

Şekil Tanıma Dersi 1. Ödev Raporu

Yazan : Şadi Evren ŞEKER

(www.sadievrenseker.com ,ses@sadievrenseker.com)

Dersin Hocası: M. Elif Karslıgil

(<http://www.ce.yildiz.edu.tr/myindex.php?id=6,elif@ce.yildiz.edu.tr>)

Teşekkür

Bu projenin yapılmasında, derslerinden faydalandığım Yrd. Doç. Elif Karslıgil'e teşekkür edeirm.

Amaç

Bu projenin amacı, verilen bir resim kümesi üzerinde öz vektörler (eigen vectorler) kullanılarak verilen grup resimlerinin kullanılarak öz yüzlerini (eigen face) oluşturmaktır.

Giriş

Projede 50 kişiye ait 10'ar resimden oluşan 500 resimlik bir havuz verilmiş olup bu havuzun 10'arlık resimler halinde öz yüzlerinin çıkartılması istenmiştir. Bunun için hazır olan bir hesaplama paketinden faydalanılmış ve gruplama, yığın işlemleri (batch) ve resim işlemleri bu paketin üzerine yazılmıştır. Bu raporda,

hazırlanmış olan bu ilave yazılımlar ve izlenen yol anlatılacaktır.

1 Ortamın belirlenmesi ve hazırlanması

Yazılım ortamı olarak, geçmiş tecrübelerle dayanarak, JAVA programlama dili seçilmiştir. Yazılımın taşınması gereken ve alt kütüphane olarak kullanılacak önemli bir kısım kaynak kod, "<http://darnok.com/>" adresinde bulunan Konrad Rzeszutek'in sitesinden temin edilmiştir. Ayrıca resim işleme ve resim dosyalarını açıp kaydetme işlemleri için JMF¹ (JAVA Media Framework) paketi yüklenmiştir. Çok büyük miktardaki matrisler ile uğraşıldığı ve standart JDK içerisinde bulunan matrix yapısının sağlamış olduğu methodlar

¹ JMF paketinin detayları için java'nın ana sayfası olan <http://java.sun.com/products/java-media/jmf/index.jsp> adresine bakılabilir.

yetersiz olduğu için , JAMA² (JAVA Matrix Package) paketi de kurulmuştur.

2 Yazılımın tasarımı

Yazılımı bir kara kutu (blackbox) olarak modelleyecek olursak, yazılımın girdileri, bir dizin içerisinde bulunan ve isimlerine göre sıralanmış, 500 adet resimden oluşmaktadır. Kara kutunun çıkışında ise 50 adet öz yüzden oluşan resim dosyası beklenmektedir.

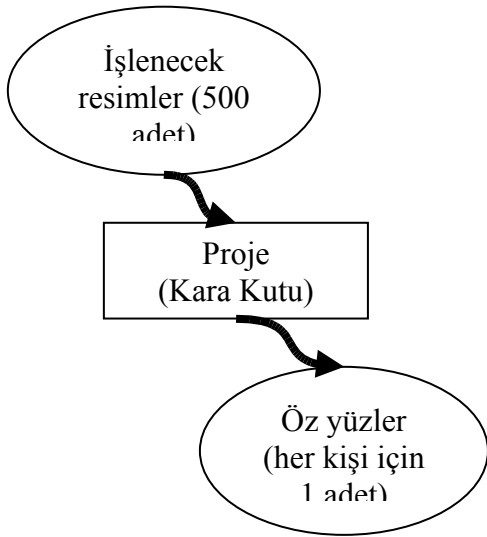


Fig. 1 Sistemin kara kutu modeli

Yukarıdaki figürde, giriş ve çıkış bilgileri gösterilmiştir. Buna göre giriş ve çıkış bilgilerinin özel durumları ve nasıl işlendiklerini inceleyelim.

