Australian Journal of Basic and Applied Sciences, 8(9) June 2014, Pages: 38-44

AENSI Journals

Australian Journal of Basic and Applied Sciences

ISSN:1991-8178

Journal home page: www.ajbasweb.com

Distributed visual enhancement on surveillance video with Hadoop Mapreduce and performance evaluation in pseudo distributed mode

Mohd Fikri Azli Abdullah, Md. Shohel Sayeed, Anang Hudaya Muhammad Amin, Nazrul Muhaimin Ahmad, Ibrahim Yusof, Liew Tze Hui, Housam Khalifa Bashier

Faculty of Information Science and Technology, Multimedia University, Jalan Ayer Keroh Lama, 75450 Melaka, Malaysia

ARTICLE INFO

Article history:
Received 2 March 2014
Received in revised form
13 May 2014
Accepted 28 May 2014
Available online 23 June 2014

Keywords:
Distributed, visual enhancement, histogram equalization, Hadoop MapReduce

ABSTRACT

Background: A large amount of surveillance video is generated due to the improvement in camera performance and the low-priced light-weighted cameras availability. Also, there are limits to the single computer to process large-scale data, such as video analysis. Thus, the advantages of parallel distributed processing of a video database by using the computational resources of a cloud computing environment should be considered. **Objective:** This work implements the distributed visual enhancement using histogram equalization algorithm on image database from surveillance cameras. The experiment is conducted in pseudo distributed mode under Hadoop MapReduce architecture. The database images were processed with different number of map tasks and the performance is analysed. **Results:** The result shows that the increases of map tasks will increase the total time because the processing time of histogram equalization is very small. Main part of the total time is contributed by the transferring time of images among map tasks. **Conclusion:** We argue that the distributed infrastructure implemented with Hadoop MapReduce is suitable for more complex problems which require higher processing capability.

To Cite This Article: Mohd Fikri Azli Abdullah, Md. Shohel Sayeed, Anang Hudaya Muhammad Amin, Nazrul Muhaimin Ahmad, Ibrahim Yusof, Liew Tze Hui, Housam Khalifa Bashier, Distributed Visual Enhancement on Surveillance Video with Hadoop Mapreduce and Performance Evaluation in Pseudo Distributed Mode. ***Aust. J. Basic & Appl. Sci.,*** *8(9): 38-44, 2014*

INTRODUCTION

Recently, with the rapid development of information technology and social stability needs, many important sites are equipped with video surveillance equipment, which for people in dealing with security incidents provides the most direct evidence of the scene. People can use these data to achieve real-time video tracking, or based on the video data in real time or after the events of all kinds to make the most direct and fair ruling. Many of the surveillance video are generally in low resolution and view a large area. As a result the images which are captured are often, for example, out of focus and below the resolution required to read a licence plate from a car, or to allow criminal to be readily identified. This situation results in a need for the pre-processing of these surveillance videos to enhance the video. However, in multi-camera surveillance system, it will produce large amounts of video information, and require a very enormous computation, so that the whole system performance of less than real-time requirements.

Also, when considering operations such as search and other types of analysis of video images recorded by video camera and stored in a database, there are limits on what can be done to improve the performance of single computers to make them able to process large-scale information. Therefore, the advantages of parallel distributed processing of a video database by using the computational resources of a cloud computing environment should be considered. Cloud computing consists of high performance computing and mass storage has unparalleled advantages, so the use of cloud computing can solve such problems. It is a new method of shared infrastructure, built on top of a large-scale clusters of cheap servers, through infrastructure and applications together to build the upper to maximum efficient use of hardware resources. The huge combined system pool provides a variety IT services, which is of the high scalability and high reliability.

Visual image enhancement algorithms have been practical for many applications in consumer, autonomous navigation, remote sensing, biomedical image analysis, and other image processing fields (Panetta *et al*., 2008a; Panetta, Zhou, *et al*., 2011; DelMarco and Agaian, 2009). As there is no single set of criteria which can universally define an ideal enhancement for all circumstances, many image enhancement techniques have been proposed.

