HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG

TRẦN THỊ NHI AN

NHẬN DẠNG NGƯỜI NÓI THEO TIẾP CẬN MÁY HỌC HIỆN ĐẠI

Chuyên ngành: HỆ THỐNG THÔNG TIN

Mã số: 8.48.01.04

TÓM TẮT LUẬN VĂN THẠC SĨ

(Theo định hướng ứng dụng)

TP. HỒ CHÍ MINH – NĂM 2022

Luận văn được hoàn thành tại:

HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG

Người hướng dẫn khoa học: **PGS.TS VŨ HẢI QUÂN**

Phản biện 1: .....

Phản biện 2: .....

Luận văn sẽ được bảo vệ trước Hội đồng chấm luận văn tại Học viện Công nghệ Bưu chính Viễn Thông

Vào lúc: ..... giờ ..... ngày ..... tháng ..... năm .....

Có thể tìm hiểu luận văn tại:

- Thư viện của Học viện Công nghệ Bưu Chính Viễn Thông.

MỞ ĐẦU

Ở thời điểm bùng nổ về CNTT-TT, IoT và CMCN 4.0 thì vai trò của Sinh trắc học càng được nhấn mạnh hơn trong nhiều lĩnh vực xã hội và đời sống. Ngày càng có nhiều công trình trên thế giới khai thác các đặc tính sinh trắc để làm cầu nối giữa ứng dụng thực tiễn và xác thực chủ thể. Tuy nhiên, nghiên cứu trong nước về lĩnh vực này lại chưa nhiều, chưa có những giải pháp thực sự thuyết phục được cộng đồng và doanh nghiệp. Do đó, luận văn mong muốn góp một phần nhỏ vào khảo sát học thuật mà cụ thể là đặc tính sinh trắc về giọng nói, nhằm làm tăng tính khả thi hơn cho ứng dụng trong nước.

Mục tiêu của đề tài là khảo sát tính khả thi của việc áp dụng các mô hình máy học hiện đại cho lĩnh vực nhận dạng người nói, kỳ vọng sẽ mang lại hiệu năng/độ chính xác cao hơn các phương pháp truyền thống. Khi mà nền tảng công nghệ được cải tiến hơn, các ứng dụng sinh trắc sẽ hấp dẫn hơn với thị trường và doanh nghiệp.

Luận văn gồm 5 chương chính với các nội dung sau:

Chương 1: Giới thiệu về lĩnh vực nghiên cứu của đề tài, các nghiên cứu liên quan trong và ngoài nước. Đồng thời, nêu rõ mục tiêu cũng như hướng nghiên cứu của đề tài.

Chương 2: Trình bày tổng quan về đề tài bao gồm nhận dạng người nói, các đặc trưng của tín hiệu giọng nói và các mô hình máy học.

Chương 3: Trình bày phương pháp nhận dạng người nói với Deep Learning cụ thể là mô hình HMM là Feedforward-DNN.

Chương 4: Trình bày chi tiết việc xây dựng bộ dữ liệu, quá trình cụ thể cài đặt mô hình cho thuật toán và đánh giá kết quả thực nghiệm trên bộ dữ liệu xây dựng với hai phương pháp đề xuất cùng với phần demo chương trình.

Chương 5: Kết luận nội dung đã được trong đề tài, nêu những khó khăn, hạn chế trong quá trình nghiên cứu đã gặp phải và đề xuất hướng phát triển tiếp theo.

Đề tài: NHẬN DẠNG NGƯỜI NÓI THEO TIẾP CẬN MÁY HỌC HIỆN ĐẠI

Tóm tắt luận văn

CHƯƠNG 1. PHẦN MỞ ĐẦU

1.1. Lĩnh vực đề tài

Đề tài thuộc lĩnh vực Sinh trắc học (Biometrics). Sinh trắc học là khoa học nghiên cứu các phương pháp phân tích và thống kê trên các dữ liệu sinh học. Các hệ thống sinh trắc đã và đang được phát triển trong các ứng dụng thực tế như: các hoạt động của chính phủ, các công ty, tổ chức thương mại – tài chính, bao gồm việc quản lý nhân công, quản lý khách 2ang, quản lý kiểm soát vào ra, đến quản lý xuất nhập cảnh, quản lý tội phạm, hệ thống bầu cử, v.v… Nhận dạng sinh trắc hiện đại đang nhận được nhiều sự quan tâm trong các lĩnh vực cần mức độ bảo mật và an toàn cao, cũng như do tính thuận tiện và năng động mà nó mang lại. Từ đó nó đã ngày càng chứng minh được tiềm năng ứng dụng rộng rãi so với các phương pháp nhận dạng truyền thống. Đề tài "**Nhận dạng người nói theo tiếp cận máy học hiện đại**", với mong muốn góp một phần nhỏ vào khảo sát học thuật mà cụ thể là đặc tính sinh trắc về giọng nói.

1.2. Tình hình nghiên cứu liên quan đến đề tài

1.2.1. Các công trình nghiên cứu trong nước

- Deep Representation Learning for Vietnamese Speaker Recognition

- Text-dependent Speaker Recognition for Vietnamese

- Vietnamese Speaker Authentication Using Deep Models

- Speaker Diarization in Vietnamese Voice

1.2.2. Các công trình nghiên cứu trên thế giới

- Deep CNNs With Self-Attention for Speaker Identification

- Novel cascaded Gaussian mixture model-deep neural network classifier for speaker identification in emotional talking environments

- An MFCC-based text-independent speaker identification system for access control

- …

1.3. Mục tiêu, ý nghĩa khoa học và thực tiễn

Mục tiêu của đề tài là khảo sát tính khả thi của việc áp dụng các mô hình máy học hiện đại cho lĩnh vực nhận dạng người nói, kỳ vọng sẽ mang lại hiệu năng/độ chính xác cao hơn các phương pháp truyền thống. Khi mà nền tảng công nghệ được cải tiến hơn, các ứng dụng sinh trắc sẽ hấp dẫn hơn với thị trường và doanh nghiệp.

Xuất phát từ những mục tiêu chính trên, luận văn hướng tới những kết quả sau:

- Tìm hiểu tổng quan về nhận dạng giọng nói.

- Tìm hiểu các thuật toán trong việc nhận dạng giọng nói.

- Tìm hiểu và xây dựng bộ dữ liệu giọng nói dùng để làm đầu vào cho mô hình

1.4. Đối tượng và phạm vi nghiên cứu

1.4.1 Đối tượng nghiên cứu

- Mô hình nhận dạng người nói tiếng Việt trong máy học

1.4.2 Phạm vi nghiên cứu

- Định danh người nói tiếng Việt độc lập văn bản và dữ liệu thực nghiệm là trên 40 người nói khác nhau

1.5. Phương pháp nghiên cứu

1.5.1. Phương pháp nghiên cứu lý thuyết

1.5.2. Phương pháp nghiên cứu thực nghiệm

1.6. Bố cục luận văn

CHƯƠNG 2. TỔNG QUAN ĐỀ TÀI

2.1. Giới thiệu chung

2.1.1. Nhận dạng người nói là gì?

Nhận dạng người qua giọng nói là một trong những nhánh được nghiên cứu phát triển mạnh trong sinh trắc học, bởi lẽ như ta đã biết trong các đặc tính sinh học trên cơ thể người, tiếng nói là một đặc điểm mang tính phổ thông, dễ phát sinh và không cần đến các thiết bị thu phức tạp. Nhiều công trình đã được nghiên cứu trên tiếng nói nhằm khai thác các thông tin từ lĩnh vực này. Cụ thể hơn, nhận dạng người nói (speaker recognition) [2] bao gồm 2 loại là: nhận dạng độc lập văn bản (text-independent) và nhận dạng phụ thuộc văn bản (text-dependent).

