

IMAGE CLASSIFICATION USING TWO DIMENSIONAL
WAVELET COEFFICIENTS WITH PARALLEL
COMPUTING

ONG YEW FAI

MASTER OF ENGINEERING SCIENCE

FACULTY OF ENGINEERING AND GREEN
TECHNOLOGY
UNIVERSITI TUNKU ABDUL RAHMAN
DECEMBER 2020

**IMAGE CLASSIFICATION USING TWO DIMENSIONAL WAVELET
COEFFICIENTS WITH PARALLEL COMPUTING**

By

ONG YEW FAI

A thesis submitted to the Department of Electronic Engineering,
Faculty of Engineering And Green Technology
Universiti Tunku Abdul Rahman,
in partial fulfillment of the requirements for the degree of
Master of Engineering Science
December 2020

ABSTRACT

IMAGE CLASSIFICATION USING TWO DIMENSIONAL WAVELET COEFFICIENTS WITH PARALLEL COMPUTING

ONG YEW FAI

Wavelet is a mathematical function that decomposes any given data signals and enabling the extraction of discontinuities and sharp spikes permeated in the signal. A two-dimensional signal which is represented by an image can be decomposed through wavelet transform into elementary forms at different resolutions and scales. The raster graphics are decomposed using various kind of wavelets together with numerous type of methods in this research to examine the process and the output coefficients of wavelet transform. Parallel processing and matrix convolution inside wavelet transform process is the most prominent study in this research. Furthermore, colour threshold analysis and edge detection algorithms are redesigned exclusively to refine the two-dimensional coefficients into a more meaningful low-level representation. The low-level features such as edges, blobs, and ridges were extracted for grouping and classification.

Object recognition and image classification comes after the wavelet transform. User is authorized to select and name any specified region from the

image, and store the colour, brightness, size, and various kind of information together with the selected area as the individual characteristics for the particular region. Whenever an area with the similar characteristic is verified in the next frame, the area will be identified and bounded with a rectangle through the object recognition algorithm of the research. Classification can be done precisely and efficiently for objects identified in the future frame.

From the study, a prevalent midrange laptop integrated with an intel i7 processor and a built-in Nvidia graphic card is served as a standardized device, a tremendously precise and efficient image classification outcome can be obtained within a minute from this research. With the aid of parallel processing method introduced in this research, both central processing core units and graphic card receive a different program instruction to execute the task assigned to them simultaneously. This method is capable to shorten the process time of the computation-intensive program in this research to process the live images frame by frame, which received from a laptop webcam to achieve real-time video processing. In addition, this research does not require any pre-stored database to train the algorithm.

This research requires the supervision from the users to train the algorithm by naming the region. This research algorithm demonstrated a very promising result with Support Vector Machines, this algorithm produces a 90% of accuracies whereas the decision tree algorithm gets 100% accuracies. Future development can be carried out by creating a database based on the user's standard of measure and accessing it whenever the user starts the program.

APPROVAL SHEET

This dissertation/thesis entitled “IMAGE CLASSIFICATION USING TWO DIMENSIONAL WAVELET COEFFICIENTS WITH PARALLEL COMPUTING” was prepared by ONG YEW FAI and submitted as partial fulfillment of the requirements for the degree of Master of Engineering Science at Universiti Tunku Abdul Rahman.

Approved by: /

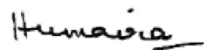


(Prof. Dr. YAP VOOI VOON)

Date: 6.12.2020

Professor

Department of Electronic Engineering
Faculty of Engineering and Green Technology
Universiti Tunku Abdul Rahman



(Prof. Dr. HUMAIRA NISAR)

Date: 6.12.2020

Co-supervisor

Department of Electronic Engineering
Faculty of Engineering and Green Technology
Universiti Tunku Abdul Rahman

FACULTY OF ENGINEERING AND GREEN TECHNOLOGY

UNIVERSITI TUNKU ABDUL RAHMAN

Date: 6.12.2020

SUBMISSION OF FINAL YEAR PROJECT /DISSERTATION/THESIS

It is hereby certified that ONG YEW FAI (ID No: 15AGM06656)
has completed this final year project/ dissertation/ thesis* entitled “ **IMAGE
CLASSIFICATION USING TWO DIMENSIONAL WAVELET
COEFFICIENTS WITH PARALLEL COMPUTING** ” under the supervision of
Dr. Yap Vooi Voon (Supervisor) from the Department of
Electronic Engineering, Faculty of Engineering and Green
Technology , and Dr. Humaira Nisar (Co-Supervisor)* from the Department of
Electronic Engineering, Faculty of Engineering and Green Technology.

I understand that University will upload softcopy of my dissertation in pdf format
into UTAR Institutional Repository, which may be made accessible to UTAR
community and public.

Yours truly,



(ONG YEW FAI)

*Delete whichever not applicable

ACKNOWLEDGEMENTS

I would like to express my very great appreciation to my main supervisor, Dr. Yap Vooi Voon, and my co-supervisor, Dr. Humaira Nisar for the suggestions of this research work. The amount of time and effort has been very much appreciated.

Finally, I wish to thank my parents for their encouragements throughout my study.

DECLARATION

I hereby declare that the dissertation is based on my original work except for quotations and citations which have been duly acknowledged. I also declare that it has not been previously or concurrently submitted for any other degree at UTAR or other institutions.

Name ONG YEW FAI

Date 6.12.2020

TABLE OF CONTENTS

	Page
ABSTRACT	ii
APPROVAL SHEET	iv
PERMISSION SHEET	v
ACKNOWLEDGEMENTS	vi
DECLARATION	vii
TABLE OF CONTENTS	viii
LIST OF TABLES	xi
LIST OF GRAPHS	xv
LIST OF FIGURES	xviii
LIST OF ABBREVIATIONS/NOTATION/GLOSSARY OF TERMS	xx
1.0 INTRODUCTION	1
1.1 Background	1
1.2 Scope	3
1.3 Research Methodology	4
1.4 Organization of Dissertation	6
2.0 MATHEMATICAL BACKGROUND	7
2.1 Introduction of Wavelet	7
2.2 History of Wavelet	8
2.3 The Fourier Transform	10
2.4 The Wavelet Transform	13
2.5 Wavelet Transform and Its Applications	16
3.0 DIGITAL IMAGE PROPERTIES, MACHINE LEARNING AND PARALLEL COMPUTING	20
3.1 Image Captured by Camera	20
3.1.1 Introduction of Image Captured by Camera	21
3.1.2 Colour Space	22
3.2 Image Segmentation	24
3.2.1 Methods Used of Image Segmentation	25
3.2.2 Colour Threshold Analysis For Image Segmentation	27

3.3	Image Features and Statistical Properties	28
3.4	Machine Learning	33
3.4.1	Uses of Machine Learning	34
3.5	Image Classification	36
3.5.1	Logistic Regression	36
3.5.2	Decision Tree	38
3.5.3	Random Forest	39
3.5.4	Gradient-Boosted Tree	40
3.5.5	Naïve Bayes Classifier	40
3.5.6	Support Vector Machine (SVM)	41
3.5.7	K-Nearest Neighbor	42
3.6	Parallel Computing	42
4.0	METHODOLOGY	45
4.1	The Image Classification Process of Research	45
4.1.1	Pre-processing phase cohesive with wavelet transform	46
4.2	Discrete Wavelet Transform (DWT)	47
4.2.1	Two-Dimensional Discrete Wavelet Transform (2D-dwt)	
	Function of Matlab Wavelet Toolbox	52
4.2.2	Method 1	53
4.2.3	Method 2	56
4.2.4	Method 3	56
4.2.5	Method 4	57
4.2.6	Method 5	60
4.2.7	Method 6	63
4.2.8	Method 7	65
4.2.9	Method 8	67
4.2.10	Method 9	72
4.2.11	Method 10	73
4.3	Colour Threshold Analysis and Features Extraction	76
4.3.1	Histogram With 256 Palettes Discrete Domain Category	77
4.3.2	Features Scaling For Smooth Regions And Edges	80
4.4	Feature Extraction and Image Segmentation	85
4.5	Define Statistics, Structural and Physical Difference	87
4.6	Machine Learning and Object-Based Classification	88

4.7	Parallelization of Wavelet Transform	91
5.0	RESULTS AND DISCUSSION	94
5.1	Wavelet Transform	94
5.1.1	Computation Time of Wavelet Transform and Matlab Wavelet Toolbox Function	96
5.1.2	Method 1	99
5.1.3	Method 2	102
5.1.4	Method 3	105
5.1.5	Method 4	108
5.1.6	Method 5	111
5.1.7	Method 6	114
5.1.8	Method 7	117
5.1.9	Method 8	120
5.1.10	Method 9	123
5.1.11	Method 10	126
5.1.12	Comparison of Average Computation Time of All 10 Methods	129
5.2	The statistical properties of wavelet coefficients	134
5.2.1	Sharpness Comparison Between Traditional Wavelet Transform and Proposed Method of The Research	148
5.3	Feature Extraction	158
5.4	Image Segmentation	164
5.5	Accuracy of Classification	170
5.5.1	Decision Trees	175
5.5.2	Support Vector Machines (SVM)	178
6.0	CONCLUSION	185
6.1	Objective of This Research	185
6.2	Brief Review of Methodology	186
6.3	Brief Review of Results and Contributions of This Research	186
6.4	Future Work	188
6.0	REFERENCE	189

LIST OF TABLES

Table		Page
4.2. a	The difference between the 10 proposed method to perform computational comparison with Matlab wavelet toolbox.	51
5.1.1. a	Computation time of Haar wavelet transform and Db3 wavelet transform recorded in second(s) with different resolutions	96
5.1.2. a	Computation time of Haar wavelet transform and Db3 wavelet transform recorded in second(s) with different resolutions without using Parallel Computing Toolbox	99
5.1.2. b	Computation time of Haar wavelet transform and Db3 wavelet transform recorded in second(s) with different resolutions using Parallel Computing Toolbox	100
5.1.3. a	Computation time of Haar wavelet transform and Db3 wavelet transform recorded in second(s) with different resolutions without using Parallel Computing Toolbox	102
5.1.3. b	Computation time of Haar wavelet transform and Db3 wavelet transform recorded in second(s) with different resolutions using Parallel Computing Toolbox	103
5.1.4. a	Computation time of Haar wavelet transform and Db3 wavelet transform recorded in second(s) with different resolutions without using Parallel Computing Toolbox	105
5.1.4. b	Computation time of Haar wavelet transform and Db3 wavelet transform recorded in second(s) with different resolutions using Parallel Computing Toolbox	106
5.1.5. a	Computation time of Haar wavelet transform and Db3 wavelet transform recorded in second(s) with different resolutions without using Parallel Computing Toolbox	108
5.1.5. b	Computation time of Haar wavelet transform and Db3 wavelet transform recorded in second(s) with different resolutions using Parallel Computing Toolbox	109
5.1.6. a	Computation time of Haar wavelet transform and Db3 wavelet transform recorded in second(s) with	111

	different resolutions without using Parallel Computing Toolbox	
5.1.6. b	Computation time of Haar wavelet transform and Db3 wavelet transform recorded in second(s) with different resolutions using Parallel Computing Toolbox	112
5.1.7. a	Computation time of Haar wavelet transform and Db3 wavelet transform recorded in second(s) with different resolutions without using Parallel Computing Toolbox	114
5.1.7. b	Computation time of Haar wavelet transform and Db3 wavelet transform recorded in second(s) with different resolutions using Parallel Computing Toolbox	115
5.1.8. a	Computation time of Haar wavelet transform and Db3 wavelet transform recorded in second(s) with different resolutions without using Parallel Computing Toolbox	117
5.1.8. b	Computation time of Haar wavelet transform and Db3 wavelet transform recorded in second(s) with different resolutions using Parallel Computing Toolbox	118
5.1.9. a	Computation time of Haar wavelet transform and Db3 wavelet transform recorded in second(s) with different resolutions without using Parallel Computing Toolbox	120
5.1.9. b	Computation time of Haar wavelet transform and Db3 wavelet transform recorded in second(s) with different resolutions using Parallel Computing Toolbox	121
5.1.10. a	Computation time of Haar wavelet transform and Db3 wavelet transform recorded in second(s) with different resolutions without using Parallel Computing Toolbox	123
5.1.10. b	Computation time of Haar wavelet transform and Db3 wavelet transform recorded in second(s) with different resolutions using Parallel Computing Toolbox	124
5.1.11. a	Computation time of Haar wavelet transform and Db3 wavelet transform recorded in second(s) with different resolutions without using Parallel Computing Toolbox	126
5.1.11. b	Computation time of Haar wavelet transform and Db3 wavelet transform recorded in second(s) with different resolutions using Parallel Computing Toolbox	127
5.1.12. a	Computation time comparison of Haar wavelet transform recorded in second(s) with and without Parallel Computing Toolbox	129

5.1.12. b	Computation time comparison of Db3 wavelet transform recorded in second(s) with and without Parallel Computing Toolbox	130
5.2. a	Statistical Properties of test images Haar wavelet coefficients of Method 6	135
5.2. b	Statistical Properties of test images Haar wavelet coefficients of Method 7	136
5.2. c	Statistical Properties of test images Haar wavelet coefficients of Method 10	136
5.2. d	Statistical Properties of test images Db3 wavelet coefficients of Method 6	137
5.2. e	Statistical Properties of test images Db3 wavelet coefficients of Method 7	137
5.2. f	Statistical Properties of test images Db3 wavelet coefficients of Method 10	138
5.2. g	Statistical Properties of test images Db6 wavelet coefficients of Method 6	138
5.2. h	Statistical Properties of test images Db6 wavelet coefficients of Method 7	139
5.2. i	Statistical Properties of test images Db6 wavelet coefficients of Method 10	139
5.2. j	Statistical Properties of standard images Haar wavelet coefficients of Method 6	141
5.2. k	Statistical Properties of standard images Haar wavelet coefficients of Method 7	142
5.2. l	Statistical Properties of standard images Haar wavelet coefficients of Method 10	142
5.2. m	Statistical Properties of standard images Db3 wavelet coefficients of Method 6	143
5.2. n	Statistical Properties of standard images Db3 wavelet coefficients of Method 7	143
5.2. o	Statistical Properties of standard images Db3 wavelet coefficients of Method 10	144
5.2. p	Statistical Properties of standard images Db6 wavelet coefficients of Method 6	144
5.2. q	Statistical Properties of standard images Db6 wavelet coefficients of Method 7	145
5.2. r	Statistical Properties of standard images Db6 wavelet coefficients of Method 10	145
5.4. a	Statistical Properties of the selected objects	167
5.5. a	Images in datasets for decision tree	171-174
5.5.1. a	10 random test images are numbered from 1 to 10	175-176
5.5.1. b	Final scores taken from 10 test images' classification	176-177

5.5.2. a	Datasets for SVM classification	178-179
----------	---------------------------------	---------

LIST OF GRAPHS

Graph		Page
5.1.1. a	Computation time of Haar wavelet transform and Db3 wavelet transform recorded in second(s) with different resolutions	97
5.1.2. a	Computation time of Haar wavelet transform and Db3 wavelet transform recorded in second(s) with different resolutions without using Parallel Computing Toolbox	99
5.1.2. b	Computation time of Haar wavelet transform and Db3 wavelet transform recorded in second(s) with different resolutions using Parallel Computing Toolbox	100
5.1.3. a	Computation time of Haar wavelet transform and Db3 wavelet transform recorded in second(s) with different resolutions without using Parallel Computing Toolbox	102
5.1.3. b	Computation time of Haar wavelet transform and Db3 wavelet transform recorded in second(s) with different resolutions using Parallel Computing Toolbox	104
5.1.4. a	Computation time of Haar wavelet transform and Db3 wavelet transform recorded in second(s) with different resolutions without using Parallel Computing Toolbox	105
5.1.4. b	Computation time of Haar wavelet transform and Db3 wavelet transform recorded in second(s) with different resolutions using Parallel Computing Toolbox	106
5.1.5. a	Computation time of Haar wavelet transform and Db3 wavelet transform recorded in second(s) with different resolutions without using Parallel Computing Toolbox	108
5.1.5. b	Computation time of Haar wavelet transform and Db3 wavelet transform recorded in second(s) with different resolutions using Parallel Computing Toolbox	109
5.1.6. a	Computation time of Haar wavelet transform and Db3 wavelet transform recorded in second(s) with different resolutions without using Parallel Computing Toolbox	111

5.1.6. b	Computation time of Haar wavelet transform and Db3 wavelet transform recorded in second(s) with different resolutions using Parallel Computing Toolbox	112
5.1.7. a	Computation time of Haar wavelet transform and Db3 wavelet transform recorded in second(s) with different resolutions without using Parallel Computing Toolbox	114
5.1.7. b	Computation time of Haar wavelet transform and Db3 wavelet transform recorded in second(s) with different resolutions using Parallel Computing Toolbox	115
5.1.8. a	Computation time of Haar wavelet transform and Db3 wavelet transform recorded in second(s) with different resolutions without using Parallel Computing Toolbox	117
5.1.8. b	Computation time of Haar wavelet transform and Db3 wavelet transform recorded in second(s) with different resolutions using Parallel Computing Toolbox	118
5.1.9. a	Computation time of Haar wavelet transform and Db3 wavelet transform recorded in second(s) with different resolutions without using Parallel Computing Toolbox	120
5.1.9. b	Computation time of Haar wavelet transform and Db3 wavelet transform recorded in second(s) with different resolutions using Parallel Computing Toolbox	121
5.1.10. a	Computation time of Haar wavelet transform and Db3 wavelet transform recorded in second(s) with different resolutions without using Parallel Computing Toolbox	123
5.1.10. b	Computation time of Haar wavelet transform and Db3 wavelet transform recorded in second(s) with different resolutions using Parallel Computing Toolbox	124
5.1.11. a	Computation time of Haar wavelet transform and Db3 wavelet transform recorded in second(s) with different resolutions without using Parallel Computing Toolbox	126
5.1.11. b	Computation time of Haar wavelet transform and Db3 wavelet transform recorded in second(s) with different resolutions using Parallel Computing Toolbox	127
5.1.12.a	Computation time comparison of Haar wavelet transform recorded in second(s) with and without Parallel Computing Toolbox	131
5.1.12.b	Computation time comparison of Db3 wavelet transform recorded in second(s) with and without Parallel Computing Toolbox	131

5.2.1.a	Bar Chart of Test Images Horizontal-Detail Wavelet Coefficients' mean	149
5.2.1.b	Bar Chart of Test Images Vertical-Detail Wavelet Coefficients' mean	149
5.2.1.c	Bar Chart of Test Images Diagonal-Detail Wavelet Coefficients' mean	150
5.2.1.d	Bar Chart of Test Images Horizontal-Detail Wavelet Coefficients' standard deviation	151
5.2.1.e	Bar Chart of Test Images Vertical-Detail Wavelet Coefficients' standard deviation	151
5.2.1.f	Bar Chart of Test Images Diagonal-Detail Wavelet Coefficients' standard deviation	152
5.2.1.g	Bar Chart of Standard Images Horizontal-Detail Wavelet Coefficients' mean	153
5.2.1.h	Bar Chart of Standard Images Vertical-Detail Wavelet Coefficients' mean	153
5.2.1.i	Bar Chart of Standard Images Diagonal-Detail Wavelet Coefficients' mean	154
5.2.1.j	Bar Chart of Standard Images Horizontal-Detail Wavelet Coefficients' standard deviation	155
5.2.1.k	Bar Chart of Standard Images Vertical-Detail Wavelet Coefficients' standard deviation	155
5.2.1.l	Bar Chart of Standard Images Diagonal-Detail Wavelet Coefficients' standard deviation	156
5.4. a	Entropy of the selected objects	167
5.4. b	Skewness and Kurtosis Deviation of the selected objects	168
5.4. c	Mean and Standard Deviation of the selected objects	168

LIST OF FIGURES

Figure		Page
2.3. a	The amplitude and phase angle of a sine wave at a particular frequency	11
2.4. a	Block diagram of filter analysis	14
2.4. b	A 3-level filter bank	15
2.4. c	Haar wavelet function and scaling function	16
3.3. a	Graphical illustration of skewness based on mean, median and mode	31
3.3. b	Graphical illustration of kurtosis distribution	32
3.5.1 a	Graph representation of sigmoid function	37
3.6. a	Illustration of Instruction-level Parallelism	43
4.1. a	An All-inclusive Flow Chart of The Image Classification Process	45
4.2. a	The four types of wavelet coefficients generated from the input image, the approximation coefficients matrix (O_a), the horizontal detail coefficients matrix (O_h), the vertical detail coefficients matrix (O_v), and the diagonal detail coefficients matrix (O_d).	49
4.2.2. a	The area of selection for wavelet transform	54
4.2.5. a	The four types of two-dimensional wavelet decomposition filters	58
4.2.6. a	Illustration of the two techniques to create a new raster graphic proposed in this research	61
4.2.9. a	The H-assessing area, V-assessing area, and D-assessing area for 2D-dwt	69
5.2. a	Test Images	134
5.2. b	Standard Images	140
5.3. a	Edge feature from wavelet coefficients of Test image 4	159

5.3. b	The 4 arrangements of sub-areas	160
5.3. c	The process of creating a local bar chart and local graph from a sub-area of test image 4	161
5.3. d	Edge features of test image 4 created by local colour threshold analysis	162
5.4. a	presents the chosen object of all test images after image segmentation to build dataset	165
5.5.2. a	Final scores taken from 10 test images' classification by support vector machine	180-183

LIST OF ABBREVIATIONS

ROI	Region of interest	
STFT	Short-time Fourier transform	
dwt	discrete wavelet transform	
2D-dwt	Two-dimensional discrete wavelet transform	
Spmd	single program, multiple data	
RGB	red, green, blue	
HSV/HSL	hue, saturation, V=lightness	
YCbCr	Luminance, Chrominance-blue, Chrominance-red	
SVM	Support Vector Machine	
FD_{low}	low – frequency analysis decomposition filter	
FD_{high}	high – frequency analysis decomposition filter	
FR_{low}	low – frequency analysis reconstruction filter	
FR_{high}	high – frequency analysis reconstruction filter	
O_a	<i>approximation coefficients matrix</i>	
O_h	<i>horizontal detailed coefficients matrix</i>	
O_v	<i>vertical detailed coefficients matrix</i>	
O_d	<i>diagonal detailed coefficients matrix</i>	

CHAPTER 1

INTRODUCTION

1.1 Background

Object classification is a common task in computer vision applications. In general, object classification is a process that aims to classify objects. This involves comparing the measured features of an unknown object with features of a known object. This process will determine whether the unknown object belongs to a particular group of known objects.

This research acknowledges that object classification is a well-researched area and this poses the question, is it worth continuing with research into object classification? With the resurgent of parallel computing and wavelet mathematical techniques and machine learning, this dissertation argues that there is still room for research into object classification.

Image is an important information source and is one of the main media of understanding of the world. The demand for image processing technology is increasing day by day and artificial intelligence algorithms are infused to improve the performance of image processing. Artificial intelligence is capable

in optimizing the image recognition and classification process, whereas it still requires a mutual technology to perform real time image segmentation to yield the region of interest.

Region of interest (ROI) is a sample selected within a data set for a particular purpose. Region of interest can improve the classification accuracy of any kind of images without computing any redundant information from peripheral areas. Only parts of the image are analyzed and statistics properties such as mean and standard deviation are drawn from it. The computation time can be improved by locating the region of interest to reduce the amount of data for processing.

The objectives of this research work are to investigate the fundamental properties of wavelet coefficients for feature selection used for segmentation to classification using the best classifier. This research work emphasized in the process known as image segmentation. The accuracy of classification is typically hinged on the quality of ROI. Thus, this research proposes various approaches of wavelet transform to harvest a sharper and high-quality wavelet coefficient for image segmentation. This research is to present the outcomes of using wavelet coefficients as the features to represent an image. The noteworthy part of this research is the effects of wavelet transform on image segmentation; classification will be much easier with meticulously segmented regions. The best classifier is then employed and collaborate with the best wavelet transform algorithm to form a fully-functional classification algorithm designed for all sorts of industries and transportations.

1.2 Scope

Wavelet analysis provides multiresolution analysis to extract regions of interest in image. Peripheral areas from different variable resolution can be removed using scaled wavelet coefficients. One of the aims of this research is to develop a more optimal wavelet transform algorithm to yield better wavelet coefficients features for image classification.

In this research, standard compression technique of wavelet transform is revised and to exploit the edge features for object-based classification through some major modification. The proposed method in this research process the wavelet coefficients to identify the region of interests and creates blob features. Blob features are created through segmentation by grouping pixels that have similar spectral and spatial characteristics. Blob features are distinguished as unidentified objects to be predict into a target class using object-based classification algorithm.

Object -based classification or object-oriented classification classifies or categorizes pixels based on their spectral characteristics, texture, shape, and solely on the spatial information in each pixel and it surrounding pixels. The classification is done by assigning each object to a class based on features, connectivity, proximity, and criteria set by the user. Another aim of this research is to utilize the features extracted from wavelet transform for image classification for different kind of images encountered by video surveillance sequences.

1.3 Research Methodology

This research begins with exploring the applications and usages of sets of complementary wavelets such as compression, information extraction and denoising in various industry fields. When a video surveillance sequence is recorded, there will be non-ideal condition such as variation in brightness or background effects. Wavelet transform is utilized to identify, classify, and solve the complexity of the two-dimensional data with shortest time, which is the key objectives in the image retrieval system. Thus, parallel computing is then introduced to speed up the process of wavelet transforms in this research to yield results in real-time image processing.

Through a two-dimensional wavelet transform, raster graphic is decomposed into two-dimensional wavelet coefficients with edges representing the colour discontinuity information. Image segmentation is performed using the wavelet coefficients to create region of interest for image classification to complete the image processing process.

There are a variety of different parameters that are used in segmentation and the most challenging task is to find the cluster center determination. Colour threshold analysis is used in this research work to dissect the wavelet coefficients that carries information correspond to the change in object, depth, illumination and surface orientation.

Image segmentation is then performed based on the threshold values to

yield region of interest. Lastly, object-based image classification for all the region of interest is carried out using a supervised machine learning algorithm.

1.4 Organization of Dissertation

The rest of the dissertation is organized as follows;

Chapter 2 discusses the mathematical background and the contents of the Fourier transform, continuous wavelet transform and discrete wavelet transform.

Chapter 3 is a literature review of digital images properties and methods used for image segmentations. This section also reviews the technology that is utterly valuable to industries with large amount of data, which is machine learning algorithm and parallel computing.

Chapter 4 begins with the process flow and the sequence to do robust image classification using the features extracted by wavelet transform. Wavelet transform is thoroughly explored in this chapter using 10 different algorithms.

Chapter 5 presents the results and discussion of this dissertation using a real-time classification algorithm and wavelet transform written in parallel algorithm.

Chapter 6 is a summary of the results presented in the dissertation and it also presents some ideas for future work.

CHAPTER 2

MATHEMATICAL BACKGROUND

This chapter will explore in the detail of wavelet as this is a key component in this research work. This discussion will also explore the simplicity, performances and applications of wavelet transform as they will be used in this research work.

2.1 Introduction of Wavelet

Wavelet analysis has enjoyed a tremendous attention and popularity over the last decade. This provides a very simple and efficient means of performing such an analysis. Based on research conducted by other researchers, there are many research works done using Fourier transform in the frequency domain. However, wavelet transform is introduced as a tool for spectral analysis which uses a computation method that can localize events both in time and in frequency in the entire time-frequency plane while enables the reduction of processing time.

Wavelet is commonly used in engineering and computer science in conventional and non-conventional applications. Conventional applications include image compression and communication signal processing. Unconventional applications include seismic geology, chemistry, neurophysiology, nondestructive evaluation, fractals, and economics. The success stories of wavelets can be found in medical devices for the detection of Alzheimer's disease, cracks detection in piping of nuclear power plants, and inspection of fluctuations of financial markets.

2.2 History of Wavelet

Wavelet is a recent discovery in mathematics and rapid developments make them more powerful than any other analytical tools. The mathematics of wavelets has its origin in Fourier series and Fourier transform. The Fourier representation only carries spectral content without any time localization of the spectral components. Therefore, a simpler prototype function known as basis function analyzes the non-stationary signals, whose spectral content change in time. They provide valuable insight to analysis of complicated functions. Furthermore, compression can be achieved from the representation of the original function if only a few of these basis function renders good approximation.

The first wavelet is discovered and developed by Alfred Haar in the early 20th century. Later, Dennis Gabor was involved in modifying Fourier transform

into short time Fourier transform and yielded Gabor atoms (Alvaro et al., 2014). He was interested in using oscillatory basis functions in a time frequency plane to represent a communication signal. Shortly after 1946, Jean Ville devised a time-frequency representation known as the Wigner-Ville transform for the energy signals in 1947 (Zeng et al., 2015). Wigner-Ville is a transform that deals with signals with linear and nonlinear frequency modulated signals.

Continuous wavelet transform is formulated by Pierre Goupillaud, Grossmann and Morlet's in 1982. They came up with an ingenious idea of using different window function for analyzing different frequency band while they encountered some very high frequency components with short time span. There is a slightly different interpretation of Alberto Calderon's work on harmonic analysis in 1964 (Calderón, 1964).

In 1983, discrete wavelets is discovered by Jan-Olov Strömberg's (Stroemberg, 1994). In 1985, Yves Meyer constructed orthogonal wavelet basis functions with very good time and frequency localization. In the meantime, Ingrid Daubechies developed wavelet frames for discretization of time and scale parameters of wavelet transform. She wrote a dissertation of Daubechies' orthogonal wavelets with compact support in 1988 and Mallat's multiresolution framework was founded in 1989 (Daubechies, 1988).

2.3 The Fourier Transform

As mentioned earlier, wavelet mathematics has its origins in Fourier transform. Therefore, it would be beneficial to look at Fourier transform. Wavelets are functions that satisfy certain mathematical requirements that can analyze and process data according to scale and resolution. There are continuous wavelet transform, discrete wavelet transform, multi-wavelet, and complex wavelet existed. The window size of a wavelet filter can extract different features such as discontinuities and sharp spikes contain in finite domains. Unlike Fourier analysis, the Fourier transform is an extension of the Fourier series that results the function's period stretched out to infinity.

The Fourier transform of the expression $f = f(t)$ with parameter c is respect to the variable t at the point ω is

$$F(\omega) = c \int_{-\infty}^{\infty} f(t)e^{-j\omega t} dt \quad (1)$$

The norm of the amplitude, $|F(\omega)|$ is called the Fourier spectrum of f . The exponent $\Phi(\omega)$ is called the phase angle. The F is often expressed in polar form though.

$$F(\omega) = |F(\omega)|e^{j\phi(\omega)} \quad (2)$$

$$|F(\omega)| = \sqrt{a^2 + b^2} \quad (3)$$

$$\phi(\omega) = \tan^{-1} \left(\frac{b}{a} \right) \quad (4)$$

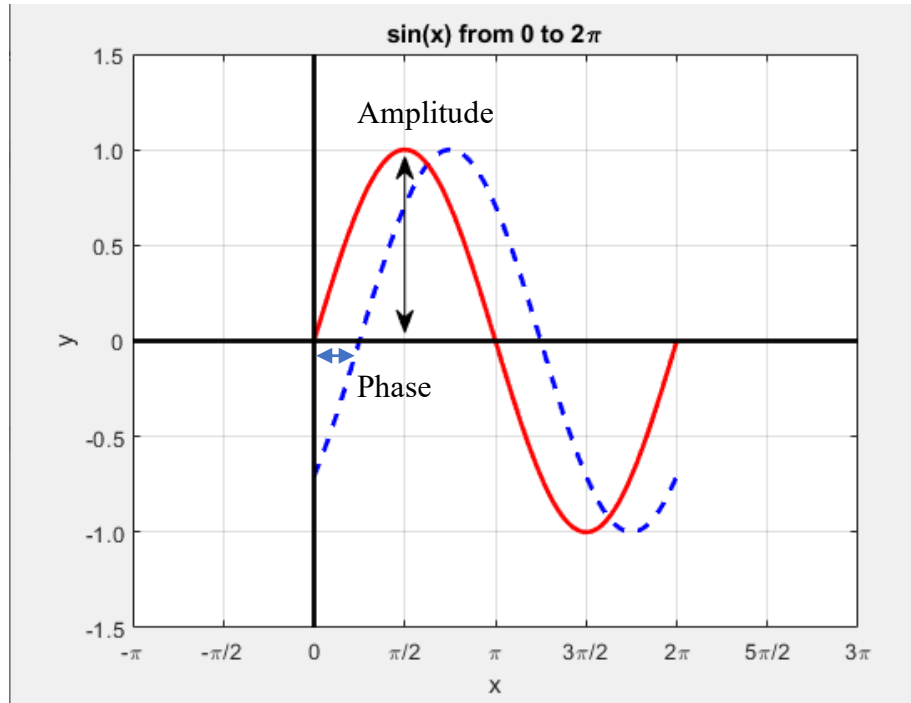


Figure 2.3.a: The amplitude and phase angle of a sine wave at a particular frequency

In most of the researches, amplitude information is evaluated whereas the phase information is discarded due to the degradation of the image after reconstruction. Fourier transform breaks a waveform into an alternate representation made up from a sum of sinusoidal basis functions. Therefore, the output of Fourier transform composed of the constituent frequencies hidden inside a signal. The inverse Fourier transform can synthesize the original function from its frequency domain representation. The reconstruction of the original image from Fourier components uses the inverse Fourier transform basis function, that is

$$f(t) = \frac{1}{2\pi} \int_{-\infty}^{\infty} F(\omega) e^{j\omega t} d\omega \quad (5)$$

Another well-known linear time-frequency analysis method is the short-time Fourier transform (STFT). STFT is a mathematical tool used to determine the sinusoidal frequency and phase content of a sections of signal. The function to be transformed is multiplied by a window function $w(\tau)$. $f(\tau, \omega)$ is transformed from the complex function $x(t)w(t - \tau)$ that represents the phase and magnitude of the signal which is written as

$$f(\tau, \omega) = \int x(t)w(t - \tau)e^{-j\omega t} dt \quad (6)$$

The short-time Fourier transform is invertible using the most commonly known overlap-add method with the inverse of $f(\tau, \omega)$ for (τ) is fixed.

$$x(t)w(t - \tau) = \frac{1}{2\pi} \int_{-\infty}^{\infty} f(\tau, \omega)e^{j\omega t} d\omega \quad (7)$$

One disadvantage of STFT is that it has a fixed resolution, However, this can be overcome by using wavelet transform and multiresolution analysis. This is related to Heisenberg uncertainty principle articulated in March 1927. In physics, there is a trade-off process between speed and position, it always a minimum amount of uncertainty in the measurement of both values known as resolution of measurement. There will be a difficulty in finding the exact frequency at an exact moment in time known as instantaneous frequency. However, wavelet tends to deconstruct signal into a load of wavelets being added together with limited time and frequency.

2.4 The Wavelet Transform

The continuous wavelet transform can translate and dilate along the signal to convolute the input signal $x(t)$ into another complex valued representation. The function $f(t)$ is composed of an arbitrary mother wavelet $\psi(t)$, the subspace of scale $a > 0$, and a translational value b is expressed by the following integral (Mallat, 2008)

$$f(t) = \frac{1}{|a|^{1/2}} \int_{-\infty}^{\infty} x(t) \psi\left(\frac{t-b}{a}\right) dt \quad (8)$$

Mother wavelet is scalable and its scale is inversely proportional to the frequency. The window size is bigger when the scale is larger to target the lower frequency band. The child wavelets of continuous wavelet transform for frequency band $\left[\frac{1}{a}, \frac{2}{a}\right]$ is expressed by the following equation

$$\psi_{a,b}(t) = \frac{1}{|a|^{1/2}} \psi\left(\frac{t-b}{a}\right) \quad (9)$$

However, it is computationally impossible to analyze all the coefficients generated from the continuous wavelet transform. The continuous wavelet transform typically uses exponential scales with a base smaller than two, when multiplied by the sampling interval, so the scale vector covering approximately four octaves with the sampling interval taken into account. For discrete wavelet transform (DWT), the base scale is always two.

Thus, a discrete subset is chosen which the corresponding wavelet coefficients are sufficient to reconstruct the original signal of finite energy. The

restriction in scale is only allows the increment in power of two whereas the translation is only integer value allowed. The corresponding child wavelet for DWTs are written as

$$\psi_{m,n}(t) = \frac{1}{|a^m|^{1/2}} \psi\left(\frac{t-nb}{a^m}\right) \quad (10)$$

The mother wavelet provides a source function to be translated and scaled to generate a daughter wavelet. Wavelet coefficients are computed from DWT using the daughter wavelet as a filter bank. Detail coefficients and approximation coefficients are generated from the high-pass filter and low-pass filter respectively. The two filter are related to each other and known as a quadrature mirror filter.

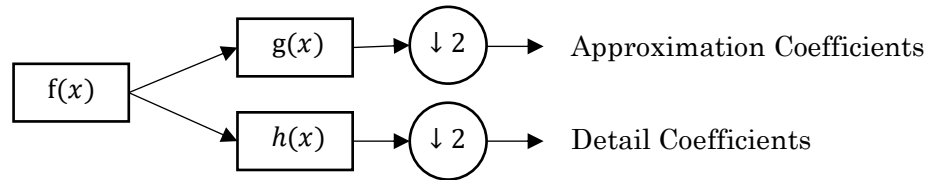


Figure 2.4.a: Block diagram of filter analysis

After one level of the transform, half the frequencies of the signal are removed according to the Nyquist's theorem. The original signal can be perfectly reconstructed using the threshold band-limit of Nyquist rate. The filtered output is only half the original signal resolution whereas the frequency resolution has been doubled. The process is repeated by cascading the filters to form a multilevel decomposition, sometimes referred to as the Mallat decomposition. Sending the filtered output to a new high-pass filter and low-pass filter will generate another down-sampled filtered output and this can be

represented in a binary tree with nodes known as filter bank.

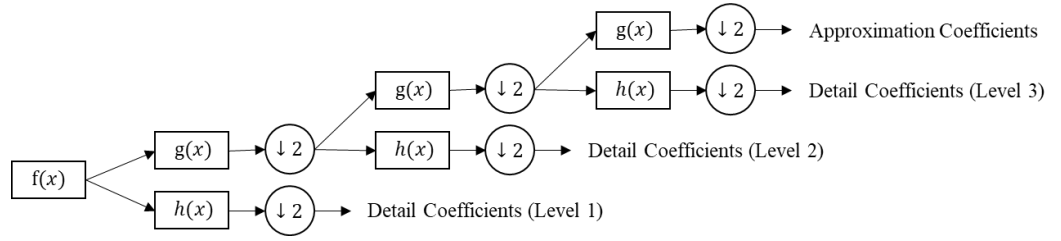


Figure 2.4.b: A 3-level filter bank

The mother wavelet is shifted and scaled by powers of two in DWT with j is the scale parameter and k is the shift parameter.

$$\psi_{j,k}(t) = \frac{1}{|2^j|^{1/2}} \psi\left(\frac{t-k2^j}{2^j}\right) \quad (11)$$

The multiresolution representation and the subband approach are integrated into the framework of the wavelet theory. The multiresolution analysis was introduced by Stephane Mallat and Yves Meyer (Mallat, 1989). The Mallat iterative algorithm for implementing the one-dimensional DWT is shown in Figure 2.4.a. Another commonly used method to implement DWT is the pyramid algorithm. The pyramid methods of image processing are introduced by Peter J. Burt, Edward H. Adelson and James L. Crowley (Crowley et al., 1987). This method made significant improvement to the computational efficiency of the DWT. Multiresolution analysis able to eliminate numerical complexity of integration.

Father wavelet is introduced to determine the right scaling for the unique scaling restriction in discrete wavelet transform. The father wavelet φ in the subspace $L^2(R)$, is the auxiliary function that keeps the time domain properties

whereas the mother wavelets keep the frequency domain properties. The simplest wavelet analysis can be shown by Haar scaling function.

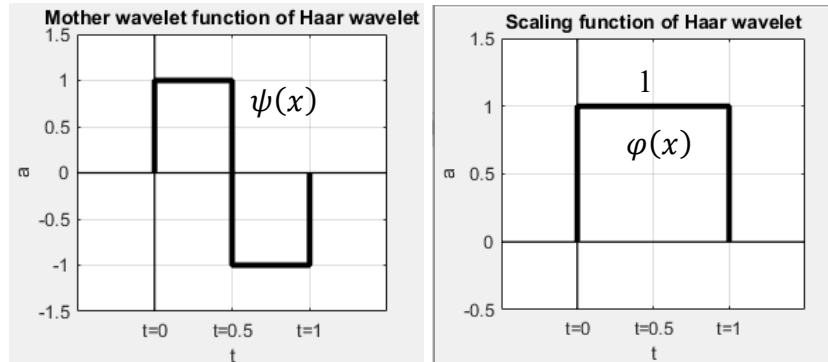


Figure 2.4.c: Haar wavelet function and scaling function

$$\varphi(x) = \begin{cases} 1, & \text{if } 0 \leq x < 1 \\ 0, & \text{otherwise} \end{cases} \quad (12)$$

$$\psi(x) = \begin{cases} 1, & 0 \leq x < 1/2 \\ -1, & 1/2 \leq x < 1 \\ 0, & \text{otherwise} \end{cases} \quad (13)$$

2.5 Wavelet Transform and Its Applications

The application areas for wavelets have been growing in various field. Apart from its original intention of analyzing the frequency components in a continuous time signal, wavelet theory is applicable to other purposes such as image processing and compression applications due to its simplicity and high compression performance (Kekrea et al., 2016).

