảo sau đánh giá đã cho phép người tham gia chia sẻ kinh nghiệm, cải tiến, thất bại, v.v. của họ trong một môi trường hợp tác cao. Vai trò của LDC (Linguistic Data Consortium) cung cấp tài liệu nói mới đầy thách thức cũng rất đáng chú ý, vì nhu cầu liên tục tăng lên (cả về lượng lời nói và yêu cầu ghi âm)
- Các bộ đánh giá trước đây (phát triển, huấn luyện và kiểm tra âm thanh và chìa khóa giải pháp) có sẵn thông qua LDC - Linguistic Data Consortium để các nhà nghiên cứu mới đánh giá hệ thống của họ mà không có áp lực cạnh tranh

5.6 Công trình điển hình

Tên công trình: The ATVS multilevel text-independent system

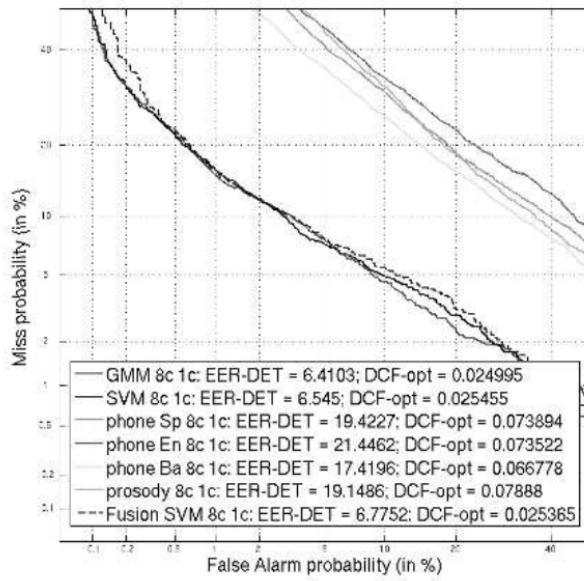


Fig. 8.5. Performance of ATVS subsystems in NIST'06 Speaker Recognition Evaluation comparing spectral (GMM and SVM), phonotactic and prosodic systems.

Hình 8: The ATVS multilevel text-independent system

Kết quả tại NIST SRE06 trong nhiệm vụ 8c1c (8 conversation for training và 1 conversation for testing), để xem hiệu suất của các hệ thống con khác nhau trên cùng một bài tập. Sự khác biệt chính của hệ thống ATVS năm 2006 so với hệ thống 2005 là việc sử dụng Ánh xạ đặc trưng trong cả GMM và SVM, việc sử dụng mở rộng đa thức bậc 3 (thay vì bậc 2) trong nhãn GLDS và việc sử dụng của một PRLM được huấn luyện với SpeechDat

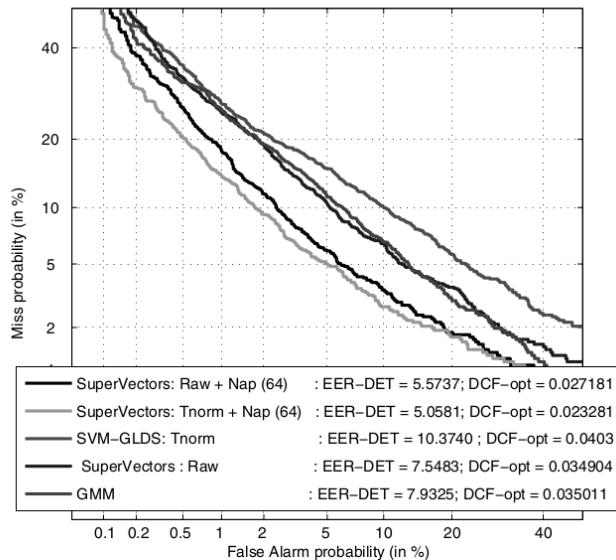


Fig. 8.6. Post-eval performance improvements over NIST'06 SRE ATVS system based on NAP channel compensation and SuperVector-GMMs (1c-1c male sub-task).

Hình 9: The ATVS multilevel text-independent system

D.2 Các phương pháp SOTA - State of The Art gần đây về Voice Biometrics

D.2.1 Giới thiệu

Những phương pháp đã tìm hiểu từ sách Handbook of Biometrics: Voice Biometrics đã cho chúng ta cái nhìn tổng quan về lĩnh vực Nhận dạng giọng nói và những phương pháp truyền thống (tạm gọi là thời kỳ trước Deep Learning) cùng với những thông tin các công trình nghiên cứu nổi bật. Tuy nhiên, gần đây với sự phát triển trong lý thuyết Học Máy cùng với những thành công đầy hy vọng của Học Sâu (Deep Learning) thì lĩnh vực Nhận dạng Giọng nói đã và đang tiếp tục phát triển.

D.2.2 Động lực nghiên cứu

Hai tác vụ lớn trong lĩnh vực Nhận dạng giọng nói (Speaker Recognition) là định danh giọng nói (Speaker Identification) và xác minh giọng nói (Speaker Verification) vẫn đã và đang phát triển, song song với sự phát triển của công nghệ.

Lĩnh vực nghiên cứu này có nhiều ứng dụng trong thực tế, áp dụng vào nhiều lĩnh vực khác như bảo mật, pháp y, xác thực sinh trắc học, nhận dạng người nói và phân cực người nói. Nhiều phương pháp SOTA (State of the Art) được đưa nhằm giải quyết bài toán nhận dạng giọng nói ngày càng tốt hơn, đặc biệt là sử dụng kỹ thuật Deep Learning.

Với mục đích mở rộng kiến thức về nó, trong phần trình bày này, nhóm xin phép trình bày những gì tìm hiểu được về những phương pháp nhận dạng giọng nói nổi bật gần đây (khoảng 2014 - đến nay)

D.2.3 Phát biểu bài toán

Tác vụ: Định danh người nói

- Đầu vào (Input): Dữ liệu âm thanh giọng nói

- Đầu ra (Output): Danh tính của người nói

Tác vụ: Xác nhận người nói

- Đầu vào (Input): Dữ liệu âm thanh giọng nói
- Đầu ra (Output): Đồng ý/ Từ chối

D.2.4 Một số kho ngữ liệu

Cũng giống như các lĩnh vực khác của Học Máy, câu hỏi quan trọng ngay từ ban đầu là dữ liệu. Trong phần này, nhóm giới thiệu một số cơ sở dữ liệu phục vụ cho việc huấn luyện các mô hình, cho các tác vụ như speaker identification, speaker verification, speaker recognition.

Kho ngữ liệu TIMIT (Garofolo et al., 1993)

- Tên đầy đủ: TIMIT Acoustic-Phonetic Continuous Speech Corpus
- Tác giả/ Nhóm tác giả: John S. Garofolo, Lori F. Lamel, William M. Fisher, Jonathan G. Fiscus, David S. Pallett, Nancy L. Dahlgren, Victor Zue
- Năm: 1993
- Ngôn ngữ: Tiếng Anh
- Mô tả sơ lược: TIMIT chứa các bản ghi âm băng thông rộng của 630 người nói tám phương ngữ chính của tiếng Anh Mỹ, mỗi người đọc mươi câu giàu ngữ âm. Kho tài liệu TIMIT bao gồm các phiên âm từ ngữ, ngữ âm và từ ngữ được căn chỉnh theo thời gian cũng như tệp dạng sóng giọng nói 16 bit, 16kHz cho mỗi câu nói

Kho ngữ liệu WSJ (Marcus et. al, 1993)

- Tên đầy đủ: CSR-I (WSJ0) Complete
- Tác giả/ Nhóm tác giả: John S. Garofolo, David Graff, Doug Paul, David Pallett
- Năm: 1993
- Ngôn ngữ: Tiếng Anh
- Mô tả sơ lược:

Kho ngữ liệu RSR2015 (Larcher et al., 2012)

- Tên đầy đủ: The RSR2015: Database for Text-Dependent Speaker Verification using Multiple Pass-Phrases
- Tác giả/ Nhóm tác giả: Anthony Larcher, Kong Aik Lee, Bin Ma, Haizhou Li
- Hội nghị: Annual Conference of the International Speech Communication association (Inter-speech), tại Portland, United States, năm 2012
- Mô tả sơ lược: Cơ sở dữ liệu RSR2015 chứa các bản ghi âm từ 300 người, 143 nữ và 157 diễn giả nam. Những người tham gia được chọn để trở thành đại diện cho sự phân bố dân tộc của người Singapore. Các diễn giả được chọn từ 17 đến 42 tuổi. Mỗi người trong số những người tham gia ghi lại chín phiên băng cách sử dụng ba thiết bị di động. Mỗi phiên bao gồm ba mươi câu ngắn.

Kho ngữ liệu Librispeech (Panayotov et.al, 2015)

- Tên đầy đủ: LibriSpeech ASR corpus
- Bài báo: LibriSpeech: an ASR corpus based on public domain audio books
- Tác giả/ Nhóm tác giả: Vassil Panayotov, Guoguo Chen, Daniel Povey and Sanjeev Khudanpur
- Được công bố tại hội nghị The international Conference on Acoustics, Speech, & Signal Processing (ICASSP), 2015
- Mô tả sơ lược: Kho ngữ liệu có kích thước khoảng 1000 giờ nói tiếng Anh với tần số 16kHz

Kho ngữ liệu VCTK (Veaux et al., 2017)

- Tên đầy đủ: SUPERSEDED - CSTR VCTK Corpus: English Multi-speaker Corpus for CSTR Voice Cloning Toolkit
- Tác giả/ Nhóm tác giả: Veaux Christophe, Yamagishi Junichi, MacDonald Kirsten
- Nhà xuất bản: University of Edinburgh. The Centre for Speech Technology Research (CSTR)
- Mô tả sơ lược: Tất cả dữ liệu giọng nói được ghi lại bằng cách sử dụng thiết lập ghi âm giống hệt nhau: micro đa hướng (DPA 4035) và micro tụ màng nhỏ với băng thông rất rộng (Sennheiser MKH 800), tần số lấy mẫu 96kHz ở 24 bit và trong một buồng phản xạ hemi của Đại học Edinburgh

Kho ngữ liệu VoxCeleb (Nagrani et. al, 2017)

- Tên đầy đủ: VoxCeleb: a large-scale speaker identification dataset
- Tác giả/ Nhóm tác giả: Arsha Nagrani, Joon Son Chung, Andrew Zisserman (Visual Geometry Group, Department of Engineering Science, University of Oxford, UK)
- Mô tả sơ lược: kho ngữ liệu chứa hơn 100.000 câu nói cho 1.251 người nói tiếng, được trích xuất từ các video tải lên YouTube. Bộ dữ liệu cân bằng về giới tính, với 55% người nói là nam giới. Những người nói thuộc các sắc tộc, giọng, nghề nghiệp và độ tuổi khác nhau. Quốc tịch và giới tính của mỗi người nói (lấy từ Wikipedia) cũng được cung cấp.

Kho ngữ liệu VoxCeleb2 (Chung et. al, 2018)

- Tên đầy đủ: VoxCeleb2: Deep Speaker Recognition
- Tác giả/ Nhóm tác giả: J. S. Chung*, A. Nagrani*, A. Zisserman
- Hội nghị: INTERSPEECH, 2018.
- Mô tả sơ lược: Là kho ngữ liệu giọng nói lớn nhất hiện tại, VoxCeleb2 chứa hơn 1 triệu câu nói cho 6.112 người nói tiếng, được trích xuất từ các video tải lên YouTube. Tập hợp phát triển của VoxCeleb2 không có sự trùng lặp với các đặc điểm nhận dạng trong tập dữ liệu VoxCeleb1 hoặc SITW.

D.2.5 Kiểm định mô hình nhận dạng giọng nói

Trong xác minh người nói, để đưa ra quyết định người ta đề xuất kiểm tra bằng cách sử dụng likelihood-ratio test.

D.2.5 Các độ đo trong nhận dạng giọng nói

Trong nhận dạng giọng nói (đặc biệt là tác vụ xác minh), có hai loại độ đo tính tương tự thường được sử dụng để tính toán xác suất nếu một quan sát thử nghiệm có phải từ người nói đích hay không. Hầu hết tất cả các phương pháp tiếp cận bằng Deep Learning mới đều sử dụng các độ đo này (trong các sơ đồ xác minh người nói): khoảng cách cosin của những vector (Cosine Distance) và PLDA (Phân tích phân tách tuyến tính theo xác suất - probabilistic linear discriminant analysis).

Khoảng cách Cosine - Cosine Distance Một trong những độ đo đơn giản mà lại hiệu quả đó là khoảng cách Cosine, việc xác minh người nói được xác nhận thông qua góc giữa hai vectors, nếu góc càng bé, nghĩa là hai giọng nói càng gần nhau và ngược lại, khi góc giữa chúng càng lớn, nghĩa là hai giọng nói càng xa nhau, khác nhau.

$$CDS(w_{\text{target}}, w_{\text{test}}) = \frac{w_{\text{target}} \cdot w_{\text{test}}}{\|w_{\text{target}}\| \cdot \|w_{\text{test}}\|}$$

Phân tích phân tách tuyến tính theo xác suất - probabilistic linear discriminant analysis
Cho tập hợp i-vectors d chiều có độ dài chuẩn đã được chuẩn hóa

$$X = \{x_{ij}; i = \overline{1..N}, j = \overline{1..N}\}$$

thu được từ N người nói huấn luyện (mỗi người nói có H_i i-vectors), i-vectors có thể được viết dưới dạng:

$$x_{ij} = \mu + Wz_i + \epsilon_{ij}$$

Với $x_{ij} \in \mathbb{R}^D$, $W \in \mathbb{R}^{M \times M}$, $z_i \in \mathbb{R}^M$, $\epsilon_{ij} \in \mathbb{R}^D$

Tập $Z = \{z_j, j = \overline{1..N}\}$ là các ẩn số

W là ma trận kích thước $D \times M$ gọi là ma trận tải dữ kiện

μ là trung bình tập X

z_i gọi là dữ kiện người nói

ϵ_{ij} nhiễu với phân bố Gaussian trung bình 0 và hiệp phương sai Σ

Cho một i-vector kiểm tra x_t và một người nói cần được kiểm tra có i-vector x_s , điểm LR được tính toán như sau :

$$S_{LR}(x_t, x_s) = \frac{P(x_s, x_t | \text{same speaker})}{P(x_s, x_t | \text{different speaker})}$$

$$\Leftrightarrow S_{LR}(x_t, x_s) = \frac{N([x_s^T x_t^T] | [\mu^T \mu^T], \hat{W} \hat{W}^T + \hat{\Sigma})}{N([x_s^T x_t^T] | [\mu^T \mu^T], \text{diag}\{WW^T + \Sigma, WW^T + \Sigma\})}$$

Trong đó:

- $\hat{W} = [W^T W^T]^T$
- $\hat{\Sigma} = \text{diag}\{\Sigma, \Sigma\}$

Sử dụng công thức chuẩn cho nghịch đảo của ma trận khối trích trong Petersen, K. B., Pedersen, M. S. (2008). The matrix cookbook. Technical University of Denmark, 7(15), 510, trích trong bài báo Ioffe, S. (2006). Probabilistic linear discriminant analysis. In: European Conference on Computer Vision. Springer, Berlin, Heidelberg, pp. 531-542. Thu được công thức:

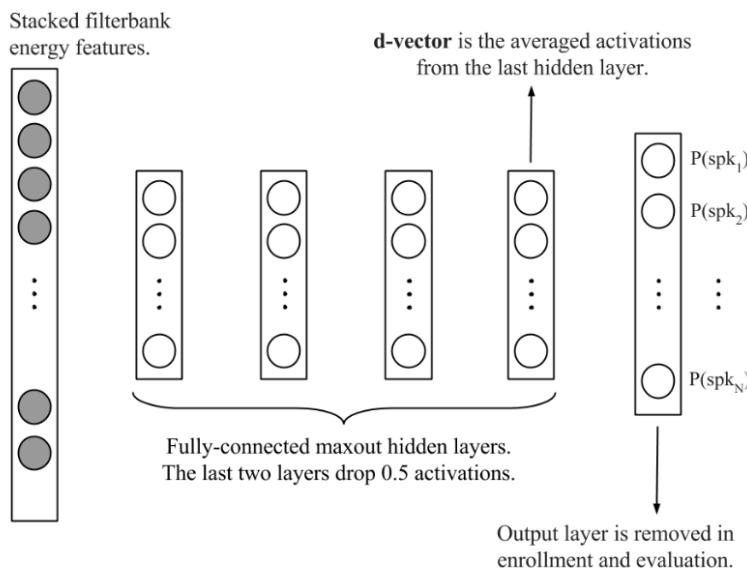
$$S_{LR}(x_s, x_t) = \text{const} + x_s^T Q x_s + x_t^T Q x_t + 2x_s^T P x_t$$

Trong đó:

- $P = \Lambda^{-1} \Gamma (\Lambda - \Gamma \Lambda^{-1} \Gamma)^{-1}; \Lambda = WW^T + \Sigma$
- $Q = \Lambda^{-1} (\Lambda - \Gamma \Lambda^{-1} \Gamma)^{-1}; \Gamma = WW^T$

D.2.5 Học Sâu trong tác vụ rút trích đặc trưng giọng nói

D.2.5.1 Về d-vectors



Hình 10: DNN model for speaker verification - Deep Neural Networks for small foot-print text-dependent speaker verification - 2014

Được giới thiệu trong bài báo "Deep neural networks for small footprint text-dependent speaker verification" **d-vectors** trở thành tiền đề cho hàng loạt các thành công sau này của lĩnh vực Nhận dạng giọng nói sử dụng Deep Learning.

Giới thiệu chung về bài báo:

- Bài báo: Deep neural networks for small footprint text-dependent speaker verification (Một bước nhỏ trong dùng mạng học sâu cho tác vụ xác minh người nói)
- Nhóm tác giả: Ehsan Variani (Johns Hopkins Univ., Baltimore, MD, USA), Xin Lei (Google Inc., USA), Erik McDermott (Google Inc., USA), Ignacio Lopez Moreno (Google Inc, Mountain View, CA, US), Javier Gonzalez-Dominguez (Google Inc., USA)
- Được publish tại hội nghị 2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), diễn ra tại Florence, Italy, vào năm 2014
- Từ khóa: Deep Neural Networks, Speaker Verification

Trong bài báo này, nhóm tác giả trình bày việc sử dụng Deep Neural Networks trong tác vụ xác minh người nói (Speaker Verification) để rút trích đặc trưng giọng nói và kích hoạt đặc trưng năm lớp ẩn cuối cùng của mạng học - gọi là d-vectors (Deep Vectors)

Trong giai đoạn đăng ký (Speaker Enrollment), mô hình DNN được huấn luyện sẵn được sử dụng để rút trích đặc trưng giọng nói của người nói ở lớp cuối cùng. Sau đó tính toán giá trị trung bình của những đặc trưng này, để cho ra d-vector của người nói

Trong giai đoạn đánh giá (Evaluation Stage), một giọng nói cần được xác minh khi vào mô hình sẽ được tính toán để cho ra một d-vector. Sau đó, dùng d-vector này so sánh với những d-vector trong quá trình đăng ký để đưa ra quyết định giọng nói của người nói này có phải là đúng với danh tính đưa ra không. Việc so sánh ở đây dùng độ đo cosine giữa hai vectors, việc đưa ra quyết định xác nhận sẽ dựa vào một ngưỡng (threshold)

Do điều kiện tài nguyên hạn chế, nhóm tác giả dùng maxout DNN sử dụng chiến lược dropout, chiến lược này giúp tránh việc over-fitting khi huấn luyện mô hình học trên tập dữ liệu huấn luyện nhỏ

Mô hình maxout DNN với 4 tầng ẩn (hidden layers) với 256 node cho mỗi tầng, không dùng DistBelief framework, kích thước pool mỗi lần là 2. Hai tầng đầu tiên không sử dụng dropout, hai tầng cuối sẽ drop 50% activations

Hàm Activation ở mỗi đơn vị neuron là một hàm phi tuyến, learning rate 0.001 với phân rã cấp số nhân (0.1 cho mỗi 5 triệu bước).

