HUST

ĐẠI HỌC BÁCH KHOA HÀ NỘI

HANOI UNIVERSITY OF SCIENCE AND TECHNOLOGY

ONE LOVE. ONE FUTURE.

ĐỒ ÁN TỐT NGHIỆP

ĐẠI HỌC
BÁCH KHOA HÀ NỘI
HANOI UNIVERSITY
OF SCIENCE AND TECHNOLOGY

Đề tài:

THIẾT KẾ ỨNG DỤNG SOẠN THẢO VĂN BẢN TIẾNG VIỆT TỰ ĐỘNG TRÊN SMARTPONE SỬ DỤNG CÔNG NGHỆ NHẬN DẠNG TIẾNG NÓI

Sinh viên: Lưu Đình Tú
MSSV: 20213016 – EE2 07 K66
GVHD: TS. Trần Thị Anh Xuân

ONE LOVE. ONE FUTURE.

HUST

hust.edu.vn fb.com/dhbkhn

NỘI DUNG TRÌNH BÀY

1. TỔNG QUAN ĐỀ TÀI
2. CƠ SỞ LÝ THUYẾT
3. THIẾT KẾ HỆ THỐNG
4. THỬ NGHIỆM VÀ ĐÁNH GIÁ KẾT QUẢ
5. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

1.1 Đặt vấn đề

- **Soạn thảo văn bản tự động ứng dụng rộng rãi:**
  - **Tiết kiệm thời gian** nhập liệu
  - Hỗ trợ người khuyết tật
  - Hỗ trợ trong ứng dụng **nhắn tin**.

a, Trợ lý ảo thông minh

b, Hỗ trợ người khuyết tật

Hình 1.1 Ứng dụng hệ thống soạn thảo văn bản tự động

→ **Động lực thực hiện đề tài:** **Thiết kế ứng dụng soạn thảo văn bản tiếng Việt tự động trên Smartphone sử dụng công nghệ nhận dạng tiếng nói**

1.2 Mục tiêu và phạm vi

Mục tiêu:

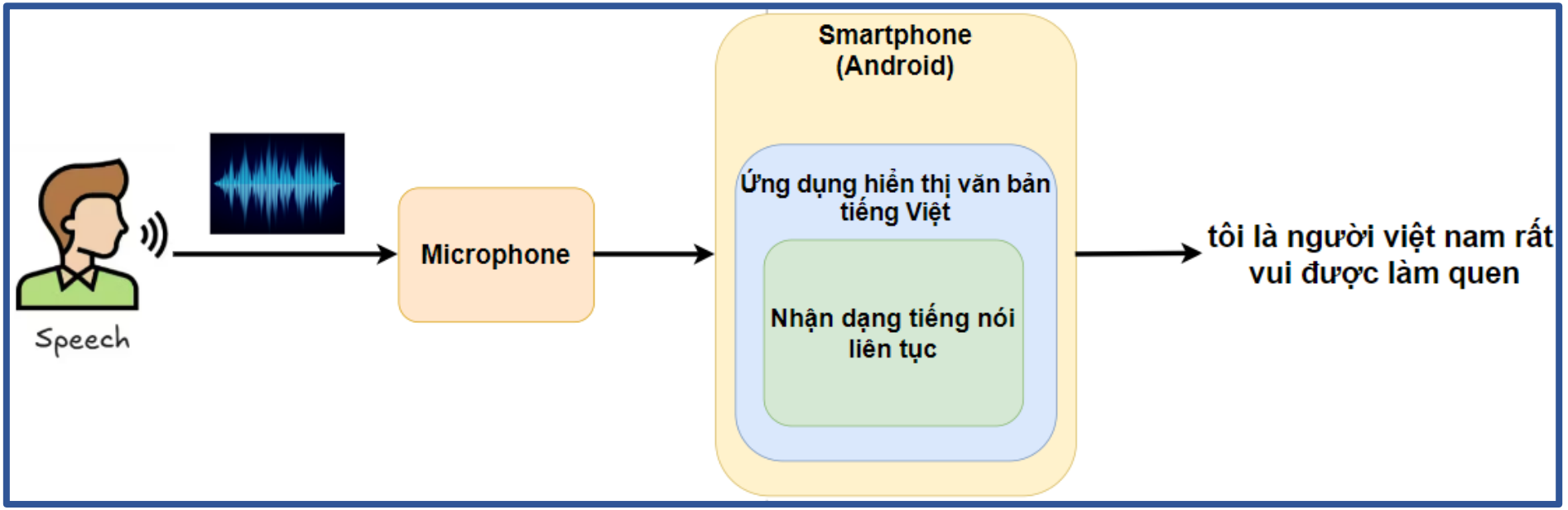
- Xây dựng ứng dụng **soạn thảo văn bản tiếng Việt bằng giọng nói** trên smartphone (HĐH Android)
- Tích hợp **hệ thống nhận dạng tiếng nói tiếng Việt** vào ứng dụng
- **Thử nghiệm và đánh giá** hiệu quả nhận dạng

Phạm vi đề tài:

- Triển khai **môi trường yên tĩnh**, ít nhiễu.
- Ứng dụng trên **smartphone Android**, giao diện đơn giản.
- Giọng **tiếng Việt chuẩn miền Bắc**.
- **Chuyển tiếng nói thành văn bản**
- **Chưa tích hợp** lệnh thoại hay xử lý ngôn ngữ nâng cao.

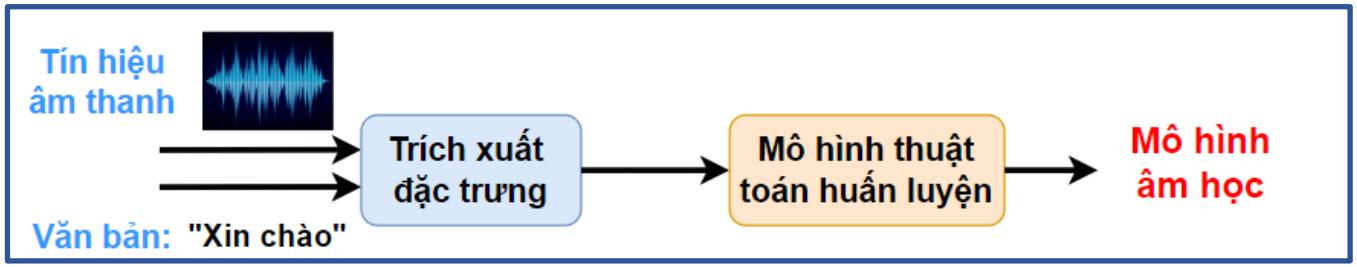
- Tổng quan hệ thống nhận dạng tiếng nói
- Những khó khăn của hệ thống nhận dạng tiếng nói liên tục
- Khối trích xuất đặc trưng
- Thuật toán huấn luyện nhận dạng tiếng nói
- Connectionist Temporal Classification (Phân loại thời gian kết nối)
- Mô hình ngôn ngữ (Language model)
- Khối giải mã
- Hệ điều hành Android

