**BỘ GIÁO DỤC VÀ ĐÀO TẠO**

**TRƯỜNG ĐẠI HỌC MỞ THÀNH PHỐ HỒ CHÍ MINH**

**NGUYỄN THANH NỞ**

**<TÊN ĐỒ ÁN NGÀNH>**

**ĐỒ ÁN NGÀNH**

**NGÀNH CÔNG NGHỆ THÔNG TIN**

**TP. HỒ CHÍ MINH, 2025BỘ GIÁO DỤC VÀ ĐÀO TẠO**

**TRƯỜNG ĐẠI HỌC MỞ THÀNH PHỐ HỒ CHÍ MINH**

**NGUYỄN THANH NỞ**

**< TÊN ĐỒ ÁN NGÀNH >**

**Mã số sinh viên: 2251050056**

**ĐỒ ÁN NGÀNH**

**NGÀNH CÔNG NGHỆ THÔNG TIN**

**Giảng viên hướng dẫn: TS. Trương Hoàng Vinh**

**TP. HỒ CHÍ MINH, 2025**

LỜI CẢM ƠN

Trong thời gian làm đồ án ngành, em đã nhận được những sự giúp đỡ, đóng góp ý kiến và chỉ bảo nhiệt tình của thầy cô, gia đình, bạn bè.

Em xin phép được gửi sự tri ân sâu sắc và lời cảm ơn chân thành nhất đối với các thầy cô của khoa Công nghệ thông tin đã truyền đạt những tri thức quý báu cho em trong suốt quá trình học tập tại trường Đại học Mở Thành phố Hồ Chí Minh. Đặc biệt, em xin trân trọng cảm ơn thầy Trương Hoàng Vinh đã nhiệt tình hướng dẫn để em có thể hoàn thành tốt bài báo cáo đồ án ngành này.

Em xin kính chúc sức khỏe đến tất cả quý thầy cô đang công tác tại trường Đại học Mở Thành phố Hồ Chí Minh và sự nghiệp giảng dạy sẽ ngày càng thành công hơn, gặt hái được nhiều thành quả tốt. Chúc trường sẽ luôn là nền tảng vững chắc cho nhiều thế hệ sinh viên trên con đường học tập.

Có thể đồ án ngành hoặc bài báo cáo của em sẽ có những chỗ thiếu sót, chưa làm hài lòng được thầy cô nhưng em mong sẽ nhận được sự cảm thông của thầy cô. Em sẽ cố gắng rút kinh nghiệm sau khi nhận được những lời nhận xét của thầy cô và cố gắng hơn nữa trên con đường học tập của mình cũng như con đường để em bước gần hơn với sự nghiệp tương lai phía trước.

Em xin chân thành cảm ơn!

NHẬN XÉT CỦA GIÁO VIÊN HƯỚNG DẪN

TÓM TẮT ĐỐ ÁN NGÀNH

ABSTRACT

**MỤC LỤC**

[LỜI CẢM ƠN 1](#_Toc202683267)

[NHẬN XÉT CỦA GIÁO VIÊN HƯỚNG DẪN 2](#_Toc202683268)

[TÓM TẮT ĐỐ ÁN NGÀNH 3](#_Toc202683269)

[ABSTRACT 4](#_Toc202683270)

[DANH MỤC TỪ VIẾT TẮT 7](#_Toc202683271)

[DANH MỤC HÌNH VẼ 8](#_Toc202683272)

[DANH MỤC BẢNG 9](#_Toc202683273)

[MỞ ĐẦU 10](#_Toc202683274)

[Chương 1. TỔNG QUAN ĐỀ TÀI 11](#_Toc202683275)

[1.1. Giới thiệu đề tài 11](#_Toc202683276)

[1.2. Lý do chọn đề tài 11](#_Toc202683277)

[1.3. Mục tiêu đề tài 11](#_Toc202683278)

[1.4. Phương pháp thực hiện 11](#_Toc202683279)

[1.5. Bố cục đề tài 11](#_Toc202683280)

[Chương 2. CƠ SỞ LÝ THUYẾT 12](#_Toc202683281)

[2.1. Chatbot 12](#_Toc202683282)

[2.2. Spring Boot 12](#_Toc202683283)

[2.2.1. Giới thiệu Spring Boot 12](#_Toc202683284)

[2.2.2. Kiến trúc Spring Boot 12](#_Toc202683285)

[2.2.3. ... 12](#_Toc202683286)

[2.3. Fluter 13](#_Toc202683287)

[2.4. Các công cụ và thư viện bổ trợ 13](#_Toc202683288)

[2.4.1. Docker 13](#_Toc202683289)

[2.4.2. CD/CI 13](#_Toc202683290)

[2.4.3. Redis 13](#_Toc202683291)

[Chương 3. HỆ THỐNG (MÔ HÌNH) <TÊN> 14](#_Toc202683292)

[3.1. Giới thiếu hệ thống (mô hình) 14](#_Toc202683293)

[3.2. Kiến trúc hệ thống 14](#_Toc202683294)

[3.3. Phân tích hệ thống 14](#_Toc202683295)

[3.4. Thiết kế hệ thống 14](#_Toc202683296)

[3.4.1. Thiết kế cơ sở dữ liệu 14](#_Toc202683297)

[3.4.2. Thiết kế giao diện 14](#_Toc202683298)

[3.4.3. ... 14](#_Toc202683299)

[3.5. Kết quả đạt được 14](#_Toc202683300)

[Chương 4. KẾT LUẬN VÀ HƯƠNG PHÁT TRIỂN 15](#_Toc202683301)

[4.1. Kết luận 15](#_Toc202683302)

[4.2. Hướng phát triển 15](#_Toc202683303)

[TÀI LIỆU THAM KHẢO 16](#_Toc202683304)

[PHỤ LỤC 17](#_Toc202683305)

DANH MỤC TỪ VIẾT TẮT

DANH MỤC HÌNH VẼ

[Hình 1.1: Tên hình 1 10](#_Toc367742554)

DANH MỤC BẢNG

[Bảng 1.1: Tên bảng 1 10](#_Toc367742567)

[Bảng 2.1: Tên bảng 1 11](#_Toc367742568)

MỞ ĐẦU

# TỔNG QUAN ĐỀ TÀI

## Giới thiệu đề tài

Trong bối cảnh hiện nay, các hệ thống quảng cáo được triển khai đồng bộ trên website của nhiều chi nhánh hoặc cửa hàng. Quảng cáo này có thể ở nhiều dạng như tĩnh (hình ảnh), động (video), âm thanh (audio)… nhưng việc có tuân thủ quảng cáo gốc hay không lại là một vấn đề rất lớn. Thực tế, những sự sai lệch nhỏ như thêm bớt nội dung, chỉnh sửa màu, thay đổi chữ, di chuyển biểu tượng, cắt bỏ hoặc thay đổi lớn có thể làm sai đi thông điệp vốn có của quảng cáo.

Cách để giải quyết vấn đề này là phải đi từng website của các chi nhánh để kiểm tra, quá trình kiểm tra thủ công này rất tốn thời gian, dễ bỏ sót những thay đổi nhỏ nhưng quan trọng (ví dụ thông tin giá, khuyến mãi…). Lâu dài có thể gây ảnh hưởng nghiêm trọng đến uy tín thương hiệu cũng như độ tin cậy của khách hàng.

Sau khi nhận thấy nguy cơ tiềm ẩn của vấn đề, em quyết định tiến hành nghiên cứu và triển khai hệ thống kiểm thử giao diện tự động cho quảng cáo trực tuyến. Hệ thống này có thể tự động thu thập những hình ảnh quảng cáo được chèn trực tiếp trên website sau đó so sánh với quảng cáo gốc dựa trên vị trí HTML đã được thống nhất. Qua đó, giúp phát hiện nhanh chóng các vi phạm trong việc triển khai quảng cáo tại từng chi nhánh, đảm bảo tính đồng bộ, chính xác và uy tín cho doanh nghiệp.

