International Journal of Computer Sciences and Engineering Open Access
Research Paper Vol.8, Issue.8, August 2020 E-ISSN: 2347-2693

Fast And Accurate System For Leaf Recognition

S.I. Mostafa[1], Y.M. Abd El-Latif [2*], N.M. Reda[3]

[1]Department of Computer Sciences, Higher Institute for Computer Sciences, Cairo, Egypt
[2]Dept. of Computing and Information Technology, Arab Academy for Science, Technology and Maritime Transport, Cairo, Egypt
[3]Department of Computer Sciences, Faculty of Sciences, Ain Shams University, Cairo, Egypt

[*]*Corresponding Author: Y.Abdellatif@sci.asu.edu.eg, Tel.: +01001719609, ORCID: 0000-0003-0701-087X*

DOI: https://doi.org/10.26438/ijcse/v8i8.7379 | Available online at: www.ijcseonline.org

Received: 15/Aug/2020, Accepted: 24/Aug/2020, Published: 31/Aug/2020

***Abstract*—** Leaf recognition is used in various applications in domains like agriculture, forest, biodiversity protection. Leaf recognition based on images is a challenging task for computer, due to the appearance and complex structure of leaves, high variability between classes, and small differences between leaves in the same class. This paper reviews a state-of-the-art application for building a fast automatic leaf recognition system. We propose a combination of shape, color, texture features and sparse representation extraction for different leaf recognition tasks. In this paper two features databases have been built using 32 classes with 1980 images for Flavia dataset. In recent trends the Graphics processing units (GPU) emerge with high parallel computing capabilities. In this paper we used the computation ability of modern GPU to execute the proposed leaf classification that achieves classification results of 99% and extreme parallelism recognition.

Keywords— Leaf recognition, leaf classification, Morphological features, Online Dictionary Learning, GPU;

I. INTRODUCTION

Plant biodiversity is important for the terrestrial ecosystem, since all living beings depend directly or indirectly on the species of plants that convert carbon dioxide into oxygen (essential for most living organisms) through photosynthesis. Furthermore, these different plants are used for a wide range of industrial and food applications. However, some species of plants are entering the process of extinction. For these reasons, approaches that identify plant species automatically have gained special attention from the pattern recognition community in the last years.

Many researchers have focused on plant identification based on the analysis of leaf images, because leaves are present in plants almost all the time and they usually preserve their shape during plant existence. Many of leaf characteristics, such as shape, texture, color and venation, are considered the features that most generally used to distinguish leaves of different species. There are many researchers have used combination of these features to recognize the leaves. So the challenging task in leaf recognition is to find the features that can distinguish different leaves species especially the symmetric leaves like in Fig.1. in group (a) there are two different leaves images (Pubescent Bamboon, Yew Plum Pine) respectively, in group (b) there are two different leaves images (Ford Woodlotus, Oleander) respectively, in group (c) there are two different leaves images (Camphortree, Japanese Flowering Cherry) respectively and in group (c) there are two different leaves images (Sweet Osmanthus – Big – fruited Holly). Due to this symmetry factor, our recognition system is based on sparse representation, if morphological recognition gives result to two different classes.

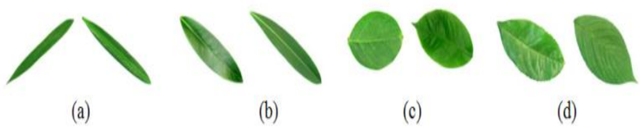

Fig.1. an example of symmetric leaves

In our research, shape, color, texture, and data dictionary features were used as a classifier to leaf class. All experimental results show that the proposed method of classification gives accuracy of 99.75% in all experiments when it was tested on Flavia dataset [1], which contains 32 kinds of plant leaves. It means that the method gives better performance compared to the original works.

GPU's highly parallel structure makes image processing and leaf recognition more efficient and faster than general. Our recognition system was achieved by parallel execution using GeForce 720M GPU. This GPU features 2 multiprocessors, 49152 bytes shared memory and 2 GB device memory. There can be a maximum of 1024 threads per block and 1536 active threads per multiprocessor. Parallel computing is divided into 4 possible classifications according to Flynn: [2] Single Instruction Single Data (SISD), Single Instruction Multiple Data (SIMD), Multiple Instruction Single Data (MISD) and Multiple Instruction Multiple Data (MIMD). To prove the effects and improvement at the level of execution time, we have at first experiment it on CPU and then experiment it on GPUs to understand the improvement in performance.

