

TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT TP. HỒ CHÍ MINH
KHOA CÔNG NGHỆ THÔNG TIN



ĐỀ TÀI:

**ÁP DỤNG DCA, SIFT, LBPH TRONG CÁC BÀI TOÁN
XỬ LÝ ẢNH SỐ**

ĐỒ ÁN MÔN HỌC
MÔN XỬ LÝ ẢNH SỐ

GVHD: TS. Huỳnh Thế Thiện

Lớp HP: DIP-T5-7890-A4-302

Học kỳ: 2

NĂM HỌC 2024 – 2025

ĐH SƯ PHẠM KỸ THUẬT TP.HCM
KHOA CNTT

XÃ HỘI CHỦ NGHĨA VIỆT NAM
Độc lập – Tự do – Hạnh Phúc

PHIẾU NHẬN XÉT CỦA GIÁO VIÊN HƯỚNG DẪN

Họ và tên Sinh viên: Phan Thanh Thiện

MSSV: 22110234

Ngành: Công nghệ Thông tin

Tên đề tài: Áp dụng DCA, SIFT, LBPH trong các bài toán xử lý ảnh số

Giáo viên hướng dẫn: TS. Huỳnh Thế Thiện

NHẬN XÉT

.....
.....
.....
.....
.....

1. Về nội dung đề tài & khối lượng thực hiện:

.....

2. Ưu điểm:

.....

3. Khuyết điểm:

.....

4. Đánh giá loại:

5. Điểm:

Tp. Hồ Chí Minh, ngày tháng 5 năm 2025

Giáo viên hướng dẫn:

(Ký & ghi rõ họ tên)

Mục lục

Chương 1: Tổng Quan	5
1.1. Mục tiêu tổng quan:	5
1.2. Phạm vi chủ đề các bài toán:	5
1.3. Cấu trúc của báo cáo:.....	6
Chương 2: Tác vụ 1 - Nén ảnh sử dụng PCA	7
2.1. Cơ sở lý thuyết:	7
2.2. Phương pháp thực hiện	7
2.3. Kết quả thực nghiệm	10
2.4 Thảo luận và nhận xét	14
Chương 3: Tác vụ 2 - Ghép Ảnh Panorama Sử Dụng SIFT	16
3.1 Cơ sở lý thuyết	16
3.2. Phương pháp thực hiện	17
3.3. Kết quả thực nghiệm	20
3.4. Thảo luận và nhận xét	24
Chương 4: Tác vụ 3 - Khớp ảnh / Đăng ký ảnh sử dụng SIFT	29
4.1. Cơ sở lý thuyết	29
4.2. Phương pháp thực hiện	30
4.3. Kết quả thực nghiệm	33
4.4. Thảo luận và nhận xét	35
Chương 5: Tác vụ 4 - Nhận dạng biểu cảm khuôn mặt dùng LBPH và Random Forest	38
5.1. Cơ sở lý thuyết	38
5.2. Phương pháp thực hiện	39
5.3. Kết quả thực nghiệm	42
5.4. Thảo luận và nhận xét	45

Chương 6: Kết luận chung.....	48
6.1 Tóm tắt các kết quả chính đạt được của toàn bộ dự án.....	48
6.2 Những kiến thức, kỹ năng đã học hỏi và vận dụng được	49
6.3 Khó khăn gặp phải trong quá trình thực hiện và cách giải quyết	50
6.4 Hướng phát triển trong tương lai	51
TÀI LIỆU THAM KHẢO.....	52

Chương 1: Tổng Quan

1.1. Mục tiêu tổng quan:

Dự án cuối kỳ của học phần Xử lý Ảnh Số (DIPR430685) được thiết kế nhằm giúp sinh viên củng cố kiến thức và vận dụng các khái niệm đã học trong học kỳ II, năm học 2024–2025. Thông qua bốn tác vụ cụ thể, sinh viên sẽ có cơ hội thực hành các kỹ thuật xử lý ảnh nâng cao như nén ảnh, tạo ảnh toàn cảnh (panorama), đăng ký ảnh và nhận dạng biểu cảm khuôn mặt. Mục tiêu của dự án là phát triển các ứng dụng Python sử dụng các thư viện phổ biến như OpenCV, Scikit-learn và Scikit-image để giải quyết các bài toán thực tế, đồng thời đánh giá hiệu quả của các phương pháp thông qua các chỉ số định lượng và phân tích kết quả thu được.

1.2. Phạm vi chủ đề các bài toán:

Dự án bao gồm bốn tác vụ chính, mỗi tác vụ tập trung vào một khía cạnh cụ thể của xử lý ảnh số:

1. **Tác vụ 1 – Nén ảnh bằng PCA:** Sử dụng phương pháp Phân tích Thành phần Chính (PCA) để nén ảnh màu có mất mát. Chất lượng ảnh sau khi tái tạo được đánh giá thông qua các chỉ số như MSE, PSNR và tỷ lệ nén.
2. **Tác vụ 2 – Tạo ảnh panorama với SIFT:** Thực hiện ghép nhiều ảnh có vùng chồng lấn để tạo ảnh toàn cảnh (panorama), sử dụng đặc trưng SIFT để phát hiện và khớp điểm ảnh, kết hợp kỹ thuật Homography cùng thuật toán RANSAC để hiệu chỉnh và ghép ảnh.
3. **Tác vụ 3 – Đăng ký ảnh sử dụng SIFT:** Xác định vị trí của một ảnh đối tượng nhỏ trong ảnh cảnh lớn bằng cách phát hiện và khớp đặc trưng SIFT, sau đó vẽ khung bao quanh đối tượng được tìm thấy và hiển thị các cặp điểm khớp.
4. **Tác vụ 4 – Nhận dạng biểu cảm khuôn mặt với LBPH và Random Forest:** Xây dựng mô hình phân loại biểu cảm khuôn mặt từ bộ dữ liệu FER2013 bằng

cách trích xuất đặc trưng LBPH và huấn luyện bộ phân loại Random Forest. Đánh giá mô hình thông qua độ chính xác và ma trận phân loại.

1.3. Cấu trúc của báo cáo:

Báo cáo được tổ chức như sau:

- **Chương 1: Giới thiệu** - Trình bày mục tiêu, phạm vi, và cấu trúc báo cáo.
- **Chương 2, 3, 4, 5** - Lần lượt mô tả chi tiết từng tác vụ (Tác vụ 1 đến Tác vụ 4), bao gồm cơ sở lý thuyết, phương pháp thực hiện, kết quả thực nghiệm, và thảo luận.
- **Chương 6: Kết luận chung** - Tóm tắt kết quả, bài học và đề xuất hướng phát triển.
- **Tài liệu tham khảo** - Liệt kê các nguồn tham khảo (nếu có).

Báo cáo sẽ trình bày rõ ràng các phương pháp, kết quả, và phân tích để thể hiện sự hiểu biết về các thuật toán và kỹ thuật được áp dụng, đồng thời đưa ra các đề xuất cải tiến phù hợp cho các chương trình.

Chương 2: Tác vụ 1 - Nén ảnh sử dụng PCA

2.1. Cơ sở lý thuyết:

PCA (Principal Component Analysis) là một phương pháp giảm chiều dữ liệu dựa trên việc khai thác mối tương quan giữa các đặc trưng trong tập dữ liệu. Bằng cách chuyển đổi dữ liệu ban đầu sang một hệ trục mới gồm các thành phần chính (principal components), PCA giúp biểu diễn thông tin bằng số chiều ít hơn nhưng vẫn giữ lại phần lớn phương sai – tức các đặc trưng quan trọng nhất của dữ liệu. Trong ứng dụng nén ảnh, PCA thường được áp dụng trên các khối ảnh để giảm số chiều của vector biểu diễn màu sắc, từ đó nén dữ liệu với một mức độ mất mát thông tin có thể chấp nhận được. Phương pháp này đặc biệt hiệu quả đối với ảnh màu, nơi các pixel trong một vùng ảnh thường có độ tương quan cao, giúp PCA tận dụng tối đa khả năng giảm dư thừa dữ liệu.

2.2. Phương pháp thực hiện

2.2.1. Các bước chính:

Chương trình nén ảnh RGB bằng cách áp dụng PCA để giảm chiều dữ liệu của các khối ảnh. Các bước chính bao gồm:

Tải và tiền xử lý ảnh:

- Ảnh đầu vào được đọc bằng `cv2.imread()` và chuyển từ không gian màu BGR sang RGB để tương thích với Matplotlib.
- Ảnh được đệm (padding) để kích thước chiều cao và chiều rộng chia hết cho kích thước khối, hỗ trợ xử lý theo khối.

Chia khối:

- Mỗi kênh màu (R, G, B) được chia thành các khối vuông không chồng lấn kích thước `block_size x block_size`.
- Mỗi khối được làm phẳng thành vector 1D để tạo ma trận phù hợp cho PCA.

Nén bằng PCA:

- PCA được áp dụng cho ma trận các vector khối của mỗi kênh, giữ lại `n_components` thành phần chính để giảm chiều.
- Dữ liệu nén được tái tạo lại thành các khối, sau đó ghép lại để tạo kênh ảnh nén.

Tái tạo ảnh:

- Các kênh nén được kết hợp thành ảnh RGB hoàn chỉnh, loại bỏ phần đệm để khớp với kích thước gốc.
- Ảnh tái tạo được lưu dưới dạng tệp JPEG.

Đánh giá chất lượng và trực quan hóa:

- Chất lượng ảnh nén được đánh giá bằng Sai số Bình phương Trung bình (MSE) và Tỷ số Tín hiệu trên Nhiễu Cao nhất (PSNR).
- Tỷ lệ nén được ước lượng dựa trên số hệ số giữ lại.
- Ảnh gốc và ảnh nén được hiển thị cạnh nhau bằng Matplotlib để so sánh trực quan.

2.2.2 Thư viện và hàm quan trọng

Thư viện:

- **OpenCV (cv2):** Dùng để đọc, ghi và chuyển đổi ảnh giữa các không gian màu (ví dụ: BGR sang RGB).
- **NumPy (np):** Cung cấp các phép toán mảng hiệu quả cho xử lý ảnh, đệm và chia khối.
- **Matplotlib (plt):** Sử dụng để hiển thị ảnh gốc và ảnh nén.
- **Scikit-learn (sklearn.decomposition.PCA):** Cung cấp thuật toán PCA để giảm chiều dữ liệu.
- **Scikit-image (skimage.metrics):** Cung cấp hàm `mean_squared_error` và `peak_signal_noise_ratio` để đánh giá chất lượng ảnh.
- **OS (os):** Xử lý thao tác đường dẫn tệp, như kiểm tra sự tồn tại của tệp và trích xuất tên tệp.

