HACETTEPE UNIVERSITY

2021-2022 SPRING

GMT234

DIGITAL DISPLAY AND INTERPRETATION

MERT ÇETİNTÜRK

21967387

Question 1

Before we start to write our code, we first imported the libraries we will use. Then we read and display the image with the help of OpenCV. Below, we have printed the gray scale dimensions of our images.

In [1]:
```
import numpy as np
import cv2
import matplotlib.pyplot as plt
from PIL import Image
```

In [2]:
```
image1 = cv2.imread('Lena.png',0)
cv2.imshow('Lena',image1)
cv2.waitKey(0)
cv2.destroyAllWindows()
```

In [3]:
```
image1.shape
```

Out[3]: (512, 512)

In [4]:
```
image2 = cv2.imread('yellow_flowers.png',0)
cv2.imshow('Yellow Flowers',image2)
cv2.waitKey(0)
cv2.destroyAllWindows()
```

In [5]:
```
image2.shape
```

Out[5]: (512, 512)

Then we wrote a function where we can easily filter the images,

In [6]:
```
def convert(image,kernel):          #here i created a function where we can filter images.
    output = np.zeros_like(image)

    image_padded = np.zeros((image.shape[0]+2,image.shape[1]+2))
    image_padded[1:-1,1:-1] = image

    for y in range(image.shape[0]):
        for x in range(image.shape[1]):
            output[x, y] = (kernel * image_padded[x: x+3,y: y+3]).sum()

    return output
```

After that, we defined the filters we will apply and we wrote our filtered images with the help of matplotlib.

```
In [7]: KERNEL = np.array([[1, 0, -1],[1, 0, -1],[1, 0, -1]])

        KERNEL2 = np.array([[0, -0.1, 0],[-0.2, 0.6, -0.2],[0, -0.1, 0]])

        KERNEL3 = np.array([[1, 0, 0],[0, 0, 0],[0, 0, 0]])

        plt.figure(figsize = (20,7))
        plt.subplot(121)
        plt.title("Original Image's Gray Scale")
        plt.imshow(image1, cmap = 'gray')

        image_con = convert(image1, KERNEL)    #Sharpening Filter

        plt.figure(figsize = (20,7))
        plt.subplot(121)
        plt.title("Lena With First Filter")
        plt.imshow(image_con, cmap = 'gray')


        image_con2 = convert(image1, KERNEL2)  #Sharpening Filter

        plt.figure(figsize = (20,7))
        plt.subplot(121)
        plt.title("Lena With Second Filter")
        plt.imshow(image_con2, cmap = 'gray')


        image_con3 = convert(image1, KERNEL3)    #Smoothing Filter

        plt.figure(figsize = (20,7))
        plt.subplot(121)
        plt.title("Lena With Third Filter")
        plt.imshow(image_con3, cmap = 'gray')

Out[7]: <matplotlib.image.AxesImage at 0x2ba1a77fd60>
```

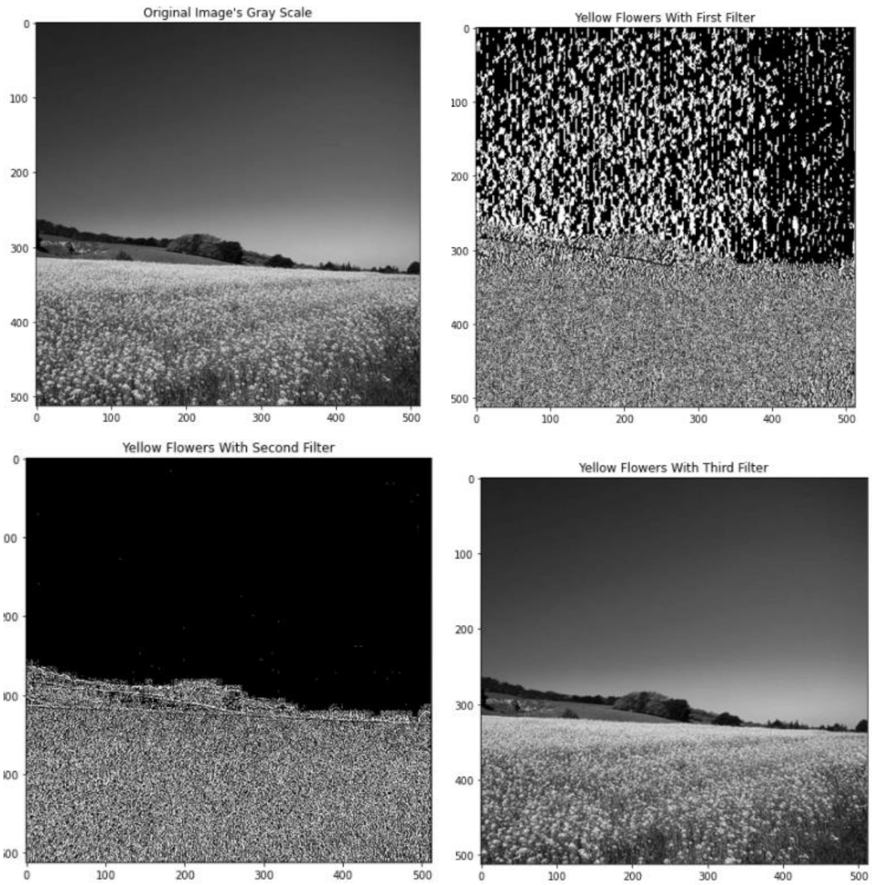
Here you can see our original image and filtered images.