Đầu vào DNN được tạo ra bằng cách xếp chồng các đặc trưng được rút trích từ một khung bằng tập các bộ lọc 40 chiều. Số chiều của vector target sẽ tương ứng với số người nói tập huấn luyện, ở đây nhóm tác giả đưa ra là 496. Cuối cùng, mô hình DNN sẽ chứa khoảng 600k tham số, tương đương với một hệ thống i-vectors nhỏ nhất!

Phương pháp thực nghiệm Tập dữ liệu chứa 646 người nói, nói cùng một câu "ok google" nhiều lần trong nhiều cách khác nhau.

Sau đó, 496 người nói sẽ được chọn một cách ngẫu nhiên dùng làm tập huấn luyện, 150 người nói còn lại sẽ được dùng để đăng ký và đánh giá

Trong quá trình huấn luyện, số lần nói của mỗi người nói sẽ thay đổi từ 60 đến 130

Trong quá trình đăng ký người nói, 20 câu nói đầu tiên sẽ dành cho việc ghi danh tính và còn lại sẽ dùng cho việc đánh giá

Các kết quả

So sánh kết quả EER của hệ thống i-vectors so với các hệ thống khác như Gaussians, LDA cho thấy sự vượt trội hơn hẳn cho hệ thống i-vectors

Table 1. Comparison of EER results of *i*-vector systems with different number of UBM Gaussian components, *i*-vector and LDA output dimensions.

#Gaussians	<i>i</i> -vector Dim	LDA Dim	#Params	EER (raw)	EER (t-norm)
1024	300	200	12.2M	2.92%	2.29%
256	200	100	2.1M	3.11%	2.92%
128	100	100	540K	3.50%	2.83%

Hình 11: - Deep Neural Networks for small foot-print text-dependent speaker verification - 2014

Các kết quả EER của hệ thống xác nhận i-vectors và d-vectors với số câu nói đăng ký khác nhau. Chúng ta có thể thấy độ lỗi giảm dần khi cho số lượng câu nói tăng lên. Tuy nhiên, hệ thống d-vectors không có nhiều sự vượt trội hơn hẳn so với i-vectors.

Table 2. EER results of *i*-vector and *d*-vector verification systems using different number of utterances for enrollment.

	# utterances in enrollment			
	4	8	12	20
<i>i</i> -vector	2.83%	2.06%	1.64%	1.21%
<i>d</i> -vector	4.54%	3.21%	2.64%	2.00%

Hình 12: - Deep Neural Networks for small foot-print text-dependent speaker verification - 2014

Đồ thị bên dưới thể hiện đường cong DET giữa i-vector và d-vector sử dụng raw và t-norm. Ta có thể thấy, xác suất đồng ý sai của d-vector khá tốt so với i-vectors trong điều kiện dữ liệu sạch và nhiễu tương đối, tuy nhiên chưa có nhiều sự vượt trội đáng kể.

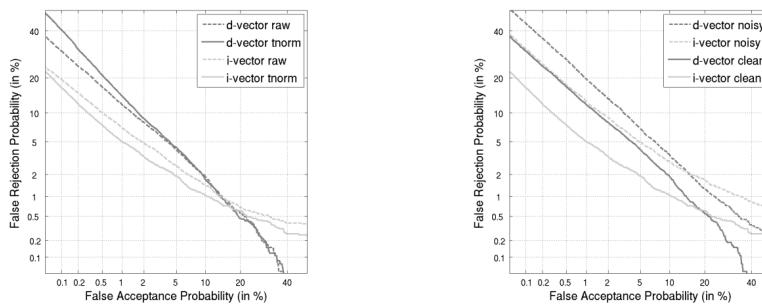


Fig. 2. Left: DET curve comparison between *i*-vector and *d*-vector speaker verification systems using raw and t-norm scores. Right: DET curve comparison of the two systems in clean and noisy conditions.

Hình 13: - Deep Neural Networks for small foot-print text-dependent speaker verification - 2014

Đồ thị bên dưới thể hiện đường cong DET giữa i-vector và d-vector sử dụng raw và t-norm. Ta có thể thấy, xác suất từ chối sai của d-vector khá tốt so với i-vectors trong điều kiện dữ liệu sạch và nhiễu tương đối, tuy nhiên chưa có nhiều sự vượt trội đáng kể.

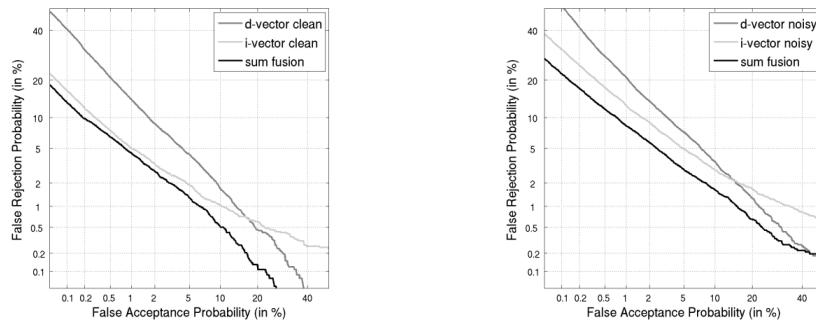


Fig. 3. DET curve for the sum fusion of the *i*-vector and *d*-vector systems in clean (left) and noisy (right) conditions.

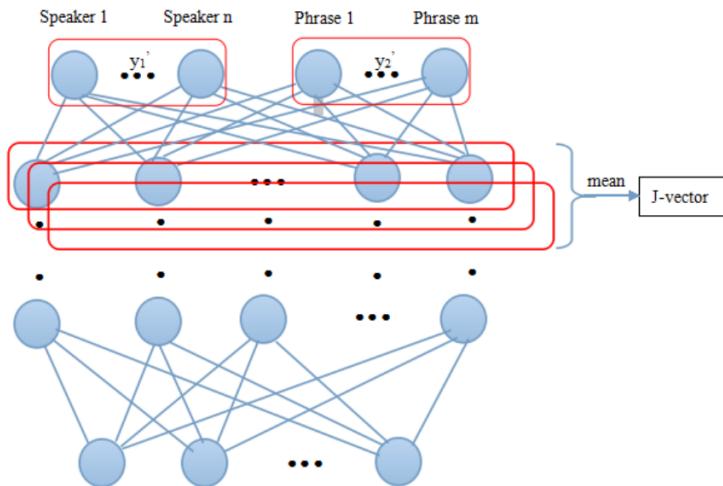
Hình 14: - Deep Neural Networks for small foot-print text-dependent speaker verification - 2014

Tuy vậy, d-vectors vẫn cho ta nhiều hy vọng, kết quả khá khả quan về việc sử dụng DNN trong xác minh người nói

Nhận xét chung

- Cơ sở của phương pháp có ý tưởng đơn giản là sử dụng Mạng Học Sâu trong việc rút trích đặc trưng và sử dụng lớp cuối cùng như một đặc trưng kích hoạt (deep vectors)
- Độ do dùng trong việc so sánh hai mô hình người nói đơn giản, so sánh cosine giữa hai vectors
- Kết quả có sự khả quan và nhiều hy vọng bằng chứng là xác suất đồng ý sai, từ chối sai thấp hơn hệ thống i-vectors trước đây
- EER vẫn chưa tốt so với hệ thống i-vectors

D.2.5.2 Về j-vectors



Hình 15: Multi-task DNN in Multi-Task Learning for Text-Dependent Speaker Verification, 2014.

Phương thức tiếp cận **d-vectors** được giới thiệu trong bài báo "Deep neural networks for small footprint text-dependent speaker verification" năm 2014 được mở rộng bằng phương thức tiếp cận học đa nhiệm (multi-task learning approach) và khái niệm về **j-vectors** được đưa ra, j-vectors có nghĩa là joint-vectors.

Giới thiệu chung về bài báo:

- Bài báo: Multi-Task Learning for Text-Dependent Speaker Verification
- Nhóm tác giả: Nanxin Chen, Yanmin Qian, Kai Yu (Shanghai Jiao Tong University, China)
- Được publish tại hội nghị INTERSPEECH 2015 16th Annual Conference of the International Speech Communication Association, diễn ra tại Dresden, Germany, từ ngày 6-10 tháng 9 năm 2015
- Từ khóa: Deep neural network, Multi-task learning (Học đa nhiệm), Speaker verification (Xác minh người nói), Discriminant analysis (Phân tích phân tách), Probabilistic linear discriminant analysis (Phân tích xác suất phân tách tuyến tính), Deep learning (Học Sâu)

Phương pháp Mô hình Học đa nhiệm Deep Neural Network sử dụng cho Xác minh người nói phụ thuộc văn bản được lấy ý tưởng từ việc sử dụng DNN với số lượng tham số cực lớn, một mô hình DNN có thể học cùng lúc việc phân tách văn bản, lẩn giọng nói

Hai hàm mất mát ban đầu là $C_1(y_1, y'_1), C_2(y_2, y'_2)$ được dùng để tạo thành hàm tổng mất mát:

$$C([y_1, y_2], [y'_1, y'_2]) = C_1(y_1, y'_1) + C_2(y_2, y'_2)$$

Trong đó:

- C_1, C_2 lần lượt là hai cross-entropy criteria cho giọng nói và văn bản
- y_1, y_2 đại diện cho nhãn đúng của từng người nói và văn bản
- y'_1, y'_2 đại diện cho nhãn đầu ra (nhận dự đoán được) của y_1, y_2 tương ứng

Mỗi khi tiến trình của multi-task neural network huấn luyện xong, lớp output sẽ bị loại bỏ, phần còn lại của neural network sẽ được dùng để rút trích biểu diễn kết hợp giữa giọng nói và văn bản (sử dụng lớp ẩn cuối cùng) sau đó lấy trung bình, gọi là **j-vectors** (hay **joint-vector**) điểm này khá tương tự với d-vectors

Có nhiều độ đo có thể sử dụng trong giai đoạn đăng ký, ở bài báo này, nhóm tác giả trình bày 2 cách thực hiện

- Joint Gaussian Discriminant Function

$$N(x|\mu_k, \Sigma_k) = \frac{1}{(2\pi)^{\frac{p}{2}} |\Sigma_k|^{\frac{1}{2}}} e^{-\frac{1}{2}(x-\mu_k)^T \Sigma^{-1} (x-\mu_k)}$$

Trong đó:

- Σ_k là hiệp phương sai lớp k
- μ_k là trung bình lớp k
- p là số chiều của vector

Mô hình LDA giả định rằng $\Sigma_k = \Sigma$, nhóm tác giả cho biết hàm discriminant function cho lớp thứ k như sau:

$$df_k(x) = -\frac{1}{2} \times (x - \mu_k)^T \Sigma^{-1} (x - \mu_k)$$

Và do trên trường hợp tập đóng, biểu thức có thể được rút gọn do giá trị $x^T \Sigma x$ có giá trị như nhau trên mọi lớp. Ta có một biểu thức tuyến tính (GDF)

$$df'_k = (\Sigma^{-1} \mu_k) x + \left(-\frac{1}{2} \mu_k^T \Sigma^{-1} \mu_k \right)$$

- Probabilistic Linear Discriminant Analysis

$$P(\mu_k) = N(\mu_k | m, \theta_b)$$

Các kết quả

Table 1: Performance of previous deep learning approaches

Feature	Classifier	EER(%)	minDCF
PLP Tandem	GMM-UBM	0.86	0.048
		0.69	0.037
d-vector	Cosine Sim.	21.05	0.818

Hình 16: Multi-task DNN in Multi-Task Learning for Text-Dependent Speaker Verification, 2014.

Table 2: Performance for different deep learning systems

Feature	Classifier	EER	minDCF
r-vector	Cosine Sim.	17.43	0.684
	Joint GDF	0.80	0.037
	Joint PLDA	1.47	0.065
d-vector	Cosine Sim.	21.05	0.818
	Joint GDF	0.71	0.033
	Joint PLDA	1.62	0.070
j-vector	Cosine Sim.	9.85	0.466
	Joint GDF	0.14	0.007
	Joint PLDA	0.54	0.027

Hình 17: Multi-task DNN in Multi-Task Learning for Text-Dependent Speaker Verification, 2014.

Table 3: Performance under unseen speakers conditions

Unseen Speakers Ratio		1/5		1/3	
Feature	Classifier	EER	minDCF	EER	minDCF
r-vector	Cosine Sim.	20.68	0.820	20.71	0.818
	Joint GDF	1.33	0.062	1.63	0.076
	Joint PLDA	1.42	0.066	1.65	0.073
d-vector	Cosine Sim.	15.75	0.644	15.75	0.654
	Joint GDF	1.43	0.063	1.78	0.079
	Joint PLDA	1.56	0.063	1.65	0.067
j-vector	Cosine Sim.	9.65	0.463	9.64	0.464
	Joint GDF	0.47	0.033	0.58	0.050
	Joint PLDA	0.50	0.022	0.50	0.024

Hình 18: Multi-task DNN in Multi-Task Learning for Text-Dependent Speaker Verification, 2014.

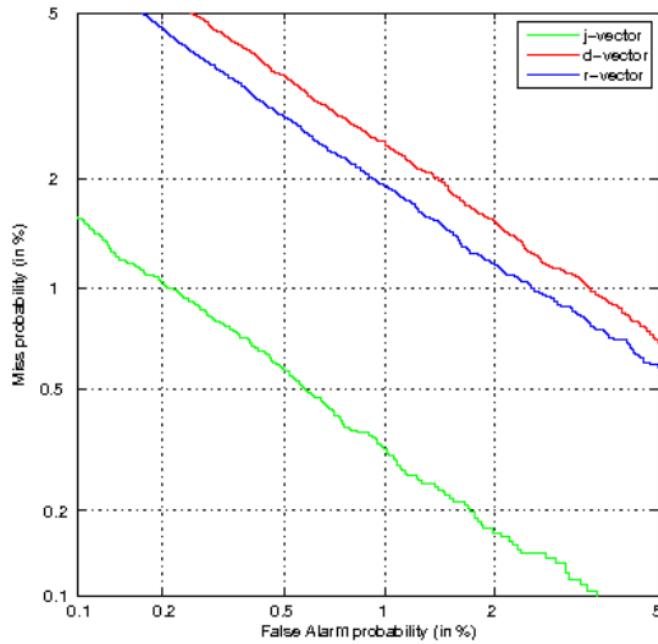


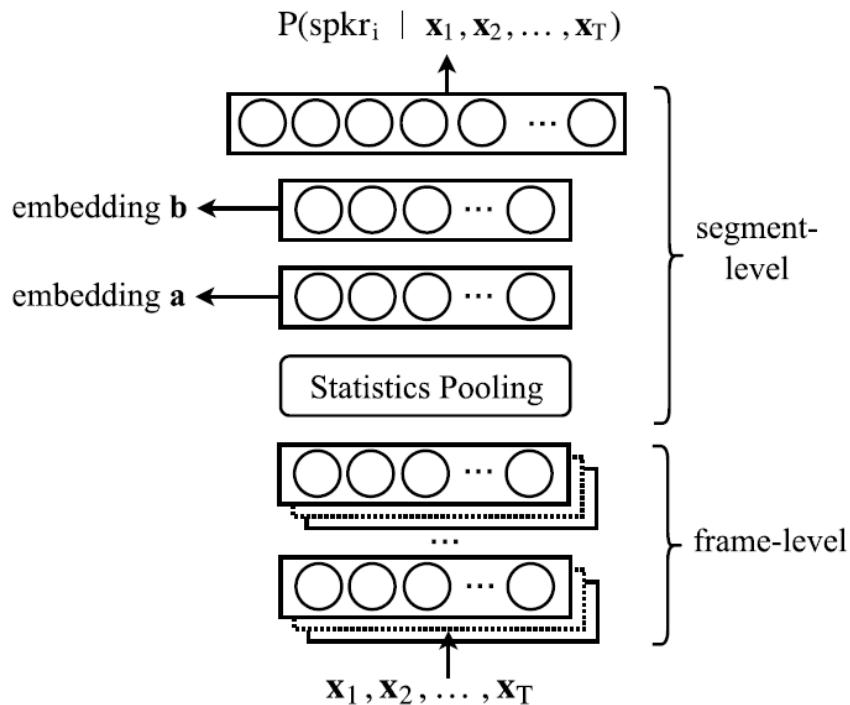
Figure 2: DET curves for RSR2015 Part I evaluation set. In all trials, target and impostor speaker pronoun the correct text

Hình 19: Multi-task DNN in Multi-Task Learning for Text-Dependent Speaker Verification, 2014.

