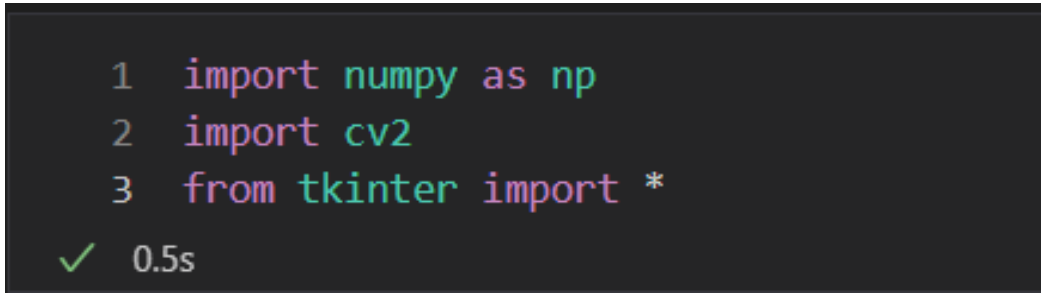
DIGITAL IMAGE PROJECT

KADIR GOKDENIZ & RAMAZAN DAGHAN



These are all libraries that we used in our programs.Numpy provides np arrays for images. Cv2 was used for some fuctions such as imread(),imshow,filter2D... Also, we import tkinter to create label and button widgets.

```
window=Tk()
window.title("Image Project")
window.geometry('690x200')

label1=Label(window,text='Number of image:(1 to 5)',fg='blue',font=('Arial',13))
label1.grid(row=0,column=0,padx=5,pady=5)

string1=StringVar()
entry1=Entry(window,width=20,textvariable=string1)
entry1.grid(row=0,column=1,padx=30,pady=5)

label2=Label(window,text='Filter: Average, Negative, Sharpen, Laplacian or Logarithm:',fg='blue',font=('Arial',13))
label2.grid(row=1,column=0,padx=5,pady=5)

string2=StringVar()
entry2=Entry(window,width=20,textvariable=string2)
entry2.grid(row=1,column=1,padx=5,pady=5)

label3=Label(window,text='Sensivity for comparison:(1,2,4,16,32,64,128 or 256)',fg='blue',font=('Arial',13))
label3.grid(row=3,column=0,padx=5,pady=5)
```

Firstly, we create the window.Then, we give the title name and size.After that, we have the label and entry parts.The aim of the this section is creating suitable environment for user to select wanted image.Also,we have a similar manner for choosing a filter.User can select each filter in Average,Negative,Sharpen ,Laplacian or Logarithm filter.In this part, user must type on keyboard only filter name with capital for first letter.

```
string2=StringVar()
entry2=Entry(window,width=20,textvariable=string2)
entry2.grid(row=1,column=1,padx=5,pady=5)

label3=Label(window,text='Sensivity for comparison:(1,2,4,16,32,64,128 or 255)',fg='blue',font=('Arial',13))
label3.grid(row=3,column=0,padx=5,pady=5)

string3=StringVar()
entry3=Entry(window,width=20,textvariable=string3)
entry3.grid(row=3,column=1,padx=5,pady=5)

label4=Label(window,text="Similarity",fg='purple',font=('Arial',13))
label4.grid(row=5,column=0,padx=5,pady=5)

string4=StringVar()
sens=Entry(window,width=20, state="readonly",textvariable=string4)
sens.grid(row=5,column=1,padx=5,pady=5)

button1=Button(window,text='Start',fg='green',font=('Arial',13),command=startFunction)
button1.grid(row=4,column=1,sticky=W)
window.mainloop()
```

We have two type of image in this project.The first one is original image and the second one is 'filtered' image. We wanted to add a section where we can see the comparison result by taking the sensitivity rate by the user with the help of a function that scales how much the images change.User can enter any number less than 255.But we think that entering multiples of two makes a more obvious difference in similarity results.We have also a start button to run the all program.We have a similarty part that the result is written.

```
26  def startFunction():
27      if string2.get()=="Average":
28          result=averageFilter(string1.get(),string3.get())
29          string4.set('%'+str(result))
30
31      if string2.get()=="Negative":
32          result=negativeFilter(string1.get(),string3.get())
33          string4.set('%'+str(result))
34
35      if string2.get()=="Sharpen":
36          result=sharpingFilter(string1.get(),string3.get())
37          string4.set('%'+str(result))
38
39      if string2.get()=="Laplacian":
40          result=LaplacianFilter(string1.get(),string3.get())
41          string4.set('%'+str(result))
42
43      if string2.get()=="Logarithm":
44          result=logarithmImage(string1.get(),string3.get())
45          string4.set('%'+str(result))
```

