Mujtaba Shahid Faizi

BSCS-5A

#131818

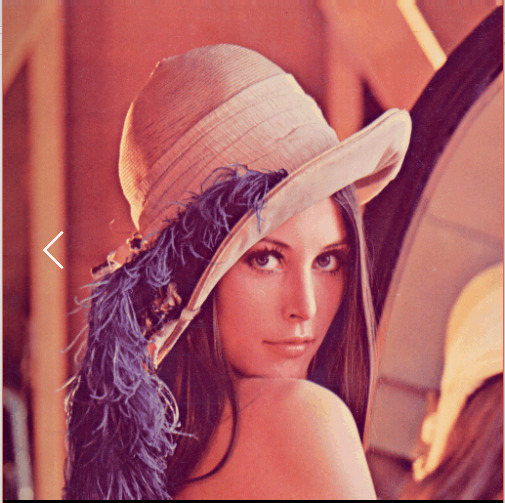
LAB 5 of DIP

**Task 1:**

**Code:**

**from** PIL **import** Image  
**from** scipy.misc **import** imread  
**import** numpy **as** np  
  
**def** negative(im):  
  
 **if** len(im.shape) == 3: *#for rgb images* height = len(im) *# width & height of the image i.e. no. of pixels* width = len(im[0])  
 **for** row **in** range(height):  
 **for** col **in** range(width):  
 red = 255 - im[row][col][0] *#subtracting 255 from pixel intensities* green = 255 - im[row][col][1]  
 blue = 255 - im[row][col][2]  
 im[row][col] = [red, green, blue]  
 im = Image.fromarray(im)  
 **return** im  
  
 **else**: *#for binary and greyscale images* arr=np.asarray(im)  
 height = len(im) *# width & height of the image i.e. no. of pixels* width = len(im[0])  
 **for** x **in** range(0, height):  
 **for** y **in** range(0, width):  
 arr[x,y]=255-arr[x,y] *#subtracting 255 from pixel intensities* im = Image.fromarray(arr)  
 **return** im  
  
rgb = imread(**'Mujtaba.png'**)  
grayscale = imread(**"JIJIJIJJ.png"**)  
binary = imread(**"jiji.jpg"**)  
  
negative(rgb).show()  
negative(grayscale).show()  
negative(binary).show()

**Screenshot:**





**Task 2:**

The horizontal gradient basically means that we are iterating all pixel columns of a particular row to calculate gradient resulting in the vertical edge detection.

**Code:**

**from** PIL **import** Image  
**import** numpy **as** np  
  
img = Image.open(**"Mujtaba.png"**)  
img = img.convert(**'L'**) *# convert image to greyscale*arr=np.asarray(img)  
height = len(arr) *# width & height of the image i.e. no. of pixels*width = len(arr[0])  
pix = np.zeros((height, width), dtype=np.uint8) *# new array of zeros***for** x **in** range(0, height):  
 **for** y **in** range(0, width):  
 **if** y==width-1: *# when reaching the last pixel column* pix[x][y]=arr[x,y]  
 **continue  
 else**:  
 pix[x, y] = abs(arr[x,y] - arr[x, y+1]) *# subtracting pixel intensity with right neighbour pixel intensity*im = Image.fromarray(pix) *# converting array to image*im.show()

**Screenshot:**

