

Görüntü İşleme ile Otomasyon Ara Rapor 1

Danışman: Mustafa Dağtekin

Öğrenci: Anıl Ertürk 1358130030

Veri toplama/giriş problemi: Modern bilgisayarlardan önce şirketlerde bottleneck insanlar(veri girişi) değildi. Ama günümüz dünyasında şirketlere baktığımız zaman çok büyük sayıda müşteriye hizmet verildiğini görüyoruz. Günümüzde bilgisayarlar çok hızlı olmasına rağmen malesef bottleneck bu sefer insan ve bilgisayar arasındaki interface oldu. Turkcell ile birlikte çalışan aile şirketimizden bu konudaki yenilik isteğinin çok yüksek olduğunun birinci elden farkındayım. Askeriye ve cezaevleri gibi yerlerde satılan iletişim hatları/kartları ile ilgili formlar dolduruluyor ve bu formlar şirkete elektronik ortamdan gönderiliyor. Daha sonra şirketin veri giriş departmanı bu formlardaki bilgileri tek tek Turkcell'in sistemine giriyor. Bu da çok zaman alıyor. Bu konuda yenilik için özel sektörde büyük bir istek var ve şirketler sürekli olarak yaptıkları işlemleri hızlandırmak için Turkcell'deki mühendislerin sistemlerini geliştirmesini istiyor.

Yukarıda sadece bir adet örnekten bahsettik. Fakat görüntü işleme ile uygulamamızı farklı şekillerde configure ederek farklı sorunlara çözüm bulabiliriz. Mesela bir derse gelen öğrenciler, öğrenci kartlarını bir kameraya gösterip yerlerine oturabilir. Bu sayede aşırı derecede yavaş ilerleyen tek tek bir adet sayfaya imza atma işlemini hızlandırmış oluruz. Ayrıca bu sayfayı toplayan öğretmen dersten sonra elektronik bir ortama tek tek girmek zorunda kalacak verileri. Görüntü işleme ile otomasyon kullandığımızda hem imza atma öğrenci sınıfa girerken otomatik olacak, hem de imzaların elektronik ortama geçişi otomatik olacak. Bu sayede sistemin birden çok noktasındaki sorunları çözmüş olduk.

Proje özeti: Derse gelen öğrencilerin bir kağıda imza atmayacak. Programımız bir pencerede kameranın aldığı video'yu gösterecek. Kullanıcı öğrenci kartını kameranın önüne getirip fotoğraf çekecek. Program görüntüyü kaydedecek.

Kaydedilen görüntü daha sonra programın OCR kısmı tarafından açılacak ve içerisindeki text alınacak. Daha sonra bu text geçerli bir bilgisayar mühendisliği öğrenci kartına aitmi değilmi karar verilecek. Eğer geçerli bir bilgisayar mühendisliği öğrencisi kartı ise, grafik arayüz "numaranız 138130030 ise ilerle tuşuna basınız" diyecek. Eğer geçerli bir bilgisayar mühendisliği kartı bulunamaz ise grafik arayüz "tekrar deneyiniz" diyecek.

Kartı yanında olmayan kişiler için manuel bir giriş bölümüde var programda. Fakat manuel giriş sadece şifreyi bilen öğretmen tarafından yapılabilir. Ayrıca görüntü işleme ile yaptığımızdaki gibi, manuel giriştede hem öğrenci numarasını hemde manuel giriş şifresini kontrol ediyoruz.

Proje detayı: Proje bu aşamada üç sınıftan oluşuyor. "Camera", "Menu", ve "OCR". Ana sınıfımız Menu sınıfı. Menu sınıfı hem grafik arayüzümüzün hemde main metodumuzun olduğu sınıf. Normalde grafik arayüz programlarında main metoduna kod konulmaz. Ama ben program açılır

açılmaz kullanıcı inputu olmadan video feed başlasın istediğim için main metodu sonunda Camera sınıfımızın objesi olan cam objesinin run methodunu yeni bir thread'de başlattım.

