Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 24 (2013) 84 – 96

Procedia Computer Science

17th Asia Pacific Symposium on Intelligent and Evolutionary Systems, IES2013

One-Shot Classification of 2-D Leaf Shapes using Distributed Hierarchical Graph Neuron (DHGN) Scheme with k-NN Classifier

Anang Hudaya Muhamad Amin[a,*], Asad I. Khan[b]

[a]*Faculty of Information Science and Technology (FIST), Multimedia University, Jalan Ayer Keroh Lama, 75450 Melaka, Malaysia*
[b]*Clayton School of Information Technology, Monash University, Clayton, 3168 VIC, Australia*

Abstract

This article presents a scalable approach for classifying plant leaves using the 2-dimensional shape feature. The proposed approach integrates a distributed recognition scheme called Distributed Hierarchical Graph Neuron (DHGN) for pattern recognition and k-nearest neighbor (k-NN) for pattern classification. With increasing amount of leaves data that can be captured using existing image gathering and processing technology, the ability for any particular classification scheme to produce high recall accuracy while adapting to large-scale dataset and data features is very important. The approach presented in this paper implements a one-shot learning mechanism within a distributed processing infrastructure, enabling large-scale data to be classified efficiently. The experimental results obtained through a series of classification tests indicate that the proposed scheme is able to produce high recall accuracy and large number of perfect recalls for a given plant leaves dataset. Furthermore, the results also indicate that the recognition procedure within the DHGN distributed scheme incurs low computational complexity and minimum processing time.

Keywords: Distributed pattern recognition; Graph Neuron (GN); k-nearest neighbor (k-NN); One-shot learning; Plant leaves classification ;

1. Introduction

Plants are an important element in life. The classification of plant leaves is a crucial process in some agricultural-based industries such as tea and cotton [1]. Leaves classification is also important in the area of botanical research in

* Corresponding author. Tel.: +60-6-2523883; fax: +60-62318840.
E-mail addresses: anang.amin@mmu.edu.my (A.H. Muhamad Amin), Asad.Khan@monash.edu (A.I. Khan).

doi:10.1016/j.procs.2013.10.030

identifying classes and families of plants. Plant leaf provides a distinguishable feature to identify species and groups of plants. The plants classification schemes have been widely developed as early as in the 18th century, by Carolus Linnaeus, a Swedish botanist [2].

The shape of leaf is a common feature used to classify different species of plants. A number of related works have been carried out in analyzing leaf shapes for plant classification. This paper presents conceptual design and implementation of one-shot classification scheme to classify 2-dimensional leaf shapes using a combined Distributed Hierarchical Graph Neuron (DHGN) recognition and k-NN classification approach. The proposed scheme considers the scalability aspect of classification procedures by incorporating one-shot learning capability for large-scale datasets. In this paper, we demonstrate the high recall accuracy over large number of plant species that can be achieved by implementing one-shot learning classifier within a distributed processing environment.

The rest of this paper is organized as follows. Section 2 discusses some of the related works in the area of plant leaves classification and other classification schemes involving shape analysis. Section 3 presents our work on the integrated k-NN and DHGN scheme for plant leaves classification. The proposed scheme combines the scalable recognition approach of DHGN and lightweight learning classification module of k-NN for large-scale data analysis. Section 4 will describe our classification analysis, based upon a selected leaves dataset of 100 different species. Experimental results obtained from the analysis are presented in Section 5. Finally, Section 6 concludes the paper.

2. Related Works

Shape recognition and classification has been mainly used to identify objects. Although it plays an important role in image processing and computer vision applications, the ability to recognize variations of shapes is still an open challenge. Much of the recent works in this area of research have been focusing on the ability to achieve high recall accuracy for shape classification. Some of the works include the shape classification using ad hoc boundary points by McNeill and Vijayakumar [3], local and global features classification by Lim and Galoogahi [4], and shape classification based on base and strand structures of shape by Temlyakov et al [5]. Apart from these, Ling and Jacobs [6] proposed the use of inner distances as a descriptor for classification of 2D shapes, while Bai et al [7] integrates contour and skeleton of shapes in their classification schemes.

One of the important aspects to consider in shape classification is the large-scale variations of shapes that correspond to large number of objects. The scalability aspect of classification has yet to be considered by most of the ongoing research works. Some of the related works that focus on scalability perspective include the research done by Tatsuma et al [8] on classification of ten thousands 3-dimensional shape models using Toyohashi Shape Benchmark (TSB). In this paper, we will consider the scalability aspect of large-scale classification, by incorporating a distributed processing mechanism in recognizing patterns of data. Our proposed scheme implements a divide-and-distribute approach for pattern recognition and classification [9].

A number of works have been carried out to improve the recognition accuracy of the classification schemes involving leaf shapes. These include the works carried out by Wu et al [10] in leaf classification using probabilistic neural network and Zhang and Feng in the leaf classification using neighborhood rough set [11]. Apart from these, Beghin et al [12] also introduces shape and texture based plant leaf classification using contour signatures and sobel operator for shape and texture analysis respectively, while Tzionas et al [13] implements a fuzzy surface selection technique on morphological features in plant leaf classification.

The techniques mentioned previously have been looking into increasing the recall accuracy of the classifiers. However, with an increase in the number of features for consideration, the recall accuracy of any particular scheme may be affected, due to the curse of dimensionality effect. A scalable classification scheme should be able to adapt to an increasing number of features obtained from existing sophisticated data acquisition methods and techniques. Existing classifiers tend to improve its scalability over large feature sets by implementing feature reduction approaches [11].

