TỔNG LIÊN ĐOÀN LAO ĐỘNG VIỆT NAM

**TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG**

**KHOA CÔNG NGHỆ THÔNG TIN**



**ĐỒ ÁN CUỐI KÌ**

**XỬ LÝ DỮ LIỆU LỚN**

**PROJECT 23**

*Người thực hiện*: **NGUYỄN THẾ TRƯỜNG – 51900780**

**MAI HOÀNG VIỆT – 51900847**

Lớp **: 19050402, 19050302**

Khoá  **: 23**

*Người hướng dẫn*: **TS. BÙI THANH HÙNG**

**THÀNH PHỐ HỒ CHÍ MINH, NĂM 2022**

TỔNG LIÊN ĐOÀN LAO ĐỘNG VIỆT NAM

**TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG**

**KHOA CÔNG NGHỆ THÔNG TIN**



**ĐỒ ÁN CUỐI KÌ**

**XỬ LÝ DỮ LIỆU LỚN**

**PROJECT 23**

*Người thực hiện*: **NGUYỄN THẾ TRƯỜNG – 51900780**

**MAI HOÀNG VIỆT – 51900847**

Lớp **: 19050402, 19050302**

Khoá  **: 23**

*Người hướng dẫn*: **TS. BÙI THANH HÙNG**

**THÀNH PHỐ HỒ CHÍ MINH, NĂM 2022**

LỜI CẢM ƠN

Chúng em xin gửi lời cảm ơn đến Thầy Bùi Thanh Hùng – người đã trực tiếp hỗ trợ chúng em trong các tiết học để chúng em có thể hiểu rõ hơn về môn Xử lý dữ liệu lớn.

Trong quá trình thực hiện vẫn có thể còn nhiều sai sót, mong thầy hỗ trợ thêm để hoàn thành tốt nhất có thể.

Một lần nữa, chúng em xin chân thành cảm ơn và gửi đến thầy lời chúc tốt đẹp nhất.

**ĐỒ ÁN ĐƯỢC HOÀN THÀNH**

**TẠI TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG**

Tôi xin cam đoan đây là sản phẩm đồ án của riêng chúng tôi và được sự hướng dẫn của TS. Bùi Thanh Hùng. Các nội dung nghiên cứu, kết quả trong đề tài này là trung thực và chưa công bố dưới bất kỳ hình thức nào trước đây. Những số liệu trong các bảng biểu phục vụ cho việc phân tích, nhận xét, đánh giá được chính tác giả thu thập từ các nguồn khác nhau có ghi rõ trong phần tài liệu tham khảo.

Ngoài ra, trong đồ án còn sử dụng một số nhận xét, đánh giá cũng như số liệu của các tác giả khác, cơ quan tổ chức khác đều có trích dẫn và chú thích nguồn gốc.

**Nếu phát hiện có bất kỳ sự gian lận nào tôi xin hoàn toàn chịu trách nhiệm về nội dung đồ án của mình.** Trường đại học Tôn Đức Thắng không liên quan đến những vi phạm tác quyền, bản quyền do tôi gây ra trong quá trình thực hiện (nếu có).

*TP. Hồ Chí Minh, ngày 22 tháng 05 năm 2022*

*Tác giả*

*(ký tên và ghi rõ họ tên)*

*Nguyễn Thế Trường*

*Mai Hoàng Việt*

PHẦN ĐÁNH GIÁ CỦA GIẢNG VIÊN

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Tp. Hồ Chí Minh, ngày tháng năm

(kí và ghi họ tên)

TÓM TẮT

Trong kỷ nguyên kỹ thuật số, không thể phủ nhận rằng nhận dạng giọng nói có ý nghĩa quan trọng trong các khía cạnh khác nhau của cuộc sống. Cùng với sự bùng nổ của trí tuệ nhân tạo, đã có rất nhiều những công trình nghiên cứu về học sâu được công bố, trong số đó lĩnh vực xử lý ngôn ngữ tự nhiên nói chung cũng như nhận dạng tiếng nói nói riêng là một trong các lĩnh vực được rất nhiều người quan tâm. Trong đó, bài toán nhận dạng tiếng nói tiếng Việt bằng phương pháp học sâu mang lại hiệu quả cao bởi tính ổn định và thích nghi rất tốt khi dữ liệu đầu vào có thay đổi. Giao tiếp bằng lời nói là phương tiện giao tiếp tiêu biểu nhất trong đời sống của con người. Giọng nói mang nhiều thông tin đến người đang nói. Để nhận được giọng nói từ một người, có một số tính năng nhất định tồn tại trong tín hiệu của giọng nói. Bởi vì thông tin có giá trị của nó, có rất nhiều ứng dụng sử dụng nhận dạng giọng nói như: trò chuyện, phát hiện người, bảo mật, kinh doanh, v.v.

