**TRƯỜNG ĐẠI HỌC CÔNG NGHỆ ĐÔNG Á**

**KHOA CÔNG NGHỆ THÔNG TIN**



**BÀI TẬP LỚN**

**HỌC PHẦN: XỬ LÝ ẢNH VÀ THỊ GIÁC MÁY TÍNH**

**ĐỀ TÀI SỐ 17: XÂY DỰNG HỆ THỐNG CẢI THIỆN ĐỘ PHÂN GIẢI ẢNH BẰNG CNN**

**Giảng viên hướng dẫn: Lương Thị Hồng Lan**

|  |  |  |  |
| --- | --- | --- | --- |
| **TT** | **Mã sinh viên** | **Sinh viên thực hiện** | **Lớp hành chính** |
| **1** | 20211420 | Phạm Duy Dũng | DCCNTT12.10.6 |
| **2** | 20211622 | Trần Thị Hương Mai | DCCNTT12.10.6 |
| **3** | 20211607 | Phạm Gia Nhật Minh | DCCNTT12.10.6 |
| **4** | 20211413 | Ngô Xuân Trường | DCCNTT12.10.6 |

**Bắc Ninh, năm 2024**

**TRƯỜNG ĐẠI HỌC CÔNG NGHỆ ĐÔNG Á**

**KHOA CÔNG NGHỆ THÔNG TIN**

**BÀI TẬP LỚN**

**HỌC PHẦN: XỬ LÝ ẢNH VÀ THỊ GIÁC MÁY TÍNH**

**ĐỀ TÀI SỐ 17: XÂY DỰNG HỆ THỐNG CẢI THIỆN ĐỘ PHÂN GIẢI ẢNH BẰNG CNN**

**Giảng viên hướng dẫn: Lương Thị Hồng Lan**

|  |  |  |  |
| --- | --- | --- | --- |
| **TT** | **Mã sinh viên** | **Sinh viên thực hiện** | **Lớp hành chính** |
| **1** | 20211420 | Phạm Duy Dũng | DCCNTT12.10.6 |
| **2** | 20211622 | Trần Thị Hương Mai | DCCNTT12.10.6 |
| **3** | 20211607 | Phạm Gia Nhật Minh | DCCNTT12.10.6 |
| **4** | 20211413 | Ngô Xuân Trường | DCCNTT12.10.6 |

|  |  |
| --- | --- |
|  |  |

**Bắc Ninh, năm 2024**

|  |  |
| --- | --- |
| TRƯỜNG ĐẠI HỌC CÔNG NGHỆ ĐÔNG Á  **KHOA CÔNG NGHỆ THÔNG TIN** | **KỲ THI KẾT THÚC HỌC PHẦN**  **HỌC KỲ 1, NĂM HỌC 2024** – **2025** |

|  |  |
| --- | --- |
| **PHIẾU CHẤM THI BÀI TẬP LỚN KẾT THÚC HỌC PHẦN**  **Mã đề thi: 17**  **Tên học phần: Xử lý ảnh và thị giác máy tính**  **Lớp Tín chỉ: DCCNTT12.10.6** | |
| **Cán bộ chấm thi 1**  *(Ký và ghi rõ họ tên)*  *Lương Thị Hồng Lan* | **Cán bộ chấm thi 2**  *(Ký và ghi rõ họ tên)* |

| **TT** | **TIÊU CHÍ** | **THANG ĐIỂM** | Phạm Duy Dũng | Trần Thị Hương Mai | Phạm Gia Nhật Minh | Ngô Xuân Trường |
| --- | --- | --- | --- | --- | --- | --- |
| 20211420 | 20211622 | 20211607 | 20211413 |
| **1** | **Nội dung báo cáo trên Word đầy đủ** | **3.5** |  |  |  |  |
| 1.1 | Có bố cục rõ ràng (mục lục, phần mở đầu, nội dung chính, kết luận). | 0,5 |  |  |  |  |
| 1.2 | Nội dung phân tích rõ ràng, logic. | 0,5 |  |  |  |  |
| 1.3 | Có dẫn chứng, số liệu minh họa đầy đủ. | 0,5 |  |  |  |  |
| 1.4 | Ngôn ngữ và trình bày chuẩn, không lỗi chính tả. | 0,5 |  |  |  |  |
| 1.5 | Có trích dẫn tài liệu tham khảo đúng quy cách. | 0,5 |  |  |  |  |
| 1.6 | Được trình bày chuyên nghiệp (canh lề, font chữ, khoảng cách dòng hợp lý). | 0,5 |  |  |  |  |
| 1.7 | Tài liệu đầy đủ, bám sát yêu cầu của đề bài. | 0,5 |  |  |  |  |
| **2** | **Nội dung thuyết trình đầy đủ** | **1.0** |  |  |  |  |
| 2.1 | Trình bày tự tin, phát âm rõ ràng, mạch lạc. | 0,5 |  |  |  |  |
| 2.2 | Nội dung thuyết trình đúng trọng tâm, không lan man. | 0,5 |  |  |  |  |
| **3** | **Slides báo cáo đầy đủ nội dung + Hỏi đáp** | **3.0** |  |  |  |  |
| 3.1 | Slides có bố cục rõ ràng (mở đầu, nội dung, kết luận). | 0,5 |  |  |  |  |
| 3.2 | Thiết kế slides đẹp, chuyên nghiệp (màu sắc, hình ảnh minh họa). | 0,5 |  |  |  |  |
| 3.3 | Nội dung trên slides ngắn gọn, dễ hiểu, súc tích. | 0,5 |  |  |  |  |
| 3.4 | Nội dung slides phù hợp với nội dung báo cáo. | 0,5 |  |  |  |  |
| 3.5 | Trả lời câu hỏi đầy đủ, chính xác. | 0,5 |  |  |  |  |
| 3.6 | Trả lời câu hỏi tự tin, thuyết phục. | 0,5 |  |  |  |  |
| **4** | **Code đầy đủ** | **2.5** |  |  |  |  |
| 1.1 | Code được trình bày rõ ràng, có chú thích đầy đủ. | 0,5 |  |  |  |  |
| 1.2 | Code chạy đúng, không lỗi. | 0,5 |  |  |  |  |
| 1.3 | Code tối ưu, không dư thừa. | 0,5 |  |  |  |  |
| 1.4 | Đáp ứng đầy đủ các yêu cầu chức năng theo đề bài. | 0,5 |  |  |  |  |
| 1.5 | Có tính sáng tạo hoặc cải thiện so với yêu cầu. | 0,5 |  |  |  |  |
| **TỔNG ĐIỂM BẰNG SỐ:** | | **10** |  |  |  |  |
| **TỔNG ĐIỂM BẰNG CHỮ:** | | *Mười tròn* |  |  |  |  |

Mục lục

[Lời mở đầu 7](#_Toc184559054)

[Chương 1: Cơ sở lý thuyết 8](#_Toc184559055)

[1.1 Tổng quan về môn học 8](#_Toc184559056)

[1.1.1Thị giác máy tính 8](#_Toc184559057)

[1.2 Các thuật toán 9](#_Toc184559058)

[1.2.1 Thuật toán Support Vector Machine (SVM) 9](#_Toc184559059)

[1.2.2 Thuật toán Logictics Regression 11](#_Toc184559060)

[1.2.3 Thuật toán Naive Bayes (MultinomialNB) 14](#_Toc184559061)

[1.2.4 Thuật toán K-Nearest Neighbors(KNN) 15](#_Toc184559062)

[1.2.5 Thuật toán K-MEAN 17](#_Toc184559063)

[Chương 2: Xây dựng hệ thống 19](#_Toc184559064)

[2.1 Mô tả bài toán 19](#_Toc184559065)

[2.2 Xây dựng hệ thống 21](#_Toc184559066)

[2.2.1 Các thư viện 21](#_Toc184559067)

[2.2.2 Ngôn ngữ lập trình 23](#_Toc184559068)

[2.2.3 Hệ thống các thuật toán 24](#_Toc184559069)

[Chương 3: Kết quả thực nghiệm 26](#_Toc184559070)

[3.1 Dữ liệu 26](#_Toc184559071)

[3.1.1 Dữ liệu huấn luyện 26](#_Toc184559072)

[3.1.2 Luồng dữ liệu 27](#_Toc184559073)

[3.2 Kết quả thực nghiệm 27](#_Toc184559074)

[3.2.1 Màn hình thao tác người dùng 27](#_Toc184559075)

[3.2.2 Kết quả sau khi chạy chương trình 28](#_Toc184559076)

[Kết luận 29](#_Toc184559077)

[1. Mục tiêu và kết quả đạt được 29](#_Toc184559078)

[2. Ưu điểm 29](#_Toc184559079)

[3. Hạn chế 29](#_Toc184559080)

[4. Hướng phát triển 29](#_Toc184559081)

[Tài liệu tham khảo 30](#_Toc184559082)

# Lời mở đầu

Trong thời đại công nghệ số hiện nay, hình ảnh đóng vai trò quan trọng trong nhiều lĩnh vực như y tế, giám sát an ninh, truyền thông, và giải trí. Tuy nhiên, việc thu thập dữ liệu hình ảnh thường bị giới hạn bởi chất lượng của thiết bị hoặc điều kiện môi trường, dẫn đến hình ảnh có độ phân giải thấp, gây khó khăn trong việc phân tích và xử lý. Vì vậy, việc cải thiện độ phân giải hình ảnh không chỉ là một yêu cầu thực tiễn mà còn là một bài toán nghiên cứu đầy thách thức.