This paper presents a different perspective on classification scheme. We intend to focus on the scalability aspect of the classifier by implementing a divide-and-distribute approach in recognizing feature patterns. This work demonstrates the functions of DHGN, as a pattern recognizer based on the artificial neural network scheme and graph-matching technique. DHGN is implemented in a purely distributed manner and performed matching-based learning function that is employed within the composition of inter-linked neurons.

In addition, a hierarchical structure of DHGN enables deep learning to be performed from initial input domain towards the recognition outputs. Apart from the scalable approach of DHGN, this work proposed a combination of k-NN lazy classifier, to enable final classification to be done in the simplest approach. Our proposed scheme is composed of two integrated stages, namely feature recognition, and classification. Further discussion on this approach will be described in the next section.

3. Integrated DHGN and k-NN Scheme

The Distributed Hierarchical Graph Neuron (DHGN) was introduced by Khan and Muhamad Amin [14]. It is derived from the extension of graph-based neural network scheme known as Graph Neuron (GN) [15]. GN implements a recognition based upon the neuron-value adjacency comparison approach. The strength of the firing within each neuron depends upon the matching function between input element and stored elements within the particular neuron, which is known as bias entries. These bias entries are the *(value, position)* pair compositions that represent pattern elements. These entries were obtained from the synaptic responses between adjacent neurons (the input values of adjacent neurons). The neuron firing mechanism in GN implementations is different from other neuron-based intelligent techniques in the sense that its synaptic plasticity is independent of weight adjustment mechanism based upon the input strength. Rather, the value-matching function between adjacent neurons is used in determining the output of each neuron. Fig. 1 and Fig. 2 show the neuron firing mechanisms in DHGN network with two possible outcomes: matched and new bias entry. Detail explanations on the fundamental recognition procedure of GN can be referred in the works of Nasution and Khan [16] on Hierarchical Graph Neuron (HGN).

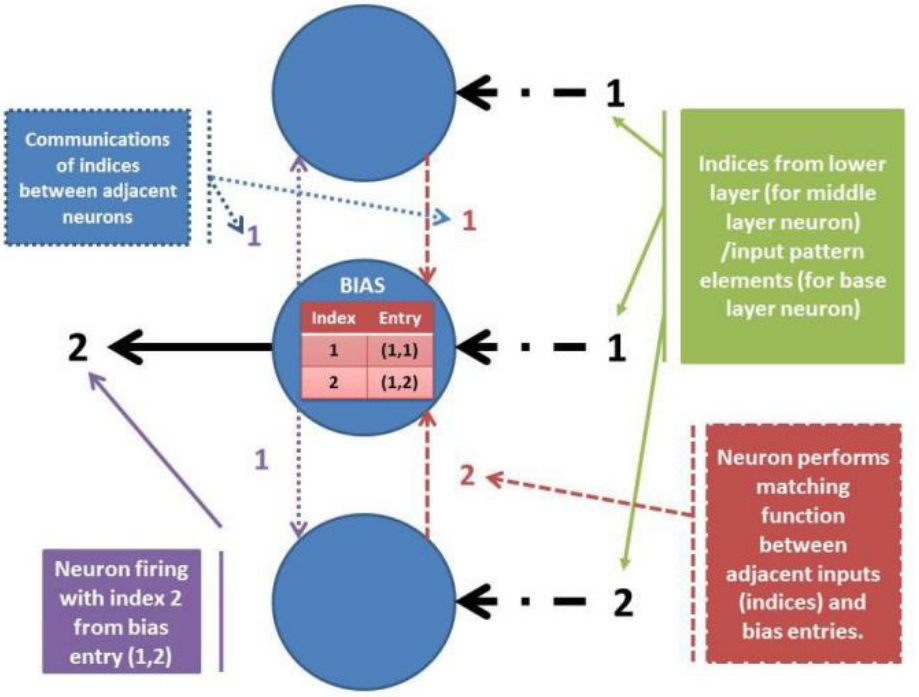

Fig. 1. Neuron firing with matched bias entry.

DHGN algorithmic structure basically follows the deep learning principle as described by Bengio [17], in which its network composition is laid out as a series of hierarchical neurons. Each neuron holds part of information

obtained from the input patterns. Thus, enabling the whole network structure to store the information related to each pattern. Fig. 3 shows how information related to each pattern is stored and represented in one DHGN subnet.

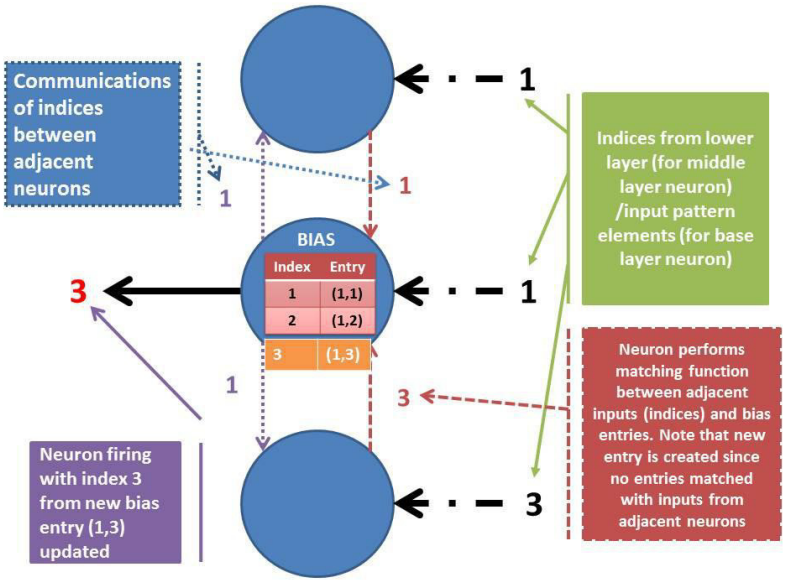

Fig. 2. Neuron firing with new bias entry (new pattern discovery).

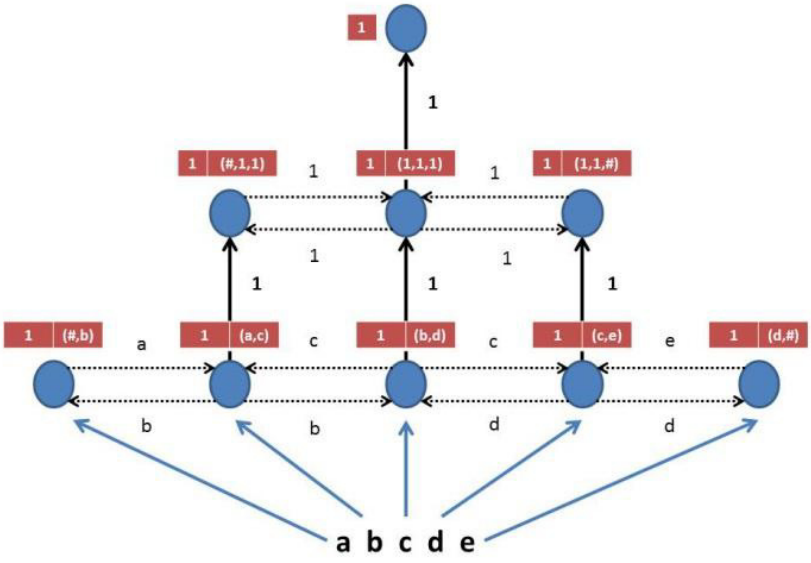

Fig. 3. Representation and storage of pattern *"abcde"* in a subnet within DHGN network.

3.1. DHGN recognition procedure

DHGN extends the capability of HGN for pattern recognition involving large-scale data [9]. Fig. 4 shows the network architecture for DHGN. Note that each DHGN subnet (a HGN composition) performs recognition on each subpattern independently from other subnets. On the other hand, the SI module (Storage-Input) node basically acts as a controller and handles input/output functions for the delivery of input pattern and collection of recognition results from all subnets.

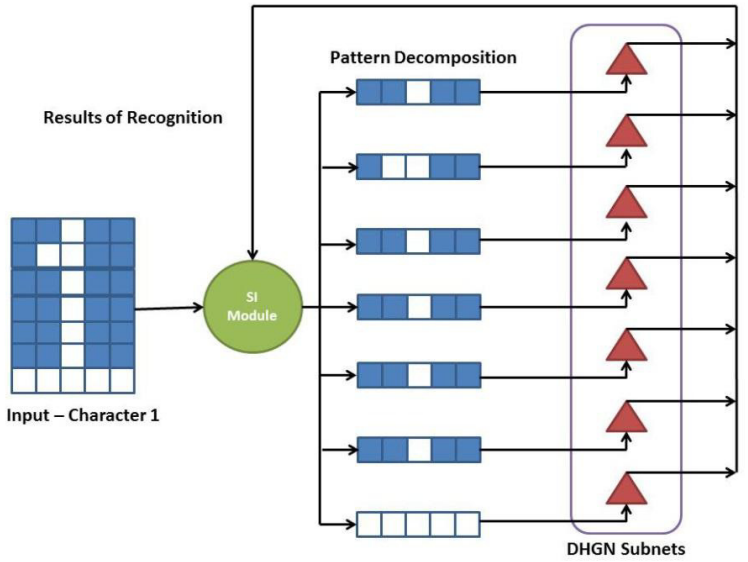

Fig. 4. Pattern recognition processes using DHGN algorithm where a bitmap of character *'1'* is mapped as a series of subpatterns over hierarchically formed GN sub-networks (adopted from [18]).

DHGN network structure in its actual formation, replicates the multiple-HGN structure with two distinctive features:

- The recognition procedure involves forward propagation approach without any backward propagation (from top neuron to the bottom-level neurons) as described in [18].
- Collective recognition in the form of voting procedure is conducted at SI module level.

The composition of neurons in each DHGN network, n_c can be represented in the form of the following equation:

$$n_c = n_s \times d; \quad d = \left(\frac{n_e + 1}{2} \right)^2 \tag{1}$$

Where n_s, n_e, and d represent the number of DHGN subnet, subpattern size, and subnet size respectively. DHGN implements a divide-and-distribute approach for scalable pattern recognition. Each input pattern with length l_p will be divided and distributed over specified numbers of DHGN subnets, n_s with equal subpattern size, n_e (refer to Eq. 1). This distribution of pattern into subpatterns is represented by Eq. 2 :

$$l_p = n_s \times n_e; \quad n_e = 2 \times \sqrt{d} + 1 \tag{2}$$

DHGN pattern representation follows the representation of patterns in graph-matching based algorithms. Each neuron in DHGN composition holds a (value, position) pair information of elements that constitutes the pattern. In correspondence towards graph-based structure, each neuron acts as a vertex that holds pattern element information (in the form of value or identification (ID)) while the adjacency communication between two or more neurons is represented by the edge of a graph.

