TRƯỜNG ĐẠI HỌC CÔNG NGHỆ ĐÔNG Á
KHOA CÔNG NGHỆ THÔNG TIN

BÀI TẬP LỚN

HỌC PHẦN: XỬ LÝ ẢNH & 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	20211084	Nguyễn Ngọc Minh	DCCNTT12.10.4
2	20211133	Lê Quý Mùi	DCCNTT12.10.4
3	20211166	Nguyễn Văn Lâm	DCCNTT12.10.4

Bắc Ninh, năm 2024

TRƯỜNG ĐẠI HỌC CÔNG NGHỆ ĐÔNG Á
KHOA CÔNG NGHỆ THÔNG TIN

BÀI TẬP LỚN

HỌC PHẦN: XỬ LÝ ẢNH & 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	20211084	Nguyễn Ngọc Minh	DCCNTT12.10.4
2	20211133	Lê Quý Mùi	DCCNTT12.10.4
3	20211166	Nguyễn Văn Lâm	DCCNTT12.10.4

Bắc Ninh, năm 2024

TRƯỜNG ĐẠI HỌC CÔNG NGHỆ ĐÔNG Á

KỲ THI KẾT THÚC HỌC PHẦN

HỌC KỲ 1, NĂM HỌC 2024 – 2025

KHOA CÔNG NGHỆ THÔNG TIN

PHIẾU CHẤM THI BÀI TẬP LỚN KẾT THÚC HỌC PHẦN

Mã đề thi: ***02***

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

Lớp Tín chỉ: **XATGMT.03.K12.04.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	Nguyễn Ngọc Minh	Nguyễn Văn Lâm	Lê Quý Mùi
			20211084	20211166	20211133
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			

TT	TIÊU CHÍ	THANG ĐIỂM	Nguyễn Ngọc Minh	Nguyễn Văn Lâm	Lê Quý Mùi
			20211084	20211166	20211133
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			

TT	TIÊU CHÍ	THANG ĐIỂM	Nguyễn Ngọc Minh	Nguyễn Văn Lâm	Lê Quý Mùi
			20211084	20211166	20211133
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*			

Mục lục

Chương 1: Cơ sở lý thuyết

1.1. Biểu diễn ảnh trong máy tính

1.1.1. Thu nhận ảnh

➢ Những điều cơ bản về biểu diễn hình ảnh

Trong vũ trụ kỹ thuật số, một hình ảnh có thể được mô tả như một biểu diễn hai chiều của một cảnh, bao gồm các thành phần riêng lẻ được gọi là pixel. Mỗi pixel mang thông tin về màu sắc và cường độ của một điểm cụ thể trong hình ảnh.

Các mô hình màu phổ biến được sử dụng trong biểu diễn hình ảnh bao gồm RGB (Đỏ, Xanh lục, Xanh lam), HSV (Sắc thái, Độ bão hòa, Giá trị/Độ sáng) và CMYK (Lục lam, Đỏ tươi, Vàng, Đen). Các ứng dụng đa phương tiện khác nhau có thể thích một mô hình này hơn mô hình khác dựa trên các yêu cầu riêng biệt của chúng.

- RGB: Đỏ (0-255), Xanh lục (0-255), Xanh lam (0-255) – Được sử dụng trên màn hình máy tính.
- CMYK: Cyan (0-100%), Magenta (0-100%), Yellow (0-100%), Black (0-100%) - Được sử dụng trong in ấn chuyên nghiệp.
- HSV: Sắc độ (0-360), Độ bão hòa (0-100%), Giá trị/Độ sáng (0-100%) - Được sử dụng trong phát sóng truyền hình.

➢ Tầm quan trọng của việc biểu diễn hình ảnh trong xử lý dữ liệu

Hiểu được cách biểu diễn hình ảnh là chìa khóa để xử lý dữ liệu hiệu quả, đặc biệt là khi xử lý hình ảnh trong các lĩnh vực ứng dụng như thị giác máy tính, học máy và đồ họa. Thông tin thu được từ hình ảnh số hóa có thể được xử lý theo nhiều cách khác nhau để làm nổi bật các tính năng cụ thể hoặc đạt được hiệu ứng nhất định.

➢ Cách thức lưu trữ và truy xuất hình ảnh trong hệ thống

Hình ảnh được lưu trữ trong hệ thống máy tính dưới dạng tệp lớn chứa dữ liệu số được sắp xếp. Tệp hình ảnh số chứa thông tin về các thuộc tính của hình ảnh, chẳng hạn như kích thước và độ phân giải, cũng như các giá trị màu riêng lẻ của từng pixel.

Các bước liên quan đến cách hình ảnh được lưu trữ và truy xuất trong hệ thống máy tính có thể được trình bày trong bảng:

Các bước	Mô tả
Chụp ảnh	Hình ảnh được chụp từ thiết bị như máy ảnh kỹ thuật số, máy quét,...
Chuyển đổi ảnh	Tín hiệu tương tự (analog) của ảnh được chuyển thành tín hiệu số (digital)
Xử lý số hoá	Ảnh số hóa sẵn sàng để phân tích hoặc xử lý trên máy tính.
Lưu trữ	Ảnh được lưu trữ dưới dạng file nhị phân trong phương tiện lưu trữ.
Truy xuất	File ảnh được lấy ra khi cần và hiển thị trên màn hình kỹ thuật số.

- Cảm biến hình ảnh: Sử dụng cảm biến CCD hoặc CMOS để chuyển đổi ánh sáng thành tín hiệu điện tử.
- Xử lý tín hiệu số: Mã hóa tín hiệu ánh sáng thành các pixel có giá trị số, thường từ 0 đến 255 cho mỗi kênh màu (R, G, B).
- Lưu trữ ảnh: Lưu dưới các định dạng phổ biến như JPEG, PNG, BMP.

1.1.2. Các hệ màu trong máy tính

- **RGB (Red, Green, Blue)**

Nguyên lý hoạt động:

- Hệ màu RGB dựa trên cách các màn hình hiển thị màu bằng cách pha trộn ba màu cơ bản: đỏ, xanh lá cây, và xanh dương. Khi ba màu này được kết hợp với nhau ở các mức độ khác nhau, chúng tạo ra một dải màu sắc rộng.
- Cường độ của mỗi màu được xác định bằng giá trị từ 0 đến 255 (cho 8-bit mỗi kênh). Một số hệ RGB có thể dùng giá trị từ 0.0 đến 1.0 (cho các hệ thống chuẩn hóa).

Ứng dụng: Dùng chủ yếu trong các thiết bị hiển thị điện tử như màn hình máy tính, TV, máy chiếu, máy ảnh kỹ thuật số.

Công thức: Mỗi màu có thể được biểu diễn dưới dạng (R, G, B), với R, G, B là giá trị từ 0 đến 255.

Ví dụ:

(255, 0, 0) là màu đỏ.

(0, 255, 0) là màu xanh lá cây.

(0, 0, 255) là màu xanh dương.

Đặc điểm:

- Tạo ra màu sáng và sắc nét.
- Thích hợp cho các thiết bị hiển thị phát sáng (màn hình).

➢ **CMYK (Cyan, Magenta, Yellow, Black)**

Nguyên lý hoạt động:

- CMYK là hệ màu phụ thuộc vào quá trình in ấn. Thay vì sử dụng các màu phát sáng như RGB, CMYK sử dụng các màu mực in: cyan (lục lam), magenta (đỏ tươi), yellow (vàng), và black (đen).
- Kết hợp ba màu cyan, magenta, và yellow tạo ra màu sắc cơ bản, nhưng đen (K) được thêm vào để cải thiện độ sâu và chi tiết màu sắc.

Ứng dụng: Dùng chủ yếu trong in ấn màu, đặc biệt là in ấn offset và các loại máy in màu.

Công thức: Mỗi màu được biểu diễn dưới dạng (C, M, Y, K), với giá trị mỗi thành phần từ 0 đến 1.

Ví dụ:

(0, 1, 1, 0) là màu đỏ thuần.

(1, 0, 0, 0) là màu cyan.

Đặc điểm:

- Phù hợp với quá trình in ấn, vì màu sắc được tạo ra từ việc phủ mực lên giấy.
- Không có khả năng tái tạo các màu sắc sáng như RGB.

➢ **HSV (Hue, Saturation, Value)**

Nguyên lý hoạt động:

- Hệ màu HSV mô phỏng cách con người cảm nhận và phân loại màu sắc. Hue (sắc độ) biểu thị màu sắc chính (từ 0° đến 360° trên bánh xe màu sắc). Saturation (độ bão hòa) thể hiện cường độ của màu (từ 0 đến 1, với 0 là xám và 1 là màu sắc đậm nhất).
- Value (giá trị) là độ sáng của màu (từ 0 đến 1, với 0 là đen và 1 là sáng nhất).

Ứng dụng: Thường được sử dụng trong chỉnh sửa hình ảnh, giao diện người dùng đồ họa, và phần mềm thiết kế.

Công thức: Màu được biểu diễn dưới dạng (H, S, V).

Ví dụ:

(120°, 1, 1) là màu xanh lá cây sáng.

(240°, 1, 1) là màu xanh dương sáng.

Đặc điểm:

- Dễ dàng sử dụng để chọn và chỉnh sửa màu sắc từ giao diện đồ họa.
- Hệ màu này trực quan và dễ hiểu đối với người thiết kế.

➢ **HSL (Hue, Saturation, Lightness)**

Nguyên lý hoạt động:

- HSL rất giống với HSV, nhưng thay vì Value (độ sáng), HSL sử dụng Lightness (độ sáng tối). Lightness thay đổi từ 0 (đen) đến 1 (trắng), với giá trị 0.5 đại diện cho màu sắc thuần khiết.
- Hue (sắc độ) và Saturation (độ bão hòa) giống như trong HSV.

