A NEW FAST FUZZY EDGE DETECTOR

Major Project submitted in partial fulfilment of the requirements for the award of degree of

Master of Technology
In
Information System

Submitted by:

Veni Jain

(19/IS/2010)

Under the Guidance of:

Prof. O. P. Verma

Department of Information Technology
Delhi Technological University
Bawana Road, Delhi – 110042
(2010-2012)

CERTIFICATE

This is to certify that **Ms. Veni Jain (19/IS/10)** has carried out the major project titled **"A New Fast Fuzzy Edge Detector"** as a partial requirement for the award of Master of Technology degree in Information Systems by Delhi Technological University.

The major project is a bonafide piece of work carried out and completed under my supervision and guidance during the academic session **2010-2012**. The matter contained in this report has not been submitted elsewhere for the award of any other degree.

(Project Guide)
Prof. O. P. Verma
Head of Department
Department of Information Technology
Delhi Technological University
Bawana Road, Delhi-110042

ACKNOWLEDGEMENT

I express my gratitude to my major project guide **Prof. O. P. Verma**, HOD, IT Dept., Delhi Technological University, for the valuable support and guidance he provided in making this major project. It is my pleasure to record my sincere thanks to my respected guide for his constructive criticism and insight without which the project would not have shaped as it has.

I humbly extend my words of gratitude to other faculty members of this department for providing their valuable help and time whenever it was required.

Veni Jain
Roll No. 19/IS/10
M.Tech (Information Systems)
E-mail: vjain1735@gmail.com
Department of Information Technology
Delhi Technological University
Bawana Road, Delhi-110042

ABSTRACT

Edge Detection is an important step in many of the image processing techniques. A wide variety of operators have been proposed for edge detection in past. But Most of the edge detection methods available in the literature are gradient based which further apply thresholding to find the final edge map in an image. In this paper, a novel method based on fuzzy logic is proposed for edge detection in gray images without using the gradient and thresholding techniques. Here, the fuzzy logic is used to conclude whether a pixel is an edge pixel or not. The technique begins by fuzzifying the gray values of pixel into two fuzzy variables namely the *black* and the *white*. Fuzzy rules are defined to find the edge pixels in the fuzzified image. The resultant edge map obtained so far may contain some extraneous edges, which are further removed from the edge map by separately examining the intermediate intensity range pixels. Finally, edge map is improved by finding some left out edge pixels by defining new membership function for the pixels which have their entire 8-neighbourhood pixel classified as white. The proposed method is compared with some of the existing standard edge detector operators available in the literature. The quantitative analysis of the proposed method is given in term of entropy value. Entropy value signifies the information content in image.

TABLE OF CONTENTS

Certificate………………………………………..……………………… i

Acknowledgement…..…………………..……………………………… ii

Abstract……………………..……………………………………….... iii

List of Figures and Tables…………………………………………………. vi

Chapter 1: Introduction…..…………………………...………………. 1

1.1 Digital Image Processing…………………………………………….... 1

1.2 Related Work……………………………………………………….... 2

1.3 Thesis Layout……………………….…………………………….. 4

Chapter 2: Edge Detection Operators……………………………….... 5

Chapter 3: Fuzzy Image Processing………………………………….. 10

3.1 Applications of Fuzzy Logic……………………………………… 10

3.2 Structure of Fuzzy Image Processing……………………………. 11

Chapter 4: Proposed Approach………………...……………………. 13

4.1 Fuzzification………………………………………………………….. 13

4.2 Fuzzy Rules………………………………………………………… 15

4.3 False Edge Removal…………………………………………........ 21

4.4 Edge Improvement…………………………………………………………….. 22

4.5 Algorithm……………………………………………………………………. 23

4.6 Pseudo-Code……………………………………………………………….. 24

Chapter 5: Experimental Results……………………………………………… 25

5.1 Comparison with standard edge detectors……………………………......... 25

5.2 Quantitative Analysis………………………………………………………… 33

Chapter 6: Conclusions…………………………………………………………… 34

References……………………………………………………………………….. 35

LIST OF FIGURES AND TABLES

Figure 2.1: Roberts Operator………………………………………………… 6

Figure 2.2: Prewitt Operator……..…………………………………………… 7

Figure 2.3: Sobel Operator……….………………………………………….. 8

Figure 2.4: Log Operator………..…………………………………………... 8

Figure 3.1: General Structure of Fuzzy Image Processing…………………… 10

Figure 4.1: Triangular Membership Function………………………………… 14

Figure 4.2: Fuzzification step of the proposed edge detection…….……….. 15

Figure 4.3: 8-neighbourhood of pixel (i, j)…………..……………………… 16

Figure 4.4(a): Class 1 Fuzzy Rules……………………………………………… 17

Figure 4.4(b): Class 2 Fuzzy Rules……………………………………………… 18

Figure 4.4(c): Class 3 Fuzzy Rules……………………………………………… 19

Figure 4.4(d): Class 4 Fuzzy Rules……………………………………………… 20

Figure 4.4(e): Class 5 Fuzzy Rules………………………………………………. 21

Figure 4.5: Membership Function for intensity range [141…250]…………… 23

Figure 5.1: Original Test Images..………………………………………….. 26

Figure 5.2: Edge Map of Lena Image..………………………………………. 27

Figure 5.3: Edge Map of Cameraman Image…..……………………………… 28

Figure 5.4: Edge Map of Baboon Image……….……………………………… 29

Figure 5.5: Edge Map of Peppers Image……….……………………………… 30

Figure 5.6: Edge Map of Pillset Image……………….…………………………… 31

Figure 5.7: Edge Map of Sample test Image…….…………………………….. 32

Table I: Entropy Values Of Different Edge Detector For Multiple Images… 33

Chapter 1
INTRODUCTION

1.1 Digital Image Processing

Digital image processing includes the processes of acquiring an image of the area containing the text, pre-processing that image, extracting (segmenting) the individual characters, describing the characters in a form suitable for computer processing, and recognizing those individual characters. Image processing operations can be roughly divided into four major categories:

a) Image Restoration- Restoration takes a corrupted image and tries to recreate a image which is close to the expected image which is more cleaner. As many sensors are subject to noise, they result in corrupted images that don’t reflect the real world scene accurately and old photograph and film archives often show considerable damage.

b) Image Enhancement- Image Enhancement alters an image to make its look and feel better to human observers. It is often used to increase the contrast in images that are substantially dark or light. Enhancement algorithms often play attention to humans’ sensitivity to contrast.

c) Image Compression- Image compression is the process that helps to represent image data with as few bits as possible by exploiting redundancies in the data while maintaining an appropriate level of quality for the user. The human eye has less spatial sensitivity to color than for luminance information. Large amounts of data are used to represent an image, so image has to be compressed when transferring from one place to another.

d) Image Segmentation- Segmentation is the process that subdivides an image into a number of uniformly homogeneous regions. Each homogeneous region is a constituent part or object in the entire scene. The objects on the land part of the scene need to be appropriately segmented and subsequently classified. Partitioning of an image is based on abrupt changes in gray level. If edges of the image can be extracted and linked, the region is described by the edge boundaries that contain it. Here comes the need for edge detection in images.

Edge detection, in image processing, plays a vital role in the many of the applications such as pattern recognition, image segmentation. Edges correspond to sharp variations of image intensity and convey vital information in an image. Edges are formed from pixels with derivative values that exceed a pre-set threshold [1]. Edge detection not only extract the edges of the interested objects from an image, but also form the basis for image fusion, shape extraction, image matching, and image tracking etc..

Goals of edge detection

- Produce a line drawing of a scene from an image of that scene.
- Important features can be extracted from the edges of an image (e.g., corners, lines, curves).
- These features are used by higher-level computer vision algorithms (e.g., recognition).

1.2 Related Work

Normally, the edge detection methods use the gradient of images and arithmetic operators. Most popular edge detection methods such as Sobel, Prewitt, Roberts [1] etc. detect edges using first-order derivative of intensity since they consider edges to be a set of pixels where there is an abrupt change in intensity of gray level. Canny edge detector [2], however, searches for the partial maximum value of image gradient. The gradient is counted by the derivative of Gauss filter. The Canny operator uses two thresholds to detect strong edge and weak edge respectively.

These edge detection methods do not consider neighbourhood of the pixel while in proposed method neighbourhood plays an important role in edge detection. These classical operators [1-2] work well in circumstances where the area of the image under study is of high contrast. The theory of fuzzy sets was first introduced by Zadeh [3], and has been applied successfully in many image processing and pattern recognition problems.

Fuzzy logic is a mathematical logic that attempts to solve problems by assigning values to an imprecise spectrum of data in order to arrive at the most accurate conclusion possible. Many edge detection techniques have been proposed based on fuzzy sets and rules. For instance, Pal et al [4] proposed a fuzzy edge detection method for X-ray images. Zhang *et al.* [5] proposed a fuzzy edge detection method which overcame the drawbacks of Pal *et al.* [5] method by new enhancement operator based on Gaussian function. Law *et al.* [6] used fuzzy reasoning for edge detection which included three stages: filtering, detection, and tracing. Alshennawy *et al.* [7] proposed a fuzzy inference system for finding the edge maps of simple binary images without using the complex derivatives and thresholding but the method works well for simple black and white images and not for complex images. Begol *et al.* [8] improved this drawback by considering each pixel as a fuzzy input and by examining fuzzy rules in its vicinity, the edge pixel is specified more clearly as compared to Alshennawy *et al.* [7]. Kiranpreet Kaur *et al.* [9] proposed a similar fuzzy logic based image edge detection algorithm having 2×2 window instead of 3×3 used in Alshennawy *et al.* [7]. Drawback of this method is the need to process the output again for better results.

Another excellent work proposed by Liang et al. [10] defines a competitive fuzzy edge detector using fuzzification for classification and then competitive rules are applied to find edges. This method gives significant results but requires users to set the edge sensitivity level.

There are other edge detection methods which use hybrid of fuzzy with other techniques like neural networks, arithmetic operators etc. For instance, Wang *et al.* [11] proposed an edge detection method which is hybrid of fuzzy and neural networks which gives better results than using individual methods. Another Fuzzy edge detection method proposed by Sinaie *et al.* [12] is a hybrid edge detection method based on fuzzy sets and cellular learning automata (CLA). It uses fuzzy sets to find edges and CLA to enhance the edges, hence, a bit time-consuming. Tizhoosh, H.R. [13] proposed fast fuzzy edge detectors based on heuristic membership functions, simple fuzzy rules, and complement are simple in implementation and fast in computing. These methods give rough edge maps which are useful in some cases where results are required urgently.

