

Contrast stretching which is a simple image processing technique and can also be called normalization and it attempts to enhance the contrast of a low contrast image making it look much more appealing to the human eye. It does that by stretching the range of the pixel values contained in the image to a desired one. When checking the histogram of a low contrast image we can observe that the distribution is towards the bright side of the histogram that's the brighter pixels but contrast stretching stretches out the distribution giving the image and the histogram even distribution of brightness making the image more appealing to the eye. We shall see some sample images that have been worked on below.

Negative is another image processing technique and it is the inverse of a normal or positive image where light or bright pixels in the image appear dark and dull or dim pixels appear light or bright. Usually inverse or negative images appear light in an enclosed space (dark place).

Code for contrast+negative.

```
#include "Image.h"
#include <math.h>

void contrast_stretch(GrayscaleImage& img, GrayscaleImage& out)
{
    int min = img(0, 0);
    int max = img(0, 0);
    int result;
    for (int y = 0; y < img.GetHeight(); y++)
    {
        for (int x = 0; x < img.GetWidth(); x++)
        {
            if (img(x, y) > max)
                max = img(x, y);
            if (img(x, y) < min)
                min = img(x, y);
        }
    }
    for (int y = 0; y < img.GetHeight(); y++)
    {
        for (int x = 0; x < img.GetWidth(); x++)
        {
            result = round((img(x, y) - min) * (255.0 / (max - min)));
            if (result > 255)
                result = 255;
            if (result < 0)
                result = 0;
            out(x, y) = result;
        }
    }
}

void negative(GrayscaleImage& img, GrayscaleImage& out)
{
    for (int y = 0; y < img.GetHeight(); y++)
    {
        for (int x = 0; x < img.GetWidth(); x++)
        {
            out(x, y) = 255 - img(x, y);
        }
    }
}
```

```

    }
}

int main() {
    GrayscaleImage img;
    img.Load("images/image_4.png");
    GrayscaleImage out(img.GetWidth(), img.GetHeight());
    contrast_stretch(img,out);/*kindly comment out when performing inverse or
negative*/
    negative(img, out);/*kindly comment out when performing contrast
stretching*/
    out.Save("images/output.png");
    return 0;
}

```

Now we shall see some sample images that I've experimented on and we shall see the differences when contrast stretching is used and the difference when negative is used. From my study I believe you still get the same results with negative when you add both contrast+negative together. You shall see some low contrast images that have been worked on.

Image set 1

Original low contrast image:



Using Contrast stretching:



We can see the difference in both images when contrast stretching is used

Using inverse or negative on the image:



We can see that all pixel values in this image have been inverted.

Image set 2

Original low contrast image:



Using contrast stretching:



We can see there is not much difference but if you look closely we can see that after contrast stretching has been applied the difference is clear that the stretched image looks better than the original.

Using inverse or negative:



Again as we can see here the pixel values have been inverted making the lighter pixels dark and the darker pixels light.

More Image samples:

Image set 3.

Original low contrast image:



Using contrast stretching:



Using Inverse or negative:

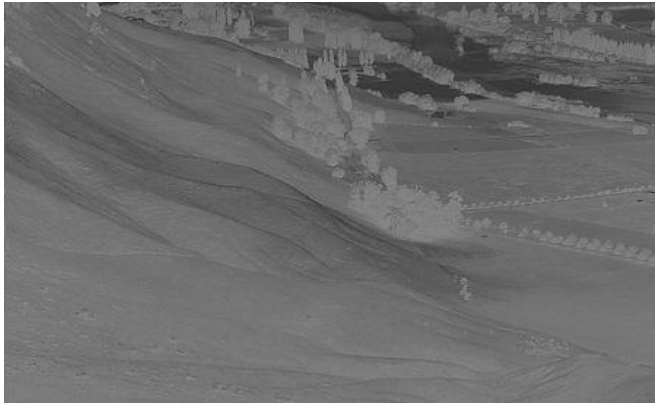


Image set 4

Original low contrast image:



Using Contrast stretching:



Using Inverse or negative:



