**Giriş Görüntüleri:**

**1.Eğitim Görüntüleri**



*Train1*

*Train2*





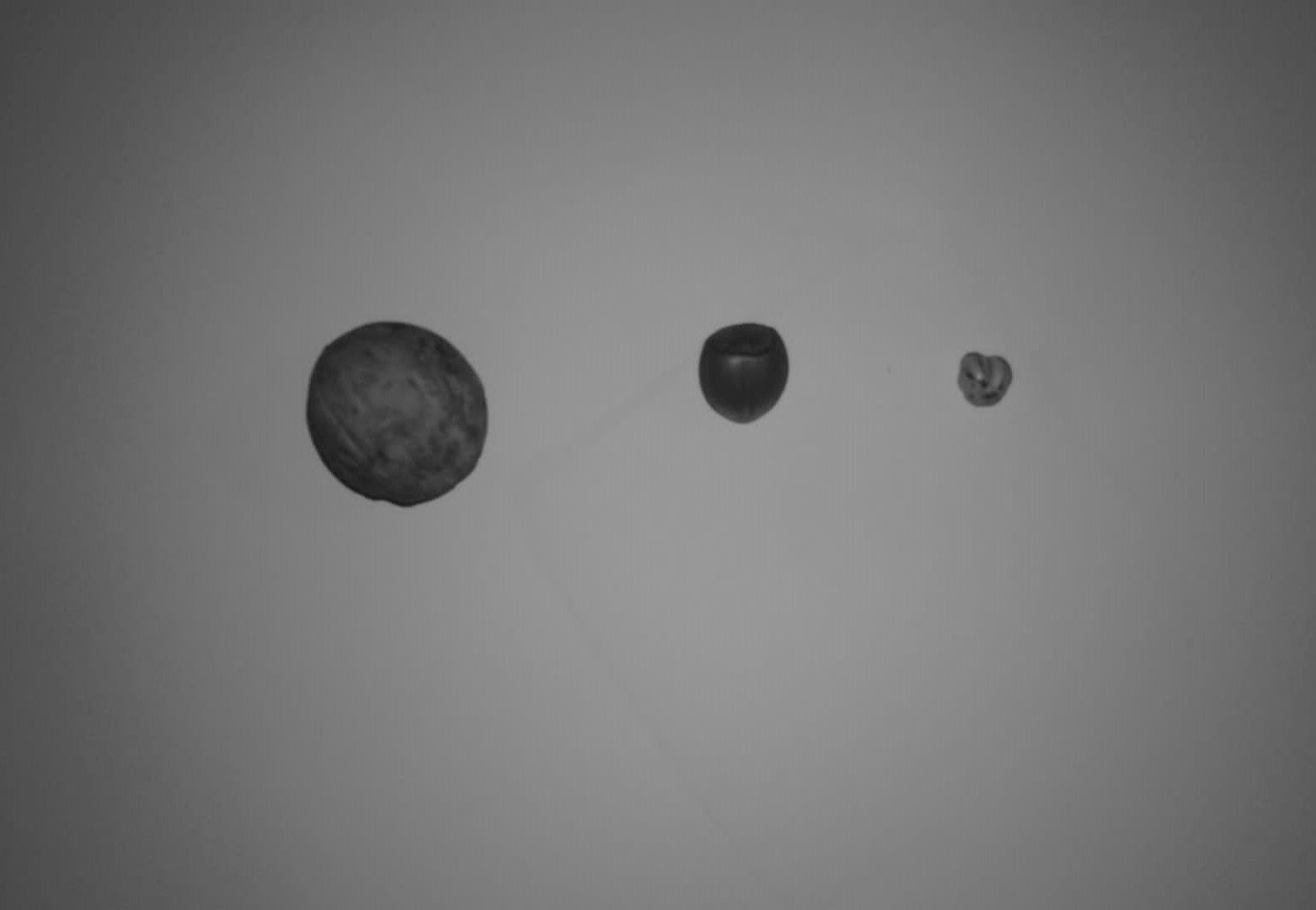
*Train3*

*Train4*



*Train5*

**2. Test Görüntüleri**



*Test2*



*Test6*

*Test5*

*Test4*

*Test3*

*Test1*

**3.Kodlar:**

*## Kütüphanelerin eklenmesi*

**import** numpy **as** np

**import** matplotlib.pyplot **as** plt

**import** cv2 **as** cv

**def** gammaCorrection(src, gamma): *## Opencv kütüphanesinde bulunan Gama Fonksiyonu kullanılmıştır.*

*## Giriş olarak kaynak görüntüsü ve gama değeri verilmiştir.*

invGamma **=** 1 **/** gamma

table **=** [((i **/** 255) **\*\*** invGamma) **\*** 255 **for** i **in** range(256)]