2.1.2. Ứng dụng công nghệ nhận dạng người nói vào đời sống

Các công nghệ nhận dạng người nói được sử dụng trong các lĩnh vực ứng dụng rộng rãi [15]. Các lĩnh vực mà các kỹ thuật nhận dạng người nói có thể được sử dụng này là xác thực, giám sát và nhận dạng trong pháp y, bảo mật, nhận dạng giọng nói, theo dõi nhiều người nói, giao diện người dùng được cá nhân hóa

2.1.3. Tổng quan về trí tuệ nhân tạo (AI)

Trí tuệ nhân tạo (AI) [16] đã trở nên rất phổ biến trong thế giới ngày nay. Trí tuệ nhân tạo là trí tuệ do con người lập trình tạo nên với mục tiêu giúp máy tính có thể tự động hóa các hành vi thông minh như con người. Trí tuệ nhân tạo khác với việc lập trình logic trong các ngôn ngữ lập trình là ở việc ứng dụng các hệ thống học máy để mô phỏng trí tuệ của con người trong các xử lý mà con người làm tốt hơn máy tính.

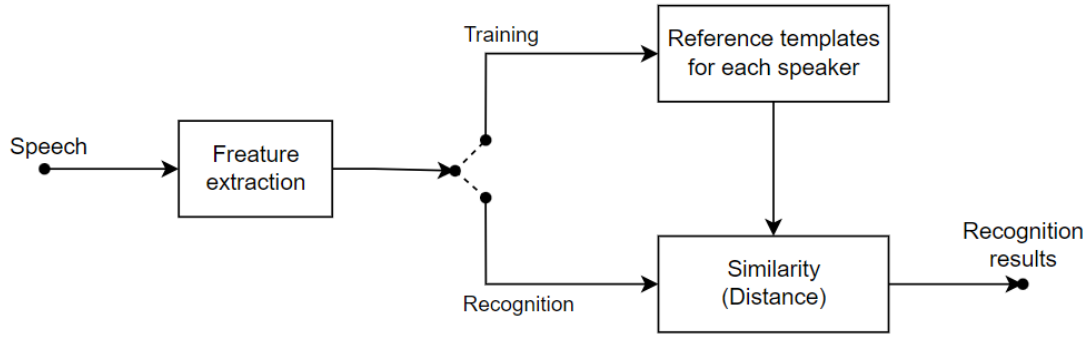
2.2 Tín hiệu giọng nói

Tín hiệu âm thanh là một đại diện của âm thanh. Nó mã hóa tất cả các thông tin cần thiết cần thiết để tái tạo âm thanh. Tín hiệu âm thanh có hai loại cơ bản: analog và digital. Analog là âm thanh được ghi lại bằng cách sử dụng các phương pháp tái tạo sóng âm thanh gốc. Ví dụ bao gồm các bản ghi vinyl và băng cassette. Âm thanh digital được ghi lại bằng cách lấy các mẫu của sóng âm thanh gốc ở một tốc độ xác định, được gọi là tốc độ lấy mẫu (sampling rate). Đĩa CD và tệp MP3 là những ví dụ

về các định dạng kỹ thuật số. Trong thế giới thực, việc chuyển đổi giữa các dạng sóng digital và analog là phổ biến và cần thiết. ADC và DAC là một phần của quá trình xử lý tín hiệu âm thanh và chúng đạt được những chuyển đổi này.

2.3 Các thành phần chính của hệ thống nhận dạng người nói

Hệ thống nhận dạng người nói thường bao gồm ba đơn vị chính [18] như hình dưới đây. Đầu vào cho giai đoạn đầu tiên hoặc cho hệ thống xử lý đầu cuối là tín hiệu giọng nói. Tại đây giọng nói được số hóa và sau đó việc rút trích đặc trưng sẽ diễn ra. Quá trình cuối cùng trong giai đoạn xử lý đầu cuối là một số hình thức bù kênh. Các thiết bị đầu vào khác nhau áp đặt các đặc điểm phổ khác nhau lên tín hiệu giọng nói, chẳng hạn như giới hạn và định hình băng tần. Do đó bù kênh được thực hiện để loại bỏ các tác dụng không mong muốn này. Thông thường nhất, một số dạng bù kênh tuyến tính, chẳng hạn như phép trừ trung bình cộng dài hạn và ngắn hạn được áp dụng cho các tính năng. Cơ bản của phép trừ quang phổ là năng lượng quang phổ của tín hiệu lời nói bị nhiễu bởi tiếng ồn bằng tổng năng lượng quang phổ của tín hiệu và nhiễu.



Hình 2.1: Cấu trúc của hệ thống nhận dạng người nói [19]

2.4 Rút trích đặc trưng

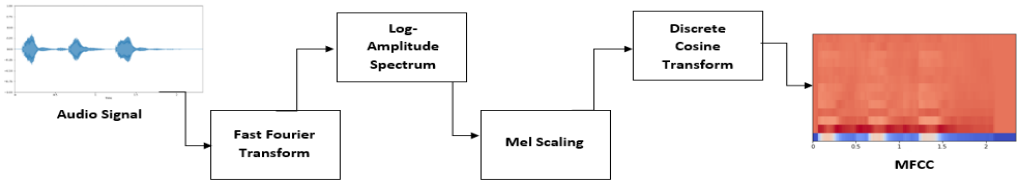
2.4.1. Rút trích đặc trưng là gì

Trích xuất đặc trưng âm thanh [18], [20] là một bước cần thiết trong xử lý tín hiệu âm thanh, là một trường con của quá trình xử lý tín hiệu. Nó liên quan đến việc xử lý hoặc thao tác các tín hiệu âm thanh, loại bỏ tiếng ồn không mong muốn và cân bằng các dải tần số thời gian bằng cách chuyển đổi tín hiệu kỹ thuật số và tín hiệu analog. Nó tập trung vào các phương pháp tính toán để thay đổi âm thanh, biến đổi

tín hiệu âm thanh thô thành một biểu diễn nhỏ gọn. Trích xuất đặc trưng là một thuật ngữ chung để chỉ các phương pháp xây dựng tổ hợp các biến để giải quyết các vấn đề này trong khi vẫn mô tả dữ liệu với độ chính xác đầy đủ.

2.4.2. Các đặc trưng âm thanh phổ biến cho việc thiết lập mô hình

a. Mức độ trừu tượng: Các danh mục thuộc phân loại này chủ yếu bao gồm các tín hiệu âm nhạc hơn là âm thanh nói chung: High-level, Mid-level, Low-level.
b. Phạm vi tạm thời: Loại của phân loại này áp dụng cho âm thanh nói chung, nghĩa là cả âm nhạc và không âm nhạc: Tức thời (Instantaneous), Cấp độ phân đoạn (Segment-level), Toàn cầu (global).
c. Khía cạnh âm nhạc: Thuộc tính âm thanh bao gồm nhịp, nhịp điệu, âm sắc (màu sắc của âm thanh), cao độ, hòa âm, giai điệu, v.v…
d. Miền tín hiệu (Signal domain) bao gồm: Miền thời gian (time-domain), Miền tần số (frequency-domain), Biểu diễn thời gian-tần số (time-frequency representation)
e. Phương pháp tiếp cận ML: bao gồm Học máy truyền thống, Học sâu:



Hình 2.2: Các bước trích xuất MFCC từ tín hiệu âm thanh (nguồn [18])

Từ hình 2.5 đã trình bày thông tin về tốc độ thay đổi trong các dải phổ của một tín hiệu được đưa ra bởi cepstrum của nó. Cepstrum về cơ bản là một phổ của nhật ký về phổ của tín hiệu thời gian. Phổ kết quả không nằm trong miền tần số cũng không nằm trong miền thời gian do đó nó được đặt tên là miền quefrency (đảo chữ cái của từ tần số). Cepstrum truyền tải các giá trị khác nhau cấu tạo nên các chất tạo thành (một thành phần đặc trưng của chất lượng âm thanh lời nói) và âm sắc của âm thanh. Vì vậy, MFCC rất hữu ích cho các mô hình học sâu.

