**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** | 20211716 | Nguyễn Văn Quyết | DCCNTT 12.10.6 |
| **2** | 20211728 | Nguyễn Quốc Khánh | DCCNTT 12.10.6 |
| **3** | 20211382 | Nguyễn Tiến Việt | DCCNTT 12.10.6 |
| **4** | 20211643 | Trương Văn Phương Quyền | DCCNTT 12.10.6 |

**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**

|  |  |  |  |
| --- | --- | --- | --- |
| **TT** | **Mã sinh viên** | **Sinh viên thực hiện** | **Lớp hành chính** |
| **1** | 20211716 | Nguyễn Văn Quyết | DCCNTT 12.10.6 |
| **2** | 20211728 | Nguyễn Quốc Khánh | DCCNTT 12.10.6 |
| **3** | 20211382 | Nguyễn Tiến Việt | DCCNTT 12.10.6 |
| **4** | 20211643 | Trương Văn Phương Quyền | DCCNTT 12.10.6 |

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

**Bắc Ninh, năm 2024**

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

|  |  |
| --- | --- |
| **PHIẾU CHẤM THI BÀI TẬP LỚN KẾT THÚC HỌC PHẦN**  **Mã đề thi: 02**  **Tên học phần: XỬ LÝ ẢNH VÀ THỊ GIÁC MÁY TÍNH**  **Lớp Tín Chỉ:**  **XATGMT.03.K12.06.LH.C04.1\_LT.1\_TH** | |
| **Cán bộ chấm thi 1**  *(Ký và ghi rõ họ tên)* | **Cán bộ chấm thi 2**  *(Ký và ghi rõ họ tên)* |

| **TT** | **TIÊU CHÍ** | **THANG ĐIỂM** | **Nguyễn Văn Quyết** | **Nguyễn Quốc Khánh** | **Nguyễn Tiến Việt** | **Trương Văn Phương Quyền** |
| --- | --- | --- | --- | --- | --- | --- |
| 20211716 | 20211728 | 20211382 | 20211643 |
| **1** | **Nội dung báo cáo trên Word đầy đủ** | **3.5** |  |  |  |  |
| 1.1 | Có bố cục rõ ràng (mục lục, phần mở đầu, nội dung chính, kết luận). | 0,5 |  |  |  |  |
| 1.2 | Nội dung phân tích rõ ràng, logic. | 0,5 |  |  |  |  |
| 1.3 | Có dẫn chứng, số liệu minh họa đầy đủ. | 0,5 |  |  |  |  |
| 1.4 | Ngôn ngữ và trình bày chuẩn, không lỗi chính tả. | 0,5 |  |  |  |  |
| 1.5 | Có trích dẫn tài liệu tham khảo đúng quy cách. | 0,5 |  |  |  |  |
| 1.6 | Được trình bày chuyên nghiệp (canh lề, font chữ, khoảng cách dòng hợp lý). | 0,5 |  |  |  |  |
| 1.7 | Tài liệu đầy đủ, bám sát yêu cầu của đề bài. | 0,5 |  |  |  |  |
| **2** | **Nội dung thuyết trình đầy đủ** | **1.0** |  |  |  |  |
| 2.1 | Trình bày tự tin, phát âm rõ ràng, mạch lạc. | 0,5 |  |  |  |  |
| 2.2 | Nội dung thuyết trình đúng trọng tâm, không lan man. | 0,5 |  |  |  |  |
| **3** | **Slides báo cáo đầy đủ nội dung + Hỏi đáp** | **3.0** |  |  |  |  |
| 3.1 | Slides có bố cục rõ ràng (mở đầu, nội dung, kết luận). | 0,5 |  |  |  |  |
| 3.2 | Thiết kế slides đẹp, chuyên nghiệp (màu sắc, hình ảnh minh họa). | 0,5 |  |  |  |  |
| 3.3 | Nội dung trên slides ngắn gọn, dễ hiểu, súc tích. | 0,5 |  |  |  |  |
| 3.4 | Nội dung slides phù hợp với nội dung báo cáo. | 0,5 |  |  |  |  |
| 3.5 | Trả lời câu hỏi đầy đủ, chính xác. | 0,5 |  |  |  |  |
| 3.6 | Trả lời câu hỏi tự tin, thuyết phục. | 0,5 |  |  |  |  |
| **4** | **Code đầy đủ** | **2.5** |  |  |  |  |
| 1.1 | Code được trình bày rõ ràng, có chú thích đầy đủ. | 0,5 |  |  |  |  |
| 1.2 | Code chạy đúng, không lỗi. | 0,5 |  |  |  |  |
| 1.3 | Code tối ưu, không dư thừa. | 0,5 |  |  |  |  |
| 1.4 | Đáp ứng đầy đủ các yêu cầu chức năng theo đề bài. | 0,5 |  |  |  |  |
| 1.5 | Có tính sáng tạo hoặc cải thiện so với yêu cầu. | 0,5 |  |  |  |  |
| **TỔNG ĐIỂM BẰNG SỐ:** | | **10** |  |  |  |  |
| **TỔNG ĐIỂM BẰNG CHỮ:** | | *Mười tròn* |  |  |  |  |

**MỤC LỤC**

[LỜI MỞ ĐẦU 1](#_Toc184597846)

[CHƯƠNG 1: CƠ SỞ LÝ THUYẾT 2](#_Toc184597847)

[1.1. Biểu diễn ảnh trong máy tính 2](#_Toc184597848)

[1.1.1 Thu nhận ảnh 2](#_Toc184597849)

[1.1.2 Các dải màu trong máy tính 4](#_Toc184597850)

[1.2 Tổng quan về học máy 10](#_Toc184597851)

[1.2.1 Học có giám sát 10](#_Toc184597852)

[1.2.2 Học không giám sát 12](#_Toc184597853)

[1.2.3 Học bán giám sát 16](#_Toc184597854)

[CHƯƠNG 2: XÂY DỰNG HỆ THỐNG NHẬN DIỆN MÀU SẮC BẰNG PANDAS VÀ OPENCV 19](#_Toc184597855)

[2.1. Thư viện 19](#_Toc184597856)

[2.1.1. CV2 (OpenCV) 19](#_Toc184597857)

[2.1.2. Pandas 23](#_Toc184597858)

[2.1.3. NumPy 29](#_Toc184597859)

[2.1.4 Tkinter 31](#_Toc184597860)

[2.2 Xây dựng hệ thống nhận diện màu sắc trong ảnh 33](#_Toc184597861)

[2.2.1 Các bước thực hiện: 33](#_Toc184597862)

[2.2.2. Một số mã nguồn quan trọng 36](#_Toc184597863)

[CHƯƠNG 3: THỰC NGHIỆM 37](#_Toc184597864)

[3.1 Dữ liệu 37](#_Toc184597865)

[3.1.1 Dữ liệu hình ảnh 37](#_Toc184597866)

[3.1.2 Dữ liệu từ file CSV 38](#_Toc184597867)

[3.2 Độ đo đánh giá 39](#_Toc184597868)

[3.2.1 Accuracy (Độ chính xác) 39](#_Toc184597869)

[3.2.2 Precision (Độ chính xác theo từng cụm) 39](#_Toc184597870)

[3.2.3 Recall (Độ nhạy) 39](#_Toc184597871)

[3.2.4 F1-Score 40](#_Toc184597872)

[3.3 Kết quả thực nghiệm 41](#_Toc184597873)

[3.3.1 Sử dụng SVM 41](#_Toc184597874)

[3.3.2 Sử dụng KNN 41](#_Toc184597875)

[3.3.3 Sử dụng Decision Tree 42](#_Toc184597876)

[KẾT LUẬN 43](#_Toc184597877)

[TÀI LIỆU THAM KHẢO 45](#_Toc184597878)

# LỜI MỞ ĐẦU

Trong thời đại công nghệ 4.0, việc ứng dụng các giải pháp kỹ thuật số để giải quyết bài toán thực tiễn đang trở nên phổ biến và cần thiết hơn bao giờ hết. Một trong những lĩnh vực nổi bật là xử lý ảnh và thị giác máy tính, nơi OpenCV và Pandas đóng vai trò quan trọng trong việc xây dựng các hệ thống thông minh.

Dự án **"Xây dựng hệ thống nhận diện màu sắc bằng Pandas và OpenCV"** không chỉ giới hạn trong việc nhận diện màu sắc cơ bản, mà còn mở rộng để phân tích và xử lý dữ liệu màu sắc phức tạp. Sự kết hợp giữa OpenCV, một thư viện mạnh mẽ trong xử lý hình ảnh, và Pandas, công cụ đắc lực để quản lý và phân tích dữ liệu, tạo ra một giải pháp toàn diện cho các ứng dụng từ công nghiệp đến đời sống hàng ngày.

Hệ thống tập trung vào việc trích xuất dữ liệu màu sắc từ hình ảnh, hiển thị và phân loại màu dựa trên các mô hình màu như RGB và HSV. Đồng thời, hệ thống còn quản lý và phân tích thông tin màu sắc thông qua Pandas, giúp hệ thống trở nên trực quan và dễ sử dụng. Với những ưu điểm về tốc độ, độ chính xác và khả năng mở rộng, dự án này không chỉ có giá trị về mặt kỹ thuật mà còn mở ra nhiều hướng phát triển ứng dụng trong thực tế.

