

**LIE DETECTION ON PUPIL SIZE**

**A THESIS SUBMITTED TO THE**  
**GRADUATE SCHOOL OF APPLIED SCIENCES**  
**OF**  
**NEAR EAST UNIVERSITY**

**By**

**FATİH VEYSEL NURÇİN**

**In Partial Fulfillment of Requirements for the Degree**  
**of Master of Science**  
**in**  
**in Biomedical Engineering**

**NICOSIA, 2015**

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, last name: Fatih Veysel Nurçin

Signature:

Date:

## **ACKNOWLEDGEMENT**

Firstly and primarily, I would like to thank to my supervisor Assist. Prof. Dr. Elbrus Imanov for his guidance, kindness, patience and encouragement and support through this thesis process. I also want to thank to Assoc. Prof. Dr. Terin Adalı for providing us such environment that we can improve ourselves, as she has been always an example of dignity and honesty who truly cares of others for betterment of humanity. I cannot forget to also thank to Mr. Ali Işın, he has been always more than a lecturer to us, as a big brother, mentor, he never rejected to help us, with constant kindness he was there. And secretly I would like to thank to number 17. Lastly I m grateful to have my family to support me throughout my life, without them I wouldn't be in the neighborhood.

## ABSTRACT

A pupillometry is a device that measures diameter of pupil. Measuring diameter can be done for different reasons. However if person is not fixed to the pupillometry, diameter value could vary depending on distance between subject and device. To overcome this problem radius of iris was taken as reference to pupil as output will be ratio of pupil diameter to iris diameter.

In this thesis, Matlab environment was used to do image processing, segmentation of iris and pupil and neural classification. Image processing toolbox and neural network toolbox was employed in Matlab environment to do tasks without complexity of coding

In this thesis one of cues of deception, pupil dilation was studied, the dilated and non-dilated iris images were image preprocessed and segmentation of pupil and iris was achieved. Images were chosen from MMU Iris database and images with big pupils are assumed to be dilated pupils that are indication of lying and small ones were assumed to be neutral. After pre-processing that follows segmentation of pupil and iris, pupil to iris ratios as 60 samples of 1 element with 10 hidden neurons were fed into neural network for classification by neural network pattern recognition tool. All images were correctly classified with %99.99 accuracy.

**Keywords:** Pupillometry, cues of deception, pupil dilation, artificial neural network, image processing, image segmentation.



## ÖZET

Pupillometri, pupil yarıçapını ölçen bir cihazdır. Pupil ölçümü farklı sebeplerden ötürü yapılabilir. Fakat, eğer kişi ile cihaz sabitlenmemişse, pupil yarıçapı cihaz ile kişinin arasındaki mesafenin değişmesiyle değer farklılığı gösterebilir, buda yanlış ölçümlere sebep olur. Bu sorunu çözmek için, iris yarıçapı referans olarak alınmıştır. Pupil yarıçapının iris yarıçapına oranı, çıkış değeri olacaktır.

Bu tez çalışmasında, Matlab programı, görüntü işleme, iris ve pupil görüntü segmentasyonu ve çıkış değerlerini yapay sinir ağı aracılığıyla sınıflandırmak için ve programlama nın karmaşıklığıyla vakit kaybetmemek için kullanılmıştır.

Bu tez çalışmasında, aldatma işaretlerinden bir tanesi, pupil büyümesi çalışılmıştır, büyümüş ve nötr pupil görüntüleri önişlenmiş ve pupil ve iris segmentasyonu başarılmıştır. Görüntüler MMU Iris database' den alınıp, pupil yarıçağı büyük olanlar, büyümüş pupil olarak, yarıçapı küçük olan pupil görüntüleri de nötr olarak varsayılmıştır. Görüntü önişleminden sonra görüntü pupil ve iris segmentasyonu başarılmış ve pupil iris oranı, Neural Network Pattern Recognition toolbox yardımı ile 60x1 matrix şeklinde, 1 elementin 60 örneği şeklinde yapay sinir ağlarıyla kullanılmıştır. Öğrenim sırasında 10 gizli nöron kullanılmıştır. Bütün görüntüler %99.99 doğruluk ile sınıflandırılmıştır.

**Anahtar Sözcükler:** Pupillometri, aldatma işaretleri, pupil büyümesi, yapay sinir ağları, görüntü işleme, görüntü segmentasyonu.

# TABLE OF CONTENTS

<b>ACKNOWLEDGEMENT.....</b>	<b>iii</b>
<b>ABSTRACT .....</b>	<b>iv</b>
<b>ÖZET .....</b>	<b>v</b>
<b>LIST OF FIGURES .....</b>	<b>x</b>
<b>LIST OF TABLES .....</b>	<b>xii</b>
<b>LIST OF ABBREVIATIONS .....</b>	<b>xiii</b>
<b>CHAPTER 1: INTRODUCTION.....</b>	<b>1</b>
1.1 Introduction.....	1
1.2 Human Eye.....	3
1.3 Seeing.....	4
1.4 Changes in Pupil Size .....	4
1.5 Pupil Dilated Related to Cue of Deception.....	5
1.5.1 Arousal.....	5
1.5.2 Feelings while lying.....	5
1.5.3 Cognitive aspects of deception .....	5
1.5.4 Attempted control of verbal and non-verbal behaviors .....	5
1.5.5 Four factor theory of deception.....	6
1.6 Task Evoked Pupillary Responses .....	6
1.7 Lie Detection Techniques .....	6
1.8 Guilty Knowledge Test.....	6
<b>CHAPTER 2: IMAGE PROCESSING PHASE .....</b>	<b>7</b>
2.1 Image Pre-processing for Pupil and Iris.....	7

2.1.1 RGB to gray conversion.....	8
2.1.2 Thesis application of RGB to gray-scale .....	8
2.2 Morphological Image Processing .....	8
2.2.1 Dilation .....	9
2.2.2 Erosion .....	11
2.3 Combining Dilation and Erosion .....	12
2.3.1 Dilation and erosion in grey-scale .....	15
2.3.2 Opening and closing of gray-scale images .....	16
2.3.3 Thesis application of gray-scale opening and closing .....	17
2.4 Spatial Filtering.....	17
2.4.1 Linear spatial filtering.....	18
2.4.2 Smoothing and low pass filters .....	23
2.4.3 Sharpening and high pass filter.....	24
2.4.4 Unsharp masking .....	26
2.4.5 Thesis application of sharpening image.....	27
2.5 Image Enhancement in Spatial Domain.....	28
2.5.1 Image negatives .....	29
2.5.2 Log transformations .....	29
2.5.3 Thesis application of log transformation .....	30
2.5.4 Power-law transformation.....	30
2.6 Lie Detection on Pupil Size: Image Segmentation Phase .....	32
2.6.1 Introduction to image segmentation.....	32
2.7 Point, Line, and, Edge Detection .....	32
2.7.1 Point detection .....	33

2.7.2 Line detection.....	33
2.7.3 Edge detection.....	34
2.7.3.1 Type of edge detection techniques.....	35
2.7.3.1.1 Sobel edge detector .....	35
2.7.3.1.2 Prewitt edge detector.....	35
2.7.3.1.3 Roberts edge detector.....	36
2.7.3.1.4 Laplacian of Gaussian detector.....	36
2.7.3.1.5 Zero crossing detector.....	36
2.7.3.1.6 Canny edge detector.....	37
2.7.3.1.7 Application of canny edge detector in thesis .....	38
2.8 Thresholding .....	39
2.8.1 Global thresholding.....	40
2.8.1.1 Otsu's method .....	40
2.8.1.2 Application of Otsu's method in thesis.....	41
2.8.2 Local thresholding .....	42
2.9 Line Detection Using Hough Transform .....	43
2.9.1 Hough transformation circle detection.....	46
2.9.2 Application of circle detection by Hough transform in the thesis .....	47
<b>CHAPTER 3: CLASSIFICATION OF PUPIL SIZES .....</b>	<b>50</b>
3.1 Artificial Neural Network.....	50
3.2 Mathematical Model of Artificial Neural Network .....	50
3.3 Backpropagation Neural Network .....	51
3.4 System Database .....	52
3.5 Feature Extraction.....	52

3.6 Output Target Formation .....	53
3.7 ANN Structure .....	54
3.8 ANN Topology .....	54
3.9 Training of Neural Network.....	55
3.10 Testing and Validation of Neural Network.....	55
<b>CHAPTER 4: RESULTS AND DISCUSSION .....</b>	<b>56</b>
4.1 Results and Discussion .....	56
4.1.1 ANN results .....	57
4.2 Discussion .....	60
<b>CHAPTER 5: CONCLUSION.....</b>	<b>61</b>
<b>REFERENCES.....</b>	<b>62</b>
<b>APPENDIX.....</b>	<b>65</b>

## LIST OF FIGURES

<b>Figure 1.1:</b> Dilated pupil recognition.....	2
<b>Figure 1.2:</b> Pathway of image processing of the human eye. ....	3
<b>Figure 1.3:</b> Pupil dilation and pupil contraction. ....	4
<b>Figure 1.4:</b> Block diagram of pre-processing of pupil.....	7
<b>Figure 1.5:</b> Block diagram of pre-processing of iris.....	7
<b>Figure 1.6:</b> RGB formatted image converted into gray-scale.....	9
<b>Figure 1.7:</b> Structure elements with different shape and radius 3. ....	10
<b>Figure 1.8:</b> Original image and dilated image. ....	10
<b>Figure 1.9:</b> Original image and eroded image ....	11
<b>Figure 1.10:</b> Translation of structuring element B in A followed by complete opening. ....	13
<b>Figure 1.11:</b> Morphological closing.....	14
<b>Figure 1.12:</b> Original image, closing of image, opening of image. ....	14
<b>Figure 1.13:</b> Original image and opening and closing applied in series ....	17
<b>Figure 1.14:</b> 3 x 3 mask and its related neighborhoods are magnified ....	19
<b>Figure 1.15:</b> Process of one-dimensional correlation and convolution. ....	20
<b>Figure 1.16:</b> Process of two dimensional correlation and convolution.....	22
<b>Figure 1.17:</b> Smoothing mask. ....	23
<b>Figure 1.18:</b> Detail enhancing mask ....	24
<b>Figure 1.19:</b> Application of detail enhancing mask.....	24
<b>Figure 1.20:</b> Example of high pass filter.....	25
<b>Figure 1.21:</b> Detail enhanced image and high pass filtered image. ....	25
<b>Figure 1.22:</b> Output of unsharp masking on right.....	26
<b>Figure 1.23:</b> Sharpened image and image before sharpening ....	27
<b>Figure 1.24:</b> Type of basic transformations. ....	28
<b>Figure 1.25:</b> Inverted pupil size for closing and detection of circles in dark polarity. ....	29
<b>Figure 1.26:</b> Image before log transformation and image after log transformation ....	30
<b>Figure 1.27:</b> Different transformation curves generated by varying $\gamma$ .....	31
<b>Figure 1.28:</b> A mask of point detection ....	33
<b>Figure 1.29:</b> Type of line detectors.....	33

<b>Figure 1.30:</b> Edge detector types; Sobel, Prewitt, Roberts .....	37
<b>Figure 1.31:</b> RGB to gray converted image, log transformation, canny edge detection .....	39
<b>Figure 1.32:</b> RGB to grey converted, series of dilations and erosions and Otsu's method .....	42
<b>Figure 1.33:</b> xy-plane and parameter space .....	44
<b>Figure 1.34:</b> Parameterization of lines, $\rho\theta$ plane and accumulator cells .....	45
<b>Figure 1.35:</b> RGB to gray conversion, opening and closing, thresholding, circle detection ...	47
<b>Figure 1.36:</b> RGB to gray, log transformation, canny edge detection, Hough transform.....	48
<b>Figure 1.37:</b> Preprocessing and segmentation of pupil.....	49
<b>Figure 1.38:</b> Preprocessing and segmentation of iris .....	49
<b>Figure 1.39:</b> Pupil to iris ratio to be fed into the neural network.....	49
<b>Figure 2.1:</b> Formal neuron .....	50
<b>Figure 2.2:</b> Illustration of backpropagation algorithm.....	51
<b>Figure 2.3:</b> Dilated and non-dilated pupil.....	52
<b>Figure 2.4:</b> Segmentation of pupil and iris and their ratio as sample value.....	53
<b>Figure 2.5:</b> Topology of the neural network .....	54
<b>Figure 3.1:</b> Best validation performance acquired at epoch 25 .....	57
<b>Figure 3.2:</b> Confusion Matrix .....	58
<b>Figure 3.3:</b> True positive rates .....	59

## LIST OF TABLES

<b>Table 1.1:</b> Example of 60x1 samples of 1 element.....	53
<b>Table 1.2:</b> Output values of classification.....	53
<b>Table 1.3:</b> Parameters of ANN.....	55
<b>Table 2.1:</b> Results of ANN.....	57



## LIST OF ABBREVIATIONS

<b>CQT:</b>	Control Question Technique
<b>GKT:</b>	The Guilty Knowledge Test
<b>JPEG:</b>	Joints Photographic Experts Group
<b>RGB:</b>	Red Green Blue
<b>ANN:</b>	Artificial Neural Network
<b>GUI:</b>	Graphic User Interface
<b>ROC:</b>	Receiver Operating Characteristic
<b>T:</b>	Threshold
<b>BP:</b>	Backpropagation
<b>IPT:</b>	Image Processing Toolbox
<b>CHT:</b>	Circle Hough Transform
<b>MMU:</b>	Multimedia University
<b>Log:</b>	Logarithmic
<b>ms:</b>	millisecond
<b>mm:</b>	millimeter

# CHAPTER 1

## INTRODUCTION

### 1.1 Introduction

Pupil size is very important parameter in psychophysiology. In psychophysiology it can be used to catch unconscious behaviors of individuals, which can work as one of the lie detection parameters (Goldwater, 1972).

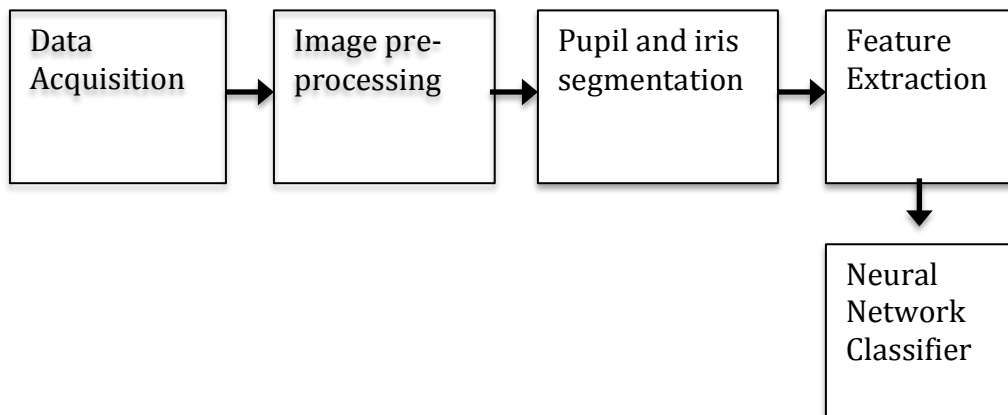
In this thesis, our focus is to develop intelligent, cost effective system that is going to use image processing and neural network to classify change in pupil size, which can detect differences in size of pupil. Current pupillometers measure diameter of the pupil but they don't take reference such as limbus or size of iris, which are anatomical structures of the eye. Problem arises from here that you have to be in certain range with pupillometry and even small changes with distance to device and can make little differences in measurement. Our solution to this problem is that we are going to take iris as reference point and compare pupil size to iris size, in this method varying distance with camera wouldn't make any difference. In this thesis, we are going to use MMU-Iris database, we picked dilated and non-dilated iris images as examples, however these images are not taken during a research about deception, dilated images are independent of real psychophysiological changes, we just want to apply our idea to iris images with image processing techniques to segmentate pupil and iris to compare their size, which can be developed on applications and used for further studies. 60 samples were separated as non-dilated and dilated depending on their size for classification with artificial neural network.

Pupil dilates (mydriasis), if there is low light to let in more light and constricts if there is over exposure, which is controlled by sympathetic, and parasympathetic nervous system unconsciously. Pupil diameter can change from 1.5 to more than 9 millimeters in man and can react in .2 seconds, beside having those changes against illumination its known fact that, dilation of pupil can occur by psychological reasons (Lowenstein & Loewenfeld, 1962). In 1920 Lowenstein reported pupil dilation can occur against suggestion-induced states as

“excitement,” “comfort,” “pleasure,” and “displeasure,” as well as suggestions of approaching pain and threat. In 1943 in another study, Berrien and Huntington measured pupil diameter by using short focus telescope with adjusting cross hairs. In experimental lie-detection situation they found that pupil dilated during %50 of the critical words and %15 of the neutral words. In 1920 Lowenstein claimed dilation of pupil could occur by intellectual processes. In 1964 by Hess and Polt, pupil dilation under mental activity observed, using verbally presented 4 multiplication problems of varied difficulties. Pupil size started to gradually dilate and reached to its peak before subject’s verbal report. The Dilation was related to degree of problem difficulty (Goldwater, 1972).

Pupil dilation can also occur for medical reasons. Damage to retina can cause permanent dilated pupil as in as in toxoplasmosis or damage to optic nerve as in avitaminosis A. It can be induced by drug or caused by disease (Mydriasis, n.d).

We went through 5 main stages as they are shown in Figure 1.1.



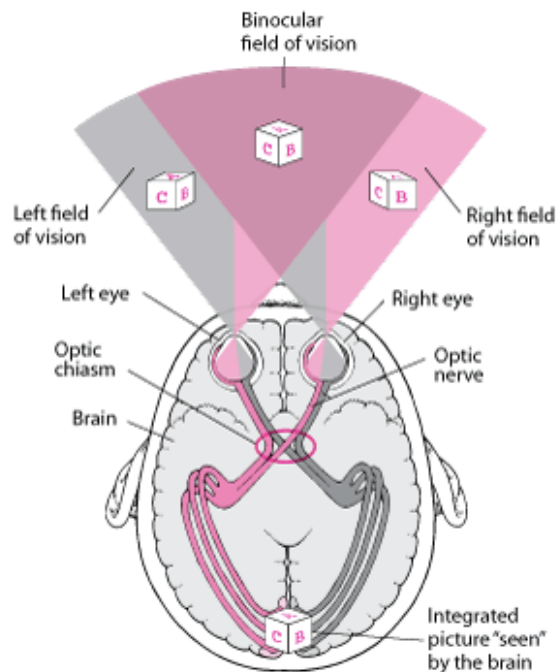
**Figure 1.1:** Dilated pupil recognition

The first part is data acquisition. Instead of taking the images from camera, we got our database from Internet, which is called MMU Iris database. Afterwards we have pre-processing, which includes noise filtering and contrast enhancement that follows image segmentation. In this part we segmentate pupil and limbus from background to measure radius of them, once we have pupil to limbus (location of iris border) ratio we can use this data as our

feature in artificial neural network and we can call it feature extraction, at final we feed our data into artificial neural network. We have total of 60 samples to feed into neural network. We used our pupil-iris ratio as our feature matrix as 1x1 single cell matrix and fed into artificial neural network as supervised learning with backpropagation method with total number of 60 inputs and corresponding 60 outputs.

## 1.2 Human Eye

It is said that, eyes are medium instead of the organ where visualization occurs, eyes carry information through optic nerve, chiasm and visual tract to certain places of brain end lobe where image we see is formed. For each eye, left side of the retina transfers left side of image through optic nerve to brain (Figure 1.2), same principle applies to right side of the retina. Then two part of image is processed by brain to create final merged image (“Eye structure and functions”, n.d.).



