HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG
KHOA CÔNG NGHỆ THÔNG TIN 1

_____ o0o _____

BÁO CÁO BÀI TẬP LỚN

BỘ MÔN: THỰC TẬP CƠ SỞ

Tên đề tài: Xây dựng hệ thống phân loại và xác thực ảnh sử dụng kết hợp mô hình PRNU, phân tích nhiễu và ELA

MÃ LỚP : 49
Số thứ tự nhóm: 01

Nguyễn Trường Giang **MSSV: B22DCKH034**
Đỗ Chí Chương **MSSV: B22DCKH015**
Cấn Đức Khôi **MSSV: B22DCKH069**

Giảng viên hướng dẫn: Ths. Vũ Hoài Thư

HÀ NỘI, 2025

LỜI CẢM ƠN

Để hoàn thành bài tập lớn, trước hết nhóm chúng em xin bày tỏ lời cảm ơn sâu sắc với cô Vũ Hoài Thư, người đã hướng dẫn chúng em vượt qua Bài tập lớn môn Thực tập cơ sở này. Nhờ có sự nghiêm khắc mà cũng đầy lòng nhiệt tình, tận tâm của cô mà chúng em đã hiểu biết hơn cách để xử lý ảnh, cách xây dựng mô hình trí tuệ nhân tạo đúng phương pháp đồng thời rút ra nhiều bài học quý báu trong việc team-work, nghiên cứu tài liệu và làm báo cáo. Em cũng xin cảm ơn gia đình đã tạo điều kiện tốt nhất để em tập trung nghiên cứu. Cuối cùng, em xin ghi nhận nỗ lực của chính bản thân nhóm chúng em đã kiên trì và quyết tâm để đạt được kết quả này.

MỤC LỤC

DANH MỤC HÌNH VẼ

DANH MỤC BẢNG BIỂU

DANH MỤC THUẬT NGỮ VÀ TỪ VIẾT TẮT

Viết tắt	Tên tiếng Anh	Tên tiếng Việt
AI	Artificial Intelligence	Trí tuệ nhân tạo
API	Application Programming Interface	Giao diện lập trình ứng dụng
AUC	Area Under the ROC Curve	Diện tích dưới đường cong ROC
BTL		Bài tập lớn
CNN	Convolutional Neural Network	Mạng nơ-ron tích chập (Mạng thần kinh tích chập)
ELA	Error Level Analysis	Phân tích mức độ lỗi
FFT	Fast Fourier Transform	Biến đổi Fourier nhanh
GAN	Generative Adversarial Network	Mạng đối nghịch tạo sinh
GPU	Graphics Processing Unit	Đơn vị xử lý đồ họa
HTML	HyperText Markup Language	Ngôn ngữ đánh dấu siêu văn bản
JPEG	Joint Photographic Experts Group	Nhóm chuyên gia ảnh phối hợp
PNG	Portable Network Graphics	Đồ họa mạng di động
PRNU	Photo Response Non-Uniformity	Đặc trưng không đồng nhất phản hồi ảnh
ReLU	Rectified Linear Unit	Đơn vị tuyến tính chỉnh lưu

CHƯƠNG 1. GIỚI THIỆU ĐỀ TÀI

1.1 Đặt vấn đề

Trong những năm gần đây, đặc biệt từ đầu thập niên 2020, cho thấy một sự thay đổi mang tính cách mạng trong lĩnh vực sáng tạo và phổ biến nội dung số, với sự bùng nổ của hình ảnh tạo bởi trí tuệ nhân tạo (AI) và sự gia tăng mạnh mẽ của các kỹ thuật chỉnh sửa ảnh ngày càng tinh vi.

Sự phát triển nhanh chóng của các kiến trúc học sâu như Mạng đối nghịch tạo sinh (GANs) và Mô hình Khuếch tán (Diffusion Models) đã cho phép tạo ra hàng chục triệu hình ảnh AI mỗi ngày với độ chân thực cao (Nguồn: TechReport, 2024-2025).

Đồng thời, việc hàng tỷ bức ảnh được chụp và chia sẻ hàng ngày, phần lớn từ thiết bị di động (Nguồn: Presets.io, Image Retouching Lab), cùng với sự dễ dàng tiếp cận các công cụ chỉnh sửa mạnh mẽ, đã khiến việc can thiệp và thay đổi nội dung hình ảnh trở nên phổ biến hơn bao giờ hết. Ước tính có tới hơn 10 tỷ bức ảnh được chỉnh sửa hàng năm (Nguồn: Market.us).

Sự gia tăng nhanh chóng của ảnh AI và ảnh chỉnh sửa đã dẫn đến một vấn đề nghiêm trọng, là sự suy giảm tính xác thực của những bức ảnh thật được chụp ra. Khi ranh giới giữa hình ảnh thật và hình ảnh được tạo ra hoặc bị can thiệp ngày càng trở nên khó phân định, khả năng lạm dụng các công nghệ này cho mục đích xấu cũng ngày càng tăng lên.

Cụ thể, hình ảnh AI và ảnh chỉnh sửa tinh vi đang trở thành công cụ đắc lực cho việc phát tán tin giả và thông tin sai lệch trên quy mô lớn. Chúng có thể được sử dụng để tạo ra bằng chứng giả mạo, bóp méo sự kiện, hoặc xây dựng những câu chuyện hoàn toàn không có thật.

Hơn nữa, nguy cơ bị lợi dụng cho các hoạt động lừa đảo cũng ngày càng hiện hữu. Công nghệ deepfake, một sản phẩm của AI tạo sinh, có khả năng tạo ra các video và hình ảnh giả mạo người thật một cách thuyết phục, dẫn đến các vụ việc đánh cắp danh tính, tống tiền, hoặc lừa đảo tài chính với thiệt hại lớn (Nguồn: Surfshark, 2025; Forbes, 2024). Các hình ảnh được tạo ra hoặc chỉnh sửa cũng có thể được sử dụng để tạo hồ sơ giả mạo, phục vụ cho các hành vi phạm tội trực tuyến hoặc các chiến dịch gây ảnh hưởng độc hại.

Bài toán đặt ra ở đây là làm thế nào để đối phó với sự gia tăng không kiểm soát của các loại hình ảnh này và giảm thiểu những tác động tiêu cực mà chúng gây ra. Nếu có những phương pháp và công cụ hiệu quả để xác định tính xác thực và phát

hiện sự can thiệp vào hình ảnh, sẽ mang lại lợi ích to lớn. Đối với cá nhân, điều này giúp họ tự bảo vệ mình khỏi thông tin sai lệch, lừa đảo và các hình thức xâm phạm danh tính. Đối với các tổ chức như cơ quan truyền thông, cơ quan thực thi pháp luật, lợi ích nằm ở việc củng cố độ tin cậy của thông tin, hỗ trợ điều tra và đảm bảo tính công bằng. Ở quy mô xã hội, việc giải quyết bài toán này góp phần xây dựng một không gian số lành mạnh hơn, tăng cường niềm tin vào thông tin.

Chính vì những lẽ đó, việc nghiên cứu và phát triển các giải pháp tiên tiến để giải quyết bài toán xác thực hình ảnh trong bối cảnh hiện nay là vô cùng cấp thiết và có ý nghĩa quan trọng.

1.2 Mục tiêu và phạm vi đề tài

Trước bối cảnh phức tạp của sự gia tăng ảnh tạo bởi trí tuệ nhân tạo (AI) và ảnh qua chỉnh sửa như đã phân tích trong phần 1.1, người ta đã đưa ra nhiều giải pháp nhằm xác thực và phân loại hình ảnh. Tổng quan các hướng tiếp cận chính đang được áp dụng: Một là, tập trung vào việc phân tích các dấu vết vật lý do cảm biến máy ảnh để lại, trong đó đặc trưng PRNU là một ví dụ điển hình, giúp truy xuất nguồn gốc thiết bị chụp. Hai là, phân tích đặc điểm thống kê hoặc cấu trúc của ảnh, như phân tích mức độ lỗi (Error Level Analysis - ELA) nhằm phát hiện sự không nhất quán trong mức độ nén JPEG, được sử dụng để tìm kiếm dấu hiệu can thiệp hoặc các đặc điểm bất thường của ảnh AI. Thứ ba, các phương pháp học sâu, chủ yếu dựa trên mạng nơ-ron tích chập (CNNs), ngày càng phổ biến nhờ khả năng tự động học các đặc trưng phân biệt từ các bộ dữ liệu lớn. Tổng quan thì mỗi hướng tiếp cận riêng biệt trên đều có những ưu điểm và hạn chế nhất định. Các phương pháp dựa trên PRNU cho thấy tiềm năng trong việc xác định nguồn gốc ảnh một cách tin cậy, tuy nhiên, hiệu quả có thể bị suy giảm nếu ảnh đã trải qua các bước xử lý mạnh hoặc thay đổi kích thước. ELA là một phương pháp hữu ích để tìm ra một số kiểu chỉnh sửa ảnh, nhưng đôi khi nó không đủ khả năng phát hiện những thay đổi nhỏ hoặc khi ảnh được lưu lại với chất lượng rất tốt. Còn xét về các mô hình học sâu hiện đại, tuy có khả năng phân loại tốt, chúng thường yêu cầu khối lượng lớn dữ liệu gán nhãn để huấn luyện, và sẽ thiếu khả năng phân loại nếu bộ dữ liệu không đủ tốt. Một hạn chế chung của từng giải pháp là việc thường chỉ tập trung giải quyết một khía cạnh đơn lẻ của bài toán, ví dụ như chỉ chuyên biệt phát hiện ảnh AI hoặc chỉ phát hiện một số loại chỉnh sửa cụ thể. Do đó, kết luận đưa ra thường mang tính nhị phân (ví dụ: "thật" hoặc "giả", "đã sửa" hoặc "chưa sửa"), chưa cung cấp được một cái nhìn toàn diện và chi tiết về trạng thái thực sự của một bức ảnh, chẳng hạn như việc phân biệt một bức ảnh gốc chụp từ máy ảnh với một bức ảnh cũng từ máy ảnh nhưng đã qua chỉnh sửa.

Từ những phân tích và đánh giá về các hạn chế hiện tại, đề tài này chúng em hướng tới giải quyết vấn đề bằng cách xây dựng một hệ thống phân loại hình ảnh đa chiều và chi tiết hơn. Mục tiêu chính của đề tài là phát triển một phần mềm ứng dụng có các chức năng chính sau:

(i) Phân biệt nguồn gốc ảnh bằng cách phát triển một mô hình học máy, cụ thể là mạng nơ-ron tích chập, có khả năng xác định liệu một bức ảnh có dấu vết của cảm biến máy ảnh thực (ảnh thật) hay được tạo ra hoàn toàn bởi thuật toán AI. Chức năng này sẽ dựa trên việc trích xuất và phân tích các đặc trưng liên quan đến nhiễu ảnh (như PRNU) và các đặc điểm thống kê của nhiễu dư.

(ii) Phát hiện dấu hiệu chỉnh sửa ảnh bằng cách xây dựng một mô hình mạng nơ-ron tích chập thứ hai, sử dụng các đặc trưng từ phân tích mức độ Lỗi (ELA) làm đầu vào, để nhận diện các dấu hiệu can thiệp hoặc chỉnh sửa trên hình ảnh, bất kể nguồn gốc ban đầu của nó là ảnh thật hay ảnh AI.

(iii) Cung cấp kết luận phân loại tổng hợp bằng cách thiết kế một cơ chế logic để tích hợp kết quả từ hai mô hình trên, nhằm đưa ra một trong các kết luận phân loại chi tiết hơn về trạng thái của ảnh, ví dụ: "ảnh gốc từ máy ảnh", "ảnh từ máy ảnh đã qua chỉnh sửa", hoặc "ảnh tạo bởi AI".

(iv) Minh họa qua giao diện người dùng: Phát triển một ứng dụng web cho phép người dùng tải ảnh lên và nhận kết quả phân loại từ hệ thống, nhằm kiểm chứng và trình diễn khả năng của giải pháp.

Phạm vi của đề tài này chúng em tập trung vào việc xử lý ảnh tĩnh ở các định dạng phổ biến (ví dụ: JPEG, PNG) hoặc ảnh gốc máy ảnh (.TIFF). Các mô hình sẽ được huấn luyện và đánh giá trên các bộ dữ liệu chứa ảnh thật, ảnh tạo bởi các công nghệ AI hiện hành (như GANs, Diffusion Models), và ảnh đã qua các dạng chỉnh sửa có khả năng để lại dấu vết mà ELA có thể phát hiện. Đề tài không đặt mục tiêu xác định chi tiết từng loại thao tác chỉnh sửa cụ thể mà tập trung vào việc nhận biết sự tồn tại của hành vi chỉnh sửa. Việc phát triển các mô hình sẽ sử dụng ngôn ngữ Python và các thư viện học máy như TensorFlow hoặc Keras.

1.3 Định hướng giải pháp

Từ các mục tiêu và phạm vi đã xác định ở phần 1.2, đề tài này đề xuất một định hướng giải pháp kết hợp nhiều kỹ thuật và công nghệ để giải quyết bài toán phân loại và xác thực hình ảnh một cách toàn diện hơn là riêng lẻ từng phương pháp:

(i) Về mặt công nghệ và phương pháp, đề tài sẽ tập trung vào học máy, cụ thể là ứng dụng mạng nơ-ron tích chập (CNN). Các mạng nơ-ron tích chập được lựa chọn do khả năng vượt trội trong việc tự động trích xuất và học các đặc trưng phức

tạp từ dữ liệu hình ảnh, đã được chứng minh hiệu quả trong nhiều bài toán thị giác máy tính. Song song đó, đề tài cũng khai thác các kỹ thuật trích xuất đặc trưng ảnh chuyên biệt bao gồm phân tích đặc trưng không đồng nhất phản hồi ảnh (PRNU) và các đặc điểm của nhiễu dư (residual noise) để nhận diện nguồn gốc ảnh, cùng với kỹ thuật phân tích mức độ lỗi (ELA) để phát hiện các dấu hiệu can thiệp, chỉnh sửa. Lý do lựa chọn các đặc trưng này là vì chúng cung cấp những manh mối quan trọng về lịch sử và tính toàn vẹn của ảnh mà các đặc trưng học được thuần túy từ nội dung có thể bỏ qua.

(ii) Giải pháp được đề xuất bao gồm việc xây dựng một hệ thống gồm hai mô hình mạng nơ-ron tích chập thành phần hoạt động song song. Mô hình thứ nhất được huấn luyện để phân biệt giữa ảnh có nguồn gốc từ cảm biến máy ảnh thực và ảnh do AI tạo ra, dựa trên đầu vào là các đặc trưng PRNU và nhiễu. Mô hình mạng nơ-ron tích chập thứ hai được thiết kế để phát hiện dấu hiệu chỉnh sửa trên ảnh, sử dụng đặc trưng từ phân tích mức độ lỗi làm đầu vào. Sau đó, kết quả dự đoán (dưới dạng xác suất hoặc nhãn lớp) từ hai mô hình này sẽ được tổng hợp thông qua một bộ quy tắc logic kết hợp. Bộ quy tắc này sẽ đưa ra một trong các kết luận tổng hợp về trạng thái của ảnh, ví dụ như "ảnh gốc từ máy ảnh", "ảnh từ máy ảnh đã qua chỉnh sửa", hoặc "ảnh tạo bởi AI", nhằm cung cấp một cái nhìn chi tiết và đa chiều hơn so với các phương pháp chỉ dựa trên một mô hình duy nhất.

(iii) Đóng góp chính của bài tập lớn này là việc đề xuất và triển khai một hệ thống phân loại ảnh kết hợp, tích hợp hiệu quả thông tin từ cả đặc trưng vật lý của cảm biến (PRNU/nhiễu) và đặc trưng do quá trình xử lý (phân tích mức độ lỗi) để nâng cao độ chính xác và độ tin cậy trong việc xác thực hình ảnh. Kết quả đạt được sẽ là một hệ thống phần mềm hoàn chỉnh bao gồm các mô hình mạng nơ-ron tích chập đã được huấn luyện, logic kết hợp quyết định, và một ứng dụng web minh họa cho phép người dùng tương tác và kiểm chứng khả năng của hệ thống trong việc phân loại ảnh thành các trạng thái chi tiết đã định nghĩa.

1.4 Bố cục bài tập lớn

Phần còn lại của báo cáo bài tập lớn này được chúng em trình bày như sau: Chương 2 sẽ tập trung vào cơ sở lý thuyết liên quan đến đề tài, giới thiệu các khái niệm về ảnh kỹ thuật số, công nghệ tạo ảnh AI, kỹ thuật chỉnh sửa ảnh và phân tích nguyên lý của các phương pháp cốt lõi như PRNU, ELA và phân tích nhiễu dư.

Chương 3 trình bày chi tiết về quá trình thực nghiệm và các kết quả đánh giá mô hình. Nội dung bao gồm mô tả bộ dữ liệu, quy trình trích xuất đặc trưng, kiến trúc các mô hình CNN (đa đầu vào cho PRNU/nhiễu dư và VGG16 cho ELA), quy trình huấn luyện, đánh giá hiệu năng và xây dựng ứng dụng web phân tích ảnh.

