

TRƯỜNG ĐẠI HỌC KỸ THUẬT – CÔNG NGHỆ CẦN THƠ

KHOA CÔNG NGHỆ THÔNG TIN

GIẢNG VIÊN HƯỚNG DẪN: THS. HÀ LÊ NGỌC DUNG

SINH VIÊN THỰC HIỆN:

Phan Nguyễn Vũ Huy MSSV: KHDL2211010

Vỏ Văn Tài MSSV: KHDL2211020

Huỳnh Chí Phi Thuận MSSV: KHDL2211038

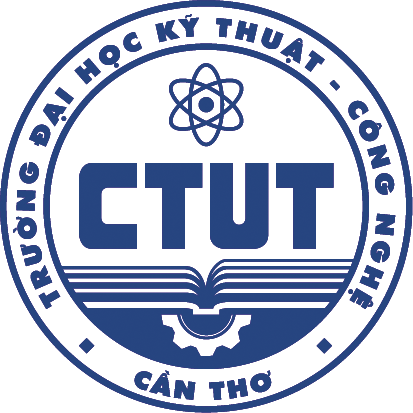
PHƯƠNG PHÁP NGHIÊN CỨU PHÂN LOẠI

ÂM THANH BẰNG CNN

NGÀNH: KHOA HỌC DỮ LIỆU

Cần Thơ, tháng 8 năm 2024

ĐỒ ÁN 1



PHƯƠNG PHÁP NGHIÊN CỨU PHÂN LOẠI

ÂM THANH BẰNG CNN

NGÀNH: KHOA HỌC DỮ LIỆU

GIẢNG VIÊN HƯỚNG DẪN: THS. HÀ LÊ NGỌC DUNG

SINH VIÊN THỰC HIỆN:

Phan Nguyễn Vũ Huy MSSV: KHDL2211010

Vỏ Văn Tài MSSV: KHDL2211020

Huỳnh Chí Phi Thuận MSSV: KHDL2211038

TRƯỜNG ĐẠI HỌC KỸ THUẬT – CÔNG NGHỆ CẦN THƠ

KHOA CÔNG NGHỆ THÔNG TIN

Cần Thơ, tháng 8 năm 2024

ĐỒ ÁN 1

PHIẾU NHẬN XÉT GIẢNG VIÊN HƯỚNG DẪN

**Tên đề tài:**

PHƯƠNG PHÁP NGHIÊN CỨU PHÂN LOẠI ÂM THANH BẰNG CNN

Tên SVTH: PHAN NGUYỄN VŨ HUY MSSV: KHDL2211010

VỎ VĂN TÀI MSSV: KHDL2211020

HUỲNH CHÍ PHI THUẬN MSSV: KHDL2211038

**Họ và tên GVHD:** ThS. HÀ LÊ NGỌC DUNG

**Nhận xét của giảng viên hướng dẫn:**

·································································································

·································································································

·································································································

·································································································

·································································································

·································································································

·································································································

·································································································

·································································································

·································································································

Cần thơ, ngày...tháng...năm 2024

**GIẢNG VIÊN HƯỚNG DẪN**

ThS. HÀ LÊ NGỌC DUNG

PHIẾU NHẬN XÉT CỦA GIẢNG VIÊN PHẢN BIỆN

Giảng viên phản biện:

**Nhận xét của giảng viên phản biện:**

·································································································

·································································································

·································································································

·································································································

·································································································

·································································································

·································································································

·································································································

·································································································

·································································································

Cần thơ, ngày...tháng...năm 2024

GIẢNG VIÊN PHẢN BIỆN

LỜI CAM ĐOAN

Chúng tôi bao gồm Vũ Huy, Phi Thuận, Văn Tài xin cam đoan rằng đề tài “Phương pháp nghiên cứu phân loại âm thanh bằng CNN” trong đồ án của chúng tôi đã được tiến hành một cách công khai và minh bạch, dựa trên sự cố gắng và nỗ lực của bản thân cũng như sự hướng dẫn tận tình từ giảng viên ThS. Hà Lê Ngọc Dung

Các số liệu nghiên cứu được trình bày trong đồ án đảm bảo tính trung thực, không có việc sao chép hay sử dụng kết quả từ bất kỳ công trình nào đã được công bố trước đó. Tôi cam đoan rằng nếu phát hiện có sự sao chép, chúng tôi sẽ chịu hoàn toàn trách nhiệm và sẵn sàng chấp nhận kỷ luật từ phía nhà trường.

Cần Thơ, ngày…tháng…năm 2024

|  |  |  |
| --- | --- | --- |
| **Sinh viên 1**  **Phan Nguyễn Vũ Huy** | **Sinh viên 2**  **Huỳnh Chí Phi Thuận** | **Sinh viên 3**  **Vỏ Văn Tài** |

LỜI CẢM ƠN

Chúng em xin gửi lời cảm ơn chân thành đến giảng viên ThS. Hà Lê Ngọc Dung Khoa Công nghệ thông tin đã trang bị cho em những kiến thức, kỹ năng cơ bản cần có để hoàn thành đề tài nghiên cứu này.

Tuy nhiên, trong quá trình nghiên cứu đề tài, do kiến thức chuyên ngành còn hạn chế nên em vẫn còn nhiều thiếu sót khi tìm hiểu, đánh giá và trình bày về đề tài. Rất mông nhận được sự quan tâm, góp ý của thầy/ cô giảng viên trong khoa để đề tài của em được đầy đủ và hoàn chỉnh hơn.

Xin chân thành cảm ơn.

Cần Thơ, ngày…tháng…năm 2024

TÓM TẮT ĐỒ ÁN

Đề tài nghiên cứu phương pháp phân loại âm thanh bằng mạng nơ-ron tích chập (CNN) nhằm khai thác tiềm năng của CNN trong việc xử lý và nhận diện các mẫu âm thanh phức tạp. Với sự phát triển mạnh mẽ của công nghệ xử lý âm thanh và nhu cầu ngày càng tăng về các hệ thống phân loại âm thanh chính xác trong các ứng dụng như trợ lý ảo, hệ thống điều khiển bằng giọng nói và phân tích âm nhạc, nghiên cứu này tập trung vào việc cải tiến các phương pháp hiện có và khám phá các kỹ thuật mới để nâng cao hiệu quả và độ chính xác của quá trình phân loại âm thanh.

MỤC LỤC

[MỤC LỤC i](#_Toc174564493)

[DANH MỤC HÌNH iii](#_Toc174564494)

[DANH MỤC TỪ VIẾT TẮT v](#_Toc174564495)

[CHƯƠNG 1: GIỚI THIỆU ĐỀ TÀI 1](#_Toc174564496)

[1.1. Lý Do Chọn Đề Tài 1](#_Toc174564497)

[1.2. Mục Tiêu Nghiên Cứu 1](#_Toc174564498)

[1.3. Phương Pháp Nghiên Cứu 2](#_Toc174564499)

[CHƯƠNG 2: CƠ SỞ LÝ THUYẾT 2](#_Toc174564500)

[2.1. Tìm Hiểu Về Máy Học 2](#_Toc174564501)

[2.1.1. Máy học là gì ? 2](#_Toc174564502)

[2.1.2. Lịch Sử Hình Thành Máy Học 3](#_Toc174564503)

[2.1.3. Phân Loại Máy Học 5](#_Toc174564504)

[2.1.4. Cách hoạt động của thuật toán trong máy học: 7](#_Toc174564505)

[2.2. Tìm Hiểu Về Deep Learning 9](#_Toc174564506)

[2.2.1. Tổng quan về Deep Learning 9](#_Toc174564507)

[2.2.2. Cách hoạt động của Deep Learning: 10](#_Toc174564508)

[2.3. Khái Niệm Về CNN 11](#_Toc174564509)

[2.3.1. Tổng quan về CNN (Convolutional Neural Networks) 11](#_Toc174564510)

[2.3.2. Cách hoạt động của CNN 13](#_Toc174564511)

[2.3.3. Kiến trúc mạng CNN 16](#_Toc174564512)

[2.4. Tìm Hiểu Về PyCharm 17](#_Toc174564513)

[2.4.1. Khái niệm PyCharm: 17](#_Toc174564514)

[2.4.2. Lý do chọn PyCharm cho đề tài: 18](#_Toc174564515)

[2.4.3. Các tính năng PyCharm hữu ích cho đề tài: 18](#_Toc174564516)

[2.4.4. Các bước thực hiện với PyCharm 20](#_Toc174564517)

[2.5. Ứng Dụng Các Cơ Sở Lý Thuyết Vào Đề Tài: 21](#_Toc174564518)

[CHƯƠNG 3: Dữ liệu âm thanh và các phương pháp xử lý 24](#_Toc174564519)

[3.1. Dữ liệu Âm Thanh 24](#_Toc174564520)

[3.1.1: Khái niệm cơ bản về âm thanh: 24](#_Toc174564521)

[3.1.2. Dữ liệu âm thanh số hóa 25](#_Toc174564522)

[3.1.3. Các Phương Pháp Xử Lý Dữ Liệu Âm Thanh 28](#_Toc174564523)

[3.1.4. Sử Dụng CNN để Phân Loại Âm Thanh 31](#_Toc174564524)

[CHƯƠNG 4: Xây dựng mô hình máy học xử lý âm thanh 35](#_Toc174564525)

[4.1. Xác định vấn đề, bài toán cần giải quyết 35](#_Toc174564526)

[4.2. Xây dựng và tìm hiểu dữ liệu 36](#_Toc174564527)

[4.2.1. Nguồn Gốc Dữ Liệu 37](#_Toc174564528)