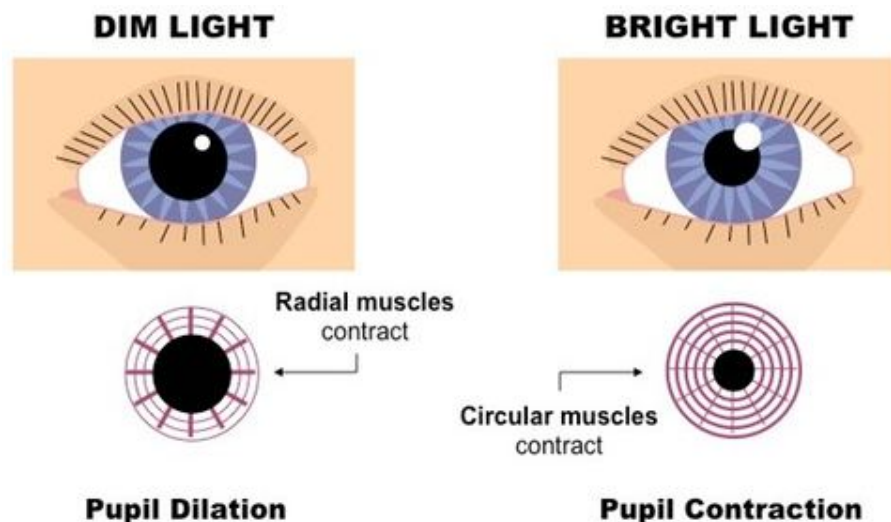
**Figure 1.2:** Pathway of image processing of the human eye (Garrity, 2013)

### 1.3 Seeing

Light rays come through cornea, pupil and lens are focused on retina, from retina image is transferred to the brain via optic nerve as explained earlier. Pupil is aperture of iris and depending on the light in the area, it can change its size. Figure 1.3 shows anatomical places of eye structures.

### 1.4 Changes in Pupil Size

Change in pupil size occurs due many reasons, it's know fact that pupil react to the low light condition as dilation and bright light condition as constriction. However changes in pupil size is also known as indicators of emotional arousal as well as medical state. Sympathetic system is responsible with pupil dilation by controlling the radial dilator muscle of pupil due activation of sympathetic system stimulation and as activation decreases diameter decreases. Parasympathetic system controls constriction of pupil with sphincter muscle of iris as reflex reaction to light via activation of parasympathetic system stimulation with efferent pathway originating in the edinger-westphal complex of oculomotor nucleus (Granholm & Steinhauer, 2004).



**Figure 1.3:** Pupil dilation and pupil contraction (“The human brain”, n.d., para 4)

### **1.5 Pupil Dilated Related to Cue of Deception**

Zuckerman et al. (1981) suggested that no single or set of behaviors would always occur during lying never occurs any other time which became widely accepted premise of him. Instead, feelings, psychological processes and kind of thoughts are more or less often during lying compared with truth telling. Then, they came up with four factors theory, where they explained generalized arousal, the specific affects experienced during deception, cognitive aspects of deception, and attempts to control behavior.

#### **1.5.1 Arousal**

Zuckerman et al. (1981) proposed that liars experience greater undifferentiated arousal compared to truth tellers. Which can be observed as greater pupil dilation, more frequent speech disturbances, higher pitch and blinking. However he generally accepted that characteristic of deception may be explained by looking at different aspects experienced during lying.

#### **1.5.2 Feelings while lying**

Liars felt guilt about lying or fear of getting caught more often than truth tellers (Zuckerman et al., 1981).

#### **1.5.3 Cognitive aspects of deception**

Zuckerman et al. (1981) conceived that lying requires more complicated tasks than truth telling. The liars should maintain his answer logical according to their claims that others have been informed already, which can result as greater cognitive challenges and that can be predicted with greater pupil dilation, longer response times, more speech hesitations, and few illustrations.

#### **1.5.4 Attempted control of verbal and non-verbal behaviors**

Liars tend to control their own behavior in order to maintain their deception, which paradoxically result in cues. Liars' behavior seems less spontaneous compared to truth tellers. Controlling each type of impression is not easy task to do which eventually result in inconsistencies (Zuckerman et al., 1981).

### **1.5.5 Four factor theory of deception**

Zuckerman et al. (1981) called all of these four factor theory of deception and reflected on subjects as lying requires greater cognitive load than truth telling, which can result as more pupil dilation, longer response times, and in other signs of load.

### **1.6 Task Evoked Pupillary Responses**

Task evoked pupillary responses can cause 0.1-0.5 mm changes in pupil size with response delay 200-300ms while peaks at 1200ms (Kahneman & Beatty, 1966).

### **1.7 Lie Detection Techniques**

According to recent studies GKT technique overcomes some of concerns about former CQT technique, which is used with polygraph, which measures arousal. Many of critics led to research alternatives, specifically into cognitive load inducing lie detection techniques (Lykken, 1988).

### **1.8 Guilty Knowledge Test**

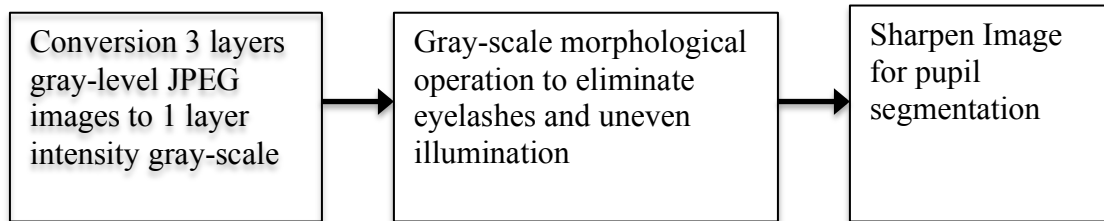
During GKT, multiple-choice questions are asked. Each question has one relevant alternative (correct answer) and several neutral (plausible distractors). The idea is that innocent person could not mark difference among answer from relative alternative. An example of relevant question is that “how was the victim killed” with having multiple-choice answers as “shot,” “stabbed,” “poisoned,” “strangled.” This question could be re-asked many times along with other questions which points to different aspects of crime scene. If examinee shows increased arousal consistently to relevant responses even without answering, examinee may be hiding information as someone who is involved in crime (Walczyk, 2013; Lykken, 1998).

## CHAPTER 2

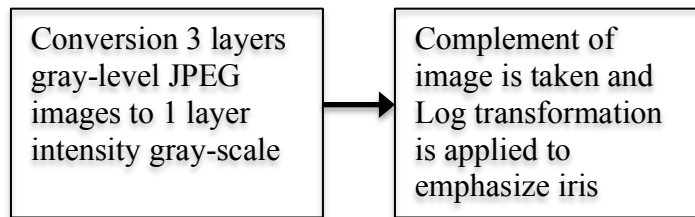
### IMAGE PROCESSING PHASE

#### 2.1 Image Pre-processing for Pupil and Iris

This chapter introduces techniques that are important in image processing and segmentation and their mathematical formulations in order to achieve operations. Before pupil and iris segmentation, the image has to be pre-processed to achieve significant level of segmentation. Figure 1.4 shows steps of processes were made.



**Figure 1.4:** Block diagram of pre-processing of pupil



**Figure 1.5:** Block diagram of pre-processing of iris

Figure 1.5 shows block diagram of iris pre-processing which leads to iris to differ from outer environment and eventually increase success of segmentation in next phase. Following topics explain included methods and their importance in the project. Every method is studied in details and at the end of each method; their application in our thesis is illustrated and explained.



### **2.1.1 RGB to gray conversion**

Raw data of iris images were in 3-layer gray-level JPEG format. To simplify image processing layers were dropped to 1 layer by MATLAB command, which is included in image processing toolbox and working as taking every 3 layer with multiplying red with 0.2989 and multiplying green with 0.5870 and blue with 11.40 for each pixel and then summing them all to have 1 intensity image (Mathworks, 2010).

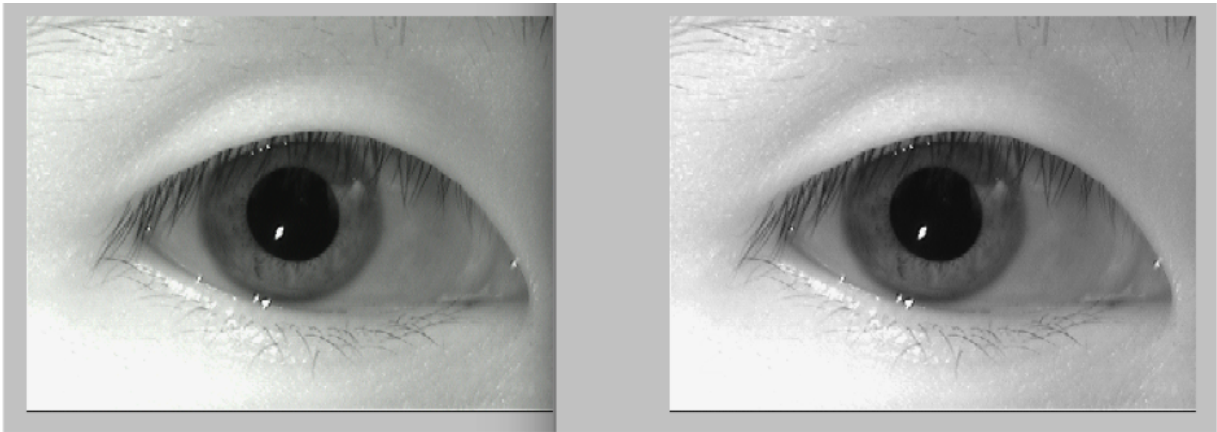
### **2.1.2 Thesis application of RGB to gray-scale**

Figure 1.6 shows conversion of 3 layers RGB formatted image to gray-scale to simplify operations. Since original image look lot like gray-scale image, we didn't lose anything by converting it to gray-scale. This conversion applied to both iris and pupil processes.

## **2.2 Morphological Image Processing**

Morphology points to branch of biology, which focuses on form and structure of animals and plants. In this context word is used as mathematical morphology, which works as a tool to extract image components to point out significant properties in meaning of region shape, which can achieve representation and description of the image.

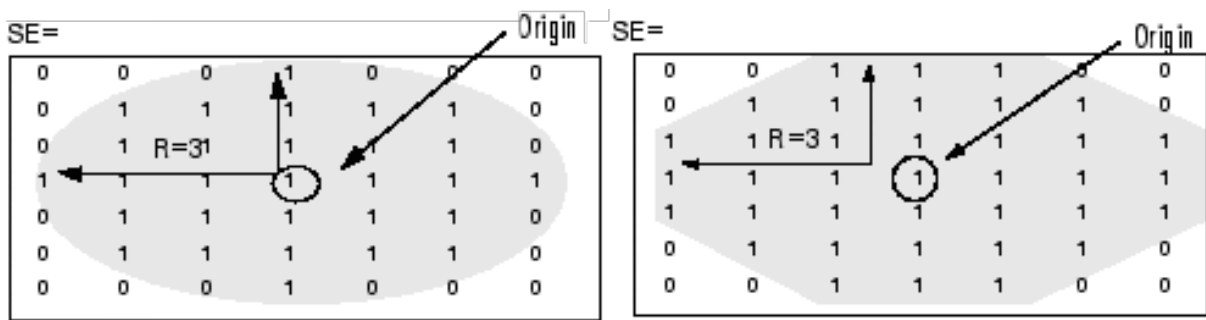
Mathematical morphology is usually applied to binary images. Image dilation and erosion are two fundamental morphological operations. For example, opening and closing are based on image dilation and erosion. As mentioned earlier, mathematical morphology is usually applied to binary images, which contain 1's and 0's, but it can be also applied on gray-scale images (Gonzales et al., 2004).



**Figure 1.6:** RGB formatted image (left) converted into gray-scale (right)

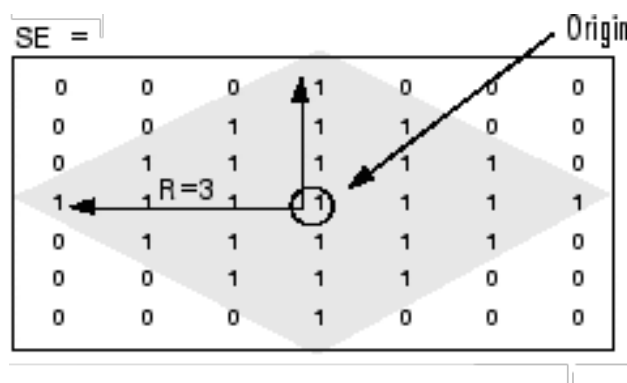
### 2.2.1 Dilation

Dilation is an operation that functions as growing or thickening of binary image. Dilation is controlled by a shape alluded as structuring element. Structuring element is formed with 1's and 0's. Origin of structuring element should be demonstrated. Different types of shapes could be used as structure element such as diamond, disk, line, octagon, rectangle and square as shown in Figure 1.7. These structuring element can be vary in size depending on the application, could be smaller or bigger and size of structuring element is defined with neighbors. Number of neighbors defines number of 1's that forms structuring element. Figure 1.8 shows dilated example of binary iris image. The image is dilated with disk shaped structuring element as shown in Figure 1.7 with radius taken as 3, which implies that, starting from center you have three 1's to the top and bottom and to the sides. Center of structuring element that targets 1 valued pixels and translated to the locations where structuring element doesn't overlap 1 valued pixels (Gonzales et al., 2004).



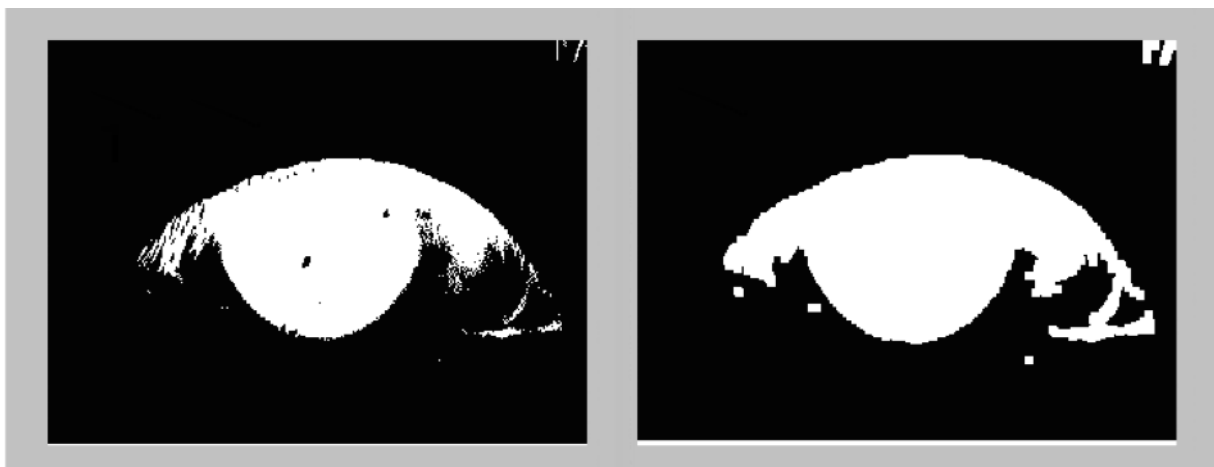
(a) Disk shaped

(b) Octagon shaped



(c) Diamond shaped

**Figure 1.7:** Structure elements with different shape and radius 3 (“Strel”, n.d.)



**Figure 1.8:** Original image (left) and dilated image (right)

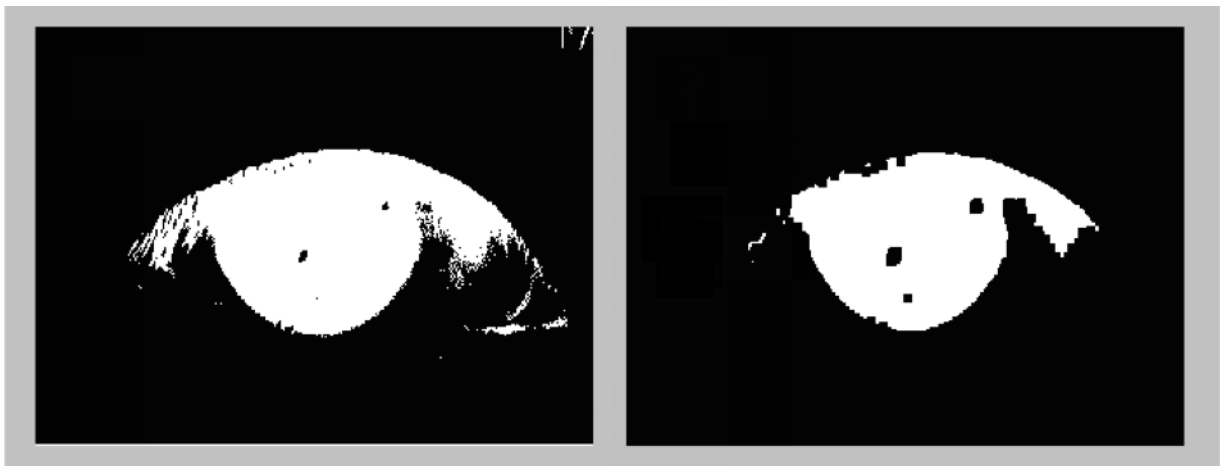
Mathematical Operation of dilation is explained as set of operations. Dilation is demonstrated as A by B, symbolized  $A \oplus B$ , is described as

$$A \oplus B = \{z | (B)_z \cap A \neq \emptyset\} \quad (1)$$

where A is binary image and B is structuring element,  $\emptyset$  is empty matrix. To explain the equation, dilation A by B is the set of all structuring element origin locations where the translated and reflected portion overlaps A (Gonzales et al., 2004).

### 2.2.2 Erosion

Erosion is an operation controlled by a shape that functions to shrink or thin the binary image, as we mentioned in dilation, extent and manner of shrinking is controlled by this shape named as structuring element. Translation of structure element goes through domain of image and inspect where it does fits as whole within the foreground pixels. If structuring element overlaps only 1-valued pixels in the input image, output value will have 1's at origin of structuring element (Gonzales et al., 2004).



**Figure 1.9:** Original image (left) and eroded image (right)

Mathematical operation of erosion is explained similar to dilation. Erosion of A by B is symbolized as  $A \ominus B$ , is defined as

$$A \ominus B = \{z | (B)_z \cap A^c \neq \emptyset\} \quad (2)$$

Figure 1.9 demonstrates eroded image.

### 2.3 Combining Dilation and Erosion

In practical applications, dilation and erosion are most used operation in morphological image processing as they are used together with various combinations. Series of Dilation and/or erosion can be applied to image with same or different structure elements (Gonzales et al., 2004).

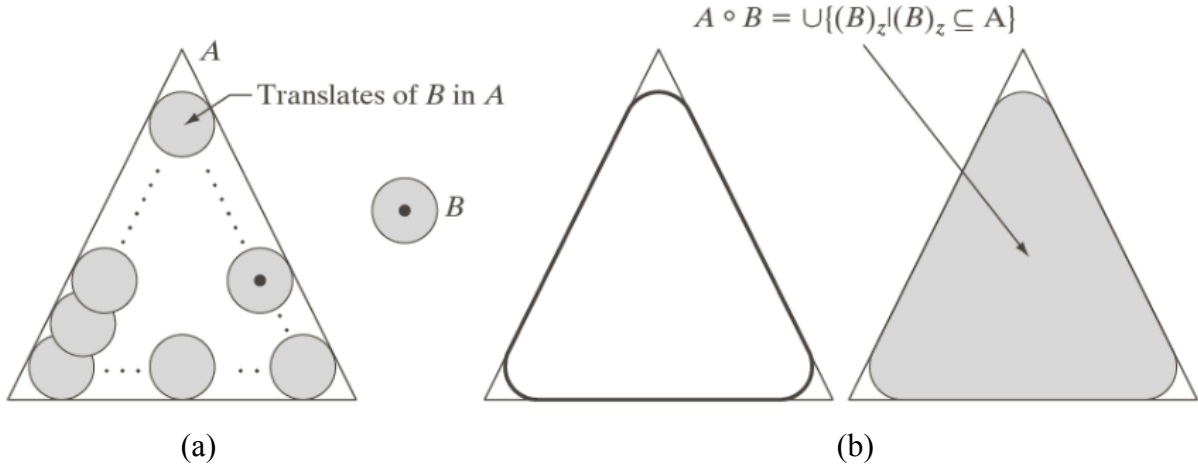
Mathematical operation of morphological opening of A by B, is symbolized  $A \diamond B$ , is statement of erosion of A by B, which is followed by dilation of the result by B:

$$A \diamond B = (A \ominus B) \oplus B \quad (3)$$

which can be also defined as

$$A \diamond B = \cup \{(B)_z \mid (B)_z \subseteq A\} \quad (4)$$

Symbol  $\cup \{\bullet\}$  describes union of all sets in the braces, and the symbol  $C \subseteq D$  is interpreted as that C is subset of D. Geometrical approach could be used to describe this formula. By bringing pieces together we can say that  $A \diamond B$  is union of all translations that fit completely within A (Gonzales et al., 2004).