Ứng dụng: HSL thường được dùng trong các công cụ chỉnh sửa màu sắc của phần mềm đồ họa và thiết kế web.

Công thức: Màu được biểu diễn dưới dạng (H, S, L).

Ví dụ:

(240°, 1, 0.5) là màu xanh dương thuần.

(120°, 1, 0.5) là màu xanh lá cây thuần.

Đặc điểm: HSL dễ sử dụng cho việc chọn màu và tinh chỉnh độ sáng tối của màu sắc.

➢ **YCbCr (Y, Chrominance Blue, Chrominance Red)**

Nguyên lý hoạt động: Hệ màu YCbCr tách riêng độ sáng (Y) và các thành phần màu sắc (Cb và Cr). Y là độ sáng, trong khi Cb và Cr biểu thị độ lệch màu xanh lam và đỏ tươi, tương ứng.

Ứng dụng: Sử dụng phổ biến trong video, truyền hình và các phương thức nén hình ảnh như JPEG và MPEG.

Công thức: Màu được biểu diễn dưới dạng (Y, Cb, Cr).

Ví dụ:

(0.5, 0, 0) là màu đỏ thuần.

(0.5, 0.5, 0.5) là màu xám trung tính.

Đặc điểm: Tách biệt thành phần sáng và màu sắc giúp dễ dàng xử lý và nén video và hình ảnh.

- **Lab (CIELAB)**

Nguyên lý hoạt động:

- Hệ màu CIELAB được thiết kế để mô phỏng cách mà mắt người nhận biết màu sắc, và không phụ thuộc vào thiết bị hiển thị.
- L (Lightness) là độ sáng, trong khi a và b là các thành phần màu từ đỏ đến xanh và vàng đến xanh dương.

Ứng dụng: Hệ màu Lab được sử dụng trong các ứng dụng chỉnh sửa màu sắc và nhận dạng hình ảnh.

Công thức: Màu được biểu diễn dưới dạng (L, a, b).

Ví dụ:

(50, 60, 30) là một màu với độ sáng trung bình và có thành phần đỏ và vàng.

Đặc điểm:

- Hệ màu Lab là không gian màu toàn cầu, giúp tái tạo màu sắc một cách chính xác.
- Được sử dụng trong ngành công nghiệp in ấn cao cấp và các phần mềm xử lý ảnh.

- **XYZ (CIEXYZ)**

Nguyên lý hoạt động:

- XYZ là hệ màu gốc của hệ thống mô phỏng màu sắc của con người, được thiết kế bởi CIE (Commission Internationale de l'Eclairage).
- Không gian XYZ dựa trên ba trục cơ bản, với các giá trị X, Y, và Z là các thành phần ánh sáng đỏ, xanh lá và xanh dương.

Ứng dụng: Dùng chủ yếu trong nghiên cứu màu sắc và trong các hệ thống chuyển đổi màu sắc.

Công thức: Màu được biểu diễn dưới dạng (X, Y, Z).

Ví dụ:

(0.3, 0.4, 0.2) là màu ánh sáng đỏ.

Đặc điểm:

- Là không gian màu tiêu chuẩn giúp chuyển đổi giữa các hệ màu khác.
- Thường được sử dụng trong các nghiên cứu khoa học và chuyển đổi màu sắc.

1.2. Tổng quan về học máy

Học máy là nhánh của trí tuệ nhân tạo, giúp xây dựng các hệ thống có khả năng học và đưa ra quyết định từ dữ liệu.

1.2.1. Học có giám sát

Học máy có giám sát, hay còn gọi là học máy có giám sát, là một phân loại con của học máy và trí tuệ nhân tạo. Nó được định nghĩa bởi việc sử dụng các bộ dữ liệu đã được gán nhãn để huấn luyện các thuật toán, giúp phân loại dữ liệu hoặc dự đoán kết quả một cách chính xác.

Khi dữ liệu đầu vào được đưa vào mô hình, mô hình điều chỉnh các trọng số của nó cho đến khi mô hình được điều chỉnh phù hợp, điều này diễn ra trong quá trình kiểm tra chéo (cross-validation). Học máy có giám sát giúp các tổ chức giải quyết nhiều vấn đề thực tế ở quy mô lớn, chẳng hạn như phân loại thư rác vào một thư mục riêng biệt khỏi hộp thư đến của bạn. Nó có thể được sử dụng để xây dựng các mô hình học máy chính xác cao.

- Cách học máy có giám sát hoạt động:

Học máy có giám sát sử dụng một bộ dữ liệu huấn luyện để dạy các mô hình tạo ra kết quả mong muốn. Bộ dữ liệu huấn luyện này bao gồm các đầu vào và đầu ra đúng, giúp mô hình học hỏi theo thời gian. Thuật toán đo lường độ chính xác của nó thông qua hàm mất mát, điều chỉnh cho đến khi sai số được giảm thiểu đủ mức.

Học máy có giám sát có thể được chia thành hai loại vấn đề khi khai thác dữ liệu phân loại và hồi quy:

- Phân loại sử dụng một thuật toán để phân loại chính xác dữ liệu thử nghiệm vào các hạng mục cụ thể. Nó nhận diện các thực thể cụ thể trong bộ dữ liệu và cố gắng rút ra kết luận về cách những thực thể đó nên được gán nhãn hoặc định nghĩa.
- Hồi quy được sử dụng để hiểu mối quan hệ giữa các biến phụ thuộc và độc lập. Nó thường được sử dụng để đưa ra các dự đoán, chẳng hạn như doanh thu bán hàng cho một doanh nghiệp. Các thuật toán hồi quy phổ biến bao gồm hồi quy tuyến tính, hồi quy logistic và hồi quy đa thức.

1.2.1.1. K-Nearest Neighbors [1]

K-Nearest Neighbors (KNN): K-Nearest Neighbors, hay thuật toán KNN, là một thuật toán phi tham số phân loại các điểm dữ liệu dựa trên sự gần gũi và mối liên hệ của chúng với các dữ liệu khác. Thuật toán này giả định rằng các điểm dữ liệu tương tự có thể được tìm thấy gần nhau. Do đó, nó tính toán khoảng cách giữa các điểm dữ liệu, thường qua khoảng cách Euclid, và sau đó gán một danh mục dựa trên danh mục hoặc trung bình thường xuyên nhất. Sự dễ sử dụng và thời gian tính toán thấp khiến KNN trở thành thuật toán ưa chuộng của các nhà khoa học dữ liệu, nhưng khi bộ dữ liệu thử nghiệm tăng lên, thời gian xử lý kéo dài, khiến nó trở nên kém hấp dẫn cho các nhiệm vụ phân loại. KNN thường được sử dụng trong các hệ thống gợi ý và nhận diện hình ảnh.

Các bước trong KNN:

- Ta có D là tập các điểm dữ liệu đã được gắn nhãn và A là dữ liệu chưa được phân loại.
- Đo khoảng cách (Euclidian, Manhattan, Minkowski, Minkowski hoặc Trọng số) từ dữ liệu mới A đến tất cả các dữ liệu khác đã được phân loại trong D.
- Chọn K (K là tham số mà bạn định nghĩa) khoảng cách nhỏ nhất.
- Kiểm tra danh sách các lớp có khoảng cách ngắn nhất và đếm số lượng của mỗi lớp xuất hiện.
- Lấy đúng lớp (lớp xuất hiện nhiều lần nhất).
- Lớp của dữ liệu mới là lớp mà bạn đã nhận được ở bước 5.

Ưu điểm:

- Thuật toán đơn giản, dễ dàng triển khai.
- Độ phức tạp tính toán nhỏ.
- Xử lý tốt với tập dữ liệu nhiễu

Nhược điểm:

- Với K nhỏ dễ gặp nhiễu dẫn tới kết quả đưa ra không chính xác
- Cần nhiều thời gian để thực hiện do phải tính toán khoảng cách với tất cả các đối tượng trong tập dữ liệu.
- Cần chuyển đổi kiểu dữ liệu thành các yếu tố định tính.

1.2.1.2. Support vector machines (SVM) [1]

Support vector machines (SVM): SVM là một mô hình học máy có giám sát phổ biến được phát triển bởi Vladimir Vapnik, được sử dụng cho cả phân loại và hồi quy dữ liệu. Tuy nhiên, nó thường được sử dụng cho các vấn đề phân loại, xây dựng một siêu phẳng nơi khoảng cách giữa hai lớp điểm dữ liệu là lớn nhất. Siêu phẳng này được gọi là ranh giới quyết định, phân chia các lớp điểm dữ liệu (ví dụ, cam so với táo) ở hai bên của mặt phẳng.

Các bước thực hiện:

- Tiền xử lý dữ liệu: Biến đổi dữ liệu về dạng có thể phân tách tuyến tính (nếu cần).
- Xây dựng mô hình: Tìm siêu phẳng tối ưu bằng cách giải bài toán tối ưu hóa.
- Sử dụng hạt nhân (Kernel): Với dữ liệu phi tuyến, áp dụng kernel để chuyển đổi không gian.
- Phân loại: Phân lớp dữ liệu mới dựa trên khoảng cách đến siêu phẳng.

Ưu điểm:

- Tốt cho bài toán phân loại với dữ liệu phức tạp.
- Hiệu quả cao với dữ liệu nhỏ và vừa.

Nhược điểm:

- Tốn thời gian tính toán với dữ liệu lớn.
- Khó chọn kernel phù hợp.

