

**T.C.
HALIÇ ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ
BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI
BİLGİSAYAR MÜHENDİSLİĞİ PROGRAMI**

**BULANIK KÜMELEMELİ YAPAY SİNİR AĞLARI İLE
BİYOMETRİK TANIMA**

YÜKSEK LİSANS TEZİ

**Hazırlayan
Hilmi ARIÇ**

**Danışmanı
Prof.Dr. Mübariz EMİNLİ**

İstanbul – 2011

T.C.
HALIÇ ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ MÜDÜRLÜĞÜNE


Bilgisayar Mühendisliği Anabilim Dalı Bilgisayar Mühendisliği Programı Tezli Yüksek Lisans öğrencisi **Hilmi ARIÇ** tarafından hazırlanan “**Bulanık Kümelemeli Yapay Sinir Ağları ile Biyometrik Tanıma**” adlı bu çalışma jürimizce Yüksek Lisans Tezi olarak kabul edilmiştir.

Sınav Tarihi : 21.06.2011

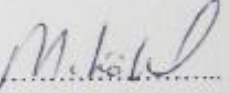
(Jüri Üyesinin Ünvanı , Adı , Soyadı ve Kurumu) :

İmzası :

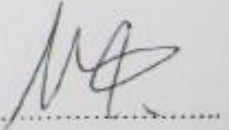
Jüri Üyesi: Prof.Dr.Mübariz EMİNLİ
Dan.-HAL.Üniv.Bilgisayar Müh. ABD Öğr.Üyesi



Jüri Üyesi : Prof.Dr.Muhammet KÖKSAL
HAL.Üniv. Bilgisayar Müh.ABD Öğr.Üyesi



Jüri Üyesi : Yrd.Doç.Dr.Murat BEKEN
HAL.Üniv. Matematik Müh.ABD Öğr.Üyesi



Jüri Üyesi : Yrd.Doç.Dr.Ulviye HACIYEVA
HAL.Üniv. Bilgisayar Müh.ABD Öğr.Üyesi (Yedek)

.....

Jüri Üyesi : Yrd.Doç.Dr.Oğuz KARAN
HAL.Üniv. Bilgisayar Müh.ABD Öğr.Üyesi (Yedek)

.....

İÇİNDEKİLER

KISALTMALAR	IV
1.1 İngilizce Kısaltmalar	IV
1.2 Türkçe Kısaltmalar	IV
1.3 Türkçe - İngilizce Sözlük.....	V
1.4 İngilizce - Türkçe Sözlük.....	V
ŞEKİLLER LİSTESİ	VI
TABLOLAR LİSTESİ	VIII
ÖNSÖZ	IX
ÖZET	X
ABSTRACT	XII
1 GİRİŞ	1
1.1 Tezin Amacı.....	4
1.2 Hedefler ve Tezin Başarı Kriterleri	5
2 BİYOMETRİK TANIMA	7
2.1 Yüz tanıma.....	9
2.2 Parmak izi tanıma.....	10
2.3 Yürüyüş tanıma	11
2.4 El ve Damar Tanıma.....	12
2.5 Retina tanıma	12
2.6 İris Tanıma.....	13
2.7 Biyometrik Sistemlerin Karşılaştırılması.....	17
3 ÖZ NİTELİKLERİN ÇIKARILMASI	20
3.1 Ayırık Kosinüs Dönüşümü	20
3.2 Dalgacık (Wavelet) Dönüşümü.....	24
4 YAPAY SİNİR AĞLARI	27
4.1 YSA'nın Tarihsel Gelişimi	28
4.2 Biyolojik Sinir Ağları ve Nöronun Biyolojik Yapısı.....	30
4.3 YSA'nın Yapısı ve İşlem Elemanı	32

4.4 Öğrenme Kuralları.....	33
4.4.1 Perceptron (İdrak-algılayıcı)	33
4.4.2 Perceptron Öğrenme Algoritması	36
4.4.3 Çok Katmanlı İdrak.....	38
4.4.4 Hatanın Geriye Yayılması Algoritması ve Genelleştirilmiş Delta Kuralı	40
4.4.5 Öğrenme ve Momentum Sabitleri	45
5 BULANIK KÜMELEME	48
5.1 Bulanık Mantığa Giriş	48
5.2 Bulanık Kümeleme (BK) Yaklaşımı	49
5.3 Bulanık Kümeler.....	50
5.4 Bulanık Kümelerin Gösterimi.....	51
5.5 Üyelik Fonksiyonları	53
5.6 Bulanık C-Ortalamalar Algoritması	54
6 BULANIK KÜMELEMELİ YAPAY SİNİR AĞLARI	58
6.1 BKYSA'nın İris Tanımda Uygulanması.....	58
6.1.1 Ön işlemler	60
6.1.2 Öznitelik Vektörü Çıkarımı	63
6.1.3 Öznitelik Vektörüne BCO Uygulanması	65
6.1.4 Sınıflandırma	68
7 TEST.....	71
7.1 ROC.....	71
7.2 DCT-YSA Test Sonuçları.....	73
7.3 DWT –YSA Test Sonuçları	75
7.4 DCT-BKYSA Test Sonuçları	78
7.5 DWT-BKYSA Test Sonuçları	81
7.6 Genel Test Sonuçları	85
8 SONUÇ.....	88
9 KAYNAKLAR.....	91
10 EKLER.....	97
EK-1: Kaynak Kodu.....	97
EK-2: Kullanıcı Arayüzü.....	103

