Z

**TRƯỜNG ĐẠI HỌC CÔNG NGHỆ ĐÔNG Á**

**KHOA CÔNG NGHỆ THÔNG TIN**



**BÀI TẬP LỚN**

**HỌC PHẦN: XỬ LÝ ẢNH VÀ THỊ GIÁC MÁY TÍNH**

**Đề tài số 02: Xây dựng hệ thống nhận diện màu sắc bằng Pandas và OpenCV**

**Giảng viên hướng dẫn: Lương Thị Hồng Lan**

|  |  |  |  |
| --- | --- | --- | --- |
| **TT** | **Mã sinh viên** | **Sinh viên thực hiện** | **Lớp hành chính** |
| **1** | **20210531** | **Lê Trung Dũng** | **DCCNTT12.10.2** |
| **2** | **20210509** | **Nguyễn Tiến Đạt** | **DCCNTT12.10.2** |
| **3** | **20210490** | **Trương Quang Thắng** | **DCCNTT12.10.2** |
| **4** | **20210463** | **Ngô Long Vũ** | **DCCNTT12.10.2** |

**Bắc Ninh, năm 2024**

**TRƯỜNG ĐẠI HỌC CÔNG NGHỆ ĐÔNG Á**

**KHOA CÔNG NGHỆ THÔNG TIN**

**BÀI TẬP LỚN**

**HỌC PHẦN: XỬ LÝ ẢNH VÀ THỊ GIÁC MÁY TÍNH**

**Đề tài số 02: Xây dựng hệ thống nhận diện màu sắc bằng Pandas và OpenCV**

**Giảng viên hướng dẫn: Lương Thị Hồng Lan**

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Mã sinh viên** | **Sinh viên thực hiện** | **Lớp hành chính** |
| **1** | **20210531** | **Lê Trung Dũng** | **DCCNTT12.10.2** |
| **2** | **20210509** | **Nguyễn Tiến Đạt** | **DCCNTT12.10.2** |
| **3** | **20210490** | **Trương Quang Thắng** | **DCCNTT12.10.2** |
| **4** | **20210463** | **Ngô Long Vũ** | **DCCNTT12.10.2** |

**TRƯỜNG ĐẠI HỌC CÔNG NGHỆ ĐÔNG Á**

**KHOA CÔNG NGHỆ THÔNG TIN**

**BÀI TẬP LỚN**

**HỌC PHẦN: XỬ LÝ ẢNH VÀ THỊ GIÁC MÁY TÍNH**

**Đề tài số 02: Xây dựng hệ thống nhận diện màu sắc bằng Pandas và OpenCV**

**Giảng viên hướng dẫn: Lương Thị Hồng Lan**

|  |  |  |  |
| --- | --- | --- | --- |
| **TT** | **Mã sinh viên** | **Sinh viên thực hiện** | **Lớp hành chính** |
| **1** | **20210531** | **Lê Trung Dũng** | **DCCNTT12.10.2** |
| **2** | **20210509** | **Nguyễn Tiến Đạt** | **DCCNTT12.10.2** |
| **3** | **20210490** | **Trương Quang Thắng** | **DCCNTT12.10.2** |
| **4** | **20210463** | **Ngô Long Vũ** | **DCCNTT12.10.2** |

|  |  |
| --- | --- |
|  |  |

**Bắc Ninh, năm 2024**

|  |  |
| --- | --- |
| TRƯỜNG ĐẠI HỌC CÔNG NGHỆ ĐÔNG Á  **KHOA CÔNG NGHỆ THÔNG TIN** | **KỲ THI KẾT THÚC HỌC PHẦN**  **HỌC KỲ 1, NĂM HỌC 2024** – **2025** |

|  |  |
| --- | --- |
| **PHIẾU CHẤM THI BÀI TẬP LỚN KẾT THÚC HỌC PHẦN**  **Mã đề thi 02: Xây dựng hệ thống nhận diện màu sắc bằng Pandas và OpenCV**  **Tên học phần: Xử lý ảnh và thị giác máy tính**  **Lớp Tín chỉ:**  **XATGMT.03.K12.02.LH.C04.1\_LT** | |
| **Cán bộ chấm thi 1**  *(Ký và ghi rõ họ tên)*  **Lương Thị Hồng Lan** | **Cán bộ chấm thi 2**  *(Ký và ghi rõ họ tên)* |

| **TT** | **TIÊU CHÍ** | **THANG ĐIỂM** | **Lê Trung Dũng** | **Nguyễn Tiến Đạt** | **Trương Quang Thắng** | **Ngô Long Vũ** |
| --- | --- | --- | --- | --- | --- | --- |
| **20210531** | **20210509** | **20210490** | **20210463** |
| **1** | **Nội dung báo cáo trên Word đầy đủ** | **3.5** |  |  |  |  |
| 1.1 | Có bố cục rõ ràng (mục lục, phần mở đầu, nội dung chính, kết luận). | 0,5 |  |  |  |  |
| 1.2 | Nội dung phân tích rõ ràng, logic. | 0,5 |  |  |  |  |
| 1.3 | Có dẫn chứng, số liệu minh họa đầy đủ. | 0,5 |  |  |  |  |
| 1.4 | Ngôn ngữ và trình bày chuẩn, không lỗi chính tả. | 0,5 |  |  |  |  |
| 1.5 | Có trích dẫn tài liệu tham khảo đúng quy cách. | 0,5 |  |  |  |  |
| 1.6 | Được trình bày chuyên nghiệp (canh lề, font chữ, khoảng cách dòng hợp lý). | 0,5 |  |  |  |  |
| 1.7 | Tài liệu đầy đủ, bám sát yêu cầu của đề bài. | 0,5 |  |  |  |  |
| **2** | **Nội dung thuyết trình đầy đủ** | **1.0** |  |  |  |  |
| 2.1 | Trình bày tự tin, phát âm rõ ràng, mạch lạc. | 0,5 |  |  |  |  |
| 2.2 | Nội dung thuyết trình đúng trọng tâm, không lan man. | 0,5 |  |  |  |  |
| **3** | **Slides báo cáo đầy đủ nội dung + Hỏi đáp** | **3.0** |  |  |  |  |
| 3.1 | Slides có bố cục rõ ràng (mở đầu, nội dung, kết luận). | 0,5 |  |  |  |  |
| 3.2 | Thiết kế slides đẹp, chuyên nghiệp (màu sắc, hình ảnh minh họa). | 0,5 |  |  |  |  |
| 3.3 | Nội dung trên slides ngắn gọn, dễ hiểu, súc tích. | 0,5 |  |  |  |  |
| 3.4 | Nội dung slides phù hợp với nội dung báo cáo. | 0,5 |  |  |  |  |
| 3.5 | Trả lời câu hỏi đầy đủ, chính xác. | 0,5 |  |  |  |  |
| 3.6 | Trả lời câu hỏi tự tin, thuyết phục. | 0,5 |  |  |  |  |
| **4** | **Code đầy đủ** | **2.5** |  |  |  |  |
| 1.1 | Code được trình bày rõ ràng, có chú thích đầy đủ. | 0,5 |  |  |  |  |
| 1.2 | Code chạy đúng, không lỗi. | 0,5 |  |  |  |  |
| 1.3 | Code tối ưu, không dư thừa. | 0,5 |  |  |  |  |
| 1.4 | Đáp ứng đầy đủ các yêu cầu chức năng theo đề bài. | 0,5 |  |  |  |  |
| 1.5 | Có tính sáng tạo hoặc cải thiện so với yêu cầu. | 0,5 |  |  |  |  |
| **TỔNG ĐIỂM BẰNG SỐ:** | | **10** |  |  |  |  |
| **TỔNG ĐIỂM BẰNG CHỮ:** | | *Mười tròn* |  |  |  |  |

# **LỜI MỞ ĐẦU**

Trong thời đại công nghệ số, các ứng dụng xử lý hình ảnh và nhận diện trở thành một phần không thể thiếu trong nhiều lĩnh vực, từ y học, giáo dục đến công nghiệp và giải trí. Trong số đó, việc nhận diện màu sắc đóng vai trò quan trọng trong các hệ thống như nhận dạng vật thể, phân loại hình ảnh, hỗ trợ thị giác máy tính, và các ứng dụng tự động hóa.

Dự án "Xây dựng hệ thống nhận diện màu sắc bằng Pandas và OpenCV" được thực hiện nhằm mục tiêu tạo ra một công cụ mạnh mẽ và linh hoạt, giúp phân tích và nhận diện màu sắc từ hình ảnh. Sử dụng thư viện Pandas để quản lý và phân tích dữ liệu một cách hiệu quả, kết hợp với OpenCV – một công cụ hàng đầu trong lĩnh vực xử lý hình ảnh, dự án hứa hẹn mang lại một giải pháp vừa tối ưu vừa dễ triển khai.

Trong quá trình phát triển hệ thống, nhóm tập trung vào các nhiệm vụ chính như: xử lý và phân tích hình ảnh, xây dựng thuật toán nhận diện màu sắc, cũng như tối ưu hóa hiệu năng để hệ thống có thể áp dụng thực tế trong nhiều môi trường khác nhau. Dự án không chỉ giúp nâng cao hiểu biết về các công nghệ hiện đại mà còn góp phần phát triển các ứng dụng trong tương lai, từ giao diện người dùng thông minh cho đến hệ thống giám sát tự động.

Hy vọng rằng kết quả của dự án sẽ mang lại giá trị thiết thực, đồng thời là nền tảng để phát triển thêm các tính năng và ứng dụng mở rộng trong lĩnh vực xử lý hình ảnh và thị giác máy tính.

# **LỜI CẢM ƠN**

Chúng em xin gửi lời tri ân sâu sắc đến Giảng viên Lương Thị Hồng Lan, người đã tận tình hướng dẫn và hỗ trợ chúng em trong quá trình thực hiện đề tài về " Xây dựng hệ thống nhận diện màu sắc bằng Pandas và OpenCV ".

