ISSN 2047-3338

A Study of Image Processing in Agriculture Application under High Performance Computing Environment

A Fakhri A Nasir, M Nordin A Rahman and A Rasid Mamat

***Abstract*—The available literatures on image processing in agriculture application under high performance computing (HPC) are limited and sometimes are not discussed in details. This paper reviewed the steps of image analysis done in some image processing focusing on agriculture application and also the details analysis of parallel and distributed image processing. The memory architecture in parallel and distributed image processing and some suitable application programming interface (API) in parallel and distributed image processing are examined. In general, this study provides basic understanding of parallel and distributed image processing for agriculture application.**

***Index Terms*— Distributed Image Processing, High Performance Computing, Image Processing in Agriculture Application and Parallel Image Processing**

I. INTRODUCTION

IN the application of agriculture science such as image processing, parallel and distributed computing reduces the computational time and as a result, plant recognition can be made much faster. With a massive volume of plant species data and extensive computing for plant recognition, the process becomes more complex and requires longer time. However, it becomes a big challenge for any system designer to design an image processing system using parallel and distributed computing. There are limited literatures on parallel and distributed image processing for agriculture application unlike other application, such as for medical imaging application. For example, [1] introduced medical image processing on a massively parallel computer using single-instruction multiple-data (SIMD) computer. Meanwhile [2], compares and contrasts the research issues involved with implementing computationally medical imaging algorithms on a SIMD and multiple-instruction multiple-data (MIMD) parallel processing computers. The testing result shows MIMD implementation is at least four times faster than the SIMD implementation.

There are various types of image processing systems for agriculture application that have been developed with different purposes. For example, to identify disease, [3] have built a system for identification of symptoms for cotton leaves, while [4] have identified the vine disease by looking at the color of vine leaves. Besides, [5] has introduced identification system to identify the bamboo species using shape features. Meanwhile, [6] also used shape features to analyze soybean leaves. The system produced by [5] and [6] are intended to identify plant species and varieties. Table I summarized some other image processing examples used in agriculture application.

The objectives in this paper are as follows:

1) Describe parallel and distributed image processing with a light and easy manner.
2) Present the study of parallel and distributed image processing with emphasis on the mechanisms used focusing on agriculture application.
3) Motivate the reader for further research work to apply parallel and distributed image processing in agriculture.

This paper also discussed in detail how the parallelism is possible for image processing in agriculture. The fundamental steps and levels of image processing in general are also examined. The paper is organized as follows: Section II describes the general concept of high performance computing, while Section III gives the explanation of image processing fundamental steps. Section IV analyzes the interrelated issues in high performance computing and image processing. The overview of parallel processing for image processing recognition in agriculture application is discussed in Section V. Conclusion of the article is placed in Section VI.

II. HIGH PERFORMANCE COMPUTING

HPC is a methodology used to solve high complexity problems such as computing huge workload and data, and intensive critical analysis. HPC reduces the computational time and consequently the results are produced efficiently and decision making can be made much faster. Parallel and distributed computing is an approach that is offered in HPC. It has successfully solved the complex problems in science, engineering and business areas.

A Fakhri A Nasir, M Nordin A Rahman, and A Rasid Mamat are with Faculty of Informatics, Universiti Sultan Zainal Abidin (UniSZA), Gong Badak Campus, 21300 Kuala Terengganu, Malaysia.

TABLE I
EXAMPLE OF PREVIOUS WORK ON IMAGE PROCESSING IN AGRICULTURE APPLICATION

No.	Author	Year	Plant	Research Scope
1	Singh et al.	2011	Bamboo	Classification of bamboo species system based on shape features
2	Fellegari and Navid	2011	Orange	Determining the size of orange (size in volume)
3	Patil	2011	Betel	Betel leaf area measurement using reference object
4	Xu et al.	2011	Tomato	Leaf color images to identify nitrogen and potassium in tomato
5	Chen and Zhou	2010	Plant root	Plant root 2D scanner by measuring surface area, length and diameter
6	Zhang and He	2010	Not Stated	Measurement of plant leaf area by using DPI (dots per inch) scanner
7	Suo et al.	2010	Cotton	Artificial neural network to predict leaf population chlorophyll
8	Shirgahi and Danesh	2010	Barberry	Fuzzy system to determine barberry product quality
9	Pengyun and Jigang	2009	Not Stated	Computer assistance spores counting using detection of exterior outline
10	Camargo and Smith	2009	Cotton	Image classification for the identification of disease causing agents
11	Ishak et al.	2009	Weed	Image classification using Gabor wavelet and gradient field distribution
12	Meunkaewjinda et al.	2008	Grape	Grape leaf disease detection by looking at grape color
13	Phadikar and Sil	2008	Rice	Prototype system for rice disease detection
14	Park et al.	2008	Aqua-plants in Korea	Utilizing venation features for efficient leaf image retrieval
15	Du et al.	2007	Various plant leaves	Leaf shape based plant species recognition
16	Christofi et al.	2007	Various hazelnut leaves	Estimating leaf area from linear measurements

A) HPC Architecture

Three architecture in achieving parallelism which is by using parallel processing hardware (a computer with multi-processors), distributed system (multi-computer connected to each other using networking system) and hybrid system which is a combination of both parallel processing hardware and distributed system. All of these approaches have their own advantages and disadvantages. However, the selection of this approach is depends on the needs of the application, which is depends on approach of parallelism and available budget. Based on memory usage on different computer processing units (CPUs), the HPC architecture can be classified as shared memory and distributed memory [7]. The classification that is often used for memory usage in parallel processing architecture is Taxanomy Flynn. Taxonomy Flynn has been used since 1966 [8]. This concept has two dimensions depending on the instruction and the data that have only two possibilities, one or many. HPC architecture can be classified into four categories namely single instruction, single data (SISD), single instruction, multiple data (SIMD), multiple instruction, single data (MISD) and multiple instruction, multiple data (MIMD) [9].

