# ỨNG DỤNG CÁC THUẬT TOÁN MÁY HỌC VÀO PHÂN LOẠI GIỐNG LÚA GẠO

## (RICE VARIETIES CLASSIFICATION)


1st Huỳnh Võ Ngọc Thanh
Khoa Khoa học máy tính
Trường Đại học Công nghệ thông tin-ĐHQG HCM, Việt Nam
21520449@gm.uit.edu.vn

2nd Nguyễn Trần Hoài Bão
Khoa Khoa học máy tính
Trường Đại học Công nghệ thông tin-ĐHQG HCM, Việt Nam
21520618@gm.uit.edu.vn

3rd Lê Văn Trường
Khoa Khoa học máy tính
Trường Đại học Công nghệ thông tin-ĐHQG HCM, Việt Nam
21522733@gm.uit.edu.vn



***Tóm tắt* - Lúa gạo là một nguồn lương thực cực kỳ quan trọng đối với con người không chỉ riêng ở các nước Châu Á mà còn phổ biến ở các nước khu vực Châu Phi, Châu Âu. Chúng có nhiều biến thể gen khác nhau với mỗi biến thể có những đặc điểm riêng biệt. Phân loại các giống lúa gạo phục vụ cho việc đánh giá và cải tiến chất lượng hạt gạo là nhu cầu cần thiết cho sức khỏe của con người. Bài báo cáo này sử dụng năm loại gạo thường được trồng tại Thổ Nhĩ Kỳ là Arborio, Basmati, Ipsala, Jasmine và Karacadag để thực hiện các hoạt động phân loại dựa trên đặc điểm hình ảnh của hạt gạo như: hình dạng, màu sắc, kích thước, độ nhẵn bóng,… Sử dụng bốn thuật toán máy học phổ biến: Support Vector Machines (SVM), k-Nearest Neighbors (kNN), Random Forest (RF) và Convolutional Neural Network (CNN) để huấn luyện mô hình phân loại. So với phương pháp Logistic Regression truyền thống chỉ phù hợp cho đa số các bài toán phân loại nhị phân, thì các phương pháp trên cho hiệu suất tốt hơn đối với phân loại đa lớp.**



***Keywords* -** *Máy học, phân loại, phân loại gạo, Support Vector Machines (SVM), k-Nearest Neighbors (kNN), Random Forest (RF), Convolutional Neural Network (CNN).*


## I. GIỚI THIỆU

Mô hình phân loại gạo tập trung vào việc dự đoán chính xác tên của loại gạo dựa trên các đặc trưng hình ảnh hạt gạo bằng các thuật toán học có giám sát. *Học có giám sát* là một mô hình máy học sử dụng dữ liệu đã được gán nhãn. Mục tiêu của các thuật toán học có giám sát là học một hàm ánh xạ từ vector đặc trưng (đầu vào) đến nhãn (đầu ra) dựa trên các dữ liệu có sẵn. Về cơ bản kích thước hạt gạo là rất nhỏ, việc nhận diện thủ công tốn nhiều thời gian và độ chính xác không đảm bảo. Hệ thống phân loại gạo tự động có thể khắc phục các hạn chế trên và giúp nhà nông, nhà nghiên cứu, các chuyên gia nông nghiệp kiểm soát chất lượng sản phẩm trong quá trình sản xuất và đảm bảo sản phẩm cuối cùng đáp ứng các tiêu chuẩn chất lượng lẫn số lượng.

Bài báo cáo này trình bày cách hoạt động cũng như phân tích và đánh giá bốn mô hình máy học: Support Vector Machines (SVM), k-Nearest Neighbors (kNN), Random Forest (RF) và Convolutional Neural Network (CNN) trên bộ dữ liệu mà nhóm thu thập từ Kaggle đồng thời rút trích những ưu nhược điểm của các mô hình. Để đánh giá mô hình nào tốt, nhóm đã tính các độ đo F1, Accuracy, Precision và Recall trên từng mô hình và điều chỉnh các tham số sao cho thuật toán có kết quả tốt nhất.

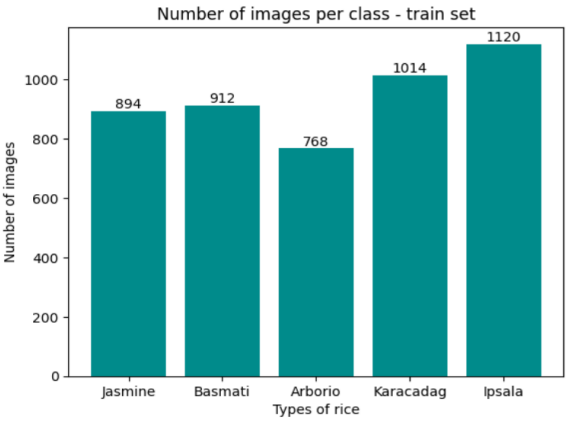
Đầu vào của bài toán là ảnh chụp của hạt gạo (thuộc một trong các loại gạo có trong bộ dữ liệu). Đầu ra là tên loại gạo có trong ảnh.