Nhận xét

- Bài báo đề xuất một framework sử dụng kỹ thuật Deep Learning cho tác vụ xác minh người nói phụ thuộc văn bản có kết quả hết sức có triển vọng
- Áp dụng học đa nhiệm (giọng nói và văn bản), rút trích ra j-vectors
- Sử dụng Gaussian Discriminant Function và Probability Linear Discriminant Analysis, hai kỹ thuật mạnh trong việc phân lớp giọng nói
- Kết quả tốt hơn, cho độ lỗi thấp hơn rất nhiều so với d-vector, r-vector trên tác vụ xác minh người nói phụ thuộc văn bản

D.2.5.3 Về x-vectors

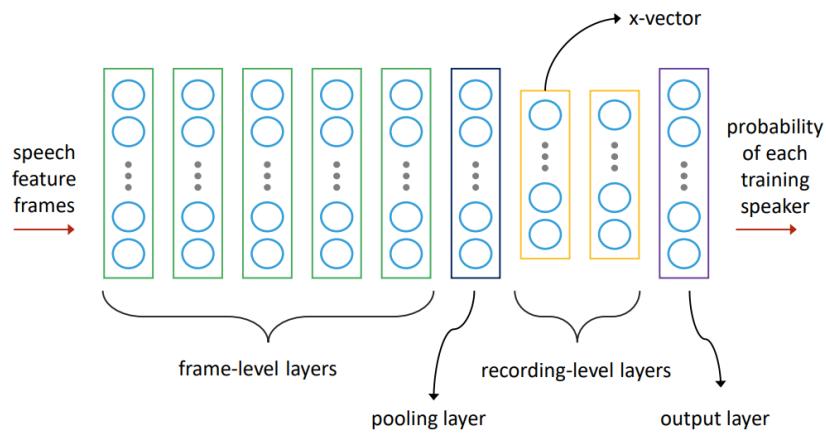


Hình 20: x-vector DNN embedding architecture in (Snyder et al., 2018)

Giới thiệu chung về bài báo:

- Bài báo: X-Vectors: Robust DNN Embeddings for Speaker Recognition
- Nhóm tác giả:
 - **David Snyder** - Center for Language and Speech Processing & Human Language Technology Center of Excellence, The Johns Hopkins University, Baltimore, MD, USA
 - **Daniel Garcia-Romero** - Center for Language and Speech Processing & Human Language Technology Center of Excellence, The Johns Hopkins University, Baltimore, MD, USA
 - **Gregory Sell** - Center for Language and Speech Processing & Human Language Technology Center of Excellence, The Johns Hopkins University, Baltimore, MD, USA
 - **Daniel Povey** - Center for Language and Speech Processing & Human Language Technology Center of Excellence, The Johns Hopkins University, Baltimore, MD, USA
 - **Sanjeev Khudanpur** - Center for Language and Speech Processing & Human Language Technology Center of Excellence, The Johns Hopkins University, Baltimore, MD, USA
- Được xuất bản tại hội nghị 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), diễn ra tại Calgary, AB, Canada, năm 2018
- Từ khóa: Speaker Recognition, Deep Neural Networks, Data Augmentation, x-vectors

Phương pháp



Hình 21: x-vectors DNN - From i-vectors to x-vectors – a generational change in speaker recognition illustrated on the NFI-FRIDA database - OxfordWave Research

Với 5 layers đầu tiên thực hiện tính toán trên các khung giọng nói với một ngữ cảnh thời gian nhỏ tập trung vào khung hiện tại, gọi là t . Ví dụ, đầu vào của frame2 là đầu ra của frame1, tại các frames $t-2$ và $t+2$, thì tổng ngữ cảnh thời gian ở frame2 là 9

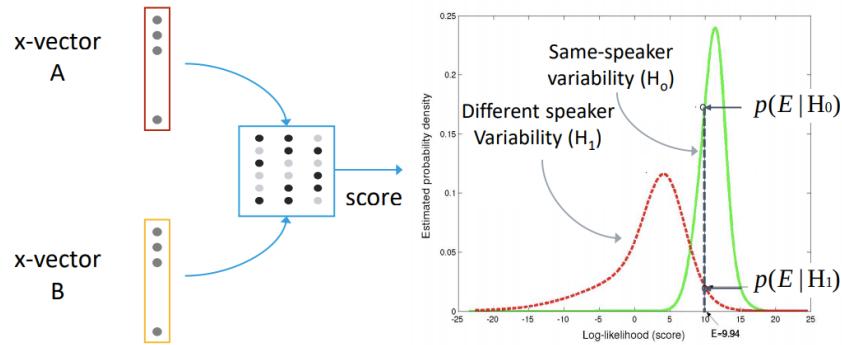
Statistics pooling layer (lớp tổng hợp thống kê) tổng hợp tất cả các đầu ra ở mức khung T từ lớp frame5 và tính toán giá trị trung bình và độ lệch chuẩn của nó. Hai giá trị gồm giá trị trung bình và độ lệch chuẩn được kết lại, được đưa qua các lớp phân đoạn và cuối cùng qua lớp đầu ra softmax

x-vectors sẽ được rút trích tại lớp phân đoạn 6 (segment6), ngay sau khi qua lớp tổng hợp thống kê

Layer	Layer context	Total context	Input x output
frame1	$[t-2, t+2]$	5	120x512
frame2	$\{t-2, t, t+2\}$	9	1536x512
frame3	$\{t-3, t, t+3\}$	15	1536x512
frame4	$\{t\}$	15	512x512
frame5	$\{t\}$	15	512x1500
stats pooling	$[0, T)$	T	$1500T \times 3000$
segment6	$\{0\}$	T	3000x512
segment7	$\{0\}$	T	512x512
softmax	$\{0\}$	T	512x N

Hình 22: Configuration Table of x-vector DNN embedding architecture in (Snyder et al., 2018)

Việc so sánh hai x-vector được mô hình như sau: Sử phân phối same-speaker và different speaker để ước lượng likelihood score để đưa ra so sánh hai vectors



Hình 23: x-vectors DNN - From i-vectors to x-vectors – a generational change in speaker recognition illustrated on the NFI-FRIDA database - OxfordWave Research

Các kết quả

		SITW Core			SRE16 Cantonese		
		EER(%)	$DCF10^{-2}$	$DCF10^{-3}$	EER(%)	$DCF10^{-2}$	$DCF10^{-3}$
4.1	Original systems	i-vector (acoustic)	9.29	0.621	0.785	9.23	0.568
		i-vector (BNF)	9.10	0.558	0.719	9.68	0.574
		x-vector	9.40	0.632	0.790	8.00	0.491
4.2	PLDA aug.	i-vector (acoustic)	8.64	0.588	0.755	8.92	0.544
		i-vector (BNF)	8.00	0.514	0.689	8.82	0.532
		x-vector	7.56	0.586	0.746	7.45	0.463
4.3	Extractor aug.	i-vector (acoustic)	8.89	0.626	0.790	9.20	0.575
		i-vector (BNF)	7.27	0.533	0.730	8.89	0.569
		x-vector	7.19	0.535	0.719	6.29	0.428
4.4	PLDA and extractor aug.	i-vector (acoustic)	8.04	0.578	0.752	8.95	0.555
		i-vector (BNF)	6.49	0.492	0.690	8.29	0.534
		x-vector	6.00	0.488	0.677	5.86	0.410
4.5	Incl. VoxCeleb	i-vector (acoustic)	7.45	0.552	0.723	9.23	0.557
		i-vector (BNF)	6.09	0.472	0.660	8.12	0.523
		x-vector	4.16	0.393	0.606	5.71	0.399

Table 2. Results using data augmentation in various systems. “Extractor” refers to either the UBM/T or the embedding DNN. For each experiment, the best results are **boldface**.

Hình 24: x-vector DNN embedding architecture in (Snyder et al., 2018)

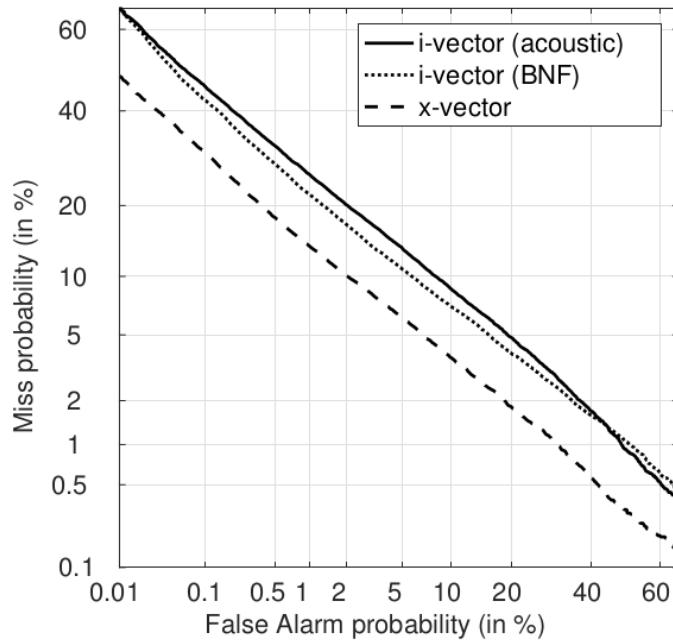


Fig. 1. DET curve for the Cantonese portion of NIST SRE16 using Section 4.5 systems.

Hình 25: x-vector DNN embedding architecture in (Snyder et al., 2018)

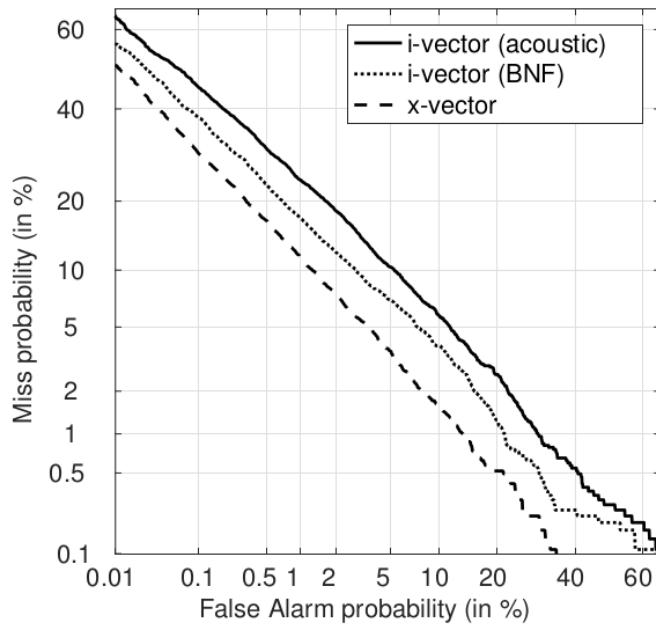


Fig. 2. DET curve for the SITW Core using Section 4.5 systems.

Hình 26: x-vector DNN embedding architecture in (Snyder et al., 2018)

Nhận xét

- Hiệu suất của x-vectors đã được chứng minh là tốt hơn đáng kể so với i-vectors, đặc biệt là ở khoảng thời gian ngắn.
- Học có giám sát, x-vectors DNN được huấn luyện bằng cách dùng dữ liệu có nhãn người nói
- DNN x-vector có khả năng khai thác lượng dữ liệu huấn luyện lớn hơn so với khung i-vector, nó bao hòa sau một lượng dữ liệu huấn luyện nhất định.
- khả năng khai thác lượng dữ liệu huấn luyện lớn hơn cũng tạo điều kiện cho một phương pháp thúc đẩy số lượng và tính đa dạng của dữ liệu huấn luyện được gọi là tăng cường dữ liệu. Quá trình này thêm tiếng ồn và nhiễu vào các mẫu huấn luyện và đưa chúng vào huấn luyện cùng với các mẫu ban đầu.
- Khả năng sử dụng cùng front-end (trích xuất đặc trưng) và back-end (so sánh vector) cho cả hệ thống i-vector và x-vector tạo điều kiện tích hợp hệ thống và cho phép so sánh trực tiếp hơn giữa hai phương pháp mô hình hóa.

D.2.5.4 So sánh d-vectors, j-vectors và x-vectors

	d-vectors	j-vectors	x-vector
Kỹ thuật rút trích	DNN	DNN	DNN
Vị trí rút trích	Tại lớp ẩn cuối cùng DNN	Tại lớp ẩn cuối cùng DNN	Sau khi qua lớp statistics pooling
Cách rút trích	Là trung bình kích hoạt tại lớp ẩn cuối cùng	Là trung bình kích hoạt tại lớp ẩn cuối cùng, kết hợp tín hiệu giọng nói và dữ liệu văn bản	Là vector phân đoạn (segment6) sau khi tính toán thống kê

D.2.5 Học Sâu trong tác vụ phân lớp giọng nói

D.2.5.1 Multi-domain features

Giới thiệu chung về bài báo:

- Tên bài báo: Multi-task Recurrent Model for Speech and Speaker Recognition
- Nhóm tác giả: Zhiyuan Tang (1) (2) , Lantian Li (1) and Dong Wang (1)
 - (1): Center for Speech and Language Technologies, Division of Technical Innovation and Development, Tsinghua National Laboratory for Information Science and Technology Center for Speech and Language Technologies, Research Institute of Information Technology, Tsinghua University
 - (2): Chengdu Institute of Computer Applications, Chinese Academy of Sciences
- Bài báo được đăng trên arXiv.org, vào ngày 31, tháng 3 năm 2016 (Phiên bản mới nhất được cập nhật vào ngày 27 tháng 9 năm 2016)
- Từ khóa: Multi-task, Recurrent Model, Speaker Recognition,

Phương pháp

Trong bài báo này, nhóm tác giả đề xuất một kiến trúc lặp lại mới có thể sử dụng để học các vụ có tính tương quan. Ý tưởng cơ bản của nó là sử dụng đầu ra của một tác vụ như một đầu vào vào của những tác vụ khác (khá tương tự với kiến trúc RNN thông thường). Kết quả đầu ra của một tác vụ ở bước thời gian trước đó $t - 1$ được sử dụng để cung cấp cho một tác vụ ở thời điểm t hiện tại.

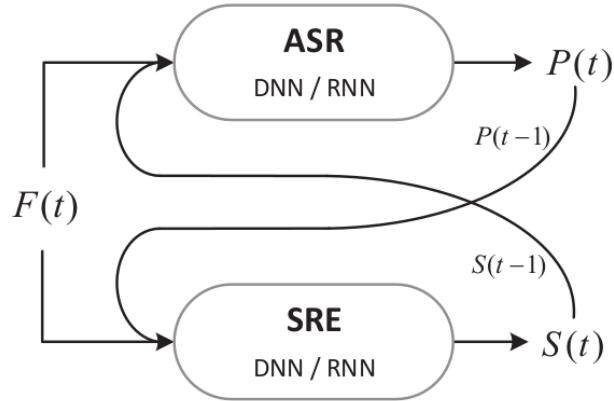


Fig. 1. Multi-task recurrent learning for ASR and SRE. $F(t)$ denotes primary features (e.g., Fbanks), $P(t)$ denotes phone identities (e.g., phone posteriors, high-level representations for phones), $S(t)$ denotes speaker identities (e.g., speaker posteriors, high-level representations for speakers).

Single-task model

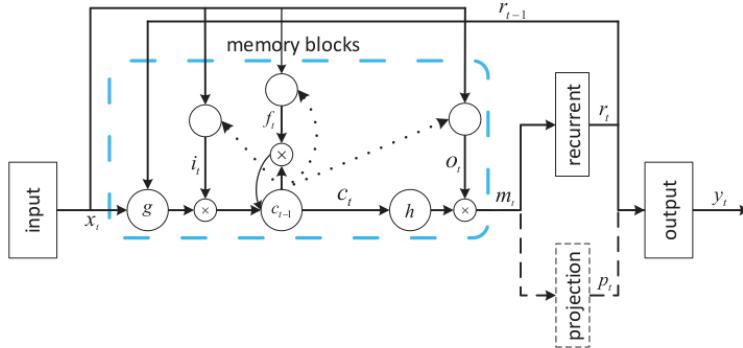


Fig. 2. Basic recurrent LSTM model for ASR and SRE single-task baselines. The picture is reproduced from [11].

$$\begin{aligned}
 i_t &= \sigma(W_{ix}x_t + W_{ir}r_{t-1} + W_{ic}c_{t-1} + b_i) \\
 f_t &= \sigma(W_{fx}x_t + W_{fr}r_{t-1} + W_{fc}c_{t-1} + b_f) \\
 c_t &= f_t \odot c_{t-1} + i_t \odot g(W_{cx}x_t + W_{cr}r_{t-1} + b_c) \\
 o_t &= \sigma(W_{ox}x_t + W_{or}r_{t-1} + W_{oc}c_t + b_o) \\
 m_t &= o_t \odot h(c_t) \\
 r_t &= W_{rm}m_t \\
 p_t &= W_{pm}m_t \\
 y_t &= W_{yr}r_t + W_{yp}p_t + b_y
 \end{aligned}$$

Multi-task model

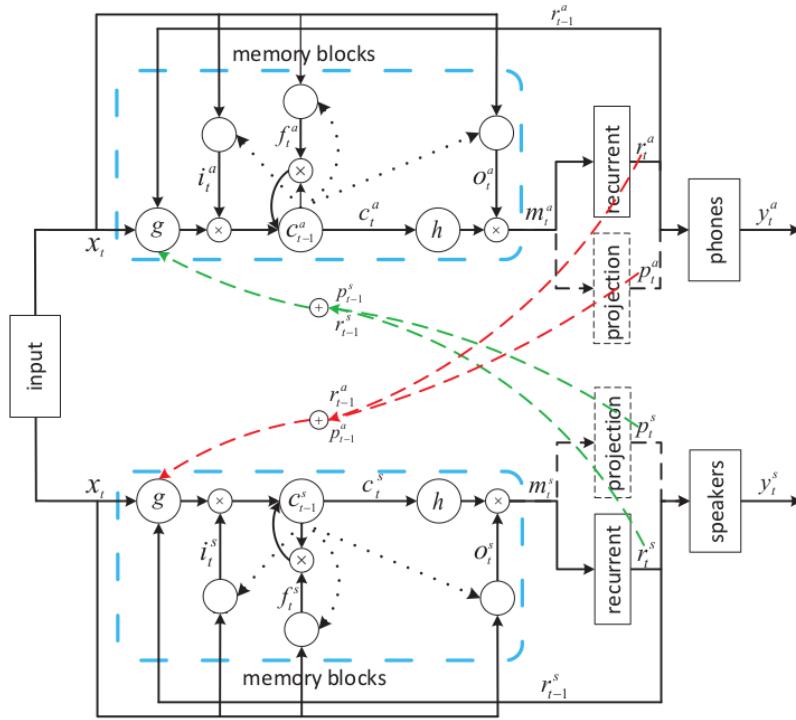


Fig. 3. Multi-task recurrent model for ASR and SRE, an example.

