**BỘ GIÁO DỤC VÀ ĐÀO TẠO**

**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**

|  |  |  |
| --- | --- | --- |
| **Sinh viên thực hiện** | **Lớp** | **Khoá** |
| Nguyễn Hoàng Gia Phúc | DCCNTT12.10.2 | K12 |
| Phạm Văn Hiếu | DCCNTT12.10.2 | K12 |
| Lê Ngọc Thái | DCCNTT12.10.2 | K12 |
| Đỗ Tuấn Anh | DCCNTT12.10.2 | K12 |
| Nguyễn Hữu Hải | DCCNTT12.10.2 | K12 |

**Bắc Ninh, năm 2024**

**BỘ GIÁO DỤC VÀ ĐÀO TẠO**

**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**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **Sinh viên thực hiện** | **Mã sinh viên** | **Điểm bằng số** | **Điểm bằng chữ** |
| 1 | Nguyễn Hoàng Gia Phúc | 20210566 |  |  |
| 2 | Phạm Văn Hiếu | 20210355 |  |  |
| 3 | Lê Ngọc Thái | 20210363 |  |  |
| 4 | Đỗ Tuấn Anh | 20210411 |  |  |
| 5 | Nguyễn Hữu Hải | 20210354 |  |  |

|  |  |
| --- | --- |
| **CÁN BỘ CHẤM 1**  *(Ký và ghi rõ họ tên)* | **CÁN BỘ CHẤM 2**  *(Ký và ghi rõ họ tên)* |

|  |  |
| --- | --- |
| 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:**  **XATGMT.03.K12.02.LH.C04.1\_LT** | |
| **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** | Nguyễn Hoàng Gia Phúc | Phạm Văn Hiếu | Lê Ngọc Thái | Đỗ Tuấn Anh | Nguyễn Hữu Hải |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 20210566 | 20210355 | 20210363 | 20210411 | 20210354 |
| **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**

[**DANH MỤC HÌNH ẢNH** vii](#_Toc184581666)

[**DANH MỤC BẢNG** viii](#_Toc184581667)

[**BẢNG PHÂN CÔNG CÔNG VIỆC** ix](#_Toc184581668)

[**LỜI NÓI ĐẦU** x](#_Toc184581669)

[**LỜI CẢM ƠN** xi](#_Toc184581670)

[**CHƯƠNG 1: CÁC KIẾN THỨC CƠ SỞ** 1](#_Toc184581671)

[**1.1. Nhận dạng đối tượng trong đề tài** 1](#_Toc184581672)

[**1.1.1. Tổng quan về nhận dạng đối tượng** 1](#_Toc184581673)

[**1.1.2. Tổng quan về các kỹ thuật tăng độ phân giải ảnh hiệu quả** 5](#_Toc184581674)

[**1.2. Các công cụ và ngôn ngữ sử dụng** 6](#_Toc184581675)

[**1.2.1. Ngôn ngữ lập trình python** 6](#_Toc184581676)

[**1.2.2. Công cụ phát triển PyCharm Community Edition** 9](#_Toc184581677)

[**CHƯƠNG 2: XÂY DỰNG HỆ THỐNG** 11](#_Toc184581678)

[**2.1. Phát biểu và mô tả bài toán** 11](#_Toc184581679)

[**2.2. Xây dựng hệ thống** 14](#_Toc184581680)

[**2.2.1. Các loại kỹ thuật sử dụng** 14](#_Toc184581681)

[**2.2.2. Các thư viện chính được sử dụng trong hệ thống** 17](#_Toc184581682)

[**2.2.3. Các phương thức huấn luyện mô hình và tăng độ phân giải của ảnh** 18](#_Toc184581683)

[**CHƯƠNG 3: KẾT QUẢ THỰC NGHIỆM** 22](#_Toc184581684)

[**3.1. Demo ứng dụng** 22](#_Toc184581685)

[**3.2. Cách thức hoạt động của ứng dụng** 23](#_Toc184581686)

[**3.3. Kết quả thực nghiệm ứng dụng** 25](#_Toc184581687)

[**KẾT LUẬN** 28](#_Toc184581688)

[**TÀI LIỆU THAM KHẢO** 30](#_Toc184581689)

## **DANH MỤC HÌNH ẢNH**

[Hình 1.1. Ví dụ minh họa về nhận dạng đối tượng trong bối cảnh giao thôn…..…….12](#_Toc184581251)

[Hình 1.2. Ví dụ minh họa về nhận dạng đối tượng trong bối cảnh y tế 13](#_Toc184581252)

[Hình 2.1. Ví dụ ảnh LR và HR………………...……………………………………...23](#_Toc184581263)

[Hình 2.2. Kết quả huấn luyện mô hình 31](#_Toc184581264)

[Hình 3.1. Giao diện chính của ứng dụng…………...…………………………………32](#_Toc184581269)

[Hình 3.2. Kết quả thực nghiệm ứng dụng 35](#_Toc184581270)

## **DANH MỤC BẢNG**

[Bảng 2.1. Các thư viện chính được sử dụng 27](#_Toc184581477)

## **BẢNG PHÂN CÔNG CÔNG VIỆC**

|  |  |  |
| --- | --- | --- |
| **Thành viên** | **Công Việc** | **Đánh Giá** |
| Nguyễn Hoàng Gia Phúc | 1.Tìm hiểu, lên kế hoạch thực hiện đề tài  2. Xây dưng và tổ chức ứng dụng | Tích cực |
| Phạm Văn Hiếu | 1. Nghiên cứu thuật toán  2. Thu thập dữ liệu  3. Thực hiện chương 1 file word báo cáo | Tích cực |
| Đố Tuấn Anh | 1. Nghiên cứu thuật toán  2. Thu thập dữ liệu  3. Thực hiện chương 2 file word báo cáo | Tích cực |
| Lê Ngọc Thái | 1. Nghiên cứu thuật toán  2. Thực hiện chương 3 file word báo cáo  3. Cải tiến hiệu suất huấn luyện | Tích cực |
| Nguyễn Hữu Hải | 1. Nghiên cứu thuật toán  2. Thực hiện chương 3 file word báo cáo  3. Đánh giá mô hình huấn luyện | Tích cực |

## **LỜI NÓI ĐẦU**

Xử lý ảnh và thị giác máy tính là một lĩnh vực khoa học công nghệ quan trọng, đóng vai trò nền tảng trong nhiều ứng dụng thực tiễn như nhận dạng khuôn mặt, giám sát giao thông, và cải thiện chất lượng hình ảnh, cải thiện độ phân giải ảnh trong nhiều lĩnh vực như trong y tế, giáo dục. Trong đó, việc cải thiện độ phân giải ảnh là một bài toán quan trọng nhằm nâng cao chất lượng hình ảnh, giúp tối ưu hóa kết quả trong các hệ thống xử lý và phân tích ảnh.

Với sự phát triển của công nghệ học sâu, đặc biệt là mạng nơ-ron tích chập (Convolutional Neural Network - CNN), các bài toán liên quan đến cải thiện độ phân giải ảnh đã đạt được những tiến bộ vượt bậc. CNN không chỉ có khả năng học và trích xuất các đặc trưng của ảnh mà còn được ứng dụng hiệu quả trong tái tạo và nâng cấp chi tiết từ ảnh có độ phân giải thấp.

Trong bối cảnh đó, nhóm chúng em đã lựa chọn thực hiện đề tài “Xây dựng hệ thống cải thiện độ phân giải ảnh bằng CNN”. Mục tiêu của đề tài là nghiên cứu lý thuyết nền tảng, áp dụng các phương pháp hiện đại để xây dựng hệ thống nâng cao độ phân giải ảnh và kiểm chứng hiệu quả qua các tập dữ liệu thực nghiệm.

Đề tài không chỉ giúp chúng em củng cố kiến thức lý thuyết mà còn phát triển kỹ năng thực hành, tư duy giải quyết vấn đề, đồng thời mở rộng hiểu biết về tiềm năng ứng dụng của công nghệ học sâu trong thực tế. Trong suốt quá trình thực hiện, nhóm đã gặp không ít khó khăn, nhưng nhờ sự hỗ trợ tận tình từ giảng viên hướng dẫn và các tài liệu tham khảo hữu ích, chúng em đã hoàn thành đề tài đúng tiến độ.

.

## **LỜI CẢM ƠN**

Trước hết, chúng em xin gửi lời cảm ơn chân thành đến tập thể quý Thầy Cô trường Đại học Công Nghệ Đông Á, đặc biệt là quý Thầy Cô khoa Công Nghệ Thông Tin, đã tận tình giảng dạy và truyền đạt cho chúng em những kiến thức quý báu. Chính nhờ nền tảng vững chắc này, chúng em mới có đủ tự tin và năng lực để thực hiện đề tài "Xây dựng hệ thống cải thiện độ phân giải ảnh bằng CNN" trong khuôn khổ học phần Xử lý ảnh và thị giác máy tính. Những bài học từ Thầy Cô không chỉ là hành trang quý giá mà còn là nguồn động lực mạnh mẽ giúp chúng em vượt qua những thử thách trong quá trình học tập và nghiên cứu.

