İSTANBUL TEKNİK ÜNİVERSİTESİ ★ FEN BİLİMLERİ ENSTİTÜSÜ

TEMEL BİLEŞEN ANALİZİ VE FISHER DOĞRUSAL AYIRICILAR YÖNTEMLERİ İLE KULAK BİYOMETRİSİ

YÜKSEK LİSANS TEZİ

Murat KUŞ

Elektronik ve Haberleşme Mühendisliği Anabilim Dalı

Elektronik Mühendisliği Programı

HAZİRAN 2013

İSTANBUL TEKNİK ÜNİVERSİTESİ ★ FEN BİLİMLERİ ENSTİTÜSÜ

TEMEL BİLEŞEN ANALİZİ VE FISHER DOĞRUSAL AYIRICILAR YÖNTEMLERİ İLE KULAK BİYOMETRİSİ

YÜKSEK LİSANS TEZİ

Murat KUŞ
(504101216)

Elektronik ve Haberleşme Mühendisliği Anabilim Dalı

Elektronik Mühendisliği Programı

Tez Danışmanı: Doç. Dr. Mürvet KIRCI

HAZİRAN 2013

İTÜ, Fen Bilimleri Enstitüsü'nün 504101216 numaralı Yüksek Lisans Öğrencisi **Murat KUŞ**, ilgili yönetmeliklerin belirlediği gerekli tüm şartları yerine getirdikten sonra hazırladığı **"TEMEL BİLEŞEN ANALİZİ VE FISHER DOĞRUSAL AYIRICILAR YÖNTEMLERİ İLE KULAK BİYOMETRİSİ "** başlıklı tezini aşağıda imzaları olan jüri önünde başarı ile sunmuştur.

Tez Danışmanı :	**Doç. Dr. Mürvet KIRCI** İstanbul Teknik Üniversitesi	……..
Jüri Üyeleri :	**Prof. Dr. Ece Olcay GÜNEŞ** İstanbul Teknik Üniversitesi	……..
	Prof. Dr. Coşkun SÖNMEZ Yıldız Teknik Üniversitesi	……..

Teslim Tarihi : **2 Mayıs 2013**

Savunma Tarihi : **6 Haziran 2013**

Aileme,

ÖNSÖZ

Öncelikle tez danışmanım Doç. Dr. Mürvet KIRCI'ya tezin hazırlanma sürecinde yapmış olduğu tüm desteği, gerek sosyal hayat gerekse tez konusunda sağlamış olduğu rehberliği, bilimsel olarak eksik olduğum konuları tamamlamama yardımcı olarak bilimsel yeterliliğimi üst düzeye çıkardığı ve tüm olumsuzluklara rağmen sabrından dolayı teşekkür ve şükranlarımı sunmak isterim.

Ayrıca tez süresinde desteğini eksiltmeden devam ettiren Prof. Dr. Ece Olcay GÜNEŞ'e de teşekkürlerimi sunmak isterim.

Öğrenimim süresince aynı dersleri alarak tanışma fırsatı bulduğum Mehmet DUMAN, Musa YAZAR, Mehmet Emin ŞAHİN ve Ersin ALAYBEYOĞLU'na 2 yıl boyunca göstermiş oldukları sabır ve sosyal desteklerden, bunun dışında özellikle meslektaşlarım Dz.Kd.Ütğm.Berkay ÇELİK ve Dz.Ütğm.Bahadır BİCAN'a tez aşamasında çeşitli konularda yardımcı olmak maksadıyla göstermiş oldukları çabalarından dolayı şükranlarımı sunarım.

Son olarak, iki yıl süren öğrenimim süresince hiçbir zaman duygusal ve mantıksal yol göstericiliklerini esirgemeyen ve her zaman her koşulda bana destek olan sevgili babam Himmet KUŞ, biricik annem Fatma KUŞ, canım ablalarım Cihan ve Canan KUŞ, eniştem İsmail KARAGÖZ ve canımdan çok sevdiğim yeğenlerim M.Tuna ve K.Erdem KARAGÖZ'e candan sevgilerimi ve teşekkürlerimi sunarım.

Mayıs 2013

Murat KUŞ
(Deniz Subayı)

İÇİNDEKİLER

Sayfa

ÖNSÖZ ..... **vii**
İÇİNDEKİLER ..... **ix**
KISALTMALAR ..... **xi**
SEMBOL LİSTESİ ..... **xiii**
ÇİZELGE LİSTESİ ..... **xv**
ŞEKİL LİSTESİ ..... **xvii**
ÖZET ..... **xix**
SUMMARY ..... **xxiii**
1. GİRİŞ **1**
1.1 Biyometri Nedir? 1
1.2 Biyometrik Sistem Temelleri 2
2. KULAK BİYOMETRİSİ **5**
2.1 Kulak Anatomisi 5
2.2 Kulak Biyometri Sistemi 6
2.3 Kulak Biyometrisi Üzerine Daha Önce Yapılan Çalışmalar 7
3. KULAK TANIMA ALGORİTMALARI **11**
3.1 Temel Bileşen Analizi (TBA) / Principal Component Analysis (PCA) 11
3.1.1 TBA'nın teorisi 12
3.1.2 TBA'nın uygulanması 14
3.1.3 TBA'nın kulak resimlerine uygulanması 25
3.2 Fisher Doğrusal Ayırıcılar Yöntemi (FDAY) / Fisher's Linear Discriminant Analysis (FLDA) 32
3.2.1 FDAY'nın teorisi 33
3.2.2 FDAY'nin uygulanması 37
3.2.3 Çoklu FDAY 40
3.2.4 Çoklu FDAY'nin uygulanması 44
4. KULAK TANIMA UYGULAMALARINDA ORTAYA ÇIKAN PROBLEMLERİN ÇÖZÜM YÖNTEMLERİ **49**
4.1 Jacobi İterasyonu ile Özdeğer/Özvektör Bulunması 49
4.2 LU ayrıştırması ile matris tersinin hesaplanması 54
4.3 QR ayrıştırma algoritması ile özdeğer/özvektör bulunması 57
5. ALGORİTMİK TESTLER **61**
5.1 Resim Matrisleriyle Elde Edilen Sonuçlar 62
5.1.1 TBA'nın test edilmesi ve elde edilen sonuçlar 62
5.1.2 FDAY'nin test edilmesi ve elde edilen sonuçlar 66
5.2 Histogram Matrisleriyle Elde Edilen Sonuçlar 69
5.2.1 TBA'nın test edilmesi ve elde edilen sonuçlar 69
5.2.2 FDAY'nin test edilmesi ve elde edilen sonuçlar 72
6. GÖMÜLÜ SİSTEM GERÇEKLEME **73**
6.1 Sistem Tasarımında Kullanılan Elemanlar 73
6.2 Sistemin Oluşturulması ve Denenmesi 77

6.2.1 Sistemin kendi içerisinde gerçeklenmesi ve testler 77
6.2.2 Kameralı gerçekleme ve testler 81
7. SONUÇ 83
KAYNAKLAR.....Hata! Yer işareti tanımlanmamış.
ÖZGEÇMİŞ.....Hata! Yer işareti tanımlanmamış.**7**

KISALTMALAR

AOVY : Ayırıcı Ortak Vektör Yaklaşımı
bps : bit per second
col : column (sütun)
cos : cosinus
cot : cotanjant
E : Eğitim kümesi
eig_i : eigenvector
FDAY : Fisher Doğrusal Ayırıcılar Yöntemi
H : Height (yükseklik)
I^2C : Inter IC Bus
FLDA : Fisher's Linear Discriminant Analysis
K-L : Karhunen - Loeve
KByte : Kilo Byte
MHz : Mega Hertz
PCA : Principal Component Analysis
RGB : Red, Green, Blue
ROM : Read Only Memory
row : row (satır)
SDRAM : **S**ynchronous Dynamic Random Access Memory
sign : signum
sin : sinus
SPI : Serial Peripheral Interface
tan : tanjant
TBA : Temel Bileşen Anlizi
TTL : Time To Live
UART : Universal Asynchronous Receiver/Transmitter
W : Width (genişlik)

SEMBOL LİSTESİ

$\mathbf{J_0}$	: Karasel Hata Kriteri
$\boldsymbol{\mu}$	: Ortalama
$\mathbf{x_k}$	: k. Örneğe ait Vektör
e	: Özvektör, TBA
$\mathbf{a_k}$	: Katsayılar Kümesi k. Elemanı
S	: Saçılma Matrisi
$\boldsymbol{\lambda}$	: Özdeğer
d	: Uzay Boyutu
$\mathbf{P_i}$	: Orijinal Veri Noktaları
$\mathbf{P'_i}$	: Ortalama Çıkarılmış Veri Noktaları
C	: Kovaryans Matrisi
S_V_x	: Sonuç verisi_x
S_V_y	: Sonuç verisi_y
S_V	: Sonuç verisi
$\mathbf{P_i_orj}$	: Tekrar elde edilen orijinal veri noktası
A	: Ortalama Çıkarılmış Matris
I	: Birim Matris
K(x,y)	: İki Boyutlu Kulak Resmi
N	: Toplam Resim Piksel Sayısı
Γi	: Kulak Resim Vektörü
Ψ	: Resim Ortalaması
$\mathbf{\Phi_i}$	: Kulak Resim Vektörlerinin Ortalamadan Çıkarılmış Hali
M	: Toplam Resim Sayısı
L	: Yardımcı Matris
v	: Özvektör
u	: Özvektör
$\boldsymbol{\mu_i}$	: Sınıf ortalaması
M'	: Ağırlıklara Göre Kullanılan Özdeğer/Özvektör Sayısı
$\boldsymbol{\vartheta_k}$	: Kulak Uzayına Yansıtılmış Vektör
$\boldsymbol{\varepsilon_u}$	: Kulak Uzayına Uzaklık
$\boldsymbol{\varepsilon_k}$	: Sınıf Uzayına Uzaklık
W	: Yansıma Matrisi.
$\tilde{\boldsymbol{\mu}}_\mathbf{i}$	: Yansıtılmış Sınıf Ortalaması
$\tilde{\boldsymbol{S}}_\mathbf{i}$	: Yansıtılmış Sınıf Saçılma Matrisi
J(w)	: Fisher Kriter Fonksiyonu
$\mathbf{D_i}$	: Sınıf Belirteci
$\mathbf{S_i}$	: Sınıfa Ait Saçılma Matrisi
$\mathbf{S_w}$	: Sınıfiçi Saçılma Matrisi
$\mathbf{S_B}$	: Sınıflar Arası Saçılma Matrisi
k	: FDAY Sınıf Sayısı
$\boldsymbol{\mu_T}$	: Tüm Örneklerin Ortalaması
$\mathbf{S_T}$	: Toplam Saçılma Matrisi

$\mathbf{y_i}$: yansıma sonucu oluşan vektör
$\tilde{\boldsymbol{S}}_{\mathbf{w}}$: Yansıtılmış Sınıfiçi Saçılma Matrisi
$\tilde{\boldsymbol{S}}_{\mathbf{B}}$: Yansıtılmış Sınıflar Arası Saçılma Matrisi
R : Düzlem Dönüşüm Matrisi
D : Dönüşüm Yardımcı Matrisi (Jacobi İterasyonu – Özdeğer Matrisi)
c : Kosinüs
s : Sinüs
t : Tanjant
V : Dönüşüm Matrisleri Çarpımı (Jacobi İterasyonu – Özvektör Matrisi)
$\mathbf{T_i}$: Köşegen Elemanların Karasel Toplamı
$\boldsymbol{\varepsilon}$: Jacobi İterasyonu için Eşik Değeri
L : Alt Üçgensel Matris (LU Ayrıştırması)
U : Üst Üçgensel Matris (LU Ayrıştırması)
R : Üst Üçgensel Matris (QR Ayrıştırması)
Q : Ortogonal Matris (QR Ayrıştırması)
T : Üçgensel Matris (QR Ayrıştırması)
$\mathbf{r_{ik}}$: R matrisine ait Elemanlar
$\mathbf{A_k}$: k. Adımda Matris (QR Ayrıştırması)

ÇİZELGE LİSTESİ

Sayfa

Çizelge 3.1 : Örnek veri noktaları. 15
Çizelge 3.2 : Ortalama çıkarılmış veri noktaları. 15
Çizelge 3.3 : Her iki temel bileşen kullanılarak elde edilen sonuç verisi. 19
Çizelge 3.4 : En büyük temel bileşen kullanılarak elde edilen sonuç verisi. 20
Çizelge 3.5 : Her iki temel bileşen kullanılarak elde orijinal veri kümesi. 23
Çizelge 3.6 : Tek bir temel bileşen kullanılarak elde orijinal veri kümesi. 23
Çizelge 3.7 : FDAY için kullanılan veri noktaları kümesi. 38
Çizelge 3.8 : Çoklu FDAY için kullanılan veri noktaları kümesi. 44
Çizelge 5.1 : TBA'da öklid uzaklığı eşit seçilerek ve tüm özdeğer/özvektör çifti kullanılarak yapılan testlerde elde edilen sonuçlar. 63
Çizelge 5.2 : TBA'da öklid uzaklıkları farklı seçilerek ve tüm özdeğer/özvektör çifti kullanılarak yapılan testlerde elde edilen sonuçlar. 63
Çizelge 5.3 : TBA'da öklid uzaklığı eşit ancak özdeğer/özvektör çiftinin %99'luk kısmının kullanılmasıyla elde edilen sonuçlar. 64
Çizelge 5.4 : TBA'da öklid uzaklığı farklı ve özdeğer/özvektör çiftinin %99'luk kısmının kullanılmasıyla elde edilen sonuçlar. 64
Çizelge 5.5 : TBA'da maksimum tanıma oranı için kullanılan değerler ve elde edilen sonuçlar. 65
Çizelge 5.6 : TBA aşamasında tüm özdeğer/özvektör çifti kullanılarak yapılan FDAY'de elde edilen sonuçlar. 66
Çizelge 5.7 : TBA aşamasında özdeğer/özvektör çiftinin %99'luk kısmının aynı öklid uzaklığı için kullanılmasıyla FDAY'de elde edilen sonuçlar. 67
Çizelge 5.8 : TBA aşamasında tüm özdeğer/özvektör çifti ile ökild uzaklığındaki revize sonunda yapılan FDAY'de elde edilen sonuçlar. 67
Çizelge 5.9 : TBA aşamasında özdeğer/özvektör çiftinin %99'luk kısmı ile öklid uzaklığındaki revize sonunda FDAY'de elde edilen sonuçlar. 68
Çizelge 5.10 : TBA'da histogram denemesi için öklid uzaklığı eşit, tüm özdeğer /özvektör çifti kullanıldığında elde edilen sonuçlar. 69
Çizelge 5.11 : TBA'da histogram denemesi için öklid uzaklıkları farklı, tüm özdeğer/özvektör çifti kullanıldığında elde edilen sonuçlar. 70
Çizelge 5.12 : TBA'da histogram denemesi için öklid uzaklığı eşit, özdeğer /özvektör çiftinin %99'luk kısmı kullanıldığında sonuçlar. 70
Çizelge 5.13 : TBA'da histogram denemesi için öklid uzaklığı farklı, özdeğer/ özvektör çiftinin %99'luk kısmı kullanıldığında sonuçlar. 71
Çizelge 5.14 : TBA histogram uygulamasında maksimum tanıma oranı için elde edilen sonuçlar. 71
Çizelge 5.15 : FDAY'nin histogram bilgilerine uygulanması ve sonuçlar. 72
Çizelge 6.1 : Sistemin TBA ile denenmesi: öklit uzaklıkları eşit ve tüm özdeğer /özvektörler kullanılarak. 78

Çizelge 6.2 : Sistemin TBA ile denenmesi: öklit uzaklıkları farklı ve tüm özdeğer /özvektörler kullanılarak. 78
Çizelge 6.3 : Sistemin TBA ile denemesi: öklit uzaklıkları eşit ve %99 sayıda özdeğer/özvektör kullanılarak. 78
Çizelge 6.4 : Sistemin TBA ile denemesi: öklit uzaklıkları farklı ve %99 sayıda özdeğer/özvektör kullanılarak. 79
Çizelge 6.5 : Sistemin TBA ile denemesi: maksimum tanıma oranı için öklit uzaklıkları ve %99 sayıda özdeğer/özvektör kullanılarak. 79
Çizelge 6.6 : Sistemin FDAY ile denenmesi: tüm özdeğer/özvektörler kullanılarak. 79
Çizelge 6.7 : Sistemin FDAY ile denenmesi: aynı öklid uzaklıkları için %99 sayıda özdeğer/özvektör kullanılarak. 79
Çizelge 6.8 : Sistemin FDAY ile denenmesi: öklid uzaklıklarında revize ve tüm özdeğer/özvektörler kullanılarak. 80
Çizelge 6.9 : Sistemin FDAY ile denenmesi: öklid uzaklıklarında revize ve %99 sayıda özdeğer/özvektör kullanılarak. 80

ŞEKİL LİSTESİ

Sayfa

Şekil 2.1 : Dokuzuncu haftada kulağın dış anatomisi. 5
Şekil 2.2 : Yetişkin bir insanın kulak anatomisi ve kısımları. 5
Şekil 2.3 : Kulak tanıma sisteminin aşamaları. 6
Şekil 2.4 : Iannarelli sistemi ölçüm yöntemi. 8
Şekil 3.1 : Örnek verinin plotlanmış hali. 15
Şekil 3.2 : Ortalama çıkartılmış veri noktalarının gösterimi. 16
Şekil 3.3 : Özvektörlerle birlikte ortalama çıkarılmış veri noktaları. 17
Şekil 3.4 : İki temel bileşen kullanılarak elde edilen sonuç verisi. 20
Şekil 3.5 : Tek temel bileşen kullanılarak elde edilen sonuç verisi. 21
Şekil 3.6 : İki temel bileşen kullanılarak elde orijinal veri. 24
Şekil 3.7 : Tek temel bileşen kullanılarak elde edilen orijinal veri. 24
Şekil 3.8 : 80x60'lık orijinal kulak resmi ve 10 adet temel bileşen kullanılarak elde edilen kulak resmi. 26
Şekil 3.9 : Resim verisinin satır vektörü formuna dönüşümü. 27
Şekil 3.10 : Kulak resminin kulak satır vektörüne dönüşümü. 27
Şekil 3.11 : W x H boyutlu bir kulak resminin (WH) x 1 boyutlu sütun vektörüne dönüşümü. 27
Şekil 3.12 : Kulak uzayı. 28
Şekil 3.13 : Örnek özkulak resimleri. 28
Şekil 3.14 : Kulak uzayı ve komşuluk. 28
Şekil 3.15 : TBA uygulandıktan sonra verinin durumu. 33
Şekil 3.16 : FDAY ile bir doğru üzerine yapılan yansıtmaya ait iki sonuç. 33
Şekil 3.17 : x_i noktasının w doğrultusundaki bir doğruya yansıtılması. 34
Şekil 3.18 : FDAY'nın kriterlerine göre doğru yansıma. 35
Şekil 3.19 : FDAY için plotlanan orijinal veri noktaları. 38
Şekil 3.20 : Veri noktaları ve en büyük özdeğerle ilişkili en uygun yansıma ekseni. 39
Şekil 3.21 : 3 farklı sınıfın 2 adet düzleme yansıtılması. 42
Şekil 3.22 : Çoklu FDAY için 3 boyutlu plotlanan orijinal veri noktaları. 45
Şekil 3.23 : Veri noktaları ve en büyük özdeğerle ilişkili yansıma ekseni. 46
Şekil 3.24 : Veri noktaları ve ikinci en büyük özdeğerle ilişkili yansıma ekseni. 47
Şekil 4.1 : Gauss eliminasyonu ve LU ayrıştırmasının karşılaştırılması. 55
Şekil 4.2 : Revize Gram Schmidt: Schwarz-Ratishauser Değişikliği yöntemi ile hesaplamanın matris üzerindeki etkisi. 59
Şekil 5.1 : Örnek bir kulak resmi ve histogramı. 62
Şekil 6.1 : STM42 M4 mikrodenetleyici kartı. 74
Şekil 6.2 : Batarya kartı. 75
Şekil 6.3 : STM32 M4 ve batarya kartı. 75
Şekil 6.4 : µCAM Serial JPEG camera module. 75
Şekil 6.5 : µUSB CE5. 76

Şekil 6.6 : STM32 M4 ve µCAM TTL. 77

TEMEL BİLEŞEN ANALİZİ VE FISHER DOĞRUSAL AYIRICILAR YÖNTEMLERİ İLE KULAK BİYOMETRİSİ

ÖZET

Bu teze konu olan çalışmada, kulak biyometrisi kullanarak herhangi bir bilgisayara ya da bir platforma bağlı olmadan çalışabilen bir tanıma sistemi tasarlanmıştır. Tasarımda temel alınan tanıma algoritma ve yöntemleri, Temel Bileşen Analizi ve Fisher Doğrusal Ayırıcılar Yöntemi'dir.

Bilindiği üzere, teknolojinin gelişmesiyle insanların güvenlik ihtiyaçları artmış, bununla paralel olarak da güvenliği sağlayacak sistemlerin geliştirilmelerine hız verilmiştir. Bu ihtiyaçtan yola çıkarak, kulak biyometrisi kullanılarak bu zaafiyetin giderilmesi amaçlanmıştır.

Kulak, yapılan çalışmalar neticesinde, her insanda biyometrik bir özelliğe sahip olmakta ve ikizler de dahil olmak üzere her birey için farklılık göstermektedir. Biyometrik bir ayraç olarak kullanımı, diğer ayraçlara göre basitlik, işlenecek verinin çok olmayışı gibi nedenlerden dolayı avantajlıdır.

Sistemin tasarlanmasında kullanılan algoritmalardan ilki Temel Bileşen Analizi'dir. TBA, çok değişkenli veri tabanlarının analizi ve büyük boyutlu verilerin daha düşük boyutlara düşürülebilmesinde kullanılan en eski istatistiksel araç olarak tanımlanabilir. Büyük boyutların daha aza indergenmesine paralel olarak TBA, çok sayıda birbiri ile ilişkili değişkenler içeren veri kümesinde, veri içerisinde varolan değişimlerin mümkün olduğunca korunarak dönüşümünü sağlar. Yöntem, eldeki veriyi daha az sayıda değişkenle ifade edebilecek en iyi dönüşümü belirler ve dönüşüm sonrasında elde edilen değişkenler ilk değişkenlerin temel bileşenleri olarak adlandırılır. Buna göre ilk temel bileşen, aralarında varyansı en büyük olan; ikincisi varyansı sonraki büyük olandır. Tüm temel bileşenler için sıralama varyans değerleri temel alınarak bu şekilde azalarak devam eder. Ortaya çıkan tüm temel bileşenler birbirlerinden bağımsızdır. Bu, veri kümesinin birbirine bağımlılığını da ortadan kaldırır. Elde edilen temel bileşenler yardımıyla başlangıçtaki veri kümesi farklı eksenler üzerinde ifade edilebilir hale gelir. Yeni eksenlerden seçilen en önemlileri ile de en az kayıpla tüm veri temsil edilebilir, gerekirse tekrar oluşturulabilir. Bu çalışmada TBA, uygulama kolaylığı ve anlaşılabilirliği nedeniyle başlangıç seviyesi olarak tercih edilmiştir.

Tasarımda kullanılan diğer algoritma ise, Fisher Doğrusal Ayırıcılar Yöntemi'dir. FDAY, genel olarak varyansların analizine dayanmaktadır. Ancak bununla birlikte başlangıç veri kümesinde bulunan farklı sınıfların birbirinden en iyi şekilde ayrılmasını da amaçlar. Başka bir deyişle FDAY genel olarak sınıflar arası ayrılığı mümkün olduğunca artırarak sınıf içi değişkenliğin en aza indirilmesi prensibine dayanır. Böylece bu sınıfları en iyi ayıran vektörler bulunacaktır. FDAY, uygulamada iki aşamadan oluşur. Öncelikle orijinal veri kümesine TBA uygulanarak tüm verinin bağımsız bileşenlerle ifade edilmesi ve boyut küçültme sağlanır,

ardından FDAY uygulanarak en iyi ayrımı sağlayacak vektörler bulunur. Ardından, verinin bu vektörler doğrultusunda yansıtılması neticesinde sınıfların ayrımı gerçekleştirilir.

Her iki yöntemin teorileri anlaşıldıktan sonra uygulama aşamasına geçilmiştir. Yöntemlere ait kodlar öncelikle bilgisayar ortamında MATLAB programı yardımıyla denenmiş, ve kulak biyometrisi ile sınıflandırmanın yüksek doğrulukla yapılabildiği görülmüştür. Müteakiben, bu çalışmanın asıl amacı olan gömülü sistem tasarımı kısmına geçilmiştir. Geçiş esnasında, yöntemlerin içeriğinde bulunan, özdeğer/özvektör çiftlerinin bulunması ve matris tersi gibi işlemlerin hesaplanması sorun teşkil etmiştir.

Mevcut sorunlar yapılan araştırmalar sonunda elde edilen nümerik yöntemlerle giderilmiştir. Bu nümerik yöntemler şu şekilde sıralanabilir: Jacobi İterasyonu, LU Ayrıştırması ve QR Ayrıştırması. Jacobi iterasyon yöntemi, simetrik ve gerçek matrislerin özdeğer/özvektör çiftlerinin hesaplanması maksadıyla kullanılmıştır. Yöntem temel olarak matris üzerinde düzlem döndürme prensibine dayanır. Jacobi iterasyonu, TBA yönteminde simetrik olan kovaryans matrisinin özdeğer/özvektörlerinin hesaplanmasında kullanılmıştır. LU ayrıştırması, FDAY'da karşımıza çıkan sınıf içi saçılma matrisinin tersi hesaplanırken kullanılmıştır. Temel olarak Gauss Eliminasyonuna benzemekle birlikte, zorluk derecesi olarak Gauss'dan daha kolaydır. QR ayrıştırması ise, simetrik olmayan matrislerde özdeğer/özvektör çiftlerini hesaplarken kullanılan bir yöntemdir. Benzer matrislerin aynı özdeğer/özvektör çiftlerine sahip olması prensibiyle işlemler gerçekleştirilir.

Sistem gerçeklenmesinde kullanılan malzemelerden ilki, MikroElektronika firmasına ait, ARM teknolojisi tabanlı, STM32 M4 mikro kontrolör kartıdır ve cortex M4 teknolojisiyle donatılmıştır. Kullanılan derleyici, yine MikroElektronika firmasınca geliştirilen, C tabanlı MicroC derleyicisidir. Derleyici, bünyesinde bazı matematiksel ve ihtiyaç duyulan diğer fonksiyonları barındırması nedeniyle avantajlıdır.

Kullanılan kamera elemanı ise, 4D Systems firması tarafından üretilen µCAM Serial JPEG Camera Module'dür. Kamera, bir çok çözünürlük ve formatı desteklemesinin yanısıra, mikro denetleyici kartının işleyebileceği özellikte resimler de sağlayabilmektedir. Ayrıca modüler ve kolay monte edilebilmesi nedeniyle seçilmiştir.

Bilgisayar ortamında yapılan denemeler neticesinde elde edilen sonuçlar, sistemin gerçeklenmesine ışık tutmuş, kulak biyometrisinin de tanıma sistemlerinde kullanılabileceğini göstermiştir. Gerçekleme sonunda elde edilen sonuçlar da, bilgisayardan bağımsız bir tanıma sistemi için yeterli doğruluklar vermiş, bilgisayar ortamında elde edilen tanıma oranlarına paralel sonuçlar elde edilmiştir.

Bu çalışma temel olarak 7 bölümden oluşmaktadır. Birinci bölümde, biyometrinin tanımından ve özelliklerinden bunlarla birlikte bir biyometrik sistemin temellerinden bahsedilmiştir.

İkinci bölüm ise kulak biyometrisini detaylarıyla irdelemekte, öncelikle bireyin yaşamında kulağın oluşumu ve gelişimi ile anatomisi ele alınmaktadır. Müteakiben, kulak biyometri sisteminin aşamaları ve nasıl icra edildiği anlatılmakta, son olarak da daha önce kulak biyometrisi üzerine yapılan çalışmalara değinilmektedir.

Müteakiben oluşturulan üçüncü bölümde, bu çalışmada kullanılan algoritmalar detaylı olarak irdelenmiş ve örneklerle uygulamalı olarak gösterilmiştir. Öncelikle

TBA teorik olarak anlatılmış daha sonra örnek uygulama yapılmıştır. Ardından aynı işlem FDAY için yapılarak üçüncü bölüm bitirilmiştir.

Dördüncü bölüm, algoritmaların gömülü bir sisteme uygulanmasında ortaya çıkan problemleri ve bunlara uygulanan çözüm yöntemlerini içerir. Detaylı olarak kullanılan nümerik yöntemler anlatılmış ve algoritmaları verilmiştir.

Beşinci bölüm, işlenen algoritma ve yöntemlerin bilgisayar ortamında denenmesini kapsar. Denemelere öncelikle TBA ile başlanmış ve TBA'ya ilişkin farklı versiyonların denemeleri ve elde edilen sonuçlar gösterilmiştir. Ardından aynı işlemler FDAY için de izlenmiş ve elde edilen sonuçlar verilmiştir. Bu bölümde alışılmışın dışında her iki yöntem için histogramlarla da denemeler yapılmış ve elde edilen sonuçlar paylaşılmıştır.

Altıncı bölümde her iki yöntemin ve daha önce işlenen nümerik yöntemlerin gömülü sistemde gerçeklenmesi anlatılmıştır. Öncelikle kullanılan malzemeler tanıtılmış, ardından yöntemlerle icra edilen testler ve sonuçları verilmiştir.

Son olarak yedinci bölümde, elde edilen sonuçlar yorumlanarak tez bitirilmiştir.

EAR BIOMETRIC WITH PRINCIPAL COMPONENT ANALYSIS AND FISHER'S LINEAR DISCRIMINANT ANALYSIS

SUMMARY

In this thesis, a fully embedded recognition system is designed independent from a regular PC or a platform and also without an operation system such as Lınux, Windows, etc. There are lots of algorithms which can be used for a recognition system but in this work, Principal Component Analysis (PCA) and Fisher's Linear Discriminant Analysis (FLDA) are used.

It's obvious that, with the progresses in technology, the security requirements of people are getting more important. Parallel to this, scientist boosted their passion to invent and to realize such systems which can be a solution of all expectations. In order to give a positive effect to these works, ear biometric is used and a system is realized.

Ear, after all the studies in years, can be considered as a good biometric tool and also is different in all human beigns, even in twins. Ear has many advantages among other biometric tools such as face, iris, finger print, etc. Some of these advantages are, it is simpler than others, doesn't have much data to be processed, etc. These advantages can be counted as the reasons why this work focuses on ear.

The first algorithm in the system design is Principal Component Analysis. PCA, is one of the oldest and useful statistical tool to analyze the multivariate data. It can also be used to reduce the dimension of very large data sets. Parallel to this function, PCA aims not to change the variations of the original data set while converting these data set to the new and easier variable space. This algorithm determines the best projection (conversion) which can be described with less variables than the original data set. These new variables can be known as the principal components of the original variables. Regarding to this, the first principal component can be described as the one which has the highest variance and the second one is chosen with respect to the second highest variance. This continues untill all the principal components are ordered with respect to the declining variance values. In this new variable set, all the elements are independent from each other. This cancels the dependence of the original data set with the variables to each other. Also, this new variable set can be used to express the original data in different axes. Choosing an amount of the highest components can lead the dimensionality reduction without loosing much details. Besides that, later one can get the original data back. So in this work, PCA is chosen as the first step of the application, because it is easy to understand and also easy to apply.

The socond algorithm in this work is Fisher's Linear Discriminant Analysis. It can be said that in general, the basics of FLDA is about analysis of the variances. But it also aims the best seperation of the different classes in the original data set. In other words, FLDA aims to maximize the between scatter while minimizing the within

scatter. This finds the best projection vectors which lead the best seperation of the classes at the end. The application of FLDA can be divided into two steps. In the first step, PCA is applied to the original data set in order to make dimensionality reduction and to have a new variable set which is independent from each other. Following to this, FLDA is applied to the new data set to find the best projection vectors. After projecting the data set along these vectors, the best seperation is achieved.

After understanding both theories, application process started. Firstly, all the algorithms and programme codes are tested in a computer with MATLAB programming language. The results show that recognition rates with ear biometrics can be achieved good enough for a recognition system. Later that, the main aim of this thesis started, implementation of the embedded system. It's not easy to apply these algorithms on an embedded system as they have been on a computer. There are some problems about implementation process. For example, finding the eigenpairs or the inverse of a matrix.

After the litrature research, some numerical algorithms have been found in order to solve these problems. These numerical algorithms are Jacobi Iteration Algorithm, LU Decomposition and QR Decomposition. Jacobi iteration is used to solve the eigenpair problem of a real symmetric matrices. The algorithm is based on plane rotations and has been used to solve the problem of the eigenpairs of covariance matrix in PCA. LU decomposition is used to determine the inverse of within scatter matrix in FLDA. Generally, LU decomposition is similar to Gauss Elimination but it's faster and simpler than that. Also, computational cost is less than Gauss. QR decomposition is used to find the eigenpair of unsymmetric real matrices. This algorithm is used in FLDA, too. It is based on the similar matrices rule such as, similar matrices have same eigenpairs.

Realization of the system is done with 2 main elements. First of them is a microcontroller circuit card which is produced by MikroElektronika company. The microcontroller is called STM 32 M4. It is a development card with lots of peripherals which can allow the development of devices with multimedia contents. The main part of the microcontroller card is 32-bit cortex M4 STM32F407VGT6. The programming language is produced by the same company and is called as MicroC. This programming language is based on C language but has some functions in its library which make the programming simpler.

The second element of this system is called as μCAM Serial JPEG Camera Module and is produced by 4D Systems company. It can provide many kinds of pictures with different resolutions, but the one which is used in this work is 80x60 raw images. This camera module has advantages as it's simple to install to the microcontroller and has a compact form.

