BỘ GIÁO DỤC VÀ ĐÀO TẠO

**TRƯỜNG ĐẠI HỌC NHA TRANG**

**KHOA CÔNG NGHỆ THÔNG TIN**



**BÀI TẬP LỚN**

**PLANT CLASSIFICATION**

**GVHD: ThS. NGUYỄN HẢI TRIỀU**

**SVTH: Phan Nguyễn Đình Vũ**

**MSSV: 61131562**

Khánh Hòa, năm 2024

BỘ GIÁO DỤC VÀ ĐÀO TẠO

**TRƯỜNG ĐẠI HỌC NHA TRANG**

**KHOA CÔNG NGHỆ THÔNG TIN**



**BÀI TẬP LỚN**

**PLANT CLASSIFICATION**

**GVHD: ThS. NGUYỄN HẢI TRIỀU**

**SVTH: Phan Nguyễn Đình Vũ**

**MSSV: 61131562**

Khánh Hòa, năm 2024

|  |  |
| --- | --- |
| **TRƯỜNG ĐẠI HỌC NHA TRANG**  KHOA CÔNG NGHỆ THÔNG TIN |  |

**PHIẾU THEO DÕI TIẾN ĐỘ VÀ ĐÁNH GIÁ**

**ĐỒ ÁN/KHÓA LUẬN/CHUYÊN ĐỀ TỐT NGHIỆP**

**(Dùng cho CBHD và nộp cùng báo cáo ĐA/KL/CĐTN của sinh viên)**

**Tên đề tài**: PLANT CLASSIFICATION

Giảng viên hướng dẫn: Ths. Nguyễn Hải Triều

Sinh viên được hướng dẫn: Phan Nguyễn Đình Vũ MSSV: 61131562

Khóa: 61 Ngành: Công nghệ thông tin

|  |  |  |  |
| --- | --- | --- | --- |
| **Lần KT** | **Ngày** | **Nội dung** | **Nhận xét của GVHD** |
| 1 |  | Tổng quan OpenCV, Scikit-learn |  |
| 2 |  | Nghiên cứu RandomForest model |  |
| 3 |  | Đóng gói ứng dụng |  |
| 4 |  | Báo cáo |  |
| **Kiểm tra giữa tiến độ của trưởng Bộ môn** | | | |
| Ngày kiểm tra:  ……………………. | | Đánh giá công việc hoàn thành: …..%  Được tiếp tục:  Không tiếp tục:  | Ký tên ………………………. |
|  | |  |  |
|  | |  |  |
|  | |  |  |

Nhận xét chung (sau khi sinh viên hoàn thành ĐA/KL/CĐTN): sinh viên Phan Nguyễn Đình Vũ có tinh thần trách nhiệm, hoàn thành tốt các nội dung được giao. Kết quả thu được thỏa mãn yêu cầu của đề tài.

Điểm hình thức: /10 Điểm nội dung: /10 Điểm tổng kết: /10

+ Đối với ĐA/KLTN:

Kết luận sinh viên: Được bảo vệ: 🗹 Không được bảo vệ: 

Khánh Hòa, ngày…….tháng 06 năm 2024

Cán bộ hướng dẫn

(Ký và ghi rõ họ tên)

Nguyễn Hải Triều

**TRƯỜNG ĐẠI HỌC NHA TRANG**

KHOA: CÔNG NGHỆ THÔNG TIN

**PHIẾU CHẤM ĐIỂM CHUYÊN ĐỀ TỐT NGHIỆP**

**(Dành cho cán bộ chấm phản biện)**

1. **Họ tên người chấm**:
2. **Sinh viên/ nhóm sinh viên thực hiện CDTN** (sĩ số trong nhóm):

(1)…………………………………………………..MSSV.............................................  
(2)…………………………………………………..MSSV.............................................  
(3)…………………………………………………..MSSV.............................................  
(4)…………………………………………………..MSSV.............................................  
Lớp:………………………..Ngành:..................................................................................

1. **Tên đề tài:**
2. **Nhận xét**

- Hình thức:

- Nội dung:

Điểm hình thức:……/10 Điểm nội dung:......./10 **Điểm tổng kết**:………/10

*Khánh Hòa, ngày…….tháng 06 năm 2024*  
**Cán bộ chấm phản biện***(Ký và ghi rõ họ tên)*

1. **LỜI CAM ĐOAN**

Tôi xin cam đoan đây là công trình nghiên cứu của tôi dưới sự hướng dẫn của ThS. Nguyễn Hải Triều.

***Nha Trang, tháng 06 năm 2024***

Sinh viên

Phan Nguyễn Đình Vũ

# **LỜI CẢM ƠN**

Em xin gửi lời cảm ơn chân thành đến Quý thầy (cô) của khoa Công nghệ thông tin, đề tài này đem lại nhiều trải nghiệm và giúp em hiểu rõ về cơ chế của hệ thống phân loại tự động. Em được học nhiều kỹ thuật, kinh nghiệm mới trong quá trình thực hiện đề tài này.

Em cũng muốn bày tỏ lòng cảm kích đến giảng viên hướng dẫn thầy Nguyễn Hải Triều, thầy đã tận tình hướng dẫn những lúc khó khăn, đóng góp những ý kiến trong quá trình thực hiện đề tài, giúp em định hướng được đề tài “***Plant Classification***”, Nhờ có sự giúp đỡ của thầy, em đã hoàn thành đề tài này một cách hoàn thiện nhất.

Em xin chân thành cảm ơn.

Sinh viên

Phan Nguyễn Đình Vũ

**MỤC LỤC**

[**TRƯỜNG ĐẠI HỌC NHA TRANG** 3](#_Toc168275316)

[KHOA CÔNG NGHỆ THÔNG TIN 3](#_Toc168275317)

[LỜI CAM ĐOAN i](#_Toc168275318)

[LỜI CẢM ƠN ii](#_Toc168275319)

[DANH MỤC CÁC TỪ VIẾT TẮT v](#_Toc168275320)

[DANH MỤC HÌNH ẢNH vii](#_Toc168275321)

[PHẦN MỞ ĐẦU ix](#_Toc168275322)

[Chương 1: TỔNG QUAN VỀ ĐỀ TÀI NGHIÊN CỨU 10](#_Toc168275323)

[**1.1** **LÝ DO CHỌN ĐỀ TÀI** 10](#_Toc168275324)

[**1.2** **MỤC TIÊU ĐỀ TÀI** 10](#_Toc168275325)

[Chương 2: CƠ SỞ LÝ THUYẾT 11](#_Toc168275326)

[**2.1** **NGÔN NGỮ LẬP TRÌNH PYTHON** 11](#_Toc168275327)

[**2.1.1** **Giới thiệu** 11](#_Toc168275328)

[**2.1.2** **Ưu điểm** 11](#_Toc168275329)

[**2.1.3** **Nhược điểm** 11](#_Toc168275330)

[**2.1.4** **Cài đặt** 12](#_Toc168275331)

[**2.2** **THƯ VIỆN PYQT** 12](#_Toc168275332)

[**2.2.1** **Giới thiệu** 12](#_Toc168275333)

[**2.2.2** **Ưu điểm** 13](#_Toc168275334)