Corresponding Author: Mohd Fikri Azli Abdullah, Faculty of Information Science and Technology, Multimedia University, Jalan Ayer Keroh Lama, 75450 Melaka, Malaysia.
Tel: +60 6-252 3381 E-mail: mfikriazli.abdullah@mmu.edu.my

Australian Journal of Basic and Applied Sciences, 8(9) June 2014, Pages: 38-44

Image enhancement algorithms can broadly be classified as either direct or indirect enhancement procedures. Indirect enhancement algorithms enhance images without explicitly defining and measuring image contrast. Such algorithms include histogram equalization (HE) and its variants (Lu *et al*., 2010; Lim *et al*., 2013; Mohan and Ravishankar, 2013) basic pixel transformations, and contrast stretching operations (Nimkar *et al*., 2013; Celik and Tjahjadi, 2012). Many of the indirect enhancement algorithms simply define a global or adaptive one-to-one mapping by which the intensity values of individual pixels are modified. Because these approaches do not directly measure image contrast, they can often times yield inadequate detail preservation or over-enhancement.

Conversely, direct enhancement algorithms have been developed based on the fact that the human visual system (HVS) is adapted to extract local structural information. Direct enhancement approaches, thus, quantitatively define a contrast measure in either a spatial or transform domain, and achieve enhancement by magnifying the measured contrast (Sengee *et al*., 2010). Accordingly, linear and non-linear means of amplifying this measured contrast have been proposed. The most basic of these algorithms is the well known un-sharp masking algorithm and its multi-scale extensions. The particular advantage of the multi-scale direct enhancement algorithms, such as the pyramidal and wavelet transform-based approaches, is that image contrast at many different scales can be measured and subsequently enhanced.

Since the HVS is an excellent image processor capable of detecting and recognizing image information, it is only natural to bridge the gap between these psychophysical attributes and the way in which images are represented and manipulated (Panetta, Zhou, *et al*., 2011; Panetta, Agaian, *et al*., 2011). The effectiveness of a direct enhancement procedure is thus based on the formulation of a suitable contrast measure which is consistent with the psycho-visual laws of the HVS. Many existing direct multi-scale enhancement algorithms are based on the mapping of detail transform coefficients, in which case the multi-scale contrast is subsequently defined in terms of absolute luminance changes. While the detail sub-bands of the pyramidal and wavelet transforms are proportional to image contrast, it is well known that for a large range of background intensities, the HVS is sensitive to relative, and not absolute, luminance changes (Yang *et al*., 2005). This phenomenon is known as the luminance masking (LM) characteristic of the HVS. Direct contrast enhancement algorithms integrating this masking effect have been proposed in the discrete wavelet transform (Tang, Liu, *et al*., 2009; DelMarco and Agaian, 2009; Xia *et al*., 2010) and Laplacian pyramid transform (Liu *et al*., 2009) domains. These algorithms were shown to perform particularly well in the enhancement of radiographies, and significantly improved the detection of breast cancer in mammograms (Tang, Rangayyan, *et al*., 2009). Although the contrast measures used in these algorithms are more consistent with the HVS than the use of detail coefficients alone, only linear mappings of these LM measures of contrast were proposed.

Another important characteristic of the HVS is that it is sensitive to relative changes in contrast. This is to say that the response of the HVS to local contrast is highly affected by its surrounding stimuli (Yang *et al*., 2005). This phenomenon is known as the contrast masking (CM) feature of the HVS. Single-scale (just-noticeable difference) JND models accounting for both the LM and CM characteristics have been considered for compression and watermarking applications (Liu *et al*., 2010).

Moreover, a single-scale parametric edge detection algorithm which includes both the LM and CM features of the HVS was shown to dramatically improve the performance of the standard Canny edge detector (Yang *et al*., 2005; Panetta *et al*., 2008b; Panetta, Agaian, *et al*., 2011). Given the relevance of the HVS phenomena, as well as the many advantages and the widespread use of multi-scale image processing tools, it is instructive to consider a way of combining the two in order to enable an integration of ad hoc and statistical multi structures. Therefore, it is advantageous to integrate the non-linear mappings, which have to date only been used on detail coefficients alone.

The rest of this paper is organized as follows. Section 2 provides the background knowledge related to this work, including an overview of the Hadoop and MapReduce programming. In section 3, we discuss the architecture of our system for processing video databases by using MapReduce on Hadoop in a distributed environment. Then, we give an overview on our experimental methodology and present the results of our experiments in section 4. Finally, we conclude the paper and propose future work in section 5.

Hadoop streaming for video processing:

A. Parallel and distributed processing on Hadoop:

Hadoop consists of two components, the Hadoop Distributed File System (HDFS) and MapReduce, performing distributed processing by single-master and multiple-slave servers. There are two elements of MapReduce, namely JobTracker and TaskTracker, and two elements of HDFS, namely DataNode and NameNode. In Fig 1, the configuration of these elements of MapReduce and HDFS on Hadoop are indicated.

