DIGITAL IMAGE ENHANCEMENT

Project Report submitted in partial fulfillment of the requirement for the degree of

Bachelor of Technology

In

Electronics and Communication Engineering

Under the Supervision of

Ms. Pragya Gupta

By

Himanshu Garg (081062)

Gaurav Mishra (081064)

Shikhar Sharma (081086)

to

JAYPEE UNIVERSITY OF INFORMATION TECHNOLOGY

Jaypee University of Information and Technology

Waknaghat, Solan – 173234, Himachal Pradesh

Certificate

This is to certify that project report entitled "**DIGITAL IMAGE ENHANCEMENT**", submitted by Himanshu Garg(081062), Gaurav Mishra(081064) and Shikhar Sharma(081086) in partial fulfillment for the award of degree of Bachelor of Technology in Electronics and Communication Engineering to Jaypee University of Information Technology, Waknaghat, Solan has been carried out under my supervision.

P Gupta 28/05/2012

Date: 28/05/2012

Ms. Pragya Gupta
(Sr.Lecturer)

This work has not been submitted partially or fully to any other University or Institute for the award of this or any other degree or diploma.

H Garg.

1. Himanshu Garg
(081062)

Gaurav Mishra

2. Gaurav Mishra
(081064)

Shikhar.

3. Shikhar Sharma
(081086)

Acknowledgement

It has been wonderful and intellectually stimulating working on **Digital Image Enhancement** which is use to enhance the features and to bring out the intricate details of an image.

We gratefully acknowledge the management and administration of Jaypee University Of Information Technology for providing us the opportunity and hence the environment to initiate and complete the project.

We would like to extend our thanks to our project guide Ms. Pragya Gupta and Dr. Vinay Kumar for providing us with the finest details of the subject. They have provided us the way to get the job done , not providing the exact way to do it, but the concept behind the complexities so that we can make better use of our existing knowledge and build up higher skills to meet the industry needs.

Date 28/05/2012

Himanshu Garg

Gaurav Mishra

Shikhar Sharma

Table of Contents

List of Figures

Abstract

The basic aim of our project '**DIGITAL IMAGE ENHANCEMENT**' is to develop an algorithm to enhance the features of a 24-bit bmp digital image. The objective is to remove various artifacts, noise for better viewing; and, to bring out intricate details of an image which are otherwise distorted or blurred. We have used MATLAB software to execute the algorithm and show various image outputs depending on the values of variables used in the algorithm. In our project we have developed a user friendly algorithm where user can use various facets of our program like Brightness enhancement, Contrast enhancement, Global histogram equalization, Local histogram equalization on user defined region and Filter for sharpening and smoothing purpose. Here user can choose any order for applying these individual algorithms and get the most enhanced image based on his perception. User will have a choice to enter his own values of variables and check for himself the different images to get best output. To make our program more user- friendly we have implemented Graphical User Interface as well. We have made different tabs each having a specific purpose. So now the user just has to press the tab and enter the region on which the operation is to be applied. Our project is strictly based on human perception and all throughout we have used X-ray images only.

CHAPTER 1

PROJECT OVERVIEW

1.1 AIM

The basic aim of our project **'DIGITAL IMAGE ENHANCEMENT'** is to develop an algorithm to enhance the features of a 24-bit bmp digital image. The objective is to remove various artifacts, noise for better viewing; and, to bring out intricate details of an image which are otherwise distorted or blurred. We have used MATLAB software to execute the algorithm and show various image outputs depending on the values of variables used in the algorithm.

1.2 BACKGROUND

One of the first application of digital images was in the newspaper industry when pictures were sent by submarine cable between London and New York. Introduction of Bartlane cable picture transmission system in early 1920s reduced the time required to transport a picture. Specialized printing equipments coded pictures for cable transmission and then reconstructed them at the receiving end.The early Bartlane systems were capable of coding images in five distinct levels of gray.This capability was increased to 15 levels in 1929.Although above examples involove digital images, they are not considered under digital image processing as computers were not involved in their creation. Thus history of digital image processing is intimately teid to the development of digital computers. Due to storage and computational process the progress has been dependent on the digital computers. The first powerful computers enough to carry out meaningful image processing task appeared in early 1960s.In parallel with space applications these techniques began in late 1960s to be used in medical imaging.The invention in early 1970s, called computerized tomography(CT),is one of the most important events in application of digital processing in medical diagnosis. From 1960s till present,this field has grown vigorously. It is used in broad range of applications.Computers programmers are used to enhance the contrast or code the intensity level into color for easier interpretation of X-Rays and other images used in industry,medicine and biological sciences.

1.3 APPLICATION

Geographers used the same or similar techniques to study pollution patterns from aerial and satellite imagery. Image enhancement and restoration procedures are used to process degraded images of unrecoverable objects or experimental results too expensive to duplicate.In archaeology,image processing methods have successfully restored blurred pictures that were the only available records of rare artifacts lost or damaged after being photographed .In physics and related fields, computer techniques routinely enhance images of experiments in areas such as high energy plasmas and electron microscopy. Similarly successful applications of image processing concepts can be found in astronomy ,biology ,nuclear medicine, law enforcement, defense and industrial applications. The principal energy source for images in use today is electromagnetic energy spectrum. Synthetic images used for modeling and visualization, are generated by computer. Major uses of imaging based on gamma rays include nuclear medicine and astronomical observations. X-rays are the oldest sources of EM radiation used for imaging .The best known use of X-rays is medical diagnostics, but they also are used extensively in industry and other areas, like astronomy. As a final illustration of image processing ,images of fingerprints are routinely processed by computer, either to enhance them or to find features that aid in automated search of database for potential matches. Used in paper currency, it include automated counting, reading of the serial number for the purpose of tracking, in law enforcement and identifying bills.

1.4 INTRODUCTION

An image is an array, or a matrix, of square pixels (picture elements) arranged in columns and rows. In a (8-bit) grayscale image each picture element has an assigned intensity that ranges from 0 to 255. A grey scale image is what people normally call a black and white image, but the name emphasizes that such an image will also include many shades of grey.

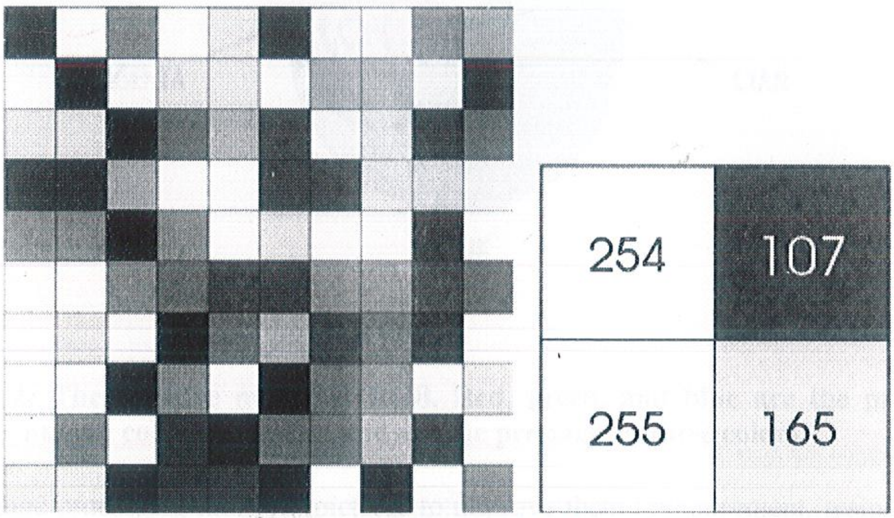


Figure1.4.1

Each pixel has a value from 0 (black) to 255 (white). The possible range of the pixel values depend on the color depth of the image, here 8 bit = 256 tones or grayscales.

Talking about images we always come across a term **RGB.**

The RGB color model relates very closely to the way we perceive color with the **r**, **g** and **b** receptors in our retinas. RGB uses additive color mixing and is the basic color model used in television or any other medium that projects color with light. It is the basic color model used in computers and for web graphics, but it cannot be used for print production.

The secondary colors of RGB – cyan, magenta, and yellow – are formed by mixing two of the primary colors (red, green or blue) and excluding the third color. Red and green combine to make yellow, green and blue to make cyan, and blue and red form magenta. The combination of red, green, and blue in full intensity makes white.

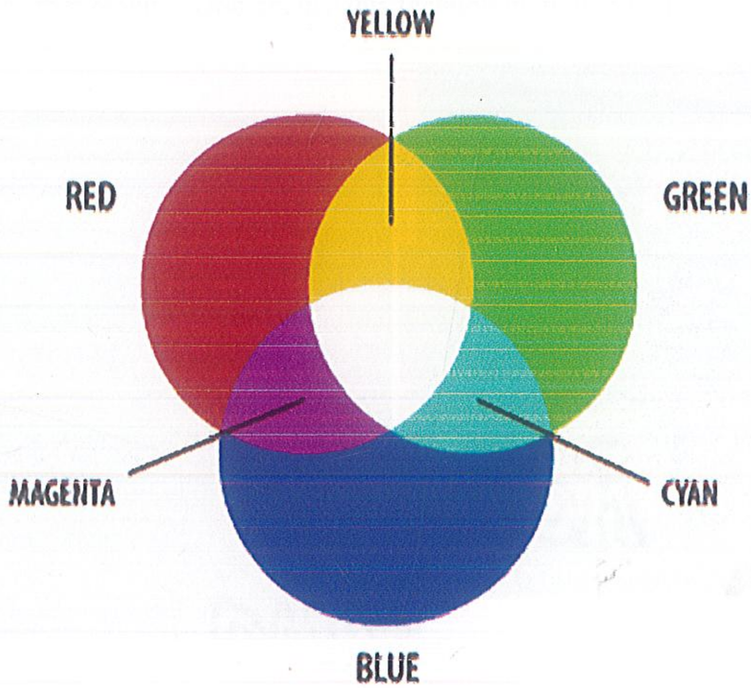
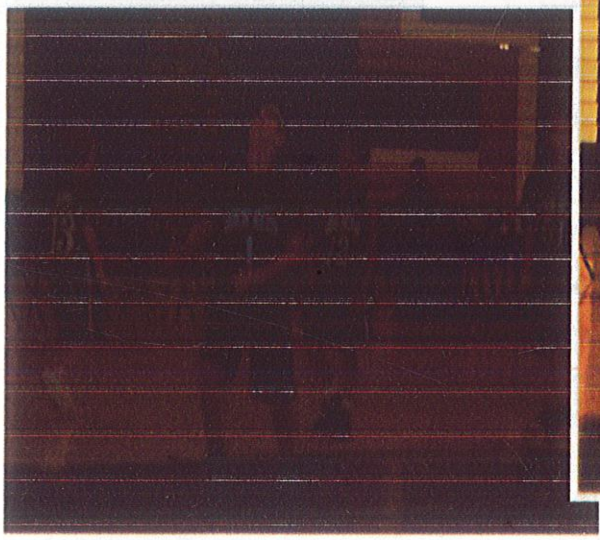


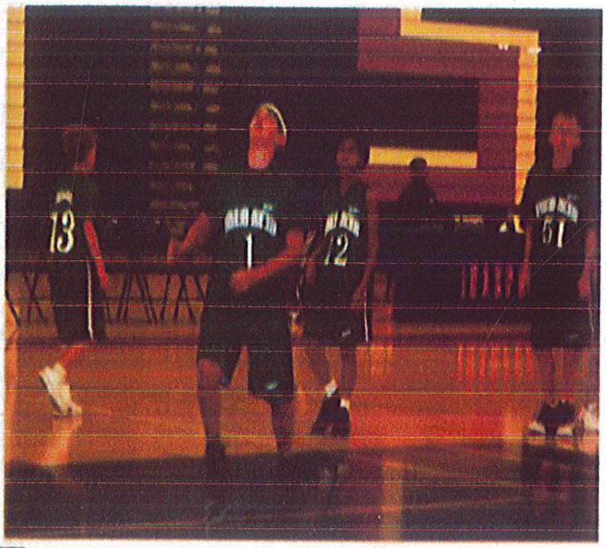
Figure1.4.2: The additive model of RGB. Red, green, and blue are the primary stimuli for human color perception and are the primary additive colors.