The compression of signals using wavelet transform is able to reduce the storage and increase the transmission speed for analysis or remote diagnostics as the redundancies of data carried by the original signal is reduced (Barsanti et al., 2013). A research stated that decimated discrete wavelet compression shows promising results in compression compared with other methods (Oinam et al., 2013). Multiresolution characteristic of DWT is exploited during the research of the efficiency in image transmission (Manhas et al., 2012; Goh et al., 2008). Wavelet transform is similar with other transform methods that encode the transformed data, such as JPEG 2000 uses biorthogonal wavelets as an image compression standard. In (Nasri et al., 2010), an embedded block coding with optimized truncation and DWT are implemented for better order of transmission to investigate the performance of the image compression with respect to the image quality and energy consumption.

With the rapid advancement hardware and software for medical imaging system, X-ray Ultrasound, MRI/ Nuclear medicine, Computerized Tomography and Dual energy X-ray, the need to store, share and retrieve images are rising leads to several terabytes of storage required yearly. Wavelet has been introduced and used in threshold filtering process to find the signal of interest for real biomedical data filtering and compression systems which shows superior performance. In (Paul Nii et al., 2019), perpetual quality at a tolerant level is preserved using the proposed method, DWTs with vector quantization to reduce the speckle and salt and pepper noises.

Studies in economic growth uses wavelet-based quantile on quantile method to decompose the raw information to stimulates the economic growth in long run. In the papers (Bin Mo et al., 2019; Dong et al., 2019), the dynamic relationship between global economic activity and crude oil prices is examined using wavelet in both time-domain and frequency domain. The finding shows significant correlation between the crude oil prices and global economic activity at high frequencies, in term of short run shows dynamic lead-lag relationship across time. Other researchers (Ko and Lee, 2015; Risse, 2019) attained a negative result while examining the link between the stock price and economic policy using wavelet analysis.

A comparative numerical analysis of mother wavelets using real fractal signals and fractal dimension enables the signal evaluation using wavelet decomposition (Makieł et al., 2017). The research (Zhang et al., 2017) shows that symlet-2 mother wavelet is the most suitable mother wavelet with the lowest mean square error and variance for both long and short signals. Relative wavelet spectrum, power-spectra, wavelet-based spectral ratio is used to evaluate periodicities and fractal characteristics of stratigraphic records. In a research (Song et al., 2018), the use of wavelet-based spectra shows a fractal trend and important cycle periods at certain spatial scales.

On the other hand, wavelet based denoising is used in seismic analysis to extract noise free seismic data. With the combination of type two fuzzy tool, the threshold is set accurately and information stored in the noisy wavelet coefficients is effectively extracted in the research (Beena mol et al., 2016). The

research (Wang et al., 2017) proposed to calculate the source wavelet generated by explosions for seismic exploration considering source wavelet amplitude and frequency. The researchers analyzed the relationship between the explosion parameters and the source wavelets, showing that main frequency of source wavelet of limestones is relatively low and needed amplification with higher denotation velocity explosion. A paper (Huang et al., 2005) shows significant improvement in the experimental results in term of perceptual error metrics and visual effect using wavelet transform domain in the denoising process.

CHAPTER 3

DIGITAL IMAGE PROPERTIES, MACHINE LEARNING AND PARALLEL COMPUTING

As one of the main concerns of this research work is the digital colour images, this chapter will discuss the characteristics of digital images and followed by image segmentation. Parallel computing and machine learning will also be discussed in section 3.4 and onwards.

3.1 Image Captured by Camera

Camera is an optical instrument with the capability to capture certain visible spectrum of light photons from the surrounding and convert to digital signal for other purposes. The digital signal is stored in a two-dimensional raster graphic with various colour tone and brightness representing different wavelength and strength of visible spectrum. Therefore, this section will discuss the differences between raster image and vector image to compare the information stored in both signals and how to sightsee the information.

3.1.1 Introduction of Image Captured by Camera

Raster graphics are two-dimensional digital images with size determined by the number of pixels. The quantity is indicated by resolution where pixels are arranged in the horizontal and vertical dimensions in grid formation. The quality of images is mainly dependent on the resolution where the information is kept. Raster images are capable of rendering complex, multi-coloured visuals, provides rich details photographs or vivid painting. Raster images are commonly created by pixel-based programs or captured with camera or scanner. Unlike the vector graphic information uses geometry and mathematical equations.

Vector graphics are created with vector software and commonly for images that used in computer-aided design, engineering, and 3D graphics. The raster images will look blurry when enlarged due to pixels are added randomly throughout the image whereas vector graphic images support image scaling without losing quality.

Raster images consent a myriad of colours presented in a two-dimensional signal and allow for colour editing beyond that of a vector image.

3.1.2 Colour Space

The human eye has the ability to perceive colour due to the sensitivity of photoreceptor cells called cones. Each type of cone responds to a range of wavelengths. Keeping this property in the human visual system, all colours can be modelled by colour space. A colour space where all possible colours that can be represented using tuple of numbers in a colour model. A colour model can be translated into another colour model for the need of different applications.

Some of the well-known colour space such as RGB (red, green, blue), HSV/HSL (hue, saturation, V=lightness), and YCbCr (Luminance, Chrominance-blue, Chrominance-red) are developed to define and represent the true colour seen by our naked eyes. A colour mode of raster graphics can be grayscale, indexed, RGB, HSL, HSV or YCbCr. Colour space can be simply interpreted as all possible colours of visible light to represent the absolute colour information of each and every pixel (Thomas, 1802). Colour information of each pixel in the raster graphics are stored in the third dimension known as colour layer. As an example, is the RGB colour model images which uses three layers of two-dimensional raster graphics to store the range of colours as tuples of numbers for each pixel. Each layer is assigned with a range of 0 to 255 intensity values of RGB components. As a result, there can be 16,777,216 combinations for a single pixel.

Numerous researches have proven the human eyes have different sensitivity to colour and brightness. A human eye is unable to see all colours

presented by the RGB colour space and thus the transformation of RGB to YCbCr or YUV is used to reduce the range of colour for every image by discarding some of the information while retain the quality of the image that can be seen by human eyes. The Luminance is very similar to the grayscale version of the original image. Cb stores the sky (blue) colour of the image, and Cr stores the reddish colour. Both Cb and Cr contain less information of green colour.

Researches have been done and conclude that human eyes basically group objects based on the colour and brightness (Smith et al., 1978). The bright objects are generally grouped together and dark objects are grouped together before differentiate using colour. Thus, HSV colour space is developed in the mid of 1970s, by Alvy Ray Smith to represents the information because it is often more natural to think about a colour in terms of hue and saturation than in terms of additive or subtractive colour components. HSL or HSV are the two most common cylindrical-coordinate representations of points in an RGB colour model.

Comparing the RGB colour space with the HSV colour space, there are three primary colours in RGB colour space contain the information of light intensity whereas the HSV colour space only use the value layer represent the brightness. Tinting is mixing the colour with white and shading is to increase the darkness (Levkowitz et al., 1993). Colour in the HSV model range from 0-360 degree in the HUE layer rather than three layers of primary colours. The saturation is the intensity of the colour which is called the colourimetric quantities excitation purity.

3.2 Image Segmentation

Image segmentation is a technique used to partition a digital image into multiple segments. The process is to locate objects and boundaries in an image by grouping pixels with the similar characteristics. The pixels in a region can have the same characteristics or computed property, such as colour, intensity, or texture. There are numerous applications that uses image segmentation, such as machine vision, medical imaging, traffic control systems, object detection, and fingerprint recognition.

Thresholding is a simplest and yet effective way of partitioning an image into foreground and background. This image segmentation method can be made completely automated to automatically select the threshold based on the high level of contrast between the adjacent pixels. Automatic object segmentation is difficult due to issues such as shadow, lighting and semantic gaps. Object-based recognition is achievable from image segmentation however the appropriate segmentation parameter is a prerequisite to obtain accurate segmentation. The most commonly used thresholding methods are histogram shape-based method, clustering-based method, entropy-based method, object attribute-based method, spatial method, and local method.

3.2.1 Methods Used of Image Segmentation

There is a research (Parida, 2018) that uses transition region-based approaches for image segmentation. The research out performed with the method proposed using the local variance, fuzzy C-means clustering, morphological thinning operation, and Otsu thresholding incorporate with the transitional features from the feature image. There are researches (Batenburg et al., 2009) that uses radiograph instead of Otsu's method in developing thresholding methods for computed tomography (CT) images.

Research has been done using fuzzy connectedness framework with local phase and feature asymmetry, incorporates a shape-based completion and regularizes the result by mean curvature flow (Rueda et al., 2015). The research (Yu, H.Y. et al., 2015) uses fuzzy partition entropy-based method which has a nice property for parameter optimization in real application and achieved promising results. An image segmentation method based on network clustering model is proposed for mass segmentation in mammograms (Jiao et al., 2018).

K-means clustering algorithm is a common approach to detect similarities in the regions of an image. The K-means algorithms is an iterative technique that used to partition an image into clusters. The K value can be selected manually, randomly or heuristic to converge the pixels into categories based on colour, intensity, texture, and location.

Histogram-based approaches are more efficient compare to K-means clustering method because they typically require only one pass through the pixels. The peaks and valleys in the histogram are used to locate the clusters in the image using colour or intensity as the measure.

The paper (Chen et al., 2018) proposed a fast density clustering algorithm to find the position information and pixel similarity, cluster centers are determined automatically by multiple linear regression analysis. The experimental results proven the algorithm robust to parameters, which can automatically locate and determine the segmentation without supervision.

The proposed method in paper (Wang et al., 2019) uses unsupervised segmentation parameter in a multi-resolution segmentation algorithm achieved high quality of segmentation. The homogeneity and heterogeneity of the image were calculated using local spatial statistics approach for obtaining segmentation with high quality. There are variety of methods were developed over the years that can effectively solve segmentation problems using domain-specific knowledge.

3.2.2 Colour Threshold Analysis For Image Segmentation

Image segmentation is inevitable for object-based image processing to create clusters of pixels. The most challenging tasks are cluster center determination, low self-adaptability, and parameter dependence on the image structure. A research paper (Bhandari et al., 2016) proposed using Otsu or Kapur's method for the optimization process to improve the colour image segmentation process and measured the performance using differential evolution, wind driven optimization and particle swarm optimization. Colour image segmentation have been done using multiscale roughness measure through simulating the human vision (Yue et al., 2012). The optimal scale for segmentation is determined by the entropy variation. Optimal multi-level image segmentation using RGB histogram is used to obtain optimal thresholds for each colour component. The article (Rajinikanth et al., 2015) demonstrated using standard RGB dataset with proposed segmentation procedure for bi-level and multi-level segmentation, validation is done using Brownian Distribution, Levy Flight and the Gaussian distribution.

Referring to the available researches, colour images can be threshold by combining the threshold value of each RGB components using AND operation. This method is based on a threshold or a clip-level to turn all RGB channels into a binary image. Each channel stores different texture discontinuities and object boundaries correspond to the edges and depth information. Gathering all three channels provides a more detailed information compared to a single layer gray-scale image.

There are human detection and tracking system using RGB-D cameras shows significant promising results. RGB-D is a sensor that provide depth and colour information in a two-dimensional signal. The paper (Xiao et al., 2019) proposed a new human tracking method to find potential positive samples together with Support Vector Machine as the classifier to recognize the target human and demonstrated with high success rate using RGB-D camera. Grayscale histogram is used in the extraction of the segments of the shadows based on colour invariant shadow in RGB colour space of the paper (Acharya et al., 2018).

3.3 Image Features and Statistical Properties

Feature is defined as the interested information used as the starting point and main primitives for any algorithm. Feature detection is a low-level image processing operation as it is the initial step of a larger algorithm. Once features are found, local image patch around the feature can be identified and known as feature descriptor or feature vector. Feature vectors are generated to represent the attribute information in another readable format and accessible by the other parts of the algorithm. Features are in term of numerical values and usually much lower dimension than the original image. The reduction in dimensionality reduces the overheads of processing.

Features play a critical role in object segmentation. Features such as edges, blobs, ridges and interest points are the best description of local regions

in the image. This information is relevant for solving the computational task such as machine learning, classification, and object recognition. The process commonly referred to as feature extraction is used to produce sets of coordinates of the image points where the features have been detected. The sets of coordinates are generated based on the interest of the user. When feature extraction is performed without user's guide, it is referred to as a feature image and can be seen as a two-dimensional data information. Feature selection or attribute selection is done in machine learning to choose a subset of relevant features for model construction. Features are extracted as some representatives of each object and its class, insensitive to lighting conditions and colour, invariant to image transformations like rotation, translation and scaling.

The local features describe the image patches of an object whereas the global features describe the image as a whole of entire object. Local features store the texture information and global features include contour representation, shape descriptions, shape matrices, invariant moment, and histogram-oriented gradients.

Foreground detection or background subtraction is a process to extract the moving objects in image sequences. This process is required to develop a background model and be robust to lighting changes, repetitive movements and long-term changes. Features can be found from some conventional approaches such as temporal average filter (Genovese et al., 2015), using frame differencing, and mean filter.

A statistic is any quantity computed from values in a sample data, and also is an observable number. The important potential properties of statistics include completeness, consistency, sufficiency, minimum mean square error, robustness, and computational convenience. Descriptive statistics provide summarizes the observations in the form of simple to understand graph or quantitative. Image resolution enhancement is possible using statistical estimation in wavelet domain (Shi and Shan, 2012). The author (Andria et al., 2013) examine the statistical properties of the noise affecting computed tomography and magnetic resonance images to assess the different effect of different noise distribution. Another paper (Kurtek et al., 2013) uses statistics values to finds a robust estimate of the representative curve that provide an overall segmentation and to automatically detect outliers and help reduce their influence in the estimation.

Other features such as skewness, kurtosis, and entropy are meaningful statistical measures describing the texture of an image. The concepts are introduced to describe the distribution because the average and measure of dispersion is not sufficient to describe the nature of the distribution. Skewness is introduced to describe the lack of symmetry when the values are not uniformly distributed around the mean. In statistics it represents the degree of distortion from the symmetrical bell curve in a probability distribution. In a symmetrical distribution, the mean, median and mode should coincide with the degree of symmetry equal to 0, the negatively skewed distribution will be represented by a negative number whereas the positively skewed distribution will be in positive number.

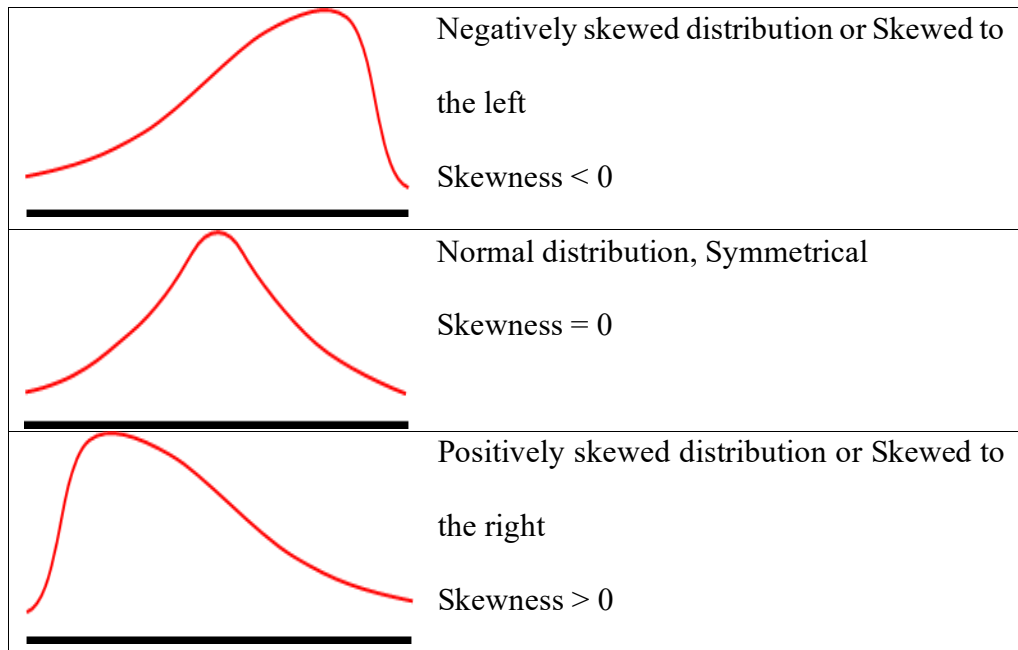


Figure 3.3.a: Graphical illustration of skewness based on mean, median and mode

Kurtosis is a measure for the degree of tailedness in the variable of distribution. Kurtosis shows the extremes of the data set rather than focusing solely on the average. The concept is similar to skewness as a descriptor of the shape of a probability distribution, such as platykurtic distribution, normal distribution, and leptokurtic distribution. Leptokurtic distribution has fatter tails whereas platykurtic has thinner tails.

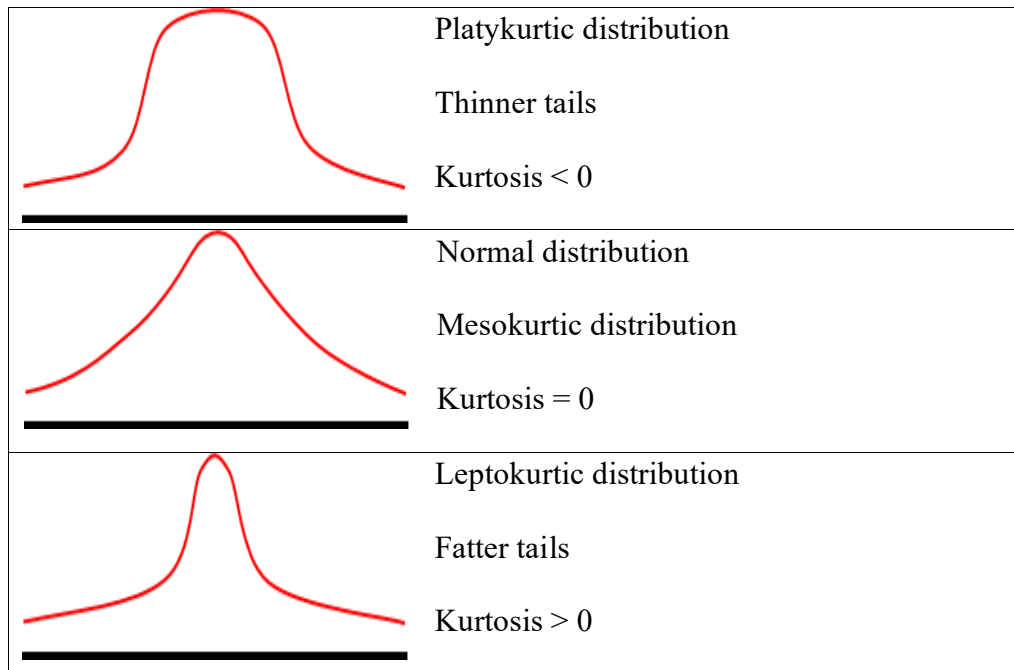


Figure 3.3.b: Graphical illustration of kurtosis distribution

Entropy is a measure of uncertainty and stay away from uniform probability distributions. Entropy allows us to make precise statements and perform computations. Shannon's entropy can calculate the disorder of the image patterns to classify textures. An image with lower entropy means low information content and the image is smoother compared to a higher entropy image.

The next few sections will explore machine learning and parallel computing in details.

3.4 Machine Learning

Machine learning is an application of artificial intelligence that comes with the ability to learn and improve the functional system by itself from experiences and data collections without any programming explicitly.

Experience without being explicitly supervised, the ability of self-learning and improving is the primary aim of building an artificial intelligence. Machine learning uses in the computer system is programmed to make predictions and decisions for a given task without further instruction given. However, machine learning algorithm requires a sample data, known as training data in order to perform the task effectively. They independently adapt and learn from previous computation to produce reliable, repeatable decisions. Machine learning also referred to as predictive analytics. Machine learning are often categorized into supervised or unsupervised.

Supervised machine learning algorithms uses known training matrix dataset and produces an inferred function to generate feature vectors to be used to predict the output associated with new inputs. The algorithm enhances the feature vectors of the dataset automatically to improve the accuracy of its output over time. Supervised machine learning is closely related to classification and regression, but the key of supervised machine learning is to learn and improve from new piece of data using a similarity learning function. The new unlabeled data will be labeled after the prediction or gradient boosting and learns by comparing it output with correct output to modify the model accordingly.

On the other hand, unsupervised machine learning algorithms uses unlabeled data and infer a function to describe the hidden structures from unlabeled data. The classification for unsupervised machine learning is done by using the patterns, structures, and features discovered by the algorithm itself. There is active learning algorithm that comes with the interactive function allows user to drop down a label. Reinforcement learning algorithms discovers through trial and error to determine the ideal behavior in yielding the best result.

3.4.1 Uses of Machine Learning

This section will briefly explore the technology utterly valuable to industries with large amount of data, and is created to work more efficiently in the competitive market. Machine learning can be found in financial services to identify important insights in data and to prevent fraud. The author in article (Barboza et al., 2017) performed bankruptcy prediction using boosting, bagging and random forest models in machine learning.

Machine learning is employed in the health care industry to access patient's health in real time, also analyses data to improve diagnoses and treatment. The paper (Krishnan et al., 2018) presented machine learning in predicting the dissolution rate of silicate glasses and paper (Arkaprabha et al., 2019) compare performance of different machine learning algorithms for screening anxiety and depression among the seafarers.

Government agencies uses machine learning to minimize criminal acts, finding new energy sources, and identify patterns and trends of transportation needs to increase profitability.

Data mining applies methods from many different areas such as statistical algorithms, machine learning, database system, and time series analysis. The previously unknown patterns from data are studied and extract information from dataset, transform into a comprehensible structure for further use.

Deep learning or artificial neural network is a subfield of machine learning. It comes with the ability to perform automatic feature extraction and scalability. The algorithm performs a task repeatedly and each time improves the outcome. The drawback of this method is the algorithm requires lots of data to learn from and demands high computing power. The datasets can be very diverse, unstructured and interconnected. Applications of deep learning are virtual assistants, translations (Marta et al., 2017), facial recognition, personalized shopping (Affonso et al., 2017) and entertainments are getting more and more user-friendly day after days.

3.5 Image Classification

The following section will briefly discuss the process in computer vision that can classify an image based on the distinct regions extracted from image segmentation. A classification model will attempt to draw conclusion from observed values. Classification predicts categorical class labels or classifies data based on the training dataset (Kuznetsova et al., 2020). Classification models include logistic regression, decision tree, random forest, gradient-boosted tree, Naïve Bayes classifier, support vector machine (SVM) (Affonso et al., 2017), and nearest neighbor. There are researches explore the recent state of art of detection methods (Jiao, L.C. et al., 2019; Zou et al., 2019). Analysis of existing typical detection models, and the architecture of exploiting these methods has been done and came out with some representative object detection methods, such as lightweight object detection, kernel SVM with constant inference time, and weakly supervised detection.

3.5.1 Logistic Regression

Logistic regression can be classified into binomial, multinomial and ordinal. This is a supervised classification algorithm requires a threshold value to model the probability of a certain class or event such as pass and fail. Instead of fitting a line to the data, logistic regression fits an “s” shaped, sigmoid function from 0 to 1. The sigmoid function is described as

$$f\{x\} = \frac{1}{1+e^{-x}} \quad (14)$$

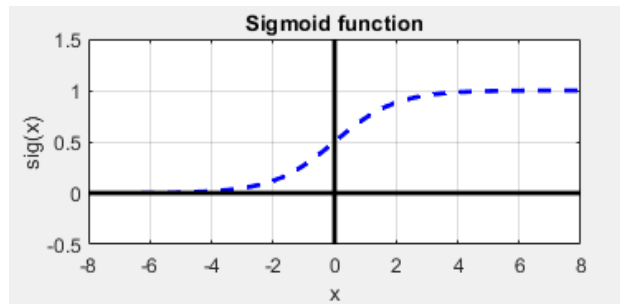


Figure 3.5.1.a: Graph representation of sigmoid function

The curve tells the probabilities of a sample data either pass or fail just like linear regression. Logistic regression's ability to provide probabilities and classify new samples using continuous and discrete measurements makes it a popular machine learning method. The threshold value of the logistic regression is computed using maximum likelihood instead of least squares, and threshold is dependent on the classification itself.

Logistic regression was employed in classify patients with cardiovascular system disorder (Anastasiya et al., 2014) draw isoprobability contours which represent the probability of stability of coal pillars (Wattimena et al., 2013).

3.5.2 Decision Tree

Decision tree is a flowchart that looks like a tree structure with each node represent an attribute. Decision tree is easy to use, easy to interpret, and can handle high dimensional data with high accuracy. Tests are performed from node to nodes in a recursive manner called recursive partitioning. The recursion stops when the subset at a node share the same value as the target variable. There is no domain knowledge or parameter setting required while setting up the decision tree. However, it prone to errors in classification when there are many classes and presented with new samples. The information gain is based on the concept of entropy, with j classes, summing the probability p_i and it is defined as below

$$f(x) = S_E(p_1, p_2, p_3 \dots p_j) = - \sum_{i=1}^j p_i \log_2 p_i \quad (15)$$

Decision tree can be used to classify microarray datasets in the diagnosis of different diseases (Akkaş et al., 2015), classify and identify various natural magnetic minerals (Mohmad et al., 2011), and reduces the amount of data from datasets to improve training time of support vector machine (Cervantes et al., 2015).

3.5.3 Random Forest

Random forest is a combination of multiple decision trees with flexibility together with bootstrap aggregation. A wide variety of trees makes more effective than individual decision trees. The training algorithm applies a general technique of bootstrap aggregating. Random selections made in sample data $X = x_1, \dots, x_n$ run down all the trees created from bootstrap aggregation. Bagging B is the process creates votes to make a decision from the regression trees. After training, predictions for unseen samples x' can be made by using the average of all regression trees.

$$\hat{f} = \frac{1}{B} \sum_{b=1}^B f_b(x'), \text{ for } b = 1, \dots, B \quad (16)$$

However, there will be part of the original data that does not end up in the bootstrap dataset. The missed-out data in the random forest are then used to measure the accuracy by running it through all the trees and calculate the votes.

3.5.4 Gradient-Boosted Tree

Gradient-boost for regression is different from linear regression or logistic regression because it is designed to be configured in any kind of ways. Gradient-boost starts by making a single leaf, instead of a tree or a stump. Average value of a variable from the sample is calculated and a tree is built from it up to the number of leaves set by the user. Based on the average value, previous tree's errors or residuals computed from the observed values and the predicted value, is known as pseudo residual and placed inside a new leaf. Gradient-boost enables scaling the contribution for new trees using learning rate defined by user. The accuracy of prediction is dependent on the number of trees and leaves defined by user.

3.5.5 Naïve Bayes Classifier

A naive Bayes classifier is a probabilistic classifier machine learning model. The classifier can handle high dimensional data, highly scalable and can handle an arbitrary number of independent variables whether continuous or categorical. The parameter estimation for naive Bayes models uses the maximum likelihood method.

It is popular in categorizing documents with the feature of word frequencies. A comparison has been made with boosted trees or random forest sows that naïve Bayes classifier outperformed them (Caruana et al., 2006).

3.5.6 Support Vector Machine (SVM)

In SVM for linearly separable binary sets, two main classes are distinguished by plotting one or more hyperplane that classifies all training data. The best choice will be the hyperplane that leaves the maximum margin from both classes. This classifier has a strong mathematical background and better generalization capability but the drawback is high computation power and large datasets are required. In the research paper (Meghana, 2012), the application of nonlinear feature extraction for human detection is presented. The features are obtained using wavelet transforms and SVM is used for classification of the data. The result of human detection using SVM shows that co-occurrence feature viz. cluster shade, cluster prominence gives success rate is of almost 100%. This research (Le et al., 2012) uses a novel combination of artificial neural network together with support vector machine after feature extraction. This research uses sub-space feature with 3-layers artificial neural networks together with 10 classes in support vector machine with average classification rate more than 80%. Another low false positive detection algorithm is presented using color segmentation, linear SVMs and Gaussian-kernel SVMs (Maldonado et al., 2007).

3.5.7 K-Nearest Neighbor

K-nearest neighbor is a super simple method to classify data. The classification requires a dataset with known categories for initial clustering, called training data. A new data sample is then classified to the nearest annotated category. The K-nearest neighbor classify the new data to the nearest neighbor with the most votes. Lower K value will be noisy and subject to the effects of outliers. High K value smooth over things but with a disadvantage of the sample will be out voted by other categories.

3.6 Parallel Computing

In the earliest computers, programs are run in single thread and only one instruction is processed at a time. With the objective of running a program in lesser time or running multiple programs simultaneously, multiprogramming (Sevcik, 1994), parallel processing, and multiprocessing are introduced. The advantages of having a system with parallel computing are I/O operation, processor-intensive calculations, and the execution of processes are allowed to carry out simultaneously with shorter execution time (Abawajy, 2006).

In parallel computing large problems are often broken down into several sub-tasks that can be processed independently and whose results are gathered upon completion. The algorithm breaks the problem into independent parts, assigns the tasks, and executes using resources such as a single computer with

multiple processors, several networked computers, or any specialized hardware. In some flow dependency cases, long chain calculations that needs prior calculations must be executed in order. Subtasks in a parallel program often called as threads, which need synchronized access to a variable or other resource. Lock and barrier are introduced to protect a variable from being accessed by two or more threads at the same time. Threads perform serialize access to resources to guarantee correct program execution.



Figure 3.6.a: Illustration of Instruction-level Parallelism

A well-known computer architecture from the mid of 1980 is instruction-level parallelism (Bernard and David, 2013), the instructions were combined into groups and execute in parallel in a clock cycle as shown in Figure 3.6.a. Scalar processors and superscalar processors works in a way when there is no data dependency between the instructions. Tasks parallelism involves the decomposition of the intensive computational task into sub-tasks and send to all processors for execution. Each processor performs different type of task simultaneously and gather the output when all processors completed the given task. Programmers need to parallelize their software code effectively to take advantage of the increasing computer power as the number of cores per processor will double every two years predicted by Moore’s law (Michael, 1987).

Regarding to parallel computing memory architectures, there are shared memory, distributed memory, and cache available for storage of variables for program execution. Different memory organizations of parallel computer platforms require different algorithm for the distribution of work and data to the processors. The distributed memory allows each processor to own a storage space and access locally without the interference from other processors. Local memory is typically faster compared to shared memory in parallel computing (David et al., 2002). Cache are temporary and fast memories located close to the processors.

CHAPTER 4

METHODOLOGY

4.1 The Image Classification Process of Research

This chapter will explore the phases and sequence to train a machine learning algorithm, and to do robust image classification using the features extracted by wavelet transform.

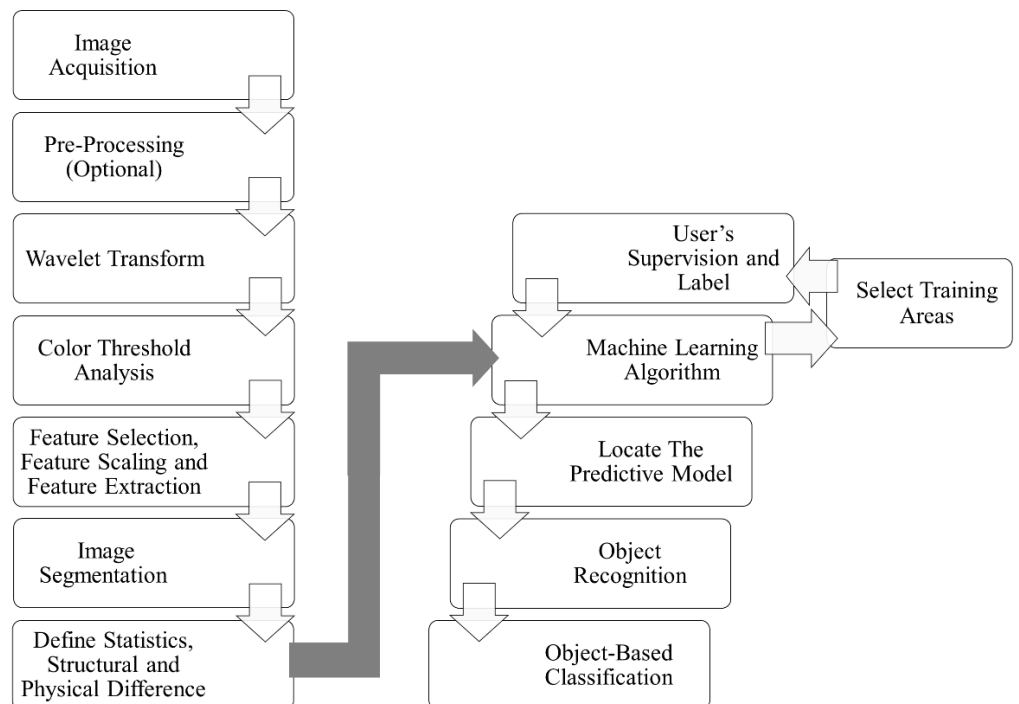


Figure 4.1.a: An All-inclusive Flow Chart of The Image Classification Process

Image classification is the reckoning process searching for the most relevant category assimilated in the system when acquired an image. In general, there are three main techniques as discussed in the section 3.4. Supervised and unsupervised classification is pixel-based whereas object-based image classification groups the pixels into representative shapes and sizes. In this research, object-based image classification is the proposed method and image segmentation is applied using the coefficients generated from discrete wavelet transform (DWT).

4.1.1 Pre-processing phase cohesive with wavelet transform

Pre-processing is a lowest level of abstraction where both input and output are intensity images. This process suppresses unwanted distortion and improve the readability of further processing stage.

Wavelet transform is well-known mathematical tool act as a denoising system. Almost all practically useful DWTs use discrete-time filter banks. These filter banks are called the wavelet and scaling coefficients in wavelet nomenclature. Thus, wavelet transform is also served as a pre-processing system to reduce noises while extract information in the form of wavelet coefficients.

As a pre-processing stage before edge detection, Gaussian method is the most popular method which can be found in most of the researches. It can

effectively reduce noises in an image. However, the disadvantage is the smoothing stage also affects the sharpness of the edges when it tarnished the concerned edges.

In this research, wavelet transform is proposed to reduce the load of computation, to reduce to processing time and mostly is the wavelet decomposition filter will correlate with the concerned edges and constructively create vigilant values inside the wavelet detailed coefficients after convolution.

4.2 Discrete Wavelet Transform (DWT)

Discrete wavelet transform is the decomposition process which extracts relevant and important information concealed in the signals. In mathematics, a wavelet series is discretized, making them suitable for numerical evaluation and implementation on digital computers. The discrete wavelet model is denoted as the wavelet filter and it is applicable in spatial domain as discussed in section 2.4. Wavelet filter is a one-dimensional finite length array of numbers. There are low-frequency and high-frequency analysis filters in both decomposition wavelet filters and reconstruction wavelet filters respectively. In this research, the decomposition wavelet filters are the key to image classification. In this research, the wavelet filters are created using the Matlab wavelet toolbox *wfilters* function. The *wfilters* function compute the four low-frequency, high-frequency, decomposition and reconstruction filters associated with wavelet name specified by the input string.

First of all, two-dimensional wavelet filter is inevitable in the process of discrete wavelet transform. There are four notable type of two-dimensional wavelet filters can be derived from the low-frequency and high-frequency analysis filters using the dot product multiplication.

A two-dimensional convolution is an integral that expresses the amount of overlap of one input raster graphics with another matrix. Convolution in DWT refers to formation of third signal using impulse decomposition, known as the wavelet coefficients. The wavelet filters will correlate with the signal if the unknown signal contains information of similar frequency. This concept of correlation is at the core of many practical applications of wavelet theory. There are four types of output coefficients accredited to the four wavelet filters. The Figure 4.2 below illustrates the four types of wavelet coefficients, the approximation coefficients matrix (O_a), the horizontal detail coefficients matrix (O_h), the vertical detail coefficients matrix (O_v), and the diagonal detail coefficients matrix (O_d).

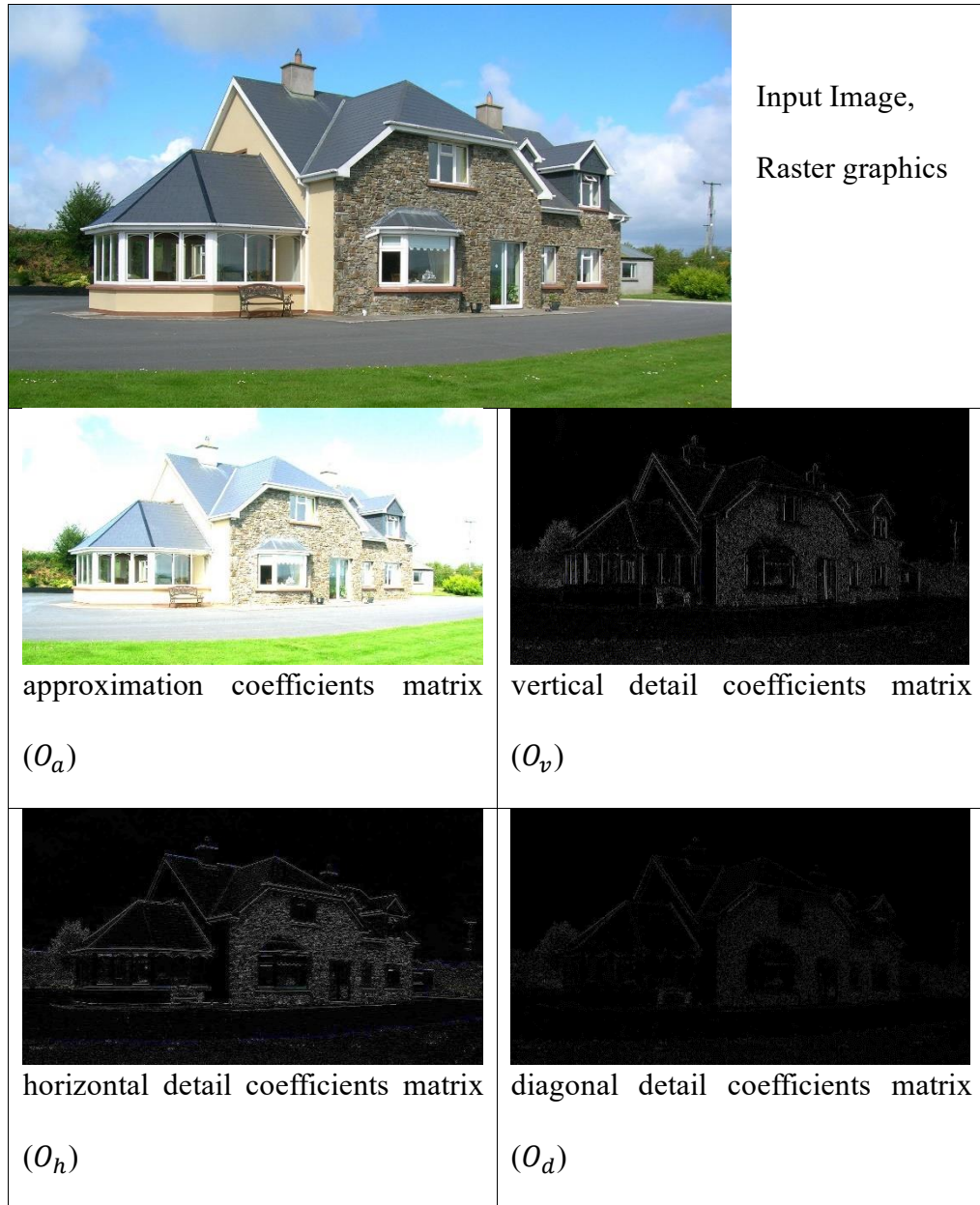


Figure 4.2.a: The four types of wavelet coefficients generated from the input image, the approximation coefficients matrix (O_a), the horizontal detail coefficients matrix (O_h), the vertical detail coefficients matrix (O_v), and the diagonal detail coefficients matrix (O_d).

In the two-dimensional DWT (2D-dwt) process, superior coefficients value is formed when the edges and ridges are aligned with the peak value of the wavelet filter. Thus, sharpened edges are indicated by the constructive pixels represented in the two-dimensional wavelet coefficients and can be found in the horizontal, vertical and diagonal detail coefficients matrices among the four wavelet coefficients.

The wavelet coefficients play an important role in the next process in image classification. Thus, improvements have to be done during the discrete wavelet transform. The Matlab wavelet toolbox has offer a 2D-dwt function which is fast and easy to use. However, there are two main disadvantages using this function, that is, it is not modifiable and the output wavelet coefficients' dimensions are down sampled and pre-set to be reduce by half compared to the input's dimensions.

When wavelet transform function is broken up into smaller blocks or sections, it becomes more feasible to edit and block of codes are replaceable with a more effective coding method or optimize the program execution. The Table 4.2.a below briefly described the differences between the 10 methods proposed in this research.