Speech recognition (ASR)

$$\begin{aligned}
 i_t^\alpha &= \sigma(W_{ix}^\alpha x_t + W_{ir}^\alpha r_{t-1}^\alpha + W_{ic}^\alpha c_{t-1}^\alpha + b_i^\alpha) \\
 f_t^\alpha &= \sigma(W_{fx}^\alpha x_t + W_{fr}^\alpha r_{t-1}^\alpha + W_{fc}^\alpha c_{t-1}^\alpha + b_f) \\
 g_t^\alpha &= g(W_{cx}^\alpha x_t^\alpha + W_{cr}^\alpha r_{t-1}^\alpha + b_c^\alpha + W_{cr}^{\alpha s} r_{t-1}^s + W_{cp}^{\alpha s} p_{t-1}^s) \\
 c_t^\alpha &= f_t \odot c_{t-1} + i_t \odot g(W_{cx}^\alpha x_t + W_{cr}^\alpha r_{t-1}^\alpha + b_c^\alpha) \\
 o_t^\alpha &= \sigma(W_{ox}^\alpha x_t^\alpha + W_{or}^\alpha r_{t-1}^\alpha + W_{oc}^\alpha c_t^\alpha + b_o^\alpha) \\
 m_t^\alpha &= o_t^\alpha \odot h(c_t^\alpha) \\
 r_t^\alpha &= W_{rm}^\alpha m_t^\alpha \\
 p_t^\alpha &= W_{pm}^\alpha m_t^\alpha \\
 y_t^\alpha &= W_{yr}^\alpha r_t^\alpha + W_{yp}^\alpha p_t^\alpha + b_y^\alpha
 \end{aligned}$$

Speaker recognition (SRE)

$$\begin{aligned}
 i_t^s &= \sigma(W_{ix}^s x_t + W_{ir}^s r_{t-1}^s + W_{ic}^s c_{t-1}^s + b_i^s) \\
 f_t^s &= \sigma(W_{fx}^s x_t + W_{fr}^s r_{t-1}^s + W_{fc}^s c_{t-1}^s + b_f) \\
 g_t^s &= g(W_{cx}^s x_t^s + W_{cr}^s r_{t-1}^s + b_c^s + W_{cr}^{s \alpha} r_{t-1}^\alpha + W_{cp}^{s \alpha} p_{t-1}^\alpha) \\
 c_t^s &= f_t \odot c_{t-1} + i_t \odot g(W_{cx}^s x_t + W_{cr}^s r_{t-1}^s + b_c^s) \\
 o_t^s &= \sigma(W_{ox}^s x_t^s + W_{or}^s r_{t-1}^s + W_{oc}^s c_t^s + b_o^s) \\
 m_t^s &= o_t^s \odot h(c_t^s) \\
 r_t^s &= W_{rm}^s m_t^s \\
 p_t^s &= W_{pm}^s m_t^s \\
 y_t^s &= W_{yr}^s r_t^s + W_{yp}^s p_t^s + b_y^s
 \end{aligned}$$

Các kết quả Các quả mà nhóm tác giả thực nghiệm có thấy nhiều kết quả đáng mong đợi và hết sức triển vọng.

- **Dữ liệu:** Nhóm tác giả sử dụng cơ sở dữ liệu WSJ, có dán nhãn cả từ lẫn người nói
 - Bộ dữ liệu huấn luyện: 90% bộ dữ liệu là dữ liệu giọng nói được chọn ngẫu nhiên từ tập huấn luyện si284, 10% còn lại được sử dụng cho việc kiểm tra nhận dạng người nói. Bao gồm 282 người nói và 33,587 lần nói với 40-144 lần nói trên mỗi người, được sử dụng để đào tạo hai hệ thống đơn tác vụ dựa trên LSTM, hệ thống cơ sở i-vector SRE và hệ thống đa nhiệm vụ được đề xuất.
 - Bộ dữ liệu kiểm tra: gồm 3 bộ datasets (dev93, eval192, eval193), thực hiện trên hai tác vụ ASR và SRE. Với tác vụ SRE, đánh giá bao gồm 21,350 target trials và 528,326 non-target trials
- **ASR baseline**

TABLE I
ASR BASELINE RESULTS.

	dev92	eval92	eval93	Total
WER%	8.36	5.14	8.06	7.41

- **SRE baseline**

TABLE II
SRE BASELINE RESULTS.

System	EER%		
	Cosine	LDA	PLDA
i-vector (200)	2.89	1.03	0.57
r-vector (256)	1.84	1.34	3.18

- **Multi-task joint training**

TABLE III
JOINT TRAINING RESULTS.

Feedback Info.	Feedback Input					ASR WER%	SRE EER%
<i>r</i>	<i>i</i>	<i>f</i>	<i>o</i>	<i>g</i>		7.41	1.84
✓	✓					7.05	0.62
✓	✓	✓				6.97	0.64
✓		✓				7.12	0.66
✓	✓		✓			7.24	0.65
✓			✓			7.26	0.65
✓	✓			✓		7.28	0.59
✓				✓		7.11	0.62
✓	✓			✓		7.11	0.67
✓	✓	✓	✓	✓		7.06	0.66
✓	✓	✓	✓	✓		7.23	0.71
✓	✓	✓	✓	✓	✓	7.05	0.55
✓	✓	✓	✓	✓	✓	7.23	0.62

Nhận xét

- Nhóm tác giả giới thiệu một kiến trúc mạng học lặp lại có thể huấn luyện cùng lúc nhiều tác vụ, có tính tương quan nghịch
- Kết quả trên cơ sở dữ liệu cho thấy nhiều triển vọng cho kiến trúc nói riêng, và những hướng tiếp cận đa nhiệm nói chung, nó đã chứng minh rằng phương pháp được trình bày có thể học các mô hình giọng nói và diễn giả đồng thời và cải thiện hiệu suất trên cả hai tác vụ

D.2.5.1 SincNet

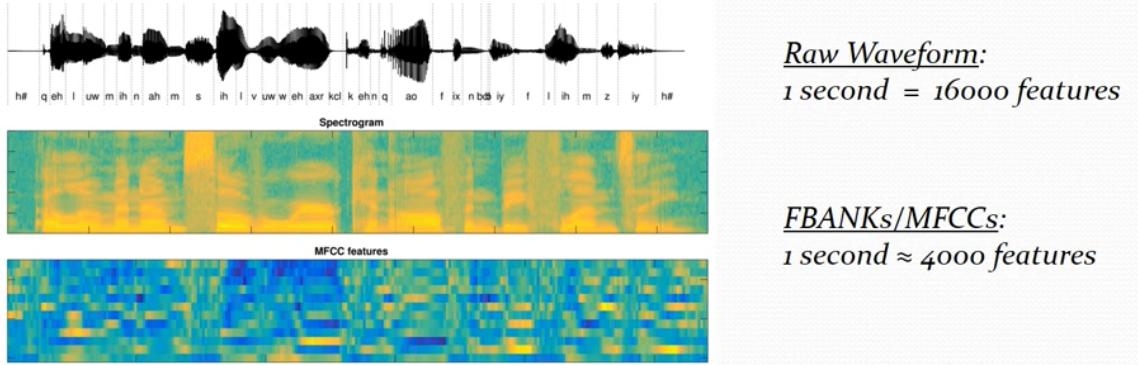
Giới thiệu chung về bài báo

- Nhóm tác giả trong bài báo này đề xuất một kiến trúc mạng CNN (Convolutional Neural Networks) mới, được gọi là SincNet, khai phá lớp tích chập đầu tiên để khám phá nhiều thông tin hơn. SincNet dựa trên các hàm *sinc* được tham số hóa, để cài đặt các bộ lọc băng thông.
- Ngược lại với CNNs chuẩn, học tất cả các phần tử của mỗi bộ lọc filter, ở đây, chúng ta chỉ có các tần số cắt thấp và cao học trực tiếp dữ liệu với phương pháp đề xuất
- Cung cấp một tập các bộ lọc mà chúng nhỏ gọn và hiệu quả trong việc tùy chỉnh với ứng dụng mà chúng ta muốn.
- Một sự kết hợp tuyệt vời giữa hai lĩnh vực khoa học lớn: Học máy (Machine Learning) và Xử lý Tín hiệu số (Digital Signal Processing).
- Bài báo được đăng công khai trên arXiv dot org lần đầu tiên vào năm 2018 bởi hai người Mirco Ravanelli, Yoshua Bengio (ông được xem là một trong 3 vị cha đẻ của phương pháp Deep Learning hiện đại), phiên bản cập nhật gần đây nhất là vào năm 2019 bằng việc thay thế hàm "sinc_conv" bằng "SincConv_fast" giúp tăng tốc độ lên 50% so với phiên bản cũ.
- Từ khóa: Speaker recognition, Convolutional Neural Networks, Raw samples

Vấn đề khi xử lý tín hiệu giọng nói: Dữ liệu đầu vào có số chiều cao

Convolutional Neural Networks - CNNs là một lựa chọn thích hợp với đầu vào là những sóng thô, nó kiến trúc phổ biến nhất để xử lý các mẫu giọng nói thô nhờ vào chia sẻ trọng số, bộ lọc cục bộ và tổng hợp giúp khám phá các biểu diễn dữ liệu và bắt biến. Vấn đề lớn nhất đối với sóng thô dựa trên mạng CNNs chính là **lớp tích chập đầu tiên**.

- Speech/Audio sequences are very high-dimensional.



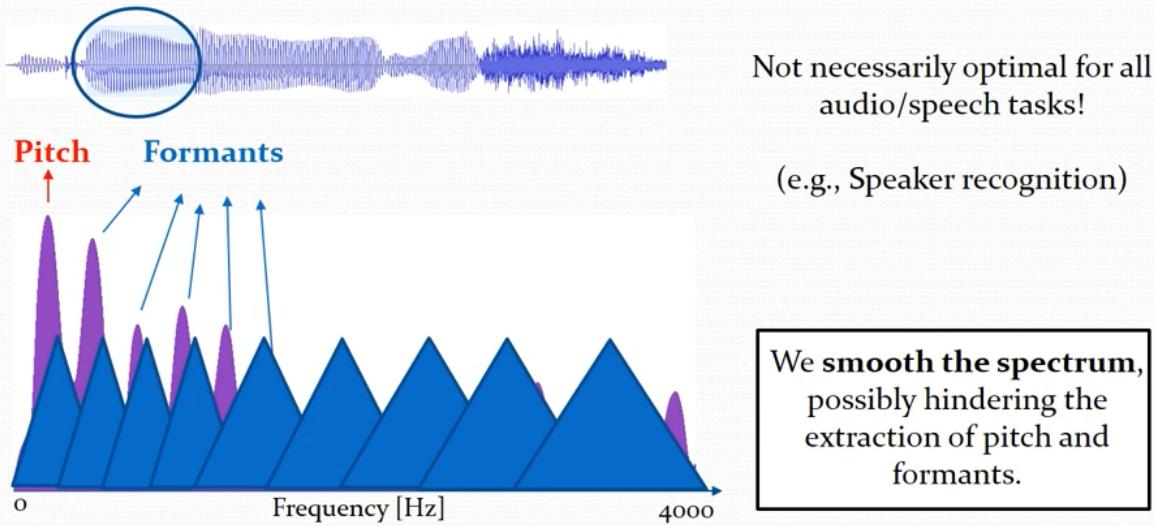
Hình 27: Minh họa chuỗi giọng nói có số chiều rất lớn. Ảnh được lấy từ video A brief introduction to SincNet thực hiện bởi Giáo sư Mirco Ravanelli

Ở lớp này, dữ liệu đầu vào là những chuỗi Speech/Audio có số chiều rất cao, ví dụ như: cứ mỗi giây thì ta lại có đến 16000 đặc trưng. Bằng những kỹ thuật thủ công ngày xưa như FBANKs hay MFCCs

thì ta có thể giúp nó giảm xuống còn 4000 đặc trưng mỗi giây, nhưng như thế vẫn còn rất nhiều!

Vấn đề khi xử lý tín hiệu giọng nói: **Những thông tin nhận dạng (đặc trưng giọng nói) dễ bị mất đi.** Những đặc trưng trong phổ tần số có thể nhận ra bằng mắt thường nhưng lại bị mất đi nếu chúng ta làm mịn chúng bằng những kỹ thuật thủ công như FBANKs hay MFCCs.

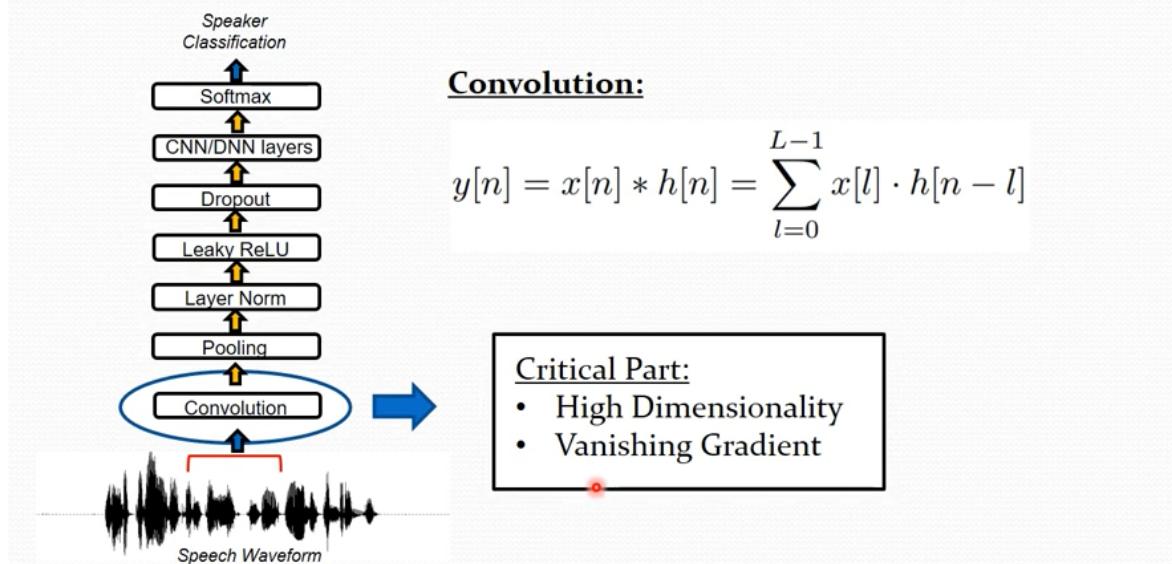
- *Hand-crafted features are designed from perceptual evidence.*



Hình 28: Ảnh được lấy từ video A bref introduction to SincNet thực hiện bởi Giáo sư Mirco Ravanelli

Vấn đề khi xử lý tín hiệu giọng nói: **Vấn đề mất dốc đạo hàm - Một vấn đề thường gặp khi làm việc với Mạng học Sâu.** Không những gặp vấn đề về số chiều dữ liệu mà còn bị ảnh hưởng nhiều hơn bởi các vấn đề về sự biến mất độ dốc đạo hàm, đặc biệt là khi sử dụng các kiến trúc rất sâu.

- Recent works have proposed directly feeding CNNs with raw waveforms.



Hình 29: Ảnh được lấy từ video A bref introduction to SincNet thực hiện bởi Giáo sư Mirco Ravanelli

Vanishing/exploding gradient problem in Deep Neural Networks

- Complex deep learning problems like image processing require multi-layer deep neural networks. One major issue with deep neural nets is the vanishing gradients problem.
- Unfortunately, gradients often get smaller and smaller as the algorithm progresses down to the lower layers. As a result, the Gradient Descent update leaves the lower layer weights virtually unchanged, and training never converges to a good solution. This is called vanishing gradients problem. Sometimes, the reverse also happens, the gradients get bigger and the algorithm diverges. This is the exploding gradients problem.
- Consider the sigmoid graph, the most common activation function used. When the input is highly positive or negative, the response saturates and gradient is almost 0 at these points.
- No gradient means no convergence. In other words, the model cannot find ideal weights.

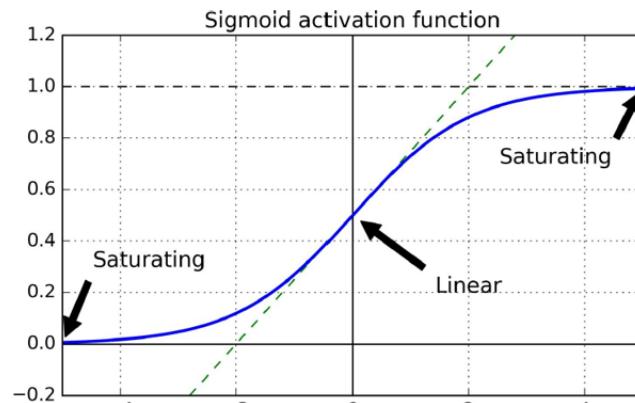


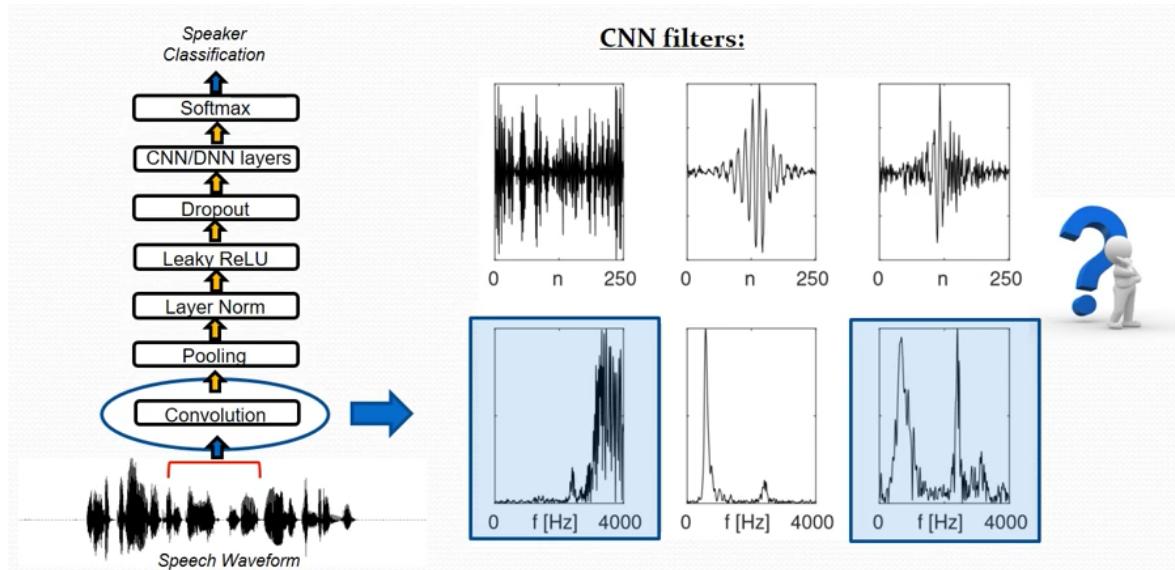
Image Credit : "Hands-on Machine Learning with Scikit-Learn and TensorFlow" by Aurelien Geron

Disrupt4.0



Hình 30: Vấn đề Vanishing Gradient trong Deep Neural Networks

Vấn đề khi xử lý tín hiệu giọng nói: Hình dạng của các bộ lọc CNN. Ngoài ra, các bộ lọc CNNs thường có những hình dạng đa băng tần không hợp lý, để hiểu nó thì với mạng Neural là điều dễ dàng, nhưng với con người thì nó không có nhiều ý nghĩa trong việc thể hiện giọng nói.



