

# OCR TABANLI FİŞ TANIMA

Mert BİLGİÇ

Bilgisayar Mühendisliği Bölümü

Kocaeli Üniversitesi

[Bilgic.mert44@gmail.com](mailto:Bilgic.mert44@gmail.com)

## Özet

Uygulama,kullanıcı tarafından arayüz ile seçilen bir fişi tesseract ile okuyarak text haline çevir.Text parse edilerek arayüze yazılır.Parse edilen verilen database yazılır.Kayıtlı veriler için search işlemi ve sıralama işlemleri yapılabilmektedir.

## 1.Giriş

Uygulama ilk açıldığında kayıtlı veri var ise tabloda görüntülenir.Kullanıcı fiş okutmak istediğinde ilgili alanı tıklar.Açılan arayüz üzerinden fişi seçer ve seçilen fiş programa yüklenir.

Dosya seçildikten sonra dosyanın adresini path değişkenine atar.Adresteki resim üzerinde işlemler yapılmaya başlanır.Önce resim gri formata çekilir.Devamında resim gauss filtresinden geçirilerek resmin netliği artırılır. Resim bu hali işleme sokularak parse fonksiyonuna yönlendirilir. Eğer aranan alanlar resminde bulunursa alanlar database yazılır.Eğer aranan parametreler bulunamadıysa scaling fonksiyonu devreye girer ve resmin üzerindeki gürültü giderilir ve resim yeniden boyutlandırılır.Resim tekrardan Tesseract ile etkileşime geçer .İstenilen alanlar bulunduysa database yazılır.Bulunamadıysa elde edilen text ekrana basılır.

## 2.Temel Bilgiler

Projenin gelişiminde;

Programla dili olarak “Java” tümleşik geliştirme ortamı olarak “Apache NetBeans” kullanılmıştır. Uygulama içeriği olarak “Swing” kullanılmıştır.

## 3.Tasarım

Proje aşağıdaki başlıklar altında geliştirilmiştir.

### 3.1 Yazılım Tasarımı

#### 3.1.1 Genel Değişkenler

- **path** kullanıcı tarafından seçilen resmin adresini tutar.
- **plug** parse edilen verileri tuttuğumum arraylist değişkeni
- **result** seçilen resmin text formatı
- **con ve statement** database işlemlerini gerçekleştirdiğim değişkenler
- **content** parse ettiğimiz verileri tutan değişken

#### 3.1.2 Fonksiyonlar

Uygulamada bulunan fonksiyonlar aşağıda listelenmiştir

### 3.1.2.1 İşlem Fonksiyonları

- **allTras()** :Kullanıcı resim seçtikten sonra tüm işlemleri çalıştırır.
- **grayScale()** : Kullanıcıdan alınan resmi gri filtreden geçirir.
- **gauss()**:Kullanıcıdan alınan resmi gauss filtresinden geçirerek netleştirir.
- **demoParseText()** :Tesseract tarafından okunan resmi database yazılacak şekilde parse eder.
- **Rescale()** : Resmi yeniden boyutlandırır.
- **addplug()**: Parse edilmiş veriyi database kayıtlar eder.
- **list()** :Databaseden çekilen verileri listeler.

### 3.1.2.2 Arayüz Fonksiyonlar

- **setTable()** list fonksiyonundan gelen verileri arayüzdeki tabloya basar.
- **labelMouseClicked()** Resim seçme işlemi gerçekleştirildiğinde seçim arayüzünün açılmasını sağlar.
- **searchBtnActionPerformed()**:Search butonunu tıkladığımda database sorgu yapmamızı sağlar.
- **upBtnActionPerformed()**: Fiyat bazında büyükten küçüğe sıralama yapar.
- **downBtnActionPerformed()**: Fiyat bazında küçükten büyüğe sıralama yapar.

