

Digital Image Processing

Title: Introduction to python for Image Processing

Objectives: Python for image processing

Tools Used: python IDLE/Jupyter

Procedure: Open idle and perform the following tasks

Task 1:

Load image and display. Convert to gray scale and display. Write the grayscale image to another directory with different extension.

Code:

```
import os
import cv2

oldpath = r'C:\Users\Naeem\Desktop\Jahanzeb\DIP\DIP Lab\4\1.JPG'
newpath = r'E:\New folder'

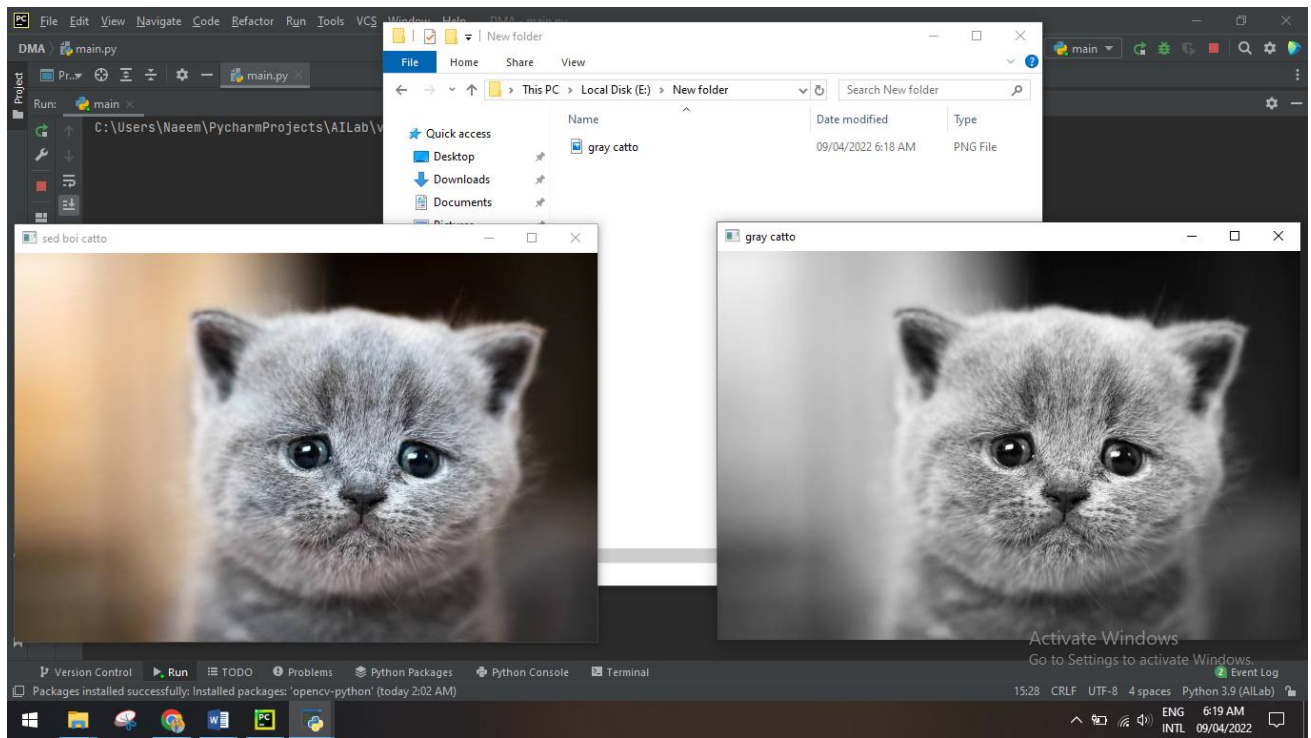
img = cv2.imread(oldpath, cv2.IMREAD_COLOR)
img2 = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)

cv2.imshow('sed boi catto', img)
cv2.imshow('gray catto', img2)

os.chdir(newpath)
cv2.imwrite('gray catto.png', img2)

cv2.waitKey(0)
```

Screenshot:



Task 2:

Binarize any image through your own logic, (without using the given library function). Loop through the pixel values as shown in lab. Ask the value of threshold in binarization from user.