Chúng em đặc biệt xin gửi lời cảm ơn sâu sắc tới Cô Lương Thị Hồng Lan, người đã dành nhiều thời gian và tâm huyết để hướng dẫn chúng em trong suốt quá trình thực hiện đề tài. Những chỉ dẫn tận tâm của Thầy không chỉ giúp chúng em hiểu rõ hơn về việc áp dụng các thuật toán học sâu, đặc biệt là mạng nơ-ron tích chập (CNN), vào bài toán nâng cao chất lượng hình ảnh mà còn khuyến khích chúng em suy nghĩ sáng tạo, tìm tòi và áp dụng kiến thức để xây dựng một hệ thống hiệu quả, có tính ứng dụng cao. Sự tận tụy và những góp ý chi tiết của Cô đã giúp chúng em vượt qua những khó khăn trong quá trình nghiên cứu và phát triển. Những lời khuyên và định hướng của Cô giúp chúng em tự tin hơn trong từng bước tiến hành.

Trong suốt quá trình thực hiện đề tài, chúng em đã nỗ lực vận dụng những kiến thức đã học, đồng thời không ngừng tìm hiểu thêm từ các tài liệu chuyên sâu và các nguồn tài nguyên khác để đảm bảo hệ thống cải thiện độ phân giải ảnh không chỉ đáp ứng các yêu cầu kỹ thuật mà còn đạt hiệu quả thực nghiệm cao. Tuy nhiên, do hạn chế về thời gian và kinh nghiệm, sản phẩm của chúng em không thể tránh khỏi những thiếu sót. Chúng em rất mong nhận được những ý kiến đóng góp và phản hồi từ Cô để có thể tiếp tục hoàn thiện kỹ năng và kiến thức của mình, từ đó chuẩn bị tốt hơn cho những dự án và công việc trong tương lai.

Chúng em xin chân thành cảm ơn!

## **CHƯƠNG 1: CÁC KIẾN THỨC CƠ SỞ**

## **1.1. Nhận dạng đối tượng trong đề tài**

### **1.1.1. Tổng quan về nhận dạng đối tượng**

Nhận dạng đối tượng trong ảnh là một trong những bài toán trọng tâm của lĩnh vực Xử lý ảnh và Thị giác máy tính, với mục tiêu phát hiện, định vị và phân loại các đối tượng xuất hiện trong ảnh số hoặc video. Đây là một bước quan trọng để giúp hệ thống máy tính hiểu được nội dung hình ảnh, từ đó hỗ trợ các ứng dụng thực tiễn như giám sát an ninh, giao thông thông minh, và chẩn đoán y khoa.

Trong thời đại công nghệ phát triển, nhận dạng đối tượng không chỉ đơn thuần là phân loại đối tượng, mà còn cần đạt độ chính xác cao trong việc xác định vị trí, kích thước, và các thuộc tính liên quan. Chẳng hạn, trong hệ thống giám sát giao thông, việc nhận dạng đối tượng giúp xác định các loại phương tiện (ô tô, xe máy, xe tải) cùng vị trí của chúng, qua đó hỗ trợ quản lý và điều phối giao thông hiệu quả hơn. Tương tự, trong lĩnh vực y tế, nhận dạng các bất thường trong ảnh X-quang hoặc MRI có thể hỗ trợ bác sĩ phát hiện sớm bệnh lý, nâng cao hiệu quả chẩn đoán và điều trị.

Công nghệ hiện đại đang đặt nền tảng cho sự phát triển mạnh mẽ của các hệ thống nhận dạng đối tượng, trong đó mạng nơ-ron tích chập (Convolutional Neural Networks - CNN) đóng vai trò cốt lõi. CNN có khả năng tự động học các đặc trưng quan trọng từ dữ liệu ảnh mà không cần can thiệp thủ công, từ đó cải thiện đáng kể hiệu suất và độ chính xác so với các phương pháp truyền thống.

Tuy nhiên, một thách thức lớn đối với bài toán nhận dạng đối tượng là chất lượng của ảnh đầu vào, đặc biệt là độ phân giải. Ảnh có độ phân giải thấp thường mất thông tin chi tiết, dẫn đến giảm hiệu quả nhận dạng, đặc biệt trong các ứng dụng yêu cầu độ chính xác cao. Để khắc phục vấn đề này, việc tăng cường chất lượng ảnh đầu vào trở thành một bước quan trọng. Các phương pháp cải thiện độ phân giải ảnh, đặc biệt là các kỹ thuật dựa trên CNN, không chỉ giúp tái tạo chi tiết ảnh mà còn nâng cao hiệu suất nhận dạng.

Do đó, bài toán nhận dạng đối tượng không chỉ tập trung vào việc xây dựng các mô hình nhận dạng tiên tiến, mà còn đòi hỏi sự tích hợp chặt chẽ với các hệ thống cải thiện chất lượng ảnh. Điều này đảm bảo rằng các ứng dụng trong thực tiễn, từ giám sát giao thông đến chẩn đoán y khoa, đạt được hiệu quả tối ưu cả về tốc độ và độ chính xác.

**Ví dụ về nhận dạng đối tượng trong ảnh theo đề tài tăng độ phân giải ảnh:**

**Ví dụ 1: Trong bối cảnh giám sát giao thông**

**A collage of a traffic light

Description automatically generated**

Hình 1.1. Ví dụ minh họa về nhận dạng đối tượng trong bối cảnh giao thông

Trong bài toán thực tế, việc nhận dạng đối tượng trong ảnh có độ phân giải thấp thường gặp nhiều khó khăn, đặc biệt khi chi tiết của các đối tượng bị mờ hoặc biến dạng. Hãy xem xét một ví dụ cụ thể:

**Bối cảnh:** Một hệ thống giám sát giao thông cần nhận dạng các phương tiện trên đường, như ô tô, xe máy, và xe buýt, từ các camera giám sát.

Ảnh đầu vào có độ phân giải thấp:

Hình ảnh thu được từ camera ở xa hoặc trong điều kiện ánh sáng kém có độ phân giải thấp.

Các phương tiện trên đường bị mờ, khiến hệ thống nhận dạng không thể phân biệt chính xác các loại phương tiện. Ví dụ: hệ thống có thể nhầm lẫn giữa xe máy và xe ô tô nhỏ do thiếu thông tin chi tiết.

Hậu quả là hệ thống không thể thực hiện các nhiệm vụ quan trọng như đếm số lượng phương tiện hoặc phát hiện vi phạm giao thông.

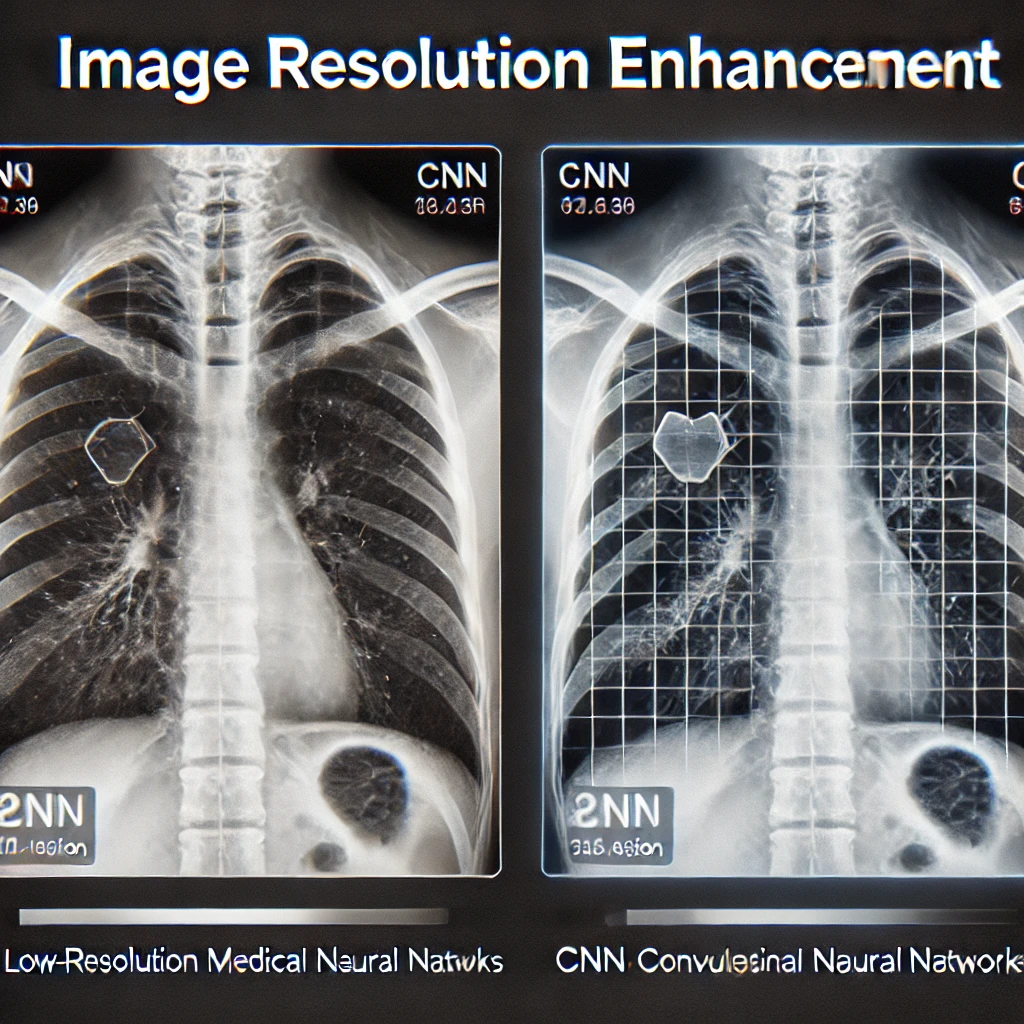
Ảnh sau khi tăng độ phân giải:

Sử dụng kỹ thuật tăng độ phân giải bằng CNN, ảnh được cải thiện chất lượng, chi tiết như các đường viền, bánh xe, và đèn xe được tái tạo rõ ràng hơn.

