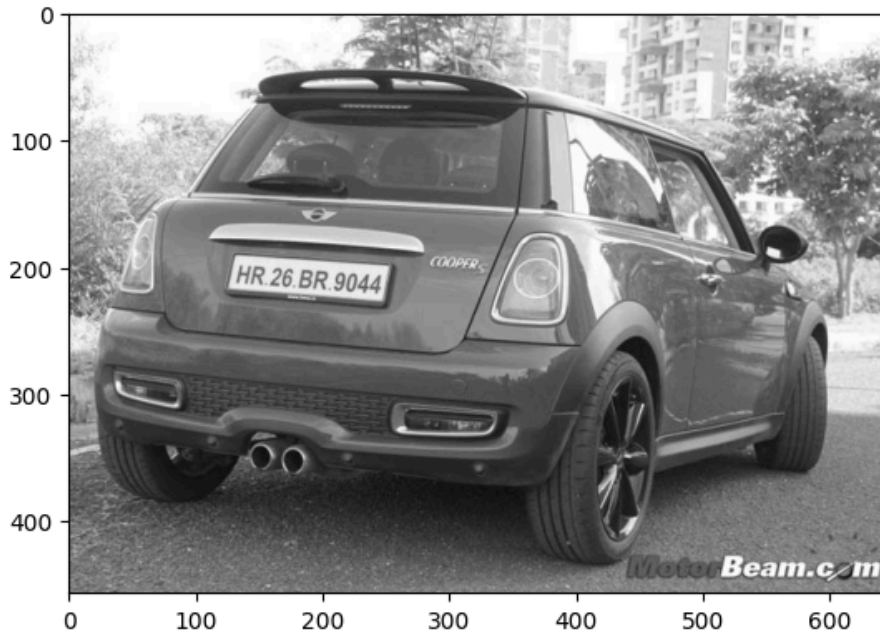```
In [12]: import cv2
          from matplotlib import pyplot as plt
          import numpy as np
          import imutils
          import easyocr
```

## 1. Read in Image, Grayscale

```
In [13]: img = cv2.imread('../data/image1.jpg')
          gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
          plt.imshow(cv2.cvtColor(gray,cv2.COLOR_BGR2RGB))
```
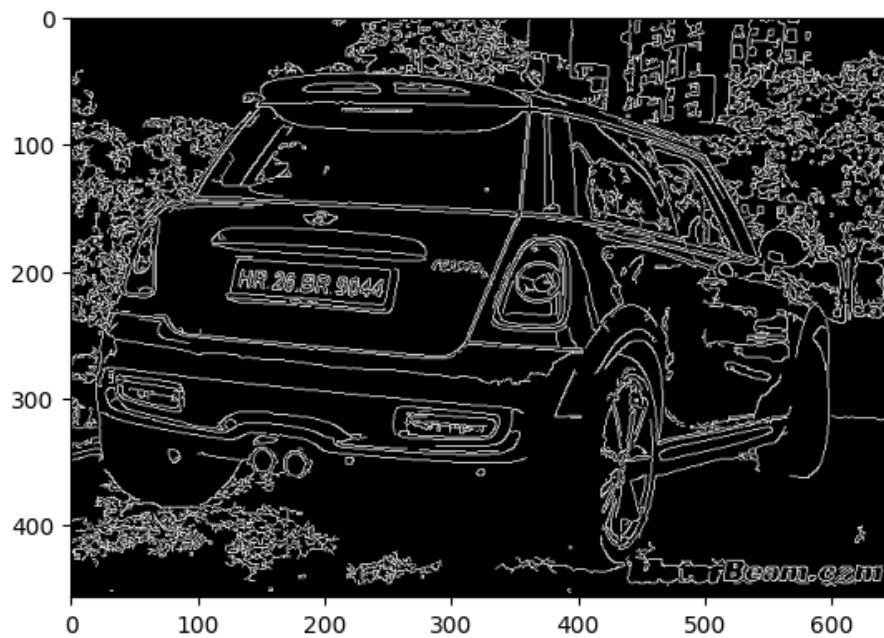
Out[13]: &lt;matplotlib.image.AxesImage at 0x29e316c60&gt;



## 2. Apply filter and find edges for localization