Image processing modifies pictures to improve them (enhancement, restoration), extract information (analysis, recognition), and change their structure (composition, image editing). Images can be processed by optical, photographic, and electronic means, but image processing using digital computers is the most common method because digital methods are fast, flexible, and precise. The digitized values are called picture elements, or "pixels," and are stored in computer memory as a digital image. A typical size for a digital image is an array of 512 by 512 pixels, where each pixel has value in the range of 0 to 255. The digital image is processed by a computer to achieve the desired result.

Image enhancement improves the quality (clarity) of images for human viewing. Removing blurring and noise, increasing contrast, and revealing details are examples of enhancement operations. For example, an image might be taken of some cell, which might be of low contrast and somewhat blurred. Reducing the noise and blurring and increasing the contrast range could enhance the image. The original image might have areas of very high and very low intensity, which mask details. An adaptive enhancement algorithm reveals these details. Adaptive algorithms adjust their operation based on the image information (pixels) being processed. In this case the mean intensity, contrast, and sharpness (amount of blur removal) could be in various areas of the image.

Image processing technology is used by planetary scientists to enhance images of Mars, Venus, or other planets. Doctors use this technology to manipulate CAT scans and MRI Images





1.5 IMAGE PROCESSING TECHNIQUES

1.5.1 **Image Histogram** - An **image histogram** is a type of histogram that acts as a graphical representation of the tonal distribution in a digital image. It plots the number of pixels for each tonal value. By looking at the histogram for a specific image a viewer will be able to judge the entire tonal distribution at a glance.

Image histograms are present on many modern digital cameras. Photographers can use them as an aid to show the distribution of tones captured, and whether image detail has been lost to blown-out highlights or blacked-out shadows.

The horizontal axis of the graph represents the tonal variations, while the vertical axis represents the number of pixels in that particular tone. The left side of the horizontal axis represents the black and dark areas, the middle represents medium grey and the right hand side represents light and pure white areas. The vertical axis represents the size of the area that is captured in each one of these zones. Thus, the histogram for a very bright image with few dark areas and/or shadows will have most of its data points on the right side and center of the graph. Conversely, the histogram for a very dark image will have the majority of its data points on the left side and center of the graph.

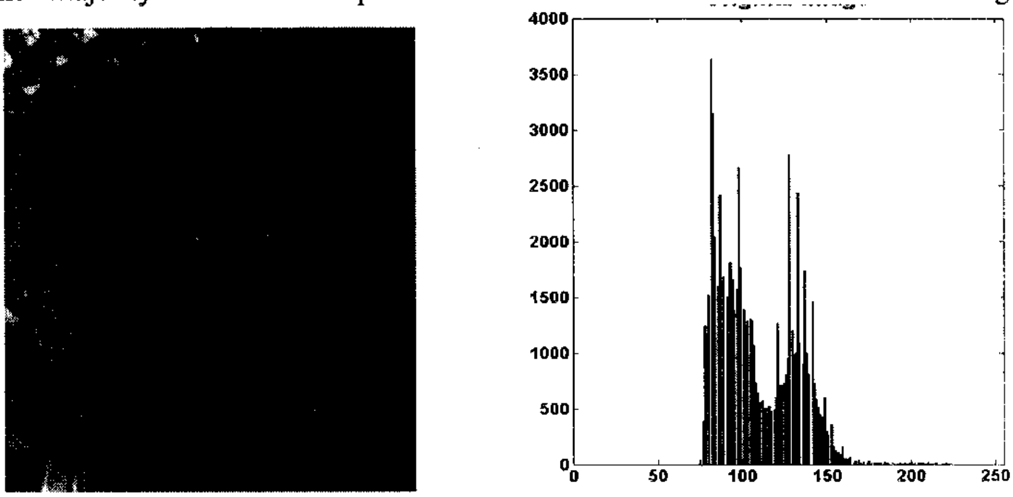


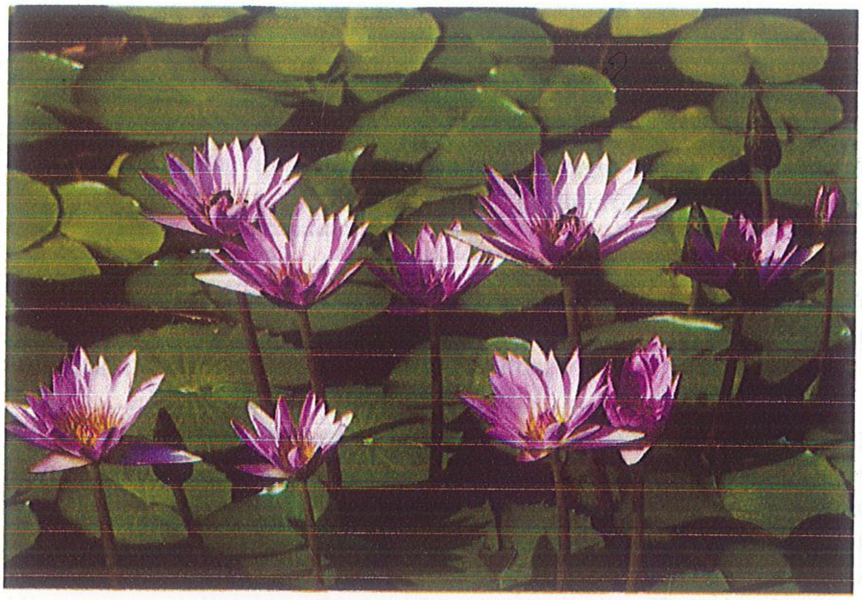
FIGURE 1.5.1

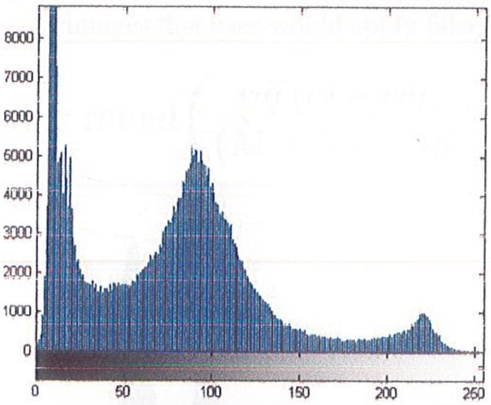
A histogram depicts problems that originate during image acquisition; like,
– Contrast
– Dynamic range
– Artifacts (due to image processing steps)

1.5.2 Cumulative Histogram

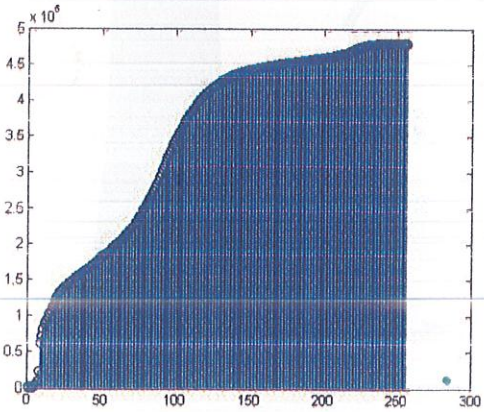
A cumulative histogram is a mapping that counts the cumulative number of observations in all of the bins up to the specified bin. That is, the cumulative histogram M_i of a histogram m_j is defined as:

$$M_i = \sum_{j=1}^{i} m_j.$$





Histogram



Cumulative Histogram

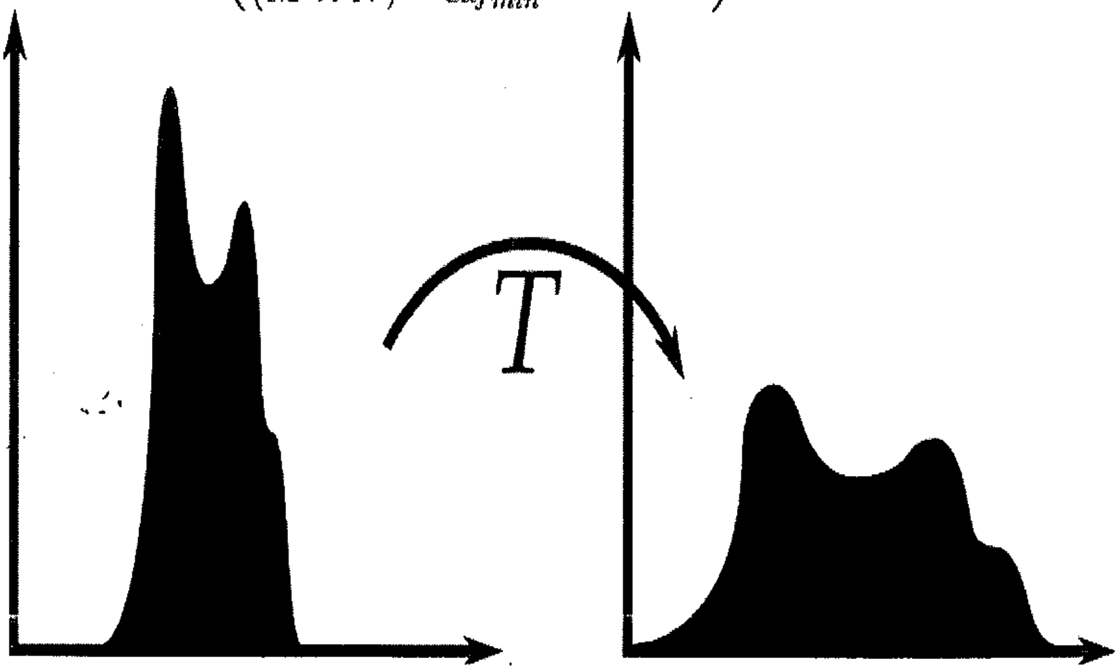
FIGURE 1.5.2

1.5.3 Point Operation - A technique in which the data from an image are digitized and various mathematical operations are applied to the data, generally with a digital computer, in order to create an enhanced image that is more useful or pleasing to a human observer, or to perform some of the interpretation and recognition tasks usually performed by humans. This is also known as picture.