We do the same for our other image.

In [8]:
```
plt.figure(figsize = (20,7))
plt.subplot(121)
plt.title("Original Image's Gray Scale")
plt.imshow(image2, cmap = 'gray')

image_con_2 = convert(image2, KERNEL)  #Sharpening Filter

plt.figure(figsize = (20,7))
plt.subplot(121)
plt.title("Yellow Flowers With First Filter")
plt.imshow(image_con_2, cmap = 'gray')


image_con_2_2 = convert(image2, KERNEL2)   #Sharpening Filter

plt.figure(figsize = (20,7))
plt.subplot(121)
plt.title("Yellow Flowers With Second Filter")
plt.imshow(image_con_2_2, cmap = 'gray')


image_con_2_3 = convert(image2, KERNEL3)   #Smoothing Filter

plt.figure(figsize = (20,7))
plt.subplot(121)
plt.title("Yellow Flowers With Third Filter")
plt.imshow(image_con_2_3, cmap = 'gray')
```

Out[8]: <matplotlib.image.AxesImage at 0x2ba1a98a1f0>

Here you can see our original image and filtered images.



And here we save our filtered images as tiff file.

In [11]:
```
lena1 = cv2.imwrite('Lena 1.tiff',image_con)
lena2 = cv2.imwrite('Lena 2.tiff',image_con2)
lena3 = cv2.imwrite('Lena 3.tiff',image_con3)

yellow_flowers1 = cv2.imwrite('Yellow Flowers 1.tiff',image_con_2)
yellow_flowers2 = cv2.imwrite('Yellow Flowers 2.tiff',image_con_2_2)
yellow_flowers3 = cv2.imwrite('Yellow Flowers 3.tiff',image_con_2_3)
```

