ĐẠI NAM
UNIVERSITY

THỰC TẬP CNTT 5: TRIỂN KHAI ỨNG DỤNG AI, IoT

GIỚI THIỆU VỀ HỌC PHẦN

ĐỀ TÀI: NHẬN DIỆN GIỌNG NÓI SANG VĂN BẢN

Giảng viên hướng dẫn: ThS. Lê Trung Hiếu

ThS. Nguyễn Thái Khánh

Nhóm 1: Đỗ Huy Dũng

Nguyễn Công Uẩn

Lê Xuân Dương

Bùi Anh Tuấn

MỤC LỤC

GIỚI THIỆU ĐỀ TÀI

Tên đề tài:

“Hệ thống Nhận Diện Giọng Nói trong Cuộc Họp và Ghi Chép Tự Động”

Lý do chọn đề tài:

Nhu cầu: Trong các cuộc họp, việc ghi chép thủ công tốn thời gian, dễ bỏ sót ý quan trọng.

Xu hướng: AI, NLP (Natural Language Processing) và Speech-to-Text đang ngày càng phổ biến.

Tiện ích: Tự động nhận diện người nói, chuyển âm thanh thành văn bản, lưu lại thành transcript giúp nâng cao hiệu suất, tiết kiệm nhân lực.

Mục tiêu tổng quan:

Tạo một giao diện web hỗ trợ người dùng quản lý cuộc họp.

Nhận diện giọng nói (STT) theo thời gian thực.

Nhận diện người nói (speaker recognition) dựa trên dữ liệu huấn luyện.

Lưu toàn bộ hội thoại thành file transcript.txt.

MỤC TIÊU NGHIÊN CỨU

Mục tiêu chính:
Xây dựng hệ thống huấn luyện và nhận diện người nói.
Tích hợp nhận diện giọng nói theo thời gian thực trên giao diện web.

Các mục tiêu phụ:
Tạo ra mô hình GMM cho từng người nói.
Đánh giá độ chính xác của mô hình trên tập test.
Lưu trữ và xuất lịch sử hội thoại.

PHẠM VI NGHIÊN CỨU

Dữ liệu sử dụng:

Tập huấn luyện: Các file âm thanh (.wav) được thu thập từ 2 người nói (Anh Uẩn và Anh Tuấn).

Tập kiểm thử: Các file âm thanh thuộc từng danh mục của người nói tương ứng.

Các bước tiền xử lý:

- Đọc dữ liệu âm thanh với tốc độ mẫu chuẩn (16 kHz).
- Trích xuất đặc trưng MFCC (13 hệ số MFCC) từ từng file âm thanh.

Giới hạn của nghiên cứu:

- Chỉ tập trung vào 2 người nói với dữ liệu đã được thu thập sẵn.
- Các yếu tố nhiễu và chất lượng âm thanh được kiểm soát trong môi trường phòng thí nghiệm.

Các vấn đề cần lưu ý:

- Xác định ngưỡng RMS để loại bỏ các đoạn âm thanh không có giọng nói.
- Điều chỉnh ngưỡng score khi nhận diện để tránh nhận diện sai.

QUY TRÌNH THỰC HIỆN

Bước 1: Thu thập và Tiền xử lý Dữ liệu
Tập hợp các file âm thanh cho từng người nói từ thư mục audio/train.
Tiền xử lý âm thanh: chuẩn hóa mẫu, lọc nhiễu.
Bước 2: Trích xuất đặc trưng MFCC
Sử dụng thư viện Librosa để chuyển đổi tín hiệu thành ma trận MFCC (đầu ra: số khung hình và 13 hệ số MFCC).
Bước 3: Huấn luyện mô hình GMM
Kết hợp tất cả các khung MFCC cho mỗi người nói.
Huấn luyện mô hình GMM với 16 thành phần, sử dụng covariance dạng ‘diag’ và max_iter=200.
Bước 4: Đánh giá mô hình
So sánh kết quả nhận diện trên tập test (đường dẫn audio/test).
Tính độ chính xác dựa trên số file nhận diện đúng trên tổng số file.
Bước 5: Tích hợp vào ứng dụng web
Phát triển server bằng Flask và Socket.IO để xử lý dữ liệu âm thanh theo thời gian thực.
Xử lý file âm thanh tạm thời, nhận diện giọng nói và gửi kết quả về client.

