|  |  |
| --- | --- |
|  |  |
| LÊ THỊ NGỌC | BỘ CÔNG THƯƠNG  TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP HÀ NỘI  --------------------------------------- |
|  |
| ĐỒ ÁN TỐT NGHIỆP ĐẠI HỌC |
| NGÀNH HỆ THỐNG THÔNG TIN |
| NGHIÊN CỨU PHÁT TRIỂN ỨNG DỤNG  TÌM KIẾM ẢNH DỰA TRÊN MÔ TẢ |
|  |
|  |
| CBHD :TS. Phạm Văn Hà |
| HỆ THỐNG THÔNG TIN | Sinh viên : Lê Thị Ngọc |
| Mã số sinh viên: 2021601311 |
|  |
|  |
|  |
| Hà Nội – Năm 2024 |
|  |

# LỜI CẢM ƠN

Sau 4 năm học tập và rèn luyện tại Trường Đại học Công Nghiệp Hà Nội, đồ án tốt nghiệp này là một dấu ấn quan trọng đánh dấu việc em - một sinh viên đã hoàn thành nhiệm vụ của mình trên ghế giảng đường Đại học. Trong suốt quá trình nghiên cứu và hoàn thành đồ án, em đã nhận được sự hỗ trợ, giúp đỡ của nhiều cơ quan, tổ chức và các cá nhân. Với tình cảm sâu sắc, chân thành, cho phép em được bày tỏ lòng biết ơn sâu sắc đến tất cả các cá nhân và cơ quan đã tạo điều kiện giúp đỡ trong quá trình học tập và nghiên cứu đề tài.

Trước hết em xin gửi tới các thầy các cô khoa Công Nghệ Thông Tin Trường Đại học Công Nghiệp Hà Nội lời chào trân trọng, lời chúc sức khỏe và lời cảm ơn sâu sắc. Với sự quan tâm, dạy dỗ, chỉ bảo tận tình chu đáo của thầy cô, đến nay em đã có thể hoàn thành đồ án tốt nghiệp, đề tài: “Nghiên cứu phát triển ứng dụng tìm kiếm ảnh bằng mô tả”.

Đặc biệt em xin gửi lời cảm ơn chân thành đến thầy TS. Phạm Văn Hà người đã tận tình giúp và hướng dẫn em hoàn thành đề tài đồ án.

Đồng thời, em xin bày tỏ lòng biết ơn đến lãnh đạo Trường Đại học Công Nghiệp Hà Nội, các Khoa, Phòng ban chức năng đã tạo điều kiện cho em được học tập tại nơi mà em yêu thích, cho em bước vào đời sống thực tế và áp dụng những kiến thức em đã học tại trường và môi trường làm việc mới của em. Qua quá trình học tập em đã tích lũy được rất nhiều kiến thức để chuẩn bị cho công việc sau này cũng như để phát triển thêm bản thân.

Trong quá trình hoàn thành đồ án tốt nghiệp không thể tránh khỏi thiếu sót, kính mong có sự góp ý từ thầy cô.

*Em xin chân thành cảm ơn!*

Lê Thị Ngọc

Trong thời đại công nghệ số, hình ảnh đóng vai trò quan trọng không chỉ trong lưu trữ và chia sẻ thông tin mà còn trong việc cung cấp các giải pháp thông minh phục vụ nhiều lĩnh vực như thương mại, giáo dục, y tế và giải trí. Với sự bùng nổ của dữ liệu hình ảnh, việc tìm kiếm và truy xuất hình ảnh một cách hiệu quả, chính xác đang trở thành một nhu cầu cấp thiết.

Các phương pháp tìm kiếm truyền thống, dựa trên từ khóa hoặc siêu dữ liệu, ngày càng bộc lộ nhiều hạn chế khi phải đối mặt với khối lượng dữ liệu lớn và yêu cầu truy vấn linh hoạt từ người dùng. Chính vì thế, việc phát triển các hệ thống tìm kiếm ảnh dựa trên mô tả bằng ngôn ngữ tự nhiên đã mở ra một hướng tiếp cận mới đầy tiềm năng, cho phép người dùng tìm kiếm trực tiếp bằng cách sử dụng các mô tả gần gũi, dễ hiểu.

Đề tài “Nghiên cứu phát triển ứng dụng tìm kiếm ảnh dựa trên mô tả” được lựa chọn với mong muốn nghiên cứu và phát triển một giải pháp tối ưu, mang tính ứng dụng cao trong việc truy xuất thông tin từ hình ảnh. Đề tài không chỉ góp phần giải quyết các bài toán thực tiễn mà còn là cơ hội để em tìm hiểu sâu hơn về các công nghệ hiện đại, đồng thời vận dụng các kiến thức đã học vào thực tế.

Để hoàn thành được đồ án tốt nghiệp này, em xin được gửi lời cảm ơn chân thành đến các thầy cô trong khoa Công nghệ thông tin Trường Đại học Công Nghiệp Hà Nội đã tận tình giảng dạy và trang bị kiến thức cho em trong suốt thời gian em học tập tại trường. Thầy giáo hướng dẫn đề tài – Tiến Sĩ Phạm Văn Hà, giảng viên Khoa Công nghệ thông tin Trường Đại học Công Nghiệp Hà Nội – đã tận tụy hết lòng giúp đỡ, hướng dẫn, chỉ dẫn tận tình để giúp em hoàn thành được đồ án.

Mục Lục

[LỜI CẢM ƠN 2](#_Toc133440515)

[LỜI NÓI ĐẦU 3](#_Toc133440516)

[DANH MỤC CÁC CHỮ VIẾT TẮT 7](#_Toc133440517)

[DANH MỤC CÁC HÌNH VẼ 8](#_Toc133440518)

[CHƯƠNG 1: TỔNG QUAN VỀ ĐỀ TÀI 9](#_Toc133440519)

[1.1. Tên đề tài 9](#_Toc133440520)

[1.2. Lý do chọn đề tài 9](#_Toc133440521)

[1.3. Mục tiêu của đề tài 10](#_Toc133440522)

[1.4. Đối tượng và phạm vi 11](#_Toc133440523)

[1.5. Kết quả dự kiến đạt được 11](#_Toc133440524)

[1.6. Tổng kết chương 1 11](#_Toc133440525)

[CHƯƠNG 2: NGHIÊN CỨU MÔ HÌNH HỌC SÂU (DEEP LEARNING) 12](#_Toc133440526)

[2.1. Tổng quan về mô hình học sâu 12](#_Toc133440527)

[2.1.1. Mô hình học sâu (Deep learning) là gì? 12](#_Toc133440528)

[2.1.2. Deeplearning hoạt động như thế nào? 13](#_Toc133440529)

[2.1.3. Ưu nhược điểm của mô hình học sâu 14](#_Toc133440530)

[2.1.3.1. Ưu điểm 14](#_Toc133440531)

[2.1.3.2. Nhược điểm 15](#_Toc133440532)

[2.1.4. Ứng dụng của mô hình học sâu 15](#_Toc133440533)

[2.2. Một số mô hình học sâu 16](#_Toc133440534)

[2.2.1. Mạng nơ-ron cổ điển 16](#_Toc133440535)

[2.2.2. Mạng nơ-ron tích chập (CNN) 17](#_Toc133440536)

[2.2.3. Mạng nơ-ron hồi quy (RNN) 18](#_Toc133440537)

[2.2.4. Mạng sinh đối nghịch (GAN) 19](#_Toc133440538)

[2.3. Tổng hợp chương 2 19](#_Toc133440539)

[CHƯƠNG 3: CÁC CÔNG CỤ PHẦN MỀM ĐƯỢC SỬ DỤNG TRONG NHẬN DẠNG KHUÔN MẶT 20](#_Toc133440540)

[3.1. Thư viện Dlib 20](#_Toc133440541)

[3.1.1. Định nghĩa 20](#_Toc133440542)

[3.1.2. Ưu, nhược điểm 20](#_Toc133440543)

[3.1.2.1. Ưu điểm 20](#_Toc133440544)

[3.1.2.2. Nhược điểm 20](#_Toc133440545)

[3.1.3. Cài đặt thực nghiệm 21](#_Toc133440546)

[3.1.3.1. Quy trình 21](#_Toc133440547)

[3.1.3.2. Code demo 22](#_Toc133440548)

[3.1.3.3. Kết quả 22](#_Toc133440549)

[3.2. Mô hình RetinaFace 23](#_Toc133440550)

[3.2.1. Tổng quan 23](#_Toc133440551)

[3.2.2. Ưu, nhược điểm 23](#_Toc133440552)

[3.2.2.1. Ưu điểm 23](#_Toc133440553)

[3.2.2.2. Nhược điểm 23](#_Toc133440554)

[3.2.3. Cài đặt thực nghiệm 24](#_Toc133440555)

[3.2.3.1. Quy trình 24](#_Toc133440556)

[3.2.3.2. Code demo 24](#_Toc133440557)

[3.2.3.3. Kết quả 25](#_Toc133440558)

[3.3. Mô hình MTCNN 25](#_Toc133440559)

[3.3.1. Định nghĩa 25](#_Toc133440560)

[3.3.2. Ưu, nhược điểm 26](#_Toc133440561)

[3.3.2.1. Ưu điểm 26](#_Toc133440562)

[3.3.2.2. Nhược điểm 26](#_Toc133440563)

[3.3.3. Cài đặt thực nghiệm 26](#_Toc133440564)

[3.3.3.1. Quy trình 26](#_Toc133440565)

[3.3.3.2. Code demo 27](#_Toc133440566)

[3.3.3.3. Kết quả 27](#_Toc133440567)

[3.4. Công cụ FaceAPI 28](#_Toc133440568)

[3.4.1. Khái niệm 28](#_Toc133440569)

[3.4.2. Ưu, nhược điểm 28](#_Toc133440570)

[3.4.2.1. Ưu điểm 28](#_Toc133440571)

[3.4.2.2. Nhược điểm 29](#_Toc133440572)

[3.4.3. Cài đặt thực nghiệm 29](#_Toc133440573)

[3.4.3.1. Quy trình 29](#_Toc133440574)

[3.4.3.2. Code demo 31](#_Toc133440575)

[3.4.3.3. Kết quả 31](#_Toc133440576)

[3.5. Đánh giá và tổng kết 32](#_Toc133440577)

[3.6. Tổng kết chương 3 33](#_Toc133440578)

[CHƯƠNG 4: XÂY DỰNG HỆ THỐNG NHẬN DIỆN KHUÔN MẶT TRẢ VỀ THÔNG TIN 34](#_Toc133440579)

[4.1. Quy trình 34](#_Toc133440580)

[4.1.1. Từ camera -> hình ảnh 34](#_Toc133440581)

[4.1.2. Từ hình ảnh -> Các ảnh khuôn mặt 34](#_Toc133440582)

[4.1.3. Các ảnh -> Mã hóa vector 128d hoặc 512d 34](#_Toc133440583)

[4.1.4. Lưu các vector đã mã hóa bằng Elasticsearch 34](#_Toc133440584)

[4.1.5. Dùng Elasticsearch truy vấn trả về kết quả 34](#_Toc133440585)

[4.2. Thu thập dữ liệu 34](#_Toc133440586)

[4.3. Cài đặt thực nghiệm 35](#_Toc133440587)

[4.3.1. Cài đặt Python 35](#_Toc133440588)

[4.3.2. Cài đặt Elasticsearch 36](#_Toc133440589)

[4.3.3. Cài đặt các thư viện cần thiết 36](#_Toc133440590)

[4.3.4. Chạy chương trình 36](#_Toc133440591)

[4.4. Đánh giá kết quả thực nghiệm 39](#_Toc133440592)

[4.5. Tổng kết chương 4 41](#_Toc133440593)

[KẾT LUẬN 42](#_Toc133440594)

[TÀI LIỆU THAM KHẢO 43](#_Toc133440596)

# DANH MỤC CÁC CHỮ VIẾT TẮT

|  |  |  |
| --- | --- | --- |
| Từ viết tắt | Tiếng anh | Tiếng Việt |
| CNN | Convolutional Neural Network | Mạng nơ-ron tích chập |
| CRNN | Convolutional-Recurrent Neural Network | Mạng nơ-ron hồi quy tích chập |
| CNTT |  | Công nghệ thông tin |
| DNN | Deep neural networks | Các mạng nơ-ron sâu |
| GAN | Generative Adversarial Networks | Mạng sinh đối nghịch |
| LSTM | Long Short-Term Memory |  |
| MTCNN | Multi-task Cascaded Convolutional Networks |  |
| RNN | Recurrent Neural Network (Mạng nơ-ron hồi quy) |  |

# DANH MỤC CÁC HÌNH VẼ

[Hình 2.1 Tổng quan về Deeplearning 13](#_Toc133440368)

[Hình 2.2 Quy trình hoạt động của Deelearning 14](#_Toc133440369)

[Hình 2.3 Mạng nơ-ron tích chập (CNN) 18](#_Toc133440370)

[Hình 2.4 Mạng nơ-ron hồi quy (RNN) 19](#_Toc133440371)

[Hình 3.1 Code demo nhận dạng khuôn mặt sử dụng Thư viện Dlib 22](#_Toc133440372)

[Hình 3.2 Kết quả nhận dạng khuôn mặt sử dụng Thư viện Dlib 22](#_Toc133440373)

[Hình 3.3 Code demo nhận dạng khuôn mặt sử dụng mô hình RetinaFace 24](#_Toc133440374)

[Hình 3.4 Kết quả nhận dạng khuôn mặt sử dụng mô hình RetinaFace 25](#_Toc133440375)

[Hình 3.5 Code demo nhận dạng khuôn mặt sử dụng mô hình MTCNN 27](#_Toc133440376)

[Hình 3.6 Kết quả nhận dạng khuôn mặt sử dụng mô hình MTCNN 28](#_Toc133440377)

[Hình 3.7 Code demo nhận dạng khuôn mặt sử dụng công cụ FaceAPI 31](#_Toc133440378)

[Hình 3.8 Kết quả nhận dạng khuôn mặt sử dụng công cụ FaceAPI 32](#_Toc133440379)

[Hình 4.1 Bộ dữ liệu ảnh các khuôn mặt sinh viên ngành CNTT khóa 14 35](#_Toc133440380)

[Hình 4.2 Demo nhận diện khuôn mặt từ ảnh chụp bằng camera 39](#_Toc133440381)

[Hình 4.3 Demo nhận diện khuôn mặt từ ảnh chọn từ thư viện của máy 39](#_Toc133440382)

# MỞ ĐẦU

## Lý do chọn đề tài

Công cụ tìm kiếm ảnh dựa trên mô tả đáp ứng nhu cầu tìm kiếm nhanh chóng và quản lý hiệu quả kho ảnh cá nhân, nâng cao trải nghiệm người dùng khi giúp họ dễ dàng truy xuất ảnh dựa trên ký ức về ngữ cảnh. Ngoài ra, công nghệ này còn có tiềm năng ứng dụng rộng rãi trong các lĩnh vực như thương mại điện tử, y tế và truyền thông, mang lại lợi ích thiết thực trong quản lý và truy xuất hình ảnh chuyên nghiệp

Nhu cầu tìm kiếm nhanh trong kho ảnh cá nhân: Ngày nay, người dùng lưu trữ rất nhiều ảnh trên các nền tảng đám mây như Google Photos, và việc tìm lại một ảnh cụ thể có thể mất nhiều thời gian nếu chỉ dựa vào duyệt thủ công. Một công cụ cho phép tìm kiếm theo mô tả ngữ nghĩa giúp người dùng dễ dàng tìm lại các ảnh dựa trên chi tiết nhớ được, như "ảnh chụp sinh nhật năm ngoái" hay "ảnh đi du lịch biển".

Quản lý và truy xuất dữ liệu cá nhân hiệu quả: Với tốc độ lưu trữ ảnh ngày càng tăng, các cá nhân thường gặp khó khăn khi cần phân loại hoặc tổ chức ảnh của mình. Công cụ tìm kiếm ảnh dựa trên mô tả cung cấp một cách tiếp cận hiệu quả hơn, giúp dễ dàng quản lý và truy xuất dữ liệu hình ảnh cá nhân khi cần.

Tăng cường trải nghiệm người dùng: Người dùng thường ghi nhớ các chi tiết và ngữ cảnh liên quan đến ảnh thay vì tên tệp hay thông tin kỹ thuật. Việc hỗ trợ tìm kiếm ngữ nghĩa dựa trên mô tả giúp người dùng tiếp cận và tận hưởng kho ảnh của mình theo cách tự nhiên và thuận tiện hơn.

Ứng dụng linh hoạt và hữu ích cho các nền tảng xã hội: Ngoài các kho ảnh cá nhân, tính năng tìm kiếm ngữ nghĩa có thể mở rộng sang các nền tảng xã hội hoặc chia sẻ ảnh công khai, giúp người dùng dễ dàng tìm kiếm và gợi nhớ các kỷ niệm hoặc sự kiện qua hình ảnh theo một cách mới mẻ và chính xác.

