

**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	20210864	Nguyễn Văn Hải	DC CNTT 12.10.3
2	20210805	Đỗ Minh Hiếu	DC CNTT 12.10.3
3	20210664	Phan Đức Huy	DC CNTT 12.10.3
4	20210724	Nguyễn Văn Tuấn	DC CNTT 12.10.3

**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**

<b>TT</b>	<b>Mã sinh viên</b>	<b>Sinh viên thực hiện</b>	<b>Lớp hành chính</b>
<b>1</b>	<b>20210864</b>	<b>Nguyễn Văn Hải</b>	<b>DC CNTT 12.10.3</b>
<b>2</b>	<b>20210805</b>	<b>Đỗ Minh Hiếu</b>	<b>DC CNTT 12.10.3</b>
<b>3</b>	<b>20210664</b>	<b>Phan Đức Huy</b>	<b>DC CNTT 12.10.3</b>
<b>4</b>	<b>20210724</b>	<b>Nguyễn Văn Tuấn</b>	<b>DC CNTT 12.10.3</b>

**PHIẾU CHẤM THI BÀI TẬP LỚN KẾT THÚC HỌC PHẦN**

**Mã đề thi: 02**

**Tên học phần: XỬ LÝ ẢNH VÀ THỊ GIÁC MÁY TÍNH**

**Lớp Tín chỉ: XATGMT.03.K12.03.LH.C04.1\_LT**

**Cán bộ chấm thi 1**

**Cán bộ chấm thi 2**

**Lương Thị Hồng Lan**

*(Ký và ghi rõ họ tên)*

*(Ký và ghi rõ họ tên)*

TT	TIÊU CHÍ	THANG ĐIỂM	Nguyễn Văn Hải	Đỗ Minh Hiếu	Phan Đức Huy	Nguyễn Văn Tuấn
			20210864	20210805	20210664	20210724
<b>1</b>	<b>Nội dung báo cáo trên Word đầy đủ</b>	<b>3.5</b>				
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				

TT	TIÊU CHÍ	THANG ĐIỂM	Nguyễn Văn Hải	Đỗ Minh Hiếu	Phan Đức Huy	Nguyễn Văn Tuấn
			20210864	20210805	20210664	20210724
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				
<b>2</b>	<b>Nội dung thuyết trình đầy đủ</b>	<b>1.0</b>				
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				
<b>3</b>	<b>Slides báo cáo đầy đủ nội dung + Hỏi đáp</b>	<b>3.0</b>				
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				
<b>4</b>	<b>Code đầy đủ</b>	<b>2.5</b>				
1.1	Code được trình bày rõ ràng, có chú thích đầy đủ.	0,5				

TT	TIÊU CHÍ	THANG ĐIỂM	Nguyễn Văn Hải	Đỗ Minh Hiếu	Phan Đức Huy	Nguyễn Văn Tuấn
			20210864	20210805	20210664	20210724
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				
<b>TỔNG ĐIỂM BẰNG SỐ:</b>		<b>10</b>				
<b>TỔNG ĐIỂM BẰNG CHỮ:</b>		<i>Mười tròn</i>				

# MỤC LỤC

MỤC LỤC .....	6
LỜI NÓI ĐẦU .....	7
CHƯƠNG 1: CƠ SỞ LÝ THUYẾT .....	8
1.1. Biểu diễn ảnh trong máy tính.....	8
1.1.1 Thu nhận ảnh.....	8
1.1.2 Các hệ màu trong máy tính (RGB, HSL, HSV, Grayscale).....	9
1.2 Tổng quan về học máy .....	12
1.2.1. Học có giám sát.....	12
1.2.2. Học không giám sát .....	18
1.2.3 Học bán giám sát.....	20
CHƯƠNG 2. XÂY DỰNG HỆ THỐNG NHẬN DIỆN MÀU SẮC BẰNG PANDAS VÀ OPENCV.....	28
2.1. Thư viện .....	28
2.1.1. Pandas .....	28
2.1.2. Opencv .....	30
2.1.3 Một số thư viện khác.....	34
2.2. Xây dựng hệ thống nhận diện màu sắc trong ảnh.....	35
2.2.1 Các bước thực hiện .....	36
CHƯƠNG 3. THỰC NGHIỆM .....	40
3.1 Dữ liệu .....	40
3.1.1 Dữ liệu hình ảnh.....	40
3.1.2 Dữ liệu từ file CSV .....	40
3.2 Độ đo đánh giá.....	41
3.3 Kết quả thực nghiệm.....	41
3.3.1. Sử dụng SVM .....	41
3.3.2. Sử dụng K-NN .....	42
3.3.3. Sử dụng Decision Tree.....	42
KẾT LUẬN.....	43
TÀI LIỆU THAM KHẢO .....	45

## LỜI NÓI ĐẦU

Trong thời đại công nghệ số phát triển như hiện nay, việc ứng dụng trí tuệ nhân tạo và học máy vào các lĩnh vực của đời sống đã trở nên phổ biến và mang lại nhiều giá trị thiết thực. Một trong những ứng dụng quan trọng của học máy là khả năng nhận diện và phân loại màu sắc, góp phần đáng kể trong nhiều ngành công nghiệp như thiết kế, in ấn, thời trang và công nghệ.

Việc phát triển một hệ thống nhận diện màu sắc chính xác và hiệu quả không chỉ đáp ứng nhu cầu thực tiễn mà còn mở ra nhiều cơ hội nghiên cứu và phát triển trong lĩnh vực thị giác máy tính. Với mong muốn tìm hiểu sâu hơn về lĩnh vực này, tôi đã thực hiện nghiên cứu đề tài "Xây dựng hệ thống nhận diện màu sắc sử dụng Pandas và OpenCV".

Đề tài này tập trung vào việc xây dựng một hệ thống có khả năng nhận diện và phân loại màu sắc từ hình ảnh thu được qua camera, sử dụng các thuật toán học máy phổ biến như K-Nearest Neighbors (KNN), Support Vector Machine (SVM) và Decision Tree. Thông qua việc kết hợp các thư viện mạnh mẽ như Pandas cho xử lý dữ liệu và OpenCV cho xử lý hình ảnh, hệ thống được phát triển nhằm mục đích tạo ra một giải pháp hiệu quả cho bài toán nhận diện màu sắc.

Mặc dù đã có nhiều nghiên cứu về nhận diện màu sắc, việc xây dựng một hệ thống hoàn chỉnh với khả năng xử lý theo thời gian thực vẫn còn nhiều thách thức. Nghiên cứu này không chỉ tập trung vào việc phát triển giải pháp kỹ thuật mà còn đánh giá và so sánh hiệu quả của các thuật toán khác nhau, từ đó đề xuất các cải tiến cho hệ thống.

# CHƯƠNG 1: CƠ SỞ LÝ THUYẾT

## 1.1. Biểu diễn ảnh trong máy tính

### 1.1.1 Thu nhận ảnh

Thu nhận ảnh là bước đầu tiên trong quá trình xử lý và phân tích ảnh trên máy tính. Nó liên quan đến việc chuyển đổi thông tin từ thế giới thực sang dạng số (digital) để có thể xử lý bằng các công cụ và thuật toán.

Quy trình thu nhận ảnh:

- Nguồn tạo ảnh:
  - + Camera kỹ thuật số: Thiết bị chụp ảnh thông qua cảm biến (CMOS hoặc CCD).
  - + Máy quét (Scanner): Chuyển đổi ảnh giấy, văn bản, hoặc phim sang dạng kỹ thuật số.
  - + Ảnh vệ tinh: Thu nhận từ các vệ tinh, thường được sử dụng trong nghiên cứu địa lý và khí tượng.
  - + Hệ thống thu ảnh y tế: Ví dụ như X-quang, MRI, CT scan.
- Chuyển đổi tín hiệu quang học thành tín hiệu số:
  - + Cảm biến ánh sáng: Thiết bị cảm biến ánh sáng (pixel sensor) nhận ánh sáng từ cảnh vật và chuyển đổi thành tín hiệu điện.
  - + Chuyển đổi tín hiệu điện sang số: Bộ chuyển đổi tương tự - số (ADC) chuyển tín hiệu điện thành dữ liệu số (bit).
- Lưu trữ dữ liệu ảnh:
  - + Ảnh sau khi thu nhận được lưu trữ dưới dạng các ma trận số, mỗi điểm ảnh (pixel) đại diện cho một giá trị ánh sáng hoặc màu sắc tại vị trí đó.
  - + Các định dạng phổ biến: JPEG, PNG, BMP, TIFF.
- Ví dụ minh họa:
  - + Khi sử dụng một camera điện thoại thông minh để chụp ảnh:



- + Ánh sáng từ cảnh vật đi qua ống kính và được cảm biến CMOS thu nhận.
- + Dữ liệu ánh sáng được chuyển đổi thành giá trị số và lưu trữ dưới dạng file ảnh (thường là JPEG).
- Tầm quan trọng của thu nhận ảnh:
  - + Đảm bảo rằng chất lượng ảnh đủ tốt để phục vụ các bước xử lý tiếp theo.
  - + Ảnh thu nhận cần đầy đủ thông tin và ít nhiễu nhất có thể để tránh ảnh hưởng đến độ chính xác của phân tích.

## 1.1.2 Các hệ màu trong máy tính (RGB, HSL, HSV, Grayscale)

### 1.1.2.1 Hệ màu RGB (Red, Green, Blue)

Định nghĩa: Hệ màu RGB là một mô hình cộng màu (additive color model) dựa trên ba màu cơ bản: Đỏ (Red), Xanh lá (Green) và Xanh dương (Blue). Đây là hệ màu phổ biến nhất được sử dụng trên các thiết bị kỹ thuật số như màn hình, máy tính, TV, và các thiết bị phát sáng khác.