Trong bối cảnh đó, mạng nơ-ron tích chập (Convolutional Neural Network - CNN) nổi lên như một công cụ mạnh mẽ để giải quyết bài toán tăng cường độ phân giải ảnh. Nhờ khả năng học đặc trưng và tái tạo chi tiết vượt trội, CNN đã đạt được những thành tựu đáng kể trong việc cải thiện chất lượng ảnh.

Đề tài "Xây dựng hệ thống cải thiện độ phân giải ảnh bằng CNN" hướng đến việc nghiên cứu và triển khai một hệ thống có khả năng nâng cao độ phân giải của ảnh từ mức thấp lên mức cao hơn, nhằm đáp ứng yêu cầu trong các ứng dụng thực tế. Qua đó, đề tài sẽ khám phá cách tối ưu hóa mạng CNN để cải thiện hiệu suất, cũng như đánh giá hiệu quả của mô hình qua các thí nghiệm thực nghiệm.

Hy vọng rằng, với cách tiếp cận này, nghiên cứu sẽ góp phần cung cấp giải pháp hiệu quả cho bài toán xử lý ảnh, mở ra tiềm năng ứng dụng rộng rãi trong nhiều lĩnh vực.

# Chương 1: Cơ sở lý thuyết

## 1.1 Tổng quan về môn học

### 1.1.1Thị giác máy tính

**Định nghĩa**

Thị giác máy tính là một lĩnh vực của trí tuệ nhân tạo (AI) tập trung vào việc giúp máy tính thu nhận, xử lý và diễn giải dữ liệu từ hình ảnh hoặc video. Với mục tiêu cải thiện chất lượng và phân tích nội dung ảnh, các kỹ thuật hiện đại như mạng nơ-ron tích chập (Convolutional Neural Networks - CNN) đã trở thành nền tảng quan trọng trong việc xây dựng hệ thống cải thiện độ phân giải ảnh.

**Mục tiêu chính:**

* Tăng cường độ phân giải và chi tiết của hình ảnh thông qua các thuật toán học sâu, đặc biệt là CNN.
* Tự động hóa việc khôi phục và nâng cao chất lượng hình ảnh để phục vụ các ứng dụng trong thực tế.
* Các ứng dụng phổ biến của việc cải thiện độ phân giải ảnh:
* Nâng cao chất lượng video: Dùng trong truyền phát video hoặc chỉnh sửa hậu kỳ.
* Hình ảnh y tế: Cải thiện độ rõ nét trong hình ảnh CT, MRI để hỗ trợ chẩn đoán chính xác hơn.
* Thực tế ảo và tăng cường (VR/AR): Đảm bảo chất lượng hình ảnh mượt mà và chi tiết trong môi trường ảo.
* Phân tích ảnh vệ tinh: Nâng cao độ chi tiết của hình ảnh phục vụ trong nông nghiệp, quân sự, và môi trường.
* Phục chế ảnh cũ: Khôi phục và làm sắc nét những hình ảnh lịch sử, bị mờ hoặc tổn hại theo thời gian.

**Các bài toán chính liên quan đến cải thiện độ phân giải ảnh:**

* Super-Resolution (Siêu phân giải): Dự đoán và tái tạo các chi tiết bị mất trong hình ảnh có độ phân giải thấp.
* Noise Reduction (Giảm nhiễu): Loại bỏ nhiễu trong hình ảnh mà vẫn giữ được chi tiết.
* Image Denoising (Khử nhiễu ảnh): Kết hợp với các kỹ thuật tăng cường độ phân giải để cải thiện chất lượng tổng thể của hình ảnh.
* Kỹ thuật CNN áp dụng trong cải thiện độ phân giải ảnh
* Super-Resolution CNN (SRCNN): Mô hình CNN cơ bản đầu tiên ứng dụng trong việc cải thiện độ phân giải ảnh.
* Enhanced Deep Super-Resolution (EDSR): Mô hình nâng cao loại bỏ các phần dư thừa trong mạng để tập trung vào chất lượng đầu ra.
* Generative Adversarial Networks (GANs): Ứng dụng GAN để tạo ra các chi tiết chân thực hơn khi tăng độ phân giải ảnh.
* Residual Networks (ResNet): Tăng hiệu quả của CNN bằng cách xử lý thông tin trong các tầng sâu hơn mà không làm mất mát dữ liệu.
* Lợi ích của hệ thống cải thiện độ phân giải ảnh bằng CNN
* Tăng cường chất lượng hình ảnh: Phục hồi chi tiết bị mất một cách chính xác và tự nhiên.
* Tự động hóa: Giảm thiểu sự can thiệp thủ công trong các công đoạn chỉnh sửa ảnh.
* Ứng dụng rộng rãi: Từ công nghiệp, y tế, giải trí đến nghiên cứu khoa học.

**Công cụ và thư viện phổ biến:**

* OpenCV: Hỗ trợ các bước xử lý tiền đề trước khi áp dụng mạng CNN.
* TensorFlow/Keras: Phát triển và huấn luyện các mô hình CNN cải thiện độ phân giải ảnh.
* PyTorch: Một framework mạnh mẽ khác để triển khai các mô hình học sâu.
* ESRGAN: Một mô hình GAN tiên tiến trong việc cải thiện độ phân giải ảnh.

## 1.2 Các thuật toán

### 1.2.1 Thuật toán Support Vector Machine (SVM)

* Định nghĩa

Support Vector Machine (SVM) là một mô hình phân loại hoạt động bằng việc xây dựng một siêu phẳng (hyperplane) có (n - 1) chiều trong không gian n chiều của dữ liệu sao cho siêu phẳng này phân loại các lớp một cách tối ưu nhất. Nói cách khác, cho một tập dữ liệu có nhãn (học có giám sát), thuật toán sẽ dựa trên dữ liệu học để xây dựng một siêu phẳng tối ưu được sử dụng để phân loại dữ liệu mới. Ở không gian 2 chiều thì siêu phẳng này là 1 đường thẳng phân cách chia mặt phẳng không gian thành 2 phần tương ứng 2 lớp với mỗi lớp nằm ở 1 phía của đường thẳng.

* Ví dụ minh họa

Ta có các điểm dữ liệu như hình dưới đây với mỗi điểm thuộc 1 trong 2 lớp cho trước:

Ảnh có chứa văn bản, hàng, biểu đồ, ảnh chụp màn hình

Mô tả được tạo tự động

Đường thẳng này chia cách khá tốt 2 lớp trên mặt phẳng. Tất cả những điểm dữ liệu nằm bên trái đường thẳng đều thuộc về lớp hình tròn và những điểm nằm ở bên phải thuộc về lớp hình vuông. Nhiệm vụ của SVM chính là tìm ra đường thẳng / siêu phẳng phân cách cách sao cho phân loại dữ liệu tốt nhất có thể.