# CHƯƠNG 1: CƠ SỞ LÝ THUYẾT

## 1.1. Biểu diễn ảnh trong máy tính

### 1.1.1 Thu nhận ảnh

Thu nhận ảnh trong máy tính là quá trình chuyển đổi hình ảnh từ thế giới thực (ảnh vật lý) thành dạng số hóa để máy tính có thể lưu trữ, xử lý và phân tích. Quá trình này đóng vai trò nền tảng trong lĩnh vực xử lý ảnh và thị giác máy tính, giúp các hệ thống tự động "nhìn thấy" và hiểu được hình ảnh.

Quá trình thu nhận ảnh bao gồm các bước chính sau:

* **Chụp ảnh bằng cảm biến:** Sử dụng các thiết bị như camera, máy quét hoặc cảm biến để thu thập ánh sáng từ môi trường. Cảm biến sẽ chuyển đổi ánh sáng này thành tín hiệu điện.
* **Chuyển đổi tín hiệu điện thành tín hiệu số:** Bộ chuyển đổi tương tự - số (ADC) sẽ biến đổi tín hiệu điện thu được từ cảm biến thành dữ liệu số.
* **Lưu trữ dữ liệu ảnh**: Sau khi số hóa, hình ảnh được lưu trữ dưới dạng ma trận các điểm ảnh (pixel), với mỗi điểm ảnh được biểu diễn bằng các giá trị số đại diện cho cường độ sáng hoặc màu sắc (ví dụ: RGB).

Kết quả của quá trình thu nhận ảnh là một tệp số hóa có thể được xử lý bởi các thuật toán máy tính, phục vụ cho các ứng dụng như nhận diện khuôn mặt, phân loại đối tượng, hoặc phân tích hình ảnh trong y tế.

Trong thực tế, việc thu nhận ảnh đòi hỏi chất lượng cảm biến cao và các phương pháp xử lý tín hiệu tối ưu để đảm bảo độ chính xác và chi tiết của hình ảnh số hóa.

Thu nhận ảnh trong máy tính là quá trình chuyển đổi các hình ảnh từ môi trường thực tế thành dữ liệu số để máy tính có thể lưu trữ, xử lý và phân tích. Đây là bước đầu tiên và quan trọng trong lĩnh vực xử lý ảnh và thị giác máy tính, đặt nền móng cho việc phân tích.

**Quy trình thu nhận ảnh**

**Chụp ảnh bằng thiết bị cảm biến:**

* Thiết bị cảm biến như máy ảnh, máy quét, hoặc các camera chuyên dụng sẽ thu thập ánh sáng hoặc các dạng sóng điện từ từ môi trường.
* Ánh sáng hoặc tín hiệu sẽ được ghi nhận bởi các phần tử cảm biến như CCD (Charge-Coupled Device) hoặc CMOS (Complementary Metal-Oxide-Semiconductor).
* Cảm biến phân tích độ sáng (luminance) và thông tin màu sắc (chrominance) của mỗi điểm ảnh (pixel).

**Chuyển đổi tín hiệu ánh sáng thành tín hiệu điện:**

* Khi ánh sáng đến bề mặt cảm biến, năng lượng ánh sáng được chuyển đổi thành tín hiệu điện tương ứng với cường độ ánh sáng.
* Các cảm biến thường sử dụng lưới điểm ảnh để chia hình ảnh thành các phần tử nhỏ (pixel), mỗi phần tử sẽ thu nhận một mức tín hiệu.

**Số hóa tín hiệu điện:**

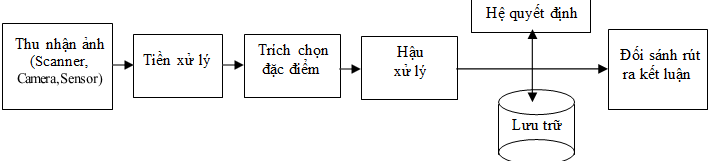
* Bộ chuyển đổi tương tự - số (ADC - Analog-to-Digital Converter) sẽ chuyển đổi tín hiệu điện tương tự từ cảm biến thành tín hiệu số.
* Kết quả là một ma trận số biểu diễn hình ảnh, với mỗi phần tử tương ứng với một pixel và chứa thông tin về độ sáng hoặc màu sắc.

**Xử lý tiền xử lý:**

Hình ảnh thu nhận thường cần qua các bước tiền xử lý để cải thiện chất lượng như: giảm nhiễu, cân bằng sáng, hoặc chỉnh sửa biến dạng

**Lưu trữ hình ảnh:**

* Dữ liệu hình ảnh số hóa được lưu trữ trong các định dạng file phổ biến như JPEG, PNG, BMP, TIFF hoặc dưới dạng ma trận số trong bộ nhớ.
* Mỗi định dạng lưu trữ có các ưu và nhược điểm riêng, phù hợp với mục đích sử dụng khác nhau (nén dữ liệu, chất lượng hình ảnh cao, hoặc lưu trữ lâu dài).



Hình 1.1. Các bước cơ bản trong quá trình xử lý ảnh [1]

**Các tham số quan trọng trong quá trình thu nhận ảnh**

**Độ phân giải (Resolution):**

Biểu thị số lượng pixel trên một hình ảnh, quyết định mức độ chi tiết của hình ảnh. Độ phân giải càng cao thì hình ảnh càng sắc nét.

**Độ sâu màu (Color Depth):**

Quy định số bit dùng để biểu diễn màu của mỗi pixel. Ví dụ, hình ảnh 24-bit RGB có thể biểu diễn hơn 16 triệu màu.

**Tỷ lệ khung hình (Aspect Ratio):**

Là tỷ lệ giữa chiều rộng và chiều cao của hình ảnh.

**Tốc độ khung hình (Frame Rate):**

Trong các ứng dụng video, tốc độ khung hình quyết định số lượng hình ảnh được thu nhận mỗi giây (fps - frame per second).

**Dải động (Dynamic Range):**

Là khả năng cảm biến ghi nhận sự khác biệt giữa các mức sáng tối trong một hình ảnh.

### 1.1.2 Các dải màu trong máy tính

Dải màu trong máy tính là các mô hình toán học hoặc không gian màu được sử dụng để biểu diễn và định nghĩa các màu sắc khác nhau trong hình ảnh và đồ họa. Máy tính dựa vào các hệ màu này để xử lý, hiển thị và phân tích thông tin màu sắc một cách chính xác và hiệu quả.

***Các hệ màu phổ biến trong máy tính***

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

**Định nghĩa:** RGB là một hệ màu cộng (additive color model) được tạo ra bằng cách kết hợp ba màu cơ bản: đỏ (Red), xanh lá cây (Green), và xanh dương (Blue).



Hình 1.2. RGB COLOR MODEL [2]

**Cách hoạt động:** Các màu khác nhau được tạo ra bằng cách điều chỉnh cường độ của ba màu cơ bản theo tỷ lệ từ 0 đến 255 (trong không gian 8-bit).

**Ứng dụng:**

* Hiển thị màu trên các thiết bị điện tử như màn hình máy tính, TV, và điện thoại.
* Phổ biến trong thiết kế đồ họa và xử lý hình ảnh kỹ thuật số.

**Ưu điểm:** Phù hợp với các thiết bị phát sáng (emissive devices).

**Hạn chế:** Không thể biểu diễn chính xác các màu trong môi trường in ấn

**CMY/CMYK (Cyan, Magenta, Yellow, Black):**

**Định nghĩa:** CMY là hệ màu trừ (subtractive color model), thường được mở rộng thành CMYK bằng cách thêm màu đen (Black) để tăng độ sâu màu sắc.

A group of multicolored circles

Description automatically generated

Hình 1.3. CMYK COLOR MODEL [2]

**Cách hoạt động:** Các màu sắc được tạo ra bằng cách trừ ánh sáng từ một nguồn sáng trắng thông qua các bộ lọc màu lục lam (Cyan), đỏ tươi (Magenta), và vàng (Yellow).

**Ứng dụng:**

* Hệ màu chuẩn trong in ấn và xuất bản.
* Dùng để kiểm soát mực in trên giấy.

**Ưu điểm:** Tạo ra màu sắc chính xác trong môi trường in ấn.

**Hạn chế:** Khó biểu diễn các màu phát sáng và ít phù hợp với thiết bị màn hình.

**HSV (Hue, Saturation, Value):**

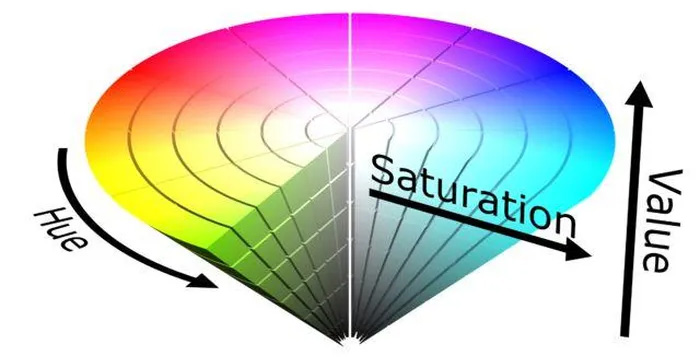
**Định nghĩa:** HSV là một hệ màu dựa trên cách con người nhận thức màu sắc, được định nghĩa bởi ba thành phần:

**Hue (Sắc thái):** Biểu diễn màu cơ bản (đỏ, vàng, xanh, v.v.).

**Saturation (Độ bão hòa):** Biểu diễn độ tinh khiết của màu (từ nhạt đến đậm).

**Value (Giá trị):** Biểu diễn độ sáng của màu.

**HSV được nhiều người sử dụng trong quá trình in ấn**



Hình 1.4.Biểu đồ màu HSV [3]

**Ứng dụng:**

Xử lý ảnh và đồ họa để dễ dàng thao tác trên từng thành phần màu.