[**2.2.3** **Nhược điểm** 13](#_Toc168275335)

[**2.3** **TỔNG QUAN VỀ THƯ VIỆN OPENCV** 14](#_Toc168275336)

[**2.3.1** **Giới thiệu** 14](#_Toc168275337)

[**2.3.2** **Thế mạnh** 14](#_Toc168275338)

[**2.3.3** **Hạn chế** 14](#_Toc168275339)

[**2.3.4** **Cài đặt** 15](#_Toc168275340)

[**2.3.5** **Ứng dụng** 15](#_Toc168275341)

[**2.3.6** **Ví dụ về bài toán Drawing and Writing** 16](#_Toc168275342)

[**2.4** **TỔNG QUAN VỀ SCIKIT-LEARN** 19](#_Toc168275343)

[**2.4.1** **Scikit-learn là gì?** 19](#_Toc168275344)

[**2.4.2** **Cách hoạt động** 19](#_Toc168275345)

[**2.4.3** **Thuật toán** 20](#_Toc168275346)

[**2.4.4** **Ưu và nhược điểm** 22](#_Toc168275347)

[**2.4.5** **Ứng dụng** 23](#_Toc168275348)

[**2.4.6** **Ví dụ về bài toán cây phân loại và hồi quy** 23](#_Toc168275349)

[**2.5** **TỔNG QUAN VỀ NSIS** 25](#_Toc168275350)

[**2.5.1** **Giới thiệu** 25](#_Toc168275351)

[**2.5.2** **Các tính năng chính** 25](#_Toc168275352)

[**2.5.3** **Ưu và nhược điểm** 25](#_Toc168275353)

[Chương 3 XÂY DỰNG BÀI TOÁN PLANT CLASSIFICATION 26](#_Toc168275354)

[**3.1** **DỮ LIỆU ĐẦU VÀO** 26](#_Toc168275355)

[**3.2** **MỤC TIÊU** 26](#_Toc168275356)

[**3.3** **PHƯƠNG PHÁP** 26](#_Toc168275357)

[**3.3.1** **Sử Dụng Histogram Màu** 26](#_Toc168275358)

[**3.3.2** **Xây Dựng Bộ Phân Loại Học Máy** 27](#_Toc168275359)

[**3.4** **CÀI ĐẶT VÀ DEMO** 31](#_Toc168275360)

[Chương 4: KẾT LUẬN 37](#_Toc168275361)

[**4.1 ĐÁNH GIÁ ĐỀ TÀI** 37](#_Toc168275362)

[**4.1.1 Kết quả** 37](#_Toc168275363)

[**4.1.2 Khuyết điểm** 37](#_Toc168275364)

[**4.2 HƯỚNG HOÀN THIỆN** 37](#_Toc168275365)

[TÀI LIỆU THAM KHẢO 38](#_Toc168275366)

1. DANH MỤC CÁC TỪ VIẾT TẮT

|  |  |
| --- | --- |
| **Ký Hiệu** | **Thuật ngữ** |
| API | Application Programming Interface |
| RBG | RED BLUE GREEN |
| NSIS | Nullsoft Scriptable Install System |
| OpenCV | Open Source Computer Vision Library |
| ML | Machine Learning |
| PCA | Principal Component Analysis |
| GUI | Graphical User Interface |
| IDLE | Integrated Development and Learning Environment |
| PIP | Preferred Installer Program |

# **DANH MỤC HÌNH ẢNH**

[*Hình 2.1 Python* 11](#_Toc168275367)

[*Hình 2.2 Trang chủ Python* 12](#_Toc168275368)

[*Hình 2.3 PyQt* 12](#_Toc168275369)

[*Hình 2.4 OpenCV* 14](#_Toc168275370)

[*Hình 2.5 Câu lệnh cài đặt OpenCV* 15](#_Toc168275371)

[*Hình 2.6 Câu lệnh kiểm tra* 15](#_Toc168275372)

[*Hình 2.7 Ứng dụng của OpenCV* 16](#_Toc168275373)

[*Hình 2.8 Ảnh đầu vào* 16](#_Toc168275374)

[*Hình 2.9 Import thư viện và hiển thị ảnh* 16](#_Toc168275375)

[*Hình 2.10 Vẽ đường thẳng và hiển thị ảnh* 17](#_Toc168275376)

[*Hình 2.11 Kết quả* 17](#_Toc168275377)

[*Hình 2.12 Vẽ hình chữ nhật* 17](#_Toc168275378)

[*Hình 2.13 Vẽ hình tròn* 17](#_Toc168275379)

[*Hình 2.14 Tạo một mảng điểm để vẽ đa giác* 18](#_Toc168275380)

[*Hình 2.15 Thêm chữ vào hình* 18](#_Toc168275381)

[*Hình 2.16 Kết quả (1)* 18](#_Toc168275382)

[*Hình 2.17 Scikit-learn* 19](#_Toc168275383)

[*Hình 2.18 Quy trình* 19](#_Toc168275384)

[*Hình 2.19 Thuật toán phân cụm trung bình* 20](#_Toc168275385)

[*Hình 2.20 Iris Datasets* 23](#_Toc168275386)

[*Hình 2.21 Code ví dụ cho bài toán* 24](#_Toc168275387)

[*Hình 2.22 Classification Report và Confusion Matrix của mô hình* 24](#_Toc168275388)

[*Hình 2.23 NSIS* 25](#_Toc168275389)

[*Hình 3.1 Input ảnh định dạng jpg* 26](#_Toc168275390)

[*Hình 3.2 Hàm init cho lớp RGBHistogram* 26](#_Toc168275391)

[*Hình 3.3 Hàm Describe* 27](#_Toc168275392)

[*Hình 3.4 Phân tích các đối số dòng lệnh* 27](#_Toc168275393)

[*Hình 3.5 Đọc ảnh và mặt nạ* 27](#_Toc168275394)

[*Hình 3.6 Kiểm tra ảnh và mặt nạ* 28](#_Toc168275395)

[*Hình 3.7 Khởi tạo RGBHistogram với bins* 28](#_Toc168275396)

[*Hình 3.8 Trích xuất các đặc điểm và nhãn từ mỗi cặp hình ảnh và mặt nạ* 28](#_Toc168275397)

[*Hình 3.9 Đọc và kiểm tra* 28](#_Toc168275398)

[*Hình 3.10 Chuyển mặt nạ sang ảnh xám* 28](#_Toc168275399)

[*Hình 3.11 Trích xuất các đặc điểm bằng RGBHistogram* 29](#_Toc168275400)

[*Hình 3.12 Trích xuất tên hoa* 29](#_Toc168275401)

[*Hình 3.13 Kiểm tra* 29](#_Toc168275402)

[*Hình 3.14 Mã hóa nhãn* 29](#_Toc168275403)

[*Hình 3.15 Chia tập dữ liệu* 29](#_Toc168275404)

[*Hình 3.16 Huấn luyện mô hình* 30](#_Toc168275405)

[*Hình 3.17 Đánh giá mô hình* 30](#_Toc168275406)

[*Hình 3.18 Dự đoán và hiển thị 10 hình ảnh ngẫu nhiên từ tập test* 30](#_Toc168275407)

