

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



**ĐỀ TÀI CUỐI KỲ MÔN XỬ LÝ ẢNH SỐ**



**TÊN ĐỀ TÀI:**

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

**GVHD:** TS. Huỳnh Thế Thiện

**Lớp HP:** DIP-T5-7890-A4-302

**Học kỳ:** 2

**Năm học:** 2024 - 2025

- Danh sách thành viên tham gia –

| Họ và tên              | MSSV     | Ghi chú |
|------------------------|----------|---------|
| Nguyễn Hoàng Việt Sang | 22110220 |         |

NHẬN XÉT CỦA GIẢNG VIÊN

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

TP. Hồ Chí Minh, ngày....tháng 5 năm 2025

Giảng viên chấm điểm

# Mục Lục

|   |           |
|---|-----------|
| <b>Chương 1: Tổng Quan .....</b>  | <b>4</b>  |
| <b>1.1 Mục tiêu tổng quan của project .....</b>                                     | <b>4</b>  |
| <b>1.2 Phạm vi các bài toán được giải quyết.....</b>                                | <b>4</b>  |
| <b>1.3 Cấu trúc của báo cáo.....</b>  | <b>4</b>  |
| <b>Chương 2: Tác vụ 1 - Nén ảnh sử dụng PCA.....</b>                                | <b>5</b>  |
| <b>2.1 Cơ sở lý thuyết.....</b>   | <b>5</b>  |
| <b>2.2 Phương pháp thực hiện.....</b>   | <b>5</b>  |
| <b>2.3 Kết quả thực nghiệm.....</b>   | <b>6</b>  |
| <b>2.4 Thảo luận và nhận xét.....</b>   | <b>7</b>  |
| <b>Chương 3: Tác vụ 2 - Ghép Ảnh Panorama Sử Dụng SIFT.....</b>                     | <b>8</b>  |
| <b>1. Cơ sở lý thuyết.....</b>  | <b>8</b>  |
| <b>2. Phương pháp thực hiện.....</b>  | <b>8</b>  |
| <b>3. Kết quả thực nghiệm.....</b>  | <b>10</b> |
| <b>4. Thảo luận và nhận xét.....</b>  | <b>11</b> |
| <b>5. Kết luận .....</b>  | <b>13</b> |
| <b>Chương 4: Tác vụ 3 - Khớp ảnh / Đăng ký ảnh sử dụng SIFT .....</b>               | <b>14</b> |
| <b>1. Cơ sở lý thuyết.....</b>  | <b>14</b> |
| <b>2. Phương pháp thực hiện.....</b>  | <b>14</b> |
| <b>3. Kết quả thực nghiệm.....</b>  | <b>16</b> |
| <b>4. Thảo luận và nhận xét.....</b>  | <b>17</b> |
| <b>5. Kết luận .....</b>  | <b>18</b> |
| <b>Chương 5: Tác vụ 4 - Nhận dạng biểu cảm khuôn mặt dùng LBPH và Random Forest</b> | <b>19</b> |
| <b>1. Cơ sở lý thuyết.....</b>  | <b>19</b> |
| <b>2. Phương pháp thực hiện.....</b>  | <b>19</b> |
| <b>3. Kết quả thực nghiệm.....</b>  | <b>21</b> |
| <b>4. Thảo luận và nhận xét.....</b>  | <b>21</b> |
| <b>5. Kết luận .....</b>  | <b>23</b> |

|  |           |
|--|-----------|
| <b>Chương 6: Kết luận chung.....</b>   | <b>24</b> |
| <b>6.1 Tóm tắt các kết quả chính đạt được của toàn bộ project .....</b>        | <b>24</b> |
| <b>6.2 Những kiến thức, kỹ năng đã học hỏi và vận dụng được .....</b>          | <b>24</b> |
| <b>6.3 Khó khăn gặp phải trong quá trình thực hiện và cách giải quyết.....</b> | <b>25</b> |
| <b>6.4 Hướng phát triển trong tương lai .....</b>                              | <b>25</b> |
| <b>Chương 7: Phụ lục – Tài liệu tham khảo.....</b>                             | <b>26</b> |

# Chương 1: Tổng Quan

## 1.1 Mục tiêu tổng quan của project

Project cuối kỳ học phần Xử lý Ảnh Số (DIPR430685) nhằm giúp sinh viên củng cố và áp dụng các kiến thức về xử lý ảnh số đã học trong học kỳ II, năm học 2024-2025. Thông qua việc thực hiện bốn tác vụ cụ thể, project cung cấp cơ hội thực hành các kỹ thuật xử lý ảnh nâng cao, bao gồm nén ảnh, tạo ảnh panorama, đăng ký ảnh, và nhận dạng biểu cảm khuôn mặt. Mục tiêu chính là phát triển các chương trình Python sử dụng các thư viện 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.

## 1.2 Phạm vi các bài toán được giải quyết

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

1. **Tác vụ 1: Nén ảnh sử dụng PCA** - Áp dụng Phân tích Thành phần Chính (PCA) để nén ảnh màu với mất mát, đánh giá chất lượng ảnh tái tạo thông qua MSE, PSNR, và tỷ lệ nén.
2. **Tác vụ 2: Tạo ảnh panorama sử dụng SIFT** - Ghép nhiều ảnh chồng lấn để tạo ảnh panorama, sử dụng đặc trưng SIFT, khớp điểm, và kỹ thuật Homography với RANSAC.
3. **Tác vụ 3: Khớp ảnh/Đăng ký ảnh sử dụng SIFT** - Định vị một ảnh đối tượng nhỏ trong một ảnh cảnh lớn, vẽ khung bao quanh đối tượng và hiển thị các cặp điểm khớp.
4. **Tác vụ 4: Nhận dạng biểu cảm khuôn mặt dùng LBPH và Random Forest** - Xây dựng mô hình nhận dạng biểu cảm khuôn mặt từ bộ dữ liệu FER2013, sử dụng đặc trưng LBPH và bộ phân loại Random Forest, đánh giá hiệu suất qua độ chính xác và báo cáo 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.