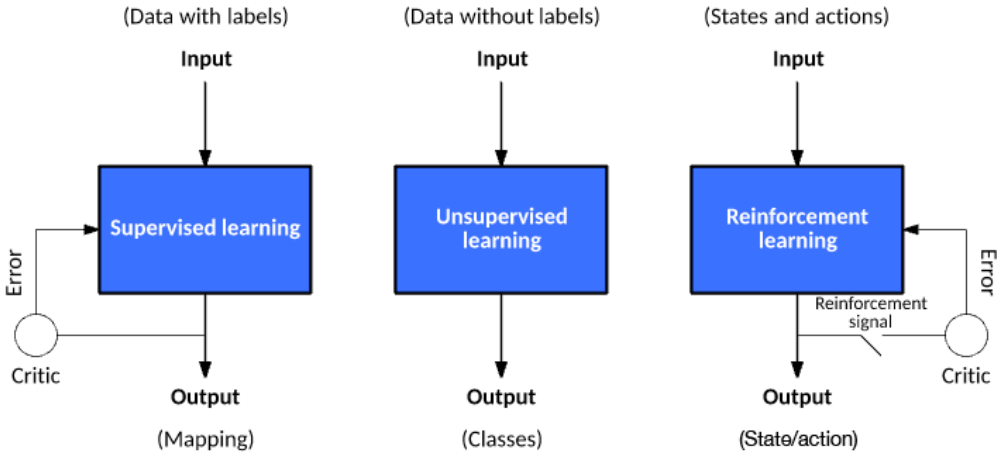
2.5. Mô hình máy học

2.5.1. Khái niệm về máy học

Học máy (Machine Learning) [22] là một tập hợp con của AI cung cấp cho máy tính khả năng tự động học hỏi dựa trên dữ liệu đưa vào mà không cần phải được lập trình cụ thể. Do đó, học máy là một phương pháp giúp máy móc giải quyết vấn đề bằng cách đạt được khả năng suy nghĩ.

2.5.2. Các loại mô hình máy học

Các thuật toán học máy không ngừng lớn mạnh và phát triển. Tuy nhiên, trong hầu hết các trường hợp, các thuật toán có xu hướng chuyển thành một trong ba mô hình cho việc học tập. Các mô hình tồn tại để tự động điều chỉnh theo một cách nào đó nhằm cải thiện hoạt động hoặc hành vi của chúng.



Hình 2.3: Ba mô hình học tập cho các thuật toán [25]

CHƯƠNG 3. NHẬN DẠNG NGƯỜI NÓI VỚI DEEP LEARNING

3.1 Mạng nơ-ron và deep learning

3.1.1. Mạng nơ-ron

Neural network [26] hay còn gọi là mạng nơ-ron được xây dựng dựa trên mạng nơ-ron sinh học. Nó là một mạng lưới gồm các nút được kết nối với nhau – gọi là nơ-ron và các cạnh nối chúng lại với nhau. Một mạng nơ-ron xử lý một vectơ đầu vào thành một vectơ đầu ra kết quả thông qua một mô hình lấy cảm hứng từ các nơ-ron và khả năng kết nối của chúng trong não. Mô hình bao gồm các lớp tế bào thần kinh được kết nối với nhau thông qua các trọng số làm thay đổi tầm quan trọng của một số đầu vào nhất định so với những đầu vào khác. Mạng nơ-ron là một mạng có cấu trúc và nhiều lớp (layer). Một mạng nơ-ron có 3 lớp chính là: input, hidden và output.

3.1.2. Deep learning

Là một phạm trù nhỏ của ML, DL tập trung giải quyết các vấn đề liên quan đến mạng thần kinh nhân tạo nhằm nâng cấp các công nghệ như nhận diện giọng nói, tầm nhìn máy tính và xử lý ngôn ngữ tự nhiên.

3.2. Phân loại / các dạng mạng neural nhân tạo

- Perceptron
- Feed Forward Neural Network
- Multilayer Perceptron
- Convolutional Neural Network
- …

3.3. Nhận dạng người nói

3.3.1. Nhận dạng người nói với HMM

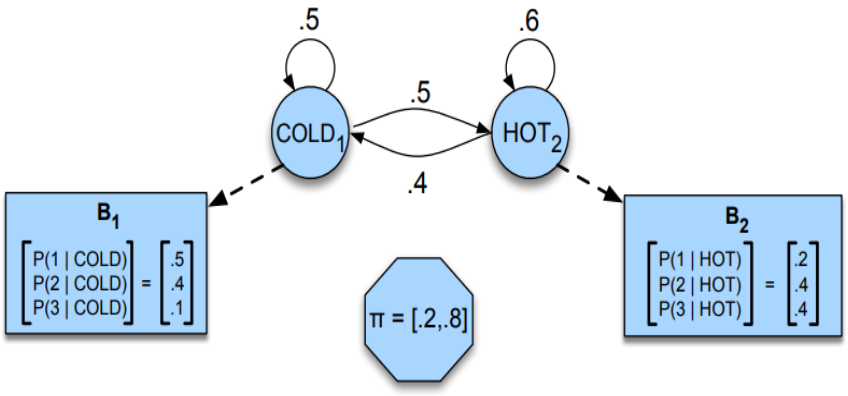
Mô hình Markov ẩn (HMM) [29] là một mô hình thống kê cũng được sử dụng trong học máy. Nó có thể được sử dụng để mô tả diễn biến của các sự kiện có thể quan sát được phụ thuộc vào các yếu tố bên trong mà không thể quan sát trực tiếp được. Đây là một loại mô hình đồ họa xác suất cho phép chúng ta dự đoán một chuỗi các biến chưa biết từ một tập hợp các biến quan sát. HMM dựa trên việc tăng cường

chuỗi Markov. Chuỗi Markov là một mô hình cho biết điều gì đó về xác suất của chuỗi các biến ngẫu nhiên, các trạng thái, mỗi biến ngẫu nhiên có thể nhận các giá trị từ một số tập hợp. Những tập hợp này có thể là các từ, hoặc thẻ, hoặc biểu tượng đại diện cho bất kỳ thứ gì, như thời tiết. Chuỗi Markov đưa ra một giả định rất mạnh mẽ rằng nếu muốn dự đoán tương lai trong chuỗi, tất cả những gì quan trọng là trạng thái hiện tại. Các trạng thái trước trạng thái hiện tại không có tác động đến tương lai ngoại trừ thông qua trạng thái hiện tại. Như thể để dự đoán thời tiết ngày mai, có thể kiểm tra thời tiết của ngày hôm nay nhưng không được phép xem thời tiết của ngày hôm qua. Chuỗi Markov rất hữu ích khi cần tính xác suất cho một chuỗi các sự kiện có thể quan sát được. Tuy nhiên, trong nhiều trường hợp, các sự kiện quan tâm bị ẩn đi. Ví dụ: ta thường không quan sát việc gán nhãn từ loại trong một văn bản. Thay vào đó, ta nhìn thấy các từ và phải suy ra các nhãn từ chuỗi từ. Ta gọi các nhãn là ẩn bởi vì chúng không được quan sát thấy.HMM cho phép nói về cả các sự kiện được quan sát (như các từ nhìn thấy trong đầu vào) và các sự kiện ẩn (như gán nhãn từ loại) mà ta coi là các yếu tố nhân quả trong mô hình xác suất. Mô hình Markov ẩn bậc nhất tạo ra hai giả thiết đơn giản hóa. Đầu tiên, như với chuỗi Markov bậc nhất, xác suất của một trạng thái cụ thể chỉ phụ thuộc vào trạng thái trước đó:

Giả thuyết Markov: $P(q_i = a|q_1 \dots q_{i-1}) = P(q_i = a|q_{i-1})$

Thứ hai, xác suất của một quan sát đầu ra o_i chỉ phụ thuộc vào trạng thái tạo ra q_i quan sát chứ không phụ thuộc vào bất kỳ trạng thái nào khác hoặc bất kỳ quan sát nào khác:

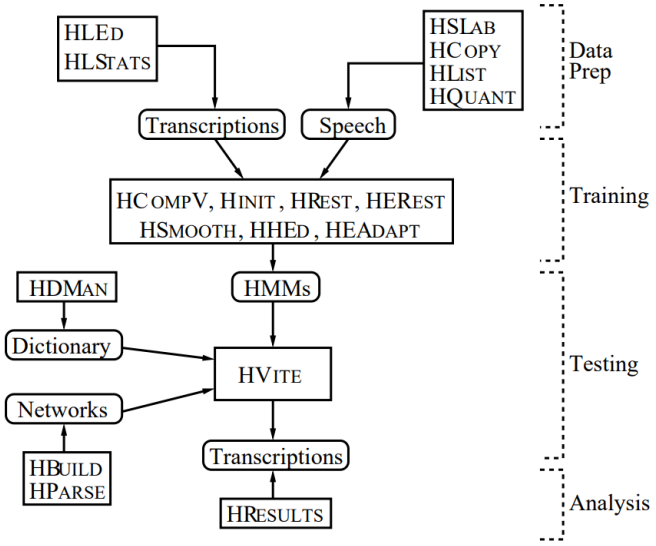
Đầu ra: $P(o_i|q_1 \dots q_i, \dots, q_T, o_1, \dots, o_i, \dots, o_T) = P(o_i|q_i)$