The results on computer leads the implementation of this system. It shows that ear biometric can be used in human recognition. And the results of the implemented system provides the recognition rates well enough for a recognition system and the rates are similar with the computer experiments.

This thesis consists of 7 main chapters. In chapter 1 an introduction to the topic is made. Later that, a discussion of the meaning of biometrics, its specifications are made. And the last, the main principles of a biometric system is described, as well.

Chapter 2 covers the ear biometric in details. It shows the steps of the ear's physical revolution and changes in its anatomy in a life time. Following that, the main steps of an ear recognition system and how to realize those steps are discussed. Chapter 2 is finished with the study of previous works on ear recognition.

Following that, Chapter 3 tells in detail about the theory of the algorithms which have been used in this work. Not only the theories have been described, but also those theories are applied with examples on MATLAB programme and every steps in these examples are described. First of all, the theory of PCA and an example related to PCA is described, later FLDA has been held in same order.

Chapter 4 includes the informations about the numerical problems of the application on an embedded system. First of all, the problems are listed and after that the numerical solutions for these problems are discussed. Those numerical solutions (algorithms) are discussed in the same order as the recognition algorithms which is in process.

Chapter 5 covers the experimental results of the recognition algorithms on a PC. These experiments have been started with PCA. Different versions of these experiments and the results of these are depicted in here. After that the same processes are held for FLDA algorithms, too. Alternatively, these two algorithms have been also tested for the histograms. The histogram informations have been implemented as an input to PCA and FLDA, respectively. The results for these experiments are represented in this chapter, too.

Chapter 6 tells the implementation of two recognition algorithms in an embedded system. In the first subsection, the elements for the application are introduced with the pictures. Their specifications and details are discussed. After this, in the second subsection, realization and the results are depicted.

At last, a conclusion of this study is made. All the researches and the experimental results are commented.

1. GİRİŞ

Gelişen teknolojiye paralel olarak son yıllarda, kişi tanıma sistemleri birçok bilimadamının ilgi alanına girmiş ve buna yönelik olarak sistem geliştirmeye başlamışlardır. Bu sistemlerde ilgilenilen ana hedef, kişinin biyometrik sınıflandırıcı elemanlarına yönelik olarak yapılacak tanımadır. Bu sınıflandırıcıların bazıları şu şekilde sıralanabilir; iris, yüz, parmak izi, ses vs. Sayılan bu sınıflandırıcılardan özellikle ses ve yüz, insan doğası gereği kendi içinde de kullanabildiği sınıflandırıcılardan sayılabilir. Bu biyometrik sınıflandırıcılardan farklı olarak bilimadamları ayrıca, kulağın da her kişide farklı olduğu ve ortalama 8 ile 70 yaş arasında değişmediğini de anlamışlardır. Bu özellik, bilimadamlarının kulak biyometrisine yönelmesinde ana neden olarak gösterilebilir. Bundan farklı olarak, başka birçok neden daha vardır. Örneğin kulak, yüz kadar ifadelerden etkilenmez. Bu nedenle de işlenip sisteme sokulması kolaydır. Ayrıca kulak, özellik itibariyle yüz kadar aydınlıktan etkilenmez, işlenecek bilgi de yüz ve örneğin irise göre çok küçüktür. Tüm bu nedenler, kulağın biyometrik tanımada kullanılmasını ve araştırmaların buraya yönelmesini sağlamıştır.

1.1 Biyometri Nedir?

Biyometri, "insan kimliğinin fiziksel özellikler (parmak izi, iris, vb.) veya davranışlar (yürüme şekli, vb.) kullanılarak tanınması" olarak tanımlanabilir. Bir başka tanıma göre biyometri ise, "bir insanın ayırıcı özelliklerini kullanarak otomatik olarak tanınması" olarak anlatılmıştır. Daha geniş manada ise, "otomatik olarak gözlemlenebilen, ayırıcı bir fiziksel karakteristik veya kişisel bir özelliğin kişiyi tanımak maksadıyla kullanılması ya da bir şahsın farzedilen kimliğinin doğrulanması" olarak anlatılmaktadır. İnsanlar binlerce yıldır bu tanımlar ışığında, ses, yüz, kişiye has karakteristikler sayesinde tanıma işlemlerini gerçekleştirmektedirler. Teknolojik gelişmeler ve modern hayat, biyometrinin hayatımızdaki yerini arttırmakta ve sağlamlaştırmaktadır.

Ölçülebilirlik, karakteristik özelliklerin bir sensör tarafından kolayca gösterilmesi, yerleştirilmesi ve işlenebilecek düzeyde dijital bir formata dönüştürülmesi demektir. Ölçülebilirlik, biyometrik sınıflandırıcının sistem içerisinde kısa bir sürede işlenerek eşleştirilme yapılması ve buna bağlı bir sınıflandırmanın ortaya çıkması açısından çok önemlidir. Ayrıca bu sayede otomatik bir sistem oluşturma imkânı da artmaktadır.

Sağlamlık, ilgilenilen biyometrik karakteristiğin zaman içerisinde ne derecede önemli bir değişikliğe uğradığı ve bunun sonucu olarak da ne kadar etkili olduğunun ölçüsünü yansıtmaktadır. Bu değişikliklerin zamanla birçok sebebi ortaya çıkacaktır. Örneğin, yaş, kişinin maruz kaldığı kazalar, hastalıklar, vb. bu gibi sebeplerden dolayı çok fazla değişikliğe uğramayan sınıflandırıcılar *"sağlam biyometrik özellikler"* olarak isimlendirilebilirler. İnsan vücudunda en sağlam biyometrik özellik olarak iris gösterilebilir. İris bir insanın hayatı boyunca çok az değiştiğinden örneğin yüz veya sese göre daha sağlam bir biyometrik özelliktir.

Ayırıcılık, insanlar arasında biyometrik özellikler içerisindeki değişimler veya farklılıkların ölçümü olarak tanımlanabilir. Bu değişim ve/veya farklılıklar ne kadar artarsa kişinin tanıma derecesi de buna paralel olarak artacaktır. Değişim oranının azalması nüfus içerisinde farklılığın azalmasına neden olacak, bu da amacımız olan tanıma ve/veya ayırmayı zorlaştıracaktır. Bu anlatılan özellikler ışığında şöyle diyebiliriz, iris ve retina, parmak izi ve ele göre ayırıcılık özelliği bakımından üstündür.

1.2 Biyometrik Sistem Temelleri

Biyometrik özellikler, kişilerin tanınmasında kullanılır. Buna bağlı olarak basit bir biyometrik sistem 3 görevi yerine getirir: elde etme (yakalama), normalize etme ve son olarak tanıma. *Elde etme (yakalama)*, dinamik bir görüntü üzerinde, kulağa benzer resimlerin bulunarak yakalanmasını kapsar. *Normalize etme*, bölme, hizalama ve normalize etme aşamalarını kapsar. Son olarak da *tanıma,* kulak resimlerinin gösterilmesi ve kimlik olarak modellenmesi ve bilinen kulak resimleriyle ilişkilerinin tespit edilerek eşleştirilmesini içerir.

Tanıma aşaması kimliklendirme ve doğrulama olarak da iki adımda gerçekleştirilir. Kimliklendirme, bir biyometrik sistemin sorduğu "X kimdir?" sorusunun yanıtını

bulmayı amaçlar. Bir kimliklendirme işleminde, biyometrik sistem örneği okur, ve okuduğu örneğe istinaden elinde bulunan kalıp ve/veya örnekler arasında eşleştirme yapar. Sistem tasarım yapısına göre, sonuç ya en iyi karşılaştırmayı verir ve "X şudur" der ya da olası tüm sonuçlar sıralanarak "X şunlar olabilir" şeklinde sergilenir. Kimliklendirme uygulamaları, suçluları, teröristleri bulmak için yaygın olarak kullanılmaktadır.

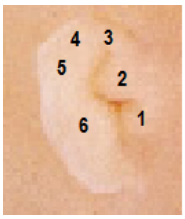
Diğer taraftan doğrulama, biyometrik sistemin kişinin X olduğu varsayımı etrafında, "Bu X mi?" sorusuna yanıt araması olarak tanımlanabilir. Herhangi bir kimlik doğrulama uygulamasında, biyometrik sistem öncelikle kullanıcıdan bir girdi bekler. Bu girdi bir şifre veya bir anahtar ya da herhangi bir kullanıcı ismi olabilir. Kullanıcı girdiği bu bilgiyle aslında sistemi veri bankasında yer alan bilgisine yönlendirir. Daha sonra kendisine ait biyometrik bilgiyle eşleştirme yapılarak kimlik doğrulaması gerçekleştirilir. Bu tür tanıma *"bire bir"* tanıma olarak isimlendirilebilir.

Kulak biyometrisi kullanılarak yapılan tanıma sistemlerinde, birçok yöntem kullanılabilmektedir. Temel Bileşen Analizi (TBA), Fisher Doğrusal Ayırıcılar Yöntemi (FDAY), Ayırıcı Ortak Vektör Yaklaşımı (AOVY) bu yöntemlerden en bilinenleri olarak sayılabilir. Ancak bu çalışma, TBA ve FDAY ile kulak biyometrisi ve bu yöntemler ile gömülü bir sistem gerçeklemesi üzerine yoğunlaşmaktadır.

2. KULAK BİYOMETRİSİ

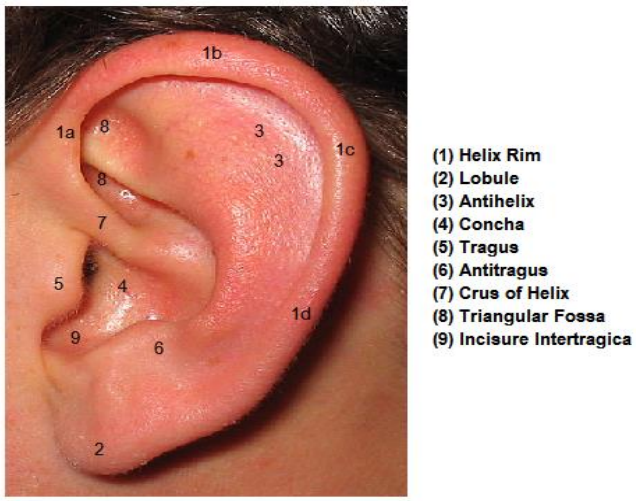
2.1 Kulak Anatomisi

Genel olarak insanda kulak, anne rahminde hamileliğin beşinci ve yedinci haftaları arasında belirgenleşmeye başlar. Ancak kulağın dış anatomisi, Şekil 2.1'de görüldüğü üzere, hamileliğin dokuzuncu haftasında tanınmaya ve görülmeye başlar.



Şekil 2.1 : Dokuzuncu haftada kulağın dış anatomisi.

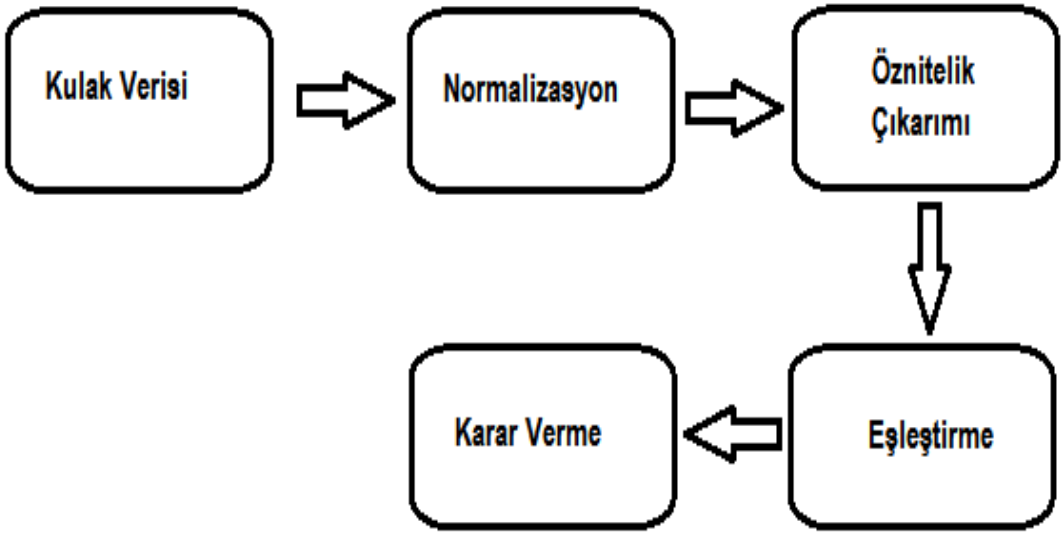
Yetişkin bir insandaki kulak anatomisi ve kulağın kısımları Şekil 2.2'de gösterilmiştir. Adli tıp raporlarına göre doğumdan dört ay sonraki kulak büyümesi büyük ölçüde doğrusaldır. Dördüncü aydan sekiz yaşına kadar kulakta gözlenen genişleme ise sekiz yaşından sonra gözlenen büyümeye göre ortalama beş kat daha fazladır. Bu büyüklük ortalama yetmiş yaşına kadar sabit kalır ve son olarak yetmiş yaşından sonra tekrar büyümeye başlar.



Şekil 2.2 : Yetişkin bir insanın kulak anatomisi ve kısımları.

2.2 Kulak Biyometri Sistemi

Bir kulak tanıma sistemi, giriş resminin belli bir sayıda öznitelik kümesine düşürüldüğü, böylece elde olan diğer resimlerle karşılaştırma yapılarak kimliğin tespit edilebildiği, tipik bir örüntü tanıma yöntemi olarak görülebilir. Kulaktan tanıma işlemi 2 boyutlu ve/veya kulak yüzeyinin 3 boyutlu resimleri ile yapılabilir. Bir kulak tanıma sisteminin iki modu vardır. Bunlar daha önce de açıklandığı üzere, "Doğrulama Modu" ve "Kimliklendirme Modu"dur. Özel bir yöntemden bağımsız olarak, genel anlamda bir kulak tanıma sisteminin aşamaları Şekil 2.3'te gösterilmiştir.



Şekil 2.3 : Kulak tanıma sisteminin aşamaları.

Şekil 2.3'te gösterilen aşamalar kısaca şu şekilde ele alınabilir:

Kulağın bulunması, kulak tanıma sisteminde öncelikli yapılması gereken işlemdir. Bu işlemde herhangi bir resimdeki kulağın tespiti yapılmalıdır. Verilen bir resimde, kulağın bulunduğu bölümü belirtmek için genelde dikdörtgen bir sınır kullanılır. Kulağın doğru bir şekilde bulunması, biyometrik sistemimiz açısından büyük önem taşımaktadır. Çünkü bu aşamada yapılan hatalar, sistemin yararlarını sekteye uğratarak başarısızlığa neden olabilecek seviyede olacaktır.

Normalizasyon aşaması, tespit edilmiş kulağın kalitesinin, dolayısıyla da resmin kullanılabilirliğinin artmasını sağlamak için bir rutine sokulur. Dahası, eğer gerek duyulursa, kulak resmi belli geometrik veya fotometrik düzeltmelere sokulduktan sonra normalizasyon işlemi gerçekleştirilir.

Normalize edilmiş görüntü karşılaştırma aşamasında direkt de kullanılabileceği gibi, çoğu sistemler, kulağı ifade etmek için belirgin özelliklerini ortaya çıkarır. Öznitelik çıkarımı, normalize edilmiş bir kulağın matematiksel bir modele uyarlanarak modelin uygulanması ve bu sayede kulağın ayırıcı özelliklerinin elde edilerek bunlara göre ifade edilmesi olarak algılanabilir.

Eşleştirme aşamasında, elde edilen öznitelikler veri bankasında tutulan bilgilerle karşılaştırılır. Buradaki amaç, girdi olarak kullanılan kulak resminin kime ait olduğunun tespitidir. En basit manada eşleştirme, veri bankasında alakasız olan kulak resimlerinin göz önüne alınmadan, iki kulak arasında en benzer çifti bularak işlemi gerçekleştirmektir. Sistem bu işlemi, her denediği eşleştirme için bir puan (Öklit uzaklığı,…) vererek de yapabilir. Eşleştirme puanı benzerlik derecesini belirleyecektir.

Son aşama olan karar aşamasında, eşleştirme puanları sonucun belirlenmesi için kullanılmaktadır. İşlemin doğrulama modunda, sistemin çıktısı "evet" veya "hayır" olarak oluşur. Buna karşılık kimliklendirme modunda ise, muhtemel eşleşmeler daha önce elde edilen eşleşme puanlarına göre sıralanır. Sıralama sonucunda eğer yeteri kadar yüksek puanlama oluşursa, eşleşme ve kimliklendirme işlemi gerçekleşecektir. Eşleşme puan kriterleri ve bunların sonuçlandırılması, sistemin programlanması aşamasında değiştirilebilecek işlemlerdir.

2.3 Kulak Biyometrisi Üzerine Daha Önce Yapılan Çalışmalar

Kulak biyometrisi üzerine yapılan en önemli çalışma Iannarelli'nin 1989 yılında yaptığı çalışmadır. Iannarelli rastgele seçilmiş 10.000'den fazla kulak üzerinde yaptığı çalışmalar neticesinde, her insanin farklı bir kulak yapısı olduğuna dair güçlü sonuçlar elde etmiştir. Iannarelli elde ettiği bu sonucun ikizler de bile geçerli olduğu sonucuna varmıştır.

Iannarelli çalışmasını yaparken, kulak üzerinde bulunan 12 nokta üzerinde yaptığı ölçümleri kullanmıştır. Iannarelli sistemi olarak adlandırılan mekanizmada kullanılan noktalar ve ölçüm uzaklıkları Şekil 2.4'te gösterilmiştir.



Şekil 2.4 : Iannarelli sistemi ölçüm yöntemi.

Iannarelli, tasarladığı sistemde sağ kulakları kullanmıştır. Sistemine göre öncelikle resimler hizalanmalı ve önceden tanımlanmış bir kalıba yerleşinceye kadar genişletilmelidir. Elde edilen resim üzerinden bu 12 nokta uzunlukları ölçülerek bir uzaklık sayısına atanır. Böylece kimliklendirme bu 12 uzaklık üzerinden yapılır.

Burge ve Burger 1998-2000 yılları arasında yaptıkları çalışmalar neticesinde, kulağın biyometrik bir nitelik olarak kullanılabileceğini öne sürmüşlerdir. Ancak Iannarelli'nin çalışmasından farklı olarak, Burge ve Burger Iannarelli Sisteminin otomatik bir tanıma sistemi için pek uygun olmadığı kannatine varmışlardır. Çünkü sistem ilk ölçüm noktasının bulunmasına dayalıdır ve eğer bu nokta doğru bir şekilde bulunamazsa, yapılacak kimliklendirme işlemi de doğru olmayacaktır. Bu nedenle 2000 yılında Burge ve Burger, Gauss'un piramid gösterimine dayalı bir yöntem geliştirmişlerdir. Kulak içerisinde bulunan kenarlar ve eğrilerden bir grafiksel gösterim elde etmişler, daha sonra grafik tabanlı bir eşleştirme algoritmasıyla kimliklendirme yapılmıştır. Ancak bu çalışmada sistemin başarı ve/veya performansı ile alakalı bir bilgi bulunmamaktadır.

1999 yılında Moreno, kulak biyometrisiyle ilgili tam otomatik bir sistem tanımlayan ilk bilimadamı olmuştur. Moreno çalışmasında, kulağın dış çeper noktaları, kulak şekli ve kırışıklıklar gibi birçok nitelik kullanmış ve bunları nöral sınıflandırıcıların

sonuçlarıyla birleştirmiştir. Sistemin testi için kullanılan veri bankası 2 türlüdür. İlkinde, her bir sınıfta 6 adet örnek olmak üzere 28 kişiden alınmış toplam 168 resim bulunmaktadır. İkincisi ise 20 farkı kişiye ait 20 resimden oluşmaktadır. Daha sonra Mu ve arkadaşları 2002 yılında Moreno'nun yaptığı bu çalışmayı genişletmişler ve kulağa ait öznitelik vektörlerini, dış kulak şekli ve iç kulak yapısının bir birleşimi olarak göstermişlerdir. Sınıflandırma ve kimliklendirme için ise nöral ağ kullanılmıştır.

2002 yılında Yuizono ve arkadaşları, kulak tanıma işlemini basit bir arama algoritması mantığıyla uygulamaya çalışmışlardır. Uyguladıkları yöntemde, Genetik Algoritma yardımıyla kulaklar arasındaki ortalama karesel hatayı azaltmayı amaçlamışlardır. Yapılan deneyler sonucunda, 110 farklı kişiye ait 660 resimde ortalama %99-100 oranında başarı elde etmişlerdir.

Yüz tanımada olduğu gibi, kulak tanımada da göz önüne alınması gereken bazı özellikler vardır. Bunlardan en önemlileri hareket ve renk faktörüdür. Ancak kullanılan çoğu algoritma renk faktörünü göz önüne almamaktadır. Bu çalışmada da renk faktörü dikkate alınmamıştır. Kullanılan resimler grayscale olarak tanımlanan gri tonlu resimlerdir. Ayrıca daha önce yapılan çalışmaların hiçbiri yaygın olarak bilinen tanıma algoritmalarını kullanmamışlardır.

Bu çalışmada ise, daha önce yüz tanıma için yaygın olarak kullanılan algoritmalardan TBA ve FDAY, kullanılmıştır. Çalışma, bu yöntemlerin kulak üzerinde daha etkili olduğunu göstermiştir.

3. KULAK TANIMA ALGORİTMALARI

Daha önce de belirtildiği üzere, geleneksel tanıma algoritmaları olarak Temel Bileşen Analizi, Fisher Doğrusal Ayırıcılar Yöntemi, vb. sayılabilir. Ancak bu yöntemlerin çalışmama konu olan kulak biyometrisine uygulanmalarına pek rastlanmamaktadır. Bu nedenle öncelikle bu yöntem ve algoritmalar teorik olarak incelenmiş; ardından bunların kulak biyometrisine uygulanabilirliği irdelenmiştir.

3.1 Temel Bileşen Analizi (TBA) / Principal Component Analysis (PCA)

TBA, çok değişkenli veri tabanlarının analizi için ve büyük boyutlu verilerin daha düşük boyutlara düşürülebilmesinde kullanılan en eski istatistiksel araç olarak tanımlanabilir. TBA ilk olarak 1901 yılında Pearson tarafından önerilmiş, teorinin geliştirilmesi ise Hotelling tarafından 1933 yılında yapılmıştır. TBA, çok boyutlu bir veri tabanı içerisindeki bilgilerin birbirleriyle ilişkisi, benzerlik veya farklılıklarını göstermek amacıyla kullanılabilir. TBA'nın en büyük avantajı ise, veri tabanı içerisindeki bilgilerin modeli bir kez tanımlandığında veri boyutu küçültülerek sıkıştırılır ve veri kaybı en aza indirilir. TBA'nın bu özelliği resimlerin de sıkıştırılmasında kullanılmaktadır. Bu sayede fazla bilgi kaybetmeden aynı resim tekrar elde edilebilir. Bunlardan farklı olarak TBA, gürültüye karşı düşük hassasiyet, bellek ve kapasite ihtiyaçlarının azalması, az boyutlu uzaylarda daha etkin çalışması gibi diğer avantajlara da sahiptir.

TBA tanıma alanında, uygulama kolaylığından dolayı en çok tercih edilen yöntemlerden biridir. Karhunen-Loeve (K-L) veya Hotelling Dönüşümü baz alınır. Ayrıca TBA, en az ortalama karesel hata oranı temelinde, fazlalıkları azaltmak için kullanılabilecek doğrusal bir metottur. Veriye ait koordinatların doğrusal dönüşümü yapılır.

Bu bölümde, TBA'nın teorisi ardından, nasıl uygulandığı örnek eşliğinde işlenmiştir. Son olarak da TBA'nın kulak biyometrisine uygulanması konusu detaylarıyla irdelenmiştir.

3.1.1 TBA'nın teorisi

TBA'da $\boldsymbol{x_1 \ldots x_n}$ olarak tanımlanan bir *n* boyutlu vektör kümesi, tek bir $\boldsymbol{x_0}$ vektörüyle ifade edilmeye çalışılır. Daha açık olmak gerekirse, bulunacak olan $\boldsymbol{x_0}$ vektörü ile bu vektör kümesine ait $\boldsymbol{x_k}$ vektörleri arasındaki karesel farkların toplamları minimum olmalıdır. TBA, bu işlemi gerçekleştirecek bir $\boldsymbol{x_0}$ vektörü bulmaya çalışır. Karesel hata kriteri J_0:

$$J_0(\boldsymbol{x}_0) = \sum_{k=1}^{n} \|\boldsymbol{x}_0 - \boldsymbol{x}_k\|^2 \quad \textbf{(3.1)}$$

olarak tanımlanırsa, amaç hata kriterini en aza indirmektir. Bu amaca ulaşmak için yapılması gereken en basit yöntem, $\boldsymbol{x}_0$'ı ortalama (μ) olarak ayarlamaktır. Veri kümesinin ortalaması ise şu şekilde hesaplanır:

$$\boldsymbol{\mu} = \frac{1}{n} \sum_{k=1}^{n} x_k \quad \textbf{(3.2)}$$

Yaptığımız varsayımlar şu şekilde doğrulanabilir:

$$J_0(\boldsymbol{x}_0) = \sum_{k=1}^{n} \|(\boldsymbol{x}_0 - \mu) - (\boldsymbol{x}_k - \mu)\|^2 \quad \textbf{(3.3a)}$$

$$J_0(\boldsymbol{x}_0) = \sum_{k=1}^{n} \|(\boldsymbol{x}_0 - \mu)\|^2 + \underbrace{\sum_{k=1}^{n} \|(\boldsymbol{x}_k - \mu)\|^2}_{x_0' dan\ bağımsız} \quad \textbf{(3.3b)}$$

Denklem (3.3b)'de bulunan ikinci toplam $\boldsymbol{x}_0$'dan bağımsız olduğundan J_0 kriteri $\boldsymbol{x}_0 = \mu$ seçimiyle minimize edilebilir.

Örnek ortalaması, veri kümesinin sıfır boyutlu bir gösterimidir. Elde bulunan veriler, örnek ortalaması etrafında olmak kaydıyla, bir doğru üzerine yansıtılırsa, daha etkin olan tek boyutlu bir gösterim elde edilir. $\boldsymbol{e}$, yansıtılacak doğru ile aynı doğrultuda bulunan bir birim vektör olsun. Bu durumda doğru denklemi:

$$\boldsymbol{x} = \boldsymbol{\mu} + a\boldsymbol{e} \quad \textbf{(3.4)}$$

olarak ifade edilir. Doğru denkleminde bulunan a, herhangi bir x noktasının μ ortalamasından uzaklığını veren skaler bir değerdir. Eğer $\boldsymbol{x}_k$, $\boldsymbol{\mu} + a_k\boldsymbol{e}$ olarak ifade edilirse, karesel hata minimize edilerek optimum a_k katsayılar kümesi bulunabilir.

$$J_1(a_1, \ldots, a_n, \boldsymbol{e}) = \sum_{k=1}^{n} \|(\boldsymbol{\mu} + a_k\boldsymbol{e}) - \boldsymbol{x}_k\|^2 = \sum_{k=1}^{n} \|a_k\boldsymbol{e} - (\boldsymbol{x}_k - \boldsymbol{\mu})\|^2 \quad \textbf{(3.5a)}$$

$$= \sum_{k=1}^{n} {a_k}^2 \|\boldsymbol{e}\|^2 - 2\sum_{k=1}^{n} a_k \boldsymbol{e}^T(\boldsymbol{x}_k - \boldsymbol{\mu}) + \sum_{k=1}^{n} \|(\boldsymbol{x}_k - \boldsymbol{\mu})\|^2 \quad \textbf{(3.5b)}$$

$\|\boldsymbol{e}\|$'nin 1 olduğu göz önüne alınır ve yukarıdaki denklem a_k'ya göre kısmi türev alındıktan sonra sıfıra eşitlenirse:

$$a_k = \boldsymbol{e}^T(\boldsymbol{x}_k - \boldsymbol{\mu}) \tag{3.6}$$

elde edilir. Geometrik olarak bakılırsa, yukarıdaki ifade $\boldsymbol{x}_k$ vektörünün $\boldsymbol{e}$ yönünde olan bir doğru üzerine, ortalama etrafında olmak kaydıyla yansıtıldığını gösterir. Bu sayede en küçük kareler probleminin de çözümü sağlanmış olur.

Elde edilen bu sonuç aslında daha ilginç bir problemi doğurur. Acaba yansıtılacak doğru için en iyi $\boldsymbol{e}$ yönü nedir? Ortaya çıkan bu problemin çözümüne *saçılma matrisi* olarak tanımlanan $\boldsymbol{S}$ kullanılır. Saçılma matrisi $\boldsymbol{S}$ şu şekilde tanımlanır:

$$\boldsymbol{S} = \sum_{k=1}^{n}(\boldsymbol{x}_k - \mu)(\boldsymbol{x}_k - \mu)^T \tag{3.7}$$

Yukarıda tanımlanan saçılma matrisi kovaryans matrisine benzer olmakla birlikte, kovaryans matrisinin $(n-1)$ katıdır. Denklem (3.6)'da bulunan a_k'yı Denklem (3.5)'de yerine koyarsak karesel hata kriteri şu hale gelir:

$$\begin{aligned} J_1(\boldsymbol{e}) &= \sum_{k=1}^{n} {a_k}^2 - 2\sum_{k=1}^{n} {a_k}^2 + \sum_{k=1}^{n}\|(\boldsymbol{x}_k - \boldsymbol{\mu})\|^2 \\ &= -\boldsymbol{e}^T\boldsymbol{S}\boldsymbol{e} + \sum_{k=1}^{n}\|(\boldsymbol{x}_k - \boldsymbol{\mu})\|^2 \end{aligned} \tag{3.8}$$

Açıkça görülüyor ki, $\boldsymbol{e}$ vektörü J_1'i minimize ederken $\boldsymbol{e}^T\boldsymbol{S}\boldsymbol{e}$'yi maksimize eder. $\boldsymbol{e}^T\boldsymbol{S}\boldsymbol{e}$, $\|\boldsymbol{e}\| = 1$ kısıtı altında *Lagrange çarpanları* kullanılarak maksimize edilir. λ'yı belirsiz bir çarpan olarak kabul edersek ve $\boldsymbol{u} = \boldsymbol{e}^T\boldsymbol{S}\boldsymbol{e} - \boldsymbol{\lambda}(\boldsymbol{e}^T\boldsymbol{e} - 1)$ denklemini $\boldsymbol{e}$'ye göre türevini alırsak:

$$\frac{\partial \boldsymbol{u}}{\partial \boldsymbol{e}} = 2\boldsymbol{S}\boldsymbol{e} - 2\lambda\boldsymbol{e} \tag{3.9}$$

elde edilir. Elde edilen bu eğim vektörü sıfıra eşitlenirse $\boldsymbol{e}$ saçılma matrisinin özvektörü olur.

$$\boldsymbol{S}\boldsymbol{e} = \boldsymbol{\lambda}\boldsymbol{e} \tag{3.10}$$

$\boldsymbol{e}^T\boldsymbol{S}\boldsymbol{e} = \boldsymbol{\lambda}\boldsymbol{e}^T\boldsymbol{e} = \boldsymbol{\lambda}$ olduğundan, $\boldsymbol{e}^T\boldsymbol{S}\boldsymbol{e}$'yi maksimize etmek için özellikle saçılma matrisinin en büyük özdeğerine karşılık gelen özvektörünü seçmek gerekir. Başka bir deyişle, elimizde bulunan verinin tek boyuta en iyi yansımasını bulmak için, saçılma matrisine ait en büyük özdeğere karşılık gelen özvektör seçilir. Seçilen özvektör üzerine ortalama etrafında verinin yansıtması yapılır.

Elde edilen bu sonuç, tek boyutlu bir yansımanın d'-boyuta yapılmasını da mümkün kılar. Verinin üzerine yansıtılacağı doğruların denklemleri genel manada şu şekilde belirtilebilir.

$$\boldsymbol{x} = \boldsymbol{\mu} + \sum_{i=1}^{d'} a_i \boldsymbol{e}_i \qquad \textbf{(3.11)}$$

Yukarıdaki denklemde $d' < d$ olarak kabul edilmiştir. Bu durumda ilgilendiğimiz J kriteri:

$$J_{d'} = \sum_{k=1}^{n} \left\| \left(\boldsymbol{\mu} + \sum_{i=1}^{d'} a_{ki} \boldsymbol{e}_i\right) - \boldsymbol{x}_k \right\|^2 \qquad \textbf{(3.12)}$$

$\boldsymbol{e}_1, \ldots, \boldsymbol{e}_{d'}$'ler saçılma matrisine ait d' adet en büyük özdeğere karşılık gelen özvektör olduğunda minimize edilmiş olur. Oluşan saçılma matrisinin en önemli özelliği, gerçek ve simetrik olmasıdır. Dolayısıyla elde edilen özvektörler birbirlerine diktir (ortogonal). Bu özvektörler, herhangi bir $\boldsymbol{x}$ özellik vektörünü tanımlamak üzere, baz vektörlerden oluşan bir küme oluştururlar. Denklem (3.11)'te gösterilen a_i katsayıları da $\boldsymbol{x}$ vektörünün *temel bileşenleri* olarak adlandırılır. Geometrik olarak açıklamaya çalışırsak, $\boldsymbol{x_1} \ldots \boldsymbol{x_n}$ olan veri noktalarının d-boyutlu hiperelipsoid şekilli bir gösteriminde, saçılma matrisinin özvektörleri bu hiperelipsoidlerin temel eksenleri olacaktır. TBA, nitelik uzayı boyutlarını, saçılmanın en yüksek olduğu doğrultu dikkate alınarak azaltır. Bir başka deyişle, TBA saçılmanın maksimum olduğu eksen/eksenlerle ilgilenir.