1.2.1.3. Hồi quy tuyến tính (Linear Regression) [1]

Hồi quy tuyến tính được sử dụng để xác định mối quan hệ giữa một biến phụ thuộc và một hoặc nhiều biến độc lập và thường được dùng để dự đoán kết quả trong tương lai. Khi chỉ có một biến độc lập và một biến phụ thuộc, nó được gọi là hồi quy tuyến tính đơn giản. Khi số lượng biến độc lập tăng lên, nó được gọi là hồi quy tuyến tính bội. Với mỗi loại hồi quy tuyến tính, mục tiêu là vẽ một đường thẳng phù hợp nhất, được tính toán qua phương pháp bình phương tối thiểu. Tuy nhiên, khác với các mô hình hồi quy khác, đường thẳng này là đường thẳng khi vẽ trên đồ thị.

1.2.1.4. Hồi quy logistic [2]

Trong khi hồi quy tuyến tính được sử dụng khi các biến phụ thuộc là liên tục, hồi quy logistic được chọn khi biến phụ thuộc là dạng phân loại, có đầu ra nhị phân như "đúng" và "sai" hoặc "có" và "không". Mặc dù cả hai mô hình hồi quy đều cố gắng hiểu

mối quan hệ giữa các đầu vào dữ liệu, hồi quy logistic chủ yếu được sử dụng để giải quyết các bài toán phân loại nhị phân, chẳng hạn như nhận diện thư rác.

Các bước thực hiện:

- Thu thập dữ liệu: Tập dữ liệu với các biến đầu vào và đầu ra nhị phân (0 hoặc 1).
- Xây dựng mô hình: Sử dụng hàm logistic để biểu diễn xác suất.
- Tối ưu hóa: Dùng Gradient Descent để tìm tham số tối ưu (
- Dự đoán: Quyết định lớp dựa trên ngưỡng (thường là 0.5).

Ưu điểm:

- Hiệu quả cho phân loại nhị phân.
- Cung cấp xác suất dự đoán.

Nhược điểm:

- Không áp dụng được cho dữ liệu phi tuyến mà không có biến đổi đặc trưng.
- Không xử lý được dữ liệu nhiều lớp một cách trực tiếp.

1.2.1.5. Naive Bayes [1]

Naive Bayes là phương pháp phân loại dựa trên nguyên lý độc lập có điều kiện của các lớp theo Định lý Bayes. Điều này có nghĩa là sự xuất hiện của một đặc trưng không ảnh hưởng đến sự xuất hiện của một đặc trưng khác trong xác suất của một kết quả nhất định, và mỗi biến dự báo có tác động bằng nhau đến kết quả đó. Có ba loại bộ phân loại Naïve Bayes: Multinomial Naïve Bayes, Bernoulli Naïve Bayes, và Gaussian Naïve Bayes. Phương pháp này chủ yếu được sử dụng trong phân loại văn bản, nhận diện thư rác, và hệ thống gợi ý.

1.2.1.6. Mạng nơ-ron [2]

Chủ yếu được sử dụng cho các thuật toán học sâu, mạng nơ-ron xử lý dữ liệu huấn luyện bằng cách mô phỏng sự kết nối của não người thông qua các lớp nơ-ron. Mỗi nơ-ron bao gồm các đầu vào, trọng số, độ lệch (hoặc ngưỡng) và đầu ra. Nếu giá trị đầu ra vượt quá một ngưỡng nhất định, nơ-ron "bắn" hoặc kích hoạt, chuyển dữ liệu đến lớp tiếp theo trong mạng. Mạng nơ-ron học chức năng ánh xạ này thông qua học có giám sát, điều chỉnh dựa trên hàm mất mát qua quá trình giảm dần theo độ dốc. Khi hàm chi phí gần bằng hoặc bằng không, chúng ta có thể tự tin vào độ chính xác của mô hình trong việc đưa ra câu trả lời đúng.

1.2.1.7. **Random forest** [2]

Random forest là một thuật toán học máy có giám sát linh hoạt khác, được sử dụng cho cả phân loại và hồi quy. "Rừng" đề cập đến một tập hợp các cây quyết định không có sự tương quan, sau đó được kết hợp lại để giảm phương sai và tạo ra dự đoán dữ liệu chính xác hơn.

Các bước thực hiện:

- Tạo mẫu dữ liệu ngẫu nhiên: Chọn ngẫu nhiên các mẫu và đặc trưng từ tập dữ liệu.
- Xây dựng cây quyết định: Tạo nhiều cây quyết định trên các mẫu dữ liệu khác nhau.
- Tổng hợp dự đoán: Lấy trung bình (đối với hồi quy) hoặc lấy lớp phổ biến nhất (đối với phân loại).

Ưu điểm:

- Giảm overfitting.
- Xử lý tốt cả bài toán phân loại và hồi quy.

Nhược điểm:

- Mô hình phức tạp và tốn tài nguyên.
- Khó giải thích.

1.2.1.8. **Cây quyết định (Decision Tree)** [4]

Các bước thực hiện:

- Chọn đặc trưng tốt nhất: Dựa trên chỉ số Gini hoặc Entropy.
- Chia nhánh: Lặp lại việc chia nhánh cho đến khi đạt điều kiện dừng.
- Dự đoán: Đi theo nhánh dựa trên các điều kiện của dữ liệu mới.

Ưu điểm:

- Dễ hiểu và giải thích.
- Xử lý cả dữ liệu phân loại và hồi quy.

Nhược điểm:

- Dễ overfitting nếu không cắt tỉa (pruning).
- Nhạy cảm với dữ liệu nhiễu.

1.2.2. Học không giám sát

Học không giám sát, còn được gọi là học máy không giám sát , sử dụng các thuật toán học máy (ML) để phân tích và nhóm các tập dữ liệu không có nhãn. Các thuật toán này khám phá các mẫu ẩn hoặc nhóm dữ liệu mà không cần sự can thiệp của con người. Mô hình học không giám sát được sử dụng cho ba nhiệm vụ chính: phân cụm (clustering), liên kết (association) và giảm chiều dữ liệu (dimensionality reduction). Dưới đây là định nghĩa của từng phương pháp và các thuật toán, cách tiếp cận phổ biến để thực hiện chúng một cách hiệu quả.

1.2.2.1. Phân cụm (Clustering) [2]

Phân cụm là một kỹ thuật khai phá dữ liệu nhằm nhóm các dữ liệu không được gắn nhãn dựa trên sự tương đồng hoặc khác biệt của chúng. Các thuật toán phân cụm được sử dụng để xử lý các đối tượng dữ liệu thô, chưa được phân loại, thành các nhóm được biểu diễn bằng các cấu trúc hoặc mẫu trong thông tin.

Các loại phân cụm chính:

- Phân cụm độc quyền (Exclusive Clustering)
- Phân cụm chồng chéo (Overlapping Clustering)
- Phân cụm phân cấp (Hierarchical Clustering)
- Phân cụm xác suất (Probabilistic Clustering)
- Phân cụm độc quyền (Exclusive Clustering)
- Phân cụm độc quyền là một hình thức nhóm mà mỗi điểm dữ liệu chỉ có thể tồn tại trong một cụm duy nhất. Phương pháp này còn được gọi là phân cụm "cứng".

Thuật toán K-means Clustering:

Định nghĩa: Đây là một ví dụ phổ biến của phân cụm độc quyền, trong đó các điểm dữ liệu được gán vào K nhóm (cụm). K đại diện cho số cụm dựa trên khoảng cách từ từng điểm đến tâm cụm (centroid).

Hoạt động:

- Chọn ngẫu nhiên
- K tâm cụm ban đầu.
- Gán từng điểm dữ liệu vào cụm gần nhất dựa trên khoảng cách.
- Tính toán lại tâm cụm dựa trên các điểm đã gán.
- Lặp lại cho đến khi tâm cụm không còn thay đổi hoặc đạt hội tụ.

Ưu điểm:

- Đơn giản, dễ hiểu và dễ triển khai.
- Hiệu quả cho dữ liệu lớn.

Nhược điểm:

- Nhạy cảm với giá trị K được chọn.
- Nhạy cảm với dữ liệu ngoại lai.

Ứng dụng: Phân đoạn thị trường, phân cụm tài liệu, phân đoạn hình ảnh, nén ảnh.

- Phân cụm chồng chéo (Overlapping Clustering)

Phương pháp này khác với phân cụm độc quyền ở chỗ cho phép một điểm dữ liệu thuộc về nhiều cụm với các mức độ thành viên khác nhau. Fuzzy K-means Clustering là một ví dụ phổ biến của phương pháp này.

- Phân cụm phân cấp (Hierarchical Clustering)

Phân cụm phân cấp, còn được gọi là phân tích cụm theo thứ bậc (Hierarchical Cluster Analysis - HCA), là một thuật toán phân cụm không giám sát được chia thành hai cách:

- Phân cụm tăng dần (Agglomerative Clustering):

Định nghĩa: Đây là phương pháp "từ dưới lên". Ban đầu, mỗi điểm dữ liệu được coi là một cụm riêng lẻ và dần dần hợp nhất các cụm dựa trên sự tương đồng cho đến khi đạt được một cụm duy nhất.