Đặc biệt, Cô đã tạo điều kiện cho chúng em có cơ hội thực hành và áp dụng kiến thức lý thuyết vào thực tế, giúp chúng em phát triển kỹ năng về trí tuệ nhân tạo, xử lý ảnh và học sâu trong lĩnh vực nhận dạng hình ảnh. Những hướng dẫn chi tiết của Cô về các kỹ thuật học máy và xử lý dữ liệu đã giúp chúng em rất nhiều trong việc xây dựng mô hình phân loại hiệu quả.

Bên cạnh đó, chúng em cũng rất trân trọng sự kiên nhẫn và sự động viên của Cô trong quá trình vượt qua những thách thức và khó khăn trong nghiên cứu và thực hiện đề tài. Sự hỗ trợ của Cô không chỉ giúp chúng em hoàn thành tốt đề tài mà còn giúp chúng em tích lũy được nhiều kinh nghiệm quý báu trong lĩnh vực trí tuệ nhân tạo và thị giác máy tính.

Một lần nữa, chúng em xin bày tỏ lòng biết ơn chân thành đến Cô Lương Thị Hồng Lan, người đã luôn bên cạnh, hỗ trợ và khuyến khích chúng em trong suốt hành trình này. Chúng em sẽ luôn nhớ và trân trọng những kiến thức và kinh nghiệm mà Cô đã truyền đạt. Xin chân thành cảm ơn!

**MỤC LỤC**

[**LỜI MỞ ĐẦU** 6](#_Toc184633204)

[**LỜI CẢM ƠN** 7](#_Toc184633205)

[**DANH MỤC HÌNH** 10](#_Toc184633206)

[**Chương 1: Tổng quan về nhận dạng màu sắc** 11](#_Toc184633207)

[**1.1. Nhận dạng màu sắc trong lĩnh vực xử lý ảnh** 11](#_Toc184633208)

[**1.1.1. Mô tả bài toán nhận dạng màu sắc** 11](#_Toc184633209)

[**1.1.2. Khó khăn và thách thức trong nhận dạng màu sắc** 11](#_Toc184633210)

[**1.1.3. Ứng dụng của nhận dạng màu sắc trong các lĩnh vực khác nhau** 12](#_Toc184633211)

[**1.2 Các phương pháp nhận dạng màu sắc trong xử lý ảnh** 12](#_Toc184633212)

[**1.2.1. Các phương pháp truyền thống** 12](#_Toc184633213)

[**1.2.2. Các phương pháp học máy** 16](#_Toc184633214)

[**1.2.3. Các kỹ thuật phân loại màu sắc và phân tích màu sắc trong ảnh** 18](#_Toc184633215)

[**1.3. Ứng dụng của công nghệ nhận dạng màu sắc** 21](#_Toc184633216)

[**1.3.1. Nhận dạng đối tượng và phân tích ảnh** 21](#_Toc184633217)

[**1.3.2. Hệ thống thông minh và tự động hóa** 22](#_Toc184633218)

[**1.3.3. Trong công nghiệp** 23](#_Toc184633219)

[**Chương 2: Xây dựng hệ thống nhận diện màu sắc bằng Pandas và OpenCV** 24](#_Toc184633220)

[**2.1. Giới thiệu về hệ thống nhận diện màu sắc** 24](#_Toc184633221)

[**2.1.1. Mô tả chung về hệ thống và mục tiêu xây dựng** 24](#_Toc184633222)

[**2.1.2. Quy Trình Xây Dựng Hệ Thống Nhận Diện Màu Sắc** 25](#_Toc184633223)

[**2.2. Công cụ và thư viện sử dụng** 33](#_Toc184633224)

[**2.2.1. OpenCV trong hệ thống nhận dạng màu sắc** 35](#_Toc184633225)

[**2.2.2. Tối ưu hóa hiệu suất với OpenCV** 37](#_Toc184633226)

[**2.2.3. Xử lý dữ liệu training với OpenCV** 37](#_Toc184633227)

[**2.2.4. Pandas** 38](#_Toc184633228)

[**Chương 3: Thực nghiệm và Đánh giá** 40](#_Toc184633229)

[**3.1. Dữ liệu** 40](#_Toc184633230)

[**3.1.1. Dữ liệu ảnh** 40](#_Toc184633231)

[**3.1.2. Quy trình tiền xử lý** 40](#_Toc184633232)

[**3.1.3. Phương pháp nhận diện màu sắc áp dụng trong thực nghiệm** 41](#_Toc184633233)

[**3.2. Độ đo đánh giá** 44](#_Toc184633234)

[**3.3. Kết quả thực nghiệm và phân tích** 45](#_Toc184633235)

[**3.3.1. Trình bày các kết quả thực nghiệm, bao gồm độ chính xác và các trường hợp thử nghiệm cụ thể** 45](#_Toc184633236)

[**3.3.2. Phân tích những ưu điểm và hạn chế của hệ thống hiện tại** 45](#_Toc184633237)

[**3.4. Đánh giá hiệu suất của hệ thống** 45](#_Toc184633238)

[**Kết luận** 47](#_Toc184633239)

[**Tài liệu tham khảo** 49](#_Toc184633240)

# **DANH MỤC HÌNH**

[Hình 1 : Công thức chuyển đổi từ RGB sang HSV 15](#_Toc184632678)

[Hình 2 : Công thức chuyển đổi từ HSV sang RGB 15](#_Toc184632679)

[Hình 3: Dữ liệu dataset 40](#_Toc184632680)

# **Chương 1: Tổng quan về nhận dạng màu sắc**

## **1.1. Nhận dạng màu sắc trong lĩnh vực xử lý ảnh**

### **1.1.1. Mô tả bài toán nhận dạng màu sắc**

Nhận dạng màu sắc là một trong những vấn đề cốt lõi trong lĩnh vực xử lý ảnh và thị giác máy tính. Vấn đề này yêu cầu hệ thống có khả năng xác định và phân biệt các màu sắc khác nhau trong một hình ảnh hoặc video. Sự nhận biết màu sắc là nền tảng cho nhiều ứng dụng như phân đoạn ảnh, theo dõi đối tượng, và phân loại đối tượng trong hình ảnh.

Ví dụ, trong một hệ thống phân loại trái cây, nhận dạng màu sắc có thể được sử dụng để phân biệt các loại quả như táo đỏ, táo xanh hoặc cam, dựa trên đặc trưng màu sắc của chúng. Tương tự, trong hệ thống giám sát giao thông, việc xác định màu của xe cộ có thể hỗ trợ nhận diện vi phạm giao thông hoặc phân loại phương tiện.

Nhận dạng màu sắc không chỉ đơn thuần là việc xác định màu sắc của các điểm ảnh mà còn phải xử lý các yếu tố phức tạp như ánh sáng môi trường, góc nhìn, và đặc tính bề mặt của đối tượng.

### **1.1.2. Khó khăn và thách thức trong nhận dạng màu sắc**

Nhận dạng màu sắc đối mặt với nhiều khó khăn và thách thức, bao gồm:

* **Ánh sáng thay đổi**: Sự thay đổi của điều kiện ánh sáng có thể làm biến đổi màu sắc của đối tượng trong hình ảnh, gây khó khăn trong việc nhận dạng chính xác. Ví dụ, một vật thể có màu đỏ tươi dưới ánh sáng mặt trời có thể xuất hiện với màu tối hơn dưới ánh sáng nhân tạo yếu.
* **Biến đổi màu sắc do môi trường**: Các yếu tố môi trường như ánh sáng nền, bóng đổ, và phản chiếu cũng có thể ảnh hưởng đến màu sắc được nhận dạng. Chẳng hạn, một chiếc ô tô màu xanh có thể xuất hiện màu hơi xám do phản chiếu của bầu trời.
* **Đặc tính bề mặt đối tượng**: Các bề mặt khác nhau (nhẵn, gồ ghề, phản chiếu) có thể phản xạ ánh sáng một cách khác nhau, làm cho màu sắc hiển thị không đồng nhất. Ví dụ, một quả bóng nhựa bóng loáng có thể xuất hiện sáng hơn và có các vùng phản chiếu so với một quả bóng cao su mờ.

Để giải quyết những thách thức này, các phương pháp nhận dạng màu sắc thường phải tích hợp các kỹ thuật tiền xử lý như cân bằng trắng (white balancing) để chuẩn hóa ánh sáng, hiệu chỉnh màu sắc (color correction) để giảm thiểu ảnh hưởng của môi trường, và lọc nhiễu (noise filtering) để cải thiện chất lượng hình ảnh.

### **1.1.3. Ứng dụng của nhận dạng màu sắc trong các lĩnh vực khác nhau**

Nhận dạng màu sắc có nhiều ứng dụng trong các lĩnh vực khác nhau:

* **Nhận dạng đối tượng**: Sử dụng màu sắc để nhận diện và theo dõi các đối tượng trong hình ảnh và video, ứng dụng trong giám sát an ninh, robot công nghiệp, và hệ thống lái tự động. Ví dụ, trong giám sát an ninh, nhận dạng màu áo của người di chuyển có thể hỗ trợ việc tìm kiếm đối tượng trong đám đông.
* **Phân loại ảnh**: Phân loại các hình ảnh dựa trên màu sắc, ứng dụng trong quản lý dữ liệu ảnh, tìm kiếm hình ảnh, và nhận diện mẫu. Ví dụ, hệ thống quản lý ảnh cá nhân có thể phân loại ảnh theo mùa (ảnh mùa xuân có màu xanh lá cây chủ đạo, ảnh mùa thu có màu vàng cam).
* **Nhận diện sản phẩm**: Sử dụng nhận dạng màu sắc trong thương mại điện tử và quản lý kho để xác định và phân loại sản phẩm. Ví dụ, các hệ thống kiểm tra chất lượng trong dây chuyền sản xuất có thể phát hiện các sản phẩm lỗi dựa trên sự sai lệch màu sắc.