Hình 3.1: Một mô hình Markov ẩn [29]

HTK

HTK là một bộ công cụ để xây dựng Mô hình Markov ẩn (HMM). HMM có thể được sử dụng để mô hình hóa bất kỳ chuỗi thời gian nào và cốt lõi của HTK cũng có mục đích chung tương tự. Tuy nhiên, HTK chủ yếu được thiết kế để xây dựng các công cụ xử lý giọng nói dựa trên HMM, cụ thể là các trình biên dịch. Do đó, phần lớn sự hỗ trợ về cơ sở hạ tầng trong HTK được dành riêng cho nhiệm vụ này. Các module của HTK được thể hiện qua hình dưới đây



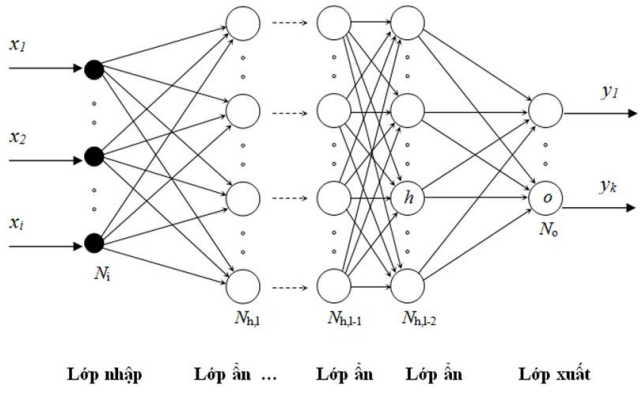
Hình 3.2: Các giai đoạn xử lý trong HTK [30]

HTK bao gồm một tập hợp các mô-đun thư viện và các công cụ có sẵn ở dạng nguồn C. Có 4 giai đoạn chính: chuẩn bị, đào tạo, kiểm tra và phân tích dữ liệu.

3.3.2. Nhận dạng người nói với Feedforward-DNN

Mạng feedforward truyền thống thường có ba lớp: lớp nhập, một lớp ẩn, và lớp xuất. Mỗi lớp gồm một hay nhiều neural. Số neural trong lớp nhập và số neural trong lớp xuất được xác định tương ứng với số biến đầu vào và số biến đầu ra của bài toán. Số neural của lớp ẩn do người thiết kế mạng quyết định dựa vào độ phức tạp của hàm ánh xạ do mạng thực hiện. Mỗi neural của lớp thứ i ($0<i<n$) liên kết với mọi neural của lớp thứ i+1, và các neural trong cùng lớp không liên kết với nhau. Kiểu kết nối giữa các lớp như thế này gọi là kết nối đầy đủ. Một mạng truyền thẳng là một mạng kết nối đầy đủ. Mỗi kết nối trong mạng được gắn một trọng số $w \in R$. Trọng số thể hiện mức độ quan trọng (hay độ mạnh) của dữ liệu đầu vào đối với quá trình xử lý thông tin tại mỗi neural. Mạng feedforward sử dụng thuật toán lan truyền ngược

để huấn luyện mạng. Mạng neural truyền thống hay mạng feedforward với nhiều lớp ẩn cho phép ta hiện thực "deep learning". Mạng feedforward nhiều lớp là một hàm toán học, ánh xạ một tập đầu vào thành các giá trị đầu ra. Hàm được hợp thành từ nhiều hàm đơn giản.Thuật toán lan truyền ngược được phát triển vào những năm 1960s – 1970s và được áp dụng vào mạng neural vào năm 1981. Vào cuối 1980s, phần lớn các ứng dụng chỉ thành công khi dùng feedforward với một lớp ẩn. Gần đây, khi khái niệm "deep learning" ra đời, mạng feedforward có kiến trúc lớn hơn nhiều so với các mạng truyền thống. Chúng có nhiều lớp ẩn hơn, mỗi lớp ẩn có nhiều nút ẩn hơn. Thiết kế này bám sát theo tư duy mô hình hóa thông tin trừu tượng. Cứ qua mỗi lớp ẩn, đặc trưng hay thông tin có ở lớp trước lại được biến đổi sang tầng biểu diễn cao hơn. Càng về sau, mức độ trừu tượng của thông tin càng cao. Khi đó một câu hỏi được đặt ra là bao nhiêu lớp ẩn và bao nhiêu nút ẩn cho mỗi lớp thì đủ. Câu trả lời vẫn luôn phụ thuộc vào qui mô, tính chất và kích thước của dữ liệu học. Người ta chọn các thông số này qua thực nghiệm.



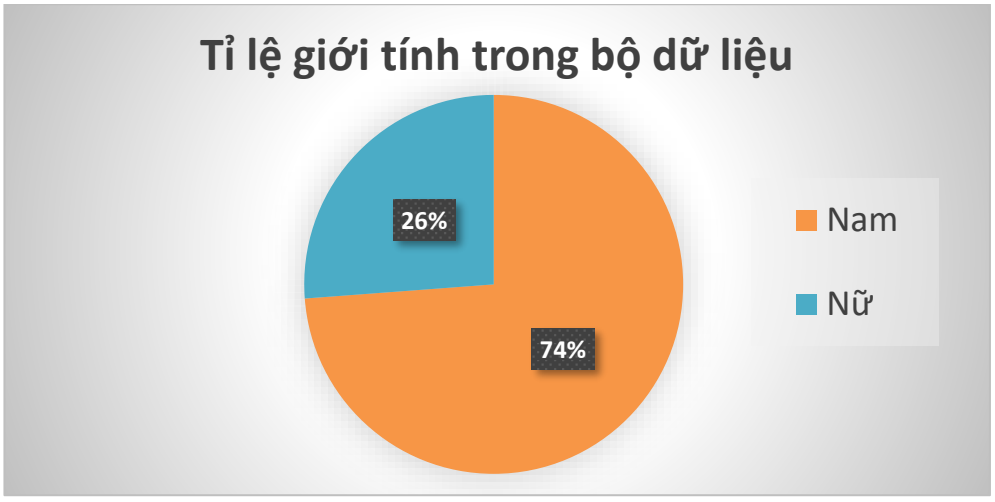
Hình 3.3: Cấu trúc mạng feedforward-DNN

CHƯƠNG 4. CÀI ĐẶT VÀ THỰC NGHIỆM

4.1. Dữ liệu thực nghiệm

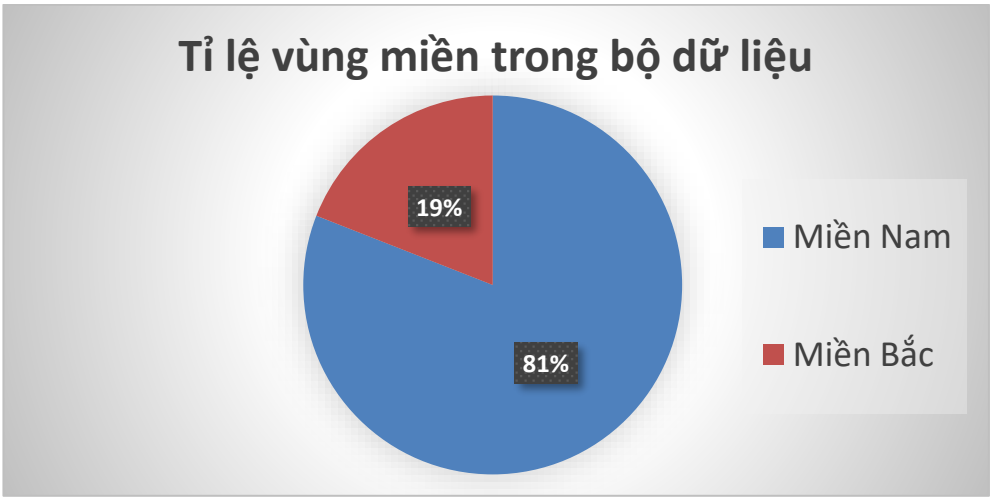
Bộ dữ liệu thực nghiệm của bài toán được tác giả thu thập tại nơi sinh sống thuộc địa bàn tỉnh Tây Ninh. Bộ dữ liệu được thu thập từ 42 người giới tính nam nữ với độ tuổi từ 12-52, chủ yếu đến từ miền Nam của Việt Nam. Nội dung của bộ dữ liệu là các file ghi âm giọng nói của các đối tượng tham gia khảo sát. Môi trường thu âm yên tĩnh, micro gần, với mỗi người tham gia ghi âm, tác giả đã chuẩn bị một nội dung văn bản tự do thu thập từ nguồn báo online để thỏa ngữ cảnh "độc lập văn bản" với thời lượng thu âm mỗi người trên dưới 10 phút và yêu cầu họ đọc đoạn nội dung một cách tự nhiên nhất. Kết quả thu được bộ dữ liệu thô bao gồm đoạn ghi âm và thông tin nhận dạng của người tham dự với các tiêu chí bao gồm: họ tên, giới tính, độ tuổi, vùng miền.