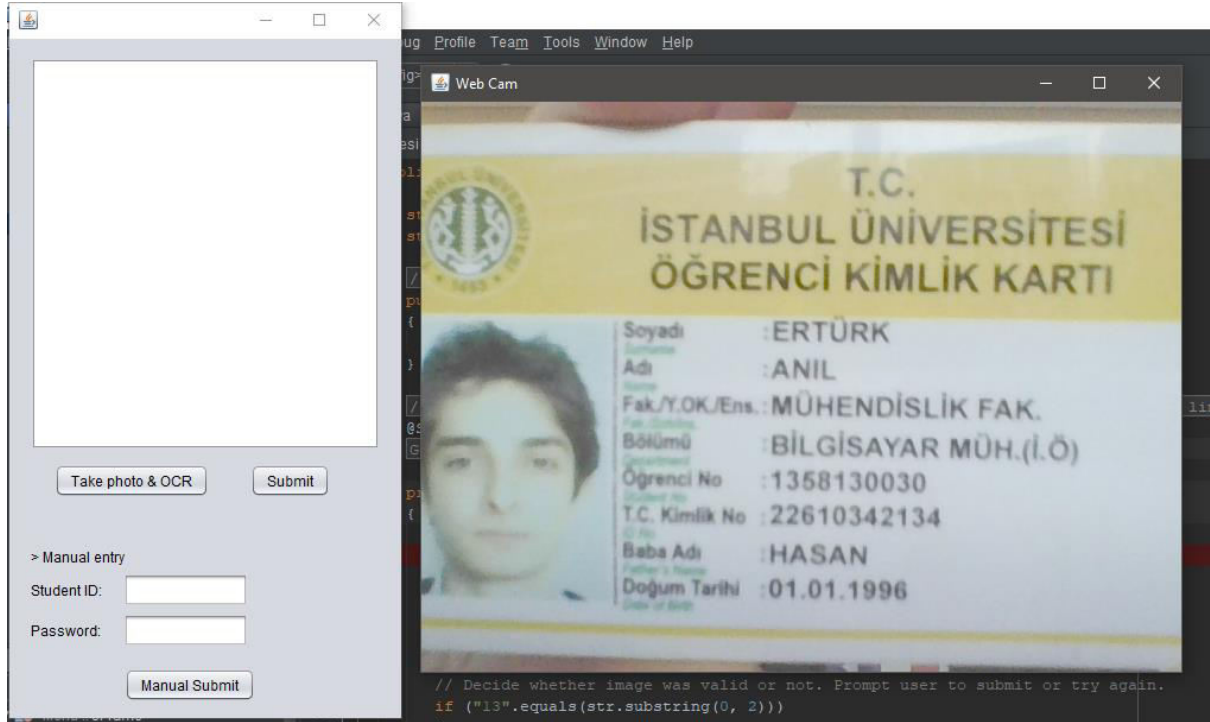
```
// Starting video feed as soon as program starts.  
new Thread(new Runnable() {  
    public void run() {  
        cam.run();  
    }  
}).start();
```

Bu yeni bir pencerede webcam yayınına başlıyor. Bu yayın sürekli olarak webcamden frame alıyor. Bu frameleri image'a dönüştürüyor ve ekranda gösteriyor.

```
// Objects that will be used for video display and taking photos.  
FrameGrabber grabber = new OpenCVFrameGrabber(0);  
OpenCVFrameConverter.ToIplImage converter = new OpenCVFrameConverter.ToIplImage();  
IplImage img;
```

```
public void run()  
{  
    try  
    {  
        /* When application starts, this method is called automatically from  
        main method of Menu class. So grabber is started before we start taking  
        photos. */  
        grabber.start();  
        while (true)  
        {  
            Frame frame = grabber.grab();  
  
            img = converter.convert(frame);  
  
            canvas.showImage(converter.convert(img));  
  
            //Thread.sleep(16);  
        }  
    }  
    catch (Exception e)  
    {  
        e.printStackTrace();  
    }  
}
```

İkinci bir pencerede ise grafik arayüz kontrol panelimiz var.



