ISSN: 2321-7782 (Online)
Volume 3, Issue 4, April 2015
International Journal of Advance Research in Computer Science and Management Studies
Research Article / Survey Paper / Case Study
Available online at: www.ijarcsms.com

Study on Classification of Plants Images using Combined Classifier

Subhankar Ghosh[1]
Dept. of Computer Science and Engineering
BMS College of Engineering
Bangalore – India

Harish Kumar V[2]
Dept. of Computer Science and Engineering
BMS College of Engineering
Bangalore – India

Pradeep Kumar[3]
Dept. of Computer Science and Engineering
BMS College of Engineering
Bangalore – India

Devaraj[4]
Dept. of Computer Science and Engineering
BMS College of Engineering
Bangalore – India

Jyothi S. Nayak[3]
Dept. of Computer Science and Engineering
BMS College of Engineering
Bangalore – India

Abstract: ***A classification problem deals with associating a given input pattern with one of the distinct classes. Plant leaf classification is a technique where a leaf is classified based on its different morphological features. There are various successful classification techniques like the k-Nearest Neighbour Classifier, Probabilistic Neural Network, Genetic Algorithm, Support Vector Machine, and Principal Component Analysis. Deciding on a specific method for classification is often a difficult task because the quality of the results can be different for different input data. Plant leaf classification has wide applications in various fields such as botany, Ayurveda, agriculture etc. In this paper we present a survey on the various classification techniques which can be used for the classification of plants based on their leaves.***

Keywords: classification; plants; probabilistic neural network; support vector machine; artificial neural networks; k-nearest neighbours; learning vector quantization.

I. INTRODUCTION

Plant recognition or classification has a broad range of applications in agriculture and medicine, and is especially significant to the biological diversity research. Plant leaf classification finds application in botany and in tea, cotton and other industries. Plants are vital for the protection of our environment. However, it is an important and difficult task to recognize the different species of plants. Many of them carry significant information for the development of human society. The urgent situation is that many plants are at the risk of extinction. So it is necessary to set up a database for plant protection. We believe that the first step is to teach a computer how to classify plants. Leaf recognition plays an important role in plant classification. Plants are basically identified based on flowers and fruits. However these are three dimensional objects and this increases the complexity of the process. Plant identification based on flowers and fruits require morphological features such as number of stamens in flower and number of ovaries in fruits. Identifying plants using such keys is a very time consuming task and has been carried out only by trained botanists. However, in addition to this time intensive task, there are several other drawbacks in identifying plants using these features such as the unavailability of required morphological information and use of botanical terms that only experts can understand. However leaves also play an important role in plant identification. Moreover, leaves can be easily found and collected everywhere at all seasons, while flowers can only be obtained during the blooming season. Shape

© 2015, IJARCSMS All Rights Reserved

of plant leaves is one of the most important features for characterising various plants visually. Plant leaves have two-dimensional structure and thus they are most suitable for machine processing.

Our paper presents survey of different classification techniques. Before classification can be done on the basis of leaves some pre-processing is needed. For classification different techniques are available. Some of them are k-Nearest Neighbor Classifier, Neural Network, Genetic Algorithm, Support Vector Machine, and Principal Component Analysis. We will discuss pre-processing to be performed on the acquired image. We have discussed overview of classification techniques and how they can be used for recognizing particular specie of a plant leaf.

II. Related Work

A. Classification Of Plant Images

Jayamala et al. carried out research on the various diseases that may affect plants. They selected three pests, prevalent in orchards, as the candidates for this research: the leaf-roller, codling moth, and apple leaf curling midge [1]. They used fast wavelet transform with special set of Doubenchies wavelet to extract the important features. To retrieve the related images, they carried out the search in two steps. The first step matched the images by comparing the standard deviations for the three color components. In the second step, a weighted version of the Euclidean distance between the feature coefficients of an image selected in the first step and those of the querying image was calculated and the images with the smallest distances were selected and sorted as matching images to the query. Stereomicroscopic method and Image analysis method were compared for usefulness of image analysis as an efficient and precise method to measure fruit traits such as size, shape dispersal related structures by Mix & Pic.