2.1 Giriş değerleri

Giriş resim dosyalarının özelliği kişilere göre numaralandırılmış olmasıdır. Örneğin 7. kişinin resimlerinin hepsi 07 ismi ile

² JAMA paketi için <http://math.nist.gov/javanumerics/jama/> adresine başvurulabilir.

başlamakta, sonrasında ise 2 sayı ile bu kişinin kaçınıcı resmi olduğunu tutmaktadır. Buna göre 7. kişinin 3. resmi 0703 ismine sahip olmalıdır. Toplamda 500 adet resim olacağı düşünülür ve her kişiye ait 10 adet resim olduğu hesaplanırsa, resimlerin matrix bilgilerini çıkararak ve resim okuyan modül için, 10ar olarak grup (bundle) oluşturmasını söyleyebiliriz. Burada karşılaşılabilecek risk, eğitim setinin içinden bir resmin ileride test bilgisi için kullanılabilecek olması ve resim sayılarının dengesizliğidir. Bu sorunu çözmek için, test resmi ile karşılaşılması durumunda bu resmi atlaması gerekmektedir.

Giriş değerleri ile ilgili bir sorun da bu değerlerin “pgm” yapısında (format) verilmiş olmasıdır. Bu dosya yapısını JAVA ile okumanın yolu, bu proje kapsamında ne yazık ki bulunamamıştır. İnternet’te³ arama sonucu bulunan bir sitede Fransızca olarak hazırlanmış olan bir paket indirilmiş, ancak ne yazık ki çalıştırmaya muvaffak olunamamıştır. Bu yüzden JAVA ile uyumlu olan JPG dosya yapısı proje kapsamında kodlanmış ve giriş değeri olarak verilen dosyaların hepsi pgm’den jpg’ye çevrilmiştir.

³ Projenin detaylı tanımı (Fransızca) ve kodları http://www.enseignement.polytechnique.fr/profs/informatique/Philippe.Chassignet/PGM/pgm_java.htm l adresinde bulunmaktadır.

2.2 Çıkış Değerleri

Projenin çıkış değeri olarak, 50 ayrı kişinin öz yüzleri istenmektedir. Bu resimler, proje kapsamında işlendikten sonra ekranda bir java frame (Java frame'ler, java programlama dilinde yazılan ve işletim sisteminde bir pencere olarak gösterilen java component (bileşen)leridir. Programın çıktısı olarak bu değerler ekranda gösterilmiştir.) içerisinde gösterilmekte ve bir resim dosyasına kaydedilmektedir.

3 Projenin gerçekleştirilmesi (kodlanması)

Bu bölüme kadar tasarımı anlatılan projenin kodlanması ve gerçekleştirilmesi bu bölümde anlatılacaktır. Projenin kodlanması daha öncede bahsedildiği gibi JAVA dilinde yapılmış olup, koda bulunan kritik noktalara, bu bölümün içerisinde alınmıştır. Ayrıca karşılaşılan güçlükler ve çözümleri de bu bölümde anlatılmaktadır.