* Ưu điểm của thuật toán

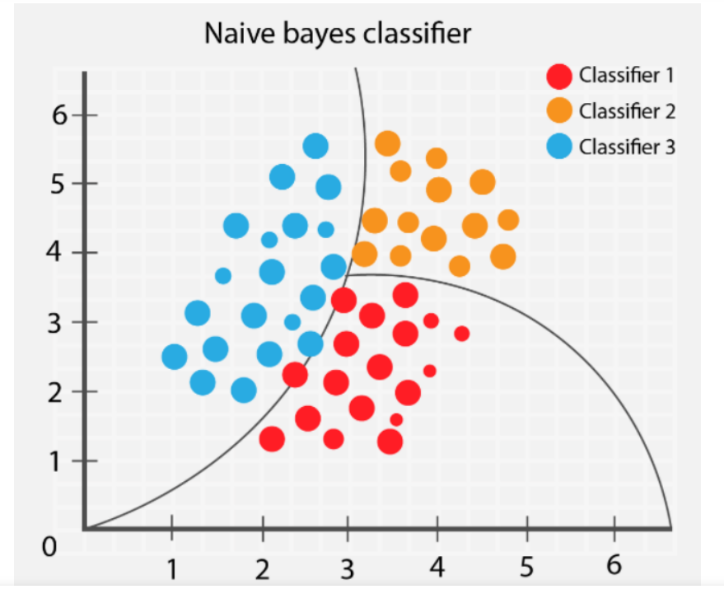
Là một kĩ thuật phân lớp khá phổ biến, SVM thể hiện được nhiều ưu điểm trong số đó có việc tính toán hiệu quả trên các tập dữ liệu lớn. Có thể kể thêm một số ưu điểm của phương pháp này như:

* Xử lý trên không gian số chiều cao: SVM là một công cụ tính toán hiệu quả trong không gian chiều cao, trong đó đặc biệt áp dụng cho các bài toán phân loại văn bản và phân tích quan điểm nơi chiều có thể cực kỳ lớn
* Tiết kiệm bộ nhớ: Do chỉ có một tập hợp con của các điểm được sử dụng trong quá trình huấn luyện và ra quyết định thực tế cho các điểm dữ liệu mới nên chỉ có những điểm cần thiết mới được lưu trữ trong bộ nhớ khi ra quyết dịnh
* Tính linh hoạt - phân lớp thường là phi tuyến tính. Khả năng áp dụng Kernel mới cho phép linh động giữa các phương pháp tuyến tính và phi tuyến tính từ đó khiến cho hiệu suất phân loại lớn hơn.
* Nhược điểm của thuật toán
* Bài toán số chiều cao: Trong trường hợp số lượng thuộc tính (p) của tập dữ liệu lớn hơn rất nhiều so với số lượng dữ liệu (n) thì SVM cho kết quả khá tồi
* Chưa thể hiện rõ tính xác suất: Việc phân lớp của SVM chỉ là việc cố gắng tách các đối tượng vào hai lớp được phân tách bởi siêu phẳng SVM. Điều này chưa giải thích được xác suất xuất hiện của một thành viên trong một nhóm là như thế nào. Tuy nhiên hiệu quả của việc phân lớp có thể được xác định dựa vào khái niệm margin từ điểm dữ liệu mới đến siêu phẳng phân lớp mà chúng ta đã bàn luận ở trên.

### 1.2.2 Thuật toán Logictics Regression

* Định nghĩa

Thuật toán Logistic Regression (Hồi quy Logistic) là một phương pháp thống kê được sử dụng trong bài toán phân loại, đặc biệt là trong trường hợp phân loại nhị phân (hai nhãn). Nó được sử dụng để dự đoán xác suất của một sự kiện xảy ra dựa trên một tập hợp các đặc trưng (biến độc lập) liên quan.



* Cách thức hoạt động

Mô hình Logistic Regression hoạt động bằng cách ước lượng xác suất xảy ra của một sự kiện dựa trên một tập hợp các đặc trưng (biến độc lập).

Đầu vào: Tập dữ liệu huấn luyện gồm các mẫu có đặc trưng (biến độc lập) và nhãn (biến phụ thuộc) đã được gán nhãn cho từng lớp phân loại.

Bước 1: Chuẩn bị dữ liệu

* Tiền xử lý dữ liệu: Bao gồm chuẩn hóa đặc trưng, xử lý các giá trị thiếu, và chia thành tập huấn luyện và tập kiểm tra.

Bước 2: Khởi tạo tham số

* Khởi tạo trọng số w và bias b ban đầu. Các giá trị ban đầu có thể là 0 hoặc được chọn ngẫu nhiên.

Bước 3: Tính toán đầu ra và xác suất

* Tính tổng trọng số z bằng cách lấy tổng tích của giá trị đặc trưng và trọng số tương ứng: z = w₁x₁ + w₂x₂ + ... + wₙxₙ + b
* Áp dụng hàm sigmoid cho tổng trọng số để tính toán xác suất dự đoán: p = 1 / (1 + e^(-z))

Bước 4: Tính toán hàm mất mát

* Sử dụng hàm mất mát (thường là Cross-Entropy Loss) để đánh giá độ sai khác giữa giá trị dự đoán và nhãn thực tế: J(w) = -(1/N) \* Σ[yᵢ \* log(pᵢ) + (1 - yᵢ) \* log(1 - pᵢ)]

Bước 5: Cập nhật tham số

* Sử dụng phương pháp tối ưu hóa như Gradient Descent, cập nhật các trọng số để giảm hàm mất mát: wⱼ = wⱼ - α \* (∂J(w) / ∂wⱼ) và b = b - α \* (∂J(w) / ∂b), trong đó α là learning rate (tỷ lệ học tập).

Bước 6: Lặp lại quá trình huấn luyện

* Tiếp tục lặp lại các bước trên bằng cách cập nhật tham số cho đến khi đạt được sự hội tụ hoặc điều kiện dừng khác được đáp ứng.

Bước 7: Đánh giá mô hình

* Sử dụng tập dữ liệu kiểm tra để đánh giá hiệu suất của mô hình, bao gồm độ chính xác (accuracy) và các độ đo đánh giá khác như độ phủ (recall), độ chính xác (precision), F1-score, v.v.

Ảnh có chứa Phông chữ, màu trắng, thư pháp, biểu tượng

Mô tả được tạo tự động

* Ưu điểm của thuật toán
* Đơn giản và dễ hiểu: Logistic Regression là một thuật toán đơn giản và dễ hiểu, không đòi hỏi kiến thức toán cao cấp và có thể được áp dụng một cách nhanh chóng.
* Tính linh hoạt: Thuật toán Logistic Regression có thể được áp dụng vào các bài toán phân loại nhị phân và đa nhãn. Nó cũng có thể được mở rộng để xử lý các vấn đề phân loại đa lớp thông qua các phương pháp như One-vs-Rest hoặc Softmax Regression.
* Xử lý tốt với dữ liệu lớn: Logistic Regression có thể xử lý tốt với các tập dữ liệu lớn mà không gặp vấn đề về hiệu suất. Thuật toán có thể được tối ưu hóa và huấn luyện nhanh chóng trên các tập dữ liệu lớn.
* Có thể xác định độ quan trọng của đặc trưng: Logistic Regression cung cấp thông tin về độ quan trọng của từng đặc trưng trong việc phân loại. Điều này cho phép hiểu rõ hơn về ảnh hưởng của từng đặc trưng lên kết quả dự đoán.
* Khả năng giải thích kết quả: Kết quả của Logistic Regression có thể được giải thích dễ dàng và logic, giúp hiểu rõ hơn về quyết định của mô hình. Trọng số của từng đặc trưng cung cấp thông tin về đóng góp của chúng vào kết quả phân loại.
* Tolerant với nhiễu và giá trị thiếu: Logistic Regression có khả năng xử lý dữ liệu nhiễu và giá trị thiếu một cách tương đối tốt. Nó có khả năng chịu được sự nhiễu và thích nghi với các điểm dữ liệu bất thường.
* Nhược điểm của thuật toán
* Giới hạn trong phân loại phi tuyến: Logistic Regression mô hình hóa các quan hệ tuyến tính giữa các đặc trưng và đầu ra. Do đó, nó không thể xử lý tốt các quan hệ phi tuyến phức tạp giữa đặc trưng và nhãn.
* Dễ bị ảnh hưởng bởi giá trị nhiễu: Logistic Regression có thể nhạy cảm với các giá trị nhiễu hoặc điểm dữ liệu bất thường. Các điểm nhiễu có thể có ảnh hưởng lớn đến hướng của ranh giới phân loại và có thể dẫn đến kết quả dự đoán không chính xác.
* Khả năng bị overfitting: Nếu không kiểm soát tốt việc quá khớp (overfitting), Logistic Regression có thể tạo ra một mô hình quá phức tạp và không thể tổng quát hóa tốt cho dữ liệu mới. Điều này xảy ra khi mô hình quá phức tạp so với sự phân bố thực sự của dữ liệu huấn luyện.
* Yêu cầu đặc trưng độc lập: Logistic Regression yêu cầu rằng các đặc trưng độc lập với nhau, tức là không có tương quan hoặc phụ thuộc tuyến tính mạnh giữa chúng. Nếu có sự tương quan mạnh hoặc đa cộng tuyến, kết quả dự đoán của mô hình có thể không chính xác hoặc không ổn định.
* Khả năng xử lý dữ liệu lớn bị hạn chế: Mặc dù Logistic Regression có khả năng xử lý tốt với dữ liệu lớn, nó có thể gặp khó khăn khi số lượng đặc trưng (biến) rất lớn so với kích thước mẫu huấn luyện. Điều này có thể dẫn đến hiện tượng quá khớp và yêu cầu các phương pháp xử lý dữ liệu phù hợp.
* Không thích hợp cho bài toán phân loại đa lớp: Logistic Regression ban đầu chỉ được thiết kế để giải quyết bài toán phân loại nhị phân. Mặc dù có thể mở rộng để xử lý bài toán đa lớp, nhưng không hiệu quả bằng các thuật toán phân loại

