

TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN
KHOA CÔNG NGHỆ THÔNG TIN



Nhóm 2 :

21120209 - Phạm Bách Chiến

21120369 - Nguyễn Minh Vũ

21120278 - Phùng Đoàn Khôi

**IMAGE FORENSICS VÀ BÀI TOÁN ỨNG
DỤNG PHÁT HIỆN ẢNH GIẢ MẠO
COPY-MOVE**

CSC16005 – Xử lí ảnh và video số

Tp. Hồ Chí Minh, tháng 1/2023

Mục lục

Mục lục	i
Tóm tắt	iv
1 Giới thiệu	1
1.1 Image forensics	1
1.1.1 Xác định nguồn gốc ảnh	2
1.1.2 Phát hiện ảnh tổng hợp	3
1.1.3 Phát hiện ảnh giả mạo	3
1.1.4 Phát biểu bài toán	5
1.1.5 Framework	5
2 Các công trình liên quan	7
2.1 Bộ dữ liệu	7
2.2 Hướng tiếp cận block-based	8
2.2.1 Nguyên lý	8
2.2.2 Detection of Image Region Duplication Forgery Using Model with Circle Block (2009)	8
2.3 Hướng tiếp cận keypoint-based	10
2.3.1 Nguyên lý	10
2.3.2 Geometric transformation invariant block based copy-move forgery detection using fast and efficient hybrid local features (2019)	11
2.4 Hướng tiếp cận deep learning	13
2.4.1 Nguyên lý	13
2.4.2 A novel deep learning framework for copy-move forgery detection in images (2020)	13
2.4.3 Một số công trình khác	14
2.5 So sánh	14

3 Phương pháp truyền thống	17
3.1 Giới thiệu	17
3.2 Rút trích đặc trưng	17
3.3 Matching	18
3.4 Hậu xử lý	18
4 Phương pháp Deep Learning	20
4.1 Tổng quát	20
4.2 Nhánh phát hiện chỉnh sửa	21
4.3 Nhánh phát hiện tương đồng	22
4.3.1 Rút trích đặc trưng	22
4.3.2 Matching	22
4.3.3 Hậu xử lý	24
4.4 Fusion	24
4.5 Hàm loss	24
5 Cài đặt và thử nghiệm	25
5.1 Phương pháp truyền thống	25
5.1.1 Dataset và Metrics	25
5.1.2 Cài đặt thuật toán	25
5.1.3 Kết quả thuật toán	26
5.2 Phương pháp Deep Learning	28
5.2.1 Dataset và Metrics	28
5.2.2 Cài đặt thuật toán	28
6 Kết luận và hướng phát triển	30
Tài liệu tham khảo	31

Danh sách hình

1.1	Một tấm hình được tạo bởi AI	3
1.2	Minh họa cho kĩ thuật splicing	4
1.3	Các bước để tạo nên một tấm ảnh copy-move	4
1.4	Framework cơ bản của bài toán phát hiện ảnh giả mạo bằng kĩ thuật copy-move	5
2.1	Một cặp dữ liệu trong bộ MICC-F220	7
2.2	Minh họa các bước cơ bản của phương pháp block-based .	8
2.3	Khối tròn trong phương pháp block-based	9
2.4	Minh họa các bước cơ bản của phương pháp keypoint-based	10
2.5	Tóm tắt các bước thực hiện của công trình [15]	11
2.6	Framework của [6]	13
3.1	Giải thuật của thuật toán DBSCAN	19
4.1	Framework của mô hình BusterNet	20
4.2	Mask deconvolution network	21
4.3	Module BN-Inception	22
5.1	Tiền xử lí ảnh	26
5.2	Feature Extraction: SIFT Keypoints	26
5.3	Matching Keypoints	27

Danh sách bảng

2.1	So sánh các bước của 3 công trình	15
2.2	So sánh ưu nhược điểm của 3 hướng tiếp cận	15
2.3	Kết quả thực nghiệm của 3 công trình trên tập dữ liệu MICC-F220	16

Giới thiệu

Trong thời đại số hóa ngày nay, việc chỉnh sửa hình ảnh đã trở nên dễ dàng hơn bao giờ hết nhờ vào các công cụ chỉnh sửa hình ảnh mạnh mẽ. Một trong những kỹ thuật giả mạo phổ biến nhất là copy-move (sao chép-di chuyển), nơi một phần của hình ảnh được sao chép và dán vào một vị trí khác trong cùng một hình ảnh. Điều này có thể tạo ra những hình ảnh giả mạo khó phát hiện.

Bài toán phát hiện giả mạo hình ảnh bằng phương pháp copy-move là một trong những vấn đề được nghiên cứu nhiều nhất trong image forensics [5]. Mục tiêu của bài báo khảo sát này là tổng hợp và đánh giá một số phương pháp tiếp cận hiện có.

1.1 Image forensics

Trong thời đại trước đây, những tấm ảnh được chụp sẽ là bằng chứng thể hiện rằng một sự kiện nào đó đã diễn ra. Tuy nhiên điều này không còn đúng trong thời đại số ngày nay, khi mà việc tạo ảnh và chỉnh sửa ảnh đã trở nên vô cùng dễ dàng. Điều này đang dần đưa chúng ta đến với viễn cảnh mà ta không thể coi tính xác thực và toàn vẹn của hình ảnh là điều hiển nhiên. Và nó đã làm suy yếu đi sự tín nhiệm của những hình ảnh số được sử dụng với mục đích pháp lý như là để làm bằng chứng trình bày trước toà án, tài liệu tài chính, tin tức,...

Để giải quyết những vấn đề này, những nghiên cứu về lĩnh vực image forensics (có thể được dịch là "pháp y hình ảnh số") sẽ giúp ta giải đáp một số câu hỏi như [13]:

- Hình ảnh này có phải là hình ảnh “gốc” hay nó đã được tạo ra bằng cách cắt và dán từ các hình ảnh khác nhau?

- Hình ảnh này có thực sự đại diện cho cảnh gốc hay nó đã bị chỉnh sửa số để lừa dối người xem?
- Những phần nào của hình ảnh đã trải qua quá trình xử lý và đến mức độ nào?
- Hình ảnh này có bắt nguồn từ nguồn X như đã tuyên bố không?

Để giải đáp những câu hỏi trên thì ta sẽ sử dụng những phương pháp khác nhau dựa trên điều kiện đầu vào. Image forensics có thể được chia ra thành ba bài toán lớn:

1. Xác định nguồn gốc ảnh (image source identification)
2. Phát hiện ảnh tổng hợp (synthetic image detection)
3. Phát hiện ảnh giả mạo (image forgery detection)

1.1.1 Xác định nguồn gốc ảnh

Bài toán xác định nguồn gốc ảnh là bài toán nhằm tìm ra những đặc điểm của thiết bị ghi hình (máy ảnh, scanner,...) đã được sử dụng cho việc tạo ra tấm ảnh trên. Sẽ có hai kiểu output mong muốn của bài toán này: thiết bị thu hình thuộc mẫu (model) nào; thiết bị thu hình là thiết bị cụ thể nào.

Ở output đầu tiên, ta sẽ xác định những thuộc tính của loại thiết bị đã được sử dụng để thu hình, nhằm đưa ra kết luận rằng hình được thu bởi model nào. Ví dụ: Iphone 15 Pro Max, Canon 90D với lens 18-55mm,... Và ở output thứ hai, ta sẽ xác định xem thiết bị cụ thể được sử dụng để thu hình là thiết bị nào. Ví dụ: trong 3 chiếc Iphone 15 thì thiết bị nào trong 3 thiết bị trên đã được sử dụng để thu tấm hình đó?

Sự thành công trong việc giải bài toán này sẽ dựa trên giả định là những thiết bị thu hình sẽ có những đặc điểm riêng về phần cứng, dẫn đến sự khác biệt trong hình ảnh thu được. Chẳng hạn như cho dù là cùng thuộc một model thì hình ảnh thu được có thể có sự khác nhau vì những sai số nhỏ trong khâu sản xuất và gia công, hay những ảnh hưởng của người dùng trong quá trình sử dụng (làm xước lens, bụi dính vào cảm biến,...)



Hình 1.1: Một tấm hình được tạo bởi AI

1.1.2 Phát hiện ảnh tổng hợp

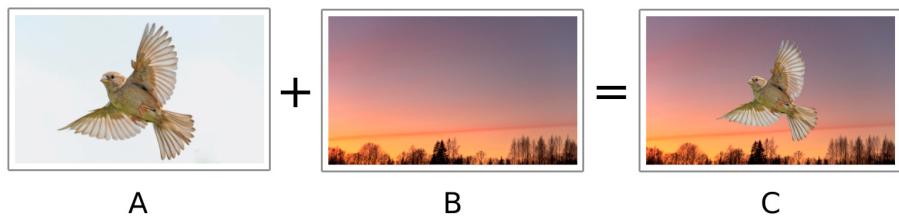
Ảnh tổng hợp là những ảnh được tạo ra một phần hoặc hoàn toàn bởi những công cụ đồ họa. Nó có thể là những hình ảnh được vẽ bởi con người hoặc tạo ra bởi AI.