table **=** np**.**array(table, np**.**uint8)

**return** cv**.**LUT(src, table)

**def** Sobel\_Keskinlestirme(img): *## Sobel filtreleri kullanarak edge bulma işlemi gerçekleştirilmştir.*

*## hem x hem y yönünde sobel filtresi uygulanır ve toplanır, elde edilen sonuç return edilir.*

*## x yönünde sobel*

filter**=** np**.**array([[**-**1,**-**2,**-**1],[0,0,0],[1,2,1]])

dst**=**cv**.**filter2D(img,**-**1,filter)

absobelx**=**abs(dst)

*## y yönünde sobel*

filter**=** np**.**array([[**-**1,0,1],[**-**2,0,2],[**-**1,0,1]])

dsty**=**cv**.**filter2D(img,**-**1,filter)

abssobely**=**abs(dsty)

*## x\_yönü + y\_yönü*

sobel**=**absobelx**+**abssobely

**return** sobel

**def** Fonksiyon (isim):

font1**=** cv**.**FONT\_HERSHEY\_COMPLEX *#opencv fonts*

original\_image**=**cv**.**imread(isim,0) *## gray scale olarak okundu*

*## Görüntüde bulunan kenar noktaları kaybetmeden gürültüleri azaltmak için*

*## bilateral filtresi kullanılmıştır*

blur **=** cv**.**bilateralFilter(original\_image,35,105,105)

*## Adaptive threshold kullanılarak görüntüde sadece tespit edilmek istenen objenin bırakılması amaçlanmıştır.*

*## Adaptive threshold kullanılarak, yerel eşik değerleri ile arka planın daha iyi kaybolduğu gözlemlenmiştir.*

th3 **=** cv**.**adaptiveThreshold(blur,255,cv**.**ADAPTIVE\_THRESH\_GAUSSIAN\_C,\

cv**.**THRESH\_BINARY,19,1)

*## Eşikleme işleminen sonra arka plan tamamiyle yok edilememiştir. Görüntüde hem arka planı yok etmek*

*## hem de gürültüleri azaltmak için önce opening sonra closing işlemi yapılmıştır.*

*## yapılandırma elemanı, tespit edilmek istenen nesneler daire olduğu için elips olarak seçilmiştir.*

kernel **=** cv**.**getStructuringElement(cv**.**MORPH\_ELLIPSE,(3,3))

opened\_img **=** cv**.**morphologyEx(th3, cv**.**MORPH\_OPEN , kernel)

closed\_img **=** cv**.**morphologyEx(opened\_img, cv**.**MORPH\_CLOSE, kernel)

*## Arka plan atıldıktan sonra kalan gürültüleri yok etmek için medyan filtresi kullanılmıştır.*

median **=** cv**.**medianBlur(closed\_img,11)

*## Arka planı atılmış gürültüleri azaltılmış görüntüde yuvarlak nesnelerin tespiti için HoughCircles fonksiyonu kullanılmıştır.*

detected\_circles **=** cv**.**HoughCircles(median,

cv**.**HOUGH\_GRADIENT, 1, median**.**shape[0]**/**7, param1 **=** 230,

param2 **=** 13, minRadius **=** 3, maxRadius **=** 165) *##*

*## görüntüde bulunan yuvarlak nesnelerin yarıçap ve koordinat bilgilerinin depolanması için listeler oluşturulmuştur.*

a\_list**=**[]

b\_list**=**[]

r\_list**=**[]

*############################# Bulunan nesnelerin üzerine yazılacak yazının ayarlamaları yapılmıştır.*

font **=** cv**.**FONT\_HERSHEY\_SIMPLEX

*# Yazı tipi ölçeği*

fontScale **=** 1

*# Yazının rengi*

color **=** (200)