### 1.2.3 Thuật toán Naive Bayes (MultinomialNB)

* Định nghĩa

Naive Bayes là một thuật toán phân loại dựa trên nguyên tắc của định lý Bayes. Giả thuyết Naive Bayes giả định rằng các đặc trưng đầu vào là độc lập với nhau trong việc dự đoán lớp. MultinomialNB là một biến thể của Naive Bayes dùng cho dữ liệu có đặc trưng đếm được hoặc đặc trưng dạng đa thức.

* Quy trình làm việc
* Tiền xử lý dữ liệu: Chuyển đổi dữ liệu văn bản thành vectơ đặc trưng sử dụng phương pháp như TF-IDF (Term Frequency-Inverse Document Frequency) để biểu diễn tần suất của từng từ trong văn bản.
* Phân chia dữ liệu: Chia dữ liệu thành tập huấn luyện và tập kiểm tra.
* Xây dựng mô hình Naive Bayes
* Tính toán xác suất tiên nghiệm (prior probability) cho mỗi lớp dựa trên tập huấn luyện.
* Tính toán xác suất hậu nghiệm (posterior probability) cho từng lớp dựa trên dữ liệu đầu vào và giả thuyết Naive Bayes. Các xác suất này được tính bằng công thức Bayes.
* Xác định lớp dự đoán dựa trên xác suất hậu nghiệm cao nhất.
* Ưu điểm thuật toán
* Hiệu suất tính toán nhanh: Với giả thuyết độc lập Naive Bayes, thuật toán có thể tính toán các xác suất dự đoán một cách nhanh chóng. Điều này làm cho MultinomialNB phù hợp với việc xử lý tập dữ liệu lớn.
* Đơn giản và dễ triển khai: Naive Bayes dựa trên một số giả định đơn giản, dễ hiểu và dễ triển khai. Việc xây dựng mô hình và áp dụng thuật toán cũng khá dễ dàng, không yêu cầu quá nhiều kiến thức toán học phức tạp.
* Độ chính xác tốt với dữ liệu lớn: Mặc dù giả thuyết Naive Bayes không thể đáp ứng đầy đủ với sự phụ thuộc giữa các đặc trưng, nhưng trong thực tế nó có thể mang lại kết quả phân loại tốt, đặc biệt là với tập dữ liệu lớn.
* Nhược điểm thuật toán
* Giả định độc lập không thực tế: Giả định Naive Bayes về sự độc lập giữa các đặc trưng có thể không phù hợp với một số bài toán thực tế, khi các đặc trưng có sự tương quan hoặc phụ thuộc vào nhau.
* Dễ bị ảnh hưởng bởi dữ liệu thiếu: Naive Bayes có thể bị ảnh hưởng bởi các từ hoặc đặc trưng không xuất hiện trong tập huấn luyện. Nếu một từ chưa được quan sát trong quá trình huấn luyện, xác suất ước lượng sẽ bằng 0 và gây ra lỗi.

Khả năng xấp xỉ xác suất: Trong một số trường hợp, Naive Bayes có thể xấp xỉ xác suất không chính xác. Điều này có thể xảy ra khi tỷ lệ mẫu giữa các lớp không cân bằng hoặc khi có các đặc trưng không tương thích với giả thuyết độc lập

### 1.2.4 Thuật toán K-Nearest Neighbors(KNN)

* Định nghĩa:

KNN (K-Nearest Neighbors) là một thuật toán học máy thuộc nhóm học không tham số (non-parametric) và học dựa trên trường hợp (instance-based learning). KNN được sử dụng cho cả bài toán phân loại và hồi quy.

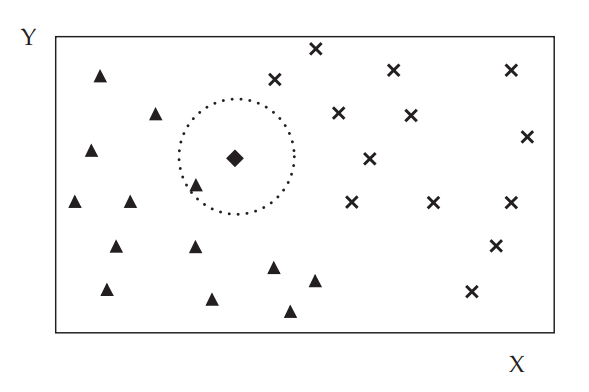
* Cách hoạt động

Giai đoạn chuẩn bị:

* Tập dữ liệu huấn luyện được lưu giữ và không trải qua bất kỳ quá trình huấn luyện nào (học dựa trên ví dụ).

Dự đoán

* Tính khoảng cách giữa điểm cần dự đoán và tất cả các điểm trong tập huấn luyện (sử dụng các phương pháp như khoảng cách Euclidean, Manhattan, hoặc Cosine).
* Chọn K điểm gần nhất dựa trên khoảng cách.
* Dựa vào K điểm này, dự đoán nhãn (phân loại) hoặc giá trị (hồi quy).



* Ưu điểm
* Dễ hiểu và triển khai: Thuật toán đơn giản, dễ cài đặt mà không cần nhiều giả định.
* Linh hoạt: Áp dụng được cho cả bài toán phân loại và hồi quy. Hoạt động tốt với các tập dữ liệu nhỏ và không yêu cầu đào tạo mô hình.
* Không cần giả định phân phối: Không yêu cầu dữ liệu tuân theo phân phối chuẩn, phù hợp với nhiều loại dữ liệu
* Tự động thích nghi: Nếu dữ liệu thay đổi (bổ sung hoặc xóa), thuật toán vẫn có thể hoạt động mà không cần huấn luyện lại.
* Nhược điểm của KNN
* Hiệu suất kém trên dữ liệu lớn:

KNN đòi hỏi tính toán khoảng cách với tất cả các điểm trong tập huấn luyện, khiến thời gian xử lý chậm trên dữ liệu lớn.

* Nhạy cảm với dữ liệu nhiễu:

Dữ liệu nhiễu hoặc không cân bằng có thể làm giảm độ chính xác, đặc biệt khi chọn K không phù hợp.

* Phụ thuộc vào giá trị K:

Giá trị K quá nhỏ có thể khiến mô hình bị overfitting (quá khớp).

Giá trị K quá lớn có thể làm giảm độ chính xác (underfitting).

Khó khăn trong xử lý dữ liệu đa chiều:

Khi dữ liệu có số chiều cao (high dimensionality), khoảng cách giữa các điểm trở nên khó đo lường chính xác (curse of dimensionality).