Code:

```
import os
import cv2

path = r'C:\Users\Naeem\Desktop\Jahanzeb\DIP\DIP Lab\4\1.JPG'

img = cv2.imread(path, cv2.IMREAD_COLOR)
img2 = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)

size = img2.shape

res = img2

T = int(input('Input threshold: '))

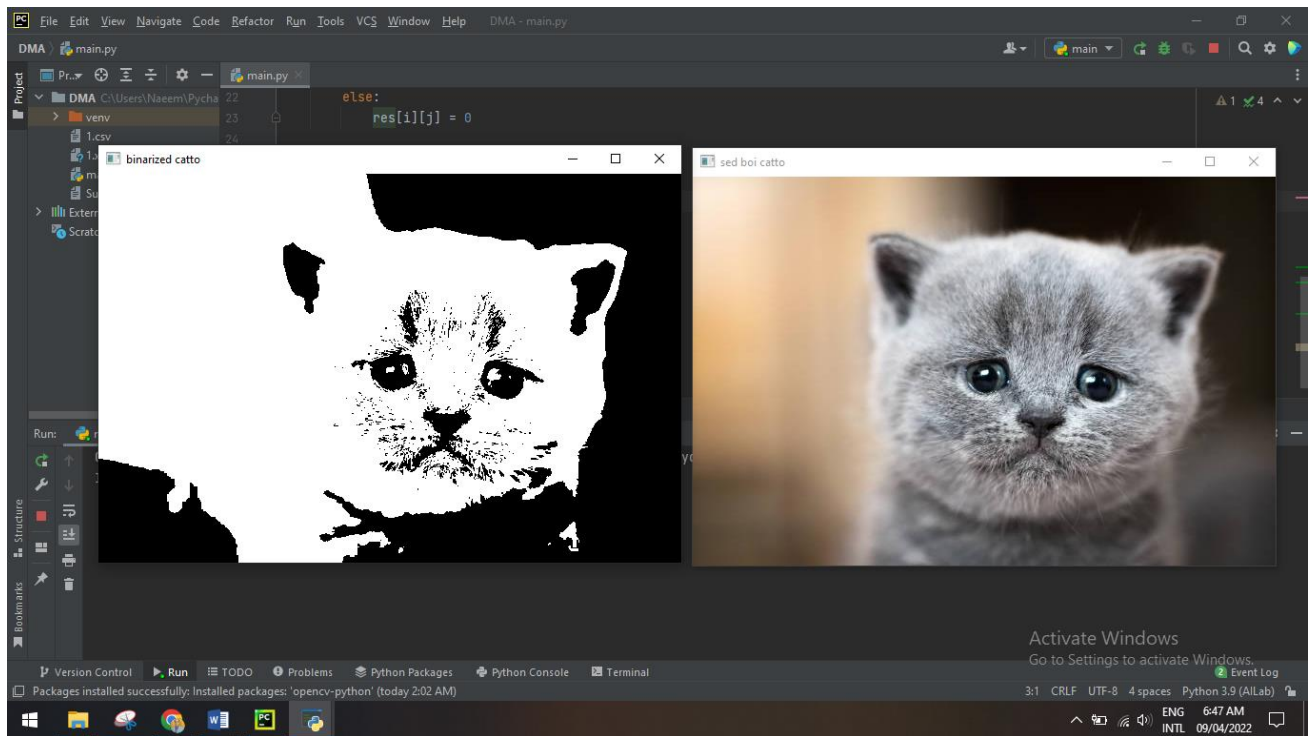
for i in range(size[0]):
    for j in range(size[1]):
        value = int(img2[i][j])

        if value >= T:
            res[i][j] = 255
        else:
            res[i][j] = 0

cv2.imshow('sed boi catto', img)
cv2.imshow('binarized catto', res)

cv2.waitKey(0)
```

Screenshot:



Task 3:

Gray scale any color image using the average method, write code from scratch. (Without using the given library function).