[*Hình 3.19 Output* 31](#_Toc168275408)

[*Hình 3.20 File cài đặt* 31](#_Toc168275409)

[*Hình 3.21 Màn hình cài đặt* 31](#_Toc168275410)

[*Hình 3.22 Màn hình cài đặt (1)* 32](#_Toc168275411)

[*Hình 3.23 Màn hình cài đặt (2)* 32](#_Toc168275412)

[*Hình 3.24 Màn hình cài đặt (3)* 33](#_Toc168275413)

[*Hình 3.25 Shortcut* 33](#_Toc168275414)

[*Hình 3.26 Giao diện chính* 33](#_Toc168275415)

[*Hình 3.27 Chọn đường dẫn cho việc train model* 34](#_Toc168275416)

[*Hình 3.28 Thông báo train xong và hiện độ chính xác* 34](#_Toc168275417)

[*Hình 3.29 Lấy ảnh từ Google* 35](#_Toc168275418)

[*Hình 3.30 Chọn ảnh để dự đoán* 35](#_Toc168275419)

[*Hình 3.31 Dự đoán* 36](#_Toc168275420)

# **PHẦN MỞ ĐẦU**

Trong thời đại công nghệ số, việc áp dụng các phương pháp tự động hóa đang trở nên ngày càng phổ biến. Một trong những ứng dụng đáng chú ý là việc phân loại các loài thực vật, đặc biệt là các loài hoa. Việc xác định loài hoa yêu cầu kiến thức sâu rộng của các nhà thực vật học và quá trình này có thể rất tốn thời gian cũng như chi phí.

Nhằm giải quyết thách thức này, em đã phát triển một hệ thống phân loại tự động các loài hoa dựa trên hình ảnh. Mục tiêu là xây dựng một bộ phân loại có thể tự động nhận dạng và phân loại các loài hoa từ hình ảnh, giúp giảm bớt khối lượng công việc thủ công. Phân loại tự động không chỉ giúp tiết kiệm thời gian và chi phí mà còn tăng tính đúng đắn và nhất quán trong phân biệt các loài hoa. Điều này có ý nghĩa quan trọng không chỉ đối với các bảo tàng và vườn bách thảo mà còn đối với việc bảo tồn thiên nhiên, nơi việc xác định chính xác các loài thực vật đóng vai trò then chốt.

Em đã dùng phương pháp học máy với histogram màu để xây dựng mô hình phân loại. Bằng cách này, em đã tạo ra một hệ thống có khả năng phân loại các loài hoa một cách chính xác và hiệu quả. Mô hình này không chỉ là một công cụ mạnh mẽ trong việc phân loại hoa mà còn là một ví dụ điển hình về việc áp dụng công nghệ hiện đại vào giải quyết các vấn đề truyền thống trong khoa học và đời sống.

Vì vậy em lựa chọn đề tài “**Plant Classification**”. Dùng ngôn ngữ Python, thư viện OpenCV, Scikit-learn, sử dụng thư viện PyQt để thiết kế GUI và sử dụng NSIS để tạo trình cài đặt.

Nội dung bài báo cáo gồm 4 chương:

**Chương 1: Tổng quan**

**Chương 2: Cơ sở lý thuyết**

**Chương 3: Xây dựng bài toán Plant Classification**

**Chương 4: Kết luận**

# **Chương 1: TỔNG QUAN VỀ ĐỀ TÀI NGHIÊN CỨU**

* 1. **LÝ DO CHỌN**

Phân biệt các loài hoa từ lâu đã là một công việc cần thiết cho nhiều ngành như thực vật học, nông nghiệp, và bảo tồn thiên nhiên. Tuy nhiên, việc này đòi hỏi nhiều kiến thức chuyên môn và thời gian của các nhà thực vật học. Một số lý do chính để chọn đề tài này bao gồm:

* **Tiết Kiệm Thời Gian và Chi Phí**: Việc phân loại hoa bằng phương pháp truyền thống rất tốn thời gian và chi phí. Bằng việc tự động hóa, sẽ giảm đáng kể thời gian và nguồn lực cần thiết.
* **Tăng Độ Chính Xác và Nhất Quán**: Phân loại thủ công có thể dẫn đến sai sót do yếu tố con người. Sử dụng máy học giúp đảm bảo tính nhất quán và tính đúng đắn cao hơn.
* **Ứng Dụng Công Nghệ Mới**: Việc sử dụng học máy và xử lý hình ảnh vào lĩnh vực thực vật học không chỉ là 1 bước ngoặt trọng yếu mà còn mở ra nhiều cơ hội nghiên cứu và ứng dụng mới.
* **Hỗ Trợ Nghiên Cứu Khoa Học**: Hệ thống phân loại tự động giúp các nhà nghiên cứu tập trung vào các nghiên cứu khoa học sâu hơn, thay vì tốn thời gian vào các công việc phân loại thủ công.
  1. **MỤC TIÊU ĐỀ TÀI**

Đề tài nghiên cứu này được xây dựng với các mục tiêu cụ thể sau:

1. **Phát Triển Hệ Thống Phân Loại Tự Động**: Xây dựng một hệ thống có khả năng tự động phân loại các loài hoa từ hình ảnh một cách chính xác và hiệu quả.
2. **Ứng Dụng Histogram Màu và Học Máy**: Sử dụng histogram màu 3D RGB để đặc trưng hóa màu sắc của các cánh hoa, kết hợp với các thuật toán học máy để xây dựng model.
3. **Đánh Giá Hiệu Suất Hệ Thống**: Thử nghiệm và đánh giá hiệu suất của hệ thống trên các dataset khác nhau để đảm bảo tính chính xác cao và tính khả thi của model.

# **Chương 2: CƠ SỞ LÝ THUYẾT**

## **NGÔN NGỮ LẬP TRÌNH PYTHON**

### **Giới thiệu**



*Hình 2.1 Python*

Python là ngôn ngữ lập trình bậc cao được ra mắt vào năm 1991 do Guido van Rossum tạo ra. Python được tạo ra cho nhiều mục đích lập trình đa năng với các ưu điểm là dễ đọc, dễ nhớ, dễ học.

Sau phiên bản đầu tiên, phiên bản Python 2 đã được ra mắt vào tháng 10 năm 2000. Và sau 8 năm, phiên bản Python 3 đã được cập nhật vào tháng 12 năm 2008. Và cho đến nay, phiên bản hiện tại là Python 3.11.2.