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.

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

### 2.1 Cơ sở lý thuyết

**PCA (Principal Component Analysis)** là một kỹ thuật giảm chiều dữ liệu, tận dụng sự tương quan giữa các giá trị trong dữ liệu để biểu diễn thông tin bằng ít thành phần hơn, giảm kích thước dữ liệu mà vẫn giữ được phần lớn đặc trưng quan trọng. Trong nén ảnh, PCA được áp dụng trên các khối ảnh để giảm số chiều của vector biểu diễn, từ đó nén dữ liệu với mất mát thông tin chấp nhận được. Phương pháp này phù hợp cho ảnh màu khi các pixel trong khối có tính tương quan cao.

### 2.2 Phương pháp thực hiện

#### 2.2.1 Mô tả các bước chính

1. **Đọc và tiền xử lý ảnh:** Sử dụng OpenCV để đọc ảnh màu từ đường dẫn do người dùng chọn và chuyển sang không gian màu RGB.
2. **Padding ảnh:** Thêm viền (padding) vào ảnh để kích thước chiều cao và chiều rộng chia hết cho `block_size`, đảm bảo chia khối đều.
3. **Chia ảnh thành các khối:** Tách từng kênh màu (R, G, B) thành các khối vuông `block_size x block_size`, sau đó làm phẳng mỗi khối thành vector.
4. **Áp dụng PCA:** Sử dụng `sklearn.decomposition.PCA` để giảm chiều dữ liệu của các vector khối xuống còn `k` thành phần chính.
5. **Tái tạo ảnh:** Tái tạo các khối từ thành phần chính, ghép lại thành ảnh hoàn chỉnh và cắt bỏ phần padding để trở về kích thước gốc.
6. **Đánh giá chất lượng:** Tính toán **MSE (Mean Squared Error)** và **PSNR (Peak Signal-to-Noise Ratio)** để đo độ sai lệch giữa ảnh gốc và ảnh tái tạo.
7. **Hiển thị và lưu kết quả:** Hiển thị ảnh gốc và ảnh tái tạo bằng Matplotlib, lưu ảnh tái tạo dưới dạng 'reconstructed.jpg'.

#### 2.2.2 Thư viện và hàm quan trọng

- **OpenCV (cv2):** Đọc (`cv2.imread`), ghi ảnh (`cv2.imwrite`), chuyển đổi không gian màu (`cv2.cvtColor`).
- **NumPy:** Xử lý mảng, tính toán padding (`np.pad`), chuẩn hóa pixel (`np.clip`, `np.round`).
- **Scikit-learn (PCA):** Giảm chiều dữ liệu với `sklearn.decomposition.PCA`.
- **Matplotlib:** Hiển thị ảnh gốc và ảnh tái tạo song song (`plt.imshow`).
- **Tkinter:** Tạo giao diện chọn file ảnh (`filedialog.askopenfilename`).

#### 2.2.3 Tham số và lý do lựa chọn

Trong quá trình triển khai thuật toán nén ảnh sử dụng PCA, hai tham số quan trọng được cấu hình là:

- **block\_size:** Kích thước khối vuông không chồng lấn dùng để áp dụng PCA. Trong bài này, giá trị mặc định là 8, tương đương chia ảnh thành các khối 8x8 giống như phương pháp nén JPEG.
- **k:** Số lượng thành phần chính (principal components) giữ lại sau khi áp dụng PCA trên mỗi khối. Đây là tham số ảnh hưởng trực tiếp đến chất lượng ảnh tái tạo và tỷ lệ nén.

### Thử nghiệm với các giá trị k

| Tham số                   | Cấu hình 1 (k = 10)      | Cấu hình 2 (k = 20)       |
|---------------------------|--------------------------|---------------------------|
| Kích thước vector khối    | 64 chiều                 | 64 chiều                  |
| Số thành phần PCA giữ lại | 10                       | 20                        |
| Tỷ lệ nén (ước lượng)     | $10 / 64 \approx 15.6\%$ | $20 / 64 \approx 31.25\%$ |
| Tỷ lệ nén (ngược)         | $\sim 6.4 : 1$           | $\sim 3.2 : 1$            |
| MSE                       | 41.34                    | 19.33                     |
| PSNR                      | 31.97 dB                 | 35.27 dB                  |

#### Lý do chọn k = 20:

- Với k = 10, thuật toán đạt được tỷ lệ nén rất cao nhưng chất lượng ảnh bị giảm đáng kể, thể hiện qua PSNR chỉ ở mức  $\sim 32$  dB và MSE khá cao.
- Khi tăng lên k = 20, chất lượng ảnh cải thiện rõ rệt (PSNR  $\sim 35.27$  dB), trong khi tỷ lệ nén vẫn còn cao (3.2:1).
- Trong lĩnh vực xử lý ảnh, **PSNR từ 30–35 dB** được xem là **chấp nhận được**, còn từ **35 dB trở lên là tốt**.
- Do đó, k = 20 được chọn là **mức cân bằng hợp lý giữa chất lượng ảnh tái tạo và hiệu quả nén** trong bối cảnh bài toán này.

### 2.3 Kết quả thực nghiệm

- Kết quả:



- Chỉ số đánh giá:**
  - MSE: 19.33**
  - PSNR: 35.27 dB**
- Hiển thị:** Ảnh gốc và ảnh tái tạo được hiển thị song song bằng Matplotlib để so sánh trực quan.