Chương 4 sẽ đưa ra kết luận và các định hướng phát triển trong tương lai, tóm tắt những đóng góp chính, phân tích những gì đã làm được, hạn chế còn tồn tại và đề xuất các hướng phát triển tiềm năng.

CHƯƠNG 2. CƠ SỞ LÝ THUYẾT

2.1 Ảnh tạo bởi trí tuệ nhân tạo và ảnh chỉnh sửa kỹ thuật số

2.1.1 Ảnh tạo bởi trí tuệ nhân tạo

Ảnh tạo bởi trí tuệ nhân tạo là những hình ảnh được tạo ra hoàn toàn hoặc phần lớn bởi các thuật toán học máy. Sự phát triển vượt bậc của các mô hình học sâu đã cho phép tạo ra những hình ảnh AI với độ chân thực ngày càng cao, đôi khi khó phân biệt với ảnh thật bằng mắt thường. Hai kiến trúc mô hình chính đóng góp vào sự bùng nổ này là:

Mạng đối nghịch tạo sinh (GANs):

Được giới thiệu bởi Goodfellow và cộng sự vào năm 2014, GANs bao gồm hai mạng nơ-ron cạnh tranh với nhau: một mạng sinh (Generator) cố gắng tạo ra dữ liệu giả (hình ảnh) sao cho giống thật nhất, và một mạng phân biệt (Discriminator) cố gắng phân biệt giữa dữ liệu thật và dữ liệu giả do mạng sinh tạo ra. Qua quá trình huấn luyện đối nghịch này, mạng sinh ngày càng tạo ra những hình ảnh chất lượng cao hơn. Các biến thể của GANs như StyleGAN, BigGAN đã đạt được những kết quả ấn tượng trong việc tạo ra ảnh chân dung, cảnh vật, và các đối tượng khác.

Mô hình khuếch tán (diffusion models):

Đây là một lớp mô hình tạo sinh mới nổi gần đây và nhanh chóng đạt được hiệu suất vượt trội, thậm chí hơn cả GANs trong một số tác vụ tạo ảnh chất lượng cao. Nguyên lý cơ bản của mô hình khuếch tán bao gồm hai quá trình: (1) quá trình thuận (forward process), trong đó nhiễu gaussian được thêm dần vào dữ liệu huấn luyện qua nhiều bước cho đến khi dữ liệu trở thành nhiễu thuần túy; và (2) quá trình ngược (reverse process), trong đó một mạng nơ-ron được huấn luyện để đảo ngược quá trình thêm nhiễu này, tức là học cách loại bỏ nhiễu từng bước một để tái tạo lại dữ liệu gốc từ nhiễu. Sau khi huấn luyện, mô hình có thể tạo ra một hình ảnh mới bằng cách bắt đầu từ một mẫu nhiễu ngẫu nhiên và thực hiện quá trình khử nhiễu ngược. Các mô hình nổi tiếng dựa trên kiến trúc này bao gồm DALL-E 2, Imagen, và Stable Diffusion.

2.1.2 Ảnh chỉnh sửa kỹ thuật số

Ảnh chỉnh sửa kỹ thuật số là một ảnh kỹ thuật số gốc đã trải qua các quá trình can thiệp, thay đổi bởi phần mềm máy tính nhằm mục đích thay đổi nội dung, hình thức, hoặc ý nghĩa của hình ảnh ban đầu. Quá trình chỉnh sửa có thể bao gồm một loạt các thao tác từ đơn giản đến phức tạp, được thực hiện bởi người dùng hoặc bởi các thuật toán tự động.

Về bản chất, một ảnh chỉnh sửa kỹ thuật số vẫn là một tập hợp các điểm ảnh được biểu diễn dưới dạng ma trận số, tương tự như một ảnh kỹ thuật số gốc [Gonzalez & Woods, 2018]. Tuy nhiên, các giá trị pixel hoặc cấu trúc của các pixel này đã bị thay đổi so với trạng thái ban đầu. Các kỹ thuật chỉnh sửa có thể tác động đến nhiều khía cạnh của ảnh:

(i) **Chỉnh sửa toàn cục - global adjustments:** Thay đổi độ sáng, độ tương phản, cân bằng màu, độ bão hòa màu trên toàn bộ ảnh.

(ii) **Chỉnh sửa cục bộ - local adjustments:** Can thiệp vào các vùng cụ thể của ảnh, ví dụ như làm sáng một vùng tối, thay đổi màu sắc của một đối tượng, hoặc làm mờ hậu cảnh.

(iii) **Thao tác hình học - geometric transformations:** Thay đổi kích thước, xoay, lật, hoặc làm biến dạng hình ảnh.

(iv) **Ghép ảnh - image splicing/compositing:** Kết hợp các phần từ hai hay nhiều ảnh khác nhau để tạo ra một ảnh mới.

(v) **Xóa/thêm đối tượng - object removal/addition:** Loại bỏ các chi tiết không mong muốn hoặc thêm vào các đối tượng mới không có trong ảnh gốc.

(vi) **Làm mịn/tăng độ nét - smoothing/sharpening:** Cải thiện hoặc thay đổi kết cấu bề mặt của các đối tượng trong ảnh.

(vii) **Sử dụng bộ lọc - filters:** Áp dụng các hiệu ứng nghệ thuật hoặc các bộ lọc được thiết kế sẵn để thay đổi diện mạo tổng thể của ảnh.

Sự phổ biến của các phần mềm chỉnh sửa ảnh mạnh mẽ và dễ sử dụng như Adobe Photoshop, đồng thời các ứng dụng đó được phổ cập trên đa nền tảng, dễ dàng tiếp cận đã khiến việc chỉnh sửa ảnh kỹ thuật số trở thành một thao tác phổ biến trong nhiều lĩnh vực từ nhiếp ảnh chuyên nghiệp, thiết kế đồ họa, quảng cáo, truyền thông xã hội cho đến sử dụng cá nhân. Việc hiểu rõ bản chất của các thao tác chỉnh sửa và những dấu vết tiềm ẩn mà chúng có thể để lại là cơ sở quan trọng cho việc phát triển các phương pháp phát hiện ảnh đã qua chỉnh sửa.

2.2 Phương pháp phân tích dựa trên đặc trưng không đồng nhất phản hồi ảnh (Photo-Response Non-Uniformity - PRNU) và nhiễu dư ảnh

2.2.1 Đặc trưng nhiễu dư ảnh và PRNU

Nhiễu dư ảnh (residual noise):

Trong quá trình thu nhận và xử lý ảnh kỹ thuật số, nhiều loại nhiễu khác nhau có thể xuất hiện, bắt nguồn từ cảm biến ảnh, mạch điện tử, điều kiện môi trường, hoặc quá trình nén ảnh. Nhiễu dư ảnh được định nghĩa là phần tín hiệu còn lại

sau khi loại bỏ đi nội dung chính (cấu trúc, chi tiết) của bức ảnh. Thông thường, nhiễu dư được ước tính bằng cách lấy ảnh gốc trừ đi một phiên bản đã được làm trơn (denoised version) của nó. Quá trình làm trơn này có thể được thực hiện bằng nhiều bộ lọc khác nhau, ví dụ như bộ lọc Gaussian, bộ lọc Wiener, hoặc các thuật toán khử nhiễu tiên tiến hơn.

Phương trình cơ bản để trích xuất nhiễu dư W từ một ảnh I có thể biểu diễn:

$$W = I - F(I)$$

Trong đó, $F(I)$ là kết quả của việc áp dụng một hàm làm trơn (bộ lọc khử nhiễu) lên ảnh I. Nhiễu dư này chứa đựng các thành phần nhiễu ngẫu nhiên cũng như các mẫu nhiễu có hệ thống, bao gồm cả PRNU.

Đặc trưng không đồng nhất phản hồi ảnh:

Đặc trưng không đồng nhất phản hồi ảnh (Photo-Response Non-Uniformity) là một loại nhiễu có hệ thống, xuất hiện do sự không hoàn hảo trong quá trình sản xuất cảm biến ảnh. Mỗi pixel riêng lẻ trên cảm biến có một sự khác biệt nhỏ về độ nhạy sáng so với các pixel khác, dẫn đến việc chúng phản hồi hơi khác nhau khi cùng tiếp xúc với một lượng ánh sáng đồng nhất. Sự không đồng nhất này tạo ra một "vân tay" đặc trưng cho từng cảm biến máy ảnh cụ thể và có tính bền vững qua nhiều bức ảnh được chụp bởi cùng một thiết bị.

PRNU được coi là một trong những đặc trưng đáng tin cậy nhất để xác định nguồn gốc máy ảnh (camera source identification) và phát hiện giả mạo, vì nó rất khó để loại bỏ hoàn toàn hoặc giả mạo một cách hoàn hảo. PRNU của một máy ảnh thường được ước tính bằng cách lấy trung bình nhiễu dư từ nhiều bức ảnh được chụp bởi chính máy ảnh đó.

2.2.2 Phân tích miền tần số của nhiễu dư

Phân tích miền tần số của nhiễu dư là một kỹ thuật phát hiện các dấu hiệu giả mạo hoặc nguồn gốc nhân tạo của hình ảnh. Bởi vì các thành phần tần số khác nhau trong nhiễu dư của ảnh mang thông tin về cấu trúc và các biến đổi của tín hiệu đó khác nhau ở hai loại ảnh nên có thể sử dụng đó làm đặc điểm phân biệt.

Cụ thể, sự khác biệt trong quá trình hình thành ảnh thường dẫn đến những đặc điểm riêng biệt trong phổ tần số của nhiễu. Ảnh thật, được chụp bởi cảm biến vật lý, thường chứa các cấu trúc nhiễu và chi tiết nhỏ lẻ (ví dụ: hạt do cảm biến, cạnh sắc nét tự nhiên) tạo nên một phổ tần số phức tạp và đa dạng. Ngược lại, ảnh AI, do được tạo ra bởi các thuật toán, có thể có xu hướng bị "*làm mượt*" (thiếu các thành phần tần số cao đặc trưng của nhiễu tự nhiên) hoặc ngược lại, tạo ra các "*phổ tần*

số giả" (ví dụ: các mẫu lặp lại hoặc các đỉnh bất thường) do kiến trúc của mô hình sinh ảnh.

Quá trình phân tích thường bao gồm các bước sau:

(i) **Chuẩn bị đầu vào:**

Đầu vào cho bước này là nhiễu dư W, đã được trích xuất từ ảnh gốc theo quy trình mô tả trong mục 2.2.1. Nhiễu dư này, $W(x, y)$, là một ma trận hai chiều (ảnh xám) biểu diễn sự khác biệt giữa ảnh gốc và phiên bản đã được làm mịn của nó, chứa đựng các thông tin nhiễu cần phân tích.

(ii) **Biến đổi Fourier hai chiều:**

Để chuyển đổi thông tin từ miền không gian (pixel) sang miền tần số, áp dụng biến đổi Fourier nhanh hai chiều (FFT) lên ma trận nhiễu dư $W(x, y)$. Phép biến đổi này phân giải tín hiệu nhiễu thành tổng của các sóng hình sin và cosin ở các tần số không gian khác nhau. Kết quả của FFT, ký hiệu là $F_W(u, v)$, là một ma trận các số phức:

$$F_W(u, v) = \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} W(x, y) e^{-j2\pi\left(\frac{ux}{M} + \frac{vy}{N}\right)} \tag{2.1}$$

Trong đó M và N là kích thước của ảnh nhiễu dư, (x, y) là tọa độ trong miền không gian, và (u, v) là tọa độ trong miền tần số. Mỗi phần tử $F_W(u, v)$ biểu diễn biên độ và pha của thành phần tần số (u, v) trong tín hiệu nhiễu.

(iii) **Dịch chuyển tâm phổ:**

Kết quả ban đầu do FFT tạo ra thường đặt thông tin về "tần số không" – đại diện cho giá trị trung bình hay mức độ nền chung của nhiễu – ở một góc của biểu đồ phổ tần số. Cách sắp xếp này có thể khiến việc quan sát và nhận định các đặc điểm tần số trở nên chưa thực sự trực quan.

Để việc phân tích và trực quan hóa trở nên thuận tiện hơn, cần thực hiện một thao tác gọi là "dịch chuyển tâm phổ". Về cơ bản, thao tác này giống như việc sắp xếp lại các kết quả tần số: thành phần "tần số không" (vốn dĩ thể hiện các đặc tính chung nhất, ít thay đổi nhất của nhiễu) được đưa vào chính giữa của biểu đồ phổ. Các thành phần tần số cao hơn sẽ nằm xa tâm hơn.

Sau khi áp dụng bước dịch chuyển tâm phổ này, việc diễn giải biểu đồ phổ tần số trở nên rõ ràng hơn rất nhiều:

- Vùng chính giữa biểu đồ lúc này sẽ tập trung các thông tin về tần số thấp,

bao gồm cả "tần số không". Đây là những thông tin nền, thể hiện các biến thiên diễn ra chậm và từ từ trong cấu trúc nhiễu của ảnh.

- Càng di chuyển ra xa khỏi vùng trung tâm, tiến về phía rìa của biểu đồ, chúng ta sẽ gặp các tần số ngày càng cao hơn. Những vùng này đại diện cho các thay đổi nhanh, đột ngột hơn, và thường tương ứng với các chi tiết nhỏ, sắc nét hơn trong nhiễu.

(iv) **Tính toán phổ biên độ:**

Từ ma trận phức $F_W(u, v)$ đã được dịch chuyển tâm, phổ biên độ (hay còn gọi là phổ năng lượng) được tính toán. Phổ biên độ cho biết độ mạnh hay năng lượng của mỗi thành phần tần số, bỏ qua thông tin về pha. Nó được tính bằng cách lấy mô-đun của mỗi giá trị phức trong ma trận $F_W(u, v)$:

$$|F_W(u, v)| = \sqrt{\mathrm{Re}(F_W(u, v))^2 + \mathrm{Im}(F_W(u, v))^2} \tag{2.2}$$

Kết quả $|F_W(u, v)|$ là một ma trận các giá trị thực, không âm.

(v) **Chuẩn hóa logarit:**

Biên độ của các thành phần tần số có thể thay đổi trong một dải động rất lớn, với các thành phần tần số thấp thường có biên độ cao hơn nhiều so với các thành phần tần số cao. Để nén dải động này và làm cho các chi tiết ở vùng tần số cao có biên độ nhỏ hơn trở nên rõ ràng hơn khi trực quan hóa hoặc sử dụng làm đặc trưng, phép biến đổi logarit được áp dụng:

$$S_L(u, v) = \log(1 + |F_W(u, v)|) \tag{2.3}$$

Việc cộng thêm 1 vào $|F_W(u, v)|$ trước khi lấy logarit là để tránh tính $\log(0)$ trong trường hợp có biên độ bằng không. Kết quả $S_L(u, v)$ là phổ biên độ đã được biến đổi logarit.

(vi) **Chuẩn hóa giá trị:**

Cuối cùng, để đưa các giá trị của phổ biên độ logarit về một dải tiêu chuẩn, thường là từ 0 đến 1, phép chuẩn hóa được thực hiện. Một cách chuẩn hóa phổ biến là Min-Max scaling:

$$S_{norm}(u, v) = \frac{S_L(u, v) - \min(S_L)}{\max(S_L) - \min(S_L)} \tag{2.4}$$

Trong trường hợp $\max(S_L) = \min(S_L)$ (ví dụ như khi ảnh đầu vào hoàn toàn

đồng nhất), $S_{\text{norm}}(u, v)$ được gán bằng 0 để tránh lỗi chia cho không. Kết quả thu được sau bước này, $S_{\text{norm}}(u, v)$, là một hình ảnh hai chiều (ma trận) biểu diễn phổ tần số đã được xử lý của nhiễu dư. Mỗi pixel trong hình ảnh này đại diện cho năng lượng chuẩn hóa của một thành phần tần số cụ thể. Hình ảnh này chính là đặc trưng phổ tần số sẽ được sử dụng làm một trong các đầu vào cho mô hình mạng nơ-ron tích chập trong đề tài.

Phân tích các đặc điểm từ đặc trưng phổ tần số:

Từ hình ảnh phổ tần số $S_{\text{norm}}(u, v)$, các mô hình học máy có thể tìm kiếm các dấu hiệu sau để phân biệt ảnh thật và ảnh AI:

(i) **Phân bố năng lượng ở các dải tần số:**

Ảnh AI thường có xu hướng thiếu năng lượng ở các dải tần số cao (do bị làm mịn quá mức trong quá trình tạo sinh) hoặc ngược lại, có thể xuất hiện các đỉnh năng lượng bất thường ở một số tần số nhất định, không giống với sự phân bố tự nhiên của nhiễu trong ảnh thật.

(ii) **Sự hiện diện của các họa tiết giả:**

Các thuật toán tạo ảnh, đặc biệt là các kiến trúc GAN có sử dụng các lớp học để tăng kích thước (upsampling layers), có thể vô tình tạo ra các mẫu lặp lại hoặc các cấu trúc dạng lưới trong miền không gian, dẫn đến sự xuất hiện của các đỉnh hoặc các đường kẻ rõ rệt, có tính chu kỳ trong miền tần số. Những họa tiết này thường không có trong phổ tần số của nhiễu từ ảnh thật.