There are some other methods also which use combination of fuzzy rules and evolutionary techniques. For instance, C.Naga Raju *et al.* [14] proposed a fuzzy logic based ant colony system used for image classification and analysis which requires extensive computation and does not produce good results for images which contains small and overlapped objects. Khalid *et al.* [15] combine the fuzzy heuristic edge detection technique with the particle swarm optimization algorithm. Another excellent approach for the edge detection using a combination of bacterial foraging algorithm (BFA) and probabilistic derivative technique derived from ant colony system is presented in [16], conveys the vitally important information about an edge structure.

1.3 Thesis Layout

The project is divided into 6 chapters. Each chapter deals with one component related to this project work. Chapter 1 being introduction to this project work gives us the brief introduction to this project work, there after chapter 2 explains the various Edge detection operators available in literature.

Chapter 3 deals with Fuzzy image processing and its advantages. Chapter 4 deals with edge detection using the proposed approach in the project. The significant development that we have made in the project as compared to previous methods is that without the use of tedious derivative based calculations of previous edge detection operators it gives very significant results.

Chapter 5 deals with the comparison of proposed method with other edge detection operators. Quantitative analysis of the proposed method is done by comparing entropy values. Chapter 6 concludes the project work.

Chapter 2
EDGE DETECTION OPERATORS

There are lot of edge detectors available in literature and a lot many have been proposed. Most of them are based on first order derivative or second derivative. Most commonly used operators are the Differential, Log, Canny operators and Binary morphology etc.

DIFFERENTIAL OPERATOR:

Differential operator can outstand grey change. There are some points where grey change is bigger. And the value calculated in those points is higher applying derivative operator. So these differential values may be regarded as relevant 'edge intensity' and gather the points set of the edge through setting thresholds for these differential values. First derivative is the simplest differential coefficient. Suppose that the image is $f(x, y)$, and its derivative operator is the first order partial derivative $\partial f / \partial x$, $\partial f / \partial y$. They represent the rate-of-change that the gray f is in the direction of x and y. Yet the gray rate of change in the direction of α is shown in the equation (1):

$$\partial f / \partial \alpha = \partial f / \partial x \cos\alpha + \partial f / \partial y \sin\alpha \qquad (1)$$

Under consecutive circumstances, the differential of the function is

$$df = \partial f / \partial x \, dx + \partial f / \partial y \, dy$$

The direction derivative of function $f(x, y)$ has a maximum at a certain point. And the direction of this point is $\arctan[(\partial f / \partial x) / (\partial f / \partial y)$. The maximum of direction derivative is $sqrt(\partial f / \partial x^{2} + \partial f / \partial y^{2})$. The vector with this direction and modulus is called as the gradient of the function f, that is.

$$\nabla f(x, y) = \left(\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}\right)'$$

So the gradient modulus operator is designed in the equation (2).

$$G[f(x,y)] = sqrt({\partial f/\partial x}^2 + {\partial f/\partial y}^2) \qquad (2)$$

For the digital image, the gradient template operator is designed as

$$G[f(i,j)] = [\Delta_x f(i,j)^2 + \Delta_y f(i,j)^2]^{1/2} \qquad (3)$$

Where $\Delta_x f(i,j) = f(i,j) - f(i-1,j)$, $\Delta_y f(i,j) = f(i,j) - f(i,j-1)$

Differential operator includes Roberts operator and Sobel operator.

(1) Roberts Operator

Roberts operator is a kind of the most simple operator which makes use of partial difference operator to look for edge. Its effect is the best for the image with steep low noise. But the borderline of the extracted image is quite thick using the Roberts operator, so the edge location is not very accurate.

Roberts operator is defined as:

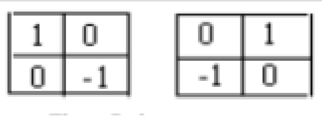
$$G[f(x,y)] = \{[f(x+1,y+1) - f(x,y)]^2 + [f(x+1,y) - f(x,y+1)]^2\}^{1/2} \, (4)$$

But absolute deviation algorithm is usually used to predigest the equation (4) in practice. The following equations (5) and (6) are the process of reduction.

$$G[f(x,y)] \approx | f(x+1,y) - f(x,y) | + | f(x,y+1) - f(x,y) | \qquad (5)$$

$$G[f(x,y)] \approx | f(x+1,y+1) - f(x,y) | + | f(x,y+1) - f(x+1,y) | \qquad (6)$$

The template of the Roberts operator is shown in Fig. 2.1



1	0
0	-1

0	1
-1	0

Fig. 2.1 Roberts Operator

(2) Sobel and Prewitt Operator

To reduce the influence of noise when detecting edge, the Prewitt operator enlarges edge detection operator template from two by two to three by three to compute difference operator. Using the Prewitt operator can not only detect edge points, but also restrain the noise. The Sobel operator counts difference using weighted for 4 neighbourhoods on the basis of the Prewitt operator. The Sobel operator has the similar function as the Prewitt operator, but the edge detected by the Sobel operator is wider.

Suppose that the pixel number in the 3×3 sub-domain of image is as follow:

$$\begin{array}{ccc} A_0 & A_1 & A_2 \\ A_7 & f(i,j) & A_3 \\ A_6 & A_5 & A_4 \end{array}$$

We order that $X = (A_0 + A_1 + A_2) - (A_6 + A_5 + A_4)$ and $Y = (A_0 + A_7 + A_6) - (A_2 + A_3 + A_4)$ .Then Prewitt operator is as follows:

$$G[f(i,j)] = (X^2 + Y^2)^{1/2} \quad (7)$$

or
$$G[f(i,j)] = |X| + |Y| \quad (8)$$

Prewitt operator is said in Fig.2.2 in the form of the template.

1	-1	1
1	-1	0
1	-1	-1

1	1	1
0	0	0
-1	-1	-1

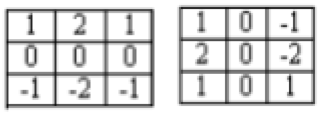
Fig. 2.2 Prewitt Operator

Sobel operator can process those images with lots of noises and gray gradient well. We order that $X = (A_0 + 2A_1 + A_2) - (A_6 + 2A_5 + A_4)$ and $Y = (A_0 + 2A_7 + A_6) - (A_2 + 2A_3 + A_4)$. Then Sobel operator is as follows:

$$G[f(i,j)] = (X^2 + Y^2)^{1/2} \quad (9)$$

or
$$G[f(i,j)] = |X| + |Y| \quad (10)$$

Sobel operator is said in Fig.2.3 in the form of the template.



1	2	1
0	0	0
-1	-2	-1

1	0	-1
2	0	-2
1	0	1

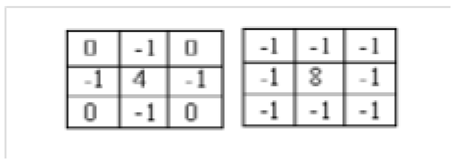
Fig. 2.3 Sobel Operator

(3) Log Operator

The Log operator is a linear and time-invariant operator. It detects edge points through searching for spots which two-order differential coefficient is zero in the image grey levels. For a continuous function *f* (*x*, *y*), the Log operator is defined as at point (x, y):

$$\Delta^2 f = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2} \tag{11}$$

The Log operator is the process of filtering and counting differential coefficient for the image. It determines the zero overlapping position of filter output using convolution of revolving symmetrical Log template and the image. The Log operator’s template is shown in Fig. 2. 4.



0	-1	0
-1	4	-1
0	-1	0

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

Fig. 2.4 Log Operator

In the detection process of the Log operator, we firstly pre-smooth the image with Gauss low-pass filter, and then find the steep edge in the image making use of the Log operator. Finally we carry on binarization with zero grey level to give birth to closed, connected outline and eliminate all internal spots. But double pixels boundary usually appears using the Log operator to detect edge, and the operator is very sensitive to noise. So the Log operator is often employed to judge the edge pixels lie in either bright section or dark section of the image.

(4) Canny Operator

The Canny operator is a sort of new edge detection operator. It has good performance of detecting edge, which has a wide application. The Canny operator edge detection is to search for the partial maximum value of image gradient. The gradient is counted by the derivative of Gauss filter. The Canny operator uses two thresholds to detect strong edge and weak edge respectively. And only when strong edge is connected with weak edge, weak edge will be contained in the output value. The theory basis of canny operator is shown in equations (12)-(15).

$$\text{Gauss: } g(x,y) = \exp[-(x^2+y^2)/2\sigma^2] \quad (12)$$

$$\text{Edge normals: } n_\perp = \nabla(g*P)/|\nabla(g*P)| \quad (13)$$

$$\text{Edge strengths: } G_n P = \frac{\partial}{\partial n_\perp}[g*P] \quad (14)$$

$$\text{Maximal strengths: } \theta = \frac{\partial}{\partial n_\perp} G_n P = \frac{\partial^2}{\partial {n_\perp}^2}[g*P] \quad (15)$$

For two-dimensional image canny operator can produce two information including the border gradient direction and intensity. Canny operator is actually using templates of different directions to do convolution to the image respectively. Then the mostly direction is taken. From the viewpoint of positioning accuracy, canny operator is better than the other operators. Therefore, this method is not easily disturbed by noise and can keep the good balance between noise and edge detection. It can detect the true weak edge.

Chapter 3
FUZZY IMAGE PROCESSING

3.1 Application of Fuzzy Logic

Fuzzy logic is a powerful problem-solving methodology with a variety of applications in embedded control and information processing. Fuzzy provides a remarkably simple way to draw definite conclusions from vague, ambiguous or imprecise information. In a sense, fuzzy logic resembles human decision making with its ability to work from approximate data and find precise solutions.

Unlike classical logic which requires a deep understanding of a system, exact equations, and precise numeric values, fuzzy logic incorporates an alternative way of thinking, which allows modeling complex systems using a higher level of abstraction originating from human knowledge and experience. Fuzzy logic allows expressing the knowledge with subjective concepts such as very hot, bright red and very small height, which are mapped into exact numeric ranges.
The notion central to fuzzy systems is that truth values (in fuzzy logic) or membership values (in fuzzy sets) are indicated by a value on the range [0.0, 1.0], with 0.0 representing absolute Falseness and 1.0 representing absolute Truth. For example, let us take the statement:

"Sham is tall."

If Sham's age was 5 feet 8 inch, we might assign the statement the truth value of 0.75. The statement could be translated into set terminology as follows:

"Sham is a member of the set of tall people."