1.1 Parallel Processing Hardware

Parallel processing hardware architecture is using shared memory that has many limitations. In this architecture, two or more processors compute many tasks simultaneously. In shared memory, there are many processsing units (PUs) sharing the same memory at one time as shown in Fig. 1. It uses the concept of global address in which each data can be accessed by all processors.

1.2 Distributed System

Distributed system is using distributed memory. It is almost same to shared memory but only sharing a local memory. Distributed memory systems require a communication network to connect inter-processor memory. Processors have their own local memory. Memory addresses in one processor

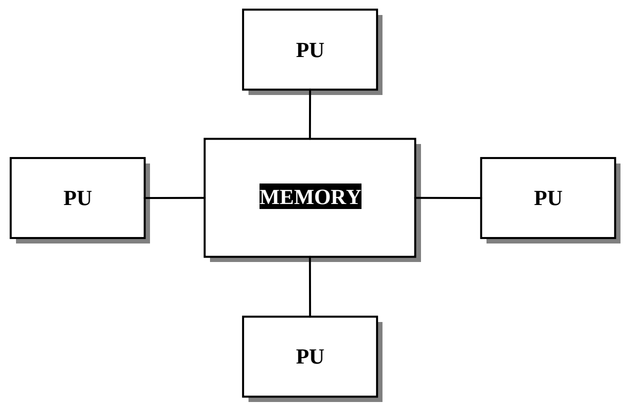


Fig. 1. Share memory [10]

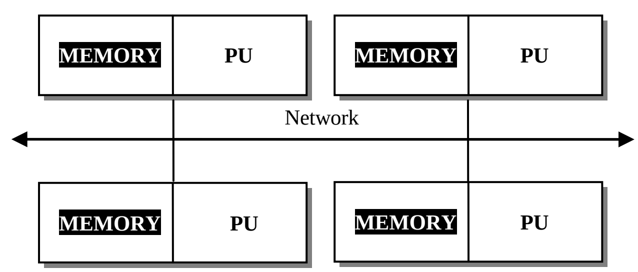


Fig. 2. Distributed memory [10]

do not map to another processor, so there is no concept of global address space across all processors. Fig. 2 illustrates distributed memory concept.

There are two main architectures in distributed system: master slave and peer-to-peer [7].

Master Slave

In this approach, the master processing unit distributes the image data to the slave processing units. All slaves processing units work in parallel to compute assigned task. Then master processing unit gathers back the image. The master-slave architecture approach uses the "Distribute and Gather" philosophy for parallel image processing.

Peer-to-Peer

In peer-to-peer architecture, each participating entity has the same capabilities and either entity can initiate a communication. The participating entities make a portion of their resources directly available to other networked participating entities, without the need for central coordination (such as master or controller in master slave architecture).

1.3 Hybrid System

The largest and fastest computers in the world today employ both shared and distributed memory architectures. Fig. 3 shows the concept of hybrid distributed-shared memory.

The distributed memory component is the networking of multiple shared memory machines, which only know about

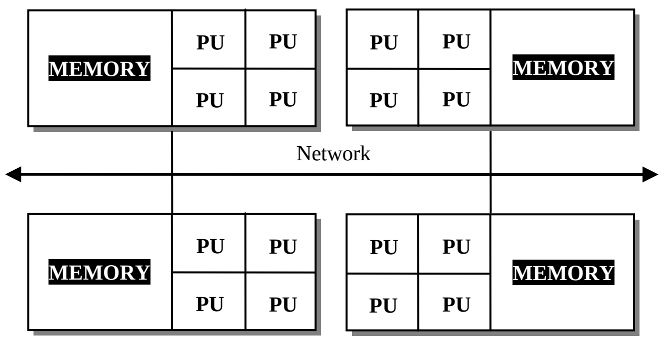


Fig. 3. Hybrid distributed-shared memory [10]

their own memory - not the memory on another machine. Therefore, network communications are required to move data from one shared memory machines to another.

B) Application Programming Interface for HPC

Application programming interface (API) is a specification intends to be used as an interface by software components to communicate with each other. An API specification can take many forms, including an International Standard such as POSIX or vendor documentation such as the Microsoft Windows API, or the libraries of a programming language, such as Standard Template Library in C++ or Java API. Some of the most widely used API for parallel and distributed image processing is presented in the following paragraphs.

OpenMP is an API for writing multithreaded applications that have a set of compiler directives and library routines for parallel application programmers. It was originally developed in Fortran and later on in C/C++ [11]. The Message Passing Interface (MPI) is defined as an application programming interface for developing parallel programs that explicitly use message passing for inter-process communication [12]. The standard is defined as the syntax and semantics of a core of library routines useful to a wide range of users writing portable message-passing programs in Fortran 77 or the C programming language.

