Clustering-Based Hierarchical Framework for Multiclass Classification of Leaf Images

Neha Goyal, Kapil Gupta, and Nitin Kumar

***Abstract*—This article introduces a multiclass classification approach accustoming the benefits of partitioning-based strategies and hierarchical techniques. The proposed hierarchical framework creates a hierarchy with the notion of grouping classes with similar traits as one group. It overcomes the deficiency of the existing multiclass extension approaches, viz., nonlinearity, imbalanced class classification, and increasing classification cost with increasing number of classes. The hierarchical framework presents the idea of decomposing several classes hierarchically, where every cluster contains a set of classes having similar traits. The approach aims to maximize the intercluster distance and minimize the intracluster distribution. The effectiveness of the proposed method is evaluated on real-world and complex problems of plant recognition. Three leaf image datasets are considered for performance evaluation using a support vector machine. The results signify that the proposed approach for multiclass classification is an efficient approach with significantly improved recognition accuracy. It is a robust and effective approach with the least computational cost. The speedup factor of the proposed approach in the binary structure is 16, 6.5, and 5.5 as compared to a one-versus-one traditional support vector machine for Flavia, Swedish, and self-collected leaf datasets, respectively.**

***Index Terms*—Dunn index (DI), hierarchical approach, multiclass classification, separability matrix, silhouette value.**

I. Introduction

CLASSIFICATION algorithms are widely practical machine learning algorithms for every aspect of real-world and complex problems. It seeks an approach to classify an object into a set of classes. Several classification algorithms are proposed to classify the test samples into a set of given classes. Consider a set of objects $\mathbb{T}$ represented through a pair, i.e., $(x, y) \in \mathbb{T}$ and l is a set of class labels, where x is a predictor and $y \in l$ is a class label. Among all the classification algorithms, the support vector machine (SVM) had confirmed excellent performance for several classification tasks [1]–[5]. It is primarily proposed for a binary classification task [6]. Later, it is extended for multiple class classification problems due to its computational cost and efficient performance. It is a useful tool for classification and regression tasks. Despite having better generalization ability, it requires a quadratic programming problem solver while solving its standard formulation. For a training dataset of size M, it requires $O(M^3)$. Extending the binary-class SVM to the multiclass SVM is a tricky approach. It disintegrates a large problem into several small binary class problems and then incorporates the solution of all binary classes. Several extension are discussed in the literature that extends binary class SVM to multi class SVM including one versus one rest (OVR) [8], binary-tree SVM (BTS) [9], directed acyclic graph SVM [10], and decision tree SVM [11].

A. Motivation

For a multiclass (k) classification problem, i.e., $k \geq 2$, OVR is the first extension of the binary SVM. It produces k classifiers for the k-class problem by training one classifier for each class, considering the rest of classes as second class, but it suffers from nonlinearity among classes due to highly unbalanced classes. The other approach for multiclass extension is the OVO approach, which produces $k(k-1)/2$ classifiers. It is a speedy and robust classifier during training due to linear separability and balanced classes. However, for classification, integrated speed is less because of a large number of classifiers. Among all the decomposition-based approaches, none of the techniques are efficient and robust simultaneously. Another strategy for multiclass classification is hierarchy based. The process is effective and efficient but consists of several deficiencies, including biased partitioning of classes, scattering samples of one type in more than one cluster, and requiring manual feedback. The multitree hierarchical SVM is presented based on the theory of binary-tree SVM [12] using the clustering method to group k classes into r clusters. The approach is robust and automates the partitioning process based on the similarity between classes. For the r clusters, it again utilizes the OVO strategy that causes higher classification time and increased computational cost.

Integrating the benefits of partitioning-based and hierarchy-based multiclass extensions, this article presents the hierarchical framework to overcome all the issues in the state-of-the-art methods for multiclass classification. The presented framework

Manuscript received September 15, 2021; revised December 14, 2021; accepted January 25, 2022. Date of publication February 25, 2022; date of current version May 20, 2022. Paper 2021-IACC-1177.R1, presented at the 2020 IEEE International Conference on Computing, Power and Communication Technologies, Greater Noida, India, Oct. 2–4, and approved for publication in the IEEE Transactions on Industry Applications by the Industrial Automation and Control Committee of the IEEE Industry Applications Society. The work of Neha Goyal was supported by the University Grant Commission (UGC), India, through the Fellowship under the UGC National Eligibility Test Junior Research Fellowship Scheme under Grant 3320/NET-JULY 2016. *(Corresponding author: Neha Goyal.)*

Neha Goyal and Kapil Gupta are with the National Institute of Technology Kurukshetra, Kurukshetra 136119, India (e-mail: neha.goyal2309@gmail.com; kapil@nitkkr.ac.in).

Nitin Kumar is with the National Institute of Technology Uttarakhand, Srinagar 246174, India (e-mail: nitin@nituk.ac.in).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TIA.2022.3153757.

Digital Object Identifier 10.1109/TIA.2022.3153757

is based on the idea of decomposing the several classes hierarchically, where every cluster contains a set of classes that are similar to each other and a module is a subpart of another module, leading to a recursive partition of clusters creating hierarchy of classes. The hierarchy is based on the notion of integrating the similar categories in one cluster and opting out dissimilar in another ones. The approach aims to maximize the intercluster differences and minimize the intracluster distance. Decomposition is based on a separability graph, where every node is an individual class and edges represent the measurement of how much connected nodes are separated. Effectiveness of the proposed hierarchical framework is evaluated on the real-world and complex problems, such as plant recognition. Leaf image datasets are considered for the same.

This article is an extension of [13]. The rest of this article is organized as follows. Section II represents the related work and the literature survey. The proposed work is represented in Section III. The experimental setup, dataset details, and classification tools used in this article are represented in Section IV. Section V discusses the performance evaluation and comparative analysis. Finally, Section VI concludes this article.

II. Related Work

This section covers a brief discussion on various research works done on plant identification and process for transforming leaf images to the feature vector.

A. Plant Identification: A Study

Recognizing plant species and confirming unknown species based on various plant organs is a challenging task. It is challenging for a specialist from different fields [14]. It requires training to learn about nature and to get familiar with the botanical background. With the increasing dominance of keyword-based search engines, such as Google, people can efficiently find information for the specific plant using suitable keywords. However, finding relevant words is again a challenging task for most people. A variety of studies have examined societal knowledge of species identification, seeing mixed results. Some studies that have been conducted have typically found low levels of plant identification skills [15]–[19].