It is an operation where we change the pixel values without changing the size, geometry, or local structure of the image. Every new pixel is dependent only on its previous pixel value, and NOT on any other neighbor value.

1.5.4 Histogram Equalization – It is a method in image process of contrast adjustment using the image's histogram. It is used to modify the histograms of two images in such a way that resulting intensity distributions are similar. It is used to generate images either for comparison or print. The goal is to perform a point operation in such a way that resulting histogram of the modified image is approximately uniform distribution. 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 that user would apply false-color to.

$$h(v) = \mathrm{round}\left(\frac{cdf(v) - cdf_{min}}{(M \times N) - cdf_{min}} \times (L-1)\right)$$



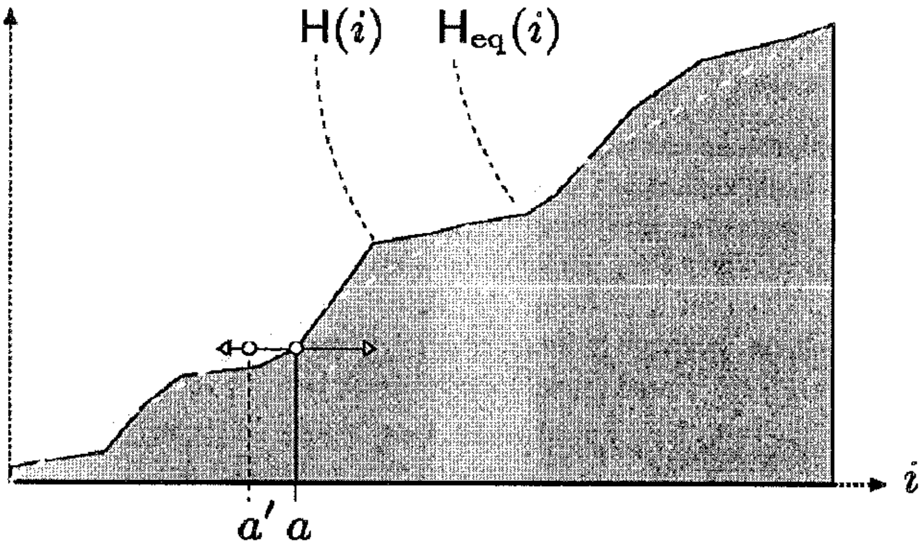


FIGURE 1.5.4

a. **Global Histogram Equalization** modify pixels by transformation functions based on entire image

b. **Local Histogram Equalization** modifies pixel values of a user-defined region and not entire image.

1.6 Recent developments in image processing fields

- They developed a mobile vision assistive device based on a head mounted display (HMD) with a video camera, which provides image magnification and contrast enhancement for patients with central field loss (CFL). Because the exposure level of the video camera is usually adjusted according to the overall luminance of the scene, the contrast of sub-images (to be magnified) may be low. They found that at high magnification levels, conventional histogram enhancement methods frequently result in over- or under-enhancement due to irregular histogram distribution of sub images. Furthermore, the histogram range of the sub-images may change dramatically when the camera moves, which may cause flickering. A piece-wise histogram stretching method based on a center emphasized histogram is proposed and evaluated by observers. The center emphasized histogram minimizes the histogram fluctuation due to image changes near the image boundary when the camera moves slightly, which therefore reduces flickering after enhancement. A piece-wise histogram stretching function is implemented by including a gain turnaround point to deal with very low contrast images and reduce the possibility of over enhancement. Six normally sighted subjects and a CFL patient were tested for their preference of images enhanced by the conventional and proposed methods as well as the original images. All subjects preferred the proposed enhancement method over the conventional method. [1]

- Image quality plays a key role in consumer purchasing decisions. So here they manipulated image quality of videos using a consumer product that enhances digital video images in real time. Videos were presented on two HDTVs, enhanced by varying amounts and subjects made pairwise comparisons. Our results showed two distinct preference groups ("Sharp" and "Smooth") in their study subjects. Preferences for enhancement depended on the video content, particularly the presence or absence of a human face.[2]

CHAPTER 2

FILTERS

The capabilities of perfect point operations are limited so we use filters for better performance. Filters also perform point operations; that is, they also produce 1 pixel for every pixel, but of different type. After any filter operation they do not change the geometry of the image. It use more than one pixel to calculate new pixel value.

Filter are classified into two types

2.1 Linear filter :

It combines the pixel values in the support region as a weighted linear combination.

$$I'(u,v) \leftarrow \frac{1}{9} \cdot \sum_{j=-1}^{1} \sum_{i=-1}^{1} I(u+i, v+j)$$

This averaging operation represents a linear filter. Filters use a set of pixels to generate a new pixel value the new pixel value is at the same location where the reference point was.

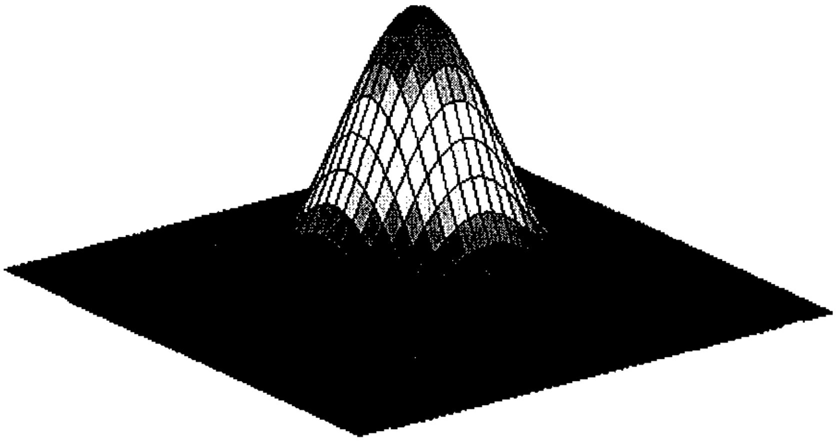
2.2 Non-Linear filter :

It combines the pixel values in the support region as a non linear combination they do not use a linear weight. Two main filters which comes into this category are

- **Gaussian Filter:**

  It is used to remove noise from an image without destroying its fine features.

  This filter also smoothes the image, but it is not as blunt as the average filter It gives more weight to the values near the **center** of the mask and, gradually, less weight to the values away from the center. Smoothing is when Pixels change their values sharply results in edges. So image is blurring if pixels have approximately same value in its proximity. So we use smoothing.



Gaussian function is as shown above. More weight is given near the centre and gradually less weight to the value away from the center

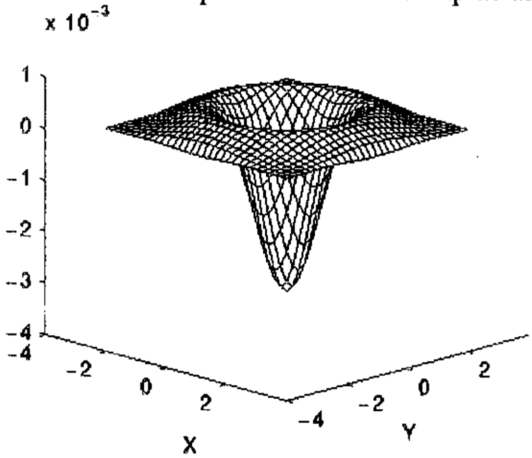
- **Laplacian Filter :**
  The Laplacian of an image highlights regions of rapid intensity change and is therefore often used for edge detection (see zero crossing edge detectors).

0	−1	0
−1	4	−1
0	−1	0

−1	−1	−1
−1	8	−1
−1	−1	−1

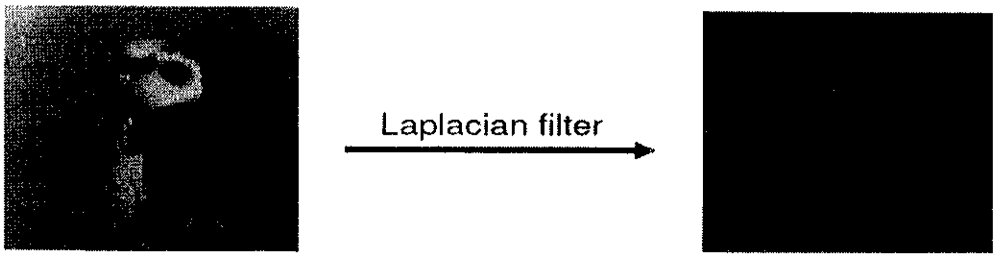
Notice that the Laplacian filter contains negative numbers and the sum of whole matrix is always zero.

The Laplacian filter makes zero the areas of the image where the intensity changes are smooth. In particular, it makes zero the areas of constant intensity. When the variation of the intensities is not smooth, the Laplacian returns non-zero values. The less smooth the variation of the intensities, the higher in absolute value the results of the Laplacian filter. The Laplacian filter detects the edges of the image.



Laplacian Function is as shown above. More weight is given near the centre in the negative direction as center weight is negative in laplacian filter and gradually less weight to the value away from the center

The result of the Laplacian filter on an image



And if we subtract the filtered image from the original one then we will get a sharper image so we can use Laplacian filter as an image sharpener.

Figure2.2

CHAPTER 3

Work Description

Our project is meant to develop an algorithm to enhance the features of a 24-bit bmp digital image. The objective is to remove various artifacts, noise for better viewing; and, to bring out intricate details of an image which are otherwise distorted or blurred. Our project is strictly based on human perception and all throughout we have used X-ray images only.

We are using BMP because -:
1. It is a format used to define device-independent bitmaps in various color resolutions. The main purpose of Device-Independent Bitmaps (DIBs) is to allow bitmaps to be moved from one device to another. A DIB is an external format, in contrast to a device-dependent bitmap, which appears in the system as a bitmap. A DIB is normally transported in BMP files.
2. It has wide range of acceptance in Windows/Linux programs.

3.1 Code meant to either enhance the contrast or the brightness level of the image in a region defined by user. (Refer Appendix [1])

Discussion about code:

Our program is basically meant to either enhance the contrast or the brightness level of the image. Firstly we have stored a colored image in a variable I. It is then converted into grey-scale image. Then we have asked for user choice whether he wants to enhance the contrast or the brightness level of the image. After that we have taken the x and y coordinates as input from user, to define the region on which the operation is to be applied. Then we defined both the operations i.e. contrast and brightness operation and at the end displayed the final image.