Method	Comparison	
1	Simplest coding format with nested for loops	One dimensional decomposition filter
2	Replaces the for loops with while loops	One dimensional decomposition filter
3	Code enhanced by removing nested for loops	One dimensional decomposition filter
4	Nested for loops removed, uses image concatenation	Two dimensional decomposition filter
5	Uses pad array function from Matlab	Two dimensional decomposition filter
6	Convolution method (change of dimension of the decomposition filters)	Two dimensional decomposition filter
7	Preserve the size of input image for wavelet coefficients	Two dimensional decomposition filter
8	Focus on a row and a column for wavelet transform	One dimensional decomposition filter
9	Focus on a row and a column for wavelet transform	Two dimensional decomposition filter
10	Convolution method (change of dimension of the decomposition filters)	Two dimensional decomposition filter
	Preserve the size of input image for wavelet coefficients	
	Focus on a row and a column for wavelet transform	

Table 4.2.a: The difference between the 10 proposed method to perform computational comparison with Matlab wavelet toolbox.

In this research, 10 different 2D-dwt methods have been proposed and compared with the function provided by Matlab wavelet toolbox. These methods will be discussed in the following sections. Apart from that, the key of the research work is to explore and study the data-intensive wavelet transform using Matlab parallel computing toolbox. The high-level constructs of parallel-enabled functions such as parallel for-loop and single program, multiple data,

spmd function are superimposed to the 10 methods to achieve parallelism. The toolbox restructures the methods to split up and run simultaneously on multiple processors as discussed section 3.6.

4.2.1 Two-Dimensional Discrete Wavelet Transform (2D-dwt) Function of Matlab Wavelet Toolbox

Matlab Wavelet Toolbox includes a function specialized for two-dimensional discrete wavelet analysis of signals and images. Using this function, images at different resolutions can be decomposed to detect gradients, edges, and other textures. The function accepts a two-dimensional array of integers with three separate raster maps and the name of the wavelet, produces four types of wavelet coefficients. Every output wavelet coefficients retained the three layers, but down sampled the spatial domain is half of the input raster graphics. The results and discussion are available in section 5.1.1.

The function provided in the Matlab wavelet toolbox (MATLAB, 2018) is shown below.

$$[O_a, O_h, O_v, O_d] = dwt2(Image, wavelet) \quad (17)$$

4.2.2 Method 1

The process of 2D-dwt is recreated from scratch in the Matlab coding environment. The objective of this method is to imitate the process of the 2D-dwt function in Matlab toolbox into a modifiable code. The 2D-dwt is fragmented into a much simpler process, which uses one-dimensional wavelet filters to perform the transformation. This method emphasizes in the use of *for* loops and its performance in Matlab environment. This method uses *parfor* to achieve parallel processing.

The process is completed using several *for* loops in nested formation forming the output coefficients pixel by pixel. Whenever the loop command is executed, the manipulated values are added up in a consecutive manner until it stops when reaches the most bottom right pixel of the raster graphic input signal. When iterating the block of code within the *for* loops, an area of the raster graphic is taken out. The area taken into calculations is altered by the vanishing moment of the wavelet filters used. Figure 4.2.2.a below illustrates the area of selection and output wavelet coefficients using wavelet filters with length of $N = 4$ as an example.

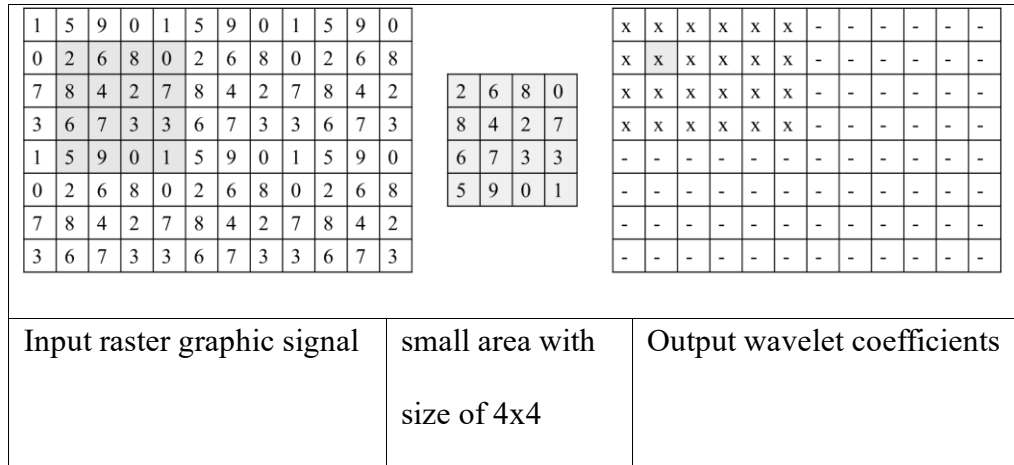


Figure 4.2.2.a: The area of selection for wavelet transform

A Matlab function is applied on the selection area to find the smallest value. Retune is done on the particular area aiming to set the offset of the smallest value to zero. The input, output, and the Matlab function (MATLAB, 2018) are shown below.

$$area_{min} = \min(small_area) \tag{18}$$

$$small_area_{min} = small_area - area_{min} \tag{19}$$

There are four equations placed inside the nested *for* loops of the program for convolution. The output of this imitated process resemblance the wavelet coefficients of the 2D-dwt of the Matlab toolbox up to four decimal places. The evidences and results are available in section 5.1.2.

The Matlab equations are listed below.

$$N, M = 2 \times \text{vanishing moment of the wavelet}$$

$$L = \text{number of rows of the image}$$

$$K = \text{number of columns of the image}$$

$$O_a = \sum_{w=1}^{\frac{L}{2}} \sum_{v=1}^{\frac{K}{2}} \left(O_a(v, w) \dots \right. \\ \left. + \sum_{y=1}^N \sum_{x=1}^M (small_area(x, y) \times FD_{low}(x) \times FD_{low}(y)) \right) \quad (20)$$

$$O_h = \sum_{w=1}^{\frac{L}{2}} \sum_{v=1}^{\frac{K}{2}} \left(O_h(v, w) \dots \right. \\ \left. + \sum_{y=1}^N \sum_{x=1}^M (small_area_{min}(x, y) \times FD_{low}(x) \times FD_{high}(y)) \right) \quad (21)$$

$$O_v = \sum_{w=1}^{\frac{L}{2}} \sum_{v=1}^{\frac{K}{2}} \left(O_v(v, w) \dots \right. \\ \left. + \sum_{y=1}^N \sum_{x=1}^M (small_area_{min}(x, y) \times FD_{high}(x) \times FD_{low}(y)) \right) \quad (22)$$

$$O_d = \sum_{w=1}^{\frac{L}{2}} \sum_{v=1}^{\frac{K}{2}} \left(O_d(v, w) \dots \right. \\ \left. + \sum_{y=1}^N \sum_{x=1}^M (small_area_{min}(x, y) \times FD_{high}(x) \times FD_{high}(y)) \right) \quad (23)$$

For $v, w, x, y = 1, 2, 3, 4 \dots$

4.2.3 Method 2

The process of 2D-dwt discussed in Method 1 is repeated except the *for* loops. The objective of this method is to assess the performance the 2D-dwt inside multiple nested *while* loops of Matlab environment. This method emphasizes in the use of *while* loops and its performance in Matlab environment. This method uses *spmd* function and *matlabpool* together with the *while* loops to achieve parallel processing.

The output of this process resemble the wavelet coefficients of the 2D-dwt of the Matlab toolbox up to four decimal places. The results and discussions are available in section 5.1.3.

4.2.4 Method 3

The process of 2D-dwt discussed in Method 1 is repeated except the *for* loops. The objective of this method is to assess the performance the 2D-dwt with the minimal number of *for* loops used. The output of this imitated process resemble the wavelet coefficients of the 2D-dwt of the Matlab toolbox up to four decimal places. This method refined some algorithm from Method 1, shorten the calculation steps using minimum number of *for* loops. Part of the steps are merged together into one process. This method uses *parfor* to achieve parallel processing. The results and discussions are available in section 5.1.4.

4.2.5 Method 4

The process of 2D-dwt discussed in Method 3 is repeated except the four equations proposed in the Method 1. The objective of this method is to assess the performance the 2D-dwt using two-dimensional decomposition filters. Firstly, four different two-dimensional wavelet decomposition filters are assembled using the low-frequency analysis decomposition filter and high-frequency analysis decomposition filter. The Figure 4.2.5.a illustrates the process of creating the filters.

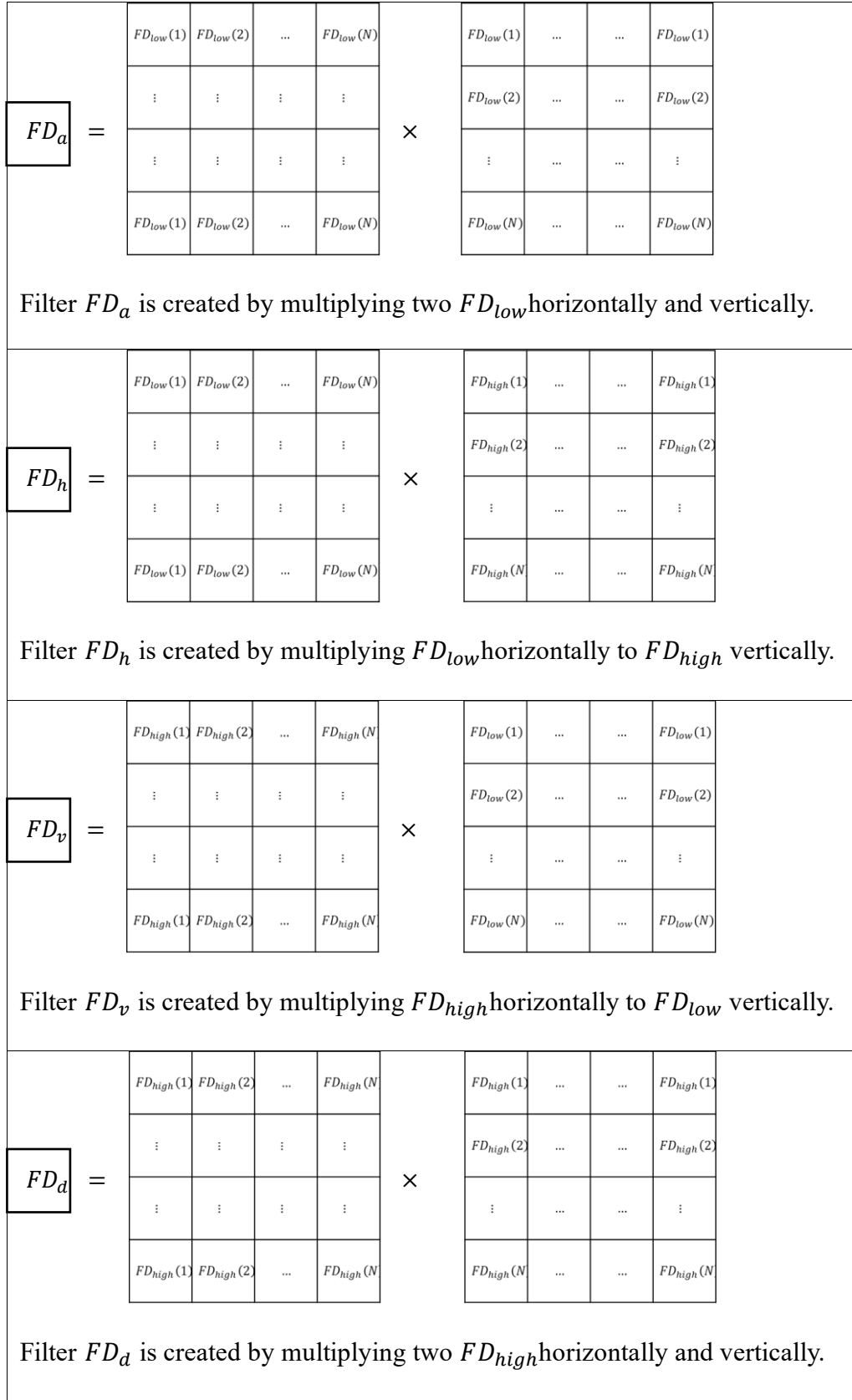


Figure 4.2.5.a: The four types of two-dimensional wavelet decomposition filters

The output of this repeated process resemble the wavelet coefficients of the 2D-dwt of the Matlab toolbox up to four decimal places. The results and discussions are available in section 5.1.5. There are four equations placed inside a single *for* loop of the program for convolution. The Matlab equations are listed below.

$L = \text{number of rows of the image}$

$K = \text{number of columns of the image}$

$$O_a = \sum_{w=1}^{\frac{L}{2}} \sum_{v=1}^{\frac{K}{2}} (O_a(v, w) + \text{small_area} \times FD_a) \quad (24)$$

$$O_h = \sum_{w=1}^{\frac{L}{2}} \sum_{v=1}^{\frac{K}{2}} (O_h(v, w) + \text{small_area}_{min} \times FD_h) \quad (25)$$

$$O_v = \sum_{w=1}^{\frac{L}{2}} \sum_{v=1}^{\frac{K}{2}} (O_v(v, w) + \text{small_area}_{min} \times FD_v) \quad (26)$$

$$O_d = \sum_{w=1}^{\frac{L}{2}} \sum_{v=1}^{\frac{K}{2}} (O_d(v, w) + \text{small_area}_{min} \times FD_d) \quad (27)$$

For $v, w = 1, 2, 3, 4 \dots$

This method emphasizes in the use of *for* loops and its performance in Matlab environment. This method uses *parfor* to achieve parallel processing.

4.2.6 Method 5

The process of 2D-dwt discussed in Method 4 is repeated in Method 5. The objective of this method is to increase the computation speed using Matlab pad array function to predetermine and pre-allocate a new matrix required for the wavelet transform. Multiple steps are required to produce the new matrix in the previous methods and performance need to be compared with the pad array function from Matlab.

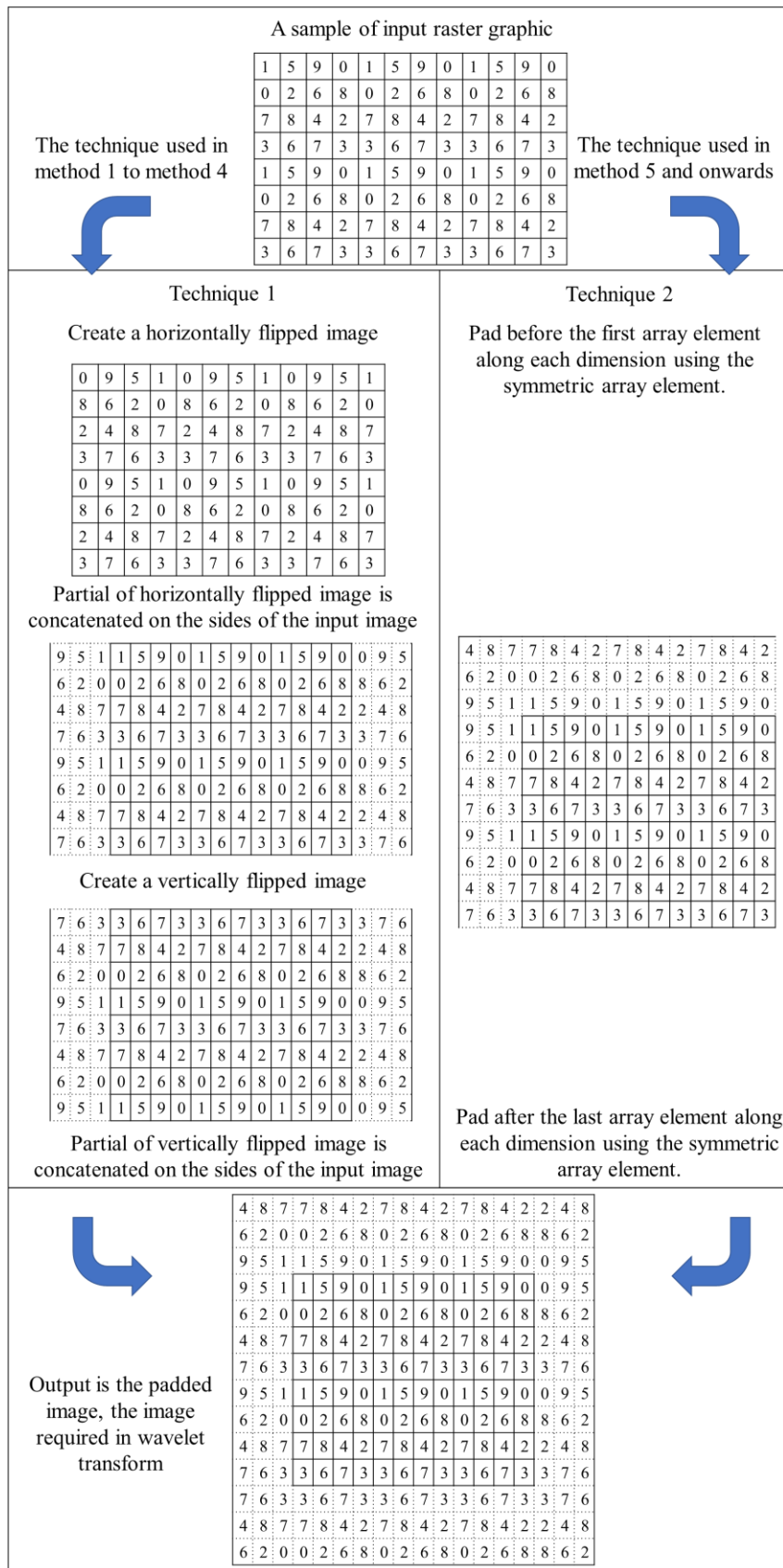


Figure 4.2.6.a: Illustration of the two techniques to create a new raster graphic proposed in this research

As shown in the Figure 4.2.6.a, a padded version of input image is created using two different techniques. The technique on the left requires more instruction lines compares to the technique on the right. It is noteworthy to mention that the beginning and the end of the image has to be concatenated or padded separately. The basis behind of this operation is the front boundaries only amended by the size of the decomposition filter. Whereas the rear boundaries depend on the dimensions of the input image and the size of the decomposition filter used. This step is indispensable to create the boundaries for the padded version of input image.

The Matlab equations (MATLAB, 2018) are listed below.

L = number of rows of the input image

K = number of columns of the input image

M = size of the wavelet decomposition filter

$$M_{height} = M + mod(L)$$

$$M_{width} = M + mod(K)$$

$$Image2_{padded} = padarray(Image, [M, M], 'symmetric', 'pre')$$

(28)

$$Image3_{padded} = padarray(Image2_{padded}, [M_{height}, M_{width}], 'symmetric', 'post')$$

(29)

The results and discussions of the computation time are available in section 5.1.6. This method emphasizes in the use of *for* loops and uses *spmd* to achieve parallel processing.

4.2.7 Method 6

The objective of this method is to reduce the number of loops by carrying out convolution on all pixels in the padded image using one element from the wavelet decomposition filter inside every loop. The loop controlling variable is determined by the length of wavelet filter instead of the size of the padded image. The process of 2D-dwt discussed in Method 5 is repeated in Method 6. The wavelet transforms repeated in Method 1, 2, 3, 4, and 5 calculate the coefficients via convolution on each and every small area taken out from padded image with all the elements in decomposition filter. The process repeat itself until it stops when reaches the last pixel of the padded image.

The Matlab equations are listed below.

$N, M = 2 \times \text{vanishing moments of the wavelet}$

$L = \text{number of rows of the padded image}$

$K = \text{number of columns of the padded image}$

$$area_{min} = \min \left(area_{min}, \sum_{y=1}^N \sum_{x=1}^M (Image3_{padded}(y + 1:2:L + y, x + 1:2:K + x)) \right) \quad (30)$$

$$O_a = O_a + \sum_{y=1}^N \sum_{x=1}^M (Image3_{padded}(y + 1:2:L + y, x + 1:2:K + x) \times FD_a(y, x)) \quad (31)$$

$$O_h = O_h + \sum_{y=1}^N \sum_{x=1}^M \left((Image3_{padded}(y + 1:2:L + y, x + 1:2:K + x) - area_{min}) \times FD_h(y, x) \right) \quad (32)$$

$$O_v = O_v + \sum_{y=1}^N \sum_{x=1}^M \left((Image3_{padded}(y + 1:2:L + y, x + 1:2:K + x) - area_{min}) \times FD_v(y, x) \right) \quad (33)$$

$$O_d = O_d + \sum_{y=1}^N \sum_{x=1}^M \left((Image3_{padded}(y + 1:2:L + y, x + 1:2:K + x) - area_{min}) \times FD_d(y, x) \right) \quad (34)$$

For $x, y = 1, 2, 3, 4 \dots$

This method uses minimal number of *for* loops and uses *parfor* to achieve parallel processing. The results and discussions of the computation time are available in section 5.1.7.

4.2.8 Method 7

The stumbling block of wavelet transforms used in from Method 1, 2, 3, 4, 5, and 6 is down sampling the spatial domain of the input raster graphics and generate a smaller two-dimensional of coefficients. Exploring the pixels in the input image using a consecutive manner reduces half of the edge and ridge information presented in the image. The objectives of Method 7 are to evaluate all the information available in the image and to generate all four layers of two-dimensional detailed wavelet coefficients which equal size to the input image.

The objective is accomplished by performing wavelet transforms on every row and every column of the input image. The algorithm proposed in Method 6 is modified to achieve the goals in Method 7. The purpose of proposing this method is because the down sampled wavelet coefficients are not usable in the colour threshold analysis. The colour threshold analysis requires the edges and ridges gathered from the wavelet coefficients align perfectly with the input image. Resizing or enlarging the wavelet coefficients produced by the previous methods is not recommended because the edges are blurred and position information are distorted.

The Matlab equations are listed below.

$N, M = 2 \times \text{vanishing moments of the wavelet}$

$L = \text{number of rows of the padded image}$

$K = \text{number of columns of the padded image}$

$$area_{min} = \min \left(area_{min}, \sum_{y=1}^N \sum_{x=1}^M (Image3_{padded}(y:1:L+y, x:1:K+x)) \right) \quad (35)$$

$$O_a = O_a + \sum_{y=1}^N \sum_{x=1}^M (Image3_{padded}(y+1:L+y, x+1:K+x) \times FD_a(y, x)) \quad (36)$$

$$O_h = O_h + \sum_{y=1}^N \sum_{x=1}^M ((Image3_{padded}(y+1:L+y, x+1:K+x) - area_{min}) \times FD_h(y, x)) \quad (37)$$

$$O_v = O_v + \sum_{y=1}^N \sum_{x=1}^M ((Image3_{padded}(y+1:L+y, x+1:K+x) - area_{min}) \times FD_v(y, x)) \quad (38)$$

$$O_d = O_d + \sum_{y=1}^N \sum_{x=1}^M ((Image3_{padded}(y+1:L+y, x+1:K+x) - area_{min}) \times FD_d(y, x)) \quad (39)$$

For $x, y = 1, 2, 3, 4 \dots$

This method uses *spmd* to achieve parallel processing and all results and discussions are available in section 5.1.8.

4.2.9 Method 8

Method 8 proposed comes with an objective to modify the wavelet coefficients into a better representation of the image through redesigning another wavelet transform technique. This method is expected to bring improvement to any of the 2D-dwt mentioned in the section 2.2. Every 2D-dwt that uses two-dimensional convolution, they sum up the real-values that are divided by the length of the filter obtained from multiplication. This is a big drawback because each coefficient carries the correlation information of the neighboring pixels. All the decomposed impulse information compacted inside a single coefficient might be useful in data compression and data transmission, nonetheless it a limitation in this research.

A two-dimensional convolution is an integral that expresses the amount of overlap of one input raster graphics with another matrix. The two-dimensional wavelet coefficients signify the impulse decomposition of the overlapping area of the input image and the wavelet decomposition filter. The dimension of the assessing area is determined by the length of the wavelet decomposition filter.

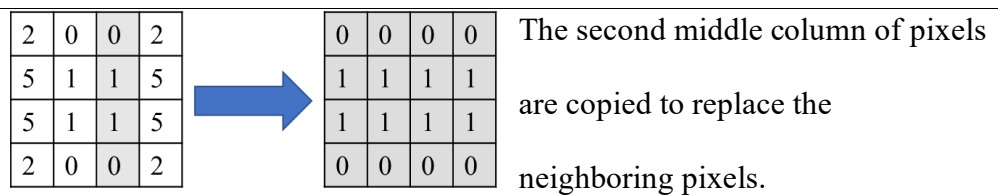
In this research, the proposed method is able to obliterate the unwanted correlation information of the neighboring pixels through replacing all the assessing pixels using the middle row of pixels and middle column of pixels of itself to creates wavelet coefficients which fixate at the center of the assessing area. There are three methods of assessing area required to determine the

horizontal detail coefficients, vertical detail coefficients, and the diagonal detail coefficients. The approximation coefficients matrix is calculated from the horizontal detail coefficients and vertical detail coefficients.

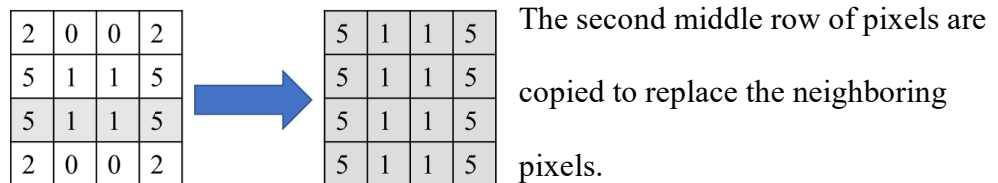
Figure 4.2.9.a below illustrates the formation of all three assessing areas with the dimensions determined by the length of the wavelet decomposition filter, $N = 4$ as an example.

4	8	7	7	8	4	2	7	8	4	2	7	8	4	2	2	4	8
6	2	0	0	2	6	8	0	2	6	8	0	2	6	8	8	6	2
9	5	1	1	5	9	0	1	5	9	0	1	5	9	0	0	9	5
9	5	1	1	5	9	0	1	5	9	0	1	5	9	0	0	9	5
6	2	0	0	2	6	8	0	2	6	8	0	2	6	8	8	6	2
4	8	7	7	8	4	2	7	8	4	2	7	8	4	2	2	4	8
7	6	3	3	6	7	3	3	6	7	3	3	6	7	3	3	7	6
9	5	1	1	5	9	0	1	5	9	0	1	5	9	0	0	9	5
6	2	0	0	2	6	8	0	2	6	8	0	2	6	8	8	6	2
4	8	7	7	8	4	2	7	8	4	2	7	8	4	2	2	4	8
7	6	3	3	6	7	3	3	6	7	3	3	6	7	3	3	7	6
7	6	3	3	6	7	3	3	6	7	3	3	6	7	3	3	7	6
4	8	7	7	8	4	2	7	8	4	2	7	8	4	2	2	4	8
6	2	0	0	2	6	8	0	2	6	8	0	2	6	8	8	6	2

The grey area represents the assessing area of the padded image for 2D-dwt



The horizontal detail coefficients are calculated from this H-assessing area.



The vertical detail coefficients are calculated from this V-assessing area.

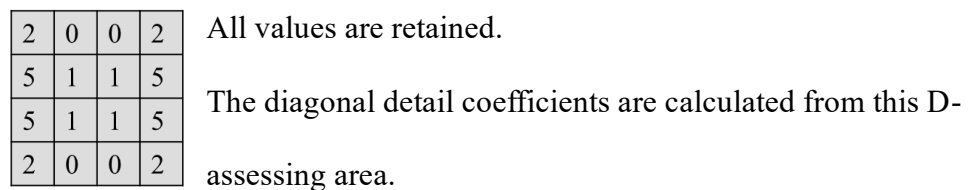


Figure 4.2.9.a: The H-assessing area, V-assessing area, and D-assessing area

for 2D-dwt

A Matlab function is applied on all three assessing areas to find the smallest value. The particular area is regulated aiming to set the offset of the smallest value to zero. The input, output, and the Matlab function (MATLAB, 2018) are shown below.

$$H_assessing_area_{min} = H_assessing_area - \min(H_assessing_area) \quad (40)$$

$$V_assessing_area_{min} = V_assessing_area - \min(V_assessing_area) \quad (41)$$

$$D_assessing_area_{min} = D_assessing_area - \min(D_assessing_area) \quad (42)$$

After obtaining the three different assessing areas, the 2D-dwt process uses one-dimensional wavelet filters to perform the convolution transformation. Method 8 uses *parfor* to achieve parallel processing, the results and discussions are available in section 5.1.9. The Matlab equations are listed below.

$N, M = 2 \times \text{vanishing moment of the wavelet}$

$L = \text{number of rows of the image}$

$K = \text{number of columns of the image}$

$$\begin{aligned} O_a &= \sum_{w=1}^L \sum_{v=1}^K \left(O_a(v, w) \right. \\ &+ \sum_{y=1}^N \sum_{x=1}^M \frac{(H_assessing_area(x, y) \times FD_{low}(x) \times FD_{low}(y))}{2} \\ &\left. + \sum_{y=1}^N \sum_{x=1}^M \frac{(V_assessing_area(x, y) \times FD_{low}(x) \times FD_{low}(y))}{2} \right) \end{aligned} \quad (43)$$

$$\begin{aligned}
O_h = \sum_{w=1}^L \sum_{v=1}^K & \left(O_h(v, w) \right. \\
& + \sum_{y=1}^N \sum_{x=1}^M H_assessing_area_{min}(x, y) \times FD_{low}(x) \\
& \left. \times FD_{high}(y) \right)
\end{aligned} \tag{44}$$

$$\begin{aligned}
O_v = \sum_{w=1}^L \sum_{v=1}^K & \left(O_v(v, w) \right. \\
& + \sum_{y=1}^N \sum_{x=1}^M V_assessing_area_{min}(x, y) \times FD_{high}(x) \\
& \left. \times FD_{low}(y) \right)
\end{aligned} \tag{45}$$

$$\begin{aligned}
O_d = \sum_{w=1}^L \sum_{v=1}^K & \left(O_d(v, w) \right. \\
& + \sum_{y=1}^N \sum_{x=1}^M D_assessing_area_{min}(x, y) \times FD_{high}(x) \\
& \left. \times FD_{high}(y) \right)
\end{aligned} \tag{46}$$

For $v, w, x, y = 1, 2, 3, 4 \dots$

4.2.10 Method 9

With Method 9, the process of 2D-dwt discussed in Method 8 is reorganized using two-dimensional decomposition filters. Four different two-dimensional wavelet decomposition filters assembled using the low-frequency analysis decomposition filter and high-frequency analysis decomposition filter as shown in Figure 4.2.5.a. This method will reduce the number of iterations while performing two-dimensional wavelet transform to the input image. The results and discussions are available in section 5.1.10 and this method uses *parfor* to achieve parallel processing. There are four equations placed inside a for loop of the algorithm for convolution. The Matlab equations are listed below.

$L = \text{number of rows of the image}$

$K = \text{number of columns of the image}$

$$O_a = \sum_{w=1}^L \sum_{v=1}^K \left(O_a(v, w) + \frac{(H_assessing_area \times FD_a)}{2} + \frac{(V_assessing_area \times FD_a)}{2} \right) \quad (47)$$

$$O_h = \sum_{w=1}^L \sum_{v=1}^K (O_h(v, w) + H_assessing_area_{min} \times FD_h) \quad (48)$$

$$O_v = \sum_{w=1}^L \sum_{v=1}^K (O_v(v, w) + V_assessing_area_{min} \times FD_v) \quad (49)$$

$$O_d = \sum_{w=1}^L \sum_{v=1}^K (O_d(v, w) + D_assessing_area_{min} \times FD_d) \quad (50)$$

For $v, w = 1, 2, 3, 4 \dots$

4.2.11 Method 10

Method 10 is a collection of Method 6, Method 7, and Method 8. Method 10 performs convolution on all pixels in the padded image using one element from the wavelet decomposition filter inside every single loop. The loop count is based on the length of the wavelet decomposition filter used. Both one-dimensional filters and two-dimensional filters are applicable in this method. However, more flexibilities and possible changes can be accommodated in the algorithm when using one-dimensional wavelet decomposition filter compare to two-dimensional wavelet decomposition filter.

The Matlab equations (MATLAB, 2018) listed below are used to compute the minimum value of the assessing areas.

$$N, M = 2 \times \text{vanishing moments of the wavelet}$$

$$L = \text{number of rows of the padded image}$$

$$K = \text{number of columns of the padded image}$$

$$H_area_{min} = \min \left(H_area_{min}, \sum_{y=1}^N \sum_{x=1}^M \left(Image3_{padded}(y:L + y - 1, \frac{N}{2}:K + \frac{N}{2}) \right) \right) \quad (51)$$

$$V_area_{min} = \min \left(V_area_{min}, \sum_{y=1}^N \sum_{x=1}^M \left(Image3_{padded}(\frac{N}{2}:L + \frac{N}{2}, x:K + x - 1) \right) \right) \quad (52)$$

$$D_area_{min} = \min \left(D_area_{min}, \sum_{y=1}^N \sum_{x=1}^M \left(Image3_{padded}(y:L + y - 1, x:K + x - 1) \right) \right) \quad (53)$$

For $y, x = 1, 2, 3, 4 \dots$

When two-dimensional wavelet decomposition filter is used, the block of code placed inside a *for* loop consists of four Matlab equations. The construction of the two-dimensional wavelet decomposition filter is identical to technique discussed in Method 4. The Matlab equations are listed below.

$$O_a = O_a + \sum_{y=1}^N \sum_{x=1}^M \left(\left(\frac{\text{Image3}_{padded}(y:L+y-1, \frac{N}{2}:K+\frac{N}{2})}{2} \right) \times FD_a(y, x) \right) + \sum_{y=1}^N \sum_{x=1}^M \left(\left(\frac{\text{Image3}_{padded}(\frac{N}{2}:L+\frac{N}{2}, x:K+x-1)}{2} \right) \times FD_a(y, x) \right) \quad (54)$$

$$O_h = O_h + \sum_{y=1}^N \sum_{x=1}^M \left((\text{Image3}_{padded}(y:L+y-1, \frac{N}{2}:K+\frac{N}{2}) - H_area_{min}) \times FD_h(y, x) \right) \quad (55)$$

$$O_v = O_v + \sum_{y=1}^N \sum_{x=1}^M \left((\text{Image3}_{padded}(\frac{N}{2}:L+\frac{N}{2}, x:K+x-1) - V_area_{min}) \times FD_v(y, x) \right) \quad (56)$$

$$O_d = O_d + \sum_{y=1}^N \sum_{x=1}^M \left((\text{Image3}_{padded}(y:L+y-1, x:K+x-1) - D_area_{min}) \times FD_d(y, x) \right) \quad (57)$$

For $x, y = 1, 2, 3, 4 \dots$

When one-dimensional wavelet decomposition filter is used, the block of code placed inside a *for* loop consists of five Matlab equations. *If – else* statements are inserted to the algorithm to reduce the load of computation. The Matlab equations are listed below.

$$O_a = O_a + \frac{\sum_{y=1}^N Image3_{padded}(y:L + y - 1, \frac{N}{2}:K + \frac{N}{2}) \times FD_{low}(y) \times sum(FD_{low})}{2} \quad (58)$$

$$O_a = O_a + \frac{\sum_{x=1}^M Image3_{padded}(\frac{N}{2}:L + \frac{N}{2}, x:K + x - 1) \times FD_{low}(x) \times sum(FD_{low})}{2} \quad (59)$$

$$O_h = O_h + \sum_{y=1}^N (Image3_{padded}(y:L + y - 1, \frac{N}{2}:K + \frac{N}{2}) - H_{area_{min}}) \times FD_{high}(y) \times sum(FD_{low}) \quad (60)$$

$$O_v = O_v + \sum_{x=1}^M (Image3_{padded}(\frac{N}{2}:L + \frac{N}{2}, x:K + x - 1) - V_{area_{min}}) \times FD_{high}(x) \times sum(FD_{low}) \quad (61)$$

$$O_d = O_d + \sum_{y=1}^N \sum_{x=1}^M \left((Image3_{padded}(y:L + y - 1, x:K + x - 1) - D_{area_{min}}) \times FD_{high}(x) \times FD_{high}(y) \right) \quad (62)$$

For $x, y = 1, 2, 3, 4 \dots$

The results and discussion are available in section 5.1.11 and this method uses *parfor* to achieve parallel processing.

4.3 Colour Threshold Analysis and Features Extraction

Colour threshold analysis comes right after the wavelet transform. It is a common method used in various image processing fields. The floating points in the two-dimensional detailed coefficients represent the smooth area or the discontinuities in image brightness before binarized the coefficients using the most appropriate threshold value. The coefficients that are greater among the others are likely to correspond to the change in object, depth, illumination and surface orientation. Thus, choosing the most proper threshold value is rarely achievable without supervision for real life images of moderate complexity. If the threshold is set correctly, the subsequent tasks therefore be much simplified. The background remover algorithm from Microsoft power-point, Adobe photoshop and image segmenter inside Matlab involve user to choose the region of interest and separating it from the background.

4.3.1 Histogram With 256 Palettes Discrete Domain Category

In this research, this threshold value is developed using a histogram shape-based method uses the peaks, valleys, and curvatures of the histogram. However, the method proposed in the research does not group the pixels intensity of the image into the intervals of the colour intensity. The proposed method uses the bins to accumulate the cardinal numbers obtained from the wavelet detailed coefficients. This method is accomplished using combination of several techniques.

Firstly, each colour component layer of the input image is separated and evaluated separately. Both input image and the detailed coefficients are partitioned into smaller sub areas to create a local bar chart for a local threshold value of the sub area. The accuracy of the local threshold values is primarily affected by the size of the sub area where the partitioning process of the image is adjustable by users. This research uses sub area with height is pre-set to be 5 times smaller than the input image to yield a better threshold value. The pre-set values are manageable variables by user from the list of initial variables, and are some basic setup parameters not vital to classification.

Subsequently, two layers of indexes for both horizontal detail coefficients and vertical detail coefficients are created. One is used to perform array accumulation for the wavelet coefficients below the mean values whereas the other layer is used to accumulate the wavelet coefficients above the mean values. The indexes that represent the wavelet coefficients above the mean value

are obtained by listing out a range of numbers at the discontinuities in image intensity indicated by the high wavelet coefficients. The algorithm is written using the Matlab equations (MATLAB, 2018) listed below.

diff = find the different between pixels, function in Matlab

le = less than and equal to a value, function in Matlab

lt = less than and not equal to a value, function in Matlab

repelem = repeat elements in an array, function in Matlab

cumsum = cummulative summation, function in Matlab

imtranslate = image translation, function in Matlab

$$O_{h2} = le(O_h, mean(O_h)) \quad (63)$$

$$O_{v2} = le(O_v, mean(O_v)) \quad (64)$$

$$D_h = diff(Image, 1, 1) \quad (65)$$

$$D_v = diff(Image, 1, 2) \quad (66)$$

$$Index_{h1low} = Image(O_{h2}) + \frac{D_h(O_{h2})}{2} \quad (67)$$

$$Index_{v1low} = Image(O_{v2}) + \frac{D_v(O_{v2})}{2} \quad (68)$$

Index_{h1high}

$$= min(256, max(1, (((0: sum(D_h(\sim O_{h2})) - 1)$$

$$- (repelem(cumsum(imtranslate(D_h(\sim O_{h2}), [1 0])), D_h(\sim O_{h2}))))).$$

$$* (repelem((lt(D_h(\sim O_{h2}), 0) * 2) - 1, D_h(\sim O_{h2}))))$$

$$+ (repelem(double(Image(\sim O_{h2}))', D_h(\sim O_{h2}))))$$

(69)

$$\begin{aligned}
& Index_{v1high} \\
& = \min(256, \max(1, (((0: \text{sum}(D_v(\sim O_{v2})) - 1) \\
& - (\text{repelem}(\text{cumsum}(\text{imtranslate}(D_v(\sim O_{v2}), [1 \ 0])), D_v(\sim O_{v2}))))). \\
& * (\text{repelem}((\text{lt}(D_v(\sim O_{v2}), 0) * 2) - 1, D_v(\sim O_{v2})))) \\
& + (\text{repelem}(\text{double}(\text{Image}(\sim O_{v2}))', D_v(\sim O_{v2}))))))
\end{aligned} \tag{70}$$

As mentioned in section 4.2, wavelet detail coefficients indicate the discontinuities or the intensity changes between the pixels. The information of the coefficients' value is undeniably suitable to detect the colour intensity between two adjacent pixels. Stacked bar chart method is used because each bar is made from the accumulation of wavelet coefficients into bins. A stacked bar chart method is used instead of group bar chart because the height of the bar chart is used for graph plotting. Thus, a graph with peaks and valleys is plotted based on the cardinal numbers accumulated inside the discrete domain category. The discrete domain category consists of 256 palettes represents the 8-bits colour intensities.