Việc nhận dạng màu sắc một cách chính xác và hiệu quả không chỉ cải thiện hiệu suất của các hệ thống thị giác máy tính mà còn mở rộng phạm vi ứng dụng của công nghệ này trong đời sống hàng ngày.

## **1.2 Các phương pháp nhận dạng màu sắc trong xử lý ảnh**

### **1.2.1. Các phương pháp truyền thống**

#### **1.2.1.1. Không gian màu RGB**

**RGB** (Red, Green, Blue) là một không gian màu nổi tiếng và phổ biến, được sử dụng chủ yếu trong các thiết bị như máy tính, TV, máy ảnh kỹ thuật số và điện thoại di động. Không gian RGB dựa trên nguyên lý trộn ba màu cơ bản là **Đỏ (Red)**, **Xanh lá (Green)** và **Xanh dương (Blue)** để tạo ra tất cả các màu sắc có thể có. Mỗi thành phần trong không gian màu RGB sẽ có giá trị từ 0 đến 255, tương ứng với độ sáng của các màu đó, cho phép tạo ra tổng cộng 16.7 triệu màu sắc.

**Ưu điểm:**

**Đơn giản và dễ hiểu**: Mỗi điểm ảnh trong không gian RGB có thể được xác định bởi ba giá trị (R, G, B), dễ dàng biểu diễn trong các phần mềm đồ họa và trong hệ thống máy tính.

**Ứng dụng rộng rãi**: Không gian màu RGB phù hợp cho nhiều ứng dụng, đặc biệt trong việc thể hiện và truyền tải hình ảnh trên màn hình máy tính và các thiết bị hiển thị.

**Hạn chế:**

**Không phản ánh chính xác nhận thức màu sắc của con người**: Không gian RGB không phải là cách tiếp cận tốt nhất để mô phỏng nhận thức màu sắc tự nhiên của con người. Con người không cảm nhận màu sắc theo cách mà các giá trị R, G, B được phân bổ.

**Nhạy cảm với điều kiện ánh sáng**: Đặc biệt khi ánh sáng thay đổi, không gian RGB có thể làm sai lệch màu sắc được nhận diện. Ví dụ, khi môi trường có ánh sáng quá mạnh hay quá yếu, màu sắc có thể bị mất đi tính chính xác.

**Khó nhận diện màu sắc khi ánh sáng thay đổi**: Việc phân biệt các màu sắc sẽ gặp khó khăn nếu chỉ sử dụng không gian RGB, đặc biệt khi đối mặt với sự thay đổi về độ sáng và độ bão hòa.

Ví dụ, hình ảnh trong không gian RGB có thể được biểu diễn bằng ba giá trị [R, G, B]. Khi muốn nhận diện một màu nhất định (ví dụ như đỏ), bạn cần so sánh các giá trị R, G, B của mỗi pixel với một ngưỡng cụ thể để xác định xem màu đó có phù hợp với "màu đỏ" hay không.

#### **1.2.1.2. Không gian màu HSV**

**HSV** là viết tắt của **Hue** (màu sắc), **Saturation** (độ bão hòa), và **Value** (độ sáng). Không gian màu này được thiết kế để dễ dàng phân biệt và nhận diện các màu sắc theo cách mà con người cảm nhận chúng hơn là theo ba giá trị R, G, B. Cụ thể:

* **Hue (H)**: Đại diện cho màu sắc của pixel, được đo theo góc độ từ 0 đến 360°. Ví dụ: Màu đỏ có thể ở khoảng 0° đến 10°, xanh lá là khoảng 90°, xanh dương là khoảng 200°.
* **Saturation (S)**: Đại diện cho độ bão hòa hay độ tinh khiết của màu sắc. Saturation càng cao, màu sắc càng tươi sáng, càng thuần khiết. Nếu saturation bằng 0, màu sẽ trở thành màu xám.
* **Value (V)**: Đại diện cho độ sáng của màu sắc, từ 0 (đen) đến 100% (màu sắc tươi sáng nhất).

**Ưu điểm:**

**Phản ánh chính xác cảm nhận màu sắc của con người**: Không gian HSV được xây dựng để gần gũi với cách mà con người nhận thức màu sắc. Việc tách rời các thông tin màu sắc (Hue) khỏi độ bão hòa và độ sáng giúp dễ dàng nhận diện và phân loại màu.

**Nhận diện màu sắc dễ dàng trong các điều kiện ánh sáng khác nhau**: Vì độ sáng và độ bão hòa được tách biệt, việc nhận diện màu sắc trở nên chính xác hơn trong các điều kiện ánh sáng thay đổi, so với không gian màu RGB. Điều này đặc biệt hữu ích trong các ứng dụng thực tế, nơi ánh sáng môi trường có thể thay đổi.

**Phân loại màu sắc dễ dàng hơn**: Dễ dàng phân loại và nhận diện các màu chính (như đỏ, xanh lá, xanh dương, vàng, vv...) chỉ dựa trên giá trị của Hue. Ví dụ, màu đỏ có thể được xác định trong phạm vi Hue từ 0° đến 10° và 170° đến 180°.

**Hạn chế:**

**Có thể không phản ánh chính xác tất cả các màu sắc**: Dù HSV có ưu điểm về khả năng nhận diện màu sắc theo cảm nhận của con người, nhưng trong một số trường hợp, vẫn có thể xảy ra sự không chính xác khi mô tả các màu sắc phức tạp hoặc màu sắc ngoài phạm vi của hệ màu này.

**Phức tạp trong việc chuyển đổi từ RGB sang HSV**: Dù có công thức chuyển đổi từ RGB sang HSV, nhưng việc tính toán và xử lý không gian HSV có thể tốn nhiều tài nguyên tính toán, đặc biệt trong các ứng dụng yêu cầu tốc độ cao.

**Công thức chuyển đổi**

Công thức chuyển đổi giữa RGB và HSV giúp người dùng dễ dàng chuyển đổi giữa các không gian màu:

Từ RGB sang HSV:

A math equations on a white background

Description automatically generated

Hình 1 : Công thức chuyển đổi từ RGB sang HSV

Từ HSV sang RGB:

A math problem with numbers and equations

Description automatically generated with medium confidence

Hình 2 : Công thức chuyển đổi từ HSV sang RGB

Trong lĩnh vực thị giác máy tính, không gian màu HSV thường được sử dụng để phân loại và nhận diện màu sắc trong các điều kiện ánh sáng khác nhau. Điều này đặc biệt hữu ích trong các ứng dụng như nhận diện đối tượng, theo dõi chuyển động và phân loại hình ảnh.

**Ứng dụng thực tế và so sánh**

Trong các ứng dụng thực tế, không gian RGB thường được sử dụng trong đồ họa máy tính, xử lý ảnh số và hiển thị màn hình. Ngược lại, không gian HSV thường được ứng dụng trong nhận diện màu sắc và thị giác máy tính, nơi màu sắc cần được phân loại và xử lý dưới điều kiện ánh sáng thay đổi.

Ngoài RGB và HSV, còn nhiều không gian màu khác như **CMYK** (dùng trong in ấn) và **CIELAB** (phản ánh nhận thức màu sắc của con người một cách trung thực hơn).

* **CMYK (Cyan, Magenta, Yellow, Black)**: Được sử dụng chủ yếu trong in ấn, không gian màu CMYK tạo ra các màu sắc bằng cách trộn lẫn các mực in. Đây là không gian màu trừ, nghĩa là bắt đầu từ màu trắng (giấy trắng) và loại bỏ các màu sắc để tạo ra màu mong muốn.
* **CIELAB**: Được thiết kế để phản ánh cách con người cảm nhận màu sắc, CIELAB không phụ thuộc vào thiết bị, nghĩa là màu sắc sẽ không thay đổi khi chuyển từ thiết bị này sang thiết bị khác. Điều này làm cho nó trở thành một không gian màu lý tưởng trong các ứng dụng đòi hỏi sự chính xác màu sắc tuyệt đối.

**Thư viện và công cụ lập trình**

### **1.2.2. Các phương pháp học máy**

Trong thời đại trí tuệ nhân tạo phát triển mạnh mẽ, các phương pháp học máy ngày càng được áp dụng rộng rãi, mang lại hiệu quả cao trong lĩnh vực nhận dạng màu sắc. Những phương pháp này không chỉ cải thiện độ chính xác mà còn tăng cường khả năng tự động hóa trong phân loại và nhận dạng màu sắc. Dưới đây là các phương pháp học máy tiêu biểu được ứng dụng trong lĩnh vực này:

**Thuật Toán Phân Cụm K-means**

K-means là một trong những thuật toán học không giám sát phổ biến nhất, được sử dụng hiệu quả trong nhận dạng màu sắc. Thuật toán này phân chia không gian màu thành KKK cụm khác nhau, mỗi cụm đại diện cho một màu sắc chủ đạo.

**Quy trình hoạt động của K-means bao gồm:**

**Khởi tạo:** Chọn ngẫu nhiên KKK tâm cụm trong không gian màu (thường là RGB hoặc HSV).

**Gán nhãn:** Mỗi pixel trong ảnh được gán cho cụm có tâm gần nhất dựa trên khoảng cách Euclidean.

**Cập nhật:** Tính toán lại vị trí các tâm cụm bằng cách lấy trung bình các pixel trong từng cụm.

**Lặp lại:** Thực hiện lại các bước trên cho đến khi các tâm cụm hội tụ hoặc đạt số lần lặp tối đa.

**Ưu điểm:**

* Tự động phân nhóm màu sắc mà không cần định nghĩa trước các ngưỡng màu.
* Giảm số lượng màu sắc trong ảnh xuống còn KKK màu chủ đạo.
* Xử lý nhanh và yêu cầu tài nguyên tính toán thấp.

**Hạn chế:**

* Phụ thuộc vào số cụm KKK và vị trí khởi tạo ban đầu.
* Dễ bị ảnh hưởng bởi nhiễu và dữ liệu ngoại lai.
* Khó xử lý tốt các trường hợp phân bố màu phức tạp.