### 3.1.3 Görüntü İşleme

Kullanıcı tarafından seçilen resim tesseract ile daha iyi etkileşime girebilmesi için tesseractın dokümanındaki işlemleri sırasıyla inceledim. Resmi mat objesine çevirerek gri hali aldım.Devamında gauss filtresi yardımıyla resmin gürültüsünü azaltıp aynı zamanda fişin üstündeki yazıları belirginleştirdim.Eğer bu işlemler sonrasında fiş okunmaz ise resim yeniden boyutlandırılıp yeniden

tesseract ile etkileşime geçer.Eğer aranan alanlar bulunduysa tabloya yazılır.Eğer aranan alanlar bulunamadıysa okunan son halini ekrana basar.

İşlemden önce:



İşlemden sonra:



### 3.1.4 Arayüz

Program arayüzünde dört bölüm bulunmaktadır. Bölümler aşağıda numaralandırılmış halde sıralanmaktadır

### 3.1.4.1 Resim (Image) Seçimi

- Resim seçim işareti tıklanır. Açılan arayüz üzerinden dosyanın adresine gidilir ve resim tıklanır
- Resim onaylandıktan sonra görüntü işleme işlemleri başlar.

### 3.1.4.2 Parsing

- Seçilen resim parse işlemlerinden geçtikten sonra parse edildiği şekilde ekrana basılır.

### 3.1.4.3 Searching

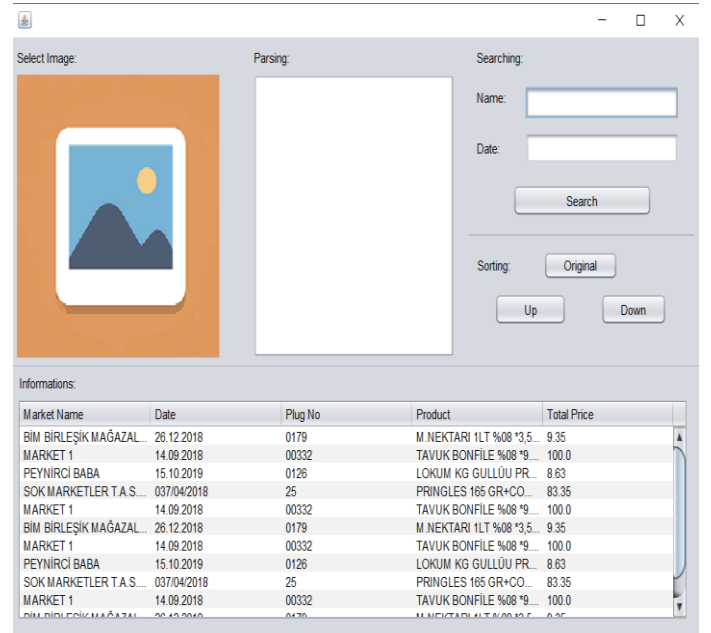
- Name ve Date alanlarından oluşur. Bu alanlardan ikisi veya biri doldurularak arama yapılır.
- Eğer boş bırakılırsa tabloda herhangi bir sonuç görünmez.

### 3.1.4.4 Sorting

- Original butonu ile search işlemi veya sıralama sonrası değişmiş tablo eski haline gelir.
- Up butonu fişleri büyükten küçüğe sıralar.
- Down butonu fişleri küçükten büyüğe sıralar.

### 3.1.4.5 Table

- Databasede kayıt tüm verilerin ekrana basıldığı kısımdır.



## 4. Kazanımlar

Kazanımlar aşağıda numralandırılmıştır.

### 4.1 OpenCV Kullanımı

Görüntü işlemenin ne olduğunun temelde hangi işlemlerin yapıldığını ve resmi opencv kullanarak nasıl iyileştirebileceği

### 4.2 Resim yazma

Üstünde işlem yapılmış bir resmin tekrar resmin üzerine yazmayı veya farklı bir yere nasıl yazıldığını öğrendim

### 4.3 Arayüz Tasarımı

Masaüstü bir uygulama için arayüz tasarımının nasıl yapıldığını java swing ile beraber öğrendim

### 4.4 Bufferedimage

Bufferedimage e kullanım yöntemlerini ve fonksiyonlarının kullanımı hakkında bilgi edindim.