KISALTMALAR

1.1 İngilizce Kısaltmalar

Kısaltma	Anlamı
FAR or FMR	False accept rate or false match rate - Yanlış Tanıma Oranı
FRR or FNMR	False reject rate or false non-match rate - Yanlış Reddetme Oranı
FCM	Fuzzy c-means - Bulanık C-Ortalamlar
ROC	Receiver Operating Characteristic
AUC	Area Under Curve
DCT	Discrete Cosine Transform – Ayrık Kosinüs Dönüşümü
DWT	Discrete Wavelet Transform – Ayrık Dalgacık Dönüşümü
MLP	Multi-Layered Perceptron - Çok Katmanlı İdrak

1.2 Türkçe Kısaltmalar

Türkçe Kısaltma	Anlamı
BKYSA	Bulanık Kümelemeli Yapay Sinir Ağları
YSA	Yapay Sinir Ağları
BK	Bulanık kümeleme
BCO	Bulanık C-Ortalamlar
EKG	Elektrokardiyogram
HD	Hough dönüşümü
YTO	Yanlış Tanıma Oranı
YRO	Yanlış Reddetme Oranı
DTO	Doğru Tanıma Oranı
ROC	Alıcı işletim karakteristiği

1.3 Türkçe - İngilizce Sözlük

Tezde Kullanılan Türkçe İfade	İngilizce Karşılığı
Benzetim	Simulation
İterasyon	Iteration
Normalleştirme	Normalization

1.4 İngilizce - Türkçe Sözlük

Tezde Kullanılan İngilizce İfade	Türkçe Karşılığı
Cutoff Value	Eşik Değeri
Positives	Pozitif kümedeki eleman sayısı (P). Sisteme tanıtılan elemanlar.
Negatives	Negatif kümedeki eleman sayısı (N).
True Positives	Pozitif küme içindeki örneğin Pozitif küme içinde olarak tanınması (TP).
True Negatives	Negatif küme içindeki örneğin Negatif küme içinde olarak tanınması (TN).
False Positives	Negatif küme içindeki örneğin Pozitif küme içinde olarak tanınması (FP).
False Negatives	Pozitif küme içindeki örneğin Negatif küme içinde olarak tanınması (FN).
Specificity	$SPC = TN / N = TN / (FP + TN) = 1 - FPR$
Sensitivity	$TPR = TP / P = TP / (TP + FN)$
Efficiency	Verimlilik.
Accuracy	Doğruluk oranı. $(ACC = (TP + TN) / (P + N))$
Positive Predictive Value	Hassasiyet, $TPR = TP / P = TP / (TP + FN)$
Negative Predictive Value	Hassasiyet, $NPV = TN / (TN + FN)$
False Positive Rate	$FPR = FP / N = FP / (FP + TN)$

ŞEKİLLER LİSTESİ

Şekil 1.1. Öğrenme işlemleri nesne tanıma sisteminin blok diyagramı (Karlık, B., 2009 a)	1
Şekil 1.2. İris veritabanından bir örnek	2
Şekil 2.1. Biyometrik sistem blok diyagramı (Biyometrik Diagram, 2011)	8
Şekil 2.2. Yüz tanıma	9
Şekil 2.3. Parmak izi tanıma (Vizyotek, 2011)	10
Şekil 2.4. Damar tanıma	12
Şekil 2.5. Göz anatomisi (İris, 2011)	13
Şekil 2.6. Schiphol havaalanında iris tanıma uygulaması (Daugman, J.)	16
Şekil 2.7. Bir nesne tanıma sisteminin genel blok diyagramı	19
Şekil 3.1. DCT ile özellik çıkarma ve resim sıkıştırma	23
Şekil 3.2. DCT ile özellik çıkarma ve Histogram Eşitleme a) Orjinal parçanın küçültülmüş hali, b) a'nın DCT uygulanmış hali, c) a'nın histogram eşitlenmiş hali, d) c'nin DCT uygulanmış hali	23
Şekil 3.3. Ayrık dalgacık dönüşümünün gerçekleştirilmesinde alt bantlara ayrışım; $g[n]$ yüksek geçiren filtre, $h[n]$ alçak geçiren filtre (Güler, İ. ve ark, 2004)	25
Şekil 4.1. Perceptron modeli	29
Şekil 4.2. Biyolojik nöron/sinir hücrelerinin şematik yapısı	30
Şekil 4.3. (a) Tek katmanlı perceptron yapısı, (b) Bir analog giriş vektörünü A ve B ile gösterilen iki sınıfa ayıran tek katmanlı bir perceptron	34
Şekil 4.4. Lineer ayrılabilirliğin gösterimi	34
Şekil 4.5. Tek ve çok katmanlı perceptronların sınırlayabileceği karar bölgeleri	35
Şekil 4.6. (n:m:p) mimarisine sahip bir çok katmanlı perceptronun yapısı	39
Şekil 4.7. Hatanın geriye yayılması algoritmasının blok diyagramı	40
Şekil 4.8. Gizli katmanı olmayan ağırlıklı hata fonksiyonu	42
Şekil 4.9. Gizli katmana ait ağırlıklı hata fonksiyonu	42
Şekil 4.10. Hatanın Geriye Yansıması Algoritmasının akış diyagramı (Karlık, B. 2009)	47
Şekil 5.1. Yaygın olarak kullanılan üyelik fonksiyonları (a) Yamuk Şekilli (b) Üçgen Şekilli (c) Gauss Üyelik Fonksiyonu (d) Çan Şekilli Not: Tüm üyelik fonksiyonları, sürekli, normal ve konvektir.	52
Şekil 5.2. Bulanık Kümeler: Yaşın üç gruba ayrıldığı bulanık kümeleme	53
Şekil 5.3. Yedi ayrı etiketli üyelik (Karlık, B., 2009 c)	54
Şekil 6.1. BKYSYA blok diyagramı	58