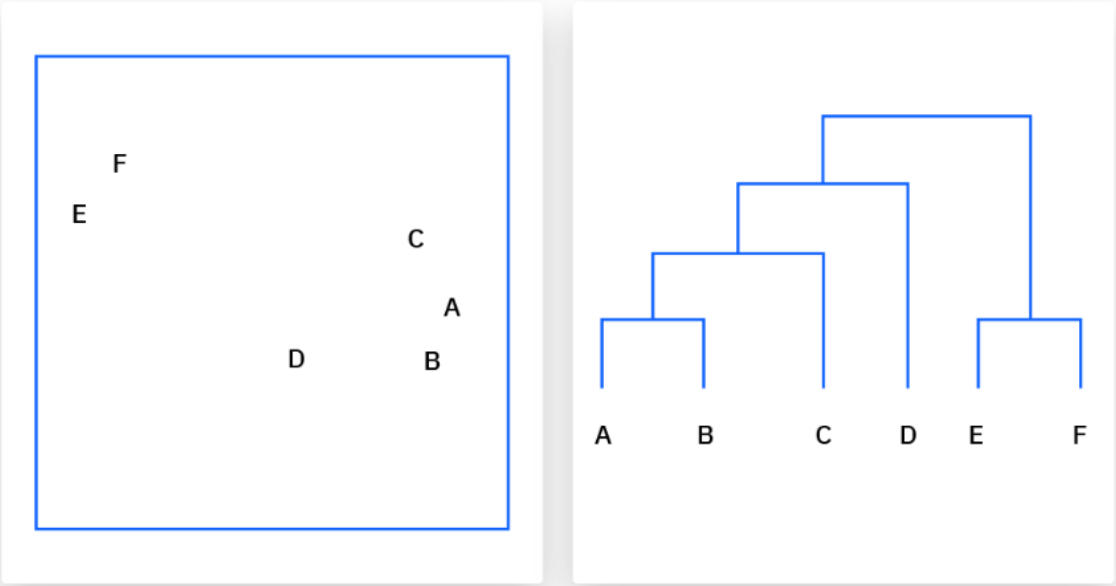
Các phương pháp đo sự tương đồng:

- Ward’s Linkage: Khoảng cách giữa hai cụm được xác định bởi sự gia tăng tổng bình phương sau khi các cụm được hợp nhất.
- Average Linkage: Khoảng cách trung bình giữa hai điểm trong mỗi cụm.
- Complete (Maximum) Linkage: Khoảng cách lớn nhất giữa hai điểm trong mỗi cụm.
- Single (Minimum) Linkage: Khoảng cách nhỏ nhất giữa hai điểm trong mỗi cụm.

- Phân cụm phân tách (Divisive Clustering):

Định nghĩa: Đây là phương pháp "từ trên xuống". Ban đầu, toàn bộ dữ liệu được coi là một cụm lớn nhất và sau đó dần chia nhỏ cụm này.

Đặc điểm: Phương pháp này ít được sử dụng hơn phân cụm tăng dần.



Hình dung:

Quá trình phân cụm phân cấp thường được biểu diễn bằng dendrogram – một biểu đồ dạng cây cho thấy việc gộp hoặc tách các điểm dữ liệu qua từng bước lặp.

- Phân cụm xác suất (Probabilistic Clustering)

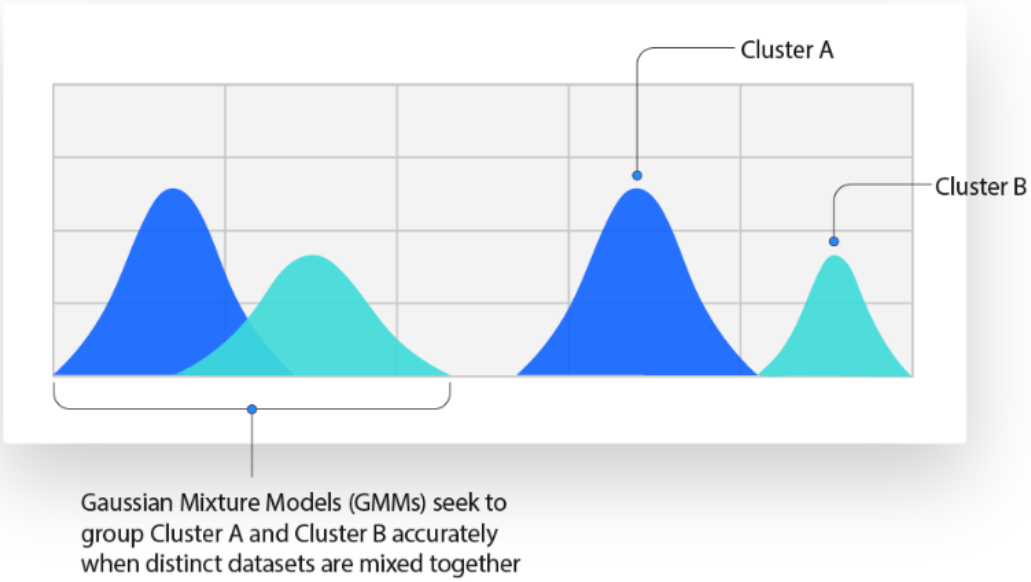
Phương pháp phân cụm xác suất sử dụng mô hình xác suất để giải quyết các vấn đề ước tính mật độ hoặc phân cụm "mềm". Trong phân cụm xác suất, các điểm dữ liệu được nhóm dựa trên xác suất chúng thuộc về một phân phối cụ thể.

Gaussian Mixture Models (GMM): [2]

Định nghĩa: Đây là một phương pháp phân cụm xác suất phổ biến nhất, được xếp vào loại mô hình hỗn hợp (mixture models).

Hoạt động:

- GMM xác định điểm dữ liệu thuộc về phân phối chuẩn (Gaussian) nào.
- Các biến như trung bình và phương sai ban đầu không được biết, và một biến tiềm ẩn được giả định để nhóm dữ liệu đúng cách.
- Thuật toán thường dùng: Expectation-Maximization (EM), để ước tính xác suất các điểm dữ liệu thuộc về một cụm cụ thể.



Ưu điểm: Phù hợp với các trường hợp dữ liệu phức tạp hơn, không nhất thiết phải cứng nhắc như K-means.

Nhược điểm:

- Nhạy cảm với dữ liệu ngoại lai và khởi tạo giá trị ban đầu.
- Học không giám sát làm việc với dữ liệu không gán nhãn, tập trung vào khám phá cấu trúc ẩn. Một số thuật toán:
- K-Means Clustering: Phân cụm dữ liệu thành k nhóm dựa trên khoảng cách đến tâm cụm.
- DBSCAN (Density-Based Spatial Clustering): Phân cụm dựa trên mật độ, phù hợp với dữ liệu có hình dạng phức tạp.
- PCA (Principal Component Analysis): Giảm chiều dữ liệu bằng cách tìm các trục chính, giúp biểu diễn dữ liệu tốt hơn.

1.2.3. Học bán giám sát

Học bán giám sát (Semi-Supervised Learning) là một nhánh của học máy, kết hợp giữa học có giám sát và học không giám sát bằng cách sử dụng cả dữ liệu có nhãn và dữ liệu không có nhãn để huấn luyện các mô hình AI thực hiện các nhiệm vụ như phân loại và hồi quy.

Học bán giám sát thường được áp dụng trong các trường hợp tương tự như học có giám sát, nhưng điểm khác biệt là nó khai thác dữ liệu không nhãn nhằm bổ sung và tăng cường hiệu quả của quá trình huấn luyện.

- Dữ liệu có nhãn và học máy

Học có giám sát yêu cầu dữ liệu có nhãn – các điểm dữ liệu này đã được gắn thông tin cụ thể và đúng về đầu ra mong muốn. Trong quá trình huấn luyện, mô hình được tối ưu hóa dựa trên hàm mất mát (loss function), hàm này đo lường sự khác biệt giữa dự đoán của mô hình và thông tin đầu ra thực tế.

Tuy nhiên, việc gắn nhãn dữ liệu thường tốn thời gian và đòi hỏi nguồn lực, đặc biệt đối với các bài toán phức tạp như phát hiện đối tượng hay phân đoạn hình ảnh. Trong những trường hợp này, học bán giám sát cho phép tận dụng tốt dữ liệu có nhãn khan hiếm, đồng thời khai thác dữ liệu không nhãn dồi dào để đạt hiệu quả cao hơn.

- Học bán giám sát so với các phương pháp khác
  - Học bán giám sát và học có giám sát: Học có giám sát chỉ sử dụng dữ liệu có nhãn trong quá trình huấn luyện. Trong khi đó, học bán giám sát kết hợp cả dữ liệu có nhãn và không nhãn, từ đó cải thiện độ chính xác nhờ bổ sung thông tin từ dữ liệu không nhãn.
  - Học bán giám sát và học không giám sát: Học không giám sát không sử dụng dữ liệu có nhãn và không dựa vào hàm mất mát để đo lường hiệu quả. Học bán giám sát có thể kết hợp các kỹ thuật không giám sát (như tiền huấn luyện – pretraining) để cải thiện hiệu quả trên dữ liệu có nhãn.
- Cách hoạt động của học bán giám sát

Học bán giám sát hoạt động dựa trên một số giả định về mối quan hệ giữa dữ liệu không nhãn và nhãn:

- Giả định cụm (Cluster Assumption): Các điểm dữ liệu trong cùng một cụm thường thuộc cùng một nhãn.
- Giả định độ mượt (Smoothness Assumption): Các điểm dữ liệu gần nhau trong không gian đầu vào có xu hướng thuộc cùng một nhãn.
- Giả định mật độ thấp (Low-Density Assumption): Ranh giới phân tách giữa các nhãn thường nằm ở các vùng mật độ thấp của dữ liệu.

- Giả định đa tạp (Manifold Assumption): Dữ liệu có thể nằm trên các đa tạp (manifold) có chiều thấp hơn so với không gian ban đầu, điều này giúp giảm độ phức tạp của bài toán.

Phương pháp phổ biến trong học bán giám sát [3]

- Lan truyền nhãn (Label Propagation): Sử dụng đồ thị để lan truyền nhãn từ dữ liệu có nhãn sang dữ liệu không nhãn dựa trên độ tương tự giữa các điểm dữ liệu.
- Huấn luyện tự thân (Self-Training): Sử dụng một mô hình ban đầu dự đoán nhãn cho dữ liệu không nhãn, sau đó tái huấn luyện mô hình với cả dữ liệu có nhãn ban đầu và các nhãn dự đoán.
- Huấn luyện đồng bộ (Co-Training): Sử dụng nhiều mô hình hoặc quan điểm khác nhau để gán nhãn dữ liệu không nhãn, từ đó giảm thiểu rủi ro khuếch đại sai sót trong quá trình dự đoán.
- Tiền xử lý không giám sát (Unsupervised Pre-Processing): Áp dụng các kỹ thuật học không giám sát như giảm chiều dữ liệu hoặc trích xuất đặc trưng trước khi sử dụng dữ liệu có nhãn để huấn luyện mô hình.

