

TRƯỜNG ĐẠI HỌC BÁCH KHOA - ĐẠI HỌC ĐÀ NẴNG
KHOA ĐIỆN TỬ - VIỄN THÔNG



BÁO CÁO CUỐI KỲ
Học phần: Trí tuệ nhân tạo

ĐỀ TÀI:

Ứng dụng Machine Learning để phân loại lá cây

Giảng viên hướng dẫn:

Hoàng Lê Uyên Thục

Sinh viên thực hiện:

NHÓM 2

Họ và tên	MSSV	Email liên hệ
Lê Phạm Công	106200221	106200221@sv1.dut.udn.vn
Phan Công Danh	106200222	106200222@sv1.dut.udn.vn
Đoàn Thế Lân	106200233	106200233@sv1.dut.udn.vn
Lê Minh Nhật	106200238	106200238@sv1.dut.udn.vn

Đà Nẵng, ngày 31 tháng 05 năm 2023

Mục lục

Tóm tắt	2
1 Giới thiệu	2
2 Phương pháp	3
2.1 Đặc trưng HOG (Histogram of Oriented Gradients)	4
2.2 Đặc trưng HU's Moments	6
2.3 Support Vector Machine (SVM)	7
2.4 K-Nearest Neighbors (KNN)	10
3 Thực nghiệm	11
3.1 Cơ sở dữ liệu	11
3.2 Kịch bản thực hiện	12
3.3 Kết quả thực hiện	12
3.3.1 Kết quả trích xuất đặc trưng HOG	12
3.3.2 Kết quả trích xuất đặc trưng HU's Moments	14
3.3.3 Kết quả phân loại	15
4 Nhận xét	16
5 Kết luận	17
6 Phụ lục	18
Tài liệu tham khảo	19

Tóm tắt

Nghiên cứu này nhằm phân loại ba loại lá cây *Cercis chinensis*, *Indigofera tinctoria*, và *Acer palmatum* nằm trong bộ dữ liệu Flavia được công bố tại [1] bằng cách sử dụng hai kỹ thuật học máy, K-Nearest Neighbors (KNN) và Support Vector Machine (SVM), với hai loại đặc trưng khác nhau: Moment Invariants (HU's Moments) và Histogram of Oriented Gradients (HOG). Đặc trưng HU's Moments được sử dụng để nắm bắt các đặc trưng hình học cơ bản của lá cây, đảm bảo tính bất biến với phép quay, dịch chuyển và tỷ lệ, trong khi đặc trưng HOG được áp dụng để trích xuất các thông tin về hình dạng và cấu trúc thông qua phân bố của các gradient hướng trong hình ảnh.

Các kết quả thu được khi lần lượt sử dụng KNN và SVM trên từng loại đặc trưng HOG và HU's Moments rất khả quan. Với HOG, KNN cho kết quả accuracy là 94.64%, SVM là 100%. Với HU's Moments, KNN mang lại accuracy là 100%, SVM là 99.5%. Đồng thời các chỉ số khác là Precision, Recall, F1-Score đều rất cao mặc dù có sự chênh lệch nhỏ với từng mô hình và loại đặc trưng.

Mã nguồn của báo cáo có thể được tìm thấy tại https://github.com/lephamcong/Classification_Of_Leaves.git

1 Giới thiệu

Phân loại các loài thực vật dựa trên hình ảnh lá cây là một lĩnh vực nghiên cứu quan trọng trong khoa học máy tính và sinh học thực vật, với ứng dụng rộng rãi trong nhiều lĩnh vực như nông nghiệp, bảo tồn sinh học, và nghiên cứu môi trường. Sự phát triển của các kỹ thuật học máy đã mở ra nhiều cơ hội mới cho việc phân loại và nhận dạng thực vật một cách tự động và chính xác.

Một trong những phương pháp phổ biến để trích xuất đặc trưng từ hình ảnh lá cây là sử dụng Moment Invariants (HU's Moments). Các đặc trưng HU's Moments, được giới thiệu vào năm 1962 trong tài liệu [2], là một bộ các đặc trưng hình học bất biến với các phép biến đổi hình học như quay, dịch chuyển và tỷ lệ.

Một phương pháp khác để trích xuất đặc trưng là Histogram of Oriented Gradients (HOG), được phát triển và công bố trong tài liệu [3]. Đặc trưng HOG tập trung vào việc phân tích sự phân bố của các gradient hướng trong hình ảnh, giúp nắm bắt các chi tiết hình dạng và cấu trúc phức tạp.

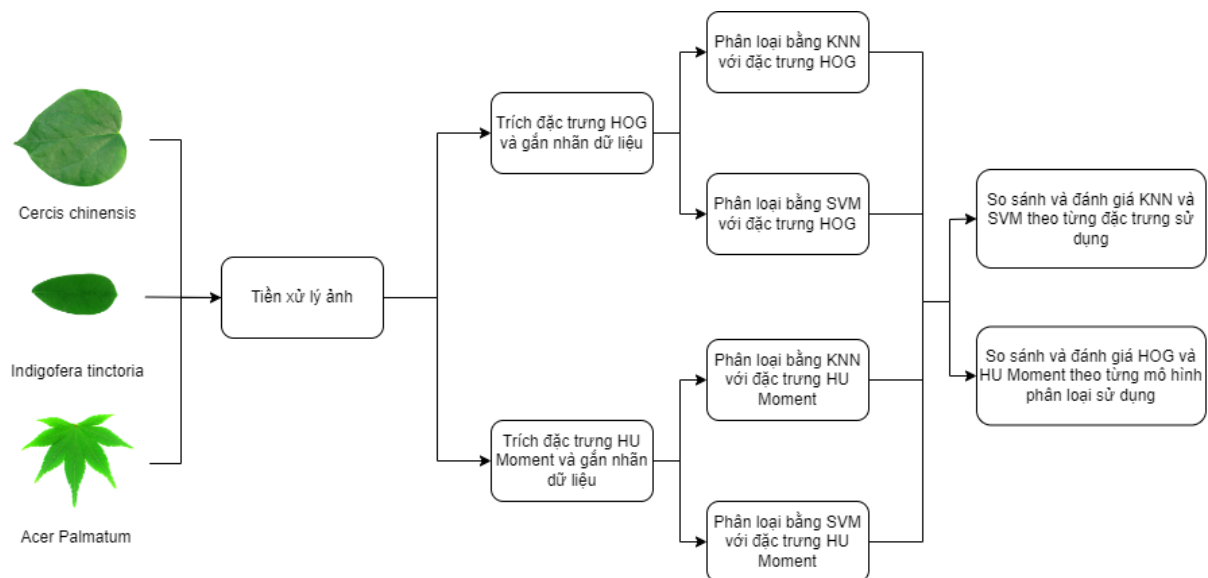
Các kỹ thuật học máy như K-Nearest Neighbors (KNN) và Support Vector Machine (SVM) đã được sử dụng rộng rãi trong phân loại hình ảnh. KNN là một phương pháp phân loại không tham số dựa trên khoảng cách giữa các điểm dữ liệu, trong khi SVM sử dụng các siêu phẳng để phân chia không gian đặc trưng và

phân loại dữ liệu.