Sau khi cải thiện độ phân giải, hệ thống nhận dạng đối tượng có thể dễ dàng phân biệt giữa các phương tiện khác nhau. Ví dụ: xe buýt được nhận dạng chính xác nhờ vào kích thước lớn và chi tiết như cửa sổ, trong khi xe máy được phát hiện nhờ hình dạng đặc trưng.

Điều này giúp tăng hiệu quả của hệ thống giám sát, đảm bảo nhận dạng đúng và hỗ trợ các chức năng như phân tích lưu lượng, phát hiện vi phạm, hoặc quản lý giao thông

**Ví dụ 2: Trong bối cảnh y tế:**



Hình 1.2. Ví dụ minh họa về nhận dạng đối tượng trong bối cảnh y tế

**Trong y tế: Ảnh có độ phân giải thấp thường xuất hiện trong các tình huống như:**

**Hạn chế về thiết bị chụp ảnh**: Các thiết bị cảm biến cũ hoặc giá rẻ thường chỉ cung cấp ảnh với độ phân giải thấp, không thể hiện được đầy đủ chi tiết của đối tượng.

**Yêu cầu tiết kiệm băng thông**: Trong các hệ thống truyền tải dữ liệu (như livestream, giám sát từ xa), ảnh hoặc video thường được nén để giảm dung lượng, dẫn đến mất chi tiết khi tái tạo.

**Ảnh hưởng từ môi trường**: Các yếu tố như ánh sáng yếu, khoảng cách xa, hoặc rung lắc trong quá trình chụp có thể làm giảm chất lượng và độ phân giải của ảnh thu được.

Với các ảnh có độ phân giải thấp, các chi tiết quan trọng thường bị mất mát hoặc mờ đi, gây khó khăn cho các bài toán xử lý ảnh và thị giác máy tính, đặc biệt là nhận dạng đối tượng. Các hệ thống nhận dạng đối tượng hoặc phân tích ảnh hiện đại thường yêu cầu ảnh đầu vào với chất lượng cao để đạt độ chính xác tối ưu. Do đó, việc tăng độ phân giải ảnh không chỉ cải thiện tính thẩm mỹ mà còn nâng cao hiệu suất và độ tin cậy của các ứng dụng thực tế.

Để giải quyết bài toán này, các phương pháp cải thiện độ phân giải ảnh, hay còn gọi là tăng độ phân giải ảnh (Super-Resolution), đã và đang được nghiên cứu mạnh mẽ. Phương pháp truyền thống thường dựa trên các kỹ thuật nội suy (interpolation) như nội suy tuyến tính (bilinear interpolation) hay nội suy bậc ba (bicubic interpolation). Tuy nhiên, các phương pháp này có hạn chế lớn là không thể tái tạo chi tiết phức tạp và thường tạo ra các đường viền mờ hoặc hiện tượng nhiễu.

Trong những năm gần đây, sự phát triển của học sâu (Deep Learning) đã mang lại bước tiến lớn cho bài toán tăng độ phân giải ảnh. Các mô hình mạng nơ-ron tích chập (CNN) và mạng đối kháng sinh (Generative Adversarial Networks - GAN) cho phép học các đặc trưng phức tạp từ dữ liệu, từ đó tái tạo ảnh có độ phân giải cao một cách tự nhiên và chi tiết hơn. CNN không chỉ cải thiện khả năng phục hồi chi tiết mà còn đảm bảo tính khả dụng trong các hệ thống thực tế nhờ tốc độ xử lý nhanh.

### **1.1.2. Tổng quan về các kỹ thuật tăng độ phân giải ảnh hiệu quả**

Tăng độ phân giải ảnh bằng mạng nơ-ron tích chập (Convolutional Neural Networks - CNN) là một trong những hướng tiếp cận tiên tiến và hiệu quả nhất hiện nay. CNN có khả năng học các đặc trưng phức tạp của dữ liệu và tái tạo ảnh có độ phân giải cao (HR) từ ảnh có độ phân giải thấp (LR). Các kỹ thuật này đã mở ra bước ngoặt quan trọng trong lĩnh vực xử lý ảnh và thị giác máy tính.

**Tổng quan về CNN trong tăng độ phân giải ảnh**

CNN được sử dụng để trích xuất đặc trưng từ ảnh đầu vào thông qua các lớp tích chập (convolutional layers), sau đó tái tạo ảnh HR. Quá trình này bao gồm:

**Tách đặc trưng (Feature Extraction)**:  
Các lớp tích chập học cách phát hiện các đặc trưng quan trọng (biên, góc, họa tiết) từ ảnh LR.

**Tái tạo không gian (Spatial Reconstruction)**:  
Dựa trên các đặc trưng đã học, CNN tái tạo lại ảnh HR bằng cách dự đoán giá trị điểm ảnh mới.

**Tăng cường độ phân giải (Upsampling)**:  
Kỹ thuật upsampling (như bilinear hoặc pixel shuffle) được sử dụng để tăng kích thước ảnh trong mạng.

**Các phương pháp CNN phổ biến**

**SRCNN (Super-Resolution Convolutional Neural Network)**

SRCNN là một trong những mô hình CNN đầu tiên được sử dụng cho tăng độ phân giải ảnh.

**Cấu trúc**:

Gồm ba lớp tích chập chính: trích xuất đặc trưng, phi tuyến, và tái tạo.

Ảnh đầu vào được nội suy trước (e.g., bicubic) để tăng kích thước, sau đó đưa vào CNN.

**Ưu điểm**: Dễ triển khai, hiệu quả với ảnh đơn giản.

**Nhược điểm**: Hiệu năng không cao với ảnh phức tạp.

**FSRCNN (Fast Super-Resolution Convolutional Neural Network)**

Phiên bản cải tiến của SRCNN, giảm thời gian tính toán.

**Cấu trúc**:

Bỏ qua bước nội suy, sử dụng module tăng kích thước ảnh (deconvolution) trong mạng.

**Ưu điểm**: Nhanh hơn SRCNN, phù hợp với thời gian thực.

**Nhược điểm**: Chất lượng thấp hơn các mô hình phức tạp hơn.

**VDSR (Very Deep Super-Resolution)**

Mô hình CNN sâu với hàng chục lớp tích chập.

**Cấu trúc**:

Sử dụng một mạng sâu để học độ chênh lệch giữa ảnh LR và HR.

Tăng cường khả năng tổng quát hóa nhờ việc sử dụng regularization.

**Ưu điểm**: Độ chính xác cao hơn SRCNN và FSRCNN.

**Nhược điểm**: Yêu cầu tài nguyên tính toán lớn.

**ESPCN (Efficient Sub-Pixel Convolutional Neural Network)**

Sử dụng cơ chế phân bố điểm ảnh (Pixel Shuffle) để tăng độ phân giải.

**Cấu trúc**:

Học trực tiếp từ ảnh LR mà không cần nội suy trước.

Chuyển đổi không gian ảnh thông qua lớp Sub-Pixel.

**Ưu điểm**: Tăng hiệu suất tính toán và chất lượng hình ảnh.

## **1.2. Các công cụ và ngôn ngữ sử dụng**

### **1.2.1. Ngôn ngữ lập trình python**

Python là ngôn ngữ lập trình bậc cao, dễ học và phổ biến rộng rãi, được thiết kế để tối ưu hóa năng suất của nhà phát triển. Python hỗ trợ lập trình đa dạng từ lập trình thủ tục, hướng đối tượng đến lập trình hàm, giúp nó trở thành lựa chọn hàng đầu cho các dự án liên quan đến trí tuệ nhân tạo, học máy, và xử lý ảnh.

**a) Ưu điểm của Python:**

**Dễ học và sử dụng**  
Python có cú pháp đơn giản, gần gũi với ngôn ngữ tự nhiên, giúp các lập trình viên dễ dàng tiếp cận và viết mã một cách hiệu quả. Điều này đặc biệt hữu ích khi thực hiện các dự án phức tạp như tăng độ phân giải ảnh bằng CNN, nơi các công cụ cần được sử dụng nhanh chóng và chính xác.

