International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 10 Issue: 11 | Nov 2023. www.irjet.net p-ISSN: 2395-0072

Conceptual and Practical Examination of Several Edge Detection Strategies

Shalini[1], Dr. N K Singh[2], Rahila Khan[3]

[1]*Assistant Professor, Dept. of CSE, ITM, GIDA Gorakhpur, U.P., India*
[2] *Professor, Dept. of CSE, ITM, GIDA Gorakhpur, U.P., India*
[3]*Assistant Professor, Dept. of CSE, KIOT, Kanpur, U.P., India*

Abstract — *Image segmentation is a method of segregating the image from the background and performs its content comprehension. This can be done by detecting the edges present in the scene. Edge detection is thus a vital step of identifying and locating sharp discontinuities in an image. Vast research has been done on various Edge detection techniques to significantly reduce the amount of data to be processed, filtering out useless information, while preserving the important structural properties in an image. This however leads to a lot of challenges in real world such as changes in lightening conditions, dynamic background, noise, missing existing edges, false or dislocated edge detection. The proposed paper would present a complete survey of different edge detection techniques that are available. Depending upon the different performance criterion, it would also offer idea for selecting the appropriate tactic for edge detection. Applying the most used experiential methods such as Sobel, Prewitt, Roberts, Canny, Laplacian of Gaussian (LoG), in this paper we would provide with a comparative assay on variegated edge detection tactics. A complete tabulation of the different segmentation techniques analyzed would also be presented at the end of the paper.*

Keywords—Image Segmentation, Edge, Edge Detection, MATLAB

1. INTRODUCTION

In the field of Computer Vision and Digital Image Processing, an image needs to be analyzed to identify the various objects present in the scene [2]. This can be done by segmenting the entire image into its constituent using image segmentation. Image Segmentation is the process of partitioning a digital image into multiple regions or sets of pixels [1] [3] [18]. Actually, partitions are different objects in image which have the same texture or color. The result of image segmentation is a set of regions that collectively cover the entire image, or a set of contours extracted from the image. All of the pixels in a region are similar with respect to some characteristic or computed property, such as color, intensity, or texture. Adjacent regions are significantly different with respect to the same characteristics. The boundaries of object surfaces in a scene often lead to oriented localized abrupt and sharp changes in intensity of an image, called edges. Thus, Edge is a boundary between two homogeneous regions. Edges characterize boundaries and are therefore a problem of fundamental importance in image processing. Edge detection refers to the process of identifying and locating such edges in an image. Image Edge detection significantly reduces the amount of data and filters out useless information, while preserving the important structural properties in an image. The Classical methods of edge detection involve smoothing differentiation of the image [2]. Smoothing results in loss of information and differentiation is an ill-conditioned problem. So it becomes difficult to design a general edge detection algorithm which performs well in all contexts and captures the requirements of subsequent processing stages. Consequently in the history of digital image processing, a variety of edge detectors have been devised which differ in their mathematical and algorithmic properties. This observation combined with a commonly held belief that edge detection is the first step in image segmentation, has fueled a long search for a good edge detection algorithm to use in image processing, thus leading to a steady stream of edge detection algorithms published in the image processing journals over the last three decades [13]. Since edge detection is a principal area of research in image processing for object detection, it is crucial to have a good understanding of edge detection algorithms. This paper is an account of some widely used approaches to detect edges for segmentation.

The paper is organized as follows. Section 1 is for the purpose of providing some information about edge detection for Image Segmentation. Section 2 is focused on showing general approach for edge detection and focused on classifying it two broad categories. Section 3 presents the study and analysis of various edge detection techniques. In section 4 the visual comparisons of various edge detection techniques have been done by developing software in MATLAB 7.0. In Section 5 the advantages and disadvantages of various edge detection techniques are discussed. Section 6 provides with the conclusion reached by analysis and visual comparison.

2. EDGE DETECTION TECHNIQUE

2.1 Description

The generic method of edge detection involves convolving the image with an operator (a 2-D filter), which is constructed to be sensitive to large gradients in the image while returning values of zero in invariable regions. Large number of edge detection operators is available; each designed to be sensitive to certain edge type. While choosing a particular edge detection operator, variables that need to be considered include Edge orientation, Noise environment and Edge structure. The operator once selected then needs to be operated on the image to detect point that might lie in an edge. Such points are known as the edge points.