### **Ưu điểm**

* **Không có cú pháp phức tạp:** Khá dễ đọc và dễ học nên người dùng có thể chú tâm vào xây dựng logic.
* **Miễn phí:** Có thể tải xuống cả mã nguồn của Python, thay đổi mã nguồn và thậm chí là phân phối mã nguồn đã thay đổi.
* **Hệ sinh thái thư viện đa dạng và phong phú:** Dùng cho các mục đích khác nhau, được sử dụng phổ biến.
* **Tích hợp nhiều ngôn ngữ:** Trong khi phát triển dự án, người dùng có thể tích hợp thêm nhiều ngôn ngữ khác qua các thư viện như Cython hay Jython,…

### **Nhược điểm**

* **Rất đơn giản:** Nếu đã quá quen sử dụng nó thì sẽ khá khó để tiếp cận với ngôn ngữ khác.
* **Khó phát hiện lỗi:** Các lỗi chỉ xuất hiện khi chạy chương trình, việc này khiến việc kiểm tra và sửa lỗi trở nên khó khăn.
* **Bảo mật kém:** Python không được coi là ngôn ngữ được đánh giá cao về mặt an toàn và có một số nguy cơ liên quan đến bảo mật. Vì thế nên có một số hạn chế về việc truy cập cơ sở dữ liệu.
* **Tiêu tốn nhiều dung lượng:** Để lưu trữ cấu trúc, với tốc độ thực thi chậm hơn so với một số ngôn ngữ khác và khả năng tiêu tốn nhiều dung lượng, có thể không phù hợp cho việc phát triển những ứng dụng di động và trò chơi yêu cầu hiệu năng cao.

### **Cài đặt**

Ở Windows có rất nhiều phương pháp để cài đặt Python như cài đặt trực tiếp hay thông qua các IDLE. Tuy nhiên, cách đơn giản nhất là tải gói cài đặt từ trang chủ của Python.



*Hình 2.2 Trang chủ Python*

## **THƯ VIỆN PYQT**

### **2.2.1 Giới thiệu**



*Hình 2.3 PyQt*

PyQt là một bộ công cụ phát triển ứng dụng giao diện đồ họa mạnh mẽ và linh hoạt cho Python, dựa trên thư viện Qt của C++. Được phát triển và duy trì bởi Riverbank Computing, PyQt cung cấp các lớp và phương thức giúp lập trình viên dễ dàng tạo ra các ứng dụng GUI chuyên nghiệp và phức tạp. PyQt hỗ trợ nhiều phiên bản của Qt, bao gồm cả Qt 4 và Qt 5, và các tính năng cần thiết để xây dựng các ứng dụng desktop mạnh mẽ.

### **Ưu điểm**

* **Đa Nền Tảng:** PyQt hỗ trợ phát triển ứng dụng trên nhiều hệ điều hành khác nhau như Windows, macOS, và Linux. Điều này giúp lập trình viên phát triển một lần và chạy trên nhiều nền tảng khác nhau mà không cần thay đổi mã nguồn.
* **Giao Diện Người Dùng Chuyên Nghiệp:** PyQt cung cấp các công cụ và widget mạnh mẽ, cho phép tạo ra các ứng dụng GUI với giao diện chuyên nghiệp. Các widget trong PyQt rất đa dạng, phong phú và có thể tùy chỉnh dễ dàng.
* **Tích Hợp Tốt Với Python:** PyQt tích hợp tốt với Python. Điều này giúp lập trình viên tận dụng được tất cả các thư viện và công cụ của Python khi phát triển ứng dụng GUI.
* **Hỗ Trợ Đồ Họa Cao Cấp:** PyQt hỗ trợ vẽ đồ họa cao cấp thông qua QPainter và các lớp liên quan, cho phép tạo ra các hình ảnh và đồ họa tùy chỉnh, phức tạp.
* **Cộng Đồng và Tài Liệu Phong Phú:** PyQt có một cộng đồng lớn và tài liệu phong phú, bao gồm nhiều hướng dẫn, ví dụ, và diễn đàn hỗ trợ, giúp lập trình viên dễ dàng học hỏi.

### **2.2.3 Nhược điểm**

* **Độ Phức Tạp:** Mặc dù PyQt mạnh mẽ, nhưng nó cũng có thể phức tạp hơn với 1 số thư viện GUI như Tkinter. Việc học và sử dụng tất cả các tính năng của PyQt có thể mất nhiều thời gian.
* **Kích Thước Ứng Dụng:** Ứng dụng được xây dựng bởi PyQt thường có dung lượng lớn so với các thư viện GUI nhẹ hơn do phải bao gồm cả thư viện Qt.
* **Hiệu Suất:** Trong một số trường hợp, PyQt chậm hơn với các GUI khác do mức độ trừu tượng cao và tính năng phong phú.
* **Khó Khăn Trong Quản Lý Bố Cục:** Mặc dù PyQt cung cấp nhiều lựa chọn về bố cục, nhưng việc quản lý bố cục phức tạp có thể trở nên khó khăn và yêu cầu hiểu biết sâu về cơ chế của các layout manager.
  1. **TỔNG QUAN VỀ THƯ VIỆN OPENCV**

### **Giới thiệu**

*Hình 2.4 OpenCV*

Là thư viện mã nguồn mở phát triển nhằm cung cấp công cụ và hỗ trợ trong việc xử lý và phân tích ảnh và video, cũng như các bài toán thị giác máy tính khác. Được phát hành lần đầu vào năm 1999 bởi Intel, OpenCV hiện nay đã trở thành một trong những thư viện phổ biến và mạnh mẽ nhất trong lĩnh vực thị giác máy tính.

### **Thế mạnh**

* **Mã Nguồn Mở:** OpenCV là mã nguồn mở và được phát hành theo Giấy phép Apache 2. Nó là miễn phí cho sử dụng thương mại.
* **Tối ưu hóa:** OpenCV là một thư viện được tối ưu hóa cao, tập trung vào các ứng dụng thời gian thực.
* **Đa nền tảng:** Giao diện C++, Python và Java hỗ trợ Linux, MacOS, Windows, iOS và Android.

### **Hạn chế**

* **Độ Phức Tạp Cao:** OpenCV rất mạnh mẽ và phong phú về tính năng, nhưng điều này cũng làm tăng độ phức tạp trong việc học.
* **Quản Lý Bộ Nhớ:** Mặc dù OpenCV đã được tối ưu hóa về hiệu suất, nhưng việc quản lý bộ nhớ có thể là một thách thức. Lập trình viên cần phải cẩn thận để tránh rò rỉ bộ nhớ.
* **Cập Nhật Chậm:** Mặc dù cộng đồng OpenCV rất tích cực, nhưng đôi khi việc cập nhật và bổ sung các tính năng mới có thể diễn ra chậm hơn so với nhu cầu và tốc độ phát triển.

### **Cài đặt**

* **Cài đặt OpenCV:** Ta có thể cài đặt OpenCV bằng pip:

****

*Hình 2.5 Câu lệnh cài đặt OpenCV*

* **Kiểm tra cài đặt:** Kiểm tra xem OpenCV đã được cài đặt bằng lệnh sau:

**A black background with white text

Description automatically generated**

*Hình 2.6 Câu lệnh kiểm tra*

Nếu không có xuất hiện lỗi, thì OpenCV đã được cài đặt thành công.

### **Ứng dụng**

OpenCV được ứng dụng rộng rãi trong nhiều lĩnh vực khác nhau, từ nghiên cứu học thuật đến công nghiệp:

* **Nhận Diện Mặt Người**: OpenCV hỗ trợ nhận diện và phát hiện khuôn mặt trong ảnh và video.
* **Thị Giác Robot**: Dùng cho hệ thống robot để theo dõi và nhận ra đối tượng.
* **Xe Tự Lái**: Hỗ trợ các hệ thống nhận dạng và phân tích hình ảnh để điều khiển xe tự lái.
* **Y Tế**: Phân tích hình ảnh y khoa để hỗ trợ chẩn đoán bệnh.
* **Nông Nghiệp Thông Minh**: Theo dõi và phân loại cây trồng, kiểm tra chất lượng nông sản.



