# PHIẾU GIAO NHIỆM VỤ ĐỒ ÁN TỐT NGHIỆP

**1. Thông tin chung**

|  |  |
| --- | --- |
| Tên đề tài | Thu thập dữ liệu và thử nghiệm mô hình nhận dạng giọng nói |
| Họ và tên sinh viên: | Đoàn Đình Phùng |
| Điện thoại liên lạc: | 0961323468 |
| Email: | Doanphung0000@gmail.com |
| Lớp: | Công Nghệ Phần Mềm K59 |
| Hệ đào tạo: | Đại học chính quy |
| Đồ án tốt nghiệp được thực hiện tại: | Hà Nội |
| Thời gian làm ĐATN: | 2019 |

**2. Mục tiêu của ĐATN**

Thu thập dữ liệu sau đó từ dữ liệu của mình bắt đầu chuẩn hoá dữ liệu để phần mềm có thể học được nhiều dữ liệu phân tích dữ liệu chuẩn xác hơn .

Từ các dữ liệu thu thập thì ta chuẩn hoá gán nhãn dữ liệu để dữ liệu sai lệch sửa chữa cho chuẩn xác hơn .

**3. Các nhiệm vụ cụ thể của ĐATN**

- Thu thập dữ liệu

- Chuẩn hoá làm sạch

- Gán nhãn dữ liệu

- Thử nghiệm mô hình nhận dạng giọng nói

**4. Lời cam đoan của sinh viên:**

Tôi – Đoàn Đình Phùng – cam kết ĐATN là công trình nghiên cứu của bản thân tôi dưới sự hướng dẫn của thầy Nguyễn Thế Lộc

Các kết quả nêu trong ĐATN là trung thực, không phải là sao chép toàn văn của bất kỳ công trình nào khác.

|  |  |
| --- | --- |
|  | *Hà Nội, ngày 25 tháng 04 năm 2019*  Tác giả ĐATN  Phùng  Đoàn Đình Phùng |

**5. Xác nhận của giáo viên hướng dẫn về mức độ hoàn thành của ĐATN và cho phép bảo vệ:**

|  |  |
| --- | --- |
|  | *Hà Nội, ngày 25 tháng 04 năm 2019*  Cán bộ hướng dẫn  Lộc  Nguyễn Thế Lộc |

# TÓM TẮT NỘI DUNG ĐỒ ÁN TỐT NGHIỆP

Phần tóm tắt nội dung đồ án tốt nghiệp được biên soạn sau khi bản đồ án được hoàn chỉnh, tóm tắt đồ án phản ánh trung thực nội dung chủ yếu của đồ án. Yêu cầu của tóm tắt là ngắn gọn, cô đọng, nêu được cấu trúc của đề tài, nêu bật được những nội dung chính của đồ án, nhấn mạnh được những nội dung cần thiết, những kết luận rút ra sau khi nghiên cứu đề tài cùng với các giải pháp, đề xuất, kiến nghị (nếu có). Phần tóm tắt nội dung đồ án được viết thành 01 đoạn văn, dài không quá 300 từ.

# LỜI CẢM ƠN

Qua đây, Em xin chân thành cảm ơn thầy N**guyễn Thế Lộc** người đã trực tiếp hướng dẫn, chỉ bảo và giúp đỡ em thực hiện đồ án trong thời gian qua. Và em xin cảm ơn các thầy cô trong bộ môn Công Nghệ Phần Mềm khoa Công Nghệ Thông Tin đã chỉ dẫn nhiệt tình cho em qua các đợt bảo vệ tiến độ để em hoàn thành tốt được đồ án tốt nghiệp của bản thân mình.

# MỤC LỤC

[PHIẾU GIAO NHIỆM VỤ ĐỒ ÁN TỐT NGHIỆP I](#_Toc12524488)

[TÓM TẮT NỘI DUNG ĐỒ ÁN TỐT NGHIỆP III](#_Toc12524489)

[LỜI CẢM ƠN IV](#_Toc12524490)

[MỤC LỤC V](#_Toc12524491)

[DANH MỤC CÁC HÌNH VẼ VII](#_Toc12524492)

[DANH MỤC CÁC BẢNG BIỂU IX](#_Toc12524493)

[Danh mục các từ viết tắt X](#_Toc12524494)

[MỞ ĐẦU 1](#_Toc12524495)

[CHƯƠNG 1 : TỔNG QUAN 4](#_Toc12524496)

[1.1 Giới thiệu về nhận viết biết giọng nói 4](#_Toc12524497)

[1.1.1 Nhận diện giọng nói là gì ? 4](#_Toc12524498)

[1.1.2 Nhận dạng tiếng nói tự động 4](#_Toc12524499)

[1.1.3 Chuyển đổi văn bản thành tiếng nói 5](#_Toc12524500)

[1.1.4 Nhận dạngPhần mềm nhận diện giọng nói hoạt động như thế nào? 6](#_Toc12524501)

[1.1.5 Nhận diện giọng nói được dùng để làm gì ? 8](#_Toc12524502)

[1.1.6 Ưu và nhược điểm của công nghệ nhận diện giọng nói hiện nay 9](#_Toc12524503)

[CHƯƠNG 2 : CƠ SỞ LÝ THUYẾT VỀ XỬ LÝ TIẾNG NÓI 9](#_Toc12524504)

[2.1 Lý thuyết về âm thanh và tiếng nói 9](#_Toc12524505)

[2.1.1 Nguồn gốc âm thanh: 10](#_Toc12524506)

[2.1.2 Tần số của âm thanh 10](#_Toc12524507)

[2.1.3 Chu kì của âm thanh 10](#_Toc12524508)

[2.1.4 Tốc độ truyền âm 10](#_Toc12524509)

[2.1.5 Cường độ âm thanh 10](#_Toc12524510)

[2.1.6 Thanh áp 10](#_Toc12524511)

[2.1.7 Âm lượng 10](#_Toc12524512)

[2.1.8 Âm sắc 10](#_Toc12524513)

[2.1.9 Các tần số của âm thanh 11](#_Toc12524514)

[2.1.10 Cơ chế tạo lập tiếng nói của con người 11](#_Toc12524515)

[2.1.11 Mô hình lọc nguồn tạo tiếng nói 12](#_Toc12524516)

[2.1.12 Hệ thống nghe của người 13](#_Toc12524517)

[2.2 Xử lý tiếng nói 14](#_Toc12524518)

[2.2.1 Lấy mẫu tín hiệu 14](#_Toc12524519)

[2.2.2 Bộ lọc tín hiệu 15](#_Toc12524520)

[2.2.3 Dò tìm điểm cuối (end point ditection ) 17](#_Toc12524521)

[2.3 Rút trích đặc trưng 19](#_Toc12524522)

[2.3.1 Các bước rút trích đặc trưng 20](#_Toc12524523)

[ Làm rõ tín hiệu 20](#_Toc12524524)

[2.3.2 Các đặc trưng tiếng nói 26](#_Toc12524525)

[CHƯƠNG 3 : frame work kaldi sử dụng trong nhận dạng giọng nói và machine learning 42](#_Toc12524526)

[3.1 Kaldi là gì ? 42](#_Toc12524527)

[3.2 Tiền xử lý và khai thác tính năng 43](#_Toc12524528)

[3.3 Model 44](#_Toc12524529)

[3.4 Mã ma trận trong Kaldi 45](#_Toc12524530)

[3.5 BLAS ( basic linear algebra subprograms) 46](#_Toc12524531)

[3.6 Gói đại số tuyến tính ( LAPACK) 46](#_Toc12524532)

[3.7 Machine learning 47](#_Toc12524533)

[3.7.1 Machine learning không phải lúc nào cũng là chiếc hộp đen kỳ diệu 48](#_Toc12524534)

[3.7.2 Chuyển âm thanh thành số 48](#_Toc12524535)

[3.7.3 Lấy mẫu liệu có được chính xác ? 53](#_Toc12524536)

[3.7.4 Tiền xử lý dữ liệu mẫu âm thanh 53](#_Toc12524537)

[3.8 : Cách cài đặt kaldi sử dụng trên hệ điều hành ubuntu 54](#_Toc12524538)

[3.9 : Thu thập dữ liệu 55](#_Toc12524539)

[3.10 Gán nhãn dữ liệu 55](#_Toc12524540)

[3.11 Chuẩn hóa dữ liệu 56](#_Toc12524541)

[3.12 : Chạy thử nghiệm mô hình nhận dạng dọng nói bằng kaldi 58](#_Toc12524542)

[CHƯƠNG 4 : kết luận và hướng phát triển 60](#_Toc12524543)

[4.1 Kết luận 60](#_Toc12524544)

[4.2 Hướng phát triển 60](#_Toc12524545)

[PHỤ LỤC 62](#_Toc12524546)

# DANH MỤC CÁC HÌNH VẼ

[**Hình 1‑1** Mô hình chuyển từ văn bản sang tiếng nói 6](#_Toc11884291)

[**Hình 1‑2** Chuyển từ tiếng nói sang ngôn ngữ máy tính 7](#_Toc11884292)

[**Hình 1‑3** Google Assistant 8](#_Toc11884293)

[**Hình 2‑1** : Mô hình lọc tiếng nói 13](#_Toc11884294)

[**Hình 2‑2**: ví dụ về lấy mẫu tisnhhiejeu f(t) trên miền thời gian 15](#_Toc11884295)

[**Hình 2‑3 :** minh hoạt động của bộ lọc FIR 16](#_Toc11884296)

[**Hình 2‑4**  : minh họa hoạt động bộ lọc IIR 16](#_Toc11884297)

[**Hình 2‑5** : Dò tìm tín hiệu đặc trưng 17](#_Toc11884298)

[**Hình 2‑6** : dò tìm điểm cuôi dựa vào mức năng lượng 18](#_Toc11884299)

[**Hình 2‑7:** sơ đồ rút trích vector đặc trưng tổng quát 18](#_Toc11884300)

[**Hình 2‑8** : Sơ đồ rút trích đặc trưng chi tiết 19](#_Toc11884301)

[**Hình 2‑9:** phân đoạn tiếng nói thành các khung chồng lấp 20](#_Toc11884302)

[**Hình 2‑10** : :Âm/a/, cửa sổ chữ nhật 24](#_Toc11884303)

[**Hình 2‑11** : 512 điểm (45ms, trái) và 64 điểm (5.6 ms,phải) 24](#_Toc11884304)

[**Hình 2‑12** : Sự khác biệt giữa các cửa sổ tín hiệu 25](#_Toc11884305)

[**Hình 2‑13 :** đồ thị biểu diễn mối quan hệ giữa Mel và Hz 29](#_Toc11884306)

[**Hình 2‑14** : Phép biết đổi Fourier 29](#_Toc11884307)

[**Hình 2‑15:** Bộ lọc trên thang Mel 30](#_Toc11884308)

[**Hình 2‑16 :** Bộ lọc trên tần số thật 30](#_Toc11884309)

[**Hình 2‑17**: Minh họa các bước biến đổi MFCC 31](#_Toc11884310)

[**Hình 2‑18** : Sơ đồ xử lý LPC dùng trích đặc trưng tiếng nói 32](#_Toc11884311)

[**Hình 2‑19** : Hình dạng tín hiệu tiếng nói 39](#_Toc11884312)

[**Hình 2‑20** : kết quả trích F0 40](#_Toc11884313)

[**Hình 2‑21** : Kết quả sau khi lọc median 41](#_Toc11884314)

[**Hình 3‑1** : Cấu trúc cơ bản của Kaldi một số thành phần chính của Kaldi 42](#_Toc11884315)

[**Hình 3‑2 :** Quá trình thực hiện sử dụng I-vector 43](#_Toc11884316)

[**Hình 3‑3 :** Mô hình Acoustic 44](#_Toc11884317)

[**Hình 3‑4 :** Echo Dot 46](#_Toc11884318)

[**Hình 3‑5 :** truyền đoạn ghi âm vào mạng nowrron và đào tạo nó ra bản dịch 47](#_Toc11884319)

[**Hình 3‑6**: chuyển âm thanh thành số (1) 48](#_Toc11884320)

[**Hình 3‑7** : chuyển âm thanh thành số (2) 49](#_Toc11884321)

[**Hình 3‑8** : Chuyển âm thanh thành số (3) 50](#_Toc11884322)

[**Hình 3‑9** : Chuyển âm thanh thành số (3) 50](#_Toc11884323)

[**Hình 3‑10 :**  Chuyển âm thanh thành số (4) 51](#_Toc11884324)

[**Hình 3‑11** : Chuyển âm thanh thành số (5) 51](#_Toc11884325)

[**Hình 3‑12 :** Mẫu đồ thị 52](#_Toc11884326)

[**Hình 3‑13**: 320 mẫu âm thanh 53](#_Toc11884327)

[**Hình 3‑14** : Đồ thị giúp chúng ta có ước lượng xấp xỉ về âm thanh gốc 53](#_Toc11884328)

[**Hình 3‑15 :** Ứng dụng về gán nhãn dữ liệu 54](#_Toc11884329)

[**Hình 3‑16** : Ứng dụng về chuẩn hóa dữ liệu 56](#_Toc11884330)

[**Hình 3‑17 :** Chạy thử nghiệm mô hình nhận dạng giọng nói 58](#_Toc11884331)

# DANH MỤC CÁC BẢNG BIỂU

# Danh mục các từ viết tắt

|  |  |  |  |
| --- | --- | --- | --- |
| STT | Từ viết tắt | Từ tiếng Anh | Từ và nghĩa tiếng Việt |
| 1 | CSDL |  | Cơ sở dữ liệu |
| 2 | DB | Database | Cơ sở dữ liệu |
| 3 | ADC | Anlog-to –Digital Converter, | Chuyển đổi tương tư sang kỹ thuật số |
| 4 | TTS | Text to speech | Chuyển văn bản thành tiếng nói |
| 6 | BLAS | Basic linear Algebra Subprograms | Chương trình con đại số tuyến tính cơ bản |
| 7 | LAPACK | Linear Algebra Package | Gói đại số tuyến tính |
| 8 | VUS | voice , Unvoice and Silence |  |
| 9 | COR | Autocorrelation coefficients | Hệ số tư tương quan |
| 10 | LPC | Linear Prediction coefficients | Hệ số dự đoán tuyến tính |
| 11 | PARCOR | Partial Correlation coefficients | Các hệ số tự tương quan một phần |
| 12 | LAR | Log Area Ratio coefficients | Hệ số phản xạ |
| 13 | PLP | Perceptional Linear Prediction | Dự đoán tuyến tính |
| 14 | LSP | Liner Spectrum Pairs | Cặp phổ vạch |
| 15 | MFCC | Mel Frequency Cepstral coefficients | Hệ số cepstral tần số Mel |
| 16 | FFT | Fast Fourie Transform | Biến đổi Fourie nhanh |
| 17 | FCT | Fast Cosine Transform | Biến đổi Cosine nhanh |
| 18 | CLIP | center clipping pitch detector | Phương pháp cắt tâm |
| 19 | AMDF | Average Magnitude Difference Function | Chức năng chênh lệch cường độ trung bình |

# MỞ ĐẦU

1. Tổng quan tình hình nghiên cứu thuộc lĩnh vực của đề tài

Hiện nay,trong thời đại phát triển của công nghệ thông tin.Việc giao tiếp giữa người với máy , máy với máy…. Luôn là vấn đề được quan tâm thì việc trao đổi thông tin bằng giọng nói có vai trò rất quan trọng .Máy tính đã ra đời giúp con người giải quyết công việc nhanh gọn và hiệu quả hơn .Tuy niên.các máy tính ngày nay chỉ nhận thông tin từ con người qua các thiết bị như bàn phím, chuột,bút ….Mặc dù tốc độ xử lý của máy tính càng ngày càng được cải thiện đáng kể ,song tốc độ tạo lập thông tin của các thiết bị này vẫn còn thấp.Con người mong muốn máy tính ngày càng mạnh hơn , thông minh hơn và một trong số các yêu cầu đó là máy tính trong tương lai phải tương tác với người sử dụng bằng tiếng nói tự nhiên. Đây sẽ là một bước tiến lớn nhằm nâng cao sức mạnh của máy tính, đồng thời cũng tăng tốc độ truyền đạt thông tin giữa máy tính và con người

2. Tính cấp thiết, ý nghĩa khoa học và thực tiễn của đề tài

Xử lý tiếng nói trở thành một trong những lĩnh vực quan trọng trong xu hướng phát triển công nghệ của xã hội hiện nay.Đặc biệt , khi công nghệ thông tin ngày càng phát triển thì các ứng dụng của xử lý tiếng nói ngày càng trở nên cấp thiết hơn. Mục đích của những nghiên cứu trong lĩnh vực xử lý tiếng nói là làm cho việc tương tác giữa người và máy ngày càng hiệu quả tự nhiên hơn.

