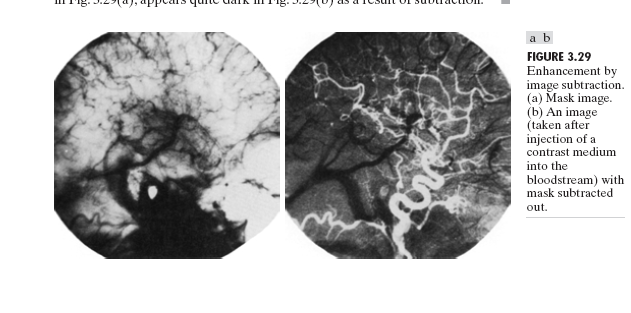
**Practical No 14.B**

**Implementation of histogram equalization - Image Subtraction.**

**Aim: Write a program to implement a histogram equalization using Image Subtraction algorithm.**

**Theory:**

Image subtraction or pixel subtraction is a process whereby the digital numeric value of one pixel or whole image is subtracted from another image. This is primarily done for one of two reasons – levelling uneven sections of an image such as half an image having a shadow on it, or detecting changes between two images. This detection of changes can be used to tell if something in the image moved. This is commonly used in fields such as astrophotography to assist with the computerized search for asteroids or Kuiper belt objects in which the target is moving and would be in one place in one image, and another from an image one hour later and where using this technique would make the fixed stars in the background disappear leaving only the target. One of the most commercially successful and beneficial uses of image subtraction is in the area of medical imaging called mask mode radiography. In this case h (x, y), the mask, is an X-ray image of a region of a patient’s body captured by an intensified TV camera (instead of traditional X-ray film) located opposite an X-ray source. The procedure consists of injecting a contrast medium into the patient’s bloodstream, taking a series of images of the same anatomical region as h (x, y), and subtracting this mask from the series of incoming images after injection of the contrast medium. The net effect of subtracting the mask from each sample in the incoming stream of TV images is that the areas that are different between f (x, y) and h(x, y) appear in the output image as enhanced detail. Because images can be captured at TV rates, this procedure in essence gives a movie showing how the contrast medium propagates through the various arteries in the area being observed.



**Conclusion: We have implemented histogram equalization using Image Subtraction algorithm.**

**Code:**

#include<iostream.h>

#include<fstream.h>

#include<conio.h>

#include<string.h>

#include<math.h>

struct pix

{

unsigned char b,g,r;

}pixel1,pixel2;

int L=255;

char Header[54];

int c;

ifstream in1,in2;

ofstream out,out1;

class sub

{

public:

void process()

{

char infile1[]="cat.bmp";

char infile2[]="cat2.bmp";

char outfile[]="cat-sub.bmp";

char imdata[]="imdata.dat";

in1.open(infile1,ios::in|ios::binary);

in2.open (infile2,ios::in|ios::binary);

in1.read(( char\*)(&Header),sizeof(Header));

in2.read(( char\*)(&Header),sizeof(Header));

out.open(outfile,ios::out| ios::binary);

out.write(( char\*)(&Header),sizeof(Header));

out1.open(imdata, ios::out);

while(!in1.eof()||!in2.eof())

{

in1.read((char \*)(&pixel1),sizeof(pixel1));

in2.read((char \*)(&pixel2),sizeof(pixel2));

out1<<"ORIGINAL : " <<(int)pixel1.r<<" , "<<(int)pixel1.g<<" , "<<(int)pixel1.b<<endl;

pixel1.r=pixel1.r-pixel2.r;

pixel1.g=pixel1.g-pixel2.g;

pixel1.b=pixel1.b-pixel2.b;

out.write((char \*)(&pixel1),sizeof(pixel1));

out1<<"MODIFIFED (subtraction) : " <<(int)pixel1.r<<" , "<<(int)pixel1.g<<" , "<<(int)pixel1.b<<endl;

}

in1.close();

in2.close();

out.close();

}

};

int main()

{

clrscr();

sub s;

s.process();

getch();

}

**Input:**

**Output:**

