Study of Some Edge Detection Techniques

A dissertation submitted in partial fulfilment

Of the requirement for the degree of

Master of Electrical Engineering

By

Nabajyoti Singha Roy

Roll No- 2010802013

Under the Guidance of

Prof. Mita Dutta

Department of Electrical Engineering

Faculty of Engineering and Technology

Jadavpur University

Kolkata-700032

2022

JADAVPUR UNIVERSITY

FACAULTY COUNCIL OF ENGINEERING & TECHNOGY

ELECTRICAL ENGINEERING DEPARTMENT

KOLKATA-700032

Certificate of Recommendation

This is to certify that the thesis entitled " ***Study of Some Edge Detection Techniques*** *" by SRI* ***Nabajyoti Singha Roy*** *(****2010802013****), submitted to the Jadavpur University Kolkata, West Bengal for the award of Master of Electrical Engineering in Electrical Measurement and Instrumentation, is a record of bonafide research work carried out by him in the Department of Electrical Engineering, under my supervision. I believe that this thesis fulfils part of the requirements for the award of degree of Master of Electrical Engineering during the academic session 2020- 2022. The results embodied in the thesis have not been submitted for the award of any other degree elsewhere.*

Prof (Dr.) SASWATI MAZUMDER	Prof. MITA DUTTA
HEAD ELECTRICAL ENGG.DEPT	ELECTRICAL ENGG DEPT
JADAVPUR UNIVERSITY	JADAVPUR UNIVERSITY

DEAN- FACULTY OF ENGINEERING AND TECHNOLOGY

JADAVPUR UNIVERSITY

FACAULTY COUNCIL OF ENGINEERING & TECHNOGY

ELECTRICAL ENGINEERING DEPARTMENT

KOLKATA-700032

CERTIFICATE OF APPROVAL

This foregoing thesis is hereby approved as a credible study of an engineering subject carried out and presented in a manner satisfactorily to warranty its acceptance as a prerequisite to the degree for which it has been submitted. It is understood that by this approval the undersigned do not endorse or approve any statement made or opinion expressed or conclusion drawn therein but approve the thesis only for purpose for which it has been submitted.

BOARD OF EXAMINERS

Committee of final examination ----------------------------------

For evaluation of Thesis ----------------------------------

Acknowledgment

Most real spirit is to achieve a goal through a way of perfection. The achievement which I have gained cannot be possible without the cooperation of numerous personalities.

I am very much thankful to various personalities who help me a lot for structuring my thesis. At the outset, I would like to express my thanks to Prof. Mita Dutta for his guidance, supervision during this thesis work. As a guide her advice always encourages me and helps me a lot to and out suitable way for successful completion of this work. Her idea, observations help me to enrich my knowledge.

I am grateful that I got lovely opportunity to study in one of the best university and under some of the best faculties. It really changed my life.

I am grateful to all the professors of our department for their advice, support, encouragement and valuable comments.

I am grateful to my friends, without whom, this journey would not be this much pleasant.

I am also grateful to my family for their blessing, advice and moral support. I want to dedicate this work to them which could not be fulfilled without their love and blessing.

Nabajyoti Singha Roy

Contents

Chapter 1

1.1 Introduction:

In computer vision, edge detection is a process which attempts to capture the significant properties of objects in the image. These properties include discontinuities in the photometrical, geometrical and physical characteristics of objects. Such information give rise to variations in the grey level image; the most commonly used variations are discontinuities (step edges), local extrema (line edges), and 2D features formed where at least two edges meet (junctions).

The purpose of edge detection is to localize these variations and to identify the physical phenomena which produce them. Edge detection must be efficient and reliable because the validity, efficiency and possibility of the completion of subsequent processing stages rely on it. To fulfil this requirement, edge detection provides all significant information about the image. For this purpose, image derivatives are computed. However, differentiation of an image is an ill-posed problem; image derivatives are sensitive to various sources of noise, i.e., electronic, semantic, and discretization/quantification effects. To regularize the differentiation, the image must be smoothed. However, there are undesirable effects associated with smoothing, i.e., loss of information and displacement of prominent structures in the image plane. Furthermore, the properties of commonly-used differentiation operators are different and therefore they generate different edges. It is difficult to design a general edge detection algorithm which performs well in many contexts and captures the requirements of subsequent processing stages. Consequently, over the history of digital image processing a variety of edge detectors have been devised which differ in their purpose (i.e., the photometrical and geometrical properties of edges which they are able to extract) and their mathematical and algorithmic properties.

This paper describes the characteristics of edges, the properties of detectors, the methodology- of edge detection, the mutual influence between them and the main idea behind the major edge detection techniques.

1.2 Early and recent history of edge detection techniques

Owing to the importance of image edges, image edge detection has been receiving a lot of attention from researchers since the time it was proposed, and Figure 1 illustrates the development of edge detection algorithms. The earliest edge detection operator was the Robert (**Ziou and Tabbone, 1998**) operator proposed by Lawrence Roberts in 1963, which is also known as the cross-differential algorithm as the simplest operator, and its underlying principle is to locate the image contour with the help of a local difference operator. It was followed in 1970 by the Prewitt operator (**Shrivakshan and Chandrasekar, 2012**), which is often applied to high noise, pixel-value fading images. Then came the Sobel operator (**Marr and Hildreth, 1980**), which introduced the idea of weights, and the Laplacian operator (**Xin Wang, 2007**), which used second-order differentiation, in the 1980s. Later the optimal Canny operator (**Canny, 1986**) was proposed in 1986, which continuously optimized the image contour information by filtering, enhancement, and detection steps, and became one of the best operators for detection in the field of edge detection at that time. After that, with the continuous development of deep learning, various methods based on CNN to achieve edge detection have emerged. In 2015 Bertasius et al. changed the traditional bottom-up idea of edge detection and proposed a top-down multi-scale divergent deep network DeepEdge (**Bertasius et al., 2015a**) for edge detection. In the same year, Xie et al. developed the holistic nested edge detection algorithm HED (**Xie and Tu, 2015**), which solved the problem of holistic image-based training and prediction and multi-scale multi-level feature learning. In 2017 Liu et al. proposed an accurate edge detector RCF (**Liu et al., 2017**) using richer convolutional features. In 2019 Deng et al. proposed a novel end-to-end edge detection system DSCD (**Deng and Liu, 2020**), which effectively utilizes multi-scale and multi-level features to produce high-quality object edge output. In 2021 Su et al. designed PiDiNet (**Su et al., 2021**), a simple, lightweight, and efficient edge detection architecture.

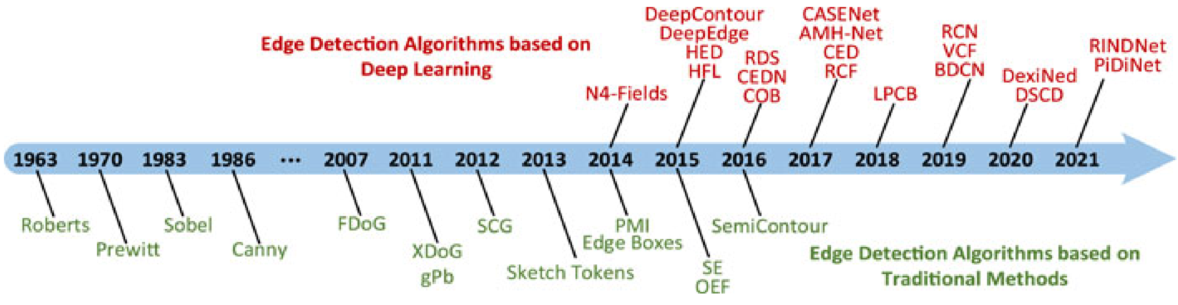


Fig. Development of edge detection algorithms based on traditional and deep learning methods.

With the development of technology, the recognition performance of edge detection networks has gradually improved, and the accuracy rate has increased. However, at the same time, the depth of the network has been deepened, leading to problems such as oversized parameters, training difficulties, and model complexity. In this paper, we will analyse and classify the classical and latest edge detection models in terms of model structure, technical difficulties, method advantages, and backbone networks from two categories based on traditional methods and deep learning methods. Then we will introduce the backbone networks (**AlexNet, VGG16, ResNet**), evaluation metrics (ODS, OIS, FPS, PR curve), and datasets (BSDS500, NYUD, PASCAL-VOC, PASCAL-Context, MultiCue, BIPED), which are closely related to edge detection. Finally, the methods mentioned in this paper are briefly summarized, and the problems found and the future directions of edge detection focus are briefly described.

1.3 Objective of the Thesis

Our objective is to apply various edge detection techniques into various images to compare each of them to extract best result. After that for further improvement of the results we will apply image enhancement (Histogram equalization) technique prior to edge detection

As there are no such in general approach to find edges of an image so we will apply and study different edge detection technique on different types of image to observe which techniques is best for which particular image.

In chapter 2 edge detection techniques in spatial domain have been discussed. Histogram equalization technique has been discussed in chapter 3. In chapter4 we have discussed about high pass filtering in frequency domain and chapter 5 refer to result and discussion.

Chapter 2

Spatial domain Edge detection techniques

2.1 Introduction

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. Discontinuities in the image intensity can be either (1) step discontinuities, where the image intensity abruptly changes from one value on one side of the discontinuity to a different value on the opposite side, or (2) line discontinuities, where the image intensity abruptly changes value but then returns to the starting value within some short distance. However, step and line edges are rare in real images. Because of low-frequency components or the smoothing introduced by most sensing devices, sharp discontinuities rarely exist in real signals. Step edges become ramp edges and line edges become roof edges, where intensity changes are not instantaneous but occur over a finite distance. Illustrations of these edge profiles are shown in Figure 2.1

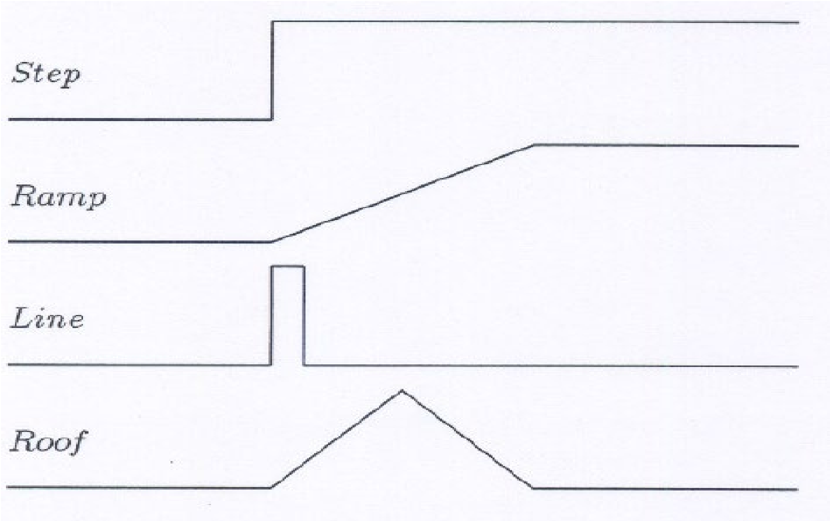


Fig 2.1 One dimensional edge profiles

2.1.1 Steps in Edge Detection

Algorithms for edge detection contain three steps:

- **Filtering:**

Filtering is commonly used to improve the performance of an edge detector with respect to noise. However, there is a trade-off between edge strength and noise reduction. More filtering to reduce noise results in a loss of edge strength.