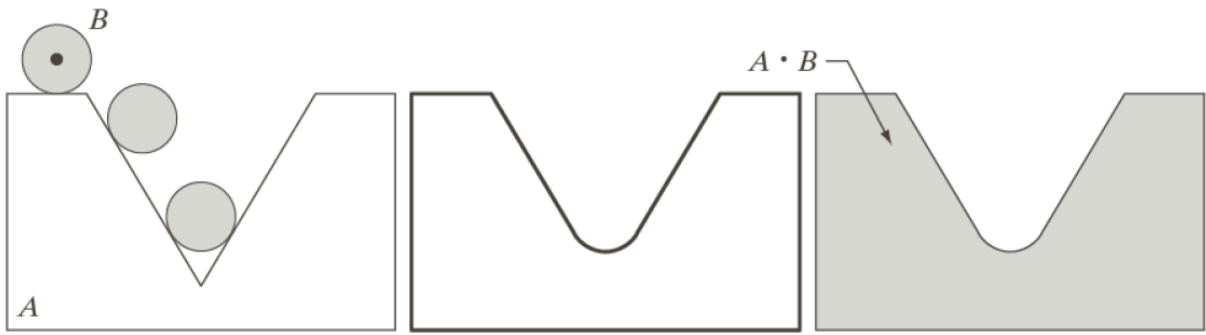
**Figure 1.10:** (a) Translation of structuring element B in A followed by complete opening (b) as shown in figure with shaded color (Gonzales & Woods, 2008)

As shown in Figure 1.10, places that structuring element couldn't fit could be seen. These regions as shown as white in shaded final image completely gets removed. In conclusion opening of image helps to smooth object corners, and removes thin spurs and connections (Gonzales et al., 2004).

Mathematical operation of morphological closing of A by B, is symbolized as  $A \bullet B$ , is statement of dilation A by B, which is followed by erosion of the result by B:

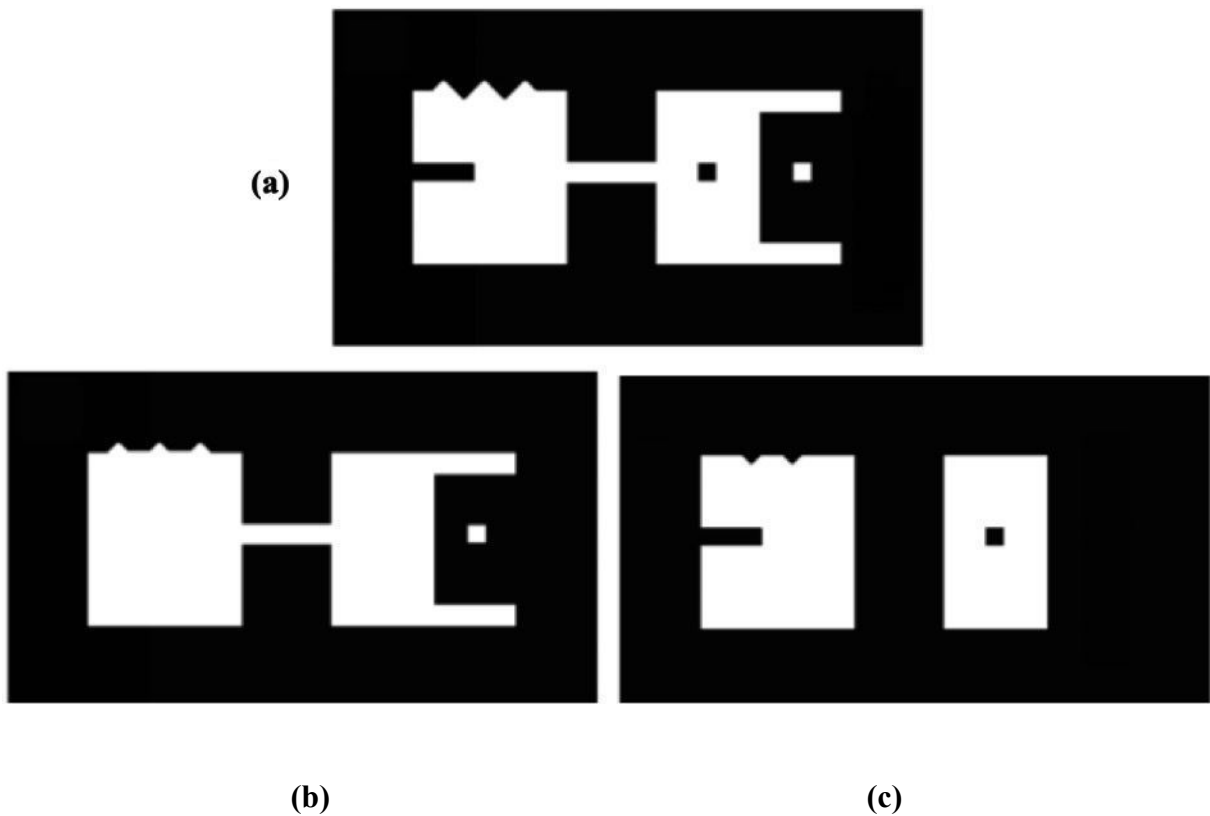
$$A \bullet B = (A \oplus B) \ominus B \quad (5)$$

Morphological Closing can be explained as taking complement of all translations of B, that don't overlap A as illustrated in Figure 1.11. In functionality and application of morphological closing, it does works to smooth contours like opening but differently it can fill spaces between corners, holes and fill rifts which are smaller than structuring element (Gonzales et al., 2004).



**Figure 1.11:** Morphological closing (Gonzales & Woods, 2008)

Another example of opening and closing that illustrates difference of opening and closing in Figure 1.12.



**Figure 1.12:** Original image (a), closing (b) and opening (c) of image, (Gonzales et al., 2004)

### 2.3.1 Dilation and erosion in grey-scale

Mathematical expression for grey-scale dilation  $f \oplus b$ , as  $f$  is image and  $b$  is structuring element is expressed as

$$(f \oplus b)(x, y) = \max\{f(x - x', y - y') + b(x', y') \mid (x', y') \in D_b\} \quad (6)$$

where domain of  $b$  is expressed as  $D_b$  and similar to convolution  $f(x, y)$  is assumed to be  $-\infty$  out of domain  $f$ . In a conceptual manner, it could be thought as similar to convolution ask structuring element rotated from its origin and translated to every pixel in an image. At Every translated location, structuring element (rotated from its origin) values are added to pixel values. The difference between convolution and grey-scale dilation is that, in grey-scale dilation  $D_b$  is a binary matrix controls which neighborhoods are included in max operation. To explain furthermore, sum of equation  $f(x - x', y - y') + b(x', y')$  as  $(x_0, y_0)$  is in the domain of  $D_b$ , the sum is included in max computation with  $D_b$  is being 1 at those coordinates, if  $D_b$  is zero, then sum is not taken into consideration for max computation. Same process goes for all location  $(x', y') \in D_b$  and coordinates  $(x, y)$  are changed each time,  $b(x, y)$  is plotted as  $x'$  and  $y'$  look like digital surface that have height at any pair of coordinates with value of  $b$  at these coordinates (Gonzales et al., 2004).

In practical application, height of  $b$  as also refer to value of  $b$  is 0 at all coordinates over  $D_b$ , which is defined as

$$b(x', y') = 0 \text{ for } (x', y') \in D_b \quad (7)$$

So the equation of gray-scale dilation simplifies to

$$(f \oplus b)(x, y) = \max\{f(x - x', y - y') \mid (x', y') \in D_b\} \quad (8)$$

Therefore, flat gray scale dilation can be called local maximum operator, shape of  $D_b$  determines the maximum over set of pixel neighbors.



Mathematical expression for grey-scale dilation  $f \ominus b$ , as  $f$  is image and  $b$  is structuring element is expressed as

$$(f \ominus b)(x, y) = \min\{f(x + x', y + y') - b(x', y') \mid (x', y') \in D_b\}, \quad (9)$$

where  $f(x, y)$  is assumed to be  $+\infty$  out of domain  $f$ . Structuring element is translated to all locations and for each location translated, in order to take minimum, image pixel values minus structuring element subtraction are performed (Gonzales et al., 2004). Since gray-scale erosion is also performed with flat structuring element same as dilation, equation is simplified to

$$(f \ominus b)(x, y) = \min\{f(x + x', y + y') \mid (x', y') \in D_b\} \quad (10)$$

### 2.3.2 Opening and closing of gray-scale images

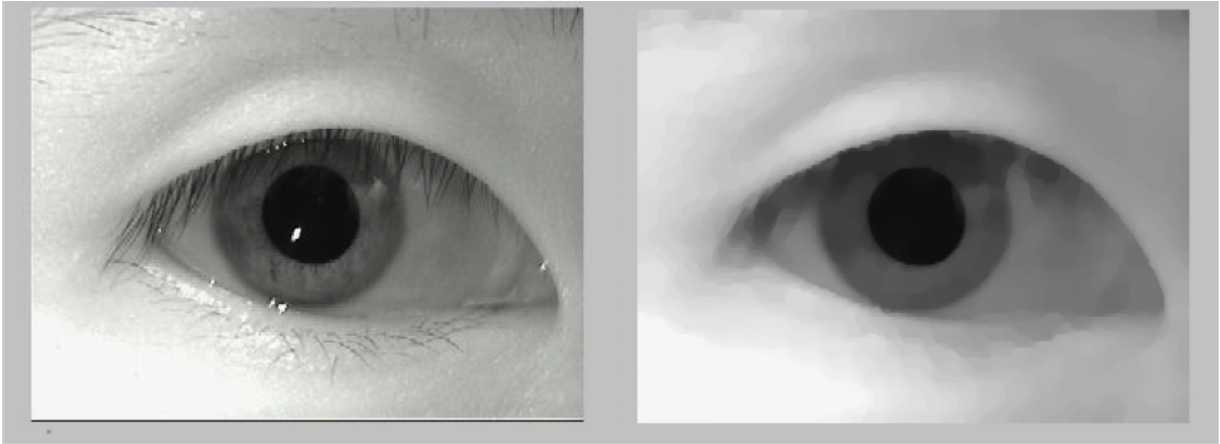
Opening and closing gray-scale images have same expression as they have in binary operations. Where opening of an image by structuring element is symbolized as  $f$

$$f \diamond b = (f \ominus b) \oplus b \quad (11)$$

$f \diamond b$  is simply erosion followed by dilation and  $f \bullet b$  is dilation followed by erosion.

$$f \bullet b = (f \oplus b) \ominus b \quad (12)$$

Opening and closing can be explained with simple geometrical approach, lets say  $f(x, y)$  is an image,  $(x, y)$  is a plane of coordinates and image values are height over the plane. So it can be explained as structuring element pushing from the underside of surface and translating all locations it moves and the highest point structuring element reaches determines construction of opening. Closing works with same principle only that it pushes from up and construction is dependent on lowest point reached by structuring element.



**Figure 1.13:** Original image (left) and opening and closing applied in series (right)

Opening and closing can be used in loop in order to open and close the image several times to achieve smoothing of background and details in object (Gonzales et al., 2004).

### 2.3.3 Thesis application of gray-scale opening and closing

Figure 1.13 illustrates application of gray-scale opening and closing on iris images, which is used in the project. Disk shaped structuring element is used in loop and loop iteration starts from 2-sized disk structuring element and ends at 5. This operation helps us get rid of eyelashes, some little part of eyebrows and reflected lights in the image. Simply it works to smooth images.

## 2.4 Spatial Filtering

When talking about spatial filtering, spatial domain has to be mentioned as well. Spatial domain is image plane and in spatial filtering, methods are applied to directly manipulate pixels in an image. Spatial filtering can be also referred to as spatial convolution, or neighborhood processing.

As mentioned before, spatial domain techniques directly manipulate pixels and these processes are denoted by this expression

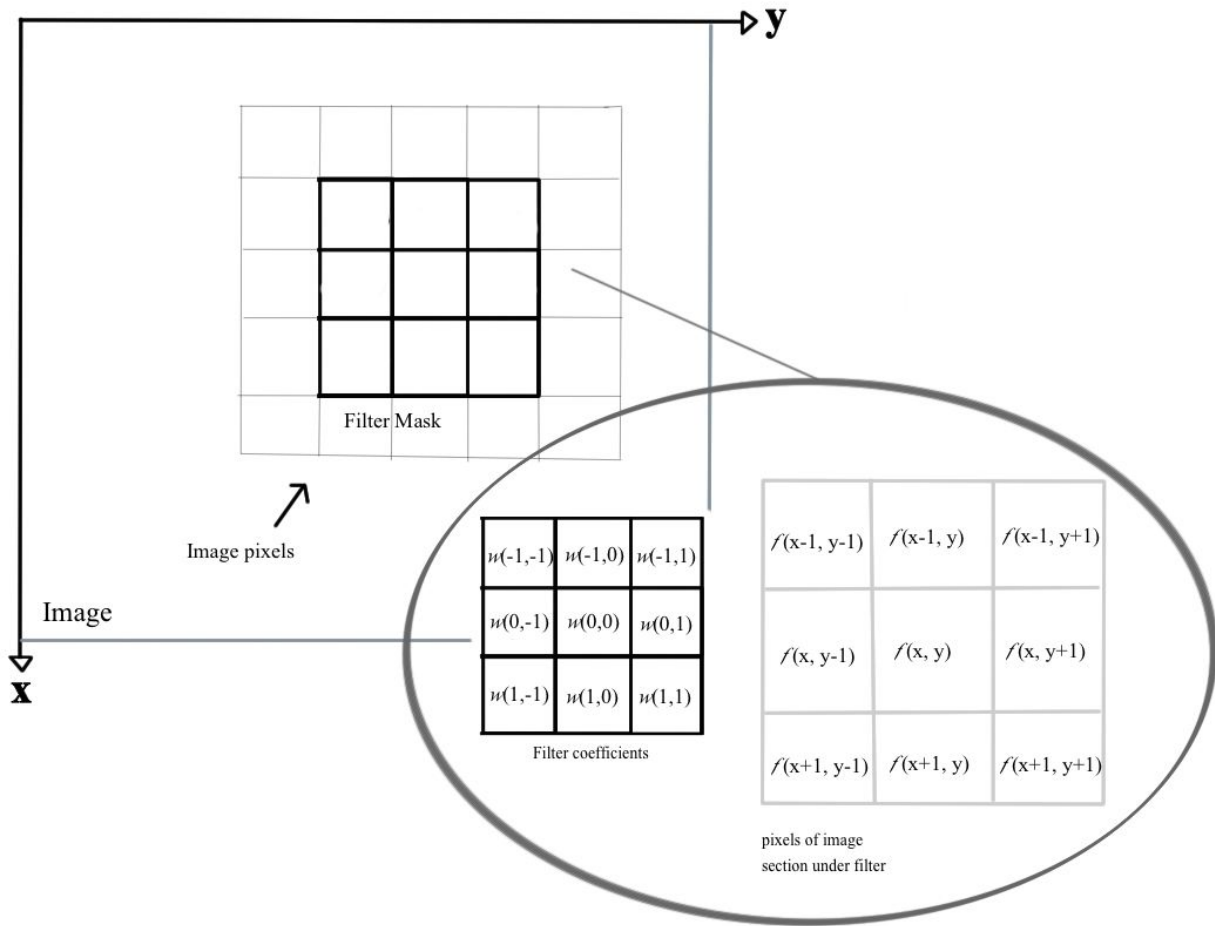
$$g(x, y) = T[f(x, y)] \quad (13)$$

where  $g(x, y)$  expresses the processed output image and  $f(x, y)$  is input image and  $T$  operates on  $f$ . To define Spatial neighborhoods for  $(x, y)$  point, square or rectangular region centered at  $(x, y)$  could be used. Center of region starts from origin and moves from one pixel to another to include different neighbors (Gonzales et al., 2004).

#### **2.4.1 Linear spatial filtering**

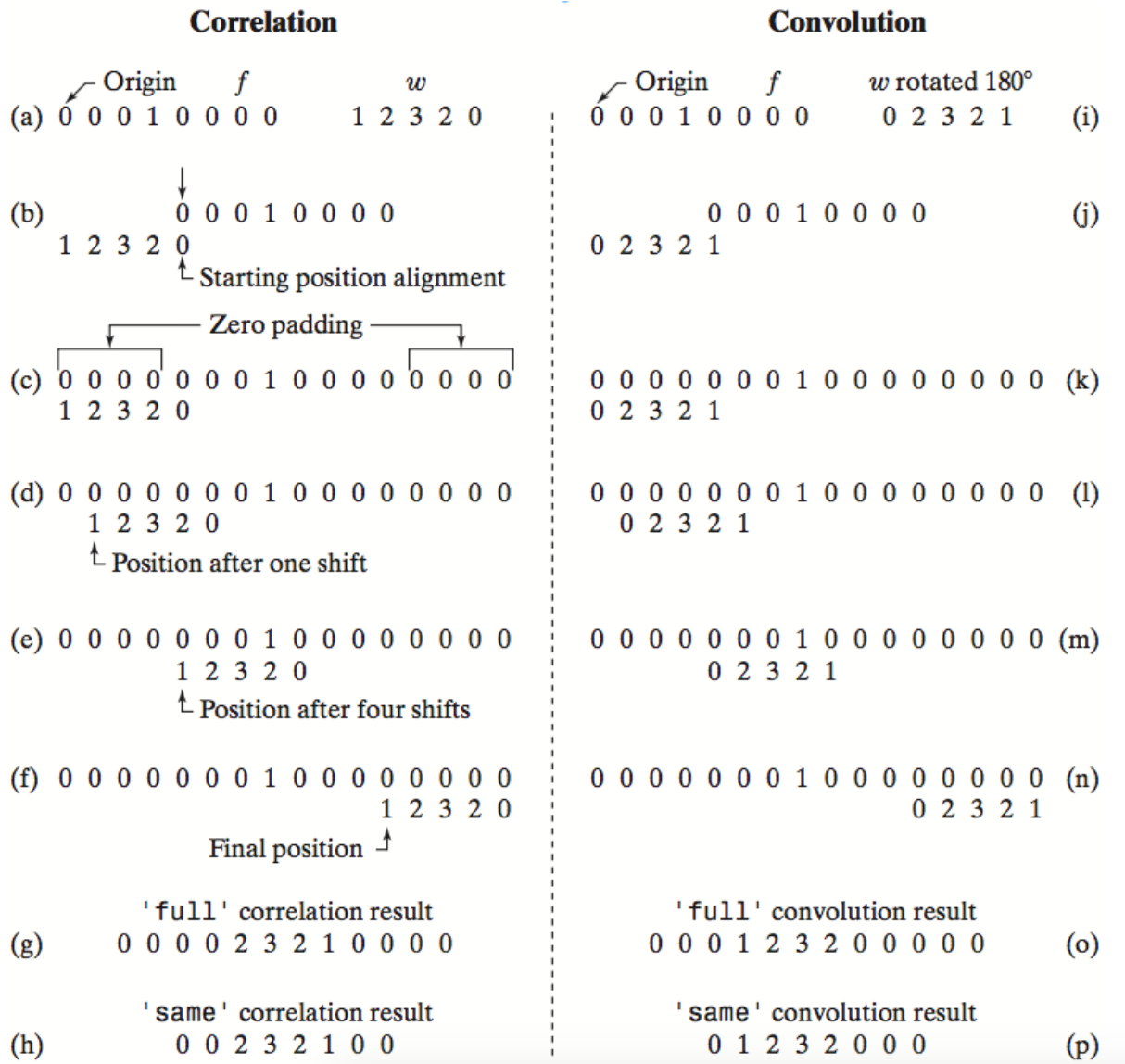
To have results at every pixel  $(x, y)$ , neighborhoods are multiplied with coefficients and summarized for the response at  $(x, y)$ . If size of neighborhood is  $m \times n$ ,  $mn$  coefficients should be given. The coefficients are ordered in a matrix called mask, filter, template, kernel, window etc.

The design and construction is shown in Figure 1.14. In the process of linear spatial filtering, center of mask is designed to move from point to point in an image. Filter mask moves through points and output of neighbors and their filter coefficients are carried out. In filter design, in term of mask size,  $m \times n$  is dimension of the mask, it is assumed that  $m = 2a + 1$  and  $n = 2b + 1$  as “ $a$ ” and “ $b$ ” are nonnegative integers, which refers to that mask should be in odd sizes as well functioning size begins with  $3 \times 3$ . The principle of having odd sized mask depends on having unique center point (Gonzales et al., 2004).