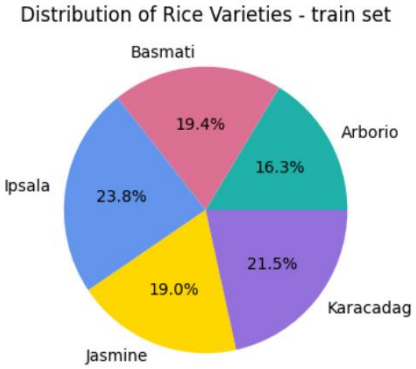
## II. BỘ DỮ LIỆU

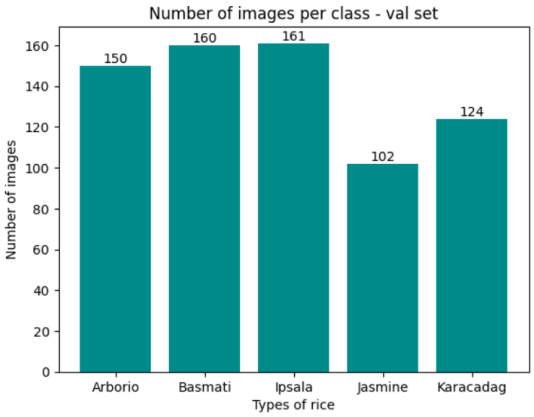
### A. *Mô tả bộ dữ liệu*

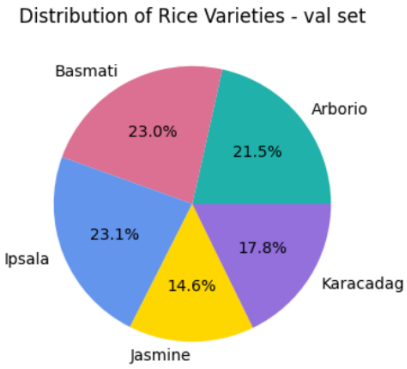
Bộ dữ liệu sử dụng là bộ dữ liệu có sẵn được lấy từ bộ dữ liệu gốc *Rice Image Dataset* trên trang web Kaggle gồm 75000 ảnh thuộc 5 lớp (mỗi lớp khoảng 15000 ảnh) tương ứng với 5 loại gạo riêng biệt: Arborio, Basmati, Ipsala, Jasmine, Karacadag.

Việc sử dụng toàn bộ dữ liệu gốc để huấn luyện các mô hình sẽ tiêu tốn rất nhiều thời gian cũng như tài nguyên máy tính không cho phép, nhóm đã tiến hành chọn lọc thủ công để có được bộ dữ liệu mới gồm tập train (4708 ảnh), tập validation (697 ảnh) và tập test (230 ảnh).



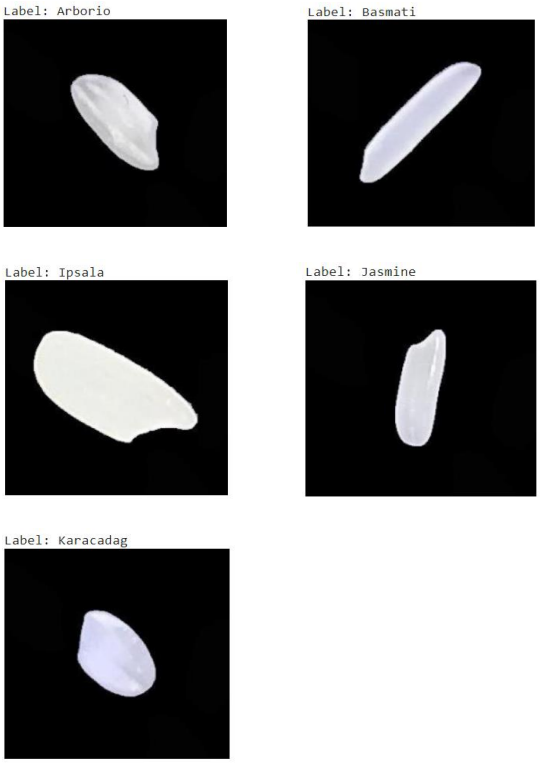






Nhìn chung cả tập train và tập validation đều có sự phân bố các mẫu tương đối đồng đều giữa các lớp. Điều này giúp tránh tình trạng mất cân bằng giữa số lượng mẫu thuộc các lớp khác nhau. Trong trường hợp mất cân bằng, mô hình có thể học chủ yếu từ lớp có số lượng mẫu lớn, gây ra hiện tượng thiên lệ trong dự đoán và đánh giá chất lượng mô hình.