- **Enhancement:**

In order to facilitate the detection of edges, it is essential to determine changes in intensity in the neighbourhood of a point. Enhancement emphasizes pixels where there is a significant change in local intensity values and is usually performed by computing the gradient magnitude.

- **Detection:**

We only want points with strong edge content. However, many points in an image have a nonzero value for the gradient, and not all of these points are edges for a particular application. Therefore, some method should be used to determine which points are edge points. Frequently, thresholding provides the criterion used for detection.
Examples at the end of this section will clearly illustrate each of these steps using various edge detectors.

Many edge detectors have been developed in the last two decades. Here we will discuss some commonly used edge detectors. As will be clear, edge detectors differ in use of the computational approach in one or more of the above three steps. We will discuss the implications of these steps after we have discussed the edge detectors.

2.1.2 Gradient

Edge detection is essentially the operation of detecting significant local changes in an image. In one dimension, a step edge is associated with a local peak in the first derivative. The gradient is a measure of change in a function, and an image can be considered to be an array of samples of some continuous function of image intensity. By analogy, significant changes in the gray values in an image can be detected by using a discrete approximation to the gradient.

The gradient is the two-dimensional equivalent of the first derivative and is defined as the vector

$$\mathbf{G}[f(x,y)] = \begin{bmatrix} G_x \\ G_y \end{bmatrix} = \begin{bmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \end{bmatrix}.$$

………… 1

There are two important properties associated with the gradient:

(1) The vector G[j(x, y)] points in the direction of the maximum rate of increase of the function j(x, y), and

(2) The magnitude of the gradient, given by

$$G[f(x,y)] = \sqrt{G_x^2 + G_y^2},$$

…………..2

equals the maximum rate of increase of j (x, y) per unit distance in the direction G. It is common practice, however, to approximate the gradient magnitude by absolute values

$$G[f(x,y)] \approx |G_x| + |G_y|$$

or

$$G[f(x,y)] \approx \max(|G_x|, |G_y|).$$

………………..3

From vector analysis, the direction of the gradient is defined as:

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

……………..4

where the angle a is measured with respect to the x axis. Note that the magnitude of the gradient is actually independent of the direction of the edge. Such operators are called isotropic operators.

2.2 Operators

2.2.1 Roberts Operator

The Roberts cross operator provides a simple approximation to the gradient magnitude:

$$G[f[i,j]] = |f[i,j] - f[i+1,j+1]| + |f[i+1,j] - f[i,j+1]|.$$

Using convolution masks, this becomes

$$G[f[i,j]] = |G_x| + |G_y|$$

where Gx and Gy are calculated using the following masks:

$$G_x = \begin{array}{|c|c|}\hline 1 & 0 \\ \hline 0 & -1 \\ \hline \end{array} \qquad G_y = \begin{array}{|c|c|}\hline 0 & -1 \\ \hline 1 & 0 \\ \hline \end{array}$$

The Roberts operator is an approximation to the continuous gradient at that point and not at the point [i,j] as might be expected. The results of Roberts edge detector are discussed in chapter 5.

2.2.2 Sobel Operator

The Sobel operator performs a 2-D spatial gradient measurement on an image and so emphasizes regions of high spatial frequency that correspond to edges. Typically it is used to find the approximate absolute gradient magnitude at each point in an input grayscale image.

Sobel differential operator is a directional differential operator in a basis of odd sizetemplate. The expressions of formula as follow

$$
\begin{aligned}
G_x(i, j) &= f[i-1, j+1] + 2 \times f[i, j+1] \\
&+ f[i+1, j+1] - f[i-1, j-1] \\
&- 2 \times f[i, j-1] - f[i+1, j-1]
\end{aligned}
$$

$$
\begin{aligned}
G_y(i, j) &= f[i+1, j-1] + 2 \times f[i+1, j] \\
&+ f[i+1, j+1] - f[i-1, j-1] \\
&- 2 \times f[i-1, j] - f[i-1, j+1]
\end{aligned}
$$

The convolution template of the Sobel operator is expressed as

$$
G_x = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}
$$

$$
G_y = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}
$$

The calculating steps of Sobel operator: first, the edge detection image is divided into matrix form

$$A = \begin{bmatrix} f_1(x-1, y-1) & f_1(x-1, y) & f_1(x-1, y+1) \\ f_1(x, y-1) & f_1(x, y) & f_1(x, y+1) \\ f_1(x+1, y-1) & f_1(x+1, y) & f_1(x+1, y+1) \end{bmatrix}$$

Multiply horizontal direction by vertical direction of the template and then multiply the vertical direction by horizontal direction of the template, Fx=Gx.*A, Fx=Gy.*A. gradient size calculation, as shown in equation 5.

$$G = \sqrt{\left(G_x^2 + G_y^2\right)} \qquad \text{……….5}$$

The formula for calculating the gradient direction is shown in equation 6.

$$\theta = \arctan\left(\frac{G_y}{G_x}\right) \qquad \text{………….6}$$

The Sobel operator introduced the weighted local average, it can not only affect the image edge detection but also suppress noise further, but the edge is wider.

2.2.3 Prewitt Operator

The Prewitt operator uses the same equations as the Sobel operator, except that the constant c = 1. Therefore:

$$s_x = \begin{array}{|c|c|c|}\hline -1 & 0 & 1 \\ \hline -1 & 0 & 1 \\ \hline -1 & 0 & 1 \\ \hline \end{array} \qquad s_y = \begin{array}{|c|c|c|}\hline 1 & 1 & 1 \\ \hline 0 & 0 & 0 \\ \hline -1 & -1 & -1 \\ \hline \end{array}$$

Note that, unlike the Sobel operator, this operator does not place any emphasis on pixels that are closer to the center of the masks.

Often, this absolute magnitude is the only output the user sees --- the two components of the gradient are conveniently computed and added in a single pass over the input image using the pseudo-convolution operator shown in Figure.

P_1	P_2	P_3
P_4	P_5	P_6
P_7	P_8	P_9

Using this kernel the approximate magnitude is given by:

$$|G| = |(P_1 + 2 \times P_2 + P_3) - (P_7 + 2 \times P_8 + P_9)| + |(P_3 + 2 \times P_6 + P_9) - (P_1 + 2 \times P_4 + P_7)|$$

2.2.4 Canny edge detection

Canny edge detection is a technique to extract useful structural information from different vision objects and dramatically reduce the amount of data to be processed. It has been widely applied in various computer vision systems. Canny has found that the requirements for the application of edge detection on diverse vision systems are relatively similar. Thus, an edge detection solution to address these requirements can be implemented in a wide range of situations

To know canny edge detection we have to know 1st Gaussian smoothing function.

- **Gaussian Smoothing**

The Gaussian smoothing operator is a 2-D convolution operator that is used to 'blur' images and remove detail and noise. In this sense it is similar to the mean filter, but it uses a different kernel that represents the shape of a Gaussian (`bell-shaped') hump. This kernel has some special properties which are detailed below.

The Gaussian distribution in 1-D has the form:

$$G(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{x^2}{2\sigma^2}} \qquad \text{…………7}$$

Where σ sigma is the standard deviation of the distribution. We have also assumed that the distribution has a mean of zero (i.e. it is centered on the line x=0). The distribution is illustrated in Figure 2.2

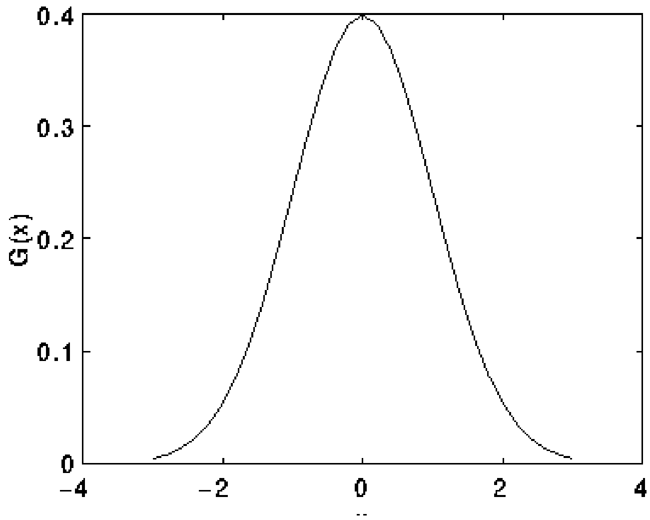


Fig 2.2 1-D Gaussian distribution with mean 0 and sigma=1

The idea of Gaussian smoothing is to use this 2-D distribution as a `point-spread' function, and this is achieved by convolution. Since the image is stored as a collection of discrete pixels we need to produce a discrete approximation to the Gaussian function before we can perform the convolution.

- **Canny edge detection process**

  The Canny edge detector is the first derivative of a Gaussian and closely approximates the operator that optimizes the product of signal-to-noise ratio and localization. The Canny edge detection algorithm is summarized by the following notation. Let I[i, j] denote the image. The result from convolving the image with a Gaussian smoothing filter using separable filtering is an array of smoothed data

$$S[i,j] = G[i,j;\sigma] \star I[i,j],$$

Where σ is the spread of the Gaussian and controls the degree of smoothing. The gradient of the smoothed array S[i,j] can be computed using the 2 x 2 first-difference approximations to produce two arrays P[i,j] and Q[i, j] for the x and y partial derivatives:

$$
\begin{aligned}
P[i,j] &\approx (S[i,j+1] - S[i,j] \\
&\quad + S[i+1,j+1] - S[i+1,j])/2 \\
Q[i,j] &\approx (S[i,j] - S[i+1,j] \\
&\quad + S[i,j+1] - S[i+1,j+1])/2.
\end{aligned}
$$

The finite differences are averaged over the 2 x 2 square so that the x and y partial derivatives are computed at the same point in the image. The magnitude and orientation of the gradient can be computed from the standard formulas for rectangular-to-polar conversion

$$
\begin{aligned}
M[i,j] &= \sqrt{P[i,j]^2 + Q[i,j]^2} \\
\theta[i,j] &= \arctan(Q[i,j], P[i,j]),
\end{aligned}
$$

where the arctan function takes two arguments and generates an angle over the entire circle of possible directions. These functions must be computed efficiently, preferably without using floating-point arithmetic. It is possible to compute the gradient magnitude and orientation from the partial derivatives by table lookup.