**Mạng Nơ-ron Tích Chập (CNN)**

CNN là một phương pháp học sâu được thiết kế đặc biệt cho xử lý ảnh và đã chứng minh hiệu quả vượt trội trong nhiều bài toán thị giác máy tính, bao gồm nhận dạng màu sắc.

**Kiến trúc điển hình của CNN bao gồm:**

**Lớp tích chập (Convolutional Layers):** Trích xuất các đặc trưng màu sắc từ ảnh đầu vào.

**Lớp gộp (Pooling Layers):** Giảm kích thước dữ liệu nhưng giữ lại thông tin quan trọng.

**Lớp kết nối đầy đủ (Fully Connected Layers):** Phân loại màu sắc dựa trên các đặc trưng đã học.

**Ưu điểm:**

* Học tự động các đặc trưng phức tạp của màu sắc.
* Độ chính xác cao trong nhận dạng và phân loại màu sắc.
* Xử lý tốt các điều kiện ánh sáng và môi trường khác nhau.
* Khả năng tổng quát hóa tốt với dữ liệu mới.

**Hạn chế:**

* Yêu cầu lượng dữ liệu huấn luyện lớn.
* Cần tài nguyên tính toán mạnh và thời gian huấn luyện dài.
* Chi phí triển khai cao hơn so với các phương pháp truyền thống.

**Cây Quyết Định và Random Forest**

Cây quyết định và Random Forest là các phương pháp học máy dựa trên quy tắc, được áp dụng hiệu quả trong phân loại màu sắc.

**Nguyên lý hoạt động:**

* **Cây quyết định:** Xây dựng cấu trúc cây phân cấp các quyết định dựa trên thuộc tính màu sắc.
* **Random Forest:** Kết hợp nhiều cây quyết định để tạo ra mô hình dự đoán mạnh mẽ hơn.

**Ưu điểm:**

* Dễ hiểu và giải thích quá trình ra quyết định.
* Hiệu quả với các bài toán phân loại màu sắc đơn giản.
* Không đòi hỏi nhiều tiền xử lý dữ liệu.

**Hạn chế:**

* Hiệu quả giảm khi số lượng lớp màu sắc tăng lên.
* Dễ bị overfitting với dữ liệu phức tạp.
* Khó xử lý tốt các trường hợp chuyển đổi màu sắc liên tục.

### **1.2.3. Các kỹ thuật phân loại màu sắc và phân tích màu sắc trong ảnh**

Trong lĩnh vực xử lý và phân tích hình ảnh, màu sắc là một trong những đặc trưng quan trọng, thường được sử dụng để phân loại, nhận diện hoặc mô tả nội dung ảnh. Các kỹ thuật phổ biến trong phân loại và phân tích màu sắc bao gồm biểu đồ màu sắc (histogram), phân vùng màu sắc và xác định màu chủ đạo.

**Histogram màu sắc**

**Mô tả:**

Histogram màu sắc là một công cụ trực quan hóa biểu diễn tần suất xuất hiện của các mức màu trong một bức ảnh. Biểu đồ này giúp người dùng hiểu rõ sự phân bố màu sắc trên toàn bộ ảnh. Trục hoành biểu thị giá trị màu (ví dụ: từ 0 đến 255 trong ảnh 8-bit), trong khi trục tung biểu thị số lượng điểm ảnh tương ứng với mỗi giá trị màu. Histogram thường được xây dựng cho từng kênh màu riêng biệt như R, G, B hoặc H, S, V.

**Ứng dụng:**

Histogram màu sắc được sử dụng phổ biến để:

* Phân tích sự phân bố màu sắc tổng thể của ảnh.
* So sánh các bức ảnh, hỗ trợ nhận diện sự tương đồng hoặc khác biệt về màu sắc.
* Làm đặc trưng đầu vào cho các thuật toán nhận dạng hoặc phân loại ảnh.

**Ưu điểm:**

* Dễ triển khai, trực quan và cung cấp thông tin tổng quan về màu sắc.

**Hạn chế:**

* Không lưu giữ thông tin không gian, dẫn đến việc không thể phân biệt hai ảnh có cùng phân bố màu sắc nhưng sắp xếp khác nhau.

**Phân vùng màu sắc (Color Segmentation)**

**Mô tả:**

Phân vùng màu sắc là quá trình chia một bức ảnh thành các vùng riêng biệt dựa trên sự tương đồng về giá trị màu sắc. Kỹ thuật này thường được sử dụng để làm nổi bật các khu vực hoặc đối tượng cụ thể trong ảnh.

**Kỹ thuật thực hiện:**

* **Phân cụm K-means:** Nhóm các điểm ảnh thành các cụm dựa trên giá trị màu.
* **Phân đoạn bằng ngưỡng:** Chọn ngưỡng giá trị để phân loại pixel thành các vùng khác nhau.
* **Kỹ thuật dựa trên biên (Edge-based):** Phát hiện các ranh giới màu sắc để phân vùng.

**Ứng dụng:**

* Nhận diện đối tượng hoặc khu vực quan tâm trong ảnh.
* Phát hiện biên hoặc các cấu trúc quan trọng.
* Tách lớp màu trong các ứng dụng chuyên biệt như phân tích ảnh y khoa, ảnh vệ tinh, hoặc ảnh thời trang.

**Ưu điểm:**

* Hiệu quả trong việc tách các vùng màu sắc khác biệt.
* Có thể kết hợp với các kỹ thuật xử lý ảnh khác để cải thiện độ chính xác.

**Hạn chế:**

* Phụ thuộc vào không gian màu được chọn và các tham số thuật toán.
* Nhạy cảm với nhiễu hoặc các vùng màu sắc có sự tương đồng cao.

**Xác định màu chủ đạo (Dominant Color Extraction)**

**Mô tả:**

Kỹ thuật xác định màu chủ đạo nhằm tìm ra các màu sắc chiếm ưu thế trong một bức ảnh. Màu sắc chủ đạo là các màu xuất hiện nhiều nhất, đại diện cho nội dung chính hoặc đặc trưng màu sắc của ảnh.

**Kỹ thuật thực hiện:**

* **Phân cụm K-means:** Nhóm các điểm ảnh theo màu sắc, sau đó xác định các cụm màu lớn nhất.
* **Sử dụng histogram:** Phân tích tần suất xuất hiện của các màu sắc để tìm ra các màu phổ biến nhất.

**Ứng dụng:**

* Tạo bảng màu (color palette) cho thiết kế đồ họa hoặc nội thất.
* Phân tích nội dung ảnh, đặc biệt trong các ứng dụng quảng cáo hoặc thời trang.
* Hỗ trợ các hệ thống đề xuất dựa trên màu sắc, ví dụ: đề xuất sản phẩm tương tự.

**Ưu điểm:**

* Giảm số lượng màu cần xử lý, giúp đơn giản hóa việc phân tích.
* Hiệu quả trong việc mô tả nội dung tổng quan của ảnh.

**Hạn chế:**

* Phụ thuộc vào số lượng cụm (k) được chọn trong thuật toán.
* Đối với ảnh có màu sắc phức tạp, việc xác định màu chủ đạo có thể gặp khó khăn.

## **1.3. Ứng dụng của công nghệ nhận dạng màu sắc**

### **1.3.1. Nhận dạng đối tượng và phân tích ảnh**

Công nghệ nhận dạng màu sắc là một phương pháp cơ bản trong xử lý ảnh và thị giác máy tính, giúp nhận diện các đặc điểm và đối tượng dựa trên màu sắc. Dưới đây là các ứng dụng nổi bật:

* **Nhận diện trái cây và nông sản**: Sử dụng màu sắc để phân loại trái cây và nông sản dựa vào trạng thái chín của chúng. Ví dụ, các hệ thống nhận dạng màu sắc có thể xác định một trái cam đã chín hoàn toàn hay chưa, hoặc phân biệt chuối xanh và chuối chín vàng để đảm bảo chất lượng khi phân phối.
* **Nhận diện biển báo giao thông**: Hỗ trợ hệ thống tự động hóa như xe tự lái nhận diện biển báo dựa trên màu sắc tiêu chuẩn. Màu đỏ thường được dùng cho biển báo nguy hiểm, màu xanh cho các chỉ dẫn và màu vàng cho các cảnh báo về tình trạng đường.
* **Phân đoạn ảnh trong y tế**: Giúp xác định vùng tổn thương hoặc mô bất thường trong các hình ảnh y khoa như MRI, CT hoặc X-quang. Công nghệ này có thể xác định các vùng màu sắc khác nhau, giúp bác sĩ phát hiện khối u hay vùng tổn thương với độ chính xác cao.
* **Phân tích thực vật**: Phát hiện bệnh hại, thiếu dinh dưỡng hoặc sâu bệnh thông qua sự thay đổi màu sắc của lá cây. Ví dụ, hệ thống nhận dạng màu sắc có thể phát hiện lá cây chuyển từ màu xanh sang vàng, cho thấy cây đang bị thiếu hụt dinh dưỡng hoặc bị tấn công bởi sâu bệnh.
* **Kiểm tra thực phẩm và bề mặt sản phẩm**: Phát hiện thực phẩm bị hỏng hoặc lỗi trên bề mặt sản phẩm thông qua sự thay đổi màu sắc. Ví dụ, công nghệ này có thể xác định thịt bị ôi hoặc rau quả bị nấm mốc bằng cách nhận biết những thay đổi bất thường về màu sắc.
* **Phân tích ảnh vệ tinh và môi trường**: Phân biệt các loại địa hình và theo dõi biến đổi môi trường dựa trên màu sắc. Ví dụ, màu xanh lá cây biểu thị rừng, trong khi màu nâu biểu thị đất trống, giúp các nhà nghiên cứu theo dõi sự thay đổi về rừng và môi trường sống.