The recognition procedure in DHGN involves finding matched indices between input pattern and stored patterns within the network composition. Each neuron compares input pattern values with the information obtained from its adjacent neurons. Adjacency information for each neuron is represented using the *(left, right)* formation. Each activated neuron therefore records the information retrieved from its adjacent left or right nodes. In the GN-based approach terminology, this adjacency information is known as bias entry where each GN maintains an array of such entries. The entries for the entire stored pattern are collectively stored in the bias arrays. Each neuron would hold a single bias array containing all the bias entries obtained in the recognition processes. Each neurons bias array only stores the unique adjacency information derived from the input patterns.

The process of pattern storage for N bias array sizes can be symbolically represented as follows:

$$E^B = \{\langle x, y\rangle; x \in N, y \in N\} \tag{3}$$

Where E^B, represents the sets of two-element ordered pair respectively, while x and y are the values within each bias entry. E^B can be also represented as one- or three-element ordered pair, depending upon the location of neuron within the subnet hierarchical structure.

Searching of matched bias entry with the adjacency information obtained for a given input pattern within each neuron involves a linear search through the bias array composition. Each entry within the bias array is unique. Therefore, the following equation is used to estimate the expected number of comparisons for each input entry C_i, given n_b number of bias entries in the array and r number of occurrences for each entry:

$$C_i = \frac{n_b + 1}{r + 1} \tag{4}$$

In essence, DHGN employs one-shot learning for pattern recognition, by implementing a graph-matching approach on its pattern representation. Within DHGN network composition, each neuron performs a forward propagation of index values obtained from the matching process as shown in Fig. 3, from the base layer neurons towards the top neuron within each subnet. The top layer neurons will then pass the final indices to the SI Module node for pattern voting procedure (please refer to the work of Muhamad Amin and Khan [18] on the voting procedure within DHGN network). The procedures only involved a one-pass cycle for each input pattern, without any iterations involving value alteration as to obtain a recognition output.

3.2. Multi-level DHGN for feature recognition

With the ability to divide the recognition operations into clusters of recognizers, DHGN allows large patterns to be classified within a single execution. Furthermore, with the advent of one-shot learning method being deployed, DHGN provides a mechanism for fast learning involving large datasets. In this paper, we intend to observe the ability for DHGN to perform seamless feature attributes reduction and recognition. It is assumed that by implementing DHGN approach in multi-level recognition perspective would enable large number of feature attributes to be reduced while seamlessly performing its recognition procedures.

The procedure for multi-level DHGN scheme involves the arrangement of DHGN networks within the multi-level architecture. Fig. 5 shows the formation of this multi-level DHGN scheme, in relations to the number of input feature attributes or vectors presented to the network.

Note that the levels within the multi-level composition can be increased or decreased depending upon the dataset requirements. For recognition at the lowest level l_0, given n_f number of feature vectors f, a set of feature vectors, $f_{sub} \in f$ will be grouped into a single DHGN subnet, in such a way that the distribution follows the size of

subpattern, n_e that each subnet is able to accept. Hence, the number of feature vector subsets n_{fsub} that will be created can be calculated as follows:

$$n_{f_{sub}} = \frac{n_f}{n_e}; \quad n_e \geq 1 \tag{5}$$

Upon completion of the recognition process at the lowest level, the outputs, in the form of pattern indices from each DHGN subnet will then be used as input feature vectors to the higher level subnets. This procedure continues until it reaches the desired number of final input feature vectors. By implementing this multi-level recognition, the feature space for given classification problem will be reduced significantly, without compromising the complexity of feature reduction process. The embedment of such feature reduction procedure in recognition process will also enable fast and scalable classification to be performed over datasets involving large number of feature vectors.

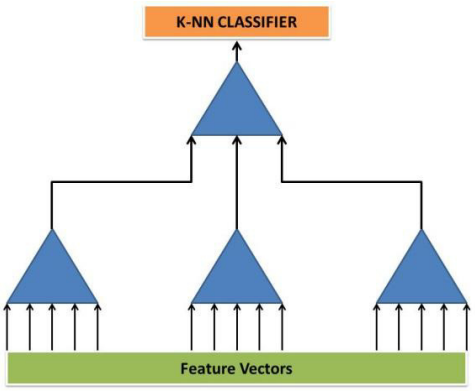

Fig. 5. The proposed two-level DHGN architecture for feature vector reduction and recognition.

3.3. k-NN classifier for final classification

In this proposed multi-level DHGN scheme, we proposed a lazy classifier for final classification of plant leaves data. This is an alternative for voting function in normal DHGN implementation. The limitation of voting function is that it only allows discrete evaluation of DHGN recognition outputs to be performed, without analyzing the close proximities of data values using any continuous distance function. By integrating lazy classifier such as k-nearest neighbor (k-NN), classification on the selected dataset can be performed by taking into account distances between data values of each class. Hence, allowing more effective and accurate classification process to be conducted.

For the plant leaves classification, k-NN classifier has been used to classify the plant species using a set of reduced leaf shape feature vectors, as results from the DHGN multi-level recognition scheme. For the purpose of comparative analysis, we have used two different distance functions, namely Euclidean and Canberra functions.