Hiện nay trên thế giới các công nghệ xử lý tiếng nói đã phát triển ,các hệ thống ứng dụng xử lý tiếng nói đã được sử dụng ở nhiều nơi, độ chính xác của các hệ thống ứng dụng xử lý tiếng nói ngày càng được cải thiện. Các ứng dụng của lĩnh vực xử lý tiếng nói rất phổ biến: nhận dạng tiếng nói , tổng hợp tiếng nói , xác thực người nói qua giọng nói và các thành tựu của chung được áp dũng vào nhiều lĩnh vực trong thực tế

Trên thế giới đã có rất nhiều hệ thống nhận dạng tiếng nói tiếng Anh và đang được ứng dụng rất hiệu quả như : Via Voice của IBM ,Spoken Toolkit của CSLU ( Central of Spoken Language Under-standing) , Speech Recognition Enigine của Microsoft , Hidden Markov Model toolkit của đại học Cambridge , CMU Sphinx của đại học Carnegie Mellon ,… ngoài ra , một số hệ thống nhận dạng tiếng nói tiếng Pháp , Đức , Trung Quốc ,… cũng khá phát triển.

Ở Việt Nam , nhận dạng tiếng nói vẫn là một lĩnh vực khá mới mẻ.Đến nay tuy đã có nhiều nghiên cứu vê nhận dạng tiếng nói tiếng Việt và đã đạt được một số thành tựu , nhưng nhìn chung vẫn chưa đạt được kết quả cần thiết để có thể tạo ra các sản phẩm mang tính ứng dụng cao. Có thể kể đến các công trình sau :

AILab : Đây là công trình được phòng thí nghiệm Trí tuệ Nhân Tạo – AILab thuộc Đại học Khoa học tự nhiên tạo ra dựa trên các công nghệ tiên tiến nhất về nhận dạng và tổng hợp tiếng nói để đáp ứng nhu cầu của người dùng. Dựa trên công nghệ xử lí tiếng nói tiếng Việt, AILab đã xây dựng phần mềm iSago chuyên hỗ trợ tìm kiếm thông tin qua tiếng nói. Thông qua ứng dụng phần mềm người sử dụng có khả năng hỗ trợ giao tiếp với điện thoại di động trực tiếp bằng lời nói . Từ đó người sử dụng tìm kiếm thông tin nhà hàng , Quán Bar, Cafe. Khi người dùng đặt câu hỏi bằng tiếng nói iSago sẽ truyền nội dung truy vấn này về server để xử lý và gửi lại kết quả tìm kiếm , dạng một danh sách : tên nhà hàng, địa chỉ. Pần mềm này cũng cho phép người dùng hiển thị địa chỉ tìm được dạng bản đồ hoặc nghe đọc địa chỉ trực tuyến bằng công nghệ tổng hợp giọng nói . Phần mềm được cung cấp miễn phí tại địa chỉ [WWW.ailab.hcmus.edu.vn](http://WWW.ailab.hcmus.edu.vn)

-Vietvoice: đây là phần mềm của một người dân Việt Nam ngụ tại Canada. Phần mềm có khả năng nói tiếng việt từ các tâp tin. Để chạy được chương trình ,cần cài đặt Microsoft Visual C++ 2005 Redistributable Package(x86) . Đối với người khiếm thị , phần mềm này cho phép sử dụng cách gõ tắt (nhận nút Ctrl và một chữ ) để chọn lựa một trong các tính năng hiển thị trên màn hình. Người dùng có thể cập nhật từ điểm các chữ việt tắ và các từ ngữ tiếng nước ngoài .

- Vspeech: Đây là một phần mềm điều khiển máy tính bằng giọng nói do một nhóm sinh viên Đại học Bách Khoa TP. HCM viết . Phần mềm sử dụng thư viện Microsoft Speech SDK để nhận dạng tiếng Anh nhưng được chuyển thành tiếng Việt. Nhóm đã khá thành công với ý tưởng này, do sử dụng lại thư viện nhận dạng engine nên thời gian thiết kế rút ngắn lại mà hiệu quả nhận dạng khá tốt . Phần mềm Vspeech có các lệnh gọi hệ thống đơn giản như gọi thư mục My Computer , nút Start,… Phiên bản mới nhất có tương tác với MS Word 2003, lướt web với trình duyệt Internet Explorer. Không có các chức năng tùy chỉnh lệch và gọi tắt các ứng dụng . Phần mềm chạy trên nền Windows XP, Microphone và card âm thanh sử dụng tiêu chuẩn thông thường.

Tuy nhiên việc ứng dụng nhận dạng giọng nói vào điều khiển máy tính còn rất nhiều hạn chế. Một số sản phẩm của nước ngoài về nhận dạng giọng nói Tiếng Việt như : Nuance (Dragon dictation và Dragon Search). Google search … Ở Việt Nam thì hầu như chỉ mới có bộ phần mềm Vspeech của nhóm sinh viên trường đại học Bách Khoa TP HCM, nhìn chung các phần mềm chưa được sử dụng thực tế vì chưa đạt trên 100 từ. Phầm mềm Vspeech được phát triển từ mã nguồn mở Microsoft Speech SDK nhận dạng tiếng Anh, thông qua dữ liệu , phương thức trung gian, việc nhận dạng được chuyển trong Vspeech để nhận biết tiếng Việt.

Lĩnh vực nhận dạng tiếng nói đã và đang tiếp tục được nghiên cứu , phát triển và các ứng dụng của nói ngày càng trở nên phổ biến và quan trọng . VÌ vậy nhận dạng giọng nói và các ứng dụng của nó đã trở thành đề tài được nhiều nhà nghiên cứu cũng như học sinh , sinh viên nghiêm ứu và phát triển.

Nội dung đồ án chia làm …

**Chương 1 : Giới thiệu về nhận biết giọng nói**

Chương này giới thiệu qua về định nghĩa nhận biết giọng nói , cách hoạt động giọng nói trong thực tế

**Chương 2 : Cơ sở lý thuyết về xử lý tiếng nói**

Chương này giới thiệu tổng quan về xử lý tiếng nói , tập trung giới thiệu tổng quan về lý thuyết âm thanh và tiếng nói,giới thiệu về xử lý tiếng nói,nhận dạng tiếng nói và cơ bản về nhận dạng tiếng Việt với những ưu nhược điểm

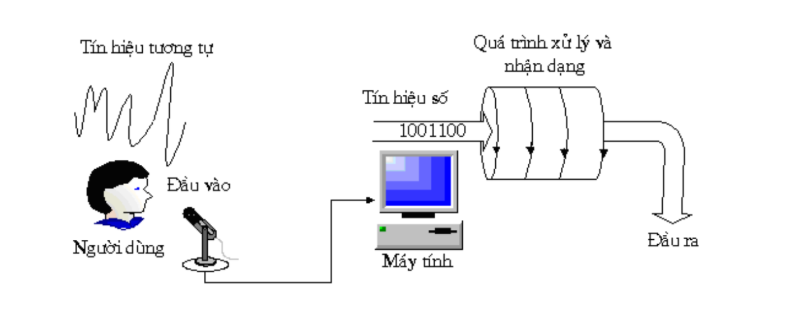
**Chương 3 : Frame work kaldi sử dụng trong nhận dạng giọng nói và machine learning**

Cài đặt frame work kaldi , thu thập dữ liệu , gán nhãn dữ liệu , chuẩn hóa dữ liệu , sử dụng mô hình nhận dạng giọng nói .

# : TỔNG QUAN

## Giới thiệu về nhận viết biết giọng nói

### Nhận diện giọng nói là gì ?



**Hình 1‑1** : Mô hình nhận dạng giọng nói

Nhận diện giọng nói là một bộ máy hoặc hệ thống có khả năng nhận và dịch các lệnh thu được từ giọng con người. Nhận dạnng giọng nói gồm 2 thuật ngữ Voice recognition và Speech recognition .

* Voice recognition liên quan đến việc xác định giọng nói chính xác của một cá nhân nào đó , tương tự một phương pháp nhận diện sinh trắc học
* Speech recognition là việc xác định những từ ngữ trong câu nói rồi dịch chúng sang ngôn ngữ máy tính

### Nhận dạng tiếng nói tự động

Nhận dạng tiếng nói là một quá trình nhận dạng mẫu , với mục đích là phân lớp (classify) thông tin đầu vào là tín hiệu tiếng nói thành một dãy tuần tự các mẫu đã được học trước đó và lưu trữ trong bộ nhớ . Các mẫu là các đơn vị nhận dạng , chúng có thể là các từ hoặc các âm vị. Nếu các mẫu này là bát biến và không thay đổi thì công việc nhận dạng tiếng nói trở nên đơn giản bằng cách so sánh dữ liệu tiếng nói cần nhận dạng với các mẫu đã được học và lưu trữ trong bộ nhớ.Khó khăn cơ bản của nhận dạng tiếng nói đó là tiếng nói luôn được biến thiên theo thời gian và có sự khác biệt lớn giữa tiếng nói của những người nói khác nhau, tốc độ nói , ngữ cảnh và môi trường âm học khác nhau,

Xác định những thông tin biến thiên nào của tiếng nói là có ích và những thông tin nào là rất quan trọng. Đây là một nhiệm vụ rất khó khăn mà ngay cả với các kỹ thuật xác suất thống kê mạnh cũng khó khăn trong việc tổng quát hóa từ các mẫu tiếng nói những biến thiên quan trọng cần thiết trong nhận dạng tiếng nói.

Các nghiện cứu về nhận dạng tiếng nói dựa trên ba nguyên tắc cơ bản :

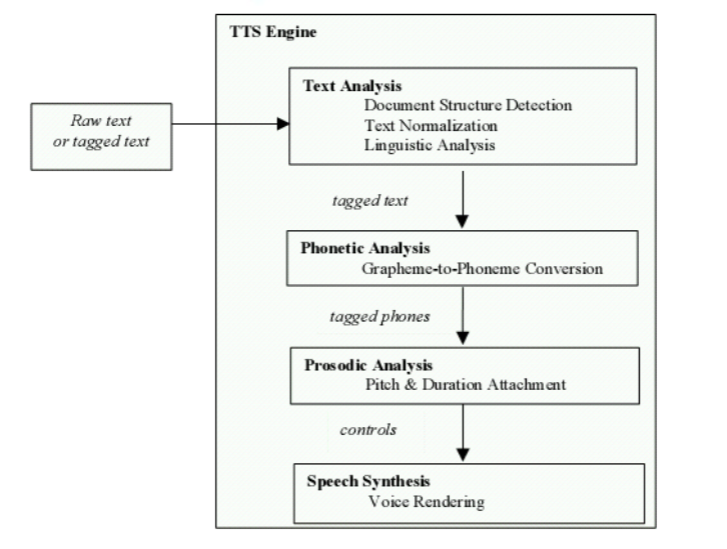
* Tín hiệu tiếng nói được biểu diễn chính xác bởi các giá trị phổ thông trong một khung thời gian ngắn (short – term amplitude spectrum). Nhờ vậy ta có thể trích ra các đặc điểm tiếng nói từ những khoảng thời gian ngắn và dùng các đặc điểm này làm dữ liệu để nhận dạng tiếng nói.
* Nội dung của tiếng nói được biểu diễn dưới dạng chữ viết , là một dãy các ký hiệu ngữ âm. Do đó ý nghĩa của một phát âm được bảo toàn khi chúng ta phiên âm phát âm thành dãy các ký hiệu ngữ âm.
* Nhận dạng tiếng nói là một quá trình nhận thức. Thông tin về ngữ nghĩa (semantics) và suy toán (pragmatics) có giá trị trong quá trình nhận dạng tiếng nói, nhất là khi thông tin về âm học là không rõ ràng

### Chuyển đổi văn bản thành tiếng nói

Các hệ thông chuyển đổi văn bản thành giọng nói có thể được xem như là hệ thống mã hóa tiếng nói cho phép lựa chọn kiểu cách nói, tốc độ , cường độ và các hiệu ứng. Hệ thống chuyển văn bản thành tiếng nói (Text- to-Speech) là một hệ thống có thể sinh ra tiếng nói gần giống với con người từ các văn bản được đưa vào (còn được gọi là hệ thống tổng hợp tiếng nói) Sự chuyển đổi các từ duowisi dạng viết sang tiếng nói là một công việc khó khăn vì hệ thông TTS cần dữ liệu từ vựng rất lớn và nhiều ngữ điệu của âm thanh

Các thành phần cơ bản của một hệ thống chuyển đổi văn bản thành tiếng nói :

* Bộ phân tích văn bản : chuẩn hóa văn bản sang dạng tích hợp cho hệ thông TTS
* Bộ phân tích ngữ âm chuyển đổi văn bản đã được xử lý thành dãy các âm tương ứng sau đó được phân tích ngữ điệu để xác định trọng âm , ngắn nhịp, thời gian ….
* Cuối cùng , bộ tổng hợp tiếng nói nhận các tham số đầuv ào từ dãy âm đã xử lý đầy đủ



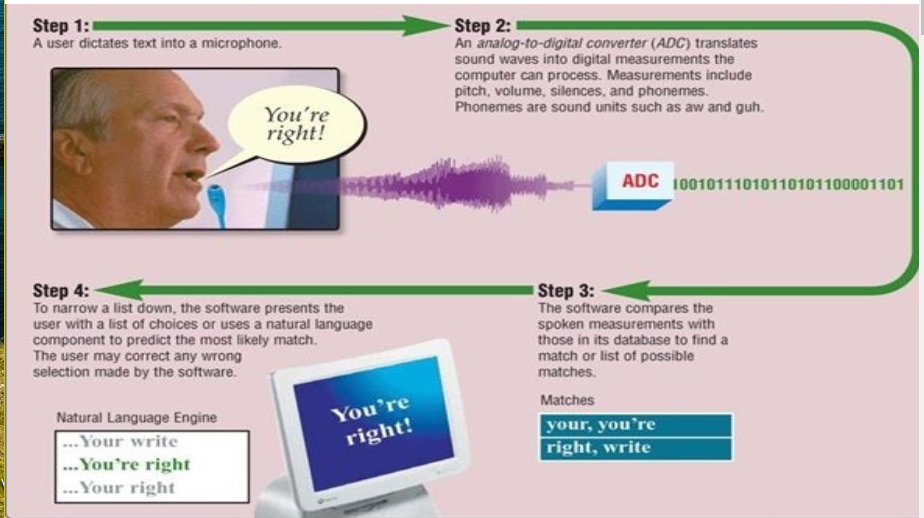
**Hình 1‑1** Mô hình chuyển từ văn bản sang tiếng nói

* Thành phần phân tích văn bản :
* Xác định cấu trúc tài liệu chuyển đổi ký hiệu,phân tích cấu trúc ngôn ngữ
* Chuyển đổi các ký hiệu sang dạng chuẩn.
* Chuyển đổi xác số sang dạng chữ tương ứng
* Phân tích khoảng trống ,dấu châm câu để xác định câu trúc ngôn ngữ
* Chuyển đổi các từ đã chuẩn hóa sang các âm vị tương ứng (với thông tin như trọng âm và thời gian phát âm)

### Nhận dạngPhần mềm nhận diện giọng nói hoạt động như thế nào?

Để chuyển giọng nói sang văn bản hoặc câu lệnh máy tính , hệ thống phải thực hiện một quá trình gồm nhiều bước phức tạp.Khi nói, bạn sẽ tạo ra những rung dộng trong không khí.Bộ chuyển đổi tín hiệu sẽ tượng tự sang số (Anlog-to –Digital Converter,ADC) chuyển các sóng tương tự(analog) này thành dữ liệu mà máy tính có thể hiểu được.

Để làm điều này , hệ thống thu thấp các mẫu(hoặc số hoá) âm thanh bằng các đo chính xác sóng âm ở các khoảng thời gian gần nhau, sau đó lọc âm thanh đã được số hoá để loại bỏ tiếng ồn, đôi khi tách chúng ra thành các dải tần số khác nhau. Nó cũng “bình thường hoá” hoặc tinh chỉnh âm thanh đến một mức âm lượng không thay đổi hoặc sắp xếp theo thời gian.Không phải lúc nào con người cũng nói với tốc độ như nhau nên âm thanh phải được điều chỉnh phù hợp với tốc độ mà âm thanh mẫu được ghi nhận trong bộ nhớ máy tính.



**Hình 1‑2** Chuyển từ tiếng nói sang ngôn ngữ máy tính

Tiếp theo tín hiệu được chia thành nhiều phần nhỏ ( thời gian khoảng vài phần trăm giây, thậm chí là phần ngàn giây trong trường hợp có phụ âm cuối khó phân biệt như “p” hoặc “t”).Chương trình sau đói đặt những phần âm thanh này vào các âm vị có sẵn trong ngôn ngữ thích hợp.

Theo Wikipedia, âm vị là phân đoạn nhỏ nhất của âm thanh dùng để cấu tạo nên sự phân biệt giữa các cách phát âm.Do đó , âm vị là một nhóm các âm thanh với sự khác biệt tương đối nhỏ cùng đảm nhận một chức năng ý nghĩa tuỳ theo người nói và phương ngữ.