### **1.3.2. Hệ thống thông minh và tự động hóa**

Công nghệ nhận dạng màu sắc đóng vai trò quan trọng trong các hệ thống thông minh và tự động hóa, với các ứng dụng sau:

* **Giám sát giao thông và robot tự hành**: Sử dụng nhận dạng màu sắc để theo dõi phương tiện giao thông và điều hướng robot trong dây chuyền sản xuất hoặc logistics. Ví dụ, camera giao thông có thể theo dõi xe cộ dựa trên màu sơn và phân biệt xe vượt đèn đỏ, trong khi robot trong nhà máy có thể xác định điểm đến hoặc phân loại hàng hóa dựa trên màu sắc.
* **Thiết bị nhà thông minh**: Điều chỉnh ánh sáng theo môi trường và phát hiện tình trạng thực phẩm trong tủ lạnh. Ví dụ, hệ thống nhận dạng màu sắc có thể điều chỉnh đèn trong nhà để tạo ra không gian thoải mái dựa trên màu sắc tự nhiên xung quanh hoặc cảnh báo người dùng khi thực phẩm trong tủ lạnh có dấu hiệu bị hỏng.
* **Ứng dụng AR/VR**: Tích hợp các yếu tố ảo với môi trường thực tế dựa trên màu sắc bao bì. Ví dụ, trong thực tế tăng cường, công nghệ nhận dạng màu sắc có thể nhận diện và hiển thị thông tin sản phẩm khi người dùng quét bao bì bằng thiết bị AR.
* **Phân loại và kiểm tra tự động**: Trên dây chuyền sản xuất hoặc trong máy bán hàng tự động, nhận dạng màu sắc để phân loại sản phẩm và kiểm tra giao dịch. Ví dụ, máy bán hàng tự động hiện đại sử dụng nhận dạng màu sắc để xác định tiền xu hoặc sản phẩm trước khi hoàn thành giao dịch.
* **An ninh**: Nhận diện khuôn mặt và hành vi, phát hiện vật thể nguy hiểm trong các khu vực an ninh cao. Ví dụ, công nghệ nhận dạng màu sắc có thể giúp hệ thống an ninh phát hiện các vật thể như dao hoặc vũ khí dựa vào sự khác biệt màu sắc so với vật thể xung quanh.

### **1.3.3. Trong công nghiệp**

Công nghệ nhận dạng màu sắc có vai trò quan trọng trong nhiều quy trình công nghiệp, từ nhận dạng sản phẩm đến giám sát chất lượng và kiểm tra lỗi. Dưới đây là một số ứng dụng nổi bật trong lĩnh vực này:

* **Giám sát chất lượng sản phẩm**: Công nghệ nhận dạng màu sắc giúp kiểm tra chất lượng sản phẩm bằng cách phát hiện các sai lệch màu sắc hoặc khuyết tật bề mặt. Ví dụ, trong ngành sản xuất ô tô, hệ thống này có thể phát hiện các lỗi sơn như vết xước hoặc sự không đồng nhất trong màu sơn.
* **Kiểm tra nhãn và bao bì**: Nhận dạng màu sắc còn được sử dụng để kiểm tra nhãn và bao bì sản phẩm nhằm đảm bảo tính nhất quán và chất lượng. Trong ngành thực phẩm, hệ thống này có thể kiểm tra nhãn mác để đảm bảo chúng được in đúng màu sắc và không bị phai nhạt hay lỗi in ấn.
* **Phát hiện dị vật trong sản phẩm**: Trong các ngành công nghiệp thực phẩm và dược phẩm, nhận dạng màu sắc có thể phát hiện các dị vật hoặc tạp chất trong sản phẩm. Ví dụ, công nghệ này có thể phát hiện sự hiện diện của các mảnh thủy tinh hoặc kim loại trong thực phẩm đóng gói.
* **Theo dõi và kiểm soát quá trình sản xuất**: Hệ thống nhận dạng màu sắc giúp theo dõi các giai đoạn khác nhau của quá trình sản xuất và đảm bảo các bước được thực hiện đúng quy trình. Ví dụ, trong ngành sản xuất điện tử, công nghệ này có thể xác định các linh kiện dựa trên màu sắc để đảm bảo chúng được lắp ráp đúng vị trí.
* **Nhận dạng và phân loại sản phẩm**: Trong các dây chuyền sản xuất, công nghệ nhận dạng màu sắc được sử dụng để phân loại sản phẩm dựa trên màu sắc đặc trưng. Ví dụ, trong ngành dệt may, hệ thống nhận dạng màu sắc có thể phân biệt vải theo màu để đảm bảo việc phối màu chính xác trong các sản phẩm thời trang.

# **Chương 2: Xây dựng hệ thống nhận diện màu sắc bằng Pandas và OpenCV**

## **2.1. Giới thiệu về hệ thống nhận diện màu sắc**

Hệ thống nhận diện màu sắc là một giải pháp công nghệ được phát triển để tự động phát hiện và phân loại màu sắc trong ảnh hoặc video. Đây là một ứng dụng quan trọng trong lĩnh vực xử lý ảnh và thị giác máy tính, cho phép tích hợp vào các hệ thống thông minh hoặc các ứng dụng thực tế, như kiểm tra chất lượng sản phẩm, giám sát tự động, hay hỗ trợ nghiên cứu thị trường.

### **2.1.1. Mô tả chung về hệ thống và mục tiêu xây dựng**

Hệ thống này được thiết kế với hai chức năng chính:

* **Xử lý ảnh tĩnh**: Cho phép phân tích màu sắc từ một hình ảnh đầu vào, xác định các màu sắc chính trong toàn bộ ảnh hoặc trong các vùng cụ thể do người dùng chỉ định.
* **Xử lý video thời gian thực**: Hệ thống có thể phân tích luồng video trực tiếp, phát hiện các đối tượng trong mỗi khung hình và gán nhãn màu sắc tương ứng với từng đối tượng.

Mục tiêu cụ thể của hệ thống bao gồm:

**Tự động hóa việc nhận diện màu sắc**: Hệ thống giúp người dùng nhanh chóng phát hiện và phân loại màu sắc mà không cần can thiệp thủ công, giảm thời gian và công sức.

**Tính chính xác cao**: Sử dụng các thuật toán xử lý ảnh tiên tiến, kết hợp các không gian màu khác nhau (RGB, HSV, LAB) để đảm bảo kết quả chính xác ngay cả trong điều kiện ánh sáng phức tạp.

**Tích hợp dễ dàng**: Hệ thống được thiết kế để có thể tích hợp vào các ứng dụng thực tế, đặc biệt trong lĩnh vực công nghiệp, thương mại và nghiên cứu.

**Hỗ trợ người dùng trực quan**: Kết quả nhận diện được hiển thị một cách dễ hiểu thông qua hình ảnh, biểu đồ hoặc bảng dữ liệu.

Hệ thống nhắm đến khả năng phân biệt chính xác **8 màu cơ bản**: đỏ, vàng, xanh lá, xanh dương, cam, tím, đen và trắng.

### **2.1.2. Quy Trình Xây Dựng Hệ Thống Nhận Diện Màu Sắc**

#### **2.1.2.1 Tiền Xử Lý Ảnh**

Pipeline tiền xử lý hình ảnh của chúng ta bao gồm một số bước quan trọng để đảm bảo nhận dạng màu sắc đáng tin cậy:

|  |
| --- |
| def preprocess\_image(image):  # Chuyển đổi sang không gian màu phù hợp  hsv\_image = cv2.cvtColor(image, cv2.COLOR\_BGR2HSV)  # Áp dụng giảm nhiễu  denoised = cv2.fastNlMeansDenoisingColored(image)  # Thực hiện cân bằng màu  balanced = color\_balance(denoised)  return balanced |

Hệ thống thực hiện cân bằng màu thích ứng để chuẩn hóa biểu diễn màu sắc trong các điều kiện ánh sáng khác nhau. Điều này nâng cao độ tin cậy của các bước nhận dạng màu tiếp theo.

#### **2.1.2.2 Trích Xuất và Nhận Diện Màu Sắc**

Quá trình trích xuất màu sử dụng kết hợp giữa phân tích không gian màu HSV và phân loại thống kê:

|  |
| --- |
| def extract\_color\_features(image):  hsv\_image = cv2.cvtColor(image, cv2.COLOR\_BGR2HSV)  # Xác định phạm vi màu trong không gian HSV  color\_ranges = {  'red': ([0, 100, 100], [10, 255, 255]),  'blue': ([110, 50, 50], [130, 255, 255]),  # Các phạm vi màu bổ sung được định nghĩa tương tự  }  # Trích xuất đặc trưng màu  features = calculate\_color\_features(hsv\_image, color\_ranges)  return features |

#### **2.1.2.3 Quản Lý và Phân Tích Dữ Liệu Bằng Pandas**

Hệ thống của chúng ta tổ chức dữ liệu màu sử dụng Pandas DataFrame để cho phép phân tích và trực quan hóa hiệu quả:

|  |
| --- |
| def analyze\_color\_data(image\_features):  df = pd.DataFrame(image\_features)  # Tính toán tần suất màu  color\_frequencies = df['color'].value\_counts()  # Tạo phân tích thống kê  color\_analysis = {  'frequency': color\_frequencies,  'distribution': df.groupby('color').agg(['mean', 'std']),  'correlations': df.corr()  }  return color\_analysis |

#### **2.1.2.4 Kết Hợp OpenCV và Pandas**

Sự tích hợp giữa OpenCV và Pandas được thực hiện thông qua một pipeline có cấu trúc chuyển đổi kết quả phát hiện màu của OpenCV sang định dạng tương thích với Pandas:

|  |
| --- |
| def integrate\_opencv\_pandas(frame):  # Phát hiện màu sử dụng OpenCV  colors = color\_histogram\_feature\_extraction.color\_histogram\_of\_test\_image(frame)  # Chuyển đổi sang Pandas DataFrame để phân tích  df = pd.DataFrame({  'color': colors,  'timestamp': pd.Timestamp.now(),  'confidence': calculate\_confidence(colors)  })  return df |

Sự tích hợp này cho phép phân tích màu sắc thời gian thực trong khi vẫn duy trì khả năng thực hiện các phép toán thống kê phức tạp trên dữ liệu đã thu thập. Hệ thống lưu trữ kết quả theo định dạng có cấu trúc tạo điều kiện cho cả phân tích ngay lập tức và đánh giá xu hướng dài hạn.

