# Department of Computing

**EE433: Digital Image Processing**

**Class: BSCS 6C**

# Lab 5: Bit Plane Slicing

**CLO2: Identify and exploit analogies between the mathematical tools used for 1D and 2D image analysis and processing**

**Time: 10 am to 1 pm**

# Instructor: Dr Khawar Khursheed

# 

# Lab 5: Bit Plane slicing

**Introduction**

This lab is an introduction to negative, gradient of image and bit-plane slicing.

**Objectives**

This lab will provide the concepts of these basic image processing tasks to the students.

**Tools/Software Requirement**

Python 2.7

**Description**

The negative of an image with grey levels in the range [0, L-1] is obtained by the negative transformation shown in figure above, which is given by the expression,

*s = L - 1 - r*

This expression results in reversing of the grey level intensities of the image thereby producing a negative like image. The ouput of this function can be directly mapped into the grey scale look-up table consisting values from 0 to L-1.

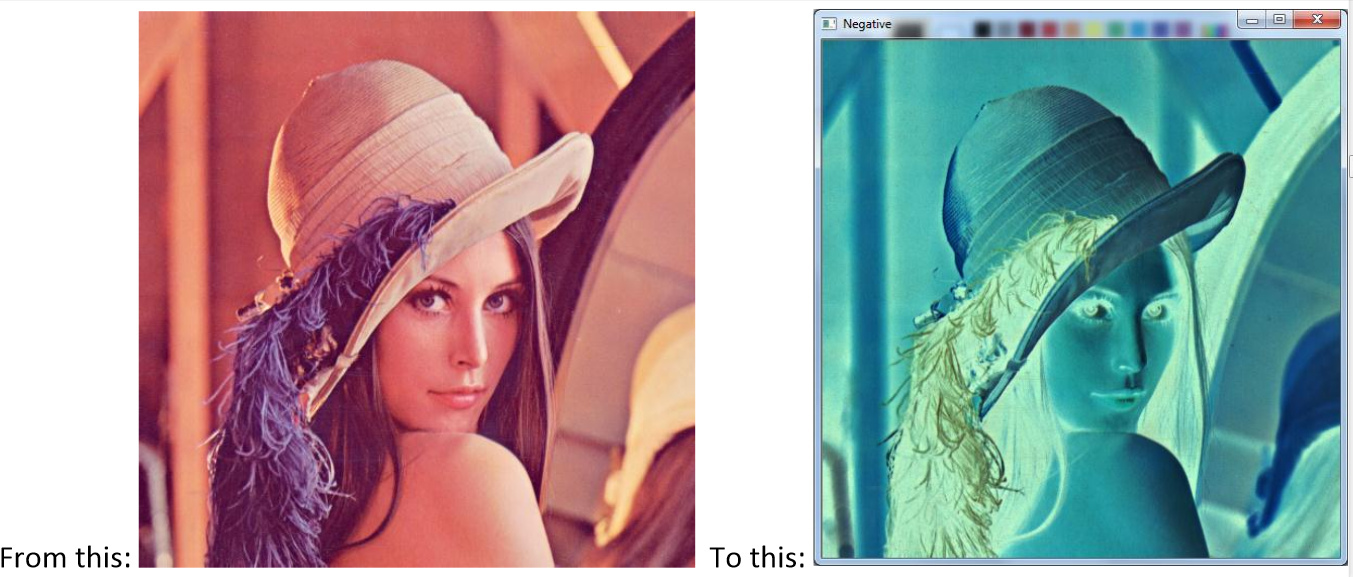
The gradient of an image measures how it is changing. It provides two pieces of information. The magnitude of the gradient tells us how quickly the image is changing, while the direction of the gradient tells us the direction in which the image is changing most rapidly.

Instead of highlighting gray level images, highlighting the contribution made to total image appearance by specific bits might be desired. Suppose that each pixel in an image is represented by 8 bits. Imagine the image is composed of 8, 1-bit planes ranging from bit plane1-0 (LSB)to bit plane 7 (MSB).

In terms of 8-bits bytes, plane 0 contains all lowest order bits in the bytes comprising the pixels in the image and plane 7 contains all high order bits.