This paper is organized in the following sections: Section II contain a review to previous work on the modelling of leaves recognition, Section III contain a description of our proposed model, Section IV contain the proposed parallel methodology, Section V contain some experimental results with discussions and Section VI concludes paper work and proposed future work.

II. RELATED WORK

There are several plant identification methods proposed by several researchers. Abdul Kadir represented in [3] a comparative experiment for 4 methods to identify plants using shape features. The experiment was done on 52 kinds of plants with 5 different leaves per plant. The performance accuracy was 64%. The method relying only on shape features dose not achieve maximum performance because of symmetry of leaves shapes. Also Abdul Kadir presented in [4] a leaf classification system using shape, color and texture features, from the images generated previously. He used Probabilistic Neural Network (PNN) as a classifier. The experiment's result gives average accuracy of 93.75 when it was performed on Flavia dataset. Cecilia Di in [5] proposed an innovative method for automatic leaf recognition using 138 features that incorporates shape, color and texture features that were used entirely to train a SVM classifier. The experimental results depend on Flavia dataset is compared with our results in the results and discussion section. Mei Fern in [6] used Centroid Contour Gradient (CCG) method to calculate the gradient between pairs of boundary points corresponding to interval angel.

The average accuracy is 96.6 but only for 5 classes. Sammerchand in [7] represented different features for leaf recognition which were length, width, area of the leaf, perimeter, hull area, hull perimeter and color histogram. The recognition was implemented using k-Nearest Neighbour and tested on 640 leaves from 32 different species created as own dataset named as Folio to obtained average accuracy of 83.5 % in the whole experiment. Badugu presented in [8] automatic polluted leaf identification using texture analysis. In this Article texture features like mean, median , skewness, kurtosis, GLCM and RMS of polluted leaves are review. Manual polluted leafs identification task is time consuming process. Automatic detection of pollutant leaves is a research save a time. Farhana Haque in [9] has designed recognition system consists of three main steps: image pre-processing, feature extraction and matching. Farhana used seven (7) leaf features derived from geometric parameters of leaf shape that were extracted from the pre-processed image and the simple principle of minimum Euclidean distance features used for finding the closest match to the input leaf image. The system used 10 species of leaves with a total of 50 leaf images from Flavia dataset for testing, and achieved accuracy above 90%.

Another factor in leaf recognition is the implementation time. When the number of samples is large in dataset, the method needs a long time to complete the matching procedure.

A contour-based shape descriptor, called the multiscale distance matrix (MDM) presented in [10] by Rongxiang. In this paper MDM was used to capture multi-scale geometric properties of a shape. This method's purpose is to decrease matching time and the researcher compared the results with the recognition time of Inner-Distance Shape Context (IDSC) and Fourier shape Descriptor (FD) which were reported in [11]. The MDM approach consumes much less time than IDSC in the experiment performed on ICL Leaf dataset and Swedish Leaf dataset. In a real IDSC needs more time to match two common leaves and the MDM approach is more suitable for a real-time recognition system. Also Nordin in [12] presented a classification system using Open Multiprocessing (Open MP) to decrease recognition execution time. In this paper two parallization strategies were implemented (fine-grain) and (coarse-grain) to compute leaf image segmentation. The segmentation methods used in this paper are Canny Edge Detector and Otsu thresholding methods. As a result of this paper the coarse-grain approach achieved less execution time for Canny Edge Detector method. From the previous presentation we conclude that when the size of the testing image is large or when the number of samples in recognition set is large, then parallel implementation is more needed. Walunj viewed in [13] an enhancement system for images. This system is as hierarchical framework of 3 layers. this system depend on image search, patch matching and image synthesis (image reconstruction). the system achieve better results in terms of Time, Recall, Precision and measure(Accuracy) and enhances the speed of reconstruction of reconstruction of image by 5-6% and also increases the accuracy by 8% to 10% by using GPU.

III. PROPOSED LEAF RECOGNITION MODEL

The main goal of our paper is presenting parallel implementation of Leaf recognition system with classifiers trained on diverse descriptors extracted from the leaf image. In our recognition system dataset was divided into training set and testing set and the method was composed of two stages: training and testing. We will describe each stage in the next subsections.

