OBJECT DETECTION

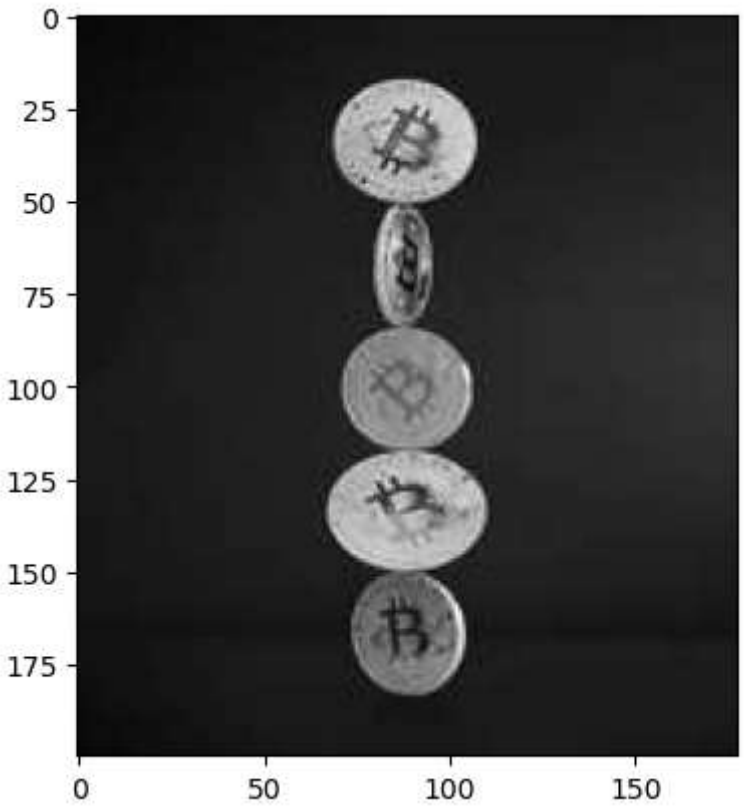
Object detection is a computer vision technique for locating instances of objects in images or videos. Object detection algorithms typically leverage machine learning or deep learning to produce meaningful results.

In this we require OpenCv is an open-source library that is useful for computer vision applications such as image processing, video processing, facial recognition, and detection, etc.

In [2]:
```
#Import libraries

import cv2
import numpy as np
import matplotlib.pyplot as plt
image = cv2.imread('lab.jpg')
gray = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)
plt.imshow(gray, cmap='gray')
```

Out[2]: `<matplotlib.image.AxesImage at 0x1b5b7e97460>`

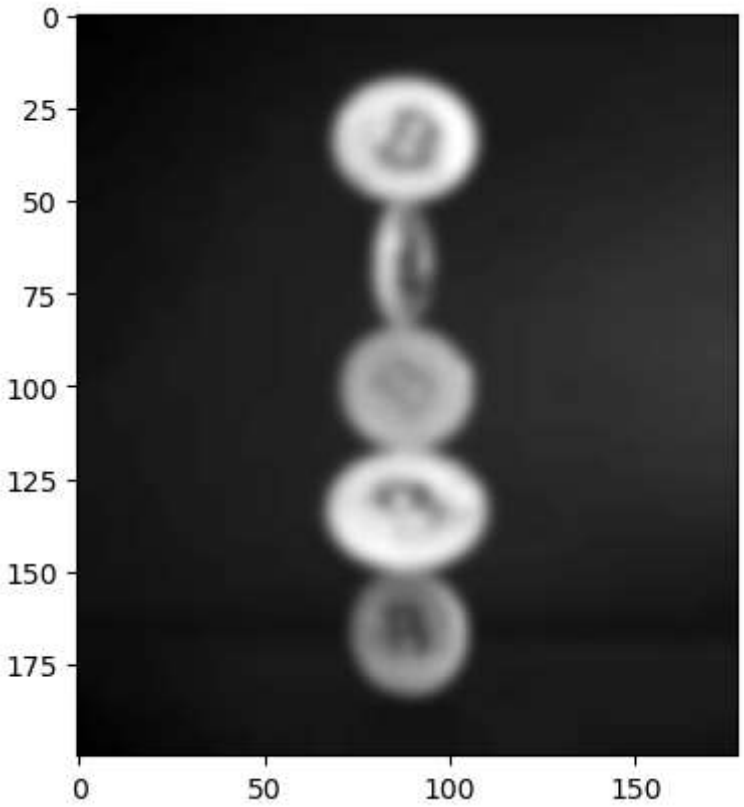


We will read the image by using “cv2.imread(image-name)” command & then convert this image into grayscale image using “cv2.cvtColor(image-name,