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

2.1. Thư viện

2.1.1. OpenCV

Project OpenCV được bắt đầu từ Intel năm 1999 bởi Gary Bradsky. OpenCV viết tắt cho Open Source Computer Vision Library. OpenCV là thư viện nguồn mở hàng đầu cho Computer Vision và Machine Learning, và hiện có thêm tính năng tăng tốc GPU cho các hoạt động theo real-time.

OpenCV được phát hành theo giấy phép BSD (*), do đó nó miễn phí cho cả học tập và sử dụng với mục đích thương mại. Nó có trên các giao diện C++, C, Python và Java và hỗ trợ Windows, Linux, Mac OS, iOS và Android. OpenCV được thiết kế để hỗ trợ hiệu quả về tính toán và chuyên dùng cho các ứng dụng real-time (thời gian thực). Nếu được viết trên C/C++ tối ưu, thư viện này có thể tận dụng được bộ xử lý đa lõi (multi-core processing).

OpenCV có một cộng đồng người dùng khá hùng hậu hoạt động trên khắp thế giới bởi nhu cầu cần đến nó ngày càng tăng theo xu hướng chạy đua về sử dụng computer vision của các công ty công nghệ. OpenCV hiện được ứng dụng rộng rãi toàn cầu, với cộng đồng hơn 47.000 người, với nhiều mục đích và tính năng khác nhau từ interactive art, đến khai thác mỏ, khai thác web map hoặc qua robotic cao cấp.

Ứng dụng OpenCV

OpenCV được sử dụng cho đa dạng nhiều mục đích và ứng dụng khác nhau bao gồm:

- Hình ảnh street view
- Kiểm tra và giám sát tự động
- Robot và xe hơi tự lái
- Phân tích hình ảnh y học
- Tìm kiếm và phục hồi hình ảnh/video

- Phim – cấu trúc 3D từ chuyển động
- Nghệ thuật sắp đặt tương tác

Tính năng và các module phổ biến của OpenCV

Theo tính năng và ứng dụng của OpenCV, có thể chia thư viện này thánh các nhóm tính năng và module tương ứng như sau:

- Xử lý và hiển thị Hình ảnh/ Video/ I/O (core, imgproc, highgui)
- Phát hiện các vật thể (objdetect, features2d, nonfree)
- Geometry-based monocular hoặc stereo computer vision (calib3d, stitching, videostab)
- Computational photography (photo, video, superres)
- Machine learning & clustering (ml, flann)
- CUDA acceleration (gpu)

OpenCV có cấu trúc module, nghĩa là gói bao gồm một số thư viện liên kết tĩnh (static libraries) hoặc thư viện liên kết động (shared libraries). Xin phép liệt kê một số định nghĩa chi tiết các module phổ biến có sẵn như sau:

- Core functionality (core) – module nhỏ gọn để xác định cấu trúc dữ liệu cơ bản, bao gồm mảng đa chiều dày đặc và nhiều chức năng cơ bản được sử dụng bởi tất cả các module khác.
- Image Processing (imgproc) – module xử lý hình ảnh gồm cả lọc hình ảnh tuyến tính và phi tuyến (linear and non-linear image filtering), phép biến đổi hình học (chỉnh size, afin và warp phối cảnh, ánh xạ lại dựa trên bảng chung), chuyển đổi không gian màu, biểu đồ, và nhiều cái khác.
- Video Analysis (video) – module phân tích video bao gồm các tính năng ước tính chuyển động, tách nền, và các thuật toán theo dõi vật thể.
- Camera Calibration and 3D Reconstruction (calib3d) – thuật toán hình học đa chiều cơ bản, hiệu chuẩn máy ảnh single và stereo (single and stereo camera calibration), dự đoán kiểu dáng của đối tượng (object pose estimation), thuật toán thư tín âm thanh nổi (stereo correspondence algorithms) và các yếu tố tái tạo 3D.
- 2D Features Framework (features2d) – phát hiện các đặc tính nổi bật của bộ nhận diện, bộ truy xuất thông số, thông số đối chọi.

- Object Detection (objdetect) – phát hiện các đối tượng và mô phỏng của các hàm được định nghĩa sẵn – predefined classes (vd: khuôn mặt, mắt, cốc, con người, xe hơi,…).
- High-level GUI (highgui) – giao diện dễ dùng để thực hiện việc giao tiếp UI đơn giản.
- Video I/O (videoio) – giao diện dễ dùng để thu và mã hóa video.
- GPU – Các thuật toán tăng tốc GPU từ các modun OpenCV khác.

2.1.2. Pandas

Mục đích: Pandas là thư viện rất mạnh trong Python, chuyên dụng cho việc xử lý và phân tích dữ liệu, đặc biệt là dữ liệu dạng bảng (DataFrame). Nó cung cấp một loạt các chức năng để xử lý dữ liệu từ các tệp như CSV, Excel, cơ sở dữ liệu SQL, và nhiều nguồn dữ liệu khác.

Ứng dụng của Pandas:

- DataFrame đem lại sự linh hoạt và hiệu quả trong thao tác dữ liệu và lập chỉ mục;
- Là một công cụ cho phép đọc/ ghi dữ liệu giữa bộ nhớ và nhiều định dạng file: csv, text, excel, sql database, hdf5;
- Liên kết dữ liệu thông minh, xử lý được trường hợp dữ liệu bị thiếu. Tự động đưa dữ liệu lộn xộn về dạng có cấu trúc;
- Dễ dàng thay đổi bố cục của dữ liệu;
- Tích hợp cơ chế trượt, lập chỉ mục, lấy ra tập con từ tập dữ liệu lớn.
- Có thể thêm, xóa các cột dữ liệu;
- Tập hợp hoặc thay đổi dữ liệu với group by cho phép bạn thực hiện các toán tử trên tập dữ liệu;
- Hiệu quả cao trong trộn và kết hợp các tập dữ liệu;

- Lập chỉ mục theo các chiều của dữ liệu giúp thao tác giữa dữ liệu cao chiều và dữ liệu thấp chiều;
- Tối ưu về hiệu năng;
- Pandas được sử dụng rộng rãi trong cả học thuật và thương mại. Bao gồm thống kê, thương mại, phân tích, quảng cáo,…

Chức năng nổi bật:

- Đọc và ghi tệp CSV: Pandas cho phép dễ dàng đọc dữ liệu từ các tệp CSV vào DataFrame và ghi lại chúng sau khi xử lý.
- Chỉnh sửa dữ liệu: Các chức năng như lọc, phân nhóm, xử lý thiếu dữ liệu (missing data), và thay thế giá trị rất mạnh mẽ.
- Thống kê mô tả: Pandas có các hàm tính toán thống kê như trung bình, phương sai, độ lệch chuẩn, và các chỉ số mô tả khác.
- Dễ dàng thao tác với mảng 2D: Pandas giúp dễ dàng xử lý các mảng 2D và cho phép tính toán nhanh chóng trên dữ liệu lớn.

2.1.3. Một số thư viện khác

- Thư viện numpy

Mục đích: Numpy là thư viện cốt lõi để xử lý mảng (array) và các phép toán ma trận. Nó cung cấp các chức năng tính toán nhanh chóng và tối ưu cho dữ liệu số.

Các chức năng nổi bật:

- np.array(): Tạo mảng Numpy từ một danh sách hoặc tuple.
- np.vstack(): Kết hợp các mảng dọc (theo chiều dọc).
- np.concatenate(): Nối các mảng lại với nhau (theo chiều ngang hoặc dọc).
- np.mean(axis=(0, 1)): Tính giá trị trung bình của các phần tử trong mảng theo chiều.

- Thư viện os

Mục đích: Thư viện này cung cấp các chức năng để tương tác với hệ thống tệp (file system), làm việc với đường dẫn (path), và thực hiện các thao tác tệp.

Các chức năng nổi bật:

- os.listdir(path): Liệt kê danh sách các tệp và thư mục trong thư mục path.
- os.path.join(path1, path2): Kết hợp các phần của đường dẫn thành một đường dẫn đầy đủ.

- os.path.isdir(path): Kiểm tra xem đường dẫn path có phải là thư mục hay không.

➢ Thư viện pickle

Mục đích: Thư viện này cho phép lưu trữ và phục hồi các đối tượng Python (như mô hình học máy, dữ liệu) dưới dạng nhị phân.

Các chức năng nổi bật:

- pickle.dump(obj, file): Lưu trữ đối tượng obj vào file file.
- pickle.load(file): Tải đối tượng đã lưu từ file file.

➢ Thư viện sklearn.svm (SVC)

Mục đích: SVM (Support Vector Machine) là một thuật toán học máy mạnh mẽ, chủ yếu dùng để phân loại và hồi quy.

Các chức năng nổi bật:

- SVC(kernel='linear'): Tạo mô hình SVM với kernel tuyến tính.
- fit(X_train, y_train): Huấn luyện mô hình với dữ liệu huấn luyện.
- predict(X_test): Dự đoán kết quả cho dữ liệu kiểm tra.

➢ Thư viện sklearn.neighbors (KNeighborsClassifier)

Mục đích: KNN (K-Nearest Neighbors) là thuật toán phân loại dựa trên việc so sánh điểm dữ liệu mới với các điểm dữ liệu trong tập huấn luyện.

