**OCR TABANLI FİŞ TANIMA PROJESİ**

Onur GÜLER

Bilgisayar Mühendisliği Bölümü

Kocaeli Üniversitesi

[45onurguler45@gmail.com](mailto:45onurguler45@gmail.com)

**1.Özet**

Bu projenin amacı herhangi bir market veya alışveriş fişinin çekilen görüntüsünün uygulamaya yüklendiğinde, fişteki bilgileri okumak ve bu bilgileri veri tabanında kayıt altına alıp tabloda bu bilgileri işletme adına ve tarihine göre istenildiğinde arama yaptırabilmek.

**2.Giriş**

Telefondan ya da herhangi bir fotoğraf makinesi ile çekilmiş market veya restoran fişinin görüntüsünün çekilip, bu görseldeki fiş bilgisayar tarafından stringe çevrilmesi istenmektedir. Görüntü eğer net değilse kırışıklıklar bir tarafı gölgeli bir tarafı parlak olan düzgün okunamayan fişleri bilgisayar tarafından okunabilecek hale getirmeye çalışmak. Stringe çevrildikten sonra işletme adı, fişin tarihi, fiş nosu ; ürünlerin isimleri , kdv oranları, ürün fiyatları ve ürünlerin toplam fiyatlarını çekip veri tabanına kaydetmek. Bu tabloda işletme adı ve tarihe göre arama yapabilmek ve sıralama yapabilmek için bir masaüstü uygulaması yapılması istenmiştir. Bu uygulamayı yaparken amaç uygulamanın basit ve kullanışlı olması gerekmektedir.

Bu proje sayesinde projeyi yapacak kişinin veri tabanı konusunda daha önce edinmiş olduğu teorik bilgileri pratiğe dönüştürmesi, OCR sistemini anlaması ve istenenlere çözüm bulabilmesi amaçlanmaktadır.

**3. Temel Bilgiler**

Proje gelişiminde;

Tümleşik geliştirme ortamı olarak “ NetBeans IDE ” kullanılmıştır.

Ek olarak, MySQL Database’i Java uygulamasına bağlamak için MySQL Connector; fiş görselinin iyileştirilmesi için OpenCV ve okunabilmesi için tesseract kullanılmıştır.

**4. Yapılan Araştırmalar**

NetBeans de arayüz tasarımı için swing kütüphanesindeki jFrame, JTable , JTextField gibi araçların kullanımını, OpenCV ve Tesseract fonksiyonlarını anlamak için kaynakçada belirtilen linklerden yardım alınmıştır.

# Java’da Değişkenler ve Veri Türleri

Bir programlama dilinin temellerini değişkenler oluşturur. Değişkenler verilerin saklandığı birimlerdir.

|  |  |
| --- | --- |
| 1  2  3 | String deneme = "zinzin"; |

Değişkenler veri türü ve değer olmak üzere iki temel unsurdan oluşur. Örneğin; yukarıdaki örneğimizde değişken adımız deneme'dir. Veri türümüz string'dir. Yani metinseldir. Değerimiz ise zinzin'dir.

## Veri Türleri

Java’da değişkenler tanımlanırken herhangi bir veri türü belirtilmez. Hangi veri türü kullanıldığını değişkenin değerinden anlayabiliriz.

**ArrayList:**

Tıpkı Arra*y* gibi index yapısını kullanmaktadır. Ekleme, silme, sıralama, arama, ters çevirme gibi işlemler yapılabilir. ArrayList*leri* Array’den daha önde tutan dinamik bir yapısının olmasıdır. Array’lerde tanımlanma aşamasında dizinin boyutu belirtilmelidir, daha sonra program içinde dizinin boyutunu arttırmak zahmetli bir iştir. (Sürekli düzenli veya düzensiz bir şekilde arttırmak gerektirdiğinizi düşünürsek)**ArrayList** dinamik bir yapıya sahip olduğu için tanımlanırken boyutunun belirtilmesine gerek yoktur, çalışma anında dahi liste içine eleman eklenebilir, silinebilir, istenilen indise eleman eklenerek öteleme yapılabilir. Array’lerde bu mümkün değildir.

**ArrayList** listesinin elemanlarına erişebilmek için tıpkı Arraylerde olduğu gibi index numarası kullanılır. Ayrıca Arrayler sadece tanımlandıkları türde veri saklayabilirken, ArrayListler farklı türdeki verileri bir liste içinde barındırabilir.Dinamik olarak genişleyebilen tek boyutlu dizilere ArrayList denir.ArrayList kullanmak için kodumuzun en yukarısına

**import java.util.\*;**

eklememiz gerekmektedir.

**ArrayList dizi\_adi=new ArrayList();**

şeklinde kullanılır.