[4.2.2. Cấu Trúc Dữ Liệu 37](#_Toc174564529)

[4.2.3. Loại Dữ Liệu 38](#_Toc174564530)

[4.2.4. Các Thách Thức 38](#_Toc174564531)

[4.2.5. Khám Phá Dữ Liệu (Data Exploration) 39](#_Toc174564532)

[4.2.6. Mục Đích Sử Dụng Dữ Liệu 39](#_Toc174564533)

[4.3. Tiền xử lý dữ liệu và biến âm thanh thành Mel Spectrogram 39](#_Toc174564534)

[4.3.1. Hàm Tạo Spectrogram Từ Tệp WAV 40](#_Toc174564535)

[4.3.2. Tạo và Lưu Spectrogram cho Dữ Liệu Âm Thanh 48](#_Toc174564536)

[4.4. Huấn Luyện Mô Hình với mạng CNN (Convolutional Neural Network) 50](#_Toc174564537)

[4.4.2. train\_model 52](#_Toc174564538)

[4.4.3. test\_model 59](#_Toc174564539)

[4.4.4. Kết Luận 63](#_Toc174564540)

[4.5. ánh Giá Mô hình 64](#_Toc174564541)

[CHƯƠNG 5: KẾT LUẬN 65](#_Toc174564542)

[5.1. Hiểu Biết về Phân Loại Âm Thanh và CNN 65](#_Toc174564543)

[5.2. Xây dựng và Đánh Giá Mô Hình CNN 65](#_Toc174564544)

[5.3. Ứng Dụng và Tinh Chỉnh 66](#_Toc174564545)

[5.4. ĐÁNH GIÁ KẾT QUẢ 66](#_Toc174564546)

[5.4.1. Đánh giá triển khai tổng quan 66](#_Toc174564547)

[5.4.2. Đề Xuất Cải Tiến 67](#_Toc174564548)

[5.5. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN 68](#_Toc174564549)

[5.5.1. Kết Luận 68](#_Toc174564550)

[5.5.2. Hướng Phát Triển 68](#_Toc174564551)

[5.6. HẠN CHẾ 69](#_Toc174564552)

[TÀI LIỆU KHAM KHẢO 70](#_Toc174564553)

DANH MỤC HÌNH

[Hình 2.1: Học máy (Machine Learning) 3](#_Toc174565942)

[Hình 2.2: Quá trình phát triển của máy học 5](#_Toc174565943)

[Hình 2.3: Phân loại trong máy học 7](#_Toc174565944)

[Hình 2.4: Quá trình hoạt động của máy học 9](#_Toc174565945)

[Hình 2.5: Quá trình hoạt động của học sâu 11](#_Toc174565946)

[Hình 2.6: Lớp tích chập 13](#_Toc174565947)

[Hình 2.7: Lớp gộp 14](#_Toc174565948)

[Hình 2.8: Lớp kết nối đầy đủ 14](#_Toc174565949)

[Hình 2.9: Mô hình CNN hoạt động 15](#_Toc174565950)

[Hình 3.1: Minh họa về âm thanh 24](#_Toc174565951)

[Hình 3.2: minh họa quá trình số hóa âm thanh 26](#_Toc174565952)

[Hình 3.3: Phép toán DFT thuận 28](#_Toc174565953)

[Hình 3.4: Phép toán DFT nghịch 28](#_Toc174565954)

[Hình 3.5: Tín hiệu trong miền thời gian P(t) được biểu diễn trên miền tần số P(v) nhờ phép biến đổi Fourier rời rạc 29](#_Toc174565955)

[Hình 3.6: STFT thời gian liên tục 29](#_Toc174565956)

[Hình 3.7: STFT thời gian rời rạc 30](#_Toc174565957)

[Hình 3.8: STFT sử dụng phân tích tín hiệu âm thanh theo thời gian 30](#_Toc174565958)

[Hình 3.9: Biểu diễn trực quang bằng Spectrogram. 31](#_Toc174565959)

[Hình 3.10: Phát hiện âm thanh bất thường với 34](#_Toc174565960)

[Hình 4.1: Hàm Tạo Spectrogram Từ Tệp WAV 40](#_Toc174565961)

[Hình 4.2: Trực quan Spectrogram trên đồ thị 44](#_Toc174565962)

[Hình 4.3: giá trị Decibel Scale của một số loại âm thanh khác nhau 45](#_Toc174565963)

[Hình 4.4: quá trình học đặc trưng (feature learning) cho âm thanh 51](#_Toc174565964)

[Hình 4.5: Các thư viện cần thiết để xử lý dữ liệu âm thanh, tạo spectrograms, và huấn luyện mô hình CNN 53](#_Toc174565965)

[Hình 4.6: Quá trình train mô hình 63](#_Toc174565966)

[Hình 4.7: Test mô hình 64](#_Toc174565967)

DANH MỤC TỪ VIẾT TẮT

|  |  |  |
| --- | --- | --- |
| **TT** | **Ký hiệu chữ viết tắt** | **Chữ viết tắt** |
| 1 | CNN | Convolutional Neural Network |
| 2 | AI | Artificial Intelligence |
| 3 | LSTM | Long Short-Term Memory |
| 4 | RNN | Recurrent Neural Network |
| 5 | GPU | Tensor Processing Unit |
| 6 | IDE | Integrated Development Environment |
| 7 | WAV | Waveform Audio File Format |
| 8 | FLAC | Free Lossless Audio Codec |
| 9 | TPU | Tensor Processing Unit |

# GIỚI THIỆU ĐỀ TÀI

## Lý Do Chọn Đề Tài

Công nghệ xử lý âm thanh đang phát triển mạnh mẽ và trở thành một phần quan trọng trong nhiều ứng dụng thực tế như nhận diện giọng nói, phân tích cảm xúc qua giọng nói và hệ thống an ninh. Việc nghiên cứu và phát triển các phương pháp mới, hiệu quả hơn để phân loại âm thanh có thể mang lại nhiều lợi ích cho các lĩnh vực này. CNN đã chứng minh được hiệu quả vượt trội trong việc xử lý và nhận diện hình ảnh, và nhờ khả năng tự động trích xuất các đặc trưng quan trọng từ dữ liệu đầu vào, CNN cũng được kỳ vọng sẽ mang lại kết quả tốt trong phân loại âm thanh. Với nhu cầu tăng cao về hệ thống phân loại âm thanh chính xác cho các ứng dụng như trợ lý ảo thông minh, hệ thống điều khiển bằng giọng nói và phân tích âm nhạc, việc nghiên cứu và cải tiến các phương pháp phân loại âm thanh trở nên cần thiết hơn bao giờ hết. Mặc dù CNN đã được ứng dụng rộng rãi trong xử lý hình ảnh, việc áp dụng CNN vào phân loại âm thanh vẫn còn nhiều thách thức, mở ra nhiều cơ hội nghiên cứu để khám phá và giải quyết các vấn đề liên quan.

## Mục Tiêu Nghiên Cứu

Mục tiêu chính của nghiên cứu này là phát triển một mô hình CNN hiệu quả để phân loại chính xác các loại âm thanh khác nhau, từ đó nâng cao hiệu quả và độ tin cậy của các hệ thống tự động. Đồng thời, nghiên cứu nhằm nâng cao độ chính xác của mô hình phân loại âm thanh thông qua việc áp dụng các kỹ thuật tiên tiến và tối ưu hóa thuật toán, nhằm đạt được kết quả tốt nhất trên các bộ dữ liệu âm thanh thực tế.

## Phương Pháp Nghiên Cứu

Nghiên cứu lý thuyết: Tìm hiểu về mô hình CNN, đặc biệt là CNN trong phân loại âm thanh, bao gồm cấu trúc mô hình, thuật toán và các cải tiến so với các phiên bản trước đó.

Thu thập và tiền xử lý dữ liệu: Thu thập các bộ dữ liệu âm thanh đa dạng từ các nguồn khác nhau. Tiền xử lý dữ liệu bao gồm cắt xén, loại bỏ nhiễu, chuẩn hóa, và chuyển đổi âm thanh thành các đặc trưng phù hợp

Xây dựng mô hình CNN: Thiết kế kiến trúc CNN với các lớp tích chập, lớp gộp, tùy chỉnh các tham số như số lượng lớp, kích thước bộ lọc. Huấn luyện mô hình trên dữ liệu âm thanh đã được tiền xử lý

Tối ưu hóa và điều chỉnh mô hình: Thực hiện các thử nghiệm để tinh chỉnh các tham số của mô hình nhằm đạt được hiệu quả cao nhất.

Đánh giá và kiểm định mô hình: Đánh giá mô hình bằng các chỉ số đo lường như độ chính xác, độ nhạy, độ đặc hiệu, và ma trận nhầm lẫn trên các tập dữ liệu kiểm tra. Kiểm định mô hình với các bộ dữ liệu âm thanh khác nhau để đảm bảo mô hình hoạt động ổn định và hiệu quả trong các tình huống thực tế.

Phân tích kết quả: Phân tích kết quả thu được từ các thử nghiệm và đánh giá để rút ra các kết luận và khuyến nghị.