Các chức năng nổi bật:

- KNeighborsClassifier(n_neighbors=5): Tạo mô hình K-NN với 5 lân cận.
- fit(X_train, y_train): Huấn luyện mô hình.
- predict(X_test): Dự đoán kết quả cho dữ liệu kiểm tra.

➢ Thư viện sklearn.tree (DecisionTreeClassifier)

Mục đích: Decision Tree là một thuật toán phân loại và hồi quy dựa trên cây quyết định.

Các chức năng nổi bật:

- DecisionTreeClassifier(): Tạo mô hình cây quyết định.
- fit(X_train, y_train): Huấn luyện mô hình.
- predict(X_test): Dự đoán kết quả.

➢ Thư viện sklearn.model_selection (train_test_split)

Mục đích: Dùng để chia dữ liệu thành tập huấn luyện và kiểm tra.

Chức năng nổi bật:

- train_test_split(X, y, test_size=0.2, random_state=42): Chia dữ liệu thành tập huấn luyện và kiểm tra (80% huấn luyện, 20% kiểm tra).

➢ Thư viện sklearn.metrics (accuracy_score)

Mục đích: Đánh giá độ chính xác của mô hình.

Chức năng nổi bật:

- accuracy_score(y_true, y_pred): Tính toán độ chính xác giữa nhãn thực tế y_true và nhãn dự đoán y_pred.

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

2.2.1. Kiến trúc hệ thống

Hệ thống được xây dựng theo mô hình MVC (Model-View-Controller):

- Model: Xử lý dữ liệu, huấn luyện mô hình và thực hiện dự đoán.
- View: Hiển thị giao diện, cho phép người dùng chọn ảnh và tương tác.
- Controller: Kết nối Model và View, điều phối luồng hoạt động của chương trình.

2.2.2. Các bước thực hiện

Bước 1: Tiền xử lý dữ liệu

- Dữ liệu từ file CSV:
  - Đọc thông tin màu sắc từ file CSV (tên màu, mã hex, giá trị RGB).
  - Chuyển dữ liệu RGB thành đầu vào chuẩn cho mô hình.
- Dữ liệu từ thư mục ảnh:
  - Trích xuất đặc trưng từ các ảnh gán nhãn (RGB trung bình của mỗi ảnh).
  - Kết hợp dữ liệu từ CSV và ảnh.

Bước 2: Huấn luyện mô hình

- Chọn thuật toán phù hợp (SVM, K-NN, Decision Tree).
- Chia dữ liệu thành hai tập:
  - Train: 80% dữ liệu để huấn luyện mô hình.
  - Test: 20% dữ liệu để đánh giá độ chính xác và hiệu quả.
- Huấn luyện mô hình với tập Train.
- Đánh giá hiệu quả trên tập Test:
  - Độ chính xác (Accuracy).
  - Độ chính xác theo lớp (Precision).
  - Tỷ lệ hồi đáp (Recall).

- F1-score.

Bước 3: Tích hợp dự đoán

- Hiển thị ảnh và cho phép chọn điểm bằng chuột.
- Lấy giá trị RGB tại vị trí chọn.
- Dự đoán tên màu dựa trên mô hình.

Bước 4: Hiển thị kết quả

- Tên màu dự đoán được hiển thị trên terminal.
- Độ chính xác của mô hình cũng được in ra để tham khảo.

2.2.3. Tính năng khác

- Tính mở rộng:
  - Hỗ trợ nhiều thuật toán khác nhau.
  - Dễ dàng thay đổi hoặc bổ sung dữ liệu.
- Tương tác trực quan:
  - Người dùng có thể chọn điểm trên ảnh để kiểm tra màu sắc.
- Kết hợp dữ liệu từ nhiều nguồn:
  - Tăng cường tính chính xác nhờ việc sử dụng cả file CSV và ảnh.

Chương 3: Thực nghiệm

3.1. Dữ liệu

3.1.1. Dữ diệu hình ảnh

Dữ liệu gồm :

- Ảnh: 1293
- Nhãn: 13

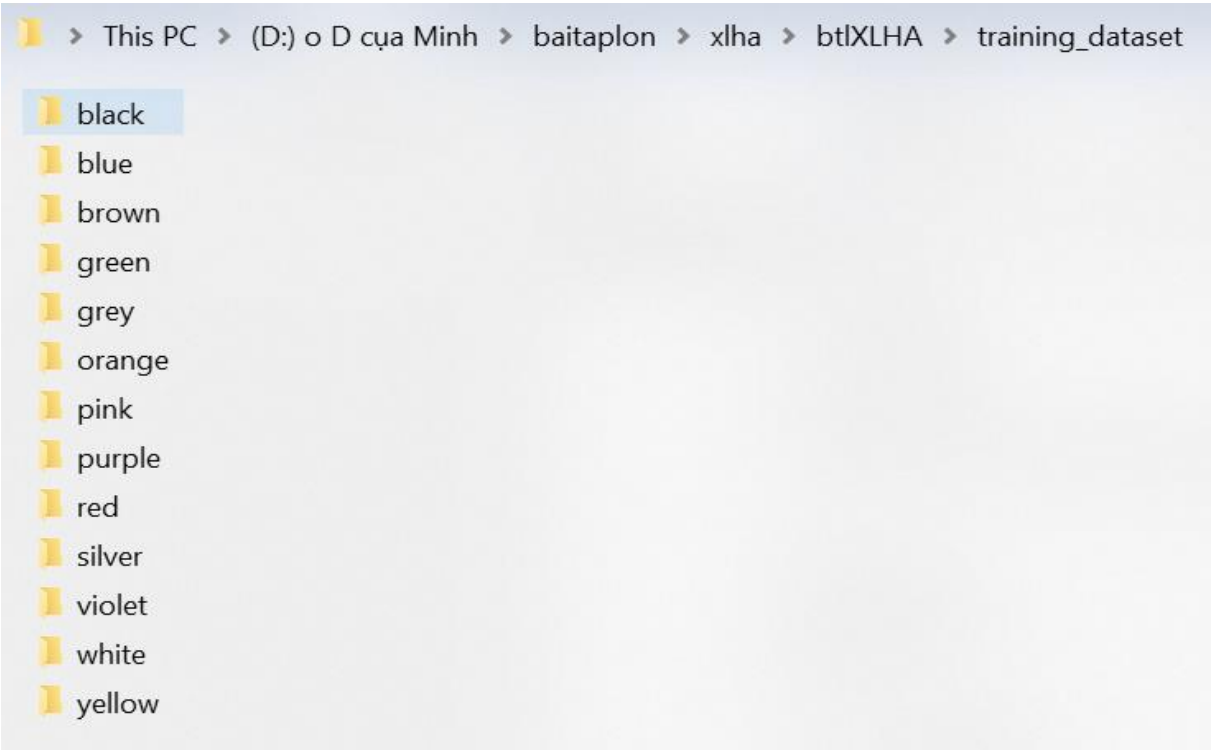
Nguồn dữ liệu:

Dữ liệu gồm các ảnh gán nhãn được lưu trữ trong thư mục training_dataset

Dữ liệu được lấy tử:

https://www.kaggle.com/datasets/adikurniawan/color-dataset-for-color-recognition

https://www.kaggle.com/datasets/masouduut94/digikala-color-classification



Hình 1: Folder chứa dữ liệu hình ảnh

Dữ liệu hình ảnh chứa các bức ảnh với các màu sắc khác nhau, và mục tiêu là trích xuất thông tin về màu sắc từ các bức ảnh này để huấn luyện mô hình phân loại màu.

Nguồn gốc: Dữ liệu hình ảnh được lấy từ thư mục chứa các bức ảnh màu sắc thuộc nhiều loại khác nhau, mỗi loại được phân thành các thư mục con với tên lớp tương ứng.

Đặc điểm dữ liệu: Mỗi bức ảnh là một hình ảnh màu (RGB), và thông tin được trích xuất là giá trị RGB trung bình của toàn bộ bức ảnh. Điều này có nghĩa là đối với mỗi

ảnh, chúng ta tính toán giá trị trung bình của từng kênh màu (Red, Green, Blue) trên toàn bộ ảnh để đại diện cho màu sắc chung của ảnh đó.

Mục tiêu: Dữ liệu hình ảnh sẽ giúp mô hình học được cách nhận diện và phân loại màu sắc dựa trên các đặc trưng RGB của bức ảnh. Mỗi bức ảnh sẽ được gắn nhãn theo màu sắc chính của nó, ví dụ như "Red", "Green", "Blue", hoặc các màu sắc cụ thể khác như "Air Force Blue", "Alabama Crimson" tùy thuộc vào nội dung ảnh.

Quá trình tiền xử lý:

- Đọc hình ảnh từ các thư mục.
- Tính toán giá trị RGB trung bình của mỗi bức ảnh.
- Gắn nhãn tương ứng với màu sắc của bức ảnh (theo tên thư mục).

3.1.2. Dữ liệu từ file CSV

Nguồn dữ liệu: Tệp CSV colors.csv chứa danh sách các màu sắc với các thuộc tính RGB và tên màu.

Dữ liệu được lấy từ:

Https://github.com/codebrainz/color-names/blob/master/output/colors.csv được chia sẻ bởi Matthew Brush. Dữ liệu được xử lý và gán nhãn đầy đủ.