Chi tiết cụ thể thống kê bộ dữ liệu được thể hiện trong các sơ đồ dưới đây (hình 4.1):



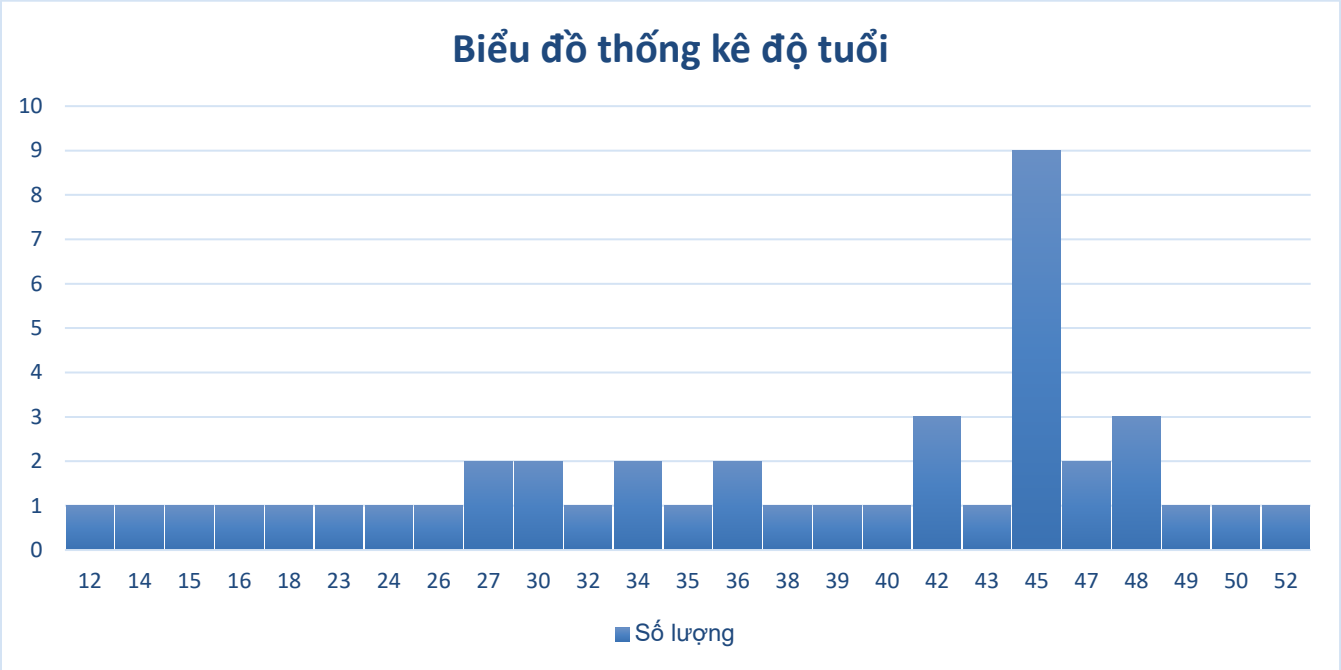
Hình 4.1: Biểu đồ hiển thị tỉ lệ giới tính trong bộ dữ liệu

Bộ dữ liệu có tổng cộng 31 người thuộc giới tính nam chiếm 74% và 11 người giới tính nữ chiếm 26%. Nhận thấy có sự chênh lệnh lớn về yếu tố giới tính trong bộ dữ liệu này.



Hình 4.2: Biểu đồ hiển thị tỉ lệ vùng miền trong bộ dữ liệu

Xét về yếu tố vùng miền (hình 4.2), đa số người đến từ đến từ miền Nam chiếm tỉ lệ 81% với tổng cộng là 34 người và 8 người đến từ miền Bắc chiếm chỉ 19%. Mặc dù Việt Nam là một quốc gia đa dân tộc, tuy nhiên, ta thấy bộ dữ liệu chưa có sự cân bằng và đa dạng về giọng nói ở nhiều địa phương, vùng miền khác nhau.



Hình 4.1: Biểu đồ thống kê độ tuổi của bộ dữ liệu

Từ biểu đồ ở Hình 4.3 ta thấy, với số lượng mẫu là 42 thì độ tuổi dao động trong khoảng từ 12-52 tuổi, trong đó số mẫu mỗi độ tuổi là khá nhỏ. Bên cạnh đó Hình 4.3 cũng cho ta thấy tỉ lệ phân bố về độ tuổi của người tham dự tập trung nhiều ở độ tuổi từ 42-48.

Nhận xét thấy bộ dữ liệu khá ít và có sự phân bố chưa đồng đều về các yếu tố giới tính, vùng miền, độ tuổi,… Sự phong phú về bộ dữ liệu là hạn chế, vì thế điều này có thể ảnh hưởng ít nhiều đến kết quả thực nghiệm của mô hình.

Về chất lượng các bản ghi âm, các bản thu được thực hiện trong môi trường khá tương đồng, giọng nói của các cá thể tham gia được ghi lại rõ ràng, rành mạch. Đây có thể là một yếu tố giúp mô hình có thể học tập dễ dàng hơn vì dữ liệu "sạch". Bảng 4.2 dưới đây mô tả thông tin chi tiết của một bản ghi âm điển hình.

Bảng 4.1: Thông tin chi tiết của một bản ghi âm

Format	Bit rate	Channel(s)	Sampling rate
Wave	768 kbps	1 channel	48 KHz

4.2. Kịch bản thực nghiệm

4.2.1. Chuẩn bị môi trường

Mã nguồn của luận văn với mô hình Feedforward-DNN được viết bằng ngôn ngữ Python kết hợp với framework Tensorflow. Bên cạnh đó mô hình HMM được xây dựng bởi bộ công cụ của HTK. HTK được cung cấp công khai và có thể được tải về để chỉnh sửa và sử dụng tùy ý. Cấu hình máy:

- CPU: Core i5, UBUNTU 20
- SSD: 80GB
- RAM: 8GB
- Software: Tensorflow (Training), HTK (GET MFCC)
  Language code: C#, Python, Perl

4.2.1. Chuẩn bị dữ liệu

Tiến hành xây dựng cấu trúc các thư mục, tệp tin như hướng dẫn của HTK, tiếp theo bắt đầu quá trình tiền xử lý dữ liệu đầu vào

1. **Tạo file listwavmfc từ folder wav**

```
perl pl/listwavmfc.pl wav txt/listwavmfc
```

2. Lấy MFCC với module HCopy

Rút trích đặc trưng sẽ chuyển đổi âm thanh ở dạng sóng sang định dạng MFCC được thực hiện bởi HCopy, nhằm chuyển đổi các tệp âm thành các tệp có phần mở rộng là .mfc.

```
bin/HCopy -C cfg/HCopy.cfg -S txt/listwavmfc
```

