Granular Computing (2024)9:29
https://doi.org/10.1007/s41066-024-00450-0

ORIGINAL PAPER

Classifying for interval and applying for image based on the extracted texture feature

Dan Nguyen-Thihong[1,2,3] · Tai Vo-Van[3]

Received: 19 October 2023 / Accepted: 4 January 2024
© The Author(s), under exclusive licence to Springer Nature Switzerland AG 2024

Abstract
This study develops a classification algorithm designed to handle interval data and to apply effectively in image processing. The proposed algorithm utilizes an innovative measure called overlap distance to assess the similarity between two intervals within multidimensional space. In addition, it integrates an improved method for determining prior probabilities by employing a fuzzy clustering technique. Furthermore, the study introduces a classification rule based on the quasi-Bayes method specifically tailored for interval data. According to this rule, an interval is assigned to a particular group if it holds the highest prior probability and the minimum distance to that group. The proposed algorithm is systematically presented in a step-by-step manner, elucidated by a numerical example, and executed using a well-established Matlab procedure. Another significant contribution of this study is its application to images, wherein texture features are extracted and represented as two-dimensional intervals. The effectiveness and superiority of the proposed algorithm are demonstrated through its application to various sets of medical images.

Keywords Interval data · Classification algorithm · Prior probability · Texture feature of image

1 Introduction

Classification is the process of assigning elements to groups with the goal of minimizing the probability of error. It is a crucial area of study in multidimensional statistics and data science, serving as the foundation of machine learning and artificial intelligence (Pham-Gia et al. 2000; Vovan 2016; Vovan et al. 2023). Its applications span diverse domains such as image and speech recognition, natural language processing, medical diagnosis, and fraud detection. For example, classification algorithms are commonly used in image recognition to identify objects or scenes such as cars, animals, or landscapes (Chen et al. 2016; Ha et al. 2022; Lethikim et al. 2022). Classification algorithms in medical diagnosis can predict the presence of a disease based on symptoms or test results (Behera et al. 2022b; Huynh-Van et al. 2023; Zhuang and Lin 2023). The classification problem has been applied to numerous practical problems across various fields, with many theoretical challenges that demand continuous attention from statisticians and information technology professionals.

The classification problem can be categorized into two primary types: numerical data and image data. Classifying for numerical data has a longer history of research compared to image data classification due to its abundant availability, captivating the interest of researchers (Pham-Gia et al. 2006; Vovan 2018; Vovan et al. 2019). However, with the rapid development of recording devices, image data are increasingly gaining popularity and becoming more abundant (Nguyentrang et al. 2023; Phamtoan and Vovan 2021). Image data hold significant value in artificial intelligence, underscoring the importance of image classification. To identify an image, the first step is to extract its features such as color, texture, or shape (Verma and Rohtagi 2023). Many researchers have explored the extraction of numeric data from images for identification purposes (VijayaLakshmi and Mohan 2016; Laleh and Shervan

✉ Tai Vo-Van
vvtai@ctu.edu.vn

[1] Vietnam National University Ho Chi Minh City, Ho Chi Minh City, Vietnam

[2] Faculty of Applied Science, Ho Chi Minh City University of Technology (HCMUT), Ho Chi Minh City, Vietnam

[3] College of Natural Science, Can Tho University, Can Tho City, Vietnam

Published online: 22 February 2024

2019; Nguyentrang et al. 2023). In recent years, some researchers have also represented image features as 2-dimensional intervals, which have shown advantages in cluster analysis for images (Ngoc et al. 2023; Phamtoan et al. 2022; Phamtoan and Vovan 2023; Le et al. 2023). This serves as a significant motivation for our continued study of the classification problem for interval data.

There exist numerous classification algorithms that fall into two main groups: traditional statistics and machine learning, including deep learning methods. Among the primary statistical classification methods are Naive Bayes, Logistic regression, Fisher's discriminant analysis, and Improved Bayes. Fisher's method has the capability to classify into multiple groups, but it requires the covariance matrix to be equal, a condition seldom met in actual data (Vovan 2016). Therefore, the Fisher method is often ineffective for practical applications. Logistic regression, often utilized for segregating groups, is typically applied to scenarios involving only two groups and is most effective when these groups have clearly separation.(Yuan et al. 2012). The Naïve Bayes method is simple in theory, but difficult to apply in practical implementation because there is almost no information about the facts and assumptions of event independence, resulting in limited outcomes when applied to classifiers (Chen et al. 2016). The improved Bayesian method (Bayes based on the density function) was proposed by Pham-Gia et al. (2000), and then interested and developed by many mathematicians (Vovan 2016; Ha et al. 2022; Vovan et al. 2022), but it has many difficulties in practical application because of problems such as estimating probability density function, determining prior probability, and solving computational complexity (Huynh-Van et al. 2023).

Machine learning offers various methods for classification. Quadratic Discriminant Analysis (QDA) is an extension of the Fisher method that allows each group to have its own covariance matrix, as opposed to a common one (Wu et al. 1996). XGBoost is a popular supervised learning algorithm that uses shallow decision trees to ensure accurate results and avoid overfitting, making it suitable for large datasets due to its scalability (Behera et al. 2022a). However, both QDA and XGBoost may not be ideal for datasets with significant overlap between groups. Support Vector Machine (SVM) stands as a potent classification tool in multidimensional space, but it may exhibit instability and high error rates when dealing with irregular data elements (Huang et al. 2018). Artificial Neural Networks (ANN) are capable of detecting complex non-linear relationships between variables, but they have drawbacks such as susceptibility to overfitting, high computational resource requirements, relying on researchers' experience for initial parameter settings (Neto et al. 2021). The Random Tree method is a supervised machine learning classifier that creates a group of decision trees and uses random subsets of variables for each tree, with the most frequent tree outputs used for general classification (Dietterich 2000). However, this method can be time-consuming to train as the number of trees increases. A Bagged classifier is an ensemble meta-estimator that fits classifiers to random subsets of the original dataset and aggregates their predictions for a final prediction. However, like the Random Tree method, it can also be time-intensive during training (Dietterich 2000; Nhu et al. 2020). Adaptive Boosting is a statistical classification meta-algorithm that can be used in conjunction with other learning algorithms to enhance performance (Wyner et al. 2017). K-nearest neighbor (kNN) is a complex method with high classification performance, relies heavily on parameter selection and distance types for each dataset (Imandoust and Bolandraftar 2013). The kNN method also has the drawback of high computation time, as it requires calculating distances to all elements in the training set for each element to be classified. Subspace-kNN combines predictions from multiple decision trees trained on different subsets, but it shares the same computational drawback as kNN (Gou et al. 2012). Convolutional neural networks (CNN) are very powerful classification methods, especially for image objects (Yamashita et al. 2018). While CNNs have proven highly effective in various computer vision tasks, they do have some disadvantages. They can be computationally intensive, especially for large and deep networks. In addition, they require a large amount of data and parameters, and training them demands a substantial amount of memory.

In the development of socio-economic aspects, alongside point data, we also store a substantial volume of interval data such as temperature, rainfall, and stock prices (Brito 2007). Interval data become crucial when precise calculations or measurements are unattainable, providing more precise information compared to ordinal or nominal data. The utilization of numerical intervals enhances our comprehension of the scale and relative distinctions among values. Interval data play a pivotal role not only in decision-making but also in predictive modeling. The ability to conduct mathematical operations on interval data empowers the creation of models that offer highly accurate predictions, aiding informed decision-making (Vovan 2023). Addressing practical needs, there arises a demand for data classification. Image recognition stands as a significant driver for classifying interval data. An image is not only identified by features extracted and represented by a vector, matrix, probability density function (Vovan 2016; Huynh-Van et al. 2023; Singh and Ganie 2022), but also by an interval (Vovan et al. 2023; Nguyentrang et al. 2023; Le et al. 2023). Among the aforementioned classification methods, the focus predominantly revolves around point data. The notion of recognizing an image through extracted

interval features presents a novel approach that often yields favorable outcomes. The classification of interval data, in our understanding, receives limited attention and research. Therefore, this study presents a classification algorithm that is specifically designed for interval data. The main contributions of this study can list as follows:

(i) Proposing an interval representation for a set of intervals and introducing a metric termed "overlap distance" to determine the similarity between intervals within a multidimensional space.
(ii) Developing an algorithm that calculates the prior probability for a classified interval based on a fuzzy clustering technique for this object. Each classified interval will have a unique prior probability for each group based on its association with that group.
(iii) Establishing a classification principle for interval data by utilizing the prior probability derived in (i) and the similarity levels between the classified interval and the groups. An interval is allocated to a specific group if it exhibits the highest prior probability and similarity to that group.
(iv) Implementing the proposed classification algorithm for image analysis based on its extracted textural feature using the gray-level co-occurrence matrix. The algorithm has been tested on numerous medical image datasets with different characteristics and has demonstrated competitive performance in comparison to other classification techniques, including machine learning and deep learning.

The remainder of the article is structured as follows. Section 2 discusses some problems related to the theory and application of the proposed algorithm, such as the distance between two intervals, the distance from an interval to a group of intervals, and estimating the probability density function. Section 3 presents the proposed algorithm and provides illustrative numerical examples. Section 4 applies the proposed algorithm to image data with specific datasets in medicine. The conclusion of the study is presented in the last section.