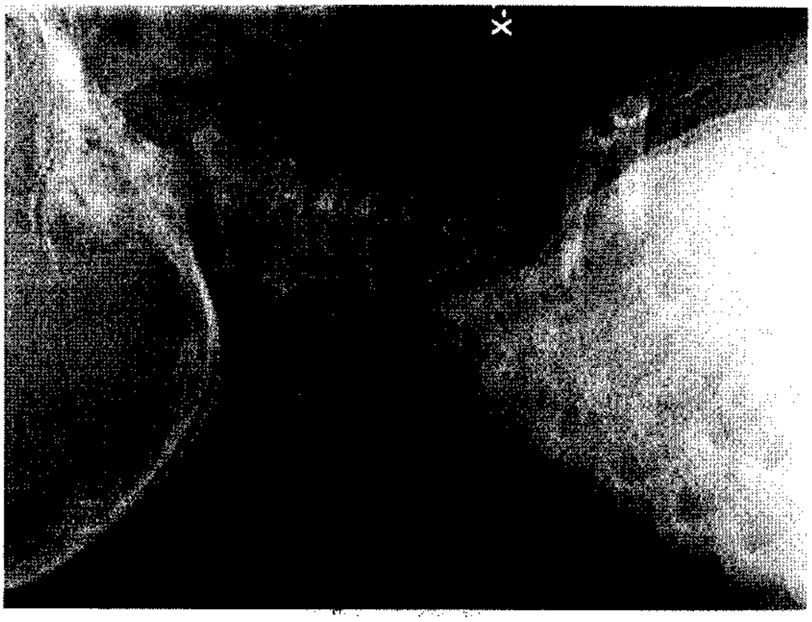


FIGURE 3.1(a)
Original image

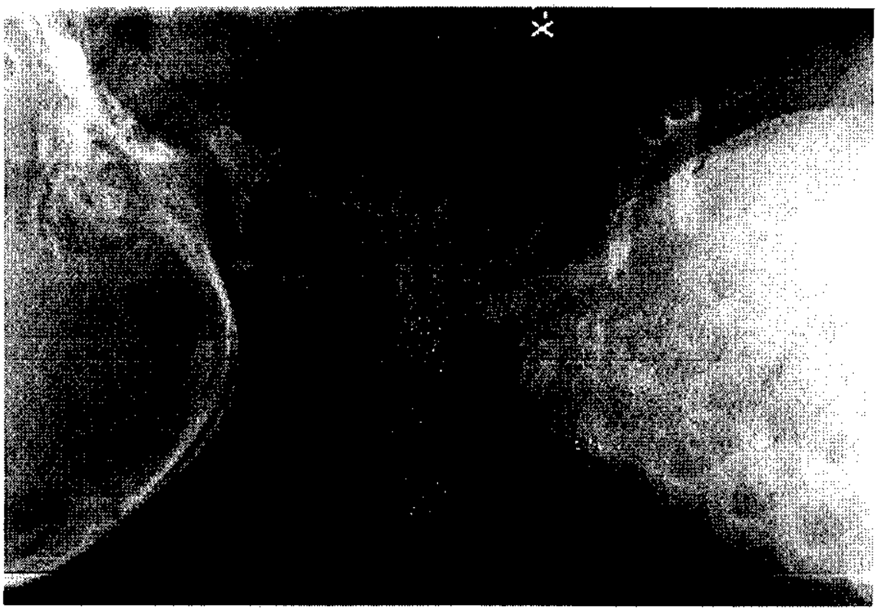


FIGURE 3.1(b)
Output when brightness value=10

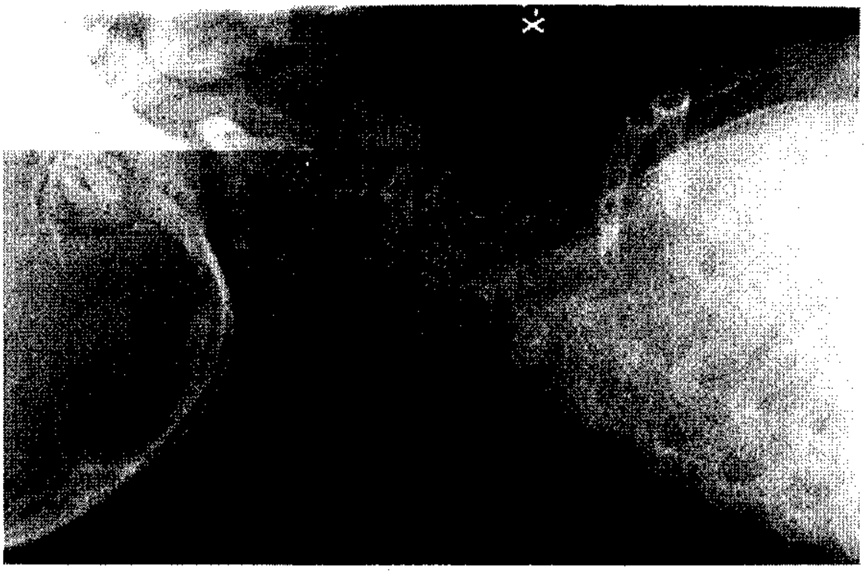


FIGURE 3.1(c)
Output when brightness value=40

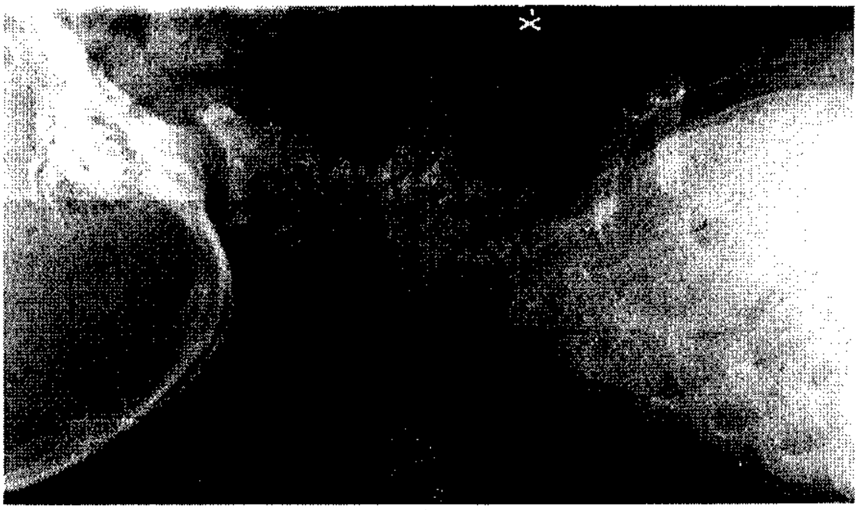
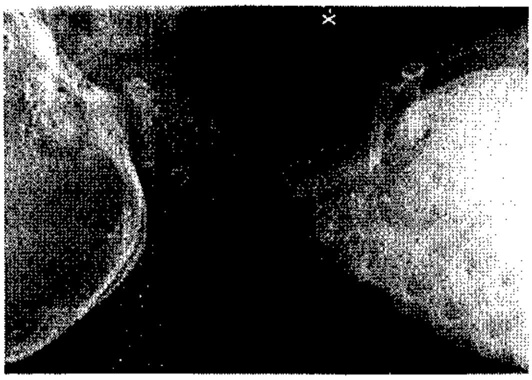


FIGURE 3.1(d)
Output when contrast factor =10

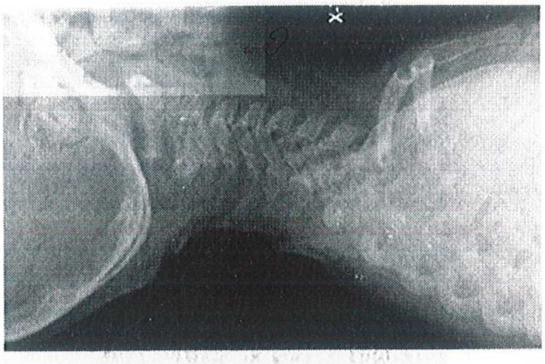
FIGURE 3.1(e)
Output when contrast factor =25

Output Analysis:
In the first case we have shown the contrasting features of the two output images based on the difference in their 'brightness levels'.

1. For value of z=10, we got the following output



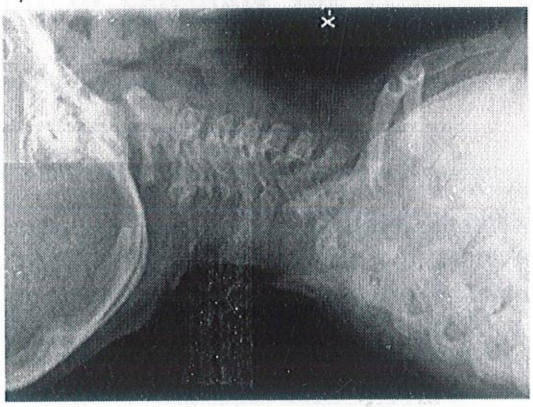
2. For value of z=40, we got the following output.



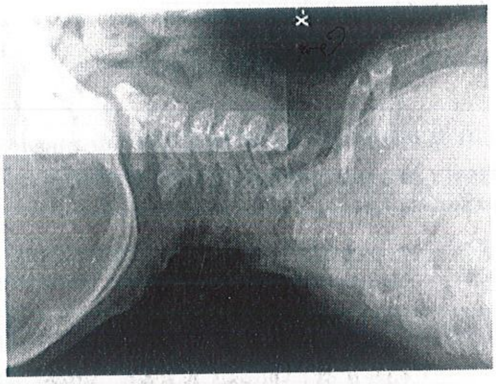
So we can clearly see that the change in the brightness levels is quite evident.

In the second case we have shown the contrasting features of the two output images based on the difference in their 'level of contrast change'

1. For value of z=10, we got the following output



2. For value of z=25, we got the following output

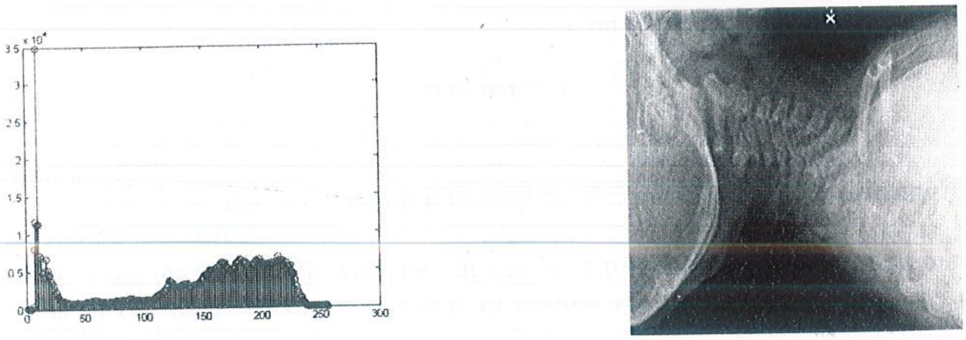


So we see that for value z=80, we get a better contrast image which is brighter as it is more towards higher intensity values of 200-255.

3.2 Program meant to calculate the Histogram of an image (Refer Appendix [2])

Discussion about code:

Here we have calculated the histogram of an image. Firstly we have stored a gray-scale in a variable i. Then we have taken a variable x to represent 1 to 256 intensity levels. After that we have started a looping function for the entire image and used a variable histo to store various pixel values recursively. And these values are displayed using stem function.