**Figure 1.14:** 3 x 3 mask and its related neighborhoods are magnified (Gonzales et al., 2004)

When performing linear spatial filtering, two concepts have to be understood. Correlation and convolution; Mechanic of 1 dimensional correlation and convolution is described in Figure 1.15. Let's call  $f$  image array and  $w$  is the mask. Mask is dragged to the image array  $f$  by the left side. In convolution, process is the same except mask is reversed from its origin.



**Figure 1.15:** One-dimensional correlation and convolution (Gonzales & Woods, 2008)

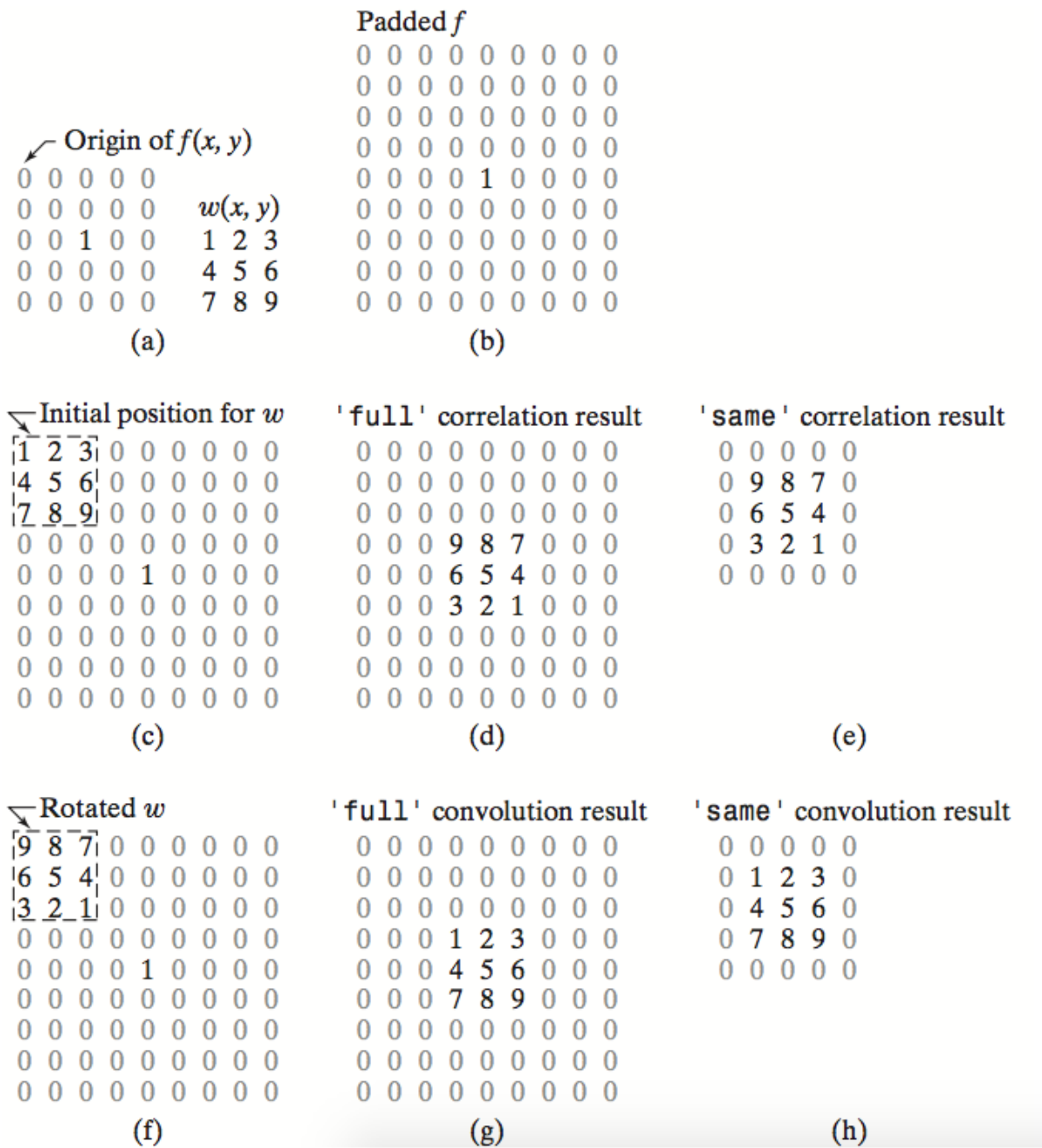
Figure 1.15 (a) illustrates that left corner of  $f$  is named as origin. To apply correlation right corner of mask  $w$  is dragged to the left side of image array  $f$  as shown in Figure 1.15 (b). When mask is dragged to the image  $f$ , there will be empty point that they don't overlap. To handle this problem in order to always have something corresponds to mask  $w$ , image  $f$  is padded with zeros as many as necessary as shown in Figure 1.15 (c). After one shift, mask and its corresponding image arrays multiplied and outcome of multiplications are summed. As

shown in Figure 1.15 (d), outcome is still zero. Finally mask meets with non-zero integer and its  $8(1)=2$ . Once mask continues to shift in same manner to the end [Figure 1.15 (f)], process will yield the result as shown in Figure 1.15 (g). The label ‘full’ indicates padded image array and ‘same’ indicates correlation same size with image  $f$  that is provided by toolbox [Figure 1.15 (h)].

In convolution same process was applied except the mask  $w$  was reversed from its origin and dragged to left corner of array  $f$ , as shown in Figure 1.15 (j). Dragging from origin of  $f$  to the end is shown from Figure 1.15 (k) to Figure 1.15 (n). Finally Figure 1.15 (o) and (p) shows ‘full’ and ‘same’ results provided by toolbox as discussed earlier with correlation.

Same principle can be applied to the images as shown in Figure 1.15. Top left corner of image  $f(x, y)$  is accepted as origin. To apply similar correlation to the image, bottom right corner of  $w(x, y)$  is dragged to the origin of  $f(x, y)$ , where they overlap as shown in Figure 1.16 (c). Zero padding is also used for reasons as it was discussed earlier for Figure 1.15. In correlation  $w(x, y)$  moves through every location and at least one of pixels should overlaps a pixel in the original image. Full correlation result is shown in Figure 1.16 (d) and same correlation result is shown in Figure 1.16 (e) as it is provided by toolbox.

Mechanics of convolution is almost same as correlation except that  $w(x, y)$  is rotated by  $180^\circ$ . In convolution any of two functions can undergoes translation and yet gives the same result. However, order matters in correlation. As result shows in Figure 1.16 (e) and (h), convolution is almost same as correlation except they are rotated by  $180^\circ$  to each other (Gonzales & Woods, 2008).



**Figure 1.16:** Two dimensional correlation and convolution (Gonzales & Woods, 2008)

### 2.4.2 Smoothing and low pass filters

Convolving with rectangular mask has where it takes average of neighborhood pixels has smoothing effect. Since it takes average of itself and its 8 neighborhoods and replace the pixel value with it, it will work to reduce noise and as low pass filter.

Figure 1.17 illustrates a rectangular mask that has smoothing effect. Second form is also given for sake of simplicity (Maintz, 2005).

Mathematical formula for  $N \times N$  with  $N$  being an odd number is denoted as:

$$g(x, y) = \begin{cases} \frac{1}{N^2}, & \text{if } x, y \in \{-(N-1)/2, \dots, (N-1)/2\} \\ 0, & \text{otherwise} \end{cases} \quad (14)$$

It is desired for  $N$  to be odd instead of even. Since only odd numbered masks are symmetric around center, odd sized masks are preferred over even sized masks.

To have isotropy, mask should be circular. In way of using square mask, image only can be approximated, if mask is smaller, worse outcome will be (Maintz, 2005).

$$\begin{array}{|c|c|c|} \hline \frac{1}{9} & \frac{1}{9} & \frac{1}{9} \\ \hline \frac{1}{9} & \frac{1}{9} & \frac{1}{9} \\ \hline \frac{1}{9} & \frac{1}{9} & \frac{1}{9} \\ \hline \end{array} = \frac{1}{9} \begin{array}{|c|c|c|} \hline 1 & 1 & 1 \\ \hline 1 & 1 & 1 \\ \hline 1 & 1 & 1 \\ \hline \end{array}$$

**Figure 1.17:** Smoothing mask (Maintz, 2005)



### 2.4.3 Sharpening and high pass filter

Figure 1.18 illustrates one example of detail enhancing mask.

-1	-1	-1
-1	9	-1
-1	-1	-1

**Figure 1.18:** Detail enhancing mask (Maintz, 2005)

To explain functionality of the filter, we can look at it and say all factors add up to 1. So it could be said, this mask will not do anything if part of image has constant values. But if there are differences between values, contrast range will be widened (Maintz, 2005).

One example of detail enhancing mask application is illustrated in Figure 1.19.

After application of detail enhancing mask, it is clear now to understand that, constant areas stayed constant and contrast between where 0's and 5's meet is greatly increased.

Detail enhancing mask can be confused with high pass filter. But in the detail-enhancing filter, constant areas stay without any change. This explains that if values are constant in some, they will stay same, even the lowest frequency (Maintz, 2005).

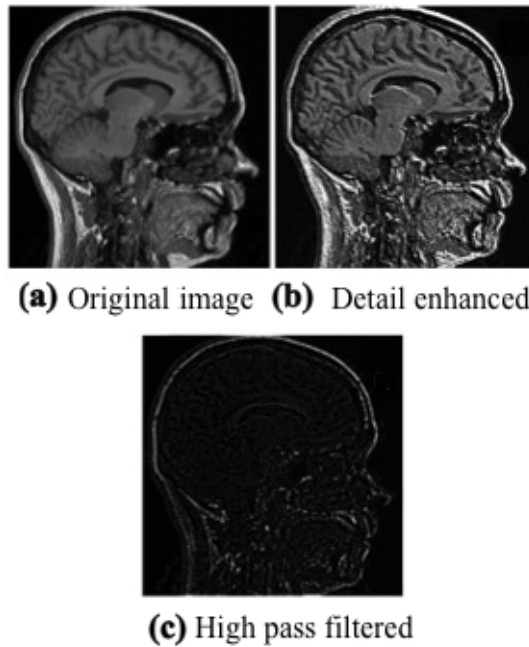
5	5	5	5	5	5	5							
5	5	5	5	5	5	5			5	5	10	15	20
5	5	5	5	0	0	0			5	10	20	-20	-15
5	5	5	0	0	0	0	→		20	25	-15	-5	0
0	0	0	0	0	0	0			-15	-10	-5	0	0
0	0	0	0	0	0	0			0	0	0	0	0
0	0	0	0	0	0	0							

**Figure 1.19:** Application of detail enhancing mask (Maintz, 2005)

$$\frac{1}{9} \cdot \begin{array}{|c|c|c|} \hline -1 & -1 & -1 \\ \hline -1 & 8 & -1 \\ \hline -1 & -1 & -1 \\ \hline \end{array} \cdot$$

**Figure 1.20:** Example of high pass filter (Maintz, 2005)

As illustrated in Figure 1.20, high pass filter looks lot like detail enhancing filter. The difference between them could be seen in Fig 1.21 where as sum of products in high pass filter is 0, which means constant pixel values will be valued as 0.



**Figure 1.21:** Detail enhanced image and high pass filtered image (Maintz, 2005)

#### 2.4.4 Unsharp masking

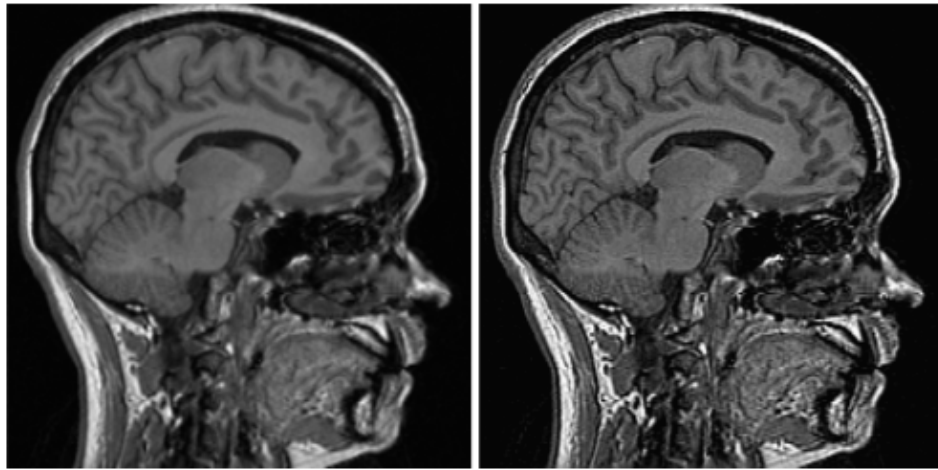
Enhancing of high frequency details as it can be called image sharpening is commonly done by printing process which requires adding original image to its high pass filtered image proportion or subtracting low pass filtered image of original image from original image (Maintz, 2005).

$$1. f_{e1}(x, y) = f(x, y) + \lambda_1 f_h(x, y) \quad (15)$$

$$2. f_{e2}(x, y) = f(x, y) - \lambda_2 f_l(x, y), \quad (16)$$

where  $f$  is denoted as original image and  $x, y$  defining coordinates,  $\lambda_1$  and  $\lambda_2$  represents positive constants,  $f_h$  is corresponding to high pass filtered image,  $f_l$  is a corresponding to pass filtered image, and at final  $f_{e1}$  and  $f_{e2}$  are the enhanced images from the processes. This method is named as unsharp masking (Maintz, 2005).

Figure 1.22 illustrates the first technique having  $\lambda_1 = 1/4$  and with  $3 \times 3$  high pass mask.



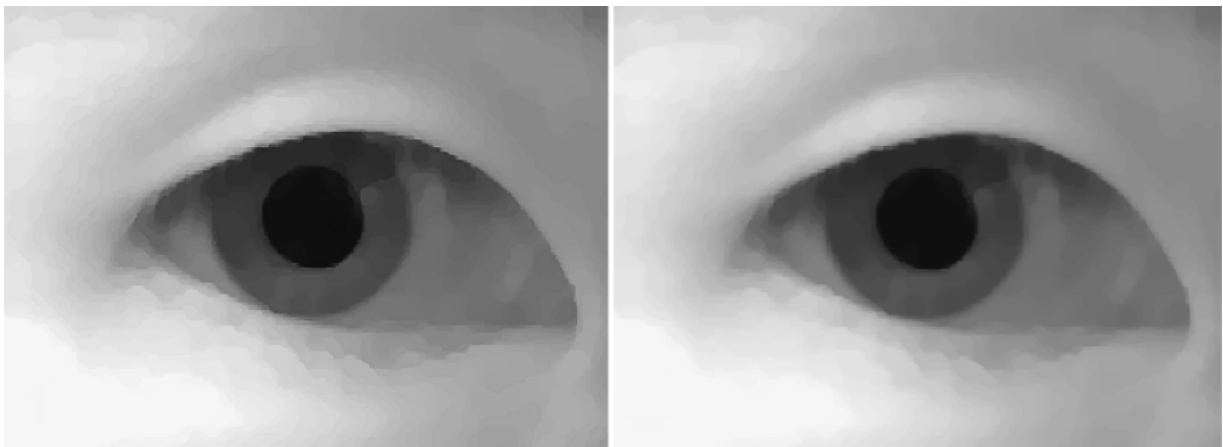
**Figure 1.22:** Output of unsharp masking on right (Maintz, 2005)

Example of sharpening of iris image is illustrated in Figure 1.22. Even though change without sharpening might seem like little difference, it is going to make difference in image segmentation.

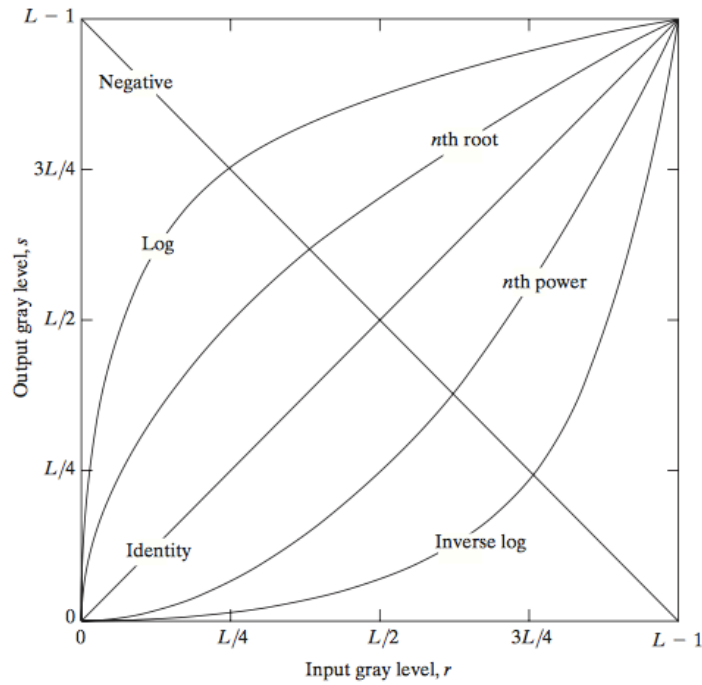
#### **2.4.5 Thesis application of sharpening image**

Image was sharpened in order to increase success rate of thresholding at next stage, which makes contrast switches differ and therefore increase the thresholding success. Figure 1.23 illustrates that pupil has sharper edges than the image without sharpening applied.

Image pre-processing of pupil is completed till here. Methods for iris preprocessing continue.



**Figure 1.23:** Sharpened image (left) and image before sharpening (right)



**Figure 1.24:** Type of basic transformations (Gonzales & Woods, 2008)

## 2.5 Image Enhancement in Spatial Domain

Expression of transformation can be understood in its simplest way, as neighborhood is 1x1 single pixel. In that case  $T$  is gray-level or intensity transformation, which is expressed as

$$s = T(r) \quad (17)$$

for  $(x, y)$  points,  $r$  denotes intensity of  $f$  and  $s$  as output of function and namely intensity of  $g$ . Transformations can be generalized under 3 functions that are broadly used in image enhancement (Gonzales & Woods, 2008),

- Linear,
- Logarithmic,
- Power-Law.

Figure 1.25 illustrates basic transformations.



**Figure 1.25:** Inverted pupil size for closing and detection of circles in dark polarity (right)

### 2.5.1 Image negatives

Negative transformation of gray level in range of  $[0, L-1]$  can be expressed as

$$S=L-1-r \quad (18)$$

This type of transformation is usually used to enhance white and gray details, which are embedded in dominant black area (Gonzales & Woods, 2008).

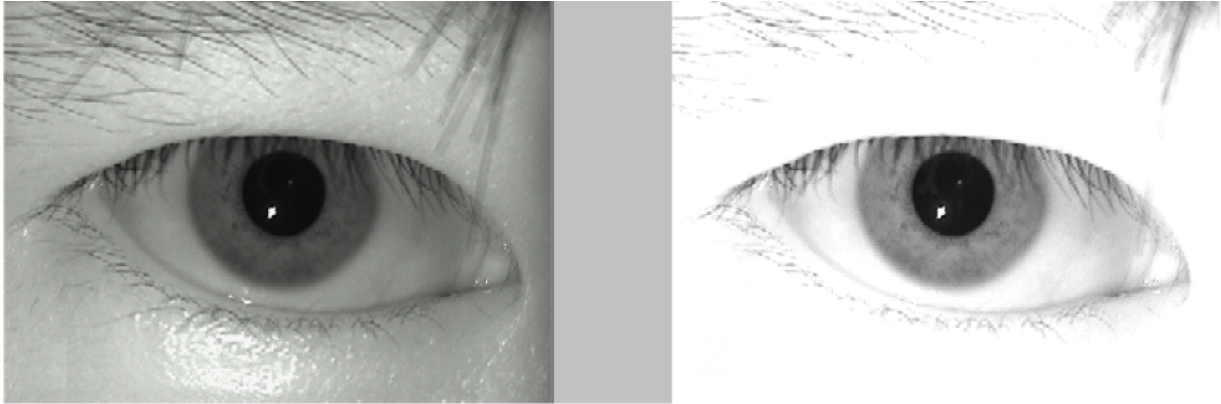
In our thesis, image negatives are taken to make image suitable for series of openings and closings and, for pupil detection as shown in Figure 1.25.