Şekil 6.2. BKYSA ile biyometrik tanıma akış diyagramı (DCT)	60
Şekil 6.3. İrisin seçilen kısmı.....	61
Şekil 6.4. Histogram Eşitleme a) Orjinal parçanın küçültülmüş hali, b) a'nın histogram eşitlenmiş hali	63
Şekil 6.5. Özellik Çıkarma adımlarındaki iris görüntüleri.....	63
Şekil 7.1. DCT-YSA – Test sonuçları ve hatanın kareleri toplamı (2 iris eğitim – 1 iris test)	74
Şekil 7.2. DCT-YSA ROC Eğrisi (2 iris eğitim – 1 iris test)	74
Şekil 7.3. DCT-YSA ROC Eğrisi (1 iris eğitim – 2 iris test)	75
Şekil 7.4. ROC Analiz Sonuçları (2 iris eğitim – 1 iris test)	75
Şekil 7.5. DWT-YSA – Test sonuçları ve hatanın kareleri toplamı (2 iris eğitim – 1 iris test)	76
Şekil 7.6. DWT-YSA için ROC Grafiği (2 iris eğitim 1 iris test).....	77
Şekil 7.7. DWT-YSA için ROC Grafiği (1 iris eğitim 2 iris test).....	77
Şekil 7.8. DCT-BKYSA – Test sonuçları ve hatanın kareleri toplamı (2 iris eğitim 1 iris test, 3 x 3 pencerelemeli)	79
Şekil 7.9. DCT-BKYSA – Test sonuçları ve hatanın kareleri toplamı (2 iris eğitim 1 iris test, 4 x 4 pencereli)	79
Şekil 7.10. ROC Grafiği (2 iris eğitim 1 iris test için, 3 x 3 pencereli)	80
Şekil 7.11. ROC Grafiği (2 iris eğitim 1 iris test için, 4 x 4 pencereli)	80
Şekil 7.12. DWT-BKYSA – Test sonuçları ve hatanın kareleri toplamı (2 iris eğitim 1 iris test için, 3x3 pencereli).....	82
Şekil 7.13. DWT-BKYSA – Test sonuçları ve hatanın kareleri toplamı (2 iris eğitim 1 iris test için, 4x4 pencereli).....	82
Şekil 7.14. DWT-BKYSA Hatanın kareleri toplamı ve test sonuçları (1 iris eğitim 2 iris test için, 4x4 pencereli)	83
Şekil 7.15. Roc Grafiği (2 iris eğitim 1 iris test için, 3x3 pencerelemeli)	84
Şekil 7.16. Roc Grafiği (2 iris eğitim 1 iris test için, 4x4 pencerelemeli)	84
Şekil 7.17. Roc Grafiği (2 iris eğitim 1 iris test için, 5x5 pencerelemeli)	85
Şekil 10.1. Resim İşleme	103
Şekil 10.2. Ön işlemler ve Öz nitelik vektörünün çıkarımı	103
Şekil 10.3. Matlab ile iki boyutlu verinin tek boyuta indirgenmesi	103
Şekil 10.4. Matlab ile elde edilen özellik vektörüne normalize eden program	104
Şekil 10.5. BCO Uygulması	104
Şekil 10.6. YSA ekranı	105
Şekil 10.7. ROC Analizi Uygulaması 1	106
Şekil 10.8. ROC Analizi Uygulaması 2	106

TABLULAR LİSTESİ

Tablo 2.1. Biyometrik Sistemlerin Karşılaştırılması.....	17
Tablo 2.2. Biyometrik sistem pazar oranları (Karlık, B. 2009).....	18
Tablo 6.1. Öznitelik vektörlerinin çıkarılması.....	64
Tablo 6.2. YSA Ağ Yapısı.....	69
Tablo 7.1. ROC analizi	72
Tablo 7.2. DCT-YSA test sonuçları	73
Tablo 7.3. DWT-YSA test sonuçları.....	76
Tablo 7.4. DCT-BKYSA test sonuçları	78
Tablo 7.5. DWT-BKYSA test sonuçları	81
Tablo 7.6. Daha önce araştırmacılar tarafından geliştirilmiş iris tanıma sistemleri ve tanıma performans sonuçları (YTO – Yanlış tanıma oranı / YRO – Yanlış reddetme oranı / DTO - Doğru tanıma oranı) (Koçer, H. E., 2007)	86
Tablo 7.7. Genel test sonuçları (Klasik YSA)	87
Tablo 7.8. Genel test sonuçları (BKYSAs)	87

ÖNSÖZ

Bu tez çalışması boyunca yardımlarını esirgemeyen değerli hocalarım Prof. Dr. Mübariz Eminli ve Prof. Dr. Bekir Karlık'a ve çalışmalarından faydalandığım Yrd. Doç. Dr. Erdiñ Koçer'e sonsuz teşekkürlerimi iletiyorum.