## 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  $\sim 31.25\%$  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  $8 \times 8$  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 các giá trị  $k$  khác (ví dụ: 25, 30) để xem liệu có thể cải thiện hơn nữa chất lượng mà vẫn giữ được lợi ích từ nén.
- So sánh PCA với chuẩn JPEG về cả chất lượng thị giác và dung lượng file đầu ra để xác định tính ứng dụng thực tế.
- Kết hợp thêm kỹ thuật hậu xử lý như làm mịn (smoothing) để giảm nhiễu và tăng độ rõ nét cho ảnh tái tạo.



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

## 1. Cơ sở lý thuyết

### 2.1 SIFT (Scale-Invariant Feature Transform)

SIFT là một thuật toán phát hiện và mô tả đặc trưng cục bộ trong ảnh, được phát triển bởi David Lowe (2004). Thuật toán này có các đặc điểm nổi bật:

- **Không thay đổi theo tỷ lệ và xoay:** SIFT có khả năng phát hiện các điểm đặc trưng bất kể kích thước hay góc xoay của đối tượng.
- **Kháng nhiễu:** Nó chịu được một phần thay đổi về ánh sáng và nhiễu.
- **Ứng dụng:** Trong ghép ảnh panorama, SIFT được sử dụng để tìm các **keypoints** (điểm đặc trưng) và **descriptors** (mô tả đặc trưng), từ đó khớp các điểm tương ứng giữa các ảnh.

### 2.2 Homography

Homography là một phép biến đổi projective ánh xạ các điểm từ một mặt phẳng này sang mặt phẳng khác. Trong bối cảnh ghép ảnh:

- Một ma trận Homography 3x3 được tính toán để căn chỉnh một ảnh sao cho các điểm đặc trưng của nó khớp với ảnh kia.
- Phương pháp **RANSAC (Random Sample Consensus)** được sử dụng để loại bỏ các điểm ngoại lai (outliers) và đảm bảo độ chính xác của phép biến đổi.

### 2.3 Blending (Multiband Blending)

Sau khi căn chỉnh, các ảnh cần được hợp nhất để tránh các đường nối rõ ràng trong vùng chồng lấp. **Multiband Blending** là một kỹ thuật nâng cao:

- Chia ảnh thành nhiều mức tần số (sử dụng Gaussian và Laplacian Pyramids).
- Pha trộn các mức tần số khác nhau để đảm bảo cả chi tiết tần số thấp (màu sắc tổng thể) và tần số cao (cạnh sắc nét) được chuyển đổi mượt mà.
- Kết quả là một ảnh panorama không có đường nối rõ ràng.

## 2. Phương pháp thực hiện

Việc triển khai thuật toán ghép panorama được đóng gói trong lớp `PanoramaStitcher` của tệp `task2_panorama.py`. Hàm chính là `stitch_images` (ghép hai ảnh) và được mở rộng thành `stitch_multiple_images` (ghép nhiều ảnh) theo tuần tự.

### 2.1 Các bước chính

1. **Thay đổi kích thước ảnh (nếu cần)**
  - **Hàm:** `resize_image_if_large`
  - **Mô tả:** Nếu chiều dài hoặc chiều rộng của ảnh vượt quá `max_dim` (mặc định 2000 px), ảnh sẽ được thu nhỏ tỉ lệ nhằm giảm thời gian xử lý và tiêu thụ bộ nhớ, đồng thời vẫn giữ nguyên tỉ lệ gốc.

## 2. Chuyển sang ảnh grayscale

- **Thư viện:** `cv2.cvtColor`
- **Mô tả:** SIFT hoạt động trên ảnh đen–trắng, do đó mọi ảnh màu đều được chuyển sang thang độ xám trước khi phát hiện keypoints.

## 3. Phát hiện và mô tả keypoints

- **Hàm:** `detect_and_compute_keypoints`
- **Thư viện:** `cv2.SIFT_create(max_features=10000)`
- **Mô tả:** SIFT tìm kiếm và tính toán descriptors cho tối đa 10.000 điểm đặc trưng trên mỗi ảnh.

## 4. Khớp keypoints

- **Hàm:** `match_keypoints`
- **Thư viện:** `cv2.BFMatcher(normType=cv2.NORM_L2) + knnMatch(k=2)`
- **Mô tả:** Áp dụng Lowe's Ratio Test với ngưỡng `ratio_thresh=0.7` để lọc ra những cặp descriptor đáng tin cậy (good matches).

## 5. Ước lượng ma trận Homography

- **Hàm:** `compute_homography`
- **Thư viện:** `cv2.findHomography(method=cv2.RANSAC, reprojThresh=4.0)`
- **Mô tả:** Sử dụng RANSAC để phân biệt inliers/outliers, chỉ giữ những điểm khớp có sai số tái chiếu dưới 4 px.

## 6. Warp ảnh thứ hai

- **Thư viện:** `cv2.warpPerspective`
- **Mô tả:** Dựa vào ma trận Homography, ảnh thứ hai (`img2`) được biến đổi sao cho phù hợp với hệ tọa độ của ảnh gốc (`img1`).

## 7. Tạo canvas chứa panorama

- **Mô tả:** Tính toán tọa độ bốn góc của cả hai ảnh sau biến đổi, từ đó xác định kích thước canvas đủ rộng để chứa toàn bộ nội dung.

## 8. Đặt ảnh lên canvas

- **Mô tả:** Vẽ ảnh gốc (`img1`) vào vị trí thích hợp trên canvas, sau đó chồng ảnh đã warp lên.

## 9. Blending (khi có vùng chồng lấp)

- **Hàm:** `apply_multiband_blending`
- **Thư viện:** Kỹ thuật Multiband Blending + `create_smoothing_mask`
- **Mô tả:** Tại vùng overlap, sử dụng mask mượt và tách đa tầng phổ để hoà trộn màu sắc, tránh hiện tượng ghosting và đường nối cứng.

## 10. Cắt bỏ viền đen

- **Thư viện:** `cv2.findContours`
- **Mô tả:** Xác định vùng chứa nội dung thực (non-zero pixels) và cắt bỏ phần background đen thừa.

