

**BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ**

**YAZILIM LABORATUVARI I, 2019-2020**

**PROJE 1**

Sinem Kulaç - 170201048

**Projenin Özeti**

Programımızın amacı aldığımız görüntü iyileştirme kullanarak daha okunabilir hale getirip tesseract algortimasına sokarak text olarak çıktı alıp bu çıktıyı parse işlemine sokup işletme adı,tarih,fiş no, toplam kdv ve toplam fiyat bilgilerini veritabanına kaydetmemizi sağlamak, ve arayüz tasarlayıp bu arayüzde veritabanında olan bilgileri kullanıcıya tablo halinde göstermektir. Çektiğimiz fotoğrafı

Veritabanından aldığımız tabloyla arama işlemini ve parse işlemini arayüzde tanımlayarak projemizi gerçekleştridik.

**1. Problem Tanımı**

Bu projede OCR tabanlı fiş tanıma uygulaması yaptık. Program dosyadan kaydedilen bir Fiş görüntüsü okuyacak, okunan görüntü (Open CV) Tesseract ile OCR işemine giricek, çıkan sonuçtan istenilen veriler parse edilecek, parse edilen veriler ekrana yazdırılacak ve ardından parse edilmiş veriler, işletme ismine göre veri tabanına eklenecek. Veritabanında kaydedelicek olan veriler ise Tarih, Fiş No, Ürünler KDV’ler Fiyatlar, Toplam Fiyat.

**2. Yapılan Araştırmalar**

Proje için öncelikle Tesseract kütüphanesi ile Open CV kurmamız gerekti. Zaten veritabanına ve gui tasarımına önceki projelerimizden aşina olduğumuzdan ağırlığı bunlara verdik. Kütüphaneleri düzgün bir şekilde kurduktan sonra ise okuma denemelerine giriştik. Gerekli ayarlarla sürekli oynayarak en ideal okuma için çalıştık. Okuma kısmı düzeldiğinde programa gui ekledik ve okduklarımız guide gösterme üstünde ve fotoğrafı gui üzerinden dosya yolu seçme üstünde uğraştık. En son veritabanıyla beraber parse işlemi için gerekli kodları ekleyip, gene küçük okuma problemlerini halledip programı son haline getirdik.

**3. Genel Yapı**

Proje JAVA programa dili üzerinden Netbeans adlı programla ve veritabanı ise MySql ile geliştirildi. Gui ekranından seçilen fotoğrafı gerekli işlemlere sokup parse ettikten sonra parse edilmiş veriler ekranda görünüyor ve veri tabanına kaydediliyor. Veritabanında arama yapılarak önceden kaydedilmiş veriler arasından aranan veri bulunabiliyor.

**4. Kaynaklar**

1.<https://medium.com/cashify-engineering/improve-accuracy-of-ocr-using-image-preprocessing-8df29ec3a033>

2.<https://www.tensorscience.com/ocr/optical-character-recognition-ocr-with-python-and-tesseract-4-an-introduction>

3.<https://docs.opencv.org/3.0-beta/doc/py_tutorials/py_imgproc/py_table_of_contents_imgproc/py_table_of_contents_imgproc.html>

4.<https://answers.opencv.org/question/120742/find-contour-opencv-java/>

5.<https://answers.opencv.org/question/61802/change-size-while-using-dilate-in-java-opencv/>

6.<https://stackoverflow.com/questions/48868951/how-to-update-jlabel-text-in-swing>

7.<https://stackoverflow.com/questions/48868951/how-to-update-jlabel-text-in-swing>

**5. Kazanımlar**

Bu proje sayesinde kendimizi veritabanı ve gui tasarımında geliştirdik ve ayrıca görüntü okuma konusunda yeni şeyler öğrendik. Deneme yanılma ile görüntü iyileştirme yaparak kodların nasıl çalıştığını ve nasıl daha kolay kullanaileceğimizi keşfettik.

**6.Psuedocode**

* veritabanını ve guiyi başlat
* gui üstünden fotoğraf seçilmesini bekle
* seçilen fotoğrafı guide göster
* tanımlanan fotoğrafta iyileştirmeleri yap
* iyileştirmeleri yapılmış halde kaydet
* kaydedilen fotoğrafı oku
* okunan veriden şirket ismini oku
* stringe kaydet
* okunan veriden tarihi oku
* stringe kaydet
* okunan veriden saati oku
* stringe kaydet
* okunan veriden fiş nonu oku
* geçici stringe kaydet
* int değerine çevir ve kaydet
* okunan veriden kdv yi oku
* geçici stringe kaydet
* float değerine çevir ve kaydet
* okunan veriden toplam fiyatı oku
* geçici stringe kaydet
* float değerine çevir ve kaydet
* Arayüzde string olarak tanımlanan tarih ve işletme ismi için değer gir
* Arama fonksiyonlarına stringi gönder ve gönderilen stringi tabloda sırala
* kaydedilen verileri veritabanına gönder

**7. Karşılaştığımız Zorluklar**

Tesseractı kurmak bizim için zorlayıcı kısımlardan biri oldu. Birçok hata alarak bunları çözmeye çalıştık. Türkçe dili tessdata’sını ekleme işlemini nasıl yapacağımızı öğrendikten sonra 2 gün içerisinde Tesseract indirme işlemini hallettik.

