

DENEY NO: 7**GÖRÜNTÜ İŞLEME UYGULAMALARI****1. GİRİŞ**

Elde edilen görüntünün bilgisayara aktarılıp üzerinde herhangi bir işlem yapılması ve ardından görüntüleyici çıkışa iletilmesine sayısal görüntü işleme adı verilir. Görüntü işleme teknikleri ile, sayısal görüntüler iyileştirilerek nesne tanıma, hedef tanıma gibi işlemler gerçekleştirilebilmektedir. Görüntü işleme, amaca göre çeşitli işlemlerden oluşmaktadır. Görüntü işleme; tasarım, imalat, güvenlik, tıp, elektronik, makine ve jeodezi gibi alanlarda çok geniş bir uygulama alanı bulmuştur.

- Askeri (hedef tanıma, izleme)
- Tıp (damar analizi, bilgisayarlı tomografi, ultrason)
- Güvenlik (nesne takibi, hareket algılama, yüz tanıma)
- Trafik (trafik kontrol, plaka tanıma, trafik ışığı (işareti) tanıma)
- Endüstri (nesne sayma, kalite kontrol, robotik uygulamalar)
- Tarımsal uygulamalar (ekin verimliliği tespiti)
- Astronomi (uydu görüntüleri ile hava tahmini)
- Jeodezi ve Fotogrametri (uzaktan algılama)
- Perakende (insan sayma, davranış analizi, mağaza izleme)
- Çevre güvenliği (çevre kirliliği tespiti)

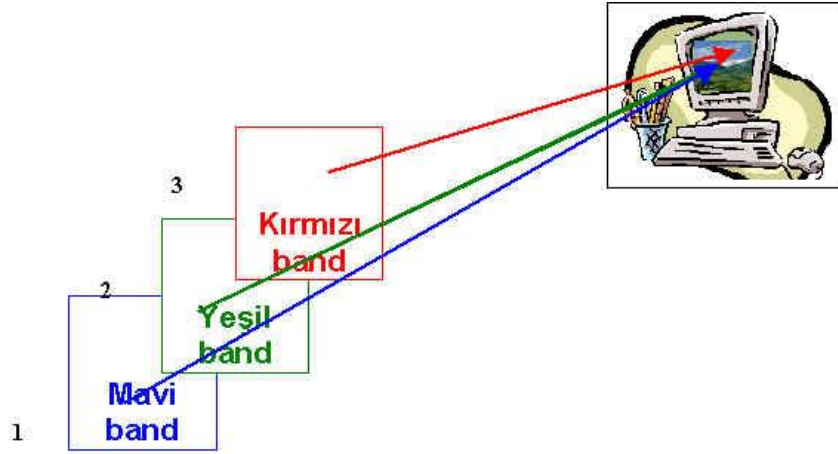
Sayısal resim haline getirilmiş olan gerçek yaşam görüntüsünün, girdi resim olarak işlenerek, resmin özelliklerinin ve görüntüsünün değiştirilmesi sonucunda yeni bir resmin oluşturulması işlemidir. Bir çok amaç için kullanılmaktadır.

- Görüntü İyileştirme
- Görüntü Sıkıştırma
- Biyometrik Tanıma
- Otomatik yüz araç obje tanıma ve takip etme

2. İkili Görüntü (Binary Image)

Bir resmin sayısallaştırılmasının açıklanması amacı ile öncelikle Siyah-Beyaz resim göz önünde bulundurulmuştur. Siyah-Beyaz resim sadece iki gri değerden oluşan bir resimdir. Böylesi bir görüntüde her bir piksel ya siyah ya da beyaz olarak oluşur. Burada sembolik olarak beyaz pikseller 1, siyah pikseller 0 değeri ile gösterilecektir. Renkli görüntüler bilgisayar ekranlarında 24 bit lik veri olarak görüntülenir. Görüntüleme R(Kırmızı), G(Yeşil), B(Mavi) kodlanmış aynı objeye ait üç adet gri düzeyli görüntünün üst üste ekrana iletilmesi ile oluşur. Elektro-manyetik spektrumda 0,4-0,5 mm dalga boyu mavi renge; 0,5-0,6 mm dalga boyu yeşil renge; 0,6-0,7 mm dalga boyu kırmızı renge karşılık gelir. Bu dalga boylarında elde edilmiş üç gri düzeyli görüntü bilgisayar ekranında sırası

ile kırmızı-yeşil-mavi kombinasyonunda üst üste düşürülecek olursa renkli görüntü elde edilmiş olur.