*Hình 2.7 Ứng dụng của OpenCV*

### **Ví dụ về bài toán Drawing and Writing**

#### **Giới thiệu**

*Hình 2.8 Ảnh đầu vào*

Có nhiều cách để vẽ các hình dạng khác nhau trên hình ảnh và video, việc này khá phổ biến để đánh dấu các đối tượng đã được phát hiện. Điều này giúp chúng ta dễ dàng kiểm tra xem các chương trình có hoạt động đúng như mong đợi hay không.

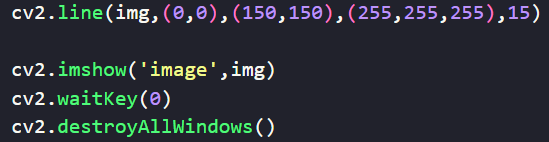
#### **2.3.6.2 Giải quyết**

* Import các thư viện cần thiết.

A screen shot of a computer

Description automatically generated

*Hình 2.9 Import thư viện và hiển thị ảnh*

* Vẽ một đường thẳng màu trắng, độ dày 50-pixel từ điểm (0,0) đến điểm (200, 300):

*Hình 2.10 Vẽ đường thẳng và hiển thị ảnh*

A watch with a leather band

Description automatically generatedKết quả nhận được:

*Hình 2.11 Kết quả*

* Một hình chữ nhật màu đỏ, độ dày 15-pixel từ điểm (500, 250) đến điểm (1000, 500):



*Hình 2.12 Vẽ hình chữ nhật*

Gồm tham số ảnh, tọa độ trái, tọa độ bên dưới bên phải, màu sắc, và độ dày nét. Tiếp đến, ta vẽ 1 hình tròn màu xanh lá cây, bán kính 63-pixel tại điểm (447, 63), -1 có nghĩa là tô kín:



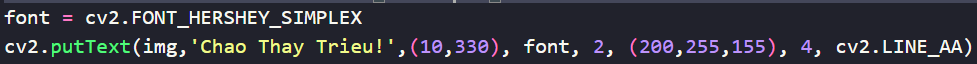
*Hình 2.13 Vẽ hình tròn*

* Vẽ 1 hình đa giác màu vàng, độ dày 3-pixel, True có nghĩa là đóng kín đa giác:



*Hình 2.14 Tạo một mảng điểm để vẽ đa giác*

* Cuối cùng là viết trên ảnh:



*Hình 2.15 Thêm chữ vào hình*

Kết quả cuối cùng nhận được:



*Hình 2.16 Kết quả (1)*

## **2.4 TỔNG QUAN VỀ SCIKIT-LEARN**

### **2.4.1 Scikit-learn là gì?**

A logo for a science company

Description automatically generated with medium confidence

*Hình 2.17 Scikit-learn*

Thường được viết tắt là sklearn, thư viện mã nguồn mở mạnh mẽ dành cho học máy trong ngôn ngữ lập trình Python. Được phát triển từ năm 2007 và hiện đang được duy trì bởi một cộng đồng rộng lớn, scikit-learn cho 1 loạt công cụ để giải quyết bài toán học máy, từ tiền xử lý dữ liệu, phân loại, hồi quy, clustering, đến đánh giá mô hình.

### **Cách hoạt động**

Scikit-learn chủ yếu dùng NumPy để thực hiện các phép tính đại số tuyến tính cũng như xử lý mảng, cho phép người dùng sử dụng, huấn luyện và đánh giá model chỉ với vài dòng code. Thư viện này cung cấp 1 bộ API cao cấp nhất quán để tạo ra quy trình hoặc luồng công việc ML.

A diagram of a training model

Description automatically generated

*Hình 2.18 Quy trình*

Người dùng Pipeline để truyền dữ liệu qua transformers mục đích trích xuất đặc trưng và qua các estimators để xây dựng model và đánh giá những dự đoán nhằm đo lường tính đúng đắn của model.

* **Transformer:** Thuật toán biến đổi hoặc xử lý dữ liệu đầu vào.
* **Estimator:** Thuật toán học máy dùng để huấn luyện hoặc tinh chỉnh dữ liệu nhằm tạo ra model, sau đó có thể được dùng để dự đoán.
* **Pipeline:** Đường dẫn để nối các Transformer và Estimator lại để xác định quy trình làm việc của ML.

### **Thuật toán**

Chủ yếu vào mô hình hóa dữ liệu và không chú trọng vào việc truyền tải, tổng hợp hay biến đổi dữ liệu, các công việc này thuộc về các thư viện như Numpy và Pandas.

A graph of a number of clusters

Description automatically generated

*Hình 2.19 Thuật toán phân cụm trung bình*

1 số nhóm thuật toán mà Sklearn cung cấp:

* **Clustering:** Nhóm các thuật toán **phân cụm** dữ liệu không gán nhãn, ví dụ như thuật toán KMeans.
* **Cross Validation** (Kiểm thử chéo): Đánh giá hiệu quả của thuật toán học giám sát bằng cách dùng validation data trong quá trình huấn luyện model.
* **Datasets**: Bao gồm các **bộ dữ liệu** có sẵn, hầu hết đã được chuẩn hóa và mang lại năng suất cao trong quá trình train như iris, digit, v.v.
* **Dimensionality Reduction**: Thuật toán **giảm số lượng thuộc tính** quan trọng của dữ liệu bằng các phương pháp như biểu diễn, tổng hợp dữ liệu và lựa chọn đặc điểm, ví dụ: thuật toán PCA.
* **Ensemble methods** (Phương pháp tập hợp)**:** Dùng các thuật toán học tập để đạt được hiệu suất dự đoán tốt hơn so với các loại thuật toán khác.
* **Feature extraction** (Trích xuất đặc trưng): Định nghĩa các thuộc tính từ dữ liệu ảnh và dữ liệu chữ.
* **Feature selection** (Lựa chọn đặc trưng): Chọn các đặc điểm có ý nghĩa trong việc huấn luyện model học có giám sát.
* **Parameter Tuning** (Tinh chỉnh tham số): Những thuật toán phục vụ cho việc lựa chọn thông số hợp lí để tối ưu hóa model.
* **Manifold Learning**: Các thuật toán học tổng hợp và **phân tích dữ liệu đa chiều** phức tạp.
* **Supervised Models** (Học giám sát):  Bao gồm nhiều thuật toán học máy phổ biến hiện nay như mô hình tuyến tính, phân tích phân biệt, naive bayes, phương pháp lazy, mạng neural, máy vector hỗ trợ và cây quyết định.

### **Ưu và nhược điểm**

* **Ưu điểm:**

**- Dễ Dùng và Học:**

+ **GUI Thân Thiện**: Scikit-learn có giao diện API nhất quán và dễ hiểu, giúp lập trình viên nhanh chóng nắm bắt và dùng các thuật toán học máy.