Tiềm năng tích hợp vào các lĩnh vực khác: Ngoài ứng dụng cá nhân, công nghệ tìm kiếm ngữ nghĩa có thể được áp dụng trong các lĩnh vực khác như thương mại điện tử (tìm kiếm sản phẩm qua ảnh), y tế (phân loại ảnh chụp X-quang, MRI), và truyền thông (quản lý kho dữ liệu hình ảnh cho quảng cáo, báo chí), hỗ trợ việc xử lý và truy xuất ảnh chuyên nghiệp.

## Mục tiêu của đề tài

Đề tài: Nghiên cứu ứng dụng tìm kiếm ảnh dựa trên mô tảđáp ứng được những mục tiêu:

Nghiên cứu mô hình học sâu, tập trung tìm hiểu các mô hình liên quan đến embedding văn bản và hình ảnh: BERT, Sentence Tranformer, Vision Transformer, Resnet,..cùng một số phương pháp tính khoảng cách giữa 2 vector embedding như Cosine Similarity,…

Nắm được các kiến thức cơ bản về python, nghiên cứu các phương pháp, công cụ để kết hợp các mô hình phục vụ cho mục tiêu tìm kiếm ảnh bằng đoạn mô tả.

Xây dựng ứng dụng demo trên tập dữ liệu hình ảnh cố định và trả về tập dữ liệu hình ảnh khớp với đoạn văn mô tả.

## Đối tượng và phạm vi

Đối tượng nghiên cứu: Các phương pháp và kỹ thuật liên quan đến tìm kiếm ảnh dựa trên mô tả ngôn ngữ tự nhiên, bao gồm các phương pháp ánh xạ giữa văn bản và hình ảnh trong không gian tiềm ẩn.

Phạm vi nghiên cứu:

* Tập trung nghiên cứu và phát triển hệ thống tìm kiếm ảnh dựa trên mô tả ngôn ngữ tự nhiên.
* Nghiên cứu các phương pháp ánh xạ dữ liệu văn bản và hình ảnh, không đi sâu vào các lĩnh vực liên quan như phân loại hình ảnh, phân tích nội dung chi tiết trong ảnh hoặc tạo dữ liệu mô tả tự động.
* Phạm vi ứng dụng giới hạn trong việc tìm kiếm và truy xuất hình ảnh từ các bộ sưu tập ảnh tĩnh, không bao gồm video hoặc các định dạng dữ liệu khác.

## Kết quả dự kiến đạt được

Báo cáo tổng quan về các phương pháp và công nghệ liên quan đến tìm kiếm ảnh dựa trên mô tả ngôn ngữ tự nhiên.

Đánh giá và so sánh hiệu quả của các mô hình vector hoá hình ảnh và văn bản.

Xây dựng hệ thống tìm kiếm ảnh dựa trên mô tả với tệp dữ liệu cố định, cho phép người dùng nhập mô tả văn bản và nhận kết quả là các hình ảnh phù hợp.

Đánh giá hiệu quả hệ thống thông qua các tiêu chí như độ chính xác, tốc độ truy vấn, và khả năng đáp ứng các mô tả phức tạp.

## Bố cục của đề tài

# CƠ SỞ LÝ THUYẾT THỰC HIỆN BÀI TOÁN TÌM KIẾM ẢNH TRÊN MÔ TẢ

## Tổng quan về trí tuệ nhân tạo, học máy và học sâu

### Trí tuệ nhân tạo

Theo như cha đẻ của trí tuệ nhân tạo, John McCarthy thì nó là *"*Khoa học và kỹ thuật của việc tạo ra những máy thông minh, đặc biệt là chương trình máy tính thông minh".

Trí tuệ nhân tạo là hướng đi của việc tạo ra máy tính, người máy điều khiển bằng máy tính hay là những phần mềm suy nghĩ thông minh hơn, tương tự như suy nghĩ thông minh của con người.

Trí tuệ nhân tạo được học như bộ não con người, như cách mà con người học, quyết định và làm việc khi giải quyết một vấn đề, và sau đó sử dụng kết quả của quá trình học đó như là nền tảng của việc phát triển phần mềm và hệ thống thông minh.

Ở thời điểm hiện tại, Thuật ngữ này thường dùng để nói đến các máy tính có mục đích không nhất định và ngành khoa học nghiên cứu về các lý thuyết và ứng dụng của trí tuệ nhân tạo. Tức là mỗi loại trí tuệ nhân tạo hiện nay đang dừng lại ở mức độ những máy tính hoặc siêu máy tính dùng để xử lý một loại công việc nào đó như điều khiển một ngôi nhà, nghiên cứu nhận diện hình ảnh, xử lý dữ liệu của bệnh nhân để đưa ra phác đồ điều trị, xử lý dữ liệu để tự học hỏi, khả năng trả lời các câu hỏi về chẩn đoán bệnh, trả lời khách hàng về các sản phẩm của một công ty, ...

A robot touching a digital device

Description automatically generated

Hình 1.1: Mô tả hình ảnh của Trí tuệ nhân tạo

Trí tuệ nhân tạo (AI) là hệ thống trí tuệ do con người tạo ra, có khả năng tư duy, học hỏi và xử lý dữ liệu ở quy mô lớn hơn, nhanh chóng và hiệu quả hơn so với con người. Công nghệ này được các tập đoàn lớn trên thế giới chú trọng phát triển nhờ tiềm năng to lớn trong việc giải quyết các vấn đề phức tạp mà con người chưa thể giải quyết.

Mặc dù mang lại nhiều lợi ích cho xã hội, AI cũng tiềm ẩn những rủi ro. Nhiều chuyên gia cảnh báo rằng nếu AI đạt đến mức tiến hóa vượt trội, nó có thể đe dọa sự tồn vong của nhân loại. Chủ đề này đã được khai thác qua nhiều góc nhìn trong các tác phẩm điện ảnh, nhằm nhấn mạnh mối nguy hiểm tiềm tàng và cảnh tỉnh về việc quản lý phát triển AI.

### Học máy

Học máy là một nhánh của trí tuệ nhân tạo (AI), tập trung vào việc phát triển các thuật toán và hệ thống cho phép máy tính học từ dữ liệu. Điều này giúp máy tính cải thiện hiệu suất thực hiện các nhiệm vụ cụ thể theo thời gian mà không cần lập trình chi tiết cho từng trường hợp. Mục tiêu của học máy là phát triển các mô hình có khả năng tự động nhận dạng, phân tích và ra quyết định dựa trên dữ liệu đầu vào.

Học máy thường được triển khai qua ba giai đoạn chính:

* Đào tạo (Training): Trong giai đoạn này, máy tính được cung cấp một tập dữ liệu lớn, được sử dụng để nhận biết các mẫu, đặc điểm và mối quan hệ bên trong dữ liệu. Quá trình này giúp máy "học" các quy luật từ dữ liệu được cung cấp.
* Dự đoán (Prediction): Sau khi được đào tạo, máy sử dụng các quy luật đã học để đưa ra dự đoán khi tiếp nhận dữ liệu mới. Đây là bước kiểm tra khả năng áp dụng của mô hình đã học vào thực tế.
* Ra quyết định (Decision Making): Dựa trên kết quả dự đoán, máy thực hiện các hành động hoặc ra quyết định phù hợp mà không cần sự can thiệp từ con người.

A diagram of a diagram of data

Description automatically generated with medium confidence

Hình 1.2: Mô tả các bước chính của học máy trong vòng đời khoa học dữ liệu

Một ví dụ cơ bản về học máy là việc nhận diện hình ảnh quả táo. Để thực hiện, hệ thống sẽ được cung cấp hàng ngàn hình ảnh, bao gồm cả hình ảnh của quả táo và các vật thể khác. Máy sẽ phân tích dữ liệu này để nhận diện các đặc điểm đặc trưng của quả táo, chẳng hạn như hình dạng, màu sắc và kết cấu. Sau đó, khi gặp hình ảnh mới, máy có thể xác định liệu hình ảnh đó có chứa quả táo hay không.

Để thực hiện huấn luyện học máy hiệu quả, ta cần phải đảm bảo các yếu tố sau:

* Lượng dữ liệu đào tạo lớn: Một lượng dữ liệu đủ lớn và đa dạng là điều kiện cần thiết để mô hình học được các mẫu và đặc điểm một cách chính xác.
* Sức mạnh tính toán: Các thuật toán học máy yêu cầu khả năng xử lý dữ liệu nhanh và hiệu quả, điều này đòi hỏi phần cứng mạnh mẽ và tài nguyên tính toán đáng kể.
* Thuật toán học máy mạnh mẽ: Thuật toán cần đủ linh hoạt và hiệu quả để trích xuất thông tin từ dữ liệu, đồng thời có khả năng cải thiện liên tục khi có thêm dữ liệu hoặc điều chỉnh theo yêu cầu mới.

Học máy là nền tảng cho nhiều ứng dụng hiện đại, từ nhận diện hình ảnh, xử lý ngôn ngữ tự nhiên, đến các hệ thống dự báo và tối ưu hóa. Bằng cách tận dụng dữ liệu và sức mạnh tính toán, học máy đóng vai trò quan trọng trong việc tự động hóa và nâng cao hiệu quả hoạt động trong nhiều lĩnh vực.

### Học sâu

Học sâu là một nhánh chuyên biệt của học máy, tập trung vào việc sử dụng các mô hình mạng lưới thần kinh nhiều lớp (deep neural networks) để xử lý và học từ dữ liệu phức tạp, trừu tượng. Các mô hình học sâu có khả năng tự động phát hiện và trích xuất các đặc trưng quan trọng từ dữ liệu mà không cần sự can thiệp thủ công của con người.

Trong khi học máy bao gồm nhiều thuật toán khác nhau như cây quyết định, máy vector hỗ trợ (SVM), và hồi quy tuyến tính, học sâu sử dụng chủ yếu mạng lưới thần kinh nhiều lớp. Điểm nổi bật của học sâu là khả năng xử lý dữ liệu lớn và phức tạp, chẳng hạn như hình ảnh, âm thanh, và ngôn ngữ tự nhiên, từ đó xây dựng các mô hình có độ chính xác cao hơn.

Học máy truyền thống thường yêu cầu bước tiền xử lý dữ liệu và trích xuất đặc trưng thủ công, trong khi học sâu có thể tự động hóa quá trình này thông qua các lớp ẩn, giúp giảm thiểu sự phụ thuộc vào chuyên môn của con người.

Mạng lưới thần kinh nhân tạo (Artificial Neural Networks) là nền tảng của học sâu, được lấy cảm hứng từ cách thức hoạt động của các tế bào thần kinh trong não người.

* Nơron nhân tạo (Artificial Neuron): Mỗi "nơron" trong mạng lưới thần kinh là một đơn vị xử lý thông tin đơn giản, nhận đầu vào, thực hiện tính toán thông qua một hàm kích hoạt và cung cấp đầu ra cho nơron kế tiếp.
* Cấu trúc nhiều lớp: Mạng lưới thần kinh trong học sâu thường bao gồm nhiều lớp ẩn (hidden layers), nơi mỗi lớp chịu trách nhiệm học các đặc trưng cụ thể từ dữ liệu.
* Kết nối và trọng số: Các nơron được kết nối với nhau thông qua các trọng số (weights), điều chỉnh trong quá trình học để tối ưu hóa hiệu suất của mô hình.

Diagram of a cell structure

Description automatically generated

Hình 1.3: Mô tả từ nơ-ron sinh học thành nơ-ron nhân tạo

Mạng lưới thần kinh hoạt động dựa trên nguyên tắc truyền thông tin qua nhiều lớp, bao gồm:

* Lớp đầu vào (Input Layer): Tiếp nhận dữ liệu thô từ bên ngoài.
* Các lớp ẩn (Hidden Layers): Xử lý và trừu tượng hóa dữ liệu thông qua các phép tính toán phức tạp, mỗi lớp trích xuất các đặc trưng cao hơn từ dữ liệu.
* Lớp đầu ra (Output Layer): Cung cấp kết quả cuối cùng, chẳng hạn như một dự đoán hoặc phân loại.

A diagram of a network

Description automatically generated

Hình 1.4: Mô tả kiến trúc cơ bản của một mô hình học sâu

Quá trình học trong mạng lưới thần kinh sử dụng thuật toán lan truyền ngược (backpropagation) để điều chỉnh trọng số dựa trên sai số giữa đầu ra dự đoán và kết quả thực tế.

Học sâu hiện nay đã được ứng dụng rộng rãi trong nhiều lĩnh vực:

* Trí tuệ nhân tạo tạo sinh (Generative AI): Tạo nội dung mới như văn bản, hình ảnh, hoặc âm thanh.
* ChatGPT: Hệ thống mô hình ngôn ngữ tự nhiên tiên tiến, có khả năng hiểu và tạo ra văn bản tự nhiên theo ngữ cảnh.
* Nhận diện hình ảnh: Phân loại và nhận diện các đối tượng trong ảnh, ứng dụng trong y tế (chẩn đoán hình ảnh) và công nghệ xe tự lái.
* Xử lý ngôn ngữ tự nhiên (NLP): Dịch tự động, phân tích cảm xúc, và tóm tắt văn bản.

Học sâu đòi hỏi các yếu tố sau để đạt được hiệu quả tối ưu:

* Sức mạnh tính toán cao: Do các mô hình mạng lưới thần kinh phức tạp, học sâu cần đến các bộ xử lý mạnh mẽ như GPU hoặc TPU để xử lý và huấn luyện mô hình.
* Dữ liệu lớn: Các mô hình học sâu yêu cầu lượng dữ liệu đào tạo khổng lồ để học và tối ưu hóa các trọng số, đặc biệt là khi xử lý dữ liệu không có cấu trúc như hình ảnh và âm thanh.
* Thuật toán phức tạp: Cần phát triển các thuật toán và kiến trúc mạng lưới tiên tiến để giải quyết các vấn đề đặc thù của từng bài toán.

Học sâu đã mở ra một kỷ nguyên mới trong AI, mang lại các giải pháp vượt trội cho các vấn đề phức tạp, nhưng cũng đi kèm với những thách thức lớn về tài nguyên và tính đạo đức trong ứng dụng.

## Định nghĩa bài toán tìm kiếm ảnh trên mô tả

Bài toán tìm kiếm ảnh dựa trên mô tả bao gồm hai thành phần chính: xử lý văn bản và nhận diện nội dung hình ảnh. Đây là công nghệ cho phép tìm kiếm các hình ảnh phù hợp dựa trên một đoạn văn bản mô tả, liên quan đến việc xử lý đầu vào là văn bản từ người dùng, ánh xạ chúng vào một không gian nhúng chung với hình ảnh và tìm kiếm các ảnh tương đồng nhất.

Theo một khảo sát về công nghệ này, tìm kiếm ảnh dựa trên mô tả là một lĩnh vực nghiên cứu quan trọng, góp phần nâng cao hiệu quả trong việc truy xuất thông tin trực quan và quản lý dữ liệu đa phương tiện. Các phương pháp ban đầu chủ yếu dựa vào việc trích xuất thủ công đặc trưng từ văn bản và hình ảnh, sử dụng các kỹ thuật xử lý truyền thống như phân tích ngữ nghĩa đơn giản hoặc nhận dạng hình dạng cơ bản. Tuy nhiên, trong những năm gần đây, sự phát triển của các mô hình học sâu đã mở ra bước ngoặt quan trọng. Đặc biệt, các mô hình tiên tiến như CLIP (Contrastive Language–Image Pretraining) và BLIP (Bootstrapped Language-Image Pretraining) đã được áp dụng rộng rãi, mang lại hiệu quả vượt trội. Những mô hình này sử dụng kiến trúc Transformer để học không gian nhúng chung giữa văn bản và hình ảnh, kết hợp với phương pháp học đối sánh (contrastive learning) hoặc huấn luyện đa nhiệm (multi-task learning) để tối ưu hóa việc liên kết ngữ nghĩa. Ngoài ra, các kiến trúc mới như Flamingo và ALIGN cũng đã chứng minh tiềm năng trong việc cải thiện độ chính xác và khả năng tổng quát hóa của bài toán tìm kiếm ảnh dựa trên mô tả.

Những tiến bộ này đã không chỉ cải thiện đáng kể hiệu suất mà còn đưa công nghệ tìm kiếm ảnh dựa trên mô tả đến gần hơn với ứng dụng thực tiễn, đặc biệt trong lĩnh vực thương mại điện tử, nơi người dùng có thể tìm kiếm sản phẩm thông qua mô tả, hoặc trong việc tổ chức và quản lý dữ liệu đa phương tiện trong các kho lưu trữ lớn.