**Thư viện phong phú**  
Python sở hữu hệ sinh thái thư viện và framework rất phong phú, hỗ trợ mạnh mẽ trong các lĩnh vực như xử lý ảnh, học sâu và trí tuệ nhân tạo. Các thư viện như OpenCV, NumPy, TensorFlow và PyTorch cung cấp các công cụ cần thiết để thực hiện các phép tính ma trận, xây dựng mô hình học sâu và tối ưu hóa quá trình huấn luyện.

**Cộng đồng lớn và hỗ trợ mạnh mẽ**  
Với cộng đồng người dùng rộng lớn, Python có sẵn rất nhiều tài liệu học tập, diễn đàn thảo luận và mã nguồn mở, giúp dễ dàng tìm kiếm sự hỗ trợ khi gặp khó khăn trong quá trình phát triển.

**Khả năng tương thích và tích hợp**  
Python dễ dàng tích hợp với các ngôn ngữ lập trình và công nghệ khác, giúp tối ưu hóa hiệu suất xử lý và triển khai các giải pháp liên quan đến học sâu và xử lý ảnh.

**b) Python trong học sâu và xử lý ảnh:**

Trong bài toán tăng độ phân giải ảnh, Python được lựa chọn nhờ khả năng hỗ trợ mạnh mẽ các thư viện và công cụ học sâu:

Thư viện NumPy

Cung cấp các công cụ mạnh mẽ cho tính toán ma trận và đại số tuyến tính, một phần quan trọng trong việc biểu diễn và xử lý dữ liệu hình ảnh.

Thư viện OpenCV

Hỗ trợ các tác vụ xử lý ảnh như đọc/ghi ảnh, thay đổi kích thước, chuyển đổi không gian màu, giúp tiền xử lý ảnh dễ dàng và hiệu quả.

TensorFlow và PyTorch

Hai thư viện mạnh mẽ hỗ trợ xây dựng và huấn luyện các mạng nơ-ron tích chập (CNN). Đây là thành phần chính trong các mô hình tăng độ phân giải ảnh, cho phép học các đặc trưng quan trọng từ ảnh có độ phân giải thấp.

Keras

Giao diện cấp cao của TensorFlow, giúp xây dựng và huấn luyện các mô hình học sâu nhanh chóng với cú pháp đơn giản.

Matplotlib và Seaborn

Cung cấp các công cụ trực quan hóa, giúp hiển thị kết quả tăng độ phân giải ảnh và phân tích quá trình huấn luyện mô hình một cách hiệu quả.

**c) Ứng dụng Python trong bài toán tăng độ phân giải ảnh bằng CNN**

Tiền xử lý dữ liệu ảnh

Đọc và xử lý ảnh: Sử dụng OpenCV để đọc ảnh đầu vào, chuyển đổi sang dạng ma trận và chuẩn hóa dữ liệu để phù hợp với mô hình CNN.

Tăng cường dữ liệu (Data Augmentation): Áp dụng các kỹ thuật như xoay, cắt, và thay đổi độ sáng để tạo ra nhiều dữ liệu huấn luyện hơn từ tập ảnh gốc, tăng độ chính xác của mô hình.

Xây dựng mô hình CNN

Mạng nơ-ron tích chập: CNN được sử dụng để học các đặc trưng từ ảnh đầu vào (low-resolution) và tái tạo ảnh có độ phân giải cao (high-resolution).

Kỹ thuật upsampling: Sử dụng các tầng deconvolution hoặc Sub-pixel để tăng kích thước ảnh trong mạng.

Huấn luyện mô hình

Python cho phép thiết lập các hàm mất mát tùy chỉnh như Mean Squared Error (MSE) hoặc Perceptual Loss để tối ưu hóa mô hình.

TensorFlow hỗ trợ các API để huấn luyện và theo dõi hiệu suất mô hình qua các epoch.

Đánh giá và triển khai mô hình

Đánh giá chất lượng ảnh: Sử dụng các chỉ số như PSNR (Peak Signal-to-Noise Ratio) và SSIM (Structural Similarity Index) để đo độ tương đồng giữa ảnh gốc và ảnh tái tạo.

Triển khai mô hình: Sử dụng TensorFlow Lite để tích hợp mô hình tăng độ phân giải ảnh vào các ứng dụng di động hoặc thiết bị nhúng.

Trực quan hóa kết quả

Sử dụng Matplotlib để hiển thị so sánh giữa ảnh đầu vào, ảnh tái tạo và ảnh gốc. Điều này giúp đánh giá trực quan hiệu quả của mô hình CNN.

### **1.2.2. Công cụ phát triển PyCharm Community Edition**

PyCharm Community Edition là một trong những môi trường phát triển tích hợp (IDE) phổ biến nhất dành cho lập trình Python, do JetBrains phát triển. Với giao diện thân thiện và tích hợp nhiều tính năng mạnh mẽ, PyCharm hỗ trợ lập trình viên tối ưu hóa hiệu quả công việc trong các dự án Python, đặc biệt phù hợp cho các dự án học thuật và nghiên cứu như tăng độ phân giải ảnh bằng CNN.

**a) Đặc điểm nổi bật của PyCharm Community Edition**

Hỗ trợ lập trình Python hiệu quả:

PyCharm cung cấp cú pháp tô màu, tự động hoàn thành mã, và gợi ý thông minh. Điều này giúp lập trình viên dễ dàng kiểm soát và phát hiện lỗi ngay khi viết mã.

Có khả năng hỗ trợ nhiều chuẩn Python, bao gồm Python 2.7, Python 3.x.

Hệ thống kiểm tra lỗi tích hợp (Code Analysis):

PyCharm tự động kiểm tra mã nguồn để phát hiện các lỗi cú pháp, cảnh báo khi sử dụng biến hoặc hàm không hợp lệ, giúp giảm thiểu sai sót khi lập trình.

Công cụ gỡ lỗi (Debugger):

Debugger mạnh mẽ tích hợp sẵn cho phép lập trình viên theo dõi trạng thái biến, dòng mã, và kiểm tra logic của chương trình theo thời gian thực. Điều này đặc biệt hữu ích khi kiểm tra thuật toán CNN để tăng độ phân giải ảnh.

Tích hợp hệ thống điều khiển phiên bản (VCS):

Hỗ trợ tích hợp Git, GitHub và các công cụ quản lý mã nguồn khác, giúp làm việc nhóm hiệu quả hơn.

Quản lý thư viện Python:

PyCharm cung cấp giao diện trực quan để cài đặt, gỡ bỏ hoặc cập nhật các thư viện Python từ PyPI, chẳng hạn như TensorFlow, PyTorch, OpenCV, hoặc NumPy. Điều này rất tiện lợi cho các dự án yêu cầu sử dụng nhiều thư viện học sâu và xử lý ảnh.

**b) Tính năng hỗ trợ dự án tăng độ phân giải ảnh bằng CNN**

Hỗ trợ học sâu và mô hình CNN:

Với khả năng tương thích cao với các thư viện như TensorFlow, Keras, và PyTorch, PyCharm giúp dễ dàng xây dựng, huấn luyện, và kiểm tra các mô hình CNN.

Quản lý và xử lý dữ liệu ảnh:

PyCharm tích hợp với các công cụ như OpenCV, Pillow để thao tác ảnh, chuẩn bị dữ liệu đầu vào cho mạng CNN.

Hỗ trợ Jupyter Notebook:

PyCharm Community có khả năng mở và chỉnh sửa các tệp Jupyter Notebook (.ipynb), rất hữu ích để thử nghiệm và ghi chú khi thực hiện các thí nghiệm tăng độ phân giải ảnh.

Hệ thống gỡ lỗi nâng cao:

Hỗ trợ gỡ lỗi các chương trình phức tạp liên quan đến xử lý ảnh và học sâu, giúp theo dõi các bước tính toán bên trong mạng CNN.

**c) Ưu điểm của PyCharm Community Edition cho dự án**

Miễn phí và dễ sử dụng:

Phiên bản Community Edition được cung cấp miễn phí, phù hợp cho các nhóm sinh viên, nghiên cứu hoặc cá nhân có nhu cầu học tập và phát triển.

Tích hợp nhiều công cụ mạnh mẽ:

Không cần cài đặt nhiều phần mềm bổ sung, PyCharm đã hỗ trợ đầy đủ từ viết mã, kiểm tra, gỡ lỗi, đến quản lý thư viện.

Tài liệu và cộng đồng hỗ trợ:

Với cộng đồng người dùng lớn và tài liệu phong phú, lập trình viên có thể dễ dàng tìm kiếm sự trợ giúp khi gặp vấn đề.

**d) Ứng dụng PyCharm Community trong bài toán thực tế**

Trong dự án tăng độ phân giải ảnh bằng CNN, PyCharm được sử dụng để:

Xây dựng cấu trúc mạng CNN: Phát triển các lớp mạng như Convolution, Pooling, và Upsampling bằng TensorFlow hoặc PyTorch.

Huấn luyện và đánh giá mô hình: Quản lý quy trình huấn luyện mạng, tính toán các chỉ số đánh giá như PSNR và SSIM.