More precisely, this statement can be represented as:

$$\mu_{tall}(sham) = 0.75$$

Where μ is the membership function, operating in this case on the fuzzy set of tall people, which returns a value between 0.0 and 1.0

3.2 Structure of Fuzzy Image Processing

Fuzzy image processing is not a unique theory. It is a collection of different fuzzy approaches to image processing. Fuzzy image processing is the collection of all approaches that understand, represent and process the images, their segments and features as fuzzy sets. The representation and processing depend on the selected fuzzy technique and on the problem to be solved. Fuzzy image processing has three main stages: image fuzzification, modification of membership values, and, if necessary, image defuzzification as shown in Fig. 5. The fuzzification and defuzzification steps are due to the fact that we do not possess fuzzy hardware. Therefore, the coding of image data (fuzzification) and decoding of the results (defuzzification) are steps that make possible to process images with fuzzy techniques. The main power of fuzzy image processing is in the middle step (modification of membership values). After the image data are transformed from gray-level plane to the membership plane (fuzzification), appropriate fuzzy techniques modify the membership values. This can be a fuzzy clustering, a fuzzy rule-based approach, a fuzzy integration approach and so on. There are many reasons to use fuzzy logic in image processing:

- Fuzzy techniques are powerful tools for knowledge representation and processing.
- Fuzzy techniques can manage the vagueness and ambiguity efficiently.
- Fuzzy logic is tolerant of imprecise data. Everything is imprecise if we look closely enough, but more than that, most things are imprecise even on careful inspection. Fuzzy reasoning builds this understanding into the process rather than tacking it onto the end.
- Fuzzy logic can model nonlinear functions of arbitrary complexity. We can create a fuzzy system to match any set of input-output data.
- Fuzzy logic is based on natural language. The basis for fuzzy logic is the basis for human communication. This observation underpins many of the other statements about fuzzy logic.
- Fuzzy logic can be blended with conventional control techniques. Fuzzy systems don't necessarily replace conventional control methods. In many cases fuzzy systems augment them and simplify their implementation.
- Fuzzy logic is conceptually easy to understand. The mathematical concepts behind fuzzy reasoning are very simple.

- Fuzzy logic is flexible. With any given system, it's easy to manage it or layer more functionality on top of it without starting again from scratch.

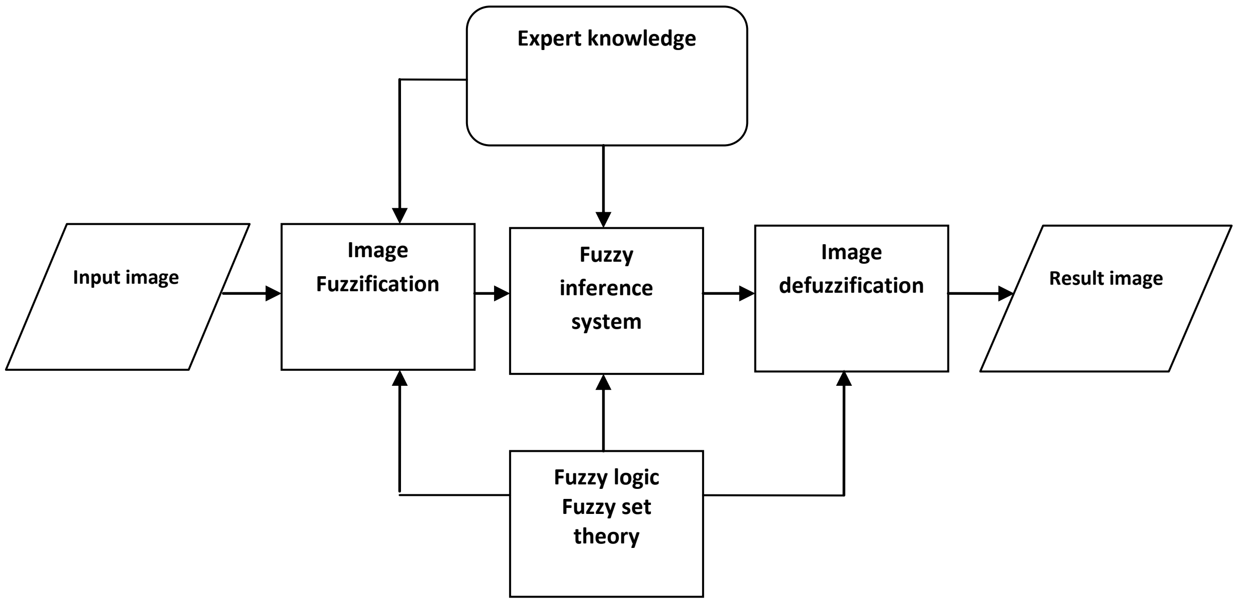


Fig. 3.1 General Structure of Fuzzy Image Processing

Chapter 4
PROPOSED APPROACH

There are a lot of ways to detect edges using fuzzy image processing. But the simplest way is to fuzzify the image which involves finding the membership value of each pixel to a particular set and then applying the defined rules on the fuzzified image to find the edge map.

The proposed edge detection method uses the defined fuzzy rules to find the candidate edge pixels, by applying the rules on each pixel along with its 8-neighbourhood. Before applying the rules, each pixel is categorised into 'white' and 'black' pixel using triangular membership function.

Proposed method comprises of 4 steps:

1. Fuzzification of input image
2. Fuzzy rules applied on the fuzzified image
3. False edge removal
4. Edge map improvement

4.1 Fuzzification

If the information in a database is inexact, incomplete or entirely not certain, then the systematic use of fuzzy logic becomes practically indispensable. In many image processing applications, the image information to be processed is uncertain and imprecise. In the proposed approach, the question of whether a pixel is darker or brighter comes under the realm of fuzzification. The darker pixels are placed in the black class whereas the brighter ones are put in white class. In order to fuzzify the image the membership of each pixel is found by using the triangular membership function shown in Fig. 4.1. Membership function of an element defines the degree to which that element belongs to the fuzzy set. Value of membership function always lies between [0…1].

The image pixels are fuzzified into two sets, viz. the black and the white using the Triangular membership function.

Membership for the fuzzy set black is defined as:

$$\mu_{black}(x) = \begin{cases} 0 & x < 0 \\ \dfrac{255 - x}{255} & 0 \leq x \leq 255 \end{cases} \tag{16}$$

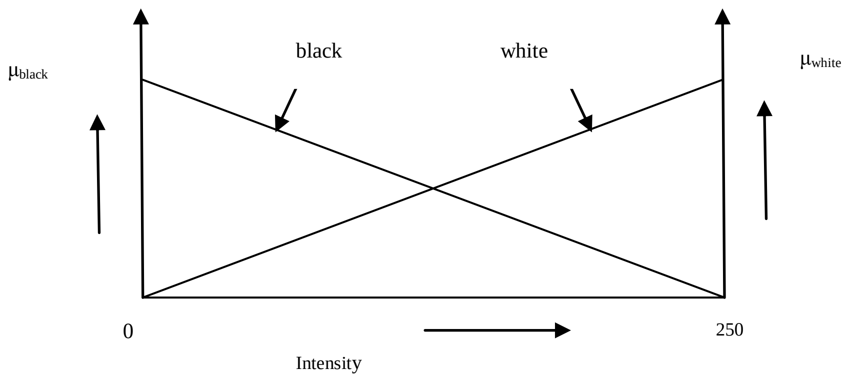


Fig. 4.1: Triangular membership function

Membership for the fuzzy set white is defined as

$$\mu_{white}(x) = \begin{cases} 0 & x < 0 \\ \dfrac{x}{255} & 0 \leq x \leq 255 \end{cases} \tag{17}$$

Here, x is intensity of pixel.

After fuzzifying the image we will have an image with pixels labelled black and white. Now we will apply the fuzzy inference rules on them to detect whether the pixel is an edge pixel or not.

After calculating $\mu_{black}(x)$ and $\mu_{white}(x)$, maximum of the membership value of both the set is found out and pixel is assigned the fuzzy variable whose membership value is high. The example of the fuzzification step for a 3×3 window is shown in Fig 4.2. Fig. 4.2(a) represents the intensity values of the pixels in 3×3 window. Fig. 4.2(b) represents the maximum of the $\mu_{black}(x)$ and $\mu_{white}(x)$ where x represents the intensity value of that pixel and finally fig 4.2(c) represents the fuzzy set whose membership value is shown in fig 4.2(b).

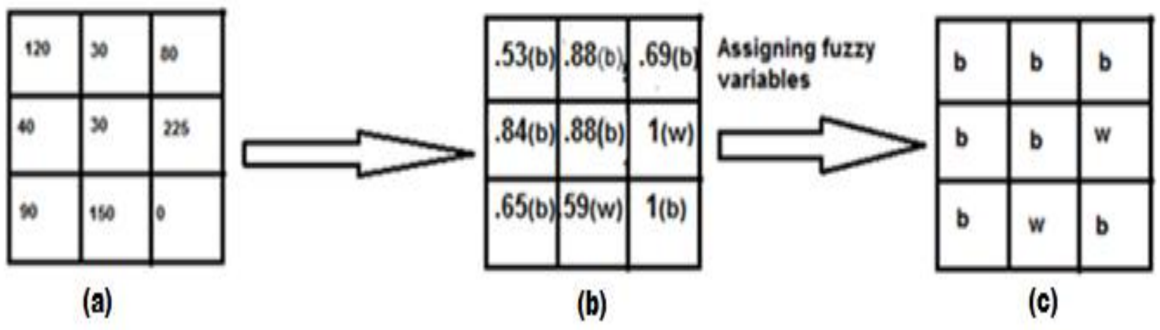


Fig. 4.2: Fuzzification step of the proposed edge detection (a) Intensity value of pixels in image (b) Final Membership of each pixel =max(μ_{black},μ_{white}) (c) Final fuzzy variable assigned to each pixel.

4.2 Fuzzy Rules

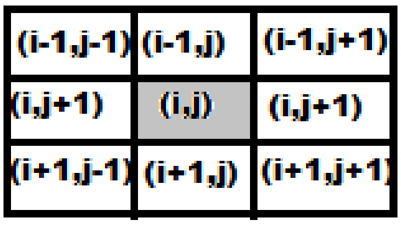
Human beings make decisions based on rules. Although, we may not be aware of it, all the decisions we make are all based on computer like if-then statements. If the weather is fine, then we may decide to go out. If the forecast says the weather will be bad today, but fine tomorrow, then we make a decision not to go today, and postpone it till tomorrow. Rules associate ideas and relate one attribute to another. Fuzzy machines, which always tend to mimic the behaviour of man, work the same way. However, the decision and the means of choosing that decision are replaced by fuzzy sets and the rules are replaced by fuzzy rules. Fuzzy rules also operate using a series of if-then statements. The decision and the means of choosing that decision are replaced by fuzzy sets and the rules are replaced by fuzzy rules. A fuzzy rule is defined as a conditional statement in the form:

IF r is A
THEN s is B

Where r and s are linguistic variables, A and B are linguistic values determined by fuzzy sets on the universe of discourse R and S, respectively. Fuzzy rules used in the proposed edge detection approach takes into account the linguistic values of the 8-neighborhood of the pixel in consideration. Here the linguistic values can be white or black.
Fuzzy rule is a control system which is used to infer decisions from a knowledge base. The knowledge base to infer edge pixel in an image is the pixel with its 8-neighborhood. The decision whether each pixel is an edge pixel or not is made using the fuzzy rules applied

on the 8 neighbourhood. The pixels in the 8 neighbourhood of a pixel may be black or white.

