

```
In [56]: import cv2
from matplotlib import pyplot as plt
import numpy as np
import imutils
import easyocr
```

1. Read in Image, Grayscale

```
In [57]: img = cv2.imread('../data/good1.jpg')
gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
plt.imshow(cv2.cvtColor(gray, cv2.COLOR_BGR2RGB))
```

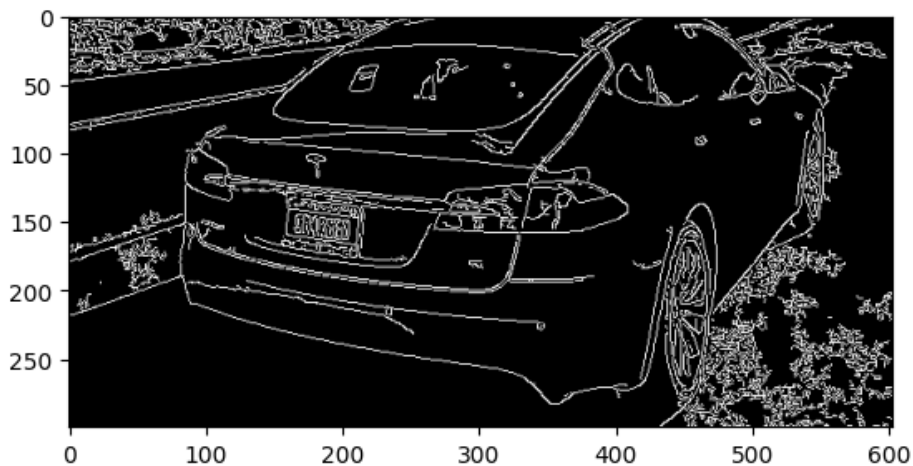
Out[57]: <matplotlib.image.AxesImage at 0x2b52e8bf0>



2. Apply filter and find edges for localization

```
In [58]: 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[58]: <matplotlib.image.AxesImage at 0x29df6f170>



3. Find contours and apply masks

```
In [59]: 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 [60]: location = None
for contour in contours:
    approx = cv2.approxPolyDP(contour, 10, True)
    if len(approx) == 4:
        location = approx
        break
```

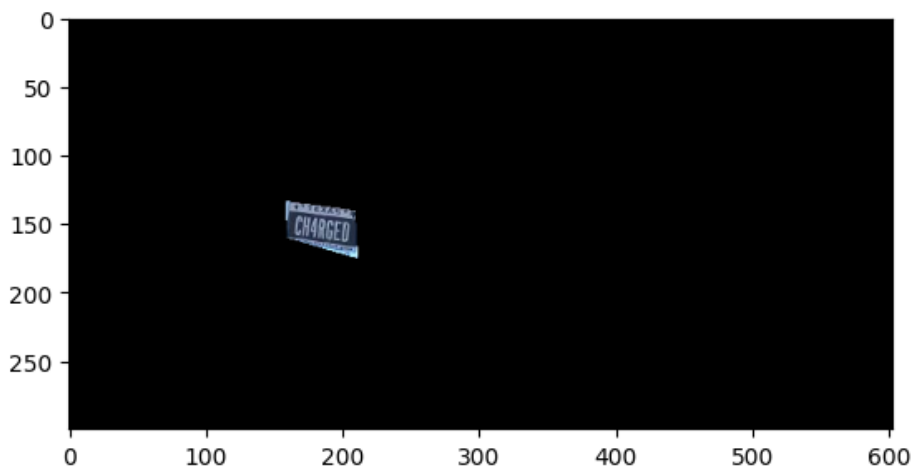
```
In [61]: location
```

```
Out[61]: array([[159, 134]],
               [[160, 160]],
               [[211, 175]],
               [[209, 141]]], dtype=int32)
```

```
In [62]: 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))
```

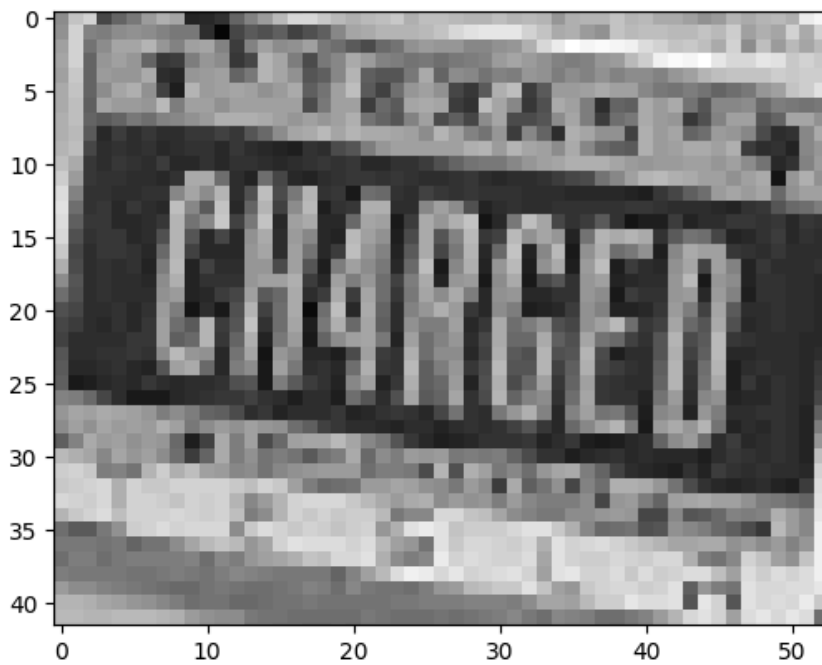
```
Out[62]: <matplotlib.image.AxesImage at 0x2ada570e0>
```



```
In [63]: (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 [64]: plt.imshow(cv2.cvtColor(cropped_image, cv2.COLOR_BGR2RGB))
```

```
Out[64]: <matplotlib.image.AxesImage at 0x29dfe7950>
```



4. Use Easy OCR to Read Text

```
In [65]: reader = easyocr.Reader(['en'])
result = reader.readtext(cropped_image)
result
```

```
Out[65]: [([np.float64(2.9580569325292183), np.float64(3.186753717828889)],
 [np.float64(52.868599866522295), np.float64(11.121857183717665)],
 [np.float64(48.04194306747078), np.float64(35.81324628217111)],
 [np.float64(-1.868599866522295), np.float64(27.878142816282335)]],
 'IcHARGED',
 np.float64(0.08439777213668437))]
```

5. Render Result

```
In [66]: 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[66]: <matplotlib.image.AxesImage at 0x2b559c3e0>
```