3.1 Đề xuất sơ đồ khối hệ thống

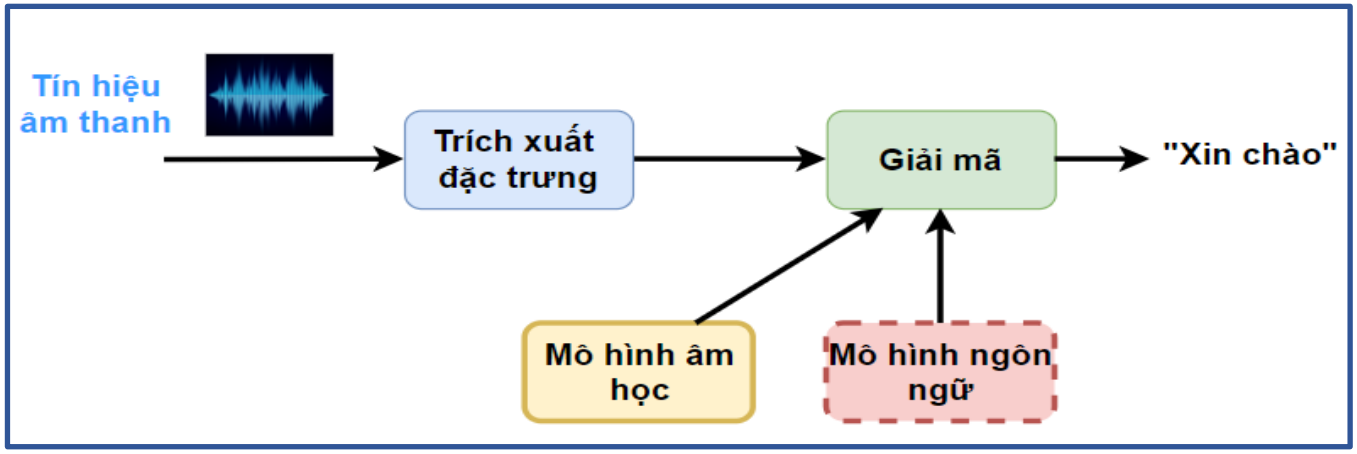


Hình 3.1 Sơ đồ khối hoạt động toàn bộ hệ thống

3.2 Thiết kế phần mềm nhận dạng tiếng nói tiếng Việt liên tục

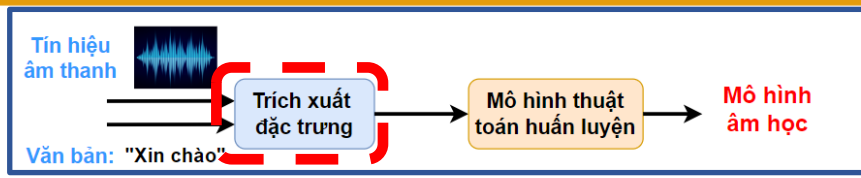


Hình 3.2 Quy trình huấn luyện mô hình trong hệ thống được đề xuất



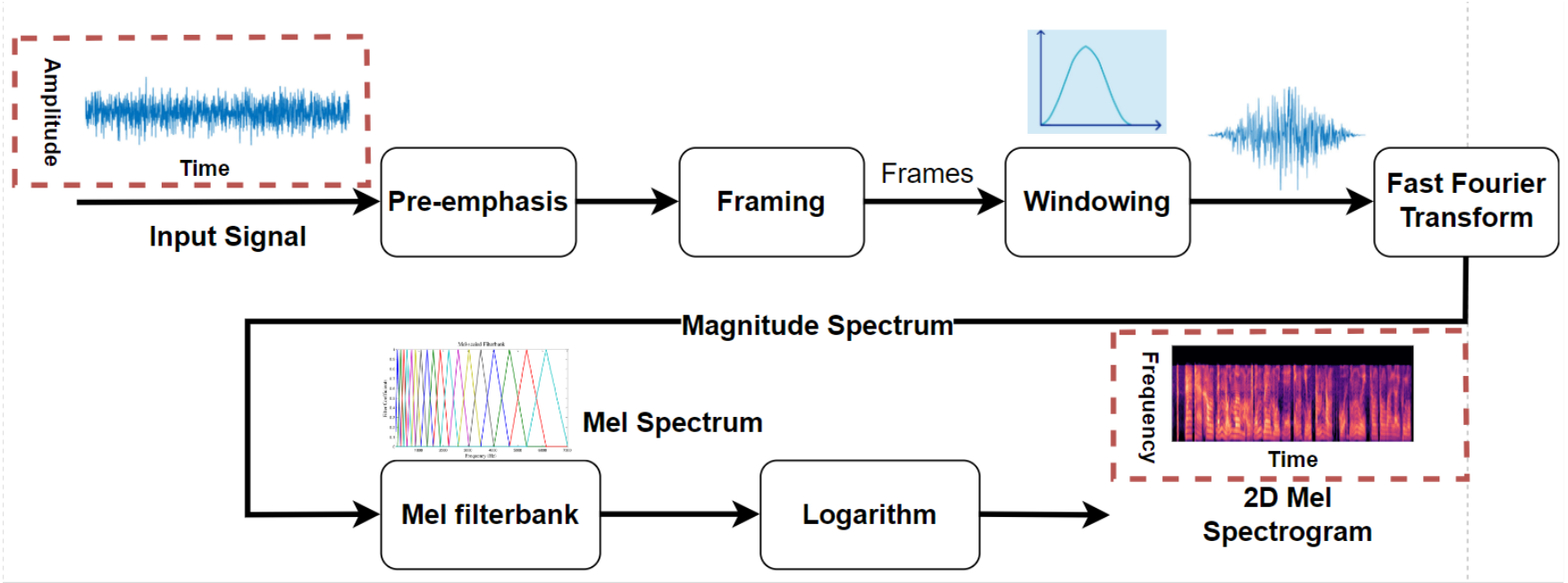
Hình 3.3 Quy trình nhận dạng tiếng nói được đề xuất

3.2.1 Khối trích xuất đặc trưng [2]



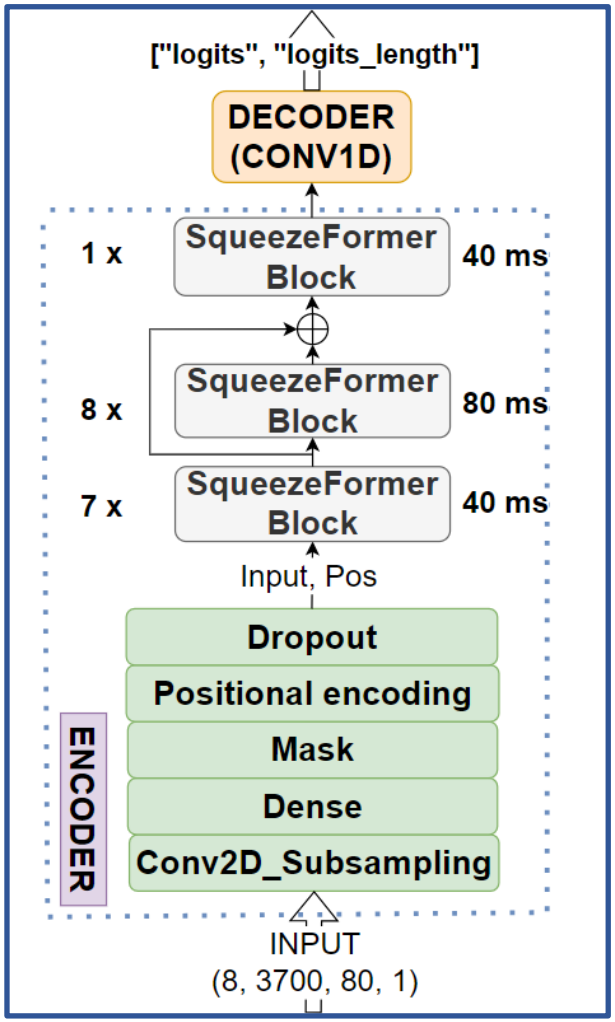
Tham số Mel-spectrogram:

- Tần số lấy mẫu: 16 kHz
- Cửa sổ: 25 ms
- Bước nhảy: 10 ms
- Số dải Mel: 80

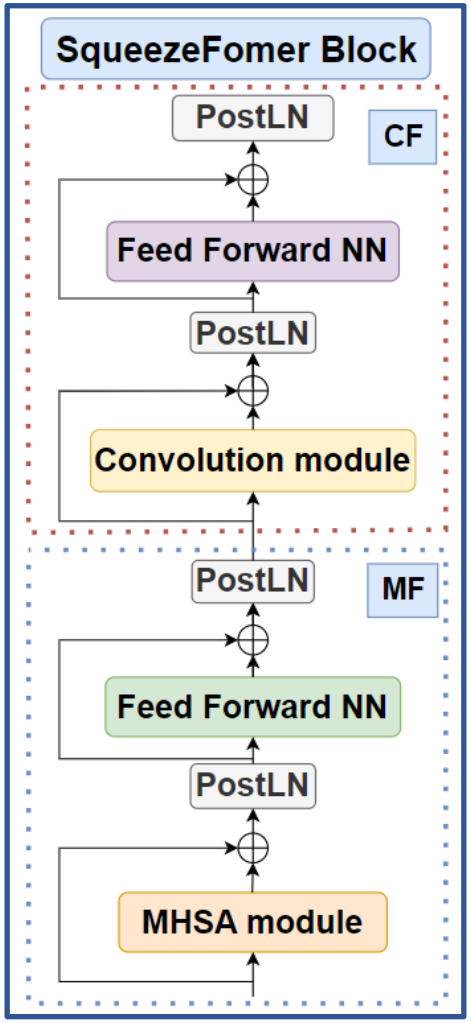


Hình 3.4 Luồng hoạt động của trích xuất Mel-spectrogram từ tín hiệu tiếng nói thô

3.2.2 Mô hình SqueezeFormer [1]



Hình 3.5 Mô hình SqueezeFormer



Hình 3.6 Khối SqueezeFormer

3.2.3 Xây dựng mô hình ngôn ngữ

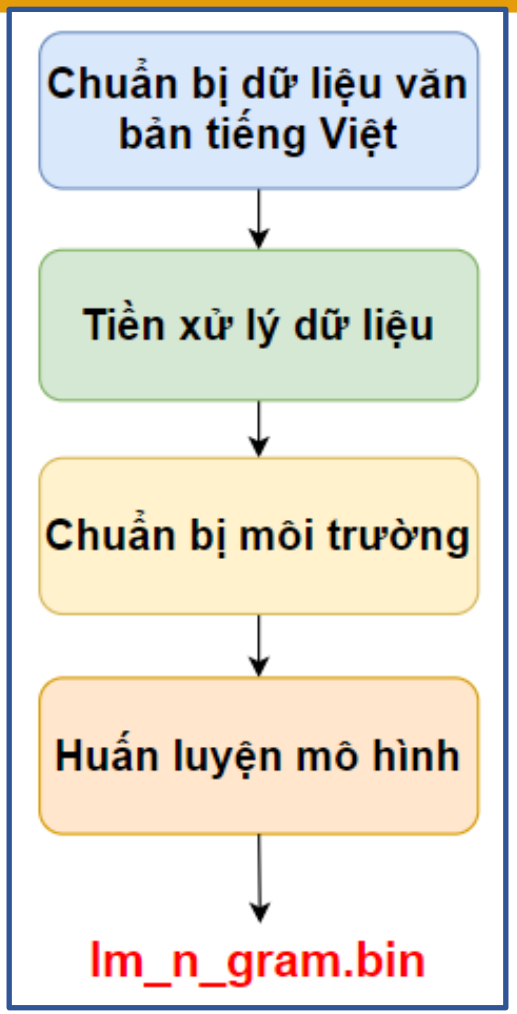
Công cụ sử dụng : KenLM

Cơ sở dữ liệu văn bản tiếng Việt:

- **Tập chuẩn:** VLSP 2020, FPT , VIVOS
- **Văn học:** Truyện cổ tích tiếng Việt
- **Báo chí:** Tuổi trẻ, báo Nhân Dân, 24h
- **Ngôn ngữ đời thường:** Chat, hội thoại

Tiền xử lý dữ liệu

- **Chuyển chữ hoa** thành **chữ thường**
- **Loại bỏ dấu câu, ký tự đặc biệt**
- **Chuẩn hóa các khoảng trắng**



Hình 3.7 Quy trình xây dựng mô hình ngôn ngữ n-gram

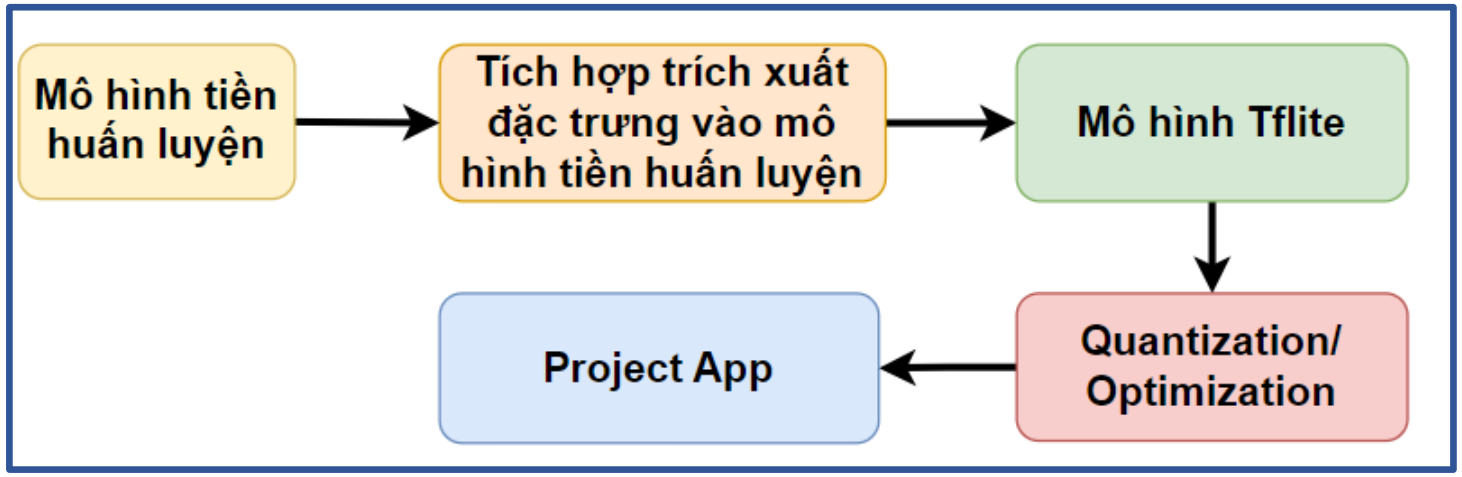
3.3.1 Thiết kế giao diện ứng dụng soạn thảo văn bản