POSIX Threads, usually referred to as Pthreads, is a POSIX standard for threads. The Pthreads library is a set of C calls that provides a mechanism for writing multithreaded code. In shared memory multiprocessor architectures, threads can be used to implement parallelism.

MATLAB's Parallel Computing Toolbox (PCT) with MATLAB Distributed Computing Server (MDCS) [13] is a very good tool support for array and metric processing. Using PCT, the MATLAB can allow solving image processing problems using multi-core processors, graphics processing units (GPUs) and computer clusters. To support this capability, high-level constructs parallel for-loops or special array types that allow us to parallelize MATLAB applications without MPI programming.

The MPJ Express software is an object-oriented Java communication interface protocol, and has been utilized to operate as communication interface protocol between processors [14]. It is an open source Java message passing library that allows application developers to write and execute parallel applications for multi-core processors and compute clusters/clouds. This software functions support point-to-point communication, group communications, synchronization etc. MPJ Express supports for the designed SPMD architecture, in which multiple processors cooperate by executing the single program placed in master processor and all workers have their own local data. The summarizations of some API for parallel processing architecture and distributed processing architecture are listed in Table II.

TABLE II
SUITABLE API FOR DIFFERENT MECHANISMS

Architecture	Suitable API
Parallel processing	OpenMP, POSIX Threads, MATLAB's PCT with MDCS, MPJ Express
Distributed computing	MPI, MATLAB's PCT with MDCS, MPJ Express

III. IMAGE PROCESSING

A digital image is a representation of a two-dimensional image as a digital value called pixels. Digital image processing is the technology of applying a number of computer algorithms to process digital image. The outcomes of this process can be either images or a set of representative characteristics of the original images. Digital image processing is used to improve pictorial information for better clarity by human interpretation and to automatic processing of scene data for interpretation by machine/non-human [15].

A) Fundamental Steps in Image Processing

The fundamental steps in image processing are image grabbing or acquisition, preprocessing, segmentation,

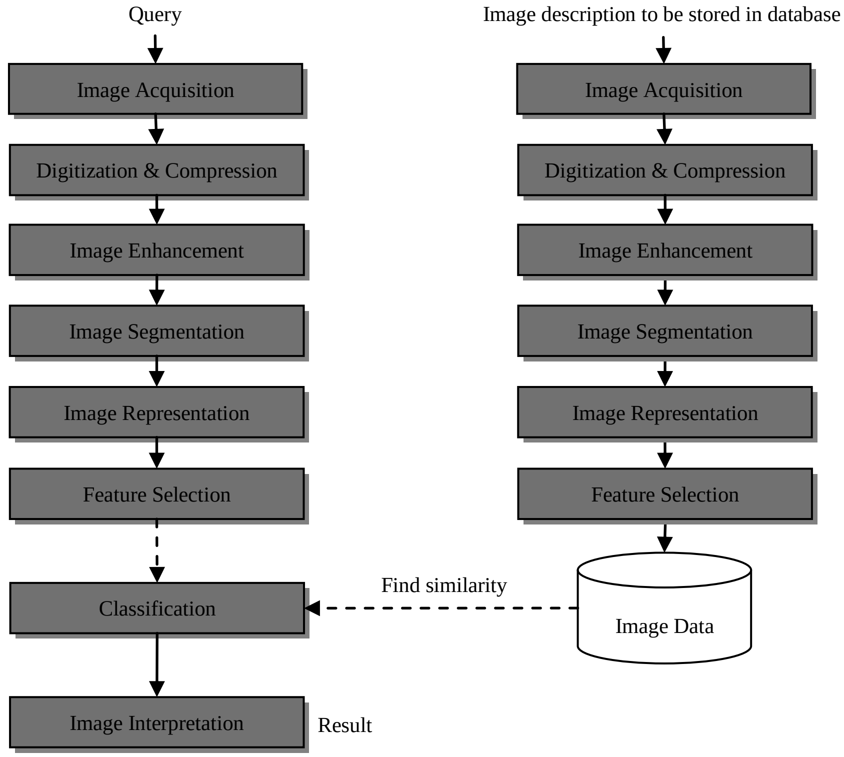


Fig. 4. Fundamental steps in image processing

representation and description, and recognition and interpretation [15]. The descriptions of these steps are given in the following subsections. The summarization of the steps in image processing is shown in Fig. 4.

1.1 Image Acquisition

An image must be converted to numerical form before processing. This conversion process is called digitization. This process is done by charge-couple device (CCD) that is embedded in modern digital camera [15]. There are many different types of digital camera has been used to acquire digital images. It was selected based on the needs and budget available for research. Some digital cameras used in agriculture application are 3M pixel real color camera [16], Kodak DC50 zoom camera [17], Olympus C-5 060 wide zoom camera [18], Nikon Coolpix P4 digital camera in macro mode [19] and Panasonic DMC-LX1 camera [20].

1.2 Image Preprocessing

After a digital image is obtained, the next step is preprocessing. The key function of preprocessing is to improve the image in order to get better results for the other processes. It typically deals with techniques for enhancing contrast, removing noise and isolating regions. There are three main categories of image preprocessing which is image compression (used to reduce the amount of computer memory needed), image enhancement (to modify the brightness and contrast of an image) and image measurement (involves segmenting the image to separate the objects of interest from the background [15]. For example in agriculture application, [21] used median filter as a method to remove noise in preprocessing stage.