Mapreduce engine consists of one JobTracker, to which client applications submit MapReduce jobs. The JobTracker pushes work out to available TaskTracker nodes in the cluster, striving to keep the work as close to the data as possible. With a rack-aware file system, the JobTracker knows which node contains the data, and which other machines are nearby. If the work cannot be hosted on the actual node where the data resides, priority is given to nodes in the same rack. This reduces network traffic on the main backbone network. If a TaskTracker fails or times out, that part of the job is rescheduled.

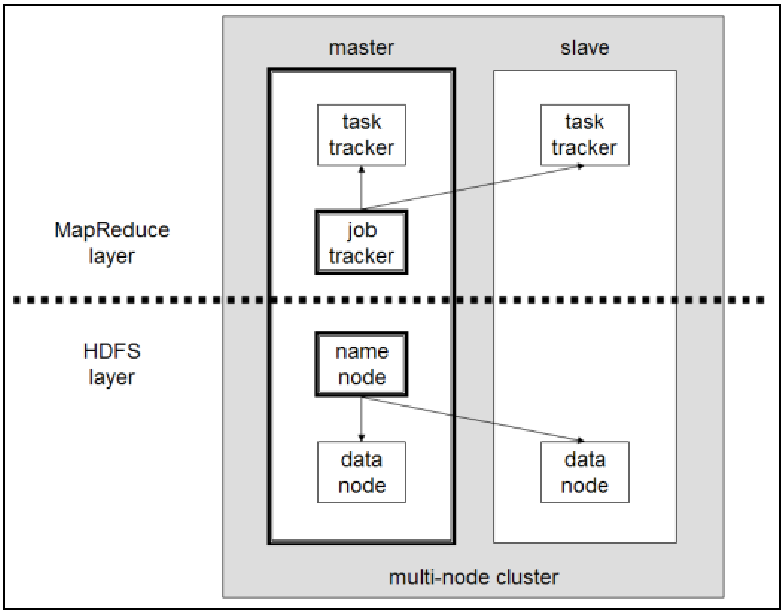

Fig. 1: A multi-node Hadoop structure.

The Hadoop distributed file system (HDFS) is a distributed, scalable, and portable file-system written in Java for the Hadoop framework. Each node in a Hadoop instance typically has a single NameNode; a cluster of DataNodes form the HDFS cluster. The situation is typical because each node does not require a DataNode to be present. Each DataNode serves up blocks of data over the network using a block protocol specific to HDFS. The file system uses the TCP/IP layer for communication. Clients use Remote procedure call (RPC) to communicate between each other. HDFS stores large files (typically in the range of gigabytes to terabytes) across multiple machines. It achieves reliability by replicating the data across multiple hosts.

Hadoop can be run in three modes which are standalone, pseudo-distributed and distributed mode. By default, Hadoop is configured to run in standalone mode. It is non-distributed mode and run using a single Java process. Standalone mode is very useful for debugging. In pseudo-distributed mode, Hadoop is also run on a single node but each Hadoop daemon runs in a separate Java process. The last mode, distributed mode, runs Hadoop with multi node cluster.

B. MapReduce:

MapReduce is a programming model for processing large data sets with a parallel, distributed algorithm on a cluster. A MapReduce program is composed of a Map() procedure that performs filtering and sorting and a Reduce() procedure that performs a summary operation. The MapReduce framework organizing the distributed servers, running the various tasks in parallel, managing all communications and data transfers between the various parts of the system, and providing for redundancy and fault tolerance.

C. Hadoop streaming:

Hadoop streaming is a utility in the Hadoop distribution. The utility allows users to create and run Map/Reduce jobs with any executable or script that are not written in Java (such as Ruby, Perl, Python, PHP, R, or C++) as the Map() or the Reduce() procedure. Using this utility, we can execute the histogram equalization algorithm written in Python programming language. Fig 2 shows a general example of an execution of a program written in Python using Hadoop streaming.

```
$HADOOP_HOME/bin/Hadoop jar $HADOOP_HOME/hadoop-streaming.jar \
    -input myInputDirs \
    -output myOutputDir \
    -mapper myMapper.py \
    -reducer myReducer.py
```

Fig. 2: A general example of Hadoop streaming execution.