### 2.5.2 Log transformations

Curve of Log transformation is illustrated in Figure 1.24. As it can be understood from the curve, log transformation compress bright values and expands dark values in the input image. Inverse Log transformation would do the opposite, compressing dark ones and expanding bright ones (Gonzales & Woods, 2008). Mathematical expression of Log transformation is

$$s=c*\log(l+r) \quad (19)$$

where  $r$  is input pixel as  $r \geq 0$  and  $c$  is a constant. There are more versatile methods than Log transformation but it can compress high dynamic range of pixels. This is reason for us to use this method in iris images. In our application to segmentate iris, it compresses all the bright pixel variations so iris, which is second darkest point in the iris images, can be segmentated (Gonzales & Woods, 2008). Figure 1.26. Shows Log transformation on the iris image.



**Figure 1.26:** Image before log transformation (left) and image after log transformation (right)

### 2.5.3 Thesis application of log transformation

In order to differentiate iris from outer environment, log transformation expand dark pixels and compress light pixels, therefor light area looks all same where iris differentiate, which helps us to segmentate iris in next stage. Value of ‘C’ was set to 2 by trial and error method. Preference of log transformation arises from that it strongly compress light pixels compared to any other techniques such as power-law transformation and histogram equalization.

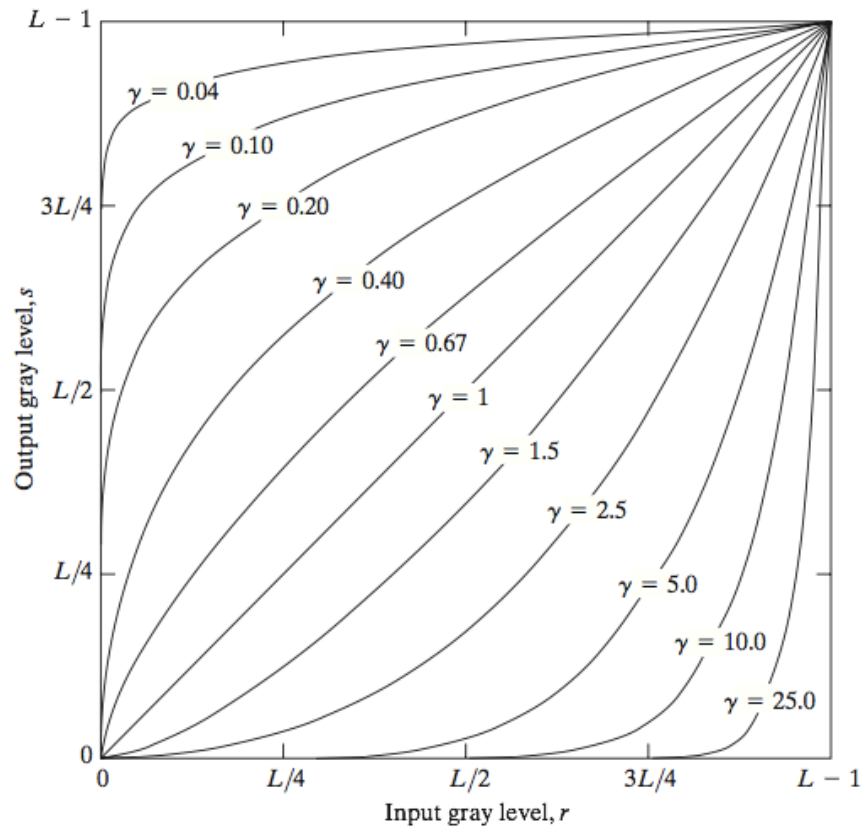
Preprocessing part on iris images are completed till here, segmentation is next step.

### 2.5.4 Power-law transformation

This image enhancement function can be also called Gamma Correction. Curves of  $n$ th power and  $n$ th root are illustrated in Figure 1.24 Mathematical formulation of Power-law Transformation is expressed as

$$s = cr^{\gamma} \quad (20)$$

where  $s$  is output,  $c$  is a constant and,  $\gamma$  is also a constant which determines degree of expansion and compression. If  $\gamma > 0$ , it expands dark pixels while compressing light ones if  $\gamma < 0$ , it expands bright pixels while compressing dark ones. The important difference between Power-Law transformation and Log transformation is that, different possible curves can be generated by varying  $\gamma$ , as it is illustrated in Figure 1.27 (Gonzales & Woods, 2008).



**Figure 1.27:** Different transformation curves (Gonzales & Woods, 2008)



## 2.6 Lie Detection on Pupil Size: Image Segmentation Phase

In this phase, segmentation methods are explained and at the end, segmentation methods applied in thesis is explained and illustrated.

### 2.6.1 Introduction to image segmentation

In image processing, methods are applied to input image to get an output, outputs are attributes that are extracted from input images, segmentation is higher level of it in same direction. Segmentation subdivides objects in image and subdivision should stop when required region is extracted. For segmentation of monochrome images, methods are dependent on similarity and discontinuity. Segmentation accuracy is very important parameter; degree of achievement determines success of computerized analyses (Gonzales et al., 2004).

### 2.7 Point, Line, and, Edge Detection

Techniques are discussed on this section depend on three basic discontinuities. Mask is used to look for discontinuities through image. Product of 3 x 3 mask coefficients with their corresponding neighbors are summed as they pass through intensity levels in the image (Gonzales et al., 2004). The response of mask, R at any point of image is expressed as

$$R = w_1z_1 + w_2z_2 + \dots + w_9z_9 \quad (21)$$

$$= \sum_{i=1}^9 w_i z_i \quad (22)$$

where  $w_i$  is corresponding mask coefficient to intensity level of pixel  $w_i$ .

$$|R| \geq T$$

-1	-1	-1
-1	8	-1
-1	-1	-1

**Figure 1.28:** A mask of point detection (Gonzales et al., 2004)

### 2.7.1 Point detection

Detection of points that are isolated in constant or nearly constant areas employs mask as shown in Figure 1.28 isolated points are detected at the location where the mask is centered if

as  $T$  is nonnegative threshold value.

### 2.7.2 Line detection

In case of line detection, 4 types of masks are illustrated in Figure 1.29 with different angle of lines to be detected. First mask would respond to horizontal single pixel lines, second one is oriented to respond  $+45^\circ$  lines, third one for vertical lines and last one for  $-45^\circ$  lines.

-1	-1	-1	-1	-1	2	-1	2	-1	2	-1	-1	-1
2	2	2	-1	2	-1	-1	2	-1	-1	2	-1	-1
-1	-1	-1	2	-1	-1	-1	2	-1	-1	-1	-1	2
Horizontal			$+45^\circ$			Vertical			$-45^\circ$			

**Figure 1.29:** Type of line detectors (Gonzales et al., 2004)

### 2.7.3 Edge detection

Edge detection is most broadly used technique for discontinuities in intensity values. These values could be detected by first- and second order derivatives. In way of using first order derivative is the gradient in image processing. The gradient of 2-D function,  $f(x, y)$  is expressed as vector where its magnitude can be calculated (Gonzales et al., 2004).

$$\nabla f = \begin{bmatrix} G_x \\ G_y \end{bmatrix} = \begin{bmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \end{bmatrix} \quad (23)$$

Calculation of magnitude of the vector is written below:

$$\nabla = \text{mag}(\nabla f) = [G_x^2 + G_y^2]^{1/2} \quad (24)$$

$$= [(\partial f / \partial x)^2 + (\partial f / \partial y)^2]^{1/2} \quad (25)$$

In order to simplify equation, excluding square root is applied to perform approximation,

$$\nabla f \approx G_x^2 + G_y^2 \quad (26)$$

or absolute values are used

$$\nabla f \approx |G_x| + |G_y| \quad (27)$$

Approximated values still works as derivatives, in constant intensity areas, value they have is zero and value of these approximations are dependent on degree of intensity shifts, magnitude of gradient including approximations can be simple called “the gradient”.

The angle of gradient vector points to maximum degree of change of  $f$  on its coordinates  $(x, y)$  (Gonzales et al., 2004).

$$A(x, y) = \tan^{-1} \left( \frac{G_y}{G_x} \right) \quad (28)$$

At matter of second derivatives, laplacian is used for computation. Mathematical expression of laplacian that is second-order derivatives of 2-D function  $f(x, y)$  is expressed

$$\nabla^2 f(x, y) = (\partial^2 f(x, y) / \partial x^2) + (\partial^2 f(x, y) / \partial y^2) \quad (29)$$

Second order derivatives are rarely used because they also double the noise along the edges, which makes hard to detect edges. But it can be used with other edge detection techniques as complementary (Gonzales et al., 2004).

The main idea behind this process is about:

1. Using first derivatives to find magnitude of intensity changes beyond the threshold,
2. Using second derivative to locate zero crossings in area.

### **2.7.3.1 Type of edge detection techniques**

There are 5 commonly known detector types and they will be discussed under this topic.

#### **2.7.3.1.1 Sobel edge detector**

In Sobel edge detector, to approximate first derivative digitally,  $G_x$  and  $G_y$ , masks are used as shown in Figure 1.30, as gradient is at center point, and neighborhood pixels are computed by mask coefficients. If  $g \geq T$ , then pixel  $(x, y)$  is said be an edge pixel (Gonzales et al., 2004).

$$g = [G_x^2 + G_y^2]^{1/2} \quad (30)$$

$$= \{[(z_7 + 2z_8 + z_9) - (z_1 + 2z_2 + z_9) - (z_1 + 2z_2 + z_3)]^2 + [(z_3 + 2z_6 + z_9) - (z_1 + 2z_4 + z_7)]^2\}^{1/2}$$

First mask and second mask is applied to the image. Squaring value of mask's results and adding two results together and having square root of the sum.

#### **2.7.3.1.2 Prewitt edge detector**

Produces slightly noisier images than Sobel edge detector, mask of Prewitt edge detector is illustrated in Figure 1.30.

#### **2.7.3.1.3 Roberts edge detector**

Its one of oldest edge detectors and rarely used because of its structure that its not symmetric, therefor cant detect edges which are multiplies of  $45^\circ$  (Gonzales et al., 2004). Mask of Roberts edge detector and its operation is illustrated in Figure 1.30.

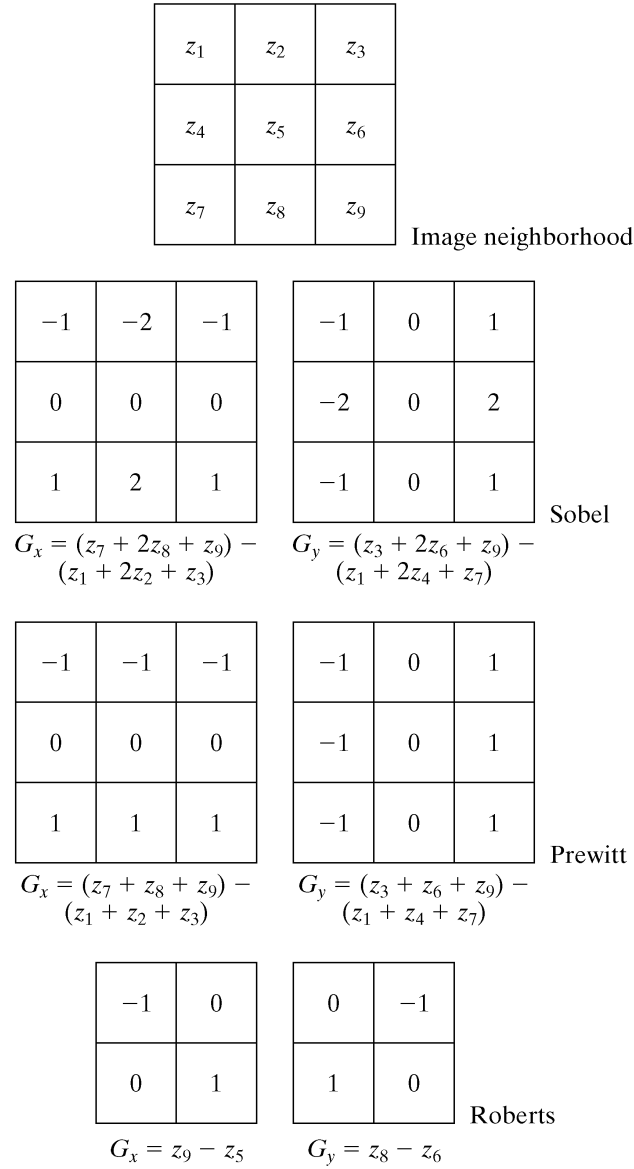
#### **2.7.3.1.4 Laplacian of Gaussian detector**

Gaussian function is smoothing function if convolved with image and standard deviation determines the degree of blurring. As it is understood from the name, image is first blurred and the Laplacian is computed which produces double-edge results. Edges will be located and then zero crossings between double edges will be found. If values are less than threshold, they will be ignored, user can determine threshold or MATLAB function will automatically choose it (Gonzales et al., 2004).

$$\nabla^2 h(r) = - \left[ \frac{r^2 - \sigma^2}{\sigma^2} \right] e^{-\frac{r^2}{2\sigma^2}} \quad (31)$$

#### **2.7.3.1.5 Zero crossing detector**

Idea of this detector works same with LoG method except its carried out with different convolution function.



**Figure 1.30:** Edge detector types; Sobel, Prewitt, Roberts (Gonzales et al., 2004)

#### 2.7.3.1.6 Canny edge detector

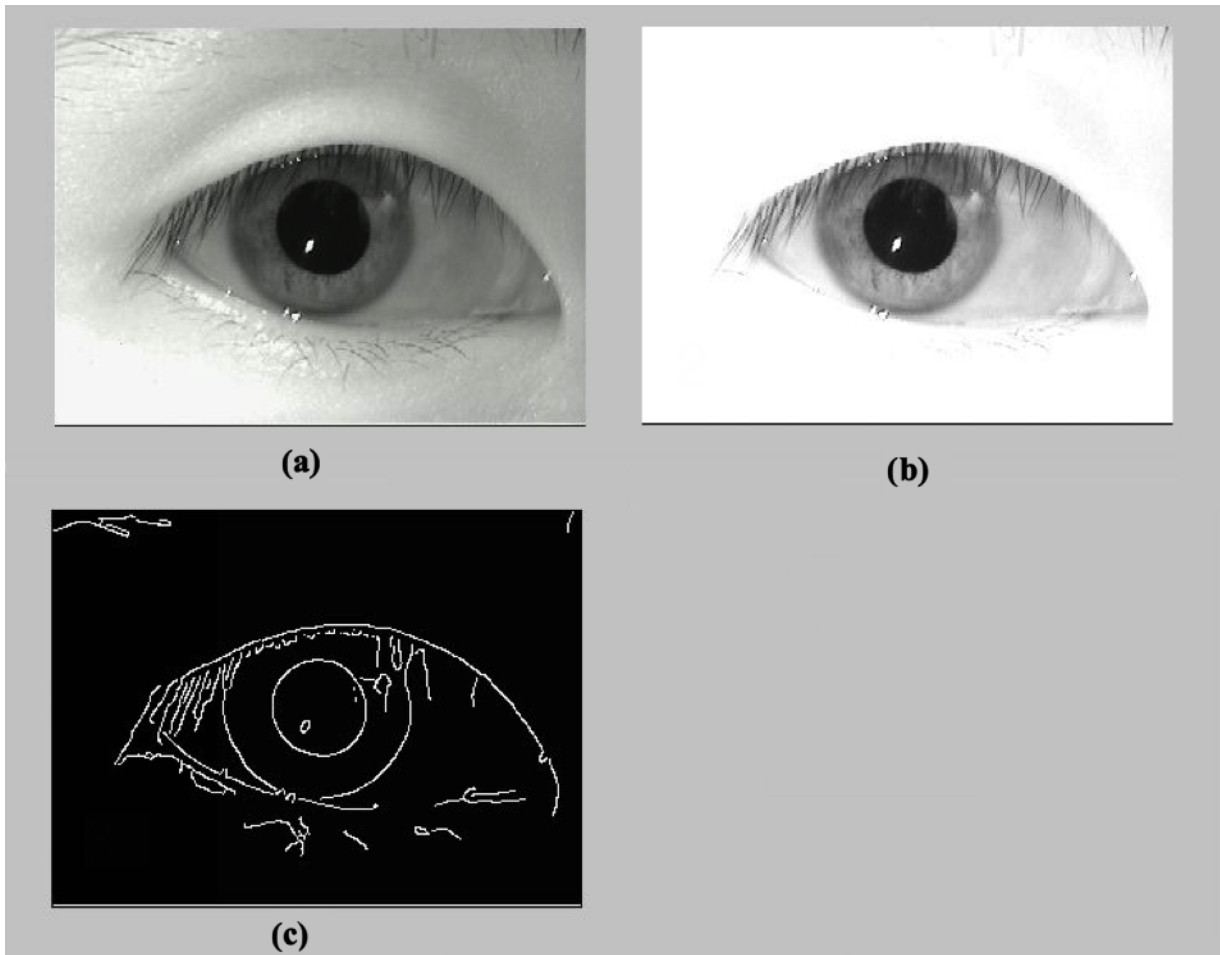
Among the edge detectors we have discussed, Canny edge detector is most powerful one. The method is also used in our thesis. Process of edge detection is explained as follows (Gonzales et al., 2004):

1. Noise reduction is applied via Gaussian filter with specified standard deviation,

2. For computation of  $G_x$  and  $G_y$ , any techniques in Fig. 30 can be used. Every point in the image computed for local gradient and edge direction. Points, which are locally maximum in the direction of gradient, defined as edges,
3. Ridges are created by gradient magnitude depending on edge points determined in 2. Pixels, which are not on the top of the ridges, are set to zero as top of the ridges are tracked by algorithm and produce thin line in the output as it is called non-maximal suppression. There are 2 thresholds,  $T_1$  and  $T_2$ . Edges, which are greater than  $T_2$ , are called strong and edges, which are between  $T_1$  and  $T_2$ , are called weak edge pixels,
4. Finally edge linking is performed by algorithm to include weak edges 8-connected with strong ones.

#### **2.7.3.1.7 Application of canny edge detector in thesis**

After preprocessing part for iris detection is completed, next step is edge detection. All of these edge detectors were tested on Iris images, and Canny edge detector gave best and simple result. It detects less but Canny Edge Detection provided strong features therefor less noise and strong edges. Figure 1.31 illustrates application of canny edge detector in thesis.



**Figure 1.31:** RGB to gray conversion (a), log transformation (b), canny edge detection (c)

## 2.8 Thresholding

In image segmentation, thresholding holds very important place due to its simplistic approach. Let's think about a light object with a dark background; its intensity histogram separates dark and light pixels. To extract the light object from its background, a threshold can be chosen (Gonzales et al., 2004). For an image, any point  $(x, y)$  will be object for  $f(x, y) \geq T$ . Mathematical formulation of thresholded image is expressed as

$$g(x, y) = \begin{cases} 1, & \text{if } f(x, y) \geq T \\ 0, & \text{if } f(x, y) < T \end{cases} \quad (32)$$



### 2.8.1 Global thresholding

Threshold can be chosen depending on image histogram analysis. That threshold will separate 2 different modes. Another way to choose threshold can be done by trial and method, where user can test threshold values till achieving the best result.