Kontrol panelimizdeki Take photo & OCR butonuna bastığımızda photoAndOcr butonumuzun metodu çalışıyor.

```
private void photoAndOcrActionPerformed(java.awt.event.ActionEvent evt)
{
    // Take a photo and save it as id.jpg.
    cam.takeAPhoto();

    /* OCR the id.jpg. Return student number if photo is valid. Return error
    message if it is invalid. */
    String str = ocr.analyze();

    // Decide whether image was valid or not. Prompt user to submit or try again.
    if ("13".equals(str.substring(0, 2)))
    {
        // Image was valid.
        jTextPanel.setText(jTextPanel.getText() + "If your student number is "
        + str + ", press Submit to continue. If not, take another photo.\n\n");
    }
    else
    {
        // Image was not valid.
        jTextPanel.setText(jTextPanel.getText() + str + "\n\n");
    }
}
```

Bu metod;

1-camera objemizin takeAPhoto metodu çalışıyor. Bu metod webcam'den bir frame alıyor. Bu frame'ı bir image'a dönüştürüyor. Daha sonra bu image'ı id.jpg olarak kaydediyor.

```
// Objects that will be used for video display and taking photos.
FrameGrabber grabber = new OpenCVFrameGrabber(0);
OpenCVFrameConverter.ToIplImage converter = new OpenCVFrameConverter.ToIplImage();
IplImage img;
```

```
public void takeAPhoto()
{
    try
    {
        //grabber.start(); // Causes error. So i moved 3 declaration to field

        Frame frame = grabber.grab();

        img = converter.convert(frame);

        cvSaveImage("id.jpg", img);

        //canvas.showImage(converter.convert(img));

        //Thread.sleep(1000);
    }
    catch (Exception e)
    {
        e.printStackTrace();
    }
}
```

2- ocr objemizin analyze metodunu çalıştırıyor ve sonucu str Stringine atıyor. (OCR sınıfı çok büyük olduğu için sadece en son işin yapıldığı kısmı koydum.) Bu metod tesseract API'ını hazır hale getiriyor ingilizce dilini okuyacak şekilde. id.jpg'i alıyor ve içindeki text'i çıkarıyor. Kullanılan kaynaklar yok ediliyor. Elde ettiğimiz text'i alıp word'lere bölüyoruz. Bu word'ler üzerinde tek tek dolaşp 10 harf ve üzerinde olanları arıyoruz. 10 harf ve üzerinde olan word'ler üzerinde birkaç işlem yapıyoruz. Öncelikle MUHENDISLIK veya BILGISAYAR textlerine eşitler mi değiller mi ona bakıyoruz. Bu bizim securityFlag değişkenimizi false'dan true yapıyor. Daha sonra karşılaştığımız 10 harf ve üzerinde word'lerden birisi 13 ile başlarsa ve securityFlag değişkenimiz true ise, bu 13 ile başlayan word'ün ilk 10 karakterini Menu sınıfına geri döndürüyoruz.

```
// Splitting text to String array elements.
String[] studentCardData = outText.getString().split(" ");

// Parsing String array elements
for (String data : studentCardData)
{
    try
    {
        /* I am not checking to see if the string is 10 characters long
        because student number is read from image with some extra
        characters attached to it. */
        if (data.length() < 10)
        {
            continue;
        }

        // Secret security code.
        if (data.contains("MUHENDISLIK") || data.contains("BILGISAYAR"))
        {
            securityFlag = true;
        }

        String subString = data.substring(0, 2);

        // We already did length test above.
        if ("13".equals(subString) && securityFlag == true)
        {
            securityFlag2 = true; // Valid student number.
            // Getting rid of extra characters.
            studentNumber = Integer.parseInt(data.substring(0, 10));

            securityFlag = false; // Resetting object to be used again
            return data.substring(0, 10);
        }
    }
}
```

3- Menu sınıfında photoAndOcr butonumuzun metoduna geri döndük. Şimdi str Stringimizin ilk iki hanesi 13 mü değilmi ona bakıyoruz. Eğer 13 ile başlıyorsa kullanıcıya “öğrenci numaranız bu ise submit tuşuna basın” diyoruz. Eğer str Stringimiz başka iki karakter ile başlıyorsa kullanıcıya “tekrar deneyiniz” diyoruz.

```
// Decide whether image was valid or not. Prompt user to submit or try again.
if ("13".equals(str.substring(0, 2)))
{
    // Image was valid.
    jTextPanel.setText(jTextPanel.getText() + "If your student number is "
        + str + ", press Submit to continue. If not, take another photo.\n\n");
}
else
{
    // Image was not valid.
    jTextPanel.setText(jTextPanel.getText() + str + "\n\n");
}
```