Một số hình ảnh trong tập train:




- Arborio: hạt gạo thường có hình hạt ngắn, dày và tròn, có màu trắng trong. Kích thước của hạt có thể lớn hơn so với một số loại gạo khác, đặc biệt là khi nấu chín.
- Basmati: hạt dài và hơi dẹp, đầu hạt nhọn, có màu trắng trong.
- Ipsala: hạt thường hơi dẹp và to tròn, thường có màu trắng ngã vàng đục.
- Jasmine: hạt dài nhưng không bằng Basmati, thon và tròn, thường có màu trắng hoặc vàng nhạt.
- Karacadag: hạt ngắn và dẹp, có hình dạng tròn, thường có màu trắng sáng.

Download bộ dữ liệu sử dụng tại đây: https://drive.google.com/drive/folders/1UcsBR2bK5hmaq4H_dTpyZQ1mbC9QhTX6?usp=sharing

Download bộ dữ liệu gốc trên kaggle: https://www.kaggle.com/datasets/muratkokludataset/rice-image-dataset/data

*B. Cơ sở chọn lọc bộ dữ liệu mới từ bộ dữ liệu ban đầu*

Bộ dữ liệu sử dụng có kích thước nhỏ hơn bộ dữ liệu gốc nhưng vẫn phải đảm bảo các tiêu chí sau:

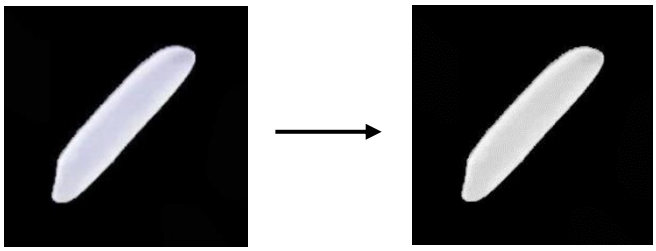
Bộ dữ liệu mới cần đại diện cho tất cả các đặc trưng quan trọng của dữ liệu gốc. Phản ánh đầy đủ và chính xác các đặc điểm hay các trường hợp đặc biệt của dữ liệu gốc. Ngoài ra, bộ dữ liệu mới phải đảm bảo sự cân bằng giữa số lượng mẫu thuộc từng lớp và bao gồm các mẫu từ các phân vùng quan trọng của phân phối dữ liệu gốc, tránh việc mô hình chỉ học từ một phần nhỏ của dữ liệu.

## III. TIỀN XỬ LÝ DỮ LIỆU

Tiền xử lý hình ảnh là quá trình chuẩn bị dữ liệu để hình ảnh sẵn sàng được đưa vào mô hình học máy. Mục tiêu của tiền xử lý dữ liệu là cải thiện chất lượng dữ liệu để làm cho mô hình có thể học được từ dữ liệu một cách hiệu quả nhất. Quá trình xử lý ảnh theo trình tự:

*A. Biến đổi ảnh màu sang ảnh xám*

Sử dụng hàm cvtColor trong thư viện OpenCV để chuyển ảnh màu sang ảnh xám nhằm giảm chiều dữ liệu, đồng thời tăng tốc quá trình huấn luyện.

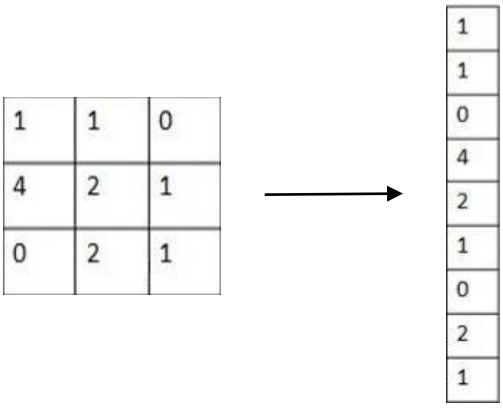


*B. Thay đổi kích thước ảnh*

Sử dụng hàm resize trong thư viện OpenCV để thay đổi kích thước cho tất cả ảnh giúp bộ dữ liệu có kích thước đồng nhất, tiết kiệm tài nguyên tính toán và tăng tính ổn định cho mô hình.

*C. Làm phẳng hình ảnh*

Sử dụng phương thức flatten biến đổi ảnh từ ma trận 2 chiều thành vector 1 chiều để phù hợp với cấu trúc đầu vào của các mô hình máy học.




*D. Chuẩn hóa dữ liệu hình ảnh*

Chuẩn hóa dữ liệu theo phân phối chuẩn (mean=0, std=1). Giảm ảnh hưởng của outliers giúp tăng tính ổn định mô hình.

