Journal of Artificial Intelligence and Data Mining (JAIDM), Vol. 11, No. 3, 2023, 391-405.

Shahrood University of Technology

Journal of Artificial Intelligence and Data Mining (JAIDM)
Journal homepage: http://jad.shahroodut.ac.ir

Research paper

Tree Bark Classification using Color-improved Local Quinary Pattern and Stacked MEETG

Laleh Armi and Elham Abbasi*

Department of Computer Science, Yazd University, Yazd, Iran.

Article Info

Article History:
Received 08 February 2023
Revised 23 April 2023
Accepted 04 July 2023

DOI:10.22044/jadm.2023.12692.2420

Keywords:
Improved Local Quinary Pattern, Extreme Learning Machine, Ensemble Learning, Bark Classification

* *Corresponding author: e.abbasi@yazd.ac.ir (E. Abbasi).*

Abstract

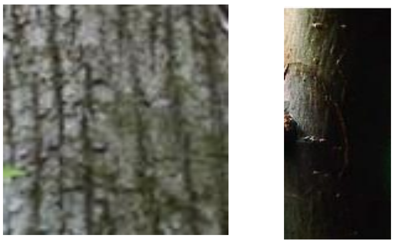
In this paper, we propose an innovative classification method for tree bark classification and tree species identification. The proposed method consists of two steps. In the first step, we take the advantages of ILQP, a rotationally invariant, noise-resistant, and fully descriptive color texture feature extraction method. Then in the second step, a new classification method called stacked mixture of ELM-based experts with a trainable gating network (stacked MEETG) is proposed. The proposed method is evaluated using the Trunk12, BarkTex, and AFF datasets. The performance of the proposed method on these three bark datasets shows that our approach provides a better accuracy than other state-of-the-art methods. Our proposed method achieves an average classification accuracy of 92.79% (Trunk12), 92.54% (BarkTex), and 91.68% (AFF), respectively. Additionally, the results demonstrate that ILQP has better texture feature extraction capabilities than similar methods such as ILTP. Furthermore, stacked MEETG has shown a great influence on the classification accuracy.

1. Introduction

Identifying tree species from bark images is a challenging problem in the field of computer vision. This project can be a practical and valuable project for assessing the condition of forests and natural resources, as well as for environmental protection. The benefits that can be mentioned are as the following:

- Time and cost reduction: Automated tree bark detection systems can reduce the time and cost associated with manual inspection and imaging of trees.
- High accuracy: The tree bark detection system based on artificial intelligence and machine learning algorithms can accurately detect tree bark with increasing the confidence of inspection results.
- Increased information: Detecting tree bark can provide more information about tree diseases, pests, growth rates, and forest conditions, which can aid in planning and decision-making for environmental protection and forest monitoring.
- Technological advancement: The tree bark detection project can serve as a foundation for developing other automated systems for forest and environmental monitoring.
- Environmental protection: Early detection of tree diseases and pests can prevent their spread, and increasing vegetation can help preserve biodiversity and protect the environment.

To identify and classify the type of trees, various parameters such as the overall shape and size of the tree, bark, leaves or needles, flowers, fruits, leaf buds, and twigs are used [1]. As mentioned, bark is one way to identify trees, which is the outer protective coating of the trunk and branches of trees that may appear gray and brown, but if looked at closely, changes in color and texture can be observed. There are various patterns, textures, and other characteristics of bark that can help identify trees. According to Michael Wojtech in [2], "If you want to know the trees, learn their bark". Forester can recognize the species of trees by differences in their bark either externally or by cutting a small slash to examine the inner structure. Experts also believe that tree bark has a stronger correlation with species compared to other phenotypic properties [3]. Thus bark is a useful diagnostic feature for plant classification. Recognizing tree species is a challenging problem that can aid in drone navigation in forest environments and autonomous management of forest inventory. Tree bark usually has a specific texture that can be used to classify tree species. As you can see in Figure 1(a), the basswood's bark is brown/gray with deep vertical fissures and in Figure 1(b), a crab apple's bark is reddish/brown, flat ridges, shallow fissures with broad flat topped scaly ridges.



(a) (b)

Figure 1. Two examples of tree bark (a) basswood's bark and (b) crab apple's bark.

Thus bark is a useful diagnostic feature for plant classification and identification, which is one of the topics that is regarded by the researchers [4].

For example, in [5] Fiel and Sablatnig have proposed a method for identification of tree species from images of the bark, leaves, and needles. For bark description, they used the scale invariant feature transform (SIFT) with bag of words approach, afterwards a combination of the gray-level co-occurrence matrices (GLCM) features (contrast, correlation, energy, and homogeneity) and wavelet features have been applied as the input of support vector machine (SVM) on the AFF dataset.

Bressane et al. [6] have extracted four statistical parameters (uniformity, entropy, asymmetry, and smoothness) used in texture classification of trunk images, and have employed a decision tree for classification. In [7], Boudra el al. have proposed a rotational invariant statistical radial binary pattern (SRBP) descriptor to characterize a bark texture.

In [8], Le-Viet et al. have presented gradient local binary pattern (GLBP) to encode the local texture of bark image. In addition to encoding, magnitude, and orientation gradient is used to create the second and the third histogram. Also they have applied k nearest neighbor classifier to classify the bark images.

Ratajczak et al. have proposed two novel algorithms for bark classification based on combination of texture and color information to reduce their dimensionality [9]. Light combination of local binary pattern (LCoLBP) in combination with color histogram descriptor provides highest accuracy [9]. A patch-based convolution neural network has been proposed by Debaleena Misra et al. [10] for the identification of tree species from bark images. Fekri-Ershad [1] proposed a method for bark texture classification based on the improved local ternary patterns (ILTP). In [1], MLPs are used as a classifier and also four different strategies are applied to evaluate the number of neurons in hidden layers.

A single learning algorithm performs better than all other algorithms for a particular problem according to both empirical studies and specific machine learning applications [11-14].

As a result, applying multiple classifiers and combination their output called as an ensemble learning system is an effective approach to improve the accuracy and the reliability of the overall learning system [11-14].

Mixture of extreme learning machine based experts with trainable gating network (MEETG) [15, 16] is a mixture of experts (ME) based ensemble learning method. In MEETG, the superiorities of ELM have been taken for designing the architecture and training process of ME. ELM has been considered due to its high generalization ability, low training time, high accuracy, and performance, reducing the likelihood of overfitting and its ability to overcome the problems and limitations of the backpropagation algorithm. Furthermore, in MEETG, a dynamic strategy has been used for the combination of the experts' output according to the input sample, which is performed by the gating network. Furthermore, in this paper, we extend the capabilities of MEETG and propose "Stacked MEETG" for tree bark classification and tree species identification, in which a meta-classifier has been applied to aggregate the outputs of the base experts. The general framework of Stacked MEETG consists of two levels. In the first level, which is the base learning stage, MEETGs act as base experts and are trained on different feature spaces. These feature spaces are extracted from tree bark images using the improved local quinary pattern (ILQP) descriptor. Since some texture descriptors are sensitive to noise and rotation, we have used the ILQP descriptor that is resistant to these changes.

In the second level, meta-learning, a meta-classifier is trained on the outputs of the first-level

classifiers to learn how to aggregate the predictions of base experts. In this paper, we take advantage of ELM as the meta-classifier. The output of base experts is considered as the output of the hidden layer neurons of ELM, and the Moore-Penrose pseudo-inverse method is applied for adjusting the output weights of ELM. The trained weights of ELM determine how much each expert contributes to the final classification.
The proposed approach has demonstrated superior classification accuracy compared to well-known methods on three benchmark datasets in the experimental results. Additionally, our method offers the advantages of being noise-resistant and rotation-invariant. In continuation, for the classification of tree bark images, the choice of classifier is also an important factor. Ensemble learning methods, in which the output results of multiple classifiers are combined, are a good approach to improve the performance of classification. To the best of our knowledge, in most previous works, tree bark classification is typically performed using a single classifier. Therefore, in this paper, we propose a hybrid ensemble system called stacked MEETG.
As a result, our approach can be effectively utilized in real-world applications, leading to reduced financial costs and human risks associated with plant species diagnosis. Specifically, our bark texture classification method shows promise in this regard.

The paper is organized as what follows. In the next section, the primary concepts will be overviewed. In Section 3, the proposed bark texture classification approach is described. In Section 4, the simulation results are reported, and finally, a general conclusion is provided in Section 5.

2. Preliminaries

In this section, improve local quinary pattern (ILQP) and mixture of ELM based experts with trainable gating network (MEETG) are briefly described.

