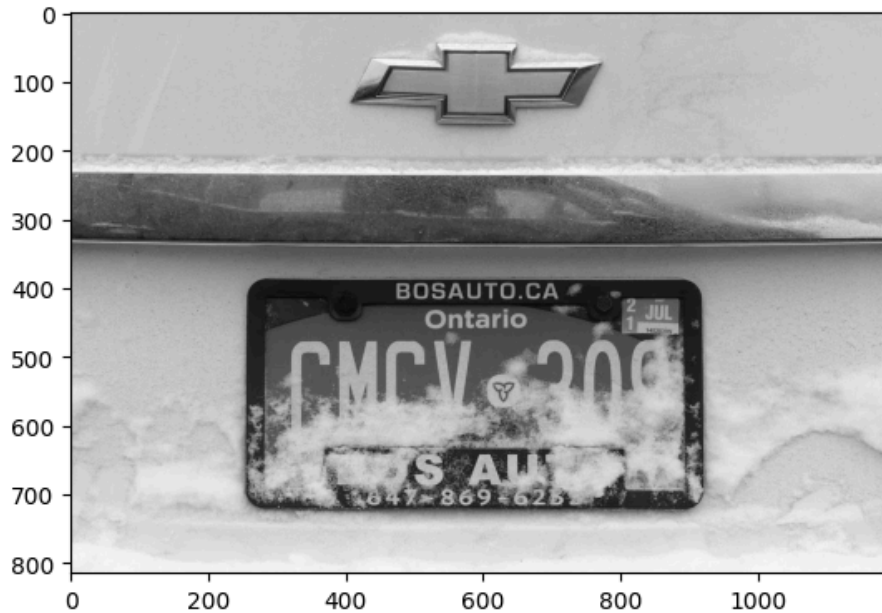```
In [132…   import cv2
           from matplotlib import pyplot as plt
           import numpy as np
           import imutils
           import easyocr
```

## 1. Read in Image, Grayscale

```
In [133…   img = cv2.imread('../data/medium3.jpg')
           gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
           plt.imshow(cv2.cvtColor(gray,cv2.COLOR_BGR2RGB))
```
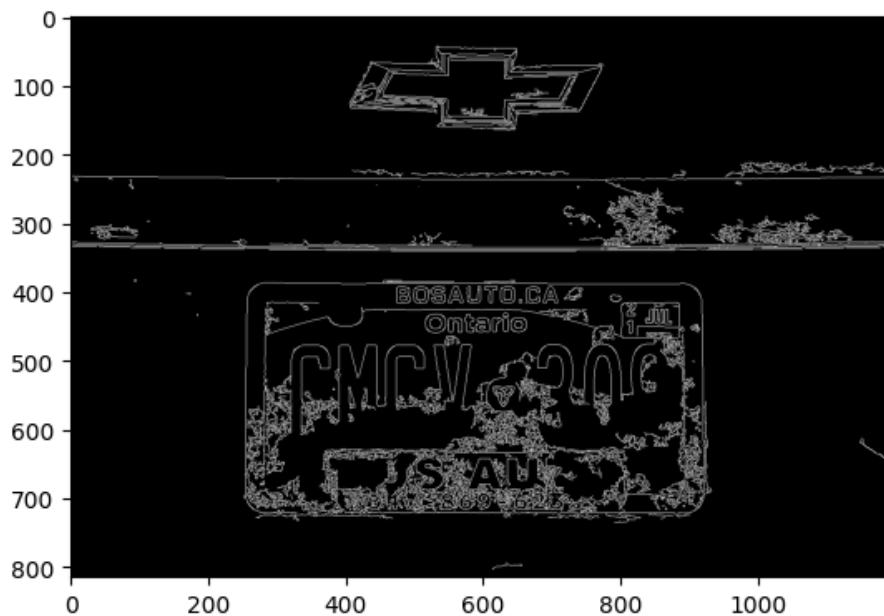
Out[133]:   <matplotlib.image.AxesImage at 0x3090b6c60>



## 2. Apply filter and find edges for localization