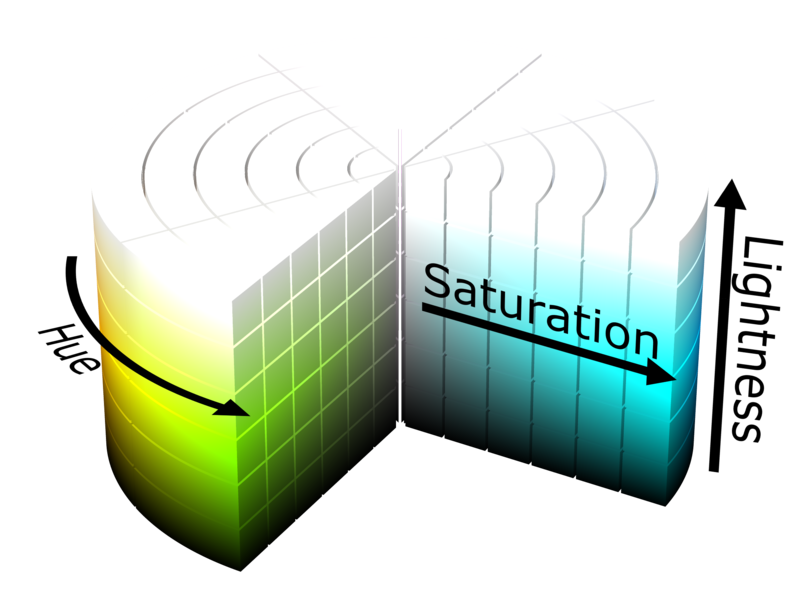
Tích hợp trong các công cụ chỉnh sửa màu sắc, như Photoshop hoặc GIMP.

**Ưu điểm:** Phù hợp với các ứng dụng chỉnh sửa và phân tích màu sắc.

**Hạn chế:** Phức tạp hơn so với RGB trong việc biểu diễn trên máy tính.

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

**Định nghĩa:** Tương tự HSV, HSL biểu diễn màu dựa trên cách con người cảm nhận, nhưng thay giá trị sáng (Value) bằng độ sáng (Lightness).



Hình 1.5. Biểu đồ màu HSL [3]

**Cách hoạt động:**

**Hue:** Màu sắc cơ bản.

**Saturation:** Độ bão hòa.

**Lightness:** Độ sáng, biểu diễn từ đen (0%) đến trắng (100%).

**Ứng dụng:**

Chỉnh sửa màu sắc trong thiết kế đồ họa.

Phổ biến trong các trình chỉnh sửa ảnh và video.

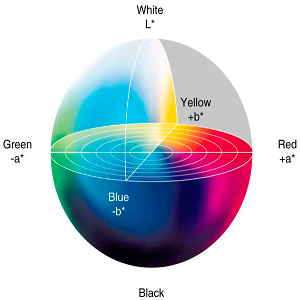
**LAB (CIELAB):**

**Định nghĩa:** LAB là một hệ màu dựa trên cách con người cảm nhận màu sắc, với ba thành phần:

**L:** Độ sáng (Lightness).

**A:** Trục màu từ xanh lá cây đến đỏ.

**B:** Trục màu từ xanh dương đến vàng.



Hình 1.6. Mô hình màu CIE Lab [3]

**Ứng dụng:**

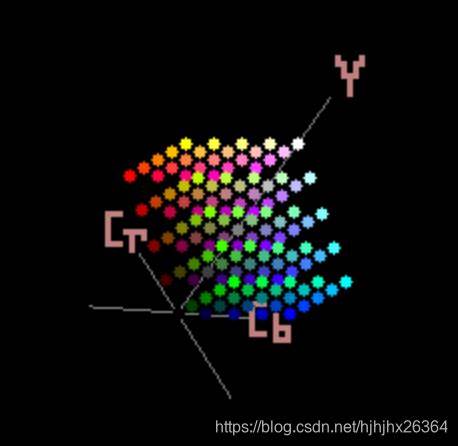
Dùng trong các hệ thống xử lý hình ảnh cao cấp.

Được sử dụng để chuyển đổi giữa các không gian màu khác nhau như RGB và CMYK.

**Ưu điểm:** Mô tả toàn bộ phổ màu mà mắt người có thể nhận biết.

**YCbCr:**

**Định nghĩa:** Hệ màu này được thiết kế đặc biệt cho xử lý video và truyền hình, với các thành phần:



Hình 1.7. **YcbCr [4]**

**Y:** Độ sáng (luminance).

**U và V (hoặc Cb và Cr):** Các thành phần màu sắc (chrominance).

**Ứng dụng:**

Chuẩn video và truyền hình, như MPEG, JPEG.

Tối ưu hóa băng thông bằng cách giảm độ phân giải màu sắc mà không ảnh hưởng nhiều đến chất lượng hình ảnh.

**Ứng dụng của thu nhận ảnh trong máy tính**

**Thị giác máy tính (Computer Vision):**

Nhận diện khuôn mặt, đối tượng, biển số xe, và phân tích chuyển động.

**Y tế:**

Chụp X-quang, MRI, và phân tích hình ảnh y khoa để chẩn đoán bệnh.

**Robot và tự động hóa:**

Hệ thống điều hướng và nhận diện vật cản trong các robot tự hành.

**Giải trí:**

Chụp ảnh, quay video và xử lý hình ảnh trong ngành công nghiệp điện ảnh.

**Phân tích dữ liệu môi trường:**

Sử dụng cảm biến để thu nhận dữ liệu từ vệ tinh hoặc các thiết bị bay không người lái (drone).

**Nông nghiệp thông minh:**

Giám sát cây trồng thông qua hình ảnh chụp từ camera hoặc drone.

***Thách thức trong thu nhận ảnh***

**Nhiễu ảnh (Noise):** Gây ra bởi cảm biến hoặc môi trường thu nhận ảnh kém.

**Biến dạng hình ảnh:** Do ống kính hoặc sai lệch trong thiết bị thu nhận.

**Ánh sáng không đồng đều:** Làm giảm độ chính xác của quá trình phân tích.

## 1.2 Tổng quan về học máy

### 1.2.1 Học có giám sát

**1.2.1.1 KNN (K-nearest neighbor)**

K-nearest neighbor (KNN) là một trong những thuật toán supervised learning đơn giản. Khi huấn luyện, thuật toán này không học một điều gì từ dữ liệu huấn luyện mà nhớ lại một cách máy móc toàn bộ dữ liệu đó. Đây cũng là lý do thuật toán này được xếp vào loại lazy learning, mọi tính toán được thực hiện khi nó cần dự đoán đầu ra của dữ liệu mới. KNN có thể áp dụng được vào cả classification và regression. KNN còn được gọi là một thuật toán instance-based hay memory-based learning. Với KNN, trong bài toán classification, nhãn của một điểm dữ liệu mới được suy ra trực tiếp từ K điểm dữ liệu gần nhất trong tập huấn luyện. Nhãn đó có thể được quyết định bằng bầu chọn theo đa số (major voting) trong số K điểm gần nhất, hoặc nó có thể được suy ra bằng cách đánh trọng số khác nhau cho mỗi trong các điểm gần nhất đó rồi suy ra kết quả. Chi tiết sẽ được nêu trong phần tiếp theo. Trong bài toán regresssion, đầu ra của một điểm dữ liệu sẽ bằng chính đầu ra của điểm dữ liệu đã biết gần nhất (trong trường hợp K = 1), hoặc là trung bình có trọng số của đầu ra của những điểm gần nhất, hoặc bằng một mối quan hệ dựa trên các điểm gần nhất đó và khoảng cách tới chúng.

**Hình ảnh** dưới đây mô tả một bài toán phân lớp với ba class: đỏ, lam, lục. Các hình tròn nhỏ với màu khác nhau thể hiện dữ liệu huấn luyện của các class khác nhau. Các vùng màu nền khác nhau thể hiện lãnh thổ của mỗi class. Tại một điểm bất kỳ, class của nó được xác định dựa trên class của điểm gần nó nhất trong trong tập huấn luyện.

A colorful background with dots

Description automatically generated

Hình 1.8. Ví dụ về 1NN.[5]

Các hình tròn là các điểm dữ liệu huấn luyện. Các hình khác màu thể hiện các lớp khác nhau. Các vùng nền thể hiện các điểm được phân loại vào lớp có màu tương ứng khi sử dựng 1NN.

**Nguyên lý hoạt động của KNN**

Khi cần phân loại hoặc dự đoán cho một điểm dữ liệu mới, KNN thực hiện các bước sau:

**Tính khoảng cách:**

KNN tính toán khoảng cách giữa điểm dữ liệu mới và tất cả các điểm dữ liệu trong tập huấn luyện.

**Các phương pháp tính khoảng cách phổ biến:**

A black background with white text

Description automatically generated

Hình 1.9. Công thức tính khoảng cách

**Tìm k láng giềng gần nhất:**

Dựa trên khoảng cách vừa tính, KNN chọn ra k điểm dữ liệu gần nhất (hàng xóm gần nhất) của điểm dữ liệu mới.

**Dự đoán kết quả:**

Phân loại: Điểm dữ liệu mới được phân vào lớp mà phần lớn các hàng xóm thuộc về (dựa trên bỏ phiếu đa số).