Şekil 1. RGM Görüntüler

3. Niteliklendirme

Görüntünün piksel değerlerinin belirli aralıklarda olması, meydana gelen görüntünün niteliğini değiştirir. Örneğin 0 beyazı ve n1 de siyahı temsil ederse ve bu değerler arası gri tonlarını ifade eder. Burada $n = 2^b$ olmak üzere, b değeri görüntünün 1 pikselini ifade etmek için gereken bit sayısıdır. Örneğin $b=8$ ise 256 adet gri tonu bulunmaktadır. $B=1$ ise resim sadece 0 ve 1 'lerden oluşur ve buna İkili resim(Binary Image) denir.

4. Matlab'da Görüntü İşleme

Söz konusu uygulamaları geliştirmek için kullanılan, Matlab'ın görüntü işleme komutlarının ve image processing tolls'un bazılarını örneklerle inceleyeceğiz.

imread: imread ile üzerinde çalışılmak istenilen fotoğraf bir değişkene atanarak matlab workspace için tanımlanır ve ardından imshow ile pencerede gösterilebilir. Workspace'te imgenin boyutu, çözünürlüğü gibi bilgileri görebilirsiniz.

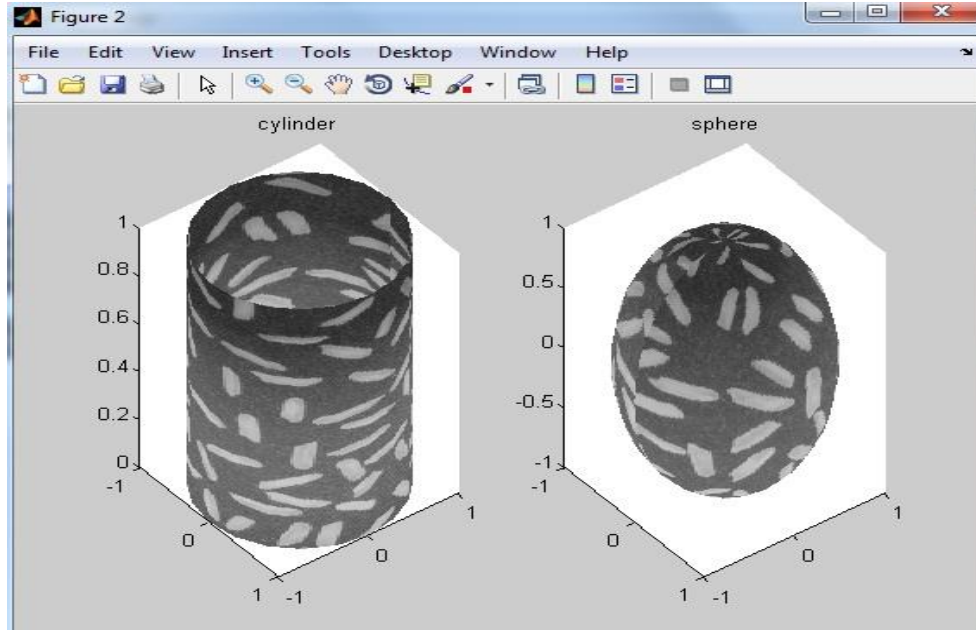
Imtool: imtool ile bazı işlemlerin yapılabileceği bir pencere açılır. Örnek kodları yazarken workspacedeki verileri temizlemek için, **clear all** ve **clc** komutlarını en başa yazmayı unutmamamız gerekir.

- `I = imread('ornek.jpg')`
`imshow(I)`
`imtool(I)`

Warp: görüntüyü (içinde yazı olan görüntü daha iyi olur) belli yüzeylerde gösterir.