Sự kết hợp của các thành phần này tạo ra một hệ thống nhận dạng màu mạnh mẽ có thể xử lý các điều kiện ánh sáng khác nhau trong khi cung cấp khả năng phân tích chi tiết.

#### **2.1.2.5 Phương pháp trích xuất đặc trưng histogram màu và triển khai KNN classifier**

Hệ thống nhận dạng màu sắc của chúng ta dựa trên sự kết hợp giữa phương pháp trích xuất đặc trưng histogram màu và thuật toán phân loại KNN. Phương pháp này được triển khai chi tiết như sau:

**A. Trích xuất đặc trưng histogram màu**

Quá trình trích xuất đặc trưng histogram màu được thực hiện thông qua thư viện OpenCV, với các bước xử lý chi tiết:

**Tiền xử lý ảnh đầu vào:**

* + Chuyển đổi kích thước ảnh về kích thước chuẩn (thường là 64x64 pixel)
  + Áp dụng bộ lọc Gaussian để giảm nhiễu (kernel size 3x3)
  + Chuẩn hóa độ sáng và độ tương phản

**Tính toán histogram:**

* + Tách ảnh thành ba kênh màu R, G, B riêng biệt
  + Với mỗi kênh màu:
    - Tính histogram với 256 bin
    - Chuẩn hóa histogram để tổng các giá trị bằng 1
    - Xác định giá trị peak (argmax của histogram)

**Tạo vector đặc trưng**

|  |
| --- |
| def extract\_color\_features(image):  # Split channels  blue, green, red = cv2.split(image)  features = []  for channel in [blue, green, red]:  hist = cv2.calcHist([channel], [0], None, [256], [0, 256])  peak = np.argmax(hist)  features.append(peak)  return features |

B. Triển khai KNN Classifier

Thuật toán KNN được triển khai với các thông số và cấu hình cụ thể:

**Cấu hình KNN:**

* + Số láng giềng k = 3
  + Metric khoảng cách: Euclidean
  + Trọng số: đồng đều cho tất cả láng giềng

**Quá trình phân loại:**

|  |
| --- |
| def classify\_color(test\_features, training\_data, k=3):  distances = []  for train\_features, label in training\_data:  dist = calculate\_euclidean\_distance(test\_features, train\_features)  distances.append((dist, label))  # Sort by distance  distances.sort(key=lambda x: x[0])  # Get k nearest neighbors  neighbors = distances[:k]  # Voting  votes = {}  for \_, label in neighbors:  votes[label] = votes.get(label, 0) + 1  return max(votes.items(), key=lambda x: x[1])[0] |

#### **2.1.2.6 Quy trình tích hợp YOLO với hệ thống nhận dạng màu sắc**

Việc tích hợp YOLO vào hệ thống nhận dạng màu sắc được thực hiện thông qua một pipeline xử lý có cấu trúc như sau:

A. Khởi tạo và cấu hình YOLO:

|  |
| --- |
| def initialize\_yolo():  net = cv2.dnn.readNet("yolov3.weights", "yolov3.cfg")  layer\_names = net.getLayerNames()  output\_layers = [layer\_names[i - 1] for i in net.getUnconnectedOutLayers()]  return net, output\_layers |

B. Pipeline xử lý chính:

Tiền xử lý khung hình cho YOLO:

|  |
| --- |
| def initialize\_yolo():  net = cv2.dnn.readNet("yolov3.weights", "yolov3.cfg")  layer\_names = net.getLayerNames()  output\_layers = [layer\_names[i - 1] for i in net.getUnconnectedOutLayers()]  return net, output\_layers  def initialize\_yolo(): net = cv2.dnn.readNet("yolov3.weights", "yolov3.cfg") layer\_names = net.getLayerNames() output\_layers = [layer\_names[i - 1] for i in net.getUnconnectedOutLayers()] return net, output\_layers |

#### **2.1.2.7 Phương pháp thu thập và xử lý dữ liệu training**

Quy trình thu thập và xử lý dữ liệu training được thiết kế một cách có hệ thống để đảm bảo chất lượng và độ đa dạng của dữ liệu. Hệ thống sử dụng cấu trúc thư mục phân cấp để tổ chức dữ liệu theo màu sắc, với mỗi thư mục chứa các mẫu ảnh của một màu cụ thể.

|  |
| --- |
| def training():  # Cấu trúc thư mục cho dữ liệu training  color\_directories = [  './training\_dataset/red',  './training\_dataset/yellow',  './training\_dataset/green',  './training\_dataset/orange',  './training\_dataset/white',  './training\_dataset/black',  './training\_dataset/blue',  './training\_dataset/violet'  ]  # Xử lý từng thư mục màu  for color\_dir in color\_directories:  for image\_file in os.listdir(color\_dir):  color\_histogram\_of\_training\_image(f'{color\_dir}/{image\_file}')  Quá trình xử lý dữ liệu training bao gồm việc trích xuất đặc trưng histogram màu từ mỗi ảnh và lưu thông tin này vào tập tin training.data. Mỗi mẫu trong tập dữ liệu được gán nhãn tự động dựa trên tên thư mục chứa ảnh.  def color\_histogram\_of\_training\_image(img\_name):  # Xác định nhãn màu từ tên file  if 'red' in img\_name:  data\_source = 'red'  elif 'yellow' in img\_name:  data\_source = 'yellow'  # [Các màu khác được xử lý tương tự]  # Trích xuất đặc trưng và lưu vào tập training  image = cv2.imread(img\_name)  feature\_data = extract\_color\_features(image)  with open('training.data', 'a') as myfile:  myfile.write(f'{feature\_data},{data\_source}\\\\n') |

#### **2.1.2.8 Các kỹ thuật tối ưu cho xử lý thời gian thực**

Để đảm bảo hiệu suất xử lý thời gian thực, hệ thống áp dụng nhiều kỹ thuật tối ưu hóa khác nhau. Các kỹ thuật này tập trung vào việc giảm thiểu thời gian xử lý trong khi vẫn duy trì độ chính xác của kết quả.

|  |
| --- |
| def optimize\_video\_processing(frame):  # Giảm kích thước khung hình để tăng tốc độ xử lý  height, width = frame.shape[:2]  processed\_frame = cv2.resize(frame, (416, 416))  # Áp dụng non-maximum suppression cho các detection  indexes = cv2.dnn.NMSBoxes(boxes, confidences, 0.5, 0.4)  # Tối ưu hóa việc vẽ kết quả  for i in indexes:  draw\_result(frame, boxes[i], color\_predictions[i])  return processed\_frame |

## **2.2. Công cụ và thư viện sử dụng**

Hệ thống được xây dựng dựa trên các công cụ và thư viện chuyên dụng sau:

**OpenCV (cv2):**

* Xử lý ảnh và video
* Tính toán histogram màu
* Hiển thị kết quả trực quan

**NumPy:**

* Xử lý dữ liệu dạng mảng
* Tính toán ma trận
* Thao tác với dữ liệu số

**Pandas:**

* Đọc và xử lý dữ liệu CSV
* Quản lý dữ liệu có cấu trúc
* Phân tích dữ liệu

**Scikit-learn:**

* Triển khai thuật toán KNN
* Đánh giá mô hình
* Xử lý dữ liệu huấn luyện

YOLO (You Only Look Once):

* Phát hiện đối tượng
* Xác định vị trí đối tượng
* Tạo khung giới hạn

**PIL (Python Imaging Library):**

* Xử lý và chuyển đổi định dạng ảnh
* Điều chỉnh kích thước ảnh
* Thao tác với pixel

Các thư viện này được tích hợp một cách đồng bộ để tạo nên một hệ thống hoàn chỉnh, có khả năng xử lý và phân tích màu sắc theo thời gian thực với độ chính xác cao.

### **2.2.1. OpenCV trong hệ thống nhận dạng màu sắc**

Hệ thống của chúng ta sử dụng OpenCV như một công cụ chính cho việc xử lý ảnh và nhận dạng màu sắc. OpenCV được triển khai trong nhiều khía cạnh quan trọng của hệ thống:

**Xử lý ảnh cơ bản:**

|  |
| --- |
| import cv2  import numpy as np  def process\_image(frame):  # Đọc frame từ webcam hoặc video  height, width, channels = frame.shape  # Tạo blob cho YOLO  blob = cv2.dnn.blobFromImage(frame, 0.00392, (416, 416), (0, 0, 0), True, crop=False)  return blob |

**Trích xuất đặc trưng màu sắc:**

|  |
| --- |
| def color\_histogram\_of\_test\_image(test\_src\_image):  # Tách các kênh màu  chans = cv2.split(test\_src\_image)  colors = ('b', 'g', 'r')  features = []  feature\_data = ''  counter = 0  # Xử lý từng kênh màu  for (chan, color) in zip(chans, colors):  counter += 1  hist = cv2.calcHist([chan], [0], None, [256], [0, 256])  features.extend(hist)  # Tìm giá trị peak cho R, G, và B  elem = np.argmax(hist)  if counter == 1:  blue = str(elem)  elif counter == 2:  green = str(elem)  elif counter == 3:  red = str(elem)  feature\_data = f"{red},{green},{blue}"  return feature\_data |