CÁC CHỨC NĂNG CHÍNH

Nhận diện người nói:

- Trích xuất đặc trưng MFCC từ file âm thanh.
- Tính toán RMS để xác định có giọng nói hay không.
- Sử dụng GMM để tính điểm số cho từng người nói và đưa ra nhận diện cuối cùng.

Giao diện Web thời gian thực:

- Hiển thị trạng thái cuộc họp: “Chưa bắt đầu”, “Đang diễn ra”, “Kết thúc”.
- Cập nhật người nói hiện tại và hiển thị văn bản nhận diện theo thời gian thực.
- Ghi lại lịch sử hội thoại và cho phép tải về file transcript.

Giao tiếp hai chiều qua Socket.IO:

- Gửi dữ liệu âm thanh từ client đến server.
- Nhận kết quả nhận diện và cập nhật giao diện ngay lập tức.

PHƯƠNG PHÁP THỰC HIỆN

Trích xuất MFCC:

- Sử dụng librosa.load() để đọc file âm thanh với tốc độ mẫu 16 kHz.
- Sử dụng librosa.feature.mfcc() để trích xuất 13 hệ số MFCC, chuyển đổi kết quả sang dạng ma trận với kích thước (số frame, 13).

Huấn luyện mô hình GMM:

- Sử dụng GaussianMixture từ scikit-learn.
- Tích hợp tất cả các khung MFCC của người nói, huấn luyện với 16 thành phần để mô hình hóa phân phối của dữ liệu.
- Sử dụng phương pháp Expectation-Maximization để tối ưu hóa mô hình.

Nhận diện người nói:

- Tính toán RMS từ file âm thanh để loại trừ các đoạn không có giọng nói.
- So sánh điểm số (score) của MFCC từ file âm thanh với các mô hình GMM đã huấn luyện.
- Thiết lập ngưỡng score để quyết định nhận diện hợp lệ hay không.

Tích hợp với ứng dụng web:

- Phát triển server với Flask để xử lý các yêu cầu HTTP.
- Sử dụng Socket.IO cho giao tiếp thời gian thực giữa client và server.
- Quá trình lưu file tạm, xử lý, gửi kết quả và xóa file sau khi xử lý.

QUY TRÌNH HOẠT ĐỘNG

Quy trình nhận diện người nói:

Gửi dữ liệu âm thanh:

- Người dùng nhấn nút “Bắt đầu cuộc họp” trên giao diện web.
- Hệ thống bắt đầu ghi âm và gửi dữ liệu qua Socket.IO.

Tiền xử lý & trích xuất MFCC:

- Server lưu dữ liệu âm thanh tạm thời.
- File được xử lý để trích xuất MFCC và tính RMS.

Nhận diện:

- Mô hình GMM tính điểm số cho mỗi người nói.
- Người nói có điểm số cao nhất được nhận diện, nếu vượt ngưỡng xác định.

Trả kết quả & cập nhật giao diện:

- Kết quả nhận diện được gửi về client để hiển thị.
- Lịch sử hội thoại được ghi lại theo thời gian thực.

Lưu trữ & xuất file transcript:

- Cuộc hội thoại được lưu trữ và cho phép tải file transcript dưới dạng .txt khi kết thúc cuộc họp.

GIAO DIỆN

Mô tả trang chính:

Thiết kế giao diện người dùng:

Giao diện chính:

- Hai nút điều khiển: “Bắt đầu cuộc họp” và “Kết thúc cuộc họp”.
- Hiển thị trạng thái cuộc họp (ví dụ: “Đang diễn ra”).
- Hiển thị người nói hiện tại được nhận diện.

Lịch sử hội thoại:

- Một khu vực cập nhật văn bản nhận diện theo thời gian thực.
- Cho phép lưu lại toàn bộ nội dung hội thoại.

Tính năng thời gian thực:

- Sử dụng Socket.IO để đảm bảo dữ liệu âm thanh và kết quả nhận diện được cập nhật tức thì.
- Giao diện được thiết kế đơn giản, dễ hiểu, phù hợp với người dùng cuối.

