

ĐẠI HỌC QUỐC GIA TP.HCM
TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN

VŨ HOÀI DANH

PHÁT HIỆN HÌNH ẢNH
SINH BỞI MÔ HÌNH TẠO SINH ẢNH

LUẬN VĂN THẠC SĨ

TP. Hồ Chí Minh - Năm 2025

ĐẠI HỌC QUỐC GIA TP.HCM
TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN

VÕ HOÀI DANH

PHÁT HIỆN HÌNH ẢNH
SINH BỞI MÔ HÌNH TẠO SINH ẢNH

Ngành: Trí tuệ nhân tạo

Mã số: 8480107

NGƯỜI HƯỚNG DẪN KHOA HỌC
1. HDC: TS. LÊ TRUNG NGHĨA

TP. Hồ Chí Minh - Năm 2025

LỜI CAM ĐOAN

Tôi xin cam đoan luận văn thạc sĩ ngành Trí tuệ nhân tạo, với đề tài **Phát hiện hình ảnh sinh bởi mô hình tạo sinh** là công trình nghiên cứu do Tôi thực hiện dưới sự hướng dẫn của TS. Lê Trung Nghĩa.

Những kết quả nghiên cứu của luận văn hoàn toàn trung thực và chính xác.

Học viên cao học
(Ký tên, ghi họ tên)

LỜI CẢM ƠN

Tôi xin gửi lời cảm ơn chân thành đến thầy TS. Lê Trung Nghĩa, giảng viên hướng dẫn của tôi, người đã tận tình hướng dẫn và hỗ trợ tôi trong suốt quá trình thực hiện đề tài "*Phát hiện hình ảnh sinh bởi mô hình tạo sinh*". Sự chỉ bảo, kinh nghiệm và kiến thức quý báu của thầy đã giúp tôi vượt qua những khó khăn và hoàn thiện luận văn này.

Tôi rất trân trọng sự nhiệt huyết và sự đồng hành của thầy trong từng bước nghiên cứu, từ việc hình thành ý tưởng cho đến việc thực hiện và hoàn thiện luận văn. Những góp ý và phản hồi của thầy không chỉ giúp tôi nâng cao chất lượng công trình nghiên cứu mà còn là nguồn động viên lớn lao đối với tôi.

Tôi cũng xin trân trọng gửi lời cảm ơn đến các thầy cô tại Trường Đại học Khoa học Tự nhiên - Đại học quốc gia TP.HCM, những người đã tạo điều kiện thuận lợi và cung cấp nền tảng kiến thức vững chắc trong suốt quá trình học tập và nghiên cứu.

MỤC LỤC

LỜI CAM ĐOAN	i
LỜI CẢM ƠN	ii
DANH SÁCH CÁC HÌNH VẼ	vi
DANH SÁCH CÁC BẢNG	ix
DANH SÁCH CÁC THUẬT NGỮ	xi
TRANG THÔNG TIN LUẬN VĂN	xxii
THESIS INFORMATION PAGE	xxv
1 GIỚI THIỆU	1
1.1 Bối cảnh và vấn đề nghiên cứu	1
1.2 Lý do thực hiện đề tài	2
1.2.1 Động lực khoa học	2
1.2.2 Động lực ứng dụng	3
1.3 Mục tiêu nghiên cứu	4
1.4 Phát biểu bài toán	4
1.4.1 Định nghĩa về Ảnh Thật và Ảnh Tạo Sinh	4
1.4.2 Phát biểu hình thức	5

1.4.3	Phương pháp giải bài toán	6
1.5	Thách thức bài toán	7
1.6	Nội dung và phạm vi nghiên cứu	9
1.6.1	Nội dung nghiên cứu	9
1.6.2	Phạm vi nghiên cứu	10
1.7	Đóng góp của luận văn	10
1.8	Cấu trúc của luận văn	11
2	NGHIÊN CỨU LIÊN QUAN	13
2.1	Mô hình tạo sinh ảnh	13
2.1.1	Mô hình GAN	14
2.1.2	Mô hình Diffusion	16
2.2	Phát hiện hình ảnh tạo sinh	19
2.2.1	Nhóm phương pháp tiếp cận trên miền không gian	20
2.2.2	Nhóm phương pháp tiếp cận miền tần số	26
2.2.3	Nhóm phương pháp hỗn hợp	32
3	PHƯƠNG PHÁP ĐỀ XUẤT	34
3.1	Thử nghiệm sơ bộ và hướng tiếp cận	34
3.1.1	Thử nghiệm 1	34
3.1.2	Thử nghiệm 2	35
3.1.3	Các hạn chế khi dùng biến đổi Fourier	37
3.2	Bộ lọc thông cao trên không gian ảnh	37
3.2.1	Bộ lọc trung bình	38
3.3	Xây dựng khối tiền xử lý (Adjacency Difference Orientation Filter (ADOF)) dựa trên bộ lọc thông cao	42

3.4	Mô hình đề xuất	44
3.4.1	Kiến trúc mô hình rút gọn từ ResNet-50	44
3.4.2	Hàm mất mát	47
3.4.3	Quy trình huấn luyện và sử dụng	47
3.5	Tối giản mô hình với kỹ thuật feature-based knowledge distillation	49
3.5.1	Kiến trúc mô hình student	49
3.5.2	Hàm mất mát	50
3.5.3	Quy trình huấn luyện và sử dụng	51
4	THỰC NGHIỆM VÀ ĐÁNH GIÁ KẾT QUẢ	53
4.1	Bộ dữ liệu ForenSynths	53
4.1.1	Thu thập và xử lý hình ảnh	53
4.1.2	Tập train: Ảnh tạo sinh từ mô hình ProGAN	54
4.1.3	Tập validation: Ảnh tạo sinh từ mô hình ProGAN	55
4.1.4	Tập test: Ảnh tạo sinh từ 8 mô hình GAN khác nhau	55
4.1.5	Dữ liệu sử dụng để đánh giá mô hình sau huấn luyện	57
4.2	Cài đặt môi trường thực nghiệm	58
4.2.1	Chuẩn bị dữ liệu	58
4.2.2	Cấu hình phần cứng và cài đặt các tham số	58
4.2.3	Kết quả huấn luyện	59
4.3	Đánh giá mô hình	59
4.3.1	So sánh, đánh giá khả năng phát hiện ảnh tạo sinh từ các phương pháp sinh ảnh khác nhau	60
4.3.2	So sánh và đánh giá về hiệu năng	61
4.4	Kết quả tối giản mô hình student bằng phương pháp feature-based knowledge distillation	63

5 KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN	64
5.1 Kết luận	64
5.2 Hướng phát triển	65
DANH MỤC CÔNG TRÌNH CỦA TÁC GIẢ	66
PHỤ LỤC	72

DANH SÁCH CÁC HÌNH VẼ

1.1	Ảnh tạo sinh (<i>trái</i>) và ảnh thật (<i>phải</i>), được phân biệt dựa vào mức độ chi tiết trên hoa văn của đối tượng. <i>Nguồn:</i> https://elearn.eb.com	8
1.2	Trung bình phổ Fourier của 2,000 hình ảnh từ tập dữ liệu Lsun (<i>trái</i>) và ImageNet (<i>phải</i>), các dấu vết khác nhau giữa mô hình StyleGAN2 và BigGAN thể hiện ở hình 2 và 4 từ trái sang.	9
2.1	Minh họa mô hình GAN.	14
2.2	Đồ thị có hướng mô tả hai quá trình của mô hình Diffusion. <i>Nguồn:</i> [4]	16
2.3	Quá trình khuếch tán thuận trong mô hình DDPMs. <i>Nguồn:</i> [4]	16
2.4	Mô tả thuật toán huấn luyện mô hình Diffusion. <i>Nguồn:</i> [21] . .	18
2.5	Mô tả quá trình sinh ảnh mô hình Diffusion. <i>Nguồn:</i> [21] . . .	19
2.6	Một ví dụ về sự thiếu vắng vùng bảo hoà (vùng đánh dấu màu xanh lá nằm ở 2 bên biểu đồ histogram của ảnh tạo sinh (<i>giữa</i>), trong khi ảnh thật (<i>trái</i>) và (<i>phải</i>) có sự xuất hiện của vùng bảo hoà. <i>Nguồn:</i> [22]	20
2.7	Cấu trúc trọng số của lớp tích chập cuối trong mạng GANs (<i>trái</i>) và Phổ bộ lọc màu của 2 máy ảnh Canon 50D và Canon 40D (<i>phải</i>). <i>Nguồn:</i> [22]	21

2.8	Ảnh gốc (a-b), ảnh biến đổi Gray-scale (c-d), ảnh áp dụng bộ lọc $L0$ (e-f)). <i>Nguồn:</i> [24]	22
2.9	Kiến trúc mô hình Fusing <i>Nguồn:</i> [28]	24
2.10	Mô tả phương pháp Rich Poor Texture Contrast của Nang-Zhong <i>Nguồn:</i> [31]	26
2.11	Tổng quan về quy trình xử lý trong phương pháp của Durall. <i>Nguồn:</i> [34]	27
2.12	Phổ hình ảnh được tạo ra bởi các mạng nơ-ron khác nhau được đào tạo trên tập dữ liệu <i>Stanford dog</i> . <i>Nguồn:</i> [35]	28
2.13	Phổ hình ảnh tương ứng với các kỹ thuật up-sampling khác nhau. <i>Nguồn:</i> [35]	28
2.14	Đường cong độ chính xác theo bước huấn luyện. <i>Nguồn:</i> [35]	29
2.15	Mô tả kiến trúc F3-Net. <i>Nguồn:</i> [38]	30
2.16	So sánh phổ công suất của dữ liệu thực và dữ liệu tạo sinh. <i>Nguồn:</i> [41]	31
2.17	Mô tả phương pháp SAFE [42].	32
3.1	Biểu đồ (<i>phải</i>) thể hiện mức xám dòng thứ 100 của ảnh thật (1) và ảnh tạo sinh (2).	35
3.2	Độ chính xác của hai bộ phân loại trên tập kiểm tra trong quá trình huấn luyện.	36
3.3	Khôi tiền xử lý dựa trên bộ lọc thông cao đã thiết kế	43
3.4	Kiến trúc mạng Resnet-50 rút gọn	45
3.5	Hình ảnh trước và sau khi được xử lý bằng khối ADOF	46
3.6	Quy trình huấn luyện	48
3.7	Quy trình sử dụng	48

3.8	Sơ đồ kiến trúc mô hình Teacher-Student	50
3.9	Quy trình huấn luyện mô hình áp dụng kỹ thuật Knowledge-Distillation	51
4.1	Một vài hình ảnh trong tập dữ liệu huấn luyện (các hình ảnh thật ở hàng trên).	55
4.2	Tổng quan hiệu năng và độ chính xác của một số hướng tiếp cận, trên tập dữ liệu Ojha [76].	60
4.3	Quy trình cơ bản của các phương pháp phát hiện ảnh tạo sinh bằng mạng học sâu	62

DANH SÁCH CÁC BẢNG

2.1	Bảng so sánh kết quả huấn luyện bộ phân loại ứng với từng kĩ thuật tiền xử lý (<i>dòng 3-5</i>), và hiệu suất của của con người (<i>dòng 1</i>) so với mạng học sâu. <i>Nguồn:[24]</i>	23
4.1	Mô tả tập dữ liệu ForenSynths được sử dụng trong quá trình tạo ảnh tổng hợp.	56
4.2	Kết quả đánh giá trên tập kiểm tra ForenSynths	59
4.3	Kết quả đánh giá trên tập Self-Synthesis 9 GANs [52].	61
4.4	Kết quả đánh giá trên tập DiffusionForensics [69].	61
4.5	Kết quả đánh giá trên tập Ojha [76].	61
4.6	Tài nguyên sử dụng và hiệu năng của các phương pháp phát hiện hình ảnh tổng hợp, trên tập dữ liệu DiffusionForensics [69]. Dấu † thể hiện phương pháp đã được huấn luyện trên cùng tập dữ liệu.	62
4.7	So sánh mô hình Teacher và Student trên ba tập dữ liệu.	63

DANH SÁCH CÁC THUẬT NGỮ

adam

Thuật toán tối ưu phổ biến trong huấn luyện mạng nơ-ron, kết hợp giữa momentum và điều chỉnh tốc độ học theo từng tham số. Tên đầy đủ là Adaptive Moment Estimation. 47, 58

ADOF

Adjacency Difference Orientation Filter - khối tiền xử lý dựa trên bộ lọc thông cao trong miền không gian. iv, 6, 34, 42, 45, 47, 49, 60, 64

attention

là cơ chế giúp mô hình học sâu tập trung vào các phần thông tin quan trọng hơn trong chuỗi đầu vào, được sử dụng phổ biến trong mô hình Transformer và các biến thể của nó. 25

azimuthal averaging

kỹ thuật tính trung bình các giá trị theo hướng góc (azimuth) trong hệ tọa độ cực, thường được áp dụng trên ảnh hoặc phổ tần số để rút gọn dữ liệu thành một hàm mật chiềut theo bán kính. 26

backbone

Hay còn gọi *mạng xương sống* trong tiếng Việt, là phần cơ bản của mô hình chịu trách nhiệm trích xuất các đặc trưng từ dữ liệu đầu vào. Các cấu trúc

phổ biến thường sử dụng làm *backbone* trong các nhiệm vụ liên quan đến hình ảnh gồm ResNet, VGG, và EfficientNet. 23–25

batch

Là một tập con của dữ liệu huấn luyện được sử dụng trong một lần lan truyền tiến và lan truyền ngược trong quá trình huấn luyện. Việc chia dữ liệu thành các batch giúp tối ưu hóa bộ nhớ và tăng tốc quá trình huấn luyện thông qua tính toán song song. 48

batch normalization

Kỹ thuật chuẩn hoá dữ liệu đầu vào của từng lớp trong mạng nơ-ron bằng cách đưa chúng về phân phối chuẩn (zero mean, unit variance) trong mỗi mini-batch, giúp tăng tốc quá trình huấn luyện và cải thiện độ ổn định của mô hình. 46

batch size

là số lượng mẫu được sử dụng trong mỗi lần cập nhật tham số trong quá trình huấn luyện. Batch size ảnh hưởng đến tốc độ huấn luyện, bộ nhớ và độ ổn định của mô hình. 59

bilinear

Hay gọi đầy đủ là *Bilinear interpolation* [1]- Nội suy song tuyến tính, là phương pháp được sử dụng phổ biến để phóng to hoặc thu nhỏ ảnh. Giá trị của điểm ảnh mới được tính toán dựa trên trung bình trọng số của 4 điểm ảnh lân cận theo cả trực ngang (x) và dọc (y). 28, 54

binary cross-entropy

là một hàm mất mát được dùng phổ biến trong các bài toán phân lớp nhị phân, giúp đo lường sự khác biệt giữa xác suất dự đoán và nhãn thực tế. 47, 50

binomial up-sampling

là kỹ thuật phóng to ảnh bằng cách chèn thêm điểm ảnh (zero-insertion) và sau đó làm mượt bằng bộ lọc nhị thức (binomial filter). Phương pháp này giúp giảm hiện tượng răng cưa và tạo ảnh mượt mà hơn so với nội suy đơn giản. 28

bottleneck

Một khối trong kiến trúc ResNet gồm ba lớp convolution liên tiếp (1×1 , 3×3 , 1×1), giúp giảm chi phí tính toán mà vẫn giữ được khả năng học đặc trưng. 46, 49

classification

là một dạng bài toán trong học máy, trong đó mô hình học cách phân loại đầu vào vào các nhóm hoặc lớp đã biết. Các loại phân loại gồm phân loại nhị phân, phân loại đa lớp, v.v.. 51

CNN

Mạng nơ-ron tích chập, thường được dùng trong các bài toán xử lý ảnh và nhận dạng mẫu. CNN sử dụng các lớp tích chập để tự động học các đặc trưng không gian từ dữ liệu ảnh đầu vào.. 10, 23, 25, 28, 44

convolution

Toán tử *convolution* [2] là một toán tử được sử dụng trong mạng tích chập, dùng để trích xuất đặc trưng của tín hiệu. 33, 42, 46

DCT

là một phép biến đổi tương tự như biến đổi Fourier, dùng để biểu diễn tín hiệu dưới dạng tổ hợp các hàm cosine với tần số khác nhau. DCT thường được sử dụng trong nén ảnh và xử lý tín hiệu do khả năng tập trung năng lượng tốt. 27, 29

deepfake

là từ ghép giữa “deep learning” (học sâu) và “fake” (giả mạo). Deepfake là nội dung đa phương tiện (ảnh, video, âm thanh) được tạo ra bằng các kỹ thuật học sâu, khiến chúng trông như thật nhưng thực tế là giả mạo. 44, 45

DFT

viết tắt của *Discrete Fourier Transform*, tức Biến đổi Fourier rời rạc. Đây là phép biến đổi dùng để phân tích một tín hiệu rời rạc theo miền tần số. 26, 37

discriminator

trong mô hình GAN, discriminator là mạng học sâu có nhiệm vụ phân biệt dữ liệu thật và dữ liệu do generator tạo ra. 14, 15

distillation

trong lĩnh vực học sâu, distillation (rút trích tri thức) là kỹ thuật huấn luyện một mô hình nhỏ (student) bằng cách học theo hành vi hoặc biểu

diễn của một mô hình lớn đã được huấn luyện trước (teacher). 50, 51

epoch

Một vòng lặp hoàn chỉnh qua toàn bộ tập dữ liệu huấn luyện. Trong một *epoch*, mô hình được cập nhật nhiều lần, mỗi lần với một batch dữ liệu. 48, 59

feature-based knowledge distillation

là một kỹ thuật trong huấn luyện mạng học sâu, trong đó một mô hình nhỏ hơn (student) học từ mô hình lớn hơn (teacher) thông qua đặc trưng trung gian (feature), thay vì chỉ học từ đầu ra cuối cùng. xxiii, xxvi, 10, 34, 49, 63, 64

FFT

viết tắt của *Fast Fourier Transform*, tức Biến đổi Fourier nhanh. Đây là thuật toán hiệu quả để chuyển đổi tín hiệu từ miền thời gian sang miền tần số, thường được dùng trong xử lý tín hiệu và phân tích ảnh. xxiii, xxvi, 26, 30, 37, 38

fully connected layer

là lớp trong mạng nơ-ron trong đó mỗi đầu vào được kết nối với tất cả các nút ở lớp tiếp theo. Lớp này thường xuất hiện ở cuối mô hình để đưa ra quyết định phân loại. 47

gaussian

là một phân phối xác suất liên tục có dạng hình chuông đối xứng, còn gọi là phân phối chuẩn (normal distribution), được sử dụng rộng rãi trong thống kê và học máy để mô hình hóa dữ liệu nhiễu hoặc phân bố tự nhiên. 17

generator

trong mô hình GAN, generator là mạng học sâu có nhiệm vụ sinh ra dữ liệu giả sao cho giống dữ liệu thật nhất có thể, nhằm đánh lừa discriminator. 14, 15

global average pooling

là một kỹ thuật trong mạng nơ-ron tích chập, thay vì dùng các lớp fully connected, GAP tính trung bình toàn bộ mỗi bản đồ đặc trưng (feature map) và tạo ra một đầu ra duy nhất cho mỗi kênh. Phương pháp này giúp giảm số lượng tham số và hạn chế overfitting. 47

GPU

viết tắt của *Graphics Processing Unit*, tức bộ xử lý đồ họa. GPU có khả năng xử lý song song cao, thường được dùng để tăng tốc quá trình huấn luyện mạng học sâu. 37, 38

gray-scale

là dạng ảnh chỉ bao gồm các mức độ xám, mỗi điểm ảnh được biểu diễn bằng một giá trị cường độ ánh sáng, thường từ 0 (đen) đến 255 (trắng). 23

histogram

Là một biểu đồ cột biểu diễn sự phân bố các giá trị màu sắc hoặc mức xám trong một hình ảnh. Biểu đồ này cung cấp một cái nhìn tổng quan về tần suất xuất hiện của các giá trị màu sắc khác nhau trong ảnh, từ đó có thể hiểu rõ hơn về đặc điểm của ảnh, chẳng hạn như độ sáng, độ tương phản, và sự cân bằng màu.. vii, 20, 21

K-means

là thuật toán phân cụm không có giám sát, chia dữ liệu thành K nhóm sao cho khoảng cách nội nhóm được tối thiểu hoá. Thuật toán hoạt động bằng cách cập nhật các tâm cụm và gán lại nhãn lặp đi lặp lại cho đến khi hội tụ. 27

kernel

trong mạng nơ-ron tích chập, kernel (hay còn gọi là filter) là một ma trận trọng số nhỏ dùng để quét qua ảnh đầu vào và trích xuất các đặc trưng như biên, góc cạnh, họa tiết,... 46

learning rate

là siêu tham số quyết định bước nhảy trong quá trình cập nhật tham số mô hình. Giá trị learning rate quá lớn có thể gây dao động, còn quá nhỏ khiến quá trình huấn luyện chậm. 58

Logistic Regression

là một mô hình học có giám sát dùng để giải bài toán phân loại nhị phân, dựa trên hàm sigmoid để ánh xạ đầu ra về khoảng xác suất $[0, 1]$. 27

max pooling

là một phép toán trong mạng nơ-ron tích chập dùng để giảm kích thước đầu ra bằng cách lấy giá trị lớn nhất trong mỗi vùng nhỏ của ảnh đặc trưng. Max Pooling giúp giảm số lượng tham số và tăng tính khái quát của mô hình. 46

MSE

là viết tắt của *Mean Squared Error*, một hàm mất mát phổ biến dùng để đo sai số bình phương trung bình giữa đầu ra dự đoán và nhãn thật trong các mô hình học máy. 50

nearest neighbor

là phương pháp nội suy đơn giản trong xử lý ảnh, trong đó mỗi điểm mới được gán giá trị của điểm lân cận gần nhất. Phương pháp này rất nhanh nhưng có thể tạo ra ảnh bậc thang và thiếu mượt mà. 28

optimizer

là thuật toán tối ưu dùng trong huấn luyện mạng học sâu, nhằm cập nhật các tham số mô hình để giảm hàm mất mát. Một số thuật toán tối ưu phổ biến gồm SGD, Adam, RMSProp. 58

overfitting

Là hiện tượng xảy ra khi mô hình học quá kỹ các chi tiết và nhiễu trong tập huấn luyện, dẫn đến hiệu suất kém khi áp dụng cho dữ liệu chưa từng thấy. Mô hình bị *overfitting* sẽ có sai số huấn luyện thấp nhưng sai số kiểm tra cao. 48

padding

là kỹ thuật thêm các giá trị (thường là 0) xung quanh biên của ảnh đầu vào trong mạng tích chập. Mục đích là để giữ nguyên kích thước không gian đầu ra hoặc kiểm soát kích thước khi thực hiện phép tích chập. 46

patch

một vùng nhỏ hình vuông hoặc hình chữ nhật được trích xuất từ hình ảnh gốc, thường được sử dụng để phân tích cục bộ hoặc huấn luyện mô hình học sâu. 24–26

pipeline

chuỗi các bước xử lý dữ liệu được tổ chức theo thứ tự nhằm thực hiện một tác vụ nhất định, thường được sử dụng trong học máy, xử lý ảnh, hoặc các hệ thống kỹ thuật số tự động. 65

poor texture regions

các vùng ảnh có rất ít thông tin kết cấu, bề mặt tương đồng đều và ít thay đổi về cường độ, khiến việc phát hiện đặc trưng trở nên khó khăn. 25, 26

PyTorch

là một thư viện mã nguồn mở chuyên dùng để xây dựng và huấn luyện các mô hình học sâu, hỗ trợ linh hoạt cả nghiên cứu và triển khai thực tế, phát triển bởi Facebook AI Research. 59

ReLU

viết tắt của *Rectified Linear Unit*, là một hàm kích hoạt trong mạng nơ-ron, được định nghĩa là $f(x) = \max(0, x)$. ReLU giúp mô hình học phi tuyến và có khả năng lan truyền gradient hiệu quả hơn. 46

rich texture regions

các vùng ảnh chứa nhiều thông tin kết cấu, có sự thay đổi rõ rệt về cường độ hoặc hoa văn, giúp trích xuất đặc trưng dễ dàng hơn. 25, 26

sigmoid

là một hàm kích hoạt trong mạng nơ-ron, được định nghĩa là $\sigma(x) = \frac{1}{1+e^{-x}}$.

Hàm sigmoid đưa đầu ra về khoảng (0, 1), thích hợp cho các bài toán phân lớp nhị phân. 47

skip connection

Kết nối bỏ qua; kỹ thuật kết nối đầu vào của một lớp với đầu ra của lớp sâu hơn, thường dùng trong mạng sâu như ResNet để giảm hiện tượng mất mát thông tin và giúp gradient lan truyền dễ dàng hơn.. 44, 46

stride

là một tham số trong lớp tích chập (convolution) hoặc pooling, biểu thị số bước mà cửa sổ trượt di chuyển trên ảnh đầu vào. Stride càng lớn thì kích thước đầu ra càng nhỏ. 46

student

Mô hình nhẹ hơn, được huấn luyện bằng cách học theo đầu ra của mô hình Teacher thông qua lan truyền tri thức. v, x, 49–52, 63, 64

SVM

là một thuật toán học có giám sát được sử dụng để phân loại và hồi quy, hoạt động bằng cách tìm siêu phẳng tối ưu phân tách các lớp dữ liệu khác nhau. 27

teacher

Mô hình được huấn luyện trước, đóng vai trò truyền đạt tri thức cho mô hình Student thông qua cơ chế lan truyền tri thức (Knowledge Distillation). x, 49–52, 63, 64

test

là giai đoạn kiểm tra hiệu suất mô hình sau khi huấn luyện, sử dụng tập dữ liệu chưa từng thấy để đánh giá khả năng tổng quát hóa. v, 55

texture

thuật ngữ trong xử lý ảnh biểu thị các mẫu lặp lại hoặc sự thay đổi cường độ ánh sáng, màu sắc trên bề mặt hình ảnh, dùng để mô tả độ mịn, nhám hoặc cấu trúc vật liệu. 21, 23, 25

train

giai đoạn huấn luyện mô hình, trong đó mô hình học từ tập dữ liệu huấn luyện (training data) để tối ưu hóa các tham số. v, 54, 55, 58

up-sampling

Toán tử *up-sampling* có chức năng tăng kích thước của ảnh bằng cách chèn các điểm mới giữa các điểm hiện có, được dùng phổ biến trong các mô hình tạo sinh, nhằm tăng kích thước của đầu ra so với đầu vào. viii, 27, 28, 33

validation

là quá trình đánh giá mô hình trong quá trình huấn luyện bằng tập dữ liệu riêng biệt, nhằm điều chỉnh siêu tham số và tránh overfitting. v, 55, 58

TRANG THÔNG TIN LUẬN VĂN

Tên đề tài luận văn: Phát hiện hình ảnh sinh bởi mô hình tạo sinh ảnh

Ngành: Trí tuệ nhân tạo

Mã số ngành: 8480107

Họ tên học viên cao học: Võ Hoài Danh

Khóa đào tạo: 32/2022

Người hướng dẫn khoa học: TS. Lê Trung Nghĩa

Cơ sở đào tạo: Trường Đại học Khoa học Tự nhiên, ĐHQG.HCM

1. TÓM TẮT NỘI DUNG LUẬN VĂN

Sự phát triển mạnh mẽ của các mô hình tạo sinh hình ảnh như Generative Adversarial Networks [3] và Diffusion Models [4] đã mở ra nhiều ứng dụng sáng tạo trong đời sống, nhưng đồng thời cũng đặt ra những nguy cơ về việc lan truyền hình ảnh giả mạo. Các hình ảnh được tạo ra bởi mô hình học máy ngày càng trở nên tinh vi, khiến cho việc phát hiện bằng mắt thường trở nên khó khăn. Trong bối cảnh thông tin số được chia sẻ nhanh chóng qua các thiết bị thông minh, việc phát hiện hình ảnh giả mạo trực tiếp trên các thiết bị có cấu hình thấp trở nên ngày càng cấp thiết.

Luận văn tập trung vào bài toán phát hiện hình ảnh tạo sinh thông qua việc kết hợp kỹ thuật tiền xử lý ảnh với mô hình học sâu. Cụ thể, luận văn đề xuất

một bộ lọc đơn giản nhưng hiệu quả, giúp nâng cao hiệu suất, độ chính xác của mô hình học sâu, bộ lọc được áp dụng ở bước tiền xử lý hình ảnh trước khi ảnh được đưa vào mạng phân loại.

Kết quả thực nghiệm cho thấy phương pháp đề xuất đạt hiệu quả vượt trội so với nhiều phương pháp hiện có trong phát hiện ảnh giả tạo sinh. Bên cạnh đó, luận văn cũng xây dựng một kiến trúc mạng đơn giản và áp dụng kỹ thuật rút trích tri thức dựa trên đặc trưng (Feature-Based Knowledge Distillation) để nén mô hình, giúp giảm đáng kể kích thước và tài nguyên tính toán mà vẫn duy trì độ chính xác so với mô hình gốc. Hướng nghiên cứu của luận văn đặc biệt chú trọng đến việc phát triển các mô hình nhỏ gọn, phù hợp để triển khai trên các thiết bị có cấu hình thấp như điện thoại thông minh hoặc thiết bị nhúng.

2. NHỮNG KẾT QUẢ MỚI CỦA LUẬN VĂN

Luận văn đạt được các kết quả chính sau:

1. Đề xuất một khôi tiền xử lý hình ảnh mới giúp cải thiện độ chính xác và giảm chi phí tính toán so với các phương pháp tiền xử lý sử dụng biến đổi Fourier nhanh (FFT), đồng thời vẫn duy trì hiệu suất cao. Phương pháp này thể hiện khả năng tổng quát tốt trên nhiều tập dữ liệu được sinh ra bởi các mô hình tạo sinh khác nhau.
2. Đề xuất một kiến trúc mô hình đơn giản nhưng hiệu quả cao trong bài toán phát hiện hình ảnh tạo sinh.

3. KHẢ NĂNG ỨNG DỤNG TRONG THỰC TIỄN

Phương pháp và kết quả nghiên cứu của luận văn có thể triển khai hiệu quả trong những tình huống thực tế sau đây:

1. Tích hợp vào các nền tảng mạng xã hội hoặc hệ thống quản lý nội dung trực tuyến nhằm hỗ trợ tự động phát hiện và cảnh báo hình ảnh giả mạo do mô hình tạo sinh sinh ra, góp phần hạn chế thông tin sai lệch.
2. Triển khai trên thiết bị cấu hình thấp như điện thoại thông minh để phát hiện hình ảnh, video giả mạo trong ứng dụng gọi video.

THESIS INFORMATION

Thesis title: Detection of Synthetic Images Generated by Generative Models

Speciality: Artificial Intelligence

Speciality code: 8480107

Name of Master Student: Võ Hoài Danh

Academic year: 32/2022

Supervisor: Dr. Lê Trung Nghĩa

At: VNUHCM - University of Science

1. SUMMARY

The rapid development of image generative models such as Generative Adversarial Networks [3] and Diffusion Models [4] has opened up many creative applications in real life, but also raised concerns about the spread of fake images. Machine-generated images have become increasingly sophisticated, making visual detection challenging. In the context of digital information rapidly shared via smart devices, detecting fake images directly on low-resource devices has become increasingly urgent.