Trong mô hình này, mỗi màu sắc được tạo ra bằng cách kết hợp ánh sáng từ ba màu cơ bản với các cường độ khác nhau. Khi kết hợp ba màu này với giá trị tối đa, ta sẽ thu được màu trắng; ngược lại, khi cả ba màu đều có giá trị tối thiểu (0), ta sẽ có màu đen.

Biểu diễn trên máy tính:

- Mỗi thành phần màu (Red, Green, Blue) được biểu diễn bằng một số nguyên từ 0 đến 255, trong đó:
  - + 0 biểu thị không có ánh sáng.
  - + 255 biểu thị cường độ ánh sáng lớn nhất của thành phần đó.
- Cách biểu diễn này thường sử dụng hệ cơ số 16 (hexadecimal) để lưu trữ giá trị màu sắc. Ví dụ:
  - + Màu đỏ: (255, 0, 0) hoặc #FF0000.
  - + Màu xanh lá: (0, 255, 0) hoặc #00FF00.
  - + Màu xanh dương: (0, 0, 255) hoặc #0000FF.

+ Màu trắng: (255, 255, 255) hoặc #FFFFFF.

+ Màu đen: (0, 0, 0) hoặc #000000.

Dữ liệu RGB được lưu trữ trong máy tính dưới dạng ma trận ba chiều ( $\text{height} \times \text{width} \times 3$ ), trong đó mỗi pixel được biểu diễn bằng ba giá trị R, G, B. Điều này cho phép mỗi pixel mang thông tin về cường độ của ba màu cơ bản.

Ứng dụng:

- Hiển thị trên màn hình: Hệ màu RGB là tiêu chuẩn cho màn hình LED, LCD và TV, vì chúng hoạt động dựa trên nguyên lý phát sáng của ba màu cơ bản.
- Xử lý ảnh kỹ thuật số: Nhiều phần mềm chỉnh sửa ảnh như Photoshop, GIMP hoặc các thư viện lập trình như OpenCV đều sử dụng RGB để biểu diễn và chỉnh sửa ảnh.
- Đồ họa máy tính: RGB là nền tảng cho các hệ thống đồ họa và các trò chơi điện tử.

#### 1.1.2.2 Hệ màu HSL (Hue, Saturation, Lightness)

Định nghĩa: Hệ màu HSL là một mô hình màu sắc tập trung vào cách con người cảm nhận màu tự nhiên. HSL bao gồm ba thành phần chính:

1. Hue (Sắc độ): Đại diện cho màu sắc chính của đối tượng, chẳng hạn như đỏ, cam, vàng, xanh lá, hoặc xanh dương. Hue được biểu diễn dưới dạng góc trong vòng tròn màu, từ  $0^\circ$  đến  $360^\circ$ , với:
  - $0^\circ$ : Đỏ
  - $120^\circ$ : Xanh lá
  - $240^\circ$ : Xanh dương
2. Saturation (Độ bão hòa): Thể hiện mức độ tinh khiết hoặc đậm nhạt của màu sắc, với giá trị từ 0% (xám hoàn toàn, không có màu) đến 100% (màu đậm nhất).

3. Lightness (Độ sáng): Thể hiện độ sáng hoặc tối của màu sắc, từ 0% (đen hoàn toàn) đến 100% (trắng hoàn toàn).

Biểu diễn trên máy tính:

- Hue được lưu trữ dưới dạng góc, thường là số thực từ 0 đến 360.
- Saturation và Lightness được biểu diễn dưới dạng tỷ lệ phần trăm hoặc giá trị số thực từ 0 đến 1.
- Một màu sắc trong HSL có thể được biểu diễn như sau:
  - Màu đỏ đậm:  $H=0^\circ$ ,  $S=100\%$ ,  $L=50\%$ .
  - Màu xanh lá nhạt:  $H=120^\circ$ ,  $S=50\%$ ,  $L=75\%$ .

Ứng dụng:

- Chỉnh sửa ảnh: HSL được sử dụng trong các công cụ chỉnh sửa ảnh như Photoshop để điều chỉnh màu sắc một cách trực quan hơn.
- Đồ họa kỹ thuật số: Dễ dàng điều chỉnh sắc thái và độ sáng mà không làm thay đổi hoàn toàn màu sắc gốc.

### 1.1.2.3. Hệ màu HSV (Hue, Saturation, Value)

Định nghĩa: HSV, giống như HSL, cũng dựa trên cách con người cảm nhận màu sắc. Tuy nhiên, HSV sử dụng Value (Giá trị) thay vì Lightness để biểu diễn độ sáng.

Thành phần:

1. Hue (Sắc độ): Tương tự HSL, biểu diễn màu chính dưới dạng góc từ  $0^\circ$  đến  $360^\circ$ .
2. Saturation (Độ bão hòa): Biểu diễn mức độ tinh khiết của màu sắc, từ 0% (xám) đến 100% (màu sắc đậm nhất).
3. Value (Giá trị): Thể hiện độ sáng của màu, từ 0% (đen hoàn toàn) đến 100% (sáng nhất).

Biểu diễn trên máy tính: HSV thường được biểu diễn bằng ba giá trị số thực hoặc phần trăm.

Ứng dụng:

- Xử lý hình ảnh: HSV được sử dụng nhiều trong các thuật toán phân đoạn màu sắc và nhận diện màu.
- Thực tế tăng cường (AR): HSV giúp dễ dàng phát hiện các đối tượng màu trong môi trường thực.

#### **1.1.2.4 Hệ màu Grayscale (Thang xám)**

Định nghĩa: Grayscale là hệ màu đơn giản nhất, chỉ sử dụng một giá trị để biểu diễn độ sáng của pixel. Không giống các hệ màu khác, Grayscale không chứa thông tin về sắc độ hoặc độ bão hòa.

Biểu diễn trên máy tính:

- Mỗi pixel được biểu diễn bằng một giá trị từ 0 (đen hoàn toàn) đến 255 (trắng hoàn toàn).
- Một hình ảnh grayscale chỉ cần ma trận hai chiều để lưu trữ dữ liệu.

Ứng dụng:

- Xử lý ảnh: Được sử dụng trong nhận diện biên cạnh, phát hiện đối tượng và các ứng dụng yêu cầu hiệu quả cao.
- In ấn: Grayscale thường được dùng cho ảnh trắng đen để tiết kiệm mực in.

## **1.2 Tổng quan về học máy**

### **1.2.1. Học có giám sát**

#### **1.2.1.1 K-NN (K-Nearest Neighbors)**

Khái niệm: K-Nearest Neighbors (K-NN) là một thuật toán học máy có giám sát dùng để giải quyết các bài toán phân loại (classification) và hồi quy (regression). KNN

hoạt động dựa trên nguyên tắc tìm kiếm các điểm dữ liệu gần nhất trong không gian đặc trưng để dự đoán nhãn hoặc giá trị cho điểm dữ liệu mới.

Ý tưởng thuật toán:

- Tìm kiếm láng giềng gần nhất:
  - + Mỗi điểm trong không gian dữ liệu được xem như một điểm trong hệ tọa độ  $n$  - chiều, với  $n$  là số đặc trưng (features).
  - + Khi cần phân loại hoặc dự đoán giá trị của một điểm mới, thuật toán sẽ tìm  $K$  điểm có khoảng cách nhỏ nhất so với điểm mới.
- Đưa ra dự đoán:
  - + Phân loại: Nhãn của điểm mới sẽ là nhãn xuất hiện nhiều nhất trong  $K$  láng giềng gần nhất (đa số phiếu).
  - + Hồi quy: Giá trị của điểm mới sẽ là giá trị trung bình (hoặc trung vị) của  $K$  láng giềng.

Giải thuật:

- Chuẩn bị dữ liệu:
  - + Tập dữ liệu huấn luyện: Bao gồm các điểm dữ liệu đã được gán nhãn.
  - + Điểm dữ liệu mới cần phân loại hoặc dự đoán giá trị.
- Tính toán khoảng cách:
  - + Tính khoảng cách giữa điểm mới và từng điểm trong tập huấn luyện.
  - + Các hàm tính khoảng cách hình học (phù hợp với các thuộc tính đầu vào là kiểu số thực): Minkowski, Manhattan, Euclid, Chebyshev.
  - + Hàm khoảng cách Hamming (phù hợp với các thuộc tính đầu vào là kiểu nhị phân).
- Xác định  $K$  điểm gần nhất:
  - + Sắp xếp các điểm trong tập huấn luyện theo thứ tự khoảng cách tăng dần so với điểm mới.
  - + Lấy  $K$  điểm đầu tiên trong danh sách này.

- Dự đoán nhãn hoặc giá trị:
  - + Phân loại: Nếu số lượng lớn nhất trong K láng giềng thuộc nhãn A, điểm mới sẽ có nhãn A.
  - + Hồi quy: Tính trung bình giá trị của K láng giềng.

#### Các ưu điểm:

- Chi phí thấp cho quá trình huấn luyện (chỉ việc lưu lại các ví dụ học).
- Hoạt động tốt với các bài toán phân loại gồm nhiều lớp.
- Về mặt lý thuyết thì k-NN có thể đạt khả năng phán đoán tối ưu khi gặp một số điều kiện.
- Rất Linh động trong việc chọn hàm khoảng cách.

