**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 9](#_Toc184658119)

[Chương 1: Cơ sở lý thuyết 10](#_Toc184658120)

[1.1 Tổng quan về thị giác máy tính 10](#_Toc184658121)

[1.2 Ứng dụng của thị giác máy tính 11](#_Toc184658122)

[1.2.1 Nhận diện và phân loại đối tượng 11](#_Toc184658123)

[1.2.2 Thị giác trong y tế 11](#_Toc184658124)

[1.2.3 Giao thông và xe tự hành 12](#_Toc184658125)

[1.2.4 Giám sát và an ninh 12](#_Toc184658126)

[1.2.5 Nông nghiệp thông minh 12](#_Toc184658127)

[1.2.6 Thương mại và bán lẻ 13](#_Toc184658128)

[1.2.8 Giải trí và truyền thông 13](#_Toc184658129)

[1.3 Một số thuật toán học sâu 14](#_Toc184658130)

[1.3.1 Mạng nơ-ron hồi tiếp (Recurrent Neural Networks - RNN) 14](#_Toc184658131)

[1.3.2 Mạng nơ-ron tích chập (Convolutional Neural Networks - CNN) 14](#_Toc184658132)

[1.3.3 SRCNN (Super-Resolution Convolutional Neural Network) 15](#_Toc184658133)

[Chương 2: Xây dựng hệ thống 16](#_Toc184658134)

[2.1 Mô tả bài toán 16](#_Toc184658135)

[2.1.1 Mục tiêu bài toán 16](#_Toc184658136)

[2.1.2 Cách hoạt động của SRCNN 17](#_Toc184658137)

[2.1.3 Ứng dụng trong thực tế 17](#_Toc184658138)

[2.2 Xây dựng hệ thống 19](#_Toc184658139)

[2.2.1 Mô hình bài toán 19](#_Toc184658140)

[2.2.2 Chi tiết hoạt động 20](#_Toc184658141)

[2.3 Ngôn ngữ lập trình và thư viện 22](#_Toc184658142)

[2.3.1 Ngôn ngữ lập trình 22](#_Toc184658143)

[2.3.2 Thư viện 23](#_Toc184658144)

[Chương 3: Kết quả thực nghiệm 28](#_Toc184658145)

[3.2 Kết quả thực nghiệm 29](#_Toc184658146)

[3.2.1 Màn hình thao tác người dùng 29](#_Toc184658147)

[3.2.2 Kết quả sau khi chạy chương trình 29](#_Toc184658148)

[Kết luận 30](#_Toc184658149)

[1. Mục tiêu và kết quả đạt được 30](#_Toc184658150)

[2. Ưu điểm 30](#_Toc184658151)

[3. Hạn chế 30](#_Toc184658152)

[4. Hướng phát triển 30](#_Toc184658153)

[Tài liệu tham khảo 31](#_Toc184658154)

Danh mục hình ảnh

|  |  |  |
| --- | --- | --- |
| STT | Hình | Trang |
| 1 | Hình 1: Mô hình bài toán | 19 |
| 2 | Hình 2: Ngôn ngữ Python | 23 |
| 3 | Hình 3: Thư viện PyTorch | 24 |
| 4 | Hình 4: Thư viện PIL | 25 |
| 5 | Hình 5: Thư viện NumPy | 25 |
| 6 | Hình 6: Thư viện Flask | 26 |
| 7 | Hình 7: Màn hình thao tác người dùng | 29 |
| 8 | Hình 8: Màn hình kết quả | 29 |

# 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 thị 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 Ứng dụng của thị giác máy tính

### 1.2.1 Nhận diện và phân loại đối tượng

* Nhận diện khuôn mặt:
* Ứng dụng:
* Hệ thống kiểm soát truy cập (mở khóa điện thoại, cửa thông minh).
* Giám sát tội phạm trong không gian công cộng.
* Tùy chỉnh quảng cáo dựa trên đặc điểm người dùng.
* Công nghệ:
* Mạng nơ-ron tích chập (Convolutional Neural Network - CNN).
* Dữ liệu đầu vào: Hình ảnh hoặc video.
* Các mô hình phổ biến: FaceNet, DeepFace.
* Nhận diện vật thể:
* Ứng dụng:
* Xe tự hành (phát hiện biển báo giao thông, người đi bộ).
* Robot công nghiệp (nhận diện và gắp vật).
* Hệ thống phân tích hình ảnh vệ tinh (phát hiện rừng, sông, thành phố).
* Công nghệ:
* YOLO (You Only Look Once), Faster R-CNN, Mask R-CNN.
* Nhận diện biển số xe:
* Ứng dụng:
* Quản lý giao thông, phát hiện vi phạm, bãi đỗ xe thông minh.
* Công nghệ:
* OCR (Optical Character Recognition) kết hợp xử lý hình ảnh.

### 1.2.2 Thị giác trong y tế

* Phân tích hình ảnh y khoa:
* Ứng dụng:
* Phát hiện bệnh tim mạch qua ảnh siêu âm.
* Chẩn đoán sớm ung thư (ví dụ: qua X-quang phổi hoặc ảnh tế bào ung thư da).
* Theo dõi sự phát triển của bệnh qua thời gian.
* Công nghệ:
* Các mô hình như U-Net, VGG16 kết hợp với dữ liệu y khoa chuyên biệt.
* Tăng cường dữ liệu (Data Augmentation) để cải thiện mô hình.
* Theo dõi sức khỏe và hỗ trợ phẫu thuật:
* Ứng dụng:
* Robot phẫu thuật hỗ trợ bằng hình ảnh thực tế ảo.
* Theo dõi nhịp tim, hô hấp qua camera mà không cần tiếp xúc.
* Công nghệ:
* Học sâu (Deep Learning) tích hợp với AR và VR.