2.1. Improve local quinary pattern

Local bainary pattern (LBP) is one of the most powerful and widely used local descriptor has been introduced by Ojala et al. [17, 18] for texture classification. In this descriptor, a neighborhood is considered for each point of the image. Then intensity value of each neighborhood is compared with the center for building the binary pattern. It generates a binary code 0 if the value of neighbor pixel is smaller than the center value of patch; otherwise, it generates a binary code 1. Finally, with a binary weighted sum of the values in the binary extraction patterns are obtained values at the base of ten [18].
The original form of LBP has some disadvantages such as low discrimination and high computational complexity, and with increasing number of neighboring points this computational complexity increases. The first modification of LBP, called the uniform, has been introduced by Ojala [19]. One of the weaknesses of LBP and modified version of it is that in cases with different structures, the same binary code is generated. In other words, if there is one or one hundred unit's intensity difference between a neighbor pixel and the center pixel, there is no distinction between the two intensity differences. One of the drawbacks of LBP is sensitivity to noise, because a small gray change of the central pixel may cause different codes for a neighborhood in an image, especially for the smooth regions. To overcome this weakness, several versions have been proposed, uch as completed LBP (CLBP) [20], local ternary pattern (LTP) [21], enhanced LTP (ELTP) [22], e local binary count (LBC) [23], improved LTP(ILTP) [1], local quainary patterns (LQP) [24] exc.
improved local quainary patterns(ILQP) is one of the extended versions of LBP [25]. The advantages of ILQP are low sensitivity to impulse noise and illumination and towards scale and rotation invariant. In addition, lower number of features and ability to expound any of the features are advantages of this descriptor [25].
In ILQP descriptor, for each pixel of an image, whit using two threshold dynamics a five-value coding is computed which is codes between 2 and -2 by $S(g_p, g_c)$ function is defined as follows:

$$S(g_p,g_c)=\begin{cases} 2 & g_p \ge g_c+\tau_2 \\ 1 & g_c+\tau_1 \le g_p < g_c+\tau_2 \\ 0 & g_c-\tau_1 \le g_p < g_c+\tau_1 \\ -1 & g_c-\tau_2 \le g_p < g_c-\tau_1 \\ -2 & g_p < g_c-\tau_2 \end{cases} \tag{1}$$

where g_c denotes the intensity value of the central pixel and g_p is the intensity value of neighboring and threshold τ_1 and τ_2 are thresholds named (MAD) and global significant value (GSV). MAD and GSV calculations are defined as follows:

$$LocalMAD = median(|G - median(G)|), \quad G=\{\ g_p \mid p=0,1,\ldots,P-1\ \} \tag{2}$$

$$MAD = median(|LM - median(lmad)|), \quad LM=\{\ LocalMAD_c \mid c=1,2,\ldots,M\times N\ \} \tag{3}$$

$$LSV = \frac{1}{P}\sum_{p=0}^{P-1}(|g_c - g_p|) \tag{4}$$

$$GSV = \frac{1}{N \times M}\sum_{i=1}^{N}\sum_{j=1}^{M} LSV_{i,j} \tag{5}$$

According to (2-(5), P refers to the number of neighboring pixels considered in the pattern, while R refers to the radius of the circle centered on the central pixel used to define the neighboring pixels. Together, P and R determine the size and shape of the local neighborhood around each pixel that is used to calculate the ILQP code.

$g_p(p=0,1,\dots,P-1)$ denotes the gray value of the neighbor, G is the set of the gray-level values in g_p local region and N, M show size of image.

Next, five-value code is divided strongly binary positive pattern, positive binary pattern, negative binary pattern and strongly negative binary pattern, according to (6) [25]. **Error! Reference source not found.** shows an example of splitting a quinary code into four binary codes.

$$ILQP_{ch} = \begin{cases} ILQP_{strongly\ positive\ pattern} = \begin{cases} 1 & if \quad x=2 \\ 0 & otherwise \end{cases} \\ ILQP_{positive\ pattern} = \begin{cases} 1 & if \quad x=2 \ or \ x=1 \\ 0 & otherwise \end{cases} \\ ILQP_{negative\ pattern} = \begin{cases} 1 & if \quad x=-2 \ or \ x=-1 \\ 0 & otherwise \end{cases} \\ ILQP_{strongly\ negative\ pattern} = \begin{cases} 1 & if \quad x=-2 \\ 0 & otherwise \end{cases} \end{cases} \tag{6}$$

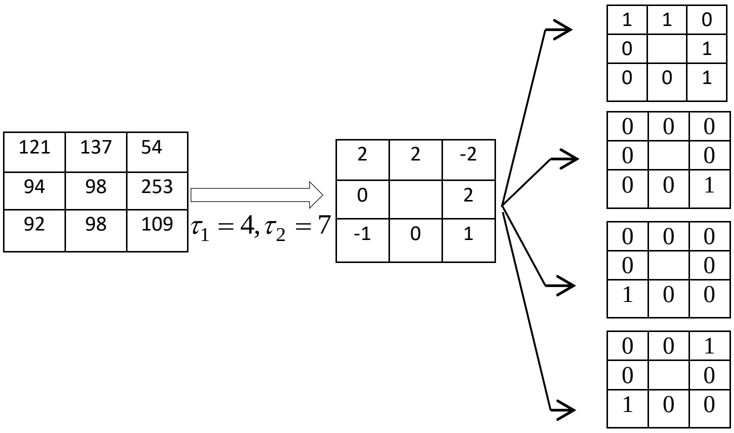


Figure 2. An example of splitting a quinary code into four binary codes.

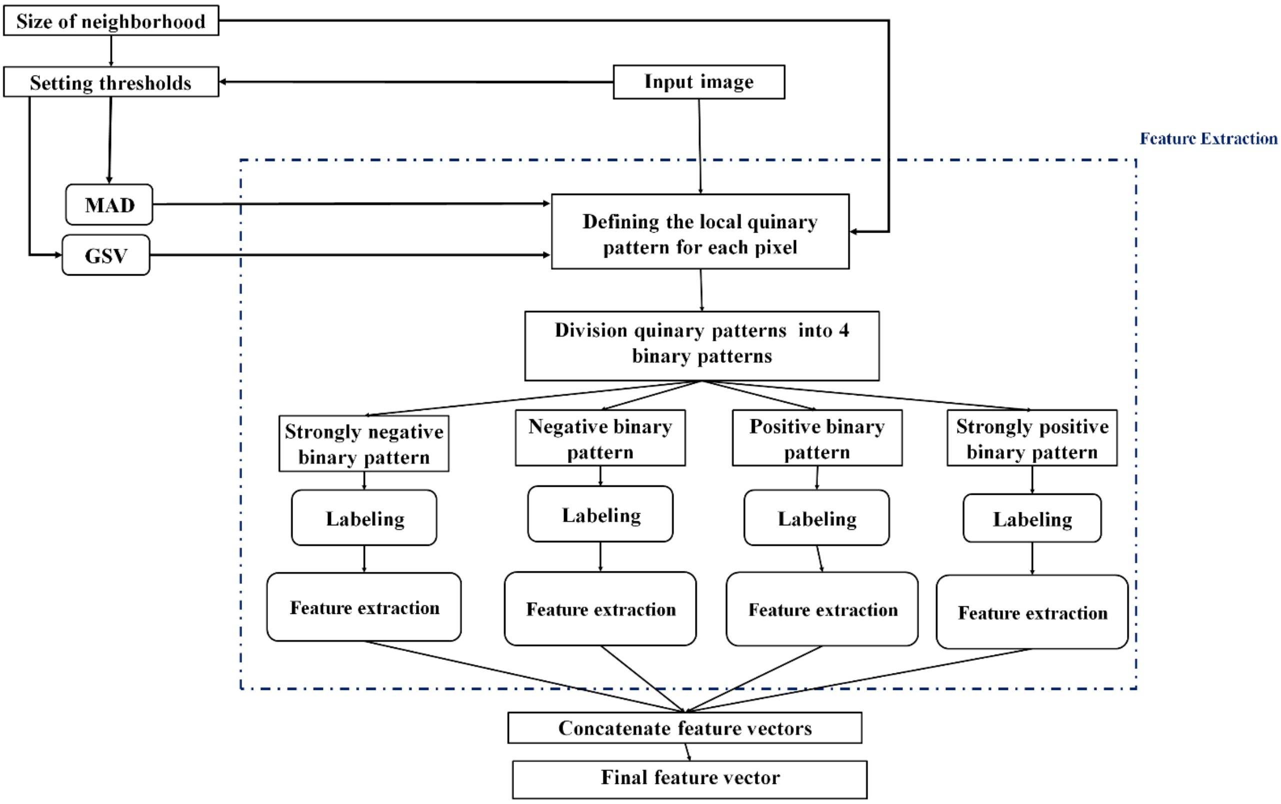


Figure 3. Block diagram of the ILQP method [25].

According to Figure 2, for converting quinary pattern (2221010) into code strongly positive pattern (1101000), it is considered by observing 2 and 1 in the quinary pattern. In owther words, this pattern shows the difference between the intensity values of neighboring pixels and center is greater than τ_2 [25].