## Ứng dụng

### Thương mại điện tử: Amazon Visual Search

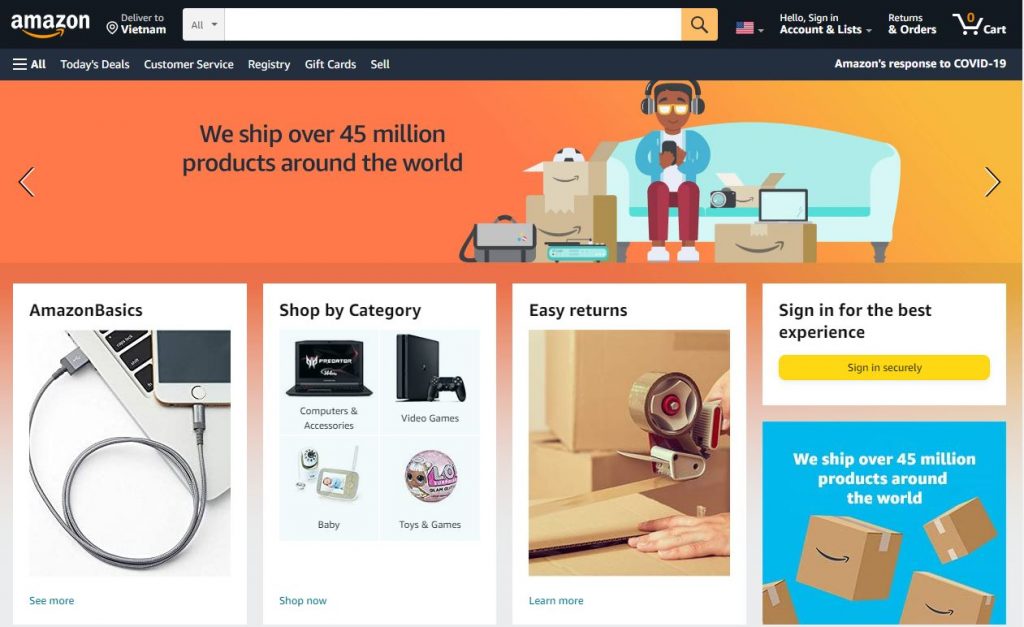
Amazon Visual Search là một tính năng cho phép người dùng tìm kiếm các sản phẩm bằng cách sử dụng mô tả văn bản hoặc hình ảnh. Công nghệ này giúp kết nối các mô tả ngôn ngữ tự nhiên với cơ sở dữ liệu hình ảnh sản phẩm khổng lồ trên Amazon.

Cách hoạt động:

* Người dùng nhập mô tả sản phẩm (ví dụ: "giày thể thao màu xanh có đế trắng") hoặc tải lên một hình ảnh.
* Hệ thống sử dụng các mô hình học sâu như CLIP để ánh xạ mô tả và hình ảnh vào cùng một không gian nhúng.
* Tìm kiếm sản phẩm dựa trên khoảng cách giữa các vector nhúng trong cơ sở dữ liệu và trả về các kết quả phù hợp nhất.

Lợi ích:

* Cải thiện trải nghiệm mua sắm bằng cách giảm thời gian tìm kiếm sản phẩm.
* Tăng doanh số bằng cách gợi ý chính xác sản phẩm mà người dùng quan tâm



Hình 1.5: Amazon Visual Search

### Công cụ tìm kiếm hình ảnh: Google Images

Google Images sử dụng công nghệ Text-to-Image Retrieval để tìm kiếm hình ảnh liên quan dựa trên mô tả văn bản từ người dùng. Đây là một trong những ứng dụng lớn nhất, được hàng tỷ người sử dụng hàng ngày.

Cách hoạt động:

* Người dùng nhập một truy vấn văn bản (ví dụ: "sunset over the mountains").
* Hệ thống phân tích truy vấn, chuyển đổi văn bản thành vector đặc trưng bằng các mô hình NLP tiên tiến như BERT hoặc CLIP.
* Các vector này được so sánh với vector đặc trưng của hình ảnh trong cơ sở dữ liệu Google Images.
* Hình ảnh phù hợp nhất được trả về theo mức độ tương đồng.

Lợi ích:

* Cung cấp cách tiếp cận nhanh chóng và hiệu quả để tìm kiếm thông tin trực quan.
* Hỗ trợ các ngành như thiết kế, báo chí, và nghiên cứu khoa học



Hình 1.6: Google Images với công nghệ Text-to-Image Retrieval

### Mạng xã hội: Pinterest Lens

Pinterest Lens là một tính năng tìm kiếm trực quan trên Pinterest, giúp người dùng khám phá ý tưởng hoặc sản phẩm mới dựa trên mô tả văn bản hoặc hình ảnh.

Cách hoạt động:

* Người dùng mô tả ý tưởng hoặc tải lên hình ảnh (ví dụ: "ý tưởng trang trí bàn tiệc cưới").
* Pinterest Lens sử dụng các mô hình học sâu như CLIP hoặc các phiên bản tùy chỉnh của Transformer để kết nối nội dung mô tả với cơ sở dữ liệu hình ảnh lớn trên nền tảng.
* Hệ thống trả về các hình ảnh hoặc bài đăng liên quan nhất, đi kèm các gợi ý sáng tạo.

Lợi ích:

* Tăng cường sự sáng tạo và khám phá, đặc biệt trong các lĩnh vực như thời trang, nội thất, và phong cách sống.
* Cải thiện mức độ tương tác và thời gian người dùng sử dụng nền tảng.



Hình 1.7: Pinterest Lens với khả năng tìm kiếm trực quan

## Các phương pháp tiếp cận bài toán

### Phương pháp cổ điển

Phương pháp cổ điển trong tìm kiếm ảnh dựa trên mô tả sử dụng các kỹ thuật như Bag of Words (BoW), TF-IDF, SIFT, và K-Means Clustering. Những phương pháp này tập trung vào việc trích xuất đặc trưng từ hình ảnh và mô tả, sau đó sử dụng các thuật toán thống kê hoặc máy học truyền thống để so sánh.

* BoW và TF-IDF: Phương pháp BoW biểu diễn mô tả hoặc văn bản liên quan dưới dạng vector tần suất từ, trong khi TF-IDF tính toán mức độ quan trọng của từ trong một tập hợp tài liệu. Khi áp dụng trong tìm kiếm ảnh, BoW và TF-IDF không trực tiếp xử lý hình ảnh mà thường sử dụng trên các mô tả liên quan đến ảnh. Tuy nhiên, hạn chế lớn của các phương pháp này là không thể nắm bắt ngữ nghĩa sâu hơn, đặc biệt khi dữ liệu đầu vào phức tạp hoặc không đồng nhất.
* SIFT (Scale-Invariant Feature Transform): Là một thuật toán trích xuất đặc trưng hình ảnh mạnh mẽ, SIFT nhận diện các điểm đặc trưng cục bộ không thay đổi khi hình ảnh bị phóng to, thu nhỏ hoặc xoay. Khi kết hợp với K-Means Clustering, các đặc trưng này có thể được nhóm lại thành từ điển trực quan (visual vocabulary) để biểu diễn ảnh. Dù hiệu quả trên các bộ dữ liệu nhỏ và hình ảnh đơn giản, SIFT không phù hợp với dữ liệu lớn và yêu cầu tính toán cao.

Nhìn chung, phương pháp cổ điển dễ triển khai và phù hợp với dữ liệu ít phức tạp. Tuy nhiên, chúng không tận dụng được khả năng học đặc trưng tự động, điều này làm giảm hiệu quả khi xử lý các bộ dữ liệu hiện đại với ngữ nghĩa phong phú.

### Phương pháp dựa trên học sâu

Học sâu đã cách mạng hóa bài toán tìm kiếm ảnh bằng việc sử dụng các mô hình tiên tiến để học đặc trưng phức tạp và liên kết không gian giữa hình ảnh và văn bản.

* Siamese Networks và Triplet Loss: Siamese Networks học cách đo khoảng cách giữa các vector đặc trưng của hình ảnh và mô tả, trong khi Triplet Loss tối ưu hóa bằng cách giảm khoảng cách giữa các cặp ảnh-mô tả tương ứng và tăng khoảng cách với các cặp không tương ứng. Các phương pháp này hiệu quả trong việc xây dựng không gian nhúng, nhưng yêu cầu bộ dữ liệu lớn và được chú thích tốt.
* Dual Encoder và Cross-Modal Attention: Dual Encoder sử dụng hai mạng nơ-ron riêng biệt để mã hóa hình ảnh và văn bản, sau đó đo sự tương đồng giữa chúng. Cross-Modal Attention cải thiện bằng cách cho phép trao đổi thông tin giữa các modal, giúp mô hình hiểu ngữ cảnh liên quan. Đây là các phương pháp mạnh mẽ nhưng cần tài nguyên tính toán lớn.
* CLIP và ALIGN: Các mô hình tiên tiến như CLIP (Contrastive Language–Image Pretraining) và ALIGN khai thác dữ liệu khổng lồ từ internet để học các nhúng chung giữa văn bản và hình ảnh. CLIP đặc biệt hiệu quả trong việc xử lý các truy vấn chưa từng thấy trước đó (zero-shot), nhưng phụ thuộc vào bộ dữ liệu huấn luyện lớn, đôi khi dẫn đến thiên lệch dữ liệu.
* Vision Transformers kết hợp BERT: Vision Transformers (ViT) học đặc trưng hình ảnh bằng cách áp dụng các cơ chế tự chú ý, trong khi BERT xử lý văn bản ở mức ngữ nghĩa sâu. Khi kết hợp, chúng tạo ra một mô hình mạnh mẽ có thể hiểu và liên kết thông tin từ hai modal, đặc biệt phù hợp với các tác vụ phức tạp.

Các phương pháp học sâu vượt trội về độ chính xác nhưng đòi hỏi tài nguyên lớn, bao gồm GPU mạnh và bộ dữ liệu được chuẩn bị kỹ lưỡng.

### Phương pháp dựa trên học tăng cường, nửa giám sát và chuyển giao

Reward-based Matching sử dụng học tăng cường để cải thiện khả năng liên kết giữa hình ảnh và mô tả. Mô hình học cách điều chỉnh các nhúng sao cho tối đa hóa một hàm thưởng (reward function), phản ánh độ phù hợp giữa hai modal. Phương pháp này đặc biệt hữu ích trong môi trường không có bộ dữ liệu chú thích đầy đủ, vì nó có thể học từ phản hồi của người dùng hoặc các hệ thống tương tác. Tuy nhiên, thiết kế hàm thường phù hợp là một thách thức lớn, và việc tối ưu hóa thường đòi hỏi nhiều thời gian và tài nguyên.

Semi-Supervised Embedding tận dụng cả dữ liệu có nhãn và không nhãn, giúp giảm chi phí gắn nhãn thủ công trong bài toán tìm kiếm ảnh. Phương pháp này thường kết hợp các chiến lược như pseudo-labeling, contrastive learning hoặc consistency regularization để học nhúng phù hợp. Dù hiệu quả cao khi dữ liệu không nhãn chiếm ưu thế, học nửa giám sát phụ thuộc nhiều vào chất lượng của nhãn giả và có thể không đạt hiệu quả tối ưu nếu dữ liệu không nhãn chứa nhiều nhiễu.

Transfer Learning tận dụng các mô hình học sâu đã được huấn luyện trước, chẳng hạn như ResNet, EfficientNet, hoặc CLIP, để trích xuất đặc trưng hình ảnh và văn bản. Sau đó, các lớp đầu ra được tinh chỉnh để phù hợp với bài toán tìm kiếm ảnh. Phương pháp này tiết kiệm thời gian và tài nguyên huấn luyện, đồng thời đạt hiệu suất cao ngay cả trên các bộ dữ liệu nhỏ. Tuy nhiên, chất lượng của mô hình phụ thuộc vào độ phù hợp của mô hình tiền huấn luyện với bài toán cụ thể.

Mỗi phương pháp có ưu và nhược điểm riêng, tùy thuộc vào tính chất dữ liệu và tài nguyên. Trong môi trường thực tiễn, việc kết hợp các phương pháp, chẳng hạn như sử dụng học chuyển giao để khởi tạo mô hình học sâu hoặc học nửa giám sát để mở rộng khả năng huấn luyện, có thể mang lại kết quả tối ưu.

# MỘT SỐ MÔ HÌNH HỌC SÂU CHO BÀI TOÁN TÌM KIẾM ẢNH DỰA TRÊN MÔ TẢ

## Một số mô hình vector hoá văn bản

### Word2Vec

#### Tổng Quan Về Word2Vec

Word2Vec là một phương pháp học sâu không giám sát, được thiết kế để biểu diễn từ ngữ dưới dạng các vector số học mang ý nghĩa. Kỹ thuật này được phát triển bởi Tomas Mikolov và các cộng sự tại Google vào năm 2013, với mục tiêu chuyển đổi từ ngữ thành các vector trong không gian số, nơi các mối quan hệ ngữ nghĩa và cú pháp giữa các từ được bảo toàn một cách hiệu quả.

Nguyên Lý Hoạt Động

Word2Vec dựa trên giả thuyết phân phối trong ngôn ngữ học, vốn cho rằng:

"*Ý nghĩa của một từ được xác định bởi ngữ cảnh xung quanh nó."*

Điều này có nghĩa là các từ thường xuyên xuất hiện trong cùng một ngữ cảnh có xu hướng mang ý nghĩa tương tự.

Word2Vec hoạt động bằng cách huấn luyện một mô hình không giám sát trên tập văn bản lớn. Mục tiêu của mô hình là tối đa hóa khả năng dự đoán mối quan hệ giữa các từ, từ đó tạo ra các vector từ (word embeddings) thể hiện:

* Quan hệ ngữ nghĩa: Ví dụ, vector của từ "king" có thể gần với "queen" hơn là "car".
* Quan hệ cú pháp: Các phép tính vector như "king" - "man" + "woman" ≈ "queen" là minh chứng cho khả năng mô hình hóa các mối quan hệ này.

Ý Nghĩa

Word2Vec đã mở ra một cách tiếp cận mới trong xử lý ngôn ngữ tự nhiên (NLP), giúp các ứng dụng như tìm kiếm thông tin, dịch máy, phân tích cảm xúc, và nhiều lĩnh vực khác đạt được hiệu quả vượt bậc. Từ khóa quan trọng trong thành công của Word2Vec là khả năng biểu diễn từ dưới dạng số liệu nhưng vẫn duy trì mối quan hệ ngữ nghĩa sâu sắc giữa các từ.

#### Hai kiến trúc chính của Word2Vec

Word2Vec bao gồm hai kiến trúc chính là Skip-gram và Continuous Bag of Words (CBOW). Cả hai đều sử dụng các phương pháp huấn luyện khác nhau để học vector embedding.

1. Skip-gram

* Mục tiêu: Mục tiêu của Skip-gram là dự đoán các từ ngữ cảnh (context words) dựa trên từ trung tâm (center word). Điều này cho phép mô hình học được mối quan hệ ngữ nghĩa giữa các từ trong văn bản.
* Đầu vào và đầu ra: Trong mô hình Skip-gram, từ mục tiêu (target word) sẽ là đầu vào, trong khi các từ ngữ cảnh (context words) xung quanh từ mục tiêu sẽ là đầu ra. Ví dụ, nếu từ mục tiêu là "yêu", mô hình sẽ cố gắng dự đoán các từ như "Anh ấy", "Bóng đá" trong một khoảng cách nhất định xung quanh từ đó.
* Cơ chế hoạt động:
* Mô hình Skip-gram là một phần quan trọng trong phương pháp Word2Vec, được sử dụng để tạo ra các vector từ (word vectors) từ một tập dữ liệu văn bản lớn. Mô hình này hoạt động dựa trên nguyên tắc rằng mỗi từ có thể được sử dụng để dự đoán các từ ngữ cảnh xung quanh nó trong một câu.
* Mô hình cố gắng tối đa hóa xác suất xuất hiện của các từ ngữ cảnh trong một cửa sổ ngữ cảnh cố định quanh từ trung tâm. Mô hình Skip-gram thường được triển khai dưới dạng một mạng nơ-ron hai lớp với một tầng ẩn không có hàm kích hoạt. Đầu vào là một vector one-hot đại diện cho từ mục tiêu, và đầu ra là xác suất cho mỗi từ trong từ vựng về việc trở thành một từ ngữ cảnh.

Hàm mục tiêu trong Skip-gram được biểu diễn như sau:

A black text on a white background

Description automatically generated

Trong đó:

* T: Số lượng từ trong tập văn bản.
* c: Kích thước cửa sổ ngữ cảnh.
* : Từ trung tâm tại thời điểm t.

A diagram of a network

Description automatically generated

Hình 2.1: *Mô hình Skip-gram*

1. Continuous Bag of Words (CBOW)

* Mục tiêu: Mục tiêu của mô hình này là dự đoán từ mục tiêu (target word) dựa trên các từ ngữ cảnh (context words) xung quanh nó.
* Đầu vào và đầu ra của mô hình: Phương pháp này lấy đầu vào là một hoặc nhiều từ context word và cố gắng dự đoán output từ đầu ra (target word) thông qua một tầng neural đơn giản . Nhờ việc đánh giá output error với target word ở dạng one-hot, mô hình có thể điều chỉnh weight, học được vector biểu diễn cho target word.
* Cơ chế:
* Mô hình sử dụng một tầng nơ-ron đơn giản để xử lý đầu vào. Các từ ngữ cảnh được chuyển đổi thành vector one-hot, sau đó được đưa qua tầng ẩn để tính toán xác suất cho từ mục tiêu thông qua hàm softmax.
* Thường hoạt động tốt với tập dữ liệu lớn và có khả năng tạo ra các vector chất lượng cao cho các từ phổ biến.
* Thường nhanh hơn Skip-gram do xử lý trên toàn bộ cửa sổ ngữ cảnh một cách đồng thời.