*# Yazının Kalınlığı*

thickness **=** 5

*###################################*

**if** detected\_circles **is** **not** **None**: *## eğer görüntü üzerinde daire tespiti yapıldıysa for un içine girer.*

detected\_circles **=** np**.**uint16(np**.**around(detected\_circles))

**for** pt **in** detected\_circles[0, :]: *## dairelerin yarıçap ve merkez koordinat bilgileri depolanır.*

a, b, r **=** pt[0], pt[1], pt[2]

a\_list**.**append(a)

b\_list**.**append(b)

r\_list**.**append(r)

*# Draw the circumference of the circle.*

**for** k **in** range(len(a\_list)): *## merkez koordinat bilgileri int türüne cast edilir.*

a\_list[k]**=**int(a\_list[k])

b\_list[k]**=**int(b\_list[k])

*## Nesnelerin sayılarının tesbiti için sayaçlar oluşturulmuştur.*

ceviz**=**0

fındık**=**0

leblebi**=**0

*## raporda da belirtildiği üzere; öznitelik olarak yarıçap seçilmiştir.Öncelikle ceviz nesnesinin yarıçapının uzunluğuna*

*##göre görüntünün çekim mesafesi belirlenmiştir.*

*## 3 farklı kategoride çekim mesafesi mevcuttur: Normal,Yakın ve Uzak. Bu kategorilere göre*

*## else if merdiveni içerisine yazılan ifadelerde ki referans yarıçaplarının değerleri değişmektedir.*

**if**(max(r\_list)**>**60 **and** max(r\_list)**<**140): *## normal cekim*

print("Normal Cekim")

**for** k **in** range(len(r\_list)):

**if** (r\_list[k]**>=**60 ):*## Ceviz*

cv**.**putText(original\_image,"Ceviz",(a\_list[k],b\_list[k]),

font, fontScale, color, thickness, cv**.**LINE\_AA)

cv**.**circle(original\_image, (a\_list[k],b\_list[k]), r\_list[k], (0, 255, 0), 10) *## Orijinal Görüntü üzerine çizim*

ceviz**=**ceviz**+**1

**elif** (r\_list[k]**>=**33 **and** r\_list[k]**<**60): *## Fındık*

cv**.**putText(original\_image,"Findik",(a\_list[k],b\_list[k]),

font, fontScale, color, thickness, cv**.**LINE\_AA)

cv**.**circle(original\_image,(a\_list[k],b\_list[k]), r\_list[k], (0, 255, 0), 10) *## Orijinal Görüntü üzerine çizim*

fındık**=**fındık**+**1

**elif** (r\_list[k]**>=** 18 **and** r\_list[k]**<**33): *##Leblebi*

cv**.**putText(original\_image,"Leblebi",(a\_list[k],b\_list[k]),

font, fontScale, color, thickness, cv**.**LINE\_AA)

cv**.**circle(original\_image, (a\_list[k],b\_list[k]), r\_list[k], (0, 255, 0), 10)

leblebi**=**leblebi**+**1

**elif**(max(r\_list)**<=** 55): *## Görüntünün uzaktan çekilmesi*

print("Uzaktan Çekim")

**for** k **in** range(len(r\_list)):

**if** (r\_list[k]**>**45 **and** r\_list[k]**<=**55 ):

cv**.**putText(original\_image,"Ceviz",(a\_list[k],b\_list[k]),

font, fontScale, color, thickness, cv**.**LINE\_AA)

cv**.**circle(original\_image, (a\_list[k],b\_list[k]), r\_list[k], (0, 255, 0), 10) *## Orijinal Görüntü üzerine çizim*

ceviz**=**ceviz**+**1

**elif** (r\_list[k]**>=**40 **and** r\_list[k]**<**45):

cv**.**putText(original\_image,"Findik",(a\_list[k],b\_list[k]),

font, fontScale, color, thickness, cv**.**LINE\_AA)

cv**.**circle(original\_image,(a\_list[k],b\_list[k]), r\_list[k], (0, 255, 0), 10) *## Orijinal Görüntü üzerine çizim*

fındık**=**fındık**+**1

**elif** ( r\_list[k]**<**40):

cv**.**putText(original\_image,"Leblebi",(a\_list[k],b\_list[k]),

font, fontScale, color, thickness, cv**.**LINE\_AA)

cv**.**circle(original\_image, (a\_list[k],b\_list[k]), r\_list[k], (0, 255, 0), 10) *## Orijinal Görüntü üzerine çizim*

leblebi**=**leblebi**+**1

**elif**(max(r\_list)**>**140): *## Görüntünün yakından çekilmesi*

print("Yakından çekim")