**Lab Tasks**

1. Implement a function for displaying negative of an input image. Note that the function must handle binary, grayscale, and RGB images. Example of RGB negative:

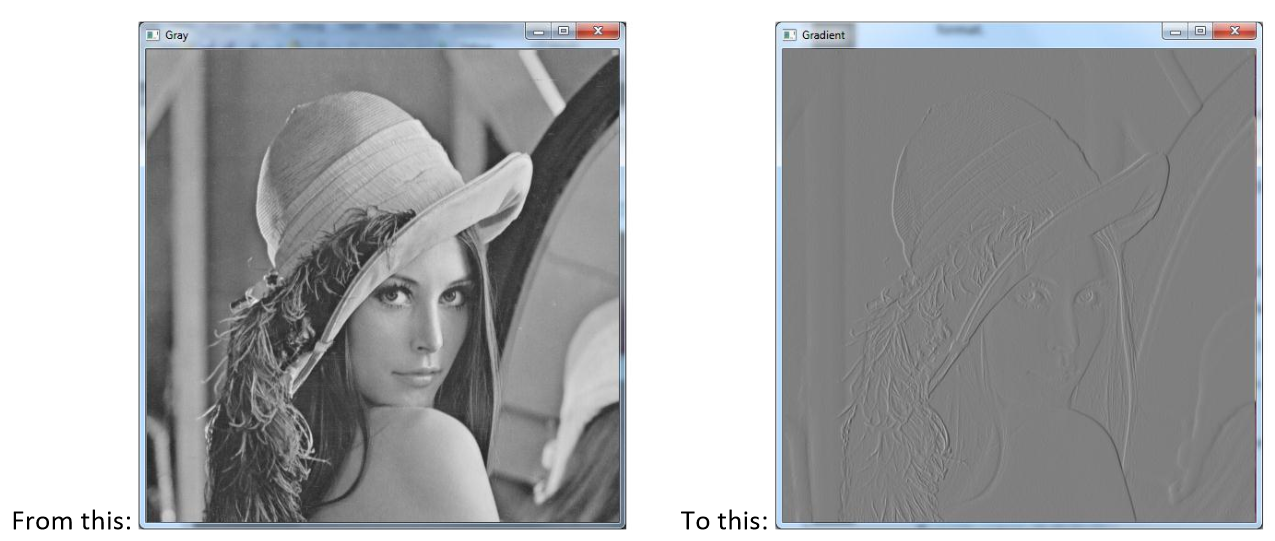


1. The horizontal gradient image can be used to detect vertical edges in an image. How, do you think? Implement a function for displaying the horizontal gradient of a grayscale image. The gradient can be approximated by forward differences:

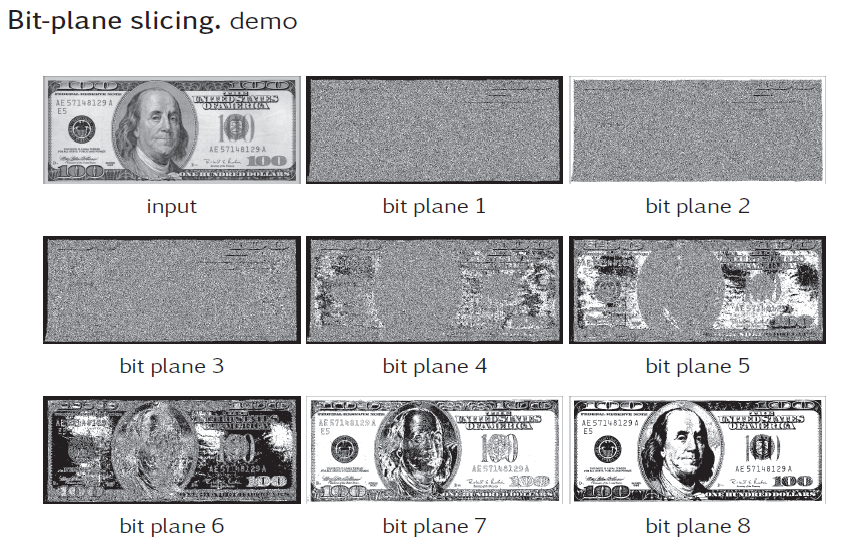
Note that the gradient values can be both positive and negative! So you need to find a way to display the gradient values in the range: 0, 1, 2, …, 255. The following link can be helpful here:

<https://www.cis.rit.edu/people/faculty/rhody/EdgeDetection.htm>

The resulting image should look something like this:



1. Perform bit slicing of an 8 bit greyscale image as discussed in the lecture. Start from the least significant bit and move towards the most significant bit. You will get eight binary images of the input image as demonstrated below.



**Deliverable**

Hand in the source code from this lab at the appropriate location on the blackboard system at LMS.