Hàm mục tiêu trong CBOW:

A black text on a white background

Description automatically generated

Trong đó:

* : là xác suất có điều kiện, trong đó từ trung tâm ​ được dự đoán dựa trên 2c từ ngữ cảnh xung quanh nó: từ ​ đến ​, ngoại trừ

A diagram of a machine

Description automatically generated

Hình 2.2: *Mô hình CBOW*

#### Kiến Trúc Mạng Nơ-ron Trong Word2Vec

Word2Vec sử dụng một mạng nơ-ron đơn giản nhưng hiệu quả gồm ba lớp:

Lớp đầu vào (Input layer)

* Mỗi từ trong từ điển được biểu diễn bằng một vector one-hot. Điều này có nghĩa là mỗi từ sẽ được mã hóa thành một vector với kích thước bằng số lượng từ trong từ điển, trong đó chỉ có một phần tử là 1 (tương ứng với vị trí của từ) và tất cả các phần tử còn lại là 0.
  + Ví dụ: Nếu từ điển có 5 từ [a,b,c,d,e][a, b, c, d, e][a,b,c,d,e], từ ccc được biểu diễn là [0,0,1,0,0][0, 0, 1, 0, 0][0,0,1,0,0].
* Trong CBOW, lớp input sẽ nhận vào nhiều từ ngữ cảnh xung quanh từ mục tiêu. Kích thước của cửa sổ ngữ cảnh (context window size) xác định số lượng từ mà mô hình sẽ xem xét để dự đoán từ mục tiêu. Ví dụ, nếu kích thước cửa sổ là 2, thì hai từ trước và hai từ sau sẽ được sử dụng làm input.
* Các vector one-hot của các từ ngữ cảnh sẽ được kết hợp lại để tạo thành một vector đầu vào duy nhất. Vector này thường được tính trung bình hoặc tổng hợp để tạo ra một đại diện cho ngữ cảnh tổng thể.

Lớp Ẩn (Embedding Layer)

* Word2Vec sử dụng một mạng nơ ron đơn giản với chỉ một lớp ẩn giữa lớp đầu vào và lớp đầu ra. Điều này giúp giảm thiểu độ phức tạp của mô hình và tăng tốc độ huấn luyện.
* Lớp ẩn được đại diện bởi một ma trận trọng số, nơi mỗi hàng tương ứng với một từ trong từ điển. Kích thước của ma trận này phụ thuộc vào số lượng từ trong từ điển và số lượng đặc trưng (features) mà bạn muốn học cho mỗi từ. Ví dụ, nếu có 10.000 từ trong từ điển và bạn chọn 300 đặc trưng, ma trận trọng số sẽ có kích thước 10.000×300. Kích thước lớp ẩn thường nhỏ hơn kích thước từ điển, giúp giảm độ phức tạp tính toán.
* Một điểm đặc biệt của lớp ẩn trong Word2Vec là không sử dụng hàm kích hoạt (activation function). Điều này có nghĩa là đầu ra của lớp ẩn là một phép nhân đơn giản giữa đầu vào và ma trận trọng số. Việc không sử dụng hàm kích hoạt giúp giữ cho quá trình tính toán đơn giản và hiệu quả hơn.

Lớp đầu ra (Output layer)

* Lớp output có kích thước bằng số lượng từ trong từ điển (vocabulary size). Mỗi nút trong lớp này tương ứng với một từ trong từ điển, cho phép mô hình dự đoán xác suất cho từng từ.
* Để tính toán xác suất cho từng từ, Word2Vec sử dụng hàm softmax. Hàm này chuyển đổi các giá trị đầu ra thành một phân phối xác suất, đảm bảo rằng tổng xác suất của tất cả các từ là 1. Công thức cho hàm softmax như sau:

A mathematical equation with black text

Description automatically generated with medium confidence

Trong đó:

* : Vector embedding của từ input.
* : Vector embedding của từ output.

#### Kỹ Thuật Tối Ưu

Để giảm độ phức tạp và cải thiện hiệu suất, Word2Vec sử dụng các kỹ thuật tối ưu hóa sau:

* Negative Sampling: Để giảm độ phức tạp tính toán khi số lượng từ trong từ điển lớn, Word2Vec thường sử dụng kỹ thuật negative sampling thay vì tính toán softmax cho tất cả các từ, mô hình chỉ chọn một số lượng nhỏ các từ âm (negative samples) để cập nhật trọng số. Điều này giúp tăng tốc quá trình huấn luyện mà vẫn duy trì chất lượng của các vector từ.
* Hierarchical Softmax: Một phương pháp khác để tối ưu hóa lớp output, nơi tổ chức các từ trong một cấu trúc cây nhị phân. Điều này giúp giảm độ phức tạp tính toán từ *O*(*V*) xuống *O*(log*V*), làm cho việc tính toán xác suất trở nên hiệu quả hơn.

Giảm Thiểu Phức Tạp Tính Toán

Với Negative Sampling và Hierarchical Softmax, Word2Vec có thể huấn luyện nhanh chóng trên các tập dữ liệu lớn, đảm bảo tính hiệu quả và độ chính xác.

Word2Vec đã đặt nền tảng cho các phương pháp embedding hiện đại trong xử lý ngôn ngữ tự nhiên. Mặc dù đã có nhiều phương pháp tiên tiến hơn như GloVe, FastText hay các mô hình ngôn ngữ dựa trên Transformer (BERT, GPT), Word2Vec vẫn được xem là cột mốc quan trọng và là công cụ mạnh mẽ trong các bài toán yêu cầu biểu diễn vector của từ.

### Transformer

#### Tổng quan về Transformer

Transformer là một kiến trúc mô hình học sâu đã cách mạng hóa lĩnh vực xử lý ngôn ngữ tự nhiên (NLP) kể từ khi được giới thiệu trong bài báo "Attention is All You Need" vào năm 2017. Kiến trúc này sử dụng cơ chế self-attention, cho phép mô hình hiểu và xử lý mối quan hệ giữa các từ trong một câu mà không cần đến cấu trúc tuần tự như các mô hình trước đây như RNN hay LSTM.

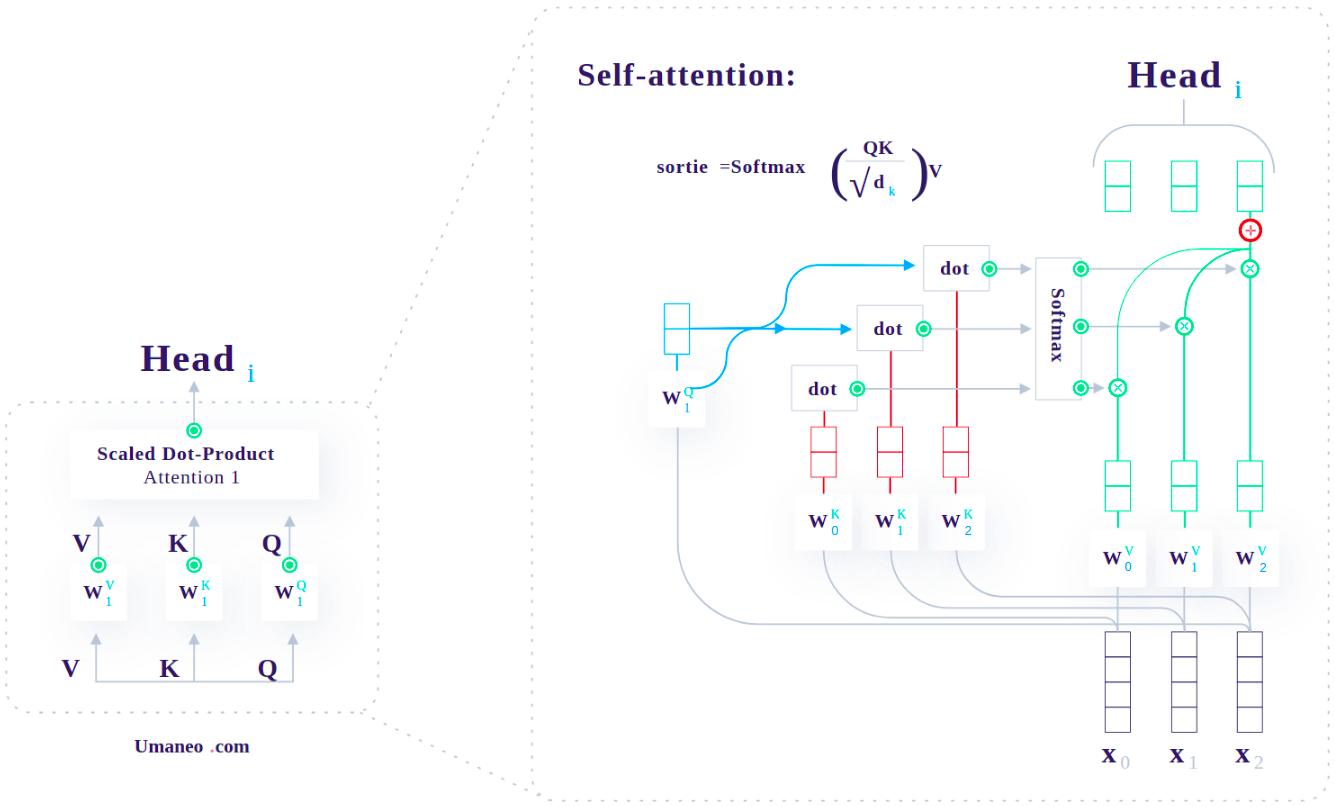
#### Cấu trúc của Transformer

Transformer được chia thành hai phần chính:

* Encoder: Chịu trách nhiệm tiếp nhận dữ liệu đầu vào và chuyển đổi chúng thành một biểu diễn ngữ nghĩa. Encoder thường bao gồm nhiều lớp (thường là 6 lớp), mỗi lớp chứa các thành phần như self-attention và feed-forward neural networks.
* Self-attention: Giúp mô hình xem xét tất cả các từ trong câu để xác định mối quan hệ giữa chúng.
* Feed-forward neural network: Xử lý thông tin sau khi đã được mã hóa.
* Decoder: Nhận đầu vào từ encoder và một chuỗi đầu vào khác để tạo ra chuỗi đầu ra cuối cùng. Decoder cũng bao gồm nhiều lớp tương tự như encoder, nhưng có thêm cơ chế attention để chú ý đến các phần quan trọng của đầu vào.

#### Cơ chế hoạt động

* Cơ chế self-attention cho phép mô hình đánh giá tầm quan trọng của từng từ trong câu so với các từ khác. Quá trình này bao gồm ba bước chính:
* Tạo các vector: Mỗi từ trong câu được chuyển đổi thành ba vector:
* Query (Q): Đặt câu hỏi về tầm quan trọng của từ.
* Key (K): Đại diện cho thông tin của từ.
* Value (V): Thông tin sẽ được sử dụng để tính toán đầu ra.
* Tính toán attention scores: Sử dụng các vector Q và K để tính toán độ tương đồng giữa các từ, từ đó xác định mức độ ảnh hưởng của từng từ đến từ hiện tại.
* Tính toán weighted sum: Kết hợp thông tin từ các vector V dựa trên scores đã tính toán, tạo ra một vector đầu ra cho mỗi từ.



Hình 2.3: *Cấu trúc Seft-attention trong mô hình Transformer*

* Để cải thiện khả năng tập trung vào nhiều khía cạnh khác nhau của dữ liệu, Transformer sử dụng multi-head attention. Điều này cho phép mô hình thực hiện nhiều phép toán self-attention song song, mỗi phép toán có thể tập trung vào các phần khác nhau của chuỗi dữ liệu.

A diagram of a process

Description automatically generated

Hình 2.4: Cấu trúc *Multi-Head attention*

### BERT

#### Tổng quan về BERT

BERT (Bidirectional Encoder Representations from Transformers) là một mô hình ngôn ngữ tiên tiến được phát triển bởi nhóm nghiên cứu của Google AI vào năm 2018. BERT mang tính cách mạng nhờ khả năng xử lý ngôn ngữ tự nhiên (NLP) theo hướng đa chiều, học được ngữ cảnh của các từ không chỉ từ trái sang phải mà còn từ phải sang trái. Đây là một bước tiến vượt bậc so với các mô hình ngôn ngữ truyền thống một chiều như GPT (Generative Pre-trained Transformer).

Nguyên Lý Cơ Bản

* Đa hướng: Khác với các mô hình một chiều, BERT sử dụng phương pháp mã hóa hai chiều, giúp nó hiểu rõ hơn về mối quan hệ giữa các từ trong câu. Tức là xem xét toàn bộ ngữ cảnh xung quanh một từ thay vì chỉ dựa vào các từ trước hoặc sau nó.
* Học biểu diễn ngữ cảnh: Biểu diễn vector của từ trong BERT thay đổi linh hoạt theo ngữ cảnh, giúp mô hình nắm bắt ý nghĩa từ một cách chính xác hơn trong các ngữ cảnh khác nhau.

#### Kiến trúc chi tiết của BERT

Cấu trúc tổng thể:

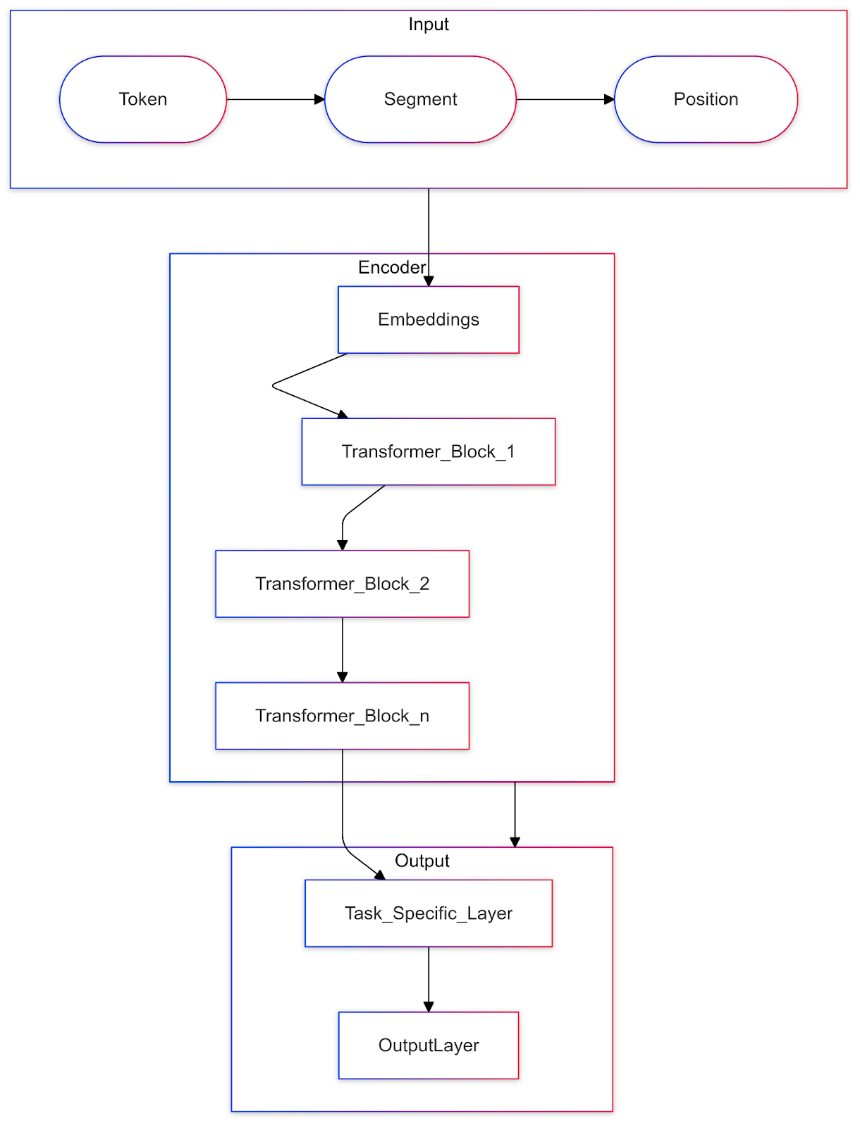
BERT sử dụng kiến trúc Transformer, cụ thể là Bidirectional Transformer Encoder. Kiến trúc này bao gồm nhiều lớp encoder, mỗi lớp có khả năng thực hiện self-attention để học mối quan hệ giữa các từ trong câu.