* Nhạy cảm với việc chuẩn hóa dữ liệu:

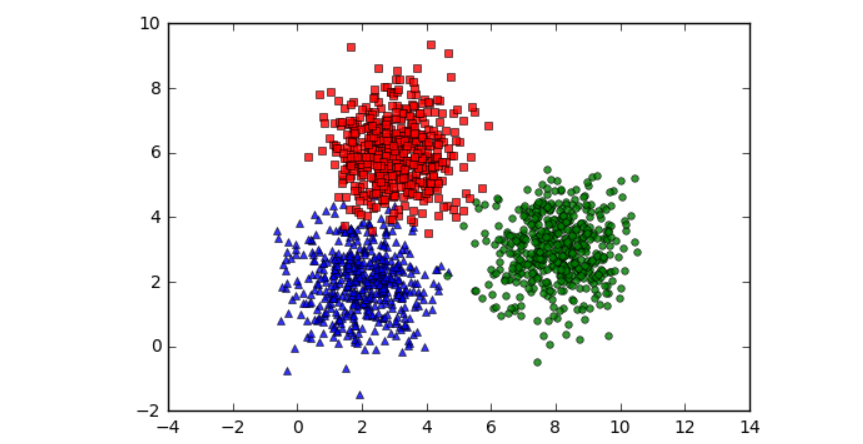
Các đặc trưng có đơn vị lớn hơn có thể chi phối kết quả. Cần chuẩn hóa dữ liệu trước khi sử dụng KNN.

### 1.2.5 Thuật toán K-MEAN

* Định nghĩa

K-Means là một thuật toán học không giám sát (unsupervised learning), được sử dụng phổ biến trong bài toán phân cụm (clustering). Mục tiêu của K-Means là chia tập dữ liệu thànhK nhóm (cụm) sao cho các điểm dữ liệu trong cùng một cụm có độ tương đồng cao nhất, và khác biệt so với các cụm khác.

* Các bước hoạt động:
* Chọn số lượng nhóm cần phân cụm.
* Chọn ngẫu nhiên các điểm trung tâm ban đầu cho từng nhóm.
* Gán từng điểm dữ liệu vào nhóm có khoảng cách gần nhất với điểm đó.
* Cập nhật lại vị trí các điểm trung tâm bằng cách tính lại trung bình các điểm trong từng nhóm.
* Lặp lại các bước 3 và 4 cho đến khi các điểm trung tâm không thay đổi nhiều hoặc đạt được số lần lặp tối đa.

****

* Ưu điểm
* Dễ hiểu và triển khai: Thuật toán K-means khá đơn giản và dễ dàng để lập trình, phù hợp với cả người mới bắt đầu.
* Tính hiệu quả caoThời gian chạy nhanh, đặc biệt với các tập dữ liệu lớn. K-means có độ phức tạp tính toán thấp
* Thích hợp với các bài toán phân cụm cơ bản: K-means hoạt động tốt trên các bộ dữ liệu có cấu trúc rõ ràng và các cụm có dạng hình cầu, đồng nhất về kích thước.
* Khả năng mở rộng: Thuật toán có thể mở rộng để xử lý dữ liệu lớn bằng các biến thể như Mini-Batch K-means hoặc áp dụng trên nền tảng phân tán như Hadoop hoặc Spark.
* Ứng dụng đa dạng: Sử dụng trong nhiều lĩnh vực như xử lý ảnh (phân vùng ảnh), phát hiện bất thường, phân nhóm khách hàng, và giảm chiều dữ liệu.
* Nhược điểm
* Phụ thuộc vào số lượng cụm: Thuật toán yêu cầu phải xác định trước số lượng cụm
* Nhạy cảm với điểm khởi tạo: Kết quả phân cụm có thể thay đổi nếu các tâm cụm ban đầu được chọn khác nhau. Điều này dẫn đến thuật toán có thể hội tụ đến cực trị cục bộ thay vì cực trị toàn cục.
* Giới hạn về hình dạng cụm: K-means hoạt động tốt với các cụm hình cầu và có kích thước tương tự. Tuy nhiên, với dữ liệu có hình dạng cụm phức tạp (hình elip, không đồng đều), K-means có thể không hiệu quả.
* Không phù hợp với dữ liệu không đồng nhất hoặc dữ liệu bị nhiễu: K-means nhạy cảm với dữ liệu ngoại lai (outliers). Một điểm nhiễu có thể làm lệch tâm cụm và ảnh hưởng đến kết quả.
* Không thích hợp với các tập dữ liệu không tuyến tính: Với các cụm không thể phân tách tuyến tính (như dữ liệu dạng mặt trăng hoặc hình vòng), K-means không thể nhận diện đúng cụm.
* Cần chuẩn hóa dữ liệu trước khi chạy: Nếu các đặc trưng dữ liệu có thang đo khác nhau, thuật toán có thể bị thiên lệch bởi các đặc trưng có giá trị lớn.

# Chương 2: Xây dựng hệ thống

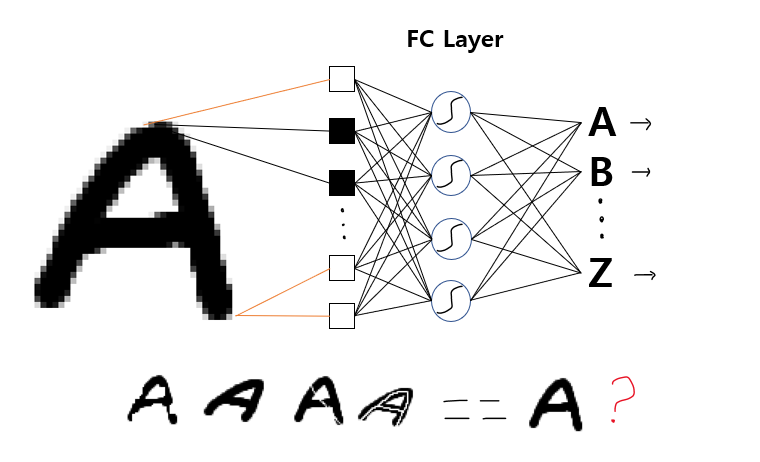
## 2.1 Mô tả bài toán

**Thuật toán Convolutional Neural Network (CNN)**

* Định nghĩa

CNN (Convolutional Neural Network) là một mô hình mạng nơ-ron sâu được sử dụng rộng rãi trong lĩnh vực xử lí ảnh và thị giác máy tính. Nó được thiết kế để tự động rút trích và hiểu các đặc trưng từ dữ liệu hình ảnh.

CNN là một kiến trúc mạng nơ-ron sâu có khả năng tự học các đặc trưng từ dữ liệu hình ảnh thông qua các lớp tích chập và pooling. Nó sử dụng các bộ lọc tích chập để tìm kiếm các đặc trưng như cạnh, góc, texture, và các đặc trưng cao cấp hơn từ các vùng nhỏ của hình ảnh. Sau đó, thông qua các lớp pooling, CNN giảm kích thước của đặc trưng và tạo ra các biểu diễn tổng quát của hình ảnh.



* Quy trình làm việc

Bước 1: Tiền xử lí: Ảnh đầu vào được chuẩn hóa kích thước và điều chỉnh độ sáng.

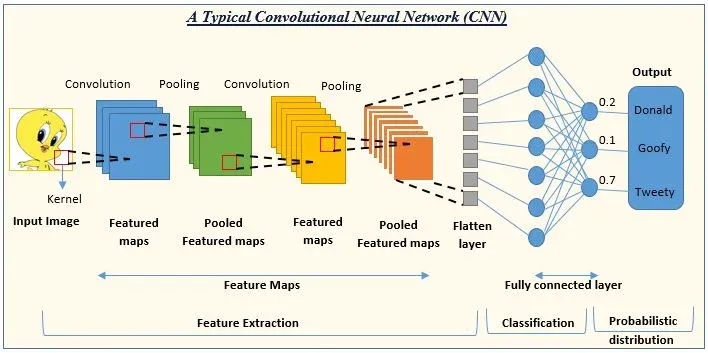
Bước 2: Lớp tích chập: Các bộ lọc tích chập được áp dụng trên ảnh để rút trích các đặc trưng như cạnh, góc, texture. Các lớp tích chập này bao gồm các bộ lọc với trọng số được học để tìm ra các đặc trưng quan trọng.

Bước 3: Lớp pooling: Lớp pooling giảm kích thước của đặc trưng bằng cách giữ lại các giá trị quan trọng nhất. Các phép lọc pooling thường là phép lọc tối đa (max pooling) hoặc phép lọc trung bình (average pooling).

Bước 4: Flatten: Các đặc trưng được làm phẳng thành một vectơ để đưa vào các lớp kết nối đầy đủ (fully connected layers).