According to (7), the degree of uniformity is calculated for each of these four binary patterns. In this equation, the number of mutations that occurred is calculated from 0 to 1 and vice versa in the binary pattern extracted from In the following, uniform neighbors are assigned labels from 0 to P, and non-uniform neighbors are assigned labels of $P+1$.
Finally, the probability of occurrence of each label in the whole image is considered a feature.

$$U(ILQP_{P,R,ch}) = |(g_{p-1}^{ch}) - (g_0^{ch})| + \sum_{p=0}^{P-1} |(g_p{}^{ch}) - (g_{p+1}{}^{ch})| \tag{7}$$

$$ILQP_{P,R,ch}^{riu} = \begin{cases} \sum_{p=0}^{P-1} (g_p{}^{ch}) & if \quad U(ILQP_{P,R,ch}) \le \frac{P}{4} \\ P+1 & otherwise \end{cases} \tag{8}$$

P and g_p^{ch} are, respectively, the number of neighbors and the value of the p^{th} neighbor in the extracted binary pattern. The ch index corresponds to each of four patterns (strongly positive, positive, negative and strongly negative). Thus feature vector for each pattern is defined as follows:

$$F_{ILQP_{ch}} = \left\langle F_0, F_1, \cdots, F_P, F_{(P+1)} \right\rangle \tag{9}$$

Due to concatenation of four feature vectors with dimension $P+2$; a feature vector with dimension $4 \times (P+2)$ will be extracted.

$$FinalFeatureVector_{ILQP} = \langle F_{ILQP_{stronglypositive}}, F_{ILQP_{positive}}, F_{ILQP_{snegative}}, F_{ILQP_{sstronglynegative}} \rangle \tag{10}$$

2.2. Mixture of ELM-based experts with trainable gating network (MEETG)

Mixture of experts (ME) has been introduced by Jacobs et al. [26] as one of the most popular ensemble methods. The architecture of this method consists of several experts and a gating network that can improve the accuracy of complex problems based on the divide and conquer principle. In ME, the input space of the problem is decomposed, and the experts are assigned to different sub-spaces by managing the gating network.
Besides the benefits of ME, there are some drawbacks. In ME, MLP is used as the experts and gating network, and gradient descent-based algorithms are used for training the model. One of the drawbacks of gradient descent-based algorithms is their dependency on parameter initialization and the complexity of the feature space, which may prevent them from always finding the global best solution and cause them to converge to a local minimum. Additionally, these algorithms require an iterative learning process that can be time-consuming.
Huang et al. proposed the Extreme Learning Machine (ELM) [27], a learning algorithm for Single Hidden Layer Feed-forward Neural Networks (SLFNs). In ELM, the weights between the input layer and hidden layer are assigned and do not need to be tuned. Additionally, the weights between the hidden layer and output layer are updated in a single step.
In order to overcome the limitations of ME, MEETG was proposed [15, 16], which takes advantage of ELM in designing its structure.
Given training set $D = \{(x_i, y_i)_{i=1,2,\cdots,N_{train}}\}$ and $x_i = [x_{i1}, x_{i2}, \cdots, x_{id}]^T$ is the input vector and $y_i = [y_{i1}, y_{i2}, \cdots, y_{im}]^T$ is the output vector. This input data is received by the experts and gating network. The parameters generated (weights and bias) between the input layer and the hidden layer of each expert are set randomly and besides with L hidden nodes and activation function $G(w_j, b_j, x_i)$ of experts and gating network in the form of $\sum_{i=1}^{L} \beta_i.G_i$ are made.
At first, the output matrix of hidden layer of each of expert and gating network is calculated according to (11).

$$H = \begin{bmatrix} G(w_1, b_1, x_1) & \cdots & G(w_L, b_L, x_1) \\ \vdots & \ddots & \vdots \\ G(w_1, b_1, x_{N_{train}}) & \cdots & G(w_L, b_L, x_{N_{train}}) \end{bmatrix}_{N_{train} \times L} \tag{11}$$

$$G(w_j, b_j, x_i) = \frac{1}{1 + exp(-(w_j.x_i + b_j))} \tag{12}$$

$\beta_i = [\beta_{i1}, \beta_{i2}, \cdots, \beta_{im}]^T$ is the weight vector connecting the ith hidden node and the output nodes, that weights between hidden layer and output layer of each expert are calculated according to (13).

$$\beta = H^{\dagger}.Y \tag{13}$$

where $H^{\dagger}$ is Moore-Penrose generalized inverse of matrix H and Y is target matrix of training data [28]. Also the target vectore of gating network which is used for training process corresponding to one sample is given in (14), which shows to what extent each expert can produce desired output y.

$$y_{Gating} = \left[\frac{\exp\|y - O^{Expert1}\|^2}{\sum_{i=1}^{k} \exp\|y - O^{Experti}\|^2}, \cdots, \frac{\exp\|y - O^{Expertk}\|^2}{\sum_{i=1}^{k} \exp\|y - O^{Experti}\|^2} \right] \tag{14}$$

where y is desired output and $O^{Experti}$ is the output of expert i . Y_{Gating} as the target matrix of the gating network for N_{train} training samples and k experts, is given in (15).

$$Y_{Gating}=\begin{bmatrix} \frac{\exp\left\|y_1-O_1^{Expert1}\right\|^2}{\sum_{i=1}^{k}\exp\left\|y_1-O_1^{Experti}\right\|^2} & \cdots & \frac{\exp\left\|y_1-O_1^{Expertk}\right\|^2}{\sum_{i=1}^{k}\exp\left\|y_1-O_1^{Experti}\right\|^2} \\ \vdots & \ddots & \vdots \\ \frac{\exp\left\|y_{N_{train}}-O_{N_{train}}^{Expert1}\right\|^2}{\sum_{i=1}^{k}\exp\left\|y_{N_{train}}-O_{N_{train}}^{Experti}\right\|^2} & \cdots & \frac{\exp\left\|y_{N_{train}}-O_{N_{train}}^{Expertk}\right\|^2}{\sum_{i=1}^{k}\exp\left\|y_{N_{train}}-O_{N_{train}}^{Experti}\right\|^2} \end{bmatrix} \quad (15)$$

Then weights between hidden layer and output layer of gating network are calculated according to (16).

$$\beta = H^{\dagger}.Y_{Gating} \quad (16)$$

Also to combine outputs of experts, gating network applies a trainable data dependent strategy according to (17), in which g_i is the weight corresponding to i^{th} expert for sample x .

$$g_i=\frac{\exp(O_{g_i})}{\sum_{j=1}^{k}(O_{g_j})} \quad (17)$$

where O_{g_i} is the i^{th} output of the gating network and k is the number of experts. g_i can be interpreted as estimation of selecting the output of i^{th} expert by the gating network. According to (17), the softmax function is applied as the gating network which satiates $g_i \geq 0$ and $\sum_i g_i = 1$.

The final output of MEETG, where each gate is multiplied by the output of acorresponding expert, and all there are aggregated, in order to produce the final output is calculated as follows:

$$O_{ens}=\sum_{i=1}^{k} O_i.g_i \quad (18)$$

where O_i is the output of i^{th} expert. Final decision-making, the maximum possibility is considered as follows:

$$C = argmax_{j=1}^{m} O_{ens} \quad (19)$$

where $O_{ens}=[O_1,O_2,\cdots,O_m]$ is the output vector of MEETG corresponding to m classes for each test sample. The details of MEETG algorithm are given in [15,16]. The structure of MEETG is shown in Figure 4.

3. Proposed Bark Texture Classification Approach

Feature extraction and classification are the two main factors that must be considered in designing a texture classification system [1, 25, 29, 30]. In the feature extraction phase, texture properties are extracted using texture descriptors. In the second phase, a classifier such as a neural network, mixture of experts, or other classifiers is trained to assign an unknown sample to a predefined texture class based on its texture properties. In this section, we describe the proposed method for the feature extraction phase and classification.

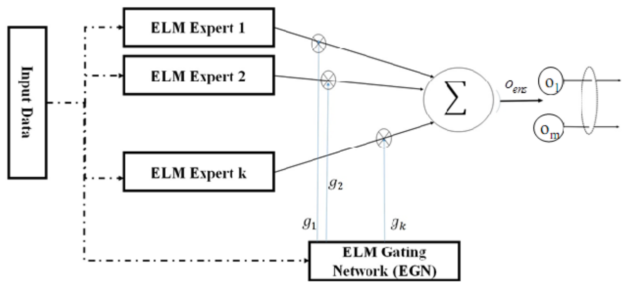


Figure 4. Block diagram of MEETG [16].