Trong tiếng Anh có khoảng 40 âm vị khác nhau.Ví dụ, mặc dù hầu hết người bản ngữ không nhận ra,tỏng đa số các ngôn ngữ , âm k tỏng mỗi tư được phát âm thực sự khác biệt nhau.Ví dụ k tỏng kit được ký âm [kʰ] và trong k trong skill được ký âm /k/ . Trong một số ngôn ngữ , một ký tự đại diện cho một âm vị, nhưng trong một số ngôn ngữ khác nhau như tiếng Anh, sự tương ứng này ít khi chính xác. Ví dụ trong tiếng Anh ký tự sh đại diện cho /f/, trong khi k cà c đều đại diện cho âm /k/ (trong kit và cat).

### Nhận diện giọng nói được dùng để làm gì ?

Chuyển thành văn bản hoặc điều khiển máy là tác dụng phổ biến nhất của nhận diện giọng nói, tuy nhiên công nghệ này còn mang đến nhiều hứa hẹn cho người khuyết tật . Một số ứng dụng như DriveSafe.ly có thể đọc tin nhắn đến và email cho những người khiếm thị, một số ứng dụng khác còn cho phép tìm kiếm hoặc nhắn tin bằng giọng nói (chuyển thành văn bản) dành cho người bị liệt.

Nhận diện giọng nói còn được sử dụng rất nhiều trong lĩnh vực trí tuệ nhân tạo (Artificial Intelligence) . Siri hay Google Assistant là những ví dụ điển hình.



**Hình 1‑3** Google Assistant

Google Assistant là hệ thống trợ lý cá nhân ảo được phát triển bởi Google, ra mắt tháng 5/2016 .Khác với Google Now, Google Assistant có thể tương tác , nói chuyển với người dùng.

### Ưu và nhược điểm của công nghệ nhận diện giọng nói hiện nay

**Ưu điểm** :

* **Khả năng truy cập** : với những người khuyết tật không thể sử dụng chuột hay bàn phím,họ có thể dùng giọng nói để hệ thống chuyển thành văn bản ,giúp nhập liệu hay điều khiển một các dễ dàng.
* **Kiểm tra chính tả** : Bạn có thể truy cập vào các công cụ chỉnh sửa tương tự một giải pháp xử lý văn bản chuẩn. Tất nhiên , mọi thứ không thể chính xác 100% nhưng phần mềm có thể nhận diện và xử lý phần lớn lỗi chính tả , ngữ pháp.
* **Tốc độ nhanh:** Phần mềm có thể năm bắt giọng nói của bạn với tốc độ nhanh hơn so với khi nhập bằng bàn phím , vì vậy tốc độ khi nhập liệu bằng giọng nói sẽ cải thiện đáng kể.

**Nhược điểm** :

* **Thiết lập và “dạy ”**: Mặc dù chúng đều hứa hẹn có thể hoạt động sau vài phút thiết lập , nhưng thực sự quá trình ghi nhận, làm quen với giọg nói , âm điệu và tốc độ nói của bạn có đôi chút phức tạp và tốn thời gian, Một số phần mềm còn bắt bạn nói lại . thậm chí không thể nhận diện được bạn đang nói gì.
* **Chưa thực sự ổn định**: Việc đang nói mà bị ngắt giữa chừng có thể khiến bạn “ cụt hứng” , đặc biệt nó có thể bối rối khi bạn lên xuống giọng hay bỗng dưng nói nhỏ lại.
* **Kho từ vựng hạn chế** : Bạn phải sẵn sàng chấp nhận trường hợp phần mềm xử lý quá lâu vì những từ vừa nói không nằm trong từ điển của nó.Đó là điều đang được cải tiến từng ngày từng giờ.

# : CƠ SỞ LÝ THUYẾT VỀ XỬ LÝ TIẾNG NÓI

## Lý thuyết về âm thanh và tiếng nói

### Nguồn gốc âm thanh:

Âm thanh là do vật thể dao động cơ học mà phát ra. Âm thanh phát ra dưới dạng sóng âm. Sóng âm là sự biết đổi các tính chất của môi trường đàn hồi khi năng lượng âm truyền qua. Âm thanh truyền được đến tai người là do môi trường dẫn âm. Sóng âm có thể truyền được trong chất rắn, chất lỏng, không khí. Có chất dẫn âm rất kém gọi là chất hút âm như : len ,da, chất xốp… Sóng âm không thể truyền trong môi trường chân không. Khi kích thích dao động âm trong môi trường không khí thì những lớp khí sẽ bị nén và dãn. Trạng thái nén dãn lần lượt được lan truyền từ nguồn âm dưới dạng sóng dọc tới nơi thu âm. Nếu cường độ nguồn âm càng lớn thì âm thanh truyền đi càng xa.

### Tần số của âm thanh

Là số lần dao động của phần tử khí trong một giây.Đơn vị là Hz,kí hiệu là : f

### Chu kì của âm thanh

Là thời gian mà âm thanh đó thực hiện một dao động hoàn toàn.Đơn vị là thời gian ký hiệu là T

### Tốc độ truyền âm

Là tốc độ truyền năng lượng âm từ nguồn tới nơi thu. Đơn vị là m/s.Tốc độ truyền âm trong không khí ở nhiệt độ từ 0 – 200 C thường là 331 – 340 m/s

### Cường độ âm thanh

Là năng lượng được sóng âm truyền trong một đơn vị thời gian qua một đơn vị diện tích đặt vuông góc với phương truyền âm.

### Thanh áp

Là lực tác dụng vào tai người nghe hoặc tại một điểm nào đó của trường âm thanh. Đơn vị : 1pa=1 N/m2 hoặc 1bar = 1dyn/cm2

### Âm lượng

Là mức độ to nhỏ của nguồn. Đơn vị là W

### Âm sắc

Trong thành phần của âm thanh, ngoài tần số cơ bản còn có các sóng hài,số lượng sóng hài biểu diễn sắc thái của âm. Âm sắc là một đặc tính của âm nhờ đó mà ta phân biệt được tiếng trầm , bổng khác nhau, Phân biệt được tiếng nhạc cụ, tiếng nam nữ , tiếng người này với người khác.

### Các tần số của âm thanh

Tần sốc ơ bản F0 là tần số giao động của dây đàn. Tần số này phụ thuốc vào giới tính và độ tuổi . F0 của nữ thường cao hơn của nam, F0 của người trẻ thường cao hơn của người già . Thường với giọng của nam, F0 nằm trong khoảng từ 80 – 20 Hz, với giọng của nữ , F0 trong khoảng 10 – 500 Hz. Sự biến đổi của F0 có tính quyết định đến thanh điệu của từ cũng như ngữ điệu của câu .

Công suất của tiếng nói , khi nói to nhỏ cũng khác nhau. Khi nói thầm công suất 10-3mW, nói bình thường 10mW, nói to 103mW

### Cơ chế tạo lập tiếng nói của con người

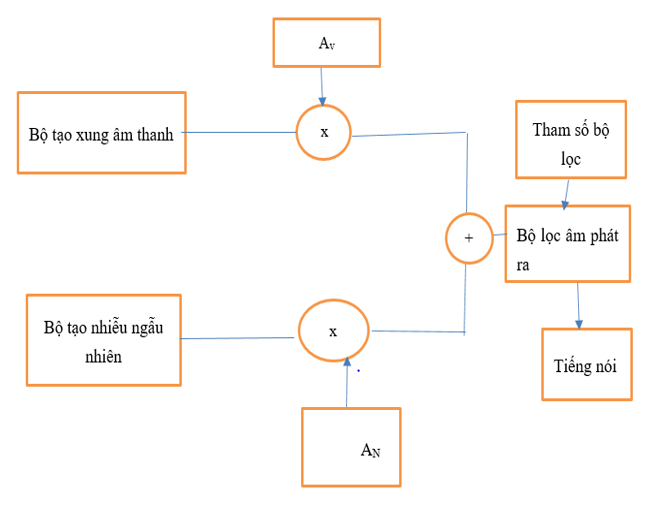
Các cơ quan phát âm của con người chủ yếu gồm phổi, khí quản ,thanh quả, bộ phận mũi và miệng . Thanh quản có hai nếp gấp gọi là dây thanh âm, dây thanh âm sẽ rung khi luồng không khí điq ua khe thanh môn và khe giữa haid ây thanh âm. Bộ phận miệng là một ống âm không đều. Bộ phận mũi cũng là một ống âm học không đều có diện tích và chiều dài cố định , bắt đầu từ lỗ mũi đến vòm miệng mềm.

Quá trình tạo ra âm phi mũi; vòm miệng mềm ngăn chặn bộ phận mũi và âm thanh phát ra thông qua môi. Đối với quá trình tạo ra âm mũi : vòm miệng mềm hạ xuống và bộ phận mũi liên kết bộ phận miệng , lúc này phía trước của bộ phận miệng khép lại hoàn toàn và âm thanh ra thông qua mũi. Đối với âm thanh nói giọng mũi , âm thanh phát ra cả mũi và môi . Âm thanh của tiếng nói có thể chia làm ba loại khác nhau :

* Âm hữu thanh : giống như âm khi chúng ta nói “a” và “e” được tạo ra khi dây thanh âm căng lên và rung khi áp suất không khí tăng lên, làm thanh mồm mở ra rồi đóng lại khi luồng không khí đi qua . Những dây thanh âm rung tạo ra dạng sóng của luồng khôn khí có dạng xấp xỉ tam giác. Chu kì cao độ âm thanh của đàn ông trưởng thành thường từ 50Hz đến 250Hz, giá trị trung bình khoảng 120Hz. Đối với phụ nữ trưởng thành , giới hạn trên cao hơn nhiều , có thể lên đến 500Hz.
* Âm vô thanh : được tạo ra khi dây thanh âm không rung. Có hai loại âm vô cùng cơ bản: âm xát và âm hơi. Đối với âm xát như khi ta nói chữ “s”. một số điểm trên bộ phận phát âm co lại khi luồng không khí ngang qua nó, hỗn loạn xảy ra tạo nên nhiễu ngẫu nhiên . Đối với âm bật hơi , như khí ta nói chữ “h” , hỗn loạn xảy ra pử gần thanh môn khi dây thanh môn khi dây thanh âm bị giữ nhẹ một phần. Ngoài hai loại âm cơ bản nói trên , còn có một loại âm trung gian vừa mang tính chất nguyên âm , vừa mang tính chất phụ âm , được gọi là bán nguyên âm hay bán phụ âm. Ví dụ như “i” , “u” trong từ “ai” và “âu” .
* Phụ âm nổ : ví dụ âm “p” ,”t” , “k” hay “đ” , “b” , “g” trong tiếng Việt được tạo ra do loại kích thích khác.

### Mô hình lọc nguồn tạo tiếng nói

Quá trình tạo tiêng nói là bộ lọc nguồn,trong đó tín ,từ nguồn âm thanh ( cũng có thể là có chu kì hay nhiều) được lọc bằng bộ lọc biến thiên theo thời gian có tính chất cộng hưởng tương tự với bộ phận phát âm. Nhự vậy có thể thu được phổ tần số của tính hiệu tiếng nói bằng cách nhận phổ của nguồn âm thanh với đặc tính tần số của bộ lọc. Hình bên dưới minh họa tiếng nói hữu thanh với đặc tính tần số của bộ lọc .Các độ lợi Av và AN xác định cường độ của nguồn tạo âm thanh và vô thanh.



**Hình 2‑1** : Mô hình lọc tiếng nói

Mô hình lọc nguồn cho quá trình tạo tiếng nói khá đơn giản nhưng không thể lọc ra được âm xát bằng cách đỉnh cộng hưởng của bộ phận phát âm thanh như âm hữu thanh hay âm bật hơi to, Vì vậy mô hình lọc nguồn hoàn hảo không chính xác cho âm xát .

### Hệ thống nghe của người

Quá trình nghe của người như sau : Sóng áp suất âm thanh tác động đến tai người , sóng này được chuyển thành chuỗi xung điện, chuỗi này được truyền tới não bộ thông qua hệ thần kinh ở não chuỗi được xử lý và giải mã.

Khi nghe một sóng âm thuần túy tức âm đơn , những điểm khác nhau trên màng dây sẽ rung động theo tần số của âm đơn đi vào tai .Điểm lệch lớn nhất trên màng đáy phụ thuộc vào tần số của âm đơn. Tần số cao tạo ra điểm lệch lớn nhất ở phía đáy và tần số thấp tạo ra điểm lệch lớn nhất ở phía đỉnh. Như vậy màng đáy làm nhiệm vụ phân tích tần số tín hiệu vào phức tạp thành những tần số khác nhau ở những điểm khác nhau dọc theo chiều dài của nó. Như vậy có thể xem mọi điểm là bộ lọc thông dải và có tần số trung tâm và băng thông xác định. Ngưỡng nghe của một âm đơn tăng lên khi có sự hiện diện của những âm đơn lân cận khác (âm mặt nạ) và chỉ có băng tần hẹp xung quoanh âm đơn mới tham gia vào hiệu ứng mặt nạ, băng tần này thường gọi là âm tân giới hạn. Giá trị của băng tần tới hạn phụ thuộc vào tần số của âm đơn cần thử. Tóm lại quá trình nghe của hệ thính giác là một dãy các bộ lọc băng thông, có đáp ứng phủ lắp lên nhau và băng thông hiệu quả của chúng xấp xỉ với các giá trị của băng tần tới hạn.

## Xử lý tiếng nói

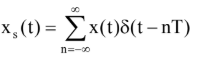
Tín hiệu (signal) là tất cả sự vật hiện tượng có mang hoặc chứa một thông tin nào đó mà chúng ta có thể hiểu, được quy ước trước. Các tín hiệu trong thế giới thực đều ở dạng liên tục ( tín hiệu tương tự) , nó hết sức phức tạp . thiếu độ chính xác cần thiếu đối với máy tính . Do đó các tín hiệu này thường biến đổi thành các tín hiệu số (số hóa), một dạng thông tin máy tính có thể xử lý.

Tiếng nói cũng là một dạng tín hiệu tương tự, do đó nó cũng cần được số hóa

### Lấy mẫu tín hiệu

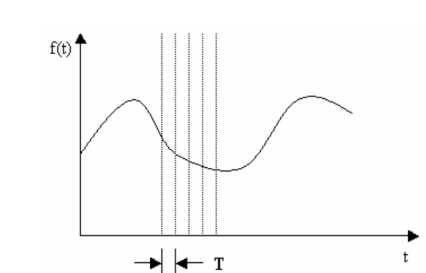
Hàm lấy mẫu là cầu nối giữa các hệ thống rời rạc và các hệ thống liên tục. Nó còn được gọi là hàm Dirac delta, hàm sàng lọc , …

**Công thức 2.2.1**



Đối với máy tính . lấy mẫu chỉ đơn giản là cứ theo một chu kỳ thời gian (đối với tín hiệu âm thanh và các dạng tương tự) , hay là chu kỳ không gian ( đối với tín hiệu là ảnh và các dạng tương tự) ta đo tín hiệu một lần.

Quá trình trên sẽ tạo ra một chuỗi các số biểu diễn cho tín hiệu , và có thể xử lý được bởi máy tính.



**Hình 2‑2**: ví dụ về lấy mẫu tisnhhiejeu f(t) trên miền thời gian

### Bộ lọc tín hiệu

Bộ lọc số có vai trò rất quan trọng trong xử lý tiếng nói , chúng được dùng với 2 mục đích chính :

* **Tách tín hiệu cần thiết** : các tín hiệu ban đầu thường chứa đựng các nhiễu hoặc các tín hiệu không mong muốn khác, các nhiễu này làm giảm đáng kể chất lượng của tín hiệu và cần phải tách riêng các tín hiệu cần thiết .

Ví dụ : đối với âm thanh được thu, tín hiệu âm thanh thường chứa thêm các tiếng ồn của môi trường, chẳng hạn như tiếng ồ ồ của quạt trần thổi vào micro ; còn đối với ảnh chụp thì các điểm lốm đốm trên những tấm ảnh cũ khi được quét ….

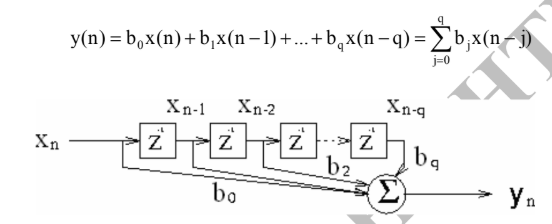
* **Khôi phục các tín hiệu bị biên dạng** : Có một số trường hợp vì một nguyên nhân nào đó ( thường là nguyên nhân liên quan đến thiết bị ) sẽ tạo ra các tín hiệu vào bị méo mó. Vì vậy cần phải chỉnh lại để tăng chất lượng của tín hiệu số.

Ví dụ : Các micro cũ sẽ cho ra các tín hiệu âm thanh không tôt ;“con mắt ” của các máy quét sẽ bị mờ sẽ làm cho các ảnh được quét bị mờ theo …….