A. Training Stage

In the training stage the features were extracted from all leaves in training set based on two descriptor algorithms (morphological and data dictionary).

Morphological descriptor was used to extract three different features (shape, color, texture) from 32 different classes in training set, each class contains 10 copies of leaf type, which produced 6720 features used to build feature dictionary database. Data dictionary descriptor used to extract sparse representation and build online dictionary database with matrix of 32 array × 3200000 features cell for all classes as far as 100000 patch samples for each leaf.

The training stage overview is presented in Fig. 2. Initially, the images were submitted to the pre-processing stage to remove backgrounds and unwanted structures, after that the features extraction was performed by two different methods, then, each extracted feature was stored in a features dictionary database and online dictionary database. The next subsections present a detailed outline of features extracted in the proposed method.

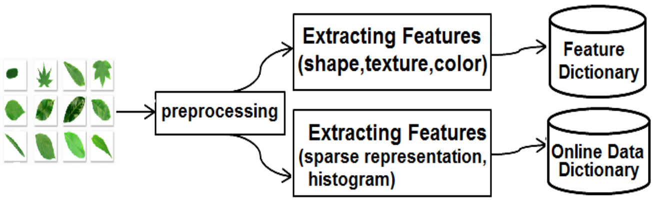

Fig. 2. Extracting features and Databases building (training stage)

1) *Features extraction*

Features were extracted from all classes in the training dataset and used to build two features datasets. The features dictionary dataset is represented as feature vector with 19 range value depending on features of shape, (17) color and texture. Online data dictionary depends on sparse representation and histogram. All the features that were used in recognition system are described as follows.

a)Shape Features:

Compound of about 17 features, from some basic features we derived and obtained another feature values quickly and automatically. Diameter, Aspect ratio, physiological width, Physiological Length, Form factor, Rectangularity, Narrow factor, and Perimeter ratio of diameter computed as in [14], Leaf Area Calculation as in [15], Solidity and Irregularity as in [16], circularity as in [16], and rest is as follows.

- Leaf perimeter Calculation (Contour extracted)

In all leaves dataset the leaves were scanned on a white sheet (background). To classify the leaf we cropped it from the background. The entered colour image was converted to a grayscale image then to a binary image and the leaf contour was extracted. The perimeter referred to as P is calculated by counting the number of pixels forming leaf margin using the black outline on white background (contour).

Leaf perimeter = length (contour)

- Convex Hull

Convex hull is points in the space around all edges of a leaf presented in a matrix form [18].

- Perimeter ratio Convex Hull

Perimeter of convex hull of the leaf is the number of points in its convex hull matrix. The convex hull K is expressed in terms of a vector of point indices arranged in a counterclockwise cycle around the hull.

Perimeter_ratio_of_conhull=length (K)

- Leaf Apex:

Determine leaf Apex by getting the peaks of the leaf using leaf contour matrix. Find peaks vector by comparing contour point to its neighbouring values. If a value of point is larger than both of its neighbours or equals inf value, then the point is a local peak. If there are no local maximum, then peaks are an empty vector.

- Leaf Tip:

Tip of the leaf is calculated by the average between all peaks and the leaf perimeter.

Leaf tip = peaks / perimeter

b) Color features (HSV):

Color features of a color image are represented in column i_{th} and row j_{th}. HSV stands for hue, saturation, and value. The HSV representation rearranges the geometry of RGB in an attempt to be more intuitive and perceptually relevant [19].

c) Texture features:

Image texture gives us information about the spatial arrangement of intensities in the leaf. It can be defined as an entity consisting of mutually related pixels and group of pixels. One of image texture method that used to measure the leaf smoothing is

- *Region Covariance Matrices*

The covariance matrix is a statistical method. It is used to calculate the covariance between pixel values using edge-based filters. It has fast computations ability because the pixel weight in image is based on the gradient magnitude at that pixel [4].

d) Sparse representation:

Sparse representation is a sparse code for the input data in the form of a linear combination of basic elements. These elements are called atoms. It is achieved by optimizing a sparse dictionary which is formed as a sparse structure $D = B * A$ Where B is a fixed base dictionary and A is a sparse matrix. The matrix D is a sparse dictionary matrix of size N × K, containing the sparse representations of the dictionary atoms over B [20].