3.1. Feature extraction phase (color improved local quinary patterns)

When extracting features from tree bark images for the purpose of identifying tree species, color information is an important characteristic to consider. The color and texture of the bark can serve as excellent features for accurate identification of tree species. As the dataset used in this study consists of color images, it is necessary to incorporate color features into the algorithm. However, the ILQP descriptor used in this work has been originally designed for gray scale images [25], presenting a challenge for combining it with color sensor information. To address this issue, we propose an approach for color-texture classification using our modified version of ILQP. To preserve color information in this color dataset, the descriptor has been applied to all three image bands, and the sum and concatenation methods are used to combine this information into a vector.

In this article, we applied a method of using color images for color-texture classification. RGB colors are called primary colors and are additive. A color image is a combination of some basic colors. In other words, each image breaks into 3 matrices down into red, green, and blue values then each one representing color features.

At the beginning of the work, to extract color-texture features, we separated each color image into three different color channels, red, green, and blue. In continuation, each of these three channels is considered as a gray scale image, and then by applying the ILQP operator, texture features are extracted from red, green, and blue channels. We use two techniques to combine the color images. In the first technique according to (20), we concatenate the features extracted from each channel and we consider them as a feature vector for color texture as follows:

$$ILQP_{P,R}^{Concat} = < F_R, F_G, F_B > \quad (21)$$

where F_R shows the extracted feature vector for red color channel using (21). Also F_G and F_B can be defined in a similar way. Finally, $ILQP_{P,R}^{Concat}$ is a vector with $12 \times (P+2)$ dimension. In second technique, the summation of extracted color features is considered to provide a main feature vector according to (22).

$$ILQP_{P,R}^{Sum} = < F_R + F_G + F_B > \quad (22)$$

Also the dimension of the final feature vector is $4 \times (P+2)$. Thus by applying the proposed feature extraction method, two feature vectors are provided for each input image. The block diagram of the proposed feature extraction method is shown in the Figure 5.

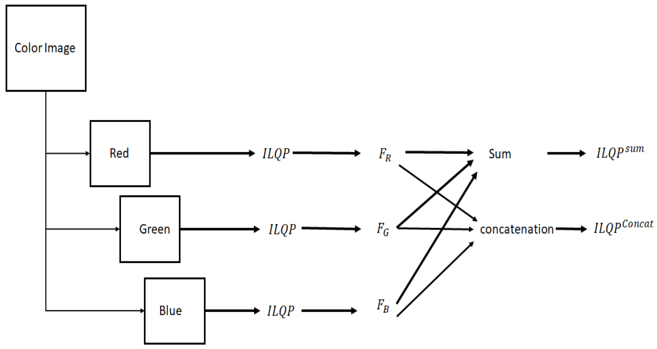


Figure 5. Block diagram of the proposed feature extraction method.

3.2. Classification phase (stacked mixture of ELM-based experts with trainable gating network)

In this sub-section, we present a new classification method named as stacked mixture of ELM based experts with trainable gating network (stacked MEETG). Our proposed method is inspired by the methods presented in [31] and [32]. Figure 6 shows the structure of stacked MEETG.

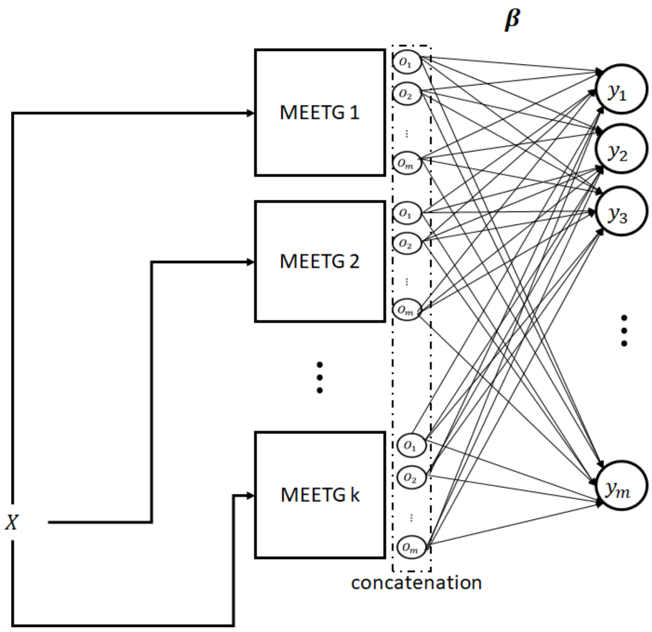


Figure 6. Block diagram of stacked MEETG.

The stacking is a generic framework which it consists of two levels of learning, base learning, and meta-learning. In the first level, the base experts are trained on the original data set. Then, a meta-classifier on top of previous models combines their outputs. In other words, the output of first-level experts is regarded as new features, and the original class labels are kept as the labels in the new data set.

The main difference between stacked and other methods of ensemble techniques is that in stacking, meta-level classification is applied as final classification. Our proposed method consists of training multiple MEETGs on different feature spaces, then in the meta-learning, training the weights directs how much each MEETG's output contributes to the final prediction.

The main difference between stacked and other methods of ensemble techniques is that in stacking, meta-level classification is applied as final classification. Our proposed method consists of training multiple MEETGs on different feature spaces, then in the meta-learning, training the

weights directs how much each MEETG's output contributes to the final prediction.

The training mechanism of MEETG is given in Algorithm 1. In this paper, we take the advantages of ELM as the meta-classier. The output of base experts is considered as the output of the hidden layer neurons of ELM and the Moore-Penrose pseudo inverse method is applied for adjusting the output weights of ELM. In other words, for a system with k MEETGs and m classes, the number of weights of ELM in this method is $c \times m$, where c is the concatenation outputs of MEETGs. In this method, we employ the output layer of ELM in the meta-level as the decision layer. If A is the output matrix of MEETGs for N training samples which is considered as the output of hidden layer neurons of ELM and β is the weight matrix between hidden layer and output layer of network, then β is calculated according to (25).

$$A = \begin{bmatrix} O_1^{11} & \cdots & O_1^{1m} & \cdots & O_1^{k1} & \cdots & O_1^{km} \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ O_N^{11} & \cdots & O_N^{1m} & \cdots & O_N^{k1} & \cdots & O_N^{km} \end{bmatrix}_{N \times c} \tag{23}$$

$$\beta = \begin{bmatrix} \beta_1 \\ \vdots \\ \beta_c \end{bmatrix}_{c \times m} \tag{24}$$

$$\beta = A^{\dagger}.Y \tag{25}$$

where $A^{\dagger}$ is the Moore-Penrose generalized inverse of matrix A. Finally, the result of the ensemble learning model is obtained by multiplying the calculated output weights in the output matrix of each base MEETG. In other words, $Y_{ens} = [y_1, y_2, \ldots, y_m]$ is the output vector of stacked MEETG corresponding to each test sample $x \in R^n$ and m classes which y_i is calculated as follows:

$$y_i = \sum_{i=1}^{K} \sum_{j=1}^{m} O^{i,j} \beta_{ij} \tag{26}$$

Therefore the final class label can be determined by the following maximum process.

$$Label = argmax_{j=1}^{m} Y_{ens} \tag{27}$$

As shown in Figure 7., we apply two MEETGs as the base classifiers of the ensemble learning model for tree bark classification. In order to increase diversity of the ensemble learning model, in addition to random input weights, different features are given to each of the classifiers.

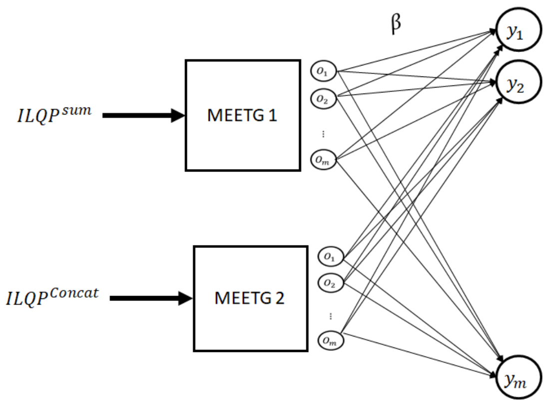


Figure 7. Block diagram of tree bark classification using $ILQP^{Concat}$, $ILQP^{Sum}$ and stacked MEETG.

In this paper, first MEETG receives $ILQP_{P,R}^{Sum}$ as a feature vector obtained according to (21) and second MEETG receives $ILQP_{P,R}^{Concat}$ as a feature vector obtained according to (22).

4. Simulation results

To evaluate the effectiveness of the proposed tree bark classification approach, we carry out experiments on three datasets called Trunk12 [33], BarkTex [34] and AFF [5]. The brief description of the datasets is expressed in the next subsection. Computer specifications to evaluate the performance of our proposed approach are shown in Table 1.

Table 1. Computer specifications used in the experiments.

Computer properties	Specification
CPU	Intel Core(TM) i5-6200U
Core	5 cores (2.40 GHz)
RAM	12 GB
Simulation	MATLAB 2018a
Operating system	64 bit

4.1. Datasets

In this subsection, the applied texture datasets are briefly described.