4.3.2 Features Scaling For Smooth Regions And Edges

A new approach of features scaling is applied in this research work to enhance the peaks' value. Firstly, wavelet coefficients that located below the mean value of the two-dimensional wavelet coefficients are subtracted by mean value calculated among the wavelet coefficients while the rest remain unchanged. This proposed method is designed to enhance the difference of the points' location in the graph to yield substantial peaks and valleys. The arrangement of the points forms peaks that represent the edges and valleys represent the smooth regions. The peaks inside the graph are entitled as threshold values that obtained from accumulating the detailed coefficients. Whereas the valleys refer to the colour of the smooth region. The Matlab equations listed below show the process of feature scaling and feature extraction creating a layer of binary image represents the edges and smooth regions.

accumarray = accumulate values to an array, function

ismember = set true if value exist in array , function

$$O_{h3} = O_h(O_{h2}) - \text{mean}(O_h) \quad (71)$$

$$O_{v3} = O_v(O_{v2}) - \text{mean}(O_v) \quad (72)$$

$$O_{array} = \text{accumarray}(\text{Index}_{h1low}, O_{h3}, [256 \ 1], [\], [\]) \\ + \text{accumarray}(\text{Index}_{v1low}, O_{v3}, [256 \ 1], [\], [\]) \quad (73)$$

$$\begin{aligned}
O_{array2} = & \text{accumarray}(Index_{h1high}, O_h(\sim O_{h2}), [256 \ 1], [\], [\]) \\
& + \text{accumarray}(Index_{v1high}, O_h(\sim O_{h2}), [256 \ 1], [\], [\])
\end{aligned}
\tag{74}$$

$$O_{binary} = \text{ismember}(Image, le(O_{array}, 0))
\tag{75}$$

The binary layer created identifies which portion of the image is edges or smooth regions based on 8-bit colour information. However, the information extracted is spread-out within the spatial domain and not grouped together. Thus, clustering method is applied.

Clustering method is then performed separately to each and every partition of the image. Clustering method is used because it is reliable to creates closed boundaries for every region of interest using the peak colour intensity obtained from the graph. The Matlab equations listed below exemplify the process of clustering the pixels using the O_{array2} . The O_{array2} is used to plot a graph using the *findpeaks* function that available in Matlab.

Peaks = number of peaks in the graph

$$Peaks_{Data} = \text{findpeaks}(O_{array2}, 0:255)
\tag{76}$$

$$Image_{cluster} = Image_{cluster}$$

$$+ \sum_{u=1}^{Peaks} Peaks_{Data} \times lt(Image3_{padded}, Peaks_{Data}(u))$$

(77)

$$u = 1,2,3 \dots$$

The number of clusters created in an image is depends on the number of peaks discovered using the stack bar chart method proposed previously. The clusters represent a group of pixels inside the image with colour intensities in between the peaks' value. Separation of image occur at both sharp edges and soft edges because the areas are bounded by specific colours in the colour space rather than only edge information. Every cluster created are made up of different shape and size. However, the spatial information is still unavailable and inaccessible after clustering.

The well-known edge detection methods such as Canny, Sobel, Prewitt, Roberts, Fuzzy-Logics, and Hough transform are not implemented in this research when wavelet transform has generated all the edges in the two-dimensional detailed wavelet coefficients layers. Whereas finding the difference between pixels, a widely used edge detection method is applied on the clustered image to generate a closed region boundary. This method is used to locate the position of the sharp edges. The Matlab equations (MATLAB, 2018) listed below exemplify the process.

abs = absolute value function in Matlab

$$Image_{h_edge} = padarray(abs(diff(image_{cluster}, 1, 1)), [1, 0], 0, 'pre')$$

(78)

$$Image_{v_edge} = padarray(abs(diff(image_{cluster}, 1, 1)), [0, 1], 0, 'pre')$$

(79)

Feature selection is then applied on the edges obtained previously. The binary layer, O_{binary} is used in removing edges of small clustered regions in

the image as shown in the Matlab equations below.

$$Image2_{h_edge} = Image_{h_edge} \& O_{binary} \quad (80)$$

$$Image2_{v_edge} = Image_{v_edge} \& O_{binary} \quad (81)$$

As part of the image processing, the process of image segmentation is unavoidable to yield the region of interest. This process converts the image into another format for analysis. There are various methods of image segmentation mentioned in section 3.2, such as thresholding method, Otsu's method, k-means clustering method, compression-based method, edge detection, region-growing method, markov random fields of graph partitioning method, and trainable segmentation.

Trainable segmentation is not considered in this research because it requires a huge domain knowledge database for deep learning neuro networks. Region growing is removed from the list because it measures the similarities with the neighboring pixels which are influenced by noise in all instances in each iteration from pixel to pixel. Otsu's method and automatic thresholding method are both disqualified because this research handles complicated real-time images acquired from the video surveillance cameras or web cameras. Both methods divide and convert input images into binary images by separating image into two parts, the background and the foreground. The threshold value often takes the mean value or the optimum separation of the two classes of the image for processing.

This research work utilizes a novel approach by combining few image segmentation methods together to make full use of the features available after the 2D-dwt to yield optimal quality sets of segments. Based on the four detail coefficients matrices obtained from the 2D-dwt, information it carries indicate the spatially coherent regions and the strength of correlation of the wavelet with the image. The higher basic number inside the coefficients matrices signify the sharp edges and ridges whereas the punier values refer to homogeneous regions. This information provided are often hampered by fragmentation, either discontinuous or over-detected that doesn't represent the closed curves of the actual region boundaries.

Thus, thresholding method, clustering method, and edge detection are merged together as a colour threshold analysis process to create set of segments with blob features. Meanwhile, new combination of methods for features selection, features scaling and feature extraction are implemented in the research to extract larger smooth regions with sharp edges.

4.4 Feature Extraction and Image Segmentation

Feature scaling is used because gradient descent converges faster and to normalize the range of independent variables or features of data. Size of image segments below the mean will be excluded for the next process. Feature selection used in choosing blob features which contribute most to the interested output area. Feature extraction is done by converting the blob features into nodes representation for graph partitioning method.

Image segmentation is performed using graph partitioning method to yield spatial information for the segments. This assemblage of image segmentation methods is guaranteed to converge where the quality is depending on the number of nodes plot on the clustered image in graph partitioning method. Graph based representation is proposed because it is more efficient than standard map-based representation due to nodes connections and intersections are signposted in mathematical ways rather than subpixels (Ramteke et al., 2011). The graphical representation provides important insights to handle the information.

The segments created from the image segmentation are stored in a list with the statistical properties of the segments, colour, intensity, texture, and location. All segments are substantial and not predefined in any category. They are referred as objects that are found in the image and are ready to use by the machine learning algorithm.

Graph partitioning method creates vector data models from the raster model generated previously while retain the geometrical properties. Points and nodes are created and linked in pairs to increase the ability to alter the scale of observation and analysis. The pair that is zero-dimensional object contain location information of horizontal or vertical vertices of the spatial domain. The connectivity of the nodes is deposited and easily accessible via a simple identification number that is given to each node. The graph plotted from the vector data tends to be much more complex as the location of each vertex has to be stored explicitly.

On the graph, polka dot pattern is used and each dot is assigned with an identification number. Each dot extends horizontally and vertically and stops when it reaches a vertex. The extension of the dots creates lines horizontally and vertically. Each line has a property of length and directly connect two nodes and stop at the vertex. From one end to another end of a line, multiple intersection points are created with the perpendicular lines. The identification number is compared at the intersection points and information of multiple nodes are shared and evaluated. Nodes are combined forming an area with size and boundaries information. The information is stored in an array for statistical and structural analysis.

4.5 Define Statistics, Structural and Physical Difference

Before categorizing each region of the image into classes using supervised classification, statistical and structural properties are evaluated to develop the process of classification. There are several types of classification, such as categorical, ordinal, adjectival and cardinal. The features available in the research are mainly extracted from a group of pixels that are similar to one another based on the spectral properties. The visual interpretation topologies are replicated in the classifications process using the colour, brightness, size, shape, and texture of the group of pixels. Colour is expressed in terms of intensities of its red, green and blue components. Extra information available in it can then be used to distinguish the uniqueness of the group of pixels. The size of the group of pixels is expressed using a two-dimensional flat plane. The aspect ratio of the group of pixels is the ratio of its sizes in different dimensions. It essentially described the shape of the region instead of actual size.

The group of pixels are identified as a single region associated with the spatial relationship of the surrounding pixels and physical properties. Features such as mean and standard deviations of the colour intensity, and more statistical properties are represented in numerical values.

4.6 Machine Learning and Object-Based Classification

As we look at an image, we can readily spot the objects. The goal of this research work is to teach the algorithm to do what comes naturally to us, to recognize and identify the objects visually available in the image. Object recognition is the technique used in image processing using deep learning and machine learning algorithms. Object detection is a subset of object recognition, that is the precise location of multiple objects inside the image is also identified.

Both deep learning and machine learning are applicable in this research. However, the deep learning was not the best approach to use in the research because of the unavailability of powerful GPU and huge database.

The general process by which we specify the features that matters in a given discrimination task is called supervised learning. We present the classification programs with examples and state the category each example belongs to, so it learns under supervision. In supervised classification the user supervises the pixel classification process. The user specifies the various pixels values or spectral signatures that should be associated with each class. These features are added to a machine learning model, which will separate these features into their distinct categories, and then use this information when analyzing and classifying new objects.

This is done by selecting representative sample sites of known cover type called Training Sites or Areas. The computer algorithm then uses the

spectral signatures from these training areas to classify the whole image. Ideally the classes should not overlap or should only minimally overlap with other classes. The accuracy of the process is significantly lowered particularly if the number of examples used in training is small. The biggest challenge is in handling the objects that does not belong to any pre-defined categories. There is a tendency of misdirecting the unknown object into any of the pre-defined categories.

Machine learning is utilized in this research by allowing the user to supervise the groups of classification. When the user chooses to add a new group into the algorithm, the algorithm can be easily trained by adding more groups to the application with training samples defined by the user. The user is given the control to overwrite the result by selecting the object. The user can name the group and define the selected object as a training sample. Whenever a new group is created, classification process such as decision trees and support vector machines (SVM) will get a new classification model and perform prediction.

Classification is the process of predicting the target class of the identified object by analyzing the training set. The training dataset forms boundary conditions that can be used to differentiate and group the object into the appropriate category. Classification can be broadly classified into linear classifiers, support vector machines (SVM), quadratic classifiers, decision trees, and neural networks. Decision trees and SVM are investigated in the research to assess the best classifier for feature extraction from wavelets for different kinds of images encountered in video surveillance sequences.

Decision tree is a type of supervised learning algorithm mostly used for classification problems and regression models. Decision trees can handle both categorical and numerical data, thus it is suggested to be implemented in the research. It breaks down a data set into smaller and smaller subsets and final result is a tree with decision nodes and leaf nodes. The topmost decision node in a tree which corresponds to the best predictor called root node. The root node of the research categorizes the object using the colour and brightness of the group of pixels following by the size of the pixels. With this technique, the groups that doesn't match with the features are eliminated at first stage, the root node. The sequential node is used to calculate the likelihood scores of input image to the sample datasets using a pattern detection function. Each internal node from root to leaf represents a test for specific classification rules.

In SVM, each data item of the features is plotted as a point in a n-dimensional space. A line or a boundary is being plotted to split the data between the two differently classified groups of data. This classification method uses Matlab *fitcsvm* function to train SVM model with three-colour components of RGB, statistical properties, size and shape features. The score generated by the Matlab *predict* function is used to classify the input image. The classification accuracies of both decision trees and SVM are investigated and evaluated in section 5.5.

4.7 Parallelization of Wavelet Transform

As mentioned in section 3.6, parallel computing splits and shares the intensive computational task to all workers in the processor for execution. Each processor performs the given task in parallel and return to the main processor once the task accomplished. Therefore, wavelet transform process is integrated with Matlab Parallel Computing Toolbox to enable the built-in multithreading and parallelism.

Matlab Wavelet Toolbox provides functions and apps for analyzing and synthesizing signals and images. The toolbox includes algorithms for discrete wavelet analysis for images to generate coefficients for various purposes. However, the algorithms are protected and non-editable by default. As discussed in section 4.2, numerous wavelet transform algorithms are written for assessments and to investigate the process of wavelet transform to achieve a better image classification.

Matlab provides a Parallel Computing Toolbox enables us to utilize the full processing power of multicore processors, GPUs, and computer clusters for data-intensive applications. The toolbox uses high-level programming language such as parallel for-loops, clusters, and distributed variables big data processing. Matlab provides two methods to take advantage of multicore and multiprocessor computers. The built-in multithreading and parallelism using Matlab workers can run any Matlab applications faster and more efficiently by using the full computational power of the system.

Matlab Parallel Computing Toolbox provides an interactive programming environment to accelerate the process by using *parfor* or *spmf* statement on multiple MATLAB workers in a parallel pool. With the highly data-intensive wavelet transform algorithms discussed in section 4.2, the toolbox uses the full processing power of multicore processor unit by executing the algorithms on all workers. Without changing the code of the wavelet transform, the loads are able to distribute across multiple Matlab workers by placing the block of code inside the parallel computing function statement. Furthermore, Matlab allows concurrent computing which inter-process communication is enabled during execution using the *lab send* and *receive* functions in distributed computing.

The general form of the *spmf* statement in Matlab coding environment is:

```
spmf  
  
    Block of code  
  
end
```

The *spmf* statement executes the *spmf* body on several Matlab workers simultaneously and only can be used only if Parallel Computing Toolbox is installed. Inside the statement, the block of code is assigned to a specific Matlab worker using the *lab index* value and the data return from the worker is stored in a composite object in Matlab environment. Each worker solves its part of the overall computational problem then Matlab reassembles the data to reach the end conclusion of the original complex problem.

The *parfor* statement of the Matlab Parallel Computing Toolbox distributes the number of iterations to all Matlab workers and significantly reduces the execution time of very large data calculations. It is because the *parfor* is different from the traditional for-loop in the way of consecutive manner and in parallel in a nondeterministic manner. The looping process must not dependent on another iteration because of the nondeterministic order. The Matlab client issues the command and coordinates with MATLAB workers to evaluate iterations in no particular order and independently of each other. In this case, a worker might receive multiple iterations at once and there is no guarantee that the iterations are synchronized in any way to reduce the communication time between the workers.

Linear algebra and some numerical functions are multithreaded as default in Matlab since the release of version 2008a. These functions execute faster on multi-core enabled machines using multiple computational threads in a single Matlab session. Whenever these functions are called within the *spmf* statement and *parfor* statement of Matlab Parallel Computing Toolbox, they are scaled back to single thread execution and thus the computation speed decreases.

CHAPTER 5

RESULTS AND DISCUSSION

5.1 Wavelet Transform

In this research work, wavelet coefficients play an important role in carrying necessary information for image processing. An in-depth research has been carried out to study the fundamental properties of wavelet coefficients as stated in section 4.2. The process of wavelet transform was elucidated in section 4 using Matlab wavelet toolbox, single-threaded algorithms, and multi-threaded algorithms to access the best solution in generating the faultless wavelet coefficients. Matlab Parallel Computing Toolbox is used in wavelet transform to improve the computation speed of wavelet transform for real time applications.

The wavelet transforms algorithms presented in this research work are useful in comparing the performance of wavelet transform with or without Parallel Computing Toolbox. With the aid of multiprocessor system, computation in which calculations or the execution of processes are possible to be carried out simultaneously to speed up the process of wavelet transform.

The computation time of wavelet transform was compared using different wavelet transform algorithms. Results were generated using the same set of wavelet filters, which are Haar wavelet, and Db3 wavelet. Both wavelet are chosen because they are conceptually simple, memory efficient, orthonormal and orthogonal.

Wavelet coefficients is a distinctive feature and the uniqueness of this feature requires a classification algorithm that can impeccably synthesizes it. The floating points from the two-dimensional detailed coefficients represents the change in object, depth, illumination and surface orientation. This information does not include the availability of an object or the number of objects captured in the image. Thus, the statistical properties computed from the wavelet coefficients could mean anything and not suggested for image classification. However, it is useful in defining the dispersion, shape of distribution, robustness, sharpness and complexity of the image. Using images from different scenes, the entropy, skewness, kurtosis, mean, and standard deviations were computed and listed in section 5.2.

5.1.1 Computation Time of Wavelet Transform and Matlab Wavelet Toolbox Function

In computing, computer performance is estimated in terms of accuracy, efficiency and speed of executing the algorithms or accomplishing the works of the program instructions (Sevcik, 1994; Abawajy, 2006). In this research work, the performance of the algorithm is vital in real-time application in term of processing speed. Thus, numerous wavelet transform methods and Matlab Parallel Computing Toolbox are studied and the performance are recorded in term of response time. The hardware used to compute all the results is an i7 laptop installed with Nvidia 860m with HDD hard disk. Performance of the algorithms varies based on the system and machines used.

Table 5.1.1.a : Computation time of Haar wavelet transform and Db3 wavelet transform recorded in second(s) with different resolutions

Resolution Wavelet	160x120	320x240	640x360	640x480	800x600	1280x720
Haar	0.00477	0.011243	0.050213	0.0956	0.149697	0.31284
db3	0.014945	0.01275	0.053594	0.105289	0.149585	0.327785



Graph 5.1.1.a : Computation time of Haar wavelet transform and Db3 wavelet transform recorded in second(s) with different resolutions

The Graph 5.1.1.a shows the computation time using Haar wavelet and Db3 wavelet on the same image in different resolution. The computation time increases gradually together with the resolution of the image. The data shows that the wavelet transform process can be completed within 0.35 seconds for the image resolution of 1280x720. Wavelet toolbox analyzes and synthesizes the image, generate wavelet coefficients using the built-in multithreading functions. The functions optimized the use of multiprocessors to deliver the results with short period of time shown in Table 5.1.1.a.

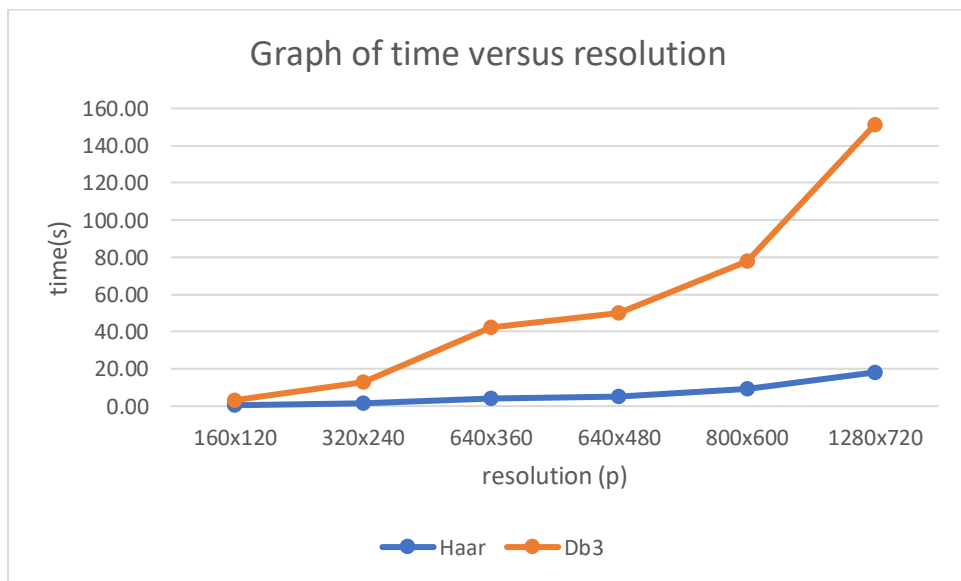
With the limitation of using the wavelet transform function provided by the Matlab wavelet toolbox, this research begins with reprogramming the wavelet transform algorithm using the Matlab workspace. Wavelet transform algorithm presented in section 4.2.2 is studied and assessed to ensure the coefficients are the same compared to the Matlab wavelet toolbox function.

Validation of the new algorithm is done by subtracting the actual values with the experimental values of wavelet coefficients from both methods. The differences between the two layers of wavelet coefficients were recorded in two-dimensional standard-form scientific notation with the given number is scaled by a power of 10. The absolute error for each pixel in the two-dimensional wavelet coefficients is always below $\pm 10^{-9}$. The relative error is relatively low and approaches zero when computed using the absolute error divided by the actual values.

5.1.2 Method 1

Table 5.1.2.a: Computation time of Haar wavelet transform and Db3 wavelet transform recorded in second(s) with different resolutions without using Parallel Computing Toolbox

Resolution \ Wavelet	160x120	320x240	640x360	640x480	800x600	1280x720
Haar	0.49	1.34	3.99	5.31	9.09	18.15
db3	3.23	12.73	42.37	49.85	77.69	151.24



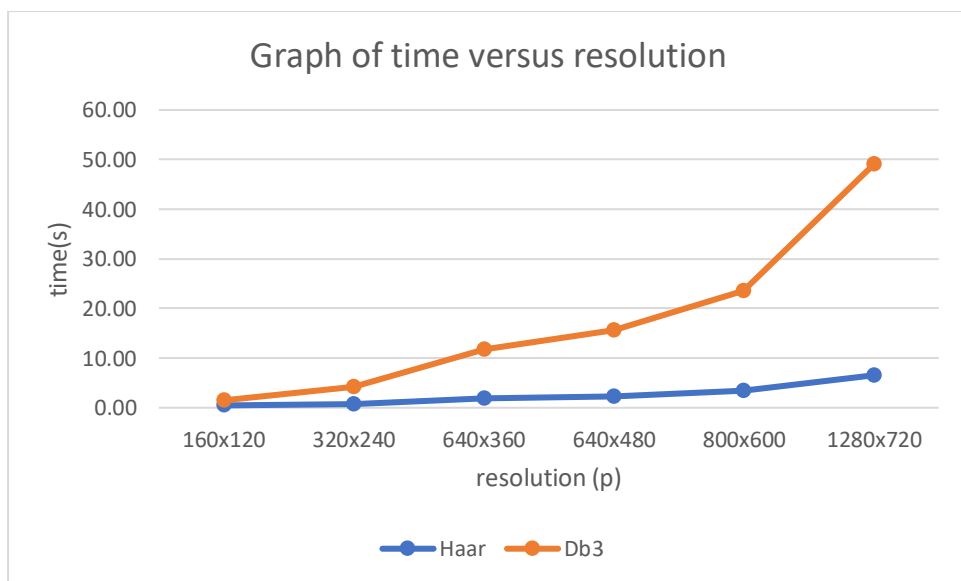
Graph 5.1.2.a: Computation time of Haar wavelet transform and Db3 wavelet transform recorded in second(s) with different resolutions without using Parallel Computing Toolbox

This method mostly consists of *for* loops in a nested formation. Each loop and every instruction execute in a single sequence which is executed one at a time. Thus, the computation time are much higher compared to the Matlab wavelet toolbox function. The computation time recorded is approximately 150

seconds to complete whole wavelet transform.

Table 5.1.2.b: Computation time of Haar wavelet transform and Db3 wavelet transform recorded in second(s) with different resolutions using Parallel Computing Toolbox

Resolution \ Wavelet	160x120	320x240	640x360	640x480	800x600	1280x720
Haar	0.48	0.78	1.94	2.32	3.37	6.54
db3	1.50	4.22	11.80	15.66	23.58	49.04



Graph 5.1.2.b: Computation time of Haar wavelet transform and Db3 wavelet transform recorded in second(s) with different resolutions using Parallel Computing Toolbox

With the aid from Matlab parallel computing toolbox, the result in Graph 5.1.2.b shows reduction in computation time from 150 seconds to 50 seconds for image with resolution 1280x720. Comparing the Graph 5.1.2.b with Graph 5.1.2.a, the reduction in computation time is observed for all available resolutions. The *Spmx* statement is used to initialize the parallel computing

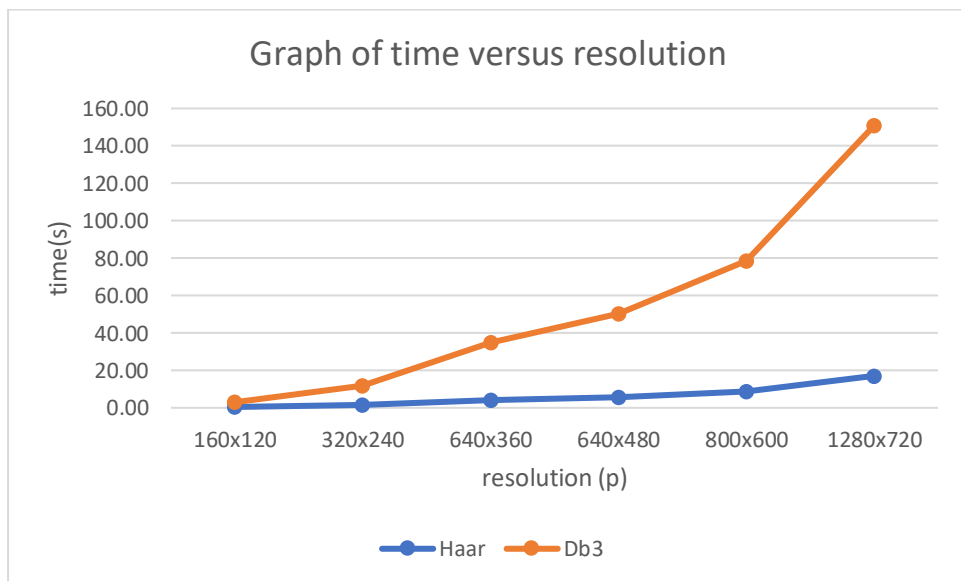
toolbox with minimal effort. The block of code from Method 1 is placed within the `spmd` body to distribute computations across available parallel computing resources without changing the Method 1 algorithm. The results indicate that multiprocessing will speed up and complete the wavelet transform 3 times faster to generate wavelet coefficients.

The average relative error for Haar wavelet comparing parallel computing and without parallel computing is 1.09 and average relative error for Db3 wavelet is 2.05.

5.1.3 Method 2

Table 5.1.3.a: Computation time of Haar wavelet transform and Db3 wavelet transform recorded in second(s) with different resolutions without using Parallel Computing Toolbox

Resolution \ Wavelet	160x120	320x240	640x360	640x480	800x600	1280x720
Haar	0.44	1.60	4.11	5.73	8.94	17.07
db3	2.99	11.82	35.13	50.29	78.42	150.79



Graph 5.1.3.a: Computation time of Haar wavelet transform and Db3 wavelet transform recorded in second(s) with different resolutions without using Parallel Computing Toolbox

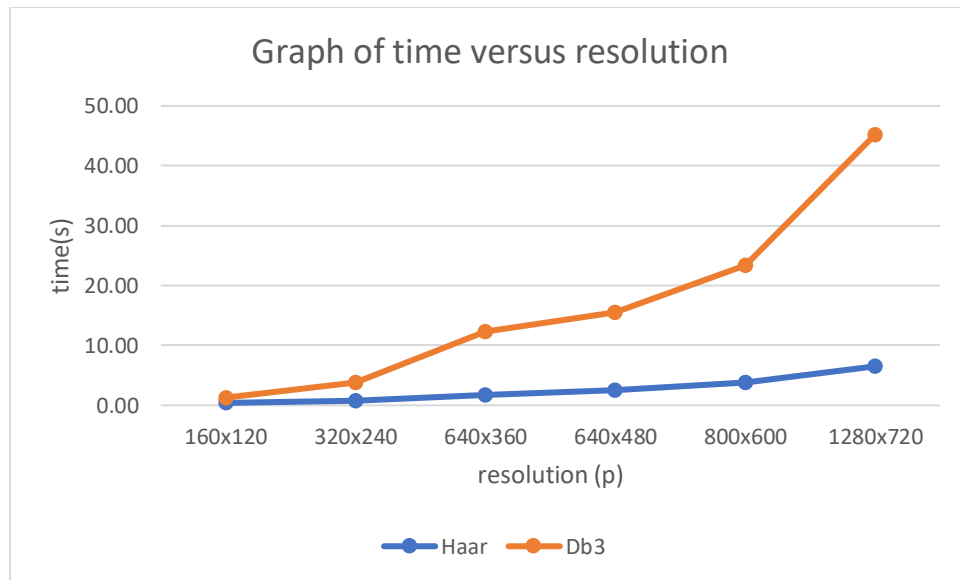
Two most commonly use statements for iteration, that is *for* loop and *while* loop, in Matlab high-level language programming environment were evaluated in this research work. Method 2 uses *while* loop instead of *for* loop inside the wavelet transform to compare the result with Method 1. Both loops allow any given set of instructions to be executed repeatedly till a specific

condition is true. However, they performed differently in the machine code level and have a negligible effect on performance. The assembly language deals with the registers, memory addresses and critical portions of a program which barely understandable by programmers.

The computation time recorded in Table 5.1.3.a are compared with Table 5.1.2.a. There is a percent error of 1.41% shows that the *while* loop overall performance is better compared with for loop.

Table 5.1.3.b: Computation time of Haar wavelet transform and Db3 wavelet transform recorded in second(s) with different resolutions using Parallel Computing Toolbox

Resolution \ Wavelet	160x120	320x240	640x360	640x480	800x600	1280x720
Haar	0.40	0.76	1.76	2.49	3.81	6.51
db3	1.29	3.83	12.37	15.45	23.45	45.21



Graph 5.1.3.b: Computation time of Haar wavelet transform and Db3 wavelet transform recorded in second(s) with different resolutions using Parallel Computing Toolbox

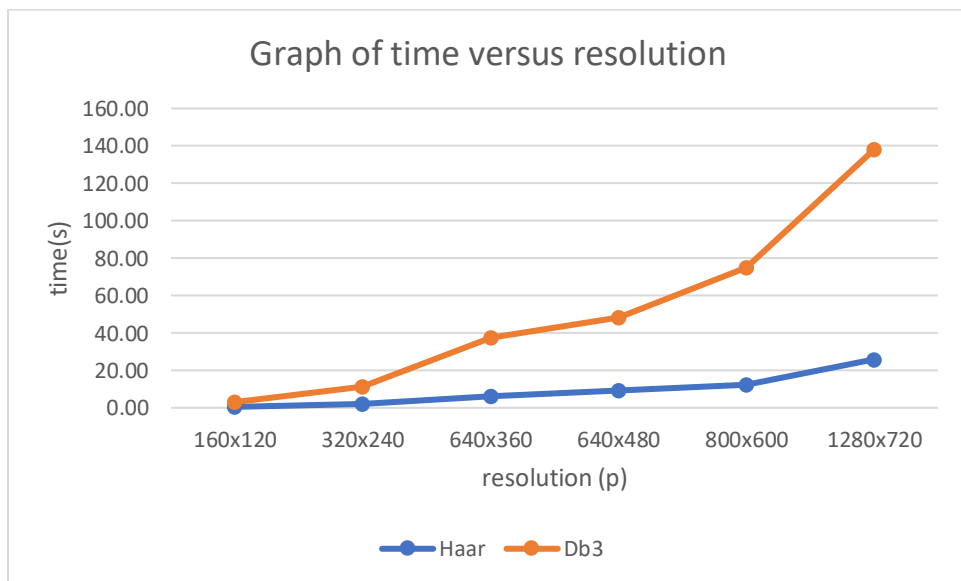
Result from Table 5.1.3.b is compared with Table 5.1.2.b to examine the outcome of using *for* and *while* statement together with parallel computing toolbox. The result shows that multiprocessing with *while* statement is faster compared with *for* statement. In parallel computing, there is an overall improvement of 2.87% in Method 2 compared with Method 1.

The average relative error for Haar wavelet comparing parallel computing and without parallel computing is 1.13 and average relative error for Db3 wavelet is 2.03.

5.1.4 Method 3

Table 5.1.4.a: Computation time of Haar wavelet transform and Db3 wavelet transform recorded in second(s) with different resolutions without using Parallel Computing Toolbox

Resolution	160x120	320x240	640x360	640x480	800x600	1280x720
Wavelet						
Haar	0.49	1.99	5.97	9.25	12.16	25.89
db3	3.05	11.49	37.29	48.22	75.05	138.04



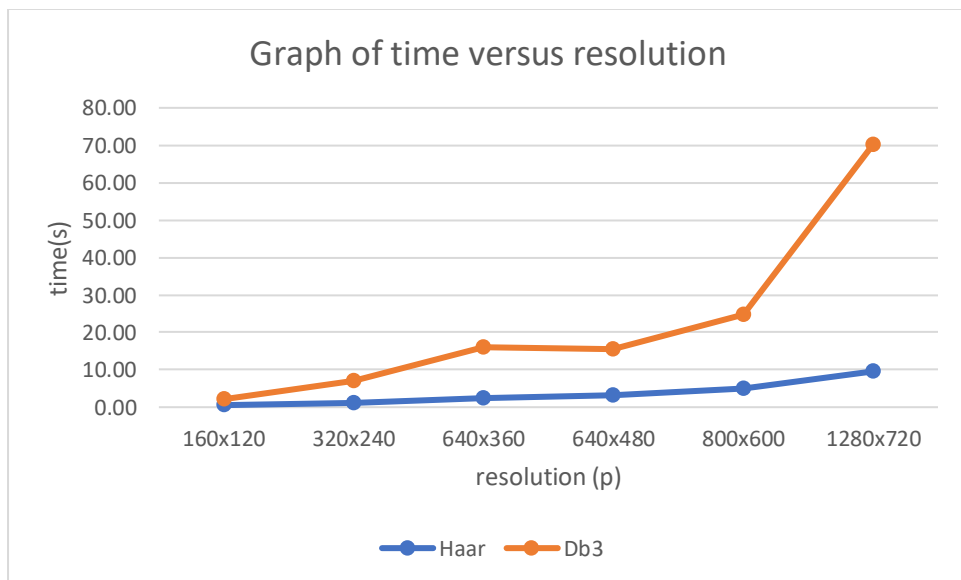
Graph 5.1.4.a: Computation time of Haar wavelet transform and Db3 wavelet transform recorded in second(s) with different resolutions without using Parallel Computing Toolbox

Percent error of computation time of Haar wavelet transform and Db3 wavelet transform are computed separately based on Table 5.1.4.a and Table 5.1.2.a. From the table and graph, it can be observed that there is a deterioration in computation time of Haar wavelet transform with percent error of 41.51%

whereas with Db3 wavelet transform there is an improvement of 7.11%. The result shows that the loop controlling variables manipulated by instructions lines slumped the performance of Haar wavelet transform. On the other hand, Db3 wavelet transform with a larger loop controlling variables perform better.

Table 5.1.4.b: Computation time of Haar wavelet transform and Db3 wavelet transform recorded in second(s) with different resolutions using Parallel Computing Toolbox

Resolution \ Wavelet	160x120	320x240	640x360	640x480	800x600	1280x720
Haar	0.54	1.12	2.45	3.13	4.94	9.56
db3	2.13	7.15	15.90	15.53	24.73	70.18



Graph 5.1.4.b: Computation time of Haar wavelet transform and Db3 wavelet transform recorded in second(s) with different resolutions using Parallel Computing Toolbox

Percent error of computation time of Haar wavelet transform and Db3 wavelet transform are computed separately based on Table 5.1.4.b and Table

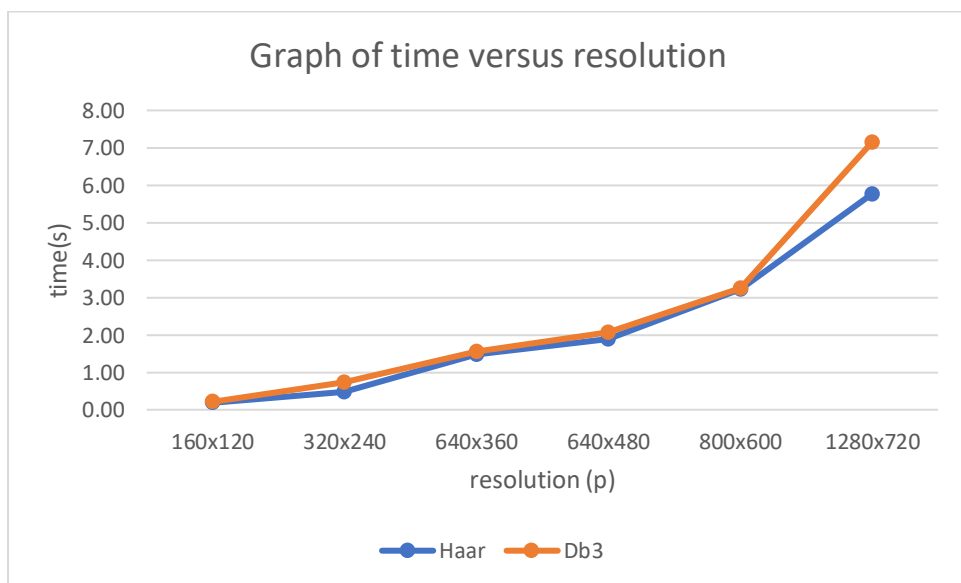
5.1.2.b. The results show a 35.53% slump in performance of Haar wavelet transform and a 32.28% slump in performance of Db3 wavelet transform. Parallel computing for Method 3 algorithm shows a downturn for both type of wavelet transform. The loop controlling variables manipulated by instructions lines slumped the performance of wavelet transform when it cannot manipulate the registers, memory addresses and critical portions of a program.

The average relative error for Haar wavelet comparing parallel computing and without parallel computing is 1.24 and average relative error for Db3 wavelet is 1.25.

5.1.5 Method 4

Table 5.1.5.a: Computation time of Haar wavelet transform and Db3 wavelet transform recorded in second(s) with different resolutions without using Parallel Computing Toolbox

Resolution \ Wavelet	160x120	320x240	640x360	640x480	800x600	1280x720
Haar	0.19	0.49	1.49	1.90	3.23	5.77
db3	0.21	0.74	1.55	2.07	3.24	7.15



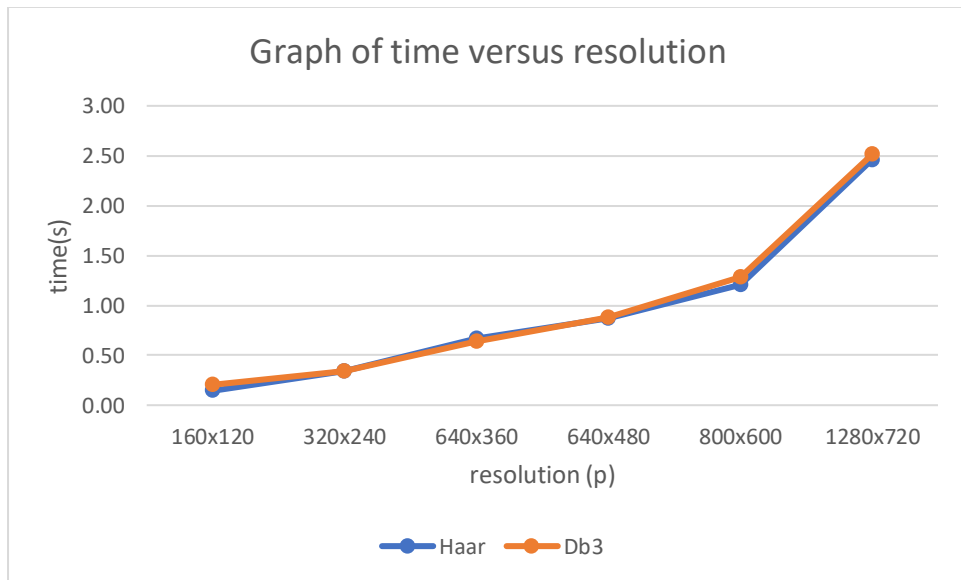
Graph 5.1.5.a: Computation time of Haar wavelet transform and Db3 wavelet transform recorded in second(s) with different resolutions without using Parallel Computing Toolbox

When two-dimensional wavelet filters were used, the Db3 wavelet transform can be accomplished in 7 seconds without using parallel computing toolbox for image with resolution of 1280x720. The computation time of Haar wavelet transform and Db3 wavelet transform in Method 4 are compared with

Method 1. The overall percent error of the computation time is 79.68%. The result indicates that uses two-dimensional wavelet filters are much more efficient compared with one-dimensional wavelet filters due to reduction in the number of loops used during wavelet transform.

Table 5.1.5.b: Computation time of Haar wavelet transform and Db3 wavelet transform recorded in second(s) with different resolutions using Parallel Computing Toolbox

Resolution \ Wavelet	160x120	320x240	640x360	640x480	800x600	1280x720
Haar	0.14	0.34	0.67	0.87	1.20	2.46
db3	0.20	0.34	0.64	0.88	1.29	2.51



Graph 5.1.5.b: Computation time of Haar wavelet transform and Db3 wavelet transform recorded in second(s) with different resolutions using Parallel Computing Toolbox

Based on the result gathered in Graph 5.1.5.b, the maximum computation time used for both Haar wavelet and Db3 wavelet does not exceed

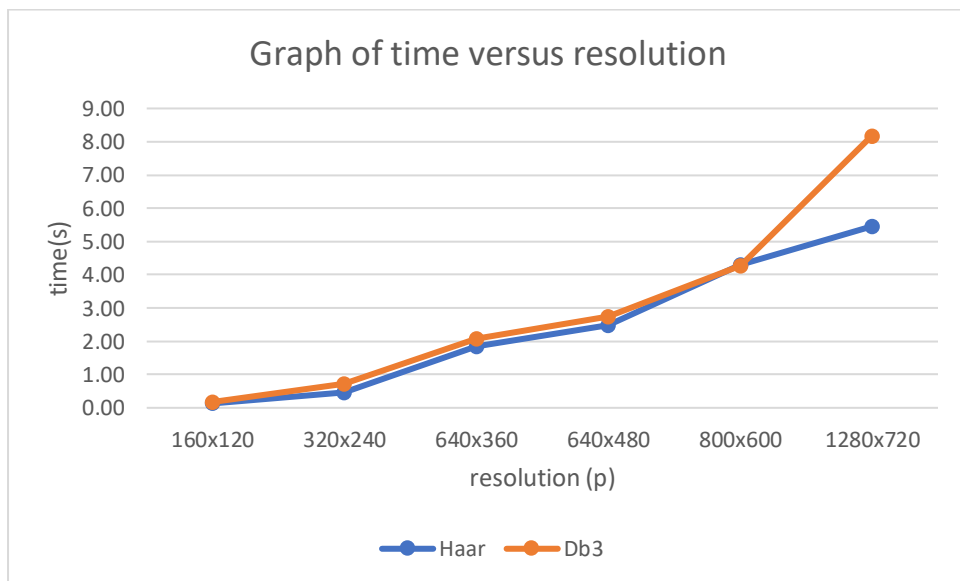
2.5 seconds with the biggest available resolution of 1280x720. The result in Table 5.1.5.b is compared with Method 1 Table 5.1.2.b to find the overall percent error, which is 78.08% better with the use of parallel computing toolbox.