Các hướng tiếp cận trước đây như: Long Short Term Memory (LSTM), Deep Neural Networks (DNNs), Hidden Markov Models (HMMs)… đã thành công trên những cấp độ và mục đích khác nhau. Trong đề tài này, nhóm chúng tôi sử dụng Fast Fourier Transform (FFT) để biến đổi tín hiệu âm thanh giọng nói thành âm phổ, phục vụ cho mô hình CNN nhằm tạo kết quả đầu ra là vùng miền. Cuối cùng kết quả là dự đoán được vùng miền của giọng nói.

MỤC LỤC

[LỜI CẢM ƠN i](#_Toc104152341)

[PHẦN ĐÁNH GIÁ CỦA GIẢNG VIÊN iii](#_Toc104152342)

[TÓM TẮT iv](#_Toc104152343)

[MỤC LỤC 1](#_Toc104152344)

[DANH MỤC CÁC HÌNH VẼ 4](#_Toc104152345)

[DANH MỤC CÁC BẢNG 4](#_Toc104152346)

[1.1 Giới thiệu về bài toán 5](#_Toc104152347)

[1.2 Phân tích yêu cầu của bài toán 5](#_Toc104152348)

[1.2.1 Yêu cầu của bài toán 5](#_Toc104152349)

[1.2.2 Các phương pháp giải quyết bài toán 6](#_Toc104152350)

[1.2.3 Phương pháp đề xuất giải quyết bài toán 10](#_Toc104152351)

[1.3 Phương pháp giải quyết bài toán 10](#_Toc104152352)

[1.3.1 Mô hình tổng quát 10](#_Toc104152353)

[1.3.2 Đặc trưng mô hình đề xuất 12](#_Toc104152354)

[1.3.2.1 Xử lý dữ liệu tiếng ồn 12](#_Toc104152355)

[1.3.2.2 Tạo tập dữ liệu training và validation 12](#_Toc104152356)

[1.3.2.3 Khởi tạo mô hình CNN 12](#_Toc104152357)

[1.3.2.4 Đào tạo mô hình 14](#_Toc104152358)

[1.3.2.5 Đánh giá mô hình 14](#_Toc104152359)

[1.4 Thực nghiệm 15](#_Toc104152360)

[1.4.1 Dữ liệu 15](#_Toc104152361)

[1.4.2 Xử lý dữ liệu 15](#_Toc104152362)

[1.4.3 Công nghệ sử dụng 16](#_Toc104152363)

[1.4.4 Cách đánh giá 16](#_Toc104152364)

[1.4.4.1 Accuracy 16](#_Toc104152365)

[1.4.4.2 RMSE 17](#_Toc104152366)

[1.5 Kết quả đạt được 17](#_Toc104152367)

[1.5.1 Tham số thực nghiệm 17](#_Toc104152368)

[1.5.2 Kết quả đạt được 18](#_Toc104152369)

[1.6 Kết luận 18](#_Toc104152370)

[1.6.1 Kết quả đạt được 18](#_Toc104152371)

[1.6.2 Hạn chế 19](#_Toc104152372)

[1.6.3 Hướng phát triển trong tương lai 19](#_Toc104152373)

[TÀI LIỆU THAM KHẢO 20](#_Toc104152374)

[TỰ ĐÁNH GIÁ 21](#_Toc104152375)

**DANH MỤC KÍ HIỆU VÀ CHỮ VIẾT TẮT**

**CÁC KÝ HIỆU**

**CÁC CHỮ VIẾT TẮT**

AI Artificial Intelligence

FFT Fast Fourier Transform

CNN Convolutional Neural Network

RMSE Root Mean Squared Error

DANH MỤC CÁC HÌNH VẼ

[Hình 1.2.2.1 Mô hình các lớp cơ bản của CNN 7](#_Toc104152376)

[Hình 1.2.2.2 Ví dụ mô hình CNN 8](#_Toc104152377)

[Hình 1.3.1.1 Mô hình tổng quát 11](file:///E:\HK2_2021-2022\MASSIVE_DATA_PROCESSING\FINAL\51900780%20-%20Nguyen%20The%20Truong%20-%2051900847%20-%20Mai%20Hoang%20Viet%20-%20So%2023%20-%20Report.docx#_Toc104152378)

[Hình 1.5.1 Chỉ số accuracy và loss trong quá trình thực nghiệm 18](#_Toc104152379)

DANH MỤC CÁC BẢNG

[Bảng 1.4.1 Phân loại bộ dữ liệu của đồ án 15](#_Toc104152380)

**NHẬN DẠNG ÂM THANH VÙNG MIỀN VIỆT NAM BẰNG PHƯƠNG PHÁP HỌC SÂU**

* 1. Giới thiệu về bài toán

Việt Nam là đất nước đa dạng các nền văn hóa khác nhau, với 54 dân tộc anh em và sự giao thoa văn hóa giữa các vùng miền tạo nên nét đặc trưng trong đời sống. Ngôn ngữ và phương ngữ địa phương cũng thực sự đa dạng. Theo xu thế đó nhận dạng và hỗ trợ giọng nói từng vùng miền đặt ra vấn đề thiết yếu đối với cuộc sống.