- Công cụ sử dụng: Android Studio IDE
- Ngôn ngữ : Kotlin
- Các chức năng chính:
  - Nút ghi âm
  - Hiển thị kết quả rõ ràng

Hình 3.8 Giao diện ứng dụng hiển thị văn bản tiếng Việt

3.3.2 Tích hợp nhận dạng tiếng nói tiếng Việt vào ứng dụng

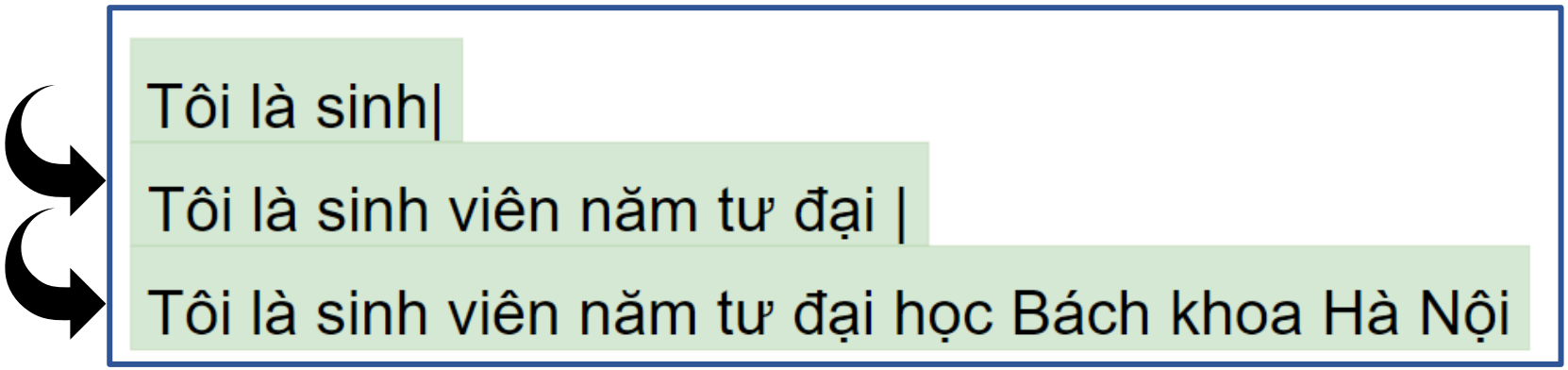
- **Bước 1:** Tích hợp **trích xuất đặc trưng (Mel-spectrogram)** trong **mô hình đã huấn luyện**
- **Bước 2:** Chuyển mô hình **TensorFlow sang** định dạng **.tflite**
- **Bước 3:** Tối ưu mô hình bằng kỹ thuật **lượng tử hóa**
- **Bước 4:** Xuất mô hình .tflite, **tích hợp vào ứng dụng hiển thị văn bản tiếng Việt.**



Hình 3.9 Quy trình chuyển mô hình tensorflow sang định dạng .tflite

3.4.1 Vai trò của nhận diện tiếng nói Online

- Xử lý tiếng nói **thời gian thực** với **độ trễ thấp**
- **Hỗ trợ các ứng dụng như:**
  - ❑ **Soạn thảo văn bản bằng tiếng nói**
  - ❑ **Điều khiển bằng giọng nói**
  - ❑ **Trợ lý ảo, giao tiếp người –máy**
- Tăng **trải nghiệm người dùng** nhờ phản hồi gần như tức thì

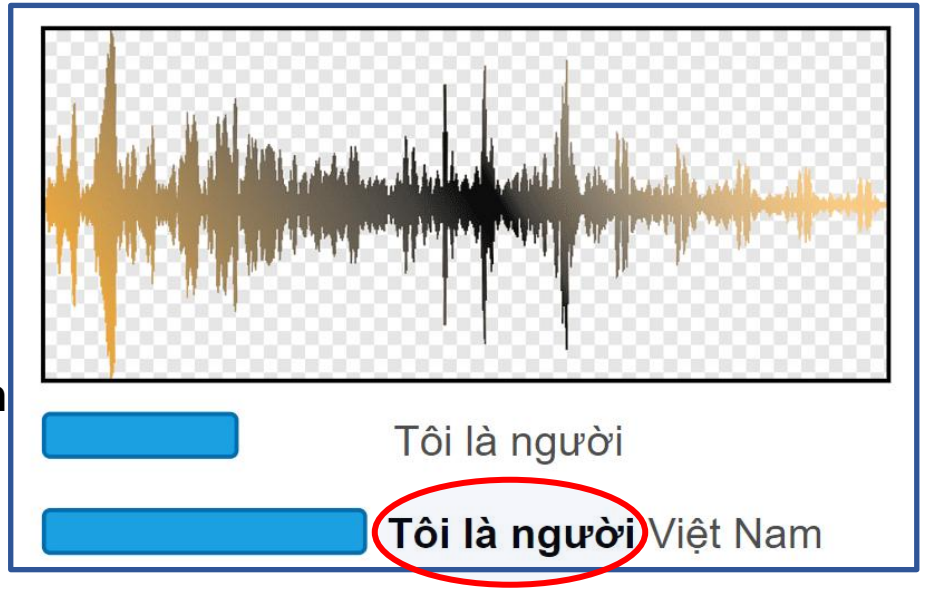


Hình 3.10 Văn bản được in ra trong khi đang ghi âm ở chế độ online ASR

3.4.2 Thuật toán Local Agreement [3]

Ý tưởng đề xuất:

- **Audio buffer lưu liên tục** tín hiệu âm thanh đầu vào
- Mỗi **t = 1.75 giây**, buffer cập nhật vào mô hình nhận dạng
- **So sánh kết quả liên tiếp** tìm **chuỗi văn bản chung dài nhất** – "đầu ra xác nhận".
- Chuỗi xác nhận được cố định.
- Khi không có chuỗi chung, toàn bộ văn bản còn lại được in ra.