Code:

```
import os
import cv2

path = r'C:\Users\Naeem\Desktop\Jahanzeb\DIP\DIP Lab\4\1.JPG'

img = cv2.imread(path, cv2.IMREAD_COLOR)
size = img.shape

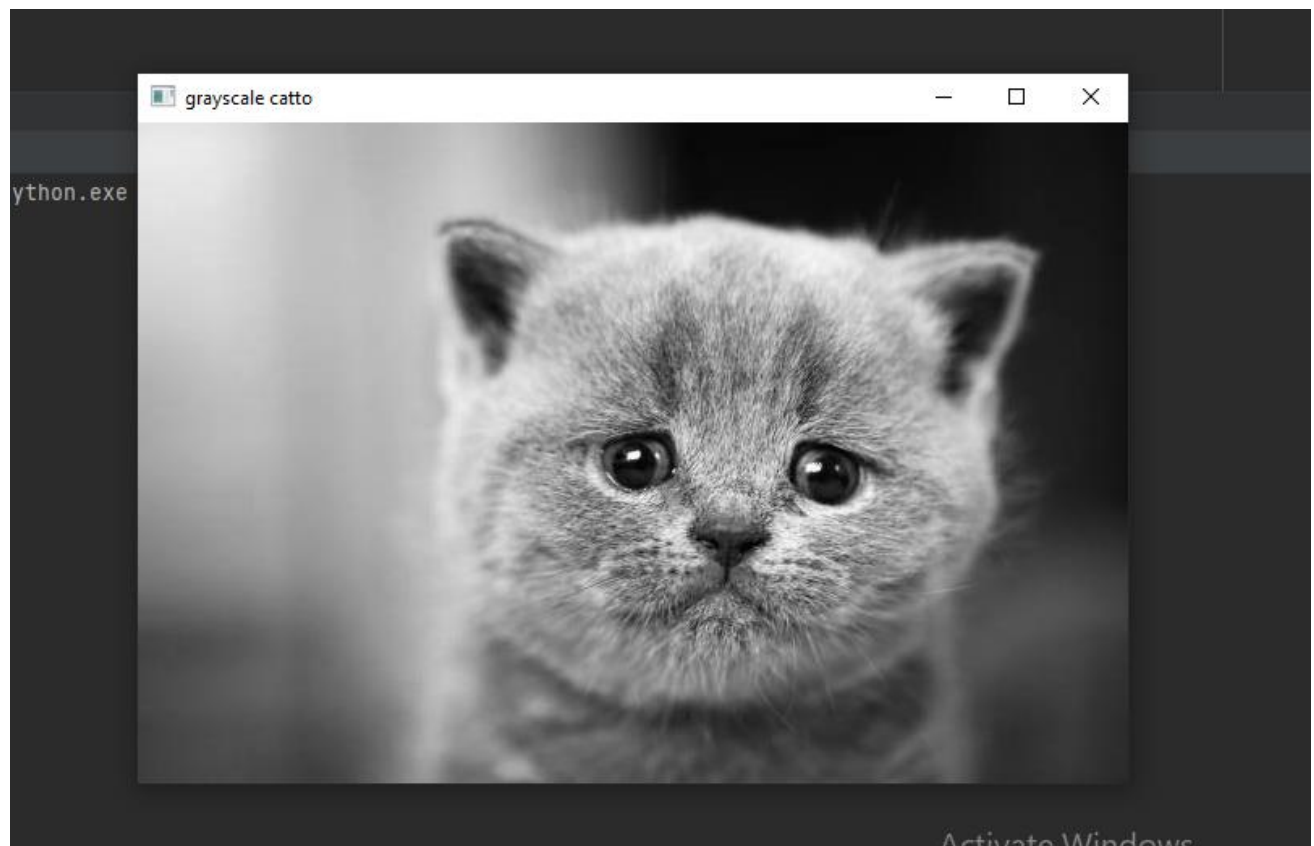
res = img

for i in range(size[0]):
    for j in range(size[1]):
        value = sum(img[i][j])
        res[i][j] = value / 3

cv2.imshow('grayscale catto', res)

cv2.waitKey(0)
```

Screenshot:



Task 4:

Open webcam and perform gray scaling and binarizing on the video frames, you can use opencv functions for gray scale and binarization.

Code:

(for grayscaling)

```
import cv2

cap = cv2.VideoCapture(1)

if not cap.isOpened():
    raise IOError("Cannot open webcam")

while True:
    ret, frame = cap.read()
    cv2.imshow('original', frame)

    gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
    cv2.imshow('gray', gray)

    c = cv2.waitKey(1)
    if c == 27:
        break
```

(for binarizing)

```
import cv2

cap = cv2.VideoCapture(1)

if not cap.isOpened():
    raise IOError("Cannot open webcam")

while True:
    ret, frame = cap.read()
    cv2.imshow('original', frame)

    gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
    ret2, bw = cv2.threshold(gray, 100, 255, cv2.THRESH_BINARY)
    cv2.imshow('gray', bw)

    c = cv2.waitKey(1)
    if c == 27:
        break
```