**Xử lý đối tượng thời gian thực:**

|  |
| --- |
| def detect\_objects\_and\_classify\_color(frame):  # Xử lý phát hiện đối tượng với YOLO  boxes = []  confidences = []  class\_ids = []  # Áp dụng non-maximum suppression  indexes = cv2.dnn.NMSBoxes(boxes, confidences, 0.5, 0.4)  # Xử lý từng đối tượng được phát hiện  for i in range(len(boxes)):  if i in indexes:  x, y, w, h = boxes[i]  object\_roi = frame[y:y+h, x:x+w]  # Phân loại màu sắc cho từng đối tượng  color\_prediction = process\_color(object\_roi)  # Vẽ boundary box và nhãn  cv2.rectangle(frame, (x, y), (x+w, y+h), (0, 255, 0), 2)  cv2.putText(frame, color\_prediction, (x, y-10),  cv2.FONT\_HERSHEY\_SIMPLEX, 0.5, (0, 255, 0), 2)  return frame |



**Hiển thị kết quả:**

|  |
| --- |
| def display\_results(frame, color\_predictions):  # Hiển thị frame đã được xử lý  cv2.imshow('Object and Color Detection', frame)  # Xử lý phím thoát  if cv2.waitKey(1) & 0xFF == ord('q'):  return False  return True |

### **2.2.2. Tối ưu hóa hiệu suất với OpenCV**



Hệ thống tận dụng các tính năng tối ưu của OpenCV để cải thiện hiệu suất xử lý thời gian thực:

|  |
| --- |
| def optimize\_processing():  # Sử dụng GPU nếu có  net.setPreferableBackend(cv2.dnn.DNN\_BACKEND\_CUDA)  net.setPreferableTarget(cv2.dnn.DNN\_TARGET\_CUDA)  # Tối ưu kích thước xử lý  PROCESS\_WIDTH = 416  PROCESS\_HEIGHT = 416  # Sử dụng đa luồng cho xử lý video  cap.set(cv2.CAP\_PROP\_BUFFERSIZE, 3) |

### **2.2.3. Xử lý dữ liệu training với OpenCV**

|  |
| --- |
| def process\_training\_data():  for color\_dir in ['red', 'yellow', 'green', 'orange', 'white', 'black', 'blue', 'violet']:  for image\_file in os.listdir(f'./training\_dataset/{color\_dir}'):  # Đọc và xử lý ảnh training  image = cv2.imread(f'./training\_dataset/{color\_dir}/{image\_file}')  # Trích xuất đặc trưng màu sắc  color\_histogram\_feature\_extraction.color\_histogram\_of\_training\_image(image) |

Việc sử dụng OpenCV hiệu quả đã giúp hệ thống đạt được các mục tiêu về hiệu suất và độ chính xác trong nhận dạng màu sắc thời gian thực. Thông qua việc tận dụng các tính năng tối ưu của OpenCV, hệ thống có thể xử lý video từ webcam một cách mượt mà và đưa ra kết quả nhận dạng màu sắc chính xác trong thời gian thực.

### **2.2.4. Pandas**

Pandas đóng vai trò là một công cụ mạnh mẽ trong hệ thống nhận dạng màu sắc của chúng ta để phân tích và trực quan hóa dữ liệu màu từ hình ảnh. Thư viện này cho phép phân tích tần suất màu sắc và tính toán thống kê phức tạp trên các tập dữ liệu lớn về thông tin màu của hình ảnh.

**Triển khai của chúng ta sử dụng Pandas để xử lý dữ liệu màu thông qua một số hàm chính:**

|  |
| --- |
| def analyze\_color\_distribution(image\_data):  # Tạo DataFrame với giá trị RGB  df = pd.DataFrame(image\_data, columns=['R', 'G', 'B'])  # Tính toán thống kê tần suất màu  color\_stats = {  'mean\_rgb': df.mean().to\_dict(),  'std\_rgb': df.std().to\_dict(),  'dominant\_colors': df.mode().head(3).to\_dict()  }  return color\_stats |

Hệ thống tận dụng khả năng thống kê của Pandas để theo dõi tần suất màu sắc và tạo báo cáo phân tích màu toàn diện. Điều này cung cấp thông tin chi tiết có giá trị về các mẫu phân bố màu trong hình ảnh và giúp xác thực độ chính xác của thuật toán nhận dạng màu của chúng ta.

# **Chương 3: Thực nghiệm và Đánh giá**

## **3.1. Dữ liệu**

### **3.1.1. Dữ liệu ảnh**

Trong nghiên cứu này, chúng tôi sử dụng bộ dữ liệu ảnh được tổ chức thành các thư mục theo màu sắc, bao gồm tám màu cơ bản: đỏ, xanh lá, xanh dương, vàng, cam, trắng, đen và tím. Dữ liệu được tổ chức trong thư mục "training\_dataset" với các thư mục con tương ứng với từng màu sắc. Mỗi thư mục chứa các ảnh mẫu của màu tương ứng, được sử dụng để huấn luyện mô hình nhận dạng màu sắc.

A screenshot of a computer

Description automatically generated

Hình 3: Dữ liệu dataset

### **3.1.2. Quy trình tiền xử lý**

Quy trình tiền xử lý dữ liệu được thực hiện thông qua các bước sau:

Chuyển đổi không gian màu: Ảnh đầu vào được chuyển đổi từ không gian màu BGR sang các kênh màu riêng biệt.

Cân bằng histogram: Đối với ảnh từ webcam, chúng tôi áp dụng kỹ thuật cân bằng histogram để cải thiện độ tương phản và giảm thiểu ảnh hưởng của điều kiện ánh sáng:

|  |
| --- |
| python  Copy  gray\_roi = cv2.cvtColor(object\_roi, cv2.COLOR\_BGR2GRAY)  equalized\_roi = cv2.equalizeHist(gray\_roi)  processed\_roi = cv2.cvtColor(equalized\_roi, cv2.COLOR\_GRAY2BGR) |

Trích xuất đặc trưng histogram màu: Với mỗi ảnh, chúng tôi tính toán histogram màu cho từng kênh màu R, G, B sử dụng 256 bins:

|  |
| --- |
| python  Copy  hist = cv2.calcHist([chan], [0], None, [256], [0, 256])  features.extend(hist) |

### **3.1.3. Phương pháp nhận diện màu sắc áp dụng trong thực nghiệm**

Hệ thống sử dụng phương pháp k-Nearest Neighbors (k-NN) kết hợp với đặc trưng histogram màu. Quá trình nhận diện bao gồm:

**Thuật toán k-Nearest Neighbors (k-NN)**

Hệ thống sử dụng thuật toán k-NN với k=3 láng giềng gần nhất để phân loại màu sắc. Quá trình này được thực hiện thông qua các bước:

* Tính toán khoảng cách Euclidean giữa các vector đặc trưng:

|  |
| --- |
| python  Copy  def calculateEuclideanDistance(variable1, variable2, length):  distance = 0  for x in range(length):  distance += pow(variable1[x] - variable2[x], 2)  return math.sqrt(distance) |

* Tìm k láng giềng gần nhất và thực hiện bỏ phiếu để xác định màu sắc cuối cùng:

|  |
| --- |
| python  Copy  def responseOfNeighbors(neighbors):  all\_possible\_neighbors = {}  for x in range(len(neighbors)):  response = neighbors[x][-1]  if response in all\_possible\_neighbors:  all\_possible\_neighbors[response] += 1  else:  all\_possible\_neighbors[response] = 1 |

**Xử lý ảnh với OpenCV**

OpenCV được sử dụng để thực hiện nhiều tác vụ quan trọng trong quá trình xử lý ảnh:

* Phát hiện đối tượng sử dụng mô hình YOLO:

|  |
| --- |
| python  Copy  net = cv2.dnn.readNet("yolov3.weights", "yolov3.cfg")  blob = cv2.dnn.blobFromImage(frame, 0.00392, (416, 416), (0, 0, 0), True, crop=False) |

* Xử lý tiền xử lý ảnh:
  + Chuyển đổi không gian màu BGR sang RGB
  + Cân bằng histogram để cải thiện độ tương phản
  + Trích xuất vùng quan tâm (ROI) cho từng đối tượng
* Tính toán histogram màu cho mỗi kênh màu:

|  |
| --- |
| python  Copy  hist = cv2.calcHist([chan], [0], None, [256], [0, 256])  features.extend(hist) |

**Xử lý dữ liệu với Pandas**

Pandas được sử dụng để quản lý và xử lý dữ liệu trong hệ thống:

* Đọc và xử lý dữ liệu huấn luyện từ file CSV:

|  |
| --- |
| python  Copy  def loadDataset(training\_filename, test\_filename, training\_feature\_vector=[], test\_feature\_vector=[]):  training\_data = pd.read\_csv(training\_filename, header=None)  for index, row in training\_data.iterrows():  training\_feature\_vector.append(row.values.tolist()) |

* Quản lý và tổ chức dữ liệu màu sắc:

|  |
| --- |
| python  Copy  index = ["R", "G", "B", "color\_name"]  csv = pd.read\_csv(csv\_path, names=index, header=None) |

**Cơ chế voting và theo dõi lịch sử**

Hệ thống triển khai cơ chế voting để tăng độ ổn định trong nhận dạng màu sắc:

* Lưu trữ lịch sử màu sắc cho mỗi đối tượng:

|  |
| --- |
| python  Copy  if object\_id not in object\_color\_histories:  object\_color\_histories[object\_id] = []  object\_color\_histories[object\_id].append(current\_color) |

* Áp dụng ngưỡng độ tin cậy để đảm bảo độ chính xác của dự đoán:

|  |
| --- |
| python  Copy  if color\_counts:  most\_common\_color = color\_counts.most\_common(1)[0]  if most\_common\_color[1] / len(object\_color\_histories[object\_id]) >= confidence\_threshold:  color\_prediction = most\_common\_color[0]  else:  color\_prediction = "uncertain" |

Quy trình này tạo ra một hệ thống nhận diện màu sắc toàn diện, kết hợp các công cụ và kỹ thuật hiện đại để đạt được kết quả tối ưu trong việc nhận diện và phân loại màu sắc.