Xử lý ảnh: Tiền xử lý dữ liệu đầu vào, như thay đổi kích thước, chuẩn hóa, và tạo dữ liệu tăng cường.

## **CHƯƠNG 2: XÂY DỰNG HỆ THỐNG**

## **2.1. Phát biểu và mô tả bài toán**

Trong nhiều lĩnh vực như y tế, giám sát an ninh, hoặc truyền thông số, chất lượng ảnh đầu ra có vai trò vô cùng quan trọng. Tuy nhiên, do hạn chế của thiết bị chụp ảnh hoặc do các yếu tố khác như nén dữ liệu, ảnh có thể bị giảm chất lượng và độ phân giải (Low-Resolution - LR). Bài toán tăng độ phân giải ảnh (Image Super-Resolution - SR) tập trung vào việc cải thiện độ phân giải của ảnh từ LR sang ảnh có độ phân giải cao (High-Resolution - HR), đồng thời giữ nguyên các đặc điểm quan trọng như chi tiết, màu sắc và cấu trúc của ảnh gốc.

**Mục tiêu chính**:

Tăng cường độ phân giải của ảnh LR sao cho ảnh HR đầu ra có chất lượng tốt, đạt được các chỉ số đánh giá như **PSNR** (Peak Signal-to-Noise Ratio), **SSIM** (Structural Similarity Index) cao hơn so với các phương pháp truyền thống.

Xây dựng một hệ thống tự động dựa trên mô hình học sâu sử dụng mạng nơ-ron tích chập (CNN), áp dụng hiệu quả cho bài toán SR.

**Mô tả chi tiết bài toán**

Đầu vào

Ảnh Low-Resolution (LR):

Là ảnh có độ phân giải thấp, thường bị mất chi tiết hoặc bị mờ.

Kích thước: M×NM \times NM×N pixel, trong đó MMM và NNN nhỏ hơn nhiều so với ảnh HR.

Định dạng ảnh: Các định dạng phổ biến như PNG, JPEG.

Đầu ra

Ảnh High-Resolution (HR):

Là ảnh có độ phân giải cao được tái tạo từ ảnh LR.

Kích thước: s⋅M×s⋅Ns \cdot M \times s \cdot Ns⋅M×s⋅N, trong đó sss là hệ số tăng độ phân giải (scaling factor).

Yêu cầu: Ảnh HR cần có chi tiết rõ nét, cấu trúc tự nhiên, và ít nhiễu hơn.

**Quy trình giải quyết bài toán**

Tiền xử lý dữ liệu:

Thu thập tập dữ liệu gồm ảnh LR và ảnh HR tương ứng.

Chuyển đổi ảnh về dạng ma trận số, chuẩn hóa để đưa vào mô hình học sâu.

Xây dựng mô hình học sâu (CNN):

Thiết kế mạng nơ-ron tích chập với kiến trúc tối ưu như SRCNN (Super-Resolution CNN), VDSR (Very Deep SR), hoặc SRGAN (Super-Resolution GAN).

Mục tiêu: Học các đặc trưng quan trọng của ảnh LR để tái tạo ảnh HR chất lượng cao.

Huấn luyện mô hình:

Sử dụng tập dữ liệu huấn luyện để tối ưu hóa các tham số của mô hình.

Sử dụng hàm mất mát (loss function) phù hợp, ví dụ: Mean Squared Error (MSE) hoặc hàm mất mát SSIM.

Đánh giá và kiểm thử:

Đánh giá chất lượng ảnh HR đầu ra bằng các chỉ số PSNR, SSIM.

Kiểm tra trên tập dữ liệu kiểm thử để đảm bảo khả năng tổng quát hóa.

**Các ứng dụng của bài toán**

Y tế: Cải thiện chất lượng ảnh chụp MRI, CT để hỗ trợ chẩn đoán bệnh.

Giám sát an ninh: Tăng độ phân giải cho các ảnh giám sát từ camera an ninh.

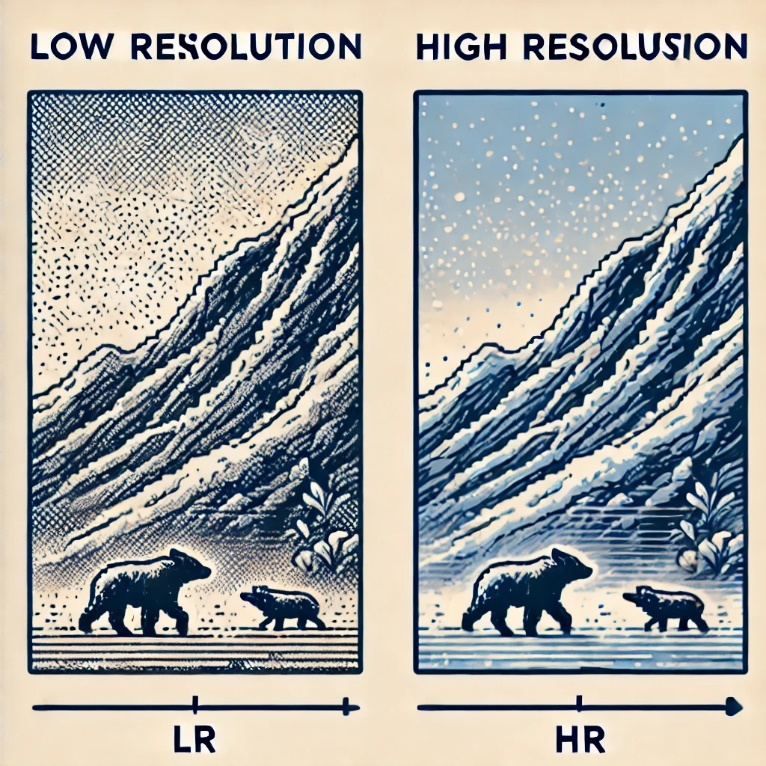
Công nghiệp phim và game: Cải thiện chất lượng ảnh hoặc video để tối ưu hóa trải nghiệm người dùng.

Truyền thông số: Tăng cường chất lượng ảnh nén hoặc ảnh bị mất chi tiết khi truyền tải qua Internet.

**Ví dụ**

Ảnh LR đầu vào: Kích thước 32×3232 \times 3232×32 pixel, chi tiết mờ nhạt.

Ảnh HR đầu ra: Kích thước 128×128128 \times 128128×128 pixel, với các đường nét và chi tiết sắc nét hơn.



Hình 2.1. Ví dụ ảnh LR và HR

Với sự hỗ trợ của mạng CNN, quá trình tăng độ phân giải ảnh không chỉ nhanh hơn mà còn có độ chính xác cao hơn so với các phương pháp nội suy truyền thống như Bilinear hoặc Bicubic.

## **2.2. Xây dựng hệ thống**

### **2.2.1. Các loại kỹ thuật sử dụng**

Các loại kỹ thuật được sử dụng để xây dựng và huấn luyện mô hình nâng cao độ phân giải ảnh (Super-Resolution):

**A. Quản lý dữ liệu:**

**Tải và chuẩn hóa dữ liệu:**

Dữ liệu được tổ chức trong thư mục Dataset, bao gồm các tập:

**DIV2K\_train\_LR\_bicubic**: Ảnh đầu vào độ phân giải thấp (Low Resolution - LR) dùng để huấn luyện.

**DIV2K\_train\_HR**: Ảnh đầu ra độ phân giải cao (High Resolution - HR) dùng làm nhãn thực (ground truth) cho huấn luyện.

**DIV2K\_valid\_LR\_bicubic**: Ảnh LR dùng để xác thực mô hình.

**DIV2K\_valid\_HR**: Ảnh HR dùng làm nhãn thực để đánh giá trên tập xác thực.

Ảnh được chuẩn hóa về giá trị pixel trong khoảng [0, 1] bằng cách chia cho 255. Điều này giúp tối ưu hóa việc huấn luyện và giảm sai số tính toán.

**Tải ảnh từ thư mục:**

Sử dụng hàm load\_images với sự hỗ trợ của thư viện **Keras** (tensorflow.keras.preprocessing.image) để tải và chuyển đổi ảnh về kích thước cố định (128x128).

Tất cả ảnh được đưa về định dạng mảng numpy (numpy array), thuận tiện cho xử lý trong TensorFlow.

**Xử lý dữ liệu bất cân xứng:**

Để đảm bảo số lượng ảnh LR và HR khớp nhau, lấy số lượng tối thiểu giữa hai tập.

**B. Mô hình hóa bằng Convolutional Neural Network (CNN):**

**Kiến trúc mô hình CNN:**

Mô hình được xây dựng dạng **Encoder-Decoder** với các thành phần:

**Encoder**:

Trích xuất đặc trưng từ ảnh LR qua nhiều tầng tích chập (Convolution).

Kết hợp **Batch Normalization**, **Activation (ReLU)**, và **Pooling (MaxPooling2D)** để giảm kích thước không gian và tăng khả năng biểu diễn.

