Batch: B1	**Roll No: 16010121045**
Experiment No.	

Title: Implement the following point processing techniques in spatial domain:

- Image Negative.
- Thresholding.
- Gray level slicing with and without background
- Bit plane slicing

Objective: To learn & understand point processing techniques.

Expected Outcome of Experiment:

CO	Outcome
CO4	Design & implement algorithms for digital image enhancement, segmentation & restoration.

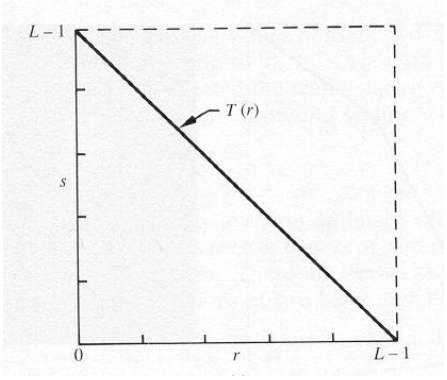
Books/ Journals/ Websites referred:

1. http://www.mathworks.com/support/
2. www.math.mtu.edu/~msgocken/intro/intro.html.
3. R. C.Gonsales R.E.Woods, “Digital Image Processing”, Second edition, Pearson Education
4. S.Jayaraman, S Esakkirajan, T Veerakumar “Digital Image Processing “Mc Graw Hill.
5. S.Sridhar,”Digital Image processing”, oxford university press, 1st edition."

Pre Lab/ Prior Concepts:

Image Negative:

Negative images are useful for enhancing white or grey detail embedded in dark regions of an image. Image negatives are obtained by using the transformation function s=T(r).

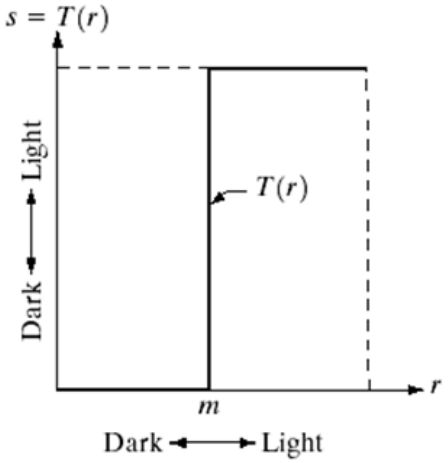


[0,L-1] is the range of gray levels

$S = L\text{-}1\text{-}r$

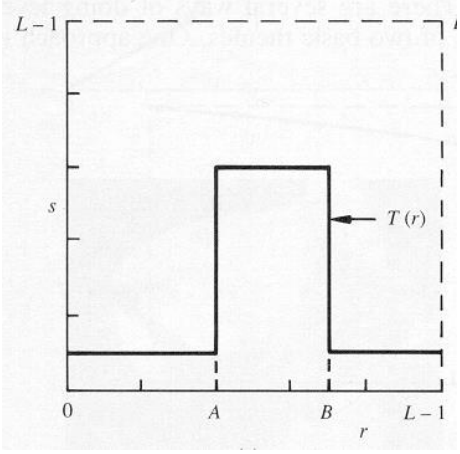
Thresholding

From a grayscale image, thresholding can be used to create binary images. The simplest thresholding methods replace each pixel in an image with a black pixel if the image intensity is less than some fixed constant T or a white pixel if the image intensity is greater than that constant.

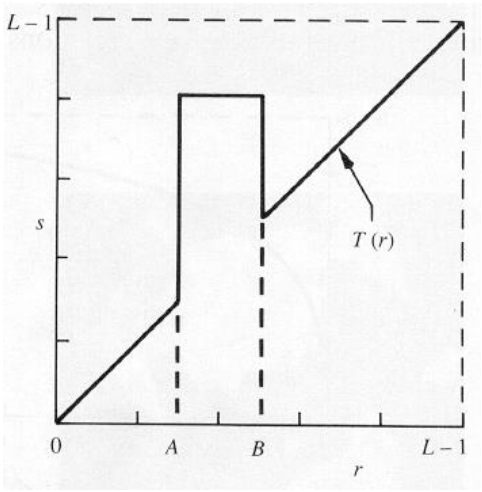


Gray Level Slicing

To highlight a specific range of gray levels in an image (e.g. to enhance certain features). One way is to display a high value for all gray levels in the range of interest and a low value for all other gray levels (binary image).

