



**İSPARTA UYGULAMALI BİLİMLER ÜNİVERSİTESİ**

**TEKNOLOJİ FAKÜLTESİ**

**BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ**

**SAYISAL GÖRÜNTÜ İŞLEME**

**Final Ödevi**

---

**Dersin Öğretim Üyesi**  
**Dr. Öğr. Üyesi Serdar PAÇACI**

---

**Furkan TOPTAŞ**

**2130696007**

# 1. GİRİŞ

## 1.1. Proje Gereksinimleri

Sayısal Görüntü İşlemi final ödevi olarak verilen proje gereksinimleri aşağıdaki gibidir;

- 1- Kendi isminizin yazdığı bir kağıt üzerine bozuk paralar (1 TL, 50 kuruş, 25 kuruş, 5 kuruş gibi) yerleştiriniz.
- 2- Herhangi bir kamera ile kameranın kağıt üzerindeki paraları ve isminizi alacağı şekilde kısa bir video çekiniz. (Max 30 saniye)
- 3- Video çekerken istediğiniz şekilde bazı paraları çekim alanından çıkarıp sonra tekrar istediğiniz sırada tekrar çekim alanına getiriniz.
- 4- Amacımız bu videoyu işleyip bozuk paraları ve değerlerini tespit etmek olacaktır. Bu doğrultuda bir algoritma geliştiriniz.
- 5- Tasarladığınız algoritma sonucunda tespit ettiğiniz paraların üzerinde herhangi bir yere paraların tutarlarını yazdırınız.
- 6- Ayrıca tespit edilen paraların toplam tutarını da videoda uygun bir alana yazdırınız. (Paraların çıkarılma ve koyulma anındaki hesaplama hataları önemli değildir.)
- 7- Algoritmanızı nasıl tasarladığınızı hangi amaçla algoritma adımlarınızı oluşturduğunuzu ve elde ettiğiniz sonuçlara dair bilgileri bir rapor halinde sununuz. Araştırma sırasında kullandığınız kaynakları belirtmeyi unutmayınız.
- 8- Son olarak algoritmanızın çalışması sonucunda oluşturduğunuz sonuç videosunu kaydediniz.

## 1.2. Proje Kodları

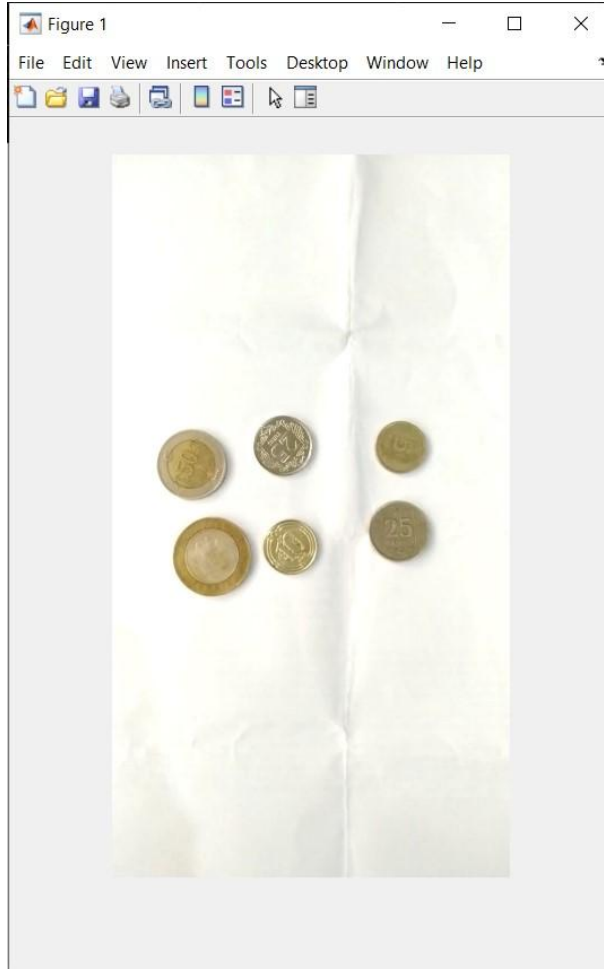
<https://github.com/furkantoptass/sayisalGoruntuIslemeFinal>

# 2. KULLANILAN ALGORİTMA

Geliştirilen uygulamada bozuk paraların tanımlanması ve daha sonrasında da boyutlarına göre videodaki bozuk paraların toplam değerlerinin hesaplanması sağlanmıştır. Bu işlemleri 6 adımda anlatacağım. İşlemlerimiz 6 adımda yapılmaktadır.