Bu aşamada kullanıcı kendi öğrenci numarasını ekranda gördüyse submit tuşuna basmalı. Eğer kullanıcı kendisine ait olmayan bir öğrenci numarası veya tekrar deneyiniz mesajı görürse kartının tekrar fotoğrafını çekmeli. Bu aşamada program kullanıcıya kartı kameradan yaklaşık 10 santimetre uzaklıkta tutması gerektiğide söylüyor.

Eğer kullanıcı submit butonuna basarsa daha bitmemiş veritabanı işlemlerini yapan metod çağırılacak ve kullanıcıya işleminiz yapılmıştır cevabı verilecek.

```
private void submitActionPerformed(java.awt.event.ActionEvent evt)
{
    if (ocr.securityFlag2 == true)
    {
        /* !! TO DO !! Create a class for database stuff, create a static object in Menu class
        field. Then call the method of that object here with student number as argument*/

        // Submit was successful.
        jTextPanel.setText(jTextPanel.getText() + "Submitted " + ocr.studentNumber
        + "\n\n");
    }
}
```

Eğer bir öğrenci kartını unutmuşsa, öğretmen öğrencinin numarasını bir şifre kullanarak manuel olarak girebilir. Kullanıcıya kolaylık olması için aşağıdakileri programa kattım;

- 1- Student ID'yi girdiğinizde fare ile Password textfield'ına tıklamanıza gerek yok. Student ID'yi girdikten sonra enter'a bastığınızda direk Password textfield'ına gideceksiniz.
- 2- Password'ü girdiğinizde fare ile Manual Submit butonuna tıklamanıza gerek yok. Password'ü girdikten sonra enter'a bastığınızda direk Manual Submit butonuna basılacak, doldurduğunuz iki textfield boşaltılacak ve son olarak tekrar otomatik olarak Student ID textfield'ına tıklanılacak. (Üst üste giriş yapmak kolay olsun diye.)

> Manual entry

Student ID:

Password:

```
// Pressing enter in studentNumber textfield takes you to password textfield
private void studentNumberActionPerformed(java.awt.event.ActionEvent evt)
{
    password.requestFocus();
}

// Pressing enter in password passwordfield invokes manualSubmit button.
private void passwordActionPerformed(java.awt.event.ActionEvent evt)
{
    manualSubmitActionPerformed(evt);
}
```