Các tác giả của [4] đã so sánh hiệu quả của KNN và SVM trong việc phân loại lá nhỏ khỏe mạnh và không khỏe mạnh và phát hiện ra rằng đối với ảnh thời gian thực, KNN (với $K=1$) cho độ chính xác cao hơn so với SVM. Độ chính xác của hệ thống đề xuất trong [4] đạt được là 90% đối với bộ phân loại SVM và 96,66% đối với bộ phân loại KNN. Điều đó đã chứng tỏ sự hiệu quả của KNN và SVM trong giải quyết các bài toán phân loại.

Nghiên cứu này nhằm đánh giá hiệu quả của các kỹ thuật KNN và SVM trong việc phân loại ba loại lá cây: *Cercis chinensis*, *Indigofera tinctoria*, và *Acer palmatum*, sử dụng hai loại đặc trưng khác nhau: HU's Moments và HOG. Mục tiêu của nghiên cứu là so sánh hiệu suất của các phương pháp phân loại và xác định phương pháp tối ưu cho từng loại đặc trưng. Kết quả của nghiên cứu này không chỉ góp phần nâng cao hiểu biết về khả năng phân loại lá cây mà còn mở ra hướng đi mới cho việc ứng dụng học máy trong phân loại thực vật.

2 Phương pháp



Hình 1: Sơ đồ tổng quan

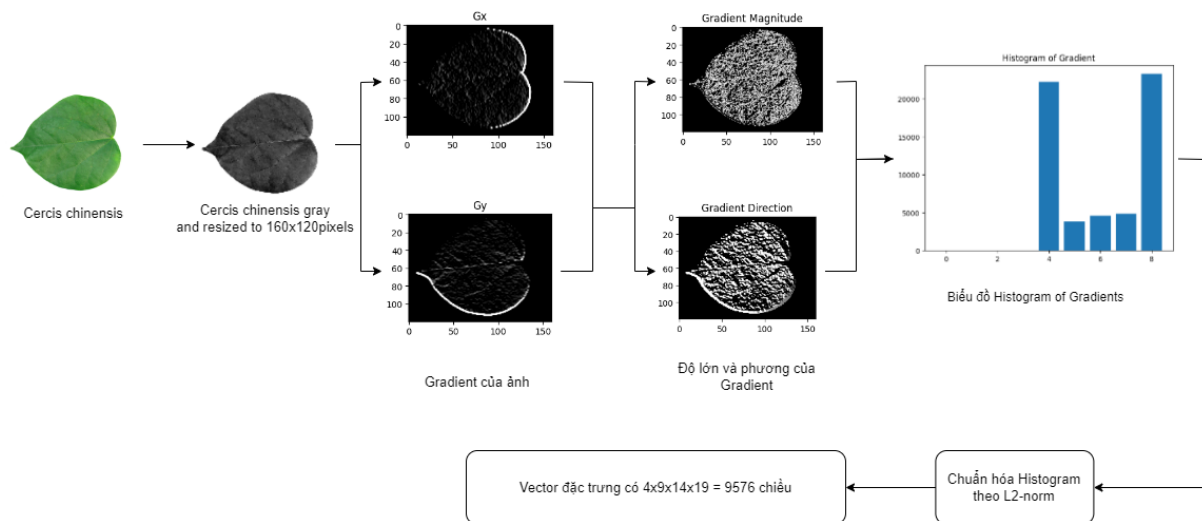
Hình 1 thể hiện tổng quan về phương pháp thực hiện trong nghiên cứu này. Dữ liệu được sử dụng là dữ liệu về 3 loại lá cây *Cercis chinensis* với 72 mẫu, *Indigofera tinctoria* với 73 mẫu và *Acer palmatum* với 56 mẫu được lấy từ bộ dữ liệu Flavia [1]. Dữ liệu được chuẩn bị là dữ liệu ảnh có kích thước 1600x1200 pixels, dữ liệu cần được tiến hành các bước tiền xử lý theo yêu cầu của quá trình trích xuất đặc trưng HOG và HU's Moment. Dữ liệu được trích xuất đặc trưng theo từng loại và

gắn nhãn theo quy ước: *Cercis chinensis* sẽ gắn nhãn là 4, *Indigofera tinctoria* gắn nhãn là 5 và *Acer Palmatum* sẽ gắn nhãn là 6, đây là quy ước riêng trong nghiên cứu này, nhãn này có thể tùy chọn theo yêu cầu của từng bài toán khác nhau. Các đặc trưng sau khi được trích xuất và gắn nhãn sẽ sử dụng làm dữ liệu cho các mô hình phân loại như Hình 1. Kết quả sẽ được tổng hợp và đánh giá theo từng mô hình phân loại và theo từng đặc trưng.

2.1 Đặc trưng HOG (Histogram of Oriented Gradients)

Histogram of Oriented Gradients (HOG) được đề cập trong [3] là một kỹ thuật phổ biến trong lĩnh vực thị giác máy tính và xử lý ảnh, được sử dụng để mô tả và phát hiện các đặc trưng quan trọng của các vật thể trong hình ảnh.

HOG dựa trên việc nhận dạng hình dạng của đối tượng cục bộ qua việc sử dụng hai ma trận quan trọng: ma trận độ lớn của gradient và ma trận hướng của gradient.



Hình 2: Sơ đồ quá trình trích xuất đặc trưng HOG

Hình 2 mô tả quá trình trích xuất đặc trưng HOG, chi tiết quá trình trích xuất đặc trưng như sau:

Bước 1: Ảnh đầu vào được chuyển sang dạng ảnh xám nhị phân và resize kích thước về 160x120pixels để giảm kích thước dữ liệu giúp đơn giản hóa quá trình xử lý và tập trung vào cấu trúc hình dạng của vật thể mà không bị ảnh hưởng bởi màu sắc. Việc thay đổi kích thước ảnh về 1 kích thước giúp thống nhất các đầu vào trong trường hợp dữ liệu chưa thống nhất.

Bước 2: Tính gradient của ảnh được thực hiện bằng cách sử dụng các mặt nạ lọc, mặt nạ 3x3 được sử dụng phổ biến. Mặt nạ lọc thông dụng là mặt

nạ Sobel 3x3 tính gradient theo hai hướng (x và y), và sau đó sử dụng các gradient này để tính toán các đặc trưng HOG.

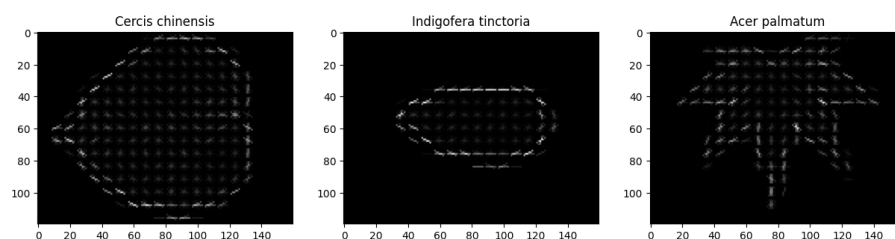
Bước 3: Biểu diễn đặc trưng của ảnh thông qua 2 thông số liên quan đến mức độ thay đổi cường độ màu sắc (gradient magnitude) và hướng thay đổi cường độ màu sắc (gradient direction).