Several years ago, Gaston and O'Neill [20] argued that the successful development of information technology skills will make automated identification tangible soon. The enormous growth of artificial intelligence techniques, advancement of approaches in digital image processing, and portable devices, i.e., cameras and smartphones, have brought the identification means closer to existence [21], [22]. In the classic era, experts use the plant's visible credentials for the standard cataloguing of various plants. The expertise from different fields exploits distinguishing highlights of flowers, fruits, leaves, plants' structure, and other organs. However, fruits and flowers are periodical plant organs and may not be available throughout the year. From the research studies conducted so far, leaves are the most prominent descriptor to classify plants. Leaves are only vastly available plant organs, and these are available throughout the year [1], [23]. The plant species identification system has a variety of applications ranging from conservation to agriculture. Various remarkable applications are available for recognizing plant species using leaf and other plant organ's images, i.e., Pl@ntNet [24], Leafsnap [25], and Apleaf [26]. The noted image processing technique and the machine learning algorithm for the automated identification model are widely discussed in research studies conducted in recent years, comprehensively surveyed by Wäldchen *et al.* [1], [23], [27] and Cope *et al.* [28]. Various researchers have proposed algorithms to extract patterns, features based on shape, vein, color, and texture, and the fusion of two or more features [29]–[36].

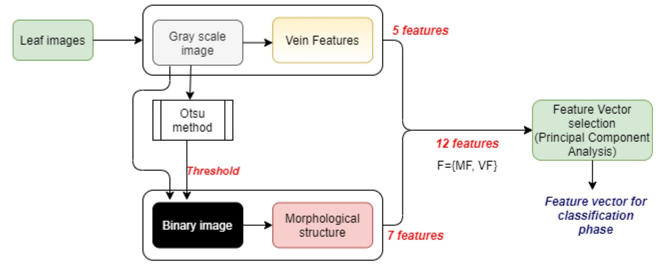

Fig. 1. Fundamental phases for transforming a leaf image to a feature vector for classification.

Plant species identification using digital images is a sequential process with multiple phases. Initially, RGB digital images are given as input. Various methods will enhance the input images in the preprocessing stage to remove undesired distortions, noise, and redundant information followed by feature extraction. Feature extraction intends to explore meaningful characteristics. This step is crucial as it converts a 3-D image into a 1-D feature vector. The last phase is the machine learning phase that trains a classification model with a feature vector extracted in the feature extraction phase. The phase assigns a plant name to unseen plant images using the trained model.

B. Transforming Leaf Images to Features Representation

This article follows the steps described in Fig. 1 to transform a leaf image into a feature vector for classification. After taking images as input, an image preprocessing phase includes averaging filtering and Laplacian filter for image smoothing, image denoising, and background removal. Leaf morphology and vein features (see Table I) are derived from processed leaf images [13], [37]. Principal component analysis is used to reduce the complexity with removing redundant information.

III. Proposed Work: Hierarchical Framework

Integrating the benefits of partitioning-based and hierarchy-based multiclass classifications, we propose the hierarchical framework for multiclass classification problems. Initially, samples from all classes are kept on the root node assuming all samples in one clusters. Classes are decomposed to form a cluster based on the similarity between each pair of classes. Each cluster maintains the notion of maximizing the intercluster separability

TABLE I
FEATURE DESCRIPTION

Feature Category	Feature Name	Description
Morphological Feature	Smooth Factor	$\frac{SA_5}{SA_2}$
	Aspect Ratio	$\frac{L_p}{W_p}$
	Form Factor	$\frac{4\times\pi\times A}{P^2}$
	Rectangularity	$\frac{L_p\times W_p}{A}$
	Narrow Factor	$\frac{D}{L_p}$
	Ratio of Perimeter to Diameter	$\frac{P}{D}$
	Perimeter ratio of L_p and W_P	$\frac{P}{L_p+W_p}$
Vein features	v_1	$\frac{av_1}{A}$
	v_2	$\frac{av_2}{A}$
	v_3	$\frac{av_3}{A}$
	v_4	$\frac{av_4}{A}$
	v_5	$\frac{av_4}{av_1}$

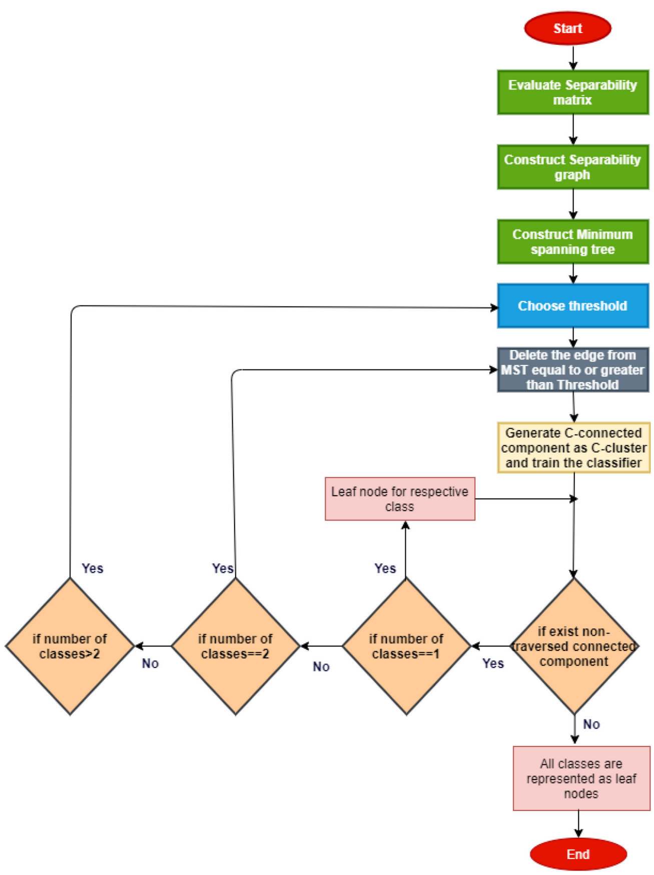

Fig. 2. Building hierarchical framework for multiclass classification.

and the intracluster similarity. The second step is to train the classification model using an OVO approach, considering one cluster as one class. In the tree structure, each nonleaf represents one classifier trained using classes on each child node as one class, and every leaf node represents an individual class. For a new sample, classification starts from the root node to a leaf node, following a traversal path. To construct the hierarchical framework, steps are given in Algorithm 1. Fig. 2 represents the flowchart for the proposed hierarchical framework for multiclass classification.