To learn a dictionary for sparsely representing features vector from an image, we collected a set of training exemplars, y_j, j = 1, 2,…, P, to learn a dictionary D sparsifying y_j by solving the optimization problem in (1) :

$$\min_{D,x_j} \frac{1}{P} \sum_{j=1}^{P} \left(\frac{1}{2} \| y_j - D x_j \|_2^2 + \lambda \| x_j \|_1 \right) \tag{1}$$

Where x_j denotes the sparse coefficient vector of y_j with respect to D and λ is a regularization parameter. This equation can be efficiently solved by performing a dictionary learning algorithm, such as the online dictionary learning in [21].

In this paper some of the previous features are achieved using some built in matlab functions. All extracted features are representing as feature vector (FV) of size 19 and stored in database respect to each class.

B. Test Stage

The testing dataset contains about 20~ 40 images for each class, amounted to almost 1000 images. Test stage shown in Fig.3. Started by pre-processing leaf image by discarded white background and extraction of leaf features. Then we used the descriptors to calculate the score and correlation to determine the leaf class.

Score method:

The extracting Fv for a testing leaf represent the satisfied comprise feature with Fv that stored in database in respect to its class if in range assign 1 otherwise assign 0. In the end we were getting the maximum visible feature vector according to the estimated class.

	1						19
Fv	1	0	1	1..0	0	0	1

$$score_{\max j} = \sum_{i=1}^{19} (fv_{ij}) \quad (2)$$

Correlation method:

Correlation is a statistical technique used to measure the strength of the relationship between two variables. To compute the correlation between images sparse representation is extracted as w for a test image then used to get the correlation between it and the data dictionary D with expected values μ_w and μ_D and standard deviations σ_w and σ_D that are defined as:

$$\text{Corr}(w,d) = ((w-\mu_w)(D-\mu_D))/\sigma_w\sigma_D \quad (3)$$

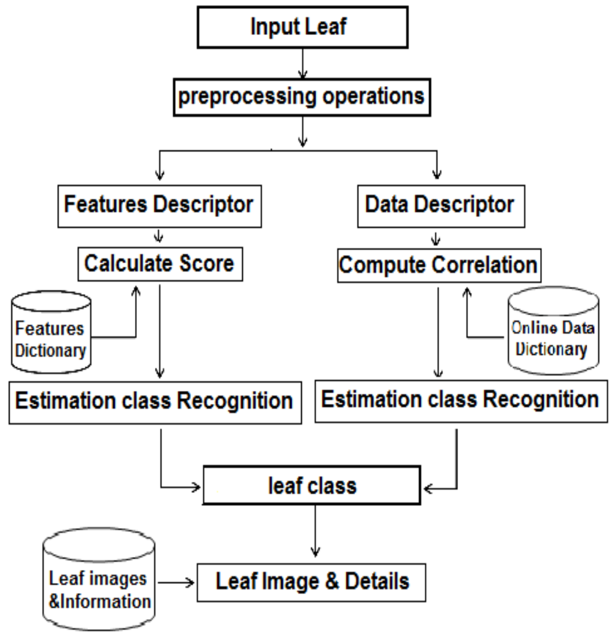

Fig. 3. Extracting features and leaf recognition (testing stage)

IV. PROPOSED PARALLIZATION IMPLEMENTATION OF LEAF RECOGNITION

A. GPU architecture

The graphics processor with its massively parallel architecture is a storehouse of tremendous computing power. The parallel capabilities of the GPU are easy development and deployment of general purpose computations especially in image processing and computer graphic. CPUs have few cores that are optimized to perform sequential computing while GPUs have thousands of cores which are specially designed for parallel processing. So a significant speedup can be achieved by executing high computational work on GPU while rest of code in CPU.

B. Proposed GPU implementation

Our parallel implementation was divided into two parts on CPU and on GPU as shown in Fig. 4. In CPU implementation part all functionality of leaf recognition system that does not take long execution time is implemented by using CPU as (loading the testing image and image pre-processing). In GPU implementation part some functionality has been implemented using CPU (host) and most of functionalities have been implemented using GPU (device) with data parallelization. Parallel implementation was divided into blocks according to feature group (shape, texture, color, sparse code), then every block divided the work on multi threads. The leaf testing set is in shared memory between all blocks.