2.3 Histogram Equalization

Histogram equalization is a method in image processing of contrast adjustment using the image's histogram.

This method usually increases the contrast of the image, especially when the image is represented by a narrow range of intensity values. Through this adjustment, the intensities can be better distributed on the histogram utilizing the full range of intensities evenly. This allows for areas of lower local contrast to gain a higher contrast. Histogram equalization accomplishes this by effectively spreading out the highly populated intensity values which are used to degrade image contrast.

The method is useful in images with backgrounds and foregrounds that are both bright or both dark. In particular, the method can lead to better views of bone structure in x-ray images, and to better detail in photographs that are either over or under-exposed. A key advantage of the method is that it is a fairly straightforward technique adaptive to the input image and an invertible operator. So in theory, if the histogram equalization function is known, then the original histogram can be recovered. The calculation is not computationally intensive. A disadvantage of the method is that it is indiscriminate. It may increase the contrast of background noise, while decreasing the usable signal.

In scientific imaging where spatial correlation is more important than intensity of signal (such as separating DNA fragments of quantized length), the small signal-to-noise ratio usually hampers visual detections.

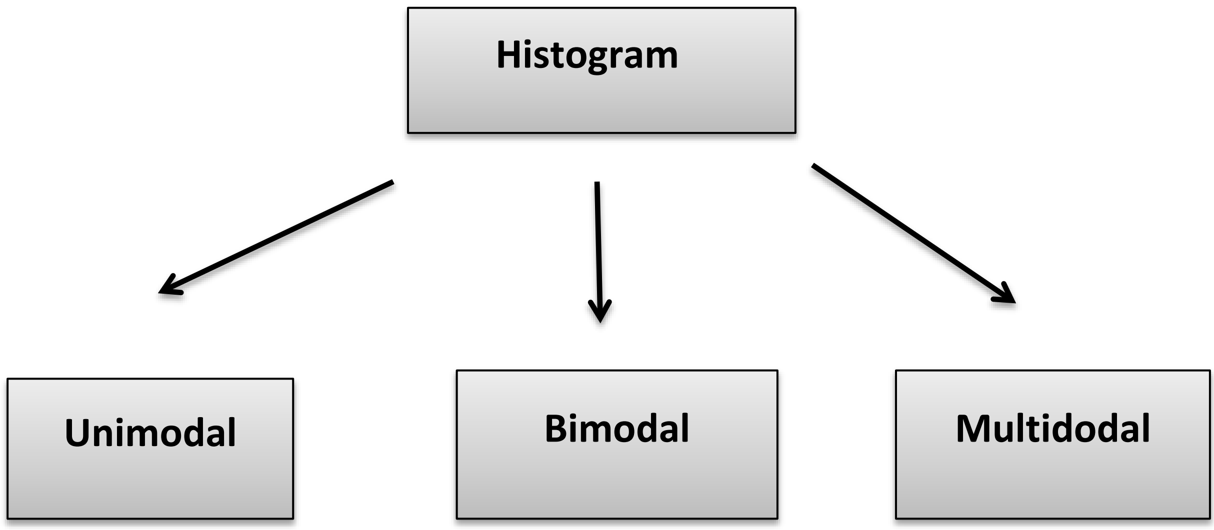
Histogram equalization often produces unrealistic effects in photographs; however it is very useful for scientific images like thermal, satellite or x-ray images, often the same class of images to which one would apply false-color. Also histogram equalization can produce undesirable effects (like visible image gradient) when applied to images with low color depth. For example, if applied to 8-bit image displayed with 8-bit gray-scale palette it will further reduce color depth (number of unique shades of gray) of the image. Histogram equalization will work the best when applied to images with much higher color depth than palette size, like continuous data or 16-bit gray-scale images.

2.3.1 Classification

Histograms are useful for showing patterns within your data and getting an idea of the distribution of your variable at a glance. The first distinguishing feature apparent in a histogram is the number of modes, or peaks, in the distribution. A peak occurs anywhere that the distribution falls and then rises again, even if it does not rise as high as the previous peak.

According to pixel distribution it has been classifed into 3 parts.

Shown in the next page.



- **Unimodal** if the empirical distribution has one local maximum.
- **Bimodal** if it has two local maximum.
- **Multimodal** if the distribution has more than one local maximum.

2.3.2 Steps involved in Histogram equalization

1. Get the input image
2. Generate the histogram for the image
3. Find the local minima of the image
4. Divide the histogram based on the local minima
5. Have the specific gray levels for each partition of the histogram
6. Apply the histogram equalization on each partition

2.3.3 Algorithm

- Compute the histogram of pixel values of the input image. The histogram places the value of each pixel $f[x,y]$ into one of L uniformly-spaced buckets $h[i]$

$$h[i] = \sum_{x=1}^{N}\sum_{y=1}^{M}\begin{cases}1, & if\ f[x,y] = i \\ 0, & Otherwise\end{cases}$$

Where L=2^8 and the image dimension is M×N

- Calculate the cumulative distribution function

$$CDF[j] = \sum_{i=1}^{j} h[i]$$

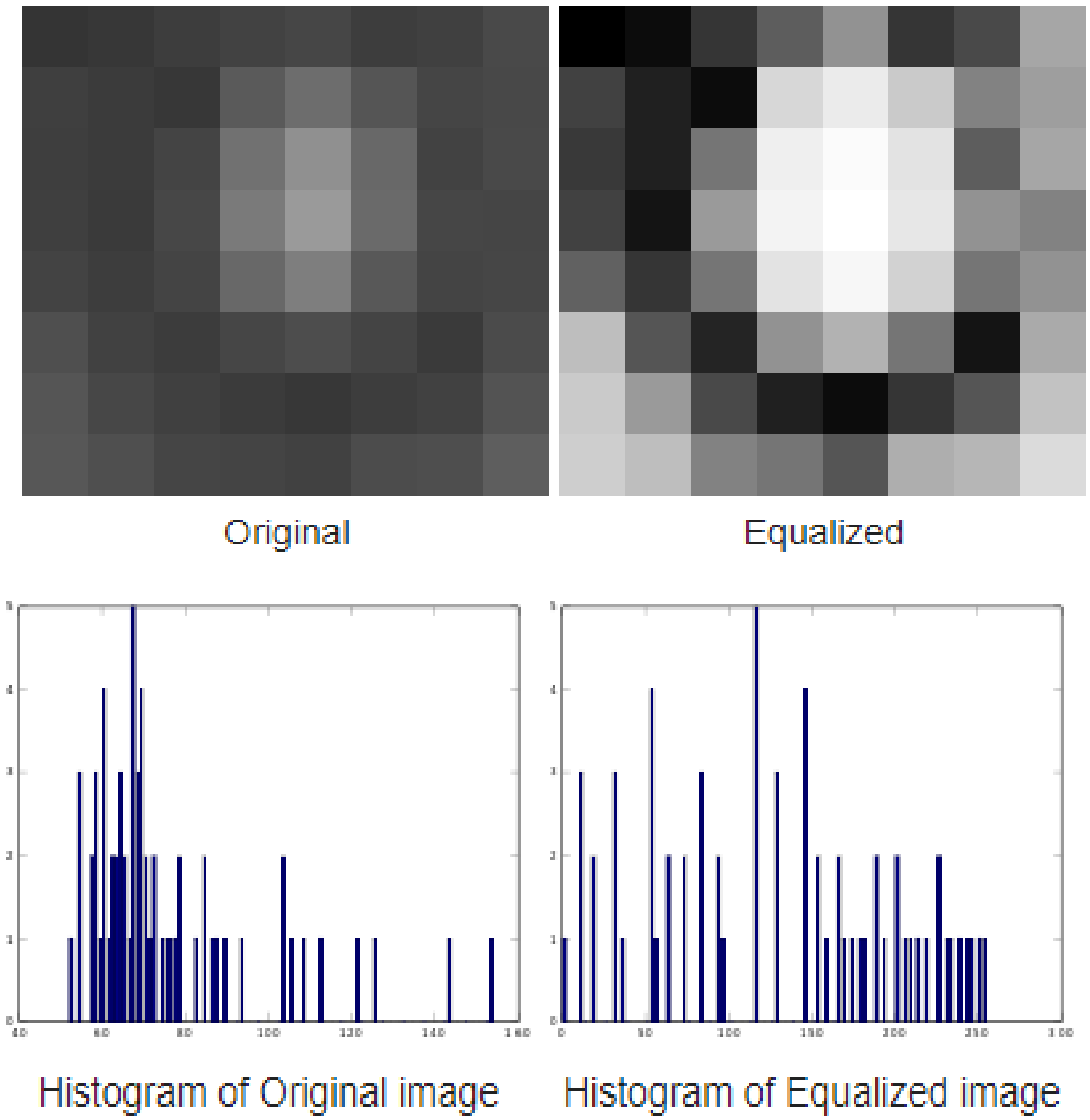
- Scale the input image using the cumulative distribution function to produce the output image.

$$g[x,y] = \frac{CDF\big[f[x,y]\big] - CDF_{min}}{(N \times M) - CDF_{min}} \times (L-1)$$

Where CDFmin is the smallest non-zero value of the cumulative distribution function.

Considering a small 8bit gray scale image, shown in the next page

Fig 2.3 The 8×8 sub-image shown in 8-bit grayscale



Chapter 3

Edge detection using frequency domain techniques

3.1 Introduction

An image can be converted from spatial domain to the frequency domain and from frequency domain to the spatial domain respectively using the 2D discrete Fourier Transform and inverse discrete Fourier Transform. The visualization of filters is more intuitive in the fact that low pass filters (image smoothening filters) and high pass filters (image sharpening filters) are better understood in the frequency domain.

This method of edge detection based upon discrete Fourier expansion of the brightness function f(m,n). The foundation of frequency domain technique is given by the following relation

$$\mathbf{F'(u,v) = \{H(u,v)\ F(u,v)\}}$$

Where F (u, v) and F' (u, v) are Fourier transform of the image (input) function f (m, n) and processed (output) image function f ' (m, n) respectively. H (u, v) is referred to as 'filter transfer function' u and v denote the spatial frequency components.

In the edge enhancement problem, f (m, n) is given and the actual problem arrives after computing the Fourier coefficients F (u, v) and that is to select a function H (u, v) which will manipulate these coefficients so that the output image is

$$\mathbf{F'(m, n)} = \boldsymbol{F^{-1}\{ H(u,v)\, F(u,v)\}} \qquad \text{..........3.1}$$

3.2 High pass filter

Sharpening can be achieved in frequency domain by a high pass filtering process which attenuates the low frequency components without disturbing

High frequency information in the Fourier transform.