1.3 Image Segmentation

The first step in image analysis generally is to segment the image. Segmentation subdivides an image into its constituent parts or objects. The segmentation process should stop when the objects of interest have been isolated. In general, autonomous segmentation is one of the most difficult tasks in image processing. For example, [19] used entropy based bi-level thresholding method for segmenting the images to facilitate identifying the infected parts of the leave. [22] also separated the background image from the major part of rice leaf image in image segmentation stage.

1.4 Image Representation and Description

Representation and description almost always follow the output of a segmentation stage. The first decision must be made whether the data should be represented as a boundary or complete region. Boundary representation is appropriate when the focus is on external shape characteristics whereas regional representation is focusing on internal properties, such as texture and skeletal shape. In plant species identification using digital morphometrics, image representation is done by using leaf shape analysis. The [23] have made a review of previous methods used to analyze the leaf shape using three ways: two-dimensional outline shape of leaf petal, the structure of the vein network and the characters of leaf margin. The two-dimensional outline shape of leaf petal is a boundary representation while the structure of the vein network and the characters of leaf margin are regional representation. Some research examples using leaf shape analysis is conducted by [6], [24], [25], [26] and [27]. A method must be specified for describing the data so that features of interest are highlighted.

Description, also called as feature selection, deals with extracting attributes that result in some quantitative information of interest. For example using content based image retrieval, the length and width of leaf in pixel, and the area of leaf in pixel^2 are three feature selections that are gained from image representation phase. These descriptors are then used in classification in order to find the distance or similarity with the descriptors stored in database.

1.5 Image Recognition

Recognition is the process that assigns a label to an object based on information provided by its descriptors. Classification is a usual process used to recognize image. Classification is needed to distinguish a plant species with other species based on the data obtained from feature selection. The descriptors from the image data stored in database are compared with the descriptors from the query image. The closer gap within those descriptors is then chosen to appoint the query image to be in which class. Artificial neural network (ANN) and fuzzy logic are the most commonly techniques used in classification. Some previous works on agriculture image processing using fuzzy classifier refer to [28] and [29]. Some previous works agriculture image processing using ANN classifier, refer to [20], [30], [18] and [17].

IV. IMAGE PROCESSING OPERATIONS

Image processing operations can be divided into two levels [31]:

1) Low level image processing
2) High level image processing

A) Low Level Image Processing

The low level is also called image pre-processing that operates at the pixel level [32]. The input to low level image processing operators is an image whereas the output is either image or data. Few examples of low level image processing operators are contrast enhancement, noise reduction, and noise removal in an image. They are also used for edge detection and various image transformations or calculate simple characteristics such as contours histograms. Low level image processing operators can be classified as point operators, neighborhood operators and global operators, with respect to the way the output pixels are determined from the input pixels [32].

B) High Level Image Processing

The high level image processing operations operate in order to generate higher abstractions [31]. They work on abstractions derived from intermediate-level image processing operators. They are used to interpret the image content such as classification and object recognition. These operations work on graphs, lists and relations among regions/objects to derive some decision.

Based on Fig. 4, image enhancement and image segmentation are situated in low level operation while image representation, feature selection and image interpretation are situated in high level operation. Most of image processing for agriculture application is a combination of low level and high level operations. This is because most of the system is made for object recognition purpose.

V. THE INTERRELATED ISSUES IN HPC AND IMAGE PROCESSING IN AGRICULTURE APPLICATION

The success factor of HPC for solving real-life high workload problems relies on the processors architecture and design. In other words, communication time between processors depends on hardware parameters. The performance of HPC model can be evaluated by calculating its speedup. Speedup shows how much of an improvement is practically possible in the best case. Assume that the speed of processors and network is constant; the speedup of k processor(s) is calculated using the following equation [33]:

$$S(k) = \frac{T_1}{T_k} \qquad (1)$$

where T_1 is the time required to execute an equivalent sequential program on one processor and T_k is the time required to execute the parallel version of the program on k processors. On the other hand, accuracy performance is calculated based on classification accuracy. There are many ways to measures performance for classification. [34] have made an experimental comparison of performance measures for classification using 18 ways of measure. The most common and simplest measure is to evaluate a classifier by calculating its accuracy that defined as the degree of right predictions of model (or conversely, the percentage of misclassification errors) [34]. In this technique, correct accuracy is calculated by adding all true prediction and then divides them by all predictions that have made while misclassification errors are calculated by adding all false prediction and then divides them by all predictions that have made. Most of image processing systems in agriculture application that was summarized in Table I is more concerned on accuracy performance rather than speed performance. The speed performance is never been discussed in any examples shown in Table I. However, with the existence of high performance computing in recent years, it is not practical if the image processing system does not fully utilize the capabilities offered by HPC.

In the last decade, there has been tremendous growth of plant species data and extensive computing for plant classification. Thus, many large image-bases (database of images) exist in plant database. High-speed image processing on image classification is really challenging and complex. Some examples for this application are classification of bamboo species [5], classification of cotton plant to identify disease [3], classification of plant leaf images with complicated background [35], plant species classification based on leaf shape [26], and many more. Real-time image processing is employed in purpose for industrial applications such as dates inspection [36] and [37], barberry quality inspection [29] and orange grading [38]. In real-time image processing, the processing should be done with high speed on rapid sequence of images or on a single image.