This thesis focuses on the problem of detecting generated images by combining image preprocessing techniques with deep learning models. Specifically, the thesis proposes a simple but effective filter that improves the accuracy and

performance of deep learning models, applied during the image preprocessing step before feeding images into the classification network.

Experimental results show that the proposed method achieves superior performance compared to many existing methods in detecting synthetic images. Additionally, the thesis develops a simple network architecture and applies feature-based knowledge distillation technique to compress the model, significantly reducing model size and computational resources while maintaining comparable accuracy to the original model. The research direction particularly emphasizes developing lightweight models suitable for deployment on low-resource devices such as smartphones or embedded systems.

2. NOVELTY OF THESIS

The thesis achieves the following main results:

1. Proposes a novel characteristic filter that improves accuracy and reduces computational cost compared to preprocessing methods using FFT, while maintaining high performance. This method demonstrates good generalization across multiple datasets generated by various generative models.
2. Proposes a simple yet highly effective model architecture for the problem of detecting generated images.

3. PRACTICAL APPLICATION POTENTIAL

The methods and results of this thesis can be effectively applied in the following real-world scenarios:

1. Integration into social media platforms or online content management systems to automatically detect and flag AI-generated fake images, thereby

- helping to reduce misinformation.
2. Deployment on low-resource devices such as smartphones to detect fake images and videos in video calling applications.

CHƯƠNG 1. GIỚI THIỆU

Chương này nhằm giới thiệu tổng quan về đề tài phát hiện hình ảnh tạo sinh, một vấn đề nổi bật trong lĩnh vực thị giác máy tính hiện đại. Mở đầu chương trình bày bối cảnh và vấn đề nghiên cứu trong bối cảnh sự phát triển nhanh chóng của các mô hình tạo sinh ảnh như GAN [3] và Diffusion [4]. Tiếp theo, chương làm rõ động lực nghiên cứu cả về mặt khoa học và ứng dụng thực tiễn, từ đó xác lập mục tiêu nghiên cứu, phát biểu hình thức của bài toán, các thách thức kỹ thuật cần giải quyết, và phạm vi triển khai. Cuối cùng, chương trình bày những đóng góp chính của luận văn như một nền tảng định hướng cho các chương tiếp theo.

1.1 Bối cảnh và vấn đề nghiên cứu

Sự phát triển vượt bậc của trí tuệ nhân tạo, đặc biệt trong lĩnh vực học sâu, đã thúc đẩy sự ra đời của các mô hình tạo sinh hình ảnh ngày càng tinh vi và chính xác. Những mô hình như Generative Adversarial Networks [3] và Diffusion [4] có khả năng tạo ra hình ảnh có chất lượng gần như tương đương với hình ảnh thật, gây khó khăn đáng kể trong việc phân biệt bằng mắt thường. Đây vừa là một bước tiến đột phá trong công nghệ xử lý ảnh, vừa đặt ra những thách thức lớn về mặt nhận diện, xác thực và quản lý thông tin.

Vấn đề đặt ra là: khi ranh giới giữa ảnh thật và ảnh tạo sinh ngày càng trở

nên mờ nhạt, làm thế nào để các hệ thống tự động có thể phân biệt chính xác hình ảnh do con người chụp với hình ảnh do máy sinh ra? Câu hỏi này đặc biệt cấp thiết trong bối cảnh ảnh tạo sinh có thể bị lợi dụng để phát tán thông tin sai lệch, giả mạo nhân thân, hoặc phục vụ các mục đích phi đạo đức và phi pháp khác.

Từ đó, bài toán phát hiện hình ảnh tạo sinh trở thành một hướng nghiên cứu quan trọng và mang tính thời sự cao. Luận văn này tập trung vào việc phân tích, đánh giá và đề xuất phương pháp phát hiện ảnh tạo sinh, nhằm góp phần vào nỗ lực xây dựng các hệ thống nhận diện hình ảnh tin cậy, đáp ứng cả yêu cầu học thuật lẫn ứng dụng thực tiễn trong bối cảnh phát triển nhanh của công nghệ tạo sinh hình ảnh.

1.2 Lý do thực hiện đề tài

1.2.1 Động lực khoa học

Phát hiện hình ảnh tạo sinh là một trong những nhiệm vụ khó khăn trong lĩnh vực thị giác máy tính, nhiệm vụ này đặt trọng tâm vào việc phân tích và nhận diện các đặc điểm không tự nhiên trong ảnh – những dấu hiệu có thể chỉ ra sự can thiệp của các mô hình tổng hợp ảnh.

Sự phát triển nhanh chóng của các mô hình tạo sinh ảnh, đặc biệt là GANs [3] và Diffusion Models [4], đã tạo ra các hình ảnh có mức độ chân thực ngày càng cao. Điều này đặt ra nhu cầu cấp thiết cho cộng đồng nghiên cứu trong việc phát triển các phương pháp phát hiện ngày càng tinh vi hơn, có khả năng thích ứng với sự thay đổi liên tục của các kỹ thuật tạo sinh.

Từ góc nhìn khoa học, việc nghiên cứu phát hiện hình ảnh tạo sinh không chỉ giúp nâng cao hiểu biết về cấu trúc và tính chất của dữ liệu hình ảnh, mà

còn góp phần vào việc phát triển các mô hình học sâu có khả năng khái quát tốt hơn. Một thách thức lớn hiện nay là hầu hết các phương pháp phát hiện chỉ hoạt động hiệu quả trên ảnh được tạo ra từ những mô hình tương tự với mô hình đã thấy trong giai đoạn huấn luyện. Do đó, việc đề xuất các phương pháp phát hiện có tính tổng quát cao là một vấn đề khoa học quan trọng, có thể thúc đẩy sự hiểu biết sâu sắc hơn về mối quan hệ giữa mô hình sinh và đặc trưng ảnh.

Tóm lại, việc nghiên cứu và hoàn thiện các phương pháp phát hiện ảnh tạo sinh không chỉ góp phần nâng cao tính minh bạch và độ tin cậy của công nghệ tạo sinh hình ảnh, mà còn đóng vai trò quan trọng trong việc đảm bảo an toàn thông tin và bảo vệ hệ thống thị giác máy tính trước các nguy cơ giả mạo.

1.2.2 Động lực ứng dụng

Trong những năm gần đây, các mô hình tạo sinh hình ảnh đã có những bước tiến vượt bậc và được ứng dụng rộng rãi trong thực tiễn. Nổi bật trong số đó là các mô hình Generative Adversarial Networks (GANs) [3] và Diffusion [4], đặc biệt là các phiên bản mới như Stable Diffusion 3 [5]. Các mô hình này có khả năng sinh ra hình ảnh chất lượng cao, gần giống ảnh thật đến mức khó phân biệt bằng mắt thường.

Các nền tảng ứng dụng như DALL-E [6], DeepArt [7] hiện đang được sử dụng hiệu quả trong nhiều lĩnh vực như thiết kế đồ họa [8, 9], thời trang [10], nội thất [11], và sáng tác nghệ thuật [12], mang lại những giá trị sáng tạo và hiệu quả vượt trội.

Tuy nhiên, song song với các ứng dụng tích cực, công nghệ này cũng bị lợi dụng cho những mục đích tiêu cực [13], chẳng hạn như tạo ra hình ảnh giả để lừa đảo [14], bôi nhọ danh dự cá nhân [15], hoặc thao túng dư luận. Trước thực

trạng này, một số quốc gia đã ban hành các quy định và chế tài để kiểm soát việc phát tán hình ảnh giả mạo [16].

Chất lượng hình ảnh do các mô hình tạo sinh tạo ra ngày càng tinh vi khiến việc phân biệt ảnh thật và ảnh tạo sinh bằng mắt thường trở nên khó khăn [17]. Hơn nữa, việc tiếp cận và sử dụng các công cụ AI tạo ảnh hiện nay rất đơn giản, cho phép bất kỳ ai cũng có thể tạo ra số lượng lớn hình ảnh giả chỉ trong thời gian ngắn.

Trong bối cảnh đó, việc phát triển các giải pháp có khả năng phát hiện ảnh tạo sinh một cách hiệu quả và tự động là hết sức cần thiết. Đặc biệt, các giải pháp này cần đảm bảo khả năng triển khai trên những thiết bị phổ thông như điện thoại thông minh hoặc máy tính bảng, vốn có tài nguyên phần cứng hạn chế, nhằm hạn chế sự lan truyền của thông tin sai lệch trên quy mô lớn.

1.3 Mục tiêu nghiên cứu

Phát triển phương pháp phân biệt ảnh tạo sinh đạt được hai mục tiêu sau:

- Phương pháp có hiệu quả trên nhiều loại mô hình tạo sinh khác nhau
- Yêu cầu sức mạnh tính toán và lưu trữ thấp, tốc độ nhanh, phù hợp triển khai trên những thiết bị có cấu hình, tài nguyên hạn chế.

1.4 Phát biểu bài toán

1.4.1 Định nghĩa vềẢnh Thật vàẢnh Tạo Sinh

Trong khuôn khổ của đề tài, ta định nghĩa và phân biệt hai loại hình ảnh (2 lớp đối tượng) của bài toán:

- **Ảnh Tạo Sinh (ảnh giả mạo):** Là những hình ảnh được tạo ra bởi mô hình GAN [3], mô hình Diffusion [4], hoặc bất kỳ mô hình tạo sinh khác.

Nội dung của hình ảnh tạo sinh là mô phỏng theo các đối tượng từ thế giới thật hoặc có thể là một đối tượng mới không có thật.

- **Ảnh Thật:** Là những hình ảnh được chụp từ thực tế bằng máy ảnh hoặc các thiết bị thu hình khác. Những hình ảnh này phản ánh chân thực các đối tượng của thế giới thực, bao gồm cả ảnh chụp các tác phẩm nghệ thuật, hoặc ảnh chụp lại một hình ảnh tạo sinh.

1.4.2 Phát biểu hình thức

Phát hiện hình ảnh tạo sinh là bài toán xác định một hình ảnh là *ảnh thật* được tạo ra bằng thiết bị thu hình như máy ảnh, máy quay phim, hay *ảnh tạo sinh* được tạo ra bằng cách sử dụng các mô hình tạo sinh như GAN [3] hoặc Diffusion [4]. Trong luận văn này, bài toán phát hiện hình ảnh tạo sinh được định hình như một bài toán phân loại trong lĩnh vực thị giác máy tính và được mô tả cụ thể như sau:

Đầu vào: Ảnh đầu vào $I \in \mathbb{R}^{w \times h \times c}$, trong đó w, h, c tương ứng chiều rộng, chiều cao và số lượng kênh màu của hình ảnh.

Đầu ra: Là kết quả dự đoán thể hiện ảnh đầu vào I là ảnh thật hay ảnh tạo sinh.

$$y = f(I) \quad (1.1)$$

Với $y \in \{0, 1\}$ là nhãn phân loại của ảnh I , $f(\cdot)$ là bộ phân loại.

- Nếu $y = 0$, thì I được phân loại là ảnh thật.
- Nếu $y = 1$, thì I được phân loại là ảnh tạo sinh.

1.4.3 Phương pháp giải bài toán

Bài toán phát hiện hình ảnh tạo sinh được tiếp cận dưới dạng bài toán phân loại nhị phân, trong đó mô hình học sâu cần phân biệt giữa ảnh thật (do máy ảnh chụp) và ảnh tạo sinh (được sinh ra bởi các mô hình tạo sinh như GAN [3] hoặc Diffusion [4]). Phương pháp được đề xuất trong luận văn gồm hai giai đoạn chính: **tiền xử lý ảnh đầu vào và huấn luyện bộ phân lớp học sâu**.

Trước khi đưa ảnh vào mô hình học sâu, ảnh đầu vào được cho qua khối tiền xử lý ADOF mà luận văn đề xuất nhằm làm nổi bật các chi tiết vi mô hoặc những nhiễu đặc trưng mà các mô hình tạo sinh thường để lại.

Ảnh sau khi được xử lý sẽ giữ lại nhiều đặc trưng hữu ích cho quá trình học, đồng thời giảm nhiễu từ nền ảnh hoặc cấu trúc toàn cục. Đây là bước quan trọng giúp tăng độ nhạy của mô hình đối với các tín hiệu tinh vi mà ảnh tạo sinh thường để lộ.

Sau bước tiền xử lý, ảnh được đưa vào mô hình máy học $f(\cdot)$ - một hàm phân lớp có đầu vào là ảnh đã xử lý và đầu ra là xác suất ảnh thuộc lớp giả mạo. Các bước cụ thể được mô tả như sau:

$$\mathbf{x}_i^{\text{filtered}} = \mathcal{F}(\mathbf{x}_i), \quad \hat{y}_i = f(\mathbf{x}_i^{\text{filtered}})$$

Trong đó, $\mathbf{x}_i^{\text{filtered}}$ là hình ảnh sau khi áp dụng bộ lọc thông cao, $\mathcal{F}(\cdot)$ là phép lọc không gian tương đương với lọc thông cao mà luận văn xây dựng, \mathbf{x}_i là ảnh gốc, và $\hat{y}_i \in [0, 1]$ là xác suất đầu ra.

Quá trình huấn luyện nhằm tìm bộ tham số θ của mô hình sao cho hàm mất mát cross-entropy [18] đạt giá trị nhỏ nhất:

$$\hat{\theta} = \arg \min_{\theta} \mathcal{L}(\theta)$$

Với hàm mất mát:

$$\mathcal{L}(\theta) = -\frac{1}{N} \sum_{i=1}^N [y_i \log(\hat{y}_i) + (1 - y_i) \log(1 - \hat{y}_i)]$$

Trong đó:

- N : số lượng mẫu huấn luyện,
- $y_i \in \{0, 1\}$: nhãn thực tế, với 0 là ảnh thật và 1 là ảnh tạo sinh,
- $\hat{y}_i = f(\mathcal{F}(\mathbf{x}_i))$: xác suất mô hình dự đoán ảnh là giả sau tiền xử lý.

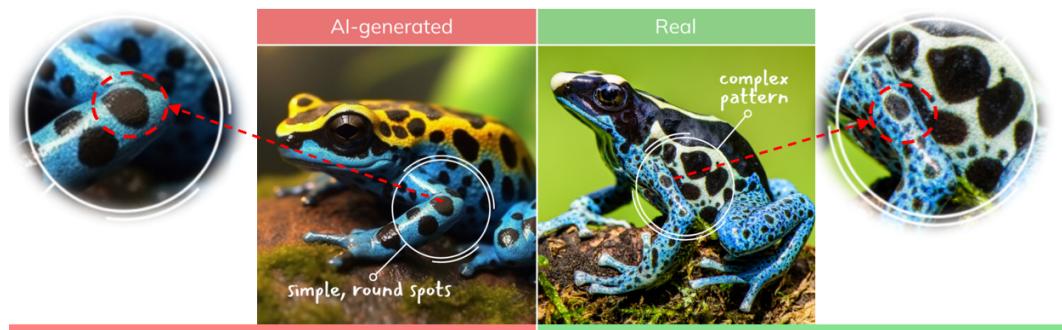
Việc kết hợp kỹ thuật tiền xử lý với mô hình học sâu giúp khai thác được cả đặc trưng miền không gian và đặc trưng tần số, từ đó nâng cao khả năng phát hiện ảnh tạo sinh, đặc biệt trong các trường hợp khó nhận biết bằng mắt thường.

1.5 Thách thức bài toán

Tốc độ phát triển của công nghệ tạo ảnh bằng mạng học sâu làm cho các phương pháp phân biệt giữa ảnh thật và ảnh tạo sinh nhanh chóng lỗi thời, kém hiệu quả trên các mô hình tạo sinh mới:

- Khó khăn với nhóm phương pháp phát hiện ảnh tạo sinh dựa trên sự không đồng nhất ở cấp độ ngữ nghĩa hình ảnh: Hướng tiếp cận này dựa vào việc phát hiện các bất hợp lý về ngữ nghĩa, màu sắc, hình dạng, hoặc những điểm mâu thuẫn với quy luật vật lý của đối tượng, trong các hình ảnh tạo sinh. Tuy nhiên, chất lượng hình ảnh tạo sinh hiện nay đã được nâng cao đáng kể so với những ngày đầu, làm cho việc áp dụng phương pháp này trở nên ngày càng khó khăn hơn (Hình 1.1).

Bên cạnh đó để huấn luyện mô hình "*hiểu*" được các điểm bất hợp lý là

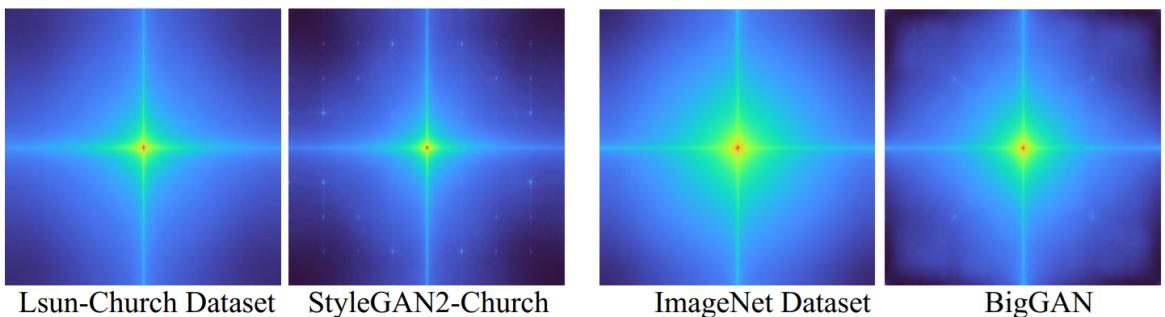


Hình 1.1: Ảnh tạo sinh (*trái*) và ảnh thật (*phải*), được phân biệt dựa vào mức độ chi tiết trên hoa văn của đối tượng. Nguồn: <https://elearn.eb.com>

bài toán khó, yêu cầu dữ liệu huấn luyện lớn và việc gán nhãn vị trí bất hợp lý trên hình ảnh cần nhiều chi phí, vì vậy hướng tiếp cận này ít được sử dụng hiện nay, tuy nhiên đây có thể là hướng phát triển tiềm năng khi kết hợp với mô hình ngôn ngữ lớn vì nó cho đưa ra được giải thích cho kết quả dự đoán.

- Phát hiện ảnh tạo sinh dựa vào phân tích, rút trích các đặc trưng tần số hay đặc trưng không gian trên hình ảnh. Hướng tiếp cận này mặc dù đem lại kết quả cao, ít phụ thuộc vào ngữ nghĩa của hình ảnh, tuy nhiên các kiến trúc mô hình khác nhau sẽ tạo ra những dấu vết khác nhau, do đó, thách thức lớn trong việc tìm ra đặc trưng chung và có hiệu quả trên nhiều mô hình tạo sinh (Hình 1.2).

Ảnh tạo sinh rất đa dạng về nội dung, hình thức và được tạo ra theo trí tưởng tượng vô hạn của người dùng. Các hướng tiếp cận cho độ chính xác cao hiện nay, đều tận dụng sức mạnh của kỹ thuật học sâu. Tuy nhiên, dữ liệu dùng để huấn luyện các mô hình này không đủ đại diện cho toàn bộ ảnh tạo sinh, cụ thể dữ liệu huấn luyện chỉ chứa một số lượng hữu hạn các đối tượng trong thế giới thực, nhưng trong thực tế các đối tượng trong ảnh tạo sinh có thể nằm ngoài tập huấn luyện, đây là một khó khăn cơ bản của bài toán này.



Hình 1.2: Trung bình phổ Fourier của 2,000 hình ảnh từ tập dữ liệu Lsun (*trái*) và ImageNet (*phải*), các dấu vết khác nhau giữa mô hình StyleGAN2 và BigGAN thể hiện ở hình 2 và 4 từ trái sang.

Khó khăn trong việc trả lời câu hỏi "*mô hình đã đưa vào đâu để đưa ra kết luận?*", đây là khó khăn chung của việc ứng dụng kỹ thuật học sâu, và trong nhiệm vụ phát hiện ảnh tạo sinh thì yêu cầu tính "*giải thích được*" càng quan trọng. Các đặc điểm thể hiện một hình ảnh là ảnh tạo sinh thường không thể nhận biết một cách trực quan mà là qua sự tổng hợp rút trích đặc trưng của mạng học sâu, do tính chất phức tạp của các mô hình này, việc cung cấp giải thích rõ ràng và minh bạch cho các quyết định của mô hình là một thách thức lớn.

1.6 Nội dung và phạm vi nghiên cứu

1.6.1 Nội dung nghiên cứu

Luận văn này tập trung nghiên cứu khả năng phát hiện hình ảnh tạo sinh thông qua các phương pháp có độ phức tạp thấp, phù hợp triển khai trên các hệ thống hạn chế về tài nguyên tính toán. Cụ thể, một kỹ thuật tiền xử lý hình ảnh dựa trên bộ lọc thông cao trong miền không gian được thiết kế với cấu trúc đơn giản nhưng hiệu quả, nhằm làm nổi bật các tín hiệu tần số cao – đặc trưng thường gặp trong ảnh tạo sinh. Kỹ thuật này sau đó được kết hợp với một mô hình học sâu nhẹ, giúp tăng cường khả năng phân biệt giữa ảnh thật và ảnh tạo sinh mà

không đòi hỏi chi phí tính toán cao.

Đầu tiên, luận văn khảo sát hai mô hình sinh ảnh phổ biến là Generative Adversarial Networks [3] và Diffusion Models [4] để phân tích các đặc trưng cấu trúc của ảnh tạo sinh. Tiếp theo, luận văn phân loại các phương pháp phát hiện ảnh tạo sinh trước đây theo miền xử lý: miền không gian và miền tần số. Trên cơ sở đó, thiết kế một thực nghiệm để đánh giá hiệu quả của bộ lọc thông cao do luận văn đề xuất. Ảnh sau khi tiền xử lý được đưa vào Convolutional Neural Network (CNN) nhỏ gọn để huấn luyện và kiểm thử trên tập dữ liệu tổng hợp từ nhiều nguồn ảnh thật và ảnh tạo sinh. Kết quả thực nghiệm cho thấy phương pháp đề xuất đạt hiệu quả phân loại cao trong khi vẫn đảm bảo mức độ nhẹ của mô hình. Cuối cùng, kỹ thuật feature-based knowledge distillation được áp dụng để nén mô hình trở nên đơn giản hơn nữa mà vẫn giữ nguyên hiệu năng, hướng đến khả năng triển khai thực tế trong các môi trường hạn chế tài nguyên.

1.6.2 Phạm vi nghiên cứu

Luận văn này chỉ tập trung vào xử lý ảnh tĩnh, không xem xét các dạng dữ liệu động như video hoặc ảnh động. Các ảnh tạo sinh được nghiên cứu trong phạm vi này được tạo ra từ những mô hình tạo sinh phổ biến hiện nay, bao gồm Generative Adversarial Networks (GAN) [3] và Diffusion Models [4]. Ngoài ra, nội dung nghiên cứu cũng tập trung vào việc khai thác hiệu quả của các phương pháp tiền xử lý hình ảnh, cụ thể trên hai miền chính: miền không gian và miền tần số.

1.7 Đóng góp của luận văn

Luận văn có những đóng góp cơ bản sau:

- Đề xuất khối tiền xử lý sử dụng bộ lọc thông cao, làm tăng độ chính xác của mô hình, tăng tốc độ hội tụ trong quá trình huấn luyện mô hình, đồng thời phương pháp này yêu cầu số lượng phép tính nhỏ hơn nhiều phương pháp khác nhưng vẫn cho độ chính xác cao.
- Xây dựng kiến trúc mô hình đơn giản nhưng có hiệu quả cao khi kết hợp với khối tiền xử lý mà luận văn đề xuất. Bộ phân loại sau khi huấn luyện có khả năng hoạt động tốt trên nhiều tập dữ liệu được sinh bởi nhiều loại mô hình tạo sinh khác nhau.

1.8 Cấu trúc của luận văn

Luận văn được tổ chức thành 5 chương như sau:

- **Chương 1 – Giới thiệu:** Trình bày bối cảnh, động lực nghiên cứu, mục tiêu, phạm vi và phát biểu bài toán. Chương cũng nêu rõ các thách thức và đóng góp chính của luận văn.
- **Chương 2 – Nghiên cứu liên quan:** Tổng quan các mô hình tạo sinh ảnh tiêu biểu và phân tích các nhóm phương pháp phát hiện hình ảnh tạo sinh, bao gồm phương pháp miền không gian, miền tần số và kết hợp.
- **Chương 3 – Phương pháp đề xuất:** Mô tả chi tiết quy trình tiền xử lý ảnh, kiến trúc mô hình học sâu, hàm mất mát sử dụng và quy trình huấn luyện mô hình.
- **Chương 4 – Thực nghiệm và đánh giá:** Trình bày các tập dữ liệu sử dụng, thiết lập thí nghiệm, các chỉ số đánh giá và kết quả so sánh với các phương pháp khác.

- **Chương 5 – Kết luận và hướng phát triển:** Tổng kết các kết quả đạt được và đề xuất một số hướng nghiên cứu mở trong tương lai.

CHƯƠNG 2. NGHIÊN CỨU LIÊN QUAN

Chương này trình bày tổng quan về các hướng nghiên cứu có liên quan đến đề tài, bao gồm hai phần chính. Phần đầu giới thiệu các mô hình tạo sinh hình ảnh tiêu biểu như GAN [3] và Diffusion [4], là cơ sở sinh ra các dữ liệu giả mạo ngày càng tinh vi. Phần tiếp theo tổng hợp các phương pháp phát hiện hình ảnh tạo sinh, được phân loại theo miền xử lý: miền không gian, miền tần số và phương pháp kết hợp. Việc phân tích các phương pháp hiện có giúp làm rõ thách thức kỹ thuật và định hướng xây dựng giải pháp phù hợp trong luận văn.

2.1 Mô hình tạo sinh ảnh

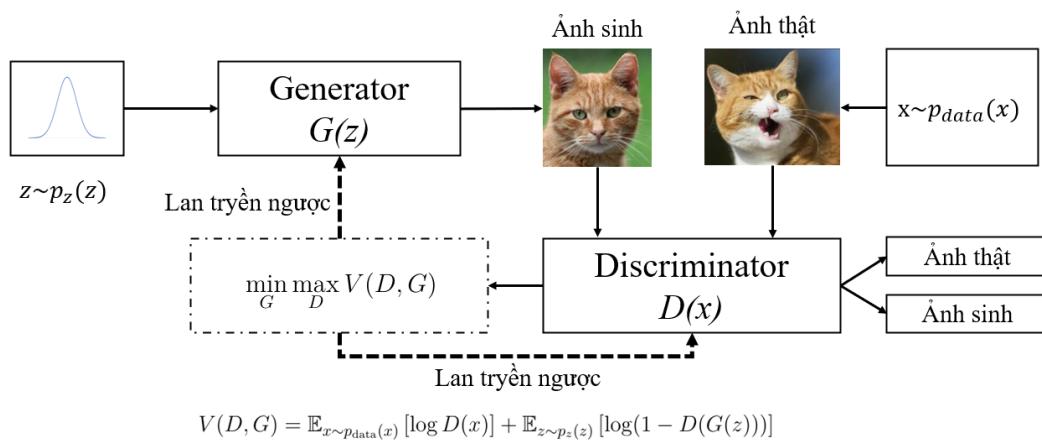
Mô hình tạo sinh ảnh là mô hình trí tuệ nhân tạo có khả năng sáng tạo ra hình ảnh mới có độ chân thật cao.

Hiện nay, các mô hình tạo sinh ảnh chủ yếu dựa trên 3 phương pháp: Generative Adversarial Networks (GANs) [3], Variational Autoencoders [19] và Diffusion Models [4]. Trong đó các mô hình GANs[3] và Diffusion [4] cho chất lượng hình ảnh tốt và được sự dụng rộng rãi hiện nay. Mô hình VAEs có ưu điểm cho kết quả đầu ra đa dạng (vì mô hình học cách biểu diễn đối tượng vào 1 phân phối xác suất), do đó nó thường được kết hợp với GANs nhằm tăng khả năng linh hoạt và độ ổn định trong việc sinh dữ liệu mới.

2.1.1 Mô hình GAN

Là một kiến trúc mạng học sâu có khả năng sinh ra dữ liệu mới chủ yếu được dùng để sinh hình ảnh và âm thanh, đặc trưng của GAN [3] là cách huấn luyện đối kháng giữa hai nhánh mạng với mục đích cuối cùng là sinh ra dữ liệu giống với mẫu huấn luyện.

Kiến trúc của GAN [3]: gồm có hai phần generator (mạng sinh dữ liệu giả) và discriminator (mạng phân biệt thật – giả) (Hình 2.1).



Hình 2.1: Minh họa mô hình GAN.

- **Discriminator:** Nhánh mạng có nhiệm vụ phân biệt một hình ảnh là thật hay được tạo ra bởi bộ generator. Đầu vào là một hình ảnh, đầu ra là dự đoán ảnh là thật hoặc ảnh là giả.
- **Generator:** Nhánh mạng này có chức năng sinh ra hình ảnh mới. Đầu vào là một nhiễu ngẫu nhiên, đầu ra là một hình ảnh được mô hình sáng tạo.

Trong quá trình huấn luyện mạng, bộ generator được khuyến khích tạo ra hình ảnh giống với hình ảnh thật và làm cho bộ discriminator phân loại sai hình ảnh. Đồng thời bộ discriminator cũng được cập nhật trọng số nhằm tăng độ chính xác trong nhiệm vụ phân loại của chính mình.

Quá trình cạnh tranh giữa hai phần này làm cho chất lượng hình ảnh được sinh bởi generator ngày càng cao, đồng thời khả năng phân loại hình ảnh của bộ discriminator được cải thiện liên tục.

Hàm mục tiêu: Mạng GAN [3] bao gồm hai nhánh mạng với mục tiêu huấn luyện đối ngược nhau và quá trình huấn luyện mạng diễn ra xen kẽ giữa Generator và Discriminator.

$$\min_G \max_D V(D, G)$$

Trong đó, hàm mục tiêu $V(D, G)$ được định nghĩa là:

$$V(D, G) = \mathbb{E}_{x \sim p_{\text{data}}(x)} [\log D(x)] + \mathbb{E}_{z \sim p_z(z)} [\log(1 - D(G(z)))]$$

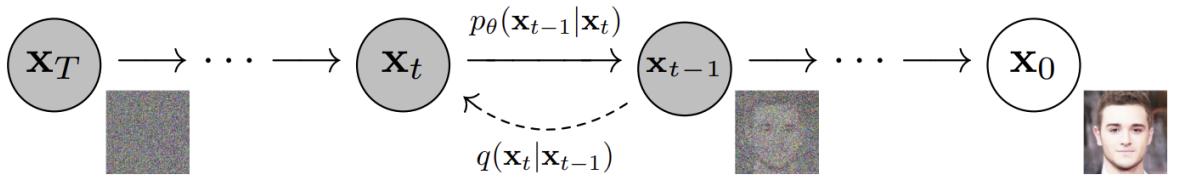
Với:

- x là dữ liệu thật từ phân phối dữ liệu thật $p_{\text{data}}(x)$.
- z là vectơ ngẫu nhiên từ phân phối nhiễu $p_z(z)$, thường là phân phối Gaussian.
- $\mathbb{E}_{z \sim p_z(z)}$ là kỳ vọng trên nhiễu z lấy từ phân phối $p_z(z)$.
- $G(z)$ là đầu ra của Generator ứng với đầu vào ngẫu nhiên z .
- $D(G(z))$ là xác suất mà Discriminator dự đoán $G(z)$ là mẫu thật.