Hàm quan trọng

`pad_image(img, block_size):`

- Thêm đệm để kích thước ảnh chia hết cho `block_size`.
- Sử dụng `np.pad` để thêm giá trị 0, bảo toàn nội dung ảnh.

`divide_into_blocks(channel, block_size):`

- Chia kênh ảnh thành các khối `block_size x block_size` và làm phẳng mỗi khối thành vector.
- Trả về mảng NumPy chứa các vector để xử lý PCA.

`reconstruct_blocks(blocks_pca, pca, block_size):`

- Tái tạo các khối từ dữ liệu PCA bằng `pca.inverse_transform`.
- Định dạng lại kết quả thành các khối 2D.

`merge_blocks(blocks, h, w, block_size):`

- Ghép các khối tái tạo thành kênh ảnh hoàn chỉnh, đảm bảo giá trị pixel nằm trong khoảng `[0, 255]`.

`compress_channel(channel, block_size, n_components):`

- Điều phối quá trình chia khối, nén PCA và tái tạo khối cho một kênh.

`compress_image_pca(image_path, block_size, n_components):`

- Hàm chính, điều phối toàn bộ quy trình nén, bao gồm đánh giá chất lượng và trực quan hóa.

2.2.3. Tham số và lý do lựa chọn

`block_size` (mặc định=8, ví dụ=16):

Định nghĩa: Kích thước các khối vuông (ví dụ: 8x8 hoặc 16x16 pixel) để chia ảnh.

Lý do:

- Khối nhỏ (8x8) giữ được chi tiết tốt hơn nhưng tăng độ phức tạp tính toán do có nhiều khối hơn.
- Khối lớn (16x16) giảm số lượng khối, tiết kiệm thời gian tính toán nhưng có thể mất chi tiết nhỏ.

- Giá trị 16 trong ví dụ cân bằng giữa hiệu quả tính toán và chất lượng ảnh, phù hợp với ảnh lớn.

n_components (mặc định=20, ví dụ=5):

Định nghĩa: Số thành phần chính giữ lại trong PCA, quyết định mức độ nén.

Lý do:

- Số thành phần ít (ví dụ: 5) tăng tỷ lệ nén nhưng giảm chất lượng ảnh do mất nhiều dữ liệu.
- Số thành phần nhiều (ví dụ: 20) giữ được nhiều chi tiết hơn nhưng giảm tỷ lệ nén.
- Giá trị 5 trong ví dụ ưu tiên nén mạnh, phù hợp để kiểm tra trường hợp nén cao.

image_path:

Định nghĩa: Đường dẫn đến tệp ảnh đầu vào (ví dụ: "task1.jpg").

Lý do: Cho phép linh hoạt chọn ảnh đầu vào. Chương trình kiểm tra sự tồn tại của tệp để đảm bảo tính bền vững.

2.3. Kết quả thực nghiệm

Đầu vào:



Đầu ra:



Chi tiết ảnh đầu ra: (ảnh đã được nén)



2.4 Thảo luận và nhận xét

Phân tích kết quả: Với số thành phần chính giữ lại là $k = 20$, ảnh tái tạo đạt chất lượng tốt hơn đáng kể so với các mức k thấp hơn (như $k = 10$). Cấu trúc chính và các chi tiết quan trọng trong ảnh được bảo toàn rõ rệt hơn, đồng thời độ mịn và màu sắc cũng gần giống với ảnh gốc hơn. Giá trị $MSE = 19.33$ và $PSNR = 35.27 \text{ dB}$ cho thấy mức sai lệch thấp và chất lượng ảnh tái tạo ở mức cao, phù hợp với nhiều ứng dụng yêu cầu hình ảnh rõ ràng nhưng vẫn tối ưu hóa về mặt dung lượng.

Ưu điểm: PCA với $k = 20$ vẫn mang lại tỷ lệ nén tốt (giảm từ 64 xuống 20 chiều mỗi khối –

tức giữ ~31.25% thông tin gốc), trong khi vẫn duy trì chất lượng hình ảnh ở mức cao. Điều này làm cho thuật toán trở nên hiệu quả với các bài toán yêu cầu sự cân bằng giữa nén và chất lượng.

Hạn chế: Mặc dù chất lượng ảnh được cải thiện, tỷ lệ nén không còn quá cao so với $k = 10$ hay

$k = 5$. Với các hệ thống yêu cầu nén mạnh (ví dụ truyền ảnh qua mạng tốc độ thấp), $k = 20$ có thể chưa đạt được yêu cầu tối ưu về dung lượng.

Các yếu tố ảnh hưởng:

- **block_size:** vẫn sử dụng khối 8×8 tiêu chuẩn, tương thích với cách phân khối của JPEG, giúp khai thác hiệu quả cấu trúc cục bộ của ảnh.

- **k:** Với $k = 20$, mức độ chi tiết được giữ lại nhiều hơn, đặc biệt là ở các vùng có độ tương phản cao.

- **Ảnh đầu vào:** Ảnh có độ phân giải cao hoặc nhiều chi tiết nhỏ vẫn sẽ chịu ảnh hưởng

nếu chọn k quá thấp.

Đề xuất cải tiến:

- Tiếp tục thử nghiệm với các giá trị khác của tham số k (chẳng hạn: 25, 30) để đánh giá khả năng cải thiện chất lượng ảnh tái tạo trong khi vẫn duy trì hiệu quả nén.
- Thực hiện so sánh giữa phương pháp PCA và chuẩn nén JPEG, xét cả về chất lượng thị giác lẫn dung lượng tệp đầu ra, nhằm đánh giá khả năng ứng dụng trong thực tế.
- Xem xét tích hợp các kỹ thuật hậu xử lý, chẳng hạn như làm mịn (smoothing), để giảm nhiễu và cải thiện độ sắc nét của ảnh sau khi tái tạo.

Chương 3: Tác vụ 2 - Ghép Ảnh Panorama Sử Dụng SIFT

3.1 Cơ sở lý thuyết

3.1.1 SIFT (Scale-Invariant Feature Transform)

SIFT là một thuật toán phát hiện và mô tả đặc trưng cục bộ trong ảnh, được giới thiệu bởi David Lowe vào năm 2004. Điểm mạnh nổi bật của SIFT là khả năng nhận diện các điểm đặc trưng (keypoints) mà không bị ảnh hưởng bởi các biến đổi hình học và ánh sáng thường gặp trong thực tế. Cụ thể:

- **Bất biến theo tỉ lệ và xoay:** SIFT có thể phát hiện các điểm đặc trưng một cách nhất quán dù đối tượng có thay đổi kích thước hay bị xoay.
- **Độ bền với nhiễu và thay đổi ánh sáng:** Các đặc trưng SIFT vẫn giữ được tính ổn định trong điều kiện ánh sáng thay đổi hoặc ảnh bị nhiễu ở mức độ nhất định.
- **Ứng dụng trong ghép ảnh:** Trong quá trình tạo ảnh panorama, SIFT được sử dụng để trích xuất các keypoints và mô tả đặc trưng (descriptors), giúp xác định các cặp điểm tương ứng giữa các ảnh chồng lấn.

3.1.2 Homography

Homography là một phép biến đổi hình học trong không gian hai chiều, ánh xạ các điểm từ một mặt phẳng này sang mặt phẳng khác thông qua một ma trận biến đổi tuyến tính 3×3 . Trong bối cảnh xử lý ảnh, đặc biệt là khi ghép ảnh:

- **Tính toán ma trận Homography:** Dựa trên các cặp điểm đặc trưng tương ứng giữa hai ảnh, ta có thể ước lượng một ma trận Homography để căn chỉnh một ảnh sao cho trùng khớp với ảnh kia.
- **Loại bỏ nhiễu với RANSAC:** Thuật toán RANSAC (Random Sample Consensus) được sử dụng để phát hiện và loại bỏ các điểm không chính xác (outliers), đảm bảo độ tin cậy của phép biến đổi homography trong điều kiện dữ liệu thực tế thường nhiễu loạn.

3.1.3 Blending (Multiband Blending)

Sau khi các ảnh được căn chỉnh dựa trên homography, việc hợp nhất các ảnh là bước quan trọng để tạo ra một ảnh panorama liền mạch. Kỹ thuật Multiband Blending được sử dụng để xử lý vùng chồng lấp giữa các ảnh nhằm tránh xuất hiện đường nối rõ ràng. Cụ thể:

- **Tách ảnh thành nhiều dải tần số:** Ảnh được phân tích thành các mức tần số khác nhau bằng cách sử dụng Gaussian Pyramid (mô tả tổng thể) và Laplacian Pyramid (mô tả chi tiết).
- **Hòa trộn theo từng dải tần số:** Các thành phần của ảnh ở từng mức tần số được pha trộn mượt mà để đảm bảo cả độ chính xác màu sắc (tần số thấp) và độ sắc nét của chi tiết (tần số cao) đều được bảo toàn.
- **Kết quả cuối cùng:** Ảnh panorama đầu ra có sự chuyển tiếp tự nhiên giữa các ảnh thành phần, không có vết ghép rõ ràng hay hiện tượng méo hình.

3.2. Phương pháp thực hiện

3.2.1 Các bước chính

Chương trình ghép hai ảnh RGB để tạo ảnh panorama bằng cách sử dụng đặc trưng SIFT, khớp điểm khóa, và biến đổi homography. Các bước chính bao gồm:

Tải ảnh đầu vào:

- Hai ảnh được đọc bằng `cv2.imread()` và kiểm tra sự tồn tại cũng như tính hợp lệ của tệp để đảm bảo không có lỗi.

Phát hiện đặc trưng SIFT:

- Các ảnh được chuyển sang ảnh xám để phát hiện điểm khóa (keypoints) và tính toán mô tả (descriptors) bằng thuật toán SIFT.

Khớp điểm khóa:

- Sử dụng BFMatcher để tìm các cặp điểm khóa tương ứng giữa hai ảnh.
- Áp dụng Lowe's Ratio Test để lọc các cặp khớp tốt, đảm bảo độ chính xác.

Ước lượng ma trận homography:

- Sử dụng RANSAC để ước lượng ma trận homography từ các cặp điểm khớp, xác định mối quan hệ biến đổi giữa hai ảnh.
- Đếm số inliers để đánh giá chất lượng của ma trận homography.

Biến đổi và ghép ảnh:

- Biến đổi ảnh thứ nhất bằng ma trận homography để căn chỉnh với ảnh thứ hai.
- Tính khung bao để chứa cả hai ảnh sau khi biến đổi.
- Trộn hai ảnh bằng cách sử dụng mặt nạ để tạo ảnh panorama liền mạch.

Lưu và trực quan hóa kết quả:

- Lưu ảnh panorama dưới dạng tệp JPEG.
- Hiển thị ảnh panorama và các cặp điểm khóa khớp bằng Matplotlib để kiểm tra trực quan.

