

TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI  
VIỆN ĐIỆN TỬ - VIỄN THÔNG



BÁO CÁO MÔN  
KỸ THUẬT LẬP TRÌNH C/C++

Đề tài:

CHƯƠNG TRÌNH  
NHÂN ĐIỆN GIỌNG NÓI

Nhóm thực hiện :	ĐOÀN QUANG LỮU	20203884
	LÊ QUANG HUY	20200273
	NGUYỄN BÁ TRUNG HIẾU	20200224
	ĐẶNG TRƯỜNG GIANG	20200183
Lớp:	Hệ thống Nhúng thông minh và IoT - K65	
Giảng viên hướng dẫn:	NGUYỄN QUANG MINH	

Hà Nội, February 13, 2023

# MỤC LỤC

<b>DANH MỤC HÌNH VẼ</b>	<b>ii</b>
<b>TÓM TẮT BÁO CÁO</b>	<b>iii</b>
<b>CHƯƠNG 1. CHƯƠNG MỞ ĐẦU</b>	<b>1</b>
<b>CHƯƠNG 2. PHÂN TÍCH ĐỀ TÀI</b>	<b>2</b>
2.1 Một số khái niệm . . . . .	2
2.1.1 Speech to text là gì ? . . . . .	2
2.1.2 Ngôn ngữ lập trình C++ . . . . .	2
2.2 Sự quan trọng của đề . . . . .	3
2.2.1 Trong môi trường doanh nghiệp . . . . .	3
2.2.2 Trong điều khiển thiết bị thông minh và giao tiếp với robot . . . . .	4
2.2.3 Trong giáo dục . . . . .	4
<b>CHƯƠNG 3. CHI TIẾT</b>	<b>5</b>
3.1 Chương trình . . . . .	5
3.1.1 Thư viện sử dụng . . . . .	5
3.1.2 Các model trong chương trình . . . . .	6
3.2 Cụ thể chương trình . . . . .	7
3.3 Để sử dụng chương trình . . . . .	10
3.4 Kết quả . . . . .	11
3.4.1 Nhân dạng trực tiếp bằng tiếng Anh . . . . .	11
3.4.2 Nhân dạng thông qua file âm thanh bằng tiếng Anh . . . . .	11
3.4.3 Nhân dạng trực tiếp bằng tiếng Việt . . . . .	11
3.4.4 Nhân dạng thông qua file âm thanh bằng tiếng Việt . . . . .	12
<b>4 KẾT LUẬN</b>	<b>13</b>

**DANH MỤC HÌNH VẼ**

Hình 3.1	Các tham số . . . . .	7
Hình 3.2	Nhận dạng giọng nói trực tiếp bằng tiếng Anh . . . . .	11
Hình 3.3	Nhận dạng giọng nói qua file bằng tiếng Anh . . . . .	11
Hình 3.4	Nhận dạng giọng nói trực tiếp bằng tiếng Việt . . . . .	11
Hình 3.5	Nhận dạng giọng nói qua file bằng tiếng Việt . . . . .	12

# TÓM TẮT BÁO CÁO

Chuyển giọng nói thành văn bản được xem là ứng dụng phổ biến nhất của phần mềm nhận dạng giọng nói hiện nay. Chúng ta có thể dễ dàng nhìn thấy những phần mềm nhận dạng giọng nói ngay trên smartphone hàng ngày đang sử dụng. Bên cạnh đó, nó còn được sử dụng rất nhiều trong lĩnh vực trí tuệ nhân tạo (Artificial Intelligence), Google Assistant là một ví dụ điển hình. Với chủ đề nhận diện giọng nói(Speech to text), nhóm em đã nghiên cứu về chương trình lập trình nhận dạng giọng nói và chuyển thành văn bản, với ngôn ngữ C++. Bài báo cáo gồm mở đầu, chi tiết báo cáo và kết luận của báo cáo.

# CHƯƠNG 1. CHƯƠNG MỞ ĐẦU

Nhận dạng giọng nói là một công nghệ cho phép nhập liệu vào hệ thống. Bạn nói chuyện với máy tính, điện thoại hoặc thiết bị của bạn và nó sử dụng những gì bạn nói là đầu vào để kích hoạt một số hành động. Công nghệ này đang được sử dụng để thay thế các phương thức nhập khác như nhập, nhấp hoặc chọn theo các cách khác. Nó là một phương tiện để làm cho các thiết bị và phần mềm thân thiện với người dùng hơn và tăng năng suất.