Số lớp (L): BERT có hai phiên bản chính:

* BERT\_BASE: 12 lớp, kích thước embedding 768, 12 đầu attention, tổng số tham số khoảng 110 triệu.
* BERT\_LARGE: 24 lớp, kích thước embedding 1024, 16 đầu attention, tổng số tham số khoảng 340 triệu

Các thành phần chính:

* Embedding Layer: Đầu vào của mô hình là các token được chuyển đổi thành các vector thông qua ba loại embedding:
* Token Embeddings: Biểu diễn các từ trong câu.
* Segment Embeddings: Phân biệt các câu khác nhau trong một cặp câu (A và B).
* Position Embeddings: Thêm thông tin về vị trí của các từ trong câu để mô hình có thể hiểu thứ tự của chúng.
* Self-Attention Mechanism: Mỗi lớp encoder sử dụng cơ chế attention để tính toán mức độ quan trọng giữa các từ trong câu. Điều này cho phép mô hình học được ngữ cảnh đầy đủ từ cả hai phía.



Hình 2.5: *Kiến trúc của mô hình BERT*

## Một số mô hình vector hóa hình ảnh

### Convolutional Neural Networks (CNNs)

Mạng CNN (Convolutional Neural Network) là một loại mạng nơ-ron nhân tạo được sử dụng rộng rãi trong lĩnh vực xử lý ảnh và thị giác máy tính. Nó được thiết kế để tự động học và phân tích các đặc trưng từ dữ liệu hình ảnh một cách hiệu quả. Cấu trúc của mạng CNN được lấy cảm hứng từ cách mà thị giác của con người hoạt động. Mạng này bao gồm các lớp tích chập (convolutional layers) và các lớp tổng hợp (pooling layers), kết hợp với các lớp kết nối đầy đủ (fully connected layers) ở cuối để thực hiện phân loại hoặc dự đoán [20].

A diagram of a diagram of a network

Description automatically generated

Hình 2.6: Minh họa cấu trúc mạng CNN.

Các lớp tích chập trong mạng CNN thực hiện các phép tích chập giữa các bộ lọc (filters) và các vùng của ảnh đầu vào. Bằng cách áp dụng các bộ lọc này, mạng CNN có khả năng tự động trích xuất các đặc trưng như cạnh, góc, texture, hoặc các đặc trưng phức tạp hơn từ ảnh. Các lớp tổng hợp được sử dụng để giảm kích thước không gian của đặc trưng, đồng thời giữ lại thông tin quan trọng. Điều này giúp giảm số lượng tham số và tính toán của mạng, đồng thời giúp ngăn chặn overfitting.

Sau khi qua các lớp tích chập và lớp tổng hợp, các đặc trưng được đưa vào các lớp kết nối đầy đủ để thực hiện phân loại hoặc dự đoán. Cuối cùng, mạng CNN tạo ra đầu ra là một phân phối xác suất trên các lớp đối tượng mà nó được huấn luyện để nhận dạng. Mạng CNN đã đạt được những thành tựu đáng kể trong các ứng dụng như nhận dạng đối tượng, phân loại ảnh, nhận dạng khuôn mặt, xe tự lái và nhiều lĩnh vực khác.

Các kiến trúc CNN phổ biến

LeNet-5

* là một trong những kiến trúc mạng nơ-ron tích chập (CNN) đầu tiên và có ảnh hưởng lớn trong lĩnh vực nhận diện hình ảnh, đặc biệt là trong nhận diện chữ viết tay. Được giới thiệu bởi Yann LeCun và các cộng sự vào năm 1998, kiến trúc này đã thiết lập nền tảng cho nhiều phát triển sau này trong học sâu. Mạng này có khả năng học các đặc điểm cục bộ từ dữ liệu mà không cần phải can thiệp thủ công, điều này giúp tăng cường hiệu suất phân loại.
* Cấu trúc của LeNet-5: bao gồm 7 lớp chính, được chia thành hai phần: khối các lớp tích chập và khối các lớp kết nối đầy đủ.
* Đặc điểm nổi bật:
* LeNet-5 sử dụng các lớp convolutional để tự động học và trích xuất các đặc trưng từ hình ảnh mà không cần can thiệp thủ công, giúp cải thiện độ chính xác trong việc phân loại.
* Việc sử dụng các bộ lọc chia sẻ (shared weights) trong các lớp convolutional giúp giảm thiểu số lượng tham số cần thiết cho việc học, từ đó tối ưu hóa quá trình tính toán.
* Lớp pooling giúp giảm kích thước dữ liệu đầu ra từ lớp convolutional, giữ lại các đặc điểm quan trọng và loại bỏ thông tin không cần thiết, làm tăng tốc độ tính toán và giảm overfitting.
* LeNet-5 đã đạt được độ chính xác cao trên tập dữ liệu MNIST, với tỷ lệ chính xác lên đến khoảng 99%, cho thấy hiệu quả của nó trong việc nhận diện ký tự viết tay.

AlexNet

* Là một trong những kiến trúc mạng nơ-ron tích chập (CNN) quan trọng nhất trong lịch sử học sâu, được phát triển bởi Alex Krizhevsky, Ilya Sutskever và Geoffrey Hinton. Được giới thiệu vào năm 2012, AlexNet đã giành chiến thắng trong cuộc thi ImageNet Large Scale Visual Recognition Challenge (ILSVRC) với một khoảng cách lớn, đánh dấu sự chuyển mình của lĩnh vực thị giác máy tính.
* Cấu trúc: bao gồm 8 lớp chính, trong đó có 5 lớp convolutional và 3 lớp fully connected.
* Đặc điểm nổi bật:
* AlexNet sử dụng hàm kích hoạt ReLU (Rectified Linear Unit), thay vì sigmoid hoặc tanh, giúp tăng tốc độ huấn luyện và cải thiện hiệu suất mạng.
* áp dụng kỹ thuật dropout trong các lớp fully connected để ngăn ngừa overfitting, giúp mô hình trở nên mạnh mẽ hơn.
* Local Response Normalization - một kỹ thuật chuẩn hóa được sử dụng để tăng cường khả năng tổng quát của mạng.
* AlexNet là một trong những mô hình đầu tiên tận dụng sức mạnh tính toán của GPU, cho phép huấn luyện nhanh chóng trên dữ liệu lớn.
* Trong cuộc thi ILSVRC năm 2012, AlexNet đạt tỷ lệ lỗi chỉ khoảng 15.3%, trong khi mô hình đứng thứ hai có tỷ lệ lỗi lên tới khoảng 26.2%, cho thấy sự vượt trội rõ rệt.

VGGNet

* VGGNet được thiết kế để cải thiện khả năng nhận diện hình ảnh bằng cách sử dụng các lớp tích chập với kích thước nhỏ (3x3) và nhiều lớp hơn. Mục tiêu chính của VGGNet là khám phá ảnh hưởng của độ sâu của mạng đến hiệu suất phân loại hình ảnh. Được phát triển bởi nhóm nghiên cứu tại Đại học Oxford gồm Karen Simonyan và Andrew Zisserman vào năm 2014. Kiến trúc này đã đạt được thành công lớn trong cuộc thi ImageNet và đã tạo ra ảnh hưởng sâu rộng đến các mô hình học sâu sau này.
* Cấu trúc: VGGNet có nhiều biến thể, phổ biến nhất là VGG-16 và VGG-19, bao gồm Khối Convolutional, Lớp Max Pooling, Lớp Fully Connected.
* Đặc điểm nổi bật:
* VGGNet chỉ sử dụng bộ lọc kích thước 3×33×3 cho tất cả các lớp convolutional, điều này giúp giảm số lượng tham số so với việc sử dụng bộ lọc lớn hơn mà vẫn giữ được khả năng trích xuất đặc trưng mạnh mẽ.
* Kiến trúc này cho thấy rằng việc tăng cường độ sâu của mạng (số lượng lớp) có thể cải thiện đáng kể độ chính xác trong nhận diện hình ảnh. VGG-16 có tổng cộng 16 lớp học được (13 lớp convolutional và 3 lớp fully connected), trong khi VGG-19 có tới 19 lớp học được.
* Việc tổ chức các khối convolutional giúp đơn giản hóa quá trình xây dựng mạng và dễ dàng mở rộng cho các kiến trúc phức tạp hơn.
* VGGNet có một số lượng tham số rất lớn, lên tới khoảng 138 triệu tham số cho VGG-16, điều này giúp mạng học được nhiều đặc trưng phức tạp từ dữ liệu.

GoogLeNet (Inception)

* GoogLeNet, còn được gọi là Inception v1, là một kiến trúc mạng nơ-ron tích chập (CNN) tiên tiến được phát triển bởi nhóm nghiên cứu tại Google. Được giới thiệu trong cuộc thi ImageNet Large Scale Visual Recognition Challenge (ILSVRC) năm 2014, GoogLeNet đã đạt được thành công lớn với độ chính xác cao trong phân loại hình ảnh. Kiến trúc này giải quyết các vấn đề về hiệu suất và tính toán trong mạng nơ-ron sâu, cho phép mạng học các đặc trưng ở nhiều quy mô khác nhau mà không làm tăng số lượng tham số quá lớn. Điều này giúp giảm thiểu vấn đề overfitting và cải thiện khả năng tổng quát của mô hình.
* Cấu trúc: bao gồm 22 lớp (27 lớp nếu tính cả các lớp gộp): Lớp đầu vào, Lớp Convolutional đầu tiên, Lớp Max Pooling, Khối Inception, Lớp Average Pooling, Lớp Fully Connected.
* Đặc điểm nổi bật:
* Khối Inception cho phép mạng học các đặc trưng từ nhiều kích thước khác nhau đồng thời, giúp cải thiện khả năng nhận diện và phân loại hình ảnh. Mỗi khối có thể chứa các bộ lọc với kích thước khác nhau, từ 1×1 đến 5×5.
* Việc sử dụng bộ lọc 1×1 trước các bộ lọc lớn hơn giúp giảm số lượng tham số cần thiết mà vẫn duy trì hiệu suất cao. GoogLeNet chỉ có khoảng 5 triệu tham số, ít hơn nhiều so với VGG-16.
* Kiến trúc này sử dụng các lớp gộp và pooling để giảm kích thước đầu vào và giữ lại thông tin quan trọng, từ đó ngăn ngừa hiện tượng overfitting.
* GoogLeNet đã giành chiến thắng trong cuộc thi ILSVRC 2014 với tỷ lệ lỗi chỉ khoảng 6.67%, cho thấy hiệu quả vượt trội của nó trong việc phân loại hình ảnh mà không cần đào tạo trên dữ liệu bên ngoài.

ResNet

* Là một kiến trúc mạng nơ-ron tích chập (CNN) được phát triển bởi Kaiming He và các cộng sự tại Microsoft Research, được giới thiệu vào năm 2015. ResNet đã đạt giải thưởng trong cuộc thi ImageNet Large Scale Visual Recognition Challenge (ILSVRC) và trở thành một trong những kiến trúc phổ biến nhất trong lĩnh vực thị giác máy tính.

ResNet được thiết kế để giải quyết vấn đề vanishing gradient, một vấn đề thường gặp khi huấn luyện các mạng nơ-ron sâu. Kiến trúc này sử dụng khái niệm "skip connections" (kết nối tắt), cho phép thông tin từ các lớp trước đó được truyền thẳng đến các lớp sau mà không bị suy yếu. Điều này giúp cải thiện khả năng huấn luyện mạng nơ-ron với độ sâu lớn mà không làm giảm hiệu suất.

* Cấu trúc: bao gồm nhiều khối Residual Block, mỗi khối có thể chứa hai hoặc ba lớp convolutional. Một số biến thể nổi bật của ResNet bao gồm ResNet-18, ResNet-34, ResNet-50, ResNet-101 và ResNet-152, với số lượng lớp khác nhau.
* Đặc điểm nổi bật:
* Việc sử dụng kết nối tắt giúp cải thiện khả năng truyền thông tin qua các lớp, cho phép mạng học hiệu quả hơn khi độ sâu tăng lên mà không bị ảnh hưởng bởi vanishing gradient.
* ResNet cho phép xây dựng các mạng nơ-ron với độ sâu lớn (hàng nghìn lớp) mà vẫn duy trì hiệu suất cao. Điều này đã mở ra khả năng nghiên cứu và phát triển các mô hình phức tạp hơn trong học sâu.
* ResNet đã đạt được độ chính xác cao trong nhiều bài toán phân loại hình ảnh và trở thành nền tảng cho nhiều ứng dụng khác nhau như nhận diện khuôn mặt, phát hiện đối tượng và phân loại hình ảnh.
* Có nhiều biến thể của ResNet với số lượng lớp khác nhau, từ ResNet-18 đơn giản đến ResNet-152 phức tạp hơn, cho phép người dùng lựa chọn kiến trúc phù hợp với yêu cầu cụ thể của bài toán.

DenseNet

* DenseNet (Densely Connected Convolutional Networks) được phát triển bởi Gao Huang và các cộng sự vào năm 2016. Kiến trúc này nổi bật với cách thức kết nối giữa các lớp, giúp cải thiện khả năng học tập và giảm thiểu vấn đề vanishing gradient. Giải quyết các vấn đề liên quan đến việc huấn luyện các mạng nơ-ron sâu bằng cách sử dụng các kết nối dày đặc giữa các lớp. Mỗi lớp trong DenseNet nhận đầu vào không chỉ từ lớp trước đó mà còn từ tất cả các lớp trước đó, tạo ra một mạng lưới kết nối phong phú giúp tối ưu hóa quá trình học.
* Cấu trúc: bao gồm nhiều khối Dense Block và Transition Layer. Đầu vào là hình ảnh, nhiều Dense Block xen kẽ với Transition Layer, lớp Fully Connected cuối cùng để thực hiện phân loại.
* Đặc điểm nổi bật:
* Mỗi lớp trong DenseNet nhận đầu vào từ tất cả các lớp trước đó, giúp duy trì thông tin và cải thiện khả năng học tập. Điều này cũng làm giảm thiểu vấn đề vanishing gradient.
* Mặc dù có nhiều kết nối, DenseNet sử dụng ít tham số hơn so với các kiến trúc khác như ResNet nhờ vào việc sử dụng bộ lọc 1×1 trong Transition Layer, giúp giảm chiều dữ liệu mà vẫn giữ lại thông tin quan trọng.
* Nhờ vào cấu trúc kết nối dày đặc và việc sử dụng dropout trong mỗi khối, DenseNet có khả năng chống lại hiện tượng overfitting rất hiệu quả.
* DenseNet đã đạt được độ chính xác cao trên nhiều bài toán phân loại hình ảnh, bao gồm cả tập dữ liệu ImageNet, cho thấy tính hiệu quả của nó trong việc nhận diện đối tượng.

Cách thức hoạt động của CNN trong vector hoá hình ảnh

* Tiền xử lý hình ảnh: Hình ảnh đầu vào cần được chuẩn bị trước khi đưa vào mạng CNN. Bao gồm thay đổi kích thước hình ảnh về một kích thước cố định, thường là 224x224 pixel, để đảm bảo tính nhất quán. Sau đó, các giá trị pixel được chuẩn hóa, đảm bảo chúng nằm trong một phạm vi cụ thể (ví dụ: từ 0 đến 1) hoặc theo phân phối trung bình và độ lệch chuẩn mà mạng đã được huấn luyện. Cuối cùng được chuyển đổi thành tensor – một dạng dữ liệu có thể xử lý trực tiếp trong mô hình học sâu.
* Trích xuất đặc trưng qua các lớp Convolution: Các lớp convolution trong CNN được thiết kế để hoạt động như các bộ lọc, trích xuất các đặc trưng quan trọng từ hình ảnh. Ở các lớp đầu tiên, mạng học các đặc trưng cơ bản như cạnh, góc, hoặc họa tiết đơn giản. Khi đi sâu hơn vào các lớp tiếp theo, mạng bắt đầu nhận diện các đặc trưng phức tạp hơn như hình dạng hoặc cấu trúc của các đối tượng trong hình ảnh. Kết quả là một tập hợp các biểu diễn đặc trưng cục bộ được lưu giữ trong các ma trận dữ liệu.
* Biểu diễn dạng Tensor: Sau khi qua các lớp convolution và pooling (các lớp làm giảm kích thước dữ liệu mà vẫn giữ lại các đặc trưng quan trọng), hình ảnh được biểu diễn dưới dạng tensor. Tensor này lưu trữ thông tin về đặc trưng đã được trích xuất ở nhiều cấp độ khác nhau. Đây là bước chuyển đổi hình ảnh thành dạng toán học có thể sử dụng để phân tích hoặc áp dụng vào các tác vụ cụ thể.
* Biểu diễn Vector: Từ tensor đầu ra, CNN sẽ sử dụng các lớp fully connected (dense layers) để chuyển đổi dữ liệu này thành một vector nhúng cuối cùng. Vector nhúng này thường có kích thước cố định (chẳng hạn 512, 1024 hoặc 2048 chiều), đại diện cho hình ảnh trong không gian vector hóa. Vector này chứa thông tin nén về toàn bộ nội dung của hình ảnh và có thể được sử dụng trong các tác vụ như tìm kiếm hình ảnh, nhận diện, hoặc phân loại.

### Vision Transformers (ViT)

Vision Transformer (ViT) là một mô hình học sâu được phát triển để xử lý các nhiệm vụ liên quan đến hình ảnh, dựa trên kiến trúc transformer, vốn trước đây chủ yếu được sử dụng trong xử lý ngôn ngữ tự nhiên. ViT đã được giới thiệu vào năm 2021 và nhanh chóng trở thành một trong những mô hình hàng đầu trong lĩnh vực thị giác máy tính.