The coordinates of its 8 neighbourhood are defined as given in Fig. 4.3



(i-1,j-1)	(i-1,j)	(i-1,j+1)
(i,j+1)	(i,j)	(i,j+1)
(i+1,j-1)	(i+1,j)	(i+1,j+1)

Fig.4.3: 8-neighbourhood of pixel (i, j)

Let I is the fuzzified image and I(i,j) represents the fuzzy value assigned to the pixel (i,j), where i & j represent the coordinates of the decision pixel. On the basis of number of white and black pixels in the neighbourhood, the total 30 rules are classified in 5 subclasses with the help of which we can decide whether a pixel is an edge pixel or not as follows:

Class 1: Rules with 3 black & 5 white pixels in neighbourhood shown in figure 4.4(a)

Rule 1: If I (i-1,j-1) AND I(i-1,j) AND I(i-1,j+1) AND I(i,j-1) AND I(i,j+1) are white and I(i+1,j-1) AND I(i+1,j) AND I(i+1,j+1) are black then (i,j) is edge pixel

Rule 2: If I(i-1,j-1) AND I(i-1,j) AND I(i-1,j+1) are black and I(i,j-1) AND I(i,j+1) AND I(i+1,j-1) AND I(i+1,j) AND I(i+1,j+1) are white then (i,j) is edge pixel

Rule 3: If I(i-1,j-1) AND I(i-1,j) AND I(i-1,j+1) are black and I(i,j-1) AND I(i,j+1) AND I(i+1,j-1) AND I(i+1,j) AND I(i+1,j+1) are black then (i,j) is edge pixel

Rule 4 If I(i-1,j-1) AND I(i,j-1) AND I(i+1,j-1) are black and I(i-1,j) AND I(i-1,j+1) AND I(i,j+1) AND I(i+1,j) AND I(i+1,j+1) are white then (i,j) is edge pixel

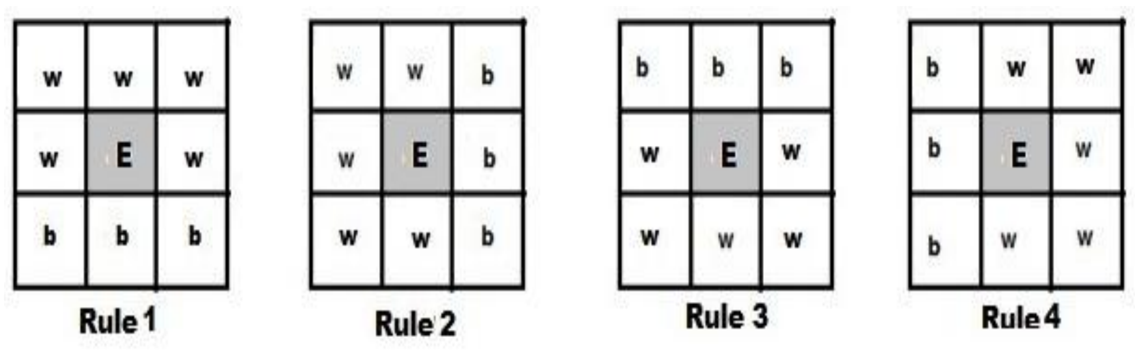


Fig. 4.4(a): Fuzzy Rules (class 1 rules)

Class 2: Rules with 4 black & 4 white pixels in neighbourhood shown in figure 4.4(b)

Rule 5: If I(i-1,j-1) AND I(i,j-1) AND I(i+1,j-1) AND I(i+1,j) are white and I(i-1,j) AND I(i-1,j+1) AND I(i,j+1) AND I(i+1,j+1) are black then (i,j) is edge pixel

Rule 6: If I(i-1,j-1) AND I(i,j-1) AND I(i+1,j-1) AND I(i-1,j) are white and I(i-1,j+1) AND I(i,j+1) AND I(i+1,j) AND I(i+1,j+1) are black then (i,j) is edge pixel

Rule 7: If I(i-1,j-1) AND I(i,j-1) AND I(i+1,j-1) AND I(i-1,j) are black and I(i-1,j+1) AND I(i,j+1) AND I(i+1,j) AND I(i+1,j+1) are white then (i,j) is edge pixel

Rule 8: If I(i-1,j-1) AND I(i,j-1) AND I(i+1,j-1) AND I(i+1,j) are black and I(i-1,j) AND I(i-1,j+1) AND I(i,j+1) AND I(i+1,j+1) are white then (i,j) is edge pixel

Rule 9: If I(i-1,j-1) AND I(i-1,j) AND I(i-1,j+1) AND I(i,j+1) are white and I(i,j-1) AND I(i+1,j-1) AND I(i+1,j) AND I(i+1,j+1) are black then (i,j) is edge pixel

Rule 10: If I(i-1,j-1) AND I(i-1,j) AND I(i-1,j+1) AND I(i,j-1) are white and I(i,j+1) AND I(i+1,j-1) AND I(i+1,j) AND I(i+1,j+1) are black then (i,j) is edge pixel

Rule 11: If I(i-1,j-1) AND I(i-1,j) AND I(i-1,j+1) AND I(i,j+1) are black and I(i,j-1) AND I(i+1,j-1) AND I(i+1,j) AND I(i+1,j+1) are white then (i,j) is edge pixel

Rule 12: If I(i-1,j-1) AND I(i-1,j) AND I(i-1,j+1) AND I(i,j-1) are black and I(i,j+1) AND I(i+1,j-1) AND I(i+1,j) AND I(i+1,j+1) are white then (i,j) is edge pixel

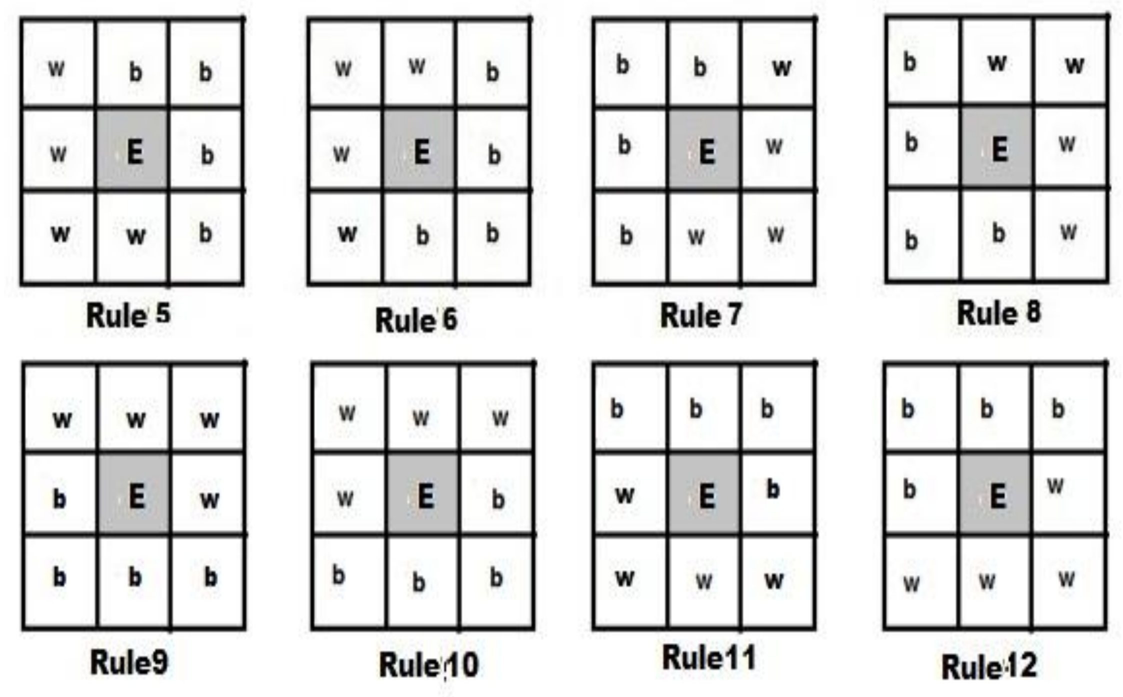


Fig. 4.4(b): Fuzzy Rules (class 2 rules)

Class 3: Rules with 5 black & 3 white pixels in neighbourhood shown in figure 4.4(c)

Rule 13: If I(i-1,j-1) AND I(i,j-1) AND I(i+1,j-1) AND I(i-1,j) AND I(i+1,j) are black and I(i-1,j+1) AND I(i,j+1) AND I(i+1,j+1) are white then (i,j) is edge pixel

Rule 14: If I(i-1,j-1) AND I(i-1,j) AND I(i-1,j+1) are white and I(i,j-1) AND I(i+1,j-1) AND I(i+1,j) AND I(i+1,j+1) AND I(i,j+1) are black then (i,j) is edge pixel

Rule 15: If I(i-1,j-1) AND I(i,j-1) AND I(i+1,j-1) are white and I(i-1,j) AND I(i-1,j+1) AND I(i,j+1) AND I(i+1,j+1) AND I(i+1,j) are black then (i,j) is edge pixel

Rule 16: If I(i,j-1) AND I(i-1,j-1) AND I(i-1,j) AND I(i-1,j+1) AND I(i,j+1) are black and I(i+1,j-1) AND I(i+1,j) AND I(i+1,j+1) are white then (i,j) is edge pixel

Rule 17: If I(i-1,j-1) AND I(i-1,j) AND I(i-1,j+1) AND I(i,j-1) AND I(i+1,j-1) are black and I(i,j+1) AND I(i+1,j) AND I(i+1,j+1) are white then (i,j) is edge pixel

Rule 18: If I(i-1,j-1) AND I(i,j-1) AND I(i+1,j-1) AND I(i+1,j) AND I(i+1,j+1) are black and I(i-1,j) AND I(i-1,j+1) AND I(i,j+1) are white then (i,j) is edge pixel

Rule 19: If I(i-1,j-1) AND I(i-1,j) AND I(i,j-1) are white and I(i,j+1) AND I(i+1,j-1) AND I(i+1,j) AND I(i+1,j+1) AND I(i-1,j+1) are black then (i,j) is edge pixel

Rule 20: If I(i,j-1) AND I(i+1,j-1) AND I(i+1,j) are white and I(i-1,j-1) AND I(i-1,j) AND I(i-1,j+1) AND I(i,j+1) AND I(i+1,j+1) are black then (i,j) is edge pixel