In [3]:

```
blur = cv2.GaussianBlur(gray, (11, 11), 0)
plt.imshow(blur, cmap='gray')
```

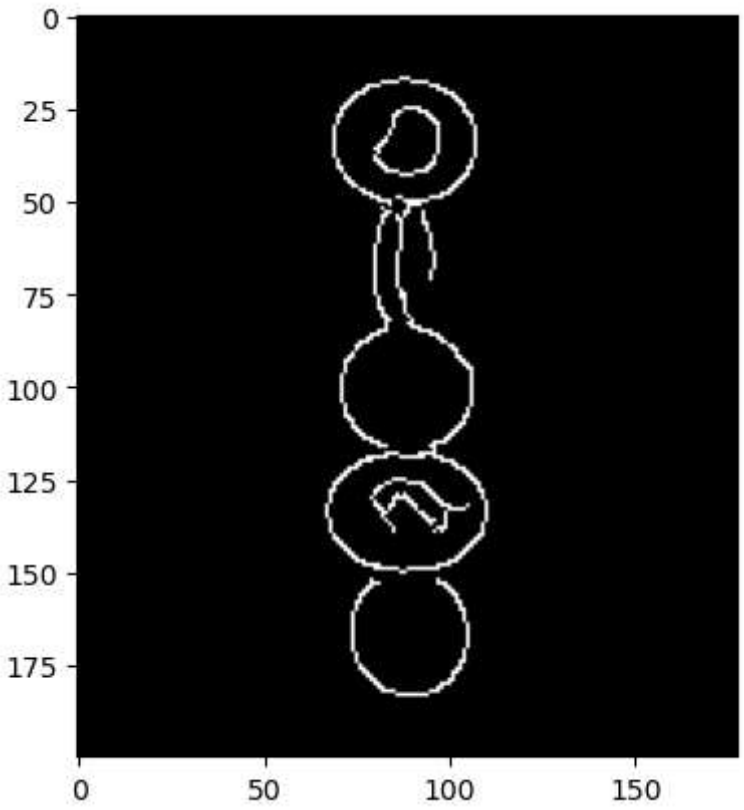
Out[3]: <matplotlib.image.AxesImage at 0x1b5b7ee3430>



We will read the image by using “cv2.imread(image-name)” command & then convert this image into grayscale image using “cv2.cvtColor(image-name, cv2.COLOR_BGR2GRAY)” command.

In [4]:
```
canny = cv2.Canny(blur, 30, 150, 3)
plt.imshow(canny, cmap='gray')
```

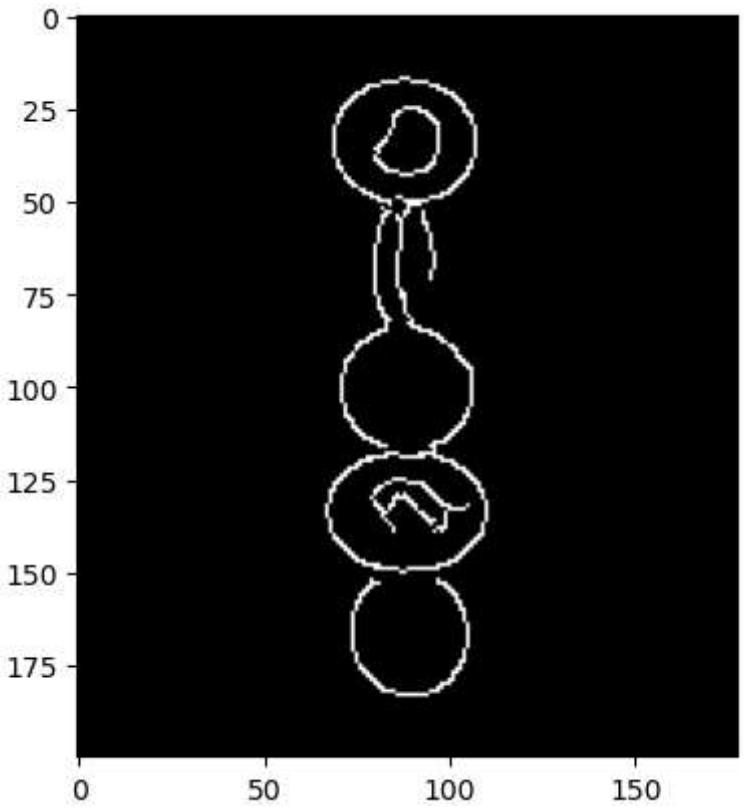
Out[4]: <matplotlib.image.AxesImage at 0x1b5b7f49a90>