#### Nhược điểm:

- Phải lựa chọn hàm tính khoảng cách (sự khác biệt) thích hợp với bài toán.
- Chi phí tính toán (thời gian, bộ nhớ) cao tại thời điểm phân loại/dự đoán.

#### Ứng dụng:

- Phân loại văn bản:
  - + Phân loại email: Phân biệt email spam và không spam.
  - + Phân loại tin tức: Phân loại các bài viết theo chủ đề (chính trị, kinh tế, thể thao...).
  - + Phân loại ngôn ngữ: Nhận diện ngôn ngữ của văn bản.
- Nhận diện hình ảnh:
  - + Nhận diện chữ viết tay: Ví dụ, phân loại chữ số viết tay (như trong bài toán MNIST).
  - + Phân loại đối tượng: Phân biệt các loại vật thể trong ảnh (xe, người, động vật...).

- + Nhận diện khuôn mặt: Xác định danh tính của một người dựa trên khuôn mặt.
- Chẩn đoán y khoa:
  - + Phân loại bệnh: Xác định bệnh dựa trên các chỉ số sinh học (nhịp tim, huyết áp, đường huyết...).
  - + Phân loại tế bào ung thư: Ví dụ, phân loại các tế bào là lành tính hay ác tính.
  - + Phát hiện bệnh tim mạch: Sử dụng KNN để dự đoán nguy cơ mắc bệnh dựa trên hồ sơ sức khỏe.
- Dự đoán thời tiết:
  - + Dự đoán nhiệt độ: Ước lượng nhiệt độ tại một địa điểm dựa trên dữ liệu từ các trạm gần đó.
  - + Dự đoán mưa hay nắng: Phân loại thời tiết dựa trên các chỉ số khí hậu (độ ẩm, áp suất, tốc độ gió...).

### 1.2.1.2 SVM (Support Vector Machine)

Khái niệm: Thuật toán SVM (Support Vector Machine) là một thuật toán học máy có giám sát (supervised learning), được sử dụng chủ yếu cho phân loại và hồi quy. Tuy nhiên, SVM thường được biết đến nhiều hơn trong các bài toán phân loại.

Ý tưởng thuật toán:

- SVM tìm một siêu phẳng (hyperplane) tốt nhất để phân tách các lớp trong không gian dữ liệu.
- Siêu phẳng này được chọn sao cho khoảng cách (margin) giữa nó và các điểm dữ liệu gần nhất của mỗi lớp là lớn nhất có thể.
- Các điểm dữ liệu gần siêu phẳng nhất được gọi là vector hỗ trợ (support vectors). Đây là các điểm dữ liệu quan trọng nhất, quyết định vị trí của siêu phẳng.

Giải thuật:

- Huấn luyện:
  - + Tìm siêu phẳng tối ưu bằng cách giải bài toán tối ưu hóa (tối đa hóa margin hoặc sử dụng kernel trick).
- Tính toán:
  - + Xác định các vector hỗ trợ (support vectors).
  - + Tính các tham số  $w$ ,  $b$  dựa trên vector hỗ trợ.
- Dự đoán:
  - + Áp dụng siêu phẳng đã tìm được để dự đoán nhãn của dữ liệu mới.

Các ưu điểm:

- Hoạt động tốt với dữ liệu có số chiều cao.
- Tổng quát hóa tốt nhờ tối đa hóa margin.
- Linh hoạt trong việc xử lý dữ liệu phi tuyến nhờ kernel trick.

Nhược điểm:

- Tốn thời gian tính toán với dữ liệu lớn.
- Khó chọn kernel và tinh chỉnh tham số  $C$ ,  $\gamma$ .
- Nhạy cảm với dữ liệu nhiễu và không cân bằng lớp.

Ứng dụng:

- Phân loại văn bản, hình ảnh.
- Nhận diện giọng nói, khuôn mặt.
- Dự đoán y khoa (phân loại tế bào ung thư).
- Phân tích tài chính (dự đoán giá cổ phiếu).

### 1.2.1.3 Decision trees

Khái niệm: Decision trees là một thuật toán học máy dựa trên mô hình cây để giải quyết các bài toán phân loại và hồi quy. Mỗi nút trong cây đại diện cho một điều kiện hoặc



câu hỏi dựa trên các đặc trưng của dữ liệu, và các nhánh dẫn đến các kết quả phân loại hoặc dự đoán cụ thể.

Ý tưởng thuật toán:

- Chia nhỏ tập dữ liệu thành các nhóm con dựa trên các thuộc tính.
- Lặp lại việc chia này theo cách tối ưu nhất cho đến khi đạt được một điều kiện dừng (chẳng hạn, cây đạt đến chiều cao tối đa hoặc nhóm con không thể chia nhỏ hơn nữa).

Cấu trúc của cây:

- Nút gốc (Root Node):
  - + Là nút bắt đầu của cây, nơi áp dụng điều kiện đầu tiên để chia dữ liệu.
- Nút bên trong (Internal Nodes):
  - + Đại diện cho các điều kiện tiếp theo được áp dụng trên dữ liệu.
- Nút lá (Leaf Nodes):
  - + Đại diện cho các kết quả cuối cùng (nhãn phân loại hoặc giá trị dự đoán).
- Nhánh (Branch):
  - + Đại diện cho các kết quả có thể xảy ra của một điều kiện tại một nút.

Quy trình xây dựng cây quyết định

- Chọn đặc trưng tốt nhất để chia dữ liệu:
- Đặc trưng nào giúp phân chia dữ liệu tốt nhất sẽ được chọn.
- Tiêu chí để đánh giá:
  - + Gini Impurity: Đo độ thuần nhất của các nhóm con.
  - + Entropy (Thông tin gain): Đo sự giảm entropy khi chia dữ liệu.
  - + Mean Squared Error (MSE): Áp dụng trong bài toán hồi quy.
  - + Chia dữ liệu:
- Sử dụng đặc trưng đã chọn để chia tập dữ liệu thành các nhóm con.
- Lặp lại quá trình:

- + Áp dụng tương tự cho các nhóm con đến khi đạt điều kiện dừng.
- Điều kiện dừng:
  - + Tất cả các điểm dữ liệu trong nhóm con thuộc cùng một lớp.
  - + Số lượng điểm dữ liệu trong nhóm con nhỏ hơn ngưỡng cho phép.
  - + Cây đạt chiều cao tối đa quy định.

Các ưu điểm:

- Dễ hiểu và trực quan.
- Không cần chuẩn hóa dữ liệu:
  - + Không yêu cầu chuẩn hóa hoặc scaling dữ liệu đầu vào.
- Có thể xử lý dữ liệu phân loại và liên tục:
  - + Đặc trưng có thể là số hoặc danh mục.
- Nhanh chóng và hiệu quả:
  - + Quá trình huấn luyện và dự đoán tương đối nhanh.

Nhược điểm:

- Nếu cây quá phức tạp (sâu), nó sẽ phù hợp quá mức với dữ liệu huấn luyện và kém hiệu quả trên dữ liệu kiểm tra.
- Dữ liệu có nhiễu sẽ làm giảm độ chính xác.
- Khả năng dự đoán kém trong bài toán hồi quy.

Ứng dụng:

- Chẩn đoán bệnh dựa trên các đặc điểm sức khỏe.
- Phân loại rủi ro tín dụng.
- Gợi ý sản phẩm hoặc dịch vụ cho khách hàng.
- Phân loại tài liệu, phân tích ngữ điệu (sentiment analysis).
- Dự đoán các hành vi gian lận hoặc bất thường.

### 1.2.2. Học không giám sát

### 1.2.2.1. K-means (K-Means Clustering)

Khái niệm: K-Means là một thuật toán phân cụm (clustering) trong học không giám sát, được sử dụng để chia tập dữ liệu thành K cụm. Mỗi cụm được đại diện bởi một tâm (centroid), và các điểm dữ liệu trong cụm có đặc điểm gần giống nhau hơn so với các cụm khác.

Ý tưởng thuật toán:

- Nhóm dữ liệu thành K cụm sao cho các điểm trong cùng một cụm có khoảng cách đến tâm cụm là nhỏ nhất.
- Tâm cụm được tính là giá trị trung bình của các điểm dữ liệu thuộc cụm đó.
- Quá trình lặp lại đến khi các tâm cụm không còn thay đổi hoặc đạt điều kiện hội tụ.

Giải thuật:

- Khởi tạo:
  - + Chọn K giá trị ban đầu làm tâm cụm (có thể chọn ngẫu nhiên hoặc dùng phương pháp K-Means++).
- Phân cụm:
  - + Tính khoảng cách từ mỗi điểm dữ liệu đến tất cả các tâm cụm.
  - + Gán điểm đó vào cụm có tâm gần nhất.
- Cập nhật tâm cụm:
  - + Tính lại tâm cụm bằng trung bình cộng của tất cả các điểm trong cụm
- Lặp lại:
  - + Quay lại bước 2 và 3 cho đến khi:
  - + Các tâm cụm không còn thay đổi.
  - + Hoặc đạt số vòng lặp tối đa.

Kết quả:

- Các cụm dữ liệu và tâm cụm cuối cùng.

Ưu điểm:

- Dễ hiểu và triển khai: Thuật toán đơn giản và dễ thực hiện với các thư viện hỗ trợ.

- Nhanh: Hoạt động hiệu quả với dữ liệu kích thước nhỏ đến vừa.
- Khả năng mở rộng: Phù hợp với dữ liệu lớn khi dùng biến thể như Mini-Batch K-Means.