The two- dimensional discrete Fourier transform pairs for an MXN dimensional image f (m, n) is given by

$$\boldsymbol{F(u,v) = \frac{1}{MN}\sum_{m=0}^{M-1}\sum_{n=0}^{N-1} f(m,n)\; e^{-2\pi j(um/M+vn/N)}} \qquad \text{.......3.2}$$

For u= 0…M-1, v=0, 1……..N-1, and

$$\boldsymbol{f(m,n) = \sum_{u=0}^{M-1}\sum_{v=0}^{N-1} F(u,v)\, e^{2\pi j(um/M+vn/N)}} \qquad \text{....3.3}$$

For m=0…… M-1

n= 0…… N-1

The strategy involved in frequency domain operation is shown in fig 3.1, where the origin of the Fourier transform of f(m,n) has been shifted to the centre of its corresponding MXN frequency square by multiplying f (m, n) by $(-1)^{m+n}$

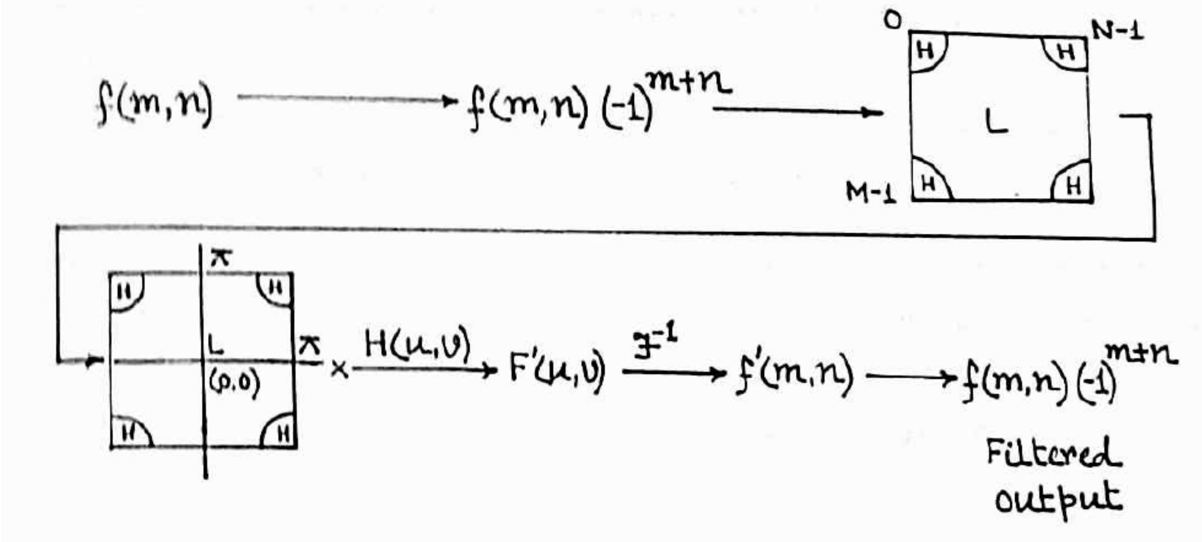


Fig 3.1 Shifting of the origin of Fourier transform of f(m,n) to the centre of its corresponding MXN frequency square.

The geometric centre of the Fourier image falls within the pixel for which u= v =0, called the zero frequency point. The word 'frequency' here comes from the fact that m and n in the expansion for the inverse fourier transformation are to be seen as spatial coordinates, so that u and v become spatial frequencies. In scanning and image it is found that the discrete gradient of the gray level values determine the spatial frequencies present. The low frequency present coefficients are the weighted sum of the pixel intensities while the high frequencies are the weighted sum of the gradients. So after high pass filtering inversion of the Fourier transform, which is also a weighted sum, produces an image where the gradients are dominant. A high pass filtering is characterized by transfer function having a relatively large magnitude for spatial frequencies far from the origin, and a relative low magnitude for frequencies near the origin. Since high spatial frequencies correspond to sharp edges, high-pass filtering enhances edges and therefore is analogous to spatial differentiation.

Fig 3.2 demonstrates the way of computing 2-D Fourier transform as a series of 1-D Fourier transform. It is to be mentioned here that same results would be obtained by first taking transform along the row of f(m,n) and then along each column of F(m,n) or vise versa.

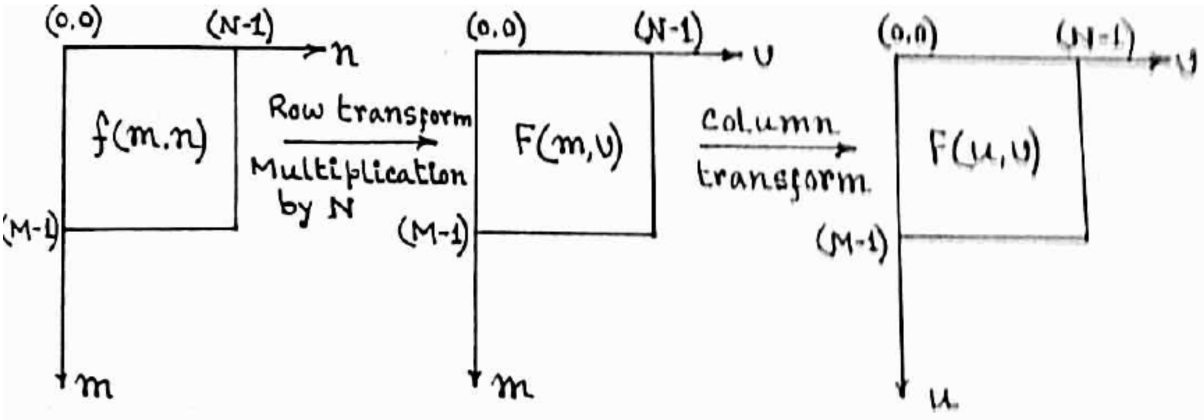


Fig 3.2 Computation of 2-D Fourier transform as a series of 1-d transform

The schematic diagram for high-pass filtering operation in Fourier domain is shown in Fig 4.3

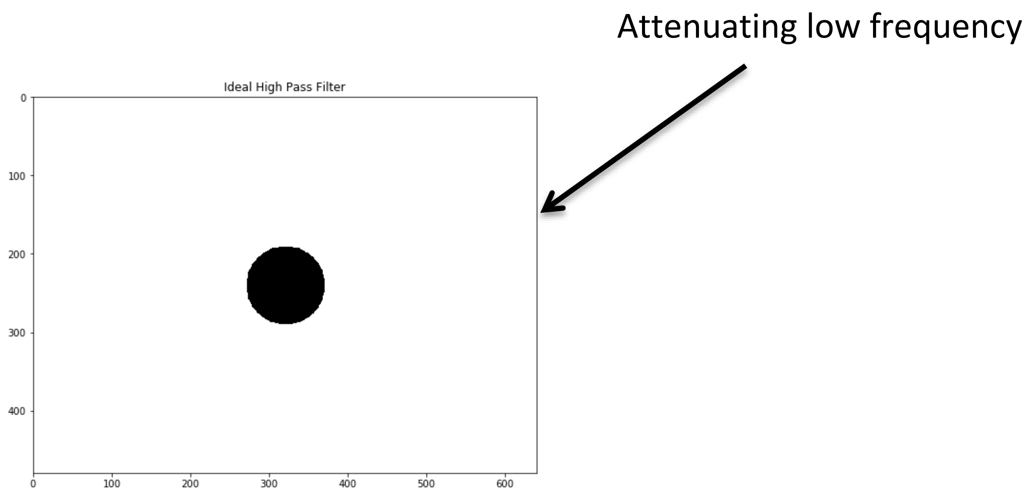


Fig 3.3 Schematic diagram for high- pass filtering

The transfer function of the **high pass filter** can be specified by the function

$$H(u,v) = \begin{cases} 0 & D(u,v) \le D_0 \\ 1 & D(u,v) > D_0 \end{cases}$$

Where, D_0 is a positive constant.

HPF passes all the frequencies outside of a circle of radius D_0 from the origin without attenuation and cuts off all the frequencies within the circle.

This D_0 is the transition point between H (u, v) = 1 and H (u, v) = 0, so this is termed as cut off frequency.

D (u, v) is the Euclidean Distance from any point (u, v) to the origin of the frequency plane, $D(u,v) = \sqrt{(u^2 + v^2)}$.

Chapter 4

Result and discussion

4.1 Introduction:

Here we have taken different images such as

- Human portrait
- X ray image
- Texture Image
- Blurred Image
- Signature

We have applied different edge detection techniques (*In spatial domain Sobel, Robert, Prewitt, Canny and in frequency domain high pass filtering)* and study the results to find the best edge detection technique for that particular type of image.

After that for better result we have applied histogram equalization technique on the images and then find the edges.

4.2 Results applying spatial domain techniques

➢ Human portrait

Shown in the next page

Here we have taken a **bimodal** image of human portrait of resolution **300X300** and gray scale level **0 to 256**.

Input Image

Histogram

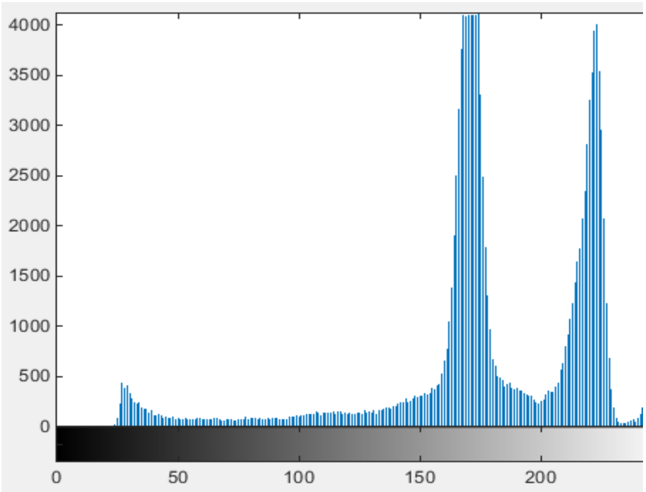


Fig 4.1

Sobel

Prewitt

Fig 4.2

Fig4.3

Canny **Robert**

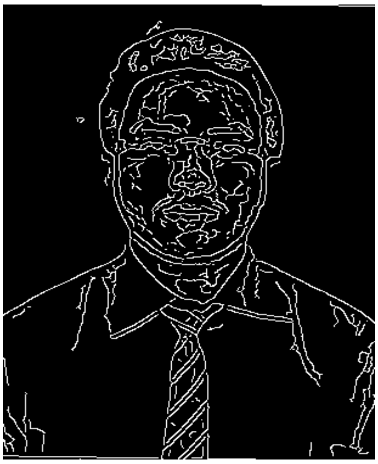


Fig4.4

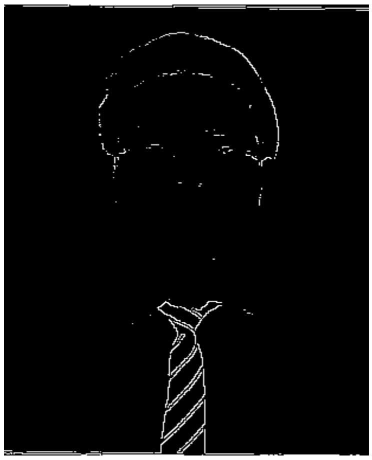


Fig4.5

Here in **robert edge** detection (fig4.5), detected edges is very week.The main reason for using the Roberts Cross operator is that it is very quick to compute. Only four input pixels need to be examined to determine the value of each output pixel, and only subtractions and additions are used in the calculation. In addition there are no parameters to set. Its main disadvantages are that since it uses such a small kernel (**2X2 Matrix used**), it is very sensitive to noise. As we can see in figure 4.5 that it produces very weak responses to genuine edges unless they are very sharp.