**image = (image - image.mean()) / image.std()**

Sau khi xử lý dữ liệu đầu vào, ta tiến hành huấn luyện mô hình trên bộ dữ liệu vừa được xử lý.

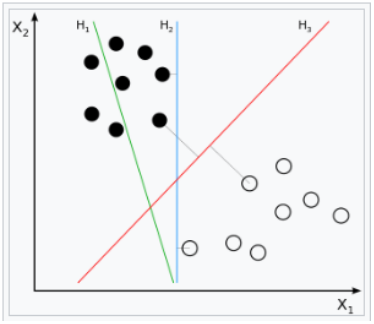
## IV. PHƯƠNG PHÁP MÁY HỌC

Bài báo cáo đề xuất 4 phương pháp máy học phổ biến phục vụ cho quá trình phân loại hình ảnh đa lớp hiệu quả: Support Vector Machines (SVM), k-Nearest Neighbors (kNN), Random Forest (RF) và Convolutional Neural Network (CNN).

*A. Support Vector Machines (SVM)*

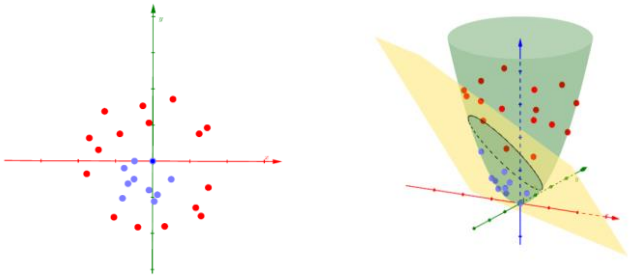
Thuật toán SVM ban đầu được tìm ra bởi Vladimir N. Vapnik và dạng chuẩn hiện nay sử dụng biên mềm được tìm ra bởi Vapnik và Corinna Cortes năm 1995. SVM tìm một siêu phẳng (hyperplane) tốt nhất phân chia giữa các điểm dữ liệu thuộc các lớp khác nhau sao cho khoảng cách từ các điểm (support vectors) đến siêu phẳng là lớn nhất [1].

SVM cố gắng tối đa hóa khoảng cách giữa siêu phẳng và các vector hỗ trợ (biên) nhằm đảm bảo tính tổng quát và khả năng phân loại tốt trên dữ liệu mới.




Đường thẳng màu đỏ phân chia hai điểm dữ liệu với biên lớn nhất.

Giới thiệu về Kernel SVM: Ý tưởng cơ bản của Kernel SVM là tìm một phép biến đổi sao cho dữ liệu ban đầu là không phân biệt tuyến tính được biến sang không gian mới. Ở không gian mới này, dữ liệu trở nên phân biệt tuyến tính [2].

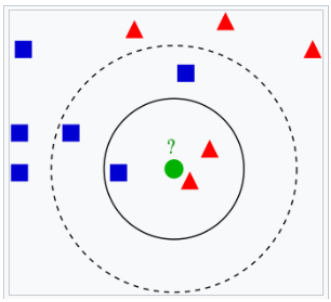


Kernel SVM là một công cụ mạnh mẽ để phân loại dữ liệu phi tuyến tính. Tuy nhiên, việc lựa chọn kernel phù hợp và xử lý dữ liệu có số chiều cao vẫn là thách thức đối với SVM [2].

Qua các lần thử nghiệm, sử dụng lớp SVC trong thư viện scikit-learn dễ cài đặt, đạt hiệu quả cao với hai tham số kernel = 'rbf' và C = 1.0. C là tham số điều chỉnh độ quan trọng của việc phạt các điểm nằm sai lệch từ ranh giới phân loại. Giá trị C lớn thì mô hình sẽ cố gắng phân loại mỗi điểm đúng trên tập huấn luyện, có thể dẫn đến việc mô hình bị overfitting. Kernel = 'rbf' là một lựa chọn phổ biến cho dữ liệu không tuyến tính.

*B. k-Nearest Neighbors (kNN)*

Thuật toán k-Nearest Neighbors được phát triển lần đầu bởi Evelyn Fix và Joseph Hodges vào năm 1951, và sau đó được mở rộng bởi Thomas Cover. kNN sử dụng đa số phiếu bầu từ k điểm gần nhất để quyết định lớp của điểm dữ liệu mới. Việc chọn k phụ thuộc vào bài toán cụ thể và đặc tính của dữ liệu [4].




Ta chọn k = 3 (vòng tròn đường liền) thì dữ liệu mới sẽ được gán cho lớp tam giác màu đỏ vì có 2 tam giác và chỉ có 1 hình vuông bên trong vòng tròn. Ta chọn k = 5 (vòng tròn nét đứt) thì nó sẽ được gán cho các hình vuông màu xanh.