Map Reduce for Video Processing:

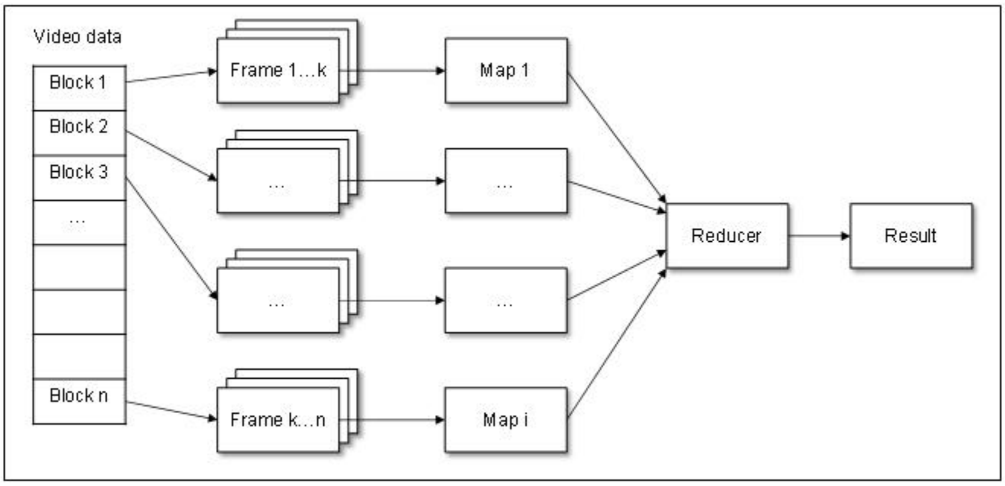

Fig 3: Video processing using Hadoop MapReduce

Video database processing is performed by splitting the data in a video database and creating key-value pairs. For example, the frame number can be used as a key for a video frame. In the case of parallel processing of a video frame, the video frame is divided into multiple block, and the block numbers can be the keys (identifiers) for these different parts. Sorting is carried out using the key number, and joining separated frames or separated parts is performed by the Reduce function. Fig 3 shows the proposed framework for the video processing using Hadoop MapReduce. In the figure, the video data is divided into *n* blocks. Each block is processed by one map task. The outputs of all map tasks are passed to the Reduce function. Lastly Reduce function joins the separated frames into a video file.

Experiment of image enhancement using Map Reduce:

The experiment uses the SCface (Surveillance Cameras Face) database that contains static images of human faces. Images were taken in uncontrolled indoor environment using five video surveillance cameras of various qualities. Database contains 4160 static images (in visible and infrared spectrum) of 130 subjects. Images from different quality cameras mimic the real-world conditions and enable robust face recognition algorithms testing, emphasizing different law enforcement and surveillance use case scenarios. This experiment considers only 2860 images of the frontal face images from surveillance and infrared cameras.

The experiment is conducted using virtual machine with the Ubuntu 12.04 as the operating system. The processor of the machine is Intel Core i3 3.30GHz with 1GB of memory. First, the MapReduce system is configured with a pseudo-distributed mode. Hadoop is composed of a master server which manages slave servers, which perform the actual image processing. Master and slave servers actually run on the same server for this configuration of the MapReduce system (pseudo-distributed mode). The number of copies of image data is set to one. The parallel processing of the image database using Hadoop Streaming is distributed over all of the map tasks in a single machine. Then all the images from the database are loaded into the HDFS of the MapReduce system. Next the program of visual enhancement is run using MapReduce programming model. The program contains histogram equalization algorithms and was written in python programming language. The time consumed for running the visual enhancement program with different number of images and different number of map tasks (between 3 until 10) was recorded. The images will be divided and processed across all map tasks which is handled by MapReduce as in Fig 3. The reducer task is to collect all the images after the visual enhancement process. The experiment is repeated for five times to get the average performance times.

RESULTS AND DISCUSSION

The processing time for the distributed visual enhancement over images is shown in Fig 4. The graph shows the performance with different amount of images. The processing time is slightly increases when the amount of images increases. The system needs slightly longer duration to process larger amount of images. However, since the processing is distributed among map tasks, the increase of time duration for each map tasks is very small.

Fig 5 shows the experiment result with different number of map tasks. The experiment is conducted with the number of map tasks between 3 and 10. The graph shows that the processing time is increases even though the number of map tasks is increases. It is expected to be decreased since the number of map tasks in increases.

It is because the amount of time needed to transfer the images among map tasks is higher than the time to process each image using histogram equalization algorithm. Thus the processing time is mainly contributed by the transferring time.