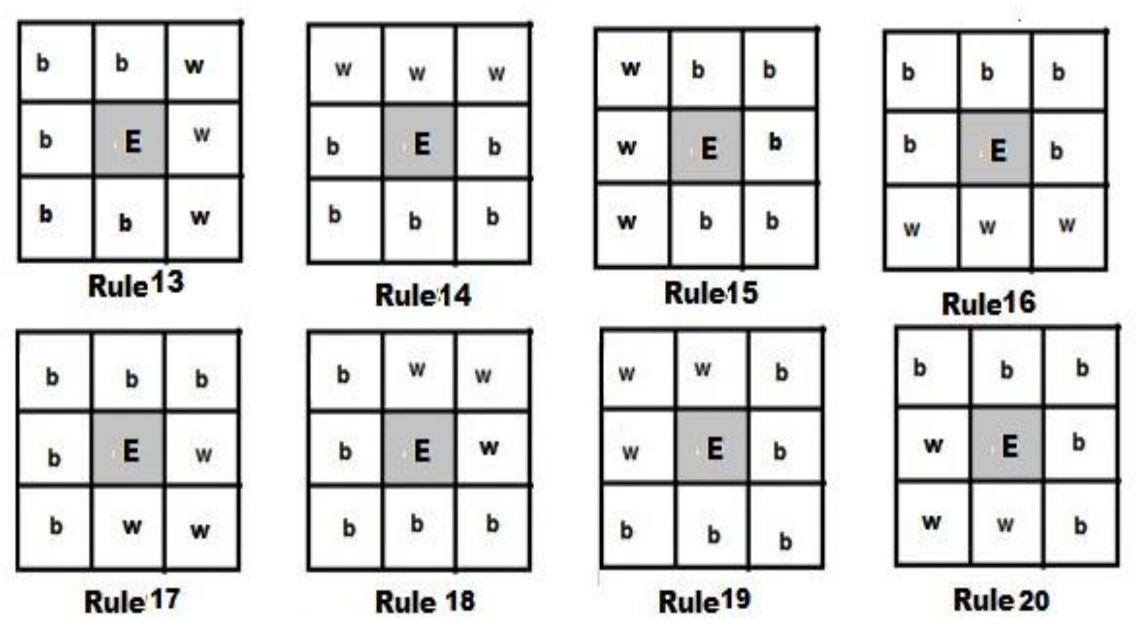


Fig. 4.4(c): Fuzzy Rules (class 3 rules)

Class 4: Rules with 6 black & 2 white pixels in neighbourhood shown in figure 4.4(d)

Rule 21: If I(i-1,j-1) AND I(i,j-1) AND I(i+1,j-1) AND I(i+1,j) AND I(i+1,j+1) AND I(i,j+1) are black and I(i-1,j) AND I(i-1,j+1) are white then (i,j) is edge pixel

Rule 22: If I(i-1,j-1) AND I(i,j-1) are white and I(i-1,j) AND I(i,j+1) AND I(i+1,j-1) AND I(i+1,j) AND I(i+1,j+1) AND I(i-1,j+1) are black then (i,j) is edge pixel

Rule 23: If I(i,j-1) AND I(i+1,j-1) are white and I(i-1,j-1) AND I(i-1,j) AND I(i-1,j+1) AND I(i,j+1) AND I(i+1,j+1) AND I(i+1,j) are black then (i,j) is edge pixel

Rule 24: If I(i-1,j-1) AND I(i-1,j) AND I(i-1,j+1) AND I(i,j-1) AND I(i+1,j) AND I(i+1,j-1) are black and I(i,j+1) AND I(i+1,j+1) are white then (i,j) is edge pixel

Rule 25: If I(i-1,j-1) AND I(i,j-1) AND I(i+1,j-1) AND I(i+1,j) AND I(i+1,j+1) AND I(i,j+1) are black and I(i-1,j) AND I(i-1,j+1) are white then (i,j) is edge pixel

Rule 26: If I(i-1,j-1) AND I(i-1,j) are white and I(i,j+1) AND I(i,j-1) AND I(i+1,j-1) AND I(i+1,j) AND I(i+1,j+1) AND I(i-1,j+1) are black then (i,j) is edge pixel

Rule 27: If I(i+1,j-1) AND I(i+1,j) are white and I(i-1,j-1) AND I(i-1,j) AND I(i,j-1) AND I(i-1,j+1) AND I(i,j+1) AND I(i+1,j+1) are black then (i,j) is edge pixel

Rule 28: If I(i-1,j-1) AND I(i-1,j) AND I(i-1,j+1) AND I(i,j-1) AND I(i,j+1) AND I(i+1,j-1) are black and I(i+1,j) AND I(i+1,j+1) are white then (i,j) is edge pixel

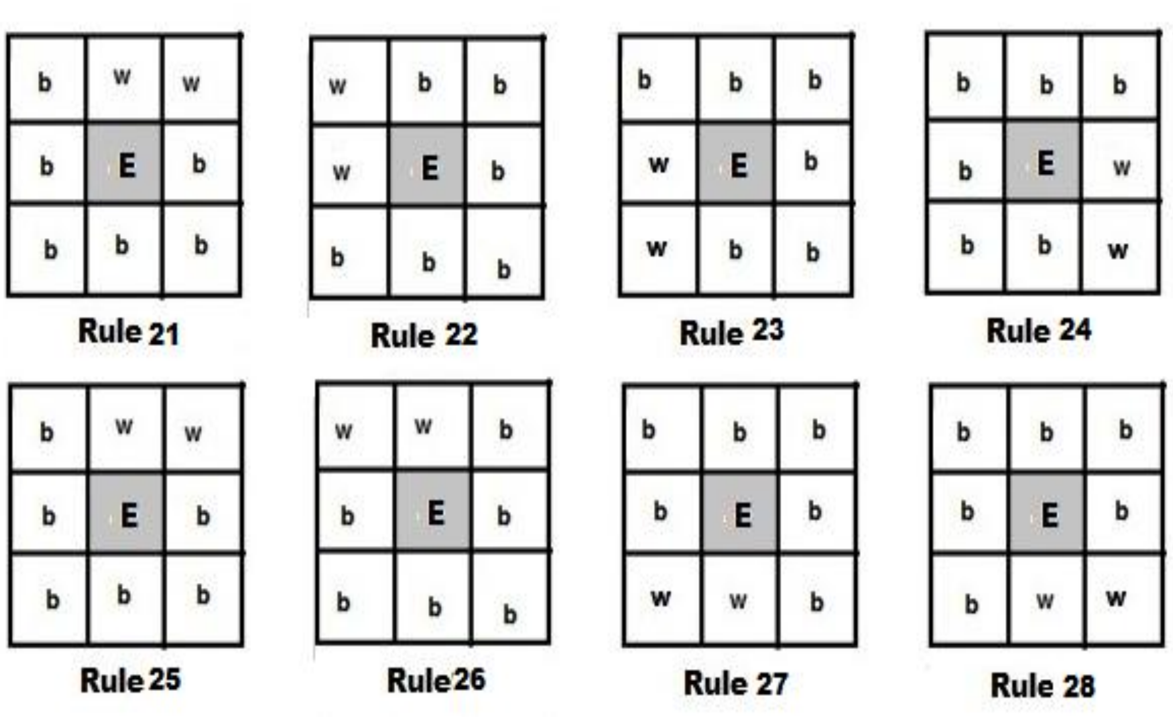


Fig. 4.4(d): Fuzzy Rules (class 4 rules)

Class 5: Rules with all black and all white pixels in neighbourhood (to remove noise) shown in figure 4.4(e)

Rule 29: If I(i,j-1) AND I(i+1,j-1) AND I(i-1,j-1) AND I(i,j+1) AND I(i-1,j) AND I(i-1,j+1) AND I(i,j+1) AND I(i+1,j+1) AND I(i+1,j) are black then (i,j) is non edge pixel

Rule 30: If I(i-1,j-1) AND I(i-1,j) AND I(i-1,j+1) AND I(i,j-1) AND I(i+1,j) AND I(i+1,j-1) AND I(i,j+1) AND I(i+1,j+1) are white then (i,j) is non edge pixel

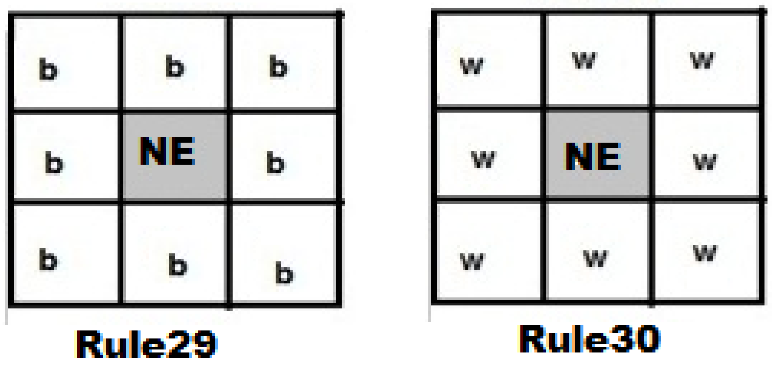


Fig. 4.4(e): Fuzzy Rules (class 5 rules)

4.3 False Edge Removal

In the proposed method some false edges also emerge out along with the true edges. Reason for these false edges is explained as follows. There may be case where there are six or more than six pixels having the intermediate range of intensity values in the 8 neighbourhood of a pixel. The difference in intensity of these pixels is although very low, still these pixels are classified among black and white because of the membership functions defined which gives rise to the false edges according to the defined fuzzy rules. Hence to remove these false edges, we simply consider the pixel as non-edge pixel if six or more than six pixels in its neighbourhood are in the intermediate intensity range. The chosen intermediate range is [117...137]. This range is found out experimentally and is giving out the best results.

4.4 Edge Improvement

The proposed method still not able to detect some of the edge pixel due to the following reason:

1) The pixels above 128 are classified as white pixels. The pixels with intensity approaching 128 are also white and pixel approaching 255 are also white. For e.g., if the intensities of 8 neighbourhood pixels are 129, 137,131,205,128,211, 230 and 245 where the intensity difference is very high still they all are considered as white pixels.

2) The pixels below 128 are classified as black pixels. The pixels with intensity approaching 128 are also black and pixel approaching 0 are also black. For e.g., if the intensities of 8 neighbourhood pixels are 127, 97, 101, 5, 1, 11, 120 and124 where the intensity difference is very high still they all are considered as black pixels.

In this situation edge pixels remain undetected by the defined approach. When the 8 neighbourhood of a pixel is satisfying any of the above condition, a new membership function defined and applied to define black and white membership for the pixel.