Program öğrenci numarası 13 ile başlıyormu, uzunluğu 10 karaktermi, sadece rakamlardan oluşuyormu ve şifre dagtekin13'mü bakar. Eğer bunlar sağlanıyorsa daha bitmemiş veritabanı işlemlerini yapan metod çağırılacak ve kullanıcıya işleminiz yapılmıştır cevabı verilecek.

```

private void manualSubmitActionPerformed(java.awt.event.ActionEvent evt)
{
    try
    {
        if ("13".equals(studentNumber.getText().substring(0, 2)) && studentNumber.
            getText().length() == 10 && "dagtekin13".equals(password.getText()))
        {
            // Checking if studentNumber contains non digit characters.
            Integer.parseInt(studentNumber.getText());
            /* !! TO DO !! Create a class for database stuff, create a static object in Menu class
            field. Then call the method of that object here with student number as argument*/

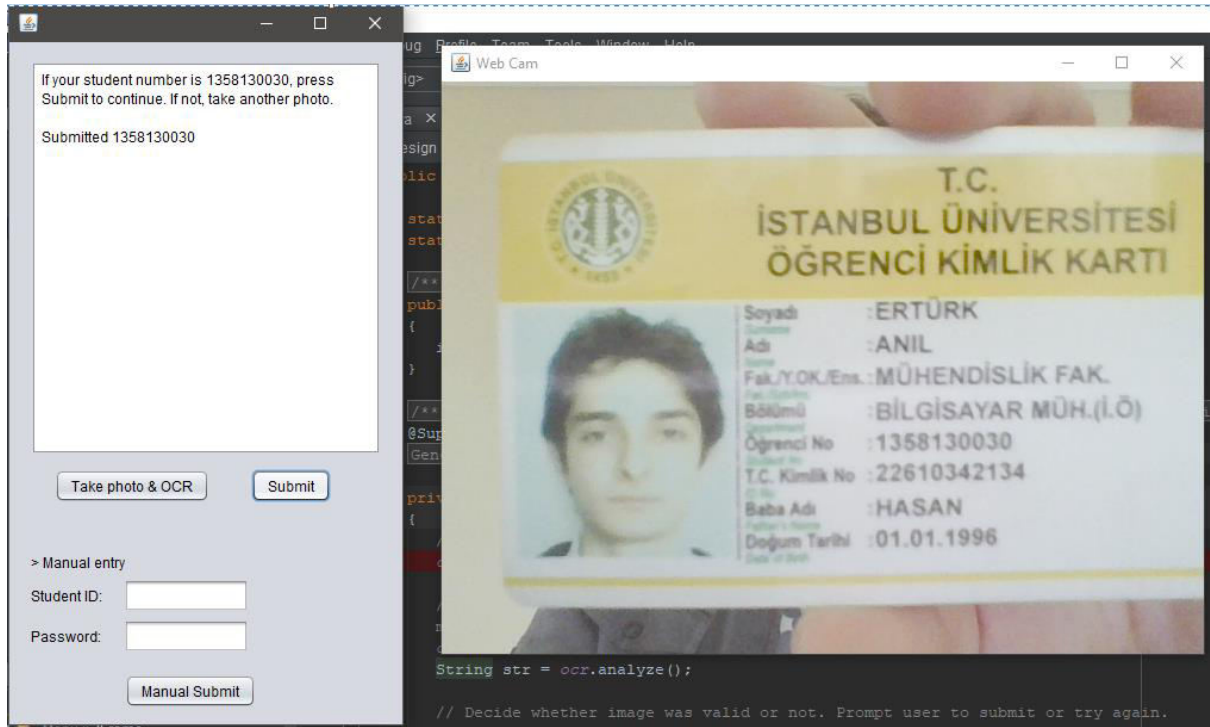
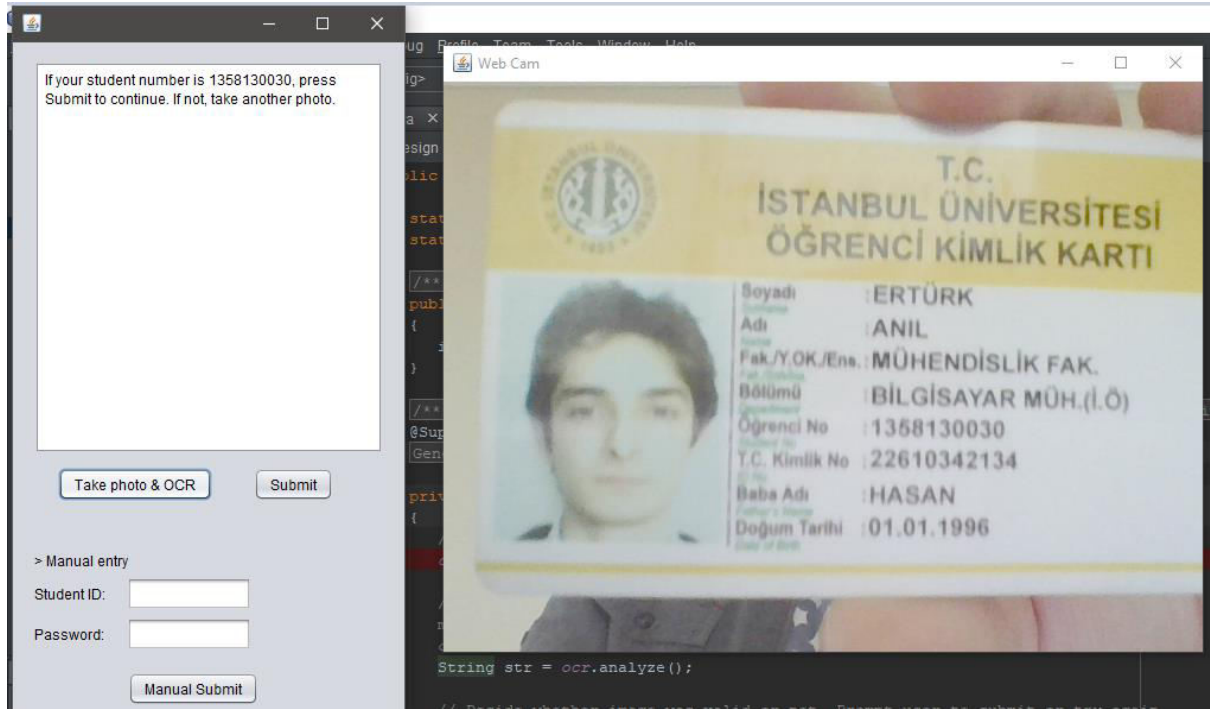
            // Submit was successful.
            jTextPanel.setText(jTextPanel.getText() + "Submitted "
                + studentNumber.getText() + "\n\n");
        }
        else
        {
            jTextPanel.setText(jTextPanel.getText() + "Student number or password "
                + "is wrong.\n\n");
        }

        // Reset both fields and focus on studentNumber field.
        studentNumber.setText("");
        password.setText("");
        studentNumber.requestFocus();
    }
    catch (StringIndexOutOfBoundsException | NumberFormatException e)
    {
        jTextPanel.setText(jTextPanel.getText() + "Student number or password "
            + "is wrong.\n\n");

        // Reset both fields and focus on studentNumber field.
        studentNumber.setText("");
        password.setText("");
        studentNumber.requestFocus();
    }
}

