```
!pip install easyocr
!pip install imutils
```

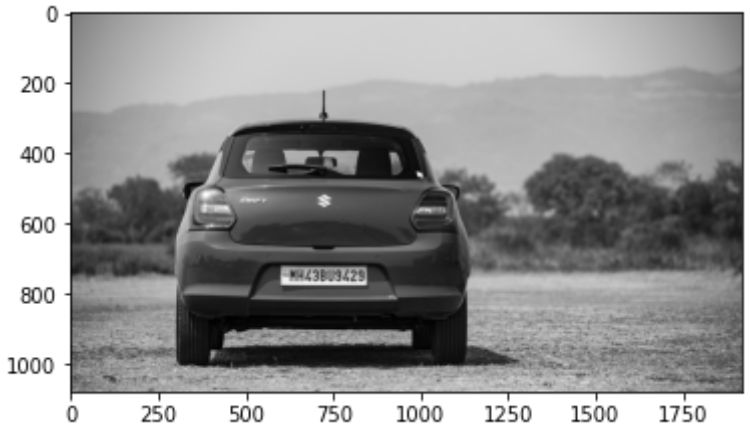
```
Collecting easyocr
  Downloading easyocr-1.4.2-py3-none-any.whl (70.8 MB)
     |████████████████████████████████| 70.8 MB 12 kB/s
Requirement already satisfied: torch in /usr/local/lib/python3.7/dist-packages (from easyocr) (1.10.0+cu111)
Requirement already satisfied: PyYAML in /usr/local/lib/python3.7/dist-packages (from easyocr) (3.13)
Requirement already satisfied: numpy in /usr/local/lib/python3.7/dist-packages (from easyocr) (1.21.6)
Requirement already satisfied: scikit-image in /usr/local/lib/python3.7/dist-packages (from easyocr) (0.18.3)
Requirement already satisfied: Pillow in /usr/local/lib/python3.7/dist-packages (from easyocr) (7.1.2)
Requirement already satisfied: scipy in /usr/local/lib/python3.7/dist-packages (from easyocr) (1.4.1)
Collecting python-bidi
  Downloading python_bidi-0.4.2-py2.py3-none-any.whl (30 kB)
Collecting opencv-python-headless<=4.5.4.60
  Downloading opencv_python_headless-4.5.4.60-cp37-cp37m-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (47.6 MB)
     |████████████████████████████████| 47.6 MB 1.3 MB/s
Requirement already satisfied: torchvision>=0.5 in /usr/local/lib/python3.7/dist-packages (from easyocr) (0.11.1+cu111)
Requirement already satisfied: typing-extensions in /usr/local/lib/python3.7/dist-packages (from torch->easyocr) (4.1.1)
Requirement already satisfied: six in /usr/local/lib/python3.7/dist-packages (from python-bidi->easyocr) (1.15.0)
Requirement already satisfied: matplotlib!=3.0.0,>=2.0.0 in /usr/local/lib/python3.7/dist-packages (from scikit-image->easyocr)
Requirement already satisfied: PyWavelets>=1.1.1 in /usr/local/lib/python3.7/dist-packages (from scikit-image->easyocr) (1.3.0)
Requirement already satisfied: networkx>=2.0 in /usr/local/lib/python3.7/dist-packages (from scikit-image->easyocr) (2.6.3)
Requirement already satisfied: imageio>=2.3.0 in /usr/local/lib/python3.7/dist-packages (from scikit-image->easyocr) (2.4.1)
Requirement already satisfied: tifffile>=2019.7.26 in /usr/local/lib/python3.7/dist-packages (from scikit-image->easyocr) (2021
Requirement already satisfied: python-dateutil>=2.1 in /usr/local/lib/python3.7/dist-packages (from matplotlib!=3.0.0,>=2.0.0->
Requirement already satisfied: cycler>=0.10 in /usr/local/lib/python3.7/dist-packages (from matplotlib!=3.0.0,>=2.0.0->scikit-i
Requirement already satisfied: kiwisolver>=1.0.1 in /usr/local/lib/python3.7/dist-packages (from matplotlib!=3.0.0,>=2.0.0->sci
Requirement already satisfied: pyparsing!=2.0.4,!=2.1.2,!=2.1.6,>=2.0.1 in /usr/local/lib/python3.7/dist-packages (from matplot
Installing collected packages: python-bidi, opencv-python-headless, easyocr
Successfully installed easyocr-1.4.2 opencv-python-headless-4.5.4.60 python-bidi-0.4.2
Requirement already satisfied: imutils in /usr/local/lib/python3.7/dist-packages (0.5.4)
```

```
import cv2
from matplotlib import pyplot as plt
import numpy as np
import imutils # convenience functions to make basic image processing functions such as translation, rotation, resizing, skeletonizat
```

```
import easyocr
 #allows us to read text from the image
```

