

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

# Resmi BGR okuma
bgr = cv2.imread("para.jpg")

# Resmi BGR'den RGB'ye çevir ve ekranda göster imshow ile
rgb = cv2.cvtColor(bgr, cv2.COLOR_BGR2RGB)
plt.imshow(rgb)
# plt.axis("off") # bunu isteğe bağlı olarak kullanabilirsin eksenleri kapatır
```

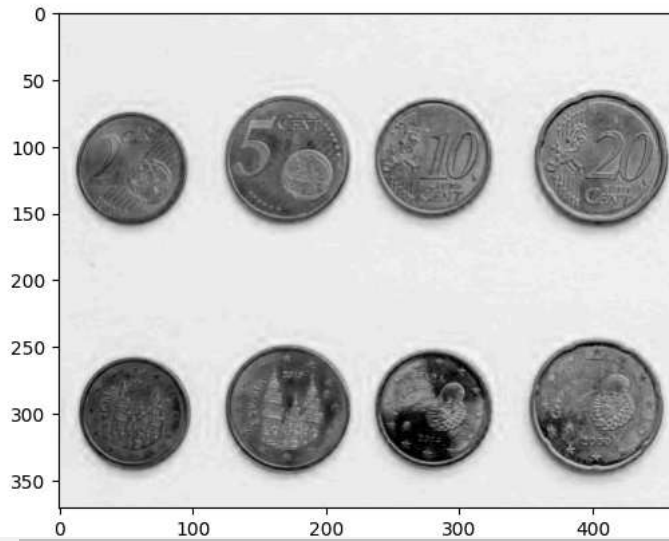
 <matplotlib.image.AxesImage at 0x7b591a305ed0>



```
# Gri tonlama resmi gri tona çevirdik ve ekranda gösterdik
r, g, b = cv2.split(rgb)
plt.imshow(r, cmap="gray")
plt.show()

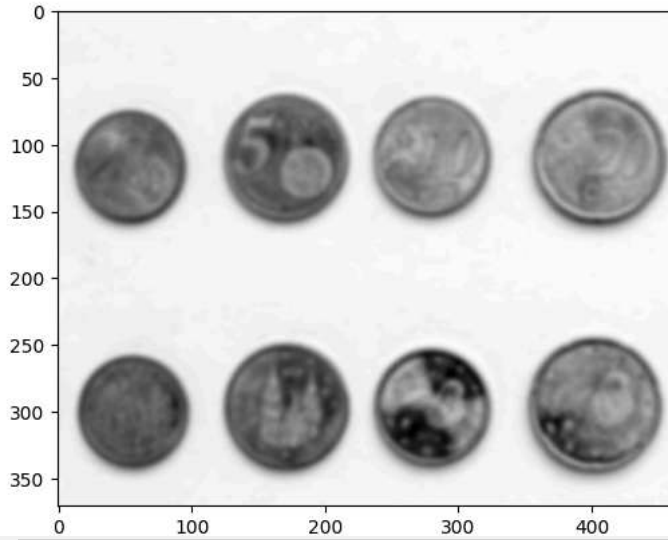
plt.imshow(g, cmap="gray")
plt.show()