The Euclidean distance, d^E takes an ordinary distance of two values, x and y according to the following:

$$d^E = |x - y| = \sqrt{\sum_{i=1}^{n} |x_i - y_i|^2} \tag{6}$$

The Euclidean distance can suitably be used for one-dimensional distance calculation. However, the problem space defined in this work, is of multi-dimensional, in which for a given data point, multiple attributes need to be considered. Hence, another form of distance calculation is required. Canberra distance is a form of multi-dimensional distance function that can be used to represent n-dimensional feature vector space for this leaf classification problem. Given two vectors, x and y, the Canberra distance d^C can be calculated as follows:

$$d^C(x,y) = |x-y| = \sum_{i=1}^{n} \frac{|x_i - y_i|}{|x| + |y|} \tag{7}$$

Apart from the selection of distance function, the following k-NN parameter values, as shown in Table 1 have been used for leaves classification based on the output feature vectors analysis using multi-level DHGN scheme.

Table 1. Parameters for k-NN in classifying 100 plant leaves species based on shape feature.

Parameters:	k	Train/Test Split Ratio	Distance Function
Values/Types:	1	0.3, 0.5, 0.7	Euclidean, Canberra

Note that we have chosen the value of $k = 1$, as the best representation for the acquired problem space in the k-NN classification. This value is selected as to find a single nearest class for a particular pattern. In addition, we have also considered multiple splitting ratios of train-to-test data, to observe the accuracy of the classifier towards changes in train-to-test data ratios.

3.4. Plant leaves dataset

The plant leaves dataset used in this paper is taken from the work of Mallah et al [19], on probabilistic integration of shape, texture, and margin features for plant leaves classification. This dataset is publicly available from UCI Machine Learning Repository [20]. It contains leaves data of 100 plant species. Each plant species contains 16 data instances, accumulating to 1600 shapes. Each leaf shape feature for this dataset consists of 64 feature vectors derived from the normalized contour signatures obtained from differential calculation of foreground and background pixel values of the shape images. Fig. 6 shows some of these shape images used in this leaves classification technique.

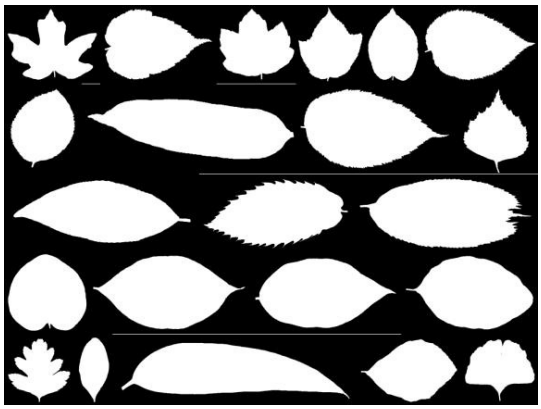

Fig. 6. Samples of leaf shape images from different plant species used in the leaves classification process.

Note that the dataset used also considers the differences between lobed and serrated leaf shapes. The plant species in which the leaves have been used are ranged from some of the common species, including species from genus Eucalyptus (*Eucalyptus Urnigera* and *Eucalyptus Neglecta*).

4. Recognition and Classification Process Configuration

For the plant leaves classification scheme, we have adopted both single-level and multi-level DHGN configuration, namely ML-DHGN I and ML-DHGN II respectively. Table 2 details out the parameters used to set up the DHGN network for both cases.

Table 2. Multi-level DHGN configurations for plant leaves classification.

Configuration Type	No. of Subnets	No. of Neurons
ML-DHGN I	13	117
ML-DHGN II	16	144

For ML-DHGN I configuration, we have deployed 13 DHGN subnets within a single level. The output vectors derived from the DHGN pattern recognition scheme are then be used as inputs to the k-NN classifier. The splitting of training and test data only being performed at the classification level, i.e. within k-NN classification scheme. We have used the k-NN classification module in RapidMiner analytical tool for this purpose. Fig. 7 shows a snapshot of the data used in this DHGN pattern recognition scheme.