Trong thực tế kỹ thuật, có hai bộ lọc tuyết tính dùng để lọc tín hiệu như sau :

* Bộ lọc **đáp ứng xung hữu hạn (**Finite Impulse Response-FIR**)** : hệ có tín hiệu ra chỉ phụ thuộc vào tín hiệu vào nên các hệ này còn được gọi là các mạch không truy hồi hay mạch không đệ qui

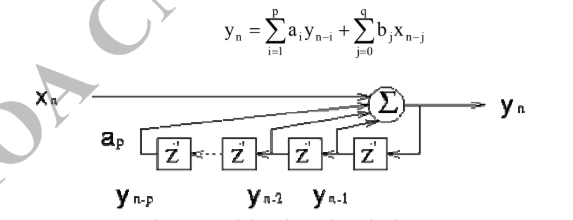
**Công thức 2.2.2**



**Hình 2‑3 :** minh hoạt động của bộ lọc FIR

* Bộ lọc **đáp ứng xung vô hạn** ( Infinite Impulse Response - IIR) : hệ xử lý có đáp ứng xung có độ dài vô hạn hay đáp ứng xung vô hạn. Tín hiệu ra không những chỉ phụ thuộc vào tín hiệu vào mà còn phụ thuộc vào quá khứ của chính tín hiệu ra, vì vậy chúng còn được gọi là các mạch có truy hồi hay đệ qui . Công thực bộ lọc

**Công thực 2.2.3**

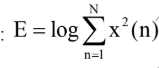


**Hình 2‑4**  : minh họa hoạt động bộ lọc IIR

### Dò tìm điểm cuối (end point ditection )

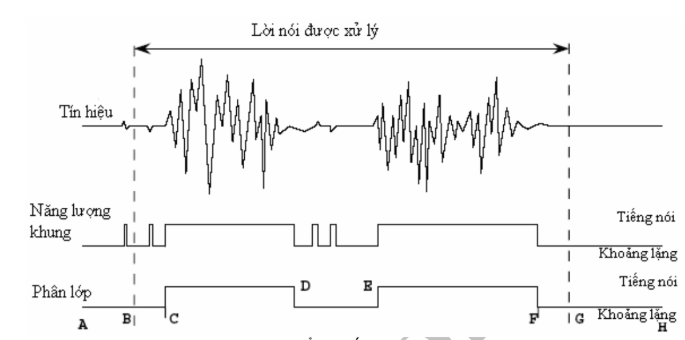
Dò tìm điểm cuối là một xử lý cố gắng tìm ra chính xác khi nào người ta bắt đầu và kết thức nói. Nó còn được dùng để xác định khi mà người ta thật sự không nói gì , hoặc nói những điều không mong đợi (như không có trong bộ từ vựng định trước ). Khi đó , dò tìm điểm cuối giúp giảm một số lượng khung mà trình nhận dạng cần phải xử lý , dẫn đến giảm tải việc tính toán. Tuy nhiên, việc dò tìm điểm cuối không dễ như ta tưởng , bởi vì có sự tồn tại của tiếng ồn nền, tiếng nói nền và sự liên kết của các âm tiết , như là sự khó khăn trong việc dò tìm đoạn vô thanh ở phần bắt đầu và kết thúc tiếng nói.

Dò tìm điểm cuối được thực hiện qua ba bước , qua mỗi bước xác định điểm cuối càng chính xác . Việc dò tìm dựa trên mức năng lượng của tín hiệu được đặc trưng bằng



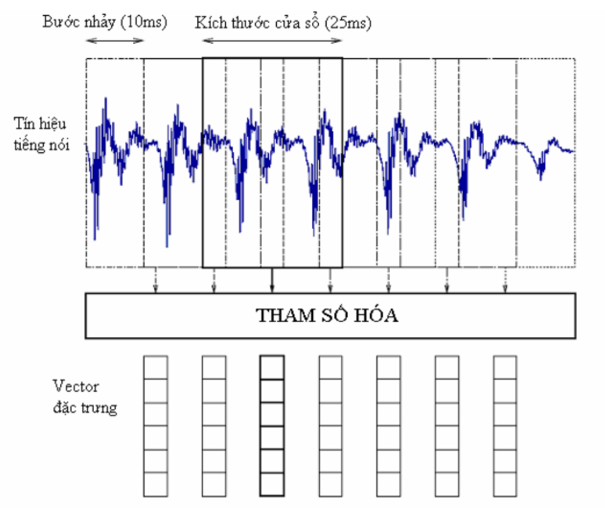
**Hình 2‑5** : Dò tìm tín hiệu đặc trưng

* **Dò tìm thô :** dựa trên kỹ thuật năng lượng ít chính xác nhất . Nó tìm một đoạn mà chức năng lượng cao hơn đoạn trước đó và cho một số khung là điểm bắt đầu thường khoảng 40 khung trước khi gặp khung mức năng lượng cao hơn. Khi một số lượng (thường khoảng 20 khung ) khung khác qua ( không cần kiểm tra bất kỳ khung nào ) được cho là điểm cuối
* **Dò tìm tinh :**  bước dò tìm tinh sẽ kiểm tra mức năng lượng của tiếng nói , nó cố lọc ra điểm đầu và cuối bằng cách cho rằng mức năng lượng của tiếng nói thì cao hơn độ ồm nền ( cao hơn một ngưỡng nào đó)
* **Kỹ thuật VUS (voice , Unvoice and Silence) :** kỹ thuật này cố phân loại từng khung thành đoạn hữu thanh, đoạn vô thanh và khoảng lặng . Việc phân loại từng khung thành đoạn hữu thanh, đoạn vô thanh và khoảng lặng. Việc phân loại dựa trên sự phân bố năng lượng trong khung , phổ biết dạng và sự phân loại khung trước đó . Phương pháp này cố loại bỏ đi những phần không phải tiếng nói như : tiếng nhép miệng , thở hoặc độ ồn nền .



**Hình 2‑6** : dò tìm điểm cuôi dựa vào mức năng lượng

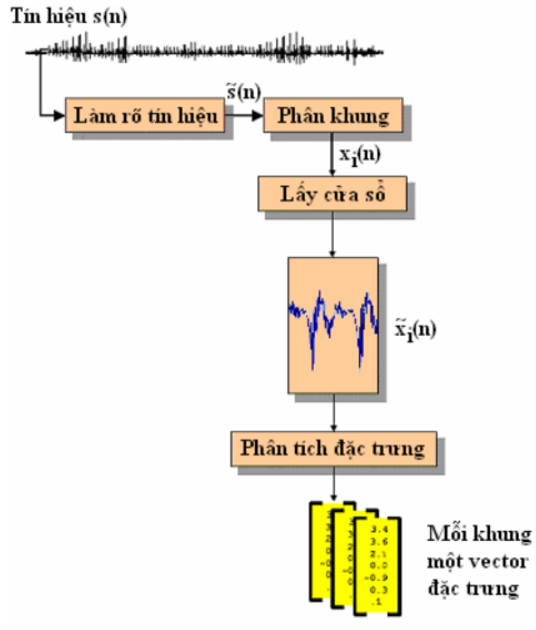
## Rút trích đặc trưng

**Hình 2‑7:** sơ đồ rút trích vector đặc trưng tổng quát

Đối với một hệ nhận dạng tiếng , việc rút trích vector đặc trưng của tiếng nói là cần thiết, Điều này giúp giảm thiểu số lượng dữ liệu trong việc huấn luyện và nhận dạng, dẫn đến số lượng công việc tính toán trong hệ giảm đáng kể. Bên cạnh đó, việc rút trích đặc trưng còn làm rõ sự khác biệt của tiếng này so với tiếng khác , làm mờ đi sự khác biệt của cùng hai lần phát ra âm khác nhau của cùng một tiếng .

Hình 2.6 minh họa các bước xử lý trong việc rút trích vector đặc trưng tiếng nói .

### Các bước rút trích đặc trưng



**Hình 2‑8** : Sơ đồ rút trích đặc trưng chi tiết

### Làm rõ tín hiệu

Ở bước này , mục đích là làm tăng cường tín hiệu , làm nổi rõ đặc trưng của tín hiệu và làm cho nó ít nhạy hơn với các hiệu ứng do độ chính xác hữu hạn ở những bước xử lý sau. Bộ làm rõ tín hiệu thường là một bộ lọc thông cao với phương trình sai phân như sau :

**Công thức 2.3.1 :**



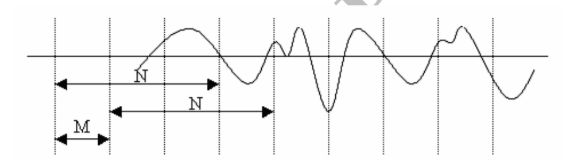
* **Phân đoạn thành các khung**

Trong bước phân đoạn khung được chia thành các khung , mỗi khung gồm N mẫu , khoảng cách giữa các khung là M mẫu . Hình 2.7 minh họa cách phân thành các khung trong trường hợp M = (1/3)N

Cụ thể khung thứ nhất gồm N mẫu tiếng nói đầu tiên ( bắt đầu từ S(0 0 đến S0(N-1))). Khung thứ 2 bắt đầu từ mẫu M và kết thúc ở vị trí M+N – 1.Tương tự , khung thứ I bắt đầu từ mẫu i\*M và kết thúc ở vị trí i9M+N-1.Tiến trình này tiếp tục cho đến khu các mẫu tiếng nói đều đã thuộc về một hay nhiều khung.

Ta dễ dàng thấy rằng nếu M <= N thì các khung kề nhau sẽ có sự chồng lấp (hình 2.7) dẫn đến kết quả là các phép rút trích đặc trưng có tương quan với nhau từ khung này sang khung kia; và khi M<<N thì khung này sang khung khác được hoàn toàn trơn. Ngược lại , nếu M>N thì sẽ không có sự chồng lấp giữa các khung kề nhau, dẫn đến một số mẫu tiếng nói bị mất ( tức là không xuất hiện trong bất kỳ khung nào). Nếu ta ký hiệu khung thứ I là xi(n) và giả sử có tất cả L khung trong tín hiệu tiếng nói thì :



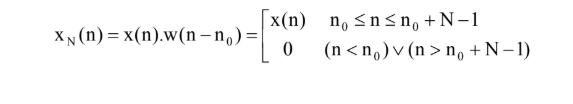


**Hình 2‑9:** phân đoạn tiếng nói thành các khung chồng lấp

* **Lấy cửa sổ.**

Bước tiếp theo trong xử lsy là lấy cửa sổ tín hiệu ứng với mỗi khung để giảm thiểu sự gián đoán tín hiệu ở đầu và cuối mỗi khung để giảm thiểu sự gián đoạn tín hiệu ở đầu và cuối mỗi khung. Một dãy tín hiệu con được lấy ra từ một tín hiệu dài hơn hoặc dài vô hạn x(n) gọi là một cửa sổ tín hiệu. Việc quan sát tín hiệu x(n) bằng một đoạn xN(n) trong khoảng n0……(n0 + N - 1) tương đương với việc nhân x(n) với một hàm cửa sổ w(n-n0)

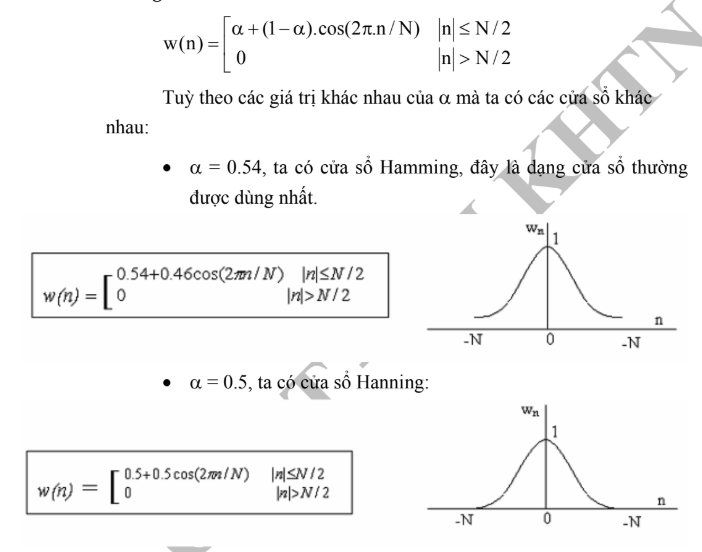
**Công thức 2.3.2**

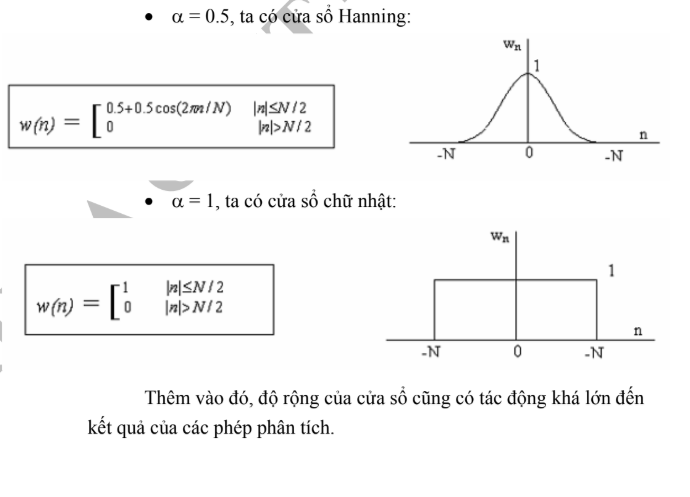


**Các dạng cửa sổ tín hiệu**

Trong xử lý tín hiệu số ,các cửa sổ thường dùng được biểu diễn thông qua cửa sổ Hamming tổng quát :

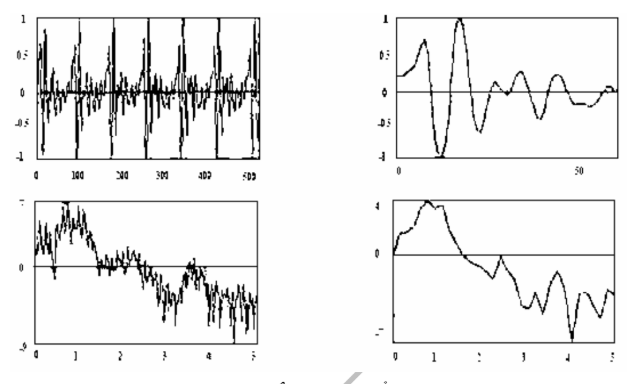
**Công thức 2.3.3**





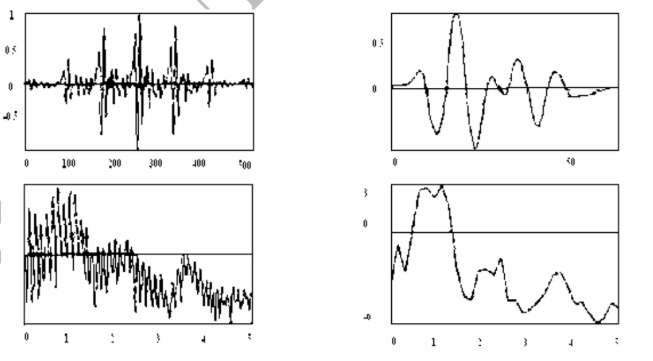
Một số cửa sổ khác cũng được sử dụng trong xử lý tiesn hiệu số như : cửa sổ tam giác , cửa sổ Kaiser , cửa sổ Blackman,cửa sổ cosin…

Sau đây một số ví dụ cho thấy sự khác biệt giữa các loại cửa sổ .