plt.imshow(b, cmap="gray")
plt.show()
```



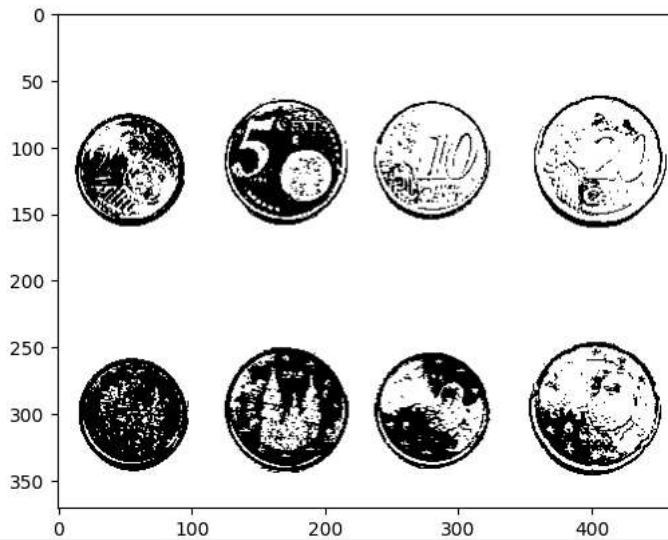
```
#Gri tonlama ile Resmi bulanıklaştırma Gauss ile
gri_para = cv2.cvtColor(rgb, cv2.COLOR_RGB2GRAY)
bulanik_para55 = cv2.GaussianBlur(gri_para,(11,11),0)
plt.imshow(bulanik_para55, cmap="gray")
```

 <matplotlib.image.AxesImage at 0x7b591a7e1ad0>



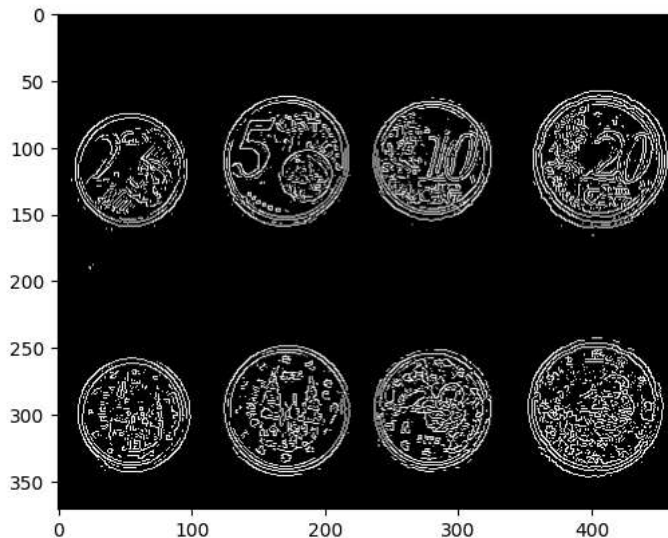
```
#Eşikleme Uygulaması (Threshold)
_, binary_para = cv2.threshold(gri_para, 155, 155, cv2.THRESH_BINARY)
plt.imshow(binary_para, cmap="gray")
```

 <matplotlib.image.AxesImage at 0x7b591a42ded0>



```
# Canny ile kenar bulma ve ekranda gösterme
kenar_para55 = cv2.Canny(gri_para, 95,95)
plt.imshow(kenar_para55, cmap="gray")
```

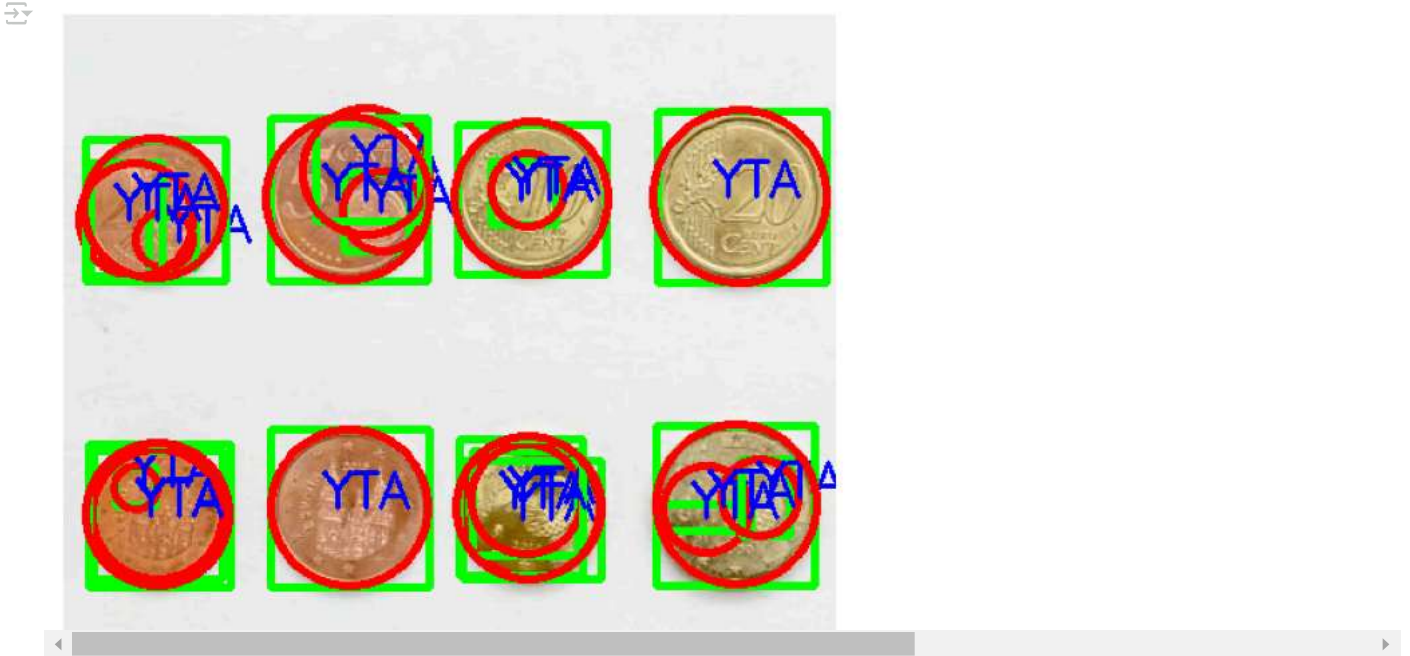
 <matplotlib.image.AxesImage at 0x7b591a365550>