```
In [14]:  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[14]: <matplotlib.image.AxesImage at 0x2ac903cb0>

3. Find contours and apply masks

```
In [15]:  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 [16]:  location = None
          for contour in contours:
              approx = cv2.approxPolyDP(contour, 10, True)
              if len(approx) == 4:
                  location = approx
                  break
```
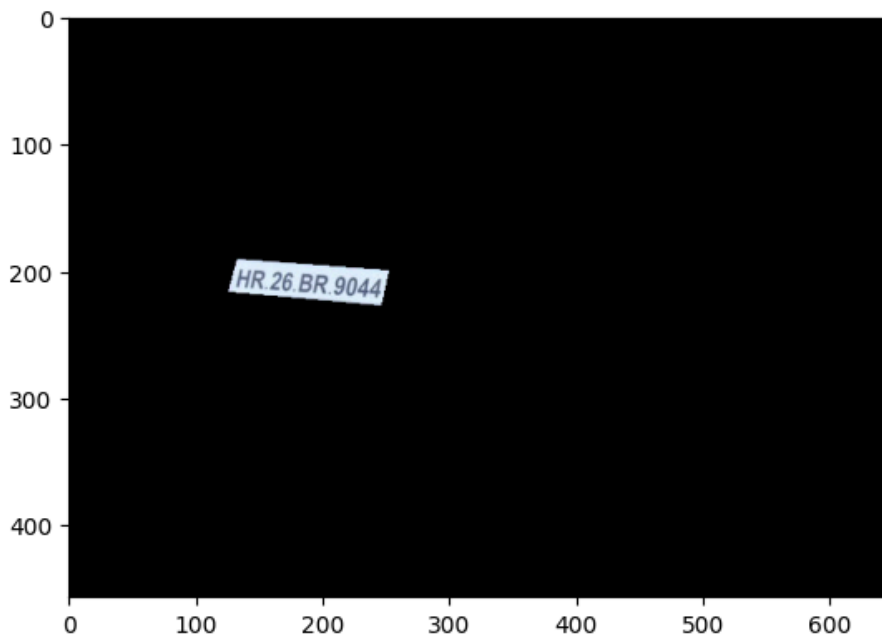
```
In [17]:  location
```

```
Out[17]:  array([[[126, 216]],

                 [[133, 191]],

                 [[252, 200]],

                 [[246, 227]]], dtype=int32)
```

```
In [18]:  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))
```
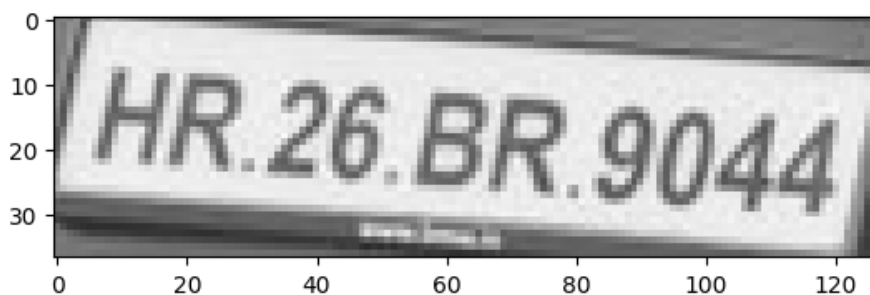
```
Out[18]:  <matplotlib.image.AxesImage at 0x2ac956330>
```

```
In [19]:   (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 [20]:   plt.imshow(cv2.cvtColor(cropped_image, cv2.COLOR_BGR2RGB))
```

Out[20]:   <matplotlib.image.AxesImage at 0x2ac95e330>

4. Use Easy OCR to Read Text

```
In [21]:   reader = easyocr.Reader(['en'])
           result = reader.readtext(cropped_image)
           result
```

Out[21]:   [([[np.int32(2), np.int32(4)],
            [np.int32(126), np.int32(4)],
            [np.int32(126), np.int32(34)],
            [np.int32(2), np.int32(34)]],
            'HR.26 BR.9044',
            np.float64(0.61014269013438))]

5. Render Result

```
In [22]:   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))
```

`<matplotlib.image.AxesImage at 0x2ac9f6420>`