# CƠ SỞ LÝ THUYẾT

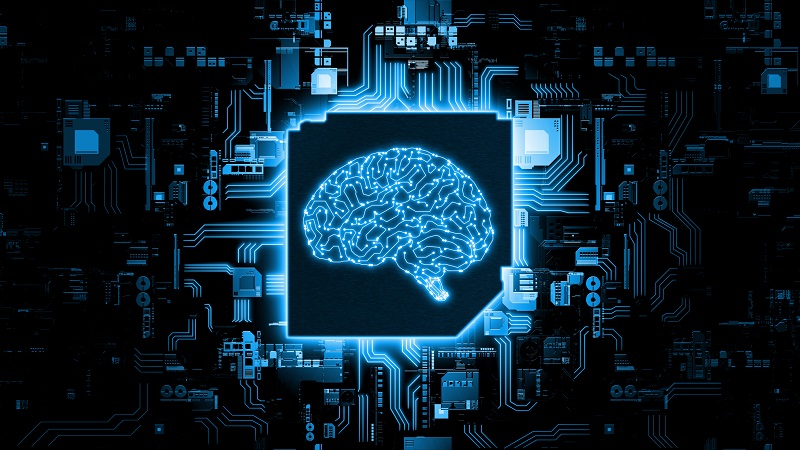
## Tìm Hiểu Về Máy Học

### Máy học là gì ?

Máy học (machine learning) là một lĩnh vực của trí tuệ nhân tạo liên quan đến việc nghiên cứu và xây dựng các kỹ thuật cho phép các hệ thống “học” tự động từ dữ liệu để giải quyết những vấn đề cụ thể. Theo định nghĩa trong cuốn sách "Machine Learning" của Tom Mitchell (1997), máy học được mô tả như sau:

“A computer program is said to learn to perform a task T from experience E, if its performance at task T, as measured by a performance metric P, improves

with experience E over time.”



Học máy (Machine Learning)

Máy học là một thành phần quan trọng của lĩnh vực khoa học dữ liệu đang phát triển. Thông qua việc sử dụng phương pháp thống kê, các thuật toán được đào tạo để phân loại hoặc dự đoán và khám phá những thông tin chi tiết trong các dự án khai thác dữ liệu.

Những thông tin chi tiết này hỗ trợ, thúc đẩy việc đưa ra quyết định trong các ứng dụng, công cụ hỗ trợ doanh nghiệp, người dùng. Khi khối lượng dữ liệu tiếp tục mở rộng và phát triển, khả năng dự đoán, phân tích chính xác của máy học sẽ tăng lên.

### Lịch Sử Hình Thành Máy Học

Lịch sử hình thành và phát triển của máy học (machine learning) có thể được chia thành nhiều giai đoạn khác nhau:

1950s - 1960s: Khởi đầu

**1950**: Alan Turing đề xuất "Turing Test" trong bài viết "Computing Machinery and Intelligence," đặt nền móng cho việc nghiên cứu trí tuệ nhân tạo (AI).

**1952**: Arthur Samuel phát triển chương trình máy tính có khả năng chơi cờ vua và tự học từ kinh nghiệm.

**1957**: Frank Rosenblatt phát triển Perceptron, mô hình mạng neuron đơn giản, đánh dấu bước tiến đầu tiên trong nghiên cứu mạng neuron.

1970s - 1980s: Thời kỳ khó khan

**1970s**: AI gặp phải "Mùa đông AI" khi những kỳ vọng quá cao không được đáp ứng, dẫn đến giảm tài trợ và nghiên cứu.

**1980**: John Hopfield và David Rumelhart tái phát triển mạng neuron bằng cách giới thiệu mô hình Hopfield và thuật toán học ngược.

1990s: Thời kỳ phục hung

**1990s**: Máy học bắt đầu phát triển mạnh mẽ trở lại với sự gia tăng của dữ liệu và sức mạnh tính toán.

**1995**: Vladimir Vapnik và Alexey Chervonenkis phát triển Support Vector Machines (SVM), một phương pháp phân loại mạnh mẽ.

2000s: Phát triển nhanh chóng

**2006**: Geoffrey Hinton và các cộng sự giới thiệu thuật toán Deep Belief Networks, đặt nền móng cho sự phát triển của deep learning.

**2010**: Sự gia tăng của dữ liệu lớn (big data) và sự phát triển của GPU giúp deep learning và các phương pháp máy học khác phát triển mạnh mẽ.

2010s - Hiện tại: Kỷ nguyên AI, phát triển và ứng dụng

**2012**: AlexNet, một mô hình deep learning, chiến thắng cuộc thi ImageNet, chứng minh sức mạnh của mạng neuron sâu.

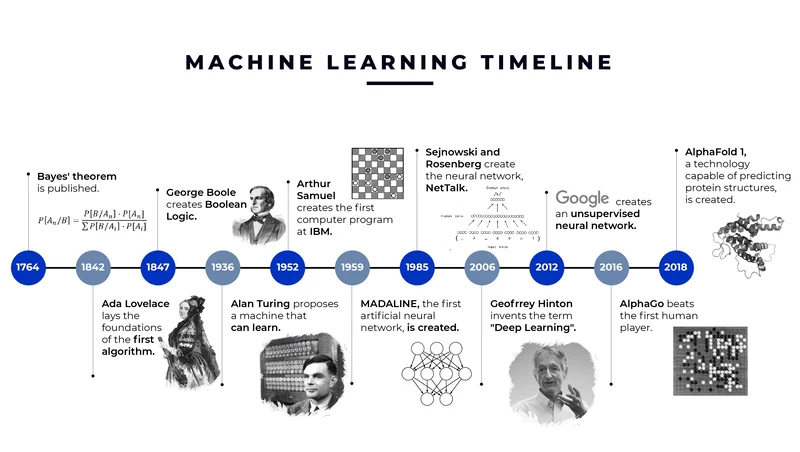
**2014**: Google DeepMind giới thiệu AlphaGo, một hệ thống AI có khả năng chơi cờ vây vượt qua người chơi chuyên nghiệp.

**2016**: AI được ứng dụng rộng rãi trong nhiều lĩnh vực như y tế, tài chính, và ô tô tự lái.

2017: Sự ra đời của Transformer (trong bài báo "Attention is All You Need" của Vaswani et al.), một mô hình mạng neuron mới đã cách mạng hóa nhiều ứng dụng trong xử lý ngôn ngữ tự nhiên (NLP).

2018: GPT-2 và sau đó là GPT-3 của OpenAI, mô hình ngôn ngữ lớn có khả năng tạo ra văn bản tự nhiên với độ phức tạp cao, đã gây ấn tượng mạnh.

2019: BERT (Bidirectional Encoder Representations from Transformers) của Google, một mô hình NLP mạnh mẽ, đã cải thiện nhiều nhiệm vụ xử lý ngôn ngữ tự nhiên.

2020: Sự phát triển của AI tự học không giám sát và học tăng cường, như AlphaFold của DeepMind, giải quyết vấn đề gấp cuộn protein, một trong những thách thức lớn nhất trong sinh học.

Quá trình phát triển của máy học

### Phân Loại Máy Học

- Học có giám sát (Supervised Learning):

Học có giám sát là một trong những phương pháp phổ biến nhất trong máy học, nơi mà mô hình học từ một tập dữ liệu đã được gán nhãn. Tập dữ liệu huấn luyện gồm các cặp đầu vào (input) và đầu ra (output) mong muốn. Mục tiêu của mô hình là học cách ánh xạ từ đầu vào đến đầu ra để có thể dự đoán chính xác đầu ra cho các dữ liệu mới chưa thấy trước đó. Trong quá trình học, mô hình điều chỉnh các thông số của nó dựa trên sự khác biệt giữa dự đoán và thực tế, nhằm giảm thiểu lỗi dự đoán.

- Học không giám sát (Unsupervised Learning):

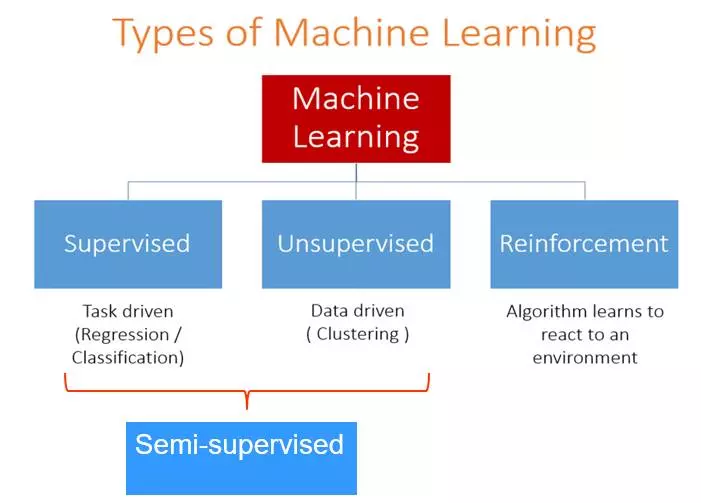
Học không giám sát là một kỹ thuật trong máy học mà không có nhãn đầu ra đi kèm với dữ liệu đầu vào. Thay vào đó, mục tiêu của mô hình là khám phá các cấu trúc, mẫu hoặc mối quan hệ ẩn trong dữ liệu. Điều này giúp hiểu rõ hơn về dữ liệu và có thể được sử dụng để phát hiện các nhóm (clusters), giảm chiều dữ liệu hoặc phát hiện các ngoại lệ (anomalies).

- Học tăng cường (Reinforcement Learning):

Học tăng cường là một kỹ thuật trong máy học mà một tác nhân (agent) học cách tương tác với môi trường để tối đa hóa phần thưởng tích lũy theo thời gian. Môi trường đưa ra phản hồi dưới dạng phần thưởng hoặc hình phạt cho các hành động của tác nhân. Tác nhân sử dụng phản hồi này để điều chỉnh chính sách hành động của mình sao cho đạt được phần thưởng tối đa.