## Lý do chọn đề tài

Trong hoạt động quảng cáo, việc triển khai đồng bộ và chính xác nội dung quảng cáo từ nhà cung cấp đến các chi nhánh là yếu tố then chốt để đảm bảo tính thống nhất của thương hiệu. Nhưng thực tế, yếu tố trên không dễ dàng thực hiện, nếu chỉ có vài chi nhánh sử dụng hình ảnh quảng cáo thì có thể đảm bảo nhưng nếu có đến vài trăm thậm chí là vài nghìn chi nhánh sử dụng quảng cáo và đâu đó trong những chi nhánh này không tuân thủ quảng cáo ban đầu thì việc kiểm tra thủ công của từng chi nhánh rất tốn chi phí, thời gian, nhân lực…

Để giải quyết vấn đề này trong phạm vi đồ án, em tập trung nghiên cứu và xây dựng hệ thống kiểm thử giao diện quảng cáo tự động với các chức năng như tự động thu thập hình ảnh quảng cáo được chèn trực tiếp trên website chi nhánh. Người dùng chỉ cần cung cấp hình ảnh quảng cáo gốc, đường dẫn website và vị trí thẻ HTML (đã được thống nhất giữa nhà cung cấp và chi nhánh). Hệ thống sẽ tiến hành so sánh hình ảnh quảng cáo hiển thị thực tế với ảnh gốc, từ đó phát hiện các trường hợp sai lệch hoặc vi phạm nguyên tắc ban đầu.

## Mục tiêu đề tài

Mục tiêu đề tài là nghiêm cứu và xây dựng một hệ thống kiểm thử giao diện quảng cáo tự động, cho phép phát hiện và phân loại mức độ sai lệch giữa quảng cáo gốc và quảng cáo được triển khai trên các website chi nhánh. Thay vì phải phụ thuộc vào phương pháp kiểm tra thủ công vốn tốn nhiều thời gian, chi phí và dễ bỏ sót chi tiết, hệ thống hướng tới việc tự động hóa toàn bộ quy trình kiểm tra, từ khâu thu thập dữ liệu đến phân tích, đối chiều và báo cáo kết quả.

## Phương pháp thực hiện

Trải qua quá trình phát triển mô hình, em quyết định đưa mô hình đến 3 nhãn để phân loại sai lệch: giống (similar), sai lệch nhẹ chấp nhận được (almost), sai lệch cần cảnh báo (different).

Ngôn ngữ và thư viện: Python, OpenCV (xử lý ảnh, Canny, ORB, histogram), cikit-learn (SVM, RandomForest, kNN, GroupKFold), XGBoost, Tesseract OCR (trích xuất văn bản), Numpy / Pandas / Seaborn (phân tích)

Các giai đoạn tiến hóa mô hình:

* Giai đoạn 1: SSIM, Histogram và Layout với nhãn 2 lớp (similar/different).
* Giai đoạn 2: Thêm OCR (tỷ lệ tương đồng từ khóa), mở rộng thành nhãn 3 lớp (similar/almost/different) để tách sai lệch nhẹ.
* Giai đoạn 3: Thêm ORB (keypoints) xử lý thay đổi bố cục nhỏ (icon, nút…).
* Version 5: Giữ đầy đủ 5 đặc trưng (SSIM, OCR, Histogram, ORB, Layout) để so sánh đa thuật toán (SVM, RF, kNN, XGBoost) với mục đích chọn phương án cân bằng hiệu năng, ổn định.

## Bố cục đề tài

Báo cáo có 4 chương

Chương 1. Tổng quan đề tài: -> mục tiêu chương

Chương 2. Cơ sở lý thuyết: -> mục tiêu chương

...

...

# CƠ SỞ LÝ THUYẾT

## Tổng quan bài toán kiểm thử giao diện tự động

Bài toán đặt ra là so sánh quảng cáo hiển thị (phiên bản triển khai tại chi nhánh) với mẫu gốc để phát hiện sai lệch và phân hạng (similar / almost / different). Hệ thống cần mức hiểu ngữ nghĩa như thay chữ, đổi icon nhỏ, dịch chuyển bố cục cục bộ, thay tông màu… Do đó cần tổ hợp nhiều đặc trưng và mô hình phân loại giám sát. Hướng tiếp cận này thuộc nhóm Visual Compliance Testing ứng dụng ML cổ điển.

## Nguy cơ “data leakage” và lý do dùng GroupKFold

Nếu dùng train\_test\_split ngẫu nhiên, accuracy có thể tăng ảo vì data leakage xảy ra khi biến thể của cùng một ảnh gốc (original\_x) xuất hiện ở cả train và test, mô hình ghi nhớ đặc trưng cụ thể thay vì học quy luật tổng quát. Hậu quả xảy ra là accuracy ảo, sụt mạnh khi gặp nhóm gốc mới. GroupKFold (group = id ảnh gốc) giải quyết bằng cách đưa toàn bộ biến thể của một gốc vào đúng một fold test ở mỗi lượt. Điều này mô phỏng tình huống triển khai khi gặp chi nhánh mới hoặc biến thể chưa thấy. Cách dùng là tạo mảng groups (cùng chiều dữ liệu), truyền vào gkf.split(X, y, groups).

## Thuật toán SVM (Support **Vector** Machine)

### Giới thiệu SVM

SVM được chọn làm trụ cột giai đoạn đầu vì phù hợp không gian đặc trưng chiều thấp (5 đặc trưng) và khả năng tạo ranh giới quyết định rõ ràng giữa các mức sai lệch. Kernel linear là lựa chọn nhất quán tốt nhất trong toàn bộ quá trình phát triển và lý do em chọn thuật toán này ngay từ ban đầu bởi vì lý do ít cần nhiều siêu tham số, dễ phát hiện overfitting qua biến thiên giữa các fold, ổn định trên dữ liệu không quá lớn.

### Triển khai SVM

Trong triển khai, phiên bản nhị phân ban đầu dùng SVM linear làm baseline ở các giai đoạn phát triển mô hình, khi thêm các đặc trưng phi tuyến tính và trải qua quá trình tuning siêu tham số thì kernel linear vẫn là tham số tối ưu nhất. Quy trình thực hiện là trích vector đặc trưng, chuẩn hóa bằng StandardScaler (tránh một chiều chi phối), huấn luyện theo GroupKFold để bảo toàn nhóm original\_x. class\_weight='balanced' giúp giữ recall lớp different. Sau huấn luyện, đánh giá bằng ma trận nhầm lẫn gộp với mục đích nếu thấy almost bị kéo về similar nhiều thì chúng ta có thể cân nhắc điều chỉnh C hoặc bổ sung thêm các đặc trưng cần thiết.

## Thuật toán RF (Random Forest)

### Giới thiệu RF

Random Forest được đưa vào như một đối trọng dạng “tập hợp cây” để kiểm chứng mức ổn định và để đọc độ quan trọng đặc trưng. Việc nhiều cây trên các bootstrap khác nhau giảm nguy cơ lệ thuộc vào nhiễu cục bộ của một đặc trưng (ví dụ OCR dao động khi chữ mờ). Nó bổ sung góc nhìn khác nhau và cho thấy nếu một đặc trưng kỳ vọng quan trọng nhưng weight thấp, đó là tín hiệu cần xem lại cách tính.