Clicking on start button triggers startFuction and this function calls suitable filter functions.All filtering functions return sensivity rate and it binds readonly entry label with '%'.

```python
def similarity(new,original,sensivity):
    list1=new.tolist()
    size1=new.size
    x=len(list1)
    y=int(size1/x)
    counter=0
    range1=round(255/(int(sensivity)))
    list2=original.tolist()
    for i in range(0,x):
        for j in range(0,y):
            originalp=list2[i][j]
            newp=list1[i][j]
            if(originalp+range1>=newp and originalp-range1<=newp):
                counter+=1
    return round(counter*100/(x*y))
```

✓ 0.4s

This is similarty function. This function gets original image,'filtered' image and sensivity.

Since we want to find the density of each pixel as an integer inside the for loops, we converted numpy arrays into list. 'counter' counts if pixel intensity is desired range.And percentage of counter is returned.

```python
def averageFilter(img,sensivity):
    filepath="C:\\Users\\Asus F15\\Desktop\\1\\"+img+".jpg"
    original=cv2.imread(filepath,0)
    original=cv2.resize(original,(450,450))
    new=cv2.imread(filepath,0)
    new=cv2.resize(new,(450,450))
    converter=np.ones((5,5),np.float32)/25
    new=cv2.filter2D(src=new, ddepth=-1, kernel=converter)
    cv2.imshow("Original",original)
    cv2.imshow("Averaged",new)
    return similarity(new,original,sensivity)
```

Average filter is a 5x5 filter consist of ones.With the help of the filter2D function, the original photo is filtered and the 'filtered' photo and the original photo are printed.

```
def negativeFilter(img,sensivity):
    filepath="C:\\Users\\Asus F15\\Desktop\\1\\"+img+".jpg"
    original=cv2.imread(filepath,0)
    original=cv2.resize(original,(450,450))
    new=cv2.imread(filepath,0)
    new=cv2.resize(new,(450,450))
    new=255-original
    cv2.imshow('Original', original)
    cv2.imshow('Negative Filter', new)
    return similarity(new,original,sensivity)
```

✓ 0.4s

Negative filter is got by 255-original image.With the help of the filter2D function,the original photo is filtered and the 'filtered' photo and the original photo are printed.

```python
def sharpingFilter(img,sensivity):
    filepath="C:\\Users\\Asus F15\\Desktop\\1\\"+img+".jpg"
    original=cv2.imread(filepath,0)
    original=cv2.resize(original,(450,450))
    new=cv2.imread(filepath,0)
    new=cv2.resize(new,(450,450))
    converter=np.array([[0,-1,0],[-1,5,-1],[0,-1,0]])
    new=cv2.filter2D(src=new, ddepth=-1, kernel=converter)
    cv2.imshow('Original', original)
    cv2.imshow('Sharpened
    return similarity(new,original,sensivity)
```

(variable) original: Mat

Sharpening filter consists of [0 –1 0] [-1 5 -1] [0 –1 0]. With the help of the filter2D function, the original photo is filtered and the 'filtered' photo and the original photo are printed.

```python
def LaplacianFilter(img,sensivity):
    filepath="C:\\Users\\Asus F15\\Desktop\\1\\"+img+".jpg"
    original=cv2.imread(filepath,0)
    original=cv2.resize(original,(450,450))
    new=cv2.imread(filepath,0)
    new=cv2.resize(new,(450,450))
    converter=np.array([[0,1,0],[1,-4,1],[0,1,0]])
    new=cv2.filter2D(src=new, ddepth=-1, kernel=converter)
    cv2.imshow('Original', original)
    cv2.imshow('Laplacian Image',new )
    return similarity(new,original,sensivity)
```