Hồi quy: KNN trả về giá trị trung bình của các giá trị mục tiêu từ k hàng xóm gần nhất.

### 1.2.2 Học không giám sát

Học không giám sát (Unsupervised Learning) là một phương pháp học máy mà trong đó các thuật toán không được cung cấp các nhãn (label) hoặc thông tin đầu ra mục tiêu tương ứng với dữ liệu. Thay vào đó, thuật toán cố gắng tự khám phá các cấu trúc, mô hình hoặc mối quan hệ ẩn trong dữ liệu dựa trên những đặc điểm của dữ liệu đầu vào.

**Cách hoạt động của học không giám sát**

Trong học không giám sát, thuật toán nhận vào dữ liệu thô (unlabeled data) và phân tích nó để:

* Nhóm các dữ liệu có đặc điểm tương đồng vào cùng một cụm (clustering).
* Tìm kiếm và biểu diễn các mẫu hoặc quan hệ tiềm ẩn trong dữ liệu.
* Giảm kích thước dữ liệu mà vẫn giữ lại thông tin quan trọng.
* A diagram of a process

  Description automatically generated
* Hình 1.10. Quy trình học máy không giám sát [6]

**Các loại bài toán học không giám sát**

**Phân cụm (Clustering):**

Mục tiêu: Phân chia dữ liệu thành các cụm (nhóm) sao cho các điểm dữ liệu trong cùng một cụm tương tự nhau nhất.

Ứng dụng:

* Phân khúc khách hàng trong marketing.
* Phân nhóm gene trong sinh học.
* Phát hiện cộng đồng trong mạng xã hội.

**Thuật toán tiêu biểu:**

* K-Means.
* DBSCAN (Density-Based Spatial Clustering of Applications with Noise).
* Hierarchical Clustering (Phân cụm phân cấp).

**Giảm chiều dữ liệu (Dimensionality Reduction):**

**Mục tiêu:** Biến đổi dữ liệu có số chiều cao (nhiều đặc trưng) thành dữ liệu có số chiều thấp hơn, nhưng vẫn giữ lại thông tin quan trọng.

**Ứng dụng:**

* Nén dữ liệu.
* Trực quan hóa dữ liệu (2D hoặc 3D).
* Tiền xử lý trước khi sử dụng dữ liệu trong học có giám sát.

Thuật toán tiêu biểu:

* PCA (Principal Component Analysis).
* t-SNE (t-Distributed Stochastic Neighbor Embedding).
* UMAP (Uniform Manifold Approximation and Projection).

**Phát hiện bất thường (Anomaly Detection):**

**Mục tiêu:** Xác định các điểm dữ liệu bất thường hoặc khác biệt so với phần còn lại của tập dữ liệu.

**Ứng dụng:**

* Phát hiện gian lận trong giao dịch tài chính.
* Bảo trì dự đoán trong các hệ thống công nghiệp.
* Phát hiện xâm nhập mạng.

**Thuật toán tiêu biểu:**

* Isolation Forest.
* Gaussian Mixture Models.

**Học biểu diễn (Representation Learning):**

**Mục tiêu:** Tìm ra cách biểu diễn tốt hơn cho dữ liệu, thường sử dụng mạng nơ-ron tự mã hóa (autoencoders).

**Ứng dụng:**

* Tiền xử lý dữ liệu cho các bài toán học sâu.
* Xử lý ảnh và văn bản.

**Ưu điểm của học không giám sát**

* Linh hoạt: Làm việc với dữ liệu chưa gắn nhãn, vốn chiếm phần lớn trong thực tế.
* Khám phá dữ liệu: Giúp tìm ra các mẫu ẩn, mối quan hệ tiềm ẩn hoặc cấu trúc dữ liệu mà con người khó nhận thấy.
* Tiết kiệm chi phí: Không cần nhãn, do đó giảm thời gian và công sức dán nhãn dữ liệu.

**Hạn chế của học không giám sát**

* Độ chính xác thấp: Không có nhãn để so sánh, kiểm tra độ chính xác khó hơn.
* Khó đánh giá: Kết quả của thuật toán thường khó đánh giá chất lượng vì không có "đáp án" cụ thể.
* Nhạy cảm với tham số: Hiệu quả phụ thuộc nhiều vào việc chọn tham số (ví dụ: số cụm kkk trong K-Means).

**Ứng dụng của học không giám sát**

**Marketing và kinh doanh:**

* Phân khúc khách hàng dựa trên hành vi hoặc sở thích.
* Gợi ý sản phẩm (Recommendation Systems).

**Y tế:**

* Phân nhóm bệnh nhân dựa trên dữ liệu y khoa.
* Tìm mẫu gene tương quan với các loại bệnh.

**Bảo mật:**

* Phát hiện các hoạt động bất thường trên mạng.
* Xác định hành vi gian lận trong giao dịch tài chính.

**Trực quan hóa dữ liệu:**

* Hiển thị các tập dữ liệu phức tạp trong không gian 2D hoặc 3D để dễ dàng phân tích.

**Xử lý ảnh và video:**

* Phân nhóm các đối tượng trong ảnh.
* Tách các thành phần quan trọng trong video

### 1.2.3 Học bán giám sát

Học bán giám sát (Semi-Supervised Learning) là một kỹ thuật học máy nằm giữa học có giám sát (Supervised Learning) và học không giám sát (Unsupervised Learning). Trong đó, thuật toán sử dụng một tập dữ liệu lớn chưa gắn nhãn (unlabeled data) và một tập dữ liệu nhỏ đã gắn nhãn (labeled data) để học và dự đoán.

Học bán giám sát tận dụng thông tin từ cả dữ liệu có nhãn và dữ liệu không nhãn để cải thiện độ chính xác của mô hình mà không yêu cầu lượng lớn dữ liệu gắn nhãn, vốn tốn kém chi phí và thời gian.

A diagram of a computer network

Description automatically generated

Hình 1.11. Quy trình học máy bán giám sát [6]

**Học từ dữ liệu có nhãn:**

Dữ liệu có nhãn được sử dụng để xây dựng một mô hình ban đầu, như trong học có giám sát.

**Tận dụng dữ liệu không nhãn:**

Sử dụng các đặc điểm hoặc mẫu từ dữ liệu không nhãn để cải thiện mô hình.

**Các phương pháp phổ biến:**

* **Pseudo-labeling:** Gán nhãn giả cho dữ liệu không nhãn bằng cách sử dụng dự đoán từ mô hình ban đầu.
* **Consistency Regularization:** Đảm bảo rằng mô hình dự đoán tương tự nhau cho cùng một dữ liệu khi áp dụng các biến đổi khác nhau.
* **Cluster Assumption:** Giả định rằng dữ liệu thuộc cùng một cụm (cluster) có khả năng thuộc cùng một nhãn.

**Ưu điểm của học bán giám sát**

* **Tiết kiệm tài nguyên:** Không cần lượng lớn dữ liệu có nhãn, giảm chi phí và thời gian gắn nhãn thủ công.
* **Cải thiện hiệu suất:** Dữ liệu không nhãn giúp mô hình học tốt hơn bằng cách khai thác các cấu trúc tiềm ẩn trong dữ liệu.
* **Ứng dụng rộng rãi:** Thích hợp với các tình huống mà dữ liệu có nhãn rất ít, nhưng dữ liệu không nhãn lại rất phong phú.

**Hạn chế của học bán giám sát**

* **Phụ thuộc vào giả định:** Hiệu quả phụ thuộc vào các giả định như dữ liệu không nhãn phải tương tự dữ liệu có nhãn.
* **Chất lượng dữ liệu không nhãn:** Nếu dữ liệu không nhãn chứa nhiều nhiễu hoặc không phù hợp, mô hình có thể bị suy giảm hiệu suất.
* **Khó khăn trong đánh giá:** Đánh giá chất lượng của học bán giám sát khó hơn vì không có nhãn để kiểm tra toàn bộ dữ liệu.

**Các thuật toán học bán giám sát phổ biến**

* **Self-training:** Sử dụng mô hình ban đầu để gán nhãn giả cho dữ liệu không nhãn, sau đó thêm dữ liệu này vào tập huấn luyện.
* **Co-training:** Sử dụng hai mô hình với hai tập đặc trưng khác nhau để gán nhãn cho dữ liệu không nhãn và cải thiện lẫn nhau.
* **Generative Models:** Sử dụng các mô hình sinh, như Autoencoders hoặc GANs, để học các đặc điểm của dữ liệu không nhãn và sử dụng chúng để hỗ trợ phân loại.
* **Graph-based Methods:** Xây dựng một đồ thị mà các điểm dữ liệu là các nút, các cạnh thể hiện độ tương đồng giữa các điểm. Dựa vào cấu trúc đồ thị để gán nhãn cho dữ liệu không nhãn.

**Ứng dụng của học bán giám sát**

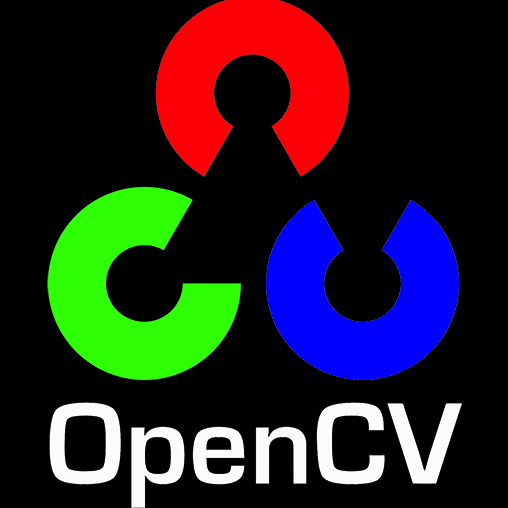
* **Xử lý ngôn ngữ tự nhiên (NLP):** Dịch máy, phân tích cảm xúc, gắn thẻ thực thể (NER).
* **Thị giác máy tính (Computer Vision):** Phân loại ảnh, nhận dạng đối tượng, phân đoạn ảnh.
* **Y tế:** Chẩn đoán bệnh từ hình ảnh y tế (MRI, X-ray).