```
In [134…   bfilter = cv2.bilateralFilter(gray, 11, 17, 17) #Noise reduction
           edged = cv2.Canny(bfilter, 10, 200) #Edge detection
           plt.imshow(cv2.cvtColor(edged, cv2.COLOR_BGR2RGB))
```

Out[134]:   <matplotlib.image.AxesImage at 0x309175a60>

3. Find contours and apply masks

In [135…
```
keypoints = cv2.findContours(edged.copy(), cv2.RETR_TREE, cv2.CHAIN_APPROX_SIMPLE)
contours = imutils.grab_contours(keypoints)
contours = sorted(contours, key=cv2.contourArea, reverse=True)[:10]
```

In [136…
```
location = None
for contour in contours:
    approx = cv2.approxPolyDP(contour, 10, True)
    if len(approx) == 4:
        location = approx
        break
```
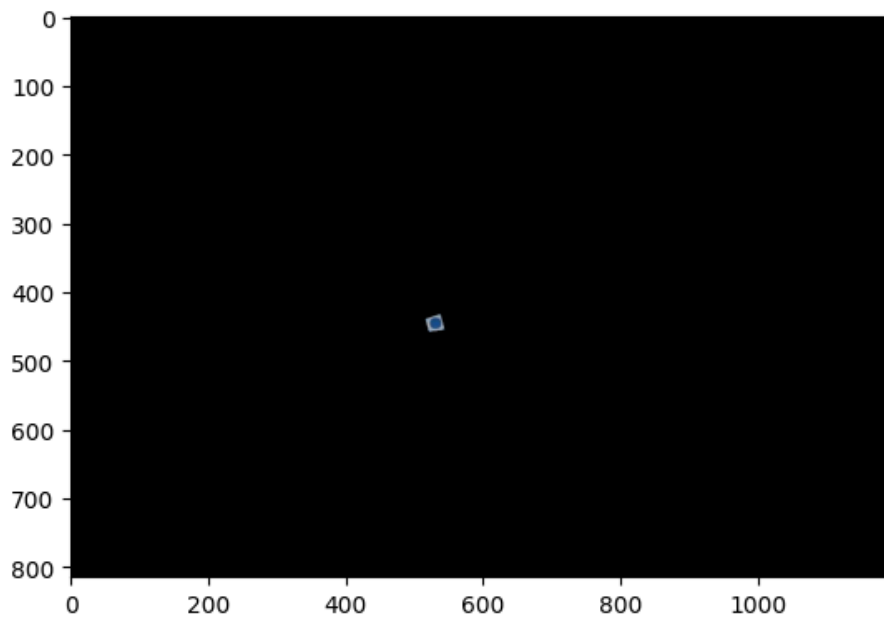
In [137…
```
location
```

Out[137]:
```
array([[[538, 433]],

       [[517, 440]],

       [[522, 458]],

       [[544, 455]]], dtype=int32)
```

In [138…
```
mask = np.zeros(gray.shape, np.uint8)
new_image = cv2.drawContours(mask, [location], 0, 255, -1)
new_image = cv2.bitwise_and(img, img, mask=mask)

plt.imshow(cv2.cvtColor(new_image, cv2.COLOR_BGR2RGB))
```
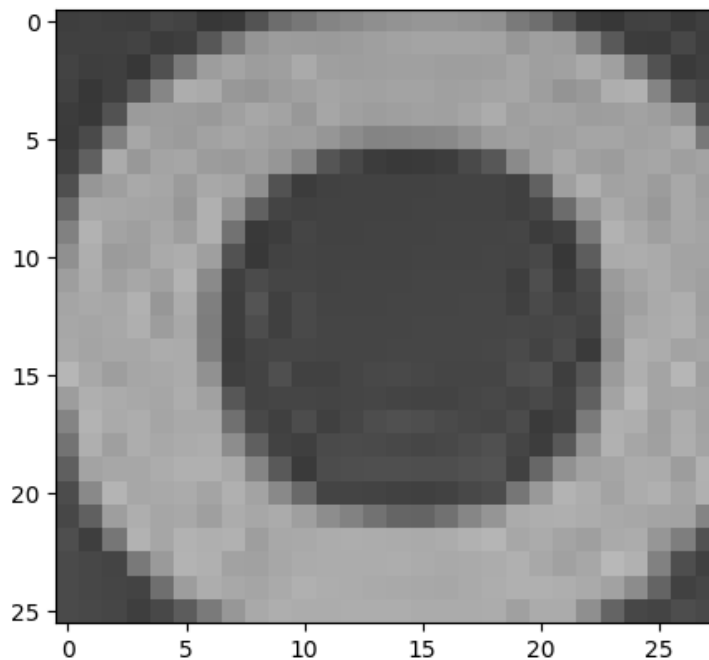
Out[138]:
```
<matplotlib.image.AxesImage at 0x3091b6300>
```

```
In [139…  (x,y) = np.where(mask==255)
          (x1, y1) = (np.min(x), np.min(y))
          (x2, y2) = (np.max(x), np.max(y))
          cropped_image = gray[x1:x2+1, y1:y2+1]
```

```
In [140…  plt.imshow(cv2.cvtColor(cropped_image, cv2.COLOR_BGR2RGB))
```

Out[140]: <matplotlib.image.AxesImage at 0x309224d70>

4. Use Easy OCR to Read Text

```
In [141…  reader = easyocr.Reader(['en'])
          result = reader.readtext(cropped_image)
          result
```

Out[141]: [([[0, 0], [28, 0], [28, 26], [0, 26]], '0', np.float64(0.7110873834763062))]

5. Render Result

```
text = result[0][-2]
font = cv2.FONT_HERSHEY_SIMPLEX
res = cv2.putText(img, text=text, org=(approx[0][0][0], approx[1][0][1]+60), fontFace=font, fontSc
res = cv2.rectangle(img, tuple(approx[0][0]), tuple(approx[2][0]), (0,255,0),3)
plt.imshow(cv2.cvtColor(res, cv2.COLOR_BGR2RGB))
```

Out[142]: <matplotlib.image.AxesImage at 0x3090e8170>