We can see that edges are not connected. We need to connect the edges, have to make more thiker & visible.

In [5]:
```
dilated = cv2.dilate(canny, (1, 1), iterations=0)
plt.imshow(dilated, cmap='gray')
```

Out[5]:
```
<matplotlib.image.AxesImage at 0x1b5b7fa9c40>
```

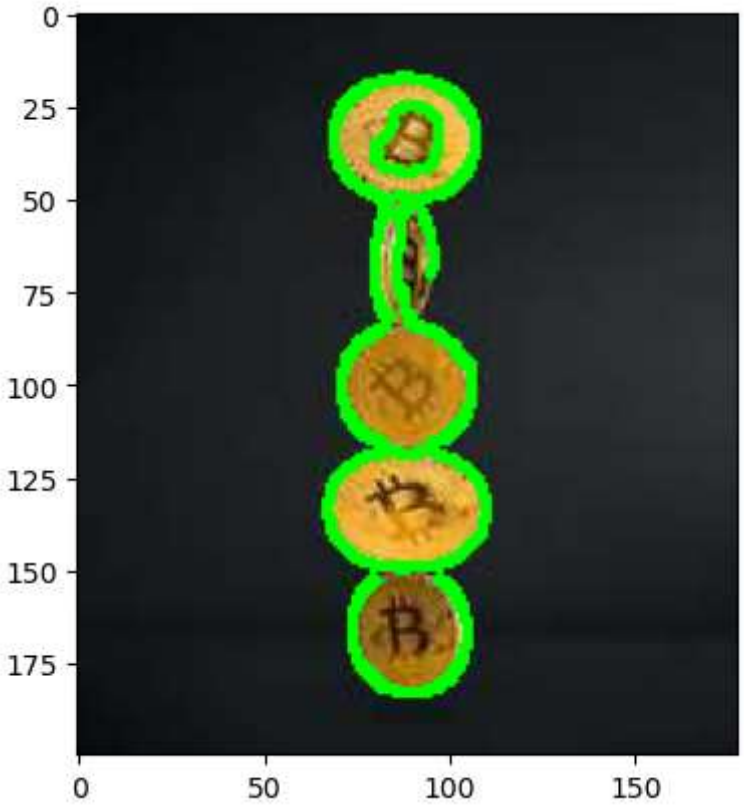


Now we have to calculate the contour in the image & convert the image into RGB from BGR & then draw the contours.

In [7]:
```
(cnt, hierarchy) = cv2.findContours(dilated.copy(), cv2.RETR_EXTERNAL, cv2.CHA
rgb = cv2.cvtColor(image, cv2.COLOR_BGR2RGB)
cv2.drawContours(rgb, cnt, -1, (0, 255, 0), 2)

plt.imshow(rgb)
```

Out[7]: <matplotlib.image.AxesImage at 0x1b5b80f2310>



canny = cv2.Canny(blur, 30, 150, 3) plt.imshow(canny, cmap='gray')

In [8]:
```
print("coins in the image : ", len(cnt))
```

```
coins in the image :  5
```

CONCLUSION : We need to Convert the image to grayscale and apply a Gaussian blur. Then Perform edge detection using the Canny algorithm. Dilate the edges so that we can have better identification of objects in image.After that we need to find contours and draw them on the original image, and print the count of detected coins.