Ayrıca, bu tez çalışması süresince bana destek olan eşim Sibel Arıç'a da minnettarlığımı burada ifade etmek istiyorum.

GENEL BİLGİLER

Adı ve Soyadı : Hilmi ARIÇ
Anabilim Dalı : Fen Bilimleri Enstitüsü
Programı : Bilgisayar Mühendisliği
Tez Danışmanı : Prof. Dr. Mübariz EMİNLİ
Tez Türü ve Tarihi : Yüksek Lisans – Haziran 2011

Jüri :

Prof. Dr. Mübariz EMİNLİ

Prof. Dr. Muhammet KÖKSAL

Yrd.Doç.Dr. Murat BEKEN

BULANIK KÜMELEMELİ YAPAY SİNİR AĞLARI İLE BİYOMETRİK TANIMA

ÖZET

Bu çalışmada Bulanık Kümelemeli Yapay Sinir Ağlarının (BKYSA) iki boyutlu bir örüntü tanıma problemi üzerine uygulanışı gerçekleştirilmiştir. Yapılan çalışma birçok biyometrik üzerine uygulanabilir şekilde geliştirilmiştir. Bu çalışma için biyometriklerden biri olan iris seçilmiş ve iris tanıma üzerine çalışma yapılmıştır. Öz nitelik vektörü elde edilirken Ayrık Dalgacık Dönüşümü (DWT) ve Ayrık Kosinüs Dönüşümü (DCT) yöntemlerinden yararlanılmıştır. Bulanık kümeleme (BK) algoritmalarından bir tanesi olan Bulanık C Ortalamalar (BCO) ile öznitelik vektörü üzerinde kümeleme yapılmış ve Yapay Sinir Ağları (YSA) ile de sınıflandırma gerçekleştirilmiştir. Çalışma yapılırken öncelikli olarak klasik YSA ile sınıflandırma gerçekleştirilmiştir. Bir sonraki adımda ise BKYSA kullanılarak sınıflandırma gerçekleştirilmiştir. İki yöntemle de yapılan sınıflandırmaların test sonuçları karşılaştırılmıştır. BKYSA ile çalışıldığında veri boyutunda çok büyük oranlarda azalma olmuş, verilerin kalitesi artmış ayrıca sonuçların doğruluğu ve bir birlerine olan yakınlığı artmış ve eğitim sürelerinde büyük oranlarda kısalma olmuştur.

Anahtar Kelimeler: YSA, BK, BCO, DCT, DWT, biyometrik sistemler, iris tanıma, görüntü işleme.

GENERAL KNOWLEDGE

Name and Surname : Hilmi ARIÇ
Field : Graduate Institute of Science and Engineering
Program : Compute Engineering
Supervisor : Prof.Dr. Mübariz EMİNLİ
Degree Awarded and Date : Master of Science – June 2011
Jury :
Prof. Dr. Mübariz EMİNLİ
Prof. Dr. Muhammet KÖKSAL
Yrd.Doç.Dr. Murat BEKEN