In fig 4.5 notice the spurious bright dots on the image which demonstrate that the operator is susceptible to noise. Also note that only the strongest edges have been detected.

Compare to robert, the **Sobel (fig 4.2) and Prewitt(fig 4.3)** operator performs much better in this respect. Because the kernel used here is **3X3 matrix.**

In this particular potrait image the result of sobel and prewitt oparator are similar.

The **Sobel** and **Prewitt** operator are slower to compute than the **Roberts Cross** operator, but its larger convolution kernel smooths the input image to a greater extent and so makes the operator less sensitive to noise. The operator also generally produces considerably higher output values for similar edges, compared with the Roberts Cross.

Note that the spurious noise that afflicted the **Roberts Cross** output image is still present in this image, but its intensity relative to the genuine lines has been reduced, and so there is a good chance that we can get rid of this entirely by thresholding. Also, notice that the lines corresponding to edges have become thicker compared with the Roberts Cross output due to the increased smoothing of the Sobel operator.

In **fig 4.4** the detected edges by **canny operator** are very strong. The effect of the Canny operator is determined by three parameters --- the width of the Gaussian kernel used in the smoothing phase, and the upper and lower thresholds used by the tracker. Increasing the width of the Gaussian kernel reduces the detector's sensitivity to noise, at the expense of losing some of the finer detail in the image. The localization error in the detected edges also increases slightly as the Gaussian width is increased.

In canny edge detection usually, the upper tracking threshold can be set quite high, and the lower threshold quite low for good results. Setting the lower threshold too high will cause noisy edges to break up. Setting the upper threshold too low increases the number of spurious and undesirable edge fragments appearing in the output.

By adjusting the lower and uper thresold values we can reduce the noice and produce better edges in canny edge detection.

- **<u>Xray Image</u>-**

A **multimodal** image of human skull x ray of of resolution **300X300** and gray level 0 to 256 has taken for examine

Input Image

Histogram

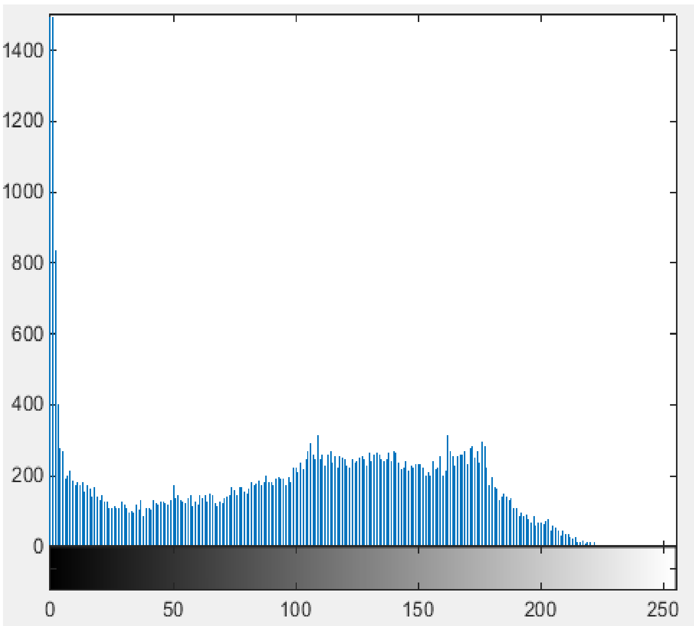


Fig 4.9

Sobel

Fig 4.10

Prewitt

Fig 4.11

Canny

Fig4.12

Robert

Fig4.13

In case of x ray image we can observe that **Canny oparator (fig4.12)** produce strong edges compare to **Robert (fig4.13**), **Sobel (fig4.10)** and **Prewitt (fig4.11).**

In Canny edge detection the presence of Gaussian filter allows removing of any noise in an image.The signal can be enhanced with respect to the noise ratio by non-maxima suppression method which results in one pixel wide ridges as the output.It Detects the edges in a noisy state.By adjusting the thresholding values the noice can be further reduced. It gives a good response and is immune to a noisy environment.

- **Some statistical mausurement are given below**.

Edge detection	Entrophy	PSNR	MSE	Execution time
Sobel	1.2820	11.406	4.7034	1.052911
Prewitt	1.2792	11.3928	4.7185	0.878266
Robert	1.2306	17.1396	1.2564	0.831094
Canny	1.5701	10.9043	5.2803	1.014961

Here Entropy is a concept in information theory. Entropy is used to measure the amount of information. Entropy is defined in terms of the probabilistic behavior of a source of information.

The mean-square error (MSE) and the peak signal-to-noise ratio (PSNR) are used to compare image compression quality. The MSE represents the cumulative squared error between the compressed and the original image, whereas PSNR represents a measure of the peak error. The lower the value of MSE, the lower the error.

Here we can also see than the PSNR and MSE value of **Canny edge** detection is better than others. A higher entropy indicates than Canny oparator provides more information than any other edge detection.But due to complexity it consume more time than Prewitt and Robert but still in this particular image it comsumming less time than Sobel.

- **Texture Image-**

A **unimodal** image of texture of resolution **300X300** and gray level **0 to 256** has taken.

Original Image

Histogram

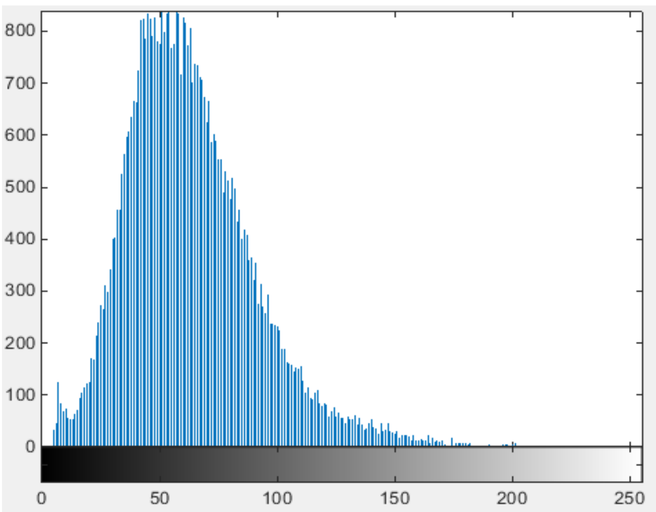
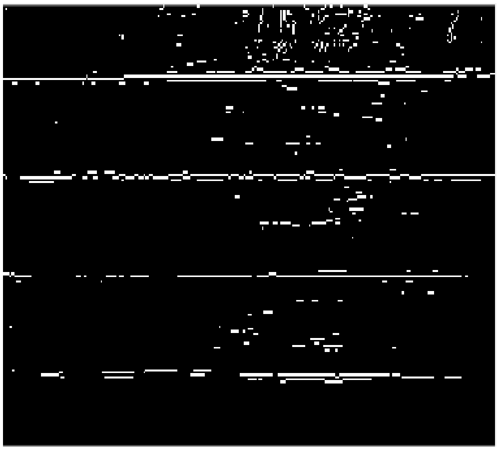


Fig 4.14

Sobel **Prewitt**



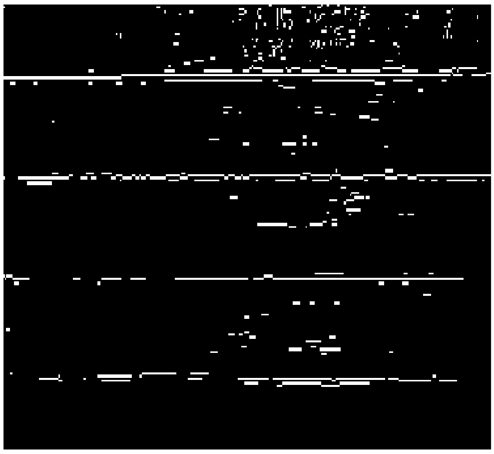
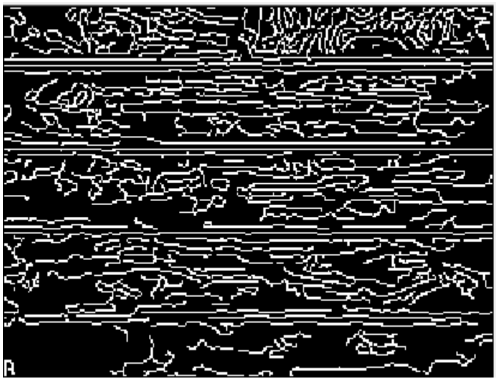


Fig 4.15 **Fig 4.16**

Canny **Robert**



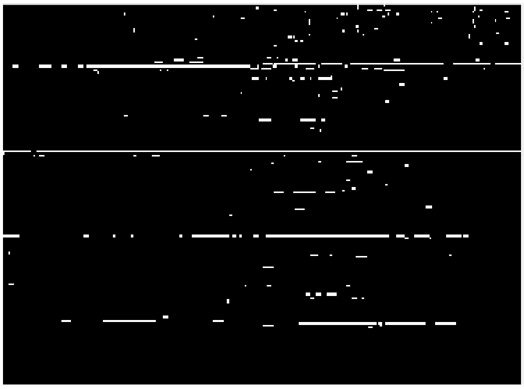


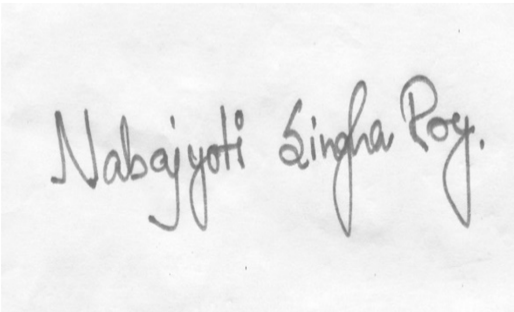
Fig 4.17 **4.18**

Here in fig 4.15 and 4.16 we can see the **sobe**l and **prewitt** edge detection respectively producing better result compare to **robert** edge detection(fig4.18). In fig 4.17 the detected edges by **canny** operator is strong comapre to others but due to higher sensitivity, false edges have been detected which can be rectify by adjusting the thresold values of canny opetaror .