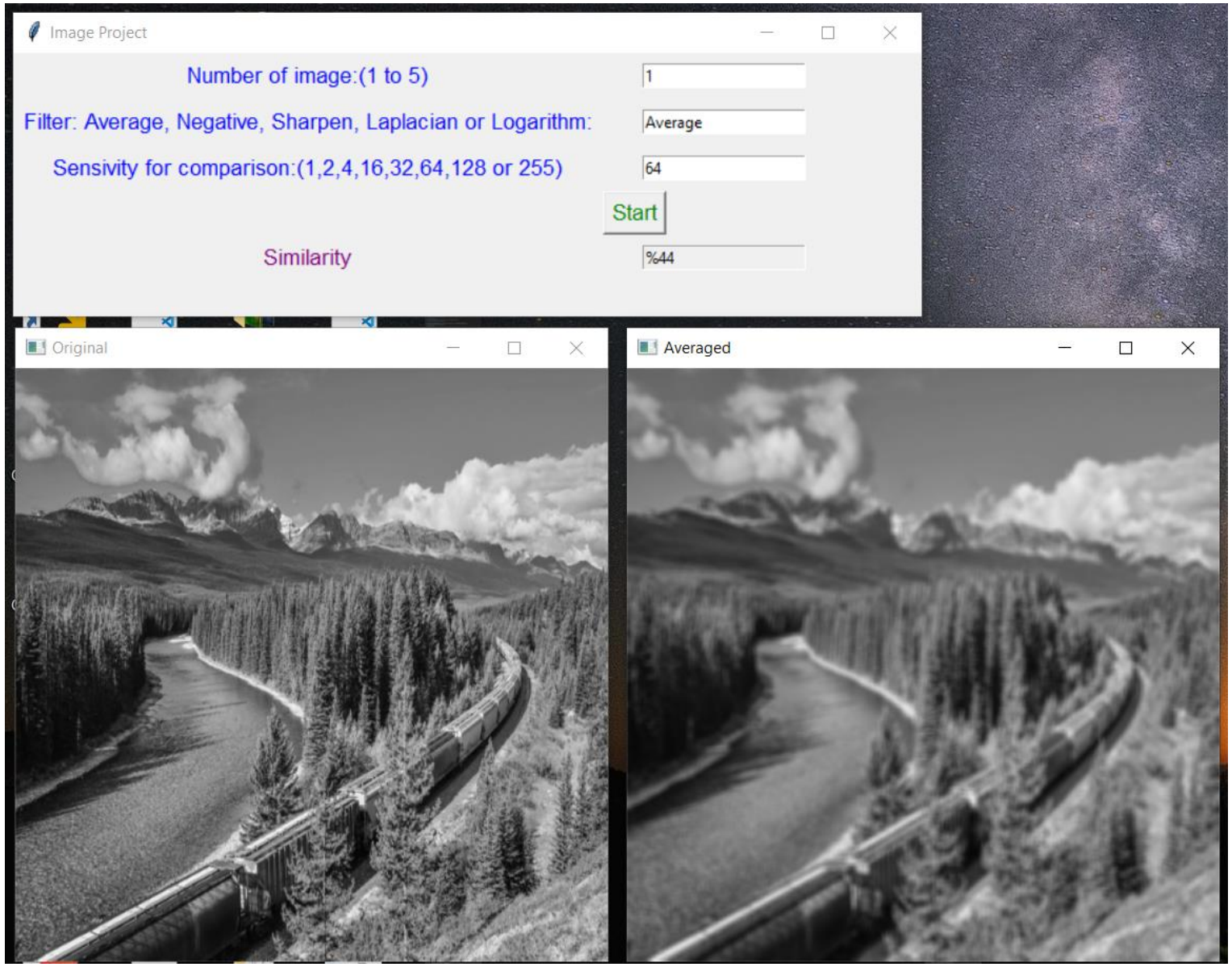
Laplacian filter consist of [0 1 0][1 –4 1][0 1 0].With the help of the filter2D function,the original photo is filtered and the 'filtered' photo and the original photo are printed.

```python
def logarithmImage(img,sensivity):
    filepath="C:\\Users\\Asus F15\\Desktop\\1\\"+img+".jpg"
    original=cv2.imread(filepath,0)
    original=cv2.resize(original,(450,450))
    new=cv2.imread(filepath,0)
    new=cv2.resize(new,(450,450))
    new=np.uint8(np.log1p(new))
    numb=1
    new=cv2.threshold(new,numb,255,cv2.THRESH_BINARY)[1]
    cv2.imshow('Original', original)
    cv2.imshow('Logarithm Image',ne
    return similarity(new,original,sensivity)
```

