

c2011050-ceng462-hw2

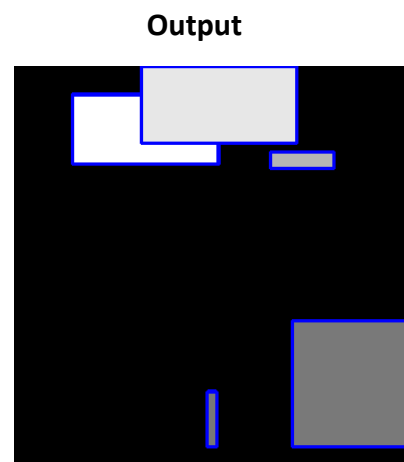
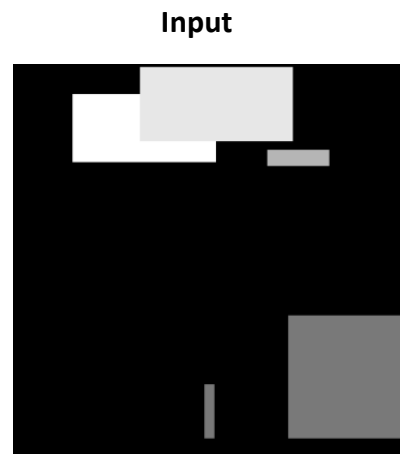
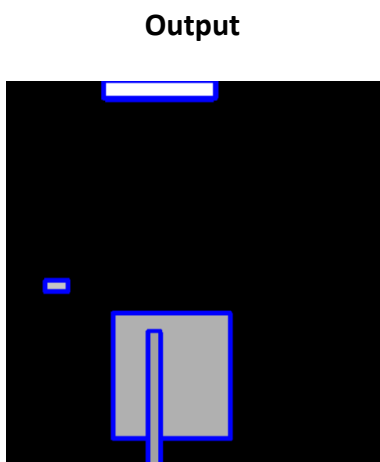
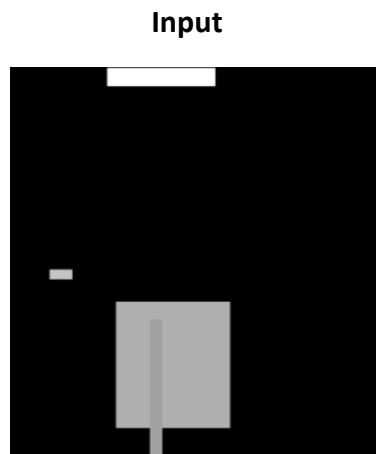
March 20, 2024

```
[14]: from PIL import Image

image=Image.open('C:/Users/dogan/Pictures/Screenshots/10.png') # Add my image
width,height=image.size #get the height and width of the image
contours=[] #The list I created to store the coordinates of the edges on the
↳image.
for y in range(1,height-1): # Loop height
    for x in range(1, width-1): # Loop width
        currentPixel=image.getpixel((x,y)) # CURRENT PIXEL
        topPixel=image.getpixel((x, y-1)) # TOP
        bottomPixel=image.getpixel((x,y+1)) # BOTTOM
        leftPixel=image.getpixel((x-1,y)) # LEFT
        rightPixel=image.getpixel((x+1, y)) # RIGHT

        if currentPixel != leftPixel or currentPixel != rightPixel or
↳currentPixel != topPixel or currentPixel != bottomPixel: # Check if the
↳current pixel is different from the edges
            contours.append((x, y)) # Add to
for point in contours: # loop for each marked point
    image.putpixel(point, (0,0,255)) # Marking Blue
image.show() # show
```

```
[ ]:
```



462 Homework 2

This is my second homework assignment. I chose Python language while doing this assignment. As a result of my long research, I was able to get sufficient output. This assignment was quite instructive. I mentioned the sources I used while doing this assignment in the reference section.

Code Work

First of all, I needed to develop an algorithm. This algorithm will cycle the pixels in order. It will visit the neighbors of the pixel it is located in, and if there is a difference in these neighbors, it will be perceived as an edge. I needed to get some output from the homework we did last week. I received 2 sample outputs. [2] I exported these images to Python via file path and used the Image.open PIL library for this. Then I took the width and height of the picture. I created a list of contours because this will hold our edge coordinates. [3] I looped around the pixels with nested loops and used image.getpixel to get the coordinates of the pixel if it was a different color than its neighbors. Then I added it to our list. [1] Using image.putpixel I added a blue pixel to each

coordinate in this list. This allowed me to create the rectangle. Finally, I got the output with image.show.

Reference

- [1] <https://www.geeksforgeeks.org/python-pil-putpixel-method/>
- [2] <https://www.geeksforgeeks.org/python-pil-image-open-method/>
- [3] <https://www.geeksforgeeks.org/python-pil-getpixel-method/>