**for** k **in** range(len(r\_list)):

**if** (r\_list[k]**>**140 ):

cv**.**putText(original\_image,"Ceviz",(a\_list[k],b\_list[k]),

font, fontScale, color, thickness, cv**.**LINE\_AA)

cv**.**circle(original\_image, (a\_list[k],b\_list[k]), r\_list[k], (0, 255, 0), 10) *## Orijinal Görüntü üzerine çizim*

ceviz**=**ceviz**+**1

**elif** (r\_list[k]**>=**100 **and** r\_list[k]**<**140):

cv**.**putText(original\_image,"Findik",(a\_list[k],b\_list[k]),

font, fontScale, color, thickness, cv**.**LINE\_AA)

cv**.**circle(original\_image,(a\_list[k],b\_list[k]), r\_list[k], (0, 255, 0), 10) *## Orijinal Görüntü üzerine çizim*

fındık**=**fındık**+**1

**elif** ( r\_list[k]**<**100):

cv**.**putText(original\_image,"Leblebi",(a\_list[k],b\_list[k]),

font, fontScale, color, thickness, cv**.**LINE\_AA)

cv**.**circle(original\_image, (a\_list[k],b\_list[k]), r\_list[k], (0, 255, 0), 10) *## Orijinal Görüntü üzerine çizim*

leblebi**=**leblebi**+**1

isim\_split**=**isim**.**split(".")

fig, ax **=** plt**.**subplots(2,3, figsize **=** (25,15))

fig**.**suptitle('ALL OUTPUTS FOR PROJECT', fontsize**=**20)

*# Line plots*

*## Gri görüntünün bastırılması*

ax[0,0]**.**set\_title('GRAY SCALE')

ax[0,0]**.**imshow(original\_image, cmap **=** "gray")

*## Bilteral Filtresi uygulanmış görüntünün bastırılması*

ax[0,1]**.**set\_title(isim\_split[0]**+**'\_BilateralFilter')

ax[0,1]**.**imshow(blur, cmap **=** "gray")

*## Adaptive Eşikleme işlemi yapılmış görüntünün Bastırılması*

ax[0,2]**.**set\_title(isim\_split[0]**+**'\_Threshold')

ax[0,2]**.**imshow(th3, cmap **=** "gray")

*## Opening işlemi uygulanmış görüntünün bastırılması*

ax[1,0]**.**set\_title(isim\_split[0]**+**'\_Opening')

ax[1,0]**.**imshow(opened\_img, cmap **=** "gray")

*## Closing işlemi uygulanmış görüntünün bastırılması*

ax[1,1]**.**set\_title(isim\_split[0]**+**'\_Closing')

ax[1,1]**.**imshow(closed\_img, cmap **=** "gray")

*## Medyan filtresi uygulanmış görüntünün bastırılması.*

ax[1,2]**.**set\_title(isim\_split[0]**+**'\_Median')

ax[1,2]**.**imshow(median, cmap **=** "gray")

*## Başlıklar arasındaki boşlukların ayarlanması*

plt**.**tight\_layout()

plt**.**show()