- ## Signature Image-

A **unimodal** signature image of resolution **300X300** and gray level **0 to 256** has taken.

Original



Histogram

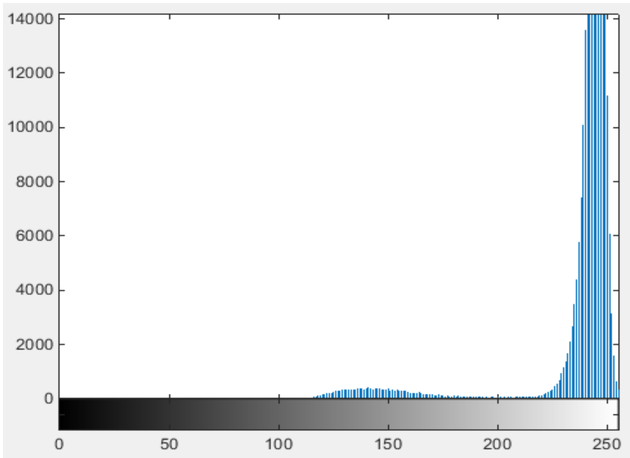


Fig 4.19

Sobel

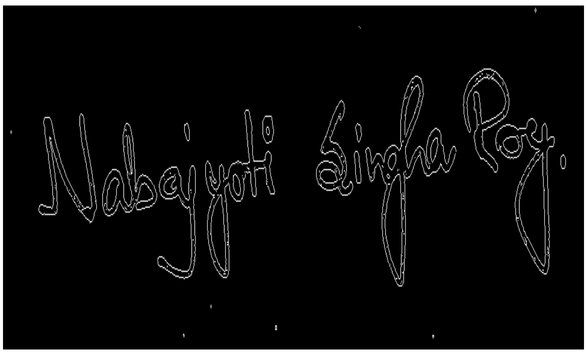


Fig 4.20

Prewitt

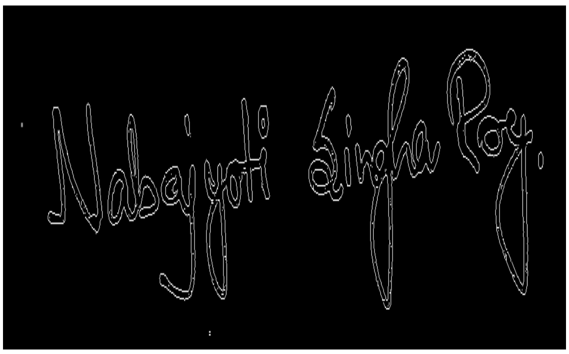


Fig 4.21

Canny **Robert**

Fig 4.22

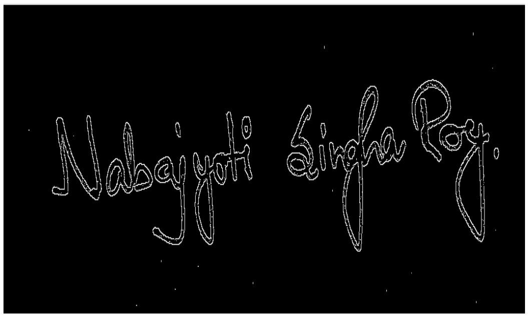


Fig 4.23

In this signature image the results of **sobel** (fig 4.20), **prewitt** (fig 4.21) and **robert** (4.23) are very similar to each other. Each of the edge detection producess proper edges of the input image.

In this signature image as the result of sobel, prewitt and robert are same then it is better to use robert edge detection for less complexity and less coputation time.

- <u>**Blur Image**</u>

  A **mutimodal** blurred image of resolution 300X300 and gray level 0 to 256 has taken for examine.

  Shown in the next page

Input Image

Histogram

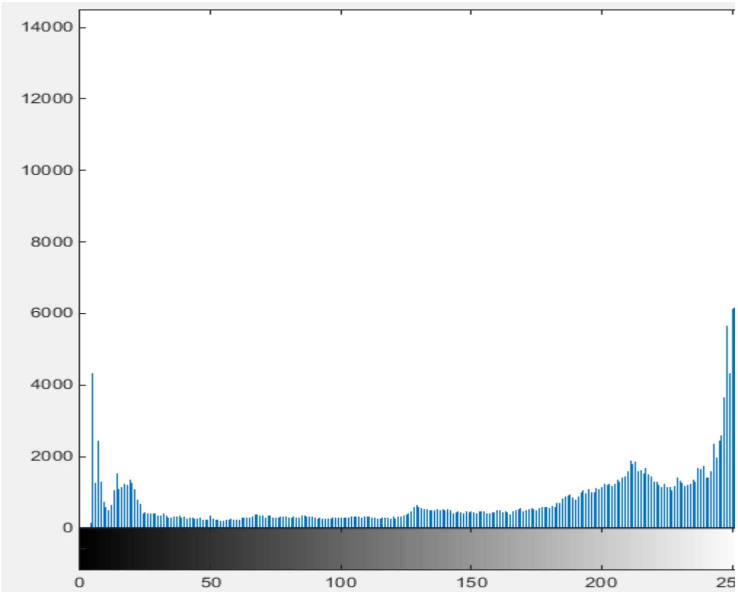


Fig 4.24

Sobel

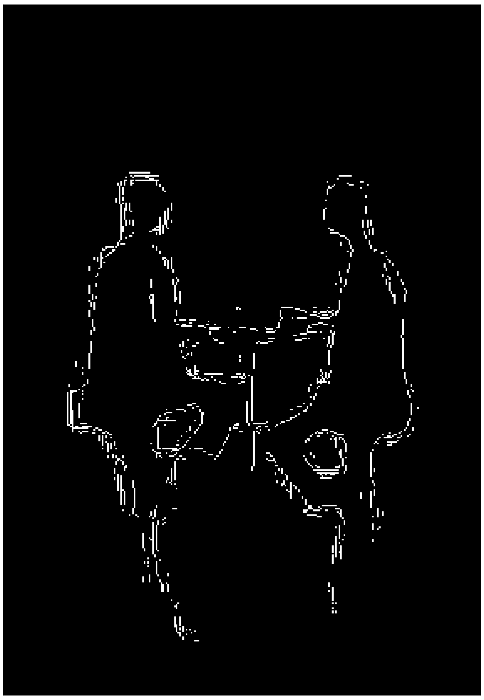


Fig 4.25

Prewitt

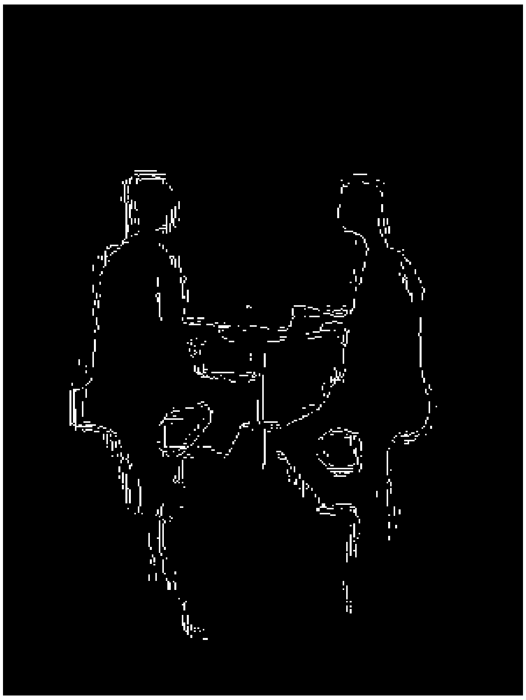


Fig4.26

Robert **Canny**

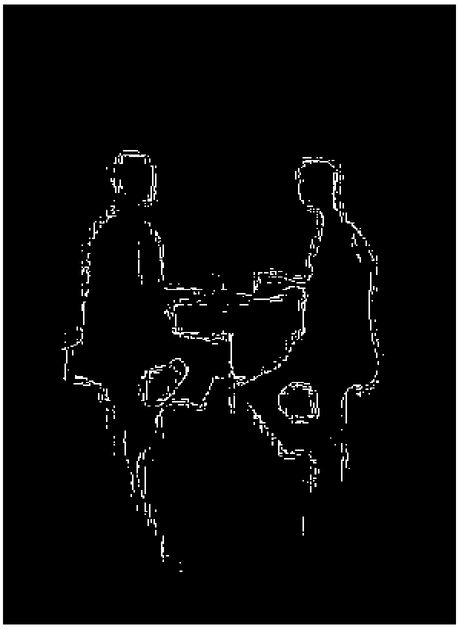


Fig 4.27

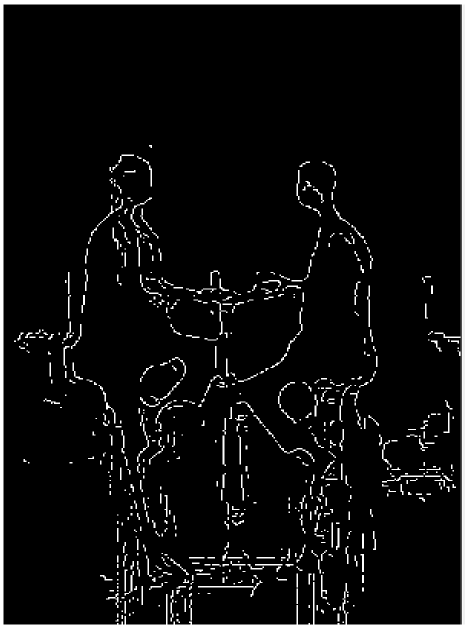


Fig 4.28

In this picture we can see the edge detection of **sobel (fig4.25**), **prewitt (4.26)** and **robert (fig 4.27)** are similar. The edges of the two man in this blurr image is clearly visible and the background also nicely removed in sobel, prewitt and robert edge detection.

Where as the result of **canny adge detection (fig 4.28**) is not as good as others.The edges of the two man is not clear and some background aslo captured in canny edge detection.