### 1.Adım:

Öncelikli olarak Bozuk paralarımızın olduğu videonun okunması işlemini yapıyoruz. Bu işlemler birlikte videonun her bir framenin resmiyi alıyoruz.

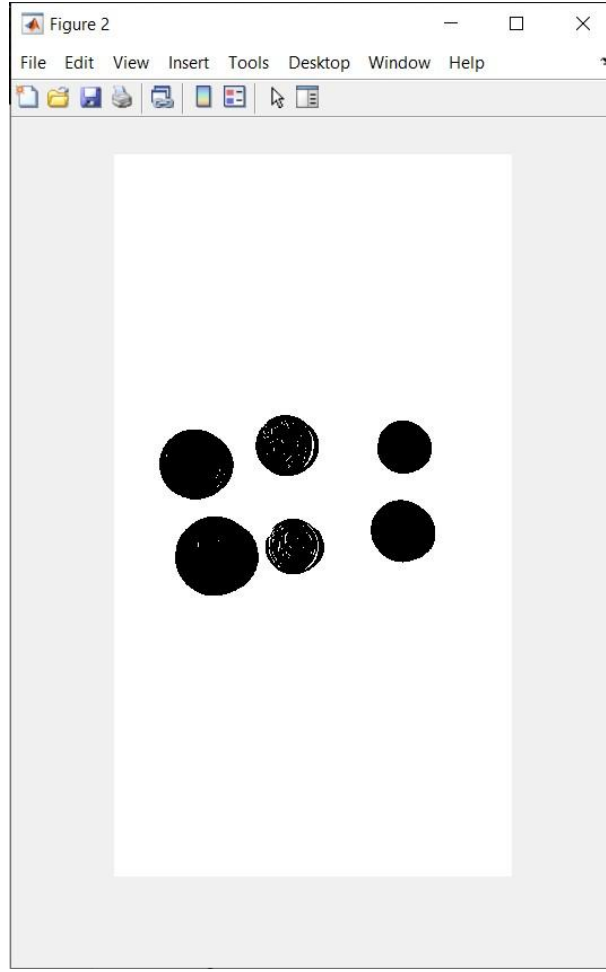


Şekil 1: Videodaki ilk görüntüyü alma

## 2.Adım:

Resmimizi okuduktan sonra önce gri-tona çevirip (2D) sonra Binary (Siyah-Beyaz) formata çeviriyoruz. Çünkü diğer işlemler için görüntü binary modda olması lazım. Bu adımda doldurma (fill) ve gereksiz bileşenlerin kaldırılması yapılacaktır.