Hiện nay, những mô hình AI tạo sinh đang phát triển vô cùng mạnh mẽ, điển hình là các mô hình GAN (generative adversarial networks) và diffusion như DALL-E hay Midjourney. Những mô hình này có khả năng tạo ra những tấm ảnh theo nhu cầu của người dùng, có thể là những tấm ảnh với độ chân thực vô cùng cao. Đây là cơ hội rất lớn đối với việc phát triển những lĩnh vực như hội họa và làm game. Tuy nhiên nó có thể bị lạm dụng để tạo ra hình ảnh nhầm phục vụ những mục đích xấu như lừa đảo. Điều này dẫn đến nhu cầu cần phải phát triển những công cụ giúp phân biệt những tấm ảnh tổng hợp tạo ra bởi AI và ảnh thực tế.

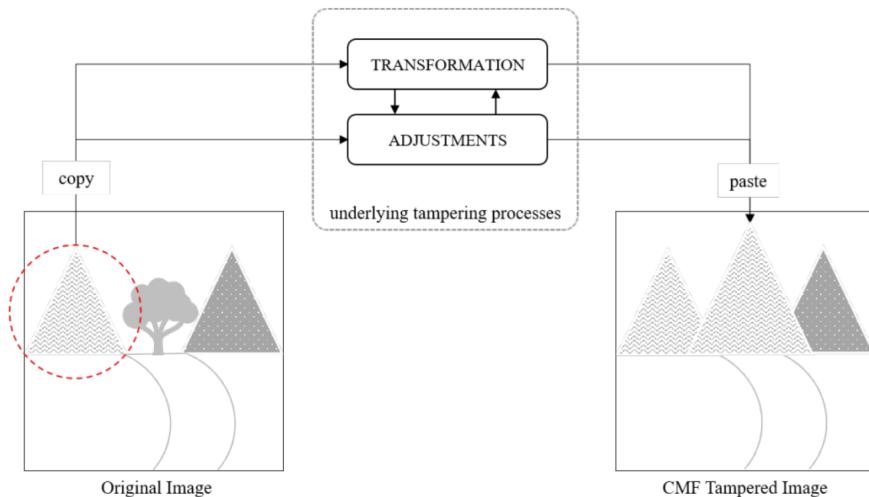
1.1.3 Phát hiện ảnh giả mạo

Ảnh giả mạo là những ảnh được chỉnh sửa với mục đích không tốt. Những hình ảnh này có thể được chỉnh sửa bằng những công cụ phổ biến như Photoshop.

Hai phương pháp chỉnh sửa ảnh phổ biến nhất chính là splicing và



Hình 1.2: Minh họa cho kĩ thuật splicing



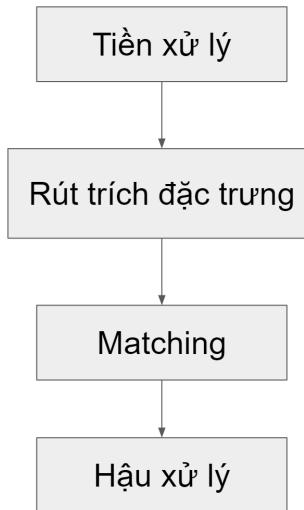
Hình 1.3: Các bước để tạo nên một tấm ảnh copy-move

copy-move. Trong đó splicing là hành động cắt một phần từ ảnh khác và dán vào ảnh hiện tại. Còn copy-move là cắt và dán trong cùng một ảnh.

Để giải quyết bài toán này thì ta sẽ có thể chia ra thành hai nhóm phương pháp: chủ động và bị động.

Để phát hiện ảnh giả mạo một cách chủ động, ta sẽ chèn watermark hoặc chữ ký số vào hình ảnh gốc trước khi gửi hình ảnh đi nơi khác. Và bằng việc so sánh dữ liệu được chèn của ảnh hiện tại và ảnh gốc thì ta có thể biết được rằng hình ảnh đã được qua chỉnh sửa hay chưa. Tuy nhiên, phương pháp này đòi hỏi phần mềm cụ thể để có thể chèn được dữ liệu trước khi phân tán hình ảnh. Và vì không phải tấm hình nào cũng đều được chèn watermark hay chữ ký số nên phương pháp này không đem lại hiệu quả cao với đại đa số hình ảnh.

Phương pháp bị động sẽ không yêu cầu thông tin cụ thể về nội dung hình ảnh hay watermark, mà chỉ cần tấm hình đó. [4].



Hình 1.4: Framework cơ bản của bài toán phát hiện ảnh giả mạo bằng kĩ thuật copy-move

1.1.4 Phát biểu bài toán

Bài toán được trình bày chính trong bài báo cáo này là phát hiện ảnh giả mạo copy-move, bởi vì đây là bài toán được nghiên cứu nhiều nhất trong lĩnh vực phát hiện ảnh được chỉnh sửa (ảnh giả mạo) [1]. Bài toán này yêu cầu đầu vào là một tấm hình muốn xác định, và đầu ra mong muốn sẽ là cho biết hình ảnh đó có được qua chỉnh sửa bởi kĩ thuật copy-move hay không.

- Input: ảnh cần được xác định
- Output: giá trị nhị phân 1/0 tương ứng với ảnh có được chỉnh sửa bởi kĩ thuật copy-move hay không

1.1.5 Framework

Hình 1.4 tương ứng với framework cơ bản để giải quyết bài toán này. Framework gồm 4 bước như sau:

- Tiền xử lý (pre-processing): ta sẽ đưa hình ảnh qua các bước tiền xử lý như chuyển qua ảnh xám, resize ảnh, chuyển về miền tần số, chia thành những khối vuông,... tuỳ thuộc vào mục đích và phương pháp sử dụng để giải quyết bài toán.

- Rút trích đặc trưng (feature extraction): ta sẽ trích xuất những đặc trưng cụ thể của từng vùng trên hình ảnh dựa theo phương pháp giải quyết bài toán và lưu trữ theo cấu trúc mong muốn.
- Matching: dựa trên những đặc trưng của từng vùng đã trích xuất được, ta sẽ tiến hành so sánh những giá trị đặc trưng này với nhau để biết được những vùng đó có giống nhau hay không.
- Hậu xử lý: dựa trên những vùng mà ta đã matching ở bước trên, ta sẽ tiến hành loại bỏ những giá trị outlier và đưa ra kết luận rằng hình ảnh có được chỉnh sửa bởi kĩ thuật copy-move hay không.

Các công trình liên quan

2.1 Bộ dữ liệu

Để phục vụ cho bài toán phát hiện ảnh giả mạo sử dụng kĩ thuật copy-move, các nhà nghiên cứu đã tạo ra những bộ dữ liệu như MICC, COVERAGE, CoMoFoD,...

Ở trong bài báo cáo này, chúng em sử dụng bộ dữ liệu MICC-F220, bởi vì đây là bộ dữ liệu được sử dụng để đánh giá ở trong nhiều công trình nên có thể dễ dàng hơn cho việc so sánh các công trình.

MICC-F220 gồm 220 tấm hình, trong đó 110 là ảnh gốc, còn 110 tấm còn lại là hình ảnh được chỉnh sửa bằng kĩ thuật copy-move. Để tạo ra bộ dữ liệu này thì tác giả chỉ sử dụng hai phương pháp tấn công là xoay và co giãn ảnh.

Ground truth của bộ dữ liệu là 1/0. 1 tương ứng với hình ảnh được chỉnh sửa, 0 tương ứng với hình ảnh gốc.

Ở hình 2.1, hình bên trái là ảnh gốc, còn hình bên phải là ảnh đã được qua chỉnh sửa bằng cách copy-move cánh cửa của căn nhà.



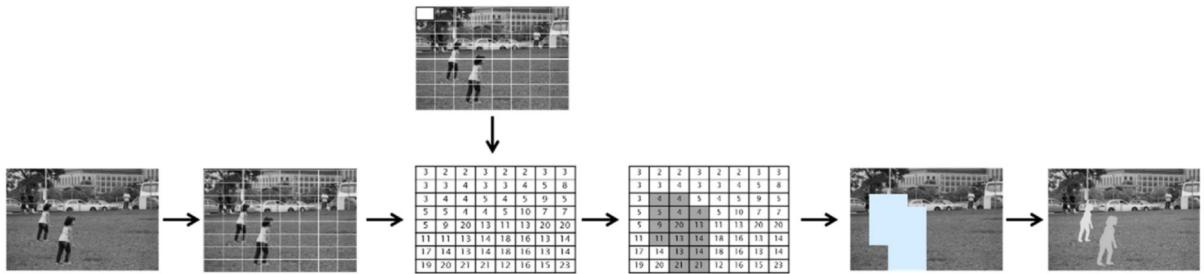
Hình 2.1: Một cặp dữ liệu trong bộ MICC-F220

2.2 Hướng tiếp cận block-based

2.2.1 Nguyên lý

Một trong những cách tiếp cận sơ khai nhất của bài toán này chính là block-based.

