**SVKM’s NMIMS**

**Mukesh Patel School of Technology Management & Engineering**

**Department of Electronics and Telecommunication Engineering**

**Subject: Image and Video Processing Program: B.Tech**

**Sem: VII ACAY: 2020-21**

**EXPERIMENT NO. 3**

**Aim:**

1. To write a program in PYTHON to obtain image negative of an image
2. To write a program in PYTHON to obtain thresholding of an image
3. To write a program in PYTHON to obtain grey level slicing of an image without background
4. To write a program in PYTHON to obtain grey level slicing of an image with background

**Software:** PYTHON

**Prerequisite:**

|  |  |
| --- | --- |
| Sr. No | Concepts |
| 1. | Point processing techniques for image enhancement |

**Outcome:**

After successful completion of this experiment students will be able to comprehend:

Concept of image enhancement in spatial domain using point processing methods

**Theory:**

**Point processing in spatial domain:**

All the processing done on the pixel values. Point processing operations take the form.

s = T ( r ) , T is referred to as a grey level transformation function or a point processing operation, s refers to the processed image pixel value and r refers to the original image pixel value

**Image Negative**

s = (L-1) – r , where L= number of grey levels

**Thresholding**

s=L-1 for r > threshold

s= 0 for r < threshold

**Grey level slicing without background**

s= L-1 for a < r < b, here a and b define some specific range of grey level

s= 0 otherwise.

**Grey level slicing with background**

s= L-1 for a < r < b, here a and b define some specific range of grey level

s= r otherwise.

|  |
| --- |
| Name of the Experiment: To perform point processing techniques on the given image |
| Roll No. C018 Name: Dhruvit Jain |
| Program: B.Tech ExTC Semester : VII |
| Date of Performance:24/07/2020 Date of Submission: 24/07/2020 |

* **Code for scaling:**

from skimage import io

from skimage.color import rgb2gray

import matplotlib.pyplot as plt

image = io.imread ('C:/Users/dhruv/Desktop/College/NOTES/IVP/Labs/lenna\_grey.jpg')

io.imshow(image)

def scale (x,a):

y=x\*a

return y

image\_scale = image.copy()

sh=image\_scale.shape

print (sh)

for row in range (0,sh[0]):

for col in range (0,sh[1]):

image\_scale[row][col]= scale (image[row][col],0.5)

plt.figure()

plt.figure(figsize=(10,10))

plt.subplot(1,2,1)

io.imshow (image)

plt.title ('Original Image')

plt.subplot(1,2,2)

io.imshow (image\_scale)

plt.title ('Scaled Image')

**Outputs for scaling:**

Scaling factor = 0.5



Scaling factor = 2



* **Code for brightness:**

from skimage import io

from skimage.color import rgb2gray

import matplotlib.pyplot as plt

image = io.imread ('C:/Users/dhruv/Desktop/College/NOTES/IVP/Labs/lenna\_grey.jpg')

io.imshow(image)

def scale (x,a):

y=x+a

return y

image\_scale = image.copy()

sh=image\_scale.shape

print (sh)

for row in range (0,sh[0]):

for col in range (0,sh[1]):

image\_scale[row][col]= scale (image[row][col],25)

plt.figure()

plt.figure(figsize=(10,10))

plt.subplot(1,2,1)

io.imshow (image)

plt.title ('Original Image')

plt.subplot(1,2,2)

io.imshow (image\_scale)

plt.title ('Scaled Image')

**Outputs for brightness modification:**

Brightness increased by 25:

****

Brightness increased by 100:



Brightness decreased by 50:



* **Code for thresholding without background:**

#thresholding without background

image\_gray = rgb2gray (image) #so that we dont have to work in 3 different planes

sh= image\_gray.shape #output will be (225,225)

th = int (input("Enter Threshold Value: "))

th

#image\_gray rgb2gray will normalise values between 0 and 1 therefore we multiply by 255

image\_gray = 255\*image\_gray

image\_gray

image\_th = image\_gray.copy()

for row in range (0,sh[0]):

for col in range (0,sh[1]):

if image\_gray[row][col] >= th:

image\_th[row][col] = 255

else:

image\_th [row][col] = 0

plt.figure()

plt.figure(figsize=(10,10))

plt.subplot(1,2,1)

io.imshow (image\_gray, cmap='gray')

plt.title ('Original Image')

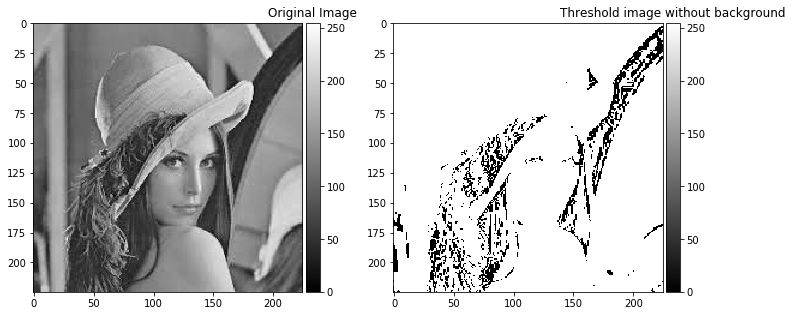
plt.subplot(1,2,2)

io.imshow (image\_th,cmap='gray')

plt.title ('Threshold image without background')

**Outputs for thresholding without background:**

Threshold value = 50:



Threshold value = 150:



Threshold value = 200:



* **Code for thresholding with background:**

#thresholding with background

image\_gray = rgb2gray (image) #so that we don’t have to work in 3 different planes

sh= image\_gray.shape #output will be (225,225)

th = int (input("Enter Threshold Value: "))

th

#image\_gray rgb2gray will normalize values between 0 and 1 therefore we multiply by 255

image\_gray = 255\*image\_gray

image\_gray

image\_th = image\_gray.copy()

for row in range (0,sh[0]):

for col in range (0,sh[1]):

if image\_gray[row][col] >= th:

image\_th[row][col] = 255

else:

pass

plt.figure()

plt.figure(figsize=(10,10))

plt.subplot(1,2,1)

io.imshow (image\_gray, cmap='gray')

plt.title ('Original Image')

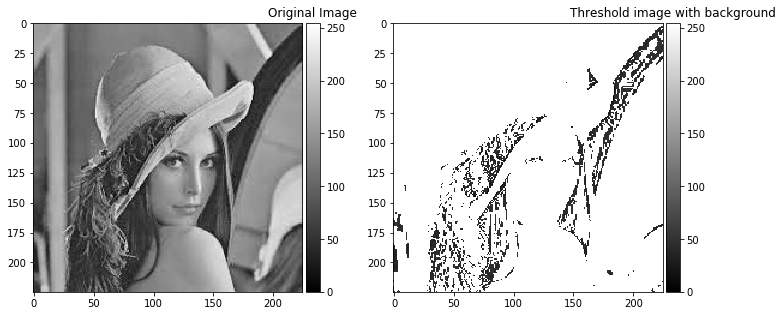
plt.subplot(1,2,2)

io.imshow (image\_th,cmap='gray')

plt.title ('Threshold image without background')

**Outputs:**

Threshold value = 50:



Threshold value = 150:



**Conclusions:**

1. To change the contrast of an image, all the pixels are multiplied by a constant value. If the constant is greater than 1, the image is transformed to a whiter image. If constant is less than 1, then the image is transformed to a darker image.
2. Overall brightness of an image is increased by adding a positive constant and can be reduced by adding a negative constant.
3. For both contrast and brightness modification, intensity value of each pixel is limited between 0 to 255.
4. A specific part of an image can be highlighted by using intensity level thresholding. The part to be highlighted should be converted to white, remaining portion is either converted to black or kept unchanged.
5. Therefore, intensity level thresholding can be used for extraction of the object with and without background.