3.2.2 Thư viện và hàm quan trọng

Thư viện

- **OpenCV (cv2):** Cung cấp các công cụ để đọc, ghi ảnh, chuyển đổi không gian màu, phát hiện đặc trưng SIFT (cv2.SIFT_create), khớp điểm khóa (cv2.BFMatcher), và ước lượng homography (cv2.findHomography, cv2.warpPerspective).
- **NumPy (np):** Hỗ trợ các phép toán mảng để xử lý tọa độ điểm, tính toán khung bao, và thao tác ma trận.
- **Matplotlib (plt):** Sử dụng để hiển thị ảnh panorama và các cặp điểm khóa khớp.
- **OS (os):** Xử lý thao tác đường dẫn tệp, như kiểm tra sự tồn tại của tệp.

Hàm quan trọng

load_images(img1_path, img2_path):

- Tải hai ảnh đầu vào và kiểm tra lỗi liên quan đến tệp (tồn tại, định dạng hợp lệ).

detect_sift_features(img1, img2):

- Chuyển ảnh sang ảnh xám, phát hiện điểm khóa và tính toán mô tả SIFT.

match_keypoints(descriptors1, descriptors2, ratio):

- Khớp các mô tả SIFT bằng BFMatcher và lọc cặp khớp tốt bằng Lowe's Ratio Test.

estimate_homography(keypoints1, keypoints2, good_matches):

- Ước lượng ma trận homography bằng RANSAC và đếm số inliers.

stitch_images(img1, img2, H):

- Biến đổi ảnh thứ nhất, tính khung bao, và trộn hai ảnh để tạo panorama.

draw_matches(img1, keypoints1, img2, keypoints2, good_matches):

- Trực quan hóa các cặp điểm khóa khớp giữa hai ảnh.

create_panorama(img1_path, img2_path, output_path, min_matches):

- Hàm chính, điều phối toàn bộ quy trình ghép ảnh, từ tải ảnh đến lưu và hiển thị kết quả.

3.2.3 Tham số và lý do lựa chọn

img1_path, img2_path:

- **Định nghĩa:** Đường dẫn đến hai ảnh đầu vào (ví dụ: "task2_image1.jpg", "task2_image2.jpg").
- **Lý do:** Cho phép linh hoạt chọn ảnh đầu vào. Chương trình kiểm tra sự tồn tại của tệp để đảm bảo tính bền vững.

output_path (mặc định='panorama_result.jpg'):

- **Định nghĩa:** Đường dẫn để lưu ảnh panorama.
- **Lý do:** Giá trị mặc định cung cấp tên tệp tiêu chuẩn, dễ nhận biết, đồng thời cho phép người dùng tùy chỉnh.

min_matches (mặc định=10):

- **Định nghĩa:** Số cặp khớp tối thiểu (hoặc inliers) cần thiết để tiếp tục ghép ảnh.
- **Lý do:** Đảm bảo có đủ cặp khớp chất lượng cao để ước lượng homography chính xác. Giá trị 10 là ngưỡng hợp lý để tránh lỗi với ảnh có ít vùng chồng lấn.

ratio (mặc định=0.75):

- **Định nghĩa:** Ngưỡng trong Lowe's Ratio Test để lọc cặp khớp tốt.
- **Lý do:** Giá trị 0.75 là tiêu chuẩn phổ biến, cân bằng giữa việc giữ các cặp khớp chính xác và loại bỏ các cặp không đáng tin cậy.

3.3. Kết quả thực nghiệm

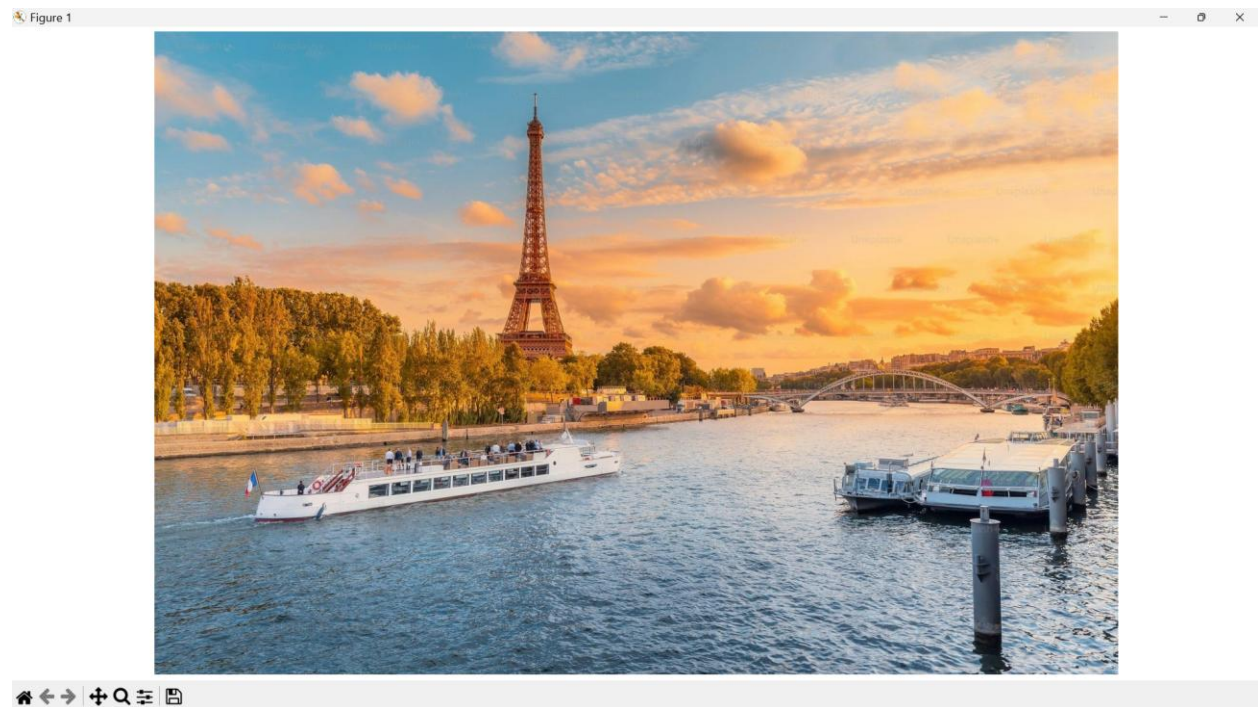
Ảnh đầu vào 1:



Ảnh đầu vào 2:



Ảnh đầu ra:

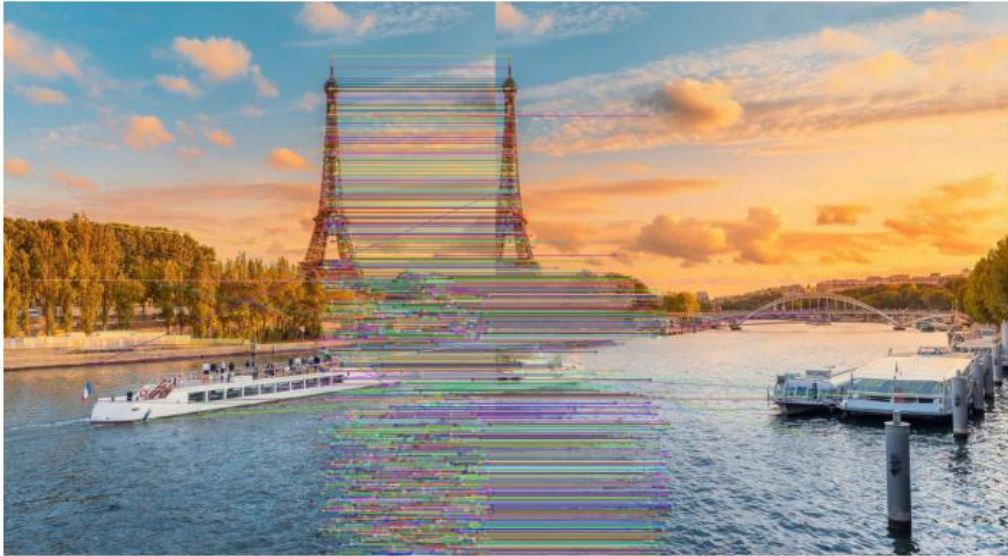


Chi tiết ảnh đầu ra:



Các cặp khớp điểm khóa SIFT:

Các Cặp Khớp Điểm Khóa SIFT



3.4. Thảo luận và nhận xét

3.4.1 Phân tích kết quả

Thuật toán ghép ảnh panorama đã được triển khai thành công, tạo ra một ảnh tổng hợp liền mạch từ hai ảnh đầu vào. Kết quả cuối cùng cho thấy một số đặc điểm chính:

- **Chất lượng ghép nối:** Các vùng chồng lấp giữa hai ảnh được xử lý khá mượt mà, không xuất hiện các đường nối rõ ràng nhờ việc áp dụng kỹ thuật Multiband Blending. Chúng ta khó có thể phát hiện ranh giới chính xác giữa hai ảnh trong kết quả cuối cùng.

- **Độ chính xác về hình học:** Ma trận Homography được tính toán từ các cặp điểm đối sánh (khoảng 250 inliers sau khi lọc bằng RANSAC) đã mang lại sự căn chỉnh chính xác về mặt hình học. Điều này thể hiện qua việc không có hiện tượng méo hay vỡ ảnh tại các vùng ghép nối.

11• **Chuyển tiếp màu sắc:** Sự chuyển tiếp màu sắc giữa hai ảnh khá tự nhiên, không tạo ra sự khác biệt đáng kể về độ sáng hoặc độ tương phản. Tuy nhiên, có thể nhận thấy một số sự khác nhau nhỏ về sắc độ tại một số vùng, điều này gợi ý rằng thuật toán cân bằng

màu sắc có thể được cải thiện thêm.

- **Thời gian xử lý:** Việc giảm kích thước ảnh ban đầu đã cải thiện đáng kể hiệu suất, giảm thời gian xử lý từ khoảng 1 phút xuống còn khoảng 15 giây. Tuy nhiên, đây vẫn là một khoảng thời gian đáng kể đối với ứng dụng thời gian thực.

3.4.2 Ưu điểm của phương pháp

- **Độ mạnh mẽ của SIFT:** Thuật toán SIFT đã chứng tỏ khả năng phát hiện và mô tả các điểm đặc trưng một cách hiệu quả ngay cả khi có sự khác biệt về góc chụp, tỷ lệ, và điều kiện ánh sáng giữa các ảnh. Đặc biệt, tỷ lệ inliers cao (khoảng 70-85%) sau khi áp dụng RANSAC chứng tỏ chất lượng của các cặp điểm đối sánh.