- Học bán giám sát (Semi-supervised Learning):

Học bán giám sát là một phương pháp kết hợp giữa học có giám sát và học không giám sát. Mô hình được đào tạo trên một tập dữ liệu có cả nhãn và không nhãn. Điều này rất hữu ích khi việc gán nhãn dữ liệu tốn kém hoặc khó khăn, nhưng có nhiều dữ liệu không nhãn sẵn có. Học bán giám sát sử dụng dữ liệu không nhãn để cải thiện hiệu suất của mô hình hơn so với chỉ sử dụng dữ liệu có nhãn.



Phân loại trong máy học

### Cách hoạt động của thuật toán trong máy học:

Bước 1: Thu thập dữ liệu (Gathering Data / Data Collection)

Đây là bước đầu tiên và quan trọng nhất trong quy trình học máy. Trong bước này, bạn cần thu thập dữ liệu từ các nguồn khác nhau mà mô hình của bạn sẽ được huấn luyện. Dữ liệu có thể được thu thập từ nhiều nguồn như cơ sở dữ liệu, cảm biến, dịch vụ web, khảo sát, hoặc thậm chí từ các tệp tin đã có sẵn. Mục tiêu của bước này là đảm bảo bạn có đủ dữ liệu chất lượng để huấn luyện mô hình.

Bước 2: Tiền xử lý dữ liệu (Data Preprocessing)

Tiền xử lý dữ liệu là một bước quan trọng để chuẩn bị dữ liệu cho mô hình học máy. Quá trình này bao gồm nhiều công việc khác nhau:

**Trích xuất dữ liệu (Data Extraction):** Lấy dữ liệu từ các nguồn thu thập và đưa vào định dạng dễ xử lý.

**Làm sạch dữ liệu (Data Cleaning):** Xóa bỏ hoặc sửa chữa các lỗi trong dữ liệu, chẳng hạn như dữ liệu bị thiếu, dữ liệu không chính xác, hoặc dữ liệu bị trùng lặp.

**Chuyển đổi dữ liệu (Data Transformation):** Chuyển đổi dữ liệu từ định dạng này sang định dạng khác để phù hợp với yêu cầu của mô hình, ví dụ như chuyển đổi dữ liệu văn bản thành số hoặc thay đổi định dạng thời gian.

**Chuẩn hóa dữ liệu (Data Normalization):** Đưa các thuộc tính dữ liệu về cùng một phạm vi giá trị để giúp mô hình học dễ dàng hơn. Ví dụ, chuẩn hóa các giá trị về khoảng từ 0 đến 1.

**Trích xuất đặc trưng (Feature Extraction):** Xác định và chọn lựa các đặc trưng (features) quan trọng từ dữ liệu gốc mà mô hình sẽ sử dụng. Đây là bước quan trọng để cải thiện hiệu suất của mô hình bằng cách loại bỏ các đặc trưng không cần thiết hoặc không liên quan.

Bước 3: Phân tích dữ liệu (Data Analysis)

Trong bước này, bạn thực hiện các phân tích sơ bộ để hiểu rõ hơn về dữ liệu của mình. Điều này có thể bao gồm việc khám phá các mối quan hệ giữa các biến, phân phối của dữ liệu, và các mẫu hoặc xu hướng có thể có trong dữ liệu. Phân tích dữ liệu giúp xác định các vấn đề tiềm ẩn và hướng dẫn việc chọn lựa các thuật toán và kỹ thuật phù hợp cho việc xây dựng mô hình.

Bước 4: Xây dựng mô hình máy học (Model Building)

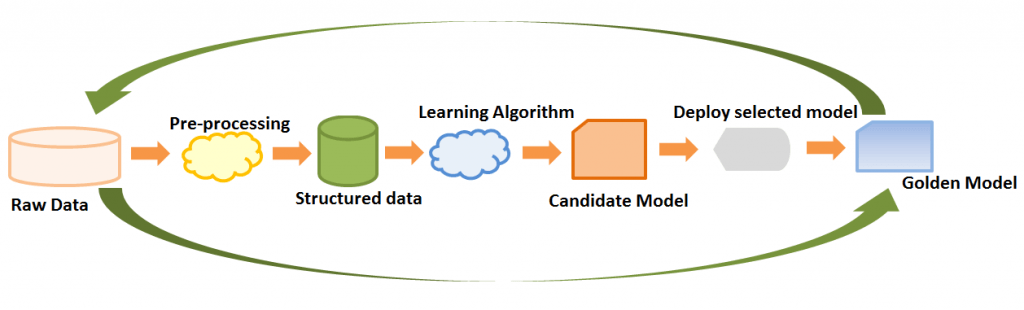
Sau khi dữ liệu đã được chuẩn bị và phân tích, bạn tiến hành xây dựng mô hình học máy. Trong bước này, bạn chọn lựa và thiết kế các thuật toán học máy phù hợp với bài toán của mình. Có nhiều loại mô hình khác nhau như hồi quy tuyến tính, cây quyết định, mạng nơ-ron, và máy vector hỗ trợ (SVM). Mô hình được xây dựng dựa trên các đặc trưng đã chọn và cấu hình các tham số của mô hình.

Bước 5: Huấn luyện mô hình (Model Training)

Huấn luyện mô hình là bước quan trọng để mô hình học máy học từ dữ liệu. Trong bước này, dữ liệu huấn luyện được sử dụng để điều chỉnh các tham số của mô hình sao cho mô hình có thể dự đoán chính xác các kết quả. Bạn sẽ chia dữ liệu thành hai tập: tập huấn luyện và tập kiểm tra. Tập huấn luyện được dùng để cập nhật các tham số của mô hình, trong khi tập kiểm tra dùng để đánh giá hiệu suất của mô hình trong quá trình huấn luyện.

Bước 6: Đánh giá mô hình (Model Evaluation)

Sau khi mô hình đã được huấn luyện, bạn cần đánh giá hiệu suất của nó để đảm bảo rằng nó hoạt động tốt và có thể dự đoán chính xác trên dữ liệu mới. Đánh giá mô hình thường được thực hiện bằng cách sử dụng các chỉ số hiệu suất như độ chính xác, độ nhạy, độ đặc hiệu. Bạn cũng có thể sử dụng các kỹ thuật như kiểm tra chéo để đảm bảo rằng mô hình không bị overfit với dữ liệu huấn luyện và có thể hoạt động tốt với dữ liệu chưa thấy.



Quá trình hoạt động của máy học

## Tìm Hiểu Về Deep Learning

### Tổng quan về Deep Learning

Học sâu (Deep Learning) là một lĩnh vực con của học máy và trí tuệ nhân tạo, tập trung vào việc sử dụng các mạng neuron nhiều lớp để học và trích xuất các đặc trưng từ dữ liệu phức tạp. Trong học sâu, các mô hình mạng neuron bao gồm nhiều lớp ẩn giữa lớp đầu vào và lớp đầu ra, cho phép hệ thống học và hiểu các biểu diễn ngày càng trừu tượng của dữ liệu. Điều này giúp mô hình có khả năng xử lý các nhiệm vụ phức tạp mà các phương pháp học máy truyền thống có thể gặp khó khăn.

Mạng nơ-ron hoạt động bằng cách điều chỉnh trọng số và hệ số bias thông qua quá trình huấn luyện để giảm thiểu sự khác biệt giữa dự đoán của mô hình và giá trị thực tế, thông qua các thuật toán tối ưu hóa như Gradient Descent. Học sâu đã đạt được những thành tựu đáng kể trong nhiều lĩnh vực, từ nhận dạng hình ảnh, phân tích và nhận diện văn bản, đến nhận diện giọng nói và tự động lái xe.

Các mô hình học sâu, như mạng nơ-ron tích chập (CNNs), mạng nơ-ron hồi tiếp (RNNs) và các mạng Transformer, đều có những ứng dụng và ưu điểm riêng. CNNs rất hiệu quả trong nhận dạng và phân loại hình ảnh nhờ khả năng học các đặc trưng không gian. RNNs và các biến thể của nó, như LSTM và GRU, được sử dụng để xử lý dữ liệu chuỗi thời gian và ngôn ngữ tự nhiên. Mạng Transformer, với cơ chế chú ý đã chứng minh sự vượt trội trong xử lý ngôn ngữ và dịch máy.

Dù học sâu mang lại nhiều lợi ích, bao gồm khả năng tự động trích xuất đặc trưng và xử lý dữ liệu phức tạp, nó cũng đối mặt với những thách thức đáng kể. Việc huấn luyện các mô hình học sâu yêu cầu một lượng lớn dữ liệu và tài nguyên tính toán mạnh mẽ, như GPU hoặc TPU. Thêm vào đó, các mô hình học sâu thường bị coi là "hộp đen", tức là khó giải thích các quyết định và dự đoán mà chúng đưa ra, làm tăng mức độ khó khăn trong việc hiểu và tin tưởng vào kết quả của mô hình.

### Cách hoạt động của Deep Learning:

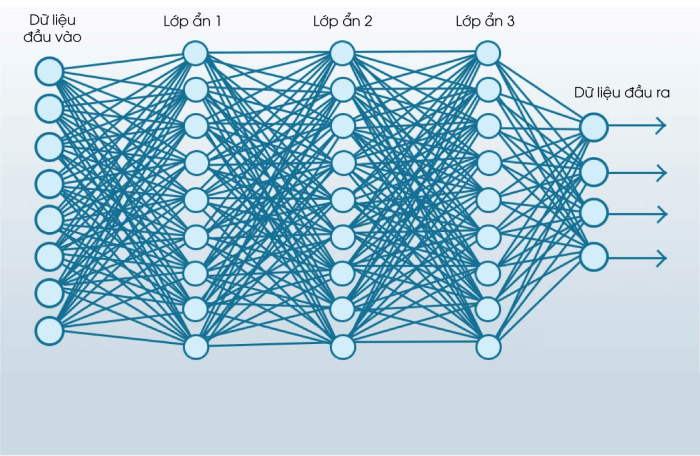
Mạng nơ-ron (Neural network) hoặc mạng nơ-ron nhân tạo (Artificial neural network) cố gắng bắt chước bộ não con người thông qua sự kết hợp của dữ liệu đầu vào, trọng số và độ lệch. Các phần tử này phối hợp với nhau để nhận dạng, phân loại và mô tả chính xác các đối tượng trong dữ liệu.