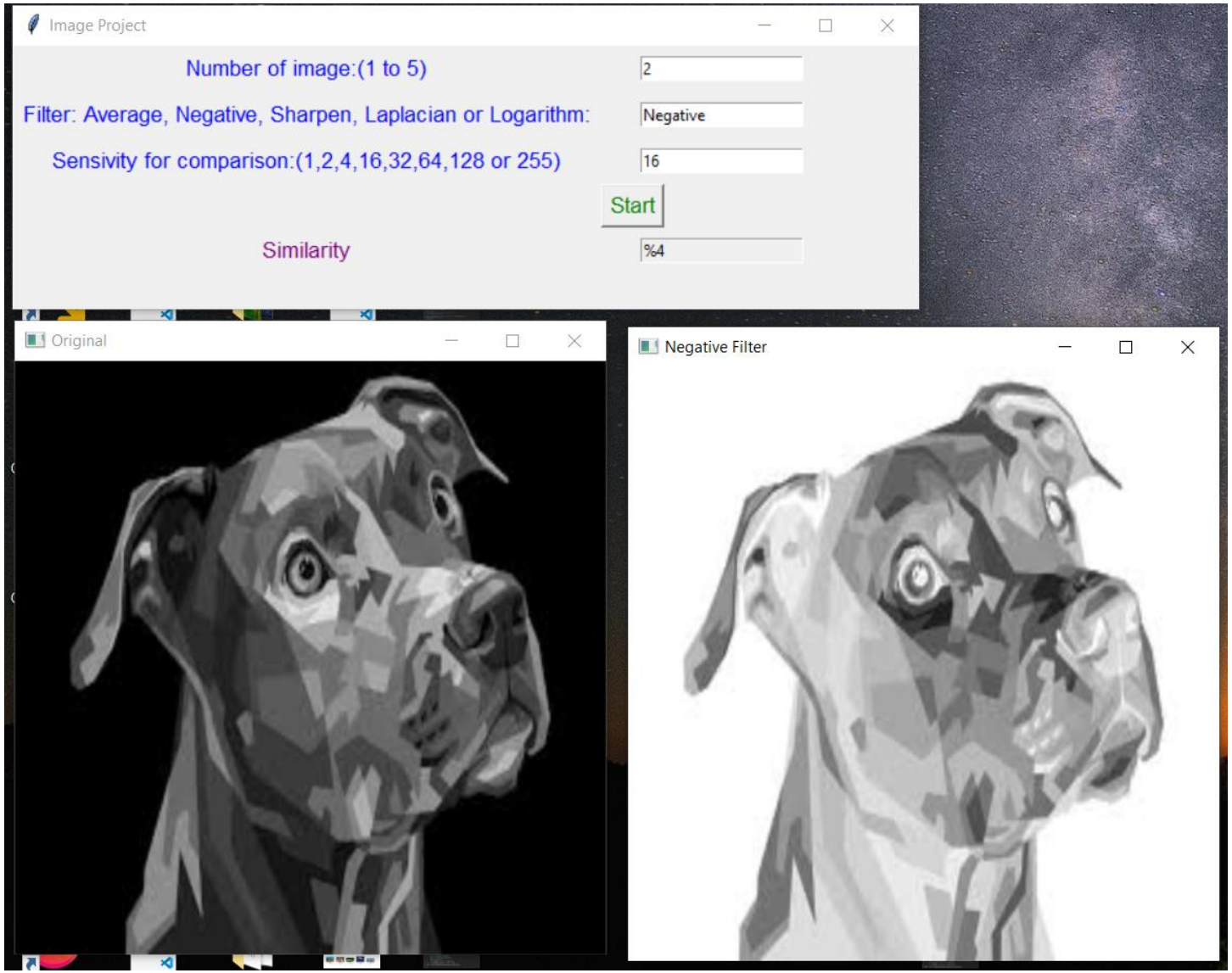
(parameter) sensivity: Any

Logarithm filter is created by treshold function.Firstly we convert the image logarithm style by typing np.uint8(np.log1p(new)). Then we need a number for using treshold function. After that, typing threshold(new,numb,255,cv2.THRESH_BINARY)[1]cause logarithm image.Finally, the original image and 'filtered' images are printed.