- `resim = imread('rice.png');`
- `[x1,y1,z1] = cylinder;`
- `[x2,y2,z2] = sphere;`
- `imshow(resim); title('Orijinal resim');`
- `figure;`

- subplot(1,2,1); warp(x1,y1,z1,resim); title('cylinder');
- subplot(1,2,2); warp(x2,y2,z2,resim); title('sphere');
-



Şekil 2. Elde edilen görüntü

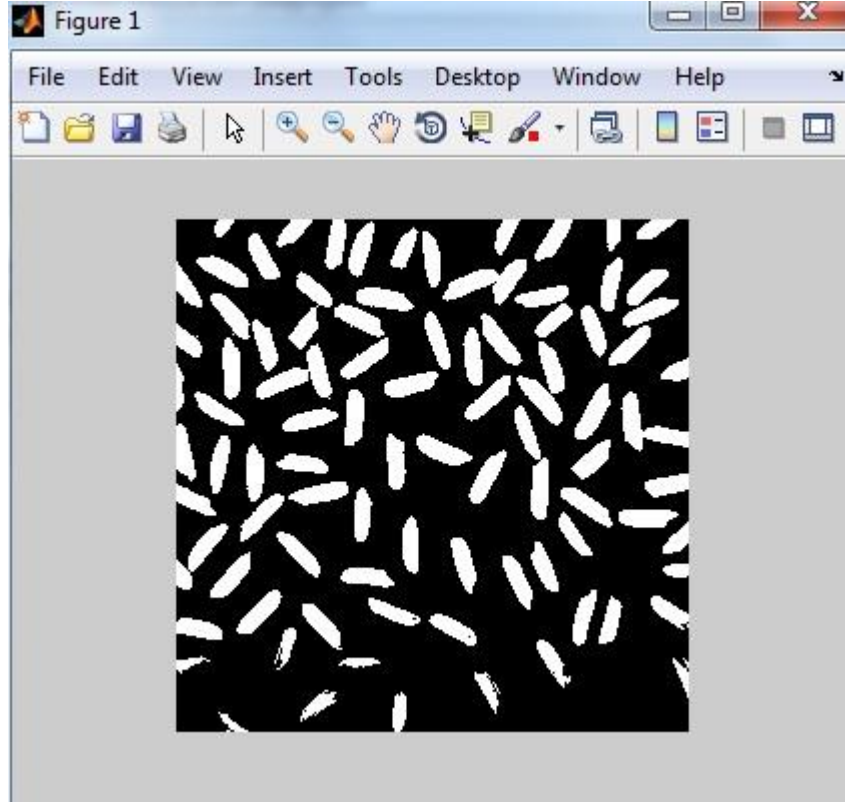
4.1. Tür Dönüşüm Komutları

rgb2gray: renkli görüntüyü gri seviyeye çevirir.

graythresh: graythreshold komutu görüntüdeki parlaklık eşikini otomatik olarak belirler ve sonuç olarak 0 – 1 arasında bir sayı (level) oluşturur. Görüntüdeki parlaklık sınırları ile yapacağımız işlemlerde graythreshold' tan elde ettiğimiz sayıyı kullanırız. Görüntü üzerinde belirli işlemleri yapabilmemiz için öncelikle gri seviyede çalışmamız gerekir. Bunun için **I= rgb2gray(I)** komutu kullanılır.

Bu kod örneği ile görüntünün binary şekli elde edilir.

```
level = graythresh(I);
bw = im2bw(I,level);
bw = bwareaopen(bw, 50);
figure; imshow(bw);
```



Şekil 3. Elde edilen görüntü

- **im2uint8:** uint8 renk hassasiyetine dönüştürür. Genelde görüntüler uint8 türündedir. Her piksel İşaretsiz 8 bit (0 – 255) arası değere sahiptir.