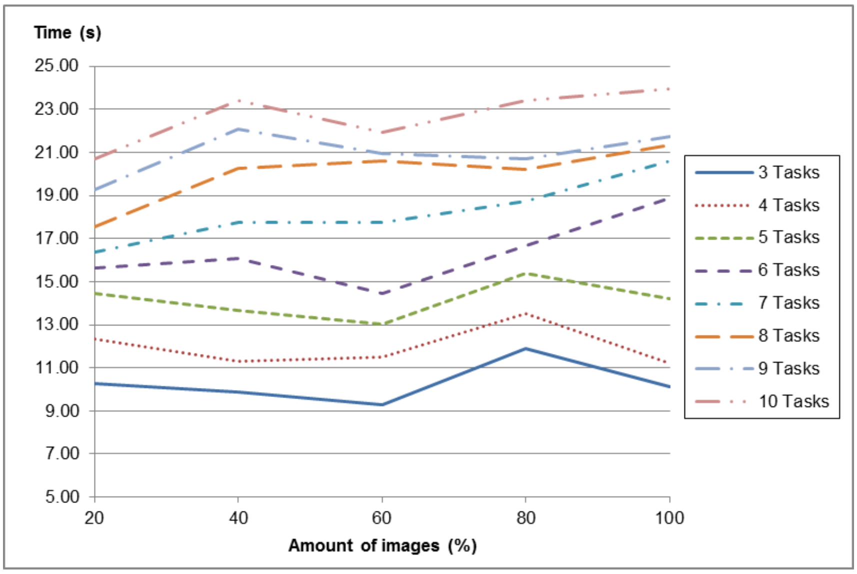

Fig. 4: Processing time of different amount of images with different number of map tasks.

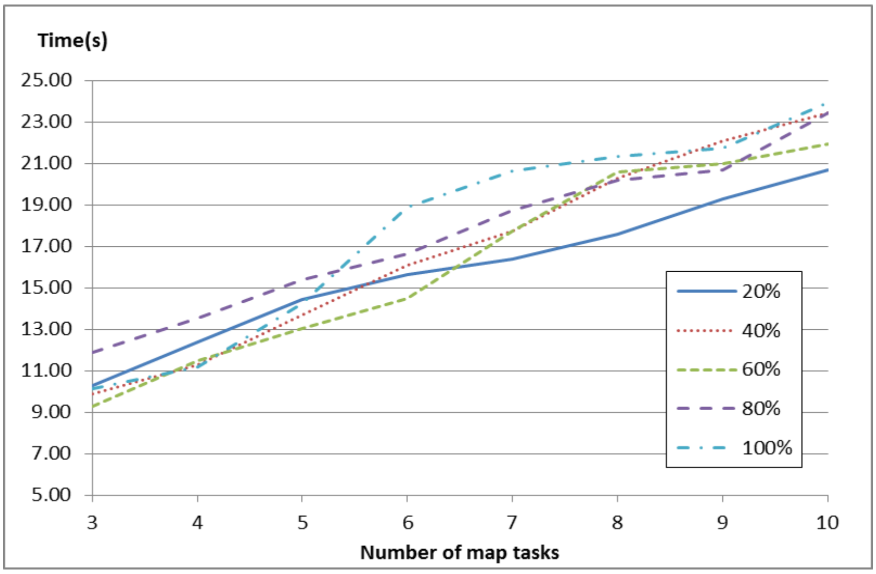

Fig. 5: Processing time using different number of map tasks for different amount of images.

Table 1: Average processing time of 2860 images using different number of map tasks.

Map Tasks	Average processing time(s)			
	Real	User	Sys	User + Sys
3	10.12	0.67	0.11	0.78
4	11.20	0.67	0.12	0.79
5	14.21	0.73	0.10	0.82
6	18.86	0.76	0.16	0.93
7	20.62	0.86	0.13	0.99
8	21.32	0.96	0.13	1.09
9	21.75	0.83	0.15	0.98
10	23.93	0.81	0.14	0.95

The experiment use the *time* function provided in Linux operating system. In the default format the function give three time values which are elapsed real time (Real), process time in user mode (User) and process time in kernel mode (Sys). Table 1 shows the average processing time of 2860 images using different number of map tasks. The actual processing time for running histogram equalization over all the images can be calculated by adding together the User and Sys times.