An extensive collection of objectives should be defined for the computation of edge points. The success of the approach used depends on the comprehensiveness of the objectives. These objectives must be decisive enough to demarcate the desired behavior of the detector while making as minimal assumptions as possible about the possible form of the solution. Operators can be emended to look for horizontal, vertical, or diagonal edges. The geometry of the operator ascertains a distinct direction in which it is most sensitive to edges. Detection and localization criteria for a class of edges are defined, and mathematical forms are presented for these criteria functional on the operator impulse response. A consequent criterion is then added to ensure that the detector has a single response to an edge. Numerous criteria in numerical optimization are used to derive detectors for several prevalent image features that include step edges. Once the analysis of step edges is complete, a natural uncertainty between detection and localization performance is established. Based on this establishment, a single operator shape is designed which is optimal at any scale. This simple detector can then be extended using operators of several widths to cope with variegated signal-to-noise ratios in the image. Edge detection is prohibitive in noisy images, since both the noise and the edges comprises of high frequency contents. One possible solution would be to reduce the noise by smoothening. This would however result in blurred and distorted edges. The size of operators is kept large when applied on noisy images, so that sufficiently large numbers of pixels are averaged to discount localized noisy pixels. However, the detected edges in this case are less accurately localized.

Also in real life scenes, the chances of an edge to have a step change in intensity are very rare. The reasons may be refraction or poor focus on object while capturing its image. The operator needs to be chosen to be responsive to such a gradual change in object boundaries in those cases. Otherwise problems of false edge detection, missing true edges, edge localization, high computational time and problems due to noise etc might arise. Therefore, speculative comparison of variegated edge detection techniques is performed and the performance of the different techniques is analyzed in different conditions.

2.2 Types

There are various ways to perform edge detection, the majority of which may be grouped into two broad categories:

(1) Gradient based Edge Detection:
The gradient method uses the maximum and minimum values of the first order derivative for detecting the edges.

(2) Laplacian based Edge Detection:
The Laplacian method uses the zero crossings in second order derivative of the image to detect edges.

Assume the intensity graph of an edge shown by the jump in figure – 1.

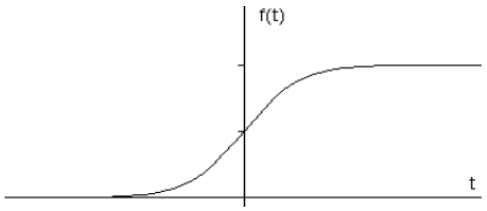

Fig – 1: Intensity graph of an edge

If we calculate the gradient of the signal below (which, in 1D, is just the first order derivative with respect to t) we get the graph as shown in figure – 2. As clear from the graph, maximum value is located at the center of the edge in original signal.

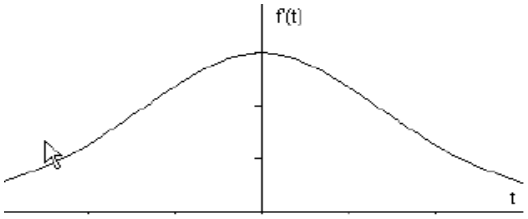

Fig – 2: Intensity graph for gradient of the signal

This is the characteristic feature of gradient method of edge detection. In this, a pixel is said to be an edge pixel if and only if the value of its gradient is greater than a specified threshold. As already stated, edges have higher pixel intensity values as those surrounding it. So once a threshold value is specified, the gradient value of the pixel is compared to the threshold value and it is marked as an edge pixel if the value exceeds the threshold. Moreover, when the first derivative for a particular pixel is at a maximum, its second order derivative is zero. Consecutively, edge pixels in an image may be found out by locating the zeros in second order derivative. This method comes under the category of Laplacian and can be graphically depicted as in figure – 3.