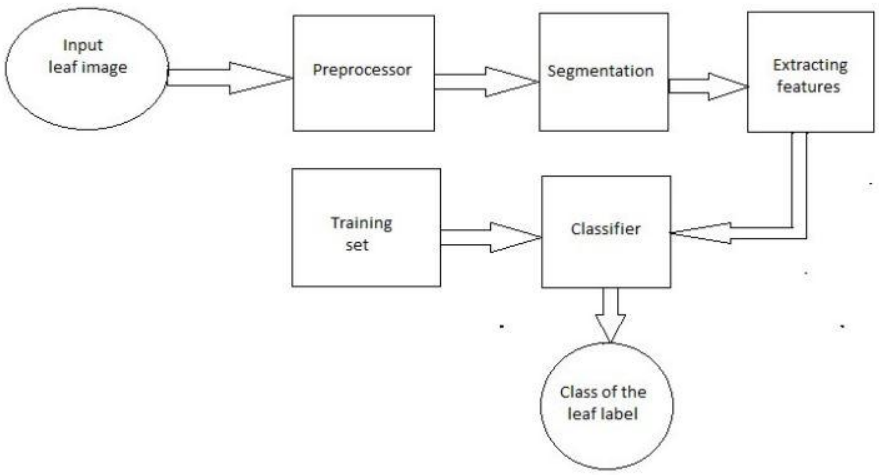

Figure 1: Block diagram for Plant Leaf classification

A neural network algorithm based on back propagation was used which indicated that both stereomicroscopic and image analysis accurately discriminated fruits of different sizes. This was reducing the subjectivity arising from human experts in detecting the plant diseases, and also damaging natural animal food chains. A common practice for plant scientists is to estimate the damage of plants (leaf, stem) because of disease by an eye on a scale based on percentage of affected area.

An efficient computer-aided plant species identification (CAPSI) approach was proposed by Xiang et al. based on plant leaf images using a shape matching technique [2]. First a Douglas - Peucker approximation algorithm was adapted to the original leaf shapes and a new shape representation was used to form the sequence of invariant attributes. Then a modified dynamic programming (MDP) algorithm for shape matching was proposed for the plant leaf recognition. Finally, the superiority of their proposed method over traditional approaches to plant species identification was demonstrated by experiment. The experimental result showed that their proposed algorithm for leaf shape matching is very suitable for the recognition of not only intact but also partial, distorted and overlapped plant leaves due to its robustness.

© 2015, IJARCSMS All Rights Reserved

Harini et al. used the techniques of PCA and Wavelets to identify the disease of an infected leaf [3]. While the PCA was used as a feature extraction algorithm, the Wavelets were used as a pre-processing algorithm in the proposed method. The work in this paper was divided into background information regarding the different type of diseases which infect tomato leaves. It deals with the basics of Wavelets and Principal component analysis. It proposed a methodology and inspired in the active field of CBIR. There is a new methodology for automatic identification of diseased leaves based on Wavelets and PCA.

Arunpriya et al. focused on tea leaves to identify the type of leaf for improving the process of classification [5]. Tea leaf images were loaded from digital cameras or scanners into the system. It consisted of three phases - pre-processing, feature extraction and classification to process the loaded image. The tea leaf images could be accurately identified in the pre-processing phase by fuzzy de-noising using Dual Tree Discrete Wavelet Transform (DT-DWT) in order to remove the noisy features and boundary enhancement to obtain the shape of the leaf accurately. In the feature extraction phase, Digital Morphological Features (DMFs) were derived to improve the classification accuracy. Radial Basis Function (RBF) was used for efficient classification. The RBF was trained by 60 tea leaves to classify them into 6 types. Experimental results proved that the proposed method classified the tea leaves with more accuracy in less time. Thus, the proposed method achieved more accuracy in retrieving the type of leaf at their disposal.

A simple yet computationally feasible method for plant recognition using leaf images was brought forward by Bong et al. [6]. Recognition of plant images is topic of extensive research in Computer Vision. Several authors suggest that object shape is more informative than its appearance properties such as texture and color as they vary between objects more than the shape. Initially leaf images were scanned which are two dimensional in nature and segmented using mathematical morphological segmentation. Then the high frequency feature of the image was extracted. For removing the noise, the image was converted into binary and then complemented and multiplied by the filtered image.

An automatic approach for plant species identification based on the visual information provided by the plant leaves was brought forward by Mouine et al. [7]. There are two sources of information- the leaf margin and the salient points on the leaf. Two shape based descriptors were used. The first one described the leaf boundary while the second represented the spatial. Then a large number of histograms were computed and compared.

Kadir et al. proposed a model of Plant Identification System using GLCM, Lacunarity and Shen Features [8]. The performance of such identification systems can be improved in various ways. Several experiments have been conducted in this area of research. As a result, a new novel approach using a combination of Gray-Level Co-occurrence Matrix, Lacunarity and Shen features along with a Bayesian classifier gave better results compared to other plant identification systems. For comparison, this research used two kinds of datasets that were used for testing the performance of each plant identification system.

A mobile application to identify the frame of a plant species was developed based on the computation of explicit leaf shape descriptors [9]. The paper focused on the characterization of the leaf contour, the extraction of its properties, and its description using botanical terms. Contour properties were investigated using the Curvature-Scale Space representation. The potential tooth was extracted and the margin was classified into a set of inferred shape classes.