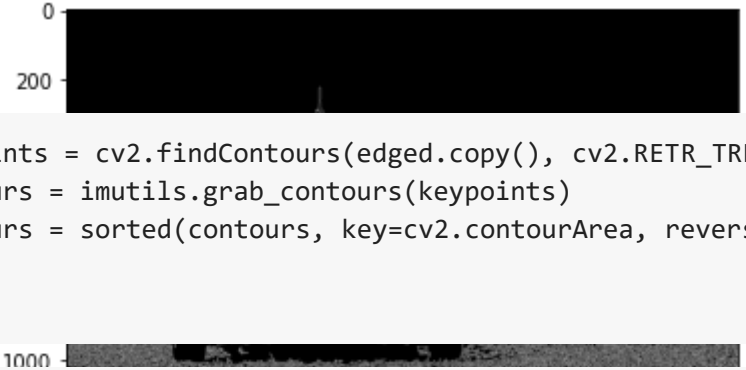
```
img = cv2.imread('swt.jpg')
gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
plt.imshow(cv2.cvtColor(gray, cv2.COLOR_BGR2RGB))
```

```
<matplotlib.image.AxesImage at 0x7f7f21fae450>
```




```
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))
```

```
<matplotlib.image.AxesImage at 0x7f7f22346510>
```




```
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]
```

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

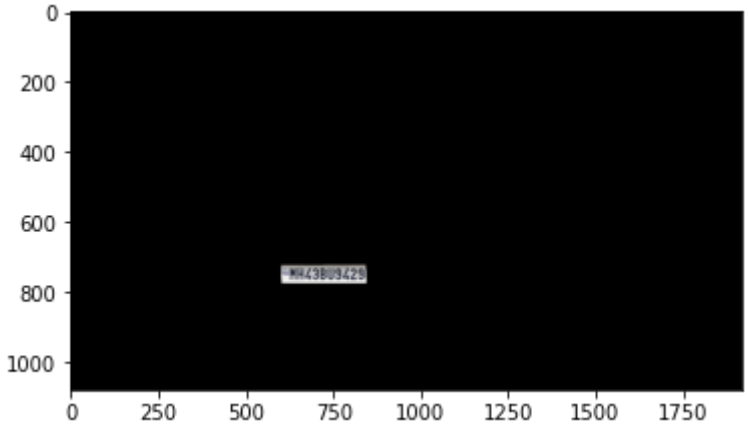
```
location
#the below mwntioned are the co-ordinartes of the number plate
# we are not present at the moment so we use masking for better identification
```

```
array([[[602, 724]],

       [[844, 723]],

       [[846, 775]],

       [[604, 775]]], dtype=int32)
```

```
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))
```

```
<matplotlib.image.AxesImage at 0x7f7f2205c850>
```




```
(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]
```

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

```
<matplotlib.image.AxesImage at 0x7f7f22043d90>
```


```
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.
[([[3, 19], [25, 19], [25, 33], [3, 33]], 'IND', 0.8171403317286938),
 ([[18, 0], [245, 0], [245, 50], [18, 50]], 'MHL3BU9L29', 0.8266663413296532)]
```

```
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
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 0x7f7f195856d0>
```