Her kişiye ait kulak resimleri, elde edilen bu özvektörlerin doğrusal bir kombinasyonu oluşturularak ifade edilebilir. Ayrıca daha önce de bahsi geçtiği gibi, boyut küçültme işlemi sonunda elde olan en büyük özdeğerlere ait özvektörler yardımıyla fazla bilgi kaybı olmadan kulak resimleri tekrar oluşturulabilir.

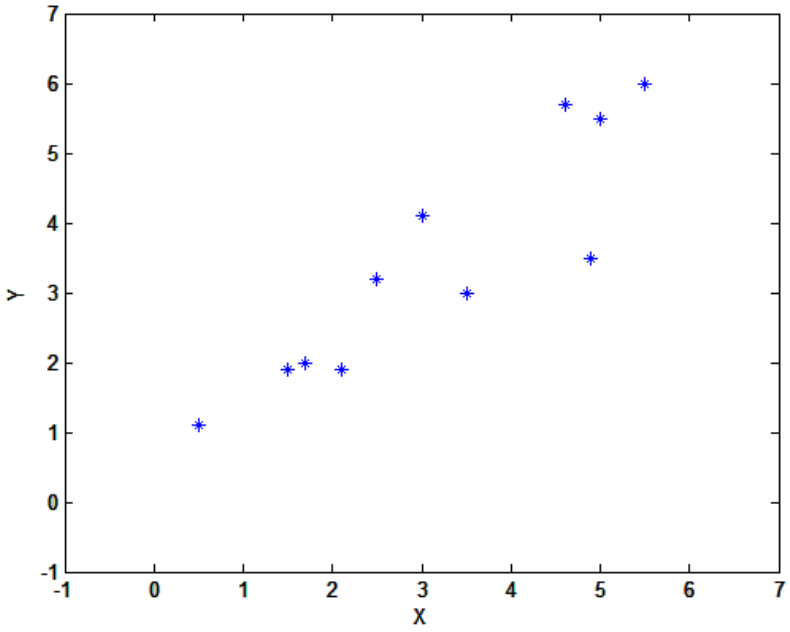
3.1.2 TBA'nın uygulanması

Bu bölümde TBA'nin teorisine bağlı kalınarak nasıl uygulanabildiği anlatılacaktır. Gösterim kolaylığı nedeniyle 2 boyutlu bir veri seçilmiştir. Bu sayede TBA'nın her aşamasında verilerin nasıl işlendiği ve değişimler gözlenebilecektir.

Örneğimizde kullanılacak veri noktaları Çizelge 3.1'de gösterilmiştir. Ayrıca Şekil 3.1'de de kullanılan veri noktalarının MATLAB Programı üzerinde çizilmiş gösterimi sunulmuştur.

Çizelge 3.1 : Örnek veri noktaları.

	P_1	P_2	P_3	P_4	P_5	P_6	P_7	P_8	P_9	P_{10}	P_{11}
x	0.5	1.5	1.7	2.1	2.5	3.5	3	4.9	5	4.6	5.5
y	1.1	1.9	2	1.9	3.2	3	4.1	3.5	5.5	5.7	6

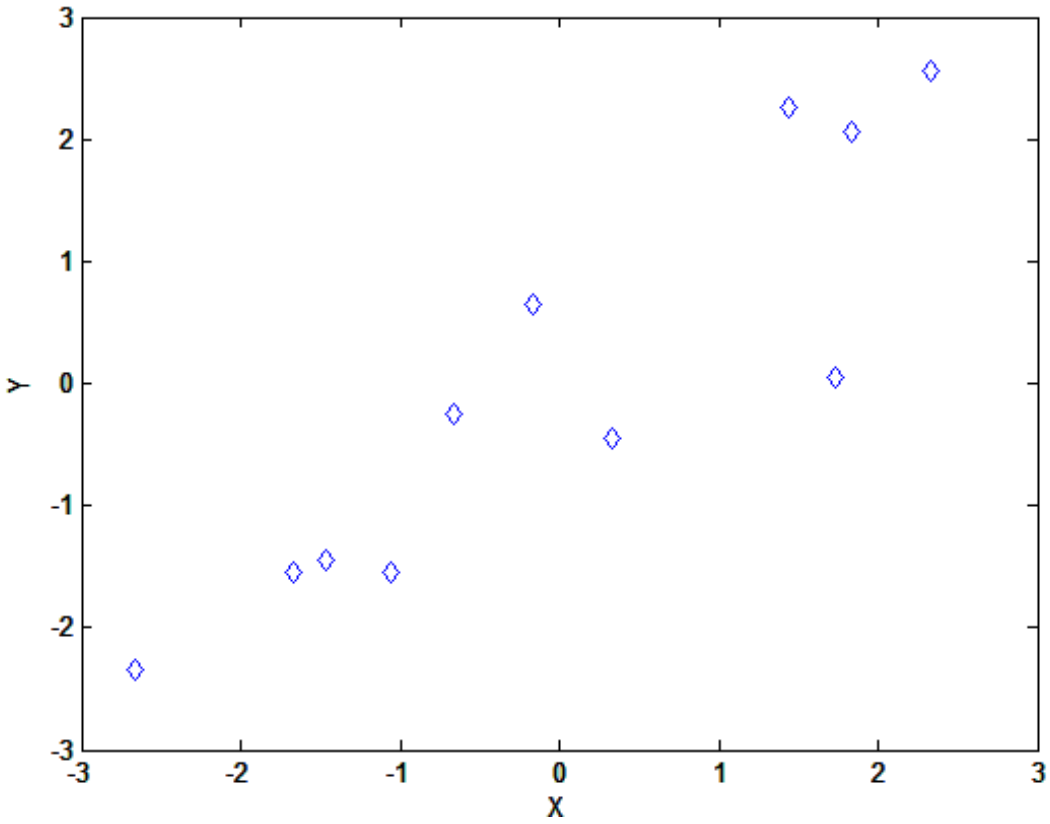


Şekil 3.1 : Örnek verinin çizilmiş hali.

TBA'nın uygulamalarda düzgün çalışabilmesi için, orijinal veride her boyuta ait noktadan o boyutun ortalamasının çıkarılması gerekir. Ortalama çıkarılmış veri noktaları, ortalaması sıfır olan yeni bir veri kümesi oluşturur. $\bar{x}$, x boyutunun ortalaması; $\bar{y}$ ise y boyutunun ortalaması olarak adlandırılır. Elimizde bulunan örneğe göre x boyutunun ortalaması 3.1636; y boyutunun ortalaması ise 3.4455'tir. Elde edilen ortalamaların veri noktalarından çıkarıldıktan sonra oluşan yeni veri kümesi Çizelge 3.2'de gösterilmiştir. Şekil 3.2'de ise, ortalama çıkarılmış veri noktalarının çizilmiş hali sunulmuştur.

Çizelge 3.2 : Ortalama çıkarılmış veri noktaları.

	$P_{1'}$	$P_{2'}$	$P_{3'}$	$P_{4'}$	$P_{5'}$	$P_{6'}$	$P_{7'}$	$P_{8'}$	$P_{9'}$	$P_{10'}$	$P_{11'}$
$\bar{x}$	-2.66	-1.66	-1.46	-1.06	-0.66	-0.33	-0.16	1.73	1.83	1.43	2.33
$\bar{y}$	-2.34	-1.54	-1.44	-1.54	-0.24	-0.44	0.65	0.05	2.05	2.25	2.55



Şekil 3.2 : Ortalama çıkartılmış veri noktalarının gösterimi.

Ortalama çıkarılmış veri kümesine ait saçılma (kovaryans) matrisi şöyle tanımlanır.

$$C = \frac{1}{n-1} AA^T \qquad \textbf{(3.13)}$$

$A = [P'_1 P'_2 \dots P'_n]$ olarak tanımlanır. Örneğimizde A matrisi 2 satırlı ortalama çıkarılmış matristir. Örneğimiz 2 boyutlu olduğundan, bulunacak kovaryans matrisi de 2x2 boyutlu olacaktır. Halihazırdaki örnek için hesaplanan kovaryans matrisi, $C = \begin{bmatrix} 2.7625 & 2.5558 \\ 2.5558 & 2.8887 \end{bmatrix}$'dir. Dikkat edilecek olursa, daha önce de bahsedildiği gibi kovaryans matrisi gerçek ve simetrik olarak karşımıza çıkmıştır. Matrisin köşegen harici elemanları pozitif çıktığından veri noktalarımız hakkında "*birlikte artarlar*" gibi bir yorum yapmak mümkündür.

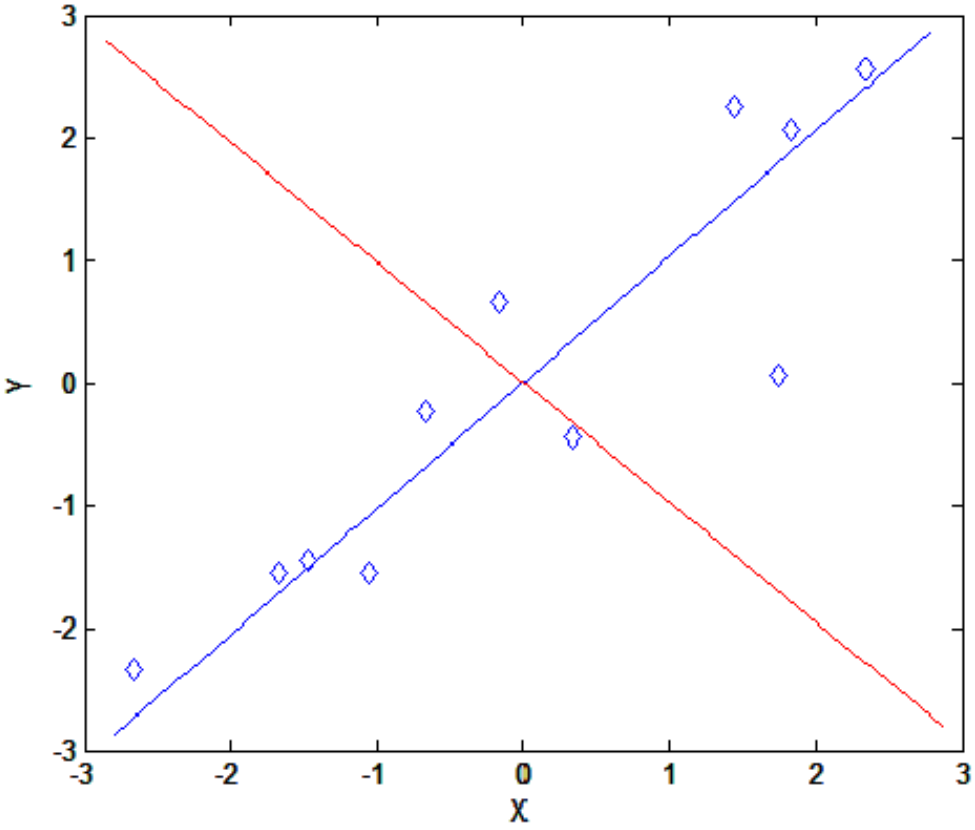
Kovaryans matrisin kare matris olması nedeniyle, elde edilen matrisin özdeğer ve özvektörleri kolayca hesaplanır. Elde edilecek özdeğer ve özvektörler aslında veri noktalarımız hakkında yararlı bilgiler taşımaktadır. Örneğimizin kovaryans matrisine ait özdeğer ve özvektörler şöyle bulunmuştur.

$$\text{özdeğerler} = \begin{bmatrix} 0.2690 \\ 5.3822 \end{bmatrix}$$

$$\text{özvektörler} = \begin{bmatrix} -0.7158 & 0.6983 \\ 0.6983 & 0.7158 \end{bmatrix}$$

Elde ettiğimiz özvektörler, birim özvektörlerdir ve uzunlukları 1'dir. bu özellik TBA'nın uygulanması aşamasında çok önemlidir.

Elde edilen özvektörlerin, örnek veri üzerindeki önemi büyüktür. Şekil 3.3'te ortalama çıkarılmış örnek noktalar, kovaryans matrisinden elde edilen özvektörlerle gösterilmektedir. Kırmızı ve mavi çizgiler özvektörleri ifade eder. Şekil incelenirse, elde edilmiş 2 adet özvektörün, örnek noktaların ilişkilerini belirlediği görülür. Veri noktalarımız güçlü bir modele sahiptir. Şekilden de gözlendiği üzere, kovaryans matrisinin köşegen olmayan elemanlarından elde ettiğimiz varsayım burada doğrulanmıştır. x ve y noktaları birlikte artmaktadırlar. Şekil, özvektörlerin birbirlerine dik olduğunu da açıkça göstermektedir.



Şekil 3.3 : Özvektörlerle birlikte ortalama çıkarılmış veri noktaları.

Plotlanmış özvektörlerden de açıkça görüleceği üzere, mavi ile çizilmiş özvektör veri noktaları arasında uzanmaktadır. Bu durum da gösteriyor ki, veri noktalarımız bu özvektör doğrusu üzerinde birbirleriyle ilişkilidir. Kırmızı ile çizilmiş olan ikinci özvektör ise, daha az öneme sahip olan diğer ilişkiyi vermektedir. Dolayısıyla, kovaryans matrisinin özdeğer ve özvektörlerini bulmak, veriyi karakterize eden doğruların elde edilmesini sağlar. Sonraki adımlar ise, verinin dönüştürülmesi ve elde edilen doğrularla ifade edilmesini sağlar.

Bileşenlerin seçilmesi ve özellik vektörünün oluşturulması aşaması, veri sıkıştırmasının ve boyut küçültmenin yapılacağı adımdır. Bir önceki adımda elde edilen özdeğerlere bakıldığında birbirlerinden oldukça farklı oldukları görülür. Buradan da anlaşılacağı üzere, büyük özdeğere ait özvektör, veri kümemizin temel bileşenidir. Verilen örnekte, veri noktaları arasında uzanan özvektör temel bileşen ve en büyük ilişkiyi sağlayandır.

Genelde uygulamalarda izlenen sıra şu şekildedir. Özdeğerler ve bunlarla ilişkili özvektörler bulunduktan sonra, özdeğerler ve dolayısıyla özvektörler büyükten küçüğe doğru sıralanır. Önem sırasına göre temel bileşenler bu şekilde sıralandıktan sonra, uygulamaya/isteğe bağlı olarak bazı bileşenler dikkate alınmaz. Bu durumda bazı bilgi kayıpları yaşanabilir; ancak özdeğer ne kadar küçükse kaybedilen bilgi de o kadar küçüktür. Eğer bazı bileşenler dikkate alınmazsa, sonuçta oluşan yeni veri kümesinin boyutu öncekine göre daha küçük olacaktır. Başka bir deyişle, eğer d-boyutlu bir veriniz varsa, d adet özdeğer ve özvektör elde edilir. Bunlardan sadece p tanesi dikkate alınırsa, sonuçta elde edilen yeni veri kümesi p-boyutlu olacaktır.

Dikkate alınan özvektörleden nitelik vektörleri matrisi oluşturulur. Nitelik vektörü matrisi, dikkate alınan özvektörlerin sütunlar halinde sıralanması ile elde edilir.

$$Nitelik\ Vektör\ Matrisi = [eig_1, eig_2, \dots eig_p] \qquad \textbf{(3.14)}$$

Örneğimizde iki adet özvektör elde edilmiştir, dolayısıyla nitelik vektör matrisi 2 sütundan oluşur. Ancak, önemsiz olan özdeğere ait özvektör dikkate alınmazsa tek sütunlu bir nitelik vektör matrisi elde edilir.

$$\begin{bmatrix} -0.7158 & 0.6983 \\ 0.6983 & 0.7158 \end{bmatrix} \text{ veya } \begin{bmatrix} 0.6983 \\ 0.7158 \end{bmatrix}$$

Buradan yeni veri kümesinin elde edilmesine geçilir. Önemine göre seçilen bileşenlerden (özvektör) oluşan nitelik vektör matrisi kurulduktan sonra transpozu alınır ve soldan, ortalama çıkarılmış orijinal veri kümesi transpozu ile çarpılır.

$$Sonuç\ Verisi = Nitelik\ Vektör\ Matrisi^T x Ortalama\ Çıkarılmış\ Veri^T \quad \textbf{(3.15)}$$

Yapılan matris çarpımındaki amaç, veri kümesinin elde edilmiş olan özvektörler cinsinden ifadesinin bulunmasıdır. Bilindiği üzere orijinal veri kümemiz x ve y olmak üzere 2 eksene/boyuta sahiptir. Bu nedenle elde olan veri kümesini herhangi 2 eksene göre ifade etmek mümkün olacaktır. Ancak dikkat edilmesi gereken husus, bu eksenlerin birbirlerine dik olması gerektiğidir. Bu nedenle, sonuç verisi elde edilirken özvektörler kullanılır. Elde edilen sonuç verisi artık x ve y eksenlerine değil, kovaryans matrisinden elde edilen özvektörlere bağlı olarak ifade edilir. Eğer bazı özvektörler dikkate alınmazsa, bu durumda ise sonuç verisi, elde tutulan özvektörlere bağlı olarak gösterilir.

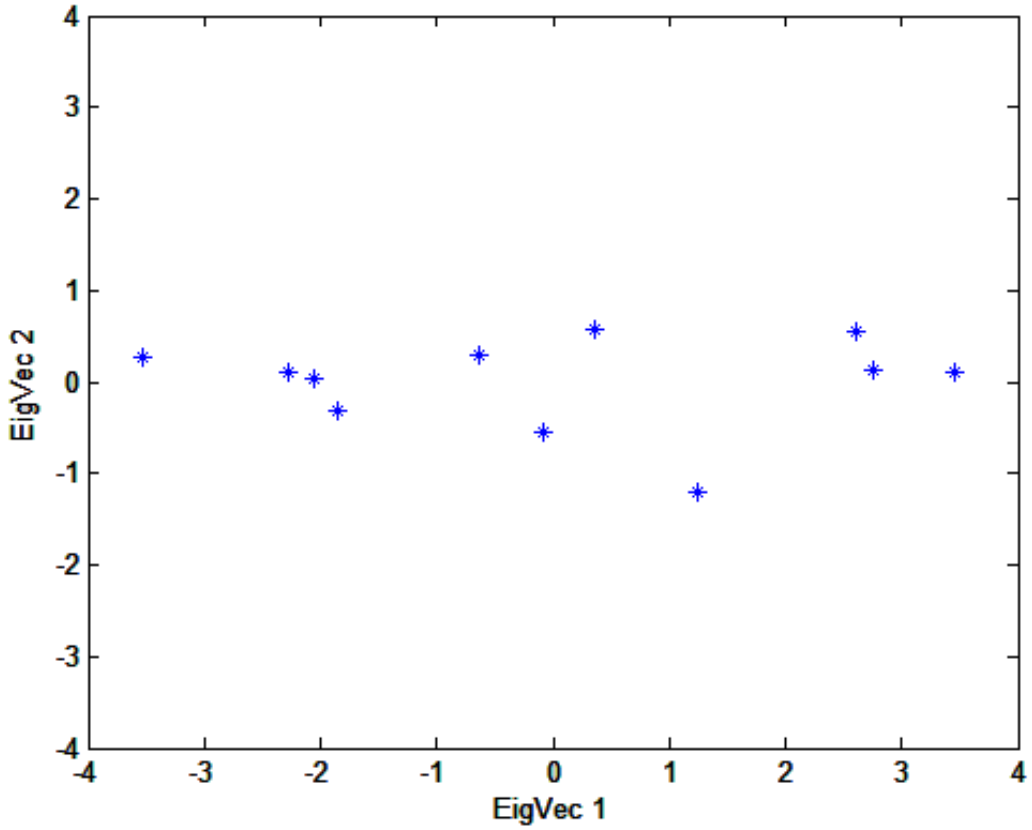
Örneğimizde bu sonucu göstermek üzere her iki durum da işlenmiştir. Tek bir özvektör seçimi ve her iki özvektörün seçimiyle oluşan sonuç verileri sırasıyla Çizelge 3.3 ve Çizelge 3.4'te olduğu gibidir. Elde edilen sonuç verilerinin MATLAB ortamında çizilmiş şekilleri Şekil 3.4 ve Şekil 3.5'te sırasıyla gösterilmiştir.

Çizelge 3.3 : Her iki temel bileşen kullanılarak elde edilen sonuç verisi.

	S_V_x	S_V_y		S_V_x	S_V_y
PS_1	0.2687	-3.5389	PS_8	-1.2048	1.2516
PS_2	0.1116	-2.2680	PS_9	0.1203	2.7530
PS_3	0.0382	-2.0567	PS_{10}	0.5463	2.6168
PS_4	-0.3179	-1.8490	PS_{11}	0.1116	3.4600
PS_5	0.3036	-0.6391			
PS_6	-0.5518	-0.0840			
PS_7	0.5742	0.3542			

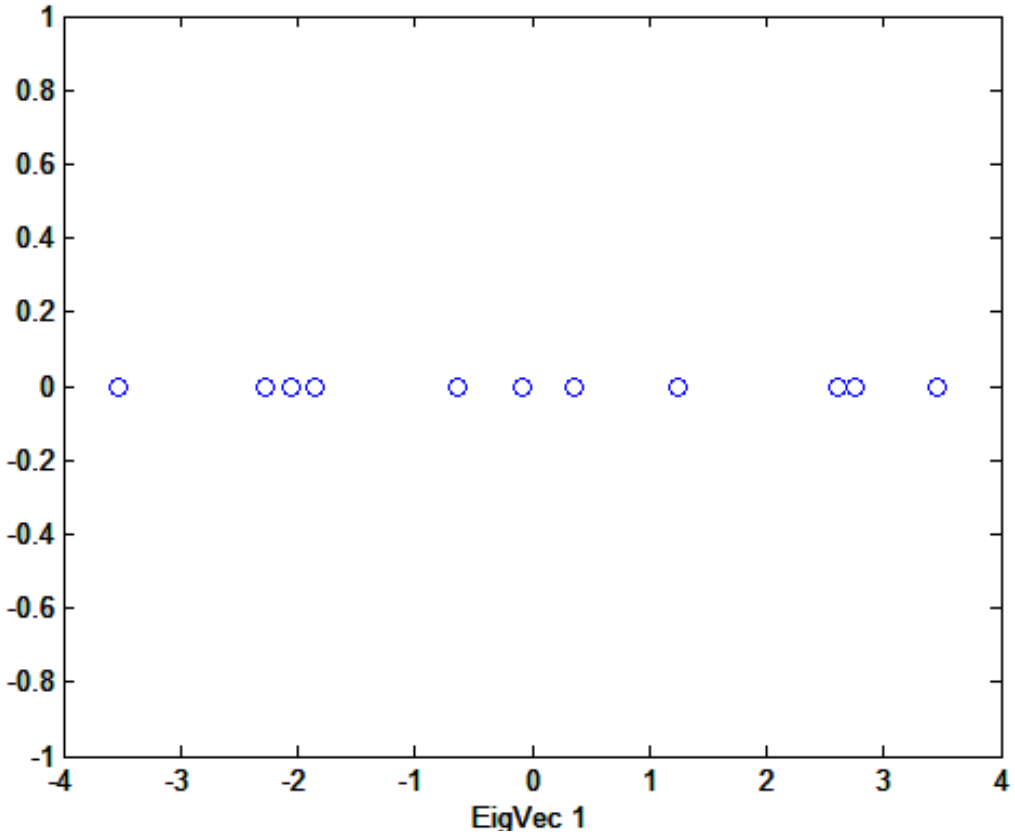
Çizelge 3.4 : En büyük temel bileşen kullanılarak elde edilen sonuç verisi.

	S_V
PS_1	-3.5389
PS_2	-2.2680
PS_3	-2.0567
PS_4	-1.8490
PS_5	-0.6391
PS_6	-0.0840
PS_7	0.3542
PS_8	1.2516
PS_9	2.7530
PS_{10}	2.6168
PS_{11}	3.4600



Şekil 3.4 : İki temel bileşen kullanılarak elde edilen sonuç verisi.

Şekil 3.4'te gösterilen sonuç verisi, dikkat edilecek olursa, başta çizilen ortalama



Şekil 3.5 : Tek temel bileşen kullanılarak elde edilen sonuç verisi.

çıkarılmış verinin döndürülmüş şeklidir. İki temel bileşen de kullanıldığından dolayı, sonuç verisinde modelle alakalı bilgi kaybı yaşanmamıştır. Sonuç verisinin yeni eksenleri ise, temel bileşenlerdir (kovaryans matrisinin özvektörleri).

Şekil 3.5'te gösterilen sonuç verisi ise, sadece en büyük temel bileşen kullanılarak elde edilen sonuç verisi olması nedeniyle tek bir doğru üzerinde görülmektedir. Bu dönüşümdeki eksen, en büyük özdeğere ait özvektördür. Çizelge 3.4'e bakıldığında da, elde edilen sonuç verisinin tek boyutlu olduğu görülür. Elde edilen boyuta ait sonuç verisi ise, seçilen özvektöre bağlı olarak Çizelge 3.3'ün ikinci sütundur. Sonuç olarak daha az öneme sahip olan ikinci temel bileşen dikkate alınmamıştır.

Bu noktaya kadar yapılanlar özetlenecek olursa, yapılan işlem elimizdeki veri kümesi, kendi aralarındaki modele göre dönüştürülmüştür. Dönüşümün temel eksenleri de kovaryans matrisinden elde edilen özvektörlerdir (temel bileşenler). Veri kümesi eksenleri başlangıçta x ve y tarafından oluşturulurken, dönüşüm sonunda tercihe bağlı olarak, her iki temel bileşen veya en önemli temel bileşen yeni eksen/eksenleri oluşturmaktadır. Ayrıca başlangıçtaki veri kümesi, eğilim eksenine göre fazla bilgi vermez. Örneğin, noktaların eğilim eksenine uzaklıkları veya üstünde

ya da altında olması gibi… Ancak dönüşüm sonunda elde edilen sonuç verisi, istenilen bu tür bilgileri de kolayca gösterme yetisine sahiptir.

Eski verinin tekrar elde edilmesi, TBA'da veri sıkıştırması kullanıldığı takdirde büyük önem arz etmektedir. Daha önce anlatılanlardan da bilineceği üzere, orijinal veri kümesinin tam olarak tekrar elde edilmesi, dönüşüm esnasında tüm temel bileşenlerin kullanılmasıyla mümkün olur. Aksi halde, bazı kayıplar olacaktır.

Denklem (3.15) hatırlanacak olursa, sonuç verisi, nitelik vektör matrisi ve ortalama çıkarılmış veri kullanılarak elde edilmişti. Bu noktada bazı matris işlemleri ile ortalama çıkarılmış veri tekrar elde edilebilir. Denklem (3.15)'in her iki tarafını nitelik vektör matrisi transpozunun tersiyle çarpalım. Yapılan işlem ve elde edilen sonuç şöyle olur.

$$\left(\begin{pmatrix} Nitelik \\ Vektör\ Matrisi \end{pmatrix}^T\right)^{-1} x \begin{pmatrix} Sonuç \\ Verisi \end{pmatrix}$$

$$= \left(\begin{pmatrix} Nitelik \\ Vektör\ Matrisi \end{pmatrix}^T\right)^{-1} x \begin{pmatrix} Nitelik\ Vektör \\ Matrisi \end{pmatrix}^T x \begin{pmatrix} Ortalama \\ Çıkarılmış\ Veri \end{pmatrix}^T$$

$$= I\ x\ Ortalama\ Çıkarılmış\ Veri^T$$

$$\rightarrow \begin{pmatrix} Ortalama \\ Çıkarılmış\ Veri \end{pmatrix}^T = \left(\begin{pmatrix} Nitelik \\ Vektör\ Matrisi \end{pmatrix}^T\right)^{-1} x \begin{pmatrix} Sonuç \\ Verisi \end{pmatrix} \qquad \textbf{(3.16)}$$

Denklem (3.16)'daki nitelik vektör matris transpozunun tersi, tüm temel bileşenler kullanıldığında transpozuna eşit olacaktır. Çünkü temel bileşenlerin hepsi birim uzunluktadır. Bu durumda Denklem (3.16),

$$\begin{pmatrix} Ortalama \\ Çıkarılmış\ Veri \end{pmatrix}^T = \left(\begin{pmatrix} Nitelik \\ Vektör\ Matrisi \end{pmatrix}^T\right)^{\mathrm{T}} x \begin{pmatrix} Sonuç \\ Verisi \end{pmatrix}$$

$$\begin{pmatrix} Ortalama \\ Çıkarılmış\ Veri \end{pmatrix}^T = \begin{pmatrix} Nitelik \\ Vektör\ Matrisi \end{pmatrix} x \begin{pmatrix} Sonuç \\ Verisi \end{pmatrix} \qquad \textbf{(3.17)}$$

halini alır. Ancak orijinal veri kümesini elde etmek için ortalamayı da eklemek gerekir.

$$\begin{pmatrix} Orjinal \\ Veri \end{pmatrix} = \begin{pmatrix} Nitelik \\ Vektör\ Matrisi \end{pmatrix} x \begin{pmatrix} Sonuç \\ Verisi \end{pmatrix} + Ortalama \qquad \textbf{(3.18)}$$

Denklem (3.18)'e istinaden her iki temel bileşen kullanılarak MATLAB ortamında tekrar elde edilen orijinal veri kümesi Çizelge 3.5'te gösterilmiştir. Çizelge 3.6'da ise tek bir temel bileşen kullanılarak elde edilen orijinal veri kümesi gösterilmiştir.

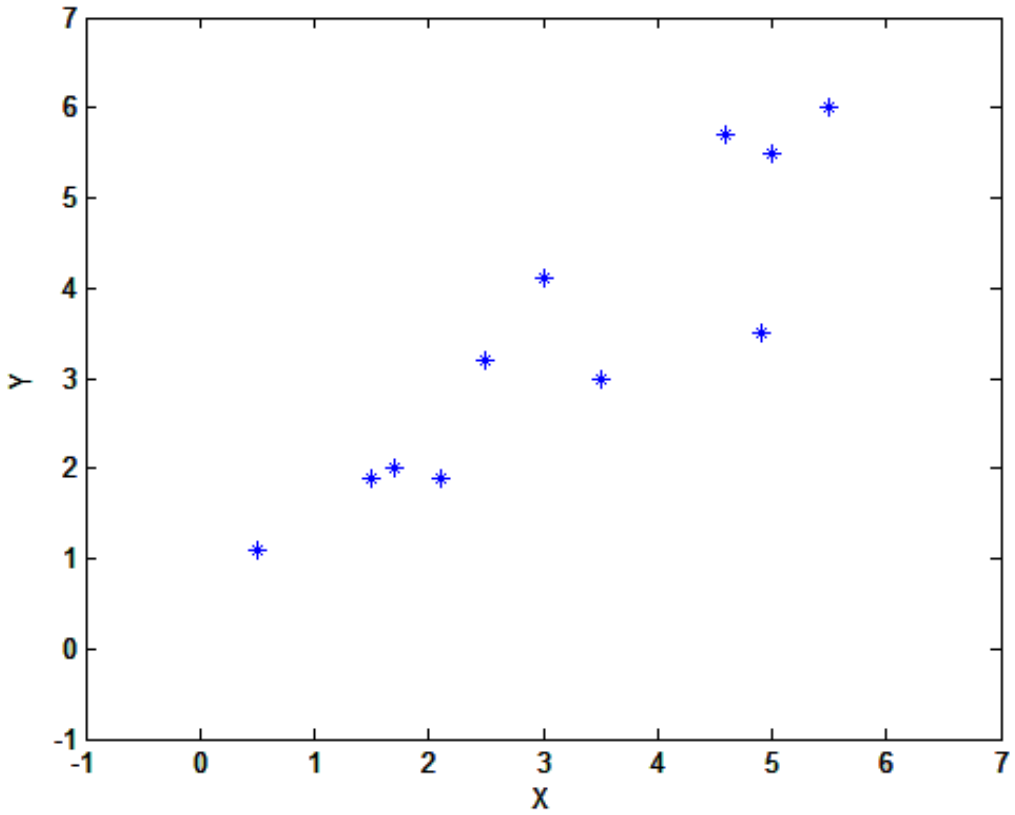
Çizelge 3.5 : Her iki temel bileşen kullanılarak elde orijinal veri kümesi.

	X	Y
P_1_orj	0.5	1.1
P_2_orj	1.5	1.9
P_3_orj	1.7	2
P_4_orj	2.1	1.9
P_5_orj	2.5	3.2
P_6_orj	3.5	3
P_7_orj	3	4.1
P_8_orj	4.9	3.5
P_9_orj	5	5.5
P_{10}_orj	4.6	5.7
P_{11}_orj	5.5	6

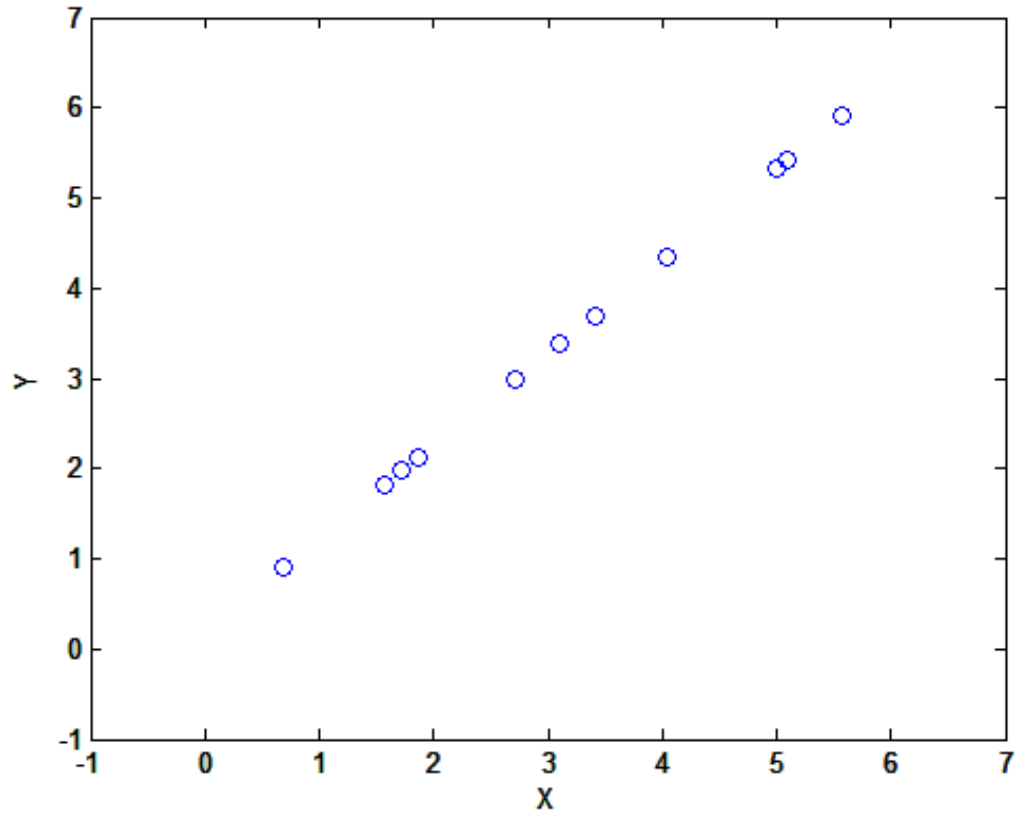
Çizelge 3.6 : Tek bir temel bileşen kullanılarak elde orijinal veri kümesi.