A plant images classification based on textural features using Combined Classifier was brought forward by Rashad et al. [10]. 30 blocks of each texture were used as a training set and another 30 blocks as a testing set. The combined classifier gave a superior performance compared to other tested methods and programming algorithms for leaf shape matching.

There are many popular methods to extract the information of leaves which include Digital Morphology, Centroid Contour Distance (also known as Centroid-Radii Model), Moment Invariant and Polar Fourier Transform [11]. The different features of leaves to classify plants include shape, vein, color and texture. This algorithm produced a feasible result in the experiment leading us to the conclusion that the increasing accuracy of recognition is proportional to the rise in the number and significance

© 2015, IJARCSMS All Rights Reserved

of features used in the algorithm. Thus the authors proposed this method to get information on tip and base of the leaf and they believed that these two features are capable of improving the recognition result.

TABLE 1
Different types of plant leaves

Class	Common Names
Class 1	Chestnut leaf
Class 2	Golden rain tree
Class 3	Trident maple
Class 4	Chinese redbull
Class 5	Horse chestnut
Class 6	Bamboo
Class 7	Rose
Class 8	Eenbruinigherfstblad
Class 9	Autumn leaf
Class 10	Pipe
Class 11	Golden maple leaf
Class 12	Japan arrow wood
Class 13	Caster aralia
Class 14	Canadian popular

B. Classification Techniques

1) Principal Component Analysis (PCA)

The purpose of PCA is to present the information of original data as the linear combination of certain linear irrelevant variables. Mathematically, PCA transforms the data to a new coordinate system such that the greatest variance by any projection of the data comes to lie on the first coordinate, the second greatest variance on the second coordinate, and so on. Each coordinate is called a principal component.

Principal component analysis is a variable reduction procedure. It is useful when you have obtained data on a number of variables (possibly a large number of variables), and believe that there is some redundancy in those variables. Intuitively, Principal components analysis is a method of extracting information from a higher dimensional data by projecting it to a lower dimension. Principal component analysis is a basically used because it reduces the dimension of input vector of neural network. This method generates a new set of variables, called principal components. Each principal component is a linear combination of the optimally-weighted observed variables. All the principal components are orthogonal to each other, so there is no redundant information. The principal components as a whole form an orthogonal basis for the space of the data. Mathematically, PCA transforms the data to a new coordinate system such that the greatest variance by any projection of the data comes to lie on the first coordinate, the second greatest variance on the second coordinate, and so on [13]. Each coordinate is called a principal component. Often the variability of the data can be captured by a relatively small number of principal components, and, as a result, PCA can achieve high dimensionality reduction with usually lower noise than the original patterns. The objective of PCA is to perform dimensionality reduction while preserving as much of the randomness in the high-dimensional space as possible But the limitation with PCA is it depends on scaling of variables and it is not always easy to interpret principal components.

2) Decision Tree:

The decision tree classifier uses a layered or hierarchical approach to classification. At each level of the tree the attributes of a measurement are matched to a number of mutually exclusive nodes. The leaf nodes assign classes to the measurement. The classification of a measurement therefore involves a sequence of tests, with each successive test narrowing the interpretation. The sequence of tests for the classifier is determined during a training period. Given some training data *T*, the ideal solution would test all possible sequence of actions on the attribute of *T* in order to find the sequence resulting in the minimum number of misclassifications. The software used for the decision tree classifier is C5.0. It needs four types of files for generating the decision tree for a given data set, out of which two files are optional.

© 2015, IJARCSMS All Rights Reserved

The first file is the *.names* file. It describes the attributes and classes. The first line of the *.names* file gives the classes, either by naming a discrete attribute (the target attribute) that contains the class value, or by listing them explicitly. The attributes are then defined in the order that they will be given for each case. The attributes can be either explicitly or implicitly defined. The value of an explicitly defined attribute is given directly in the data. The value of an implicitly defined attribute is specified by a formula. The second file is the *.data* file. It provides information on the *training* cases from which C5.0 will extract patterns. The entry for each case consists of one or more lines that give the values for all explicitly defined attributes. The `?' mark is used to denote a value that is missing or unknown. It easy to choose datasets that had no missing features. Also, `N/A' denotes a value that is not applicable for a particular case.

The third file used by C5.0 consists of new test cases on which the classifier can be evaluated and is the *.test* file. This file is optional and, if used, has exactly the same format as the *.data* file. The last file is the *.costs* file. This file is also optional and sets out differential misclassification costs.

3) Naive–Bayes classifier (NBC):

A Naive Bayes classifier is a simple probabilistic classifier based on applying Bayes' theorem (from Bayesian statistics) with strong (naive) independence assumptions. A more descriptive term for the underlying probability model would be "independent feature model". In simple terms, a naive Bayes classifier assumes that the presence (or absence) of a particular feature of a class is unrelated to the presence (or absence) of any other feature. For example, a fruit may be considered to be an apple if it is red, round, and about 4" in diameter. Even if these features depend on each other or upon the existence of the other features, a naive Bayes classifier considers all of these properties to independently contribute to the probability that this fruit is an apple.