### 1.2.3 Giao thông và xe tự hành

* Xe tự lái:
* Ứng dụng:
* Nhận diện vật cản, làn đường, tín hiệu giao thông.
* Tự động phanh khi phát hiện người qua đường.
* Công nghệ:
* LiDAR kết hợp với thị giác máy tính.
* Deep Learning (ResNet, EfficientNet).
* Phân tích lưu lượng giao thông:
* Ứng dụng:
* Dự đoán ùn tắc, tối ưu hóa luồng giao thông.
* Đếm số lượng xe, phân loại xe tải, xe hơi, xe máy.
* Công nghệ:
* Xử lý video thời gian thực kết hợp phân đoạn ảnh (Image Segmentation).

### 1.2.4 Giám sát và an ninh

* Camera thông minh:
* Ứng dụng:
  + Nhận diện hành vi khả nghi (bỏ quên hành lý, leo rào, gây rối).
  + Giám sát số lượng người trong đám đông.
* Công nghệ:
  + AI kết hợp nhận diện tư thế (Pose Estimation) và học máy.
  + Kiểm soát ra vào:
* Ứng dụng:
  + Nhận diện danh tính qua khuôn mặt hoặc cử chỉ tay.
* Công nghệ:
  + Face Detection và Tracking.

### 1.2.5 Nông nghiệp thông minh

* Giám sát cây trồng:
* Ứng dụng:
  + Phát hiện sớm bệnh trên lá cây.
  + Theo dõi sự phát triển của mùa vụ qua ảnh drone.
* Công nghệ:
  + Chụp ảnh đa phổ (Multispectral Imaging) kết hợp với CNN.
* Tự động phân loại sản phẩm:
* Ứng dụng:
  + Phân loại nông sản dựa trên màu sắc, hình dạng, kích thước.
* Công nghệ:
  + Image Classification với các mô hình như MobileNet, Inception.

### 1.2.6 Thương mại và bán lẻ

* Tìm kiếm sản phẩm qua hình ảnh:
* Ứng dụng:
  + Tìm sản phẩm tương tự từ hình ảnh chụp.
* Công nghệ:
  + Học sâu và Học chuyển giao (Transfer Learning).
* Phân tích hành vi:
* Ứng dụng:
  + Ghi nhận thời gian khách hàng đứng tại quầy hàng.
  + Dự đoán sản phẩm yêu thích dựa trên khuôn mặt và hành vi.
* Công nghệ:
  + Deep Learning và phân tích dữ liệu lớn.

1.2.7 Công nghiệp và sản xuất

* Kiểm tra chất lượng sản phẩm:
* Ứng dụng:
  + Phát hiện lỗi trên sản phẩm như nứt, trầy xước.
* Công nghệ:
  + Xử lý ảnh thời gian thực trên dây chuyền sản xuất.
  + Mô hình Segmentation (phân đoạn ảnh).
* Robot công nghiệp:
* Ứng dụng:
  + Robot tự định vị và thao tác trong môi trường phức tạp.
* Công nghệ:
  + Thị giác máy tính tích hợp điều khiển tự động.

### 1.2.8 Giải trí và truyền thông

* Hiệu ứng video thông minh:
* Ứng dụng:
  + Thay đổi hậu cảnh video (background removal).
  + Hiệu ứng 3D hoặc chuyển động thực tế ảo.
* Công nghệ:
  + GAN (Generative Adversarial Networks) và xử lý ảnh thời gian thực.
* Tăng cường thực tế:
* Ứng dụng:
  + AR trong game hoặc huấn luyện kỹ năng.
* Công nghệ:
  + Học máy tích hợp cảm biến và camera.

## 1.3 Một số thuật toán học sâu

### 1.3.1 Mạng nơ-ron hồi tiếp (Recurrent Neural Networks - RNN)

* Đặc điểm:
* Cấu trúc: Mỗi nút (neuron) trong mạng RNN kết nối với chính nó, tạo ra một vòng lặp. Điều này giúp mạng lưu giữ trạng thái của bước trước đó để xử lý bước tiếp theo.
* Dữ liệu đầu vào: Các chuỗi dữ liệu như văn bản, chuỗi số, âm thanh, hoặc tín hiệu thời gian.
* Lan truyền ngược: Sử dụng Backpropagation Through Time (BPTT) để cập nhật trọng số qua nhiều bước thời gian.
* Ưu điểm:
* Lưu giữ ngữ cảnh: Hiểu mối quan hệ giữa các phần tử trong chuỗi, như ngữ cảnh của từ trong câu.
* Khả năng tái sử dụng trọng số: Giảm yêu cầu bộ nhớ, vì cùng một trọng số được sử dụng cho các bước thời gian khác nhau.
* Thích hợp cho dữ liệu tuần tự: Áp dụng tốt cho xử lý ngôn ngữ tự nhiên (NLP), nhận dạng giọng nói, dự đoán chuỗi thời gian.
* Nhược điểm:
* Vanishing Gradient: Khi chuỗi quá dài, các giá trị gradient giảm dần về 0, khiến mạng không học được thông tin dài hạn.
* Dẫn đến khó khăn trong việc dự đoán phụ thuộc xa (long-term dependency).
* Không song song hóa: Vì dữ liệu phải xử lý theo thứ tự, mạng không tận dụng được khả năng song song hóa của GPU.
* Đào tạo chậm: Thời gian tính toán lớn, đặc biệt khi độ dài chuỗi tăng lên.
* Ứng dụng:
* Xử lý ngôn ngữ tự nhiên:
* Dịch ngôn ngữ (Google Translate).
* Tự động hoàn thành câu (GPT).
* Dự đoán chuỗi thời gian:
* Giá cổ phiếu, thời tiết, lưu lượng mạng.
* Âm nhạc và giọng nói:
* Nhận dạng giọng nói, phân tích cảm xúc từ âm thanh.