Có rất nhiều ứng dụng và khu vực nơi nhận dạng giọng nói được sử dụng, bao gồm quân đội, như một trợ giúp cho những người bị suy yếu (tưởng tượng một người bị tàn tật hoặc không có bàn tay hoặc ngón tay), trong lĩnh vực y tế, trong robot vv Trong tương lai gần, gần như tất cả mọi người sẽ được tiếp xúc với nhận dạng giọng nói do tuyên truyền của nó trong số các thiết bị phổ biến như máy tính và điện thoại di động.

Vì vậy, nhóm em đã nghiên cứu và tìm hiểu về chương trình nghiên cứu giọng nói bằng ngôn ngữ C++

Nội dung của báo cáo gồm:

- Phần mở đầu giới thiệu đề tài.
- Chi tiết phần lập trình
- Kết luận.

## CHƯƠNG 2. PHÂN TÍCH ĐỀ TÀI

### 2.1 Một số khái niệm

#### 2.1.1 *Speech to text là gì ?*

[1] Speech to Text (Speech2Text) là một công nghệ giúp máy tính nhận dạng âm thanh của tiếng nói người và tạo ra chuỗi văn bản tương ứng. Khởi đầu, tiếng nói sẽ được ghi nhận qua microphone và lưu trữ trong máy tính dưới dạng các tín hiệu số. Để máy tính có thể nhận dạng dữ liệu tiếng nói, rất nhiều kỹ thuật xử lý tín hiệu số và xử lý ngôn ngữ tự nhiên được sử dụng. Gần đây, với sự tiến bộ của kỹ thuật học sâu (một nhánh đang phát triển rất mạnh của trí tuệ nhân tạo - AI), việc nhận dạng dạng tiếng nói nói riêng có thể đạt được độ chính xác rất cao trên rất nhiều ngôn ngữ.

Nhận dạng giọng nói là công nghệ quan trọng nhất để tạo nên các ứng dụng tương tác thông minh qua giọng nói, như tìm kiếm qua giọng nói trên smartphone, smartTV, các ứng dụng trợ lý ảo như Siri trên iOS, Cortana trên Windows, hoặc các trợ lý nhà thông minh như Echo, Alexa của Amazon, Google Nest và Google Home của Google... Công nghệ nhận dạng giọng nói mở ra kỷ nguyên mới cho việc tương tác người máy. Để yêu cầu thiết bị thông minh thực hiện chỉ thị của mình, giờ đây bạn chỉ cần ra lệnh bằng giọng nói, thay vì phải thao tác qua nút ấn hoặc màn hình. Vậy rõ ràng cuộc sống cá nhân của con người được giúp ích rất nhiều với công nghệ nhận dạng giọng nói. Còn đối với các công ty hoặc tổ chức, liệu rằng công nghệ nhận dạng giọng nói hay Speech to Text có thể mang đến lợi ích gì? Bài viết này sẽ mang đến cho bạn cái nhìn gần hơn về lợi ích của Speech To Text cho doanh nghiệp của bạn.

#### 2.1.2 *Ngôn ngữ lập trình C++*

[2] C++ (C Plus Plus, CPP, IPA: /si pls pls/) là một loại ngôn ngữ lập trình bậc trung (middle-level). Đây là ngôn ngữ lập trình đa năng được tạo ra bởi Bjarne Stroustrup như một phần mở rộng của ngôn ngữ lập trình C, hoặc "C với các lớp Class", Ngôn ngữ đã được mở rộng đáng kể theo thời gian và C ++ hiện đại có các tính năng: lập trình tổng quát, lập trình hướng đối tượng, lập trình thủ tục, ngôn ngữ đa mẫu hình tự do có kiểu tĩnh, dữ liệu trừu tượng, và lập trình đa hình, ngoài ra còn có thêm các tính năng, công cụ để thao tác với bộ nhớ cấp thấp. Từ thập niên 1990, C++ đã trở thành một trong những ngôn ngữ thương mại ưa thích và phổ biến của lập trình viên.