**Decoder**:

Sử dụng các tầng **UpSampling2D** để khôi phục kích thước không gian của ảnh, kết hợp với các tầng **Convolution** và **Activation (ReLU)** để tái tạo ảnh HR.

Tầng cuối cùng dùng hàm kích hoạt **Sigmoid** để đảm bảo giá trị đầu ra nằm trong khoảng [0, 1].

**Chi tiết các tầng:**

**Convolutional Layers:**

Các tầng tích chập sử dụng bộ lọc kích thước (3x3) với số lượng filter tăng dần: 64 → 128 → 256, giúp học đặc trưng ngày càng phức tạp.

**Pooling và UpSampling:**

**MaxPooling2D** giảm kích thước không gian ảnh trong giai đoạn mã hóa.

**UpSampling2D** khôi phục kích thước không gian trong giai đoạn giải mã.

**Regularization:**

**Dropout**: Loại bỏ ngẫu nhiên một số nút (với tỷ lệ 20-30%) để giảm thiểu hiện tượng overfitting.

**Hàm mất mát và tối ưu hóa:**

**Loss Function**: Mean Squared Error (MSE), đo lường sự khác biệt giữa ảnh HR thực và dự đoán.

**Optimizer**: Sử dụng thuật toán Adam với tốc độ học 1e-4.

**C. Kỹ thuật huấn luyện và tối ưu hóa:**

**Huấn luyện:**

Sử dụng hàm model.fit để huấn luyện mô hình với dữ liệu LR và HR.

Tham số:

**Epochs**: Số lần quét qua toàn bộ dữ liệu huấn luyện.

**Batch Size**: Kích thước lô, giúp xử lý hiệu quả trên GPU.

**Callbacks:**

**TrainingProgress**: Theo dõi tiến trình huấn luyện và tính toán các chỉ số như precision, recall, f1\_score trên tập xác thực.

**EarlyStopping**: Dừng sớm khi mô hình không cải thiện trong 5 epoch liên tiếp.

**ReduceLROnPlateau**: Giảm tốc độ học khi không có cải thiện về loss.

**Đánh giá:**

Các chỉ số được tính toán trên tập xác thực:

**Loss**: Độ lỗi MSE.

**Accuracy**: Độ chính xác tái tạo ảnh.

**Precision, Recall, F1 Score**: Đánh giá khả năng tái tạo các điểm ảnh đúng.

**D. Trực quan hóa và lưu kết quả:**

**Lưu mô hình:**

Mô hình được lưu dưới dạng file .h5 (trained\_model.h5) để tái sử dụng.

**Lưu lịch sử huấn luyện:**

Lưu lịch sử về các chỉ số (loss, accuracy, precision, recall, f1\_score) vào file .csv (training\_history.csv).

**Trực quan hóa:**

Biểu đồ thể hiện tiến trình huấn luyện (loss, accuracy, precision, recall, f1\_score) qua các epoch được vẽ bằng **Matplotlib** và lưu trong thư mục training\_plots (training\_metrics.png).

**E. Cấu trúc hệ thống:**

**Tổ chức file:**

Dataset/: Chứa dữ liệu ảnh.

training\_plots/: Lưu đồ thị huấn luyện.

trained\_model.h5: Mô hình đã huấn luyện.

training\_history.csv: Lịch sử huấn luyện.

**Các file mã nguồn:**

**Main.py:** Chạy chương trình chính.

**Training.py:** Huấn luyện mô hình.

**Update.py:** Cập nhật và thử nghiệm mô hình.

### **2.2.2. Các thư viện chính được sử dụng trong hệ thống**

Bảng 2.1. Các thư viện chính được sử dụng

|  |  |  |
| --- | --- | --- |
| **Thư viện** | **Chức năng chính** | **Đánh giá** |
| TensorFlow/Keras | - Cung cấp các công cụ xây dựng và huấn luyện mô hình học sâu (Deep Learning), đặc biệt là CNN.  - Hỗ trợ xử lý Tensor, tối ưu hóa loss, và triển khai mô hình SR như SRCNN, VDSR. | - Ưu điểm: Dễ sử dụng, hỗ trợ GPU/TPU tăng tốc huấn luyện, tài liệu phong phú.  - Nhược điểm: Yêu cầu phần cứng mạnh, khó tối ưu trên tập dữ liệu lớn. |
| NumPy | - Hỗ trợ thao tác với dữ liệu ảnh dưới dạng ma trận số.  - Chuyển đổi định dạng ảnh để chuẩn bị đầu vào cho mô hình học sâu. | - Ưu điểm: Xử lý dữ liệu nhanh, ổn định, hỗ trợ tích hợp tốt với TensorFlow/Keras.  - Nhược điểm: Không hỗ trợ GPU, chỉ hoạt động tốt trên CPU. |
| Matplotlib | - Hiển thị biểu đồ theo dõi các chỉ số: Loss, PSNR, SSIM qua các epoch huấn luyện.  - Phân tích hiệu suất của mô hình SR thông qua các biểu đồ. | - Ưu điểm: Hỗ trợ trực quan hóa tốt, dễ dàng xuất biểu đồ đánh giá mô hình tăng độ phân giải.  - Nhược điểm: Không hiệu quả khi làm việc với dữ liệu lớn hoặc ma trận Tensor. |
| tqdm | - Hiển thị thanh tiến trình trong quá trình tải dữ liệu và huấn luyện mô hình.  - Theo dõi trực quan quá trình đọc ảnh và huấn luyện mạng CNN. | - Ưu điểm: Cải thiện trải nghiệm người dùng, giúp giám sát tiến độ dễ dàng.  - Nhược điểm: Chỉ cung cấp thông tin tiến độ, không hỗ trợ đánh giá chi tiết các tác vụ. |
| scikit-learn | - Cung cấp công cụ tính các chỉ số như Precision, Recall, F1-Score, PSNR, SSIM để đánh giá chất lượng mô hình.  - Hỗ trợ đánh giá hiệu quả mô hình SR trên cả ảnh thực tế lẫn ảnh tổng hợp. | - Ưu điểm: Tích hợp dễ dàng với dữ liệu đầu ra của TensorFlow/Keras.  - Nhược điểm: Không hỗ trợ GPU, hiệu suất hạn chế trên tập dữ liệu lớn. |
| os | - Quản lý và thao tác với hệ thống tệp, hỗ trợ truy xuất và tải ảnh từ thư mục.  - Tự động phân loại ảnh LR và HR trong các tập dữ liệu huấn luyện và đánh giá. | - Ưu điểm: Đơn giản, hỗ trợ linh hoạt việc xử lý dữ liệu đầu vào cho mạng học sâu.  - Nhược điểm: Cần kết hợp với thư viện khác như PIL hoặc OpenCV để xử lý ảnh chi tiết. |
| time | - Đo thời gian cho các bước huấn luyện như mỗi epoch hoặc toàn bộ quá trình.  - Đánh giá tốc độ xử lý khi thay đổi kiến trúc mạng hoặc kích thước tập dữ liệu. | - Ưu điểm: Giám sát hiệu suất thời gian, hỗ trợ tối ưu hóa quá trình thực thi mô hình.  - Nhược điểm: Chỉ ghi nhận thời gian tổng quát, không chi tiết theo bước huấn luyện cụ thể. |

### **2.2.3. Các phương thức huấn luyện mô hình và tăng độ phân giải của ảnh**

**1. Mô hình CNN được huấn luyện thông qua các bước chi tiết sau:**

**A. Chuẩn bị dữ liệu**

**Dữ liệu huấn luyện (Train) và xác thực (Validation):**

Ảnh độ phân giải thấp (**Low Resolution - LR**) và độ phân giải cao (**High Resolution - HR**) được tải từ các thư mục.

Các ảnh được resize về kích thước chuẩn (128, 128) và chuẩn hóa giá trị pixel về khoảng [0, 1].

**Xử lý dữ liệu:**

Tải ảnh từ thư mục bằng hàm load\_images().

Chuyển đổi ảnh thành mảng numpy (NumPy array) để tương thích với TensorFlow.

**Phân bổ dữ liệu:**

Tập huấn luyện bao gồm dữ liệu từ DIV2K\_train\_LR\_bicubic và DIV2K\_train\_HR.

Tập xác thực bao gồm dữ liệu từ DIV2K\_valid\_LR\_bicubic và DIV2K\_valid\_HR.

**B. Kiến trúc và tối ưu hóa mô hình**

**Xây dựng mô hình CNN:**

Mô hình CNN có cấu trúc Encoder-Decoder:

**Encoder:** Trích xuất đặc trưng bằng các tầng tích chập (Convolution), chuẩn hóa (BatchNormalization) và giảm kích thước (MaxPooling).

**Decoder:** Tăng kích thước ảnh qua các tầng UpSampling kết hợp Convolution để tái tạo ảnh HR từ ảnh LR.

Sử dụng hàm kích hoạt ReLU cho các tầng ẩn và sigmoid cho tầng đầu ra.

**Thông số kỹ thuật:**

Optimizer: Adam với learning rate 1e-4.

Loss function: **Mean Squared Error (MSE)** đo lường sai số giữa ảnh HR thực và dự đoán.