*## Tespit edilen nesne sayılarının döndürülmesi*

**return** ceviz,fındık,leblebi

**for** k **in** range(len(r\_list)):

**if** (r\_list[k]**>=**60 ):*## Ceviz*

cv**.**putText(original\_image,"Ceviz",(a\_list[k],b\_list[k]),

font, fontScale, color, thickness, cv**.**LINE\_AA)

cv**.**circle(original\_image, (a\_list[k],b\_list[k]), r\_list[k], (0, 255, 0), 10) *## Orijinal Görüntü üzerine çizim*

ceviz**=**ceviz**+**1

**elif** (r\_list[k]**>=**33 **and** r\_list[k]**<**60): *## Fındık*

cv**.**putText(original\_image,"Findik",(a\_list[k],b\_list[k]),

font, fontScale, color, thickness, cv**.**LINE\_AA)

cv**.**circle(original\_image,(a\_list[k],b\_list[k]), r\_list[k], (0, 255, 0), 10) *## Orijinal Görüntü üzerine çizim*

fındık**=**fındık**+**1

**elif** (r\_list[k]**>=** 18 **and** r\_list[k]**<**33): *##Leblebi*

cv**.**putText(original\_image,"Leblebi",(a\_list[k],b\_list[k]),

font, fontScale, color, thickness, cv**.**LINE\_AA)

cv**.**circle(original\_image, (a\_list[k],b\_list[k]), r\_list[k], (0, 255, 0), 10)

leblebi**=**leblebi**+**1

**elif**(max(r\_list)**<=** 55): *## Görüntünün uzaktan çekilmesi*

print("Uzaktan Çekim")

**for** k **in** range(len(r\_list)):

**if** (r\_list[k]**>**45 **and** r\_list[k]**<=**55 ):

cv**.**putText(original\_image,"Ceviz",(a\_list[k],b\_list[k]),

font, fontScale, color, thickness, cv**.**LINE\_AA)

cv**.**circle(original\_image, (a\_list[k],b\_list[k]), r\_list[k], (0, 255, 0), 10) *## Orijinal Görüntü üzerine çizim*

ceviz**=**ceviz**+**1

**elif** (r\_list[k]**>=**40 **and** r\_list[k]**<**45):

cv**.**putText(original\_image,"Findik",(a\_list[k],b\_list[k]),

font, fontScale, color, thickness, cv**.**LINE\_AA)

cv**.**circle(original\_image,(a\_list[k],b\_list[k]), r\_list[k], (0, 255, 0), 10) *## Orijinal Görüntü üzerine çizim*

fındık**=**fındık**+**1

**elif** ( r\_list[k]**<**40):

cv**.**putText(original\_image,"Leblebi",(a\_list[k],b\_list[k]),

font, fontScale, color, thickness, cv**.**LINE\_AA)

cv**.**circle(original\_image, (a\_list[k],b\_list[k]), r\_list[k], (0, 255, 0), 10) *## Orijinal Görüntü üzerine çizim*

leblebi**=**leblebi**+**1

**elif**(max(r\_list)**>**140): *## Görüntünün yakından çekilmesi*

print("Yakından çekim")

**for** k **in** range(len(r\_list)):

**if** (r\_list[k]**>**140 ):

cv**.**putText(original\_image,"Ceviz",(a\_list[k],b\_list[k]),

font, fontScale, color, thickness, cv**.**LINE\_AA)

cv**.**circle(original\_image, (a\_list[k],b\_list[k]), r\_list[k], (0, 255, 0), 10) *## Orijinal Görüntü üzerine çizim*

ceviz**=**ceviz**+**1

**elif** (r\_list[k]**>=**100 **and** r\_list[k]**<**140):

cv**.**putText(original\_image,"Findik",(a\_list[k],b\_list[k]),

font, fontScale, color, thickness, cv**.**LINE\_AA)

cv**.**circle(original\_image,(a\_list[k],b\_list[k]), r\_list[k], (0, 255, 0), 10) *## Orijinal Görüntü üzerine çizim*

fındık**=**fındık**+**1

**elif** ( r\_list[k]**<**100):

cv**.**putText(original\_image,"Leblebi",(a\_list[k],b\_list[k]),

font, fontScale, color, thickness, cv**.**LINE\_AA)

cv**.**circle(original\_image, (a\_list[k],b\_list[k]), r\_list[k], (0, 255, 0), 10) *## Orijinal Görüntü üzerine çizim*

leblebi**=**leblebi**+**1

isim\_split**=**isim**.**split(".")

fig, ax **=** plt**.**subplots(2,3, figsize **=** (25,15))

fig**.**suptitle('ALL OUTPUTS FOR PROJECT', fontsize**=**20)