C++ được thiết kế hướng tới lập trình hệ thống máy tính và phần mềm nhúng trên các mạch vi xử lý, bao gồm cả hệ thống có tài nguyên hạn chế và tài nguyên khổng lồ, với ưu điểm vượt trội về hiệu suất, hiệu quả và tính linh hoạt cao.[6] C ++ có thể tìm thấy ở mọi nơi, với những điểm mạnh là cơ sở hạ tầng phần mềm và các ứng dụng bị hạn chế tài nguyên. bao gồm: phần mềm ứng dụng máy tính cá nhân, trò chơi điện tử, các hệ thống máy chủ (ví dụ:

phần mềm thương mại điện tử, cỗ máy tìm kiếm trên web hoặc máy chủ SQL) và các ứng dụng ưu tiên về hiệu suất (ví dụ: tổng đài thông tin liên lạc hoặc thiết bị thăm dò không gian).[7] C++ hầu hết được thực thi dưới dạng là một ngôn ngữ biên dịch, có thể chạy trên nhiều nền tảng khác nhau như Windows, Mac OS, Linux, các phiên bản Unix. Nhiều nhà cung cấp cung cấp các trình biên dịch C++, bao gồm Tổ chức Phần mềm Tự do, Microsoft, Intel và IBM.

C++ được Tổ chức Tiêu chuẩn hóa Quốc tế (ISO) chuẩn hóa, với phiên bản tiêu chuẩn mới nhất được ISO phê chuẩn và công bố vào tháng 12 năm 2017 là ISO / IEC 14882: 2017 (được gọi một cách không chính thức là C++ 17).[8] Ngôn ngữ lập trình C++ ban đầu được chuẩn hóa vào năm 1998 là ISO / IEC 14882: 1998, sau đó được sửa đổi theo tiêu chuẩn C++ 03, C++ 11 và C++ 14. Tiêu chuẩn C++ 17 hiện tại thay thế các tính năng mới này và một thư viện tiêu chuẩn mở rộng. Trước khi tiêu chuẩn hóa ban đầu vào năm 1998, C++ được phát triển bởi Bjarne Stroustrup tại Bell Labs từ năm 1979, như một phần mở rộng của ngôn ngữ C khi ông muốn một ngôn ngữ hiệu quả và linh hoạt tương tự như C, cũng cung cấp các tính năng cấp cao cho tổ chức chương trình.[9] C++ 20 là tiêu chuẩn được lên kế hoạch tiếp theo sau đó, phù hợp với chuỗi hiện tại của một phiên bản mới cứ sau ba năm.[10]

## **2.2 Sự quan trọng của đề**

### **2.2.1 Trong môi trường doanh nghiệp**

*Speech to Text có thể đơn giản hóa các công việc thường ngày ở công ty bạn, giảm sai sót, tiết kiệm nhân công và giúp tăng tính hiệu quả của công việc ở công ty bạn. Một số lợi ích của Speech To Text có thể kể đến như:*

- Giảm thời gian tương tác: Trong thực tế, thời gian để nói ra một câu luôn ngắn hơn thời gian để nhập câu đó vào máy tính. Cũng như vậy, thời gian để đọc được một tin nhắn luôn ngắn hơn thời gian để nghe tin nhắn thoại cùng một nội dung. Vậy, nếu ta có thể triển khai được một hệ thống nhắn tin nội bộ nhận tin nhắn thoại ở đầu vào và cho ra tin nhắn văn bản ở đầu ra, thời gian tương tác giữa các nhân viên nội bộ có thể được rút ngắn đáng kể.
- Giảm thời gian nhập liệu: Nhập liệu là một công việc nhàm chán và dễ sai sót. Với tính năng Speech to Text, nhân viên nhập liệu có thể nhập dữ liệu trực tiếp bằng giọng nói của mình thay vì đọc dữ liệu từ tài liệu nguồn và gõ lại trên bàn phím. Nhờ vậy cả lỗi đánh máy và thời gian nhập liệu đều được tối thiểu hóa.
- Dễ dàng tìm kiếm tài liệu: Với chức năng Speech To Text, bạn có thể tìm kiếm tài liệu trực tiếp bằng giọng nói, thay vì gõ từ khóa vào thanh công cụ tìm kiếm trên màn hình.
- Tự động tạo biên bản cuộc họp: Mỗi ngày có rất nhiều cuộc họp quan trọng cần được ghi biên bản. Speech to Text có thể dễ dàng tự động chuyển đổi nội dung ghi âm của cuộc họp thành văn bản, nhờ đó giảm được áp lực và sai sót của thư ký ghi biên bản.