Metrics: Đo lường độ chính xác (accuracy).

**C. Quá trình huấn luyện**

**Cấu hình callback:**

**TrainingProgress:** Ghi nhận tiến trình và tính toán các chỉ số như Precision, Recall, và F1-Score trên tập xác thực.

**EarlyStopping:** Dừng sớm nếu lỗi (loss) trên tập xác thực không giảm sau 5 epoch liên tiếp.

**ReduceLROnPlateau:** Giảm learning rate khi loss không cải thiện trong 3 epoch, đảm bảo hội tụ tối ưu.

**Huấn luyện mô hình:**

Sử dụng model.fit() với các tham số:

batch\_size=32

epochs=50 (tăng số epoch nếu cần khi áp dụng trên toàn bộ tập dữ liệu lớn).

**Lưu trữ kết quả:**

Mô hình được lưu vào file trained\_model.h5 để sử dụng cho các lần dự đoán sau.

Kết quả huấn luyện (loss, accuracy, precision, recall, f1-score) được lưu vào file training\_history.csv.

Các biểu đồ loss và các chỉ số khác theo epoch được lưu vào thư mục training\_plots.

**2. Phương pháp tăng độ phân giải của ảnh**

Mô hình sử dụng mạng CNN để tái tạo ảnh HR từ ảnh LR. Phương pháp bao gồm các bước sau:

**A. Tiền xử lý ảnh đầu vào**

Ảnh đầu vào (LR) được resize về kích thước chuẩn (128, 128) và chuẩn hóa giá trị pixel về [0, 1].

Định dạng đầu vào: (batch\_size, 128, 128, 3).

**B. Quá trình tái tạo ảnh**

**Trích xuất đặc trưng (Feature Extraction):**

Các tầng Conv2D và BatchNormalization trích xuất đặc trưng từ ảnh LR.

Hàm kích hoạt ReLU giúp mô hình học các đặc trưng phức tạp và phi tuyến tính.

**Giảm kích thước và học đặc trưng sâu (Downsampling):**

Sử dụng tầng MaxPooling2D để giảm kích thước không gian ảnh, đồng thời giữ lại thông tin quan trọng.

**Tăng kích thước và tái tạo (Upsampling):**

Các tầng UpSampling2D kết hợp với Conv2D tăng kích thước không gian ảnh từng bước, tái tạo chi tiết cho ảnh HR.

Tầng cuối cùng sử dụng hàm kích hoạt sigmoid để giá trị pixel của ảnh HR đầu ra nằm trong khoảng [0, 1].

**C. Dự đoán ảnh HR**

**Dự đoán:**

Đầu vào: Một ảnh LR hoặc tập ảnh LR.

Đầu ra: Ảnh HR được mô hình dự đoán.

Sử dụng hàm model.predict() để nhận ảnh HR đầu ra.

**Hậu xử lý ảnh:**

Chuyển đổi giá trị pixel từ [0, 1] về [0, 255] nếu cần hiển thị hoặc lưu ảnh.

Có thể lưu ảnh dự đoán bằng tf.keras.preprocessing.image.save\_img().

**D. Đánh giá chất lượng tái tạo**

**Chỉ số đánh giá:**

So sánh ảnh HR dự đoán với ảnh HR thực trên các chỉ số:

**MSE (Loss):** Đánh giá độ sai khác giữa hai ảnh.

**Precision, Recall, F1 Score:** Đánh giá khả năng tái tạo chi tiết trong ảnh.

**Trực quan hóa:**

Hiển thị ảnh LR, HR thực và HR dự đoán để so sánh chất lượng.

**Kết quả huấn luyện mô hình**

**A graph of different colored lines

Description automatically generated with medium confidence**

**A screenshot of a computer

Description automatically generated**

Hình 2.2. Kết quả huấn luyện mô hình

## **CHƯƠNG 3: KẾT QUẢ THỰC NGHIỆM**

## **3.1. Demo ứng dụng**

A screenshot of a computer

Description automatically generated

Hình 3.1. Giao diện chính của ứng dụng

**Mô tả giao diện:**

Ứng dụng được xây dựng bằng thư viện tkinter và tích hợp thêm các công cụ xử lý hình ảnh từ Pillow và mô hình học sâu từ TensorFlow. Giao diện được thiết kế với hai phần chính: **khung chức năng bên trái** và **khung hiển thị ảnh bên phải**, cùng với các tiêu đề và nút điều khiển.

**A. Khung chức năng**

Khung này bao gồm các nút chức năng chia thành hai phần chính: **mô hình & chức năng** và **quản lý ảnh**.

**Mô hình & Chức năng:**

**Nút “Tải Mô Hình Đã Huấn Luyện”**

Mở hộp thoại để tải mô hình CNN đã huấn luyện.

Chức năng: Sử dụng mô hình đã tải để cải thiện chất lượng ảnh.

**Nút “Tải Lịch Sử Huấn Luyện”**

Chọn tệp .csv chứa lịch sử huấn luyện của mô hình.

**Nút “Vẽ Biểu Đồ Lịch Sử”**

Hiển thị biểu đồ loss và accuracy trong quá trình huấn luyện.

**Quản lý ảnh:**

**Nút “Tải Ảnh”**

Mở hộp thoại để người dùng chọn một ảnh từ máy tính.

Ảnh sẽ được xử lý trước bằng làm mờ (Gaussian Blur).

**Nút “Tăng Độ Nét & Độ Phân Giải”**

Tăng độ nét và áp dụng mô hình CNN để tăng chất lượng ảnh.

**Nút “Lưu Ảnh”**

Lưu ảnh đã xử lý vào máy tính.

**B. Khung hiển thị ảnh (Bên phải)**

**Khung “Ảnh Đầu Vào”**

**Hiển thị ảnh gốc được người dùng tải lên** (sau khi làm mờ).

**Nội dung phụ:**

Hiển thị độ phân giải của ảnh đầu vào (gốc).

**Khung “Ảnh Đã Xử Lý”**

**Hiển thị ảnh đã qua xử lý** (tăng chất lượng hoặc sắc nét).

**Nội dung phụ:**

Hiển thị độ phân giải của ảnh sau khi tăng cường.

## **3.2. Cách thức hoạt động của ứng dụng**

**A. Chuẩn bị và cấu hình ứng dụng**

**Môi trường:** Ứng dụng sử dụng tkinter làm giao diện, Pillow để xử lý ảnh, và TensorFlow cho mô hình học sâu (CNN).

**Mô hình CNN (Residual Network):**

Mô hình CNN được thiết kế để tăng độ phân giải bằng cách:

Sử dụng các **residual block** để giữ thông tin quan trọng trong ảnh.

**UpSampling2D** giúp phóng to kích thước ảnh đầu ra.

**B. Các chức năng chính của ứng dụng**

**Tải mô hình đã huấn luyện**

Người dùng nhấn **"Tải Mô Hình Đã Huấn Luyện"** để chọn mô hình .h5 đã được huấn luyện trước.

**Hoạt động:**

Mô hình được tải bằng tf.keras.models.load\_model.

Mô hình được lưu trữ trong biến self.model và sẵn sàng để xử lý ảnh.

**Tải ảnh đầu vào**

Người dùng nhấn **"Tải Ảnh"** để chọn ảnh từ máy tính.

**Hoạt động:**

Ảnh gốc được đọc bằng thư viện Pillow.

Ứng dụng giảm chất lượng ảnh bằng cách làm mờ với GaussianBlur.

Ảnh làm mờ được hiển thị trong **khung Ảnh Đầu Vào**.

**Tăng độ phân giải và độ nét ảnh**

Người dùng nhấn **"Tăng Độ Nét & Độ Phân Giải"** để cải thiện ảnh.

**Hoạt động:**

Nếu người dùng không tải mô hình, ứng dụng sử dụng Pillow để tăng độ nét bằng ImageEnhance.Sharpness và áp dụng bộ lọc DETAIL.

Nếu mô hình đã được tải:

Ảnh đầu vào được chuyển đổi thành tensor (numpy array) và chuẩn hóa.

Ứng dụng sử dụng mô hình CNN để dự đoán ảnh chất lượng cao hơn.

Ảnh kết quả được hiển thị trong **khung Ảnh Đã Xử Lý**.

**Hiển thị và lưu ảnh**

Ứng dụng hiển thị cả hai ảnh: ảnh đầu vào và ảnh đã qua xử lý.

Độ phân giải ảnh được hiển thị bên dưới từng khung hình.

Người dùng có thể nhấn **"Lưu Ảnh"** để lưu ảnh đã xử lý dưới dạng tệp .png.

**C. Các chức năng phụ trợ**

**Tải lịch sử huấn luyện**

Người dùng nhấn **"Tải Lịch Sử Huấn Luyện"** để chọn tệp .csv chứa thông tin loss và accuracy của quá trình huấn luyện.

**Hoạt động:**

Tệp .csv được đọc bằng pandas và lưu trong biến self.history\_data.

**Vẽ biểu đồ lịch sử huấn luyện**

Sau khi tải lịch sử, người dùng nhấn **"Vẽ Biểu Đồ Lịch Sử"**.