Mạng nơ-ron bao gồm nhiều lớp node được kết nối với nhau, mỗi lớp được xây dựng dựa trên lớp trước đó để tinh chỉnh và tối ưu hóa dự đoán hoặc phân loại. Tiến trình tính toán này thông qua mạng được gọi là truyền xuôi (forward propagation).

Các lớp đầu vào và đầu ra của mạng nơ-ron được gọi là các lớp hiển thị. Lớp đầu vào là nơi mô hình deep learning nhập dữ liệu để xử lý và lớp đầu ra là nơi đưa ra dự đoán hoặc phân loại cuối cùng.

Một quy trình khác gọi là truyền ngược (back propagation) sử dụng các thuật toán như giảm độ dốc, để tính toán các lỗi trong dự đoán, sau đó điều chỉnh trọng số và độ lệch của hàm bằng cách di chuyển ngược qua các lớp trong mô hình.

Truyền ngược và truyền xuôi cho phép mạng nơ-ron đưa ra dự đoán và sửa lỗi ngay lập tức. Theo thời gian, thuật toán dần trở nên chính xác hơn.



Quá trình hoạt động của học sâu

## Khái Niệm Về CNN

### Tổng quan về CNN (Convolutional Neural Networks)

Khái niệm:

Mạng nơ-ron tích chập (Convolutional Neural Network - CNN) là một loại mạng nơ-ron sâu, được thiết kế đặc biệt để xử lý dữ liệu có cấu trúc dạng lưới, chẳng hạn như hình ảnh và âm thanh. CNN nổi bật bởi khả năng tự động trích xuất và học các đặc trưng từ dữ liệu đầu vào thông qua các lớp tích chập (convolutional layers) và lớp gộp (pooling layers).

Cấu trúc của CNN

Lớp Tích Chập (Convolutional Layer): Sử dụng các bộ lọc (filters) để quét qua dữ liệu đầu vào, tạo ra các bản đồ đặc trưng (feature maps). Mỗi bộ lọc học cách nhận diện một đặc trưng cụ thể từ dữ liệu, như cạnh, góc, hoặc kết cấu.

Lớp Gộp (Pooling Layer): Giảm kích thước của các bản đồ đặc trưng bằng cách lấy giá trị cực đại (max pooling) hoặc giá trị trung bình (average pooling) từ các vùng nhỏ. Lớp này giúp giảm số lượng tham số và làm mô hình ít nhạy cảm với các thay đổi nhỏ trong dữ liệu đầu vào.

Lớp Kết Nối Đầy Đủ (Fully Connected Layer): Các nơ-ron trong lớp này kết nối với tất cả các nơ-ron trong lớp trước đó, giúp thực hiện các tác vụ phân loại dựa trên các đặc trưng đã trích xuất.

Lý do sử dụng CNN cho phân loại âm thanh:

Khả năng tự động trích xuất đặc trưng: CNN có khả năng tự động trích xuất các đặc trưng quan trọng từ dữ liệu âm thanh mà không cần các bước tiền xử lý phức tạp. Điều này giúp giảm thiểu công sức và thời gian so với các phương pháp truyền thống đòi hỏi phải thiết kế và lựa chọn đặc trưng thủ công.

Hiệu suất vượt trội trong nhiệm vụ phân loại: CNN đã chứng minh hiệu suất vượt trội trong các nhiệm vụ phân loại dữ liệu phức tạp như hình ảnh và âm thanh. Các mô hình CNN có thể học và phân biệt các mẫu âm thanh phức tạp với độ chính xác cao, phù hợp với yêu cầu của đề tài.

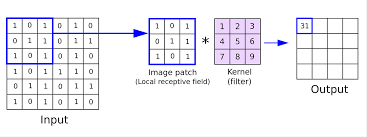
Khả năng học tập phân cấp: CNN học các đặc trưng theo cách phân cấp, từ các đặc trưng đơn giản ở các lớp thấp đến các đặc trưng phức tạp ở các lớp cao. Điều này rất hữu ích trong việc nhận diện các đặc trưng âm thanh phức tạp và trừu tượng.

Khả năng xử lý dữ liệu đa dạng: CNN có thể xử lý hiệu quả nhiều loại dữ liệu âm thanh khác nhau, từ tiếng nói, tiếng nhạc, đến tiếng động môi trường. Điều này giúp mở rộng phạm vi ứng dụng của mô hình trong thực tế.

### Cách hoạt động của CNN

Quá trình hoạt động của CNN

Convolution Layer: Convolution (lớp tích chập) là lớp đầu tiên trích xuất các đặc tính từ hình ảnh. Tham số lớp này bao gồm một tập hợp các bộ lọc có thể học được. Các bộ lọc đều nhỏ thường có kích cỡ hai chiều đầu tiên khoảng 3x3 hoặc 5x5, .... và có độ sâu bằng với độ sâu của đầu vào đầu vào. Bằng cách trượt dần bộ lọc theo chiều ngang và dọc trên ảnh, chúng thu được một Feature Map chứa các đặc trưng được trích xuất từ trên hình ảnh đầu vào. Quá trình trượt các bộ lọc thường có các giá trị được quy định bao gồm:

* padding: quy định bộ đệm của bộ lọc hay chính là phần màu xám được thêm vào ảnh
* stride: quy định bước nhảy trong quá trình thực hiện.

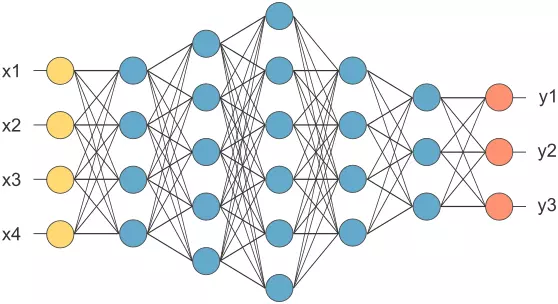
Lớp tích chập

Pooling layer: Pooling layer thường được dùng giữa các convolutional layer, để giảm kích thước dữ liệu nhưng vẫn giữ được các thuộc tính quan trọng. Kích thước dữ liệu giảm giúp giảm việc tính toán trong model. Trong quá trình này, quy tắc về stride và padding áp dụng như phép tính convolution trên ảnh.

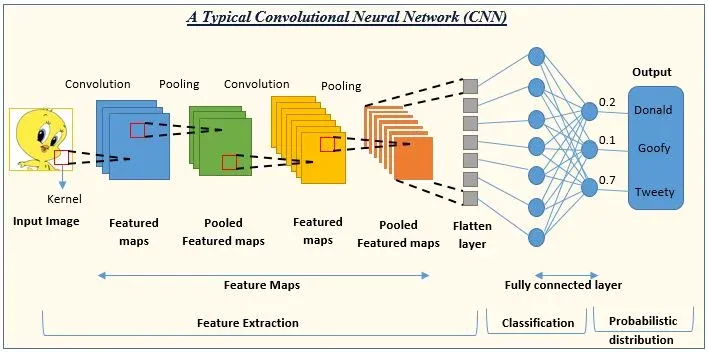


Lớp gộp

Fully connected layer: Sau khi ảnh được truyền qua nhiều convolutional layer và pooling layer thì model đã học được tương đối các đặc điểm của ảnh thì tensor của output của layer cuối cùng sẽ được là phẳng thành vector và đưa vào một lớp được kết nối như một mạng nơ-ron. Với FC layer được kết hợp với các tính năng lại với nhau để tạo ra một mô hình. Cuối cùng sử dụng softmax hoặc sigmoid để phân loại đầu ra.



Lớp kết nối đầy đủ

Cách hoạt động

Mô hình CNN hoạt động

Dữ Liệu Đầu Vào (Input Image): Hình ảnh ban đầu được đưa vào mạng CNN. Hình ảnh đầu vào là hình của một con vật hoạt hình (Tweety).

Lớp Tích Chập Đầu Tiên (First Convolutional Layer):

* Kernel: Bộ lọc (kernel) quét qua hình ảnh đầu vào để trích xuất các đặc trưng. Kernel thực hiện tích chập bằng cách di chuyển qua các vùng nhỏ của hình ảnh và tính toán tích vô hướng giữa kernel và vùng đó.
* Bản đồ đặc trưng (Feature Maps): Kết quả của tích chập là các bản đồ đặc trưng, thể hiện các đặc trưng được trích xuất từ hình ảnh đầu vào.

Lớp Gộp Đầu Tiên (First Pooling Layer): Quá trình gộp (pooling) giảm kích thước các bản đồ đặc trưng bằng cách lấy giá trị lớn nhất (max pooling) hoặc giá trị trung bình (average pooling) từ các vùng nhỏ không chồng lấn của bản đồ đặc trưng. Điều này giúp giảm số lượng tham số và làm mô hình ít nhạy cảm hơn với các biến đổi nhỏ trong dữ liệu đầu vào.

Lớp Tích Chập Thứ Hai (Second Convolutional Layer): Các bản đồ đặc trưng từ lớp gộp đầu tiên tiếp tục được đưa qua lớp tích chập thứ hai, sử dụng các kernel khác nhau để trích xuất các đặc trưng phức tạp hơn.

