

```
In [110... import cv2
from matplotlib import pyplot as plt
import numpy as np
import imutils
import easyocr
```

## 1. Read in Image, Grayscale

```
In [111... img = cv2.imread('../data/medium2.jpg')
gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
plt.imshow(cv2.cvtColor(gray, cv2.COLOR_GRAY2RGB))
```

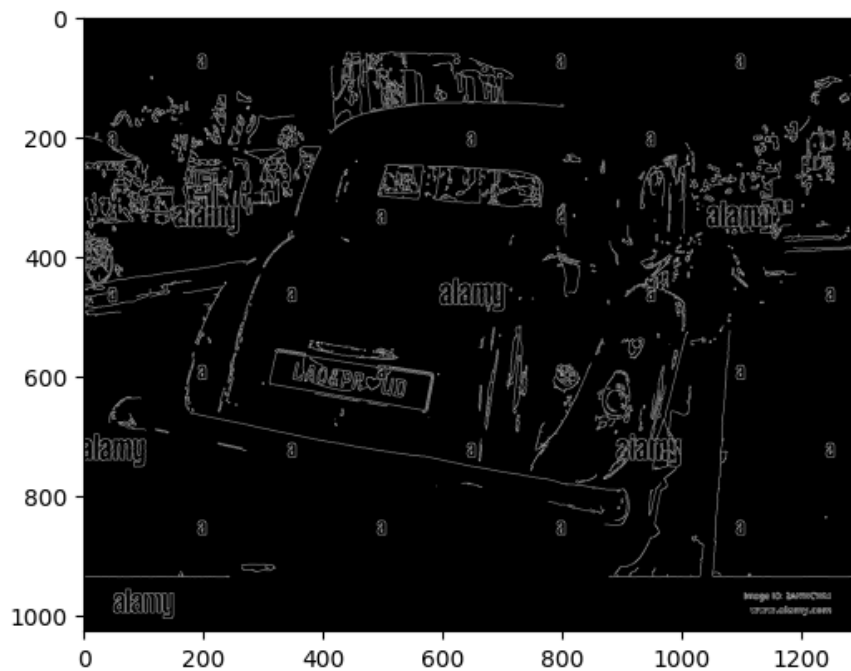
Out[111]: <matplotlib.image.AxesImage at 0x2bdc7e8a0>



## 2. Apply filter and find edges for localization

```
In [112... bfilter = cv2.bilateralFilter(gray, 11, 17, 17) #Noise reduction
edged = cv2.Canny(bfilter, 10, 200) #Edge detection
plt.imshow(cv2.cvtColor(edged, cv2.COLOR_GRAY2RGB))
```

Out[112]: <matplotlib.image.AxesImage at 0x2bdceae10>



### 3. Find contours and apply masks

```
In [113...] 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 [114...] location = None
for contour in contours:
    approx = cv2.approxPolyDP(contour, 10, True)
    if len(approx) == 4:
        location = approx
        break
```

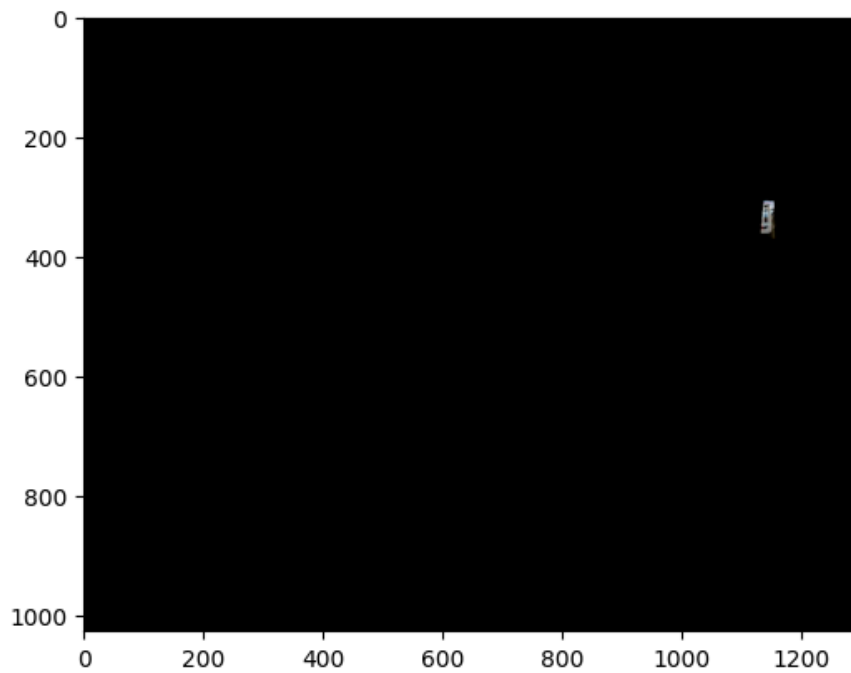
```
In [115...] location
```

```
Out[115]: array([[1137, 308]],
                [[1132, 362]],
                [[1155, 372]],
                [[1154, 309]]], dtype=int32)
```

```
In [116...] mask = np.zeros(gray.shape, np.uint8)
new_image = cv2.drawContours(mask, [location], 0, 255, -1)
new_image = cv2.bitwise_and(img, img, mask=new_image)

plt.imshow(cv2.cvtColor(new_image, cv2.COLOR_BGR2RGB))
```

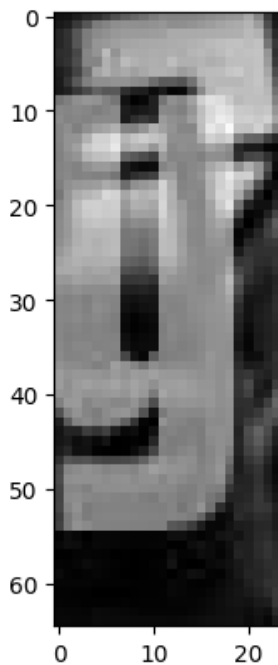
```
Out[116]: <matplotlib.image.AxesImage at 0x301859160>
```



```
In [117... (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 [118... plt.imshow(cv2.cvtColor(cropped_image, cv2.COLOR_BGR2RGB))
```

```
Out[118]: <matplotlib.image.AxesImage at 0x2bd8a35f0>
```



## 4. Use Easy OCR to Read Text

```
In [119... reader = easyocr.Reader(['en'])
result = reader.readtext(cropped_image)
result
```

```
Out[119]: [([[0, 0], [24, 0], [24, np.int32(54)], [0, np.int32(54)]],  
          '4',  
          np.float64(0.15936436965393241))]
```

## 5. Render Result

```
In [120]: text = result[0][-2]  
font = cv2.FONT_HERSHEY_SIMPLEX  
res = cv2.putText(img, text=text, org=(approx[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[120]: <matplotlib.image.AxesImage at 0x2bdc06e40>
```