Depending on the precise nature of the probability model, naive Bayes classifiers can be trained very efficiently in a supervised learning setting. In many practical applications, parameter estimation for naive Bayes models uses the method of maximum likelihood; in other words, one can work with the naive Bayes model without believing in Bayesian probability or using any Bayesian method.

4) Bayesian Classification:

Bayesian classifiers are statistical classifiers. They can predict class membership probabilities, such as the probability that a given data item belongs to a particular class label. Bayesian classification [15] is based on Bayes Theorem as stated below: Let X is a data sample whose class label is not known and let H be some hypothesis, such that the data sample X belongs to a specified class.

$$\text{P (H/X)} = \frac{P\left(\frac{X}{H}\right).P(H)}{P(X)}$$

An advantage of the naive Bayes classifier is that it only requires a small amount of training data to estimate the parameters (means and variances of the variables) necessary for classification. Because independent variables are assumed, only the variances of the variables for each class need to be determined and not the entire covariance matrix.

4) Probabilistic Neural Network

Probabilistic neural networks can be used for classification problems. It has parallel distributed processor that has a natural tendency for storing experiential knowledge. PNN is derived from Radial Basis Function (RBF) Network. PNN basically works with 3 layers. First layer is input layer. The input layer accepts an input vector. When an input is presented, first layer computes distances from the input vector to the training input vectors and produces a vector whose elements indicate how close the input is to a training input [17]. The second layer sums these contributions for each class of inputs to produce as its net output a vector

© 2015, IJARCSMS All Rights Reserved

of probabilities. Radial Basis Layer evaluates vector distances between input vector and row weight vectors in weight matrix. These distances are scaled by Radial Basis Function nonlinearly [17]. The last layer i.e. competitive layer in PNN structure produces a classification decision, in which a class with maximum probabilities will be assigned by 1 and other classes will be assigned by 0.

5) Support Vector Machine

Support vector machine (SVM) is a non-linear classifier. The idea behind the method is to nonlinearly map the input data to some high dimensional space, where the data can be linearly separated, thus providing great classification performance. Support Vector Machine is a machine learning tool and has emerged as a powerful technique for learning from data and in particular for solving binary classification problems [17]. The main concepts of SVM are to first transform input data into a higher dimensional space by means of a kernel function and then construct an OSH (Optimal Separating Hyper Plane) between the two classes in the transformed space [17]. For plant leaf classification it will transform feature vector extracted from leaf's contour. SVM finds the OSH by maximizing the margin between the classes. Data vectors nearest to the constructed line in the transformed space are called the support vectors. The SVM estimates a function for classifying data into two classes. Using a nonlinear transformation that depends on a regularization parameter, the input vectors are placed into a high-dimensional feature space, where a linear separation is employed. To construct a nonlinear support vector classifier, the inner product (x, y) is replaced by a kernel function K (x, y), as in (1).

$$f(x) = \operatorname{sgn}\left(\sum_{i=1}^{l} (a_i y_i K(x_i x) + b)\right)$$

f(x) determines the membership of x. We assume normal subjects were labeled as -1 and other subjects as +1.The SVM has two layers [6]During the learning process, the first layer selects the basis K (xi, x), i=1, 2….N from the given set of kernels, while the second layer constructs a linear function in the space. This is equivalent to finding the optimal hyper plane in the corresponding feature space. The SVM algorithm can construct a variety of learning machines using different kernel functions. Fig 4 shows the linear separating hyper plane where support vector are encircled.

The application of Support vector machine (SVM) method to Text Classification has been propose by [22]. The SVM need both positive and negative training set which are uncommon for other classification methods. These positive and negative training set are needed for the SVM to seek for the decision surface that best separates the positive from the negative data in the n dimensional space, so called the hyper plane. The document representatives which are closest to the decision surface are called the support vector. SVM classifier method is outstanding from other with its effectiveness [19] to improve performance of text classification [24] combining the HMM and SVM where HMMs are used to as a feature extractor and then a new feature vector is normalized as the input of SVMs, so the trained SVMs can classify unknown texts successfully, also by combing with Bayes [23] use to reduce number of feature which as reducing number of dimension .SVM is more capable [25] to solve the multi-label class classification.