# CHƯƠNG 2: XÂY DỰNG HỆ THỐNG NHẬN DIỆN MÀU SẮC BẰNG PANDAS VÀ OPENCV

## 2.1. Thư viện

### 2.1.1. CV2 (OpenCV)

**OpenCV (Open Source Computer Vision Library)** là một thư viện mã nguồn mở nổi tiếng được phát triển nhằm phục vụ cho việc xử lý hình ảnh và thị giác máy tính. Dưới đây là toàn bộ khái niệm và thông tin cơ bản về OpenCV:



Hình 2.1. Thư viện OpenCVt [7]

* **Giới thiệu về OpenCV**

**Được phát triển bởi:** Intel vào năm 1999, hiện nay do cộng đồng mã nguồn mở và một số công ty công nghệ duy trì.

**Mục tiêu chính:**

* Xây dựng các ứng dụng thị giác máy tính mạnh mẽ.
* Đẩy nhanh quá trình nghiên cứu và triển khai các thuật toán xử lý hình ảnh.

**Ngôn ngữ hỗ trợ:**

OpenCV hỗ trợ nhiều ngôn ngữ lập trình phổ biến như Python, C++, Java, MATLAB.

**Hệ điều hành hỗ trợ:** Windows, macOS, Linux, Android, iOS.

* **Các tính năng chính của OpenCV**

OpenCV cung cấp các công cụ để làm việc với:

**Xử lý hình ảnh cơ bản:**

* Chuyển đổi giữa các không gian màu (RGB, Grayscale, HSV, YUV...).
* Làm mờ, lọc, và làm sắc nét hình ảnh.
* Xử lý các ngưỡng (thresholding).
* Phát hiện cạnh (Canny, Sobel, Laplacian).

**Xử lý video:**

* Đọc và ghi video từ các file hoặc camera.
* Phân tích khung hình (frame analysis).

**Nhận diện đối tượng:**

Phát hiện khuôn mặt (Haar Cascades, Deep Learning).

Nhận diện vật thể (YOLO, SSD, Faster R-CNN).

Phân đoạn hình ảnh (Image Segmentation).



Hình 2.2. Nhận diện khuôn mặt bằng OpenCVt [8]

**Xử lý hình học:**

* Biến đổi affine và perspective.
* Phát hiện và theo dõi các điểm đặc trưng (Feature Detection and Matching).
* Chỉnh sửa méo ảnh (Distortion Correction).

**Thị giác máy tính nâng cao:**

* Phát hiện chuyển động.
* Theo dõi đối tượng (Object Tracking).
* Phân loại hình ảnh.

**Machine Learning:**

* Cung cấp các module học máy tích hợp như k-NN, SVM, Decision Trees.
* Hỗ trợ tích hợp với các thư viện Deep Learning (TensorFlow, PyTorch).

**3D Vision:**

* Phát hiện và khôi phục hình dạng 3D từ hình ảnh.
* Xử lý dữ liệu từ cảm biến chiều sâu (Depth Sensors).
* **Kiến trúc của OpenCV**

OpenCV được tổ chức thành các module chính:

**core:** Các hàm cơ bản của xử lý ảnh và toán học.

**imgproc:** Các thuật toán xử lý hình ảnh như lọc, phát hiện cạnh, biến đổi hình học.

**highgui:** Công cụ hiển thị hình ảnh và video.

**video:** Xử lý video như phân đoạn chuyển động và theo dõi đối tượng.

**calib3d:** Hiệu chỉnh camera và xử lý ảnh 3D.

**ml:** Các thuật toán học máy.

**features2d:** Phát hiện và mô tả các đặc trưng.

**objdetect:** Phát hiện đối tượng (như khuôn mặt, mắt...).

**photo:** Các thuật toán xử lý ảnh (loại bỏ nhiễu, ghép ảnh panorama).

**dnn:** Tích hợp các mạng học sâu (Deep Neural Networks).

**stitching:** Ghép ảnh tạo panorama.

* **Ứng dụng thực tế của OpenCV**

**Nhận diện khuôn mặt và vật thể:** Dùng trong an ninh, mở khóa bằng khuôn mặt.

**Thực tế tăng cường (AR):** Dùng để ghép các hình ảnh ảo vào thế giới thực.

**Xe tự lái:** Phân tích video và hình ảnh từ cảm biến.

**Robot công nghiệp:** Hướng dẫn và kiểm tra chất lượng sản phẩm.

**Y học:** Phân tích ảnh X-quang, CT scan.

* **Ưu điểm của OpenCV**
* **Mã nguồn mở và miễn phí:** Linh hoạt, dễ sử dụng, và được cộng đồng hỗ trợ mạnh mẽ.
* **Tối ưu hóa cao:** Tích hợp với các công nghệ tăng tốc phần cứng như CUDA, OpenCL.
* **Đa nền tảng:** Hỗ trợ đa hệ điều hành và thiết bị
* **Nhược điểm của OpenCV**
* **Hạn chế về học sâu**: Không tối ưu như TensorFlow, PyTorch trong Deep Learning.
* **Phức tạp cho người mới**: Khó tiếp cận do yêu cầu hiểu biết về thị giác máy tính và toán học.
* **Tối ưu hóa hiệu suất khó**: Cần kinh nghiệm khi làm việc với GPU hoặc ứng dụng thời gian thực.

**Ứng dụng của OpenCV trong chủ đề nghiên cứu**

Chức năng chính:

* Xử lý và chuyển đổi không gian màu (BGR, RGB, HSV).
* Làm mờ ảnh để giảm nhiễu.
* Hiển thị và thao tác trên ảnh với nhiều định dạng.
* Ứng dụng trong dự án:
* Chuyển đổi ảnh từ không gian BGR sang HSV hoặc RGB (cv2.cvtColor).
* Làm mờ ảnh bằng Gaussian Blur (cv2.GaussianBlur) trước khi phân tích.
* Hiển thị kết quả phân cụm màu sắc trên giao diện.

### 2.1.2. Pandas

**Pandas là gì?**

Pandas là một thư viện mã nguồn mở được phát triển bởi Wes McKinney vào năm 2008. Pandas được sử dụng chủ yếu để thao tác, phân tích và dọn dẹp dữ liệu. Pandas cung cấp rất nhiều cấu trúc dữ liệu cũng như các phép tính hỗ trợ thao tác dữ liệu số và dữ liệu thời gian(time series). Pandas nhanh, mạnh và hiệu quả.

**Ưu điểm của Pandas**

* Nhanh và hiệu quả trong thao tác và phân tích dữ liệu.
* Có thể lấy dữ liệu từ nhiều nguồn dữ liệu khác nhau.
* Dễ dàng xử lí các dữ liệu bị thiếu(đại diện bởi : NaN).
* Kích thước linh hoạt: có thể dễ dàng insert và delete dữ liệu.
* Linh hoạt trong việc reshape và pivot dataset.
* Cung cấp các chức năng time-series cũng như các chức năng nhóm dữ liệu(group).

**Bắt đầu với Pandas**

Để có thể sử dụng thư viện Pandas, sau khi cài đặt trong hệ thống, bạn cần phải import thư viện trong text editor hoặc IDE để có thể sử dụng. Câu lệnh import như sau:

|  |
| --- |
| **import pandas as pd** |

Trong đó pd có thể được hiểu là alias của Pandas. Sau khi import xong có thể sử dụng pd để gọi các phương thức của thư viện.

Thông thường Pandas cung cấp 2 cấu trúc dữ liệu chính để thao tác với dữ liệu là: **Series** và **DataFrame**.

**Tìm hiểu về 2 cấu trúc này.**

**Series**

Pandas Series là mảng một chiều được gắn label mà có thể lưu trữ bất kì kiểu dữ liệu nào(integer, string, float…) Có thể hiểu đơn giản Pandas Series giống như là 1 cột trong excel. Mỗi giá trị sẽ có một label trỏ đến chỉ mục(index) của nó, bạn có thể đặt tên cho label với đối số index. Sử dụng label để có thể truy cập đến một giá trị cụ thể

[A screenshot of a computer

Description automatically generated](https://labs.flinters.vn/wp-content/uploads/2021/02/pandas-series.png)

Hình 2.3 Series [9]

Pandas cung cấp nhiều các phép toán để có thể thao tác với Series như: tạo mới, truy cập phần tử, các phép tính nhị phân… Tuy nhiên trong bài viết hôm nay, để nội dung không quá dài mình xin giới thiệu phép toán cơ bản nhất để đó chính là tạo một Series

**Cách tạo một Series**

Trong thực tế, một Pandas Series sẽ được tạo ra bằng cách lấy dữ liệu từ các kho lưu trữ dữ liệu, đó có thể là SQL Database, file CSV, file Excel…

Có nhiều cách để tạo ra một Pandas Series như là tạo từ: Lists, Dictionary hoặc thậm chí là một scalar(một giá trị đơn)… Dưới đây sẽ là một vài cách đơn giản để tạo ra một Series:

**Tạo Series từ array:**

Chúng ta sẽ import module numpy và sử dụng  phương thức array() để tạo

|  |
| --- |
| import pandas as pd  import numpy as np  data = np.array(['f','l','i','n','t','e','r','s'])  series = pd.Series(data)  print(series) |

Output:

[A screenshot of a phone

Description automatically generated](https://labs.flinters.vn/wp-content/uploads/2021/02/Screen-Shot-2021-02-24-at-16.56.51.png)

**Tạo series từ 1 Lists:**

Để tạo 1 Series từ list thì cần phải tạo 1 list trước đó.

import pandas as pd

list = ['f','l','i','n','t','e','r','s']

series = pd.Series(list)

print(list)

**Output:**

[](https://labs.flinters.vn/wp-content/uploads/2021/02/Screen-Shot-2021-02-24-at-17.00.59.png)

**DataFrame**

Pandas DataFrame là cấu trúc dữ liệu bảng không đồng nhất, có 2 chiều và kích thước có thể thay đổi. DataFrame cũng có các labels tương ứng với các hàng và cột. Nói một cách đơn giản thì DataFrame cũng giống như 1 bảng SQL hoặc 1 spreadsheet vậy. Pandas DataFrame bao gồm 3 thành phần chính là : data, rows và columns.