*# Line plots*

*## Gri görüntünün bastırılması*

ax[0,0]**.**set\_title('GRAY SCALE')

ax[0,0]**.**imshow(original\_image, cmap **=** "gray")

*## Bilteral Filtresi uygulanmış görüntünün bastırılması*

ax[0,1]**.**set\_title(isim\_split[0]**+**'\_BilateralFilter')

ax[0,1]**.**imshow(blur, cmap **=** "gray")

*## Adaptive Eşikleme işlemi yapılmış görüntünün Bastırılması*

ax[0,2]**.**set\_title(isim\_split[0]**+**'\_Threshold')

ax[0,2]**.**imshow(th3, cmap **=** "gray")

*## Opening işlemi uygulanmış görüntünün bastırılması*

ax[1,0]**.**set\_title(isim\_split[0]**+**'\_Opening')

ax[1,0]**.**imshow(opened\_img, cmap **=** "gray")

*## Closing işlemi uygulanmış görüntünün bastırılması*

ax[1,1]**.**set\_title(isim\_split[0]**+**'\_Closing')

ax[1,1]**.**imshow(closed\_img, cmap **=** "gray")

*## Medyan filtresi uygulanmış görüntünün bastırılması.*

ax[1,2]**.**set\_title(isim\_split[0]**+**'\_Median')

ax[1,2]**.**imshow(median, cmap **=** "gray")

*## Başlıklar arasındaki boşlukların ayarlanması*

plt**.**tight\_layout()

plt**.**show()

*## Tespit edilen nesne sayılarının döndürülmesi*

**return** ceviz,fındık,leblebi

cv**.**circle(original\_image,(a\_list[k],b\_list[k]), r\_list[k], (0, 255, 0), 10) *## Orijinal Görüntü üzerine çizim*

fındık**=**fındık**+**1

**elif** ( r\_list[k]**<**100):

cv**.**putText(original\_image,"Leblebi",(a\_list[k],b\_list[k]),

font, fontScale, color, thickness, cv**.**LINE\_AA)

cv**.**circle(original\_image, (a\_list[k],b\_list[k]), r\_list[k], (0, 255, 0), 10) *## Orijinal Görüntü üzerine çizim*

leblebi**=**leblebi**+**1

isim\_split**=**isim**.**split(".")

fig, ax **=** plt**.**subplots(2,3, figsize **=** (25,15))

fig**.**suptitle('ALL OUTPUTS FOR PROJECT', fontsize**=**20)

*# Line plots*

*## Gri görüntünün bastırılması*

ax[0,0]**.**set\_title('GRAY SCALE')

ax[0,0]**.**imshow(original\_image, cmap **=** "gray")

*## Bilteral Filtresi uygulanmış görüntünün bastırılması*

ax[0,1]**.**set\_title(isim\_split[0]**+**'\_BilateralFilter')

ax[0,1]**.**imshow(blur, cmap **=** "gray")

*## Adaptive Eşikleme işlemi yapılmış görüntünün Bastırılması*

ax[0,2]**.**set\_title(isim\_split[0]**+**'\_Threshold')

ax[0,2]**.**imshow(th3, cmap **=** "gray")

*## Opening işlemi uygulanmış görüntünün bastırılması*

ax[1,0]**.**set\_title(isim\_split[0]**+**'\_Opening')

ax[1,0]**.**imshow(opened\_img, cmap **=** "gray")

*## Closing işlemi uygulanmış görüntünün bastırılması*

ax[1,1]**.**set\_title(isim\_split[0]**+**'\_Closing')

ax[1,1]**.**imshow(closed\_img, cmap **=** "gray")

*## Medyan filtresi uygulanmış görüntünün bastırılması.*

ax[1,2]**.**set\_title(isim\_split[0]**+**'\_Median')

ax[1,2]**.**imshow(median, cmap **=** "gray")

*## Başlıklar arasındaki boşlukların ayarlanması*

plt**.**tight\_layout()

plt**.**show()