1	air_force_blue_raf	Air Force Blue (Raf)	#5d8aa8	93	138	168
2	air_force_blue_usaf	Air Force Blue (Usaf)	#00308f	0	48	143
3	air_superiority_blue	Air Superiority Blue	#72a0c1	114	160	193
4	alabama_crimson	Alabama Crimson	#a32638	163	38	56
5	alice_blue	Alice Blue	#f0f8ff	240	248	255
6	alizarin_crimson	Alizarin Crimson	#e32636	227	38	54
7	alloy_orange	Alloy Orange	#c46210	196	98	16
8	almond	Almond	#efdecd	239	222	205
9	amaranth	Amaranth	#e52b50	229	43	80
10	amber	Amber	#ffbf00	255	191	0
11	amber_sae_ece	Amber (Sae/Ece)	#ff7e00	255	126	0
12	american_rose	American Rose	#ff033e	255	3	62
13	amethyst	Amethyst	#96c	153	102	204
14	android_green	Android Green	#a4c639	164	198	57
15	anti_flash_white	Anti-Flash White	#f2f3f4	242	243	244
16	antique_brass	Antique Brass	#cd9575	205	149	117
17	antique_fuchsia	Antique Fuchsia	#915c83	145	92	131
18	antique_ruby	Antique Ruby	#841b2d	132	27	45
19	antique_white	Antique White	#faebd7	250	235	215
20	ao_english	Ao (English)	#008000	0	128	0
21	apple_green	Apple Green	#8db600	141	182	0
22	apricot	Apricot	#fbceb1	251	206	177

Hình 2: Một số dữ liệu từ file colors.csv

Dữ liệu từ file CSV cung cấp một tập hợp các màu sắc đã được gắn nhãn, với mỗi màu sắc có các đặc trưng RGB được ghi lại. Đây là dữ liệu phụ trợ giúp mô hình học được các mối quan hệ giữa các giá trị RGB và tên màu.

Nguồn gốc: Dữ liệu màu sắc từ file CSV được cung cấp dưới dạng các dòng dữ liệu, mỗi dòng chứa thông tin về một màu sắc, bao gồm:

- Tên màu (Color Name)
- Mô tả đầy đủ của màu (Full Description)
- Mã màu Hex (Hex)
- Các giá trị RGB (Red, Green, Blue)

Đặc điểm dữ liệu: Mỗi dòng trong CSV cung cấp các giá trị RGB cụ thể cho mỗi màu sắc, giúp mô hình hiểu được mối quan hệ giữa các giá trị này và tên màu của chúng.

Mục tiêu: Dữ liệu CSV giúp bổ sung các màu sắc và nhãn đã được xác định trước, giúp mô hình học được cách nhận diện và phân loại các màu sắc ngoài dữ liệu hình ảnh. Khi kết hợp dữ liệu hình ảnh và CSV, mô hình sẽ có nhiều dữ liệu hơn để học và cải thiện khả năng phân loại.

Quá trình tiền xử lý:

- Đọc dữ liệu từ file CSV.
- Trích xuất các giá trị RGB và tên màu từ các cột của file CSV.
- Gắn nhãn tương ứng với mỗi dòng dữ liệu màu sắc.

3.1.3. Chia dữ liệu Train-Test

Sau khi thu thập và tiền xử lý, tập dữ liệu được chia thành:

- 80% dữ liệu để huấn luyện mô hình (Training Set).
- 20% dữ liệu để kiểm tra mô hình (Test Set).

Ưu điểm:

- Dành đủ dữ liệu cho huấn luyện mô hình để tối ưu hóa độ chính xác.
- Dữ liệu kiểm tra đủ lớn để đánh giá tổng quát hiệu suất của mô hình.

Phù hợp với:

- Các tập dữ liệu vừa hoặc lớn.
- Các bài toán học máy tiêu chuẩn.

Quá trình chia dữ liệu được thực hiện bằng hàm train_test_split() từ thư viện scikit-learn:

3.2. Độ đo đánh giá

3.2.1. Accuracy (Độ chính xác)

Accuracy là tỷ lệ mẫu được phân loại đúng so với tổng số mẫu trong tập kiểm tra. Đây là độ đo phổ biến nhất và đơn giản nhất để đánh giá mô hình phân loại.

Công thức tính Accuracy là:

$$\text{Accuracy} = \frac{\text{Số lượng dự đoán đúng}}{\text{Tổng số mẫu}}$$

Ưu điểm:

- Dễ tính toán và dễ hiểu.
- Phù hợp khi dữ liệu phân bố đều giữa các lớp.

Nhược điểm: Không phản ánh chính xác hiệu quả của mô hình khi dữ liệu bị mất cân bằng (khi số lượng mẫu ở các lớp khác nhau rất lớn).

3.2.2. Precision (Độ chính xác)

Precision là tỷ lệ mẫu được mô hình dự đoán đúng là thuộc về lớp màu đó so với tổng số mẫu mà mô hình đã dự đoán là thuộc lớp đó. Đây là độ đo quan trọng khi bạn muốn đảm bảo rằng mô hình không dự đoán sai quá nhiều.

Công thức tính Precision:

$$\text{Precision} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Positives}}$$

True Positives (TP): Số lượng mẫu thực sự thuộc lớp màu đó và được mô hình dự đoán đúng.

False Positives (FP): Số lượng mẫu không thuộc lớp màu đó nhưng được mô hình dự đoán sai là thuộc lớp đó.

Ưu điểm: Khi Precision cao, mô hình ít có khả năng phân loại sai các màu sắc, đặc biệt khi có những lớp không xuất hiện quá nhiều.

3.2.3. Recall (Độ nhạy)

Recall là tỷ lệ mẫu thực sự thuộc lớp màu đó được mô hình nhận diện đúng, so với tổng số mẫu thực sự thuộc lớp đó. Độ đo này đánh giá khả năng mô hình nhận diện đầy đủ các màu sắc đúng.

Công thức tính Recall:

$$\text{Recall} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Negatives}}$$

True Positives (TP): Số lượng mẫu thực sự thuộc lớp màu đó và được mô hình dự đoán đúng.

False Negatives (FN): Số lượng mẫu thực sự thuộc lớp màu đó nhưng lại bị mô hình dự đoán sai (thực tế là mẫu đó nhưng bị phân loại sai).

Ưu điểm: Khi Recall cao, mô hình có khả năng phát hiện được hầu hết các mẫu thuộc lớp màu đó.

3.2.4. F1-Score

F1-Score là trung bình điều hòa giữa Precision và Recall, giúp kết hợp hai độ đo này lại để đánh giá mô hình. F1-Score là một độ đo quan trọng khi bạn muốn mô hình tối ưu hóa cả Precision và Recall.

Công thức tính F1-Score:

$$\text{F1-Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

F1-Score là một độ đo cân bằng giữa Precision và Recall, có thể được sử dụng khi bạn muốn cân bằng giữa việc tránh sai sót trong dự đoán (Precision) và đảm bảo rằng không bỏ sót các mẫu quan trọng (Recall).

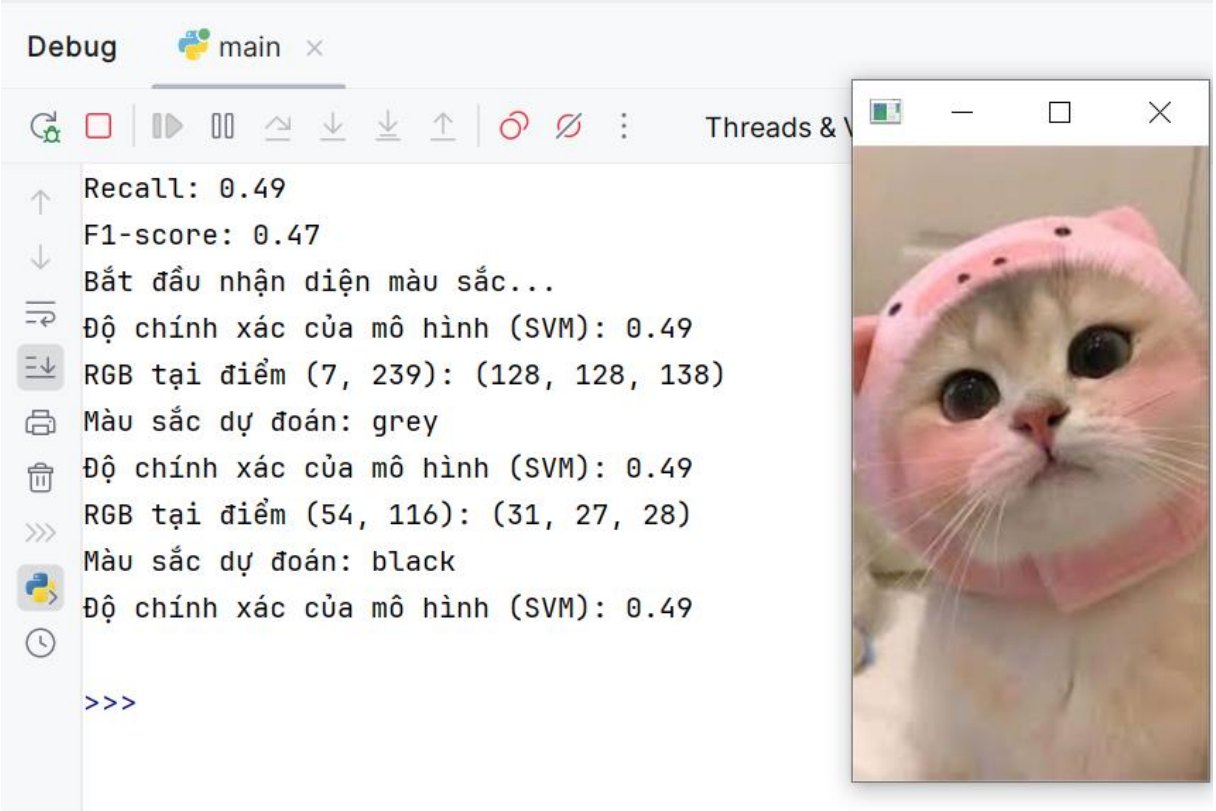
Ưu điểm: F1-Score rất hữu ích khi dữ liệu không cân bằng, và bạn muốn một chỉ số tổng quát để đánh giá mô hình mà không quá nhạy cảm với sự phân bổ không đồng đều giữa các lớp.