- Trunk12 dataset

The Trunk12 dataset contains 393 images of tree barks belonging to 12 different trees that are found in Slovenia in .JPG image format, with a resolution of 3000×4000 pixels. The number of images per class varies between 30 and 45 images. Bark images are captured under controlled scale, illumination and pose conditions. Some examples of this dataset are shown in Figure 8.

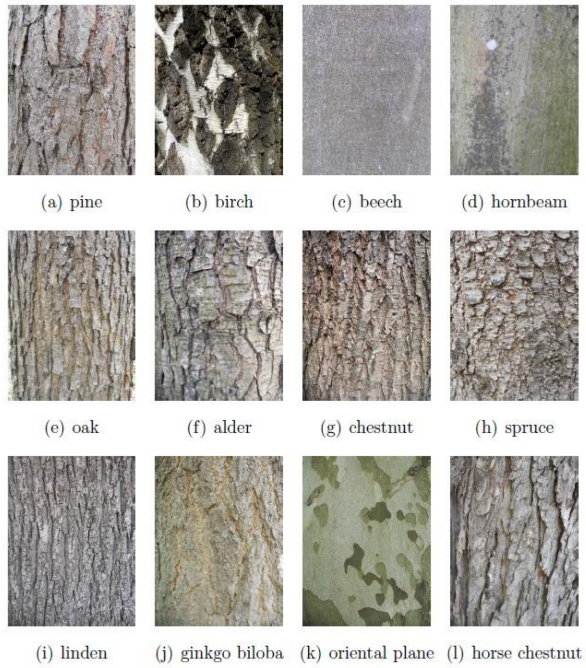


Figure 8. Some examples of Trunk12 dataset.

- BarkTex dataset

This dataset contains a collection of 408 color textures for the computer vision community. The pictures show the bark of six different European trees (betula pendula, fagus silvatica, picea abies, pinus silvestris, quercus robur, and robinia pseudacacia). The collection contains 68 images corresponding to each class. All pictures were taken from different trees under natural lighting conditions. This image collection was acquired for classification experiments in color texture analysis. All images in the BarkTex dataset are stored as raw ppm (P6) files. The images have small (128 × 192) resolution, and they have unequal natural illumination and scale. Some examples of this dataset are shown in Figure 9.

Figure 9. Images of the BarkTex dataset, one example for each class (left-right):(Birch, Beech, Spruce, Pine, Oak, Robinia).

- AFF dataset

The AFF bark dataset provided by österreichische Bundesforste, Austrian Federal Forests (AFF), is a collection of the most common Austrian trees [5]. The dataset contains 1182 bark samples 1000×(478-1812) belonging to 11 classes. The size of each class varying between 16 and 213 images. AFF samples are captured at different scales, and under different illumination conditions [5]. Some examples of this dataset are shown in Figure 10.

Figure 10. Images of the AFF dataset, one example for each class (left-right and top-bottom): (Ash, Beech, Black pine, Fir, Hornbeam, Larch, Mountain oak, Scots pine, Spruce, Swiss stone pine, and Sycamore maple).

4.2. Experimental results

In this paper, an ensemble learning method based on extreme learning machine is proposed for tree bark classification in which MEETG is used as the base classifier. To demonstrate performance of the proposed method, experimental results of our method are reported on three datasets consisting of Trunk12 [33], BarkTex [34], and AFF [5]. In this section, the performance of MEETG and stacked MEETG are compared on different feature vectors. The results of experiments are presented with different experts in Table 2, Table 3 and Table 4. As it is shown in the Tables, in the experiments, different radiuses (R) and neighbors (P) are employed for $ILQP_{P,R}$.

Table 2. Classification accuracy (%) of stacked MEETG and MEETG on Trunk12 dataset with 10-fold cross-validation.

Method	3 experts	5 experts	7 experts
$ILQP_{8,1}^{concat} + MEETG$	81.63	83.60	85.83
$ILQP_{8,1}^{sum} + MEETG$	81.84	83.89	84.59
$\boldsymbol{ILQP_{8,1}^{sum} + ILQP_{8,1}^{concat} + stackedMEETG}$	**84.89**	**87.35**	**88.32**
$ILQP_{16,2}^{concat} + MEETG$	82.25	83.77	84.79
$ILQP_{16,2}^{sum} + MEETG$	77.27	78.04	79.08
$\boldsymbol{ILQP_{16,2}^{sum} + ILQP_{16,2}^{concat} + stackedMEETG}$	**83.06**	**84.27**	**85.27**
$(ILQP_{8,1} + ILQP_{16,2})^{concat} + MEETG$	89.80	91.77	91.05
$(ILQP_{8,1} + ILQP_{16,2})^{sum} + MEETG$	86.83	87.84	88.08
$\boldsymbol{(ILQP_{8,1} + ILQP_{16,2})^{sum} + (ILQP_{8,1} + ILQP_{16,2})^{concat} + stackedMEETG}$	**91.50**	**91.77**	**92.79**

Table 3. Classification accuracy (%) of stacked MEETG and MEETG on BarkTex dataset with 10-fold cross-validation.

Method	3 experts	5 experts	7 experts
$ILQP_{8,1}^{concat} + MEETG$	88.02	87.56	87.07
$ILQP_{8,1}^{sum} + MEETG$	88.9	87.47	87.72
$\boldsymbol{ILQP_{8,1}^{sum} + ILQP_{8,1}^{concat} stackedMEETG}$	**89.91**	**90.14**	**90.13**
$ILQP_{16,2}^{concat} + MEETG$	88.71	87.5	87.75
$ILQP_{16,2}^{sum} + MEETG$	87.54	87.57	88.29
$\boldsymbol{ILQP_{16,2}^{sum} + ILQP_{16,2}^{concat} + stackedMEETG}$	**90.17**	**90.16**	**90.63**
$(ILQP_{8,1} + ILQP_{16,2})^{concat} + MEETG$	92.06	91.34	91.84
$(ILQP_{8,1} + ILQP_{16,2})^{sum} + MEETG$	91.09	90.84	91.36
$\boldsymbol{(ILQP_{8,1} + ILQP_{16,2})^{sum} + (ILQP_{8,1} + ILQP_{16,2})^{concat} + stackedMEETG}$	**92.07**	**91.10**	**92.54**

Table 4. Classification accuracy (%) of stacked MEETG and MEETG on AFF dataset with 10-fold cross-validation.

Method	3 experts	5 experts	7 experts
$ILQP_{8,1}^{concat} + MEETG$	86.30	86.05	86.98
$ILQP_{8,1}^{sum} + MEETG$	83.80	84.21	84.81
$\boldsymbol{ILQP_{8,1}^{sum} + ILQP_{8,1}^{concat} stackedMEETG}$	**89.32**	**88.95**	**89.82**
$ILQP_{16,2}^{concat} + MEETG$	84.04	83.20	84.29
$ILQP_{16,2}^{sum} + MEETG$	80.61	80.69	80.69
$\boldsymbol{ILQP_{16,2}^{sum} + ILQP_{16,2}^{concat} + stackedMEETG}$	**89.15**	**89.16**	**89.15**
$(ILQP_{8,1} + ILQP_{16,2})^{concat} + MEETG$	90.84	91.35	91.09
$(ILQP_{8,1} + ILQP_{16,2})^{sum} + MEETG$	87.33	86.99	87.50
$\boldsymbol{(ILQP_{8,1} + ILQP_{16,2})^{sum} + (ILQP_{8,1} + ILQP_{16,2})^{concat} + stackedMEETG}$	**91.35**	**91.68**	**91.35**

Based on the approach used in Figure 7 (Stacked MEETG method), two MMETG classifiers are utilized, and thus two feature vectors are needed. For example, in Table 2, $\mathrm{ILQP}_{16,2}^{\mathrm{sum}} + \mathrm{ILQP}_{16,2}^{\mathrm{concat}} + \mathrm{stackedMEETG}$ means that the stacked MEETG method is used as a classifier, with $\mathrm{ILQP}_{16,2}^{\mathrm{sum}}$ used as the first feature vector for the first classifier, and $\mathrm{ILQP}_{16,2}^{\mathrm{concat}}$ used as the second feature vector for the second classifier. Similarly, $(\mathrm{ILQP}_{8,1} + \mathrm{ILQP}_{16,2})^{\mathrm{sum}} + (\mathrm{ILQP}_{8,1} + \mathrm{ILQP}_{16,2})^{\mathrm{concat}} + \mathrm{stackedMEETG}$ means that the first feature vector merges $\mathrm{ILQP}_{8,1}^{\mathrm{sum}}$ and $\mathrm{ILQP}_{16,2}^{\mathrm{sum}}$ for MEETG1, while the second feature vector merges $\mathrm{ILQP}_{16,2}^{\mathrm{concat}}$ and $\mathrm{ILQP}_{16,2}^{\mathrm{concat}}$ for MEETG2. Furthermore, $(\mathrm{ILQP}_{8,1} + \mathrm{ILQP}_{16,2})^{\mathrm{sum}} + \mathrm{MEETG}$ in the table implies that $\mathrm{ILQP}_{8,1}^{\mathrm{sum}}$ and $\mathrm{ILQP}_{16,2}^{\mathrm{sum}}$need to be merged to form a feature vector for the classifier, using the MEETG method, which requires a feature vector.