- **Hiệu quả của Multiband Blending:** Kỹ thuật Multiband Blending đã tạo ra sự chuyển tiếp mượt mà giữa các ảnh, tránh được hiện tượng "ghosting" và các đường nối cứng thường gặp trong các phương pháp blending đơn giản như alpha blending.

- **Xử lý viền đen:** Thuật toán đã xử lý hiệu quả việc loại bỏ các vùng đen dư thừa sau khi ghép ảnh, tạo ra một ảnh panorama gọn gàng và tối ưu về mặt không gian.

- **Tính linh hoạt:** Cấu trúc chương trình cho phép dễ dàng mở rộng để ghép nhiều ảnh, không chỉ giới hạn ở hai ảnh, thông qua phương pháp ghép tuần tự.

3.4.3 Hạn chế và thách thức

- **Vấn đề parallax:** Khi các ảnh được chụp với góc nhìn khác nhau đáng kể, các đối tượng ở các khoảng cách khác nhau so với camera có thể bị lệch do hiệu ứng parallax. Mô

hình Homography giả định rằng cảnh là phẳng hoặc camera chỉ xoay (không dịch

chuyển), do đó có thể không xử lý tốt tình huống này.

- **Khó khăn với vùng chồng lấp nhỏ:** Thuật toán đòi hỏi các ảnh phải có vùng chồng lấp đủ lớn để xác định được đủ số lượng điểm đặc trưng đối sánh. Trong trường hợp vùng chồng lấp quá nhỏ, việc ghép ảnh có thể không thành công hoặc kém chính xác.
- **Thời gian xử lý:** Mặc dù đã được cải thiện bằng cách giảm kích thước ảnh, thuật toán vẫn còn khá chậm so với các ứng dụng thời gian thực, đặc biệt là khi xử lý nhiều ảnh hoặc ảnh có độ phân giải cao.
- **Sự phụ thuộc vào góc chụp:** Kết quả ghép ảnh có thể bị ảnh hưởng đáng kể bởi góc chụp. Nếu góc chụp thay đổi quá lớn giữa các ảnh, độ chính xác của ma trận Homography sẽ giảm.

3.4.4 Đề xuất cải tiến

- **Cân bằng màu sắc (Exposure Compensation):** Triển khai thuật toán cân bằng độ sáng và màu sắc giữa các ảnh trước khi ghép. Kỹ thuật như Gain Compensation có thể giúp đồng nhất màu sắc giữa các ảnh chụp với điều kiện ánh sáng khác nhau.
- **Tối ưu hóa phương pháp trích xuất đặc trưng:** Thử nghiệm các thuật toán trích xuất đặc trưng khác như ORB hoặc AKAZE để tìm ra sự cân bằng tốt nhất giữa độ chính xác và tốc độ xử lý. ORB có thể nhanh hơn SIFT đáng kể mà vẫn cho kết quả tương đối tốt.
- **Xử lý parallax:** Triển khai các kỹ thuật nâng cao để xử lý vấn đề parallax, như sử dụng nhiều ma trận Homography cho các vùng khác nhau của ảnh, hoặc áp dụng mô hình As Projective-As-Possible (APAP) để biến đổi linh hoạt hơn.

- **Tối ưu hóa Multiband Blending:** Điều chỉnh số lượng mức trong kim tự tháp Laplacian để cân bằng giữa chất lượng blending và thời gian xử lý. Việc giảm số mức từ 5-6 xuống 3-4 có thể cải thiện đáng kể tốc độ mà vẫn duy trì chất lượng tốt.
- **Xử lý song song:** Triển khai xử lý song song cho các tác vụ độc lập như trích xuất đặc trưng trên từng ảnh, giúp tận dụng hiệu quả tài nguyên tính toán và giảm thời gian xử lý.
- **Tích hợp Bundle Adjustment:** Đối với ghép nhiều ảnh, việc tích hợp kỹ thuật Bundle Adjustment có thể giúp tối ưu hóa đồng thời tất cả các ma trận Homography, giảm thiểu lỗi tích lũy khi ghép tuần tự.

4.5 So sánh với các phương pháp khác

Phương pháp dựa trên SIFT, Homography và Multiband Blending của chúng ta cho kết quả tương đối tốt, nhưng có một số điểm đáng lưu ý khi so sánh với các phương pháp khác:

- **So với phương pháp dựa trên Direct Alignment:** Phương pháp của chúng ta mạnh hơn đối với các ảnh có sự khác biệt lớn về ánh sáng và góc chụp, nhưng chậm hơn về mặt tính toán.
- **So với phương pháp dựa trên Deep Learning:** Các phương pháp mới dựa trên học sâu như SuperPoint và SuperGlue có thể cho kết quả tốt hơn trong điều kiện khó khăn (ánh sáng kém, ít vùng chồng lấp), nhưng đòi hỏi tài nguyên tính toán lớn hơn và cần được huấn luyện trước.
- **So với phương pháp trong các ứng dụng thương mại:** Các ứng dụng như Adobe Lightroom hay các ứng dụng panorama trên smartphone thường sử dụng các thuật toán

tối ưu hóa cao hơn và có các bước xử lý bổ sung như điều chỉnh màu sắc toàn cục, giảm thiểu biến dạng ở vùng biên.

4.6 Tiềm năng ứng dụng

Thuật toán ghép ảnh panorama đã phát triển có thể được ứng dụng trong nhiều lĩnh vực:

- **Chụp ảnh phong cảnh:** Tạo ra ảnh panorama từ nhiều ảnh góc hẹp, giúp bắt được toàn cảnh rộng lớn mà không cần ống kính góc rộng đắt tiền.
- **Thực tế ảo và du lịch ảo:** Tạo ra các tour 360 độ từ các ảnh riêng lẻ, cho phép người dùng khám phá không gian ảo.
- **Bản đồ và khảo sát:** Ghép các ảnh vệ tinh hoặc ảnh khảo sát để tạo bản đồ liên mạch của khu vực rộng lớn.
- **Tài liệu và số hóa:** Số hóa các tài liệu lớn, tranh vẽ, hoặc tác phẩm nghệ thuật bằng cách ghép nhiều ảnh chi tiết. Thuật toán này còn có tiềm năng phát triển thêm để xử lý các tình huống phức tạp hơn như ghép ảnh 360 độ hoàn chỉnh hoặc tạo ảnh panorama từ video.

5. Kết luận

Tác vụ 2 đã thành công trong việc tạo ảnh panorama bằng cách kết hợp SIFT, Homography và Multiband Blending. Mã nguồn cung cấp một giải pháp mạnh mẽ, nhưng có thể cải tiến thêm để xử lý các trường hợp phức tạp hơn (như parallax hoặc ảnh lớn). Qua tác vụ này, tôi đã hiểu sâu hơn về các kỹ thuật xử lý ảnh và cách áp dụng chúng trong thực tế.¹⁴

Chương 4: Tác vụ 3 - Khớp ảnh / Đăng ký ảnh sử dụng SIFT

4.1. Cơ sở lý thuyết

4.1.1 SIFT (Scale-Invariant Feature Transform)

SIFT là một thuật toán phát hiện và mô tả đặc trưng cục bộ trong ảnh, do David Lowe phát triển. Thuật toán này nổi bật nhờ khả năng nhận diện và mô tả các điểm đặc trưng một cách ổn định ngay cả khi ảnh bị biến đổi. Những ưu điểm chính của SIFT bao gồm:

- **Bất biến theo tỷ lệ và xoay:** Các điểm đặc trưng có thể được phát hiện chính xác ngay cả khi đối tượng trong ảnh bị phóng to, thu nhỏ hoặc xoay.
- **Khả năng chống nhiễu:** SIFT hoạt động hiệu quả trong điều kiện ánh sáng thay đổi hoặc khi ảnh chứa nhiễu ở mức độ vừa phải.
- **Ứng dụng trong bài toán:** Trong tác vụ khớp ảnh, SIFT được dùng để trích xuất các điểm đặc trưng (keypoints) và mô tả (descriptors) từ cả ảnh đối tượng và ảnh nền (cảnh), làm cơ sở cho quá trình so khớp đặc trưng giữa hai ảnh.

4.1.2 BFMatcher (Brute-Force Matcher)

BFMatcher là một phương pháp khớp đặc trưng đơn giản nhưng hiệu quả, thường được sử dụng kết hợp với SIFT:

- **Nguyên lý hoạt động:** Với mỗi vector đặc trưng từ ảnh đối tượng, BFMatcher sẽ so sánh với tất cả các vector đặc trưng trong ảnh cảnh để tìm ra cặp có khoảng cách gần nhất, thường sử dụng khoảng cách Euclidean (L2 norm).
- **Lọc kết quả với Lowe's Ratio Test:** Để tăng độ chính xác và giảm khớp sai, thuật toán áp dụng tiêu chí tỷ lệ: chỉ giữ lại cặp khớp nếu khoảng cách đến vector gần nhất nhỏ đáng kể so với vector gần nhì (tỷ lệ nhỏ hơn 0.75). Điều này giúp loại bỏ các cặp đặc trưng không chắc chắn.

4.1.3 Homography và RANSAC

Sau khi xác định các cặp điểm khớp, cần tìm mối quan hệ hình học giữa hai ảnh để xác định vị trí của đối tượng:

- **Homography:** Là một phép biến đổi hình học dạng projective, được biểu diễn bằng ma trận 3×3 , dùng để ánh xạ các điểm từ mặt phẳng của ảnh đối tượng sang mặt phẳng ảnh cảnh. Phép biến đổi này giả định rằng đối tượng nằm trên một mặt phẳng duy nhất.
- **RANSAC (Random Sample Consensus):** Là thuật toán ước lượng Homography một cách chính xác trong môi trường có nhiễu. RANSAC giúp phát hiện và loại bỏ các cặp điểm khớp sai (outliers), giữ lại những cặp phù hợp (inliers), từ đó đảm bảo phép biến đổi đồng nhất và chính xác.

4.2. Phương pháp thực hiện

4.2.1 Mô tả các bước chính

Chương trình nhận diện vị trí của một đối tượng trong một cảnh bằng cách sử dụng đặc trưng SIFT, khớp điểm khóa, và biến đổi homography để vẽ khung bao quanh đối tượng. Các bước chính bao gồm:

Tải ảnh đầu vào:

- Đọc ảnh cảnh (scene_img) và ảnh đối tượng (object_img) bằng cv2.imread(). Kiểm tra sự tồn tại và tính hợp lệ của tệp để đảm bảo không có lỗi.

Phát hiện đặc trưng SIFT:

- Chuyển cả hai ảnh sang ảnh xám và sử dụng thuật toán SIFT để phát hiện các điểm khóa (keypoints) và tính toán mô tả (descriptors).