Histogram of image FIGURE 3.2 Image

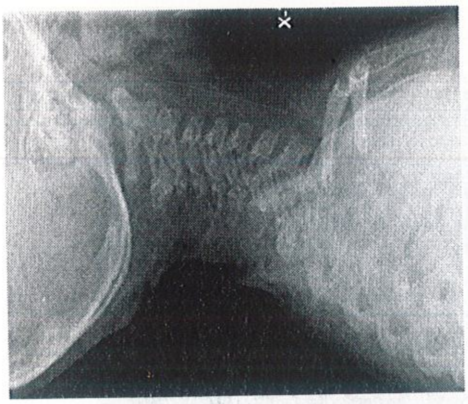
Output analysis:

In the output histogram we see that there are peaks at 0-10 intensity levels which clearly show the presence of pitch black color that we can clearly observe in our image. Moreover there are also some peaks towards the end that represents white color.

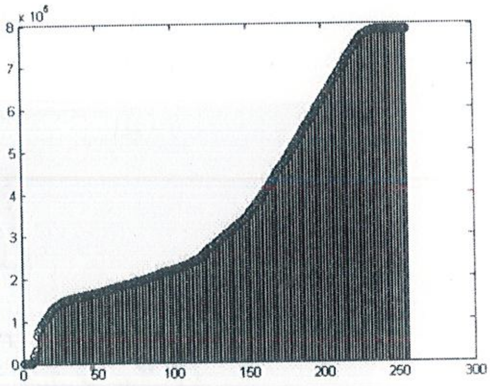
3.3 Program meant to Calculate Cumulative Histogram (Refer Appendix [3])

Discussion about code:

Here we have calculated the cumulative histogram of an image. Here we have stored the image in variable i and calculated the histogram using looping statements and storing pixel values in variable histo. Then we have used a cumsum function to calculate the cumulative histogram of the image and displayed the result.



Image



Cumulative Histogram of Image

FIGURE 3.3

Output analysis:

As we know that a cumulative histogram is a monotonically increasing function so the output is justified.

Also when we compare it with the original histogram here also we see that initially its slope is more which becomes constant in middle region and then abruptly increases towards the end.

3.4 Implementing Global Histogram Equalization (Refer Appendix [4])

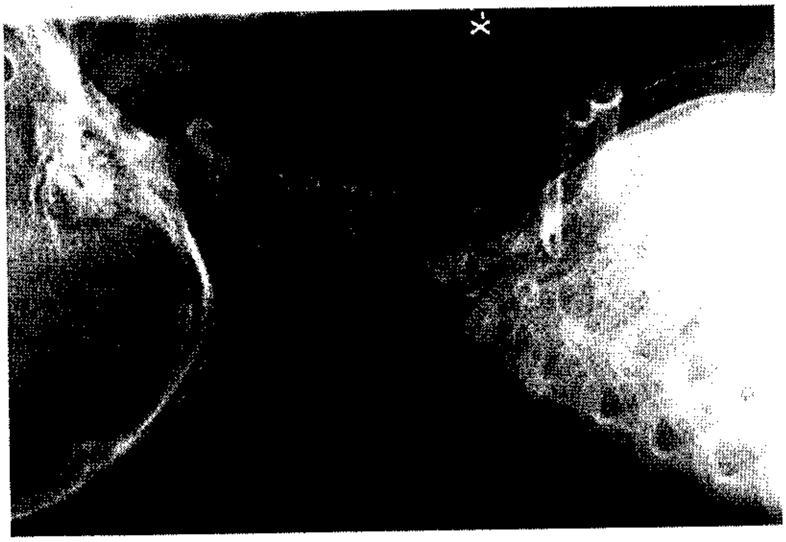
Discussion about code:

Here we have basically implemented Global histogram equalization technique. Here firstly we have stored of the image in variable A. Then we have found out the histogram and the cumulative histogram matrix of the image as in previous cases. After that we have applied the following function on the cumulative histogram matrix to get an equalized matrix.

$$h(v) = \text{round}\left(\frac{cdf(v) - cdf_{min}}{(M \times N) - cdf_{min}} \times (L-1)\right)$$

Where M *N represents the size of the image while L represents the number of intensity values present in the image (In this case it is 256). At the end we have used a reshaping function to create our new image matrix and equalized image.

Original Image



Equalized Image

FIGURE 3.4

Output analysis:

So we can see that the second image i.e. the equalized image has better brightness levels and comparative more enhanced features as compared to the first image

3.5 Implementing Local Histogram Equalization on user-defined region (Refer Appendix [5])

Discussion about code:

In this case instead of equalizing the whole imagé we have equalized only a portion of image i.e. a local region .This technique is called Local histogram equalization. For this purpose firstly we have stored our image in variable e. After that we have taken the x and y coordinates as input from user, to define the region on which the operation is to be applied.

Here what we have done is that we have converted our original image into two images.

Image 1: In this image we have extracted the user-defined portion i.e. to be equalized by multiplying the rest of image with zero.

Image 2: This is the rest of image left after multiplying the user defined portion with zero.

Now we take Image 1 and equalize it using the same method and formula used previously i.e.

$$h(v) = \text{round}\left(\frac{cdf(v) - cdf_{min}}{(M \times N) - cdf_{min}} \times (L-1)\right)$$

After equalizing the image 1, we superimpose image 1 and image 2 to get our final output image.

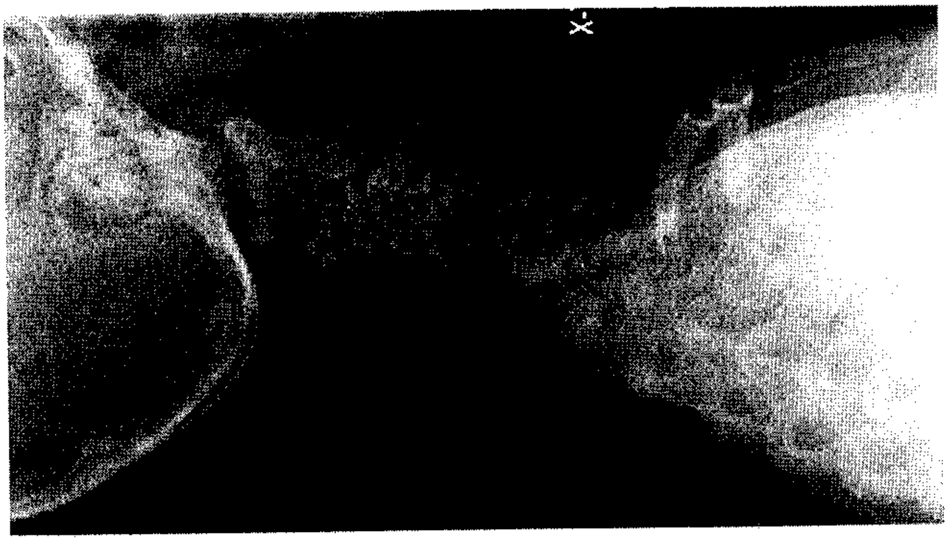


FIGURE 3.5(a)
Original Image

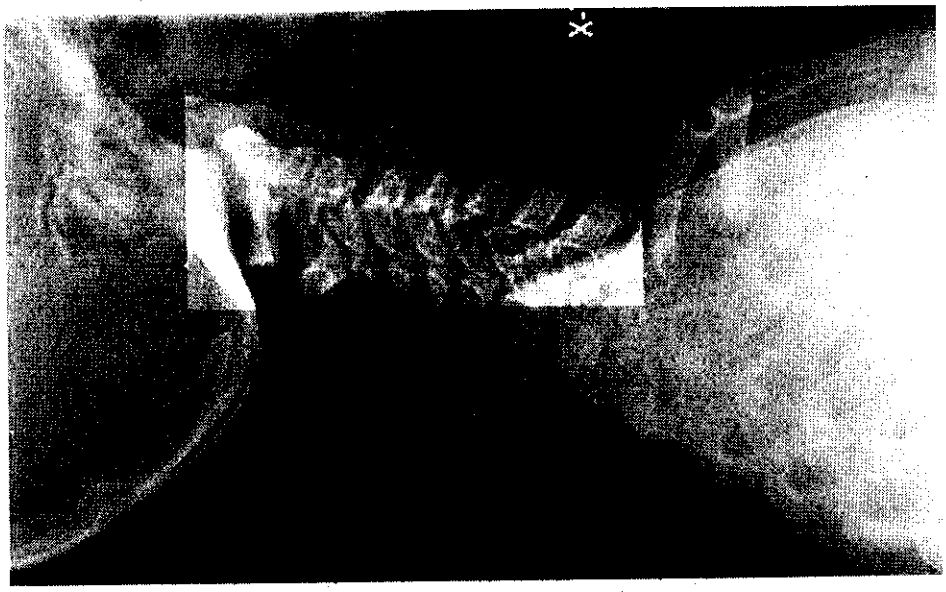


FIGURE 3.5(b)
Equalized Image

Output Analysis:

Here we have input the coordinates as: i=400 and j=700 and we can clearly see the output image being equalized within that region only. The output image is clearer and the features are enhanced as compared to original one within that region

3.6 Program meant for Image Sharpening (Refer Appendix [6])

Discussion about code:

Here we have sharpened the image with the use of Laplacian filter. The Laplacian of an image highlights regions of rapid intensity change and is therefore often used for edge detection. And if we subtract the filtered image from the original one then we will get a sharper image so we can use Laplacian filter as an image sharpener

