**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: 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 trức 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

Xây dựng hệ thống kiểm thử giao diện tự động phát hiện và phân loại mức độ sai lệch quảng cáo hình ảnh giữa phiên bản gốc và phiên bản triển khai tại các website chi nhánh.

## 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:

* Version 1: SSIM đơn đặc trưng với nhãn 2 lớp (similar/different).
* Version 2: Thêm Histogram (phân bố màu) + Layout để giảm phụ thuộc SSIM.
* Version 3: Thêm OCR (tỷ lệ tương đồng từ khóa), mở rộng thành nhãn 3 lớp (similar/almost/different) để tách sai lệch nhẹ.
* Version 4: 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: 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: 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

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: 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: gặp chi nhánh mới hoặc biến thể chưa thấy. Cách dùng: tạo mảng groups (cùng chiều dữ liệu), truyền vào gkf.split(X, y, groups).

## Tiền xử lý ảnh

Ảnh được chuyển grayscale, giảm kích thước theo tỷ lệ nếu vượt ngưỡng diện tích (600k–700k pixel) tránh resize cứng và lọc median nhẹ để giảm nhiễu mà không mờ cạnh quan trọng. Histogram giữ nguyên ảnh màu; OCR bổ sung bước nhị phân hóa thích nghi; các đặc trưng còn lại dùng cùng phiên bản ảnh xám đã chuẩn hóa.

## 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 RBF giúp nắm được quan hệ phi tuyến nhẹ giữa các tín hiệu (ví dụ trường hợp SSIM cao nhưng OCR thấp hoặc ORB giảm). 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.

### Triển khai SVM

Trong triển khai, phiên bản nhị phân ban đầu dùng SVM linear làm baseline (Version 1, Version 2); khi mở rộng lên 3 lớp và đủ đặc trưng chuyển sang RBF (C≈2–5, gamma mặc định scale). Quy trình: 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; nếu thấy almost bị kéo về similar nhiều có thể cân nhắc điều chỉnh C hoặc bổ sung ORB.

## 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, mô hình dùng 100 cây, độ sâu giới hạn (depth≈10) để tránh học quá kỹ từng nhóm nhỏ. Không cần chuẩn hóa vì cây 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=5 với weights='distance' giảm tác động của hàng xóm xa; 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 (n\_estimators=100, max\_depth=6, learning\_rate=0.1, subsample=0.8, colsample\_bytree=0.8) cân bằng bias–variance; không dùng scaler. Việc thêm regularization nhẹ (reg\_lambda giữ mặc định) đủ để tránh phình cây trên tập vừa. Huấn luyện theo từng fold (GroupKFold) rồi tổng hợp ma trận nhầm lẫn; kết quả std thấp nhất trong các mô hình có thể được chọn làm ứng viên triển khai. 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 dùng: nhanh, ổn định hơn MSE / diff; 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 (≈600k–700k) rồi median blur nhẹ (kernel 3) 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 – 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: (a) độ tương đồng chuỗi toàn cục (SequenceMatcher) phản ánh trật tự, (b) 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; 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; 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; 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: 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; 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 thiếu 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 (600k–700k tùy phiên bản), 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); 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 & 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/rbf 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.

## 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

chức năng quan trọng => hình kết quả => diễn giải sử dụng và kết quả...

# 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*)

PHỤ LỤC