Based on the table, the amount of time for User + Sys is nearly constant even with different number of map tasks. On the other hand, the amount of Real time is increasing when the number of map tasks is increasing. That is because the amount of time needed to process all images is smaller than the amount of time used to transfer the images among map tasks. Thus, increasing the map tasks will resulting in the increase of the total time when the processing time is small.

Conclusion and Future Works:

In this work, a distributed visual enhancement is applied to database images from surveillance cameras. In the experiment, the images are processed with histogram equalization algorithm which was implemented using MapReduce programming architecture. The result shows that the increases of map tasks will increase the total time because the processing time of histogram equalization is very small. Main part of the total time is contributed by the transferring time of images among map tasks. Thus, we argue that the distributed infrastructure implemented with Hadoop MapReduce is suitable for more complex problems which require higher processing capability. In the future, we are planning to execute more complex problems in distributed environment. Moreover, our next step is to build a distributed environment that combines multiple machines.

ACKNOWLEDGEMENT

This work was supported by Multimedia University MiniFund (MMUI/130104).

REFERENCES

Celik, T. and T. Tjahjadi, 2012. Automatic image equalization and contrast enhancement using Gaussian mixture modeling. Image Processing, IEEE Transactions on, 21(1):145–156.

DelMarco, S. and S. Agaian, 2009. The design of wavelets for image enhancement and target detection. In SPIE Mobile Multimedia/Image Processing, Security, and Applications.

Lim, S.H., 2013. A new histogram equalization method for digital image enhancement and brightness preservation. Signal, Image and Video Processing, pp: 1–15.

Liu, A., 2010. Just noticeable difference for images with decomposition model for separating edge and textured regions. Circuits and Systems for Video Technology, IEEE Transactions on, 20(11), pp: 1648–1652.

Liu, X., 2009. A multiscale contrast enhancement algorithm for breast cancer detection using Laplacian Pyramid. In Information and Automation, 2009. ICIA’09. International Conference on, pp: 1167–1171.

Lu, L., 2010. Comparative study of histogram equalization algorithms for image enhancement. In Proc. SPIE 7708, Mobile Multimedia/Image Processing, Security, and Applications.

Mohan, S. and M. Ravishankar, 2013. Modified contrast limited adaptive histogram equalization based on local contrast enhancement for mammogram images. In Mobile Communication and Power Engineering. Springer, pp: 397–403.

Nimkar, S., S. Varghese and S. Shrivastava, 2013. Contrast Enhancement And Brightness Preservation Using Multi-Decomposition Histogram Equalization. arXiv preprint arXiv:1307.3054.

Panetta, K., Y. Zhou, 2011. Nonlinear unsharp masking for mammogram enhancement. Information Technology in Biomedicine, IEEE Transactions on, 15(6): 918–928.

Panetta, K., S. Agaian, 2011. Parameterized logarithmic framework for image enhancement. Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on, 41(2): 460–473.

Panetta, K.A., S.S. Agaian, 2011. Shape-dependent canny edge detector. Optical Engineering, 50(8): 87008.

Panetta, K.A., E.J. Wharton and S.S. Agaian, 2008a. Human visual system-based image enhancement and logarithmic contrast measure. Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on, 38(1): 174–188.

Panetta, K.A., E.J. Wharton and S.S. Agaian, 2008b. Logarithmic Edge Detection with Applications. Journal of Computers, 3(9).

Sengee, N., A. Sengee and H.K. Choi, 2010. Image contrast enhancement using bi-histogram equalization with neighborhood metrics. Consumer Electronics, IEEE Transactions on, 56(4): 2727–2734.

Tang, J., R.M. Rangayyan, 2009. Computer-aided detection and diagnosis of breast cancer with mammography: recent advances. Information Technology in Biomedicine, IEEE Transactions on, 13(2): 236–251.

Tang, J., X. Liu and Q. Sun, 2009. A direct image contrast enhancement algorithm in the wavelet domain for screening mammograms. Selected Topics in Signal Processing, IEEE Journal of, 3(1): 74–80.

Xia, J., K. Panetta and S. Agaian, 2010. Wavelet Transform Coefficient Histogram-Based Image Enhancement Algorithms. In SPIE Mobile Multimedia/Image Processing, Security, and Applications.

Australian Journal of Basic and Applied Sciences, 8(9) June 2014, Pages: 38-44

Yang, F., Y. Chang and S. Wan, 2005. Gradient-threshold edge detection based on the human visual system. Optical Engineering, 44(2): 20505.