Hình 3.11 Minh họa Local Agreement

4.1 Phân chia dữ liệu sử dụng

- **Không trùng lặp người nói/ tệp âm thanh** → tính khách quan & tổng quát
- **Chia người nói** thành 3 nhóm: **train / valid / test**
- **Phân bố số lượng tệp hợp lý** → **cân bằng hiệu quả** huấn luyện

❑ **VLSP**

Tập dữ liệu	Loại	Số lượng (files)	Cách chia	Thời lượng
Speaker	Train	11064	600 người	12h51m
	Valid	1412	80 người	1h38m
	Test	1387	80 người	1h35m
Database	Train	15621	Tỷ lệ 8:1:1	20h47m
	Valid	2000		2h23m
	Test	2000		2h21m

Bảng 4.1 Phân chia dữ liệu tập VLSP 2020

4.1 Phân chia dữ liệu sử dụng

❑ VIVOS

Tập dữ liệu	Loại	Số lượng (files)	Cách chia	Thời lượng
VIVOS	Train	9875	40 người	10h
	Valid	1698	6 người	1h47m
	Test	757	19 người	45m

Bảng 4.2 Phân chia dữ liệu tập VIVOS

❑ FPT & CStt

Tập dữ liệu	Loại	Số lượng (files)	Cách chia	Thời lượng
FPT	Train	6317	Tỷ lệ 8:1:1	7h21m
	Valid	1000		53m
	Test	1000		52m
CStt	Test	1190	33 người	98m

Bảng 4.3 Phân chia dữ liệu tập FPT & CStt

4.2 Một số thông số đánh giá chất lượng hệ thống

❖ Word Error Rate (WER)

Ý nghĩa: WER càng thấp thì **độ chính xác** nhận dạng **càng cao** và ngược lại

Công thức tính:

$$WER = \frac{S + I + D}{N} \quad \text{(Pt 4.1)}$$

Trong đó:

- **S (Substitutions):** số từ **bị thay thế**
- **I (Insertions):** số từ bị **chèn thêm** không đúng
- **D (Deletions):** số từ **bị thiếu hoặc bị xóa**
- **N: tổng số từ** trong bản **phiên âm đúng**

4.2 Một số thông số đánh giá chất lượng hệ thống

❖ Real-Time Factor (RTF)

Công thức tính:

$$RTF = \frac{T_{processing}}{T_{input}} \quad \text{(Pt 4.2)}$$

Trong đó:

- $T_{processing}$: **thời gian** hệ thống nhận dạng cần để **xử lý** âm thanh đầu vào
- T_{input}: **thời lượng** thực tế của **đoạn âm thanh đầu vào.**

❖ Latency (Độ trễ)

Công thức tính:

$$Latency = T_{output} - T_{start} \quad \text{(Pt 4.3)}$$

Trong đó:

- T_{output}: **thời gian hệ thống trả về kết quả đầu tiên**
- T_{start}: **thời điểm hệ thống bắt đầu xử lý nhận dạng**

4.3 Thông số huấn luyện

❖ **Tốc độ học:**

- **Chiến lược: Linear Decay**
- **Giá trị khởi tạo: 0.001**
- **Giảm dần tuyến tính sau mỗi epoch**
- **Giá trị nhỏ nhất: 1e-5 (sau 160 epoch)**
- **Mục tiêu: Học nhanh ban đầu, ổn định cuối, tránh dao động và quá khớp**

❖ **Batch size:**

- **Giá trị: 32**
- **Ưu điểm:**
  - ➢ **Tăng tốc huấn luyện** nếu tăng dữ liệu
  - ➢ **Cân bằng** giữa **hiệu suất và** khả năng **tổng quát hóa**
  - ➢ **Hạn chế ảnh hưởng** của **nhiễu** từ mẫu nhỏ.

4.4 Thử nghiệm 1: Đánh giá chất lượng mô hình âm học

Dữ liệu sử dụng: Speaker + VIVOS

Mục tiêu :

- Lựa chọn **mô hình âm học tối ưu**
- Cân bằng **chất lượng nhận dạng** và **tính khả thi huấn luyện**
- Cấu hình fine-tuning phù hợp để áp dụng cho dữ liệu thử nghiệm.

Cách tiến hành:

- Chiến lược :
  - **Decoder:** mở khóa hoàn toàn (**unfreeze**)
  - **Encoder**: **mở dần dần** từ **lớp cao nhất**
- Thử nghiệm **2 phương án** số lớp đầu ra của mô hình âm học
  - **128 lớp :** bao gồm **từ phụ trợ, ký tự duy nhất**
  - **93 lớp:** Chỉ gồm **ký tự trong bảng chữ cái tiếng Việt.**.

4.4.2 Phương án 1: Sử dụng 128 lớp đầu ra cho mô hình âm học

- **128 lớp đầu ra của mô hình âm học.**

“”, “<s>”, “</s>”, “_t”, “ng”, “_c”, “_đ”, “nh”, “_v”, “_th”, “_l”, “_h”, “_m”, “_b”, “_ch”, “_tr”, “_n”, “_k”, “_nh”, “_s”, “_ng”, “_g”, “_kh”, “_p”, “_d”, “_ph”, “_là”, “ông”, “iệ”, “_cá”, “_q”, “_qu”, “_gi”, “_và”, “ên”, “_r”, “ườ”, “_x”, “ới”, “_có”, “_củ”, “_”, “n”, “h”, “t”, “i”, “c”, “g”, “a”, “m”, “u”, “đ”, “à”, “o”, “ư”, “v”, “l”, “r”, “á”, “y”, “b”, “p”, “ô”, “k”, “s”, “ó”, “ế”, “ạ”, “ộ”, “ờ”, “ệ”, “ả”, “ê”, “ì”, “d”, “â”, “ố”, “ớ”, “ấ”, “ơ”, “ề”, “q”, “ủ”, “ể”, “ă”, “ị”, “ợ”, “í”, “ậ”, “e”, “x”, “ầ”, “ự”, “ú”, “ữ”, “ọ”, “ứ”, “ã”, “ở”, “ồ”, “ụ”, “ắ”, “ừ”, “ổ”, “ò”, “ũ”, “ù”, “ặ”, “ý”, “ỉ”, “ẽ”, “ỏ”, “ử”, “ằ”, “é”, “ĩ”, “ễ”, “ẩ”, “ẫ”, “ỗ”, “ẹ”, “ỹ”, “ẻ”, “ỳ”, “è”, “õ”, “ỡ”, “ẳ”

Trường hợp 1: Unfreeze khối Decoder

Trường hợp 2: Unfreeze SqueezeFormer Block 15, Dense Encoder và Decoder

Trường hợp 3: Unfreeze SqueezeFormer Block 14-15, Dense Encoder và Decoder

Trường hợp 4: Unfreeze SqueezeFormer Block 13-15, Dense Encoder và Decoder

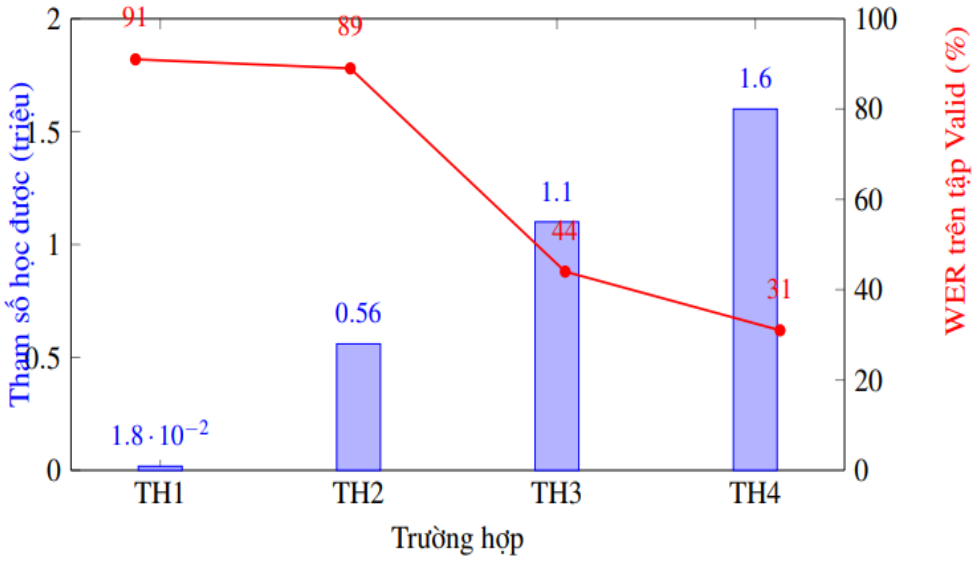
4.4.2 Phương án 1: Sử dụng 128 lớp đầu ra cho mô hình âm học