For the black set the membership function is given as

$$\mu'_{black}(x) = \left\{ \begin{array}{cc} 1 & x < 141 \\ \dfrac{255 - x}{255 - 141} & 141 \le x \le 250 \end{array} \right\} \qquad (18)$$

For the white set the membership function is given as

$$\mu'_{white}(x) = \left\{ \begin{array}{cc} 0 & x < 141 \\ \dfrac{x - 141}{250 - 141} & 141 \le x \le 250 \end{array} \right\} \qquad (19)$$

.

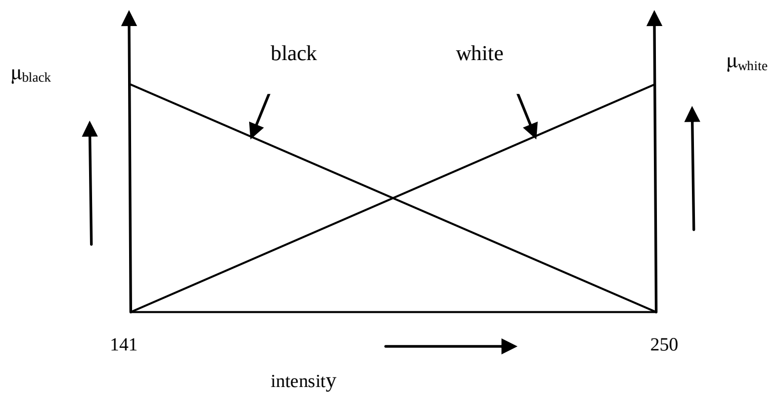


Fig. 4.5: Membership Function for intensity range [141…250]

Now we again find the fuzzy set to which the pixel belong according to the new membership function. The previously defined fuzzy rules are applied to find whether the pixel is edge pixel or not.

4.5 Algorithm

The algorithm for the proposed approach comprises of following steps:

Step 1: Input gray scale image of size m x n.

Step 2: Scan image pixel by pixel considering its 8-neighbourhood.

Step 3: Count the number of the pixels in 8-neighborhood of the scanned pixel in the range of 117-137, denoted by Nint and the number of the pixels in 8-neighborhood of the scanned pixel in the range of 141-250, denoted by Nwhite.

Step 4: If Nint< 6, the scanned pixel is non-edge and go to step 2.

Step 5: If Nwhite <8, Find membership of entire 8-neighboorhood pixel of the scanned pixel's membership by (16) and (17) for black and white fuzzy set.

Step 6: Else, find membership of entire 8-neighboorhood pixel of the scanned pixel's membership by equation (18) and (19) for black and white set.

Step 7: Find maximum of black and white membership for each pixel and assign the corresponding fuzzy variable to the pixel.

Step 8: Apply fuzzy rules stated to find whether the pixel is edge pixel or not.

Step 9: If all pixel not scanned, go to step 2.

Step 10: End.

4.6 Pseudo-code

```
FuzzyEdge (Image Img)
// m & n are dimensions of image Img

For  i=1 to m;
      For  j=1 to  n;
            For (x,y) in 8-Neighbourhood of  (x,y)
                  If (Img(x,y) >= 117  AND Img(x,y) =< 137)
                        Nint = Nint + 1;
                    End
                  Else If (Img(x,y) >= 140  AND Img(x,y) =<250)
          Nwhite = Nwhite + 1;
                 End
            End for
            If ( Nwhite == 8)
                  For (x,y) in 8-neighbourhood of (i,j)
                  Compute $\mu_{black}(x,y)$ and $\mu_{white}(x,y)$ using 18 and 19 respectively.
                End for
            Else
                  For (x,y) in 8-neighbourhood of (i,j)
           Compute $\mu_{black}(x,y)$ and $\mu_{white}(x,y)$ using 16 and 17 respectively.
                  End for
            End
            If Nint >= 6, pixel is non-edge
                 Break;
            Else
                  Apply fuzzy rules to find whether (i, j) is edge pixel or not
            End
      End for
End for
```

Chapter 5

EXPERIMENTAL RESULTS

5.1 Comparison with standard edge detectors

The proposed method has been implemented in MATLAB 9.7.0.471 and it is run on Core i3, 2 GHz processor with 3 GB RAM for detecting the edge map in gray scale images. No pre-processing is required prior to the application of this algorithm.

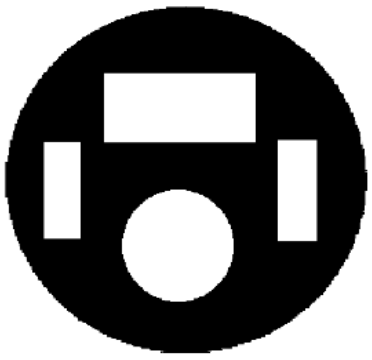
To demonstrate the efficiency of the proposed approach, we carried out computer experiments on gray-level images. We selected a few standard images which are ‘The Lena’, ‘Baboon’, ‘Cameraman’, ‘Peppers’ , ‘PillSet’. The resolution of all images is 8-bit per pixel. Along with the standard images, one of the simple test image used in A.Alshennawy[7] is also selected. The above test images are shown in fig. 5.1.

The proposed system was tested with above mentioned images, its performance being compared to that of the classical operators (Sobel, Prewitt, Canny) and the proposed method. The edge maps of the images using Sobel operator and Canny operator are found using the image processing toolbox in Matlab. The edge maps visually imply that the edges found by the proposed approach are significant and few results are better than most of the standard edge detectors and also the A.Alshennawy[7] edge detector.

The result of these test images using the classical operators, A.Alshennawy[7] method and the proposed approach are shown in Fig. 5.2-5.7.

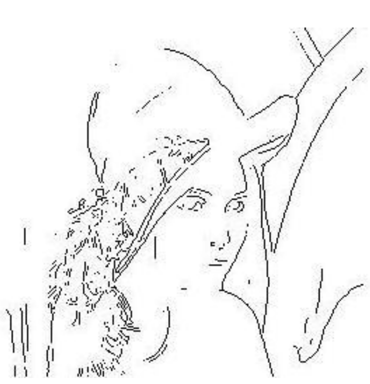
(a) (b)

(c) (d)

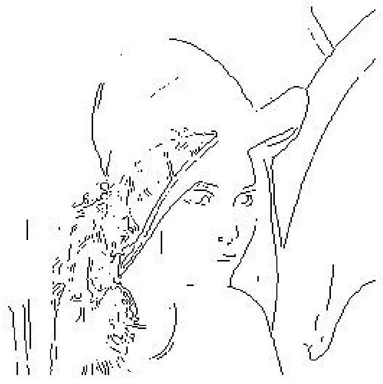


(e) (f)

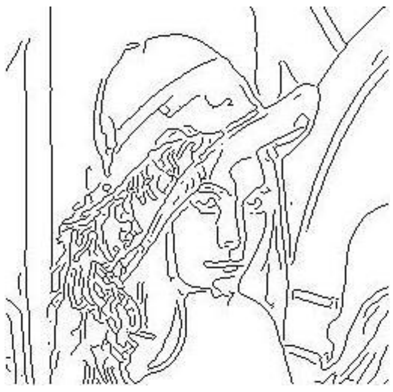
Fig. 5.1 Original Test Images (a) The Lena (b) The Cameraman (c) Baboon (d) Peppers (e) Pillset (f) Simple test image



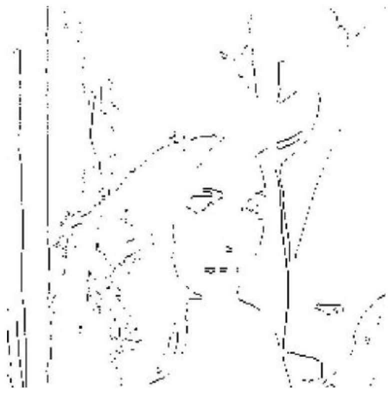
(a)



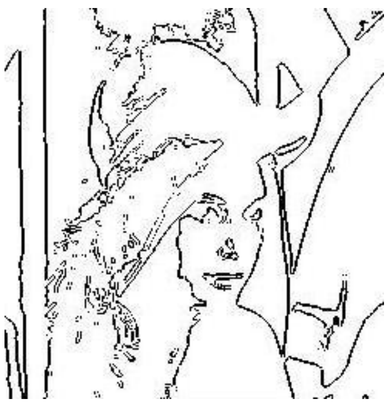
(b)



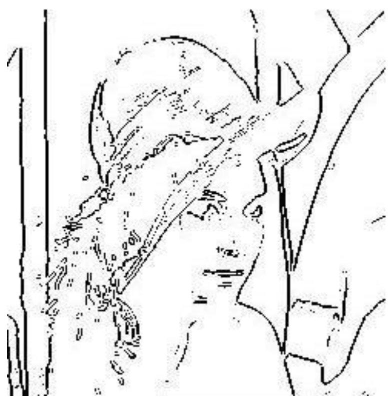
(c)



(d)

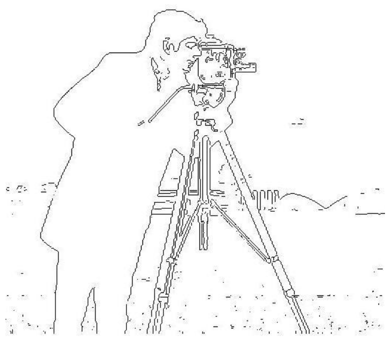


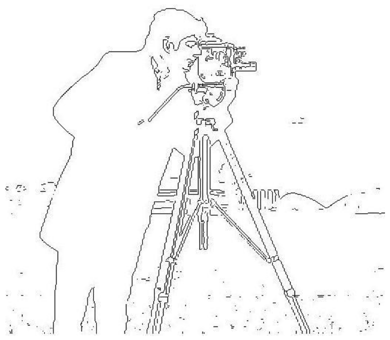
(e)



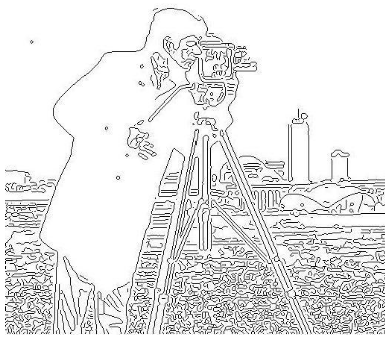
(f)

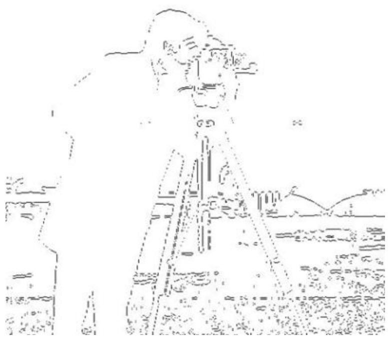
Fig. 5.2 Edge map of Lena image by (a) Sobel operator (b) Prewitt operator (c) Canny operator (d) A Alshennawy *et al.*[7] Edge Detector (e) Proposed approach (f) modification in proposed approach