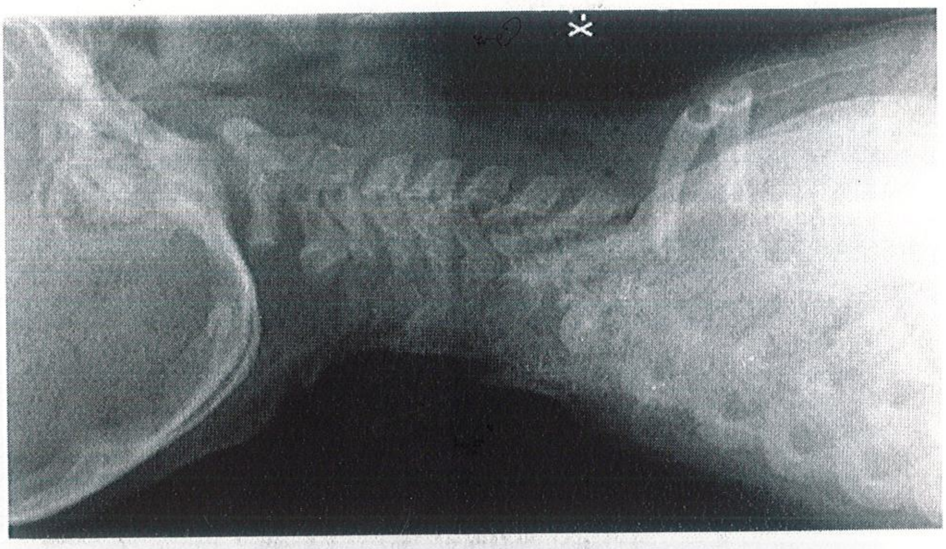


FIGURE 3.6(a)
Original Image

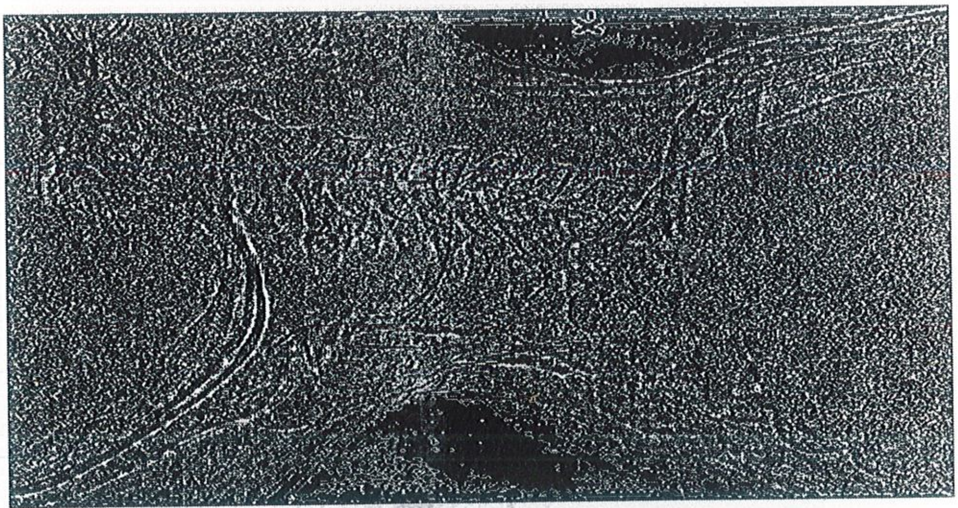


FIGURE 3.6(b)
Output after Laplacian filter

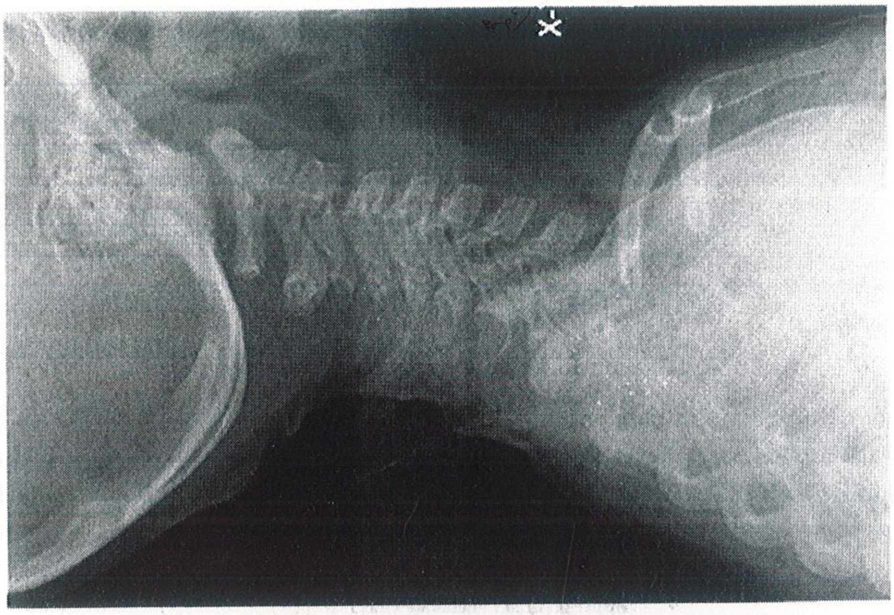


FIGURE 3.6(c)
Final Output Image Using [0 1 0; 1 -4 1; 0 1 0]

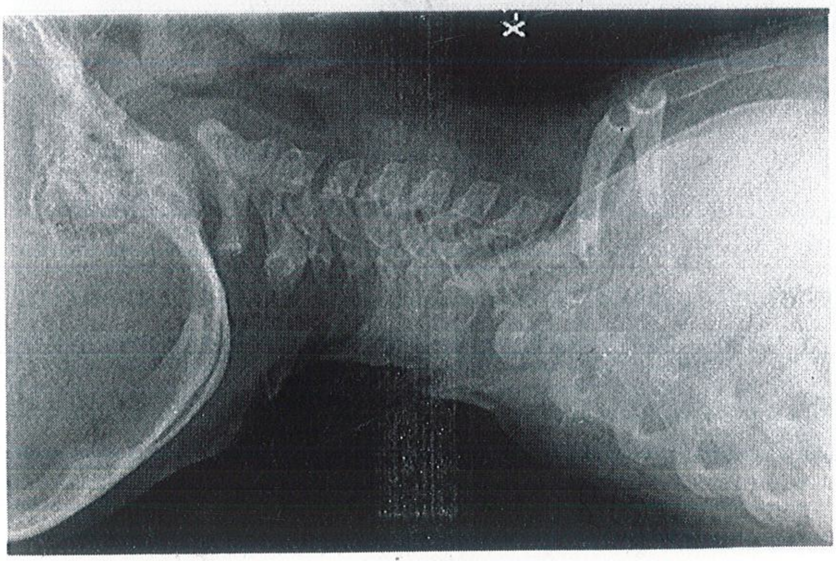


FIGURE 3.6(d)
Final Output Image Using [0 2 0; 2 -8 2; 0 2 0]

Output Analysis:

Here we have the output image being sharpen. The output image is clearer and the features are enhanced as compared to original one.

3.7 Menu Driven (Refer Appendix [7])

Discussion about code:

Here we have combined all the codes into a single menu driven code using looping and conditional statement. User has to give an input only once and he/she can apply different function or algorithm on that image. This code provides us the facility that one can apply different functions over the same image and check the output using different combination.

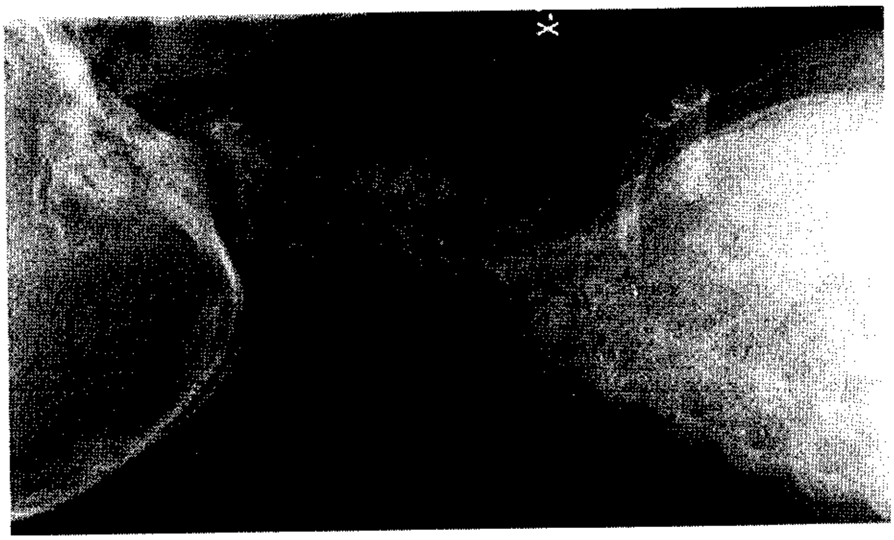


FIGURE 3.7(a)
Original Image

FIGURE 3.7(b)
Output after increasing brightness by 15

FIGURE 3.7(c)
Output after increasing contrast 10 basis point on FIGURE 4.1.7(b)

FIGURE 3.7(d)
Output after Globalised Histogram Equalization on FIGURE 4.1.7(c)

FIGURE 3.7(e)
Output after Localized Histogram Equalization on FIGURE 4.1.7(d)

3.8 Program implementing Graphical User Interface.

Discussion about code:

In the code we have made different tabs each having a specific purpose. So now the user just has to press the tab and enter the region on which the operation is to be applied. We have made the Graphical User Interface using GUIDE(GUI Design Environment) module.

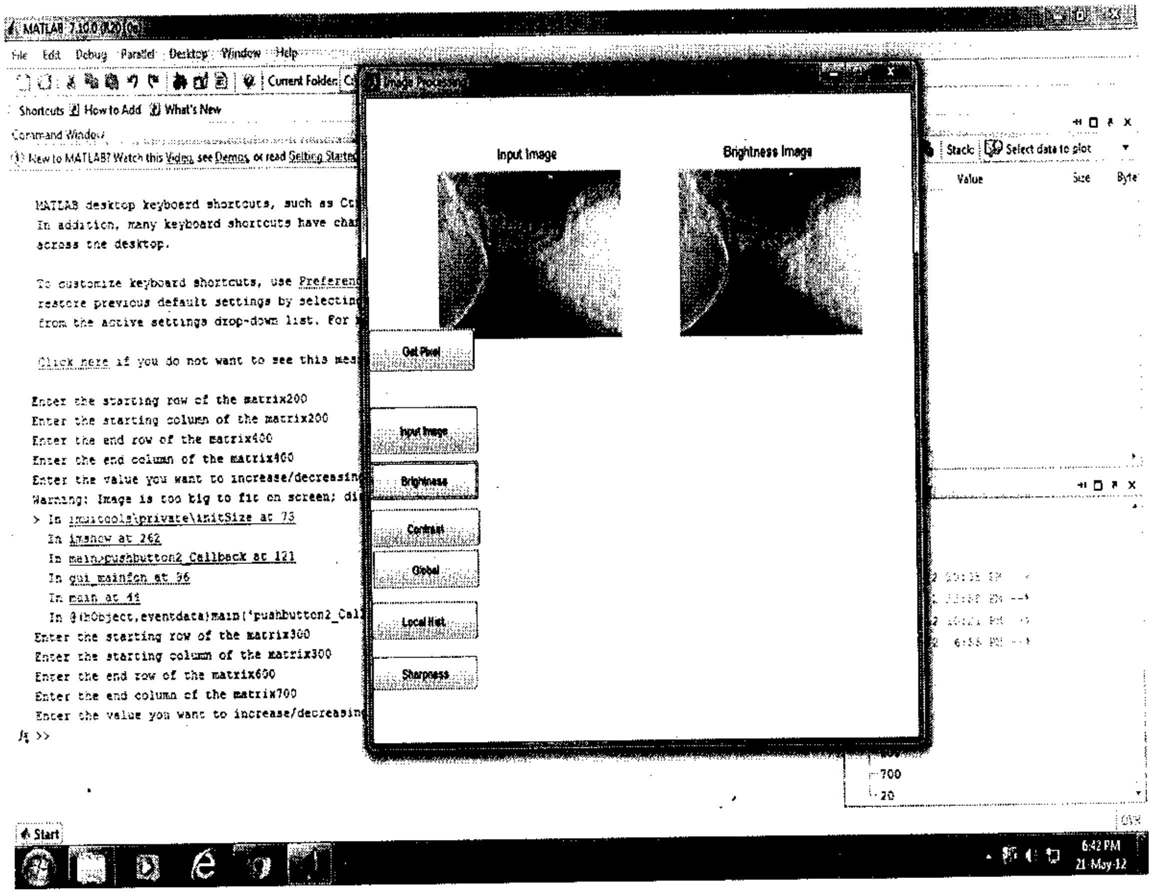


FIGURE 3.8(a)
Increasing the Brightness in the region x:300 to 600 ; y:300 to 700 by 20