Lớp Gộp Thứ Hai (Second Pooling Layer): Quá trình gộp tương tự được áp dụng để giảm kích thước các bản đồ đặc trưng từ lớp tích chập thứ hai, tạo ra các bản đồ đặc trưng đã gộp.

Lớp Phẳng (Flatten Layer): Các bản đồ đặc trưng cuối cùng từ lớp gộp thứ hai được chuyển đổi thành một vector phẳng. Quá trình này giúp chuẩn bị dữ liệu cho lớp fully connected.

Lớp Kết Nối Đầy Đủ (Fully Connected Layer):

* **Kết nối đầy đủ:** Vector phẳng từ lớp flatten được kết nối với tất cả các nơ-ron trong lớp fully connected. Lớp này hoạt động như một mạng nơ-ron truyền thống (MLP), với các kết nối đầy đủ giữa các nơ-ron.
* **Phân loại (Classification):** Lớp fully connected thực hiện nhiệm vụ phân loại, dựa trên các đặc trưng đã học từ các lớp trước đó.
* **Lớp Đầu Ra (Output Layer):** Lớp đầu ra cung cấp các xác suất cho từng lớp trong bài toán phân loại. Trong hình, xác suất đầu ra cho các lớp là "Donald," "Goofy," và "Tweety," với xác suất cao nhất thuộc về lớp "Tweety."

### Kiến trúc mạng CNN

Mạng CNN là tập hợp những Convolutional layer xếp chồng lên nhau, đồng thời mạng sử dụng những hàm như ReLU và Tanh để kích hoạt các trọng số trong các node. Các lớp này sau khi qua các hàm activation sẽ có trọng số trong những node và có thể tạo ra những thông tin trừu tượng hơn đến với các lớp kế tiếp trong mạng.

Mạng này có tính kết hợp cà tính bất biến. Tức là, nếu cùng một đối tượng mà sử dụng chiếu theo các góc độ khác nhau thì sẽ có ảnh hưởng đến độ chính xác. Với dịch chuyển, co giãn hay quay ma trận ảnh thì lớp Pooling sẽ được dùng để hỗ trợ làm bất biến các tính chất này. Chính vì vậy mà mạng nơ ron này sẽ đưa ra những kết quả có độ chính xác tương ứng với từng mô hình.

Trong đó, lớp Pooling sẽ có khả năng tạo tính bất biến với phép dịch chuyển, co giãn và quay. Còn tính kết hợp cục bộ sẽ cho thấy những cấp độ biểu diễn, dữ liệu từ thấp đến cao với mức trừu tượng thông qua Convolution từ filter. Mạng CNN có những lớp liên kết nhau dựa vào cơ chế Convolution.

Các lớp tiếp theo sẽ là kết quả từ những lớp trước đó, vì vậy mà bạn sẽ có những liên kết cục bộ phù hợp nhất. Trong quá trình huấn luyện mạng, mạng nơ ron này sẽ tự học hỏi những giá trị thông qua filter layer dựa theo cách thức mà bạn thực hiện.

Cấu trúc cơ bản của một mô hình mạng CNN thường bao gồm 3 phần chính bao gồm:

* **Trường cục bộ/ Local receptive field:** Lớp này sử dụng để tách lọc dữ liệu, thông tin hình ảnh để từ đó có thể lựa chọn các vùng có giá trị sử dụng hiệu quả cao nhất.
* **Trọng số chia sẻ/ Shared weights and bias:** Lớp này hỗ trợ làm giảm các tham số đến mức tối thiểu trong mạng CNN. Trong từng lớp convolution sẽ chứa các feature map riêng và từng feature thì sẽ có khả năng phát hiện một vài feature trong hình ảnh.
* **Lớp tổng hợp/ Pooling layer:** Đây là lớp cuối cùng và sử dụng để làm đơn giản các thông tin output. Tức là, sau khi tính toán xong và quét qua các layer trong mạng thì pooling layer sẽ được dùng để lược bỏ các thông tin không hữu ích. Từ đó cho ra kết quả theo kỳ vọng người dùng.

## Tìm Hiểu Về PyCharm

### Khái niệm PyCharm:

PyCharm là một môi trường phát triển tích hợp (IDE) mạnh mẽ dành riêng cho ngôn ngữ lập trình Python, được phát triển bởi JetBrains. PyCharm cung cấp nhiều tính năng tiên tiến giúp tối ưu hóa quá trình phát triển phần mềm, đặc biệt là cho các dự án liên quan đến học máy (machine learning) và khoa học dữ liệu (data science).

### Lý do chọn PyCharm cho đề tài:

Hỗ trợ mạnh mẽ cho Python: PyCharm được thiết kế đặc biệt cho Python, với các tính năng như kiểm tra cú pháp, tự động hoàn thành mã, và gợi ý mã nguồn giúp việc viết mã trở nên dễ dàng và hiệu quả.

Hỗ trợ tích hợp cho khoa học dữ liệu: PyCharm cung cấp hỗ trợ tích hợp cho các công cụ và thư viện phổ biến trong khoa học dữ liệu như NumPy, Pandas, Matplotlib, và Jupyter Notebook, giúp sinh viên dễ dàng thực hiện các phân tích dữ liệu và thử nghiệm mô hình.

Quản lý môi trường ảo: PyCharm tích hợp các công cụ để tạo và quản lý môi trường ảo, giúp tách biệt các dự án và quản lý các gói thư viện dễ dàng.

Công cụ gỡ lỗi mạnh mẽ: PyCharm cung cấp công cụ gỡ lỗi tiên tiến, cho phép sinh viên dễ dàng theo dõi và sửa lỗi trong mã nguồn của họ.

Tích hợp với các hệ thống kiểm soát phiên bản: PyCharm hỗ trợ tích hợp với các hệ thống kiểm soát phiên bản như Git, Mercurial, và SVN, giúp quản lý phiên bản mã nguồn hiệu quả.

### Các tính năng PyCharm hữu ích cho đề tài:

Hỗ trợ lập trình Python:

* Kiểm tra cú pháp và tự động hoàn thành mã: PyCharm cung cấp kiểm tra cú pháp tức thì và gợi ý mã nguồn thông minh, giúp sinh viên viết mã nhanh chóng và chính xác.
* Gợi ý mã nguồn: PyCharm cung cấp gợi ý mã nguồn dựa trên ngữ cảnh, giúp tăng tốc quá trình viết mã.

Hỗ trợ khoa học dữ liệu:

* Jupyter Notebook: PyCharm hỗ trợ tích hợp Jupyter Notebook, cho phép sinh viên viết và chạy các cell mã lệnh, thực hiện phân tích dữ liệu và huấn luyện mô hình CNN trực tiếp trong IDE.
* Biểu đồ và trực quan hóa dữ liệu: PyCharm hỗ trợ trực quan hóa dữ liệu với Matplotlib và các thư viện biểu đồ khác, giúp sinh viên dễ dàng phân tích và hiểu dữ liệu.

Quản lý môi trường ảo:

* Tạo và quản lý môi trường ảo: PyCharm cho phép tạo và quản lý môi trường ảo trực tiếp từ IDE, giúp tách biệt các dự án và quản lý các gói thư viện dễ dàng.
* Tích hợp với Conda: PyCharm hỗ trợ tích hợp với Conda, cho phép quản lý các môi trường và gói thư viện Conda một cách dễ dàng.

Công cụ gỡ lỗi:

* Gỡ lỗi trực quan: PyCharm cung cấp công cụ gỡ lỗi trực quan với các tính năng như điểm dừng (breakpoints), theo dõi biến (variable tracking), và kiểm tra dòng mã (step-by-step execution), giúp sinh viên dễ dàng phát hiện và sửa lỗi.
* Gỡ lỗi từ xa: PyCharm hỗ trợ gỡ lỗi từ xa, cho phép sinh viên gỡ lỗi mã nguồn trên các máy chủ hoặc thiết bị từ xa.

Tích hợp với các hệ thống kiểm soát phiên bản:

* Quản lý phiên bản mã nguồn: PyCharm tích hợp sẵn các công cụ quản lý phiên bản mã nguồn như Git, Mercurial, và SVN, giúp sinh viên quản lý phiên bản mã nguồn một cách hiệu quả.
* Hợp tác làm việc nhóm: PyCharm cung cấp các công cụ hỗ trợ hợp tác làm việc nhóm, giúp sinh viên làm việc cùng nhau một cách hiệu quả.

Hỗ trợ Docker và các công cụ triển khai khác:

* Docker: PyCharm hỗ trợ tích hợp với Docker, cho phép sinh viên tạo và quản lý các container Docker trực tiếp từ IDE.
* Hỗ trợ triển khai: PyCharm cung cấp các công cụ hỗ trợ triển khai ứng dụng, giúp sinh viên dễ dàng triển khai mô hình CNN vào môi trường thực tế.

### Các bước thực hiện với PyCharm

Cài đặt và thiết lập môi trường:

* Cài đặt PyCharm và các tiện ích mở rộng cần thiết như Python, Jupyter, TensorFlow, Keras, và PyTorch.
* Tạo và kích hoạt môi trường ảo Python, cài đặt các gói thư viện cần thiết.

Chuẩn bị dữ liệu:

* Sử dụng PyCharm để viết mã xử lý dữ liệu âm thanh, bao gồm tiền xử lý dữ liệu và chia dữ liệu thành các tập huấn luyện và kiểm tra.
* Sử dụng Jupyter Notebook trong PyCharm để trực quan hóa và phân tích dữ liệu.