Kết quả tốt nhất TH4:

- **WER** trên tập **Validation: 31%**
- **WER** trên tập **Test: 36%**
- **Số tham số huấn luyện: 1.6** triệu

Nhận xét:

- Càng **nhiều tham số huấn luyện**
  → chất lượng mô hình càng
  cải thiện
- **1.6 M tham số** là mức **cân bằng tốt nhất** giữa **hiệu suất** và **giới hạn phần cứng**



Hình 4.1 Ảnh hưởng của số tham số học được đến WER trên tập Valid

4.4.3 Phương án 2: Sử dụng 93 lớp đầu ra cho mô hình âm học

- **93 lớp đầu ra của mô hình âm học**

“”, “<s>”, “</s>”, “ ”, “n”, “h”, “t”, “i”, “c”, “g”, “a”, “m”, “u”, “đ”, “à”, “o”, “ư”, “l”, “v”, “r”, “y”, “á”, “b”, “ô”, “p”, “k”, “s”, “ó”, “é”, “ạ”, “ộ”, “ờ”, “ả”, “ê”, “ệ”, “d”, “ì”, “â”, “ố”, “ấ”, “ớ”, “ơ”, “ề”, “q”, “ủ”, “ể”, “ợ”, “ị”, “ă”, “ậ”, “x”, “í”, “ầ”, “e”, “ú”, “ữ”, “ự”, “ọ”, “ã”, “ứ”, “ở”, “ồ”, “ắ”, “ụ”, “ò”, “ừ”, “ổ”, “ũ”, “ù”, “ặ”, “ỉ”, “ẽ”, “ý”, “ỏ”, “ử”, “é”, “ằ”, “ĩ”, “ễ”, “ẩ”, “ẫ”, “ỗ”, “ẹ”, “ẻ”, “ỹ”, “è”, “ỳ”, “õ”, “ẳ”, “ỡ”, “ẵ”, “ỷ”

- **Giữ nguyên** dữ liệu đánh giá so với Phương án 1 **(Speaker + VIVOS)**
- **Thay đổi** so với **Phương án 1 – Trường hợp 4:**
  - ❑ **Giảm đầu ra**: từ **128** → **93 lớp**
  - ❑ **Điều chỉnh fine-tuning**:
    - ➢ Unfreeze **khối SqueezeFormer 6**, **14**, **15 ở encoder** (thay vì 13, 14, 15)
    - ➢ **Giữ nguyên**: lớp Dense trong encoder & decoder

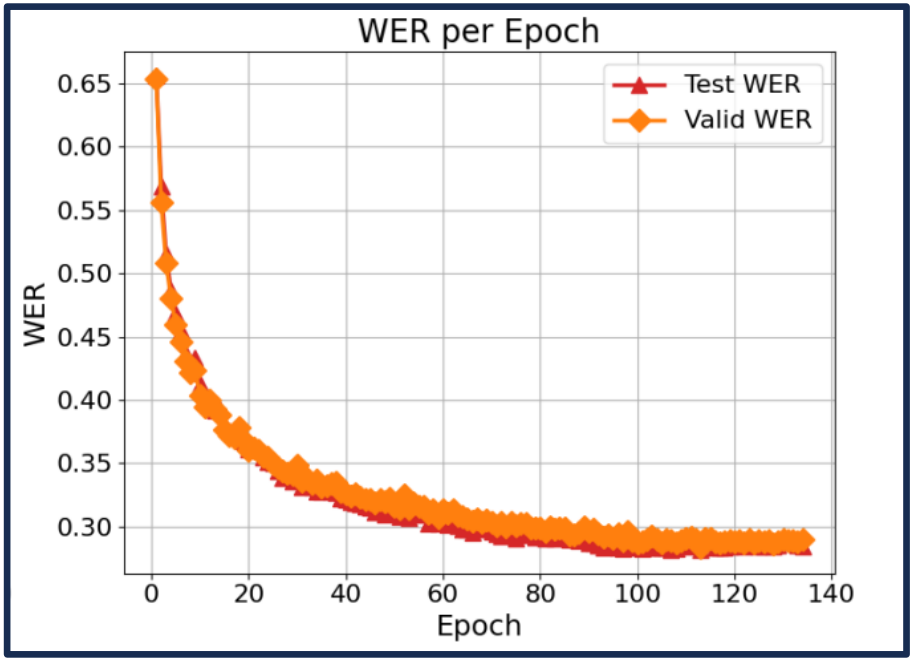
4.4.3 Phương án 2: Sử dụng 93 lớp đầu ra cho mô hình âm học

Nhận xét:

- **Không overfit**, học **ổn định** trong suốt quá trình huấn luyện.
- **WER đạt 28 %, 26%** trên tập **Valid, Test**
- Khối **SqueezeFormer Block 6** hoạt động ở mức thời gian **40 ms**

  → **bảo toàn thông tin âm thanh chi tiết hơn**

Mô hình đạt **hiệu quả tốt nhất trong Thử nghiệm 1**
→ Phù hợp với **giới hạn tài nguyên phần cứng hiện có**.



Hình 4.2 Kết quả WER theo epoch trên tập Valid, Test

4.5 TN2: Đánh giá vai trò kích thước dữ liệu huấn luyện với mô hình nhận dạng

- **Mục tiêu:**
  - **Đánh giá vai trò của kích thước dữ liệu huấn luyện với mô hình nhận dạng tiếng nói**
- **Cách tiến hành:**
  - Giữ nguyên phương pháp **unfreeze thử nghiệm 1 – Phương án 2:**
    - Unfreeze **khối SqueezeFormer 6, 14, 15 ở encoder**
    - Lớp **Dense trong encoder** & toàn bộ **decoder**
  - **Tăng dữ liệu** huấn luyện và đánh giá:
    - **Trường hợp 1: Tăng tập dữ liệu (thêm 7000 files trong Database so với Thử nghiệm 1)**
    - **Trường hợp 2: Tăng tập dữ liệu** (thêm **tập FPT + 3800 files trong Database** so với trường hợp 1)
    - **Trường hợp 3: Tăng tập dữ liệu (thêm 8200 files trong Database so với trường hợp 2)**

4.5 TN2: Đánh giá vai trò kích thước dữ liệu huấn luyện với mô hình nhận dạng

Tập dữ liệu	Loại	WER TH1 (%)	WER TH2 (%)	WER TH3 (%)
Speaker + VIVOS	Train	18.90	18.03	19.95
	Valid	27.23	25.40	25.38
	Test	26.40	24.36	24.50
FPT	Train	33.93	23.02	23.23
	Valid	35.15	28.67	28.61
	Test	34.43	28.57	28.12
Database X (TH1 : X = 1, TH2 : X = 2, TH3: X = 3)	Train	23.62	26.60	31.93
	Valid	37.41	33.98	40.54
	Test	37.70	32.16	40.72
Tự thu	Test	34.04	32.17	29.80