Ngày nay, việc áp dụng công nghệ thông tin vào hầu hết các lĩnh vực của đời sống đã trở nên quá quen thuộc. Tại Việt Nam, trí tuệ nhân tạo đã và đang được ứng dụng mạnh mẽ trong nhiều lĩnh vực như y tế, giáo dục, nông nghiệp, giao thông, thương mại điện tử, ... Công nghệ AI cũng đã mang lại cho Việt Nam sự phát triển vượt bậc trong thời gian qua.

Đồ án trình bày một phương pháp để nhận dạng tiếng nói tiếng Việt bằng việc áp dụng phương pháp học sâu (CNN). Bằng việc sử dụng FFT để biến đổi tín hiệu giọng nói thành âm phổ và kết hợp sử dụng Mạng nơ ron tích chập để phân biệt giọng nói theo vùng miền nước Việt Nam (Bắc, Trung, Nam). Chúng tôi tiến hành thử nghiệm trên tập dữ liệu âm thanh Vivos và bộ dữ liệu tự thu thập đạt độ chính xác tốt về dự đoán vùng miền của giọng nói tiếng Việt. Các nghiên cứu gần đây cho thấy mạng nơ ron tích chập CNN mang lại hiệu quả tốt trong nhận dạng vì nó có tính thích nghi cao và hoạt động ổn định.

* 1. Phân tích yêu cầu của bài toán
     1. Yêu cầu của bài toán

Xây dựng mô hình nhận dạng âm thanh bằng các phương pháp học sâu để trích xuất đặc trưng và phân loại, nhận dạng giọng nói. Qua việc nghiên cứu các phương pháp học sâu cho bài toán nhận dạng giọng nói kết hợp với ngôn ngữ lập trình Python, bài toán này yêu cầu nhận dạng âm thanh đầu vào, xử lý và đưa ra kết quả vùng miền của âm thanh đó: Bắc, Trung, Nam.

Để thực hiện được mục tiêu đề ra, đồ án hướng đến nghiên cứu các vấn đề sau:

* Tập trung chính vào đối tượng nghiên cứu của luận văn là tiếng nói của con người, các kỹ thuật nhận dạng âm thanh và tiếng nói.
* Phạm vi nghiên cứu các phương pháp học sâu, hiệu năng và cách sử dụng, từ đó đề xuất một phương pháp cụ thể hiệu quả nhất.
  + 1. Các phương pháp giải quyết bài toán

Nhóm tôi tiến hành tìm kiếm, tổng hợp và nghiên cứu các tài liệu về Mạng nơ-ron tích chập (CNN), phương pháp biến đổi tín tính hiệu âm thanh giọng nói thành âm phổ, các kỹ thuật trong nhận dạng âm thanh và tiếng nói. Tìm hiểu các kiến thức liên quan đến kỹ thuật, ngôn ngữ lập trình Python.

Mạng nơ-ron tích chập (Convolutional Neural Network - CNN) là một loại mạng nơron nhân tạo (Artificial Neural Networks - ANN) đã được chứng minh là có hiệu suất cao trên các tác vụ hình ảnh khác nhau, bao gồm phân loại hình ảnh, phân đoạn hình ảnh, truy xuất hình ảnh, phát hiện đối tượng, chú thích hình ảnh, nhận dạng khuôn mặt, ước lượng tư thế, nhận dạng dấu hiệu, xử lý giọng nói, …

Mạng nơ-ron tích chập ra đời với cấu trúc khác hẳn mạng nơ-ron truyền thẳng. Thay vì toàn bộ ảnh nối trực tiếp với một node thì chỉ có một phần cục bộ trong ảnh nối với một node trong lớp tiếp theo. Dữ liệu hình ảnh ban đầu qua các lớp của mô hình mạng nơ-ron tích chập sẽ học được ra các đặc trưng để tiến hành phân lớp hiệu quả.