Looking at the architecture of standalone computer, it is found that standalone computer has its own limitation in speed performance and memory beyond which it cannot be extended in present architecture. Moreover, the computation demand is always found ahead of computational performance. The current computer hardware technology has reached at its limit as no substantial improvement in CPU speed has been observed since last few years. As more sophisticated applications are demanded by users, less processing time and faster response are required from applications. However, the problem of time and space complexity can be solved with the great advantages offered in parallel computing techniques. Furthermore, the independence of recognition process tasks and databases makes parallel processing platform for image recognition in plant is very appropriate.

VI. Parallel Processing for Image Recognition in Agriculture Application

To enhance the image processing system, some important steps need to be carried out [7]:

1) The image processing operations need to be identified in order to analyze parallelism.
2) Approach of parallelism in image processing applications has to be applied using three ways data parallel, task parallel and pipeline parallel.
3) The architecture for achieving parallelism has to be identified whether to use parallel processing hardware or distributed system.

These methods are done to ensure no data loss occurs during the data delivery process. In addition, it also to ensure the maximum algorithm processing time can be achieved. Parallelism can be achieved by processing simultaneously using two or more processing units. Meanwhile, in agriculture image recognition application the parallelism can be applied using three methods that are data parallel, task parallel and pipeline parallel [7].

A) Data Parallel

In data parallel approach, data will be divided into several computing units. Among the challenges in data parallel approach is to produce an efficient parallel execution. Two key issues to be addressed are data locality and load balancing. Image data must be divided to all available PUs therefore the communication between PUs that is not necessary can be reduced. Each of the image data portion shall be distributed to each PU so that each PU will get the same load balancing. Fig. 5 shows division of an image to eight portions. This example is generated based on [18] to predict leaf population chlorophyll content from cotton plant image.

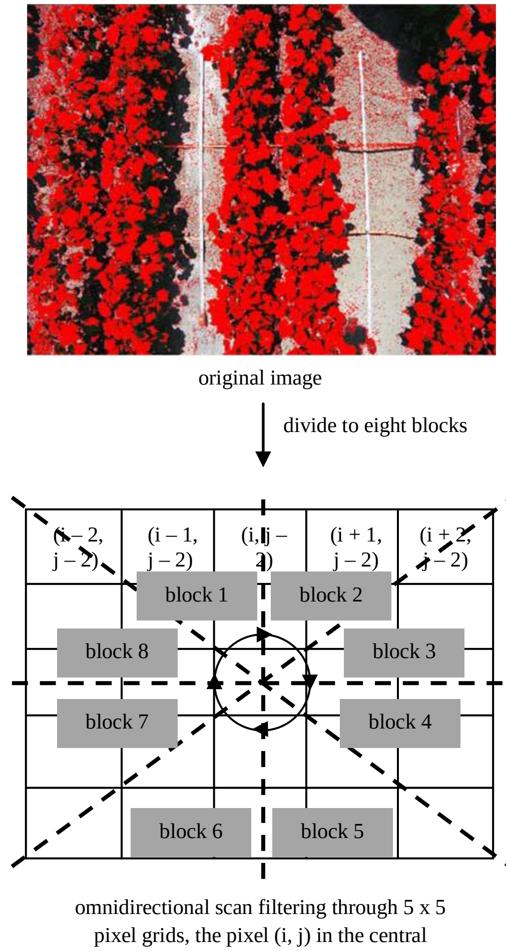
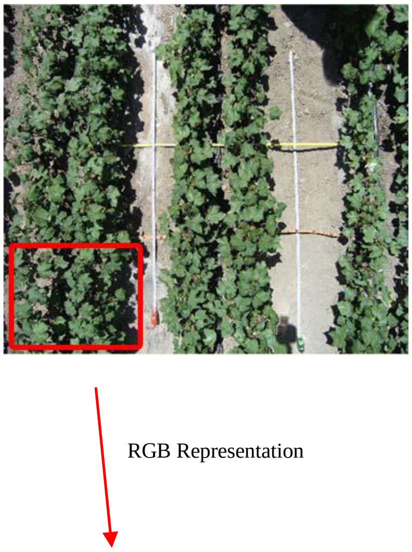


Fig. 5. Division of an image into eight blocks for parallel processing [18]

The omnidirectional scan algorithm was introduced by [18] to remove the separated "salt and pepper" noises (image preprocessing stage). The original image is divided into eight different blocks and each block has six pixel points. [18] produced the algorithm that scan the noises in sequential process (one pixel to one pixel) in omnidirectional (begin in block 1 and ended in block 8). The process will continued again and again until all pixels in plant image have been exhausted. This shows that it needs a huge workload (many iterations) to scan all pixels. The workload becomes larger when the size of pixels is larger. In order to make this scanning process faster, data parallel approach is needed. In this approach, each block with almost the same size (load)



1 R = 124 G = 153 B = 122	2 R = 88 G = 112 B = 83	3 R = 126 G = 157 B = 126
4 R = 73 G = 102 B = 64	5 R = 116 G = 149 B = 114	6 R = 107 G = 142 B = 93
7 R = 119 G = 149 B = 114	8 R = 94 G = 125 B = 93	9 R = 57 G = 76 B = 51

Fig.. 6. An original image of cotton plant represent in RGB

will be distributed to each PU. Processor 1 scans the noises in the block 1 while the noises in block 2 are scanned by processor 2 and so on. The optimum processing time will be achieved if the same load to each processor is distributed. However, the distribution of blocks of image is limited to parallel computers, which have multiple PUs.