3. Tạo file train.scp

File train này sẽ bao gồm tất cả dữ liệu âm thanh của bài toán. Từ file gốc này tách ra thành 2 phần với tỉ lệ 7:3 để phục vụ cho việc huấn luyện và kiểm thử.

```
perl pl/mkTrainFile.pl wav txt/train.scp
```

4. Xây dựng file gram.txt

Để sử dụng các mô hình mà HTK cung cấp, ta phải định nghĩa nên kiến trúc cơ bản của trình nhận dạng (hay còn gọi là task grammar). Trong HTK, task grammar được viết trong tệp văn bản (thường đặt tên là gram.txt). File gram này sẽ chứa tên để định danh của người nói. Bấy nhiêu người nói là bấy nhiêu định danh. Sau đó thực hiện lệnh HParse để biên dịch task grammar (được mô tả trong gram.txt) thành task network lưu vào file wdnet.txt.

```
bin/HParse txt/gram.txt txt/wdnet.txt
```

5. Xây dựng file promts.txt

Hệ thống cần phải biết rằng HMM nào tương ứng với từng biến của grammar, vì thế ở đây chúng ta cần xây dựng thêm task dictionary

```
perl pl/mkPromt.pl wav txt/prompts.txt
```

6. Tạo label cho mô hình

```
perl pl/prompts2mlf.pl mlf/phones0.mlf txt/prompts.txt

perl pl/prompts2wlist.pl txt/prompts.txt txt/wlist.txt

bin/HDMan -m -w txt/wlist.txt -n ph/monophones0 txt/dict txt/dict.dct
```

4.2.3. Xây dựng mô hình và huấn luyện

Bước đầu tiên trong đào tạo **HMM** là xác định một mô hình nguyên mẫu. Các tham số của mô hình này không quan trọng, mục đích của nó là xác định cấu trúc liên

kết của mô hình. Một trong các cấu trúc liên kết tốt để sử dụng là 3-state left-right, ta tiến hành xây dựng lên nó trong file proto như sau:

```
~o <VecSize> 39 <MFCC_0_D_A>

~h "proto"

<BeginHMM>

 <NumStates> 5

 <State> 2 <NumMixes> 8

      <Mixture> 1 0.1

            <Mean> 39

              0.0 0.0 0.0 …

            <Variance> 39

              1.0 1.0 1.0 …

      <State> 4 <NumMixes> 8

      <Mixture> 1 0.1

            <Mean> 39

              0.0 0.0 0.0 …

            <Variance> 39

              1.0 1.0 1.0 …

      …

      <Mixture> 8 0.2

            <Mean> 39

              0.0 0.0 0.0 …

            <Variance> 39

              1.0 1.0 1.0 …

 <TransP> 5

  0.0 1.0 0.0 0.0 0.0
```

```
  0.0 0.6 0.4 0.0 0.0

  0.0 0.0 0.6 0.4 0.0

  0.0 0.0 0.0 0.7 0.3

  0.0 0.0 0.0 0.0 0.0

<EndHMM>
```

Trong đó mỗi vec-tơ có độ dài là 39. Con số này được tính bởi độ dài của vectơ tĩnh được tham số hóa (MFCC_0 = 13) cộng với hệ số delta (+13) cộng với hệ số gia tốc (+13). Tiếp theo, sử dụng HCompV để quét qua tập dữ liệu, tính toán giá trị trung bình và phương sai toàn cục và đặt tất cả Gaussian trong một HMM nhất định để có cùng phương sai và giá trị trung bình. Với danh sách tất cả các tệp đào tạo được lưu trữ trong file train_70.scp, thực thi câu lệnh sau:

```
bin/HCompV -C cfg/HCompV.cfg -f 0.01 -m -S txt/train_70.scp -M hmm0
proto
```

Model được lưu trữ trong thư mục hmm0 được ước tính lại bằng cách HERest

```
bin/HERest -C cfg/HERest.cfg -I mlf/phones0.mlf -t 250.0 150.0 1000.0
-S txt/train_70.scp -H hmm0/macros -H hmm0/hmmdefs -M hmm1 ph/monophones0
```

Tác dụng của việc này là tải tất cả các mô hình trong hmm0 được liệt kê trong danh sách các mô hình ở monophones0. Sau đó, chúng được ước tính lại bằng cách sử dụng dữ liệu được liệt kê trong train_70.scp và tập hợp mô hình mới được lưu trữ trong thư mục hmm1. Tiếp tục huấn luyện thêm cho mô hình.

```
bin/HERest -C cfg/HERest.cfg -I mlf/phones0.mlf -t 250.0 150.0 1000.0
-S txt/train_70.scp -H hmm1/macros -H hmm1/hmmdefs -M hmm2
ph/monophones0

bin/HERest -C cfg/HERest.cfg -I mlf/phones0.mlf -t 250.0 150.0 1000.0
-S txt/train_70.scp -H hmm2/macros -H hmm2/hmmdefs -M hmm3
ph/monophones0
```

Đến đây mô hình cơ bản đã được huấn luyện xong. Tiếp theo sẽ huấn luyện với **Feedforward-DNN**. Với đầu vào tương tự như HMM, thực hiện qua các bước sau:

1. **Chuyển file mfc ở dạng nhị phân sang dạng text**

```
perl pl/mkListMFCC.pl wav txt/log_mfcc.sh

sh ./txt/log_mfcc.sh
```

2. **Chuyển đổi mfc sang vec-tơ mean + vec-vơ varian**

```
perl ./pl/createMeanAndVar.pl wav dnn_mean
```

3. **Tạo file log chứa dữ liệu từ tất cả các file đặc trưng ở folder dnn_mean**

```
perl ./pl/createFeature.pl dnn_mean dnn/log.dnn
```

4. **Chia tập dữ liệu thành train và test với tỉ lệ 8:2**
5. **Tiến hành training**

Ở đây ta xây dựng bộ phân loại qua 3 lớp ẩn rồi đưa mô hình vào huấn luyện

```
# Define classifer

classifier = tf.estimator.DNNClassifier(

        feature_columns=my_feature_columns,

        hidden_units=[1024,512,256],

        n_classes=45,

        model_dir='mode_dnn/')

# Train the Model

classifier.train(

        input_fn=lambda:train_input_fn(train_x, train_y,
        args.batch_size),

        steps=args.train_steps)
```

Tiến hành gọi lệnh training với batch size là 100 và số lần chạy là tăng dần từ 1000-50000 để quan sát và có được kết quả tốt, phù hợp nhất.

```
python3 dnn.py --batch_size 100 --train_steps 30000 >
100_30000_log.txt
```

4.3. Đánh giá và so sánh các bộ dữ liệu với TrackEval

4.3.1. Độ đo đánh giá

Với HMM, HTK sử dụng HResults đọc một tập hợp các tệp chứa nhãn và so sánh chúng với các tệp phiên âm tham chiếu tương ứng. Dòng đầu tiên cung cấp độ chính xác ở cấp độ câu dựa trên tổng số tệp chứa nhãn giống với tệp phiên âm. Dòng thứ hai là độ chính xác của từ dựa trên sự trùng khớp DP giữa các tệp nhãn và phiên âm. Ở dòng thứ hai, H là số nhãn đúng, D là số lần xóa, S là số lần thay thế, I là số lần chèn và N là tổng số nhãn trong các tệp phiên âm đã xác định. Số phần trăm nhãn được nhận dạng chính xác và độ chính xác của nó được tính toán bởi công thức sau:

$$\%Correct = \frac{H}{N} \times 100\% \,;\, Accuracy = \frac{H - I}{N} \times 100\%$$

Với Feedforward-DNN, độ chính xác (hay còn gọi là accuracy) sẽ được sử dụng trong trường hợp này. Độ chính xác là một thước đo để đánh giá các mô hình phân loại. Nói chính xác hơn thì độ chính xác là một phần nhỏ của các dự đoán mà mô hình đã đúng. Về mặt hình thức, độ chính xác được định nghĩa là bằng tỉ lệ giữa số lượng dự đoán chính xác và tổng tất cả các dự đoán, công thức như sau:

$$Accuracy = \frac{Number\ of\ correct\ predictions}{Total\ number\ of\ predictions}$$

Trong khuôn khổ luận văn với hai mô trình HMM và Feedforward-DNN, ta đều cùng quan sát độ đo là accuracy để xem xét mô hình được xây dựng để nhận dạng giọng nói tiếng Việt có chính xác và hiệu quả hay không.