Cấu trúc và Cách hoạt động

ViT hoạt động bằng cách chia nhỏ hình ảnh thành các phần nhỏ hơn gọi là patches. Mỗi patch sau đó được biến đổi thành một vector nhúng thông qua một lớp tuyến tính. Các vector này, cùng với thông tin vị trí (positional embeddings), được đưa vào một mô hình transformer để học các đặc trưng quan trọng từ hình ảnh.

* Chia nhỏ hình ảnh: Hình ảnh đầu vào có thể được chia thành các patch có kích thước cố định, ví dụ như 16×16 pixel. Điều này giúp mô hình dễ dàng xử lý hình ảnh ở các độ phân giải khác nhau và không cần phải lo lắng về cấu trúc không gian.
* Embedding và thông tin vị trí: Mỗi patch sau khi được chia nhỏ sẽ được chuyển đổi thành một vector nhúng có chiều cố định, thường là 768 chiều. Thông tin về vị trí của các patch được mã hóa và thêm vào vector nhúng để giữ lại thông tin không gian giữa các patch.
* Mô hình Transformer: Các vector nhúng và thông tin vị trí sau đó được đưa vào các lớp encoder của mô hình transformer. Tại đây, các cơ chế attention sẽ giúp mô hình học các mối quan hệ giữa các patch, từ đó trích xuất các đặc trưng quan trọng từ toàn bộ hình ảnh.

ViT đã chứng tỏ được khả năng vượt trội trong nhiều nhiệm vụ xử lý hình ảnh, bao gồm nhận dạng, phân loại, và phân đoạn hình ảnh. Với cấu trúc transformer mạnh mẽ, ViT có thể học cách hiểu các mối quan hệ phức tạp trong hình ảnh và trích xuất các đặc trưng có ý nghĩa từ chúng.

Ưu điểm và nhược điểm

Ưu điểm:

* Khả năng học mối quan hệ toàn cục: ViT không giới hạn trong việc học các đặc trưng cục bộ như CNN, mà có thể học mối quan hệ phức tạp giữa các patch trong hình ảnh, cho phép trích xuất các đặc trưng sâu sắc hơn.
* Hiệu quả với dữ liệu lớn: ViT hoạt động rất hiệu quả khi được huấn luyện trên các tập dữ liệu lớn. Với khả năng học các mối quan hệ toàn cục, ViT thường có độ chính xác vượt trội trong các tác vụ phân loại và nhận diện hình ảnh so với các mô hình truyền thống.
* Linh hoạt và dễ dàng mở rộng: ViT có thể được áp dụng cho nhiều tác vụ khác nhau trong thị giác máy tính, từ phân loại, nhận diện đối tượng đến phân đoạn hình ảnh, và có thể dễ dàng tích hợp với các mô hình khác.

Nhược điểm:

* Yêu cầu dữ liệu lớn: ViT đòi hỏi một lượng dữ liệu huấn luyện lớn để có thể học được các đặc trưng hiệu quả. Nếu không có đủ dữ liệu, mô hình có thể gặp khó khăn trong việc hội tụ và dễ bị overfitting.
* Tốn tài nguyên tính toán: ViT yêu cầu nhiều bộ nhớ và tài nguyên tính toán hơn so với các mô hình CNN, đặc biệt là khi xử lý các ảnh có kích thước lớn và số lượng patch cao.
* Khó khăn trong việc tối ưu trên dữ liệu nhỏ: ViT có thể không hiệu quả khi được áp dụng cho các tập dữ liệu nhỏ vì sự phức tạp của mô hình.

### Hybrid Models (CLIP, ALIGN)

Trong những năm gần đây, các mô hình học sâu đa phương thức (multi-modal learning) đã thu hút sự chú ý lớn trong cộng đồng nghiên cứu, đặc biệt là trong các ứng dụng liên quan đến hình ảnh và văn bản. Các mô hình hybrid như CLIP (Contrastive Language-Image Pretraining) và ALIGN (A Large-scale ImaGe and Noisy-text embedding) đã mở ra một hướng đi mới trong việc kết hợp thông tin từ cả hình ảnh và văn bản, giúp cải thiện khả năng hiểu và tìm kiếm hình ảnh.

Hybrid models như CLIP và ALIGN kết hợp khả năng xử lý hình ảnh và văn bản để học các biểu diễn nhúng (embeddings) chung. Điều này giúp các mô hình học được mối quan hệ giữa hình ảnh và văn bản, từ đó có thể thực hiện các tác vụ như tìm kiếm hình ảnh dựa trên văn bản mô tả hoặc sinh mô tả văn bản từ hình ảnh.

* CLIP, được phát triển bởi OpenAI, là một mô hình học sâu đa phương thức, được huấn luyện trên một lượng lớn dữ liệu hình ảnh và văn bản, không có sự phân biệt giữa chúng. Mô hình này học cách ánh xạ hình ảnh và văn bản vào cùng một không gian nhúng, cho phép thực hiện các tác vụ như tìm kiếm hình ảnh bằng văn bản và ngược lại.
* ALIGN là mô hình học sâu đa phương thức được phát triển bởi Google Research, tương tự như CLIP nhưng được huấn luyện trên một tập dữ liệu khổng lồ bao gồm hình ảnh và văn bản không gắn nhãn (noisy text). Mô hình này học cách kết hợp hình ảnh và văn bản trong một không gian nhúng chung để cải thiện khả năng hiểu và tìm kiếm hình ảnh.

Cách thức hoạt dộng:

CLIP sử dụng hai mô hình riêng biệt: một mô hình xử lý hình ảnh (thường là CNN hoặc Vision Transformer - ViT) và một mô hình xử lý văn bản (thường là Transformer). Các mô hình này được huấn luyện bằng cách sử dụng loss function contrastive, giúp tối ưu hóa mối quan hệ giữa hình ảnh và văn bản liên quan.

ALIGN sử dụng mô hình Transformer cho cả hình ảnh và văn bản để học biểu diễn nhúng, và cũng sử dụng một chiến lược huấn luyện contrastive. Dữ liệu văn bản không gắn nhãn giúp mô hình học được các đặc trưng tổng quát hơn về mối quan hệ giữa hình ảnh và văn bản, không cần phải có dữ liệu gắn nhãn chính xác.

Các mô hình hybrid như CLIP và ALIGN thực hiện vector hóa hình ảnh thông qua một quá trình kết hợp giữa hình ảnh và văn bản, nhằm tạo ra một không gian nhúng chung mà trong đó các hình ảnh và văn bản liên quan có thể được biểu diễn dưới dạng các vector.

* Tiền xử lý hình ảnh: hình ảnh sẽ được thay đổi kích thước để phù hợp với yêu cầu của mô hình (ví dụ như 224x224 pixel). Các giá trị pixel của hình ảnh sẽ được chuẩn hóa, giúp các giá trị này trở nên đồng nhất và dễ dàng xử lý bởi các mạng neuron. Sau đó, hình ảnh sẽ được chuyển đổi thành tensor, dạng dữ liệu mà mô hình có thể hiểu và xử lý.
* Biểu diễn hình ảnh (Feature Extraction): Sau khi tiền xử lý, hình ảnh sẽ được đưa qua một mô hình học sâu để trích xuất các đặc trưng. Các mô hình như Vision Transformer (ViT) hoặc ResNet sẽ nhận hình ảnh và xử lý chúng để chia thành các phần nhỏ (patches) hoặc thông qua các lớp convolution để trích xuất các đặc trưng cục bộ. Những đặc trưng này sau đó được chuyển thành các vector nhúng, đại diện cho nội dung hình ảnh, giúp mô hình hiểu rõ các đối tượng và cấu trúc trong hình ảnh.
* Biểu diễn văn bản: Song song với việc xử lý hình ảnh, văn bản mô tả về hình ảnh cũng được đưa vào mô hình, xử lý thông qua các mô hình ngôn ngữ, chủ yếu là các biến thể của Transformer chuyển thành một vector nhúng riêng biệt. Mô hình này sẽ học được các mối quan hệ ngữ nghĩa trong văn bản, giúp hiểu rõ nội dung mà văn bản mô tả. Các vector nhúng này không chỉ mô tả hình ảnh mà còn gắn kết chặt chẽ với văn bản mô tả, tạo ra một hệ thống có thể liên kết thông tin từ cả hai nguồn.
* Không gian nhúng chung: nơi các vector của hình ảnh và văn bản sẽ được ánh xạ vào không gian đồng nhất. Sử dụng kỹ thuật contrastive learning để tối ưu hóa việc đưa các vector nhúng của các cặp hình ảnh và văn bản liên quan lại gần nhau trong không gian nhúng, trong khi đó tăng khoảng cách giữa các vector của các cặp không liên quan. Quá trình này giúp mô hình học được mối quan hệ chặt chẽ giữa hình ảnh và văn bản, tối ưu hóa khả năng hiểu và tìm kiếm.

Đặc Điểm Của Vector Nhúng Hình Ảnh trong CLIP và ALIGN

Mã hóa đặc trưng hình ảnh: Các vector này chứa thông tin quan trọng về các đặc trưng của hình ảnh, như các đối tượng, cấu trúc, cảnh quan, v.v., nhưng không phụ thuộc vào cấu trúc không gian như trong các mô hình CNN truyền thống. Việc chia nhỏ hình ảnh thành các patch (trong ViT) hay trích xuất đặc trưng từ các lớp CNN cho phép mô hình học được những thông tin tổng quát hơn về hình ảnh.

Không gian nhúng chung: Điều này là một điểm mạnh của CLIP và ALIGN, khi mà hình ảnh và văn bản không chỉ được biểu diễn riêng biệt mà còn chia sẻ cùng một không gian vector. Điều này giúp mô hình học được mối quan hệ giữa hình ảnh và văn bản mô tả mà không cần phải tái huấn luyện mô hình cho từng tác vụ cụ thể.

## Phương pháp đo lường sự tương đồng giữa các vector trong không gian embedding

### Cosine similarity

Độ tương tự cosine là một phương pháp phổ biến để đo lường mức độ tương đồng giữa hai vector trong không gian vô hướng. Nó dựa trên giá trị cosine của góc giữa hai vector, cho phép xác định mối quan hệ giữa chúng mà không bị ảnh hưởng bởi kích thước và độ dài của vector.

Công thức:

Cosine Similarity (A, B) =

Trong đó:

* AB là tích vô hướng (dot product) của hai vector
* ||A|| và ||B|| là độ dài (norm) của vector A và B, được tính bằng công thức :

A square root of a function

Description automatically generated

Độ tương tự cosine nằm trong khoảng từ -1 đến 1; giá trị gần 1 cho thấy hai vector rất giống nhau, trong khi giá trị gần -1 cho thấy chúng trái ngược nhau.

Ứng dụng:

* Tìm kiếm thông tin: Trong lĩnh vực tìm kiếm thông tin, Cosine Similarity được sử dụng để xác định mức độ liên quan giữa một truy vấn tìm kiếm và các tài liệu trong cơ sở dữ liệu. Khi các tài liệu được biểu diễn dưới dạng vector (thường là vector tf-idf), Cosine Similarity giúp đánh giá mức độ tương đồng giữa truy vấn và tài liệu, từ đó xác định tài liệu nào là phù hợp nhất để trả về cho người dùng.
* Phân loại văn bản: Cosine Similarity có thể được áp dụng trong các hệ thống phân loại văn bản để xác định xem một tài liệu mới có thuộc về một nhóm nào đó hay không. Bằng cách so sánh vector của tài liệu mới với các vector của các nhóm đã biết, hệ thống có thể phân loại tài liệu vào nhóm có độ tương đồng cao nhất.
* Hệ thống gợi ý: Trong các hệ thống gợi ý (recommendation systems), Cosine Similarity giúp tìm ra những sản phẩm hoặc nội dung tương tự mà người dùng có thể thích dựa trên sở thích và hành vi trước đó. Ví dụ, trong một nền tảng thương mại điện tử, nếu hai sản phẩm có vector tương tự, chúng có thể được gợi ý cho cùng một người dùng.

### Khoảng cách Euclidean

Khoảng cách Euclidean là một khái niệm cơ bản trong toán học, được sử dụng để đo lường độ dài giữa hai điểm trong không gian Euclid. Khoảng cách này có thể được tính toán bằng định lý Pythagoras và thường được áp dụng trong nhiều lĩnh vực như thống kê, học máy, và phân tích dữ liệu.

Khoảng cách Euclidean giữa hai vector AA và BB trong không gian n*n*-chiều được định nghĩa bằng công thức:

A black and white symbol

Description automatically generated

Trong đó:

và là các thành phần của vector A và B.

d(A, B) là khoảng cách Euclidean giữa hai vector.

Để giảm bớt độ phức tạp khi tính toán, khoảng cách bình phương (squared Euclidean distance) thường được sử dụng, được tính như sau:

A black and white math equation

Description automatically generated

Ứng dụng:

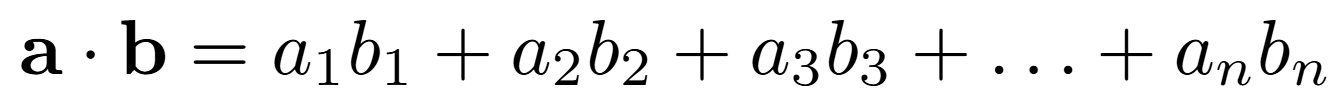
* Tìm kiếm hàng xóm gần nhất: Trong bài toán tìm kiếm, khoảng cách Euclidean được sử dụng để xác định các embedding gần nhất với một embedding truy vấn. Ví dụ, nếu có một ma trận embedding E chứa các vector nhúng của các đối tượng và một vector truy vấn q, khoảng cách giữa q và mỗi vector nhúng ​ trong E được tính bằng khoảng cách Euclidean.
* Phân loại và phân nhóm: Khoảng cách Euclidean giúp phân loại hoặc phân nhóm dữ liệu bằng cách xác định các đối tượng gần nhau trong không gian nhúng, từ đó tạo ra các nhóm có đặc điểm tương tự

### Tích vô hướng (Dot Product)

Tích vô hướng là một phép toán cơ bản trong đại số tuyến tính, được sử dụng để xác định mối quan hệ giữa hai vector. Kết quả của phép toán này là một giá trị vô hướng (scalar), không phải là một vector mới.

Công thức:

* Công thức tổng quát để tính tích vô hướng giữa hai vector a và b trong không gian *n*-chiều được định nghĩa như sau:



Trong đó:

a = () và b = () là các vector trong không gian n-chiều.

* Ngoài cách tính dựa trên thành phần của vector, tích vô hướng cũng có thể được định nghĩa thông qua độ lớn của các vector và góc giữa chúng:

A black letter on a white background

Description automatically generated

Trong đó:

* |a| và |b| là độ dài của hai vector
* là góc giữa hai vector

Ứng dụng:

## Đánh giá và tổng kết

Dlib, FaceAPI, MTCNN và RetinaFace đều là các công cụ nhận dạng khuôn mặt được phát triển để giúp xác định và nhận dạng các thuộc tính khuôn mặt. Tuy nhiên, từng công cụ có những ưu và nhược điểm của riêng nó:

* Dlib là một thư viện xử lý hình ảnh C++ hoạt động tốt trên cả các ảnh độ phân giải cao và thấp. Nó cho phép phát hiện khuôn mặt và đưa ra các điểm mốc khuôn mặt bao gồm cả các điểm mốc mắt, mũi, miệng.
* FaceAPI là một API dựa trên điện toán đám mây và được đào tạo với hàng triệu ảnh khuôn mặt. Nó có khả năng phát hiện khuôn mặt chính xác và nhận dạng thuộc tính như tuổi, giới tính, biểu cảm,…
* MTCNN là mô hình phức tạp hơn, có thể phát hiện và xác định các trọng tâm, kích thước và hướng của khuôn mặt. Tuy nhiên khả năng nhận diện các khuôn mặt khác nhau của MTCNN có thể không đạt hiệu suất cao trên dữ liệu nhiều đầu vào khác nhau.
* RetinaFace là một mô hình nhận dạng khuôn mặt mới được phát triển và bao gồm cả phát hiện, phân đoạn landmark và phân tích thuộc tính khuôn mặt. Nó có khả năng xử lý các khuôn mặt với nhiều biến tắc và màn nhỏ, độ chính xác trên tổng thể.

Tóm lại, MTCNN hoạt động tốt khi phải xử lý hàng nghìn ảnh khuôn mặt trong thời gian ngắn. Dlib có khả năng hoạt động trên các ảnh với điểm ảnh thấp, nhiễu. FaceAPI là API tiện ích cho việc phân tích chi tiết khuôn mặt như tuổi, giới tính, biểu cảm,… RetinaFace đưa ra độ chính xác cao và xử lý rất nhanh.

Đề tài xây dựng hệ thống nhận diện khuôn mặt để trả về thông tin cần độ chính xác cao nên em sẽ sử dụng thư viện dlib.