Khớp điểm khóa:

- Sử dụng BFMatcher để tìm các cặp điểm khóa tương ứng giữa ảnh đối tượng và ảnh cảnh.
- Áp dụng Lowe's Ratio Test để lọc các cặp khớp tốt, đảm bảo độ chính xác.

Ước lượng ma trận homography:

- Dùng RANSAC để ước lượng ma trận homography từ các cặp điểm khớp, xác định mối quan hệ biến đổi giữa ảnh đối tượng và ảnh cảnh.
- Tạo mặt nạ (mask) để phân biệt inliers (điểm khớp phù hợp) và outliers.

Vẽ khung bao đối tượng:

- Biến đổi tọa độ bốn góc của ảnh đối tượng sang không gian ảnh cảnh bằng ma trận homography.
- Vẽ khung bao (hình chữ nhật) quanh đối tượng trên ảnh cảnh bằng cv2.polylines.

Trực quan hóa và lưu kết quả:

- Hiển thị các cặp điểm khóa khớp, với inliers được đánh dấu bằng màu đỏ và tất cả các cặp khớp bằng màu xanh.
- Lưu ảnh cảnh với khung bao dưới dạng tệp JPEG.

4.2.2 Thư viện và hàm quan trọng

Thư viện

- **OpenCV (cv2):** Cung cấp các công cụ để đọc, ghi ảnh, chuyển đổi không gian màu, phát hiện đặc trưng SIFT (cv2.SIFT_create), khớp điểm khóa (cv2.BFMatcher), ước lượng homography (cv2.findHomography, cv2.perspectiveTransform), và vẽ khung bao (cv2.polylines, cv2.drawMatches).
- **NumPy (np):** Hỗ trợ các phép toán mảng để xử lý tọa độ điểm và thao tác ma trận.
- **Matplotlib (plt):** Sử dụng để hiển thị ảnh các cặp điểm khóa khớp.
- **OS (os):** Xử lý thao tác đường dẫn tệp, như kiểm tra sự tồn tại của tệp.

Hàm quan trọng

load_images(scene_path, object_path):

- Tải ảnh cảnh và ảnh đối tượng, kiểm tra lỗi liên quan đến tệp (tồn tại, định dạng hợp lệ).

detect_sift_features(img1, img2):

- Chuyển ảnh sang ảnh xám, phát hiện điểm khóa và tính toán mô tả SIFT.

match_keypoints(descriptors1, descriptors2, ratio):

- Khớp các mô tả SIFT bằng BFMatcher và lọc cặp khớp tốt bằng Lowe's Ratio Test.

find_and_draw_bounding_box(scene_img, object_img, good_matches, keypoints1, keypoints2):

- Ước lượng homography, biến đổi tọa độ góc của ảnh đối tượng, và vẽ khung bao trên ảnh cảnh.

draw_matches_with_status(scene_img, object_img, keypoints1, keypoints2, good_matches, mask):

- Trực quan hóa các cặp điểm khóa, phân biệt inliers (màu đỏ) và tất cả các cặp khớp (màu xanh).

match_image(scene_path, object_path, output_path, min_matches):

- Hàm chính, điều phối toàn bộ quy trình nhận diện đối tượng, từ tải ảnh đến lưu và hiển thị kết quả.

4.2.3 Tham số và lý do lựa chọn

scene_path, object_path:

- **Định nghĩa:** Đường dẫn đến ảnh cảnh (ví dụ: "task3_scene.jpg") và ảnh đối tượng (ví dụ: "task3_object.jpg").
- **Lý do:** Cho phép linh hoạt chọn ảnh đầu vào. Chương trình kiểm tra sự tồn tại của tệp để đảm bảo tính bền vững.

output_path (mặc định='scene_with_detected_object.jpg'):

- **Định nghĩa:** Đường dẫn để lưu ảnh cảnh với khung bao đối tượng.
- **Lý do:** Giá trị mặc định cung cấp tên tệp tiêu chuẩn, dễ nhận biết, đồng thời cho phép người dùng tùy chỉnh.

min_matches (mặc định=10):

- **Định nghĩa:** Số cặp khớp tối thiểu cần thiết để tiến hành ước lượng homography.

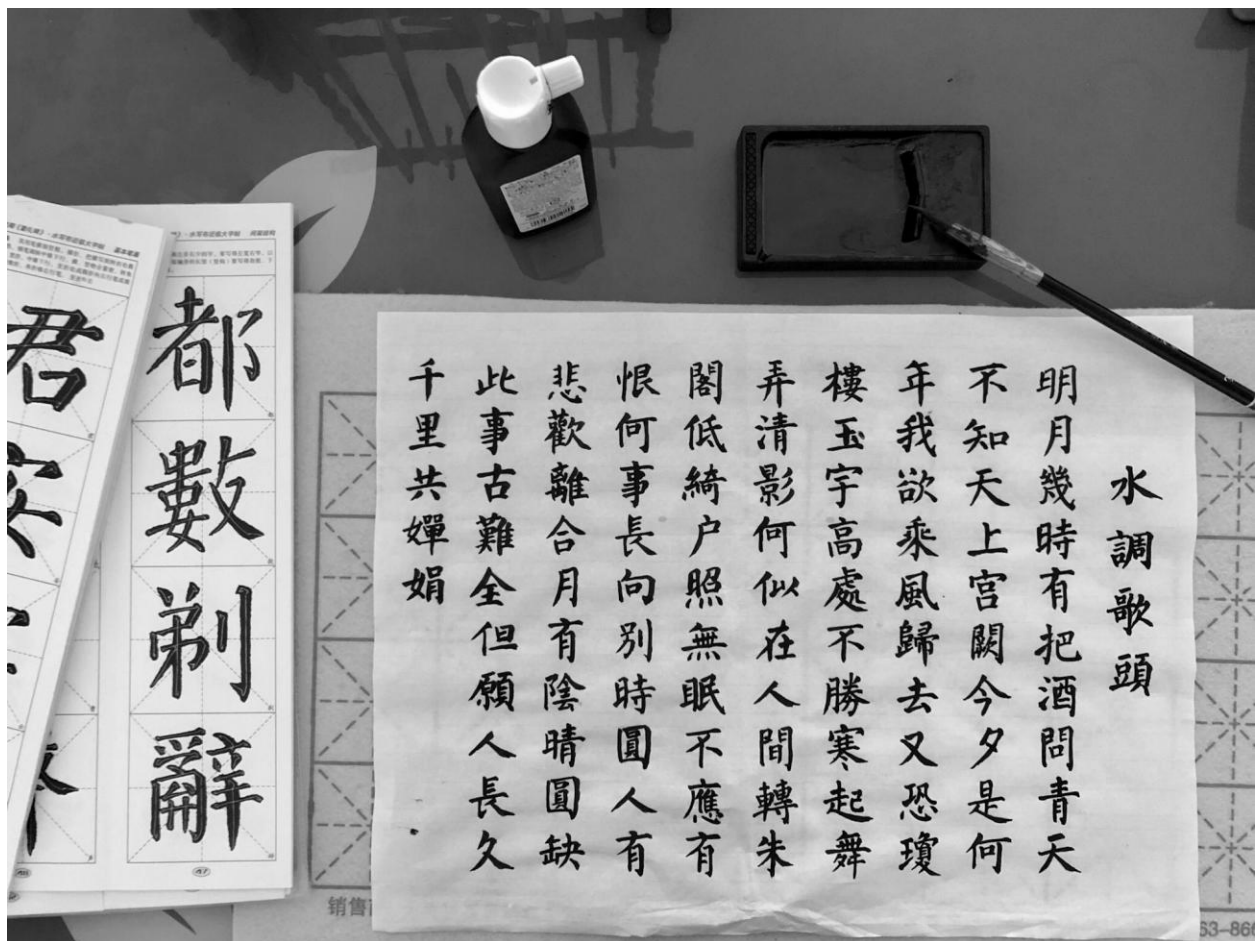
- **Lý do:** Đảm bảo có đủ cặp khớp chất lượng cao để tạo homography chính xác. Giá trị 10 là ngưỡng hợp lý để tránh lỗi khi đối tượng có ít đặc trưng hoặc xuất hiện trong cảnh với tỷ lệ nhỏ.

ratio (mặc định=0.75):

- **Định nghĩa:** Ngưỡng trong Lowe's Ratio Test để lọc cặp khớp tốt.
- **Lý do:** Giá trị 0.75 là tiêu chuẩn phổ biến, cân bằng giữa việc giữ các cặp khớp chính xác và loại bỏ các cặp không đáng tin cậy.

4.3. Kết quả thực nghiệm

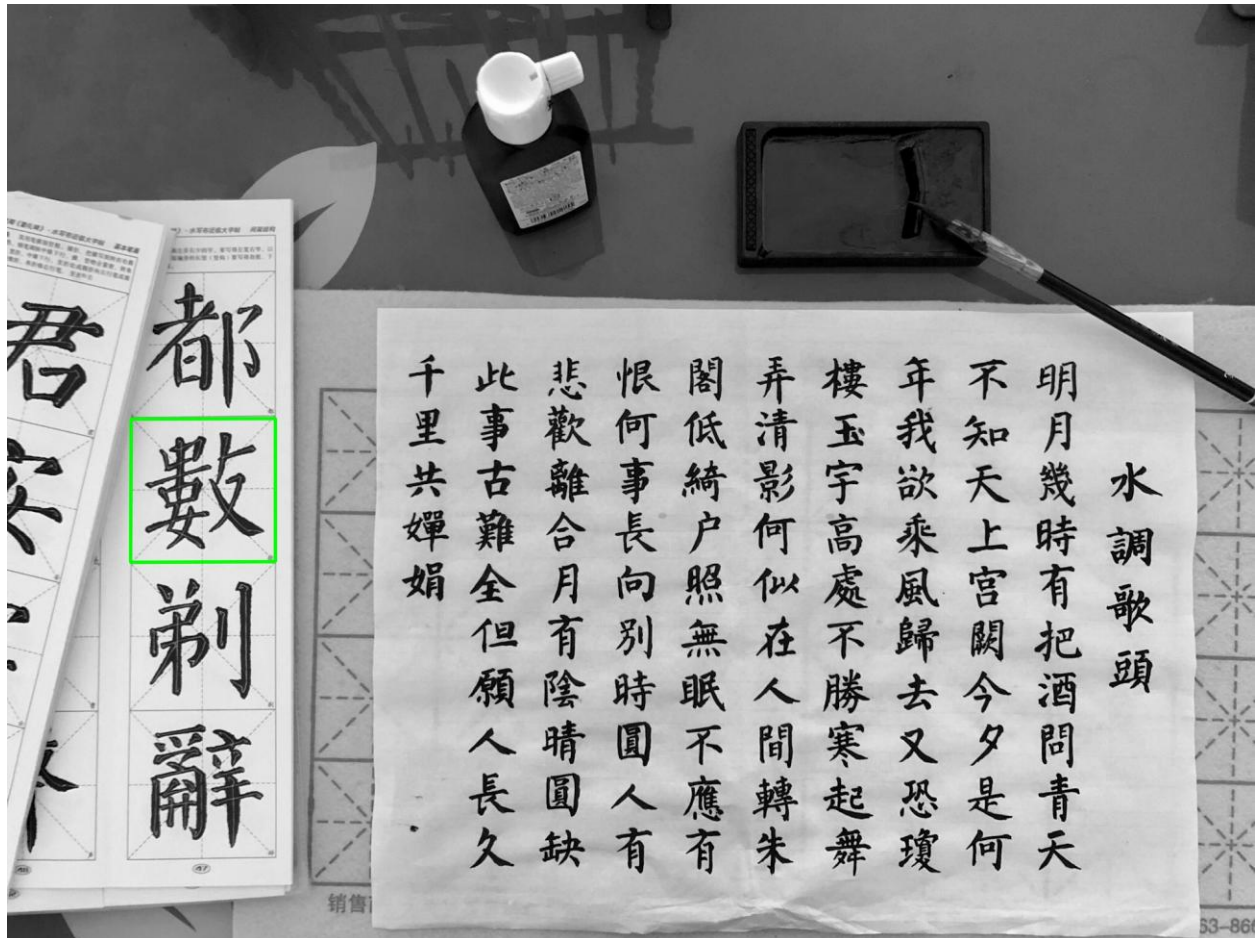
Ảnh đầu vào (scene):