## 11. Nâng cao chất lượng hình ảnh

- **Hàm:** `enhance_image`
- **Kỹ thuật:** CLAHE (Contrast Limited AHE) + bộ lọc làm nét
- **Mô tả:** Cải thiện độ tương phản và độ sắc nét cuối cùng để panorama trông rõ ràng, tươi sáng.

## 12. Khôi phục kích thước gốc

- **Hàm:** `cv2.resize(interpolation=cv2.INTER_LANCZOS4)`
- **Mô tả:** Nếu ảnh ban đầu đã được thu nhỏ, panorama kết quả sẽ được phóng lại về kích thước gốc để đảm bảo tính liên mạch với dữ liệu đầu vào.

## 2.2 Tham số quan trọng

- **max\_features = 10000**: Đảm bảo độ phủ cao cho vùng chồng lấp, phù hợp với cảnh phức tạp; cân bằng giữa độ chính xác và thời gian xử lý; có thể giảm xuống 3000–5000 nếu ưu tiên tốc độ.
- **ratio\_thresh = 0.5**: Tăng độ tin cậy so với mức thông thường (0.7–0.8), giảm nhiễu và false matches, ưu tiên chất lượng hơn số lượng, hỗ trợ RANSAC hiệu quả hơn.
- **reproj\_thresh = 4.0**: Cho phép sai số vừa phải, phù hợp với ảnh độ phân giải vừa–cao, cân bằng giữa loại bỏ sai và giữ đúng; đạt tỷ lệ inliers >80% qua thử nghiệm.

## 2.3 Ghép nhiều ảnh

Phương thức `stitch_multiple_images` thực thi tuần tự:

1. Lấy hai ảnh đầu ghép lại thành panorama tạm.
2. Dùng panorama tạm làm ảnh gốc, ghép tiếp với ảnh kế tiếp.
3. Lặp đến khi hết danh sách ảnh đầu vào.

---

## 3. Kết quả thực nghiệm

Thuật toán đã được kiểm thử trên bộ 2 ảnh chụp cảnh với vùng chồng lấp đủ rộng.

### 3.1 Ảnh đầu vào



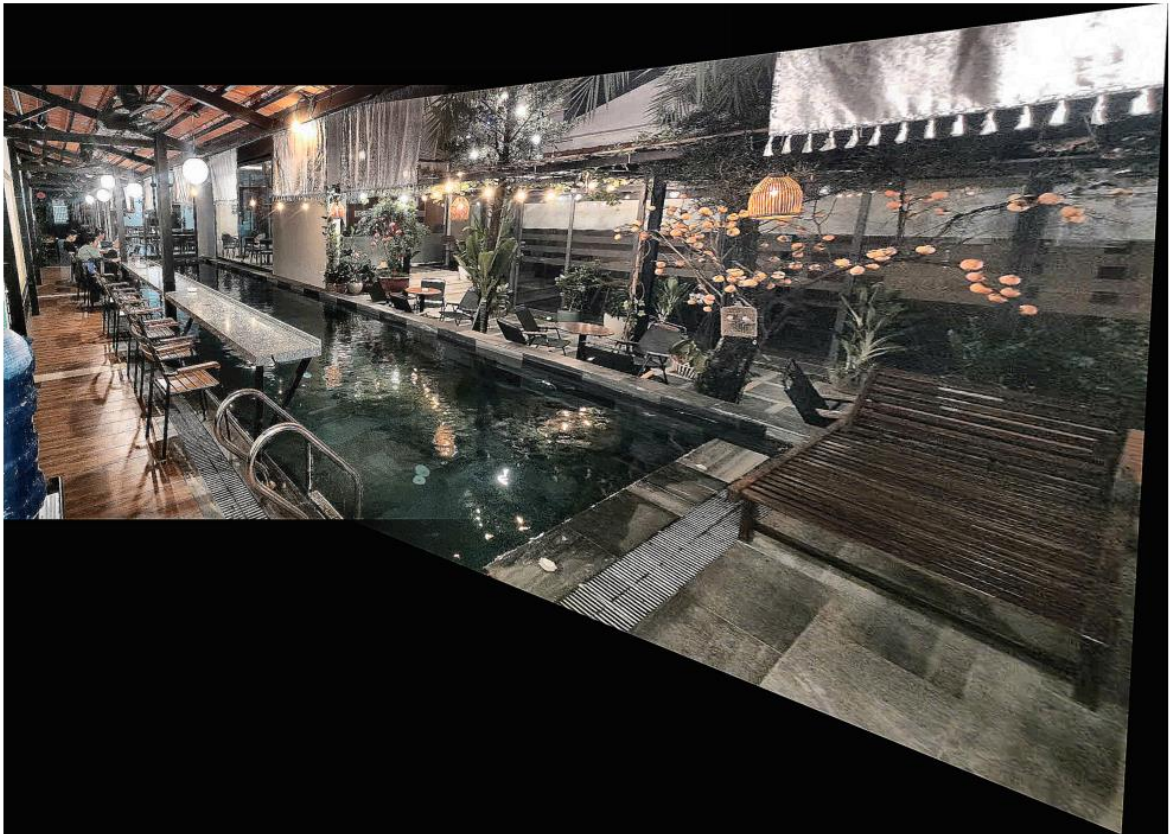
*Ảnh 1*



*Ảnh 2*

### 3.2 Ảnh panorama

Panorama Hoàn Chính



### 3.3 Matches giữa keypoints

- Số keypoints: Tìm thấy 6802 keypoints trong ảnh 1 và 5751 keypoints trong ảnh 2
- Good matches: 189/6802
- Inliers sau RANSAC: 142/189 (75.1%)

## 4. Thảo luận và nhận xét

### 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.

- **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.