6) Artificial Neural Networks

An Artificial Neural Network (ANN) is an information processing paradigm that is inspired by the way the human brain processes information. A great deal of literature is available explaining the basic construction and similarities to biological neurons. The discussion here is limited to a basic introduction of several components involved in the ANN implementation. The network architecture or topology, comprising: number of nodes in hidden layers, network connections, initial weight assignments, and activation functions, plays a very important role in the performance of the ANN, and usually depends on the problem at hand. Figure 2 shows a simple ANN and its constituents. In most cases, setting the correct topology is a heuristic model selection. Whereas the number of input and output layer nodes is generally suggested by the dimensions of the input and

© 2015, IJARCSMS All Rights Reserved

the output spaces, determining the network complexity is yet again very important. Too many parameters lead to poor generalization (over fitting), and too few parameters result in inadequate learning (under fitting) [20].

Every ANN consists of at least one hidden layer in addition to the input and the output layers. The number of hidden units governs the expressive power of the net and thus the complexity of the decision boundary. For well-separated classes fewer units are required and for highly interspersed data more units are needed. The number of synaptic weights is based on the number of hidden units. It represents the degrees of freedom of the network. Hence, we should have fewer weights than the number of training points. As a rule of thumb, the number of hidden units is chosen as n/10, where n is the number of training points [20] [21]. But this may not always hold true and a better tuning might be required depending on the problem.

7. K-Nearest Neighbors

K-NN classifier is a case-based learning [26] algorithm that is based on a distance or similarity function for pairs of observations, such as the Euclidean distance or Cosine similarity measure's. This method is try for many application [27] Because of its effectiveness, non-parametric and easy to implementation properties, however the classification time is long and difficult to find optimal value of k .The best choice of k depends upon the data; generally, larger values of k reduce the effect of noise on the classification, but make boundaries between classes less distinct. A good k can be selected by various heuristic techniques .to overcome this drawback [28] modify traditional KNN with different K-values for different classes rather than fixed value for all classes Fang Lu have been try to improve performance of KNN by using WKNN [29].A major drawback of the similarity measure used in k-NN is that it uses all features in computing distances. In many document data sets, only smaller number of the total vocabulary may be useful in categorizing documents. A possible approach to overcome this problem is to learn weights for different features (or words in document data etc.). [29] Proposed the Weight Adjusted k-Nearest Neighbor (WAKNN) classification algorithm that is based on the k-NN classification paradigm. With the help of KNN can improve the performance of text classification [30] from training set and also accuracy can improve with combination of KNN [31] with another method.

8. Genetic Algorithms

In 1975, Holland introduced an optimization procedure that mimics the process observed in natural evolution called Genetic Algorithms – GAs (Holland 1975). A GA is a search process that is based on the laws of natural selection and genetics. As originally proposed, a simple GA usually consists of three processes Selection, Genetic Operation and Replacement. A typical GA cycle and its high-level description are shown in Figure 1.The population comprises a group of chromosomes that are the candidates for the solution. The fitness values of all chromosomes are evaluated using an objective function (performance criteria or a system's behavior) in a decoded form (phenotype). A particular group of parents is selected from the population to generate offspring by the defined genetic operations of crossover and mutation. The fitness of all offspring is then evaluated using the same criterion and the chromosomes in the current population are then replaced by their offspring, based on a certain replacement strategy. Such a GA cycle is repeated until a desired termination criterion is reached. If all goes well throughout this process of simulated evolution, the best chromosome in the final population can become a highly evolved and more superior solution to the problem.

9. Learning Vector Quantization (LVQ)

Learning Vector Quantization (LVQ) is a supervised version of vector quantization that can be used when we have labeled input data. This learning technique uses the class information to reposition the Voronoi vectors slightly, so as to improve the quality of the classifier decision regions. It is a two stage process. This is particularly useful for pattern classification problems. The first step is feature selection – the unsupervised identification of a reasonably small set of features in which the essential information content of the input data is concentrated. The second step is the classification where the feature domains are assigned to individual classes.

© 2015, IJARCSMS All Rights Reserved

C. Classifier Combination:

It has gone through parallel routes within pattern recognition and machine learning, and perhaps in other areas such as data fusion.

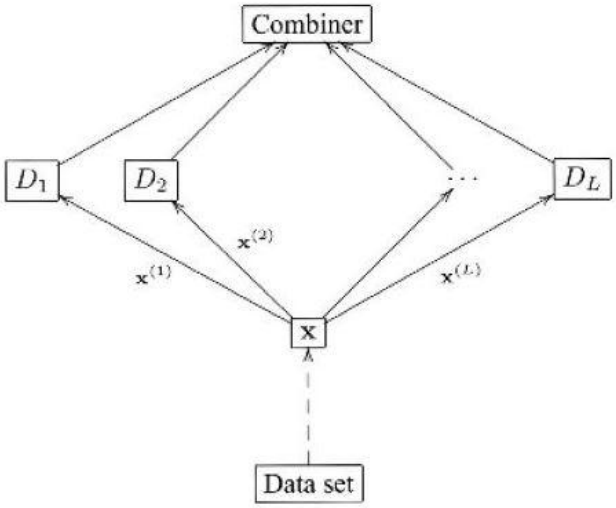

Figure 2: Approaches to build classifier ensembles.