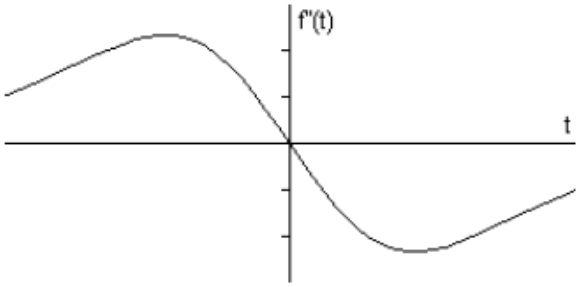

Fig – 3: Laplacian Method

3. EDGE DETECTION FOR IMAGE SEGMENTATION

3.1 Introduction

The edge detection techniques firstly convert image to edge image using the abrupt changes that occur in the intensity of the image due to edges and boundaries. An edge image is achieved without performing any changes in actual physical qualities of the original image [12] [16]. Objects in an image might consist of different color levels. As already stated, an edge in an image is a significant local change in the image intensity, usually associated with a discontinuity in either the image intensity or the first derivative of the image intensity. These discontinuities in the image intensity can be either Step edge, in which the image intensity abruptly changes from one value on one side of the discontinuity to a different value on the opposite side, or a Line Edges, where the image intensity abruptly changes value but then returns to the starting value within some short distance [3]. However, Step and Line edges are quite rare in real images. Because of reflection effects, limitations of image capturing device, low frequency components or the smoothing introduced by most sensing devices, sharp discontinuities rarely exist in real life scenes .Thus, step edges become Ramp Edges and Line Edges are replaced by Roof edges, where intensity changes are not instantaneous but occur over a finite distance [15]. Exemplification of these edge shapes are shown in figure – 4.

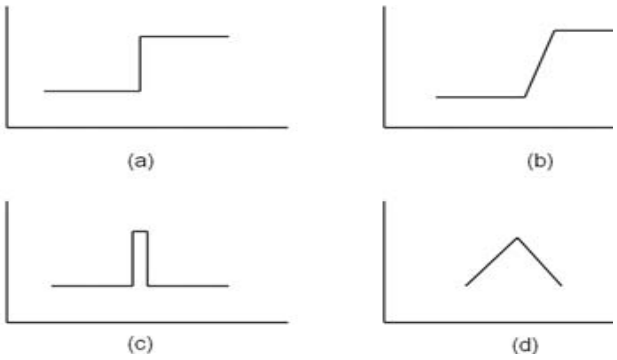

Fig – 4: Type of Edges: (a) Step Edge (b) Ramp Edge (c) Line Edge (d) Roof Edge

3.2 Steps in Edge Detection

Generally, the edge detection process comprises of the three common steps namely Filtering, Enhancement and Detection. The brief description of these steps in edge detection is as follows:

3.2.1. Filtering

Often image is corrupted by random variations in the intensity values of the image, which is referred to as the noise. Some of its common types being salt and pepper, impulse, Gaussian, uniform, erlang (gamma) noise. Filtering is done to remove these noises from the image, however more filtering to reduce noise results in a loss of edge strength [17]. Hence, there is a trade-off between edge strength and noise reduction.

3.2.2. Enhancement

This step is performed to facilitate the edge detection process. It emphasizes those pixels where there is a considerable change in local intensity value. It is often done by calculating the local gradient magnitude [14].

3.2.3. Detection

There may be many points in the image having non-zero values for gradient that do not qualify to be the edge pixels. So some processing should be done to determine which of such points are edge points and which are not. Most commonly, thresholding is done to achieve this purpose. [13]

3.3. Edge Detection Methods

There are five types of most frequently used edge detection methods. These are (1) Robert's Cross Edge Detection Method, (2) Sobel Edge Detection Method, (3) Prewitt Edge Detection Method, (4) Laplacian of Gaussian Edge Detection Method, and (5) Canny Edge Detection Method [18]. The details of methods as follows:

3.3.1. Robert's cross operator: [20]

The Roberts Cross operator is simple, quick to compute and performs fast 2-D spatial gradient measurement on an image. Pixel values at each point in the output 2-D intensity image represent the estimated absolute magnitude of the spatial gradient of the input image at that point. This operator consists of a pair of 2×2 convolution kernels which is shown in figure – 5. One kernel can simply be obtained by rotating the other by 90° [4].

+1	0
0	-1

Gx

0	+1
-1	0

Gy

Fig – 5: Robert's Cross Operator Mask

These kernels are designed in such a way that is responds maximally to the edges running at 45° to the pixel grid, each kernel for one of the two perpendicular orientations. The kernels can initially be applied to the input image separately, producing separate measurements of the gradient component in each orientation (let it be G_x and G_y). These are then combined to calculate the absolute magnitude of the gradient at each point and the orientation of that gradient. The gradient magnitude is determined by the formula:

$$|G| = \sqrt{Gx^2 + Gy^2}$$

However, most commonly an approximate magnitude is computed using:

$$|G| = |Gx| + |Gy|$$

which is much faster to compute.