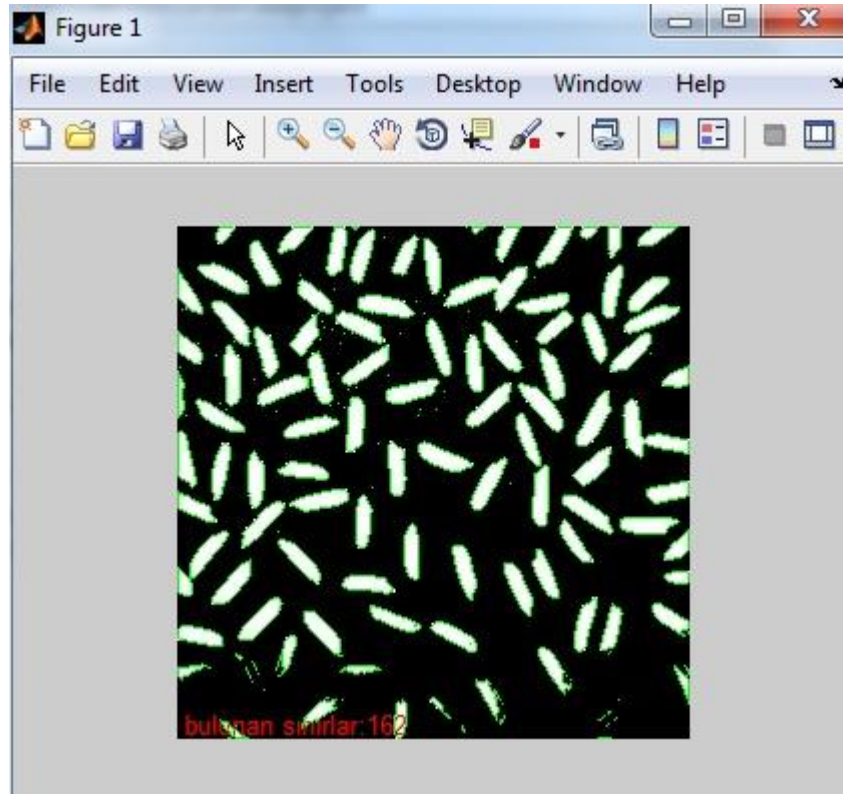
Diğer dönüşüm komutları; demosaic, gray2ind, grayslice, im2int16, label2rgb, im2double, im2uint16, mat2gray...

4.2. Görüntü Analizi Komutları

Bwboundaries: binary modda bölgelerin sınırlarını belirler.

Bu örnekte, yeşil çizgiyle sınırlanmış alanların sayısını belirleyebiliriz. Bu örnekle daha sonra bahsedeceğimiz, morfolojik işlemler gerçekleştirilerek basit nesne sayma uygulaması yapılabilir.

```
I = imread('rice.png');
BW = im2bw(I, graythresh(I));
B = bwboundaries(BW);
figure; imshow(BW);
text(10,10, strcat('\color{red} bulunan sınırlar:', num2str(length(B))))
hold on;
for k = 1:length(B)
    boundary = B{k};
    plot(boundary(:,2), boundary(:,1), 'g', 'LineWidth', 1)
end
```



Şekil 4. Sınırı belirlenen görüntü

Edge: özel filtreler yardımıyla gri seviyedeki görüntülerin sınırları belirlenir.

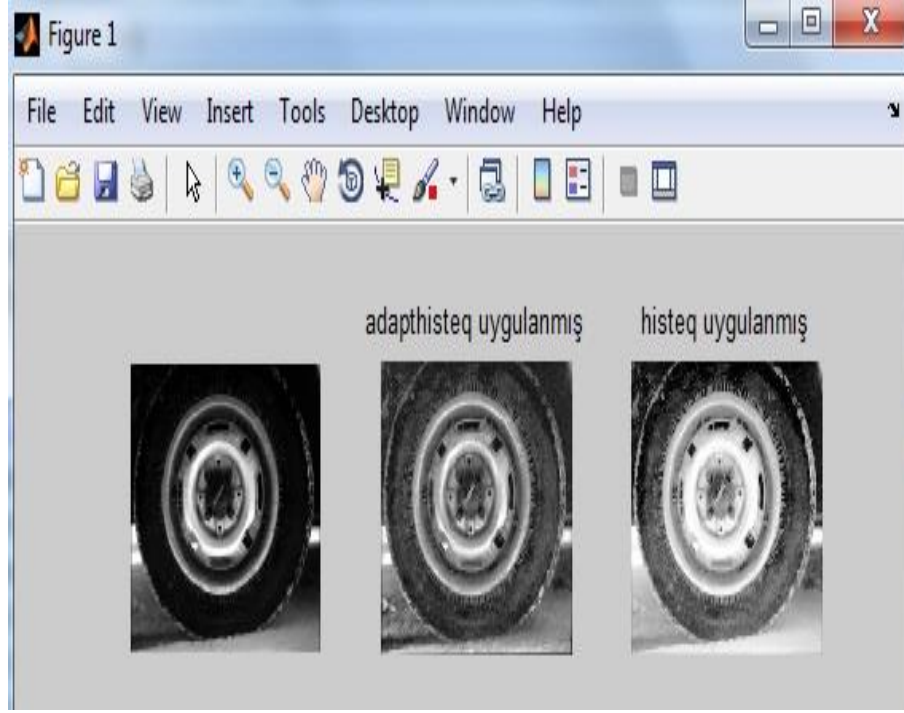
```
I = imread('rice.png');  
imshow(I);  
BW1 = edge(I,'prewitt');  
BW2 = edge(I,'canny');  
figure; imshow(BW1);  
figure; imshow(BW2);
```