Giải thích ý nghĩa hàm mục tiêu: generator sẽ cố gắng tối thiểu hóa $V(D, G)$ bằng cách tạo ra hình ảnh giống thật để đánh lừa discriminator tức mong muốn $D(G(z))$ càng lớn gần 1 càng tốt. Ngược lại discriminator sẽ hướng đến tối đa hóa giá trị $V(D, G)$ bằng cách phân loại chính xác ảnh thật và ảnh do generator tạo ra.

2.1.2 Mô hình Diffusion

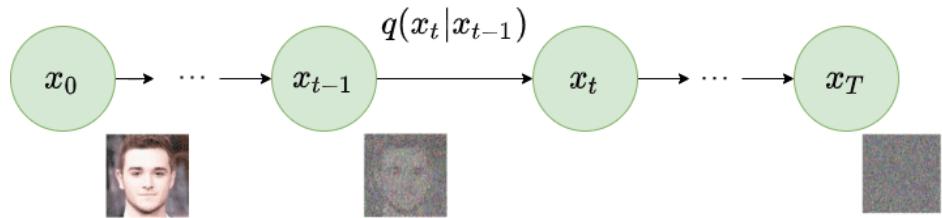
Là mô hình sinh ảnh chất lượng cao, được công bố trong nghiên cứu của Jonathan Ho năm 2020 với tên đầy đủ là "*Mô hình xác suất khuếch tán nhiễu (DDPMs)* [4]", kiến trúc mô hình trong thực tế là sự kết hợp của kỹ thuật học sâu và nguyên lý của quá trình khuếch tán trong nhiệt động lực học [20]. Nguyên lý của mô hình bao gồm hai quá trình chính (Hình 2.2): quá trình khuếch tán thuận và quá trình đảo nghịch. Trong đó tạo sinh ảnh là quá trình nghịch, mô hình học cách khôi phục hình ảnh từ nhiễu.



Hình 2.2: Đồ thị có hướng mô tả hai quá trình của mô hình Diffusion. Nguồn: [4]

Quá trình khuếch tán thuận:

Trong mô hình DDPM [4], quá trình này mô phỏng lại hiện tượng khuếch tán trong tự nhiên bằng cách lặp lại nhiều lần việc thêm nhiễu Gaussian vào hình ảnh gốc cho đến khi ảnh trở thành nhiễu hoàn toàn (Hình 2.3). Mỗi bước của quá trình được biểu diễn bằng một phân phối xác suất (Phương trình 2.1).



Hình 2.3: Quá trình khuếch tán thuận trong mô hình DDPMs. Nguồn: [4]

$$q(\mathbf{x}_t | \mathbf{x}_{t-1}) = \mathcal{N}(\mathbf{x}_t; \sqrt{1 - \beta_t} \mathbf{x}_{t-1}, \beta_t \mathbf{I}) \quad (2.1)$$

Trong đó \mathcal{N} thể hiện phân phối Gaussian, $\beta_t \in (0, 1)$ là hệ số điều chỉnh trung bình và phương sai của nhiễu được thêm vào ở bước t , (trong mô hình DDPM [4] thì hệ số β là một hàm tuyến tính theo t), và \mathbf{I} là ma trận đơn vị, x_t là ảnh ở bước t , ($t = 0$ ứng với ảnh gốc).

Toàn bộ quá trình khuếch tán thuận từ \mathbf{x}_0 đến \mathbf{x}_T được mô tả bằng phương trình 2.2, và phương trình 2.3 là cách tính nhanh $q(\mathbf{x}_t | \mathbf{x}_0)$ trực tiếp mà không cần phải tính \mathbf{x}_{t-1} .

$$q(\mathbf{x}_{1:T} | \mathbf{x}_0) = \prod_{t=1}^T q(\mathbf{x}_t | \mathbf{x}_{t-1})q(\mathbf{x}_0) \quad (2.2)$$

$$q(\mathbf{x}_t | \mathbf{x}_0) = \mathcal{N}(\mathbf{x}_t; \sqrt{\bar{\alpha}_t}\mathbf{x}_0, (1 - \bar{\alpha}_t)\mathbf{I}) \quad (2.3)$$

Trong đó:

- Đặt $\alpha_t := 1 - \beta_t$, $\bar{\alpha}_t := \prod_{s=1}^t \alpha_s$

Mục đích chính của quá trình này là tạo dữ liệu cho quá trình huấn luyện mô hình DDPM [4]. Với đầu vào là x_t và đầu ra mong muốn là x_0 thuộc phân phối mong muốn (tức loại hình ảnh cần tạo sinh)

Quá trình đảo nghịch:

Quá trình khuếch tán thuận đã tạo ra một phân phối Gaussian đơn giản từ hình ảnh có phân phối rất phức tạp. Quá trình đảo nghịch (Phương trình 2.4) sẽ thực hiện khử nhiễu để thu được phân phối dữ liệu mong muốn từ một phân phối Gaussian.

$$p_\theta(\mathbf{x}_{t-1} | \mathbf{x}_t) = \mathcal{N}(\mathbf{x}_{t-1}; \frac{1}{\sqrt{\alpha_t}}(\mathbf{x}_t - \frac{1 - \alpha_t}{\sqrt{1 - \bar{\alpha}_t}}\epsilon_t, \frac{1 - \bar{\alpha}_{t-1}}{1 - \bar{\alpha}_t} \cdot \beta_t)) \quad (2.4)$$

Trong thực tế, người ta dùng một mạng nơ-ron để dự đoán nhiễu đã được

thêm vào của từng bước thời gian chứ không trực tiếp hình ảnh ở bước thời gian trước đó.

Thuật toán huấn luyện mô hình:

Quá trình được mô tả chi tiết trong hình 2.4, và được diễn giải như sau:

- Tập ảnh huấn luyện \mathbf{x} được chia thành các mẻ (*batch*) \mathcal{B} .
- Ứng với mỗi lần duyệt qua hình ảnh \mathbf{x}_i trong \mathcal{B} , chọn tùy ý một bước thời gian $t \sim Uniform[1, \dots T]$, và một nhiễu $\epsilon \in \mathcal{N}(0, I)$.
- Thêm nhiễu ở bước thời gian t vào hình ảnh \mathbf{x}_i ban đầu để thu được ảnh nhiễu của bước t . Lượng nhiễu được thêm sẽ phụ thuộc vào t và ϵ .
- Cho hình ảnh \mathbf{x}_i đã thêm nhiễu vào mô hình $(\mathbf{g}_t(\cdot))$ để dự đoán nhiễu được thêm vào ở bước trước.
- Tính giá trị hàm mục tiêu ℓ_i là bình phương sai số giữa nhiễu dự đoán và nhiễu thực sự ϵ
- Cập nhật trọng số cho mô hình và lặp lại quá trình.

Algorithm 18.1: Diffusion model training

```

Input: Training data  $\mathbf{x}$ 
Output: Model parameters  $\phi_t$ 
repeat
  for  $i \in \mathcal{B}$  do                                // For every training example index in batch
     $t \sim Uniform[1, \dots T]$                   // Sample random timestep
     $\epsilon \sim Norm[\mathbf{0}, \mathbf{I}]$               // Sample noise
     $\ell_i = \left\| \mathbf{g}_t \left[ \sqrt{\alpha_t} \mathbf{x}_i + \sqrt{1 - \alpha_t} \epsilon, \phi_t \right] - \epsilon \right\|^2$  // Compute individual loss
    Accumulate losses for batch and take gradient step
  until converged

```

Hình 2.4: Mô tả thuật toán huấn luyện mô hình Diffusion. Nguồn: [21]

Thuật toán sinh ảnh của mô hình Diffusion:

Sau khi đã huấn luyện xong mô hình Diffusion [4], quá trình sinh ra ảnh mới từ

một điểm dữ liệu trong phân phối chuẩn diễn ra như sau.

- Tạo một nhiễu ngẫu nhiên $\mathbf{z}_T \in \mathcal{N}(0, I)$, với T là bước thời gian được chọn tùy ý (T càng lớn thì chất lượng ảnh sẽ cao nhưng đồng thời cũng tăng thời gian tạo ảnh).
- Ứng với mỗi bước thời gian $t \in [T, T-1, \dots, 2]$ ta thực hiện quá trình khử nhiễu dần dần đến khi thu được ảnh không nhiễu.
 - Dự đoán nhiễu ở bước $t-1$, khử nhiễu ta thu được ảnh $\hat{\mathbf{z}}_{t-1}$.
 - Thêm vào $\hat{\mathbf{z}}_{t-1}$ một lượng nhiễu $\epsilon \in \mathcal{N}_\epsilon(0, I)$ (nhằm mô phỏng lại quá trình thuận)
 - Lặp lại quá trình cho đến khi thu được \mathbf{z}_0 (ở đây $\mathbf{x} = \mathbf{z}_0$ là ảnh tạo sinh cuối cùng)

Algorithm 18.2: Sampling

Input: Model, $\mathbf{g}_t[\bullet, \phi_t]$
Output: Sample, \mathbf{x}
 $\mathbf{z}_T \sim \text{Norm}_{\mathbf{z}}[\mathbf{0}, \mathbf{I}]$ // Sample last latent variable
for $t = T \dots 2$ **do**

$$\left| \begin{array}{l} \hat{\mathbf{z}}_{t-1} = \frac{1}{\sqrt{1-\beta_t}} \mathbf{z}_t - \frac{\beta_t}{\sqrt{1-\alpha_t} \sqrt{1-\beta_t}} \mathbf{g}_t[\mathbf{z}_t, \phi_t] \\ \epsilon \sim \text{Norm}_{\epsilon}[\mathbf{0}, \mathbf{I}] \\ \mathbf{z}_{t-1} = \hat{\mathbf{z}}_{t-1} + \sigma_t \epsilon \end{array} \right. \quad // \text{Predict previous latent variable}$$

$$// \text{Draw new noise vector}$$

$$// \text{Add noise to previous latent variable}$$

$$\mathbf{x} = \frac{1}{\sqrt{1-\beta_1}} \mathbf{z}_1 - \frac{\beta_1}{\sqrt{1-\alpha_1} \sqrt{1-\beta_1}} \mathbf{g}_1[\mathbf{z}_1, \phi_1] \quad // \text{Generate sample from } \mathbf{z}_1 \text{ without noise}$$

Hình 2.5: Mô tả quá trình sinh ảnh mô hình Diffusion. Nguồn: [21]

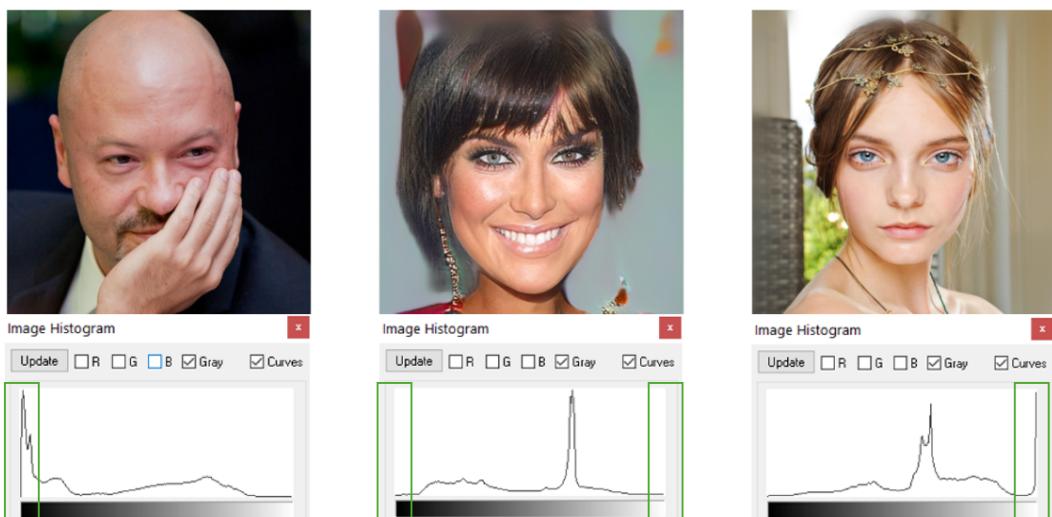
2.2 Phát hiện hình ảnh tạo sinh

Luận văn tập trung ta nghiên cứu và tiếp cận nhóm phương pháp phát hiện hình ảnh tạo sinh dựa trên phân tích, rút trích đặc trưng trên *miền không gian* và *miền tần số* của hình ảnh.

2.2.1 Nhóm phương pháp tiếp cận trên miền không gian

Nhóm phương pháp này dựa vào phân tích, rút trích các đặc trưng từ tín hiệu màu sắc, biên cạnh, hoặc nhiễu trực tiếp từ điểm ảnh.

Phương pháp của Scott McCloskey (2018) [22] đã phân tích cấu trúc của mạng GANs [3] và tập trung đến cách mà mô hình tạo ra màu sắc. Trong quá trình sinh ảnh, các giá trị màu đã được mô hình chuẩn hóa nhằm hạn chế số lượng các điểm ảnh bão hòa¹. Hơn nữa, những lớp mạng cuối cùng có nhiệm

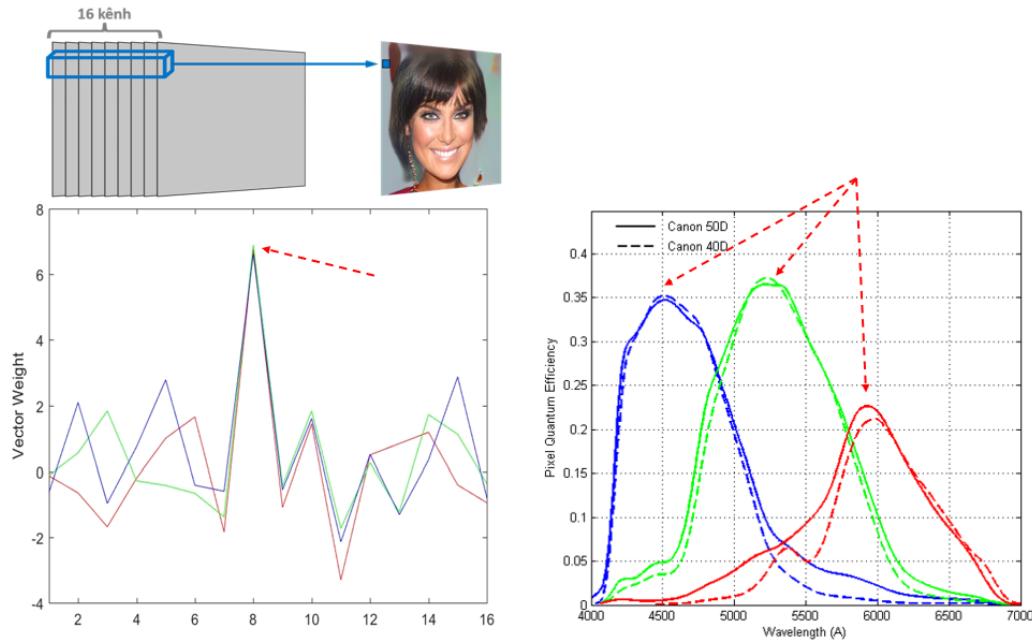


Hình 2.6: Một ví dụ về sự thiếu vắng vùng bảo hoà (vùng đánh dấu màu xanh lá nằm ở 2 bên biểu đồ histogram của ảnh tạo sinh (*giữa*), trong khi ảnh thật (*trái*) và (*phải*) có sự xuất hiện của vùng bảo hoà. *Nguồn:* [22]

vụ tổng hợp ba kênh màu cơ bản đỏ, xanh lá cây và xanh lam từ một ma trận nhiều chiều bằng cách sử dụng các lớp tích chập (cùng một bộ trọng số cho tất cả điểm ảnh), tuy nhiên tỉ lệ trọng số giữa các kênh có cấu trúc khác xa so với bộ lọc 3 màu cơ bản trong máy ảnh (Hình 2.7), qua quan sát tác giả nhận thấy rằng các thành phần màu sắc có xu hướng tương quan mạnh với nhau trong mạng GANs [3], ngược lại đối với hình ảnh thật, phổ của bộ lọc màu có sự chồng

¹là giá trị cường độ của một hoặc nhiều kênh màu đạt được cực đại (255) và cực tiểu (0) đối với định dạng 8-bit cho mỗi kênh màu

chéo và đỉnh của chúng không trùng nhau (các vị trí đỉnh nơi mũi tên màu đỏ hướng đến trong hình 2.7) . Hai quan sát trên là cơ sở mà Scott McCloskey dùng để phân biệt hình ảnh tạo sinh. Triển khai phương pháp của mình trong thực tế Scott McCloskey và cộng sự đã tinh chỉnh mạng FMIHNet [23] được đề xuất bởi Cheng (2018)²



Hình 2.7: Cấu trúc trọng số của lớp tích chập cuối trong mạng GANs (*trái*) và Phổ bộ lọc màu của 2 máy ảnh Canon 50D và Canon 40D (*phải*). Nguồn: [22]

-Ưu điểm của phương pháp: Đơn giản, hiệu quả, dữ liệu phân tích rõ ràng và quan sát được một cách trực quan.

-Hạn chế của phương pháp: Các dấu vết về sự thiếu vắng các điểm ảnh bảo hoà trong ảnh tạo sinh dễ dàng bị loại bỏ có chủ đích để qua mặt phương pháp này.

Phương pháp Gram-Net [24] do Liu và cộng sự đề xuất năm 2020. Phương pháp này phân biệt ảnh thật và ảnh tạo sinh dựa vào các kết cấu bề mặt (texture)

²FMIHNet là kiến trúc mạng cho phép phân biệt được các dấu vết làm mờ giả tạo trên ảnh được chỉnh sửa và lấy nét quan học có trên ảnh thật có đầu vào là biểu đồ tần suất (histogram) của hình ảnh.

của ảnh.

Tác giả đã thực hiện thử nghiệm để kiểm tra mức độ ảnh hưởng của kết cấu bề mặt ảnh tác động lên hiệu suất của mô hình phân loại. Thực hiện cắt khu vực chứa vùng da mặt trên ảnh và áp dụng lần lược kỹ thuật tiền xử lý (Hình 2.8), trước khi đưa hình ảnh vào huấn luyện bộ phân loại sử dụng kiến trúc CNN [25].



Hình 2.8: Ảnh gốc (a-b), ảnh biến đổi Gray-scale (c-d), ảnh áp dụng bộ lọc $L0$ (e-f).
Nguồn: [24]

- *Chuyển các màu về ảnh xám (Gray-scale)*: Nhầm loại bỏ ảnh hưởng của màu sắc.
- *Áp dụng bộ lọc $L0$ [26]*: Bộ lọc làm mịn các kết cấu nhỏ, tuy nhiên vẫn giữ lại màu sắc và hình dạng của đối tượng.

Kết quả thử nghiệm chỉ ra rằng (Bảng 2.1 dòng 4-5): Mô hình phân loại được huấn luyện với các hình ảnh gray-scale nhưng vẫn chứa đầy đủ texture cho độ chính xác giảm nhẹ so với mô hình huấn luyện trên ảnh gốc. Điều này chứng tỏ màu sắc không ảnh hưởng nhiều đến mô hình phân loại.

Ngược lại, hiệu suất của mô hình giảm mạnh (khoảng 20%) khi áp dụng bộ lọc $L0$. Cho thấy texture đóng vai trò quan trọng trong việc phân biệt ảnh tạo sinh, và mô hình CNN cũng đã trích xuất được các đặc trưng này sau quá trình huấn luyện.

Input	Human vs. CNNs	StyleGAN vs. CelebA-HQ	StyleGAN vs. FFHQ	PGGAN vs. CelebA-HQ
Full image	Human Beings	75.15%	63.90%	79.13%
Full image	ResNet	99.99%	99.96%	99.99%
Original (skin)	ResNet	99.93%	99.61%	99.96%
Gray-scale (skin)	ResNet	99.76%	99.47%	99.94%
$L0$ -filtered (skin)	ResNet	78.64%	76.84%	72.02%

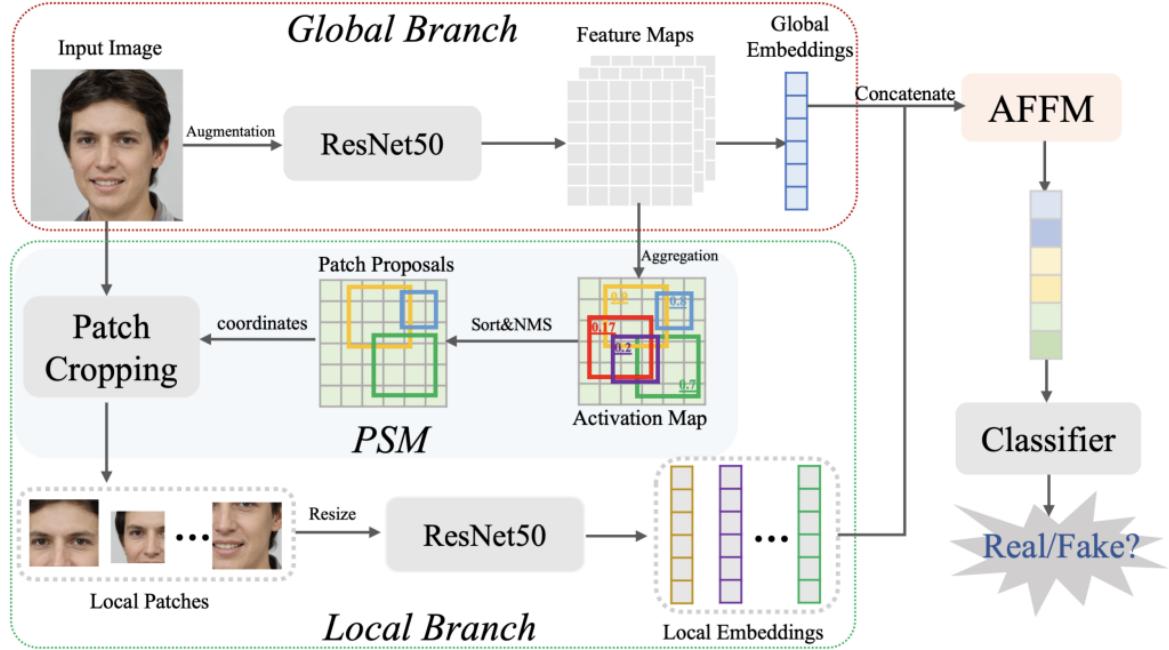
Bảng 2.1: Bảng so sánh kết quả huấn luyện bộ phân loại ứng với từng kĩ thuật tiền xử lý (dòng 3-5), và hiệu suất của của con người (dòng 1) so với mạng học sâu. Nguồn:[24]

-Đóng góp của phương pháp: Phân tích và thực hiện thử nghiệm chứng tỏ tầm quan trọng của texture trong bài toán phát hiện ảnh tạo sinh. Để xuất kiến trúc khối Gram, cho phép tích hợp các backbone CNN [25] tạo nên kiến trúc Gram-Net [24], tăng cường độ chính xác so với mạng các mạng CNN cơ bản như ResNet [27].

-Hạn chế của phương pháp: Yêu cầu độ phức tạp của mô hình lớn, đồng thời hiệu suất giảm mạnh khi thực hiện kiểm tra chéo trên các tập dữ liệu khác nhau.

Phương pháp Fusing [28] công bố trong bài báo *Fusing Global and Local Features for Generalized AI-Synthesized Image Detection* của Yan-Ju năm 2022.

Trong nghiên cứu này Yan-Ju và cộng sự thiết kế mô hình (Hình 2.9) gồm hai nhánh, nhằm kết hợp đặc trưng không gian toàn cục từ toàn bộ hình ảnh và các đặc trưng cục bộ từ nhiều mảnh ảnh nhỏ (patch).



Hình 2.9: Kiến trúc mô hình Fusing Nguồn: [28]

-Nhánh mô hình *Global Branch*: Dùng backbone ResNet50 [27] để trích xuất đặc trưng toàn cục. Cụ thể đầu vào là hình ảnh màu $\mathbb{I} \in \mathbb{R}^{3 \times w \times h}$, bản đồ đặc trưng toàn cục $\mathbb{F} \in \mathbb{R}^{C \times W_f \times H_f}$ thu được từ lớp tích chập cuối cùng của ResNet50 [27].

-Nhánh mô hình *Local Branch*: Có nhiệm vụ trích xuất thông tin từ một số mảnh vá mang nhiều thông tin nhất.

- Cơ chế Attention [29] được áp dụng để tính toán lượng thông tin của mảnh vá, và được tích hợp bên trong mô-đun *Patch Selection Module (PSM)*, với đầu vào là bản đồ đặc trưng toàn cục \mathbb{F} và đầu ra là toạ độ của các patch trên ảnh \mathbb{I} đủ tiêu chuẩn.

- Vec-tơ đặc trưng cục bộ được tính tương tự với đặc trưng toàn cục (tức sử dụng backbone ResNet50 [27]), tuy nhiên đầu vào là những patch (được chọn ra bởi mô-đun PSM), thay vì là toàn bộ hình ảnh.
- Đặc trưng toàn cục và đặc trưng cục bộ được kết hợp với nhau thông qua cơ chế attention [29] có trong mô-đun *Attention-based Feature Fusion*, trước khi đưa qua bộ phân loại.

-Ưu điểm của phương pháp: Cho phép kết hợp đặc trưng toàn cục và đặc trưng cục bộ của hình ảnh, giúp tăng tính khái quát của mô hình.

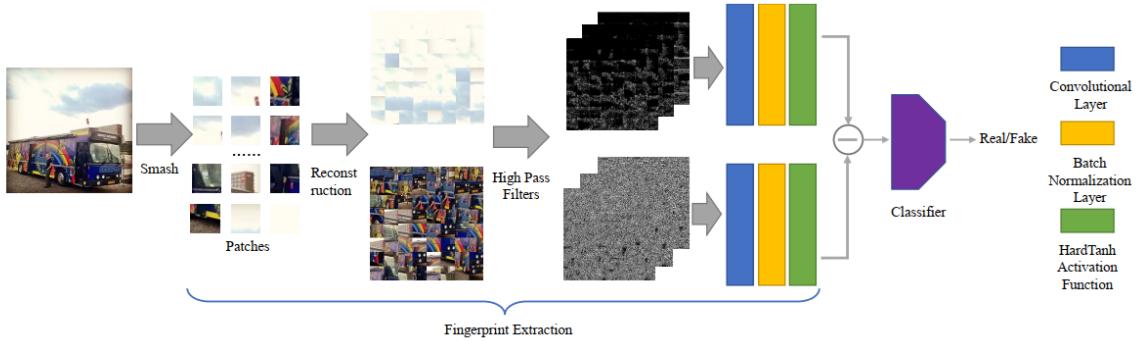
-Hạn chế của phương pháp: Kiến trúc có 2 nhánh mạng và đều sử dụng backbone ResNet50 [27], nghĩa là độ phức tạp của mô hình tăng gấp 2 lần.

Một số phương pháp cùng hướng tiếp cận:

Công trình nghiên cứu *Multi-Attentional Deepfake Detection* của Zhao và cộng sự (2021) [30] đã trích xuất và tổng hợp các texture ở nhiều mức độ phóng to khác nhau, kết hợp với mô-đun Attention [29] nhằm tích hợp chúng với các đặc trưng ngữ nghĩa trừu tượng hơn từ các lớp sâu của mạng.

Nan-Zhong công bố nghiên cứu *Rich and Poor Texture Contrast: A Simple yet Effective Approach for AI-generated Image Detection* [31](2023). Phương pháp này khai thác sự tương phản trong mối tương quan giữa các vùng kết cấu phức tạp (rich texture regions) và các vùng kết cấu đơn giản (poor texture regions), bằng phẳng trong một bức ảnh.

Trước tiên, cắt hình ảnh gốc thành nhiều patch và tái cấu trúc chúng thành ảnh chứa rich texture regions và ảnh chứa poor texture regions. Sau đó sử dụng mạng CNN [25] rút trích, kết hợp các loại đặc trưng khác nhau. Các đặc trưng sau khi tổng hợp sẽ được đưa qua một mạng nơ-ron đa tầng đơn giản làm nhiệm vụ phân loại ảnh thật/giả mạo (Hình 2.10).



Hình 2.10: Mô tả phương pháp Rich Poor Texture Contrast của Nang-Zhong Nguồn: [31]

-Ưu điểm của phương pháp: Việc tái cấu trúc lại hình ảnh thành rich texture regions và poor texture regions giúp mô hình trích xuất đặc trưng có hiệu quả hơn, làm tăng độ chính xác của mô hình. Hơn nữa việc tái cấu trúc này sẽ phá vỡ ngữ nghĩa của hình ảnh, làm giảm hiện tượng thiên lệch mẫu huấn luyện, giúp mô hình nâng cao mức độ khái quát.

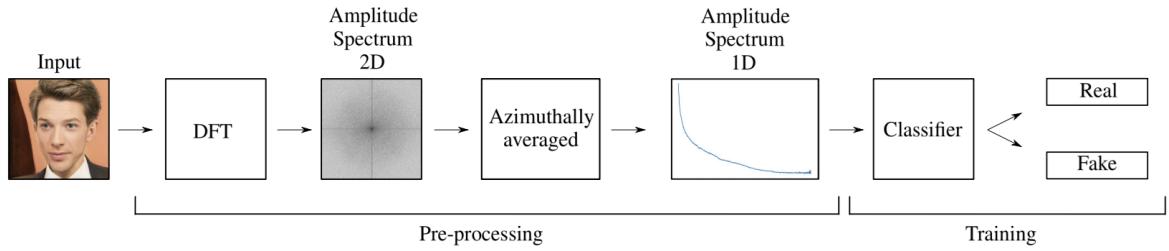
-Hạn chế của phương pháp: Khó khăn khi lựa chọn kích thước patch phù hợp, vì mỗi hình ảnh sẽ có cấu trúc khác nhau. Thực nghiệm cũng cho thấy việc sử dụng kích thước patch khác với quá trình huấn luyện, sẽ làm giảm độ chính xác của phương pháp.

2.2.2 Nhóm phương pháp tiếp cận miền tần số

Nhóm phương pháp này thực hiện chuyển đổi hình ảnh từ miền không gian sang miền tần số bằng các phép biến đổi như FFT [32] hoặc biến đổi Fourier rời rạc (DFT) [33]. Khi tập trung vào các đặc điểm tần số, các phương pháp này cho kết quả phát hiện các dấu vết giả tạo tốt hơn so với miền không gian.

