

OCR TABANLI FİŞ TANIMA UYGULAMASI

Muhammed Emre KARA

160202094

Bilgisayar Mühendisliği Bölümü
Kocaeli Üniversitesi

mailemrek@gmail.com

1. Özet

OCR Tabanlı Fiş Tanıma Uygulaması. Fiş tanıma ve sorgulama uygulamasını desktop için JAVA’da kodladım (Maven Proejct). Veri tabanı olarak ise MySQL kullandım.

2. Giriş

Kullanıcıdan okutmak istediği fişleri dosya seçme menüsü içerisinde seçmesi beklenir. Daha sonra belirtilen fiş veya fişler Tesseract ile OCR işlemine sokulur ve stringler ayıklanır elde edilen tüm veri ekrana yazdırılır. Daha sonra bu ayıklanan metinler içerisinde **İşletme Adı, Fiş No, Tarih, Ürünler ve Toplam** bölümleri parse edilir, veritabanına kayıt edilir.

Kullanıcı Veritabanına kaydedilmiş veriler arasında sorgu yapabilir. İşletme adına veya tarihe göre arama yapabilir veya tüm verileri görebilir. Kullanıcıya sunum hem varolan ana ekrana yerleşik olarak hem de yeni açılacak bir ekranda mümkündür. Herhangi bir ürünün ayrıntılarına tıklandığında yine ek bir pencere içinde ürünler gösterilir.

3. Temel Bilgiler

Projeyi gerçekleştirirken kullandığımız teknolojiler aşağıda verilmiştir.

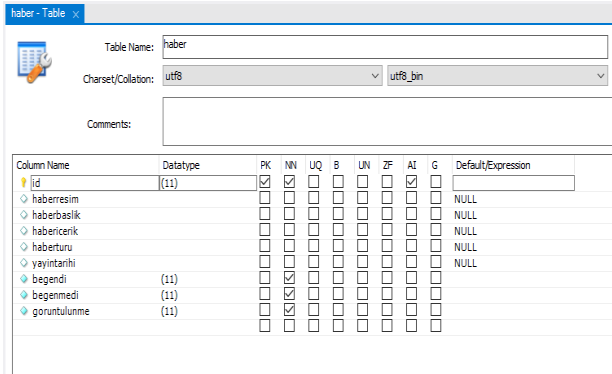
JAVA: Java, [Sun Microsystems mühendislerinden James Gosling](#) tarafından geliştirilmeye başlanmış açık [kodlu](#), nesneye yönelik, zeminden bağımsız, yüksek verimli, çok işlevli, yüksek seviye, adım adım işletilen (yorumlanan-interpreted) bir [dildir](#).

Veritabanı : Veri tabanı, birbiriyle ilişkili bilgilerin bir arada tutulduğu sistemlerdir. Veri tabanı üzerinde bilgiler düzenli bir şekilde tutulduğu için bu bilgilere erişim, bilgilerin yönetimi, yeni bilgi ekleme ve bilgilerin güncellenmesi çok kolay ve çok hızlıdır. Veri tabanı hayatımızın her alanında yer almaktadır.

Bellek alanı israfını önler ve veri bütünlüğü sağlamamıza yardımcı olur.

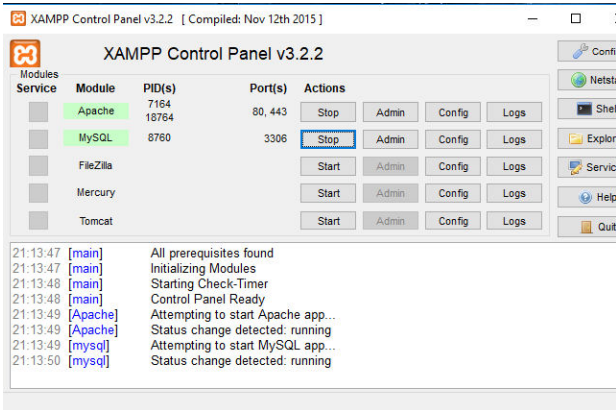
OCR : ‘Optik Karakter Tanıma’ veya **‘Optik Karakter Okuyucu’** olarak adlandırılan bu yazılımlar, taranan bir belgede, bir fotoğrafta, el yazısıyla yazılmış bir metinde yer alan yazıların bilgisayar ortamına aktarılmasını sağlamak için geliştirilmiş yazılımlardır. Normal şartlarda bilgisayara tarayıcı vasıtasıyla aktarılan grafik dosyaları, bilgisayar tarafından resim olarak algılanır. Bu grafik dosyadaki yazılar **‘OCR’ yazılımları** tarafından çözümlenip metin dosyasına çevrilebilir. Böylelikle klavye kullanmaya gerek kalmadan metin bilgisayara aktarılacaktır.