Nhược điểm:

- Phụ thuộc vào số cụm K: Yêu cầu xác định trước K, khó khăn nếu không rõ số cụm phù hợp.
- Nhạy cảm với giá trị khởi tạo: Lựa chọn tâm cụm ban đầu có thể ảnh hưởng lớn đến kết quả (giải pháp: dùng K-Means++).
- Không hoạt động tốt với cụm không hình cầu: Dữ liệu phân bố phức tạp hoặc có hình dạng không đồng nhất sẽ làm giảm hiệu quả.
- Dễ bị ảnh hưởng bởi ngoại lệ: Các điểm dữ liệu xa tâm cụm có thể làm sai lệch kết quả.

### 1.2.3 Học bán giám sát

#### 1.2.3.1 ANN (Artificial Neural Networks)

Khái niệm: ANN là mô hình tính toán được lấy cảm hứng từ cấu trúc và chức năng của mạng nơ-ron sinh học trong não người. Nó gồm nhiều lớp nơ-ron kết nối với nhau để xử lý và phân tích dữ liệu.

Cấu trúc cơ bản:

- Input Layer: Tiếp nhận dữ liệu đầu vào, mỗi nơ-ron trong lớp này tương ứng với một đặc trưng của dữ liệu.
- Hidden Layers: Lớp trung gian thực hiện các phép tính và trích xuất đặc trưng. Số lượng lớp và nơ-ron trong mỗi lớp phụ thuộc vào bài toán cụ thể.
- Output Layer: Trả về kết quả của mô hình, ví dụ như xác suất thuộc về các lớp (trong phân loại) hoặc giá trị số (trong hồi quy).

Hoạt động:

- Dữ liệu đầu vào được nhân với trọng số, sau đó cộng thêm bias.

- Kết quả được đưa qua hàm kích hoạt để tạo tín hiệu đầu ra.
- Quy trình này lặp lại ở mỗi lớp, từ lớp đầu vào đến lớp đầu ra.

Hàm kích hoạt:

- Sigmoid, ReLU, Tanh, Softmax.

Đào tạo mô hình:

- Lan truyền tiến (Forward Propagation): Dữ liệu đi qua các lớp để tạo ra đầu ra.
- Lan truyền ngược (Backpropagation): Điều chỉnh trọng số dựa trên sai số giữa đầu ra thực tế và đầu ra dự đoán.
- Hàm mất mát: MSE (Mean Squared Error) cho hồi quy, Cross-Entropy cho phân loại.

Ứng dụng:

- Nhận dạng ký tự viết tay.
- Dự báo giá chứng khoán.
- Phân tích cảm xúc từ văn bản.

### **1.2.3.2 CNN (Convolutional Neural Networks)**

Khái niệm: Thuật toán bán giám sát trong mạng nơ-ron tích chập (CNN) là một phương pháp học máy sử dụng cả dữ liệu có nhãn (labeled data) và dữ liệu không có nhãn (unlabeled data) để huấn luyện mô hình. Trong trường hợp này, mô hình CNN được áp dụng để phân tích dữ liệu hình ảnh (hoặc dữ liệu không gian) và học từ cả hai loại dữ liệu, giúp cải thiện hiệu quả học so với chỉ sử dụng dữ liệu có nhãn.

Ý tưởng thuật toán:

- Trong học bán giám sát, mô hình không chỉ dựa vào dữ liệu có nhãn mà còn tận dụng thông tin từ dữ liệu không có nhãn, từ đó cải thiện khả năng tổng quát của mô hình khi chỉ có một lượng dữ liệu có nhãn hạn chế.
- Cách thức hoạt động của CNN trong học bán giám sát:
  - + Dữ liệu có nhãn: Các mẫu dữ liệu này đi kèm với nhãn và được sử dụng để huấn luyện mô hình như trong học giám sát truyền thống.
  - + Dữ liệu không có nhãn: Các mẫu dữ liệu không có nhãn sẽ được sử dụng để hỗ trợ mô hình học, giúp mô hình khai thác các đặc trưng ẩn trong dữ liệu, mặc dù không có nhãn rõ ràng.
- Các phương pháp học bán giám sát trong CNN thường áp dụng các kỹ thuật bổ sung để tận dụng dữ liệu không có nhãn, ví dụ như semi-supervised loss functions hoặc các kỹ thuật học sâu khác như pseudo-labeling, consistency regularization, và self-training.

Các kỹ thuật phổ biến trong học bán giám sát với CNN:

- Pseudo-Labeling:
  - + Pseudo-labeling là một trong những kỹ thuật phổ biến trong học bán giám sát, đặc biệt trong các mạng nơ-ron CNN.
  - + Quy trình:
    - + Đầu tiên, huấn luyện mô hình CNN với một phần dữ liệu có nhãn.
    - + Sau đó, sử dụng mô hình đã huấn luyện để dự đoán nhãn cho dữ liệu không có nhãn.
    - + Những dự đoán này (pseudo-labels) được sử dụng như là nhãn giả cho dữ liệu không có nhãn trong các lần huấn luyện tiếp theo.
    - + Mô hình được huấn luyện lại với cả dữ liệu có nhãn và dữ liệu không có nhãn đã gán nhãn giả.
- Ưu điểm: Pseudo-labeling tận dụng được nhiều dữ liệu không có nhãn mà không cần phải gán nhãn thủ công cho toàn bộ dữ liệu.
- Consistency Regularization:

- + Consistency regularization là một phương pháp học bán giám sát nơi mô hình được huấn luyện sao cho các dự đoán của nó không thay đổi nhiều khi dữ liệu đầu vào bị thay đổi nhẹ.
- + Mục tiêu là giữ cho mô hình ổn định khi dữ liệu không có nhãn thay đổi một chút (ví dụ: khi áp dụng các phép biến đổi hình ảnh như xoay, phóng to, cắt ảnh, hay làm mờ ảnh).
- + Một ví dụ về phương pháp này là Virtual Adversarial Training (VAT), trong đó, mô hình học sao cho các dự đoán với dữ liệu có nhãn và dữ liệu không có nhãn phải tương đồng với nhau.
- + Việc này giúp mô hình khai thác được cấu trúc ẩn trong dữ liệu không có nhãn.
- Self-training:
  - + Self-training là một phương pháp học bán giám sát trong đó mô hình tự học từ chính dự đoán của mình.
- Quy trình:
  - + Mô hình CNN ban đầu được huấn luyện với dữ liệu có nhãn.
  - + Sau đó, sử dụng mô hình này để gán nhãn cho dữ liệu không có nhãn.
  - + Những dữ liệu có nhãn giả này sau đó được đưa vào quá trình huấn luyện tiếp theo, làm tăng số lượng dữ liệu có nhãn mà mô hình có thể học.
  - + Self-training có thể được thực hiện theo nhiều vòng, mỗi vòng sẽ sử dụng dữ liệu không có nhãn đã được gán nhãn giả để tiếp tục huấn luyện.
- Generative Models:
  - + Các mô hình sinh như Generative Adversarial Networks (GANs) và Variational Autoencoders (VAEs) có thể được sử dụng trong học bán giám sát để tạo ra các nhãn giả cho dữ liệu không có nhãn.
  - + Các mô hình này học cách tái tạo các đặc trưng trong dữ liệu và có thể tạo ra dữ liệu mới (cả nhãn và không có nhãn) dựa trên phân phối đã học từ dữ liệu có nhãn.
- Ladder Networks:

- + Ladder Networks là một kiến trúc mạng sâu với một cấu trúc đặc biệt giúp cải thiện học bán giám sát.
- + Mạng này có hai phần: một phần học với dữ liệu có nhãn và phần học với dữ liệu không có nhãn.
- + Mô hình học từ các tín hiệu được tái tạo lại từ một tầng ẩn của mạng, giúp cải thiện quá trình học mà không cần nhiều dữ liệu có nhãn.

#### Các ưu điểm:

- Tận dụng tối đa dữ liệu không có nhãn: CNN có thể học từ dữ liệu không có nhãn, giúp tiết kiệm thời gian và chi phí gán nhãn dữ liệu.
- Cải thiện hiệu suất mô hình: Việc sử dụng cả dữ liệu có nhãn và không có nhãn giúp mô hình học tốt hơn, đặc biệt khi dữ liệu có nhãn hạn chế.
- Tăng tính tổng quát của mô hình: Bằng cách học từ một lượng dữ liệu lớn hơn (bao gồm cả dữ liệu không có nhãn), mô hình có thể tổng quát tốt hơn khi gặp dữ liệu mới.

#### Các nhược điểm:

- Dự đoán không chính xác (Pseudo-labeling): Các nhãn giả có thể không chính xác, dẫn đến mô hình học không chính xác.
- Khó khăn trong việc lựa chọn phương pháp phù hợp: Việc chọn kỹ thuật bán giám sát phù hợp với từng bài toán có thể là một thử thách.
- Yêu cầu tính toán cao: Các phương pháp như pseudo-labeling hay training với dữ liệu không có nhãn đòi hỏi tính toán tốn kém, đặc biệt là khi dữ liệu không có nhãn rất lớn.

#### Ứng dụng:

- Nhận diện đối tượng trong hình ảnh: Sử dụng dữ liệu không có nhãn để nhận diện các đối tượng trong ảnh mà không cần phải gán nhãn cho toàn bộ tập ảnh.



- Phân loại ảnh y tế: Trong y học, có thể sử dụng học bán giám sát để phân loại các bức ảnh y tế (như X-quang hoặc MRI) khi chỉ có một số lượng nhỏ hình ảnh đã được gán nhãn.
- Nhận diện văn bản: Phân loại và nhận diện các ký tự trong ảnh, đặc biệt khi số lượng văn bản có nhãn là hạn chế.
- Xử lý dữ liệu không gian: Các bài toán liên quan đến nhận dạng đối tượng trong không gian 3D cũng có thể sử dụng học bán giám sát với CNN.