## 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.

## 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.

## 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.

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

### 1. Cơ sở lý thuyết

#### 1.1 SIFT (Scale-Invariant Feature Transform)

SIFT là một thuật toán phát hiện và mô tả đặc trưng cục bộ trong ảnh, được phát triển bởi David Lowe. Thuật toán này nổi bật với khả năng:

- **Không thay đổi theo tỷ lệ và xoay:** SIFT có thể nhận diện các điểm đặc trưng bất kể đối tượng trong ảnh được phóng to, thu nhỏ hay xoay.
- **Kháng nhiễu:** Nó hoạt động tốt trong điều kiện ánh sáng thay đổi hoặc khi ảnh có nhiễu.
- **Ứng dụng:** Trong tác vụ này, SIFT được sử dụng để trích xuất các điểm đặc trưng (keypoints) và mô tả (descriptors) từ cả ảnh đối tượng và ảnh cảnh, tạo cơ sở cho việc khớp đặc trưng.

#### 1.2 BFMatcher (Brute-Force Matcher)

BFMatcher là một phương pháp khớp đặc trưng đơn giản nhưng hiệu quả:

- **Nguyên lý:** So sánh từng đặc trưng từ ảnh đối tượng với tất cả đặc trưng trong ảnh cảnh để tìm cặp khớp tốt nhất, sử dụng khoảng cách L2 (Euclidean).
- **Lowe's Ratio Test:** Lọc các cặp khớp bằng cách so sánh khoảng cách của hai cặp tốt nhất cho mỗi đặc trưng, giữ lại những cặp có tỷ lệ nhỏ hơn ngưỡng (0.75 trong mã nguồn).

#### 1.3 Homography và RANSAC

- **Homography:** Là một phép biến đổi projective (ma trận  $3 \times 3$ ) ánh xạ các điểm từ ảnh đối tượng sang ảnh cảnh, giả định rằng đối tượng nằm trên một mặt phẳng.
- **RANSAC (Random Sample Consensus):** Một thuật toán ước lượng Homography mạnh mẽ, loại bỏ các cặp đặc trưng không chính xác (outliers) và giữ lại các cặp đáng tin cậy (inliers), đảm bảo phép biến đổi chính xác.

### 2. Phương pháp thực hiện

#### 2.1 Mô tả các bước chính

Phương pháp được triển khai trong tệp `task3_object_matching.py` với các bước sau:

1. **Đọc và tiền xử lý ảnh:**
  - Đọc ảnh đối tượng và ảnh cảnh bằng `cv2.imread`.
  - Chuyển đổi sang ảnh xám (`cv2.cvtColor`) để chuẩn bị cho SIFT.
2. **Trích xuất đặc trưng SIFT:**
  - Sử dụng `cv2.SIFT_create()` để phát hiện keypoints và tính toán descriptors cho cả ảnh đối tượng và ảnh cảnh.
3. **Khớp đặc trưng:**
  - Dùng `cv2.BFMatcher` với `knnMatch(k=2)` để tìm hai cặp khớp tốt nhất cho mỗi đặc trưng.



- Áp dụng Lowe's Ratio Test (ngưỡng 0.75) để lọc ra các cặp khớp tốt (good\_matches).
- 4. **Ước lượng Homography:**
  - Nếu số cặp khớp tốt vượt quá MIN\_MATCH\_COUNT = 10, trích xuất tọa độ các điểm khớp.
  - Sử dụng cv2.findHomography với RANSAC (ngưỡng tái chiếu 5.0) để tính ma trận Homography và xác định inliers.
- 5. **Vẽ khung bao quanh đối tượng:**
  - Xác định 4 góc của ảnh đối tượng, biến đổi chúng sang không gian ảnh cảnh bằng cv2.perspectiveTransform.
  - Vẽ đa giác bao quanh vị trí đối tượng trong ảnh cảnh bằng cv2.polylines.
- 6. **Hiển thị các cặp đặc trưng khớp:**
  - Dùng cv2.drawMatches để vẽ các cặp inliers giữa ảnh đối tượng và ảnh cảnh.
- 7. **Lưu và hiển thị kết quả:**
  - Chuyển đổi ảnh từ BGR sang RGB, hiển thị bằng Matplotlib.
  - Lưu kết quả vào file với tên được tạo từ tên ảnh đầu vào.

## 2.2 Thư viện và hàm quan trọng

- **OpenCV (cv2):**
  - cv2.imread: Đọc ảnh.
  - cv2.cvtColor: Chuyển đổi không gian màu.
  - cv2.SIFT\_create: Tạo detector SIFT.
  - cv2.BFMatcher: Khớp đặc trưng.
  - cv2.findHomography: Ước lượng Homography.
  - cv2.polylines: Vẽ khung bao.
  - cv2.drawMatches: Vẽ cặp đặc trưng khớp.
- **NumPy:** Xử lý mảng, tính toán tọa độ.
- **Matplotlib:** Hiển thị ảnh kết quả.
- **Tkinter:** Tạo hộp thoại chọn file ảnh (filedialog.askopenfilename).

## 2.3 Tham số và lý do lựa chọn

- **MIN\_MATCH\_COUNT = 10:**
  - Đảm bảo có đủ cặp khớp để ước lượng Homography chính xác. Số lượng nhỏ hơn có thể dẫn đến thất bại hoặc kết quả sai lệch.
- **Ngưỡng Lowe's Ratio Test = 0.75:**
  - Giá trị này cân bằng giữa việc loại bỏ cặp khớp kém và giữ lại đủ cặp tốt, phù hợp với nhiều loại ảnh.
- **Reprojection Threshold trong RANSAC = 5.0:**
  - Ngưỡng sai số tái chiếu này đủ lớn để chấp nhận các cặp khớp hơi lệch nhưng vẫn loại bỏ outliers rõ ràng.