Để xác định k điểm gần nhất, ta thực hiện tính khoảng cách giữa các điểm dữ liệu trong tập huấn luyện và điểm dữ liệu mới:

Khoảng cách Euclidean:

$$d(q, x_i) = \sqrt{\sum_{f \in F} (q - x_i)^2}$$

Khoảng cách Manhattan:

$$d(q, x_i) = \sum_{f \in F} |q - x_i|$$

Khoảng cách Minkowski:

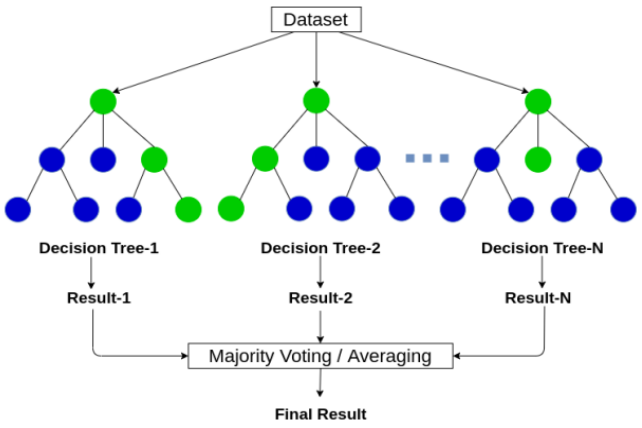
$$d(q, x_i) = (\sum_{f \in F} |q - x_i|)^{\frac{1}{p}}$$

Qua các lần thử nghiệm, sử dụng *KneighborsClassifier* trong thư viện scikit-learn dễ cài đặt, đạt hiệu quả cao với ba tham số n-neighbors=3, weights='distance', p=1.

N-neighbors quyết định số lượng láng giềng gần nhất mà mô hình sẽ xem xét khi đưa ra dự đoán cho một điểm dữ liệu mới. weights='distance' chỉ định các láng giềng gần hơn sẽ được coi là quan trọng hơn (có trọng số lớn hơn). Tham số p=1 thì sẽ tính bằng khoảng cách Manhattan.

### *C. Random Forest (RF)*

Random Forest là một phương pháp học kết hợp các cây quyết định được huấn luyện theo kỹ thuật bagging (hoặc pasting). [7]

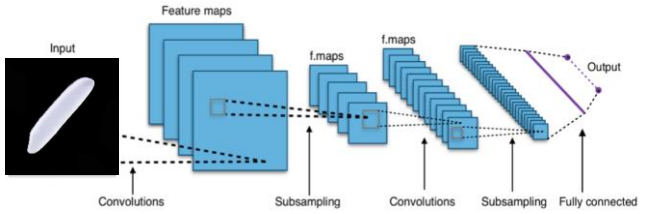



Thuật toán này tạo ra nhiều cây quyết định mà mỗi cây quyết định được huấn luyện dựa trên nhiều mẫu con khác nhau và kết quả dự báo là bầu cử (voting) từ toàn bộ những cây quyết định. Kết quả dự báo được tổng hợp từ nhiều mô hình nên kết quả của chúng sẽ không bị chệch. Đồng thời kết hợp kết quả dự báo từ nhiều mô hình sẽ có phương sai nhỏ hơn so với chỉ một mô hình. Điều này giúp cho mô hình khắc phục được hiện tượng quá khớp (overfitting). [8]

Qua các lần thử nghiệm, sử dụng lớp *RandomForestClassifier* trong thư viện scikit-learn dễ cài đặt, đạt hiệu quả cao với hai tham số n-estimators=100, max_depth=10. N-estimators là số lượng cây quyết định được tạo trong Random Forest. Số cây nhiều hơn thường giúp cải thiện hiệu suất của mô hình, nhưng cũng có thể làm tăng thời gian huấn luyện. max_depth là chiều sâu tối đa của mỗi cây quyết định, nó kiểm soát độ phức tạp của mỗi cây và có thể giúp kiểm soát overfitting (quá mức khớp).

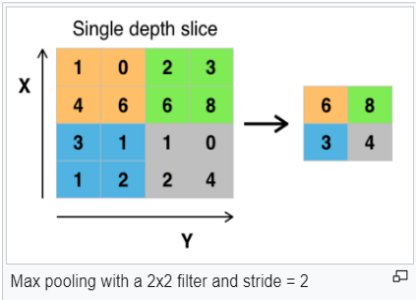
### *D. Convolutional Neural Network (CNN)*

CNN là kiến trúc mạng nơ-ron thường được sử dụng trong lĩnh vực thị giác máy tính và xử lý ảnh. Về cơ bản, một CNN sử dụng tầng tích chập để học các đặc trưng từ ảnh, tầng pooling để giảm kích thước và giữ lại thông tin quan trọng, và các tầng fully connected để thực hiện quyết định cuối cùng.