Algorithm 1: Building Hierarchical Structure.

1: Add all samples to the root node with the corresponding class label.
2: Find the number of classes at the current node (r). If $r > 2$, then go to the next step. If $r = 2$, then go to step 4.
3: Find $c-$ cluster for $k-$ classes using Algorithm 2.
4: Train classification model considering every cluster as a class using the partitioning-based approach.
5: Create $k-$ child node of the current node, each cluster as a child node.
6: Repeat steps 2–5 for each child node created.

Algorithm 2: Find Clusters.

Input: Training data $\mathbb{T}$
Output: $c-$ clusters

1: Evaluate separability matrix using training samples $\mathbb{T}$.
2: Construct separability graph with k nodes correspond to k classes and $k(k-1)/2$ edges representing the separability between corresponding classes
3: Build minimum spanning tree using the separability graph, i.e., k nodes correspond to k classes and $k-1$ edges representing the separability between corresponding classes
4: Find threshold (W, i.e., weights of edges in MST) case 1:

$$\text{threshold} = \frac{1}{k}\sum_{i=1,\ldots,k-1} W_i$$

case 2:

$$\text{threshold} = \max_{i=1,\ldots,k-1}(W_i)$$

5: Remove the edges greater than or equal to the threshold
6: $c-$ connected component after deleting edges representing $c-$ clusters

A. Separability Matrix

In the process of hierarchy generation, the separability matrix is computed using the training data and corresponding class labels. It measures the goodness of clustering by considering how well the clusters are separated and how compact the clusters are. In this article, an individual class is viewed as a cluster. A similarity index is used to generate hierarchy. Two other widely used clustering evaluation metrics, viz., silhouette value and Dunn index (DI), are explored to identify the similarity between each pair of classes [38]–[41].

1) Similarity Index: Similarity index represent a metric for the given training samples grouped in two classes, how similar that point is to points in its own cluster compared to points in other clusters, and ranges from 0 to +1. These metrics define distance and distribution of samples in the high-dimensional space.

Mean: mean_i defines the center of the given ith class

$$\text{mean}_i = \frac{1}{n_i}\sum_{s=1}^{n_i} x_s. \tag{1}$$

Distance: $\text{dist}_{i,j}$ defines the corresponding distance between ith and jth classes

$$\begin{aligned}\text{dist}_{i,j} &= ||\text{mean}_i - \text{mean}_j|| \\ &= \left|\left|\frac{1}{n_i}\sum_{s=1}^{n_i} x_s - \frac{1}{n_j}\sum_{t=1}^{n_j} x_t\right|\right|. \end{aligned} \tag{2}$$

Distribution: D_i represents the maximum distance of any sample from the mean of the given class

$$D_i = \max_{s=1,\ldots,n_i} ||x_s - \text{mean}_i||. \tag{3}$$

Based on these three metrics, similarity between each pair of classes can be defined as

$$\text{SIM}_{i,j} = \frac{(D_i^2 - D_j^2)}{\text{dist}_{i,j}}. \tag{4}$$

Similarity indicates the classification area between two classes; higher similarity means minimum classification area. In other words, to maximize the classification area between two classes, replace similarity with separability

$$\text{SEP}_{i,j} = \frac{1}{\text{SIM}_{i,j}}. \tag{5}$$

2) Silhouette Value (S_V): The silhouette value is a metric used to determine the goodness of a clustering technique and ranges from -1 to 1. The value closer to -1 represents an overlapping cluster, and well-clustered classes will take a value near to 1. The zero value shows insignificant results

$$S_V = \frac{b-a}{\max(b,a)}. \tag{6}$$

Here, a and b signify the average intracluster distance and the average intercluster distance, respectively.

3) Dunn Index: The DI confirms the compactness of the clusters. The higher value of the DI indicates that clusters are well separated from one another. DI values are evaluated by dividing the minimum intercluster distance by the maximum cluster size. It is inferred that maximizing the intercluster distances (better separation) and smaller cluster sizes (more compact clusters) leads to a higher DI value.

IV. Experimental Setup

This section details the experimental setup, the dataset, and the classification tool used for training an efficient recognition system. For comparative analysis, various classification metrics are also discussed.

A. Dataset Used

This article represents the application of the proposed framework for recognizing plant species using leaf images. For this purpose, two publicly available datasets, i.e., Flavia and Swedish, and one dataset containing self-collected leaf images

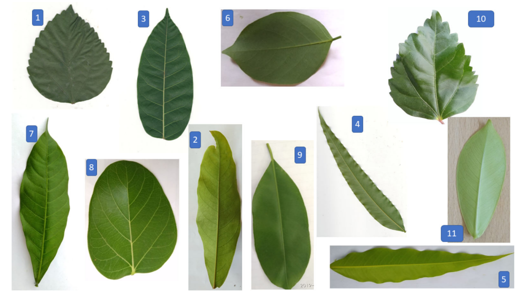

Fig. 3. Sample images from the self-collected dataset.

are used. As Flavia and Swedish datasets contain high-quality scanned images with optimized scanning devices, Flavia leaf images are very much dissimilar to each other. In addition, all the images are clean and high-resolution images with a white background. Swedish leaves are also scanned images with a white background, but the leaves are similar in shape. The leaves contain stalk and represent clear and noiseless images. The Swedish dataset is a challenging dataset as leaves are similar in shape. For evaluating the framework, experiments are performed on these two datasets along with self-collected leaf images. The self-collected dataset contains 747 leaf images representing different 11 classes. Leaf images are captured on a homogeneous background using an android mobile phone and a scanning device. Leaf images are noisy and contain shadows. It can be observed from Fig. 3 that some of the leaves are very much similar in shape and vein networks. One of the plant categories in the self-collected dataset contains damaged and diseased leaves, i.e., class 2.

B. Classification Tool and Evaluation Metrics

Different analyses with the Flavia dataset guarantee that the proposed algorithm is an efficient and effective approach with significantly improved computational cost and classification accuracy [13]. Similar to this article, for the SVM, the LibSVM [42] with a radial basis function kernel is used to train the hierarchical structure and classification of a new sample. This article discusses the experiments and comparative analysis of both hierarchical structures using the similarity index and two benchmark clustering evaluation metrics, i.e., silhouette value and DI. The dataset is partitioned into seven distinct training testing subsets according to the training-to-testing ratio (RTT) {60:40; 70:30; 75:25; 80:20; 85:15; 90:10; 95:5}. These subsets are designed so that the RTT is identical for all n classes. For comparative analysis, accuracy and other evaluation criteria based on confusion metric, i.e., precision, recall, F-score, and area under the curve (AUC), are explored.