Unmasking DeepFakes with simple Features [34] (2019) của Durall. Bước đầu hình ảnh được chuyển thành phổ công suất bằng DFT, sau đó áp dụng phương pháp trung bình theo phương góc (azimuthal averaging) để chuyển phổ công suất từ hai chiều về một chiều nhằm mục đích phù hợp cho quá trình huấn

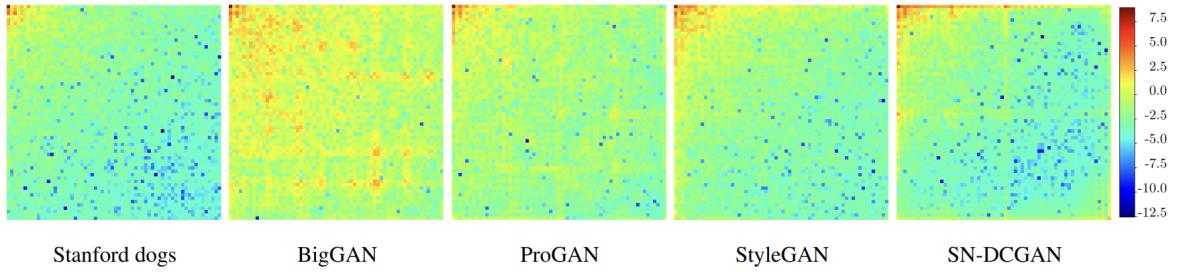
luyện bộ phân loại (Hình 2.11).



Hình 2.11: Tổng quan về quy trình xử lý trong phương pháp của Durall. Nguồn: [34]

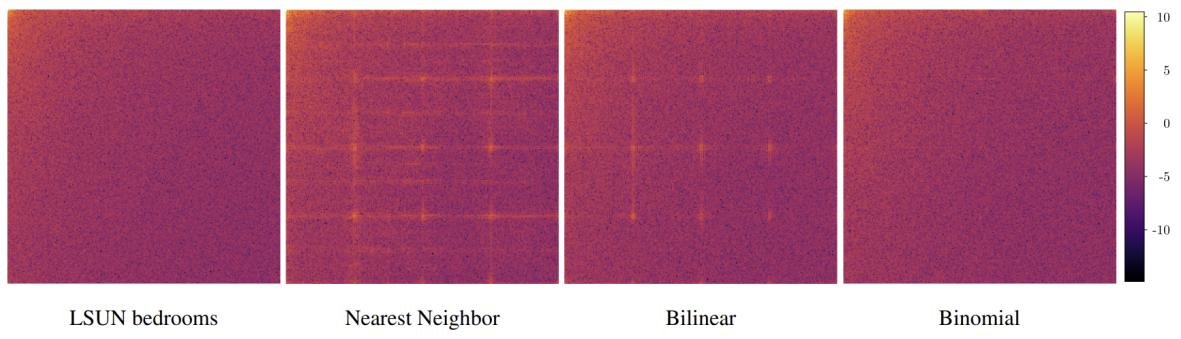
Nhận xét phương pháp: Đây là phương pháp đơn giản, yêu cầu dữ liệu huấn luyện ít, và sử dụng các bộ phân loại cơ bản như: Support Vector Machine (SVM), K-means, hồi quy logistic (Logistic Regression) nhưng cho kết quả tương đối cao trên một số bộ dữ liệu cụ thể được dùng trong bài báo. Các kết quả thử nghiệm cho thấy tiềm năng hứa hẹn của việc sử dụng các phương pháp tiếp cận trên miền tần số trong việc phát hiện ảnh tạo sinh.

Các phân tích của Frank[35] cung cấp trong bài báo "*Leveraging Frequency Analysis for Deep Fake Image Recognition (2020)*" chỉ ra bằng chứng biểu diễn tần số của hình ảnh sẽ làm lộ ra các dấu vết tạo tác mà mô hình để lại, quan sát biểu đồ phổ (Hình 2.12) được vẽ từ ảnh trung bình của 10,000 hình ảnh được thực hiện biến đổi Cosine rời rạc (Discrete Cosine Transform – DCT) trong tập dữ liệu *Stanford dog* có thể nhận thấy được bằng mắt thường một số khác biệt giữa phổ của ảnh thật (vị trí ngoài cùng bên trái) so với các ảnh còn lại. Cụ thể, với ảnh thật, năng lượng tập trung ở khu vực tần số thấp (góc trên bên trái) và dần đều về phía tần số cao (góc dưới bên phải), trong khi đó ở ảnh tạo sinh, sẽ xuất hiện đột ngột các dãy tần số có năng lượng cao, theo quy luật (các đường tương tự như lưới vuông trong hình). Tiếp tục mở rộng các thử nghiệm của mình Frank khảo sát ảnh hưởng của toán tử up-sampling tác động lên hình ảnh tạo sinh, bằng cách vẽ lại biểu đồ phổ trung bình (xem Hình 2.13) sau khi thay



Hình 2.12: Phổ hình ảnh được tạo ra bởi các mạng nơ-ron khác nhau được đào tạo trên tập dữ liệu *Stanford dog*. Nguồn: [35]

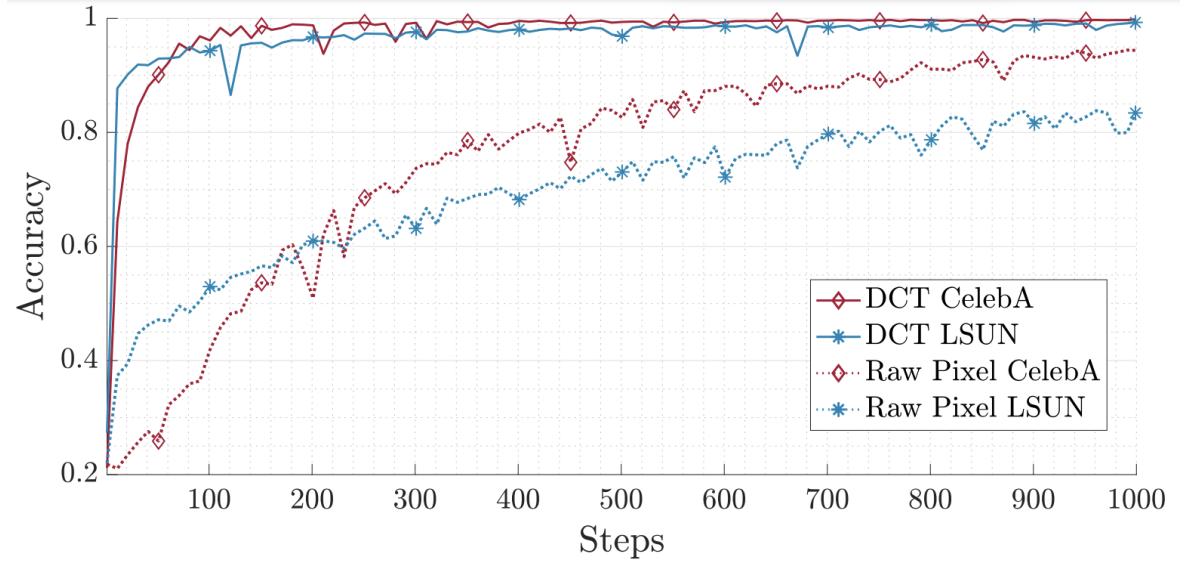
thế các toán tử up-sampling khác nhau gồm: phương pháp láng giềng gần nhất (nearest neighbor), bilinear, nội suy nhị thức (binomial up-sampling). Tác giả huấn luyện 3 phiên bản StyleGAN [36] trên tập dữ liệu LSUN-Bedroom [37], phiên bản đầu tiên giữ nguyên toán tử bilinear, phiên bản thứ hai và ba lần lượt thay thế bằng nearest neighbor và binomial up-sampling. Sự khác biệt của các dấu vết tạo tác xuất hiện rõ ràng dưới dạng lưới ô vuông trong hình 2.13 ứng với nearest neighbor, bilinear và rất mờ với binomial up-sampling khi so sánh với ảnh thật (ngoài cùng bên trái).



Hình 2.13: Phổ hình ảnh tương ứng với các kỹ thuật up-sampling khác nhau. Nguồn: [35]

Ngoài ra, bằng thực nghiệm, tác giả chứng minh rằng bộ phân loại dựa trên biểu diễn tần số mang lại độ chính xác cao, đồng thời yêu cầu ít tham số hơn so với giữ nguyên giá trị điểm ảnh. Hình 2.14 cung cấp thông tin quá trình huấn luyện bộ phân loại sử dụng kiến trúc CNN, kết quả cho thấy tốc độ hội

tụ nhanh vượt trội của mô hình khi áp dụng DCT [33] cho hình ảnh so với giữ nguyên giá trị điểm ảnh gốc.

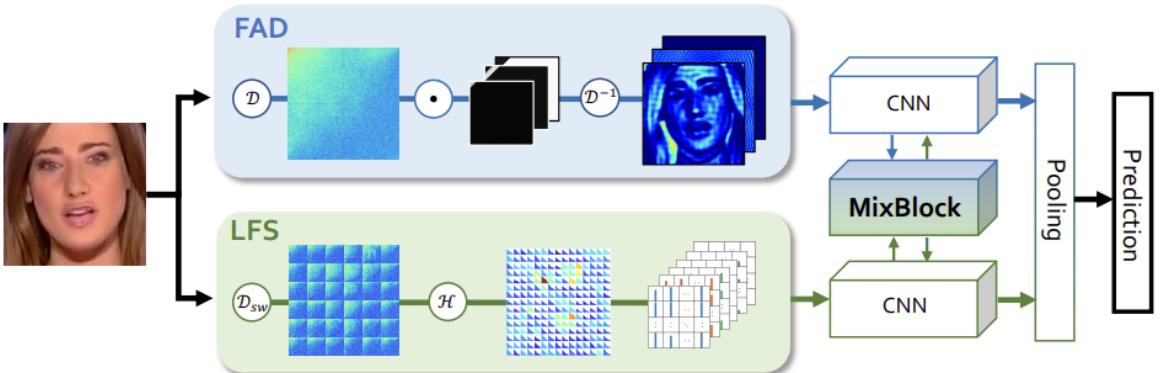


Hình 2.14: Đường cong độ chính xác theo bước huấn luyện. Nguồn: [35]

Phương pháp F3-Net [38] (Hình 2.15) do Qian đề xuất năm 2020. Phương pháp này tập trung phân tích, rút trích các đặc trưng giả mạo trên miền tần số tại các khu vực cục bộ trên ảnh thông qua hai khối, *Frequency-aware Decomposition (FAD)* và *Local Frequency Statistics (LFS)*. FAD có chức năng phân rã 1 ảnh đầu vào thành $N = 3$ ảnh đầu ra, mỗi ảnh mang thông tin về phổ tần số khác nhau. LFS có chức năng tạo một mặt nạ thể hiện thống kê tần số cục bộ có tương quan không gian với ảnh đầu vào.

-Ưu điểm của phương pháp: Qian thực hiện phân tích tần số trên những khu vực ảnh cục bộ và biểu diễn lại các vị trí này trên ma trận, nhờ đó giữ được mối tương quan về không gian, đảm bảo tương thích với kiến trúc CNN [25]. Cách làm này nhằm bổ sung khuyết điểm mất thông tin về không gian khi thực hiện biến đổi ảnh sang miền tần số.

-Hạn chế của phương pháp: Các biến đổi sang miền tần số trên những khu



Hình 2.15: Mô tả kiến trúc F3-Net. Nguồn: [38]

vực ảnh nhỏ sẽ bị hạng ché về dải tần số, hơn nữa tại biên của các khu vực nhỏ này sẽ xuất hiện hiệu ứng rò rỉ (spectral leakage [39]), sinh ra các tín hiệu nhiễu tần số cao, gây bất lợi cho quá trình huấn luyện.

Phương pháp BiHPF [40] của Jeong (2021). Trong nghiên cứu này tác giả đề xuất bộ lọc thông cao song phương BiHPF với đầu vào là một biến đổi FFT [32] của hình ảnh, giúp khuếch đại ảnh hưởng của các dấu hiệu tạo tác trong miền tần số.

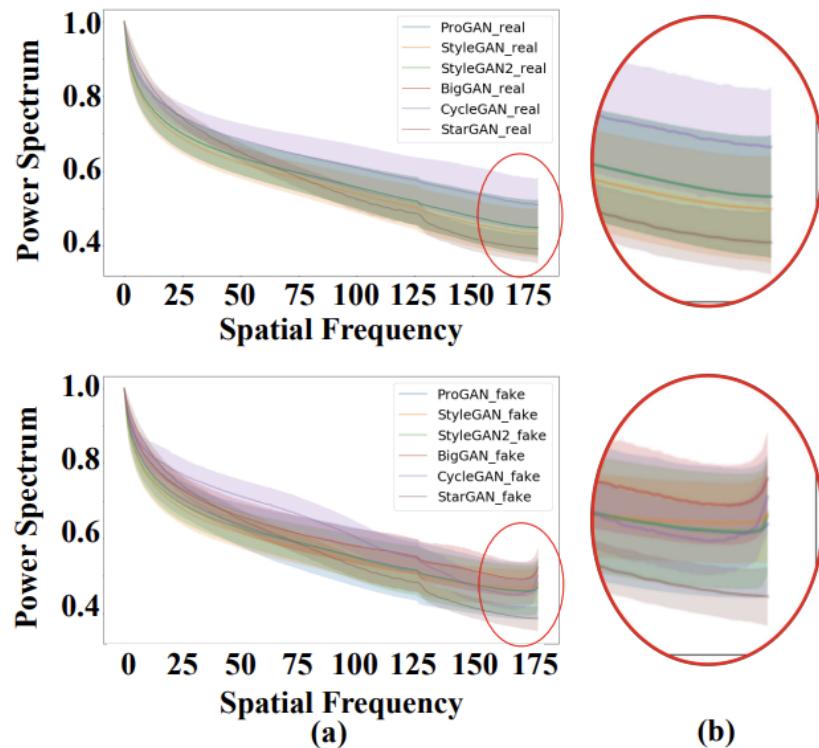
Để phân tích, tìm ra các thành phần tần số đặc trưng có trong ảnh tạo sinh, Jeong thiết kế và huấn luyện một mô-đun tên là *Artifact Compression Map*, có khả năng tách biệt các thành phần tần số đặc trưng của ảnh tạo sinh. Từ đó rút ra kết luận "*Các dấu vết tạo tác xuất hiện chủ yếu trong các thành phần tần số cao và vùng nền của hình ảnh ở mức độ điểm ảnh*". Nói cách khác, việc phân tích phổ tần số cao của các vùng ảnh có kết cấu đơn giản, bằng phẳng sẽ đem lại hiệu quả cao hơn trong nhiệm vụ phát hiện ảnh tạo sinh.

Do đó, BiHPF được thiết kế để trích xuất các dấu vết tạo tác có trong ảnh tạo sinh trên miền tần số, tại các vùng ảnh bằng phẳng. Đầu ra của bộ lọc này sau đó cho qua bộ phân loại sử dụng kiến trúc Resnet50 [27]. thực hiện nhiệm vụ phát hiện ảnh tạo sinh.

-Đóng góp của phương pháp: Dựa ra được bằng chứng "*Các dấu vết tạo tác xuất hiện chủ yếu trong các thành phần tần số cao và vùng nền của hình ảnh ở mức độ điểm ảnh*", cung cấp thông tin hữu ích cho nhiều nghiên cứu.

-Hạn chế của phương pháp: Quá trình huấn luyện phức tạp. **Phương pháp FrePGAN** [41] được Jeong giới thiệu vào năm 2022.

Thông qua phân tích phổ tần số giữa ảnh thật và tạo sinh, tác giả phát hiện sự gia tăng năng phổ lượng đáng kể ở vùng tần số cao trong ảnh tạo sinh, các dấu hiệu này dễ dàng quan sát được bằng mắt thường (Hình 2.16).



Hình 2.16: So sánh phổ công suất của dữ liệu thực và dữ liệu tạo sinh. Nguồn: [41]

Ngoài ra các phương pháp sử dụng miền tần số có xu hướng quá khớp trong quá trình huấn luyện. Để khắc phục các hạn chế này, phương pháp FrePGAN sẽ tạo ra một bản đồ nhiễu trong miền tần số và đưa vào trong hình ảnh với mục đích tạo ra các mẫu dữ liệu khó, từ đó giúp mô hình tập trung tìm kiếm nhiều

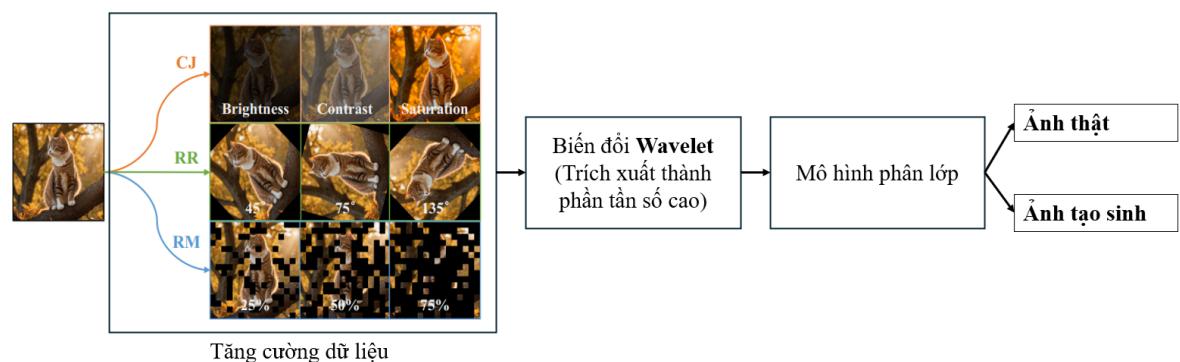
đặc trưng hữu ích hơn. -**Ưu điểm của phương pháp:** Huấn luyện một mạng GAN [3] có khả năng sinh ra các nhiễu loạn ở mức tần số, từ đó khuyến khích mô hình phân loại học thêm nhiều đặc trưng mới, làm tăng khả năng khái quát của mô hình.

-**Hạn chế của phương pháp:** Đào tạo mạng GAN [3] sinh nhiễu là phức tạp và khó kiểm soát. Hơn nữa thêm nhiễu vào dữ liệu tuy sẽ tăng tính khái quát của mô hình nhưng đồng thời cũng có xu hướng làm giảm độ chính xác.

2.2.3 Nhóm phương pháp hỗn hợp

Hướng tiếp cận này là sự kết hợp của nhiều phương pháp hoặc nhiều loại mô hình với nhau.

Li và cộng sự (2024) đã công bố phương pháp SAFE [42]: Tác giả đã thực hiện nhiều kỹ thuật tăng cường dữ liệu bao gồm: ngẫu nhiên thay đổi các thuộc tính màu sắc, xoay, tạo ra mặt nạ ngẫu nhiên che khuất một phần hình ảnh, trước khi thực hiện trích xuất đặc trưng tần số cao trên miền tần số. Các bước thực hiện được minh họa ở Hình 2.17.



Hình 2.17: Mô tả phương pháp SAFE [42].

-**Đóng góp của phương pháp:** Kết quả của nghiên cứu đã chứng tỏ ảnh hưởng tích cực của các kỹ thuật tăng cường dữ liệu đến tính khái quát của mô hình phân loại.

Ngoài ra tác giả cũng làm thực nghiệm để kiểm chứng mối tương quan cục bộ mạnh giữa các điểm ảnh kề nhau trong ảnh tạo sinh, được cho là gây ra bởi việc sử dụng toán tử up-sampling và convolution trong các mô hình tạo sinh.

CHƯƠNG 3. PHƯƠNG PHÁP ĐỀ XUẤT

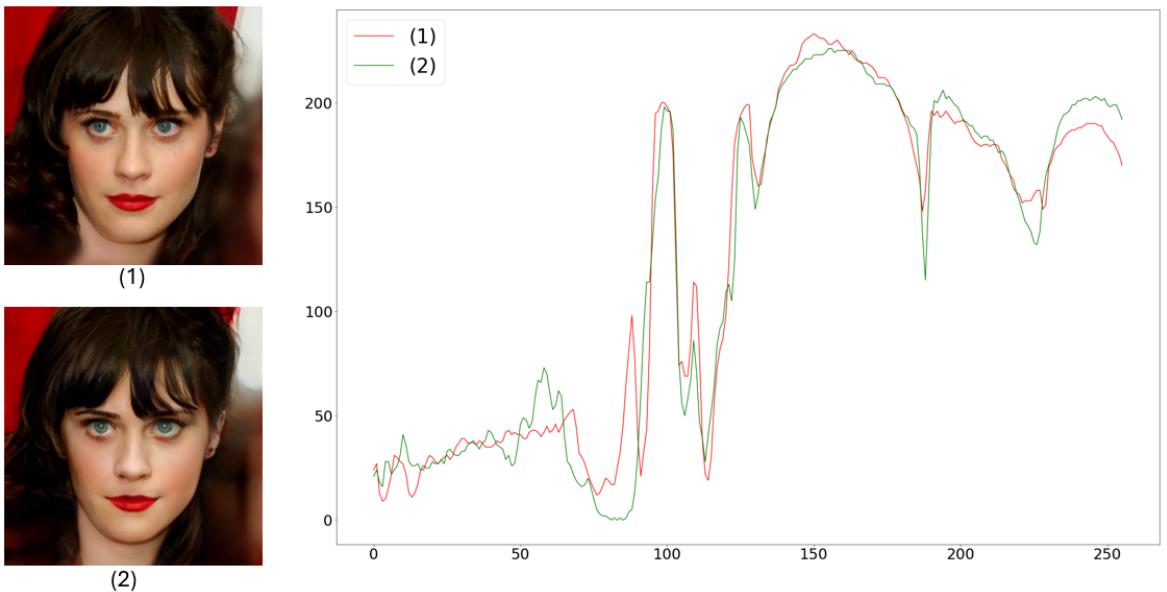
Trong chương này, luận văn trình bày chi tiết quá trình phân tích và đề xuất phương pháp phát hiện hình ảnh tạo sinh dựa trên đặc trưng tần số cao. Trước tiên, một số thử nghiệm sơ bộ được tiến hành nhằm khảo sát ảnh hưởng của các thành phần tần số đến hiệu quả phân loại giữa ảnh thật và ảnh tạo sinh. Dựa trên kết quả thu được, chương tiếp tục giới thiệu một khối tiền xử lý áp dụng bộ lọc thông cao ADOF hiệu quả với chi phí tính toán thấp

Tiếp theo, một mô hình phân loại được xây dựng trên nền kiến trúc ResNet [27], kết hợp với khối tiền xử lý tần số để tăng cường khả năng nhận biết các đặc trưng vi mô của ảnh tạo sinh. Nhằm đáp ứng yêu cầu triển khai thực tế trên các hệ thống hạn chế tài nguyên, chương này cũng đề xuất một phiên bản rút gọn của mô hình, bằng cách sử dụng kỹ thuật feature-based knowledge distillation. Phần cuối cùng mô tả chi tiết quy trình huấn luyện và triển khai mô hình trong cả hai giai đoạn: đào tạo và suy luận.

3.1 Thử nghiệm sơ bộ và hướng tiếp cận

3.1.1 Thử nghiệm 1

Trong Hình 3.1, ta sử dụng ảnh thật (1) lấy từ tập dữ liệu FFHQ và tạo sinh ảnh tương ứng (2) bằng phương pháp ReStyle [43]. Việc sử dụng cặp ảnh thật–giả có cùng đặc điểm khuôn mặt giúp ta khảo sát trực quan độ khó của nhiệm vụ



Hình 3.1: Biểu đồ (*phải*) thể hiện mức xám dòng thứ 100 của ảnh thật (1) và ảnh tạo sinh (2).

phân biệt hình ảnh tạo sinh. Kết quả cho thấy hai ảnh có mức độ tương đồng cao về hình dáng và chi tiết, gây khó khăn cho việc phân biệt bằng mắt thường. Điều này có thể lý giải bởi thực tế rằng thị giác con người chủ yếu nhận biết thông tin ở các vùng ảnh có kích thước lớn và có độ tương phản cao, trong khi các đặc biệt đặc trưng để phân biệt ảnh thật và ảnh tạo sinh rất có thể tồn tại ở mức độ điểm ảnh.

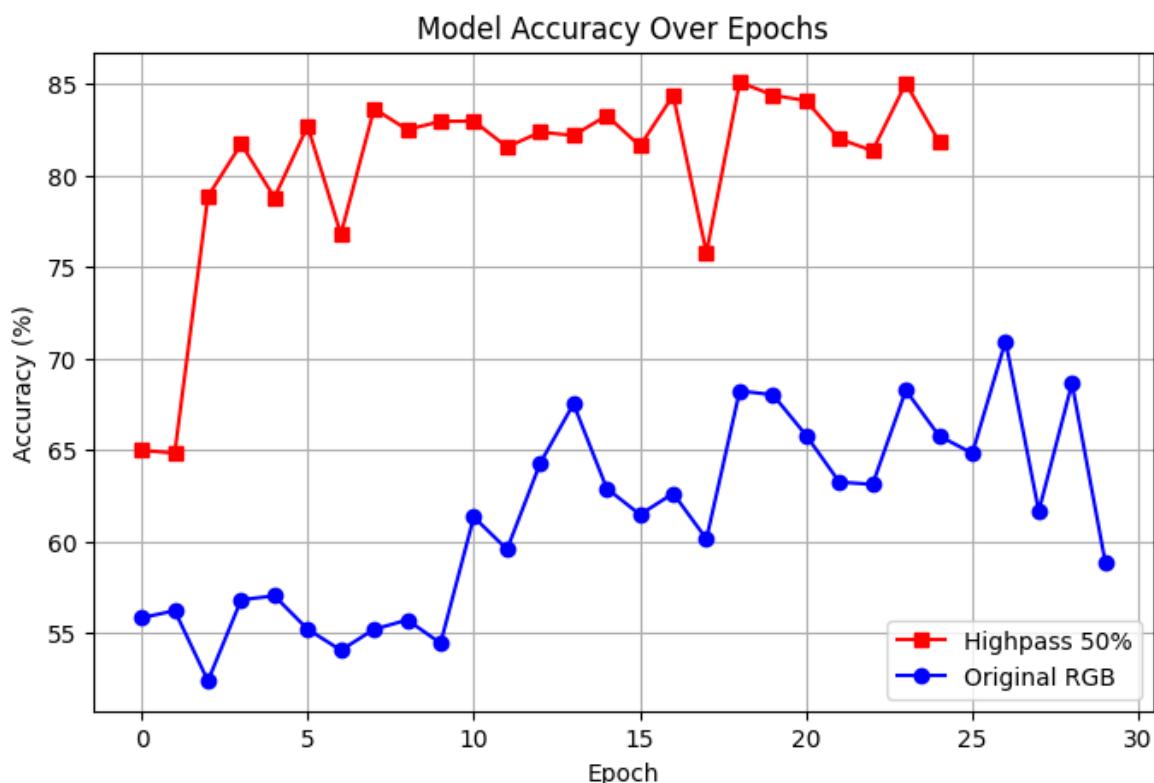
Các nghiên cứu trước đây [40], [35], [41] cũng chỉ ra rằng các đặc trưng nhân tạo có xu hướng tập trung ở vùng tần số cao và đã tận dụng điều này để phát triển bộ phân loại dựa trên đặc trưng tần số. Phần 3.1.2, ta tiến hành thử nghiệm nhằm đánh giá sự ảnh hưởng của đặc trưng tần số đến hiệu xuất bộ phân loại ảnh thật, giả.

3.1.2 Thử nghiệm 2

Bước tiếp theo của quá trình phân tích, ta thực hiện huấn luyện hai bộ phân loại dựa trên kiến trúc ResNet [27] và sử dụng cùng bộ dữ liệu [25].

Bộ phân loại số 1: (Original RGB) ảnh đầu vào là ảnh màu RGB gốc, giữ nguyên toàn bộ thông tin tần số.

Bộ phân loại số 2: (Highpass 50%) Mô hình này sử dụng cùng kiến trúc với Bộ phân loại số 1, tuy nhiên ảnh đầu vào đã được xử lý bằng bộ lọc thông cao, trong đó 50% thành phần tần số thấp đã bị loại bỏ thông qua biến đổi Fourier. Kết quả huấn luyện được trình bày trong Hình 3.2



Hình 3.2: Độ chính xác của hai bộ phân loại trên tập kiểm tra trong quá trình huấn luyện.

Thử nghiệm đã cho thấy các thành phần tần số cao đóng vai trò quan trọng trong quá trình phân loại ảnh thật – giả. Cụ thể, mô hình được huấn luyện trên ảnh đã qua bộ lọc thông cao (Highpass 50%) đạt độ chính xác cao hơn so với mô hình sử dụng ảnh RGB gốc. Điều này gợi ý rằng thông tin tần số cao chứa các đặc trưng quan trọng giúp phân biệt ảnh tạo sinh, trong khi các thành phần

tần số thấp có thể chứa nhiều thông tin nền không hữu ích cho quá trình nhận diện.

Những quan sát từ thử nghiệm cho thấy rằng việc tăng cường khai thác thông tin ở dải tần số cao có thể mang lại lợi ích đáng kể cho nhiệm vụ phát hiện ảnh tạo sinh. Do đó, **hướng tiếp cận nghiên cứu trong phần tiếp theo sẽ tập trung vào khai thác đặc trưng tần số cao để cải thiện độ chính xác phân loại**.

3.1.3 Các hạn chế khi dùng biến đổi Fourier

Trong quá trình thử nghiệm, ta đã sử dụng FFT [32] để phân tích tín hiệu trong miền tần số và thực hiện việc loại bỏ các thành phần tần số thấp nhằm tập trung khai thác đặc trưng tần số cao. Việc này giúp tăng hiệu quả phân loại ảnh thật và giả mạo. Tuy nhiên, mặc dù FFT có ưu điểm về tốc độ tính toán nhanh hơn nhiều so với DFT, quá trình áp dụng FFT và cắt lọc tần số cũng tồn tại những hạn chế nhất định.

- Khó khăn trong việc tối ưu hoá tốc độ bằng tính toán song song tận dụng sức mạnh của GPU.
- Tốn nhiều tài nguyên bộ nhớ do các biến đổi ở dạng số phức nên yêu cầu thêm bộ nhớ.
- Hiệu ứng biến trong biến đổi FFT làm xuất hiện các tín hiệu nhiễu không mong muốn.

3.2 Bộ lọc thông cao trên không gian ảnh

Để khắc phục các hạn chế của FFT trong việc lọc tần số cao, một hướng tiếp cận thay thế là xây dựng bộ lọc thông cao trực tiếp trong miền không gian ảnh,

cho phép loại bỏ thành phần tần số thấp mà không cần thực hiện biến đổi FFT và biến đổi FFT ngược. Cách làm này không chỉ giúp giảm độ phức tạp tính toán mà còn phù hợp hơn cho các hệ thống yêu cầu xử lý nhanh, có tài nguyên hạn chế hoặc dễ dàng tính toán song song nếu hệ thống hỗ trợ GPU.

Trong phần này, tôi sẽ trình bày cơ sở lý thuyết của phương pháp xây dựng bộ lọc thông cao dựa trên việc biến đổi từ một bộ lọc thông thấp đơn giản – cụ thể là bộ lọc trung bình.