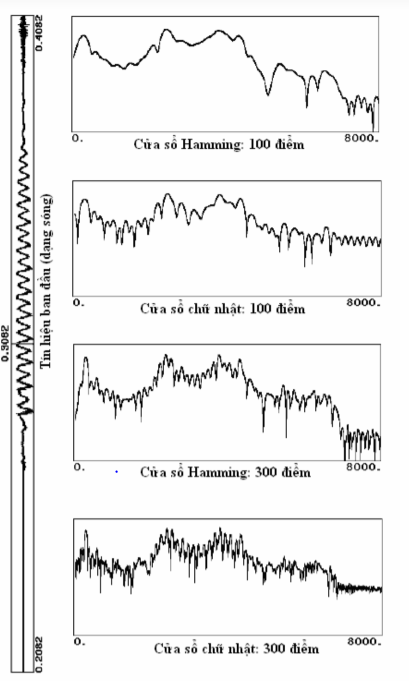


**Hình 2‑10** : :Âm/a/, cửa sổ chữ nhật

512 điểm (45ms, trái) và 64 điểm (5.6ms, phải)



**Hình 2‑11** : 512 điểm (45ms, trái) và 64 điểm (5.6 ms,phải)



**Hình 2‑12** : Sự khác biệt giữa các cửa sổ tín hiệu

### Các đặc trưng tiếng nói

Để rút trích đặc trưng , ta cần phải chọn đặc trưng thỏa mãn những vấn đề sau đây :

* Có khả năng diễn đjat thôn tin tiếng nói độc lập với người nói
* Dễ dàng tính toán
* Ổn định theo thời gian
* Xảy ra tự nhiên và liên tục trong tiếng nói
* Ít thay đổi theo môi trường nói
* Không ảnh hưởng bởi sự biến dạng bóp méo
* Không ảnh hưởng bởi độ ồn nền và băng tần giới hạn
* Không ảnh hưởng bởi trạng thái người nói
* **Đặc trưng có tất cả những đặc tính như thế không tồn tại !**

**Các dạng đặc trưng hiện nay**

* **Đặc trưng miền âm**
* Autocorrelation coefficients (COR)
* Linear Prediction coefficients (LPC)
* Partial Correlation coefficients (PARCOR)
* Log Area Ratio coefficients (LAR)
* Perceptional Linear Prediction (PLP)
* **Đặc trưng miền tần số và Cepstral**
* Liner Spectrum Pairs (LSP)
* Bank of filters (tuyến tính)
* Bank of filters(mel)
* Mel Frequency Cepstral coefficients (MFCC)

**Biến đổi tín hiệu sang miền tần số**

Có hai cách biến đổi :

1. Phép biến đổi Fourier rời rạc

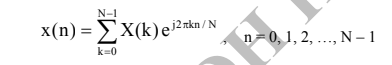
Phép biến đổi Fourier là phép biến đổi thuận nghịch ,dùng để biến đổi tín hiệu sang miền tần số, nó dùng các công thức để biến đổi rời rạc sau :

Phép biến đổi thuận :

**Công thức** :



Phép biến đổi nghịch :

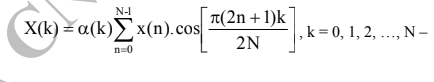


1. **Phép biến đổi cosin rời rạc**

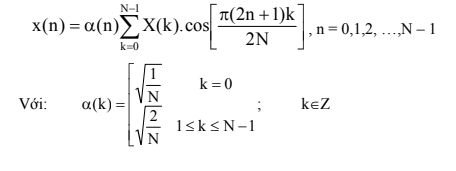
Biến đổi Cosin là một phép biến đổi mạnh , được dùng trong xử lý nén ảnh JPEG , nó cũng là nột phép biến đổi chuyển tín hiệu sang miền tần số , ta có các công thức :

Phép biến đổi thuận :

**Công thức** :

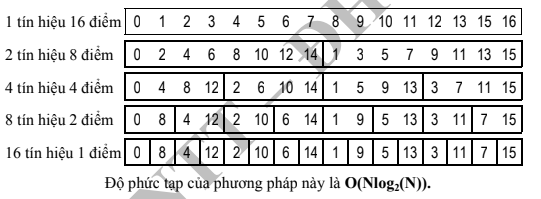


Phép biến đổi nghịch :



Cả hai phép biến đổi trên đều có phiên bản biến đổi hanh, điều này giúp tăng tốc độ xử lý ,thích hợp trong việc xử lý cần thời gian thực như xử lý âm thanh. Đó là FFT (Fast Fourie Transform) và FCT ( Fast Cosine Transform). Các phép biến đổi nhanh này đều dựa trên kỹ thuật phân chia theo cơ số 2 , nghĩa là thay vì biến đổi trên toàn bộ tín hiệu thì phép biến đổi này sẽ phân chia chuỗi tín hiệu thành 2 chuỗi tín hiệu con và lại áp dụng phép biến đổi này lần nữa cho 2 phần này một cách đệ quy. Do phép chia cho 2 , nên chuỗi tín hiệu đòi hỏi phải có chiều dài là lũy thừa của 2 (điều này có thể dễ dàng giải quyết được bằng cách tăng kích thước chuỗi tín hiệu là điều 0 vào)

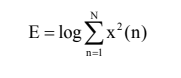
Ví dụ việc phân chia và biến đổi sẽ được thực hiện trên chuỗi tín hiệu có chiều dài 16 điểm như sau :



**Đặc trưng năng lượng**

Năng lượng tín hiệu được thể hiện thông qua mức độ, số lượng tín hiệu có trong một đơn vị thời gian . Năng lượng của tín hiệu tiếng nói là một đặc trưng vật lý của tín hiệu, được dùng như là tham số trong vector đặc trưng trong nhận dạng tiếng nói và còn được để dò tìm khoảng lặng trong tín hiệu nói . Tính toán năng lượng tín hiệu thường dựa trên sự phân khung là lấy cửa số , bằng cách lấy tổng các bình phương chuỗi tín hiệu x(n) trong cửa sổ tín hiệu.

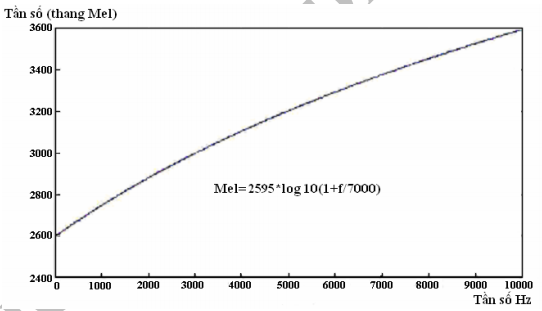
Đặc trưng năng lượng ở đây được tính bằng cách lấy log năng lượng tín hiệu được tính bằng công thức sau :



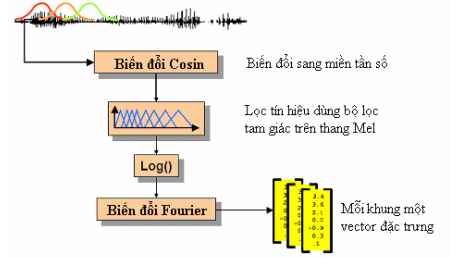
**Đặc trưng MFCC**

Các nghiên cứu cho ta thấy rằng hệ thống thính giác của con người thu nhận âm thanh với tốc độ lớn các tần số âm thanh không theo thang tuyến tính. Do đó, cac âm thanh đã rời cho phù hợp với sự tiếp nhận của thính giác con người .

Các thang được xây dựng bằng thực nghiệm , cho nên người ta xây dựng các công thức để xấp xỉ sự chuyển đổi này. Trong các thang và công thức dạng đó thì đặc MFCC sử dụng thang Mel . Thang Mel được thể hiện thông qua đồ thị sau :

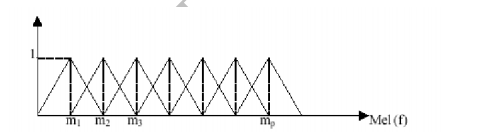


**Hình 2‑13 :** đồ thị biểu diễn mối quan hệ giữa Mel và Hz

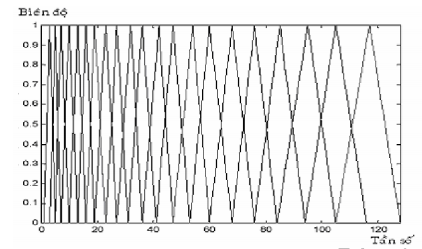


**Hình 2‑14** : Phép biết đổi Fourier

Ta dùng phép biến đổi Fourier để chuyển tín hiệu từ miền thời gian sang miền tầng số. Sau đó ta dùng dãy bộ lọc để lọc tín hiệu , đó là dãy bộ lọc tam giác có tần số giữa đều nhau trên thang Mel.

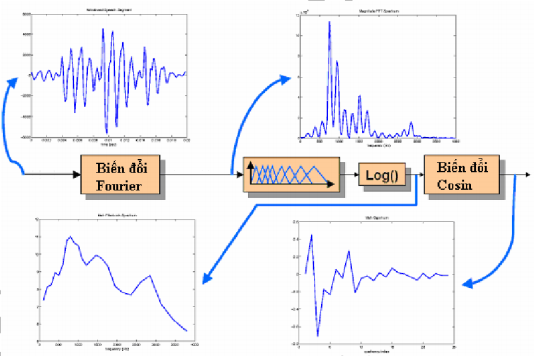


**Hình 2‑15:** Bộ lọc trên thang Mel



**Hình 2‑16 :** Bộ lọc trên tần số thật

Lấy log trên dãy kết quả từ dãy bộ lọc và thực hiện biến đổi cosin rời rạc ta thu được các hệ số đặc trưng MFCC

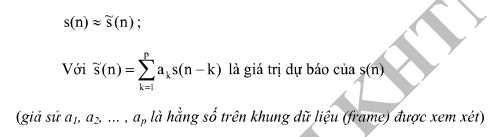


**Hình 2‑17**: Minh họa các bước biến đổi MFCC

**Đặc trưng LPC**

Ý tưởng cơ bản của phương pháp LPC là tại thời điểm n , mẫu tiếng nói s(n) có thể được xấp xỉ bởi một tổ hợp tuyến tính của p mẫu trước đó

**Công thức :**

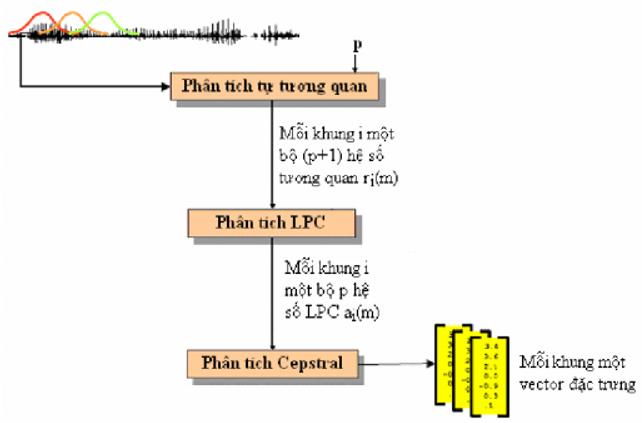


Chúng ta chuyển quan hệ trên thành dạng đẳng thức bằng cách thêm vào số hạng G.u(n) gọi là nguồn kích thích :

**Công thức :**



Để tìm tập hợp các hệ số ai , k= 1, 2,…, p trên khung được phân tích , cách tiếp cận cơ bản là ta cực tiểu hóa sai số bình phương trung bình. Khi đó sẽ dẫn đến việc ta phải giải một hệ phương trình với p ẩn số. Có nhiều phương pháp để giải hệ phương trình đó , nhưng trong thực tế , phương pháp thường được dùng là phương pháp phân tích tự quan.



**Hình 2‑18** : Sơ đồ xử lý LPC dùng trích đặc trưng tiếng nói

Hình 2.13 trình bày sơ đồ chi tiết của quá tình xử lý LPC để rút trích đặc trưng tieesg nói. Các bước cơ bản trong tiến trình xử lý như sau :

**Phân tích tự quan :**

Mỗi khung sau khi được lấy cửa sổ sẽ được đưa qua bước phân tích tự tương quan và cho ra (p + 1) hệ số tự quan :

**Công thức :**



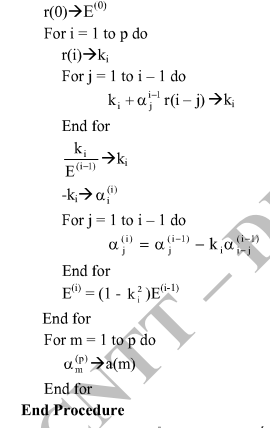
Trong đó giá trị tương quan cao nhất , p , được gọi là cáp của phân tích LPC. Thông thường , ta sử dụng các giá trị p trong khoảng tư 8 đến 16.

**Phân tích LPC :**

Bước này , ta sẽ chuyển mỗi khung gồm (p+ 1) hệ số tự tương quan thành p hệ số LPC bằng cách dùng thuật toán Levinson – Durbin.

Thuật toán Levinson – Durbin thể hiện qua mã giả sau , với dữ liệu vào là p +1 hệ số tự tương quan chứa trong r, kết quả ra là p hệ số LPC chứa trong a.

**Procedure Levinson \_ Durbin (Vector a , Vector r )**



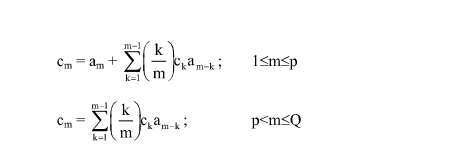
Lúc này , ta có thể dùng các hệ số LPC làm vector đặc trưng cho từng khung. Tuy nhiên , có một phép biến đổi tạo ra dạng hệ số khác có độ tập trung cao hơn từ các hệ số LPC, đó là phép tích phân Cepstral

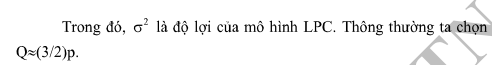
**Phân tíc cepstral**

Từ p hệ số LPC ở mỗi khung , ta dẫn xuất ra q hệ số cepstral c (m)

Theo công thức đệ quy sau :







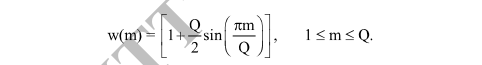
**Đặt trọng số cho các hệ số cepstral**

Do độ nhạy của các hệ số cepstral cấp thấp làm cho phổ bị đổ dốc và do độ nhạy của các hệ số cep cấp cao gây ra nhiễu nên ta thường sử dụng kỹ thuật đặt trọng số để làm giảm thiểu các độ nhạy này :



Với w(m) là hàm đặt trọng số . Hàm đặt trọng số thích hợp thường là bộ lọc thông dài :

**Công thức :**



**Nhận xét:**

Mô hình LPC là mô hình đặc biệt thích hợp cho tín hiệu tiếng nói . Với miền tiếng nói hữu thanh có trạng thái gần ổn định , mô hình tất cả các điểm cực đại của LPC cho ta một xấp xỉ tốt đối với đường bao phổ âm. Với tiếng nói vô thanh , mô hình LPC tỏ ra ít hữu hiệu hơn so với hữu thanh , nhưng nó vẫn là mô hình hữu ích cho các mục đích nhận dạng tiếng nói. Mô hình LPC đơn giản và dễ cài đặt trên phần cứng lẫn phần mềm . Đặc biệt , kinh nghiệm đã chứng tỏ rằng phương pháp LPC thực hiện tôt hơn so với bộ trích đặc trưng bằng dãy bộ lọc.

**Đặc trưng tầm số cơ bản**

Tần số cơ bản đóng một vai trò quan trọng trong nhận dạng tiếng nói. Tư tần số cơ bản , ta có thể có những phân biệt các tiếng theo một số đặc điểm ngữ âm . Tầm số cơ bản còn thể hiện sắc thái , thanh điệu , giọng người nói … Do đó, xác định tần số cơ bản là một phần công việc không thể thiếu trong các hệ nhận dạng tiếng nói, đặc biệt là tiếng nói có thanh điệu như tiếng Việt

Sự thể hiện của các thanh điệu liên quan đến giá trị và sự biến đổi của tần số cơ bản . Trong xử lý tiếng nói , tín hiệu được chia thành các khung liên tiếp nhau, nên thanh điệu sẽ được thể hiện bằng tần số cơ bản trong từng khung tín hiệu cũng như sự vận động của nó từ khung này sang khung khác.

Sự thể hiện của các thanh điệu liên quan đến giá trị và sự biến đổi của tần số cơ bản. Trong xử lý tiếng nói , tín hiệu được chia thành các khung liên tiếp nhau, nên thanh điệu sẽ được thể hiện bằng tần số cơ bản trong từng khung tín hiệu cũng như sự vận động của nó từ khung này sang khung khác.

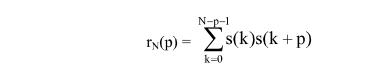
Tín hiệu đầu vào của các phương pháp trên là tín hiệu tiếng nói thô , hoặc tín hiệu đã được xử lý bằng một phép toán phi tuyến ( như cắ tâm) hay dùng lỗi dự báo (trong mô hình LPC )

Tần số cơ bản chỉ có trong các âm hữu thanh , nên việc rút trích tần số cơ bản cũng phải đảm nhận luôn việc phân biệt giữa các âm vô thanh và hữu thanh .

Để tăng hiệu quả , người ta tiến hành một số bước tiền xử lý cho tín hiệu tiếng nói thô, nhằm tăng độ chính xác , giảm khối lượng tính toán. Thông thường tín hiệu thô được xử lý qua 2 bước trước khi dùng để trích F0 :

* Lọc thông thấp : tín hiệu tiếng nói được cho qua bộ lọc thôn thấp để loại bỏ các thành phần có tần số cao hơn Fmax  (tần số cơ bản lớn nhất có thể của tiếng nói ). Thông thường Fmax  = 900 Hz .
* Thực hiện việc lấy mẫu lại, giảm bớt kích thước sóng âm. Tần số lấy mẫu được giảm xuống còn 2 KHz ( theo định luật Nyquist : tần số lấy mẫu phải lớn hơn hoặc bằng 2 lần tần số cơ bản lớn nhất) . Lấy mẫu lại tần số giúp giảm đáng kể khối lượng tính toán. Ở các phương pháp tìm F0 thông thường, khối lượng tính toán giảm Z2  lần. với Z tỉ lệ giảm tần số lấy mẫu
* Phương pháp tự tương quan : Thực hiện hàm tự tương quan trên khung tín hiệu tiếng nói độ dài N

**Công thức**



Trong đó ,p được giới hạn trong vùng có âm cơ bản. Nếu tín hiệu s(n) là tuần hoàn thì sẽ có các đỉnh tại I =0, P , 2P,… (P là chu kỳ âm cơ bản). Nếu ngưỡng quyết định thường là rN (p) > 0.8 rN (0) . Có một vài ý tưởng tạo ra ngưỡng động dựa vào tương quan năng lượng của khung tín hiệu và năng lượng trung bình của cả tính hiệu.