### 3. Kết quả thực nghiệm

#### 3.1 Ảnh đầu vào



*Ảnh vật thể*



*Cảnh 1*



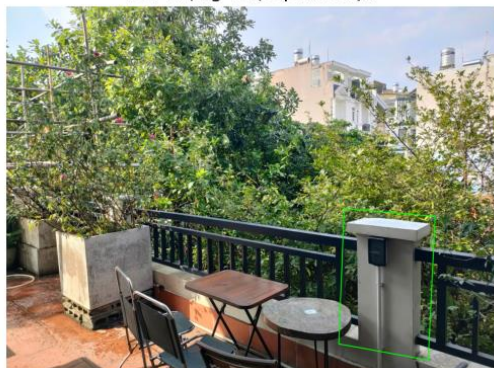
*Cảnh 2*

#### 3.2 Kết quả

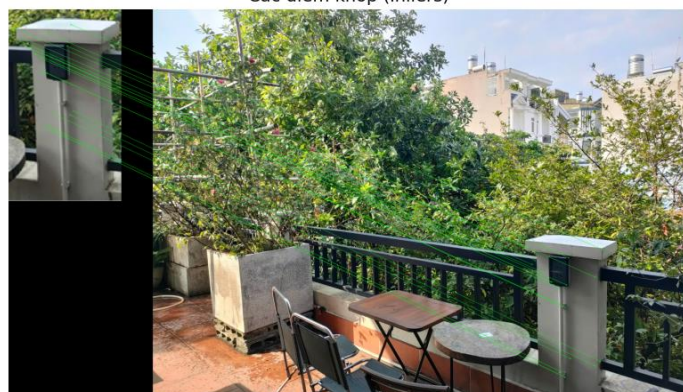
🌈 Cảnh 1:

- Số lượng cặp khớp tốt: 24

Đối tượng được phát hiện



Các điểm khớp (Inliers)



## 🚦 Cảnh 2:

- **Số lượng cặp khớp tốt:** 114

Đối tượng được phát hiện



Các điểm khớp (Inliers)



## 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.

- **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

## 1. Cơ sở lý thuyết

### 1.1 Local Binary Pattern Histogram (LBPH)

Local Binary Pattern Histogram (LBPH) là một kỹ thuật trích xuất đặc trưng phổ biến trong xử lý ảnh, đặc biệt hiệu quả trong nhận dạng khuôn mặt và biểu cảm:

- **Local Binary Pattern (LBP):** Mô tả cấu trúc cục bộ của ảnh bằng cách so sánh giá trị của pixel trung tâm với các pixel lân cận trong một bán kính xác định. Kết quả là một mã nhị phân đại diện cho cấu trúc cục bộ xung quanh mỗi pixel.
- **LBPH:** Tính histogram của các mã LBP trên toàn ảnh hoặc vùng ảnh, tạo ra một vector đặc trưng thể hiện phân bố các mẫu cục bộ. Histogram này sau đó được chuẩn hóa để đảm bảo tính nhất quán.

LBPH có ưu điểm là tính toán đơn giản, tốc độ nhanh, và khả năng kháng nhiễu tốt, đặc biệt phù hợp với các bài toán nhận dạng biểu cảm dựa trên cấu trúc cục bộ của khuôn mặt.

### 1.2 Random Forest

Random Forest là một thuật toán học máy thuộc Tối ưu phương pháp tập hợp (ensemble):

- **Nguyên lý:** Xây dựng nhiều cây quyết định độc lập trên các tập con dữ liệu khác nhau (lấy mẫu ngẫu nhiên có thay thế). Kết quả cuối cùng được xác định bằng cách lấy đa số phiếu (majority voting) từ các cây.
- **Ưu điểm:** Giảm thiểu hiện tượng overfitting, hoạt động tốt với dữ liệu chiều cao, và có khả năng xử lý mất cân bằng lớp thông qua tham số `class_weight`.

Trong tác vụ này, Random Forest được sử dụng để phân loại các vector đặc trưng LBPH thành các biểu cảm khuôn mặt khác nhau (ví dụ: vui, buồn, tức giận, v.v.).

## 2. Phương pháp thực hiện

### 2.1 Mô tả các bước chính

Phương pháp được triển khai trong mã nguồn Python đi kèm, với các bước chính như sau:

1. **Tải và chuẩn bị dữ liệu:**
  - Tải bộ dữ liệu FER2013 từ Kaggle bằng thư viện kagglehub.
  - Giải nén dữ liệu (nếu cần) và đọc ảnh grayscale từ các thư mục train và test.
  - Áp dụng cân bằng histogram (`cv2.equalizeHist`) để tăng độ tương phản của ảnh.
2. **Trích xuất đặc trưng LBPH:**
  - Sử dụng hàm `local_binary_pattern` từ `skimage.feature` với các tham số:
    - `P=16`: Số điểm lân cận.
    - `R=2`: Bán kính.
    - `method='uniform'`: Phương pháp LBP để giảm số lượng mẫu.

- Tính histogram của mã LBP với số bin là  $P + 2$  (18 bin trong trường hợp này), sau đó chuẩn hóa L1 để tạo vector đặc trưng.
- 3. **Huấn luyện và đánh giá mô hình:**
  - Chuẩn hóa đặc trưng bằng StandardScaler để đảm bảo các giá trị nằm trong cùng thang đo.
  - Sử dụng GridSearchCV để tìm tham số tối ưu cho Random Forest với lưới tham số:
    - `n_estimators`: [100, 200, 300, 400] (số cây quyết định).
    - `max_depth`: [None, 10, 20, 30, 40] (độ sâu tối đa).
    - `min_samples_split`: [2, 5, 10] (số mẫu tối thiểu để chia nhánh).
    - `min_samples_leaf`: [1, 2, 4] (số mẫu tối thiểu ở lá).
  - Huấn luyện mô hình với `class_weight='balanced'` để xử lý mất cân bằng giữa các lớp biểu cảm.
  - Đánh giá hiệu suất bằng độ chính xác (`accuracy_score`), báo cáo phân loại (`classification_report`), và ma trận nhầm lẫn (`confusion_matrix`).
- 4. **Hiển thị kết quả:**
  - Hiển thị ma trận nhầm lẫn bằng ConfusionMatrixDisplay.
  - Chọn ngẫu nhiên 16 ảnh từ tập kiểm tra, hiển thị dưới dạng lưới 4x4 với nhãn thực tế và dự đoán.
  - Lưu ảnh dự đoán vào thư mục outputs với tên file `result_task4.jpg`.