### 1.2.3.3 R-CNN (Region-based Convolutional Neural Networks)

Khái niệm: R-CNN là một thuật toán học sâu (deep learning) được thiết kế để giải quyết bài toán phát hiện đối tượng (object detection) trong hình ảnh. Thuật toán này kết hợp mạng nơ-ron tích chập (CNN) với phương pháp tìm kiếm vùng (region proposals) để xác định và phân loại các đối tượng trong ảnh. R-CNN là một trong những bước tiến quan trọng trong việc phát triển các mô hình phát hiện đối tượng, trước khi xuất hiện các cải tiến như Fast R-CNN và Faster R-CNN.

Ý tưởng thuật toán:

- Tìm kiếm vùng (Region Proposal): Trước tiên, R-CNN sử dụng một phương pháp gọi là Selective Search để tạo ra các vùng (region proposals) tiềm năng trong ảnh. Những vùng này có thể chứa đối tượng và là ứng viên cho việc phân loại.
- Trích xuất đặc trưng (Feature Extraction): Sau khi tìm được các vùng tiềm năng, R-CNN sử dụng một mạng nơ-ron tích chập (CNN) đã được huấn luyện trước (pre-trained CNN, ví dụ: AlexNet) để trích xuất các đặc trưng từ mỗi vùng. Điều này cho phép R-CNN chuyển đổi từng vùng vào một vector đặc trưng mà sau đó có thể sử dụng để phân loại.
- Phân loại vùng (Region Classification): Các đặc trưng trích xuất từ CNN sẽ được đưa qua một bộ phân loại (thường là SVM - Support Vector Machine)

để phân loại xem mỗi vùng chứa đối tượng gì, hoặc có phải là nền (background) hay không.

- Hồi quy đối tượng (Bounding Box Regression): Cuối cùng, R-CNN sử dụng một mô hình hồi quy để điều chỉnh (fine-tune) các hộp bao quanh (bounding boxes) sao cho phù hợp hơn với đối tượng thực tế trong ảnh.

Quy trình hoạt động:

- Chia ảnh thành các vùng tiềm năng: Sử dụng Selective Search để tìm các vùng (region proposals) có thể chứa đối tượng.
- Trích xuất đặc trưng từ mỗi vùng: Mỗi vùng được đưa vào một CNN để trích xuất đặc trưng. R-CNN thường sử dụng các mô hình CNN đã huấn luyện trước, như AlexNet hoặc VGG.
- Phân loại với SVM: Mỗi đặc trưng được đưa qua một bộ phân loại SVM để xác định đối tượng trong vùng đó hoặc là nền.
- Hồi quy bounding box: R-CNN sử dụng một mô hình hồi quy để tinh chỉnh các hộp bao quanh, giúp các vùng phát hiện được chính xác hơn.

Ưu điểm:

- Kết hợp CNN với phương pháp tìm kiếm vùng: R-CNN đã cải thiện đáng kể hiệu suất phát hiện đối tượng so với các phương pháp phát hiện đối tượng truyền thống.
- Được huấn luyện với dữ liệu thực tế: R-CNN cho phép huấn luyện trực tiếp với dữ liệu thực tế và có khả năng phát hiện nhiều loại đối tượng trong ảnh.
- Khả năng phát hiện đối tượng đa dạng: R-CNN có thể phát hiện nhiều loại đối tượng khác nhau trong ảnh nhờ vào việc phân loại riêng biệt cho mỗi vùng.

Nhược điểm:

- Tính toán chậm: Quy trình R-CNN rất chậm, vì mỗi vùng (region proposal) phải được đưa qua một mạng CNN riêng biệt, gây tốn thời gian tính toán rất lớn, đặc biệt khi có nhiều vùng tiềm năng.
- Bộ phân loại SVM không tối ưu: Sử dụng SVM cho phân loại từng vùng là một bước không tối ưu, vì phải huấn luyện một bộ phân loại riêng cho từng lớp và cho mỗi đối tượng.
- Dự đoán chậm trong quá trình inference: R-CNN rất chậm trong quá trình suy luận (inference), vì mỗi ảnh cần được xử lý qua hàng nghìn vùng (region proposals), mỗi vùng lại cần trích xuất đặc trưng từ CNN.
- Không hỗ trợ fine-tuning hiệu quả: Quá trình fine-tuning các mạng CNN và SVM không đồng bộ, dẫn đến sự kém hiệu quả khi áp dụng trên các bài toán thực tế.

#### Ứng dụng:

- Phát hiện đối tượng trong hình ảnh: R-CNN được sử dụng rộng rãi trong các bài toán phát hiện đối tượng như nhận diện khuôn mặt, nhận diện phương tiện giao thông, nhận diện các đối tượng trong ảnh y tế, v.v.
- Phân đoạn đối tượng (Instance Segmentation): R-CNN có thể mở rộng để phân đoạn đối tượng trong ảnh, giúp tìm và phân loại các đối tượng với độ chính xác cao hơn.
- Nhận diện các đối tượng trong video: R-CNN cũng có thể được áp dụng trong các bài toán nhận diện đối tượng trong video bằng cách phát hiện các đối tượng trong từng khung hình và theo dõi chúng qua các khung hình.

## CHƯƠNG 2. XÂY DỰNG HỆ THỐNG NHẬN DIỆN MÀU SẮC BẢNG PANDAS VÀ OPENCV

### 2.1. Thư viện

#### 2.1.1. Pandas

Pandas là một thư viện mạnh mẽ và linh hoạt trong Python, chủ yếu được sử dụng để xử lý, phân tích và thao tác dữ liệu. Pandas cung cấp các cấu trúc dữ liệu hiệu quả như DataFrame và Series, giúp dễ dàng làm việc với dữ liệu dạng bảng (như từ các tệp CSV, Excel, SQL). Đây là công cụ không thể thiếu cho các nhà phân tích dữ liệu, nhà khoa học dữ liệu, và các lập trình viên muốn làm việc với dữ liệu lớn.

Tính năng chính của Pandas:

- Cấu trúc dữ liệu mạnh mẽ:
  - + DataFrame: Đây là một bảng hai chiều trong Pandas, giống như một bảng trong SQL với hàng và cột. Mỗi cột có thể có kiểu dữ liệu khác nhau (số nguyên, số thực, chuỗi, v.v.). DataFrame cho phép lưu trữ nhiều loại dữ liệu trong cùng một cấu trúc, dễ dàng truy cập, thao tác và xử lý.
  - + Series: Đây là một mảng một chiều trong Pandas, thường được sử dụng để lưu trữ một cột của DataFrame hoặc một dãy dữ liệu đơn giản. Nó có khả năng truy vấn chỉ mục linh hoạt và dễ dàng thao tác các phần tử riêng lẻ.
- Xử lý dữ liệu dễ dàng:
  - + Đọc và ghi dữ liệu từ/đến nhiều định dạng: Pandas hỗ trợ các định dạng dữ liệu phổ biến như CSV, Excel, SQL, JSON, HTML, và HDF5. Dễ dàng đọc và ghi dữ liệu từ và vào các định dạng này với các phương thức `read_csv()`, `read_excel()`, `read_sql()`, `to_csv()`, `to_excel()`, `to_sql()`, và `to_json()`.

- + Xử lý dữ liệu thiếu: Pandas cung cấp nhiều cách để xử lý các giá trị thiếu như xóa, thay thế bằng giá trị mặc định hoặc sử dụng phương pháp `fillna()` để lấp đầy giá trị thiếu bằng giá trị trung bình hoặc giá trị phổ biến nhất của cột.
  - + Chỉ mục linh hoạt: Dễ dàng truy vấn và lọc dữ liệu với chỉ mục linh hoạt, sử dụng phương thức `loc` và `iloc` để chọn hàng, cột hoặc phần tử cụ thể trong `DataFrame` hoặc `Series`.
  - + Thao tác trên hàng và cột: Thực hiện các phép toán cộng, trừ, nhân, chia giữa các cột hoặc hàng, sử dụng các phép toán phần tử-wise, giúp làm việc với dữ liệu số liệu dễ dàng hơn.
- Thao tác dữ liệu nâng cao:
- + Lọc dữ liệu: Tạo các `DataFrame` mới chỉ chứa các hàng thỏa mãn điều kiện cụ thể, sử dụng phương thức `query()` để thực hiện truy vấn với biểu thức điều kiện.
  - + Sắp xếp dữ liệu: Sắp xếp các `DataFrame` theo cột hoặc hàng sử dụng phương thức `sort_values()` để sắp xếp theo giá trị, hoặc `sort_index()` để sắp xếp theo chỉ mục.
  - + Nhóm dữ liệu (Group by) và thực hiện các thao tác tổng hợp (Aggregation): Dễ dàng nhóm dữ liệu theo cột và thực hiện các phép tính thống kê như `sum`, `mean`, `count`, `min`, `max` với phương thức `groupby()`.
- Phân tích và trực quan hóa:
- + Tính toán thống kê cơ bản: Pandas cung cấp các phương thức như `mean()`, `std()`, `median()`, `min()`, `max()` để tính toán các thống kê cơ bản cho từng cột hoặc toàn bộ `DataFrame`.
  - + Trực quan hóa dữ liệu: Dễ dàng sử dụng các thư viện như `Matplotlib` hoặc `Seaborn` để vẽ biểu đồ cột, đường, `pie chart`, `histogram`, `scatter plot` để trực quan hóa dữ liệu từ `DataFrame`.