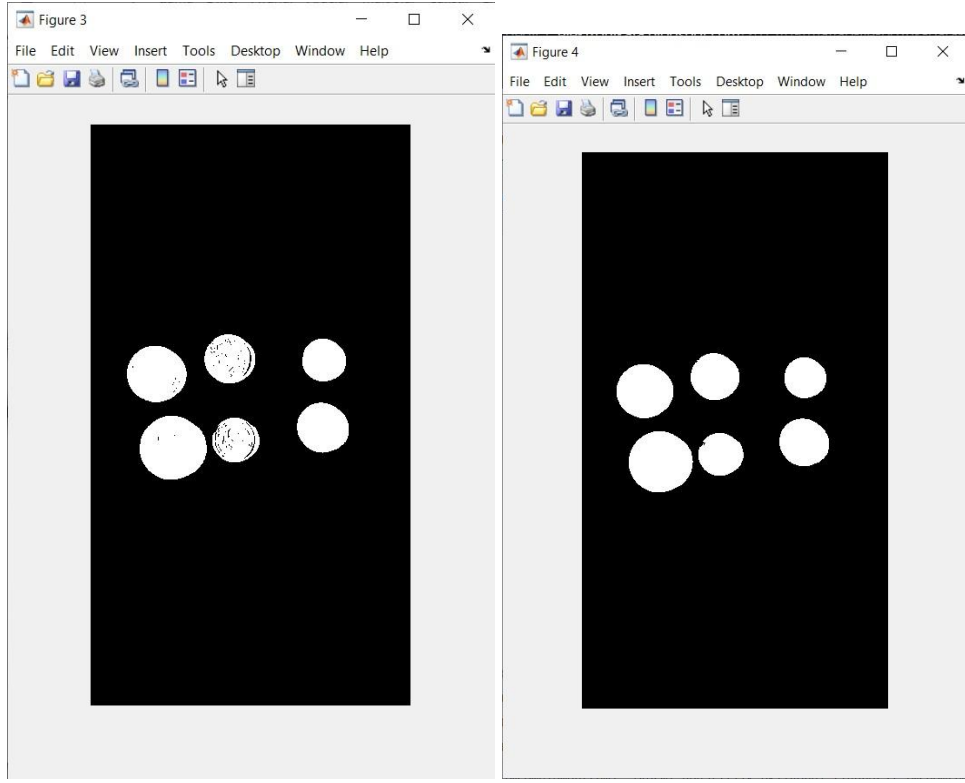
imcomplement: binary modda siyah alanlar beyaz, beyaz alanlar siyaha çevrilir. Bu kodu eklememizdeki amaç imfill komutunu hatasız kullanabilmemiz içindir. İmfill komutunu hatasız kullanmak için nesnelerimiz beyaz arkaplan beyaz olması gerekmektedir.



Şekil 2: Siyah Beyaz formata çevirme

### 3.Adım:

Bu bölümde resmimize morfolojik filtreler uyguluyoruz. Öncelikle strel dairesel bir yapısal element oluşturup imerode komutu ile aşındırma işlemi yapıyoruz. Bunu yapmamızdaki amaç yapışık olan paraların ayırmaktır. Böylece daha düzgün sonuçlar alabiliriz.

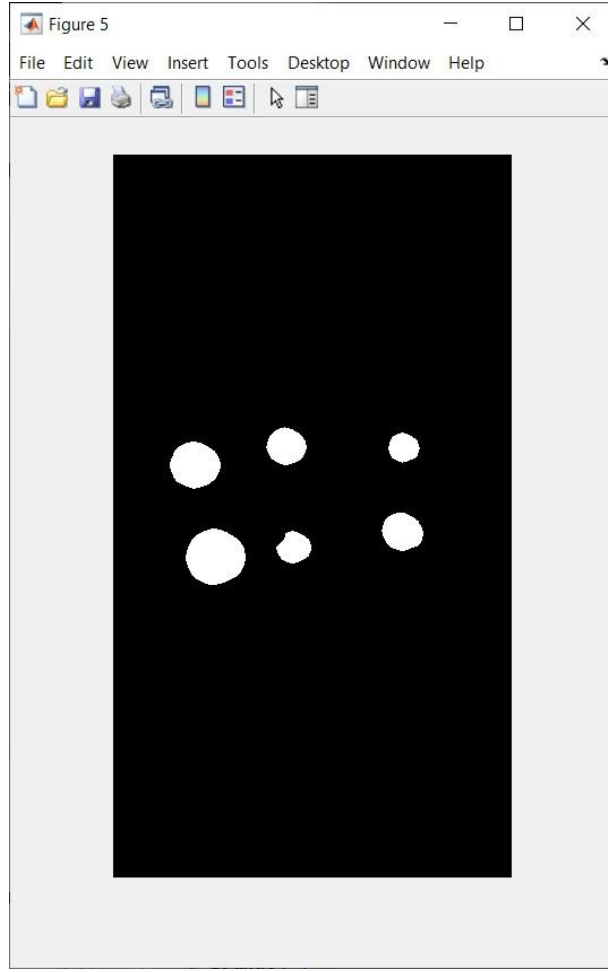


Şekil 3: Paraların Birbirinden Ayırma İşlemi

#### 4.Adım:

Resmimizle ilgili ön hazırlık bitmiştir. Resmimiz hazır haldedir. Bundan sonra asıl komutlara geçeceğiz.