Bước 4: Tạo một bộ mô tả có thể chuyển ảnh thành vector thể hiện thông tin của gradient magnitude và gradient direction. Một biểu đồ Histogram thống kê độ lớn Gradient sẽ được tính toán trên mỗi ô cục bộ. Kích thước mỗi ô là 8x8pixels, Histogram của gradient có 9 thanh, mỗi khối là 2x2 ô và phần chồng lấn giữa các khối là 1x1 ô.

Bước 5: Chuẩn hóa Histogram của Gradient bằng chuẩn L2-Hys (sự kết hợp giữa chuẩn L2 và kỹ thuật hysteresis để đảm bảo rằng các vector đặc trưng được chuẩn hóa tốt và tránh được các vấn đề liên quan đến giá trị gradient quá nhỏ hoặc quá lớn).

Bước 6: Xuất ra vector đặc trưng của ảnh. Trong phạm vi nghiên cứu này, ảnh đầu vào được resize thành 160x120 pixels, kích thước 1 ô là 8x8 pixels, kích thước khối là 2x2 ô, 9 giá trị/ 1 ô. Suy ra Vector đặc trưng có số chiều là [1, 9576].

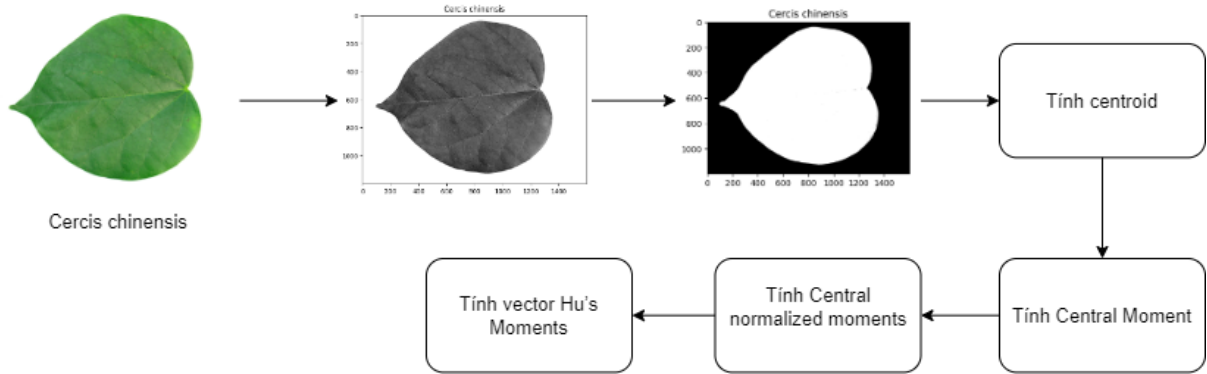
Hình 3 thể hiện vector đặc trưng HOG của 1 mẫu dữ liệu trong mỗi loại lá cây.



Hình 3: vector đặc trưng HOG của 3 loại lá cây

2.2 Đặc trưng HU's Moments

Hu's moments (Hu's moment invariants) là một tập hợp gồm 7 số được tính bằng cách sử dụng khoảng khắc trung tâm (central moments) bất biến đối với các phép biến đổi hình ảnh. 6 khoảng khắc đầu tiên đã được chứng minh là bất biến đối với chuyển động tịnh tiến, tỷ lệ, xoay và phép lật. Trong khi dấu hiệu của khoảng khắc thứ 7 thay đổi để phản ánh hình ảnh.



Hình 4: Sơ đồ quá trình trích xuất đặc trưng HU's Moments

Để tính được HU's Moment ảnh cần được chuẩn hóa về cùng một kích thước ảnh (nghiên cứu này sử dụng kích thước 1600x1200 pixels để tính HU's Moments) và chuyển sang dạng ảnh nhị phân với giá 0 cho nền và giá trị 1 cho đối tượng. Quá trình tính HU's Moment được mô tả như Hình 4.

Cách tính Hu's moments: [2]

Bước 1. Tính Centroid

$$\bar{x} = \frac{\sum_x \sum_y x \cdot s(x, y)}{\sum_x \sum_y s(x, y)}, \quad \bar{y} = \frac{\sum_x \sum_y y \cdot s(x, y)}{\sum_x \sum_y s(x, y)} \quad (1)$$

Bước 2. Tính Central Moment

$$m_{pq} = \sum_x \sum_y (x - \bar{x})^p (y - \bar{y})^q s(x, y), \quad p, q = 0, 1, 2, 3, \dots \quad (2)$$

Bước 3. Tính Central Normalized Moments

$$M_{pq} = \frac{m_{pq}}{m_{00}^{\left(\frac{p+q}{2}+1\right)}} \quad (3)$$

Bước 4. Tính vector HU's Moments

$$S_1 = M_{20} + M_{02} \quad (4)$$

$$S_2 = (M_{20} - M_{02})^2 + 4M_{11}M_{11} \quad (5)$$

$$S_3 = (M_{30} - 3M_{12})^2 + (3M_{21} - M_{03})^2 \quad (6)$$

$$S_4 = (M_{30} + M_{12})^2 + (M_{21} + M_{03})^2 \quad (7)$$

$$S_5 = (M_{30} - 3M_{12})(M_{30} + M_{12})[(M_{30} + M_{12})^2 - 3(M_{21} + M_{03})^2] \\ + (3M_{21} - M_{03})(M_{21} + M_{03})[3(M_{30} + M_{12})^2 - (M_{21} + M_{03})^2] \quad (8)$$

$$S_6 = (M_{20} - M_{02})[(M_{30} + M_{12})^2 - (M_{21} + M_{03})^2] \\ + 4M_{11}(M_{30} + M_{12})(M_{21} + M_{03}) \quad (9)$$

$$S_7 = (3M_{21} - M_{03})(M_{30} + M_{12})[(M_{30} + M_{12})^2 - 3(M_{21} + M_{03})^2] \\ - (M_{30} - 3M_{12})(M_{21} + M_{03})[3(M_{30} + M_{12})^2 - (M_{21} + M_{03})^2] \quad (10)$$

2.3 Support Vector Machine (SVM)

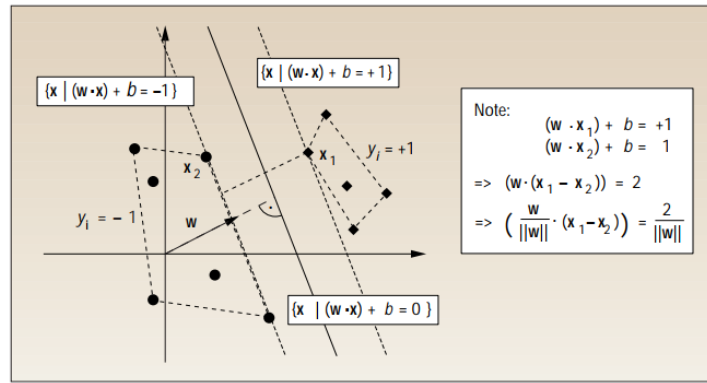
Các vector đặc trưng HOG và HU's Moments lần lượt được đưa vào bộ phân loại SVM được đề cập trong [5], mô hình này phân tách dữ liệu thành các lớp thông qua việc tối đa hóa khoảng cách giữa các điểm dữ liệu gần nhất của mỗi lớp, được gọi là vector hỗ trợ, và mặt phẳng quyết định. SVM là một công cụ phân loại phổ biến được sử dụng trong nhận dạng mẫu và các vấn đề phân loại khác.