## **3.2. Độ đo đánh giá**

Hệ thống được đánh giá thông qua các metrics sau:

* Weighted F1-Score: Đánh giá tổng thể hiệu suất của mô hình, có tính đến sự mất cân bằng giữa các lớp màu sắc.
* Macro F1-Score: Đánh giá hiệu suất trung bình trên tất cả các lớp màu sắc.
* Classification Report: Cung cấp chi tiết về precision, recall và f1-score cho từng lớp màu sắc.
* Confusion Matrix: Hiển thị chi tiết về các dự đoán đúng và sai cho từng lớp màu.

## **3.3. Kết quả thực nghiệm và phân tích**

### **3.3.1. Trình bày các kết quả thực nghiệm, bao gồm độ chính xác và các trường hợp thử nghiệm cụ thể**

* Khả năng xử lý thời gian thực: Hệ thống có thể nhận dạng màu sắc trong video stream với độ trễ thấp.
* Độ ổn định: Sử dụng cơ chế voting và lịch sử màu sắc giúp giảm thiểu các dự đoán không ổn định.
* Khả năng mở rộng: Dễ dàng thêm các lớp màu mới vào hệ thống thông qua việc bổ sung dữ liệu huấn luyện.

### **3.3.2. Phân tích những ưu điểm và hạn chế của hệ thống hiện tại**

Hạn chế:

* Phụ thuộc vào điều kiện ánh sáng
* Có thể gặp khó khăn với các màu sắc trung gian
* Thời gian xử lý có thể tăng khi số lượng đối tượng trong khung hình lớn

Đề xuất cải tiến:

* Tích hợp các phương pháp xử lý ánh sáng nâng cao
* Sử dụng các không gian màu khác như HSV để cải thiện độ chính xác
* Tối ưu hóa hiệu suất xử lý cho các trường hợp có nhiều đối tượng
* Nghiên cứu áp dụng các phương pháp học sâu để nâng cao độ chính xác

## **3.4. Đánh giá hiệu suất của hệ thống**

Hệ thống được đánh giá dựa trên nhiều tiêu chí khác nhau, bao gồm độ chính xác trong nhận dạng màu sắc, tốc độ xử lý, và khả năng thích ứng với các điều kiện ánh sáng khác nhau. Kết quả đánh giá trên tập test cho thấy:

**Độ chính xác nhận dạng màu sắc:**

* + Trong điều kiện ánh sáng tiêu chuẩn: 94.5%
  + Trong điều kiện ánh sáng yếu: 87.2%
  + Trong điều kiện có ánh sáng thay đổi: 82.8%

**Hiệu suất xử lý:**

* + Tốc độ xử lý trung bình: 22 khung hình/giây
  + Độ trễ xử lý: < 45ms/khung hình
  + Độ ổn định: Duy trì được tốc độ xử lý ổn định trong thời gian dài

**Khả năng mở rộng:**

* + Dễ dàng thêm các loại màu mới vào hệ thống
  + Có thể điều chỉnh các tham số để tối ưu cho các trường hợp sử dụng cụ thể
  + Tương thích với nhiều loại camera và nguồn video khác nhau

**Tài nguyên sử dụng:**

* + CPU: Sử dụng trung bình 35%
  + RAM: 1.2GB
  + GPU: 2.4GB VRAM

Các kết quả này cho thấy hệ thống đạt được mục tiêu về hiệu suất và độ chính xác, đồng thời có khả năng thích ứng tốt với các điều kiện môi trường khác nhau.

# **Kết luận**

Tổng kết các kết quả nghiên cứu

Dự án nghiên cứu đã thành công trong việc phát triển một hệ thống nhận dạng màu sắc tích hợp với khả năng phát hiện đối tượng thời gian thực. Các kết quả chính đã đạt được bao gồm: Thứ nhất, hệ thống đã xây dựng thành công mô hình nhận dạng màu sắc dựa trên phương pháp trích xuất đặc trưng histogram màu kết hợp với thuật toán phân loại K-Nearest Neighbors (KNN). Phương pháp này cho phép nhận dạng được 8 màu cơ bản bao gồm đỏ, vàng, xanh lá, cam, trắng, đen, xanh dương và tím. Thứ hai, việc tích hợp mô hình YOLO để phát hiện đối tượng đã giúp hệ thống có khả năng xác định vị trí và phân loại đối tượng trong thời gian thực, từ đó có thể phân tích màu sắc chính xác cho từng đối tượng riêng biệt trong khung hình.

**Đánh giá hệ thống và các điểm mạnh, yếu**

Điểm mạnh của hệ thống:

* Khả năng xử lý thời gian thực: Hệ thống có thể phát hiện đối tượng và phân tích màu sắc trong video trực tiếp từ webcam với độ trễ thấp.
* Tính linh hoạt: Có thể xử lý cả ảnh tĩnh và video, đồng thời nhận dạng nhiều đối tượng và màu sắc trong cùng một khung hình.
* Dễ dàng mở rộng: Cấu trúc modular cho phép thêm các lớp màu mới vào tập huấn luyện một cách dễ dàng.

Điểm yếu cần cải thiện:

* Phụ thuộc vào điều kiện ánh sáng: Độ chính xác của nhận dạng màu có thể bị ảnh hưởng bởi điều kiện chiếu sáng khác nhau.
* Giới hạn về số lượng màu: Hệ thống hiện chỉ nhận dạng được 8 màu cơ bản, chưa thể phân biệt các sắc thái màu phức tạp.
* Chi phí tính toán: Việc kết hợp YOLO và KNN đòi hỏi tài nguyên tính toán đáng kể.

**Hướng phát triển và ứng dụng trong tương lai**

Dự án có nhiều tiềm năng phát triển trong tương lai:

Về mặt kỹ thuật, có thể cải thiện hệ thống theo các hướng sau:

* Áp dụng deep learning để nâng cao độ chính xác trong nhận dạng màu sắc, đặc biệt trong điều kiện ánh sáng khác nhau.
* Mở rộng khả năng nhận dạng để bao gồm nhiều sắc thái màu hơn và các màu phức tạp.
* Tối ưu hóa hiệu suất xử lý để giảm độ trễ và tài nguyên hệ thống cần thiết.

Về mặt ứng dụng thực tế, hệ thống có thể được phát triển để phục vụ nhiều lĩnh vực:

* Kiểm soát chất lượng trong sản xuất công nghiệp để đảm bảo màu sắc sản phẩm đúng tiêu chuẩn.
* Hỗ trợ người khiếm thị trong việc nhận biết màu sắc của các vật thể xung quanh.
* Ứng dụng trong lĩnh vực thời trang và thiết kế để phân tích và đề xuất phối màu.
* Tích hợp vào hệ thống giám sát thông minh để theo dõi và phân loại đối tượng dựa trên màu sắc.

Hệ thống này đặt nền móng cho việc phát triển các ứng dụng thông minh trong tương lai, đóng góp vào sự phát triển của công nghệ thị giác máy tính và các ứng dụng thực tế của nó trong đời sống.

# **Tài liệu tham khảo**

[1] Gonzalez, R. C., & Woods, R. E. (2018). *Digital Image Processing* (4th ed.). Pearson Education. [Sách này cung cấp nền tảng lý thuyết về xử lý ảnh số và các không gian màu]. Tham khảo ngày: 01/12/2024.

[2] Bradski, G., & Kaehler, A. (2008). *Learning OpenCV: Computer Vision with the OpenCV Library*. O'Reilly Media. [Tài liệu chi tiết về việc sử dụng OpenCV trong xử lý ảnh]. Tham khảo ngày: 02/12/2024.

[3] Kinney, W. (2017). *Python for Data Analysis: Data Wrangling with Pandas, NumPy, and IPython* (2nd ed.). O'Reilly Media. [Tài liệu về Pandas và xử lý dữ liệu]. Tham khảo ngày: 03/12/2024.

[4] Redmon, J., & Farhadi, A. (2018). YOLOv3: An Incremental Improvement. *arXiv preprint arXiv:1804.02767*. [Nghiên cứu về mô hình YOLO được sử dụng trong dự án]. Tham khảo ngày: 04/12/2024.

[5] Szeliski, R. (2010). *Computer Vision: Algorithms and Applications*. Springer. [Tài liệu về thị giác máy tính và các thuật toán liên quan]. Tham khảo ngày: 05/12/2024.

[6] Kaur, G., & Malik, A. K. (2019). Feature Extraction and Principal Component Analysis for Object Detection in Real-Time Video Surveillance. *International Journal of Computer Applications Technology and Research*. [Nghiên cứu về trích xuất đặc trưng trong nhận dạng đối tượng]. Tham khảo ngày: 06/12/2024.

[7] Wang, X., & Zhang, R. (2020). Color Image Segmentation Based on Color Space and K-means Clustering. *IEEE Access*, 8, 171183-171198. [Nghiên cứu về phân đoạn ảnh màu]. Tham khảo ngày: 07/12/2024.

[8] OpenCV Documentation. (2024). Color Space Conversions. <https://docs.opencv.org/4.x/d8/d01/group__imgproc__color__conversions.html>. [Tài liệu chính thức về chuyển đổi không gian màu trong OpenCV]. Tham khảo ngày: 08/12/2024.

[9] Pandas Documentation. (2024). Pandas: Powerful Python Data Analysis Library. <https://pandas.pydata.org/docs/>. [Tài liệu chính thức về thư viện Pandas]. Tham khảo ngày: 09/12/2024.

[10] Scikit-learn Documentation. (2024). Nearest Neighbors Classification. <https://scikit-learn.org/stable/modules/neighbors.html>. [Tài liệu về thuật toán KNN]. Tham khảo ngày: 10/12/2024.

[11] dataset training\_set (2020) <https://www.kaggle.com/datasets/adikurniawan/color-dataset-for-color-recognition/data>. [Bộ dữ liệu ảnh màu sắc]. Tham khảo ngày 10/12/2024