The average relative error for Haar wavelet comparing parallel computing and without parallel computing is 1.02 and average relative error for Db3 wavelet is 1.22.

5.1.6 Method 5

Table 5.1.6.a: Computation time of Haar wavelet transform and Db3 wavelet transform recorded in second(s) with different resolutions without using Parallel Computing Toolbox

Resolution \ Wavelet	160x120	320x240	640x360	640x480	800x600	1280x720
Haar	0.13	0.45	1.85	2.48	4.30	5.46
db3	0.18	0.71	2.07	2.75	4.29	8.18



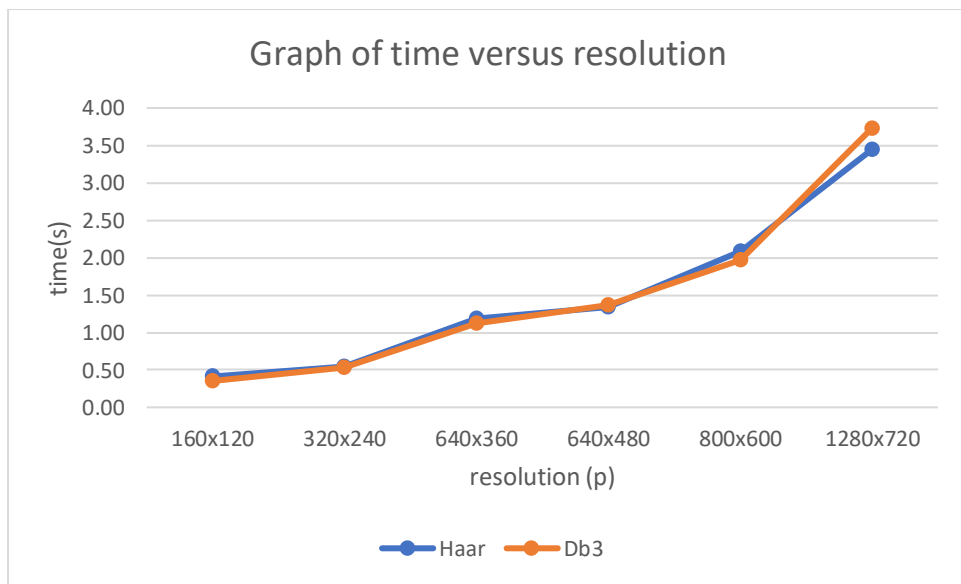
Graph 5.1.6.a: Computation time of Haar wavelet transform and Db3 wavelet transform recorded in second(s) with different resolutions without using Parallel Computing Toolbox

As mentioned in methodology section 4.2.6, performance of both algorithms in Method 4 and Method 5 was evaluated in measurement of computation time. The overall percent error between Table 5.1.5.a and Table 5.1.6.a is 11.63% for Matlab pad array function is slower compared with subroutine created by lines of code. Matlab function is much easier to use, but

it takes extensive time to accomplish the same tasks.

Table 5.1.6.b: Computation time of Haar wavelet transform and Db3 wavelet transform recorded in second(s) with different resolutions using Parallel Computing Toolbox

Resolution \ Wavelet	160x120	320x240	640x360	640x480	800x600	1280x720
Haar	0.41	0.54	1.19	1.35	2.09	3.45
db3	0.35	0.54	1.12	1.37	1.97	3.73



Graph 5.1.6.b: Computation time of Haar wavelet transform and Db3 wavelet transform recorded in second(s) with different resolutions using Parallel Computing Toolbox

Table 5.1.6.b displays the computation time using multiprocessing wavelet transform together with the Matlab pad array function placed outside the *spmf* body. In other words, the pad array function is not distributed to different core by the parallel computing toolbox. The overall percent error is computed from Table 5.1.5.b and Table 5.1.6.b. There is a 70.88% increase in

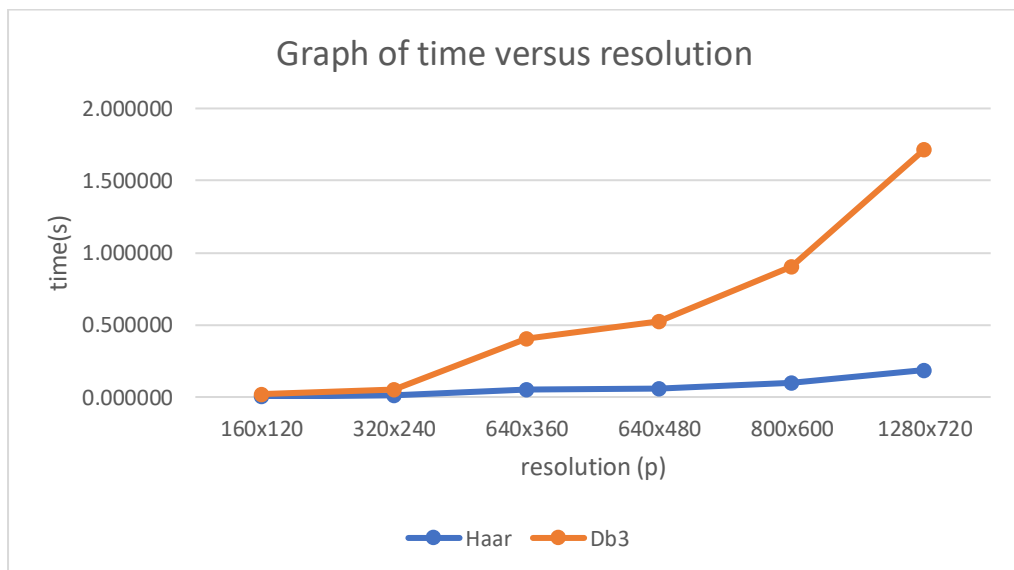
computation time of Method 5 compared with Method 4 using parallel computing toolbox.

The average relative error for Haar wavelet comparing parallel computing and without parallel computing is 0.65 and average relative error for Db3 wavelet is 0.84.

5.1.7 Method 6

Table 5.1.7.a: Computation time of Haar wavelet transform and Db3 wavelet transform recorded in second(s) with different resolutions without using Parallel Computing Toolbox

Resolution \ Wavelet	160x120	320x240	640x360	640x480	800x600	1280x720
Haar	0.002009	0.010439	0.053134	0.059507	0.096894	0.187154
db3	0.020224	0.051310	0.402390	0.523574	0.904464	1.715621



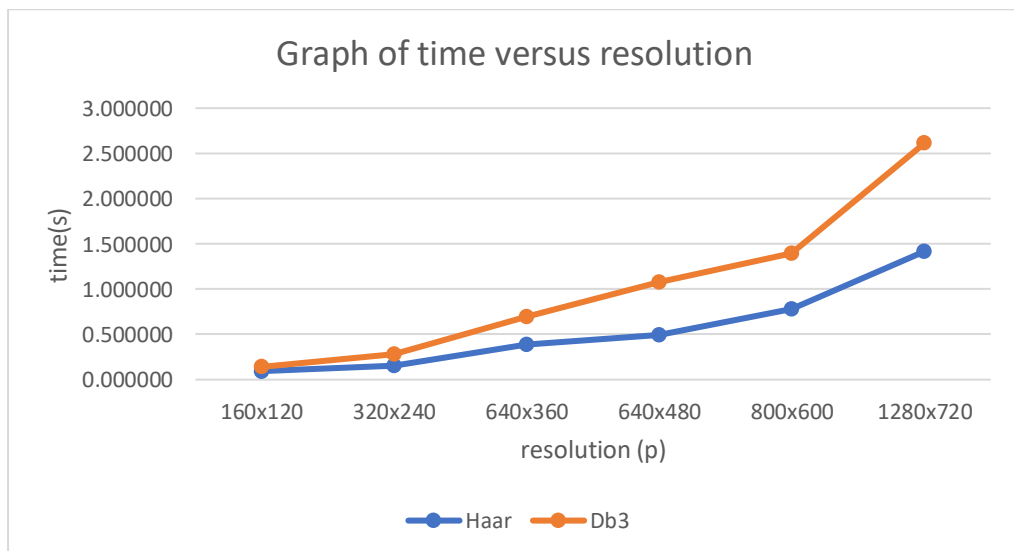
Graph 5.1.7.a: Computation time of Haar wavelet transform and Db3 wavelet transform recorded in second(s) with different resolutions without using Parallel Computing Toolbox

Based on the results obtained from section 5.1.2 until section 5.1.6, the number of iterations has a significant effect on the computation time of wavelet transform either with or without parallel computing toolbox. Thus, Method 6 is proposed to study the computation time of wavelet transform using different iteration method as mentioned in section 4.2.7.

After the loop controlling variable was changed, Table 5.1.7.a is created and compared with Table 5.1.6.a to compute the overall percent error. The percent error shows a 90.50% reduction in the computation time in wavelet transform. The statistical properties of wavelet coefficients from different input images are recorded and discussed in section 5.2.

Table 5.1.7.b: Computation time of Haar wavelet transform and Db3 wavelet transform recorded in second(s) with different resolutions using Parallel Computing Toolbox

Resolution \ Wavelet	160x120	320x240	640x360	640x480	800x600	1280x720
Haar	0.094414	0.157114	0.390074	0.494169	0.778015	1.416362
db3	0.141023	0.277730	0.699751	1.081355	1.400659	2.611159



Graph 5.1.7.b: Computation time of Haar wavelet transform and Db3 wavelet transform recorded in second(s) with different resolutions using Parallel Computing Toolbox

In addition, Table 5.1.7.b is compared with Table 5.1.6.b to find the overall percent error with *parfor* statement from parallel computing toolbox. The wavelet transform algorithm from Method 6 is placed inside the body of *parfor* statement for multiprocessing. The percent error is 52.17% shorter in computation time with multiple processors Method 6 compared to Method 5.

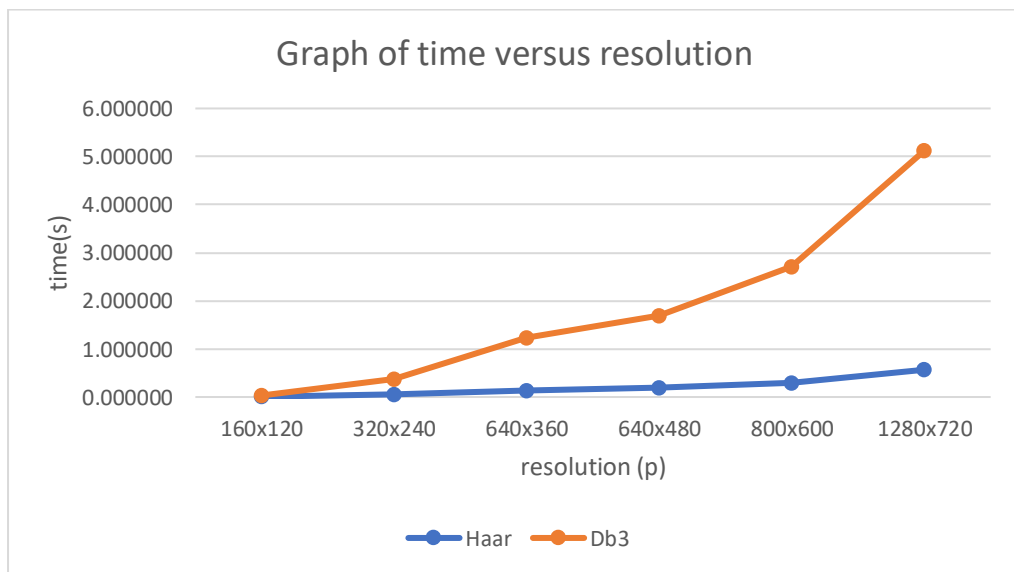
Table 5.1.7.b is compared with Table 5.1.7.a to inspect the different in computation time using different parallel computing methods. The overall percent error shows an increase of computation time as much as 8.38 times faster with the use of *parfor* statement. The drop-in computation time comprised of the process of dividing the workloads and distributing to all workers and assembling the results from all workers.

The average relative error for Haar wavelet comparing parallel computing and without parallel computing is 0.90 and average relative error for Db3 wavelet is 0.55.

5.1.8 Method 7

Table 5.1.8.a: Computation time of Haar wavelet transform and Db3 wavelet transform recorded in second(s) with different resolutions without using Parallel Computing Toolbox

Resolution \ Wavelet	160x120	320x240	640x360	640x480	800x600	1280x720
Haar	0.006213	0.049392	0.141563	0.195969	0.297125	0.566550
db3	0.031359	0.375399	1.238963	1.696561	2.702007	5.122547



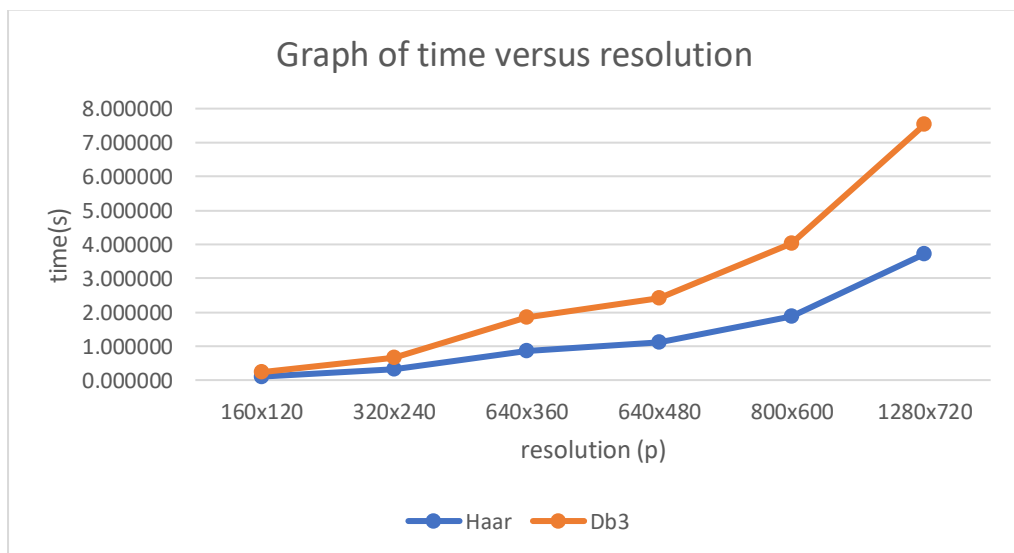
Graph 5.1.8.a: Computation time of Haar wavelet transform and Db3 wavelet transform recorded in second(s) with different resolutions without using Parallel Computing Toolbox

The computation time of wavelet transform of every row and column of the image is recorded in Table 5.1.8.a. The result is compared with Table 5.1.7.a and overall percent error shows 2.42 times increased in computation time. This is primarily caused by the size of the input image used in wavelet transform. This upsurge of computation time which is unavoidable due to computation of

larger two-dimensional wavelet coefficients with more details and information. The statistical properties of the wavelet coefficients of different type of images are available in section 5.2. Sharpness and image quality were discussed in the aspect of statistical values.

Table 5.1.8.b: Computation time of Haar wavelet transform and Db3 wavelet transform recorded in second(s) with different resolutions using Parallel Computing Toolbox

Resolution \ Wavelet	160x120	320x240	640x360	640x480	800x600	1280x720
Haar	0.103250	0.317479	0.869520	1.113099	1.872058	3.714766
db3	0.233501	0.654243	1.847784	2.410127	4.026295	7.523555



Graph 5.1.8.b: Computation time of Haar wavelet transform and Db3 wavelet transform recorded in second(s) with different resolutions using Parallel Computing Toolbox

The performance of parallel computing on this method were examined and tabulated in Table 5.1.8.b. Percent error is computed from Table 5.1.8.b with Table 5.1.7.b. With the parallel computing toolbox, there is 1.27 times increased

in computation time compared with Method 6. Another comparison between Table 5.1.8.a and Table 5.1.8.b shows 4.23 times increased in computation time due to distribution and collection of resources in message passing interface.

The average relative error for Haar wavelet comparing parallel computing and without parallel computing is 0.86 and average relative error for Db3 wavelet is 0.43.

5.1.9 Method 8

Table 5.1.9.a: Computation time of Haar wavelet transform and Db3 wavelet transform recorded in second(s) with different resolutions without using Parallel Computing Toolbox

Resolution \ Wavelet	160x120	320x240	640x360	640x480	800x600	1280x720
Haar	1.47	6.30	17.99	25.26	38.09	72.58
db3	17.88	68.61	196.27	256.59	412.22	786.23



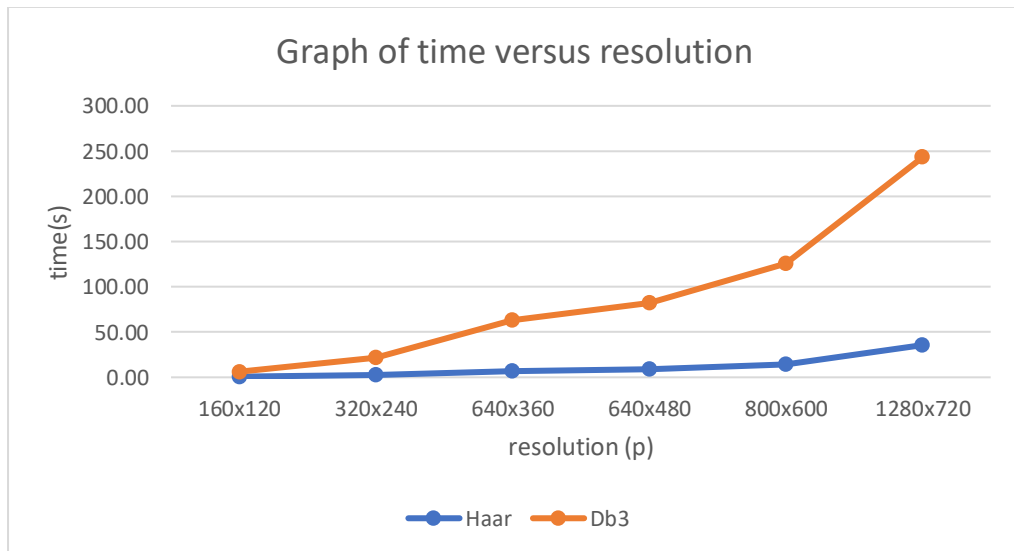
Graph 5.1.9.a: Computation time of Haar wavelet transform and Db3 wavelet transform recorded in second(s) with different resolutions without using Parallel Computing Toolbox

Table 5.1.9.a records the computation time of wavelet transform with loop controlling variables defined by the size of input image. The computation time of db3 wavelet transform on image with resolution 1280x720 reaches 786 seconds whereas Haar wavelet transform uses 72 seconds to complete the tasks. Most of the tasks in the previous methods are computed using multithreaded

multiplications, for this part of the research work, the tasks were substituted by single-threaded loops when one-dimensional wavelet filters were used for wavelet transform. Parallel computing toolbox is utilized to achieve multiprocessing.

Table 5.1.9.b: Computation time of Haar wavelet transform and Db3 wavelet transform recorded in second(s) with different resolutions using Parallel Computing Toolbox

Resolution \ Wavelet	160x120	320x240	640x360	640x480	800x600	1280x720
Haar	0.77	2.56	6.79	8.56	14.31	35.58
db3	6.21	21.40	62.75	82.61	125.42	242.90



Graph 5.1.9.b: Computation time of Haar wavelet transform and Db3 wavelet transform recorded in second(s) with different resolutions using Parallel Computing Toolbox

When the algorithm placed inside the *spmd* body of Parallel Computing Toolbox, the workloads were distributed to all the available workers of the

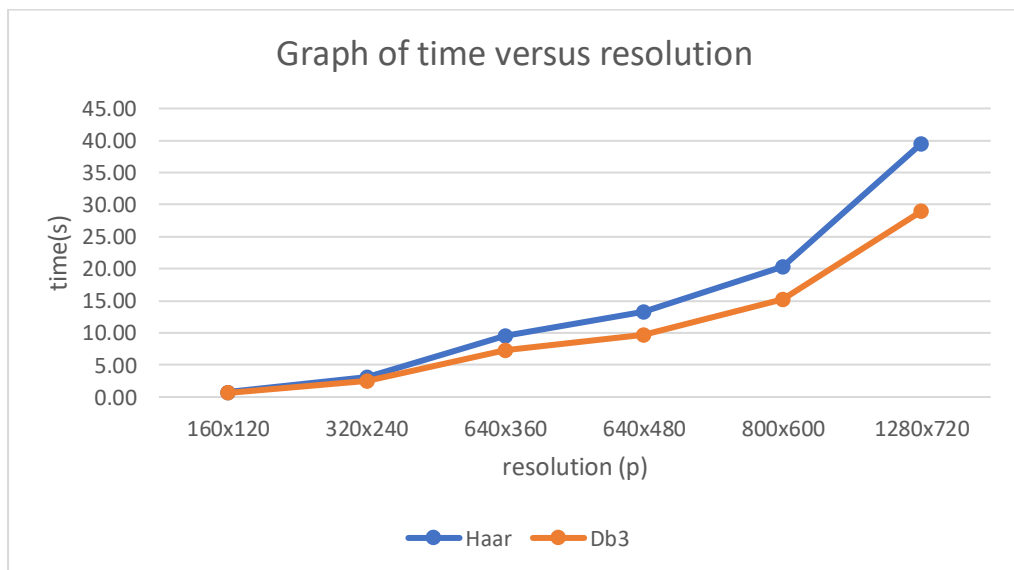
machine and the results were tabulated in Table 5.1.9.b. Overall percent error is computed based on the Table 5.1.9.a and it can be observed that parallel computing does shorten the computation time by 63.13%.

The average relative error for Haar wavelet comparing parallel computing and without parallel computing is 1.45 and average relative error for Db3 wavelet is 2.14.

5.1.10 Method 9

Table 5.1.10.a: Computation time of Haar wavelet transform and Db3 wavelet transform recorded in second(s) with different resolutions without using Parallel Computing Toolbox

Resolution	160x120	320x240	640x360	640x480	800x600	1280x720
Wavelet						
Haar	0.79	3.13	9.50	13.33	20.28	39.52
db3	0.64	2.58	7.35	9.72	15.26	28.96



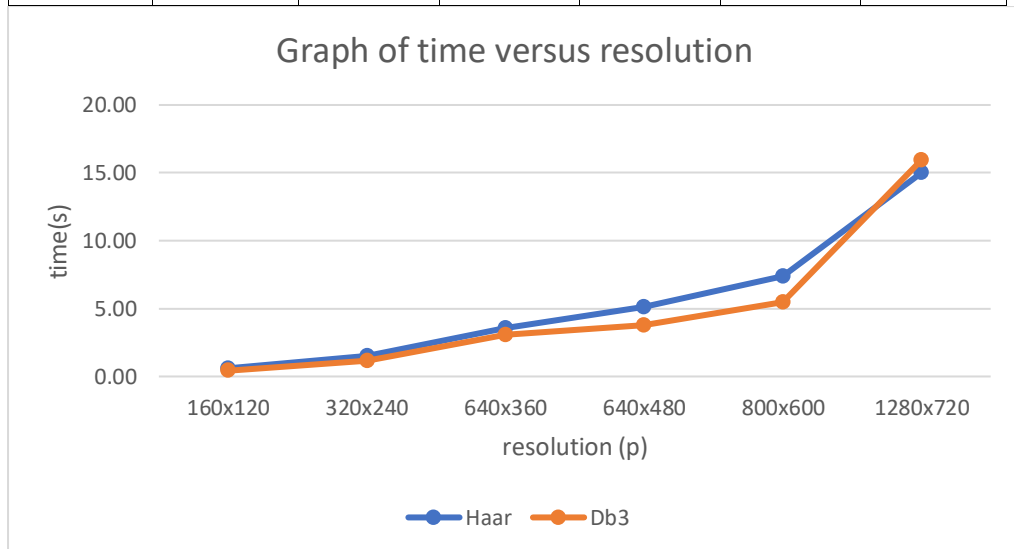
Graph 5.1.10.a: Computation time of Haar wavelet transform and Db3 wavelet transform recorded in second(s) with different resolutions without using Parallel Computing Toolbox

Table 5.1.10.a shows the computation time of wavelet transform using two-dimensional filters instead of one-dimensional wavelet filter presented in section 4.2.9. The percent error between Table 5.1.10.a and Table 5.1.9.a is 71.77% with shorter computation time for two-dimensional filters used. The result indicates that two-dimensional wavelet filter in wavelet transform uses

the Matlab built-in multithreading of matrix multiplication to speed up the wavelet transform.

Table 5.1.10.b: Computation time of Haar wavelet transform and Db3 wavelet transform recorded in second(s) with different resolutions using Parallel Computing Toolbox

Resolution \ Wavelet	160x120	320x240	640x360	640x480	800x600	1280x720
Haar	0.60	1.52	3.56	5.14	7.40	15.00
db3	0.43	1.19	3.06	3.78	5.50	15.94



Graph 5.1.10.b: Computation time of Haar wavelet transform and Db3 wavelet transform recorded in second(s) with different resolutions using Parallel Computing Toolbox

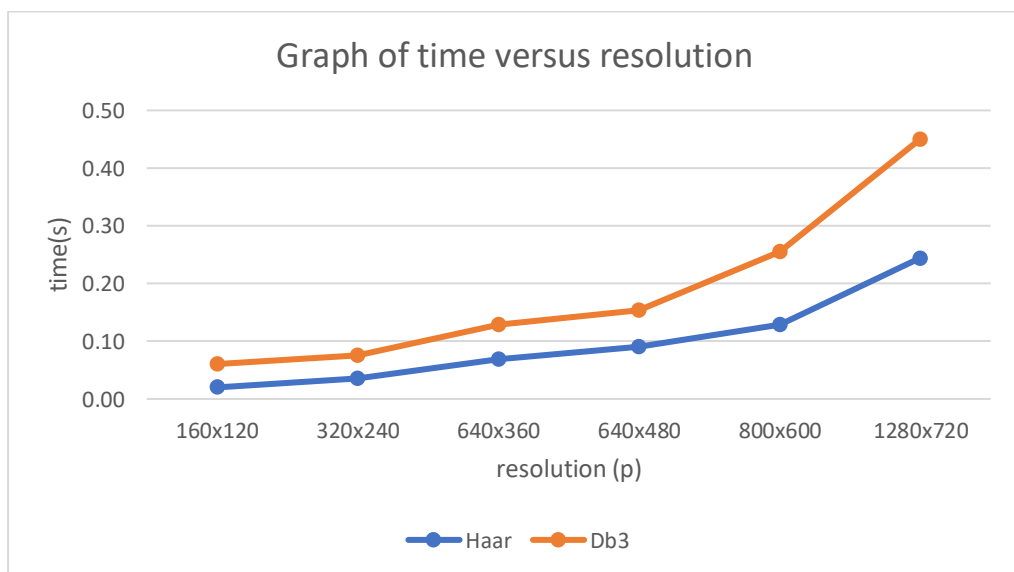
Table 5.1.10.b records the computation time of Method 9 with Parallel Computing Toolbox. The overall percent error of Table 5.1.10.b compared to Table 5.1.9.b is 68.58%. When the built-in multithreading functions were combined with Parallel Computing Toolbox, the computation time improved by 53.36%.

The average relative error for Haar wavelet comparing parallel computing and without parallel computing is 1.33 and average relative error for Db3 wavelet is 1.21.

5.1.11 Method 10

Table 5.1.11.a: Computation time of Haar wavelet transform and Db3 wavelet transform recorded in second(s) with different resolutions without using Parallel Computing Toolbox

Resolution \ Wavelet	160x120	320x240	640x360	640x480	800x600	1280x720
Haar	0.02	0.03	0.07	0.09	0.13	0.24
db3	0.06	0.08	0.13	0.15	0.26	0.45



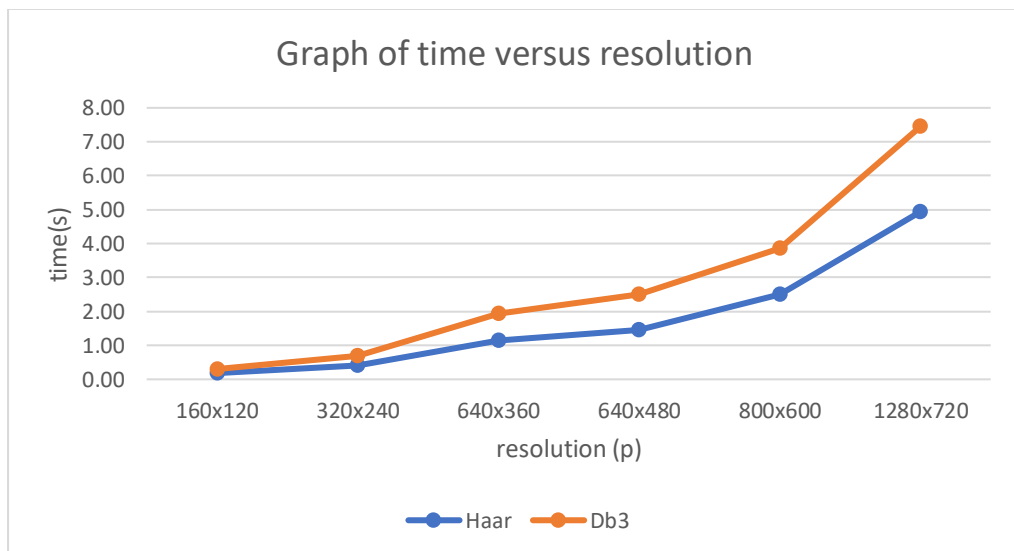
Graph 5.1.11.a: Computation time of Haar wavelet transform and Db3 wavelet transform recorded in second(s) with different resolutions without using Parallel Computing Toolbox

After evaluating and examining the results from Method 1 until Method 9, Method 10 is explored and the results were tabulated in Table 5.1.11.a. From the table, it can be seen that the process of wavelet transform is able to complete within a second for all listed resolution. The percent error of this algorithm is

compared with Table 5.1.10.a and is 97.90% faster in computation time. Using Method 10, the statistical properties of the wavelet coefficients of different type of images are available in section 5.2. Sharpness and image quality were discussed in terms of statistical values.

Table 5.1.11.b: Computation time of Haar wavelet transform and Db3 wavelet transform recorded in second(s) with different resolutions using Parallel Computing Toolbox

Resolution \ Wavelet	160x120	320x240	640x360	640x480	800x600	1280x720
Haar	0.18	0.41	1.14	1.46	2.50	4.94
db3	0.30	0.69	1.92	2.50	3.87	7.44



Graph 5.1.11.b: Computation time of Haar wavelet transform and Db3 wavelet transform recorded in second(s) with different resolutions using Parallel Computing Toolbox

Table 5.1.11.b recorded the result of parallel computing on Method 10 algorithm. Parallel Computing Toolbox is activated using the *parfor* statement.

The overall percent error shows the computation time decreased for 53.33% compared to Table 5.1.10.b. However, the computation time of Parallel Computing Toolbox of Method 10 increased by 13.19 times compared to Table 5.1.11.a. The parallel computing toolbox uses extra time to distribute the workload to all workers which is perceptible compared to the algorithm without parallel computing.

The average relative error for Haar wavelet comparing parallel computing and without parallel computing is 0.93 and average relative error for Db3 wavelet is 0.91.

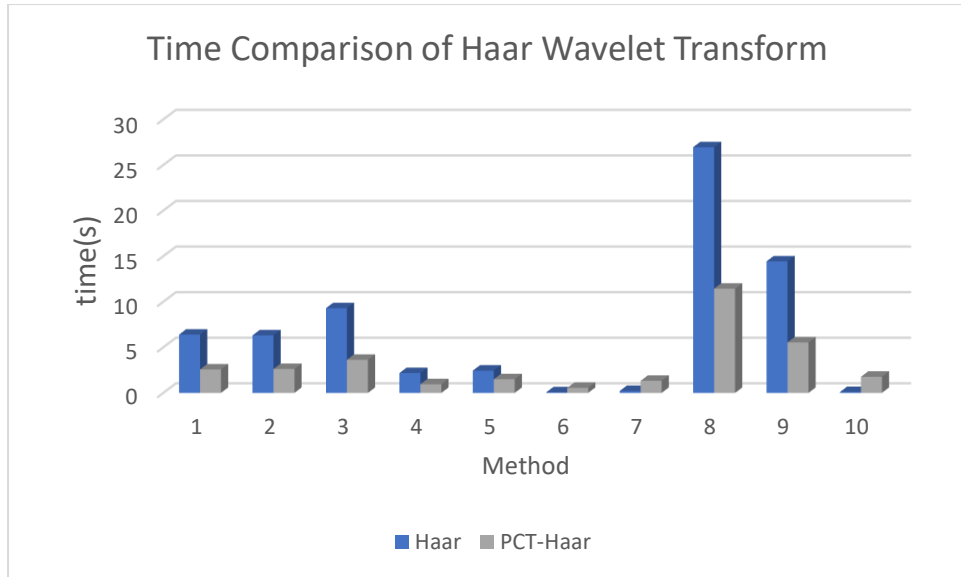
5.1.12 Comparison of Average Computation Time of All 10 Methods

Table 5.1.12.a: Computation time comparison of Haar wavelet transform recorded in second(s) with and without Parallel Computing Toolbox

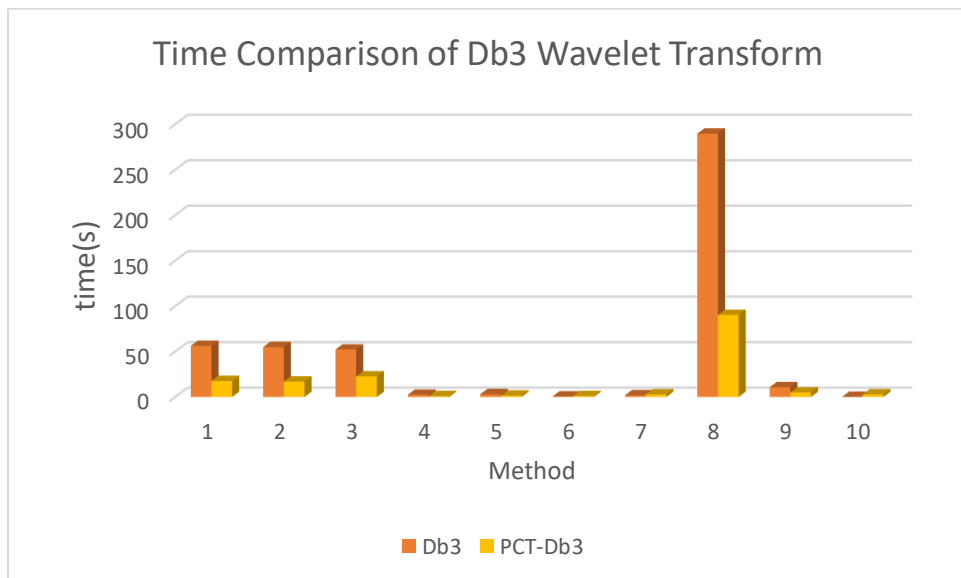
Method	Parallel Computing Toolbox		Ratio	Percentage
	Without (sec)	With (sec)	Without/With	(Without/With)x100%
1	6.39	2.57	2.49	248.64
2	6.32	2.62	2.41	241.22
3	9.29	3.63	2.56	255.92
4	2.18	0.95	2.29	229.47
5	2.44	1.5	1.63	162.67
6	0.07	0.56	0.13	12.50
7	0.21	1.33	0.16	15.79
8	26.95	11.43	2.36	235.78
9	14.43	5.54	2.60	260.47
10	0.1	1.77	0.06	5.65

Table 5.1.12.b: Computation time comparison of Db3 wavelet transform recorded in second(s) with and without Parallel Computing Toolbox

Method	Parallel Computing Toolbox		Ratio	Percentage
	Without (sec)	With (sec)	Without/With	(Without/With)x100%
1	56.19	17.63	3.19	318.72
2	54.91	16.93	3.24	324.34
3	52.19	22.60	2.31	230.93
4	2.49	0.98	2.54	254.08
5	3.03	1.51	2.01	200.66
6	0.60	1.04	0.58	57.69
7	1.86	2.78	0.67	66.91
8	289.63	90.21	3.21	321.06
9	10.75	4.98	2.16	215.86
10	0.19	2.79	0.07	6.81



Graph 5.1.12.a: Computation time comparison of Haar wavelet transform recorded in second(s) with and without Parallel Computing Toolbox



Graph 5.1.12.b: Computation time comparison of Db3 wavelet transform recorded in second(s) with and without Parallel Computing Toolbox

Based on Table 5.1.12.a and Table 5.1.12.b, Graph 5.1.12.a and Graph 5.1.12.b were plotted respectively. From Graph 5.1.12.a and Graph 5.1.12.b, it can be observed that Method 6, Method 7 and Method 10 complete the wavelet

transform with the shortest computation time. The results from Method 1,2,3,4,5,8, and 9 shows that data that requires heavy computation and repetition can be accomplished with shorter processing time using Matlab Parallel Computing Toolbox. Communication often is the main performance problem for jobs running on multicore systems. The parallel processing method used multicore of the machine to compute the given sub tasks and deliver it back to the main processor to assemble. Thus, small tasks are not recommended to use parallel computing to avoid spending excessive time in tasks distribution.

Method 6, Method 7 and Method 10 are chosen to generating different sets of wavelet coefficients to study the fundamental properties of wavelet coefficients using statistical properties. Method 6 represents the default wavelet transform whereas Method 7 and Method 10 are the enhanced algorithm in boosting the level of details in the two-dimensional wavelet coefficients.

Based on both Table 5.1.12.a and Table 5.1.12.b, the Method 6, Method 7, and Method 10 perform faster without parallel computing. The reason behind of Method 6, Method 7, and Method 10 with shorter computation time without parallel computing is because of Matlab routines that have parallel algorithms in the background as appropriate for large matrixes. Each and every pixel within the image are being measured and processed together using convolution function, whereas the use of for loop and parallel computing require to distribute the pixels to different processors and process the pixels in sequence.

However, the results from Table 5.1.12.a and Table 5.1.12.b also indicate the parallel computing toolbox are beneficial in shortening the processing time at different wavelet processing algorithms. The parallel computing toolbox trimmed down the processing time of for loops and while loops in Method 1, Method 2, Method 3, Method 4, Method 5, Method 8, and Method 9 by 200% or more.

. The two-dimensional wavelet coefficients do not include the availability of an object or the number of objects captured in the image. Wavelet coefficients are just a collection of floating points represent the change in object, depth, illumination and surface orientation. Whenever a higher vanishing moment wavelet filter is used, there will be no sharp boundary identified with a lower mean and standard deviation value from statistical properties, when image could not be segmented properly, there will be no classifications. Whenever the image length and width are reduced by half after wavelet coefficients, the image clarity will drop and with a poor result from statistical properties, classification is impossible without a proper image segmentation. Wavelet coefficients will indirectly affect the classification because the statistical properties play a crucial part in binarizing the image and segment the image into sub regions.

Refer to the previous statements, the wavelet coefficients require some analysis before image segmentation. Thus, section 5.2 will explain the statistical properties of the wavelet coefficients from Method 6, Method 7, and Method 10. This section 5.2 will compare the statistical properties of Method 6, Method 7, and Method 10 to find the best method for image segmentation.

5.2 The statistical properties of wavelet coefficients



Figure 5.2.a: Test Images

Test images presented in Figure 5.2.a is a frame randomly chosen from short video clips to investigate the statistical properties of the two-dimensional wavelet coefficients. The wavelets were chosen based on the number of vanishing moments. Vanishing moment limits the wavelets ability to represent polynomial behavior or information of the image. For example, Haar wavelet has one vanishing moment and Db6 wavelet with three vanishing moments that

encodes 3-polynomials. Haar wavelet, Db3 wavelet, and Db6 wavelet filters were used for wavelet transform of Method 6, Method 7, and Method 10 were applied on the test images listed in Figure 5.2.a. The statistic values of entropy, skewness, kurtosis, mean, and standard deviation of all images were tabulated in the tables below.

The wavelet coefficients used in investigation are horizontal-detail coefficients (HDC), vertical-detail coefficients (VDC), and diagonal-detail coefficients (DDC).