diğer analiz komutları: **hough, houghlines, corner...**

4.3. Görüntü İyileştirme Komutları

- **imadjust:** görüntü yoğunluğu değerini ve renk haritasını ayarlar.
histeq: histogramı eşitleyerek kontrastı artırır.
adapthisteq: CLAHE(kontrast sınırlı adaptif histogram eşitleme) [algoritması](#) kullanarak kontrastı artırır.

```
I = imread('tire.tif');  
A = adapthisteq(I,'clipLimit',0.02,'Distribution','rayleigh');  
B = histeq(I);  
subplot(231),imshow(I);  
subplot(232),imshow(A); title('adapthisteq uygulanmış');  
subplot(233),imshow(B); title('histeq uygulanmış');
```



Şekil 5. İyileştirme uygulanan görüntü

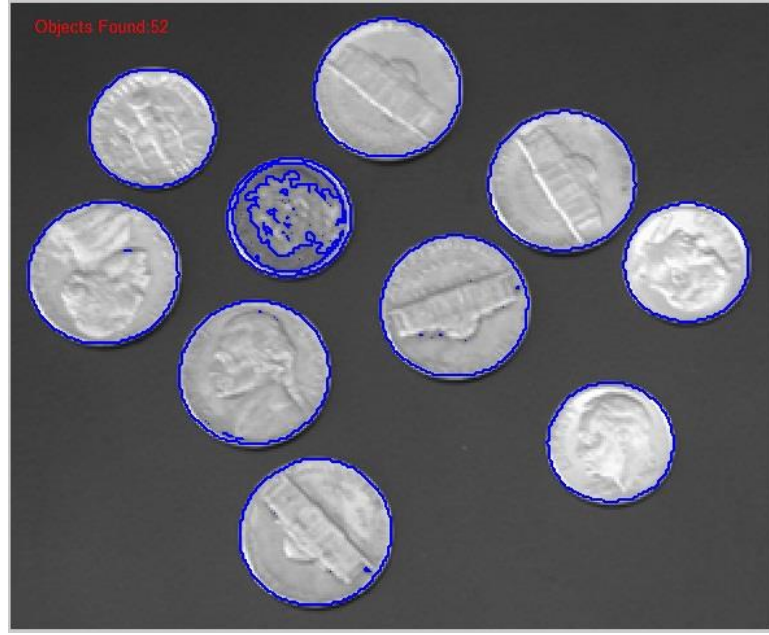
- **medfilt2:** 2 boyutlu medyan filtreleme yapar.
wiener2: 2 boyutlu adaptif gürültü temizleme filtresidir.

```
I=imread('eight.tif');  
no=imnoise(I,'salt & pepper',0.02);  
filt1=medfilt2(no); filt2=wiener2(no);  
subplot(2,2,1); imshow(I), title('Orijinal resim')  
subplot(2,2,2); imshow(no), title('Gürültülü resim')  
subplot(2,2,3); imshow(filt1), title('Medyan filtresi uygulanmış')  
subplot(2,2,4); imshow(filt2), title('Wiener filtresi uygulanmış')
```

5. MATLAB ile Görüntü İşleme Uygulaması

Bu uygulamada resmin içindeki bozuk paraları bulmaya, etiketlemeye ve değerlerine göre hesaplamaya çalışılmıştır. Görüntü olarak Matlab programında hazır bulunan 'coins.png' yi kullanılmıştır.

- Matlab' de görüntü işleme ile ilgili uygulama geliştirirken yapmamız gereken ilk adımlar gri seviyeye, binary seviyeye dönüştürme işlemleridir. Uygulamanın ilk satırlarında otomatik threshold değerine göre imgeyi binary seviyeye dönüştürülmüştür.
- Görüntüler im2bw komutu ile binary seviyeye dönüştürüldükten sonra aşağıdaki resimde görüldüğü gibi birçok küçük alanlar oluşabilir. Görüntüyü bölümleyebilmek için bu alanları doldurmamız gerekir. Bunun için imfill işlemini gerçekleştirilmiştir.