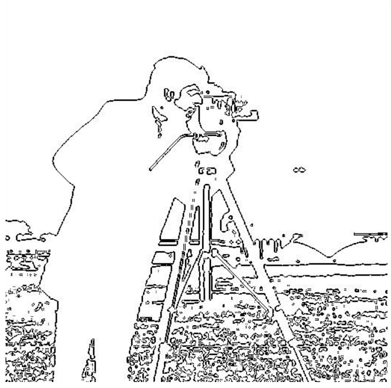


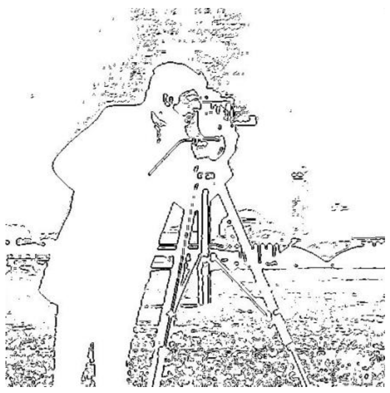
(a) (b)





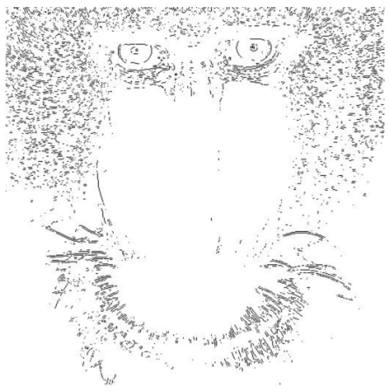
(c) (d)



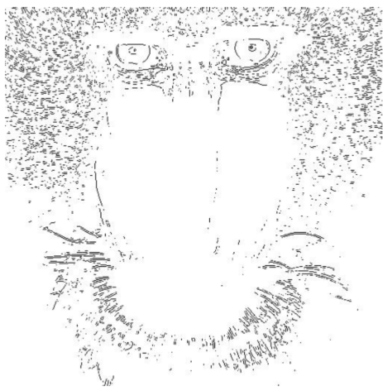


(e) (f)

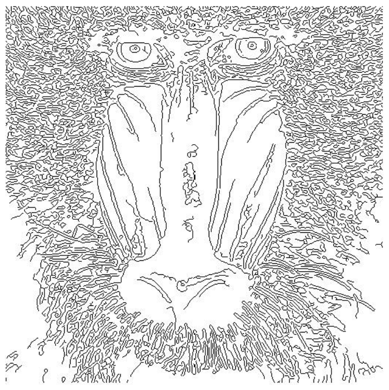
Fig. 5.3 Edge map of Cameraman image by (a) Sobel operator (b) Prewitt operator (c) Canny operator (d) A Alshennawy *et al.*[7] Edge Detector (e) Proposed approach (f) modification in proposed approach



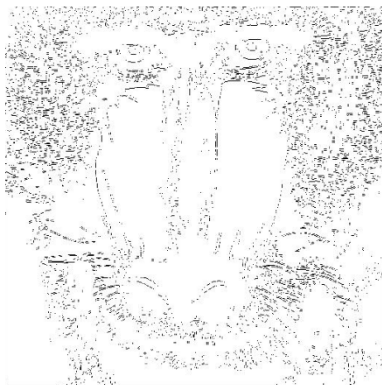
(a)



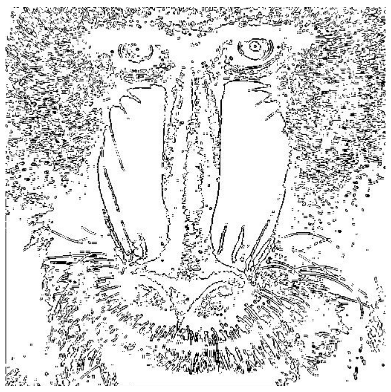
(b)



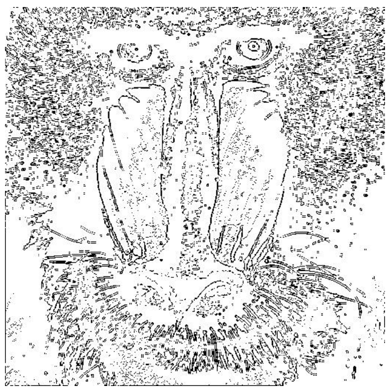
(c)



(d)

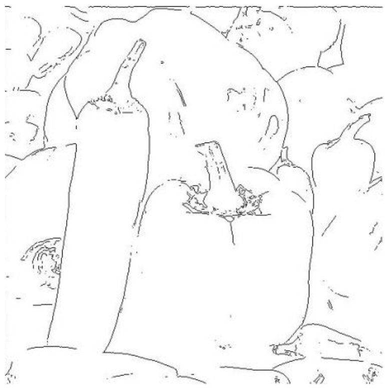


(e)

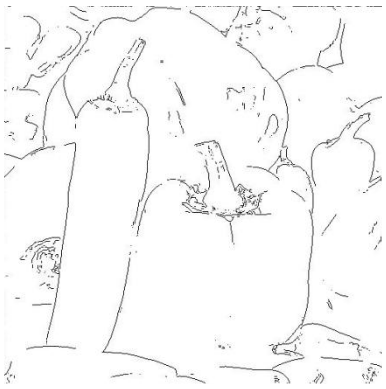


(f)

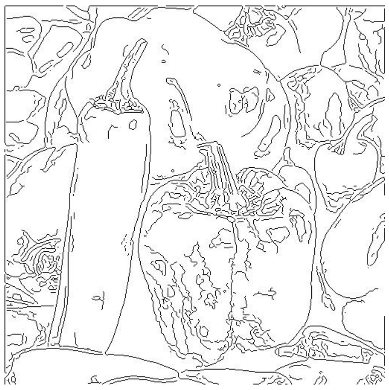
Fig. 5.4 Edge map of Baboon image by (a) Sobel operator (b) Prewitt operator (c) Canny operator (d) A Alshennawy *et al.*[7] Edge Detector (e) Proposed approach (f) modification in proposed approach



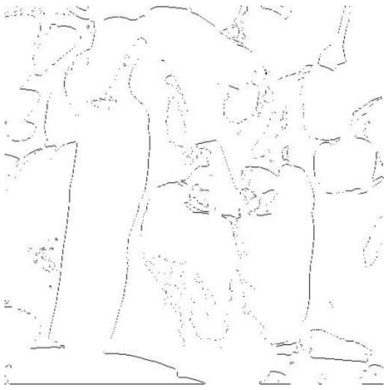
(a)



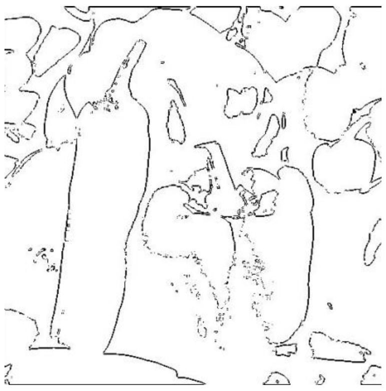
(b)



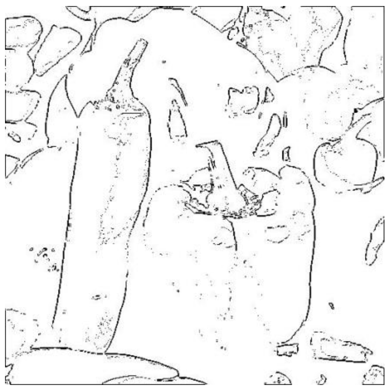
(c)



(d)



(e)



(f)

Fig. 5.5 Edge map of Peppers image by (a) Sobel operator (b) Prewitt operator (c) Canny operator (d) A Alshennawy *et al.*[7] Edge Detector (e) Proposed approach (f) modification in proposed approach

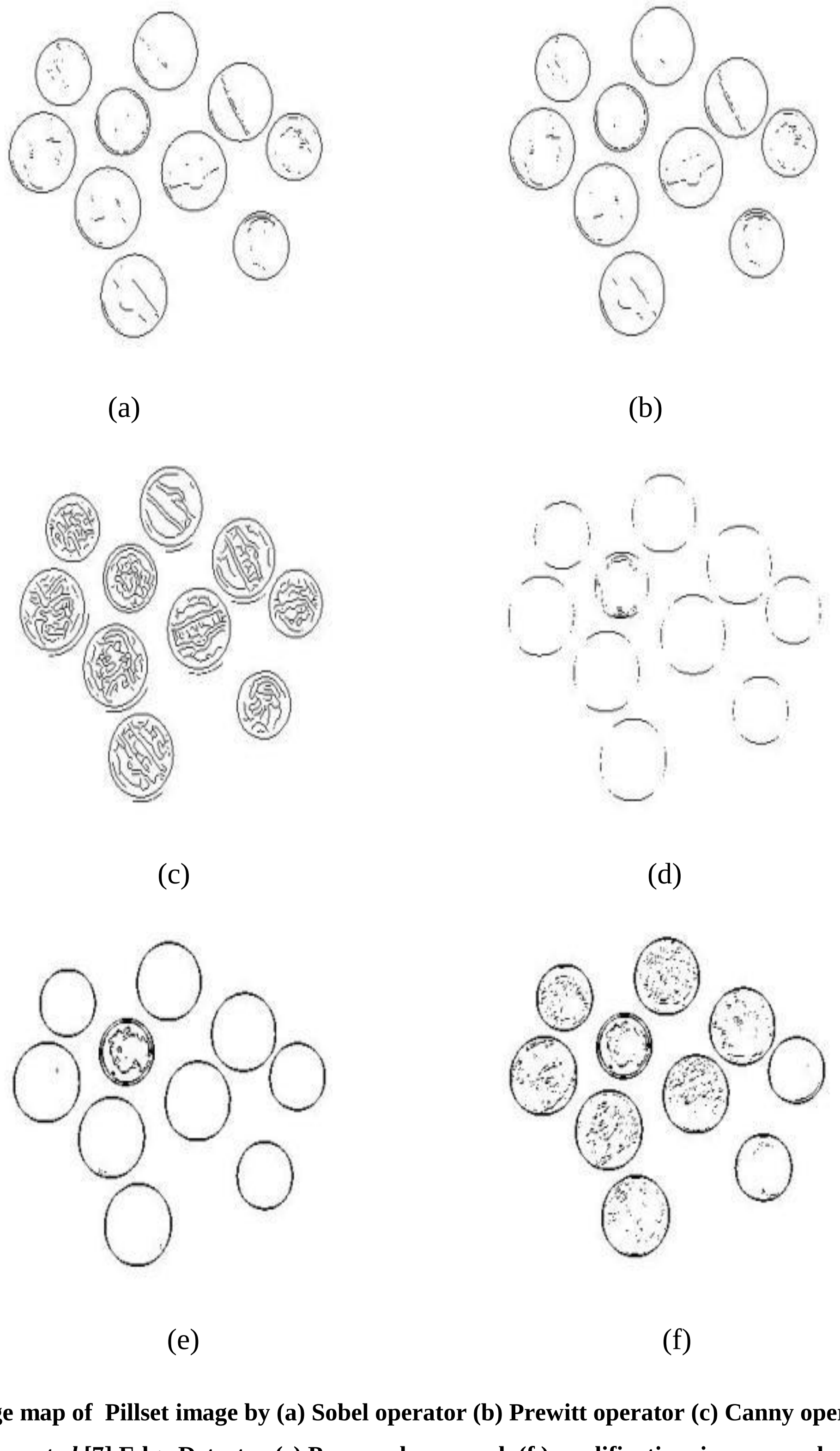
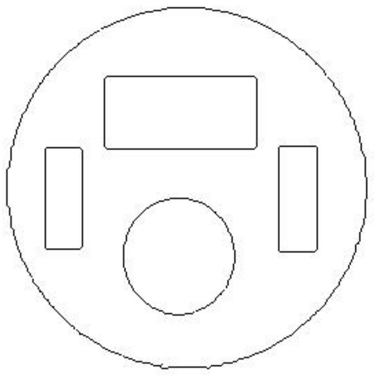
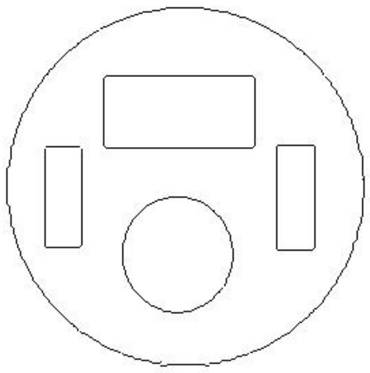
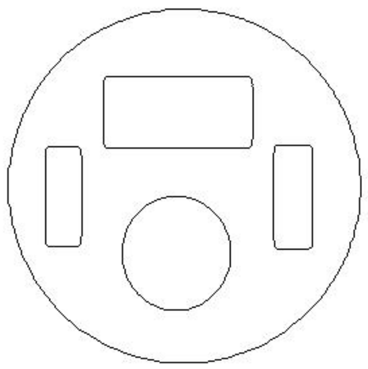