## Tổng kết chương 3

Trong chương này, em đã nghiên cứu và tìm hiểu các công cụ phần mềm được sử dụng trong nhận dạng khuôn mặt: Thư viện Dlib, Mô hình RetinaFace, Mô hình MTCNN, Công cụ FaceAPI. Cài đặt thực nghiệm, đánh giá các mô hình và đưa ra phương án phù hợp để xây dựng hệ thống nhận diện khuôn mặt.

# THỰC NGHIỆM

Giới thiệu chương: Trình bày các thực nghiệm nhằm kiểm tra và đánh giá hiệu quả của hệ thống tìm kiếm ảnh dựa trên mô tả. Tóm tắt các nội dung chính trong chương.

## Dữ liệu thực nghiệm

Flickr30k là một tập dữ liệu nổi tiếng trong lĩnh vực chú thích ảnh, được sử dụng rộng rãi để phát triển và đánh giá các mô hình học máy, đặc biệt là trong bài toán sinh mô tả cho ảnh.

Thông tin chung về Flickr30k:

* Số lượng hình ảnh: Tập dữ liệu chứa 31,000 hình ảnh được thu thập từ Flickr.
* Chú thích: Mỗi hình ảnh đi kèm với 5 câu mô tả (captions) do con người ghi lại, giúp cung cấp nhiều góc nhìn khác nhau về nội dung hình ảnh.
* Mục đích sử dụng: Tập dữ liệu này thường được sử dụng để huấn luyện và kiểm tra các mô hình học sâu trong việc tạo ra chú thích cho hình ảnh, cho phép nghiên cứu các phương pháp như CNN (Convolutional Neural Networks) kết hợp với RNN (Recurrent Neural Networks) hoặc LSTM (Long Short-Term Memory) để sinh ra văn bản mô tả.

Cấu trúc và Phân chia Dữ liệu:

* Phân chia dữ liệu: Thông thường, tập dữ liệu được chia thành ba phần:
* Tập huấn luyện: 6000 hình ảnh
* Tập phát triển (validation): 1000 hình ảnh
* Tập kiểm tra (test): 1000 hình ảnh
* Định dạng lưu trữ: Hình ảnh thường được lưu dưới định dạng JPEG, và các chú thích được lưu trong các file văn bản riêng biệt, với mỗi dòng tương ứng với một câu mô tả cho một hình ảnh cụ thể.

Ứng dụng

Flickr30k không chỉ hữu ích cho việc phát triển các hệ thống chú thích tự động mà còn đóng vai trò quan trọng trong việc nghiên cứu các phương pháp học máy khác nhau. Nó đã được sử dụng để thử nghiệm nhiều mô hình tiên tiến, như Transformer và các kiến trúc mạng sâu khác nhằm cải thiện độ chính xác và khả năng sinh mô tả14.

Tóm lại, Flickr30k là một nguồn tài nguyên quý giá cho các nhà nghiên cứu và phát triển trong lĩnh vực thị giác máy tính và xử lý ngôn ngữ tự nhiên, cung cấp nền tảng cho nhiều nghiên cứu và ứng dụng thực tế.

## Phương pháp thực nghiệm

Hệ thống được xây dựng

Mô tả kiến trúc và cách hoạt động của hệ thống (CLIP):

Hệ thống tìm kiếm ảnh dựa trên mô tả sử dụng mô hình CLIP (Contrastive Language–Image Pretraining). CLIP là mô hình học sâu do OpenAI phát triển, cho phép ánh xạ đồng thời dữ liệu văn bản và hình ảnh vào cùng một không gian nhúng.

* Kiến trúc: CLIP bao gồm hai thành phần chính: một encoder cho văn bản (Text Encoder) và một encoder cho hình ảnh (Image Encoder). Các encoder này sử dụng Transformer để mã hóa đầu vào thành các vector đặc trưng. Vector đặc trưng của văn bản và hình ảnh được so sánh thông qua hàm cosine similarity.
* Cách hoạt động: Hệ thống nhận mô tả bằng văn bản từ người dùng, mã hóa mô tả này thành vector. Tương tự, tất cả hình ảnh trong tập dữ liệu cũng được mã hóa trước thành vector. Sau đó, hệ thống tìm kiếm các hình ảnh có vector gần nhất với vector mô tả của người dùng.

Các công cụ, framework được sử dụng:

* FAISS (Facebook AI Similarity Search): là một thư viện mã nguồn mở được phát triển bởi Facebook AI, thiết kế để tìm kiếm gần đúng các vector trong không gian nhiều chiều. FAISS đặc biệt phù hợp với các bài toán yêu cầu xử lý và tìm kiếm dữ liệu lớn trong các ứng dụng liên quan đến Machine Learning và AI.

Tính năng nổi bật:

* FAISS hỗ trợ tìm kiếm các vectơ tương tự bằng cách sử dụng nhiều thuật toán khác nhau, giúp xác định các điểm gần nhất trong không gian vectơ một cách nhanh chóng và hiệu quả.
* Thư viện cung cấp nhiều loại chỉ mục (index) để tối ưu hóa tốc độ và độ chính xác của việc tìm kiếm, như IndexFlatL2 cho khoảng cách Euclid.
* FAISS có thể xử lý các tập dữ liệu lớn mà không cần phải tải toàn bộ vào RAM, nhờ vào các phương pháp như lượng tử hóa (quantization) và KNN xấp xỉ.
* Mặc dù chủ yếu được mã hóa bằng C++ nhưng nó hỗ trợ đầy đủ tích hợp Python/NumPy. Một số thuật toán chính của nó cũng có sẵn để thực thi GPU.

Cách hoạt động:

FAISS hoạt động bằng cách xây dựng một cấu trúc dữ liệu từ tập hợp các vectơ ban đầu. Khi có một vectơ mới cần tìm kiếm, FAISS sẽ sử dụng cấu trúc chỉ mục đã tạo để nhanh chóng xác định các vectơ tương tự. Quá trình này bao gồm hai bước chính:

* Tạo chỉ mục: Chỉ mục được xây dựng từ tập hợp vectơ ban đầu, cho phép truy vấn nhanh chóng.
* Tìm kiếm: Khi nhận được một vectơ truy vấn, FAISS sẽ thực hiện tìm kiếm trong chỉ mục để trả về danh sách các vectơ tương tự nhất.

A diagram of a diagram of a bus

Description automatically generated with medium confidence

* NumPy (Numerical Python): là một thư viện mã nguồn mở mạnh mẽ dành cho ngôn ngữ lập trình Python, chủ yếu được sử dụng trong lĩnh vực khoa học máy tính và phân tích dữ liệu. Thư viện này cung cấp khả năng xử lý các mảng đa chiều (multidimensional arrays) và thực hiện các phép toán số học một cách nhanh chóng và hiệu quả.

A diagram of a computer code

Description automatically generated with medium confidence

Tính năng chính:

* Mảng đa chiều (ndarray): NumPy cung cấp một đối tượng mảng N-chiều, cho phép lưu trữ và thao tác với dữ liệu theo nhiều chiều khác nhau. Mỗi phần tử trong mảng phải có cùng kiểu dữ liệu, giúp tối ưu hóa bộ nhớ và tăng tốc độ xử lý.
* Broadcasting: Tính năng này cho phép thực hiện các phép toán giữa các mảng có kích thước khác nhau mà không cần phải sao chép dữ liệu, giúp tiết kiệm bộ nhớ và tăng hiệu suất.
* Hàm toán học cao cấp: NumPy cung cấp một loạt các hàm toán học như đại số tuyến tính, biến đổi Fourier, và các phép toán thống kê, giúp người dùng thực hiện các phép tính phức tạp một cách dễ dàng.
* Tích hợp với ngôn ngữ khác: NumPy cho phép tích hợp mã C/C++ và Fortran, giúp tối ưu hóa hiệu suất cho các ứng dụng yêu cầu tính toán nặng.

Lợi ích:

* Hiệu suất cao: Các phép toán trên mảng NumPy nhanh hơn nhiều so với việc sử dụng danh sách Python thông thường nhờ vào cách tổ chức dữ liệu trong bộ nhớ.
* Tiết kiệm bộ nhớ: Mỗi mảng NumPy sử dụng ít bộ nhớ hơn so với danh sách Python vì tất cả các phần tử đều có cùng kiểu dữ liệu.
* Cơ sở cho các thư viện khác: NumPy là nền tảng cho nhiều thư viện phổ biến trong hệ sinh thái Python như Pandas, SciPy, Matplotlib, và scikit-learn, làm cho nó trở thành một công cụ thiết yếu cho bất kỳ ai làm việc trong lĩnh vực khoa học dữ liệu hoặc máy học.
* Streamlit: là một framework mã nguồn mở được phát triển bằng Python, cho phép người dùng dễ dàng tạo ra các ứng dụng web tương tác phục vụ cho việc phân tích dữ liệu và triển khai mô hình machine learning. Ra mắt vào năm 2019, Streamlit nhanh chóng trở thành công cụ phổ biến trong cộng đồng khoa học dữ liệu nhờ vào tính đơn giản và hiệu quả, không cần kiến thức sâu về phát triển web mà vẫn tạo được giao diện trực quan và tương tác cao.

Hoạt động theo nguyên tắc chạy từng dòng mã Python từ trên xuống dưới. Mỗi khi người dùng tương tác với ứng dụng, toàn bộ mã sẽ được thực thi lại để cập nhật dữ liệu và giao diện. Điều này giúp đảm bảo rằng người dùng luôn thấy dữ liệu mới nhất mà không cần phải viết thêm mã callback phức tạp.

Đặc điểm nổi bật:

* Người dùng chỉ cần biết Python để xây dựng ứng dụng mà không cần có kiến thức về HTML, CSS hay JavaScript. Điều này giúp giảm bớt rào cản cho những người mới bắt đầu trong lĩnh vực phát triển ứng dụng web.
* Streamlit tự động cập nhật giao diện người dùng mỗi khi mã nguồn thay đổi, giúp tiết kiệm thời gian phát triển.
* Hỗ trợ trực tiếp nhiều thư viện phân tích dữ liệu như Pandas, NumPy và Matplotlib, cho phép hiển thị dữ liệu một cách trực quan.
* Thư viện cung cấp nhiều hàm để tạo ra các thành phần giao diện như văn bản, biểu đồ, bảng và các điều khiển tương tác như selectbox và slider.

Streamlit với các công cụ tương tự

|  |  |  |  |
| --- | --- | --- | --- |
| Tiêu chí | Streamlit | Dash | Flask + HTML |
| Độ dễ sử dụng | Rất dễ, không cần HTML/CSS | Cần biết thêm về Plotly | Cần biết về HTML/CSS |
| Tốc độ phát triển | Nhanh chóng | Trung bình | Chậm hơn |
| Đối tượng sử dụng | Nhà khoa học dữ liệu, ML | Nhà phát triển web và ML | Nhà phát triển web |
| Mức độ tuỳ biến | Trung bình | Cao | Rất cao |

* Pillow (Python Imaging Library): là một thư viện xử lý hình ảnh trong Python, được phát triển như một phiên bản nâng cấp của thư viện Python Imaging Library (PIL). Pillow cung cấp các công cụ mạnh mẽ để mở, thao tác và lưu trữ hình ảnh, đồng thời hỗ trợ nhiều định dạng tệp khác nhau. Thư viện này được cập nhật thường xuyên và được đánh giá cao trong cộng đồng lập trình viên.

Tính năng chính

* Mở và lưu hình ảnh: Pillow cho phép người dùng mở nhiều định dạng hình ảnh như JPEG, PNG, GIF, BMP, và nhiều định dạng khác. Bạn có thể dễ dàng lưu hình ảnh dưới định dạng mong muốn mà không cần chỉ định đuôi tệp nếu tệp đã có đuôi phổ biến.
* Chỉnh sửa hình ảnh: Thư viện hỗ trợ nhiều thao tác chỉnh sửa như cắt (crop), thay đổi kích thước (resize), xoay (rotate), và biến đổi màu sắc (convert). Người dùng có thể cắt một phần của hình ảnh bằng cách chỉ định tọa độ của vùng cần cắt.
* Xử lý màu sắc: Pillow cho phép chuyển đổi giữa các chế độ màu khác nhau, chẳng hạn như từ RGB sang đen trắng (L). Bạn cũng có thể tách các kênh màu riêng biệt để chỉnh sửa và sau đó gộp lại thành một hình ảnh mới.
* Thêm bộ lọc: Thư viện cung cấp một bộ sưu tập các bộ lọc để xử lý hình ảnh, như làm mờ (blur), làm sắc nét (sharpen), và tìm cạnh (find edges).
* Tạo thumbnail: Bạn có thể tạo ra các phiên bản thu nhỏ của hình ảnh mà vẫn giữ nguyên các đặc điểm chính của chúng.
* Sentence\_transformers: là một thư viện Python mạnh mẽ được phát triển để tạo ra các vector nhúng (embeddings) cho câu, đoạn văn và hình ảnh. Thư viện này được xây dựng dựa trên mô hình BERT và các biến thể của nó, giúp cải thiện khả năng hiểu ngữ nghĩa của các câu so với các phương pháp nhúng truyền thống như Word2Vec hay GloVe. Sentence Transformers rất hữu ích trong các ứng dụng như tìm kiếm ngữ nghĩa, phân cụm và xác định độ tương đồng giữa các câu.

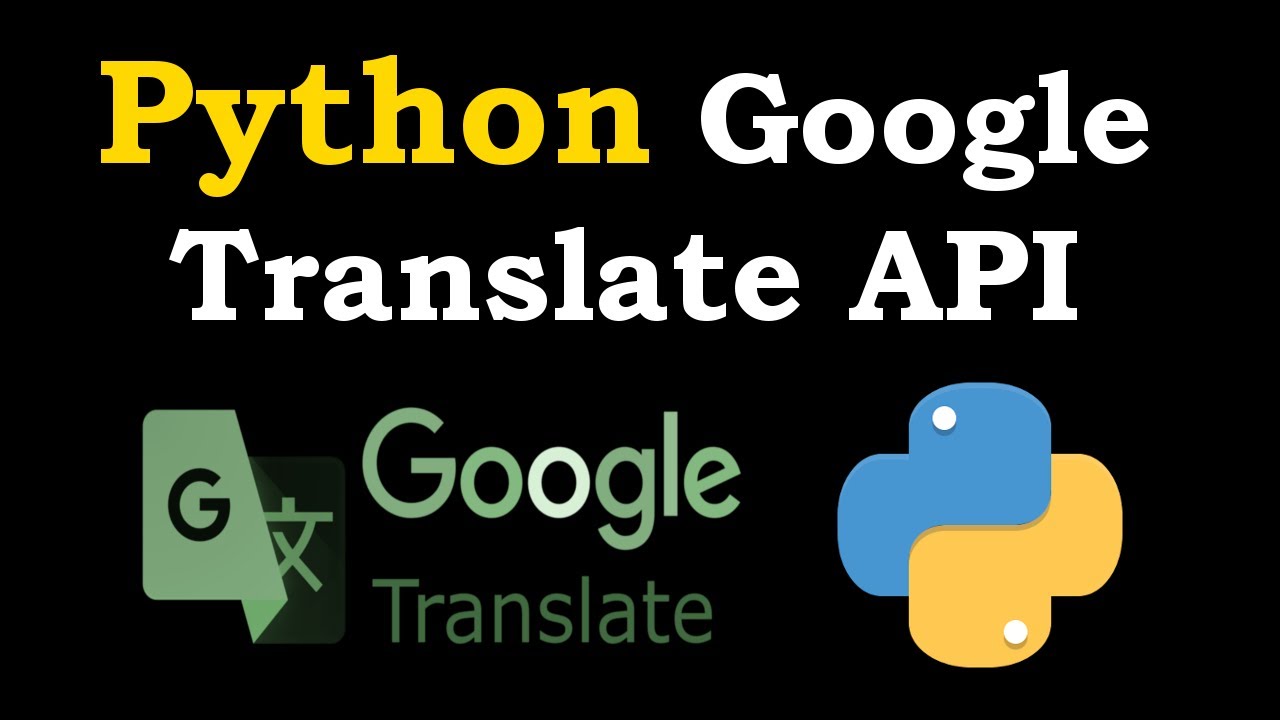
Sentence Transformers sử dụng kiến trúc siamese để huấn luyện trên các cặp câu. Mỗi câu được đưa qua cùng một mô hình để tạo ra hai vector nhúng, sau đó kết hợp chúng để tạo ra một vector đại diện cho mối quan hệ giữa hai câu. Quá trình này giúp mô hình hiểu rõ hơn về ngữ nghĩa của các câu trong bối cảnh.