Trực quan hóa thông tin:

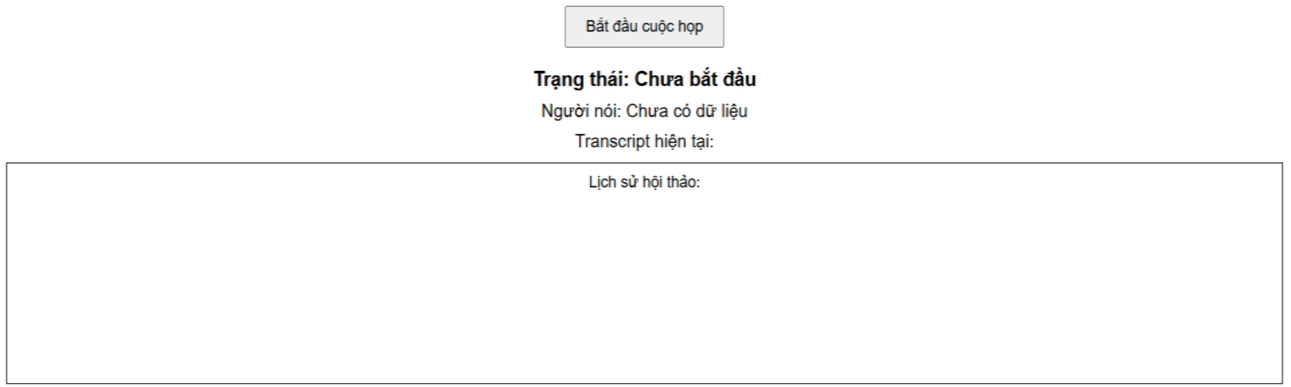
- Có thể bổ sung các biểu đồ hoặc hình ảnh minh họa cho quá trình xử lý tín hiệu và nhận diện.

Tính năng thời gian thực:

- Sử dụng Socket.IO để đảm bảo dữ liệu âm thanh và kết quả nhận diện được cập nhật tức thì.
- Giao diện được thiết kế đơn giản, dễ hiểu, phù hợp với người dùng cuối.

Trực quan hóa thông tin:

- Có thể bổ sung các biểu đồ hoặc hình ảnh minh họa cho quá trình xử lý tín hiệu và nhận diện.



KẾT QUẢ ĐẠT ĐƯỢC

Hiệu quả của mô hình:

Độ chính xác được tính dựa trên tập kiểm thử: ví dụ, đạt trên 85% nhận diện đúng trong môi trường kiểm soát.

Các kết quả score của từng người nói được in ra để kiểm tra, giúp phân tích hiệu năng mô hình.

Thành tích của hệ thống:

Nhận diện người nói với ngưỡng năng lượng RMS phù hợp, loại bỏ các đoạn âm thanh không có giọng.

Tích hợp giao diện web cho phép hiển thị kết quả theo thời gian thực và lưu trữ lịch sử hội thoại.

Hệ thống hoạt động ổn định và có thể mở rộng với dữ liệu nhiều người nói hơn.

Biểu đồ & số liệu:

(Có thể trình bày biểu đồ so sánh số file nhận diện đúng/sai cho từng người nói.)

Bảng số liệu điểm số (score) được hiển thị trên log để kiểm tra hiệu quả.

```
Anh Uẩn: -60.29190225918141
Anh Tuấn: -61.17684765703042
Nhận diện: Anh Uẩn (Score: -60.29190225918141)
```

KHÓ KHĂN VÀ HƯỚNG PHÁT TRIỂN

Khó khăn trong triển khai và nghiên cứu:

- Chất lượng âm thanh:
- Nhiễu môi trường, âm lượng thấp hoặc biến động có thể ảnh hưởng đến quá trình trích xuất MFCC.
- Tối ưu hóa mô hình:
- Mô hình GMM có thể cần cải thiện khi mở rộng sang nhiều người nói và môi trường thực tế phức tạp.

Hướng phát triển trong tương lai:

- Cải tiến mô hình:
- Nghiên cứu và áp dụng các mô hình deep learning như CNN, RNN hoặc transformer cho nhận diện giọng nói.
- Tích hợp thêm các chức năng:
- Nhận diện cảm xúc từ giọng nói, phân tích ngữ cảnh cuộc hội thoại.
- Mở rộng dữ liệu:
- Thu thập thêm dữ liệu từ nhiều người nói và các môi trường âm thanh khác nhau để cải thiện độ chính xác.

ĐẠI NAM
UNIVERSITY

Thank You