Table 5.2.a: Statistical Properties of test images Haar wavelet coefficients of Method 6

	Wavelet Coefficients	Test image 1	Test image 2	Test image 3	Test image 4	Test image 5	Test image 6
Entropy	HDC	1.37	1.35	1.34	1.43	1.28	1.37
	VDC	1.55	1.37	1.37	1.55	1.44	1.42
	DDC	1.48	1.13	1.39	1.59	1.39	1.21
Skewness	HDC	-0.50	0.32	0.60	0.52	0.27	3.21
	VDC	0.85	-2.37	-0.22	0.28	0.51	2.88
	DDC	-0.56	-0.16	-0.18	0.25	0.24	-0.46
Kurtosis	HDC	36.75	61.17	15.04	57.68	34.88	110.49
	VDC	169.73	101.85	17.41	58.39	76.14	214.96
	DDC	99.11	138.51	18.03	66.13	82.99	262.51
Mean	HDC	4.22	2.39	10.37	4.21	3.51	1.78
	VDC	1.77	1.64	10.12	2.96	2.65	1.26
	DDC	0.79	0.47	7.33	1.14	0.99	0.43
Standard Deviation	HDC	7.26	6.02	17.28	8.89	7.22	3.96
	VDC	3.93	4.89	18.07	6.04	6.71	3.75
	DDC	1.64	1.27	12.64	2.31	2.53	1.10

Table 5.2.b: Statistical Properties of test images Haar wavelet coefficients of Method 7

	Wavelet Coefficients	Test image 1	Test image 2	Test image 3	Test image 4	Test image 5	Test image 6
Entropy	HDC	1.43	1.43	1.41	1.47	1.37	1.45
	VDC	1.52	1.41	1.38	1.55	1.48	1.48
	DDC	1.50	1.20	1.39	1.59	1.43	1.30
Skewness	HDC	-0.52	0.37	0.30	0.66	0.26	3.01
	VDC	0.92	-1.86	-0.11	0.18	0.37	2.29
	DDC	0.17	0.31	-0.17	0.40	0.15	0.10
Kurtosis	HDC	36.09	61.56	15.09	58.35	34.93	107.61
	VDC	162.17	99.47	16.92	56.31	70.12	199.17
	DDC	76.38	131.71	18.03	57.56	75.80	217.70
Mean	HDC	4.24	2.47	10.30	4.21	3.59	1.83
	VDC	1.85	1.72	10.03	3.06	2.77	1.36
	DDC	0.91	0.54	7.35	1.26	1.08	0.50
Standard Deviation	HDC	7.21	6.06	17.04	8.85	7.24	3.97
	VDC	3.98	4.91	17.55	6.14	6.73	3.74
	DDC	1.79	1.37	12.58	2.43	2.59	1.18

Table 5.2.c: Statistical Properties of test images Haar wavelet coefficients of Method 10

	Wavelet Coefficients	Test image 1	Test image 2	Test image 3	Test image 4	Test image 5	Test image 6
Entropy	HDC	1.14	1.04	1.16	1.14	1.05	1.06
	VDC	1.10	0.98	1.12	1.16	1.09	1.03
	DDC	1.51	1.22	1.40	1.61	1.44	1.30
Skewness	HDC	-0.48	0.32	0.15	0.63	0.36	3.14
	VDC	0.75	-1.84	-0.11	0.12	0.50	1.81
	DDC	0.17	0.31	-0.17	0.40	0.15	0.10
Kurtosis	HDC	36.09	62.13	13.74	55.53	37.73	112.07
	VDC	136.78	99.24	14.86	53.91	75.59	195.76
	DDC	76.38	131.71	18.03	57.56	75.80	217.70
Mean	HDC	4.36	2.53	12.70	4.41	3.76	1.89
	VDC	2.06	1.80	12.47	3.31	2.95	1.44
	DDC	0.91	0.54	7.35	1.26	1.08	0.50
Standard Deviation	HDC	7.42	6.22	21.14	9.18	7.68	4.14
	VDC	4.36	5.09	21.58	6.59	7.21	3.92
	DDC	1.79	1.37	12.58	2.43	2.59	1.18

Table 5.2.d: Statistical Properties of test images Db3 wavelet coefficients of Method 6

	Wavelet Coefficients	Test image 1	Test image 2	Test image 3	Test image 4	Test image 5	Test image 6
Entropy	HDC	3.11	4.14	2.71	3.33	3.72	4.19
	VDC	4.05	4.42	2.73	3.82	3.94	4.46
	DDC	4.32	4.60	2.85	4.29	4.25	4.59
Skewness	HDC	-0.42	-0.50	0.30	-0.61	-0.06	-2.20
	VDC	-0.63	0.25	-0.34	0.01	-0.09	-0.99
	DDC	-0.21	-0.58	-0.31	-0.55	-0.33	-0.27
Kurtosis	HDC	33.89	59.54	15.23	61.62	36.11	119.95
	VDC	131.65	100.40	16.03	67.70	73.65	203.01
	DDC	68.71	167.08	18.65	80.46	87.14	275.35
Mean	HDC	3.36	1.64	9.64	2.96	2.34	1.19
	VDC	1.30	1.19	9.61	2.03	1.80	0.89
	DDC	0.72	0.46	7.17	0.92	0.79	0.43
Standard Deviation	HDC	5.83	3.85	15.14	6.21	4.61	2.40
	VDC	2.76	3.29	15.89	4.29	4.44	2.11
	DDC	1.34	0.94	12.10	1.75	1.87	0.84

Table 5.2.e: Statistical Properties of test images Db3 wavelet coefficients of Method 7

	Wavelet Coefficients	Test image 1	Test image 2	Test image 3	Test image 4	Test image 5	Test image 6
Entropy	HDC	3.12	4.16	2.75	3.35	3.72	4.20
	VDC	3.99	4.41	2.73	3.77	3.95	4.43
	DDC	4.37	4.60	2.90	4.34	4.30	4.62
Skewness	HDC	-0.48	-0.17	-0.22	-0.38	-0.05	-2.12
	VDC	-0.57	0.62	-0.09	-0.22	-0.35	-1.89
	DDC	-0.12	-0.25	-0.23	0.03	-0.22	-0.09
Kurtosis	HDC	33.62	64.32	15.65	60.29	34.99	117.35
	VDC	135.01	97.16	17.20	68.40	80.04	207.29
	DDC	85.21	153.70	18.58	98.71	100.09	311.69
Mean	HDC	3.34	1.62	9.63	2.97	2.32	1.17
	VDC	1.27	1.15	9.58	1.98	1.76	0.86
	DDC	0.67	0.44	7.17	0.87	0.76	0.41
Standard Deviation	HDC	5.77	3.83	15.23	6.18	4.60	2.43
	VDC	2.76	3.20	16.04	4.24	4.44	2.14
	DDC	1.27	0.89	12.16	1.70	1.87	0.81

Table 5.2.f: Statistical Properties of test images Db3 wavelet coefficients of Method 10

	Wavelet Coefficients	Test image 1	Test image 2	Test image 3	Test image 4	Test image 5	Test image 6
Entropy	HDC	2.99	3.77	2.55	3.18	3.40	3.87
	VDC	3.56	3.85	2.35	3.48	3.47	3.94
	DDC	4.37	4.60	2.90	4.34	4.30	4.62
Skewness	HDC	-0.47	-0.13	-0.19	-0.33	-0.05	-2.08
	VDC	-0.60	0.54	-0.16	-0.13	-0.44	-1.24
	DDC	-0.12	-0.25	-0.23	0.03	-0.22	-0.09
Kurtosis	HDC	33.71	63.23	13.21	56.22	34.63	113.41
	VDC	118.69	97.21	14.52	64.06	82.17	197.32
	DDC	85.21	153.70	18.58	98.71	100.09	311.69
Mean	HDC	3.43	1.71	12.22	3.14	2.50	1.26
	VDC	1.45	1.25	12.12	2.19	1.92	0.97
	DDC	0.67	0.44	7.17	0.87	0.76	0.41
Standard Deviation	HDC	5.90	3.93	19.35	6.41	4.93	2.54
	VDC	3.03	3.29	20.04	4.52	4.80	2.28
	DDC	1.27	0.89	12.16	1.70	1.87	0.81

Table 5.2.g: Statistical Properties of test images Db6 wavelet coefficients of Method 6

	Wavelet Coefficients	Test image 1	Test image 2	Test image 3	Test image 4	Test image 5	Test image 6
Entropy	HDC	3.19	4.21	2.77	3.41	3.86	4.30
	VDC	4.31	4.48	2.87	4.02	4.18	4.59
	DDC	4.49	4.64	3.00	4.46	4.44	4.64
Skewness	HDC	0.41	-0.32	0.65	-0.25	-0.05	0.64
	VDC	-0.25	-0.44	-0.14	0.29	0.36	1.27
	DDC	-0.59	-0.40	-0.19	-0.45	0.22	-0.92
Kurtosis	HDC	28.43	65.94	16.43	55.85	30.54	101.73
	VDC	117.09	79.57	17.28	64.92	80.85	174.71
	DDC	95.00	169.92	18.64	103.33	97.85	293.34
Mean	HDC	3.21	1.47	9.57	2.79	2.10	1.08
	VDC	1.18	1.08	9.60	1.82	1.62	0.80
	DDC	0.62	0.41	7.18	0.78	0.70	0.38
Standard Deviation	HDC	5.39	3.37	14.72	5.57	4.02	2.20
	VDC	2.52	2.80	15.66	3.89	4.05	1.92
	DDC	1.14	0.81	11.99	1.54	1.70	0.73

Table 5.2.h: Statistical Properties of test images Db6 wavelet coefficients of Method 7

	Wavelet Coefficients	Test image 1	Test image 2	Test image 3	Test image 4	Test image 5	Test image 6
Entropy	HDC	3.18	4.22	2.79	3.41	3.84	4.28
	VDC	4.18	4.45	2.86	3.91	4.12	4.52
	DDC	4.48	4.64	3.01	4.42	4.44	4.66
Skewness	HDC	0.37	0.06	0.14	-0.04	0.00	0.63
	VDC	-0.06	0.03	0.07	0.07	0.17	0.77
	DDC	-0.20	-0.16	-0.16	0.07	0.09	-0.32
Kurtosis	HDC	29.01	60.49	15.42	53.88	31.70	104.76
	VDC	110.36	84.75	17.41	63.49	73.70	171.97
	DDC	77.67	140.62	17.77	107.86	95.38	319.23
Mean	HDC	3.24	1.49	9.56	2.80	2.11	1.09
	VDC	1.20	1.10	9.56	1.85	1.65	0.81
	DDC	0.65	0.42	7.19	0.81	0.71	0.39
Standard Deviation	HDC	5.47	3.39	14.62	5.61	4.04	2.19
	VDC	2.51	2.90	15.61	3.93	4.04	1.89
	DDC	1.15	0.81	11.97	1.55	1.70	0.74

Table 5.2.i: Statistical Properties of test images Db6 wavelet coefficients of Method 10

	Wavelet Coefficients	Test image 1	Test image 2	Test image 3	Test image 4	Test image 5	Test image 6
Entropy	HDC	3.12	4.15	2.74	3.33	3.75	4.19
	VDC	3.97	4.38	2.66	3.75	3.96	4.45
	DDC	4.48	4.64	3.01	4.42	4.44	4.66
Skewness	HDC	0.37	0.05	0.13	-0.03	-0.04	0.59
	VDC	0.15	0.03	0.12	0.01	0.16	0.51
	DDC	-0.20	-0.16	-0.16	0.07	0.09	-0.32
Kurtosis	HDC	29.39	58.58	12.56	50.31	30.87	99.50
	VDC	101.57	86.19	14.15	60.64	74.63	164.76
	DDC	77.67	140.62	17.77	107.86	95.38	319.23
Mean	HDC	3.32	1.59	12.18	2.96	2.29	1.17
	VDC	1.37	1.19	12.10	2.03	1.80	0.91
	DDC	0.65	0.42	7.19	0.81	0.71	0.39
Standard Deviation	HDC	5.57	3.49	18.76	5.82	4.36	2.29
	VDC	2.77	2.97	19.62	4.16	4.36	2.02
	DDC	1.15	0.81	11.97	1.55	1.70	0.74







 <p>Standard image 1 - Watch</p>	 <p>Standard image 2 - Pool</p>
 <p>Standard image 3 - bowl_fruit</p>	 <p>Standard image 4 - peppers</p>
 <p>Standard image 5 - fruits</p>	 <p>Standard image 6 - cameraman</p>

Figure 5.2.b: Standard Images

Standard images presented in Figure 5.2.b is to investigate the statistical properties of the two-dimensional wavelet coefficients. Haar wavelet, Db3 wavelet, and Db6 wavelet filters were used for wavelet transform of Method 6, Method 7, and Method 10 were applied on the standard images listed in Figure 5.2.b. The statistic values of entropy, skewness, kurtosis, mean, and standard deviation of all images were tabulated in the tables below. The image binarization process is discussed below the Table 5.2.r.

The wavelet coefficients used in investigation are horizontal-detail

coefficients (HDC), vertical-detail coefficients (VDC), and diagonal-detail coefficients (DDC).

Table 5.2.j: Statistical Properties of standard images Haar wavelet coefficients of Method 6

	Wavelet Coefficients	Standard image 1	Standard image 2	Standard image 3	Standard image 4	Standard image 5	Standard image 6
Entropy	HDC	1.41	1.56	1.27	1.01	0.93	1.12
	VDC	1.41	1.56	1.37	1.13	0.97	1.38
	DDC	1.43	1.57	1.59	1.44	1.19	1.58
Skewness	HDC	3.88	11.95	7.69	5.91	6.80	4.26
	VDC	5.73	12.30	6.67	6.08	6.85	4.49
	DDC	6.09	9.17	6.45	6.42	4.72	4.39
Kurtosis	HDC	23.83	189.09	104.53	53.93	72.31	28.38
	VDC	53.63	214.06	83.86	57.67	72.22	28.35
	DDC	64.34	111.43	102.89	70.93	37.17	32.35
Mean	HDC	4.09	1.74	2.19	4.04	5.03	7.31
	VDC	2.96	1.70	1.76	3.08	4.94	8.23
	DDC	1.50	0.92	0.92	1.57	2.58	3.77
Standard Deviation	HDC	9.25	7.35	3.58	7.48	9.54	14.18
	VDC	7.77	6.75	2.66	5.51	9.61	19.08
	DDC	3.92	2.67	1.19	2.50	3.72	8.14

Table 5.2.k: Statistical Properties of standard images Haar wavelet coefficients of Method 7

	Wavelet Coefficients	Standard image 1	Standard image 2	Standard image 3	Standard image 4	Standard image 5	Standard image 6
Entropy	HDC	1.40	1.56	1.27	1.02	0.94	1.09
	VDC	1.41	1.56	1.34	1.17	0.97	1.34
	DDC	1.43	1.58	1.59	1.49	1.19	1.56
Skewness	HDC	3.89	11.87	7.75	5.94	6.77	4.29
	VDC	5.69	12.08	6.81	6.27	6.95	4.46
	DDC	6.05	9.44	5.76	6.19	4.70	4.36
Kurtosis	HDC	23.79	187.89	106.25	54.93	71.46	29.13
	VDC	52.18	207.43	87.32	62.32	74.44	27.31
	DDC	63.00	123.16	88.84	65.36	37.06	31.47
Mean	HDC	4.08	1.73	2.18	4.02	5.01	7.26
	VDC	2.96	1.69	1.82	3.01	4.92	8.38
	DDC	1.50	0.93	0.92	1.49	2.57	3.87
Standard Deviation	HDC	9.25	7.18	3.57	7.42	9.47	13.96
	VDC	7.78	6.63	2.71	5.53	9.64	19.56
	DDC	3.92	2.67	1.14	2.42	3.70	8.17

Table 5.2.l: Statistical Properties of standard images Haar wavelet coefficients of Method 10

	Wavelet Coefficients	Standard image 1	Standard image 2	Standard image 3	Standard image 4	Standard image 5	Standard image 6
Entropy	HDC	1.00	1.00	0.79	0.62	0.62	0.72
	VDC	0.97	1.00	0.82	0.69	0.63	0.88
	DDC	1.44	1.60	1.63	1.53	1.22	1.58
Skewness	HDC	3.97	11.64	7.46	5.93	6.45	4.13
	VDC	5.50	11.85	6.60	6.42	6.64	4.30
	DDC	6.05	9.44	5.76	6.19	4.70	4.36
Kurtosis	HDC	24.44	177.46	101.46	54.89	65.15	26.28
	VDC	47.35	191.91	86.84	65.99	68.76	25.53
	DDC	63.00	123.16	88.84	65.36	37.06	31.47
Mean	HDC	4.33	1.93	2.36	4.29	5.55	8.32
	VDC	3.24	1.88	2.03	3.33	5.49	9.38
	DDC	1.50	0.93	0.92	1.49	2.57	3.87
Standard Deviation	HDC	10.06	7.67	3.75	7.81	10.20	16.12
	VDC	8.73	7.16	2.95	6.05	10.34	21.18
	DDC	3.92	2.67	1.14	2.42	3.70	8.17

Table 5.2.m: Statistical Properties of standard images Db3 wavelet coefficients of Method 6

	Wavelet Coefficients	Standard image 1	Standard image 2	Standard image 3	Standard image 4	Standard image 5	Standard image 6
Entropy	HDC	5.28	6.01	4.86	3.86	2.90	4.10
	VDC	5.59	5.74	5.08	4.61	3.02	4.64
	DDC	5.76	6.01	6.39	5.80	4.01	5.49
Skewness	HDC	4.56	11.48	6.28	6.11	5.92	3.90
	VDC	7.36	14.40	4.95	6.51	6.12	4.52
	DDC	7.40	8.83	3.79	6.90	4.58	4.91
Kurtosis	HDC	34.85	176.30	85.69	65.46	58.25	25.78
	VDC	84.23	308.19	67.55	71.28	60.84	31.92
	DDC	89.17	112.96	49.50	85.06	39.92	40.37
Mean	HDC	3.00	1.27	1.45	2.74	4.14	6.08
	VDC	1.85	1.27	1.24	1.99	4.01	7.71
	DDC	1.19	0.86	0.77	1.22	2.11	3.22
Standard Deviation	HDC	6.74	4.87	1.91	4.64	6.79	10.95
	VDC	5.54	4.50	1.40	3.36	6.89	16.90
	DDC	3.18	2.04	0.82	1.88	2.68	6.87

Table 5.2.n: Statistical Properties of standard images Db3 wavelet coefficients of Method 7

	Wavelet Coefficients	Standard image 1	Standard image 2	Standard image 3	Standard image 4	Standard image 5	Standard image 6
Entropy	HDC	5.28	6.03	4.89	3.87	2.89	4.23
	VDC	5.60	5.75	5.21	4.54	3.02	4.73
	DDC	5.75	6.05	6.29	5.74	4.01	5.59
Skewness	HDC	4.55	12.46	6.41	5.94	5.95	4.09
	VDC	7.92	14.99	4.91	6.54	6.09	4.52
	DDC	7.47	8.99	4.59	6.52	4.49	4.86
Kurtosis	HDC	34.55	213.34	87.19	59.42	58.62	27.55
	VDC	100.71	327.21	67.89	74.92	59.88	32.54
	DDC	91.17	121.63	68.65	76.63	38.55	39.80
Mean	HDC	2.99	1.29	1.45	2.73	4.17	6.18
	VDC	1.86	1.28	1.20	2.02	4.03	7.66
	DDC	1.19	0.85	0.82	1.22	2.12	3.20
Standard Deviation	HDC	6.70	5.13	1.95	4.65	6.88	11.42
	VDC	5.66	4.60	1.35	3.39	6.94	16.72
	DDC	3.19	2.02	0.91	1.84	2.68	6.89

Table 5.2.o: Statistical Properties of standard images Db3 wavelet coefficients of Method 10

	Wavelet Coefficients	Standard image 1	Standard image 2	Standard image 3	Standard image 4	Standard image 5	Standard image 6
Entropy	HDC	3.74	4.44	4.16	3.37	2.57	3.65
	VDC	3.62	4.22	4.36	3.76	2.64	3.90
	DDC	5.75	6.05	6.29	5.74	4.01	5.59
Skewness	HDC	4.54	11.58	6.00	5.76	5.47	3.94
	VDC	7.08	13.31	4.95	6.48	5.57	4.20
	DDC	7.47	8.99	4.59	6.52	4.49	4.86
Kurtosis	HDC	33.94	185.74	79.73	56.00	50.41	25.30
	VDC	78.96	258.99	70.04	74.14	51.00	27.95
	DDC	91.17	121.63	68.65	76.63	38.55	39.80
Mean	HDC	3.29	1.57	1.66	3.02	4.67	7.16
	VDC	2.21	1.53	1.44	2.37	4.57	8.58
	DDC	1.19	0.85	0.82	1.22	2.12	3.20
Standard Deviation	HDC	7.40	5.50	2.15	4.98	7.37	13.23
	VDC	6.48	5.03	1.63	3.85	7.43	18.05
	DDC	3.19	2.02	0.91	1.84	2.68	6.89

Table 5.2.p: Statistical Properties of standard images Db6 wavelet coefficients of Method 6

	Wavelet Coefficients	Standard image 1	Standard image 2	Standard image 3	Standard image 4	Standard image 5	Standard image 6
Entropy	HDC	5.48	6.18	4.95	3.90	2.85	4.41
	VDC	6.02	5.90	5.27	4.62	2.99	4.86
	DDC	6.12	6.20	6.50	5.86	4.04	5.66
Skewness	HDC	4.51	12.17	5.41	5.34	5.21	4.06
	VDC	8.01	14.54	3.95	6.33	5.39	4.39
	DDC	7.36	8.06	3.66	6.48	4.08	4.85
Kurtosis	HDC	34.57	208.50	65.96	46.94	46.03	27.59
	VDC	101.63	326.33	48.67	76.53	46.78	30.76
	DDC	89.01	98.28	47.02	77.12	33.41	38.31
Mean	HDC	2.83	1.27	1.39	2.59	4.09	6.20
	VDC	1.76	1.25	1.16	1.92	3.94	7.42
	DDC	1.18	0.83	0.76	1.19	2.05	3.09
Standard Deviation	HDC	6.04	4.88	1.72	4.09	6.23	11.31
	VDC	5.41	4.21	1.22	3.04	6.31	15.88
	DDC	3.03	1.86	0.82	1.76	2.45	6.69

Table 5.2.q: Statistical Properties of standard images Db6 wavelet coefficients of Method 7

	Wavelet Coefficients	Standard image 1	Standard image 2	Standard image 3	Standard image 4	Standard image 5	Standard image 6
Entropy	HDC	5.48	6.19	5.02	3.90	2.85	4.37
	VDC	6.03	5.90	5.33	4.63	2.99	4.81
	DDC	6.12	6.19	6.36	5.85	4.03	5.58
Skewness	HDC	4.56	11.58	5.40	5.61	5.25	3.84
	VDC	7.69	14.36	3.90	6.40	5.40	4.38
	DDC	7.41	8.11	4.05	6.21	4.03	4.79
Kurtosis	HDC	35.59	186.15	67.07	56.04	46.71	25.50
	VDC	92.31	315.35	49.95	79.40	47.45	30.77
	DDC	90.20	100.22	53.61	70.90	32.29	38.67
Mean	HDC	2.83	1.25	1.35	2.60	4.09	6.14
	VDC	1.75	1.24	1.13	1.92	3.93	7.47
	DDC	1.17	0.84	0.80	1.18	2.05	3.16
Standard Deviation	HDC	6.06	4.71	1.65	4.14	6.25	10.93
	VDC	5.30	4.13	1.17	3.06	6.29	15.85
	DDC	3.03	1.88	0.87	1.71	2.44	6.65

Table 5.2.r: Statistical Properties of standard images Db6 wavelet coefficients of Method 10

	Wavelet Coefficients	Standard image 1	Standard image 2	Standard image 3	Standard image 4	Standard image 5	Standard image 6
Entropy	HDC	4.90	5.19	4.52	3.56	2.58	4.25
	VDC	4.88	4.91	4.73	4.08	2.66	4.66
	DDC	6.12	6.19	6.36	5.85	4.03	5.58
Skewness	HDC	4.49	10.65	4.98	5.41	4.78	3.73
	VDC	6.83	12.52	4.02	6.27	4.94	4.00
	DDC	7.41	8.11	4.05	6.21	4.03	4.79
Kurtosis	HDC	34.05	159.69	59.01	52.25	39.47	23.56
	VDC	72.71	241.26	50.33	73.93	40.59	25.65
	DDC	90.20	100.22	53.61	70.90	32.29	38.67
Mean	HDC	3.17	1.55	1.57	2.88	4.57	7.10
	VDC	2.14	1.49	1.38	2.25	4.44	8.48
	DDC	1.17	0.84	0.80	1.18	2.05	3.16
Standard Deviation	HDC	6.74	5.06	1.87	4.46	6.69	12.69
	VDC	6.06	4.54	1.47	3.51	6.71	17.16
	DDC	3.03	1.88	0.87	1.71	2.44	6.65

The Table 5.2.a to Table 5.2.r shows the statistical properties of the two-dimensional wavelet coefficients, however, the statistic values were not able to use as a feature to classify the images. In general, statistic values elucidate the texture of the foreground and background of the images instead of specific object. As discussed in section 4.3.2, mean and standard deviation values are used to distinct the smooth regions and edges from the two-dimensional wavelet coefficients. Thus, colour threshold analysis is proposed to use the mean and standard deviations obtained from the wavelet coefficients to define the threshold values of the wavelet coefficients. The threshold value binarized the wavelet coefficients to identify the location of the edges to segment an image into smaller regions. A binarized image will define the true area and the false area of every sub region as they lower than mean value or higher than mean value. Kurtosis, skewness, and entropy are statistical properties best to describe the distribution of data because they detect the lack of symmetry when values are not uniformly distributed around the mean. Thus, it can be meaningful measures to describe the texture in a two-dimensional signal as mentioned in section 4.5. Section 5.3 discussed the usage of binarized wavelet coefficients serve as an edge feature for colour threshold analysis instead of image segmentation as discussed in section 4.4.

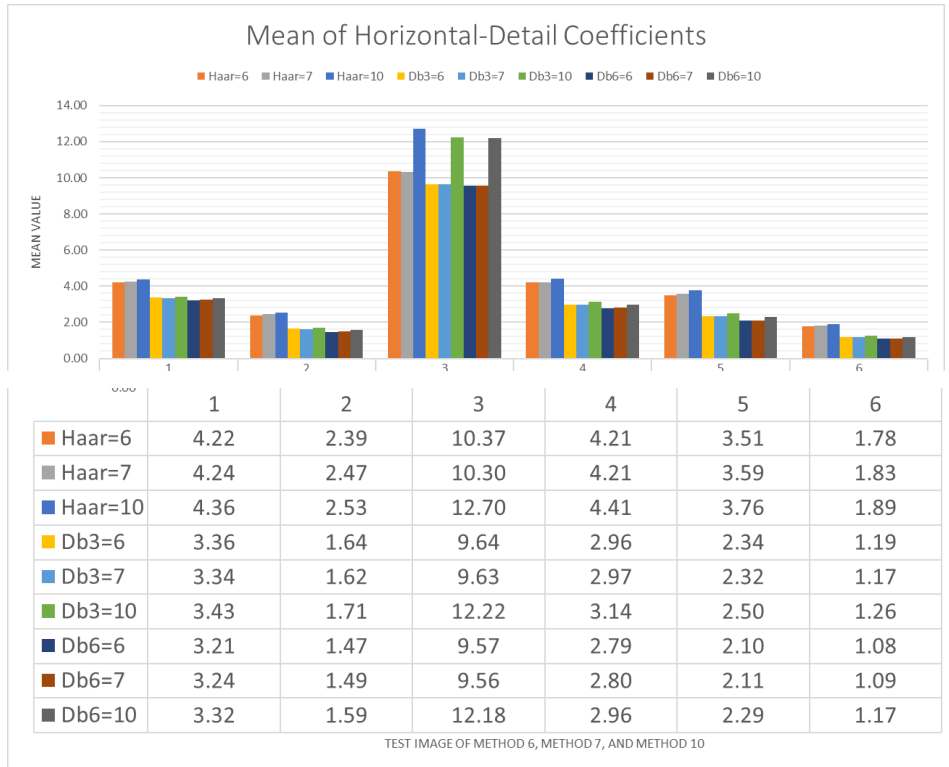
In the following section 5.2.1, statistical properties such as standard deviation and mean of the test images and standard images serve as the threshold value to distinguish the edges out of the two-dimensional data within the test images and standard images. Mean values measure the generality of the data and indicate which data is spread furthest away from the mean values. Whereas

standard deviation is a useful measure of the scatter of observation in the data. It measures the differences of each observation from the mean.

Entropy is not discussed in section 5.2.1 because it represents the disorder or uncertainty of the data. Entropy usually used in measuring the unpredictability of the state, of the data average information content. As discussed in section 3.3, skewness data is used to describe the relations between mean, mode, and median of the two-dimensional data. When the skewness skewed to the right, data that are skewed to the right have a long tail that extends to the right with a positive value. In this situation, the texture of the two-dimensional data is presented with less edges, fine and smooth texture. The texture of the test images and standard images is described as coarse or grained with a long tailed skewed to the left. Along with skewness, kurtosis is an important descriptive statistic of data distribution and the distribution tails of skewness.

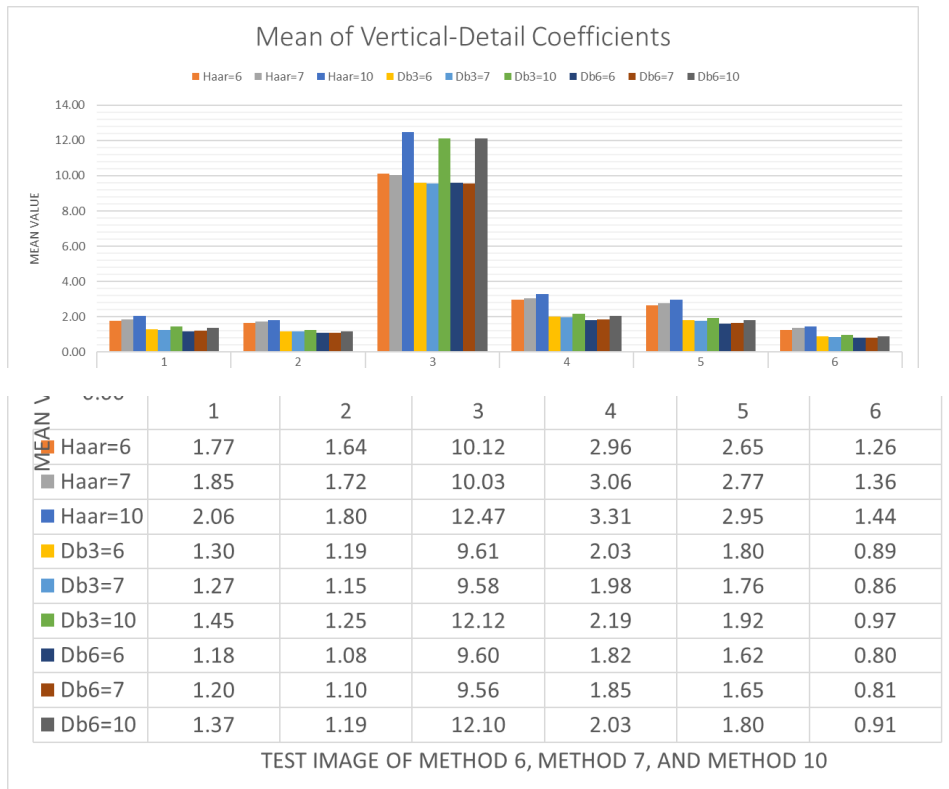
5.2.1 Sharpness Comparison Between Traditional Wavelet Transform and Proposed Method of The Research

One of the most challenging problems for researchers is the image quality assessment. The dominating factor in image quality assessment is image sharpness or blurriness. The sharpness of the wavelet coefficients is evaluated based on the standard deviation and mean of the two-dimensional wavelet coefficients. The standard deviation is a summary measure of the differences of each observation from the mean. The mean is often called the average and its value changes based on the type of images. The traditional wavelet transform is represented by the algorithm from Method 6 and Method 7, whereas the proposed method is Method 10.



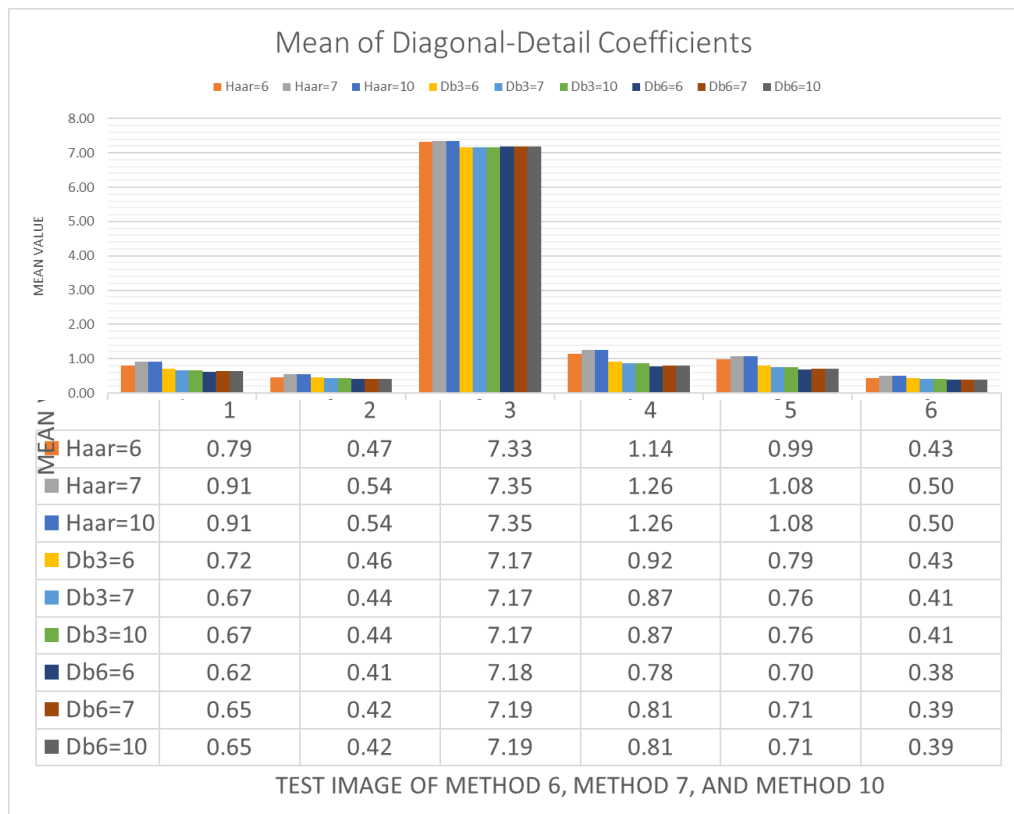
Graph 5.2.1.a: Bar Chart of Test Images Horizontal-Detail Wavelet Coefficients'

mean



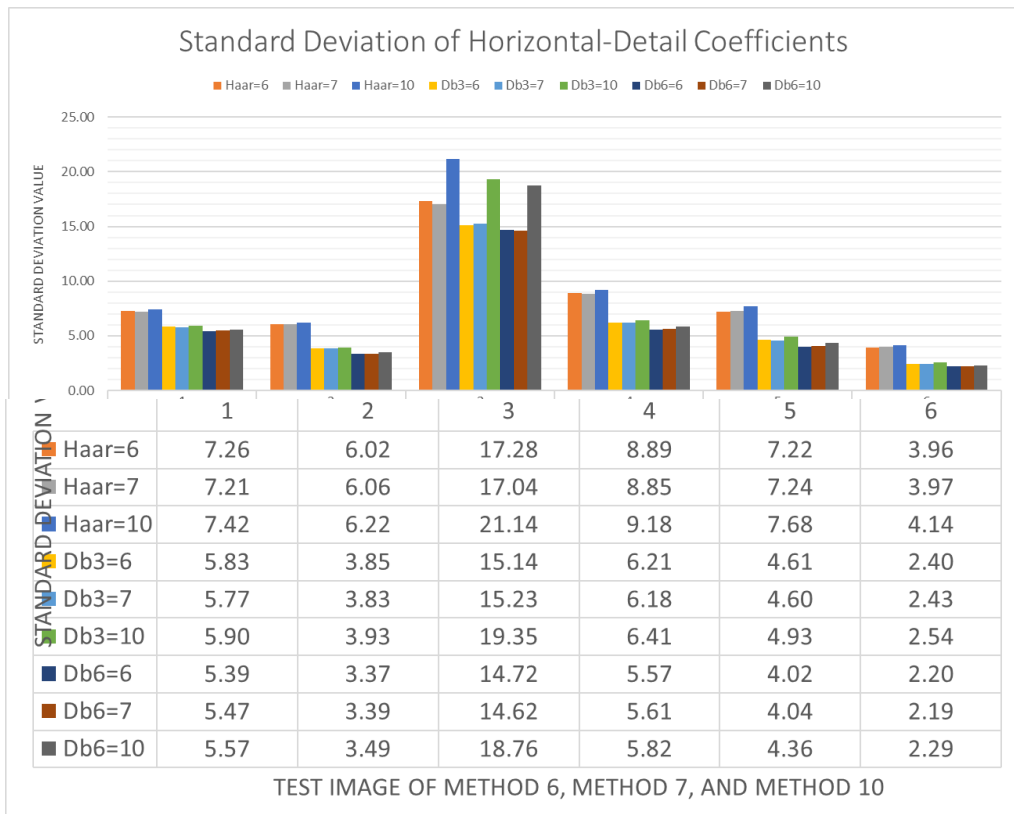
Graph 5.2.1.b: Bar Chart of Test Images Vertical-Detail Wavelet Coefficients'

mean



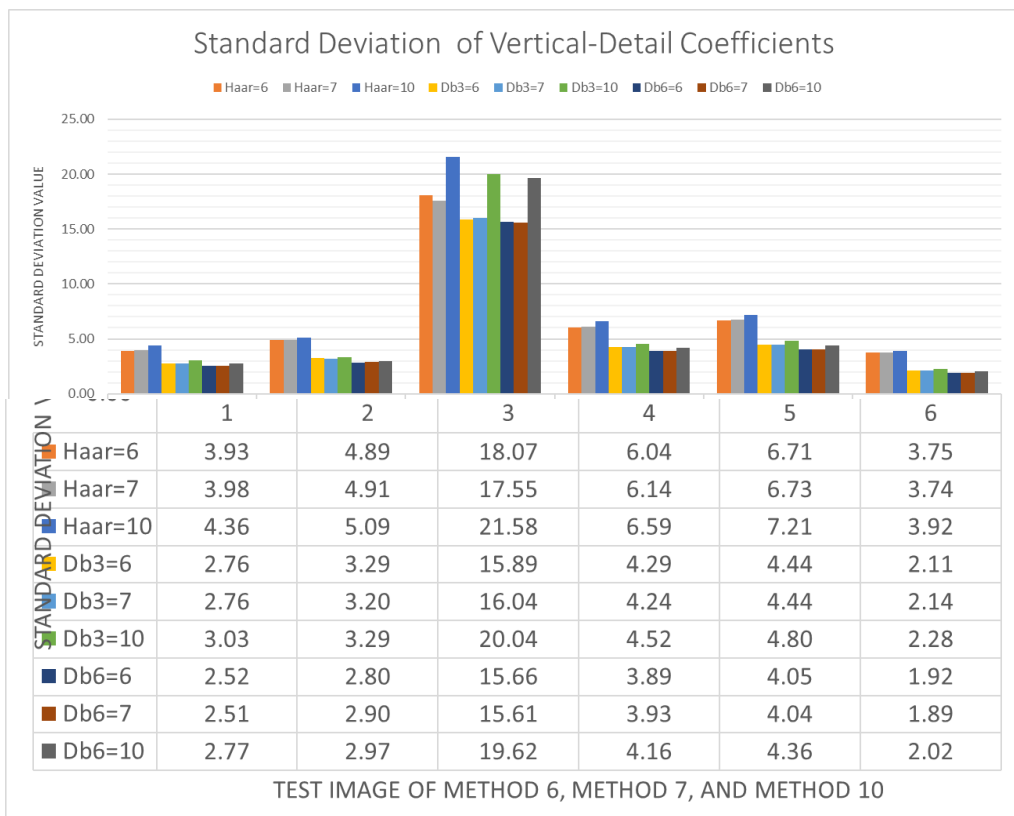
Graph 5.2.1.c: Bar Chart of Test Images Diagonal-Detail Wavelet Coefficients' mean

Based on Graph 5.2.1.a to Graph 5.2.1.c, Method 10 generates the highest mean of wavelet coefficients from all test images. Besides, Haar wavelet filter offers the highest mean among the wavelet filters in all aspects. Test image 3 creates wavelet coefficients with the highest mean value, the result proves that test image 3 have the most edges among the test images. On the other hand, wavelet coefficients from test image 6 has the lowest mean in all aspects. This suggests that test image 6 is the least complex among the test images.