SVM sử dụng một siêu mặt phẳng để phân tách một mẫu huấn luyện như công thức 11 [5] tương ứng với hàm quyết định như công thức 12 [5]

$$(\mathbf{w} \cdot \mathbf{x}) + b = 0 \quad w \in R^N, b \in R \quad (11)$$

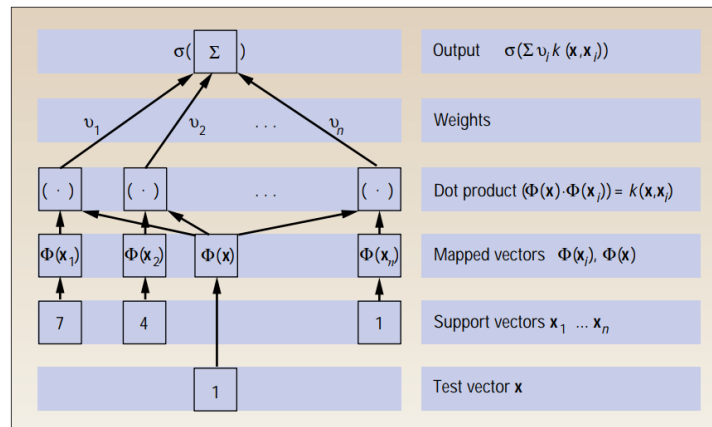
$$f(x) = \text{sgn}(\mathbf{w} \cdot \mathbf{x} + b) \quad (12)$$

Trong đó \mathbf{w} là vector trọng số, \mathbf{x} là vector đầu vào và b là hệ số điều chỉnh.



Hình 5: Minh họa về bài toán phân loại với SVM [5]

Hình 5 là minh họa về một bài toán phân loại với svm. Mặt phẳng tối ưu vuông góc với đường ngắn nhất nối các vỏ lõi của hai lớp dữ liệu (được chấm) và cắt nó ở chính giữa. Có một vector trọng số w và một ngưỡng b sao cho $y_i \cdot ((\mathbf{w} \cdot \mathbf{x}) + b) > 0$. Khi tái tỉ lệ w và b sao cho điểm gần nhất với mặt phẳng thỏa mãn $(\mathbf{w} \cdot \mathbf{x}) + b = 1$, nhận được một dạng (w, b) của mặt phẳng với $(\mathbf{w} \cdot \mathbf{x}) + b \geq 1$. Lưu ý rằng lẽ, đo vuông góc với mặt phẳng, bằng $\frac{2}{\|\mathbf{w}\|}$. Để tối đa hóa lẽ phải tối thiểu hóa $\|\mathbf{w}\|$ với điều kiện $y_i((\mathbf{w} \cdot \mathbf{x}) + b) \geq 1$.



Hình 6: Kiến trúc của svm [5]

Hình 6 mô tả kiến trúc SVM, bao gồm các bước từ việc nhập vector đầu vào đến việc tính toán đầu ra, là lớp dự đoán:

- 1) Các vector đầu vào được biến đổi thông qua hàm kernel để tạo ra một đặc trưng không gian nhiều chiều.
- 2) Các tích vô hướng giữa các vector đặc trưng và vector trọng số được tính toán.
- 3) Các giá trị này sau đó được tổng hợp cùng với hệ số b để tạo ra đầu ra cuối cùng, là lớp dự đoán của dữ liệu đầu vào.

5) Công thức tính toán đầu ra là:

$$\text{Output} = \text{sgn} \left(\sum_i \alpha_i y_i k(x_i, x) + b \right) \quad (13)$$

Trong đó:

Output: Đây là đầu ra của mô hình SVM, quyết định lớp của dữ liệu đầu vào.

$\text{sgn}()$: Là hàm dấu, trả về +1 hoặc -1 dựa trên dấu của đối số bên trong.

α_i : Là các hệ số Lagrange tối ưu hóa tìm được trong quá trình huấn luyện SVM. Các hệ số này không âm và phần lớn sẽ bằng không. Chỉ các vector hỗ trợ mới có hệ số α_i khác không.

y_i : Là nhãn của vector hỗ trợ i

$k(x_i, x)$: Là hàm kernel, đo lường sự tương đồng giữa vector đầu vào x và các vector hỗ trợ x_i .

b : hệ số điều chỉnh

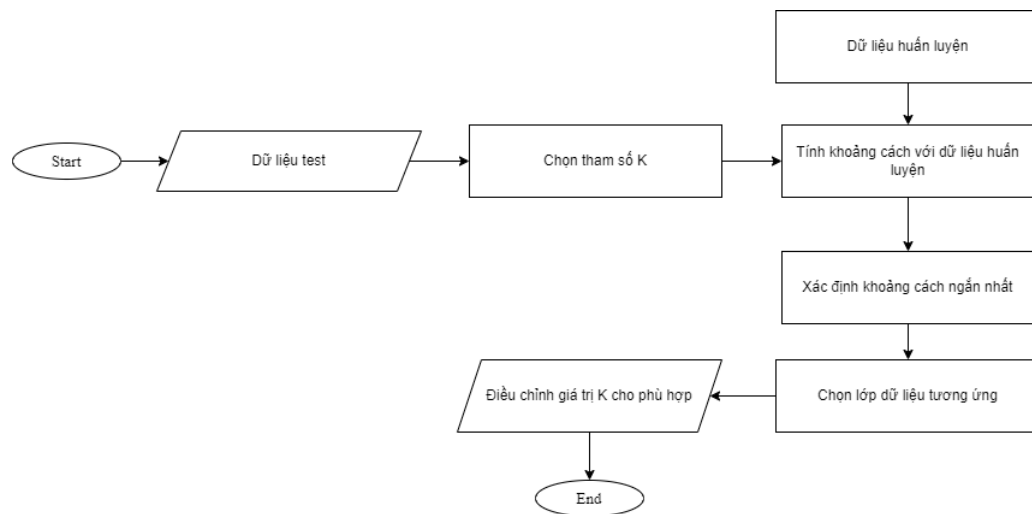
Trong bài toán phân loại đa lớp, việc sử dụng SVM cần thực hiện việc tối ưu hóa, mục tiêu của quá trình tối ưu hóa là tìm ra mặt phẳng (hoặc siêu mặt phẳng trong không gian nhiều chiều) phân chia các lớp dữ liệu với khoảng cách lớn nhất có thể tới các điểm dữ liệu gần nhất (các vector hỗ trợ). Hàm mục tiêu thường được biểu diễn như công thức 14:

$$\min_{\mathbf{w}, b} \left(\frac{1}{2} \|\mathbf{w}\|^2 + C \sum_{i=1}^n \xi_i \right) \quad (14)$$