B) Task Parallel

In the task parallel approach, the step in image processing is a task and each of the tasks will be assigned to different PUs. There are various types of operation in image processing for agriculture. Some of these operations are independent whereby result of any operation does not depending on the result of other operations and vice versa. The major challenge in task parallel approach is to determine efficient task decomposition and result composition. Fig. 6 shows an original captured cotton plant image whereby the red square area (3 x 3 pixels) is represented in red-green-blue (RGB). This example is also taken from [18]. The objective of this project is to predict leaf population chlorophyll content from cotton plant image.

In image preprocessing stage, [18] have converted RGB color mode of an original captured cotton plant image to intensity-hue-saturation (IHS) color mode. The IHS color

TABLE III

SUITABLE MECHANISMS FOR DIFFERENT APPROACHES

Approach of Parallelism	Suitable Mechanisms
Data Parallel	Parallel processing hardware
Task Parallel	Parallel processing hardware and/or distributed system
Pipeline Parallel	Distributed system

values are obtained from the RGB values by the following conversion equations [18]:

$$I = \frac{(R+G+B)}{3} \quad (2)$$

$$S = 1 - \frac{3}{R+G+B}[\min(R,G,B)] \quad (3)$$

$$W = \cos^{-1}\left\{ \frac{\frac{1}{2}[(R-G)+(R-B)]}{\sqrt{(R-G)^2+(R-B)(G-B)}} \right\} \quad (4)$$

$$H = W \quad \text{if } B \le G$$

$$H = 2\pi - W \quad \text{if } B \ge G$$

Referring to all above equations (2)–(3), the values of *I*, *S* and *H* are all depending on the values of *R*, *G* and *B*. Meanwhile, (3) is not depending on (2) and (4) is also not depending on (3). By looking at this situation, task parallel can be done to this process whereby equation 1 (task 1) is calculated by processor 1, equation 2 (task 2) is calculated by processor 2 and equation 3 (task 3) is calculated by processor 3.

Fig. 7 shows a comparison of serial vs. parallel execution for three pixels. According to Fig. 7, nine steps are needed for serial execution while only three steps are needed in parallel execution. By using this method, six steps can be reduced just only on three pixels calculation. The bigger pixels calculation will give bigger effect on speed performance.

C) Pipeline Parallel

There are also image processing in agriculture application that have consecutive stages. If a system requires multiple images or data to be processed, then pipeline parallel approach can be done. This approach is suitable to be used when many data to be processed at one time. Fig. 8 shows a simple neural network diagram that can be used to represent the parallelism of image processing in the pipeline model. By using this example, if users query to identify plant species, the descriptors from query image (I1, I2 and I3) will be compared (in distance or similarity) with the descriptors from image data in database (H1, H2 and H3) to obtain output A, B and C. Let say the connections from input layer to H1 which are I1 to HI, I2 to H1 and I3 to H1 are task 1, the connections from input layer to H2 which are I1 to H2, I2 to H2 and I3 to H2 are task 2, and the connections from input layer to H3 which are I1 to H3, I2 to H3 and I3 to H3 are task 3. Each of these data should be sent to PUs for comparing process. Each

Serial Execution			Time	Parallel Execution				
Pixel	PU 1	Step		Pixel	PU 1	PU 2	PU 3	Step
1	Calculate *I*	1	t1	1	Calculate *I*	Calculate *I*	Calculate *I*	1
	Calculate *S*	2	t2	2	Calculate *S*	Calculate *S*	Calculate *S*	2
	Calculate *H*	3	t3	3	Calculate *H*	Calculate *H*	Calculate *H*	3
2	Calculate *I*	4	t4				No. of steps	3
	Calculate *S*	5	t5					
	Calculate *H*	6	t6					
3	Calculate *I*	7	t7					
	Calculate *S*	8	t8					
	Calculate *H*	9	t9					
	No. of steps	9						

Fig. 7. Comparison of serial vs. parallel execution

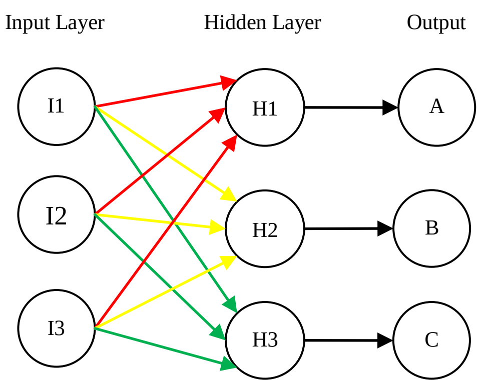


Fig. 8. A simple neural network

Time	Task1 on PU1	Task2 on PU2	Task3 on PU3
t1	(I1 – H1)	(I3 – H2)	(I2 – H3)
t2	(I2 – H1)	(I1 – H2)	(I1 – H3)
t3	(I3 – H1)	(I2 – H2)	(I3 – H3)

Fig. 9. Pipeline parallel image processing

of the data shall be distributed to each PU so that each PU will get the same load balancing. In pipeline processing, images data will be in different tasks at the same time. Fig. 9 shows the idea using pipeline parallel based on the concept of neural network.