Lớp tích chập (convolutional layer) là khối xây dựng cốt lõi của một mạng nơ-ron tích chập. Các tham số của lớp này bao gồm một bộ lọc (hoặc nhân). Trong quá trình chuyển tiếp (forward pass), mỗi bộ lọc được tích chập qua chiều rộng và chiều cao của khối đầu vào, tính toán tổng trọng số giữa các thành phần của bộ lọc và đầu vào, tạo ra một bản đồ hoạt động 2 chiều cho bộ lọc đó. Do đó, mạng học được các bộ lọc kích hoạt khi nó phát hiện một loại đặc trưng cụ thể tại một vị trí không gian nào đó trong đầu vào. [9]

Lớp pooling dùng để giảm dần kích thước không gian của biểu diễn, từ đó giảm số lượng tham số, kích thước bộ nhớ và lượng tính toán trong mạng nơ-ron, và kiểm soát quá mức khớp. Thông thường, có thêm một lớp pooling giữa các lớp tích chập liên tiếp (mỗi lớp thường được theo sau bởi một hàm kích hoạt, như một lớp ReLU) trong kiến trúc của một CNN.[9]




Sau một số lớp tích chập và lớp max pooling, quá trình phân loại cuối cùng được thực hiện thông qua các lớp kết nối đầy đủ (fully connected layers). Các hoạt động của chúng có thể được tính toán như một biến đổi tuyến tính, với nhân ma trận tiếp theo bởi một điểm độ lệch. [9]

**Output = activation(input·weights + bias)**

Tuy nhiên trong bài báo cáo này, nhóm có thêm lớp dropout để ngăn chặn overfitting. Lớp đầu ra là lớp Softmax Activation để chuyển đổi các đầu ra tuyến tính thành xác suất. Hàm softmax tính xác suất mà điểm dữ liệu đó thuộc vào từng lớp.

## V. ĐÁNH GIÁ MÔ HÌNH

### *A. Các độ đo đánh giá:*

Để đánh giá hiệu suất của từng mô hình trên nhiều phương diện, ta tiến hành đánh giá trên 4 độ đo: Accuracy, Recall, Precision, F1-score:

1. Độ đo Accuracy:

Cho biết tỷ lệ các trường hợp được dự đoán đúng trên tổng số các trường hợp. Độ chính xác càng cao thì mô hình của chúng ta càng chuẩn xác.

$$\text{Accuracy} = \frac{\text{TP+TN}}{\text{totalsample}}$$

True Positive (TP): số lượng điểm của lớp positive được phân loại đúng là positive.
True Negative (TN): số lượng điểm của lớp negative được phân loại đúng là negative.
False Positive (FP): số lượng điểm của lớp negative bị phân loại nhầm thành positive.
False Negative (FN): số lượng điểm của lớp positive bị phân loại nhầm thành negative.

2. Độ đo Precision:

Precision đo lường tỷ lệ của các trường hợp dự báo positive mà thực sự là positive, hay nói cách khác, số lượng dự đoán đúng positive so với tổng số các trường hợp dự đoán positive.

$$\text{Precision} = \frac{\text{TP}}{\text{TP+FP}}$$

Với các bài toán có nhiều lớp dữ liệu như hiện tại, hai phép đánh giá dựa trên Precision và Recall nên được sử dụng là micro-average và macro-average. Macro-average precision là trung bình cộng của các precision theo lớp, tương tự với Macro-average recall.

3. Độ đo Recall:

Recall đo lường tỷ lệ của các trường hợp positive thực tế mà mô hình đã dự đoán đúng, tức là số lượng trường hợp positive dự đoán đúng so với tổng số các trường hợp positive thực tế.

$$\text{Recall} = \frac{\text{TP}}{\text{TP+FN}}$$

4. Độ đo F1-score:

F1 score tính toán một giá trị đo lường sự cân bằng giữa việc đạt được độ chính xác cao (precision) và việc đảm bảo mức độ phủ sóng cao (recall).

$$\text{F1} = \frac{2*Precision*Recall}{\text{Precision+Recall}}$$

*B. Kết quả đánh giá*

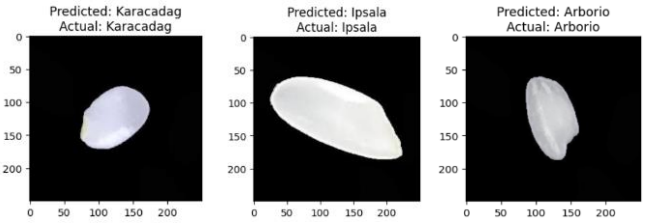
1. Đánh giá trên tập validation:

| | Accuracy | Precision | Recall | F1-score |
|---|---|---|---|---|
| SVM | 0.971 | 0.97 | 0.97 | 0.97 |
| KNN | 0.974 | 0.97 | 0.98 | 0.97 |
| RF | 0.969 | 0.97 | 0.97 | 0.97 |
| CNN | 0.978 | 0.98 | 0.98 | 0.98 |