Trong đó:

- $\min_{\mathbf{w}, b}$: Biểu thị rằng chúng ta đang tìm giá trị nhỏ nhất của hàm mục tiêu với các biến là \mathbf{w} và b
- $\frac{1}{2} \|\mathbf{w}\|^2$: Phần đầu tiên của hàm mục tiêu, tối thiểu hóa bình phương độ lớn của vector trọng số, giúp tối ưu đường biên giới giữa các lớp. Điều này phản ánh một phần của nguyên tắc lề tối đa mà SVM cố gắng đạt được.
- $C \sum_{i=1}^n \xi_i$: Phần thứ hai của hàm mục tiêu, thêm một thuật ngữ phạt cho mỗi lỗi phân loại (hoặc vi phạm lề). Các biến ξ_i được gọi là biến slack, cho phép các điểm dữ liệu vi phạm lề quyết định tới mức độ nhất định. C là tham số điều chỉnh mà bằng cách tăng hoặc giảm nó, người dùng có thể điều chỉnh sự cân bằng giữa việc giữ cho siêu mặt phẳng càng đơn giản càng tốt và tối đa hóa số lượng điểm dữ liệu được phân loại chính xác.

2.4 K-Nearest Neighbors (KNN)



Hình 7: Sơ đồ thuật toán KNN [6]

K-Nearest Neighbors (KNN) là một thuật toán học máy phổ biến và đơn giản, thuộc nhóm học có giám sát. Nó được sử dụng rộng rãi cho cả bài toán phân loại (classification) và hồi quy (regression). Đặc điểm nổi bật của KNN là việc không cần một mô hình cụ thể cho việc huấn luyện, mà dựa trên lưu trữ và so sánh trực tiếp với dữ liệu mẫu (dữ liệu huấn luyện). Hình 7 mô tả quá trình thực hiện của thuật toán như sau:

Bước 1. Chuẩn bị dữ liệu để huấn luyện và đánh giá

Bước 2. Chọn số lượng (K) các điểm dữ liệu gần nhất mà thuật toán sẽ xem xét khi dự đoán nhãn cho một điểm dữ liệu mới. Trong nghiên cứu này chọn $K=3$ là một giá trị được sử dụng phổ biến trong các ứng dụng phân loại của KNN.

Bước 3. Tính toán khoảng cách giữa điểm dữ liệu mới và các điểm dữ liệu trong tập huấn luyện bằng cách sử dụng một phép đo khoảng cách Euclidian công thức sau:

$$d_t = \sqrt{\sum_{i=1}^p |x_{2i} - x_{1i}|^2} \quad (15)$$

Trong đó:

x_{1i} : dữ liệu huấn luyện

x_{2i} : dữ liệu test

d_t : khoảng cách Euclidian

Bước 4. Chọn ra k điểm dữ liệu gần nhất với điểm dữ liệu mới. Dựa trên nhãn của các điểm dữ liệu, dự đoán nhãn cho điểm dữ liệu mới bằng cách sử dụng trọng số của các nhãn. Chọn nhãn xuất hiện nhiều nhất trong các điểm dữ liệu gần nhất dựa trên khoảng cách.

Bước 5. Đánh giá hiệu suất của mô hình bằng cách sử dụng các phương pháp đánh giá. Tùy chỉnh tham số K để cải thiện hiệu suất của mô hình.

3 Thực nghiệm

Quá trình thực nghiệm được thực hiện như sau: Trích đặc trưng HU's Moment và HOG sau khi đã thực hiện tiền xử lý theo yêu cầu của từng loại đặc trưng được mô tả trong mục 2.1 và 2.2. Sau khi trích đặc trưng, tiến hành bước phân loại với KNN và SVM, đánh giá bằng phương pháp đánh giá chéo với tỷ lệ 1:5 (five-fold cross validation). Môi trường thực nghiệm:

Ngôn ngữ lập trình: Python phiên bản 3.11.2

Cấu hình xử lý: Intel core i7-5600U@2.6GHz (4CPUs), RAM 12GB

Thư viện được sử dụng:

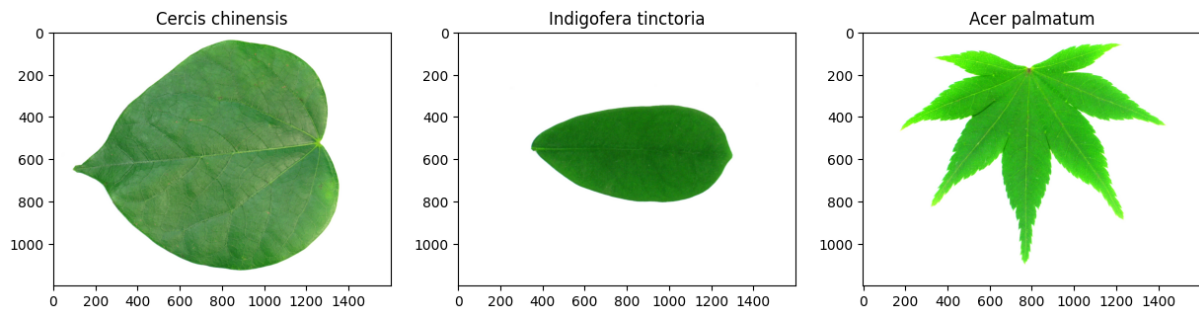
```
opencv-python==4.9.0
numpy==1.26.4
matplotlib==3.7.1
pandas==1.5.3
scikit-image==0.22.0
seaborn==0.13.2
scikit-learn==1.3.2
```

3.1 Cơ sở dữ liệu

Nghiên cứu này thực hiện với bộ dữ liệu về 3 loại lá cây *Cercis chinensis* với 72 mẫu, *Indigofera tinctoria* với 73 mẫu và *Acer palmatum* với 56 mẫu được lấy từ bộ dữ liệu Flavia [1].

Mỗi loại lá cây được gán nhãn theo số thứ tự của nó trong bộ dữ liệu Flavia. Cụ thể: *Cercis chinensis* được gán nhãn là 4; *Indigofera tinctoria* gán nhãn là 5; *Acer palmatum* được gán nhãn là 6.