The angle of orientation of the edge (relative to the pixel grid orientation) is determined by:

$$\theta = \arctan(Gy / Gx) - 3\pi/4$$

3.3.2. Sobel Operator [20]

The Sobel operator comprises of a pair of 3×3 convolution kernels as shown in figure – 6. As in case of Robert's Cross, in this technique also one kernel can be obtained simply by rotating the other by 90°.

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

Gx

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

Gy

Fig – 6: Sobel Operator Mask

These kernels are designed in such a way that is responds maximally to the edges running vertically and horizontally relative to the pixel grid, one kernel for each of the two perpendicular orientations. The kernels can initially be applied to the input image separately, producing separate measurements of the gradient component in each orientation (let it be Gx and Gy). These are then combined to calculate the absolute magnitude of the gradient at each point and the orientation of that gradient. The gradient magnitude is determined by the formula:

$$|G| = \sqrt{Gx^2 + Gy^2}$$

However, most commonly an approximate magnitude is computed using:

$$|G| = |Gx| + |Gy|$$

which is much faster to compute.

The angle of orientation of the edge (relative to the pixel grid orientation) is determined by:

$$\theta = \arctan(Gy / Gx)$$

3.3.3. Prewitt's operator: [20]

Prewitt operator [5] works very similarly to the Sobel operator and is used for detecting vertical and horizontal edges present in images as shown in figure – 7.

-1	0	+1
-1	0	+1
-1	0	+1

Gx

+1	+1	+1
0	0	0
-1	-1	-1

Gy

Fig -7: Prewitt's Operator Mask

3.3.4. Laplacian of Gaussian: [20]

The Laplacian of any image highlights the regions of abrupt intensity change and is therefore often used for detection of edges. The Laplacian is a 2-D isotropic measure of the 2nd order spatial derivative of an image. To compute the Laplacian, it is first smoothed, often using Gaussian Smoothing Filter to reduce its sensitivity to noise, then its 2nd order derivative is computed. The Laplacian normally takes a gray level image as input and produces another gray level image as output. The Laplacian L(x, y) of an image with its intensity value for each pixel I(x, y) is given by:

$$L(x,y) = \frac{\partial^2 I}{\partial x^2} + \frac{\partial^2 I}{\partial y^2}$$

As a digital image can be represented by a set of discrete pixels, a discrete convolution kernel needs to be devised that can approximate the second order derivatives

as per the definition of the Laplacian [5]. The most common kernels for this purpose are shown in figure – 8.

1	1	1
1	-8	1
1	1	1

-1	2	-1
2	-4	2
-1	2	-1

0	1	0
1	-4	1
0	1	0

Fig – 8: Three commonly used discrete approximations to the Laplacian filter.

These kernels are very sensitive to noise, as they are approximating a second derivative measurement on the image. So as already discussed, it must be Gaussian Smoothed before applying the Laplacian filter.

This helps in reducing the high frequency noise components, prior to the differentiation process. In fact, one can design a hybrid filter by convolving the Gaussian smoothing filter with the Laplacian filter, to achieve the desired result. This would be advantageous in two ways: firstly only a few arithmetic operations would be required as both Gaussian and the Laplacian kernels are much smaller than the image. Secondly, as the LoG (i.e., the Laplacian of Gaussian) [6] is pre-computed so only one convolution would be performed at run-time on the image. The 2-D LoG function [7] with Gaussian standard deviations, centered on zero has the following formula as shown in figure – 9.

$$LoG(x,y) = -1/\pi\sigma^4 \left[1 - \left(\frac{x^2+y^2}{2\sigma^2}\right)\right] e^{-\frac{x^2+y^2}{2\sigma^2}}$$

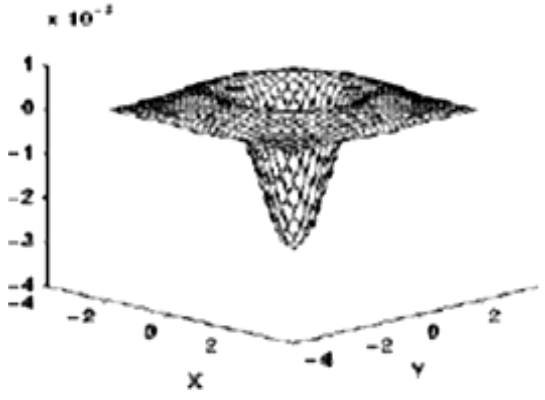

Fig – 9: The 2-D Laplacian of Gaussian (LoG) function. The x and y axes are marked in standard deviations (σ).