BIOMETRIC RECOGNITION WITH FUZZY CLUSTERING NEURAL NETWORKS

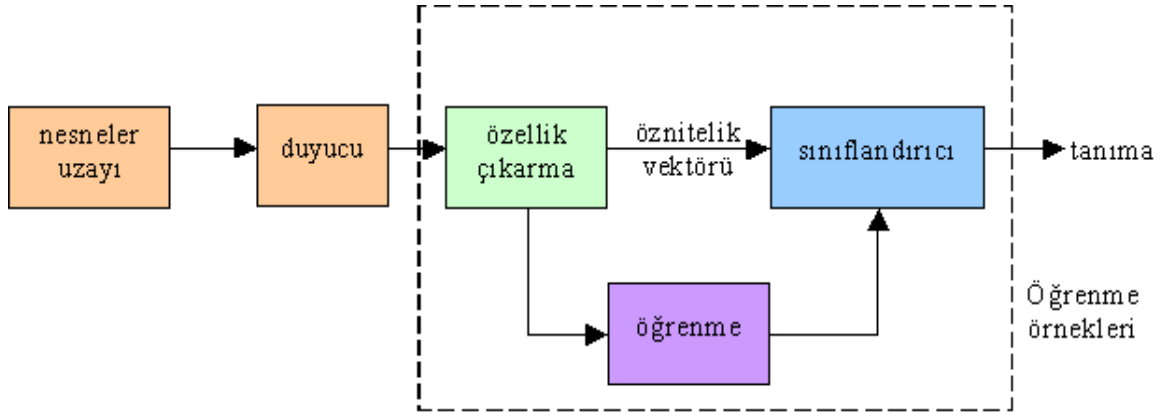
ABSTRACT

In this thesis we applied Fuzzy Clustering Neural Networks to the 2D pattern recognition problem. The methodology is developed in this thesis can be applied to lots of biometrics. For this research Iris is chosen as a biometric and iris recognition is studied. Discrete Wavelet Transform (DWT) and Discrete Cosine Transform (DCT) are used for feature extraction. Feature Vectors are clustered using one of the Fuzzy Clustering algorithms which is Fuzzy C Means (FCM) and this clusters are classified with Artificial Neural Networks. First of all clusters are classified with classical Artificial Neural Networks. At the next step, clusters are classified with Fuzzy Clustering Neural Networks. Test results from both methods are compared. When working with Fuzzy Clustering Neural Networks the amount of data size decreased, quality of data increased, accuracy and precision of the results increased and also training time decreased.

Key Words: Artificial Neural Networks, Fuzzy Clustering Neural Networks, FCM, DCT, DWT, Biometric Systems, Iris Recognition, Image Processing.

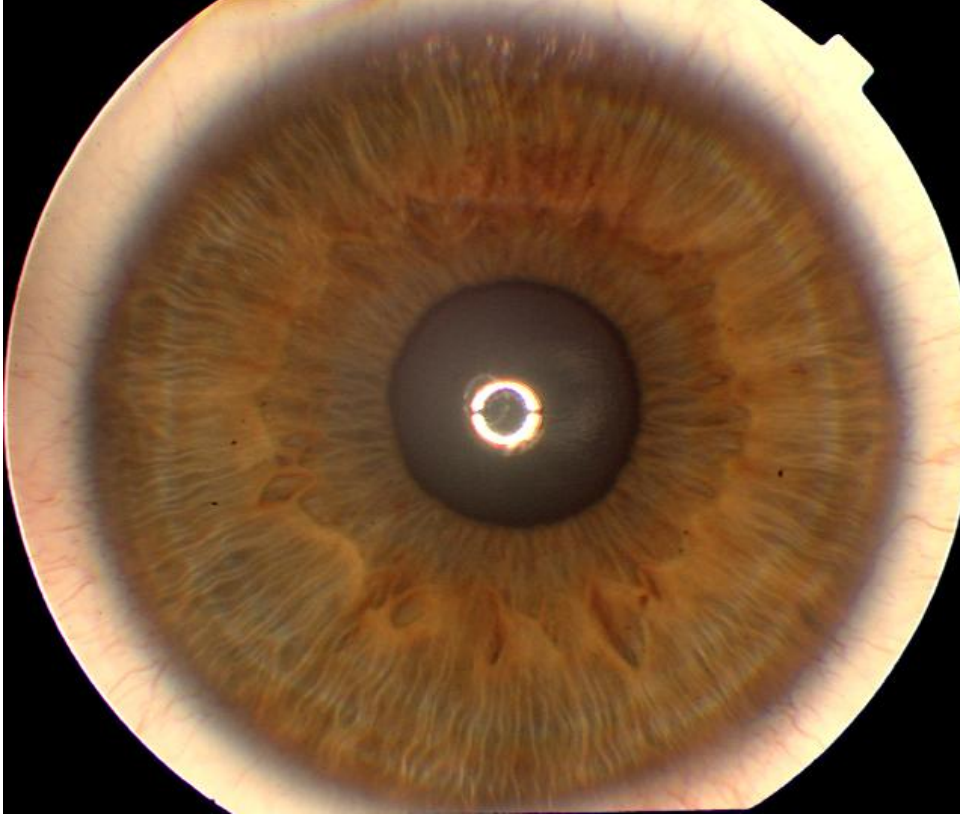
1 GİRİŞ

Biyometrik Sistemler bireyin belli biyometrik özelliğini sadece o kişiye özel tek ve benzersiz bir koda dönüştürür ve bu kod sayesinde karşısındaki bireyi tanır. Bu tez çalışmasında da bir biyometrik sistem geliştirilmiştir. Geliştirilen biyometrik sistem belli kriterleri sağlayan birçok biyometrik için uygulanabilir. Bu çalışmada, biyometrik sistemler içerisinde en güvenilirlerden biri olan ve insanın fiziksel bir özelliği olarak güvenlik ve tanıma alanında kullanılan iris tanıma üzerinde durulmuştur. Ancak geliştirdiğimiz yaklaşım parmak izi, yüz tanıma, damar tanıma gibi değişik biyometriklere de uygulanabilir olacaktır. Tanıma işlemi (Şekil 1.1) genel olarak şu aşamalardan oluşmaktadır: Görüntü kaydetme, Ön-işleme, Görüntü işleme – Öz nitelik vektörünü çıkarma, Nesne tanıma, Sınıflandırma.



Şekil 1.1. Öğrenme işlemli nesne tanıma sisteminin blok diyagramı (Karlık, B., 2009 a)

İris görüntüsünün alınması işlemi için genellikle CCD (Charge Coupled Device) kameralar kullanılmaktadır. Tez çalışmasında kullanılan göz görüntüleri Michal Dobeš ve Libor Machala tarafından oluşturulan bir iris veritabanından elde edilmiştir (Iris Database, 2009). İris veritabanı 3 x 128 iris görüntüsü (3 x 64 sol ve 3 x 64 sağ) içermektedir. Görüntüler; 24 bit - RGB, 576 x 768 piksel boyutunda ve PNG dosya formatındadır (şekil 1.2). İrisler TOPCON TRC50IA optik cihazının bağlı olduğu SONY DXC-950P 3CCD makine ile alınmıştır. (Dobeš, M. ve ark, 2006, Dobeš, M. ve ark, 2004).