### 1.3.2 Mạng nơ-ron tích chập (Convolutional Neural Networks - CNN)

* Đặc điểm:
* Gồm các lớp tích chập (convolutional), lớp phi tuyến (ReLU), và lớp pooling (giảm chiều dữ liệu).
* CNN hoạt động dựa trên các bộ lọc (filters/kernels) để học đặc trưng từ ảnh (như cạnh, góc).
* Mỗi bộ lọc chỉ tập trung vào một vùng nhỏ của ảnh (receptive field), giúp giảm số lượng tham số.
* Ưu điểm:
* Tự động trích xuất đặc trưng: Không cần trích xuất thủ công như các phương pháp truyền thống (SIFT, HOG).
* Tổng quát hóa cao: Hoạt động tốt với các biến đổi về dịch chuyển, xoay, hoặc tỷ lệ của ảnh.
* Song song hóa tốt: Mỗi phép tích chập có thể chạy đồng thời, tận dụng tối đa khả năng GPU.
* Nhược điểm:
* Hạn chế về ngữ cảnh: Không hiểu mối quan hệ tuần tự hoặc dài hạn trong dữ liệu, chỉ tập trung vào đặc trưng cục bộ.
* Phụ thuộc vào dữ liệu lớn: Hiệu quả học thấp nếu dữ liệu huấn luyện không đủ đa dạng.
* Kích thước lớn của mô hình: Các mạng CNN sâu như ResNet, DenseNet cần nhiều tài nguyên để huấn luyện.
* Ứng dụng:
* Xử lý ảnh: Phân loại ảnh (ImageNet), phát hiện đối tượng (YOLO, Faster R-CNN).
* Thị giác máy tính: Nhận diện khuôn mặt, phân đoạn ảnh.
* Video và hoạt họa: Phân tích video, chuyển động.

### 1.3.3 SRCNN (Super-Resolution Convolutional Neural Network)

* Đặc điểm:
* Cấu trúc: SRCNN là một biến thể của CNN, thiết kế đặc biệt cho bài toán tăng cường độ phân giải ảnh (Super-Resolution).
* Hoạt động: 3 bước chính:
* Trích xuất đặc trưng từ ảnh nội suy: Dùng các bộ lọc nhỏ để học thông tin từ ảnh đầu vào.
* Ánh xạ phi tuyến: Tăng cường khả năng học các chi tiết phức tạp.
* Tái tạo ảnh: Kết hợp đặc trưng đã học để tái tạo ảnh đầu ra độ phân giải cao.
* Ưu điểm:
* Kiến trúc đơn giản: Dễ dàng triển khai và huấn luyện.
* Chất lượng ảnh cao: Tăng cường độ phân giải mà vẫn giữ được chi tiết gốc tốt.
* Hiệu quả trên ảnh nhỏ: Đặc biệt phù hợp cho các bài toán phục hồi ảnh cũ, ảnh mờ.
* Nhược điểm:
* Phụ thuộc nội suy: Hiệu quả thấp nếu ảnh đầu vào không được nội suy trước.
* Hiệu suất hạn chế: Các mô hình hiện đại như EDSR, ESRGAN vượt trội về chất lượng.
* Không linh hoạt: Chỉ áp dụng cho bài toán tăng cường độ phân giải ảnh.
* Ứng dụng:
* Nhiếp ảnh và phục hồi ảnh cũ: Làm sắc nét ảnh cũ, ảnh bị mờ.
* Giám sát và an ninh: Tăng độ chi tiết của ảnh hoặc video từ camera.
* Chỉnh sửa đồ họa: Tăng cường chất lượng ảnh trong game hoặc phim hoạt hình.

# Chương 2: Xây dựng hệ thống

## 2.1 Mô tả bài toán

### 2.1.1 Mục tiêu bài toán

Bài toán SRCNN là một mạng nơ-ron tích chập (CNN) nhằm mục đích tái tạo một ảnh có độ phân giải cao (HR) từ một ảnh có độ phân giải thấp (LR). Mục tiêu là phục hồi các chi tiết bị mất trong quá trình giảm độ phân giải của ảnh gốc, đồng thời giữ nguyên được các đặc trưng của ảnh như màu sắc, cấu trúc, và các chi tiết nhỏ.