2. Đánh giá trên tập test:

| | Accuracy | Precision | Recall | F1-score |
|---|---|---|---|---|
| SVM | 0.969 | 0.97 | 0.97 | 0.97 |
| KNN | 0.947 | 0.95 | 0.95 | 0.95 |
| RF | 0.965 | 0.97 | 0.97 | 0.97 |
| CNN | 0.965 | 0.97 | 0.97 | 0.97 |

*C. Một số kết quả thực nghiệm*




*D. Ưu điểm - nhược điểm của từng phương pháp*

1. Support Vector Machines:

**Ưu điểm:**

SVM cho phép phân loại các dữ liệu phi tuyến và có khả năng tạo ra các đường ranh giới phức tạp hơn để phân loại các lớp dữ liệu không tuyến tính [3].

SVM linh hoạt trong việc áp dụng các kernel khác nhau. Điều này cho phép SVM thích ứng với đa dạng các bài toán phân loại và mô hình dữ liệu [3].

Hiệu suất tốt trên tập dữ liệu lớn.

**Nhược điểm:**

Việc tính toán trong không gian cao hơn có thể tạo ra một số thách thức về hiệu suất và tốn nhiều thời gian huấn luyện.

Hiệu suất của Kernel SVM phụ thuộc rất nhiều vào lựa chọn kernel phù hợp. Việc chọn sai kernel có thể dẫn đến kết quả phân loại không tốt hoặc overfitting [3].

Đối với dữ liệu có số chiều cao: Khi số lượng thuộc tính của dữ liệu lớn hơn số lượng mẫu, SVM có thể không hoạt động hiệu quả và cho kết quả không tốt [3].

2. k-Nearest Neighbors

**Ưu điểm:**

kNN có thể được áp dụng cho dữ liệu mà không thể mô tả dưới dạng vector đặc trưng nếu có một độ đo tương đồng có sẵn. Do đó, kNN có thể được sử dụng trong những tình huống mà các thuật toán máy học khác không thể áp dụng [4].

kNN có một số kỹ thuật giảm nhiễu có thể cải thiện độ chính xác của bộ phân loại.

**Nhược điểm:**

Phân loại "biểu quyết theo đa số" cơ bản xảy ra khi phân phối lớp bị sai lệch, các đối tượng của một lớp phổ biến hơn có xu hướng thống trị dự đoán của đối tượng mới.

kNN rất nhạy cảm với các đặc trưng không liên quan hoặc trùng lặp vì tất cả các đặc trưng đóng góp vào độ tương đồng. Điều này có thể được cải thiện bằng việc lựa chọn đặc trưng hoặc đánh trọng số cho đặc trưng một cách cẩn thận.

3. Random Forest

**Ưu điểm:**

Random Forest được huấn luyện trên nhiều tập dữ liệu con khác nhau, bao gồm cả tập loại bỏ outliers. Điều này giúp mô hình ít bị nhạy cảm với dữ liệu outliers hơn.

Random Forest sử dụng nhiều cây độc lập, làm giảm khả năng overfitting của từng cây riêng lẻ. Mỗi cây trong Random Forest được huấn luyện trên các bộ dữ liệu con khác nhau, tạo ra sự đa dạng trong quyết định của mô hình và giúp tổng quát hóa trên dữ liệu nhiều chiều.

**Nhược điểm:**

Một mô hình Random Forest có thể bao gồm hàng trăm cây quyết định, điều này làm cho việc hiểu và diễn giải mô hình trở nên khó khăn.

Random Forest có thể mất thời gian đáng kể để huấn luyện, đặc biệt là khi có một số lượng cây lớn và tập dữ liệu lớn.

4. Convolutional Neural Netwwork

**Ưu điểm:**

CNN sử dụng các lớp tích chập để tự động học các đặc trưng cấp thấp và cấp cao từ dữ liệu, giúp chúng hiệu quả trong việc trích xuất thông tin hình ảnh.

CNN có khả năng tự động học cấp cao từ các đặc trưng cấp thấp, điều này giúp chúng mạnh mẽ trong việc hiểu cấu trúc và bối cảnh của dữ liệu.

CNN có thể dễ dàng tích hợp nhiều lớp để tăng sức mạnh biểu diễn của mô hình và nâng cao khả năng học đặc trưng phức tạp. CNN thường là lựa chọn hiệu quả cho việc xử lý dữ liệu ảnh, với khả năng giữ lại thông tin cấu trúc không gian quan trọng [9].