3.2.1 Bộ lọc trung bình

Bộ lọc trung bình rời rạc $h_{LP}[n]$ có thể được định nghĩa theo công thức:

$$h[n] = \begin{cases} \frac{1}{M}, & 0 \leq n < M \\ 0, & \text{ngược lại} \end{cases}$$

Trong đó, M là kích thước của cửa sổ trượt. Tín hiệu đầu ra của bộ lọc trung bình được tính bằng:

$$y_{LP}[n] = \frac{1}{M} \sum_{k=0}^{M-1} x[n - k]$$

Để tìm đáp ứng tần số, ta tiến hành biến đổi Fourier rời rạc của bộ lọc:

$$H_{LP}(\omega) = \sum_{n=0}^{M-1} \frac{1}{M} e^{-j\omega n} = \frac{1}{M} \cdot \frac{1 - e^{-jM\omega}}{1 - e^{-j\omega}}$$

Biểu thức này có thể được rút gọn thành:

$$H_{LP}(\omega) = \frac{1}{M} e^{-j\omega \frac{M-1}{2}} \cdot \frac{\sin\left(\frac{M\omega}{2}\right)}{\sin\left(\frac{\omega}{2}\right)}$$

Với biên độ đáp ứng tần số là:

$$|H_{LP}(\omega)| = \left| \frac{1}{M} \cdot \frac{\sin\left(\frac{M\omega}{2}\right)}{\sin\left(\frac{\omega}{2}\right)} \right|$$

Phân tích đặc tính đáp ứng biên độ

- Khi $\omega \rightarrow 0$, theo định lý L'Hopital, ta có:

$$\lim_{\omega \rightarrow 0} |H_{LP}(\omega)| = 1$$

- Khi $\omega \rightarrow \pi$, kết quả phụ thuộc vào giá trị của M :

- Với M chẵn:

$$\sin\left(\frac{M\pi}{2}\right) = 0 \Rightarrow |H_{LP}(\pi)| = 0$$

- Với M lẻ:

$$\sin\left(\frac{M\pi}{2}\right) = \pm 1 \Rightarrow |H_{LP}(\pi)| = \frac{1}{M}$$

Nhận xét: Để triệt tiêu hoàn toàn tần số cao, ta nên chọn M chẵn.

Thiết kế bộ lọc thông cao

Bộ lọc thông cao có thể được xây dựng từ bộ lọc thông thấp thông qua một quan hệ tuyến tính. Cụ thể, tín hiệu đầu ra của bộ lọc thông cao là:

$$y_{HP}[n] = x[n] - y_{LP}[n]$$

Với $h_{HP}[n]$ là đáp ứng xung của bộ lọc thông cao, ta có:

$$h_{HP}[n] = \delta[n] - h_{LP}[n]$$

Đáp ứng tần số của bộ lọc thông cao được cho bởi:

$$H_{\text{HP}}(\omega) = 1 - H_{\text{LP}}(\omega)$$

Thay thế biểu thức của $H_{\text{LP}}(\omega)$, ta có:

$$H_{\text{HP}}(\omega) = 1 - \frac{1}{M} e^{-j\omega \frac{M-1}{2}} \cdot \frac{\sin\left(\frac{M\omega}{2}\right)}{\sin\left(\frac{\omega}{2}\right)}$$

Biên độ đáp ứng tần số của bộ lọc thông cao là:

$$|H_{\text{HP}}(\omega)| = \left| 1 - \frac{1}{M} \cdot \frac{\sin\left(\frac{M\omega}{2}\right)}{\sin\left(\frac{\omega}{2}\right)} \right|$$

Phân tích đặc tính đáp ứng biên độ

- Khi $\omega \rightarrow 0$:

$$|H_{\text{HP}}(0)| = |1 - H(0)| = 0$$

- Khi $\omega \rightarrow \pi$, kết quả cũng phụ thuộc vào giá trị của M :

Với M chẵn:

$$|H_{\text{HP}}(\pi)| = 1 - 0 = 1, \quad \text{tùy bộ tần số cao nhất được bảo toàn}$$

Với M lẻ:

$$|H_{\text{HP}}(\pi)| = 1 - \frac{1}{M} < 1, \quad \text{một phần tần số cao nhất bị suy hao}$$

Nhận xét: Để giữ lại tần số cao nhiều nhất, ta nên chọn M chẵn.

Cửa sổ bộ lọc thông cao

Từ biểu thức đáp ứng Xung của Bộ Lọc Thông Cao:

$$h_{HP}[n] = \delta[n] - h_{LP}[n]$$

Với $n = 0$:

$$h_{HP}[0] = \delta[0] - h_{LP}[0] = 1 - \frac{1}{M}$$

Với $1 \leq n < M$:

$$h_{HP}[n] = \delta[n] - \frac{1}{M} = 0 - \frac{1}{M} = -\frac{1}{M}$$

Với $n \geq M$:

$$h_{HP}[n] = \delta[n] - 0 = 0$$

Do đó, đáp ứng xung của bộ lọc thông cao có dạng:

$$h_{HP}[n] = \begin{cases} 1 - \frac{1}{M}, & n = 0 \\ -\frac{1}{M}, & 1 \leq n < M \\ 0, & \text{ngược lại} \end{cases}$$

Biểu thức tổng quát cho bộ lọc thông cao có thể được diễn đạt dưới dạng vector độ dài M :

$$h_{HP}[n] = \left[1 - \frac{1}{M}, -\frac{1}{M}, -\frac{1}{M}, \dots, -\frac{1}{M} \right] \quad (3.1)$$

Bộ lọc thông cao có thể được xây dựng từ bộ lọc trung bình thông qua một

phép toán đơn giản. Qua thực nghiệm, ta thấy rằng $M = 2$ cho kết quả tối ưu cho mô hình. Với $M = 2$, biểu thức cho đáp ứng xung của bộ lọc thông cao từ phương trình (3.1) trở thành:

$$h_{\text{HP}}[n] = \left[\frac{1}{2}, -\frac{1}{2} \right] \quad (3.2)$$

3.3 Xây dựng khối tiền xử lý (ADOF) dựa trên bộ lọc thông cao

Dựa trên phân tích tại mục 3.2.1, ta chọn độ dài cửa sổ lọc $M = 2$ vì giá trị M chắn giữ lại vùng tần số cao tốt hơn so với giá trị lẻ. Đồng thời, lựa chọn M nhỏ giúp giảm chi phí tính toán.

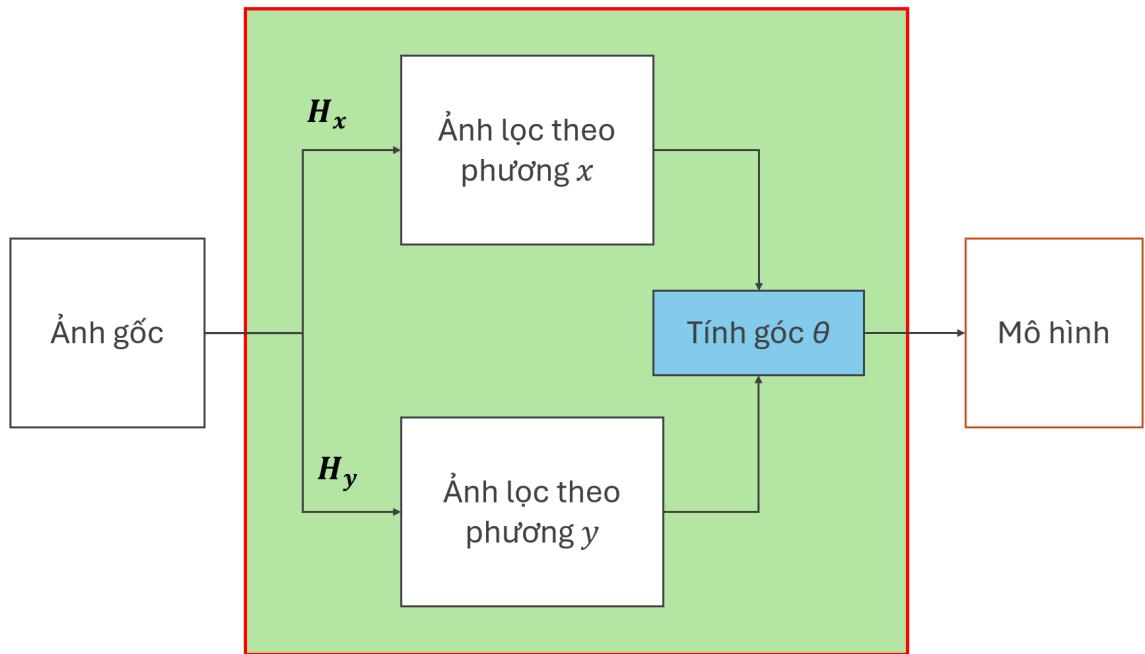
Trong cài đặt, hệ số $\pm \frac{1}{2}$ được thay bằng ± 1 để đơn giản hóa tính toán. Thực chất đây chỉ là nhân toàn bộ bộ lọc với một hằng số, nên kết quả convolution chỉ thay đổi về thang đo mà không ảnh hưởng đến đặc trưng. Đây là cách chuẩn hoá hệ số thường dùng trong mạng nơ-ron tích chập.

$$H_x = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & -1 \\ 0 & 0 & 0 \end{bmatrix} \quad H_y = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & -1 & 0 \end{bmatrix}$$

Để kết hợp thông tin từ hai bộ lọc, ta sử dụng kết quả lọc theo phương x (nhân chập ảnh với bộ lọc H_x) và phương y (nhân chập ảnh với bộ lọc H_y). Hai thành phần này có thể xem như hai cạnh của một vectơ trong hệ tọa độ (x, y) . Khi đó, góc θ của vectơ này được tính theo công thức:

$$\theta = \arctan \left(\frac{H_y}{H_x} \right).$$

Cách tính này giúp ta gộp thông tin của hai hướng lại thành một giá trị duy nhất, giúp đơn giản hoá đầu vào cho mạng nơ-ron.



Hình 3.3: Khối tiền xử lý dựa trên bộ lọc thông cao đã thiết kế

Ngoài ra, phép nhân chập ảnh I với bộ lọc H_x là tương đương với việc lấy hiệu giữa hai ảnh con: một ảnh được tạo bởi các cột từ 0 đến $n - 2$ và một ảnh được tạo bởi các cột từ 1 đến $n - 1$. Tương tự, phép nhân chập với H_y cũng có thể biểu diễn bằng hiệu của hai ảnh con theo chiều y .

Với bộ lọc H_x ta có:

$$(H_x * I)(x, y) = I(x, y) - I(x + 1, y).$$

Với bộ lọc H_y ta có:

$$(H_y * I)(x, y) = I(x, y) - I(x, y + 1).$$

Cách triển khai này giúp giảm khối lượng tính toán vì ta chỉ cần dịch ảnh và

trừ trực tiếp, thay vì thực hiện phép nhân chập ma trận.

3.4 Mô hình đề xuất

Trong đề tài này, mục tiêu hướng đến là xây dựng một phương pháp có cấu trúc đơn giản, tối ưu về tốc độ xử lý, và phù hợp với các hệ thống có tài nguyên hạn chế trong nhiệm vụ phát hiện hình ảnh tạo sinh. Để đạt được mục tiêu này, kiến trúc ResNet [27] được lựa chọn làm nền tảng cho mô hình phân loại.

ResNet [27] là một trong những kiến trúc CNN hiệu quả nhất được đề xuất bởi He et al. (2015), nổi bật nhờ cơ chế skip connection, cơ chế này cho phép huấn luyện các mô hình có chiều sâu lớn hơn mà vẫn duy trì hiệu quả hội tụ, đồng thời hạn chế hiện tượng mất mát thông tin trong quá trình lan truyền ngược.

So với các kiến trúc phức tạp hơn như EfficientNet [44], Vision Transformer [45], ResNet có ưu điểm là dễ triển khai, ít tham số hơn, và có thể tùy chỉnh dễ dàng số lượng tầng để phù hợp với yêu cầu tính toán của hệ thống. Trong phạm vi nghiên cứu này, một phiên bản rút gọn từ ResNet-50 [27] được sử dụng nhằm đảm bảo cân bằng giữa độ chính xác và chi phí tính toán.

Bên cạnh đó, ResNet [27] cũng cho thấy khả năng trích xuất đặc trưng tốt ở các tác vụ phân biệt hình ảnh có sai khác nhỏ, chẳng hạn như sự khác biệt về mô giữa ảnh thật và ảnh tạo sinh – điều này đặc biệt quan trọng trong bối cảnh deepfake ngày càng tinh vi.

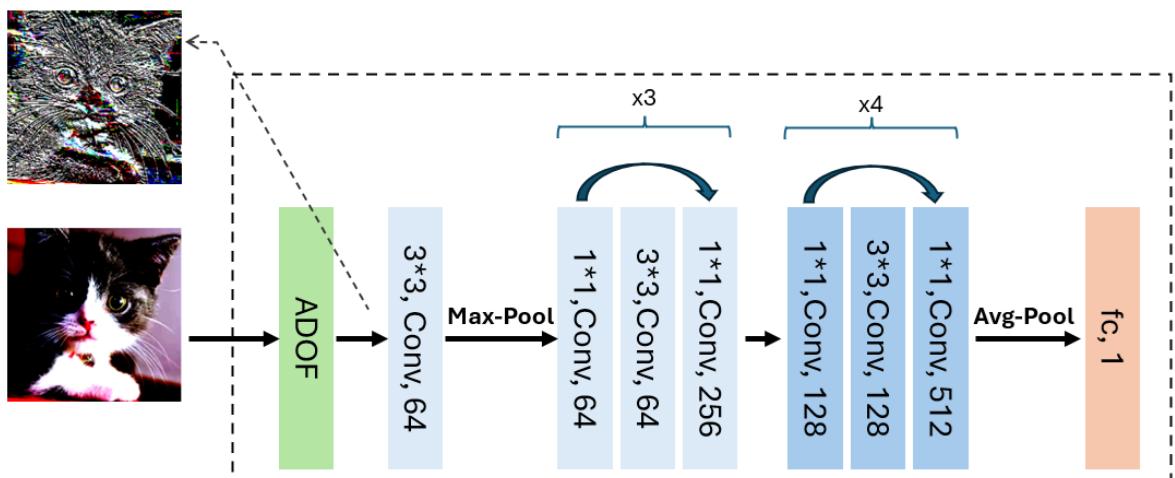
3.4.1 Kiến trúc mô hình rút gọn từ ResNet-50

Mô hình đề xuất được mô tả trong Hình 3.4), trong đó chỉ sử dụng hai khối đầu tiên của kiến trúc ResNet-50 (tương ứng với Layer1 và Layer2) để thực hiện trích xuất đặc trưng, kết hợp với một bộ phân loại nhị phân đơn giản ở

đầu ra. Việc tinh giản này giúp giảm số lượng tham số từ khoảng 23 triệu (của ResNet-50 đầy đủ) xuống còn khoảng 1.4 triệu tham số.

Lựa chọn này có cơ sở ở chỗ các dấu hiệu phân biệt giữa ảnh thật và ảnh tạo sinh thường nằm ở các đặc trưng mức thấp (*low-level features*) như biên, cạnh, và kết cấu cục bộ, trong khi các tầng sâu hơn của ResNet-50 chủ yếu học đặc trưng mức cao (*high-level semantics*) phục vụ nhận dạng đối tượng, vốn ít liên quan đến nhiệm vụ phát hiện deepfake. Việc chỉ giữ lại hai khối đầu vừa giúp tập trung vào đặc trưng phù hợp với bản chất của bài toán, vừa tránh hiện tượng *overfitting* do mô hình quá lớn so với dữ liệu huấn luyện, đồng thời giảm đáng kể chi phí tính toán và độ trễ suy luận. Cách tiếp cận này cũng phù hợp với các báo cáo gần đây trong lĩnh vực phát hiện ảnh tạo sinh, trong đó nhấn mạnh vai trò quan trọng của đặc trưng mức thấp và tín hiệu miền tần số trong việc nhận diện.

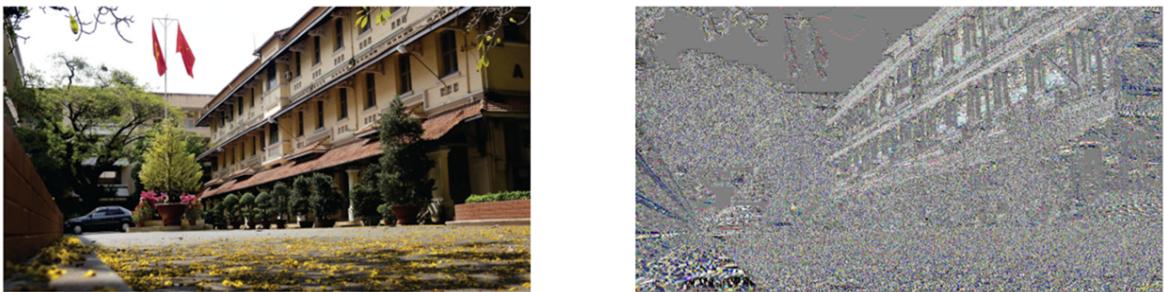
Mô hình bao gồm các thành phần chính như sau:



Hình 3.4: Kiến trúc mạng Resnet-50 rút gọn

- **Khối tiền xử lý - ADOF:**

Thành phần này đã được trình bày tại mục 3.3.



Hình 3.5: Hình ảnh trước và sau khi được xử lý bằng khối ADOF

- **Tầng đầu vào:**

- Một lớp convolution với kernel kích thước 3×3 , stride = 2, padding = 1, không dùng bias.
- Tiếp theo là lớp batch normalization và hàm kích hoạt ReLU (Rectified Linear Unit).
- Sau cùng là lớp max pooling với kernel 3×3 , stride = 2, padding = 1.

- **Tầng layer1:**

- Gồm 3 khối bottleneck, được kết nối bằng skip connection, nhằm học các đặc trưng cơ bản như biên cạnh, vùng chuyển tiếp và kết cấu bề mặt ảnh.

- **Tầng layer2:**

- Gồm 4 khối bottleneck, tiếp tục trích xuất đặc trưng ở mức trừu tượng cao hơn, giúp mô hình nhận diện các khác biệt tinh vi giữa ảnh thật và ảnh tạo sinh.

- **Bộ phân loại đầu ra:**

- Một lớp global average pooling để chuyển đổi bản đồ đặc trưng thành vector đặc trưng cố định chiều.
- Một lớp fully connected layer với một nút đầu ra duy nhất, sử dụng hàm kích hoạt sigmoid để phân loại ảnh đầu vào là thật hay tạo sinh.

3.4.2 Hàm mất mát

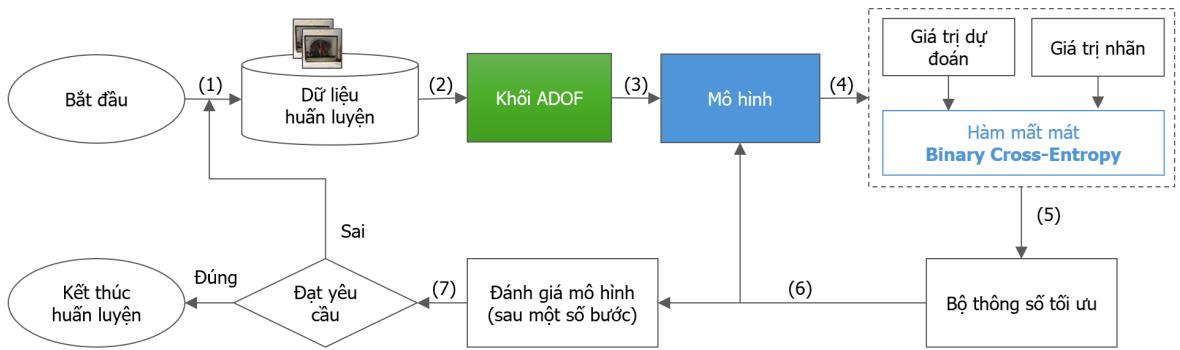
Để huấn luyện mô hình phân loại ảnh thật và ảnh tạo sinh, luận văn sử dụng hàm mất mát nhị phân binary cross-entropy, được định nghĩa như sau:

$$\mathcal{L}_{\text{BCE}} = -\frac{1}{N} \sum_{i=1}^N [y_i \log(\hat{y}_i) + (1 - y_i) \log(1 - \hat{y}_i)] \quad (3.3)$$

Trong đó, $y_i \in \{0, 1\}$ là nhãn thực tế của ảnh đầu vào, và $\hat{y}_i \in (0, 1)$ là xác suất do mô hình dự đoán.

3.4.3 Quy trình huấn luyện và sử dụng

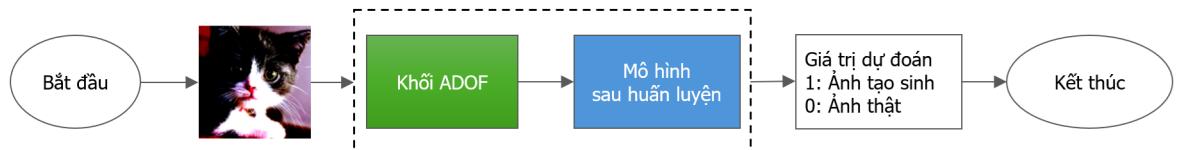
Quá trình huấn luyện được mô tả trong Hình 3.6, ảnh đầu vào sẽ được chuẩn hóa và tiền xử lý nhằm đồng nhất kích thước, tăng tính đa dạng qua các kỹ thuật tăng cường dữ liệu xoay và lật ảnh. Sau bước này, ảnh được đưa qua khối ADOF – có chức năng lọc lấy những tín hiệu ở tần số cao trong ảnh, nơi mà các mô hình tạo sinh thường để lại dấu vết. Đầu ra từ khối ADOF sau đó được đưa vào một kiến trúc ResNet-50 [27] đã được rút gọn và điều chỉnh nhằm trích xuất đặc trưng và thực hiện phân loại. Kiến trúc này được giảm bớt số lượng tầng so với phiên bản đầy đủ để giảm chi phí tính toán nhưng vẫn giữ lại khả năng biểu diễn cần thiết cho bài toán nhị phân. Đầu ra cuối cùng của mạng là một giá trị xác suất thể hiện mức độ tin tưởng rằng ảnh đầu vào là ảnh tạo sinh. Hàm mất mát được sử dụng trong quá trình huấn luyện là binary cross-entropy, kết hợp với thuật toán tối ưu adam. Sự lựa chọn này đặc biệt phù hợp với bài toán



Hình 3.6: Quy trình huấn luyện

phân loại nhị phân, giúp mô hình hội tụ nhanh và ổn định.

Quá trình huấn luyện được tổ chức thành nhiều bước cập nhật trọng số trong mỗi epoch. Mỗi bước tương ứng với một batch dữ liệu được đưa vào mô hình để tính toán hàm mất mát và cập nhật trọng số thông qua lan truyền ngược. Sau khi hoàn tất một epoch – tức là toàn bộ tập huấn luyện đã được xử lý – mô hình sẽ được đánh giá trên tập dữ liệu kiểm thử nhằm theo dõi hiệu năng tổng thể và điều chỉnh chiến lược huấn luyện nếu cần thiết. Quá trình này tiếp tục cho đến khi mô hình đạt đến số epoch tối đa đã định hoặc khi các chỉ số đánh giá dừng cải thiện trong một khoảng thời gian xác định trước. Quy trình huấn luyện như trên giúp mô hình học được các đặc trưng phân biệt hiệu quả giữa ảnh thật và ảnh tạo sinh, đồng thời hạn chế hiện tượng overfitting.



Hình 3.7: Quy trình sử dụng

Sau khi quá trình huấn luyện hoàn tất, mô hình đã sẵn sàng cho việc dự đoán

trên dữ liệu mới (chi tiết được mô tả trong Hình 3.7). Trong giai đoạn này, ảnh đầu vào sẽ được chuẩn hoá theo đúng các thông số đã sử dụng trong quá trình huấn luyện nhằm đảm bảo tính nhất quán về phân phối dữ liệu. Tiếp theo, ảnh được đưa qua khối ADOF để trích xuất các tín hiệu tần số cao. Đầu ra từ khối này được chuyển vào mô hình đã huấn luyện để thực hiện suy luận và đưa ra kết quả dự đoán cuối cùng.

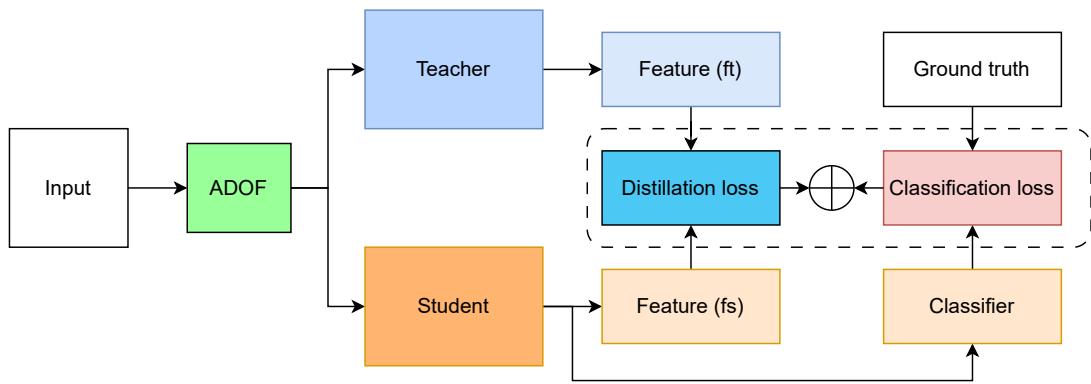
3.5 Tối giản mô hình với kỹ thuật feature-based knowledge distillation

Nhằm tối ưu khả năng triển khai thực tế trên các thiết bị có tài nguyên hạn chế, sau khi hoàn tất huấn luyện mô hình với kiến trúc trình bày ở phần 3.4.1, ta tiếp tục áp dụng kỹ thuật feature-based knowledge distillation để rút gọn mô hình.

Cụ thể, mô hình đã huấn luyện sẽ đóng vai trò là mô hình teacher, từ đó truyền đạt kiến thức cho một mô hình student nhỏ gọn hơn. Thay vì chỉ dựa vào đầu ra cuối cùng, mô hình student được hướng dẫn học theo các đặc trưng trung gian từ teacher, giúp duy trì hiệu suất nhận dạng trong khi giảm đáng kể kích thước và chi phí suy luận của mô hình.

3.5.1 Kiến trúc mô hình student

Trong mô hình student, ta chỉ giữ lại một khối bottleneck tại Layer1 (thay vì ba khối) và một khối bottleneck tại Layer2 (thay vì bốn khối) từ mô hình teacher (Hình 3.4). Việc lựa chọn số lượng layer được giữ lại trong mô hình student là kết quả của quá trình thử nghiệm thực tế, với tiêu chí cân bằng giữa độ chính xác và độ phức tạp của mô hình.



Hình 3.8: Sơ đồ kiến trúc mô hình Teacher-Student

3.5.2 Hàm mất mát

Hàm mất mát trong quá trình huấn luyện mô hình student là sự kết hợp của hai thành phần:

- 1. Hàm mất mát phân loại (Classification Loss):** Là hàm binary cross-entropy, được sử dụng trực tiếp để huấn luyện mô hình student theo nhãn thật. Chi tiết về binary cross-entropy đã được trình bày tại phần 3.4.2.
- 2. Hàm mất mát lan truyền tri thức (Distillation Loss):** Thành phần này được sử dụng để truyền đạt tri thức từ mô hình teacher sang student thông qua việc so sánh các biểu diễn đặc trưng tại một lớp trung gian, cụ thể là đầu ra của Layer2. Để đo lường mức độ sai khác giữa hai biểu diễn này, ta sử dụng hàm mất mát Mean Squared Error (MSE).

$$\mathcal{L}_{\text{distill}} = \frac{1}{N} \sum_{i=1}^N \|f_i^{(T)} - f_i^{(S)}\|^2 \quad (3.4)$$

trong đó, $f_i^{(T)}$ và $f_i^{(S)}$ lần lượt là véc-tơ đặc trưng đầu ra tại lớp trung gian của mô hình teacher và student ứng với mẫu dữ liệu thứ i .

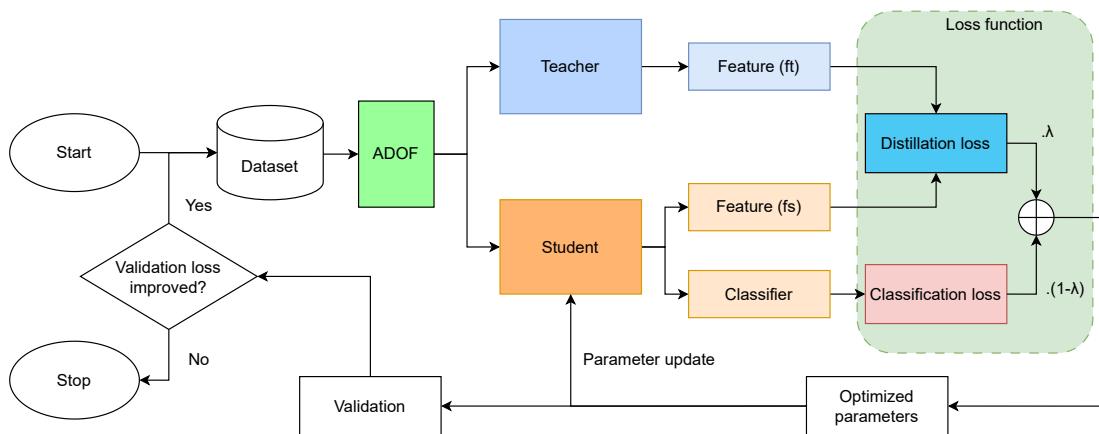
Tổng hàm mất mát được tính theo công thức:

$$\mathcal{L}_{\text{total}} = \lambda \cdot \mathcal{L}_{\text{distill}} + (1 - \lambda) \cdot \mathcal{L}_{\text{BCE}} \quad (3.5)$$

trong đó $\lambda \in [0, 1]$ là siêu tham số điều chỉnh mức độ ảnh hưởng của mỗi thành phần trong tổng hàm mất mát. Việc lựa chọn giá trị λ phù hợp được xác định thông qua quá trình hiệu chỉnh và thử nghiệm thực tế.

3.5.3 Quy trình huấn luyện và sử dụng

Quy trình huấn luyện trải qua các bước tương tự với như phần 3.4.3. Điểm khác biệt nằm ở quá trình lan truyền thuận, hàm mất mát và việc cập nhật bộ trọng số cho mô hình.



Hình 3.9: Quy trình huấn luyện mô hình áp dụng kỹ thuật Knowledge-Distillation

- Lan truyền thuận: Hình ảnh từ đầu vào sẽ được đi qua cả hai mô hình teacher và student. Sau đó ta trích xuất véc-tơ đặc trưng ở đầu ra của Layer2 trong mỗi mô hình và đưa vào hàm mất mát.
- Tối ưu hàm mất mát: Là sự kết hợp của 2 hàm mất mát thành phần distillation và classification.

- Cập nhật trọng số: Trong suốt quá trình huấn luyện, mô hình teacher chỉ sử dụng duy nhất một bộ trọng số đã được huấn luyện trước. Việc cập nhật bộ trọng số chỉ được áp dụng cho mô hình student.

Sau khi quá trình huấn luyện hoàn tất, mô hình student đã sẵn sàng cho việc dự đoán trên dữ liệu mới (chi tiết xem trên Hình 3.7).