	X	Y		X	Y
P_1_orj	0.6923	0.9124	P_{10}_orj	4.9910	5.3185
P_2_orj	1.5799	1.8221	P_{11}_orj	5.5799	5.9221
P_3_orj	1.7274	1.9733			
P_4_orj	1.8725	2.1220			
P_5_orj	2.7173	2.9880			
P_6_orj	3.1050	3.3854			
P_7_orj	3.4110	3.6990			
P_8_orj	4.0377	4.3413			
P_9_orj	5.0861	5.4160			

Çizelge 3.6'dan da görüldüğü üzere, sadece en önemli temel bileşen kullanıldığında bazı bilgi kayıpları ortaya çıkmıştır. Tekrar elde edilen orijinal verilerin MATLAB programında plotlanmış şekilleri sırasıyla Şekil 3.6 ve Şekil 3.7'de gösterilmiştir.



Şekil 3.6 : İki temel bileşen kullanılarak elde orijinal veri.



Şekil 3.7 : Tek temel bileşen kullanılarak elde edilen orijinal veri.

Şekil 3.6 ve Şekil 3.7'den açıkça görülüyor ki, önemli olmayan temel bileşen dikkatealınmadığında orijinal veriyle alakalı kayıplar yaşanmaktadır. Örnekte oluşan kayıp, orijinal verinin eğilim doğrusuna göre üstte veya altta olduğunu gösteren konum bilgisi olmuştur. Şekil 3.6'da örnek verimiz tüm bilgileriyle beraber kayıpsız tekrar elde edilmiştir.

3.1.3 TBA'nın kulak resimlerine uygulanması

Daha önce, TBA'nın teorisi ve basit bir örnekle birlikte elde bulunan veri üzerinde nasıl etkileri olduğu incelenmiştir. Bu kısımdaki ana hedef, TBA'nın kulak resimleri üzerinde nasıl uygulanabileceği ile ilgili olacaktır.

Teori ve örnekli kısımda da belirtildiği üzere, girişte verilen veri kümesinin, birbirleri arasında bulunan bazı modeller çerçevesinde dönüştürülmesi mümkün olmaktadır. Bu çalışma çerçevesinde ele alınan giriş veri kümesi *kulak resimleridir*. Kulak resimleri üzerinde olabilecek modeller şekle, uzaklıklara veya renge bağlı olabilir. Temel alınan bu modellerde TBA ile elde edilen temel bileşenlere *özkulak* denir.

TBA teorisine göre eğitim kümesinde bulunan her orijinal kulak resmi, kendisiyle alakalı bir özkulağa dönüştürülür. Dahası, dönüştürülmüş bu özkulaklar yardımıyla daha önce örnekte olduğu gibi orijinal kulak resimleri tekrar elde edilebilir. Özkulaklar, herbir kulağa ait karakteristik niteliklerdir. Bu nedenle, eğer doğru şekilde geri dönüşüm işlemine tabi tutulurlarsa, birebir orijinal resmi elde edebiliriz. Her bir özkulak orijinal kulağa ait belli bir niteliği gösterir; ayrıca belli bir ağırlığa/öneme bağlı olarak bulunurlar. Bu nitelikler orijinal resimlerde görülmeyebilir ancak gözle görünen bir niteliğe sahip olan özkulağın önemi de büyük olacaktır. Yani elde edildiği resme ait bilgi oranı, sahip olduğu ağırlıkla/önemle orantılıdır. Bu nedenle, orijinal kulak tekrar elde edilirken, ağırlığı/önemi büyük özkulağın geri dönüşümde etkisi de büyük olacaktır. Eğer özkulaklardan geri dönüşüm sonunda, hiçbir bilgi kaybı olmadan tamamen orijinal resmin geri getirilmesi isteniyorsa, özkulaklar ağırlıklarına göre toplanmalıdırlar. Orijinal kulak resminden elde edilen özkulaklardan en önemli 10 adedini kullanarak tekrar elde edilen kulak resmine ait örnek Şekil 3.8'te gösterilmiştir.

Teoride orijinal kulak resminin tekrar elde edilmesi için tüm özkulakların kullanılması gerekse de; pratikte bu çoğunlukla uygulanmamaktadır. Bu nedenle de özkulaklardan elde edilen resimler, orijinal resmin yaklaşığı olarak görülür. Bu

durum Şekil 3.8'te açıkça görülmektedir. Sonuçta görülen kayıplar, özkulakların en önemlilerinin seçilmesiyle minimize edilebilir.



Şekil 3.8 : 80x60'lık orijinal kulak resmi ve 10 adet temel bileşen kullanılarak elde edilen kulak resmi.

Özkulakların sahip olduğu ağırlıkların öneminden bahsedilmişti. Özkulaklar yardımıyla hesaplanan ağırlıklar ile orijinal kulak tekrar elde edilerek sonuca gidilebildiği gibi, bunun tersi de mümkündür. Yani elde edilen ağırlıklar ve özkulaklar yardımıyla orijinal resim elde edilmeden de tanıma işlemi gerçekleştirilebilir. Ağırlıklar kulaklarla ilgili 2 önemli hususu tespit etmede kullanılabilir.

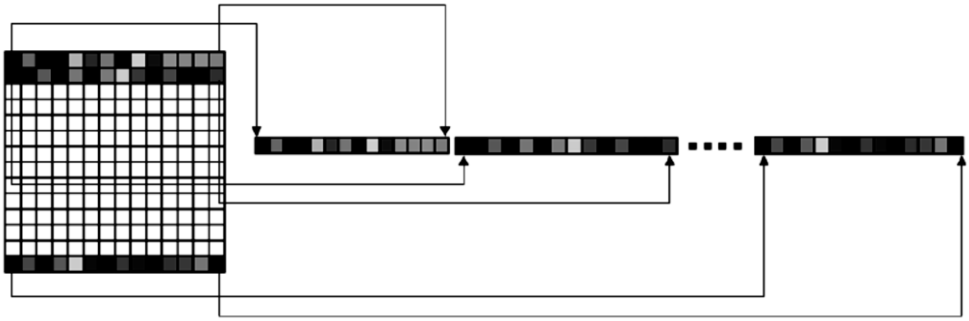
1. Ağırlıklar işleme tabi tutulan resmin kulak olup olmadığını tespitte kullanılabilirler. Elde edilen ağırlık, bilinen kulak resimlerinden elde edilen ağırlıktan çok farklı ise söz konusu resim kulak değildir.

2. Benzer kulaklar benzer niteliklere sahiptir. Dolayısıyla da bunlardan hesaplanan ağırlıklar da birbirlerine yakın olacaktır. Bu yakınlık, kulak resimlerinin kümelenmesine neden olur. Sonuçta, birbirine yakın ağırlıklara sahip tüm kulaklar benzerdir.

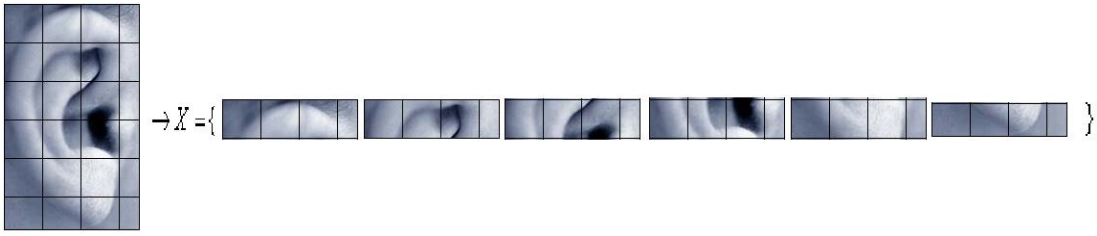
İki boyutlu olarak elde edilen kulak resimleri, TBA'ya tabi tutulmadan önce vektör haline getirilmelidirler.

Bir kulak resminin $K(x,y)$ olmak üzere iki boyutlu ve W piksel genişliğinde, H piksel yüksekliğinde olduğu varsayılsın. Bu durumda, resimde bulunan piksellerin sayısı $W\ x\ H$ olacaktır. Dolayısıyla da elde edilecek kulak vektörü boyutu $W\ x\ H$ olacaktır. Kulak resimleri her satır bir diğerinin ardından yanına birleştirilecek

şekilde bir satır vektörü formuna dönüştürülür. Örneğin 128 x 128'lik bir resim toplamda 16.384-boyutlu bir satır vektörü olarak ifade edilebilir. Bir kulak resminin satır vektörü formuna dönüşümü Şekil 3.9 ve Şekil 3.10'da gösterilmiştir.

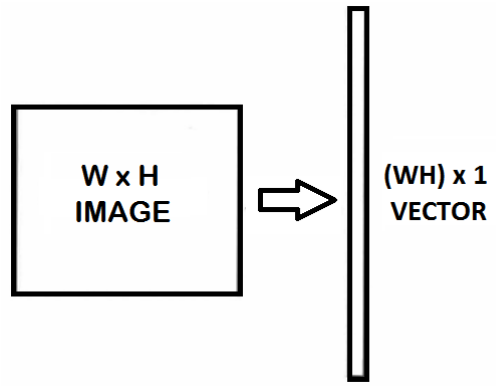


Şekil 3.9 : Resim verisinin satır vektörü formuna dönüşümü.



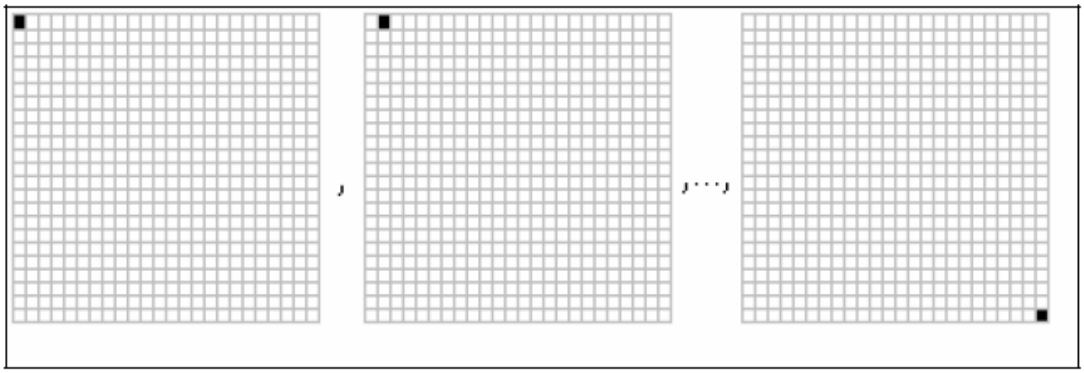
Şekil 3.10 : Kulak resminin kulak satır vektörüne dönüşümü.

Kulak resminde bulunan tüm sütunların ise alt alta sıralanmasıyla, $W\ x\ H$ boyutlu bir sütun vektörü elde edilir.



Şekil 3.11 : W x H boyutlu bir kulak resminin (WH) x 1 boyutlu sütun vektörüne dönüşümü.

Kulaklara ait resimler, eğitim esnasında oluşturulan bir kulak uzayına aittirler. Uzay içerisinde bulunan her resmin boyutu $W\ x\ H$'dir.



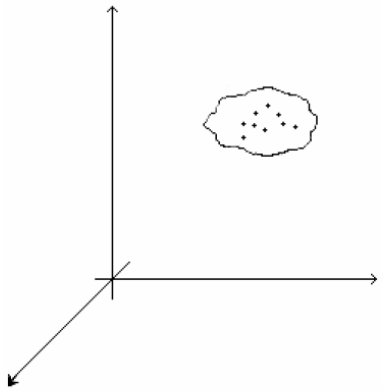
Şekil 3.12 : Kulak uzayı.

Ayrıca oluşturulan bu kulak uzayı TBA ile elde edilen özvektörler dikkate alınarak oluşturulur. Bu özvektörler orijinal resme ait kovaryans matrisinin özvektörleridir ve görünürde kulağa benzerler.



Şekil 3.13 : Örnek özkulak resimleri.

Kulak uzayı içerisinde birbirine benzer kulakların komşuluğuyla bir de alt uzaylar oluşur. Bu altuzaylara ait kulak vektörleri arasındaki uzaklık diğerlerine göre daha azdır. Bu durum, mevcut altuzay içerisindeki kulak resimlerinin muhtemelen aynı kişiye ait olduğunu gösterir.



Şekil 3.14 : Kulak uzayı ve komşuluk.

Kulak uzayı içerisinde, ağılığı fazla olan bileşenler birbirlerine daha yakın konumlanırlar. O yüzden, uzay içerisinde gereksiz fazladan bileşenler oluşur.

Gereksiz olan bileşenlerin tanıma işleminde katkıları yoktur. TBA varyansı dikkate aldığından, boyut küçültme işlemi sırasında, hesaplanan varyansın mümkün olan en az bileşen yardımıyla mümkün olduğunca büyük olmasına çalışır. Bilindiği üzere, büyük boyutlu verilerle çalışmak her zaman zahmetli ve zaman kaybına sebep olur. Özellikle kulak tanıma sistemlerinde eğitim esnasında işlenilmesi gereken çok sayıda resim olduğu da göz önüne alındığında, herbir resmin boyutunun küçültülmesi mantıklı olacaktır.

TBA ile tanıma sisteminin 2 aşaması vardır.

1. Eğitim aşaması
2. Test ve tanıma aşaması

TBA'nın eğitim aşamasında, kullanılacak kulak resimleri toplandıktan sonra, tüm resimler özkulaklara dönüştürülür. Ardından özkulaklara ilişkin ağırlıklar elde edilir. Eğitim aşamasında detaylı olarak şu işlemler sırasıyla uygulanır:

1. Her kulak resminin $W\,x\,H = N$ pikselden oluştuğunu varsayalım. Toplamda eğitim kümesinde M adet resmin olduğu düşünülürse, eğitim kümesi;

$$\boldsymbol{E} = \{\boldsymbol{\Gamma}_1, \boldsymbol{\Gamma}_2, \dots, \boldsymbol{\Gamma}_M\} \tag{3.19}$$

olarak ifade edilebilir. Toplam M adet resmin ortalaması ise şu şekilde hesaplanır:

$$\boldsymbol{\psi} = \frac{1}{M}\sum_{i=1}^{M} \boldsymbol{\Gamma}_i \tag{3.20}$$

2. Daha sonra tüm orijinal kulak resimleri elde edilen ortalamadan çıkarılır. Ortaya yeni bir kulak kümesi çıkar.

$$\boldsymbol{\phi}_i = \boldsymbol{\Gamma}_i - \boldsymbol{\psi} \tag{3.21}$$

3. Şimdi kovaryans matrisi hesaplanır.

$$\boldsymbol{C} = \boldsymbol{A}\boldsymbol{A}^T \tag{3.22}$$

Buradaki $\boldsymbol{A}$ matrisi, $\boldsymbol{A} = \{\boldsymbol{\phi}_1, \boldsymbol{\phi}_2, \dots, \boldsymbol{\phi}_M\}$ olarak ifade edilebilir. Halihazırda $\boldsymbol{A}$ matrisi NxM; $\boldsymbol{A}^T$ matrisi de MxN boyutlu olduğundan kovaryans matrisimiz NxN boyutlu olacaktır. Her bir resimin boyutu dikkate alındığında elde edilen kovaryans matrisi çok büyük boyutta oluşur. Bir sonraki adımda amacımız, kovaryans matrisine ait özdeğer ve özvektörü hesaplamak olduğundan, kovaryans matrisimizin NxN'lik boyutu bu durumu oldukça zorlaştıracaktır. Çünkü toplamda N^2 adet özdeğer ve özvektör bulunması gerekir. Dolayısıyla daha uygulanabilir bir yönteme ihtiyaç duyulur.

Eğer örnek resim sayısı, oluşan resim uzayının boyutundan küçükse (ki genelde hep bu şekilde olur), elde edilebilecek sadece $M-1$ adet anlamlı özdeğer ve özvektör bulunur. Yani, standart kovaryans matrisinizde N^2 adet olan özdeğerden sadece $M-1$ tanesi anlamlıdır. Anlamlı teriminden maksat, özdeğerlerin sıfıra eşit olmamasıdır. Standart kovaryans matrisindeki bu sorun öncelikle, MxM boyutlu matrisin özdeğerleri bulunması sonrasında ise $\boldsymbol{\phi}_i$ ile ifade edilen ortalama çıkarılmış kulakların özvektör ağırlıklarıyla uygun bir doğrusal kombinasyonu yapılarak çözülür. Bu işlem için kullanılacak MxM boyutlu matris şu şekilde ifade edilir.

$$\boldsymbol{L = A^T A} \tag{3.23}$$

Yukarıdaki MxM boyutlu matris için özdeğer/özvektör koşulu yazılırsa,

$$\boldsymbol{Lv = \lambda v} \tag{3.24a}$$

$$\boldsymbol{A^T Av = \lambda v} \tag{3.24b}$$

Her iki taraf da soldan A matrisi ile çarpılırsa,

$$\boldsymbol{AA^T Av = A\lambda v} \tag{3.24c}$$

Burada λ skaler olduğunda yer değişebilir.

$$\boldsymbol{AA^T Av = \lambda Av} \tag{3.24d}$$

$$\boldsymbol{C = AA^T \xrightarrow{\substack{değişim\\yapılırsa}} Cv = \lambda Av} \tag{3.24e}$$

$$\boldsymbol{Av = u \xrightarrow{\substack{değişim\\yapılırsa}} Cu = \lambda u} \tag{3.24f}$$

Denklem (3.24b)'den de anlaşılacağı üzere $\boldsymbol{L}$ matrisine ait $\boldsymbol{v}_i$ özvektörleri hesaplandıktan sonra, bunların $\boldsymbol{\phi}_i$'ler ile doğrusal bir kombinasyonu bize denklem (3.24f)'deki sonuca götürür. Böylece standart kovaryans matrisine ait özvektörler ($\boldsymbol{u_i}$) hesaplanmış olur.

4. Bu şekilde elde edilen $M-1$ adet anlamlı özdeğerden herbirinin resim üzerindeki ağırlığı eşit olmayacaktır. İşte bu noktada TBA'nın boyut küçültme işlevi devreye sokulur. Herbir resim, kendisini yeterince iyi vaziyette temsil eden özdeğer/özvektörlerin kombinasyonu olarak ifade edilmelidir. Bu nedenle, M' adet ağırlığı yüksek özdeğer seçilir. Seçilecek özdeğer sayısı programcının inisiyatifine kalmakla birlikte, genelde 95%-99% arasındaki doğruluğa karşılık gelen sayı, herbir kulak resmini temsil etmede yeterli olacaktır.

$$95\% - 99\% = \frac{\lambda_1,\lambda_2,\ldots,\lambda_{M'}}{\lambda_1,\lambda_2,\ldots,\lambda_M} \tag{3.25}$$

5. Standart kovaryans matrisine ait u_i özvektörleri yardımıyla her $\boldsymbol{\Gamma}_i$ kulak resmi, kulak uzayına yansıtılır. Oluşan herbir vektör M' boyutunda olup, şu şekilde elde edilir:

$$\boldsymbol{\vartheta}_k = \boldsymbol{u}_i^T(\boldsymbol{\Gamma}_k - \boldsymbol{\psi})\ \{i = 1,2,\dots,M'\};\{k = 1,2,\dots,M\} \qquad \textbf{(3.26)}$$

6. Eğitim aşamasının en son adımı, test aşamasında sınıflandırmayı sağlayacak iki adet eşik değerini belirlemektir. Bu değerlerden biri test resminin uzaya olan uzaklığını (ε_u) karşılaştırırken; diğeri, resmin herbir sınıfa olan uzaklığını (ε_k) karşılaştırır. Eşik değerleri, gelen resmin kulak olup olmadığı eğer kulaksa hangi sınıfa ait olduğunu belirlemede kullanılacaktır. Bu nedenle de programcı tarafından dikkatle seçilmelidir. Eşik değerlerinin seçimi, sistemin doğrulaması yapılırken ayarlanabilecek bir büyüklüktür. Ancak en son belirlenen değerler programda kaydedilip, test aşamasında kullanılmalıdır.

Eğitim aşamasından sonra sisteme test için sokulan resmin, eğitim süresince elde edilen bilgilere göre karşılaştırılması gerekir. Genel manada test resmi için yapılanlar, eğitim aşamasında yapılanlara benzerdir. Yapılan işlemler detaylı olarak şu şekildedir.

1. Sisteme giren herhangi bir kulak resmi, eğitim aşamasında elde edilen özvektörler yardımıyla kulak uzayına yansıtılmalıdır.

$$\boldsymbol{\vartheta}_{test} = \boldsymbol{u}_i^T(\boldsymbol{\Gamma}_{test} - \boldsymbol{\psi})\ \ \{i = 1,2,\dots,M'\} \qquad \textbf{(3.27)}$$

2. Denklem (3.27)'a gore yansıtılan test resmi, herbir sınıfla karşılaştırılarak arasındaki farka bakılır. Bu işlem en kolay *Öklid uzaklığı* kullanılarak hesaplanır.

$$\varepsilon_k^2 = \|\boldsymbol{\vartheta}_{test} - \boldsymbol{\vartheta}_k\|^2\ \ \{k = 1,2,\dots,M\} \qquad \textbf{(3.28)}$$

3. Kulak resimleri haricinde, eğer sisteme başka bir resim verilirse, sistem test resminin kulak resmi olmadığını anlamalıdır. Bunun için de eğitim aşamasında kulak uzayından tekrar elde edilen resimler ile ortalama çıkarılmış test resmi arasındaki karesel uzaklığa bakılır.

$$\varepsilon^2 = \left\|\boldsymbol{\phi}_{test} - \boldsymbol{\phi}_f\right\|^2 \qquad \textbf{(3.29)}$$

Burada $\boldsymbol{\phi}_{test} = \boldsymbol{\Gamma}_{test} - \boldsymbol{\psi}$ ve $\boldsymbol{\phi}_f = \boldsymbol{u}_i^T(\boldsymbol{\Gamma}_k - \boldsymbol{\psi})\boldsymbol{u}_i$ olarak ifade edilir.

4. Elde edilen bu uzaklıklar sonucu bir karara varılır. Sonuçta 3 olasılıktan biri mümkün olacaktır.

a. $\varepsilon \geq \varepsilon_u$ olursa, test resmi kulak resmi değildir.

b. $\varepsilon < \varepsilon_u$ ve $\forall k$, $\varepsilon_k \geq \varepsilon_k$ olursa, test resmi bilinmeyen bir kişiye aittir.

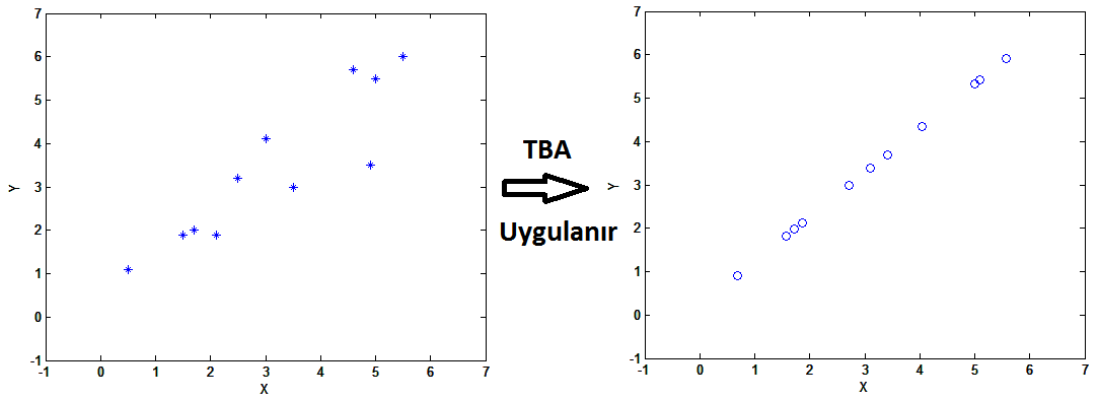
c. $\varepsilon < \varepsilon_u$ ve $\varepsilon_{k^*} = \underbrace{min}_{k}\{\varepsilon_k\} < \varepsilon_k$ olursa, test resmi i. kişiye aittir.

3.2 Fisher Doğrusal Ayırıcılar Yöntemi (FDAY) / Fisher's Linear Discriminant Analysis (FLDA)

FDAY, nitelik çıkarmak ve boyut azaltmada kullanılan ve bilinen en yaygın yöntemlerden biridir. FDAY'nın tarihi 1924 yılına kadar dayanır. İngiliz biyolog ve istatistikçi Sir Ronald Aylmer Fisher tarafından öne sürülmüştür. Teorinin özünde varyansların analizi yatmaktadır.

FDAY, istatistikte, model tanımada ve yapay öğrenme konularında kullanılan bir yöntemdir. Genel olarak, sınıflar arası saçılma oranını maksimize ederken sınıf içi saçılmayı minimize eder. Amaç, iki veya daha fazla sınıfı veya olayı, kendilerini karakterize eden niteliklerden doğrusal bir kombinasyon elde ederek ayırmaktır. Daha önce de bahsedildiği üzere, TBA, veriler arasında bir nitelik ve bileşen oluşturarak onları ifade etmeyi amaçlar. Ancak elde edilen bu nitelikler, farklı sınıflardan olan verilerin ayrılması için her zaman doğru sonucu vermez. Başka bir deyişle, TBA ile dikkate alınmayan bileşen, bazen FDAY için sınıfların ayrımında kullanılan bileşen olabilmektedir. Bu demektir ki, TBA'nın bulduğu bileşen hiçbir zaman sınıfların doğru bir şekilde ayrılacağını garanti edemez. Konuyla ilgili basit bir örnek vermek gerekirse, S ve Ş harfleri birbirlerinden ayrılmaya çalışılsın. TBA muhtemelen iki harfi de tanıyacak ancak aralarındaki farkı ayıramayacaktır. Çünkü Ş harfindeki uzantı, veriyi ifade ederken az ağırlığa sahiptir. Görüldüğü üzere, TBA ile yapılan analizde göz ardı edilen özellik, aslında iki sınıfı ayırmada kullanılması gereken yegane malzemedir. TBA elde bulunan verilerin en iyi gösterim yönünü bulmaya çalışırken, FDAY ise sınıflar için gerekli en iyi ayırım yönünü bulur. Şekil 3.15'te TBA uygulandığında verinin nasıl yansıtıldığı gösterilmektedir. Anımsatma amacıyla TBA bölümünde yapılan örneğe ait şekiller kullanılmıştır.

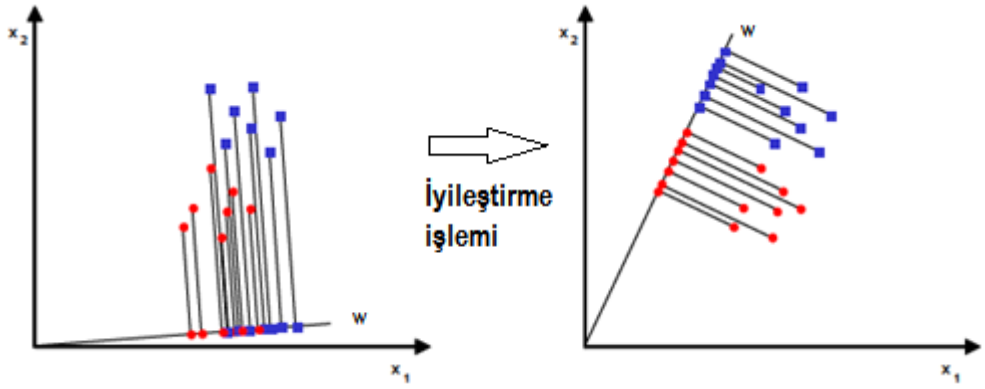
Bu bölümde, FDAY'nin teorisi irdelenmiştir. Ardından, nasıl uygulandığı örnek eşliğinde işlenmiştir. Daha sonra, FDAY'nin çok sınıflı bir veriye uygulanması teorik ve bir örnekle gösterilmiştir.



Şekil 3.15 : TBA uygulandıktan sonra verinin durumu.

3.2.1 FDAY'nın teorisi

FDAY'da ilgilendiğimiz problem aslında elimizde bulunan verinin d-boyuttan tek bir doğru üzerine yansıtmaktır. Eldeki veri aslında iyi bir şekilde ayrılmış olsa da, bir doğru üzerine yapılan yansıtma sonunda karışık bir durum oluşabilir. Yansıtılan örneklerin birbiri içine girmesi sonunda ise ayrım zorlaşır, dolayısıyla da tanıma doğruluğu düşer. Ancak yansıtılan doğrunun açısı ve yönü değiştirilmek kaydıyla, daha iyi bir ayrım elde edilebilir. Bu durum Şekil 3.16'da gösterilmiştir.



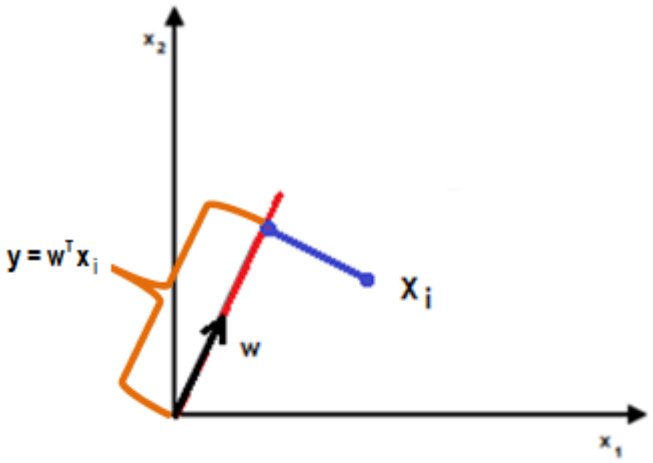
Şekil 3.16 : FDAY ile bir doğru üzerine yapılan yansıtmaya ait iki sonuç.

FDAY'nın amacı, Şekil 3.16'da sağ tarafta gösterilen şekilde bir ayrım yapmaktır. Çünkü, mavi ve kırmızı noktalar arasındaki ayırım bu doğru ile maksimize edilebilir.

$\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_n$ olmak üzere n tane d-boyutlu veri kümemiz olsun. Bunlardan n_1 tanesi w_1 etiketiyle D_1 kümesinde; n_2 tanesi w_2 etiketiyle D_2 kümesinde olsun. Eğer $\boldsymbol{x}$'in bileşenlerinden doğrusal bir kombinasyon yapmak istersek, skaler çarpım karşımıza çıkar.

$$y = \boldsymbol{w}^T \boldsymbol{x} \tag{3.30}$$

Elde edilen sonuç, $y_1, y_2, \ldots, y_n$ olmak üzere n tane ve Y_1 ve Y_2 olmak üzere iki kümeye ayrılmış olacaktır.



Şekil 3.17 : x_i noktasının w doğrultusundaki bir doğruya yansıtılması.

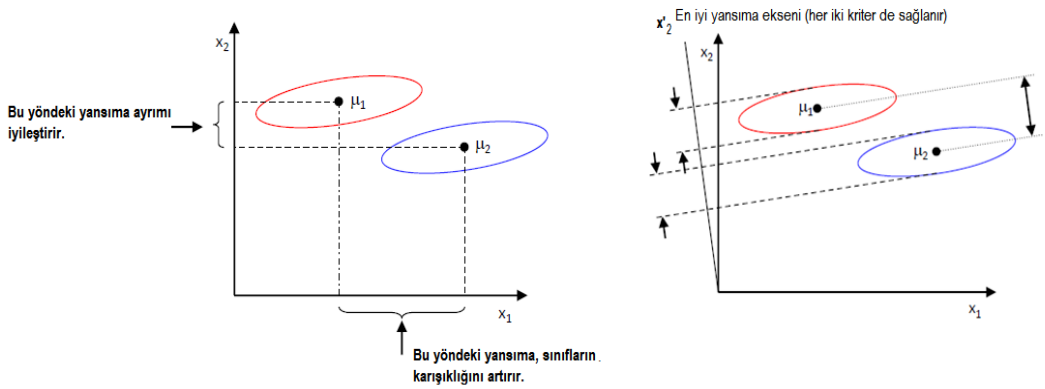
Geometrik olarak eğer, $\|\boldsymbol{w}\| = 1$ ise, her y_i kendisiyle ilişkili $\boldsymbol{x}_i$'nin $\boldsymbol{w}$ doğrultusundaki bir doğruya yansımasıdır. Yansıtma işleminde $\boldsymbol{w}$, y'nin ölçeğini ayarlar. Bu nedenle aslında, $\boldsymbol{w}$'nın büyüklüğü önemli değildir. Buna karşılık ise yönü önemlidir. Çünkü yansıtılan yön, ayrımın en iyi olduğu yön olmalıdır. Şekil 3.16 görsel olarak bu durumu özetler. Sınıfların orijinalde birbiri içerisine karışmış durumda olması, ayrımı bazı durumlarda mümkün kılmayabilir. Böyle durumlarda "en iyi" yön seçilse de amaca ulaşılamadığından FDAY kullanılamaz.