(iii) **Độ hỗn loạn của phổ:**

Ảnh thật với nhiễu ngẫu nhiên thường có entropy phổ cao hơn (phân bố tần số đa dạng, ít trật tự hơn), trong khi ảnh AI có thể có entropy thấp hơn do các quy luật hoặc sự làm mịn trong quá trình tạo.

Trong bài, nhóm em lấy hình ảnh $S_{\text{norm}}(u, v)$ (đặc trưng phổ tần số đã qua xử lý) đưa trực tiếp vào một nhánh của mô hình CNN, cho phép mạng tự học các đặc điểm phân biệt quan trọng từ miền tần số của nhiễu dư để hỗ trợ việc phân loại ảnh AI và ảnh thật.

2.2.3 Bản đồ nhất quán nhiễu dư

Ngoài phân tích các đặc điểm nhiễu dư trong miền không gian và tần số, việc đánh giá sự đồng nhất của nhiễu trên toàn bộ bề mặt ảnh cũng cung cấp những manh mối để phát hiện các can thiệp hoặc nguồn gốc nhân tạo của ảnh. Bởi vì nhiễu phát sinh từ cảm biến trong ảnh thật thường có xu hướng phân bố một cách ngẫu nhiên và tương đối đồng đều trên toàn bộ diện tích ảnh. Ngược lại, ảnh do AI

tạo ra có thể có nhiễu được thêm vào một cách có chủ đích theo thuật toán, hoặc ngược lại, bị làm mịn quá mức ở một số vùng, dẫn đến sự không đồng đều trong phân bố nhiễu. Do đó, chúng em xây dựng bản đồ nhất quán nhiễu dư (residual noise consistency map) để làm cơ sở phân biệt nguồn gốc ảnh.

Quy trình xây dựng và phân tích bản đồ nhất quán nhiễu dư được mô tả như sau:

(i) **Chuẩn bị đầu vào:**

Đầu vào chính cho quá trình này là ảnh nhiễu dư $W(x, y)$, đã được trích xuất từ ảnh gốc như mô tả tại mục 2.2.1. Các tham số quan trọng khác bao gồm kích thước cửa sổ phân tích (window_size), xác định kích thước của các vùng ảnh cục bộ sẽ được kiểm tra, và bước nhảy (stride), quy định khoảng cách dịch chuyển giữa các cửa sổ trượt liên tiếp khi quét qua ảnh.

(ii) **Tính toán đặc trưng nhiễu cục bộ:**

Ảnh nhiễu dư W được chia thành các vùng nhỏ (patches), thường là các khối vuông có kích thước bằng window_size. Các vùng này được trích xuất theo kiểu không chồng lấn, sau khi xử lý một vùng, cửa sổ sẽ dịch chuyển một khoảng bằng chính kích thước của nó.

Đối với mỗi patch ảnh này, một đặc trưng đo lường mức độ nhiễu được tính toán. Lựa chọn đặc trưng độ lệch chuẩn của cường độ các pixel trong vùng đó. Độ lệch chuẩn cục bộ phản ánh sự biến thiên của các giá trị pixel trong từng vùng nhỏ, qua đó ước lượng mức độ nhiễu tại khu vực đó.

$$\sigma_{\text{local}}\ (P_{i,j}) = \sqrt{\frac{1}{N_p - 1} \sum_{(x,y)\in P_{i,j}} \left(W(x, y) - \mu_{\text{local}}\ (P_{i,j})\right)^2} \tag{2.5}$$

Trong đó:

- N_p là tổng số pixel trong vùng $P_{i,j}$ (ví dụ: $N_p = window\_size \times window\_size$).
- $W(x, y)$ là giá trị pixel nhiễu dư tại tọa độ (x, y) bên trong vùng $P_{i,j}$.
- $\mu_{\text{local}}(P_{i,j})$ là giá trị trung bình của các pixel nhiễu dư bên trong vùng $P_{i,j}$:

$$\mu_{\text{local}}(P_{i,j}) = \frac{1}{N_p} \sum_{(x,y)\in P_{i,j}} W(x, y)$$

Ý nghĩa của độ lệch chuẩn cục bộ:

Độ lệch chuẩn cục bộ σ_{local} phản ánh mức độ phân tán hay sự biến thiên của các giá trị pixel nhiễu dư xung quanh giá trị trung bình của chúng trong một

vùng nhỏ.

(i) Nếu các giá trị pixel nhiễu trong vùng đó có sự chênh lệch lớn (ví dụ, có cả giá trị rất dương và rất âm, hoặc dao động mạnh), σ_{local} sẽ cao. Điều này thường tương ứng với một vùng có "nhiễu mạnh" hoặc "kết cấu nhiễu rõ rệt".

(ii) Nếu các giá trị pixel nhiễu trong vùng đó gần như đồng nhất hoặc có sự chênh lệch nhỏ, σ_{local} sẽ thấp. Điều này thường tương ứng với một vùng "ít nhiễu" hoặc "nhiễu rất mịn".

(iii) **Tính toán đặc trưng nhiễu toàn cục:**

Sau khi đã phân tích nhiễu dư ở quy mô cục bộ bằng cách tính độ lệch chuẩn cho từng patch ảnh, bước tiếp theo là xác định một giá trị tham chiếu đại diện cho đặc tính nhiễu trên toàn bộ ảnh là nhiễu toàn cục. Nó đóng vai trò như một cái nền để so sánh với các đặc trưng nhiễu cục bộ, từ đó phát hiện ra những vùng có hành vi nhiễu bất thường.

$$\sigma_{\text{global}} = \sqrt{\frac{1}{M \times N - 1} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} \big(W(x,y) - \mu_{\text{global}}\big)^2} \tag{2.6}$$

Trong đó:

- M và N là kích thước (chiều cao và chiều rộng) của toàn bộ ảnh nhiễu dư W.
- $W(x,y)$ là giá trị pixel nhiễu dư tại tọa độ (x,y).
- μ_{global} là giá trị trung bình của tất cả các pixel nhiễu dư trong toàn bộ ảnh W:

$$\mu_{\text{global}} = \frac{1}{M \times N} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} W(x,y)$$

Ý nghĩa của độ lệch chuẩn toàn cục:

Giá trị σ_{global} cho biết mức độ phân tán trung bình của các giá trị nhiễu dư trên toàn bộ ảnh.

(i) Một giá trị σ_{global} cao cho thấy ảnh có mức độ nhiễu tổng thể lớn, các giá trị pixel nhiễu biến động mạnh.

(ii) Một giá trị σ_{global} thấp cho thấy ảnh có mức độ nhiễu tổng thể nhỏ, các giá trị pixel nhiễu tương đối đồng nhất hoặc gần bằng 0.

Giá trị σ_{global} này, sau khi được tính toán, sẽ được sử dụng ở bước tiếp theo (xây dựng bản đồ nhất quán nhiễu) để so sánh với từng giá trị $\sigma_{\text{local}}(P_{i,j})$ của các vùng ảnh cục bộ, nhằm xác định mức độ "bất thường" hay "không nhất quán" của nhiễu tại mỗi vùng. Đây là một bước quan trọng để làm nổi bật các khu vực có thể đã bị can thiệp hoặc có nguồn gốc tạo sinh khác biệt so với phần còn lại của ảnh.

(iv) **Xây dựng bản đồ nhất quán nhiễu:**

Bản đồ nhất quán nhiễu được tạo ra bằng cách so sánh đặc trưng nhiễu cục bộ của mỗi vùng với đặc trưng nhiễu toàn cục. Cụ thể, đối với mỗi vùng ảnh cục bộ, sự khác biệt giữa độ lệch chuẩn cục bộ và độ lệch chuẩn toàn cục được tính toán. Một thước đo về sự không nhất quán cho một vùng có thể được định nghĩa là:

$$C_{\text{patch}} = \frac{|\sigma_{\text{local}} - \sigma_{\text{global}}|}{\sigma_{\text{global}} + \epsilon} \tag{2.7}$$

Trong đó

- σ_{local} là độ lệch chuẩn của vùng cục bộ,
- σ_{global} là độ lệch chuẩn của toàn bộ ảnh nhiễu dư,
- ϵ là một hằng số nhỏ để tránh chia cho không.

Giá trị C_{patch} này sau đó được gán cho tất cả các pixel thuộc vùng cục bộ tương ứng trên một bản đồ mới (heatmap). Các vùng có giá trị C_{patch} cao cho thấy sự khác biệt đáng kể giữa nhiễu cục bộ và nhiễu toàn cục, được coi là các vùng có nhiễu bất thường. Cuối cùng, bản đồ này thường được chuẩn hóa giá trị (ví dụ, về khoảng $[0, 1]$) để thuận tiện cho việc hiển thị và xử lý tiếp theo.

(v) **Phân tích và ứng dụng bản đồ nhất quán nhiễu:**

Kết quả thu được là một bản đồ nhiệt (heatmap), $M_{\text{consis}}(x, y)$, biểu thị mức độ không nhất quán của nhiễu tại mỗi vị trí trên ảnh.

- Trong ảnh thật chưa qua chỉnh sửa, bản đồ này thường có xu hướng khá đồng đều với các giá trị thấp, phản ánh sự phân bố ngẫu nhiên và đồng nhất của nhiễu cảm biến.
- Ngược lại, trong ảnh AI hoặc ảnh đã bị chỉnh sửa, bản đồ nhất quán nhiễu có thể cho thấy các vùng có giá trị nổi bật (bất thường). Những vùng này có thể tương ứng với các khu vực bị làm mịn quá mức (dẫn đến σ_{local} rất thấp so với σ_{global}) hoặc các khu vực có nhiễu giả được thêm vào một cách

nhân tạo (dẫn đến σ_{local} rất cao hoặc có tính chất khác biệt).

Bản đồ nhất quán nhiễu $M_{\text{consis}}(x, y)$ sau khi được chuẩn hóa sẽ được sử dụng như một kênh đặc trưng trực quan để làm đầu vào cho mô hình mạng nơ-ron ýtích chập. Điều này cho phép mô hình tự học cách nhận diện các mẫu bất thường về sự phân bố nhiễu để hỗ trợ quá trình phân loại ảnh.

2.3 Phương pháp phân tích dựa trên mức độ lỗi (Error Level Analysis - ELA)

2.3.1 Đặc trưng mức độ lỗi ELA

Phân tích mức độ lỗi (Error Level Analysis - ELA) là một kỹ thuật được sử dụng để phát hiện các dấu hiệu can thiệp hoặc chỉnh sửa trên ảnh kỹ thuật số, đặc biệt là những thay đổi liên quan đến việc lưu lại ảnh ở định dạng nén mất mát như JPEG. Nguyên lý cơ bản của ELA là khi một ảnh JPEG được lưu lại nhiều lần, các vùng ảnh khác nhau sẽ có tốc độ suy giảm chất lượng (do quá trình nén lặp lại) khác nhau. Các vùng ảnh gốc, chưa bị chỉnh sửa, khi được nén lại ở một mức chất lượng nhất định sẽ có mức độ lỗi tương đối đồng đều và thấp khi so sánh với chính nó ở phiên bản nén đó. Ngược lại, các vùng đã bị chỉnh sửa (thêm đối tượng từ một nguồn khác, sao chép một vùng trong ảnh, hoặc các thao tác làm thay đổi cục bộ cấu trúc pixel) khi được nén lại sẽ có mức độ lỗi khác biệt đáng kể so với các vùng gốc, thường là cao hơn. Điều này xảy ra do các vùng bị can thiệp thường có đặc điểm nén JPEG khác với phần còn lại của ảnh gốc.

2.3.2 Quy trình thu thập và phân tích mức độ lỗi ELA

(i) **Chuẩn bị ảnh gốc:**

Ảnh đầu vào là ảnh màu, được xử lý để đảm bảo tính nhất quán cho việc phân tích như chuyển đổi sang một không gian màu chuẩn RGB và thay đổi kích thước về một kích thước đồng nhất. Ký hiệu ảnh gốc đã chuẩn bị này là I_{orig}.

(ii) **Nén lại ảnh gốc:**

Ảnh I_{orig} được lưu lại dưới định dạng JPEG với một mức chất lượng được xác định trước. Quá trình này tạo ra một phiên bản nén của ảnh gốc, ký hiệu là I_{comp}. Việc lựa chọn mức chất lượng nén này là một yếu tố quan trọng:

- Nếu mức chất lượng quá cao, sự khác biệt về lỗi có thể không đủ rõ ràng.
- Nếu mức chất lượng quá thấp, toàn bộ ảnh có thể bị suy giảm chất lượng mạnh mẽ, làm mờ đi các dấu vết chỉnh sửa tinh vi.

(iii) **Tính toán ảnh chênh lệch:**

Ảnh chênh lệch, ký hiệu là I_{diff}, được tạo ra bằng cách tính toán giá trị tuyệt đối của sự khác biệt giữa từng cặp pixel tương ứng của ảnh gốc I_{orig} và ảnh đã

nén I_{comp} cho từng kênh màu. Đối với mỗi pixel (x, y):

$$R_{\text{diff}}(x, y) = |R_{\text{orig}}(x, y) - R_{\text{comp}}(x, y)|$$
$$G_{\text{diff}}(x, y) = |G_{\text{orig}}(x, y) - G_{\text{comp}}(x, y)|$$
$$B_{\text{diff}}(x, y) = |B_{\text{orig}}(x, y) - B_{\text{comp}}(x, y)|$$

Pixel mới tại (x, y) trên I_{diff} sẽ có giá trị $(R_{\text{diff}}(x, y), G_{\text{diff}}(x, y), B_{\text{diff}}(x, y))$. I_{diff} này thể hiện "mức độ lỗi" do quá trình nén lại gây ra.

(iv) **Tăng cường độ sáng:**

Các giá trị chênh lệch trong I_{diff} thường rất nhỏ và khó quan sát trực tiếp. Để làm nổi bật những khác biệt này, ảnh chênh lệch I_{diff} được tăng cường độ sáng.

Đầu tiên, tìm giá trị chênh lệch tối đa (max_diff) trên tất cả các kênh màu của I_{diff}.

Sau đó, một hệ số khuếch đại ($scale$) được tính toán (dựa trên max_diff). Mỗi giá trị pixel trong I_{diff} sau đó được nhân với hệ số $scale$ này và thường được cắt bỏ ở một giá trị tối đa (khoảng 255) để đảm bảo nằm trong dải màu hiển thị hợp lệ.

Kết quả thu được là ảnh ELA cuối cùng. Trong ảnh này:

(i) Các vùng có độ sáng cao hơn (gần với màu trắng) cho thấy mức độ lỗi lớn hơn khi nén lại, gợi ý rằng đó có thể là các vùng đã bị chỉnh sửa.

(ii) Các vùng tối hơn (gần với màu đen) có mức độ lỗi thấp, thường tương ứng với các vùng gốc của ảnh.

2.3.3 Phân tích và ứng dụng ảnh ELA

Ảnh ELA không trực tiếp chỉ ra loại chỉnh sửa cụ thể nào đã được thực hiện, mà chủ yếu làm nổi bật các vùng có đặc điểm nén JPEG khác biệt so với phần còn lại của ảnh. Khi phân tích một ảnh ELA, các đặc điểm sau thường được quan sát:

(i) Các vùng đồng nhất, tối màu => thường chỉ ra các vùng ảnh gốc, chưa bị can thiệp đáng kể hoặc có lịch sử nén đồng nhất với toàn bộ ảnh.

(ii) Các vùng sáng hơn, có kết cấu rõ rệt hoặc các cạnh sắc nét nổi bật => dấu hiệu của việc thêm đối tượng từ một ảnh khác do có lịch sử nén JPEG khác biệt, sao chép và dán một vùng trong cùng một ảnh, hoặc các thao tác chỉnh sửa cục bộ làm thay đổi cấu trúc nén.

(iii) Không nhất quán trong độ sáng các cạnh của đối tượng do trong một ảnh JPEG gốc, các cạnh sắc nét tự nhiên thường có mức độ lỗi hơi cao hơn so với

các vùng phẳng do bản chất của thuật toán nén JPEG. Tuy nhiên, nếu một đối tượng được thêm vào từ một nguồn khác, các cạnh của nó trong ảnh ELA có thể sáng hơn nhiều so với các cạnh tự nhiên khác trong ảnh, tạo ra sự không nhất quán.

2.4 Các độ đo đánh giá mô hình phân loại ảnh

Trong quá trình đánh giá hiệu suất của mô hình phân loại ảnh theo nguồn gốc (ảnh thật hoặc ảnh do AI tạo ra), nhóm chúng em đã sử dụng một tập hợp các độ đo phổ biến trong học máy nhằm phản ánh đầy đủ và toàn diện khả năng phân loại của mô hình. Các độ đo này được tính toán dựa trên bốn chỉ số cơ bản trong ma trận nhầm lẫn (confusion matrix) như sau:

- **TP (True Positive):** Số lượng ảnh thật được mô hình phân loại đúng là ảnh thật.
- **TN (True Negative):** Số lượng ảnh AI được mô hình phân loại đúng là ảnh AI.
- **FP (False Positive):** Số lượng ảnh AI bị mô hình phân loại sai là ảnh thật.
- **FN (False Negative):** Số lượng ảnh thật bị mô hình phân loại sai là ảnh AI.