- **Trả lời tự động:** Với các trung tâm chăm sóc khách hàng, có rất nhiều câu hỏi về dịch vụ được lặp đi lặp lại. Với chức năng nhận dạng giọng nói và Text To Speech, một hệ thống trả lời tự động có thể hiểu được câu hỏi của khách hàng, truy vấn cơ sở dữ liệu và tự tạo ra câu trả lời dưới dạng âm thanh để phản hồi cho khách hàng. Hệ thống này có thể tiết kiệm cho doanh nghiệp kinh phí rất lớn khi có thể phục vụ khách hàng 24/7 mà không cần phải tốn chi phí cho nhân viên trực điện thoại.
- **Nhận và chuyển tiếp tin nhắn của khách hàng:** Khách hàng vẫn có thể gọi điện đến công ty của bạn sau giờ làm việc và để lại lời nhắn. Khi sử dụng Speech To Text, bạn sẽ không phải bật nghe lại các đoạn tin nhắn này mà chỉ cần đọc chúng dưới dạng văn bản, vốn tốn ít thời gian hơn. Hơn thế nữa, hệ thống có thể hiểu được đơn vị nào của công ty cần phải xử lý tin nhắn này và chuyển tiếp ngay sau khi nhận được.
- **Hiểu thông tin khách hàng để marketing hướng mục tiêu:** Các công nghệ nhận dạng giọng nói hiện đại có thể dễ dàng đoán được độ tuổi, giới tính và vùng miền của con người thông qua giọng nói. Điều này rất có lợi cho công ty bạn khi cần biết một số thông tin cá nhân của khách hàng để thực hiện chiến lược marketing đúng mục tiêu.

### ***2.2.2 Trong điều khiển thiết bị thông minh và giao tiếp với robot***

- Công nghệ nhận dạng giọng nói giúp khoảng cách giao tiếp giữa người và máy được rút ngắn. Việc sử dụng, ra lệnh, truy vấn thông tin ngày nay dễ dàng hơn rất nhiều cho người già, trẻ em hoặc người khuyết tật.
- Các thiết bị IoT nhà thông minh luôn cần một thiết bị trợ lý ảo để điều khiển chúng. Công nghệ nhận dạng giọng nói hiện nay giúp chủ nhân có thể điều khiển ngôi nhà thông minh của mình mà không cần dùng một nút ấn nào.
- Giao tiếp với robots hiện nay không còn là ý tưởng trong mơ nữa. Rất nhiều robots gần đây có khả năng giao tiếp như người thật. Thậm chí vào năm 2017, Sophia, một robot có khả năng giao tiếp và biểu hiện sắc thái được phát triển từ Hong Kong vào năm 2016, còn được Arab Saudi cấp quyền công dân. Tất cả khả năng giao tiếp của robot đều là thành tựu của trí tuệ nhân tạo và xử lý ngôn ngữ tự nhiên.

### ***2.2.3 Trong giáo dục***

Rất nhiều ứng dụng dạy ngôn ngữ hiện nay sử dụng trí tuệ nhân tạo và nhận dạng tiếng nói như công nghệ chính đánh giá khả năng ngôn ngữ của người học. Những kỹ thuật này có thể thay thế giáo viên trong việc đánh giá và hoàn thiện kỹ năng phát âm của học viên.



## CHƯƠNG 3. CHI TIẾT

[3]

### 3.1 Chương trình

#### 3.1.1 Thư viện sử dụng

Chương trình sử dụng thư viện whisper là thư viện nhân diện giọng nói và ggml là thư viện dùng cho học máy dùng để load được model sử dụng, là hai thư viện chính để thực hiện chức năng, ngoài ra chương trình còn sử dụng một số thư viện như thư viện thuật toán, thư viện random, thư viện biểu thức chính quy ...