Hình 31: Ảnh được lấy từ video A brief introduction to SincNet thực hiện bởi Giáo sư Mirco Ravanelli

Những ý tưởng bắt nguồn từ Xử lý tín hiệu số và điều chỉnh lại CNN chuẩn

Với một CNN chuẩn, việc tích chập trong miền thời gian giữa input waveform và một số đáp ứng xung hữu hạn (Finite Impulse Response - FIR) được cho bởi công thức:

$$y[n] = x[n] * h[n] = \sum_{l=0}^{L-1} x[l] \cdot h[n-l]$$

Trong đó:

- $x[n]$: đoạn tín hiệu giọng nói
- $h[n]$: một mảng nạp ứng với chiều dài L
- $y[n]$: giá trị đầu ra

Trong khi đó, Sincnet thực hiện các phép tích chập của nó với hàm g , hàm này phụ thuộc vào một tham số θ . Công thức như sau:

$$y[n] = x[n] * g[n, \theta]$$

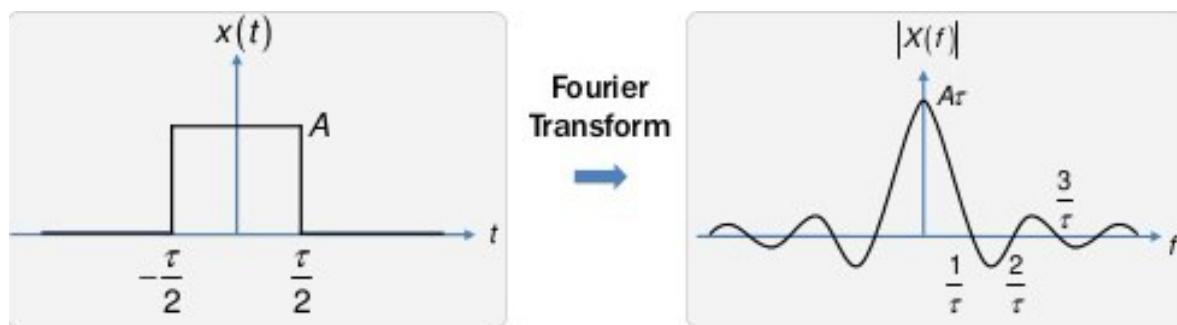
Trong xử lý tín hiệu số, g được định nghĩa như một filter-bank gồm các bộ lọc (filter) băng thông hình chữ nhật. Trong miền tần số, độ lớn của một bộ lọc băng thông tổng quát có thể được tính như hiệu số giữa 2 bộ lọc thông tần số thấp

Với f_1, f_2 lần lượt là tần số cắt thấp (low) và cao (high) đã được xác định, $\text{rect}(\cdot)$ là hàm rectangular trong miền tần số.

$$G[f, f_1, f_2] = \text{rect}\left(\frac{f}{2f_2}\right) - \text{rect}\left(\frac{f}{2f_1}\right)$$

Công thức trên đang ở trong miền tần số, để có thể trở lại miền thời gian được, ta sử dụng phép biến đổi Fourier Ngược

Note: Biến đổi Fourier cho hàm Rectangular



$$x(t) = A \text{rect}\left(\frac{t}{\tau}\right)$$

Biến đổi:

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

$$X(\omega) = \int_{-\frac{\tau}{2}}^{\frac{\tau}{2}} A e^{-j\omega t} dt = -\frac{2A}{\omega} \left[\frac{e^{-j\omega\frac{\tau}{2}} - e^{j\omega\frac{\tau}{2}}}{2j} \right] = \frac{2A}{\omega} \left[\sin\left(\frac{\omega\tau}{2}\right) \right] = A\tau \left[\frac{\sin(\frac{\omega\tau}{2})}{\frac{\omega\tau}{2}} \right]$$

Hàm sinc được định nghĩa

$$sinc(x) = \frac{\sin(x)}{x}$$

Theo đó:

$$X(\omega) = A\tau sinc\left(\frac{\omega\tau}{2}\right)$$

Áp dụng công thức:

Biến đổi ngược:

$$G[f, f_1, f_2] = rect\left(\frac{f}{2f_2}\right) - rect\left(\frac{f}{2f_1}\right)$$

Ta được hàm tham chiếu g

$$g[n, f_1, f_2] = 2f_2sinc(2\pi f_2 n) - 2f_1sinc(2\pi f_1 n)$$

Các tần số cắt (cut-off frequencies) có thể được khởi tạo một cách ngẫu nhiên trong khoảng $[0, \frac{f_2}{2}]$, trong đó f_s là tần số mẫu của tín hiệu đầu vào.

Tần suất lấy mẫu có thể thay đổi theo loại dữ liệu chúng ta đang thử nghiệm. Hệ thống IVR có tần số lấy mẫu là 8Khz, trong khi hệ thống âm thanh nổi có tần số lấy mẫu là 44khz.

Chúng ta có thể khởi tạo các bộ lọc dựa trên các tần số cắt của bộ lọc mel-scale filter-bank. Ưu điểm chính của việc chỉ định bộ lọc theo cách này là nó có lợi thế là phân bổ trực tiếp nhiều bộ lọc hơn ở phần dưới của phổ có thông tin duy nhất về giọng nói của người nói.

Để đảm bảo $f_1 \geq 0$ và $f_2 \geq f_1$, chương trình phía trên được cung cấp bởi các tham số sau:

$$f_1^{abs} = |f_1|$$

$$f_2^{abs} = f_1 + |f_2 - f_1|$$

Ở đây, không có giới hạn nào đối với f_2 , tức là không có tác nhân nào tác động lên f_2 để nó có thể nhỏ hơn tần số Nyquist (tốc độ tối thiểu mà tín hiệu có thể được lấy mẫu mà không có lỗi, gấp đôi tần số cao nhất hiện có trong tín hiệu) như mô hình học điều này trong khi huấn luyện. Các lớp tiếp theo khác nhau quyết định mức độ quan trọng nhiều hơn hoặc ít hơn cho mỗi đầu ra bộ lọc.

Bộ lọc băng thông lý tưởng cần có vô số phần tử L . Một bộ lọc băng thông lý tưởng là nơi băng thông hoàn toàn phẳng và độ suy giảm trong băng thông dừng là vô hạn. Bất kỳ sự cắt ngắn nào của g chắc chắn dẫn sẽ đến sự xấp xỉ của bộ lọc lý tưởng, được đặc trưng bởi các gợn sóng trong băng thông và suy giảm giới hạn dừng băng thông.

Vì vậy, giải pháp cửa sổ (windowing) được thực hiện để giải quyết vấn đề này. Nó được thực hiện chỉ bằng cách nhân hàm bị cắt ngắn g với cửa sổ w , nhằm mục đích làm phẳng các điểm gián đoạn đột ngột ở cuối g .

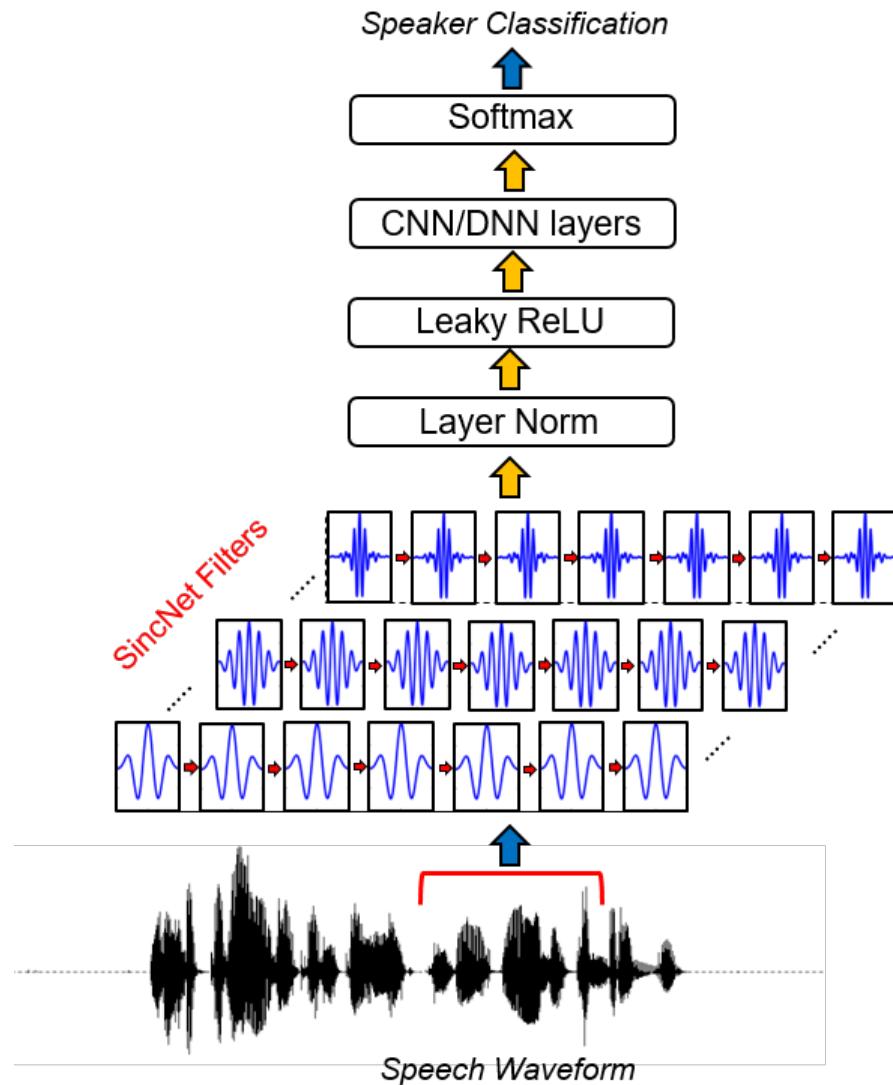
$$g_w[n, f_1, f_2] = g[n, f_1, f_2] \cdot w[n]$$

Trong bài báo, tác giả sử dụng Hamming Window, được định nghĩa bởi công thức:

$$w[n] = 0.54 - 0.46 \cdot \cos\left(\frac{2\pi n}{L}\right)$$

Chúng ta có thể có được tính chọn lọc tần số cao với việc sử dụng cửa sổ Hamming. Chúng ta cũng có thể sử dụng các cửa sổ khác như Hann, Blackman, Kaiser window. Một lưu ý quan trọng ở đây là do tính đối xứng, các bộ lọc có thể được tính toán hiệu quả bằng cách xem xét một nửa bộ lọc và kế thừa kết quả cho nửa còn lại.

Tần số cắt của các bộ lọc có thể được tối ưu với các thông số CNN sử dụng Stochastic Gradient Descent (SGD) hoặc các phương pháp tối ưu Gradient khác. Như mô hình bên dưới, CNN pipeline (Pooling, Normalization, Activations, Dropout) có thể được sử dụng sau tích chập dựa trên Sinc Convolution đầu tiên. Multiple standard convolutional, fully-connected hoặc recurrent layers có thể đặt chồng lên ở giai đoạn sau đó để cuối cùng qua Softmax Classifier (Bộ phân lớp Softmax) để phân lớp giọng nói.



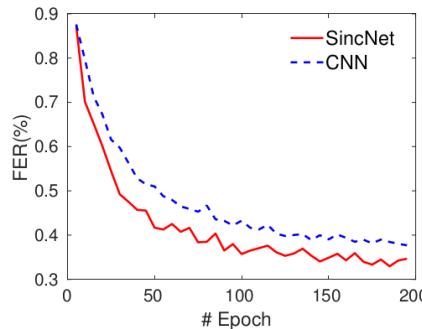
Hình 32: Kiến trúc mạng Sincnet

Đặc điểm của mô hình mạng SincNet

- **Tính hội tụ nhanh**

- Sincnet được thiết kế theo cách mà nó buộc mạng phải tập trung vào các thông số lọc ảnh hưởng đến tốc độ của nó. Phong cách kỹ thuật lọc này giúp thích ứng với dữ liệu trong khi nắm bắt được tri thức giống như kỹ thuật trích xuất đặc trưng trên dữ liệu âm thanh. Tiền tri thức này làm cho việc học các đặc tính của bộ lọc dễ dàng hơn nhiều, giúp SincNet hội

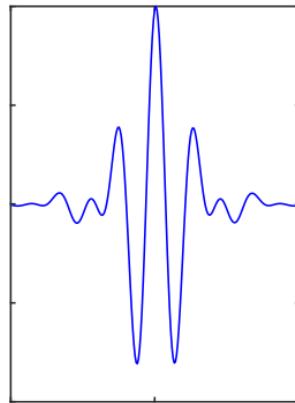
tụ nhanh hơn đáng kể đến một giải pháp tốt hơn. Chúng ta có được sự hội tụ nhanh chóng trong vòng 10–15 epochs đầu tiên.



Hình 33: Độ hội tụ của SincNet so với CNN

- **Tính hiệu quả**

- Do các hàm kernel $g(\cdot)$ là đối xứng nên ta có thể thực hiện phép tích chập trên một phần filter và kế thừa kết quả này trên phần còn lại. Điều này sẽ tiết kiệm 50% việc tính toán.



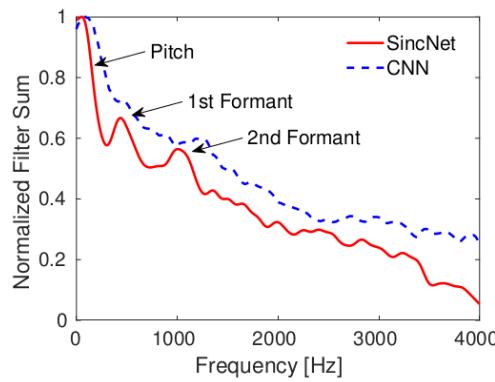
Hình 34: Hàm kernel

- **Cần ít tham số cho việc huấn luyện mô hình**

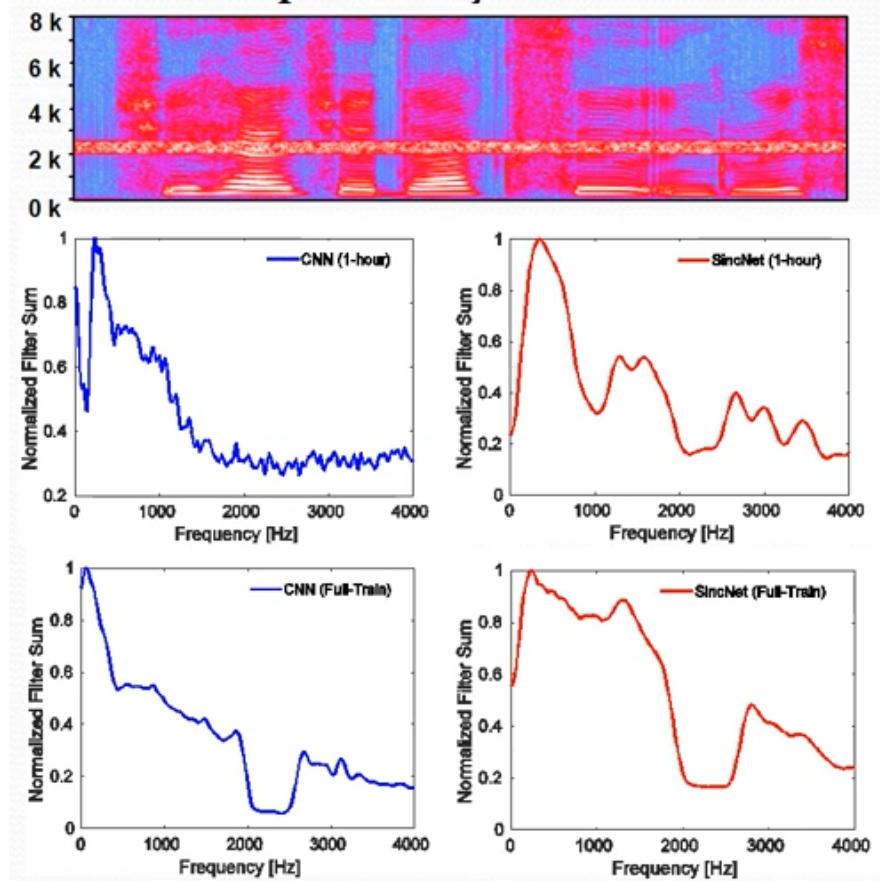
- SincNet giảm đáng kể số lượng tham số trong lớp chập đầu tiên. Ví dụ: nếu chúng ta xem xét một lớp bao gồm các bộ lọc F có độ dài L , một CNN tiêu chuẩn sử dụng các tham số $F * L$, so với $2F$ được SincNet xem xét. Nếu $F = 90$ và $L = 100$, chúng ta sử dụng 9000 tham số cho CNN và chỉ 180 cho SincNet. Hơn nữa, nếu chúng ta tăng gấp đôi độ dài bộ lọc L , một CNN chuẩn sẽ tăng gấp đôi số lượng tham số của nó (ví dụ: chúng ta đi từ 9000 lên 18000), trong khi SincNet có số lượng tham số không thay đổi (chỉ có hai tham số được sử dụng cho mỗi bộ lọc, bất kể độ dài L của nó). Điều này cung cấp khả năng tạo ra các bộ lọc rất chọn lọc với nhiều lần nhấn, mà không thực sự thêm các tham số vào vẫn đề tối ưu hóa. Hơn nữa, sự nhỏ gọn của kiến trúc SincNet làm cho nó phù hợp trong trường hợp ít mẫu.

- **Tính giải nghĩa/ diễn giải**

- Các feature maps của SincNet sau khi thực hiện lớp tích chập đầu tiên rất dễ hiểu và con người có thể hiểu được so với những cách tiếp cận khác. Trên thực tế, các filter-bank chỉ phụ thuộc vào các tham số có ý nghĩa vật lý rõ ràng.



Hình 35: Khả năng diễn giải của SincNet so với CNN



Hình 36: Khả năng diễn giải của SincNet so với CNN

Đối chiếu Convolution Neural Networks với SincNet

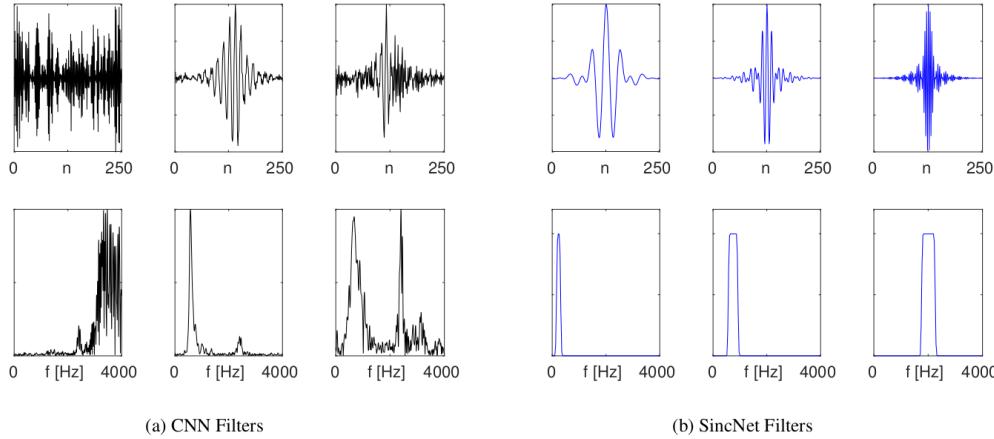


Fig. 2: Examples of filters learned by a standard CNN and by the proposed SincNet (using the Librispeech corpus). The first row reports the filters in the time domain, while the second one shows their magnitude frequency response.

Hình 37: Ví dụ về các bộ lọc được học bởi CNN tiêu chuẩn và bởi SincNet (sử dụng kho ngữ liệu Librispeech). Hàng đầu tiên thể hiện các bộ lọc trong miền thời gian, trong khi hàng thứ hai hiển thị phản hồi tần số cường độ của chúng.