Normal durumlarda, en iyi $\boldsymbol{w}$ yönü iki faktöre bağlıdır. Bunlar;

1. Sınıfların ortalamaları birbirlerinde mümkün olduğunca uzakta,
2. Varyansları da 1. durum sağlandığında ve karışma olmadığında mümkün olduğunca büyük,

olmalıdır. Şekil 3.18 bu kriterleri dikkate alarak yapılmış en iyi yansımayı gösterir. $\boldsymbol{\mu}_i$, i. sınıfa ait d-boyutlu örneklem ortalaması vektörü olarak kabul edilirse,

$$\boldsymbol{\mu}_i = \frac{1}{n_i} \sum_{x \in D_i} \boldsymbol{x} \qquad \textbf{(3.31)}$$



Şekil 3.18 : FDAY'nın kriterlerine göre doğru yansıma.

olarak ifade edilir. Başlangıç verisine ait ortalama bu şekilde iken, yansıma sonundaki y noktalarına ait ortalama ise şöyle ifade edilir.

$$\widetilde{\boldsymbol{\mu}}_i = \frac{1}{n_i}\sum_{y\in Y_i} y \qquad \textbf{(3.32a)}$$

$$= \frac{1}{n_i}\sum_{\boldsymbol{x}\in D_i} \boldsymbol{w}^T\boldsymbol{x} = \boldsymbol{w}^T\boldsymbol{\mu}_i \qquad \textbf{(3.32b)}$$

Yukarıdaki denklem takımı aynı zamanda ortalamaların da yansıtıldığını gösterir. Daha sonra, yansıtılmış ortalamalar arasındaki uzaklığa bakılmak istenirse;

$$|\widetilde{\boldsymbol{\mu}_1} - \widetilde{\boldsymbol{\mu}_2}| = |\boldsymbol{w}^T(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)| \qquad \textbf{(3.33)}$$

olarak bulunur. FDAY'nın iki kriterine göre, ortalamalar arasındaki fark mümkün olduğunca büyük olurken sınıfların birbirlerine karışmaması zorunludur. Örneklerin varyanslarını oluşturmaktansa, yansıtılmış örneklerin saçılmaları göz önüne alınır. w_i etiketli yansıtılmış örneklerin saçılması şöyle ifade edilir.

$$\tilde{s}_i^2 = \sum_{y\in Y_i}(y - \widetilde{\boldsymbol{\mu}}_i)^2 \qquad \textbf{(3.34)}$$

Böylece, $1/n\left(\tilde{s}_1^2 + \tilde{s}_2^2\right)$ toplu olarak tüm verinin tahmini varyansı olarak ifade edilebilir. Bu arada $\left(\tilde{s}_1^2 + \tilde{s}_2^2\right)$'ye *toplam saçılma matrisi* denir. Bu sayede Fisher doğrusal ayırıcıları, kriter fonksiyonu olarak adlandırılan,

$$J(\boldsymbol{w}) = \frac{|\widetilde{\boldsymbol{\mu}_1} - \widetilde{\boldsymbol{\mu}_2}|^2}{(\tilde{s}_1^2 + \tilde{s}_2^2)} \qquad \textbf{(3.35)}$$

$J(\boldsymbol{w})$'yi maksimum yapmak için $\boldsymbol{w}^T\boldsymbol{x}$ dönüşümünü sağlarlar. $J(\boldsymbol{w})$ dikkat edilecek olursa $\|\boldsymbol{w}\|$'den bağımsızdır. $J(.)$ maksimize edilirken, aynı zamanda sınıflar arasındaki en iyi ayrım da sağlanmış olur.

$J(.)$'nın $\boldsymbol{w}$'nın açık bir fonksiyonu olarak ifade edilmek istenirse, öncelikle saçılma matrisleri tanımlanmalıdır. $\boldsymbol{S}_i$ ve $\boldsymbol{S}_w$ şu şekilde tanımlanır.

$$\boldsymbol{S}_i = \sum_{\boldsymbol{x}\in D_i}(\boldsymbol{x}-\boldsymbol{\mu}_i)(\boldsymbol{x}-\boldsymbol{\mu}_i)^T \tag{3.36}$$

ve

$$\boldsymbol{S}_w = \boldsymbol{S}_1 + \boldsymbol{S}_2 \tag{3.37}$$

Daha sonra ise yansıtılmış saçılma matrisleri şu şekilde ifade edilir:

$$\tilde{s}_i^2 = \sum_{\boldsymbol{x}\in D_i}(\boldsymbol{w}^T\boldsymbol{x}-\boldsymbol{w}^T\boldsymbol{\mu}_i)^2 \tag{3.38a}$$

$$= \sum_{\boldsymbol{x}\in D_i}\boldsymbol{w}^T(\boldsymbol{x}-\boldsymbol{\mu}_i)(\boldsymbol{x}-\boldsymbol{\mu}_i)^T\boldsymbol{w} \tag{3.38b}$$

$$= \boldsymbol{w}^T\boldsymbol{S}_i\boldsymbol{w} \tag{3.38c}$$

Dolayısıyla yansıtılmış saçılma matrislerinin toplamı da,

$$\tilde{s}_1^{\,2} + \tilde{s}_2^{\,2} = \boldsymbol{w}^T\boldsymbol{S}_w\boldsymbol{w} \tag{3.39}$$

olarak ifade edilebilir.

Benzer bir mantıkla, ortalamalar arasındaki ayrım durumu da şöyle ifade edilebilir:

$$(\tilde{\mu}_1 - \tilde{\mu}_2)^2 = (\boldsymbol{w}^T\boldsymbol{\mu}_1 - \boldsymbol{w}^T\boldsymbol{\mu}_2)^2 \tag{3.40a}$$

$$= \boldsymbol{w}^T\underbrace{(\boldsymbol{\mu}_1-\boldsymbol{\mu}_2)(\boldsymbol{\mu}_1-\boldsymbol{\mu}_2)^T}_{\boldsymbol{S}_B}\boldsymbol{w} \tag{3.40b}$$

$$= \boldsymbol{w}^T\boldsymbol{S}_B\boldsymbol{w} \tag{3.40c}$$

Denklem (3.40c)'deki $\boldsymbol{S}_B$ şu şekilde ifade edilir:

$$\boldsymbol{S}_B = (\boldsymbol{\mu}_1-\boldsymbol{\mu}_2)(\boldsymbol{\mu}_1-\boldsymbol{\mu}_2)^T \tag{3.41}$$

$\boldsymbol{S}_w$ matrisi, *sınıf içi saçılma matrisi* olarak adlandırılır. Sınıf içi saçılma matrisi, örneklere ait d-boyutlu verinin kovaryans matrisiyle orantılı bir büyüklüktür. $\boldsymbol{S}_w$ matrisi, simetrik matris olmakla birlikte genelde eğer örnek sayısı boyuttan büyükse tekil değildir.

Aynı şekilde, $\boldsymbol{S}_B$'de *sınıflar arası saçılma matrisi* olarak adlandırılır. $\boldsymbol{S}_B$ matrisi de simetrik bir matristir. Ancak iki vektörün dış çarpımı olduğundan rankı en fazla birdir. Özellikle herhangi bir $\boldsymbol{w}$ için, $\boldsymbol{S}_B\boldsymbol{w}$ çarpımı, $\boldsymbol{\mu}_1-\boldsymbol{\mu}_2$ yönünde olup tekildir.

$\boldsymbol{S}_B$ ve $\boldsymbol{S}_w$ büyüklüklerine bağlı olarak $J(.)$ Kriter fonksiyonu,

$$J(\boldsymbol{w}) = \frac{\boldsymbol{w}^T \boldsymbol{S}_B \boldsymbol{w}}{\boldsymbol{w}^T \boldsymbol{S}_w \boldsymbol{w}} \quad \textbf{(3.42)}$$

olarak ifade edilir. Tekrar FDAY'ya dönersek, $J(.)$'yi maksimum yapacak $\boldsymbol{w}$ vektörü,

$$\boldsymbol{S}_B \boldsymbol{w} = \boldsymbol{\lambda} \boldsymbol{S}_w \boldsymbol{w} \quad \textbf{(3.43)}$$

sağlamalıdır. Bu eşitlikte $\boldsymbol{\lambda}$ skaler bir değerdir. Denklem (3.43) ise standart özdeğer/özvektör problemi olarak adlandırılır. Bu durum aynı zamanda, $\boldsymbol{w}$'de oluşabilecek herhangi bir değişmenin $J(\boldsymbol{w})$'nin ekstremum noktalarında bir değişime neden olmamasından da anlaşılabilir. Çünkü $J(\boldsymbol{w})$ fonksiyonun oranında değişme olmayacaktır. Elde edilen tüm veriler sonunda, eğer $\boldsymbol{S}_w$ matrisimiz tekil değilse, özdeğer problemimiz şu şekilde ifade edilebilir.

$$\boldsymbol{S}_w^{-1} \boldsymbol{S}_B \boldsymbol{w} = \boldsymbol{\lambda} \boldsymbol{w} \quad \textbf{(3.44)}$$

$\boldsymbol{S}_B \boldsymbol{w}$ çarpımı, $\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2$ yönünde olduğundan yukarıda ifade edilen özdeğer/özvektör problemini çözmenin bir anlamı kalmaz. Çünkü $\boldsymbol{w}$'nin büyüklük faktörü bu durumda önemsizdir. Dolayısıyla $J(.)$'yı optimize eden problemin sonucu doğrudan şu şekilde yazılabilir.

$$\boldsymbol{w} = \boldsymbol{S}_w^{-1}(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2) \quad \textbf{(3.45)}$$

Böylece, Fisher doğrusal ayırıcıları için gereken $\boldsymbol{w}$ vektörü bulunur. Elde edilen vektör, sınıflar arası saçılma matrisinin sınıf içi saçılma matrsine oranını maksimum yapar. Böylece, sınıflandırma problemi d-boyuttan daha yönetilebilir, hesaplaması daha kolay olan tek boyuta indirgenir. Bu dönüşüm, çok boyuttan tek boyuta dönüşüm olarak bilinir ve teoride, eğer çok büyük bir eğitim kümesi mevcutsa, hata oranını küçültmede başarılı değildir. Genelde tek boyutta çalışmanın avantajlarından dolayı, teoride ulaşılabilen yüksek performanstan fedakârlık etmek gerekir. Bundan sonraki işlemler, yansıtılmış noktaların tek boyutta ayrıştırılması için gereken eşik değerlerini bulmak için yapılır.

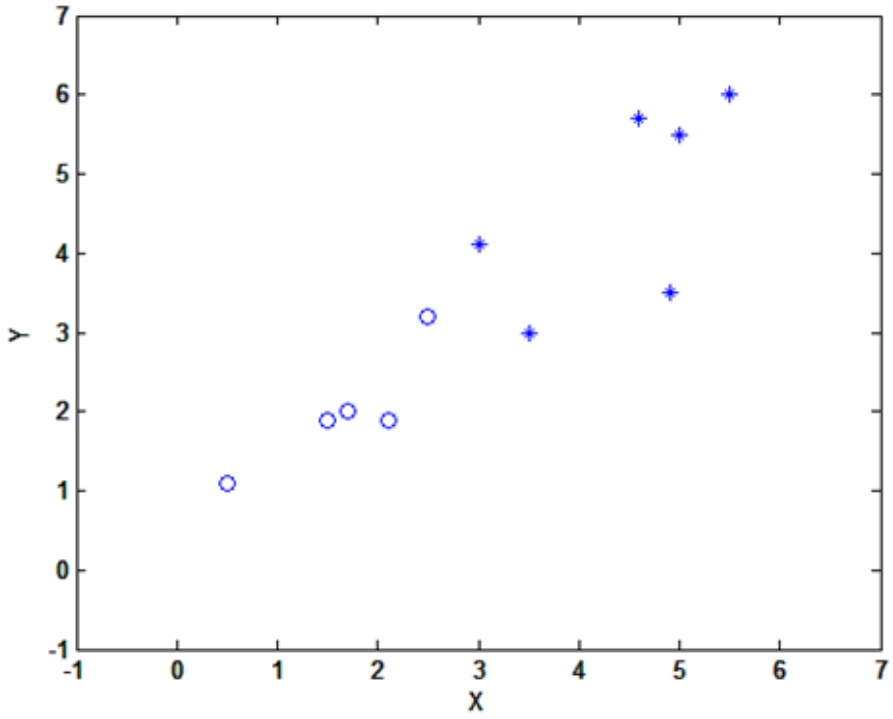
3.2.2 FDAY'nin uygulanması

Bu bölümde, teorisi anlatılan FDAY'nın nasıl uygulandığı anlatılacaktır. Uygulama için TBA bölümünde kullanılan veri noktaları kullanılmıştır. Bu şekilde TBA ve FDAY arasında sınıflandırma teorilerini ve elde edilen sonuçları daha rahat gözlemleyebiliriz. FDAY için uygulanan adımlar sırasıyla şu şekildedir.

TBA ile işlenen örnek, FDAY için iki sınıfa ayrılmıştır. Çünkü amaç, sınıfları birbirinden doğrusal bir şekilde ayırmak olacaktır. Kullanılan veri noktaları sınıflarıyla birlikte Çizelge 3.7'de, mütekiben bu veri noktalarının iki sınıf halinde MATLAB ortamında çizilmiş şekli ise Şekil 3.19'da gösterilmiştir.

Çizelge 3.7 : FDAY için kullanılan veri noktaları kümesi.

Sınıflar	Noktalar	X	Y
1. Sınıf	P_2	0.5	1.1
	P_4	1.5	1.9
	P_7	1.7	2
	P_8	2.1	1.9
	P_{10}	2.5	3.2
2. Sınıf	P_1	3.5	3
	P_3	3	4.1
	P_5	4.9	3.5
	P_6	5	5.5
	P_9	4.6	5.7
	P_{11}	5.5	6



Şekil 3.19 : FDAY için plotlanan orijinal veri noktaları.

1. Daha sonra her sınıfa ait ortalamanın hesaplanması gerekir. Denklem (3.31)'e istinaden hesaplanan sınıflar için ortalama değerleri ise, $\mu_1 = [1.66 \quad 2.02]$ ve $\mu_2 = [4.4167 \quad 4.6333]$ olarak hesaplanmıştır.

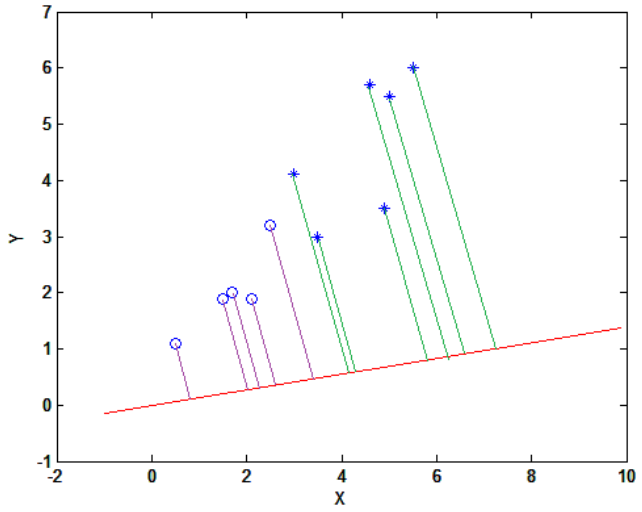
2. Ardından her sınıfa ait saçılma matrisi, bir başka deyişle kovaryans matrisi Denklem (3.36)'a göre hesaplanır. MATLAB'de yapılan hesaplama sonunda, $S_1 = \begin{bmatrix} 0.568 & 0.506 \\ 0.506 & 0.567 \end{bmatrix}$ ve $S_2 = \begin{bmatrix} 0.9257 & 0.7773 \\ 0.7773 & 1.5987 \end{bmatrix}$ olarak hesaplanmıştır.

3. Her sınıfın kendine ait saçılma matrisi hesaplandıktan sonra, sınıf içi saçılma matrisi Denklem (3.37)'a göre hesaplanır. Yapılan işlem sonunda $S_W = \begin{bmatrix} 1.4937 & 1.2833 \\ 1.2833 & 2.1657 \end{bmatrix}$ olarak bulunmuştur.

4. Sınıflar arası saçılma matrisi, Denklem (3.41) baz alınarak hesaplanır. Yapılan hesaplama sonunda $S_B = \begin{bmatrix} 7.5992 & 7.2041 \\ 7.2041 & 6.8295 \end{bmatrix}$ olarak hesaplanmıştır.

5. Sınıflar arası ve sınıf içi saçılma matrislerini hesapladıktan sonra, Denklem (3.43)'da olduğu gibi standart özdeğer/özvektör problemi karşımıza çıkar. Bu eşitlik çözülür ve (sınıf sayısı-1) tane en değerli özvektör seçilir. Örneğimizde 2 adet sınıfımız olduğundan, büyük değerli özdeğerle ilişkili özvektör yansıma yönü olarak seçilir. Seçilen özvektör : $w = \begin{bmatrix} 0.9904 \\ 0.1384 \end{bmatrix}$ 'dir.

6. Buna göre MATLAB ortamında çizilmiş veri ve en önemli özvektöre göre elde edilmiş yansıma ekseni Şekil 3.20'de olduğu gibidir.



Şekil 3.20 : Veri noktaları ve en büyük özdeğerle ilişkili en uygun yansıma ekseni.

Şekil 3.20 ve Şekil 3.3 incelenirse, TBA ve FDAY arasındaki temel fark ortaya çıkar. Daha önce de belirtildiği gibi, TBA tüm veri kümesini ele alır ve maksimum varyansın oluşacağı doğruyu bulmaya çalışır. Buna karşılık FDAY ise, her iki sınıfı ele alır ve sınıfların ortalamaları birbirlerinden olabildiğince uzakta olacağı ve sınıfların kendi içindeki varyanslarının da olabildiğince maksimum olacağı bir yansıma doğrusu bulunur.

3.2.3 Çoklu FDAY

Bir önceki bölümde, FDAY'nın uygulaması, gösterim ve anlatımı daha anlaşılır olması maksadıyla iki sınıf olarak yapılmıştır. Bu bölümde ele alınacak "çoklu FDAY", ikiden fazla sınıf için ele alınacaktır.

Başlangıçta, k-adet sınıf olduğu farzedilirse, FDAY için genel olan bir varsayımla "$k-1$ adet ayırıcı fonksiyon bulunur" denilebilir. Bu nedenle, FDAY sırasında yapılan yansıtma, d-boyutlu bir uzaydan, daha küçük olan $k-1$ boyutlu uzaya olacaktır ($d \geq k$ olduğu kabul edilmiştir). Sınıf içi saçılma matrisinin k-adet sınıf için genelleştirilmiş hali şu şekildedir.

$$\boldsymbol{S}_w = \sum_{i=1}^{k} \boldsymbol{S}_i \tag{3.46}$$

Denklem (3.46)'da bulunan $\boldsymbol{S}_i$ daha önceden de ifade edildiği gibi,

$$\boldsymbol{S}_i = \sum_{\boldsymbol{x} \in D_i} (\boldsymbol{x} - \boldsymbol{\mu}_i)(\boldsymbol{x} - \boldsymbol{\mu}_i)^T \tag{3.47}$$

ve

$$\boldsymbol{\mu}_i = \frac{1}{n_i} \sum_{x \in D_i} \boldsymbol{x} \tag{3.48}$$

Görüldüğü gibi, ortalama vektörü ve sınıf içi matrisinin k sınıf için genelleştirilmesi bu şekilde iken, sınıflar arası saçılma matrisi $\boldsymbol{S}_B$'nin genelleştirilmesi bu kadar açık şekilde olamamaktadır. $\boldsymbol{S}_B$'nin genelleştirilmesi için toplam ortalama vektörü ($\boldsymbol{\mu}$) ve toplam saçılma matrisi tanımlanmalıdır. Dolayısıyla, toplam ortalama vektörü ve toplam saçılma matrisi sırasıyla şu şekilde tanımlanır.

$$\boldsymbol{\mu}_T = \frac{1}{n} \sum_{\boldsymbol{x}} \boldsymbol{x} = \frac{1}{n} \sum_{i=1}^{k} n_i \boldsymbol{\mu}_i \tag{3.49}$$

ve

$$\boldsymbol{S}_T = \sum_{\boldsymbol{x}} (\boldsymbol{x} - \boldsymbol{\mu}_T)(\boldsymbol{x} - \boldsymbol{\mu}_T)^T \tag{3.50}$$

Denklem (3.50)'de her terime sınıf ortalamaları eklenir ve çıkarılırsa,

$$\boldsymbol{S}_T = \sum_{i=1}^{k} \sum_{x \in D_i} (\boldsymbol{x} - \boldsymbol{\mu}_i + \boldsymbol{\mu}_i - \boldsymbol{\mu}_T)(\boldsymbol{x} - \boldsymbol{\mu}_i + \boldsymbol{\mu}_i - \boldsymbol{\mu}_T)^T \quad \textbf{(3.51a)}$$

$$= \boldsymbol{S}_W + \underbrace{\sum_{i=1}^{k} n_i \, (\boldsymbol{\mu}_i - \boldsymbol{\mu}_T)(\boldsymbol{\mu}_i - \boldsymbol{\mu}_T)^T}_{\boldsymbol{S}_B} \quad \textbf{(3.51b)}$$

elde edilir. Denklem (3.51b)'de elde edilen sonuçta ikinci terim, sınıflar arası saçılma matrisinin genelleştirilmiş hali olarak ifade edilebilir.

$$\boldsymbol{S}_B = \sum_{i=1}^{k} n_i \, (\boldsymbol{\mu}_i - \boldsymbol{\mu}_T)(\boldsymbol{\mu}_i - \boldsymbol{\mu}_T)^T \quad \textbf{(3.52)}$$

Sırasıyla Denklem (3.51) ve Denklem (3.52) incelenirse, toplam saçılma matrisi, sınıf içi saçılma matrisi ve sınıflar arası saçılma matrisinin toplamı olarak ifade edilebilir.

$$\boldsymbol{S}_T = \boldsymbol{S}_w + \boldsymbol{S}_B \quad \textbf{(3.53)}$$

Gerekli olan tüm bilgiler elde edildikten sonra, örneklerin yansıtılması yapılır. d-boyutlu bir uzaydan $k-1$-boyutlu bir uzaya yansıtma işlemi $k-1$ adet ayırıcı fonksiyon ile gerçekleştirilir.

$$y_i = \boldsymbol{w}_i^T \boldsymbol{x} \quad , \; i = 1,2, \dots, k-1 \quad \textbf{(3.54)}$$

y_i'leri $\boldsymbol{y}$ vektörünün birer bileşeni ve $\boldsymbol{w}_i$'leri de d satır $k-1$ sütunlu bir $\boldsymbol{W}$ matrisin her bir sütunu olarak düşünürse, Denklem (3.54), tek bir matris işlemi olarak şu şekilde ifade edilebilir.

$$\boldsymbol{y} = \boldsymbol{W}^T \boldsymbol{x} \quad \textbf{(3.55)}$$

İki sınıflı durumda olduğu gibi, $\boldsymbol{x}_1, \dots, \boldsymbol{x}_n$ olarak elde bulunan örnek noktaları, bunlarla ilişkili $\boldsymbol{y}_1, \dots, \boldsymbol{y}_n$ noktalarına yansıtıldığında, elde edilen sonuçlar kendilerine ait ortalama ve saçılma matrisleriyle ifade edilebilirler.

$$\widetilde{\boldsymbol{\mu}_\iota} = \frac{1}{n_i} \sum_{y \epsilon y_i} \boldsymbol{y} \quad \textbf{(3.56)}$$

$$\widetilde{\boldsymbol{\mu}_T} = \frac{1}{n} \sum_{i=1}^{k} n_i \widetilde{\boldsymbol{\mu}_\iota} \quad \textbf{(3.57)}$$

$$\widetilde{\boldsymbol{S}_w} = \sum_{i=1}^{k} \sum_{y \in y_i} (\boldsymbol{y} - \widetilde{\boldsymbol{\mu}_\iota})(\boldsymbol{y} - \widetilde{\boldsymbol{\mu}_\iota})^T \quad \textbf{(3.58)}$$

$$\widetilde{\boldsymbol{S}_B} = \sum_{i=1}^{k} n_i \, (\widetilde{\boldsymbol{\mu}_\iota} - \widetilde{\boldsymbol{\mu}_T})(\widetilde{\boldsymbol{\mu}_\iota} - \widetilde{\boldsymbol{\mu}_T})^T \quad \textbf{(3.59)}$$

Yukarıda gösterilen formülasyon sonunda,

$$\widetilde{\boldsymbol{S}_w} = \boldsymbol{W}^T \boldsymbol{S}_w \boldsymbol{W} \quad \textbf{(3.60)}$$

ve

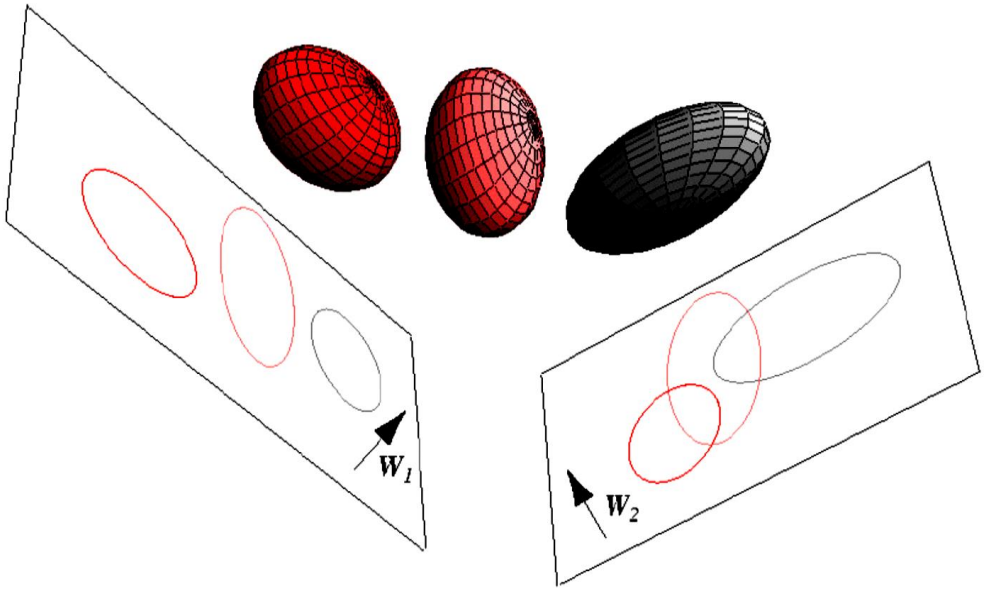
$$\widetilde{S_B} = \boldsymbol{W}^T \boldsymbol{S}_B \boldsymbol{W} \tag{3.61}$$

elde edilir.

Denklem (3.60) ve Denklem (3.61), yansıma ile saçılma matrislerinin de nasıl dönüştürüldüğünü gösterir. FDAY'daki asıl amacı bir kez daha hatırlarsak, öyle bir $\boldsymbol{W}$ matrisi bulunmalı ki, sınıflar arası saçılma matrisinin sınıf içi saçılma matrisine oranı maksimum olsun. Basit olarak göstermek gerekirse Şekil 3.21, 3 sınıfın yansıtılmasını gösterir.

Şekil 3.21'deki örnekte 3 farklı sınıf 2 adet düzleme yansıtılmıştır ($k-1$ ayırıcı prensibi). Burada yansıtılan düzlemler $\boldsymbol{W}_1$ ve $\boldsymbol{W}_2$ normal vektörleriyle ifade edilir. 2'den fazla sınıf olduğunda, FDAY toplam sınıf içi saçılmalara isitnaden en uygun ayırıcı düzlemi bulmaya çalışır. Bu örnekte $\boldsymbol{W}_1$ ile ifade edilen düzlem bunu en iyi sağlayandır.

Denklem (3.60) ve Denklem (3.61) ile elde edilen saçılma matrislerinin basit ölçümü, determinant hesabıyla yapılır. Determinant hesabı, özdeğerlerin çarpımı ve



Şekil 3.21 : 3 farklı sınıfın 2 adet düzleme yansıtılması.

dolayısıyla da temel yönlerdeki varyansların çarpımıdır. Bu hesabı kullanarak, FDAY için kriter fonksiyonu şu şekilde ifade edilir.

$$J(\boldsymbol{W}) = \frac{|\widetilde{S_B}|}{|\widetilde{S_w}|} = \frac{|\boldsymbol{W}^T \boldsymbol{S}_B \boldsymbol{W}|}{|\boldsymbol{W}^T \boldsymbol{S}_w \boldsymbol{W}|} \tag{3.62}$$

$J(.)$'yi maksimum yapan $\boldsymbol{W}$ matrisinin hesabı bu koşullarda zahmetli ve biraz da matematiksel zorluğu olan işlemlerin çözümünü gerektirir. Ancak sonuçta elde edilen matrisin her bir sütunu, genelleştirilmiş bir özdeğer/özvektör problemi sonucu oluşan özvektörlerdir. Oluşan özdeğer/özvektör problemi şu şekilde ifade edilebilir.

$$\boldsymbol{S}_B\boldsymbol{w}_i = \boldsymbol{S}_w\boldsymbol{\lambda}_i\boldsymbol{w}_i \tag{3.63}$$

$\boldsymbol{S}_w$, sol tarafa geçirilirse denklem şu şekli alır.

$$\boldsymbol{S}_w{}^{-1}\boldsymbol{S}_B\boldsymbol{w}_i = \boldsymbol{\lambda}_i\boldsymbol{w}_i \tag{3.64}$$

Denklem (3.64)'dan da anlaşılacağı üzere, FDAY'nın sonuca ulaşması için sınıf içi saçılma matrisinin tersi alınmalıdır. FDAY'da ise en zor kısımlardan biri de bu işlemdir. Çünkü $\boldsymbol{S}_w$ matrisi tekil olma riskindedir. Eğer $\boldsymbol{S}_w$, tekil değilse, söz konusu özdeğer/özvektör problemi Denklem (3.64) çözülerek bulunur (Burada ayrıca şu da akıldan çıkarılmamalıdır ki $\boldsymbol{S}_w{}^{-1}\boldsymbol{S}_B$ simetrik değildir. Bu özellik daha sonra gömülü sistem tasarımında çözüm için kullanılacaktır).

$\boldsymbol{S}_w$'nin tekil olmaması durumunda, Denklem (3.63)'ün daha basit olarak,

$$|\boldsymbol{S}_B - \boldsymbol{\lambda}_i\boldsymbol{S}_w| = 0 \tag{3.65}$$

ile ifade edilen karaktristik polinomunun öncelikle kökleri hesaplanıp daha sonra da

$$(\boldsymbol{S}_B - \boldsymbol{\lambda}_i\boldsymbol{S}_w)\boldsymbol{w}_i = 0 \tag{3.66}$$

Denklem (3.66)'deki $\boldsymbol{w}_i$ ile ifade edilen özvektörler için doğrudan çözümü yapılarak hesaplanabilir. Yapılan çözümün sonunda görülür ki, $\boldsymbol{S}_B$'nin rankı en fazla $k-1$ veya daha az olacaktır. Çünkü $\boldsymbol{S}_B$, k adet 1 veya daha az ranklı matrislerin toplamıdır ve bunlardan sadece $k-1$ tanesi bağımsız olacaktır. Bu yüzden, elde edilen özdeğerlerden sadece $k-1$ tanesi sıfırdan farklıdır ve dolayısıyla $k-1$ tane bunlarla ilişkili anlamlı özvektörler mevcuttur. Eğer sınıf içi saçılma matrisi izotropik ise, elde edilen özvektörler sadece $\boldsymbol{S}_B$'nin özvektörleridir ve elde edilen sıfırdan farklı özdeğerlerle ilişkili özvektörler, $\boldsymbol{\mu}_i - \boldsymbol{\mu}_T$ vektörlerinin oluşturduğu uzayı oluşturur. Bu durumda, $\boldsymbol{W}$ matrisinin her bir sütunu, $k-1$ adet $\boldsymbol{\mu}_i - \boldsymbol{\mu}_T$ $(i = 1, \dots, k-1)$ vektörüne Gram-Schmidt ortonormalizasyon işlemi uygulanarak bulunabilir. Bu durumda, genelde $\boldsymbol{W}$'nin çözümü tek değildir. Buna rağmen kriter fonksiyonu $J(.)$ bir oranı temsil ettiğinden, elde edilen sonuç ne olursa olsun yapılan yansıma ve sınıflandırma değişmez.