4.3.2. Thực nghiệm và so sánh

- **HMM**

Kiểm tra trên tập train: thực thi đoạn lệnh dưới đây ta sẽ thu được kết quả như như hình bên dưới.

```
bin/HVite -C cfg/HVite.cfg -H hmm3/macros -H hmm3/hmmdefs -S
txt/train_70.scp -i recout_train.mlf -w txt/wdnet.txt txt/dict.dct
txt/wlist.txt

bin/HResults -f -t -I mlf/phones0.mlf txt/wlist.txt recout_train.mlf
> result_train.mlf
```

```
---------------------- Overall Results -------------------------
SENT: %Correct=95.76 [H=3504, S=155, N=3659]
WORD: %Corr=95.76, Acc=95.76 [H=3504, D=0, S=155, I=0, N=3659]
================================================================
```

Hình 4.2: Kết quả thống kê trên tập huấn luyện

Kết quả từ hình 4.4 cho thấy, dòng bắt đầu bằng SENT: cho biết rằng trong số 3659 câu nói huấn luyện, có 3504 câu được nhận dạng chính xác chiếm tỉ lệ 95,76% và 155 câu bị nhận dạng nhầm qua người khác. Vì HMM ở trường hợp này được tiếp cận theo hướng nhận dạng ra người nói (không theo khuynh hướng nhận dạng văn bản), nên sẽ bỏ qua việc thống kê trên từ vựng.

Kiểm tra trên tập test

```
bin/HVite -C cfg/HVite.cfg -H hmm3/macros -H hmm3/hmmdefs -S
txt/test_30.scp -i recout_test.mlf -w txt/wdnet.txt txt/dict.dct
txt/wlist.txt

bin/HResults -f -t -I mlf/phones0.mlf txt/wlist.txt recout_test.mlf >
result_test.mlf
```

Sau khi chạy thống kê thu được kết quả như sau:

```
---------------------- Overall Results -------------------------
SENT: %Correct=93.04 [H=321, S=24, N=345]
WORD: %Corr=93.04, Acc=93.04 [H=321, D=0, S=24, I=0, N=345]
================================================================
```

Hình 4.3: Kết quả thống kê trên tập kiểm thử

Trong số 345 câu nói kiểm thử, có 321 câu được nhận dạng chính xác chiếm tỉ lệ 93,04% và có 24 câu bị nhận dạng sai (hình 4.5). Kết quả từ tập kiểm thử có chênh lệch một ít so với kết quả thống kê từ dữ liệu huấn luyện, nhận thấy rằng đây chính là kết quả chính xác cuối cùng của mô hình. Tỉ lệ chính xác cao, cho thấy mô hình xây dựng chạy nhận dạng tốt, phù hợp với bộ dữ liệu xây dựng.

❖ **Feedforward-DNN**

Với Feedforward-DNN ta cũng dựa trên độ chính xác (accuracy) để đánh giá mô hình. Tiến hành chạy thống kê mô hình trên bộ dữ liệu kiểm thử với câu lệnh sau:

```
eval_result = classifier.evaluate(

        input_fn=lambda:eval_input_fn(test_x, test_y,
```

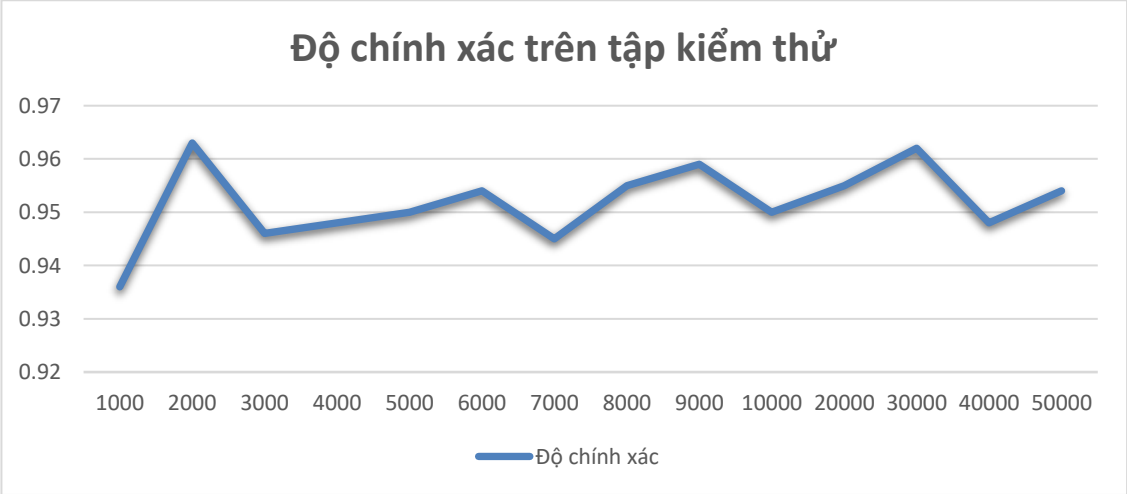
```
                                                    args.batch_size))

print('\nTest set accuracy: {accuracy:0.3f}\n'.format(**eval_result))
```

Tương ứng với các lần tăng dần chạy ta có được kết quả trong bảng 4.4 như sau:

Bảng 4.2: Độ chính xác của mô hình qua số lần chạy training

Lần chạy	Độ chính xác
1000	0.936
2000	0.963
3000	0.946
4000	0.948
5000	0.950
6000	0.954
7000	0.945
8000	0.955
9000	0.959
10000	0.950
20000	0.955
30000	0.962
40000	0.948
50000	0.954



Hình 4.4: Biến thiên độ chính xác theo số lần chạy mô hình

Quan sát biểu đồ ở hình 4.6 ở các lần chạy khác nhau, ta thấy mô hình ban đầu cho kết quả độ chính xác khá cao 93,6%. Đồng thời khi tăng số lần chạy mô hình lên dần độ chính xác cũng thay đổi nhưng cũng không đáng kể. Qua 14 lần chạy training, nhận thấy độ chính xác trung bình của mô hình khoảng 95%. So với độ chính xác từ mô hình HMM, Feedforward-DNN có lớn hơn đôi chút.

4.3.3. Xây dựng mô hình và huấn luyện

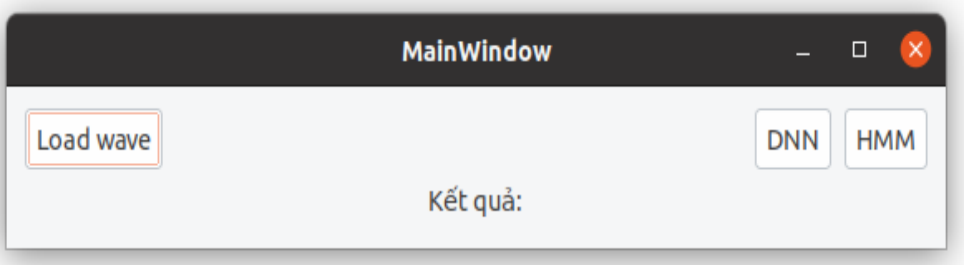
Kết quả từ việc chạy huấn luyện và kiểm thử mô hình cho thấy với HMM và Feedforward-DNN kết quả đạt khá cao 93.04%, 95% trên cùng bộ dữ liệu. Từ đó, nhận thấy với phương pháp học máy hiện đại, việc xây dựng mô hình nhanh chóng và dễ dàng, ít cấu hình hơn nhiều so với sử dụng bộ công cụ HTK. Bên cạnh đó, cách xây dựng phát triển mô hình cũng trực quan dễ hiểu, có thể chỉnh sửa để phát huy hơn nữa mà kết quả mang lại cũng khả thi hơn khi xem xét trên cùng tập dữ liệu đào tạo.