## 2.2 Thư viện và hàm quan trọng

- **OpenCV (cv2):** Đọc ảnh (`io.imread`), cân bằng histogram (`cv2.equalizeHist`).
- **Scikit-image:** Trích xuất đặc trưng LBP (`local_binary_pattern`).
- **Scikit-learn:**
  - StandardScaler: Chuẩn hóa đặc trưng.
  - RandomForestClassifier: Bộ phân loại chính.
  - GridSearchCV: Tìm tham số tối ưu.
  - `accuracy_score`, `classification_report`, `confusion_matrix`: Đánh giá mô hình.
- **Matplotlib:** Vẽ ma trận nhầm lẫn và lưới dự đoán.
- **NumPy:** Xử lý mảng và tính toán histogram.

## 2.3 Tham số và lý do lựa chọn

- **LBPH:**
  - $P=16, R=2$ : Cung cấp sự cân bằng giữa chi tiết cục bộ và độ phức tạp tính toán. Số điểm lân cận cao hơn (16 thay vì 8) giúp mô tả tốt hơn các cấu trúc phức tạp trên khuôn mặt, trong khi  $R=2$  đảm bảo vùng lân cận đủ rộng để bắt đặc trưng.
  - `method='uniform'`: Giảm số lượng mẫu LBP từ  $2162^{16}$  xuống còn 18 ( $P + 2$ ), tăng hiệu quả mà vẫn giữ thông tin quan trọng.
- **Random Forest:**
  - GridSearchCV: Tự động tìm tham số tối ưu trong không gian tham số lớn, đảm bảo hiệu suất tốt nhất.
  - `class_weight='balanced'`: Điều chỉnh trọng số lớp để xử lý mất cân bằng dữ liệu (ví dụ: một số biểu cảm như "vui" có thể xuất hiện nhiều hơn "ghê tởm").



### 3. Kết quả thực nghiệm

#### 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".

#### 3.2 Kết quả

- **Độ chính xác:** Đạt khoảng 30-35% 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 16 ảnh ngẫu nhiên với nhãn thực tế và dự đoán, lưu dưới dạng result\_task4.jpg.



### 4. Thảo luận và nhận xét

#### 4.1 Phân tích kết quả

- **Hiệu suất thấp (30-35%):** 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:** Đơn giản, tính toán nhanh, và không nhạy với thay đổi ánh sáng đơn sắc, phù hợp cho các hệ thống tài nguyên hạn chế.
- **Random Forest:** Xử lý tốt dữ liệu mất cân bằng nhờ `class_weight='balanced'`, ít bị overfitting so với các mô hình khác như SVM.
- **Tính trực quan:** Ma trận nhầm lẫn và dự đoán trực quan giúp dễ dàng đánh giá hiệu suất và nhận diện lỗi.

## 4.3 Hạn chế và thách thức

- **Độ chính xác thấp:** Với chỉ 30-35%, mô hình không đáp ứng được yêu cầu thực tế cho các ứng dụng nhận dạng biểu cảm.
- **Giới hạn của LBPH:** Không đủ mạnh để nắm bắt các đặc trưng phức tạp của biểu cảm, đặc biệt với ảnh độ phân giải thấp.
- **Mất cân bằng dữ liệu:** Một số lớp (như "Disgust") có ít mẫu hơn, ảnh hưởng đến hiệu suất phân loại.
- **Thời gian xử lý:** GridSearchCV với không gian tham số lớn tiêu tốn nhiều thời gian huấn luyện.

## 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.

## 5. Kết luận

Tác vụ 4 đã triển khai thành công một hệ thống nhận dạng biểu cảm khuôn mặt sử dụng LBPH và Random Forest trên bộ dữ liệu FER2013. Tuy nhiên, với độ chính xác chỉ đạt khoảng 30-35%, mô hình chưa đáp ứng được yêu cầu thực tế. Các hạn chế chủ yếu đến từ chất lượng dữ liệu, tính phức tạp của bài toán, và giới hạn của LBPH. Các cải tiến trong tương lai, đặc biệt là sử dụng học sâu và tiền xử lý dữ liệu tốt hơn, có thể nâng cao đáng kể hiệu suất.



## Chương 6: Kết luận chung

### 6.1 Tóm tắt các kết quả chính đạt được của toàn bộ project

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 khác nhau 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  $k=20$ , tỷ lệ nén đạt khoảng **3.2:1**, trong khi chất lượng ảnh tái tạo được duy trì ở mức tốt với **PSNR  $\approx$  35.27 dB** và **MSE = 19.33**. Kết quả này chứng minh PCA là một phương pháp hiệu quả để giảm dung lượng ảnh mà vẫn giữ được các đặc trưng quan trọng.
- **Tác vụ 2: Tạo ảnh panorama sử dụng SIFT:** Hai ảnh chồng lấp đã được ghép thành công để tạo ra một ảnh panorama liền mạch. Sử dụng thuật toán SIFT để trích xuất và khớp đặc trưng, Homography để căn chỉnh hình học, và Multiband Blending để hợp nhất ảnh, kết quả cho thấy không có đường nối rõ ràng, với tỷ lệ **inliers sau RANSAC đạt khoảng 75-83%**, đảm bảo độ chính xác cao về hình học và màu sắc.
- **Tác vụ 3: Khớp ảnh/Đăng ký ảnh sử dụng SIFT:** Dự án đã định vị chính xác đối tượng trong ảnh cảnh, vẽ khung bao quanh đối tượng và hiển thị các cặp điểm khớp. Số lượng cặp khớp tốt dao động từ **24 đến 114**, tùy thuộc vào ảnh đầu vào. Phương pháp này hoạt động hiệu quả với các đối tượng phẳng, nhưng gặp hạn chế với các đối tượng 3D hoặc góc chụp lệch lớn.
- **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 dựa trên bộ dữ liệu FER2013, đạt độ chính xác khoảng **30-35%** trên tập kiểm tra. Dù kết quả chưa cao so với các phương pháp học sâu, đây vẫn là một giải pháp đơn giản và khả thi trong điều kiện tài nguyên hạn chế.