```

Proje çalışırken alınmış ekran görüntüleri:



Try again. Make sure your student ID card is about 10 centimeters from the camera.

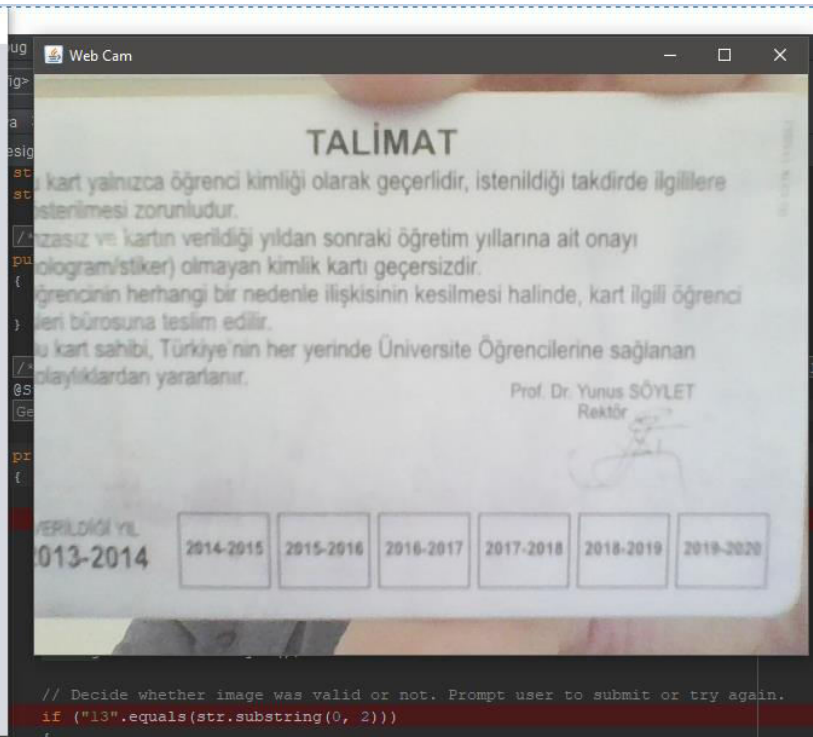
Take photo & OCR Submit

> Manual entry

Student ID:

Password:

Manual Submit



Try again. Make sure your student ID card is about 10 centimeters from the camera.

You need to take a photo with a valid student ID card first.

Take photo & OCR Submit

> Manual entry

Student ID:

Password:

Manual Submit