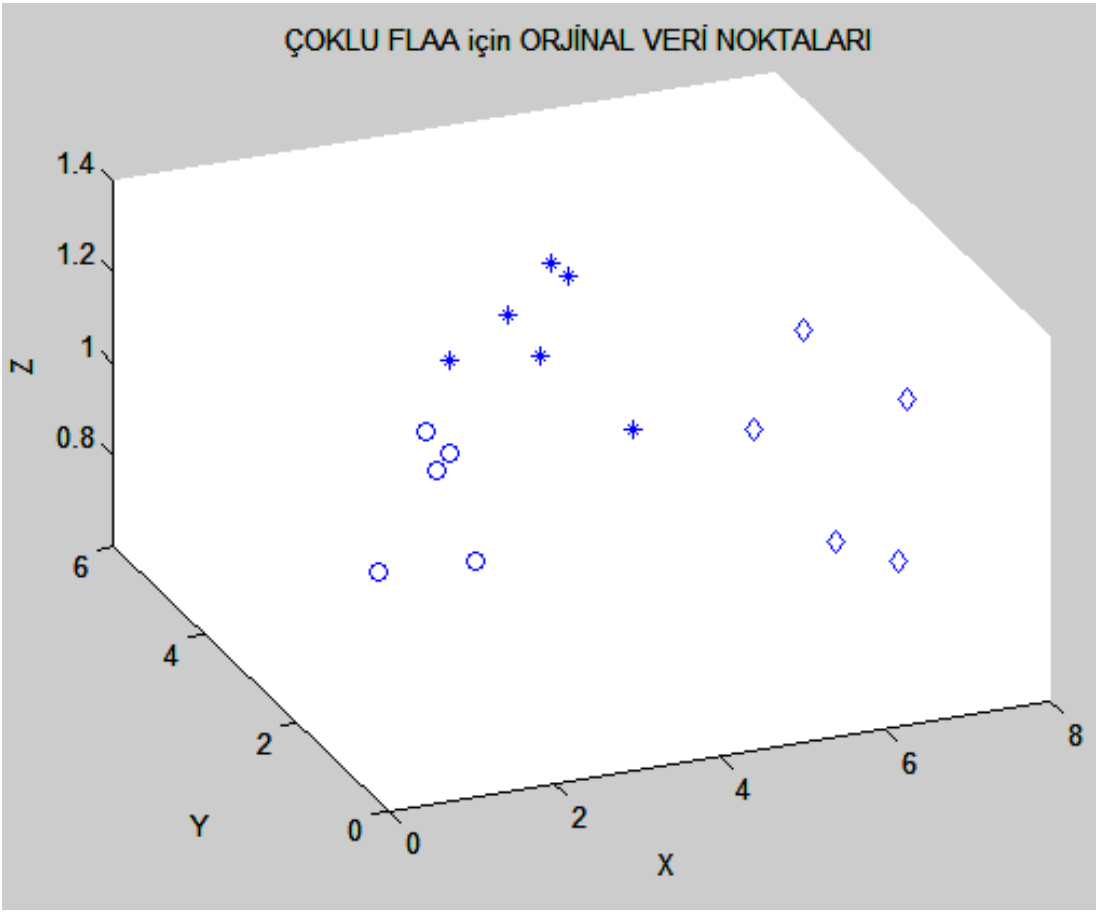
3.2.4 Çoklu FDAY'nin uygulanması

Bu bölümde, teorisi anlatılan çoklu FDAY'nın nasıl uygulandığı anlatılacaktır. Yapılan uygulamada kullanılan veri noktaları Çizelge 3.8'de gösterilmiştir ve uygulanan adımlar ise sırasıyla şöyledir.

1. Çoklu FDAY'daki amaç, elde olan veri noktaları ile oluşturulmuş 3 boyutlu sınıfların en doğru şekilde ayrıştırılmasıdır. Kullanılan veri noktaları sınıflarıyla birlikte Çizelge 3.8'dedir. Müteakiben, Çizelge 3.8'de verilen 3 boyutlu veri noktalarının MATLAB ortamında çizilmiş hali, Şekil 3.22'de gösterilmiştir.

Çizelge 3.8 : Çoklu FDAY için kullanılan veri noktaları kümesi.

Sınıflar	Noktalar	X	Y	Z
1. Sınıf	P_1	0.5	1.1	1
	P_2	1.5	1.9	1.2
	P_3	1.7	2	1.1
	P_4	2.1	1.9	0.9
	P_5	2.5	3.2	1
2. Sınıf	P_6	3.5	3	1.2
	P_7	3	4.1	1.1
	P_8	4.9	3.5	0.95
	P_9	5	5.5	1.12
	P_{10}	4.6	5.7	1
	P_{11}	5.5	6	1.03
3. Sınıf	P_{12}	6	1.8	1.3
	P_{13}	6.5	2	0.8
	P_{14}	7.8	2.8	1
	P_{15}	7.2	1.9	0.75
	P_{16}	5.9	2.7	1



Şekil 3.22 : Çoklu FDAY için 3 boyutlu plotlanan orijinal veri noktaları.

2. Daha sonra her sınıfa ait ortalamanın hesaplanması gerekir. Denklem (3.48)'e istinaden sınıflar için hesaplanan ortalama değerleri ise, $\boldsymbol{\mu}_1 = [1.66 \quad 2.02 \quad 1.04]$, $\boldsymbol{\mu}_2 = [4.4167 \quad 4.6333 \quad 1.0667]$ ve $\boldsymbol{\mu}_3 = [6.68 \quad 2.24 \quad 0.97]$ olarak hesaplanmıştır. Ardından her sınıfa ait saçılma matrisi, bir başka deyişle kovaryans matrisi Denklem (3.47)'ye göre hesaplanır. MATLAB'de yapılan hesaplama sonunda,

$$\boldsymbol{S}_1 = \begin{bmatrix} 0.568 & 0.506 & -0.018 \\ 0.506 & 0.567 & -0.0035 \\ -0.018 & -0.0035 & 0.013 \end{bmatrix}, \quad \boldsymbol{S}_2 = \begin{bmatrix} 0.9257 & 0.7773 & -0.0493 \\ 0.7773 & 1.5987 & -0.0357 \\ -0.0493 & -0.0357 & 0.0082 \end{bmatrix} \quad \text{ve}$$

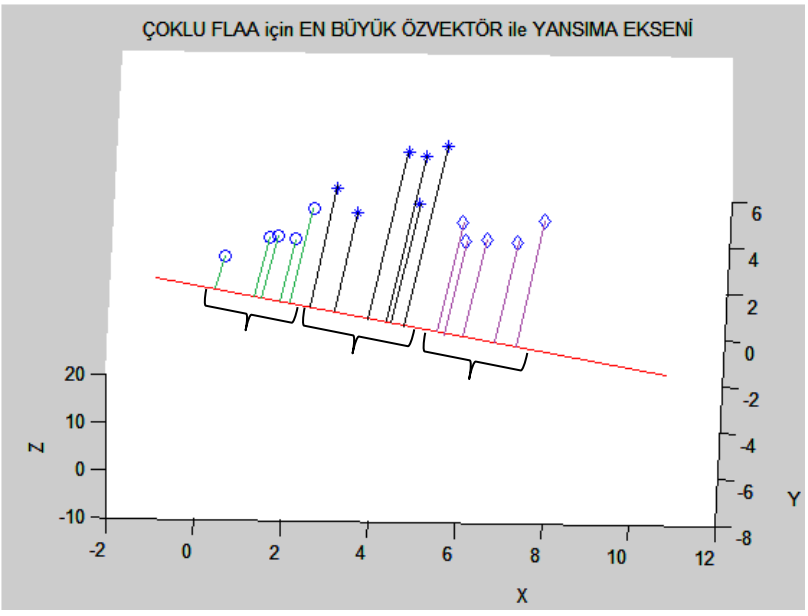
$$\boldsymbol{S}_3 = \begin{bmatrix} 0.657 & 0.1085 & -0.0745 \\ 0.1085 & 0.223 & 2.5\times10^{-4} \\ -0.0745 & 2.5\times10^{-4} & 0.047 \end{bmatrix} \text{ olarak elde edilmiştir.}$$

3. Her sınıfın kendine ait saçılma matrisi hesaplandıktan sonra, sınıf içi saçılma matrisi Denklem (3.46)'e göre hesaplanır. Yapılan işlem sonunda $\boldsymbol{S}_W = \begin{bmatrix} 2.1507 & 1.3918 & -0.1418 \\ 1.3918 & 2.3887 & -0.0389 \\ -0.1418 & -0.0389 & 0.0682 \end{bmatrix}$ olarak bulunmuştur.

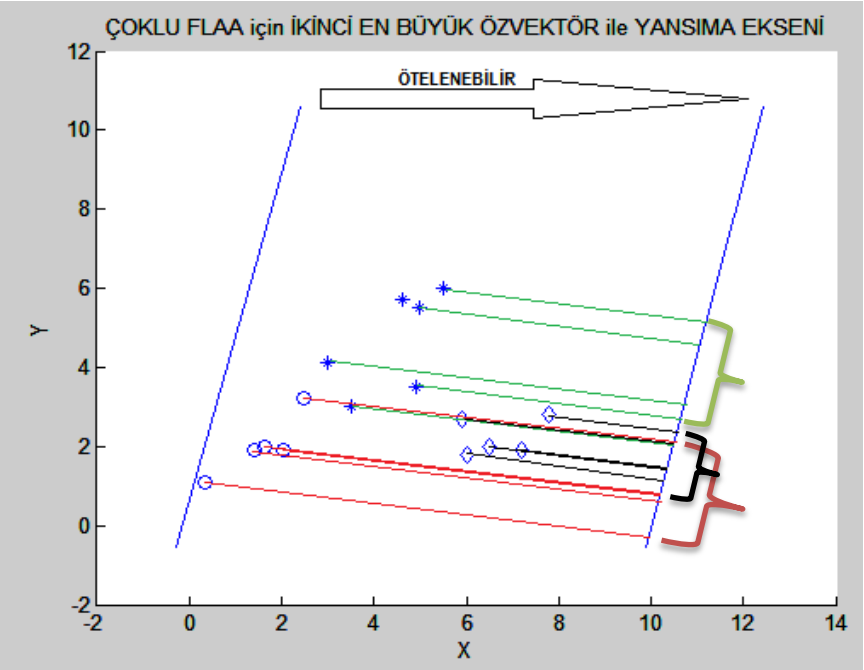
4. Sınıflar arası saçılma matrisi ise, Denklem (3.52) baz alınarak hesaplanır. Yapılan hesaplama sonunda $\boldsymbol{S}_B = \begin{bmatrix} 63.2292 & 5.0766 & -0.8215 \\ 5.0766 & 23.621 & 0.5404 \\ -0.8215 & 0.5404 & 0.0265 \end{bmatrix}$ olarak hesaplanmıştır.

5. Sınıflar arası ve sınıf içi saçılma matrislerini hesapladıktan sonra, Denklem (3.68)'da olduğu gibi standart özdeğer/özvektör problemi karşımıza çıkar. Bu eşitlik çözülür ve (sınıf sayısı-1) tane en değerli özvektör seçilir. Örneğimizde 3 adet sınıfımız olduğundan, büyük değerli özdeğerlerle ilişkili özvektörler sırasıyla yansıma yönü olarak seçilir. Seçilen özvektörler : $\boldsymbol{w}_1 = \begin{bmatrix} 0.5415 \\ -0.3402 \\ 0.7688 \end{bmatrix}$ ve $\boldsymbol{w}_2 = \begin{bmatrix} -0.0969 \\ -0.5287 \\ 0.8432 \end{bmatrix}$ 'dir.

6. Buna göre MATLAB ortamında çizilmiş olan veri ve en önemli özvektörlere göre elde edilmiş yansıma eksenleri sırasıyla Şekil 3.23 ve Şekil 3.24'te olduğu gibidir.



Şekil 3.23 : Veri noktaları ve en büyük özdeğerle ilişkili yansıma ekseni.



Şekil 3.24 : Veri noktaları ve ikinci en büyük özdeğerle ilişkili yansıma ekseni.

Şekil 3.23 ve Şekil 3.24 incelenirse, en büyük özdeğere ilişkin olan özdeğer kullanılarak yapılan yansıtma işlemi tüm sınıfların doğru şekilde ayrılmasını sağlamaktadır. Ancak ikinci en büyük özdeğere ilişkin özvektörle yapılan yansımada sınıflar nispeten birbirlerinin içine girmektedir. Ancak yine de en düşük özdeğerle ilişkili özvektöre göre ayırıcılığı daha fazladır.

4. KULAK TANIMA UYGULAMALARINDA ORTAYA ÇIKAN PROBLEMLERİN ÇÖZÜM YÖNTEMLERİ

Kulak biyometrisi ile tanımada, yüz tanımada olduğu gibi uygulama alanlarında birçok nümerik problem ortaya çıkmaktadır. Dolayısıyla bu bölümde ortaya çıkan problemler ve bunların çözüm yöntemleri irdelenmiştir.

Bölüm 3'te anlatılan *"Kulak Tanıma Algoritmaları"* uygulama alanlarında bazı zorlukları da beraberinde getirir. Oluşan bu nümerik zorlukların bilgisayar ortamındaki uygulamalarda çözümü kolay olmakla birlikte; bilgisayardan bağımsız bir uygulama alanında ise bir hayli zahmetli olmaktadır. Bu zorluklar genel olarak şu şekilde ifade edilebilir:

1. TBA uygulamasında oluşan kovaryans matrisinin özdeğer/özvektörlerinin bulunması,
2. FDAY uygulamasında sınıf içi saçılma matrisinin ($\boldsymbol{S}_w$) tersinin alınması,
3. Tersi alınan sınıf içi saçılma matrisinden sonra Denklem (3.64) ile elde edilen $\boldsymbol{S}_w{}^{-1}\boldsymbol{S}_B$ çarpım sonunda, bu sistemin özdeğer/özvektörlerinin bulunması.

Bilgisayardan bağımsız bir uygulamada ortaya çıkan nümerik problemlere ilişkin uygulanan çözüm yöntem ve algoritmalar ise şöyledir.

4.1 Jacobi İterasyonu ile Özdeğer/Özvektör Bulunması

Jacobi iterasyonu, özdeğer/özvektörlerin bulunması için anlaşılması en kolay algoritmalardandır. Jacobi iterasyonunun uygulanması için özdeğer/özvektörleri bulunacak sistemin gerçek ve simetrik olması gerekir. Bu koşulları sağlayan sistem de daha önceden bahsedildiği üzere TBA algoritmasında ortaya çıkar. Elde edilen kovaryans matrisi hatırlanacak olursa simetrik bir matristir. Her ne kadar yavaş işlese de Jacobi iterasyonu gerçek simetrik matrislere ait özdeğer/özvektör çiftlerini hesaplamada çok başarılıdır.

Bu algoritma uygulanırken iki varsayım kabullenilerek işlem yapılır.

1. Sistemin tek bir çözümü vardır,

2. İşlemin uygulanacağı matrisin köşegeninde sıfır yoktur. Herhangi bir elemanın sıfır olması durumunda satır değişikliğiyle bu şart sağlanır.

Jacobi iterasyonu, düzlemlerin döndürülmesi prensibine göre çalışır. Bu iterasyonda bulunacak öyle bir R dönüşüm matrisi vardır ki nxn olmak üzere kare bir matristir, $Y = RX$ doğrusal dönüşümünü sağlar. R dönüşüm matrisi aşağıdaki formu sağlamalıdır.

$$R = \begin{bmatrix} 1 & \dots & 0 & \dots & 0 & \dots & 0 \\ \vdots & & & & & & \vdots \\ 0 & \dots & cos\emptyset & \cdots & sin\emptyset & \dots & 0 \\ \vdots & & & & & & \vdots \\ 0 & \dots & -sin\emptyset & \dots & cos\emptyset & \dots & 0 \\ \vdots & & & & & & \vdots \\ 0 & \dots & 0 & \dots & 0 & \dots & 1 \end{bmatrix} \quad \begin{matrix} \\ \\ \text{satır p} \\ \\ \text{satır q} \\ \\ \\ \end{matrix} \qquad \begin{matrix} \\ \\ \\ \\ \\ \\ \text{sütun p} \quad \text{sütun q} \end{matrix} \tag{4.1}$$

$Y = RX$'de X n-boyutlu bir uzayı temsil eder. Dikkat edilecek olursa, matrisin köşegen elemanları $cos\emptyset$ haricinde 1; köşegen olmayan elemanları da $\mp sin\emptyset$ haricinde 0'dır. $Y = RX$ ile de düzlem döndürmesi sağlanmış olur.

$$j \neq p \text{ ve } j \neq q \text{ iken } y_j = x_j \tag{4.2a}$$

$$y_p = x_p cos\emptyset + x_q sin\emptyset, \tag{4.2b}$$

$$y_q = -x_p sin\emptyset + x_q cos\emptyset \tag{4.2c}$$

Denklem (4.2) takımı, uygun bir $\emptyset$ açısıyla n-boyutlu bir uzayın $x_p x_q$-düzlemince döndürüldüğünü ifade eder. Döndürme esnasında, $y_p = 0$ veya $y_q = 0$ elde edilir. Kullanılan R dönüşüm matrisi, ortogonal bir matristir ($R^T = R^{-1}$). Başlangıçtaki özdeğer problemimizi şu şekilde kabul edersek,

$$AX = \lambda X \tag{4.3}$$

A matrisinden dönüşümle bir D matrisi tanımlayalım,

$$D = R^T AR \tag{4.4}$$

yeni bir özdeğer problemi ortaya çıkar,

$$DY = \lambda Y \tag{4.5}$$

Başlangıçta elimizde olan özdeğer problemi ile Denklem (4.5) ile elde edilen özdeğer problemi karşılaştırıldığında ortaya çıkan sonuçlar şunlardır:

i. A simetrik matris ve R dönüşüm matrisi de ortogonal olduğunda, elde edilen D matrisi simetrik bir matris olur.

ii. D matrisinin özdeğerleri 1. maddede sayılan nedenlerden dolayı A matrisine eşittir.

Jacobi iterasyonunun her bir adımı, başlangıçtaki A matrisine ait köşegen olmayan elemanlardan sadece ikisini, a_{pq} ve a_{qp}'yi sıfıra yaklaştırmayı başarır. İlk ortogonal matris R_1 ile işlemlere başlanır.

$$D_1 = R_1^T A R_1 \tag{4.6}$$

R_1 de Denklem (4.1) ile ifade edilen dönüşüm matrisi ile aynı düzeni gösterir. Denklem (4.6) ile gösterilen dönüşüm sadece p ve q satırı ile p ve q sütununda değişim yapar. A matrisinin R_1 ile sağdan çarpılmasıyla $B = AR_1$ şu formda elde edilir:

$$B = \begin{bmatrix} a_{11} & \cdots & a_{1p} & \cdots & a_{1q} & \cdots & a_{1n} \\ a_{p1} & \cdots & a_{pp} & \cdots & a_{pq} & \cdots & a_{pn} \\ a_{q1} & \cdots & a_{qp} & \cdots & a_{qq} & \cdots & a_{qn} \\ a_{n1} & \cdots & a_{np} & \cdots & a_{nq} & \cdots & a_{nn} \end{bmatrix} \begin{bmatrix} 1 & \cdots & 0 & \cdots & 0 & \cdots & 0 \\ 0 & \cdots & c & \cdots & s & \cdots & 0 \\ 0 & \cdots & -s & \cdots & c & \cdots & 0 \\ 0 & \cdots & 0 & \cdots & 0 & \cdots & 1 \end{bmatrix} \tag{4.7}$$

Elde edilen B matrisinde 1'den $p-1$'inci ile $p+1$'den $q-1$'inci ve $q+1$'den n'inci sütuna kadar bir değişme olmamıştır. Sadece p ve q'uncu sütunlar değişmiştir.

$$k \neq p \text{ ve } k \neq q \text{ iken } b_{jk} = a_{jk} \tag{4.8a}$$

$$b_{jp} = ca_{jp} - sa_{jq}; \qquad j = 1,2,\dots,n \tag{4.8b}$$

$$b_{jq} = sa_{jp} + ca_{jq}; \qquad j = 1,2,\dots,n \tag{4.8c}$$

Aynı mantıkla A matrisinin R_1^T ile soldan çarpımı da sadece p ve q'uncu satırlarda değişimi sağlar. Sonuç olarak, Denklem (4.6) ile başlanan dönüşümde p ve q'uncu sütun ile p ve q'uncu satırda değişim sağlanmıştır. Genel olarak, birinci iterasyonun sonucu D_1 matrisinin d_{jk} elemanları, düzlem dönüşümlerini de içerecek şekilde şöyle hesaplanır:

$$j \neq p \text{ ve } j \neq q \text{ iken } d_{jp} = ca_{jp} - sa_{jq} \tag{4.10a}$$

$$j \neq p \text{ ve } j \neq q \text{ iken } d_{jq} = sa_{jp} + ca_{jq} \tag{4.10b}$$

$$d_{pp} = c^2 a_{pp} + s^2 a_{qq} - 2csa_{pq}, \tag{4.10c}$$

$$d_{qq} = s^2 a_{pp} + c^2 a_{qq} + 2cs a_{pq}, \quad \textbf{(4.10d)}$$

$$d_{pq} = (c^2 - s^2) a_{pq} + cs(a_{pp} - a_{qq}) \quad \textbf{(4.10e)}$$

diğer elemanlar da simetriklik özelliği ile bulunur.

Bu aşamada, döndürmeyi sağlayacak $\emptyset$ açısının nasıl belirleneceği önemlidir. Bunun için bazı trigonometrik dönüşümlerden faydalanılır.

$$\theta = cot2\emptyset = \frac{c^2 - s^2}{2cs} \quad \textbf{(4.11a)}$$

$a_{pq} \neq 0$ olduğunda $d_{pq} = 0$'a ulaşmak istiyoruz, o zaman Denklem (4.10e)'yi kullanarak,

$$0 = (c^2 - s^2) a_{pq} + cs(a_{pp} - a_{qq}) \quad \textbf{(4.11b)}$$

elde edilir. Bu denklemi düzenlersek

$$(c^2 - s^2)/_{cs} = (a_{qq} - a_{pp})/_{a_{pq}} \quad \textbf{(4.11c)}$$

elde edilir ve bu da Denklem (4.11a)'nın çözümümde kullanılırsa,

$$\theta = \frac{(a_{qq} - a_{pp})}{2a_{pq}} \quad \textbf{(4.11d)}$$

elde edilir. Daha sonra $tan\emptyset$ şu şekilde tanımlanır:

$$t = tan\emptyset = \frac{s}{c} \quad \textbf{(4.11e)}$$

Denklem (4.11a)'da pay ve payda c^2'ye bölünürse,

$$\theta = \frac{1 - s^2/c^2}{2s/c} \quad \textbf{(4.11f)}$$

ve

$$t^2 + 2t\theta - 1 = 0 \quad \textbf{(4.11g)}$$

elde edilir. Denklem (4.11g)'nin hesaplanan küçük kökü, küçük döndürme açısına ($|\emptyset| = \pi/4$) karşılık gelir. Ancak bundan farklı olarak, Denklem (4.11g)'nin çözümü için kullanılabilecek özel bir quadratik form olan,

$$t = -\theta \mp (\theta^2 + 1)^{1/2} = \frac{sign(\theta)}{|\theta| + (\theta^2 + 1)^{1/2}} \quad \textbf{(4.12)}$$

kullanıldığında c ve s şöyle elde edilir:

$$c = \frac{1}{(t^2+1)^{1/2}} \quad \textbf{(4.13a)}$$

$$s = ct \quad \textbf{(4.13b)}$$

Sonuçta Denklem (4.13) takımı ile kullanılacak trigonometrik değişkenler hesaplanmıştır. Tüm bunların yanında, iterasyonlar esnasında kullanılan dönüşüm matrisleri çarpımlarını da unutmamak gerekir. n. adımda Jacobi iterasyonunu durdurduğumuzu varsayarsak elimizde,

$$V_n = R_1 R_2 \dots R_n \quad \textbf{(4.14)}$$

matris çarpımı olur. Burada V_n'in ortogonal olduğunu unutmamak gerekir. Ayrıca V_n matrisi, özvektörlerin tutulduğu matristir. n. adımda elde edilen özvektörler için en başta $V = I$ ile başlamak en kolayı olacaktır. Çünkü her dönüşüm adımında birim matrise ait ilgili elemanlar da orantılı olarak değişecek ve anlık durum için ana matristeki özdeğerlerle ilişkili özvektörler hesaplanacaktır.

Jacobi iterasyonun en önemli noktası da, iterasyonun n. adımda durdurulmasını sağlayacak bir eşik değeri belirlenmesidir. ε ile ifade edilebilecek bu eşik değeri, programcı tarafından seçilebilir. Ne kadar küçük olursa, Jacobi iterasyonunun sonuca ulaşması o kadar zaman almasına karşılık, elde edilen sonuçlar o kadar doğru olacaktır. Eşik değeriyle matris elemanları satırlarında sırayla karşılaştırılır. Daha sonra, köşegen elemanlarının karesel toplamları hesaplanır.

$$T_0 = \sum_{j=1}^{n} \left|a_{jj}\right|^2 \quad \textbf{(4.15a)}$$

daha sonra,

$$T_1 = \sum_{j=1}^{n} \left|d_{jj}\right|^2 \quad \textbf{(4.15b)}$$

olarak hesaplama devam eder. Sonuç olarak her adımda elde edilen D_j'ler dizisi, köşegen matrise yakınsar. Eşik değeri yardımıyla,

$$\left|a_{pq}\right| > \varepsilon \left(\frac{T_0}{n}\right)^{1/2} \quad \textbf{(4.16)}$$

ile karşılaştırılan a_{pq} elemanı yukarıdaki şartı sağladığında sıfırlanır. Bu şekilde A matrisi, sonuçta köşegen matrise çevrilir ve köşegenlerde özdeğerler elde edilir. Jacobi iterasyonuna ait algoritma adım adım aşağıda sunulmuştur.

1. Jacobi iterasyonundan çıkmak için kullanılacak eşik değerine karar verilir. Özdeğer ve özvektöre ilişkin başlangıç matrisleri belirlenir. Başlangıçta özdeğer matrisinin A; özvektör matrisinin de birim matris olarak belirlenmesi işlemleri kolaylaştırır.
2. A matrisinde bulunan en büyük elemanın satır ve sütunu bulunur.
3. Yukarıda verilen formüller yardımıyla tanjant, kosinüs ve sinüs değerleri hesaplanır.
4. Elde edilen kosinüs ve sinüs değerleri ile dönüşüm matrisi belirlenir.
5. Denklem (4.6) ile gösterilen D matrisi hesaplanır.
6. D matrisinden sonra, Denklem (4.14)'ya istinaden özvektörlerin tutlacağı V matrisi bulunur.
7. Elde edilen D matrisinde en büyük elemanın satır ve sütunu bulunur.
8. Denklem (4.16) yardımıyla da karşılaştırma yapılır. Eğer belirtilen şart sağlanırsa, döngü o eleman için sonlanır ve diğer elemanlara geçilir.
9. Tüm elemanlar için aynı şart sağlanana kadar 3 ile 8 arasındaki işlemler tekrarlanır.

4.2 LU Ayrıştırması ile Matris Tersinin Hesaplanması

LU ayrıştırma yöntemi, doğrusal sistemlerin çözümünde kullanılan bir yöntemdir. Bunun dışında, determinantları sıfır olmadığı takdirde, büyük boyutlu matrislerin tersini bulmak için de kullanılabilir. Bu çalışmada, LU ayrıştırması, matris tersi bulmak için kullanılmıştır. Tersi bulunan matris ise, FDAY'da ortaya çıkan,

$$\boldsymbol{S_w}^{-1}\boldsymbol{S_B}\boldsymbol{w_i} = \boldsymbol{\lambda_i}\boldsymbol{w_i} \qquad \textbf{(4.17)}$$

sistemdeki sınıf içi saçılma matrisidir ($\boldsymbol{S_w}$).

LU ayrıştırması teoride şu temel doğrultusunda çalışır: Herhangi bir tekil olmayan A matrisi, biri alt üçgensel; diğeri üst üçgensel matris olmak üzere iki matrisin çarpımı olarak yazılabiliyorsa, "*A matrisi LU ayrıştırmasına sahiptir*" denir.

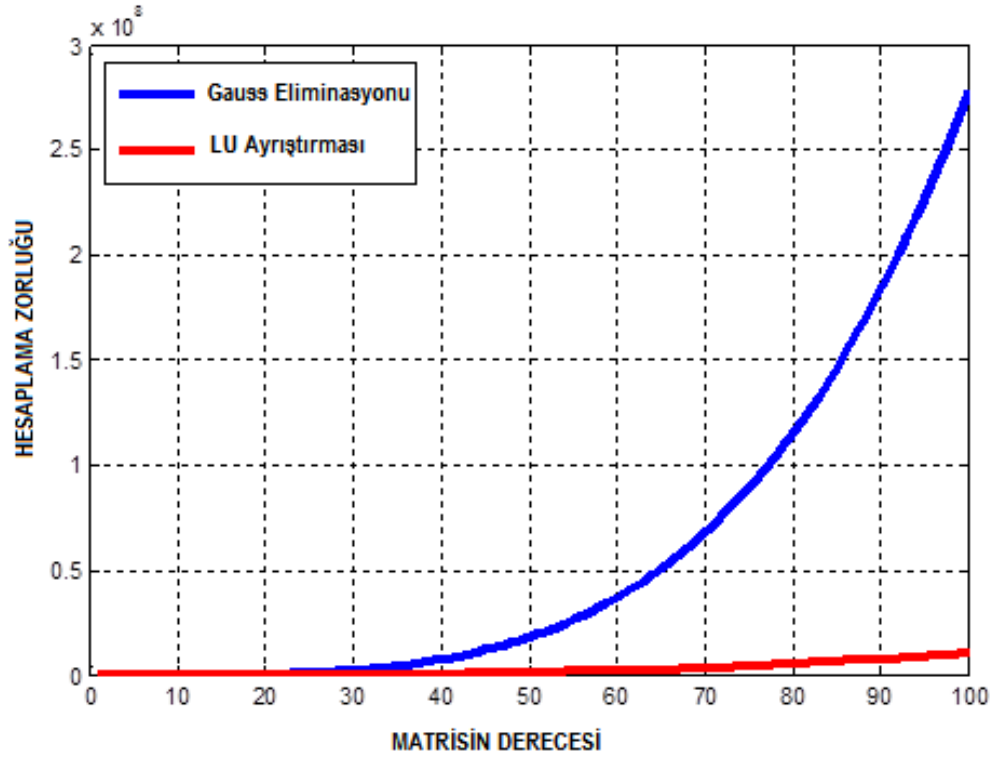
$$\boldsymbol{A = LU} \qquad \textbf{(4.18)}$$

Tanımı yapılan ayrıştırma Denklem (4.18) ile ifade edilebilir. Denkleme göre ***L***, alt üçgensel; ***U*** ise üst üçgensel matrisi ifade eder. Aynı denklem matrissel olarak da şu şekilde gösterilebilir.

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ a_{41} & a_{42} & a_{43} & a_{44} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ m_{21} & 1 & 0 & 0 \\ m_{31} & m_{32} & 1 & 0 \\ m_{41} & m_{42} & m_{43} & 1 \end{bmatrix} \begin{bmatrix} u_{11} & u_{12} & u_{13} & u_{14} \\ 0 & u_{22} & u_{23} & u_{24} \\ 0 & 0 & u_{33} & u_{34} \\ 0 & 0 & 0 & u_{44} \end{bmatrix} \quad \textbf{(4.19)}$$

Denklem (4.19)'de gösterilen matris formundan da anlaşılacağı üzere, alt üçgensel matrisin köşegenlerinde 1 bulunur. ***A*** matrisinin tekil olmaması ise ayrıştırma sonunda üst üçgensel matriste (***U***) köşegen elemanlarının sıfır olmamasını sağlar ($u_{kk} \neq 0$).

LU ayrıştırması, büyük matrislerin tersini hesaplamada yeterli bir yol sunar. Bilindiği üzere matris tersi, *Gauss Eliminasyonu* yöntemiyle de hesaplanabilir. Ancak LU ayrıştırması, Gauss yöntemine göre hesaplama kolaylığı sağlamakla birlikte, işlem hızını da arttırmaktadır. Gauss eliminasyonu ile LU ayrıştırmasının matris tersi hesaplanırken hız olarak karşılaştırılması Şekil 4.1'de gösterilmiştir.



Şekil 4.1 : Gauss eliminasyonu ve LU ayrıştırmasının karşılaştırılması.

LU ayrıştırmasının doğrusal sistemlerin çözümünde kullanılan bir yöntem olduğundan konu başında bahsedilmişti. Bu bilgi ışığında, teoriyi kolayca kavramak

açısından, $\boldsymbol{A}$ matrisi, doğrusal bir sistemin katsayılar matrisi olarak düşünülsün. LU ayrıştırması uygulanırken $\boldsymbol{A}$ matrisinde işlem kolaylığı açısından Gauss eliminasyonunda olduğu gibi satır sütun işlemleri gerekebilir. Ancak, köşegendeki bazı elemanların (pivot elemanları) sıfır olması durumunda satır değişikliğine gidilir. Bu işlem ileriki aşamalarda ortaya çıkacak $^{sayı}/_{0}$ belirsizliğini ortadan kaldırmak içindir. $\boldsymbol{AX = B}$ doğrusal sistemin çözümü,

$$\boldsymbol{LUX = B} \qquad \textbf{(4.20)}$$

denklem sisteminin çözümü ile bulunur.

Burada işlemler sırasıyla;

1. $\boldsymbol{Z = UX}$ dersek öncelikle

$$\boldsymbol{LZ = B} \qquad \textbf{(4.21a)}$$

sistemini $\boldsymbol{Z}$ için,

2. Daha sonra da

$$\boldsymbol{UX = Z} \qquad \textbf{(4.21b)}$$

sistemini $\boldsymbol{X}$ için çözerek uygulanır.

Bu bilgiden sonra, doğrusal sistem çözümünden matris tersini bulma işlemine dönecek olursak, $\boldsymbol{AX = B}$ sistemi şu şekilde düşünüldüğünde sonuca ulaşılır. $\boldsymbol{X}$ matrisi $\boldsymbol{A}$ matrisinin tersi olsun ($\boldsymbol{A^{-1} = X}$). Dolayısıyla,

$$\boldsymbol{AX = I} \qquad \textbf{(4.22)}$$

halini alır. Daha sonra $\boldsymbol{A}$'nın LU ayrıştırılması bulunduğunda, yukarıdaki iki adım uygulanarak matrisin tersi hesaplanmış olur. Burada $\boldsymbol{B}$ matrisi birim matris olduğundan, 1. adımda yapılan her hesaplamada birim matrisin sırayla sütunlarından faydalanılır. Sonuçta elde edilen $\boldsymbol{X}$ matrisi, tersi aranan $\boldsymbol{A}$ matrisinin tersidir.