#### - Thư viện whisper.cpp

- Suy luận hiệu suất cao của mô hình nhận dạng giọng nói tự động Whisper (ASR) của OpenAI:
- Triển khai C/C++ đơn giản mà không cần phụ thuộc
- Công dân hạng nhất silicon của Apple - được tối ưu hóa thông qua khung Arm Neon và Tăng tốc
- Hỗ trợ nội tại AVX cho kiến trúc x86
- Hỗ trợ nội tại VSX cho kiến trúc POWER
- Độ chính xác hỗn hợp F16 / F32
- Sử dụng bộ nhớ thấp (Chú ý Flash)
- Cấp phát bộ nhớ bằng không khi chạy
- Chạy trên CPU
- API kiểu C
- Hệ điều hành Mac (Intel và Arm) iOS , Android , Linux / FreeBSD
- Toàn bộ việc triển khai mô hình được chứa trong 2 tệp nguồn:
- Hoạt động của thang đo: ggml.h / ggml.c

### ***3.1.2 Các model trong chương trình***

Các model có sẵn của chương trình: ggml-large.bin, ggml-small.bin, ... Tuy nhiên model large thì nhanh nhưng nặng và độ chính xác cũng không hơn base nhiều, model small thì nhẹ, nhanh tuy nhiên độ chính xác kém, vì vậy qua khảo sát, nhóm chọn 2 model với 2 mục đích tương ứng như sau :

- Model ggml-base.bin sử dụng cho đa ngôn ngữ
- Model ggml-base.en.bin sử dụng cho tiếng Anh

## 3.2 Cụ thể chương trình

- ctx : nơi text được nhận diện wparams : tham số
- pcmf32 : dữ liệu âm thanh
- Các tham số :

```
options:
-h,      --help                [default] show this help message and exit
-t N,    --threads N          [4      ] number of threads to use during computation
-p N,    --processors N       [1      ] number of processors to use during computation
-ot N,   --offset-t N         [0      ] time offset in milliseconds
-on N,   --offset-n N         [0      ] segment index offset
-d N,    --duration N         [0      ] duration of audio to process in milliseconds
-mc N,   --max-context N      [-1     ] maximum number of text context tokens to store
-ml N,   --max-len N          [0      ] maximum segment length in characters
-bo N,   --best-of N          [5      ] number of best candidates to keep
-bs N,   --beam-size N        [-1     ] beam size for beam search
-wt N,   --word-thold N       [0.01   ] word timestamp probability threshold
-et N,   --entropy-thold N    [2.40   ] entropy threshold for decoder fail
-lpt N,  --logprob-thold N    [-1.00  ] log probability threshold for decoder fail
-su,     --speed-up           [false  ] speed up audio by x2 (reduced accuracy)
-tr,     --translate          [false  ] translate from source language to english
-di,     --diarize            [false  ] stereo audio diarization
-nf,     --no-fallback        [false  ] do not use temperature fallback while decoding
-otxt,   --output-txt         [false  ] output result in a text file
-ovtt,   --output-vtt         [false  ] output result in a vtt file
-osrt,   --output-srt         [false  ] output result in a srt file
-owts,   --output-words       [false  ] output script for generating karaoke video
-ocsv,   --output-csv         [false  ] output result in a CSV file
-of FNAME, --output-file FNAME [       ] output file path (without file extension)
-ps,     --print-special      [false  ] print special tokens
-pc,     --print-colors       [false  ] print colors
-pp,     --print-progress     [false  ] print progress
-nt,     --no-timestamps      [true   ] do not print timestamps
-l LANG,  --language LANG     [en     ] spoken language ('auto' for auto-detect)
          --prompt PROMPT     [       ] initial prompt
-m FNAME, --model FNAME       [models/ggml-base.en.bin] model path
-f FNAME, --file FNAME        [       ] input WAV file path
```

Hình 3.1 Các tham số

- Lệnh (`whisper_full_parallel(ctx, wparams, pcmf32.data(), pcmf32.size(), params.n_processors) != 0`) sử dụng để nhận diện âm thanh, chạy liên tục.

```

}

if (whisper_full_parallel(ctx, wparams, pcmf32.data(), pcmf32.size(), params.n_processors) != 0) {
    fprintf(stderr, "%s: failed to process audio\n", argv[0]);
    return 10;
}
}

if (whisper_full(ctx, wparams, pcmf32.data(), pcmf32.size()) != 0) {
    fprintf(stderr, "%s: failed to process audio\n", argv[0]);
    return 6;
}
}