Về cơ bản, CNN bao gồm các lớp sau: lớp tích chập (Convolutional layer), lớp hàm kích hoạt (Activation layer), lớp pooling (Pooling layer) hay còn gọi là lớp subsampling, lớp kết nối đầy đủ (Fully Connected layer).

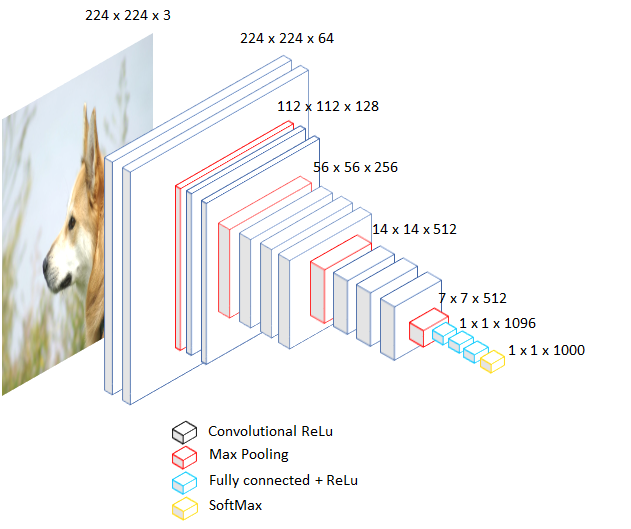


Hình 1.2.2.1 Mô hình các lớp cơ bản của CNN[[1]](#footnote-1)

Trong mô hình CNN (Hình 1.2.2.1), các lớp được liên kết với nhau thông qua cơ chế tích chập. Lớp tiếp theo là kết quả của tích chập của lớp trước đó, nhờ vậy mà ta có được các kết nối cục bộ. Nghĩa là mỗi nơ-ron ở lớp tiếp theo được sinh ra từ các bộ lọc áp đặt lên một vùng ảnh cục bộ của nơ-ron ở lớp trước đó.

Mỗi lớp như vậy được áp đặt các bộ lọc khác nhau, thông thường có vài trăm đến vài nghìn bộ lọc như vậy. Một số lớp khác như pooling/subsampling được dùng để chắt lọc lại các thông tin hữu ích hơn.

Trong suốt quá trình huấn luyện, mạng nơ-ron tích chập tự động học được các thông số cho các bộ lọc.



Hình 1.2.2.2 Ví dụ mô hình CNN[[2]](#footnote-2)

**Convolutional Layer**

Đây là lớp quan trọng nhất của CNN, lớp này có nhiệm vụ thực hiện mọi tính toán. Những yếu tố quan trọng của một lớp tích chập là: stride, padding, filter map, feature map.

* CNN dùng các bộ lọc để áp dụng vào các vùng của hình ảnh. Những bộ lọc này được gọi là ma trận 3 chiều với các con số bên trong và nó là các tham số.
* Stride là sự chuyển dịch của bộ lọc theo pixel dựa vào giá trị từ trái sang phải. Khi stride là 1 thì sẽ di chuyển các bộ lọc từng pixel một. Khi stride là 2 thì các bộ lọc nhảy 2 pixel tại một thời điểm.
* Padding cho phép kiểm soát kích thước đầu ra. Áp dụng tích chập cho một đầu vào nếu giảm kích thước đầu ra dễ dẫn đến mất thông tin. Để tránh điều đó, thông thường sẽ thêm các giá trị 0 vào lớp input.
* Feature Map là kết quả của mỗi lần bộ lọc quét qua input. Sau mỗi lần quét sẽ xảy ra các quá trình tính toán.

**Relu Layer**

Relu layer hay còn gọi là Activation Function là hàm kích hoạt trong mạng nơ-ron. Tác dụng chính của hàm này là mô phỏng các nơ-ron có tỷ lệ xung qua axon. Các hàm cơ bản là: Sigmod, Tanh, Relu, Maxout,…

Nó được sử dụng nhiều cho các nhu cầu huấn luyện mạng nơ-ron vì có nhiều ưu điểm nổi bật như: việc tính toán sẽ trở nên nhanh hơn,... Relu Layer được sử dụng sau khi bộ lọc được tính ra và áp dụng hàm Relu lên những giá trị của bộ lọc đó.

**Pooling Layer**

Pooling Layer thường được sử dụng sau các lớp tích chập, giúp giảm kích thước của khối nếu đầu vào quá lớn. Việc giảm kích thước dữ liệu có tác dụng làm giảm được số lượng tham số cũng như tăng hiệu quả tính toán. Các tham số của lớp Pooling bao gồm kích thước bộ lọc và Stride.

