In [99]:
```
import cv2
from matplotlib import pyplot as plt
import numpy as np
import imutils
import easyocr
```

1. Read in Image, Grayscale

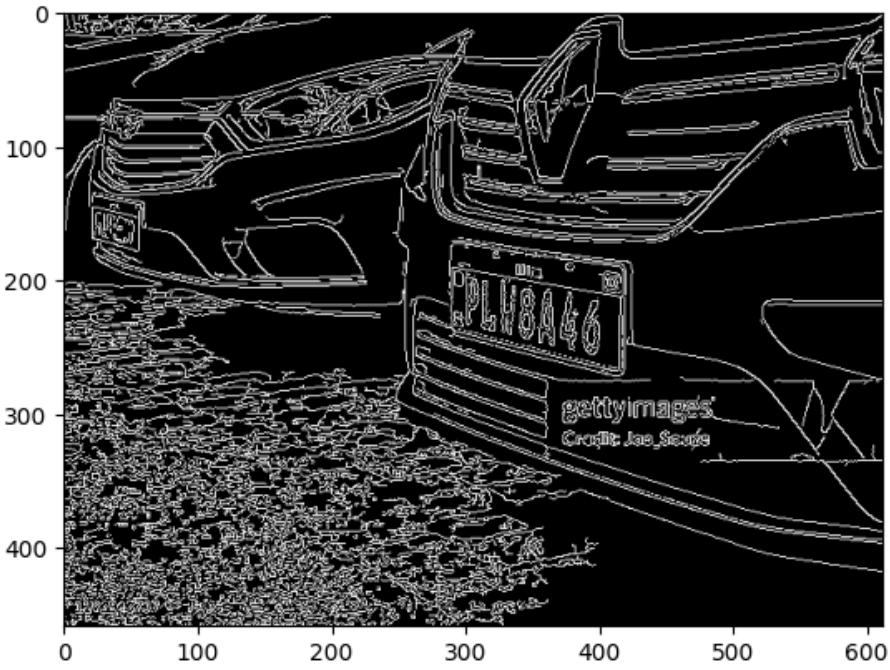
In [100…
```
img = cv2.imread('../data/medium1.jpg')
gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
plt.imshow(cv2.cvtColor(gray,cv2.COLOR_BGR2RGB))
```

Out[100]:
```
<matplotlib.image.AxesImage at 0x2bda4b050>
```

2. Apply filter and find edges for localization

In [101…
```
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[101]:
```
<matplotlib.image.AxesImage at 0x2ad969e80>
```



3. Find contours and apply masks

In [102…
```
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]
```

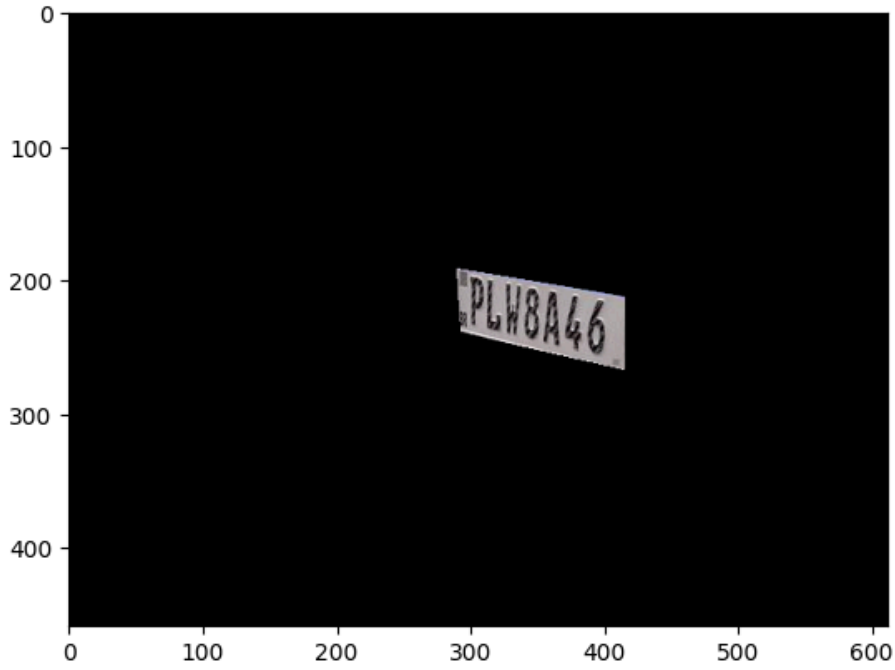
In [103…
```
location = None
for contour in contours:
    approx = cv2.approxPolyDP(contour, 10, True)
    if len(approx) == 4:
        location = approx
        break
```

In [104…
```
location
```

Out[104]:
```
array([[[290, 192]],

       [[415, 213]],

       [[415, 267]],

       [[293, 239]]], dtype=int32)
```

In [105…
```
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))
```

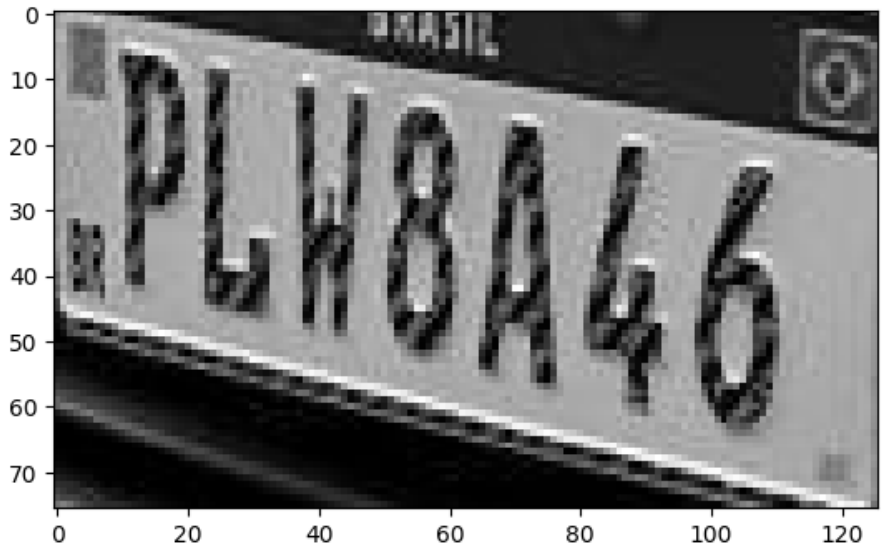
Out[105]:
```
<matplotlib.image.AxesImage at 0x2b55270b0>
```



In [106…
```
(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 [107…
```
plt.imshow(cv2.cvtColor(cropped_image, cv2.COLOR_BGR2RGB))
```

Out[107]:
```
<matplotlib.image.AxesImage at 0x2b5af3560>
```



4. Use Easy OCR to Read Text

In [108…
```
reader = easyocr.Reader(['en'])
result = reader.readtext(cropped_image)
result
```

Out[108]:
```
[([[np.float64(5.1856870722510084), np.float64(-6.944484372567638)],
   [np.float64(129.91195253775297), np.float64(22.975882237554604)],
   [np.float64(114.81431292774899), np.float64(77.94448437256764)],
   [np.float64(-8.911952537752974), np.float64(47.0241177624454)]],
  'PLHBAL6',
  np.float64(0.44828177155152443))]
```

5. Render Result

In [109…
```
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[109]: <matplotlib.figure.AxesImage at 0x29e2990a0>