2 Distance between intervals and problem of estimating the probability density function

2.1 Distance between two intervals

Definition 1 Given two n-dimensional intervals a and b, $n \geq 1$:

$$a = \left(a^1, a^2, \ldots, a^n\right) = \left([a_1, \widehat{a_1}], [a_2, \widehat{a_2}], \ldots, [a_n, \widehat{a_n}]\right),$$
$$b = \left(b^1, b^2, \ldots, b^n\right) = \left(\left[b_1, \widehat{b_1}\right], \left[b_2, \widehat{b_2}\right], \ldots, \left[b_n, \widehat{b_n}\right]\right).$$

Then, we have the following common distances between a and b:

Hausdorff distance: $d_H(a,b) = \sum_{i=1}^{n}(\max\{|a_i - b_i|, |\hat{a_i} - \hat{b_i}|\})$.

City-block distance: $d_C(a,b) = \sum_{i=1}^{n}(|a_i - b_i| + |\hat{a_i} - \hat{b_i}|)$.

Euclidean distance: $d_E(a,b) = \sqrt{\sum_{i=1}^{n}[(a_i - b_i)^2 + (\hat{a_i} - \hat{b_i})^2]}$.

Overlap distance:

$$d_O(a,b) = \sum_{i=1}^{n} \max\left\{d_{OL}\left(a^i, b^i\right), d_{OL}\left(b^i, a^i\right)\right\}, \quad (1)$$

where $d_{OL}(a,b)$ is the distance between two one-dimensional intervals, defined as follows:

$$d_{OL}(a,b) = D(a,b)\left(1 - \frac{OA(a,b)}{2r_a + 1}\right), \quad (2)$$

where $OA(a; b)$ is the measure of the overlap area between a and b, $D(a; b)$ is calculated by (3).

$$D(a,b) = \max_{a' \in [a,\hat{a}]}\{\min_{b' \in [b,\hat{b}]}\{d_E(a', b')\}\}. \quad (3)$$

Remark 1 Given two one-dimension interval $a = [a, \hat{a}], b = [b, \hat{b}]$. Set $c_a = \frac{a+\hat{a}}{2}$, $r_a = \frac{\hat{a}-a}{2}$, $c_b = \frac{b+\hat{b}}{2}$ và $r_b = \frac{\hat{b}-b}{2}$. Then, the overlap distance between a and b can be divided into five cases as follows:

(i) The interval a completely overlaps the interval $b : |c_a - c_b| \leq r_b - r_a$.
(ii) The interval b completely overlaps the interval $a : |c_a - c_b| \leq r_a - r_b$.
(iii) The interval b overlaps with the interval a on the left boundary of $a : r_a = r_b = 0$.
(iv) The interval b overlaps with the interval a on the right boundary of $a : |r_a - r_b| < |c_a - c_b| < r_a + r_b$.
(v) The interval b either lies to the left of interval a, or lies to the right of the interval a and there is no overlap between them: $|c_a - c_b| \geq r_a + r_b$.

From these five cases, the overlap distance between a and b is specifically defined as follows:

$$d_{OL}(a,b) = \begin{cases} 0 & \text{for } (i), \\ (|c_a - c_b| + r_a - r_b)\left(1 - \dfrac{2r_b}{2r_a + 1}\right) & \text{for } (ii), \\ |c_a - c_b| & \text{for } (iii), \\ (|c_a - c_b| + r_a - r_b)\left(1 - \dfrac{r_a + r_b - |c_a - c_b|}{2r_a + 1}\right) & \text{for } (iv), \\ (|c_a - c_b| + r_a - r_b)\left(1 + \dfrac{|c_a - c_b| - (r_a + r_b)}{2r_a + 1}\right) & \text{for } (v). \end{cases}$$

Remark 2 The differences between two intervals, as measured by d_E, d_H, and d_C, are calculated solely based on the endpoints of the intervals. In contrast, the similarity between two intervals is evaluated using the overlapping distance d_O, which takes into account not only the endpoints of the intervals but also their overlap. Therefore, when comparing two intervals, using d_O is a more appropriate measure of similarity.

2.2 The distance between an interval and a group of intervals

Given N intervals $\{a_1, a_2, ..., a_N\}$ of c groups $\{w_1, w_2, ..., w_c\}$.

Definition 2 The matrix $U = [\mu_{ij}]_{c\times N}$ with $\mu_{ij} \in [0,1], \sum_{=1}^{c} \mu_{ij} = 1$ and $0 < \sum_{j=1}^{N} \mu_{ij} < N$ is called the partition matrix of N given intervals.

Definition 3 The interval defined by

$$v_i = \frac{\sum_{k=1}^{N} (\mu_{ik})^2 v_k}{\sum_{k=1}^{N} (\mu_{ik})^2}, 1 \le i \le c \tag{4}$$

is called the representative interval of group w_i.

Let V be a set of intervals, and let a_0 be an interval. Similarly, for discrete data, we can calculate the distance between a_0 and V using the minimum, maximum, and mean distances (Vovan and Nguyentrang 2017). However, for intervals, computing the distances between all pairs of elements can be time-consuming, making it unsuitable for classification problems. In this study, we propose to evaluate the similarity of a_0 and V by comparing a_0 with v, which is the interval that represents the set V, in order to solve this problem.

2.3 Estimating the probability density function

Typically, discrete elements are used to store data that needs to be estimated before applying the Bayesian method. There are several methods available for estimating probability density functions (pdfs), including parametric and non-parametric methods. Among non-parametric methods, the kernel function method is widely used nowadays and is considered to have several advantages (Vovan and Pham-Gia 2010; Vovan 2016; Nguyentrang et al. 2023). In this study, to compare the proposed algorithm with the Bayesian method, we utilize both the kernel function and Copula methods.

i) *The kernel function method:*

The pdf estimated by the kernel function method has the form:

$$\widehat{f}\ (x) = \frac{1}{Nh_1h_2...h_n} \sum_{i=1}^{N} \prod_{j=1}^{n} f_j\left(\frac{x_j - x_{ij}}{h_j}\right), \tag{5}$$

where $x_j, j = 1, 2, ..., n$ are variables, $x_{ij}, i = 1, 2, ..., N$ is ith data of jth variable; h_j is bandwidth for the jth variable, $f_j(.)$ is kernel function of jth variable, which is usually normal, Epanechnikov, biweight and triweight.

According to this method, the selection of bandwidth and kernel function plays a crucial role in the obtained results. Although many authors, such as Vovan (2016), Nguyentrang et al. (2023), and Nguyentrang et al. (2023), have discussed this issue extensively, the optimal choice has not been definitively determined yet. In this study, the bandwidth parameter is chosen based on the approach proposed by Terrell and Scott (1992) and Vovan (2016), and the Gaussian kernel function is utilized.

ii) *The Copula method:*

Let $X = (X_1, \ldots, X_d)$ be a vector of d random variables, with a joint probability density function f. According to Sklar's theorem (Sklar 1959), every multivariate distribution F with marginals $F_1, \ldots, F_d$ can be expressed as follows:

$$F(x_1, \ldots, x_d) = C(F_1(x_1), \ldots, F_d(x_d)), \tag{6}$$

According to Sklar's theorem, for some appropriate d-dimensional copula C, and using the chain rule, we have for an absolutely continuous F with continuously differentiable marginals $F_1, \ldots, F_d$ that

$$f(x_1, ..., x_d) = \left[\prod_{k=1}^{d} f_k(x_k)\right] \times c(F_1(x_1), \ldots, F_d(x_d)), \tag{7}$$

where $c(.)$ denotes the copula density. Currently, there are many common Copula families used, such as Gauss, Student, etc. However, according to Mejdoub and Arab (2018), there is no Copula family considered to be optimal in applications so far. In this study, we use the Gaussian Copula family in our numerical applications.

3 The proposed algorithm

3.1 Algorithm

Given c groups $w_1, w_2, \ldots, w_c$, where group w_i has n_i intervals I_{n_i} for $i = 1, 2, \ldots, c$, and $n_1 + n_2 + \ldots + n_c = N$, where N is the total number of intervals, and an interval I_0 that needs to be classified into one of the above groups. The classification algorithm for I_0 consists of the following steps:

Step 1. Establish the initial partition matrix $U^{(0)}$ as follows:

$$U^{(0)} = \left[u_{ij}^{(0)}\right]_{c\times(N+1)} = \begin{bmatrix} u_{11}^{(0)} & u_{12}^{(0)} & \ldots & u_{1N}^{(0)} & u_{1(N+1)}^{(0)} \\ u_{21}^{(0)} & u_{22}^{(0)} & \ldots & u_{2N}^{(0)} & u_{2(N+1)}^{(0)} \\ \ldots & \ldots & \ldots & \ldots & \ldots \\ u_{c1}^{(0)} & u_{c2}^{(0)} & \ldots & u_{cN}^{(0)} & u_{c(N+1)}^{(0)} \end{bmatrix}. \quad (8)$$

The matrix described above consists of c rows and $N+1$ columns, where initial N columns represent the probability indicating the likelihood of the j^{th} interval being allocated to group w_i, $u_{ij} \in [0,1], i = 1, 2, \ldots, c; j = 1, 2, \ldots, N+1$. In the first step, the assignment of value for u_{ij} follows these criteria:

* For the initial N columns, $u_{ij} = 1$ if the j^{th} interval belongs to group w_i, and $u_{ij} = 0$ if it is not.

* For the last column, $u_{i(N+1)}$ pertains to the probability of assigning I_0 to specific groups as initially specified. This value, $v_{i(N+1)} = 1/c, i = 1, 2, \ldots, c$, implies an equal probability distribution among the c groups for assigning I_0.

Step 2. Determine the representative interval $v_i^{(1)}$ for groups and compute the d_O of all intervals from I_k to $v_i^{(1)}, 1 \le i \le c, 1 \le k \le N+1$.