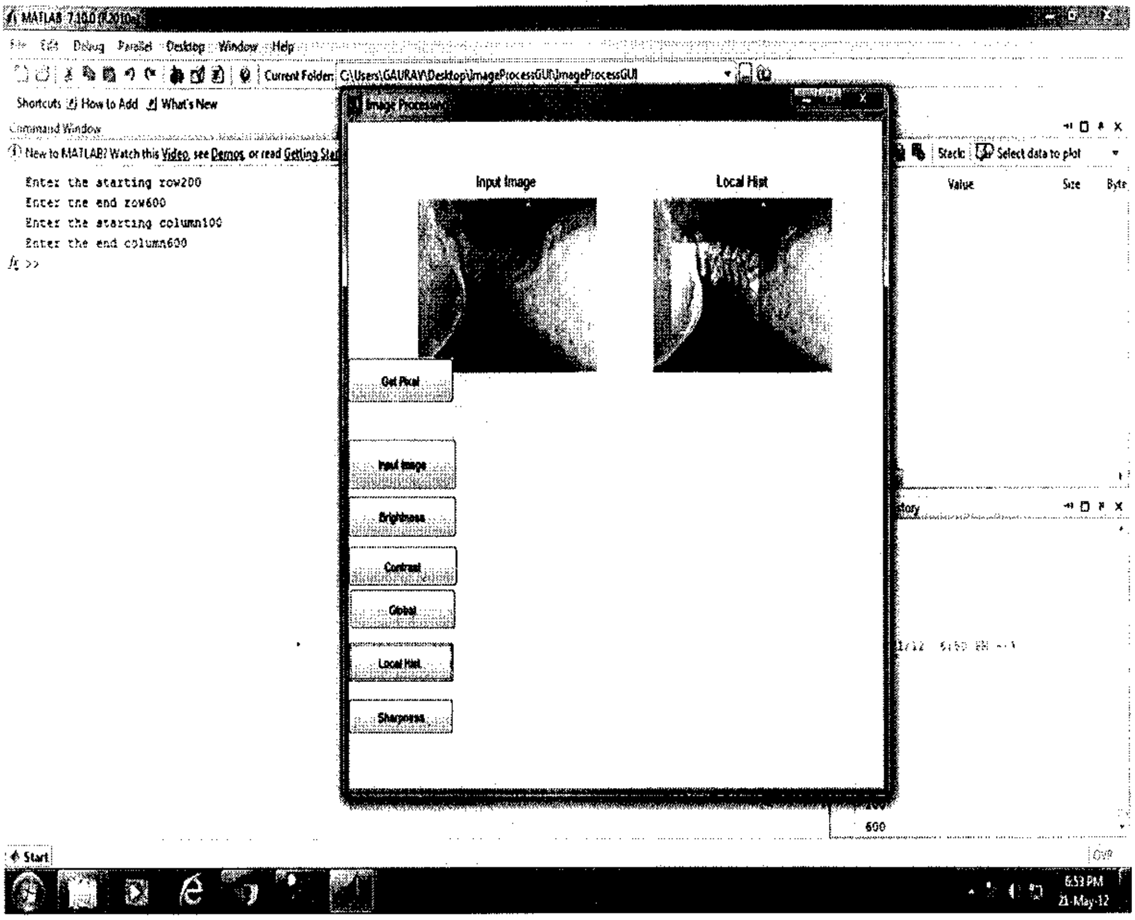


FIGURE 3.8(b)
Local Histogram in the region x:200 to 600 ; y:100 to 600

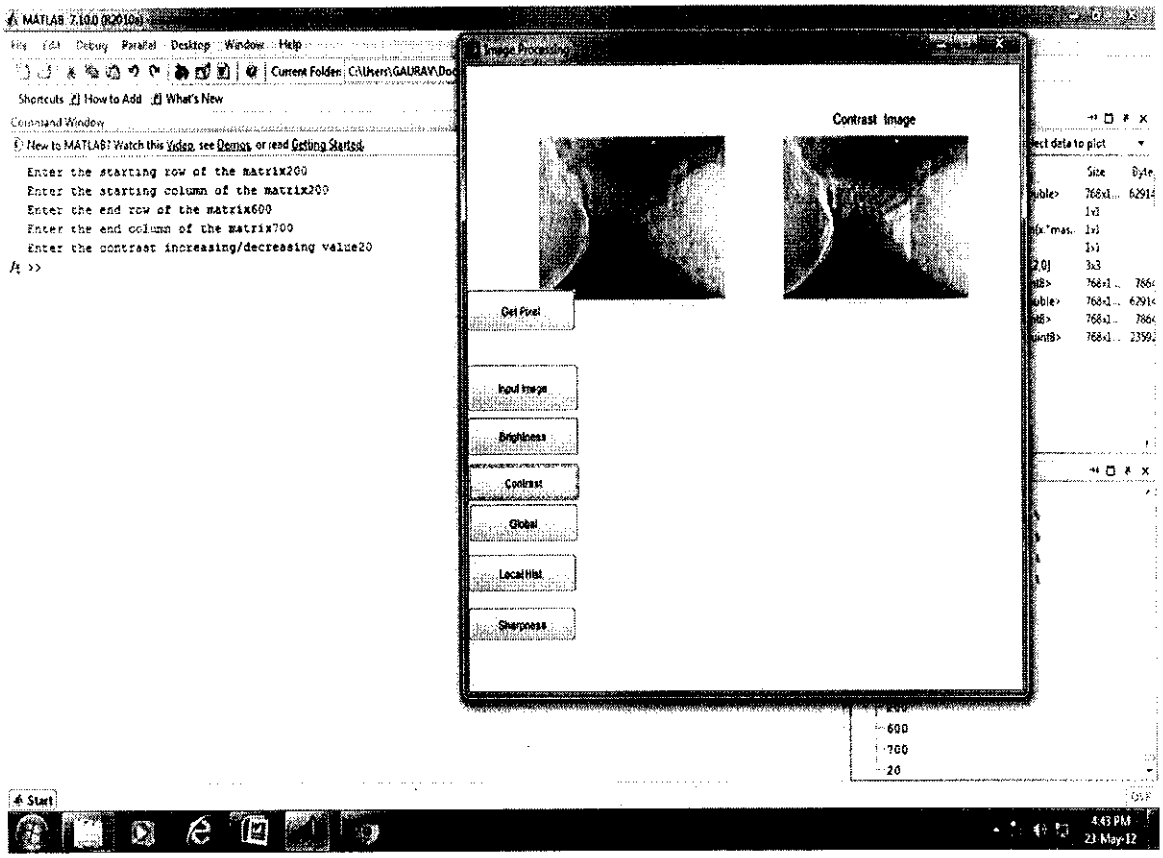


FIGURE 3.8(c)
Increasing the Contrast in the region x:200 to 600 ; y:200 to 700 by 20

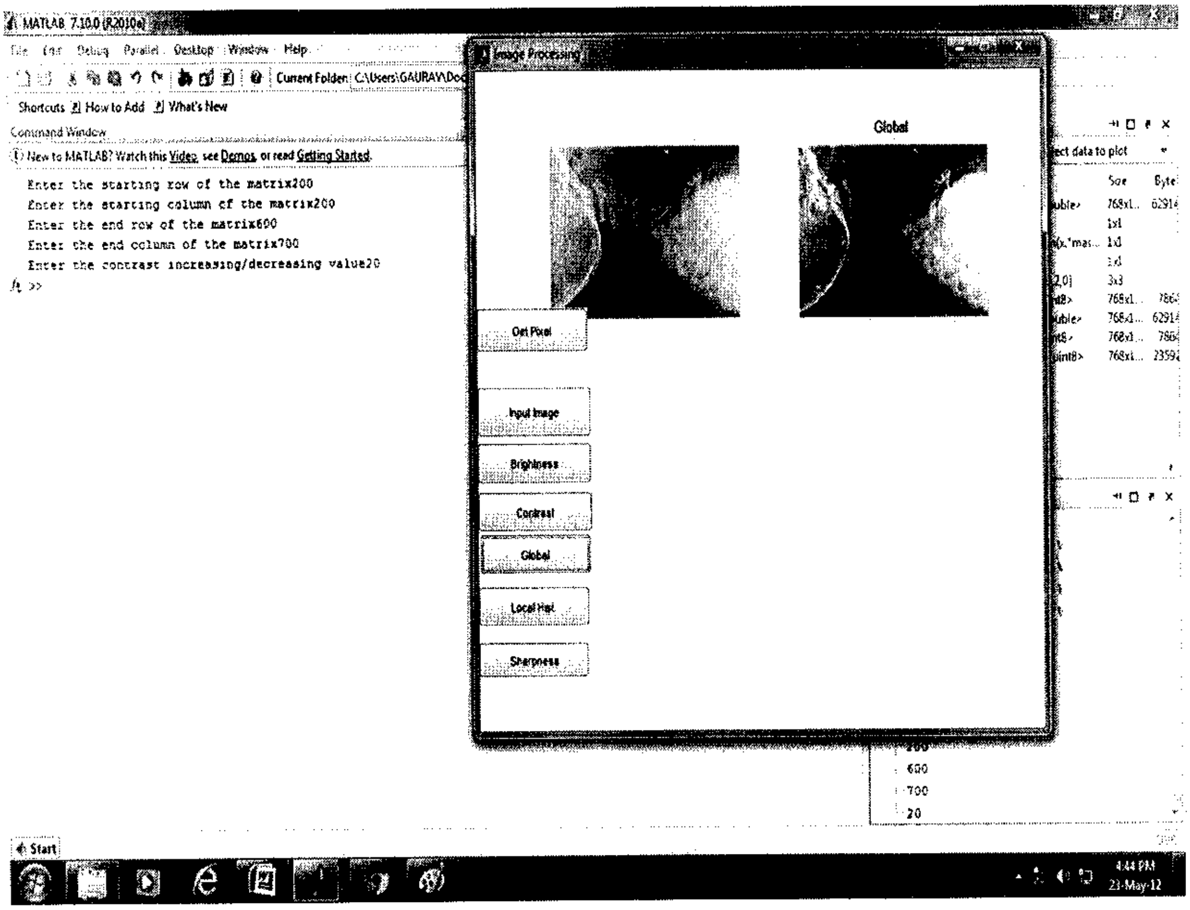


FIGURE 3.8(d)
Globalised Histogram of an Image

APPENDIX

1.

```
 clear all
clc
I=imread('input.bmp');
I=rgb2gray(I);
imshow(I)

a=input('Enter 1.For increasing brightness 2.For increasing contrast');
q=input('Enter the x-coordinate up to which operation is to be applied');
w=input('Enter the y-coordinate up to which operation is to be applied");
if(a==1)
 z=input('Enter the factor by which you want to increase brightness ');
     for i=1:q
     for j=1:w
     I(i,j)=I(i,j)+z;
     if(I(i,j)>255)
       I(i,j)=255;
     else
        I(i,j)=I(i,j);
     end
     end
     end
   elseif(a==2)
     z=input('Enter the factor by which you want to increase contrast');
     for i=1:q
     for j=1:w
     I(i,j)=I(i,j)*(1+(z/100));
     if(I(i,j)>255)
       I(i,j)=255;
     else
        I(i,j)=I(i,j);
     end
     end
     end
   else
     disp('Enter either 1 or 2');
end

figure,imshow(I)
```