`bwboundaries` komutu ile para adetini öğrenip döngüye sokup her paranın alanlarını öğreniyoruz ve bu alanlara göre paralarının tanımını yapıp toplam parayı ekrana yansıtıyoruz.



Şekil 4: Paraların Alanını Hesaplama

**5.Adım:**

Bu adımda hesaplanan alanlara göre ekrana paraların değerlerinin yazılması işlemi yapıyoruz.



Şekil 5: Paraların değerleri ve toplamını gösteren ekran

#### 6.Adım:

Aslında diğer adımlarla bütün işlemlerini yapıyoruz fakat videomuzda hesaplanan değerlerin videosunu kaydetmemiz gerekmektedir. Bu amaçla son adım olarak da bunu ekledim.

```
v = VideoWriter('newfile.mp4');  
vidObj = VideoReader('C:\Users\topta\Desktop\matlab_odev\para.mp4');  
% vidObj.CurrentTime = 1;  
open(v)
```

Şekil 6: VideoWriter Tanımlama Görüntüsü

```

frameR = getframe(gcf);
writeVideo(v, frameR)

end
close(v)

```

Şekil 7: Framedeki görüntüleri videoya yazma işlemi

### 3. KULLANILAN ARAÇLAR

Sistem geliştirilirken Matlab IDE'si üzerinde geliştirilmeler sağlanmıştır.

#### 3.1. MATLAB

MATLAB® , dünyamızı dönüştüren sistemleri ve ürünleri analiz etmek ve tasarlamak için mühendisler ve bilim adamları için özel olarak tasarlanmış bir programlama platformudur . MATLAB'ın kalbi, hesaplamalı matematiğin en doğal ifadesini sağlayan matris tabanlı bir dil olan MATLAB dilidir.

### 6. Referanslar

1. Matlab VideoWriter(<https://www.mathworks.com/help/matlab/ref/videowriter.html>)
2. Matlab VideoReader([https://www.mathworks.com/help/matlab/import\\_export/read-video-files.html](https://www.mathworks.com/help/matlab/import_export/read-video-files.html))
3. Matlab Youtube Channels Görüntü İşleme ile Bozuk Para Sayma (<https://www.youtube.com/watch?v=PfuIapC4zr4>)
4. Matlab Youtube Channels Görüntü İşleme ile Bozuk Para Sayma (<https://www.youtube.com/watch?v=LcXeij4CkyE>)
5. Matlab Görüntü İşleme ile Bozuk Para Sayma Blog (<http://elektronikafa.blogspot.com/2015/09/matlab-ile-bozuk-para-sayma-algoritmas.html>)
6. MatlabWorks Example (<https://www.mathworks.com/solutions/image-video-processing/video-processing.html#commonapplications>)
7. Matlab Works Object Detection Example(<https://www.mathworks.com/help/vision/ug/motion-based-multiple-object-tracking.html>)
8. MatlabWorks Video Progressing in Matlab (<https://www.mathworks.com/matlabcentral/fileexchange/68852-code-examples-from-video-processing-in-matlab>)
9. Görüntü İşleme Uygulamaları (<http://bil.muh.firat.edu.tr/sites/bil.muh.firat.edu.tr/files/G%C3%B6r%C3%BCnt%C3%BC%20%C4%B0%C5%9Fleme%20Uygulamalar%C4%B1.pdf>)
10. Proje Kodları (<https://github.com/furkantoptass/sayisalGoruntuIslemeFinal>)