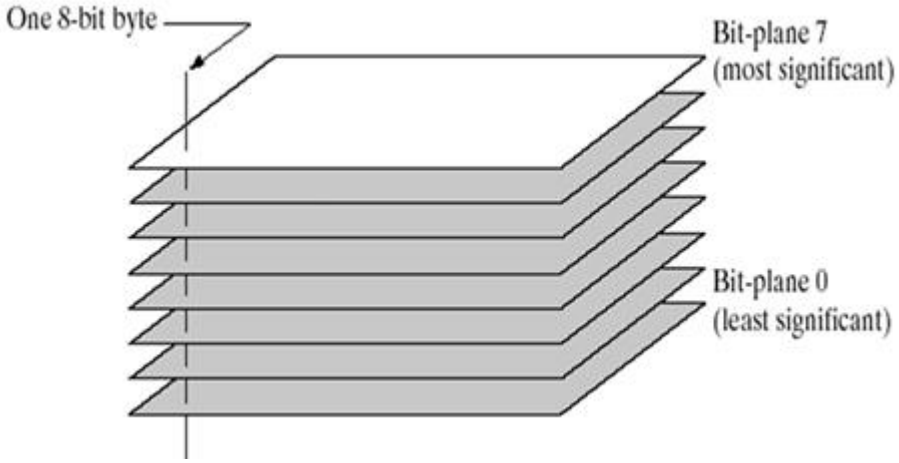


The second approach is to brighten the desired range of gray levels but preserve the background and gray-level tonalities in the image:



Bit plane slicing

Bit plane slicing is used to highlight the contribution made to the total image appearance by specific bits. Assuming that each pixel is represented by 8 bits, the image is composed of 8 1-bit planes. Plane 0 contains the least significant bit and plane 7 contains the most significant bit. Only the higher order bits (top four) contain visually significant data. The other bit planes contribute the more subtle details. Plane 7 corresponds exactly with an image thresholded at gray level 128.

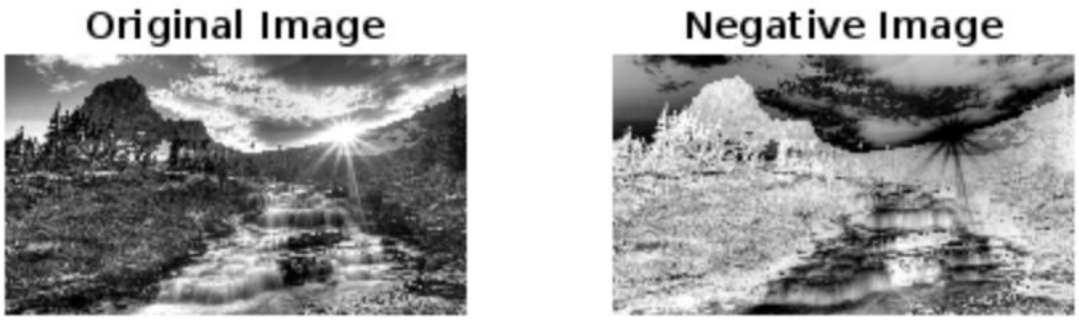


Implementation steps with screenshots:

Image Negetive:

```
originalImage = imread('example.jpg');
if size(originalImage, 3) == 3
   grayImage = rgb2gray(originalImage);
else
   grayImage = originalImage;
end

maxIntensity = max(grayImage(:));

negativeImage = maxIntensity - grayImage;

subplot(1,2,1), imshow(grayImage), title('Original Image');
subplot(1,2,2), imshow(negativeImage), title('Negative Image');

imwrite(negativeImage, 'negative_image.jpg');
```

Output:



Thresholding

```
originalImage = imread('example.jpg');

if size(originalImage, 3) == 3
   grayImage = rgb2gray(originalImage);
else
   grayImage = originalImage;
end

maxIntensity = max(grayImage(:));

threshold = 150;

binaryImage = grayImage > threshold;

subplot(1,2,1), imshow(grayImage), title('Original Image');
subplot(1,2,2), imshow(binaryImage), title('Thresholded Image');

   imwrite(binaryImage, 'thresholded_image.jpg');
```

Output:

Gray Level with, without background

```
originalImage = imread('example.jpg');

if size(originalImage, 3) == 3
   grayImage = rgb2gray(originalImage);
else
   grayImage = originalImage;
end

lowThreshold = 50;
highThreshold = 200;

img = (grayImage >= lowThreshold) & (grayImage <= highThreshold);

backgroundMask = ~(grayImage >= lowThreshold & grayImage <= highThreshold);

enhancedImage = grayImage;
```

```
enhancedImage(backgroundMask) = 255;

subplot(1,3,1), imshow(grayImage), title('Original Image');
subplot(1,3,2), imshow(img), title('Gray Slicing');
subplot(1,3,3), imshow(enhancedImage), title('Gray slicing with BG');

imwrite(enhancedImage, 'enhanced_image_with_background.jpg');
```