Step 3. Update the partition matrix. At iteration $t = 1, 2, \ldots$, each element of $U^{(t)}$ is determine as follows:

* If $d_O(I_k, v_i^{(t)}) > 0, \forall i = 1, 2, \ldots, c$, $u_{ik}^{(t)}$ is updated as follows:

$$u_{ik}^{(t)} = \frac{1}{\sum_{j=1}^{c} [d_O(I_k, v_i^{(t)})/d_O(I_k, v_j^{(t)})]^2}, \quad 1 \le i \le c, 1 \le k \le N+1. \quad (9)$$

* If $\exists i' : d_O(I_k, v_{i'}^{(t)}) = 0$ then $u_{ik}^{(1)} = 0, \forall i \ne i'$ and $u_{ik}^{(t)}$ is chosen randomly at $i = i'$ such that $\sum_{i=1}^{c} u_{ik}^{(t)} = 1$.

Step 4. Repeat Step 2 and Step 3 until

$$J^{(t)} = |J\left(V^{(t)}, U^{(t)}\right) - J\left(V^{(t-1)}, U^{(t-1)}\right)| < \epsilon, \quad (10)$$

where

$U^{(t)} = [u_{ij}^{(t)}]_{c\times(N+1)}$ is the partition matrix at the t^{th} iteration,

$V^{(t)} = \{v_1^{(t)}, v_2^{(t)}, \ldots, v_c^{(t)}\}$ is set of the representative interval of groups at the t^{th} iteration,

$\epsilon > 0$ is a very small positive number. In this study, ϵ is chosen as 0.0001,

The value of $J\left(V^{(t)}, U^{(t)}\right)$ is computed as follows:

$$J\left(V^{(t)}, U^{(t)}\right) = \sum_{i=1}^{c}\sum_{k=1}^{N} u_{ik}^2 [d_O(I_k, v_i^{(t)}]^2. \quad (11)$$

When Step 4 ends, we obtain the matrix $U^{(t)}$, where the last column is given as follows:

$$u = \begin{bmatrix} u_1 \\ u_2 \\ \ldots \\ u_c \end{bmatrix} = \begin{bmatrix} u_{1(N+1)}^{(t)} \\ u_{2(N+1)}^{(t)} \\ \ldots\ldots\ldots \\ u_{c(N+1)}^{(t)} \end{bmatrix}.$$

Since u is the initial probability to assign I_0 to the groups, it is taken as the prior probability.

Step 5. Find the representative intervals for groups with prior probability obtained from Step 4 and compute $d_O(I_0, v_i^{(t)})$, the distance of I_0 and these representative intervals of groups with $u_{ik}^{(t)}$ obtain from Step 4. Normalize these distances to [0;1] to have $r_i, i = 1, 2, \ldots, c$.

Step 6. Compute the value called the posterior similarity (PS) of I_0 to group $w_i, i = 1, 2, \ldots, c$:

$$PS(i) = u_i.(1 - r_i),$$

and classify I_0 to group w_i by the following rule:

$$w_i = \arg\max_i \{PS(i)\}. \quad (12)$$

The developed algorithm has 2 phases. Phase 1 includes Step 1 to Step 4 and Phase 2 includes Step 5 and Step 6.

* Phase 1 involves establishing the prior probability for the classified interval. Traditional methods include choosing the same values for all groups, using the ratio based allocation derived from group sizes, or application the Laplace method (Pham-Gia et al. 2000; Miller et al. 2001; Pham-Gia et al. 2006; Vovan 2016). However, these methods do not consider the classified element itself. Recent research has explored cluster analysis techniques for determining the prior probability in both numerical and interval data (Nguyentrang and Vovan 2017). In our proposed algorithm, we leverage the fuzzy cluster analysis technique tailored to intervals to establish the prior probability. This estimation is rooted in the fuzzy relationship between the classified interval and the relevant groups. We contend that this approach stands as a more rational alternative to traditional methods.

* Phase 2 involves formulating the classification rule using the quasi-Bayesian method. The $PS(i)$ (posterior similarity) of I_0 to group w_i is calculated by multiplying its prior probability with the normalized distance to that specific group. When the $PS(i)$ attains its the maximum value, signifying the closet proximity of I_0 to group w_i, and the classification assigns I_0 to this group. This refined classification rule has notably enhanced the accuracy of our classification outcomes.

* In Step 1, the values in the matrix $U^{(0)}$ can be randomly chosen. The selection of these values may affect the runtime but not the result because the proposed algorithm is convergence. Based on numerical examples, the parameters proposed in the manuscript seem reasonable in terms of computation and user-friendliness. In the proposed algorithm, Step 4 also has a parameter ϵ that needs to be chosen during execution. This parameter measures the difference in the value $J^{(t)}$ between two consecutive iterations. As ϵ increases, the number of algorithm iterations decreases, and vice versa. From experience, we have chosen $\epsilon = 0.0001$ to obtain reasonable results for the prior probability in numerical examples and applications.

Pseudocode of the proposed algorithm is presented in Algorithm 1.

Algorithm 1 Pseudocode of the proposed algorithm

```
Begin
In put the train set of groups {w_1, w_2, ..., w_c} with N intervals, the classified interval I_0, and ε = 0.001.
Establish the initial partition matrix U^(0) = [u_ik^(0)]_{c×(N+1)} using (8).
Determine the representative interval for each group using (4) to have the series intervals V^(0).
for do t = 1, 2, 3, ...
    Update the initial partition matrix U^(t) = [u_ik^(t)]_{c×(N+1)} using ( 9).
    Compute J(V^(t), U^(t)) using (11);
    if then |J(V^(t), U^(t)) − J(V^(t−1), U^(t−1))| < ε
        Break
    else
        Establish partition matrix U^(t) = [u_ik^(t)]_{c×(N+1)}.
    end if
end for
Establish the prior probability u_i to assign I_0 into group w_i, i = 1, 2, ..., c.
Find the representative intervals for groups, V^(t);
Compute the overlap distance of I_0 and the representative interval of groups, and normalize them in [0, 1] to have r_i, i = 1, 2, ..., c.
Compute PS(i) = u_i.(1 − u_i).
Classify I_0 to w_i by rule: w_i = arg max_i {PS(i)}, i = 1, 2, ..., c.
End
```

3.2 Illustrative example

Given 21 two-dimension intervals $\{a_1, a_2, ..., a_{21}\}$ which belong to 3 groups $w_1 = \{a_1, a_2, ..., a_7\}; w_2 = \{a_8, a_9, ..., a_{14}\}; w_3 = \{a_{15}, a_{16}, ..., a_{21}\}$ and an classified interval (a_{22}). The detail of the intervals are shown below:

$$
\begin{aligned}
&a_1 = ([4.3, 5.7], [2.7, 3.7]), a_2 = ([2.9, 4.5], [2.0, 5.5]),\\
&a_3 = ([2.1, 7.9], [3.0, 3.9]), a_4 = ([2.6, 2.6], [4.2, 5.9]),\\
&a_5 = ([3.1, 4.8], [4.0, 4.9]); a_6 = ([3.8, 4.0], [2.2, 7.7]),\\
&a_7 = ([4.7, 7.5], [2.1, 2.3]), a_8 = ([2.2, 6.4], [3.6, 4.0]),\\
&a_9 = ([6.1, 8.9], [5.5, 7.4]), a_{10} = ([6.2, 9.9], [6.4, 9.3]),\\
&a_{11} = ([5.3, 10.8], [6.7, 8.1]), a_{12} = ([5.6, 6.9], [5.6, 6.1]),\\
&a_{13} = ([6.7, 8.6], [5.7, 9.6]), a_{14} = ([5.8, 7.5], [5.5, 6.8]),\\
&a_{15} = ([11.2, 12.4], [11.3, 14.7]),\\
&a_{16} = ([12.3, 12.9], [10.9, 15.8]),\\
&a_{17} = ([12.3, 15.8], [11.4, 11.7]),\\
&a_{18} = ([10.1, 14.2], [10.5, 14.0]),\\
&a_{19} = ([12.1, 13.7], [11.4, 16.0]),\\
&a_{20} = ([10.4, 11.8], [11.0, 12.2]),\\
&a_{21} = ([10.8, 11.8], [10.5, 16.5]),\\
&a_{22} = ([9.5, 11], [9, 12]).
\end{aligned}
$$

where $\{a_1, a_2, ..., a_8\}$ belong to group w_1, $\{a_9, a_{10}, ..., a_{14}\}$ belong to group w_2, and $\{a_{15}, a_{16}, ..., a_{21}\}$ belong to group w_3.

These intervals are illustrated by Fig. 1.

Upon examining Fig. 1, it is evident that a_{22} is closest to group w_3. Therefore, a suitable classification would be to assign a_{22} to w_3. The steps of the proposed algorithm for this dataset are outlined as follows:

Step 1. Establish the initial partition matrix $U^{(0)}$, using (8):

$$
\begin{aligned}
U^{(0)} &= [u_{ij}]_{3\times 22}\\
&= \begin{pmatrix}
1 & 1 & 1 & \dots & 0 & 0 & 0 & \dots & 0 & 0 & 0 & 1/3\\
0 & 0 & 0 & \dots & 1 & 1 & 1 & \dots & 0 & 0 & 0 & 1/3\\
0 & 0 & 0 & \dots & 0 & 0 & 0 & \dots & 1 & 1 & 1 & 1/3
\end{pmatrix}.
\end{aligned}
$$

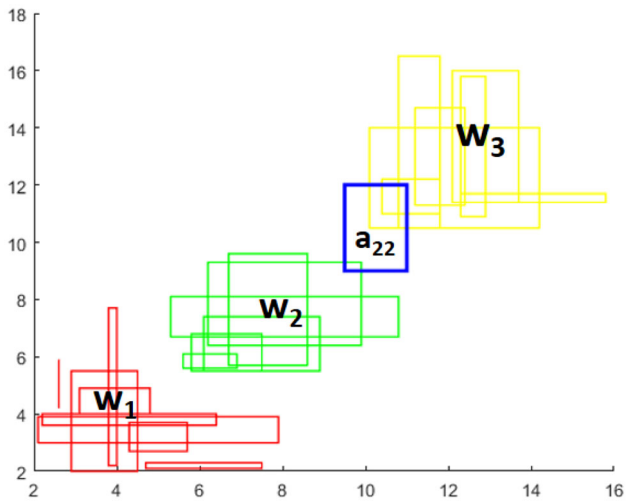
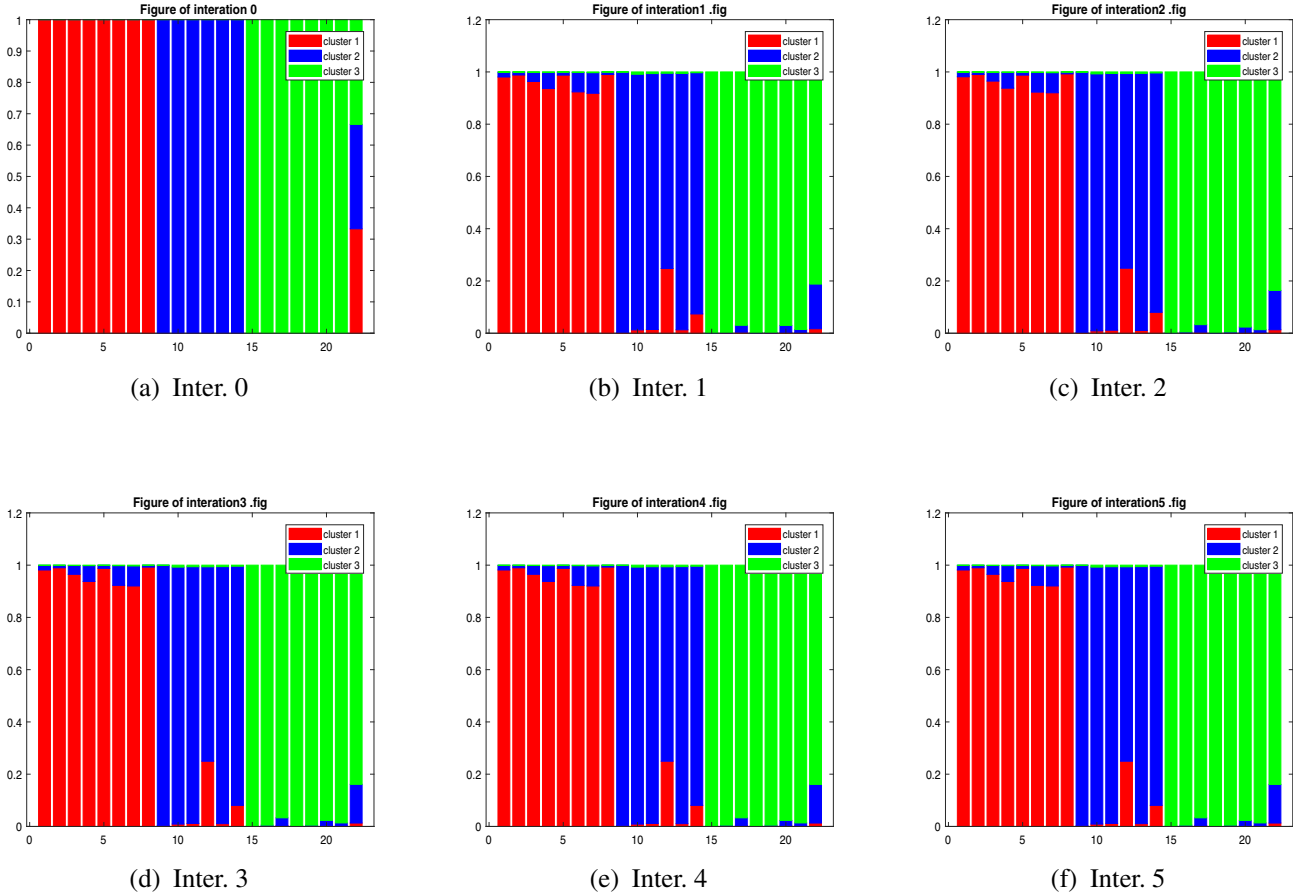


Fig. 1 The demonstration for 22 intervals of three groups



(a) Inter. 0 (b) Inter. 1 (c) Inter. 2

(d) Inter. 3 (e) Inter. 4 (f) Inter. 5

Fig. 2 The prior probability for iterations

The above matrix has 3 rows and 22 columns, where

* For the 8 first columns, $u_{1j} = 1, u_{2j} = u_{3j} = 0, j = \overline{1,8}$.

* For the next 6 columns, $u_{2j} = 1, u_{1j} = u_{3j} = 0, j = \overline{9,14}$.

* For the next 7 columns, $u_{3j} = 1, u_{1j} = u_{2j} = 0, j = \overline{15,21}$.

* For the last column, $u_{i(22)} = 1/3, i = 1,2,3$ represents the probability of belonging to each group for a_{22}.

Step 2. Compute representative interval $V^{(0)}$ for the groups based on the initial partition matrix $U^{(0)}$, using (4):

$$V^{(0)} = \begin{bmatrix} ([3.2985, 5.5012], [3.0574, 4.8368]) \\ ([6.0144, 8.8072], [5.9563, 7.9580]) \\ ([11.2860, 13.1938], [10.9688, 14.3766]) \end{bmatrix}.$$

After that, compute the overlap distance of all intervals $a_j, j = 1,2,...,22$ to the representative intervals $V_i^{(0)}$, we obtain the matrix:

$$\begin{bmatrix} 1.4999 & 1.1090 & ... & 29.1381 \\ 10.8233 & 10.3844 & ... & 9.2171 \\ 61.5669 & 50.1633 & ... & 4.2302 \end{bmatrix}_{3\times 22}.$$

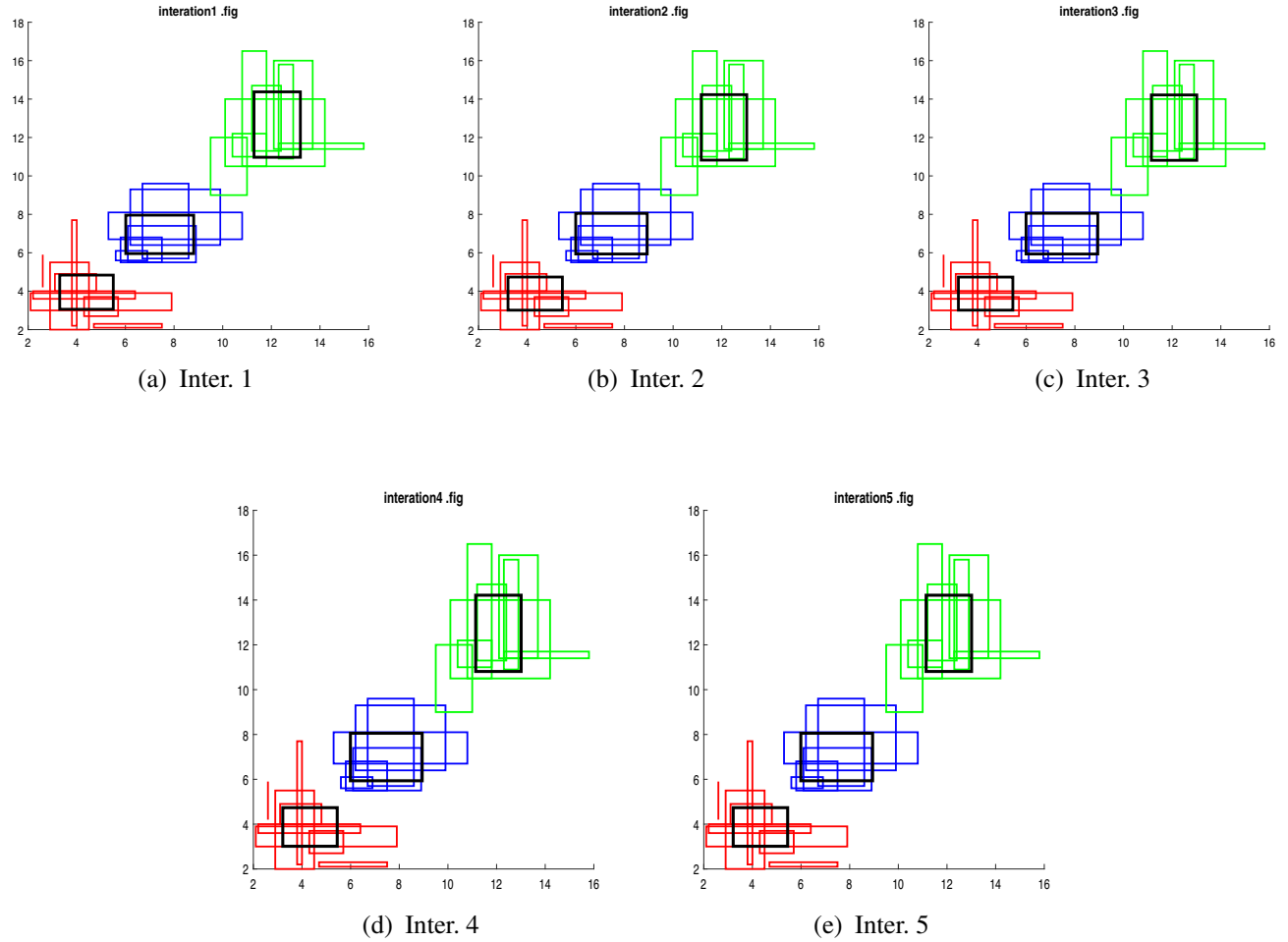
Step 3. Update the partition matrix using (9), we have the new matrix as follows:

$$U^{(1)} = U^1_{3x22} = \begin{bmatrix} 0.9806 & 0.9882 & ... & 0.0171 \\ 0.0188 & 0.0113 & ... & 0.1710 \\ 0.0006 & 0.0005 & ... & 0.8119 \end{bmatrix}_{3\times 22}.$$

Based on the results of $V^{(0)}, U^{(0)}, V^{(1)}, U^{(1)}$ and using (10), we compute $J^{(1)} = 5.2908$.

Step 4. Repeat Step 2 and Step 3, after 5 iterations, the stop condition is satisfied. The detailed results of the matrices $U^{(0)}$, $U^{(1)}$, $U^{(2)}$, $U^{(3)}$, $U^{(4)}$, $U^{(5)}$ are demonstrate by Fig. 2 and the representative groups over the iterations given by Fig. 3.

In Fig. 2 (a), the initial 8 columns exhibit a value of 1, signifying that the 8 intervals $a_1, ..., a_8$ possess a prior probability of being assigned to w_1 all equal to 1. The subsequent 7 columns display a value of 1, representing the



(a) Inter. 1 (b) Inter. 2 (c) Inter. 3

(d) Inter. 4 (e) Inter. 5

Fig. 3 The representative intervals of groups for iterations

7 intervals $a_8, ..., a_{14}$ with a prior probability of being assigned to w_2 also equal to 1. Following the initial 6 columns with a value of 1, correspond to the 6 intervals $a_{15}, ..., a_{21}$ holding a prior probability of being assigned to w_3 all equal to 1. The final column displays three equally segmented colors denoting the equal prior probability for classifying a_{22} into 3 groups. After 5 iterations, Fig. 2 (f) across columns, notably emphasizing the probability of assignment to w_3, in the last column, with a minimal probability assigned to w_1, barely discernible in this column. Moving to Fig. 3 ortrays the intervals representing the groups throughout 5 iterations. While the intervals pertaining to the three groups remain unaltered, their representative intervals (depicted as black rectangles) undergo changes across these iterations.

Besides that, the matrix $U^{(5)}$ also provides the prior probability of the element a_{22}, that is

$$u = \begin{bmatrix} 0.0130 \\ 0.1478 \\ 0.8392 \end{bmatrix}.$$

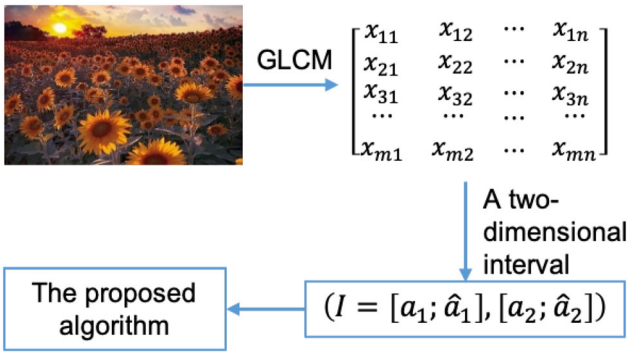


Fig. 4 The extraction and classifying an image

Step 5.

Step 5.1 Find the representative intervals for groups with the prior probability obtained from Step 4, we have

$$V^{(5)} = \begin{bmatrix} ([3.2108; 5.4495], [3.0120; 4.7348]) \\ ([5.9909; 8.9417], [5.9346; 8.0549]) \\ ([11.1419; 13.0115], [10.8112; 14.2142]) \end{bmatrix}.$$

Step 5.2 Compute $d_O(a_{22}, V^{(5)})$, and normalize them in [0,1], we have

$$r = \begin{bmatrix} 1.0000 \\ 0.2969 \\ 0.1246 \end{bmatrix}.$$

Step 6. Compute the probability of the a_{22} belongs to each group, we have

$$PS(1) = u_1.(1 - r_1) = 0; PS(2) = u_2.(1 - r_2) = 0.1039;$$
$$PS(3) = u_3.(1 - r_3) = 0.7346.$$

Since $w_3 = \arg\max_i \{PS(i)\}, a_{22}$ is classified in group w_3. This is a suitable classification.

4 Application for medical figure data

In the realm of medical science, classification holds utmost significance as it mirrors the actual process of diagnosing a disease. When an individual seeks medical evaluation for a specific ailment, determining the presence or absence of that disease becomes imperative. If the diagnosis confirms the disease, categorizing it correctly becomes crucial for formulating an effective treatment plan. All of these issues essentially translate into classification problems. Historically, doctors relied on qualitative and quantitative indicators for diagnosis and disease classification. However, with the strong development of recording devices, images have become indispensable input data. To decipher these images, the initial step involves feature extraction. Typically, the features of an image are extracted from color, texture, and shape. In this study, our focus is on extracting texture features from the image to facilitate recognition. These extracted texture features are then translated into a range. Subsequently, the image's identification hinges on this extracted range, employing the aforementioned algorithm proposed in this study.

Table 1 Notation for the process of performance

Train set	Test set	Notation
{Fold 1, Fold 2, Fold 3, Fold 4}	Fold 5	Case 1
{Fold 1, Fold 2, Fold 3, Fold 5}	Fold 4	Case 2
{Fold 1, Fold 2, Fold 4, Fold 5}	Fold 3	Case 3
{Fold 1, Fold 3, Fold 4, Fold 5}	Fold 2	Case 4
{Fold 2, Fold 3, Fold 4, Fold 5}	Fold 1	Case 5

4.1 Extracting the image to two-dimension interval

In this study, we use gray-level co-occurrence matrix (GLCM) to extract the texture feature of image. The data obtained from the GLCM provide feature values that represent a specific image in intensity and neighborhood. The GLCM of an image refers to an $M \times N$ matrix, denoted as P, of size $g \times g$, here the value of g represents the gray scale used to construct the matrix. Each element $P(i, j)$ indicates the probability of gray level of pair i and j with a distance d and direction origin θ. The GLCM is computed by (13):

$$P(i,j) = \#\{((r,c),(r',c')) \in M \times N | d = d_E((r,c),(r',c')), \\ \theta = \Theta((r,c),(r',c')), I(r,c) = i, I(r',c') = j\}, \tag{13}$$

where

$d_E(.)$ is Euclidean distance between two pixels,

Θ is the angle formed by the vector $((r,c);(r',c'))$ and the unit vector,

$\#\{.\}$ is the force of the set.

After obtaining the GLCM for each image, we continue to calculate the feature intervals using (14):

$$([L_x - r_1/2, L_x + r_1/2][L_y - r_2/2, L_y + r_2/2]), \tag{14}$$

where r_1 and r_2 are the values of the uniform distribution in the range [1, 4]:

$$L_x = \frac{1}{N}\sum_{j}^{N}\left(\frac{1}{M}\sum_{i}^{M}(i)p_{d\theta}(i,j)\right); \\ L_y = \frac{1}{M}\sum_{i}^{M}\left(\frac{1}{N}\sum_{j}^{N}(j)p_{d\theta}(i,j)\right), \tag{15}$$

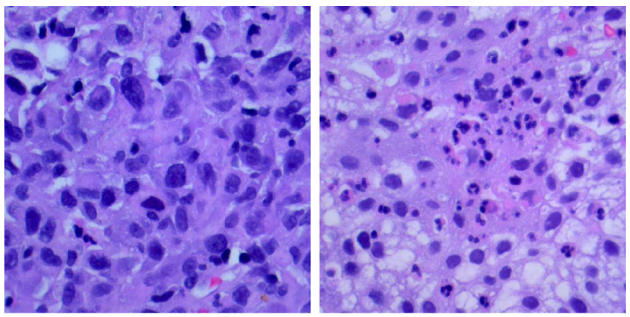
with M and N are the first and second size of each image, respectively, and the value of $P(i, j)$ is calculated by the formula (13). The formula (15) is considered as the average of the first and second dimension features of the matrix GLCM. By doing this, we generate the two-dimension interval according to the formula (14). Essentially, each image is then represented by a two-dimension interval. The steps to extract an image into a two-dimensional interval in this study are illustrated in Fig. 4.

4.2 Data and method for evaluation

To evaluate the effectiveness of the proposed algorithm, this study considers 4 medical image datasets. The datasets are obtained for free from website https://www.kaggle.

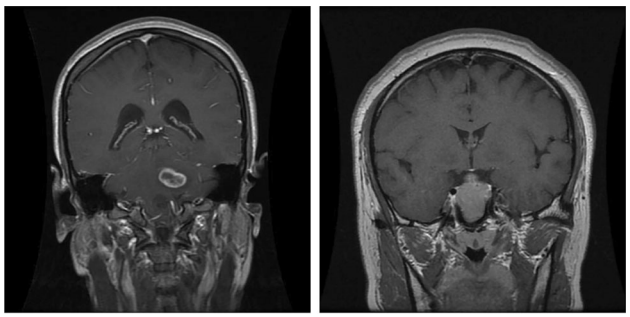
Table 2 General information about datasets

Data	No. of images	No. of train set	No. of test set	No of groups
1	8,955	7,164	1,791	2
2	15,000	12,000	3,000	2
3	13,808	11,047	2,761	2
4	3,256	2,605	651	4



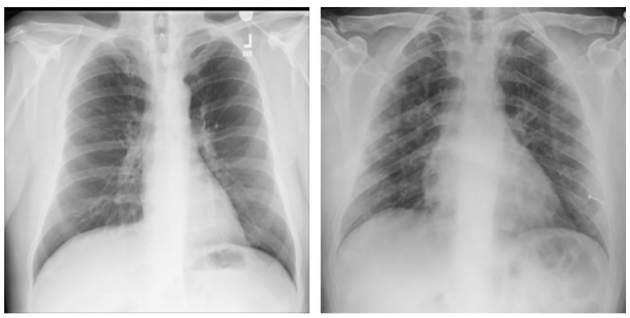
(a) LBT (b) LSC

Fig. 5 The image sample for two groups LBT and LSC



(a) Glioma (b) Pituitary

Fig. 6 The two image samples of glioma and pituitary tumor



(a) Covid19 (b) Non-Covid19

Fig. 7 Two image samples of people infected with Covid-19 and not infected with Covid-19

com. They are the datasets about Lung Coon Cancer histopathological image, Brain Cancer, Covid-19, and Acute Lymphoblastic Leukemia.

For evaluation, the study performs as follows. Divide each dataset into fivefold (Fold 1, Fold 2, Fold 3, Fold 4, Fold 5). Randomly, take fourfold as the training set and the remaining a fold as the test set. As a result, for each dataset, we perform 5 times for classification with the rate of 80% for the training set and 20% for the test set. The notation for the training and test sets for the 5 cases are presented in Table 1 and general information about datasets are given in Table 2.

Data 1 is Lung Coon Cancer histopathological image set. It includes two groups: benign lung tissue (LTP) with 5000 images and malignant tissue (LSC) with 3,955 images. Two sample images of two groups for Data 1 are illustrated in Fig. 5.

Data 2 is Multi-cancer images. In this study, we focus on glioma and pituitary tumor, two types of brain cancer, to compare classification algorithms. There are 10,000 images of gliomas and 5,000 images of pituitary tumors. Figure 6 illustrates sample images from the two groups.

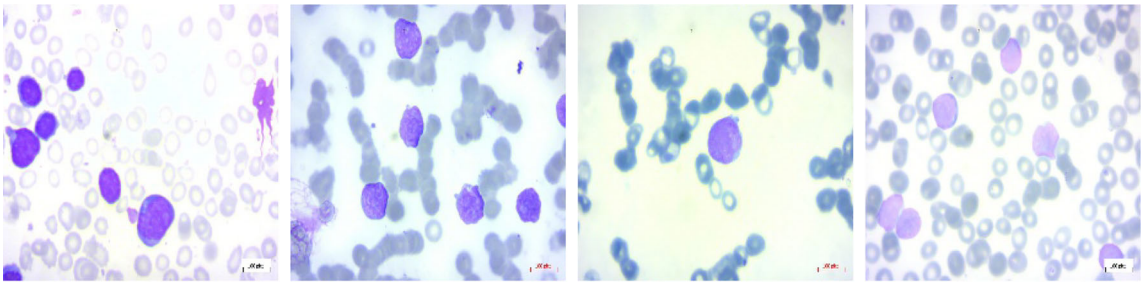
Data 3 has 13,808 lung images, with 3,616 images depicting people infected with Covid-19 and 10,192 images of people not infected with Covid-19. Two sample images from each group are provided in Fig. 7.

Data 4 classifies Acute Lymphoblastic Leukemia dataset with 3,256 images JPG. It is divided into 4 groups: Benign, Early, Pre, and Pro. Four sample images from each category are shown in Fig. 8.

The proposed algorithm is compared with others, including both statistical methods and machine learning methods such as Fisher, Logistic, Naive Bayes, SVM, QDA, XGBoost, kNN, Subspace-kNN, Subspace-Discriminant, Random Tree, AdaBoost, ANN, and Bayes. For the Bayesian method, after extracting the GLCM matrix, most subsequent studies used only four important features representing the texture (Celebi and Alpkocak 2000). This article employs four image features, including energy, uniformity, contrast, and correlation coefficient, to characterize each image (Laleh and Shervan 2019). The features for image are presented in Table 3.

where μ_i, μ_j are the mean of row i and column j in the GLCM matrix, respectively; δ_j, δ_j are the standard deviation of row i and column j in the GLCM matrix, respectively.

The proposed algorithm is compared to the Bayesian method in many cases. These include the Bayesian method



(a) Benign (b) Early (c) Pre (d) Pro

Fig. 8 Four groups of Acute Lymphoblastic Leukemia image dataset

Table 3 Four texture features of an image

Feature	Formula
Energy	$\sum_{i,j} p(i,j)^2$
Contrast	$\sum_{i,j} \lvert i-j\rvert^k p^l(i,j)$
Uniformity	$\sum_{i,j} \frac{p(i,j)}{1+\lvert i-j\rvert}$
Correlation coefficient	$\sum_{i,j} \frac{(i-\mu_i)(j-\mu_j)p(i,j)}{\delta_i \delta_j}$

when the pdfs are estimated using the kernel density method with a uniform prior, using the Laplace method, and using the ratio of contributions (denoted as Bayes K-U, Bayes-K-L, Bayes-K-T, respectively). The Bayesian method is also conducted when the pdfs are estimated using the Copula method with a uniform prior, using the Laplace method, and using the ratio of contributions (denoted as Bayes-C-U, Bayes-C-L, Bayes-K-T, respectively). In each case, the accuracy metric (ACC) of test set is used to compare approaches.

4.3 The result of performance

For Data 1, we present quite detailed results of the proposed algorithm to reach conclusions and compare them with other methods. For the remaining three datasets, we only present the comparative results of the proposed algorithm with other algorithms.

i) For Data 1: With this dataset, we evaluate the effectiveness of the methods according to fivefold as described above.

We consider the training set and test set of Case 1 which consists of 7164 LTP and 1791 LSC images, respectively. For the test set, there are 1000 images of LBT and 791 images of LSC.

After performing 8 iterations of computing the a prior probability to classify images into 2 groups and obtaining a partition matrix $U^{(8)}$ that satisfies the algorithm's stopping condition, we can determine the prior probability of the 1791 images in the test data illustrated in Fig. 9.

Performing Phase 2 of the proposed algorithm, we have the *PS* to classify 1791 images into 2 groups as illustrated in Fig. 10.

From Figs. 9 and 10, we can see that the prior probability and posterior similarity of two groups have very clear differences. This is the sign to get a good classification result.