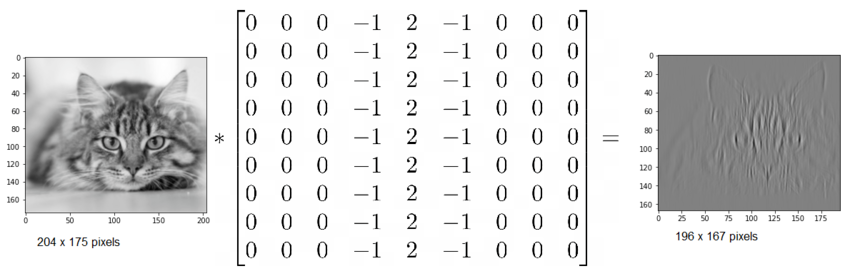
Bước 5: Lớp kết nối đầy đủ: Lớp kết nối đầy đủ nhận các đặc trưng đã được làm phẳng và thực hiện phân loại các lớp dựa trên các đặc trưng đó.

Bước 6: Đầu ra: Đầu ra cuối cùng của CNN là xác suất phân loại cho mỗi lớp.



* Ví dụ minh họa

Một ví dụ đơn giản về CNN là việc phân loại ảnh chó và mèo. CNN sẽ học các đặc trưng như mắt, mũi, tai và các chi tiết khác của chó và mèo từ các bức ảnh huấn luyện. Sau khi huấn luyện, CNN có thể dự đoán xác suất một bức ảnh mới là chó hoặc mèo.



* Ưu điểm của CNN
* Có khả năng tự động rút trích các đặc trưng từ dữ liệu hình ảnh.
* Phù hợp với việc xử lý dữ liệu ảnh 2D và dữ liệu không gian.
* Hiệu suất tốt trong việc phân loại hình ảnh và nhận dạng đối tượng.
* Giảm số lượng tham số cần huấn luyện thông qua các lớp tích chập và pooling.
* Nhược điểm của CNN
* Yêu cầu lượng dữ liệu huấn luyện lớn để đạt được hiệu suất tốt.
* Đòi hỏi tài nguyên tính toán cao, đặc biệt là khi mạng rất sâu và kích thước hình ảnh lớn.
* Khả năng hiểu và diễn giải các quyết định của CNN không được rõ ràng.

## 2.2 Xây dựng hệ thống

### 2.2.1 Các thư viện

* PyTorch
* Khái niệm:

PyTorch là một thư viện mã nguồn mở dành cho học máy và học sâu, được phát triển bởi Facebook's AI Research lab (FAIR). Nó chủ yếu được sử dụng để xây dựng và huấn luyện các mô hình học sâu (deep learning) như mạng nơ-ron nhân tạo (ANN), mạng nơ-ron tích chập (CNN), và mạng nơ-ron hồi tiếp (RNN).

* Ưu điểm:
* Dễ sử dụng: PyTorch có cú pháp đơn giản và dễ hiểu, rất phù hợp với các nhà nghiên cứu và lập trình viên mới bắt đầu.
* Tính động (Dynamic Computation Graph): PyTorch sử dụng computation graph động, cho phép xây dựng và sửa đổi mô hình một cách linh hoạt trong quá trình huấn luyện.
* Tích hợp GPU: PyTorch hỗ trợ mạnh mẽ việc tính toán trên GPU, giúp tăng tốc huấn luyện mô hình, đặc biệt khi làm việc với các bộ dữ liệu lớn.
* Cộng đồng mạnh mẽ: PyTorch có một cộng đồng người dùng và nhà phát triển lớn, cung cấp nhiều tài nguyên, tài liệu, và các mô hình đã được huấn luyện sẵn.
* Tính mở rộng và dễ bảo trì: PyTorch dễ dàng tích hợp với các thư viện và công cụ học máy khác, đồng thời dễ bảo trì khi phát triển các mô hình phức tạp.
* Nhược điểm:
* Thiếu tính ổn định trong môi trường sản xuất: Mặc dù PyTorch rất mạnh mẽ cho nghiên cứu và thử nghiệm, nhưng TensorFlow vẫn thường được ưa chuộng hơn trong môi trường sản xuất.
* Không tối ưu cho mobile: Mặc dù PyTorch đã bắt đầu hỗ trợ triển khai trên mobile, nhưng về cơ bản, nó vẫn chưa tối ưu bằng TensorFlow khi triển khai trên các thiết bị di động.
* PIL
* Khái niệm:

PIL (Python Imaging Library) là một thư viện xử lý ảnh trong Python, giúp làm việc với các loại ảnh khác nhau, thực hiện các thao tác như mở, lưu, cắt, thay đổi kích thước, v.v. Pillow là một fork của PIL, hiện tại đang là thư viện phổ biến nhất cho xử lý ảnh trong Python.

* Ưu điểm:
* Đơn giản và dễ sử dụng: Pillow cung cấp API đơn giản và dễ tiếp cận cho các tác vụ xử lý ảnh cơ bản.
* Hỗ trợ đa định dạng: Pillow hỗ trợ rất nhiều định dạng ảnh như JPEG, PNG, GIF, TIFF, BMP, và nhiều hơn nữa.
* Chuyển đổi ảnh giữa các không gian màu: Pillow có thể dễ dàng chuyển đổi giữa các không gian màu như RGB, RGBA, L (grayscale), v.v.
* Tiết kiệm tài nguyên: Pillow rất nhẹ và nhanh chóng cho các tác vụ xử lý ảnh cơ bản.
* Nhược điểm:
* Không mạnh mẽ cho các tác vụ phức tạp: Pillow chỉ phù hợp với các tác vụ xử lý ảnh cơ bản, nếu bạn cần thực hiện các thao tác phức tạp như phân tích ảnh hoặc nhận diện đối tượng, bạn sẽ cần một thư viện mạnh mẽ hơn như OpenCV.
* Hạn chế về khả năng xử lý video: Pillow không hỗ trợ xử lý video, vì vậy bạn sẽ cần thư viện khác nếu muốn làm việc với video.
* Numpy
* Khái niệm:

NumPy là thư viện tính toán khoa học trong Python, cung cấp các công cụ mạnh mẽ để xử lý các mảng và ma trận, cũng như các phép toán số học hiệu quả. Đây là một trong những thư viện cơ bản nhất và quan trọng nhất cho khoa học dữ liệu và học máy.

* Ưu điểm:
* Hiệu suất cao: NumPy cung cấp các phép toán vector hóa, giúp các phép tính trên mảng và ma trận trở nên nhanh chóng và hiệu quả.
* Được sử dụng rộng rãi: Là thư viện cốt lõi trong nhiều ứng dụng khoa học dữ liệu và học máy, NumPy có thể dễ dàng tích hợp với các thư viện khác như SciPy, scikit-learn, TensorFlow, và PyTorch.
* Mảng n-dimensional (ndarray): NumPy cho phép làm việc với các mảng n chiều, rất hữu ích cho việc xử lý dữ liệu ảnh và dữ liệu đa chiều.
* Hỗ trợ nhiều phép toán tuyến tính và thống kê: NumPy cung cấp nhiều hàm tính toán các phép toán như phép nhân ma trận, tính tổng, trung bình, chuẩn hóa, v.v.
* Nhược điểm:
* Không có hỗ trợ GPU: Mặc dù NumPy rất nhanh với các phép toán trên CPU, nhưng không hỗ trợ tính toán trên GPU. Để sử dụng GPU, bạn cần kết hợp với các thư viện như CuPy, TensorFlow hoặc PyTorch.
* Hạn chế trong xử lý dữ liệu không phải số học: NumPy được tối ưu cho dữ liệu số học và không hỗ trợ tốt cho các loại dữ liệu khác như chuỗi hoặc văn bản.
* Flask
* Khái niệm:

Flask là một micro web framework cho Python, cho phép bạn xây dựng các ứng dụng web nhanh chóng và dễ dàng. Flask nhẹ nhàng và không yêu cầu cấu trúc phức tạp, giúp bạn dễ dàng tùy chỉnh theo nhu cầu riêng của ứng dụng.