Şekil 6. coins.png görüntüsü

- Program iki çıktı vermektedir. Birincisinde sınırları çizilen bozuk paraların sayısı belirlenir; ikincisinde ise bozuk paralara alanlarına göre değer atanıp, değerleri sayılır.

5.1. Bölgelerin belirlenmesi

Sınırlar **bwboundaries** işlemi ile çizilir. For döngüsü çizim sayısı kadar döndürülerek çizimlerin resmin üzerinde gösterilmesine olanak sağlar.

► **plot** işleminin parametreleri değiştirilerek çizimin kalınlığı, rengi istenilen şekilde belirlenebilir.

```
B = bwboundaries(coin2);
```

- `text(10,10, strcat('\color{red}Objects Found:', num2str(length(B))))`
- `hold on;`
- `for k=1:length(B),`
- `boundary = B{k};`
- `plot(boundary(:,2), boundary(:,1), 'b', 'LineWidth', 2);`
- `end`

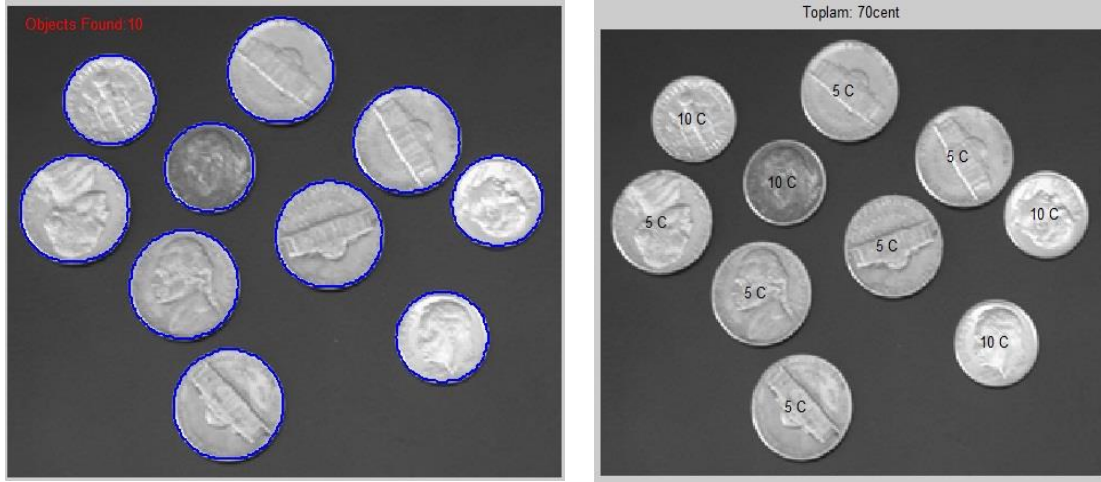
5.2. Alanların hesaplanması

- **bwlabel** işlemi ile bağlı bileşenlerin (bozuk paralar) bilgileri bir değişkene atanır.

► **regionprops** ile bölgelerin özellikleri çıkarılır. Daha sonra alanlar hesaplanır (for döngüsü) alanı belli bir değer üzerinde olanlar '5 C' ile diğerleri '10 C' ile etiketlenir.

- `[Bilgi Number]=bwlabel(coin2);`
- `prop=regionprops(Bilgi, 'Area', 'Centroid');`
- `total=0;`
- `figure; imshow(imread('coins.png')); hold on`
- `for n=1:size(prop,1)`
- `cent=prop(n).Centroid;`
- `X=cent(1); Y=cent(2);`

- if prop(n).Area>2000
- text(X-10,Y,'5 C')
- total=total+5;
- else
- total=total+10;
- text(X-10,Y,'10 C')
- end
- End



Şekil 7. Görüntüdeki farklı nesneleri tanıma

Matlab Kodu:

- clc;
- clear all;
- figure; imshow('coins.png');
- coin1 = im2bw(imread('coins.png'));
- coin2 = imfill(coin1,'holes');
- figure; imshow(imread('coins.png'));
- B = bwboundaries(coin2);
- text(10,10, strcat('\color{red} Objects Found:', num2str(length(B))))
- hold on;
- for k=1:length(B),
- boundary = B{k};
- plot(boundary(:,2),boundary(:,1),'b','LineWidth',2);
- end
- [Bilgi Number]=bwlabel(coin2);
- prop=regionprops(Bilgi,'Area','Centroid');
- total=0;
- figure; imshow(imread('coins.png'));hold on
- for n=1:size(prop,1)
- cent=prop(n).Centroid;
- X=cent(1);Y=cent(2);
- if prop(n).Area>2000
- text(X-10,Y,'5 C')
- total=total+5;

- else
- total=total+10;
- text(X-10,Y,'10 C')
- end
- end
- hold on
- title(['Toplam: ',num2str(total),'cent'])

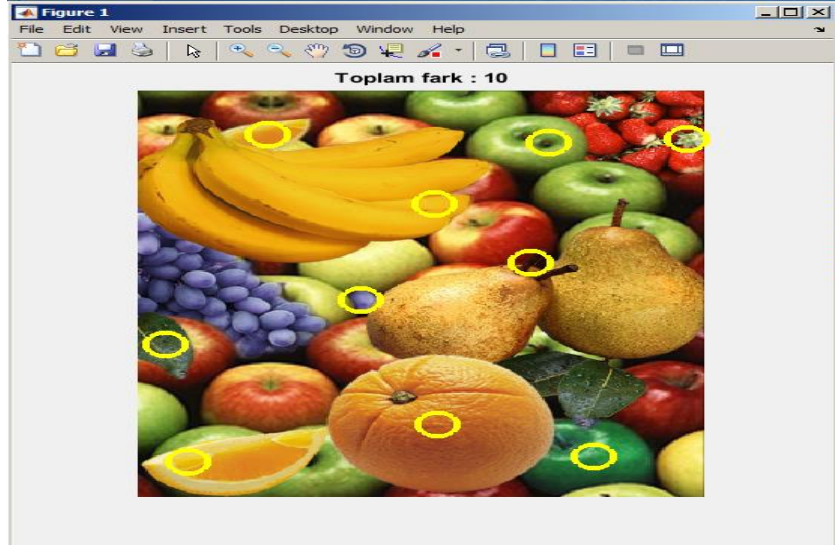
6. DENEY

Bu uygulamada verilen benzer iki resim arasındaki farklar bulunup bu farklar hem sayıca hesaplanmakta hem de işaretlenmektedir.



Şekil 8. Benzer iki farklı resim

- im1 = rgb2gray(imread('resim1.png'));
- im2 = rgb2gray(imread('resim2.png'));
- fark = imabsdiff(im1,im2);
- bw = bwareaopen(fark,55);
- bw = imfill(bw,'holes');
- SE = strel('square',1);
- bw2 = imerode(bw,SE);
- fark = regionprops(bw2, 'all');
- c = [fark.Centroid];
- imshow('resim2.png');
- title(['Toplam fark : ',num2str(length(fark))]);
- hold on;
- x = c(1:2:end);
- y = c(2:2:end);
- plot(x,y,'yo','MarkerSize',20,'LineWidth',4);



Şekil 9. İki resim arasındaki farkların bulunması

- Uygulama 2'deki iki farklı resmi kullanarak aynı matlab kodu kullanılarak sonuç olarak 3 farklı resim çıktı alınması gerekmektedir(Figure 1, Figure 2 ve Figure 3).
- Figure 1: Standart 1. resmi ifade edecek.
- Figure 2: Farklı olan kısımları daire içine alınmış olan resmi ifade edecek.
- Figure 3: Standart 2.resmi gri ton kodlamalı şekilde gösterecek resmi ifade edecek.

Not: Deney esnasında deney föyündeki ve slayttaki tüm bilgi ve kod içeriğinden sorumlu olduğunuzu unutmayınız...

KAYNAKLAR

- www.mathworks.com
- <http://www.elektrikport.com/>
- http://web.firat.edu.tr/iaydin/bmu357/Bolum_1.pdf
- <http://www.yildiz.edu.tr/~bayram/sgi>
- <http://yzgrafik.ege.edu.tr/~tekrei/dosyalar/sunum/gi.pdf>