```
# Kontur tespiti
contours, _ = cv2.findContours(kenar_para55, cv2.RETR_EXTERNAL, cv2.CHAIN_APPROX_SIMPLE)
kontorlu_para = rgb.copy()

# Kontur işlemleri
for kontor in contours:
    if cv2.contourArea(kontor) > 65: # Daha küçük nesneleri de kapsayacak şekilde eşik düşürüldü
        # Bounding box hesaplama
        x, y, w, h = cv2.boundingRect(kontor)
        cv2.rectangle(kontorlu_para, (x, y), (x + w, y + h), (0, 255, 0), 3) # Yeşil bounding box
        # Kırmızı daire çizme
        (cx, cy), radius = cv2.minEnclosingCircle(kontor)
        center = (int(cx), int(cy))
        radius = int(radius)
        cv2.circle(kontorlu_para, center, radius, (255, 0, 0), 3) # Kırmızı daire
        # "YTA" metni yazma
        text_x, text_y = x + w // 3, y + h // 2
        cv2.putText(kontorlu_para, "YTA", (text_x, text_y), cv2.FONT_HERSHEY_SIMPLEX, 1, (0, 0, 255), 2) # Mavi baş harfler

# Sonucu gösterme
plt.figure(figsize=(10, 6))
plt.imshow(kontorlu_para)
plt.axis("off")
plt.show()
```



```
##### Bonus

# Kontur tespiti
contours, _ = cv2.findContours(kenar_para55, cv2.RETR_EXTERNAL, cv2.CHAIN_APPROX_SIMPLE)
kontorlu_para = rgb.copy()

# Konturları alanlarına göre sırala (büyükten küçüğe)
contours = sorted(contours, key=cv2.contourArea, reverse=True)

# En büyük ve en küçük paraları bulma
en_buyuk_kontor = contours[0]
en_kucuk_kontor = contours[-1]

# En büyük paranın içine 8 yazma
x, y, w, h = cv2.boundingRect(en_buyuk_kontor)
text_x, text_y = x + w // 2, y + h // 2
cv2.putText(kontorlu_para, "8", (text_x, text_y), cv2.FONT_HERSHEY_SIMPLEX, 1, (255, 0, 0), 2)
```

```
# En küçük paranın içine 1 yazma
x, y, w, h = cv2.boundingRect(en_kucuk_kontor)
text_x, text_y = x + w // 2, y + h // 2
cv2.putText(kontorlu_para, "1", (text_x, text_y), cv2.FONT_HERSHEY_SIMPLEX, 1, (255, 0, 0), 2)

# Sonucu gösterme
plt.figure(figsize=(10, 6))
plt.imshow(kontorlu_para)
plt.axis("off")
plt.show()
```



Bonus

```
# Kontur tespiti
contours, _ = cv2.findContours(kenar_para55, cv2.RETR_EXTERNAL, cv2.CHAIN_APPROX_SIMPLE)
kontorlu_para = rgb.copy()

# Konturları alanlarına göre sırala (büyükten küçüğe)
contours = sorted(contours, key=cv2.contourArea, reverse=True)

# En büyük ve en küçük paraları bulma
en_buyuk_kontor = contours[0]
en_kucuk_kontor = contours[-1]

# En büyük paranın içine 8 yazma
x, y, w, h = cv2.boundingRect(en_buyuk_kontor)
text_x, text_y = x + w // 2, y + h // 2
cv2.putText(kontorlu_para, "8", (text_x, text_y), cv2.FONT_HERSHEY_SIMPLEX, 1, (255, 0, 0), 2)

# En küçük paranın içine 1 yazma
x, y, w, h = cv2.boundingRect(en_kucuk_kontor)
text_x, text_y = x + w // 2, y + h // 2
```

```
cv2.putText(kontorlu_para, "1", (text_x, text_y), cv2.FONT_HERSHEY_SIMPLEX, 1, (255, 0, 0), 2)
```

```
# Sonucu gösterme  
plt.figure(figsize=(10, 6))  
plt.imshow(kontorlu_para)  
plt.axis("off")  
plt.show()
```