4.3 Results applying frequency domain high pass filtering

- **Human Portrait**

Input Image

Fig 4.29

Hpf (cut off freq >10)

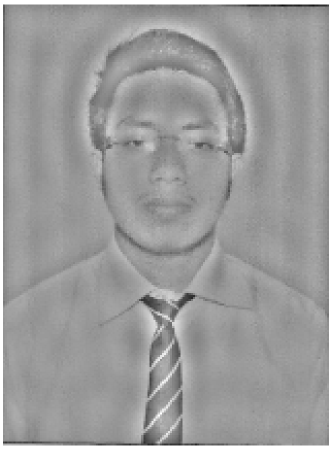


Fig 4.30

Hpf (cut off freq >30)

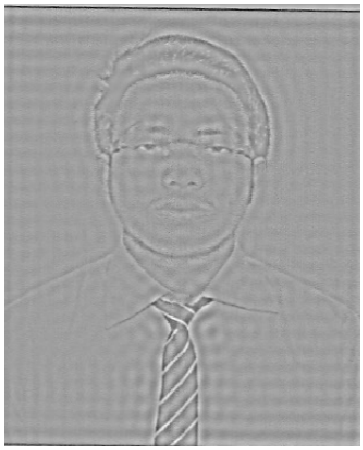


Fig 4.31

The main purpose of highpass filtering is to block the low frequencies and pass the high frequencies signals of the image. As a result of attenuating (or blocking) the low frequencies, areas of constant intensity in the input image are zero in the output of the highpass filter. Areas of a strong intensity gradient, containing the high frequencies, have positive and negative intensity values in the filter output.

In fig 4.30 we can see middle gray value for low frequency areas and dark and light values for the edges.

We can observe the Ringing effect (*Ringing effect so known as Gibbs phenomenon in mathematical methods of image processing is the annoying* *effect in images and video appeared as rippling effect near sharp edges. This effect is caused by distortion or loss of high frequency information in image*) in fig 4.30.

To reduce the effect we increase the cut off frequency greater than 30 shown in fig 4.31.

Applying high pass filtering on different images.

- **X ray image**

Input image **Hpf (cut off freq >10)** **Hpf (cut off freq >30)**

Fig 4.32

Fig 4.33

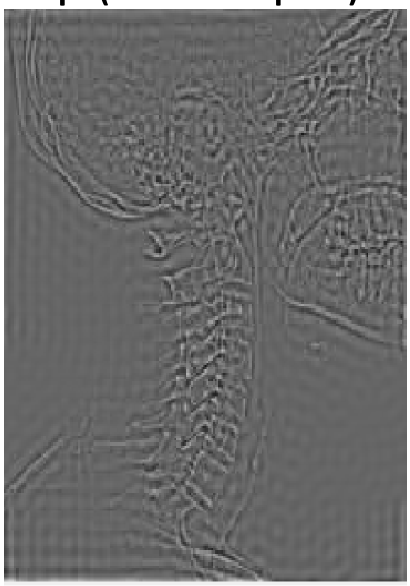


Fig 4.34

- **Texture image**

Input image **Hpf (cut off freq >10)** **Hpf (cut off freq >30)**

Fig 4.35

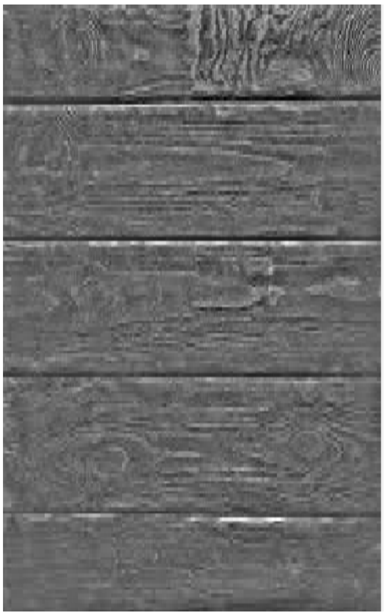


Fig 4.36

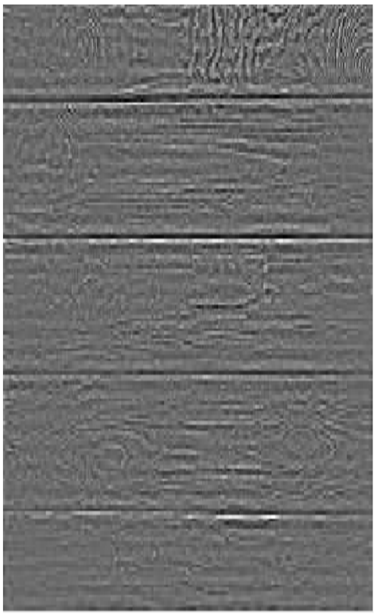


Fig 4.37

- **Signature image**

Input image

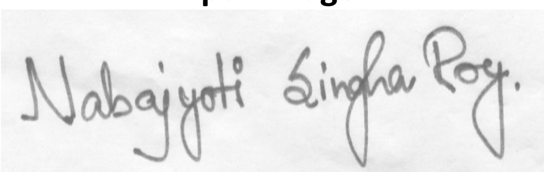


Fig 4.38

Hpf (cut off freq >10)

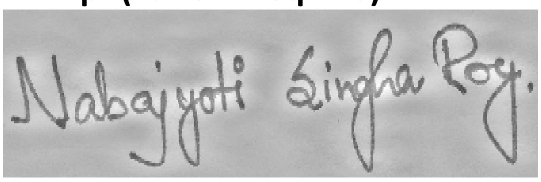


Fig 4.39

Hpf (cut off freq >30)

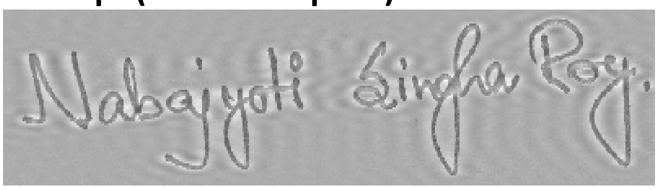


Fig 4.40

- **Blurred image**

Input image

Fig 4.41

Hpf (cut off freq >10)

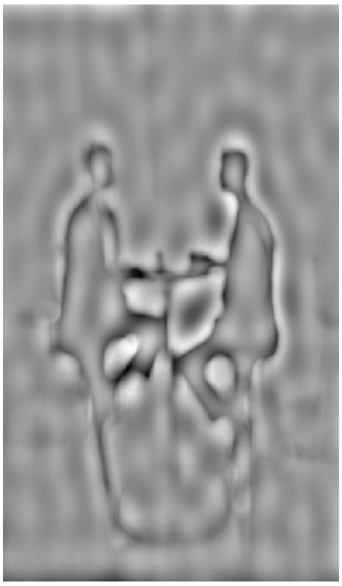


Fig 4.42

Hpf (cut off freq >30)

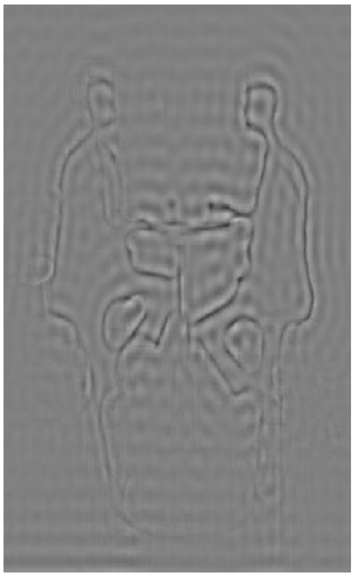


Fig 4.43

4.4 Results of edge detections applying histogram equalization

Here we have applied histogram equalization and then detected the ages of image for better accomplishment of the result.

- **X ray image**

Input image

Fig 4.4.1

Pixel density before histogram equalization

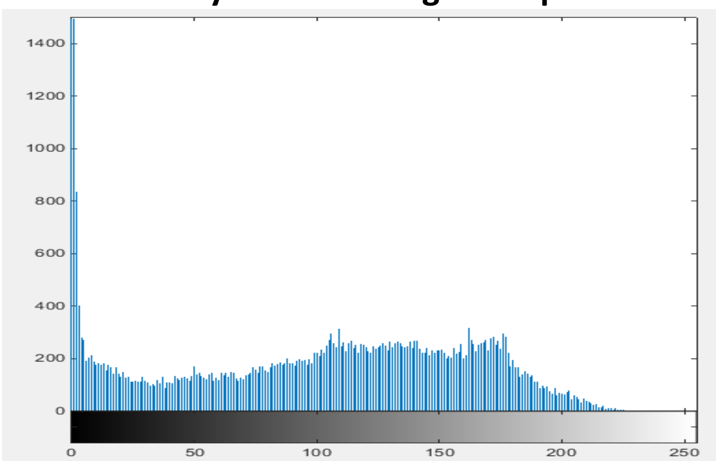


Fig 4.4.2

After histogram equalization **Pixel density after histogram equalization**

Fig 4.4.3

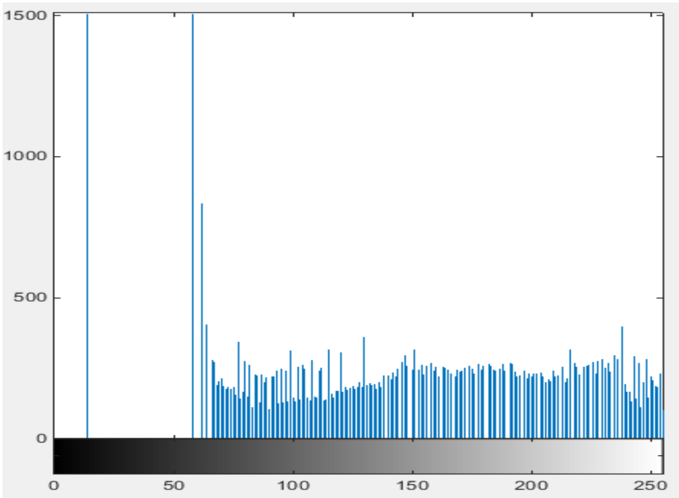


Fig 4.4.4

In the above **figure4.4.2**, X-axis represents the tonal scale (black at the left and white at the right), and Y-axis represents the number of pixels in an image. Here, the histogram shows the number of pixels for each brightness level (from black to white), and when there are more pixels, the peak at the certain brightness level is higher.

After histogram equalization in **fig 4.4.4** we can see that the pixels are equality distributed along all the gray levels (0 to 256). This allows for areas of lower local contrast to gain a higher contrast and a higher contrast image produces good result in edge detection.

Applying edge detection techniques after histogram equalization:

Shown in the next page

> ### ➢ Sobel

Input image | **Before Histogram equalization** | **After Histogram equalization**

Fig 4.4.5

Fig 4.4.6

Fig 4.4.7

In **fig 4.4.7** we can observe that the noise around the eye and teeth of the skull image has much reduced after histogram equalization as compared to **fig 4.4.6** which is obtain before histogram equalization by sobel operator.

Similarly we have applied other edge detection techniques and observed the difference.