Hai dạng Pooling Layer phổ biến là Max Pooling Layer và Averge Pooling Layer. Max Pooling Layer được sử dụng phổ biến hơn Averge Pooling Layer

**Fully Connected Layer**

Sau một số lớp tích chập và Pooling, CNN thường kết thúc với một số lớp kết nối đầy đủ.

Lớp kết nối đầy đủ này được thiết kế hoàn toàn tương tự như trong mạng nơron truyền thống, tức là mỗi nút trong một lớp fully-connected layer được kết nối trực tiếp với mọi nút trong cả lớp trước và trong lớp tiếp theo. So với mạng nơ-ron truyền thống, các ảnh đầu vào của lớp này đã có kích thước được giảm bớt rất nhiều, đồng thời vẫn đảm bảo các thông tin quan trọng cho việc nhận dạng. Do vậy, việc tính toán nhận dạng sử dụng mô hình truyền thẳng đã không còn phức tạp và tốn nhiều thời gian như trong mạng nơ-ron truyền thống.

Sau khi nghiên cứu lý thuyết, xác định vấn đề bài toán, đề xuất mô hình, đồ án sẽ thực nghiệm mô hình trên một phương pháp học sâu, kết hợp giữa những lợi thế của mô hình CNN tự tạo và các thư viện thích hợp hỗ trợ cho việc xây dựng một hệ thống nhận âm thanh đầu vào, qua quá trình xử lý cho ra kết quả là nhận dạng vùng miền của âm thanh, giọng nói.

* + 1. Phương pháp đề xuất giải quyết bài toán

Đồ án đề xuất một phương pháp dự đoán vùng miền của giọng nói tiếng Việt dựa trên âm phổ sử dụng phương pháp học sâu. Từ tập dữ liệu thô, tiến hành giai đoạn tiền xử lý để đưa tập dữ liệu âm thanh về cùng một tiêu chuẩn tần số và thời gian. Sau đó biến đổi tín hiệu âm thành thành âm phổ sử dụng FFT rồi đưa vào mô hình học sâu - Mạng nơ ron tích chập (CNN) để tiến hành đào tạo và tối ưu hóa. Các thực nghiệm được thực hiện trên tập dữ liệu âm thanh Vivos và bộ dữ liệu tự thu thập đạt độ chính xác tốt về dự đoán vùng miền của giọng nói tiếng Việt.

* 1. Phương pháp giải quyết bài toán
     1. Mô hình tổng quát

Đào tạo mô hình

Đánh giá mô hình trên tập validation

Khởi tạo mô hình

Tạo tập dữ liệu training và validation

Xử lý dữ liệu tiếng ồn

Hình 1..1.1 Mô hình tổng quát

Mô hình gồm có 5 phần:

* Phần 1: Xử lý dữ liệu tiếng ồn về cùng một sample rate và có độ dài 1 giây.
* Phần 2: Tạo tập dữ liệu training và validation từ tập dữ liệu thô, đưa về cùng sample rate và thêm các tiếng ồn đã xử lý ở trên vào.
* Phần 3: Khởi tạo mô hình CNN.
* Phần 4: Đào tạo mô hình với tập dữ liệu training
* Phần 5: Đánh giá mô hình trên tập validation
  + 1. Đặc trưng mô hình đề xuất
       1. Xử lý dữ liệu tiếng ồn

Quy trình thực hiện như sau:

* Lấy ra noise\_paths chứa danh sách đường dẫn của tất cả các mẫu tiếng ồn, có 6 tệp trong 2 thư mục.
* Chuyển tần số âm thanh của tất cả các mẫu tiếng ồn về 16000Hz.
* Chia các mẫu tiếng ồn thành các mẫu có độ dài 1 giây và đưa vào noises. Tổng cộng có 354 mẫu.
  + - 1. Tạo tập dữ liệu training và validation

Quy trình thực hiện như sau:

* Lấy danh sách các đường dẫn của tệp âm thanh (audio\_paths) cùng với nhãn tương ứng (labels) (nhãn chính là tên thư mục chứa tệp âm thanh đó). Có 623 tệp thuộc 3 nhãn (bắc, trung, nam).
* Chuyển tần số âm thanh các tệp âm thanh trong audio\_paths về 16000Hz.
* Xáo trộn các đường dẫn trong audio\_paths và labels.
* Chia danh sách tệp thành 2 phần tương ứng với training và validation với tỷ lệ VALID\_SPLIT = 0.2.
* Traning: train\_audio\_paths, train\_labels (499 tệp).
* Validation: valid\_audio\_paths, valid\_label (124 tệp).
* Tạo tập dữ liệu train\_ds, valid\_ds từ train\_audio\_paths và valid\_audio\_paths.
* Thêm các tiếng ồn vào tập dữ liệu train\_ds và valid\_ds.
* Chuyển đổi sóng âm của tập dữ liệu train\_ds, valid\_ds thành âm phổ.
  + - 1. Khởi tạo mô hình CNN

