

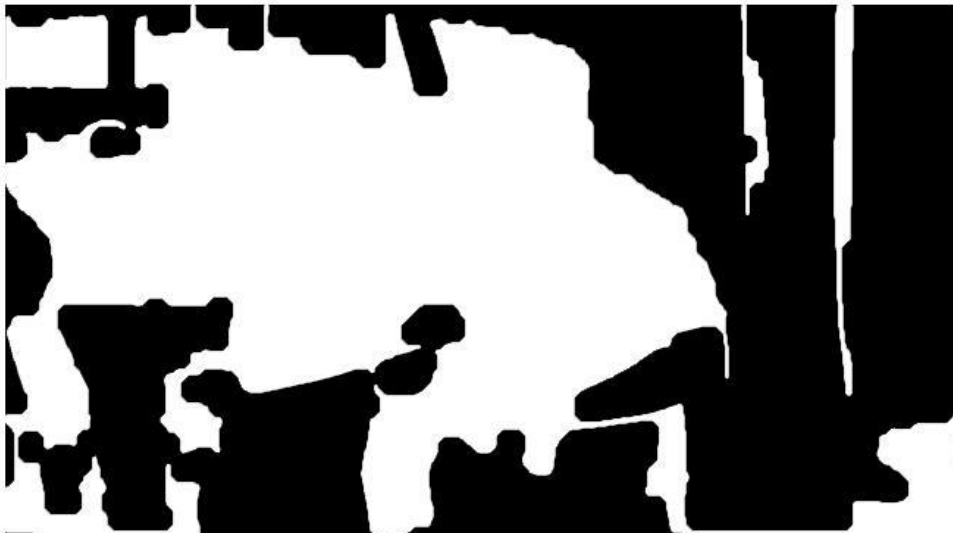
```
resim=imread('C:\Users\muura\Desktop\ai12.jpg');
imshow(resim);
sobel=rgb2gray(resim);
A=double(sobel);
    for i=1:size(A,1)-2
        for j=1:size(A,2)-2
            %Yatay sobel kernel matrisi
            Gx=((2*A(i+2,j+1)+A(i+2,j)+A(i+2,j+2))-(2*A(i,j+1)+A(i,j)+A(i,j+2)));
            %Dikey sobel kernel matrisi
            Gy=((2*A(i+1,j+2)+A(i,j+2)+A(i+2,j+2))-(2*A(i+1,j)+A(i,j)+A(i+2,j)));
            sobel(i,j)=sqrt(Gx.^2+Gy.^2);
        end
    end
image=sobel %Resmi Okuma
figure(1), imshow(image); %Resmi Gösterme
level=graythresh(image) %Parlaklık eşiğini otomatik belirlendi ve 0 ile 1 arasında sayı
oluşturdu
bw=im2bw(image,0.7); %Resim tamamen siyah-beyaz piksellere dönüştü.
figure(3),imshow(bw);
bw=bwareaopen(bw,30); %30px den daha az sayıda olan nesneler kaldırılıyor.
figure(4),imshow(bw);
se=strel('disk',10); %Yarıçapı 10px olan disk biçiminde yapısal element oluşturuyoruz.
bw=imclose(bw,se); %Yapısal element yardımıyla iç kısımdaki boşluklar kayboldu.
figure(5),imshow(bw);
bw=imfill(bw,'holes'); %Resimde çukur diye nitelendirilen yerleri dolduruyoruz.
figure(5), imshow(bw);
[B,L] = bwboundaries(bw,'noholes'),disp(B)
```

```
hold on
for k = 1:length(B)
    boundary = B{k}; %'k' etiketindeki nesnenin sınır kordinatlarını (X,Y) belirler
    plot(boundary(:,2), boundary(:,1), 'w', 'LineWidth', 2)
end
fprintf('Nesneler işaretlenmiştir. Toplam Nesne Sayısı=%d\n',length(B))
maskedRgbImage = bsxfun(@times, resim, cast(bw, 'like', resim));
figure;imshow(maskedRgbImage);
```

YAZMIŞ OLDUĞUM ALGORİTMANIN İLK RESİM ÜZERİNDEKİ İŞLEVLERİ SIRASIYLA GÖSTERİLMİŞTİR.







BİRİNCİ RESMİN MASKELEME İLE ÇIKAN SON HALİ



İKİNCİ RESMİN ALGROİTMADAN ÇIKMIŞ HALLERİ AŞAĞIDA VERİLMİŞTİR.







İKİNCİ RESMİN SON HALİ