+ **Tài Liệu Phong Phú**: Thư viện có tài liệu phong phú và chi tiết, bao gồm các ví dụ thực hành và hướng dẫn sử dụng, giúp người dùng dễ dàng học tập và áp dụng.

**- Đa Dạng Thuật Toán:**

+ **Đầy Đủ Các Thuật Toán Cơ Bản**: Scikit-learn cung cấp một bộ sưu tập phong phú các thuật toán học máy như phân loại, hồi quy, clustering, giảm chiều, và các phương pháp đánh giá mô hình.

+ **Hỗ Trợ Nhiều Phương Pháp Tiền Xử Lý Dữ Liệu**: Chuẩn hóa, tiêu chuẩn hóa, xử lý giá trị thiếu, và mã hóa nhãn.

- **Hiệu Suất Cao:**

+ **Tối Ưu Hóa Hiệu Suất**: Scikit-learn được xây dựng trên nền tảng NumPy, SciPy và Matplotlib, giúp thư viện đạt hiệu suất cao trong việc xử lý các tác vụ học máy.

+ **Tích Hợp Dễ Dàng Với Các Thư Viện Python Khác**: Dễ dàng kết hợp với các thư viện như Pandas để xử lý dữ liệu và TensorFlow hoặc PyTorch để học sâu.

**- Tính Linh Hoạt:**

+ **Tích Hợp Dễ Dàng**: Dễ dàng tích hợp vào các pipeline học máy và các hệ thống sản xuất khác.

+ **Khả Năng Mở Rộng**: Cho phép người dùng tùy chỉnh và mở rộng các thuật toán hiện có hoặc thêm các thuật toán mới.

* **Nhược điểm:**

- **Xử Lý Bộ Dữ Liệu Lớn:** Scikit-learn không được thiết kế để xử lý các bộ dữ liệu cực lớn, vì nó chủ yếu hoạt động trên bộ nhớ. Các thư viện như Dask-ML hoặc Spark MLlib có thể thích hợp hơn cho các bài toán cần xử lý dữ liệu lớn.

- **Thiếu Hỗ Trợ Học Sâu:** Scikit-learn không hỗ trợ tốt cho các mô hình học sâu phức tạp như Deep Neural Networks. Các thư viện như TensorFlow và PyTorch được sử dụng phổ biến hơn cho học sâu.

- **Hạn Chế Tùy Chỉnh Thuật Toán:** Mặc dù cung cấp nhiều thuật toán, nhưng khả năng tùy chỉnh chi tiết các thuật toán này có thể hạn chế so với việc viết mã từ đầu hoặc dùng các framework học máy khác.

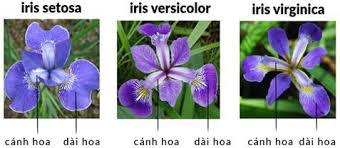
### **Ứng dụng**

Scikit-learn được sử dụng rộng rãi trong nhiều lĩnh vực và ứng dụng khác nhau, bao gồm:

* **Dự Đoán Kinh Doanh**: Dự đoán doanh thu, phân tích khách hàng, và phân loại rủi ro tài chính.
* **Y Tế**: Chẩn đoán bệnh, phân loại các mẫu y sinh, và dự đoán tiến triển bệnh.
* **Thương Mại Điện Tử**: Phân tích hành vi người tiêu dùng, gợi ý sản phẩm, và phát hiện gian lận.
* **Khoa Học Dữ Liệu**: Phân tích dữ liệu lớn, xây dựng mô hình dự đoán, và khám phá tri thức từ dữ liệu.

### **Ví dụ về bài toán cây phân loại và hồi quy**

#### **2.4.6.1 Xây dựng bài toán**

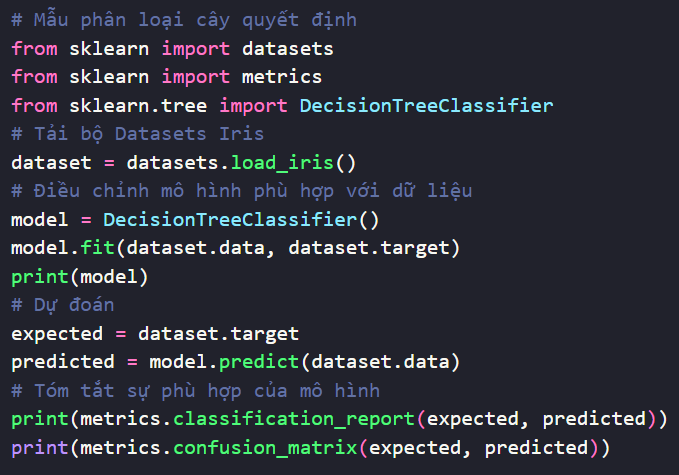
Dùng Decision Tree để phân loại và mô hình hóa Datasets hoa Iris:

*Hình 2.20 Iris Datasets*

#### **Chuẩn bị dữ liệu**

Datasets được cung cấp dưới dạng sample trong thư viện và có thể tải xuống. Thuật toán phân loại sẽ phù hợp với dữ liệu và thực hiện dự đoán trên tập train. Iris Datasets dùng để huấn luyện model ban đầu và dự đoán trên các tập huấn luyện.

#### **Xây dựng Model**

 Đánh giá hiệu quả của model bằng cách quan sát độ chính xác (accuracy) và ma trận nhầm lẫn (confusion matrix) của hai tập label: label thực tế và label dự đoán của model:

*Hình 2.21 Code ví dụ cho bài toán*

Ta có thể thấy các chi tiết về cách mà model Decision Tree được train với các thông số cụ thể, và mỗi thông số này ảnh hưởng đáng kể đến hiệu quả của model:

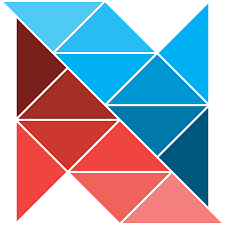
A computer screen shot of a program

Description automatically generated

*Hình 2.22 Classification Report và Confusion Matrix của mô hình*

## **TỔNG QUAN VỀ NSIS**

### **Giới thiệu**



*Hình 2.23 NSIS*

NSIS là một công cụ mã nguồn mở để tạo các trình cài đặt cho phần mềm trên hệ điều hành Windows. Được phát triển bởi Nullsoft, NSIS cho phép xây dựng các gói cài đặt nhỏ gọn và linh hoạt, với khả năng tùy chỉnh cao.

### **Các tính năng chính**

NSIS cung cấp một loạt các tính năng mạnh mẽ và linh hoạt để tạo trình cài đặt phần mềm:

* **Nhỏ Gọn và Nhanh Chóng**: NSIS tạo ra các trình cài đặt có kích thước nhỏ và có tốc độ cài đặt nhanh.
* **Tùy Chỉnh Cao**: Sử dụng ngôn ngữ kịch bản (scripting language) của NSIS, các nhà phát triển có thể tùy chỉnh gần như mọi khía cạnh của quá trình cài đặt.
* **Hỗ Trợ Đa Ngôn Ngữ**: NSIS hỗ trợ nhiều ngôn ngữ, cho phép tạo các trình cài đặt đa ngôn ngữ dễ dàng.
* **Mở Rộng và Plugin**: NSIS hỗ trợ các plugin, cho phép mở rộng các tính năng của trình cài đặt.
* **Khả Năng Tự Phục Hồi**: Trình cài đặt NSIS có thể tự kiểm tra và khắc phục các lỗi trong quá trình cài đặt.
* **Tích Hợp Gọn Nhẹ**: NSIS có thể tích hợp các tệp và thư viện cần thiết trực tiếp vào trình cài đặt, giúp giảm sự phụ thuộc vào các yếu tố bên ngoài.

### **Ưu và nhược điểm**

**Ưu Điểm**

* **Kích Thước Nhỏ**: Tạo ra các trình cài đặt có kích thước nhỏ gọn.
* **Linh Hoạt và Tùy Chỉnh Cao**: Cho phép tùy chỉnh hầu như mọi khía cạnh của quá trình cài đặt.
* **Miễn Phí và Mã Nguồn Mở**: NSIS là công cụ mã nguồn mở và miễn phí sử dụng.

**Nhược Điểm**

* **Độ Phức Tạp**: Có thể phức tạp với người mới do phải viết script.
* **Thiếu GUI Trong Việc Tạo Script**: Không có giao diện đồ họa mạnh mẽ để tạo script, hầu hết các thao tác phải thực hiện bằng cách viết mã.
* **Chủ Yếu Hỗ Trợ Windows**: NSIS chủ yếu hỗ trợ cài đặt trên Windows, không thích hợp cho các hệ điều hành khác.

# **Chương 3 XÂY DỰNG BÀI TOÁN PLANT CLASSIFICATION**

## **DỮ LIỆU ĐẦU VÀO**

Gồm 1002 ảnh được lấy nguồn từ Kaggle.

*Hình 3.1 Input ảnh định dạng jpg*

## **MỤC TIÊU**

Xây dựng một bộ phân loại tự động các loài hoa từ hình ảnh, giúp tiết kiệm thời gian và chi phí, cho thấy được tỉ lệ và xác suất thành công.

## **PHƯƠNG PHÁP**

### **Sử Dụng Histogram Màu**

Ta sử dụng histogram 3D RGB để mô tả màu sắc của cánh hoa. Mô hình này giúp phân loại các loài hoa dựa trên các đặc điểm màu sắc cụ thể của chúng. Các bước chính trong quá trình này bao gồm:

1. **Xây dựng mô tả ảnh**: tạo một lớp Python **RGBHistogram** để tính toán histogram màu từ ảnh hoa.

2. **Tạo histogram màu**: Histogram này được tính toán từ các kênh màu (R, G, B) của ảnh, sau đó được chuẩn hóa để đảm bảo tính nhất quán.

Định nghĩa lớp **RGBHistogram** để trích xuất histogram màu từ ảnh: Khởi tạo số lượng bins cho histogram.

A screenshot of a computer screen

Description automatically generated

*Hình 3.2 Hàm init cho lớp RGBHistogram*

Tính toán và chuẩn hóa histogram 3D trong không gian màu RGB, có thể sử dụng mặt nạ để tập trung vào một vùng cụ thể của ảnh. Kết quả được làm phẳng thành một vector:

A screenshot of a computer program

Description automatically generated

*Hình 3.3 Hàm Describe*

A black background with colorful text

Description automatically generated

*Hình 3.4 Phân tích các đối số dòng lệnh*

Sử dụng **argparse** để phân tích các đối số dòng lệnh:

* **-i, --images:** Đường dẫn đến tập ảnh.
* **-m, --masks:** Đường dẫn đến tập mặt nạ.
* **args:** Chứa các đối số đã phân tích dưới dạng từ điển.

### **Xây Dựng Bộ Phân Loại Học Máy**

Dùng 1 model máy học là **RandomForestClassifier** để phân loại các loài hoa. Các bước chính bao gồm:

1. **Chuẩn bị dữ liệu**: Lấy các đường dẫn của ảnh và mặt nạ (masks) từ thư mục.
2. **Xây dựng model**: Được huấn luyện trên các vector đặc trưng màu của ảnh hoa và được đánh giá trên một tập test validation.
3. **Phân loại và đánh giá**: Kết quả phân loại được đánh giá bằng cách so sánh dự đoán của model với các label thực tế.

Tìm và sắp xếp các đường dẫn tới các file ảnh và mặt nạ trong các thư mục được chỉ định:

A computer code with white text

Description automatically generated with medium confidence

*Hình 3.5 Đọc ảnh và mặt nạ*

Kiểm tra xem có ảnh và mặt nạ không. Nếu không, thông báo lỗi và thoát chương trình. Nếu có, in số lượng ảnh và mặt nạ tìm thấy:A screenshot of a computer code

Description automatically generated

*Hình 3.6 Kiểm tra ảnh và mặt nạ*

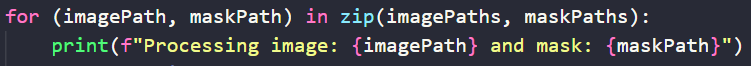
Khởi tạo các danh sách **data** và **target** để lưu trữ đặc trưng và nhãn lớp. Tạo đối tượng **RGBHistogram** với số lượng **bins** là **[8, 8, 8]:**

A blue background with white text

Description automatically generated

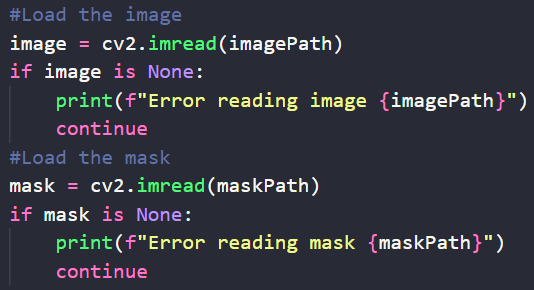
*Hình 3.7 Khởi tạo RGBHistogram với bins*

Xử lý từng cặp ảnh và mặt nạ:



*Hình 3.8 Trích xuất các đặc điểm và nhãn từ mỗi cặp hình ảnh và mặt nạ*

+ Đọc ảnh và mặt nạ từ đường dẫn, kiểm tra xem ảnh và mặt nạ có được đọc thành công không, nếu không thì bỏ qua:



*Hình 3.9 Đọc và kiểm tra*

+ Chuyển đổi mặt nạ sang ảnh xám:



*Hình 3.10 Chuyển mặt nạ sang ảnh xám*

+ Trích xuất các đặc trưng từ ảnh sử dụng **RGBHistogram** và lưu trữ vào danh sách **data**:

A close up of a sign

Description automatically generated

*Hình 3.11 Trích xuất các đặc điểm bằng RGBHistogram*

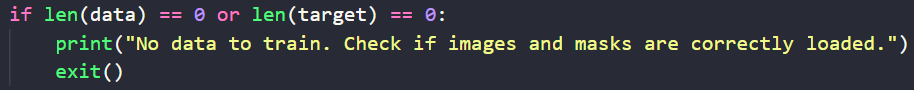
+ Trích xuất tên hoa từ tên file ảnh và lưu vào danh sách **target**:

A screenshot of a computer screen

Description automatically generated

*Hình 3.12 Trích xuất tên hoa*

+ Kiểm tra xem có dữ liệu để huấn luyện không. Nếu không, thông báo lỗi và thoát chương trình:



*Hình 3.13 Kiểm tra*

Mã hóa các nhãn lớp thành các số nguyên:

* **np.unique:** Lấy danh sách các nhãn lớp duy nhất.
* **LabelEncoder:** Khởi tạo đối tượng mã hóa nhãn.
* **fit\_transform:** Mã hóa nhãn lớp thành các số nguyên.