Şekil 1.2. İris veritabanından bir örnek

Ön işleme aşamasında ilk yapılan işlem göz resminden iris desenlerinin ayrıştırılmasıdır. Ayrıştırma işleminde irisin göz kapağı, kirpik ve göz bebeği gibi gürültü içermeyen bölümleri çıkarılmıştır. Her bir göz resminden 160x135 piksellik bölümler çıkarılmış, bu işlemle ilgili detay bilgiler aşağıda verilmiştir.

İris veritabanından 3x64 sol irisle çalışma yapılmıştır. Yapılan testlerde irisler iki kombinasyonda kullanılmıştır. Bunlardan ilkinde 3 iristen 2 tanesi eğitim, 1 tanesi test için; diğerinde ise 1 tanesi eğitim 2 tanesi test için kullanılmıştır.

Öz nitelik vektörleri çıkartılırken Ayrık Kosinüs Dönüşümü (DCT) ve Ayrık Dalgacık Dönüşümü (DWT) kullanılmıştır. İris deseninin göz resminden çıkarılmasından sonra sayısallaştırma işlemi yani iris görüntülerinin öz nitelik vektörleri oluşturulmuştur. Öz nitelik vektörü biyometrikleri temsil etmesinden dolayı ve tanımda kullanılacağından dolayı biyometrik sistemin en önemli parçasıdır. Öz nitelik vektörünün boyutu eğitim süresini kesin azaltır denilemese de test süresini (karşılaştırma süresini) kesin azaltmaktadır. Ancak boyutun azalması, görüntüyü niteleyen bazı öz niteliklerin kaybolması anlamına gelmektedir. Bu nedenle öz nitelik vektörü çıkartılırken ve öz nitelik vektörü üzerinde kümele yapılırken, özelliklerin kaybolmamasına çalışılmıştır. Tez çalışmasında, öz nitelik çıkarımı için iki farklı yol izlenmiştir.

Sınıflandırma aşamasında iki yöntem kullanılmıştır. Bunlardan bir tanesi klasik YSA bir diğeri ise Bulanık Kümelemeli Yapay Sinir Ağlarıdır (BK YSA).

YSA biyolojik sinir ağlarının modellenmesi ile ilgili yapılan çalışmaların sonucunda ortaya çıkmıştır. Geliştirilen birçok YSA, biyolojik sinir ağlarının bilinen temel birkaç özelliğini (öğrenme kabiliyeti gibi) benzetmek üzere geliştirilmiştir. YSA ile sınıflandırma sırasında çok katmanlı idrak (Multi-Layered Perceptron - MLP) sinir ağı modeli uygulanmıştır. YSA'nın eğitimi aşamasında geriye yayılım algoritması ve genelleştirilmiş delta kuralı kullanılmıştır.

BK YSA, İki alt ağ katmanının bütünleşmesinden oluşan bir yapıdır. İlk katmanda kendi kendine organize olan BK yöntemi vardır. İkinci katman ise eğitilmeli bir ağ olan YSA yönteminden oluşur. BK katmanı, sınıflandırılması istenilen verinin bir ön sınıflandırmasını gerçekleştirir (danışmansız öğrenme). YSA katmanı, ön sınıflandırma kısmından gelen verileri girdi olarak alır ve kesin sınıflandırmayı yapar (danışmanlı öğrenme). Bulanık kümele yönteminde Bulanık C Ortalamalar (BCO) algoritması kullanılmıştır. DCT ve DWT işlemlerinden sonra (öz nitelik vektörlerinin çıkarılmasından sonra) görüntü karesel olarak çeşitli boyutlarda (3x3, 4x4, 5x5) pencereler kullanılarak giriş görüntüsü alt görüntülere bölünecek ve her bir alt görüntüye bulanık kümeleme yöntemi uygulanarak her bir alt parçadan ikiye bölünecek kümeler elde edilmiştir. Elde edilen kümeler bir araya getirilip YSA ile sınıflandırma gerçekleştirilmiştir. Şuana kadar bahsedilen tüm işlem adımları, bir blok şema halinde Şekil 1.1'de gösterilmiştir.

Yapılan çalışmada öznitelik vektörü çıkarımı tamamlandıktan sonra klasik YSA ile sınıflandırma gerçekleştirilmiş ve sonuçlar kaydedilmiştir. Bir sonraki aşamada ise öz nitelik vektöründen Bulanık Kümeleme ile küme merkezleri elde edilmiş ve elde edilen küme merkezlerinin YSA ile sınıflandırması yapılmıştır. İki yöntem ile yapılan sınıflandırma sonuçları karşılaştırılmıştır. Daha önceki çalışmaların çoğunda sınıflandırıcı başarımına bakılarak karar verilmiştir. Bu çalışmada sınıflandırıcı başarımlarıyla birlikte ROC (Receiver Operating Characteristic) eğrisinin altında kalan alan (AUC) hesaplanarak, karşılaştırmalarda kullanılmıştır.