To choose best threshold value in automatic manner, Gonzales and Woods [2002] describe their method:

1. T is selected by estimation, (Suggestion is between maximum and minimum intensity values.)
2. Segmentation will be done by using T, which will produce 2 different set of pixels as the ones are above the threshold  $G_1$  and below the threshold  $G_2$ ,
3. Average intensity values  $\mu_1$  and  $\mu_2$  are computed for  $G_1$  and  $G_2$  regions,
4. New Threshold value is computed,

$$T = 1/2(\mu_1 + \mu_2)$$

5. Step 2 and 4 are iterated till the difference in iterations is smaller than predefined Threshold.

#### 2.8.1.1 Otsu's method

It is a histogram-based method. In examination of this method, normalized histogram is handled as discrete probability function, as

$$P_r(r_q) = n_q / n \quad q = 0, 1, 2, \dots, L - 1 \quad (33)$$

total number of pixels are symbolized by  $n$ ,  $n_q$  as number of pixels that have intensity  $r_q$ , and total number of intensity levels are symbolized by  $L$ . If  $k$  is a threshold value, there will be 2 set of pixels, as  $C_0$  is  $[0, 1, \dots, k - 1]$  and  $C_2$  is  $[k, k + 1, \dots, L - 1]$ . Threshold value  $k$ , which is determined by Otsu's method, maximizes the between-class variance (Otsu, 1979). Between-class variance  $\sigma^2_B$  can be expressed as

$$\sigma^2_B = w_0(\mu_0 - \mu_T)^2 + w_1(\mu_1 - \mu_T)^2 \quad (34)$$

$$w_0 = \sum_{q=0}^{k-1} p_q(r_q) \quad (35)$$

$$w_1 = \sum_{q=k}^{L-1} p_q(r_q) \quad (36)$$

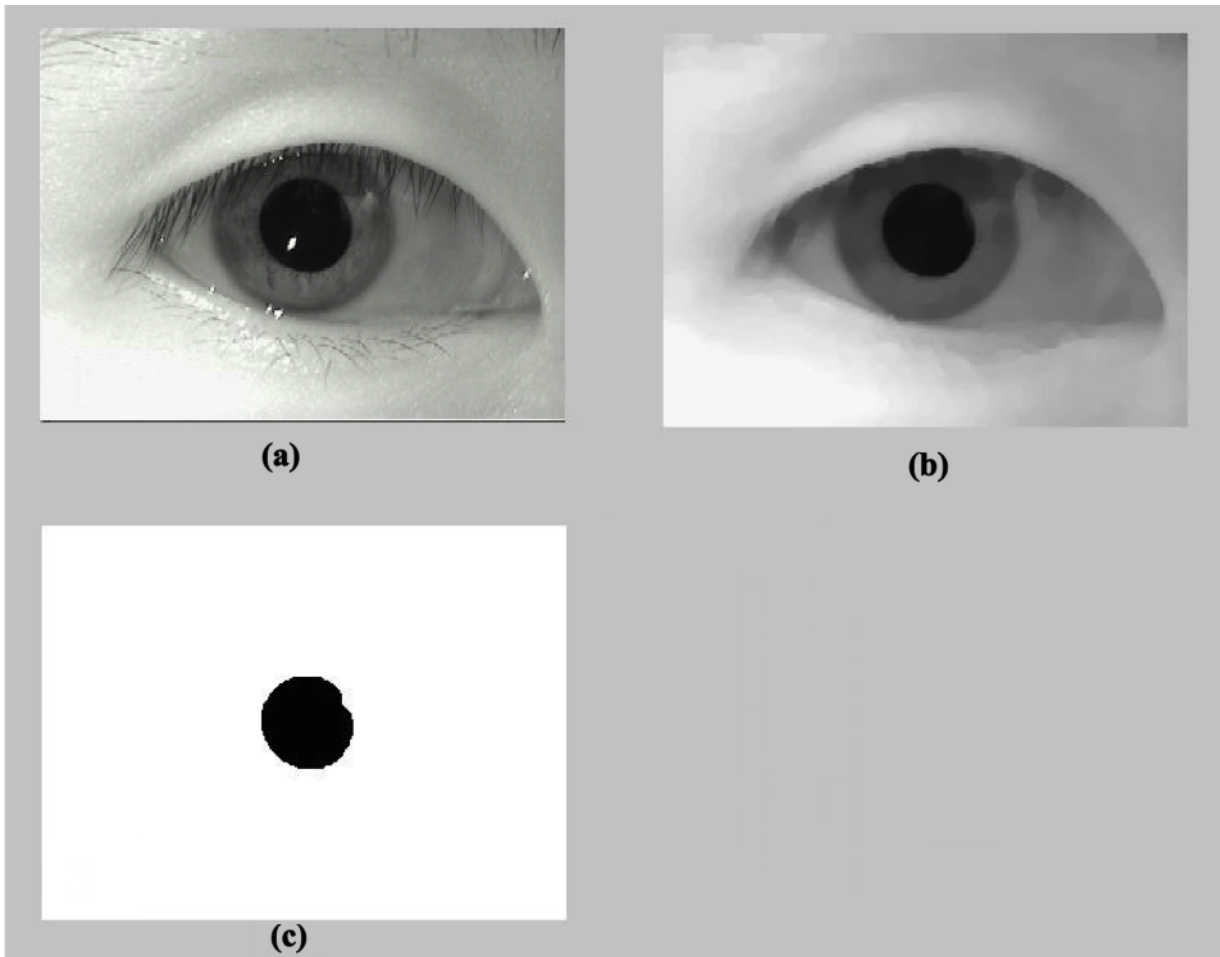
$$\mu_0 = \sum_{q=0}^{k-1} p_q(r_q) / w_0 \quad (37)$$

$$\mu_1 = \sum_{q=k}^{L-1} p_q(r_q) / w_1 \quad (38)$$

$$\mu_T = \sum_{q=0}^{L-1} qp_q(r_q) \quad (39)$$

### 2.8.1.2 Application of Otsu's method in thesis

After image-preprocessing part is completed in, next step is towards to thresholding, where pupil is completely in different gray level than its environment and thresholding is now easier to apply. Application of thresholding to the image depends on the property of pupil to be darkest point in the entire image, so that applying global thresholding as it only segmentates pixels above the threshold, which helps to only segmentate pupil and resulting with less noise and faster process time compared to local thresholding. Figure 1.32 illustrates thresholding of pupil. Thresholding level was set as 0.15 by trial and error method.



**Figure 1.32:** RGB to grey conversion (a), opening and closing (b) and Otsu's method (c)

After some trials, thresholding value was chosen 0.15. To prevent any error, image closing is applied right after thresholding.

### 2.8.2 Local thresholding

Global thresholding methods are not so reliable when background illumination is uneven. Solution to this problem is to compensate uneven illumination with preprocessing and then global thresholding can be applied. Morphological top-hat operation is applied and then Otsu's method is used to achieve segmentation. The Process can be presented as equivalent of thresholding  $f(x, y)$  that locally varies with threshold function  $T(x, y)$  (Gonzales et al., 2004) :

$$g(x, y) = \begin{cases} 1, & \text{if } f(x, y) \geq T(x, y) \\ 0, & \text{if } f(x, y) < T(x, y) \end{cases} \quad (40)$$

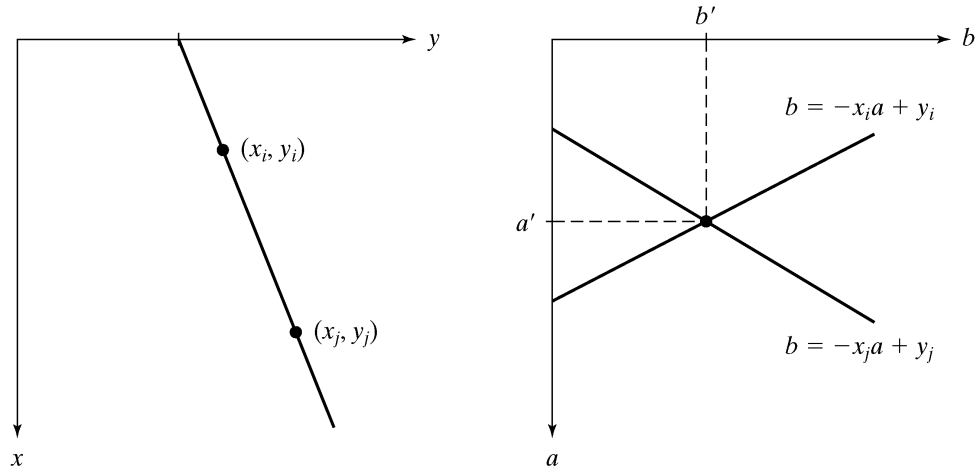
where

$$T(x, y) = f_0(x, y) + T_0 \quad (41)$$

Morphological opening of image  $f$  is denoted as  $f_0(x, y)$  and,  $T_0$  is result of Otsu's method as it is applied to  $f_0$  in order to determine constant thresholding value.

## 2.9 Line Detection Using Hough Transform

The methods for edge detection are discussed previously. In practical applications, detection of edges is rarely achieved because of noisy environment and uneven illumination and some other effects that cause same problem. Therefore, linking edge detection algorithm get help from linking procedures to complete successful edge detection. One of the approaches can be in direction of Hough Transform (Hough, 1962). In Hough Transform, a point  $(x_i, y_i)$  and all lines pass through a line considered. Number of lines that pass through a line, that slope-intercept equation can be used for all of these lines  $y_i = ax_i + b$ . The equation can be written as  $b = -x_i a + y_i$  and parameter space produces the equation of single line for fixed pair  $(x_i, y_i)$ . In addition, a second point  $(x_j, y_j)$  also fits to this equation, and it intersects the line associated with  $(x_i, y_i)$  at  $(a', b')$ . Slope is denoted as  $a'$  and,  $b'$  refers to intercept of the line in the  $xy$ -plane that contains  $(x_i, y_i)$  and  $(x_j, y_j)$ . It can be said that all points that are present in the line have lines in parameter space where they meet at  $(a', b')$ . Points in  $xy$  plane and parameter-space are illustrated in Figure 1.33.



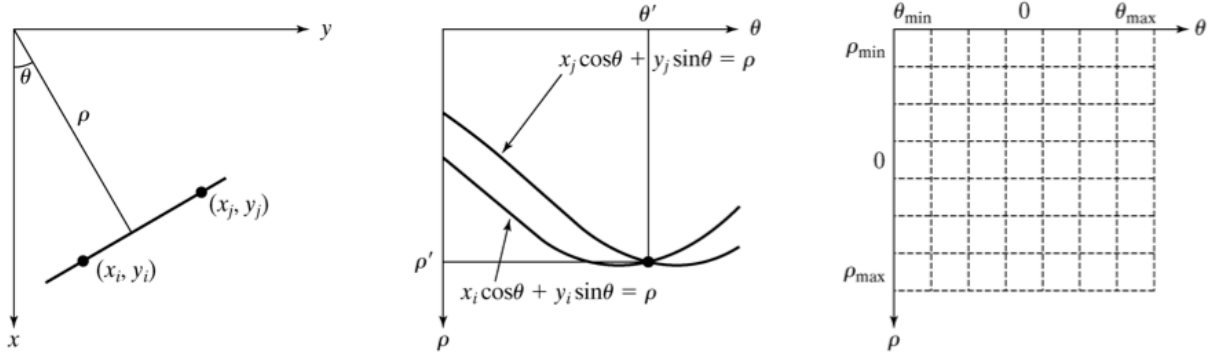
**Figure 1.33:**  $xy$ -plane (left) and parameter space (right) (Gonzales et al., 2004)

For all image points, parameter space lines could be plotted and lines are defined by intersection of large numbers of parameter space lines. However, problem arises from here that, as a line approaches the vertical direction, slope of line approaches to infinity. One solution is that using normal presentation of line.

$$x \cos \theta + y \sin \theta = \rho \quad (42)$$

Interpretation of  $\rho$  and  $\theta$  are illustrated in Figure 1.33. A horizontal line as  $\theta = 0^\circ$  and  $\rho$  is on  $x$ -intercept. Conceptually, a vertical line as  $\theta = 90^\circ$  and  $\rho$  is on  $y$ -intercept which is illustrated at Figure 1.33 (1). Figure (2) presents set of lines, which pass at certain point  $(x_i, y_i)$ . The point in Figure 1.33 (2) presents intersection of  $\rho$  and  $\theta$  which means that lines pass through it, pass through both  $(x_i, y_i)$  and  $(x_j, y_j)$ .  $\rho\theta$  parameters are subdivided into accumulator cells. Expected range of parameters are between  $(\rho_{\min}, \rho_{\max})$  and  $(\theta_{\min}, \theta_{\max})$ . Range of values are between  $-90^\circ \leq \theta \leq 90^\circ$  and  $-D \leq \rho \leq D$  as  $D$  presents distance from corner to corner in the image. Coordinates  $(i, j)$  with its accumulator value is presented at accumulator space by  $(\rho_i, \theta_j)$ . Every cell in the beginning is set to zero, then every point  $(x_k, y_k)$  for every  $\theta$ , that is divided in accumulation cells are used to find corresponding  $\rho$  by solving  $x_k \cos \theta + y_k \sin \theta = \rho$  and the  $\rho$  is rounded to closest  $\rho$  that exist in accumulator cells. Then its equivalent accumulator cell is incremented. This method, as it is finalized refer to value of  $\theta$  in

A  $(i, j)$  to  $\theta$  points in the  $xy$ -plane that is on the  $x \cos \theta_j + y \sin \theta_j = \rho_i$  (Figure 1.34). Accuracy of co-linearity is dependent of number of subdivisions.



**Figure 1.34:** Parameterization of lines (left),  $\rho\theta$  plane (middle) and accumulator cells (right)  
(Gonzales et al., 2004)

### 2.9.1 Hough transformation circle detection

This technique is used for feature extraction, which aims to extract class of shapes pertaining to one group.

The method is used, because of its ability that it can be efficient in noisy environment, occlusions, and uneven illumination. CHT is not very specified algorithm but it has 3 essential steps (Find Circles, n.d).

1. Accumulator array computation,
2. Center estimation,
3. Radius estimation.

Presentation of circle is expressed as

$$(x - c_1)^2 + (y - c_2)^2 = c_3^2 \quad (43)$$

where a and b are coordinates of center of radius and r denotes radius. Difference between line and circle detection is that, circle has 3 parameters which comes out as accumulator cells being in  $A(i, j, k)$  form in 3 dimension as cube like cells (Gonzales & Woods, 2008).

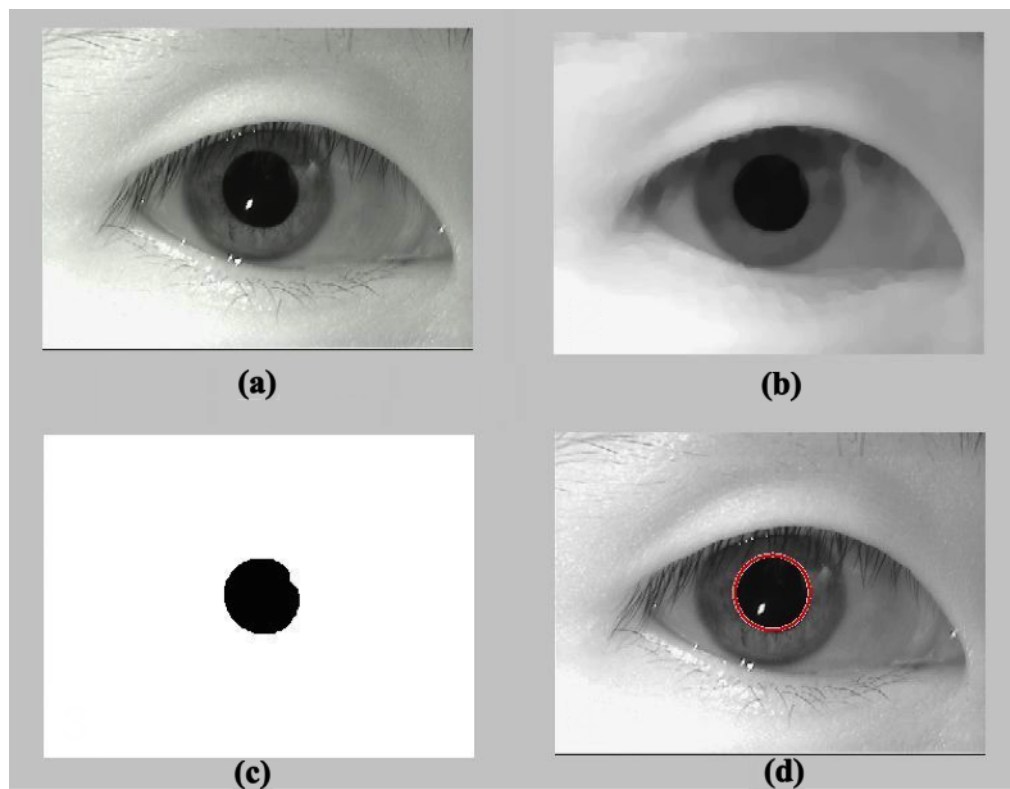
Then edge linking is considered as follows:

1. Binary image is obtained,
2. Subdivisions of  $\rho\theta$ -plane are specified,
3. Number of accumulator cells for high pixel concentrations are examined,
4. Relationship between chosen and other pixels (primarily continuity) are examined.

Gaps between disconnected pixels are computed corresponding to given accumulator cell. Gaps between pixels are connected depending on specified threshold (Gonzales & Woods, 2008).

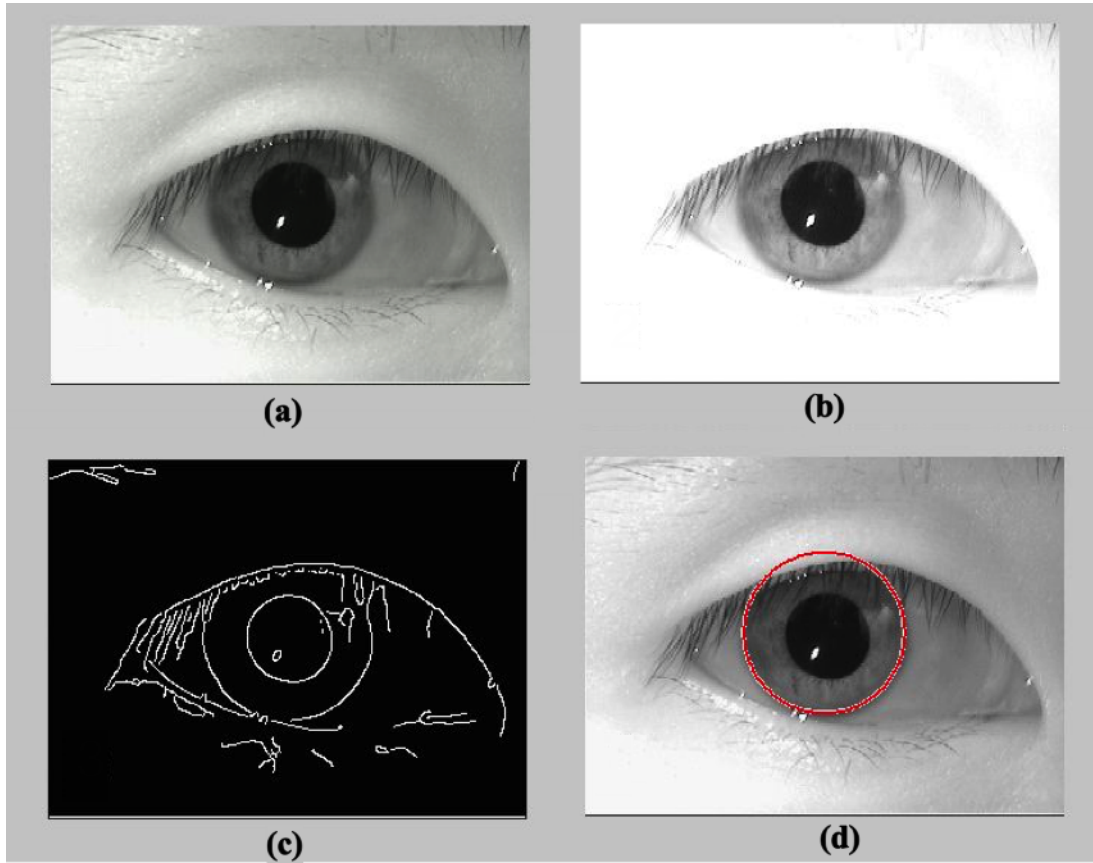
### 2.9.2 Application of circle detection by Hough transform in the thesis

Circle Detection is both applied to Iris Segmentation and Pupil Segmentation. IPT toolbox provides the code to achieve it, circle range parameters were arranged by trial and error method as 15 50 for pupil and 46 70 for iris along with 93 and 95 sensitivity respect to it. Figure 1.35 and Figure 1.36 illustrates pupil and iris segmentation by Hough transform respectively.