Theo hướng tiếp cận này, ta sẽ chia hình ảnh đầu vào thành những khối (có thể đè lên nhau). Sau đó ta sẽ rút trích đặc trưng của từng khối và tiến hành so sánh những khối đó với nhau để tìm ra xem những khu vực khối nào đã được copy-move.



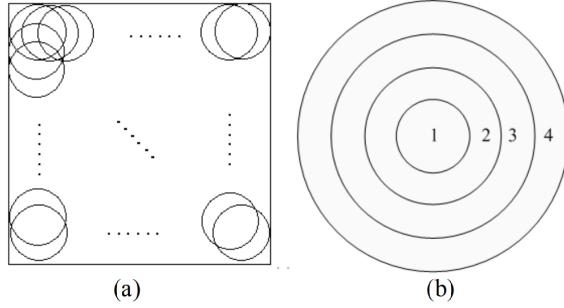
Hình 2.2: Minh họa các bước cơ bản của phương pháp block-based

Hình 2.2 minh họa các bước cơ bản cho một phương pháp sử dụng hướng tiếp cận block-based.

Đa phần những phương pháp sử dụng hướng tiếp cận block-based sẽ khác nhau ở cách rút trích đặc trưng. [7] rút trích đặc trưng của khối bằng cách sử dụng biến đổi cosine rời rạc (discrete cosine transformation) với số chiều vector đặc trưng là 256. Ở phương pháp [12], tác giả rút trích đặc trưng bằng cách tính trung bình giá trị của từng kênh màu Red, Green, Blue của khối tương ứng với ba chiều đầu tiên của vector đặc trưng.

2.2.2 Detection of Image Region Duplication Forgery Using Model with Circle Block (2009)

Trong công trình [17], tác giả sử dụng những khối hình tròn thay vì khối hình vuông. Các bước cụ thể của phương pháp như sau:



Hình 2.3: Khối tròn trong phương pháp block-based

Tiền xử lý Hình ảnh đầu vào sẽ được chuyển thành ảnh xám. Sau đó ta biến đổi ảnh sử dụng Gaussian Pyramid để tạo ảnh G_1 có kích thước bằng $1/4$ ảnh ban đầu. Trên ảnh G_1 , ta sẽ chia hình thành những khối đè lên nhau có hình tròn giống như hình 2.3 (a). Mỗi khối tròn sẽ được chia thành 4 hình tròn đồng tâm $\Omega_1, \Omega_2, \Omega_3, \Omega_4$ với bán kính lần lượt là 1, 2, 3, 4 giống như hình 2.3 (b).

Rút trích đặc trưng Ở bước này, với mỗi khối tròn thì ta sẽ tính một vector đặc trưng $V_i = [\theta_1, \theta_2, \theta_3, \theta_4]$ bằng công thức:

$$\theta_k = \frac{\sum x_{i,j}}{|\Omega_k|}, x_{i,j} \in \Omega_k$$

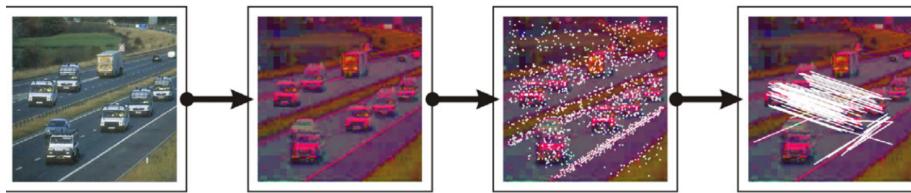
θ cho biết giá trị trung bình của các pixel trong một hình tròn.

Matching Với mỗi cặp khối tròn j, k , ta sẽ tính giá trị tương đồng bằng công thức:

$$SIM(V_j, V_k) = \sqrt{\sum_{i=1}^4 (\theta_i^j - \theta_i^k)^2}$$

Sau đó ta sẽ so sánh giá trị tương đồng trên với ngưỡng T_s . Nếu giá trị tương đồng của hai khối i, j bé hơn hoặc bằng T_s và khoảng cách euclidean của tâm hai khối lớn hơn hoặc bằng T_d thì ta sẽ đánh dấu cặp pixel $I(x_j, y_j) = I(x_k, y_k) = 255$

Hậu xử lý Trên ảnh nhị phân I , ta sẽ thực hiện các phép biến đổi morphology để điền vào những lỗ trong các phần được đánh dấu, và tiến



Hình 2.4: Minh họa các bước cơ bản của phương pháp keypoint-based

hành loại bỏ những vùng có diện tích bé hơn ngưỡng T_a . Nếu ảnh I vẫn còn những vùng được đánh dấu thì ta sẽ kết luận ảnh ban đầu đã qua chỉnh sửa bằng kĩ thuật copy-move.

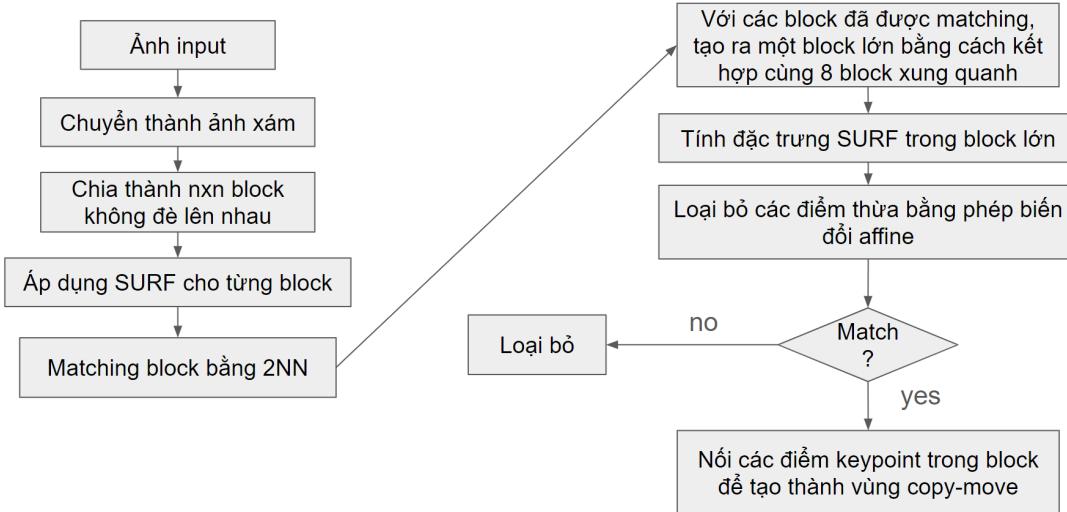
2.3 Hướng tiếp cận keypoint-based

2.3.1 Nguyên lý

Một trong những nhược điểm của hướng tiếp cận block-based là không ổn định với phép xoay và có độ phức tạp tính toán cao. Do đó các nhà nghiên cứu đã phát triển ra một hướng tiếp cận mới sử dụng keypoint. Keypoint được định nghĩa là những điểm có entropy cao, hay là những điểm có tổng độ dị biệt về màu sắc lớn. Thay vì trích xuất đặc trưng của tất cả các khối trong hình ảnh thì ta sẽ chỉ cần tính vector đặc trưng cho các keypoint, và tiến hành matching dựa trên những vector đặc trưng này.

Hình 2.4 minh họa các bước cơ bản của một phương pháp sử dụng hướng tiếp cận keypoint-based. Đa phần các phương pháp keypoint-based sử dụng đặc trưng SURF [3] hoặc SIFT [11]. Điểm khác biệt chủ yếu giữa các phương pháp keypoint-based sẽ nằm ở khâu matching và hậu xử lý.

Ưu điểm của hướng tiếp cận keypoint-based là độ phức tạp tính toán thấp, vì tổng số lượng keypoint trong một hình thường sẽ thấp hơn nhiều so với số lượng khối. Ngoài ra, những phương pháp keypoint-based thường hiệu quả với phép quay và co giãn vì SIFT và SURF bất biến với hai phép biến đổi trên.



Hình 2.5: Tóm tắt các bước thực hiện của công trình [15]

2.3.2 Geometric transformation invariant block based copy-move forgery detection using fast and efficient hybrid local features (2019)

Công trình [15] kết hợp cả hai hướng tiếp cận block-based và keypoint-based. Ở phương pháp này, hình ban đầu sẽ được chia thành các khối vuông không đè lên nhau, sau đó đặc trưng SURF sẽ được trích xuất từ khíaけ keypoint trong những khối vuông này. Hình 2.5 mô tả tóm tắt các bước thực hiện trong phương pháp. Chi tiết của các bước như sau:

Tiền xử lý Ta sẽ chuyển hình ảnh ban đầu thành hình xám. Sau đó chia hình thành các khối $n \times n$ không đè lên nhau (tác giả chọn $n = 8$).