Phát triển và huấn luyện mô hình CNN:

* Sử dụng PyCharm để viết mã phát triển mô hình CNN, bao gồm việc xây dựng kiến trúc mạng, định nghĩa các hàm mất mát và tối ưu hóa.
* Sử dụng Jupyter Notebook để huấn luyện mô hình và theo dõi các chỉ số hiệu suất.

Đánh giá và tinh chỉnh mô hình:

* Đánh giá hiệu suất mô hình trên tập kiểm tra, sử dụng các chỉ số như độ chính xác, độ nhạy, độ đặc hiệu.
* Tinh chỉnh mô hình bằng cách điều chỉnh các siêu tham số và thử nghiệm các kiến trúc mạng khác nhau.

Triển khai và trình bày kết quả:

* Triển khai mô hình đã huấn luyện vào một ứng dụng thực tế nếu cần thiết.
* Sử dụng PyCharm để viết tài liệu và báo cáo kết quả, bao gồm việc mô tả quá trình thực hiện và các kết quả đạt được.

## Ứng Dụng Các Cơ Sở Lý Thuyết Vào Đề Tài:

Trong đề tài phân loại âm thanh, các lý thuyết về máy học, CNN (Convolutional Neural Networks), và học sâu (Deep Learning) đóng vai trò then chốt, không chỉ trong việc xây dựng mà còn trong quá trình tối ưu hóa mô hình phân loại. Những lý thuyết này cung cấp các công cụ và phương pháp tiếp cận cần thiết để phát triển các hệ thống phân loại âm thanh hiện đại, mạnh mẽ và chính xác, giúp đề tài đạt được mục tiêu nghiên cứu một cách hiệu quả.

**Máy học (Machine Learning)** là nền tảng cơ bản của đề tài, cung cấp các thuật toán học từ dữ liệu để mô hình hóa các mối quan hệ phức tạp giữa các mẫu âm thanh và các nhãn phân loại tương ứng. Trong bối cảnh đề tài này, các thuật toán máy học được phát triển và áp dụng để tự động phân loại các mẫu âm thanh thành các nhóm hoặc danh mục khác nhau dựa trên các đặc trưng mà chúng học được từ dữ liệu. Những thuật toán này không chỉ giúp nhận diện các mẫu âm thanh thông thường mà còn có khả năng xử lý các biến động trong dữ liệu, chẳng hạn như nhiễu âm hoặc biến đổi trong tín hiệu âm thanh. Máy học cho phép mô hình trở nên linh hoạt và có khả năng tổng quát hóa cao, từ đó đạt được hiệu suất phân loại tốt ngay cả khi dữ liệu huấn luyện không hoàn toàn tương đồng với dữ liệu thực tế.

**Mạng nơ-ron tích chập (CNN)** là một kiến trúc quan trọng trong học sâu, được áp dụng để trích xuất các đặc trưng quan trọng từ các biểu diễn hình ảnh của âm thanh, chẳng hạn như Mel-spectrogram hoặc MFCC (Mel Frequency Cepstral Coefficients). CNN hoạt động bằng cách sử dụng các lớp tích chập để tự động học các mẫu trong dữ liệu đầu vào, từ các đặc trưng cơ bản như các cạnh, họa tiết, đến các mẫu phức tạp hơn liên quan đến bản chất của âm thanh. Điều này giúp CNN có khả năng nhận diện các đặc trưng độc đáo trong âm thanh một cách hiệu quả và phân loại chính xác các mẫu âm thanh, ngay cả khi dữ liệu bị nhiễu hoặc có sự biến đổi lớn. CNN không chỉ mạnh mẽ trong việc xử lý các dữ liệu không gian như hình ảnh mà còn đặc biệt hiệu quả trong việc phân tích dữ liệu âm thanh sau khi nó được biểu diễn dưới dạng hình ảnh.

**Học sâu (Deep Learning)**, với việc sử dụng các mạng nơ-ron sâu nhiều lớp, cung cấp một phương pháp tiên tiến để mô hình hóa các mối quan hệ phức tạp trong dữ liệu âm thanh. Trong học sâu, mỗi lớp của mạng nơ-ron có thể học các đặc trưng ở mức độ trừu tượng khác nhau, từ các đặc trưng đơn giản đến các đặc trưng phức tạp, giúp mô hình hiểu sâu hơn về bản chất của dữ liệu. Điều này rất quan trọng trong việc tối ưu hóa quá trình huấn luyện, giúp mô hình không chỉ đạt được độ chính xác cao mà còn cải thiện khả năng tổng quát hóa khi gặp phải dữ liệu mới. Học sâu cũng hỗ trợ việc học không giám sát (unsupervised learning) và bán giám sát (semi-supervised learning), giúp mô hình có thể học từ các tập dữ liệu lớn mà không cần nhãn hoặc chỉ cần một số lượng nhãn nhỏ, qua đó tiết kiệm thời gian và công sức trong việc thu thập và gán nhãn dữ liệu.

**PyCharm**, với tư cách là một môi trường phát triển tích hợp (IDE) mạnh mẽ, đóng vai trò quan trọng trong việc hỗ trợ triển khai các mô hình học sâu này. PyCharm cung cấp một môi trường phát triển lý tưởng với nhiều công cụ và tính năng hỗ trợ như kiểm tra cú pháp, tự động hoàn thành mã, và gợi ý mã, giúp tối ưu hóa quá trình viết mã và giảm thiểu lỗi cú pháp. Đặc biệt, PyCharm còn hỗ trợ mạnh mẽ trong việc quản lý môi trường ảo và các thư viện Python, cho phép người dùng dễ dàng quản lý các phiên bản thư viện và tránh xung đột giữa các dự án khác nhau. Điều này đặc biệt quan trọng khi làm việc với các dự án học sâu, nơi mà việc đảm bảo tính nhất quán của các phiên bản thư viện có thể ảnh hưởng lớn đến kết quả huấn luyện và hiệu suất của mô hình.

Bên cạnh đó, PyCharm còn tích hợp với Jupyter Notebook, cung cấp một môi trường lý tưởng để thử nghiệm và trực quan hóa các kết quả huấn luyện mô hình. Jupyter Notebook cho phép các nhà nghiên cứu và sinh viên dễ dàng viết các đoạn mã ngắn, thực hiện các phép tính toán học phức tạp, và trực quan hóa kết quả ngay lập tức. Điều này giúp tối ưu hóa quá trình thử nghiệm, cho phép nhanh chóng kiểm tra các giả thuyết và điều chỉnh mô hình theo nhu cầu thực tế. Ngoài ra, công cụ gỡ lỗi mạnh mẽ của PyCharm hỗ trợ theo dõi quá trình huấn luyện mô hình, phát hiện và sửa lỗi kịp thời, đảm bảo rằng mô hình hoạt động ổn định và chính xác.

# DỮ LIỆU SINH VIÊN VÀ PHƯƠNG PHÁP XỬ LÝ

## Dữ liệu sinh viên

# XÂY DỰNG MÔ HÌNH MÁY HỌC DỰ ĐOÁN HIỆU XUẤT HỌC TẬP

# KẾT LUẬN

## Hiểu Biết về Phân Loại Âm Thanh và CNN

- **Khái niệm cơ bản:** Nhóm đã nắm vững các khái niệm cơ bản về phân loại âm thanh, bao gồm cách dữ liệu âm thanh được chuyển đổi thành các đặc trưng như hình ảnh phổ, tần số cơ bản (F0), và các formant (F1, F2). Các kỹ thuật như Biến đổi Fourier nhanh (FFT) giúp mô hình học máy sử dụng các đặc trưng này để phân loại âm thanh hiệu quả.

- **Mô hình CNN:** Nhóm đã hiểu rõ vai trò của mạng nơ-ron tích chập (CNN) trong phân loại âm thanh. CNN có khả năng trích xuất các đặc trưng quan trọng từ hình ảnh phổ và các đặc trưng âm thanh khác, giúp nâng cao độ chính xác của việc phân loại thông qua các lớp tích chập, gộp, và kết nối đầy đủ.

## Xây dựng và Đánh Giá Mô Hình CNN

- **Xây Dựng Mô Hình:** Thay vì sử dụng các kiến trúc RNN như LSTM vốn phổ biến trong xử lý dữ liệu âm thanh do tính chất chuỗi thời gian của nó, nhóm đã chọn sử dụng CNN để xây dựng mô hình phân loại âm thanh. Điều này được thực hiện bằng cách chuyển đổi âm thanh thành hình ảnh phổ (spectrogram), cho phép CNN xử lý dữ liệu âm thanh như một vấn đề thị giác. Mô hình CNN sau đó trích xuất các đặc trưng quan trọng từ hình ảnh phổ này, từ đó học cách phân loại các loại âm thanh khác nhau một cách hiệu quả.

- **Đánh Giá Hiệu Suất:** Sau khi huấn luyện, mô hình CNN đã được kiểm tra trên tập dữ liệu thử nghiệm, trong đó các tệp âm thanh được chuyển đổi thành hình ảnh phổ (spectrogram). Mô hình được tải từ tệp model.h5 và tiến hành dự đoán các lớp cho từng hình ảnh phổ. Quá trình dự đoán bao gồm việc tính toán xác suất cho mỗi lớp và chọn lớp có xác suất cao nhất làm kết quả cuối cùng.Kết quả dự đoán của mô hình được so sánh với nhãn thực tế trong tập kiểm thử. Những chỉ số như độ chính xác (accuracy) và độ nhạy (sensitivity) đều đạt mức cao, cho thấy mô hình có khả năng phân loại âm thanh hiệu quả. Nhóm đã hiển thị một số kết quả dự đoán và nhận thấy rằng chúng tương đối khớp với nhãn thực tế, khẳng định rằng mô hình CNN hoạt động tốt trên dữ liệu kiểm thử, và có thể ứng dụng trong phân loại âm thanh một cách tin cậy.