MYSQL : MYSQL altı milyondan fazla sistemde yüklü bulunan çoklu iş parçacıklı, çok kullanıcı, hızlı bir açık kaynak veri tabanıdır.



Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G	Default/Expression
id	(11)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
haberesim		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
habereslik		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
habericik		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
haberturu		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
yayintarihi		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
begendi	(11)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
begenmedi	(11)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
goruntulume	(11)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL

XAMPP : Kolay bir şekilde yerel web sunucusu oluşturulmasını sağlayan, bir apache dağıtımdır. Xampp, kelimesinde her harfin bir anlamı vardır. X:çapraz platform, A:apache, M:mysql, P:php ve son P harfi de perl anlamına gelmektedir.



Service	Module	PID(s)	Port(s)	Actions
Apache	Apache	7164 18764	80, 443	Stop Admin Config Logs
MySQL	MySQL	8760	3306	Stop Admin Config Logs
FileZilla	FileZilla			Start Admin Config Logs
Mercury	Mercury			Start Admin Config Logs
Tomcat	Tomcat			Start Admin Config Logs

21:13:47 [main] All prerequisites found
21:13:47 [main] Initializing Modules
21:13:48 [main] Starting Check-Timer
21:13:48 [main] Control Panel Ready
21:13:49 [Apache] Attempting to start Apache app...
21:13:49 [Apache] Status change detected: running
21:13:49 [mysql] Attempting to start MySQL app...
21:13:50 [mysql] Status change detected: running

Java Maven Project:

Maven *build, reporting, dependency management* ve *documentation* işlemlerini gerçekleştirir.

Build işleminin iki yönünü ele alır: projenin nasıl build olacağını ve projenin dependency'lerini tanımlar. Build işlemi için kendisinden önceki araçların aksine kabul edilen bir düzen kullanır ve sadece istisnaların yazılması gerekir.

Dependency management olarak Maven library ulaşmak için sırasıyla bulana kadar local, central ve uzak depolara bakar, indirilmesi gerekiyorsa indirir. Bu mekanizma sayesinde geliştiriciyi büyük bir iş yükünden kurtarır.

Konfigürasyon için **pom.xml** dosyasını kullanır. Bu dosya projeyi build etmek için gerekli bütün bilgileri içerir.

Maven başlıca Java projeleri için kullanılsa da C#, Ruby, Scala gibi farklı diller için de kullanılabilir.

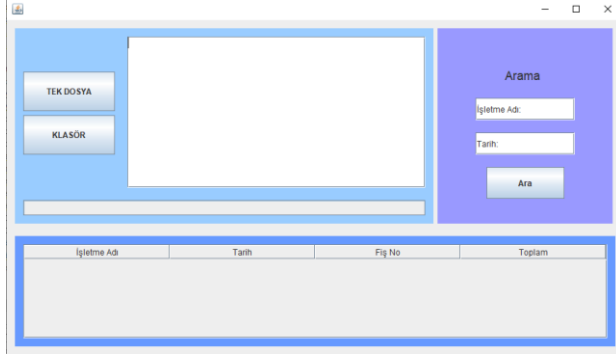
Tesseract :

Tesseract, çeşitli işletim sistemleri için optik karakter tanıma motorudur. Apache License, Version 2.0 altında yayımlanan ücretsiz bir yazılımdır ve 2006'dan bu yana Google'ın geliştirilmesine destek olmuştur. 2006'da Tesseract, açık kaynak kodlu OCR motorlarından biri olarak kabul edildi.

