Python Useful Commands for Project 1

CMPE 685 Teaching Assistant: Nilesh Pandey np9207@rit.edu

*****Important*****
In Project 1, some functions are meant to be your own implementation like Convolution, Median filter, and Gaussian Filter, so accordingly provide your own code without using libraries.

If you are using resources for your code, be sure to reference these resources.

Useful Commands and Code Examples

Use may use opencv or scikit for this project. Here are some useful commands:

Edge filters example

(SKimage) Edge operators

```
from skimage.filters import sobel #loads sobel filter
Import matplotlib.pyplot as plt  # load matplot similar to matlab plotting
library

Image = plt.imread(“path to the image”)
edge_sobel = sobel(image)
fig = plt.figure(dpi=70,figsize=(18,19)) # Decides size and dots per inch
fig.add_subplot(111) # (V,H,N) 6 images, can be plotted as 3,2,1-6
# 3 is rows, 2 is column, 1-6 number of the image currently in the place.
plt.imshow(edge_sobel) # when using gray image, mention ,cmap=’gray’
plt.title(“Sobel filter”)
plt.show()
fig.savefig(“path to save the result”) #save the image
```

(OpenCV) Edge detection

```
import cv2
import numpy as np
from matplotlib import pyplot as plt

img = cv2.imread('path to the file')

sobelx = cv2.Sobel(img,cv2.CV_64F,1,0,ksize=5) #sobel on X gradient refer
the attached link
sobely = cv2.Sobel(img,cv2.CV_64F,0,1,ksize=5) #sobel on Y gradient refer
the attached link
final = cv2.hconcat((sobelx, sobely)) # horizontal concatenation

cv2.namedWindow('image', cv2.WINDOW_NORMAL) # window object named image
cv2.imshow('image',final) # image is displayed inside window object
cv2.imwrite(“Path to save the file”,final) # save the image
cv2.waitKey(0)
cv2.destroyAllWindows() # destroys the window
```

FFT and FFTSHIFT

(SKimage) scikits.image.transform.fft and scikits.image.transform.fftshift

(Numpy) numpy.fft.fft2 and numpy.fft.fftshift

Gaussian Noise and Salt & Pepper Noise

(SKimage) Random Noise

```
import matplotlib.pyplot as plt
from skimage.util import random_noise

Image = plt.imread("path to the image")
Salt_pepper_noise = random_noise(Image,'s&p',mean,variance)
```

Server
The server doesn't have the same display as local machines have. The code should be modified accordingly for local machine and server respectively.

Example code for the sobel filter on the server.

```
import warnings
warnings.filterwarnings("ignore")

import matplotlib
import matplotlib.pyplot as plt
matplotlib.use('Agg')
plt.switch_backend('agg')
from skimage.filters import sobel #loads sobel filter
# load matplot similar to matlab plotting library

Image = plt.imread("path to the image")
edge_sobel = sobel(image)

plt.ioff()
# Turns the plt interactive off, since there is no way to display plt
interactive on server.

fig = plt.figure(dpi=70,figsize=(18,19))
# Decides size and dots per inch
fig.add_subplot(111) # (V,H,N) 6 images, can be plotted as 3,2,1-6
# 3 is rows, 2 is column, 1-6 number of the image currently in the place.
plt.imshow(edge_sobel) # when using gray image, mention ,cmap='gray'
plt.title("Sobel filter")
plt.show()
fig.savefig("path to save the result") #save the image
```

--