* Ưu điểm:
* Đơn giản và dễ sử dụng: Flask rất dễ học và sử dụng, với ít cấu hình yêu cầu. Nó là lựa chọn tuyệt vời cho những người mới bắt đầu phát triển ứng dụng web.
* Tính linh hoạt cao: Flask không yêu cầu bạn phải sử dụng bất kỳ thư viện hay công cụ nào ngoài Flask. Điều này mang lại sự linh hoạt lớn khi bạn muốn tùy chỉnh hoặc tích hợp thêm các công cụ khác.
* Cộng đồng hỗ trợ mạnh mẽ: Flask có một cộng đồng người dùng và nhà phát triển rộng lớn, giúp bạn dễ dàng tìm kiếm tài liệu và giải pháp cho các vấn đề phát sinh.
* Khả năng mở rộng: Mặc dù Flask là micro-framework, nhưng nó dễ dàng mở rộng và tích hợp với các công cụ và thư viện khác để phát triển các ứng dụng lớn.
* Nhược điểm:
* Thiếu các tính năng tích hợp sẵn: So với các framework lớn như Django, Flask không có sẵn nhiều tính năng như ORM, quản lý người dùng, hoặc hệ thống quản lý dữ liệu.
* Khả năng mở rộng giới hạn: Flask có thể trở thành khó khăn khi xây dựng các ứng dụng web lớn và phức tạp. Đối với các dự án lớn, bạn có thể cần sử dụng một framework phức tạp hơn như Django.

### 2.2.2 Ngôn ngữ lập trình

Python là một ngôn ngữ lập trình hướng đối tượng cấp cao, mạnh mẽ được giới thiệu bởi Guido van Rossum. Nó rất dễ học và nhanh chóng trở thành một trong những ngôn ngữ lập trình tốt nhất cho người mới bắt đầu. Python sử dụng lưu trữ được nhập động và dữ liệu được nhập động. Python là ngôn ngữ lập trình cấp cao cung cấp cách tiếp cận đơn giản và hiệu quả cho lập trình hướng đối tượng. Nó sử dụng cú pháp đơn giản của Python, linh hoạt, rõ ràng và dễ hiểu, đồng thời có thể được sử dụng để viết kịch bản và phát triển ứng dụng trong nhiều ngành và lĩnh vực.

Python là một ngôn ngữ lập trình nổi bật với các tính năng:

* Ngôn ngữ đơn giản, dễ hiểu: Python với cú pháp đơn giản và dễ hiểu. Dễ học và dễ lập trình hơn nhiều ngôn ngữ lập trình khác như C++, Java, C#. Ngôn ngữ Python làm cho việc lập trình trở nên thú vị và trao cho các lập trình viên các giải pháp tốt hơn trong mọi lĩnh vực.
* Mã nguồn mở và miễn phí: Ngôn ngữ này được sử dụng miễn phí, ngay cả với mục đích thương mại. Nó là mã nguồn mở nên bạn có thể dễ dàng thay đổi nó. Ngôn ngữ lập trình Python có một cộng đồng lớn và không ngừng cải tiến qua mỗi bản cập nhật.
* Có thể lập trình trên nhiều hệ điều hành: Mã chương trình Python có thể được chuyển từ nền tảng này sang nền tảng khác mà không làm thay đổi cấu trúc của chương trình.
* Khả năng nhúng và khả năng mở rộng của chương trình: Python dễ dàng tích hợp C, C++ và các ngôn ngữ lập trình khác vào mã Python. Điều này cho phép các ứng dụng có khả năng vượt trội mà ít ngôn ngữ lập trình đơn lẻ khác có thể cung cấp.
* Phiên dịch ngôn ngữ ở mức độ cao: Không giống như C/C++, việc quản lý bộ nhớ trong Python không khó. Điều này là do khi mã được thực thi, mã lập trình sẽ tự động được dịch sang ngôn ngữ máy để hoàn thành công việc.
* Một thư viện tiêu chuẩn lớn và phổ biến: Python có một số thư viện tiêu chuẩn giúp lập trình dễ dàng hơn nhiều vì bạn không phải tự viết tất cả mã. Tất cả những gì quan trọng là hiểu và sử dụng các thư viện có sẵn trong bối cảnh Python.
* Hướng đối tượng: Một ngôn ngữ lập trình hướng đối tượng (OOP) giúp giải quyết nhiều vấn đề phức tạp một cách trực quan. Đối với các vấn đề phức tạp, OOP giúp tạo các đối tượng chi tiết.

Ngày nay, Python vẫn là ngôn ngữ lập trình phổ biến cho nhiều lĩnh vực của cuộc sống, bao gồm: Lập trình ứng dụng web, khoa học máy tính, dự đoán tự nhiên, chẩn đoán y tế, phát triển game. Cùng với các tính năng tuyệt vời, Python có một số hạn chế lớn như:

* Chậm tính toán và thực thi.
* Tiêu tốn nhiều dung lượng lưu trữ.
* Hạn chế trong phát triển ứng dụng di động.
* Hạn chế trong thiết kế ứng dụng.

### 2.2.3 Hệ thống các thuật toán

* Flask Framework

Flask là một micro framework cho phép tạo ứng dụng web nhanh chóng và đơn giản. Trong trường hợp này, Flask được sử dụng để xây dựng một ứng dụng web cho phép người dùng tải ảnh lên, cải thiện chất lượng ảnh, và sau đó tải ảnh đã được cải thiện về.

Route & Request: Flask sử dụng các route (@app.route()) để xử lý các yêu cầu HTTP (GET, POST). Ứng dụng cho phép người dùng tải ảnh lên thông qua phương thức POST và hiển thị các kết quả thông qua GET.

Render Template: Flask sử dụng render\_template để gửi dữ liệu từ backend (như thông tin ảnh gốc, ảnh đã cải thiện, thời gian cải thiện, v.v.) tới frontend (HTML).

* Xử lý ảnh với Pillow (PIL)

Pillow (Python Imaging Library) được sử dụng để mở, xử lý và lưu ảnh. Cụ thể:

Image.open(): Mở ảnh từ file đã tải lên.

img.convert('L'): Chuyển ảnh sang chế độ màu đen trắng (grayscale) để giảm độ phức tạp của mô hình.

resize\_image(): Giảm kích thước ảnh nếu kích thước của nó vượt quá giới hạn tối đa (max\_size=1024) để tiết kiệm tài nguyên tính toán và đảm bảo ảnh không quá lớn.

img.save(): Lưu ảnh đã cải thiện vào thư mục được chỉ định.

* Mô hình học sâu với PyTorch

**load\_model()**: Đây là hàm để tải mô hình đã được huấn luyện từ file srcnn\_x3.pth.

**model()**:Mô hình được sử dụng để cải thiện ảnh đầu vào (ảnh có độ phân giải thấp). Trong mã này, ảnh được chuyển sang tensor và đưa vào mô hình CNN để cải thiện.

**torch.no\_grad()**: Được sử dụng để tắt việc tính toán gradients (không cần thiết khi chỉ sử dụng mô hình để dự đoán, giúp tiết kiệm bộ nhớ và tăng tốc tính toán).

* Xử lý Tensor với PyTorch

**Tensor**: PyTorch sử dụng đối tượng tensor (ma trận n chiều) để biểu diễn dữ liệu. Trong trường hợp này, ảnh được chuyển thành tensor bằng ToTensor(), sau đó được đưa vào mô hình.

**ToTensor()**: Hàm này chuyển đổi ảnh Pillow thành một tensor với các giá trị pixel trong khoảng từ 0 đến 1.

unsqueeze(0): Hàm này thêm một chiều vào tensor để tạo thành một batch (do mô hình dựa trên batch input).

**PSNR (Peak Signal-to-Noise Ratio)**: Là một thước đo chất lượng ảnh, được tính toán để so sánh độ tương phản giữa ảnh gốc và ảnh đã được cải thiện.

* Tính toán PSNR (Peak Signal-to-Noise Ratio)

**PSNR** là một thước đo độ nhiễu giữa ảnh gốc và ảnh được cải thiện. Trong mã, PSNR được tính bằng cách đo lường sai lệch giữa ảnh gốc và ảnh đã cải thiện.

Công thức: PSNR được tính từ MSE (Mean Squared Error), và công thức này giúp đánh giá chất lượng của ảnh sau khi được cải thiện.

**torch.sqrt(mse):** Tính căn bậc hai của MSE, sau đó tính giá trị PSNR bằng công thức logarit.

* Quản lý tệp và kích thước ảnh

Quản lý tệp tải lên: Mã kiểm tra các tệp tải lên có đúng định dạng (png, jpg, jpeg) hay không và đảm bảo kích thước ảnh không vượt quá 20MB.

**os.makedirs():** Dùng để tạo các thư mục lưu trữ ảnh nếu chúng chưa tồn tại.

**send\_file():** Hàm này có thể được sử dụng để gửi tệp ảnh đã cải thiện tới người dùng, tuy nhiên trong mã của bạn, tính năng này chưa được triển khai trong các route trả về template HTML.