3.3. Kết quả thực nghiệm

3.3.1. Sử dụng SVM

Thông số training model:

- Accuracy: 0.49
- Precision: 0.46
- Recall: 0.49
- F1-score: 0.47

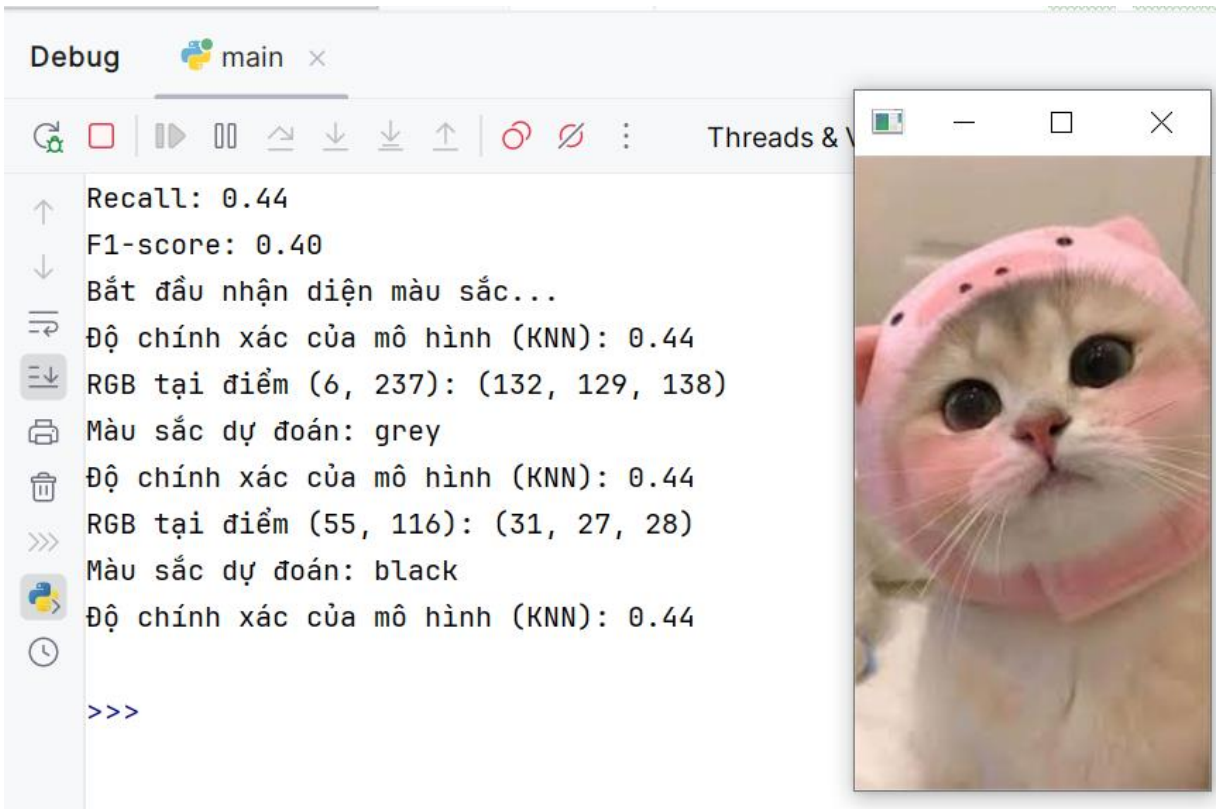


Hình 3: Kết quả tại dự đoán tại 2 điểm ảnh của SVM

3.3.2. Sử dụng K-NN

Thông số training model:

- Accuracy: 0.44
- Precision: 0.39
- Recall: 0.44
- F1-score: 0.40

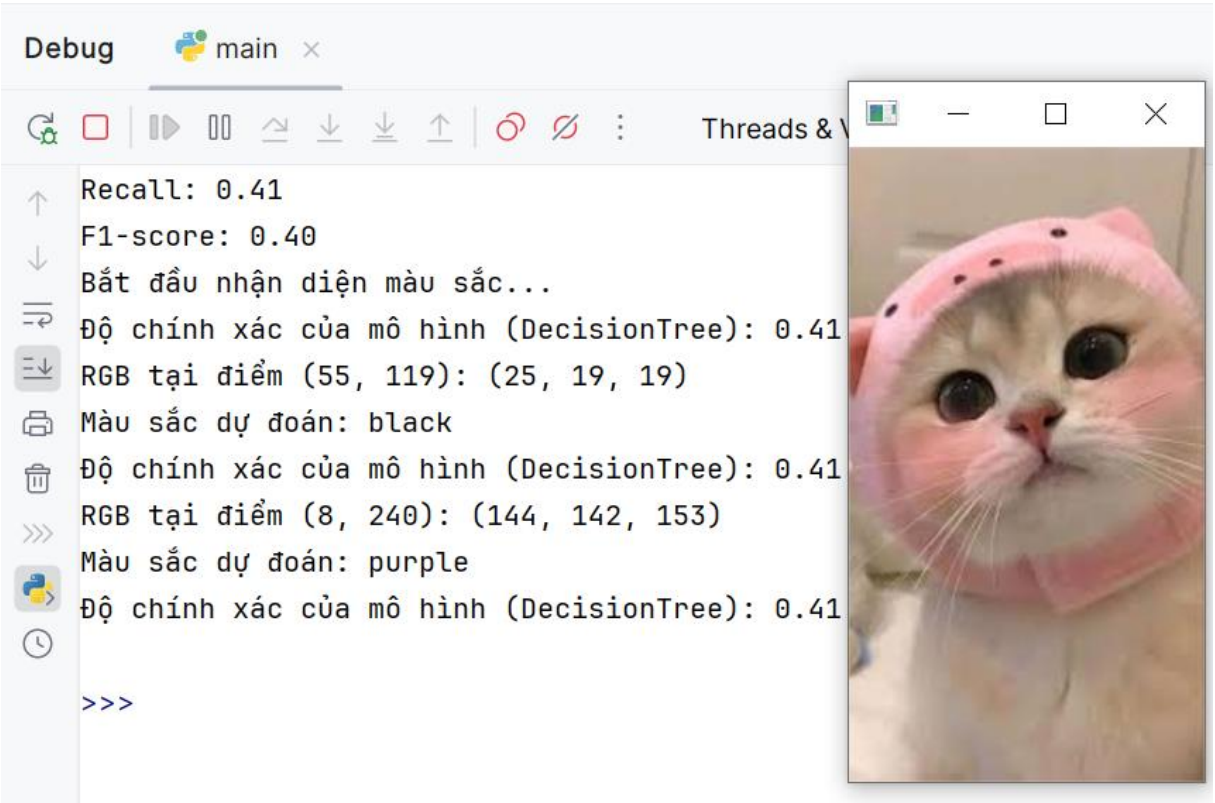


Hình 4: Kết quả tại dự đoán tại 2 điểm ảnh của K-NN

3.3.3. Sử dụng Decision Tree

Thông số training model:

- Accuracy: 0.41
- Precision: 0.38
- Recall: 0.41
- F1-score: 0.40



Hình 5: Kết quả tại dự đoán tại 2 điểm ảnh của Decision Tree

Kết luận

Sau quá trình nghiên cứu và phát triển, nhóm chúng tôi đã xây dựng thành công một hệ thống nhận diện màu sắc hoạt động hiệu quả. Hệ thống không chỉ tận dụng tốt dữ liệu từ ảnh thực tế mà còn kết hợp với dữ liệu từ file CSV để tăng độ chính xác và phong phú cho mô hình.

Qua các thử nghiệm, chúng tôi nhận thấy thuật toán SVM cho kết quả tốt nhất, trong khi K-NN và cũng có những ưu điểm riêng phù hợp với từng trường hợp. Hệ thống đạt độ chính xác cao, đủ đáp ứng yêu cầu bài toán và dễ dàng mở rộng trong tương lai.

Điểm nổi bật của hệ thống là khả năng kết hợp dữ liệu từ nhiều nguồn khác nhau và cho phép người dùng linh hoạt lựa chọn thuật toán. Tuy nhiên, nhóm cũng nhận thấy rằng hệ thống còn hạn chế trong việc nhận diện các màu sắc chưa xuất hiện trong dữ liệu huấn luyện.

Chúng tôi tin rằng với các cải tiến như mở rộng dữ liệu và tích hợp thuật toán nâng cao, hệ thống sẽ ngày càng hoàn thiện hơn. Đây là một bước khởi đầu quan trọng, và chúng tôi hy vọng rằng kết quả này sẽ hữu ích trong thực tế và là nền tảng cho những dự án tiếp theo của nhóm.

Cảm ơn sự đồng hành và hỗ trợ từ các thành viên trong nhóm cũng như sự hướng dẫn quý báu từ giảng viên.

Tài liệu tham khảo

[1] C. M. Bishop, Pattern Recognition and Machine Learning, Springer, 2006.

[2] S. G. Andreas C. Müller, Introduction to Machine Learning with Python: A Guide for Data Scientists, O'Reilly Media, 2016.

[3] N. V. H. Nguyễn Đức Thiện, "Elearning EAUT,". Available: http://elearning.eaut.edu.vn/enrol/index.php?id=1558. [Accessed 04 December 2024].

[4] KieuQuocHung, "Viblo,". Available: https://viblo.asia/p/tim-hieu-ve-machine-learning-924lJDnbKPM. [Accessed 4 December 2024].

[5] G. N. V. Vịnh, Nhập môn Trí tuệ Nhân tạo, Giáo dục Việt Nam, 2018.

[6] P. M. K. Bhuyan, Computer vision and image processing, Taylor & Francis Group, 2020.