Rút trích đặc trưng Với mỗi khối vuông, những keypoint trong đó sẽ được rút trích đặc trưng SIFT bằng cách tính phản hồi Haar wavelet đối với vùng lân cận của mỗi keypoint. Vector đặc trưng rút trích được sẽ có số chiều là 64.

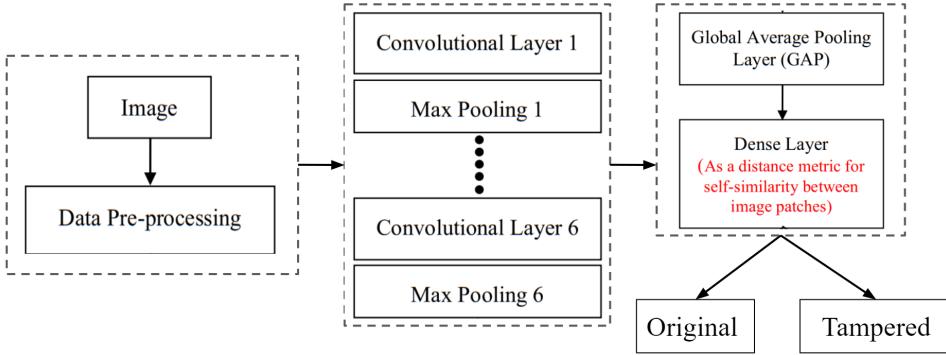
Matching Quá trình matching được thực hiện bằng cách so sánh đặc trưng SURF của từng cặp keypoint với nhau. Nhưng thay vì sử dụng hai vòng lặp để so sánh tất cả các cặp keypoint thì tác giả sử dụng kĩ thuật two nearest neighbor (2NN). Trong 2NN, tỉ lệ khoảng cách từ điểm hiện tại tới lảng giềng gần nhất và lảng giềng gần thứ hai sẽ được so sánh với

một ngưỡng. Gọi d_1 và d_2 lần lượt là khoảng cách euclidean từ vector đặc trưng của keypoint hiện tại tới vector đặc trưng của láng giềng gần nhất và gần thứ hai. Nếu $\frac{d_1}{d_2} \leq \epsilon$ thì keypoint hiện tại với láng giềng gần nhất sẽ được coi là matching. Kết quả thực nghiệm của tác giả cho thấy rằng ngưỡng $\epsilon = 0.6$ là hiệu quả nhất. Và dựa trên vị trí của những keypoint đã được matching thì những cặp khối tương ứng cũng sẽ được xác định là matching.

Kết nối 8 vùng lân cận Để tăng độ hiệu quả của phương pháp, thì với mỗi cặp khối đã được matching, tác giả sẽ mở rộng khối bằng cách liên kết với 8 khối liền kề, tạo thành 2 cặp khối với kích thước gấp 9 lần khối ban đầu. Lí do cho việc tạo ra các khối lớn này là vì kích thước của những vùng bị copy-move trong ảnh gốc có thể lớn hơn kích thước khối 8×8 ban đầu, do đó việc tạo thành những khối lớn này sẽ có thể bao phủ tốt hơn những vùng copy-move đó. Những khối lớn này sẽ được sử dụng cho bước tiếp theo.

Xác định vùng MSER và matching MSER (maximally stable extremal regions) là một phương pháp được sử dụng rộng rãi trong việc tìm những điểm tương đồng giữa hai hình ảnh. Dựa trên những khối lớn đã xác định ở bước trước, ta sẽ xác định vùng MSER tương ứng với các khối này. Sau đó, ở mỗi vùng MSER thì ta sẽ rút trích đặc trưng SURF và tiến hành matching bằng 2NN.

Hậu xử lý Sau khi matching những keypoint ở các vùng MSER ở bước trên, ta sẽ tiến hành tính toán biến đổi affine đã được sử dụng giữa những vùng tương đồng. Sau đó loại bỏ những điểm dư thừa. Những keypoint đã matching còn sót lại sẽ được lưu vào một mảng *Match*. Ảnh được kết luận là đã qua chỉnh sửa bằng kĩ thuật copy-move nếu số phần tử của *Match* ≥ 1 .



Hình 2.6: Framework của [6]

2.4 Hướng tiếp cận deep learning

2.4.1 Nguyên lý

Sự phát triển của deep learning đã mở ra một hướng đi mới trong việc giải quyết bài toán image forensics nói chung và phát hiện ảnh giả mạo nói riêng.

Ở trong bài toán phát hiện ảnh giả mạo bằng kĩ thuật copy-move, những phương pháp deep learning có thể được áp dụng vào khâu trích xuất đặc trưng nhằm học cách để lấy được những đặc trưng mang nhiều ý nghĩa, hoặc áp dụng vào khâu matching để học cách ghép cặp những vùng có đặc trưng giống nhau một cách hiệu quả hơn.

2.4.2 A novel deep learning framework for copy-move forgery detection in images (2020)

Hình 2.6 mô tả framework của công trình [6]. Phương pháp này đạt được độ chính xác cao và tốc độ nhanh dựa trên việc áp dụng mạng nơ-ron tích chập (CNN). Chi tiết cụ thể của mô hình như sau:

Tiền xử lý Hình ảnh đầu vào được resize thành một kích cỡ cố định mà không cắt đi phần nào.

Rút trích đặc trưng Quá trình rút trích đặc trưng được thực hiện thông qua 6 lớp tích chập, theo sau mỗi lớp tích chập là một lớp max-pooling. Mỗi lớp tích chập sẽ tạo ra những feature map nhờ vào filter

cụ thể của nó. Feature map được đưa qua lớp max-pooling cuối cùng sẽ được chuyển thành dạng những vector để làm input cho lớp GAP (global average pooling).

Phân lớp Ở bước phân lớp, một dense layer sẽ được sử dụng như một độ đo tương đồng giữa các vùng trong hình ảnh. Lớp dense layer này có một hàm kích hoạt soft-max nhằm phân lớp hình ảnh đầu vào thành một trong hai lớp: ảnh đã qua chỉnh sửa bởi copy-move hoặc ảnh chưa qua chỉnh sửa.

2.4.3 Một số công trình khác

Nhược điểm chung của nhiều phương pháp deep learning trong bài toán phát hiện ảnh giả mạo bằng kĩ thuật copy-move là chỉ có thể phân lớp hình ảnh đầu vào thành 2 lớp (đã qua chỉnh sửa hoặc chưa) chứ không thể chỉ ra được những vùng copy-move ở trong ảnh.

Công trình [16] (2023) sử dụng bag-of-visual-words (BoVW) và đạt hiệu suất state of the art trong bài toán phát hiện ảnh giả mạo copy-move. Mô hình này có khả năng trả về mask tương ứng với vùng bị copy-move với độ chính xác vô cùng cao.

Công trình [10] (2020) sử dụng SuperGlue đạt hiệu suất ngang với những mô hình state of the art.

Những mô hình sử dụng deep learning có khả năng giải quyết bài toán phát hiện ảnh giả mạo nói chung chư không đơn thuần là copy-move.

Công trình [9] (2023) sử dụng kiến trúc mạng Transformers có khả năng giải quyết cả hai bài toán phát hiện ảnh chỉnh sửa bởi copy-move.

2.5 So sánh

Bảng 2.1 so sánh 3 công trình đã phân tích ở trên qua từng công đoạn. Bảng 2.2 so sánh ưu nhược điểm của 3 hướng tiếp cận. Bảng 2.3 so sánh kết quả thực nghiệm của 3 công trình trên tập dữ liệu MICC-F220.

Công trình	Tiền xử lý	Trích xuất đặc trưng	Matching	Hậu xử lý
[17]	Chuyển thành ảnh xám, chia thành các khối tròn đè lên nhau. Mỗi khối tròn chia thành 4 hình tròn đồng tâm với bán kính 1, 2, 3, 4	Tính giá trị trung bình của các pixel trong 4 hình tròn và ghép thành 1 vector đặc trưng	So sánh từng cặp vector đặc trưng	Loại bỏ những vùng có diện tích nhỏ hơn threshold
[15]	Chuyển thành ảnh xám và chia thành các khối hình vuông không đè lên nhau	Speed up robust features (SURF)	2NN	Tính toán công thức biến đổi affine đã sử dụng để loại các điểm outlier
[6]	Resize ảnh về kích thước định sẵn	Sử dụng các lớp tích chập và pooling để tạo feature map	Sử dụng lớp GAP để chuẩn hóa	Phân lớp bằng hàm kích hoạt soft-max của lớp dense layer

Bảng 2.1: So sánh các bước của 3 công trình

Hướng tiếp cận	Ưu điểm	Nhược điểm
Block-based	Đơn giản	Kém ổn định với phép quay và co giãn, độ phức tạp cao
Keypoint-based	Tốc độ cao, ổn định với phép quay và co giãn	Kém ổn định với những hình có độ tương phản thấp
Deep learning	Độ chính xác cao, có thể áp dụng cả cho bài toán splicing	Phụ thuộc vào dữ liệu huấn luyện, đa phần các mô hình là phân lớp