[A screenshot of a data analysis

Description automatically generated](https://labs.flinters.vn/wp-content/uploads/2021/02/dataframe.png)

Hình 2.4 Data Frame [9]

**Cách tạo một DataFrame:**

Cũng giống như Series, một Pandas DataFrame cũng có thể được tạo bằng cách lấy dữ liệu từ các kho dữ liệu như SQL Database, CSV file, Excel file. DataFrame cũng chấp nhận nhiều kiểu input như là : list, dictionary, một list các dictionary…

Mình xin giới thiệu một số cách tạo 1 DataFrame đơn giản:

**Tạo DataFrame từ List:**

Có thể sử dụng 1 list hoặc 1 list các list để làm đối số truyền vào.

import pandas as pd

lst = ['Hello','Flinters','How','Are','You']

df = pd.DataFrame(lst)

print(df)

**Output:**

[A black background with white text

Description automatically generated](https://labs.flinters.vn/wp-content/uploads/2021/02/Screen-Shot-2021-02-25-at-08.38.40.png)

**Tạo từ một dictionary:**

Ở đây mình sẽ truyền vào một dict các lists. Độ dài của các list cần phải giống nhau. Nếu index được truyền vào thì nó cũng phải tương ứng với độ dài của list. Còn nếu không được truyền thì kết quả sẽ mặc định lấy index theo độ dài của list.

import pandas as pd

dic = {

"one": [1,2,3,4,5],

"two": [5,4,3,2,1]

}

df = pd.DataFrame(dic, index=["a","b","c","d","e"])

print(df)

Output:

[A black background with white text

Description automatically generated](https://labs.flinters.vn/wp-content/uploads/2021/02/Screen-Shot-2021-02-25-at-08.46.22.png)

**Tạo từ một Numpy Arrays:**

Mình sẽ truyền vào một Numpy Array tương tự như cách tạo một DataFrame với một List

import pandas as pd

import numpy as np

arr = np.array([[1,2,3], [4,5,6], [7,8,9]])

df = pd.DataFrame(arr, columns = ['a','b','c'])

print(df)

Output:

[A number on a black background

Description automatically generated](https://labs.flinters.vn/wp-content/uploads/2021/02/Screen-Shot-2021-02-25-at-08.54.58.png)

**Ứng dụng của Pandas trong chủ đề nghiên cứu**

**Chức năng chính:**

* Xử lý dữ liệu từ file CSV.
* Tìm kiếm màu gần nhất từ danh sách các màu đã định nghĩa.
* Ứng dụng trong dự án:
* Đọc file colors.csv và colors2.csv để lấy thông tin về các màu (RGB, HSV, tên màu).
* Dùng thông tin màu để xác định màu gần nhất thông qua khoảng cách Euclidean.

A screen shot of a computer

Description automatically generated

### 2.1.3. NumPy

**NumPy (Numerical Python)** là một thư viện mã nguồn mở mạnh mẽ dành cho tính toán khoa học và kỹ thuật trong Python. Nó được thiết kế để xử lý các phép toán trên mảng và ma trận, đồng thời cung cấp nhiều hàm toán học để xử lý dữ liệu nhanh chóng và hiệu quả.



Hình 2.4 Thư Viện NumPy [10]

* **Giới thiệu về NumPy**

**Tên đầy đủ:** Numerical Python.

**Chức năng chính:**

* Làm việc với **mảng (array)** và **ma trận (matrix)** đa chiều.
* Thực hiện các phép toán số học và đại số tuyến tính.

**Mục tiêu:**

Giảm thời gian xử lý dữ liệu bằng cách tối ưu hóa hiệu suất trên các tập dữ liệu lớn.

**Ngôn ngữ hỗ trợ:** Python, với các phần cốt lõi được viết bằng C để tăng tốc độ.

* **Các tính năng chính của NumPy**
* **Mảng đa chiều (ndarray):** Xử lý dữ liệu dưới dạng mảng n chiều với hiệu suất cao hơn danh sách Python.
* **Các hàm toán học:** Thực hiện các phép toán như cộng, trừ, nhân, chia, căn bậc hai, logarit, sin, cos… trên toàn bộ mảng cùng lúc.
* **Đại số tuyến tính:** Hỗ trợ tính toán ma trận như nhân ma trận, nghịch đảo, định thức, trị riêng (eigenvalues).
* **Tích hợp với các thư viện khác:** NumPy là nền tảng cho các thư viện như Pandas, SciPy, Matplotlib, và TensorFlow.
* **Sinh số ngẫu nhiên:** Sinh các dãy số ngẫu nhiên hoặc mẫu thống kê phục vụ phân tích dữ liệu.
* **Hỗ trợ broadcasting:** Cho phép thực hiện các phép toán giữa mảng có kích thước khác nhau một cách tự động và hiệu quả.

**3. Ưu điểm của NumPy**

* **Hiệu suất cao:** Tốc độ xử lý nhanh hơn nhiều so với danh sách Python thông thường nhờ sử dụng các mảng liên tục trong bộ nhớ.

**Dễ sử dụng:** Giao diện đơn giản và trực quan, dễ dàng tích hợp vào các dự án khoa học dữ liệu hoặc học máy.

* **Tích hợp tốt:** Hỗ trợ tốt khi kết hợp với các thư viện Python khác.

**4. Nhược điểm của NumPy**

* **Không hỗ trợ dữ liệu phi số:** NumPy chủ yếu làm việc với số liệu, không tối ưu cho dữ liệu dạng chuỗi hoặc hỗn hợp.
* **Khả năng mở rộng hạn chế:** Không linh hoạt như các thư viện học máy hoặc phân tích dữ liệu phức tạp (ví dụ: Pandas, TensorFlow).

**Ứng dụng của NumPy trong chủ để nghiên cứu**

* Thực hiện các phép toán trên mảng, tính toán khoảng cách giữa các màu sắc.
* Scikit-learn (KMeans): Áp dụng thuật toán KMeans để phân cụm màu sắc chính trên ảnh.

### 2.1.4 Tkinter

Tinker là một module được tích hợp sẵn trong Python, dùng để xây dựng giao diện người dùng đồ họa (Graphical User Interface - GUI). Đây là một công cụ rất phổ biến, đơn giản và dễ sử dụng cho việc phát triển các ứng dụng có giao diện trong Python.



Hình 2.5 Thư Viện Tkinter [11]

* **Giới thiệu Tkinter**
* Tên đầy đủ: Tk Interface.
* Mục đích: Cung cấp một công cụ xây dựng GUI dễ tiếp cận và có khả năng chạy trên nhiều nền tảng (Windows, macOS, Linux).
* Được tích hợp sẵn: Tkinter đi kèm với hầu hết các bản cài đặt Python, nên bạn không cần phải cài đặt riêng.

**Các thành phần chính của Tkinter:**

* Tkinter cung cấp các widget (thành phần giao diện) để xây dựng GUI, bao gồm:
* Label: Hiển thị văn bản hoặc hình ảnh.
* Button: Nút bấm để thực hiện hành động.
* Entry: Ô nhập liệu văn bản.
* Text: Khu vực nhập liệu văn bản lớn.
* Canvas: Khu vực vẽ đồ họa hoặc hiển thị hình ảnh.
* Menu: Thanh menu với các tùy chọn.
* Frame: Khung chứa các widget khác.
* Scrollbar: Thanh cuộn.
* **Các tính năng chính của Tkinter**
* Tạo cửa sổ chính: Dễ dàng tạo và cấu hình cửa sổ chính của ứng dụng.
* Sự kiện và hành động: Xử lý các sự kiện như nhấp chuột, nhập liệu, hoặc bấm phím.
* Tùy chỉnh giao diện: Thay đổi kích thước, màu sắc, kiểu chữ, và nhiều thuộc tính khác của widget.