Acer Campestre	0.003906	0.003906	0.027344	0.033203	0.007812	0.017578	0.023438	0.005859	0	0.015625
Acer Campestre	0.005859	0.013672	0.027344	0.025391	0.013672	0.029297	0.019531	0	0.001953	0.021484
Acer Campestre	0.011719	0.001953	0.027344	0.044922	0.017578	0.042969	0.023438	0	0.003906	0.019531
Acer Campestre	0.013672	0.011719	0.037109	0.017578	0.011719	0.087891	0.023438	0	0	0.027344
Acer Campestre	0.007812	0.009766	0.027344	0.025391	0.001953	0.005859	0.015625	0	0.005859	0.017578
Acer Campestre	0.015625	0.003906	0.015625	0.046875	0.013672	0.064453	0.017578	0.003906	0	0.033203
Acer Campestre	0.007812	0.03125	0.048828	0.015625	0.005859	0.013672	0.03125	0	0.009766	0.017578
Acer Campestre	0.023438	0.013672	0.039062	0.015625	0.025391	0.025391	0.023438	0	0.009766	0.021484
Acer Campestre	0.007812	0.013672	0.029297	0.021484	0.009766	0.03125	0.015625	0	0.001953	0.042969
Acer Campestre	0.019531	0.009766	0.023438	0.03125	0.021484	0.054688	0.025391	0.003906	0	0.021484
Acer Campestre	0.005859	0.021484	0.035156	0.015625	0.005859	0.005859	0.025391	0	0.011719	0.021484
Acer Campestre	0.023438	0.027344	0.007812	0.017578	0.005859	0.011719	0.021484	0	0	0.017578
Acer Campestre	0.017578	0.041016	0.017578	0.005859	0.003906	0.027344	0.017578	0.003906	0	0.017578
Acer Campestre	0.015625	0.009766	0.025391	0.027344	0.001953	0.001953	0.011719	0	0.001953	0.013672
Acer Campestre	0.009766	0.021484	0.019531	0.027344	0.003906	0.025391	0.023438	0	0.001953	0.023438
Acer Campestre	0.017578	0.011719	0.023438	0.019531	0.003906	0.011719	0.015625	0	0	0.03125
Acer Capillipes	0	0	0.013672	0.015625	0.048828	0	0.033203	0	0.003906	0.013672
Acer Capillipes	0.001953	0.001953	0.025391	0.017578	0.029297	0.005859	0.041016	0	0.007812	0.011719
Acer Capillipes	0	0	0.021484	0.037109	0.029297	0	0.041016	0.003906	0.007812	0.007812
Acer Capillipes	0.001953	0	0.009766	0.019531	0.03125	0.001953	0.029297	0	0.007812	0.017578
Acer Capillipes	0	0	0.013672	0.015625	0.035156	0	0.023438	0	0.005859	0.017578
Acer Capillipes	0.001953	0	0.011719	0.029297	0.033203	0	0.017578	0	0.005859	0.009766
Acer Capillipes	0.001953	0	0.017578	0.001953	0.054688	0.001953	0.019531	0	0.009766	0.009766
Acer Capillipes	0.003906	0.001953	0.03125	0.011719	0.037109	0.001953	0.042969	0	0.009766	0.013672
Acer Capillipes	0.001953	0	0.017578	0.013672	0.027344	0	0.009766	0	0.007812	0.013672
Acer Capillipes	0.001953	0	0.011719	0.013672	0.033203	0	0.037109	0	0.007812	0.021484
Acer Capillipes	0	0	0.021484	0.015625	0.041016	0	0.048828	0	0.009766	0.009766
Acer Capillipes	0.005859	0.001953	0.005859	0.019531	0.041016	0	0.027344	0	0.005859	0.017578
Acer Capillipes	0	0.001953	0.015625	0.013672	0.029297	0.001953	0.052734	0	0.007812	0.025391
Acer Capillipes	0	0	0.019531	0.015625	0.041016	0.003906	0.021484	0.001953	0.013672	0.005859
Acer Capillipes	0	0.001953	0.015625	0.011719	0.033203	0.003906	0.021484	0	0.009766	0.025391
Acer Capillipes	0.001953	0.001953	0.013672	0.011719	0.042969	0	0.021484	0	0.009766	0.011719

Fig. 7. A snapshot of leaf shape data in the form of normalized contour signature values for a range of plant species.

Differently for ML-DHGN II, DHGN network composition is divided into two different subnet levels as shown in Fig. 5. Output vectors produced by subnets at level I will be fed into subnets at level 2. The output vectors from subnets at level II will then be used by the k-NN classifier. In this regards, the number of input vectors for k-NN classifier in ML-DHGN II is significantly less, in comparison with ML-DHGN I.

The reason behind implementing these two different configurations is to analyze the level of accuracy for classification by reducing the input feature vectors within the proposed classification scheme. In analyzing the

efficiency and effectiveness of the proposed scheme, two performance measurements have been taken into consideration. These include the recall accuracy of the classification scheme, as well as the recognition time for the distributed one-shot pattern memorization using DHGN. The following section will discuss the experimental results using these two performance parameters in details.

5. Experimental Results

The classification of leaf shapes using the integrated multi-level DHGN and k-NN involves one-shot recognition of shape feature vectors using a composition of DHGN network and classification of the output vectors using k-NN classifier. Each species class in the given dataset consists of 16 data instances with each instance correspond to 64 feature vectors.

5.1. Recall accuracy

In regards to the recall accuracy of the proposed scheme, the results of recognition and classification of leaf shapes from 100 plant species are tabulated in Table 3. The recall accuracy of the proposed scheme is based upon the rate of true positive recalls for each data instances used in the experiment.

Table 3. Comparative evaluation of average recall accuracy of k-NN ML-DHGN I and II using Euclidean and Canberra distance functions for each plant species class with different train/test splitting ratio values.

Scheme	Distance Function	Splitting Ratio	Average Recall Accuracy (%)
ML-DHGN I	Euclidean	0.3	24.64
		0.5	42.12
		0.7	42.50
	Canberra	0.3	36.52
		0.5	28.62
		0.7	30.42
ML-DHGN II	Euclidean	0.3	60.71
		0.5	67.38
		0.7	71.04
	Canberra	0.3	61.79
		0.5	68.38
		0.7	71.46

The results of the experiments indicate that the classification accuracy of k-NN classifier is higher in ML-DHGN II than ML-DHGN I. We can deduce that this is caused by the reduction of feature vector space through multi-level DHGN recognition scheme. The ability for k-NN to classify the leaves data more effectively could be achieved using this approach. In addition, the number of perfect recalls for individual plant classes is considerably high; for classification of leaf shapes involving 0.7 splitting ratio with ML-DHGN type II, about 38% of the 100 plant classes indicate perfect recall by the k-NN classifier.

Apart from comparison between different multi-level configurations of DHGN, we also observed a comparison between k-NN and the proposed integrated k-NN ML-DHGN scheme in regards to their average recall accuracy. Table 4 shows this comparison on 1600 leaves data in the form of 2-dimensional shape images for 100 different plant species.