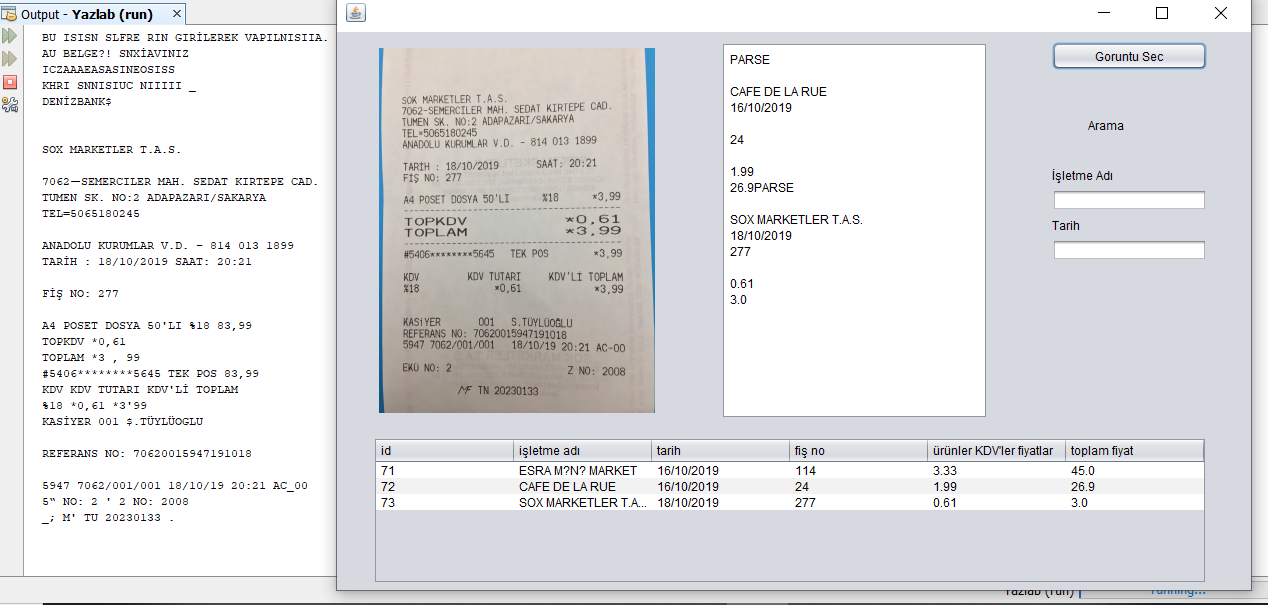
Görüntü iyileştirme işlemi için beyaz arka plan siyah yazı daha iyi sonuç veriyormuş bu bilgiyi öğrendikten sonra birçok iyileştirme versiyonu denedik ancak en iyi sonucu siyah arka plan beyaz yazıda alabildik dolayısıyla tesseract’a da görüntüyü bu şekilde gönderdik.

Birçok iyileştirme tekniği denememize ragmen bu konuda başarılı olduğumuzu söyleyemeyiz.Görüntü iyileştirme en zorlandığımız ve eksik kaldığımız konu oldu.Python diliyle gördüğümüz örnekler çok daha basitken java diliyle bunları yapmakta zorlandık.

Tarih bilgisini date formatında kaydetmeye karar verdik ancak fişlerde bulunan tarih formatı gün-ay-yıl şeklinde iken veritabanında yıl-ay-gün şeklindeydi.Dolayısıyla bulduğumuz değeri ters çevirirerek date formatında kaydetmeyi başardık.

Fişten okuduğumz tarih bilgisini en başta string olarak değil date formatında kaydetmeyi planladık. Araştırdıktan sonra stringten doğru şekilde date formatına çevirdik fakat bu seferde veritabanına date formatıyla kaydetmede sıkıntılarla karşılaştık. Bu durumda eski string formatına geri dönmek zorunda kaldık. Util.date sql.date’e çevrilemediğinden bu sıkıntıyla karşılaştık.

**Proje İçinden Görseller**

****