Tính năng nổi bật

* Mô hình nhúng câu: Sentence Transformers cho phép mã hóa toàn bộ câu hoặc đoạn văn thành các vector có kích thước cố định, giữ lại ngữ cảnh và ý nghĩa của câu. Điều này giúp cải thiện hiệu suất trong các tác vụ xử lý ngôn ngữ tự nhiên (NLP).
* Khả năng tương tác: Thư viện cung cấp các phương pháp dễ dàng để tính toán độ tương đồng giữa các vector, hỗ trợ cho việc tìm kiếm và phân tích dữ liệu.
* Hỗ trợ nhiều mô hình: Có hàng trăm mô hình đã được huấn luyện sẵn có sẵn trên nền tảng Hugging Face, cho phép người dùng lựa chọn mô hình phù hợp với nhu cầu của mình.
* Tích hợp dễ dàng: Việc sử dụng Sentence Transformers rất đơn giản, chỉ cần vài dòng mã để tải mô hình và mã hóa câu.
* Googletrans: là một thư viện Python cho phép người dùng dễ dàng truy cập và sử dụng dịch vụ Google Dịch (Google Translate) thông qua API không chính thức. Thư viện này giúp thực hiện các tác vụ dịch thuật tự động từ một ngôn ngữ này sang ngôn ngữ khác mà không cần phải tương tác trực tiếp với giao diện web của Google Dịch.

Tính năng chính

* Hỗ trợ dịch qua hơn 100 ngôn ngữ: Danh sách các ngôn ngữ hỗ trợ rất phong phú, bao gồm cả các ngôn ngữ phổ biến như tiếng Anh, tiếng Trung, tiếng Việt và các ngôn ngữ hiếm hơn như tiếng Hawaii hoặc tiếng Swahili.
* Googletrans tự động xác định ngôn ngữ của văn bản đầu vào nếu không chỉ định ngôn ngữ nguồn (src). Điều này rất hữu ích khi xử lý các văn bản đa ngôn ngữ mà không cần biết trước ngôn ngữ gốc.
* Hỗ trợ dịch cả đoạn văn dài hoặc danh sách nhiều dòng văn bản trong một lệnh.
* Googletrans có API trực quan, dễ sử dụng ngay cả với người mới. Cách sử dụng các phương thức của thư viện rất giống với các hàm Python thông thường.



Thiết lập thực nghiệm

Thông số mô hình:

* Sử dụng mô hình CLIP phiên bản ViT-B/32 (Vision Transformer) để đạt hiệu suất tốt.
* Batch size: 256 ảnh /mỗi lần.

Dữ liệu ảnh có thể chiếm một lượng bộ nhớ đáng kể, đặc biệt khi có hàng nghìn hoặc hàng triệu ảnh. Việc xử lý tất cả ảnh một lần trong một batch lớn sẽ làm quá tải bộ nhớ và khiến quá trình xử lý không hiệu quả. Vì vậy cần chia nhỏ các ảnh thành các nhóm với kích thước 256 ảnh mỗi lần. Điều này giúp giảm bớt chi phí bộ nhớ và cải thiện hiệu suất khi tính toán các embeddings cho nhiều ảnh, giúp tăng tốc quá trình mã hóa bằng cách tận dụng tối đa khả năng xử lý song song của phần cứng.

* Image size: 224x224 pixel (mặc định sau khi resize ảnh). Mô hình CLIP tự động điều chỉnh kích thước ảnh (resize) trước khi mã hóa ảnh.

Kích thước ảnh 224x224 là một chuẩn phổ biến trong các mô hình học sâu, giúp tối ưu hóa hiệu suất và độ chính xác mà không làm giảm quá nhiều chi tiết quan trọng trong ảnh. Kích thước này đủ nhỏ để giảm chi phí tính toán, nhưng đủ lớn để giữ lại thông tin quan trọng trong ảnh.

* Embedding size: 512 (mỗi ảnh hoặc văn bản sẽ được mã hóa thành một vector 512 chiều).

Khi sử dụng mô hình CLIP, mỗi ảnh và văn bản đầu vào được mã hóa thành một vector có kích thước 512 chiều. Đây là chiều của không gian nhúng mà CLIP sử dụng để kết nối hình ảnh và văn bản trong một không gian chung.

* FAISS Index: IndexFlatIP, sử dụng tích vô hướng để tính toán độ tương đồng. Là phương pháp hiệu quả khi làm việc với không gian nhúng mà không cần sử dụng các tối ưu hóa phức tạp như nén hoặc phân mảnh. Chỉ mục này không nén các vector mà lưu trữ chúng dưới dạng tệp đơn giản, giúp duy trì độ chính xác cao trong việc tính toán độ tương đồng giữa các embeddings.

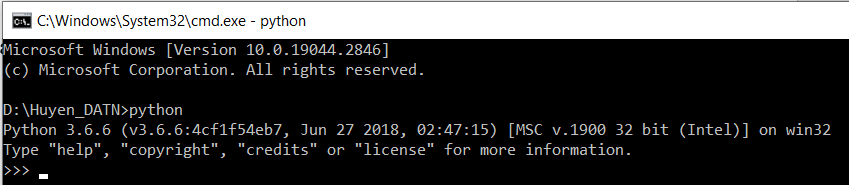
Môi trường thực nghiệm:

* Phần cứng:
* GPU: NVIDIA A100
* CPU:
* RAM: Tối thiểu 8GB
* Bộ nhớ lưu trữ:
* Phần mềm:
* Hệ điều hành: Ubuntu 20.04 hoặc Windows 11.
* Thư viện: FAISS, NumPy, Streamlit, Pillow, Googletrans, Sentence-transformers.
* Python: Phiên bản 3.12.0

## Cài đặt thực nghiệm cho bài toán

### Cài đặt Python

Để cài đặt python truy cập địa chỉ https://[www.python.org/downloads/](http://www.python.org/downloads/) để download và cài đặt phiên bản thích hợp. Sau đó vào cmd để kiểm tra xem cài đặt đã thành công hay chưa ở đây chúng ta sử dụng phiên bản 3.12.0



### Cài đặt các thư viện cần thiết

* Liệt kê các thư viện cần cài đặt để sử dụng trong chương trình vào một file định dạng txt, tên file có thể đặt là *requirements.txt*

A screenshot of a computer

Description automatically generated

* Sử dụng lệnh cd để điều hướng đến thư mục chứa đoạn code của chương trình

Ví dụ: cd T:\DATN\_Search\_Image\SearchImage\_App

* Đảm bảo file requirements.txt đã liệt kê đầy đủ các thư viện cần thiết. Chạy lệnh sau để tự động tải và cài đặt: pip install -r requirements.txt

### Cài đặt chương trình

A screenshot of a computer

Description automatically generated

Bước 1: Nhập các thư viện cần thiết

A screen shot of a computer code

Description automatically generated

Bước 2: Khai báo 2 hàm kiểm tra câu mô tả nhập vào có phải là tiếng Việt không và hàm dịch tiếng Việt sang tiếng anh sử dụng thư viện Translate của python

A computer screen shot of a code

Description automatically generated

Hàm is\_vietnamese(text): Hàm này dùng để kiểm tra xem một đoạn văn bản có chứa ký tự tiếng Việt hay không. Nó sử dụng một biểu thức chính quy (regex) để so khớp các ký tự tiếng Việt có dấu (cả chữ hoa và chữ thường). Nếu tìm thấy ký tự tiếng Việt, hàm trả về True; ngược lại, trả về False. Điều này hữu ích khi cần phân biệt văn bản tiếng Việt với các ngôn ngữ khác.

Hàm translate\_to\_english(text): Hàm này dịch văn bản từ tiếng Việt sang tiếng Anh bằng thư viện googletrans. Đầu tiên, nó tạo một đối tượng Translator, sau đó gọi phương thức translate() để dịch văn bản từ nguồn là tiếng Việt (src='vi') sang đích là tiếng Anh (dest='en'). Nếu dịch thành công, hàm trả về văn bản đã dịch. Trong trường hợp xảy ra lỗi (ví dụ không thể kết nối dịch vụ), một thông báo lỗi được hiển thị trên giao diện (sử dụng st.error()), và hàm trả về đoạn văn bản gốc.

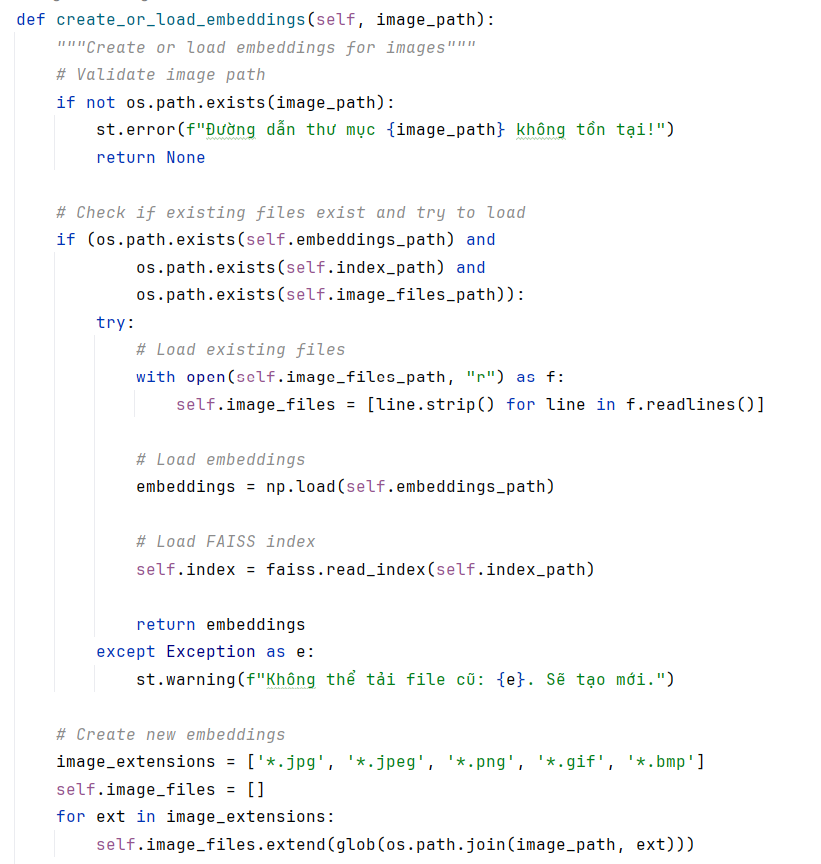
Bước 3: Xây dựng lớp ImageSearchApp với hàm khởi tạo khai báo sử dụng model 'clip-ViT-B-32'

A screen shot of a computer program

Description automatically generated

Đoạn mã khởi tạo các thuộc tính cần thiết trong lớp ImageSearchApp, nhằm quản lý dữ liệu và tối ưu hóa quá trình tìm kiếm ảnh. Đầu tiên, mô hình SentenceTransformer được khởi tạo với tên mô hình (model\_name), dùng để chuyển đổi văn bản hoặc hình ảnh thành các vector biểu diễn (embedding). Thuộc tính index được khởi tạo với giá trị None và sẽ lưu chỉ mục FAISS để tìm kiếm các vector tương đồng. Danh sách image\_files được khởi tạo rỗng để lưu các đường dẫn hình ảnh. Các thuộc tính embeddings\_path, index\_path, và image\_files\_path lần lượt là các đường dẫn để lưu trữ embedding, chỉ mục FAISS, và danh sách tệp hình ảnh đã xử lý. Việc lưu trữ này giúp tái sử dụng dữ liệu đã xử lý, giảm thiểu thời gian tính toán lại mỗi khi chương trình khởi động.

Bước 4: Xây dựng hàm tạo file embedding hình ảnh và tạo chỉ số index cho từng vector nhúng



A screenshot of a computer program

Description automatically generated

Phương thức create\_or\_load\_embeddings có nhiệm vụ tạo mới hoặc tải lại các vector biểu diễn (embeddings) cho các hình ảnh, giúp tối ưu hóa quá trình tìm kiếm. Đầu tiên, phương thức kiểm tra xem đường dẫn đến thư mục hình ảnh (image\_path) có tồn tại không; nếu không, nó sẽ báo lỗi và kết thúc. Sau đó, chương trình kiểm tra xem các tệp đã xử lý trước đó (embedding, index, và danh sách tệp hình ảnh) có tồn tại không. Nếu có, nó cố gắng tải dữ liệu từ các tệp này: danh sách tệp hình ảnh từ tệp văn bản, embeddings từ tệp .npy, và chỉ mục FAISS từ tệp chỉ mục. Nếu tải thành công, các dữ liệu này được sử dụng lại.

Nếu không tồn tại hoặc không tải được dữ liệu cũ, phương thức sẽ tạo embeddings mới. Đầu tiên, nó quét thư mục hình ảnh, tìm các tệp có phần mở rộng phù hợp (như .jpg, .png, v.v.). Nếu không tìm thấy ảnh nào, nó báo lỗi và kết thúc. Tiếp theo, phương thức xử lý từng nhóm hình ảnh (được chia thành các lô nhỏ để tránh quá tải bộ nhớ), chuyển đổi mỗi lô thành embeddings thông qua mô hình SentenceTransformer.

Sau khi xử lý xong, các embeddings được lưu vào tệp .npy. Phương thức cũng xây dựng một chỉ mục FAISS từ các embeddings này và lưu chỉ mục ra tệp. Cuối cùng, danh sách các đường dẫn hình ảnh được ghi vào tệp văn bản để sử dụng lại trong tương lai. Toàn bộ dữ liệu này được trả về để sử dụng trong quá trình tìm kiếm.

A computer screen shot of text

Description automatically generated

Bước 1: Tài bộ dữ liệu flickr30k\_images từ Kaggle theo đường link sau: https://www.kaggle.com/datasets/hsankesara/flickr-image-dataset.

* Cập nhật đường dẫn thư mục trong mã nguồn (biến IMAGE\_FOLDER).

## So sánh và đánh giá các mô hình text embedding trong thực nghiệm

## Xây dựng ứng dụng tìm kiếm ảnh dựa trên mô tả

Viết các module trong hệ thống (Các hàm kèm giải thích các hàm đó có tác dụng gì, dùng cái gì, thực hiện ra sao, tại sao lại dùng cái đó, … )

## Tổng kết chương

Trong chương này, em thực hiện xây dựng hệ thống nhận diện khuôn mặt để trả về thông tin sử dụng Thư viện Dlib để nhận diện khuôn mặt và thông qua Elasticsearch để truy vấn dữ liệu. Em đã đẩy lên thành API và demo bằng app android đơn giản. App cho phép chụp ảnh trực tiếp bằng camera hoặc chọn một ảnh có sẵn trong thư viện để nhận diện khuôn mặt và trả về thông tin mã sinh viên tương ứng.

# KẾT LUẬN



* Kết quả đạt được

Báo cáo đã trình bày được cơ bản nghiên cứu về mô hình học sâu; tìm hiểu, cài đặt thực nghiệm và đánh giá các công cụ phần mềm được sử dụng trong nhận diện khuôn mặt (Dlib, RetinaFace, MTCNN, FaceAPI). Từ đó xây dựng được hệ thống nhận diện khuôn mặt trả về thông tin sử dụng thư viện Dlib và Elasticsearch. Nội dung chi tiết các phần mà báo cáo đã thực hiện được:

* Nghiên cứu tổng quan về mô hình học sâu, mạng nơ ron nghiên cứu chi tiết về mạng nơ-ron cổ điển, mạng nơ-ron tích chập (CNN), mạng nơ-ron hồi quy (RNN), mạng sinh đối nghịch (GAN).
* Tìm hiểu, cài đặt thực nghiệm và đánh giá các công cụ phần mềm được sử dụng trong nhận diện khuôn mặt (Dlib, RetinaFace, MTCNN, FaceAPI)
* Xây dựng hệ thống nhận diện khuôn mặt trả về thông tin sử dụng thư viện Dlib và Elasticsearch.
* Xây dựng app android dùng để demo nhận diện khuôn mặt và trả về thông tin.
* Hướng nghiên cứu trong tương lai
* Cải thiện độ chính xác và tốc độ nhận diện: Nghiên cứu để tăng tính nhất quán và đáng tin cậy trong việc nhận diện khuôn mặt nhanh hơn và chính xác hơn.
* Tự động Realtime với Camera để nhận diện ngay lập tức khuôn mặt và định danh sinh viên.
* Áp dụng điểm danh sinh viên khi lên lớp học.
* Tích hợp với các mô hình khác để xây dựng ứng dụng thông minh hơn, ví dụ như Face ID trên các thiết bị di động.
* Điều chỉnh cho nhu cầu sử dụng cụ thể: cho phép nhận diện được khuôn mặt khi người đó đeo khẩu trang hoặc mang kính.
* Phát triển trên các nền tảng khác nhau.

# TÀI LIỆU THAM KHẢO

[1].ThS.Trần Hùng Cường, Ths.Nguyễn Phương Nga (2014), *Giáo trình trí tuệ nhân tạo,* Đại học công nghiệp Hà Nội.

[2]. <https://justadudewhohacks.github.io/face-api.js/docs/index.html>

[3]. <https://icsdweb.aegean.gr/awid/download-dataset>*.*

[4]. <https://github.com/serengil/retinaface>

[5]. <https://topdev.vn/blog/elasticsearch-la-gi/>

[6]. <http://dlib.net/>

[7]. <https://sefiks.com/2020/09/09/deep-face-detection-with-mtcnn-in-python/>

[8]. <https://thigiacmaytinh.com/lam-quen-voi-thu-vien-dlib-python/>

[9]. <https://aws.amazon.com/vi/what-is/deep-learning/>