Đường dẫn đến cơ sở dữ liệu đã tổng hợp tại đây: https://github.com/lephamcong/Classification_Of_Leaves/tree/main/Leaf_nhom_02



Hình 8: Đại diện của 3 loại lá cây trong cơ sở dữ liệu

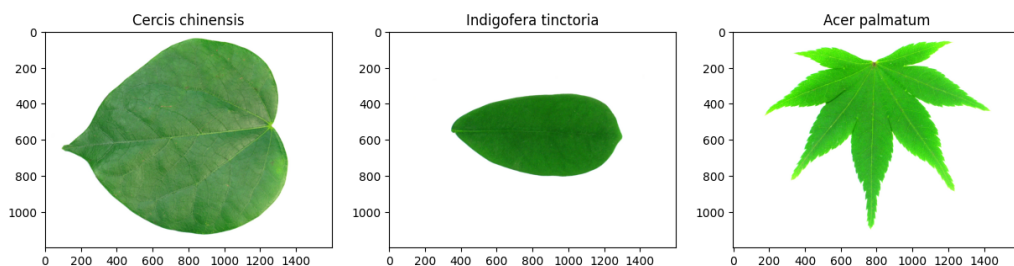
3.2 Kịch bản thực hiện

Dữ liệu được chọn ra từ bộ dữ liệu Flavia, dữ liệu gồm 3 loại lá cây *Cercis chinensis* với 72 mẫu, *Indigofera tinctoria* với 73 mẫu và *Acer palmatum* với 56 mẫu. Dữ liệu sau khi tổng hợp sẽ tiến hành tiền xử lý, trích xuất đặc trưng, các đặc trưng được tổng hợp vào file csv. Sau khi đã có các đặc trưng thì tiến hành phân loại với KNN và SVM theo từng loại đặc trưng bao gồm HOG và HU's Moments.

3.3 Kết quả thực hiện

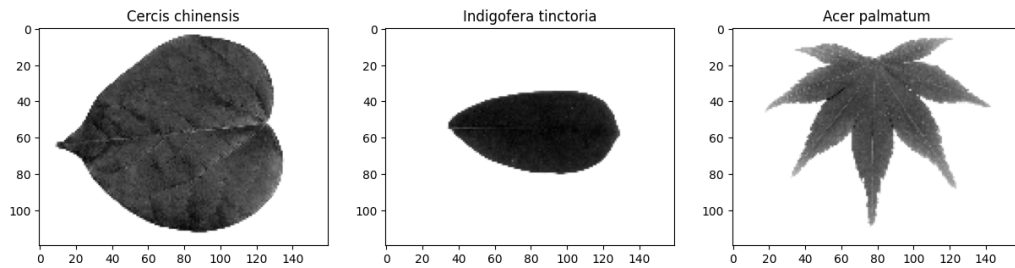
Các kết quả thực hiện được mô tả trong các mục 3.3.1, 3.3.2 và 3.3.3.

3.3.1 Kết quả trích xuất đặc trưng HOG



Hình 9: Hình ảnh đại diện cho mỗi loại lá cây

Hình 9 là 3 mẫu dữ liệu đại diện cho mỗi loại lá cây trong tập dữ liệu. Dữ liệu sẽ được tiến hành tiền xử lý, với yêu cầu của đặc trưng HOG được sử dụng trong nghiên cứu này, ảnh sẽ được chuyển sang dạng ảnh xám, và kích thước ảnh giảm đi 10 lần, từ ảnh gốc có kích thước 1600x1200x3 pixels thu được ảnh có kích thước 160x120x1 pixels (Kết quả thu được như Hình 10).



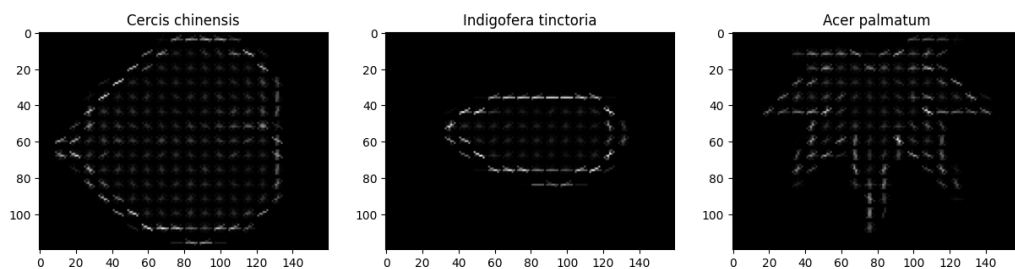
Hình 10: Dữ liệu sau khi thực hiện bước tiền xử lý

Việc chuyển ảnh sang dạng ảnh xám giúp giảm độ phức tạp bằng cách loại bỏ thông tin màu sắc và tập trung vào độ tương phản, từ đó nổi bật các đặc điểm hình học và kết cấu của đối tượng. Thay đổi kích thước ảnh sẽ đảm bảo sự nhất quán của các đặc trưng được trích xuất trên tất cả các mẫu và cải thiện hiệu suất tính toán bằng cách giảm bớt số lượng tính toán không cần thiết.

Bảng 1: Bảng tham số hàm tính đặc trưng HOG

Tham số	Giá trị
kernel	bộ lọc Sobel 3x3
orientations	9
pixels_per_cell	(8,8)
cells_per_block	(2,2)
block_norm	L2-Hys
transform_sqrt	True

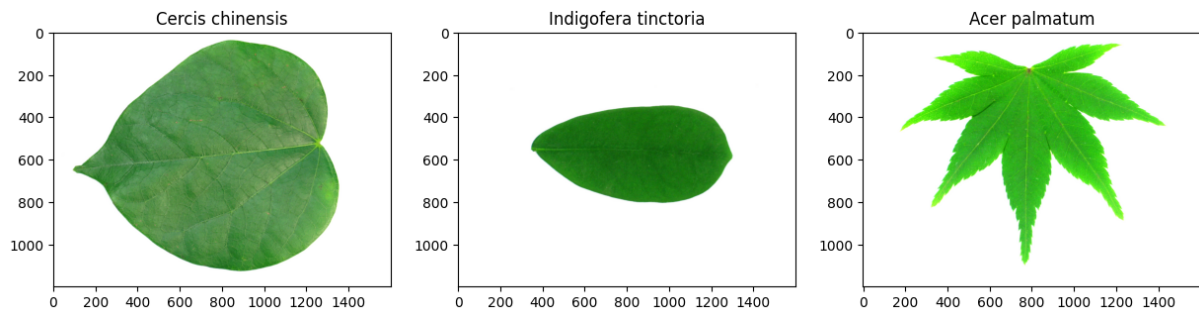
Bảng 1 thiết lập các tham số trong hàm HOG được hỗ trợ bởi thư viện scikit-image, ngoài một số tham số được đề cập trong mục 2.1 có một tham số được thiết lập để tối ưu các đặc trưng.