The combined k-NN and ML-DHGN scheme basically outperforms k-NN classifier for all train-to-test data configurations using multiple splitting ratios. The results also indicate the ability of the proposed scheme to perform classification with higher recall accuracy for large-scale datasets with minimum increase in the computational complexity.

Table 4. Comparison on average recall accuracy between k-NN and integrated k-NN and ML-DHGN II classifiers.

Train/Test Splitting Ratio	Average Recall Accuracy (%)	
	k-NN	Integrated k-NN & ML-DHGN II
0.3	50.89	61.79
0.5	57.00	68.38
0.7	65.62	71.46

5.2. Recognition time evaluation

Recognition time was measured during the DHGN recognition process in all subnets within the network composition. Fig. 8 shows the time taken to recognize 1600 leaf shape subpatterns on each DHGN subnet. Each subpattern is part of the total composition of 64 shape feature vectors. The recognition time was taken from the point where the pattern element is distributed across the DHGN subnet, until the output index is received from the top neuron within each subnet.

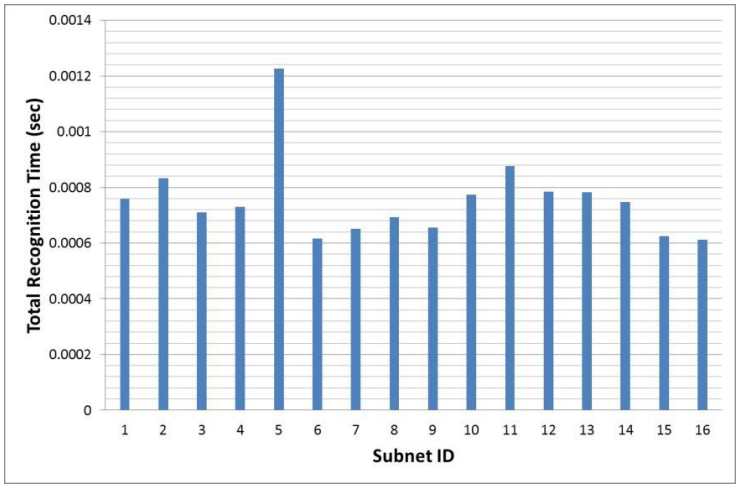

Fig. 8. Recognition time for each DHGN subnet involving 1600 subpatterns of five leaf shape vectors.

On average, the recognition time for the entire leaves dataset would only require 1 millisecond of execution. The longest recognition time experienced by the DHGN network composition is about 1.2 milliseconds, which may due to the delay by the CPU processing capacity. The entire simulation was performed using a virtual machine (VM) equipped with Ubuntu Linux operating system, on *Intel i5 2.5 GHz* CPU.

6. Conclusions

In this work, a classification scheme for large-scale data analysis was proposed to classify 100 plant species using 2-dimensional leaves shapes. The ability to perform recognition procedure in a distributed manner was considered. The proposed classification approach entails a two-step procedure. Firstly, the recognition of 2-d shapes of leaves using feature vectors was done using a divide-and-distribute approach within a multi-level DHGN (ML-DHGN) scheme. The process involves seamless recognition and reduction of feature vectors. The efficient recognition using adjacency comparison approach within DHGN was adopted. The outputs of this DHGN recognition process, in the form of indices generated by each DHGN subnet within the network composition are then to be used in the second

step of the classification procedure, i.e. the plant species classification. In this step, a k-NN lazy classifier is used to classify the leaf shape data according to their respective classes (plant species).

The results show that the proposed scheme is able to produce high recall accuracy for large-scale data classification. With 0.7 ratio of train/test splitting, our approach was able to perfectly classify 38 out of 100 plant species, with an average recall accuracy of 71.5%. In addition, the experimental results indicate that an increase in the number of subnet levels from one to two also increases the average recall accuracy of the proposed classifier. The increase in the number of levels used would consequently reduce the number of feature vectors that will be inserted into the final classifier, such as k-NN. By comparing our proposed integrated k-NN and multi-level DHGN scheme with k-NN classifier, the ability for the scheme to outperform k-NN classification, in terms of the recall accuracy has also being observed.

Apart from recall accuracy, the proposed ML-DHGN scheme also perform fast recognition in which each DHGN subnet takes on average 1 millisecond for recognition process involving 1600 leaf shape subpatterns. This could only be achieved by using one-shot learning mechanism being adopted by DHGN recognition approach.

The classification of plant leaves can be considered as a large-scale data analysis problem. The works that have been carried out in this paper demonstrate the classification of large-scale data with large feature vector space by minimizing the increase in the computational complexity of the classification scheme through seamless recognition and feature space reduction. The proposed approach is relatively simple and able to scale up with the increase in the dimension and size of data. Further works will be carried out in determining the effects of adding multiple leaf features, including texture and margin for plant species classification.

Acknowledgements

We would like to acknowledge the Department of Higher Education, Ministry of Education (MOE), Malaysia for the funding used in this research project under the Fundamental Research Grant Scheme (FRGS).