A computer code with green and blue text

Description automatically generated

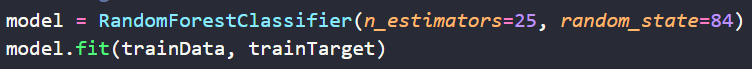
*Hình 3.14 Mã hóa nhãn*

Chia dữ liệu thành tập huấn luyện và kiểm tra với tỉ lệ 70/30, **random\_state=42** đảm bảo kết quả chia dữ liệu có thể tái lập:



*Hình 3.15 Chia tập dữ liệu*

Khởi tạo mô hình **RandomForestClassifier** với 25 cây quyết định và train model trên tập train:



*Hình 3.16 Huấn luyện mô hình*

Đánh giá mô hình trên tập test và in báo cáo phân loại, bao gồm các thông số như **precision, recall** và **f1-score:**



*Hình 3.17 Đánh giá mô hình*

Chọn ngẫu nhiên 10 ảnh từ tập dữ liệu:

* Đọc ảnh và mặt nạ tương ứng.
* Chuyển đổi mặt nạ sang ảnh xám.
* Trích xuất đặc trưng từ ảnh.
* Dự đoán loại hoa từ đặc trưng đã trích xuất.
* In kết quả dự đoán và hiển thị ảnh.

A computer screen shot of text

Description automatically generated

*Hình 3.18 Dự đoán và hiển thị 10 hình ảnh ngẫu nhiên từ tập test*

Kết quả: Mô hình đã dữ đoán đúng và đạt được độ chính xác cao trong việc phân loại các loài hoa.

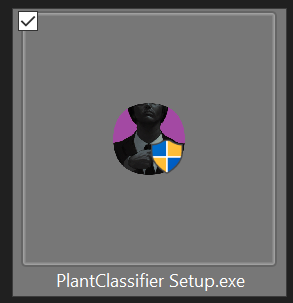
A screenshot of a computer

Description automatically generated

*Hình 3.19 Output*

## **CÀI ĐẶT VÀ DEMO**

Chạy File cài đặt:



*Hình 3.20 File cài đặt*

Sau khi chạy file cài đặt thì màn hình cài đặt xuất hiện ta ấn **Next:**

A screenshot of a computer

Description automatically generated

*Hình 3.21 Màn hình cài đặt*

Sau khi ấn Next thì ta ấn **I Agree**:

A screenshot of a computer

Description automatically generated

*Hình 3.22 Màn hình cài đặt (1)*

Sau khi ấn I Agree ta tiếp tục ấn **Next**:

A screenshot of a computer

Description automatically generated

*Hình 3.23 Màn hình cài đặt (2)*

Sau khi ấn Next thì ta ấn **Install** để cài vào máy:

A screenshot of a computer

Description automatically generated

*Hình 3.24 Màn hình cài đặt (3)*

Sau khi Install thì **Shortcut** sẽ xuất hiện ở màn hình Desktop:

A person in a suit and tie

Description automatically generated

*Hình 3.25 Shortcut*

Giao diện màn hình chính với 1 nơi hiện ảnh, 1 nơi dự đoán, 1 nơi hiện độ chính xác model, 1 nút chọn ảnh và 1 nút huấn luyện model:

A screenshot of a computer

Description automatically generated

*Hình 3.26 Giao diện chính*

Đầu tiên ta chọn **Train Model** và chọn đường dẫn ảnh và mặt nạ:

A screenshot of a computer

Description automatically generated

*Hình 3.27 Chọn đường dẫn cho việc train model*

Sau khi chọn xong sẽ thông báo đã train xong và hiện thị độ chính xác của model:

A screenshot of a computer

Description automatically generated

*Hình 3.28 Thông báo train xong và hiện độ chính xác*

Tiếp theo ta thử lấy hình trên Google:

A screenshot of a computer

Description automatically generated

*Hình 3.29 Lấy ảnh từ Google*

Ấn vào **Select Image** chọn ảnh ta vừa lưu về máy:A screenshot of a computer

Description automatically generated

*Hình 3.30 Chọn ảnh để dự đoán*

Kết quả: Dự đoán chính xác tên hoa.

A screenshot of a computer

Description automatically generated

*Hình 3.31 Dự đoán*

# **Chương 4: KẾT LUẬN**

## **4.1 ĐÁNH GIÁ ĐỀ TÀI**

### **4.1.1 Kết quả**

* Mô hình đã có thể dự đoán chính xác.
* Đã thành công trong việc sử dụng histogram màu và mô hình máy học để tự động hóa quá trình phân loại loài hoa.
* Chứng minh rằng các kỹ thuật thị giác máy và học máy có thể ứng dụng hiệu quả trong lĩnh vực sinh học và bảo tồn
* Đã tạo được một giao diện tương tác người dùng.
* Bổ sung và làm vững hơn nền tảng kiến thức về ngôn ngữ Python.

### **4.1.2 Khuyết điểm**

* Các chức năng còn đơn giản, chưa hoàn toàn tối ưu.
* Giao diện đơn giản, các chức năng còn ít.
* Chưa dự đoán được đa dạng các loại hoa khác nhau.

## **4.2 HƯỚNG HOÀN THIỆN**

* Phát triển dự đoán đa dạng các loại hoa: Các bước tiếp theo có thể bao gồm mở rộng bộ dữ liệu huấn luyện, cải thiện độ chính xác của mô hình và tích hợp thêm các đặc trưng khác như hình dạng và kết cấu của hoa.
* Phát triển toàn diện: Hướng phát triển này thì mô hình sẽ trở nên tốt nhất. Nhưng để hoàn thiện được thì việc bỏ ra thời gian, tài nguyên và nhân lực cũng sẽ rất lớn.

# **TÀI LIỆU THAM KHẢO**

[1] “OpenCV Documentation” [online]

<<https://docs.opencv.org/>>

[2] “Scikit-learn: Machine Learning in Python (2011)” [online]

< [https://jmlr.org/papers/v12/pedregosa11a.html/](%20https://jmlr.org/papers/v12/pedregosa11a.html/)>

[3] “Qt for Python” [online]

< [https://doc.qt.io/qtforpython-6/](%20https://doc.qt.io/qtforpython-6/)>

[4] Nguyễn Hải Triều (2022). Bài giảng Lập trình Python. Trường ĐH Nha Trang.

[5] “NSIS Users Manual” [online]

< [https://nsis.sourceforge.io/Docs/](%20https://nsis.sourceforge.io/Docs/)>

[6] Dr. Adrian Rosebrock (2016). Practical Python and OpenCV: Case Studies, 3rd Edition.