İbrahim Talha ASAN
COE-64170019

Assignment 4 Report

Problem 1:

Definition:

The purpose of Problem 1 is to create a class called ImageFilt that can handle digital picture in 2 different ways (blurred and sharpened). In this class, there will be the height, width and image array fields of the image and their constructive method.Additionally, the load_im () method allows the image to be loaded. The filter_im () method uses the blue filter if f = 0 and the sharpening filter for f = 1.

Finally, the plot_im () method displays the filtered images in a single window.

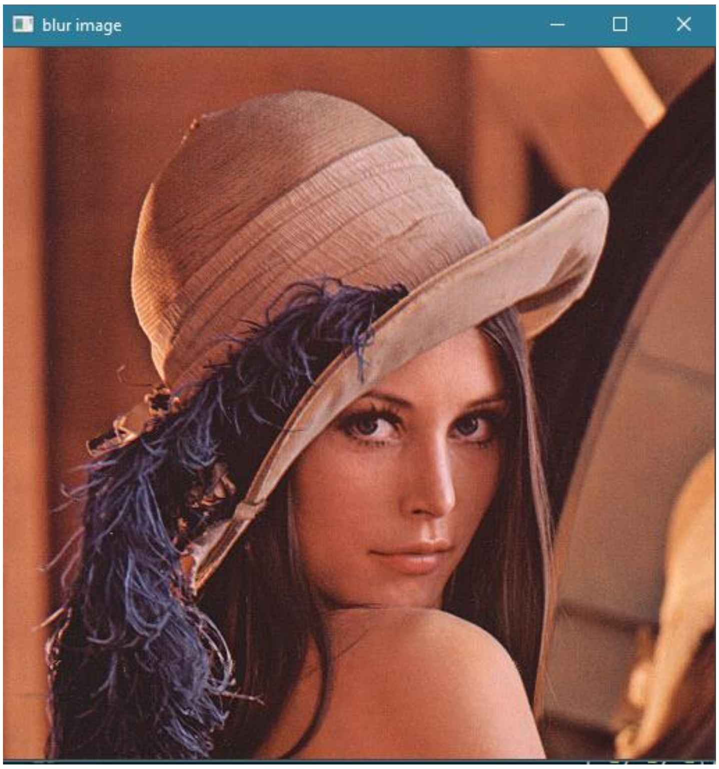
Program Code:

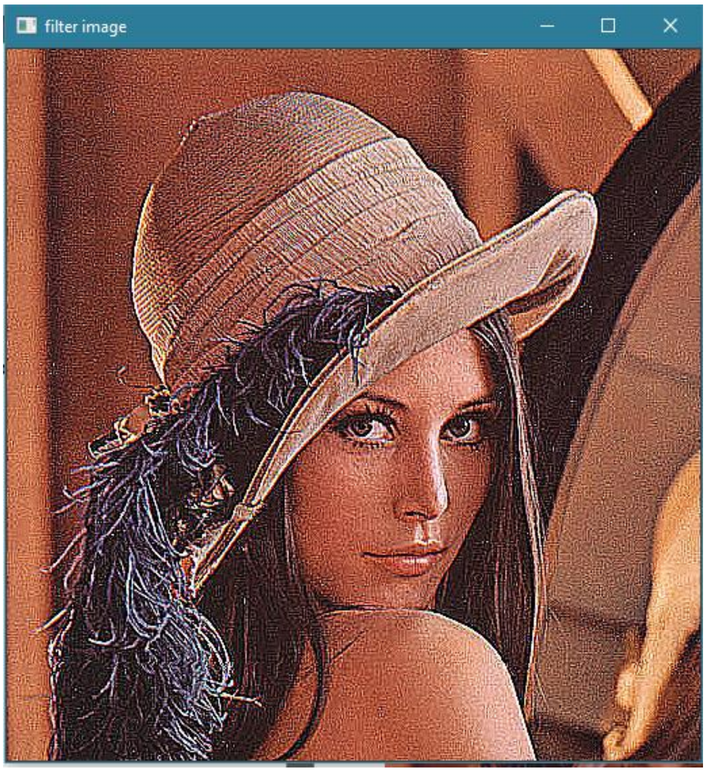
```
import numpy as np
import cv2

class ImageFilt:
   def _init_(self):
      self.image_array = []
      self.width = 0
      self.height = 0

def load_im(self):
      self.image = cv2.imread('C:\\Users\\pc\\.spyder-py3\\openCV\\lena.png')

      if self.image.all == None:
          print("not finding image ")
          return False
      else:
          self.height, self.width, _ = self.image.shape
          self.image_array = np.asarray(self.image)

          return True
   def filter_im(self, f):
       if f == 0:
           self.kernel = np.ones((3, 3), np.float32) / 9

       elif f == 1:
           self.kernel = np.array([[-1,-1,-1],
                                  [-1, 9,-1],
                                  [-1,-1,-1]])

       self.filtered_image = cv2.filter2D(self.image, -1, self.kernel)

   def plot_im(self):

      print(cv2.imshow("blur image",self.image), cv2.imshow("filter image",self.filtered_image))
```

İbrahim Talha ASAN
COE-64170019

```
image = ImageFilt()
image.load_im()

image.filter_im(0)
image.plot_im()

image.filter_im(1)
image.plot_im()
```

Program Outputs:





Discussions:

I couldn't get the image file searched in folders and I solved this problem by doing the necessary operations on only one image file. There is no other problem in solving the question.