Với hàm residual\_block dùng để:

* Tạo Conv1D dùng để lấy đặc trưng của giọng nói với các tham số:
* filters : số bộ lọc của Convolutional.
* kernel\_size: Chỉ định độ dài của cửa sổ 1D Convolutional Layers.
* padding: có thể là valid, causal hoặc same. Với same thì có nghĩa kích thước đầu vào và đầu ra là giống nhau.
* Tạo Activation Layers với tham số activation là relu.
* Tạo Add Layers để thêm danh sách các input.
* Hàm residual\_block  trả về Pooling Layers dùng để giảm param khi train nhưng vẫn giữ được đặc trưng của giọng nói với các tham số:
* pool\_size: kích thước ma trận để lấy max.
* stride: số bước nhảy của giọng nói.

Với hàm build\_model dùng để:

* Tạo Input Layer với các tham số:
* Gọi lại hàm residual\_block với các tham số filter là 32, 64, 128.
* Tạo AveragePooling1D Layer dùng để lấy trung bình cho từng size với các tham số:
* pool\_size: kích thước ma trận để lấy max.
* stride: số bước nhảy của giọng nói.
* Gọi lại hàm residual\_block với các tham số filter là 32, 64, 128.
* Tạo AveragePooling1D Layer dùng để lấy trung bình cho từng size với các tham số:
* pool\_size: kích thước ma trận để lấy max.
* stride: số bước nhảy của giọng nói.
* Tạo Flatten Layer không tham số.
* Tạo Dense Layer với các tham số:
* units: chiều của không gian output.
* activation: chọn activation là relu.
* Hàm build\_model trả về Model với các tham số inputs, outputs đã được định nghĩa trước đó.

Khởi tạo model truyền vào tham số là input\_shape và số lượng các class names.

Compile model sử dụng thuật toán tối ưu Adam với loss là sparse\_categorical\_crossentropy dùng để tính toán tổn thất chéo giữa các nhãn, dự đoán và độ đo là accuracy.

Xây dựng các callbacks:

* ModelsCheckpoint: luôn giữ cho mô hình có độ chính xác tốt nhất.
* EarlyStopping : dừng việc huấn luyện khi chất lượng mô hình không được cải thiện.
  + - 1. Đào tạo mô hình

Tiến hành đào tạo mô hình với các thông số:

* Dữ liệu training: train\_ds.
* epoch: 100.
* validation\_data: valid\_ds
* callbacks: [earlystopping\_cb, mdlcheckpoint\_cb]
  + - 1. Đánh giá mô hình
* Đánh giá mô hình trên tập dữ liệu valid\_ds: model.evaluate(valid\_ds).
* Tạo tập dữ liệu test\_ds từ các đường dẫn tệp âm thanh dùng để validate (valid\_audio\_paths) và nhãn tương ứng (valid\_labels).
* Thêm các tiếng ồn vào tập dữ liệu test\_ds.
* Lấy ngẫu nhiên 20 mẫu trong tập test\_ds, với mỗi mẫu tiến hành đưa vào mô hình để dự đoán kết quả. Sau khi dự đoán xong, in ra màn hình kết quả dự đoán được để so sánh với kết quả thực tế.
  1. Thực nghiệm
     1. Dữ liệu

Tất cả các tệp âm thanh đều được định dạng với đuôi wav.

Bộ dữ liệu được chia thành 2 thư mục:

* audio (dữ liệu âm thanh giọng nói): chứa 3 thư mục “bắc”, “trung”, “nam” tương ứng với giọng nói của 3 vùng miền Bắc, Trung, Nam. Với 2 thư mục “bắc” và “nam”, các tệp âm thanh được thu thập từ bộ dữ liệu Vivos Corpus thuộc Lab khoa Khoa học máy tính, Trường Đại học Khoa học Tự nhiên Thành phố Hồ Chí Minh. Tuy nhiên, bộ dữ liệu này chỉ có những tệp âm thanh của miền Bắc và miền Nam, do vậy nhóm tôi đã tự thu thập các tệp âm thanh của miền Trung và đưa vào thư mục “trung”.
* noise (dữ liệu tiếng ồn): chứa 2 thư mục background\_noise, other chứa các tệp âm thanh tiếng ồn. Tập dữ liệu này được lấy từ trang chủ keras.io, trong bài toán Speaker Recognition.