CHƯƠNG 4. THỰC NGHIỆM VÀ ĐÁNH GIÁ KẾT QUẢ

4.1 Bộ dữ liệu ForenSynths

Bộ dữ liệu **ForenSynths** [46] được sử dụng trong quá trình nghiên cứu và huấn luyện. Các hình ảnh thật chủ yếu được trích từ bộ dữ liệu LSUN [47], ImageNet [48], trong khi các hình ảnh giả mạo được tạo ra từ các mô hình GAN [3] khác nhau (ProGAN [49], StyleGAN [36], CycleGAN [50], GauGAN [51], Deepfakes [16] và nhiều mô hình khác) chi tiết về bộ dữ liệu được thống kê trong Bảng 4.1. Đây là bộ dữ liệu được nhiều nghiên cứu sử dụng cho nhiệm vụ phát hiện ảnh tạo sinh.

4.1.1 Thu thập và xử lý hình ảnh

Phương pháp thu thập và xử lý dữ liệu ảnh thật và ảnh tạo sinh được Wang [46] thực hiện với các bước chính như sau:

1. Ảnh tạo sinh được tạo ra từ các mô hình sinh ảnh mà không áp dụng xử lý hậu kỳ bổ sung. Nếu bộ ảnh tạo sinh đã được phát hành chính thức, ảnh được tải về trực tiếp.
2. Số lượng ảnh thật được chọn bằng với số lượng ảnh tạo sinh, lấy từ tập huấn luyện tương ứng của từng loại mô hình.
3. Ảnh thật được tiền xử lý theo quy trình được chỉ định bởi từng mô hình, nhằm làm cho phân phối ảnh thật và giả càng giống nhau càng tốt.

4. Độ phân giải ảnh được chuẩn hóa như sau:

- Đối với các mô hình có đầu ra độ phân giải 256×256 (CycleGAN, StarGAN, ProGAN LSUN, GauGAN COCO, IMLE), kích thước ảnh được giữ nguyên.
- Với mô hình tạo ảnh có độ phân giải thấp hơn (DeepFake), ảnh được phóng to bằng phép bilinear sao cho cạnh ngắn bằng 256, giữ nguyên tỉ lệ khung hình.
- Với mô hình tạo ảnh có độ phân giải cao hơn (ProGAN, StyleGAN, SAN, SITD), ảnh giữ nguyên độ phân giải gốc.

5. Ảnh được cắt thành các phần 224×224 pixel:

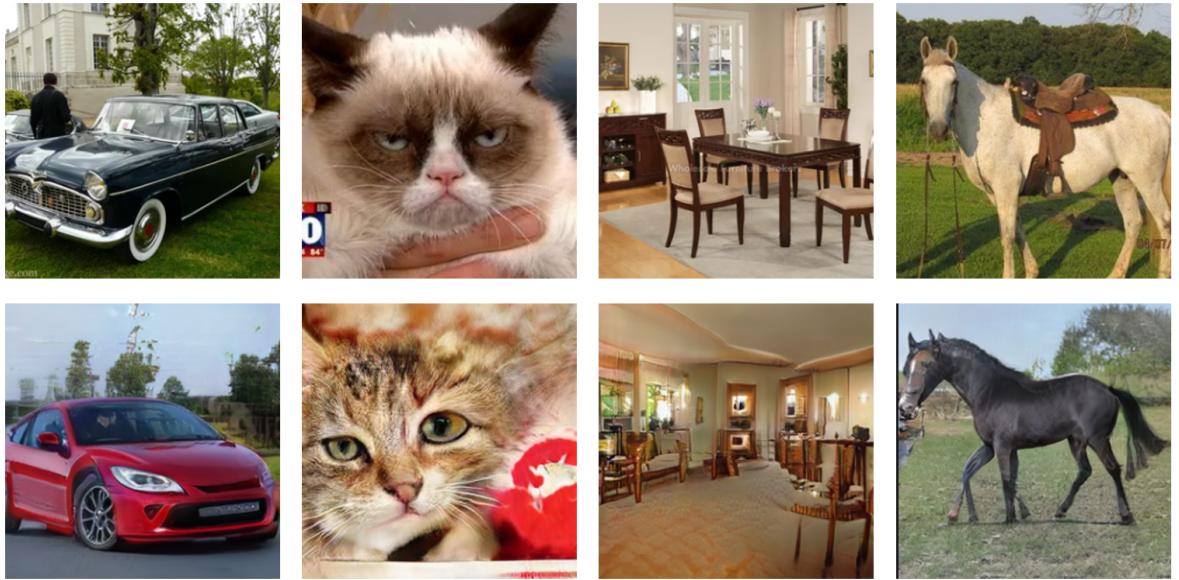
- Cắt ngẫu nhiên trong quá trình huấn luyện.
- Cắt phần trung tâm trong quá trình kiểm thử.

4.1.2 Tập train: Ảnh tạo sinh từ mô hình ProGAN

Các hình ảnh sinh bởi mô hình ProGAN [49] trong bộ dữ liệu **Forensynths** [46] được sử dụng cho quá trình huấn luyện của luận văn. Cụ thể, tập dữ liệu này bao gồm 20 lớp đối tượng (*airplane, bicycle, bird, boat, bottle, bus, car, cat, chair, cow, dining table, dog, horse, motorbike, person, potted plant, sheep, sofa, train, tv-monitor*), trong đó mỗi lớp có 18.000 hình ảnh thật được trích từ bộ dữ liệu LSUN [37], 18.000 hình ảnh tạo sinh bằng mô hình ProGAN [49], với vector nhiễu đầu vào \mathbf{z} được lấy mẫu từ phân phối chuẩn đa chiều $\mathcal{N}(0, I)$.

Tuy nhiên, chỉ những hình ảnh thuộc bốn trong hai mươi danh mục đối tượng được lựa chọn để huấn luyện gồm: *car, cat, chair* và *horse*. Điều này đồng nghĩa với việc tập huấn luyện chính thức có 144.000 hình ảnh, trong đó 72.000 hình ảnh được sinh từ mô hình và 72.000 hình ảnh thật được lấy từ tập huấn luyện

của từng phương pháp tương ứng. Việc lựa chọn các lớp này tương tự với các nghiên cứu [52], [41], [40], với mục đích thuận tiện hơn trong việc so sánh kết quả.



Hình 4.1: Một vài hình ảnh trong tập dữ liệu huấn luyện (các hình ảnh thật ở hàng trên).

4.1.3 Tập validation: Ảnh tạo sinh từ mô hình ProGAN

Tương tự như tập train, các hình ảnh trong tập validation thuộc bốn trong hai mươi danh mục đối tượng gồm: *car*, *cat*, *chair* và *horse*.

4.1.4 Tập test: Ảnh tạo sinh từ 8 mô hình GAN khác nhau

Tập dữ liệu kiểm định được xây dựng từ bộ dữ liệu **ForenSynths** [46], bao gồm hình ảnh được tạo ra bởi tám mô hình sinh ảnh khác nhau: BigGAN [54], CycleGAN [50], Deepfakes [16], GauGAN [51], ProGAN [49], StarGAN [58], StyleGAN [36] và StyleGAN2[53].

Đối với mỗi mô hình, 200 ảnh tạo sinh và 200 ảnh thật tương ứng được chọn ngẫu nhiên, đảm bảo sự cân bằng giữa hai lớp. Việc sử dụng đa dạng các mô

Bảng 4.1: Mô tả tập dữ liệu **ForenSynths** được sử dụng trong quá trình tạo ảnh tổng hợp.

STT	Mô hình	Dữ liệu ảnh thật	Mô tả
1	ProGAN [49]	LSUN [37]	Các hình ảnh được sinh bởi 20 mô hình ProGAN khác nhau, tương ứng với 20 lớp đổi tượng, mỗi hình ảnh có kích thước 256×256 .
2	StyleGAN [36]	LSUN [37]	Gồm ba lớp đổi tượng: bedroom, cat, car; ảnh sinh có truncation 0.5; ảnh thật được thay đổi về kích thước tương ứng: 256×256 hoặc 512×384 theo đúng kích thước của ảnh sinh.
3	StyleGAN2[53]	LSUN [37]	Gồm bốn lớp đổi tượng: church, cat, horse, car; ảnh sinh có truncation 0.5; ảnh thật được thay đổi về kích thước tương ứng.
4	BigGAN [54]	ImageNet [48]	Các lớp đổi tượng trích theo phân phối đều; ảnh sinh có truncation 0.4; ảnh thực được cắt theo vùng vuông ở trung tâm, kích thước cạnh bằng cạnh ngắn của ảnh gốc, sau đó được thay đổi kích thước về 256×256 .
5	CycleGAN [50]	Cityscapes [55], CMP Fa-cade [56], UT Zap-pos50K [57], ImageNet [48], Internet	Gồm sáu lớp đổi tượng: apple, orange, horse, zebra, summer, winter
6	StarGAN [58]	CelebA [59]	Các ảnh sinh là khuôn mặt dựa trên dữ liệu khuôn mặt của nhiều người nổi tiếng trong tập CelebA.
7	GauGAN [51]	COCO [60]	Các ảnh sinh là những đổi tượng thường gấp (Có tất cả 80 lớp đổi tượng).
8	CRN [61]	GTA [62]	Các hình ảnh được trích xuất từ trò chơi Grand Theft Auto, gồm nhiều đổi tượng và khung cảnh.
9	IMLE [63]	GTA [62]	Các hình ảnh được trích xuất từ trò chơi Grand Theft Auto, gồm nhiều đổi tượng và khung cảnh.
10	SITD [64]	SITD [64]	Ảnh sinh từ mô hình SITD được huấn luyện trên dữ liệu hình ảnh chụp bằng máy ảnh Sony và Fuji.
11	SAN [65]	DIV2K [66]	Đổi tượng chủ yếu gồm các ảnh tự nhiên đa dạng như cảnh thiên nhiên, kiến trúc, vật thể hàng ngày, con người, cây cối.
12	DeepFake [67]	FaceForensics++ [68]	Hình ảnh thật gồm các ảnh khuôn mặt đã được cắt từ các video gốc trong tập dữ liệu.

hình sinh ảnh trong tập kiểm định giúp đánh giá khả năng tổng quát hoá của mô hình phát hiện ảnh tạo sinh trên các nguồn dữ liệu chưa từng được huấn luyện trực tiếp.

4.1.5 Dữ liệu sử dụng để đánh giá mô hình sau huấn luyện

Tập dữ liệu kiểm tra bao gồm các hình ảnh đã được sử dụng trong nhiều công trình nghiên cứu trước đó. Nhằm đánh giá khả năng khái quát của mô hình phát hiện giả mạo, các hình ảnh thật và hình ảnh tạo sinh, được sinh từ những mô hình thuộc họ GAN [3] và Diffusion [4], tương tự như cách tiếp cận trong nghiên cứu của Tan [52].

- **Hình ảnh từ 9 mô hình GAN thuộc bộ dữ liệu Self-Synthesis** [52], các hình ảnh tạo sinh được thu thập từ 9 mô hình GAN [3] khác nhau, bao gồm: AttGAN, BEGAN, CramerGAN, InfoMaxGAN, MMDGAN, RelGAN, S3GAN, SNGAN, STGAN. Tổng cộng gồm 36,000 hình ảnh, số lượng hình ảnh thật và giả mạo trong tập dữ liệu là ngang nhau.
- **Hình ảnh từ 8 mô hình Diffusions** [4] trong **DIRE** [69], tác giả sử dụng 8 mô hình Diffusion [4] khác nhau để sinh hình ảnh, bao gồm: ADM [70], DDPM [71], IDDPM [72], PNDM [73], LDM [74], Stable Diffusion v1 [74], Stable Diffusion v2 [74], Vqdiffusion [75]. Những mô hình này được huấn luyện trên tập ảnh LSUN-Bedroom [37] và ImageNet [48], hình ảnh thật cũng được trích từ hai tập dữ liệu này. Tổng số lượng hình ảnh là 464,000, trong đó 50% là hình ảnh thật.
- **Hình ảnh từ 4 mô hình Diffusions** trong **Ojha** [76], bao gồm 16,000 hình ảnh, trong đó các hình ảnh thật được trích xuất từ tập dữ liệu LAION [77], các ảnh tạo sinh được tạo từ 4 mô hình Diffusion [4] bao gồm: ADM [70], Glide [78], DALL-E-Mini [79], LDM [74]. Quá trình sinh ảnh sử dụng các câu mô tả hình ảnh thật tương ứng để cung cấp thông tin cho mô hình.

4.2 Cài đặt môi trường thực nghiệm

4.2.1 Chuẩn bị dữ liệu

Tập train: Hình ảnh gồm bốn lớp đối tượng: *car*, *cat*, *chair* và *horse*. Các ảnh tạo sinh được tạo ra bằng mô hình sinh ProGAN [49], trích xuất từ tập dữ liệu tổng hợp **ForenSynths** [46].

Tập validation: Tập dữ liệu được sử dụng trong quá trình huấn luyện nhằm theo dõi hiệu suất mô hình. Các hình ảnh trong tập này được trích xuất từ mô hình Progan Chi tiết tập dữ liệu được trình bày tại 4.1.2

Lý do chọn tập dữ liệu:

- Thuận lợi khi so sánh giữa các phương pháp vì đây là tập huấn luyện được nhiều nghiên cứu sử dụng như: Wang [46], NPR [52], LGrad [80], Ojha [76].
- Việc lựa chọn hình ảnh cho tập train chỉ từ một mô hình sinh ảnh duy nhất (ProGAN [49]) nhằm đánh giá khả năng khái quát và hiệu quả thực sự của phương pháp phát hiện. Cách tiếp cận này phản ánh bối cảnh thực tế, trong đó các mô hình sinh ảnh mới liên tục xuất hiện, trong khi dữ liệu huấn luyện thường đến từ các nguồn cũ và có tốc độ cập nhật chậm hoặc không được cập nhật.

4.2.2 Cấu hình phần cứng và cài đặt các tham số

Được thiết lập tương tự như các phương pháp [46, 52, 80] để giảm tác động của các yếu tố ngẫu nhiên đến kết quả cuối cùng, làm mất tính khách quan khi so sánh, đánh giá giữa nhiều phương pháp. Giá trị thiết lập các tham số bao gồm:

- Optimizer: Adam
- Learning rate: 2×10^{-4}

- Batch size: 64
- Framework: Các thử nghiệm được xây dựng bằng thư viện PyTorch.
- Cấu hình máy tính: CPU AMD Ryzen 5 5600X 6-Core, 1 × GPU NVIDIA RTX A4000, 1 × 16 GB bộ nhớ RAM.

4.2.3 Kết quả huấn luyện

Mô hình ngừng cải thiện trên tập kiểm tra sau 9 epoch, cho thấy quá trình huấn luyện đã đạt đến ngưỡng hội tụ. Do đó, mô hình tại epoch thứ 9 được chọn làm mô hình cuối cùng để đánh giá hiệu năng.

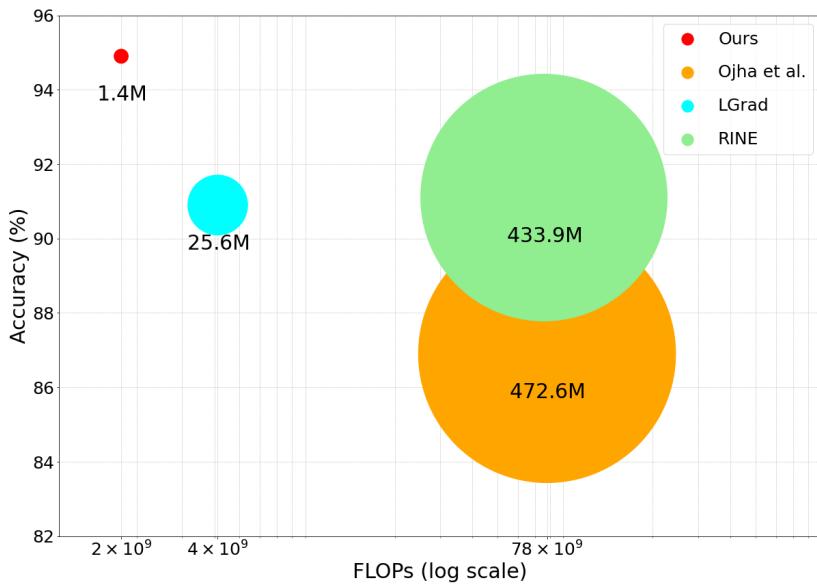
Bảng 4.2: Kết quả đánh giá trên tập kiểm tra ForensSynths

Giá trị thể hiện độ chính xác (accuracy, %).

Method	ProGAN	StyleGAN	StyleGAN2	BigGAN	CycleGAN	StarGAN	GauGAN	Deepfake	Mean
CNNDetection [46](2020)	91.4	63.8	76.4	52.9	72.7	63.8	63.9	51.7	67.1
Frank [35](2020)	90.3	74.5	73.1	88.7	75.5	99.5	69.2	60.7	78.9
Durall [81](2020)	81.1	54.4	66.8	60.1	69.0	98.1	61.9	50.2	67.7
Patchfor [82](2020)	97.8	82.6	83.6	64.7	74.5	100.0	57.2	85.0	80.7
F3Net [38](2020)	99.4	92.6	88.0	65.3	76.4	100.0	58.1	63.5	80.4
SelfBland [83](2022)	58.8	50.1	48.6	51.1	59.2	74.5	59.2	93.8	61.9
GANDetection [84](2022)	82.7	74.4	69.9	76.3	85.2	68.8	61.4	60.0	72.3
BiHPF [40](2021)	90.7	76.9	76.2	84.9	81.9	94.4	69.5	54.4	78.6
FrePGAN [41](2022)	99.0	80.7	84.1	69.2	71.1	99.9	60.3	70.9	79.4
LGrad [80](2023)	99.9	94.8	96.0	82.9	85.3	99.6	72.4	58.0	86.1
Ojha [76](2023)	99.7	89.0	83.9	90.5	87.9	91.4	89.9	80.2	89.1
NPR [52](2023)	99.8	96.3	97.3	87.5	95.0	99.7	86.6	77.4	92.5
ADOF(our)	99.8	99.6	99.5	81.1	86.1	96.8	73.4	86.0	90.3

4.3 Đánh giá mô hình

Nội dung đánh giá bao gồm hai phần chính: Thứ nhất, so sánh và đánh giá độ chính xác của nghiên cứu so với các phương pháp tiên tiến nay (Bảng 4.3,4.4,4.5). Thứ hai, phân tích hiệu quả sử dụng tài nguyên tính toán và hiệu năng của một số hướng tiếp cận tiêu biểu (xem Bảng 4.6).



Hình 4.2: Tổng quan hiệu năng và độ chính xác của một số hướng tiếp cận, trên tập dữ liệu Ojha [76].

4.3.1 So sánh, đánh giá khả năng phát hiện ảnh tạo sinh từ các phương pháp sinh ảnh khác nhau

Kết quả đánh giá trong mục này là một phần nội dung chính trong bài báo [85] đã được công bố tại hội nghị quốc tế SOICT 2024.

Luận văn thực hiện so sánh độ chính xác của phương pháp đề xuất với 10 phương pháp tiên tiến trên nhiều tập dữ liệu có phân phối khác so với tập dữ liệu huấn luyện, bao gồm: CNNDetection [46], Frank [35], Durall [81], Patchfor [82], F3Net [38], SelfBland [83], GANDetection [84], LGrad [80], Ojha [76], NPR [52]. Kết quả thí nghiệm trong các Bảng 4.3, 4.4, và 4.5 cho thấy phương pháp đề xuất vượt trội hơn so với các phương pháp hiện có.

Trên bộ dữ liệu 9-GAN, ADOF đạt độ chính xác cao nhất là 94.2%, vượt qua Ojha [76] với chỉ 77.6% [4.3], và NPR [52] đạt 93.2% (xem Bảng 4.3).

Độ chính xác đạt 98.3% trên tập dữ liệu DiffusionForensics [69], cao nhất trong 10 phương pháp, trong khi đó NPR [52] giữ vị trí thứ hai với 95.3% (xem Bảng 4.4). Kết quả này cao hơn DIRE [69] với 97.9% ngay trên bộ dữ liệu

của chính họ, mặc dù mô hình đề xuất trong luận văn được huấn luyện trên Forensynths [25], trong khi DIRE được huấn luyện trên DiffusionForensics.

Ngoài ra, độ chính xác cũng trội hơn RINE [86] và Ojha [76](91.1%) (xem Bảng 4.5). Đáng chú ý, cả hai phương pháp này đều sử dụng mô hình CLIP [87] có số lượng tham số rất lớn, yêu cầu nhiều tài nguyên tính toán.

Bảng 4.3: Kết quả đánh giá trên tập Self-Synthesis 9 GANs [52].

Method	AttGAN		BEGAN		CramerGAN		InfoMaxGAN		MMDGAN		RelGAN		S3GAN		SNGAN		STGAN		Mean	
	Acc.	A.P.	Acc.	A.P.	Acc.	A.P.	Acc.	A.P.	Acc.	A.P.	Acc.	A.P.	Acc.	A.P.	Acc.	A.P.	Acc.	A.P.	Acc.	A.P.
CNNDetection [46]	51.1	83.7	50.2	44.9	81.5	97.5	71.1	94.7	72.9	94.4	53.3	82.1	55.2	66.1	62.7	90.4	63.0	92.7	62.3	82.9
Frank [35](2020)	65.0	74.4	39.4	39.9	31.0	36.0	41.1	41.0	38.4	40.5	69.2	96.2	69.7	81.9	48.4	47.9	25.4	34.0	47.5	54.7
Durall [81](2020)	39.9	38.2	48.2	30.9	60.9	67.2	50.1	51.7	59.5	65.5	80.0	88.2	87.3	97.0	54.8	58.9	62.1	72.5	60.3	63.3
Patchfor [82](2020)	68.0	92.9	97.1	100.0	97.8	99.9	93.6	98.2	97.9	100.0	99.6	100.0	66.8	68.1	97.6	99.8	92.7	99.8	90.1	95.4
F3Net [38](2020)	85.2	94.8	87.1	97.5	89.5	99.8	67.1	83.1	73.7	99.6	98.8	100.0	65.4	70.0	51.6	93.6	60.3	99.9	75.4	93.1
SelfBland [83](2022)	63.1	66.1	56.4	59.0	75.1	82.4	79.0	82.5	68.6	74.0	73.6	77.8	53.2	53.9	61.6	65.0	61.2	66.7	65.8	69.7
GANDetection [84]	57.4	75.1	67.9	100.0	67.8	99.7	67.6	92.4	67.7	99.3	60.9	86.2	69.6	83.5	66.7	90.6	69.6	97.2	66.1	91.6
LGrad [80](2023)	68.6	93.8	69.9	89.2	50.3	54.0	71.1	82.0	57.5	67.3	89.1	99.1	78.5	86.0	78.0	87.4	54.8	68.0	68.6	80.8
Ojha [76](2023)	78.5	98.3	72.0	98.9	77.6	99.8	77.6	98.9	77.6	99.7	78.2	98.7	85.2	98.1	77.6	98.7	74.2	97.8	77.6	98.8
NPR [52](2023)	83.0	96.2	99.0	99.8	98.7	99.0	94.5	98.3	98.6	99.0	99.6	100.0	79.0	80.0	88.8	97.4	98.0	100.0	93.2	96.6
ADOF(ours)	99.5	100.0	92.2	100.0	96.0	99.6	94.1	99.1	96.0	99.7	100.0	100.0	77.5	86.7	94.8	99.3	97.8	99.7	94.2	98.2

Bảng 4.4: Kết quả đánh giá trên tập DiffusionForensics [69].

Method	ADM				DDPM				IDDPMP				LDM				PNDM				VQ-Diffusion				Stable Diffusion v1		Stable Diffusion v2		Mean	
	Acc.	A.P.	Acc.	A.P.	Acc.	A.P.	Acc.	A.P.	Acc.	A.P.	Acc.	A.P.	Acc.	A.P.	Acc.	A.P.	Acc.	A.P.	Acc.	A.P.	Acc.	A.P.	Acc.	A.P.	Acc.	A.P.	Acc.	A.P.		
CNNDetection [46]	53.9	71.8	62.7	76.6	50.2	82.7	50.4	78.7	50.8	90.3	50.0	71.0	38.0	76.7	52.0	90.3	51.0	79.8												
Frank [35]	58.9	65.9	37.0	27.6	51.4	65.0	51.7	48.5	44.0	38.2	51.7	66.7	32.8	52.3	40.8	37.5	46.0	50.2												
Durall [81]	39.8	42.1	52.9	49.8	55.3	56.7	43.1	39.9	44.5	47.3	38.6	38.3	39.5	56.3	62.1	55.8	47.0	48.3												
Patchfor [82]	77.5	93.9	62.3	97.1	50.0	91.6	99.5	100.0	50.2	99.9	100.0	100.0	90.7	99.8	94.8	100.0	78.1	97.8												
F3Net [38]	80.9	96.9	84.7	99.4	74.7	98.9	100.0	100.0	72.8	99.5	100.0	100.0	73.4	97.2	99.8	100.0	85.8	99.0												
SelfBland [83]	57.0	59.0	61.9	49.6	63.2	66.9	83.3	92.2	48.2	48.2	48.2	77.2	82.7	46.2	68.0	71.2	73.9	63.5	67.6											
GANDetection [84]	51.1	53.1	62.3	46.4	50.2	63.0	51.6	48.1	50.6	79.0	51.1	51.2	39.8	65.6	50.1	36.9	50.8	55.4												
LGrad [80]	86.4	97.5	99.9	100.0	66.1	92.8	99.7	100.0	69.5	98.5	96.2	100.0	90.4	99.4	97.1	100.0	88.2	98.5												
Ojha [76]	78.4	92.1	72.9	78.8	75.0	92.8	82.2	97.1	75.3	92.5	83.5	97.7	56.4	90.4	71.5	92.4	74.4	91.7												
NPR [52]	88.6	98.9	99.8	100.0	91.8	99.8	100.0	100.0	91.2	100.0	100.0	100.0	97.4	99.8	93.8	100.0	95.3	99.8												
ADOF(ours)	93.5	99.0	99.6	100.0	99.2	100.0	99.9	100.0	97.4	99.9	97.1	99.8	99.8	100.0	99.9	100.0	98.3	99.8												

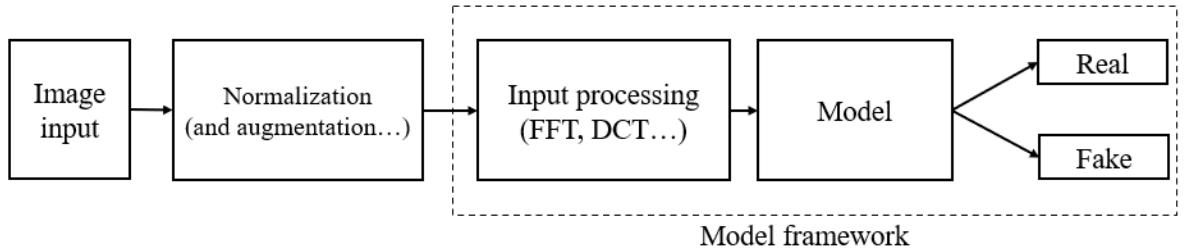
Bảng 4.5: Kết quả đánh giá trên tập Ojha [76].

Method	DALLE				Glide _(100, 10)				Glide _(100, 27)				ADM				LDM ₍₁₀₀₎				LDM ₍₂₀₀₎				LDM _(200, cfg)				Mean	
	Acc.	A.P.	Acc.	A.P.	Acc.	A.P.	Acc.	A.P.	Acc.	A.P.	Acc.	A.P.	Acc.	A.P.	Acc.	A.P.	Acc.	A.P.	Acc.	A.P.	Acc.	A.P.	Acc.	A.P.	Acc.	A.P.	Acc.	A.P.		
CNNDetection [46]	51.8	61.3	53.3	72.9	53.0	71.3	54.2	76.0	54.9	66.6	51.9	63.7	52.0	64.5	51.6	63.1	52.8	67.4												
Frank [35]	57.0	62.5	53.6	44.3	50.4	40.8	52.0	42.3	53.4	52.5	56.6	51.3	56.4	50.9	56.5	52.1	54.5	49.6												
Durall [81]	55.9	58.0	54.9	52.3	48.9	46.9	51.7	49.9	40.6	42.3	62.0	62.6	61.7	61.7	58.4	58.5	54.3	54.0												
Patchfor [82]	79.8	99.1	87.3	99.7	82.8	99.1	84.9	98.8	74.2	81.4	95.8	99.8	95.6	99.9	94.0	99.8	86.8	97.2												
F3Net [38]	71.6	79.9	88.3	95.4	87.0	94.5	88.5	95.4	69.2	70.8	74.1	84.0	73.4	83.3	80.7	89.1	79.1	86.5												
SelfBland [83]	52.4	51.6	58.8	63.2	59.4	64.1	64.2	68.3	58.3	63.4	53.0	54.0	52.6	51.9	52.6	52.6	56.3	58.7												
GANDetection [84]	67.2	83.0	51.2	52.6	51.1	51.9	51.7	53.5	49.6	49.0	54.7	65.8	54.9	65.9	53.8	58.9	54.3	60.1												
LGrad [80]	88.5	97.3	89.4	94.9	87.4	93.2	90.7	95.1	86.6	100.0	94.8	99.2	94.2	99.1	95.9	99.2	90.9	97.2												
Ojha [76]	89.5	96.8	90.1	97.0	97.2	91.1	97.4	75.7	85.1	90.5	97.0	90.2	97.1	77.3	88.6	86.9	94.5													
NPR [52]	94.5	99.5	98.2	99.8	97.8	99.7	98.2	99.8	75.8	81.0	99.3	99.9	99.1	99.9	99.0	99.8	98.6	98.7	91.1	99.0										
RINE [86](2024)	95.0	99.5	90.7	99.2	88.9	99.1	92.6	99.5	76.1	96.6	98.7	99.9	98.3	99.9	88.2	98.7	91.1	99.0	94.9	98.2										
ADOF(ours)	92.1	98.3	98.6	100.0	98.7	100.0	98.4	99.9	75.9	87.6	98.8	100.0	98.6	99.9	98.5	99.9	94.9	98.2												

4.3.2 So sánh và đánh giá về hiệu năng

Để so sánh về hiệu năng, luận văn thực hiện đo lường các tiêu chí về độ phức tạp của mô hình cũng như số lượng phép toán cần thiết cho mỗi dự đoán (chi

tiết xem Bảng 4.6). Tất cả được thực hiện trên máy tính có CPU AMD Ryzen 5 5600X 6-Core, GPU NVIDIA RTX A4000, 16 GB bộ nhớ RAM. Hình ảnh đầu vào có kích thước $256 \times 256 \times 3$. Các tiêu chí đánh giá bao gồm:



Hình 4.3: Quy trình cơ bản của các phương pháp phát hiện ảnh tạo sinh bằng mạng học sâu

- **Number of Parameters:** Số lượng tham số của mô hình, thể hiện độ phức tạp trong kiến trúc.
- **Input Processing Time:** Đo lường thời gian cần thiết để xử lý hình ảnh trước khi đưa vào mô hình dự đoán, các bước xử lý này thay đổi theo hướng tiếp cận cụ thể (xem Hình. 4.3).
- **Inference Time:** Thời gian cần sử dụng cho một dự đoán.
- **FLOPs:** Số lượng phép tính dấu chấm động trong 1 giây, luận văn sử dụng thư viện `fvcore` để ước tính khối lượng tính toán mà mỗi mô hình cần thực hiện cho một dự đoán.