Từ các chỉ số này, nhóm sử dụng các độ đo cụ thể như sau:

a, Accuracy (Độ chính xác tổng thể)

Ý nghĩa: Accuracy đo lường tỷ lệ phần trăm các dự đoán đúng trên toàn bộ tập dữ liệu, phản ánh mức độ chính xác chung của mô hình. Công thức tính như sau:

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$

Ứng dụng: Đây là một độ đo trực quan, tuy nhiên có thể gây hiểu lầm trong trường hợp tập dữ liệu bị mất cân bằng, tức là số lượng ảnh thật và ảnh AI có sự chênh lệch lớn.

b, Precision (Độ chính xác theo lớp)

Ý nghĩa: Precision thể hiện trong số các ảnh được mô hình phân loại là một lớp cụ thể (ví dụ: ảnh thật), có bao nhiêu ảnh thực sự thuộc về lớp đó. Công thức tính cho lớp dương, ví dụ đối với ảnh thật là:

$$\text{Precision} = \frac{TP}{TP + FP}$$

Ứng dụng: Precision đặc biệt quan trọng trong các tình huống mà việc xác nhận sai một ảnh thuộc lớp này thành lớp kia (ví dụ: ảnh AI là ảnh thật) có thể gây ra

hậu quả nghiêm trọng (ví dụ: kiểm duyệt nội dung, nhận diện chứng cứ giả).

c, Recall (Độ bao phủ lớp/ Độ nhạy)

Ý nghĩa: Recall thể hiện khả năng mô hình không bỏ sót các mẫu của một lớp cụ thể (ví dụ: ảnh thật), đo tỷ lệ các mẫu thuộc lớp đó được mô hình phát hiện đúng trên tổng số mẫu thực sự thuộc lớp đó. Công thức tính cho lớp dương (ví dụ: ảnh thật) là:

$$\text{Recall} = \frac{TP}{TP + FN}$$

Ứng dụng: Recall là chỉ số quan trọng trong các hệ thống yêu cầu độ phủ cao, ví dụ như xác thực hình ảnh thật trong báo chí, truyền thông hoặc điều tra pháp lý, nơi việc bỏ sót một trường hợp thật có thể gây ra vấn đề.

d, F1-Score (Điểm điều hòa giữa Precision và Recall)

Ý nghĩa: F1-Score là trung bình điều hòa (harmonic mean) giữa Precision và Recall, được sử dụng để cân bằng giữa hai mục tiêu là độ chính xác và khả năng phát hiện đầy đủ.

$$\text{F1-Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

Ứng dụng: Chỉ số này thường được sử dụng trong bài toán phân loại với dữ liệu mất cân bằng, hoặc khi cả hai yếu tố Precision và Recall đều quan trọng và cần được tối ưu đồng thời.

e, AUC (Area Under the Curve – Diện tích dưới đường cong ROC)

Ý nghĩa: Đường cong ROC (Receiver Operating Characteristic) biểu diễn mối quan hệ giữa tỷ lệ True Positive Rate (chính là Recall hay Độ nhạy) và tỷ lệ False Positive Rate (tỷ lệ ảnh AI bị nhận nhầm là thật) tại các ngưỡng phân loại khác nhau. AUC là diện tích nằm dưới đường cong ROC này, có giá trị từ 0 đến 1. AUC càng gần 1, khả năng phân biệt giữa hai lớp ảnh (ảnh thật và ảnh AI) của mô hình càng tốt, bất kể ngưỡng phân loại được chọn.

Ứng dụng: AUC hữu ích trong việc đánh giá tổng thể khả năng phân biệt của mô hình tại nhiều ngưỡng khác nhau và thường được sử dụng để so sánh hiệu năng giữa các mô hình khác nhau. Nó cũng có thể hỗ trợ việc tối ưu hóa ngưỡng phân loại (threshold) phù hợp với yêu cầu cụ thể của bài toán thực tế.

Ảnh ELA đã được xử lý và chuẩn hóa thường được sử dụng làm đặc trưng đầu vào cho các mô hình học máy như mạng nơ-ron tích chập (CNN). Các mô hình này được huấn luyện để học cách phân biệt các mẫu ELA đặc trưng của ảnh gốc và ảnh đã qua chỉnh sửa, từ đó hỗ trợ việc đưa ra dự đoán về tính toàn vẹn và nguồn gốc của ảnh đầu vào.

CHƯƠNG 3. THỰC NGHIỆM VÀ ĐÁNH GIÁ KẾT QUẢ

3.1 Xây dựng model đặc trưng không đồng nhất phản hồi ảnh (Photo-Response Non-Uniformity - PRNU)

3.1.1 Chuẩn bị và phân chia tập dữ liệu

Các hình ảnh trong thư mục real được thu thập từ các nguồn bao gồm: "RAISE Dataset (Raw Images Dataset): tập hợp ảnh RAW và JPEG từ máy ảnh DSLR chuyên dụng", "Unsplash Images Collection: ảnh tự nhiên, phong cảnh, con người chụp bằng máy ảnh thật", "Kaggle – ai-generated-images-vs-real-images: thư mục real - tập hợp ảnh chụp thực tế" Các hình ảnh trong thư mục fake được tổng hợp từ nguồn: "Kaggle - ai-generated-images-vs-real-images: Thư mục fake - tập hợp các ảnh tạo bởi AI"

Nhóm chúng em gán nhãn bộ dữ liệu cho mô-đun PRNU bao gồm 2 label: fake và real. Thư mục **real** chứa các ảnh gốc, được chụp bởi máy ảnh thật sự còn thư mục **fake** bao gồm các ảnh không phải là ảnh gốc nguyên bản, cụ thể là ảnh do AI tạo ra bởi nhiều mô-đun sinh ảnh khác nhau.

Cụ thể hơn, thư mục real là tập hợp gồm 6000 hình ảnh được thu thập trực tiếp từ cảm biến của đa dạng thiết bị máy ảnh (máy ảnh DSLR chuyên nghiệp, máy ảnh compact, camera trên điện thoại thông minh, webcam, v.v.). Điều kiện tiên quyết là các ảnh này phải được giữ ở trạng thái nguyên sơ nhất, không chịu bất kỳ sự can thiệp nào từ phần mềm chỉnh sửa ảnh sau khi chụp. Lý tưởng nhất, đây là các tệp ảnh RAW (nếu hệ thống có khả năng xử lý) hoặc các tệp JPEG được lưu trữ với chất lượng cao nhất (mức nén tối thiểu) ngay từ thiết bị chụp. Việc này đảm bảo tín hiệu PRNU không bị suy yếu hay biến đổi bởi các thuật toán xử lý bên ngoài.

Đối với thư mục ảnh fake, nó bao gồm 6000 hình ảnh được tổng hợp hoàn toàn bởi các thuật toán trí tuệ nhân tạo, đặc biệt là các mô hình sinh ảnh tiên tiến như Generative Adversarial Networks (GANs), Diffusion Models (ví dụ: Stable Diffusion, DALL-E, Midjourney), và các kiến trúc tương tự khác. Chúng không phải là kết quả của việc chụp một cảnh thực tế bằng máy ảnh vật lý.

Các ảnh trong thư mục ảnh fake có các đặc điểm cụ thể khác so với ảnh thật: (i) Ảnh AI thường không mang dấu vết PRNU đặc trưng của bất kỳ cảm biến máy ảnh vật lý nào. Nếu có sự hiện diện của nhiễu, nó thường không nhất quán trên toàn bộ ảnh hoặc không khớp với bất kỳ mẫu PRNU đã biết nào. (ii) Thay vì PRNU tự nhiên, ảnh AI chứa các mẫu nhiễu hoặc các sai sót, dấu vết nhân tạo đặc thù của quá trình sinh ảnh, bao gồm các chi tiết phi logic, sự lặp lại bất thường của họa tiết,

hoặc cấu trúc không tự nhiên ở một số vùng nhất định.

3.1.2 Quy trình trích xuất đặc trưng cho mô hình PRNU

Bước 1: Đọc và chuẩn bị ảnh gốc:

Ảnh đầu vào từ đường dẫn được đọc bằng thư viện Pillow và chuyển đổi thành một mảng numpy và chuẩn hóa về định dạng RGB.

Bước 2: Trích xuất nhiễu dư (extract_single):

Ảnh màu RGB đầu vào được chuyển đổi thành ảnh xám. Nếu ảnh đầu vào là 2 chiều thì sẵn đã là ảnh xám, còn nếu là ảnh 3 chiều thì là ảnh chuẩn để chuyển đổi sang ảnh xám. Hoặc nếu là ảnh 4 kênh RGBA, kênh alpha sẽ được loại bỏ trước khi chuyển đổi.

Ảnh xám sau đó được làm mịn bằng cách áp dụng bộ lọc Gaussian. Tham số σ được truyền vào extract_single với giá trị mặc định là 1.5 quyết định mức độ làm mịn. Bộ lọc Gaussian này có kích thước kernel được tính bằng công thức dưới, luôn đảm bảo k_size là một số lẻ

$$k\_size = (6 \cdot \sigma + 1) \tag{3.1}$$

Tiếp theo, một trục tọa độ một chiều ax được tạo ra, chứa ksize điểm cách đều nhau trong khoảng từ $-(ksize//2)$ đến $ksize//2$. Từ trục ax này, một lưới tọa độ hai chiều (xx, yy) được hình thành. Các giá trị của nhân lọc Gaussian 2D sau đó được tính toán tại mỗi điểm trên lưới (xx, yy) bằng công thức của hàm Gaussian:

$$\text{kernel} = e^{-\frac{xx^2+yy^2}{2\sigma^2}} \tag{3.2}$$

Cuối cùng, nhân lọc được chuẩn hóa bằng cách chia tất cả các phần tử của nó cho tổng giá trị của tất cả các phần tử, đảm bảo rằng tổng các trọng số của nhân lọc bằng 1 để bảo toàn độ sáng tổng thể của ảnh sau khi lọc.

Quá trình tích chập 2D được thực hiện bằng cách nhân lọc Gausian2D với ảnh xám với tham số mode = 'same' để đảm bảo đầu ra có cùng kích thước với ảnh đầu vào và boundary = 'symm', sử dụng điều kiện biên đối xứng để xử lý các pixel ở rìa ảnh nhằm giảm thiểu các hiệu ứng không mong muốn ở biên.

Nhiễu dư được thu nhận bằng cách lấy ảnh thang xám gốc trừ đi phiên bản đã được lọc Gaussian của nó.

Nhiễu dư thu được sau đó được xử lý qua hàm zero_mean_total. Hàm này chia

ảnh nhiễu thành bốn lưới con xen kẽ và áp dụng hàm zero_mean cho từng lưới con. Hàm zero_mean thực hiện việc loại bỏ giá trị trung bình theo từng kênh màu, sau đó loại bỏ giá trị trung bình theo từng hàng và từng cột để giảm thiểu các thành phần cấu trúc còn sót lại và làm nổi bật hơn các thành phần nhiễu ngẫu nhiên.

Ảnh nhiễu dư đã được chuẩn hóa sẽ được thay đổi kích thước về một kích thước cố định là $(224, 224)$ pixel nhằm đảm bảo đầu vào đồng nhất cho mô hình CNN. Kết quả của bước này là một ảnh nhiễu dư 2D, N_{residue}, có kích thước $(224, 224)$.

Bước 3: Trích xuất đặc trưng phổ tần số từ nhiễu dư:

Ảnh nhiễu dư N_{residue} thu được ở bước trên được đưa vào hàm freqq để phân tích trong miền tần số. Phép biến đổi Fourier nhanh hai chiều (2D FFT) được áp dụng cho ảnh nhiễu dư để chuyển nó từ miền không gian sang miền tần số.

Kết quả FFT được dịch chuyển để thành phần tần số thấp (DC component) nằm ở trung tâm của phổ, giúp việc phân tích và trực quan hóa dễ dàng hơn. Từ kết quả phức của FFT, tính các phổ biên độ có trong miền tần số.

Để tăng cường các chi tiết có biên độ thấp và đưa giá trị về một dải phù hợp, phép biến đổi logarit (log1p – log của 1 cộng với giá trị biên độ) được áp dụng cho phổ biên độ. Sau đó, kết quả được chuẩn hóa Min-Max để các giá trị nằm trong khoảng $[0, 1]$. Kết quả của bước này là một ảnh phổ tần số 2D, N_{fft}, có kích thước $(224, 224)$, biểu diễn năng lượng của các thành phần tần số trong nhiễu dư.

Bước 4: Trích xuất bản đồ nhất quán nhiễu dư:

Ảnh nhiễu dư N_{residue} cũng được sử dụng để tạo ra bản đồ nhất quán nhiễu. Đầu tiên, độ lệch chuẩn của tất cả các pixel trong toàn bộ ảnh nhiễu dư N_{residue} được tính toán (σ_{global}). Ảnh nhiễu dư được quét qua bằng một cửa sổ trượt với kích thước được xác định bởi tham số window_size (giá trị truyền vào là 8). Bước nhảy của cửa sổ là window_size // 2 (tức là 4 pixel), tạo ra sự chồng lấn giữa các vùng. Đối với mỗi patch ảnh được trích xuất bởi cửa sổ trượt, độ lệch chuẩn của các pixel bên trong vùng đó được tính toán (σ_{local}). Sự khác biệt tương đối giữa σ_{local} của mỗi vùng và σ_{global} được tính theo công thức

$$\frac{|\sigma_{\text{local}} - \sigma_{\text{global}}|}{\sigma_{\text{global}} + \epsilon} \tag{3.3}$$

(với ϵ là một giá trị nhỏ để tránh chia cho không).

Giá trị này được gán cho các pixel tương ứng trên một bản đồ mới. Bản đồ nhiệt thu được sau đó được chuẩn hóa Min-Max để các giá trị nằm trong khoảng $[0, 1]$. Kết quả của bước này là một bản đồ nhất quán nhiễu 2D, N_{consis}, có kích thước $(224, 224)$, làm nổi bật các vùng có đặc tính nhiễu khác biệt so với phần còn lại của ảnh.

Bước 5: Tổng hợp đặc trưng và nhãn:

Ba ảnh đặc trưng thu được (N_{residue}, N_{fft}, N_{consis}) mỗi ảnh đều có kích thước $(224, 224)$. Chúng được mở rộng thêm một chiều (channel) để có dạng $(224, 224, 1)$ và sau đó được lưu trữ cùng với nhãn label (0 cho ảnh AI, 1 cho ảnh thật) tương ứng của ảnh đầu vào. Các mảng numpy của N_{residue}, N_{fft}, N_{consis} và nhãn y được tập hợp lại thành các danh sách X_noise, X_fft, X_consis và y. Sau khi xử lý toàn bộ tập dữ liệu, các danh sách này được chuyển đổi thành mảng numpy và phân chia thành các tập huấn luyện, kiểm định và kiểm thử để sử dụng cho việc huấn luyện và đánh giá mô hình CNN.

3.1.3 Kiến trúc mô hình CNN phân loại

Mô hình nhận ba đầu vào riêng biệt, mỗi đầu vào là một ảnh thang xám một kênh có kích thước $(224, 224, 1)$ tương ứng với ba loại đặc trưng đã được trích xuất từ mỗi ảnh đầu vào: nhiễu dư, phổ tần số của nhiễu dư, và bản đồ nhất quán nhiễu dư. Vì vậy mô hình cũng bao gồm ba nhánh xử lý đặc trưng riêng biệt, sau đó các đặc trưng học được từ mỗi nhánh sẽ được hợp nhất để đưa ra quyết định phân loại cuối cùng.

(i) ip -> đầu vào cho ảnh nhiễu dư.

(ii) fft_in -> đầu vào cho ảnh phổ tần số của nhiễu dư.

(iii) cm_in -> đầu vào cho ảnh bản đồ nhất quán nhiễu dư.

a, Nhánh xử lý nhiễu dư (xử lý ip)

(a.i) Khối đầu tiên bao gồm hai lớp Conv2D với 64 bộ lọc, kích thước nhân 3×3, và padding='same'. Mỗi lớp Conv2D được theo sau bởi một lớp BatchNormalization để ổn định và tăng tốc quá trình huấn luyện, và một hàm kích hoạt LeakyReLU với hệ số $\alpha = 0.1$ để tránh hiện tượng chết ReLU và cho phép một lượng nhỏ gradient âm đi qua. Kết thúc khối này là một lớp MaxPooling2D với kích thước cửa sổ 2×2 để giảm chiều dữ liệu. Hai khối tiếp theo sử dụng kết nối tắt, một kỹ thuật phổ biến trong các mạng dư.

(a.ii) Khối thứ hai bắt đầu bằng việc chiếu đầu vào qua một lớp Conv2D với 128 bộ lọc, kích thước nhân 1×1 (padding='same') để tạo shortcut. Sau đó, luồng chính đi qua hai lớp Conv2D với 128 bộ lọc, nhân 3×3, mỗi lớp đều có Batch-

Normalization và LeakyReLU($\alpha = 0.1$). Kết quả từ luồng chính này sau đó được cộng (Add) với shortcut đã tạo. Cuối cùng là một lớp LeakyReLU($\alpha = 0.1$) và MaxPooling2D(2×2).