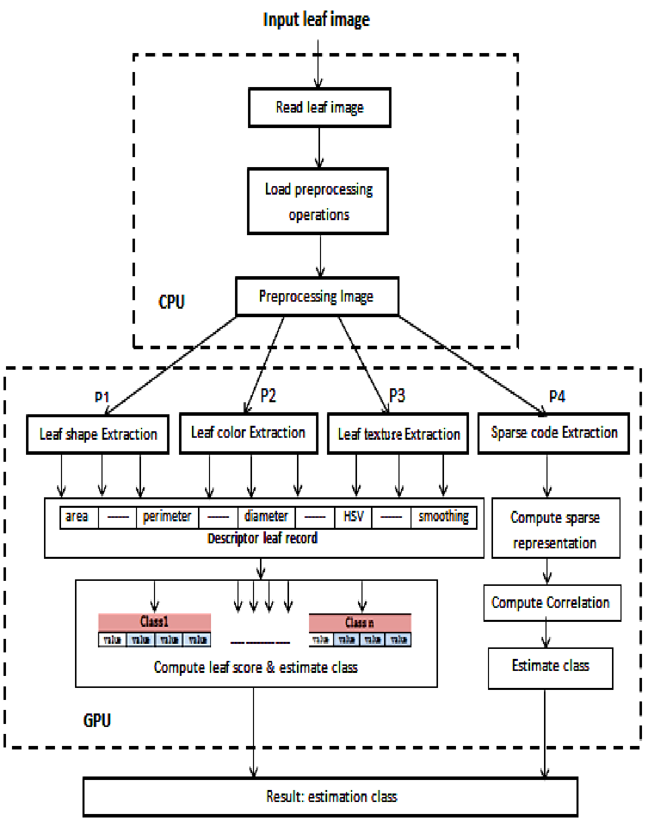

Fig. 4. Proposed parallel methodology

GPU based on feature descriptor recognition implementation
comprises of two main steps
1) Extracting shape, color and texture features.
2) Calculating the features score depending on features database.

GPU based on data descriptor recognition implementation comprises of two main steps:
1) Extracting sparse representation features.
2) Calculating the correlation between sparse representations extracted from input leaf and sparse data dictionary.

At the final step the result from descriptor detects the leaf name and other information.

V. RESULTS AND DISCUSSION

To measure the performance of the proposed leaf recognition system approach we used "Flavia leaf image dataset". This section presents the main characteristics of the dataset, the proposed parallel implementation, and a comparison with state-of-the-art approaches to assess the performance of the proposed system.

A. *Dataset*

Flavia leaf images dataset is a popular dataset. It consists of 32 classes of leaf images where each class contains 40-60 images. All images in the dataset is about 1907 images. They are full color images with white background.

In our proposed system the dataset is divided into 10 common leaf types as shown in Table I, the leaf types are linear, Hastate, Toothed, unorganized, elliptical, peltate, obovate, reniform, lanceolate, Cordate, Oblong and Ovate leaf.

Table 1. Proposed Dataset Classification

Leaf type	Number of images
Linear	314
Hastate	174
Toothed	65
Unorganized	164
Elliptical	491
Peltate	111
Obvate	172
Reniform	62
Lanceolat	54
Cordate	64
Oblong	56
Ovate	180

B. *Experiment Environment*

Our experiments were implemented in matlab R2014a (64 bits version) on a GPU computer equipped with GeForce GTX 770 processor. In addition to the function and tools that are used within matlab, all the experiments used ompbox1, ompbox10 multi-threaded C coded loops and mtimesx_20110223 in sparse code level. MTIMESX3 attempts to do most efficient algorithms for memory access, and in some cases can outperform MATLAB by 3x- 4x for faster speed.

In our experiments, we have used 95% of the images in the datasets. Furthermore, this dataset's images have also been extensively used by other researchers, which allow us to compare the results achieved by the proposed approach with other approaches in the literature. Also we compare execution time using sequential implementation with the execution time in parallel implementation.

C. *Proposed system implementation*

Proposed recognition system GUI is shown in Fig. 5. Our system starts with loading an image to the system. Then, the features are extracted using analysis button. Then, the score is computed using the feature vector method as shown previously. Then we compared all the scores to get the class that has maximum score that denote to proposed class for the tested image. In case of similar lea the classes scores are close, so data correlation is the solution. After class recognition, classification is performed using plant information that translate the class index into name and category and other information about the plant.