Các kết quả đối với tác vụ định danh người nói: Như đã dẫn chứng ở trên, ở hình 4 trong bài báo, cho thấy learning curves của SincNet so với CNN. Ta có thể thấy rằng, Frame Error Rate giảm thực sự nhanh khi dùng SincNet. Hơn nữa, SincNet hội tụ với hiệu suất tốt hơn với FER 33.0% so với FER 37.7% của CNN

	TIMIT	LibriSpeech
DNN-MFCC	0.99	2.02
CNN-FBANK	0.86	1.55
CNN-Raw	1.65	1.00
SINCNET	0.85	0.96

Table 1: Classification Error Rate (CER%) of speaker identification systems trained on TIMIT (462 spks) and Librispeech (2484 spks) datasets. SincNets outperform the competing alternatives.

Hình 38: Bảng kết quả SicNet trong tác vụ nhận dạng giọng nói - SI

Bảng trên đây là một bảng báo cáo về tỉ lệ phân lớp lỗi (**Classification Error Rates - CER%**), khi thực nghiệm SincNet cùng với số kỹ thuật khác như **DNN-MFCC**, **CNN-FBANK**, **CNN-Raw** trên hai tập dữ liệu **TIMIT** và **LibriSpeech**. Nhìn chung, SincNet luôn dẫn đầu về độ lỗi tốt (có độ lỗi thấp nhất). Độ lỗi của **CNN-Raw** thật sự lớn khi tiến hành với tập TIMIT, điều này cho thấy SincNet của chúng ta hoạt động tốt ngay cả khi có không lớn dữ liệu huấn luyện có sẵn. Khi huấn luyện với **LibriSpeech**, độ lỗi **CNN-Raw** giảm xuống, chúng ta có 4% độ lỗi được giảm xuống, điều này cho thấy tốc độ hội tụ của SincNet cải thiện rõ ràng (1200 và 1800 epochs). Với **DNN-MFCC**, **CNN-FBANK**, hai kỹ thuật này hoạt động tốt trên **TIMIT** (vì đơn giản là **TIMIT** không lớn cho lắm như **LibriSpeech**), khi sang **LibriSpeech**, chúng có vẻ mất đi tính ổn định, độ lỗi cao lên. Các

kết quả đối với tác vụ xác minh người nói: Thủ nghiệm cuối cùng mà nhóm tác giả trình bày ở trong bài báo là tác vụ **xác minh giọng nói - Speaker Verification**. Bảng dưới đây, được trích ra trong bài báo, báo cáo về chỉ số **Equal Error Rate (EER%)** khi thực nghiệm trên tập **LibriSpeech**.

	d-vector	DNN-class
DNN-MFCC	0.88	0.72
CNN-FBANK	0.60	0.37
CNN-Raw	0.58	0.36
SINCNET	0.51	0.32

Table 2: Speaker Verification Equal Error Rate (EER%) on Librispeech datasets over different systems. SincNets outperform the competing alternatives.

Hình 39: Bảng kết quả SicNet trong tác vụ xác minh giọng nói - SV

Tất cả các mô hình DNN đều cho thấy hiệu suất đầy hứa hẹn, **các chỉ EER thấp hơn 1% trong mọi trường hợp**. Bảng cũng cho thấy rằng **SincNet lại một lần nữa hoạt động tốt hơn các mô hình khác**, cho thấy sự cải thiện hiệu suất tương đối khoảng **11% so với mô hình CNN**. Các mô hình lớp DNN hoạt động tốt hơn đáng kể so với các **d-vector**. Bất chấp hiệu quả của cách tiếp cận sau này, một mô hình DNN mới phải được huấn luyện (hoặc tinh chỉnh) cho mỗi người nói mới được thêm vào nhóm. Điều này làm cho cách tiếp cận này hoạt động tốt hơn, nhưng kém linh hoạt hơn so với **d-vector**.

Để hoàn thiện hơn, nhóm tác giả cũng tiến hành các thí nghiệm khác với các **i-vector** tiêu chuẩn. Tuy nhiên so sánh chi tiết với kỹ thuật này nằm ngoài phạm vi của bài báo nên nhóm tác giả chỉ nêu ra những điểm đáng chú ý nhất trong kết quả. Hệ thống **i-vector** tốt nhất của nhóm tác giả đạt được **EER = 1,1%, khá xa so với những gì đạt được với hệ thống DNN**. Tài liệu nổi tiếng rằng **i-vector** cung cấp hiệu suất cạnh tranh khi sử dụng nhiều dữ liệu huấn luyện hơn cho mỗi người nói và khi các câu kiểm tra dài hơn được sử dụng. Trong các điều kiện thách thức phải đổi mặt trong công việc này, mạng neural đạt được khả năng tổng quát hóa tốt hơn.

Nhận xét về kiến trúc SincNet

- Cơ sở lý thuyết Toán học vững vàng: Kỹ thuật band-pass filter, Window trong Xử lý tín hiệu số.
- Tính toán nhanh và gọn nhẹ: Như đã nói, đây là một đặc điểm của SincNet nhờ vào dùng ít tham số, kernel đối xứng.
- Kết hợp với Deep Learning một cách hiệu quả.
- Sử dụng DNN-Class trong đánh giá, cho kết quả đầy hứa hẹn, có độ lỗi EER thấp

Nhưng vẫn có hạn chế

- DNN-class tuy có EER thấp nhưng đánh đổi nhiều sự linh hoạt so với d-vectors

E. Thực nghiệm của nhóm

E.1 Phương pháp

Cài đặt mô hình Nhận dạng Người nói và đánh giá nó với tiếng Việt và tiếng Anh với SincNet. Các tác vụ thành phần của mô hình: Identification (Định danh) và Verification (Xác minh)

E.2 Kho ngữ liệu

Tiếng Anh

Sử dụng hai tập dữ liệu đã được đề cập trong bài báo

Với **TIMIT**, ta có một kho ngữ liệu với 462 người nói, các khoảng không phải lời nói ở đầu và cuối mỗi câu đã bị xóa, những tập tin về nội dung câu nói của TIMIT cũng được loại bỏ. Sau khi tinh chỉnh toàn bộ dữ liệu, tác giả dùng 5 câu nói của mỗi người nói để huấn luyện, 3 câu nói của mỗi người nói dùng để kiểm tra.

Với tập ngữ liệu **LibriSpeech**, những phần với độ im lặng bên trong kéo dài hơn 125 ms được chia thành nhiều phần nhỏ. Việc chia tập huấn luyện (training set), tập kiểm tra (testing set) là ngẫu nhiên bằng cách chọn 12-15 giây dữ liệu huấn luyện của mỗi người nói và các câu kiểm tra kéo dài từ 2-6 giây.

Tiếng Việt

Sử dụng tập dữ liệu Son et al. Dataset

Nguồn dữ liệu từ bài báo Vietnamese Speaker Authentication Using Deep Models

- Dung lượng của tập dữ liệu: 535 MB
- Số mẫu trong tập dữ liệu: 400 mẫu
- Bộ dữ liệu gồm: hai tập Men và Women, mỗi tập con chứa 10 thư mục người nói. Mỗi thư mục người nói chứa 20 đoạn ghi âm, chia ra Long và Short (mỗi loại 10 đoạn)
- Nội dung câu nói
 - Câu ngắn: "Tôi là sinh viên chuyên ngành công nghệ thông tin"
 - Câu dài: "Tôi là sinh viên Học viện Công nghệ Bưu chính Viễn thông, chương trình đào tạo khá nặng đỏi hỏi sinh viên phải học tập và nghiên cứu rất nhiều nhưng tôi tự hào vì đó là ngành đã và đang làm thay đổi cuộc sống xã hội loài người".
- Điểm hạn chế: Bộ dữ liệu có kích thước khá nhỏ

E.3 Thực nghiệm

E.3.1 Thực nghiệm trên tập TIMIT

- Xử lý dữ liệu
 - Việc xử lý dữ liệu không cần nhiều sự phức tạp, nhưng cần phải thiết lập lại đường dẫn về dạng lowercase "train/dr1/fcjf0/si1027.wav"
 - Chạy file TIMIT_preparation.py chuẩn hóa dữ liệu
- Mô hình
 - Các cửa sổ có $fs = 16000$, tín hiệu được cắt thành những chunks với $cw_len = 200$, $overlap_shift = 10s$
 - Lớp Input: sử dụng 80 bộ lọc SincNet có kích thước $L = 251$, max pool - 3, sử dụng Layer Norm cho cả input và output, không dùng Batch Norm, hàm kích hoạt activation leaky-ReLU, dropout = 0
 - Hai lớp CNN: sử dụng 2 lớp CNN, với mỗi lớp dùng 60 bộ lọc có kích thước $L = 5$, sử dụng Layer Norm cho cả input và output, không dùng Batch Norm, hàm kích hoạt activation leaky-ReLU, dropout = 0
 - Ba lớp DNN: sử dụng 3 lớp DNN (Multi Layer Perceptron) fully-connected với 2048 neurons, Layer Norm cho input, Batch Norm cho output, các lớp ẩn (hidden layers) dùng leaky-ReLU
 - Lớp Output: Multi Layer Perceptron, 462 nodes, không dropout, không LayerNorm, không BatchNorm cho cả input và output, hàm activation function dùng softmax
 - Hàm mất mát: Negative Log Likelihood Loss
- Hyper parameters

- learning rate lr = 0.001
- $\alpha = 0.95$
- $\epsilon = 10^{-7}$
- batch_size = 128
- N_epochs = 100
- N_batches = 800
- N_eval_epoch = 8
- seed = 1234

E.3.2 Thực nghiệm trên tập Librispeech

- Xử lý dữ liệu
 - Việc xử lý dữ liệu không cần nhiều sự phức tạp, nhưng cần phải thiết lập lại đường dẫn về dạng lowercase "train/dr1/fcjf0/si1027.wav"
- Mô hình
 - Các cửa sổ có fs = 8000, tín hiệu được cắt thành những chunks với cw_len = 375, overlap cw_shift = 10s
 - Lớp Input: sử dụng 80 bộ lọc SincNet có kích thước $L = 251$, max pool = 3, sử dụng Layer Norm cho cả input và output, không dùng Batch Norm, hàm kích hoạt activation leaky-ReLU, dropout = 0
 - Hai lớp CNN: sử dụng 2 lớp CNN, với mỗi lớp dùng 60 bộ lọc có kích thước $L = 5$, sử dụng Layer Norm cho cả input và output, không dùng Batch Norm, hàm kích hoạt activation leaky-ReLU, dropout = 0
 - Hai lớp DNN: sử dụng 2 lớp DNN (Multi Layer Perceptron) fully-connected với 2048 neurons, Layer Norm cho input, Batch Norm cho output, các lớp ẩn (hidden layers) dùng leaky-ReLU làm activation cho lớp DNN thứ nhất, lớp kia dùng linear
 - Lớp Output: 2 lớp Multi Layer Perceptron, 2048 nodes cho mỗi lớp, không dropout, Layer Norm cho input, Batch Norm cho output, hàm activation function lớp thứ nhất dùng leaky-ReLU, lớp thứ hai dùng softmax
 - Hàm mất mát: Negative Log Likelihood Loss
- Hyper parameters
 - learning rate lr = 0.001
 - $\alpha = 0.95$
 - $\epsilon = 10^{-7}$
 - batch_size = 128
 - N_epochs = 100
 - N_batches = 100
 - N_eval_epoch = 10
 - reg_factor = 1000
 - fact_amp=0.2
 - seed = 1234

E.3.2 Thực nghiệm trên tập Son et al. Dataset

- Xử lý dữ liệu
 - Việc xử lý dữ liệu không cần nhiều sự phức tạp, nhưng cần phải thiết lập lại đường dẫn về dạng lowercase "train/dr1/fcjf0/si1027.wav"
 - Có một chút thay đổi về source code so với tác giả, nhằm chuẩn hóa dữ liệu âm thanh về dạng mono channel
- Mô hình
 - Các cửa sổ có $fs = 16000$, tín hiệu được cắt thành những chunks với $cw_len = 200$, overlap $cw_shift = 10s$
 - Lớp Input: sử dụng 80 bộ lọc SincNet có kích thước $L = 251$, max pool - 3, sử dụng Layer Norm cho cả input và output, không dùng Batch Norm, hàm kích hoạt activation leaky-ReLU, dropout = 0
 - Hai lớp CNN: sử dụng 2 lớp CNN, với mỗi lớp dùng 60 bộ lọc có kích thước $L = 5$, sử dụng Layer Norm cho cả input và output, không dùng Batch Norm, hàm kích hoạt activation leaky-ReLU, dropout = 0
 - Ba lớp DNN: sử dụng 3 lớp DNN (Multi Layer Perceptron) fully-connected với 2048 neurons, Layer Norm cho input, Batch Norm cho output, các lớp ẩn (hidden layers) dùng leaky-ReLU
 - Lớp Output: Multi Layer Perceptron, 18 nodes, không dropout, không LayerNorm, không BatchNorm cho cả input và output, hàm activation function dùng softmax
 - Hàm mất mát: Negative Log Likelihood Loss
- Hyper parameters
 - learning rate lr = 0.001
 - $\alpha = 0.95$
 - $\epsilon = 10^{-7}$
 - batch_size = 128
 - N_epochs = 300
 - N_batches = 100
 - N_eval_epoch = 1
 - seed = 1234

E.4 Đánh giá mô hình

Nhóm sử dụng một số độ đo trong việc đánh giá mô hình như sau

- loss_tr: là mất mát huấn luyện trung bình (tức là hàm cross-entropy) được tính ở mọi khung.
- err_tr: là lỗi phân loại (đo ở mức khung) của dữ liệu huấn luyện. Lưu ý rằng chia tín hiệu giọng nói thành các phần 200ms với 10ms chồng lên nhau. Lỗi được tính trung bình cho tất cả các phần của tập dữ liệu huấn luyện.
- loss_te là mất mát kiểm tra trung bình (tức là hàm cross-entropy) được tính ở mọi khung.
- err_te: là phân loại sai (đo ở mức khung) của dữ liệu thử nghiệm.
- err_te_snt: là phân loại sai (đo ở mức câu) của dữ liệu kiểm tra.

Lưu ý: Theo bài báo, nhóm chia tín hiệu giọng nói thành các phần 200ms với 10ms chồng lên nhau. Đối với mỗi đoạn, SincNet thực hiện dự đoán trên bộ dữ liệu người nói. Để tính toán tỷ lệ lỗi phân loại này, nhóm lấy trung bình các dự đoán và đối với mỗi câu, nhóm chọn người nói có xác suất trung bình cao nhất.

Với tác vụ SI, nhóm chia dữ liệu ra làm 2 loại, người nói mang id chẵn, người nói mang id lẻ, sau đó tính toán ROC AUC (Compute Area Under the Receiver Operating Characteristic Curve), AP(average precision)

AUC (Area Under The Curve) - ROC (Receiver Operating Characteristics) là một phương pháp tính toán hiệu suất của một mô hình phân loại theo các ngưỡng phân loại khác nhau. Giả sử với bài toán phân loại nhị phân (2 lớp) sử dụng hồi quy logistic (logistic regression), việc chọn các ngưỡng phân loại [0..1] khác nhau sẽ ảnh hưởng đến khả năng phân loại của mô hình và ta cần tính toán được mức độ ảnh hưởng của các ngưỡng

ROC là một đường cong biểu diễn xác suất và AUC biểu diễn mức độ phân loại của mô hình

Ý nghĩa: - Xác suất rằng một mẫu Positive được lấy ngẫu nhiên sẽ được xếp hạng cao hơn một mẫu Negative được lấy ngẫu nhiên. Biểu diễn theo công thức, ta có $AUC = P(score(x+) > score(x-))$
- Chỉ số AUC càng cao thì mô hình càng chính xác trong việc phân loại các lớp.

Theo Machinelearning cơ bản:

Với bài toán phân loại mà tập dữ liệu của các lớp là chênh lệch nhau rất nhiều có một phép đo hiệu quả thường được sử dụng là Precision-Recall.

Với một cách xác định một lớp là positive, Precision được định nghĩa là tỉ lệ số điểm true positive trong số những điểm được phân loại là positive ($TP + FP$).

Recall được định nghĩa là tỉ lệ số điểm true positive trong số những điểm thực sự là positive ($TP + FN$).

$$\text{Precision} = \frac{\text{True Positive}}{\text{True Positive} + \text{False Positive}}$$

$$\text{Recall} = \frac{\text{True Positive}}{\text{True Positive} + \text{False Negative}}$$

Precision cao đồng nghĩa với việc độ chính xác của các điểm tìm được là cao. Recall cao đồng nghĩa với việc True Positive Rate cao, tức tỉ lệ bỏ sót các điểm thực sự positive là thấp.

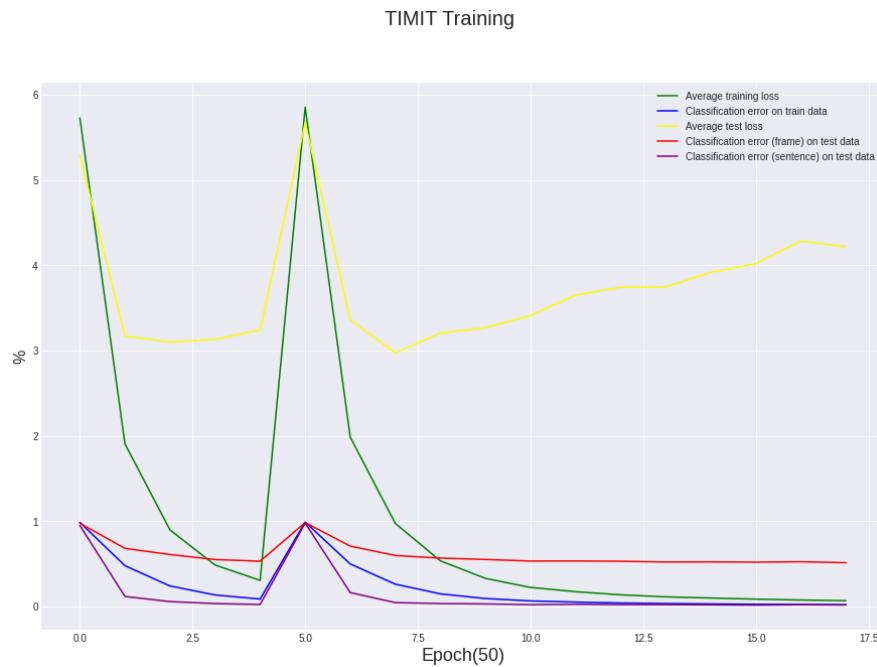
Tương tự như ROC curve, chúng ta cũng có thể đánh giá mô hình dựa trên việc thay đổi một ngưỡng và quan sát giá trị của Precision và Recall. Khái niệm Area Under the Curve (AUC) cũng được định nghĩa tương tự. Với Precision-Recall Curve, AUC còn có một tên khác là Average precision (AP).