Çalışmanın son aşaması olan sınıflandırma işleminde hem bulanık kümelemeli yapay sinir ağları hem de klasik yapay sinir ağları yöntemleri uygulanacak olup geliştirilen sistemin performansı test edilmiştir.

BKYSA, literatürde bir boyutlu (1-D) işaretlerde kullanılmıştır. Örneğin, aktif mikrodalga aygıtlarının nitelendirilmesinde (Karlık B., 2003), çok fonksiyonlu kol protezi uygulamasında (Karlık, B. ve ark, 2003), Elektrokardiyogram (EKG) ile kalp ritim bozukluklarının tespitinde (Özbay, Y. ve ark, 2006), koku analizinde (Karlık, B. ve ark, 2007) BKYSA başarı ile uygulanmıştır. Bu uygulamalarda kullanılan işaretlerin çoğunluğu tek boyutlu işaretlerdir.

İris işareti, doğası gereği iki boyutlu bir işarettir. Bu tez çalışmasında, BKYSA algoritmasının iris tanıma problemine uygulanması hedeflenmiştir. İris işareti, iki boyutlu (2-D) bir işaret olduğundan bu çalışma, BKYSA algoritmasının benzeri diğer işaretler için kullanılabilmesi için ilk adımlardan biri olacaktır. İris tanıma, örüntü tanıma konusunun bir parçasıdır. İyi korunan bir organ olması, ömür boyu sabit kalan bir yapıda olması, tek yumurta ikizlerinde bile iris deseninin farklı olması, yine aynı insanın irislerinin birbirinden farklılık arz etmesi, iris deseninin çok fazla farklılık içermesi nedeni ile çalışmaların yoğunlaştığı bir alandır.

1.1 Tezin Amacı

Bu tez çalışması genel manada bir araştırma tezidir. Çalışmada BKYSA yönteminin, iki boyutlu bir örüntü tanıma problemi olan Iris tanıma işlemine uygunluğu araştırılmıştır. Geliştirilen yöntem genel olarak iris, yüz, parmak izi gibi değişik biyometrik yapılara uygulanabilir olmuştur.

İki boyutlu örüntüden ilk olarak özellik vektörü çıkarılmıştır. Bu amaçla örüntüye DWT ve DCT uygulanmıştır. Daha sonra BK ile öznelik vektöründen kümeler elde edilip YSA ile de sınıflandırma gerçekleştirilmiştir. Geliştirilen algoritma kullanılarak biyometrik tanıma verileri elde edilip bulunan sonuçlar yorumlanmıştır.

Yapılan çalışmada bulanık kümeleme işlemi ile eğitilecek verinin büyüklüğü azaltılarak YSA'na verilecektir. Bunun önemli bir sonucu, YSA'nın eğitim süresinin ciddi anlamda düşmesidir. Ayrıca bu boyut düşürmesi, tanımanın da kalitesini artırmaktadır. Çünkü gereksiz ve tanımada hiçbir faydası olmayacak, hatta tanımayı olumsuz etkileyebilecek olan özellikler (features) hem YSA'nın eğitim süresini uzatır, hem de eğitimin, dolayısı ile tanımanın başarısını negatif yönde etkilemektedir. Yapılan bu işlem BKYSA'nın klasik YSA'na göre en önemli üstünlüklerinden biridir. Yapılan çalışma sonucunda sürelerin kısalması yanında sonuçların kalitesi yani doğruluk oranlarında da artma beklenilmektedir. Bu yönü de ikinci en önemli üstünlüğü olacaktır.

Yapılan çalışmada klasik YSA'nın zayıf kaldığı noktalarda iyileştirme getirilmesi amaçlanmaktadır. İki yöntem arasındaki farkın anlaşılması için iki yöntemle de aynı veri kümeleri üzerinde testler yapılacak ve sonuçlar karşılaştırılmıştır. İki sınıflandırma yönteminde de aynı öz nitelik vektörü üzerinde çalışma yapılmıştır.

1.2 Hedefler ve Tezin Başarı Kriterleri

Tezin hedefi BKYSA yönteminin 2 boyutlu biyometrik resimler üzerine uygulanması ve sonuçlarının değerlendirilmesidir.

Temel başarı kriterleri:

- Yanlış Tanıma Oranı (YTO), bir test irisinin veri tabanındaki irislerden birisi ile eşlenmiş olmasına rağmen gerçekte o irisin veri tabanında kayıtlı olmamasıdır ya da yanlış eşleşmesidir.
- Yanlış Reddetme Oranı (YRO), veri tabanında kayıtlı bir irise ait bir örneğin karşılaştırma sırasında veri tabanında bulunamamasıdır.
- Doğru Tanıma Oranı (YRO), veri tabanında kayıtlı bir irise ait bir örneğin karşılaştırma sırasında veri tabanında bulunması ve doğru olarak eşleştirmesidir. Tanıma oranı belirtilen eşik değerinin üstünde olması gerekmektedir.

- AUC, ROC eğrisinin altında kalan alandır (AUC). ROC basit anlamda doğru tanımların, yanlış tanımlara olan kesri olarak da ifade edilebilir.