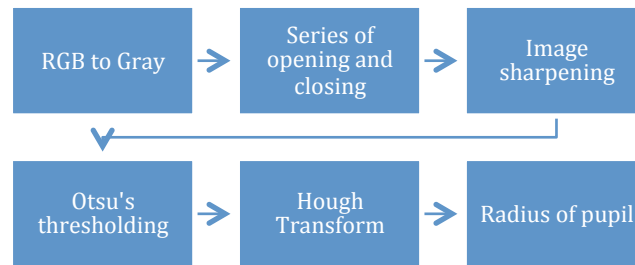
**Figure 1.35:** RGB to gray (a), opening and closing (b), thresholding (c), circle detection (d)



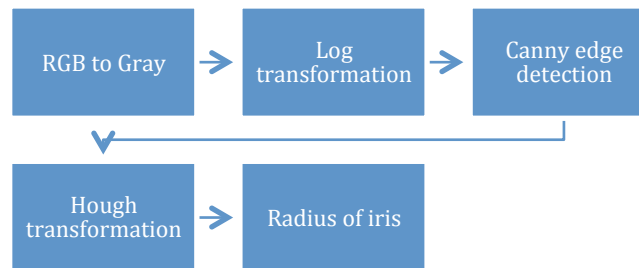


**Figure 1.36:** RGB to gray (a), log transformation (b), edge detection (c), Iris detection (d)

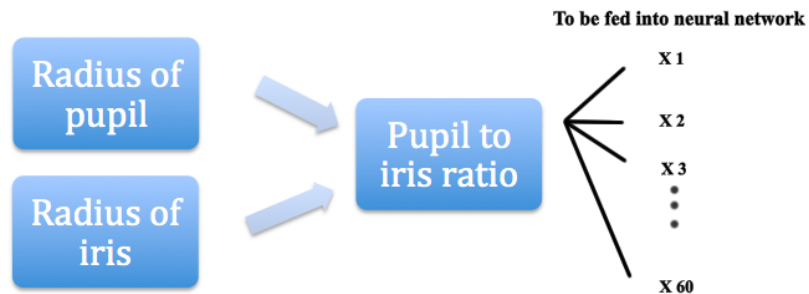
Before stepping into the neural network classification, all the process carried out till here is segmentation of pupil and iris and ratio of pupil to iris was taken as input to neural network. All steps for image preprocessing and segmentation for pupil is illustrated in Figure 1.37, and all steps for preprocessing and segmentation of iris is illustrated in Figure 1.38 and Figure 1.39 illustrated 60 samples of inputs to feed into the neural network.



**Figure 1.37:** Preprocessing and segmentation of pupil



**Figure 1.38:** Preprocessing and segmentation of iris



**Figure 1.39:** Pupil to iris ratio to be fed into the neural network

## CHAPTER 3

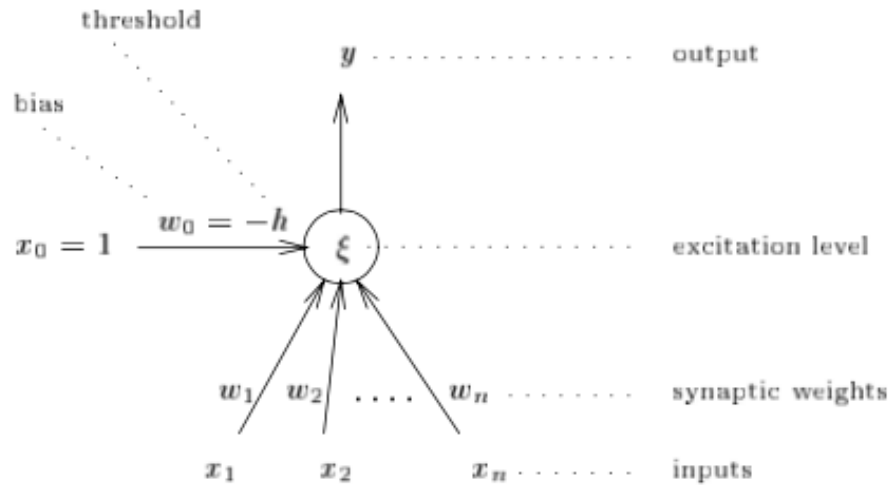
### CLASSIFICATION OF PUPIL SIZES

#### 3.1 Artificial Neural Network

Understanding neurophysiology made possible to create simplified mathematical models to solve practical tasks from artificial intelligence (Jain, Mao & Mohiuddi, 1996).

#### 3.2 Mathematical Model of Artificial Neural Network

Function if biological neuron is re-formulated to obtain a formal neuron, which will be mathematical basis for mathematical modeling of Artificial Neural Network (Sima, 1998).

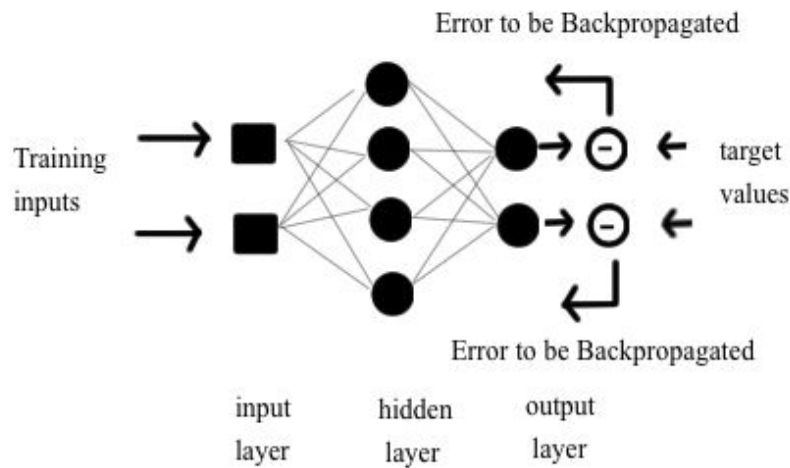


**Figure 2.1:** Formal Neuron (Sima, 1998)

Where  $x_1, \dots, x_n$  are inputs, which is the signal coming from dendrites in terms of biological neuron. Inputs have their synaptic weights labeled as  $w_1, \dots, w_n$ , which is about measurement of their permeability (Figure 2.1).

### 3.3 Backpropagation Neural Network

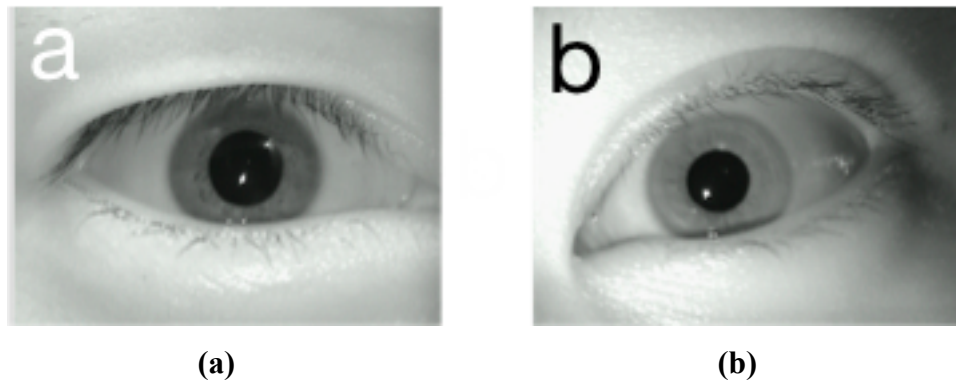
Feed-forward neural networks can be defined as computational graph, which is consisting of nodes. Nodes are computing units and their directed edges carry numerical information to another node. Nodes are computing units and numerical information is carried to another node by directed edges. Input of each node has its own primitive function and in big picture, input to a pattern is formed by computation of chain of function compositions, which is also called output vector. Input to output space is process of composite function that is called network. Success of network depends Network function  $\varphi$  that approximates the weights as close as possible, which is gained by examples. Backpropagation is a feed forward network which is consisted of input layer, hidden layer and output layer, and this method is very popular in supervised learning case (Werbros., 1974 & Rumelhart et al., 1986). The BP algorithm as it can be resembled from its name, output of network is compared with desired output and the error propagates backwards to adjust weights (Figure 2.2). In implementation of algorithm, hyperbolic tangent function is usually used.



**Figure 2.2:** Illustration of Backpropagation algorithm

### 3.4 System Database

Image processing material is taken from MMU Iris database (MMU Iris database, 2004). 60 samples were taken as 30 of them are dilated and 30 of them are non-dilated in 320x240 size. Dilated pupils are assumed as condition of lying (Figure 2.3). However, they are not acquired during such investigation, database is originally created for iris identification. In lack of database, we figured it out that, if we develop image processing techniques combined with classifier, these samples can be changed with real samples acquired during such investigation and it would differentiate dilated and non dilate pupil size as it does on assumed samples.

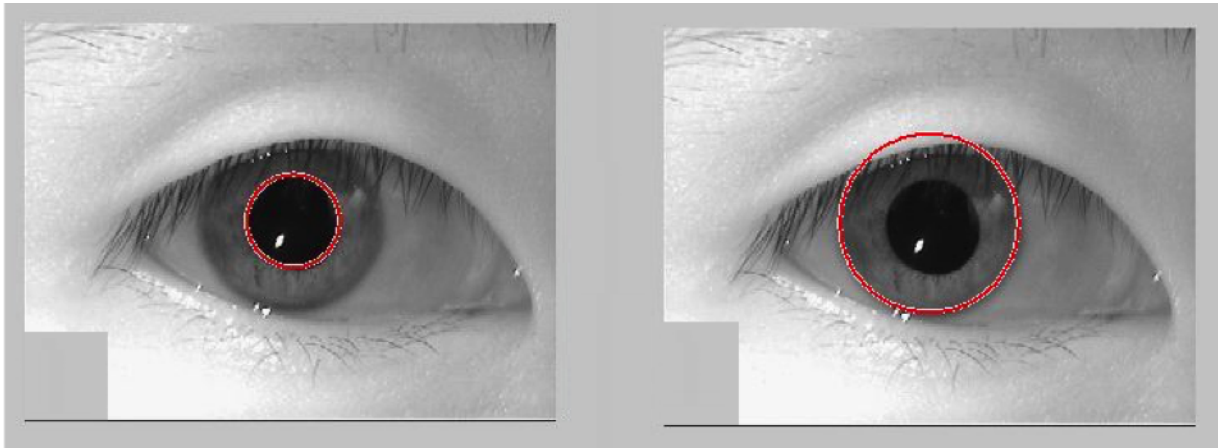


**Figure 2.3:** Dilated (a) and non-dilated (b) pupil

As explained earlier, pupil and iris size is extracted, and then  $\text{pupil/iris} = x$  is fed into neural network.

### 3.5 Feature Extraction

As explained earlier, pupil and iris size is segmented from iris images and radius of them are compared (Figure 2.4), ratio of pupil to iris is taken as input value (Table 1.1), 60 sample of 1 element (60x1 matrix) is fed into neural network with %25 validation %25 testing and %50 training ratio corresponding to 60 sample of 1 element (60x1 matrix).



**Figure 2.4:** Segmentation of pupil (left) and iris (right) and their ratio as sample value

Non-dilated	0.351851851851852
Dilated	0.518518518518519
Non-Dilated	0.442307692307692
Dilated	0.538461538461538
...	...
...	...

**Table 1.1:** Example of 60x1 samples of 1 element

### 3.6 Output Target Formation

Since there is only two possibilities, dilated or non-dilated, our output formation is pretty simple, it can be 1 or 0. These 1's and 0's corresponds to the dilated and non-dilated examples as shown in Table 1.2.

Pupil Situation	Output value
Dilated	1
Non-dilated	0

**Table 1.2:** Output values of classification

### 3.7 ANN Structure

In this thesis, Matlab GUI was used to generate neural network code for classification. A two-layer feed-forward network with sigmoid hidden and softmax output neurons is provided from Neural Pattern Recognition application. More precisely, network is trained with scaled conjugate gradient backpropagation, which is a two layer feed-forward network. Network is fed with pupil to iris ratio 60 samples of 1 element and 60 samples of corresponding output of 1 element as consisted of 10 hidden neurons. Rest of parameters is default by Matlab Neural network pattern recognition application.

### 3.8 ANN Topology

The network was created in Matlab environment with pattern recognition tool. Back propagation method was chosen. Input and outputs were provided to network and number of hidden neurons was determined as 10 by trial and error method. Network structure is defined by pattern recognition tool. Scaled conjugate gradient algorithm for training of network with the function 'trainscg' is used and input and target values are randomly separated with the function 'dividerand' and cross-entropy error is used instead of mean-square error default by pattern recognition tool. Number of maximum iterations is set to 1000.

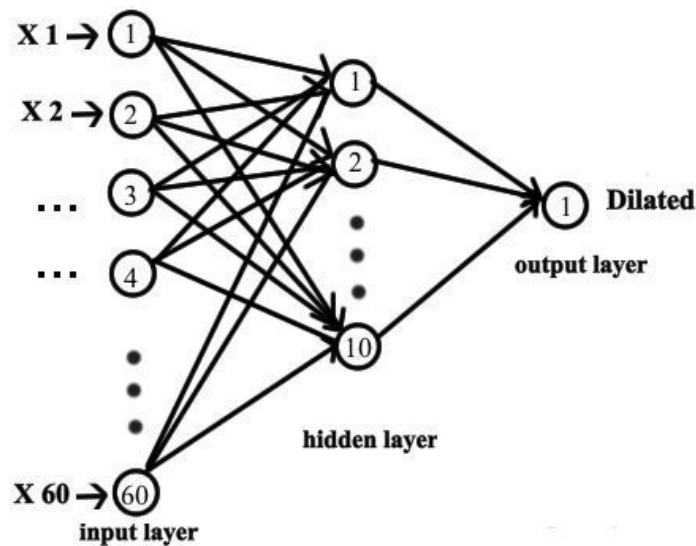


Figure 2.5: Topology of the neural network

$X_1 \dots X_{60}$  are samples that were fed into neural network. Arrows between neurons presents weights (Figure 2.5).

Number of neurons in input layer	60
Number of neurons in output layer	60
Number of neurons in hidden layer	10
Iteration number	26
Training time	2 sec
Activation function	Sigmoid

**Table 1.3:** Parameters of ANN

All parameters of ANN are explained in Table 2.1.

### 3.9 Training of Neural Network

60x1 matrix as input with 60 samples of 1 element is fed into neural network; %50 of samples is used for training for neural network. Training stops when there is increase cross-entropy error of the validation of samples. After training is finished, accuracy of system is calculated, which is subtraction of neural network's actual output from maximum output (set as "1") 8 and division of that number to total number of samples.

### 3.10 Testing and Validation of Neural Network

Validation measure generalization of network and stops when generalization decreases and testing is used for measurement of network performance during and after training. %25 of samples are used for validation and another %25 is used for testing (Vaxevanidis et al., 2013).



## **CHAPTER 4**

### **RESULTS AND DISCUSSION**

#### **4.1 Results and Discussion**

Iris images that are taken from MMU database were preprocessed and segmentation of iris and pupil was achieved, ratio of pupil to iris was taken as sample of matrix to fed into neural network at total of 60 samples. %50 of data are used for training the network and %25 of data for validation and another %25 for validation. For sake of simplicity, scaled conjugate gradient backpropagation method was used, which is multi-layer feed-forward network.

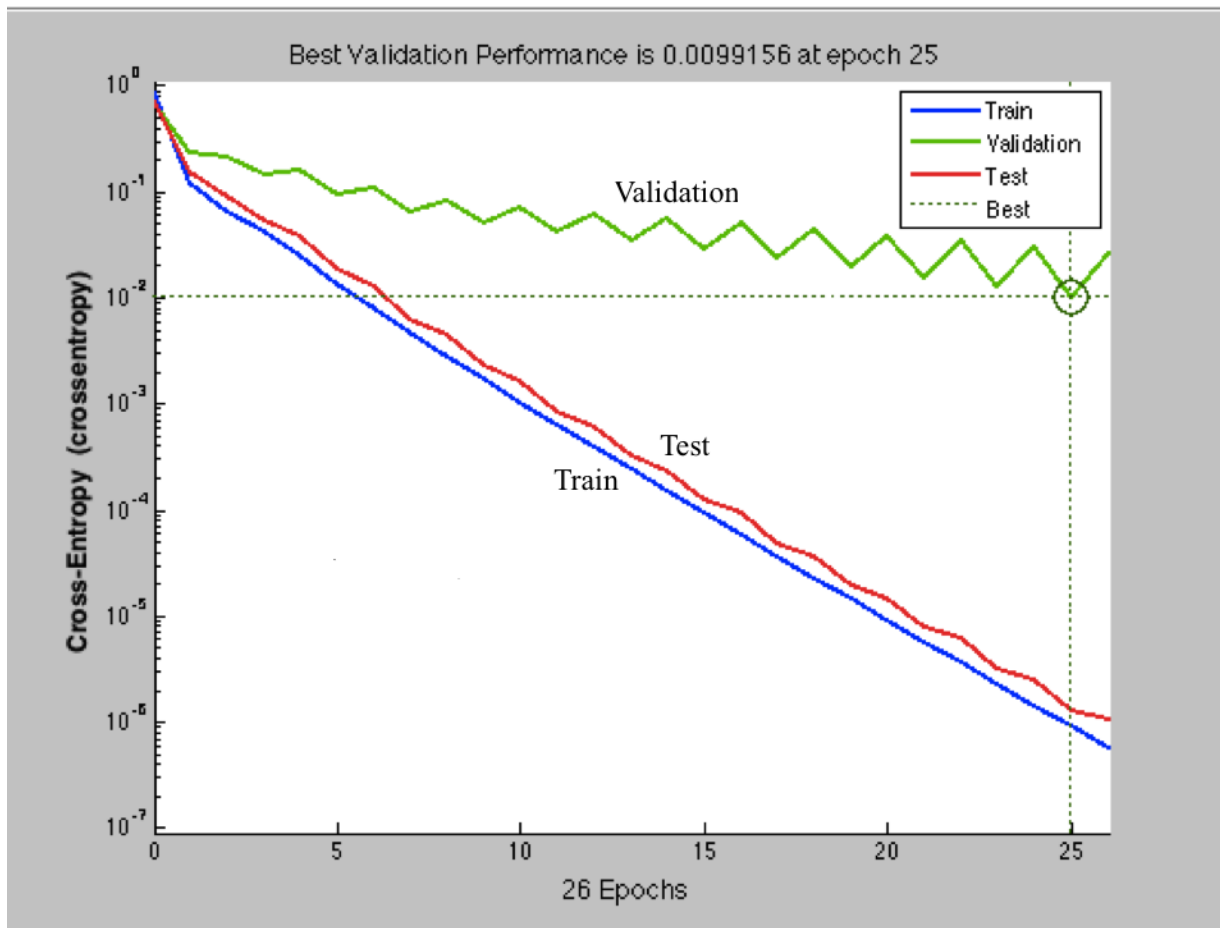
One input and one output was created which was fed with 60x1 samples of 1 element and corresponding 1 output with 60x1 samples of 1 element. Hidden layers were tested as 2, 7, 10, 20, 50 and best result was acquired with 10 hidden layers. Diffraction between misclassified samples is zero, which mean there is no misclassification, so system works with %99.99 accuracy.