### Triển khai RF

Trong thực hiện, với số cây vừa đủ để ổn định sai số, giới hạn độ sâu để tránh học quá kỹ từng nhóm nhỏ. Không cần chuẩn hóa vì cây ra quyết định dựa trên ngưỡng. class\_weight='balanced' cân bằng ảnh hưởng lớp hiếm. Sau 5 folds, độ lệch chuẩn nhỏ cho thấy tính ổn định, feature\_importances\_ được ghi lại để đối chiếu với XGBoost. Khi nhận thấy hiệu năng kém nhẹ so với SVM/XGBoost nhưng vẫn trong biên chấp nhận, Random Forest vẫn giữ vai trò mô hình giải thích bổ sung thay vì mô hình triển khai chính.

## Thuật toán kNN (k-Nearest Neighbors)

### Giới thiệu kNN

kNN đóng vai trò baseline phi tham số nhằm kiểm tra “tính phân tách tự nhiên” của không gian đặc trưng. Nếu không gian đủ rõ ràng, kNN với k nhỏ sẽ đạt kết quả gần các mô hình tinh vi hơn, qua đó xác nhận chiến lược thiết kế đặc trưng đúng hướng. Ưu điểm của kNN là không giả định hình dạng ranh giới, nhược điểm là suy luận chậm khi dữ liệu lớn và rất nhạy với tỷ lệ thang đo giữa các chiều.

### Triển khai kNN

Trong pipeline, k nhỏ (vài láng giềng) với trọng số giảm theo khoảng cách để giảm ảnh hưởng điểm xa, sau đó toàn bộ vector đặc trưng được chuẩn hóa bằng StandardScaler (bắt buộc để OCR/ORB không lép vế trước SSIM). Thuật toán được đánh giá bằng GroupKFold như các mô hình khác, độ chính xác tương đương XGBoost nhưng độ lệch chuẩn cao hơn cho thấy nhạy cảm với phân bố từng fold. Mặc dù không chọn để triển khai cuối (vì chi phí suy luận tuyến tính theo số mẫu), kết quả của kNN dùng làm kiểm tra chéo, nếu kNN tụt mạnh trong khi mô hình khác vẫn cao, có thể đang overfit kiến trúc thay vì đặc trưng.

## Thuật toán XGBoost (Extreme Gradient Boosting)

### Giới thiệu XGBoost

XGBoost được bổ sung ở giai đoạn sau nhằm tối ưu sự cân bằng giữa độ chính xác trung bình và độ ổn định giữa các fold. Cơ chế boosting tuần tự cho phép tập trung sửa các lỗi còn sót lại sau khi đã có đủ đặc trưng (đặc biệt các trường hợp biên almost với different). Ngoài ra, XGBoost cung cấp nhiều dạng độ quan trọng đặc trưng, giúp xác thực lại đóng góp tương đối của OCR và ORB so với các thành phần toàn cục.

### Triển khai XGBoost

Trong triển khai, cấu hình khởi đầu cân bằng bias–variance (độ sâu trung bình, tốc độ học vừa), regularization nhẹ (reg\_lambda giữ mặc định) để tránh overfit. Huấn luyện theo từng fold (GroupKFold) rồi tổng hợp ma trận nhầm lẫn cho ra kết quả std ổn định và cũng được dùng để kiểm tra chéo với các mô hình trước đó. Feature importance hiển thị thứ tự đóng góp (thường OCR / ORB nổi bật cùng SSIM) hỗ trợ biện minh thiết kế đặc trưng trong báo cáo. Nếu mở rộng dữ liệu tương lai, có thể hạ learning\_rate và tăng n\_estimators để cải thiện mượt mà mà không thay đổi kiến trúc.

## Các đặc trưng xử lý ảnh

### SSIM (Structural Similarity Index Measurement)

#### Giới thiệu SSIM

Đặc trưng SSIM phản ánh mức tương đồng cấu trúc sáng và tương phản giữa hai ảnh, thay thế so sánh pixel thô vốn nhạy với dịch chuyển nhỏ hoặc nhiễu. Nó giúp ghi nhận chênh lệch lớn về bố cục tổng thể, cắt xén vùng lớn hay thay đổi nền rõ rệt mà vẫn giữ chi phí tính toán thấp. Lý do sử dụng vì nhanh, ổn định và phù hợp vai trò trục “nền” để các đặc trưng khác bổ sung phần thông tin bị bỏ sót như màu, chữ hay điểm cục bộ. SSIM được sử dụng vì tốc độ thực thi nhanh, cảm nhận thị giác gần giống với mắt thường khi chúng ta nhìn và so sánh hình ảnh.

#### Triển khai SSIM

Hai ảnh được chuyển grayscale, giảm kích thước theo tỉ lệ nếu vượt ngưỡng pixel 700k rồi median blur nhẹ để triệt nhiễu muối tiêu mà không làm mờ cấu trúc lớn. Hàm ssim(grayscale1, grayscale2, data\_range=255) trả về giá trị trong khoảng gần 0–1 (càng cao càng giống). Giá trị được đưa thẳng vào vector đặc trưng mà không chuẩn hóa riêng vì phân bố đã hẹp. Ngoại lệ (ảnh lỗi) gán giá trị trung tính 0.5 để tránh phá vỡ cân bằng huấn luyện.

### OCR (Optical Character Recognition)

#### Giới thiệu OCR

Thành phần OCR nắm thông tin nội dung chữ và là lớp rủi ro nghiệp vụ (giá, khuyến mãi) mà các phép đo thuần thị giác thường bỏ sót vì hình dạng tổng thể không đổi. Trong nhiều trường hợp banner bị sửa con số hoặc thay cụm chữ ngắn nhưng nền và icon giữ nguyên, SSIM, Histogram và Layout đều cao, bổ sung điểm tương đồng văn bản làm giảm false negative với vi phạm nội dung.

#### Triển khai OCR

Sau khi chuyển grayscale và median blur, áp dụng adaptive threshold (block 21, C=8) để tăng tương phản ký tự. Tesseract (--oem 3 --psm 6) trích text, kết quả làm sạch bằng regex giữ chữ và số, tách thành danh sách từ thường. Hai thước đo được tính là độ tương đồng chuỗi toàn cục (SequenceMatcher) phản ánh trật tự và giao hội trên tối đa (Jaccard) phản ánh phủ nội dung. Điểm cuối = 0.7*chuỗi + 0.3*Jaccard, kẹp trong [0.1,0.95], trường hợp cả hai ít từ (<3) trả giá trị trung lập 0.8, một bên rỗng 0.3. Cache theo đường dẫn ảnh tránh tính lại nhiều lần.

### Layout

#### Giới thiệu Layout

Đặc trưng Layout tập trung vào hình dạng tổng thể và vị trí khối nội dung. Khi một banner bị cắt bớt, chèn thêm panel, dịch chuyển cụm chữ lớn hoặc đảo bố cục, sự trùng khớp biên giảm mạnh dù màu hoặc văn bản còn giống. Nó giúp tách những thay đổi cấu trúc khỏi thay đổi nhỏ về màu hay ký tự, giảm phụ thuộc đơn thuần vào SSIM vốn có thể giữ cao khi dịch chuyển nhẹ kèm nền đồng đều.

#### Triển khai Layout

Hai ảnh (grayscale đã tiền xử lý) được trích biên bằng Canny (ngưỡng 50–150). Kết quả chuyển thành mặt nạ nhị phân, tính intersection và union để lấy IoU = inter/union. Trường hợp union = 0 (hiếm, ảnh trắng) trả về 1.0. Giá trị càng thấp gợi ý thay đổi bố cục đáng kể. Đặc trưng này không cần chuẩn hóa bổ sung và đưa trực tiếp vào vector. Nhiễu chữ dày có thể làm dày edge và median blur nhẹ trước Canny giúp giảm biên giả.

