Online practice lab assignment (III)

Course Code : CSE - 406

Course Name : Digital image processing Laboratory

Name: Abdul Mukit

Roll : 2407

Exam Roll: 170491

Session : 2018 - 19

Submitted to

Prof. Dr. Liton Jude Rozario

Department of Computer Science and Engineering

Jahangirnagar University.

Exp:01 Read and display digital images with python, scilab and matlab

Python code

```
import skimage.io as io
import matplotlib.pyplot as plt

gray_img = io.imread ('Cameraman.tiff')
color_img = io.imread ('color.tiff')
indexed_img = io.imread ('indexed.png')
plt.subplot(1,3,1)
plt.imshow (gray_img, cmap = "gray")
plt.subplot (1,3,2)
plt.imshow (color_img)
plt.subplot (1,3,3)
plt.imshow (indexed_img)
plt.show()
```

Matlab code

```
gray_img = imread ('cameraman.tiff');
color_img = imread ('color.tiff');
indexed_img = imread ('indexed.png');

subplot (1,3,1);
imshow (gray_img);
subplot (1,3,2)
imshow (color_img)
subplot (1,3,3)
imshow (indexed_img)
```

Exp :02 Visualize image histogram and perform histogram equalization with python

⇒

```
import matplotlib.pyplot as plt
import skimage.io as io
import cv2
import skimage.exposure as ex
img01 = io.imread('cameraman.tif')
eq_img = cv2.equalizeHist(img01)
plt.subplot(1,3,1)
plt.imshow(img01, cmap='gray')
plt.subplot(1,3,2)
plt.imshow(eq_img, cmap='gray')
plt.subplot(1,3,3)
plt.hist(img01.flatten(), bins=256)
plt.show()
```

```
# RGB image
img2 = io.imread('colon.tiff')

r_channel = img2[:,:,0]
g_channel = img2[:,:,1]
b_channel = img2[:,:,2]

plt.subplot(1,5,1)
plt.imshow(img2)
plt.subplot(1,5,2)
plt.hist(r_channel.flatten, bins=256)
plt.subplot(1,5,3)
plt.hist(g_channel.flatten(), bins=256)
plt.subplot(1,5,4)
plt.hist(b_channel.flatten(), bins=256)
Ihsv = colon.rgb2hsv(img2)
V = ex.equalize_hist(Ihsv[:,:,2])
Ihsv[:,:,2] = V
eq_img2 = colon.hsv2rgb(Ihsv)
plt.subplot(1,5,5)
plt.hist(eq_img2.flatten(), bins=256)
plt.show()
```

Exp: 03 : Filtering an image with python (Gaussian, Laplacian)

=>

```
import numpy as np
import skimage.io as io
import scipy.ndimage as ndi
import matplotlib.pyplot as plt
img01 = io.imread ('cameraman.tif')
img02 = ndi.gaussian_filter (img01, 1, truncate=1)
laplacian_filter = np.array([[0,1,0], [1,-4,1],
                             [0,1,0]])
img03 = ndi.convolve (img01, laplacian_filter
                      mode='constant')
plt.subplot(1,3,1)
plt.imshow (img01, cmap='gray')
plt.subplot (1,3,2)
plt.imshow (img02)
plt.subplot (1,3,3)
plt.imshow (img03)
plt.show()
```

Exp:4 Transforming an image with python

① FFT ② DWT

```
import skimage.io as io
from skimage.color import rgb2gray
import numpy as np
import matplotlib.pyplot as plt
import pywt
img = io.imread ('cameraman.tif')
dft = np.fft.fft2(img)
dft_shift = np.fft.fftshift (dft)
dft_result = 20 * np.log (np.abs(dft_shift))

plt.subplot (1,2,1)
plt.imshow(img)
plt.subplot (1,2,2)
plt.imshow (dft_result)
plt.show()
```

```
# DWT
Coeffs = pywt.dwt2(img, 'haar')
CA, (CH, CV, CD) = coeffs
plt.subplot(2,2,1)
plt.imshow(img)
plt.subplot(2,2,2)
plt.imshow(CA)
plt.subplot(2,2,3)
plt.imshow(CH)
plt.subplot(2,2,4)
plt.imshow(CV)
plt.show()
```

Exp: 5 Performing segmentation on image with python (i) Thresholding (ii) edge detection

=>

```
import skimage.io as io
import numpy as np
import matplotlib.pyplot as plt
from skimage import filters
img1 = io.imread ('cameraman.tif')
r,c = img.shape
x,y = np.mgrid[0:r,0:c].astype('float')
P2 = 255.0 - img1 + y/2
plt.subplot(1,3,1)
plt.imshow((img1>40) & (img1<30), cmap='gray
plt.subplot(1,3,2)
plt.imshow(img1>40, cmap='gray')
plt.subplot(1,3,3)
plt.imshow(P2)
plt.show()
```

```
#edge detection
edge_p = filters.prewitt(img1)
edge_r = filters.roberts(img1)
edge_s = filters.sobel(img1)
plt.subplot(1,3,1)
plt.imshow(edge_p, cmap='gray')
plt.subplot(1,3,2)
plt.imshow(edge_r, cmap='gray')
plt.subplot(1,3,3)
plt.imshow(edge_s, cmap='gray')
plt.show()
```