As the result illustrated in Table 2, the best accuracy on the Trunk12 is achived with considering 7 experts for both MEETG and stacked MEETG compared with 5 and 3 experts. Also the results show that stacked MEETG performs better than MEETG corresponding to each feature. According to Table 3, the maximum accuracy on the BarkTex dataset is 92.54% by using stacked MEETG and combination of $ILQP_{8,1}$ and $ILQP_{16,2}$ with 7 experts. Also according to the achieve results for AFF dataset that are shown in Table 4, the best accuracy with $ILQP_{8,1}$ is 89.82%, with $ILQP_{16,2}$ is 89.15% and with combination of $ILQP_{8,1}$ and $ILQP_{16,2}$ is 91.68%. Overall the results indicate that stacked MEETG performs better than MEETG and also combination of $ILQP_{8,1}$ and $ILQP_{16,2}$ provides the maximum accuracy.

Table 5. Comparison between the results (%) of the proposed method and different bark classification methods.

Approach	Trunk12	BarkTex	AFF
$SRBP_{CT} + LBP_{8,1}^{riu2}$ [7]	62.84	84.55	60.49
sum and difference histograms [35]	-	87	-
GLBP [8]	78.39	**94.39**	72.21
Concatenating LBP and Gradient [8]	73.45	91.33	71.33
$1DLBP_{8,1} + 1DLBP_{16,2}$ [36]	77.32	-	70.96
Color Histogram (H=30) [9]	64.43	55.4	50.51
Color Histogram (H=80) [9]	69.00	61.3	55.62
LCoLBP + Color Histogram (H=80) [9]	84.2	91.7	80.7
LCoLBP + Color Histogram (H=30) [9]	84.2	92.4	80.7
GWs + Color Histogram(H=30) [37]	74.3	66.2	64.7
GWs + Color Histogram (H=80) [37]	76.10	69.6	66.5
$sSRBP(sch\#1) + 1 - NN$[38]	77.86	-	78.25
$SRBP(sch\#2) + 1 - NN$[38]	81.17	-	77.83
$ILTP_{8,1} + ILTP_{16,2} + MLP$ [1]	86.76	-	82.93
$\boldsymbol{(ILQP_{8,1} + ILQP_{16,2})^{sum} + (ILQP_{8,1} + ILQP_{16,2})^{concat} + stackedMEETG}$	**92.79**	92.54	**91.68**

4.3. Comparison between proposed method and different bark classification methods

The main objective of this paper is to provide a bark classification approach with high performance. Table 5 shows a comparison between the performance of the proposed method with some other methods in this area.

As seen in Table 5, our approach vastly outperforms all compared methods on the Trunk12 and AFF datasets and has the second best result on the BarkTex dataset.

According to Table 6, stacked MEETG improves the performance about 6% for Trunk12 dataset. As seen in this tables the accuracy of ILQP+ stacked MEETG has increased bout 12% with $P = 8, R = 1$ compared with ILTP + MLP and also with a $P = 16, R = 2$ increased about 10%.

Table 6. Comparison of the proposed method with method [1] on Trunk12 dataset with 10-fold.

Proposed method	Accuracy (%)	[1]	Accuracy (%)
$ILQP_{8,1}^{sum} + ILQP_{8,1}^{concat} + stackedMEETG$	**88.32**	$ILTP_{8,1} + MLP$	75.82
$ILQP_{16,2}^{sum} + ILQP_{16,2}^{concat} + stackedMEETG$	**85.27**	$ILTP_{16,2} + MLP$	81.17
$(ILQP_{8,1} + ILQP_{16,2})sum + (ILQP_{8,1} + ILQP_{16,2})^{concat} + stackedMEETG$	**92.79**	$ILTP_{8,1} + ILTP_{16,2} + MLP$	86.72

4.4. Tree bark classification in the presence of noise

To evaluate the performance in presence of noise, we first apply two type of impulse noises (salt & pepper and speckle) on texture images and extract the texture features using the proposed method. The results are shown in Table 7.

In Table 7, speckle noise level with the variance of 0.02 and also salt & pepper noise with a density ratio of 5%, 10%, 20%, and 30% are applied to tree bark images. Comparison between the proposed method and the other efficient methods in this domain is carried out with 10-fold cross-validation in the presence of a variety of noise on Trunk12 dataset.

Table 7. Comparison classification accuracy (%) of the proposed method on Trunk12 dataset.

Method	Without Noise	Salt and pepper				Speckle
		5%	10%	20%	30%	0.02
$1DLBP_{8,1} + 1DLBP_{16,2}$ [36]	77.32	72.26	69.41	61.64	58.43	64.96
$ILTP_{8,1} + ILTP_{16,2} + MLP$[1]	86.76	82.49	80.4	72.26	68.44	73.53
$(\boldsymbol{ILQP_{8,1} + ILQP_{16,2}})^{sum} + (\boldsymbol{ILQP_{8,1} + ILQP_{16,2}})^{concat} + \boldsymbol{stackedMEETG}$	**92.79**	**91.59**	**90.06**	**87.85**	**85.84**	**89.07**

4.5. Evaluation of stacked MEETG

Confusion matrix is an $N \times N$ matrix applied to evaluate the performance of a classification model, where N is the number of target classes.

In this matrix, the actual level of data for each class is displayed in the rows and columns, and the number of samples that have been assigned correctly or incorrectly to each class is entered in the corresponding cells. Generally, in a confusion matrix, the diagonal elements represent the number of samples that have been correctly assigned to their own class. In other words, these elements show how many samples has been correctly classified by the algorithm or model. The off-diagonal elements of the confusion matrix represent the number of samples that have been incorrectly assigned to each class. In other words, these elements show how many samples the algorithm or model has incorrectly classified as other classes. Based on the confusion matrix, various metrics can be calculated to evaluate the performance of an algorithm or model in multi-class problems.

Accuracy: The ratio of the number of samples that have been correctly classified to the total number of samples. To calculate accuracy, the sum of the diagonal elements of the confusion matrix is divided by the total number of samples.

Precision: The ratio of the number of samples that belong to class i and have been correctly classified to the number of samples that the algorithm has classified as class i. To calculate precision for each class, the number of diagonal elements corresponding to that class is divided by the number of elements in the same column of that class.

Recall: The ratio of thenumber of samples that belong to class i and have been correctly classified to the total number of samples that actually belong to class i. To calculate recall for each class, the number of diagonal elements corresponding to that class is divided by the number of elements in the same row of that class.

The confusion matrix of Trunk12 dataset, BarkTex dataset, and AFF dataset is shown in Figure 11, Figure 12, and Figure 13.

For example, Figure 12 shows the confusion matrix of the BarkTex dataset. As it can be seen, the Beech class has been correctly classified, and no samples have been incorrectly assigned to this class. Additionally, out of 68 samples in the Spruce class, 3 samples have been misclassified to other classes, and 6 samples from other classes have been incorrectly classified as Spruce.

In addition to overall classification accuracy, some other criteria can be measured to evaluate the performance of the proposed method such as precision and recall. Precision and Recall are useful measures of success of prediction when the classes are very imbalanced which is calculated according to (27) and (28).

$$Precision = \frac{TP}{TP+FP} \tag{28}$$

$$Recall = \frac{TP}{TP+FN} \tag{29}$$

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \tag{29}$$

where TP is number of true positives, FP number of false positives, and FN is number of false negatives.

Using these two metrics (Precision and Recall), the performance of an algorithm can be evaluated more completely and accurately. For example, an algorithm that has a high precision but low recall means that the algorithm correctly identifies most of the cases that are identified as positive, but the number of cases that are not identified as positive is very high. In other words, this algorithm finds positive cases that are truly positive, but does not pay attention to many positive cases that are actually positive. Therefore, to make the algorithm work well and provide accurate results, both of these metrics should be examined simultaneously.

For example, in cases where both precision and recall are high, it means that the algorithm has correctly identified positive cases and has also found many positive cases. As a result, precision and recall should be examined together to fully evaluate the performance of the algorithm. The performance of stacked MEETG with$(ILQP_{8,1} + ILQP_{16,2})^{sum} + (ILQP_{8,1} + ILQP_{16,2})^{concat}$ based on these metrics are presented in Table 8.

Table 8. Precision, recall, and accuracy (%) metrics for the stacked MEETG with 10-fold cross-validation.