*## Tespit edilen nesne sayılarının döndürülmesi*

**return** ceviz,fındık,leblebi

ceviz\_t1,fındık\_t1,leblebi\_t1**=**Fonksiyon("train1.jpg")

print("Train-1 görüntüsü icin nesne sayıları:")

print("Ceviz Sayısı: " **+** str(ceviz\_t1))

print("Fındık Sayısı: " **+** str(fındık\_t1))

print("Leblebi Sayısı: " **+** str(leblebi\_t1))

ceviz\_t2,fındık\_t2,leblebi\_t2**=**Fonksiyon("train2.jpg")

print("Train-2 görüntüsü icin nesne sayıları:")

print("Ceviz Sayısı: " **+** str(ceviz\_t2))

print("Fındık Sayısı: " **+** str(fındık\_t2))

print("Leblebi Sayısı: " **+** str(leblebi\_t2))

ceviz\_t3,fındık\_t3,leblebi\_t3**=**Fonksiyon("train3.jpg")

print("Train-3 görüntüsü icin nesne sayıları:")

print("Ceviz Sayısı: " **+** str(ceviz\_t3))

print("Fındık Sayısı: " **+** str(fındık\_t3))

print("Leblebi Sayısı: " **+** str(leblebi\_t3))

ceviz\_t4,fındık\_t4,leblebi\_t4**=**Fonksiyon("train4.jpg")

print("Train-4 görüntüsü icin nesne sayıları:")

print("Ceviz Sayısı: " **+** str(ceviz\_t4))

print("Fındık Sayısı: " **+** str(fındık\_t4))

print("Leblebi Sayısı: " **+** str(leblebi\_t4))

ceviz\_t5,fındık\_t5,leblebi\_t5**=**Fonksiyon("train5.jpg")

print("Train-5 görüntüsü icin nesne sayıları:")

print("Ceviz Sayısı: " **+** str(ceviz\_t5))

print("Fındık Sayısı: " **+** str(fındık\_t5))

print("Leblebi Sayısı: " **+** str(leblebi\_t5))

*######################### TEST VERİLERİ #######################*

ceviz\_te1,fındık\_te1,leblebi\_te1**=**Fonksiyon("test1.jpg")

print("Test-1 görüntüsü icin nesne sayıları:")

print("Ceviz Sayısı: " **+** str(ceviz\_te1))

print("Fındık Sayısı: " **+** str(fındık\_te1))

print("Leblebi Sayısı: " **+** str(leblebi\_te1))

ceviz\_te2,fındık\_te2,leblebi\_te2**=**Fonksiyon("test2.jpg")

print("Test-2 görüntüsü icin nesne sayıları:")

print("Ceviz Sayısı: " **+** str(ceviz\_te2))

print("Fındık Sayısı: " **+** str(fındık\_te2))

print("Leblebi Sayısı: " **+** str(leblebi\_te2))

ceviz\_te3,fındık\_te3,leblebi\_te3**=**Fonksiyon("test3.jpg")

print("Test-3 görüntüsü icin nesne sayıları:")

print("Ceviz Sayısı: " **+** str(ceviz\_te3))

print("Fındık Sayısı: " **+** str(fındık\_te3))

print("Leblebi Sayısı: " **+** str(leblebi\_te3))

ceviz\_te4,fındık\_te4,leblebi\_te4**=**Fonksiyon("test4.jpg")

print("Test-4 görüntüsü icin nesne sayıları:")

print("Ceviz Sayısı: " **+** str(ceviz\_te4))

print("Fındık Sayısı: " **+** str(fındık\_te4))

print("Leblebi Sayısı: " **+** str(leblebi\_te4))

ceviz\_te5,fındık\_te5,leblebi\_te5**=**Fonksiyon("test5.jpg")

print("Test-5 görüntüsü icin nesne sayıları:")

print("Ceviz Sayısı: " **+** str(ceviz\_te5))

print("Fındık Sayısı: " **+** str(fındık\_te5))

print("Leblebi Sayısı: " **+** str(leblebi\_te5))

ceviz\_te6,fındık\_te6,leblebi\_te6**=**Fonksiyon("test6.jpg")

print("Test-6 görüntüsü icin nesne sayıları:")

print("Ceviz Sayısı: " **+** str(ceviz\_te6))

print("Fındık Sayısı: " **+** str(fındık\_te6))

print("Leblebi Sayısı: " **+** str(leblebi\_te6))