```

- Code xử lý âm thanh tiếp theo sau một thời gian nhất định( thời gian có thể sửa ở tham số) :

```

if (!use_vad) {
    while (true) {
        audio.get(params.step_ms, pcmf32_new);

        if ((int) pcmf32_new.size() > 2*n_samples_step) {
            fprintf(stderr, "\n\n%s: WARNING: cannot process audio fast enough, dropping audio ...\n\n", __func__);
            audio.clear();
            continue;
        }

        if ((int) pcmf32_new.size() >= n_samples_step) {
            audio.clear();
            break;
        }

        SDL_Delay(1);
    }

    const int n_samples_new = pcmf32_new.size();

    // take up to params.length_ms audio from previous iteration
    const int n_samples_take = std::min((int) pcmf32_old.size(), std::max(0, n_samples_keep + n_samples_len - n_samples_n

    //printf("processing: take = %d, new = %d, old = %d\n", n_samples_take, n_samples_new, (int) pcmf32_old.size());

    pcmf32.resize(n_samples_new + n_samples_take);

    for (int i = 0; i < n_samples_take; i++) {
        pcmf32[i] = pcmf32_old[pcmf32_old.size() - n_samples_take + i];
    }

    memcpy(pcmf32.data() + n_samples_take, pcmf32_new.data(), n_samples_new*sizeof(float));

    pcmf32_old = pcmf32;
} else {
    const auto t_now = std::chrono::high_resolution_clock::now();
    const auto t_diff = std::chrono::duration_cast<std::chrono::milliseconds>(t_now - t_last).count();

    if (t_diff < 2000) {
        std::this_thread::sleep_for(std::chrono::milliseconds(100));

        continue;
    }

    audio.get(2000, pcmf32_new);

    if (vad_simple(pcmf32_new, WHISPER_SAMPLE_RATE, 1000, params.vad_thold, params.freq_thold, false)) {
        audio.get(params.length_ms, pcmf32);
    } else {
        std::this_thread::sleep_for(std::chrono::milliseconds(100));

        continue;
    }

    t_last = t_now;
}

// run the inference

```

- Đoạn code tạo 1 file text để xuất kết quả ( trường hợp nhận diện giọng nói qua file âm thanh bằng tiếng Việt)

```
// output to text file
if (params.output_txt) {
    const auto fname_txt = fname_outp + ".txt";
    output_txt(ctx, fname_txt.c_str());
}
```

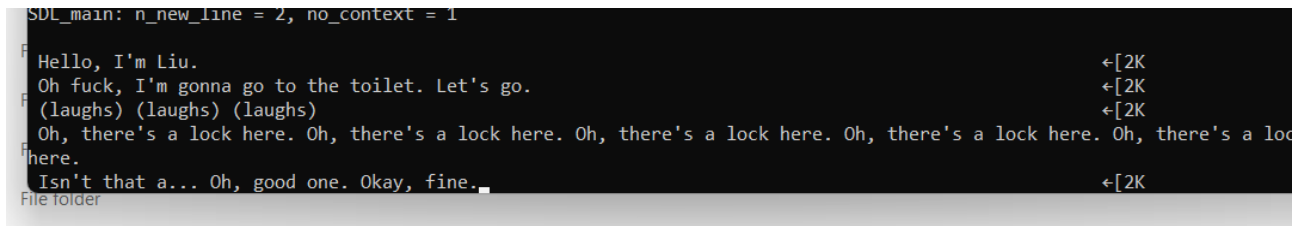
### 3.3 Để sử dụng chương trình

Ban đầu cmd của folder trên Windows, sau đó lần lượt gõ các dòng sau với các chức năng tương ứng :

- main -m ./models/ggml-small.bin -l vi -f ./samples/output.wav -otxt ( nhận diện từ file tiếng việt thêm -tr để dịch sang tiếng anh) -> ổn
- stream -m ./models/ggml-base.bin -l vi -t 8 ( trực tiếp tiếng việt thêm -tr để dịch sang tiếng anh) -> lỗi font , còn nhận diện ổn , dịch hơi sai
- stream ( nhận diện tiếng anh ) -> ổn, trực tiếp tiếng anh
- main -m ./models/ggml-base.en.bin -f ./samples/jfk.wav -otxt ( từ file wav tiếng anh thêm otxt để ghi vào text) ( thêm -ml 16 chia giây) -> ổn nhất

