

Importing libraries Opencv,tesseract,mutils,matplotlib,numpy

```
import pytesseract
pytesseract.pytesseract.tesseract_cmd=r'C:\Program Files\Tesseract-OCR\tesseract.exe'

import easyocr
import cv2
import imutils
import matplotlib.pyplot as plt
import numpy as np
```

Taking an image and converting it to gray and bilateral filter

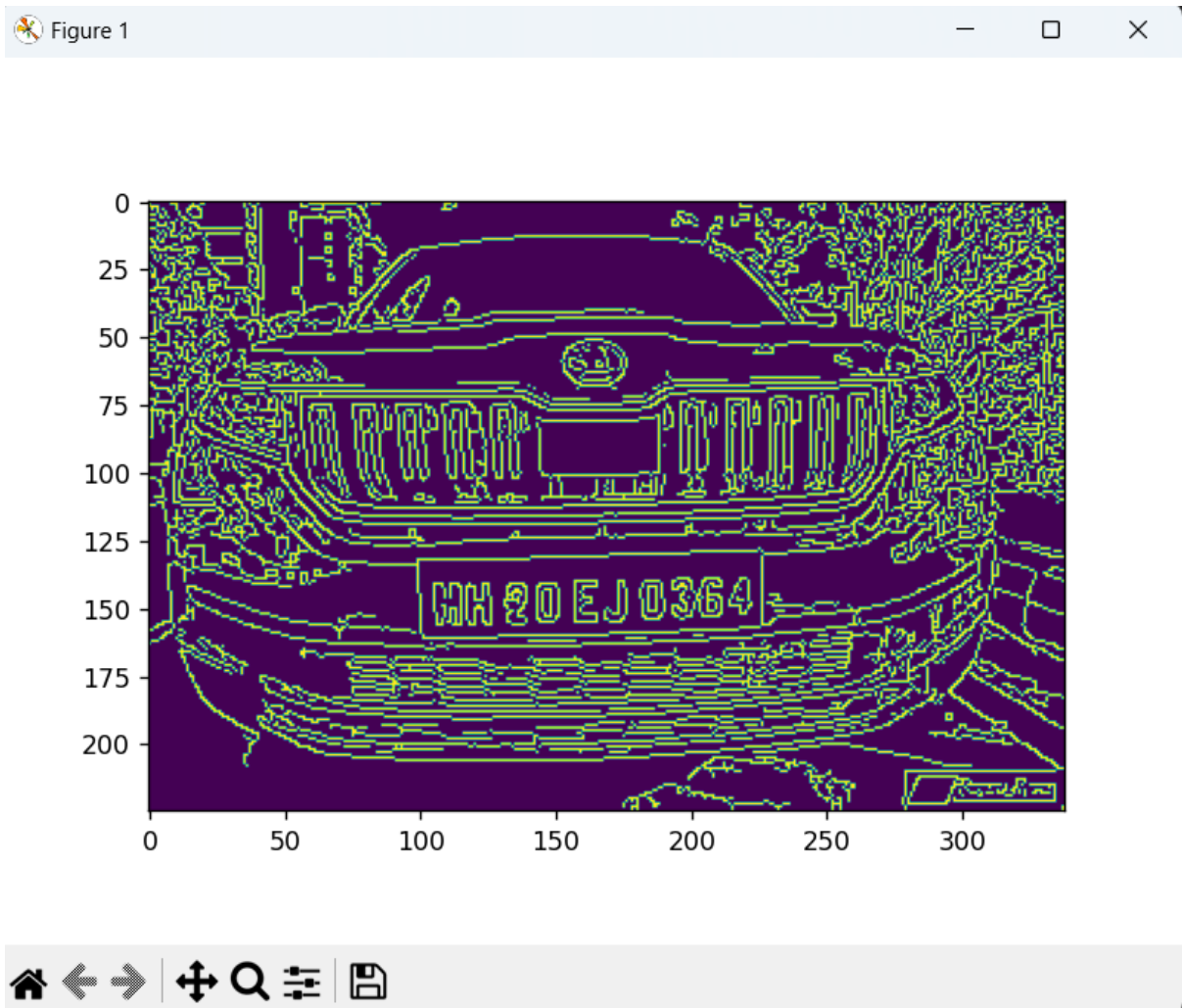
```
img = cv2.imread('2.png')
gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
plt.imshow(gray)
plt.show()

#Noise reduction using bilateralfilter
bf = cv2.bilateralFilter(gray, 11, 17, 17)
plt.imshow(bf)
plt.show()
```



Canny Edge detection

```
#Edge detection by canny  
edged = cv2.Canny(bf, 30, 200)  
plt.imshow(edged)  
plt.show()
```



finding contours and sorting it to among top12

```
#Now finding contours in the edged image
keypoints = cv2.findContours(edged.copy(), cv2.RETR_TREE, cv2.CHAIN_APPROX_SIMPLE)
contours = imutils.grab_contours(keypoints)

#sorting the contours and selecting only top 12
contours = sorted(contours, key=cv2.contourArea, reverse=True)[:12]

location = []
```