Hình 11: Kết quả thu được khi visualize vector HOG lên ảnh

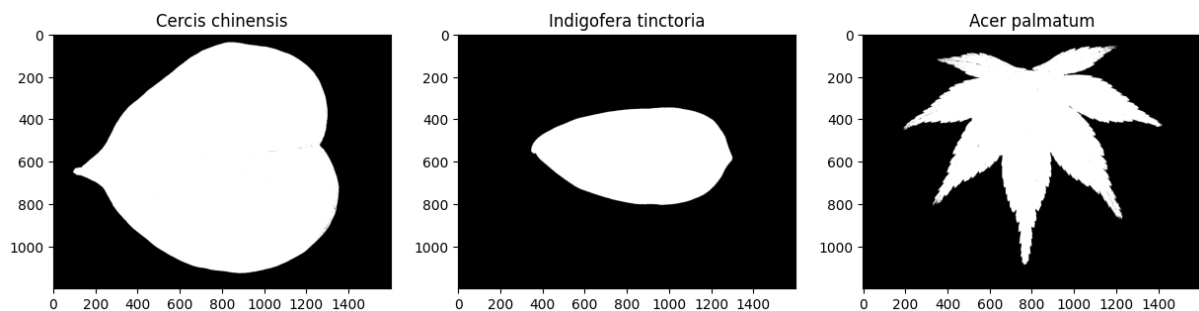
Kết quả vector đặc trưng thu được là vector có kích thước (1, 9576). Các vector đặc trưng này được lưu vào file csv và gán nhãn theo quy ước.

3.3.2 Kết quả trích xuất đặc trưng HU's Moments



Hình 12: Hình ảnh đại diện cho mỗi loại lá cây

Hình 12 là 3 mẫu dữ liệu đại diện cho mỗi loại lá cây trong tập dữ liệu. Dữ liệu sẽ được tiến hành tiền xử lý, với yêu cầu của đặc trưng HU được sử dụng trong nghiên cứu này, ảnh sẽ được chuyển sang dạng ảnh nhị phân với giá trị 0 cho nền và giá trị 1 cho đối tượng (lá), kích thước ảnh thay đổi ảnh gốc có kích thước 1600x1200x3 pixels thu được ảnh có kích thước 1600x1200x1 pixels (Kết quả thu được như Hình 13).



Hình 13: Kết quả tiền xử lý ảnh

Quá trình tính toán vector đặc trưng HU's Moments sử dụng thư viện OpenCV với hàm moments để tính moment của ảnh bao gồm moments và centroid, hàm HuMoments để tính toán Hu's Moments từ các moment đã tìm được. Hu's Moments được trả về dưới dạng mảng 2 chiều và sẽ được làm phẳng để chuyển thành dạng 1 chiều. Kết quả thu được là vector HU's Moments có kích thước (1, 7) và được gán nhãn theo quy ước.

3.3.3 Kết quả phân loại

Bảng 2: Kết quả phân loại của KNN và SVM sử dụng đặc trưng HOG (đơn vị: %)

	Loại lá cây	Accuracy	Precision	Recall	F1-Score
KNN	Cercis chinensis	100	96.28	100	98.08
	Indigofera tinctoria	100	93.2	100	96.44
	Acer palmatum	83.93	100	82.55	90.17
	Trung bình	94.64	96.49	94.18	94.9
SVM	Cercis chinensis	100	100	100	100
	Indigofera tinctoria	100	100	100	100
	Acer palmatum	100	100	100	100
	Trung bình	100	100	100	100

Bảng 3: Kết quả phân loại của KNN và SVM sử dụng đặc trưng HU's Moments (đơn vị: %)

	Loại lá cây	Accuracy	Precision	Recall	F1-Score
KNN	Cercis chinensis	100	100	100	100
	Indigofera tinctoria	100	100	100	100
	Acer palmatum	100	100	100	100
	Trung bình	100	100	100	100
SVM	Cercis chinensis	100	98.57	100	99.26
	Indigofera tinctoria	98.63	100	98.57	99.26
	Acer palmatum	100	100	100	100
	Trung bình	99.5	99.52	99.52	99.51

Bảng 2 và bảng 3 tổng hợp các kết quả thu được bao gồm chỉ số accuracy, precision, recall, F1-score khi thực hiện đánh giá mô hình KNN và SVM với 2 loại đặc trưng là HOG và HU's Moments theo phương pháp đánh giá chéo với tỷ lệ 1:5 (five-fold cross validation).

Thiết lập với KNN: KNN sử dụng số neighbor là 3 ($K=3$), đây là chỉ số được sử dụng trong nhiều nghiên cứu sử dụng KNN giải quyết bài toán phân loại, phép đo khoảng cách được sử dụng là Euclidian. Tham số K và phép đo khoảng cách được sử dụng cho cả 2 đặc trưng HOG và HU's Moments.

Thiết lập với SVM: Nhằm mục đích tối ưu các tham số để mang lại kết quả tốt nhất khi sử dụng SVM với mỗi loại đặc trưng, nghiên cứu này đã sử dụng

một kỹ thuật có tên “GridSearchCV” trong thư viện scikit-learn của Python, được sử dụng để tìm kiếm các tham số tối ưu cho một mô hình học máy. Kỹ thuật này thuộc về lĩnh vực học có giám sát và là một phần của quá trình điều chỉnh siêu tham số (hyperparameter tuning). Các tham số được sử dụng để thực hiện quá trình điều chỉnh như sau:

‘kernel’: [‘rbf’, ‘linear’]

‘C’: [0.0001, 0.001, 0.005, 0.1, 0.5, 1, 10, 100]

‘gamma’: [0.0000001, 0.000001, 0.00001, 0.0001, 0.001, 0.005, 0.01, 0.1, 1]

Sau khi thực hiện quá trình điều chỉnh và đánh giá từng bộ tham số và lựa chọn bộ tham số tối ưu nhất theo từng đặc trưng mang lại kết quả như sau:

Với đặc trưng *HOG*: bộ tham số ‘C’: 0.005, ‘gamma’: 1e-07, ‘kernel’: ‘linear’ mang lại kết quả tốt nhất với accuracy là 100%

Với đặc trưng *HU’s Moments*: ‘C’: 100, ‘gamma’: 1, ‘kernel’: ‘rbf’ mang lại kết quả tốt nhất với accuracy là 99.38%

4 Nhận xét

Các kết quả thu được như bảng 2 và bảng 3. Các nhận xét và so sánh về từng loại đặc trưng và từng loại mô hình Machine Learning như sau:

1) So sánh KNN và SVM theo từng loại đặc trưng:

Đặc trưng *HOG*:

KNN: Thuật toán KNN cho thấy hiệu suất cao với điểm F1 trung bình đạt 94.9%, trong đó hiệu suất cho loại lá *Acer palmatum* thấp hơn đáng kể so với các loại lá khác.

SVM: Thuật toán SVM cho thấy hiệu suất ổn định hơn với mọi loại lá, với điểm F1 trung bình đạt 100%. Điều này cho thấy SVM có khả năng phân loại ổn định và chính xác hơn KNN khi sử dụng đặc trưng *HOG*, đặc biệt là với lá *Acer palmatum*.

Đặc trưng *HU’s Moments*:

KNN: KNN tiếp tục cho thấy hiệu suất tối ưu với điểm F1 trung bình là 100%, cho thấy khả năng phân loại rất tốt với loại đặc trưng này.