V. Result Discussion and Analysis

This section details the results and experiments performed with the proposed framework. The proposed framework is a robust and efficient hierarchical approach for multiclass classification. Two different cases are discussed with the mean value as a threshold that transforms n classes to k connected components ($k \geq 2$), i.e., an n-ary tree. The

TABLE II
CLASSIFICATION ACCURACY OF TEST LEAF IMAGES FROM DIFFERENT DATASETS WITH THE PROPOSED FRAMEWORK

(a) Flavia dataset

RTT	N-ary (case 1)			Binary (case 2)		
	Similarity index	Silhouette Value	Dunn index	Similarity index	Silhouette Value	Dunn index
60:40	86.37	84.40	85.06	83.26	82.44	85.32
70:30	85.66	81.52	83.94	83.25	82.56	83.42
75:25	86.01	85.59	84. 34	84.97	84.13	86.85
80:20	87.11	84. 21	86.84	86.32	83.68	85.26
85:15	87.76	88.11	89.51	87.06	89. 16	88.46
90:10	86.53	87.56	86. 53	83.94	87.05	87.56
95:05	92.92	81.82	82.83	88.89	80.81	81.82

(b) Swedish dataset

RTT	N-ary (case 1)			Binary (case 2)		
	Similarity index	Silhouette Value	Dunn index	Similarity index	Silhouette Value	Dunn index
60:40	93.33	93.59	91.54	93.33	92.56	90.51
70:30	93.65	92.98	91.97	92.64	89.30	92.31
75:25	93.93	91.50	91.09	93.52	91.90	92.71
80:20	93.85	92.82	92.82	95.38	90.26	91.79
85:15	92.31	90.21	90.91	92.31	93.01	92.31
90:10	90.38	90.38	90.38	89.42	89.42	90.38
95:05	94.23	92.31	90.38	94.23	90.38	90.38

(c) Self-collected dataset

RTT	N-ary (case 1)			Binary (case 2)		
	Similarity index	Silhouette Value	Dunn index	Similarity index	Silhouette Value	Dunn index
60:40	89.30	88.63	87.96	89.63	86.62	85.28
70:30	91.93	88.34	88.34	90.58	88.79	87.44
75:25	92.02	85.64	87.23	91.49	86.17	88.83
80:20	92.62	92.62	90.60	92.62	91.95	91.95
85:15	93.75	91.96	89.29	93.75	91.96	91.96
90:10	92.11	89.47	89.47	90.79	88.16	89.47
95:05	97.22	91.67	91.67	100.00	91.67	97.22

second case utilizes the maximum weight of minimum spanning tree (MST) as a threshold, provides exactly two connected components, and creates a binary tree structure. The experimental results discussed in Tables II and III with Figs. 4–6 confirmed the significance and robustness of the proposed framework. Table II outlines the classification accuracy, and Table III represents the AUC as a performance metric.

Table II describes the efficiency, i.e., measures the percentage of correctly classified samples on leaf image datasets. In the table, results correspond to the n-ary structure to confirm the efficiency of the proposed approach in contrast to the other two metrics of the separability matrix, i.e., silhouette value and DI. For the Flavia dataset, classification improved by 11% with the similarity index as the separability matrix. The proposed hierarchical approach improves Swedish leaf image and self-collected datasets by 4% and 6%, respectively. A binary tree structure with the maximum weight from MST as threshold performs equally with three separability matrices on Flavia and Swedish leaf images. With the self-collected leaf images, the proposed method, i.e., similarity index, classifies leaf images with 100% accuracy and improves the outcomes by 9% compared to the silhouette value. Only an accuracy measure is inadequate to assess the performance in a dataset with multiple classes and irregular distribution. Other evaluation metrics, i.e., precision, recall, and F-score, are considered to evaluate the performance of the classification model.

Precision measures how many leaf samples we labeled as specific plant species are actually from that plant. Precision $=$ $\mathrm{TP}/(\mathrm{TP}+\mathrm{FP})$. A precision metric is considered when we want to emphasize the confidence value of true positives. For the Flavia dataset (see Fig. 4), in five out of seven subsets of datasets, the similarity index is outperforming. The precision value of 94.74% for the proposed approach using Flavia leaf images indicates the optimal performance with 5% misclassification. In contrast, the misclassification rate of the silhouette value and the DI is approximately 14%. Similar to the Flavia leaf image, the proposed framework's performance in the other two datasets strengthens the efficiency and the robustness (see Figs. 5 and 6). The precision value of 100% for the similarity index with own dataset puts more confidence on the classification model. Higher precision value on all the datasets with the proposed classification framework with the similarity index assures the efficiency and the effectiveness of the framework.

The recall is the ratio of the correctly classified sample in one class using the classifiers to all samples actually lying in that class weather predicted in any class. Recall $=\mathrm{TP}/(\mathrm{TP}+\mathrm{FN})$. A recall metric is explored when misclassification of the negative class sample is unacceptable. The best recall value of the Flavia dataset with the similarity index in an n-ary hierarchical structure is 93%. The recall value of 93% interprets that, on average, 7% representatives from every class are misclassified. On the same subset of the training dataset, the strictly binary structure classifies 89% samples correctly in own class. The similarity index performance with the Swedish leaf images ensures the classification model robustness. With the Swedish leaves, the model gives a recall value of 94% with the mean

TABLE III
ANALYSING PERFORMANCE USING THE AUC FOR TEST LEAF IMAGES FROM DIFFERENT DATASETS WITH THE PROPOSED FRAMEWORK

(a) Flavia dataset

RTT	N-ary (case 1)			Binary (case 2)		
	Similarity index	Silhouette Value	Dunn index	Similarity index	Silhouette Value	Dunn index
60: 40	92.78	91.55	91.94	91.19	90.53	92.08
70:30	92.25	90.08	91.37	91.04	90.63	91.09
75: 25	92.43	92.16	91.63	91.92	91.49	93.01
80:20	93.15	91.51	92.98	92.87	91.19	92.29
85:15	93.40	93.51	94.27	93.17	94.11	93.75
90:10	92.72	93.18	92.89	91.50	93.05	93.28
95:05	96.24	90.33	90.87	94.09	89.79	90.33