Bảng 4.4 Kết quả WER no LM (Beam search) trên các tập dữ liệu các trường hợp thử nghiệm 2

4.6 Thử nghiệm 3: Tối ưu hóa tham số Alpha và Beta

Mục tiêu:

- Tìm bộ tham số **Alpha (α)** và **Beta (β)** tối ưu
- Cải thiện hiệu suất **giải mã kết hợp** giữa mô hình **âm học** & mô hình **ngôn ngữ**

Ý nghĩa:

- **Alpha (α):** Tăng ảnh hưởng của mô hình ngôn ngữ (LM)
- **Beta (β):** Điều chỉnh thưởng/phạt theo **độ dài câu**

Cách tiến hành:

- **Đánh giá trên 300 files âm thanh từ tập test Speaker + VIVOS sử dụng Beam Search + LM**
- **Quét tham số:**
  - **Alpha, Beta:** từ **0.2 → 1.5** (bước 0.2)
  - Với mỗi cặp (α, β), giải mã toàn bộ tập thử nghiệm và đánh giá bằng **WER**

→ Chọn được **bộ số (α, β):** **(1.2, 1.5)**

4.7 Thử nghiệm 4: Đánh giá vai trò mô hình ngôn ngữ

- **Mục tiêu :**
  - Đánh giá hiệu quả của **mô hình ngôn ngữ 4-gram** trong hệ thống nhận dạng tiếng nói tiếng Việt
- **Cách tiến hành:**
  - Dùng mô hình huấn luyện từ **Thử nghiệm 1 – Phương án 2** (93 lớp đầu ra)
  - Đánh giá lại trên các trường hợp trong **Thử nghiệm 2**, có tích hợp 4-gram
  - Áp dụng giải mã **Beam Search + LM 4-gram**
  - Sử dụng bộ tham số **Alpha = 1.2, Beta = 1.5** (chọn từ **Thử nghiệm 3**)
  - Đánh giá kết quả bằng chỉ số **WER**

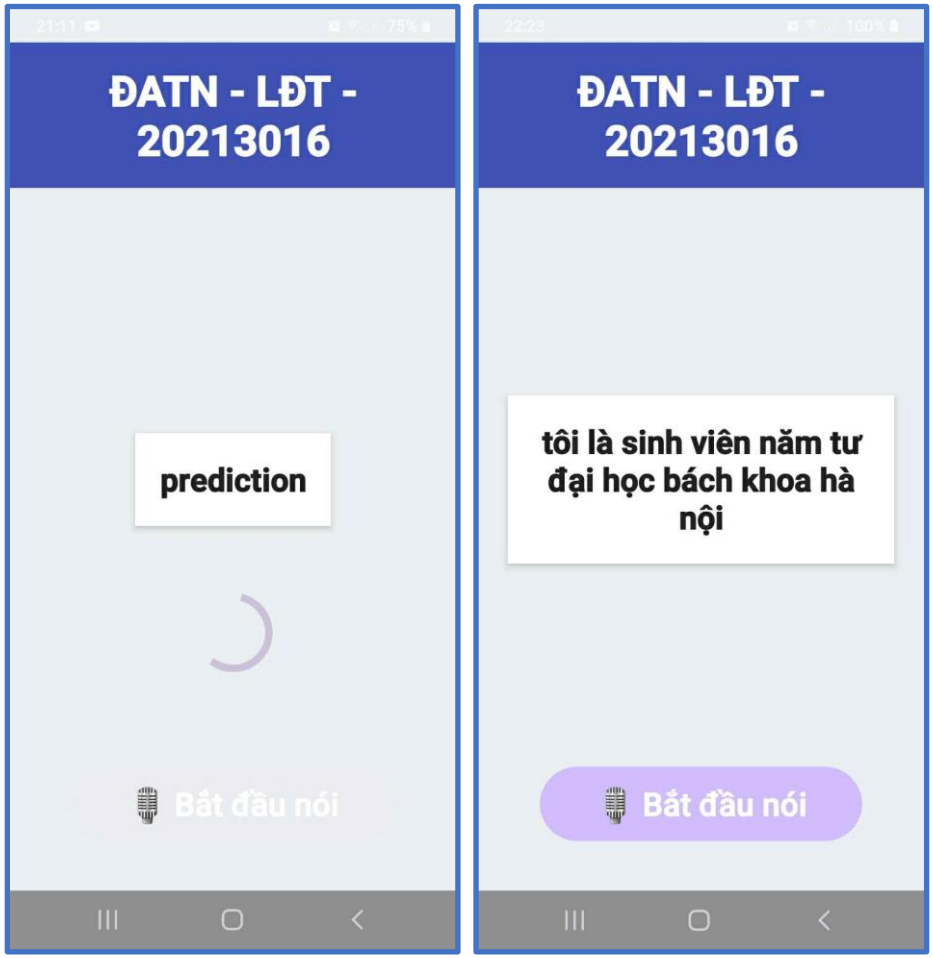
4.7 Thử nghiệm 4: Đánh giá vai trò mô hình ngôn ngữ

Tập dữ liệu	Loại	WER no LM (%)	WER with LM 4-gram (%)
Speaker + VIVOS	Train	19.95	3.53
	Valid	25.38	5.88
	Test	24.50	5.89
Database	Train	23.23	12.10
	Valid	28.61	18.72
	Test	28.12	19.02
FPT	Train	31.93	4.52
	Valid	40.54	8.20
	Test	40.72	7.71
Tự thu	Test	29.80	7.54

Bảng 4.5 Kết quả WER (sử dụng LM 4-gram) Thử nghiệm 2 – trường hợp 3

4.8 Thử nghiệm 5: Tích hợp mô hình âm học vào ứng dụng trên Smartphone

- **Mục tiêu :**
  - Đánh giá khả năng tích hợp **mô hình âm học** vào ứng dụng **hiển thị văn bản tiếng Việt** trên **thiết bị Android**
- **Cách tiến hành:**
  - Chọn mô hình có hiệu quả tốt nhất từ **Thử nghiệm 2 – Trường hợp 3**
  - Chuyển mô hình sang định dạng .tflite
  - Tích hợp model.tflite vào ứng dụng Android



Hình 4.3 Quy trình xử lý và kết quả nhận dạng trên ứng dụng Android

4.9 Thử nghiệm 6: Đánh giá hiệu quả chế độ nhận dạng tiếng nói online

- **Mục tiêu :**
  - Đánh giá **chất lượng & độ ổn định** hệ thống ở chế độ **online ASR**
- **Cách tiến hành:**
  - **Xây dựng cơ chế ASR trực tuyến**
  - Sử dụng mô hình âm học từ **Thử nghiệm 2 – Trường hợp 3,** kết hợp:
    - ❑ **Beam Search**
    - ❑ Mô hình ngôn ngữ **4-gram**
  - Thiết kế hệ thống gồm 2 luồng **xử lý song song:**
    - ❑ **Luồng thu âm**
    - ❑ **Luồng nhận dạng:** Xử lý mỗi 1.75 giây