### Histogram

#### Giới thiệu Histogram

Histogram màu biểu diễn phân bố cường độ của từng kênh RGB, giúp phát hiện thay đổi bảng màu nền, pha tông hoặc can thiệp chỉnh màu tổng thể khi cấu trúc hình học vẫn gần như giữ nguyên (khi đó SSIM còn cao). Việc thêm đặc trưng này giảm rủi ro bỏ sót trường hợp chỉnh tông thương hiệu, đảo màu hoặc thay background phẳng. Nó đóng vai trò mở rộng chiều “ngữ cảnh thị giác toàn cục” mà edge hay keypoint không thể hiện.

#### Triển khai Histogram

Đọc ảnh màu gốc (không ép grayscale), tùy chọn co lại nhẹ với ảnh rất lớn để giảm thời gian. Với mỗi kênh R, G, B tính histogram 32 bins, chuẩn hóa L1 để loại bỏ ảnh hưởng độ sáng tuyệt đối và đo tương tự bằng hệ số tương quan (CORREL) rồi ánh xạ (corr+1)/2 về [0,1]. Điểm cuối là trung bình ba kênh. Các lỗi đọc ảnh được gán 0.5. Không chia ô không gian (spatial bins) để giữ mô hình nhẹ và phần cấu trúc đã do SSIM/Layout gánh.

### ORB (Oriented FAST and Rotated BRIEF)

#### Giới thiệu ORB

Đặc trưng ORB bắt thay đổi cục bộ tinh vi như thêm hoặc xóa icon nhỏ, dịch chuyển logo, chỉnh nút CTA, thay mảng chi tiết nhỏ mà histogram và SSIM không nhạy. Khi phần lớn ảnh giữ nguyên nhưng một đối tượng thương hiệu bị thay, số và chất lượng cặp điểm đặc trưng trùng khớp sẽ giảm rõ, giúp mô hình đẩy mẫu vào almost hoặc different đúng mức.

#### Triển khai ORB

Ảnh grayscale tiền xử lý được đưa vào ORB (nfeatures=150) để phát hiện và mô tả điểm bằng binary descriptor. Dùng BFMatcher Hamming và KNN (k=2) áp dụng Lowe ratio 0.75 lọc match kém. Tính ratio = số match tốt / min(len(kp1), len(kp2)) và quality = 1 − mean(distance)/256, sau đó điểm ORB = 0.5*ratio + 0.5*quality (clip [0,1]). Nếu không có descriptor hoặc không match hợp lệ trả 0.5. Số nfeatures cố ý vừa phải cân bằng độ nhạy và thời gian và việc tăng cao hơn cho dữ liệu hiện tại ít cải thiện do nhiều vùng phẳng.

# HỆ THỐNG KIỂM THỬ GIAO DIỆN TỰ ĐỘNG

## Giới thiệu hệ thống

Hệ thống được xây dựng với mục đích bài toán so khớp ảnh quảng cáo gốc với biến thể từ các chi nhánh để phát hiện sai lệch (màu, bố cục, văn bản, chi tiết cục bộ…)

## Kiến trúc hệ thống

A diagram of a software development

AI-generated content may be incorrect.

Hình ‑. Kiến trúc hệ thống

### Các thành phần của hệ thống

- Input: Cặp (original.png, variant.png). Mỗi original\_x có nhiều biến thể với mục tiêu là so sánh một – một theo cặp.

- Loader & Group: Đọc thư mục original\_image và variant\_image/original\_x. Sau đó tạo danh sách theo đường dẫn dựa vào tên thư mục (path\_original, path\_variant, label). Cuối cùng là gán group\_id = số x để dùng GroupKFold.

- Preprocessing: Đọc grayscale. Downscale nếu tổng pixel vượt ngưỡng 700k, medianBlur(3) giảm nhiễu nhẹ.

- Feature Extraction: Mỗi đặc trưng chuẩn hóa về thang [0,1] để dễ học:

+ SSIM: Đánh giá cấu trúc và độ tương đồng cường độ sau đó đưa đầu ra gần bằng [0,1]

+ Histogram: So sánh phân bố màu (3 kênh, 32 bins) và dùng hệ số tương quan CORREL rồi map (corr+1)/2.

+ Layout: Canny edges để có thể IoU giữa hai mặt nạ cạnh và phản ánh thay đổi bố cục lớn

+ OCR: Adaptive threshold sau đó Tesseract và đưa ra chuỗi token. Kết hợp 0.7 \* SequenceMatcher + 0.3 \* Jaccard và xử lý trường hợp text quá ít

+ ORB Keypoints: Phát hiện và mô tả điểm đặc trưng, lọc Lowe ratio, tính score = trung bình giữa mật độ match tương đối và chất lượng (1 - mean(distance)/256)

- 5D Vector: Ghép các đặc trưng SSIM, OCR, Hist, ORB, Layout để tiến hành quá trình huẩn luyện và kiểm tra

- Models:

+ SVM: Kernel linear cần StandardScaler

+ kNN: k=5, weights=distance vì nhạy scale nên dùng scaler

+ Random Forest: 100 cây, depth giới hạn và không cần scaler, cung cấp feature importance

+ XGBoost: khai thác quan hệ phi tuyến và xử lý imbalance qua scale\_pos\_weight

- Output: nhãn dự đoán similar / almost / different

- Evaluation:

+ GroupKFold (k=5) với mỗi original\_x chỉ nằm trong 1 fold test để loại bỏ leakage

+ Thu thập các giá trị mean accuracy ± std và confusion matrix, tầm quan trọng của từng feature với mục đích so sánh độ ổn định std. Sau đó có thể đưa ra quyết định có cần cải tiến mô hình nữa hay không.

### Luồng hoạt động của hệ thống

Hệ thống bắt đầu từ các cặp Input, được chuyển vào Loader & Group để liệt kê các trường hợp và nhóm theo group\_ids. Sau đó dữ liệu được đưa qua bước Preprocessing gồm chuyển sang ảnh xám, thay đổi kích thước và giảm nhiễu. Tiếp theo, 5 đặc trưng được tính toán song song, sau đó hợp nhất lại thành 1 vector 5 chiều. Quá trình huẩn luyện và đánh giá sử dụng GroupKFold, chia dữ liệu thành nhiều fold dựa trên nhóm. Với mỗi fold, mô hình được huẩn luyện trên tập train, sau đó dự đoán trên tập test và ghi lại output cùng các chỉ số đánh giá. Từ kết quả từng fold, hệ thống gom thống kê và tiến hành đánh giá tổng hợp, bao gồm trung bình và độ lệch chuẩn, ma trận nhầm lẫn và tầm quan trọng của từng đặc trưng. Cuối cùng dựa trên kết quả tổng hợp này, em sẽ đưa ra quyết định mô hình và thực hiện các điều chỉnh cần thiết.

## Các giai đoạn phát triển hệ thống

### Giai đoạn 1

#### Mục tiêu và bối cảnh

Giai đoạn đầu tiên được thiết kế như một phép thử nền để xác nhận toàn bộ chuỗi thao tác tải cặp ảnh, gán nhãn, tạo nhóm và chạy được một vòng huấn luyện – đánh giá trước khi mở rộng. Chỉ có hai lớp similar và different và chỉ dùng một đặc trưng duy nhất là SSIM vì đặc trưng này dễ sử dụng, phản ánh mức tương đồng cấu trúc toàn cục, giúp phát hiện các khác biệt lớn và đủ để đo xem dữ liệu có tín hiệu phân tách sơ bộ hay không. Với mục tiêu không nhằm tối đa hóa độ chính xác tuyệt đối mà nhằm kiểm chứng tính đúng đắn của cơ chế group, phát hiện sớm nguy cơ data leakage và định hình baseline để so sánh các cải tiến sau.