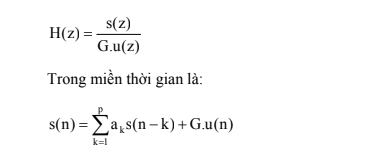
**Nhận xét** :

* Thông thường , tín hiệu được nhận với mổ hàm cửa sổ để giảm sự tác động do sự thay đổi âm điệu.
* Nếu áp dụng phương pháp này cho tín hiệu tiếng nói thô thì tỏ ra không tốt , đỉnh xuất hiện không rõ.
* Cần một bước tiền xử lý để loại bỏ thông tin của dãy âm .
* Áp dụng phương pháp này cho e (n) tốt hơn (phương pháp Simplifie Inverse Filter Tracking)
* Có lấy vài đỉnh trong một khung tín hiệu sau đó dựa vào phương pháp Dynamic Programing để tìm ra chuỗi F0 trong một đoạn các khung liên tiếp .

Một phương pháp dẫn xuất từ phương pháp này là dùng hiệp tương quan giữa hai tín hiệu x(n) và y(n) , y(n) = x(n + P) (tín hiệu y(n) là do tín hiệu x(n) dịch đi P đơn vị

**Lỗi LPC và phương pháp SIFT**

Mô hình LPC đặc trưng bằng hàm truyền đạt có dạng như sau :



Trong đó , G.u(n) chính là nguồn kích thích , trong trường hợp âm hữu thanh, G.u(n) chính là miêu tả chính xác dạng dao động của dây thanh hay F0.

Chúng ta cũng định nghĩa lỗi của ước lượng , e~(n) như sau :



Như vậy mô hình LPC đã tạo ra tín hiệu lỗi dự báo e~(n) chứa thông tin về nguồn kích thích , và do đó việc xách định F0 trong trường hợp hữu thah trở nên dễ dàng hơn.

Đối với phương pháp SIFT ( Simplified Inverse Filter Tracking) , phương pháp này áp dụng phương pháp tự tương quan với tín hiệu vào là e~(n) thu được ở trên .

**Phương pháp dùng cepstral**

Phương pháp này có thể mô tả đơn giản như sau :

* Dùng phép phân tích Cepstral thực cho tín hiệu vào . Tín hiệu vào này có thể sử dụng trực tiếp tiếng nói thô .
* Tìm đỉnh trong vùng thích hợp của tín hiệu cn

Nhận xét

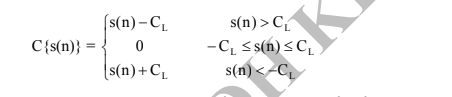
* Đỉnh được tìm khá chính xác , ít bị lấy nhầm hài âm .
* Dùng tốt trong trường hợp tiếng nói có cao độ thấp
* Việc xác định ngưỡng để quyết định có đỉnh tại cn0 không tùy thuộc vào người nói => khó phân biệt vô thanh/hữu thanh

**Phương pháp CLIP**

Phương pháp CLIP (center clipping pitch detector) tương tự như phương pháp tự tương quan ở trên, những tín hiệu được xử lý để loại bỏ thông tin về các phoocmăng (thông tin về đường phát âm)

Có một vài giải pháp cho việc này. Cụ thể là phương pháp cắt tâm (center clipping) . Phương pháp này sẽ loại bỏ bớt các đỉnh nhỏ trên sóng âm , làm cho sóng âm nhìn giống dạng xung hơn.

Phép toán cắt C được mô tả như sau :



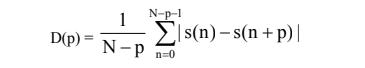
Trong đó CL là ngưỡng cắt , thường được lấy bằng 30% giá trị lớn nhất của tín hiệu

**Hàm AMDF**

Phương pháp (Average Magnitude Difference Function ) giống phương pháp tự tương quan ở trên , nhưng khối lượng tính toán sẽ giảm xuống do không phải dùng phép nhân.

Chúng ta định nghĩa hàm trung bình hiệu biên độ như sau :

**Công thức**



Sau khi tính D(p) trong vùng có khả năng xuất hiện P0 . Chọn điểm cực tiểu D(P0) , P0 là chu kỳ tần số cơ bản .

**Phương pháp so khớp biên độ**

Chuỗi tín hiệu tiếng nói đưa vào máy tính có dạng hình sin. Do đó, ta sẽ tìm hai dao động cùng pha , khoảng thời gian giữa hai điểm đó chính là chu kỳ T. Tư T, ta sẽ tìm ra tần số f.

Tuy nhiên cần chú ý rằng tín hiệu tiếng nói là sự tổng hợp của nhiều tần số do đó , hai điểm dao động cùng pha được xét phải là 2 điểm cắt zero . Bên cạnh đó , ta cũng phải xác định dùng 2 điểm cắt zero để tạo thành chu kỳ của F0, vì các dao động cộng hưởng cũng có thể gây ra điểm cắt zero

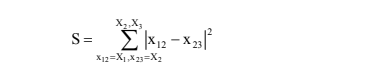


**Hình 2‑19** : Hình dạng tín hiệu tiếng nói

Phương pháp so khớp biên độ được tiến hành như sau :

* Dò tìm điểm cẳ zero thứ nhất theo một chiều nào đó (ví dụ đi lên như trong hình vẽ ) đặt tên là X1
* Dò tìm 2 điểm cẳ zero cùng chiều tiếp theo, đặt tên là X2 , X3. Với khoảng thời gian giữa X1 X2 và X2 X3 là tương đương nhau và nằm trong khoảng ngưỡng thời gian xác định chu kỳ .
* Lần lượt so sánh biên độ các điểm tương ứng trong hai khoảng X1 X2 và X2 X3 . Gọi tổng bình phương các độ sai lệch biên độ là S.

**Công thức**



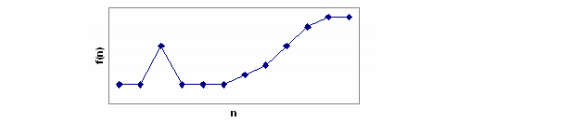
* Nếu S nhỏ hơn ngưỡng độ lệch thì kết luận mỗi khoảng đó là một chu kỳ. Nếu không , thay đổi đổi khoảng thời gian . nghĩa là dò tìm các điểm cắt zero khác .

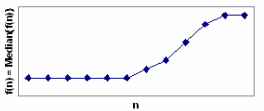
**Làm trơn kết quả F0 bằng bộ lọc median**

Bộ lọc median được dùng khá rộng rãi trong việc khử nhiễu. Nội dung kỹ thuật được thể hiện như sau :

* ĐỂ có một tín hiệu ra, mở cửa sổ các tín hiệu vào liền nhau được chọn
* Sắp xếp các dữ liệu trong cửa sổ tín hiệu kể trên
* Giá trị trung tâm của dãy đã sắp xếp được chọn làm median của tập hợp các mẫu trong cửa sổ

Có nghĩa là bộ lọc median sẽ tính lại giá trị một điểm bằng các lấy điểm có giá trị trung bình trong các điểm xung quoanh.

**Hình 2‑20** : kết quả trích F0



**Hình 2‑21** : Kết quả sau khi lọc median

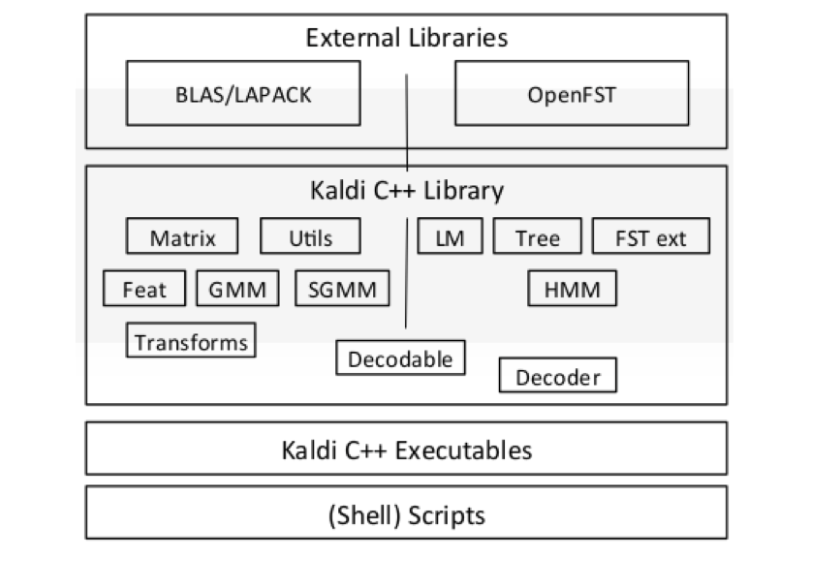
# : frame work kaldi sử dụng trong nhận dạng giọng nói và machine learning

Một hệ thống nhận dạng tiếng nói tốt là hệ thống có khả năng nhận dạng được mọi câu nói của người sử dụng .Thực tế cho thấy ngay cả con người trong một số trường hợp cũng không thể hiểu được cùng một câu nói nếu có được phát âm bởi một người ở địa phương khác. Làm sao để hệ thống sau khi đã huấn luyện có thể vẫn nhận dạng được các giọng nói mới với độ chính xác mong muốn .

## Kaldi là gì ?

Kaldi là một bộ công cụ mã nguồn mở được tạo ra để xử lý dữ liệu giọng nói , nói được sử dụng trong các ứng dụng liên quan đến giọng nói , chủ yếu để nhận dạng giọng nói mà còn các tác vụ khá – như nhận dạng loa và tăng âm loa. Bộ công cụ này đã khá cũ (khoảng 7 tuổi) nhưng vẫn liên tục được cập nhật và phát triển thêm bởi một cộng đồng khá lớn. Kaldi được áp dụng rộng rãi cả trong Học viện ( hơn 400 trích dẫn trong 2015) và công nghiệp

Kaldi được viết chủ yếu bằng C / và C ++ nhưng bộ công cụ được gói bằng các tập lệnh Bash và Python . Đối với việc sử dụng cơ bản gói này không cần phải đi quá sâu vào mã nguồn mở



**Hình 3‑1** : Cấu trúc cơ bản của Kaldi một số thành phần chính của Kaldi

## Tiền xử lý và khai thác tính năng

Ngày nay , hầu hết các mô hình xử lý dữ liệu âm thanh đều hoạt động với một số biểu diễn dự trên pixel của dữ liệu đó. Khi bạn muốn trích xuất đại diện như vậy , bạn sẽ hầu như muốn sử dụng các tính nắn sẽ tốt cho cả hai điều :

* Xác định âm thanh của lời nói của con người
* Loại bỏ bất kỳ tiếng ồn không cần thiết

Trong những năm qua đã có nhiều cố găng để làm cho những tính năng và ngày nay phương pháp trích trọn đặc trưng sử dụng bộ lọc và thang tần số Mel (MFCC) được sử dụng rộng rãi trong ngành công nghiệp .

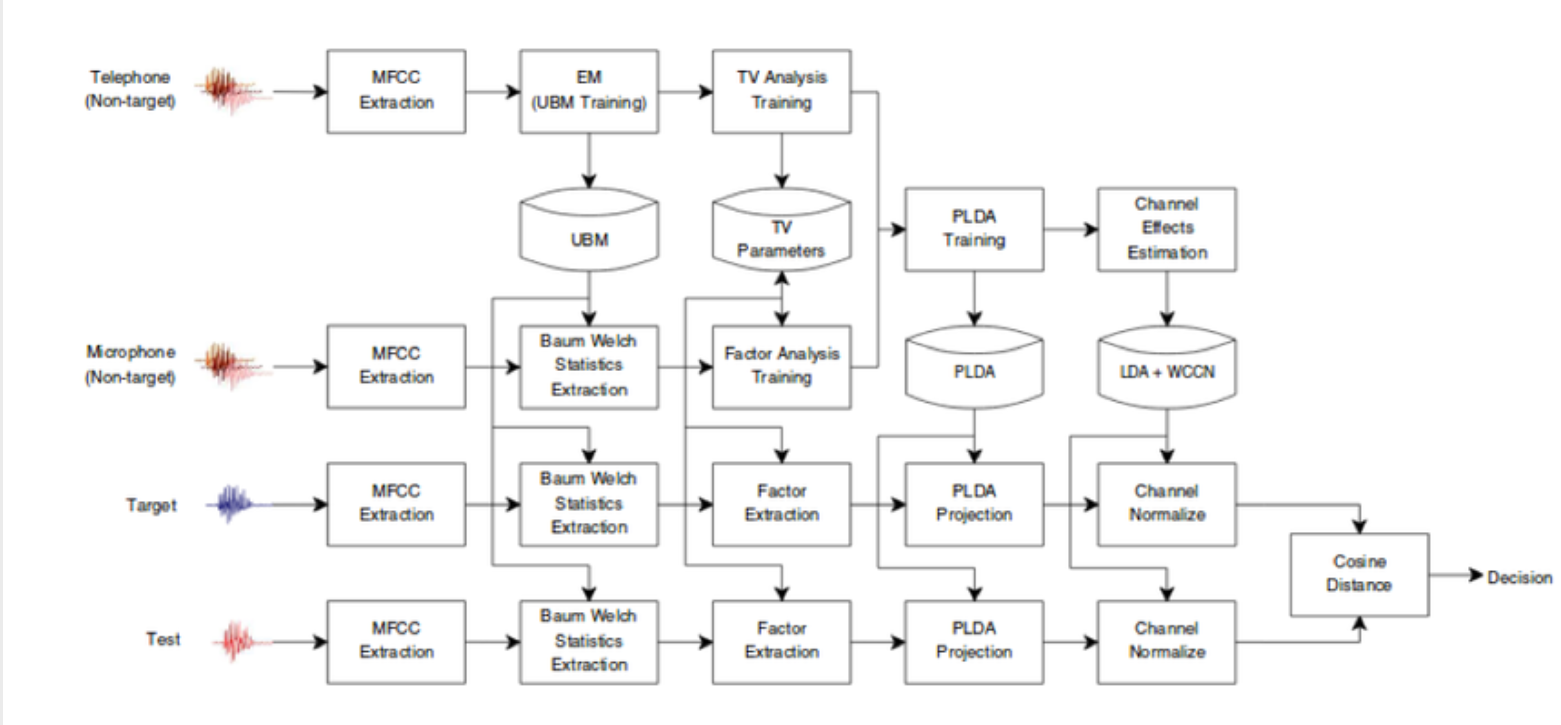
MFCC là viết tắt của hệ số cepstral Mel – tần số và nói gàn như đã trở thành một tiêu chuẩn trong ngành kể từ khi nó được phát minh vào những năm 80 bởi Davis và Mermelstein. Bạn có thể có được một lời giải thích lý thuyết tốt hơn về MFCC trong bài viết tuyệt vời dễ đọc này. Đối với việc sử dụng cơ bản , tất cả những gì bạn cần biết là MFCC . Đối với việc sử dụng cơ bản, tất cả những gì bạn cần biết là MFCC chỉ tính đến những âm thanh được nghe bằng tai tốt nhất.

Trong Kaldi, chúng tôi sử dụng hai tính năng khác :

CMVN được sử dụng để chuẩn hóa tốt hơn MFCC

I-vector được sử dụng để hiểu rõ hơn về các phương sai trong miền. Ví dụ : tạo một đại diện phụ thuộc loa. I-vector dựa trên cùng một ý tưởng của JFA (phân tích nhân tốt chung) , nhưng phù hợp hơn để hiểu cả phương sai của kênh và người nói.

Quá trình thực hiện sử dụng I-Vectors



**Hình 3‑2 :** Quá trình thực hiện sử dụng I-vector

Để hiểu cơ bản về các khái niệm này , hay nhỡ điều sau đây :

MFCC và CMVN được sử dụng để thể hiện nội dung của từng âm vị cách phát âm

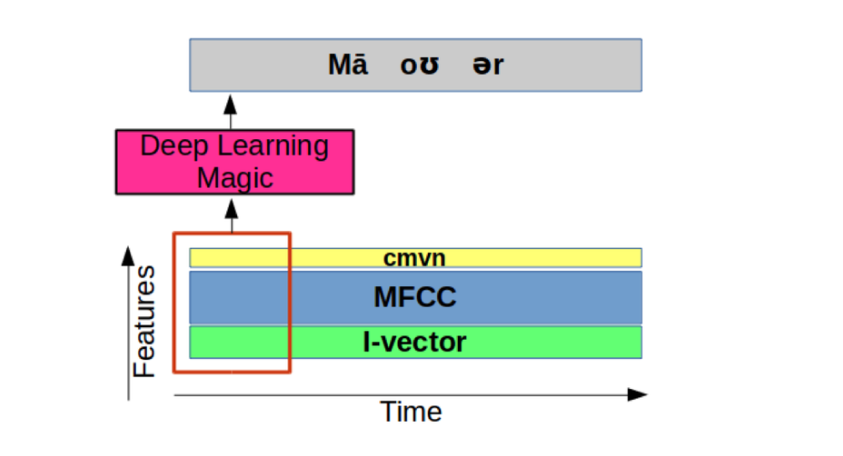
I-vector được sử dụng để thể hiện đặc trưng của từng phát ngôn hoặc âm thanh

## Model

Các ma trận toán học đăng sau Kaldi được thực hiện một trong hai BLAS ( Basic linear Algebra Subprograms ) và LAPACK – Linear Algebra Package (viết tắt bằng fortran!) . Hoặc với một sự lụa chọn thực hiệu GPU dự trên Cuda. Do sử dụng các gói âm thấp như vậy,Kaldi có hiệu quả cao trong việc thực hiện các nhiệm vụ đó.