İbrahim Talha ASAN
COE-64170019

Problem 2:

Definition:

The purpose of Problem 2 in part a is to evaluate the t value 0 <= t <= 10 (with a size of 0.001 steps) according to the given function f (t). The purpose of option b is to draw graphs of f (t) and t. The purpose of c is to run the f (t) function if the t values are between 0 and 4, and the -f (t) function for the other values.

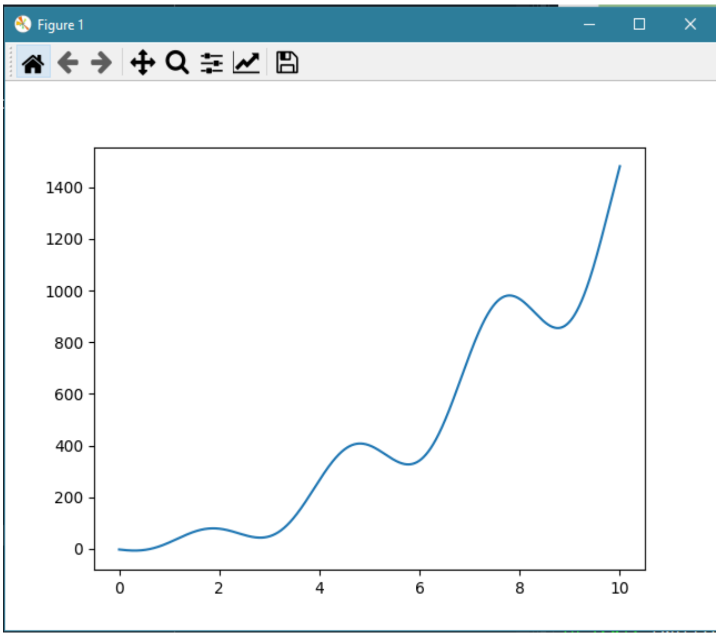
Finally, in the option of d, f (t) is used in the same graph with the original function using the median filter.

Program Code:

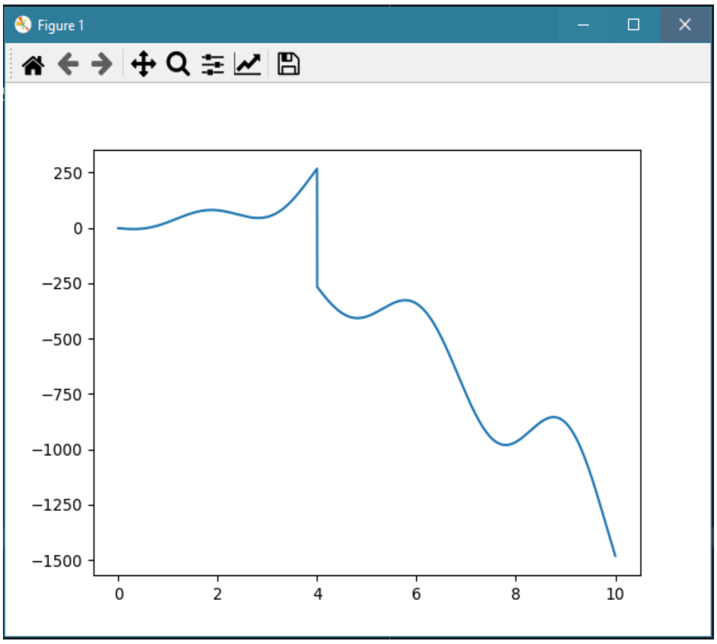
```
import matplotlib.pyplot as plt
from math import e, cos, pi
from PIL import Image, ImageFilter

ft = []
tVal = []
gt=[]

def f(t):
   return (3*(e *-3*t)*cos(2*pi*t/3))-(5*t*(e*-t))-(2*(e **-t))

def g(t):

   if 0 <= t <= 4:
      return f(t)
   else:
      return -f(t)

t = 0
while t <= 10:
   tVal.append(t)
   ft.append(f(t))
   gt.append(g(t))
   t += 0.001

plt.plot(tVal, ft)
plt.show()

plt.plot(tVal, gt)
plt.show()

im1=Image.open('C:\\Users\\pc\\.spyder-py3\\openCV\\f(t).png')
im2=im1.filter(ImageFilter.MedianFilter(size=3))
im2.show()
```

