**Practical No 12.A**

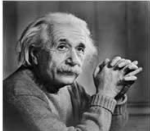
**Implementation of Basic Intensity Transformation - Image negative.**

**Aim: Write a program to implement a Basic Intensity Transformation using Image negative algorithm.**

**Theory:**

The second linear transformation is negative transformation, which is invert of identity transformation. In negative transformation, each value of the input image is subtracted from the L-1 and mapped onto the output image. The result is somewhat like this.

**Input Image**

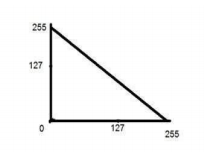


**Output Image**



**In this case the following transition has been done.**

s = (L – 1) – r; since the input image of Einstein is an 8 bpp image, so the number of levels in this image are 256. Putting 256 in the equation, we get this s = 255 – r So each value is subtracted by 255 and the result image has been shown above. So, what happens is that, the lighter pixels become dark and the darker picture becomes light. And it results in image negative.



**Conclusion: We have implemented Basic Intensity Transformation using Image negative algorithm.**

**Code:**

#include<iostream.h>

#include<fstream.h>

#include<conio.h>

#include<string.h>

struct pix {

unsigned char b,g,r;

}

pixel;

int L=255;

char Header[54];

ifstream in;

ofstream out,out1;

void main() {

clrscr();

char infile[]="C:\\TURBOC3\\BIN\\cat.bmp";

char outfile[]="C:\\TURBOC3\\BIN\\output.bmp";

char imdata[]="C:\\TURBOC3\\BIN\\logdata.rtf";

in.open(infile,ios::in|ios::binary);

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

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

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

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

while(!in.eof()) {

in.read((char\*) (&pixel),sizeof(pixel));

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

pixel.r=L-pixel.r;

pixel.g=L-pixel.g;

pixel.b=L-pixel.b;

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

out1<<"MODIFIED (NEGATIVE) : "<<(int)pixel.r<<" "<<(int)pixel.g<<" "<<(int)pixel.b<<endl; }

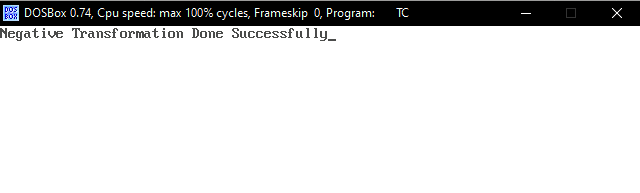
in.close();

cout<<"Negative Transformation Done Successfully";

getch();

}

**Output:**



Before Transformation After Transformation