- + Xử lý dữ liệu chuỗi thời gian (Time Series): Pandas hỗ trợ các công cụ mạnh mẽ để làm việc với dữ liệu chuỗi thời gian, sử dụng DatetimeIndex để lọc, truy vấn, và phân tích dữ liệu theo thời gian. Phương thức resample() giúp thống kê theo các khoảng thời gian nhất định (ngày, tuần, tháng, quý) như tính tổng, trung bình, min, max.
- Ứng dụng của Pandas:
  - + Phân tích dữ liệu: Làm sạch, biến đổi và chuẩn bị dữ liệu trước khi đưa vào các mô hình học máy. Pandas hỗ trợ rất tốt cho việc xử lý dữ liệu, giúp giảm bớt thời gian và công sức khi cần phân tích dữ liệu lớn.
  - + Chuẩn bị dữ liệu: Làm sạch và chuẩn hóa dữ liệu từ các nguồn khác nhau, thực hiện loại bỏ các giá trị không hợp lệ, loại bỏ dữ liệu thiếu, chuẩn hóa giá trị (scaling, min-max normalization, z-score normalization).
  - + Tích hợp với học máy: Chuẩn bị dữ liệu cho các mô hình học máy với các thư viện như Scikit-learn, TensorFlow, PyTorch. Pandas giúp bạn dễ dàng trích xuất dữ liệu theo các dạng phù hợp với yêu cầu của các mô hình này.

### 2.1.2. Opencv

OpenCV (Open Source Computer Vision Library) là một thư viện mã nguồn mở mạnh mẽ trong xử lý hình ảnh và video, được sử dụng rộng rãi trong các ứng dụng xử lý ảnh, thị giác máy tính, và robot. Thư viện này cung cấp một loạt các công cụ mạnh mẽ và thuật toán hiệu quả để phát hiện đối tượng, nhận diện khuôn mặt, xử lý ảnh và video, hỗ trợ làm việc với dữ liệu hình ảnh từ các nguồn khác nhau.

Tính năng chính của OpenCV:

- Xử lý ảnh:

- + Đọc, ghi và hiển thị ảnh: OpenCV hỗ trợ đọc và ghi ảnh từ nhiều định dạng khác nhau như JPEG, PNG, BMP. Dễ dàng hiển thị ảnh với imshow() và waitKey() để chờ xử lý các sự kiện bàn phím.
  - + Chuyển đổi giữa các không gian màu: Hỗ trợ chuyển đổi giữa các không gian màu (RGB, Grayscale, HSV, YCrCb) với các hàm như cvtColor() để dễ dàng xử lý màu sắc trong ảnh.
  - + Làm mịn ảnh: Dùng các bộ lọc như Gaussian Blur, Median Blur để giảm nhiễu, làm mịn ảnh, giữ lại các chi tiết quan trọng, sử dụng blur(), GaussianBlur(), medianBlur().
- Xử lý video:
- + Đọc và ghi video: OpenCV cung cấp các phương thức để thao tác với video, như VideoCapture() để mở video từ file hoặc camera, VideoWriter() để ghi video ra file.
  - + Chuyển đổi khung hình: Dễ dàng chuyển đổi giữa các khung hình trong video, cắt hoặc ghép các đoạn video lại với nhau.
  - + Xử lý và phân tích video: Các chức năng như phát hiện chuyển động, theo dõi đối tượng, sử dụng các thuật toán như Optical Flow, KLT (Kanade-Lucas-Tomasi) để theo dõi các đối tượng chuyển động.
- Phát hiện đối tượng:
- + Nhận diện khuôn mặt: Sử dụng thuật toán Haar Cascades hoặc mạng nơ-ron DNN (Deep Neural Network) để nhận diện khuôn mặt trong ảnh và video. OpenCV cung cấp bộ dữ liệu pre-trained của các khuôn mặt phổ biến.
  - + Phát hiện đối tượng khác: Sử dụng Cascade Classifier để phát hiện các đối tượng khác nhau trong ảnh và video, ví dụ như xe cộ, biển báo giao thông. OpenCV cũng hỗ trợ các mô hình YOLO, SSD, Faster R-CNN từ thư viện DNN.

- Xử lý hình học:
  - + Phát hiện cạnh: Sử dụng thuật toán Canny Edge Detection để phát hiện các cạnh trong ảnh, giúp dễ dàng nhận dạng các đường thẳng, đường cong.
  - + Phép biến đổi hình học: OpenCV cung cấp các hàm như warpAffine(), warpPerspective() để thay đổi hình dạng và độ lệch của ảnh, giúp thay đổi góc nhìn, cắt ảnh, và thay đổi kích thước ảnh.
  - + Hough Transform: Phát hiện đường thẳng, đường cong, hình tròn trong ảnh sử dụng thuật toán Hough Transform với các phương thức HoughLines(), HoughCircles().
- Ứng dụng học máy và deep learning:
  - + Tích hợp với TensorFlow, PyTorch: OpenCV hỗ trợ các mô hình học sâu từ TensorFlow và PyTorch, cho phép bạn sử dụng các mạng nơ-ron đã được huấn luyện sẵn để nhận diện đối tượng, phân loại hình ảnh, và các tác vụ thị giác máy khác.
  - + Mô hình DNN (Deep Neural Network): Sử dụng module DNN trong OpenCV để chạy các mô hình học sâu như YOLO, SSD, Faster R-CNN để phát hiện đối tượng, nhận diện khuôn mặt trong ảnh và video.
- Trực quan hóa và hỗ trợ cộng đồng:
  - + Thư viện mẫu và công cụ trực quan hóa: OpenCV đi kèm với nhiều mẫu ứng dụng và công cụ giúp dễ dàng học tập, thử nghiệm và phát triển các thuật toán xử lý hình ảnh, video.
  - + Hỗ trợ đa nền tảng: OpenCV được hỗ trợ trên nhiều nền tảng như Windows, Linux, macOS và Android, iOS. Điều này giúp dễ dàng triển khai các ứng dụng thị giác máy trên nhiều thiết bị khác nhau.

Ứng dụng của Pandas và OpenCV:



- Ứng dụng của Pandas:
  - + Phân tích dữ liệu: Dùng để phân tích dữ liệu thống kê trong các báo cáo, nghiên cứu khoa học, thống kê kinh doanh.
  - + Xử lý dữ liệu trong học máy: Chuẩn bị dữ liệu từ các tập tin lớn, dọn dẹp dữ liệu, tạo các tập dữ liệu mẫu cho việc huấn luyện các mô hình học máy.
  - + Xử lý dữ liệu thời gian: Lọc, nhóm, phân tích và trực quan hóa dữ liệu thời gian để nhận dạng xu hướng và dự đoán.
- Ứng dụng của OpenCV:
  - + Xử lý ảnh và video: Phát hiện đối tượng, nhận diện khuôn mặt, xử lý hình ảnh thời gian thực, ghép và chuyển đổi giữa các định dạng video.
  - + Ứng dụng robot và thiết bị thông minh: Phát hiện và nhận diện đối tượng trong các ứng dụng robot, nhận dạng chuyển động, điều khiển robot tự động.
  - + Ứng dụng trong ngành công nghiệp và bảo mật: Xử lý và phân tích hình ảnh để phát hiện các bất thường, xác định hành vi người dùng, giám sát an ninh, theo dõi đối tượng di chuyển.

Cách sử dụng Pandas và OpenCV trong thực tế:

- Pandas: Sử dụng Pandas để đọc, viết, và thao tác dữ liệu trong các tập tin CSV, Excel, SQL. Sau đó, tiến hành xử lý dữ liệu như lọc, nhóm, thống kê, và trực quan hóa để có được những hiểu biết sâu hơn về dữ liệu.
- OpenCV: Dùng OpenCV để xử lý ảnh và video, phát hiện đối tượng, nhận diện khuôn mặt, phát hiện chuyển động, làm mịn ảnh và trực quan hóa kết quả xử lý. Sử dụng OpenCV kết hợp với Pandas để xử lý dữ liệu hình ảnh lớn, từ đó rút ra các số liệu thống kê, phân tích và dự đoán.

Thông qua việc kết hợp Pandas và OpenCV, các nhà khoa học dữ liệu, kỹ sư phần mềm và các lập trình viên có thể xử lý dữ liệu phức tạp, trực quan hóa dữ liệu hình ảnh và video, từ đó tạo ra các giải pháp mạnh mẽ trong lĩnh vực dữ liệu lớn, học máy, và thị giác máy.

### 2.1.3 Một số thư viện khác

`sklearn`

`sklearn.model_selection`:

Chức năng: Cung cấp công cụ để chia dữ liệu thành tập huấn luyện và kiểm tra.

Vai trò:

Sử dụng `train_test_split()` để chia dữ liệu thành các tập `X_train`, `X_test`, `y_train`, và `y_test` với tỷ lệ 80% huấn luyện, 20% kiểm tra.

`sklearn.neighbors`:

Chức năng: Cung cấp thuật toán K-Nearest Neighbors (KNN) cho phân loại và hồi quy.

Vai trò:

Sử dụng `KNeighborsClassifier` để tạo mô hình KNN với `n_neighbors=3`.

Huấn luyện mô hình bằng `fit()` và dự đoán nhãn trên tập kiểm tra bằng `predict()`.

`sklearn.metrics`:

Chức năng: Đánh giá hiệu năng của mô hình.

Vai trò:

Dùng `accuracy_score()` để tính toán độ chính xác (phần trăm nhãn dự đoán đúng so với nhãn thực tế).

`sklearn.svm`:

Cung cấp các thuật toán SVM (Support Vector Machine) cho phân loại, hồi quy và phân cụm.