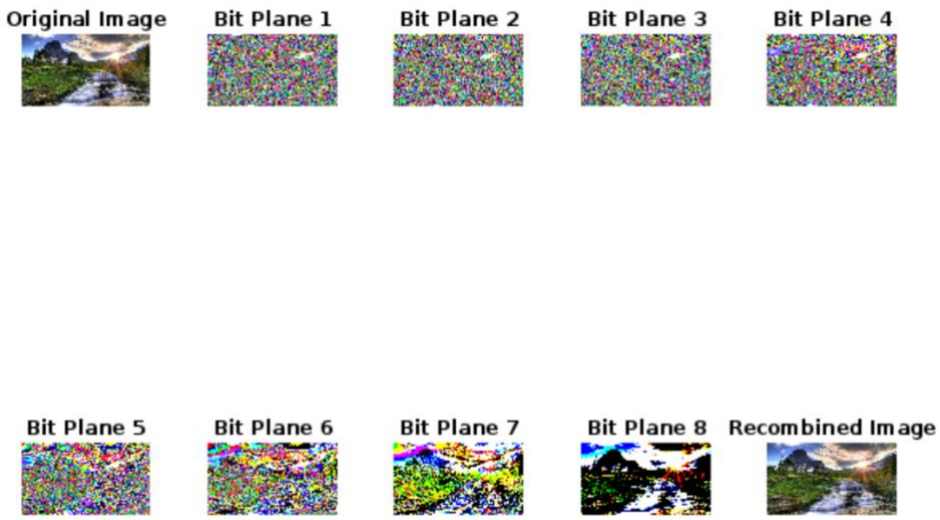
Output:

Bit Plane Slicing:

```
c = imread('example.jpg');

cd = double(c);
c1 = mod(cd, 2);
c2 = mod(floor(cd/2), 2);
c3 = mod(floor(cd/4), 2);
c4 = mod(floor(cd/8), 2);
c5 = mod(floor(cd/16), 2);
c6 = mod(floor(cd/32), 2);
c7 = mod(floor(cd/64), 2);
c8 = mod(floor(cd/128), 2);
cc = (2 * (2 * (2 * (2 * (2 * (2 * (2 * c8 + c7) + c6) + c5) + c4) + c3) + c2) + c1);
subplot(2, 5, 1);
imshow(c);
title('Original Image');
subplot(2, 5, 2);
imshow(c1);
title('Bit Plane 1');
subplot(2, 5, 3);
imshow(c2);
title('Bit Plane 2');
subplot(2, 5, 4);
imshow(c3);
title('Bit Plane 3');
subplot(2, 5, 5);
imshow(c4);
title('Bit Plane 4');
```

```
subplot(2, 5, 6);
imshow(c5);
title('Bit Plane 5');
subplot(2, 5, 7);
imshow(c6);
title('Bit Plane 6');
subplot(2, 5, 8);
imshow(c7);
title('Bit Plane 7');
subplot(2, 5, 9);
imshow(c8);
title('Bit Plane 8');
subplot(2, 5, 10);
imshow(uint8(cc));
title('Recombined Image');
```

Output



Conclusion:-

Successfully completed the given tasks for the experiment and learnt the implementation of the above in matlab.

Post Lab Descriptive Questions

1. **Explain the role of bit plane slicing in achieving Steganography concept.**

Bit plane slicing is a technique used in steganography to hide information within digital images. Steganography is the practice of concealing messages or data within other non-secret data to avoid detection. In the context of images, bit plane slicing involves decomposing the image into its constituent bit planes, where each bit plane represents a specific bit (such as the most significant bit, second most significant bit, and so on) of the pixel values in the image.

The role of bit plane slicing in achieving steganography lies in its ability to manipulate the least significant bits (LSBs) of the pixel values in the image. Since the LSBs contribute the least to the overall visual quality of the image, they can be altered without significantly affecting the perceptual quality of the image. By replacing the LSBs of certain bit planes with the bits of the secret message, one can embed the message into the image.

For example, if you have an 8-bit grayscale image, you can divide it into eight bit planes, each representing one bit of the pixel values (from the most significant bit to the least significant bit). You can then replace the LSBs of selected bit planes with the bits of the secret message. The resulting image will appear visually similar to the original image, but it will contain the hidden message embedded within it. This process allows for the covert transmission of information within seemingly innocuous digital images.

2. Explain the use of gray level slicing

Gray level slicing, also known as intensity slicing, is a technique used in image processing to enhance or highlight specific regions or features in an image based on their intensity levels. In gray level slicing, different intensity levels or ranges are segmented and displayed with varying degrees of brightness or contrast, making it easier to distinguish them from the rest of the image.

The use of gray level slicing is particularly helpful in enhancing the visibility of certain features or objects in an image that may be obscured by noise or other elements. By selectively enhancing the contrast or brightness of specific intensity ranges, important details can be brought to the forefront, aiding in visual analysis or interpretation of the image.