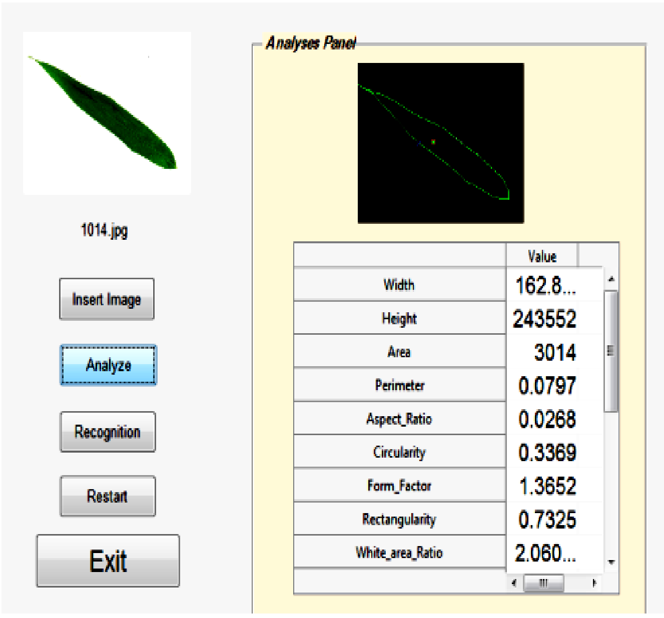

Fig. 5. Proposed Recognition System GUI

D. *Result and discussion*

In the proposed system only 40% of dataset is used for training and 60% for testing to be shure of system accuracy. Accuracy of all experiments was 100% except for 3 types where their accuracy were between 98.5 ~ 99.5.

These types were very close leaves label #1, label #20 and leaf label #26. For those types recognition by data descriptor using sparse representation is supported. Leaf label #31 and leaf label #32 also were difficult because there leaves have not exact shape features.

Table 2. Experiments Result Comparisons

Common name	Label	Species Samples	[5]	[9]	Our proposed
Pubescent Bamboo	1	59	98.2%	100%	99.8%
Chinese Horse Chestnut	2	63	98.4%	--	100%
Chinese Redbud	3	72	98.2%	--	100%
True Indigo	4	73	100%	--	100%
Japanese Maple	5	56	100%	100%	100%
Nanmu	6	62	98%	--	100%
Castor Aralia	7	52	100%	--	100%
Goldenrain Tree	8	59	100%	--	100%
Chinese Cinnamon	9	55	100%	--	100%
Anhui Barberry	10	65	98.2%	--	100%
Big-fruited Holly	11	50	100%	--	100%
Japanese Cheesewood	12	63	98%	--	100%
Wintersweet	13	52	98%	--	100%
Camphortree	14	65	100%	--	100%
Japanese Viburnum	15	60	100%	--	100%
Sweet Osmanthus	16	56	100%	--	100%
Deodar	17	77	100%	--	100%
Ginkgo Maidenhair Tree	18	62	100%	60%	100%
Crape Myrtle	19	61	100%	100%	100%
Oleander	20	66	98.3%	--	98.5%
Yew Plum Pine	21	60	98%	--	100%
Japanese Flowering Cherry	22	55	100%	--	100%
Glossy Privet	23	55	100%	--	100%
Chinese Toon	24	65	98.2%	--	100%
Peach	25	54	98%	--	100%
Ford Woodlotus	26	52	94%	--	98.5%
Trident Maple	27	53	100%	100%	100%
Beale's Barberry	28	55	100%	80%	100%
Southern Magnolia	29	57	100%	100%	100%
Canadian Poplar	30	64	98%	100%	100%
Chinese Tulip Tree	31	53	100%	100%	100%
Tangerine	32	56	100%	100%	100%

Our recognition system achieves the greatest accuracy and the fastest implementation compared to state of the art applications as shown in Table II. We included the result presented in [5] there are 183 features denoted to repeat extract and more time to execute. In our proposed application only 21 features were used and has achieved more efficient accuracy. In [9] seven (7) leaf features were used for finding the closest match to the input leaf image. This system used 10 species of leaves with a total of 50 leaf images to obtain accuracy above 90%. Compared with our proposed system, it is slow and was not tested on a large dataset, while our system achieved more accuracy.

To verify our performance in execution time using GPU we compared our implementation with traditional CPU implementation for the same recognition. In all results the parallel execution time is less by about 4 times than sequential execution. Time is measured by millisecond.

VI. CONCLUSION AND FUTURE SCOPE

In this paper we propose a developed plant leaf recognition system based on geometric and morphological features implemented in GPU.

The propsed leaf recognition system based on two descriptor algorithms (morphological and data dictionary). Morphological descriptor was extract different 19 features combined from three groubs (shape, color, texture) .In data dictionary descriptor sparse representation was extracted and build online dictionary database .

The proposed system implementation was divided into two parts on CPU and GPU. On CPU part, the system implemented all functions that not take long execution time as (loading testing image and image pre-processing). On GPU part, most system functions were implemented. In Parallel implementation GPU device the execution was divided into blocks according to features groups (shape, texture, color, sparse code), then every block task was divided on multi threads, and the leaf testing set was used as shared memory between all blocks.

The system performs the computations correctly and it takes a minimum time to process the images and find the closest match, with 99% of accuracy. By using best two matching techniques were used in recognition and displaying leaf information as a result.

In the future of leaf recognition system, we can provide better discrimination in features domain than the traditional morphological features for plant leaves. Also we will improve the plant recognition system to cover variant datasets and to display more information about plants. We will continue to learn about the best option to classify leaves to improve leaf recognition system based on GPU which can carry a lot of computations in a parallel way. We will test our implementations with different dataset to test our results in a better way with huge types of data. We will test the performance of the learning transfer, doing a little research about classifying.

REFERENCES

[1] J. Hsiao, L. Kang, C. Chang , C. Lin ," *Learning-Based Leaf Image Recognition Frameworks*", Springer International Publishing Switzerland , Intelligent Systems in Science and Information 2014, Studies in Computational Intelligence , pp. 77-91,**2015**.

[2] M. J. Flynn. "*Some computer organizations and their effectiveness",* IEEE Transactions on Computers, Vol.**21**, Issue.**9** , pp. 948–960, **1972**.

[3] A.Kadir, L. E. Nugroho , A. Susanto, P. I. Santosa , " *A comparative experiment of several shape methods in recognizing plants*", International Journal of Computer Science & Information Technology, vol.**3**, No **3**, pp.256–263 , **2011**.

[4] A. Kadir, L. E. Nugroho, "*Leaf Classification Using Shape, Color, and Texture Features* ", International Journal of Computer Trends and Technology, **July** to **Aug** Issue, pp.225-230, **2011**.

[5] C.D. Ruberto, L. Putzu, "*A fast leaf recognition algorithm based on SVM classifier and high dimensional feature vector*", In Proceedings of the 9th International Conference on Computer Vision Theory and Applications, pp. 601-609, **2014**.

[6] B.M. Fern, G. bin, "*Leaf Recognition Based on Leaf Tip and Leaf Base Using Centroid Contour Gradient*", American Scientific Publishers, vol.**20**, pp.209-212, **2014**.

[7] S. Pudaruth, T. Munisami, M. Ramsum, "*Plant Leaf recognition using shape features and color histogram with K-nearest neighbour classifier*" , Science Direct, Procedia Computer Science , vol. **58** , pp. 740–747, **2015**.

[8] S.K. Badugu, R.K. Kontham, V.K. Vakulabharanam, B. Prajna, "Calculation of Texture Features for Polluted Leaves," International Journal of Scientific Research in Computer Science and Engineering, Vol.**6**, Issue.**1**, pp.11-21, **2018**.

[9] F. Haque ,"*Plant recognition system using leaf shape features and minimum Euclidean distance*", ICTACT Journal on Image and Video Processing, vol. **9**, issue.**2**, **2018**.

[10] R.Hu,"*Multiscale Distance Matrix for Fast Plant Leaf Recognition*", IEEE Transactions on Image Processing, vol. **21**, Issue **11**, pp. 4667 - 4672 , **2012**

[11] H. Ling , D.W. Jacobs, "*Shape Classification Using the Inner-Distance*" IEEE Transactions on Pattern Analysis and Machine Intelligence, vol.**29**, no. **2**, pp. 286-299, **2007**

[12] M. Nordin , A. Fakhri , "*Image Segmentation Using OpenMP and its application in plant species classification*" International Journal of Software Engineering and Its Application, vol.**9**, no. **5**, **2015**.