Bảng 4.6: Tài nguyên sử dụng và hiệu năng của các phương pháp phát hiện hình ảnh tổng hợp, trên tập dữ liệu DiffusionForensics [69]. Dấu \dagger thể hiện phương pháp đã được huấn luyện trên cùng tập dữ liệu.

Method	Parameters	Processing (ms)	Inference Time (ms)	FLOPs	Means (acc/ap)
LGrad [80] (2023)	25.56×10^6	11.6	4.81	4.12×10^9	88.2/98.5
DIRE \dagger [69] (2023)	25.56×10^6	4,502.7	4.81	4.12×10^9	97.9/100
Ojha [76] (2023)	427.62×10^6	None	29.19	77.83×10^9	74.4/91.7
ADOF(ours)	1.44×10^6	0.40	2.43	1.74×10^9	98.3/99.8

4.4 Kết quả tối giản mô hình student bằng phương pháp feature-based knowledge distillation

Trong phần này, chúng tôi trình bày kết quả thực nghiệm của mô hình student được xây dựng bằng cách rút gọn kiến trúc từ mô hình teacher thông qua phương pháp feature-based knowledge distillation. Mô hình teacher chính là kiến trúc đã được sử dụng và đánh giá trong công trình công bố tại hội nghị SOICT 2024 [85]. Mục tiêu của quá trình rút gọn là giảm thiểu độ phức tạp tính toán và kích thước mô hình, từ đó nâng cao khả năng triển khai trên các thiết bị có tài nguyên hạn chế, mà vẫn duy trì hiệu suất nhận diện ở mức chấp nhận được. Chi tiết về kiến trúc và quy trình huấn luyện đã được trình bày tại Mục 3.5.

Mô hình Student đạt tổng số tham số khoảng **456,771**, chỉ bằng 1/3 so với **1.44 triệu** tham số của mô hình Teacher. Điều này cho thấy tiềm năng đáng kể của phương pháp feature-based knowledge distillation trong việc tối ưu hóa mô hình mà không đánh đổi quá nhiều về mặt hiệu suất. Mặc dù số lượng tham

Bảng 4.7: So sánh mô hình Teacher và Student trên ba tập dữ liệu.

Bộ dữ liệu	Mô hình Student (acc/ap)	Mô hình Teacher (acc/ap)
GANGen-Detection	94.5 / 98.0	94.2 / 98.2
DiffusionForensics	98.1 / 99.8	98.3 / 99.8
UniversalFakeDetect	93.4 / 98.0	94.9 / 98.2

số giảm mạnh, mô hình Student vẫn đạt được hiệu suất đáng kể trên cả ba tập dữ liệu. Các kết quả trong Bảng 4.7 được ghi nhận sau khi huấn luyện mô hình Student trong **8 epoch**, sử dụng dữ liệu và chiến lược huấn luyện tương tự như mô hình Teacher. Mô hình Student đã duy trì được độ chính xác cao, đồng thời giảm đáng kể độ phức tạp, phù hợp với mục tiêu triển khai trên các thiết bị hạn chế tài nguyên.

CHƯƠNG 5. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

5.1 Kết luận

Luận văn đã đề xuất một khối tiền xử lý hình ảnh đơn giản nhưng hiệu quả, được đặt tên là ADOF, nhằm khai thác các biến thiên mức xám cục bộ giữa các điểm ảnh lân cận. Bằng cách xem hình ảnh như một tín hiệu số rời rạc và áp dụng kỹ thuật sai phân, phương pháp đã loại bỏ các thành phần tần số thấp của tín hiệu — vốn thường mang tính ngữ nghĩa cao nhưng lại không hữu ích trong việc phân biệt giữa ảnh thật và ảnh tạo sinh.

Khối tiền xử lý tập trung vào việc làm nổi bật các dấu vết tinh vi còn lại, từ đó hỗ trợ mô hình học sâu trong việc cải thiện đáng kể cả về độ chính xác và khả năng tổng quát hóa, đồng thời giúp giảm độ phức tạp của mô hình. Kết quả thực nghiệm cho thấy phương pháp ADOF hoạt động hiệu quả ngay cả trên các tập dữ liệu chưa từng được thấy trước đó.

Đặc biệt, khi được tích hợp vào quy trình rút gọn mô hình – feature-based knowledge distillation, ADOF vẫn duy trì được lợi ích ban đầu của nó, giúp mô hình student đạt hiệu năng gần tương đương với mô hình gốc (teacher) nhưng với cấu trúc nhẹ hơn đáng kể. Điều này cho thấy tiềm năng ứng dụng của ADOF trong các hệ thống thực tế yêu cầu tính hiệu quả và khả năng triển khai cao trên thiết bị giới hạn tài nguyên.

5.2 Hướng phát triển

Dựa trên những kết quả khả quan mà khôi tiền xử lý ADOF mang lại, một số hướng phát triển tiềm năng trong tương lai bao gồm:

- **Mở rộng sang dữ liệu video:** Áp dụng bộ lọc này trong bối cảnh xử lý video – một loại dữ liệu mà tốc độ và độ chính xác đóng vai trò đặc biệt quan trọng – nhằm phát hiện kịp thời các nội dung giả mạo trong chuỗi khung hình liên tục.
- **Tích hợp vào các pipeline phát hiện khác nhau:** Khảo sát hiệu quả của bộ lọc khi được tích hợp như một bước tiền xử lý trong các hệ thống phát hiện hình ảnh tạo sinh sử dụng nhiều kiến trúc khác nhau. Mục tiêu là đánh giá tính tương thích và khả năng giúp mô hình nâng cao hiệu suất trong các kịch bản thực tế đa dạng.
- **Mở rộng ứng dụng trong pháp y hình ảnh:** Nghiên cứu áp dụng bộ lọc ADOF vào các kỹ thuật phát hiện thao tác chỉnh sửa ảnh truyền thống như cắt ghép, làm mờ, che giấu chi tiết, hoặc nguy trang nội dung.

DANH MỤC CÔNG TRÌNH CỦA TÁC GIẢ

1. *Minimalist Preprocessing Approach for Image Synthesis Detection*, Vo Hoai Danh, Le Trung Nghia. Trong: Buntine, W., Fjeld, M., Tran, T., Tran, M.T., Huynh Thi Thanh Binh, Miyoshi, T. (chủ biên). **Information and Communication Technology**. Hội thảo SOICT 2024. Communications in Computer and Information Science, tập 2350. Springer, Singapore, 2025.
https://doi.org/10.1007/978-981-96-4282-3_8

Tiếng Anh

- [1] Mieczysław Mastyło. “Bilinear interpolation theorems and applications”. In: *Journal of Functional Analysis* 265.2 (2013), pp. 185–207.
- [2] Giovanni Capobianco et al. “Image convolution: a linear programming approach for filters design”. In: *Soft Computing* 25 (July 2021). doi: 10.1007/s00500-021-05783-5.
- [3] Ian J. Goodfellow et al. “Generative adversarial networks”. In: *Communications of the ACM* 63 (2014), pp. 139–144. URL: <https://api.semanticscholar.org/CorpusID:1033682>.
- [4] Jonathan Ho, Ajay Jain, and P. Abbeel. “Denoising Diffusion Probabilistic Models”. In: *ArXiv* abs/2006.11239 (2020). URL: <https://api.semanticscholar.org/CorpusID:219955663>.
- [5] Patrick Esser et al. “Scaling Rectified Flow Transformers for High-Resolution Image Synthesis”. In: *ArXiv* abs/2403.03206 (2024). URL: <https://api.semanticscholar.org/CorpusID:268247980>.
- [6] OpenAI. *DALL·E 2*. <https://openai.com/index/dall-e-2/>. Accessed: 2024-08-21. 2024.
- [7] DeepArt. *DeepArt.io*. <https://www.artvy.ai/ai-tools/deepartio>. Accessed: 2024-08-21. 2024.
- [8] Joana Casteleiro-Pitrez. “Generative Artificial Intelligence Image Tools among Future Designers: A Usability, User Experience, and Emotional Analysis”. In: *Digit.* 4 (2024), pp. 316–332.
- [9] Philip Woontae Shin et al. “Can Prompt Modifiers Control Bias? A Comparative Analysis of Text-to-Image Generative Models”. In: *ArXiv* abs/2406.05602 (2024).
- [10] Ildar Lomov and Ilya Makarov. “Generative Models for Fashion Industry using Deep Neural Networks”. In: *2019 2nd International Conference on Computer Applications & Information Security (ICCAIS)*. 2019, pp. 1–6. doi: 10.1109/CAIS.2019.8769486.
- [11] Zixuan Chen and Xiang Wang. “Application of AI technology in interior design”. In: *E3S Web of Conferences* (2020). URL: <https://api.semanticscholar.org/CorpusID:226695888>.
- [12] *AI won an art contest, and artists are furious*. <https://edition.cnn.com/2022/09/03/tech/ai-art-fair-winner-controversy/index.html>. Accessed: 2024-03-26.
- [13] Luke A. Bauer and Vincent Bindschaedler. “Generative Models for Security: Attacks, Defenses, and Opportunities”. In: *CoRR* abs/2107.10139 (2021). arXiv: 2107.10139. URL: <https://arxiv.org/abs/2107.10139>.
- [14] *Finance worker pays out \$25 million after video call with deepfake*. <https://edition.cnn.com/2024/02/04/asia/deepfake-cfo-scam-hong-kong-intl-hnk/index.html>. Accessed: 2024-03-26.
- [15] Kirsten Korosec. “Deepfake revenge porn is now illegal in Virginia”. In: (2019). Accessed: 2024-09-24. URL: <https://techcrunch.com/2019/07/01/deepfake-revenge-porn-is-now-illegal-in-virginia/>.
- [16] Cara Curtis. “California makes deepfakes illegal to curb revenge porn and doctored political videos”. In: (2019). Accessed: 2024-09-24. URL: <https://thenextweb.com/news/california-makes-deepfakes-illegal-to-curb-revenge-porn-and-doctored-political-videos>.

- [17] Britannica for Schools. *Spotting AI: Knowing How to Recognise Real vs AI Images*. <https://elearn.eb.com/real-vs-ai-images/>. Accessed: 2024-08-21. 2024.
- [18] Anqi Mao, Mehryar Mohri, and Yutao Zhong. “Cross-Entropy Loss Functions: Theoretical Analysis and Applications”. In: *arXiv e-prints*, arXiv:2304.07288 (Apr. 2023), arXiv:2304.07288. doi: 10.48550/arXiv.2304.07288. arXiv: 2304.07288 [cs.LG].
- [19] Diederik P. Kingma and Max Welling. “Auto-Encoding Variational Bayes”. In: *CoRR* abs/1312.6114 (2013). URL: <https://api.semanticscholar.org/CorpusID:216078090>.
- [20] Jascha Sohl-Dickstein et al. “Deep Unsupervised Learning using Nonequilibrium Thermodynamics”. In: *Proceedings of the 32nd International Conference on Machine Learning*. Ed. by Francis Bach and David Blei. Vol. 37. Proceedings of Machine Learning Research. Lille, France: PMLR, July 2015, pp. 2256–2265. URL: <https://proceedings.mlr.press/v37/sohl-dickstein15.html>.
- [21] Simon J.D. Prince. *Understanding Deep Learning*. The MIT Press, 2023. URL: <http://udlbook.com>.
- [22] Scott McCloskey and Michael Albright. “Detecting GAN-Generated Imagery Using Saturation Cues”. In: *2019 IEEE International Conference on Image Processing (ICIP)*. Sept. 2019, pp. 4584–4588. doi: 10.1109/ICIP.2019.8803661.
- [23] Can Chen, Scott McCloskey, and Jingyi Yu. “Focus Manipulation Detection via Photometric Histogram Analysis”. In: *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2018), pp. 1674–1682. URL: <https://api.semanticscholar.org/CorpusID:53875533>.
- [24] Zhengzhe Liu, Xiaojuan Qi, and Philip H.S. Torr. “Global Texture Enhancement for Fake Face Detection in the Wild”. In: *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2020, pp. 8057–8066. doi: 10.1109/CVPR42600.2020.00808.
- [25] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. “ImageNet classification with deep convolutional neural networks”. In: *Communications of the ACM* 60 (2012), pp. 84–90. URL: <https://api.semanticscholar.org/CorpusID:195908774>.
- [26] Li Xu et al. “Image smoothing via L0 gradient minimization”. In: *ACM Trans. Graph.* 30.6 (Dec. 2011), pp. 1–12. ISSN: 0730-0301. doi: 10.1145/2070781.2024208. URL: <https://doi.org/10.1145/2070781.2024208>.
- [27] Kaiming He et al. “Deep Residual Learning for Image Recognition”. In: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2015), pp. 770–778. URL: <https://api.semanticscholar.org/CorpusID:206594692>.
- [28] Yan Ju et al. “Fusing Global and Local Features for Generalized AI-Synthesized Image Detection”. In: *2022 IEEE International Conference on Image Processing (ICIP)*. 2022, pp. 3465–3469. doi: 10.1109/ICIP46576.2022.9897820.
- [29] Ashish Vaswani et al. “Attention is All you Need”. In: *Neural Information Processing Systems*. 2017. URL: <https://api.semanticscholar.org/CorpusID:13756489>.
- [30] Hanqing Zhao et al. “Multi-Attentional Deepfake Detection”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2021, pp. 2185–2194.

- [31] Nan Zhong et al. “Rich and poor texture contrast: A simple yet effective approach for ai-generated image detection”. In: *arXiv preprint arXiv:2311.12397* (2023).
- [32] S P Arunachalam, S. M. Khairnar, and B. S. Desale. “The Fast Fourier Transform Algorithm and Its Application in Digital Image Processing”. In: *Mathematical theory and modeling* 3 (2013), pp. 267–273. URL: <https://api.semanticscholar.org/CorpusID:60800158>.
- [33] N. Ahmed, T. Natarajan, and K.R. Rao. “Discrete Cosine Transform”. In: *IEEE Transactions on Computers* C-23.1 (1974), pp. 90–93. DOI: 10.1109/T-C.1974.223784.
- [34] Ricard Durall et al. “Unmasking deepfakes with simple features”. In: *arXiv preprint arXiv:1911.00686* (2019).
- [35] Joel Cameron Frank et al. “Leveraging Frequency Analysis for Deep Fake Image Recognition”. In: *ArXiv* abs/2003.08685 (2020). URL: <https://api.semanticscholar.org/CorpusID:213175447>.
- [36] Tero Karras, Samuli Laine, and Timo Aila. “A Style-Based Generator Architecture for Generative Adversarial Networks”. In: *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2018), pp. 4396–4405. URL: <https://api.semanticscholar.org/CorpusID:54482423>.
- [37] Fisher Yu et al. “LSUN: Construction of a Large-scale Image Dataset using Deep Learning with Humans in the Loop”. In: *ArXiv* abs/1506.03365 (2015). URL: <https://api.semanticscholar.org/CorpusID:8317437>.
- [38] Yuyang Qian et al. “Thinking in Frequency: Face Forgery Detection by Mining Frequency-aware Clues”. In: *ArXiv* abs/2007.09355 (2020). URL: <https://api.semanticscholar.org/CorpusID:220647499>.
- [39] National Instruments. *Spectral Leakage*. Accessed: 2024-08-27. n.d. URL: <https://www.ni.com/docs/en-US/bundle/ni-scope/page/spectral-leakage.html>.
- [40] Yonghyun Jeong et al. “BiHPF: Bilateral High-Pass Filters for Robust Deepfake Detection”. In: *2022 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)* (2021), pp. 2878–2887. URL: <https://api.semanticscholar.org/CorpusID:237386257>.
- [41] Yonghyun Jeong et al. “FrePGAN: Robust Deepfake Detection Using Frequency-level Perturbations”. In: *AAAI Conference on Artificial Intelligence*. 2022. URL: <https://api.semanticscholar.org/CorpusID:246634415>.
- [42] Ouxiang Li et al. “Improving Synthetic Image Detection Towards Generalization: An Image Transformation Perspective”. In: *arXiv preprint arXiv:2408.06741* (2024).
- [43] Yuval Alaluf, Or Patashnik, and Daniel Cohen-Or. “Restyle: A residual-based stylegan encoder via iterative refinement”. In: *Proceedings of the IEEE/CVF international conference on computer vision*. 2021, pp. 6711–6720.
- [44] Nan Zhong et al. *PatchCraft: Exploring Texture Patch for Efficient AI-generated Image Detection*. 2024. arXiv: 2311.12397 [cs.CV]. URL: <https://arxiv.org/abs/2311.12397>.
- [45] Alexey Dosovitskiy et al. “An image is worth 16x16 words: Transformers for image recognition at scale”. In: *arXiv preprint arXiv:2010.11929* (2020).

- [46] Sheng-Yu Wang et al. “CNN-Generated Images Are Surprisingly Easy to Spot... for Now”. In: *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2019), pp. 8692–8701. URL: <https://api.semanticscholar.org/CorpusID:209444798>.
- [47] Fisher Yu et al. “Construction of a Large-scale Image Dataset using Deep Learning with Humans in the Loop”. In: 2015.
- [48] Jia Deng et al. “ImageNet: A large-scale hierarchical image database”. In: *2009 IEEE Conference on Computer Vision and Pattern Recognition*. 2009, pp. 248–255. DOI: [10.1109/CVPR.2009.5206848](https://doi.org/10.1109/CVPR.2009.5206848).
- [49] Tero Karras et al. “Progressive Growing of GANs for Improved Quality, Stability, and Variation”. In: *ArXiv* abs/1710.10196 (2017). URL: <https://api.semanticscholar.org/CorpusID:3568073>.
- [50] Jun-Yan Zhu et al. “Unpaired Image-to-Image Translation Using Cycle-Consistent Adversarial Networks”. In: *2017 IEEE International Conference on Computer Vision (ICCV)*. 2017, pp. 2242–2251. DOI: [10.1109/ICCV.2017.244](https://doi.org/10.1109/ICCV.2017.244).
- [51] Taesung Park et al. “Semantic Image Synthesis With Spatially-Adaptive Normalization”. In: *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2019), pp. 2332–2341. URL: <https://api.semanticscholar.org/CorpusID:81981856>.
- [52] Chuangchuang Tan et al. “Rethinking the Up-Sampling Operations in CNN-based Generative Network for Generalizable Deepfake Detection”. In: *ArXiv* abs/2312.10461 (2023). Source code available at: <https://github.com/chuangchuangtan/NPR-DeepfakeDetection>. URL: <https://api.semanticscholar.org/CorpusID:266348433>.
- [53] Tero Karras et al. “Analyzing and Improving the Image Quality of StyleGAN”. In: *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2019), pp. 8107–8116. URL: <https://api.semanticscholar.org/CorpusID:209202273>.
- [54] Andrew Brock, Jeff Donahue, and Karen Simonyan. “Large Scale Training for High Fidelity Natural Image Synthesis”. In: *ArXiv* abs/1809.11096 (2018). URL: <https://api.semanticscholar.org/CorpusID:52889459>.
- [55] Marius Cordts et al. “The Cityscapes Dataset for Semantic Urban Scene Understanding”. In: *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2016.
- [56] Radim Tyleček and Radim Šára. “Spatial Pattern Templates for Recognition of Objects with Regular Structure”. In: *Proc. GCPR*. Saarbrücken, Germany, 2013.
- [57] Aron Yu and Kristen Grauman. “Fine-Grained Visual Comparisons with Local Learning”. In: *2014 IEEE Conference on Computer Vision and Pattern Recognition*. 2014, pp. 192–199. DOI: [10.1109/CVPR.2014.32](https://doi.org/10.1109/CVPR.2014.32).
- [58] Yunjey Choi et al. “StarGAN: Unified Generative Adversarial Networks for Multi-domain Image-to-Image Translation”. In: *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2018, pp. 8789–8797. DOI: [10.1109/CVPR.2018.00916](https://doi.org/10.1109/CVPR.2018.00916).
- [59] Ziwei Liu et al. “Deep Learning Face Attributes in the Wild”. In: *Proceedings of International Conference on Computer Vision (ICCV)*. Dec. 2015.

- [60] Tsung-Yi Lin et al. “Microsoft coco: Common objects in context”. In: *Computer vision–ECCV 2014: 13th European conference, zurich, Switzerland, September 6–12, 2014, proceedings, part v 13*. Springer. 2014, pp. 740–755.
- [61] Qifeng Chen and Vladlen Koltun. “Photographic Image Synthesis with Cascaded Refinement Networks”. In: *2017 IEEE International Conference on Computer Vision (ICCV)* (2017), pp. 1520–1529. URL: <https://api.semanticscholar.org/CorpusID:8191987>.
- [62] Hrishav Bakul Barua et al. “GTA-HDR: A large-scale synthetic dataset for HDR image reconstruction”. In: *2025 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*. IEEE. 2025, pp. 7876–7886.
- [63] Ke Li, Tianhao Zhang, and Jitendra Malik. “Diverse Image Synthesis From Semantic Layouts via Conditional IMLE”. In: *2019 IEEE/CVF International Conference on Computer Vision (ICCV)* (2018), pp. 4219–4228. URL: <https://api.semanticscholar.org/CorpusID:53997451>.
- [64] Chen Chen et al. “Learning to See in the Dark”. In: June 2018, pp. 3291–3300. DOI: [10.1109/CVPR.2018.00347](https://doi.org/10.1109/CVPR.2018.00347).
- [65] Tao Dai et al. “Second-Order Attention Network for Single Image Super-Resolution”. In: *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2019, pp. 11057–11066. DOI: [10.1109/CVPR.2019.01132](https://doi.org/10.1109/CVPR.2019.01132).
- [66] Eirikur Agustsson and Radu Timofte. “NTIRE 2017 Challenge on Single Image Super-Resolution: Dataset and Study”. In: *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*. July 2017.
- [67] Tao Dai et al. “Second-Order Attention Network for Single Image Super-Resolution”. In: *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2019, pp. 11057–11066. DOI: [10.1109/CVPR.2019.01132](https://doi.org/10.1109/CVPR.2019.01132).
- [68] Andreas Rössler et al. *FaceForensics++: Learning to Detect Manipulated Facial Images*. 2019. arXiv: 1901.08971 [cs.CV]. URL: <https://arxiv.org/abs/1901.08971>.
- [69] Zhendong Wang et al. “DIRE for Diffusion-Generated Image Detection”. In: *2023 IEEE/CVF International Conference on Computer Vision (ICCV)* (2023), pp. 22388–22398. URL: <https://api.semanticscholar.org/CorpusID:257557819>.
- [70] Prafulla Dhariwal and Alexander Nichol. “Diffusion models beat gans on image synthesis”. In: *Advances in neural information processing systems* 34 (2021), pp. 8780–8794.
- [71] Jonathan Ho, Ajay Jain, and Pieter Abbeel. “Denoising diffusion probabilistic models”. In: *Advances in neural information processing systems* 33 (2020), pp. 6840–6851.
- [72] Alex Nichol and Prafulla Dhariwal. “Improved Denoising Diffusion Probabilistic Models”. In: *ArXiv* abs/2102.09672 (2021).
- [73] Luping Liu et al. “Pseudo Numerical Methods for Diffusion Models on Manifolds”. In: *ArXiv* abs/2202.09778 (2022).
- [74] Robin Rombach et al. “High-Resolution Image Synthesis with Latent Diffusion Models”. In: *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2021), pp. 10674–10685. URL: <https://api.semanticscholar.org/CorpusID:245335280>.

- [75] Shuyang Gu et al. “Vector Quantized Diffusion Model for Text-to-Image Synthesis”. In: *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2021), pp. 10686–10696. URL: <https://api.semanticscholar.org/CorpusID:244714856>.
- [76] Utkarsh Ojha, Yuheng Li, and Yong Jae Lee. “Towards Universal Fake Image Detectors that Generalize Across Generative Models”. In: *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2023), pp. 24480–24489. URL: <https://api.semanticscholar.org/CorpusID:257038440>.
- [77] Christoph Schuhmann et al. “LAION-400M: Open Dataset of CLIP-Filtered 400 Million Image-Text Pairs”. In: *CoRR* abs/2111.02114 (2021).
- [78] Alex Nichol et al. “Glide: Towards photorealistic image generation and editing with text-guided diffusion models”. In: *arXiv preprint arXiv:2112.10741* (2021).
- [79] Boris Dayma et al. *DALL·E Mini*. July 2021. DOI: 10.5281/zenodo.5146400. URL: <https://github.com/borisdayma/dalle-mini>.
- [80] Chuangchuang Tan et al. “Learning on Gradients: Generalized Artifacts Representation for GAN-Generated Images Detection”. In: *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2023), pp. 12105–12114. URL: <https://api.semanticscholar.org/CorpusID:259226993>.
- [81] Ricard Durall, Margret Keuper, and Janis Keuper. “Watch Your Up-Convolution: CNN Based Generative Deep Neural Networks Are Failing to Reproduce Spectral Distributions”. In: *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2020), pp. 7887–7896. URL: <https://api.semanticscholar.org/CorpusID:211988680>.
- [82] Lucy Chai et al. “What makes fake images detectable? Understanding properties that generalize”. In: *European Conference on Computer Vision*. 2020. URL: <https://api.semanticscholar.org/CorpusID:221266121>.
- [83] Kaede Shiohara and T. Yamasaki. “Detecting Deepfakes with Self-Blended Images”. In: *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2022), pp. 18699–18708. URL: <https://api.semanticscholar.org/CorpusID:248227916>.
- [84] Sara Mandelli et al. “Detecting Gan-Generated Images by Orthogonal Training of Multiple CNNs”. In: *2022 IEEE International Conference on Image Processing (ICIP)* (2022), pp. 3091–3095. URL: <https://api.semanticscholar.org/CorpusID:247244841>.
- [85] Hoai-Danh Vo and Trung-Nghia Le. “Minimalist Preprocessing Approach for Image Synthesis Detection”. In: *Information and Communication Technology*. Ed. by Wray Buntine et al. Singapore: Springer Nature Singapore, 2025, pp. 88–99. ISBN: 978-981-96-4282-3. DOI: 10.1007/978-981-96-4282-3_8.
- [86] Christos Koutlis and Symeon Papadopoulos. “Leveraging Representations from Intermediate Encoder-blocks for Synthetic Image Detection”. In: *arXiv preprint arXiv:2402.19091* (2024).
- [87] Alec Radford et al. “Learning Transferable Visual Models From Natural Language Supervision”. In: *CoRR* abs/2103.00020 (2021).

PHỤ LỤC

Minimalist Preprocessing Approach for Image Synthesis Detection

Hoai-Danh Vo^{1,2}, Trung-Nghia Le^{*1,2}

¹ University of Science, VNU-HCM, Ho Chi Minh City, Vietnam

² Vietnam National University, Ho Chi Minh City, Vietnam

22c15025@student.hcmus.edu.vn, ltngia@fit.hcmus.edu.vn

Abstract. Generative models have significantly advanced image generation, resulting in synthesized images that are increasingly indistinguishable from authentic ones. However, the creation of fake images with malicious intent is a growing concern. Low-configured smart devices have become highly popular, making it easier for deceptive images to reach users. Consequently, the demand for effective detection methods is increasingly urgent. In this paper, we introduce a simple yet efficient method that captures pixel fluctuations between neighboring pixels by calculating the gradient, which highlights variations in grayscale intensity. This approach functions as a high-pass filter, emphasizing key features for accurate image distinction while minimizing color influence. Our experiments on multiple datasets demonstrate that our method achieves accuracy levels comparable to state-of-the-art techniques while requiring minimal computational resources. Therefore, it is suitable for deployment on low-end devices such as smartphones. The code is available at <https://github.com/vohoadanh/adof>.

Keywords: Image synthesis detection · Lightweight model · Low-level computation

1 Introduction

In recent years, significant advancements in image generation have been achieved, particularly with Generative Adversarial Networks (GANs) [11] and Diffusion models [12, 14]. These approaches produce high-quality images that closely resemble real-world visuals [31] and have garnered attention in academic and societal circles. Generative models have found applications in various fields, including virtual try-ons and personalized fashion recommendations in the fashion industry [25], as well as in image editing [4, 39] and interior design [6].

Despite the valuable applications of image generation technology, significant drawbacks exist. According to a survey conducted by Bauer and Bind-schaedlerr [2], generative models can create fake information, particularly deep-fakes, which depict fabricated scenarios involving famous individuals. In response to these dangers, several US states [3, 20] have outlawed the malicious use of

* Corresponding author.

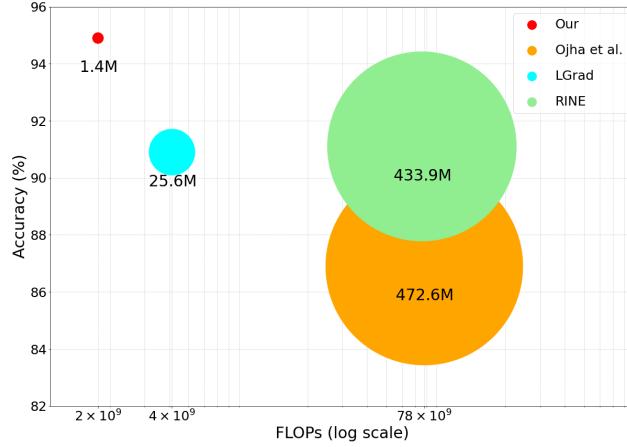


Fig. 1. Comparison of different synthetic image detection methods on the Ojha dataset [28]. Our proposed method is simple yet efficient, significantly reduces FLOPs, total parameters, while achieving comparable accuracy state-of-the-art methods.

deepfake technology, especially for harmful content like revenge and celebrity pornography. To address the threats posed by synthetic images on digital communication platforms and social media, it is essential to develop effective countermeasures for verifying image authenticity directly on mobile devices. Given the ubiquity and portability of these devices, real-time detection of generated images is crucial for preventing misinformation and preserving the integrity of visual content. However, the constrained computational capacity of mobile devices presents a significant challenge. This paper introduces a simple yet efficient solution for synthesized image detection, specifically the Adjacency Difference Orientation Filter (ADOF) for data preprocessing; this filter allows us to compute the gradient in both the x and y directions. The direction of the gradient reflects the behavior of grayscale variation among neighboring pixels, assisting in distinguishing between real and generated images. Focusing on extracting useful low-level features, our approach ensures generalization while utilizing a lightweight CNN architecture for detecting generated images, without demanding extensive computational resources. This strategy effectively reduces irrelevant information, enabling the model to concentrate on fine-grained variations, ultimately leading to improved performance and generalization. In contrast to existing methods [21, 28, 35] that require large deep learning architectures, such as CLIP [30], ViT [7], Resnet50 [13], and significant computational resources, our approach demands fewer resources while still ensuring generalization and achieving comparable accuracy. Fig. 1 presents a comparative overview of results, highlighting the advantages of this strategy.