A discrete kernel that would approximate the above function (for a Gaussian σ = 1.4) is shown in figure - 10.

0	1	1	2	2	2	1	1	0
1	2	4	5	5	5	4	2	1
1	4	5	3	0	3	5	4	1
2	5	3	-12	-24	-12	3	5	2
2	5	0	-24	-40	-24	0	5	2
2	5	3	-12	-24	-12	3	5	2
1	4	5	3	0	3	5	4	1
1	2	4	5	5	5	4	2	1
0	1	1	2	2	2	1	1	0

Fig – 10: Discrete approximation to LoG function with Gaussian σ = 1.4

Note that on a discrete grid, the simple Laplacian can be seen as a limiting case of the LoG for narrow Gaussians [8]-[10]. This is because smoothing with a very narrow Gaussian on a discrete grid has no effect.

3.3.4. Canny Edge Detection Algorithm [20]

The Canny Edge Detection Algorithm is a multi stage algorithm which was designed by John F. Canny in 1986 to detect a wide range of edges in variety of images. This technique is also referred to as the Optimal Edge Detector which aims at:

1. Congenial detection – able to mark as many real edges in the image as possible.
2. Congenial localization – able to mark as close as possible to the edge in the real image.
3. Minimal response – each edge should be marked exactly once and false edges because of noise should not be created.

He succeeded in achieving his goal and published it in his paper "A Computational Approach to Edge Detection"[19] in which he combined and improved upon the methods that were already stated at that time. The prime consideration for that approach is striving for low error rate. Second issue being well localized edge points. Lastly being the single response to each edge. Based on the above stated criterion, canny edge detector uses the following steps:

1. The original image is filtered out before locating and detecting any edges. This is performed using a suitable Gaussian Smoothing Filter. The Convolution mask used for this purpose is usually much smaller than the actual image. So, it is rolled over the entire image, manipulating only a square of pixels at a time. The

localization error in the detected edges increases slightly as the Gaussian width is increased and if the width of Gaussian mask is increased, it lowers the detector's sensitivity to noise.

2. Once smoothing of image is done by eliminating the noise using Gaussian Filter, the edge strength is to calculate by taking its gradient. To do so, Sobel operator is used which has been already discussed.
3. Compute the direction of the edge using the gradient in x and y direction. Whenever the gradient in the x direction is equal to zero, the edge direction must be equal to 90 degrees or 0 degrees, depending on what the value of the gradient in the y-direction is equal to. If Gy has a value of zero, the edge direction will equal 0 degrees. Otherwise, the edge direction will equal 90 degrees. The formula for finding the edge direction is as follows:

Theta = invtan (Gy / Gx)

4. After finding the direction of edge in previous step, that direction is related to a standard one. To understand this, let us consider the pixels of a 5X5 image:

0	0	0	0	0
0	0	0	0	0
0	0	**1**	0	0
0	0	0	0	0
0	0	0	0	0

It is evident that there are only four possible directions for the pixel with value "1": horizontal direction (0 degrees), positive diagonal (45 degrees), vertical direction (90 degrees), or negative diagonal (135 degrees). Hence, the pixel in question needs to be resolved in one of the four directions depending on which direction it is closest to. So now the edge orientation must be resolved into one of these four directions depending on which direction it is closest to. This can be understood visually by taking a semi circle and dividing it into 5 regions as shown in figure – 11.