Finding the location points of the number plate and presenting via terminal

```
for i in range(5,12):
    if len(location) != 4 :
        for contour in contours:
            points = cv2.approxPolyDP(contour, i, True)
            if len(points) == 4:
                if points[3][0]-points[0][0]>=30 or points[3][0]-points[1][0]<=-30 or points[3][0]-points[1][0]>=30:
                    location = points
                    break
print("Points location = ")
print(location)
```

```
Points location =
[[[227 130]]

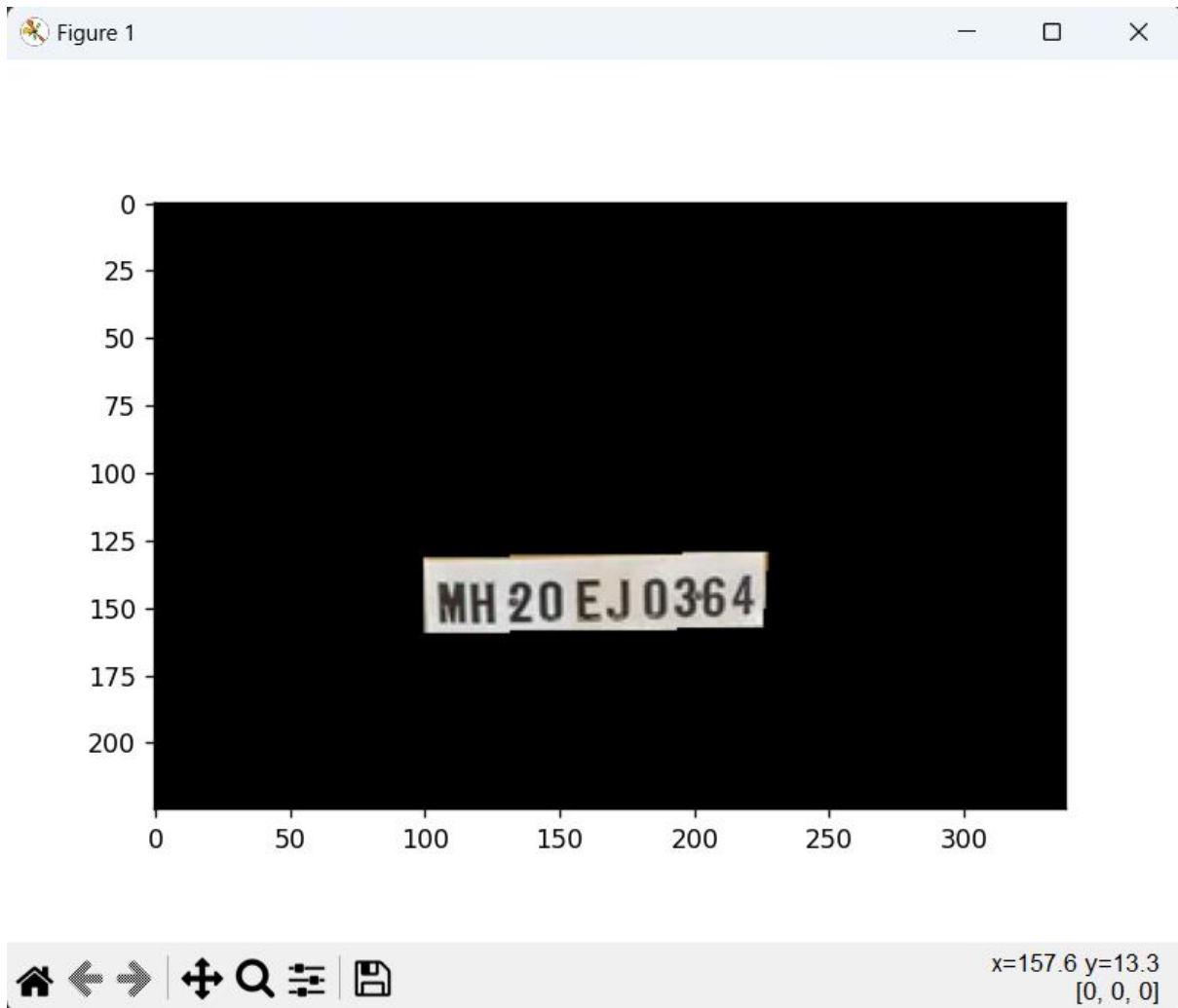
[[225 157]]

[[100 159]]

[[100 132]]]
```

Setting the number plate in a dark mask background only vision of the number plate area

```
mask = np.zeros(gray.shape, np.uint8)
numpla_im = cv2.drawContours(mask, [location], 0, 255, -1)
numpla_im = cv2.bitwise_and(img, img, mask=mask)
plt.imshow(numpla_im)
plt.show()
```



Using tesseract and easyocr to read the number plate in the last image

```
t=pytesseract.image_to_string(numpla_im)

#using Easyocr to read the test in the image
result = easyocr.Reader(['en'],gpu=False).readtext(numpla_im)

#declaring the test if pytesseract doesn't read anything it will get from using opencv
if t=='':
    for i in range(0,len(result)):
        t = t+ result[i][-2]

#if there is nothing detected from the image
if t=='':
    print("couldn't identify/read the number plate")
    exit()
else:
    print("The aquired number plate is "+ t)
```

Declaring the font size, marking a rectangle around the number plate on the final image as the aquired data written on the image

```
final_image = cv2.putText(img, text=t, org=(points[3][0][0]-10, points[1][0][1]+60), fontFace=cv2.FONT_HERSHEY_DUPLEX, fontScale=1,
                           color=(205,25,20), thickness=2, lineType=cv2.LINE_AA)
final_image = cv2.rectangle(img, tuple(points[0][0]), tuple(points[2][0]), (255,0,255),2)

#The final result
plt.imshow(cv2.cvtColor(final_image, cv2.COLOR_BGR2RGB))
plt.show()
```