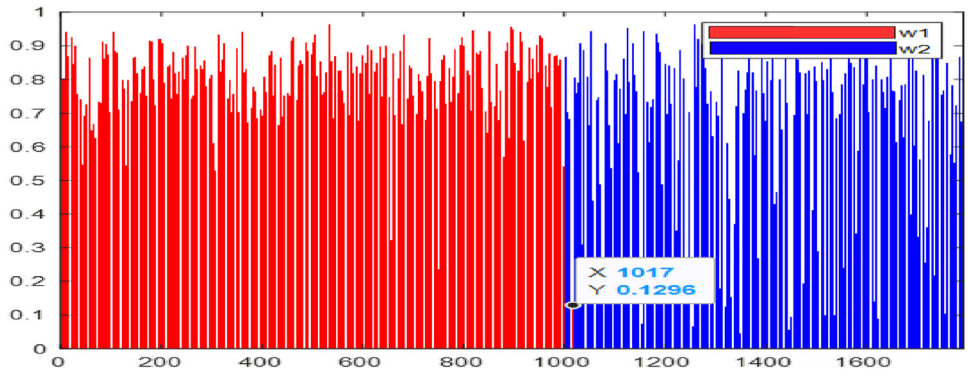


Fig. 9 The prior probability of the 1791 images in the test set to two groups

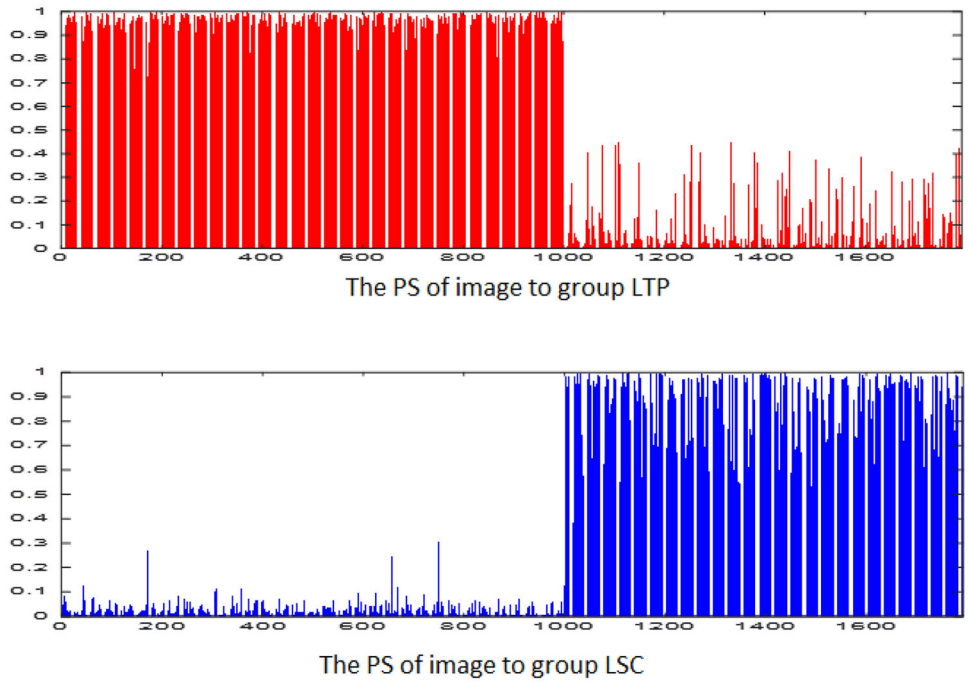


Fig. 10 The *PS* of classifying 1791 images into 2 groups

Table 4 The *ACC* in classifying Lung and Colon images of Case 1

Method	*ACC*	Method	*ACC*
Fisher (1938)	0.5812	Proposed-E	0.9978
Logistic (Bisong 2019)	0.5812	SVM (Huang et al. 2018)	0.8816
Naive Bayes (Yang 2018)	0.5783	QDA (Wu et al. 1996)	0.9811
BayesK-U	0.9676	XGBoost (Behera et al. 2022a)	0.9875
BayesK-T	0.9665	kNN (Imandoust and Bolandraftar 2013)	0.9814
BayesK-L	0.9665	Subspace-kNN (Ma et al. 2021)	0.9891
BayesC-U	0.9629	Subspace-Discriminant (Patil and Jalan 2022)	0.9811
BayesC-T	0.9682	Random forest (Liu et al. 2012)	0.9802
BayesC-L	0.9682	AdaBoost (Wang 2012)	0.9835
Proposed-C	0.9989	ANN (Bala and Kumar 2017)	0.9902
Proposed-H	0.9978	Proposed algorithm	0.9994

Applying the proposed classification rule, we have the following result for the test set:

* 1000 images of LBT are classified correctly.

* There is 1 misclassified image out of 701 images taken.

Therefore, the *ACC* of the proposed algorithm in this case is $1790/1791 = 0.9994$.

With the training set and testing set of Case 1, performing with other methods, we get the results in Table 4.

For other cases: we can perform the same process as in Case 1 for the remaining 4 cases, and then compare the *ACC* of algorithms. The result of comparison is presented in Table 5.

Table 5 demonstrates that the proposed algorithm consistently outperforms the other algorithms in all 5 cases. As a result, the average value of *ACC* for the proposed algorithm is the largest. With its accurate and consistent classification results, the proposed algorithm holds great potential for practical applications in classifying this data.

ii) For Data 2, Data 3 and Data 4: perform the same procedure as for *Data 1* and compare with other algorithms, we have the average *ACC* in Table 6.

According to Table 6, for Data 2, the values of *ACC* for the algorithms range from 81.9% to 94.23%. Approaches such as Fisher, Logistic Regression, Naive Bayes, QDA, and XGBoost do not yield satisfactory results with this dataset, as their average *ACC* is smaller than 90%. On the other hand, the remaining approaches demonstrate relatively good results, with *ACC* higher than 91%. Notably, the proposed algorithm has the highest average *ACC* at 94.23%, which is an outstanding result when compared to other approaches.

Table 5 The *ACC* in classifying Lung and Colon images of approaches

Method	Case 1	Case 2	Case 3	Case 4	Case 5	Average
Fisher (Fisher 1938)	0.5812	0.6929	0.9598	0.9570	0.9670	0.8849
Logistic (Bisong 2019)	0.5892	0.9716	0.9643	0.8777	0.9682	0.8726
Naive Bayes (Yang 2018)	0.5783	0.6982	0.9702	0.8803	0.9699	0.8194
BayesK-U	0.9676	0.9677	0.9676	0.9610	0.9609	0.9650
BayesK-T	0.9665	0.9665	0.9626	0.9604	0.9604	0.9633
BayesK-L	0.9665	0.9665	0.9626	0.9604	0.9604	0.9633
BayesC-U	0.9629	0.9622	0.9602	0.9590	0.9690	0.9607
BayesC-T	0.9682	0.9681	0.9679	0.9671	0.9621	0.9667
BayesC-L	0.9682	0.9681	0.9679	0.9671	0.9621	0.9667
Proposed-C	0.9989	0.9770	0.9927	0.9764	0.9877	0.9865
Proposed-H	0.9978	0.9754	0.9933	0.9754	0.9855	0.9855
Proposed-E	0.9978	0.9760	0.9939	0.9765	0.9877	0.9864
SVM (Huang et al. 2018)	0.8816	0.8967	0.8833	0.8777	0.8816	0.8842
QDA (Wu et al. 1996)	0.9811	0.9071	0.9108	0.9761	0.9174	0.9385
XGBoost (Behera et al. 2022a)	0.9875	0.9272	0.9274	0.9811	0.9211	0.9489
kNN (Ma et al. 2021)	0.9814	0.9173	0.9213	0.9802	0.9199	0.9442
Subspace-kNN (Ma et al. 2021)	0.9891	0.9324	0.9283	0.9831	0.9346	0.9440
Subspace-Discriminant (Patil and Jalan 2022)	0.9811	0.9079	0.9175	0.9781	0.9191	0.9407
Random forest (Liu et al. 2012)	0.9802	0.9022	0.9121	0.9711	0.9181	0.9348
AdaBoost (Wang 2012)	0.9835	0.9211	0.9202	0.9868	0.9181	0.9465
ANN (Bala and Kumar 2017)	0.9902	0.9802	0.9485	0.9897	0.9825	0.9782
Proposed algorithm	**0.9994**	**0.9855**	**0.9961**	**0.9811**	**0.9894**	**0.9903**

Table 6 The *ACC* of approaches for Data 1, Data 2 and Data 3

Method	Data 2	Data 3	Data 4
Fisher (Fisher 1938)	0.8190	0.7266	0.7382
Logistic (Bisong 2019)	0.8924	0.6784	0.6308
Naive Bayes (Yang 2018)	0.8332	0.7115	0.6815
BayesK-U	0.9192	0.8128	0.7135
BayesK-T	0.9247	0.8707	0.7002
BayesK-L	0.9247	0.8744	0.7027
BayesC-U	0.9175	0.8098	0.7145
BayesC-T	0.9261	0.8894	0.7158
BayesC-L	0.9247	0.8907	0.7279
Proposed-C	0.9362	0.9186	0.9028
Proposed-H	0.9385	0.9259	0.9129
Proposed-E	0.9333	0.9259	0.9189
SVM (Huang et al. 2018)	0.9216	0.7674	0.7382
QDA (Wu et al. 1996)	0.8894	0.8129	0.8391
XGBoost (Behera et al. 2022a)	0.8859	0.8118	0.8492
kNN (Ma et al. 2021)	0.9233	0.9144	0.8121
Subspace-kNN (Ma et al. 2021)	0.9273	0.9358	0.8390
Subspace-Discriminant (Patil and Jalan 2022)	0.9366	0.9384	0.8715
Random forest (Liu et al. 2012)	0.9158	0.9142	0.8984
AdaBoost (Wang 2012)	0.9125	0.9085	0.8672
ANN (Bala and Kumar 2017)	0.9375	0.9088	0.7224
Proposed algorithm	0.9423	0.9428	0.9379

In comparison to Data 1 and Data 2, the *ACC* for Data 3 is lower. Table 6 displays the average *ACC* of the methods, ranging from 67.84% to 94.28%. Methods such as Logistic Regression, Naive Bayes, and SVM have *ACC* values less than 80%. The approaches with *ACC* values smaller than 90% and larger than 80% include BayesK-U, BayesK-T, BayesK-L, BayesC-U, BayesC-L, BayesC-T, QDA, and XGBoost. The *ACC* of the remaining approaches is larger than 90%, with the proposed algorithm giving the highest value.

For Data 4, the *ACC* is generally lower than the three datasets considered above. Many algorithms exhibit *ACC* values below 74%, such as Fisher, Logistic Regression, Naive Bayes, BayesK-U, BayesK-T, BayesK-L, BayesC-U, BayesC-L, BayesC-T, SVM, and ANN. Only three methods have *ACC* values above 90%, among which the proposed method has the highest value with an *ACC* of 93.79%.

For the 4 datasets examined in the medical field, which vary in characteristics, sample size, and number of groups, we observe that the proposed algorithm demonstrates stability in its results. It exhibits relatively good experimental classification accuracy, being competitive with many other methods. However, similar to many other classification methods, the strong competitiveness of the proposed algorithm is not universal across all image datasets. It depends on various factors of the images, which we will delve into more deeply in the near future.

5 Conclusion

While scientists have extensively explored classification methods for discrete data, the interest and exploration of such methods for interval data remain relatively limited. This study introduced an effective classification algorithm tailored specifically for interval data. Its efficacy stemmed from a combination of three pivotal factors. First, it employed a pertinent metric known as overlap distance to evaluate similarities between intervals and between an interval and a cluster. In addition, the method utilized fuzzy cluster analysis techniques to determine prior probabilities for classification elements, thereby enhancing classification efficiency. Finally, the classification principle hinged on maximizing the prior probability while minimizing the distance between the classified element and the groups, thereby elevating the likelihood of accurate classification. A significant highlight of this study involved the application of the proposed algorithm to images, wherein the gray-level co-occurrence matrix of each image is transformed into a two-dimensional interval. Furthermore, the algorithm was employed across various medical image datasets showcasing diverse properties, exhibiting superior performance compared to alternative approaches. The experiments underscored the stability and high correct classification ratio of the proposed method, which could be swiftly implemented using established Matlab procedures on different datasets. In the future, the proposed classification algorithm will be further applied to real-life problems in various domains.

Acknowledgements This research is funded by Ministry of Education and Training in Vietnam under grant number B2023-TCT-06.

Data Availability The datasets analyzed during this study are openly available from the public data on the website, and given specifically in the article.