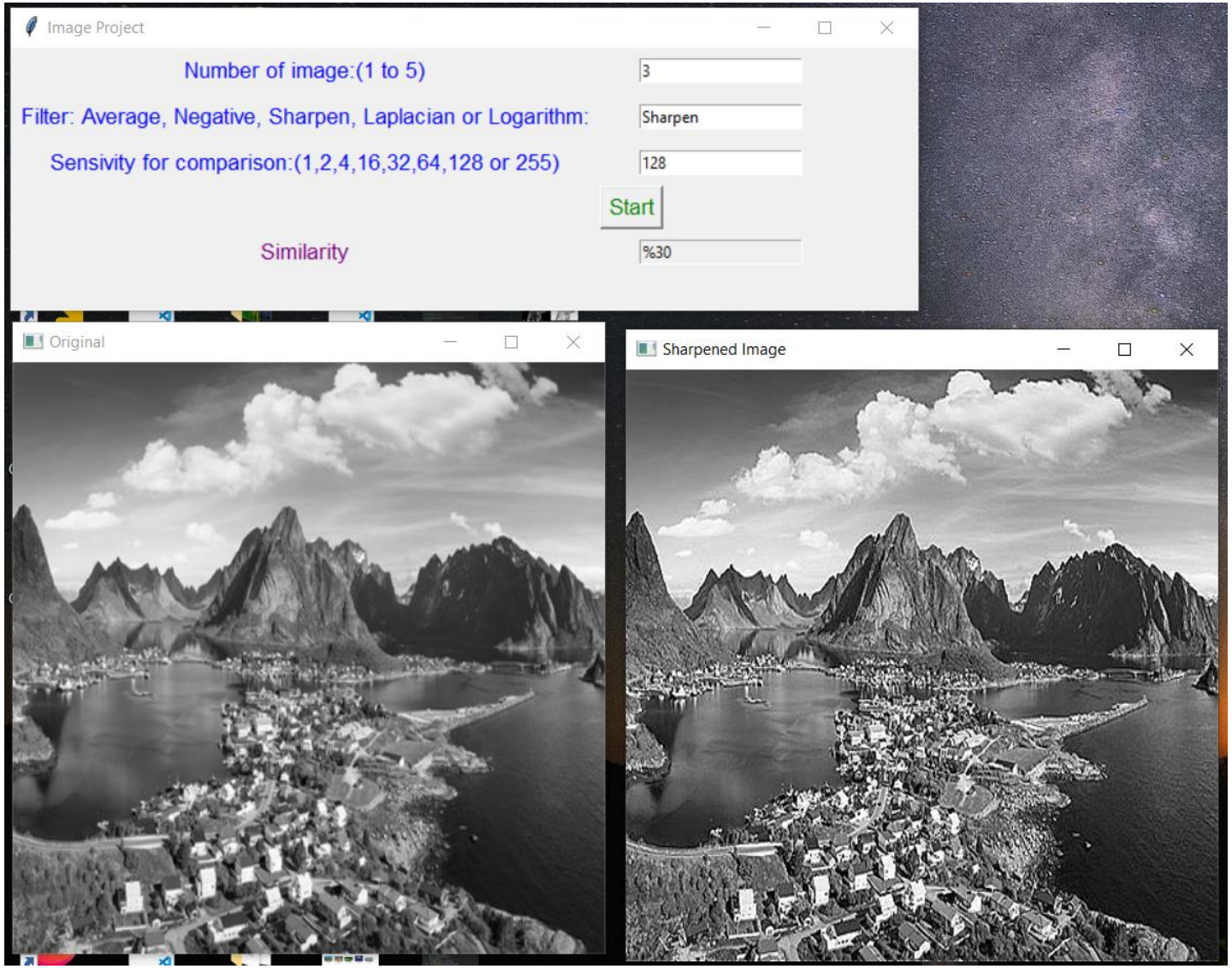
AVERAGE FILTER



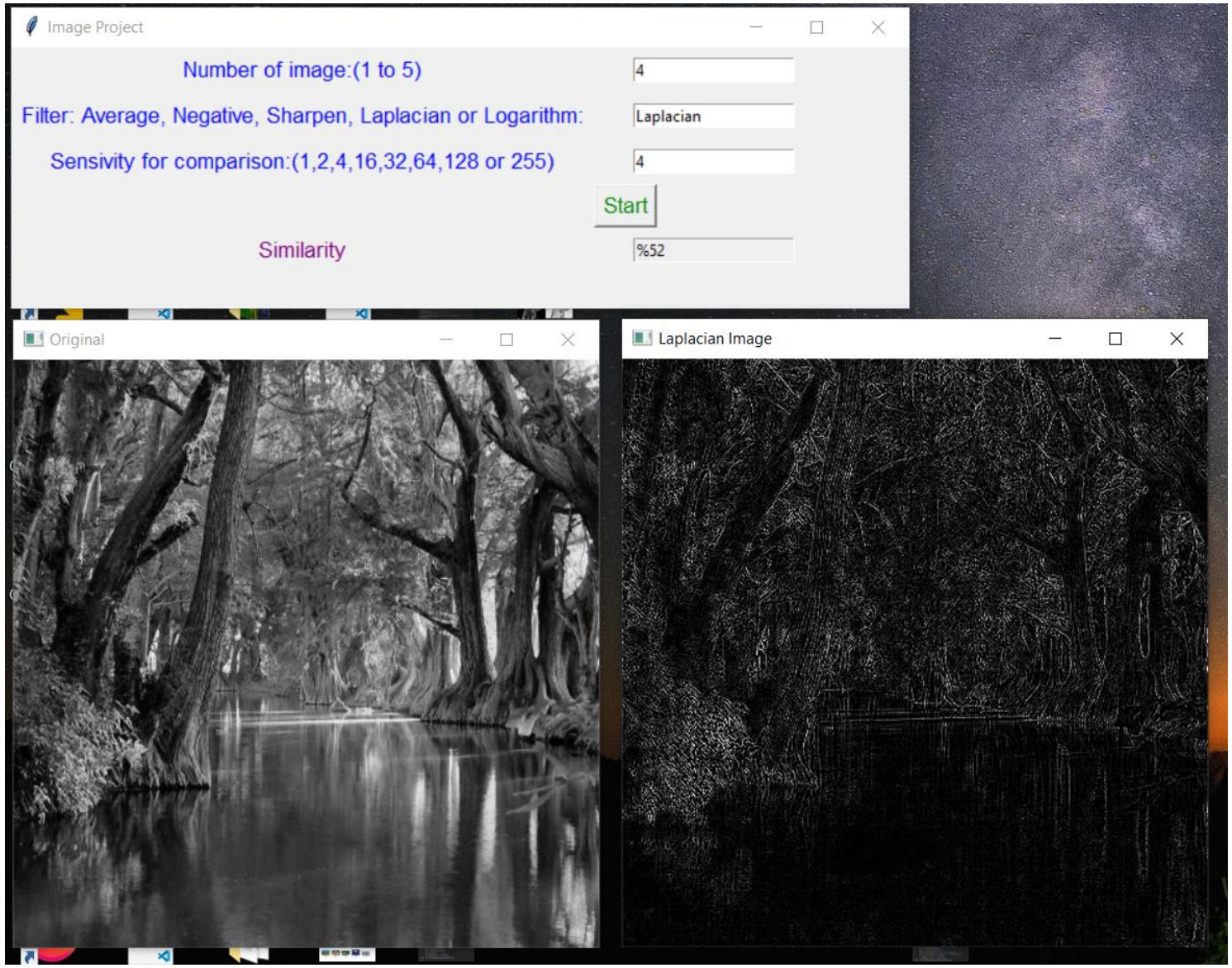
NEGATIVE FILTER



SHARPENING FILTER



LAPLACIAN FILTER



LOGARITHM FILTER