### 3.4 Kết quả

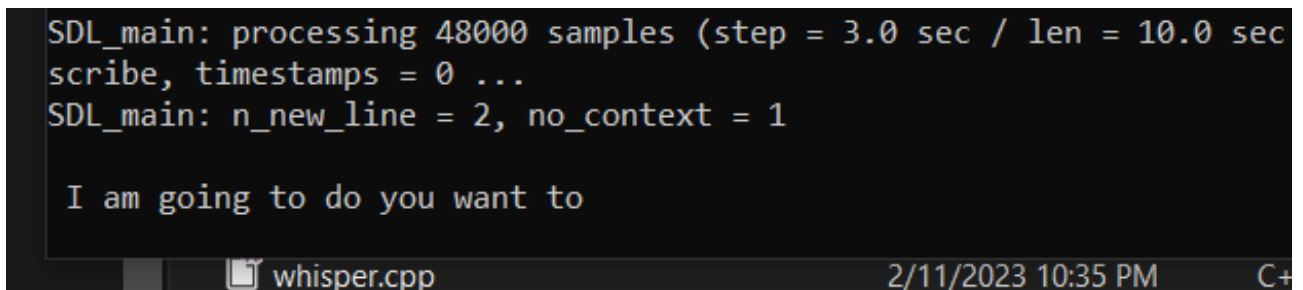
#### 3.4.1 Nhận dạng trực tiếp bằng tiếng Anh