**String:** Metinsel veri türüdür.

|  |  |
| --- | --- |
| 1  2  3 | String deneme00 = "dedendeneme";      String deneme01 = "1234";      String deneme02 = '9Cdeneme'; |

Bir değişkeni string olarak tanımlayabilmek için = operatöründen sonra tırnak işareti içerisinde değişkenin değerini yazmalıyız. Yukarıdaki örnekte deneme00 adlı bir değişken tanımladık ve değerini zinzinzibidi şeklinde string olarak belirledik. deneme01 değişkeni de string'dir. Çünkü değeri sayısal olmasına rağmen tırnak işaretleri içinde yazdık. Tek tırnak ('') işareti içerisine yazılan değerler de string olarak algılanır.

**int ve float:** Sayısal veri türüdür.

|  |  |
| --- | --- |
| 1  2 | int deneme03 = 12;  // Tamsayı      floatdeneme04 = 2.3; // Ondalıklı sayı |

**Boolean:** Mantıksal veri türüdür.

|  |  |
| --- | --- |
| 1  2 | int deneme05 = true;      int deneme06 = false; |
|  |  |

Değişkenin değeri ya true (doğru) ya da false (yanlış) olabilir.

**Char**: Karakter veri türüdür.

|  |  |
| --- | --- |
| 1  2 | chardeneme05 = “s”;      int deneme06 = “r”; |

Bir değişkeni char olarak tanımlayabilmek için = operatöründen sonra tırnak işareti içerisinde değişkenin değerini yazmalıyız. Yukarıdaki örnekte deneme05 adlı bir değişken tanımladık ve değerini s şeklinde string olarak belirledik. deneme06 değişkeni de char'dır.Çünkü değeri sayısal olmasına rağmen tırnak işaretleri içinde yazdık. Tek tırnak ('') işareti içerisine yazılan değerler de char olarak algılanır.

**OpenCv:**

OpenCv, bir görüntü işleme kütüphanesidir. Platformdan bağımsızdır. Tüm işletim sistemlerinde çalışabiliyor.

**opencv\_core :** Temel fonksiyonlar, temel veri yapıları ve aritmetik fonksiyonların bulunduğu modüldür.  
**opencv\_imgproc :** Bu modülde temel görüntü işleme fonksiyonlarının bulunur.  
**opencv\_highgui :** Bu modülde resim ve video okuma/yazma işlemleri ve bazı basit kullanıcı arayüzlerinin bulunur.  
**opencv\_features2d :** Görüntü özellikleri ayırımı ve özellik karşılaştırma ile ilgili fonksiyonları barındıran modüldür.  
**opencv\_calib3d :** Bu modüşde kamera kalibrasyon, 2 boyutlu uzambilgisi tahmini ve stereo fonksiyonları bulunur.  
**opencv\_video :** Hareket çıkarımı, özellik işaretleme ve arkaplan ayırımı gibi fonksiyon ve sınıfları içeren modüldür.  
**opencv\_objdetect :** Bu modülde yüz tanıma, insan işaretleme gibi algılama fonksiyonlarının bulunur.

**5. Kazanımlar**

**Swing kütüphanesi**

Java’da swing kütüphanesindeki buton, textfield, table, textarea gibi yapıları kullanmayı öğrendim.

**OpenCV ve Tesseract**

OpenCv deki threshold ve cvt Color gibi fonksiyonları kullanarak resimin üstünde düzenlemeler yapmayı ve tesseract ile fotoğraftaki yazıları string olarak çekmeyi öğrendim.

**MySQL işlemleri**

Bu proje ile, bir programa bir veritabanı bağlantısı kurup öğrendiğimiz teorik SQL kodlarının bir programda veritabanından veri okuyabilmek ve güncelleme yapmak gibi SQL kodlarının işleyişini deneyimleme şansı buldum.

**6.Tasarım**

**a)Yalancı Kod(Pseudo)**

1.Uygulama çalıştırıldığında açılan pencerede jButton1 isimli Ekle butonuna basılırsa j1t1 isimli textfielda yazılan dosya adını al.

1.1 j1t1’de verilen resim uzantılı dosyayı ara

Eğer dosya bulunamıyorsa j1t2 isimli textfielda ‘Dosya bulunamadı’ stringini yaz ve 1’e geri dön.

Dosya bulunduysa resim dosyasını pencerede jLabel1 isimli label’da göster ve resimin boyutunu 300x300 boyutuna getir.

1.2 Daha sonra filtre clasındaki filt fonksiyonuna resim dosyasının ismini gönder.

1.2.1 Gönderdikten sonra OpenCv’deki cvtColor fonksiyonu ile gelen resimi gri tonlarına çevir.

1.2.2 Daha sonra GaussianBlur, threshold, morphologyEx gibi fonksiyonlarla resimdeki arka planı çıkartıp fişi algıla.

1.2.3 Fişi tesseractın daha iyi okuyabilmesi için resim boyutu küçükse boyutları 2 katına çıkart ve 300 dpi’a çevir.

1.2.4 300 dpi’a çevirdikten sonra tesseract için daha okunabilir bir hale gelmesi için threshold, morphologyEx fonksiyonlarını kullan ve resimi siyah beyaza indirge.