(a.iii) Khối thứ ba có cấu trúc tương tự khối thứ hai nhưng sử dụng 256 bộ lọc cho cả lớp tạo shortcut và các lớp Conv2D trong luồng chính. Sau ba khối tích chập này, một lớp GlobalAveragePooling2D được áp dụng để chuyển bản đồ đặc trưng 2D thành một vector đặc trưng 1D, giúp làm giảm số lượng tham số.

b, Nhánh xử lý phổ tần số (xử lý fft_in)

(b.i) Khối đầu tiên có cấu trúc Conv2D (48 bộ lọc, nhân 3×3, padding='same'), BatchNormalization, LeakyReLU($\alpha = 0.1$), và MaxPooling2D(2×2).

(b.ii) Khối thứ hai có cấu trúc Conv2D (96 bộ lọc, nhân 3×3, padding='same'), BatchNormalization, LeakyReLU($\alpha = 0.1$), và MaxPooling2D(2×2). Cuối cùng, một lớp GlobalAveragePooling2D được sử dụng.

c, Nhánh xử lý bản đồ nhất quán (xử lý cm_in)

(c.i) Khối đầu tiên có cấu trúc Conv2D (48 bộ lọc, nhân 3×3, padding='same'), BatchNormalization, LeakyReLU($\alpha = 0.1$), và MaxPooling2D(2×2).

(c.ii) Khối thứ hai có cấu trúc Conv2D (96 bộ lọc, nhân 3×3, padding='same'), BatchNormalization, LeakyReLU($\alpha = 0.1$), và MaxPooling2D(2×2). Cuối cùng, một lớp GlobalAveragePooling2D được sử dụng.

d, Hợp nhất đặc trưng và các lớp kết nối đầy đủ

Các vector đặc trưng 1D thu được từ ba nhánh xử lý (sau các lớp GlobalAveragePooling2D) được hợp nhất lại bằng một lớp. Vector đặc trưng hợp nhất này sau đó được đưa qua hai khối Dense để học các mối quan hệ phức tạp hơn giữa các loại đặc trưng.

(d.i) Khối Dense đầu tiên có 512 đơn vị, không sử dụng hàm kích hoạt trực tiếp trong định nghĩa lớp Dense mà theo sau bởi BatchNormalization và LeakyReLU($\alpha = 0.1$). Một lớp Dropout với tỷ lệ 0.4 được thêm vào để giảm thiểu overfitting.

(d.ii) Khối Dense thứ hai có 256 đơn vị, cũng theo sau bởi BatchNormalization, LeakyReLU($\alpha = 0.1$), và một lớp Dropout với tỷ lệ 0.3.

e, Lớp đầu ra

Cuối cùng, một lớp Dense với 1 đơn vị và hàm kích hoạt sigmoid được sử dụng để đưa ra dự đoán xác suất. Hàm sigmoid phù hợp cho bài toán phân loại nhị phân, với đầu ra là một giá trị trong khoảng $(0, 1)$ biểu thị xác suất ảnh thuộc về lớp ảnh thật (hoặc ảnh AI).

3.1.4 Đánh giá hiệu năng của mô hình PRNU

a, Hiệu năng trên tập huấn luyện (train) và tập kiểm tra (validation)

Mô hình CNN kết hợp đã được huấn luyện trên tập huấn luyện (X_noise_train, X_fft_train, X_consis_train, và y_train) và hiệu năng của nó được theo dõi trên tập kiểm định (X_noise_val, X_fft_val, X_consis_val, và y_val) qua từng epoch. Các thiết lập huấn luyện là:

(i) **Số epochs tối đa:** 100.

(ii) **Kích thước batch:** 16.

(iii) **Optimizer** Adam với learning rate ban đầu là 0.0005.

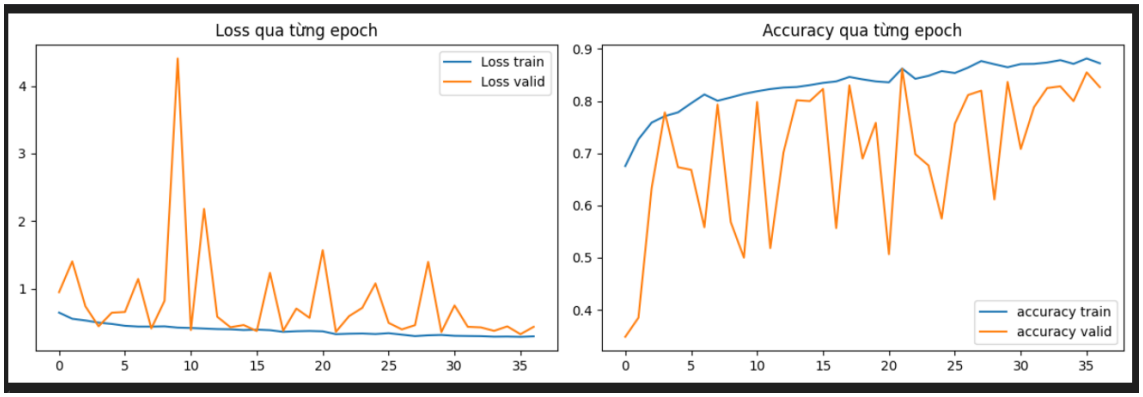
(iv) **Hàm mất mát:** binary_crossentropy.

(v) **Độ đo theo dõi:** Accuracy, AUC, Precision, và Recall

(vi) **Callbacks:**

- ReduceLROnPlateau -> tự động giảm tốc độ học khi val_loss không cải thiện sau 5 epochs.
- EarlyStopping -> dừng quá trình huấn luyện nếu val_accuracy không cải thiện sau 15 epochs, đồng thời khôi phục lại trọng số của mô hình tại epoch có val_accuracy tốt nhất (restore_best_weights=True).
- ModelCheckpoint: -> lưu lại mô hình tốt nhất (best_model.keras) dựa trên val_accuracy.

Diễn biến quá trình huấn luyện:



Hình 3.1: Biểu đồ hiệu năng mô hình PRNU

Mất mát trên tập huấn luyện) giảm nhanh chóng trong khoảng 4–5 epochs đầu tiên, từ giá trị ban đầu khoảng 0.66 xuống dưới 0.45. Sau đó, loss train tiếp tục giảm từ từ và ổn định ở mức thấp (quanh 0.3) từ khoảng epoch thứ 20 trở đi, cho thấy mô hình đang học tốt trên dữ liệu huấn luyện.

Còn mất mát trên tập kiểm định Có sự biến động mạnh hơn so với loss train. Trong những epoch đầu, loss valid giảm cùng với loss train, đạt giá trị thấp nhất là 0.3647 tại epoch thứ 22. Tuy nhiên, sau đó, loss valid có xu hướng tăng trở lại và dao động nhiều hơn, ví dụ như tăng vọt lên trên 1.0 ở các epoch 10, 11, và sau epoch 30 lại có xu hướng tăng nhẹ. Sự tăng trở lại của loss valid trong khi loss train vẫn tiếp tục giảm là một dấu hiệu của hiện tượng overfitting, tuy nhiên, cơ chế EarlyStopping và ModelCheckpoint đã giúp chọn ra mô hình tại thời điểm tối ưu trên tập kiểm định.

Độ chính xác trên tập huấn luyện tăng nhanh trong các epoch đầu, từ khoảng 0.67 lên trên 0.80 sau khoảng 5–6 epochs. Sau đó, accuracy train tiếp tục tăng dần và đạt mức khá cao, ổn định trên 0.85 từ khoảng epoch 20 và tiệm cận 0.90 ở các epoch cuối.

Còn độ chính xác trên tập kiểm định Cũng có sự biến động, nhưng nhìn chung có xu hướng tăng. Accuracy valid đạt giá trị cao nhất là 0.8617 (86.17%) tại epoch thứ 22. Mặc dù có những dao động mạnh (ví dụ, giảm xuống gần 0.5 ở epoch 10), đường accuracy valid cho thấy mô hình có khả năng tổng quát hóa tốt ở một số thời điểm.

Kết quả tốt nhất trên tập kiểm định:

Dựa trên log huấn luyện và việc sử dụng callback EarlyStopping và ModelCheckpoint theo dõi val_accuracy, mô hình đã dừng lại ở epoch thứ 37 và khôi phục trọng số từ epoch thứ 22. Đây là epoch mà mô hình đạt được hiệu năng tốt nhất trên tập kiểm định, với các chỉ số như sau:

- Validation Loss: 0.3647
- Validation Accuracy: 0.8617 (86.17%)
- Validation AUC: 0.9210
- Validation Precision: 0.8653
- Validation Recall: 0.8567

Sự chênh lệch không quá lớn giữa accuracy trên tập huấn luyện (khoảng 0.8550 tại epoch 22) và tập kiểm định (0.8617 tại epoch 22) tại thời điểm mô hình tốt nhất được chọn cho thấy mô hình không bị overfitting nghiêm trọng tại điểm dừng này. Tuy nhiên, sự dao động của val_loss và val_accuracy ở các epoch sau đó gợi ý rằng việc tiếp tục huấn luyện có thể dẫn đến overfitting nếu không có cơ chế dừng sớm. Tốc độ học cũng đã được tự động giảm bởi ReduceLROnPlateau tại epoch thứ 21 và 27, cho thấy mô hình đã cố gắng tìm điểm hội tụ tốt hơn.

b, Hiệu năng trên tập thử nghiệm (test) qua các ngưỡng Threshold với F1 score

```
19/19 ━━━━━━━━━━━━━━━━━━━━ 4s 170ms/step
Threshold 0.30 → F1: 0.8489
Threshold 0.32 → F1: 0.8519
Threshold 0.34 → F1: 0.8536
Threshold 0.36 → F1: 0.8571
Threshold 0.38 → F1: 0.8553
Threshold 0.40 → F1: 0.8516
Threshold 0.42 → F1: 0.8515
Threshold 0.44 → F1: 0.8487
Threshold 0.46 → F1: 0.8501
Threshold 0.48 → F1: 0.8500
Threshold 0.50 → F1: 0.8451
Threshold 0.52 → F1: 0.8416
Threshold 0.54 → F1: 0.8319
Threshold 0.56 → F1: 0.8301
Threshold 0.58 → F1: 0.8316
Threshold 0.60 → F1: 0.8283
Threshold 0.62 → F1: 0.8229
Threshold 0.64 → F1: 0.8188
Threshold 0.66 → F1: 0.8175
Threshold 0.68 → F1: 0.8096
Threshold 0.70 → F1: 0.7955
```

Hình 3.2: Hiệu năng mô hình PRNU trên tập test

F1-score cao nhất đạt được là 0.8571 tại ngưỡng quyết định là 0.36. Khi ngưỡng tăng dần từ 0.36 lên 0.70, điểm F1-score có xu hướng giảm dần, ví dụ tại ngưỡng 0.50 là 0.8451 và tại ngưỡng 0.70 chỉ còn 0.7955.

Với F1-score tốt nhất là 0.8571 (tại ngưỡng 0.36), mô hình cho thấy khả năng phân loại khá tốt trên tập kiểm thử. Điểm F1-score này phản ánh sự cân bằng tốt giữa khả năng xác định đúng các ảnh thật (Recall cho lớp "thật") và khả năng không nhầm lẫn ảnh AI thành ảnh thật (Precision cho lớp "thật"), cũng như tương tự cho lớp "ảnh AI".

3.2 Xây dựng model phân tích mức độ lỗi (Error Level Analysis - ELA)

3.2.1 Chuẩn bị và phân chia tập dữ liệu

Dữ liệu cho việc huấn luyện mô hình ELA được lấy từ các nguồn sau:

(i) Ảnh gốc (không qua chỉnh sửa): “Unsplash Images Collection: ảnh tự nhiên, phong cảnh, con người chụp bằng máy ảnh thật”, “Kaggle – ai-generated-images-vs-real-images: thư mục real - tập hợp ảnh chụp thực tế”.

(ii) Ảnh đã qua chỉnh sửa: “Columbia Image Splicing Detection Dataset: bộ ảnh chứa ảnh gốc và ảnh đã bị splicing để phục vụ nghiên cứu phát hiện ảnh chỉnh sửa.” Ngoài ra, nhóm em cũng tự tạo ảnh chỉnh sửa bằng cách sử dụng các công cụ phần mềm như Adobe Photoshop và GIMP để tạo ra các ảnh giả có can thiệp chỉnh sửa (như cắt ghép, làm mờ, chỉnh sáng, v.v.)

Chuẩn bị một tập dữ liệu chứa khoảng 12000 ảnh, trong đó, các ảnh gốc và ảnh đã chỉnh sửa sẽ được lưu trong hai thư mục riêng biệt. Thực hiện trích xuất ảnh ELA cho từng ảnh và lưu trữ vào danh sách X_ELA chứa các ảnh ELA dưới dạng numpy array $(H, W, 3)$. Mỗi một lần trích xuất và lưu vào danh sách, chúng ta cần thêm nhãn cho từng ảnh và lưu vào danh sách y, với ảnh gốc sẽ mang nhãn 0 và ảnh đã chỉnh sửa mang nhãn 1. Sau khi thực hiện trích xuất, gắn nhãn và lưu vào danh sách cho 12000 ảnh, chúng ta sẽ chuyển đổi danh sách X_ELA và y thành dạng numpy array để tăng tốc độ xử lý và hỗ trợ xử lý vector.

Dữ liệu trên nhóm em chia làm 3 tập chính để phục vụ huấn luyện và đánh giá mô hình: (i) Tập huấn luyện train-dataset là tập dữ liệu chính để mô hình học các đặc trưng từ ảnh như sự khác biệt về mức độ lỗi giữa ảnh thật và ảnh đã qua chỉnh sửa. (ii) Tập kiểm tra validation-dataset là tập dữ liệu được sử dụng để đánh giá hiệu suất của mô hình trong quá trình huấn luyện, giúp phát hiện các vấn đề như overfitting hoặc underfitting và điều chỉnh các tham số như learning rate. (iii) Tập thử nghiệm test-dataset là tập dữ liệu được giữ riêng để đánh giá hiệu suất cuối cùng của mô hình trên dữ liệu mới chưa từng thấy.

3.2.2 Quy trình trích xuất đặc trưng cho mô hình ELA

Bước 1: Chuẩn bị ảnh gốc

Ảnh đầu vào được chuyển đổi sang không gian màu RGB và thay đổi kích thước về 224×224 pixel để chuẩn hóa dữ liệu cho mô hình học sâu. Kích thước này được lựa chọn vì nó phù hợp với yêu cầu đầu vào của các kiến trúc mạng nơ-ron tích chập (CNN). Ảnh sau khi xử lý được gọi là "*original*".

Bước 2: Nén ảnh

Ảnh gốc được lưu lại dưới dạng JPEG với chất lượng nén mặc định là 90, sau đó mở lại để tạo ảnh nén ("*compressed*").

Bước 3: Tính toán sự khác biệt của ảnh gốc và ảnh nén

Sự khác biệt giữa ảnh gốc (*original*) và ảnh nén (*compressed*) được tính toán bằng cách lấy giá trị tuyệt đối của chênh lệch màu (R, G, B) tại mỗi pixel. Kết quả là một ảnh "*diff*" – ảnh ELA thô, thể hiện mức độ lỗi do quá trình nén lại. Trong đó các giá trị lớn hơn biểu thị mức độ lỗi cao hơn, thường liên quan đến vùng đã bị chỉnh sửa.

Bước 4: Chuẩn hóa và tăng cường độ sáng

Để làm rõ vùng lỗi, ảnh *diff* được xử lý thêm:

(i) Hàm getextrema được sử dụng để tìm giá trị cực đại và cực tiểu trong từng

kênh màu của ảnh *diff*, giúp xác định phạm vi khác biệt lớn nhất.

(ii) Một hệ số *scale* được tính toán dựa trên giá trị cực đại này để kéo giãn các giá trị khác biệt, tăng độ tương phản giữa các vùng.

(iii) Hàm enhance_brightness tăng cường độ sáng của ảnh *diff*, làm nổi bật các vùng có khả năng bị chỉnh sửa.

Bước 5: Đầu ra

Ảnh ELA được tạo ra, trong đó các vùng sáng hơn so với tổng thể biểu thị mức độ lỗi nén cao chênh lệch cho với trung bình lỗi nén trên toàn ảnh, thường là dấu hiệu của chỉnh sửa.

3.2.3 Mô tả về mô hình sử dụng

a, Kiến trúc VGG16

VGG16 bao gồm 16 lớp (13 lớp tích chập và 3 lớp fully connected). Các lớp tích chập trong VGG16 chủ yếu sử dụng kernel 3x3 liên tiếp nhau. Các phép tích chập được thực hiện với stride bằng 1. Các lớp tích chập được tổ chức thành nhiều khối xếp chồng lên nhau. Cụ thể, những lớp đầu tiên trong mạng học các đặc trưng đơn giản như cạnh, góc, màu sắc. Trong khi đó, cá lớp sâu hơn sẽ kết hợp những đặc trưng cơ bản này để nhận diện cấu trúc và đối tượng có độ phức tạp cao hơn. Sau mỗi lớp tích chập, một hàm kích hoạt phi tuyến tính ReLU được áp dụng. Hàm ReLU đóng vài trò quan trọng trong việc giúp mạng học được các mối quan hệ phi tuyến trong dữ liệu, đồng thời giải quyết vấn đề vanishing gradient do đặc tính không bị giới hạn ở miến giá trị dương và đạo hàm không bị thu nhỏ mạnh. Tiếp sau mỗi khối các lớp tích chập là một lớp max-pooling. Chức năng chính của nó là làm giảm kích thước không gian của bản đồ đặc trưng, giúp giảm số lượng tham số và khối lượng tính toán cho các lớp tiếp theo trong mạng.