**Ưu điểm của Tkinter:**

* Dễ sử dụng: Cú pháp đơn giản, phù hợp cho người mới học.
* Đa nền tảng: Ứng dụng Tkinter có thể chạy trên Windows, macOS, và Linux mà không cần chỉnh sửa mã.
* Tích hợp sẵn: Không cần cài đặt thêm, tiết kiệm thời gian thiết lập môi trường.
* Hỗ trợ đầy đủ: Đáp ứng hầu hết các nhu cầu cơ bản của GUI.
* **Nhược điểm của Tkinter**
* Giao diện cơ bản: Không phù hợp để tạo các ứng dụng hiện đại, phức tạp.
* Khả năng mở rộng hạn chế: So với các framework GUI khác như PyQt hoặc Kivy, Tkinter có ít tính năng và tùy chỉnh hơn.
* Hiệu năng: Không tối ưu khi xây dựng ứng dụng lớn hoặc xử lý đồ họa nặng.

**Ứng dụng của Tkinter trong bài toán nghiên cứu**

* Xây dựng giao diện người dùng để chọn ảnh hoặc mở camera.

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

### 2.2.1 Các bước thực hiện:

**Các bước thực hiện**

**Bước 1: Chuẩn bị dữ liệu**

*Dữ liệu màu (CSV):*

* File colors.csv chứa thông tin màu trong không gian HSV với cột H, S, V, và ColorName.
* File colors2.csv chứa thông tin màu trong không gian RGB với cột R, G, B, và Color Name.

**Bước 2: Phân tích và xử lý hình ảnh**

*KMeans clustering:*

* Giảm kích thước ảnh để tăng tốc độ xử lý.
* Reshape ảnh thành tập hợp các pixel và áp dụng thuật toán KMeans để phân cụm các màu.
* Kết quả là danh sách các màu chủ đạo (dominant colors) và bản đồ nhãn (label map) cho mỗi pixel.

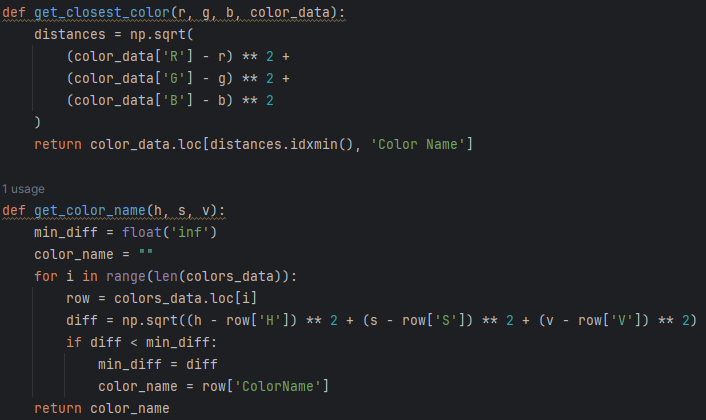
A computer screen shot of a program

Description automatically generated

Hình 2.6 . Hàm xử lý ảnh và trích xuất màu sắc chính

*So sánh với bảng màu:*

* Với mỗi màu chủ đạo, tính khoảng cách Euclidean đến các màu trong colors2.csv để tìm tên màu gần nhất.



Hình 2.7 . Hàm tính toán màu gần nhất

**Bước 3: Hiển thị kết quả**

Tô màu các pixel theo cụm trên ảnh gốc và hiển thị tên màu tương ứng trên từng cụm.

Sử dụng OpenCV để hiển thị các cụm màu dưới dạng cửa sổ riêng biệt.

**Bước 4: Nhận diện màu từ camera**

*Chuyển đổi sang HSV:*

* Chuyển đổi khung hình camera từ không gian màu BGR sang HSV để nhận diện màu sắc ổn định hơn với ánh sáng.
* Lấy trung bình màu của một vùng nhỏ xung quanh tâm ảnh để giảm nhiễu.

*Tìm tên màu:*

* So sánh giá trị HSV trung bình với bảng màu colors.csv để xác định màu gần nhất.

*Hiển thị kết quả:*

* Vẽ một vòng tròn tại tâm khung hình và hiển thị tên màu gần nhất trên video trực tiếp.

**Bước 5: Đánh giá phân cụm**

*Tạo ground truth giả lập:*

* Sử dụng dữ liệu nhãn ngẫu nhiên làm ground truth (vì không có ground truth thực tế).

*Đánh giá KMeans:*

* Tính các chỉ số như Accuracy, Precision, Recall, F1-Score để đánh giá hiệu quả phân cụm.
* Hiển thị chi tiết các giá trị đo lường của Accuracy, Precision, Recall, F1-Score của từng cụm.

### 2.2.2. Một số mã nguồn quan trọng

A screenshot of a computer

Description automatically generated

Hình 2.8. Mã nguồn

**Mô tả mã nguồn**

|  |  |  |
| --- | --- | --- |
| **STT** | **Tên** | **Ý nghĩa** |
| **1** | btl.py | Để triển khai thuật toán Kmeans. |
| **2** | btl\_DecisionTree.py | Để triển khai thuật toán Decision Tree. |
| **3** | btl\_KNN.py | Để triển khai thuật toán K-Nearest Neighbors (KNN). |
| **4** | btl\_SVM.py | Để triển khai thuật toán Support Vector Machine (SVM). |
| **5** | colors.csv | Chứa dữ liệu liên quan đến màu sắc HSV |
| **6** | colors2.csv | Chứa dữ liệu liên quan đến màu sắc RGB |

# CHƯƠNG 3: THỰC NGHIỆM

## 3.1 Dữ liệu

### 3.1.1 Dữ liệu hình ảnh

*Chương trình sử dụng dữ liệu hình ảnh từ hai nguồn chính:*

* **Dữ liệu từ camera:** Người dùng có thể mở camera để thu thập dữ liệu màu sắc theo thời gian thực. Hệ thống xử lý từng khung hình, xác định màu sắc trung tâm và hiển thị kết quả.



Hình 3.1. Đầu vào dữ liệu từ camera

* **Dữ liệu từ tệp hình ảnh:** Người dùng có thể chọn một tệp ảnh (JPEG, PNG) từ hệ thống file. Ảnh được xử lý bằng kỹ thuật phân cụm màu để xác định các màu chính trong ảnh.

A shirt and pants and shoes

Description automatically generated

Hình 3.2. Đầu vào dữ liệu từ tệp hình ảnh [12]

### 3.1.2 Dữ liệu từ file CSV

*Dữ liệu màu sắc được lấy từ hai tệp CSV:*

* **colors.csv:** Chứa thông tin màu sắc trong không gian HSV (Hue, Saturation, Value), cùng tên màu tương ứng.

A screenshot of a computer screen

Description automatically generated

Hình 3.3. Dữ liệu màu tệp colors.csv

* **colors2.csv:** Chứa thông tin màu sắc trong không gian RGB (Red, Green, Blue), cùng tên màu tương ứng.

A screenshot of a computer

Description automatically generated

Hình 3.4. Dữ liệu màu tệp colors2.csv

## 3.2 Độ đo đánh giá

Hệ thống đánh giá hiệu suất phân cụm màu dựa trên các độ đo sau:

### 3.2.1 Accuracy (Độ chính xác)

Cách đơn giản và hay được sử dụng nhất là *accuracy* (độ chính xác). Cách đánh giá này đơn giản tính tỉ lệ giữa số điểm được dự đoán đúng và tổng số điểm trong tập dữ liệu kiểm thử *[13]*

### 3.2.2 Precision (Độ chính xác theo từng cụm)

Precision đo lường khả năng hệ thống chỉ ra đúng màu sắc trong từng cụm. Sử dụng hàm precision\_score với tham số average='weighted'.

A black text on a white background

Description automatically generated

Hình 3.5. Công thức tính Precision [13]

### 3.2.3 Recall (Độ nhạy)

Recall đánh giá khả năng phát hiện đầy đủ các màu thực sự thuộc về từng cụm. Công thức tính:

A black and white math equation

Description automatically generated

Hình 3.6. Công thức tính Recall [13]

Trong đó:

* **Số dự đoán đúng cho mỗi cụm** là số lượng mẫu mà mô hình đã phân loại đúng vào cụm cụ thể.
* **Tổng số mẫu thực tế trong cụm** là tổng số mẫu thực sự thuộc về cụm đó (dựa trên nhãn thực tế).

### 3.2.4 F1-Score

F1-Score là trung bình điều hòa giữa Precision và Recall, sử dụng hàm f1\_score. Công thức:

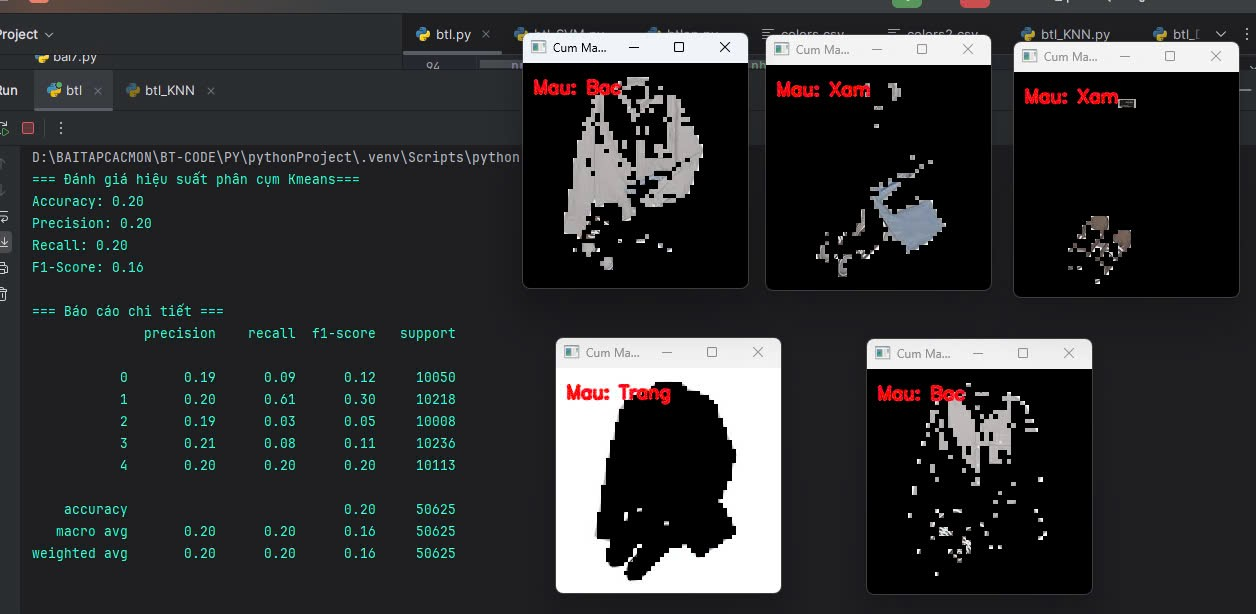
A math equation with black text

Description automatically generated

Hình 3.7. Công thức tính F1-Score [13]