* Tính Ứng Dụng Cao trong Thực Tế
* Ứng dụng trong Y tế: Cải thiện độ phân giải ảnh trong y tế, như ảnh chụp X-quang, MRI, hoặc siêu âm, giúp các bác sĩ và chuyên gia có thể nhận diện rõ hơn các chi tiết, phát hiện bệnh sớm và đưa ra phương pháp điều trị chính xác hơn. Việc tái tạo ảnh có độ phân giải cao giúp nâng cao chất lượng chẩn đoán mà không cần phải thu thập lại các hình ảnh tốn kém và mất thời gian.
* Giám sát an ninh: Hệ thống giám sát an ninh, đặc biệt là trong các môi trường có ánh sáng yếu hoặc chất lượng camera kém, có thể cải thiện khả năng nhận diện và phân tích các đối tượng từ các video hoặc hình ảnh có độ phân giải thấp. Việc sử dụng SRCNN giúp tăng cường các chi tiết trong ảnh để dễ dàng nhận diện các đối tượng, cải thiện độ chính xác của các hệ thống nhận diện khuôn mặt, biển số xe, v.v.
* Vệ tinh và nghiên cứu không gian: Cải thiện ảnh vệ tinh với độ phân giải thấp, giúp phân tích chi tiết các yếu tố môi trường, tài nguyên thiên nhiên hoặc trong các nghiên cứu quân sự.
* Khả Năng Nâng Cao Chất Lượng Hình Ảnh
* SRCNN cung cấp một phương pháp hiệu quả để cải thiện chất lượng của các ảnh có độ phân giải thấp mà không cần phải thu thập lại dữ liệu. Điều này đặc biệt quan trọng trong các trường hợp mà việc thu thập ảnh chất lượng cao là khó khăn hoặc tốn kém, ví dụ như trong giám sát vệ tinh hoặc các tình huống khẩn cấp.
* Việc nâng cao độ phân giải của ảnh mà không làm mất đi chi tiết quan trọng có thể hỗ trợ trong việc tối ưu hóa các quy trình phân tích và ra quyết định, đặc biệt trong các lĩnh vực yêu cầu độ chính xác cao.
* Sự Tiến Bộ trong Công Nghệ Học Sâu
* SRCNN là một bước đột phá trong học sâu: SRCNN là một trong những mô hình đầu tiên áp dụng học sâu (deep learning) vào bài toán cải thiện độ phân giải ảnh. Việc sử dụng các mạng nơ-ron tích chập (CNN) giúp mô hình học được các đặc trưng phức tạp từ dữ liệu mà không cần phải thiết kế thủ công các thuật toán. Điều này đánh dấu sự chuyển mình mạnh mẽ trong lĩnh vực xử lý ảnh.
* Kỹ thuật học sâu (Deep Learning) ngày càng trở nên phổ biến: Các mô hình học sâu đã trở thành công cụ không thể thiếu trong các nghiên cứu và ứng dụng xử lý ảnh hiện đại. Việc chọn đề tài này không chỉ giúp tìm hiểu về SRCNN mà còn giúp nắm bắt được xu hướng và phát triển trong lĩnh vực học máy và học sâu.
* Tính Mới Mẻ và Thách Thức Kỹ Thuật
* Đảm bảo độ chính xác và hiệu quả: Mặc dù SRCNN đã chứng minh được khả năng cải thiện độ phân giải ảnh, nhưng việc tối ưu hóa mô hình và cải thiện hiệu suất vẫn là một thách thức, đặc biệt khi áp dụng vào các trường hợp thực tế. Việc chọn đề tài này cho phép nghiên cứu và cải tiến các yếu tố như kiến trúc mạng, hàm mất mát, và chiến lược huấn luyện để đạt được kết quả tốt hơn.
* Khả năng nâng cấp mô hình: SRCNN có thể không phải là mô hình tối ưu nhất trong mọi trường hợp (ví dụ, với ảnh có độ phân giải cực kỳ thấp), và việc cải thiện hoặc kết hợp SRCNN với các phương pháp khác (như VDSR, EDSR) để giải quyết các bài toán phức tạp hơn là một hướng nghiên cứu đầy tiềm năng.
* Dễ Dàng Triển Khai và Tinh Chỉnh
* SRCNN là một mô hình đơn giản và dễ triển khai: So với các mô hình học sâu phức tạp khác, SRCNN có kiến trúc mạng khá đơn giản với ba lớp chính, điều này giúp việc triển khai và tối ưu hóa mô hình trở nên dễ dàng hơn. Việc sử dụng SRCNN trong các dự án thực tế giúp rút ngắn thời gian nghiên cứu và phát triển.
* Cộng đồng nghiên cứu mạnh mẽ: SRCNN là một chủ đề nghiên cứu phổ biến, với nhiều tài liệu và các mô hình đã được phát triển và công nhận trong cộng đồng học sâu. Điều này giúp các nhà nghiên cứu dễ dàng tiếp cận các nguồn tài liệu phong phú và phát triển mô hình một cách hiệu quả.
* Cải Thiện Các Mô Hình Học Máy Khác
* SRCNN là bước đệm cho các mô hình nâng cao: Mặc dù SRCNN là mô hình cơ bản, nhưng nó là nền tảng quan trọng cho các mô hình cải tiến sau này như VDSR (Very Deep Super-Resolution), EDSR (Enhanced Deep Super-Resolution Network), và các mô hình GAN (Generative Adversarial Networks). Việc nghiên cứu và hiểu rõ SRCNN sẽ giúp nâng cao khả năng phát triển các mô hình SR hiện đại và tinh vi hơn.

### 2.1.2 Cách hoạt động của SRCNN

* Tiền xử lý ảnh đầu vào (Low-Resolution - LR): Chia ảnh thành các khối con (patches)
* Kết quả đầu ra (HR): Ảnh có độ phân giải cao (HR)
* Sau khi các đặc trưng đã được học và xử lý qua các lớp tích chập, mô hình sẽ tạo ra ảnh HR (ảnh có độ phân giải cao hơn).
* Kết quả cuối cùng của mô hình sẽ là một ảnh có kích thước và độ phân giải cao hơn so với ảnh đầu vào LR. Ảnh HR này có thể được tái tạo lại từ các khối nhỏ (patches) mà mô hình đã xử lý trước đó.
* Một số mô hình sẽ sử dụng một chiến lược để ghép các patches đã xử lý lại với nhau để tạo ra ảnh hoàn chỉnh.