Ví dụ: SVC (cho phân loại), SVR (cho hồi quy).

`sklearn.tree`:

Chức năng: Cung cấp thuật toán máy vector hỗ trợ (Support Vector Machine - SVM).

Vai trò:

SVC được sử dụng với kernel tuyến tính (`kernel='linear'`) để huấn luyện mô hình.

`probability=True` cho phép tính xác suất dự đoán, hữu ích nếu cần dự đoán phân phối xác suất.

`pickle`

Chức năng: Lưu và tải mô hình đã huấn luyện dưới dạng file nhị phân.

Vai trò:

Sử dụng `pickle.dump()` để lưu mô hình KNN và đối tượng `LabelEncoder` vào file `color_knn_model.pkl` để sử dụng sau này.

## **2.2. Xây dựng hệ thống nhận diện màu sắc trong ảnh**

### **2.2.1 Mô tả bài toán**

Bài toán nhận diện màu sắc là một vấn đề trong lĩnh vực xử lý hình ảnh và phân loại. Mục tiêu chính là xây dựng một hệ thống có khả năng nhận diện và dự đoán tên màu sắc dựa trên giá trị RGB (Red, Green, Blue) từ camera trực tiếp. Hệ thống sẽ sử dụng Pandas để xử lý dữ liệu màu sắc và OpenCV để xử lý hình ảnh từ camera, đồng thời tích hợp các thuật toán học máy như K-NN, SVM, Decision tree để dự đoán tên màu sắc chính xác.

Bài toán bao gồm các bước chính:

- Xử lý dữ liệu:
- Sử dụng Pandas để đọc file CSV và chuẩn hóa dữ liệu màu sắc.
- Chuẩn bị tập dữ liệu huấn luyện và kiểm tra bằng cách chia tỷ lệ phù hợp.
- Xử lý hình ảnh:
- Sử dụng OpenCV để trích xuất giá trị RGB từ camera.
- Dự đoán:
- Áp dụng thuật toán học máy (KNN, SVM, Decision Tree) để dự đoán tên màu sắc.
- Hiển thị kết quả:
- Tên màu sắc dự đoán và giá trị RGB hiển thị trực tiếp trên màn hình khi sử dụng camera.

### 2.2.1 Các bước thực hiện

#### Bước 1: Thu thập dữ liệu

- Nguồn dữ liệu: dữ liệu được thu thập từ nguồn dữ liệu mở trên github (link: <https://github.com/Smoltpob/XKCDColors-Dataset?tab=readme-ov-file>).
- File dữ liệu được lưu dưới dạng file CSV với cấu trúc các cột: 'r', 'g', 'b', 'colorname' tương ứng với các giá trị của màu red, green, blue và tên nhãn màu sắc.

#### Bước 2: Xử lý và phân tích dữ liệu

- Đầu tiên, dữ liệu màu sắc được thu thập từ một file CSV chứa các giá trị RGB (đỏ, xanh lá, xanh dương) và tên màu sắc tương ứng. Sau đó, sử dụng Pandas để đọc và kiểm tra dữ liệu, đảm bảo rằng không có giá trị thiếu và dữ liệu được chuẩn bị sẵn sàng cho quá trình huấn luyện. Các cột dữ liệu bao gồm r, g, b (đại diện cho các giá trị màu sắc) và colorname (tên màu sắc). Sau khi đọc dữ liệu, mã hóa các tên màu sắc thành các giá trị số bằng cách sử dụng LabelEncoder từ thư viện Scikit-learn để giúp mô hình dễ dàng xử lý.

- Tiếp theo, chia dữ liệu thành hai phần: tập huấn luyện và tập kiểm tra, với tỷ lệ chia là 80% cho huấn luyện và 20% cho kiểm tra, nhằm đánh giá hiệu quả của mô hình sau khi huấn luyện. Sau khi dữ liệu được chia, sử dụng ba thuật toán học máy phổ biến là K-Nearest Neighbors (KNN), Support Vector Machine (SVM) và Decision Tree để huấn luyện trên tập huấn luyện. Các mô hình này sẽ được huấn luyện độc lập và so sánh hiệu quả với nhau.
- Đánh giá mô hình, sử dụng các độ đo accuracy, precision, recall và F1-score để xem xét hiệu suất của từng thuật toán. Các kết quả đánh giá này giúp xác định mô hình hoạt động tốt nhất. Cuối cùng, mô hình tốt nhất sẽ được lưu lại và tích hợp vào ứng dụng thực tế để dự đoán tên màu sắc từ các giá trị RGB lấy từ camera.
- Khi hệ thống được triển khai, giá trị RGB từ các pixel trong hình ảnh được lấy, và mô hình sẽ dự đoán tên màu sắc tương ứng, hiển thị kết quả trực tiếp.

### Bước 3: Xây dựng mô hình

- Thuật toán sử dụng: K-Nearest Neighbors (KNN), Support Vector Machine (SVM) và Decision Tree. Dưới đây là lý do chọn và cách hoạt động của từng thuật toán.
- K-Nearest Neighbors (KNN):
- Lý do chọn: KNN là thuật toán đơn giản và dễ hiểu, sử dụng chiến lược dự đoán dựa trên các điểm gần nhất trong không gian đặc trưng. Thuật toán này không yêu cầu huấn luyện trực tiếp mà chỉ cần lưu trữ dữ liệu, giúp nó linh hoạt trong việc xử lý các bài toán với tập dữ liệu nhỏ đến trung bình.
- Cách hoạt động: KNN hoạt động bằng cách tính khoảng cách giữa điểm cần dự đoán và các điểm dữ liệu trong tập huấn luyện. Sau đó, mô hình sẽ chọn K điểm gần nhất và dựa trên tần suất xuất hiện của nhãn để đưa ra dự đoán.
- Support Vector Machine (SVM):

- Lý do chọn: SVM là thuật toán mạnh mẽ cho các bài toán phân loại và đặc biệt hiệu quả với các bài toán phân loại phi tuyến tính. Với bài toán nhận diện màu sắc, SVM có thể tìm ra các đường biên phân chia tối ưu giữa các lớp màu.
- Cách hoạt động: SVM tìm kiếm một siêu phẳng (hyperplane) có thể phân chia các lớp trong không gian đặc trưng sao cho khoảng cách giữa các điểm và siêu phẳng này là lớn nhất. Đặc biệt, SVM sử dụng kernel để xử lý các bài toán phi tuyến tính, chuyển các đặc trưng vào không gian có chiều cao hơn.
- Decision Tree:
- Lý do chọn: Decision Tree là một thuật toán phân loại dễ hiểu và có thể trực quan hóa. Nó xây dựng các quy tắc phân loại dựa trên các thuộc tính của dữ liệu. Trong trường hợp nhận diện màu sắc, Decision Tree có thể phân chia các màu sắc dựa trên các giá trị RGB để đưa ra dự đoán.
- Cách hoạt động: Decision Tree xây dựng một cây quyết định từ dữ liệu bằng cách chia nhỏ dữ liệu vào các nhánh, chọn các đặc trưng phân chia tối ưu ở mỗi nút của cây. Mỗi nhánh đại diện cho một phân loại hoặc dự đoán cuối cùng. Thuật toán này sử dụng các chỉ số như Gini Index hoặc Entropy để quyết định các đặc trưng phân chia tại mỗi bước.
- Huấn luyện từng mô hình:
- Phân chia dữ liệu: chia dữ liệu thành hai phần: tập huấn luyện và tập kiểm tra, sử dụng tỷ lệ 80-20 (80% cho huấn luyện, 20% cho kiểm tra).
- Huấn luyện trên tập huấn luyện: Mỗi mô hình sẽ được huấn luyện độc lập với tập huấn luyện.
- Lưu mô hình:
- Sau khi huấn luyện mô hình, lưu mô hình vào file để có thể tái sử dụng sau này mà không cần phải huấn luyện lại từ đầu.
- Mô hình đã lưu có thể được tải lại và sử dụng cho việc dự đoán trong tương lai mà không cần huấn luyện lại.

#### Bước 4: Đánh giá mô hình

- Accuracy: Accuracy là một trong những độ đo phổ biến nhất để đánh giá hiệu suất của mô hình trong học máy, đặc biệt là trong các bài toán phân loại. Nó cho biết tỷ lệ phần trăm các dự đoán đúng trên tổng số mẫu dữ liệu.
- Accuracy đo lường hiệu quả tổng quát của mô hình trong việc phân loại đúng các mẫu dữ liệu. Giá trị của Accuracy nằm trong khoảng  $[0,1]$  hoặc có thể biểu diễn dưới dạng phần trăm (0% đến 100%).
- Precision: Precision là một độ đo quan trọng trong các bài toán phân loại, đặc biệt là khi cần đánh giá mức độ chính xác của các dự đoán dương (positive). Nó cho biết trong tất cả các dự đoán là dương, bao nhiêu dự đoán là đúng.
- Precision đánh giá mức độ chính xác của các dự đoán dương: mô hình có dự đoán sai nhiều dương giả hay không. Precision cao nghĩa là khi mô hình dự đoán một mẫu là dương, thì khả năng cao nó thực sự là dương.
- Recall: Recall, còn được gọi là Sensitivity hoặc True Positive Rate (TPR), là một độ đo dùng để đánh giá hiệu suất của mô hình phân loại, đặc biệt là khả năng phát hiện các mẫu thuộc lớp dương (positive). Recall cho biết tỷ lệ phần trăm các mẫu thực sự dương mà mô hình dự đoán đúng.
- Recall cao có nghĩa là mô hình nhận diện tốt hầu hết các mẫu thực sự dương.
- Điều này đặc biệt quan trọng trong các bài toán mà việc bỏ sót các trường hợp dương có thể gây hậu quả nghiêm trọng.
- F-measure (F1-score): F1-Score là độ đo phổ biến kết hợp giữa độ chính xác (Precision) và độ phủ (Recall). Đối với bài toán phân cụm, F1-Score đánh giá mức độ một cụm khớp với nhãn thực tế (nếu có nhãn).
- F1-Score cao nghĩa là mô hình đạt được sự cân bằng tốt giữa Precision và Recall. Khi F1-Score gần 1, mô hình có khả năng dự đoán chính xác và đầy đủ các mẫu dương.
- F1-Score thấp nghĩa là có sự thiếu cân bằng giữa Precision và Recall. Ví dụ, nếu Precision cao nhưng Recall thấp (hoặc ngược lại), mô hình có thể không hiệu quả trong việc phát hiện đúng các mẫu dương mà không gây quá nhiều dương giả.