4.4. Chương trình demo

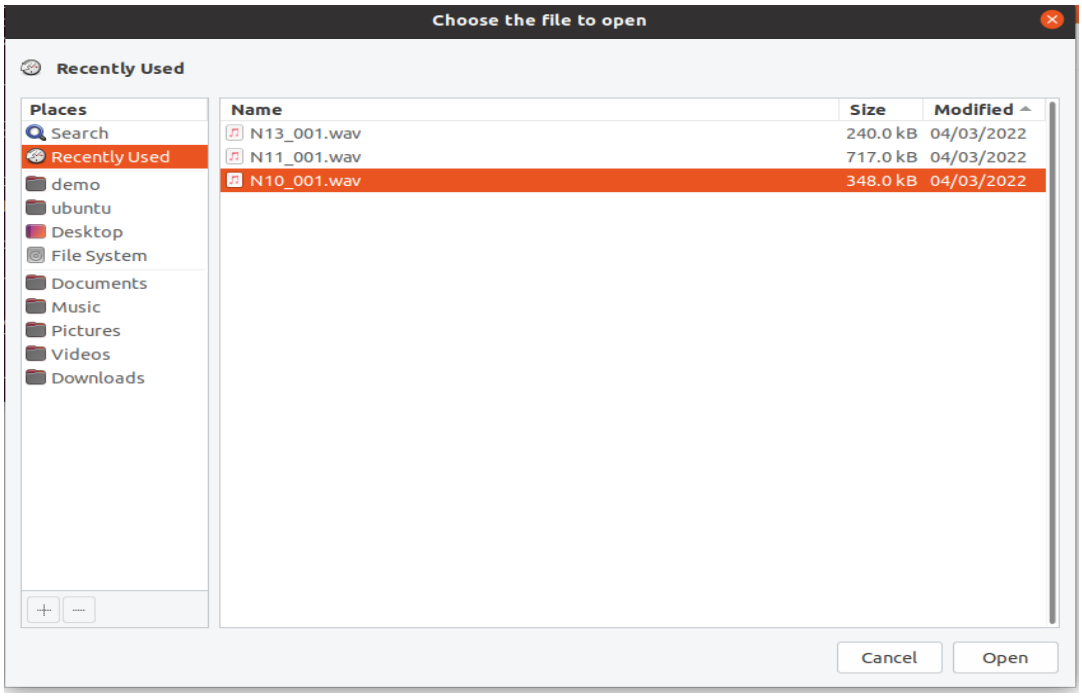
Nhận dạng người nói là một mảng nghiên cứu lớn, do thời gian hạn hẹp nên chỉ tập trung vào chứng minh khái niệm cho lĩnh vực này. Xây dựng chương trình demo chỉ mang tính minh họa và xác thực cho kết quả thực nghiệm, làm tiền đề cho xây dựng ứng dụng về sau. Cụ thể với hai mục tiêu:

1) Kiểm tra độ chính xác của mô hình: đã nhận dạng được đúng người với giọng nói hay chưa.
2) So sánh độ chính xác của hai mô hình HMM và Feedforward-DNN.

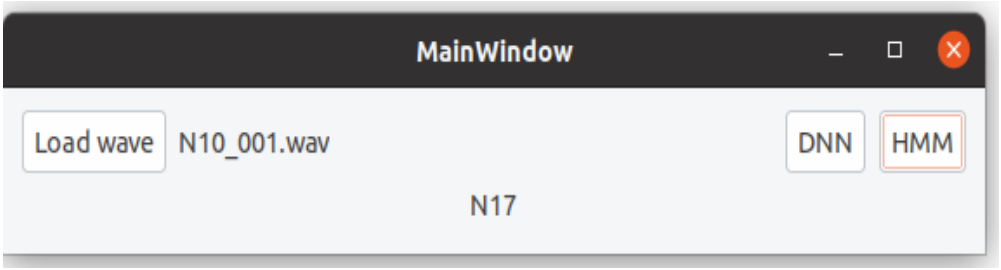
Xây dựng chương trình demo đơn giản để cho thấy khả năng dự đoán thực tế của mô hình, đầu tiên khởi động chương trình ta được giao diện như Hình 4.7. Giao diện đơn giản bao gồm nút Load wave bên trái dùng để mở giao diện chọn file âm thanh cần nhận dạng, nút DNN và HMM ở bên phải dùng để chọn mô hình dự đoán. Và cuối cùng kết quả mà mô hình dự đoán được sẽ được hiển thị ở phần giữa của chương trình. Hình ảnh minh họa chương trình Demo như bên dưới.



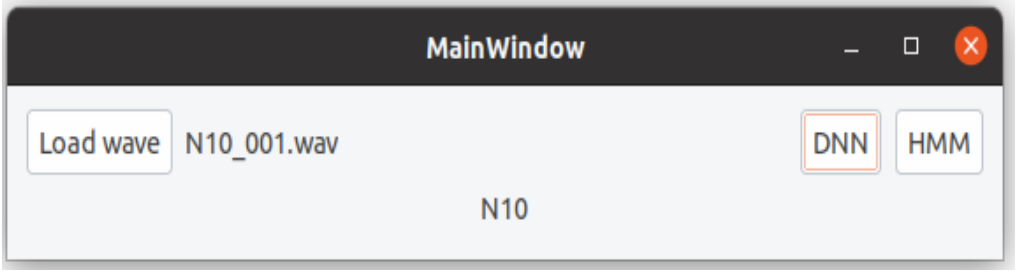
Hình 4.5: Giao diện chương trình demo



Hình 4.6: Chọn file âm thanh dể tiến hành nhận dạng



Hình 4.7: Trường hợp nhận dạng với HMM



Hình 4.8: Trường hợp nhận dạng với Feedforward-DNN

Kết quả từ chương trình chạy demo cho thấy, với HMM mô hình đã dự đoán chưa chính xác người nói. Cụ thể nhãn chính xác của file ghi âm là N10 nhưng mô hình dự đoán ra N17. Ngược lại, DNN đã dự đoán chính xác nhãn phân loại, chứng tỏ rằng mặc dù độ chính xác của hai mô hình có sự chênh lệnh khá nhỏ nhưng thực tế DNN lại hoạt động hiệu quả hơn.

CHƯƠNG 5. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

5.1. Kết quả nghiên cứu của đề tài

Với đề tài luận văn này, tác giả đã tập trung nghiên cứu các cách thức và ứng dụng học sâu để tiến hành nhận dạng người nói tiếng Việt trên bộ dữ liệu tự xây dựng của mình để đánh giá, so sánh sức mạnh của 2 mô hình. Cụ thể là nghiên cứu hai mô hình HMM và Feedforward-DNN, sử dụng bộ dữ liệu kết hợp với mô hình xây dựng kiểm chứng độ chính xác. Độ chính xác của mô hình xây dựng với HMM, Feedforward-DNN lần lượt là 93.04% và 95%. Ngoài ra việc áp dụng mô hình chạy dự đoán người nói với kết quả khả quan. Tuy nhiên với kết quả này thì luận văn vẫn còn có thể cải thiện thêm để có thể đạt được hiệu năng tốt hơn trên bộ data tốt hơn nữa.

5.2. Kết luận và hướng phát triển

Hạn chế luận văn

- Vì thời điểm thu thập dữ liệu này trùng với dịch Covid diễn ra phức tạp nên bộ dữ liệu chưa được đa dạng. Bộ dữ liệu vẫn còn nhỏ và mức độ phân bố rộng khắp về mặt độ tuổi, vùng miền chưa cao. Bên cạnh đó về chất lượng giọng nói của người tham dự chưa phong phú về biểu cảm, cảm xúc khi nói.
- Phần ứng dụng demo còn đơn giản do quỹ thời gian hạn hẹp.
- Đây mới chỉ là bài nghiên cứu, chưa áp dụng vào thực tế.

Vấn đề kiến nghị và hướng đi tiếp theo của nghiên cứu:

- Đưa kết quả nghiên cứu vào ứng dụng thực tế.
- Tìm thêm các cách xử lý tối ưu dữ liệu, tìm hoặc tự xây dựng hoặc tối ưu hóa mô hình, hiệu chỉnh hơn nữa độ chính xác của mô hình. Nghiên cứu tìm kiếm các phương pháp tiếp cận khác.
- Làm phong phú hơn cho bộ dữ liệu như: thu thập giọng nói ở nhiều vùng, tỉnh thành, mở rộng phạm vi độ tuổi, độ cao thấp của giọng nói, ảnh hưởng của môi trường (yên lặng, ồn ào,...).