#### Dữ liệu và đánh giá

Bộ dữ liệu tổ chức theo thư mục, mỗi original\_x (một ảnh gốc) đi kèm một thư mục con chứa nhiều biến thể với các kiểu thay đổi, ở giai đoạn này các tên biến thể được ánh xạ gộp về hai nhãn similar hoặc different tùy mức sai lệch tổng thể so với ảnh gốc. Mỗi cặp (đường dẫn original, đường dẫn variant, nhãn) được gắn group\_id chính là chỉ số x của ảnh gốc để bảo toàn tính ràng buộc nhóm. Rủi ro căn bản nếu chia ngẫu nhiên là nhiều biến thể của cùng original\_x rơi đồng thời vào tập huấn luyện và kiểm tra khiến mô hình nhìn thấy trước đặc trưng nền tảng của ảnh đó và kết quả bị thổi phồng không phản ánh khả năng khái quát sang một ảnh gốc mới. Vì lý do đó chiến lược đánh giá chuẩn ngay từ thời điểm này sử dụng GroupKFold với số lần gập cố định, đảm bảo toàn bộ biến thể của một gốc chỉ xuất hiện trong đúng một fold kiểm tra ở mỗi vòng xoay, mô phỏng tình huống triển khai thực tế phải xử lý một nguồn ảnh chưa từng gặp. Các thước đo được thu thập gồm độ chính xác tổng và ma trận nhầm lẫn hai lớp với trọng tâm đặc biệt vào khả năng phát hiện đúng lớp different do bỏ sót vi phạm có chi phí nghiệp vụ cao hơn nhiều so với cảnh báo nhầm. Các giá trị định lượng cụ thể được trình bày ở mục kết quả riêng, tại đây chỉ xác lập phương pháp và lý do lựa chọn.

#### Quy trình kỹ thuật và chọn mô hình

Mỗi cặp ảnh đi qua chuỗi tiền xử lý thống nhất, đọc kênh xám, nếu tổng số pixel vượt ngưỡng kinh nghiệm trong khoảng 700k pixel thì thu nhỏ đẳng tỷ lệ để kiềm chế chi phí và tránh méo hình, sau đó áp dụng median blur nhẹ nhằm triệt nhiễu muối tiêu mà vẫn giữ biên chính. Chỉ số SSIM được tính trực tiếp trên hai ảnh xám đã chuẩn hóa, giá trị vốn ở miền gần [0,1] nên không yêu cầu thêm bước scaling riêng. Bộ phân loại lựa chọn là SVM tuyến tính với cơ chế cân bằng trọng số lớp để tránh thiên lệch do phân bố nhãn và giảm số siêu tham số cần kiểm soát; chưa tiến hành tinh chỉnh mà giữ cấu hình mặc định để mọi thay đổi hiệu năng tương lai có thể quy trực tiếp cho không gian đặc trưng mở rộng.

#### Hạn chế và ý tưởng cải tiến

Quan sát định tính các lỗi cho thấy nhiều trường hợp thay đổi nhỏ về văn bản, biểu tượng hoặc điều chỉnh nhẹ tông màu vẫn bị phân loại similar vì SSIM duy trì cao, trong khi một số thay đổi trung bình không quá nghiêm trọng lại bị đẩy thẳng thành different do thiếu lớp đệm almost khiến hai cực nhãn gom quá nhiều biến thể khác nhau. Nguyên nhân kỹ thuật là SSIM chỉ đo tương quan cấu trúc và độ chói tổng thể, ít nhạy các thay đổi cục bộ tinh vi, không mã hóa phân bố màu riêng kênh và hoàn toàn bỏ qua nội dung ký tự. Hệ quả nghiệp vụ tiềm ẩn là nguy cơ bỏ sót sửa đổi văn bản nhạy cảm và thiếu cơ chế ưu tiên xử lý bán tự động do chỉ có hai mức quyết định. Kết luận giai đoạn này dẫn tới quyết định bổ sung trước hai đặc trưng Histogram và Layout ở bước kế tiếp để tăng độ nhạy với biến đổi tông màu và hình thái cạnh tổng thể, vẫn giữ nguyên số lớp nhằm cô lập lợi ích của từng đặc trưng mới trước khi nâng độ phân giải nhãn với almost và thêm kênh văn bản.

### Giai đoạn 2

#### Mục tiêu và bối cảnh

Sau khi thiết lập baseline, yêu cầu nổi bật là giảm tỷ lệ bỏ sót những thay đổi về màu nền, phối tông hoặc cắt hoặc chèn khối cấu trúc lớn mà SSIM đơn đôi khi giữ điểm cao, đồng thời vẫn trì hoãn việc phức tạp hóa không gian nhãn. Giai đoạn này tập trung mở rộng chiều biểu diễn bằng hai đặc trưng bổ sung là Histogram để mã hóa phân bố màu toàn cục và Layout dựa trên so khớp biên nhằm phản ánh biến đổi cấu trúc lớn, với kỳ vọng thu hẹp nhóm lỗi different bị gán nhầm similar do đổi palette hoặc dịch chuyển khối rõ rệt.

#### Dữ liệu và đánh giá

Bản chất tổ chức dữ liệu (cặp original\_x và nhiều biến thể) và cơ chế gán group không thay đổi, tiếp tục giữ ánh xạ nhãn về hai lớp để cô lập ảnh hưởng thuần túy của việc bổ sung đặc trưng mà không pha trộn hiệu ứng do tách lớp mới. Rủi ro leakage vẫn y hệt giai đoạn đầu nên GroupKFold tiếp tục đóng vai trò then chốt trong đánh giá; việc giữ nguyên số lớp giúp so sánh trực tiếp ma trận nhầm lẫn trước và sau thêm Histogram và Layout để xác định loại lỗi nào giảm. Các thước đo thu thập không đổi: độ chính xác tổng, cấu trúc nhầm lẫn hai chiều và quan sát định tính các mẫu biên. Các giá trị số được dời sang phần kết quả, ở đây nhấn mạnh phương pháp luận nhất quán nhằm bảo đảm tính so sánh.

#### Quy trình kỹ thuật và chọn mô hình

Chuỗi tiền xử lý ảnh xám giữ nguyên, tiếp tục thu nhỏ đẳng tỷ lệ theo ngưỡng pixel và median blur nhẹ. Đặc trưng SSIM được giữ làm trục nền, Histogram được tính trên ảnh màu gốc bằng phân phối kênh trong số bin cố định rồi chuẩn hóa để loại bỏ ảnh hưởng độ sáng tuyệt đối, sau đó chuyển hệ số tương quan về miền [0,1] và Layout được suy ra qua phát hiện biên Canny trên hai ảnh xám đã xử lý và tính IoU giữa mặt nạ biên nhằm đo mức chồng lấp cấu trúc hình dạng. Bộ phân loại tiếp tục là SVM tuyến tính với cân bằng lớp để đảm bảo thay đổi hiệu năng đến từ chiều đặc trưng mới chứ không phải thay đổi thuật toán; chưa áp dụng tinh chỉnh mở rộng vì mục tiêu vẫn là đánh giá đóng góp đặc trưng.