**Nhược điểm:**

Với dữ liệu ít, có khả năng mô hình sẽ học quá mức và không tổng quát hóa tốt trên dữ liệu mới (overfitting).

CNN có thể yêu cầu một lượng tính toán lớn, đặc biệt là đối với các mô hình sâu và dữ liệu lớn. Điều này có thể ảnh hưởng đến hiệu suất trong môi trường có tài nguyên hạn chế.

## VI. KẾT LUẬN – HƯỚNG PHÁT TRIỂN

### A. *Kết luận*

Dựa vào hình ảnh hạt gạo chứa các đặc trưng quan trọng và sử dụng các phương pháp máy học phù hợp, nhóm đã thành công xây dựng mô hình cơ bản trong việc hỗ trợ phân loại các loại gạo. Điều này chứng tỏ vai trò vượt trội của các phương pháp máy học trong lĩnh vực nông nghiệp.

Qua quá trình huấn luyện và đánh giá trên bộ dữ liệu, ta thấy mô hình CNN đạt hiệu quả tốt nhất trên tập validation với accuracy là 97.8% do CNN có thể tích hợp nhiều lớp dẫn đến khả năng học các đặc trưng mạnh mẽ. Các mô hình còn lại cũng cho hiệu quả tốt không kém với KNN (97.4%), SVM (97.1%) và RF (96.9%). Tuy nhiên, đối với tập test thì mô hình SVM lại cho hiệu quả cao nhất với accuracy là 96.9%, cùng xếp thứ hai là mô hình CNN và RF với accuracy là 96.5%, không bỏ quá xa so với SVM. Cho hiệu suất thấp nhất là mô hình KNN (94.7%) do tập test được chọn ngẫu nhiên nên có thể có sự trùng lặp về các đặc trưng.

### B. *Hướng phát triển*

Mặc dù các thực nghiệm hiện tại của nhóm chỉ tập trung chủ yếu trong phạm vi đồ án môn học, do đó còn nhiều hạn chế về dữ liệu và quy mô, nhóm sẽ tiếp tục nghiên cứu và cải tiến hệ thống này trong tương lai bằng cách tìm hiểu thêm về các mô hình mới và phương pháp cải tiến, cũng như tìm kiếm những nguồn dữ liệu uy tín khác và áp dụng các bước xử lý dữ liệu tiên tiến hơn. Với mục tiêu xây dựng một mô hình có thể nhận diện tốt từ 20 loại gạo trở lên, bao gồm đa dạng gạo ở tất cả các nước và các loại gạo có đặc điểm tương đối giống nhau, chúng em luôn hi vọng một ngày nào đó sẽ hoàn thiện và áp dụng thành công mô hình Rice Varieties Classification vào thực tiễn để góp phần phát triển nền nông nghiệp trong thời đại số.

## VII. NGUỒN TÀI LIỆU THAM KHẢO


[1] Support vector machine: https://en.wikipedia.org/wiki/Support_vector_machine

[2] Kernel: https://machinelearningcoban.com/2017/04/22/kernelsmv/

[3] https://viblo.asia/p/support-vector-machine-trong-hoc-may-mot-cai-nhin-don-gian-hon-XQZkxoQmewA

[4] KNN: https://en.wikipedia.org/wiki/K-nearest_neighbors_algorithm

[5] Distance: https://www.ibm.com/topics/knn

[6] RandomForest: https://en.wikipedia.org/wiki/Random_forest

[7] RandomForest: https://courses.uit.edu.vn/pluginfile.php/416121/mod_resource/content/1/Ch%C6%B0%C6%A1ng%207-%20H%E1%BB%8Dc%20k%E1%BA%BFt%20h%E1%BB%A3p.pdf

[8] https://phamdinhkhanh.github.io/deepai-book/ch_ml/RandomForest.html

[9] Convolutional Neural Network: https://en.wikipedia.org/wiki/Convolutional_neural_network

[10] https://www.ibm.com/topics/convolutional-neural-networks

[11] https://ieeexplore.ieee.org/document/10111346


Bảng phân công công việc

| STT | Họ và tên | MSSV | Công việc | Mức độ hoàn thành (đúng hạn) |
|---|---|---|---|---|
| **1** | Huỳnh Võ Ngọc Thanh | 21520449 | Tìm hiểu về bài toán<br>Viết báo cáo<br>Training models<br>Thuyết trình | 100% |
| **2** | Nguyễn Trần Hoài Bão | 21520618 | Tìm hiểu về bài toán<br>Training models<br>Thuyết trình | 100% |
| **3** | Lê Văn Trường | 21522733 | Tìm hiểu về bài toán<br>Viết báo cáo<br>Làm Slide<br>Thuyết trình | 100% |