This approach is possible to be done in classification process in descriptors matching phase as shown in the above idea. By using non-parametric classifier such as ANN, it consists of a series of layers. High computational rate achieved by their massive parallelism resulting from an arrangement of interconnections (weights) and simple processors (neurons) that permits real-time processing of very large datasets. This approach eliminates computationally demanding and requires more time for processing. For each approach of parallelism in image processing, the suitable mechanisms for efficient parallel processing are summarized in Table III.

VII. Conclusion

The evolution of parallel processing hardware and distributed system is rapidly increased time by time. With the existence of an advance parallel processing hardware and distributed system, the image processing algorithm should be enhanced to fully utilize the capabilities of these tools. Therefore, a deeper understanding of how to achieve parallelism is important. Based on the analysis that has been made in producing parallel image processing, we can conclude the parallelism is also possible for image processing in agriculture application. HPC is possible to be implemented in all steps in image processing starting from preprocessing to classification process. This study is also can help to provide some understanding in parallel and distributed computing area.

The proposed future work is to enhance the previous work on agriculture image processing algorithm. One of the main aims for doing this is to confirm the theoretical approach on achieving parallelism that has been discussed. Furthermore, the research work also can help to identify other challenges in parallel and distributed computing such as scheduling of the tasks on PUs. In addition, the research work on an object oriented programming approach for image processing and parallel processing are also important. The use of Java language is essential for image processing since it has an easy and faster way to develop graphical user interface (GUI) [39]. The development of GUI is a must for image processing in order to make it more users friendly. Besides, Java is also an object oriented programming language and offers a comfortable environment to be implemented in HPC. Since late 1990s, there are many active projects [40], [14], [33] and [41] underway whose aim to analyze the feasibility of Java for computational tasks associated with HPC.

References

[1] J. P. Kerr and E. B. Bartlett, "Medical image processing utilizing neural networks trained on a massively parallel computer," *Computers in Biology and Medicine*, vol. 25, no. 4, pp. 393–403, Feb. 1995.

[2] G. E. Christensen, "MIMD vs. SIMD parallel processing: a case study in 3D medical image registration," *Parallel Computing – Special double issue on biomedical applications,* vol. 24, no. 9, pp. 1369–1383, Sept. 1998.

[3] A. Camargo and J.S. Smith, "Image pattern classification for the identification of disease causing agents in plants," *Computers and Electronis in Agriculture,* vol. 66, no. 2, pp. 121–125, Feb. 2009.

[4] A. Meunkaewjinda, P. Kumsawat, K. Attakitmongcol, and A. Srikaew, "Grape leaf disease detection from color imagery using hybrid intelligent system," in *2008 Proc. of ECTI-CON,* pp. 513–516.

[5] K. Singh, I. Gupta, and S. Gupta, S, "Classification of bamboo plant based on digital image processing by central moment," in 2011 *Int. Conf. on Image Information Processing.*

[6] J. C. Neto, G. E. Meyer, D. D. Jones, and A. K. Samal, "Plant species identification using elliptic Fourier leaf shape analysis," *Computers and Electronics in Agriculture,* vol. 50, no. 2, pp. 121–134, Feb. 2006.

[7] H. B. Prajapati and S. K. Vij, "Analytical study of parallel and distributed image processing," in *2011 Int. Conf. on Image Information Processing,* pp. 1–6.

[8] A. Wachsmann, "Concepts of parallel computing: clustering and shared memory," unpublished.

[9] M. J. Flynn and K. W. Rudd, "Parallel architectures," *ACM Computing Surveys,* vol. 28, no. 1, pp. 67–70, Mar. 1996.

[10] B. Barney. (2010, May 10). Introduction to parallel computing. [Online]. Available: https://computing.llnl.gov/tutorials/parallel_comp/

[11] T. Mattson and R. Eigenmann. (1999). *OpenMP: an API for writing portable SMP application software.* [Online]. Available: http://www.openmp.org/presentations/sc99/sc99_tutorial.pdf

[12] W. Gropp, E. Lusk, and A. Skjellum, *Using MPI: Portable Parallel Programming with the Message-Passing Interface.* 1st ed. Cambridge, MA, USA: MIT Press, 1999.

[13] G. Sharma and J. Martin, "Matlab: A language for parallel computing," *International Journal of Parallel Computing,* vol. 37, no. 1, pp. 3–36, 2009.

[14] M. Baker, B. Carpenter, and A. Shafi, "Works in Progress: A pluggable architecture for high-performance Java messaging," *IEEE Distributed Systems Online*, vol. 6, no. 10, 2005.

[15] S. Annadurai and R. Shanmugalakshmi, *Fundamentals of Digital Image Processing.* India: Pearson Education, 2007, pp. 3–10.

[16] X. Pengyun and L. Jigang, "Computer assistance image processing spores counting," in *2009 Proc. Int. Asia Conf. on Informatics in Control, Automation and Robotics,* pp. 203–206.

[17] C. C. Yang, S. O. Prasher, J. A. Landry, H. S. Ramaswamy, and A. Ditommaso, "Application of artificial neural networks in image recognition and classification of crop and weeds," *Canadian Agricultural Engineering,* vol. 42, no. 3, pp. 147 – 152, 2000.

[18] X.-m. Suo, Y.-t. Jiang, M. Yang, S.-k. Li, K.-r. Wang, and C.-t. Wang, "Artificial neural network to predict leaf population chlorophyll content from cotton plant images," *Agricultural Sciences in China,* vol. 9, no. 1, pp. 38–45, Jan. 2010.

