

*imutils*====>Çevirme, döndürme, yeniden boyutlandırma ve Matplotlib görüntülerini görüntüleme gibi temel görüntü işleme işlevlerini OpenCV ve Python ile kolaylaştıran bir dizi kolaylık işlevi

*NumPy*====> çok sayıda çeşitli matematiksel işlem içerir.

```
import cv2
import imutils
import numpy as np
```

*Matplotlib*====> Python'da bir kütüphanedir ve NumPy kütüphanesinin sayısal – matematiksel uzantısıdır. Pyplot, MATLAB benzeri bir arabirim sağlayan bir Matplotlib modülüne yönelik durum tabanlı bir arabirimdir. Pyplot'ta kullanılabilecek çeşitli grafikler vardır: Çizgi Plot, Kontur, Histogram, Dağılım, 3D Plot, vb.

```
from matplotlib import pyplot as plt
```

```
!pip install easyocr
```

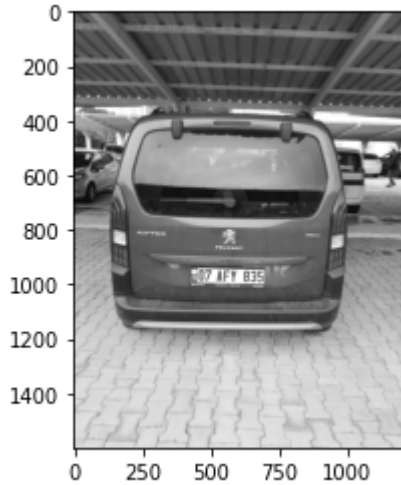
```
Requirement already satisfied: easyocr in /usr/local/lib/python3.7/dist-packages (1.4.1)
Requirement already satisfied: numpy in /usr/local/lib/python3.7/dist-packages (from ea
Requirement already satisfied: opencv-python-headless in /usr/local/lib/python3.7/dist-
Requirement already satisfied: scikit-image in /usr/local/lib/python3.7/dist-packages (
Requirement already satisfied: torch in /usr/local/lib/python3.7/dist-packages (from ea
Requirement already satisfied: scipy in /usr/local/lib/python3.7/dist-packages (from ea
Requirement already satisfied: python-bidi in /usr/local/lib/python3.7/dist-packages (f
Requirement already satisfied: PyYAML in /usr/local/lib/python3.7/dist-packages (from e
Requirement already satisfied: torchvision>=0.5 in /usr/local/lib/python3.7/dist-packag
Requirement already satisfied: Pillow<8.3.0 in /usr/local/lib/python3.7/dist-packages (
Requirement already satisfied: typing-extensions in /usr/local/lib/python3.7/dist-packa
Requirement already satisfied: six in /usr/local/lib/python3.7/dist-packages (from pyth
Requirement already satisfied: networkx>=2.0 in /usr/local/lib/python3.7/dist-packages
Requirement already satisfied: PyWavelets>=1.1.1 in /usr/local/lib/python3.7/dist-packa
Requirement already satisfied: tifffile>=2019.7.26 in /usr/local/lib/python3.7/dist-pac
Requirement already satisfied: imageio>=2.3.0 in /usr/local/lib/python3.7/dist-packages
Requirement already satisfied: matplotlib!=3.0.0,>=2.0.0 in /usr/local/lib/python3.7/di
Requirement already satisfied: python-dateutil>=2.1 in /usr/local/lib/python3.7/dist-pa
Requirement already satisfied: pyparsing!=2.0.4,!>=2.1.2,!>=2.1.6,>=2.0.1 in /usr/local/l
Requirement already satisfied: kiwisolver>=1.0.1 in /usr/local/lib/python3.7/dist-packa
Requirement already satisfied: cycycler>=0.10 in /usr/local/lib/python3.7/dist-packages (
```

*EasyOCR (Optical Character Recognition)*=====> s'nin Optik Karakter Tanıma işlemini gerçekleştirmesini sağlayan bir Python paketidir.

```
import easyocr
```

```
img = cv2.imread('/content/y.h.jpg')#read the image
gri = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)#yeniden reklendirme (cv2) blue green red tanir
plt.imshow(cv2.cvtColor(gri, cv2.COLOR_BGR2RGB))#img bgr de o yuzden daha net olabilmesi icir
```

<matplotlib.image.AxesImage at 0x7f44fde2e610>



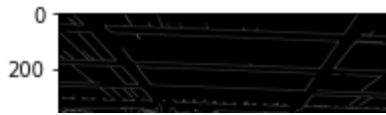
(*Bilateral Filter*)=====> görüntüler için nonlinear, kenar koruyucu ve gürültü azaltıcı bir yumuşatma filtresidir. Her pikselin yoğunluğunu, yakındaki piksellerden gelen ağırlıklı ortalama yoğunluk değerleriyle değiştirir. Bu keskin kenarları korur.