Các lớp fully connected đảm nhận nhiệm vụ trích xuất các đặc trưng ở cấp độ cao. Các lớp này tổng hợp toàn bộ thông tin đặc trưng đã được học từ những lớp trước đó trong mạng để hình thành nên các biểu diễn đặc trưng mang tính toàn cục hơn. Từ đó, chúng thực hiện chức năng đưa ra dự đoán phân loại.

Dữ liệu đầu vào cho mô hình VGG16 là ảnh được biểu diễn dưới dạng numpy arayy (224,224,3). Lớp đầu ra bao gồm 1000 nơ-ron, mỗi nơ-ron đại diện cho một trong các danh mục thuộc bộ dữ liệu ImageNet, và kết quả đầu ra thể hiện xác suất để ảnh đầu vào được phân loại vào từng danh mục tương ứng

b, Tinh chỉnh và các lớp tùy chỉnh

(i) Ở đây, chúng em sẽ tạm bỏ đi các lớp FC của kiến trúc VGG16 gốc để fine-tuning cho các lớp tích chập.

(ii) Hầu hết các lớp tích chập của VGG16 được giữ nguyên (freeze), mô hình sẽ giữ nguyên các trọng số đã học từ dữ liệu lớn ImageNet. Chúng em chỉ mở 4 lớp cuối để tinh chỉnh, cụ thể là cho phép mô hình sử dụng lan truyền ngược để cập nhật các trọng số. Điều này giúp tận dụng các đặc trưng cấp thấp đã học được (như cạnh, góc) trong khi điều chỉnh các đặc trưng cấp cao cho dữ liệu ELA.

(iii) Đầu vào (sau lớp tích chập VGG16 đã tinh chỉnh) là ảnh ELA dưới dạng numpy array $(224, 224, 3)$ sau khi qua các lớp tích chập.

(iv) Đầu ra (từ các lớp tích chập VGG16 đã tinh chỉnh) là tensor đa chiều (multi-dimensional array) có shape $(7, 7, 512)$.

(v) Vì bỏ đi các lớp FC gốc, chúng em xây dựng lại phần FC cho mô hình:

- Lớp Flatten => Chuyển đổi đầu ra của các lớp tích chập (ví dụ: $(7, 7, 512)$) thành vector 1 chiều.
- Hai lớp Dense => Mỗi lớp có 4096 đơn vị với hàm kích hoạt ReLU.
- Lớp Dropout => Giảm nguy cơ overfitting bằng cách ngẫu nhiên bỏ qua một số đơn vị trong quá trình huấn luyện.
- Lớp đầu ra => Lớp Dense với 1 đơn vị và hàm kích hoạt sigmoid, xuất ra xác suất từ 0 đến 1 để phù hợp cho bài toán phân loại nhị phân.

c, Biên dịch và huấn luyện

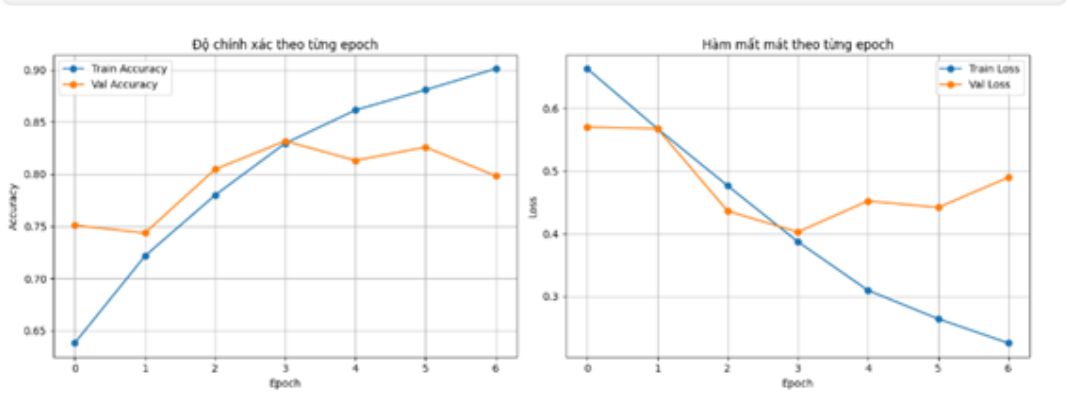
Mô hình dùng bộ tối ưu Adam với learning rate là 0.0001, hàm mất mát binary_crossentropy dành cho phân loại nhị phân, và theo dõi chỉ số độ chính xác (accuracy). Ngoài ra, chúng em còn sử dụng cơ chế EarlyStopping, nó sẽ dừng huấn luyện sớm nếu val_loss không cải thiện sau 3 epoch liên tiếp, nhằm tránh lãng phí tài nguyên và giảm overfitting.

3.2.4 Đánh giá hiệu năng của mô hình ELA

a, Hiệu năng trên tập huấn luyện (train) và tập kiểm tra (validation)

Tính từ epoch 1, train accuracy tăng dần đều, đến epoch 7 đạt $\sim 90.4\%$, loss giảm rõ rệt. Validation accuracy dao động khoảng $74\% - 83\%$, có lúc tăng (epoch 4), nhưng sau đó có xu hướng giảm nhẹ ở các epoch cuối. Loss validation cũng có xu hướng giảm nhưng có một số epoch loss tăng nhẹ.

Từ đó có thể đưa ra nhận xét: (i) Mô hình học khá tốt trên tập train (train accuracy rất cao, loss giảm). (ii) Validation accuracy thấp hơn train, biến động nhẹ, có thể xảy ra tình trạng overfitting nhẹ. (iii) Loss trên tập validation không giảm liên tục, có lúc tăng nhẹ, thể hiện model chưa ổn định hoàn toàn trên dữ liệu chưa thấy.



Hình 3.3: Biểu đồ hiệu năng mô hình ELA

b, Hiệu năng trên tập thử nghiệm (test) qua các ngưỡng Threshold với F1 score

```
59/59 ━━━━━━━━━━━━━━━━━━━━ 13s 209ms/step
Threshold 0.30 → F1: 0.8012
Threshold 0.35 → F1: 0.8302
Threshold 0.40 → F1: 0.8121
Threshold 0.45 → F1: 0.7947
Threshold 0.50 → F1: 0.7773
Threshold 0.55 → F1: 0.7635
Threshold 0.60 → F1: 0.7445
Threshold 0.65 → F1: 0.7296
Threshold 0.70 → F1: 0.7002
```

Hình 3.4: Hiệu năng trên tập test mô hình ELA

F1 score tốt nhất đạt khoảng 0.8302 tại threshold 0.35.

F1 score khá ổn định từ 0.35 đến 0.60, dao động nhẹ.

3.3 Xây dựng web phân tích ảnh

Nhóm chúng em đã phát triển một ứng dụng web nhỏ, có khả năng hỗ trợ phân tích và phân loại hình ảnh. Ứng dụng sẽ thử nghiệm hai phương pháp chính: Error Level Analysis (ELA) để tìm kiếm những dấu hiệu của việc ảnh bị nén hoặc chỉnh sửa, và Photo Response Non-Uniformity (PRNU) để cố gắng nhận diện những đặc trưng riêng của từng thiết bị chụp ảnh. Chúng em hy vọng dự án này sẽ phần nào giúp ích trong việc xác minh thông tin hình ảnh, từ đó giảm thiểu những thông tin sai lệch trong môi trường số.

3.3.1 Kiến trúc phần mềm và công nghệ sử dụng

Phần backend nhóm em sử dụng ngôn ngữ Python và framework Flask, đóng vai trò xử lý các yêu cầu từ người dùng, phân tích ảnh và trả về kết quả

Phần front-end chúng em sử dụng HTML, CSS, JavaScript để tạo giao diện đơn

giản để tương tác với người dùng, cho phép người dùng tải hình ảnh và hiển thị kết quả.

Website sử dụng thư viện PIL để xử lý ảnh cơ bản, numpy để thực hiện các phép tính trên mảng, scikit-image để áp dụng các thuật toán xử lý ảnh phức tạp và Keras để tải và sử dụng các mô hình học máy đã được huấn luyện

3.3.2 Các loại phân tích và kết quả

Website cung cấp ba loại phân tích hình ảnh khác nhau để đáp ứng nhu cầu đa dạng của người dùng.

(i) Loại phân tích đầu tiên là "Phân tích kết hợp" (combined), sử dụng cả hai phương pháp ELA và PRNU để đưa ra một đánh giá toàn diện về hình ảnh. Trong loại phân tích này, hệ thống sẽ phân loại hình ảnh thành một trong bốn dạng: ảnh thật và chưa qua chỉnh sửa, ảnh thật nhưng đã qua chỉnh sửa, ảnh tạo bởi AI, hoặc không thể xác định rõ.

(ii) Loại phân tích thứ hai là "Ảnh gốc vs Ảnh đã chỉnh sửa" (original_vs_edited), chỉ sử dụng phương pháp ELA để xác định xem hình ảnh đã qua chỉnh sửa hay chưa.

(iii) Loại phân tích thứ ba là "Ảnh thật vs Ảnh AI" (real_vs_ai), chỉ sử dụng phương pháp PRNU để phân biệt giữa hình ảnh chụp từ camera thật và hình ảnh được tạo ra bởi AI. Kết quả phân tích được trả về dưới dạng một thông báo dễ hiểu cho người dùng, cùng với các giá trị dự đoán từ các mô hình để người dùng có thể hiểu rõ hơn về cơ sở của kết luận. Các ngưỡng quyết định đã được thiết lập dựa trên nhiều thử nghiệm và đánh giá để đảm bảo độ chính xác cao nhất có thể.

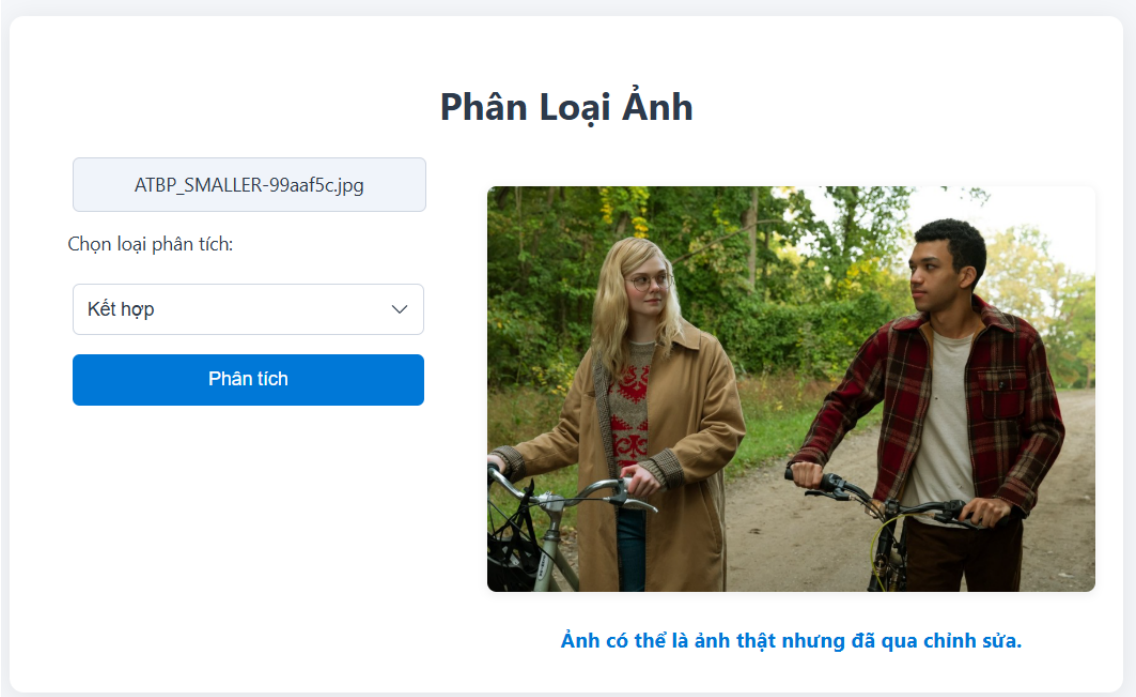
3.3.3 Quy trình xử lý và phân tích ảnh

Khi người dùng tải một hình ảnh lên thông qua giao diện web, quy trình xử lý và phân tích hình ảnh bắt đầu. (1) Hệ thống sẽ kiểm tra tính hợp lệ của file, chỉ chấp nhận các định dạng hình ảnh phổ biến như PNG, JPG và JPEG. (2) Hình ảnh được mở và chuyển đổi sang không gian màu RGB để đảm bảo tính nhất quán trong quá trình xử lý. (3) Hình ảnh sau đó được thay đổi kích thước thành 224x224 pixel, kích thước mà các mô hình học máy đã được huấn luyện để xử lý. Hình ảnh đã được chuẩn hóa được lưu vào thư mục uploads để có thể truy cập sau này và được đưa vào các hàm phân tích. (4.1) Đối với phương pháp ELA, hình ảnh được chuyển đổi thành một biểu diễn ELA sử dụng hàm convert_to_ela, sau đó được đưa vào mô hình ELA đã được huấn luyện để dự đoán. (4.2) Đối với phương pháp PRNU, nhiễu được trích xuất từ hình ảnh sử dụng hàm extract_single với tham số sigma được lấy từ file cấu hình, sau đó được xử lý thông qua các hàm freqq và consis_map để tạo ra các đặc trưng bổ sung. (5) Các đặc trưng này sau đó được đưa vào mô hình PRNU

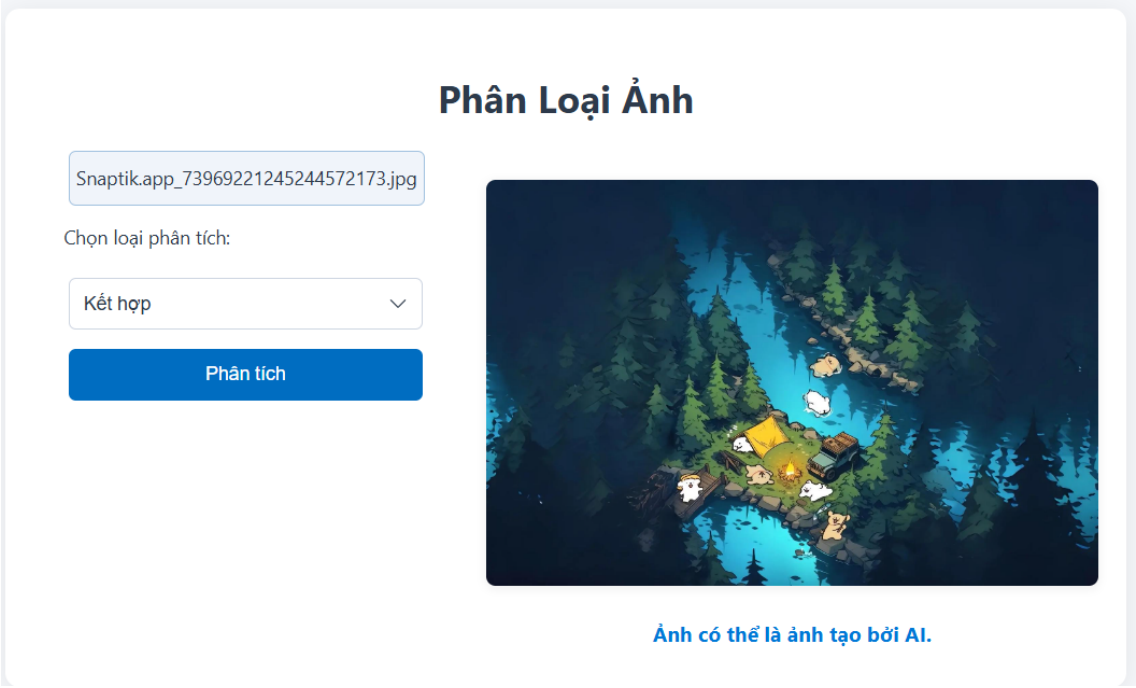
để dự đoán.

Dựa trên loại phân tích mà người dùng đã chọn (combined, original_vs_edited hoặc real_vs_ai), hệ thống sẽ kết hợp các kết quả dự đoán từ hai phương pháp và đưa ra kết luận cuối cùng về hình ảnh.