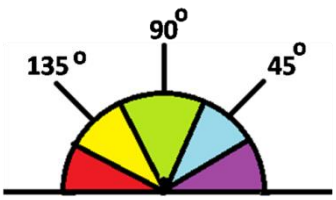

Fig – 11: Edge orientation

Each of the regions can be interpreted as follows:

- Horizontal Edge (0^o) for any edge falling in the purple or red region (0 to 22.5 & 157.5 to 180 degrees)
- Positive diagonal Edge (45^o) for any edge falling in the blue region (22.5 to 67.5 degrees)
- Vertical Edge (90^o) for any edge falling in the green region (67.5 to 112.5 degrees)
- Negative diagonal Edge (135^o) for any edge falling in the yellow region (112.5 to 157.5 degrees)

5. Next the non-maximum suppression is applied after resolving the edge direction. It is done by tracing along standard edge directions and discarding all those pixels that do not fall in that trace, thereby giving only a thin line as output.
6. The output image may contain streaking effect caused by the operator output fluctuating above and below the threshold. This is shown as breaking up of the edge contour. So, needs to be eliminated. This is done by using two thresholds instead of one. In a single threshold technique, if T1 is applied to the image as the threshold value and edge itself has the average strength equal to T1, then there may be cases where due to noise, the edge strength dips below or extends above the threshold resulting in edge pixel being discarded and thus producing a dashed line. So, to avoid it, hysteresis method uses two threshold values: T1 and T2. Hence any pixel which has strength as low as the lo. To avoid this, hysteresis uses 2 thresholds, a high and a low. Any pixel in the image that has a value greater than T1 is presumed to be an edge pixel, and is marked as such immediately. Then, any pixels that are connected to this edge pixel and that have a value greater than T2 are also selected as edge pixels. If you think of following an edge, you need a gradient of T2 to start but you do not stop till you hit a gradient below T1.

4. EXPERIENTIAL ANALYSIS OF VARIOUS EDGE DETECTION ALGORITHMS

All the above stated Edge Detection techniques have been applied on the figure – 12 to provide the readers with visual comparisons among all the methods.

Fig – 12: Original Image

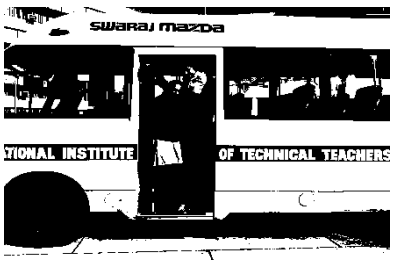

Fig – 13: Resultant Binary Image

The resultant edge detected images are as shown below:

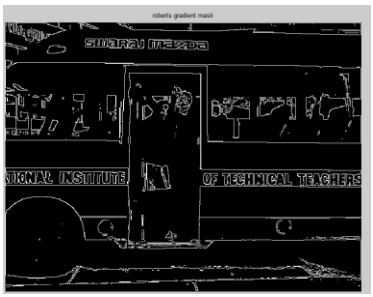

Fig – 14: Robert's Cross Image

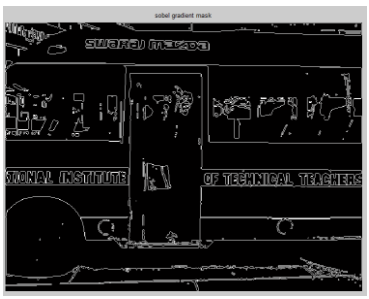

Fig – 15: Sobel Image

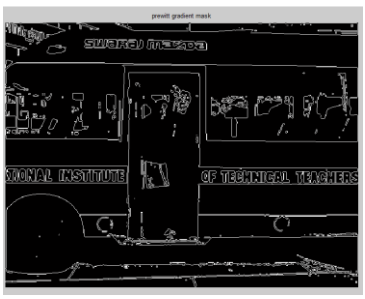

Fig – 16: Prewitt Image

Fig – 17: LoG Image

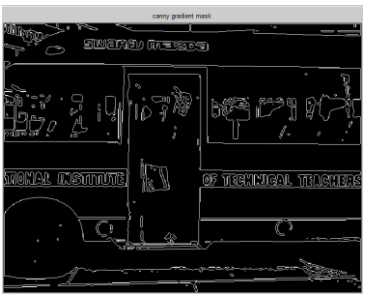

Fig – 18: Canny Image

As clear from the images as expected Canny provides with the best results. This is because non maximal suppression is used to give thin edges and hysteresis is used to avoid streaking. The results of another image being detected are also given to affirm the above results.