❖ Online ASR trên laptop

```
--- Bắt đầu ghi âm mới ---

hôm nay là thứ
```

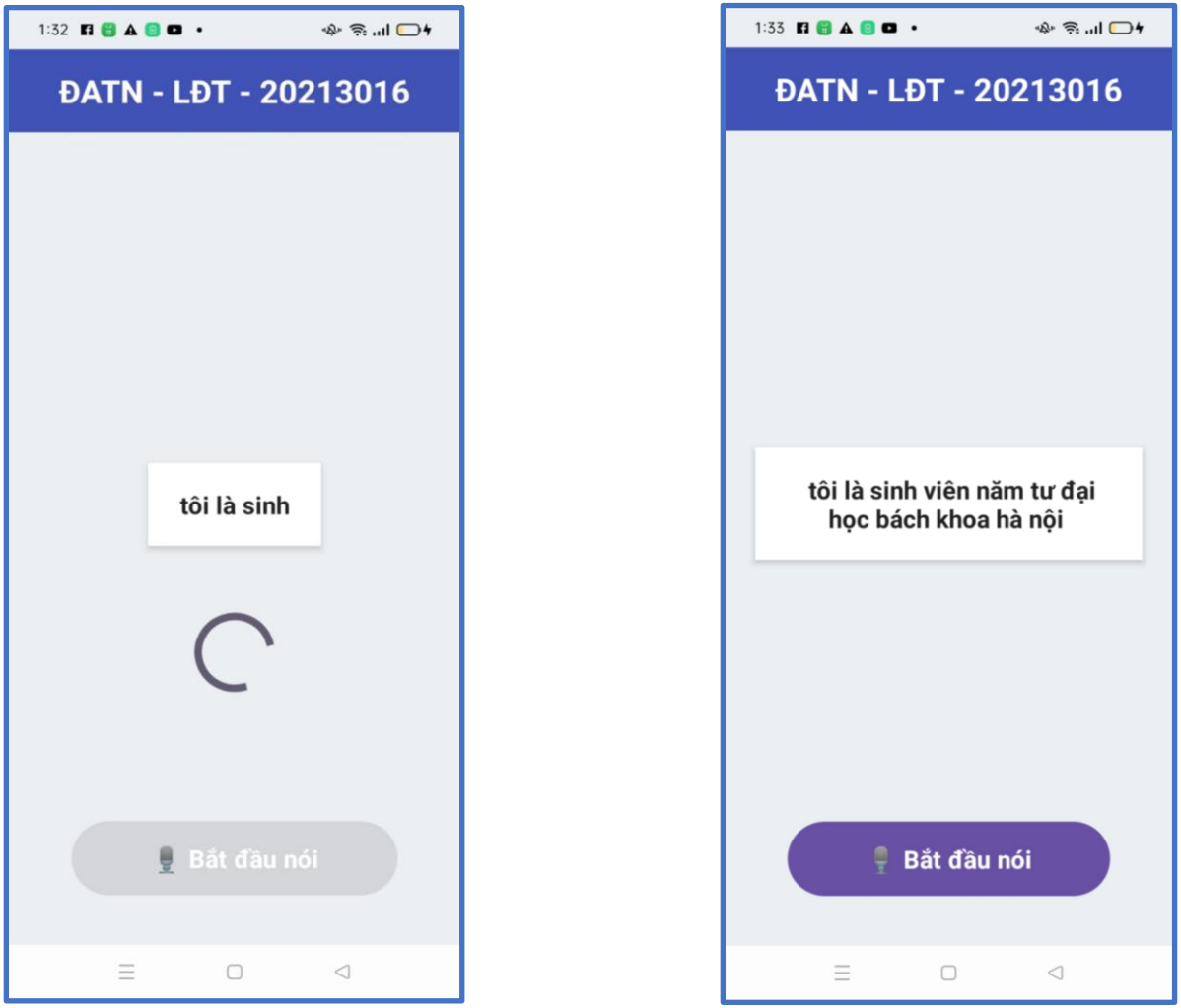
```
--- Bắt đầu ghi âm mới ---

hôm nay là thứ hai trời nắng
```

```
--- Bắt đầu ghi âm mới ---

hôm nay là thứ hai trời nắng ngày mai là thứ ba trời
```

```
--- Bắt đầu ghi âm mới ---

hôm nay là thứ hai trời nắng ngày mai là thứ ba trời có thể mưa giông
```

Hình 4.4 Thử nghiệm chế độ online ASR trên laptop

❖ Online ASR trên ứng dụng Smartphone



Hình 4.5 Thử nghiệm chế độ online ASR trên ứng dụng trên smartphone

4.10 Thử nghiệm 7: Đánh giá chất lượng hệ thống với RTF và Latency (trên laptop)

- **Chế độ offline ASR:**
  - **Số câu thử nghiệm:** 7 câu (độ dài tối đa 8 giây)
  - **Độ trễ trung bình (Latency): ≈ 0.56 giây**
  - **Hệ số thời gian thực (RTF): ≈ 0.097**
- **Chế độ online ASR:**
  - **Số câu thử nghiệm:** 7 câu (độ dài tối đa 8 giây)
  - **Độ trễ trung bình (Latency): ≈ 0.59 giây**
  - **Hệ số thời gian thực (RTF): ≈ 0.1627**

4.10 TN7: Đánh giá chất lượng hệ thống với RTF và Latency (ứng dụng Android)

- **Offline ASR:**

Thiết bị	Thời lượng (s)	Độ trễ (s)	RTF
Samsung M10	5	13.73	1.75
OPPO A16	5	12.80	1.56
Vivo X100s Pro	5	6.10	0.22

Bảng 4.6 Đánh giá Latency và RTF ở chế độ offline ASR trên thiết bị smartphone

- **Online ASR:**

Thiết bị	Thời lượng (s)	Độ trễ (s)	RTF
Samsung M10	5	12.50	2.90
OPPO A16	5	14.00	3.00
Vivo X100s Pro	5	5.10	0.96

Bảng 4.7 Đánh giá Latency và RTF ở chế độ online ASR trên thiết bị smartphone

❖ **Kết luận:**

- Xây dựng thành công ứng dụng soạn thảo văn bản tiếng Việt tự động trên smartphone
- Hệ thống hoạt động cả chế độ offline và online ASR
- Hệ thống nhận dạng tiếng nói tiếng Việt đạt **WER < 10 %** trên CStt, VIVOS + Speaker, FPT

❖ **Hướng phát triển:**

- Mở rộng tập dữ liệu huấn luyện
- Bổ sung dữ liệu nhiễu, kết hợp lọc nhiễu đầu vào
- Áp dụng tăng cường dữ liệu: thay đổi tốc độ, thêm nhiễu,…

TÀI LIỆU THAM KHẢO

[1] S. Kim, A. Gholami, A. Shaw, et al., "Squeezeformer: An Efficient Transformer for Automatic Speech Recognition," arXiv:2206.00888, 2022.

[2] T. Cejrowski, J. Szymanski, "Detection of anomalies in bee colony using transitioning state and contrastive autoencoders," *Computers and Electronics in Agriculture*, vol. 200, 2022, doi: 10.1016/j.compag.2022.107207.

[3] J.-W. Zwart, "Local agreement," in *Linguistic Agreement*, pp. 317–339, 2006, doi: 10.1075/la.92.14zwa.

HUST

hust.edu.vn fb.com/dhbkhn

THANK YOU !