SVM: Tương tự như KNN, SVM cũng cho thấy hiệu suất cao với điểm F1 trung bình 99.51%. Tuy nhiên, hiệu suất cho *Indigofera tinctoria* là 99.26%, chỉ thấp hơn một chút so với KNN.

2) So sánh HOG và HU's Moments theo từng mô hình machine learning:

KNN

Đặc trưng HU's Moments cho thấy hiệu suất nhỉnh hơn so với HOG. Điều này có thể do đặc trưng HU's Moments cung cấp thông tin hình học phức tạp hơn, giúp KNN phân loại chính xác hơn.

SVM

Với SVM, cả hai loại đặc trưng đều cho thấy hiệu suất cao, nhưng HU's Moments có điểm F1 trung bình cao hơn một chút so với HOG. Điều này cho thấy sự ưu việt của HU's Moments trong việc cung cấp các đặc điểm có thể giúp SVM phân tích tốt hơn.

Nhìn chung, SVM nhìn chung có hiệu suất cao hơn KNN khi sử dụng cùng một loại đặc trưng, đặc biệt là với đặc trưng HOG. Đặc trưng HU's Moments tỏ ra hiệu quả hơn HOG trong cả hai mô hình KNN và SVM, cho thấy tính ưu việt trong việc cải thiện khả năng phân loại của các mô hình học máy. Tuy nhiên, về mặt thời gian xử lý 1 mẫu dữ liệu (1 vector đặc trưng) để cho ra kết quả dự đoán, với đặc trưng HU's Moments thì SVM cho thời gian xử lý là 1.998ms, nhanh hơn khoảng 2 lần so với KNN là 3.999ms; với đặc trưng HOG, SVM cho thời gian xử lý là 105.995ms, nhanh hơn khoảng 2.76 lần so với KNN là 293.005ms. SVM nhanh hơn KNN trong cả hai loại đặc trưng, nhưng sự khác biệt lớn nhất nằm ở đặc trưng HOG, nơi mà SVM xử lý nhanh hơn KNN gần 3 lần. Điều này làm cho SVM trở thành lựa chọn tối ưu hơn nếu tốc độ là yếu tố quan trọng.

SVM với HU's Moments là lựa chọn tối ưu nhất khi cân bằng giữa hiệu suất và thời gian xử lý, đặc biệt khi tài nguyên phần cứng đủ mạnh để xử lý nhanh. Đây là sự lựa chọn tốt cho các ứng dụng cần đến phân loại chính xác và nhanh chóng. SVM với HOG cũng là một lựa chọn tốt nếu ưu tiên hiệu suất phân loại cao hơn, mặc dù thời gian xử lý lâu hơn so với HU's Moments nhưng vẫn nhanh hơn đáng kể so với KNN.

5 Kết luận

Trong đề tài nghiên cứu này, chúng tôi đã khám phá và đánh giá hiệu quả của hai thuật toán học máy phổ biến, K-Nearest Neighbors (KNN) và Support Vector Machines (SVM), trong nhiệm vụ phân loại các loại lá cây. Để thực hiện điều này, chúng tôi đã sử dụng hai loại đặc trưng khác nhau, Histogram of Oriented Gradients (HOG) và Hu's Moments, nhằm xác định đặc điểm hình thái và cấu trúc hình học của các loại lá.

Qua quá trình phân tích và thử nghiệm, chúng tôi nhận thấy rằng: SVM đã cho thấy hiệu suất phân loại tổng thể cao hơn KNN khi sử dụng cùng một loại đặc trưng. Điều này được thể hiện rõ ràng qua các chỉ số Accuracy, Precision, Recall, và F1-Score, đặc biệt là khi áp dụng đặc trưng HOG. Đối với HU's Moments, cả hai thuật toán đều đạt được kết quả ấn tượng, với chỉ số F1 trung bình gần như hoàn hảo, cho thấy khả năng tuyệt vời của cả hai mô hình trong việc lấy các đặc điểm hình học để phân loại.

SVM có thời gian xử lý nhanh hơn đáng kể so với KNN trong cả hai loại đặc trưng. Điều này làm cho SVM trở thành lựa chọn ưu tiên cho các ứng dụng cần đến tốc độ xử lý nhanh và hiệu quả.

HU's Moments cho thấy khả năng phân loại vượt trội so với HOG trong môi trường thử nghiệm của chúng tôi, dù cho mỗi thuật toán. Điều này chỉ ra rằng các đặc trưng liên quan đến hình học của lá mang lại thông tin giá trị cao hơn cho việc phân loại.

Kết quả của đề tài này không chỉ góp phần vào việc lựa chọn công cụ phù hợp cho các ứng dụng phân loại hình ảnh trong lĩnh vực sinh học và môi trường mà còn mở ra hướng điều chỉnh và cải tiến các thuật toán học máy để tăng cường khả năng ứng dụng trong thực tiễn. Với sự phát triển của deeplearning và các mô hình CNN, việc sử dụng CNN để phân loại lá cây sẽ mang lại hiệu quả tích cực hơn, đồng thời mang lại nhiều hướng phát triển cao hơn nữa.

6 Phụ lục

1). Mã nguồn:

https://github.com/lephamcong/Classification_Of_Leaves.git

2). Tập dữ liệu thô (ảnh):

https://github.com/lephamcong/Classification_Of_Leaves/tree/main/Leaf_nhom_02

3). Tập dữ liệu HOG:

https://github.com/lephamcong/Classification_Of_Leaves/blob/main/HOG_nhom_02/HOG_nhom_02.csv

4). Tập dữ liệu HU's Moments:

https://github.com/lephamcong/Classification_Of_Leaves/blob/main/HU_nhom_02/HU_nhom_02.csv

Tài liệu tham khảo

- [1] Wu, Stephen Gang, et al. "A leaf recognition algorithm for plant classification using probabilistic neural network." *2007 IEEE international symposium on signal processing and information technology. IEEE*, 2007.
- [2] Hu, Ming-Kuei. "Visual pattern recognition by moment invariants." *IRE transactions on information theory* 8.2 (1962): 179-187.
- [3] Dalal, Navneet, and Bill Triggs. "Histograms of oriented gradients for human detection." *2005 IEEE computer society conference on computer vision and pattern recognition (CVPR'05)*. Vol. 1. Ieee, 2005.
- [4] Bharate, Anil A., and M. S. Shirdhonkar. "Classification of grape leaves using KNN and SVM classifiers." *2020 Fourth International Conference on Computing Methodologies and Communication (ICCMC)*. IEEE, 2020.
- [5] M. A. Hearst, S. T. Dumais, E. Osuna, J. Platt and B. Scholkopf, "Support vector machines," in *IEEE Intelligent Systems and their Applications*, vol. 13, no. 4, pp. 18-28, July-Aug. 1998, doi: 10.1109/5254.708428.
- [6] Lubis, Zulkarnain, Poltak Sihombing, and Herman Mawengkang. "Optimization of K Value at the K-NN algorithm in clustering using the expectation maximization algorithm." *IOP Conference Series: Materials Science and Engineering*. Vol. 725. No. 1. IOP Publishing, 2020.