## CHƯƠNG 3. THỰC NGHIỆM

### 3.1 Dữ liệu

#### 3.1.1 Dữ liệu hình ảnh

Mô tả: Dữ liệu hình ảnh được ghi lại từ camera trực tiếp. Mỗi hình ảnh sẽ chứa các pixel có màu sắc cụ thể, từ đó có thể phân tích và nhận diện màu sắc trong hình ảnh.

Các kỹ thuật xử lý dữ liệu hình ảnh:

- Lấy dữ liệu hình ảnh từ camera: Sử dụng thư viện OpenCV để thu thập hình ảnh trực tiếp từ camera.
- Chuyển đổi hình ảnh thành các đặc trưng: Các hình ảnh được xử lý để trích xuất các đặc trưng màu sắc.
- Dự đoán màu sắc: Sau khi trích xuất đặc trưng màu sắc, mô hình học máy (được huấn luyện trước đó) sẽ phân loại hình ảnh và gán nhãn cho màu sắc tương ứng (ví dụ: đỏ, xanh, vàng).

#### 3.1.2 Dữ liệu từ file CSV

Mô tả: Dữ liệu từ file CSV là dạng lưu trữ thông tin về màu sắc của các hình ảnh dưới dạng các giá trị RGB cùng với nhãn màu sắc.

Các kỹ thuật xử lý dữ liệu từ file CSV:

- Lưu trữ thông tin trong file CSV: Dữ liệu hình ảnh được chuyển thành các giá trị RGB của các pixel trong ảnh và được lưu trữ trong file CSV. Mỗi dòng trong CSV sẽ đại diện cho một hình ảnh hoặc một mẫu màu sắc với các cột là các giá trị RGB và nhãn màu sắc.
- Đọc dữ liệu từ file CSV: Để huấn luyện mô hình học máy, sử dụng Pandas để đọc dữ liệu file CSV. Dữ liệu này sẽ được sử dụng làm đầu vào cho mô hình.
- Phân chia dữ liệu: Sau khi đọc dữ liệu, tách dữ liệu thành hai phần: đặc trưng màu sắc (RGB) và nhãn màu sắc.
- Huấn luyện mô hình: Mô hình học máy sẽ sử dụng dữ liệu CSV đã tách để học cách phân loại màu sắc dựa trên các giá trị RGB.



- Dự đoán và đánh giá: Sau khi huấn luyện xong, mô hình sẽ sử dụng dữ liệu kiểm tra để dự đoán màu sắc và đánh giá hiệu suất của mô hình sử dụng các độ đo accuracy, precision, recall và f1-score.

### 3.2 Độ đo đánh giá

Hệ thống nhận diện màu sắc đã được huấn luyện trên tập dữ liệu CSV với hơn 1 triệu giá trị màu RGB khác nhau. Kết quả huấn luyện của từng thuật toán như sau:

- Độ chính xác thuật toán KNN là 75.04%.
- Độ chính xác thuật toán SVM là 70.52%.
- Độ chính xác thuật toán Decision tree là 61.83%.

Đánh giá kết quả huấn luyện:

- Thuật toán KNN đạt độ chính xác cao nhất với 75.04%. Điều này cho thấy KNN có khả năng xử lý tốt nhất trong việc phân loại màu sắc.
- Thuật toán SVM đứng thứ hai với độ chính xác 70.52%.
- Thuật toán Decision Tree có độ chính xác thấp nhất ở mức 61.83%.

### 3.3 Kết quả thực nghiệm

#### 3.3.1. Sử dụng SVM

Thuật toán SVM đạt được độ chính xác tổng thể là 70.52%. Trong quá trình thử nghiệm, SVM thể hiện những đặc điểm sau:

Điểm mạnh:

- Khả năng phân loại tốt với các màu cơ bản và màu có độ tương phản cao.
- Ổn định trong điều kiện ánh sáng bình thường.
- Thời gian xử lý nhanh, phù hợp cho ứng dụng thời gian thực.

Hạn chế:

- Hiệu suất giảm đáng kể với các màu có sắc độ gần nhau.
- Độ nhạy cao với thay đổi điều kiện ánh sáng.
- Khó khăn trong việc phân biệt các sắc thái của cùng một màu cơ bản.

### **3.3.2. Sử dụng K-NN**

K-NN đạt hiệu suất cao nhất với độ chính xác 75.04%. Các kết quả thử nghiệm chi tiết cho thấy:

Hiệu suất theo điều kiện ánh sáng:

- Ánh sáng tự nhiên: Độ chính xác 78.2%.
- Ánh sáng nhân tạo: Độ chính xác 73.8%.
- Điều kiện ánh sáng yếu: Độ chính xác 69.5%.

### **3.3.3. Sử dụng Decision Tree**

Decision Tree đạt độ chính xác 61.83%, với các đặc điểm thực nghiệm như sau:

Phân tích hiệu suất:

- Thời gian xử lý nhanh nhất trong ba thuật toán.
- Độ chính xác cao với các màu có ranh giới rõ ràng.
- Hiệu suất thấp với các màu có độ chuyển sắc tinh tế.

## KẾT LUẬN

### **Kết quả đạt được:**

Hệ thống nhận diện màu sắc đã được xây dựng thành công với sự tích hợp của các công nghệ và thư viện hiện đại. Cụ thể, hệ thống đã đạt được những thành tựu quan trọng trong việc kết hợp OpenCV để xử lý hình ảnh và Pandas để quản lý dữ liệu, tạo nên một quy trình hoàn chỉnh từ việc thu nhận hình ảnh đến dự đoán màu sắc.

Về mặt hiệu suất, thuật toán K-Nearest Neighbors (KNN) đã thể hiện kết quả tốt nhất với độ chính xác 75.05%, cao hơn so với Support Vector Machine (SVM) đạt 70.52% và Decision Tree đạt 61.83%. Hệ thống có khả năng xử lý và nhận diện màu sắc từ camera theo thời gian thực, đồng thời cho phép người dùng tương tác trực tiếp với kết quả nhận diện.

### **Một số hạn chế:**

Mặc dù đã đạt được những kết quả khả quan, hệ thống vẫn còn một số hạn chế cần được cải thiện. Độ chính xác của các mô hình còn khá thấp, chưa đạt được mức độ tin cậy cao trong việc nhận diện màu sắc. Điều này có thể do nhiều nguyên nhân như điều kiện ánh sáng thay đổi, chất lượng camera, và sự đa dạng của màu sắc trong thực tế.

Ngoài ra, hệ thống còn phụ thuộc nhiều vào chất lượng của dữ liệu huấn luyện và có thể gặp khó khăn khi xử lý các trường hợp màu sắc đặc biệt hoặc điều kiện môi trường không thuận lợi. Việc xử lý theo thời gian thực đôi khi còn gặp độ trễ, ảnh hưởng đến trải nghiệm người dùng.

### **Hướng phát triển:**

Để cải thiện hiệu quả của hệ thống, một số hướng phát triển như sau:

1. Nâng cao chất lượng dữ liệu huấn luyện bằng cách mở rộng tập dữ liệu và đa dạng hóa các mẫu màu sắc.
2. Tích hợp các kỹ thuật xử lý ảnh nâng cao để giảm thiểu ảnh hưởng của điều kiện ánh sáng và nhiễu.
3. Thử nghiệm với các mô hình học sâu như CNN hoặc các biến thể của các thuật toán hiện có để cải thiện độ chính xác.

4. Phát triển giao diện người dùng thân thiện hơn và tối ưu hóa hiệu suất xử lý theo thời gian thực.
5. Tích hợp các tính năng bổ sung như khả năng học và cập nhật mô hình dựa trên phản hồi của người dùng.

## TÀI LIỆU THAM KHẢO

- [1] <https://nguyenvanhieu.vn/thu-vien-pandas-trong-python/>
- [2] <https://pypi.org/project/opencv-python/>
- [3] <https://colorme.vn/blog/tat-tan-tat-ve-02-he-mau-cmyk-va-rgb>
- [4] <https://chatgpt.com/>
- [5] [https://vi.wikipedia.org/wiki/H%E1%BB%8Dc\\_m%C3%A1y](https://vi.wikipedia.org/wiki/H%E1%BB%8Dc_m%C3%A1y)
- [6] <https://www.guru99.com/vi/unsupervised-machine-learning.html>
- [7] <https://cafedev.vn/tu-hoc-ml-hoc-tap-co-giam-sat-va-khong-giam-sat/>
- [8] <https://tailieu.tv/tai-lieu/giao-trinh-mon-xu-ly-anh-32566/>