## Ứng Dụng và Tinh Chỉnh

- **Ứng dụng thực tế:** Mô hình đã chứng minh khả năng ứng dụng trong các tình huống thực tế như nhận diện âm thanh từ các tệp WAV, ví dụ như phân loại “tiếng chó sủa” hay “tiếng còi xe”. Điều này mang lại hiệu quả rõ rệt trong việc tự động hóa và cải thiện quy trình làm việc trong các hệ thống giám sát âm thanh và nhận diện môi trường.

- **Tinh chỉnh:** Nhóm đã thực hiện các tinh chỉnh cần thiết để cải thiện hiệu suất của mô hình, bao gồm việc tối ưu hóa các siêu tham số của CNN và nâng cao chất lượng dữ liệu đầu vào thông qua việc chuyển đổi âm thanh thành hình ảnh phổ (spectrogram). Những điều chỉnh này đã giúp nâng cao độ chính xác và khả năng phân loại âm thanh của mô hình.

## ĐÁNH GIÁ KẾT QUẢ

### Đánh giá triển khai tổng quan

Hiệu quả triển khai:

Việc triển khai mô hình CNN đã chứng tỏ khả năng phân loại âm thanh hiệu quả và mạnh mẽ. Mô hình đã đạt được độ chính xác cao trong việc nhận diện các loại âm thanh khác nhau, ví dụ như “tiếng chó sủa”, “tiếng còi xe”, và nhiều âm thanh khác. Sự linh hoạt của mô hình trong việc xử lý các dạng âm thanh khác nhau cho thấy sự phù hợp của nó với các ứng dụng thực tế. Mô hình đã thực hiện tốt trong các tình huống khác nhau, từ môi trường kiểm thử đến các điều kiện thực tế, cho thấy khả năng ứng dụng rộng rãi và hiệu quả trong việc tự động hóa quy trình phân loại âm thanh.

Tuy nhiên, trong quá trình triển khai, nhóm đã gặp phải một số khó khăn. Việc điều chỉnh mô hình để nhận diện các loại âm thanh mới hoặc chưa thấy trước đó là một thách thức đáng kể. Điều này thường yêu cầu phải có kiến thức sâu rộng về cấu trúc và hoạt động của mô hình CNN cũng như các kỹ thuật tiền xử lý dữ liệu. Bên cạnh đó, việc tối ưu hóa mô hình cho các loại âm thanh chưa được định danh cũng đòi hỏi nhiều thử nghiệm và tinh chỉnh, điều này có thể làm tăng thời gian và công sức cần thiết cho việc phát triển mô hình.

### Đề Xuất Cải Tiến

#### Tối ưu hóa mô hình:

Để nâng cao hiệu suất của mô hình CNN, nên xem xét việc áp dụng các kiến trúc mạng nơ-ron tiên tiến hơn, chẳng hạn như các mạng nơ-ron tích chập sâu (Deep CNNs) hoặc các mô hình mạng nơ-ron tích chập biến thể (CNN Variants) như ResNet hoặc DenseNet. Các phương pháp học sâu tiên tiến, như học chuyển giao (Transfer Learning) hoặc tăng cường học sâu (Deep Learning Enhancement), cũng có thể giúp cải thiện khả năng phân loại của mô hình bằng cách tận dụng các mô hình đã được huấn luyện trên các tập dữ liệu lớn và đa dạng.

#### Cải thiện chất lượng dữ liệu:

Để nâng cao chất lượng của mô hình, việc thu thập và làm phong phú thêm tập dữ liệu âm thanh là cần thiết. Điều này có thể bao gồm việc mở rộng quy mô dữ liệu bằng cách thu thập thêm các loại âm thanh từ nhiều nguồn khác nhau, cũng như sử dụng các kỹ thuật tăng cường dữ liệu (Data Augmentation) như thay đổi tốc độ, tăng cường âm lượng, hoặc thêm nhiễu để làm phong phú tập dữ liệu. Các kỹ thuật này giúp mô hình học được nhiều đặc điểm và biến thể của âm thanh hơn, từ đó cải thiện độ chính xác và khả năng tổng quát**.**

#### Tinh chỉnh siêu tham số:

Để đạt được kết quả tốt nhất, cần tiếp tục tinh chỉnh các siêu tham số của mô hình CNN, bao gồm số lượng lớp tích chập, kích thước kernel, và tỷ lệ dropout. Việc thử nghiệm với các cấu hình khác nhau của các tham số như tốc độ học (learning rate), kích thước batch, và các phương pháp tối ưu hóa (như Adam, RMSprop) cũng có thể giúp tối ưu hóa hiệu suất của mô hình. Quá trình tinh chỉnh này yêu cầu thử nghiệm và đánh giá thường xuyên để xác định cấu hình tốt nhất cho các bài toán phân loại âm thanh cụ thể.

## KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

### Kết Luận

**Tổng kết:**

Nghiên cứu đã chứng minh rằng mô hình CNN là một công cụ mạnh mẽ và hiệu quả trong việc phân loại âm thanh. Mô hình đã đạt được kết quả tốt trong việc nhận diện chính xác các loại âm thanh khác nhau, từ các âm thanh phổ biến như “tiếng chó sủa” đến các loại âm thanh khác. Khả năng của mô hình trong việc phân loại chính xác và linh hoạt cho thấy tiềm năng ứng dụng rộng rãi trong nhiều lĩnh vực thực tế, chẳng hạn như hệ thống giám sát âm thanh và tự động hóa các quy trình phân loại âm thanh.

**Ý nghĩa:**

Việc áp dụng CNN trong phân loại âm thanh không chỉ giúp tự động hóa quy trình phân loại mà còn cải thiện đáng kể hiệu suất làm việc trong các ứng dụng liên quan đến âm thanh. Mô hình CNN cho phép xử lý và phân tích âm thanh một cách hiệu quả, giảm thiểu sự can thiệp của con người và nâng cao độ chính xác trong việc nhận diện âm thanh.

### Hướng Phát Triển

Hướng nghiên cứu tiếp theo:

Để nâng cao khả năng tổng quát và hiệu suất của mô hình, nghiên cứu nên mở rộng bằng cách khám phá các kiến trúc mạng nơ-ron sâu hơn hoặc tích hợp các phương pháp học sâu tiên tiến như học không giám sát (unsupervised learning) hoặc học tăng cường (reinforcement learning). Việc này có thể giúp mô hình học được các đặc trưng âm thanh phức tạp và cải thiện khả năng phân loại các loại âm thanh chưa thấy trước đó.

Ứng dụng thực tiễn:

Khuyến khích triển khai mô hình vào các ứng dụng thực tiễn, chẳng hạn như hệ thống nhận diện âm thanh trong môi trường công nghiệp hoặc giám sát an ninh. Đồng thời, cần theo dõi và áp dụng các cải tiến công nghệ mới để nâng cao hiệu suất và khả năng của mô hình, giúp đáp ứng các yêu cầu ngày càng cao của thực tiễn.

## HẠN CHẾ

-**Hiểu biết chưa đầy đủ về các kỹ thuật xử lý tín hiệu:** Nhóm nhận thức rằng vẫn còn một số kỹ thuật xử lý tín hiệu âm thanh, như STFT, wavelet, chưa được khai thác hết tiềm năng.

-**Kiến thức về các kiến trúc mạng CNN còn hạn chế:** Nhóm chưa khám phá hết các biến thể của CNN như ResNet, Inception, và chưa tối ưu hóa được siêu tham số của mô hình.

-**Chưa hiểu sâu về lý thuyết học sâu**: Một số khái niệm như overfitting, underfitting, gradient vanishing, exploding gradient vẫn còn chưa được nắm vững hoàn toàn.

-**Chất lượng dữ liệu:** Dữ liệu âm thanh sử dụng để huấn luyện mô hình còn hạn chế về số lượng và đa dạng, ảnh hưởng đến khả năng tổng quát của mô hình.

-**Tính toán:** Việc huấn luyện mô hình đòi hỏi nhiều tài nguyên tính toán, nhóm chưa khai thác tối đa các công cụ và thư viện hỗ trợ.

TÀI LIỆU KHAM KHẢO

* 1. Myad S. E. O. (2024, July 18). Thuật toán CNN là gì? Thông tin cấu trúc của mạng CNN. FPT Aptech
  2. Học máy (Machine Learning) là gì? Cách hoạt động và ứng dụng. (n.d.). FPT IS - Công ty TNHH Hệ thống thông tin FPT.
  3. 4. T.-Q. lý N. (2023, February 27). Deep Learning là gì? Hiểu tổng quan về Deep Learning và ứng dụng. Tanca.io; Tanca là Sự kết hợp giữa trải nghiệm, hiểu biết sâu sắc, chiến lược con người với sự hỗ trợ của trí tuệ nhân tạo AI.
  4. PHÉP BIẾN ĐỔI FOURIER NHANH VÀ ỨNG DỤNG CỦA CHÚNG TRÊN NỀN CÔNG NGHỆ NHÚNG ARM TRONG NGHIÊN CỨU THIẾT KẾ, CHẾ TẠO THIẾT BỊ GIÁM SÁT CÁC GIẢI TẦN SỐ VÔ TUYẾN ĐIỆN. (n.d.). Org.vn.
  5. (N.d.). Wikipedia.org.
  6. SuNT. (n.d.). SuNT’s blog. Github.Io.