#### Hạn chế và ý tưởng cải tiến

Mặc dù hai đặc trưng mới giúp giảm đáng kể các trường hợp đổi tông màu hoặc dịch chuyển bố cục lớn bị xem là similar, những lỗi liên quan đến sửa chữ nhỏ, thay con số hoặc chỉnh chi tiết biểu tượng vẫn tồn tại vì hệ thống chưa có khả năng đọc văn bản hay định vị thay đổi cục bộ tinh vi. Đồng thời việc chỉ có hai lớp khiến một mảng biến thể trung gian bị dồn vào different tạo ra nhiễu cho quy trình nghiệp vụ vốn cần ưu tiên những sai lệch mạnh hơn. Điều này chỉ ra hai khoảng trống là thiếu thông tin ngữ nghĩa chữ và thiếu cấp phân loại trung gian để giảm nhầm lẫn hai cực. Quyết định cho giai đoạn kế tiếp là bổ sung OCR nhằm trích nội dung văn bản và mở rộng không gian nhãn thêm lớp almost, với giả thuyết rằng lớp chuyển tiếp sẽ giảm trao đổi sai giữa similar và different, đồng thời kênh văn bản giúp bắt các thay đổi mà SSIM, Histogram và Layout cùng bỏ sót.

### Giai đoạn 3

#### Mục tiêu và bối cảnh

Sau khi đã kiểm soát tốt hơn thay đổi màu và bố cục toàn cục, nhu cầu tiếp theo là phát hiện chính xác các chỉnh sửa nội dung chữ (giá, khuyến mãi, câu slogan) và tránh việc đẩy mọi sai lệch vừa phải lên nhóm different, gây quá tải kiểm thử thủ công. Giai đoạn này vì thế đồng thời mở rộng không gian nhãn từ hai lên ba lớp bằng cách thêm almost làm tầng đệm ngữ nghĩa mức sai lệch chấp nhận được và tích hợp đặc trưng OCR để mô hình có thông tin trực tiếp về mức tương đồng văn bản giữa hai ảnh.

#### Dữ liệu và đánh giá

Cấu trúc thư mục giữ nguyên nhưng ánh xạ nhãn thay đổi, các biến thể liên quan thay đổi cục bộ nhẹ hoặc biến đổi text hoặc layout mức nhỏ được gom vào lớp gần mới almost thay vì ép vào similar hoặc different. GroupKFold tiếp tục đảm bảo mỗi original\_x chỉ xuất hiện một lần trong tập kiểm tra ở mỗi vòng, tăng độ tin cậy khi thêm một lớp mới vốn dễ làm tăng phương sai đánh giá nếu rơi vào tình huống phân bổ không đồng đều giữa folds. Việc đo lường vẫn lấy độ chính xác tổng và ma trận nhầm lẫn, nhưng diễn giải tập trung hơn vào hai cặp nhầm lẫn then chốt là similar với almost và almost với different vì chúng đại diện đường biên cần tối ưu.

#### Quy trình kỹ thuật và chọn mô hình

Pipeline kế thừa toàn bộ bước tiền xử lý ảnh xám, tính SSIM, Histogram và Layout từ giai đoạn trước. Thành phần mới là OCR, mỗi ảnh xám sau median blur được nhị phân hóa thích nghi để tăng tương phản ký tự, đưa qua OCR để trích chuỗi ký tự, làm sạch rồi đánh giá mức tương đồng bằng kết hợp các độ đo chuỗi và tập từ nhằm cân bằng độ nhạy thứ tự và phổ từ vựng, kết quả nén về một điểm nằm trong khoảng giá trị ổn định tránh cực trị khi thiếu văn bản. Vector đặc trưng lúc này có thêm chiều OCR, được chuẩn hóa đồng nhất qua scaler trước khi huấn luyện SVM (tiếp tục dùng dạng tuyến tính hoặc RBF nhẹ nếu xuất hiện tương tác phi tuyến giữa chiều nội dung và màu / cấu trúc, nhưng vẫn giữ tư tưởng hạn chế tuning). Việc thêm lớp almost buộc mô hình học ranh giới mềm giữa tương đồng hoàn toàn và khác biệt rõ, tạo tiền đề cho đánh giá chi tiết hơn ở các giai đoạn sau khi thêm đặc trưng cục bộ.

#### Hạn chế và ý tưởng cải tiến

Mặc dù việc bổ sung OCR và lớp almost giảm nhiều nhầm lẫn trực tiếp giữa similar và different, vẫn còn tỷ lệ đáng kể các trường hợp thay đổi rất nhỏ về biểu tượng, logo hoặc chi tiết đồ họa cục bộ không tác động đáng kể lên Histogram, Layout hay OCR, khiến mô hình đôi khi giữ chúng ở similar hoặc almost thay vì đẩy lên mức cảnh báo. Nguyên nhân là thiếu một đặc trưng nhạy với cấu trúc vi mô và quan hệ điểm đặc trưng giữa hai ảnh, các đặc trưng hiện tại quá thiên về thống kê toàn cục. Do đó quyết định kế tiếp là tích hợp ORB để khai thác mô tả keypoint và phép so khớp cục bộ, với kỳ vọng tăng recall của lớp different trên các thay đổi nhỏ nhưng quan trọng mà ba chiều toàn cục và chiều nội dung chưa nắm bắt.

### Giai đoạn 4

#### Mục tiêu và bối cảnh

Sau khi đã có trục cấu trúc (SSIM, Layout), màu (Histogram) và nội dung chữ (OCR), khoảng trống còn lại nằm ở khả năng phát hiện các chỉnh sửa cục bộ tinh vi là thêm / xóa icon, dịch chuyển logo nhỏ, thay nút hành động. Giai đoạn này đưa vào đặc trưng ORB để bổ sung lớp nhìn cấp điểm đặc trưng, hoàn thiện vector năm chiều nhằm tăng độ nhạy với thay đổi vi mô và cải thiện khả năng phân tách giữa almost và different nơi ranh giới trước đó còn mềm.

#### Dữ liệu và đánh giá

Ánh xạ nhãn ba lớp giữ nguyên, phân bố expected giữa similar, almost và different được theo dõi qua biểu đồ để xác nhận không có lệch quá mạnh gây suy giảm độ tin cậy của kết quả cross-validation. GroupKFold tiếp tục đảm bảo đánh giá công bằng theo nhóm và cho phép so sánh xuyên giai đoạn vì điều kiện chia nhóm không đổi. Việc quan sát ma trận nhầm lẫn tập trung vào hai loại cải thiện mong đợi đó là tăng bắt đúng different khi thay đổi cục bộ nhỏ và không làm trầm trọng thêm sự hoán đổi giữa similar và almost. Các con số chi tiết vẫn hoãn đến phần kết quả, mô tả ở đây nhấn mạnh duy trì nhất quán phương pháp đánh giá trong khi thêm chiều đặc trưng mới.

#### Quy trình kỹ thuật và chọn mô hình

Pipeline kế thừa đầy đủ tiền xử lý và bốn đặc trưng trước. Thành phần ORB được tính trên hai ảnh xám đã chuẩn hóa bằng cách phát hiện tập điểm đặc trưng với số lượng tối ưu thực nghiệm cân bằng chi phí và độ phủ, trích descriptor nhị phân, thực hiện so khớp KNN với ngưỡng ratio để loại bỏ match kém chất lượng, sau đó tổng hợp mật độ match tương đối và chất lượng trung bình thành một điểm chuẩn hóa. Giá trị fallback trung tính được dùng khi không có đủ descriptor hợp lệ để tránh đưa tín hiệu nhiễu cực đoan. Vector năm chiều sau đó được chuẩn hóa (đối với mô hình cần) và huấn luyện qua SVM, lúc này có thể cân nhắc kernel phi tuyến nếu quan sát thấy tương tác giữa đặc trưng cục bộ và nội dung chữ cải thiện ranh giới. Tuning sâu vẫn hạn chế nhằm bảo toàn khả năng quy kết cải tiến cho đặc trưng mới.