It contains four levels

- Combiner level: In combiner level different combiners are used.
- Classifier level: In classifier level different classifiers are used.
- Feature level: In feature level different feature subsets are used.
- Data level: In data level different data subsets are used.

Fusion and selection are the two main strategies in combining classifiers. In classifier fusion, each ensemble member must know the whole feature space. In classifier selection, each ensemble member must know well a part of the feature space and be responsible for objects of this part. The fusion approach thus applies combiners like the average and majority vote whereas the selection approach selects one classifier to label the input x. There are combination schemes lying between the two strategies. This is taking the average of the outputs with coefficients that depend on the input x. Thus the local (with respect to x) competence of the classifiers is measured by the weights. So more than one classifier is responsible for x and the outputs of all the classifiers involved are fused.

1) Simple Combining Methods:

- *Uniform Voting*: In this combining schema, each classifier has the same weight. A classification of an unlabelled instance is performed according to the class that obtains the highest number of votes.
- *Distribution Summation*: This combining method was presented by Clark and Boswell (1991). The idea is to sum up the conditional probability vector obtained from each classifier. The selected class is chosen according to the highest value in the total vector.
- *Bayesian Combination*: This combining method was investigated by Buntine (1990). The idea is that the weight associated with each classifier is the posterior probability of the classifier given the training set.

© 2015, IJARCSMS All Rights Reserved

- *Dempster–Shafer*: The idea of using the Dempster–Shafer theory of evidence (Buchanan and Shortliffe, 1984) for combining models has been suggested by Shilen (1990; 1992). This method uses the notion of basic probability assignment defined for a certain class *ci* given the instance *x*:

- *Entropy Weighting*: The idea in this combining method is to give each classifier a weight that is inversely proportional to the entropy of its classification vector.

Meta-combining Methods

Meta-learning means learning from the classifiers produced by the inducers and from the classifications of these classifiers on training data. The following sections describe the most well-known meta-combining methods.

- *Stacking*: Stacking is a technique whose purpose is to achieve the highest generalization accuracy. By using a meta-learner, this method tries to induce which classifiers are reliable and which are not. Stacking is usually employed to combine models built by different inducers. The idea is to create a meta-dataset containing a tuple for each tuple in the original dataset. However, instead of using the original input attributes, it uses the predicted classification of the classifiers as the input attributes. The target attributes remains as in the original training set.

- *Arbiter Trees*: This approach builds an arbiter tree in a bottom-up fashion (Chan and Stolfo, 1993). Initially the training set is randomly partitioned into *k* disjoint subsets. The arbiter is induced from a pair of classifiers and recursively a new arbiter is induced from the output of two arbiters. Consequently for *k* classifiers, there are $\log_2(k)$ levels in the generated arbiter tree.

Combiner Trees: The way combiner trees are generated is very similar to arbiter trees. A combiner tree is trained bottom-up. However, a combiner, instead of an arbiter, is placed in each non-leaf node of a combiner tree (Chan and Stolfo, 1997). In the combiner strategy, the classifications of the learned base classifiers form the basis of the meta-learner's training set. A composition rule determines the content of training examples from which a combiner (meta-classifier) will be generated. In classifying an instance, the base classifiers first generate their classifications and based on the composition rule, a new instance is generated. The aim of this strategy is to combine the classifications from the base classifiers by learning the relationship between these classifications and the correct classification.

Two schemes of composition rule were proposed. The first one is the stacking schema. The second is like stacking with the addition of the instance input attributes. Chan and Stolfo (1995) showed that the stacking schema per se does not perform as well as the second schema. Although there is information loss due to data partitioning, combiner trees can sustain the accuracy level.

III. CONCLUSION

From the study of above classification techniques we have come up with a set of conclusions. The nearest-neighbour method is perhaps the simplest of all algorithms for predicting the class of a test example. An obvious disadvantage of the k-NN method is the time complexity of making predictions. Considerable amount of work has been done for recognizing plant species using k Nearest Neighbour technique. Classifying using PNN and SVM can be explored further by researchers, SVM being relatively a new machine learning tool. The most important advantage of PNN is that training is easy and instantaneous.