## 5. Pseudo Code

Projemizde dört farklı sınıf bulunmamaktadır.

- Database
- Menu
- PlugData
- Transactions

Database sınıfında bağlantısı için gerekli değişkenleri final olarak tanımlandı.

Örn:public static final String user\_name="root";

Menu sınıfı kullanıcı için arayüz tasarımının yapıldığı ve aynı zamanda butonlara ve alanlara fonksiyonların atandığı kısımdır.

DefaultTableModel model; değişkeniyle tablomuza verileri basıyoruz. setTable(ArrayList<PlugData> result) ile databaseden çekilmiş verileri tablomuza basıyoruz.

labelMouseClicked(java.awt.event.MouseEvent evt) fonksiyonu yardımıyla alan tıklandığında resim seçimi için gerekli fonksiyonumuzu çağırıyoruz.

Menü sınıfının kullandığı fonksiyonlar Transcation sınıfı içinden geliyor. Transcation sınıfı şeklinde bir sınıf kullanmamızın sebebi projenin her yerinde tüm metodları kullanarak daha az satırda daha fazla iş yapabilmeyi sağlamak.Arayüz üzerinden resim seçildiğinde Transcation sınıfından public String allTras() metodu çağırılır ve resim işleme geçer.

grayScale(path);

gauss(path);

result=imageRead(path);

result = result.toUpperCase();ArrayList<PlugData>

plug = demoParseText(result)status=addplug(plug);

işlemlerinden geçer.Bu işlemler sırasında okunan resimin sonucu database yazılabilsin diye parse meotduyla etkileşime geçer.

for(String v : var2){

startLine = text.indexOf(v);

colone = text.indexOf(":",startLine); boolean reslt =cursorLine(text, colone+2)if(result){

endline=text.indexOf("

",colone+2);}else{endline=text.indexOf("\n",colone+2);}

key=text.substring((startLine),(colone)).trim()

value=text.substring((colone+1),endline).trim();

value = controlNumber(value);

parseResult+=key+" : "+value+"\n";

content.put(key, value);

Şeklinde ilgili alanlar liste kayıt edilir.Bu işlemten dönen sonu addplug() metoduna

yönlendirilir.query="INSERT INTO plug (ID,date,plugNo,product,total) VALUES (?,?,?,?)"

preparedstatement=con.prepareStatement(query);

preparedstatement.setInt(1,

id);preparedstatement.setString(2,date);preparedstat

ement.setString(3,no);preparedstatement.setString(

4,pt); preparedstatement.setFloat(5, total);

preparedstatement.executeUpdate(); ile database

yazma işlemi gerçekleştirilir.Gelen veriler burada

bazı fonksiyonlardan geçer yeni bir şirkettese alana ekleme yapılır.

public void addCompany(String name){ String

query="INSERT INTO company (Name) VALUES

(?)"; tüm işlem bittiğinde arayüzdeki parse alanına

veriler basılır.

## Kaynakça

[https://opencv-java\\_tutorials.readthedocs.io/en/latest/](https://opencv-java_tutorials.readthedocs.io/en/latest/)

<https://www.tutorialspoint.com/opencv>

<https://github.com/tesseract-ocr>

<https://stackoverflow.com/questions/4756268/how-to-resize-the-buffered-image-n-graphics-2d-in-java/4756906#4756906>

<https://www.geeksforgeeks.org/image-processing>

<https://medium.com/free-code-camp/getting-started-with-tesseract-part-i-2a6a6b1cf75e>

<https://stackoverflow.com/questions/9480013/image-processing-to-improve-tesseract-ocr-accuracy?rq=1>