(b) Swedish dataset

RTT	N-ary (case 1)			Binary (case 2)		
	Similarity index	Silhouette Value	Dunn index	Similarity index	Silhouette Value	Dunn index
60:40	96.39	96.53	95.42	96.39	95.97	94.86
70:30	96.56	96.20	95.65	96.01	94.20	95.83
75:25	96.71	95.39	95.18	96.49	95.61	96.05
80:20	96.67	96.11	96.11	97.50	94.72	95.56
85:15	95.83	94.70	95.08	95.83	96.21	95.83
90:10	94.79	94.79	94.79	94.27	94.27	94.79
95:05	96.88	95.83	94.79	96.88	94.79	94.79

(c) Self-collected dataset

RTT	N-ary (case 1)			Binary (case 2)		
	Similarity index	Silhouette Value	Dunn index	Similarity index	Silhouette Value	Dunn index
60: 40	94.02	93.63	93.20	94.19	92.17	91.82
70:30	95.86	94.00	93.94	95.15	94.23	93.50
75: 25	95.64	91.69	92.86	94.99	92.17	93.51
80:20	95.24	95.54	93.82	95.19	95.19	94.71
85:15	96.80	95.81	94.49	96.71	95.82	95.73
90:10	96.01	94.57	94.57	95.29	93.94	94.57
95:05	98.73	95.79	95.79	100.00	96.17	98.73

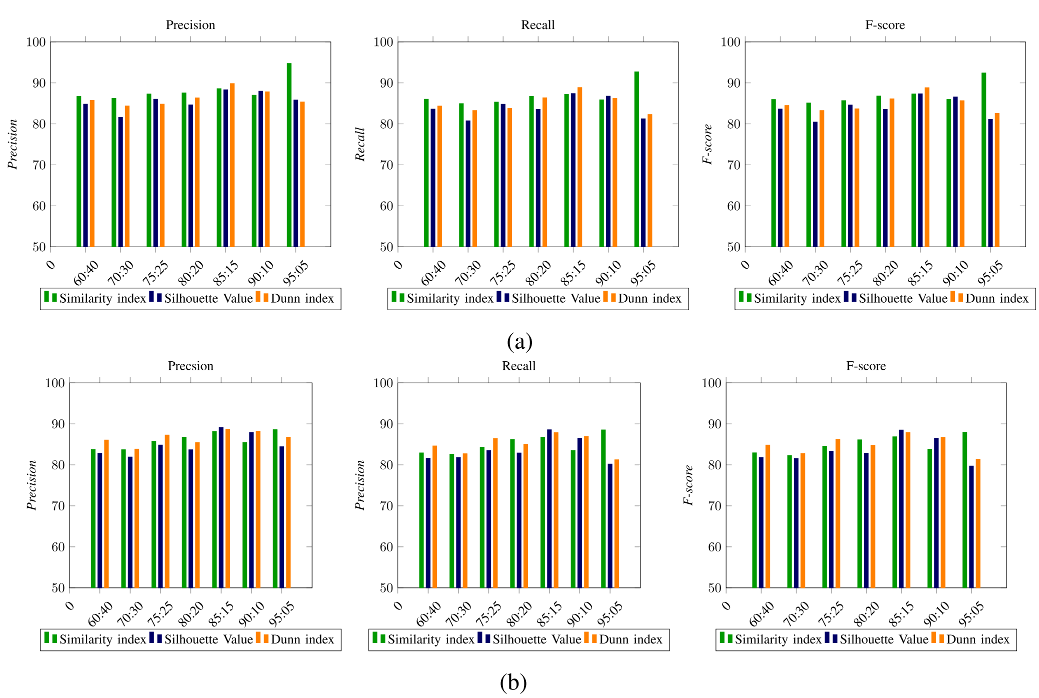

Fig. 4. Performance analysis on the Flavia dataset. (a) N-ary structure (Case 1 threshold). (b) Binary structure (Case 2 threshold).

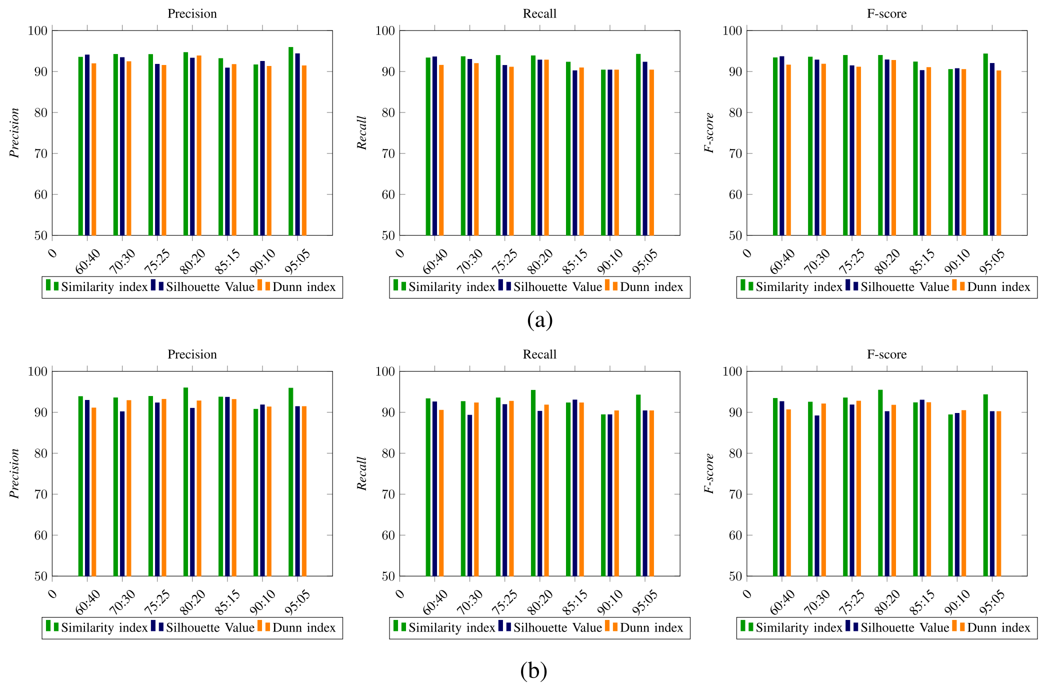

Fig. 5. Performance evaluation using various measures on Swedish leaves. (a) N-ary structure (Case 1 threshold). (b) Binary structure (Case 2 threshold).