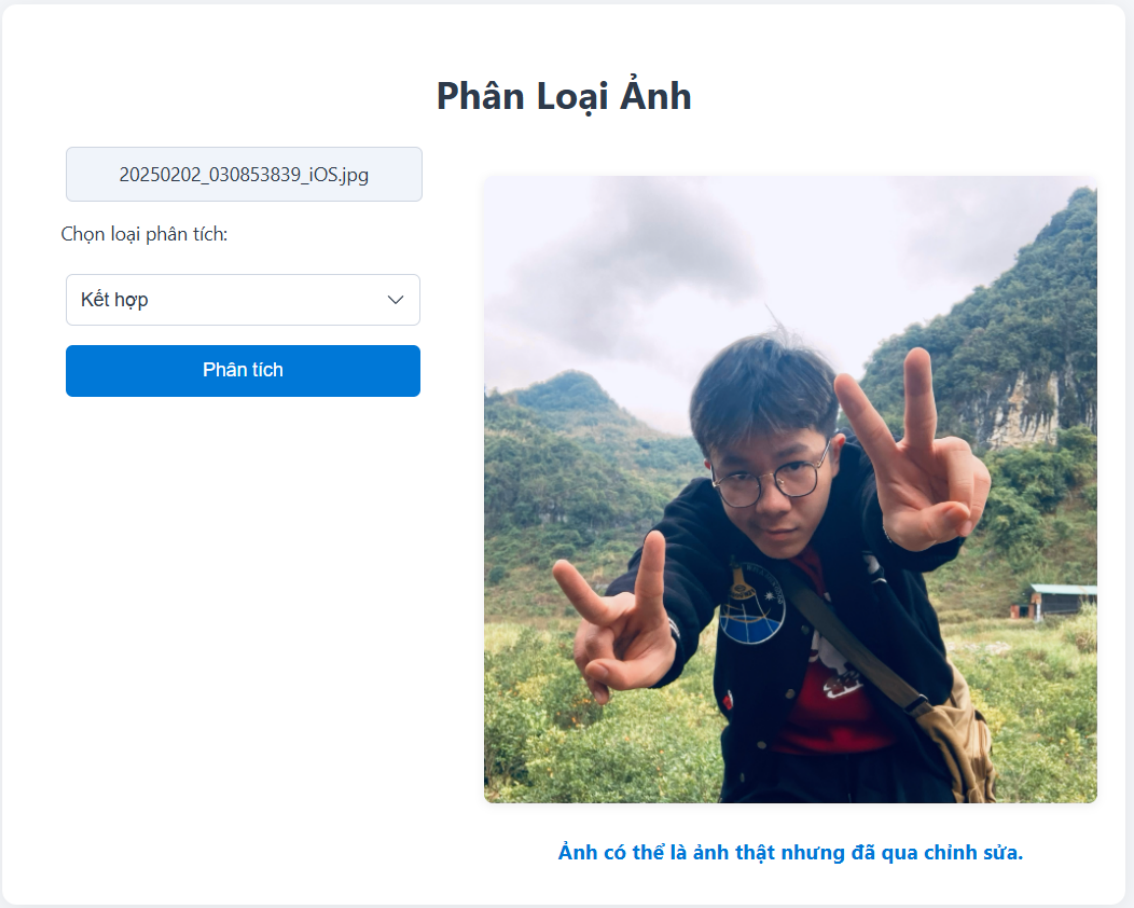
3.3.4 Demo kết quả phân tích của web:



Hình 3.5: Ảnh chụp từ bộ phim đã chỉnh sửa



Hình 3.6: Ảnh hoạt hình



Hình 3.7: Ảnh chụp người đã qua app làm đẹp

CHƯƠNG 4. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

4.1 Kết luận

Đề tài kết hợp mô hình nhiễu dư, Photo Response Non-Uniformity (PRNU) và Error Level Analysis (ELA) của nhóm chúng em sơ khai là ý tưởng muốn tạo một bộ phân loại ảnh thật chụp từ máy ảnh và ảnh tạo bởi các mô hình AI. Để tạo ra được dự án này, nhóm chúng em lấy ý tưởng tham khảo từ nhiều nguồn, nhưng bài báo "Detection of AI-Created Images Using Pixel-Wise Feature Extraction and Convolutional Neural Networks" của tác giả Fernando Martin-Rodriguez , Rocio Garcia-Mojon và Monica Fernandez-Barciela trên trang nghiên cứu Sensors đã truyền cho chúng em nhiều cảm hứng nhất.

Một số **cải tiến** của nhóm chúng em so với ý tưởng trong bài báo trên có thể kể tới:

(i) Trích xuất được thêm các đặc trưng trích xuất từ nhiễu dư của ảnh, cụ thể là đặc trưng tần số từ nhiễu dư và bản đồ nhất quán nhiễu dư, nhằm làm tăng độ tin cậy và độ chính xác cho lớp mô hình phân loại ảnh thật và ảnh AI

(ii) Tìm ra và cân bằng được các tham số ngưỡng quyết định cụ thể cho từng model, từ đó mà xây dựng được một bộ phân loại đa chiều, kết hợp cả hai lớp mô hình. Mỗi sự dự đoán nằm trong ngưỡng quyết định khác nhau mà được tính toán kĩ lưỡng, từ đó mà phân loại đáng tin cậy hơn, chính xác hơn.

Những đóng góp của nhóm chúng em vào dự án này có thể kể tới:

1. Xây dựng quy trình phân loại nguồn gốc ảnh dựa trên trích xuất nhiễu dư ảnh và các đặc trưng từ nhiễu dư của ảnh, thông qua kiến trúc CNN đa đầu vào để phân tích đồng thời nhiều loại đặc trưng nhiễu

   (i) **Dẫn dắt/Giới thiệu vấn đề:**

   Việc phân biệt giữa ảnh được chụp bởi máy ảnh thực và ảnh do AI tạo ra ngày càng trở nên quan trọng. Trong thực tế, ảnh AI và ảnh thật thường có những khác biệt tinh vi trong cấu trúc nhiễu, là những dấu vết thuộc về bản chất quá trình hình thành ảnh. Tuy nhiên, nếu chỉ dựa vào mỗi một đặc trưng đơn lẻ là nhiễu dư của ảnh thì có thể không đủ mạnh mẽ để bao quát hết các biến thể đa dạng của ảnh AI, vốn được tạo ra từ nhiều loại mô hình khác nhau. Thách thức đặt ra là làm sao để xây dựng được một bộ đặc trưng nhiễu đủ phong phú và mang tính phân biệt cao.

   Chúng em chọn ra thêm hai đặc trưng tiềm năng là phổ tần số của nhiễu

và bản đồ nhất quán nhiễu. Tuy nhiên nếu chỉ đơn thuần kết hợp chúng lại và đưa vào một mạng nơ-ron tích chập chúng em nghĩ không phải là cách tiếp cận tối ưu. Do đó, vấn đề tiếp theo là thiết kế một kiến trúc mạng có khả năng học các biểu diễn chuyên biệt từ mỗi loại đặc trưng trước khi tổng hợp chúng lại để đưa ra quyết định phân loại cuối cùng.

Trong quá trình thực hiện, nhóm đã gặp phải một số khó khăn cụ thể. Thứ nhất, sự hiểu biết ban đầu chưa thực sự rõ ràng về sự khác biệt và vai trò của nhiễu dư ảnh và đặc trưng PRNU nói riêng đã dẫn đến một giai đoạn tốn thời gian trong việc định hướng phương pháp trích xuất đặc trưng phù hợp. Thứ hai, giới hạn về phần cứng máy tính cá nhân không đủ mạnh để thực hiện việc trích xuất đặc trưng trên quy mô lớn một cách nhanh chóng, cũng như không đủ để huấn luyện các mô hình học sâu phức tạp với lượng dữ liệu lớn.

(ii) **Giải pháp:**

Để giải quyết những thách thức trên, nhóm đã đề xuất và triển khai một quy trình toàn diện, bao gồm các giải pháp cho từng vấn đề:

Nhóm đã chủ động thu thập một bộ dữ liệu đa dạng. Đối với ảnh AI, dữ liệu được tổng hợp từ nhiều mô hình tạo ảnh khác nhau để bao quát các kỹ thuật tạo sinh phổ biến. Đối với ảnh thật, dữ liệu được lấy từ nhiều nguồn máy ảnh khác nhau, trong đó có một phần đáng kể từ bộ dữ liệu VISION và nhiều bộ dữ liệu lẻ tẻ khác, nhằm đảm bảo sự đa dạng về đặc tính cảm biến.

Để khắc phục hạn chế về phần cứng, nhóm đã chủ động sử dụng các nền tảng cung cấp tài nguyên tính toán trực tuyến như Kaggle Notebooks và Google Colaboratory. Đồng thời, nhóm cũng áp dụng các biện pháp tiết kiệm tài nguyên để có thể hoàn thành các tác vụ tính toán nặng như trích xuất đặc trưng và huấn luyện mô hình mà không vượt quá giới hạn cho phép của các nền tảng này.

Đối với quy trình trích xuất đa đặc trưng nhiễu, thay vì chỉ dựa vào một loại nhiễu, nhóm đã xây dựng quy trình trích xuất đồng thời ba loại đặc trưng từ nhiễu dư của ảnh đầu vào. Sau đó nhóm đã thiết kế một kiến trúc CNN với ba nhánh xử lý song song, mỗi nhánh nhận một loại đặc trưng làm đầu vào để khai thác hiệu quả ba loại đặc trưng trên. Trong đó nhánh xử lý nhiễu dư được thiết kế sâu hơn hai nhánh còn lại và có tích hợp kết nối tắt để học các đặc trưng phức tạp.

(iii) **Kết quả đạt được:**

Quy trình trích xuất đặc trưng và kiến trúc CNN đa đầu vào này đã được huấn luyện và đánh giá trên bộ dữ liệu đã chuẩn bị. Dựa trên kết quả từ quá trình huấn luyện và kiểm định, mô hình tốt nhất đã đạt được các chỉ số hiệu năng đáng khích lệ trên tập kiểm định, trong đó độ chính xác mà mô hình dự đoán được đạt 86.17

Những kết quả này cho thấy việc kết hợp nhiều loại đặc trưng nhiễu và phân tích chúng đồng thời thông qua một kiến trúc CNN đa nhánh là một hướng tiếp cận hiệu quả, giúp mô hình học được các dấu hiệu phân biệt phức tạp giữa ảnh thật và ảnh AI. Việc chủ động giải quyết các thách thức về hiểu biết ban đầu và giới hạn phần cứng cũng là một đóng góp quan trọng trong quá trình thực hiện đề tài.

2. Xây dựng quy trình phát hiện chỉnh sửa ảnh dựa trên trích xuất mức độ lỗi của ảnh

(i) **Dẫn dắt/Giới thiệu vấn đề:**

Không chỉ tìm kiếm nguồn gốc ảnh, việc phát hiện ảnh đã qua chỉnh sửa kỹ thuật số hay chưa cũng là một vấn đề quan trọng không kém. Nhóm em cho rằng, nhiều kỹ thuật chỉnh sửa tinh vi không để lại dấu vết rõ ràng cho mắt thường. Phân tích mức độ lỗi là một phương pháp có khả năng làm nổi bật sự khác biệt trong đặc tính nén JPEG giữa các vùng ảnh gốc và các vùng đã bị can thiệp. Do đó, vấn đề đặt ra là xây dựng một quy trình tự động, dựa trên học máy, để phân tích ảnh ELA và đưa ra dự đoán về tính toàn vẹn của ảnh.

Thêm vào đó, việc huấn luyện một mô hình học sâu từ đầu cho nhiệm vụ này đòi hỏi một lượng lớn dữ liệu ELA đã gán nhãn và tài nguyên tính toán đáng kể. Thách thức cho chúng em là làm sao để xây dựng một mô hình hiệu quả mà không cần một bộ dữ liệu quá khổng lồ hoặc thời gian huấn luyện quá dài.

(ii) **Giải pháp:**

Để giải quyết vấn đề trên, nhóm em triển khai một quy trình dựa trên phân tích ELA kết hợp với học chuyển giao. Cụ thể, đối với mỗi ảnh đầu vào, đặc trưng ELA được trích xuất bằng cách tính toán sự chênh lệch sau khi nén lại ảnh với một mức chất lượng JPEG xác định (cụ thể nhóm em chọn quality=90) và sau đó tăng cường độ sáng của ảnh chênh lệch này. Ảnh ELA thu được, với kích thước chuẩn hóa là (224, 224, 3), sẽ được sử dụng

làm đầu vào cho một mô hình mạng nơ-ron tích chập dựa trên kiến trúc VGG16. Nhóm đã áp dụng kỹ thuật tinh chỉnh, giữ cố định phần lớn các lớp tích chập của VGG16 và chỉ huấn luyện lại bốn lớp cuối cùng, đồng thời thêm vào một đầu phân loại tùy chỉnh gồm các lớp kết nối đầy đủ Dense với hàm kích hoạt ReLU, Dropout và một lớp đầu ra sigmoid duy nhất cho bài toán phân loại nhị phân. Quá trình huấn luyện được thực hiện trên bộ dữ liệu ELA từ CASIA2, áp dụng thêm cả trọng số lớp và cơ chế dừng sớm để tối ưu hiệu năng.

(iii) **Kết quả đạt được:**

Quy trình trích xuất đặc trưng ELA và mô hình VGG16 tinh chỉnh đã được huấn luyện và đánh giá.

Sau quá trình huấn luyện, mô hình tốt nhất đã được chúng em sử dụng để đánh giá trên tập kiểm thử. Để đưa ra quyết định phân loại cuối cùng từ xác suất đầu ra của mô hình sigmoid, chúng em sử dụng các ngưỡng (threshold) từ 0.30 đến 0.70 đã được thử nghiệm để tối ưu hóa điểm F1-score trên tập kiểm thử. Kết quả từ cho thấy F1-score cao nhất đạt được là 0.8302 tại ngưỡng quyết định là 0.35. Kết quả này cho thấy phương pháp ELA kết hợp với học chuyển giao trên kiến trúc VGG16 có khả năng tốt trong việc phát hiện các dấu hiệu chỉnh sửa ảnh.

3. Phát triển hệ thống phân loại ảnh đa trạng thái kết hợp phân tích nhiễu PRNU và phân tích mức độ lỗi (ELA)

(i) **Dẫn dắt/Giới thiệu vấn đề:**

Trong quá trình thực hiện đề tài, nhóm đã đối mặt với hai thách thức chính khi thực hiện kết hợp hai luồng phân tích ảnh. Thứ nhất là sự thiếu định hướng ban đầu trong ý tưởng xây dựng mô hình kết hợp. Ban đầu, nhóm đã cân nhắc việc ghép chung các đặc trưng từ phân tích nhiễu PRNU và đặc trưng từ phân tích ELA vào cùng một mô hình phân loại chung. Tuy nhiên, sau khi nhận thấy cách tiếp cận này có thể không tối ưu do sự khác biệt về bản chất và mục đích của hai loại đặc trưng, nhóm đã xin ý kiến tham khảo từ giảng viên hướng dẫn. Sự hướng dẫn kịp thời đã giúp nhóm định hình lại giải pháp, đó là tách riêng thành hai mô hình chuyên biệt và sau đó xây dựng một cơ chế logic để kết hợp kết quả, điều này nhóm xin chân thành cảm ơn sự chỉ bảo của cô.

Thách thức thứ hai là việc xác định một ngưỡng đáng tin cậy để phân định rõ ràng giữa ảnh thật chưa qua chỉnh sửa và ảnh thật đã bị chỉnh sửa, cũng

như giữa chúng với ảnh AI. Việc các mô hình đưa ra dự đoán dưới dạng xác suất đòi hỏi phải có một phương pháp hợp lý để chuyển đổi các xác suất này thành các nhãn phân loại cụ thể và có ý nghĩa, đặc biệt khi kết hợp kết quả từ hai mô hình khác nhau.

(ii) **Giải pháp:**

Để giải quyết những thách thức trên và hướng tới một hệ thống phân loại ảnh đa trạng thái, nhóm đã đề xuất giải pháp kết hợp thông tin từ hai mô hình phân tích riêng biệt: một mô hình dựa trên các đặc trưng nhiễu dư và PRNU để xác định nguồn gốc ảnh (AI hay thật), và một mô hình dựa trên đặc trưng ELA để phát hiện dấu hiệu chỉnh sửa.

Quy trình tổng thể của hệ thống này bao gồm việc xử lý ảnh đầu vào qua cả hai quy trình trích xuất đặc trưng (nhiễu dư/phổ tần số/bản đồ nhất quán cho mô hình PRNU; và ELA cho mô hình VGG16). Sau đó, mỗi mô hình sẽ đưa ra một dự đoán xác suất:

- *predictionPRNU* -> xác suất ảnh là ảnh thật.
- *predictionELA* -> xác suất ảnh đã qua chỉnh sửa.

Dựa trên kết quả đánh giá F_1-score tốt nhất trên các tập kiểm định/kiểm thử của từng mô hình, nhóm em đã xác định các ngưỡng quyết định phù hợp cho từng dự đoán. Cụ thể, ngưỡng cho *predictionPRNU* được chọn là 0.36 (dựa trên kết quả F_1-score tốt nhất 0.8302 từ hình ảnh phân tích hiệu năng mô hình PRNU) và ngưỡng cho *predictionELA* được chọn là 0.67 (dựa trên thông tin F_1-score tốt, ví dụ 0.8175 tại ngưỡng 0.66 từ hình ảnh phân tích hiệu năng mô hình ELA, hoặc đã được tinh chỉnh thêm).