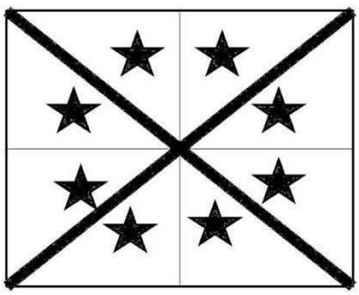

Fig – 19: Original Binary Image

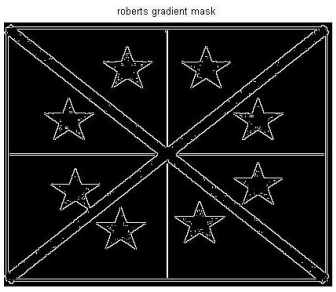

Fig – 20: Robert's Cross Image

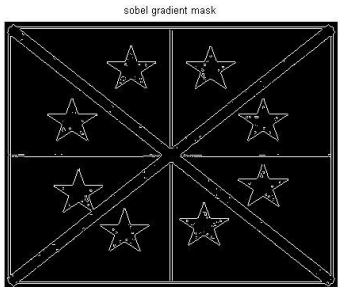

Fig – 21: Sobel Image

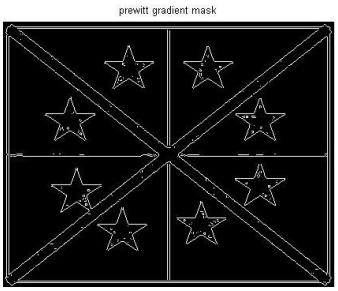

Fig – 22: Prewitt Image

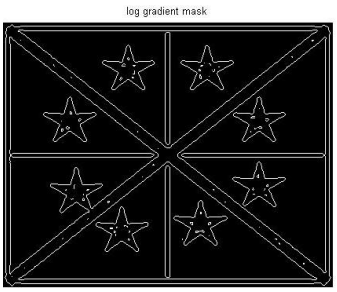

Fig – 23: LoG Image

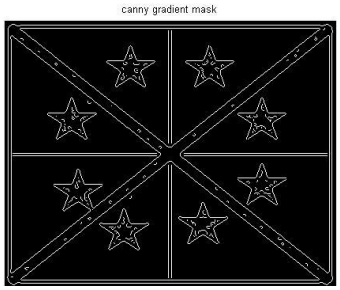

Fig – 24: Canny Image

5. ADVANTAGES AND DISADVANTAGES OF EDGE DETECTOR

As edge detection is a fundamental step in computer vision, it is necessary to point out the true edges to get the best results from the matching process. That is why it is important to choose edge detectors that fit best to the application. In this respect, we first present some advantages and disadvantages of Edge Detection Techniques [13]-[21] within the context of our classification in table [1] – [4].

OPERATOR:	Classical (Robert's Cross, Sobel, Prewitt)
ADVANTAGES:	Simple and fast.
DISADVANTAGES:	Inaccurate and prone to error due to noise.

Table – 1: Classical Operators

OPERATOR:	Zero-Crossing Methods (Laplacian, Second Order Derivative)
ADVANTAGES:	Good detection of edges and correct orientations. Symmetric in all edge directions.
DISADVANTAGES:	Responds to only some of the existing edges, Noise Sensitive.

Table – 2: Zero-Crossing Methods

OPERATOR:	Laplacian of Gaussian (LoG)
ADVANTAGES:	Good edge localization, Analyses wider area around the edge pixel.
DISADVANTAGES:	Inefficient results at the corners, curves and where the gray level intensity function varies. The Laplacian filter does not find the edge orientation.

Table – 3: Laplacian of Gaussian (LoG) Operator

OPERATOR:	Gaussian (Canny)
ADVANTAGES:	Error rate is quite low, Good Localization and response. Improving signal to noise ratio. Better results in noisy conditions.
DISADVANTAGES:	Slow and Complex Computations, False zero crossing, Time consuming.