Dataset	Precision	Recall	Accuracy
Trunk12	91.62	92.34	92.79
BarkTex	92.66	92.72	92.54
AFF	89.39	92.68	91.68

Predicted classes

Actual classes	Al	Be	Bi	Ch	Gi	Ho	Horse	Li	Oa	Or	Pi	Sp
Alder	34	0	0	0	0	0	0	0	0	0	0	0
Beech	0	28	0	0	1	1	0	0	0	0	0	0
Birch	2	0	33	0	0	0	0	1	0	0	0	1
Chestnut	2	0	0	24	0	0	1	0	2	0	0	3
Ginkgo biloba	0	0	0	0	30	0	0	0	0	0	0	0
Hornbeam	0	1	0	0	0	28	0	0	1	0	0	0
Horse chestnut	0	0	0	0	0	0	32	0	0	0	0	1
Linden	3	0	0	0	0	0	2	22	2	1	0	0
Oak	0	0	0	0	1	0	0	0	29	0	0	0
Oriental plane	0	0	0	0	0	0	1	0	0	29	1	1
Pine	0	0	0	0	0	0	0	0	0	0	30	0
Spruce	0	0	0	0	0	0	0	0	0	0	0	45

Figure 11. Confusion matrix of Trunk12 dataset.

Predicted classes

Actual classes	Bi	Be	Sp	Pi	Oa	Ro
Birch	67	0	1	0	0	0
Beech	0	68	0	0	0	0
Spruce	1	0	65	0	2	0
Pine	0	0	0	62	4	2
Oak	0	0	4	1	56	7
Robinia	0	0	1	1	6	60

Figure 12. Confusion matrix of the BarkTex dataset.

Predicted classes

	Ash	Beech	BP	Fir	HO	Larch	MO	SP	Spruce	SSP	SM
Ash	24	0	1	0	0	0	2	2	1	3	0
Beech	0	15	0	1	0	0	0	0	0	0	0
Black pine	0	0	153	1	0	3	0	8	0	1	0
Fir	0	0	0	124	0	2	0	1	0	0	0
Hornbeam	0	0	0	0	41	0	0	1	0	0	0
Larch	0	0	7	1	0	182	0	9	0	1	0
Mountain oak	2	0	0	0	0	1	68	0	1	5	0
Scots pine	0	0	6	0	0	13	0	171	0	0	0
Spruce	0	0	0	3	0	2	0	1	206	1	0
viss stone pine	1	0	1	0	0	3	1	2	1	86	1
camore maple	0	0	2	0	0	1	0	1	1	2	15

Figure 13. Confusion matrix of the AFF dataset.

Table 9. TP-rate and FP-rate for stacks MEETG, and the method proposed in [1] on Trunk12 dataset with 10-fold cross-validation.

Class	TP-rate (Proposed)	TP-rate [1]	FP-rate (Proposed)
alder	**1.0**	0.882	**0.01**
beech	**0.933**	0.900	**0.002**
birch	**0.891**	0.973	**0.000**
chestnut	**0.75**	0.625	**0.000**
ginkgo biloba	**1.0**	0.933	**0.005**
hornbeam	**0.933**	0.900	**0.002**
horse chestnut	**0.969**	0.909	**0.005**
linden	**0.73**	0.667	**0.002**
oak	**0.966**	0.767	**0.013**
oriental plane	**0.906**	0.938	**0.002**
pine	**1.0**	0.967	**0.002**
spruce	**1.0**	0.911	**0.017**

4.5. Diversity measures of stacked MEETG

Diversity among the members of a team of classifiers is deemed to be a key issue in classifier combination. Several measures have been defined for quantitative assessment of diversity. The simplest ones are pair-wise measures, defined between two classifiers [39, 40].

	$D_j(correc)$	$D_j(incorrect)$
$D_i(correc)$	a	b
$D_i(incorrect)$	c	d

where a is the fraction of instances that are correctly classified by both classifiers, b is the fraction of instances correctly classified by D_i but incorrectly classified by D_j, and so on. Then the following pairwise diversity measures can be defined:

$$Q = \frac{(ad - bc)}{(ad + bc)} \tag{30}$$

$$\rho = \frac{(ad - bc)}{\sqrt{(a+b)(c+d)(a+c)(b+d)}}, \quad 0 \le \rho \le 1 \tag{31}$$

- The Q statistic:

Diversity is measured as the Q statistics according to (30). Q assumes positive values if the same instances are correctly classified by both classifiers, and negative values, otherwise, maximum diversity is obtained for $Q = 0$.

- The correlation coefficient ρ:

Diversity is measured as the correlation between two classifier outputs, defined according to (31). Maximum diversity is obtained for $\rho = 0$, indicating that the classifiers are uncorrelated [39].

Table 10. Then measures of diversity Q statistics and correlation stacked MEETG.

Dataset	Q-statistics	Correlation (ρ)
Trunk12	0.79	0.29
BarkTex	0.94	0.53
AFF	0.84	0.36

5. Conclusion

Tree species classification and identification through tree bark is a topic of interest among the researchers. This paper presents a novel ensemble learning method for bark classification based on textural and color features. Two techniques are applied to extract features from color images. As the experimental results are shown, combination of $ILQP_{8,1}$ and $ILQP_{16,2}$ provides maximum accuracy.

For the classification of tree bark images, the choice of classifier is also an important factor. Ensemble learning methods, in which the output results of multiple classifiers are combined, are a good approach to improving the performance of classification. To the best of our knowledge, in most previous works, tree bark classification is typically performed using a single classifier. Therefore, in this paper, we propose a hybrid ensemble system called stacked MEETG. Stacking involves applying a learning algorithm to combine the predictions of several other learning algorithms. The general framework of stacked MEETG consists of two levels. In the first level, base classifiers or base experts are trained. We apply MEETG as base experts for designing the structure of stacked MEETG.

In the second level, meta-learning, a meta-classifier is trained based on the outputs of the first-level classifiers to learn how to aggregate the predictions of base experts. In this paper, we take advantage of ELM as the meta-classifier. The output of base experts is considered as the output of the hidden layer neurons of ELM, and the Moore-Penrose pseudo-inverse method is applied for adjusting the output weights of ELM. The trained weights of ELM determine how much each expert contributes to the final classification.

We compared our proposed method with the recent state-of-the-art method the context of tree bark classification. Experimental results proposal method performs better than the state-of-the-art method in this area. We believe that $ILQP_{P,R}^{concat}$ and $ILQP_{P,R}^{sum}$ can be generalized on other applications for feature extraction and also stacked MEETG can be applied in many complex classification problems.

References

[1] Sh. Fekri-Ershad, “Bark texture classification using improved local ternary patterns and multilayer neural network,” *Expert Systems with Applications*, vol. 158, pp.113509, 2020.

[2] M. Wojtech, Bark A Field Guide to Trees of the Northeast,” *University Press of New England*, 2011.

[3] Z.K. Huang, C.H. Zheng, J.X. Du, and Y.y. Wan, “Bark classification based on textural features using artificial neural networks,” In: *International Symposium on Neural Networks*, pp. 355–360. 2006.

[4] S. Boudra, I. Yahiaoui, and A. Behloul, “A comparison of multi-scale local binary pattern variants for bark image retrieval,” *In: International Conference on Advanced Concepts for Intelligent Vision Systems*, pp. 764–775, 2015.

[5] S. Fiel, and R. Sablatnig, “Automated identification of tree species from images of the bark, leaves or needles,” *na*, pp. 67-74, 2010.

[6] A. Bressane, J.A. Roveda, and A.C. Martins, “Statistical analysis of texture in trunk images for biometric identification of tree species,” *Environmental monitoring and assessment*, vol.187, no. 4, pp. 1-9, 2015.

[7] S. Boudra, I. Yahiaoui, and A. Behloul, “Statistical radial binary patterns (srbp) for bark texture identification,” *In: International conference on advanced concepts for intelligent vision systems*, pp. 101–113, 2017.

[8] T. Le-Viet, V. Truong Hoang, “Local binary pattern based on image gradient for bark image classification,” *In: Tenth International Conference on Signal Processing Systems*, vol. 11071, p. 110710P ,2019.

[9] R. Ratajczak, S. Bertrand, C. Crispim-Junior, and L. Tougne, “Efficient bark recognition in the wild,” In: International conference on computer vision theory and applications (VISAPP 2019), 2019.

[10] D. Misra, C. Crispim-Junior, and L. Tougne, “Patch-based cnn evaluation for bark classification,” *In: European Conference on Computer Vision*, pp. 197–212, 2020.

[11] F. Alimoğlu, and E. Alpaydin, “Combining multiple representations for pen-based handwritten digit recognition,” *Turkish Journal of Electrical Engineering & Computer Sciences*, vol. 9, no. 1, pp. 1–12, 2001.