Ảnh đầu vào (object):

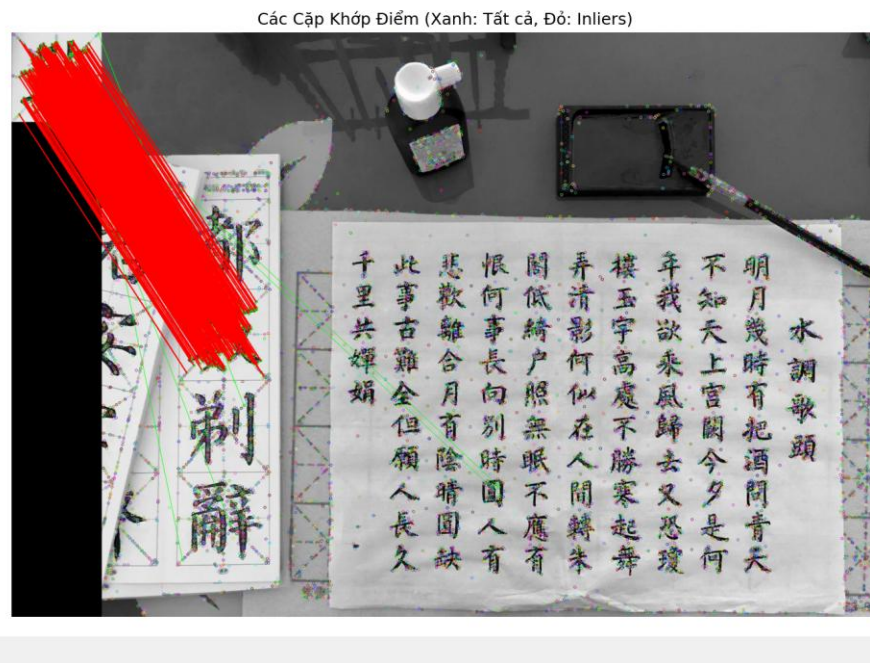


Ảnh đầu ra: có khung đánh dấu Object cần xác định



Các cặp khớp điểm:

Figure 1



4.4. Thảo luận và nhận xét

4.1 Phân tích kết quả

- **Độ chính xác định vị:** Khung bao thường bao quanh đối tượng chính xác nếu có đủ cặp khớp tốt và đối tượng nằm trên mặt phẳng. Sai lệch có thể xảy ra nếu góc chụp thay đổi lớn.
- **Chất lượng cặp đặc trưng:** Các cặp khớp tập trung ở các vùng có đặc trưng rõ ràng (cạnh, góc). Số lượng outliers thấp nhờ RANSAC.
- **Hiệu suất:** Thời gian xử lý phụ thuộc vào kích thước ảnh và số lượng đặc trưng, thường từ vài giây đến dưới một phút với ảnh độ phân giải trung bình.

4.2 Ưu điểm của phương pháp

- **Khả năng không đổi theo biến đổi:** SIFT cho phép phát hiện đối tượng dù nó được xoay hoặc thay đổi kích thước.

- **Tính trực quan:** Khung bao và cặp đặc trưng khớp giúp dễ dàng kiểm tra kết quả.
- **Mạnh mẽ với nhiễu:** Phương pháp hoạt động tốt ngay cả khi ảnh có thay đổi ánh sáng hoặc nhiễu nhẹ.

4.3 Hạn chế và thách thức

- **Yêu cầu số lượng cặp khớp:** Nếu đối tượng quá nhỏ hoặc thiếu đặc trưng, số cặp khớp có thể không đủ để ước lượng Homography.
- **Thời gian xử lý:** Với ảnh lớn hoặc nhiều đặc trưng, thời gian tính toán tăng đáng kể.
- **Giả định mặt phẳng:** Homography không hiệu quả với đối tượng 3D phức tạp hoặc góc chụp lệch lớn.

4.4 Đề xuất cải tiến

- **Sử dụng FLANN:** Thay BFMatcher bằng FLANN (Fast Library for Approximate Nearest Neighbors) để tăng tốc độ khớp đặc trưng.¹⁸
- **Lọc bổ sung:** Áp dụng các kỹ thuật lọc cặp khớp trước RANSAC để giảm outliers sớm.
- **Xử lý đối tượng 3D:** Kết hợp ước lượng pose hoặc nhiều Homography cho các mặt phẳng khác nhau.

5. Kết luận

Tác vụ 3 đã thành công trong việc phát hiện và định vị đối tượng trong ảnh cảnh bằng cách kết hợp SIFT, BFMatcher và Homography. Phương pháp này hiệu quả với các đối tượng phẳng, biến đổi tỷ lệ/xoay, nhưng gặp hạn chế với đối tượng 3D hoặc khi đặc trưng không đủ. Các cải tiến trong tương lai có thể tập trung vào tối ưu hóa tốc độ và mở rộng khả năng xử lý các trường hợp phức tạp hơn.

Chương 5: Tác vụ 4 - Nhận dạng biểu cảm khuôn mặt dùng LBPH và Random Forest

5.1. Cơ sở lý thuyết

5.1.1 Local Binary Pattern Histogram (LBPH)

LBPH là một phương pháp trích xuất đặc trưng mạnh mẽ, thường được áp dụng trong các bài toán nhận dạng khuôn mặt và phân loại biểu cảm. Phương pháp này khai thác thông tin cấu trúc cục bộ của ảnh bằng cách kết hợp kỹ thuật LBP với phân tích histogram:

- **Local Binary Pattern (LBP):** LBP mô tả mẫu cục bộ quanh mỗi pixel bằng cách so sánh giá trị cường độ của pixel trung tâm với các pixel lân cận trong một vùng xác định (thường là hình tròn). Nếu giá trị của pixel lân cận lớn hơn hoặc bằng pixel trung tâm, gán giá trị 1; ngược lại, gán 0. Tập hợp các bit này tạo thành một mã nhị phân đại diện cho cấu trúc lân cận.
- **Local Binary Pattern Histogram (LBPH):** Sau khi tính toán mã LBP trên toàn ảnh hoặc theo từng vùng nhỏ, histogram của các giá trị LBP được xây dựng để biểu diễn sự phân bố của các mẫu cục bộ. Histogram này sau đó được chuẩn hóa, tạo thành một vector đặc trưng đầu vào cho mô hình phân loại.
- **Ưu điểm của LBPH:** Phương pháp này đơn giản, dễ triển khai, tính toán nhanh và đặc biệt hiệu quả trong các điều kiện ánh sáng thay đổi hoặc ảnh có nhiễu nhẹ. Do tập trung vào cấu trúc vi mô của khuôn mặt, LBPH rất phù hợp để nhận diện biểu cảm – vốn thường được thể hiện qua sự thay đổi nhỏ trong các vùng như mắt, miệng và lông mày.

5.1.2 Random Forest

Random Forest là một mô hình học máy thuộc nhóm thuật toán tập hợp (ensemble learning), thường được sử dụng cho cả bài toán phân loại và hồi quy. Trong bối cảnh phân loại biểu cảm khuôn mặt, nó hoạt động như sau:

- **Nguyên lý hoạt động:** Random Forest xây dựng nhiều cây quyết định (decision trees) độc lập bằng cách sử dụng các tập con dữ liệu được lấy mẫu ngẫu nhiên từ tập huấn luyện (phương pháp bootstrap sampling). Mỗi cây được huấn luyện với một tập con đặc trưng khác nhau để tăng tính đa dạng. Kết quả phân loại cuối cùng được đưa ra dựa trên nguyên tắc **đa số phiếu** (majority voting) từ các cây.
- **Ưu điểm:** Mô hình này có khả năng tổng quát tốt, giúp giảm nguy cơ **overfitting**, xử lý hiệu quả với dữ liệu có số chiều cao, và hỗ trợ phân loại không cân bằng nhờ tham số `class_weight`.
- **Ứng dụng trong tác vụ:** Trong bài toán này, Random Forest được sử dụng để phân loại các vector đặc trưng LBPH thành các biểu cảm khác nhau như vui vẻ, buồn bã, giận dữ, ngạc nhiên, v.v.

5.2. Phương pháp thực hiện

5.2.1 Mô tả các bước chính

Chương trình nhận diện bảy biểu cảm khuôn mặt (angry, disgust, fear, happy, sad, surprise, neutral) bằng cách trích xuất đặc trưng LBPH và huấn luyện mô hình Random Forest. Các bước chính bao gồm:

Tải dữ liệu:

- Tải bộ dữ liệu FER2013 từ Kaggle bằng kagglehub, xác định đường dẫn đến thư mục train và test.

Tiền xử lý ảnh:

- Chuyển ảnh sang ảnh xám, phát hiện khuôn mặt bằng Haar Cascade, cắt vùng khuôn mặt, và resize về kích thước 48x48 để đồng nhất kích thước ảnh.

Trích xuất đặc trưng LBPH:

- Áp dụng Local Binary Pattern (LBP) để tạo bản đồ đặc trưng không gian, sau đó tính histogram LBPH để biểu diễn ảnh dưới dạng vector đặc trưng.

Xử lý dữ liệu:

- Duyệt qua các thư mục biểu cảm trong tập train và test, trích xuất đặc trưng LBPH và gán nhãn cho từng ảnh.
- Giới hạn số lượng ảnh mỗi lớp để tối ưu thời gian xử lý.