Hình 3.2 Nhận dạng giọng nói trực tiếp bằng tiếng Anh

#### 3.4.2 Nhận dạng thông qua file âm thanh bằng tiếng Anh

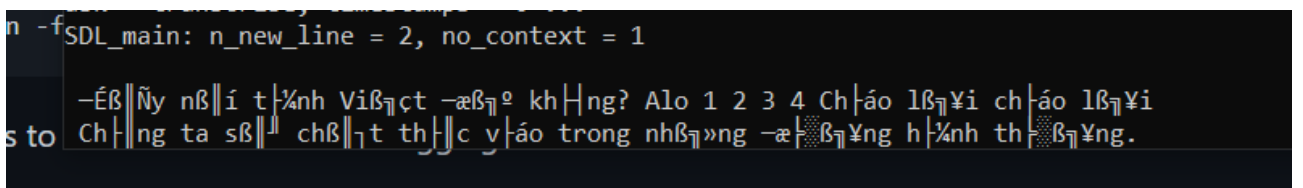
Kết quả ổn



Hình 3.3 Nhận dạng giọng nói qua file bằng tiếng Anh

#### 3.4.3 Nhận dạng trực tiếp bằng tiếng Việt

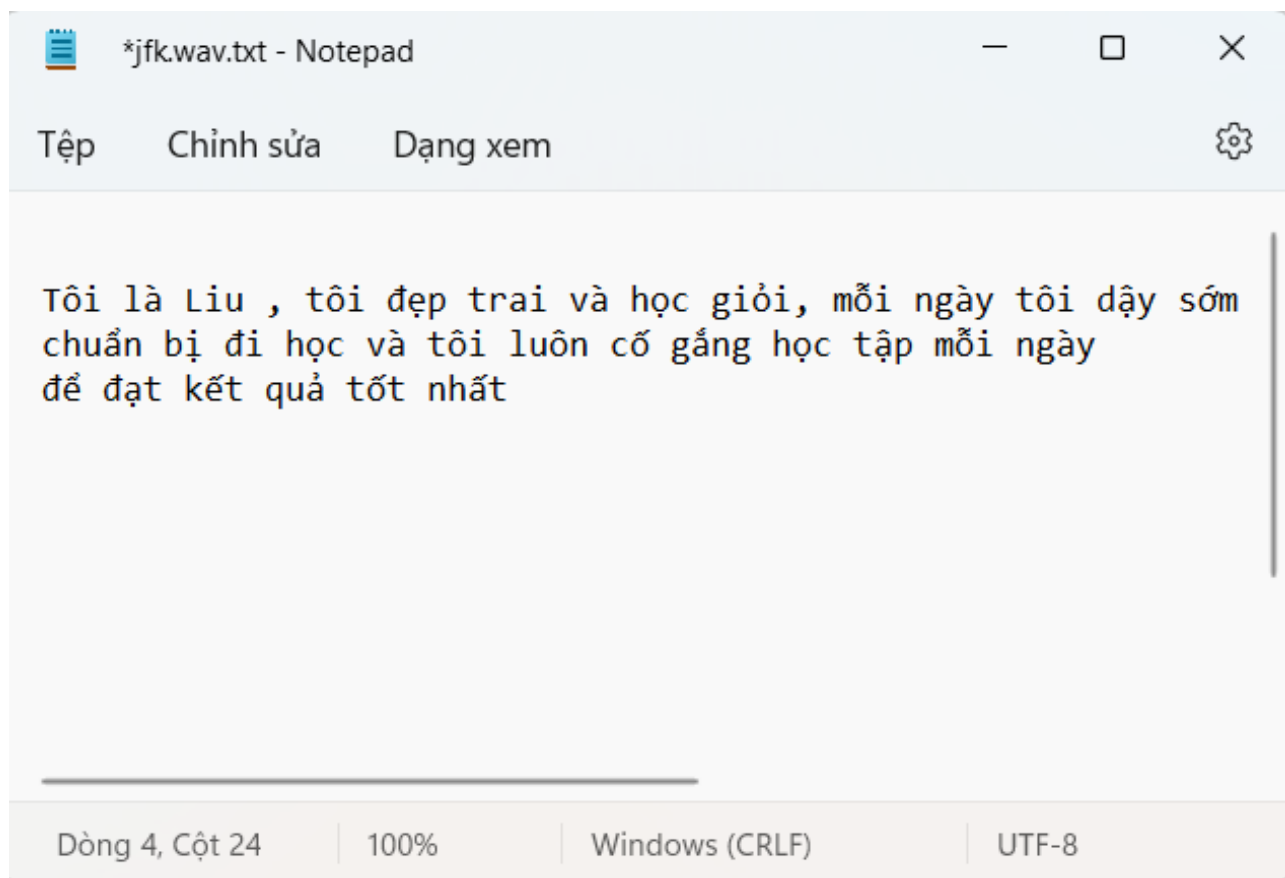
Lỗi font, tuy nhiên nhận diện ổn.



Hình 3.4 Nhận dạng giọng nói trực tiếp bằng tiếng Việt

#### 3.4.4 Nhận dạng thông qua file âm thanh bằng tiếng Việt

Sau khi nhận diện, tạo 1 file text để tránh lỗi font => Kết quả ổn



**Hình 3.5 Nhận dạng giọng nói qua file bằng tiếng Việt**



## 4 KẾT LUẬN

Nhóm chúng em đã hoàn thành bài tập lớn theo đề tài: Chương trình nhận dạng giọng nói của môn học này với một sản phẩm khá cần thiết với thực tế. Qua bài tập lớn lần này, chúng em đã cải thiện thêm cả năng lực làm việc nhóm của mình, đồng thời thành thạo hơn trong việc lập trình cũng như hiểu hơn về ngôn ngữ C++ sử dụng cho sản phẩm này và các sản phẩm về sau.

Chúng em xin chân thành cảm ơn thầy Nguyễn Quang Minh đã tận tình giảng dạy và củng cố các kiến thức của môn học để chúng em có thể hoàn thành bài tập lớn này. Trong quá trình làm Bài tập lớn chắc chắn chúng em sẽ gặp phải các sai sót, kính mong thầy có thể góp ý, chỉnh sửa để nhóm hoàn thiện bài tập và có thêm kinh nghiệm trong các sản phẩm về sau.

## TÀI LIỆU THAM KHẢO

- [1] vione, “Phần mềm nhận diện giọng nói hoạt động như thế nào?,” [Online]. Available: <https://vione.ai/phan-mem-nhan-dien-giong-noi-hoat-dong-nhu-the-nao/>.
- [2] Wikipedia, “C++,” [Online]. Available: <https://vi.wikipedia.org/wiki/C>
- [3] ggerganov, “Speech to text,” [Online]. Available : <https://github.com/ggerganov/whisper.cppfbclid=IwAR0Jxc8JwAWgjydtEZKBJ6letAFymVtxc1fVhd7fw8-JNnrjA>, 2022.