Bảng 2.2: So sánh ưu nhược điểm của 3 hướng tiếp cận

Công trình	Precision	Recall	F1-score	Tốc độ (s)
[17]	-	-	-	-
[15]	81	97.55	89	8.80
[6]	100	100	100	0.14

Bảng 2.3: Kết quả thực nghiệm của 3 công trình trên tập dữ liệu MICC-F220

Phương pháp truyền thống

3.1 Giới thiệu

Trong phương pháp truyền thống SOTA về bài toán Copy-move Detection, ta sử dụng công trình [14] trong bài báo "**Keypoints based enhanced multiple copy-move forgeries detection system using density-based spatial clustering of application with noise clustering algorithm**". Phương pháp trong bài báo dựa trên công trình [2] sử dụng SIFT Keypoint rất phổ biến. Điểm cải tiến của công trình [14] là trong giai đoạn **Hậu xử lý**, ta sẽ phân cụm bằng thuật toán DBSCAN (Density-based Clustering) thay vì sử dụng HAC (Hierarchical Agglomerative Clustering). Framework của công trình ta đang khảo sát như sau:

- Tiền xử lí: Ta sẽ chuyển ảnh ban đầu sang ảnh xám.
- Rút trích đặc trưng: Sử dụng phương pháp SIFT [11].
- Matching: Sử dụng phương pháp generalized 2NN được trình bày trong công trình [2].
- Hậu xử lý: Sau khi xác định được các cặp Matching Keypoint, ta biểu diễn những Keypoint này trên miền không gian ảnh. Phân cụm chúng bằng thuật toán DBSCAN.

3.2 Rút trích đặc trưng

Trong phần rút trích đặc trưng, ta sử dụng phương pháp SIFT [11] được trình bày bởi David Lowe để lấy Keypoints. Sau khi thực hiện các bước được trình bày trong phương pháp, ta sẽ rút ra được các điểm Keypoints tương ứng với các vector đặc trưng 128 chiều.

3.3 Matching

Sau khi có được một tập các vector đặc trưng tương ứng với từng Keypoints rút trích được ở bước trước đó, ta tiến hành Matching. Giả sử có n Keypoint $\{k_1, k_2, \dots, k_n\}$ ứng với n vector đặc trưng $\{f_1, f_2, \dots, f_n\}$. Xét một keypoint và một vector đặc trưng tương ứng. Ta thực hiện các bước sau:

- B1: Xét khoảng cách Euclidean từ vector đặc trưng đó với $n - 1$ vector đặc trưng còn lại.
- B2: Sắp xếp các khoảng cách tìm được theo thứ tự tăng dần. Giả sử ta có được danh sách đã sắp xếp $\{d_1, d_2, \dots, d_{n-1}\}$
- B3: Thực hiện vòng lặp xét tỉ số khi thỏa $\frac{d_i}{d_{i+1}} < T$. Trong đó, T là ngưỡng ta đặt ra để thỏa hai điểm nối với nhau. Trong bài báo sử dụng $T = 0.6$. Nếu tỉ số $\frac{d_i}{d_{i+1}} \geq T$ thì dừng và ta có tập hợp các cặp vector đặc trưng tương đồng.

Sau khi thực hiện lần lượt với từng vector đặc trưng, ta có được tập hợp những cặp Keypoints tương đồng.

3.4 Hậu xử lý

Ta sẽ sử dụng thuật toán DBSCAN với dữ liệu là vị trí của các Keypoint trên miền không gian ảnh. Giải thuật của thuật toán như sau:

Thuật toán này rất đơn giản, xoay quanh 2 tham số chính là epsilon (ϵ) và min_points. Trong đó:

- Epsilon: Bán kính của vòng tròn thể hiện lân cận của điểm hiện tại.
- min_points: Số điểm tối thiểu cần có trong lân cận của điểm hiện tại.

Từ hai thuộc tính trên, điểm hiện tại (x) sẽ có hai thuộc tính:

- Điểm trung tâm: Nếu số điểm lân cận điểm $x \geq \text{min_points}$.

```

Input : The data set S
Parameter :  $\Psi$  and min_points
procedure DBSCAN( $S$ )
    for each point  $X$  in  $S$  do
        if  $X$  is a core point and not processed then
             $C \leftarrow$  retrieve all points density reachable from  $X$ 
            mark all points in  $C$  as processed
            report  $C$  as a cluster
        else
            mark  $X$  as outliers
        end if
    end for
end procedure

```

Hình 3.1: Giải thuật của thuật toán DBSCAN

- Điểm không trung tâm (điểm biên): Ngược lại trường hợp trên.

Sau khi đã xác định được tính trung tâm của mọi điểm trong tập, ta thực hiện các bước sau:

- B1: Xét lần lượt từng điểm trong tập dữ liệu. Nếu điểm hiện tại là điểm trung tâm và chưa được đánh dấu thì bắt đầu lan rộng bằng cách thêm vào chúng tất cả những điểm lân cận. Có 2 trường hợp xảy ra, nếu điểm lân cận của nó cũng là điểm trung tâm thì thực hiện lan rộng tiếp tục tại điểm đó. Còn nếu điểm lân cận là điểm biên thì không lan rộng tại điểm đó nữa.
- Sau khi đánh dấu xong, những điểm không được đánh dấu sẽ là outlier (điểm ngoài lề).

Sau khi thực hiện phân cụm, ta sẽ tiến hành phân ngưỡng. Nếu số cụm lớn hơn 1 thì sẽ là ảnh giả mạo. (Nghĩa là có 2 vùng giống nhau trở lên). Thuật toán này nhanh ở việc gom cụm những Keypoints trên miền không gian, tuy nhiên nó sẽ không thể hiện vùng bị giả mạo mà chỉ gom nhóm để xét ngưỡng.

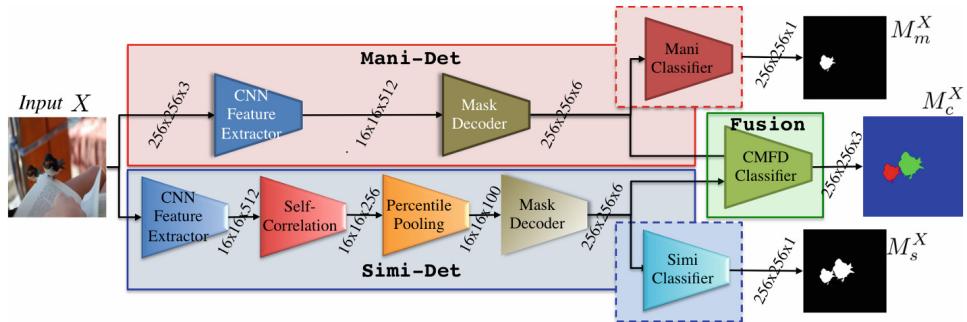
Phương pháp Deep Learning

4.1 Tổng quát

Phần trước đã giới thiệu về hai phương pháp truyền thống chính: block-based và key-point based. Tất nhiên, cả hai phương pháp này đều tồn đọng nhiều vấn đề chẳng hạn đối với block-based sẽ tốn rất nhiều thời gian để cho ra kết quả hay với key-point based khó phát hiện các ảnh có độ tương phản thấp.

Chính vì vậy, để có thể giải quyết các bất cập của phương pháp thủ công, mục tiêu là đưa ra phương pháp DNN-based, áp dụng mạng nơ-ron học sâu vào bài toán CMFD sao cho có thể giải quyết những vấn đề như: mô hình training end-to-end, không cần điều chỉnh tham số hay các quy tắc và có thể cung cấp cho người dùng vùng ảnh giả mạo và vùng ảnh gốc.

Một giải pháp phù hợp là sử dụng một mô hình gồm 2 thành phần đặc trưng hoạt động tách biệt nhau. Đó chính là *BusterNet*[19], một kiến trúc mạng nơ-ron học sâu hai nhánh.



Hình 4.1: Framework của mô hình BusterNet

Trong đó, nhánh **Mani-Det** dùng để phát hiện các vùng ảnh bị thay đổi, chỉnh sửa và nhánh **Simi-Det** dùng để phát hiện vùng ảnh được cho là giống nhau. Cuối cùng ta sẽ dùng cả hai đặc trưng từ hai nhánh này

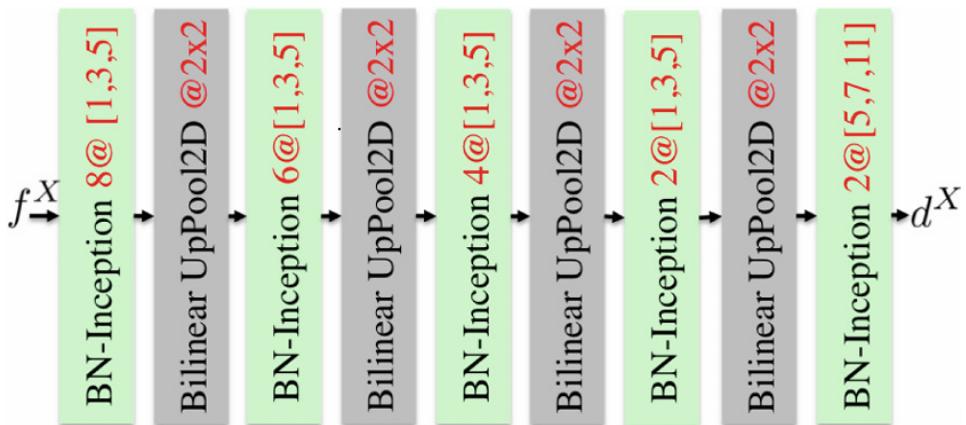
đưa vào **Fusion** để đưa ra dự đoán theo cấp độ pixel phân ra các vùng "*gốc*" tương trưng **màu xanh biển**, "*gốc bị copy-move*" tương trưng **màu xanh lá** và "*được copy-move*" tương trưng **màu đỏ**.