Additionally, neural networks are tolerant to noisy inputs. But in neural network it is difficult to understand the structure of the algorithm. SVM was found competitive with the best available machine learning algorithms in classifying high-dimensional data sets. In SVM computational complexity is reduced to quadratic optimization problem and it is easy to control complexity of decision rule and frequency of errors. Drawback of SVM is that it is difficult to determine optimal parameters when training data is not linearly separable. Also SVM is more complex to understand and implement. Another technique we studied is the genetic algorithm. Genetic algorithms are good at refining irrelevant and noisy features selected for classification. But representation of training/output data in genetic programming is complicated. Genetic algorithms provide a comprehensive

© 2015, IJARCSMS All Rights Reserved

search methodology for machine learning and optimization. PCA is used because it has advantage of reduced vector. The main limitation of PCA is that it does not consider class separability as it does not take into account the class label of the feature vector. Future direction for researchers can be to explore more robust techniques for recognition of plant leaves using a combination of classifying techniques such as SVM, k-NN, PNN.

ACKNOWLEDGEMENT

The authors would like to express their sincere thanks to Dr. K Mallikharjuna Babu, Principal, BMSCE, Dr. H S Gurupasad, Head of the Department of Computer Science whose support and guidance were invaluable. The work reported in this paper is supported by the college through the TECHNICAL EDUCATION QUALITY IMPROVEMENT PROGRAMME [TEQIP-II] of the MHRD, Government of India.

References

1. Jayamala K. Patil , Raj Kumar, "Advances In Image Processing For Detection of Plant Diseases", Journal of Advanced Bioinformatics Applications and Research, Vol. 2, Issue 2, pp. 135-141, June-2011.Applications and Research,Vol. 2,Issue 2, pp 135-141, June-2011.
2. Ji-Xiang Du, D.S.Huang, Xiao-Feng Wang, and Xiao Gu, "Computer-Aided Plant Species Identification (CAPSI) Based on Leaf Shape Matching Technique," Transactions of the Institute of Measurement and Control, vol. 28, no. 3, pp. 275-284, 2006 (SCI, EI).
3. D.N.D.Harini,D.Lalitha Bhaskari,"Identification of leaf Diseases in Tomato Plant based on wavelets", IEEE 2011 World Congress on Information and Communication Technologies, MIR Labs, Mumbai,India,978-1-4673-0125-1,pp 1398-1403
4. Stephen Gang Wu, Forrest Sheng Bao, Eric You Xu, Yu-Xuan Wang, Yi-Fan Chang and Qiao-Liang Xiang (2007) "A Leaf Recognition Algorithm for Plant Classification Using Probabilistic Neural Network", arXiv:0707.4289vl [cs.AI].
5. Arunpriya C., Antony Selvadoss Thanamani An Effective Tea Leaf Recognition Algorithm for Plant Classification Using Radial Basis Function Machine International OPEN ACCESS Journal of Modern Engineering Research (IJMER)
6. Anant Bhardwaj, Manpreet Kaur, Anupam Kumar. .Recognition of plants by Leaf Image using Moment Invariant and Texture Analysis International Journal of Innovation and Applied Studies, Vol. 3, No. 1. (May 2013), pp. 237-248
7. S. Mouine, I. Yahiaoui, A. Verroust-Blondet. Plant species recognition using spatial correlation between the leaf margin and the leaf salient points. ICIP 2013 (Special sessions: Image Processing and Pattern Recognition for Ecological Applications), September 2013.
8. A. Kadir, L. E. Nugroho, A. Susanto, P. Insap Santosa, "Leaf Classification Using Shape, Color, and Texture Features", International Journal of Computer Trends and Technology- July to Aug Issue 2011.
9. Leaf margins as sequences: A structural approach to leaf identification-Guillaume Cerutti,Laure Tougne,Didier Coquin,Antoine Vacavant Pattern Recognition Letters (Impact Factor: 1.06). 11/2014; 49:177–184. DOI: 10.1016/j.patrec.2014.07.016.
10. M. Z. Rashad, B.S.el-Desouky, Manal S .Khawasik, Plants Images Classification Based on Textural Features using Combined Classifier, International Journal of Computer Science & Information Technology (IJCSIT) Vol 3, No 4, August 2011.
11. Leaf Recognition Based on Leaf Tip and Leaf Base Using Centroid Contour Gradient.Fern, Bong Mei; Sulong, Ghazali Bin; Rahim, Mohd Shafry Mohd. Advanced Science Letters, Volume 20, Number 1, January 2014, pp. 209-212(4).
12. Automated Tool for Plant Leaf Classification Using Morphological Features Aamod Chemburkar, Anand Sartape, Ajinkya Gawade, Prasad Somawanshi International Journal of Engineering & Computer Science Volume/Issue: Vol. 3 - Issue 11 (November - 2014) e- ISSN: 2319-7242
13. Krishna Singh, Indra Gupta, Sangeeta Gupta, SVM-BDT PNN and Fourier Moment Technique for classification of Leaf, International Journal of Signal Processing, Image Processing and Pattern Recognition Vol. 3, No. 4, December, 2010.
14. Top 10 algorithms in data mining Xindong Wu, Vipin Kumar, J. Ross Quinlan, Joydeep Ghosh, Qiang Yang, Hiroshi Motoda, Geoffrey J. McLachlan, Angus Ng, Bing Liu, Philip S. Yu, Zhi-Hua Zhou, Michael Steinbach, David J. Hand, Dan Steinberg © Springer-Verlag London Limited 2007
15. Introduction to data mining – Pearson
16. X.-F. Wang, J.-X. Du, and G.-J. Zhang, "Recognition of leaf images based on shape features using a hypersphere classifier," in Proceedings of International Conference on Intelligent Computing 2005, ser. LNCS 3644. Springer, 2005.
17. Krishna Singh, Dr. Indra Gupta and Dr Sangeeta Gupta, Retrieval and classification of leaf shape by support vector machine using binary decision tree, probabilistic neural network and generic Fourier moment technique: a comparative study, IADIS International Conferences Computer Graphics, Visualization, Computer Vision and Image Processing 2010
18. J.-X. Du, X.-F. Wang, and G.-J. Zhang, Leaf shape based plant species recognition. Applied Mathematics and Computation, vol. 185, 2007
19. S.B. Cho and J.H. Kim, "Multiple Network Fusion Using Fuzzy Logic," IEEE Trans. Neural Networks, vol. 6, no. 2, pp. 497-501, 1995.
20. Duda., R.O., Hart, P.E., Stork, D.G. (2001), Pattern Classification , Second Edition, Wiley-Interscience Publications
21. Lawrence, S., Giles, C.L., Tsoi, A.C.(1997), "Lessons in Neural Network Training: Overfitting May be Harder than Expected", Proceedings of the Fourth National Conference on Artificial Intelligence, AAAI-97, pp 540-545.
22. K.S. Woods, K. Bowyer, and W.P. Kergelmeyer, "Combination of Multiple Classifiers Using Local Accuracy Estimates," Proc. CVPR '96, pp. 391-396, 1996.
23. J. Kittler, A. Hojjatoleslami, and T. Windeatt, "Weighting Factors in Multiple Expert Fusion," Proc. British Machine Vision Conf., Colchester, England, pp. 41-50, 1997.
24. J. Kittler, A. Hojjatoleslami, and T. Windeatt, "Strategies for Combining Classifiers Employing Shared and Distinct Pattern Representations," Pattern Recognition Letters, to appear.