Mô hình của Kaldi có thể được chia làm hai phần chính : Phần đầu tiên là Mô hình âm thanh , từng là GMM nhưng giờ đây nó đã được thay thế một cách dữ dội bởi các mạng lưới Deep neural. Mô hình đó sẽ phiên âm các tính năng âm thanh mà chúng tôi đã tạo thành một số chuỗi âm vị phụ thuộc vào ngữ cảnh .

Mô hình Acoustic :



**Hình 3‑3 :** Mô hình Acoustic

Phần thứ hai là Đồ thị giải mã , lấy các âm vị và biến chúng thành các mạng. Một mạng là một đại diện của các chuỗi từ thay thế có khả năng cho một phần âm thanh cụ thể. Đây thường là đầu ra mà bạn muốn nhận được trong một hệ thống nhận dạng giọng nói. Biểu đồ giải mã có tính đến ngữ pháp của dữ liệu của bạn, cũng như phân phối và xác suất của các từ cụ thể liền kề (n - gram) .

Đây là một sự đơn giản hóa về cách thức mà mô hình hoạt động. Thực tế có rất nhiều chi tiết về việc kết nối hai mô hình với cây quyết định về các bạn thể hiện các âm vị, nhưng sự đơn giản hóa này có thể giúp bạn nắm được quy trình thực hiện giải mã của Kaldi.

## Mã ma trận trong Kaldi

Là một trình bao bọc trên đầu các thư viên đại số tuyến tính BLAS và LAPACK. Mã này đã được thiết kế để linh hoạt nhất về các thư viện mà n sử dụng. Hiện tại nó hỗ trợ 4 tùy chọn:

* ATLAS là một phát triển của BLAS cùng với một tập hợp LAPACK
* Một phát triển BLAS với CLAPACK
* MKL của intel cung cấp cả BLAS với LAPACK
* OpenBLAS ,cung cấp BLAS với LAPACK

Mã được biết lựa chọn nào trong bốn tùy trọn này đang được sử dụng , bởi vì mặc dù về nguyên tắc BLAS và LAPACK được tiêu chuẩn hóa, có một số khác biệt trong giao diện. Mã Kaldi yêu cầu chính xác một trong ba chuỗi Lawr\_ATLAS, Lawr\_CLAPACK, Lawr\_OPENBLAS hoặc Lawr\_MKL để được xác dịnh ( ví dụ : sử dụng – DHAVE\_ATLAS làm tùy chọn cho trình biên dịch ). Sau đó nó phải được liên kết với các thư viện thích hợp. MÃ liên quan trực tiếp đến việc bao gồm các thư viện bên ngoài và thiết lập các typedef thích hợp và định nghĩa là trong kaldi-blas.h. Tuy nhiên , phần còn lại của mã ma trận không được cách ly hoàn toàn khỏi các vấn đề này bởi vì các phiên bản cao cấp hơn ATLAS và CLAPACK gọi khác nhau.

Tập lênh “ configure” trong thử mục “ src” chịu trách nhiện thiết lập Kaldi để sử dụng các thư viện. Nó thực hiện điều này bằng các tạo tệp “kaldi.mk” trong thư mục “src”, cung cấp các cờ thích hợp cho trình biên dịch. Nếu được gọi mà không có đối số , nó sẽ sử dụng bất kỳ cài đại ATLAS nào, nó có thể tìm thấy ở những nơi “ bình thường” trong hệ thống của bạn, nhưng nó có thể cấu hình được.

## BLAS ( basic linear algebra subprograms)

Là một thư viện phần mềm chứa các thuật toán xử lý ma trận trong đại số tuyến tính. BLAS có nhiều ứng dụng trong tính toán hiệu năng cao.Blas có nhiều phiên bản cho các kiến trúc máy tính khác nhau và ngôn ngữ lập trình khác nhau , ví dụ ACML ( ADM core math library) phát triển cho chip AMD athlon và Opteron, intel MKL (intel Math kernel library) cho chip xử lý của Intel.Intel MKL có hiệu suất rất cao .

Trong phần này BLAS là một tập hợp các khai báo chương trình con tương ứng với các phép toán vector ma trận mức thấp. Có blas cấp 1 (vector-vector) ,cấp 2 (vector-matrix) và cấp 3 (matrix - matrix). Chúng có các tên như daxpy ( cho độ chính xác kép a\*x +y) và dgemm ( cho phép nhân ma trận tổng quát ma trận kép). BLAS có nhiều triển khai thực tế khác nhau. “BLAS tham chiếu” được cung cấp bởi Netlib( những người cung cấp cho chúng ta phiên bản LAPACK phổ biến nhất)

## Gói đại số tuyến tính ( LAPACK)

Lapack là một tập hợp các thói quen đại số tuyến tính, ban đầu được viết bằng Fortran. Nó bao gồm các thói quen cao cấp hơn BLAS , chẳng hạn như đảo ngược ma trận .. Netlib đã thực hiện điều này. LAPACK yêu cầu Blas. Có thể kết hợp các triển khai LAPACK và BLAS.

CLAPACK là phiên bản LAPACK đã được tự động chuyển đổi từ Fortan sang C vang tiên ích f2c. Khí chúng ta nói về việc sử dụng LAPACL, chúng ta thực sự đang nói về việc sử dụng CLAPACK. Vì CLAPACK đã được chuyển đổi thành C bằng tiện ích f2c, khi chúng tôi liên kết với nói,chúng tôi cần bao gồm thư viện f2c.

## Machine learning

Nhận diện giọng nói đang xâm nhập vào cuộc sống hiện đại . Nó được cài đặt trong những chiếc điện thoại , điều khiển trò chơi hay những chiếc đồng hồ thông minh . Chỉ với khoảng 50$ bạn có thể có Amazon Echo Dot – một chiếc hộp thần kì cho phép bạn đặt pizza. Nhận thông tin dự báo thời tiết hoặc thậm chí ma những vật dụng chỉ bằng mệnh lệnh:



**Hình 3‑4 :** Echo Dot

Echo Dot trở nên phổ biến trong kỳ nghỉ đến nỗi Amazon cháy hàng .

.

Nhưng nhận diện giọng nói đã được biến đến hàng thấp kỷ , tại sao chỉ đến bây giờ công nghệ mới thực sự bùng nổ ? Sự ra đời của Deep learning đã giúp nhận diện giọng nói chính xác, thậm chí ở ngoài môi trường phòng lab.

Andrew Ng đã dự đoán từ lâu ngay khi độ chính xác của nhận diện giọng nói đạt ngưỡng 99% nó sẽ trở thành phương thức giao tiếp chủ yếu với máy tính . và nhờ có Deep learning cuối cùng chúng ta đã có thể chạm tới ngưỡng này

### Machine learning không phải lúc nào cũng là chiếc hộp đen kỳ diệu

Nếu bạn biết các mạng dịch thuật toán học bạn có thể đoán tằng : chúng ta chỉ cần truyền đoạn ghi âm vào mạng nowrron và đào tạo nó ra bản dịch



**Hình 3‑5 :** truyền đoạn ghi âm vào mạng nowrron và đào tạo nó ra bản dịch

Đó cũng là điều mà nhận diện giọng nói với deep learning hướng tới nhưng chúng ta chưa đạt đến trình độ đó (có thể cần phải thêm vài năm nữa) .

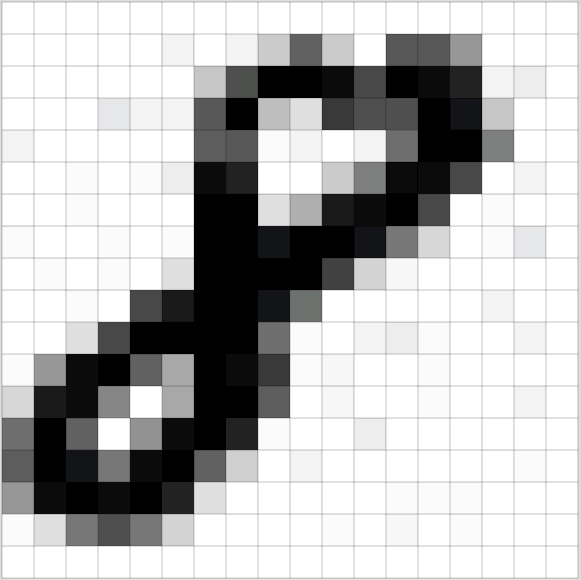
Vấn đề lớn nhất chính là tốc độ nói biến thiên. Một người có thể nói “Hello” rất nhanh và người khác nói “heeellllooooo” cực chậm , tạo ra âm thanh dài hơn với nhiều dữ liệu hơn. Cả 2 âm đều nên được nhận dạng chính xác là từ một “ hello”. Tự động chỉnh file âm thanh với nhiều chế độ dài khác nhau của từng từ để tạo ra văn bản đồng nhất lại khá khó .

Để xử lý vấn đề này, chúng ta sẽ sử dụng một số kỹ thuật đặc biệt và thêm một vài bước vào mạng deep learning .

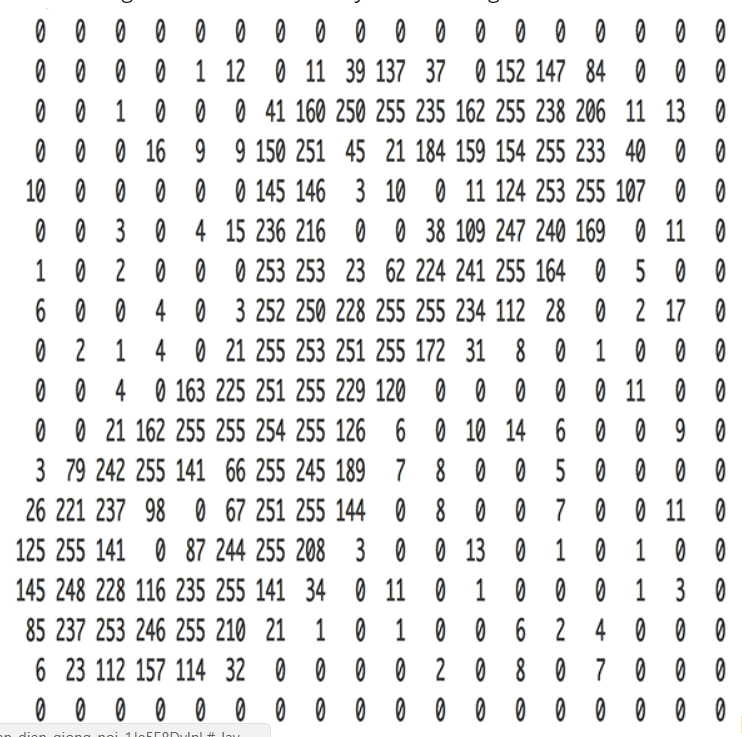
### Chuyển âm thanh thành số

Bước đầu tiên trong nhận diện giọng nói khá rõ ràng – chúng ta cần truyền sóng âm vào máy tính

Chúng ta học coi hình ảnh là tập hợp giá trị, với mỗi giá trị đại diện cho độ sáng của điểm ảnh, để truyền vào mạng nơron



**Hình 3‑6**: chuyển âm thanh thành số (1)

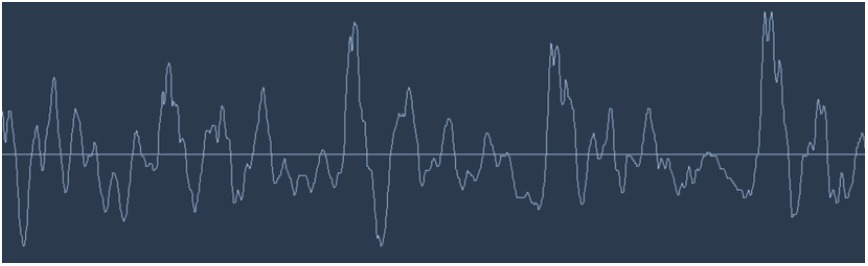


**Hình 3‑7** : chuyển âm thanh thành số (2)

Được truyền qua sóng âm .

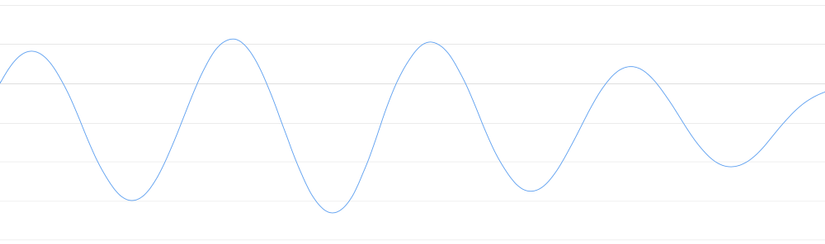
Làm thế nào để chúng ta chuyển sóng âm thành số?

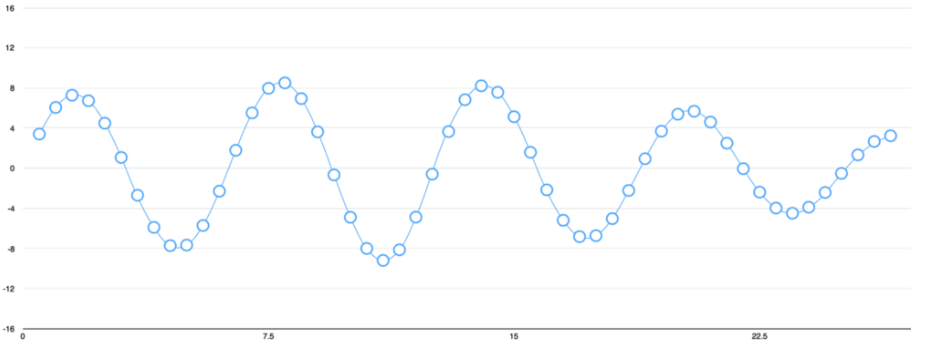
Hãy sử dụng đoạn âm nói “Hello” dưới đây :



**Hình 3‑8** : Chuyển âm thanh thành số (3)

Sóng âm có một chiều dữ liệu. Ở mọi thời điểm chúng ta có một giá trị cao độ. Hãy phóng to một đoạn nhỏ sóng âm để nhìn rõ hơn :





**Hình 3‑9** : Chuyển âm thanh thành số (3)





**Hình 3‑10 :**  Chuyển âm thanh thành số (4)

Phương pháp này gọi là sampling – lấy mẫu . Chúng ta đọc mẫu mỗi 1/1000s và ghi lại con số đại diện chiều cao của sóng âm. Đây chính là file.wav khi không bị nén.

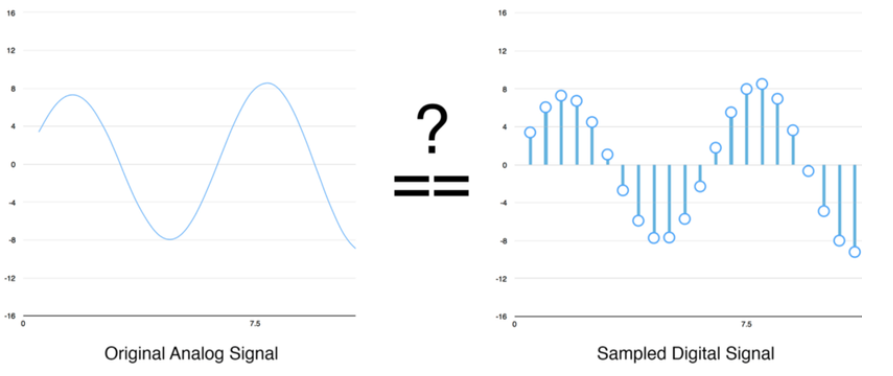
Nhưng âm thanh chất lượng tốt được ghi ở tần số 44.1khz (44,100 lần đọc mỗi giây). Nhưng với nhận diện giọng nói , tốc độ lấy mẫu ở 16khz (16,000 mẫu mỗi giây) là quá đủ.

Thử lấy mẫu giọng nói “hello ” trong ví dụ trên với tần số 16,000 lần mỗi giây .Và đây là 100 mẫu đầu tiên .