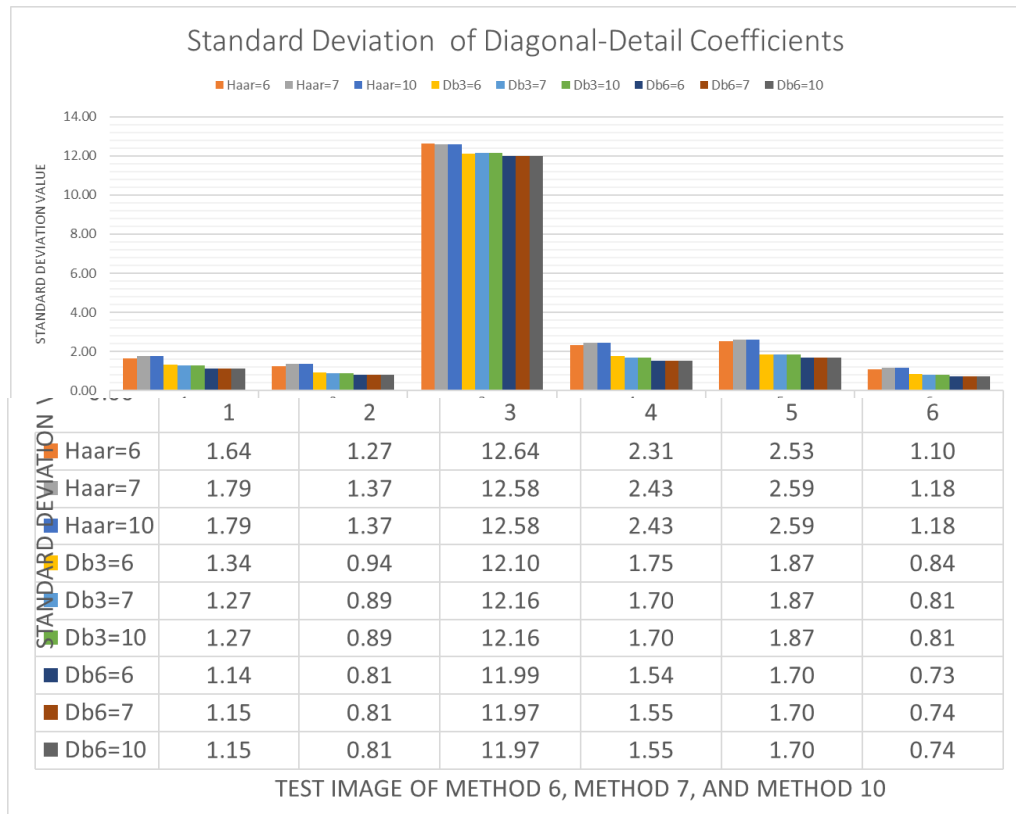
Graph 5.2.1.d: Bar Chart of Test Images Horizontal-Detail Wavelet Coefficients'

standard deviation



Graph 5.2.1.e: Bar Chart of Test Images Vertical-Detail Wavelet Coefficients'

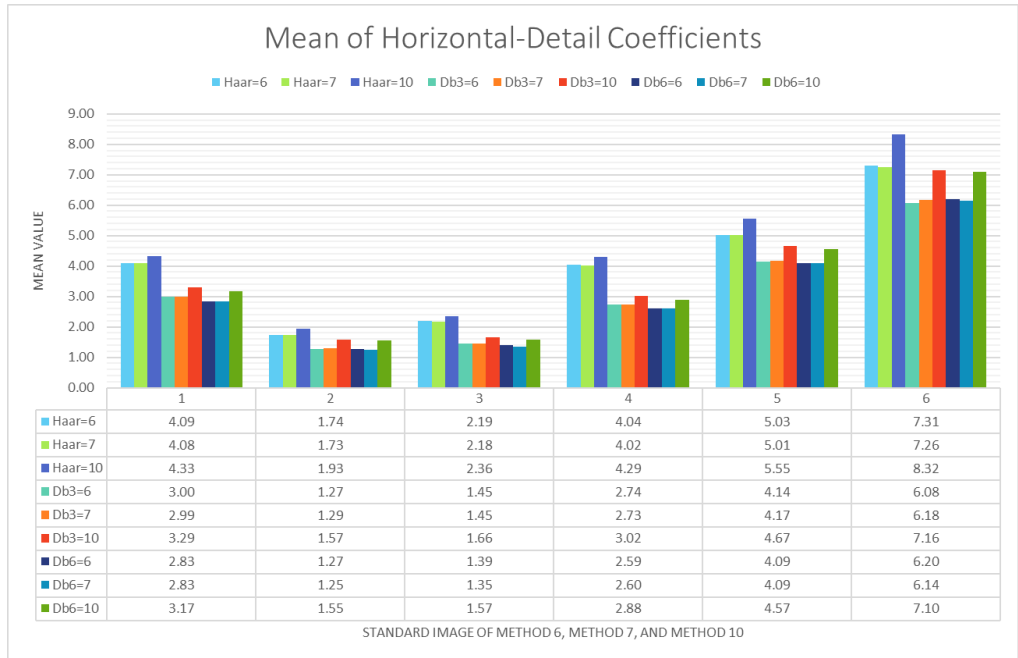
standard deviation



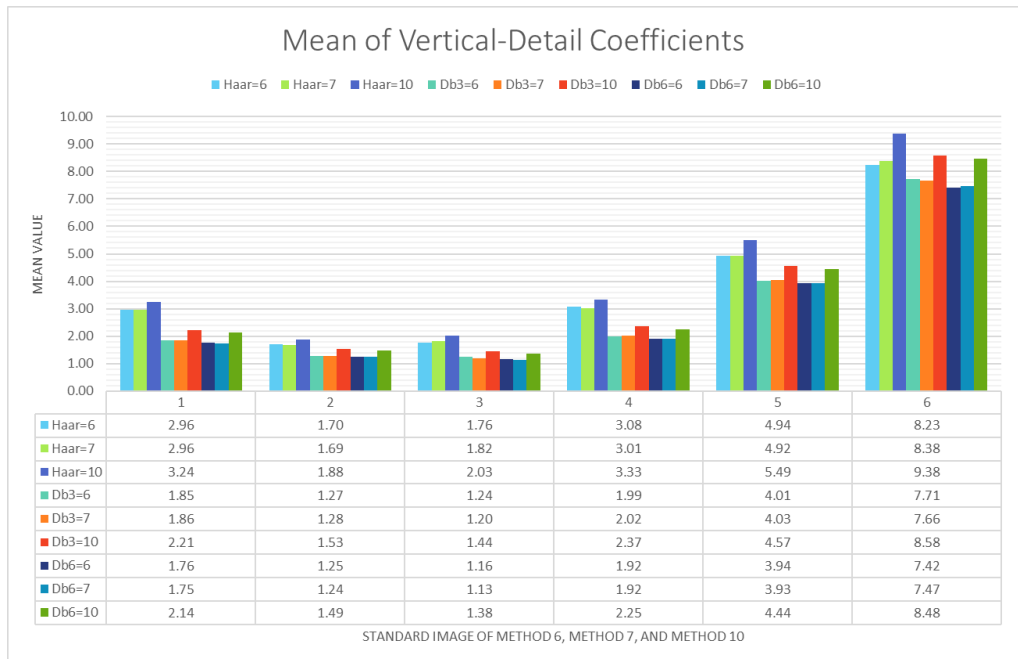
Graph 5.2.1.f: Bar Chart of Test Images Diagonal-Detail Wavelet Coefficients'

standard deviation

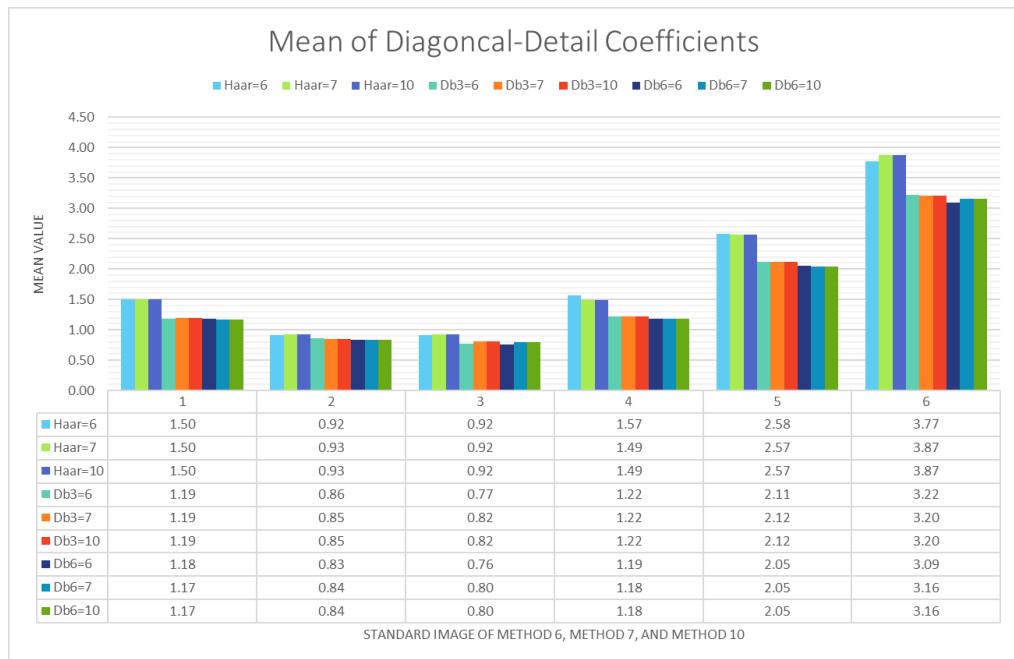
Based on Graph 5.2.1.d until Graph 5.2.1.f, Method 10 generates the highest standard deviation of wavelet coefficients from all test images. Besides, Haar wavelet filter scores the highest standard deviation among the wavelet filters in all aspects. Test image 3 creates wavelet coefficients with the highest standard deviation value, the result indicates that test image 3 edges are the sharpest among the test images. On the other hand, wavelet coefficients from test image 6 has the lowest standard deviation in all aspects. This shows that test image 6 is the smoothest among the test images.



Graph 5.2.1.g: Bar Chart of Standard Images Horizontal-Detail Wavelet Coefficients' mean

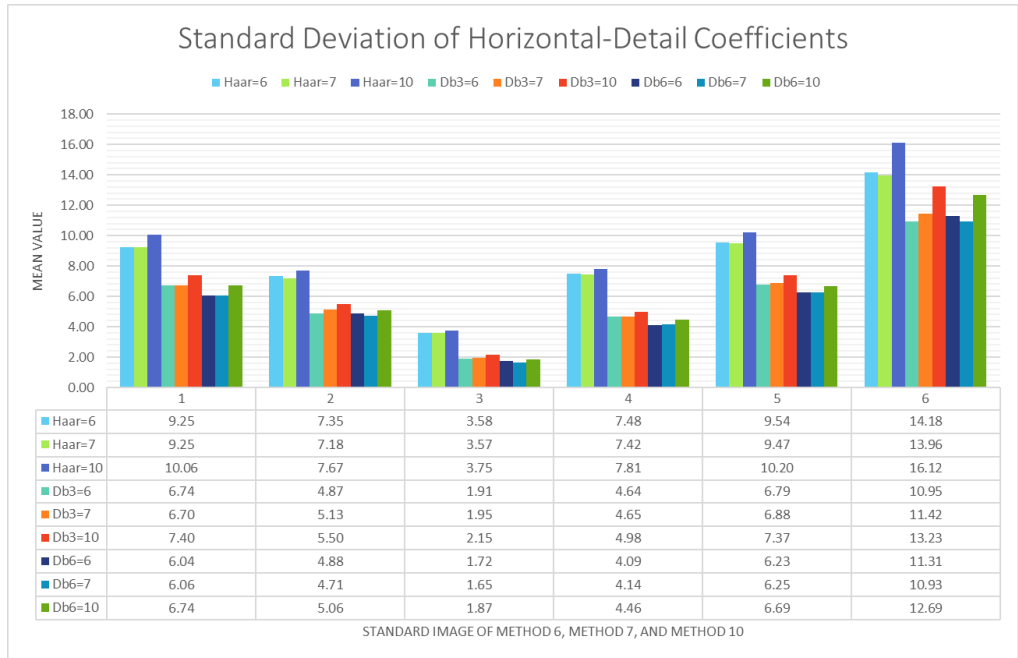


Graph 5.2.1.h: Bar Chart of Standard Images Vertical-Detail Wavelet Coefficients' mean

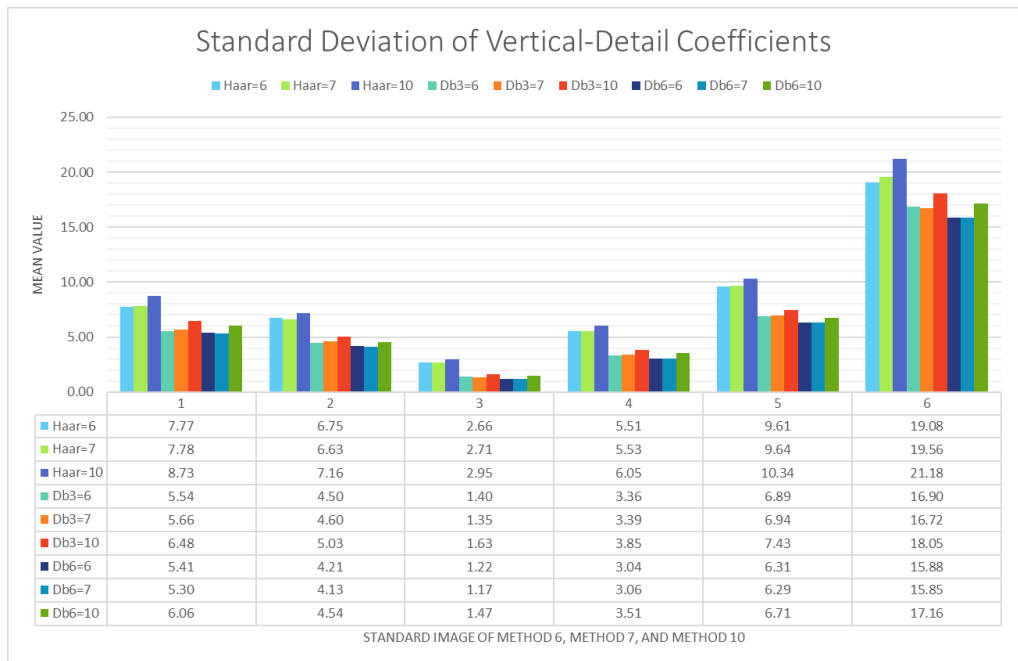


Graph 5.2.1.i: Bar Chart of Standard Images Diagonal-Detail Wavelet Coefficients' mean

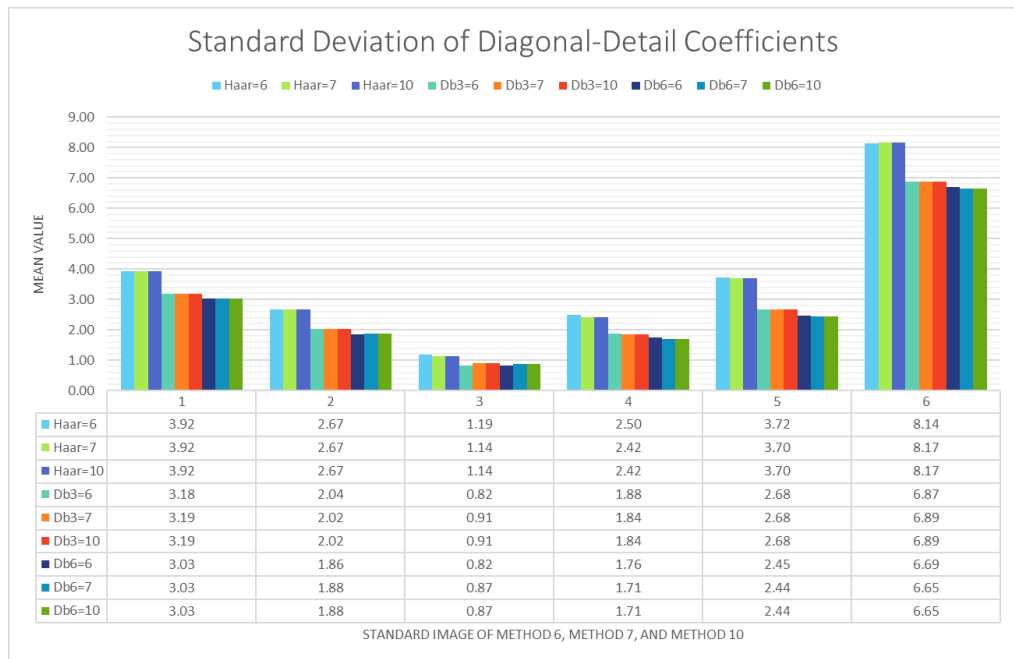
Based on Graph 5.2.1.g to Graph 5.2.1.i, Method 10 generates the highest mean of horizontal-detail and vertical-detail wavelet coefficients from all standard images. Besides, Haar wavelet filter offers the highest mean among the wavelet filters in all aspects. Standard image 6 creates wavelet coefficients with the highest mean value, the result proves that standard image 6 have the most edges among the standard images. On the other hand, wavelet coefficients from standard image 3 has the lowest mean in all aspects. This suggests that standard image 3 is the least complex among the standard images.



Graph 5.2.1.j: Bar Chart of Standard Images Horizontal-Detail Wavelet Coefficients' standard deviation



Graph 5.2.1.k: Bar Chart of Standard Images Vertical-Detail Wavelet Coefficients' standard deviation



Graph 5.2.1.l: Bar Chart of Standard Images Diagonal-Detail Wavelet Coefficients' standard deviation

Based on Graph 5.2.1.j until Graph 5.2.1.l, Method 10 generates the highest standard deviation of horizontal-detail and vertical-detail wavelet coefficients from all standard images. Besides, Haar wavelet filter scores the highest standard deviation among the wavelet filters in all aspects. Standard image 6 creates wavelet coefficients with the highest standard deviation value, the result indicates that standard image 6 edges are the sharpest among the standard images. On the other hand, wavelet coefficients from standard image 3 has the lowest standard deviation in all aspects. This shows that standard image 3 is the smoothest among the standard images.

The quality of input images together with the method used in wavelet transform will reduce the accuracy of the classification process. In this section, the features and statistical properties from the coefficients illustrate the

characteristic of features accessible from the wavelet coefficients, which delivers a reliable information to the training process. The superlative statistical properties of the wavelet coefficients from method 10 proven this method can retains more information from the input image and harvest a sharper and high-quality wavelet coefficient, provide more reliable information for image segmentation process.

5.3 Feature Extraction

In this section after the binarization of wavelet coefficients using mean value as a threshold, only the strong coefficients will be distinguished as edges in the image. The edges correspond to the pixels' value above the mean whereas the punier values are ignored.

The mean value from the wavelet coefficients is the expected threshold value of a discrete set of numbers. This threshold value from the wavelet coefficients are instinctive in identifying and differentiating the edges from the image. The biggest drawback of this edge feature after binarization is the undetected edges came from the soft edge region. The undetected soft edges below the mean value creates an opening along the edge line as shown in Figure 5.3.a. This results in the clustering process. Regions that are supposed to be separated will merge together.

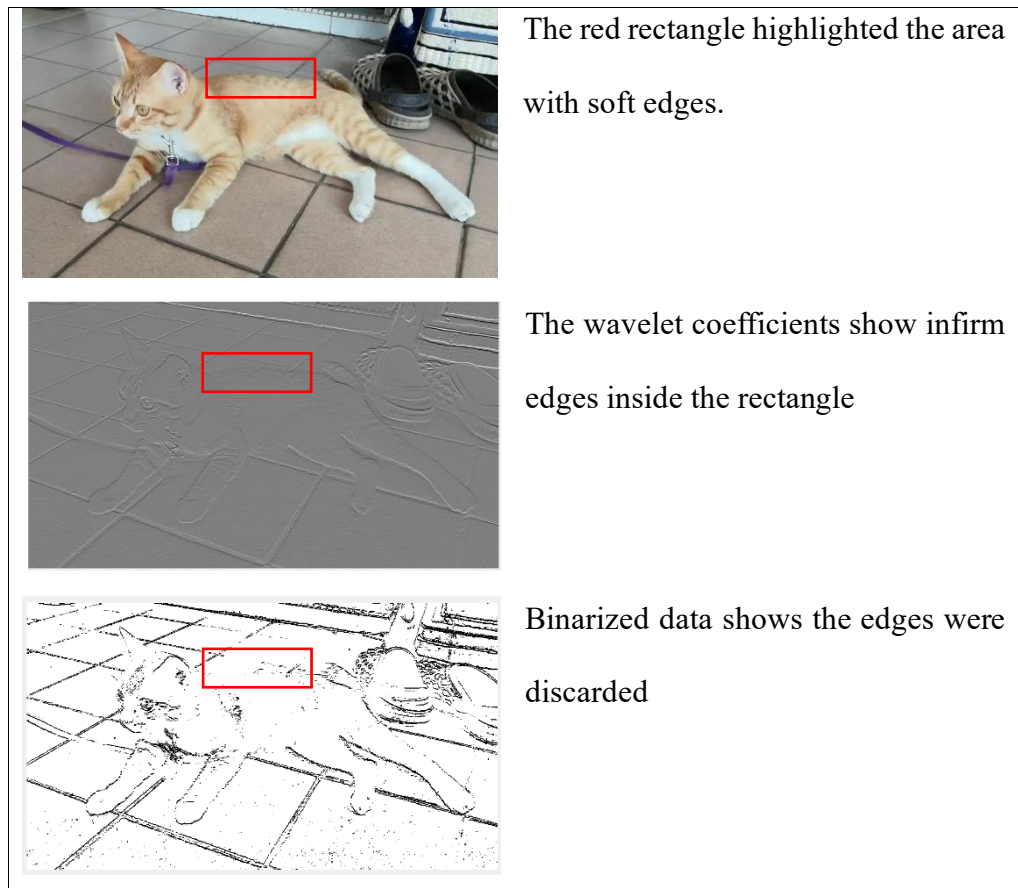


Figure 5.3.a: Edge feature from wavelet coefficients of Test image 4

In this research work, it is envisaged that colour threshold analysis can assist in resolving the problem as discussed in section 4.3. Edges are defined by colour intensities instead of the binarized layers from wavelet coefficients. Local bar charts were created and graphs were plotted to determine the peak and valley of colour intensity of a sub-area divided from the original image. Peaks and valleys from a graph denote the local colour thresholds. Local bar charts were created from the sub-areas after the image is partitioned into smaller image size as shown in Figure 5.3.b. In the example, image is divided into 45 sub-areas of similar size in all 4 arrangements.

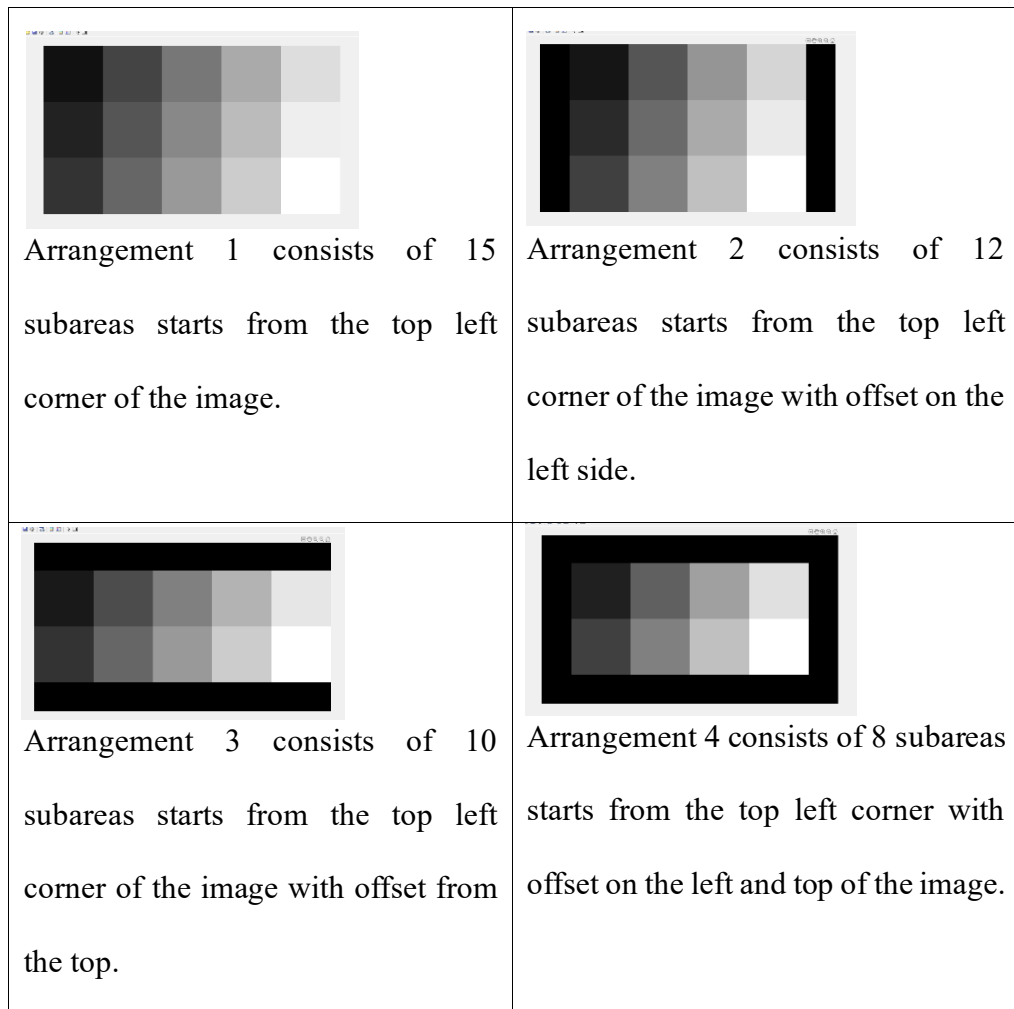


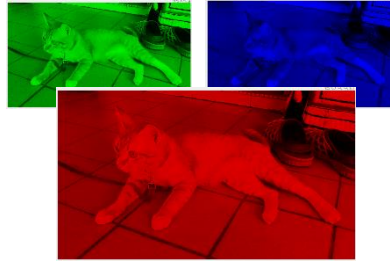
Figure 5.3.b: The 4 arrangements of sub-areas

As mentioned in section 4.3.1, local threshold value of the sub area is calculated using a local bar chart. The divisor of sub-areas is a controllable variable in this research work. Each sub-area carries its local colour threshold values change accordingly when the sub-areas size transformed as shown in the Figure 5.3.c below.

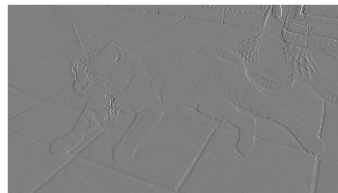
Figure 5.3.c illustrates the process of creating a local bar chart from a sub-area of horizontal-detail coefficients after wavelet transform using Haar wavelet from the red layer of test image 4.



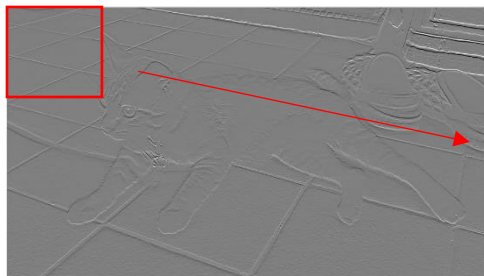
Original size of test image 4



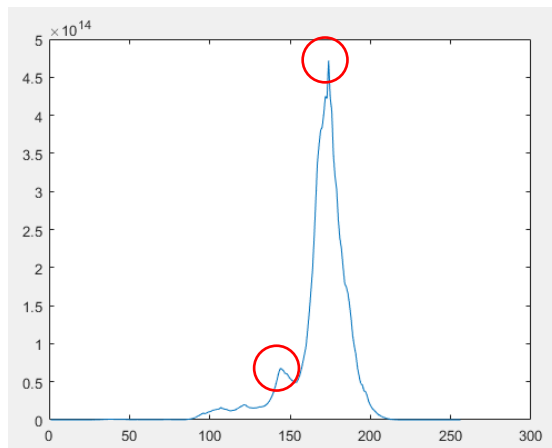
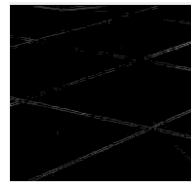
Red colour component is chosen for Haar wavelet transform



Horizontal-detail coefficients and vertical-detail coefficients were divided into sub-areas



with equal size.



A graph is plotted from bar chart. Bar chart is made from the accumulation of wavelet coefficients into bins range from 0 to 255.

Based on the graph, two colour intensities were used as colour thresholds to segment the sub-area of the red layer of test image 4. Image segmentation is further discussed in chapter 5.4.

Figure 5.3.c: The process of creating a local bar chart and local graph from a sub-area of test image 4

After processing the sub-areas in all the 4 arrangements of all 3 colour layers, the edges were united using logical OR operator. Figure 5.3.d shows the binarized test image 4.

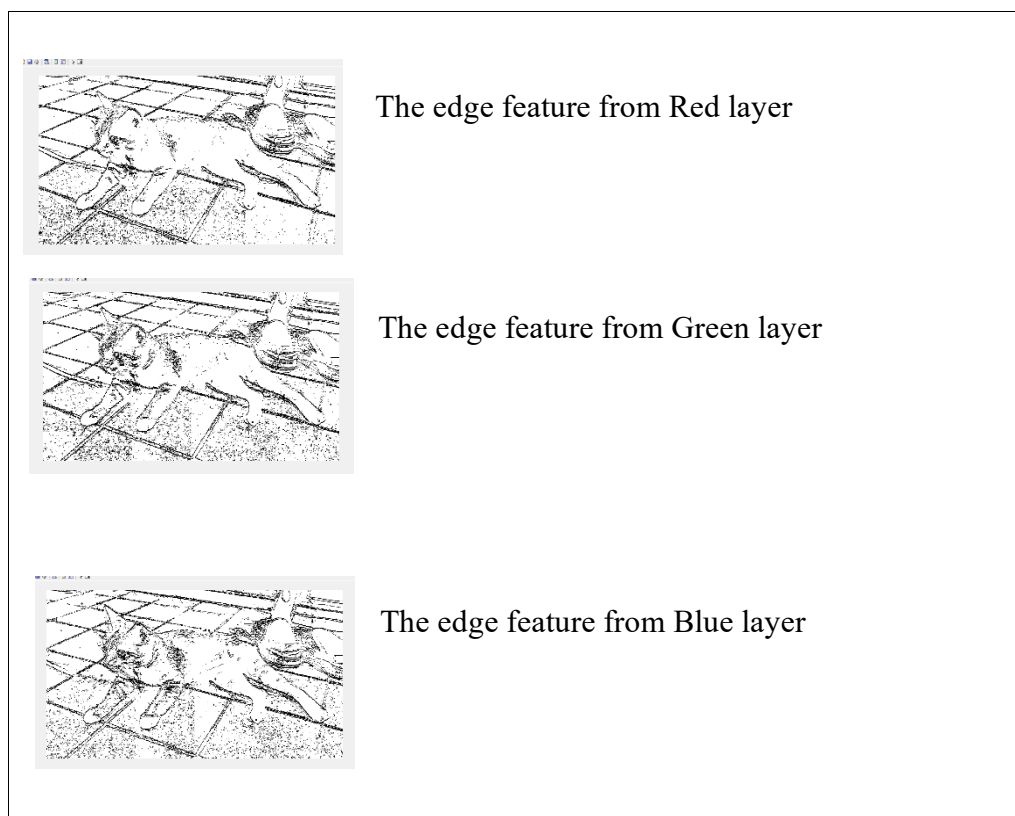


Figure 5.3.d: Edge features of test image 4 created by local colour threshold analysis

The edges presented in Figure 5.3.d were generated based on colour variance between pixels. Edges were found surrounding regions with brightness or colour that are constant or approximately constant. Enclosed regions were created and distinguished by the colour thresholds. Blob detection is used to

detect regions and could signal the presence of objects or parts of objects in the image. Section 5.4 discussed the purpose of image segmentation based on the enclosed regions to verify presence of the objects.

5.4 Image Segmentation

Before classification, image segmentation is done based on the regions separated by the edges and blobs features. The segmented area represents group of pixels with identical colour or similar colour, which signifies an object or part of the object. Image segmentation is crucial in specifying the desired area of the image to train the classification algorithm. After image segmentation, statistical properties of the desired areas were then computed to represent the uniqueness of specific object. Section 4.4 deliberated that the colour, brightness, size, shape, and texture of the group of pixels also serve as the categorical features in classification.

Group of pixels selected from the test images as shown in Figure 5.4.a, are based on the edges and blobs features made from colour threshold analysis of wavelet coefficients. Haar wavelet filter is chosen for the Method 10 wavelet transform to generate the wavelet coefficients. The sharpness of the wavelet coefficients as presented in section 5.2 is vital in the process of creating edges and blobs features. If Db3 or Db6 wavelet filters were used in generating wavelet coefficients, the insignificant edges created from wavelet transform of all the test images will lead to failure in detecting blob features. Image segmentation may be unsuccessful, blue rectangle will not be available as the pixels were not grouped accordingly. Method 10 is proposed to generate the sharpest wavelet coefficients for edges and blobs features based on the results in section 5.2.



Figure 5.4.a presents the chosen object of all test images after image segmentation to build dataset

As shown in Figure 5.4.a, segmented object bounded inside the square box in test image 1 is the bottom half of a pylon. The top half pylon was not grouped together caused by the white stripe. There is an edge drawn in the middle of the pylon during the colour threshold analysis. The white stripe on the red cone forms a remarkable change in colour intensity in between the adjacent pixels. The top half of the pylon thus separated from the bottom half after image segmentation process. The object segmented from test image 2

experience the same problem. The pylons as shown in test image 1 and test image 2 were segmented into 3 notable regions. All three regions will be identified as an individual part of the object and compute the statistical properties separately and assigned with a different object ID.

Test image 3 shows the yellow painted area of a road sign is identified as an object with the traffic light symbol is excluded after image segmentation. Test image 4 presented the lower body part of a cat is segmented as an object separated the head from the body. There is a dark brown region near the neck of the cat with smooth edges that split the cat into two different objects. The colour tone of near the neck of the cat is darker due to the gloominess, different in light intensity caused by the environment.

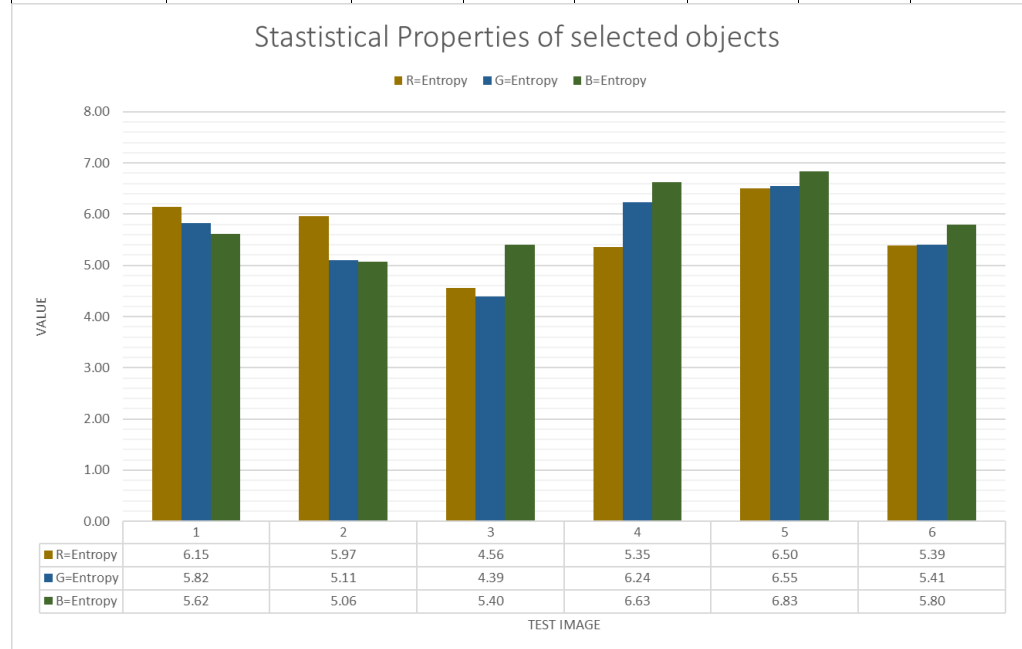
Test image 5 presented a cat with rough surface texture and the cat body is divided into numerous small regions after image segmentation. Thus, the cat could not be grouped as a single object but only can be identified as a combinational object. However, the vibratory bowl feeder was able to identified as an object from test image 5.

Test image 6 presented the body part of a blue colour cup is successfully segmented as a single object. However, there is a colour different between the internal part of the cup. The internal part is embodied with group of darker blue pixels compared to the outer part of the cup. Thus, the internal part of the cup is identified as another object in image segmentation.

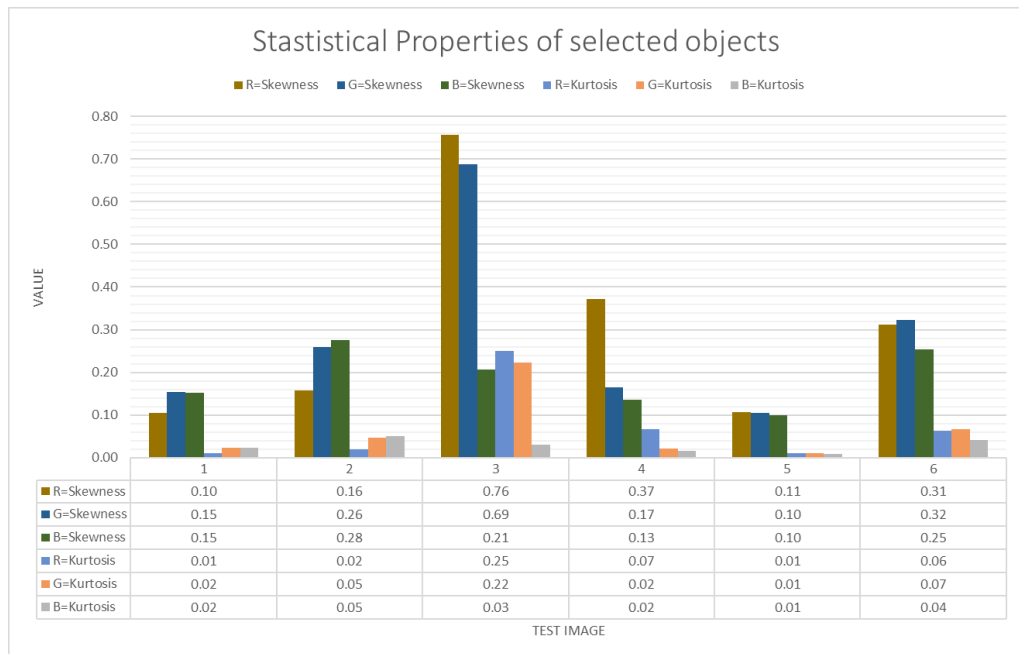
The statistical properties such as entropy, skewness, kurtosis, mean, and standard deviation of the chosen object from test images were tabulated in Table 5.4.a.

Table 5.4.a: Statistical Properties of the selected objects

	R G B layer	Test image 1	Test image 2	Test image 3	Test image 4	Test image 5	Test image 6
Entropy	R	6.15	5.97	4.56	5.35	6.50	5.39
	G	5.82	5.11	4.39	6.24	6.55	5.41
	B	5.62	5.06	5.40	6.63	6.83	5.80
Skewness	R	0.10	0.16	0.76	0.37	0.11	0.31
	G	0.15	0.26	0.69	0.17	0.10	0.32
	B	0.15	0.28	0.21	0.13	0.10	0.25
Kurtosis	R	0.01	0.02	0.25	0.07	0.01	0.06
	G	0.02	0.05	0.22	0.02	0.01	0.07
	B	0.02	0.05	0.03	0.02	0.01	0.04
Mean	R	115.92	110.95	235.35	205.97	61.53	26.44
	G	47.94	70.34	212.36	200.13	60.57	34.30
	B	54.08	63.95	133.57	179.85	63.41	89.35
Standard Deviation	R	24.08	17.09	6.32	12.05	26.71	12.68
	G	31.71	9.30	5.63	19.03	27.48	12.16
	B	33.75	9.05	12.75	28.10	33.08	15.03



Graph 5.4.a: Entropy of the selected objects



Graph 5.4.b: Skewness and Kurtosis of the selected objects



Graph 5.4.c: Mean and Standard Deviation of the selected objects

Graph 5.4.a and Graph 5.4.b presents the entropy, skewness and kurtosis of the selected objects without any outrageous characteristics. Based on Graph 5.4.c, pylon, road sign, cat, vibratory bowl feeder, and cup presented exotic mean values. Test image 1 and test image 2 pylons share a higher R-layer mean

value. The road sign shows higher mean value on both R and G components, whereas all the mean values of the cat and the vibratory bowl feeder exhibit similar mean value of all colour components. Lastly, the selected object in test image 6 revealed that B-layer has the highest mean value among the three-colour components. Image segmentation is a vital process to extract the region of interest inside all images. Statistical properties from region of interest is strategic in object-based classification which can improve the precision of classification by discarding the redundant information from neighboring pixels. Classification results is presented in the following section.

5.5 Accuracy of Classification








In this research work, object is distinguished by a group of pixels with identical colour or similar colour. Object classification is done based on the colour, brightness, texture, and shape characteristics of the object segmented from an image. From section 1, it is stated that this research work objective is to access the best classifier for image classification, the proposed object-based image classification is done with colour, brightness, and shape of the segmented object using decision trees and support vector machines.