2.

```
clear all
clc
i=imread('input.bmp');
x=1:256;
n=size(i);
histo=zeros(1,256);
for u=1: n(1,1)
   for v=1:n(1,2)
      histo(i(u,v)+1)=histo(i(u,v)+1)+1;
   end
end
stem(x,histo);
```

3.

```
clear all
clc
x=1:256
i=imread('input.bmp');
n=size(i);
histo=zeros(1,256);
for u=1:n(1,1)
   for v=1:n(1,2)
      histo(i(u,v)+1)=histo(i(u,v)+1)+1;
   end
end
cumulative=cumsum(histo);
stem(x,cumulative);
```

4.

```
A=imread('input.bmp');
a=double(A);
n=size(a);
n1=n(1,1)*n(1,2);
prob=zeros(1,256);
for i=1:1:n(1,1)
   for j=1:1:n(1,2)
      prob(a(i,j)+1)=prob(a(i,j)+1)+1;
   end
```

```
        end
        for i=1:1:256
           prob(i)=prob(i)/n1;
        end
        cumulative=cumsum(prob);
        for i=1:1:n(1,1)
           for j=1:1:n(1,2)
              final(i,j)=cumulative((a(i,j)+1));
           end
        end
        res=reshape(round(final*255),size(A,1),size(A,2));
        imshow(A)
        figure,imshow(uint8(res))
```

5.

```
clc
clear all
e=imread('input.bmp');
d1=input('Enter the starting row');
d2=input('Enter the end row');
f1=input('Enter the starting column');
f2=input('Enter the end column');

for i=1:1:d1
   for j=1:1:1024
      s(i,j)=I(i,j);
   end
end
for i=(d2+1):1:768
   for j=1:1:1024
      s(i,j)=I(i,j);
   end
end
for i=(d1+1):1:d2
   for j=(f1+1):1:f2
      s(i,j)=0 ;
   end
end
for i=(d1+1):1:d2
   for j=1:1:f1
      s(i,j)= I(i,j);
   end
end
for i=(d1+1):1:d2
```

```
    for j=(f1+1):1:1024
      s(i,j)= I(i,j);
    end
end
prob=zeros(1,256);
for i=(d1+1):1:d2
    for j=(f1+1):1:f2
        prob(I(i,j)+1)=prob(I(i,j)+1)+1;
    end
end

for i=1:1:256
    prob(i)=prob(i)/(768*1024);
end
cumulative=cumsum(prob);
for i=(d1+1):1:d2
    for j=(f1+1):1:f2
        l(i,j)=cumulative((I(i,j)+1));
    end
end

for i=1:1:d1
    for j=1:1:1024
        l(i,j)= 0;
    end
end
for i=(d2+1):1:768
    for j=1:1:1024
        l(i,j)= 0;
    end
end
for i=(d1+1):1:d2
    for j=1:1:f1
        l(i,j)= 0;
    end
end
for i=(d1+1):1:d2
    for j=f2:1:1024
        l(i,j)= 0;
    end
end

for i=1:1:768
    for j=1:1:1024
        m(i,j)= s(i,j)+round(l(i,j)*255);
```

```
  end
end
x=m;
imshow(x)
end
```

6.

```
x=imread('input.bmp');
s=rgb2gray(x);
a=double(s);

mask1=[0 2 0 ; 2 -8 2 ; 0 2 0];
imhpf=@(x) sum(sum(x.*mask1));
res=(blkproc(a,[1 1],[1 1], imhpf));
for i=1:1:768
  for k=1:1:1024
    q(i,k)=s(i,k)-res(i,k);
  end
end

imshow(x,256)
figure,imshow(q,256)
```

7.

```
clear all
clc
I=imread('input.bmp');
I=rgb2gray(I);
imshow(I)
while(1)
a=input('Enter 1.For increasing brightness \n2.For increasing contrast  \n4.For global histogram equalization\n5.For local histogram equalization on a predefined region \n6. For local histogram equalization on a user defined region \n 7.Sharpening of image');

if(a==1)
q=input('Enter the row of the matrix');
w=input('Enter the column of the matrix');
 z=input('Enter the value you want to increase/decreasing brightness ');
    for i=1:q
    for j=1:w
    I(i,j)=I(i,j)+z;
    if(I(i,j)>255)
      I(i,j)=255;
```

```
        else
          I(i,j)=I(i,j);
        end

        end
        end
x=I;
figure,imshow(x)
end

   if(a==2)
q=input('Enter the row of the matrix');
w=input('Enter the column of the matrix');
        z=input('Enter the contrast increasing/decreasing value');
        for i=1:q
        for j=1:w
        I(i,j)=I(i,j)*(1+(z/100));
        if(I(i,j)>255)
          I(i,j)=255;
        else
          I(i,j)=I(i,j);
        end
        end
        end
        x=I;

figure,imshow(x)
   end


if(a==4)

a=double(I);
n=size(a);
n1=n(1,1)*n(1,2);
prob=zeros(1,256);
for i=1:1:n(1,1)
   for j=1:1:n(1,2)
      prob(a(i,j)+1)=prob(a(i,j)+1)+1;
   end
end
for i=1:1:256
   prob(i)=prob(i)/n1;
end
cumulative=cumsum(prob);
```

```
for i=1:1:n(1,1)
   for j=1:1:n(1,2)
      final(i,j)=cumulative((a(i,j)+1));
   end
end
res=reshape(round(final*255),size(I,1),size(I,2));
x=uint8(res);
figure,imshow(x)
end

if(a==5)
d=input('Enter the row');
f=input('Enter the column');

for i=(d+1):1:760
   for j=(f+1):1:1020
      s(i,j)=I(i,j);
   end
end
for i=1:1:d
   for j=1:1:f
      s(i,j)=0 ;
   end
end
for i=(d+1):1:760
   for j=1:1:f
      s(i,j)= I(i,j);
   end
end
for i=1:1:d
   for j=(f+1):1:1020
      s(i,j)= I(i,j);
   end
end
prob=zeros(1,256);
for i=1:1:d
   for j=1:1:f
      prob(I(i,j)+1)=prob(I(i,j)+1)+1;
   end
end

for i=1:1:256
   prob(i)=prob(i)/(d*f);
end
cumulative=cumsum(prob);
```

```
for i=1:1:d
   for j=1:1:f
      l(i,j)=cumulative((l(i,j)+1));
   end
end

for i=(d+1):1:760
   for j=(f+1):1:1020
      l(i,j)= 0;
   end
end
for i=(d+1):1:760
   for j=1:1:f
      l(i,j)= 0;
   end
end
for i=1:1:d
   for j=(f+1):1:1020
      l(i,j)= 0;
   end
end

for i=1:1:760
   for j=1:1:1020
      m(i,j)= s(i,j)+round(l(i,j)*255);
   end
end
x=m;
imshow(x)
end


if(a==6)
   d1=input('Enter the starting row');
d2=input('Enter the end row');
f1=input('Enter the starting column');
f2=input('Enter the end column');

for i=1:1:d1
   for j=1:1:1024
      s(i,j)=l(i,j);
   end
end
for i=(d2+1):1:768
   for j=1:1:1024
```

```
      s(i,j)=I(i,j);
    end
end
for i=(d1+1):1:d2
    for j=(f1+1):1:f2
      s(i,j)=0 ;
    end
end
for i=(d1+1):1:d2
    for j=1:1:f1
      s(i,j)= I(i,j);
    end
end
for i=(d1+1):1:d2
    for j=(f1+1):1:1024
      s(i,j)= I(i,j);
    end
end
prob=zeros(1,256);
for i=(d1+1):1:d2
    for j=(f1+1):1:f2
      prob(I(i,j)+1)=prob(I(i,j)+1)+1;
    end
end

for i=1:1:256
    prob(i)=prob(i)/(768*1024);
end
cumulative=cumsum(prob);
for i=(d1+1):1:d2
    for j=(f1+1):1:f2
      l(i,j)=cumulative((I(i,j)+1));
    end
end

for i=1:1:d1
    for j=1:1:1024
      l(i,j)= 0;
    end
end
for i=(d2+1):1:768
    for j=1:1:1024
      l(i,j)= 0;
    end
end
```

```
for i=(d1+1):1:d2
   for j=1:1:f1
      l(i,j)= 0;
   end
end
for i=(d1+1):1:d2
   for j=f2:1:1024
      l(i,j)= 0;
   end
end

for i=1:1:768
   for j=1:1:1024
      m(i,j)= s(i,j)+round(l(i,j)*255);
   end
end
x=m;
imshow(x)
end

if(a==7)

u=double(I);

mask1=[0 2 0 ; 2 -8 2 ; 0 2 0];
imhpf=@(x) sum(sum(x.*mask1));
res=(blkproc(u,[1 1],[1 1], imhpf));
for i=1:1:768
   for k=1:1:1024
      q(i,k)=s(i,k)-res(i,k);
   end
end

imshow(x,256)
figure,imshow(q,256)
end
g=input('do you want to continue /n 8.press 8for yes /n 9.press 9 for no');
I=x;
if(g==9)
   break;
end
end
```

CONCLUSION

We were successfully able to develop and run a user-friendly program with a Graphical User Interface. The user can enhance the features of an image through various techniques in our program that are embedded into one. And the user can select any region for the operation as per his purpose. Hence the objective of our project is fulfilled.

Everyday technological advancements are being made in this field and this area of image enhancement promises to be a part of various research projects in the future. Technology never stops developing and there is always a future scope of expansion.

REFRENCES

1. Zhengzhou Li, Gang Luo, Eli Pel Schepens "**Factors Affecting Image Quality Preferences**" Eye Research Institute, Department of Ophthalmology, Harvard Medical School.
2. PremNandhini Satgunam, Russell L Woods, P Matthew Bronstad, Eli Peli "**Image enhancement of high digital magnification for patients with central vision loss**" Factors Affecting Image Quality Preferences .
3. Rafael C. Gonzalez, Richard E. Woods and Steven L. Eddins (2004). *Digital Image Processing using MATLAB*. Pearson Education.
4. Fundamental of Digital Image Processing by A.K .Jain
5. Image contrast enhancement based on the generalized histogram by Byoung-Woo Yoon and Woo-Jin Song