Chúng ta sẽ sử dụng dữ liệu đầu vào là ảnh RGB với kích thước $256 \times 256 \times 3$.

4.2 Nhánh phát hiện chỉnh sửa

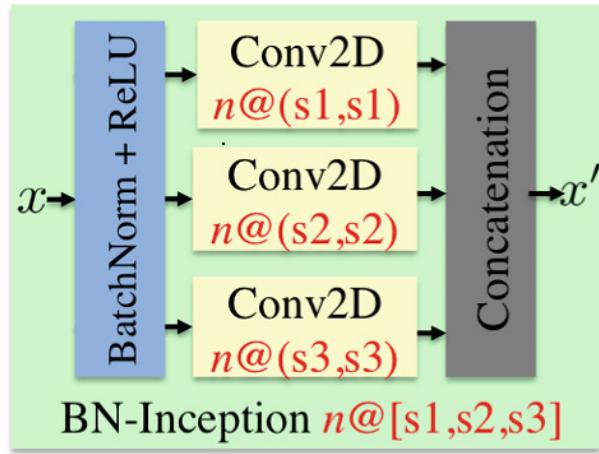
Nhánh phát hiện chỉnh sửa có thể hiểu như một mạng phân loại đặc biệt với mục đích là phân loại vùng bị chỉnh sửa. Cụ thể, với ảnh đầu vào X , rút trích các đặc trưng sử dụng **CNN Feature Extractor**, tăng kích thước của map đặc trưng về kích thước ảnh gốc thông qua **Mask Decoder** và áp dụng **Binary Classifier** để tạo ra vùng chỉnh sửa M_m^X .

Ta có thể sử dụng bất kỳ mạng nơ-ron tích chập nào làm **CNN Feature Extractor**. Đây ta dùng 4 blocks đầu tiên của kiến trúc VGG16 để đơn giản hơn [8]. Kết quả sau khi rút trích sẽ là đặc trưng f_m^X với kích thước $16 \times 16 \times 512$. Sau đó ta cần phải decode đặc trưng này, khôi phục lại độ phân giải gốc tại **Mask Decoder** [18]. Cấu trúc sẽ bao gồm 5 lớp **BN-Inception** và 4 **BilinearUpPool2D** để tạo ra 1 tensor d_m^X với kích thước $256 \times 256 \times 6$. Với 4 lớp **BilinearUpPool2D** [18], chiều không gian ảnh ban đầu là 16×16 sẽ được tăng lên gấp $2^4 = 16$ lần.



Hình 4.2: Mask deconvolution network

Bên trong lớp **BN-Inception** sẽ gồm có 3 lớp **Conv2D**, mỗi lớp có 2 bộ lọc đầu ra nhưng với kích thước kernel 5×5 , 7×7 và 11×11 nhận phản hồi từ lớp BatchNorm + ReLU và gộp lại với nhau thành một chiều $3 \times 2 = 6$.



Hình 4.3: Module BN-Inception

Cuối cùng ta dự đoán M_m^X vùng chỉnh sửa ở cấp độ pixel thông qua **Binary Classifier**, gồm 1 lớp **Conv2D** với bộ lọc kernel 3×3 và 1 hàm kích hoạt sigmoid.

4.3 Nhánh phát hiện tương đồng

Nhánh phát hiện tương đồng lấy ảnh đầu vào X , trích xuất đặc trưng sử dụng **CNN Feature Extractor**, tính toán độ tương đồng của các đặc trưng dựa vào module **Self-Correlation**, lọc ra các thông kê có ích thông qua **Mask Decoder** và áp dụng **Binary Classifier** để tạo ra vùng bị copy-move M_s^X .

4.3.1 Rút trích đặc trưng

Nhánh Simi-Det trích xuất đặc trưng của ảnh thông qua 4 block đầu tiên của mô hình VGG16. Cụ thể, mô hình này chia ảnh thành các block nhỏ và trích xuất đặc trưng của các block này bằng các lớp tích chập. Các đặc trưng f_s^X sau khi được trích xuất sẽ có kích thước $16 \times 16 \times 512$.

4.3.2 Matching

Trước khi matching các đặc trưng, ta cần xác định sự tương quan giữa các đặc trưng này thông qua việc tính hệ số tương quan Pearson ρ .

Xét 2 đặc trưng $f_m^X[i]$ và $f_m^X[j]$, ta tính hệ số tương quan Pearson ρ theo công thức:

$$\rho(i, j) = (\tilde{f}_m^X[i])^T \times \tilde{f}_m^X[j] / 512$$

Trong đó, $(.)^T$ là chuyển vị ma trận và $\tilde{f}_m^X[i]$ là chuẩn hóa của $f_m^X[i]$ bằng cách hiệu với trung bình $\mu_m^X[i]$ và chia cho độ lệch chuẩn $\sigma_m^X[i]$ theo công thức:

$$\tilde{f}_m^X[i] = (f_m^X[i] - \mu_m^X[i]) / \sigma_m^X[i]$$

Lặp lại quá trình này với các $f_m^X[j]$ còn lại và ta sẽ có một vec-tơ điểm $S^X[i]$:

$$S^X[i] = [\rho(i, 0), \dots, \rho(i, j), \dots, \rho(i, 255)]$$

Tiếp tục với $S^X[i]$ còn lại ta sẽ được kết quả là một tensor S^X với kích thước $16 \times 16 \times 256$.

Sau khi có được các ứng viên, ta phải tiến hành chọn các ứng viên phù hợp nhất sao cho mô hình không tồn quá nhiều thời gian mà vẫn giữ được sự chính xác. Để làm được như vậy, tác giả đã đề xuất thêm một lớp gộp **Percentile Pooling**. Đầu tiên, ta sẽ sắp xếp các vec-tơ điểm S^X theo thứ tự giảm dần như sau:

$$S'^X[i] = \text{sort}(S^X[i])$$

Để ý rằng, giả sử ta tạo một đồ thị đường cong $(k, S^X[i][k])$, nếu đặc trưng $f_s^X[i]$ matching với đặc trưng khác thì trên đồ thị lúc này sẽ xuất hiện vài điểm trên đồ thị bị rơi ra. Đây chính là cách mà tác giả đã sử dụng để matching các ứng viên lại với nhau.

Tuy nhiên, vì số lượng ứng viên trong 1 ảnh phụ thuộc vào kích thước của ảnh đó và điều đó làm cho kết quả của mô hình bị thất thoát nên thay vì phải vét cạn hết các ứng viên, tác giả chỉ chọn ra K điểm trong tập S'^X để tạo vec-tơ điểm percentile pool $P^X[i]$:

$$P^X[i][k] = S'^X[i][k']$$

với $k' = \text{round}(p_k \cdot (L - 1))$.

Từ một tensor S^X ma trận $16 \times 16 \times 256$, sau khi được matching với nhau

để tạo ra tensor P^X có kích thước $16 \times 16 \times 100$.

4.3.3 Hậu xử lý

Sau khi qua lớp **Percentile Pooling**, ta sử dụng lớp **Mask Decoder** để đưa kích thước của đặc trưng P^X về với kích thước của ảnh gốc để từ đó qua 1 lớp **Binary Classifier** tạo vùng copy-move M_s^X .

4.4 Fusion

Như đã đề cập ở hình 4.1, **Fusion** module lấy dữ liệu đầu vào từ **Mask Decoder** từ cả 2 nhánh, kí hiệu d_m^X và d_s^X . Cụ thể ta tích hợp 2 đặc trưng d_m^X và d_s^X sử dụng **BN-Inception** với tham số $3@[1, 3, 5]$ và sử dụng **Conv2D** với một bộ lọc kernel kích thước 3×3 và một hàm kích hoạt softmax để đưa ra dự đoán 3 lớp phân vùng.