© 2015, IJARCSMS All Rights Reserved

25. J. Kittler, "Improving Recognition Rates by Classifier Combination: A Theoretical Framework," Frontiers of Handwriting Recognition, A.G. Downton and S. Impedovo, eds. World Scientific,pp. 231-247, 1997.
26. Gongde Guo, Hui Wang, David Bell, Yaxin Bi and Kieran Greer, "KNN Model-Based Approach in Classification", Proc. ODBASE pp- 986 – 996, 2003
27. Eiji Aramaki and Kengo Miyo, "Patient status classification by using rule based sentence extraction and bm25-knn based classifier", Proc. of i2b2 AMIA workshop, 2006.
28. Muhammed Miah, "Improved k-NN Algorithm for Text Classification", Department of Computer Science and Engineering University of Texas at Arlington, TX, USA.
29. Fang Lu Qingyuan Bai, "A Refined Weighted K-Nearest Neighbours Algorithm for Text Categorization", IEEE 2010.
30. Kwangcheol Shin, Ajith Abraham, and Sang Yong Han, "Improving kNN Text Categorization by Removing Outliers from Training Set", Springer-Verlag Berlin Heidelberg 2006.
31. Methods Ali Danesh Behzad Moshiri "Improve text classification accuracy based on classifier fusion methods". 10th International Conference on Information Fusion, 1-6 2007.
32. Goldberg, D.E. (1989), Genetic Algorithm in Search, Optimization and Machine Learning, Addison Wesley Publishing Company.Holland, J. (1975). Adaptation In Natural and Artificial Systems. The University of Michigan Press, Ann Arbor.

AUTHOR(S) PROFILE

Subhankar Ghosh, is a final year UG student pursuing B.E. in Computer Science and Engineering in BMS College of Engineering Bangalore.

Harish Kumar V, is a final year UG student pursuing B.E. in Computer Science and Engineering in BMS College of Engineering Bangalore.

Pradeep Kumar, is a final year UG student pursuing B.E. in Computer Science and Engineering in BMS College of Engineering Bangalore.

Devaraj, is a final year UG student pursuing B.E. in Computer Science and Engineering in BMS College of Engineering Bangalore.

Jyothi S. Nayak, is an Associate Professor in the Department of Computer Science and Engineering in BMS College of Engineering Bangalore. Her research interests include Image Processing, Pattern Classification and Machine Learning.

© 2015, IJARCSMS All Rights Reserved