Exp:6 Restoring images from noises with python. (i) salt & pepper noise
(ii) periodic noise

```
=> import skimage.io as io
import numpy as np
import matplotlib.pyplot as plt
import skimage.util as util
import scipy.ndimage as ndi
from numpy.fft import fftshift, fft2, ifft2
                      ifftshift
```

```
inport skimage.exposure as ex
sp = io.imread ('img1.tiff')
gp = io.imread ('img2.tiff')
img_med = ndi.median_filter (sp, 3)
gf = fftshift (fft2 (sp))
mg_spectrum = np.abs (gf)
temp = ex.rescale_intensity (abs(gf), out_range
                             = (0,1))
gf2 = util.img_as_ubyte (temp)
i,j = np.where (gf2 == gf2.max())
z = np.sqrt((x-512) ** (y-512)**2)

k = 1
d = np.sqrt(5832)
bn = (7<np.floor (d-k)) | (z>np.ceil (d+k))

gfn = gf* bn
filtered_gp = np.real (ifft2 (ifftshift (gfn)))
plt.subplot (1,2,1)
plt.imshow (img_med, cmap = 'gray')
plt.subplot (1,2,2)
plt.imshow (filtered_gp, cmap = 'gray')
plt.show ()
```

Exp.07 perform morphological processing of images (dilation & erosion)

⇒

```
import skimage.io as io
import numpy as np
from skimage.filters import threshold_otsu
from skimage.color import rgb2gray
from scipy.ndimage import binary_erosion,
                          binary_dilation

bw = io.imread('text.png')
se1 = np.ones((4,4), dtype = bool)
se2 = np.array([[0,0,0,0,0]
                [0,0,0,0,0]
                [0,0,1,0,0]
                [0,0,0,0,0]
                 [0,0,0,0,0]])
bw1 = binary_dilation(bw, structure = se1)
bw2 = binary_erosion(bw, structure = se2)
plt.subplot(1,2,1)
plt.imshow(bw1)
plt.subplot(1,2,2), plt.imshow(bw2)
plt.show()
```

Exp:08 processing color images with python

(i) converting RGB to HSV / HSI
(ii) Separate RGB channels

⇒

```
import numpy as np
import matplotlib.pyplot as plt
from skimage import io, color
img = io.imread('color.tiff')
hsv_img = color.rgb2hsv(img)
lab_img = color.rgb2lab(img)
r_channel = img[:,:,0]
g_channel = img[:,:,1]
b_channel = img[:,:,2]
plt.subplot(2,3,1)
plt.imshow(hsv_img)
plt.subplot(2,3,2)
plt.imshow(hsl_img)
plt.subplot(2,3,3)
plt.imshow(lab_img)
plt.subplot(2,3,4)
plt.imshow(r_channel)
```

```
plt.subplot(2,3,5)
plt.imshow(g_channel)
plt.subplot(2,3,6)
plt.imshow(b_channel)
plt.show()
```

Exp:10 misclence ~~pop~~ topic in image

processing
① special effect
② video processing

①
```
import numpy as np
import skimage.io as io
import skimage.color as color
import matplotlib.pyplot as plt
f= io.imread('iris1.png')
fg = color.rgb2gray(f)
rows, cols = fg.shape
y,x = np.mgrid[0:cols, 0:rows]
x2 = clip(x+2*1.32, 0, rows-1)
```

```
ripple2 = np.reshape(fg[x2.ravel(), y2.ravel()]
                     (rows, cols)).T]

ripple3 = np.reshape(fg[x2.ravel(), y2.ravel()]
                     (rows, cols)).T)

r, theta = cart2polar(x, y)
r2 = r + np.mod(r, 30)
ripple4 = polar2im(fg, r2, theta)
plt.subplot(2,2,1)
plt.imshow(ripple1)
plt.subplot(2,2,2)
plt.imshow(ripple2)
plt.subplot(2,2,3)
plt.imshow(ripple3)
plt.subplot(2,2,4)
plt.imshow(ripple4)
plt.show()
```

(11)
```
import numpy as np
import cv2 as cv
import scipy.ndimage as ndi
cap = cv.VideoCapture('cat.mp4')
angle =0
cap.set (cv.~~cat~~CAP_PROP_FRAME_WIDTH ,256)
cap.set (cv.CAP_PROR_FRAME_HEIGHT ,144)
if not cap.isOpened():
    print ("cant open video file")
    exit()
while True:
    ret, frame = cap.read()
    if not ret:
        print("cant reciere frame. Exiting...)
        break
    cv2.imshow (original Video, frame)
    gray = cv.cvtColor (frame, cv.COLOR_BGR
                        2GRAY)
    cv.imshow ('Grayscale video, gray)
```

```
angle = (angle+1)%360
output = ndi.rotate(frame, angle, reshape
                                = false)
cv.imshow('Rotate FRAME', output)
  if cv.waitkey(1) == 27:
      break
cap.release()
cv.destroyAllWindows()
```