* Flask Template Rendering

**render\_template():** Được sử dụng để trả về một template HTML và truyền dữ liệu từ backend vào template. Các thông tin như đường dẫn ảnh gốc, ảnh đã cải thiện, thời gian cải thiện, tỷ lệ phân giải và PSNR được đưa vào trang web.

Thông tin hiển thị trên giao diện người dùng: Tất cả các thông tin liên quan đến ảnh gốc và ảnh đã cải thiện (kích thước, thời gian xử lý, PSNR) được truyền vào template HTML để người dùng có thể thấy kết quả trực quan.

* Đo lường thời gian cải thiện

**time.time():** Được sử dụng để tính thời gian bắt đầu và kết thúc quá trình cải thiện ảnh. Sự khác biệt giữa hai thời điểm này cho phép tính thời gian cần thiết để cải thiện một ảnh.

* Quản lý Uploads và Tạo ảnh Enhanced

Thư mục lưu trữ ảnh: Các ảnh tải lên và ảnh đã cải thiện được lưu trong các thư mục static/uploads/ và static/enhanced/.

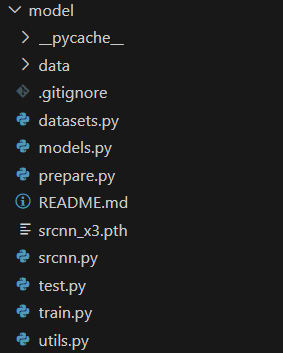
Kích thước ảnh: Mỗi ảnh tải lên sẽ được thay đổi kích thước sao cho chiều rộng hoặc chiều dài không vượt quá giới hạn tối đa (max\_size=1024), điều này giúp giảm bớt việc tính toán và tiết kiệm bộ nhớ khi xử lý ảnh.

# Chương 3: Kết quả thực nghiệm

## 3.1 Dữ liệu

### 3.1.1 Dữ liệu huấn luyện

Nguồn dữ liệu: thư mục lưu trữ ảnh dùng để huấn luyện và mô hình huấn luyện



* Argparse: Cung cấp cách dễ dàng để bạn xác định các tham số đầu vào cần thiết từ dòng lệnh, ví dụ như --input hoặc --output, và có thể tự động hiển thị hướng dẫn sử dụng khi cần
* Globb: Dễ dàng liệt kê các tệp có định dạng cụ thể hoặc khớp với một mẫu nào đó trong một thư mục
* H5py: Thư viện này cho phép bạn đọc và ghi dữ liệu vào các tệp HDF5, cung cấp các cấu trúc dữ liệu để lưu trữ các mảng n-dimensional, cùng với khả năng lưu trữ metadata, nhóm dữ liệu, và các mô tả khác
* Numpy: Numpy giúp bạn làm việc với dữ liệu dạng ma trận, ma trận chuyển vị, phép nhân ma trận, giải hệ phương trình tuyến tính, v.v.
* Pil: Pillow cung cấp các hàm để mở, cắt, thay đổi kích thước, xoay, chỉnh sửa màu sắc của ảnh, v.v. Thư viện này hỗ trợ rất nhiều định dạng ảnh phổ biến như PNG, JPG, BMP, GIF, v.v.

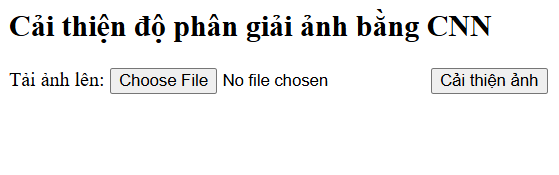
### 3.1.2 Luồng dữ liệu

* Người dùng chọn tệp ảnh và nhấn nút tải lên.
* Kiểm tra định dạng và kích thước tệp: Tệp ảnh phải có định dạng hợp lệ (png, jpg, jpeg) và không vượt quá kích thước tối đa (20MB). Nếu tệp không hợp lệ, hệ thống sẽ trả về thông báo lỗi.
* Chuyển đổi ảnh sang ảnh đen trắng: Ảnh được mở và chuyển đổi sang chế độ màu xám (đen trắng) với img.convert('L') để đơn giản hóa việc xử lý.
* Đổi kích thước ảnh:
* Hàm resize\_image() được gọi để thay đổi kích thước ảnh sao cho không vượt quá 1024px theo chiều dài hoặc chiều rộng lớn nhất.
* Sau khi thay đổi kích thước, ảnh sẽ được lưu lại.
* Chuyển đổi ảnh thành tensor và xử lý bằng mô hình
* Ảnh sau khi được xử lý được chuyển thành một tensor (đối tượng kiểu mảng đa chiều) sử dụng ToTensor().
* Tensor này sẽ được đưa vào mô hình SRCNN để cải thiện chất lượng ảnh.
* Tính toán PSNR (Peak Signal-to-Noise Ratio): PSNR là chỉ số dùng để đo lường mức độ nhiễu giữa ảnh gốc và ảnh đã cải thiện. Giá trị PSNR càng cao thì ảnh càng ít nhiễu.
* Sau khi quá trình xử lý xong, ứng dụng Flask sẽ trả về kết quả cho người dùng thông qua trang web. Trang web hiển thị các thông tin sau:
* Ảnh gốc và Ảnh đã cải thiện: Liên kết để người dùng tải xuống hoặc xem ảnh.
* Kích thước ảnh gốc và Kích thước ảnh đã cải thiện: Thông tin về kích thước của hai ảnh.

## 3.2 Kết quả thực nghiệm

### 3.2.1 Màn hình thao tác người dùng

Người dùng tải hình ảnh lên và nhấp vào “cải thiện ảnh”để chạy chương trình



### 3.2.2 Kết quả sau khi chạy chương trình

Sau khi chạy xong chương trình sẽ trả về kết quả là ảnh gốc và ảnh sau khi cải thiện kèm theo kích thước của ảnh



# Kết luận

## Mục tiêu và kết quả đạt được

* Mục tiêu:
* Xây dựng một ứng dụng web có khả năng cải thiện chất lượng ảnh thông qua việc sử dụng mô hình CNN (Convolutional Neural Network).
* Đo lường chất lượng hình ảnh qua các chỉ số như PSNR và thời gian xử lý.
* Kết quả đạt được:
* Ứng dụng web đã được triển khai thành công, hỗ trợ người dùng tải ảnh lên và nhận lại ảnh đã được cải thiện.
* Ảnh sau cải thiện có độ phân giải cao hơn và được đánh giá chất lượng bằng chỉ số PSNR.
* Quá trình xử lý diễn ra nhanh chóng, với giao diện thân thiện và dễ sử dụng.

## Ưu điểm

* Mô hình CNN có khả năng cải thiện chi tiết ảnh rõ ràng.
* Thời gian xử lý được tối ưu hóa nhờ sử dụng GPU nếu có.
* Giao diện thân thiện, dễ thao tác, phù hợp với người dùng phổ thông.
* Chỉ số PSNR giúp người dùng thấy được mức độ cải thiện của ảnh.

## Hạn chế

* CNN là mô hình cơ bản, có thể không đạt hiệu quả cao bằng các mô hình hiện đại như EDSR hoặc ESRGAN.
* Ứng dụng giới hạn ảnh đầu vào tối đa 20MB và kích thước ảnh được giảm về tối đa 1024 pixel để xử lý.
* Không xử lý ảnh màu: Ứng dụng hiện tại chỉ hỗ trợ xử lý ảnh đen trắng (grayscale).

## Hướng phát triển

* Cải tiến mô hình: Nâng cấp lên các mô hình hiện đại hơn như ESRGAN hoặc SwinIR để cải thiện chất lượng ảnh.
* Hỗ trợ ảnh màu: Mở rộng để xử lý ảnh RGB, tăng tính ứng dụng thực tế.
* Triển khai thực tế: Đưa ứng dụng lên các nền tảng đám mây (AWS, Google Cloud) để phục vụ nhiều người dùng hơn.
* Tích hợp thêm tính năng: Hỗ trợ tải ảnh hàng loạt.

# Tài liệu tham khảo

[1] N. T. Tuan, Deep learning cơ bản, 2020.

[2] N. T. Tuan, Deep learning cơ bản, 2020.

[3] P. J. Braspenning, F. Thuijsman and A. J. M. M. Weijters, Artificial Neural

[4] Cornell University Library PSEC Documentation Committee, Feb. 2010.

[Online]. Available: *http://www.library.cornell.edu/resrch/citmanage/apa.*