#### Hạn chế và ý tưởng cải tiến

Sau khi tích hợp ORB, phần lớn lỗi bỏ sót thay đổi cục bộ giảm rõ, nhưng lúc này câu hỏi chuyển trọng tâm từ mở rộng biểu diễn sang tối ưu lựa chọn mô hình theo tiêu chí ổn định giữa các folds, khả năng diễn giải và chi phí suy luận. SVM cho kết quả nhất quán song vẫn có dao động nhất định ở một vài nhóm và chưa cho góc nhìn trực tiếp về độ quan trọng tương đối của từng đặc trưng như mô hình cây hay sự nhạy khoảng cách như kNN. Việc tiếp tục bổ sung thêm đặc trưng trở nên ít hiệu quả biên, do đó quyết định bước tiếp là giữ nguyên vector năm chiều và mở sang giai đoạn so sánh đa mô hình (SVM, kNN, Random Forest, XGBoost) để chọn ứng viên triển khai cuối cùng dựa trên sự cân bằng giữa mean performance, độ lệch chuẩn, khả năng giải thích và tốc độ dự đoán.

### Giai đoạn 5

#### Mục tiêu và bối cảnh

Với không gian đặc trưng đã được củng cố đủ chiều toàn cục và cục bộ, nhu cầu cuối cùng là đánh giá khách quan các thuật toán phân loại khác nhau trên cùng một vector nhằm xác định mô hình triển khai tối ưu và kiểm chứng rằng hiệu năng thu được không phải sản phẩm ngẫu nhiên của một kiến trúc đơn lẻ. Giai đoạn này tiến hành huấn luyện và đánh giá song song các mô hình đại diện cho nhiều họ: SVM (margin-based), kNN (phi tham số đo khoảng cách), Random Forest (tập hợp cây bagging) và XGBoost (boosting tuần tự).

#### Dữ liệu và đánh giá

Tập dữ liệu và ánh xạ ba lớp giữ nguyên hoàn toàn so với giai đoạn ngay trước, đảm bảo mọi khác biệt hiệu năng phản ánh thuần túy lựa chọn thuật toán. GroupKFold tiếp tục là khung đánh giá duy nhất để loại trừ leakage và cho phép so sánh độ ổn định bằng cách đối chiếu độ dao động giữa các folds cho từng mô hình. Các thước đo sẽ gồm độ chính xác trung bình, ma trận nhầm lẫn gộp và (đối với mô hình cây / boosting) phân tích tầm quan trọng đặc trưng, kNN bổ sung góc nhìn về sự phân tách tự nhiên của không gian vì nó không học tham số toàn cục. Các trị số cụ thể được dành cho phần kết quả, ở đây chỉ mô tả thiết kế thử nghiệm đồng nhất giữa các thuật toán để đảm bảo tính công bằng.

#### Quy trình kỹ thuật và chọn mô hình

Tầng trích đặc trưng giữ nguyên năm chiều. SVM tiếp tục sử dụng chuẩn hóa với StandardScaler để đảm bảo các chiều có cùng thang đo. kNN cũng áp dụng scaler vì nhạy mạnh với scale. Random Forest và XGBoost sử dụng trực tiếp đặc trưng thô vì cây quyết định theo ngưỡng và boosting xử lý chuẩn hóa nội bộ không bắt buộc. Mỗi mô hình được huấn luyện độc lập trên từng fold, lưu lại dự đoán để xây dựng ma trận nhầm lẫn hợp nhất, quá trình tinh chỉnh được thực hiện ở mức lưới nhỏ có kiểm soát nhằm tránh tối ưu quá sâu trên tập giới hạn và vẫn ưu tiên so sánh tương đối giữa họ thuật toán. Feature importance từ RF và XGBoost cung cấp bằng chứng bổ sung về đóng góp cân bằng giữa OCR, ORB và các trục toàn cục (SSIM, Histogram, Layout), trong khi kNN đóng vai trò kiểm tra lại rằng cấu trúc khoảng cách sau chuẩn hóa không gây lệch chiều.

#### Hạn chế và ý tưởng cải tiến

Kết quả so sánh cho thấy mỗi mô hình mang lại điểm mạnh riêng đó là kNN minh họa tính phân tách của không gian nhưng phương sai cao hơn ở một vài nhóm và suy luận tốn chi phí tuyến tính theo số mẫu, Random Forest cung cấp độ quan trọng đặc trưng trực quan nhưng đôi khi kém ổn định hơn boosting ở ranh giới almost – different, SVM giữ độ gọn nhẹ và ổn định tốt, XGBoost thường đạt sự cân bằng giữa độ chính xác trung bình và độ lệch chuẩn thấp nhất, đồng thời vẫn cung cấp chỉ số tầm quan trọng đa chiều. Giới hạn còn lại chủ yếu nằm ở khả năng mở rộng khi dữ liệu tăng mạnh (kNN suy luận chậm, boosting cần điều chỉnh tham số) và thiếu cơ chế giải thích cục bộ từng dự đoán. Quyết định tổng kết là chọn một mô hình trọng tâm (ví dụ XGBoost hoặc SVM tùy ưu tiên triển khai giữa ổn định và chi phí) kèm một mô hình cây làm tham chiếu giải thích và duy trì kNN như baseline giám sát drift, các cải tiến tương lai hướng đến localization vùng khác biệt và mở rộng dữ liệu để cân nhắc đặc trưng học sâu nhẹ nếu quy mô thay đổi đáng kể.

## Phân tích hệ thống

Use case => đặc tả (nghiệp vụ trọng yếu) => kết hợp vẽ sequence

## Thiết kế hệ thống

### Thiết kế cơ sở dữ liệu

Lược đồ CSDL

=> giải thích vai trò từng bảng, và giải thích các mối quan hệ từng bảng.

### Thiết kế giao diện

### ...

## Kết quả đạt được

A blue rectangular bar graph

AI-generated content may be incorrect.

Hình ‑2. Biểu đồ phân bố nhãn giai đoạn 1 – 2

A bar graph with blue squares

AI-generated content may be incorrect.

Hình ‑3. Biểu đồ phân bố nhãn giai đoạn 3 – 4 – 5

### Giai đoạn 1

A graph with a line

AI-generated content may be incorrect.

Hình ‑4. Độ chính xác Cross-validation theo từng Fold (Version 1 - SVM)

A graph with blue squares

AI-generated content may be incorrect.

Hình ‑. Ma trận nhầm lẫn (Version 1 - SVM)

Biểu đồ phân bố nhãn cho thấy hai lớp similar và different lệch (different chiếm ưu thế), vì vậy cần ưu tiên xem recall lớp different thay vì chỉ nhìn accuracy. Accuracy 5 fold dao động gần bằng 91%–94%, trung bình 92.3% ±0.8%, cho thấy mô hình một chiều (chỉ SSIM) chưa ổn định theo group. Ma trận nhầm lẫn cho thấy lỗi chính là different bị dự đoán thành similar ở các thay đổi chữ nhỏ hoặc chi tiết cục bộ làm SSIM vẫn cao. Chiều ngược lại (similar sang different) hầu như không xuất hiện nên precision lớp different đạt 100%. Kết quả per-class của different là Precision 100.00%, Recall 90.42%, F1 94.97%. Đây là baseline để so sánh tác động của việc bổ sung đặc trưng.

### Giai đoạn 2

A graph with a blue line

AI-generated content may be incorrect.

Hình ‑5. Độ chính xác Cross-validation theo từng Fold (Version 2 - SVM)

A graph with blue squares

AI-generated content may be incorrect.