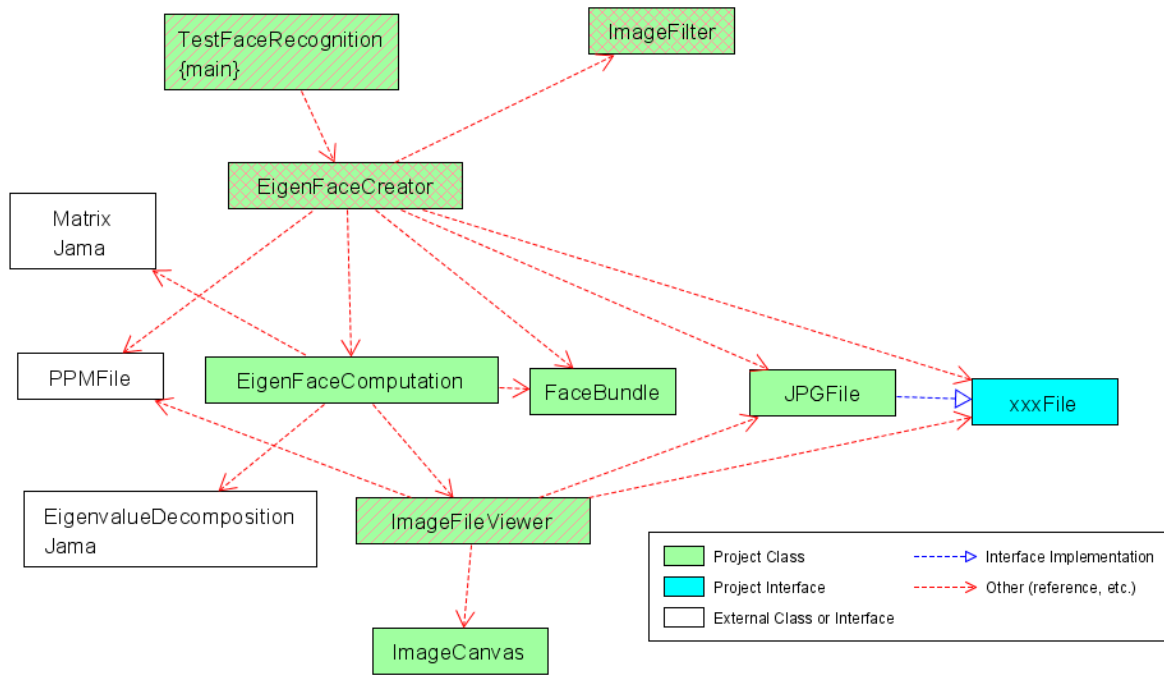


Fig 2. Projenin kapalı UML diyagramı

Yukarıdaki figürde görülen sınıf hiyerarşisinde bulunan nesnelere tanımları aşağıdaki şekildedir.

TestFaceRecognition: Bu sınıf, ana sınıfımız olup, eigenfacecreator class'ını çağırır. Bu sayede, verilen parametreleri işlemiş ve eigenfacecreator, class'ının işleyebileceği şekle getirmiş olur.

EigenFaceCreator: Bu sınıfın görevi,

Proje tasarımına bağlı olarak, JAVA'nın da nesne yönelimli (object oriented) bir dil olmasından faydalanarak, yukarıdaki sınıf hiyerarşisi (class hierarchy) modeli çıkartılmıştır.

3.1 Giriş değerlerini okuyan modlün gerçekleştirilmesi

Giriş değerlerinin atlamasını söyleyen bir koşul yazılmıştır.

```
if(sadifile.equals((String)filenames.get(j+MAGIC_SETNR*i))){
System.out.println("ayni dosyayi buldum train sete koymuyorum");
set[j] = (String)filenames.get(j-1+MAGIC_SETNR*i);
System.out.println(" "+set[j]);
continue;
}
```

yukarıdaki kod parçası bu durumu tespit edip, içinde bulunduğu döngüyü atlatmaktadır.



Fig. 3. Örnek giriş resmi

Giriş değerlerinden örnek bir resim yukarıdaki figürde gösterilmiştir

3.2 Öz Yüz (Eigenface) algoritmasının gerçekleştirilmesi

Kullanılan algoritmanın pseudo kodu aşağıdaki şekildedir⁴:

Algorithm: EigenFaces (probe, gallery, training)	
Input: Face sets probe , gallery , training	
Output: Displays each face in probe along with its best matching face in gallery	
Steps	
Setup (gallery, training)	1
for each $\Gamma \in \text{probe}$	2
match = <i>Recognize</i> (Γ)	3
print Γ , match [1][1] // display face & best match	4

Fig. 4. Eigen Face Algoritması

Buna göre, aranacak olan resim, bir eğitim seti için uygun olan öz yüzlere uzaklığına bakılarak tespit edilmektedir. Burada ortaya çıkan problem, uzaklık ölçümü için mümkün olan yollardan birisinin seçilmesidir.

İsmi geçen makalede her ne kadar Euclidean veya Mahalanobis uzaklıklarının kullanılması önerilse de, projenin bu kısmı için bir şart olmadığından basit bir vector resim farkı

⁴ Algoritma, Preserving Privacy by De-identifying Facial Images,

Elaine Newton Latanya Sweeney Bradley Malin, Carnegie Mellon University, 2003, makalesinden alınmıştır.

alınarak bu modül yazılmıştır. Kodu aşağıdaki şekildedir:

```
satır 1.double[] distance = new
    double[MAGIC_NR];
satır 2.double[] minDistance = new
    double[MAGIC_NR];
satır 3.idx = 0;
satır 4.for (image = 0; image <
    nrfaces; image++) {
satır 5.temp = 0.0;
satır 6.for (j = 0; j < MAGIC_NR;
    j++) {
satır 7.distance[j] =
    Math.abs(input_wk[j] -
    wk[image][j]);
satır 8.//System.out.print(distance[
    j)+"\t");
satır 9.}
satır 10.if (image == 0)
satır 11.System.arraycopy(distance, 0
    ,minDistance, 0,MAGIC_NR);
satır 12.if (sum(minDistance) >
    sum(distance)) {
satır 13.this.idx = image;
satır 14.System.arraycopy(distance, 0
    ,minDistance, 0,MAGIC_NR);
satır 15.}
satır 16.}
```