Cuối cùng, nhóm tôi đã thu thập được bộ dữ liệu sau:

|  |  |
| --- | --- |
| **Khu vực** | **Số lượng** |
| Miền Bắc | 215 |
| Miền Trung | 246 |
| Miền Nam | 162 |
| Dữ liệu tiếng ồn | 6 |

Bảng 1.4.1 Phân loại bộ dữ liệu của đồ án

* + 1. Xử lý dữ liệu

Trên mỗi tập tin dữ liệu âm thanh, chúng tôi thực hiện xử lý dữ liệu bằng cách cắt các tập tin âm thanh có thời lượng 1 giây, sau đó giảm tần số âm thanh xuống 16000 Hz nhằm mục đích đưa các mẫu dữ liệu về một chuẩn chung.

Cụ thể, đối với dữ liệu tiếng ồn:

* Sử dụng các thư viện xử lý tệp tin và âm thanh để giảm tần số âm thanh xuống 16000Hz.
* Sau đó chia nhỏ các tệp này thành các tệp con có độ dài 1 giây.

Đối với dữ liệu âm thanh giọng nói:

* Sử dụng các thư viện xử lý tệp tin và âm thanh để giảm tần số âm thanh xuống 16000Hz.
* Sau đó, thực hiện xáo trộn các tệp tin và chia ra 2 phần dùng để huấn luyện và đánh giá (train, valid).
* Tiếp tục, tạo ra 2 tập dữ liệu từ 2 phần được chia ra ở trên.
* Thêm các âm thanh tiếng ồn vào 2 tập dữ liệu.
* Chuyển đổi sóng âm của 2 tập dữ liệu thành âm phổ.

Mục đích của quá trình này là chuẩn hóa dữ liệu đầu vào giúp cho mạng CNN học được các đặc trưng của giọng nói tốt hơn. Ngoài ra còn đảm bảo được sự đồng nhất trong dữ liệu huấn luyện.

* + 1. Công nghệ sử dụng
* Ngôn ngữ lập trình: Python.
* Môi trường phát triển: Google Colab, Google Drive.
* Các thư viện được sử dụng: os, shutil, numpy, tensorflow, pathlib, IPython, soundfile, librosa.
  + 1. Cách đánh giá

Để đánh giá mô hình xây dựng đã tối ưu hay chưa ta cần có các phương pháp đánh giá chuẩn xác. Đối với đồ án này thì các độ đo dùng để đánh giá là Accuracy và RMSE.

* + - 1. Accuracy

Accuracy dùng để đo độ chính xác của mô hình phổ biến nhất. Accuracy giúp cho chúng ta biết được khái quát tỷ lệ các trường hợp được dự báo đúng trên tổng số các trường hợp là bao nhiêu. Accuracy giúp ta đánh giá hiệu quả dự báo của mô hình trên một bộ dữ liệu. Accuracy càng cao thì mô hình của chúng ta càng chuẩn xác.

Tuy nhiên Accuracy có một nhược điểm chính là do nó sử dụng tần suất nên đối với bài toán có sự chênh lệch về số lượng mẫu, việc sử dụng Accuracy sẽ không hoàn toàn chính xác.

Accuracy được tính theo công thức:

* + - 1. RMSE

RMSE là thước đo mức độ hiệu quả của một mô hình. Nó thực hiện điều này bằng cách đo sự khác biệt giữa các giá trị dự đoán và giá trị thực tế. RMSE càng nhỏ tức là sai số càng bé thì mức độ ước lượng cho thấy độ tin cậy của mô hình có thể đạt cao nhất.

Trong đó:

là giá trị ước lượng

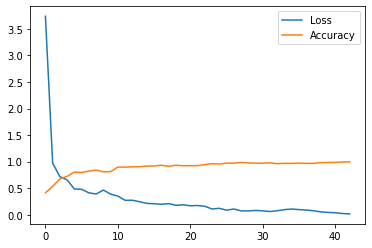
là biến độc lập

là tổng lượng quan sát

là tổng lượng biến

* 1. Kết quả đạt được
     1. Tham số thực nghiệm
* Tập dữ liệu truyền vào: train\_ds.
* epochs: 100.
* validation\_data: valid\_ds.
* callbacks: 2 callback là earlystopping\_cb, mdlcheckpoint\_cb đã định nghĩa.