#### 4.1.1 ANN results

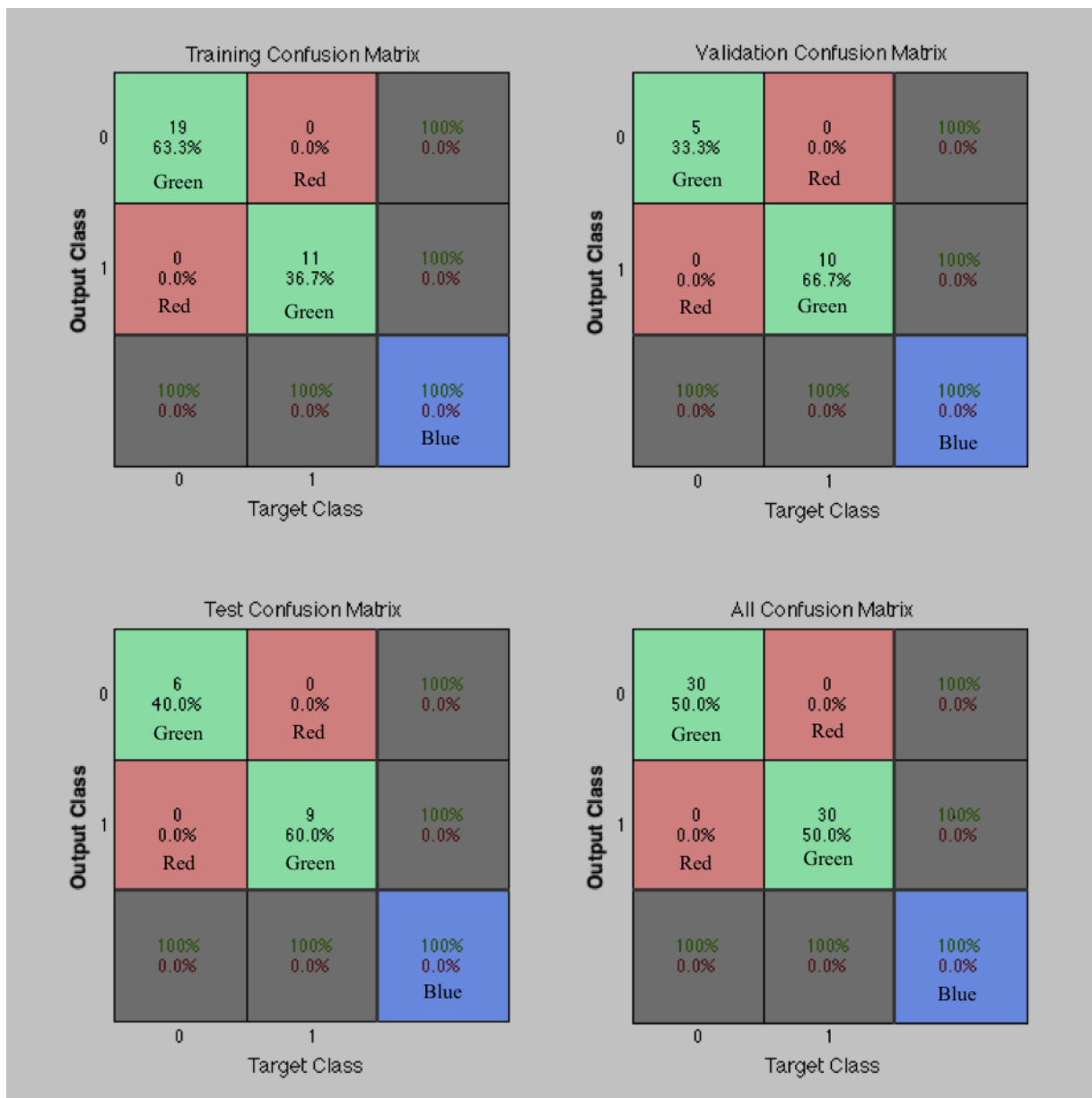
Results were provided by GUI application of toolbox as follows (Table 2.1 & Figure 3.1 & Figure 3.2 & Figure 3.3):

Number of input layer	1
Number of output layer	1
Number of hidden neuron	10
Number of Epochs	26 iteration
Time for All ANN process	2 seconds
Error	0.002

**Table 2.1:** Results of ANN

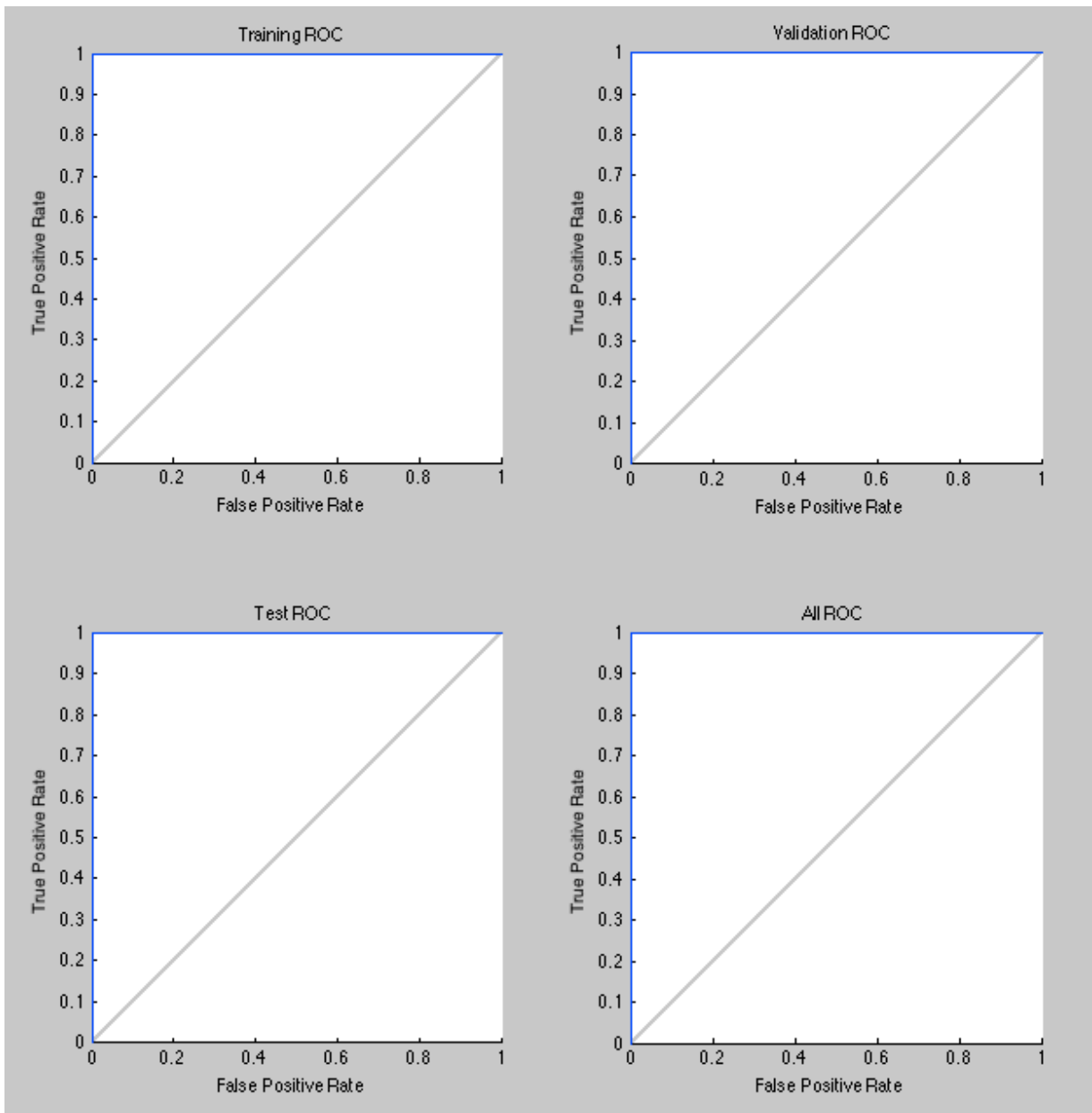


**Figure 3.1:** Best Validation Performance Acquired at epoch 25



**Figure 3.2:** Confusion Matrix

Red squares shows incorrect responses, which are zero in all boxes, green squares shows high number of correct responses and blue squares shows overall accuracy of classification which is %100 (Figure 3.2).



**Figure 3.3:** True positive rates

True positive rates exhibit %100 sensitivity and specificity for all receiver-operating characteristics (Figure 3.3).

## 4.2 Discussion

Results show that, we had high number of final results with 10 hidden layer exhibit excellent results. With help of preprocessing and following successful segmentation, should have excellent results since neural classifier is working on numbers in this case. Radius of pupil and iris was taken instead of full images, which results in much faster time. Preprocessing increases chance of segmentation, without sufficient preprocessing such as filtering and noise removal, we wouldn't be able to segmentate pupil and iris, which leads to performance of neural network. Total of 60 samples are used for Neural Classification with %50 training and %25 validations and %25 testing ratio. At the final, %99.99 accuracy was achieved as images are fully classified correctly with Artificial Neural Network Pattern Recognition GUI tool.

These images are taken from MMU-Iris database (MMU Iris database, 2004) and the ones with larger pupil are assumed to be dilated ones and ones with small pupil are assumed to be non-dilated ones. These images were chosen to show how system works. By having the same camera with real investigation of lying, person of interest could be asked with multiple choices of questions with GKT techniques as mentioned before (Walczyk, 2013; Lykken, 1998), in the questions asked, if person of interest shows arousal it will result in dilated pupil. After this process, images of dilated and non-dilated examples could be fed into neural network to train the network and then other images, which are taken during investigation, could be also tested. Since our design takes ratio of pupil to iris, it will not get affected from distances variations as we planned it to work in that direction.

## **CHAPTER 5**

### **CONCLUSION**

In this thesis, one of the cues of deception, namely pupil dilation was studied. Formulation of pupil dilation in situation of lying was supported by researches in this field. Iris database was used in order to study classification of dilated and non-dilated samples. Samples were separated as dilated and non-dilated into two groups depending on pupil size. Images were preprocessed in order to be available for segmentation. Two different image-processing methods were used for pupil segmentation and iris segmentation differently. Preprocessing of samples for successful segmentation was applied to images, once for pupil and once for iris. Preprocessing methods were explained among the chosen methodology. Preprocessing made segmentation doable. Segmentation of pupil was done by Otsu's method and followed by Hough transform to determine its radius and iris was done by canny edge detection and was followed by Hough transform for same reason. Radius of iris and pupil was segmented successfully. Ratio of pupil to iris was used as data to be fed into the neural network as 60 samples. In order to be cost efficient neural network was employed within Matlab environment by using the neural network pattern recognition tool and back-propagation method was applied to the samples. Matlab 2014a the neural network pattern recognition application was used to classify samples by using backpropagation algorithm. Application provides many details in its default settings. Only input, output and hidden neurons were given to the system and classification was done by application, which also provides code line of the application. All of the samples were correctly classified.

## REFERENCES

Eye structure and functions (n.d.) retrieved January 5, 2015, from [en.excimerclinic.ru/press/stroenieglaza](http://en.excimerclinic.ru/press/stroenieglaza)

“Find Circles Using Hough Transform”, (n.d.). Retrieved February 20, 2015, from <http://www.mathworks.com/help/images/ref/imfindcircles.html>

Garrity J. (2013). Structure and function of the eyes. Retrieved January 5, 2015, from <http://www.merckmanuals.com/home/eye-disorders/biology-of-the-eyes/structure-and-function-of-the-eyes>

Goldwater, B. C. (1972). Psychological significance of pupillary movements. *Psychological bulletin*, 77(5), 340-355.

Gonzalez, R. C., Woods, R. E., & Eddins, S. L. (2004). Image processing using MATLAB, 1<sup>st</sup> ed. Prentice-Hall.

Gonzalez, R. C., & Woods, R. E. (2008). Digital Image Procesing, 3<sup>nd</sup> ed. Prentice-Hall.

Gonzales, R. C., & Woods, R. E. (2002). Digital Image Procesing, 2<sup>nd</sup> ed., Prentice Hall.

Granholm, E., & Steinhauer, S. R. (2004). Pupillometric measures of cognitive and emotional processes. *International Journal of Psychophysiology*, 52(1), 1-6.

Hough, P.V.C. (1962). Methods and Means for Recognizing Complex Patterns. U.S. Patent 3,069,654.

Multimedia University, MMU Iris Image Database, retrieved October 17, 2014, from <http://pesona.mmu.edu.my/ccteo>

Jain, A. K., Mao, J., & Mohiuddi, K. (1996). Artificial Neural Networks: A Tutorial. *IEEE Computational Science & Engineering*, 29(3), 31-44.

Kahneman, D., & Beatty, J. (1966). Pupil diameter and load on memory. *Science*, 154, 1583-1585.

Lykken, D. T. (1998). *A Tremor in the Blood: Uses and Abuses of the Lie Detector*. New York: McGraw-Hill

Maintz, T. (2005). *Digital and Medical Image Processing*. Available from, <http://www.cs.uu.nl/docs/vakken/ibv/reader/readerINFOIBV.pdf>

Matworks. (2008). How do I convert my RGB image to grayscale without using the Image Processing Toolbox. Retrieved January 30, 2015, from <http://www.mathworks.com/matlabcentral/answers/99136-how-do-i-convert-my-rgb-image-to-grayscale-without-using-the-image-processing-toolbox>

mydriasis. (n.d.). In *Miller-Keane Encyclopedia and Dictionary of Medicine, Nursing, and Allied Health*, Retrieved June 30, 2015, from <http://medical-dictionary.thefreedictionary.com/mydriasis>

Otsu, N. (1979). A Threshold selection Method from Gray-Level Histograms. *IEEE Trans. Systems, Man, and Cybernetics*, 62(1), 55-59.

Rumelhart, D.E., Hinton, G. E., & Williams, R. J. (1986). Learning internal representations by error propagation. In D. E. Rumelhart & J. L. McClelland (Eds.), *Parallel Distributed Processing: Explorations in the Microstructure of Cognition* (pp. 318–362). Cambridge: MIT Press.



Sima, J. (1998). Introduction to Neural Networks: Mathematical Model of Neural Network. Retrieved March 17, 2015, from [http://www2.fiit.stuba.sk/~cernans/nn/nn\\_download/Sima\\_introduct.pdf](http://www2.fiit.stuba.sk/~cernans/nn/nn_download/Sima_introduct.pdf)

Strel. (n.d.). Retrieved January 15, 2015, from <http://www.mathworks.com/help/images/ref/strel.html>

The human brain. (n.d). Retrieved January 5, 2015, from <http://www.ib.bioninja.com.au/options/option-e-neurobiology-and-2/e5-the-human-brain.html>

Vaxevanidis, N. M., Kechagias, J. D., Fountas, N. A., & Manolakos, D. E. (2013). Three Component Cutting Force System Modeling and Optimization in Turning of AISI D6 Tool Steel Using Design of Experiments and Neural Networks. *In Proceedings of the World Congress on Engineering* (pp. 629-633). London, UK: International Association of Engineers.

Walczyk, J. J., Igou, F. P., Dixon, A. P., & Tcholakian, T. (2013). Advancing Lie Detection by Inducing Cognitive Load on Liars: A Review of Relevant Theories and Techniques Guided by Lessons from Polygraph-Based Approaches. *Frontiers in Psychology*, doi:10.3389/fpsyg.2013.00014

Werbos, P. J. (1974). Beyond Regressions: New Tools for Prediction and Analysis in the Behavioral Sciences. PhD Thesis, Harvard University, Cambridge, MA.

Zuckerman, M., DePaulo, B. M., & Rosenthal, R. (1981). Verbal and nonverbal communication of deception. In L. Berkowitz (Ed.), *Advances in experimental social psychology* (pp. 1–59). New York, NY: Academic Press.

## APPENDIX

### SOURCE CODE

- **IMAGE PROCESSING AND SEGMENTATION**

```
%%LOCATION OF SAMPLES%%
z=(' /Users/fatihnurcin/Documents/MATLAB/samples/dilated')
cd(z)
list = dir('*.bmp');
% preallocate cell array
img = cell(length(list),1);
f = cell(length(list), 250-100);
for i = 1:length(list)
    img{i} = imread(list(i).name);
end
for i = 1:length(list)
%%NOISE REMOVAL%%
a=img{i};
%%RGB TO GRAY%%
x1=rgb2gray(a);
fasf=x1;
for k=2:5
se=strel('disk',k);
fasf=imclose(imopen(fasf, se), se);
end
%%SHARPENING OF BLACK AREA FOR MORE SUCCESFUL
SEGMENTATION%%
fasf=imsharpen(fasf);
se2=strel('disk',10);
BW = im2bw(fasf,0.15);
%%THRESHOLDING OF PUPIL%%
```

```

BW12=1-BW;
BW2=imclose(BW12,se2);
BW23=1-BW2;
%%HOUGH TRANSFORM FOR PUPIL SEGMENTATION%%
[centers, radii] = imfindcircles(BW23,[15
50],'ObjectPolarity','dark')
[centers, radii] = imfindcircles(BW23,[15
50],'ObjectPolarity','dark', ...
    'Sensitivity',0.93);
imshow(x1);
h = viscircles(centers,radii);
    f{i}=ceil(radii);

end
g=f(:,1);
f=cell2mat(g)
%%IRIS SEGMENTATION PART%%
z=(' /Users/fatihnurcin/Documents/MATLAB/samples/dilated')
cd(z)
list = dir('*.bmp');
% preallocate cell array
img = cell(length(list),1);
f1 = cell(length(list), 250-100);
for i = 1:length(list)
    img{i} = imread(list(i).name);
end
for i = 1:length(list)
    a=img{i};
    a1=rgb2gray(a);
    a12=imcomplement(a1);
    a1=im2double(a1);

```

```

c=2;
%%LOGARITHMIC TRANSFORMATION%%
S=c*log(1+a1);
%%CANNY EDGE DETECTION%%
BW4 = edge(S,'canny',0.10) ;
[centers, radii] = imfindcircles(BW4,[46
70],'ObjectPolarity','dark')
[centers, radii] = imfindcircles(BW4,[46
70],'ObjectPolarity','dark', ...
    'Sensitivity',0.95);
imshow(a1);
h = viscircles(centers,radii);
    f1{i}=ceil(radii);

end
g=f1(:,1);
f1=cell2mat(g);
%%RATIO PUPIL TO IRIS%%
fx=f./f1;

%%SAME PROCESS FOR NON-DILATED SAMPLES%%

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

z=(' /Users/fatihnurcin/Documents/MATLAB/samples/nondilated
')
cd(z)
list = dir('*.bmp');
% preallocate cell array
img = cell(length(list),1);
v = cell(length(list), 250-100);

```

```

for i = 1:length(list)
    img{i} = imread(list(i).name);
end
for i = 1:length(list)
    a=img{i};
    x1=rgb2gray(a);
    fasz=x1;
    for k=2:5
        se=strel('disk',k);
        fasz=imclose(imopen(fasz, se), se);
    end
    fasz=imsharpen(fasz);
    se2=strel('disk',10);
    BW = im2bw(fasz,0.15);
    BW12=1-BW;
    BW2=imclose(BW12,se2);
    BW23=1-BW2;
    [centers, radii] = imfindcircles(BW23,[15
50],'ObjectPolarity','dark')
    [centers, radii] = imfindcircles(BW23,[15
50],'ObjectPolarity','dark', ...
        'Sensitivity',0.93);
    imshow(x1);
    h = viscircles(centers,radii);
    v{i}=ceil(radii);

end
g=v(:,1);
v=cell2mat(g)

z=('Users/fatihnurcin/Documents/MATLAB/samples/dilated')

```

```

cd(z)
list = dir('*.bmp');
% preallocate cell array
img = cell(length(list),1);
v1 = cell(length(list), 250-100);
for i = 1:length(list)
    img{i} = imread(list(i).name);
end
for i = 1:length(list)
    a=img{i};
    a1=rgb2gray(a);
    a12=imcomplement(a1);
    a1=im2double(a1);
    c=2;
    S=c*log(1+a1);
    BW4 = edge(S,'canny',0.10) ;
    [centers, radii] = imfindcircles(BW4,[46
70],'ObjectPolarity','dark')
    [centers, radii] = imfindcircles(BW4,[46
70],'ObjectPolarity','dark', ...
        'Sensitivity',0.95);
    imshow(a1);
    h = viscircles(centers,radii);
    v1{i}=ceil(radii);

end

%%%PUTTING INPUT AND SUPERVISED OUTPUT VALUES TO ARRAY%%%
g=v1(:,1);
v1=cell2mat(g);
vx=v./v1;

```

```

    i=1;
    j=1;

    while i<61
        inpt(i,1)=vx(j,1);
        inpt(i+1,1)=fx(j,1);
        i=i+2;
        j=j+1;
    end
    i=1;
    j=1;
    while i<61
        outpt(i,1)=0;
        outpt(i+1,1)=1;
        i=i+2;
        j=j+1;
    end
end

```

- **NEURAL NETWORK**

```

% Solve a Pattern Recognition Problem with a Neural
Network
% Script generated by Neural Pattern Recognition app
% Created Sun May 10 20:47:17 EEST 2015
%
% This script assumes these variables are defined:
%
%   inpt - input data.
%   outpt - target data.

x = inpt';
t = outpt';

```

```

% Create a Pattern Recognition Network
hiddenLayerSize = 10;
net = patternnet(hiddenLayerSize);

% Setup Division of Data for Training, Validation, Testing
net.divideParam.trainRatio = 50/100;
net.divideParam.valRatio = 25/100;
net.divideParam.testRatio = 25/100;

% Train the Network
[net,tr] = train(net,x,t);

% Test the Network
y = net(x);
e = gsubtract(t,y);
tind = vec2ind(t);
yind = vec2ind(y);
percentErrors = sum(tind ~= yind)/numel(tind);
performance = perform(net,t,y)

% View the Network
view(net)

% Plots
% Uncomment these lines to enable various plots.
%figure, plotperform(tr)
%figure, plottrainstate(tr)
%figure, plotconfusion(t,y)
%figure, plotroc(t,y)
%figure, ploterrhist(e)

```