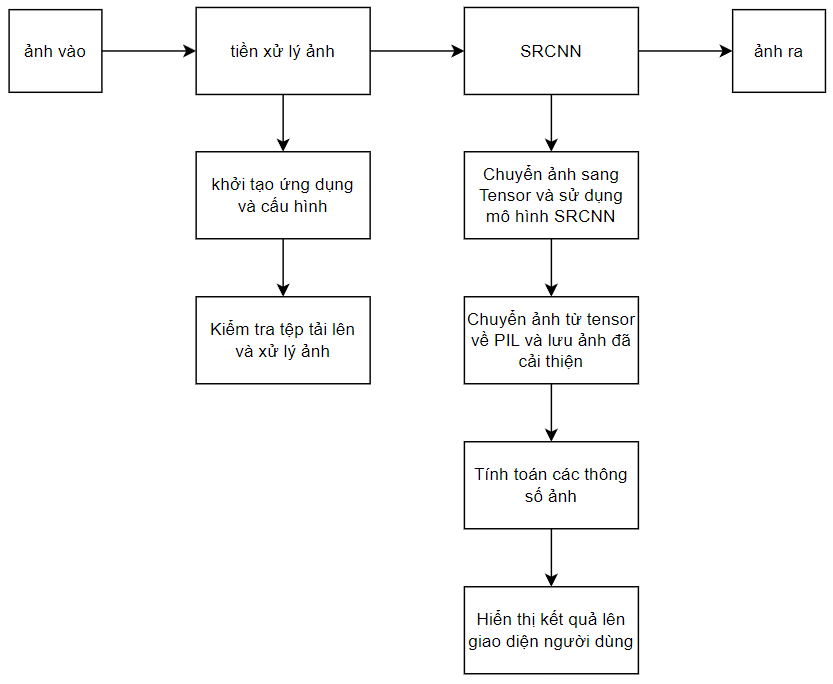
### 2.1.3 Ứng dụng trong thực tế

Bài toán cải thiện độ phân giải ảnh bằng SRCNN có rất nhiều ứng dụng quan trọng trong thực tế, đặc biệt trong các lĩnh vực yêu cầu xử lý ảnh chất lượng cao và có độ phân giải thấp. Dưới đây là một số ứng dụng cụ thể của bài toán này trong nhiều ngành công nghiệp và lĩnh vực khác nhau:

* Y tế
* Chẩn đoán hình ảnh y tế: Trong y tế, các ảnh như X-quang, MRI, hoặc CT scan thường có độ phân giải thấp do hạn chế về thiết bị hoặc băng thông truyền tải. Việc cải thiện độ phân giải của các ảnh này giúp các bác sĩ và chuyên gia y tế có thể nhận diện chi tiết hơn các cấu trúc tế bào, khối u hoặc các vấn đề sức khỏe khác mà ảnh gốc có thể không rõ ràng.
* Siêu âm: Các hình ảnh siêu âm thường bị nhiễu và có độ phân giải thấp, nhưng việc cải thiện độ phân giải giúp các bác sĩ có thể quan sát các chi tiết nhỏ hơn trong cơ thể, đặc biệt là khi xác định các vấn đề trong thai kỳ hoặc các bệnh lý.
* Giảm thiểu chi phí chụp ảnh y tế: Việc cải thiện ảnh từ độ phân giải thấp có thể tiết kiệm chi phí chụp hình, tránh việc phải tái chụp hình ảnh với máy móc đắt tiền và giảm thiểu thời gian chờ đợi.
* Giám sát an ninh
* Camera giám sát: Trong các hệ thống camera an ninh, đặc biệt ở các khu vực có ánh sáng yếu hoặc các camera có độ phân giải thấp, việc cải thiện độ phân giải ảnh giúp nhận diện và phân tích các đối tượng chính xác hơn. Điều này có thể hỗ trợ trong việc nhận diện khuôn mặt, biển số xe, hoặc các tình huống đặc biệt trong các hệ thống giám sát an ninh công cộng.
* Cải thiện chất lượng video giám sát: Với sự phát triển của các hệ thống giám sát, việc xử lý và cải thiện độ phân giải của video ghi lại giúp tăng cường khả năng phát hiện các sự kiện quan trọng, hỗ trợ việc điều tra tội phạm và đảm bảo an ninh.
* Vệ tinh và khảo sát không gian
* Hình ảnh vệ tinh: Trong các nghiên cứu về môi trường, nông nghiệp, hoặc quan sát trái đất, các ảnh vệ tinh có độ phân giải thấp có thể được cải thiện để giúp phân tích chi tiết hơn về các yếu tố như thảm thực vật, biến đổi khí hậu, ô nhiễm môi trường, hoặc các thay đổi trong hạ tầng.
* Giám sát thiên tai: Khi thiên tai xảy ra, các ảnh vệ tinh thu thập từ trên không có thể có độ phân giải thấp. Việc cải thiện độ phân giải của ảnh vệ tinh giúp đánh giá nhanh chóng mức độ thiệt hại, hỗ trợ công tác cứu hộ và phục hồi.
* Nghệ thuật số và đồ họa máy tính
* Nâng cấp hình ảnh trong thiết kế đồ họa: Trong ngành thiết kế đồ họa và sản xuất phim, cải thiện độ phân giải ảnh có thể giúp phục hồi và nâng cao chất lượng của các hình ảnh cũ hoặc bị mờ. Điều này đặc biệt hữu ích trong các ứng dụng tái tạo hình ảnh cũ, phục hồi phim ảnh hoặc phục chế các bức tranh nghệ thuật kỹ thuật số.
* Chỉnh sửa ảnh: Đối với các nhiếp ảnh gia, việc cải thiện độ phân giải ảnh giúp tăng khả năng chỉnh sửa và tái tạo ảnh, đặc biệt khi họ cần làm sắc nét các chi tiết mà không làm mất chất lượng.
* Chụp ảnh và video
* Chụp ảnh từ thiết bị di động: Trong các thiết bị di động, đặc biệt là những thiết bị có camera với độ phân giải thấp, việc cải thiện ảnh từ ảnh gốc có thể giúp tạo ra những bức ảnh có chất lượng cao hơn, thậm chí gần bằng với các camera chuyên dụng. Điều này giúp người dùng có thể chụp ảnh rõ nét hơn trong nhiều điều kiện ánh sáng và môi trường.
* Video độ phân giải cao: Việc cải thiện độ phân giải của video có thể giúp nâng cao chất lượng video khi xem hoặc chỉnh sửa, đặc biệt đối với những video có chất lượng thấp được quay từ các thiết bị không phải là máy quay chuyên nghiệp.
* Ứng dụng trong tự động hóa và công nghiệp
* Nhận diện đối tượng trong sản xuất: Các hệ thống nhận diện trong dây chuyền sản xuất, kiểm tra chất lượng sản phẩm, hay nhận diện lỗi trong các sản phẩm có thể sử dụng các ảnh với độ phân giải cao hơn để phát hiện chi tiết nhỏ mà mắt thường không thể nhìn thấy.
* Robot và xe tự lái: Các hệ thống robot và xe tự lái sử dụng hình ảnh từ cảm biến để điều hướng và nhận diện môi trường. Việc cải thiện độ phân giải ảnh giúp các hệ thống này nhận diện rõ hơn các vật thể xung quanh, tăng độ chính xác và an toàn trong hoạt động.
* Ứng dụng trong pháp lý và tội phạm học
* Giải mã và phân tích chứng cứ: Trong các tình huống điều tra tội phạm, ảnh hoặc video chất lượng thấp (ví dụ, từ camera giám sát hoặc từ các thiết bị an ninh) có thể được cải thiện để phát hiện các chi tiết quan trọng như khuôn mặt, biển số xe hoặc các đặc điểm nhận dạng khác của nghi phạm.
* Phân tích ảnh hiện trường: Việc cải thiện độ phân giải ảnh hiện trường vụ án giúp các chuyên gia phân tích chi tiết hơn và phục hồi các thông tin quan trọng từ các ảnh mờ hoặc bị nhiễu.
* Giải trí và truyền thông
* Nâng cao chất lượng phim ảnh: SRCNN có thể được sử dụng để cải thiện độ phân giải của các bộ phim cổ điển hoặc phim ảnh có chất lượng thấp trước khi chuyển thể lên các nền tảng phát trực tuyến hoặc Blu-ray, nhằm cung cấp cho người xem trải nghiệm hình ảnh tốt hơn.
* Trò chơi điện tử: Trong ngành công nghiệp game, việc cải thiện độ phân giải của các cảnh quay hoặc các đồ họa 3D từ các trò chơi có thể mang lại trải nghiệm người dùng chân thực hơn.
* Quản lý dữ liệu và nén ảnh
* Nén ảnh và truyền tải dữ liệu: Trong trường hợp cần truyền tải hoặc lưu trữ ảnh có độ phân giải thấp (do băng thông hoặc dung lượng lưu trữ hạn chế), việc sử dụng SRCNN để cải thiện độ phân giải sau khi giải nén sẽ giúp phục hồi chi tiết ảnh mà không làm giảm chất lượng khi so với ảnh gốc.

## 2.2 Xây dựng hệ thống

### 2.2.1 Mô hình bài toán



*Hình 1: Mô hình bài toán*

### 2.2.2 Chi tiết hoạt động

* Input: đưa ảnh vào hệ thống (ảnh có độ phân giải thấp)
* Tiền xử lý ảnh:
* Khởi tạo ứng dụng và cấu hình
* Khởi tạo Flask app: tạo ra một đối tượng *Flask. \_\_name\_\_* được sử dụng để Flask biết nơi tìm kiếm các tệp mẫu HTML và các tệp tĩnh khác.

|  |
| --- |
| app = Flask(\_\_name\_\_) |

* Cấu hình thư mục tải ảnh và lưu ảnh đã cải thiện
* *UPLOAD\_FOLDER*: Thư mục chứa các ảnh mà người dùng tải lên.
* *ENHANCED\_FOLDER*: Thư mục chứa các ảnh đã được cải thiện độ phân giải.
* *os.makedirs()* giúp đảm bảo rằng thư mục đã tồn tại, nếu chưa có thì sẽ được tạo ra.

|  |
| --- |
| UPLOAD\_FOLDER = 'static/uploads/'  ENHANCED\_FOLDER = 'static/enhanced/'  os.makedirs(UPLOAD\_FOLDER, exist\_ok=True)  os.makedirs(ENHANCED\_FOLDER, exist\_ok=True) |

* Giới hạn kích thước tệp tải lên: Giới hạn dung lượng tối đa của tệp ảnh tải lên là 20MB. Flask sẽ tự động kiểm tra và trả về lỗi nếu tệp tải lên vượt quá dung lượng này.

|  |
| --- |
| MAX\_CONTENT\_LENGTH = 20 \* 1024 \* 1024  app.config['MAX\_CONTENT\_LENGTH'] = MAX\_CONTENT\_LENGTH |

* Kiểm tra định dạng tệp ảnh hợp lệ: Ứng dụng chỉ cho phép tải lên các tệp ảnh có định dạng .png, .jpg, hoặc .jpeg.