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

Quá trình thực hiện dự án đã giúp Tôi 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:** Tôi đã hiểu sâu hơn về các thuật toán cốt lõi như PCA (giảm chiều dữ liệu), SIFT (trích xuất 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:** Tôi đã sử dụng thành thạo các thư viện Python như:
  - **OpenCV:** Xử lý ảnh (đọc, ghi, chuyển đổi màu).
  - **Scikit-learn:** Triển khai PCA, Random Forest, và tối ưu hóa tham số (GridSearchCV).
  - **Scikit-image:** Trích xuất đặc trưng LBPH.
  - **Matplotlib:** Trực quan hóa kết quả.
- **Phân tích và đánh giá:** Các chỉ số như **MSE, PSNR, độ chính xác**, và **ma trận nhầm lẫn** được áp dụng để đánh giá hiệu suất, giúp Tôi định lượng và so sánh chất lượng của các phương pháp.
- **Giải quyết vấn đề:** đã xử lý các thách thức thực tế như mất cân bằng dữ liệu (Tác vụ 4), nhiễu trong ảnh (Tác vụ 2), và tối ưu hóa tham số (Tác vụ 1), qua đó rèn luyện khả năng tư duy phản 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, nh đã đối mặt với một số khó khăn và đưa ra các giải pháp tương ứng:

- **Tác vụ 1: Nén ảnh sử dụng PCA**
  - **Khó khăn:** Chọn giá trị k phù hợp để cân bằng giữa tỷ lệ nén và chất lượng ảnh tái tạo.
  - **Giải pháp:** Thử nghiệm nhiều giá trị k (10, 20, v.v.) và đánh giá qua **PSNR** và **MSE**, cuối cùng chọn k=20 để đạt chất lượng tốt ( $\text{PSNR} \approx 35.27 \text{ dB}$ ) với tỷ lệ nén hợp lý (3.2:1).
- **Tác vụ 2: Tạo ảnh panorama sử dụng SIFT**
  - **Khó khăn:** Xử lý ảnh với vùng chồng lấp nhỏ hoặc hiệu ứng parallax gây ra sai lệch hình học.
  - **Giải pháp:** Giảm kích thước ảnh để tăng tốc độ xử lý và áp dụng **Multiband Blending** để giảm thiểu đường nối, cải thiện chất lượng ảnh panorama.
- **Tác vụ 3: Khớp ảnh/Đăng ký ảnh sử dụng SIFT**
  - **Khó khăn:** Đảm bảo có đủ cặp đặc trưng khớp để ước lượng Homography chính xác.
  - **Giải pháp:** Sử dụng **Lowe's Ratio Test** (ngưỡng 0.75) và **RANSAC** (ngưỡng tái chiếu 5.0) để lọc outliers, đảm bảo độ tin cậy của các cặp khớp.
- **Tác vụ 4: Nhận dạng biểu cảm khuôn mặt sử dụng LBPH và Random Forest**
  - **Khó khăn:** Độ chính xác thấp (30-35%) do hạn chế của LBPH và dữ liệu FER2013 không đồng đều.
  - **Giải pháp:** Tối ưu hóa tham số Random Forest bằng **GridSearchCV** và sử dụng **class\_weight='balanced'** để xử lý mất cân bằng lớp, đồng thời đề xuất cải tiến bằng học sâu trong tương lai.

### 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 cho từng tác vụ:

- **Tác vụ 1: Nén ảnh sử dụng PCA** Kết hợp PCA với các kỹ thuật nén khác như **JPEG** để so sánh hiệu quả và cải thiện chất lượng ảnh tái tạo, hoặc thử nghiệm các giá trị k lớn hơn (25, 30) để tối ưu hơn nữa.
- **Tác vụ 2: Tạo ảnh panorama sử dụng SIFT** Triển khai các kỹ thuật xử lý hiệu ứng parallax (như **APAP**) và phát triển khả năng ghép ảnh 360 độ, đồng thời tối ưu hóa tốc độ bằng cách sử dụng các thuật toán nhanh hơn như **ORB**.
- **Tác vụ 3: Khớp ảnh/Đăng ký ảnh sử dụng SIFT** Ứng dụng các phương pháp khớp đặc trưng nhanh hơn như **FLANN**, hoặc mở rộng khả năng xử lý đối tượng 3D bằng cách sử dụng nhiều Homography hoặc ước lượng pose.
- **Tác vụ 4: Nhận dạng biểu cảm khuôn mặt sử dụng LBPH và Random Forest** Chuyển sang các mô hình học sâu như **CNN** để cải thiện độ chính xác (nhắm tới trên 70%), kết hợp kỹ thuật tăng cường dữ liệu (data augmentation) để xử lý mất cân bằng lớp và nâng cao hiệu suất.

## **Chương 7: Phụ lục – Tài liệu tham khảo**

- Edward, S. (2024). Panorama Image Stitching [GitHub repository]. GitHub.  
<https://github.com/stanleyedward/panorama-image-stitching>