Kết quả thực nghiệm:



Hình 1.5.1 Chỉ số accuracy và loss trong quá trình thực nghiệm

* + 1. Kết quả đạt được

Kết quả đạt được:

* Chỉ số accuracy: 0.9597.
* Chỉ số RMSE: 0.2978416985906353.
  1. Kết luận
     1. Kết quả đạt được

Đồ án đã đề xuất giải pháp cho việc phân loại vùng miền của giọng nói tiếng Việt bằng cách sử dụng mô hình mạng nơron tích chập (CNN), kết hợp biến đổi sóng âm âm thanh thành âm phổ và sử dụng CNN để xác định vùng miền tiếng Việt. Đồ án đã tiến hành thử nghiệm trên tập dữ liệu tự thu thập và bộ dữ liệu Vivos Corpus thuộc Lab khoa Khoa học máy tính – Trường Đại học Khoa học Tự nhiên Thành phố Hồ Chí Minh để đánh giá kết quả. Kết quả đạt được như sau:

* Phân tích và đánh giá mô hình đề xuất.
* Xử lý dữ liệu âm thanh phù hợp với mạng CNN.
* Xây dựng được hệ thống nhận dạng tiếng nói tiếng Việt bằng phương pháp học sâu sử dụng mạng nơ ron tích chập CNN.
  + 1. Hạn chế

Đồ án chỉ thực hiện biến đổi âm thanh bằng cách chuyển đổi sóng âm thành âm phổ và sử dụng mô hình học sâu CNN để giải quyết, chưa tiến hành thử nghiệm trên các cách trích xuất đặc trưng, mô hình, thông số khác nên chưa thể tìm ra phương pháp tối ưu, hiệu quả nhất.

* + 1. Hướng phát triển trong tương lai
* Chuẩn hóa, xây dựng bộ dữ liệu âm thanh có độ tương đồng cao, giảm tối đa độ nhiễu.
* Ngoài việc dùng FFT kết hợp mạng nơ-ron tích chập CNN, cần thực hiện thử nghiệm các phương pháp trích xuất đặc trưng, mô hình, thông số khác nhau để đưa ra kết quả so sánh, đánh giá.
* Tối ưu hóa giải thuật, lựa chọn các thông số phù hợp cho mạng CNN nhằm tăng khả năng nhận dạng giọng nói vùng miền Việt Nam.

TÀI LIỆU THAM KHẢO

1. (n.d.). Wikipedia, the free encyclopedia. Retrieved May 22, 2022, from https://arxiv.org/ftp/arxiv/papers/1908/1908.07656.pdf.
2. *[2107.11412] Using Deep Learning Techniques and Inferential Speech Statistics for AI Synthesized Speech Recognition*. (2021, July 23). arXiv. Retrieved May 22, 2022, from https://arxiv.org/abs/2107.11412
3. A Novel Method for Recognizing Vietnamese Voice Commands on Smartphones with Support Vector Machine and Convolutional Neural Networks. (2020, March 1). Hindawi. Retrieved May 22, 2022, from https://www.hindawi.com/journals/wcmc/2020/2312908/
4. Vietnamese Speech Command Recognition using Recurrent Neural Networks. (n.d.). Semantic Scholar. Retrieved May 22, 2022, from https://pdfs.semanticscholar.org/aa38/40903314eb34025de28fa5362b9e7bb82832.pdf.
5. Badine, F. (2020, June 14). Speaker Recognition - Audio Data. Keras. Retrieved May 22, 2022, from https://keras.io/examples/audio/speaker\_recognition\_using\_cnn/
6. Trần, H. T. (2021, 9). NHẬN DẠNG TIẾNG NÓI TIẾNG VIỆT BẰNG PHƯƠNG PHÁP HỌC S U. LUẬN VĂN THẠC SĨ.
7. *Download*. (n.d.). AILab. Retrieved May 22, 2022, from https://ailab.hcmus.edu.vn/vivos

TỰ ĐÁNH GIÁ

(Với nhóm có 2 thành viên)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Câu | Nội dung | Điểm chuẩn | Tự chấm | Ghi chú |
| 1  (8.5) | **1.1 Giới thiệu về bài toán** | 0.5 | 0.5 |  |
| **1.2 Phân tích yêu cầu của bài toán** | 1.0 | 1.0 |  |
| **1.3 Phương pháp giải quyết bài toán** | 1.5 | 0.75 |  |
| **1.4 Thực nghiệm** | 4 | 3.5 |  |
| **1.5 Kết quả đạt được** | 1 | 0.25 |  |
| **1.6 Kết luận** | 0.5 | 0.5 |  |
| 2 | **Điểm nhóm** | 0.5 | 0.5 |  |
| 3 | **Báo cáo** (chú ý các chú ý 2,3,4,6 ở trang trước, nếu sai sẽ bị trừ điểm nặng) | 1 | 1 |  |
| **Tổng điểm** | | | 8.0 |  |

1. https://topdev.vn/blog/thuat-toan-cnn-convolutional-neural-network/ [↑](#footnote-ref-1)
2. https://towardsdatascience.com/convolutional-neural-networks-in-practice-406426c6c19a [↑](#footnote-ref-2)