[13] S.M. Walunj, S.V. Gaikar, A.D. Potgantwar, "Accelerate Image Reconstruction Using GPU," International Journal of Scientific Research in Network Security and Communication, Vol.5, Issue.3, pp.68-75, **2017**.

[14] V. Satti, A. Satya, S. Sharma, "*An Automatic Leaf Recognition System for Plant Identification Using Machine Vision Technology*", International Journal of Engineering Science and Technology (IJEST), Vol.**5** , No.**04** , **2013**.

[15] A. N. Cheeran, A. K. Chaudhari, "*Fast and Accurate Method for Leaf Area Measurement*", International Journal of Computer Applications, Vol.**49**, No.**9**, pp.22 – 25, **2012**.

[16] A.H. Kulkarni1, H.M.Rai, "*A Leaf Recognition Technique for Plant Classification Using RBPNN and Zernike Moments*", International Journal of Advanced Research in Computer and Communication Engineering, Vol.**2**, Issue.**1**, **2013**.

[17] X.Feng, "*Recognition of Leaf Images Based on Shape Features Using a Hypersphere Classifier*", Springer-Verlag Berlin Heidelberg, vol.**3644**, pp. 87-96, **2005**.

[18] M. Berg, O. Cheong, M. Kreveld , M. Overmars, "*Computational Geometry: Algorithms and Applications*", Springer-Verlag Press, New York, pp.45-61, **2000**.

[19] D. Naglot , P. S. Kasliwal , S. J. Gaikwad , N. D. Agrawal, "*Indian Plant Recognition System Using Convolutional Neural Network*", International Journal of Computer Sciences and Engineering (JCSE), Vol.**7**, Issue.**6**, **2019**.

[20] S. Mallat, "*A Wavelet Tour of Signal Processing*", Third Edition, Elsevier Inc., United States, pp.1-31, **2008**.

[21] S. Ibrahem, Y. M. Abd El-Latif, N. M. Reda , "*A novel data dictionary learning for leaf recognition*", Signal & Image Processing: An International Journal (SIPIJ), Vol.**10**, No.**3**, **2019.**

AUTHORS PROFILE

Yasser M. Abd El-Latif, PhD

Associate Professor of Computer Science, Faculty of Science, Ain Shams University, Egypt. Egyptian, born in Kuwait on 15/04/1971, and graduated from the Faculty of Science in Mathematics and Computer Sciences in 1992. I obtained my MSc in Computational Geometry in 2000 and PhD in the field of Subdivision Surfaces in 2005 from the Faculty of Science, Ain Shams University, Cairo, Egypt. I worked as a Manager of Ain Shams University MIS project (2006-2014).

Currently I am Associate Professor of Computer Sciences, College of Computer and Information Technology (CCIT), Arab Academy of Science, Technology and Maritime Transport (AASTMT) since Sep. 2018 tell now.

I published 22 articles in Computer Graphics and Computational Geometry. I supervised 3 MSc theses and 4 PhD theses in Computational Geometry and Bioinformatics and Computational Biology.

I am a member of the Egyptian Mathematical Society (EMS) since 1995, and a reviewer in the American Mathematical Society (AMS) since 2008 (MR: 057789) and a reviewer in the Arabian Journal for Science and Engineering (AJSE).

Naglaa M. Reda is currently an Assistant Professor of Computer Science at the Division of Computer Science, Department of Mathematics, Faculty of Science, Ain Shams University, Cairo, Egypt. She received her M.Sc. in Computer Science from the University of Ain Shams in 1998. She received her Ph.D. in Computer Science from the University of Ain Shams in 2005. She has published some papers in international referred journals and in proceedings of international conferences. Her research interests focus on parallel algorithms, high performance parallel and distributed computing systems and IOT.

Mrs Shimaa I M has B.Sc. of Computer Science, Faculty of Computer Science from New Cairo Academy, May 2007 with Grade: Excellent with Honor Degree and Master of Science of Computer Science from Faculty of Science, Suez Canal University, Egypt, 2015 with Title: "Parallel Image Processing Algorithms ". She is registers for Ph.D.degree of Computer Science in faculty of Science at Ain Shams University, Cairo, 2016 With Title: Parallel Algorithms for plant leaf recognition. She Interests in image processing Applications – Design and Analysis of parallel Algorithms

