2019

Digital Image Processing

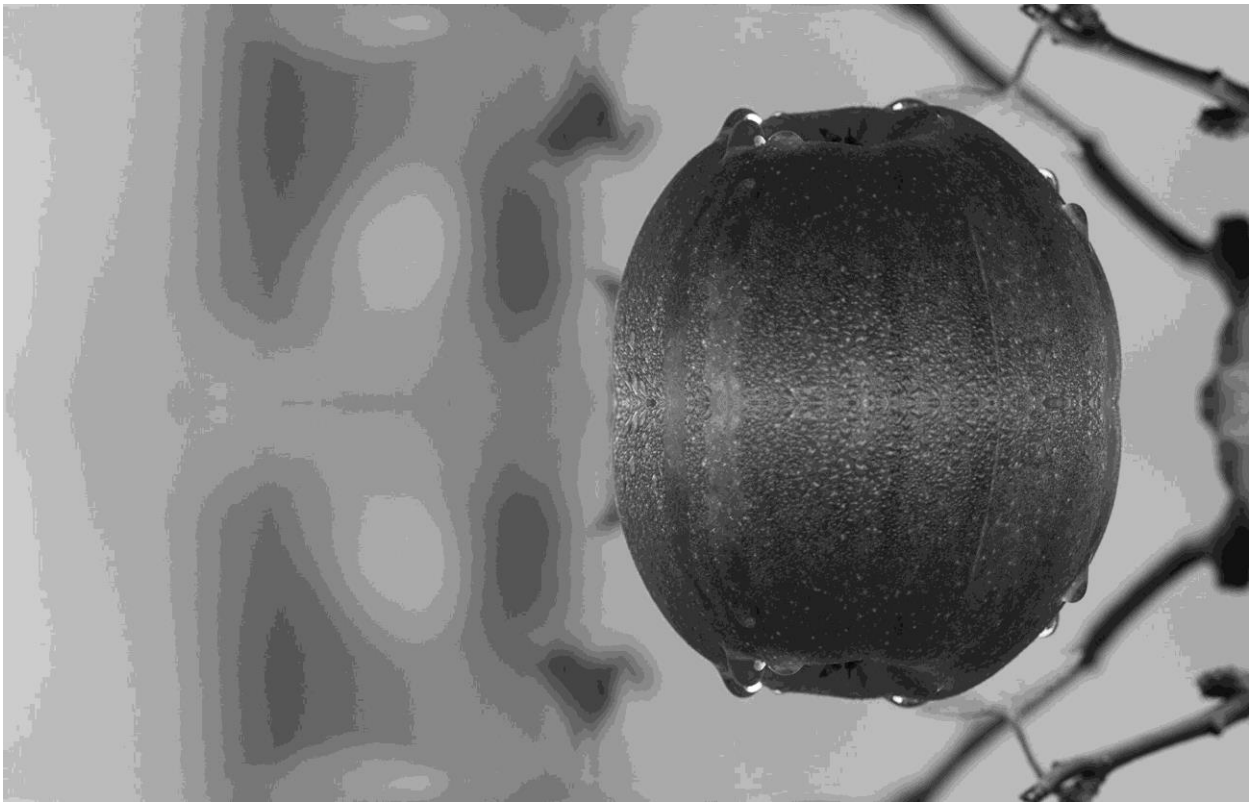
CE-38-B
MISHAAL SAFEER
SAMEEN ABBAS
FOHA KHALID
HASHIR SHOAIB

Tasks:

1: Read any image that you want and save it in gray scale. Now mirror the image that you have read at center i.e. the upper half of the image should be the copy of the lower half but mirrored. Write the image to the disk.

```
import numpy as np
import cv2 as cv

my_img = cv.imread("gray.jpeg", 0)
size = np.shape(my_img)
rows = size[0]
col = size[1]

ind = rows//2

my_img[:ind-1] = np.flipud(my_img[ind+1:])

cv.imwrite("invImage.jpg", my_img)
```

ORIGINAL



MIRRORED

2: Now create a generic code that create a border around any landscape image as shown below. The length of right and left borders must be 10% of the original horizontal length of the image. The length of upper and lower border must be such that the image now have same number of rows as columns. Save the image.

```
border = int(col*(10/100))
newcol = 2*border+col
upperBorder = (newcol-rows)//2

my_img1 = cv.imread("gray.jpeg", 0)

app_img = np.zeros((newcol, newcol), dtype=np.uint8)

app_img[upperBorder:upperBorder+rows, border:newcol-border] = my_img1

cv.imwrite("borderImage.jpg", app_img)
```

using rows and col from prev code

3: Using the following formula f (i, j) = sin (2 π f (i + j)) where i are j indices of a pixel, draw an image with different frequencies (input from user).

```
f = 0.002
p = 1204
gen_img = np.zeros((p, p), np.uint8)

for i in range(0, p):
    for j in range(0, p):
        gen_img[i][j] = (np.sin(2*np.pi*f*(i+j)))*127.5

cv.imwrite("genImage.jpg", gen_img)
```