Hình ‑. Ma trận nhầm lẫn (Version 2 - SVM)

Thêm Histogram (màu) và Layout (cấu trúc) giúp accuracy tăng lên 970% ±1.6%, giảm phương sai giữa các group. Lớp different cải thiện rõ, Precision 100%, Recall 96.25%, F1 98.09% (FN giảm mạnh so với 90.42% recall ở Giai đoạn 1). Nguyên nhân do thay đổi palette và bố cục lớn nay được tách dễ hơn thay vì bị “ẩn” sau SSIM cao. Tuy vậy vẫn còn cụm mẫu thay đổi rất nhỏ về text/cục bộ bị đẩy thẳng vào different (mức độ nghiêm trọng không đồng nhất), tạo áp lực cần một lớp trung gian. Điều này củng cố quyết định mở rộng nhãn (thêm almost) ở giai đoạn tiếp theo.

### Giai đoạn 3

A graph with a line

AI-generated content may be incorrect.

Hình ‑6. Độ chính xác Cross-validation theo từng Fold (Version 3 - SVM)

A graph with blue squares and white text

AI-generated content may be incorrect.

Hình ‑. Ma trận nhầm lẫn (Version 3 - SVM)

Mở rộng thành ba lớp (similar / almost / different) và thêm OCR. Phân bố nhãn là Similar 20% (60), Almost 40% (120), Different 40% (120) đúng mục tiêu “đệm” lớp almost. Accuracy đa lớp đạt 90.33% (giảm tương đối nhỏ so với 2 lớp do nhiệm vụ khó hơn). Kết quả trung bình của từng chỉ số là similar có P 98.36%, R 100%, F1 99.17% (không có mẫu similar bị nhầm sang almost/different). Almost có P 89.57%, R 85.83%, F1 87.66%. Different: P 87.10%, R 90%, F1 88.52%. Lỗi chủ yếu dồn vào việc phân ranh almost và different đồng thời almost hấp thụ phần lớn nhầm lẫn trước đây giữa similar và different (không có chuyển thẳng similar sang different vì recall similar = 100%). OCR giúp kéo các thay đổi chữ ra khỏi similar (precision similar vẫn cao 98.36% trong khi recall = 100%). Hạn chế còn lại đó là các thay đổi cục bộ rất nhỏ (icon/logo) chưa đủ tín hiệu trong SSIM/Histogram/Layout/OCR tạo động lực bổ sung đặc trưng keypoints ở giai đoạn 4.

### Giai đoạn 4

A graph with a line and a blue dot

AI-generated content may be incorrect.

Hình ‑7. Độ chính xác Cross-validation theo từng Fold (Version 4 - SVM)

A graph with blue squares and white text

AI-generated content may be incorrect.

Hình ‑. Ma trận nhầm lẫn (Version 4 - SVM)

Bổ sung ORB (đặc trưng cục bộ) giúp cải thiện toàn diện, accuracy tăng lên 93.67% (tăng +3.34% so với 90.33%). Lớp different cải thiện recall từ 90.00% lên 93.33% (+3.33%), F1 từ 88.52% lên 92.56%. Lớp almost cũng tăng cả precision (89.57% lên 93.16%) và recall (85.83% lên 90.83%), chứng tỏ ORB giúp tách rõ các trường hợp sai lệch cục bộ nhỏ thay vì “rơi” chung vào almost hoặc bị nhận nhầm giữa almost/different. Similar vẫn giữ mức an toàn, Precision 98.36%, Recall 100%. Phần lỗi còn lại tập trung ở mẫu OCR suy giảm (chữ mờ / nền nhiễu) làm tín hiệu văn bản không ổn định. Sau bước này không còn thiếu rõ rệt về loại đặc trưng, trọng tâm chuyển sang so sánh và lựa chọn thuật toán.

### Giai đoạn 5

Trên vector 5 đặc trưng (SSIM, OCR, Histogram, ORB, Layout):

* SVM (linear, C=5.0, gamma=scale): Accuracy 93.67%, Different cho ra P 91.80%, R 93.33% và F1 92.56%
* XGBoost: Accuracy 93.67%, Different cho ra P 91.06%, R 93.33% và F1 92.18%.
* Random Forest: Accuracy 93.33%, Different cho ra P 90.98%, R 92.50% và F1 91.74%
* kNN: Accuracy 91.00%, Different cho ra P 90.60%, R 88.33% và F1 89.45%

SVM và XGBoost đạt cùng accuracy cao nhất (≈93.67%) và recall lớp different 93.33% (cao hơn RF và kNN). XGBoost/RF đạt precision similar tuyệt đối (100%) nhưng SVM vẫn giữ mức rất cao (98.36%) đồng thời có F1 lớp almost và different cân bằng, tránh overfit cục bộ. kNN kém hơn về ổn định và suy luận tốn kém tuyến tính theo số mẫu cho thấy không phù hợp để triển khai chính. Tầm quan trọng đặc trưng (RF/XGBoost) xác nhận OCR và ORB đóng góp cao, phù hợp mục tiêu phát hiện thay đổi chữ và chi tiết nhỏ.

#### RF (Random Forest)

A graph with a line and a blue dot

AI-generated content may be incorrect.

Hình ‑. Độ chính xác Cross-validation theo từng Fold (RF)

A graph with blue squares and white text

AI-generated content may be incorrect.

Hình ‑. Ma trận nhầm lẫn (Version 5 - RF)

#### kNN (k-Nearest Neighbors)

A graph with a line

AI-generated content may be incorrect.

Hình ‑. Độ chính xác Cross-validation theo từng Fold (kNN)

A graph with blue squares and white text

AI-generated content may be incorrect.

Hình ‑. Ma trận nhầm lẫn (Version 5 - kNN)

#### XGBoost (Extreme Gradient Boosting)

A graph with a line and a blue dot

AI-generated content may be incorrect.

Hình ‑. Độ chính xác Cross-validation theo từng Fold (XGBoost)

A graph with blue squares and white text

AI-generated content may be incorrect.

Hình ‑. Ma trận nhầm lẫn (Version 5 - XGBoost)

# KẾT LUẬN VÀ HƯƠNG PHÁT TRIỂN

## Kết luận

=> đạt được kết quả

=> giải quyết gì

=> ưu và khuyêt điểm

## Hướng phát triển

=> vấn đề gì, phương pháp => business, giải pháp, kiến trúc...

TÀI LIỆU THAM KHẢO

(Theo chuẩn IEEE – *bỏ dòng này khi viết báo cáo*)

<https://www.researchgate.net/publication/394449177_Challenges_in_GUI_Test_Automation_for_Dynamic_Web_Applications_A_Systematic_Review>

<https://www.researchgate.net/publication/369365215_Image_quality_assessment_based_on_the_perceived_structural_similarity_index_of_an_image>

<https://www.researchgate.net/publication/391528309_Enhancing_Bounding_Box_Regression_for_Object_Detection_Dimensional_Angle_Precision_IoU-Loss>

<https://www.researchgate.net/publication/383459245_Image_Retrieval_Using_Multilayer_Feature_Aggregation_Histogram>

<https://www.researchgate.net/publication/368189389_A_Comparative_Analysis_of_SIFT_SURF_and_ORB_on_Sketch_and_Paint_based_images>

<https://www.researchgate.net/publication/373513855_Advancements_in_OCR_A_Deep_Learning_Algorithm_for_Enhanced_Text_Recognition>

<https://www.researchgate.net/publication/381131594_Scikit-ANFIS_A_Scikit-Learn_Compatible_Python_Implementation_for_Adaptive_Neuro-Fuzzy_Inference_System>

<https://www.researchgate.net/publication/394048129_Comparative_Performance_Analysis_of_Decision_Tree_And_SVM_Algorithms_in_Detecting_Multiple_System_Atrophy_Based_on_Clinical_Features>

<https://www.researchgate.net/publication/372479561_A_Scalable_Tree_Boosting_System_XG_Boost>

PHỤ LỤC

* Bổ sung bộ testcase
* So sánh các mô hình