Dựa trên hai dự đoán xác suất này và các ngưỡng đã chọn, một logic kết hợp đơn giản được áp dụng để đưa ra kết luận về trạng thái của ảnh:

- Nếu *predictionELA* < 0.67 và *predictionPRNU* ≥ 0.36: Kết luận Ảnh có thể là ảnh được chụp bởi máy ảnh và chưa qua chỉnh sửa, cắt ghép!
- Nếu *predictionELA* ≥ 0.67 và *predictionPRNU* ≥ 0.36: Kết luận Ảnh có thể là ảnh được chụp bởi máy ảnh nhưng đã qua chỉnh sửa, cắt ghép!
- Nếu *predictionELA* ≥ 0.67 và *predictionPRNU* < 0.36: Kết luận Ảnh có thể là ảnh được tạo ra bởi AI! (Trường hợp này có thể bao gồm cả ảnh AI đã qua chỉnh sửa).
- Nếu *predictionELA* < 0.67 và *predictionPRNU* < 0.36: Kết luận Ảnh có thể là ảnh được tạo ra bởi AI hoặc đã qua chỉnh sửa nặng! (Trường

hợp này ưu tiên kết luận là AI nếu PRNU thấp, hoặc nếu ảnh AI sạch đến mức ELA không phát hiện được chỉnh sửa).

(iii) **Kết quả đạt được :**

Thay vì chỉ đưa ra kết luận nhị phân, hệ thống có khả năng cung cấp một cái nhìn chi tiết hơn về bản chất của bức ảnh, giúp người dùng hiểu rõ hơn liệu ảnh có nguồn gốc tự nhiên hay nhân tạo, và có dấu hiệu bị can thiệp hay không.

Hệ thống kết hợp được sức mạnh của mô hình PRNU/nhiễu trong việc xác định các dấu vết phần cứng của cảm biến giúp phân biệt nguồn gốc AI/thật và khả năng của ELA trong việc phát hiện các bất thường do quá trình nén lại sau chỉnh sửa.

4. Phát triển ứng dụng web minh họa cho hệ thống phân loại ảnh

(i) **Dẫn dắt/Giới thiệu vấn đề:**

Mặc dù xây dựng các mô hình học máy hiệu quả mới là cốt lõi của đề tài, chúng em thấy có một giao diện trực quan cho phép người dùng cuối tương tác và kiểm chứng kết quả là rất quan trọng để minh họa tính ứng dụng thực tế. Thách thức đặt ra là làm thế nào để tích hợp các mô hình đã huấn luyện, vốn yêu cầu môi trường Python và các thư viện chuyên biệt, vào một ứng dụng web đơn giản, dễ sử dụng, cho phép người dùng tải ảnh lên và nhận kết quả phân loại một cách nhanh chóng. Đồng thời, cần đảm bảo quy trình xử lý ảnh trên backend được thực hiện hiệu quả.

(ii) **Giải pháp:**

Công nghệ sử dụng:

Backend: Ứng dụng được phát triển bằng Python với framework web Flask. Flask được lựa chọn vì tính gọn nhẹ, linh hoạt và dễ dàng tích hợp với các thư viện khoa học dữ liệu và học máy của Python (như NumPy, Pillow, TensorFlow/Keras) vốn đã được sử dụng để xây dựng và huấn luyện các mô hình.

Frontend: Giao diện người dùng được xây dựng bằng HTML, CSS, và JavaScript cơ bản để tạo một trang web cho phép người dùng chọn và tải tệp ảnh lên, sau đó hiển thị kết quả dự đoán.

Kiến trúc và luồng hoạt động:

Để cung cấp một minh chứng trực quan cho hệ thống phân loại ảnh, nhóm chúng em phát triển một ứng dụng web đơn giản sử dụng Python với

framework Flask cho backend và HTML, CSS, JavaScript cơ bản cho frontend. Về kiến trúc và luồng hoạt động, người dùng sẽ chọn và tải lên một tệp ảnh thông qua giao diện web. Ảnh này sau đó được gửi đến backend, nơi các mô hình đã huấn luyện được tải để xử lý. Backend sẽ thực hiện các bước trích xuất đặc trưng tương ứng cho từng mô hình, sau đó đưa các đặc trưng này vào mô hình CNN để nhận dự đoán xác suất. Cuối cùng, dựa trên xác suất và ngưỡng quyết định, kết quả phân loại sẽ được định dạng dưới dạng JSON và trả về để hiển thị trên frontend cho người dùng.

(iii) **Kết quả đạt được:**

Nhóm chúng em đã phát triển thành công một ứng dụng web minh họa cơ bản, cho phép người dùng tương tác trực tiếp với hệ thống phân loại ảnh.

Ứng dụng này không chỉ là một công cụ để kiểm chứng các mô hình đã xây dựng mà còn cho thấy tiềm năng triển khai giải pháp này trong các kịch bản thực tế, nơi người dùng không chuyên cũng có thể nhanh chóng kiểm tra tính xác thực của một hình ảnh.

4.2 Hướng phát triển

Trước tiên, để hoàn thiện các chức năng và nhiệm vụ đã thực hiện, một số công việc cần thiết bao gồm:

(i) **Đánh giá toàn diện hệ thống kết hợp:**

Hiện tại, hai mô hình thành phần (PRNU/Nhiễu và ELA) của nhóm chúng em vẫn đang được đánh giá riêng lẻ => Cần xây dựng một bộ dữ liệu được gán nhãn đa trạng thái chi tiết (ví dụ: "ảnh gốc", "ảnh gốc đã sửa nhẹ", "ảnh gốc đã sửa nặng", "ảnh AI gốc", "ảnh AI đã sửa") để có thể đánh giá một cách chính xác hiệu năng của logic kết hợp và toàn bộ hệ thống phân loại đa trạng thái.

(ii) **Tối ưu hóa ngưỡng quyết định:**

Các ngưỡng hiện tại cho logic kết hợp được chúng em xác định dựa trên F1-score của các mô hình thành phần => Cần nghiên cứu thêm các phương pháp tối ưu hóa ngưỡng một cách hệ thống cho bài toán phân loại đa lớp dựa trên đầu ra xác suất từ hai mô hình, có thể sử dụng các kỹ thuật như tối ưu hóa dựa trên đường cong ROC đa lớp hoặc các hàm mục tiêu phức hợp hơn.

(iii) **Cải thiện tốc độ xử lý:**

Quy trình trích xuất đặc trưng, đặc biệt là các phép tích chập và biến đổi Fourier, chúng em thấy khá tốn thời gian để thực hiện => Cần nghiên cứu các

phương pháp tối ưu hóa mã nguồn hoặc sử dụng các thư viện tính toán hiệu năng cao hơn để giảm độ trễ, đặc biệt quan trọng đối với ứng dụng web.

Sau khi hoàn thiện các khía cạnh trên, nhóm đề xuất các hướng đi mới để cải thiện và nâng cấp hệ thống với các chức năng nâng cao hơn:

(i) Phát triển thêm module dựa trên đặc trưng PRNU không chỉ để phân biệt ảnh thật/AI mà còn có khả năng xác định hoặc phân nhóm loại máy ảnh đã chụp bức ảnh thật. Để có được như vậy đòi hỏi nhóm phải xây dựng một cơ sở dữ liệu PRNU tham chiếu từ nhiều dòng máy ảnh khác nhau.

(ii) Thay vì chỉ kết luận "ảnh AI", một hướng phát triển thú vị là cố gắng xác định xem bức ảnh được tạo ra bởi loại mô hình AI nào (ví dụ: GAN cụ thể, Diffusion model cụ thể). Nhóm có thể dựa trên việc phân tích các vân tay hay artifact đặc trưng mà mỗi kiến trúc AI để lại trong quá trình tạo sinh để thực hiện.

(iii) Đối với các ảnh được phát hiện là đã qua chỉnh sửa, một bước tiến xa hơn là cố gắng xác định hoặc phân loại loại công cụ hoặc ứng dụng chỉnh sửa nào đã được sử dụng (ví dụ: Photoshop, GIMP, ứng dụng di động cụ thể). Chúng em cần phải dựa trên việc phân tích các dấu vết đặc trưng mà mỗi công cụ để lại trong siêu dữ liệu, cấu trúc nén, hoặc các artifact do thuật toán xử lý riêng của chúng.

(iv) Thay vì chỉ phát hiện sự tồn tại của việc chỉnh sửa, có thể phát triển khả năng khoanh vùng chính xác các khu vực đã bị can thiệp trên ảnh.

(v) Nghiên cứu và tích hợp các kỹ thuật để tăng cường tính bền vững của mô hình trước các tấn công, nơi kẻ xấu cố tình tạo ra những thay đổi nhỏ trên ảnh để đánh lừa hệ thống phát hiện.

(vi) Tích hợp các kỹ thuật XAI để cung cấp cho người dùng những giải thích rõ ràng hơn về lý do tại sao mô hình đưa ra một quyết định phân loại cụ thể, thay vì chỉ là một nhãn kết quả để giúp tăng tính minh bạch và độ tin cậy của hệ thống.

(vii) Mở rộng khả năng phân tích sang các định dạng khác như video (ví dụ: phát hiện video deepfake).

Những hướng phát triển này là mong muốn của chúng em không chỉ giúp nâng cao giá trị khoa học và thực tiễn của đề tài mà còn góp phần vào nỗ lực chung trong việc xây dựng một môi trường thông tin số an toàn và đáng tin cậy hơn.

TÀI LIỆU THAM KHẢO

[1] F. Martin-Rodriguez, R. Garcia-Mojon, and M. Fernandez-Barciela, "Detection of AI-created images using pixel-wise feature extraction and convolutional neural networks," *Sensors*, vol. 23, no. 22, p. 9037, 2023. DOI: 10.3390/s23229037. [Online]. Available: https://doi.org/10.3390/s23229037.

[2] S.-Y. Wang, O. Wang, R. Zhang, A. Owens, and A. A. Efros, *CNN-generated images are surprisingly easy to spot... for now*, 2020. arXiv: 1912.11035v2 [cs.CV].

[3] J. Lukáš, J. Fridrich, and M. Goljan, "Digital camera identification from sensor pattern noise," *IEEE Transactions on Information Forensics and Security*, vol. 1, no. 2, pp. 205–214, 2006. DOI: 10.1109/TIFS.2006.873602. [Online]. Available: https://doi.org/10.1109/TIFS.2006.873602.

[4] D. Shullani, M. Fontani, M. Iuliani, O. Al Shaya, and A. Piva, "VISION: A video and image dataset for source identification," *EURASIP Journal on Information Security*, vol. 2017, no. 1, 2017. DOI: 10.1186/s13635-017-0067-2. [Online]. Available: https://doi.org/10.1186/s13635-017-0067-2.

[5] Y. Liu, Y. Xiao, and H. Tian, "Plug-and-play PRNU enhancement algorithm with guided filtering," *Sensors*, vol. 24, no. 23, p. 7701, 2024. DOI: 10.3390/s24237701. [Online]. Available: https://doi.org/10.3390/s24237701.

[6] K. Zhang, W. Zuo, Y. Chen, D. Meng, and L. Zhang, "Beyond a gaussian denoiser: Residual learning of deep CNN for image denoising," *IEEE Transactions on Image Processing*, vol. 26, no. 7, pp. 3142–3155, 2017. DOI: 10.1109/TIP.2017.2662206. [Online]. Available: https://doi.org/10.1109/TIP.2017.2662206.

[7] K. Zhang, W. Zuo, and L. Zhang, *FFDNet: Toward a fast and flexible solution for CNN based image denoising*, 2018. arXiv: 1710.04026v2 [cs.CV].

[8] I. Goodfellow, J. Pouget-Abadie, M. Mirza, *et al.*, "Generative adversarial nets," in *Advances in neural information processing systems 27*, Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, and K. Weinberger, Eds., Curran Associates, Inc., 2014, pp. 2672–2680.

[9] T. Karras, S. Laine, and T. Aila, "A style-based generator architecture for generative adversarial networks," *IEEE Transactions on Pattern Analysis and*

Machine Intelligence, vol. 43, no. 12, pp. 4217–4228, 2021. DOI: 10.1109/TPAMI.2020.2970919.

[10] J. Ho, A. Jain, and P. Abbeel, "Denoising diffusion probabilistic models," in *Advances in Neural Information Processing Systems 33*, H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, Eds., 2020, pp. 6840–6851.

[11] J. Song and S. Ermon, "Generative modeling by estimating gradients of the data distribution," in *Advances in Neural Information Processing Systems 32*, H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché Buc, E. Fox, and R. Garnett, Eds., 2019, pp. 11 917–11 928.

[12] P. Dhariwal and A. Nichol, "Diffusion models beat GANs on image synthesis," in *Advances in Neural Information Processing Systems 34*, M. Ranzato, A. Beygelzimer, Y. Dauphin, P. Liang, and J. Vaughan, Eds., 2021, pp. 8780–8794.

[13] S.-Y. Wang, O. Wang, R. Zhang, A. Owens, and A. A. Efros, "CNN-generated images are surprisingly easy to spot... for now," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020, pp. 8695–8704. DOI: 10.1109/CVPR42600.2020.00872.

[14] J. Frank, T. Eisenhofer, L. Schönherr, A. Fischer, D. Kolossa, and T. Holz, "Leveraging frequency analysis for deep fake image recognition," in *Proceedings of the 37th International Conference on Machine Learning (ICML)*, H. Daumé III and A. Singh, Eds., ser. Proceedings of Machine Learning Research, vol. 119, PMLR, 2020, pp. 3241–3251.

[15] M. Chen, J. Fridrich, M. Goljan, and J. Lukáš, "Determining image origin and integrity using sensor noise," *IEEE Transactions on Information Forensics and Security*, vol. 3, no. 1, pp. 74–90, 2008. DOI: 10.1109/TIFS.2007.916285.

[16] J. Fridrich, "Digital image forensics," *IEEE Signal Processing Magazine*, vol. 26, no. 2, pp. 26–37, 2009. DOI: 10.1109/MSP.2008.931078.

[17] M. Goljan, J. Fridrich, and T. Filler, "Large scale test of sensor fingerprint camera identification," in *Media Forensics and Security XI*, E. J. Delp III, J. Dittmann, N. D. Memon, and P. W. Wong, Eds., International Society for Optics and Photonics (SPIE), vol. 7254, 2009, p. 72540I. DOI: 10.1117/12.805536.

[18] A. Cheddad, J. Condell, K. Curran, and P. Mc Kevitt, "Digital image forensics for identifying computer generated and digital camera images," *International Journal of Digital Crime and Forensics (IJDCF)*, vol. 1, no. 1, pp. 48–69, 2009. DOI: 10.4018/jdcf.2009010103.

[19] F. Marra, D. Gragnaniello, D. Cozzolino, and L. Verdoliva, "Do GANs leave artificial fingerprints?" In *2019 IEEE Conference on Multimedia Information Processing and Retrieval (MIPR)*, IEEE, 2019, pp. 506–511. DOI: 10.1109/MIPR.2019.00098.

[20] N. Krawetz, *Error level analysis*, Hacker Factor Blog, [Online]. Available: http://www.hackerfactor.com/blog/index.php?/archives/1-Error-Level-Analysis.html (visited on May 23, 2025).

[21] Create & Grow, *How many AI images are generated daily? (2024 stats)*, Create & Grow website, [Online]. Available: https://createandgrow.co/blog/how-many-ai-images-are-generated-daily (visited on May 23, 2025), Apr. 2024.

[22] AIPRM, *Generative AI statistics and trends (2024)*, AIPRM website, [Online]. Available: https://www.aiprm.com/generative-ai-statistics/ (visited on May 23, 2025), May 2024.

[23] Presets.io, *How many photos will be taken in 2024?* Presets.io website, [Online]. Available: https://presets.io/blog/how-many-photos-will-be-taken-in-2024 (visited on May 23, 2025).

[24] Image Retouching Lab, *How many photos are taken each day globally? [2024 stats]*, Image Retouching Lab website, [Online]. Available: https://imageretouchinglab.com/how-many-photos-are-taken-each-day/ (visited on May 23, 2025), Jan. 2024.

[25] Market.us, *Photo Editing Software Market Size To Worth USD 1,818 Mn by 2034*, Market.us website, [Online]. Available: https://market.us/report/photo-editing-software-market/ (visited on May 23, 2025), Apr. 2024.

[26] B. Marr, "2024: The Year Of The Deepfake And AI-Powered Fraud And Scams," *Forbes*, Jan. 2024, [Online]. Available: https://www.forbes.com/sites/bernardmarr/2024/01/22/2024-the-year-of-the-deepfake-and-ai-powered-fraud-and-scams/ (visited on May 23, 2025).

PHỤ LỤC

CHƯƠNG A. PHÂN CÔNG CÔNG VIỆC VÀ ĐÁNH GIÁ THÀNH VIÊN

STT	Họ và tên (MSSV)	Nhiệm vụ	Đóng góp (%)
1	Cấn Đức Khôi (B22DCKH069)	• Tìm hiểu và xây dựng mô hình PRNU. • Thu thập dữ liệu và tiền xử lý cho mô hình PRNU. • Huấn luyện, đánh giá và tinh chỉnh mô hình PRNU. • Xây dựng mô hình phân loại kết hợp • Viết báo cáo bằng LaTeX	35%
2	Nguyễn Trường Giang (B22DCKH034)	• Tìm hiểu và xây dựng mô hình ELA. • Thu thập dữ liệu và tiền xử lý cho mô hình ELA. • Huấn luyện, đánh giá và tinh chỉnh mô hình ELA. • Xây dựng và thiết kế web mô hình phân loại kết hợp	35%
3	Đỗ Chí Chương (B22DCKH015)	• Tìm hiểu các mô hình • Thu thập dữ liệu cho cả hai mô hình • Thiết kế web cho mô hình phân loại kết hợp • Chuẩn bị nội dung viết báo cáo	30%