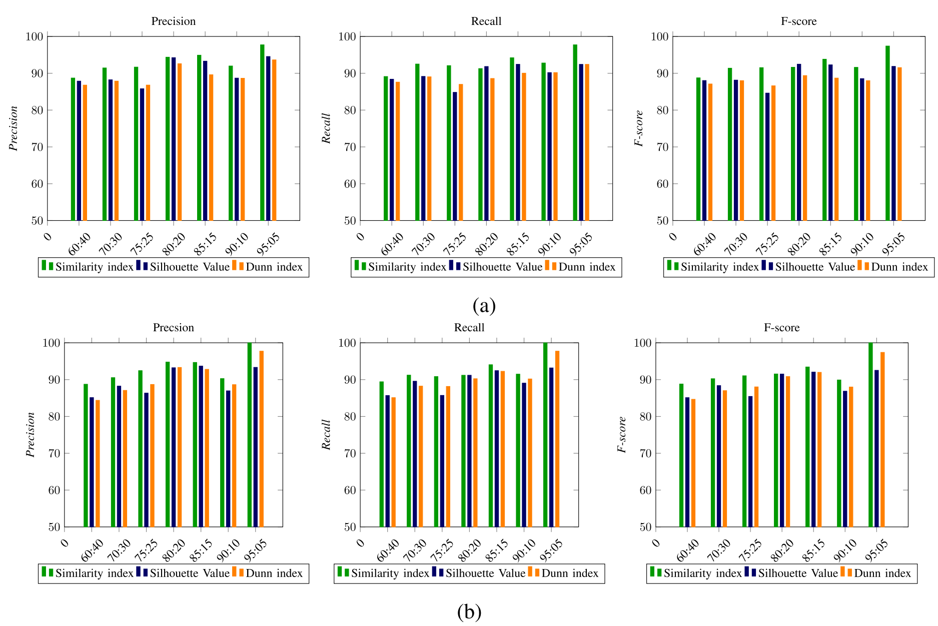

Fig. 6. Performance analysis on the self-collected dataset. (a) N-ary structure (Case 1 threshold). (b) Binary structure (Case 2 threshold).

value as threshold and a recall value of 95% with the binary structure. The classifier recall value is 98% and 100% with n-ary and binary forms, respectively, using a self-collected dataset. F-measure produces a value that balances precision and recall. The F-score is usually more valuable than accuracy, especially with the datasets having imbalanced classes. The F-score value for all the experiments confirms the discussed results. The higher value of F-score indicates the accuracy of the model. For the self-collected leaf images, the minimum F-score value is 88.76%, and the maximum F-score of 97.40% infers the significance of the proposed model for multiclass classification. For the strictly binary structure, the F-score ranges from 88.81% to 100%. For the Flavia leaf images, the F-score is improved by approximately 10% using the similarity index with the n-ary hierarchical structure and 6% in binary format.

The recall is the ratio of the correctly classified sample in one class using the classifiers to all samples lying in that class weather predicted in any class. Recall $=$ TP$/($TP $+$ FN$)$. A recall metric is explored when misclassification of the negative class sample is unacceptable. The best recall value of the Flavia dataset with the similarity index in an n-ary structure is 93%, which interprets that, on average, 7% representatives from every class are misclassified. The similarity index performance with the Swedish leaf images ensures the model robustness with a recall value of 94% in n-ary form and 95% with the binary structure. The classifier recall value is 98% and 100% with n-ary and binary forms, respectively, using a self-collected dataset. F-measure produces a value that balances precision and recall. The F-score is usually more valuable than accuracy, especially with the datasets having imbalanced classes. For the self-collected leaf images, the F-score ranges from 88.81% to 100% and 88.76% to 97.40% with binary and n-ary hierarchies, respectively. For the Flavia leaf images, the F-score is improved by approximately 10% using the similarity index with the n-ary hierarchical structure and 6% in binary format.

Table III shows the AUC as another evaluation metric. It is the measure of the ability of a classifier to distinguish two classes and is explored as a review of the receiver operating characteristic curve. The average of AUC measures is represented for all the classes from the related dataset in the table. Again, the similarity index proves its efficiency over the rest two with significant results on all datasets. It is also noticed that results on self-collected leaf images show better performance despite having noise and diseased and damaged leaves.

A. Analysis and Model Validation

The proposed framework extends two-class classification tasks to multiclass classification problems with combined benefits of two widely used strategies. It is a robust and efficient hierarchical approach that offers several advantages with the two different tree structures, i.e., binary tree and n-ary tree. The proposed method overcomes the various challenges caused by studies discussed in the literature. The experimental results confirm the efficiency of the proposed hierarchical framework with a similarity matrix and elaborate advantages over the existing multiclass classification strategies. The most widely partition-based approach explored in the existing studies is the OVR method. This approach combines the number of classes as one class that causes nonlinearity and imbalance class classification problems. The other most widely used technique, i.e., OVO, needs many classifiers that increase the computational cost. Discussed hierarchical approaches are not efficient because of biased partitioning, scattering samples in more than one cluster. The proposed hierarchical approach overcomes all the challenges and an efficient classification system with the comparable accuracy and the least computational cost.

Classification accuracy and other performance measures are described in tables and graphs. The number of classifications in both cases is significantly less than that of OVO partitioning-based multiclass extension. For the Flavia dataset with 32 classes, the OVO traditional approach of multiclass needs 496 classifiers to train, and the proposed hierarchy with the binary structure requires only 31 classifiers. Similarly, with the other two datasets, i.e., Swedish and self-collected datasets, the number of classifiers needed to train reduced to 12 and 10 from 78 and 55, respectively. To classify a new sample in the known category, OVO requires experimenting with all the classifiers. Still, the proposed one deals with classifiers lying on the traversing path of hierarchy. Another benefit of the proposed classification is a fair and automated approach. The hierarchy does not require human feedback. It also reduces the impact of imbalanced class distribution and nonlinearity by grouping similar classes in clusters. Most distinguished types are separated at the upper level; the lower hierarchy level represents that the classes with lesser separability lead to better generalization. Furthermore, the least computational cost to train the classification model strengthens the classification model with the faster classification of new samples. With the significant improvement in classification performance using the similarity index, the speedup factors of the proposed approach are 16, 6.5, and 5.5 with the proposed algorithm in the binary structure as compared to the traditional OVO SVM. The results discussed in this article highlight the impact of the similarity index to enhance the proposed framework compared to widely discussed clustering evaluation metrics. Having several advantages, the other side of the algorithm needs improvement. With the distinct classes, all the categories represented as individual clusters at the initial level will lead us toward the OVO approach. With better generalization and efficient results, the model will require a high computational cost to train and test a new sample with increased classification time. Another challenge is to increase the model validity toward the leaf model in the hierarchy, classes with lesser separability. To validate the model, we have performed fivefold cross validation to tune the hyperparameters of individual classifiers in the hierarchy. Furthermore, the Wilcoxon rank sum test is performed with the assumption that the classification model is similar to comparing the three different separability measures. Table IV represents the p-value with the test result either rejected the null hypothesis or failed to reject. Classification performance on Swedish leaf images and self-collected leaf image confirms the model strength.