➢ Prewitt

Input image	Before Histogram equalization	After Histogram equalization

Fig 4.4.8

Fig 4.4.9

Fig 4.4.10

➢ Robert

Input image	Before Histogram equalization	After Histogram equalization

Fig 4.4.11

Fig 4.4.12

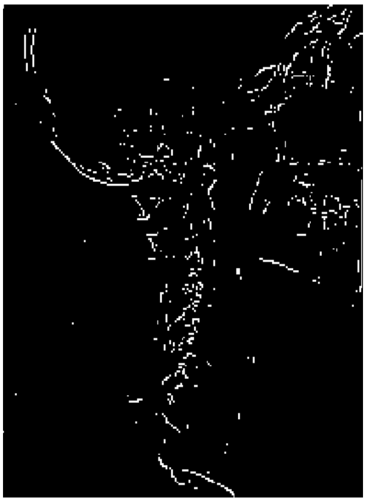


Fig 4.4.13

➢ Canny

Input image	Before Histogram equalization	After histogram equalization

Fig 4.4.14

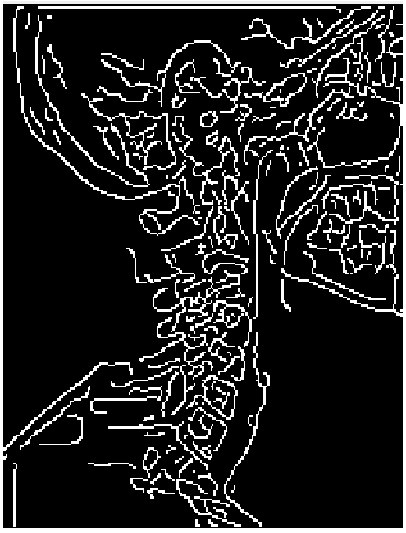


Fig 4.4.15

Fig 4.4.16

In fig 4.4.16 we can see the result of canny after histogram equalization is better than fig 4.4.15 which is without histogram equalization. The noise around neck and eye area is much reduced.

But due to boost the contrast level of background it detects some false edges from background of the image.

➢ Blurred image

Input image

Fig 4.4.17

Pixel density before histogram

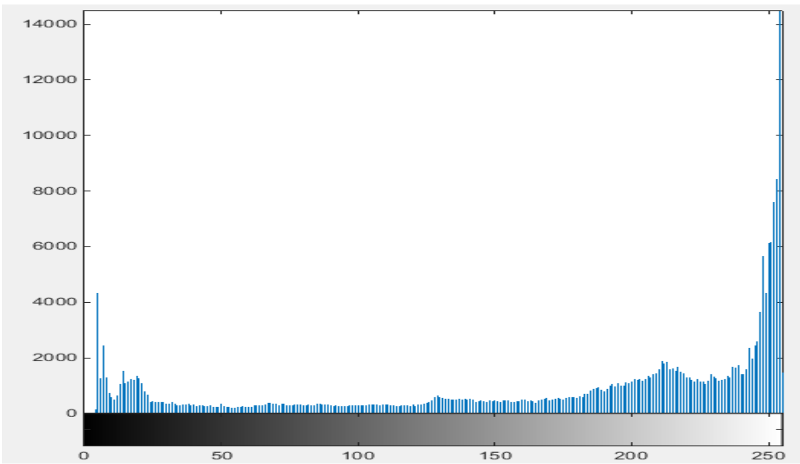


Fig 4.4.18

After histogram equalization

Fig 4.4.19

Pixel density after Histogram equalization

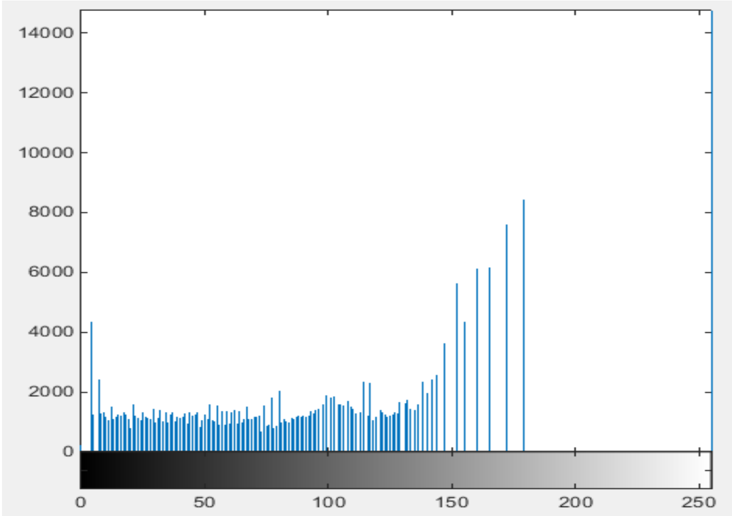


Fig 4.4.20

Applying edge detection techniques after histogram equalization

- **Sobel**

Before Histogram equalization

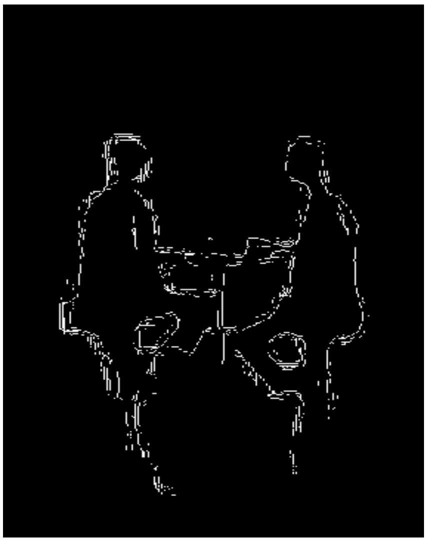


Fig 4.4.21

After Histogram equalization

Fig 4.4.22

In fig 4.4.20 in case of blurred image we can see that after histogram equalization the low contrast area has been boosted. Due to this the background of the image is merged with the subject.

As a result the sobel edge detection after histogram equalization in fig 4.4.22 could not properly detect the edges whereas sobel edge detection without histogram equalization in fig 4.4.21 is given better result.

In fig 4.4.23 high pass filtering without histogram equalization producing far better result than after histogram equalization in fir 4.4.30.

- **Highpass filtering (cut off freq >30)**

Before histogram equalization

Fig 4.4.23

After histogram Equalization

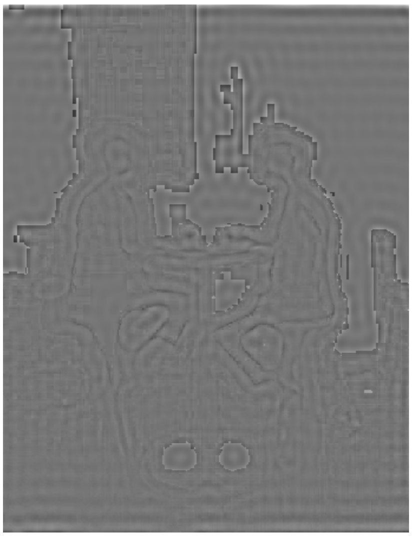


Fig 4.4.24

In fig 4.4.24 we can see that by applying histogram equalization the lower contrast region of the image has been boosted so when we applied highpass filtering it could not able to detect the edges clearly. Histogram equalization converting the lower frequency signal of the image into higher frequency as a result high pass filter unable to cut of the lower frequencies and producing poor result.

➢ Comparison:

Image	Robert	Sobel	Prewitt	Canny	Highpass Filtering
Human Portrait image	Due to small kernel the detected edges are very weak.	Detected edges are very strong as it uses large kernel.	Producing strong and clear edges.	Due to higher sensitivity detect strong edges but produce more noise.	Clear and smooth edges has been detected and by increasing cut off frequency ringing effect reduced.
X ray image	Produces weak edges	Detected edges are very clear	Detected edges are clear.	Detected edges are more clear compare to others.	Produces clear and smooth edges.
Texture image	Strong edges have been detected with low noise.	Detected edges are very clear but producing some noise.	Clear edges with some noise.	Detected false edges and produced poor result.	Clear and smooth edges
Signature image	Due to Bimodal histogram it produced clear and strong edges.	Strong edges	Strong edges	Detect false edges due to higher sensitivity.	Clear edges.
Blurred image	Detected edges are very weak.	Edges are weak and not smooth.	Detected edges are weak and poor.	Can't distinguish the background and produced false edges.	It produced better result compare to all spatial domain technique. Nice and smooth edges have been detected clearly.

Chapter 5
Conclusion and future scope

Edge detection is the initial step for object recognition. We have different type of operator for edge detection. All techniques and algorithm have their own advantages and disadvantages. The analysis of various edge detection techniques are done on the basis of certain parameters.

We have observed that in case of bimodal and unimodal images sobel and prewitt operators are producing much better result than canny operator. But when it comes to multimodal image, edges detected by canny operator are far better than other operators.

For particularly blurred image the results after high pass filtering in frequency domain is better than any other spatial domain techniques.

In the results we have seen that using histogram equalization we can enhance the quality of detected edges since it is a challenging task to the research communities to detect the exact image without noise from the original image.

Edge detection is an important field in image processing, machine vision and computer vision, particularly in the areas of feature detection and feature extraction, which aims at identifying points in a digital image at which the image brightness changes sharply or, more formally, has discontinuities.

The future scope will be to study the reasons for this in detail and improve the image enhancement and filtering method to produce strong and smooth edges without affecting the highlighting of true edge.

➢ References

1. Gonzalez R. C. and Woods R. E., “Digital Image Processing”, Third Ed., Prentice Hall, 2002.
2. Canny J. “A Computational Approach to Edge Detection,” IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 8, No. 6, pp. 679-698, 1986.
3. Cui F. Y., Zou L. J. and Song B., “Edge Feature Extraction Based on Digital Image Processing Techniques”, International Conference on Automation and Logistics, IEEE, pp. 2320-2324, 2008.
4. Deng C., Ma W. and Yin Y, “An Edge Detection Approach of Image Fusion Based on Improved Sobel Operator”, 4th International Congress on Image and Signal Processing, IEEE, pp. 1189-1193, 2011.
5. Nadernejad E., Sharifzadeh S. and Hassanpour H., “Edge Detection Techniques: Evaluations and Comparisons”, Applied Mathematical Sciences, Vol. 2, No. 31, pp.1507- 1520, 2008.
6. Niblack W., “An Introduction to Digital Image Processing”, Prentice Hall International, 1986.
7. Zheng Y., Rao J. and Wu L., “Edge Detection Methods in
8. Digital Image Processing”, The 5th International Conference on Computer Science & Education, IEEE, pp. 471-473, 2010.

9. Wang B. and Fan S. S., “An improved CANNY edge detection algorithm”,Second International Workshop on Computer Science and Engineering, IEEE, pp. 497-500, 2009.
10. Prewitt J. M. S., "Object enhancement and extraction," Academic Press, B. Lipkin, A. Rosenfeld, (Eds.): Picture Processing and Psychopictorics, New York, pp. 75-149, 1970.