|  |
| --- |
| ALLOWED\_EXTENSIONS = {'png', 'jpg', 'jpeg'} |

* Kiểm tra tệp tải lên và xử lý ảnh
* Kiểm tra xem tệp có hợp lệ không: Hàm allowed\_file kiểm tra xem tên tệp có đuôi hợp lệ không (kết thúc bằng .png, .jpg, .jpeg).

|  |
| --- |
| def allowed\_file(filename):  return '.' in filename and filename.rsplit('.', 1)[1].lower() in ALLOWED\_EXTENSIONS |

* Xử lý ảnh tải lên:
* Kiểm tra kích thước tệp ảnh: Kiểm tra xem kích thước của tệp ảnh có vượt quá 20MB không. Nếu có, sẽ trả về lỗi yêu cầu người dùng tải lên ảnh nhỏ hơn

|  |
| --- |
| if len(file.read()) > MAX\_CONTENT\_LENGTH:  return render\_template("index.html", error="Tệp quá lớn. Vui lòng tải lên ảnh nhỏ hơn 20MB.") |

* Lưu tệp ảnh vào thư mục uploads: Lưu tệp ảnh tải lên vào thư mục UPLOAD\_FOLDER với tên tệp gốc của nó

|  |
| --- |
| file.seek(0)  original\_img\_filename = file.filename  original\_img\_path = os.path.join(UPLOAD\_FOLDER, original\_img\_filename)  file.save(original\_img\_path) |

* Mở ảnh và chuyển sang chế độ grayscale (đen trắng): Mở ảnh từ tệp vừa lưu và chuyển nó sang chế độ grayscale (đen trắng). Điều này có thể giúp đơn giản hóa bài toán cho mô hình SRCNN, vốn có thể được huấn luyện trên ảnh grayscale.

|  |
| --- |
| img = Image.open(original\_img\_path).convert('L') |

* Thay đổi kích thước ảnh nếu cần thiết: sẽ thay đổi kích thước ảnh sao cho chiều dài hoặc chiều rộng không vượt quá 1024px. Điều này nhằm đảm bảo ảnh không quá lớn, giúp mô hình xử lý hiệu quả hơn.

|  |
| --- |
| img = resize\_image(img, max\_size=1024) |

* SRCNN
* Chuyển ảnh sang Tensor và sử dụng mô hình SRCNN
* Ảnh PIL sẽ được chuyển thành tensor bằng *ToTensor()*. Sau đó, hàm *unsqueeze(0)* thêm một chiều ở vị trí đầu tiên để tạo thành tensor có kích thước [1, C, H, W] (1 ảnh, 1 kênh màu, chiều cao, chiều rộng).
* Tensor sau đó được chuyển sang thiết bị GPU hoặc CPU tùy thuộc vào khả năng phần cứng.

|  |
| --- |
| img\_tensor = ToTensor()(img).unsqueeze(0).to(device) |

* Ảnh tensor được đưa qua mô hình SRCNN để cải thiện độ phân giải. torch.no\_grad() giúp đảm bảo rằng PyTorch không tính toán gradient trong quá trình này, giúp tiết kiệm bộ nhớ và tăng tốc tính toán.

|  |
| --- |
| with torch.no\_grad():  enhanced\_img = model(img\_tensor) |

* Chuyển ảnh từ tensor về PIL và lưu ảnh đã cải thiện
* Chuyển tensor thành ảnh PIL: Sau khi ảnh được cải thiện bởi mô hình SRCNN, kết quả trả về là tensor 4D. *squeeze(0)* loại bỏ chiều đầu tiên *(batch size)*, trả về tensor 3D (C, H, W), sau đó chuyển tensor này thành ảnh PIL với *ToPILImage().*

|  |
| --- |
| enhanced\_img = enhanced\_img.squeeze(0)  enhanced\_img = ToPILImage()(enhanced\_img) |

* Lưu ảnh đã cải thiện: Ảnh đã được cải thiện sẽ được lưu vào thư mục *ENHANCED\_FOLDER* với tên mới bắt đầu bằng "*enhanced\_*" và sau đó lưu với tên tệp gốc.

|  |
| --- |
| enhanced\_img\_filename = f"enhanced\_{original\_img\_filename}"  enhanced\_img\_path = os.path.join(ENHANCED\_FOLDER, enhanced\_img\_filename)  enhanced\_img.save(enhanced\_img\_path) |

* Tính toán các thông số ảnh
* Tính toán PSNR (Peak Signal-to-Noise Ratio)
* PSNR đo lường mức độ nhiễu giữa ảnh gốc và ảnh cải thiện. Giá trị PSNR càng cao, chất lượng ảnh cải thiện càng tốt.
* PSNR được tính từ MSE (Mean Squared Error) giữa ảnh gốc và ảnh đã cải thiện.

|  |
| --- |
| def psnr(img1, img2):  mse = torch.mean((img1 - img2) \*\* 2)  if mse == 0:  return 100  return 20 \* torch.log10(255.0 / torch.sqrt(mse)) |

* Tính toán thời gian cải thiện ảnh: Thời gian cải thiện ảnh sẽ được tính từ lúc bắt đầu cho đến khi ảnh đã được cải thiện xong.

|  |
| --- |
| start\_time = time.time()  # Cải thiện ảnh  end\_time = time.time()  enhancement\_time = round(end\_time - start\_time, 2) |