Benim Projemde:

GirisFrame:

Kullanıcı arayüzü için java JFrame sınıfıdır. Tüm GUI işlemleri burada bulunur. Tasarımı aşağıda gösterilen şekildedir:



Kullanıcı tek bir imaj dosyası veya imaj dosyalarını içeren bir klasörü vererek Tesseract ile OCR işlemini başlatır. Tesseract'den dönen string ekranın ortasında bulunan metin kutusunun içerişe yazdırılır.

Sağ taraftaki arama kutusu kullanılarak kullanıcı isterse tüm sonuçları listeler veya İşletme Adı, Tarih ikilisinin kombinasyonlarında aramalar gerçekleştirebilir. Bu aramaların sonucu isterse aşağıda yer alan tablo içerisinde, isterse de ek bir pencerede gösterilir. Listelenen fiş bilgilerinin herhangi birine çift tıklayarak ek bir pencere içinde ürünler listelenebilir.

Filtration:

Tesseract'den elde edilen metinlerin içerisinde anlamlı kısımların ayırt edilmesi için kullanılır.

Tarih, Fiş No , Toplam ve Ürünler için Java REGEX kullanılır.

Tarih:

```
Pattern.compile("\\d{2}\\p{Punct}\\d{2}\\p{Punct}\\d{4}")
```

Fiş No:

```
Pattern.compile("[fF][uU][sS]$"?[nN][oOuUüÜO] ?\\: *\\d*")
```

Ürünler:

```
Pattern.compile("\\n.*[x*X ]\\d*,\\d*");
```

İşletme Adı:

Java regex kullanılmaz, verilen textin ilk 2 satırı kullanılır.

ConDB:

Bu sınıfta veritabanına işlemleri bulunur. Bunlar: **Kayıt(Insertion), Sorgu(Selection) ve Bağlantı(Connection)** işlemleridir.

MySQL bağlantısını sağlamak için **JDBC MySQL Connector** kullanılır.

Sorgular:

SelectAll():

Veritabanında bulunan tüm kayıtları döndürür:

```
"SELECT * FROM tesseractdb.liste "
```

Select İşletmeAdi():

İşletme Adına göre sorgu çalıştırır.

```
"SELECT * FROM tesseractdb.liste WHERE isletmeAdi LIKE ?"
```

SelectTarih():

Tarihe göre sorgu çalıştırır.

```
"SELECT * FROM tesseractdb.liste WHERE tarih = ?"
```

SelectBoth():

Hem İşletme Adına hem Tarihe göre sorgu çalıştırır.

```
"SELECT * FROM tesseractdb.liste WHERE isletmeAdi = ? AND tarih = ?"
```

SelectUrunler:

Veritabanında verilen fiş bilgileri için ürünleri döndürür.

```
"SELECT * FROM tesseractdb.products WHERE isletmeAdi LIKE ? AND fisNo LIKE ?"
```

Veri Tabanının Yapısı:

Liste:

#	Name	Datatype	Length/Set	Unsigned	Allow NULL	Zerofill	Default
1	id	INT	11	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	AUTO, INCREMENT
2	isletmeAdi	VARCHAR	200	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	No default
3	tarih	VARCHAR	20	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	No default
4	fisNo	VARCHAR	20	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	No default
5	toplamFiyat	VARCHAR	50	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	No default

Products:

#	Name	Datatype	Length/Set	Unsigned
1	id	INT	11	<input checked="" type="checkbox"/>
2	isletmeAdi	VARCHAR	250	<input type="checkbox"/>
3	fisNo	VARCHAR	20	<input type="checkbox"/>
4	urun	VARCHAR	500	<input type="checkbox"/>

Web Servis Database Bağlantısı

Kaynakça

- 1) <https://docs.oracle.com/javase/7/docs/api/java/util/regex/Pattern.html>
- 2) <https://www.regular-expressions.info/posixbrackets.html>
- 3) <https://www.geeksforgeeks.org/regular-expressions-in-java/>
- 4) <https://www.geeksforgeeks.org/regular-expressions-in-java/>
- 5) <http://www.mysqltutorial.org/mysql-date/>