

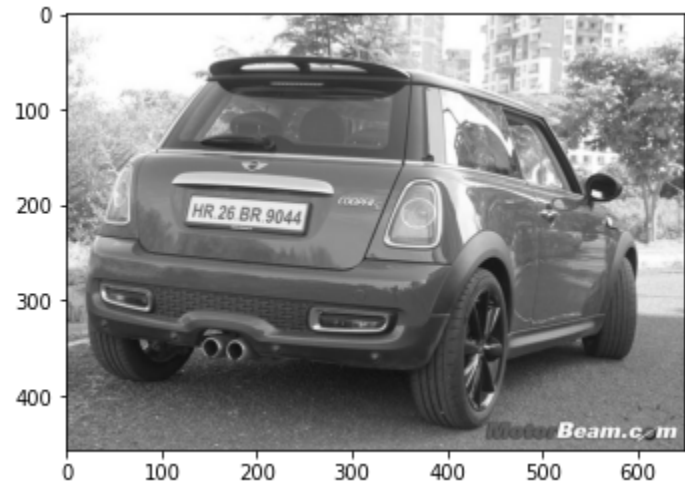
Import Libraries

```
In [1]: import cv2
from matplotlib import pyplot as plt
import numpy as np
import imutils
import easyocr
```

1. Read in Image, Grayscale and Blur

```
In [22]: img = cv2.imread('image1.jpg')
gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
plt.imshow(cv2.cvtColor(gray, cv2.COLOR_BGR2RGB))
```

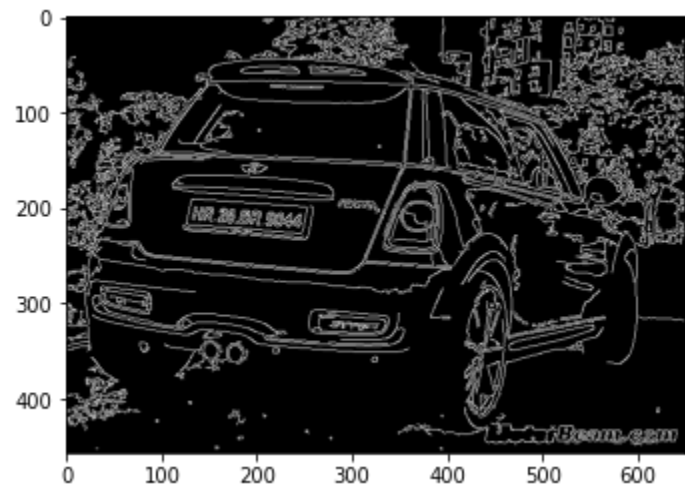
Out[22]: <matplotlib.image.AxesImage at 0x2d9dd49cee0>



2. Apply filter and find edges for localization

```
In [23]: bfilter = cv2.bilateralFilter(gray, 11, 17, 17) #Noise reduction
edged = cv2.Canny(bfilter, 30, 200) #Edge detection
plt.imshow(cv2.cvtColor(edged, cv2.COLOR_BGR2RGB))
```

Out[23]: <matplotlib.image.AxesImage at 0x2d9d81b99d0>



3. Find Contours and Apply Mask

```
In [24]: 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 [25]: location = None
for contour in contours:
    approx = cv2.approxPolyDP(contour, 10, True)
    if len(approx) == 4:
        location = approx
        break
```

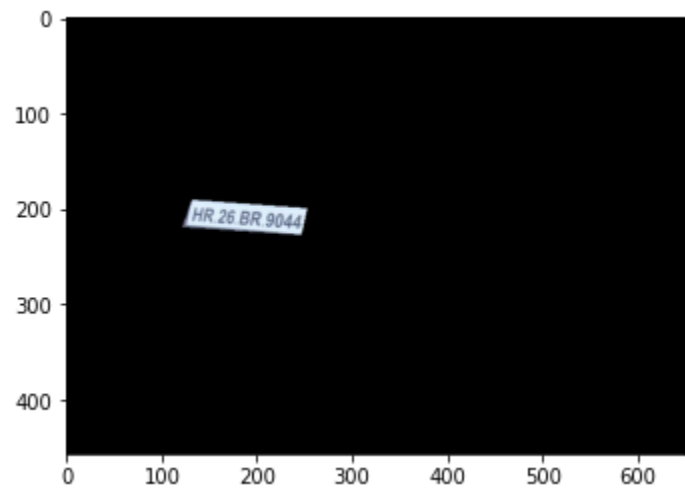
```
In [94]: location
```

Out[94]: array([[300, 540]],
 [[306, 589]],
 [[543, 592]],
 [[538, 543]]], dtype=int32)

```
In [26]: 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)
```

```
In [27]: plt.imshow(cv2.cvtColor(new_image, cv2.COLOR_BGR2RGB))
```

Out[27]: <matplotlib.image.AxesImage at 0x2d9ddab9fa0>



```
In [28]: (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 [29]: plt.imshow(cv2.cvtColor(cropped_image, cv2.COLOR_BGR2RGB))
```

Out[29]: <matplotlib.image.AxesImage at 0x2d9d82d9160>



4. Use Easy OCR To Read Text

```
In [30]: reader = easyocr.Reader(['en'])
result = reader.readtext(cropped_image)
result
```

CUDA not available - defaulting to CPU. Note: This module is much faster with a GPU.

Out[30]: [[['6, 4], [128, 4], [128, 34], [6, 34]], 'HR.26 BR.9044', 0.5728024956877317]]

5. Render Result

```
In [31]: 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, fontScale=1, color=(0,255,0), thickness=2, lineType=cv2.LINE_AA)
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[31]: <matplotlib.image.AxesImage at 0x2d9e78a4b50>