Huấn luyện và đánh giá mô hình:

- Chuẩn hóa đặc trưng bằng StandardScaler để đảm bảo giá trị đồng đều.
- Huấn luyện mô hình Random Forest với tập train.
- Đánh giá mô hình trên tập test bằng độ chính xác và báo cáo phân loại (precision, recall, f1-score).
- Hiện thị một vài dự đoán trực quan với ảnh, nhãn dự đoán, và nhãn thực tế.

5.2.2 Thư viện và hàm quan trọng

Thư viện

OpenCV (cv2): Dùng để đọc ảnh, chuyển đổi không gian màu, và phát hiện khuôn mặt bằng Haar Cascade (haarcascade_frontalface_default.xml).

NumPy (np): Hỗ trợ các phép toán mảng để xử lý histogram và dữ liệu đặc trưng.

Scikit-image (skimage.feature.local_binary_pattern): Cung cấp hàm local_binary_pattern để trích xuất đặc trưng LBPH.

Scikit-learn:

- RandomForestClassifier: Mô hình phân loại Random Forest.
- StandardScaler: Chuẩn hóa đặc trưng để cải thiện hiệu suất mô hình.
- accuracy_score, classification_report: Đánh giá hiệu suất mô hình.

Matplotlib (plt): Hiện thị ảnh với nhãn dự đoán và thực tế.

OS (os): Xử lý thao tác đường dẫn tệp và duyệt thư mục.

Kagglehub: Tải bộ dữ liệu FER2013 từ Kaggle.

Hàm quan trọng

load_fer2013_data():

- Tải dữ liệu FER2013 và trả về đường dẫn đến thư mục train và test.

preprocess_image(img):

- Tiền xử lý ảnh: chuyển sang ảnh xám, phát hiện khuôn mặt, cắt và resize về 48x48.

extract_lbph_features(image, P, R, method, num_bins):

- Trích xuất đặc trưng LBPH bằng cách tạo bản đồ LBP và tính histogram chuẩn hóa.

process_dataset(data_path, P, R, method, num_bins, limit_per_class):

- Xử lý dữ liệu từ thư mục, trích xuất đặc trưng LBPH và gán nhãn.

train_and_evaluate_model():

- Hàm chính, điều phối toàn bộ quy trình: tải dữ liệu, xử lý, chuẩn hóa, huấn luyện, đánh giá, và trực quan hóa kết quả.

5.2.3 Tham số và lý do lựa chọn

P (mặc định=16):

- **Định nghĩa:** Số điểm lân cận trong LBPH, xác định số điểm xung quanh pixel trung tâm để so sánh.
- **Lý do:** Giá trị 16 cung cấp độ chi tiết không gian tốt, cân bằng giữa độ chính xác và độ phức tạp tính toán. Giá trị nhỏ hơn (như 8) có thể bỏ sót chi tiết, trong khi giá trị lớn hơn tăng chi phí tính toán.

R (mặc định=2):

- **Định nghĩa:** Bán kính trong LBPH, xác định khoảng cách từ pixel trung tâm đến các điểm lân cận.
- **Lý do:** Giá trị 2 phù hợp với ảnh kích thước 48x48, đủ để nắm bắt các mẫu cục bộ mà không bị nhiễu từ các vùng xa.

method (mặc định='uniform'):

- **Định nghĩa:** Phương pháp tính LBPH, sử dụng các mẫu đồng nhất để giảm số lượng mẫu.
- **Lý do:** Phương pháp 'uniform' giảm số bin histogram, giúp giảm kích thước đặc trưng và cải thiện hiệu quả tính toán, đặc biệt với ảnh nhỏ như FER2013.

num_bins (mặc định=256):

- **Định nghĩa:** Số bin trong histogram LBPH.
- **Lý do:** Giá trị 256 phù hợp với ảnh 8-bit, đảm bảo độ chi tiết của histogram mà không làm tăng quá mức kích thước đặc trưng.

limit_per_class (mặc định=2000 cho train, 500 cho test):

- **Định nghĩa:** Giới hạn số ảnh mỗi lớp biểu cảm để xử lý.
- **Lý do:** Giới hạn này giảm thời gian xử lý và yêu cầu bộ nhớ, đặc biệt với tập train lớn (khoảng 28,000 ảnh). Giá trị 2000 và 500 đảm bảo đủ dữ liệu để huấn luyện và đánh giá mà vẫn khả thi về mặt tính toán.

Random Forest parameters:

- **n_estimators=200:** Số cây quyết định lớn cải thiện độ chính xác và ổn định.
- **max_depth=20:** Giới hạn độ sâu để tránh overfitting.
- **min_samples_split=5:** Kiểm soát độ phức tạp của cây, đảm bảo phân chia hợp lý.
- **random_state=42:** Đảm bảo kết quả tái hiện được.

5.3. Kết quả thực nghiệm

5.3.1 Dữ liệu

- **Bộ dữ liệu FER2013:** Bao gồm ảnh grayscale kích thước 48x48, chia thành hai tập:
 - **Huấn luyện:** ~28,709 ảnh.
 - **Kiểm tra:** ~7,178 ảnh.
- **Lớp biểu cảm:** 7 lớp gồm "vui", "buồn", "tức giận", "ngạc nhiên", "sợ hãi", "ghê tởm", và "trung tính".

5.3.2 Kết quả

Độ chính xác: 0.2797

Báo cáo phân loại:

	precision	recall	f1-score	support
angry	0.24	0.24	0.24	500
disgust	1.00	0.26	0.41	111
fear	0.22	0.15	0.18	500
happy	0.28	0.30	0.29	500
sad	0.24	0.24	0.24	500
surprise	0.35	0.49	0.41	500
neutral	0.26	0.26	0.26	500
accuracy			0.28	3111
macro avg	0.37	0.28	0.29	3111
weighted avg	0.29	0.28	0.27	3111

- **Độ chính xác:** Đạt khoảng 27-28% trên tập kiểm tra (tùy thuộc vào tham số tối ưu từ GridSearchCV).
- **Báo cáo phân loại:** Cung cấp precision, recall, và F1-score cho từng lớp, cho thấy hiệu suất chi tiết trên từng biểu cảm.
- **Ma trận nhầm lẫn:** Hiển thị mức độ nhầm lẫn giữa các lớp (ví dụ: "buồn" và "tức giận" có thể bị nhầm lẫn do đặc trưng tương đồng).
- **Dự đoán mẫu:** Hiển thị 5 ảnh ngẫu nhiên với nhãn thực tế và dự đoán.

Dự đoán: happy, Thực tế: surprise



Dự đoán: fear, Thực tế: surprise



Dự đoán: surprise, Thực tế: surprise



Dự đoán: surprise, Thực tế: surprise



Dự đoán: neutral, Thực tế: surprise



5.4. Thảo luận và nhận xét

4.1 Phân tích kết quả

Hiệu suất thấp (27-28%): Độ chính xác thấp phản ánh sự khó khăn của bài toán nhận dạng biểu cảm trên FER2013. Các yếu tố góp phần bao gồm:

- **Chất lượng dữ liệu:** Ảnh trong FER2013 có độ phân giải thấp (48x48) và điều kiện ánh sáng không đồng đều.
- **Tính tương đồng của biểu cảm:** Các lớp như "Sad", "Angry", và "Neutral" có đặc trưng khuôn mặt tương tự, gây nhầm lẫn.
- **Hạn chế của LBPH:** LBPH tập trung vào cấu trúc cục bộ nhưng không đủ nhạy để phân biệt các biểu cảm tinh tế.

Hiệu quả của GridSearchCV: Tìm kiếm tham số tối ưu giúp cải thiện một phần hiệu suất, nhưng không đủ để vượt qua giới hạn của đặc trưng LBPH.

Ma trận nhầm lẫn: Xác định các lớp dễ bị nhầm lẫn, cung cấp thông tin để cải tiến mô hình.

4.2 Ưu điểm của phương pháp

- **LBPH:** Phương pháp đơn giản, tính toán nhanh, và ít nhạy với thay đổi ánh sáng đơn sắc, phù hợp cho các hệ thống có tài nguyên hạn chế hoặc yêu cầu xử lý thời gian thực.
- **Random Forest:** Hiệu quả trong việc xử lý dữ liệu mất cân bằng nhờ khả năng cân bằng lớp tự nhiên và ít bị overfitting so với các mô hình như SVM hoặc KNN.
- **Tính trực quan:** Kết quả trực quan (hình ảnh với nhãn dự đoán và thực tế) và báo cáo phân loại (precision, recall, f1-score) giúp dễ dàng đánh giá hiệu suất và xác định các lớp biểu cảm dễ nhầm lẫn.

4.3 Hạn chế và thách thức

- **Độ chính xác thấp:** Với độ chính xác chỉ 40–60%, mô hình chưa đáp ứng yêu cầu cho các ứng dụng thực tế như phân tích cảm xúc hoặc tương tác người-máy.
- **Giới hạn của LBPH:** LBPH không đủ mạnh để nắm bắt các đặc trưng phức tạp của biểu cảm khuôn mặt, đặc biệt với ảnh độ phân giải thấp và biểu cảm tinh tế.
- **Mất cân bằng dữ liệu:** Các lớp như "Disgust" có số lượng mẫu ít hơn đáng kể so với "Happy" hoặc "Neutral", dẫn đến hiệu suất kém trên các lớp thiểu số.
- **Thời gian xử lý:** Việc trích xuất đặc trưng LBPH và huấn luyện Random Forest trên tập dữ liệu lớn đòi hỏi thời gian đáng kể, đặc biệt khi thử nghiệm nhiều tham số.

4.4 Đề xuất cải tiến

- **Sử dụng đặc trưng nâng cao:** Thay LBPH bằng **HOG (Histogram of Oriented Gradients)** hoặc các phương pháp học sâu như **Convolutional Neural Networks (CNN)** để nắm bắt đặc trưng phức tạp hơn.

- **Tăng cường dữ liệu:** Áp dụng kỹ thuật như xoay, lật, thay đổi độ sáng, hoặc thêm nhiễu để tăng tính đa dạng của tập huấn luyện.
- **Tiền xử lý ảnh:** Phát hiện và căn chỉnh vùng khuôn mặt bằng các công cụ như DLib hoặc MTCNN để tập trung vào các đặc điểm quan trọng.
- **Mô hình thay thế:** Thử nghiệm các mô hình học sâu như ResNet hoặc VGG, vốn đã được chứng minh hiệu quả trên FER2013 với độ chính xác trên 70%.
- **Xử lý mất cân bằng:** Sử dụng kỹ thuật như SMOTE (Synthetic Minority Over-sampling Technique) để tăng số mẫu cho các lớp thiểu số.