Declarations

Conflict of interest No potential conflict of interest was reported by the authors.

References

Bala R, Kumar D (2017) Classification using ANN: A review. Int J Comput Intell Res 13(7):1811–1820

Behera DK, Das M, Swetanisha S (2022) Follower link prediction using the XGBoost classification model with multiple graph features. Wirel Pers Commun 127:695–714

Behera TK, Khan MA, Bakshi S (2022) Brain MR image classification using superpixel-based deep transfer learning. IEEE J Biomed Health Inform. https://doi.org/10.1109/JBHI.2022.3216270

Bisong E (2019) Logistic regression. Building machine learning and deep learning models on google cloud platform. Apress, Berkeley, pp 243–250

Brito P (2007) Modelling and analysing interval data. In: Decker R, Lenz HJ (eds) Advances in data analysis. Studies in classification, data analysis, and knowledge organization. Springer, New York, pp 197–208

Celebi E, Alpkocak A (2000) Clustering of texture features for content-based image retrieval. International Conference on Advances in Information Systems. Springer, New York, pp 216–225

Chen Y, Liu C, Chou K, Wang S (2016). Real-time and low-memory multi face detection system design based on naive Bayes classifier using FPGA. In: International Automatic Control Conference (CACS), Berlin. p 7-12

Dietterich T (2000) An experimental comparison of three methods for constructing ensembles of decision trees: Bagging, Boosting, and Randomization. J. Mach Learn Mach Learn 40(2):139–157

Fisher RA (1938) The statistical utilization of multiple measurements. Ann Eugen 8(4):376–386

Gou J, Du L, Zhang Y, Xiong T (2012) A new distance-weighted k-nearest neighbor classifier. J. Inf. Comput. Sci. 9(6):1429–1436

Ha CN, Thao NT, Nguyen BT, Trung NT, Tai VV (2022) A new approach for face detection using the maximum function of probability density functions. Ann Oper Res 312:99–119

Huang S, Cai N, Pacheco PP, Narrandes S, Wang Y, Xu W (2018) Applications of support vector machine (SVM) learning in cancer genomics. Cancer Genom Proteom 15(1):41–51

Huynh-Van H, Le-Hoang T, Thai-Minh T, Nguyen-Dinh H (2023) Classifying the lung images for people infected with Covid-19 based on the extracted feature interval. Comput. Methods Biomech. Biomed. Eng.: Imaging Vis. 11(3):856–865

Huynh-Van H, Le-Hoang T, Vo-Van T (2023) Classifying for images based on the extracted probability density function and the quasi Bayesian method. Comput Stat. https://doi.org/10.1007/s00180-023-01400-1

Imandoust SB, Bolandraftar M (2013) Application of k-nearest neighbor (KNN) approach for predicting economic events: theoretical background. Int. J. Eng. Res. 3(5):605–610

Laleh M, Shervan FE (2019) Texture image analysis and texture classification methods - a review. Int. J. Image Process. Pattern Recognit. 2(1):1–29

Lethikim N, Nguyentrang T, Vovan T (2022) A new image classification method using interval texture feature and improved Bayesian classifier. Multimed. Tools. Appl. 81:36473–36488

Le KNT, Nguyenthihong D, Vovan T (2023) Fuzzy cluster analysis algorithm for image data based on the extracted feature intervals. Granul. Comput. 8:2067–2081

Liu Y, Wang Y, Zhang J (2012) New machine learning algorithm: random forest. In: Liu B, Ma M, Chang J (eds) Information computing and applications. ICICA 2012. Lecture notes in computer scienceb, vol 7473. Springer, Berlin, Heidelberg, pp 245–261

Ma X, Yang T, Chen J, Liu Z (2021) k-Nearest Neighbor algorithm based on feature subspace. In: International Conference on Big Data Analysis and Computer Science (BDACS). Kunming, China. p 225–228.

Mejdoub H, Arab MB (2018) Impact of dependence modelling of non-life insurance. Res Int Bus Finance 45:208–218

Miller G, Inkret WC, Little TT, Martz HF, Schillaci ME (2001) Bayesian prior probability distributions for internal dosimetry. Radiat Prot Dosimetry 94(4):347–52

Neto JG, Ozorio LV, De Abreu TCC, Dos Santos BF, Pradelle F (2021) Modeling of biogas production from food, fruits and vegetables wastes using artificial neural network (ANN). Fuel 285:119081

Ngoc L, Tuan Lh, Tai V (2023) Automatic clustering algorithm for interval data based on overlap distance. Commun Stat Simul Comput 52(5):2194–2209

Nhu VH, Zandi D, Shahabi H, Chapi K, Shirzadi A, Al-Ansari N, Singh SK, Dou J, Nguyen H (2020) Comparison of support vector machine, Bayesian logistic regression, and alternating decision tree algorithms for shallow landslide susceptibility mapping along a mountainous road in the west of Iran. Appl Sci 10(15):5047

Nguyentrang T, Vovan T (2017) A new approach for determining the prior probabilities in the classification problem by Bayesian method. Adv Data Anal Classif 11:629–643

Nguyentrang T, Nguyenthoi T, Vovan T (2023) Globally automatic fuzzy clustering for probability density functions and its application for image data. Appl Intell 53:18381–18397

Nguyentrang T, Nguyenthoi T, Nguyenthi KN (2023) Balance-driven automatic clustering for probability density functions using metaheuristic optimization. Int J Mach Learn Cybern 14:1063–1078

Patil S, Jalan AK (2022) Ensemble subspace discriminant classifiers for misalignment fault classification Using Vibro-acoustic Sensor Data Fusion. J. Vib. Eng. Technol. 10:3169–3178

Pham-Gia T, Turkkan N, Vovan T (2000) Statistical discrimination analysis using the maximum function. Commun Stat Simul Comput 37(2):320–336

Pham-Gia T, Turkkan N, Bekker A (2006) Bounds for the Bayes error in classification: A Bayesian approach using discriminant analysis. J. Ital. stat. soc. 16(1):7–26

Phamtoan D, Vovan T (2021) Automatic fuzzy genetic algorithm in clustering for images based on the extracted intervals. Multimed. Tools. Appl. 80:35193–35215

Phamtoan D, Nguyenhuu K, Vovan T (2022) Fuzzy clustering algorithm for outlier-interval data based on the robust exponent distance. Appl Intell 52:6276–6291

Phamtoan D, Vovan T (2023) The fuzzy cluster analysis for interval value using genetic algorithm and its application in image recognition. Comput Stat 38:25–51

Sklar M (1959) Fonctions de repartition n dimensions et leurs marges. Université Paris 8(8):229–231

Singh S, Ganie AH (2022) Applications of a picture fuzzy correlation coefficient in pattern analysis and decision-making. Granul. Comput. 7:353–367

Terrell GR, Scott DW (1992) Variable kernel density estimation. The Ann. Stat. 20(3):1236–1265

Verma R, Rohtagi B (2023) Novel similarity measures between picture fuzzy sets and their applications to pattern recognition and medical diagnosis. Granul. Comput. 7:761–777

Vovan T, Pham-Gia T (2010) Clustering probability distributions. J Appl Stat 37(11):1891–1910

Vovan T (2016) L^1-distance and classification problem by Bayesian method. J Appl Stat 44(3):385–401

Vovan T, Nguyentrang T (2017) Cluster similar of cluster for probability density functions. Commun. Stat. Theory Methods 47(8):1792–1811

Vovan T (2018) Some results of classification problem by Bayesian method and application in credit operation. J Stat Theory Pract. 2(2):150–157

Vovan T, Tranphuoc L, Chengoc H (2019) Classifying two populations by Bayesian method and applications. Commun. Math. Stat. 7(2):141–161

Vovan T, Lekim N, Nguyentrang T (2023) An efficient robust automatic clustering algorithm for interval data. Commun Stat Simul Comput 52(10):4621–4635

Vovan T, Chengoc H, Ledai N (2022) A New Strategy for short-term stock investment using Bayesian approach. Comput Econ 59:887–911

Vovan T (2023) Building the forecasting model for interval time series based on the fuzzy clustering technique. Granul. Comput. 8:1341–1357

VijayaLakshmi B, Mohan V (2016) Kernel-based PSO and FRVM: An automatic plant leaf type detection using texture, shape, and color features. Comput Electron Agric 125:99–112

Wang R (2012) AdaBoost for Feature Selection, Classification and Its Relation with SVM: A Review. Phys Procedia 25:800–807

Wyner AJ, Olson M, Bleich J, Mease D (2017) Explaining the success of AdaBoost and random forests as interpolating classifiers. J Mach Learn Res 18(48):1–33

Wu W, Mallet Y, Walczak B, Penninckx W, Massart DL, Heuerding V, Erni F (1996) Comparison of regularized discriminant analysis linear discriminant analysis and quadratic discriminant analysis applied to NIR data. Anal Chim Acta 329(3):257–265

Yamashita R, Nishio M, Do RKG (2018) Convolutional neural networks: an overview and application in radiology. Insights Imaging 9:611–629

Yang FJ (2018) An Implementation of Naive Bayes Classifier. 2018 International Conference on Computational Science and Computational Intelligence (CSCI), Las Vegas, NV, USA, 2018, pp. 301-306

Yuan W, Xiaoqian J, Jihoon K, Lucila OM (2012) Grid binary Logistic regression glore: building shared models without sharing data. J Am Med Inform Assoc 19(5):758–764

Zhuang SJ, Lin CJ (2023) Defect classification of glass substrate using deep neuro-fuzzy network with optimal parameter combination. Granul. Comput. 8:839–849

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Springer Nature or its licensor (e.g. a society or other partner) holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.