There are numerous factors that can alter the accuracy of classification in this research, such as the brightness, sharpness, size, and complexity of the images, type of wavelet filters, method used in wavelet transform, colour threshold analysis, image segmentation, and the training dataset. In the research, decision trees classify objects based on the standards set by user whereas support vector machines use datasets chosen by user. The user is required to supervise the machine learning process before classification. The user can define and label each and every object into a class while the algorithm will compute the features and statistical properties for the classes. After image segmentation process, a list of objects will be ready and allow user to choose for training. The list will then go through image classification algorithm to classify the objects.










Both classification methods real-time performance was assessed using the same settings in wavelet transform, colour threshold analysis, and image



segmentation. In section 5.5.1, decision tree is trained with a small dataset consist of 35 images with different name defined in main category and sub-category. Section 5.5.2 uses the SVM model with 59 train images.

Table 5.5.a: Images in dataset for decision tree

No	Main Category	Sub-category	Train Images
1.	None	None	
2.	Flower	PurpleFlower	
3.	Flower	PurpleFlower	
4.	Flower	RedFlower	
5.	Flower	RedFlower	
6.	Flower	PinkFlower	
7.	Flower	PinkFlower	
8.	Flower	YellowFlower	

9.	Flower	YellowFlower	
10.	Pet	Cat	
11.	Pet	Cat	
12.	Pet	Cat	
13.	Pet	Cat	
14.	Pet	Dog	
15.	Pet	Dog	
16.	Pet	Dog	
17.	Pet	Dog	





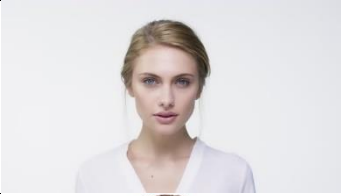

18.	Pet	Dog	
19.	Human	Face	
20.	Human	Face	
21.	Human	Face	
22.	Human	Face	
23.	Human	Face	
24.	Car	BlueCar	
25.	Car	BlueCar	
26.	Car	OrangeCar	

27.	Car	OrangeCar	
28.	Car	RedCar	
29.	Car	SilverCar	
30.	Fruit	Strawberry	
31.	Fruit	Orange	
32.	Fruit	Orange	
33.	Fruit	Banana	
34.	Fruit	GreenApple	
35.	Fruit	RedApple	

5.5.1 Decision Trees

The feasibility of this approach is verified using ten test images. Trained dataset was inserted into the classification algorithm and the final scores acquired from the classification are recorded in Table 5.5.1.b. Each and every column of Table 5.5.1.b indicates the score calculations from all the child nodes of the decision trees trained dataset.

Table 5.5.1.a: 10 random test images are numbered from 1 to 10

1. Flower, Amaryllis	
2. Mountain cornflower	
3. Apple	
4. Banana	
5. Female Face	
6. Male face	





7. Dog	
8. Cat	
9. Luxury Car	
10. Hatchback Car	

Table 5.5.1.b: Final scores taken from 10 test images' classification

Test Image	1	2	3	4	5	6	7	8	9	10
None	0	0	0	0	0	0	0	0	0	0
PurpleFlower	0	1446	0	0	0	0	0	0	0	0
PurpleFlower	0	1418	0	0	0	0	0	0	0	0
RedFlower	1680	0	0	0	0	0	0	0	0	0
RedFlower	1535	0	1385	0	0	0	0	0	0	0
PinkFlower	0	0	0	0	0	0	0	0	0	0
PinkFlower	0	0	0	0	0	0	0	0	0	0
YellowFlower	0	0	0	0	0	0	0	0	0	0
YellowFlower	0	0	0	0	0	0	0	0	0	0
Cat	0	0	0	0	0	1323	1361	0	0	0
Cat	0	0	0	0	0	0	0	1518	1326	0
Cat	0	0	0	0	0	0	0	0	0	0
Cat	0	0	0	0	0	0	1564	1482	0	0
Dog	0	0	0	0	0	0	0	0	0	0
Dog	0	0	0	0	1564	0	1493	1503	0	0
Dog	0	0	0	0	0	0	1555	1495	1331	0
Dog	0	0	0	0	0	0	0	0	0	1166
Dog	0	0	0	0	0	0	0	0	0	0
Face	0	0	0	0	1513	1494	1414	1422	0	0
Face	0	0	0	0	0	0	0	0	0	1152
Face	0	0	0	0	1561	0	1500	1440	1298	0

Face	0	0	0	0	1467	1478	0	1396	1258	0
Face	0	0	0	0	1396	1523	1285	1345	0	0
BlueCar	0	0	0	0	0	0	0	0	0	0
BlueCar	0	0	0	0	0	0	0	0	0	0
OrangeCar	0	0	0	0	0	0	0	0	1430	0
OrangeCar	0	0	0	0	0	0	0	0	1593	1429
RedCar	0	0	0	0	0	0	0	0	0	1401
SilverCar	0	0	0	0	0	0	0	1226	1388	0
Strawberry	0	0	0	0	0	0	0	0	0	0
Orange	0	0	0	0	0	0	0	0	0	0
Orange	0	0	0	0	0	0	0	0	0	0
Banana	0	0	0	1196	0	0	0	0	0	0
GreenApple	0	0	0	0	0	0	0	0	0	0
RedApple	0	0	1681	0	0	0	0	0	0	0

As shown in Table 5.5.1.b, test images are labelled from 1 to 10 and the scores are corresponding to the sub-categories at the first column. By inspecting the obtained results, all test images were successfully classified into its respective group through a collection of scores. In this research, test images were classified within a unique training set with high successful rate. The final scores gathered from decision tree went through a final recognition algorithm by excluding the impracticable scores with the remnants marked as the identified class. Final recognition stage performs basic arithmetic operations to the scores, turn out to be fraction, represents a part of a whole dataset. This classification is simple to understand and have value even with little hard data.

In section 5.5.2, SVM is trained with a small dataset consist of 58 images.

5.5.2 Support Vector Machines (SVM)

SVM is discussed in this research work section 3.5.6, this research requires the user to perform supervised training before SVM classification. The accuracy of the classification improves with more datasets stored by user. SVM will separate objects based on the name-value pairs defined by user during supervision training. SVM model trained by Matlab function *fitcsvm* stores the hyperplane and reinitialize the training process whenever user inserted new train image into the algorithm. The name-value pair shown in Table 5.5.2.a consists of main category and sub category of the trained images. In addition, the algorithm will discard the non-relevant features from the input image, and thus contributing towards a better-quality prediction score.

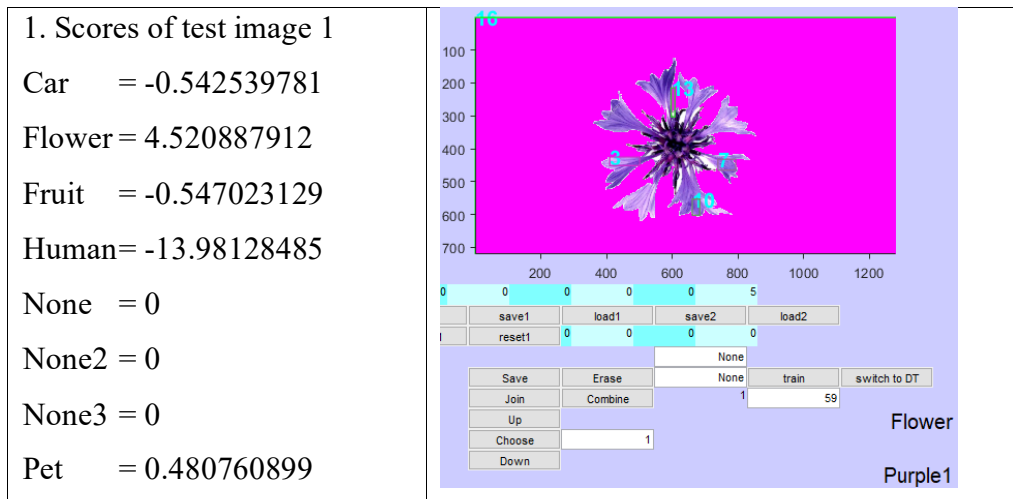
Table 5.5.2.a: Datasets for SVM classification

No	Main Category	Sub-category
1.	None	Non-relevant features1
2.	None2	Non-relevant features2
3.	None3	Non-relevant features3
4.	Flower	Red1
5.	Fruit	Banana1
6.	None3	Non-relevant features4
7.	None3	Non-relevant features5
8.	Fruit	Orange1
9.	None3	Non-relevant features6
10.	Fruit	GreenApple1
11.	None3	Non-relevant features7
12.	Fruit	RedApple1
13.	None3	Non-relevant features8
14.	Car	RedSport1
15.	None3	Non-relevant features9
16.	Human	Face1
17.	Human	Body1
18.	Human	Face2
19.	Human	Arm1

20.	None3	Non-relevant features10
21.	Pet	Cat1
22.	None3	Non-relevant features11
23.	Flower	Red2
24.	None3	Non-relevant features12
25.	Pet	Cat2
26.	Pet	Cat3
27.	Pet	Cat4
28.	Pet	Dog1
29.	Pet	Dog2
30.	Pet	Dog3
31.	Pet	Dog4
32.	Pet	Dog5
33.	Pet	Cat5
34.	Flower	Purple1
35.	Flower	Purple2
36.	Flower	Purple3
37.	Flower	Purple4
38.	Flower	Pink1
39.	Flower	Pink2
40.	Flower	Yellow1
41.	Flower	Yellow2
42.	Flower	Yellow3
43.	Car	TYTSilver1
44.	Car	Blue1
45.	Car	Blue2
46.	Car	Blue3
47.	Car	OrangeSport1
48.	Car	OrangeSport2
49.	Car	OrangeSport3
50.	Car	OrangeSport4
51.	Human	Face4
52.	Human	Face5
53.	Human	Face6
54.	Human	Face7
55.	Human	Face8
56.	Fruit	Strawberry1
57.	Fruit	Orange2
58.	Fruit	Orange3

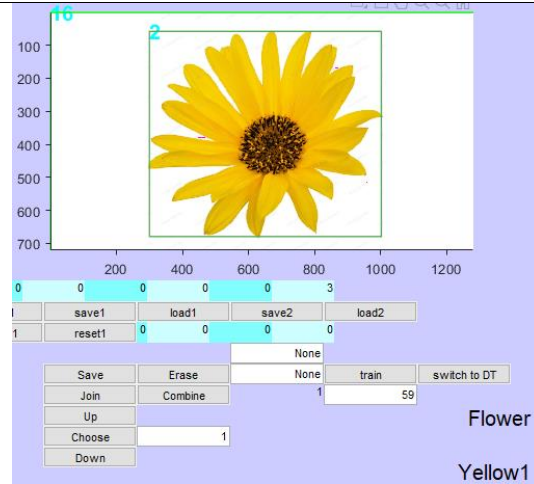
The predict function from Matlab generates the prediction scores. The research application compares these prediction scores and the labels of these sequences assigns high prediction scores to the correct class. Non-relevant features tend to pull down the scores of the white backgrounds and small fragmented areas. Therefore, objects with prediction score lower than mean threshold will not be given an ID number.

This method is proposed in section 4.6 and was given a sample size of 58 datasets to perform object-based image classification. Support vector machine model is trained using the *fitcsvm* function provided by Matlab. The 58 samples are from the same objects and different objects that were predefined with name-value pair argument.



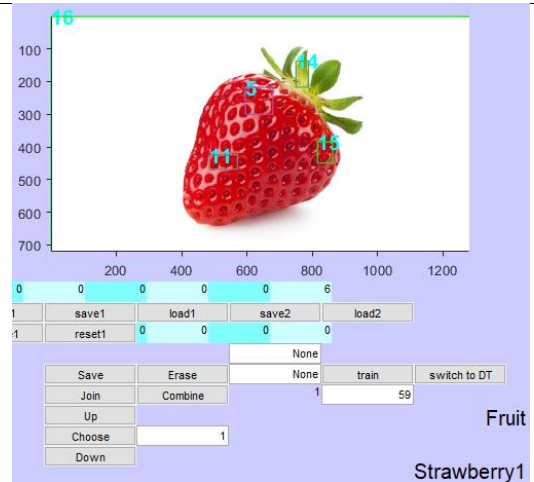
2. Scores of test image 2

Car = -5.320391714
 Flower = 11.11507022
 Fruit = -6.917091553
 Human = -27.83934665
 None = 0
 None2 = 0
 None3 = 0
 Pet = -5.899326503



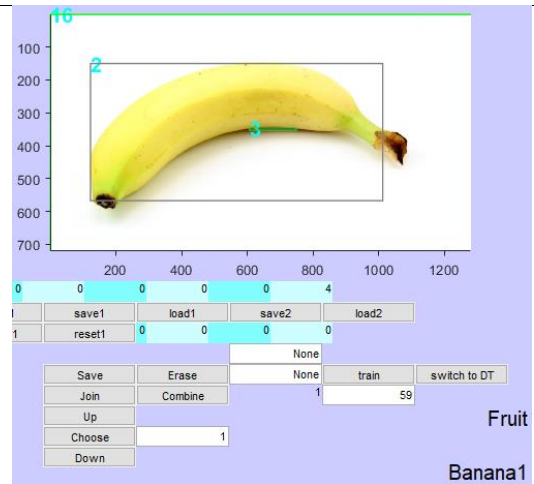
3. Scores of test image 3

Car = -0.686064551
 Flower = -1.804352453
 Fruit = 37.21746306
 Human = -4.610379096
 None = 0
 None2 = 0
 None3 = 0
 Pet = 0.637364902



4. Scores of test image 4

Car = -6.233850347
 Flower = -9.686592127
 Fruit = 33.46044512
 Human = -22.86575094
 None = 0
 None2 = 0
 None3 = 0
 Pet = 1.822343215



5. Scores of test image 5

Car = -2.302628928

Flower = -2.952728724

Fruit = 1.41440249

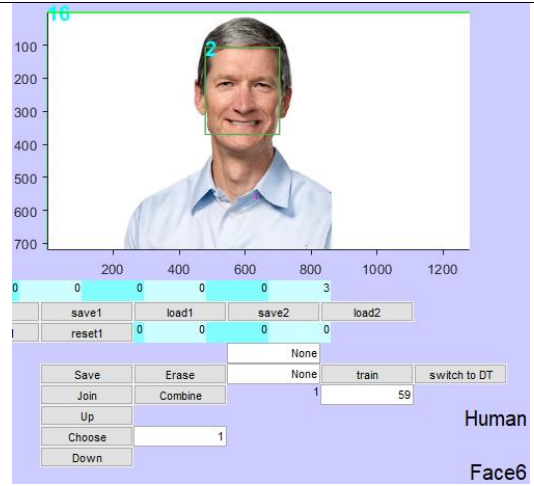
Human = 18.18239795

None = 0

None2 = 0

None3 = 0

Pet = 3.573838535



6. Scores of test image 6

Car = 1.906953181

Flower = 3.606675405

Fruit = 0.697376926

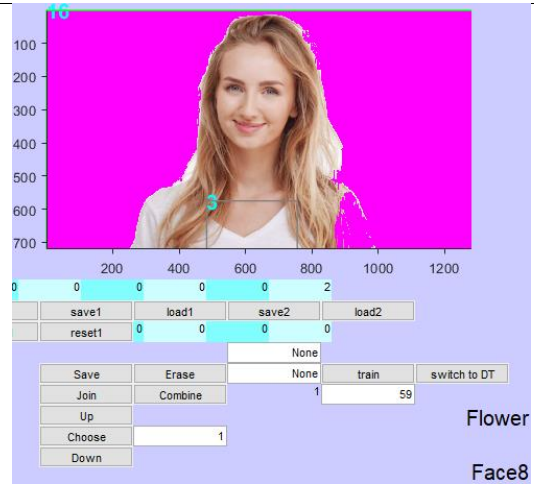
Human = 0.387885569

None = 0

None2 = 0

None3 = 0

Pet = 2.195119297



7. Scores of test image 7

Car = 0.453873309

Flower = 2.581605146

Fruit = 0.429796735

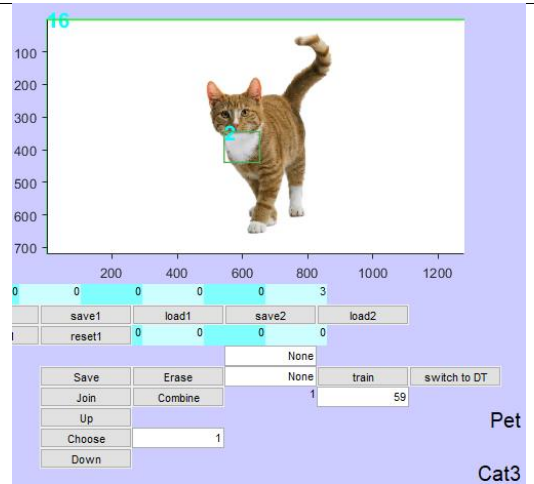
Human = -7.730157568

None = 0

None2 = 0

None3 = 0

Pet = 4.400722139



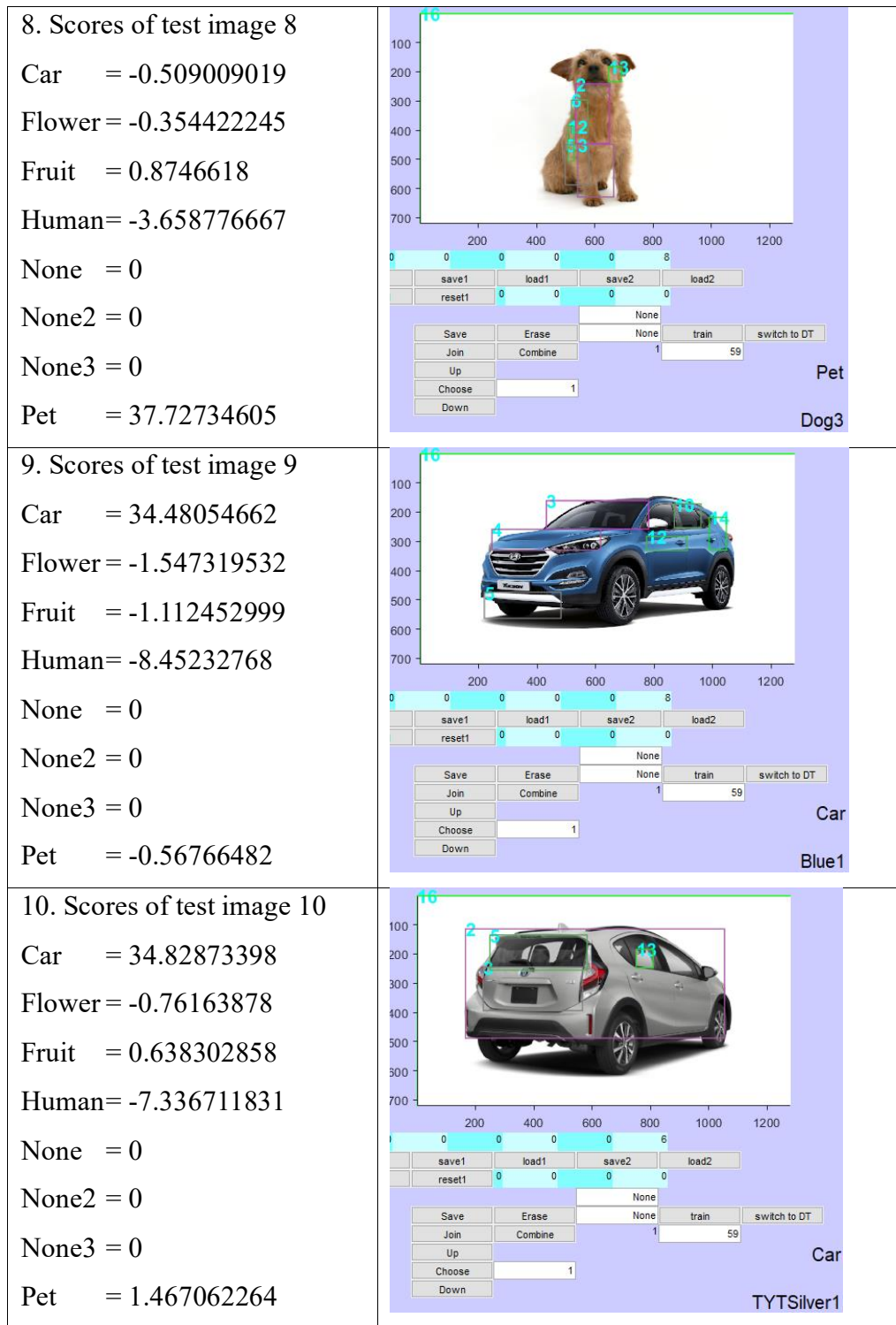


Figure 5.5.2.a: Final scores taken from 10 test images' classification

by support vector machine

As shown in Figure 5.5.2.a, Matlab *predict* function is used to identify the new objects through the scores output by the support vector machine model.

When an image is presented to this research algorithm, image is segmented and the features were processed by the *predict* function. This function returns a matrix of numbers correlate in the likelihood of the objects to the categories. Category with the highest score will be selected automatically, and an identification number will be given at the top left corner of all positive rectangular boxes. Non-relevant features scores inside the None, None2, and None3 groups will not receive an identification number and return zero by default.

In general, the classification algorithm in this research is very responsive with-it optimal performances. This SVM classification experimental results show the robustness of this algorithm and it is highly reliable and accurate because it is capable in detecting various kind of objects.

CHAPTER 6

CONCLUSION

In this section, the findings and results are summarized and interpreted. This research involves the use of wavelet coefficient features to identify and classify different images encountered by video surveillance systems.

6.1 Objective of This Research

There are two main objectives in this research. Primarily this research work investigates the fundamental properties of wavelet coefficients for feature selection for image classification. Parallel computing is integrated into this wavelet transform algorithm to resolve the real-time requirement constraints of this image classification application with a many-core architecture. The key of this research work is to enhance the productivity, application analyzability, and liability of this classifier in all sorts of situations and harsh environment.

6.2 Brief Review of Methodology

In this study, fundamental properties of wavelet coefficients are studied through various wavelet transform methods proposed in section 4. This research introduced 10 different methods to generate various wavelet coefficients. Parallel computing is then introduced to boost the process of wavelet transforms in this research to achieve real-time image processing. Wavelet coefficients is then processed using color threshold analysis algorithm presented in this research work to identify the edges for image segmentation. The most challenging step in this process is to determine the desired cluster center for a prodigious image segmentation. Classification methods such as decision trees and support vector machine are employed in this research work to classify the features attained from wavelet transform.

6.3 Brief Review of Results and Contributions of This Research

Results suggest the wavelet transform presented in Method 10 yield the best result among the rest with the best sharpness, highest mean and standard deviation value for all six test images. In addition, the computation time of Method 10 without employing Matlab Parallel Computing Toolbox is lower compared to other methods. However, workload distribution causes longer computation period when parallel computing is implemented. Evidence from subjective experiments of Method 8 and Method 9 shows a notable improvement in the performance using parallel computing when vast amount of

iteration is shared among the available workers.

Wavelet transform outperformed in compressing an image while preserving the details of the image for reconstruction. However, wavelet with smaller vanishing moment such as Haar wavelet is proposed for image classification because it reflects only the changes between adjacent pixel pairs, it uses only two scaling and wavelet function coefficients, thus calculating pair wise averages and differences. Haar wavelet does not have overlapping windows whereas Daubechies wavelet family uses overlapping windows, resulting in reflects the changes of intensities between multiple pixel pairs thus smoothen the wavelet coefficients by averaging all pixel pairs.

In addition, this research utilized the colour threshold analysis method to determine the boundaries of the objects. Blob features is obtained by combining all the edge features acquired from colour threshold analysis. Thus, classification is then performed using the blob features. Decision trees classifier is proposed for the research because the results show that decision tree classification method in object-based classification outperformed compared to support vector machines. Decision trees performs classification based on standards set by the user whereas the support vector machines use the automated hyperplane to determine the likelihood of the unidentified objects.

6.4 Future Work

Image classification of this research work is developed and accustomed to compatible with the features output from the wavelet transform algorithms. Future works such as enhancing the wavelet transform output and establish a better image classification algorithm to improve the malleability of the system. Wavelet transform algorithm method 10 is proposed to act jointly with parallel computing in this research work for shorter processing time. This process is flexible to make any changes in the future work such as permeating color threshold analysis into this process which will massively reduce the computation time of image classification.

In conclusion, the work in this research shows that it is feasible to improve the computation time of object recognition and image classification using wavelet coefficients in the future work.

REFERENCE

- Abawajy, 2006 Abawajy, J.H., 2006, "Adaptive parallel I/O scheduling algorithm for multiprogrammed systems," *Future, Generation Computer Systems*, Vol. 22, Issue 5, pp. 611-619
- Acharya et al., 2018 Acharya, K., and Ghoshal, D., 2018, "Detection of A Shadow of Animated Video Frames in RGB Color Space," *Procedia Computer Science*, Vol. 132, pp. 103-108
- Affonso et al., 2017 Affonso, C., et al., 2017, "Deep learning for biological image classification," *Expert Systems with Applications*, Vol. 85, pp. 114-122
- Akkaş et al., 2015 Akkaş, E., Akin, L., Çubukçu, H., and Artuner, H., 2015, "Application of Decision Tree Algorithm for classification and identification of natural minerals using SEM–EDS," *Computer and Geosciences*, Vol. 80, pp. 38-48
- Alvaro et al., 2014 Alvaro, G. Z., Joao, B. F. and Odemir, M. B., 2014, "Gabor wavelets combined with volumetric fractal dimension applied to texture analysis," *Pattern Recognition Letters*, Vol. 36, pp. 135-143
- Anastasiya et al., 2014 Anastasiya, M., Vadim, S. and Gerhard-Wilhelm, W., 2014, "Sample size determination for logistic regression," *Journal of Computational and Applied Mathematics*, Vol. 255, pp. 743-752
- Andria et al., 2013 Andria, G., Attivissimo, F., and Lanzolla, A.M.L., 2013, "A statistical approach for MR and CT images comparison," *Measurement*, Vol. 46, pp. 57-65
- Arkaprabha et al., 2019 Arkaprabha, S. and Ishita, B., 2019, "Screening of anxiety and depression among the seafarers using machine learning technology," *Informatics in Medicine Unlocked*, Vol. 16, pp. 100149

- Barboza et al., 2017 Barboza, F., Kimura, H., and Altman, E., 2017, "Machine learning models and bankruptcy prediction," *Expert Systems with Applications*, Vol. 83, pp. 405-417
- Barsanti et al., 2013 Barsanti, R.J., and Athanason, A., 2013, "Signal compression using the discrete wavelet transform and the discrete cosine transform," *2013 Proceedings of IEEE Southeastcon, Jacksonville, FL 2013*, pp. 1-5
- Batenburg et al., 2009 Batenburg, K. J. and Sijbers, J., 2009, "Adaptive thresholding of tomograms by projection distance minimization," *Pattern Recognition*. Vol. 42, pp. 2297–2305
- Batenburg et al., 2009 Batenburg, K. J. and Sijbers, J., 2009, "Optimal Threshold Selection for Tomogram Segmentation by Projection Distance Minimization", *Transactions on Medical Imaging*, Vol. 28, pp. 676–686
- Beena mol et al., 2016 Beena mol, M., Mohanalin, J., Prabavathy, S., Jordina Torrents-Barrena and Domenec Puig, 2016, "A novel wavelet seismic denoising method using type II fuzzy," *Applied Soft Computing*, Vol. 48, pp. 507-521
- Bernard and David, 2013 Bernard, G. and David, P., 2013, "Limits of Instruction-Level Parallelism Capture," *Procedia Computer Science*, Vol. 18, pp. 1664-1673
- Bhandari et al., 2016 Bhandari, A.K., Kumar, A., Chaudhary, S. and Singh, G.K., 2016, "A novel color image multilevel thresholding based segmentation using nature inspired optimization algorithms," *Expert Systems with Applications*, Vol. 63, pp. 112-133
- Bin Mo et al., 2019 Bin Mo, Cuiqiong Chen, He Nie and Yonghong Jiang, 2019, "Visiting effects of crude oil price on economic growth in BRICS countries: Fresh evidence from wavelet-based quantile-on-quantile tests," *Energy*, Vol. 178, pp. 234-251
- Calderón, 1964 Calderón, 1964. "Intermediate spaces and interpolation, the complex Method," *Studia Mathematica*. Vol. 24, pp. 113–190
- Caruana et al., 2006 Caruana, R. and Niculescu-Mizil, 2006, "An empirical comparison of supervised learning algorithms," *Proc. 23rd International Conference on Machine Learning*, pp. 161-168
- Cervantes et al., 2015 Cervantes, J., Lamont, F.G., López-Chau, A., Mazahua, L.R., and Sergio Ruíz, J., 2015, "Data selection based on decision tree for SVM classification on large data sets," *Applied Soft Computing*, Vol. 37, pp. 787-798

- Chen et al., 2018 Chen, J.Y., Zheng, H.B., Lin, X., Wu, Y.Y., and Su, M.M., 2018, "A novel image segmentation method based on fast density clustering algorithm," *Engineering Applications of Artificial Intelligence*, Vol. 73, pp. 92-110
- Crowley et al., 1987 Crowley, J. L. and Sanderson, A. C., 1987, "Multiple resolution representation and probabilistic matching of 2-D gray-scale shape," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 9, pp. 113–121
- Daubechies, 1988 Daubechies, I., 1988, "Orthonormal bases of compactly supported wavelets," *Communications on pure and applied mathematics*, Vol. 41, Issue 7, pp. 909-996
- David et al., 2002 David J. Larson and John S. Nasstrom, 2002, "Shared- and distributed-memory parallelization of a Lagrangian atmospheric dispersion model," *Atmospheric Environment*, Vol. 36, Issue 9, pp. 1559-1564
- Dong et al., 2019 Dong, M.Y., Chang, C.P., Gong, Q., and Chu, Y., 2019, "Revisiting global economic activity and crude oil prices: A wavelet analysis," *Economic Modelling*, Vol. 78, pp. 134-149
- Genovese et al., 2015 Genovese, M., Bifulco, P., De Caro, D., Napoli, E., Petra, N., Romano, M., Cesarelli, M., and Strollo, A.G.M., 2015, "Hardware implementation of a spatio-temporal average filter for real-time denoising of fluoroscopic images," *Integration*, Vol. 49, pp. 114-124
- Goh et al., 2008 Goh, Craciun, S., Rao, S., Cheney, D., Gugel, K., Sanchez, J. C., and Principe, J.C., 2008, "Wireless Transmission of Neuronal Recordings Using a Portable Real-Time Discrimination/Compression Algorithm," *In Proc. the 30th Annual International Conference on Engineering in Medicine and Biology Society*, pp. 4439 – 4442
- Huang et al., 2005 Huang, K.Q., Wu, Z.Y., George, S.K. Fung, and Francis, H.Y. Chan, 2005, "Color image denoising with wavelet thresholding based on human visual system model," *Signal Processing: Image Communication*, Vol. 20, pp. 115-127
- Jiao et al., 2018 Jiao, Y., Wu, J.S., and Jiao, L.C., 2018, "An image segmentation method based on network clustering model," *Physica A: Statistical Mechanics and its Applications*, Vol. 490, pp. 1532-1542
- Jiao, L.C. et al., 2019 Jiao, L.C., et al., 2019, "A Survey of Deep Learning-Based Object Detection," *in IEEE Access*, Vol. 7, pp. 128837-128868

- Kekrea et al., 2016 H.B. Kekrea, Prachi Natu and Tanuja Sarode, 2016, "Color Image Compression Using Vector Quantization and Hybrid Wavelet Transform," *Twelfth International Multi-Conference on Information Processing-2016, India*, Vol 89, pp. 778-784
- Ko and Lee, 2015 Ko J.H., and Lee, C.M., 2015, "International economic policy uncertainty and stock prices: Wavelet approach," *Economic Letters*, Vol. 134, pp. 118-122
- Krishnan et al., 2018 Krishnan, N.M. Anoop, Mangalathu, S., Smedskjaer, M. M., Tandia, A., Burton, H., and Bauchy, M., 2018, "Predicting the dissolution kinetics of silicate glasses using machine learning," *Journal of Non-Crystalline Solids*, Vol. 487, pp. 37-45
- Kurtek et al., 2013 Kurtek, S., Su, J.Y., Grimm, C., Vaughan, M., Sowell, R., and Srivastava, A., 2013, "Statistical analysis of manual segmentations of structures in medical images," *Computer Vision and Image Understanding*, Vol. 117, pp. 1036-1050
- Kuznetsova et al., 2020 Kuznetsova, A., Rom, H., Alldrin, N. et al., 2020, "The Open Images Dataset V4," *Int J Comput Vis.* 128, pp. 1956-1981
- Le et al., 2012 Le, H.T., Tran, S.H., and Nguyen, T.T., 2012, "Image Classification using Support Vector Machine and Artificial Neural Network," *Information Technology and Computer Science*, Vol. 5, pp. 32-38
- Levkowitz et al., 1993 Levkowitz, Haim, Herman and Gabor T., 1993, "GLHS: A Generalized Lightness, Hue and Saturation Color Model," *CVGIP: Graphical Models and Image Processing*, Vol. 55, pp. 271-285
- Makieł et al., 2017 Makieł, W., and Gogolewski, D., 2017, "Variability Evaluation of Signal in Two-dimensional Wavelet Decomposition Using Fractal Dimension," *Procedia Engineering*, Vol. 192, pp. 243-248
- Maldonado et al., 2007 Maldonado, S.B., Arroyo, S.L., Jimenez, P.G., Moreno, H.G., and Ferreras, F.L., 2007, "Road-sign Detection and Recognition Based On Support Vector Machines," *IEEE transactions on intelligent transportation systems*, Vol. 8, no. 2, pp. 264-278
- Mallat, 1989 Mallat, S.G., 1989, "A theory for multiresolution signal decomposition: the wavelet representation," *IEEE Transactions on Pattern Recognition and Machine Intelligence*, Vol. 11, pp. 674-693

- Mallat, 2008 Mallat, S., 2008, "A Wavelet Tour of Signal Processing", 3rd ed. *The Sparse Way*, United States: Academic Press
- Manhas et al., 2012 Manhas, E., Brante, G., Souza, R. and Pellenz, M., 2012, "Energy-Efficient Cooperative Image Transmission Over Wireless Sensor Networks," In *Proc. the 2012 IEEE Wireless Communications and Networking Conference: Mobile and Wireless Networks*, Vol. 2, pp. 2014-2019
- Marta et al., 2017 Marta, R. Costa-jussà, Allauzen, A., Barrault, L., Cho, K., and Schwenk, H., 2017, "Introduction to the special issue on deep learning approaches for machine translation," *Computer Speech and Language*, Vol. 46, pp. 367-373
- MATLAB, 2018 MATLAB and Statistics Toolbox Release 2018b, The MathWorks, Inc., Natick, Massachusetts, United States.
- Meghana, 2012 Meghana, M. Deshpande, 2012, "Intelligent Video Surveillance System based on Wavelet Transform and Support Vector Machine", *Department of Electronics & Telecommunication MGM's Jawaharlal Nehru Engineering College, Aurangabad-431003, Maharashtra, India, International Journal of Computer Applications*, Vol. 48–No.14, pp. 42-45
- Michael, 1987 Michael, E.D., 1987, "The Convergence of Moore's/Mooers' law's," *Information Processing & Management*, Vol. 23, Issue 6, pp. 583-592
- Mohmad et al., 2011 Mohmad, B., Hesham, M., Khaled, B., and Ibrahim, A.A., 2011, "Suite of decision tree-based classification algorithms on cancer gene expression data," *Egyptian Informatics Journal*, Vol. 12, pp. 73-82
- Nasri et al., 2010 Nasri, M., Helali, A., Sghaier, H., and Maaref, H., 2010, "Adaptive image transfer for wireless sensor networks (WSNs)," In *Proc. 2010 International Conference on Design & Technology of Integrated Systems in Nanoscale Era*, Vol. 1, pp. 1 – 6
- Oinam et al., 2013 Oinam, S., Kumar, P.H., and Patil, S.B., 2013, "Compression of Time series signal using Wavelet Decomposition, Wavelet Packet and Decimated Discrete Wavelet compression Transforms Techniques and their Comparison," *International Journal of Advanced Research in Computer and Communication Engineering*, Vol. 2, Issue 3

- Parida, 2018 Parida, P., 2018, "Fuzzy clustering based transition region extraction for image segmentation," *Future Computing and Informatics Journal*, Vol. 3, pp. 321-333
- Paul Nii et al., 2019 Paul Nii, Ammah, T., and Owusu, E., 2019, "Robust medical image compression based on wavelet transform and vector quantization," *Informatics in Medicine Unlocked*, Vol. 15, pp. 100183
- Rajinikanth et al., 2015 Rajinikanth, V., and Couceiro, M.S., 2015, "RGB Histogram Based Color Image Segmentation Using Firefly Algorithm," *Procedia Computer Science*, Vol. 46, pp. 1449-1457
- Ramteke et al., 2011 Ramteke, M., and Srinivasan, R., 2011, "Integrating Graph-based Representation and Genetic Algorithm for Large-Scale Optimization: Refinery Crude Oil Scheduling," *Computer Aided Chemical Engineering*, Vol. 29, pp. 567-571
- Risse, 2019 Risse, M., 2019, "Combining wavelet decomposition with machine learning to forecast gold returns," *International Journal of Forecasting*, Vol. 35, pp. 601-615
- Rueda et al., 2015 Rueda, S., Knight, C.L., Papageorghiou, A.T., and Noble, J.A., 2015, "Feature-based fuzzy connectedness segmentation of ultrasound images with an object completion step," *Medical Image Analysis*, Vol. 26, pp. 30-46
- Sevcik, 1994 Sevcik, K.C., 1994, "Application scheduling and processor allocation in multiprogrammed parallel processing systems," *Performance Evaluation*, Vol. 19, Issues 2-3, pp. 107-140
- Shi and Shan, 2012 Shi, J.L., and Shan, Z.L., 2012, "Image resolution enhancement using statistical estimation in wavelet domain," *Biomedical Signal Processing and Control*, Vol. 7, pp. 571-578
- Smith et al., 1978 Smith and Ray, A., 1978, "Color gamut transform pairs," *Computer Graphics*, Vol. 12, pp. 12-19
- Song et al., 2018 Song, M.H., Li, K.H., and Kim, S.N., 2018, "Evaluation of periodicities and fractal characteristics by wavelet analysis of well log data," *Computer Science*, Vol. 119, pp. 29-38
- Stroemberg, 1994 Stroemberg, J.O., 1994, "Wavelet Transforms with the Franklin System and Applications in Image Processing," *Proceedings Of The Scandinavian Conference On Image Analysis*, Vol. 1, pp. 199-199

- Thomas, 1802 Thomas, Y., 1802, "II. The Bakerian Lecture. On the theory of light and colours," *Phil. Trans. R. Soc.* Vol. 92, Soc.92, pp. 12-48
- Wang et al., 2017 Wang, Y.M., Tian, G., and Yu, L.Y., 2017, "The effect of explosion parameters on seismic source wavelet calculation and its characteristics," *Journal of Applied Geophysics*, Vol. 145, pp. 50-58
- Wang et al., 2019 Wang, Y.J., Qi, Q.W., Liu, Y., Jiang, L.L., and Wang, J., 2019, "Unsupervised segmentation parameter selection using the local spatial statistics for remote sensing image segmentation," *International Journal of Applied Earth Observation and Geoinformation*, Vol. 81, pp. 98-109
- Wattimena et al., 2013 Wattimena, R.K., Kramadibrata, S., Sidi, I.D., and Azizi, M.A., 2013, "Developing coal pillar stability chart using logistic regression," *International Journal of Rock Mechanics and Mining Sciences*, Vol. 58, pp. 55-60
- Xiao et al., 2019 Xiao, Y., Kamat, V.R., and Menassa, C.C., 2019, "Human tracking from single RGB-D camera using online learning," *Image and Vision Computing*, Vol. 88, pp. 67-75
- Yu, H.Y. et al., 2015 Yu, H.Y., Zhi, X.B., and Fan, J.L., 2015, "Image segmentation based on weak fuzzy partition entropy," *Neurocomputing*, Vol. 168, pp. 994-1010
- Yue et al., 2012 Yue, X.D., Miao, D.Q., Zhang, N., Cao, L.B., and Wu, Q., 2012, "Multiscale roughness measure for color image segmentation," *Information Sciences*, Vol. 216, pp. 93-112
- Zeng et al., 2015 Zeng, W., Wang, H.T., Tian, G.Y., and Hu, G.X., 2015, "Application of laser ultrasound imaging technology in the frequency domain based on Wigner-Ville algorithm for detecting defect," *Optic and Laser Technology*, Vol. 74, pp. 72-78
- Zhang et al., 2017 Zhang, Z.W., VanSwearingen, J., Brach, J.S., Perera, S., and Sejdić, E., 2017, "Most suitable mother wavelet for the analysis of fractal properties of stride interval time series via the average wavelet coefficient method," *Computer in Biology and Medicine*, Vol. 80, pp. 175-184
- Zou et al., 2019 Zou, Z.X., Shi, Z.W., Guo, Y.H., and Ye, J.P., 2019, "Object Detection in 20 Years; A Survey," *Computer Vision and Pattern Recognition*, Vol. 1905 05055