Experiments on well-known datasets [28, 36, 37] demonstrate the effectiveness of our method, achieving impressive accuracy of 94.9% on the Ojha dataset [28]

and 98.3 % on the DiffusionForensis [37]. Additionally, there is a reduction in computational load by 97.8% compared to RINE [21] and 57.8% compared to LGrad [35]. These results underscore the advantages of our approach, as illustrated in Fig. 1. The code for reproducing our results is publicly available at <https://github.com/vohoaidanh/adof>. Our contributions are as follow:

- We introduce a simple yet efficient approach for detecting synthetic images, and our approach is more generalized than existing methods.
- We present a filter-based method that computes pixel intensity gradients to capture pixel fluctuations and reduce color influence, leading to improved model performance with faster inference speed and lower complexity.
- Our proposed method reduces the number of parameters and FLOPs while maintaining accuracy compared to state-of-the-art methods.

2 Related Work

Various methods have been developed to address the challenge of distinguishing synthetic images from real ones, utilizing both traditional machine learning techniques and modern deep learning approaches. Durall *et al.* [8] applied a Fourier Transform [1] to grayscale images and used azimuthal averaging to convert the 2D frequency data into a 1D feature vector, retaining essential information for classification. They then employed either Support Vector Machines or K-means clustering to detect GAN-generated images. Alternatively, methods like RINE [21] and Ojha *et al.* [28], along with similar approaches, leverage pre-trained deep learning networks such as CLIP to enhance performance. This integration contributing to consistently high success rates in detecting synthesized images through the integration of these networks into their frameworks. Notably, the FatFormer [23] method focuses on the contrastive objectives between adapted image features and text prompt embeddings, providing valuable information that enables the deep learning models to learn more robust and discriminative representations, ultimately improving their ability to accurately classify real and generated images.

Frequency domain-based methods involve transforming images from the spatial domain to the frequency domain using transformations such as Fast Fourier Transform or Discrete Cosine Transform (DCT). By focusing on frequency characteristics, these methods effectively capture artifacts that might not be evident in the spatial domain. This allows classifiers to distinguish between real and fake images by analyzing the unique patterns that emerge in the spectral domain. Frank *et al.* [10] utilized the DCT to analyze images in the frequency domain, revealing unique spectral differences between real images and those produced by GAN models. Qian *et al.* introduced $F^3\text{-Net}$ [29] to decompose the spectrum into various bands, enabling the analysis of these components to identify unusual distributions. This method effectively detects subtle artifacts, enhancing the ability to recognize synthetic image manipulations.

Tan *et al.* proposed FreqNet [33], which emphasizes high-frequency details and directs the detector to concentrate on these features across spatial and chan-

nel dimensions, rather than utilizing the full spectrum of frequency bands as is common in many other approaches. BiHPF [16], a method by Jeong *et al.*, amplifies frequency-level artifacts commonly found in images generated by generative models to tackle the challenge of identifying images from previously unobserved models. Jeong *et al.* [15] generated perturbation maps added to training images to prevent overfitting to frequency-specific features, reducing high-frequency noise and enhancing classifier generalization.

Spatial-based methods analyze images directly on pixel values, as seen in models like CNNDetection [36] and Gram-Net [24]. A key issue, however, is that raw images often contain excessive, irrelevant information, such as semantic content, which Nang *et al.* [40] identified as detrimental to image classification effectiveness. This extraneous information, unnecessary for distinguishing real from fake images, can disrupt the model’s learning process and reduce its effectiveness. Wang *et al.* [36] developed a comprehensive detector to distinguish real images from CNN-generated ones [22]. Using a dataset of images from 11 CNN-based generators, they showed that, with effective pre-/post-processing and data augmentation, a classifier trained on ProGAN [18] generalizes well to other models, including StyleGAN2 [19]. By dividing images into small patches categorized as either rich texture or flat, PatchCraft [40] exploits the inter-pixel correlation contrast between these regions. This approach breaks the semantic coherence present in traditional methods, addressing a key limitation and enhancing the model’s ability to generalize more effectively. Tan *et al.* introduced the concept of Neighboring Pixel Relationships (NPR) [34] to capture and characterize generalized structural artifacts that arise from up-sampling operations, which are commonly used in image generation models to enhance image quality. This method shows a significant improvement over other techniques within the same approach.

Frequency-based approaches achieve faster convergence with smaller models but may lose essential spatial information needed to distinguish real from generated images. In contrast, spatial-based methods often require larger models and may struggle with new domain data. Our method leverages the strengths of both approaches by focusing on pixel perturbations between neighboring pixels, effectively discarding much of the pixel color information. Additionally, our filter functions as a high-pass filter, removing low-frequency components to emphasize the relevant features.

3 Proposed Method

3.1 Overview

Generative models, such as GAN [11] and Diffusion [14], currently use CNN layers for image synthesis, meaning that neighboring pixel regions are correlated to a certain extent. We hypothesize that synthetic images exhibits a stronger correlation between adjacent pixels compared to real images. Furthermore, due to the design of neural networks, noise in synthetic images tends to be averaged out, whereas in real images, noise typically remains more prominent. To

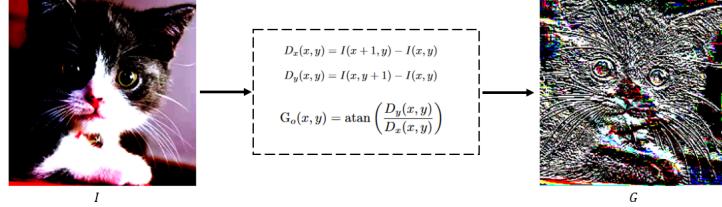


Fig. 2. The left image represents the original image, the middle shows the gradient calculation applied, and the right image illustrates the resulting gradient map.

investigate and evaluate the impact of noise on adjacent pixels, we designed a simple yet effective filter to capture these variations. This filter aids the CNN in learning the differences in noise distribution between real and synthetic images. The overall architecture of ADOF is illustrated in Fig. 2. This approach captures noise information by calculating the differences between adjacent pixels and incorporating these differences into the gradient to account for variations in both the x and y directions.

3.2 Adjacency Difference Orientation Filter (ADOF)

Finite Difference. The finite difference technique [27, 38] is a mathematical approach for estimating intensity variations across neighboring points in a grid or matrix. In image processing, it calculates gradients by measuring differences in pixel values, thereby detecting changes in intensity in both horizontal and vertical directions. This technique aids in edge detection and texture analysis by highlighting contrasts between adjacent pixels.

The general formula for calculating the gradient in a given direction u is $\text{Gradient}_u = I(x + \Delta x, y + \Delta y) - I(x, y)$, where Δx and Δy define the direction of the difference.

Filter Construction. We have applied the *finite difference* to compute the gradients of intensity in our images. This helps in identifying intensity variations at each pixel and supports the detection of important geometric features in our approach. The formula that computes the difference between adjacent pixels along the x and y -direction is given by:

$$D_x(x, y) = I(x + 1, y) - I(x, y), \quad (1)$$

$$D_y(x, y) = I(x, y + 1) - I(x, y), \quad (2)$$

where I represents the image in which the difference is being calculated, $D_x(x, y)$ represents the difference between the pixel value at $(x + 1, y)$ and the pixel value at (x, y) . This filter captures variations in pixel intensity along the horizontal direction. Similarly, $D_y(x, y)$ captures variations in pixel intensity along the

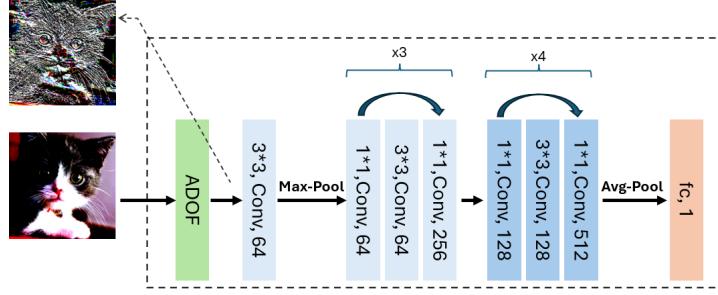


Fig. 3. Architecture of our lightweight model.

vertical direction. To determine the gradient magnitude and orientation, these values are computed from D_x and D_y .

$$G_m(x, y) = \sqrt{D_x(x, y)^2 + D_y(x, y)^2}, \quad (3)$$

$$G_o(x, y) = \arctan\left(\frac{D_y(x, y)}{D_x(x, y)}\right), \quad (4)$$

where $D_x(x, y)$ and $D_y(x, y)$ are as previously defined. The gradient orientation G_o , which represents the overall angle of gray-level changes at a pixel and indicates the direction of these combined intensity variations, is referred to as the **Adjacency Difference Orientation Filter (ADOF)** in this paper. Meanwhile, the gradient magnitude G_m quantifies the strength of intensity changes at that pixel. The result of this computational process is illustrated in Figure 2.

3.3 Lightweight Model Architecture

To evaluate the effectiveness of our filter ADOF on images, we use basic CNN architectures. Specifically, this work employs a modified ResNet50 [13] model with `layer3` and `layer4` removed. To capture information from 8-connected neighboring pixels more effectively, the kernel size of the `conv1` layer was adjusted from 7 to 3. The architecture is depicted in Fig. 3.

4 Experiments

4.1 Implementation Details

In practice, we are more concerned with the flat regions of an image rather than the edge areas where there is a significant variation in gray levels between the x and y directions. This is because, in regions with large changes in gray levels in one direction compared to the other, the gradient angles are close to $\pm\frac{\pi}{2}$. Although these angles both indicate edge regions in the image, the gradient

angles at edges typically take values of $\pm\frac{\pi}{2}$, which are numerically distant from each other despite conveying similar edge information. To exclude these areas, we set the gradient values approaching $\pm\frac{\pi}{2}$ to 0, the experimental process has demonstrated that this approach leads to higher accuracy for the model.

All experiments are conducted on a computing system using a NVIDIA RTX A4000 GPU with 16 GB of memory and an AMD Ryzen 5 5600X 6-Core CPU. We trained our model using parameters that are closely aligned with those used in common methods [34–36] to ensure a fair comparison and demonstrate the effectiveness of our method independent of specific hyperparameters. Furthermore, we utilized the source code provided by NPR [34] to streamline the training process and maintain consistency. The model was trained using the Adam optimizer with a learning rate of 2×10^{-4} and a batch size of 32. To accelerate the training process, we adjusted the learning rate every 5 epochs instead of every 10 epochs and utilized 4 out of the 20 classes (*car, cat, chair, horse*) for training, similar to the protocol used in existing works [15, 16, 34, 36].

4.2 Dataset

Training set. To facilitate comparison between methods, we used the same ForenSynths dataset with existing methods [17, 28, 34–36]. This dataset consists of 20 object classes selected from the LSUN dataset. Each class contains 18,000 real-world images, with corresponding generative images generated using the ProGAN [18] model. To verify the generalization of methods, all compared method was trained on a subset of the ForenSynths [36] dataset consisting of 4 classes: car, cat, chair, horse.

Evaluation set. To investigate the generalization of methods, our evaluation was conducted using the Self-Synthesis 9 GANs [34], which contains 36,000 images sourced from LSUN, ImageNet, CelebA, CelebA-HQ, COCO, and FaceForensics++, generated using models like AttGAN, BEGAN, and CramerGAN. The second dataset, DiffusionForensics [37], comprises 40,000 images from LSUN and ImageNet, utilizing models such as ADM, DDPM, and IDDP. Lastly, the Ojha Test Set [28] includes 16,000 images from LAION and ImageNet, generated with ADM, Glide, DALL-E-Mini, and LDM.

4.3 Comparison with State-of-the-Art Methods

We conduct a performance comparison of our method with 10 State-of-the-Art methods, including CNNDetection [36], Frank [10], Durall [9], Patchfor [5], F3Net [29], SelfBland [32], GANDetection [26], LGrad [35], Ojha [28], NPR [34]. The experimental results in Tables 1, 2, and 3 demonstrate that our method exceeds existing approaches. On the 9-GAN dataset, ADOF delivers the highest accuracy at 94.2%, surpassing Ojha [28] with a mere 77.6% 1, and NPR [34] at 93.2% (see Table 1). Notably, our approach achieves a remarkable 98.3% accuracy on the DiffusionForensics [37] dataset, outperforming the NPR method [34],

Table 1. Evaluation results on the Self-Synthesis 9 GANs [34].

Method	AttGAN		BEGAN		CramerGAN		InfoMaxGAN		MMDGAN		RelGAN		S2GAN		SNGAN		STGAN		Mean	
	Acc.	A.P.	Acc.	A.P.	Acc.	A.P.	Acc.	A.P.	Acc.	A.P.	Acc.	A.P.	Acc.	A.P.	Acc.	A.P.	Acc.	A.P.	Acc.	A.P.
CNNDetection [36]	51.1	83.7	50.2	44.9	81.5	97.5	71.1	94.7	72.9	94.4	53.3	82.1	55.2	66.1	62.7	90.4	63.0	92.7	62.3	82.9
Frank [10]	65.0	74.4	39.4	39.9	31.0	36.0	41.1	41.0	38.4	40.5	69.2	96.2	69.7	81.9	48.4	47.9	25.4	34.0	47.5	54.7
Durall [9]	39.9	38.2	48.2	30.9	60.9	67.2	50.1	51.7	59.5	65.5	80.0	88.2	87.3	97.0	54.8	58.9	62.1	72.5	60.3	63.3
Patchfor [5]	68.0	92.9	97.1	100.0	97.8	99.9	93.6	98.2	97.9	100.0	99.6	100.0	66.8	68.1	97.6	99.8	92.7	99.8	90.1	95.4
F3Net	85.2	94.8	87.1	97.5	89.5	99.8	67.1	83.1	73.7	99.6	98.8	100.0	65.4	70.0	51.4	93.6	60.4	99.9	75.4	93.1
SelfBland [32]	63.1	66.1	56.4	59.0	75.1	82.4	79.0	82.5	68.6	74.0	73.6	77.8	53.2	53.9	61.6	65.0	61.2	66.7	65.8	69.7
GANDetection [26]	57.4	75.1	67.9	100.0	67.8	99.7	67.6	92.4	67.7	99.3	60.9	86.2	69.6	83.5	66.7	90.6	69.6	97.2	66.1	91.6
LGrad [35]	68.6	93.8	69.9	89.2	50.3	54.0	71.1	82.0	57.5	67.3	89.1	99.1	78.5	86.0	78.0	87.4	54.8	68.0	68.6	80.8
Ojha [28]	78.5	98.3	72.0	98.0	77.6	99.8	77.6	98.9	77.6	99.7	78.2	98.7	85.2	98.1	77.6	98.7	74.2	97.8	77.6	98.8
NPR [34]	83.0	96.2	99.0	99.8	98.7	99.0	94.5	98.3	98.6	99.0	99.6	100.0	79.0	80.0	88.8	97.4	98.0	100.0	93.2	96.6
ADOF(ours)	99.5	100.0	92.2	100.0	96.0	99.6	94.1	99.1	96.0	99.7	100.0	100.0	77.5	86.7	94.8	99.3	97.8	99.7	94.2	98.2

Table 2. Evaluation results on the test set of DiffusionForensics dataset [37].

Method	Stable Diffusion v1												Stable Diffusion v2												Mean
	ADM	DDPM	IDDPDM	LDM	PNDM	VQ-Diffusion	ADM	DDPM	IDDPDM	LDM	PNDM	VQ-Diffusion	ADM	DDPM	IDDPDM	LDM	PNDM	VQ-Diffusion	ADM	DDPM	IDDPDM	LDM	PNDM	VQ-Diffusion	
Acc.	A.P.	Acc.	A.P.	Acc.	A.P.	Acc.	Acc.	A.P.	Acc.	A.P.	Acc.	Acc.	A.P.	Acc.	A.P.	Acc.	A.P.	Acc.	A.P.	Acc.	A.P.	Acc.	A.P.	Acc.	
CNNDetection [36]	53.9	71.8	62.7	76.6	50.2	82.7	50.4	78.7	50.8	90.3	50.0	71.0	38.0	76.7	52.0	90.3	51.0	79.8	51.0	79.8	51.0	79.8	51.0	79.8	
Frank [10]	58.9	65.9	37.0	27.6	51.4	65.0	51.7	48.5	44.0	38.2	51.7	66.7	32.8	52.3	40.8	37.5	46.0	50.2	58.9	50.2	58.9	50.2	58.9	50.2	
Durall [9]	39.8	42.1	52.9	49.8	55.3	56.7	43.1	39.9	44.5	47.3	38.6	38.3	39.5	56.3	62.1	55.8	47.0	48.3	58.9	48.3	58.9	48.3	58.9	48.3	
Patchfor [5]	77.5	93.9	62.3	97.1	50.1	91.6	99.5	100.0	50.2	99.4	100.0	100.0	90.7	99.8	94.8	100.0	78.1	97.8	85.8	99.0	85.8	99.0	85.8	99.0	
F3Net [29]	80.9	96.9	84.7	99.4	74.7	98.9	90.0	100.0	72.8	99.5	100.0	100.0	73.4	97.2	99.8	100.0	88.6	99.0	88.6	99.0	88.6	99.0	88.6	99.0	
SelfBland [32]	57.0	59.0	61.9	49.6	63.2	66.9	83.3	92.2	48.2	48.2	77.2	82.7	46.2	68.0	71.2	73.9	63.5	67.6	63.5	67.6	63.5	67.6	63.5	67.6	
GANDetection [26]	51.1	53.1	62.3	46.4	50.2	63.0	51.6	48.1	50.6	79.0	51.1	51.2	39.8	65.6	50.1	36.9	50.8	55.4	50.8	55.4	50.8	55.4	50.8	55.4	
LGrad [35]	86.4	97.5	99.9	100.0	66.1	92.8	99.7	100.0	69.5	98.5	96.2	100.0	90.4	99.4	97.1	100.0	88.2	98.5	88.2	98.5	88.2	98.5	88.2	98.5	
Ojha [28]	78.4	92.1	72.9	78.8	75.0	92.8	82.2	97.1	75.3	92.5	83.5	97.7	56.4	90.4	71.5	92.4	74.4	91.7	74.4	91.7	74.4	91.7	74.4	91.7	
NPR [34]	88.6	98.9	99.8	100.0	91.8	99.8	100.0	100.0	91.2	100.0	100.0	100.0	100.0	97.4	99.8	93.8	100.0	95.3	99.8	95.3	99.8	95.3	99.8		
ADOF(ours)	93.5	99.0	99.6	100.0	99.2	100.0	99.9	100.0	97.4	99.9	97.1	99.8	99.8	100.0	99.9	100.0	98.3	99.8	98.2	98.2	98.2	98.2	98.2	98.2	

Table 3. Evaluation results on the diffusion test set of Ojha [28].

Method	DALLE												Glide												Mean
	100	10	Glide	100	27	Glide	50	27	ADM	DDPM	IDDPDM	LDM	PNDM	VQ-Diffusion	100	LDM	200	LDM	200	cfg	ADM	DDPM	IDDPDM	LDM	PNDM
Acc.	A.P.	Acc.	A.P.	Acc.	A.P.	Acc.	A.P.	Acc.	A.P.	Acc.	A.P.	Acc.	A.P.	Acc.	A.P.	Acc.	A.P.	Acc.	A.P.	Acc.	A.P.	Acc.	A.P.	Acc.	
CNNDetection [36]	51.8	61.3	53.3	72.9	53.0	71.3	54.2	76.0	54.9	66.6	51.9	63.7	52.0	64.5	51.6	63.1	52.8	67.4	52.8	67.4	52.8	67.4	52.8	67.4	
Frank [10]	57.0	62.5	53.6	44.3	50.4	40.8	52.0	42.3	53.4	52.5	56.6	51.3	56.4	50.9	56.5	52.1	54.5	49.6	54.5	49.6	54.5	49.6	54.5	49.6	
Durall [9]	55.9	58.0	54.9	52.3	48.9	46.9	51.7	49.9	40.6	42.8	62.0	62.6	61.7	61.7	58.4	58.5	54.3	54.0	54.3	54.0	54.3	54.0	54.3	54.0	
Patchfor [5]	79.8	99.1	87.3	99.7	82.8	99.1	84.9	98.8	74.2	81.4	95.8	99.8	95.6	99.9	94.0	99.8	86.8	97.2	86.8	97.2	86.8	97.2	86.8	97.2	
F3Net [29]	71.6	79.9	88.3	95.4	87.0	94.5	88.5	95.4	69.2	70.8	74.1	84.0	73.4	83.3	80.7	89.1	79.1	86.5	79.1	86.5	79.1	86.5	79.1	86.5	
SelfBland [32]	52.4	51.6	58.8	63.2	59.4	64.1	64.2	68.3	58.3	63.4	53.0	54.0	52.6	51.9	52.6	52.6	53.8	58.9	54.3	58.7	54.3	58.7	54.3	58.7	
GANDetection [26]	67.2	83.0	51.2	52.6	51.1	51.9	51.7	53.5	49.6	49.0	54.7	65.8	54.9	65.9	53.8	58.9	54.3	60.1	54.3	60.1	54.3	60.1	54.3	60.1	
LGrad [35]	88.5	97.3	89.4	94.9	87.4	93.2	90.7	95.1	86.6	100.0	94.8	99.2	94.2	99.1	95.9	99.2	90.9	97.2	90.9	97.2	90.9	97.2	90.9	97.2	
Ojha [28]	89.5	96.8	90.1	97.0	90.7	97.2	91.1	97.4	75.7	85.1	90.5	97.0	90.2	97.1	77.3	88.6	86.9	94.5	86.9	94.5	86.9	94.5	86.9	94.5	
NPR [34]	94.5	99.5	98.2	99.8	97.8	99.7	98.2	99.8	99.8	81.0	99.3	99.9	99.1	99.9	99.0	99.8	95.2	97.4	95.2	97.4	95.2	97.4	95.2	97.4	
RINE [21]	95.0	99.5	90.7	99.2	88.9	99.1	92.6	99.5	76.1	96.6	98.7	99.8	99.3	99.8	88.2	98.7	91.1	99.0	91.1	99.0	91.1	99.0	91.1	99.0	
ADOF(ours)	92.1	98.3	98.6	100.0	98.7	100.0	98.4	99.9	75.9	87.6	98.8	100.0	98.6	99.9	98.5	99.9	94.9	98.2							

which only reaches 95.3% (see Table 2). It also surpasses DIRE [37], which reports 97.9% accuracy on its own dataset, despite our model

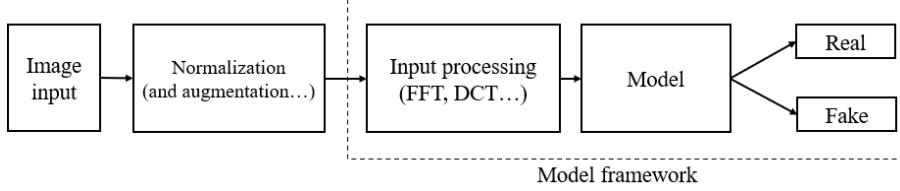


Fig. 4. Standard pipeline for Image Synthesis Methods

Table 4. Resource usage and performance of synthetic image detection methods on the DiffusionForensics [37]. The method marked with [†] indicates it was trained on this dataset.

Method	Parameters	Processing (ms)	Inference Time (ms)	FLOPs	Means (acc/ap)
LGrad [35]	25.56×10^6	11.6	4.81	4.12×10^9	88.2/98.5
DIRE [†] [37]	25.56×10^6	4,502.7	4.81	4.12×10^9	97.9/ 100
Ojha [28]	427.62×10^6	None	29.19	77.83×10^9	74.4/91.7
ADOF(ours)	1.44×10^6	0.40	2.43	1.74 × 10⁹	98.3/99.8

- **Input Processing Time:** We measure the time required for processing images before they are fed into the model, where these processing steps are tailored to the specific method used (See Fig. 4).
- **Inference Time:** We record the time taken for the model to process an image and produce a result.
- **FLOPs (Floating Point Operations Per Second):** We leveraged the `fvcore` library to estimate the FLOPs required by each model during inference, providing valuable insights into their computational demands.

Our method requires substantially fewer parameters and FLOPs while achieving faster inference and the highest mean accuracy (98.3%) compared to existing methods, including the DIRE [37] (see Table 4), which is trained on the same dataset but does not achieve comparable performance. This demonstrates its superior performance in synthetic image detection.

5 Conclusion

In this paper, we proposed a simple yet highly effective filter, namely ADOF, for capturing pixel-level variations. By treating an image as a discrete digital signal, this method eliminates the average components of the signal. These components typically carry semantic information, which is less helpful for distinguishing between real and synthetic images compared to the subtle traces that the proposed filter is designed to detect. Experimental results indicate that our proposed method significantly reduces model complexity while enhancing both accuracy and generalization, even on previously unseen data.

Acknowledgment

This research is funded by Vietnam National University - Ho Chi Minh City (VNU-HCM) under Grant Number C2024-18-25.

References

1. Arunachalam, S., Khairnar, S., Desale, B.: The fast fourier transform algorithm and its application in digital image processing. *New J Chem* **35**(5) (2013)
2. Bauer, L.A., Bindschaedler, V.: Generative models for security: Attacks, defenses, and opportunities. arXiv:2107.10139 (2021)
3. Cara Curtis: California makes deepfakes illegal to curb revenge porn and doctored political videos (2019), <https://bit.ly/4f40oaX>, accessed: 2024-09-24
4. Casteleiro-Pitrez, J.: Generative artificial intelligence image tools among future designers: A usability, user experience, and emotional analysis. *Digital* **4**(2), 316–332 (2024)
5. Chai, L., Bau, D., Lim, S.N., Isola, P.: What makes fake images detectable? understanding properties that generalize. In: Eur. Conf. on Computer Vision. (2020)
6. Chen, Z., Wang, X.: Application of ai technology in interior design **179** (2020)
7. Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., Houlsby, N.: An image is worth 16x16 words: Transformers for image recognition at scale. CoRR [abs/2010.11929](https://arxiv.org/abs/2010.11929) (2020)
8. Durall, R., Keuper, M., Pfreundt, F.J., Keuper, J.: Unmasking deepfakes with simple features. ArXiv [abs/1911.00686](https://arxiv.org/abs/1911.00686) (2019)
9. Durall, R., Keuper, M., Keuper, J.: Watch your up-convolution: Cnn based generative deep neural networks are failing to reproduce spectral distributions pp. 7890–7899 (2020)
10. Frank, J.C., Eisenhofer, T., Schönher, L., Fischer, A., Kolossa, D., Holz, T.: Leveraging frequency analysis for deep fake image recognition. ArXiv (2020)
11. Goodfellow, I.J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A.C., Bengio, Y.: Generative adversarial networks. *Communications of the ACM* **63**, 139–144 (2014)
12. Gu, S., Chen, D., Bao, J., Wen, F., Zhang, B., Chen, D., Yuan, L., Guo, B.: Vector quantized diffusion model for text-to-image synthesis. CVPR pp. 10696–10706 (2022)
13. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition pp. 770–778 (2016)
14. Ho, J., Jain, A., Abbeel, P.: Denoising diffusion probabilistic models. ArXiv [abs/2006.11239](https://arxiv.org/abs/2006.11239) (2020)
15. Jeong, Y., Kim, D., Ro, Y., Choi, J.: Freqgan: Robust deepfake detection using frequency-level perturbations. In: AAAI Conference on Artificial Intelligence (2022)
16. Jeong, Y., Kim, D., Min, S., Joe, S., Gwon, Y., Choi, J.: Bihpf: Bilateral high-pass filters for robust deepfake detection. WACV pp. 48–57 (2022)
17. Ju, Y., Jia, S., Ke, L., Xue, H., Nagano, K., Lyu, S.: Fusing global and local features for generalized ai-synthesized image detection. In: ICIP. pp. 3465–3469 (2022)
18. Karras, T., Aila, T., Laine, S., Lehtinen, J.: Progressive growing of gans for improved quality, stability, and variation. ArXiv [abs/1710.10196](https://arxiv.org/abs/1710.10196) (2017)

19. Karras, T., Laine, S., Aittala, M., Hellsten, J., Lehtinen, J., Aila, T.: Analyzing and improving the image quality of stylegan pp. 8110–8119 (2020)
20. Korosec, K.: Deepfake revenge porn is now illegal in virginia (2019), <https://techcrunch.com/2019/07/01/deepfake-revenge-porn-is-now-illegal-in-virginia/>, accessed: 24 Sep
21. Koutlis, C., Papadopoulos, S.: Leveraging representations from intermediate encoder-blocks for synthetic image detection. arXiv:2402.19091 (2024)
22. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. Communications of the ACM **60**, 84 – 90 (2012)
23. Liu, H., Tan, Z., Tan, C., Wei, Y., Wang, J., Zhao, Y.: Forgery-aware adaptive transformer for generalizable synthetic image detection. In: CVPR. pp. 10770–10780 (2024)
24. Liu, Z., Qi, X., Torr, P.H.: Global texture enhancement for fake face detection in the wild. In: CVPR. pp. 8060–8069 (2020)
25. Lomov, I., Makarov, I.: Generative models for fashion industry using deep neural networks. In: ICCAIS. pp. 1–6. IEEE (2019)
26. Mandelli, S., Bonettini, N., Bestagini, P., Tubaro, S.: Detecting gan-generated images by orthogonal training of multiple cnns pp. 3091–3095 (2022)
27. Mickens, R.E.: Difference equations: theory, applications and advanced topics. CRC Press (2015)
28. Ojha, U., Li, Y., Lee, Y.J.: Towards universal fake image detectors that generalize across generative models pp. 24480–24489 (2023)
29. Qian, Y., Yin, G., Sheng, L., Chen, Z., Shao, J.: Thinking in frequency: Face forgery detection by mining frequency-aware clues. ArXiv **abs/2007.09355** (2020)
30. Radford, A., Kim, J.W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., et al.: Learning transferable visual models from natural language supervision pp. 8748–8763 (2021)
31. for Schools, B.: Spotting ai: Knowing how to recognise real vs ai images. <https://elearn.eb.com/real-vs-ai-images/> (2024), accessed: 2024-08-21
32. Shiohara, K., Yamasaki, T.: Detecting deepfakes with self-blended images pp. 18720–18729 (2022)
33. Tan, C., Zhao, Y., Wei, S., Gu, G., Liu, P., Wei, Y.: Frequency-aware deepfake detection: Improving generalizability through frequency space domain learning **38**(5), 5052–5060 (2024)
34. Tan, C., Zhao, Y., Wei, S., Gu, G., Liu, P., Wei, Y.: Rethinking the up-sampling operations in cnn-based generative network for generalizable deepfake detection pp. 28130–28139 (2024)
35. Tan, C., Zhao, Y., Wei, S., Gu, G., Wei, Y.: Learning on gradients: Generalized artifacts representation for gan-generated images detection pp. 12105–12114 (2023)
36. Wang, S.Y., Wang, O., Zhang, R., Owens, A., Efros, A.A.: Cnn-generated images are surprisingly easy to spot... for now pp. 8695–8704 (2020)
37. Wang, Z., Bao, J., Zhou, W., Wang, W., Hu, H., Chen, H., Li, H.: Dire for diffusion-generated image detection pp. 22445–22455 (2023)
38. Wikipedia Contributors: Finite difference (2024), https://en.wikipedia.org/wiki/Finite_difference, accessed: 2024-08-21
39. Wootaek Shin, P., Ahn, J.J., Yin, W., Sampson, J., Narayanan, V.: Can prompt modifiers control bias? a comparative analysis of text-to-image generative models. arXiv e-prints (2024)
40. Zhong, N., Xu, Y., Li, S., Qian, Z., Zhang, X.: Patchcraft: Exploring texture patch for efficient ai-generated image detection (2024)