İbrahim Talha ASAN
COE-64170019

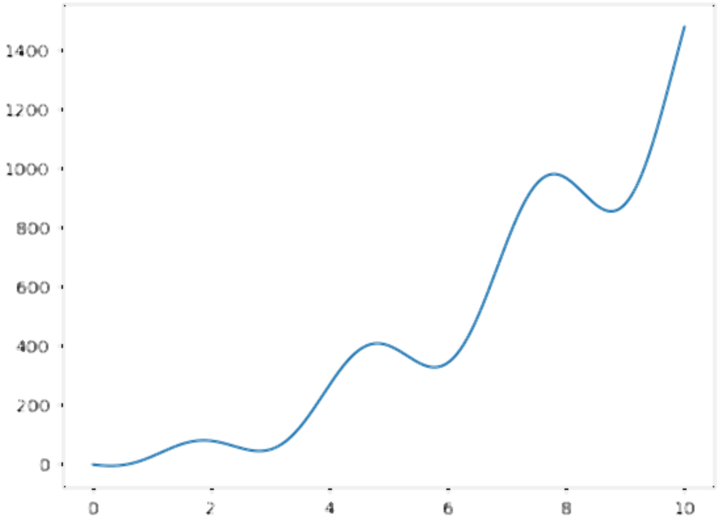
Program Outputs:



(b)



(c)



(d)

Discussions:

In the d option of this problem, I could not combine the graph of the function f (t) with the graph of the filtered function f (t), I added the graph results separately to the report. There is no other problem in solving the question.

İbrahim Talha ASAN
COE-64170019

Problem 3:

Definition:

The purpose of Problem 3 is to perform various analyzes of the given TimeSeries.xlsx data in the form of numpy array. In option a in the problem, the first 150 samples from numdat are extracted to a new vector named numdat_1. In the b option, numdat for 150 samples according to the rule To extract the data in numdat_2 into a new vector called numdat_2. To apply linear interpolation in case of C. In addition, to compare the missing data in numdat_2 with the original time-series.In option d, in numdat_2, 4. and 5. To be able to apply Polynomial Regression to polynomial degrees using polyfit, polyval functions and compare with original time-series.

Finally, in the case of e, it is necessary to plot the original time-series in numdat_1 and compare and evaluate the results of 2 different techniques used in options c and d.

Program Code:

```
import numpy as np
import pandas as pd
from scipy import interpolate
import matplotlib.pyplot as plt

numdat = pd.read_excel('C:\\Users\\pc\\.spyder-py3\\openCV\\TimeSeries.xlsx')

numdat_1 = numdat[0:149]
num1=np.array(numdat_1)

numdat_2=[]*149
for k in range(0,len(num1)-99):
   value=3*k+1
   print(value)
   numdat_2.append(num1[value])

plt.plot(num1,numdat_2, marker="o", ls="")
sx=np.log10(num1)
xi_ = np.linspace(sx.min(),sx.max(), num=201)
xi = 10**(xi_)

f = interpolate(sx,numdat_2, kind="cubic")
yi = f(xi_)
plt.plot(xi,yi, label="cubic spline")


f1 = interpolate(sx,numdat_2, kind="line")
yi = f1(xi_)
plt.plot(xi,yi, label="line spline")

plt.gca().set_xscale("log")
plt.legend()
plt.show()
```

```
degree = 4
poly_fit = np.poly1d(np.polyfit(num1,numdat_2, degree))

xx = np.linspace(0, 30, 100)
plt.plot(xx, poly_fit(xx), c='r',linestyle='-')
plt.title('Polynomial')
plt.xlabel('num1')
plt.ylabel('num2')
plt.axis([0, 25, 0, 100])
plt.grid(True)
plt.scatter(num1, numdat_2)
plt.show()

degree = 5
poly_fit = np.poly1d(np.polyfit(num1,numdat_2, degree))

xx = np.linspace(0, 30, 100)
plt.plot(xx, poly_fit(xx), c='r',linestyle='-')
plt.title('Polynomial')
plt.xlabel('num1')
plt.ylabel('num2')
plt.axis([0, 25, 0, 100])
plt.grid(True)
plt.scatter(num1, numdat_2)
plt.show()
```

Program Outputs:

I could not get a program output from the code.

Discussions:

In this problem, I successfully performed options a and b. However, although I wrote the code for options c and d, I could not get a printout. For this reason, I could not do the e option. In addition, I got an error in option c and could not solve this error.

```
x and y must have same first dimension, but have shapes (149, 1) and (50, 1)
```