4.5 Hàm loss

Để train mô hình BusterNet, thay vì train tất cả các module với nhau, tác giả tách 3 phần riêng biệt nhau - (i) mỗi nhánh sẽ được train cho tác vụ của nhánh đó, (ii) sau đó đóng băng 2 nhánh lại và tiếp tục train cho module **fusion**, (iii) cuối cùng bỏ đóng băng toàn bộ mô hình và fine-tune lại BusterNet. Đối với các tác vụ phụ như tạo mask của từng nhánh, tác giả sử dụng Adam optimizer với tốc độ học $1e - 2$ kèm với hàm loss **binary _ crossentropy**. Đối với tác vụ chính như gộp mask và sinh output, tác giả cũng sử dụng Adam optimizer nhưng với hàm loss **category _ crossentropy** và tốc độ học $1e - 2$ khi training module **fusion** và $1e - 5$ khi fine-tune.

Cài đặt và thử nghiệm

5.1 Phương pháp truyền thống

5.1.1 Dataset và Metrics

Ta sẽ sử dụng bộ Dataset MICC-F220 đã được trình bày ở phía trên. Đồng thời, độ đo được sử dụng để kiểm tra sẽ là *F1_score* và *Accuracy*. Vì Output groundtruth của dataset trả về 0 hoặc 1 nên ta sẽ xem độ dự đoán chính xác đúng dựa trên điều này.

5.1.2 Cài đặt thuật toán

Bộ dataset này được public trên nền tảng Kaggle. Vì thế, nhóm sẽ cài đặt và sử dụng môi trường Notebook trên chính nền tảng này.

Trong cài đặt thuật toán, nhóm sử dụng OpenCV với các thư viện hỗ trợ trong việc tính SIFT. Nhóm cài đặt thủ công g2NN với sự hỗ trợ của hàm BFMatcher() do **OpenCV** cung cấp. Trong bước hậu xử lý, sử dụng thư viện DBSCAN của thư viện **sklearn**.

Một số thay đổi so với công trình đã trình bày ở mục 3:

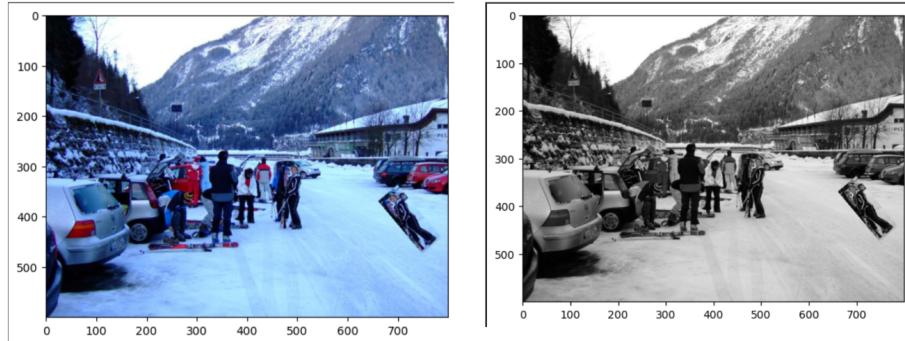
- Trong bước cài đặt matching, nhóm chỉ sử dụng 20 khoảng cách gần nhất thay vì xét hết trên toàn bộ $n - 1$ các vector đặc trưng còn lại. Việc này cũng được đề cập trong công trình rằng, với những vector đặc trưng có tương đồng thì khoảng cách của chúng có xu hướng gần và ngược lại. Khoảng cách gần và xa này rất lớn dựa trên quan sát nhóm nghiên cứu công trình. Vì thế, nhóm tận dụng việc này để giảm thiểu tính toán.
- Trong bước DBSCAN, khi xét các Keypoint trên miền không gian, nhóm đưa miền ảnh về cùng kích cỡ 1x1 để có thể thực hiện trên nhiều

ảnh có nhiều kích cỡ. Đồng thời, nhóm sử dụng tham số epsilon = 0.08 và min_points = 5.

5.1.3 Kết quả thuật toán

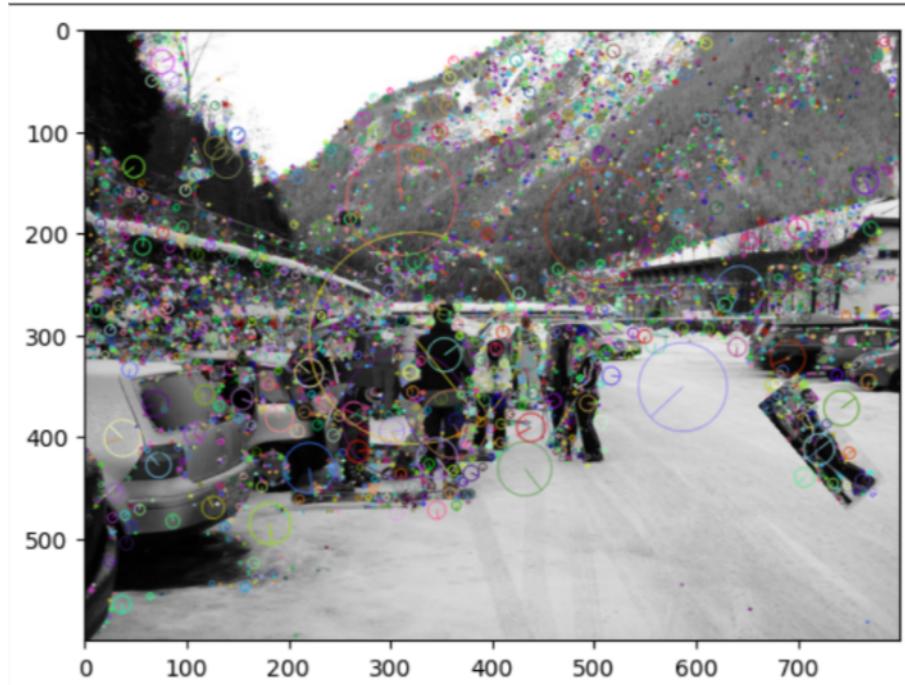
Kết quả với một ảnh

Bước tiền xử lí, chuyển ảnh sang ảnh xám:



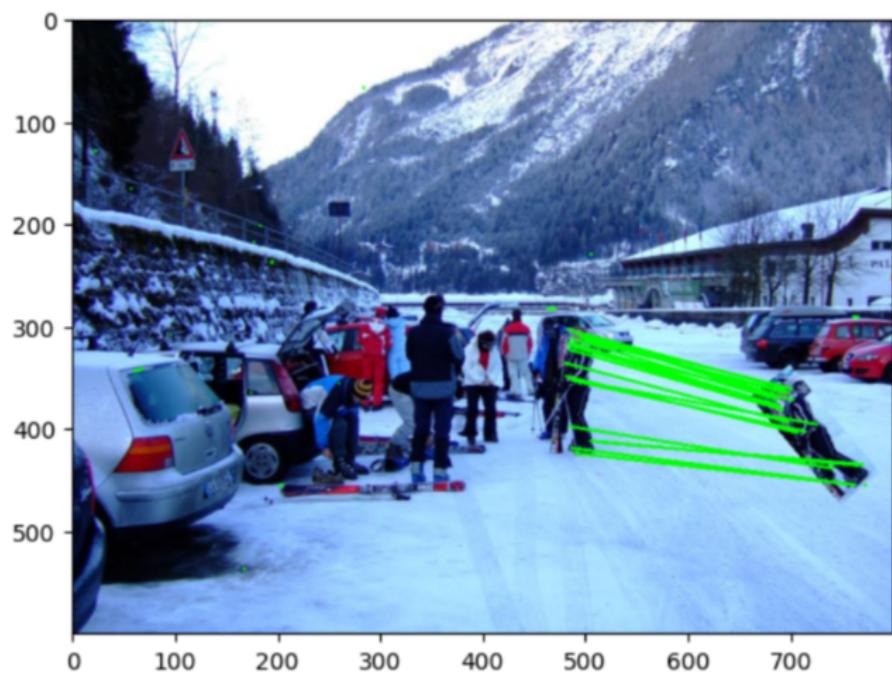
Hình 5.1: Tiền xử lí ảnh

Trích xuất đặc trưng: Sử dụng SIFT tìm ra các Keypoints.



Hình 5.2: Feature Extraction: SIFT Keypoints

Matching: Sử dụng g2NN và biểu diễn.



Hình 5.3: Matching Keypoints

Sau đó sử dụng DBSCAN và lọc ra kết quả bằng cách đếm số nhóm.

Kết quả trên toàn bộ tập dữ liệu và đánh giá

Sau khi thực hiện trên 220 tấm ảnh, nhóm thu được kết quả sau:

- Thời gian chạy: 70.86s
- Recall: 0.936
- Precision: 0.824
- Accuracy: 0.868
- F_1 score: 0.876

Kết quả cho thấy Recall cao tức số ảnh dự đoán đúng là ảnh giả mạo, thỏa yêu cầu bài toán. Tuy nhiên, Precision lại khá thấp, thể hiện rằng mô hình có xu hướng cho ra kết luận là ảnh giả mạo.