E.5 Các kết quả

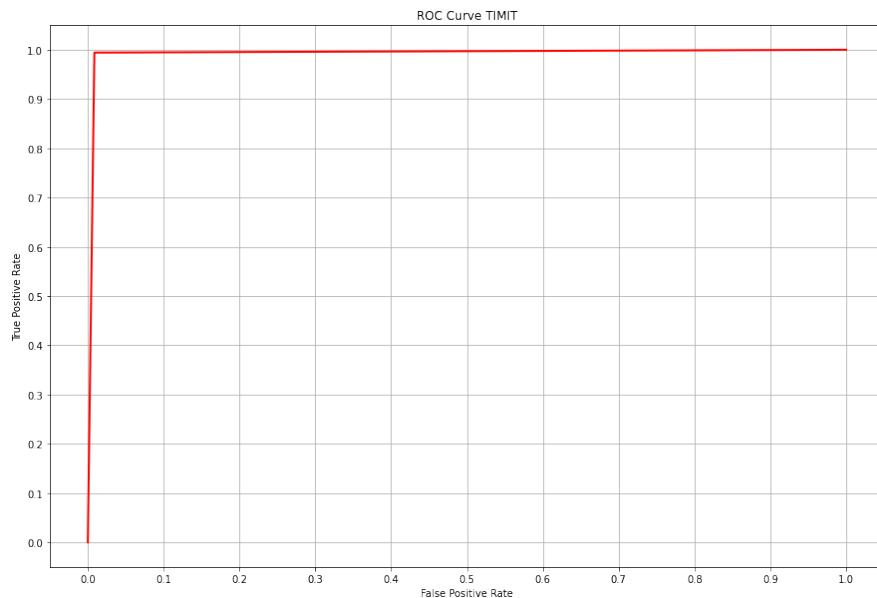
E.5.1 Kết quả thực nghiệm trên tập TIMIT

```
number 1383 from 1386
number 1384 from 1386
number 1385 from 1386
loss_te=4.217127 err_te=0.513561 err_te_snt=0.018038
[0, 0, 0, 1, 1, 1, 2, 2, 2, 3, 3, 3, 4, 4, 4, 5, 5, 5, 5, 6, 6, 6, 6, 7, 7, 7
[0, 0, 0, 1, 1, 1, 2, 2, 2, 3, 3, 3, 4, 4, 4, 5, 5, 5, 5, 6, 6, 6, 6, 7, 7, 7
```

Hình 40: Kết quả thực nghiệm trên TIMIT Dataset



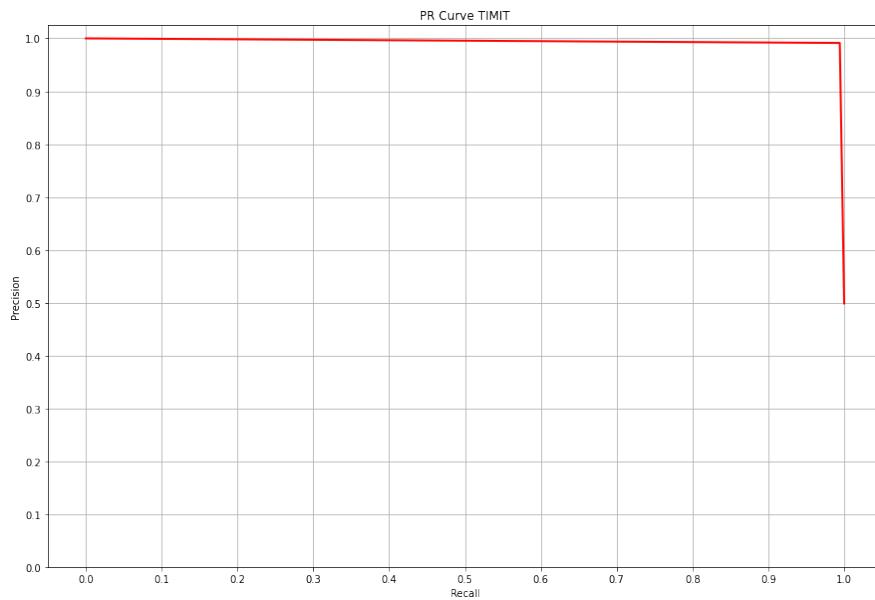
Hình 41: Đồ thị kết quả training TIMIT Dataset



Hình 42: Đồ thị ROC Curve TIMIT Dataset

Kết quả đánh giá:

- EER = 0.0086
- AUC = 0.99



Hình 43: Đồ thị PR Curve TIMIT Dataset

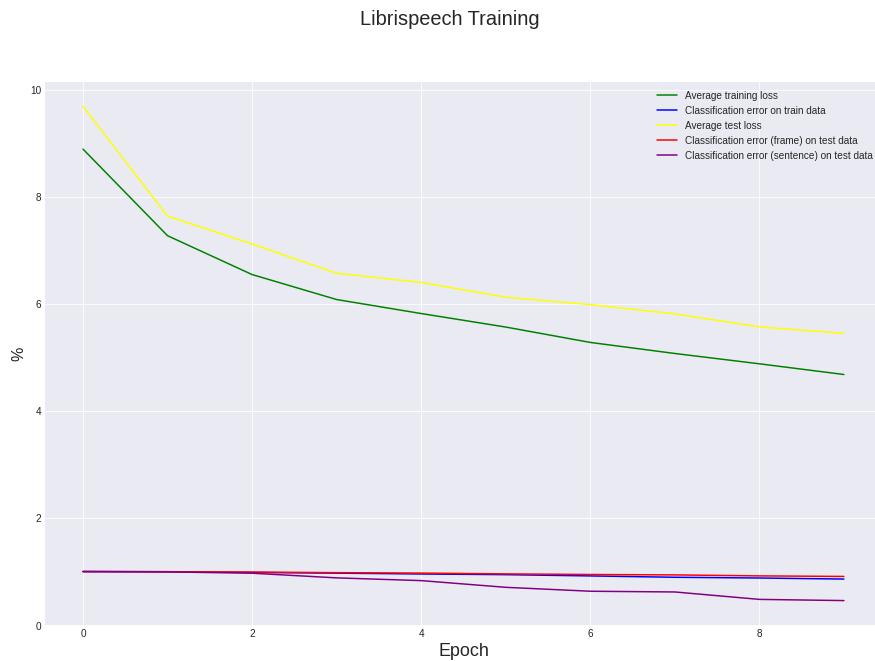
Kết quả đánh giá:

- AP = 0.99

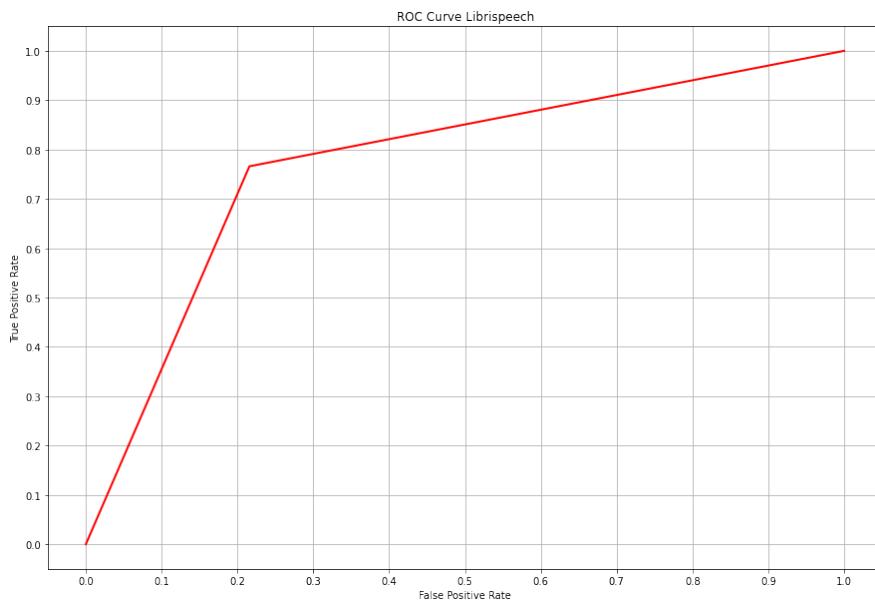
E.5.2 Kết quả thực nghiệm trên tập LibriSpeech

```
number 7445 from 7452
number 7446 from 7452
number 7447 from 7452
number 7448 from 7452
number 7449 from 7452
number 7450 from 7452
number 7451 from 7452
loss_te=5.448840 err_te=0.907977 err_te_snt=0.456924
[1, 1, 1, 2, 2, 2, 0, 0, 0, 4, 4, 4, 5, 5, 5, 3, 3, 3, 7, 7, 7,
[1, 1887, 1244, 1964, 1733, 1919, 0, 0, 0, 2159, 2159, 285, 5,
```

Hình 44: Kết quả thực nghiệm trên LibriSpeech Dataset



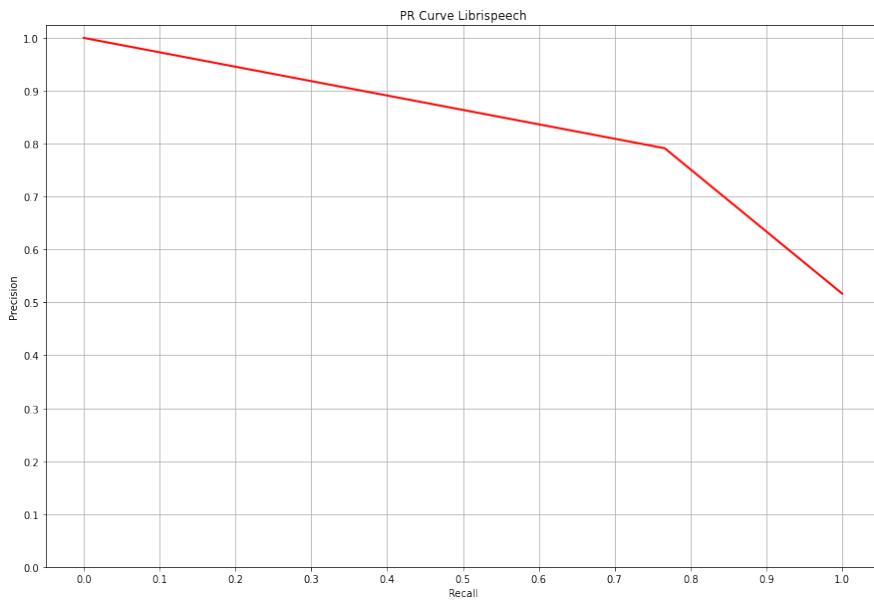
Hình 45: Kết quả thực nghiệm trên Librispeech Dataset



Hình 46: Đồ thị ROC Curve Librispeech Dataset

Kết quả đánh giá:

- EER = 0.22
- AUC = 0.78



Hình 47: Đồ thị PR Curve Librispeech Dataset

Kết quả đánh giá:

- AP = 0.73

E.5.3 Kết quả thực nghiệm trên tập Son et al. Dataset

```
number 67 from 72
number 68 from 72
number 69 from 72
number 70 from 72
number 71 from 72
loss_te=0.113859 err_te=0.031011 err_te_snt=0.000000
[0, 0, 0, 0, 1, 1, 1, 1, 2, 2, 2, 2, 3, 3, 3, 3, 4, 4, 4, 4, 4, 5, 5,
 [0, 0, 0, 0, 1, 1, 1, 1, 2, 2, 2, 2, 3, 3, 3, 3, 4, 4, 4, 4, 4, 5, 5,
```

Hình 48: Kết quả thực nghiệm trên Son et al. Dataset

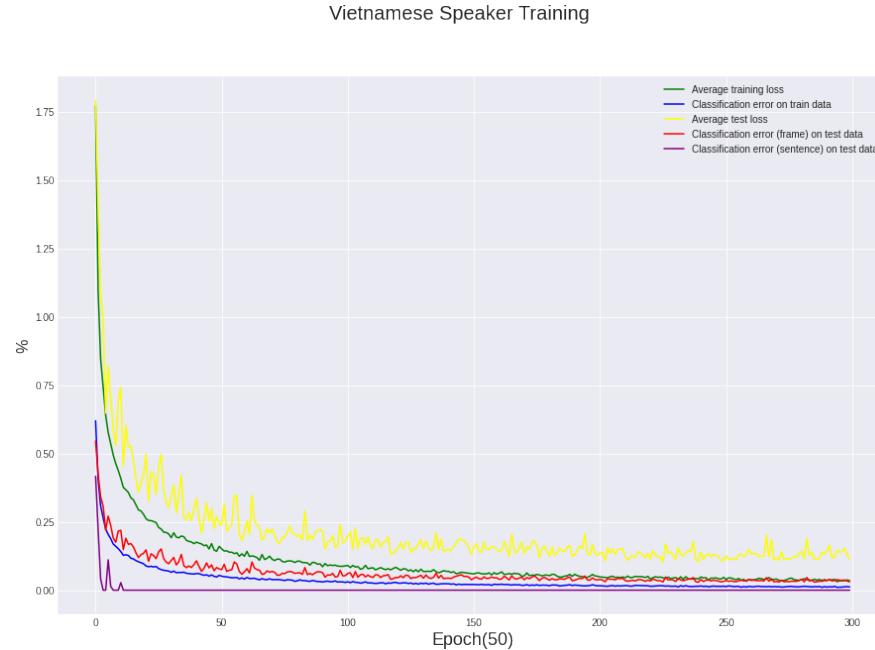
```

res.res X
264 epoch 263, loss_tr=0.043678 err_tr=0.014453 loss_te=0.122491 err_te=0.034715 err_te_snt=0.000000
265 epoch 264, loss_tr=0.042996 err_tr=0.012969 loss_te=0.123601 err_te=0.035475 err_te_snt=0.000000
266 epoch 265, loss_tr=0.039295 err_tr=0.013594 loss_te=0.123835 err_te=0.033832 err_te_snt=0.000000
267 epoch 266, loss_tr=0.041003 err_tr=0.012500 loss_te=0.201843 err_te=0.045690 err_te_snt=0.000000
268 epoch 267, loss_tr=0.046464 err_tr=0.014062 loss_te=0.121270 err_te=0.032715 err_te_snt=0.000000
269 epoch 268, loss_tr=0.036485 err_tr=0.012187 loss_te=0.184504 err_te=0.042124 err_te_snt=0.000000
270 epoch 269, loss_tr=0.039752 err_tr=0.013359 loss_te=0.11025 err_te=0.030692 err_te_snt=0.000000
271 epoch 270, loss_tr=0.040501 err_tr=0.012422 loss_te=0.118827 err_te=0.032187 err_te_snt=0.000000
272 epoch 271, loss_tr=0.036301 err_tr=0.013047 loss_te=0.108904 err_te=0.029903 err_te_snt=0.000000
273 epoch 272, loss_tr=0.034507 err_tr=0.011250 loss_te=0.140276 err_te=0.036521 err_te_snt=0.000000
274 epoch 273, loss_tr=0.035937 err_tr=0.011797 loss_te=0.111528 err_te=0.030920 err_te_snt=0.000000
275 epoch 274, loss_tr=0.038486 err_tr=0.013125 loss_te=0.116042 err_te=0.031999 err_te_snt=0.000000
276 epoch 275, loss_tr=0.042671 err_tr=0.013437 loss_te=0.114230 err_te=0.031305 err_te_snt=0.000000
277 epoch 276, loss_tr=0.044173 err_tr=0.014062 loss_te=0.117546 err_te=0.032913 err_te_snt=0.000000
278 epoch 277, loss_tr=0.038138 err_tr=0.012578 loss_te=0.121886 err_te=0.032787 err_te_snt=0.000000
279 epoch 278, loss_tr=0.037533 err_tr=0.012422 loss_te=0.126570 err_te=0.032560 err_te_snt=0.000000
280 epoch 279, loss_tr=0.035381 err_tr=0.012266 loss_te=0.135943 err_te=0.036351 err_te_snt=0.000000
281 epoch 280, loss_tr=0.039425 err_tr=0.013281 loss_te=0.142036 err_te=0.039428 err_te_snt=0.000000
282 epoch 281, loss_tr=0.041706 err_tr=0.014375 loss_te=0.114884 err_te=0.030905 err_te_snt=0.000000
283 epoch 282, loss_tr=0.038156 err_tr=0.013516 loss_te=0.189182 err_te=0.046511 err_te_snt=0.000000
284 epoch 283, loss_tr=0.037572 err_tr=0.013125 loss_te=0.127604 err_te=0.033644 err_te_snt=0.000000
285 epoch 284, loss_tr=0.036046 err_tr=0.011875 loss_te=0.138393 err_te=0.036829 err_te_snt=0.000000
286 epoch 285, loss_tr=0.040144 err_tr=0.013125 loss_te=0.112067 err_te=0.029580 err_te_snt=0.000000
287 epoch 286, loss_tr=0.036607 err_tr=0.012422 loss_te=0.112889 err_te=0.031038 err_te_snt=0.000000
288 epoch 287, loss_tr=0.039669 err_tr=0.013594 loss_te=0.116345 err_te=0.032431 err_te_snt=0.000000
289 epoch 288, loss_tr=0.031790 err_tr=0.010469 loss_te=0.138557 err_te=0.037250 err_te_snt=0.000000
290 epoch 289, loss_tr=0.039106 err_tr=0.013203 loss_te=0.131656 err_te=0.034593 err_te_snt=0.000000
291 epoch 290, loss_tr=0.035024 err_tr=0.011016 loss_te=0.144353 err_te=0.036220 err_te_snt=0.000000
292 epoch 291, loss_tr=0.038430 err_tr=0.012891 loss_te=0.164588 err_te=0.039235 err_te_snt=0.000000
293 epoch 292, loss_tr=0.037788 err_tr=0.011719 loss_te=0.116891 err_te=0.033663 err_te_snt=0.000000
294 epoch 293, loss_tr=0.039578 err_tr=0.014062 loss_te=0.121106 err_te=0.035298 err_te_snt=0.000000
295 epoch 294, loss_tr=0.038085 err_tr=0.010547 loss_te=0.144321 err_te=0.038643 err_te_snt=0.000000
296 epoch 295, loss_tr=0.034566 err_tr=0.010937 loss_te=0.139187 err_te=0.036375 err_te_snt=0.000000
297 epoch 296, loss_tr=0.034118 err_tr=0.011250 loss_te=0.140623 err_te=0.035937 err_te_snt=0.000000
298 epoch 297, loss_tr=0.035262 err_tr=0.012578 loss_te=0.153683 err_te=0.037338 err_te_snt=0.000000
299 epoch 298, loss_tr=0.037386 err_tr=0.013047 loss_te=0.129246 err_te=0.033569 err_te_snt=0.000000
300 epoch 299, loss_tr=0.034833 err_tr=0.012031 loss_te=0.113608 err_te=0.030943 err_te_snt=0.000000
301

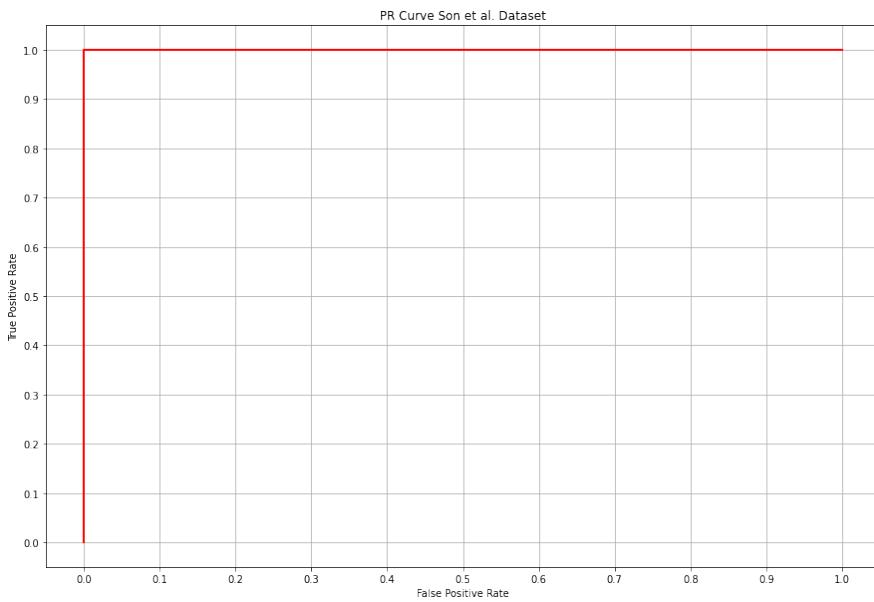
```