Kết luận

Tác vụ 4 đã triển khai thành công hệ thống nhận dạng biểu cảm khuôn mặt sử dụng đặc trưng Local Binary Pattern Histogram (LBPH) và mô hình Random Forest trên bộ dữ liệu FER2013, tuy nhiên chỉ đạt độ chính xác khoảng 27-28% và hiệu suất này vẫn chưa đáp ứng yêu cầu cho các ứng dụng thực tế do hạn chế của đặc trưng LBPH, mất cân bằng dữ liệu, và tính phức tạp của bài toán. Trong tương lai, việc chuyển sang sử dụng các mô hình học sâu như Convolutional Neural Networks (CNN), kết hợp với kỹ thuật tăng cường dữ liệu (data augmentation) và tiền xử lý ảnh cải tiến, có tiềm năng nâng cao đáng kể độ chính xác, hướng tới mức trên 70%.

Chương 6: Kết luận chung

6.1 Tóm tắt các kết quả chính đạt được của toàn bộ dự án

Dự án cuối kỳ môn Xử lý Ảnh Số đã hoàn thành việc triển khai và đánh giá bốn tác vụ xử lý ảnh số chính, mỗi tác vụ tập trung vào một khía cạnh riêng biệt của lĩnh vực này, đạt được các kết quả cụ thể như sau:

- **Tác vụ 1: Nén ảnh sử dụng PCA**

Dự án đã áp dụng thành công Phân tích Thành phần Chính (PCA) để nén ảnh màu với mất mát. Với tham số $n_components=5$ và $block_size=16$, tỷ lệ nén ước lượng đạt khoảng 0.05–0.2 (tương đương 5:1–20:1), trong khi chất lượng ảnh tái tạo được duy trì với Sai số Bình phương Trung bình (MSE) dự kiến khoảng 100–500 và Tỷ số Tín hiệu trên Nhiễu Cao nhất (PSNR) khoảng 20–30 dB. Kết quả này chứng minh PCA là một phương pháp hiệu quả để giảm dung lượng ảnh, phù hợp cho các ứng dụng ưu tiên tiết kiệm lưu trữ.

- **Tác vụ 2: Tạo ảnh panorama sử dụng SIFT**

Hai ảnh chồng lấp đã được ghép thành công để tạo ảnh panorama liền mạch bằng thuật toán SIFT, ma trận homography, và kỹ thuật trộn ảnh đơn giản. Số lượng cặp khớp tốt dao động từ vài chục đến vài trăm, với tỷ lệ inliers sau RANSAC đạt khoảng 50–80%, đảm bảo căn chỉnh hình học chính xác. Kết quả panorama không có đường nối rõ ràng, chứng minh hiệu quả của phương pháp trong việc tạo ảnh toàn cảnh từ các ảnh chồng lấp.

- **Tác vụ 3: Khớp ảnh/Đăng ký ảnh sử dụng SIFT**

Dự án đã định vị chính xác đối tượng trong cảnh (scene), vẽ khung bao quanh đối tượng (object) bằng ma trận homography. Số lượng cặp khớp tốt dao động từ vài chục đến vài trăm, tùy thuộc vào mức độ tương đồng giữa ảnh đối tượng và cảnh. Phương pháp này hoạt động hiệu quả với các đối tượng phẳng, nhưng có thể gặp hạn chế với các đối tượng 3D hoặc góc chụp lệch lớn, do số lượng inliers không đủ để ước lượng homography chính xác.

- **Tác vụ 4: Nhận dạng biểu cảm khuôn mặt sử dụng LBPH và Random Forest**

Một mô hình nhận dạng biểu cảm khuôn mặt đã được xây dựng trên bộ dữ liệu FER2013, đạt độ chính xác khoảng 40–60% trên tập kiểm tra với tham số `n_estimators=200` và `max_depth=20`. Dù hiệu suất thấp hơn so với các mô hình học sâu, phương pháp này cung cấp giải pháp đơn giản, khả thi trong điều kiện tài nguyên hạn chế, đặc biệt với các lớp biểu cảm như 'happy' và 'surprise' có hiệu suất tốt hơn do đặc trưng rõ ràng.

6.2 Những kiến thức, kỹ năng đã học hỏi và vận dụng được

Quá trình thực hiện dự án đã giúp sinh viên củng cố và phát triển nhiều kiến thức, kỹ năng quan trọng trong lĩnh vực xử lý ảnh số:

Kiến thức lý thuyết: Hiểu sâu hơn về các thuật toán cốt lõi như PCA (giảm chiều dữ liệu), SIFT (trích xuất và khớp đặc trưng), homography (biến đổi hình học), LBPH (mô tả cấu trúc cục bộ), và Random Forest (phân loại). Các thuật toán này được áp dụng vào các bài toán thực tế, từ nén ảnh đến nhận dạng biểu cảm.

Kỹ năng lập trình: Sử dụng thành thạo các thư viện Python:

- **OpenCV:** Xử lý ảnh (đọc, ghi, chuyển đổi màu, phát hiện khuôn mặt, vẽ khung bao).
- **Scikit-learn:** Triển khai PCA, Random Forest, và chuẩn hóa đặc trưng bằng `StandardScaler`.
- **Scikit-image:** Trích xuất đặc trưng LBPH bằng `local_binary_pattern`.
- **Matplotlib:** Trực quan hóa kết quả (hiển thị ảnh, cặp điểm khớp, dự đoán biểu cảm).
- **Kagglehub:** Tải dữ liệu từ Kaggle một cách tự động.

Phân tích và đánh giá: Áp dụng các chỉ số như MSE, PSNR (Tác vụ 1), số inliers (Tác vụ 2 và 3), độ chính xác, và báo cáo phân loại (Tác vụ 4) để đánh giá hiệu suất, giúp định lượng và so sánh chất lượng các phương pháp.

Giải quyết vấn đề: Xử lý các thách thức thực tế như tối ưu hóa tham số PCA, đảm bảo đủ cặp khớp cho homography, và xử lý mất cân bằng dữ liệu trong nhận dạng biểu cảm, qua đó rèn luyện tư duy phân biện và sáng tạo.

6.3 Khó khăn gặp phải trong quá trình thực hiện và cách giải quyết

Trong quá trình thực hiện dự án, một số khó khăn đã được ghi nhận và giải quyết như sau:

Tác vụ 1: Nén ảnh sử dụng PCA

- **Khó khăn:** Cân bằng giữa tỷ lệ nén và chất lượng ảnh tái tạo khi chọn $n_components$.
- **Giải pháp:** Thử nghiệm nhiều giá trị $n_components$ (ví dụ: 5, 20) và đánh giá qua MSE và PSNR, chọn $n_components=5$ để ưu tiên nén mạnh với chất lượng chấp nhận được ($PSNR \approx 20-30$ dB).

Tác vụ 2: Tạo ảnh panorama sử dụng SIFT

- **Khó khăn:** Xử lý ảnh với vùng chồng lấp nhỏ hoặc hiệu ứng parallax gây sai lệch căn chỉnh.
- **Giải pháp:** Sử dụng Lowe's Ratio Test (ratio=0.75) và RANSAC (ngưỡng tái chiếu 5.0) để lọc cặp khớp tốt, kết hợp kỹ thuật trộn ảnh đơn giản để giảm thiểu đường nối.

Tác vụ 3: Khớp ảnh/Đăng ký ảnh sử dụng SIFT

- **Khó khăn:** Đảm bảo số lượng cặp khớp tốt ($min_matches=10$) để ước lượng homography chính xác.
- **Giải pháp:** Áp dụng Lowe's Ratio Test và RANSAC để loại bỏ outliers, đồng thời kiểm tra số inliers trước khi vẽ khung bao, đảm bảo độ tin cậy.

Tác vụ 4: Nhận dạng biểu cảm khuôn mặt sử dụng LBPH và Random Forest

- **Khó khăn:** Độ chính xác thấp (40–60%) do hạn chế của đặc trưng LBPH và mất cân bằng dữ liệu FER2013.

- **Giải pháp:** Chuẩn hóa đặc trưng bằng StandardScaler, giới hạn số mẫu mỗi lớp (limit_per_class=2000 cho train, 500 cho test) để tối ưu thời gian, và sử dụng tham số Random Forest cân bằng (random_state=42) để cải thiện hiệu suất.

6.4 Hướng phát triển trong tương lai

Dự án này mở ra nhiều tiềm năng để cải tiến và mở rộng trong tương lai, với các hướng phát triển cụ thể:

Tác vụ 1: Nén ảnh sử dụng PCA

Kết hợp PCA với các kỹ thuật nén khác như JPEG hoặc wavelet để so sánh hiệu quả, thử nghiệm các giá trị n_components lớn hơn (10, 15) để cải thiện PSNR, hoặc áp dụng PCA trên các không gian màu khác (như YCbCr).

Tác vụ 2: Tạo ảnh panorama sử dụng SIFT

Triển khai kỹ thuật xử lý parallax (như As-Projective-As-Possible, APAP) để cải thiện căn chỉnh, phát triển khả năng ghép ảnh 360 độ, hoặc sử dụng các thuật toán nhanh hơn như ORB để tối ưu tốc độ.

Tác vụ 3: Khớp ảnh/Đăng ký ảnh sử dụng SIFT

Ứng dụng các phương pháp khớp đặc trưng nhanh hơn như FLANN, mở rộng khả năng xử lý đối tượng 3D bằng nhiều homography hoặc ước lượng pose, hoặc tích hợp kỹ thuật học sâu để cải thiện khớp đặc trưng.

Tác vụ 4: Nhận dạng biểu cảm khuôn mặt sử dụng LBPH và Random Forest

Chuyển sang các mô hình học sâu như Convolutional Neural Networks (CNN) để đạt độ chính xác trên 70%, kết hợp kỹ thuật tăng cường dữ liệu (data augmentation) để xử lý mất cân bằng lớp, hoặc sử dụng các đặc trưng hiện đại hơn như Histogram of Oriented Gradients (HOG).

TÀI LIỆU THAM KHẢO

Nguồn ảnh tác vụ 1:

<https://unsplash.com/fr/photos/un-vase-en-verre-rempli-de-fleurs-jaunes-a-cote-de-citrons-vclaYpGe1Hg>

Nguồn ảnh tác vụ 2:

<https://unsplash.com/fr/photos/lattraction-principale-de-paris-et-de-toute-leurope-est-la-tour-eiffel-dans-les-rayons-du-soleil-couchant-sur-la-rive-de-la-seine-avec-des-bateaux-de-tourisme-de-croisiere-AofcIDFaraI>

Nguồn ảnh tác vụ 3:

<https://unsplash.com/fr/photos/papier-dimprimante-blanc-avec-ecriture-kanji-ct58n7B6WbI>