**Hoạt động:**

Ứng dụng sử dụng matplotlib để vẽ biểu đồ:

Loss (mất mát) qua các epoch.

Accuracy (độ chính xác) qua các epoch.

Biểu đồ được hiển thị trong cửa sổ mới.

**Quy trình xử lý ảnh**

**Chuẩn bị ảnh đầu vào:**

Ảnh được làm mờ để giảm chất lượng giả lập (nếu người dùng không tải mô hình).

Hoặc ảnh đầu vào được resize thành kích thước phù hợp với mô hình CNN.

**Xử lý bằng CNN:**

Ảnh đầu vào được đưa qua các tầng tích chập (Convolutional Layers).

Các **Residual Block** giúp bảo toàn chi tiết ảnh gốc.

Tầng **UpSampling2D** tăng kích thước ảnh đầu ra.

**Tăng cường chi tiết ảnh:**

Nếu không sử dụng mô hình, ảnh được cải thiện độ sắc nét bằng Pillow.

**Hiển thị ảnh đã xử lý:**

Ứng dụng hiển thị ảnh kết quả kèm độ phân giải.

## **3.3. Kết quả thực nghiệm ứng dụng**

A screenshot of a computer

Description automatically generated

Hình 3.2. Kết quả thực nghiệm ứng dụng

**Mô tả kết quả:**

Kết quả thực nghiệm được đánh giá dựa trên khả năng của ứng dụng trong việc tăng độ phân giải và cải thiện độ sắc nét của ảnh. Các kết quả được ghi nhận qua hai trường hợp chính:

**(1) Xử lý ảnh bằng các phương pháp cơ bản**

**(2) Xử lý ảnh bằng mô hình CNN**.

**1. Trường hợp 1: Xử lý ảnh bằng các phương pháp cơ bản (không dùng mô hình CNN)**

**Ảnh đầu vào:**

**Loại ảnh:** Ảnh phong cảnh và chân dung, độ phân giải thấp (dưới 128x128).

**Quy trình xử lý:**

Ảnh đầu vào được làm mờ bằng bộ lọc Gaussian Blur (giảm chất lượng).

Ảnh được tăng cường bằng bộ lọc **ImageEnhance.Sharpness** và xử lý chi tiết bằng bộ lọc DETAIL.

**Kết quả:**

**Cải thiện độ sắc nét:**

Ảnh rõ nét hơn so với ảnh đầu vào mờ.

Một số chi tiết nhỏ được làm nổi bật, như đường nét hoặc kết cấu của ảnh.

**Độ phân giải không thay đổi:**

Ảnh không được phóng to hoặc cải thiện độ phân giải thực tế, chỉ làm rõ nét hơn.

**Ưu điểm:**

Xử lý nhanh, không yêu cầu mô hình phức tạp.

Phù hợp với các ảnh nhỏ hoặc yêu cầu làm rõ chi tiết nhanh chóng.

**Hạn chế:**

Không cải thiện độ phân giải thực sự.

Chất lượng ảnh chỉ cải thiện nhẹ, phụ thuộc vào bộ lọc.

**2. Trường hợp 2: Xử lý ảnh bằng mô hình CNN (khi tải mô hình huấn luyện trước)**

**Ảnh đầu vào:**

**Loại ảnh:** Ảnh chất lượng thấp (64x64, 128x128) với chi tiết bị mờ.

**Quy trình xử lý:**

Ảnh được resize về kích thước cố định (128x128).

Ảnh được xử lý qua mô hình CNN đã huấn luyện trên tập dữ liệu ảnh chất lượng cao (tăng độ phân giải gấp 2 lần hoặc hơn).

**Kết quả:**

**Cải thiện độ phân giải:**

Ảnh đầu ra có độ phân giải cao hơn, ví dụ: từ 128x128 lên 256x256.

Các chi tiết như cạnh, kết cấu và màu sắc được tái tạo rõ ràng.

**Cải thiện độ sắc nét:**

So với ảnh gốc, ảnh đã xử lý rõ ràng và sắc nét hơn.

**Độ mượt của ảnh:**

Mô hình CNN tái tạo ảnh mềm mại, không gây nhiễu hạt như các phương pháp thông thường.

**Ưu điểm:**

Cải thiện cả độ phân giải lẫn độ sắc nét.

Kết quả gần với ảnh gốc chất lượng cao (tùy thuộc vào mức độ huấn luyện mô hình).

**Hạn chế:**

Yêu cầu mô hình đã được huấn luyện tốt trên tập dữ liệu tương tự.

Thời gian xử lý ảnh lâu hơn so với các phương pháp cơ bản.

## **KẾT LUẬN**

**Thành tựu đạt được:**

Mô hình CNN được xây dựng trong bài tập lớn đã hoàn thành nhiệm vụ chính là tăng độ phân giải ảnh. Các kết quả huấn luyện cho thấy mô hình có khả năng học và tái tạo các đặc trưng quan trọng từ ảnh gốc thông qua việc giảm loss một cách đáng kể qua các epoch.

Giá trị các chỉ số như Loss (0.0818) và Recall (0.4986) phản ánh rằng mô hình có hiệu quả nhất định trong việc tái tạo các chi tiết của ảnh đầu ra, dù còn những hạn chế cần cải thiện.

Việc triển khai mô hình đã minh họa rõ ràng cách một kiến trúc CNN cơ bản có thể được áp dụng vào bài toán xử lý hình ảnh thực tế.

**Hạn chế:**

Độ chính xác tổng thể thấp: Chỉ số Accuracy (0.3616) và F1 Score (0.3851) chưa cao, cho thấy mô hình còn gặp khó khăn trong việc phân biệt chính xác các chi tiết phức tạp của ảnh.

Sự dao động của Precision và F1 Score: Những biến động này phản ánh sự thiếu ổn định trong quá trình tối ưu hóa mô hình, có thể do kiến trúc mạng chưa đủ phức tạp hoặc dữ liệu huấn luyện chưa đủ phong phú.

Dữ liệu huấn luyện: Số lượng dữ liệu và tính đa dạng của tập huấn luyện có thể còn hạn chế, dẫn đến hiện tượng mô hình học chưa đủ sâu các đặc trưng quan trọng.

**Hướng cải thiện trong tương lai:**

Để cải thiện hiệu năng và áp dụng tốt hơn vào thực tế, cần cân nhắc các hướng sau:

Cải thiện mô hình CNN:

Thử nghiệm các kiến trúc mạng tiên tiến hơn như ResNet, UNet, hoặc EfficientNet để tăng khả năng học và tái tạo các chi tiết phức tạp từ ảnh.

Kết hợp các kỹ thuật như attention mechanism hoặc dilated convolutions để tăng cường khả năng xử lý các đặc trưng quan trọng của ảnh.

Tăng cường dữ liệu (Data Augmentation):

Áp dụng các kỹ thuật như xoay, lật, phóng to/thu nhỏ, thêm nhiễu để làm phong phú tập dữ liệu huấn luyện, giúp mô hình học được nhiều tình huống hơn.

Điều chỉnh quá trình huấn luyện:

Sử dụng learning rate scheduler để tự động giảm tốc độ học khi loss không giảm.

Áp dụng các kỹ thuật regularization (Dropout, Batch Normalization) để giảm overfitting, từ đó làm tăng hiệu suất tổng thể.

Đánh giá sâu hơn:

Thực hiện thêm các thử nghiệm với tập dữ liệu lớn hơn hoặc các ảnh có độ phức tạp khác nhau để kiểm tra hiệu năng của mô hình.

Tập trung vào các chỉ số cụ thể như F1 Score và Precision để đánh giá chi tiết khả năng cân bằng giữa các dự đoán đúng và sai.

**Tổng kết:**

Mặc dù mô hình CNN trong bài tập lớn đã đạt được những kết quả khả quan trong bài toán tăng độ phân giải ảnh, nhưng vẫn còn nhiều không gian để cải thiện. Các hướng nghiên cứu mở rộng, bao gồm cải tiến mô hình, tăng cường dữ liệu, và tối ưu hóa quy trình huấn luyện, sẽ giúp nâng cao hơn nữa hiệu năng.

## **TÀI LIỆU THAM KHẢO**

[1] *Lương Mạnh Bá, Nguyễn Thanh Thuỷ.* Nhập môn xử lý ảnh. *Nhà xuất bản Khoa học và Kỹ thuật, 2003. – Tham khảo ngày 20-11-2024*

[2] *Võ Đức Khánh, Hoàng Văn Kiếm.* Giáo trình xử lý ảnh số. *Nhà xuất bản Đại học Quốc gia Thành phố Hồ Chí Minh. – Tham khảo ngày 28-11-2024*

[3] *Nguyễn Kim Sách.* Xử lý ảnh và video số. *Nhà xuất bản Khoa học và Kỹ thuật, 1997.*

*– Tham khảo ngày 02-2-2024*

[4] *Nguyễn Quốc Trung.* Xử lý tín hiệu và lọc số. *Nhà xuất bản Khoa học và Kỹ thuật, 2004 – Tham khảo ngày 05-2-2024*