1.3 OpenCv işlemlerinden sonra dTest1 clasına işlenmiş resimi gönder.

1.3.1 Tesseract’a setLanguage fonksiyonu ile Türkçe dilini ekle.

1.3.2 Gelen resmi doOCR fonksiyonu ile gelen stringi Arayüze geri gönder.

1.4 Daha sonra gelen stringi regex clasına gönder ve regex clasında gelen stringi daha iyi parçalayabilmek için stra adlı ArrayList’ine satır satır bölerek at.

1.4.1 İlk 2 satıra bak ve ‘A.S.’ geçen satır varsa o satırı işletme adı olarak al eğer A.S. yı bulamazsan ilk satırı işletme adı olarak al.

1.4.2 Belirtilen tarih formatına uyan bir karakter dizisi varsa o karakterleri tarih olarak al.

1.4.3 Herhangi bir satırda Fiş No stringini bulursan fiş no için fiş nonun yanındaki sayıları al.

1.4.4 Daha sonra içinde % geçen satırları ara %’nin solunu ürün ismi, %’nin 2 karakter sağını kdv olarak, 3 karakter sonrasını ürün fiyatı olarak al.

1.4.5 Ürünlerden sonra Toplam stringini bulduysan sağındaki karakterleri toplam fiyat olarak al.

1.4.6 Bulunan bu verileri arayüze geri gönder.

1.5 j1t2 isimli textfielda regexten dönen bilgileri yaz.

1.6 Verileri tabloya kaydet butonuna basılırsa j1t2 isimli textfieldaki bilgileri veritanbanına kaydet, daha önce işletme adı, tarihi ve fiş nosu aynı olan bir kayıt varsa bu bilgileri kaydetme, farklı ise veritabanına kaydet ve tabloyu güncelle.

1.7 Ara isimli jButton2 ye basılırsa ta1 ve ta2 deki stringlere göre veritabanında arama yap aranan bilgilerde kayıt varsa tabloda göster yoksa tabloyu boşalt.

1.8 İşletme adına göre sırala ve Tarihe göre sırala butonlarına basılırsa tablodaki bilgileri işetme adına göre sırala butonun basılırsa işletme adına göre sırala, tarihe göre sırala butonuna basılırsa tarihe göre sırala ve arayüzde bulunan tabloyu basılan butona göre düzenle.

**b)Fonksiyonlar**

**tabsif()**

Arayüzdeki tabloya yeni veri eklendiği zaman ya da aram yapıldığında tabloyu temizlemek için kullanıldı.

**tabdol(String a)**

Arayüzdeki tablolar güncellendiğinde yeniden yazmak için kullandığım tablo fonksiyonu

**filt(String a)**

OpenCV işlemlerinin olduğu fonksiyon

**c) Yazılım Mimarisi**

Projeyi çalıştıran kullanıcı açılan pencerede en üstteki textfielda resim dosyasının ismin yazıp ekle butonuna bastığında resim varsa arayüzde göster. Ve resimi daha okunabilir bir hale getir daha sonra tesseracta gönder ve gelen stringi regex ile parçalayıp j1t2 textfieldında göster. Kullanıcı verileri tabloya kaydet butonuna basarsa alttaki tabloda o fiş yoksa fiş bilgilerini veritabanına kaydet ve tabloyu güncelle.

Sağ üstteki textfieldları doldurup ara butonuna basılırsa veritabanından arana verileri çekip tabloyu istenen bilgileri varsa güncelle.

Kullanıcı işletme adına göre sırala veya tarihe göre sırala butonlarından birine basarsa bastığı butona göre tablodaki verileri sıralı bir şekilde tabloyu tekrardan güncellle.

# 7. Kaynakça

jTable’daki satırları doldurma

<https://stackoverflow.com/questions/3549206/how-to-add-row-in-jtable>

SQL sorgu örnekleri

[https://www.mertcanilter.com.tr/mysql-sorgu- ornekleri-select-insert-into-update-delete-count-sum-like/](https://www.mertcanilter.com.tr/mysql-sorgu-%20%20%20ornekleri-select-insert-into-update-delete-count-sum-like/)

OpenCV fonksiyonları

<https://opencv-java-tutorials.readthedocs.io/en/latest/>

<https://docs.opencv.org/master/javadoc/index.html>

Tesseract fonksiyonları

<https://www.geeksforgeeks.org/tesseract-ocr-with-java-with-examples/>

Regex fonksiyonları

<https://ceaksan.com/tr/regex-regular-expressions-nedir/>

**8. Deneysel Sonuçlar**