Yukarıdaki figürde görüldüğü üzere, her bir resimin kendisi için hesaplanan eigen vector değerinden çıkarılması sonucu bulunan değerlerin toplamı kadar uzaklıkta olmaktadır. Bu değerler daha sonra hesaplamada kolaylık sağlaması açısından eğitim dosyalarının bulunduğu dizinde bir .cache dosyası olarak saklanmaktadır. Şayet program bu dosyayı bulursa, tekrar hesaplama yapmamakta, hazır hesaplamaları kullanmaktadır.

3.3 Çıkış değerlerini işleyen modül

Proje'nin hesaplama ve öz yüzleri bulmasından sonra, projenin bu sonuçlarını gösteren bir modül yazılması ihtiyacı doğmuştur. Ayrıca bulunan değerler, bir double dizide (array) bulunduğu için bu değerlerin direk olarak ekran görülmesini sağlayan ilave bir sınıf (class) yazılmış olup bu sınıf kullanılmıştır. Kısaca, JAVA'da ekrana resim basmak için kullanılan canvas sınıfından nesnelere kullanılmıştır. Ancak bu canvas sınıfının içinde bulunan Image nesnesi, dosya kaydetme işlemleri için kullanılan BufferedImage nesnesi ile birbirine dönüşmemektedir. Bunu çözmek için geçici bir graphics2d objesi yapılmış ve bunun içine resim çizdirilerek sonuç, dosyaya aktarılmıştır.

```
satır 1.memImage =  
    createImage(new  
        MemoryImageSource(width,  
            height, pix, 0, width));  
  
satır 2.BufferedImage image =  
    new BufferedImage(w, h,  
        BufferedImage.TYPE_INT_RGB)  
    ;  
  
satır 3.Graphics2D g2 =  
    image.createGraphics();  
  
satır 4.paint(g2);  
  
satır 5.g2.dispose();  
  
satır 6.try{  
  
satır 7.ImageIO.write(image,  
    "png", new  
        File(""+id+".png"));  
  
satır 8.}catch(IOException  
    e){}
```

Yukarıdaki kod parçasında, 1. satırda, canvas içerisine yüklü olan resim parçası, hafıza görüntüsü olarak saklanıyor. 2. satırda BufferedImage tipindeki class'tan yapılmış olan bir nesne içerirse bu hafız resmi büyüklüğünde bir yer açılıyor. Daha sonra 3. ve 4. satırlarda bu resimden oluşturulan graphics2D nesnesinin içerisine hafızada bulunan resim çiziliyor. Daha sonra oluşturulan bu resim dosyaya basılıyor. Dikkat edilecek bir diğer nokta ise, oluşturulan bu resimlerin dosya isimlerinin verilmesidir. Dosya isimleri öz yüzü oluşturulan kişinin numarasıyla atanmaktadır ve bu bilgi, sisteme dosya okuyan modülden buraya kadar taşınmaktadır.



Fig. 5. Örnek çıkış resmi

4 Sonuç

Proje kapsamında, resim işleme işlemleri için bir yazılım geliştirilmiş, öz resimlerin nasıl hesaplanacağı, ve mesafe hesaplama yöntemleri incelenmiş ve yapılan araştırmalar sonucunda elde edilen bilgiler ile bir yazılım geliştirilmiştir. Yazılımın çıktısının renkli bir resim olması, beklenmeyen bir durumdur ancak bunun resimleri pgm formatından jpg formatına çevirmekten kaynaklandığı tahmin edilmektedir. Ayrıca her ne kadar projenin amacı yüz tanıma işlemi olmasa da, one-against-all metodu kullanılarak basit bir tanıma işlemi yazılmış ancak mesafe hesaplama fonksiyonlarının tam olmamasından dolayı başarı elde edilememiştir.

Sonuç olarak bu proje sayesinde resim işleme ve biometrik konusuna bir giriş yapılmış olup burada öğrenilen bilgiler ile konunun daha da detaylı öğrenilmesinin yolu açılmış oluyor.