[12] R. Ebrahimpour, S.A.A.A. Arani, and S. Masoudnia, “Improving combination method of ncl experts using gating network,” *Neural Computing and Applications*, vol. 22, no. 1, pp. 95–101, 2013.

[13] M. Pietikäinen, T. Ojala, and Z. Xu, “Rotation-invariant texture classification using feature distributions,” *Pattern recognition*, vol. 33, no. 1, pp. 43–52, 2000.

[14] L.Y. Yang, Z. Qin, and R. Huang, “Design of a multiple classifier system,” *In: Proceedings of 2004 International Conference on Machine Learning and Cybernetics* (IEEE Cat. no. 04EX826), vol. 5, pp. 3272–3276, 2004.

[15] L. Armi, E. Abbasi, and J. Zarepour-Ahmadabadi, “Mixture of experts based on ELM with trainable gating network,” *arXiv preprint arXiv:2105.11706*, 2021.

[16] L. Armi, E. Abbasi, and J. Zarepour-Ahmadabadi “Texture images classification using improved local quinary pattern and mixture of ELM based experts,” *Neural Comput & Applic*, pp.1–24, 2021.

[17] T. Ojala, and M. Pietikäinen, “Harwood, D.: A comparative study of texture measures with classification based on featured distributions,” *Pattern recognition*, vol. 29, no.1, pp. 51–59, 1996.

[18] M. Pietikäinen, T. Ojala, and Z. Xu, “Rotation-invariant texture classification using feature distributions,” *Pattern recognition*, vol. 33, no. 1, pp, 43–52, 2000.

[19] T. Ojala, M. Pietikainen, and T. Maenpaa, “Multi-resolution gray-scale and rotation invariant texture classification with local binary patterns,” *IEEE Transactions on pattern analysis and machine intelligence*, vol. 24, no. 7, pp. 971–987, 2002.

[20] Z. Guo, L. Zhang, and D. Zhang, "A completed modeling of local binary pattern operator for texture classification," *IEEE transactions on image processing*, vol. 19, no. 6, pp. 1657- 1663, 2010.

[21] X. Tan, and B. Triggs, "Enhanced local texture feature sets for face recognition under difficult lighting conditions," *International Workshop on Analysis and Modeling of Faces and Gestures*, pp. 168-182, 2010.

[22] J.H. Yuan, H.D. Zhu, Y. Gan, and L. Shang, "Enhanced Local Ternary Pattern for Texture Classification," *International Conference on Intelligent Computing*, pp. 443-448, 2014.

[23] Y. Zhao, D.-S. Huang, and W. Jia, “Completed Local Binary Count for Rotation Invariant Texture,” *IEEE Trans. on Image Process*, vol. 21, no, 10, pp. 4492–4497, 2012.

[24] L. Nanni, A. Lumini, and S. Brahnam, “Local binary patterns variants as texture descriptors for medical image analysis,” *Artificial intelligence in medicine*, vol. 49, no. 2, pp. 117-125, 2010.

[25] L. Armi, and S. Fekri-Ershad, “Texture image classification based on improved local quinary patterns,” *Multimedia Tools and Applications*, vol. 78, no. 14, pp. 18995–19018, 2019.

[26] R.A. Jacobs, M.I. Jordan, S.J. Nowlan, and G.E. “Hinton, Adaptive mixtures of local experts,” *Neural computation*, vol. 3, no. 1, pp. 79–87, 1991.

[27] G.B. Huang, Q.Y. Zhu, C.K. Siew, “Extreme learning machine: a new learning scheme of feedforward neural networks,” *In: 2004 IEEE international joint conference on neural networks* (IEEE Cat. No. 04CH37541), vol. 2, pp. 985–990, 2004.

[28] G.B. Huang, Q.Y. Zhu, C.K. Siew, “Extreme learning machine: theory and applications,” *Neurocomputing*, vol. 70, no. 1-3, pp. 489–50,1 2006.

[29] L. Armi, and S. Fekri-Ershad, “Texture image analysis and texture classification methods - A review,” *International Online Journal of Image Processing and Pattern Recognition*, vol. 2, no. 1, pp. 1–29, 2019.

[30] Y. Song, S. Zhang, B. He, Q. Sha, Y. Shen, T. Yan, R. Nian, and A. Lendasse, “Gaussian derivative models and ensemble extreme learning machine for texture image classification,” *Neurocomputing*, vol. 277, pp. 53–64, 2018.

[31] Y. Cai, X. Liu, Y. Zhang, and Z. Cai, “Hierarchical ensemble of extreme learning machine,” *Pattern Recognition Letters*, vol. 116, pp. 101–106, 2018.

[32] S. Masar, “Pattern classification based on ensemble of extreme learning machines,” M.S. thesis, Shiraz University, 2018.

[33] M. Švab, “Computer-vision-based tree trunk recognition,” Ph.D. thesis, Bsc Thesis, (Mentor: doc. dr. Matej Kristan), Fakulteta za racunalništvo in informatiko, Univerza v Ljubljani, 2010.

[34] R. Lakmann, “Statistische modellierung von farbtexturen,” *Ph.D. thesis, University of Koblenz-Landau*, Germany, Fölbach, 1998.

[35] C. Münzenmayer, H. Volk, C. Küblbeck, K. Spinnler, and T. Wittenberg, “Multi-spectral texture analysis using interplane sum-and difference-histograms,” *In: Joint Pattern Recognition Symposium*, pp. 42–49, 2002.

[36] Y. Song, S. Zhang, B. He, Q. Sha, Y. Shen, T. Yan, R. Nian, and A. Lendasse, “Gaussian derivative models and ensemble extreme learning machine for texture image classification,” *Neurocomputing*, vol. 277, pp. 53–64, 2018.

[37] S. Bertrand, R.B. Ameur, G. Cerutti, D. Coquin, L. Valet, and L. Tougne, “Bark and leaf fusion systems to improve automatic tree species recognition,” *Ecological Informatics*, vol. 46, pp. 57–73, 2018.

[38] S. Boudra, I. Yahiaoui, and A. Behloul, “A set of statistical radial binary patterns for tree species identification based on bark images,” *Multimedia tools and applications*, (2020).

[39] R. Polikar, “Ensemble based systems in decision making”. *IEEE Circuits and systems magazine*, vol. 6, no. 3, pp. 21–45, 2006.

[40] M. Rahimi, A. A. Taheri, and H. Mashayekhi, “Learning a Nonlinear Combination of Generalized Heterogeneous Classifiers,” *Journal of Artificial Intelligence & Data Mining,* vol. 11, no. 1, pp. 77–93, 2023.

طبقه‌بندی پوسته درخت با استفاده از الگوی پنج‌تایی بهبود یافته با رنگ و Stacked MEETG

لاله آرمی و الهام عباسی*

گروه علوم کامپیوتر، دانشکده علوم ریاضی، دانشگاه یزد، یزد، ایران.

ارسال ۲۰۲۳/۰۲/۰۸؛ بازنگری ۲۰۲۳/۰۴/۲۳؛ پذیرش ۲۰۲۳/۰۷/۰۴

چکیده:

در این مقاله، یک روش دسته‌بندی نوآورانه برای دسته‌بندی پوسته درختان و شناسایی گونه‌های درختی پیشنهاد می‌کنیم. روش پیشنهادی شامل دو مرحله است. در مرحله اول، از مزایای ILQP که یک روش استخراج ویژگی بافت رنگی می‌باشد شامل مقاومت در برابر نویز، توصیف کامل و بی انحراف در قابلیت چرخش استفاده کرده‌ایم. در مرحله دوم، یک روش دسته‌بندی جدید با نام stacked MEETG پیشنهاد کرده‌ایم که شامل ترکیبی از روش اختلاط خبره‌ها مبتنی بر ماشین یادگیر سریع با شبکه میانجی آموزش‌پذیر است. روش پیشنهادی با استفاده از مجموعه داده‌های Trunk12، BarkTex و AFF ارزیابی شده است. عملکرد روش پیشنهادی در این سه مجموعه داده پوسته نشان می‌دهد که رویکرد ما دقت بهتری نسبت به سایر روش‌های برتر دیگر دارد. روش پیشنهادی ما میانگین دقت دسته‌بندی٪ ۹۲,۷۹ (Trunk12)، ٪۹۲,۵۴ (BarkTex) و٪ ۹۱,۶۸ (AFF) را به دست آورده است. علاوه بر این، نتایج نشان می‌دهند که ILQP قابلیت استخراج ویژگی بافت بهتری نسبت به روش‌های مشابه مانند ILTP دارد. به علاوه، stacked MEETG تأثیر بسیار خوبی بر دقت دسته‌بندی دارد.

کلمات کلیدی: الگوی پنج‌تایی محلی بهبود یافته، ماشین یادگیری سریع، یادگیری مرکب، طبقه‌بندی پوست درخت.