Table – 4: Gaussian (Canny) Operator

6. CONCLUSION

As edge detection plays a primary step in identifying the various objects in an image, so it is very essential to understand each of the edge detection filters. In this paper, we studied edge detection techniques namely Gradient-based and Laplacian based. The software used to for performing all the edge detection techniques on a binary image is MATLAB 2010. Gradient-based algorithms have a major drawback that these are quite sensitive to noise. The performance of the Canny algorithm on the other hand, relies mainly on the changing parameters i.e., the size of

Gaussian Filter and other subsequent parameters. The large size of Gaussian filter detects large edges quickly but is less effective on noise. The algorithm can be modified by the user by changing the parameters to suit the different environments. As clear from the both speculative and experiential analysis of all the edge detection techniques, Canny's algorithm performs better than the other operators

REFERENCES

1. Orlando J. Tobias and Rui Seara,"Image Segmentation by Histogram Thresholding Using Fuzzy Sets", IEEE Transactions on Image Processing, Vol.11, No.12, 2002, pp. 1457-1465.

2. Djemel Ziou and Salvatore Tabbone,"Edge Detection Techniques- An Overview", International Journal of Pattern Recognition and Image Analysis, 1998.

3. M. Abdulghafour, "Image segmentation using Fuzzy logic and genetic algorithms", Journal of WSCG, vol. 11, no.1, 2003.

4. L. G. Roberts. "Machine perception of 3-D solids" ser. Optical and Electro-Optical Information Processing.MIT Press, 1965.

5. R. C. Gonzalez and R. E. Woods. "Digital Image Processing". 2nd ed. Prentice Hall, 2002.

6. V. Torre and T. A. Poggio. "On edge detection". IEEE Trans. Pattern Anal. Machine Intell., vol. PAMI-8, no.2, pp. 187-163, Mar. 1986.

7. E. R. Davies. "Constraints on the design of template masks for edge detection". Partern Recognition Lett.,vol. 4, pp. 11 1-120, Apr. 1986.

8. W. Frei and C.-C. Chen. "Fast boundary detection: A generalization and a new algorithm ". lEEE Trans.Comput., vol. C-26, no. 10, pp. 988-998, 1977.

9. W. E. Grimson and E. C. Hildreth. "Comments on Digital step edges from zero crossings of second directional derivatives''. IEEE Trans. Pattern Anal. Machine Intell., vol. PAMI-7, no. 1, pp. 121-129, 1985.

10. R. M. Haralick. "Digital step edges from zero crossing of the second directional derivatives," IEEE Trans. Pattern Anal. Machine Intell., vol. PAMI-6, no. 1, pp. 58-68, Jan. 1984.

11. Ibrahiem M. M. El Emary, "On the Application of Artificial Neural Networks in Analyzing and Classifying the Human Chromosomes", Journal of Computer Science,vol.2(1), 2006, pp.72-75.

12. J. Canny. "Finding edges and lines in image". Master's thesis, MIT, 1983.

13. Mantas Paulinas and Andrius Usinskas, "A Survey of Genetic Algorithms Applicatons for Image Enhancement and Segmentation", Information Technology and Control Vol.36, No.3, 2007, pp.278-284.

14. Xian Bin Wen, Hua Zhang and Ze Tao Jiang, "Multiscale Unsupervised Segmentation of SAR Imagery Using the Genetic Algorithm", Sensors, vol.8, 2008, pp.1704-1711.

15. Metin Kaya, "Image Clustering and Compression Using an Annealed Fuzzy Hopfield Neural Network", International Journal of Signal Processing,2005, pp.80-88.

16. N. Senthilkumaran and R. Rajesh, "A Study on Edge Detection Methods for Image Segmentation", Proceedings of the International Conference on Mathematics and Computer Science (ICMCS-2009), 2009,Vol.I, pp.255-259.

17. N. Senthilkumaran and R. Rajesh, "A Study on Split and Merge for Region based Image Segmentation", Proceedings of UGC Sponsored National Conference Network Security (NCNS-08) , 2008, pp.57-61.

18. N. Senthilkumaran and R. Rajesh, "Edge Detection Techniques for Image Segmentation - A Survey", Proceedings of the International Conference on Managing Next Generation Software Applications (MNGSA-08), 2008, pp.749-760.

19. J. F. Canny. "A computational approach to edge detection". IEEE Trans. Pattern Anal. Machine Intell., vol. PAMI-8, no. 6, pp. 679-697, 1986

20. Raman Maini and Dr. Himanshu Aggarwal "Study and Comparison of
Various Image Edge Detection Techniques",International Journal of Image Processing (IJIP), Volume (3) : Issue (1)