Düzenlemek için çift tıklayın (veya enter tuşuna basın)

*Canny*=====> kenar detektörü, görüntülerdeki çok çeşitli kenarları algılamak için çok aşamalı bir algoritma kullanan bir kenar algılama operatörüdür.

```
bilfilter = cv2.bilateralFilter(gri, 11, 17, 17) #Noise reduction
edged = cv2.Canny(bilfilter, 30, 200) #Edge detection
plt.imshow(cv2.cvtColor(edged, cv2.COLOR_BGR2RGB))
```

```
<matplotlib.image.AxesImage at 0x7f44fd148250>
```



Düzenlemek için çift tıklayın (veya enter tuşuna basın)



```
key_nokta = cv2.findContours(edged, cv2.RETR_TREE, cv2.CHAIN_APPROX_SIMPLE)#sonuclar nasıl ol
contours = imutils.grab_contours(key_nokta)#konturlarimizin gercekte nasıl dondurulecegini b
contours = sorted(contours, key=cv2.contourArea, reverse=True)[:10]#siralama,/nasıl sıralamak
```

```
1400 -
```

conturlarin herbiri arasinda dongu yapmak ve herhangi bir kare mi yoksa plakayi mi temsil ediyor kontrol ediyoruz 4 kilit nokta olup olmadigina kontrol ediyoruz

```
location = None
for contour in contours:
    approx = cv2.approxPolyDP(contour, 10, True)#enough to detect a square but it may change
    if len(approx) == 4: # 4 tane kilit nokta varsa muhtemelen aradigimiz sey budur
        location = approx
        break
```

location

```
array([[453, 944]],
      [[452, 993]],
      [[696, 998]],
      [[696, 947]]], dtype=int32)
```

```
mask = np.zeros(gri.shape, np.uint8)#black mask same shape as our original image/
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 0x7f44fdd94890>
```

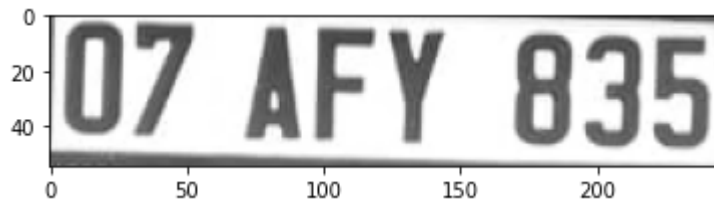


```
(x,y) = np.where(mask==255)
(x1, y1) = (np.min(x), np.min(y))
(x2, y2) = (np.max(x), np.max(y))
cropped_image = gri[x1:x2+1, y1:y2+1]
```



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

```
<matplotlib.image.AxesImage at 0x7f44fdd6ed50>
```

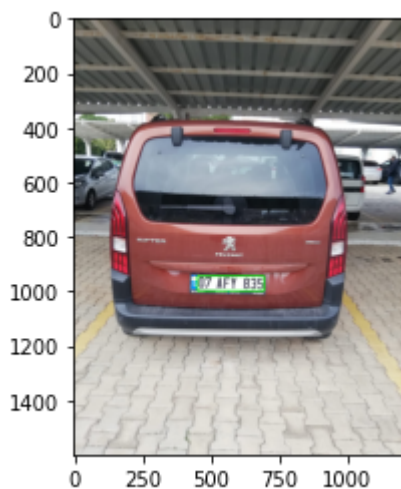


```
reader = easyocr.Reader(['en'])
result = reader.readtext(cropped_image)
result
```

```
CUDA not available - defaulting to CPU. Note: This module is much faster with a GPU.
[[[[0, 0], [245, 0], [245, 55], [0, 55]], '07 AFY 035', 0.9005792227168155]]
```

```
text = result[0][-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 0x7f44fc200950>
```



```
print('Plate number is || ',text,' ||')
```

```
Plate number is || 07 AFY 035 ||
```

✓ 0 sn. tamamlanma zamanı: 17:18



reCAPTCHA hizmetiyle bağlantı kurulamadı. Lütfen internet bağlantınızı kontrol edin ve reCAPTCHA testi almak için sayfayı yeniden yükleyin.