TABLE IV
P-VALUE OF THE WILCOXON RANK SUM TEST

Dataset	significance	Comparative analysis		
		Proposed v/s S_V	Proposed v/s DI	S_V v/s DI
Flavia	10%	0.09(R)	0.17(FR)	0.71(FR)
Swedish	5%	0.4(R)	0.3(R)	0.40(FR)
Self-collected	5%	0.04 (R)	0.01(R)	0.47(FR)

R: Reject; FR: Failed to reject.

VI. CONCLUSION

This article introduced an approach for multiclass classification by grouping similar class samples as one cluster from a pool of classes. The process was repeated until all the classes were represented individually. This article discussed the similarity index based on the distance and distribution of instances to combine similar classes. The similarity index tended to maximize the intercluster distance and minimize the intracluster distribution. For comparative analysis of the hierarchical framework with the similarity index, the most widely discussed clustering evaluation metrics, i.e., silhouette value and DI, were used to know the cluster's compactness and identify similar classes from the pool. The generated hierarchy represented two different structures, viz., binary tree and n-ary tree, using two distinct types of thresholds. Several experiments with three leaf image datasets assure the efficiency and the robustness of the proposed approach. Few concluding statements are as follows.

1) Hierarchy reduces the impact of imbalanced class distribution.
2) It overcomes the issue of nonlinearity, as clusters are formed based on similar classes.
3) The least computational cost is required to train the model with lesser classifiers with faster classification.
4) In the automated approach, human feedback is not required for selecting similar categories.

References

[1] J. Wäldchen, M. Rzanny, M. Seeland, and P. Mäder, "Automated plant species identification—Trends and future directions," *PLoS Comput. Biol.*, vol. 14, no. 4, 2018, Art. no. e1005993.

[2] C.-W. Hsu and C.-J. Lin, "A comparison of methods for multiclass support vector machines," *IEEE Trans. Neural Netw.*, vol. 13, no. 2, pp. 415–425, Mar. 2002.

[3] I. Guyon, J. Weston, S. Barnhill, and V. Vapnik, "Gene selection for cancer classification using support vector machines," *Mach. Learn.*, vol. 46, no. 1, pp. 389–422, 2002.

[4] T. Nguyen, A. Khosravi, D. Creighton, and S. Nahavandi, "Classification of healthcare data using genetic fuzzy logic system and wavelets," *Expert Syst. Appl.*, vol. 42, no. 4, pp. 2184–2197, 2015.

[5] M. A. Chandra and S. Bedi, "Survey on SVM and their application in image classification," *Int. J. Inf. Technol.*, vol. 13, pp. 1–11, 2018.

[6] C. Cortes and V. Vapnik, "Support vector machine," *Mach. Learn.*, vol. 20, no. 3, pp. 273–297, 1995.

[7] J. H. Friedman, "Another approach to polychotomous classification," *Dept. Statist.*, Stanford Univ., Stanford, CA, USA, Tech Rep., 1996.

[8] L. Bottou *et al.*, "Comparison of classifier methods: A case study in handwritten digit recognition," in *Proc. 12th IAPR Int. Conf. Pattern Recognit.*, 1994, pp. 77–82.

[9] L. Cheng, J. Zhang, J. Yang, and J. Ma, "An improved hierarchical multiclass support vector machine with binary tree architecture," in *Proc. Int. Conf. Internet Comput. Sci. Eng.*, 2008, pp. 106–109.

[10] J. C. Platt, N. Cristianini, and J. Shawe-Taylor, "Large margin dags for multiclass classification," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2000, pp. 547–553.

[11] F. Nie, W. Zhu, and X. Li, "Decision tree SVM: An extension of linear SVM for non-linear classification," *Neurocomputing*, vol. 401, pp. 153–159, 2020.

[12] C. Dong, B. Zhou, and J. Hu, "A hierarchical SVM based multiclass classification by using similarity clustering," in *Proc. Int. Joint Conf. Neural Netw.*, 2015, pp. 1–6., Kapil

[13] N. Goyal, K. Gupta, and N. Kumar, "A hierarchical approach for multiclass classification of leaf images," in *Proc. IEEE Int. Conf. Comput., Power Commun. Technol.*, 2020, pp. 268–273.

[14] B. Press, "Mabberley's plant-book: A portable dictionary of plants, their classification and uses," *J. Botanical Res. Inst. Texas*, vol. 12, no. 2, pp. 578–578, 2018.

[15] B. C. Stagg and M. Donkin, "Teaching botanical identification to adults: Experiences of the UK participatory science project 'open air laboratories'," *J. Biol. Educ.*, vol. 47, no. 2, pp. 104–110, 2013.

[16] A. Bebbington, "The ability of A-level students to name plants," *J. Biol. Educ.*, vol. 39, no. 2, pp. 63–67, 2005.

[17] S. Gatt, S. D. Tunnicliffe, K. Borg, and K. Lautier, "Young Maltese children's ideas about plants," *J. Biol. Educ.*, vol. 41, no. 3, pp. 117–122, 2007.

[18] M. Dallimer *et al.*, "Biodiversity and the feel-good factor: Understanding associations between self-reported human well-being and species richness," *BioScience*, vol. 62, no. 1, pp. 47–55, 2012.

[19] B. S. Robinson, R. Inger, and K. J. Gaston, "A rose by any other name: Plant identification knowledge & socio-demographics," *PLoS One*, vol. 11, no. 5, 2016, Art. no. e0156572.