* Tính toán tỷ lệ độ phân giải: Tỷ lệ độ phân giải giữa ảnh đã cải thiện và ảnh gốc được tính bằng cách lấy tỷ lệ chiều rộng và chiều cao của ảnh đã cải thiện so với ảnh gốc.

|  |
| --- |
| resolution\_ratio = round((enhanced\_img\_size[0] / original\_img\_size[0]) \* (enhanced\_img\_size[1] / original\_img\_size[1]), 2) |

* Hiển thị kết quả lên giao diện người dung
* *original\_img*: Đường dẫn ảnh gốc.
* *enhanced\_img*: Đường dẫn ảnh đã cải thiện.
* *original\_img\_size* và *enhanced\_img\_size*: Kích thước của ảnh gốc và ảnh đã cải thiện.
* *enhancement\_time*: Thời gian thực hiện cải thiện ảnh.
* *resolution\_ratio*: Tỷ lệ độ phân giải.
* *original\_psnr\_value* và *enhanced\_psnr\_value*: Giá trị PSNR của ảnh gốc và ảnh đã cải thiện.

|  |
| --- |
| return render\_template("index.html",  original\_img=original\_img\_path,  enhanced\_img=enhanced\_img\_path,  original\_img\_size=original\_img\_size,  enhanced\_img\_size=enhanced\_img\_size,  enhancement\_time=enhancement\_time,  resolution\_ratio=resolution\_ratio,  original\_psnr\_value=original\_psnr\_value,  enhanced\_psnr\_value=round(enhanced\_psnr\_value.item(), 2),  show\_original\_info=True) |

* Output: dưa ra ảnh đã được xử lý với độ phân giải cao hơn

## 2.3 Ngôn ngữ lập trình và thư viện

### 2.3.1 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.



*Hình 2: Ngôn ngữ Python*

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.

### 2.3.2 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).



*Hình 3: Thư viện PyTorch*

* 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.



*Hình 4: Thư viện PIL*

* 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.



*Hình 5: Thư viện NumPy*

* 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.



*Hình 6: Thư viện Flask*

* 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.

# Chương 3: Kết quả thực nghiệm

3.1 Dữ liệu

3.1.1 Dữ lệu huấn luyện

Mô hình SRCNN được huấn luyện với một bộ dữ liệu chứa ảnh có độ phân giải cao (HR - High-Resolution) và các ảnh có độ phân giải thấp tương ứng (LR - Low-Resolution) tạo ra từ các ảnh HR này. Bộ dữ liệu huấn luyện giúp mô hình học cách chuyển đổi từ ảnh LR sang ảnh HR.

* DIV2K Dataset:
* DIV2K là bộ dữ liệu nổi tiếng cho các tác vụ cải thiện độ phân giải ảnh. Bộ dữ liệu này bao gồm 2.000 ảnh với độ phân giải cao (HR) từ các thể loại như cảnh vật, con người, vật thể, v.v.
* Các ảnh trong bộ dữ liệu này có thể được giảm kích thước để tạo ra các ảnh LR, từ đó sử dụng chúng để huấn luyện mô hình SRCNN.
* Bộ dữ liệu DIV2K thường được sử dụng cho các bài toán Super-Resolution trong các cuộc thi như NTIRE (New Trends in Image Restoration and Enhancement).

*Link:* [*https://paperswithcode.com/dataset/cifar-10*](https://paperswithcode.com/dataset/cifar-10)

* BSD100 (Berkeley Segmentation Dataset):
* BSD100 là bộ dữ liệu với 100 ảnh có độ phân giải cao (HR), được sử dụng trong các nghiên cứu về phân đoạn ảnh và siêu phân giải ảnh.
* Bộ dữ liệu này có thể được dùng để giảm kích thước (downscale) các ảnh gốc và tạo ra ảnh LR tương ứng.

*Link:* [*https://paperswithcode.com/dataset/bsd100*](https://paperswithcode.com/dataset/bsd100)

* Set5, Set14, và Urban100:
* Đây là các bộ dữ liệu phổ biến trong các nghiên cứu về Super-Resolution, chứa các ảnh với chất lượng cao, chủ yếu là các bức ảnh với các đối tượng tự nhiên, đô thị, hoặc các ảnh thể thao

*Link:* [*https://figshare.com/articles/dataset/BSD100\_Set5\_Set14\_Urban100/21586188*](https://figshare.com/articles/dataset/BSD100_Set5_Set14_Urban100/21586188)

3.1.2 Dữ liệu đưa vào chương trình

* ImageNet: Mặc dù ImageNet chủ yếu dùng cho các bài toán phân loại ảnh, nhưng bạn cũng có thể lấy các ảnh trong ImageNet và tạo các ảnh có độ phân giải thấp (LR) để huấn luyện mô hình.

*Link:* [*https://www.image-net.org/*](https://www.image-net.org/)

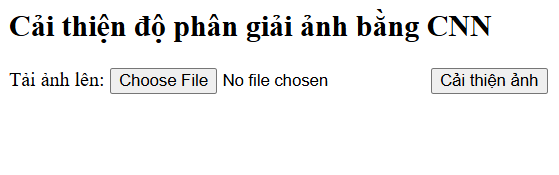
* COCO (Common Objects in Context): Bộ dữ liệu COCO cũng chứa nhiều ảnh với các đối tượng đa dạng và có thể sử dụng để tạo ra các ảnh LR cho bài toán Super-Resolution.

*Link:* [*https://cocodataset.org/#home*](https://cocodataset.org/#home)

## 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



Hình 7: Màn hình thao tác người dùng

### 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



Hình 8: Màn hình kết quả

# Kết luận

## 1. 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.

## 2. 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.

## 3. 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).

## 4. 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.*