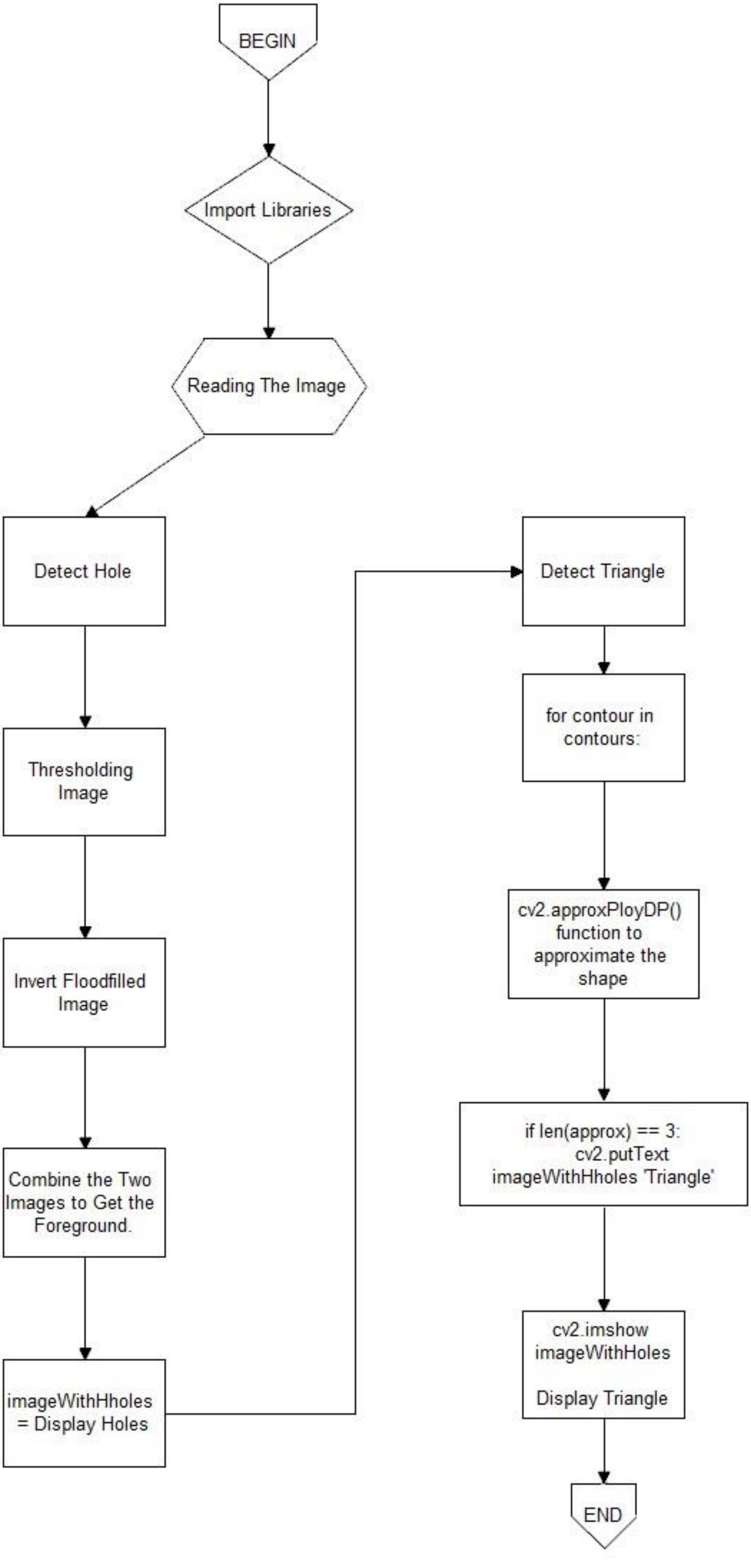
Question 2

BEGIN

```
import cv2

import numpy as np


image = Read Image


th, image_threshold = Thresholding Set values equal to or above
220 to 0. Set values below 220 to 255.


im_floodfill = Copy the thresholded image


h, w = image_threshold Shape


mask = Notice the size needs to be 2 pixels than the image.


Floodfill from point (0, 0)


Invert floodfilled image


Combine the two images to get the foreground.


imageWithHholes = Display Holes




import cv2

import numpy as np

from matplotlib import pyplot as plt


imageWithHoles = reading image with cv2.imread
```

```
contours, _ = using a findContours() function


for contour in contours:

    if i == 0:

        i = 1

        continue


    approx = cv2.approxPloyDP() function to approximate the shap
e


    using drawContours() function


    putting shape name at center of each shape


    if len(approx) == 3:

        cv2.putText imageWithHholes 'Triangle'


    elif len(approx) == 4:

        cv2.putText imageWithHholes 'Quadrilateral'


    else:

        cv2.putText 'circle'


cv2.imshow imageWithHoles


Display Triangle
```

END


BEGIN
Import Libraries
Reading The Image
Detect Hole
Thresholding Image
Invert Floodfilled Image
Combine the Two Images to Get the Foreground.
imageWithHholes = Display Holes
Detect Triangle
for contour in contours:
cv2.approxPloyDP() function to approximate the shape
if len(approx) == 3: cv2.putText imageWithHholes 'Triangle'
cv2.imshow imageWithHoles
Display Triangle
END

Question 3

Google Lens

Google is a significant brand in Augmented Reality. The Google Lens software recognizes the contents of the camera and displays information about them on the screen. This application, which provides a wealth of information to its users, such as the name of the flower, when you show the camera a flower you see while crossing the street, is the most recent update to the Google and Google Goggles concept, which uses the computing power of the cloud to identify the contents.

Any CAD program

AR is extensively used in CAD applications to display conceptual designs on a computer as they are intended to look in the real world. This is referred to as visualization and it allows the user to test various options of an object before it's built. Besides visualization, AR is applied in actual manufacturing processes like quality control inspections, training, and maintenance. Some CAD software employs AR such as SOLIDWORKS, Autodesk, PTC, and Augment. With the use of AR in CAD, a user can display their design in full scale and in the environment of interest, which means clients can see an exact visual of a product/part concept and simulate them on the real environment.

Pokémon Go

While Star Trek's holodeck is assumed as science fiction, engineering industries are using virtual reality (VR) in various engineering applications. For instance, VR is used in the gaming industry. VR enables games to feel like a real 3D environment that can interact with other players throughout the game. Since the VR concept has definitely made a footprint on industrial applications, there is an opportunity to integrate virtual reality with real-life experiences to create augmented reality. This is basically an improved reality where an operator sees and experiences the real world surrounding them, but with the incorporation of computer-generated graphics in connection with what is "really" there. A great example of this is Nintendo's Pokémon Go for smartphones, which integrates the game as a layer on top of your actual geographic location for a fully interactive experience.