Trong đó:

* **Precision** là tỉ lệ giữa số dự đoán đúng (True Positives) và tổng số dự đoán là dương (True Positives + False Positives).
* **Recall** là tỉ lệ giữa số dự đoán đúng (True Positives) và tổng số mẫu thực sự là dương (True Positives + False Negatives).



Hình 4.8. Kết quả đánh giá đối với Kmeans

## 3.3 Kết quả thực nghiệm

### 3.3.1 Sử dụng SVM

A screenshot of a computer

Description automatically generated

Hình 4.9. Kết quả đánh giá đối với SVM

### 3.3.2 Sử dụng KNN

A screenshot of a computer

Description automatically generated

Hình 4.10. Kết quả đánh giá đối với KNN

### 3.3.3 Sử dụng Decision Tree

A screenshot of a computer

Description automatically generated

Hình 4.11. Kết quả đánh giá đối với Decision Tree

# KẾT LUẬN

**Kết quả đạt được**

Dự án "Xây dựng hệ thống nhận diện màu sắc bằng Pandas và OpenCV" đã đạt được những kết quả quan trọng trong việc phát triển và triển khai một hệ thống nhận diện màu sắc hiệu quả, dựa trên sự kết hợp giữa OpenCV và Pandas. Hệ thống đã thành công trong việc trích xuất và phân tích màu sắc từ hình ảnh, sử dụng các mô hình màu sắc như RGB và HSV để nhận diện và phân loại màu chính xác. Nhờ vào thuật toán KMeans, các màu sắc chủ yếu trong ảnh được xác định và phân loại, đồng thời tên màu cũng được gán từ cơ sở dữ liệu màu sắc có sẵn. Việc sử dụng Pandas giúp quản lý và phân tích dữ liệu màu sắc một cách hiệu quả, cho phép truy vấn, phân tích và xử lý dữ liệu dễ dàng. Hệ thống không chỉ nhận diện màu mà còn hiển thị màu sắc trên ảnh, giúp người dùng dễ dàng nhận ra và đánh giá các màu sắc chính trong từng khu vực của ảnh. Các chỉ số đánh giá như Accuracy, Precision, Recall và F1-Score đã được sử dụng để đảm bảo độ chính xác và hiệu quả của hệ thống. Kết quả đạt được cho thấy hệ thống có khả năng phân loại màu sắc đúng và nhanh chóng, đồng thời có khả năng mở rộng và ứng dụng vào nhiều lĩnh vực như công nghiệp sơn, thời trang, thiết kế đồ họa và phân tích dữ liệu hình ảnh. Giao diện người dùng của hệ thống, được xây dựng bằng Tkinter, đơn giản và dễ sử dụng, giúp người dùng dễ dàng thao tác với ảnh và nhận kết quả nhận diện màu sắc ngay lập tức. Tổng kết lại, dự án không chỉ đạt được mục tiêu về mặt kỹ thuật mà còn mở ra nhiều cơ hội phát triển ứng dụng thực tế trong các ngành nghề đa dạng.

**Hạn chế**

* Khả năng xử lý ánh sáng yếu và độ tương phản thấp: Hệ thống hiện tại gặp khó khăn trong việc nhận diện màu sắc chính xác khi hình ảnh có độ sáng thấp hoặc sự thay đổi đột ngột về ánh sáng. Điều này có thể dẫn đến việc phân loại sai màu sắc hoặc thiếu độ chính xác trong kết quả.
* Giới hạn trong việc phân loại màu sắc phức tạp: Hệ thống chủ yếu sử dụng các mô hình màu cơ bản như RGB và HSV.

**Hướng phát triển**

* Áp dụng học sâu để nâng cao độ chính xác: Việc sử dụng các mô hình học sâu như mạng nơ-ron nhân tạo (CNN) có thể giúp cải thiện khả năng phân loại màu sắc trong các điều kiện phức tạp, từ đó giúp hệ thống nhận diện màu sắc chính xác hơn trong môi trường thay đổi.
* Tối ưu hóa xử lý ảnh trong điều kiện ánh sáng thay đổi: Phát triển các thuật toán xử lý ảnh nâng cao như cân bằng trắng, tăng cường độ tương phản và giảm nhiễu sẽ giúp hệ thống hoạt động tốt hơn trong các điều kiện ánh sáng yếu hoặc có sự thay đổi ánh sáng đột ngột, đảm bảo độ chính xác khi nhận diện màu.
* Mở rộng và hỗ trợ màu sắc phức tạp hơn: Hệ thống có thể được mở rộng để nhận diện các màu sắc pha trộn phức tạp, bao gồm màu gradient hoặc màu sắc phản chiếu, thay vì chỉ phân loại các màu cơ bản. Điều này sẽ giúp mở rộng khả năng ứng dụng trong các ngành công nghiệp như thiết kế thời trang, đồ họa và sản xuất.

# TÀI LIỆU THAM KHẢO

[1] TS Đỗ Năng Toàn, Giáo trình môn học Xử lý ảnh': [Microsoft Word - GTXulyanh.doc](https://pdaotao.duytan.edu.vn/uploads/mark/giao%20trinh%20xu%20ly%20anh.pdf), accessed Dec. 8, 2024."

[2] Idei Club, "CMYK Colour model": [Cmyk colour model - 55 photo](https://en.idei.club/37520-cmyk-colour-model.html)”, accessed Dec. 8, 2024

[3] MachineMFG , “Diferencia de color de pintura: Cómo comprobar y determinar los criterios": [Diferencia de color de pintura: Cómo comprobar y determinar los criterios | MachineMFG](https://www.machinemfg.com/es/check-the-paint-color-difference/)”, accessed Dec. 8, 2024

[4] WikiPedia, “YBCR”: [YCbCr - Wikipedia](https://en.wikipedia.org/wiki/YCbCr), accessed Dec. 8, 2024.

[5] Vũ Hữu Tiệp “Machine Learning Cơ bản”: 27/03/2018, accessed Dec. 8, 2024.

[6] Tiến Sĩ Lương Thị Hồng Lan “Bài giảng Xử lý ảnh và thị giác máy tính”. 23/10/2024, accessed Dec. 8, 2024.

[7] WikiPedia, “OpenCV”: [OpenCV - Wikipedia](https://en.wikipedia.org/wiki/OpenCV), accessed Dec. 8, 2024

[8] Cursa, “Curso Básico de OpenCV Python”: [Curso Básico de OpenCV Python | Cursa Cursos Online Gratuitos Com Certificado Grátis](https://cursa.com.br/home/course/curso-b%C3%A1sico-de-opencv-python/299), accessed Dec. 8, 2024

[9] Flinters “Pandas trong Python (Bài 1): Giới thiệu chung”: [Pandas trong Python (Bài 1): Giới thiệu chung – FLINTERS Developer's Blog](https://labs.flinters.vn/pandas/pandas-trong-python-bai-1-gioi-thieu-chung/), accessed Dec. 8, 2024

[10] MyGPT “11 thư viện Python sử dụng cho học máy và trí tuệ nhân tạo”: [11 thư viện Python sử dụng cho học máy và trí tuệ nhân tạo - MyGPT](https://mygpt.vn/11-thu-vien-python-su-dung-cho-hoc-may-va-tri-tue-nhan-tao/). accessed Dec. 8, 2024

[11] ZenVN“Lập trình phần mềm với Tkinter” : [Lập trình phần mềm với Python Tkinter](https://zendvn.com/khoa-hoc-lap-trinh-phan-mem-voi-tkinter-python) accessed Dec. 8, 2024

[12] Mabu Studio “7 mẹo chụp trải quần áo đẹp lung linh dành cho shop online”: [7 mẹo chụp trải quần áo đẹp lung linh dành cho shop online - Mabu Studio](https://mabustudio.com/7-meo-chup-trai-quan-ao-dep-lung-linh-danh-cho-shop-online/) accessed Dec. 8, 2024

[13] Funda “Machine ”Bài 33: Các phương pháp đánh giá một hệ thống phân lớp”: [Machine Learning cơ bản](https://machinelearningcoban.com/2017/08/31/evaluation/#-precision-recall-curve-va-average-precision) accessed Dec. 8, 2024

Github của nhóm: https://github.com/TV-PhuongQuyen/BTL\_XLA\_TGMT\_Nhom2\_IT6\_K12