BM3000
Foundations of Natural Intelligence

Assignment-1

Syed Saqib Habeeb
BM20BTECH11015

Formulation of algorithm:

The algorithm I designed is to detect black dots on a white background. Here all the white pixels are grouped together and perceived as one block and all the black pixels are perceived as another block and hence this algorithm revolves around the gestalt law “similarity”.

In this the image file is read in the form of a grayscale image. A threshold limit for contour is set for a particular pixel to be considered as a black pixel or a white pixel. If any pixel in the image has a contour more than the threshold, then the program returns a message “Dot/line detected”. If there is no pixel in the image file which has a contour greater than the threshold the program returns a message “No dot/line detected”.

Proof:

Below are the images that I used in the program.

—> Image 1 (Line on a white background)

—> Image 2 (White background with no dots/lines

—> Image 3 (White background with dots on it)

I first read the image file in its grayscale format, set the threshold for the contour of the pixels and then returned the message.

In [53]:
```
import cv2
```

In [54]:
```
image1=cv2.imread('image1.png',0)
```

In [55]:
```
image2=cv2.imread('image2.png',0)
```

In [56]:
```
image3=cv2.imread('image3.png',0)
```

In [57]:
```
th,threshed = cv2.threshold(image1, 100, 225,cv2.THRESH_BINARY_INV|cv2.THRESH_OTSU)
```

In [58]:
```
cnts = cv2.findContours(threshed, cv2.RETR_LIST,cv2.CHAIN_APPROX_SIMPLE)[-2]
```

In [59]:
```
s1 = 3
s2 = 50000000000000000
xcnts = []
for cnt in cnts:
    if s1<cv2.contourArea(cnt) <s2:
        xcnts.append(cnt)
```

In [60]:
```
if (len(xcnts)>0):
    print("Dot/line detected")

else:
    print("No dot/line detected")
```