5.2 Phương pháp Deep Learning

5.2.1 Dataset và Metrics

Ta sẽ sử dụng bộ dataset CASIA CMFD được public trên Kaggle. Đây là bộ dataset mà tác giả của mô hình này đã tinh chỉnh lại từ bộ dataset CASIA gốc, đó là thêm groundtruth của vùng bị copy-move. Vì output của mô hình này là thể hiện vùng bị copy-move nên không có bộ dataset nào cung cấp output đủ chính xác. Đồng thời, độ đo được sử dụng cho mô hình này là *precision*, *recall* và *F₁* scores.

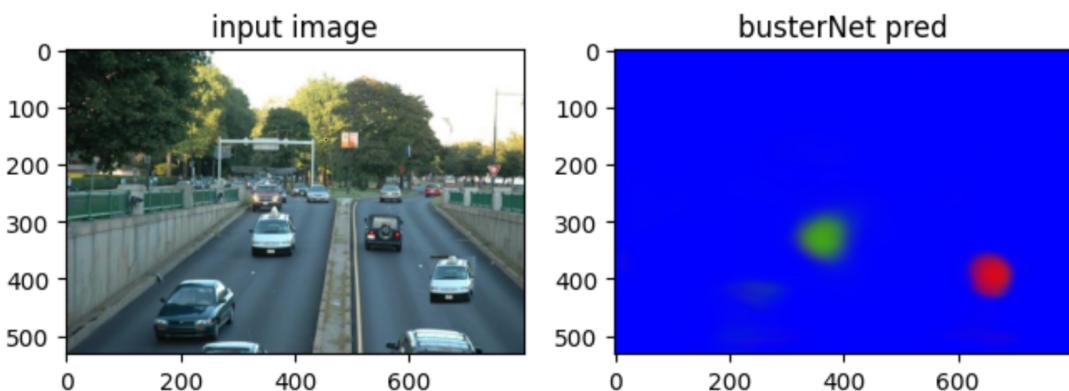
5.2.2 Cài đặt thuật toán

Tất cả các modules trong mô hình đều được cài đặt từ những layers cơ bản ngoại trừ modules **Self-Correlation** và **Percentile Pooling**. Đối với **Self-Correlation**, ta sẽ cài đặt dựa trên các công thức toán đã giới thiệu.

Còn với **Percentile Pooling** sẽ là hàm pooling có tính quyết định. Tuy nhiên đối với việc cài đặt chúng ta chỉ cần cài đặt tương tự **MaxPooling** ở phần back-propagation.

Kết quả thuật toán

Kết quả một ảnh



Các vùng có màu xanh dương là vùng gốc, xanh lá là vùng gốc bị copy-move, đỏ là vùng đã được copy-move.

Kết quả trên bộ dữ liệu CASIA CMFD

- Thời gian chạy: 35 phút
- Recall: 73.89
- Precision: 78.22
- F_1 score: 75.98

Có thể thấy kết quả trên bộ dataset ở mức khá nhưng vì hạn chế của bộ dataset nên nhóm cũng hài lòng với kết quả đạt được.

Kết luận và hướng phát triển

Bài báo cáo khảo sát này đã cung cấp một cái nhìn cơ bản về các kỹ thuật và phương pháp được dùng trong lĩnh vực image forensics nói chung và phát hiện ảnh giả mạo bởi kỹ thuật copy-move nói riêng. Các hướng tiếp cận truyền thống block-based, keypoint-based cũng như những tiến bộ gần đây trong hướng tiếp cận sử dụng mạng học sâu đã được thảo luận.

Tuy lĩnh vực này đã có những tiến bộ đáng kể nhưng vẫn còn đó một số thách thức cần được giải quyết trong tương lai. Một trong số đó là việc nghiên cứu nên một phương pháp có thể phát hiện hầu hết các phép chỉnh sửa ảnh giả mạo với độ chính xác cao, bất biến với các phép tấn công và có chi phí tính toán thấp.

Ngoài ra, với sự phát triển của những công cụ chỉnh sửa thì các phương pháp tạo ảnh giả mạo mới có thể xuất hiện trong tương lai, điều này đòi hỏi việc nghiên cứu và phát triển liên tục của các phương pháp phát hiện giả mạo mới.

Một trong những tiềm năng phát triển hiện nay chính là việc sử dụng mô hình ngôn ngữ - ảnh để tận dụng được sức mạnh của những mô hình ngôn ngữ.

Tóm lại, mặc dù lĩnh vực phát hiện copy-move đã đạt được nhiều kết quả tốt, nhưng vẫn còn đó những công việc cần làm. Hy vọng rằng bài khảo sát này sẽ có thể cung cấp những thông tin hữu ích cho người đọc.

Tài liệu tham khảo

Danh sách

- [1] Nor Bakiah Abd Warif et al. “Copy-move forgery detection: survey, challenges and future directions”. In: *Journal of Network and Computer Applications* 75 (2016), pp. 259–278.
- [2] Irene Amerini et al. “A SIFT-Based Forensic Method for Copy–Move Attack Detection and Transformation Recovery”. In: *IEEE Transactions on Information Forensics and Security* 6.3 (2011), pp. 1099–1110. DOI: [10.1109/TIFS.2011.2129512](https://doi.org/10.1109/TIFS.2011.2129512).
- [3] Herbert Bay, Tinne Tuytelaars, and Luc Van Gool. “Surf: Speeded up robust features”. In: *Computer Vision–ECCV 2006: 9th European Conference on Computer Vision, Graz, Austria, May 7–13, 2006. Proceedings, Part I* 9. Springer. 2006, pp. 404–417.
- [4] Gajanan K Birajdar and Vijay H Mankar. “Digital image forgery detection using passive techniques: A survey”. In: *Digital investigation* 10.3 (2013), pp. 226–245.
- [5] Vincent Christlein et al. “An evaluation of popular copy-move forgery detection approaches”. In: *IEEE Transactions on information forensics and security* 7.6 (2012), pp. 1841–1854.
- [6] Mohamed A Elaskily et al. “A novel deep learning framework for copy-moveforgery detection in images”. In: *Multimedia Tools and Applications* 79 (2020), pp. 19167–19192.
- [7] Jessica Fridrich, David Soukal, Jan Lukas, et al. “Detection of copy-move forgery in digital images”. In: *Proceedings of digital forensic research workshop*. Vol. 3. 2. Cleveland, OH. 2003, pp. 652–63.

- [8] Andrew Zisserman Karen Simonyan. “VERY DEEP CONVOLUTIONAL NETWORKS FOR LARGE-SCALE IMAGE RECOGNITION”. In: *IEEE Signal Processing Letters* (2014). URL: <https://doi.org/10.48550/arXiv.1409.1556>.
- [9] Yaqi Liu et al. “TBFormer: Two-Branch Transformer for Image Forgery Localization”. In: *IEEE Signal Processing Letters* 30 (2023), pp. 623–627. ISSN: 1558-2361. DOI: 10.1109/lsp.2023.3279018. URL: <http://dx.doi.org/10.1109/LSP.2023.3279018>.
- [10] Yaqi Liu et al. *Two-Stage Copy-Move Forgery Detection with Self Deep Matching and Proposal SuperGlue*. 2020. arXiv: 2012.08697 [cs.CV].
- [11] David G Lowe. “Distinctive image features from scale-invariant keypoints”. In: *International journal of computer vision* 60 (2004), pp. 91–110.
- [12] Weiqi Luo, Jiwu Huang, and Guoping Qiu. “Robust detection of region-duplication forgery in digital image”. In: *18th International Conference on Pattern Recognition (ICPR’06)*. Vol. 4. IEEE. 2006, pp. 746–749.
- [13] Husrev T Sencar and Nasir Memon. “Overview of state-of-the-art in digital image forensics”. In: *Algorithms, Architectures and Information Systems Security* (2009), pp. 325–347.
- [14] Badal Soni, Pradip Das, and Dalton Thounaojam. “Keypoints based enhanced multiple copy-move forgeries detection system using density-based spatial clustering of application with noise clustering algorithm”. In: *IET Image Processing* 12 (July 2018). DOI: 10.1049/iet-ipr.2018.5576.
- [15] Badal Soni, Pradip K Das, and Dalton Meitei Thounaojam. “Geometric transformation invariant block based copy-move forgery detection using fast and efficient hybrid local features”. In: *Journal of information security and applications* 45 (2019), pp. 44–51.

- [16] Chao Wang et al. “Shrinking the semantic gap: spatial pooling of local moment invariants for copy-move forgery detection”. In: *IEEE Transactions on Information Forensics and Security* 18 (2023), pp. 1064–1079.
- [17] Junwen Wang et al. “Detection of image region duplication forgery using model with circle block”. In: *2009 International conference on multimedia information networking and security*. Vol. 1. IEEE. 2009, pp. 25–29.
- [18] Zbigniew Wojna et al. “The devil is in the decoder: Classification, regression and gans”. In: *International Journal of Computer Vision* 127 (2019), pp. 1694–1706.
- [19] Yue Wu, Wael Abd-Almageed, and Prem Natarajan. “Busternet: Detecting copy-move image forgery with source/target localization”. In: (2018), pp. 168–184.