Hình 49: Kết quả thực nghiệm trên Son et al. Dataset

Quá trình huấn luyện loss_tr=0.034833 (NLLLoss), err_tr=0.012031 (FER), loss_te=0.113608 (NLLLoss), err_te=0.030943 (FER), err_te_snt=0.000000 (CER)



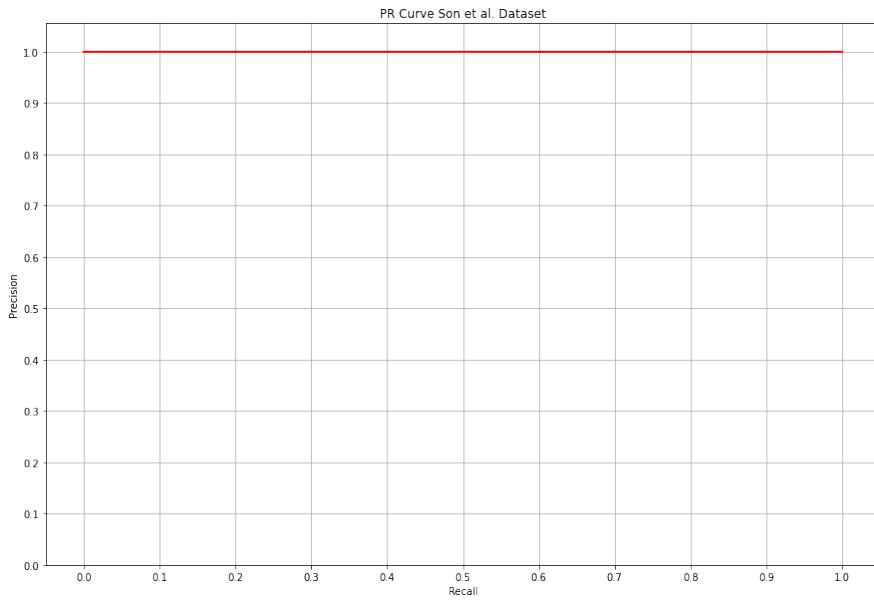
Hình 50: Kết quả thực nghiệm trên Son et al. Dataset



Hình 51: Đồ thị ROC Curve Son et al. Dataset

Kết quả đánh giá:

- EER = 0.0
- AUC = 1.0



Hình 52: Đồ thị PR Curve Son et al. Dataset

Kết quả đánh giá:

- AP = 1.0

E.5.4 Suy luận từ mô hình nhận dạng giọng nói tiếng Việt

Kiểm tra suy luận từ mô hình với giọng nói thu từ trình duyệt Google Colab

```

def get_audio():
    display(HTML(AUDIO_HTML))
    data = eval_js("data")
    binary = b64decode(data.split(',')[1])

    process = [ffmpeg
        .input('pipe:0')
        .output('pipe:1', format='wav')
        .run_async(pipe_stdin=True, pipe_stdout=True, pipe_stderr=True, quiet=True, overwrite_output=True)
    ]
    output, err = process.communicate(input=binary)

    riff_chunk_size = len(output) - 8
    # Break up the chunk size into four bytes, held in b.
    q = riff_chunk_size
    b = []
    for i in range(4):
        q, r = divmod(q, 256)
        b.append(r)

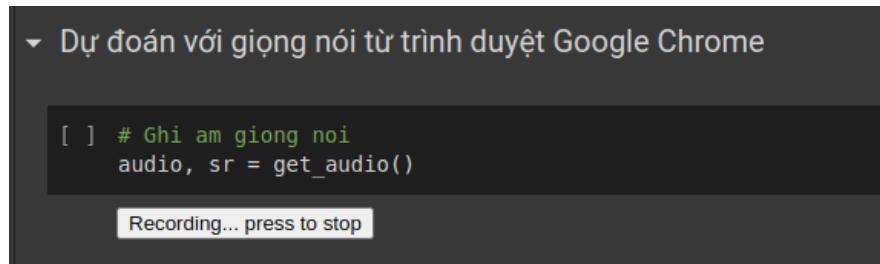
    # Replace bytes 4:8 in proc.stdout with the actual size of the RIFF chunk.
    riff = output[:4] + bytes(b) + output[8:]

    sr, audio = wav_read(io.BytesIO(riff))

    return audio, sr

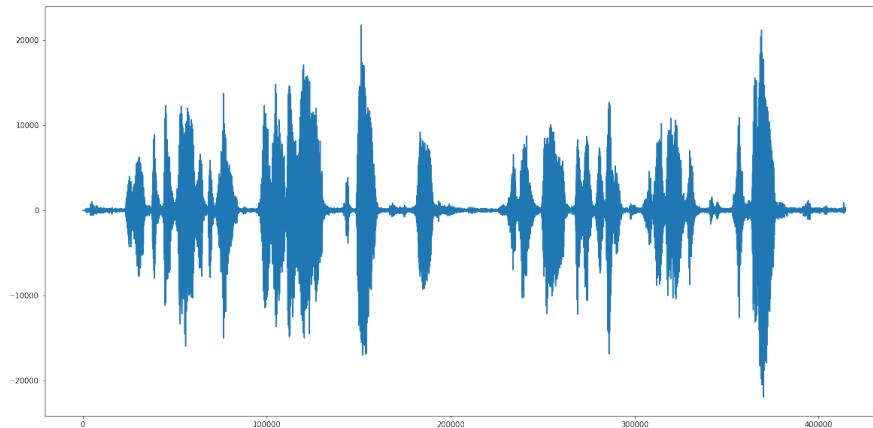
```

Hình 53: Hàm thu âm giọng nói từ trình duyệt



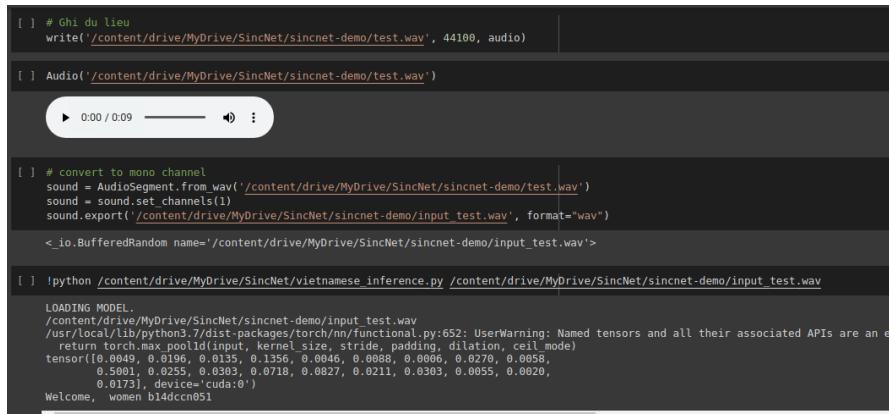
Hình 54: Thu giọng nói từ trình duyệt

Trực quan hình ảnh dạng tần số của âm thanh giọng nói



Hình 55: Trực quan hình ảnh dạng tần số của âm thanh giọng nói

Tiến hành tải mô hình đã được huấn luyện, và đưa ra suy luận



```
[ ] # Ghi du lieu
    write('/content/drive/MyDrive/SincNet/sincnet-demo/test.wav', 44100, audio)

[ ] Audio('/content/drive/MyDrive/SincNet/sincnet-demo/test.wav')

[ ] 0:00 / 0:09 ━━━━━━ ◁ : [ ] # convert to mono channel
    sound = AudioSegment.from_wav('/content/drive/MyDrive/SincNet/sincnet-demo/test.wav')
    sound = sound.set_channels(1)
    sound.export('/content/drive/MyDrive/SincNet/sincnet-demo/input_test.wav', format="wav")
    <_io.BufferedReader name='/content/drive/MyDrive/SincNet/sincnet-demo/input_test.wav'>

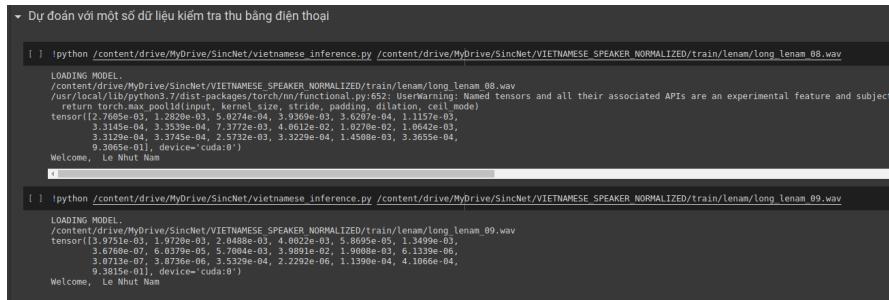
[ ] !python /content/drive/MyDrive/SincNet/vietnamese_inference.py /content/drive/MyDrive/SincNet/sincnet-demo/input_test.wav

LOADING MODEL.
/content/drive/MyDrive/SincNet/sincnet-demo/input_test.wav
/usr/local/lib/python3.7/dist-packages/torch/mn/functional.py:652: UserWarning: Named tensors and all their associated APIs are an experimental feature and subject to change. https://github.com/pytorch/pytorch/pull/16352
    return torch.max_pool2d(input, kernel_size, stride, padding, dilation, cell_mode)
tensor([0.0049, 0.0196, 0.0135, 0.1356, 0.0046, 0.0088, 0.0006, 0.0278, 0.0058,
       0.5001, 0.0255, 0.0303, 0.0718, 0.0827, 0.0211, 0.0303, 0.0055, 0.0020,
       0.0173], device='cuda:0')
Welcome, women b14dcc061
```

Hình 56: Suy luận từ mô hình

Kết quả cho thấy mô hình dự đoán sai, không xuất ra "Welcome, Le Nhut Nam". Điều này được nhóm cho là mô hình bị mắc phải vấn đề overfitting do tập dữ liệu huấn luyện khá nhỏ.

Dự đoán từ những âm thanh thu từ micro điện thoại



```
▼ Dự đoán với một số dữ liệu kiểm tra thu bằng điện thoại

[ ] !python /content/drive/MyDrive/SincNet/vietnamese_inference.py /content/drive/MyDrive/VIETNAMESE_SPEAKER NORMALIZED/train/lenam/long_lenam_08.wav

LOADING MODEL.
/content/drive/MyDrive/VIETNAMESE_SPEAKER NORMALIZED/train/long_lenam_08.wav
/usr/local/lib/python3.7/dist-packages/torch/mn/functional.py:652: UserWarning: Named tensors and all their associated APIs are an experimental feature and subject to change. https://github.com/pytorch/pytorch/pull/16352
    return torch.max_pool2d(input, kernel_size, stride, padding, dilation, cell_mode)
tensor([2.7605e-03, 1.2820e-03, 5.0274e-04, 3.9369e-03, 5.6207e-04, 1.1157e-03,
       3.6222e-03, 4.8612e-03, 1.0270e-02, 1.0642e-02,
       3.3129e-04, 3.3745e-04, 2.5732e-03, 5.1229e-04, 1.4988e-03, 3.3835e-04,
       9.3065e-01], device='cuda:0')
Welcome, Le Nhut Nam

[ ] !python /content/drive/MyDrive/SincNet/vietnamese_inference.py /content/drive/MyDrive/SincNet/VIETNAMESE SPEAKER NORMALIZED/train/lenam/long_lenam_09.wav

LOADING MODEL.
/content/drive/MyDrive/SincNet/VIETNAMESE SPEAKER NORMALIZED/train/lenam/long_lenam_09.wav
/usr/local/lib/python3.7/dist-packages/torch/mn/functional.py:652: UserWarning: Named tensors and all their associated APIs are an experimental feature and subject to change. https://github.com/pytorch/pytorch/pull/16352
    return torch.max_pool2d(input, kernel_size, stride, padding, dilation, cell_mode)
tensor([1.9750e-03, 6.8468e-03, 4.0602e-03, 5.6585e-03, 1.3405e-03,
       3.6760e-07, 6.0370e-05, 5.7084e-03, 3.9881e-02, 1.9098e-03, 6.1339e-06,
       3.0713e-07, 3.0736e-06, 3.3329e-04, 2.2292e-06, 1.1398e-04, 4.1066e-04,
       9.3815e-01], device='cuda:0')
Welcome, Le Nhut Nam
```

Hình 57: Suy luận từ mô hình

Kết quả cho thấy mô hình dự đoán đúng, xuất ra "Welcome, Le Nhut Nam". Điều này cho thấy dữ liệu khớp với những mẫu âm thanh giọng nói thu từ điện thoại hơn là thu được từ máy tính.

E.6 Những nhận xét

Tổng kết lại những gì tìm hiểu được

- Về lý thuyết:
 - Sinh trắc học giọng nói là một lĩnh lớn có nhiều ứng dụng, với các phương pháp truyền thống giải quyết tương đối bài toán này
 - Các phương pháp hiện đại sử dụng các phương pháp rút trích đặc trưng dựa vào mạng Học Sâu (d-vectors, j-vectors và x-vectors) đem lại nhiều kết quả khả quan, đầy mong đợi
 - Đặc biệt, với bài toán phân lớp giọng nói việc kết hợp đa nhiệm trở nên một hướng giải quyết tốt cho bài toán này. Ngoài ra, SincNet đem lại một làn gió mới, khi tận dụng và tối ưu lại CNN truyền thống bằng các bộ lọc hình Sinc, giúp tác vụ nhận dạng người nói có nhiều triển vọng khi kết hợp với d-vectors, DNN-class.
- Về thực hành:
 - Nhóm đã thực hiện thu thập dữ liệu, huấn luyện cơ bản, nắm cách hoạt động của một Speaker Recognition Pipeline
 - Đọc hiểu source code Pytorch implement SincNet của tác, thiết lập và tùy chỉnh lại theo đúng với dữ liệu của nhóm

- Trực quan kết quả, đạt được những kết quả tương đối tốt và gần giống với kết quả từ bài báo
- Thử nghiệm dự đoán từ mô hình SincNet với tiếng Việt, bị overfitting do tập dữ liệu quá nhỏ.

Tài liệu

- [1] Github repository: Wavencoder - a python library for encoding raw audio with pytorch backend.
- [2] Wav2vec 2.0: Learning the structure of speech from raw audio.
- [3] S. Chopra, R. Hadsell, and Y. LeCun. Learning a similarity metric discriminatively, with application to face verification. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, volume 1, pages 539–546 vol. 1, 2005.
- [4] Juan M. Coria, Hervé Bredin, Sahar Ghannay, and Sophie Rosset. A comparison of metric learning loss functions for end-to-end speaker verification, 2020.
- [5] Juan M. Coria, Hervé Bredin, Sahar Ghannay, and Sophie Rosset. Github repository: Companion repository for the paper "a comparison of metric learning loss functions for end-to-end speaker verification" published at slsp 2020, 2020.
- [6] Oscar Forth (1) Finnian Kelly (1), Anil Alexander (1) and David van der Vloed (2). From i-vectors to x-vectors – a generational change in speaker recognition illustrated on the nfi-frida database.
- [7] Evans Kiplagat Francesco Grauso. Github repository: Keras (tensorflow) implementation of sincnet (mirco ravanelli, yoshua bengio, 2018).
- [8] Omid Ghahabi and Javier Hernando. Deep learning backend for single and multisession i-vector speaker recognition. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 25(4):807–817, 2017.
- [9] Anil K. Jain, Patrick Flynn, and Arun A. Ross. *Handbook of Biometrics*. Springer Publishing Company, Incorporated, 1st edition, 2010.
- [10] Andy T. Liu, Shu-wen Yang, Po-Han Chi, Po-chun Hsu, and Hung-yi Lee. Mockingjay: Unsupervised speech representation learning with deep bidirectional transformer encoders. *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, May 2020.
- [11] Kai Yu Nanxin Chen, Yanmin Qian. Multi-task learning for text-dependent speaker verification. In *INTERSPEECH 2015 16th Annual Conference of the International Speech Communication Association*, pages 185–189, 2015.
- [12] Son T. Nguyen, Viet D. Lai, Quyen Dam-Ba, Anh Nguyen-Xuan, and Cuong Pham. Vietnamese speaker authentication using deep models. In *Proceedings of the Ninth International Symposium on Information and Communication Technology*, SoICT 2018, page 177–184, New York, NY, USA, 2018. Association for Computing Machinery.
- [13] Santiago Pascual, Mirco Ravanelli, Joan Serrà, Antonio Bonafonte, and Yoshua Bengio. Learning problem-agnostic speech representations from multiple self-supervised tasks, 2019.
- [14] Mirco Ravanelli and Yoshua Bengio. Github repository: Sincnet original code written in pytorch by the autor, 2019.
- [15] Mirco Ravanelli and Yoshua Bengio. Speaker recognition from raw waveform with sincnet, 2019.
- [16] Steffen Schneider, Alexei Baevski, Ronan Collobert, and Michael Auli. wav2vec: Unsupervised pre-training for speech recognition, 2019.
- [17] David Snyder, Daniel Garcia-Romero, Gregory Sell, Daniel Povey, and Sanjeev Khudanpur. X-vectors: Robust dnn embeddings for speaker recognition. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5329–5333, 2018.

- [18] Zhiyuan Tang, Lantian Li, and Dong Wang. Multi-task recurrent model for speech and speaker recognition, 2016.
- [19] Ehsan Variani, Xin Lei, Erik McDermott, Ignacio Lopez Moreno, and Javier Gonzalez-Dominguez. Deep neural networks for small footprint text-dependent speaker verification. In *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4052–4056, 2014.
- [20] Jee weon Jung, Seung bin Kim, Hye jin Shim, Ju ho Kim, and Ha-Jin Yu. Improved rawnet with feature map scaling for text-independent speaker verification using raw waveforms, 2020.