[19] S. Phadikar and J. Sil, "Rice disease identification using pattern recognition techniques," in *Proc. 11th Int. Conf. on Computer and Information Technology,* 2008, Khulna, Bangladesh, pp. 420–423.

[20] J. I. Arribas, G. V. Sanchez-Ferrero, G. Ruiz-Ruiz, and J. Gomez-Gil, "Leaf classification in sunflower crops by computer vision and neural networks," *Computers and Electronics in Agriculture,* vol. 78, no.1, pp. 9–18, Aug. 2011.

[21] Y. Chen and X. Zhou, "Plant root image processing and analysis based on 2D scanner," in *Proc. IEEE 5th Int. Conf. on Bio-Inspired Computing: Theories and Application,* 2010, Ghangsha, China, pp. 1216–1220.

[22] S. Thaiparnit and J. Jakkree Srinonchat, "Apply image processing technique to determine the correlation chlorophyll," presented at the 6th Int. Conf. on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology, Thailand, May. 6–9, 2009, pp. 1116–1119.

[23] J. S. Cope, D. Corney, J. Y. Clark, P. Remagnino, and P. Wilkin, "Plant species identification using digital morphometrics: a review," *Experts Systems with Applications,* vol. 39, pp. 7562–7573, 2012.

[24] Z. Wang, Z. Chi, and D. Feng, "Shape based leaf image retrieval," *Vision, Image & Signal Processing,* vol. 150, no. 1, pp. 34–43, Feb. 2003.

[25] Y. Nam, E. Hwang, and D. Kim, "A similarity-based leaf image retrieval scheme: joining shape and venation features," *Computer Vision and Image Understanding,* vol. 110, no. 2, pp. 245–249, May 2008.

[26] J.-X. Du, X.-F. Wang, and G.-J. Zhang, "Leaf shape based plant species recognition," *Applied Mathematics and Computation,* vol. 185, no. 2, pp. 883–893, Feb. 2007.

[27] O. M. Bruno, R. d. O. Plotze, R. M. Falvo, and M. d. Castro, "Fractal dimension applied to plant identification," *Information Sciences,* vol. 178, no. 12, pp. 2722–2733, June 2008.

[28] R. Fageth, W. G. Allen, U. and Jager, U. "Fuzzy logic classification in image processing," *Fuzzy Sets and Systems,* vol. 82, no. 3, pp. 265–278, Sept. 1996.

[29] H. Shirgahi and N. Danesh, "Presented a fuzzy system to determine barberry product quality based on average color spectra by image processing," *Journal of King Saud University,* vol. 23, no. 1, pp. 105– 109, July 2010.

[30] S. G. Wu, F. S. Bao, E. Y. Xu, Y-X. Wang, Y.-F. Chang, and Q.-L. Xiang, "A leaf recognition algorithm for plant classification using probabilistic neural network," presented at the IEEE International Symposium on Signal Processing and Information Technology, Cairo, Egypt, Dec. 15–18, 2007, pp. 11–16.

[31] T. Braunl, S. Feyrer, W. Rapf, and M. Reinhardt, *Parallel Image Processing*, 6th ed. New York: Springer, 2001, pp. 1–2.

[32] C. Nicolescu and P. Jonker, "Parallel low-level image processing on distributed-memory system," in *2000 Proc. IPDPS Workshops on Parallel and Distributed Processing,* pp. 226–233.

[33] M. Baker, B. Carpenter, and A. Shafi, "MPJ Express: Towards thread safe Java HPC," presented at the *IEEE Int. Conf. on Cluster Computing*, Barcelona, Spain, Sep. 25–28, 2006.

[34] C. Ferri, J. Hernandez-Orallo, and R. Modroiu, "An experimental comparison of performance measures for classification," *Pattern Recognition Letters,* vol. 30, no. 1, pp. 27–38, Jan. 2009.

[35] X.-F. Wang, D.-S. Huang, J.-X. Du, H. Hu, and L. Heutte, "Classification of plant leaf images with complicated background," *Applied Mathematics and Computation,* vol. 205, no. 2, pp. 916–926, Nov. 2008.

[36] A. A. Al-Janobi, "Date inspection by color machine vision," *Journal of King Saud University,* vol. 12, no. 1, pp. 66–79, 2000.

[37] Y. Al-Ohali, "Computer vision based date fruit grading system: design and implementation," *Journal of King Saud University,* vol. 23, pp. 29–36, 2011.

[38] R. Fellegari and H. Navid, "Determining the orange volume using image processing," in *2011 Int. Conf. on Food Engineering and Biotechnology,* pp. 180–184.

[39] E. Koffman and U. Wolz, "A simple Java package for GUI-like interactivity," in *2001 Proc. of the Third-Second SIGCSE Technical Symposium on Computer Science Education,* pp. 11–15.

[40] V. Getov, G. V. Laszewski, M. Philippsen, and I. Foster, "Multiparadigm communications in Java for grid computing," *Communications of the ACM,* vol. 44, no. 10, pp. 118–125, Oct. 2001.

[41] N. Mohamed, J. Al-Jaroodi, H. Jiang, and D. Swanson, "JOPI: A Java object-passing interface," *Concurrency and Computation: Practice & Experience,* vol. 17, no. 7–8, pp. 775–795, June 2005.