LU ayrıştırmasıyla matris tersini hesaplamak için uygulanması gereken algoritma adım adım aşağıda olduğu gibidir.

1. Gauss eliminasyonunda olduğu gibi, satır ve sütun işlemleriyle köşegenlerdeki pivot noktaları bulunur ve pivotlara istinaden elde edilen katsayılar yardımıyla pivot altında kalan sayılar satır işlemi yapılarak basitleştirilir.
2. Basitleştirilen matris LU ayrıştırması için uygundur. Elde bulunan matris $\boldsymbol{L}$ ve $\boldsymbol{U}$ matrislerinin ikisini de içerir. Başka bir deyişle şu formu sağlar:

$$\begin{bmatrix} u_{11} & u_{12} & u_{13} & u_{14} \\ m_{21} & u_{22} & u_{23} & u_{24} \\ m_{31} & m_{32} & u_{33} & u_{34} \\ m_{41} & m_{42} & m_{43} & u_{44} \end{bmatrix}$$

3. Burada her iki matrisi de ayırmak gerekir. ayırma işlemi Denklem (4.21)'deki form gözetilerek yapılır.
4. Önce $\boldsymbol{LZ = B}$; ardından $\boldsymbol{UX = Z}$ çözülerek matris tersi hesaplanır.

4.3 QR Ayrıştırma Algoritması ile Özdeğer/Özvektör Bulunması

QR ayrıştırma algoritması, doğrusal cebir hesaplamalarında özdeğer/özvektör çiftlerini bulmak için kullanılan, en bilinen algoritmalardandır. QR ayrıştırma algoritmasındaki amaç, $\boldsymbol{A}$ kare matrisini, özelliklerini koruyarak, iki adet farklı matris çarpımı olarak yazmaktır. $\boldsymbol{A}$ kare matrisinin QR ayrıştırması şu şekilde ifade edilir.

$$\boldsymbol{A = QR} \qquad \textbf{(4.23)}$$

Denklem (4.23)'te gösterilen formda, $\boldsymbol{R}$ matrisi üst üçgensel; $\boldsymbol{Q}$ matrisi ise ortogonal matristir ($\boldsymbol{Q^T Q = I}$). QR ayrıştırmasında elde edilen sonuç, $\boldsymbol{A}$ matrisinin tersi mevcut olduğunda tek bir çözüme ulaşır. Söz konusu ayrıştırma, kare matrislere uygulanmasının yanısıra daha genel bir ifade ile dikdörtgen matrislere de uygulanabilmektedir.

Bu çalışma içerisinde QR ayrıştırması, LU ayrıştırması ile sınıf içi saçılma matrisinin tersi elde edildikten sonra ortaya çıkan ve Denklem (4.17) ile hatırlatılan sisteme ait özdeğer/özvektörlerin hesaplanmasında kullanılmıştır. Bu sistem simetrik olmadığından, daha önce detaylarıyla işlenen Jacobi iterasyon yöntemi bu sistemde kullanılamamaktadır.

$\boldsymbol{A}$ kare matrisine ait QR ayrıştırması, basit bir mantıkla şu yöntem izlenerek bulunabilir. Öncelikle, $\boldsymbol{A}$ matrisine benzer ve özdeğer/özvektör çifti kolay hesaplanabilen bir $\boldsymbol{B}$ matrisi bulunur. $\boldsymbol{A}$ ve $\boldsymbol{B}$ matrisleri benzer olduğundan, her ikisi de aynı özdeğer/özvektör çiftine sahiptir. Dolayısıyla da $\boldsymbol{A}$ matrisinin özdeğer/özvektör çifti de kolayca hesaplanmış olur.

Bu mantığın uygulanması için şöyle bir yol izlenebilir. $\boldsymbol{Q}$ ve $\boldsymbol{R}$ matris çifti, $\boldsymbol{A}$ matrisine ait olsun. $\boldsymbol{Q}$ ortogonal olduğundan $\boldsymbol{Q^{-1} = Q^T}$'dir.

$$\boldsymbol{Q^T AQ = Q^T (QR) Q = (Q^T Q) RQ = RQ} \qquad \textbf{(4.24)}$$

Denklem (4.24)'dan da anlaşılacağı üzere, $\boldsymbol{QR}$ ve $\boldsymbol{RQ}$ matrisleri benzer matrisler olarak karşımıza çıkar.

$\boldsymbol{A}^{(1)} = \boldsymbol{R}^{(1)}\boldsymbol{Q}^{(1)}$ diyelim. Yukarıdaki denklemle elde edilen sonuca göre $\boldsymbol{A}^{(1)}$ matrisi $\boldsymbol{A}$ matrisine benzer olur. Aynı zamanda, $\boldsymbol{A}^{(1)} = \boldsymbol{Q}^{(2)}\boldsymbol{R}^{(2)}$ olarak da ifade edilebilir. Bu işlem takımına bir dizi olarak devam edilirse, $\boldsymbol{A}$'ya benzer matrisler dizisi elde edilir. Uygulanan bu işlemler takımı, bir üst üçgensel matrise ($\boldsymbol{T}$) yakınsar. Elde edilen matrisin köşegenleri de kendisine ait özdeğerlerdir. Benzerlik kuralına göre $\boldsymbol{T}$ matrisi $\boldsymbol{A}$ matrisine benzer olduğundan köşegenlerdeki özdeğerler aynı zamanda $\boldsymbol{A}$'nın da özdeğerleridir.

QR ayrıştırmasında, $\boldsymbol{Q}$ ve $\boldsymbol{R}$ matrisleri her r_{ik} için sütun sütun bulunur. Adım adım bu işlem şöyle ifade edilir. Öncelikle r_{ik} şu şekilde hesaplanabilir:

$$r_{ik} = col\boldsymbol{A}_i^{(k)^T} col\boldsymbol{Q}_k \qquad \textbf{(4.25)}$$

Ardından,

$$\boldsymbol{A}^{(j)} = \boldsymbol{A} - \sum_{i=1}^{j-1} col\boldsymbol{Q}_i row\boldsymbol{R}_i^T = \sum_{i=j}^{n} col\boldsymbol{Q}_i row\boldsymbol{R}_i^T \qquad \textbf{(4.26)}$$

gözetilerek, $i < k$ iken $\boldsymbol{A}^{(i)}$'ye ait k. sütunu,

$$\boldsymbol{A}^{(j)}\boldsymbol{e}_k = col\boldsymbol{A}_k^{(j)} = col\boldsymbol{A}_k - \sum_{i=1}^{j-1} col\boldsymbol{Q}_i r_{ik} = \sum_{i=j}^{k} col\boldsymbol{Q}_i r_{ik} \qquad \textbf{(4.27)}$$

olarak hesaplanır. $i > k$ için $row\boldsymbol{R}^T\boldsymbol{e}_k = 0$ ve $\boldsymbol{R}$ üçgensel bir matris olduğundan Denklem (4.29)'un sağ tarafındanki toplam işaretinin üst sınırı k'dır. Bu nedenle, $col\boldsymbol{A}_{\boldsymbol{k}}^{(k)}$ şöyle hesaplanır:

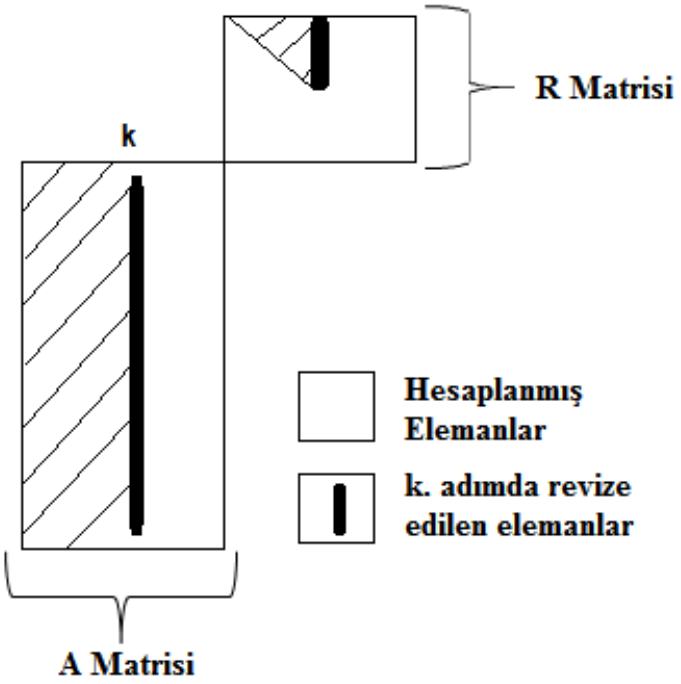
$$col\boldsymbol{A}_k^{(1)} = col\boldsymbol{A}_k \qquad \textbf{(4.28a)}$$

$$col\boldsymbol{A}_k^{(j)} = col\boldsymbol{A}_k^{(j-1)} - col\boldsymbol{Q}_{(j-1)} r_{j-1,k} \quad j = 2, \dots, k \qquad \textbf{(4.28b)}$$

Denklem takımı (4.28) yukarıda da bahsedildiği üzere, $\boldsymbol{Q}$ ve $\boldsymbol{R}$ matrislerinin sütun sütun hesaplandığını gösterir.

$$col\boldsymbol{A}_{\boldsymbol{k}}^{(i+1)} = col\boldsymbol{A}_{\boldsymbol{k}}^{i} - col\boldsymbol{Q}_i r_{ik} \qquad \textbf{(4.29)}$$

Bu yöntemle elde edilen hasaplama, matris üzerinde Şekil 4.2'deki gibidir. Şekle göre, kullanılan yöntem ile $\boldsymbol{A}$ matrisinde k. adımda, $k+1$'den n. sütuna kadar değişme gözlenmez ve hesaplama bu adım için sütun olarak yapılır. Buna paralel olarak da $\boldsymbol{R}$ matrisi aynı şekilde hesaplanır.



Şekil 4.2 : QR Algoritması ile hesaplamanın matris üzerindeki etkisi.

5. ALGORİTMİK TESTLER

Bu çalışmanın konusu olan kulak biyometrisi kullanarak TBA ve FDAY tabanlı gömülü bir tanıma sistemi tasarımına, öncelikle bilgisayar ortamında kullanılan analiz yöntemleri ve algoritmaların test edilmesiyle başlanmıştır.

Bu çalışmada kullanılan veri tabanına ait resimler, daha önceden Bahattin Kocaman tarafından sağlanan *Midas'ın Kulakları* isimli kulak veri tabanıdır. Veri tabanı, 50 farklı kişiye ait kulak resimlerinden oluşur. Her kişi için 6 adet sağ 6 adet sol olmak üzere toplam 12 resim bulunmaktadır. Resimler 0 ile 15 derece arasında değişen açılarda ve aynı ışık şartları altında çekilmiştir. Algoritmik testlerde 20 kişi eğitim kümesi için seçilmiş, her kişiye ait 5 adet resim olmak üzere toplamda 100 adet resim kullanılmıştır.

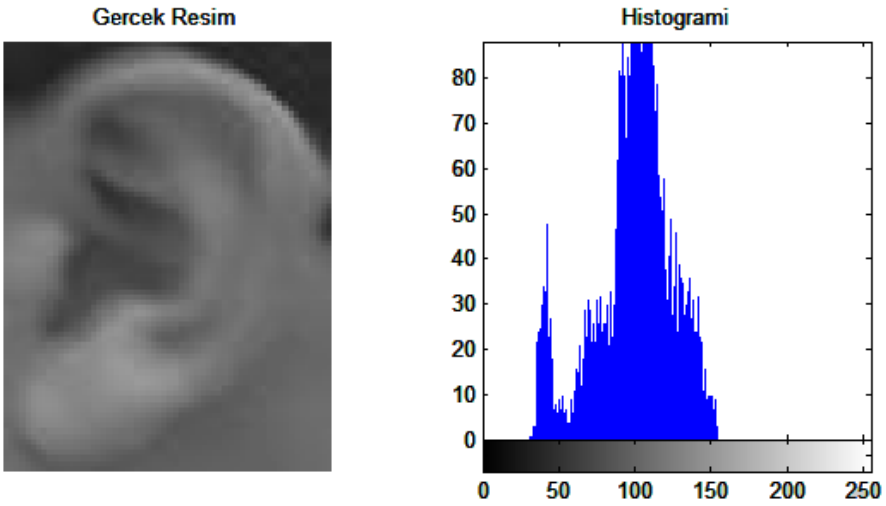
Test kümesi oluşturulurken, eğitim kümesinde kullanılan 20 kişinin farklı olan 1'er adet resmi ile kalan 30 kişi arasından seçilen rastgele 20 kişinin 1'er resmi ve kulak resminden farklı 10 adet değişik resimle toplam 50 adet resim kullanılmıştır.

Yapılan testler esnasında, eldeki resimler 80x60 boyutuna indirgenmiş ve gri seviyeye çevrilmiştir. Testler yapılırken, sisteme eğitim ve test aşamalarında gri seviyeye çevrilmiş resimlere ait matrisler girdi olarak uygulanmıştır.

Ancak resim matrislerinden farklı olarak yapılan bir başka yöntemde, algoritmalara girdi olarak herbir resime ait histogram bilgisi verilmiştir. Bilindiği üzere histogram, bir resimdeki renk değerlerinin sayısını gösteren bir grafiktir. 0-255 arası değerler, gri seviye bir resimde beyazdan siyaha kadar olan aradaki renkleri tanımlar. Dolayısıyla, tanımdan da anlaşılacağı üzere histogram, bu seviyedeki renklerin dağılım grafiğini gösterir. Şekil 5.1'de örnek olarak herhangi bir kulak resmi ve buna ait histogramını göstermektedir.

Bu bölümde, bilgileri verilen tanıma yöntemlerinin bilgisayar ortamında yapılan testlerine ait elde edilen sonuçlar gösterilecek ve irdelenecektir. Öncelikle girdi olarak resim matrisleri kullanıldığında elde edilen sonuçlar; ardından histogram

matrisleri kullanıldığında elde edilen sonuçlar değerlendirilecektir. Yapılan testlerde karşılaştırma ve karar aşaması için *Öklid Uzaklığı* yöntemi tercih edilmiştir.



Şekil 5.1 : Örnek bir kulak resmi ve histogramı.

5.1 Resim Matrisleriyle Elde Edilen Sonuçlar

Bu bölümde, test edilen TBA ve FDAY yöntemlerine girdi olarak resim matrisleri uygulanmıştır. Sırasıyla öncelikle TBA, daha sonra FDAY test edilmiş ve elde edilen sonuçlar çizelgelerle sunulmuş, bölüm sonlarında elde edilen sonuçlar irdelenmiştir.

5.1.1 TBA'nın test edilmesi ve elde edilen sonuçlar

TBA, doğruluk ve yeterlilik olarak bilgisayar ortamında, MATLAB programı yardımıyla test edilmiştir. Bölüm 3.1'de teorik olarak işlenen TBA, kendi özelliği gereği içerisinde boyut küçültme işlemi de gerçeklenerek testler icra edildiği gibi elde edilen tüm özdeğer/özvektör çiftleriyle de testler yapılmıştır. Yapılan testler esnasında, test ve tanıma aşaması için kullanılan öklid uzaklığında iki farklı yöntem izlenmiştir. Buna göre öklid uzaklığı,

1. Kulak uzayı ve sınıf uzayı için aynı,
2. Kulak uzayı ve sınıf uzayı için farklı,

seçilmiştir. Bunun için değer atanması, testler sırasında MATLAB programı üzerinde elde edilen sonuçlar gözlenerek ve bu sonuçlara istinaden ayırımı ve tanımayı en iyi sağlayacak değerler olarak belirlenmiştir. Buna göre elde edilen sonuçlar ve kullanılan öklid uzaklıkları ile birlikte hazırlanan çizelgeler şu şekildedir.

1. TBA'da öklid uzaklığı eşit ve kovaryans matrisinden elde edilen tüm özdeğer/özvektör çiftleri kullanılarak yapılan testlerde elde edilen sonuçlar:

Çizelge 5.1 : TBA'da öklid uzaklığı eşit seçilerek ve tüm özdeğer/özvektör çifti kullanılarak yapılan testlerde elde edilen sonuçlar.

Sağ Kulak ($\varepsilon = \varepsilon_k = 1150$)		Sol Kulak ($\varepsilon = \varepsilon_k = 1250$)	
Örnek Sayısı	Tanıma Oranı (%)	Örnek Sayısı	Tanıma Oranı (%)
2	72	2	80
3	86	3	90
4	92	4	92
5	92	5	92

2. TBA'da öklid uzaklıkları farklı ve kovaryans matrisinden elde edilen tüm özdeğer/özvektör çiftleri kullanılarak yapılan testlerde elde edilen sonuçlar:

Çizelge 5.2 : TBA'da öklid uzaklıkları farklı seçilerek ve tüm özdeğer/özvektör çifti kullanılarak yapılan testlerde elde edilen sonuçlar.

Sağ Kulak ($\varepsilon_k = 950; \varepsilon = 1150$)		Sol Kulak ($\varepsilon_k = 1150; \varepsilon = 1250$)	
Örnek Sayısı	Tanıma Oranı (%)	Örnek Sayısı	Tanıma Oranı (%)
2	76	2	76
3	86	3	86
4	94	4	94
5	94	5	94

3. TBA'da öklid uzaklıkları eşit; ancak kovaryans matrisinden elde edilen özdeğer/özvektör çiftlerinden ağırlığı yüksek olan, toplam adedin %99'una karşılık gelecek sayıda çiftin kullanılmak üzere seçilmesi ile yapılan testlerde elde edilen sonuçlar:

Çizelge 5.3 : TBA'da öklid uzaklığı eşit ancak özdeğer/özvektör çiftinin %99'luk kısmının kullanılmasıyla elde edilen sonuçlar.

Sağ Kulak ($\varepsilon = \varepsilon_k$ =1150)			Sol Kulak ($\varepsilon = \varepsilon_k$ =1250)		
Örnek Sayısı	M'/M	Tanıma Oranı (%)	Örnek Sayısı	M'/M	Tanıma Oranı (%)
2	28/40	64	2	29/40	72
3	39/60	76	3	39/60	86
4	48/80	84	4	48/80	86
5	54/100	84	5	57/100	90

4. TBA'da öklid uzaklıkları farklı ve kovaryans matrisinden elde edilen özdeğer/özvektör çiftlerinden ağırlığı yüksek olan, toplam adedin %99'una karşılık gelecek sayıda çiftin kullanılmak üzere seçilmesi ile yapılan testlerde elde edilen sonuçlar:

Çizelge 5.4 : TBA'da öklid uzaklığı farklı ve özdeğer/özvektör çiftinin %99'luk kısmının kullanılmasıyla elde edilen sonuçlar.

Sağ Kulak (ε_k = 950; ε =1150)			Sol Kulak (ε_k =1150; ε =1250)		
Örnek Sayısı	M'/M	Tanıma Oranı (%)	Örnek Sayısı	M'/M	Tanıma Oranı (%)
2	28/40	70	2	29/40	78
3	39/60	82	3	39/60	88
4	48/80	90	4	48/80	90
5	54/100	92	5	57/100	92

5. Yapılan 4 deneme sonucunda, oklid uzaklıklarının tanıma oranlarına etkisinin yadsınamayacak derecede önemli olduğu görülmüştür. Bu nedenle, tanıma oranının arttırılması için, kullanılan özdeğer/özvektör çifti sayısı değiştirilmeden, farklı öklid uzaklıklarıyla denemeler yapılmaya devam edilmiştir. Elde edilen sonuçlara göre, tanıma yüzdesinin en yüksek olduğu öklid uzaklıkları ve bunlara ilişkin sonuçlar Çizelge 5.5'te sunulmuştur.

Çizelge 5.5 : TBA'da maksimum tanıma oranı için kullanılan değerler ve elde edilen sonuçlar.

Sağ Kulak (ε_k = 950; ε =1500)			Sol Kulak (ε_k = 1200; ε =1350)		
Örnek Sayısı	M'/M	Tanıma Oranı (%)	Örnek Sayısı	M'/M	Tanıma Oranı (%)
2	28/40	88	2	29/40	84
3	39/60	92	3	39/60	92
4	48/80	94	4	48/80	92
5	54/100	94	5	57/100	92

Çizelge 5.1 ve Çizelge 5.3 incelendiğinde, aynı öklid uzaklığı kullanıldığında kullanılan özdeğer/özvektör çiftlerinin sayısı tanıma oranında etkili olduğu görülmektedir. TBA'nın teorik olarak anlatıldığı Bölüm 3.1'den de hatırlanacağı üzere, özdeğer/özvektör çiftleri veri hakkında çeşitli bilgiler taşır ve kullanılan çiftlerin sayısı azaldığında, elde bulunan bilgi seviyesi de azalmaktadır. Dolayısıyla, Çizelge 5.1 ve Çizelge 5.3'e göre kullanılan çiftlerin sayısı azaldığında, tanıma oranları da azalmıştır. Bunlara paralel olarak, Çizelge 5.2 ve Çizelge 5.4'e göre de benzer sonuçlar elde edilmiştir. Testlerde kullanılan öklid uzaklıkları sınıf ve kulak uzayları için farklı olmakla birlikte, aynı değerler kullanarak yapılmıştır. Yapılan gözlemler neticesinde, öklid uzaklıkları sınıf uzayı ve kulak uzayı için farklı seçilmesine rağmen, kullanılan özdeğer/özvektör çifti sayısı azaldığından tanıma oranı da azalmıştır.

Test sonuçları farklı bir yönden irdelendiğinde, tüm özdeğer/özvektör çiftleri kullanıldığında, Çizelge 5.1 ve Çizelge 5.2'ye göre öklid uzaklıkları sınıf ve kulak uzayları için farklı seçildiğinde tanıma oranları artmıştır. Aynı husus, kullanılan çift sayısının %99'a karşılık gelecek şekilde seçilmesi durumunda da gözlenmiştir. Kullanılan çiftlerin sayısı azalmasına rağmen, öklid uzaklıklarının farklı seçilmesi, tanıma oranını arttırmıştır.

Çizelge 5.5'te ise bununla paralel olarak, öklid uzaklıklarında yapılan bir revizyon ile elde edilen tanıma oranları arttırılabilmiştir. Buna göre, her sınıf için 2 ve 3 örnek bulunması halinde elde edilen tanıma oranları artmıştır. Ancak sol kulak için 4 ve 5 örnekli denemede elde edilen oran %2 seviyesinde bir azalma göstermiş, sağ kulak için ise aynı kalmıştır.

5.1.2 FDAY'nin test edilmesi ve elde edilen sonuçlar

FDAY, TBA gibi doğruluk ve yeterlilik olarak bilgisayar ortamında, MATLAB programı yardımıyla test edilmiştir. FDAY uygulanmadan önce, resim matrislerine TBA uygulanmış, TBA sonucu elde edilen yansıtılmış resim matrisleri FDAY için eğitim kümesi olarak kullanılmıştır. FDAY için yapılan testler iki farklı şekilde uygulanmıştır:

1. TBA aşamasında tüm özdeğer/özvektör çiftleri kullanılarak
2. TBA aşamasında %99'a karşılık gelen sayıda özdeğer/özvektör çifti kullanılarak

FDAY'da kullanılan öklid uzaklığı tektir. Çünkü her sınıf tek bir değerle temsil edilir ve elde edilen sonucun veri tabanında olup olmadığı kontrol edilir. Eğer veri tabanında olduğu sonucuna varılırsa, hangi sınıfa yakınsa o sınıf için sonuca varılır.

FDAY için elde edilen sonuçlar ve bunlara istinaden hazırlanan çizelgeler şu şekildedir.

1. TBA aşamasında tüm özdeğer/özvektör çiftleri kullanılarak yapılan denemelerde elde edilen sonuçlar:

Çizelge 5.6 : TBA aşamasında tüm özdeğer/özvektör çifti kullanılarak yapılan FDAY'de elde edilen sonuçlar.

Sağ Kulak			Sol Kulak		
Örnek Sayısı	Tanıma Oranı (%)	ε	Örnek Sayısı	Tanıma Oranı (%)	ε
2	100	565	2	92	600
3	90	475	3	100	390
4	90	430	4	100	370
5	100	300	5	100	300

2. TBA aşamasında özdeğer/özvektör çiftlerinden ağırlığı yüksek olan, toplam adedin %99'una karşılık gelecek sayıda çiftin kullanılmak üzere seçilmesi ile yapılan testlerde aynı öklid uzaklığı için elde edilen sonuçlar:

Çizelge 5.7 : TBA aşamasında özdeğer/özvektör çiftinin %99'luk kısmının aynı öklid uzaklığı için kullanılmasıyla FDAY'de elde edilen sonuçlar.

Sağ Kulak				Sol Kulak			
Örnek Sayısı	M'/M	Tanıma Oranı (%)	ε	Örnek Sayısı	M'/M	Tanıma Oranı (%)	ε
2	28/40	84	565****	2	29/40	84	600***
3	39/60	96	475**	3	39/60	94	390***
4	48/80	96	430**	4	48/80	98	370*
5	54/100	96	300**	5	57/100	96	300**

3. TBA aşamasında tüm özdeğer/özvektör çiftleri kullanılarak, öklid uzaklığında yapılan revize sonunda icra edilen denemelerde elde edilen sonuçlar:

Çizelge 5.8 : TBA aşamasında tüm özdeğer/özvektör çifti ile ökild uzaklığındaki revize sonunda yapılan FDAY'de elde edilen sonuçlar.

Sağ Kulak			Sol Kulak		
Örnek Sayısı	Tanıma Oranı (%)	ε	Örnek Sayısı	Tanıma Oranı (%)	ε
2	100	565	2	92	600
3	96	325[§]	3	100	390
4	96	300[§]	4	100	370
5	100	300	5	100	300

4. TBA aşamasında özdeğer/özvektör çiftlerinden ağırlığı yüksek olan, toplam adedin %99'una karşılık gelecek sayıda çiftin kullanılmak üzere seçilmesi ve öklid uzaklığındaki revize sonunda yapılan testlerde elde edilen sonuçlar:

Çizelge 5.9 : TBA aşamasında özdeğer/özvektör çiftinin %99'luk kısmı ile öklid uzaklığındaki revize sonunda FDAY'de elde edilen sonuçlar.

Sağ Kulak				Sol Kulak			
Örnek Sayısı	M'/M	Tanıma Oranı (%)	ε	Örnek Sayısı	M'/M	Tanıma Oranı (%)	ε
2	28/40	84	560‡‡‡	2	29/40	90	530‡‡
3	39/60	98	603	3	39/60	96	500‡
4	48/80	100	505	4	48/80	100	425
5	54/100	100	420	5	57/100	100	360

Çizelge 5.6 ve Çizelge 5.7 incelendiğinde, aynı öklid uzaklıkları kullanıldığında kullanılan özdeğer/özvektör çiftlerinin sayısı tanıma oranında etkili olduğu görülmektedir. Her ne kadar çizelgelerden, 3 ve 4 örnek sayısı için özdeğer/özvektör sayısı azaldığında tanıma oranın arttığı gözlense de, elde edilen veriler tanıma esnasında "*" ile gösterilen koşullar için veri tabanında olmasına rağmen bazı test resimlerinin tanınamadığını göstermiştir. Buna göre, '*' ile gösterilen koşullarda 1'er; '**' ile gösterilen koşullarda 2'şer; '***' ile gösterilen koşullarda 3'er ve son olarak '****' ile gösterilen koşullarda 4'er adet test resminin veri tabanında olmasına rağmen tanınamadığı tespit edilmiştir.

Elde edilen sonuçlar farklı bir yönden irdelendiğinde, öklid uzaklığının önemi tekrar gündeme gelmektedir. Çizelge 5.8 ve Çizelge 5.9'a göre, revize edilen öklid uzaklığı, tanıma oranlarında artışa neden olmıştur.

Çizelge 5.8'de 3 ve 4 örnekli durum için bahsedilen husus daha iyi anlaşılacaktır. Tüm özdeğer/özvektör çiftleri kullanıldığında ve öklid uzaklığında revizyon yapıldığında, bu durumlar için tanıma oranları %6 seviyesinde artmıştır. Ancak elde edilen sonuçlara göre bu hususun bir başka negatif tarafı da Çizelge 5.7'de gözlendiği gibi, bazı test resimleri veri tabanında olmasına rağmen tanınamamıştır. '§' ile ifade eden durmlarda 2'şer adet test resmi bu durumdadır.

Çizelge 5.9 da tüm bunlara paralel olarak sonuçlar vermektedir. TBA aşamasında kullanılan özdeğer/özvektör sayısı azalmasına rağmen, öklid uzaklığındaki revize ile tanıma oranı arttırılmıştır. Ancak yine bazı test resimleri veri tabanında olmasına rağmen tanınamamıştır. Buna göre, '‡' ile ifade edilen durumda 1; '‡‡' ile ifade edilen durumda 4; '‡‡‡' ile ifade edilen durumda ise 5 adet test resmi bu şekildedir.

5.2 Histogram Matrisleriyle Elde Edilen Sonuçlar

Bu bölümde, test edilen TBA ve FDAY yöntemlerine girdi olarak histogram matrisleri uygulanmıştır. Sırasıyla öncelikle TBA, daha sonra FDAY test edilmiş ve elde edilen sonuçlar çizelgelerle sunulmuş, bölüm sonlarında elde edilen sonuçlar irdelenmiştir.

5.2.1 TBA'nın test edilmesi ve elde edilen sonuçlar

TBA, histogram matrisleri girdi olarak uygulanmak kaydıyla, bilgisayar ortamında MATLAB programıyla test edilmiştir. Yapılan testler esnasında öncelikle tüm özdeğer/özvektör çiftleri kullanılmış, ardından boyut küçültme işlemi uygulanmıştır. Bunlara göre elde edilen sonuçlar şu şekildedir.

1. TBA'ya girdi olarak histogram bilgisi verildiğinde, öklid uzaklığı eşit ve kovaryans matrisinden elde edilen tüm özdeğer/özvektör çiftleri kullanılarak yapılan testlerde elde edilen sonuçlar:

Çizelge 5.10 : TBA'da histogram denemesi için öklid uzaklığı eşit, tüm özdeğer /özvektör çifti kullanıldığında elde edilen sonuçlar.

Sağ Kulak ($\varepsilon = \varepsilon_k = 132$)		Sol Kulak ($\varepsilon = \varepsilon_k = 136$)	
Örnek Sayısı	Tanıma Oranı (%)	Örnek Sayısı	Tanıma Oranı (%)
2	62	2	58
3	62	3	60
4	64	4	66
5	68	5	70

2. TBA'ya girdi olarak histogram bilgisi verildiğinde, öklid uzaklıkları farklı ve kovaryans matrisinden elde edilen tüm özdeğer/özvektör çiftleri kullanılarak yapılan testlerde elde edilen sonuçlar:

Çizelge 5.11 : TBA'da histogram denemesi için öklid uzaklıkları farklı, tüm özdeğer /özvektör çifti kullanıldığında elde edilen sonuçlar.

Sağ Kulak (ε_k = 110; ε =132)		Sol Kulak (ε_k =120; ε =136)	
Örnek Sayısı	Tanıma Oranı (%)	Örnek Sayısı	Tanıma Oranı (%)
2	64	2	60
3	66	3	64
4	66	4	70
5	74	5	72

3. TBA'da öklid uzaklıkları eşit; ancak kovaryans matrisinden elde edilen özdeğer/özvektör çiftlerinden ağırlığı yüksek olan, toplam adedin %99'una karşılık gelecek sayıda çiftin kullanılmak üzere seçilmesi ile yapılan histogramlı testlerde elde edilen sonuçlar:

Çizelge 5.12 : TBA'da histogram denemesi için öklid uzaklığı eşit, özdeğer/özvektör çiftinin %99'luk kısmı kullanıldığında sonuçlar.

Sağ Kulak (ε= ε_k =132)			Sol Kulak (ε = ε_k =136)		
Örnek Sayısı	M'/M	Tanıma Oranı (%)	Örnek Sayısı	M'/M	Tanıma Oranı (%)
2	30/40	56	2	30/40	58
3	41/60	60	3	41/60	60
4	51/80	66	4	51/80	62
5	59/100	66	5	60/100	66

4. TBA'da öklid uzaklıkları farklı ve kovaryans matrisinden elde edilen özdeğer/özvektör çiftlerinden ağırlığı yüksek olan, toplam adedin %99'una karşılık gelecek sayıda çiftin kullanılmak üzere seçilmesi ile yapılan histogramlı testlerde elde edilen sonuçlar:

Çizelge 5.13 : TBA'da histogram denemesi için öklid uzaklığı farklı, özdeğer /özvektör çiftinin %99'luk kısmı kullanıldığında sonuçlar.

Sağ Kulak (ε_k = 110; ε =132)			Sol Kulak (ε_k =120; ε =136)		
Örnek Sayısı	M'/M	Tanıma Oranı (%)	Örnek Sayısı	M'/M	Tanıma Oranı (%)
2	30/40	62	2	30/40	64
3	41/60	68	3	41/60	68
4	51/80	68	4	51/80	76
5	59/100	72	5	60/100	76

5. Histogramın TBA'ya girdi olarak verilmesiyle elde edilen sonuçlar da göstermiştir ki, öklid uzaklığının tanıma oranına etkisi çok yüksektir. Bu sonuça göre, tanıma yüzdesinin en yüksek olduğu öklid uzaklıkları ve bunlara ilişkin sonuçlar Çizelge 5.14'te sunulmuştur.

Çizelge 5.14 : TBA histogram uygulamasında maksimum tanıma oranı için elde edilen sonuçlar.

Sağ Kulak (ε_k = 105; ε =130)			Sol Kulak (ε_k = 120; ε =140)		
Örnek Sayısı	M'/M	Tanıma Oranı (%)	Örnek Sayısı	M'/M	Tanıma Oranı (%)
2	30/40	68	2	30/40	66
3	41/60	64	3	41/60	68
4	51/80	70	4	51/80	76
5	59/100	76	5	60/100	76

Yapılan denemeler neticesinde elde edilen sonuçlar, daha önceki denemelerle verılan sonuçlarla paralel ve bunları doğrular niteliktedir. Çizelgeler teker teker incelendiğinde, kullanılan özdeğer/özvektör çiftinin adedi azaldıkça tanıma oranı düşmekte; ancak bu negatif durum öklid uzaklıkları ile pozitife çevrilerek tanıma oranı arttırılabilmektedir.