References

1. F. Mokhtarian, S. Abbasi, Matching shapes with self-intersections:application to leaf classification, Image Processing, IEEE Transactions on 13 (5) (2004) 653–661. doi:10.1109/TIP.2004.826126.
2. C. v. Linne, L. Salvius, Caroli Linnaei...Systema naturae per regna tria naturae, Vol. v.2, Holmiae :Impensis Direct. Laurentii Salvii,, 1759, http://www.biodiversitylibrary.org/bibliography/542. URL http://www.biodiversitylibrary.org/item/10278
3. G. McNeill, S. Vijayakumar, 2d shape classification and retrieval, in: Proceedings of the 19th international joint conference on Artificial intelligence, IJCAI'05, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2005, pp. 1483–1488. URL http://dl.acm.org/citation.cfm?id=1642293.1642528
4. K.-L. Lim, H. Galoogahi, Shape classification using local and global features, in: Image and Video Technology (PSIVT), 2010 Fourth Pacific-Rim Symposium on, 2010, pp. 115–120. doi:10.1109/PSIVT.2010.26.
5. A. Temlyakov, B. Munsell, J. Waggoner, S. Wang, Two perceptually motivated strategies for shape classification, in: Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on, 2010, pp. 2289–2296. doi:10.1109/CVPR.2010.5539912.
6. H. Ling, D. Jacobs, Shape classification using the inner-distance, Pattern Analysis and Machine Intelligence, IEEE Transactions on 29 (2) (2007) 286–299. doi:10.1109/TPAMI.2007.41.
7. X. Bai, W. Liu, Z. Tu, Integrating contour and skeleton for shape classification, in: Computer Vision Workshops (ICCV Workshops), 2009 IEEE 12th International Conference on, 2009, pp. 360–367. doi:10.1109/ICCVW.2009.5457679.
8. A. Tatsuma, H. Koyanagi, M. Aono, A large-scale shape benchmark for 3d object retrieval: Toyohashi shape benchmark, in: Signal Information Processing Association Annual Summit and Conference (APSIPA ASC), 2012 Asia-Pacific, 2012, pp. 1–10.
9. A. Amin, A. Khan, A divide-and-distribute approach to single-cycle learning hgn network for pattern recognition, in: Control Automation Robotics Vision (ICARCV), 2010 11th International Conference on, 2010, pp. 2118–2123. doi:10.1109/ICARCV.2010.5707852.
10. S. Wu, F. Bao, E. Xu, Y.-X. Wang, Y.-F. Chang, Q.-L. Xiang, A leaf recognition algorithm for plant classification using probabilistic neural network, in: Signal Processing and Information Technology, 2007 IEEE International Symposium on, 2007, pp. 11–16. doi:10.1109/ISSPIT.2007.4458016.
11. S. Zhang, Y. Feng, Plant leaf classification using plant leaves based on rough set, in: Computer Application and System Modeling (ICCASM), 2010 International Conference on, Vol. 15, 2010, pp. V15–521–V15–525. doi:10.1109/ICCASM.2010.5622528.
12. T. Beghin, J. Cope, P. Remagnino, S. Barman, Shape and texture based plant leaf classification, in: J. Blanc-Talon, D. Bone, W. Philips, D. Popescu, P. Scheunders (Eds.), Advanced Concepts for Intelligent Vision Systems, Vol. 6475 of Lecture Notes in Computer Science, Springer Berlin Heidelberg, 2010, pp. 345–353. doi:10.1007/978-3-642-17691-3.32. URL http://dx.doi.org/10.1007/978-3-642-17691-3.32

13. P. Tzionas, S. Papadakis, D. Manolakis, Plant leaves classification based on morphological features and a fuzzy surface selection technique, in: Fifth International Conference on Technology and Automation, 2005, pp. 365–370.
14. A. Khan, A. Amin, One shot associative memory method for distorted pattern recognition, in: M. Orgun, J. Thornton (Eds.), AI 2007: Advances in Artificial Intelligence, Vol. 4830 of Lecture Notes in Computer Science, Springer Berlin Heidelberg, 2007, pp. 705–709. doi:10.1007/978-3-540-76928-6.79.
15. A. Khan, P. Mihailescu, Parallel pattern recognition computations within a wireless sensor network, in: Pattern Recognition, 2004. ICPR 2004. Proceedings of the 17th International Conference on, Vol. 1, 2004, pp. 777–780 Vol.1. doi:10.1109/ICPR.2004.1334332.
16. B. B. Nasution, A. I. Khan, A hierarchical graph neuron scheme for real-time pattern recognition, Trans. Neur. Netw. 19 (2) (2008) 212–229. doi:10.1109/TNN.2007.905857. URL http://dx.doi.org/10.1109/TNN.2007.905857
17. Y. Bengio, A. Courville, P. Vincent, Representation learning: A review and new perspectives, Tech. Rep. arXiv:1206.5538, U. Montreal (2012). URL http://arxiv.org/abs/1206.5538
18. A. Muhamad Amin, A. Khan, Single-cycle image recognition using an adaptive granularity associative memory network, in: W. Wobcke, M. Zhang (Eds.), AI 2008: Advances in Artificial Intelligence, Vol. 5360 of Lecture Notes in Computer Science, Springer Berlin Heidelberg, 2008, pp. 386–392. doi:10.1007/978-3-540-89378-3.39. URL http://dx.doi.org/10.1007/978-3-540-89378-3.39
19. C. Mallah, J. Cope, J. Orwell, Plant leaf classification using probabilistic integration of shape, texture and margin features, in: Computer Graphics and Imaging / 798: Signal Processing, Pattern Recognition and Applications (CGIM2013), Acta Press, 2013. doi:10.2316/P.2013.798-098.
20. Frank A., Asuncion A., Uci machine learning repository, Online (2010). URL http://archive.ics.uci.edu/ml