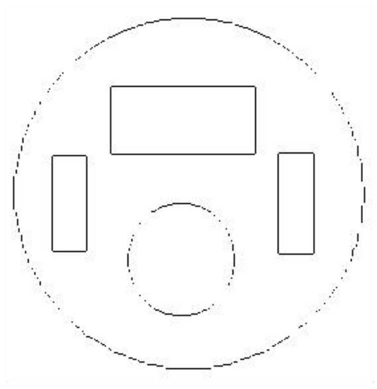
Fig. 5.6 Edge map of Pillset image by (a) Sobel operator (b) Prewitt operator (c) Canny operator (d) A Alshennawy *et al.*[7] Edge Detector (e) Proposed approach (f) modification in proposed approach



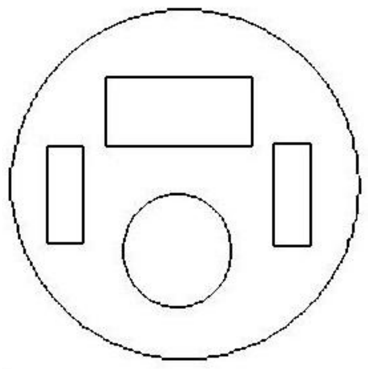


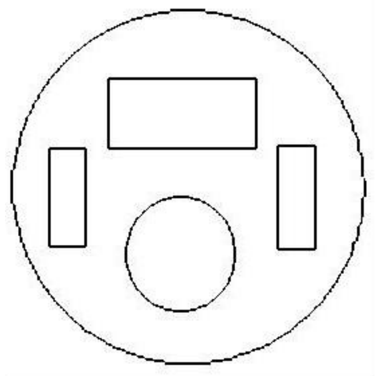
(a) (b)





(c) (d)





(e) (f)

Fig. 5.7 edge map of test image by (a) Sobel operator (b) Prewitt operator (c) Canny operator (d) A Alshennawy *et al.*[7] Edge Detector (e) Proposed approach (f) modification in proposed approach

5.2 Quantitative Analysis

To evaluate the performance of the method, a quantitative measure named Shannon entropy value is calculated for each of the images. Shannon's entropy gives the indefiniteness in an image and is calculated as:

$$H(I) = -\sum_{i=0}^{L-1} p_i \log p_i$$

Where I stand for image whose entropy is to be measured, p_i is the frequency of occurrence pixel with intensity i,vL is number of intensity levels in image.

Table I shows the entropy values for the various test images by applying several edge detectors like Sobel, Canny etc. Higher the value of the entropy higher is the information content. However, a very large value of entropy reflects high noise content or double edges. The Canny edge detector produce double edges and therefore the entropy values with this method is higher than those obtained with the proposed method in 3 out of 6 images. The higher entropy value with the proposed approach is due to the noise but the proposed method gives significant results without the use of derivatives and thresholding and also is computationally simple. The other edge detectors namely Sobel and A.Alshennawy *et al.* [7] however, give very less edge information; therefore the entropy values using these methods are lesser than those of our method

TABLE I

ENTROPY VALUES OF DIFFERENT EDGE DETECTOR FOR MULTIPLE IMAGES.

Image	Sobel	Canny	A.Alshennawy *et al.*[7]	Proposed
Lena	0.6327	0.9730	0.3756	0.9629
Peppers	0.4365	0.8596	0.1461	0.6506
Coins	0.4821	0.9201	0.1239	0.7935
Baboon	0.7866	1.4009	0.1434	1.5563
Cameraman	0.4026	0.9245	0.5057	0.9551
Test Image	0.3300	0.3258	0.3474	0.5278

Chapter 6
CONCLUSIONS

This paper provides a new methodology to detect edges which is quite simple as compared to other edge detection methodology available so far. With addition of more rules, proposed method has significantly improved the edge detection as compared to the technique present in [7]. Our method not only detects edges in simple images but also detect the edges in complex gray scale images.

In order to obtain more edges, the modification that is suggested is also helpful. The range of intensity for which membership is to be redefined can be found out experimentally so as to get maximum edges. But while adding more edges, there are chances of getting noise. Here, intensity range [141…250] gives better results with more edges and lesser noise. This range has been found out experimentally.

The proposed method does not use any derivative or complex calculation and is robust in finding edges by the use of simple fuzzy rules; however detect some false edges also. The designed fuzzy rules are an attractive solution to improve the quality of edges as much as possible. Clearly, the proposed method provides a better edge detection results than the previous methods do for the illustrated images in relatively less time.

The proposed approach has been designed only to detect images for gray scale images. But results are significant for gray scale images using a relatively simple derivative free and fast edge detector.

REFERENCES

[1]R.C. Gonzalez, R.E. Woods, Digital Image Processing, Addison-Wesley, New York, 1992.

[2] J.F. Canny, ‘A computational approach to edge detection’, IEEE Trans. Pattern Analysis Mach. Intell. 8 (1986), pp 679–698.

[3] L. A. Zadeh, “Fuzzy Sets”, Information and Control”, 1965, pp. 338-353.

[4] Pal S K, King R A., ‘On Edge detection of X-ray images using fuzzy sets’, IEEE Trans. Patt. Anal. and Machine Intell. 1983, 5(1):69-77.

[5] Zhang Jinping; Lian Yongxiang; Dong Linfu; Zhao Xueguang, Liu Jie; “A New Method of Fuzzy Edge Detection Based On Gauss Function”, Proceedings of 2nd International Conference on Computer and Automation Engineering, Singapore, Feb 26-28, 2010, pp. 559 – 562.

[6] T. Law, H. Itoh, H. Seki, Image filtering, edge detection, and edge tracing using fuzzy reasoning, IEEE Trans. Pattern Analysis Mach. Intell. 18 (1996) 481-491.

[7] Abdallah A. Alshennawy, and Ayman A. Aly ‘Edge Detection in Digital Images Using Fuzzy Logic Technique’, World Academy of Science, Engineering and Technology 51 2009, pp 178-186

[8] Begol, Moslem and Maghooli, Keivan, ‘Improving Digital Image Edge Detection by Fuzzy Systems’ World Academy of Science, Engineering and Technology 81 2011, pp 76-79

[9] Er Kiranpreet Kaur, Er Vikram Mutenja, Er Inderjeet Singh Gill ‘Fuzzy Logic Based Image Edge Detection Algorithm in MATLAB’ 2010 International Journal of Computer Applications (0975 - 8887) Volume 1 – No. 22

[10] Lily Rui Liang, Carl G. Looney, ‘Competitive fuzzy edge detection’, Applied Soft Computing, Volume 3, Issue 2, September 2003, pp 123–137 x

[11] R. Wang, L. Gao, S. Yang, and Y. Liu, "An Edge detection method by combining fuzzy logic and neural networks," Machine Learning and Cybernetics, 7(2005), Guangzhou, China, 18-21 Aug 2005, pp 4539-4543.

[12] S. Sinaie, A. Ghanizadeh , and S.M. Shamsuddin , E.M. Majd 'A Hybrid Edge Detection Method Based on Fuzzy Set Theory and Cellular Learning Automata' 2009 International Conference on Computational Science and Its Applications, june 2009 Suwon ,Korea.

[13] Tizhoosh, H.R.; "Fast fuzzy edge detection," Fuzzy Information Processing Society, 2002. Proceedings. NAFIPS. 2002 Annual Meeting of the North American, vol., no., pp. 239- 242, 2002

[14] C.Naga Raju, O.Rama Devi, Sharada Mani, Sanam Nagendram, 'An Improved Ant Colony Optimization Technique by using Fuzzy Inference Rules for Image Classification and Analysis' International Journal of Advanced Engineering & Application, Jan. 2010.

[15] Khalid, N.E.A.; Manaf, M.; Aziz, M.E.," Fusion of Fuzzy Heuristic and Particle Swarm Optimization as an edge detector", Proceedings of International Conference on Information Retrieval & Knowledge Management, Shah Alam, Selangor, March 17-18, 2010, pp. 250 – 254.

[16] Om Prakash Verma, Madasu Hanmandlu, Puneet Kumar, Sidharth Chhabra, Akhil Jindal, A novel bacterial foraging technique for edge detection, Pattern Recognition Letters, Volume 32, Issue 8, 1 June 2011, pp 1187-1196.