5.2.2 FDAY'nin test edilmesi ve elde edilen sonuçlar

Her resme ait histogram bilgileri, TBA ile öncelikle işlemden geçirildikten sonra FDAY uygulanmıştır.

Yapılan testler esnasında elde edilen sonuçlar, histogram bilgilerine FDAY uygulamanın iyi sonuçlar vermediğini göstermiştir. Özellikle tüm özdeğer/özvektör çifti kullanıldığında tanıma oranları bir hayli düşmüştür. Bunun nedeni olarak da FDAY'de yapılan yansıma ve normalizelerin daha fazla olması ve bunların da veriyi birbirine iyice yaklaştırarak ayrımı zorlaştırması olarak değerlendirilebilir. Ancak gözlenen bir özellik olarak, kullanılan özdeğer/özvektör çiftinin adedi azaldığında tanıma oranı artmıştır. Tabi bu husus 4 ve 5 örnekli durum için geçerlidir. Buna göre, bahsedilen durumlar için elde edilen sonuç şu şekilde karşımıza çıkar.

Çizelge 5.15 : FDAY'nin histogram bilgilerine uygulanması ve sonuçlar.

Sağ Kulak				Sol Kulak			
Örnek Sayısı	M'/M	ε	Tanıma Oranı (%)	Örnek Sayısı	M'/M	ε	Tanıma Oranı (%)
4	48/80	50	66 **	4	48/80	45	78 *
5	60/100	40	78 **	5	60/100	50	74 **

Çizelge 5.15'te, '*' ile ifade edilen durumda 8 adet test resmi, eğitim aşamasında bilgileri alınmasına rağmen tanınamamıştır. Buna paralel olarak, '**' ile ifade edilen durumda ise 9 adet test resmi aynı şekilde tanınamamıştır.

Tüm yapılan deneme ve testlerden elde edilen sonuçlara göre, histogram bilgisine ile yapılan tanımanın,

1. Tam olarak doğru ve güvenilir bir şekilde yapılmasının güç olduğu,
2. Ancak bu yöntemin, ön bilgi sağlayabileceği,

değerlendirilmektedir.

6. GÖMÜLÜ SİSTEM GERÇEKLEME

Kulak biyometrisiyle tanıma sistemi tasarımı, herhangi bir bilgisayardan tamamen bağımsız, gömülü bir sistem olarak düşünülmüştür. Tüm işletim ve gücü kendi üzerinde sağlayabilecek bir kabiliyete sahip olmalıdır. Dolayısıyla bu bölümde, çalışmanın gerçeklenmesi ve tasarımında kullanılan malzemeler ve özellikleri, tasarımın aşamaları, irdelenecektir. Ayrıca Bölüm 4'te teorik olarak anlatılan nümerik problemler ve çözüm yöntemlerinin nasıl uygulandığına da değinilecektir.

Öncelikle, sistem tasarımında kullanılan elektronik karta ait özellikler detaylarıyla anlatılacak, ardından gerçek zamanlı bir tanımada gerekli olan kameraya ait bilgiler paylaşılacaktır. Sonrasında nümerik problemlerin çözümü irdelenerek sistemin gerçeklenmesi ve sonuçlar aktarılacaktır.

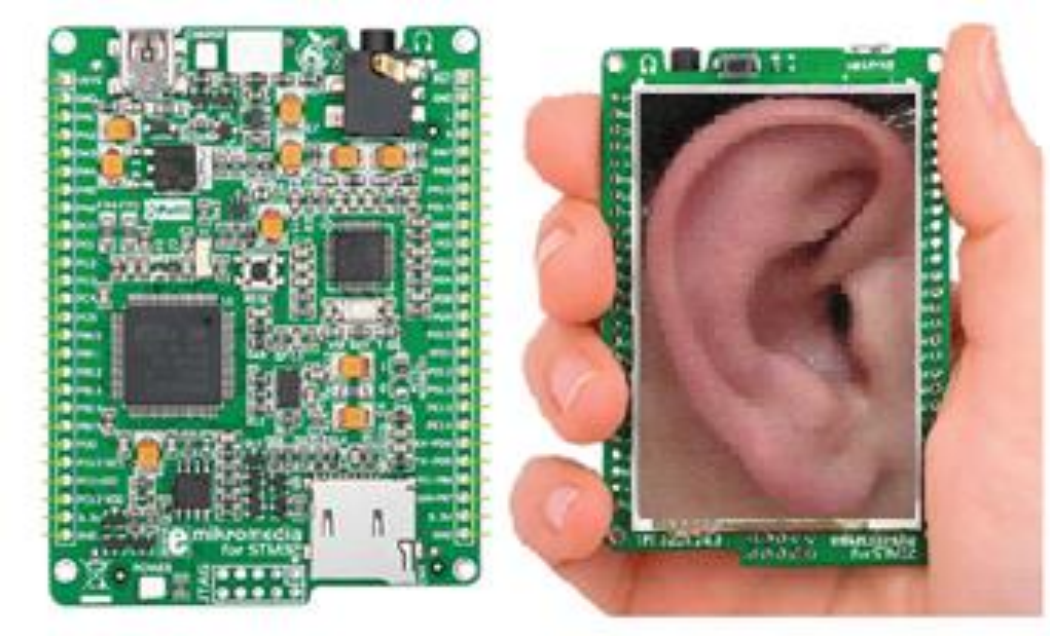
6.1 Sistem Tasarımında Kullanılan Elemanlar

Kulak biyometrisiyle gömülü sistem tasarımı için ihtiyacımız olan elemanlar amacımızı destekleyecek şekilde seçilmiştir. Yapılan araştırmalar neticesinde, "*MikroElektronika*" firmasına ait *Mikromedia for STM32 M4* kartı kullanılmıştır. STM32 M4 Şekil 6.1'de gösterilmiştir.

STM32 M4 mikrodenetleyici kartı, ARM tabanlı bir işlemci kartı olup bünyesinde Cortex M4 olarak bilinen ve ARM tarafından geliştirilen en yeni gömülü işlemciyi barındırır. Kart aslen, geliştirme ve deneme kartı olarak üretilmesine rağmen, ihtiyacımız olan özellikleri sağlamaktadır. Kartta bulunan özellikleri sıralayacak olursak;

1. Kartın merkezi birimi 32 bitlik Cortex M4 STM32F407VGT6 mikro denetleyicidir.
2. 320x240 boyutunda TFT dokunmatik ekrana sahiptir.
3. Üzerinde ivme ölçer mevcuttur.
4. Mikro SD kart okuyucu bulunmaktadır.
5. 168 MHz işlem hızına sahiptir.

6. 192 KByte SDRAM bulunur.
7. 1 MByte ROM hafıza bulunur.
8. Ayrıca MP3 kodeği sayesinde kulaklıkla müzik dinleme imkânı da sağlar.
9. UART, I^2C ve SPI protokolleriyle çalışabilme özelliğine sahiptir.



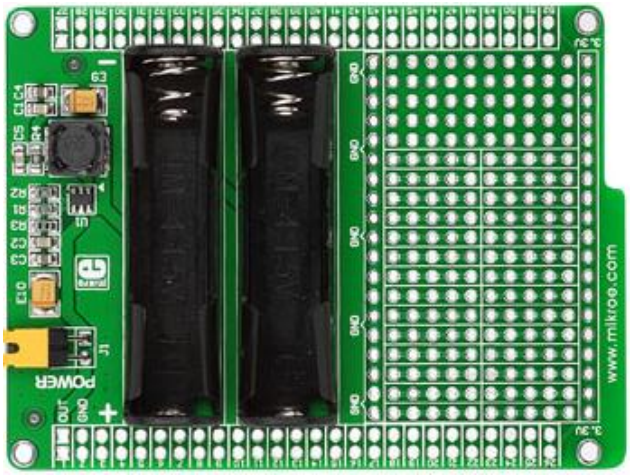
Şekil 6.1 : STM42 M4 mikrodenetleyici kartı.

STM32 M4 mikro denetleyici kartı, *MicroC Pro for ARM* olarak adlandırılan, C tabanlı bir derleyici dili kullanmaktadır. Derleyici içerisinde ihtiyaç duyulabilecek birçok kütüphane mevcuttur. Bu da programlama aşamasında gereken bazı matematiksel veya başka fonksiyonlardan çoğunun hazır olarak bulunmasını sağlamaktadır.

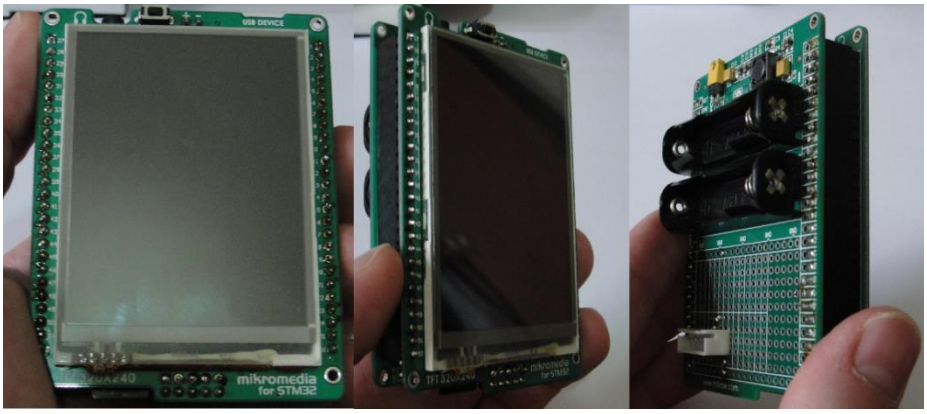
STM32 M4 gücünü, USB kablosu ile bilgisayardan almaktadır. Ancak düşük güç tüketimi sayesinde, arkasına donanım olarak takılabilen bir batarya kartında 2 adet AAA model pil ile de çalışabilmektedir. Batarya kartı Şekil 6.2'de olduğu gibidir. Batarya kartı monte edilmiş STM32 M4 ise Şekil 6.3'te gösterilmiştir.

Bununla birlikte sistem tasarımında kullanılan kamera, "*4D Systems*" firması tarafından üretilen ve *µCAM TTL* olarak da bilinen *µCAM Serial JPEG Camera Module* **'**dür. Kullanılan kamera Şekil 6.4'te gösterilmiştir.

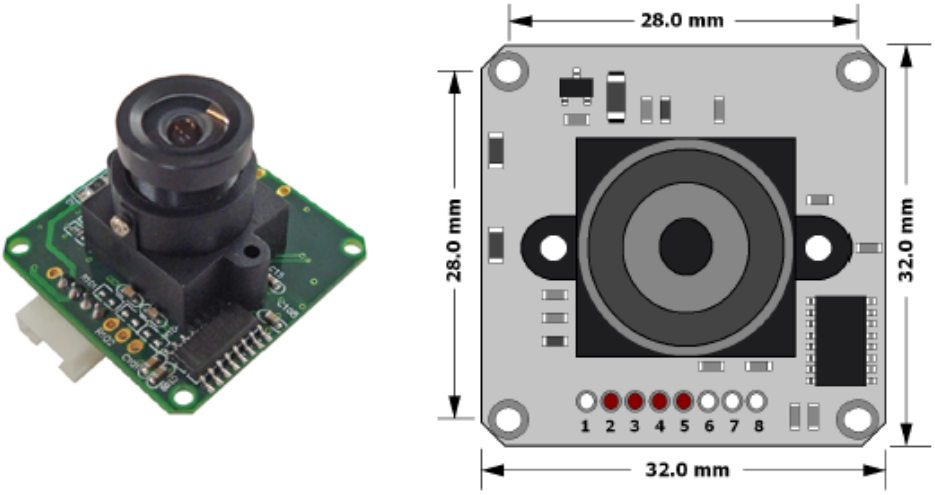
Kamera, düşük güç tüketimi ve üzerinde hazır bulunan seri arayüzler (TTL veya RS232) yardımıyla herhangi bir gömülü sistem veya elemana kolayca monte



Şekil 6.2 : Batarya kartı.



Şekil 6.3 : STM32 M4 ve batarya kartı.



Şekil 6.4 : µCAM Serial JPEG camera module.

edilebilecek bir yapıya sahiptir. Kamera, OmniVision CMOS VGA renkli bir sensörle donatılmıştır ve kullanıcı kodları basit bir seri protokol ile gönderilir.

Kameranın özellikleri şu şekilde sıralanabilir:

1. Düşük çözünürlüklü (160x120 veya 80x60) tek kare raw resim veya yüksek çözünürlüklü (640x480 veya 320x240) JPEG resimleri gösterebilir.
2. Güç ve veri alışverişi sadece 4 kablo bağlantılı bir konektör ile yapılabilir. Kamera UART protokolü ile bağlanır.
3. Sadece 3.3 V'luk bir DC kaynakla beslenebilmektedir ki bu, elimizdeki kart ile yapılabileceği anlamına gelir.
4. 2-bit, 4-bit ve 8-bit gri seviye ile 8-bit, 12-bit ve 16-bit RGB veya standart bir JPEG resim alınabilir.
5. Kamera modül, 14400bps, 56000bps, 57600bps ve 115200bps olmak üzere 4 adet baud-rate'le çalışma özelliğine sahiptir. Kullanıcı bunlardan birini tanımlamak zorundadır. Ancak başalngıçta modül otomatik olarak da baud-rate'ı tespit edebilir.

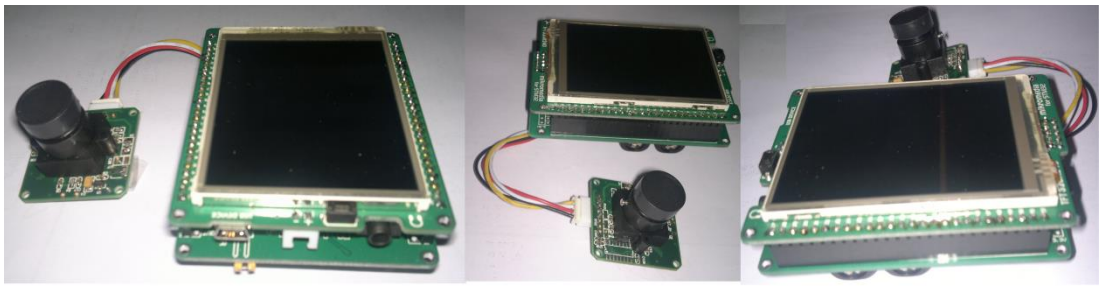
Mikro denetleyici kart ile kameranın iletişimi için UART protokolü kullanılmaktadır. Kameradan resimleri almadan önce, öncelikle iki elemanın senkronize edilmesi gerekir. Ardından başlangıç kodları ile kamera istenilen resmin özellikleri için hazırlanır.

Kamera modülü ayrıca bilgisayara da bağlanabilmektedir. Kamera bilgisayara bağlanarak, "4D µCAM Demo" isimli program ile tüm resim formatları ve kameranın düzgün çalıştığı test edilebilir. Bunun için *µUSB CE5* olarak ifade edilen seri köprü bağlantı aparatı kullanılmalıdır. Bu aparat Şekil 6.5'te gösterilmiştir.



Şekil 6.5 : µUSB CE5.

STM32 M4 ile µCAM TTL kamera modülün birlikte bağlanmış şekli Şekil 6.6'da olduğu gibidir.



Şekil 6.6 : STM32 M4 ve µCAM TTL.

6.2 Sistemin Oluşturulması ve Denenmesi

Kulak biyometrisiyle gömülü sistem tasarımı, öncelikle daha önce teorik olarak anlatılmış, ardından bilgisayar ortamında testleri yapılarak sonuçları elde edilmiş algoritma ve yöntemlerin mikro denetleyici kartta uygulanmasıyla gerçekleştirilmiştir. Bu bölümde, yapılan gerçeklemeyle alakalı detaylı bilgiler verilecek ve sistem irdelenecektir.

Bu bölüm iki alt başlık altında incelenecektir. Öncelikle sistemde eğitim ve test kümesi, mevcut hafızada kaydedilmiş ve test kendi içerisinde yapılmıştır. İlk alt başlıkta bu konu irdelenecektir. Ardından, kamera sisteme entegre edilmiş ve gerçek zamanlı bir tanıma gerçeklenmeye çalışılmıştır.

6.2.1 Sistemin kendi içerisinde gerçeklenmesi ve testler

STM32 M4 mikro denetleyici kart ile gerçekleştirilen testlerde, eğitim ve test kümesi sistemde mevcut hafıza bölümlerine kaydedilmiştir. Sistemin testi sırasında kullanılan eğitim ve test kümeleri, bilgisayar ortamında icra edilen testlerde kullanılan kümelerle aynıdır. Bunlar, sistemde mevcut olan 1MBytle'lık ROM hafıza içerisinde saklanmıştır. Resimler hafızanın ortalama %75'ini kaplar.

Sistem denemesi iki algoritma üzerinde yapılmıştır. Bunlar daha önceden teorik olarak da irdelenmiş olan TBA ve FDAY'dir. Ayrıca bu iki yöntem uygulanırken ortaya çıkan nümerik problemler de Bölüm 4'te anlatılan yöntem ve algoritmalarla çözümlenmiştir. Sistemdeki denemeler Bölüm 5'de izlenen sırayla yapılmıştır.

Elde edilen sonuçlar sırasıyla çizelgelerde detaylı olarak gösterilmiştir.

Çizelge 6.1 : Sistemin TBA ile denenmesi: öklit uzaklıkları eşit ve tüm özdeğer/özvektörler kullanılarak.

Sağ Kulak ($\varepsilon = \varepsilon_k = 1150$)		Sol Kulak ($\varepsilon = \varepsilon_k = 1250$)	
Örnek Sayısı	Tanıma Oranı (%)	Örnek Sayısı	Tanıma Oranı (%)
2	72	2	80
3	90	3	92
4	90	4	90
5	92	5	92

Çizelge 6.2 : Sistemin TBA ile denenmesi: öklit uzaklıkları farklı ve tüm özdeğer/özvektörler kullanılarak.

Sağ Kulak ($\varepsilon_k = 950$; $\varepsilon = 1150$)		Sol Kulak ($\varepsilon_k = 1150$; $\varepsilon = 1250$)	
Örnek Sayısı	Tanıma Oranı (%)	Örnek Sayısı	Tanıma Oranı (%)
2	76	2	84
3	88	3	92
4	94	4	92
5	94	5	94

Çizelge 6.3 : Sistemin TBA ile denemesi: öklit uzaklıkları eşit ve %99 sayıda özdeğer/özvektör kullanılarak.

Sağ Kulak ($\varepsilon = \varepsilon_k = 1150$)			Sol Kulak ($\varepsilon = \varepsilon_k = 1250$)		
Örnek Sayısı	M'/M	Tanıma Oranı (%)	Örnek Sayısı	M'/M	Tanıma Oranı (%)
2	28/40	64	2	29/40	78
3	39/60	78	3	39/60	86
4	48/80	86	4	48/80	88
5	54/100	86	5	57/100	90

Çizelge 6.4 : Sistemin TBA ile denemesi: öklit uzaklıkları farklı ve %99 sayıda özdeğer/özvektör kullanılarak.

Sağ Kulak (ε_k = 950; ε =1150)			Sol Kulak (ε_k =1150; ε =1250)		
Örnek Sayısı	M'/M	Tanıma Oranı (%)	Örnek Sayısı	M'/M	Tanıma Oranı (%)
2	28/40	70	2	29/40	82
3	39/60	82	3	39/60	88
4	48/80	92	4	48/80	92
5	54/100	94	5	57/100	92

Çizelge 6.5 : Sistemin TBA ile denemesi: maksimum tanıma oranı için öklit uzaklıkları ve %99 sayıda özdeğer/özvektör kullanılarak.

Sağ Kulak (ε_k = 950; ε =1500)			Sol Kulak (ε_k = 1200; ε =1350)		
Örnek Sayısı	M'/M	Tanıma Oranı (%)	Örnek Sayısı	M'/M	Tanıma Oranı (%)
2	28/40	86	2	29/40	80
3	39/60	90	3	39/60	90
4	48/80	92	4	48/80	92
5	54/100	96	5	57/100	92

Çizelge 6.6 : Sistemin FDAY ile denenmesi: tüm özdeğer/özvektörler kullanılarak.

Sağ Kulak			Sol Kulak		
Örnek Sayısı	Tanıma Oranı (%)	ε	Örnek Sayısı	Tanıma Oranı (%)	ε
2	72	565	2	68	600
3	80	475	3	92	390
4^+	98	430	4^+	98	370
5^+	96	300	5^+	92	300

Çizelge 6.7 : Sistemin FDAY ile denenmesi: aynı öklid uzaklıkları için %99 sayıda özdeğer/özvektör kullanılarak.

Sağ Kulak				Sol Kulak			
Örnek Sayısı	M'/M	Tanıma Oranı (%)	ε	Örnek Sayısı	M'/M	Tanıma Oranı (%)	ε
2	28/40	80	565****	2	29/40	72	600***
3	39/60	94	475**	3	39/60	94	390***
4	48/80	96	430**	4	48/80	96	370**
5	54/100	96	300**	5	57/100	92	300***

Çizelge 6.8 : Sistemin FDAY ile denenmesi: öklid uzaklıklarında revize ve tüm özdeğer/özvektörler kullanılarak.

Sağ Kulak			Sol Kulak		
Örnek Sayısı	Tanıma Oranı (%)	ε	Örnek Sayısı	Tanıma Oranı (%)	ε
2	72	565	2	68	600
3	90	325§	3	92	390
4+	96	300§	4+	98	370
5+	96	300	5+	92	300

Çizelge 6.9 : Sistemin FDAY ile denenmesi: öklid uzaklıklarında revize ve %99 sayıda özdeğer/özvektör kullanılarak.

Sağ Kulak				Sol Kulak			
Örnek Sayısı	M'/M	Tanıma Oranı (%)	ε	Örnek Sayısı	M'/M	Tanıma Oranı (%)	ε
2	28/40	84	560‡‡	2	29/40	74	530‡‡‡
3	39/60	94	603‡	3	39/60	98	500‡
4	48/80	100	505	4	48/80	98	425‡
5	54/100	100	420‡	5	57/100	96	360‡‡

Yukarıdaki çizelgeler incelendiğinde, bilgisayarla yapılan denemelerle paralele sonuçlar elde edilmiştir. Ancak, bilgisayardaki denemelerden farklı olarak, FDAY denemelerinde Çizelge 6.6 ve Çizelge 6.8'de "+" ile gösterilen testler hafızanın

yeterli olmamasından dolayı 60 özdeğer/özvektör çifti kullanılarak yapılmıştır. Buna göre tanıma oranları az miktarda düşmüştür.

Çizelge 6.7'ye göre "**" ile gösterilen durumlarda 2'şer; "***" ile gösterilen durumlarda 3'er; "****" ile gösterilen durumlarda ise 4'er adet test resmi, Çizelge 6.8'e göre "§" ile gösterilen durumlarda 2'şer adet test resmi, Çizelge 6.9'a göre ise "‡" 1'er; "‡‡" 2'şer, "‡‡‡" 4'er adet test resmi eğitim kümesinde olmasına rağmen tanınamamıştır.

6.2.2 Kameralı gerçekleme ve testler

Kulak biyometrisi ile tanıma sistemi, gerçek zamanlı bir gerçekleme yapılmak maksadıyla, daha önce anlatılan kamera modül mikro kontrolör kartı ile entegre edilmiştir. Eğitim kümesi daha önceden belirtildiği üzere hafızada saklanmış, test resimleri ise kameradan alınmıştır.

Kameradan alınan resimlerin formatları 80x60 boyutlarında .raw formatındadır. Kameradan alınan resimler, sistem içerisindeki testlerde olduğu gibi işleme sokularak test işlemi gerçekleştirilmiştir.

Daha önce yapılan testlerden farklı olarak kameralı gerçekleme her sınıf için 5 örnek koşulu ile yapılmıştır. Bunun haricinde, tarafımdan çekilen farklı kişilere ait resimler, eğitim kümesine eklenmiştir. Bu denemede amaç, eğitim kümesine katılan yeni 2 sınıfı hem kendi içinde hem de kamera ile test etmektir.

Yapılan testler sonucunda, kameralı entegre sistemimizin tanıma oranı %96 seviyesine kadar çıkmıştır. Ayrıca yeni eklenen iki sınıfa ait kişiler de doğru olarak tanınmıştır.

7. SONUÇ

Son yirmi yılda gelişen tekolojiyle ortaya çıkan güvenlik gereksinimleri, insanların biyometrik ayırıcılarıyla tanınması ve bunların gerçeklenmesine olanak sağlamıştır. Bu çalışmadaki amacımız, insanların biyometri ile tanınmasının yanı sıra gerçeklenecek sistemin tamamen bilgisayardan bağımsız ve taşınabilir olmasıdır. Yapılan araştırma ve çalışmalar sonucunda, istenilen özelliklerde bir sistem gerçeklenmiştir. Sistem gerçeklenmesinde kullanılan yöntemlerden TBA ile % 92; FDAY ile %98'lere ulaşan bir doğruluk elde edilmiştir. Yapılan çalışma neticesinde,

1. Kulak biyometrisinin, tanıma sistemleri için kullanılabilecek ayırıcı bir özellik olduğu,
2. Uygulamanın yüz, parmak izi, iris, vb. yöntemlere oranla daha basit olduğu,
3. Özdeğer/özvektör sayısının ve öklit uzaklıklarının tanıma oranlarına doğrudan etki ettiği,
4. TBA algoritmasında elde edilen sonuçların, herhangi bir tanıma sistemi için yeterli olduğu ve özellikle diğer biyometrik ayraçlarla elde edilen tanıma oranlarındandan yüksek olduğu,
5. FDAY algoritmasında tamamına yakın bir oranda tanıma gerçekleştirildiği,
6. Yapılan sistemin kendi kendine çalışabilmesi, hiçbir işletim sistemine bağımlı olmaması ve taşınabilir olması nedeniyle, birçok güvenlik alanında kullanılabilir olduğu, işletiminin zahmetsiz ve küçük meblağlara mâl olması sebebiyle yakın gelecekte tercih edilebilir olduğu,

değerlendirilmektedir.

Resimlerle yapılan tüm bu çalışmalardan farklı olarak, Bölüm 5'te de incelendiği üzere histogramlar ile denemeler de icra edilmiştir. Eğitim kümesine ait resimlerin histogram bilgileri, her iki algoritma için giriş verisi olarak uygulanmış ve bu bilgilerle test resimlerine ait histogramlar tanınmaya çalışılmıştır. Elde edilen sonuçlar göstermiştir ki,

1. Histogram bilgileri yeteri kadar kararlı sonuçlar üretememiş,

2. Elde edilen tanıma oranları çok düşük kalmıştır.

Elde edilen oranlardan da anlaşılacağı üzere, histogramın biyometrik sistemler için giriş verisi olarak kullanılmasının doğruluk ve kararlılık açısından uygun olmayacağı değerlendirilmektedir.

Bu çalışmanın amacı, günümüzde yaygın olarak çalışılan yüz tanımaya alternatif olarak kararlı bir biyometrinin öne çıkarılması ve bununla insanların tanınmasının sağlanabileceğini göstermektir. Yukarıda maddelerle sıralanan denemeler ve çalışma sonuçları incelendiğinde, bu çalışmanın amacına ulaştığı, bununla beraber kulak biyometrisinin yüz tanımaya oranla daha doğru ve kararlı sonuçlar verdiği gösterilmiştir.

KAYNAKLAR

Abaza, A., Ross, A., Hebert , C., Harrison ,M.A.F., Nixon ,M.S. (2010). A Survey on Ear Biometrics, ACM Transactions on Embedded Computing Systems, Vol. 9, No. 4, Article 39**.**

Ağır, K., (2006). Null Space Approach of Fisher Discriminant Analysis for Face Recogition, İstanbul Teknik Üniversitesi, M.Sc. Thesis.

Balakrishnama, S., Ganapathiraju, A., Linear Discriminant Analysis – A Brief Tutorial**,** Institude for Signal Processing and Information Processing, Department of Electrical and Computer Engineering, Mississippi State University.

Belhumeur P., Hespanha J. and Kriegman D., (1997). Eigenfaces vs. Fisherfaces: recognition using class specific linear projection, *IEEE Trans. On Pattern Analysis and Machine Intelligence,* vol. 19, pp. 711-720.

Burge, M., Burger, W., (2000). Ear Biometrics in Computer Vision, 15th International Conference on Pattern Recognition,Vol.2, p.822-826.

Chang, K., Bowyer, K.W., Sarkar, S., Victor, B., (2003). Comparison and Combination of Ear and Face Images in Appearance-Based Biometrics, IEEE Trans. On Pattern Analysis and Machine Intelligence, Vol. 25, no.9, p.1160-1165.

Chen, L. F., Liao H. Y. M., Ko M. T., Lin J. C. and Yu G. J. (2000). A New LDA-Based Face Recognition System Which Can Solve the Small Sample Size Problem, *Pattern Recognition,* vol. 33, pp. 1713-1726.

Duda, R. O., Hart, P. E., Stork, D. G., (2001). Pattern Classification, John Wiley & Sons, Inc., New York.

Fukunaga, K., (1990). Introduction to Statistical Pattern Recognition, Boston: Academic Press.

Iannarelli, A., (1989). Ear Identification, Paramount Publishing Company, Freemont, California.

Kocaman, B., Kırcı, M., Güneş, E.O., Çakır, Y., Özbudak, Ö., (2009). On Ear Biometrics, Eurocon 2009, pp. 327-332.

Mathews, J.H., Fink, K.D., (1999). Numerical Methods Using MATLAB, Third Edition, Prentice Hall.

Shailaja, D.N., (2006). A Simple Geometric Approach to Ear Recogniton**,** Departments of Computer Science and Engineering, Indian Institute of Technology, Bc. Thesis.

Shlens, J., (2005). A Tutorial on Principal Component Analysis Version 2, Systems Neurobiology Laboratory, Salk Insitute for Biological Studies and Institute for Nonlinear Science, University of California.

Smith, L.I., (2002). A Tutorial on Principal Component Analysis.

Stoilov, T., Stoilova, K., (2004). Algorithm and Software Implementation of QR Decomposition of Rectangular Matrices, CompSysTech'2004, International Conference on Computer Systems and Technologies.

Turk, M., Pentland, A., (1991). Eigenfaces for Recognition, Journal of Cognitive Neuroscience, Vol 3, No. 1. 71-86.

Url-1< http://openbio.sourceforge.net/ resources/eigenfaces/> Giriş tarihi: 29 Eylül 2012.

Url-2<http://college.cengage.com/mathematics/larson/elementary_linear/4e/shared/downloads/c10s2.pdf > Giriş tarihi: 4 Şubat 2013.

Url-3< http://www.sci.wsu.edu/math/faculty/watkins/pdfiles/uqr2.pdf > Giriş tarihi: 4 Şubat 2013.

Url-4<www.inf.ethz.ch/personal/gander/papers/qrneu.pdf > Giriş tarihi: 2 Şubat 2013.

Url-5<www.math.ucla.edu/~yanovsky/Teaching/Math151B/handouts/GramShmidt.pdf > Giriş tarihi: 2 Şubat 2013.

Url-6< http://people.math.gatech.edu/~klounici6/2605/Lectures%20notes%2Carlen/chap4.pdf > Giriş tarihi: 4 Şubat 2013.

Url-7<http://mathforcollege.com/nm/mtl/com/04sle/mtl_com_sle_ppt_ludecomp.pdf > Giriş tarihi: 12 Şubat 2013.

Url-8<matpc41.ee.tut.fi/ML02/Matrix%20Algebra/Lecture%20Notes/chapter2.pdf > Giriş tarihi: 5 Şubat 2013.

Url-9<http://www.mikroe.com/downloads/get/1753/mikromedia_stellaris_m3_manual_v101b.pdf > Giriş tarihi: 28 Eylül 2012.

Victor, B., Bowyer, K.W., Sarkar, S., (2002). An Evaluation of Face and Ear Biometrics, Proc. International Conference on Pattern Recognition**,** p.429-432.

Wang, L., Ji, H., Shi, Y., (2011). Face Recognition Using Maximum Local Fisher Dscriminant Analysis, 18th IEEE International Conference on Image Processing, p.1737-1740.

Webb, A., Copsey, K., (2011). Statistical Pattern Rcognition – Third Edition, John Wiley & Sons.

ÖZGEÇMİŞ



Ad Soyad: Murat KUŞ

Doğum Yeri ve Tarihi: Gölcük 1984

Adres: Denizevler Mah. Adalet Cad. No:35 K:2 Gölcük/KOCAELİ

E-Posta: pikeman2334@gmail.com

Lisans: Deniz Harp Okulu

Yüksek Lisans: İstanbul Teknik Üniversitesi

Mesleki Deneyim ve Ödüller: 2006-2011 Deniz Kuvvetleri K.Lığı Muvazzaf Subay

Yayın ve Patent Listesi: ARM Based Ear Recognition Embedded System (Henüz Yayınlanmadı)

An Embedded Biometric System
(Henüz Yayınlanmadı)

TEZDEN TÜRETİLEN YAYINLAR/SUNUMLAR

▪ **Kuş M.**, Kaçar Ü., Kırcı M., Güneş E.Olcay, 2013: ARM Based Ear Recognition Embedded System (Henüz Yayınlanmadı). EUROCON ‘13, Temmuz 1-4, 2013 Zagrep, Hırvatistan.

▪ Kaçar Ü., **Kuş M.**, Kırcı M., Güneş E.Olcay, 2013: An Embedded Biometric System (Henüz Yayınlanmadı). 16th International Conference on Information, Fusion ‘13, Temmuz 9-12, 2013 İstanbul, Türkiye.