[20] K. J. Gaston and M. A. O'Neill, "Automated species identification: Why not?," *Philos. Trans. Roy. Soc. London. Ser. B: Biol. Sci.*, vol. 359, no. 1444, pp. 655–667, 2004.

[21] C. A. Rademaker, "The classification of plants in the United States patent classification system," *World Pat. Inf.*, vol. 22, no. 4, pp. 301–307, 2000.

[22] P. N. Belhumeur *et al.*, "Searching the world's herbaria: A system for visual identification of plant species," in *Proc. Eur. Conf. Comput. Vis.*, 2008, pp. 116–129.

[23] J. Wäldchen and P. Mäder, "Machine learning for image based species identification," *Methods Ecol. Evol.*, vol. 9, no. 11, pp. 2216–2225, 2018.

[24] H. Goëau *et al.*, "PlntNet mobile app," in *Proc. 21st ACM Int. Conf. Multimedia*, 2013, pp. 423–424.

[25] N. Kumar *et al.*, "Leafsnap: A computer vision system for automatic plant species identification," in *Proc. Eur. Conf. Comput. Vis.*, 2012, pp. 502–516.

[26] Z.-Q. Zhao, L.-H. Ma, Y.-M. Cheung, X. Wu, Y. Tang, and C. L. P. Chen, "ApLeaf: An efficient android-based plant leaf identification system," *Neurocomputing*, vol. 151, pp. 1112–1119, 2015.

[27] S. Sachar and A. Kumar, "Survey of feature extraction and classification techniques to identify plant through leaves," *Expert Syst. Appl.*, vol. 167, 2021, Art. no. 114181.

[28] J. S. Cope, D. Corney, J. Y. Clark, P. Remagnino, and P. Wilkin, "Plant species identification using digital morphometrics: A review," *Expert Syst. Appl.*, vol. 39, no. 8, pp. 7562–7573, 2012.

[29] H. Ling and D. W. Jacobs, "Shape classification using the inner-distance," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, no. 2, pp. 286–299, Feb. 2007.

[30] R. Hu, W. Jia, H. Ling, and D. Huang, "Multiscale distance matrix for fast plant leaf recognition," *IEEE Trans. Image Process.*, vol. 21, no. 11, pp. 4667–4672, Nov. 2012.

[31] S. G. Wu, F. S. Bao, E. Y. Xu, Y.-X. Wang, Y.-F. Chang, and Q.-L. Xiang, "A leaf recognition algorithm for plant classification using probabilistic neural network," in *Proc. IEEE Int. Symp. Signal Process. Inf. Technol.*, 2007, pp. 11–16.

[32] E. Yigit, K. Sabanci, A. Toktas, and A. Kayabasi, "A study on visual features of leaves in plant identification using artificial intelligence techniques," *Comput. Electron. Agriculture*, vol. 156, pp. 369–377, 2019.

[33] R. M. Haralick, K. Shanmugam, and I. H. Dinstein, "Textural features for image classification," *IEEE Trans. Syst., Man, Cybern.*, vol. SMC-3, no. 6, pp. 610–621, Nov. 1973.

[34] S. Mahajan, A. Raina, X.-Z. Gao, and A. K. Pandit, "Plant recognition using morphological feature extraction and transfer learning over SVM and AdaBoost," *Symmetry*, vol. 13, no. 2, 2021, Art. no. 356.

[35] F. Huang, J. Zhang, Q. Shan, C. Cai, and H. Liu, "The research of the plant leaves identification method based on 3-layers BP neutral network," *Cluster Comput.*, vol. 22, no. 5, pp. 11143–11152, 2019.

[36] P. P. Kaur and S. Singh, "Classification of herbal plant and comparative analysis of SVM and KNN classifier models on the leaf features using machine learning," in *Soft Computing for Intelligent Systems*, N. Marriwala, C. C. Tripathi, S. Jain, and S. Mathapathi, Eds. Singapore: Springer, 2021, pp. 227–239.

[37] N. Goyal, K. Gupta, and N. Kumar, "Multiclass twin support vector machine for plant species identification," *Multimedia Tools Appl.*, vol. 78, no. 19, pp. 27785–27808, 2019.

[38] L. Kaufman and P. J. Rousseeuw, *Finding Groups in Data: An Introduction to Cluster Analysis*, vol. 344. Hoboken, NJ, USA: Wiley, 2009.

[39] G. Ogbuabor and F. Ugwoke, "Clustering algorithm for a healthcare dataset using silhouette score value," *Int. J. Comput. Sci. Inf. Technol.*, vol. 10, no. 2, pp. 27–37, 2018.

[40] T. Thinsungnoena, N. Kaoungkub, P. Durongdumronchaib, K. Kerdprasopb, and N. Kerdprasopb, "The clustering validity with silhouette and sum of squared errors," in *Proc. Int. Conf. Ind. Appl. Eng.*, 2015, pp. 44–51.

[41] U. Maulik and S. Bandyopadhyay, "Performance evaluation of some clustering algorithms and validity indices," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 12, pp. 1650–1654, Dec. 2002.

[42] C.-C. Chang and C.-J. Lin, "LIBSVM: A library for support vector machines," *ACM Trans. Intell. Syst. Technol.*, vol. 2, no. 3, pp. 1–27, 2011.

Neha Goyal received the bachelor's and master's degrees in computer science from Kurukshetra University, Kurukshetra, India, in 2010 and 2012 respectively. She is currently working toward the Ph.D. degree with the National Institute of Technology Kurukshetra, Kurukshetra.

Her research interests include image processing and machine learning.

Ms. Goyal is the recipient of a Gold Medal in M.Sc. (computer science).

Kapil Gupta received the Master of Technology and Doctor of Philosophy degrees from Jawaharlal Nehru University, New Delhi, India, in 2009 and 2012 respectively, both in machine learning.

He is currently an Assistant Professor with the National Institute of Technology Kurukshetra, Kurukshetra, India. He has authored or coauthored various papers in the journals of international repute. His research interests include machine learning methods, wireless sensor networks, and their applications.

Nitin Kumar received the Master of Technology and Doctor of Philosophy degrees from the School of Computer and Systems Sciences, Jawaharlal Nehru University, New Delhi, India, in 2011 and 2015, respectively.

He is currently working as Assistant Professor with the Department of Computer Science and Engineering, National Institute of Technology Uttarakhand, Srinagar, India, since 2013. His current research interests include visual attention modeling, pattern recognition, face recognition and image processing.