**Hình 3‑11** : Chuyển âm thanh thành số (5)

### Lấy mẫu liệu có được chính xác ?

Bạn có thể nghĩ rằng, lấy mẫu chỉ tạo ra đồ thị xấp xỉ so với sóng âm, bởi vì nó chỉ đọc dữ liệu theo từng khoảng .Liệu chúng ta có bị mất dữ liệu giữa mỗi lần đọc.



**Hình 3‑12 :** Mẫu đồ thị

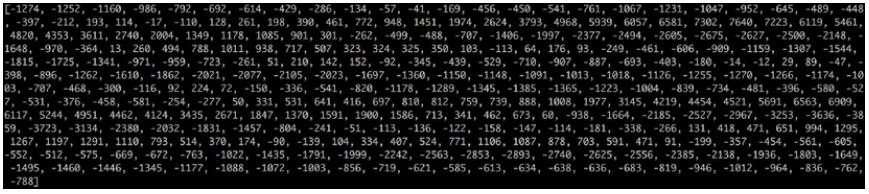
Nhờ có lý thuyết Nyquist , chúng ta có thể sử dụng toán học để tái tạo chính xác sóng âm gốc từ những mẫu tách biệt – miễn là chúng ta lấy mẫu với tần số gấp đôi tần số âm chúng ta muốn ghi lại. Không phải cứ lấy mẫu với tần số càng cao thì âm chất lượng âm thanh càng tốt .

### Tiền xử lý dữ liệu mẫu âm thanh

Chúng ta đã có dãy số với mỗi số đại diện cho cao độ âm tại 1/16000s

Ta có thể truyền những dãy số này vào mạng nơ ron, nhưng cố gắng nhận diện cấu trúc âm thanh trực tiếp bằng những mẫu này rất khó. Thay vào đó, chúng ta giải quyết vấn đề dễ hơn bằng cách tiền xử lý dữ liệu.

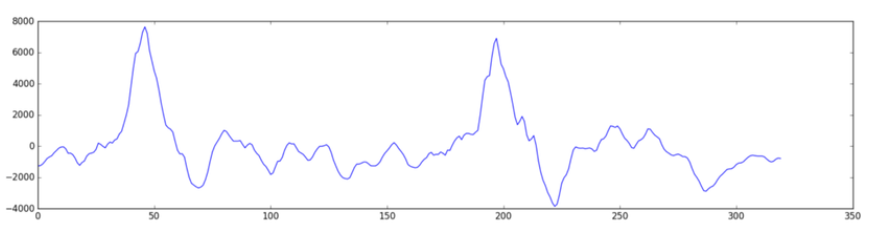
Hãy bắt đầu nhóm mẫu âm thanh trong khoảng 20ms .Và đây là 320 mẫu âm thanh tỏng 20ms đó :



**Hình 3‑13**: 320 mẫu âm thanh

Ghi lại những con số này

Trong đồ thị giúp chúng ta có ước lượng xấp xỉ về âm thanh gốc trong chu kỳ 20ms



**Hình 3‑14** : Đồ thị giúp chúng ta có ước lượng xấp xỉ về âm thanh gốc

Bản ghi âm này chỉ khoảng 1/50s. Nhưng thậm chí một đoạn ghi âm rất ngắn là một mớ hỗn độn cao độ âm khác nhau.Có những âm thấp , âm trung và thậm chí cả âm cao. Nhưng cùng với nhau , những âm này tạo nên giọng nói.

Để giúp mạng nơron xử lý dữ liệu dễ hơn,ta tách sóng âm phức tạp này thành từng phần : phần chứa âm thấp, âm cao hơn,cao hơn nữa … Sau đó, ta tính tổng năng lượng ở những dải tần số (từ thấp đến cao) và kết nối lại tạo ra fingerprint – nhận dạng duy nhất cho từng đoạn trích âm thanh .

## : Cách cài đặt kaldi sử dụng trên hệ điều hành ubuntu

Trước khi cài kaldi việc đầu tiền chúng ta phải làm là cài đặt Git.

Lệnh cài đặt : **sudo apt-get install git**

Tiếp theo cài đặt kaldi :

Lệnh cài đặt : **git clone https : //github.com/kaldi-ars/kaldi.git kaldi – trunk –orihin golden**

**Cd kaldi-trunk/tools/ ;make ;**

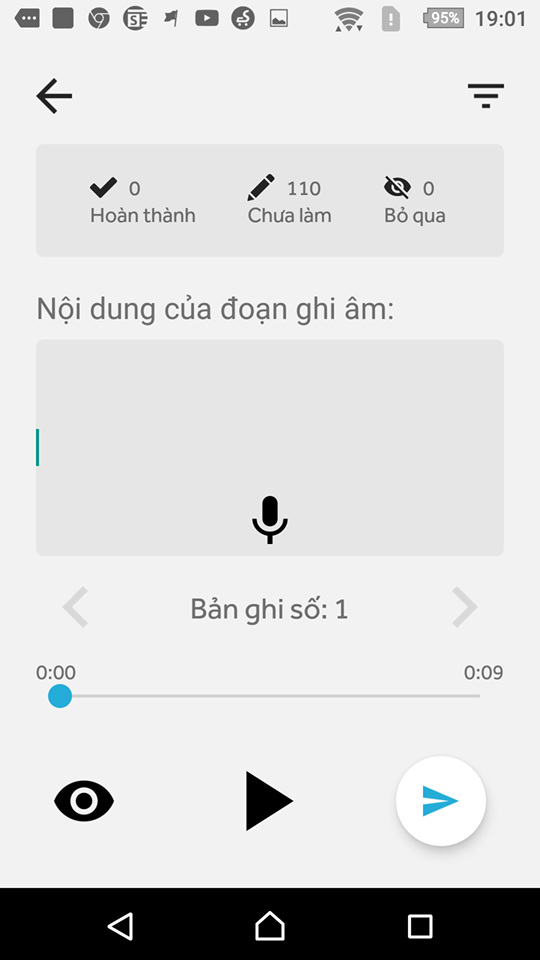
Tiếp theo nhập lệnh: **extras/install\_openblas.sh**

Tiếp theo : **cd ../src; --openblas-root = ../tools/OpenBLAS/install ; make**

## : Thu thập dữ liệu

Các tập dữ liệu có đuôi .Wav là đuôi file âm thanh dùng để giúp cho mô hình dữ liệu học giúp khả năng nhận dạng tiếp nói mô hình càng ngày càng chính xác hơn

## Gán nhãn dữ liệu



**Hình 3‑15 :** Ứng dụng về gán nhãn dữ liệu

Đây là mô hình gán nhãn dữ liệu .

Nhấn vào nút  để cho dữ liệu được đọc .

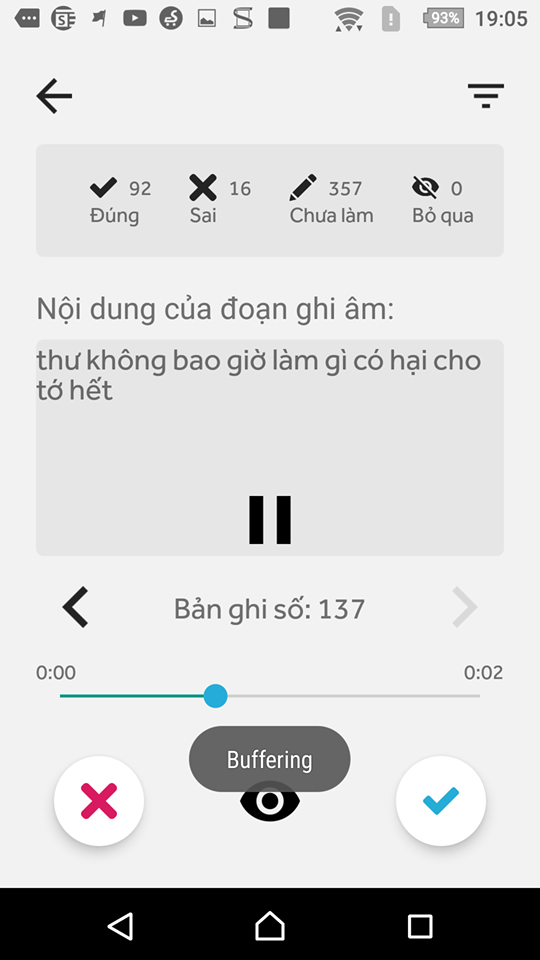
Sau đó những gì gì ta nghe được sẽ ghi vào phần *nội dung của đoạn ghi âm* là đoạn text .

Nếu dữ liệu chúng ta nghe rõ và điều xong vào đoạn text nhấn nút để gửi dữ liệu lên máy chủ

Nếu dữ liệu nghe không rõ không điền được chúng ta nhấn nút  để bỏ qua đoạn audio đó chuyển sang audio mới .

## Chuẩn hóa dữ liệu

Sau khi dữ liệu được làm xong thì bước tiếp theo chúng ta làm là chuẩn hóa dữ liệu



**Hình 3‑16** : Ứng dụng về chuẩn hóa dữ liệu

Bước chuẩn hóa dữ liệu và gán nhãn dữ liệu cần phải được làm chính xác ở phần chuẩn hóa ta sẽ nghe file audio và nhìn đoạn text

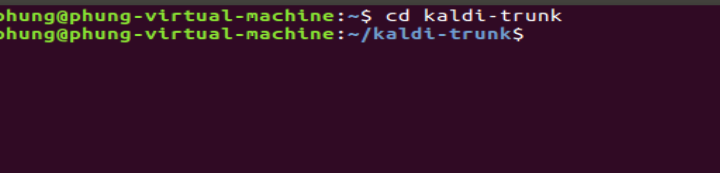
Nếu lời nói trong file audio và đoạn text trùng nhau chúng ta nhấn vào nút để gửi dữ liệu lên sever làm đoạn khác

Nếu đoạn dữ liệu và audio không không khớp nhau có thể do máy không nghe rõ thì  để báo sai đoạn dữ liệu

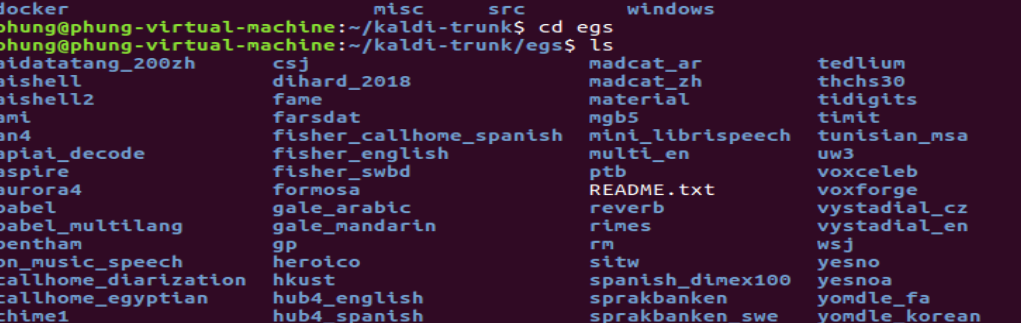
Hoặc nhấn nút  để bỏ qua đoạn audio.

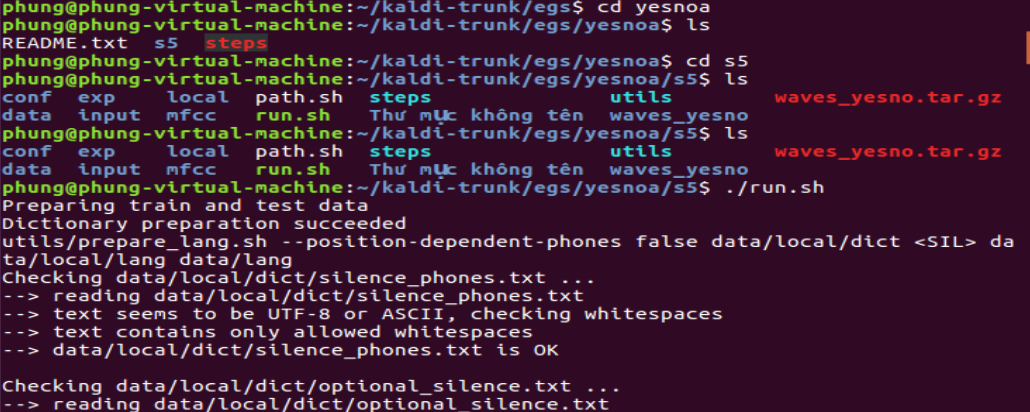
## : Chạy thử nghiệm mô hình nhận dạng dọng nói bằng kaldi

Vào thư mục

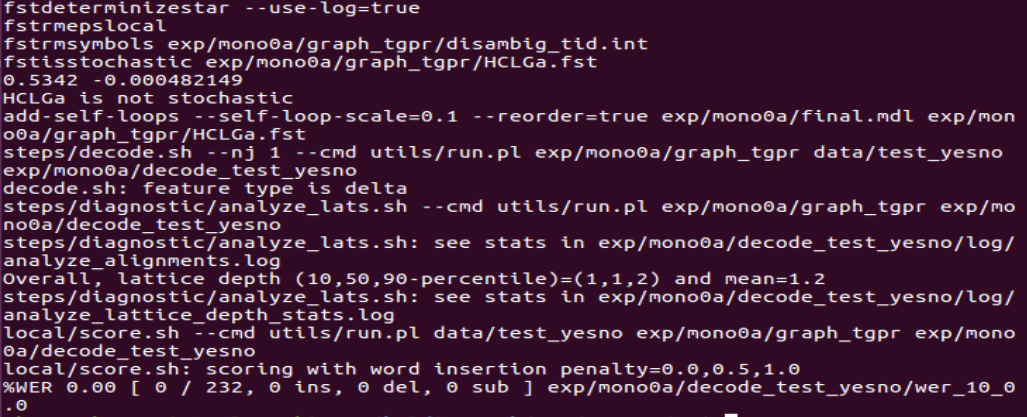


Trỏ vào thư mục chứa mô hình thử nghiệm :





**Hình 3‑17 :** Chạy thử nghiệm mô hình nhận dạng giọng nói



**Hình 3‑18 :** Chạy thử nghiệm mô hình nhận dạng giọng nói



# : kết luận và hướng phát triển

## Kết luận

Đồ án tốt nghiệp này được hoàn thành quá trình học tập và dưới sự hướng dẫn của thầy Nguyễn Thế Lộc em đã hoàn thành việc tìm hiểu về “ mô hình nhận dạng giọng nói”.

## Hướng phát triển

Ngày nay việc nhận dạng giọng nói ngày càng phát triển và ứng dụng trên các ứng dụng thông tin đại chúng trên điện thoại , máy tính, các thiết biết hiện đại .Việc phát triển mô hình nhận dạng ngày càng chuẩn xác và tối ưu hơn giúp cho việc con người sử dụng cho các thiết bị hiện đại càng dễ dàng không cần dùng trình điều khiển bằng tay .

Việc ứng dụng trong thực tế cao sẽ làm cho việc nhận dạng giọng nói sẽ ngày càng phát triển mạnh trong tương lai .

TÀI LIỆU THAM KHẢO

|  |  |
| --- | --- |
| [1] | A. Tuner, Introduction to Neogeography, Sebastopol, CA USA: O'Reilly Media, 2006. |
| [2] | Hoàng Anh Đức, Lê Văn Hưng, Thiết kế website, Hanoi: NXB Giáo dục Việt Nam, 2016. |
| [3] | "vi.scribd.com," Nguyễn Ngọc Đăng, [Online]. Available: https://vi.scribd.com/document/35954094/Tim-hi%E1%BB%83u-cac-ph%C6%B0%C6%A1ng-phap-phan-tich-%C4%91%E1%BA%B7c-tr%C6%B0ng-ti%E1%BA%BFng-noi. |
| [4] | "VN REVIEw," Phúc Thịnh, [Online]. Available: https://vnreview.vn/tin-tuc-khoa-hoc-cong-nghe/-/view\_content/content/2265035/phan-mem-nhan-dien-giong-noi-hoat-dong-nhu-the-nao. |
| [5] | [Online]. Available: https://tailieu.vn/doc/bai-giang-xu-ly-tieng-noi-truong-dai-hoc-hang-hai-viet-nam-1807682.html?fbclid=IwAR0fwH1N3mXfX1u\_QvYzE6Nb0hkEfIWFIyQPV-QVsloeme\_wkb35KY11oq4. |
| [6] | [Online]. Available: https://techblog.vn/framework-kaldi-su-dung-trong-nhan-dang-tieng-noi. |
| [7] | [Online]. Available: https://vi.wikipedia.org/wiki/BLAS. |
| [8] | n. đ. tùng. [Online]. Available: https://viblo.asia/p/machine-learning-that-thu-vi-6-nhan-dien-giong-noi-1Je5E8DylnL. |

# PHỤ LỤC

Các mã nguồn, các bảng biểu lớn, các phụ lục cho vào mục này. Trong phần đồ án, phần PHỤ LỤC không được tính vào tổng số trang của đồ án tốt nghiệp.

Các mã nguồn sử dụng Style “Code”, ví dụ như sau:

**<?php**

**echo “Hello world!”;**

**?>**