Karşılaştırmada her yöntem bir eşik değerine bakarak karar vermektedir. Eğer eşik değeri düşük tutulursa YTO yükselir. Eğer eşik değeri yükseltirise YRO yükselir. Bu nedenle de her yöntem için seçilecek eşik değeri, bir parametre ayarlaması sonrası tespit edilmektedir. En önemli başarı kriteri YTO ve YRO değerlerinin düşük olmasıdır. Eğitim süresinin kısa olması, Testlerde doğru eşleme oranının yüksek olması.

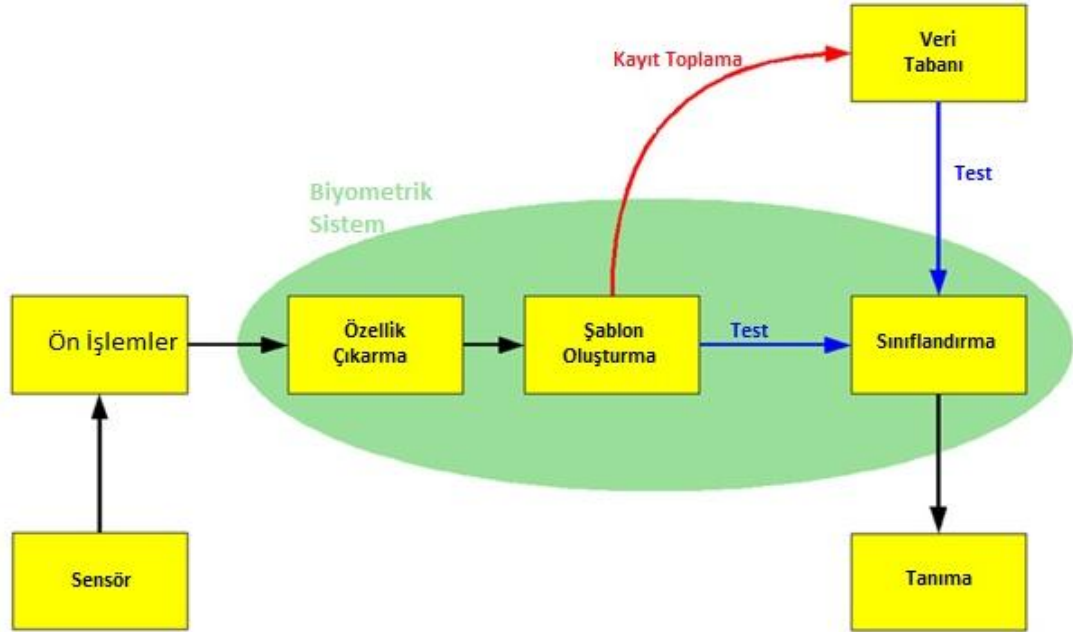
2 BİOMETRİK TANIMA

Biyometrik; kişilerin kendilerine has ölçülebilir fizyolojik özelliklerini kullanarak kimlik saptamak üzere geliştirilmiş sistemler için kullanılan bir terimdir.

Biyometrik Sistemler bireyin belli biyometrik özelliğini sadece o kişiye özel tek ve benzersiz bir koda dönüştürür ve bu kod sayesinde karşısındaki bireyi tanır. Biyometrik sistemler yetkili kişilerin geçişine/erişimine izin verildiği yerlerde kişinin kim olduğunun belirlenmesinde kullanılırlar.

Kart ve şifre gibi kaybolabilen, çalınabilen veya unutulabilen tanıtıcılar yerine direkt olarak insanın kendisinin kullanılması sistemi daha güvenilir ve daha kullanışlı hale getirmektedir. Ayrıca, Kimlik Saptama kişinin fiziksel veya davranışsal özelliği ile yapıldığı için kişi özelliğini başkasına devredemez, unutamaz veya kaybedemez. Dolayısı ile kartlı veya şifreli sistemlerde yaşanan kaybedilme, unutulma veya çalınma gibi problemler de yaşanmaz.

En çok kullanılan biyometrik parmak izidir, bununla birlik yüz ve iris tanıma da çok çalışılan biyometrik çeşidi durumuna gelmiştir. Biyometrik sistemlerin çalışma prensipleri hepsinde aynıdır. İlk olarak tarama yapılır. Tarama sonucunda elde edilen bilgiler görüntü işleme yöntemleri ile analiz edilir ve öznelikleri çıkarılır. Öznelikler sayısal ortamda kaydedilir. Gerekirse güvenlik amacıyla bu bilgiler şifrelenerek veritabanında tutulur. Sisteme giriş yapacak kişinin bilgileri ile veritabanına önceden kaydedilmiş olan bilgiler karşılaştırılarak giriş onaylanır veya reddedilir. (Koçer, H. E., 2007) Genel anlamda bir biyometrik sistemin aşamaları Şekil 2.1 de gösterilmiştir.



Şekil 2.1. Biyometrik sistem blok diyagramı (Biyometrik Diagram, 2011)

Genel olarak bilinen biyometrikler şunlardır:

Fizyolojik:

İris,
Yüz,
Parmak izi,
El damar (Elin tersi)
geometrisi,
Retina,
DNA,
Kulak şekli,
Vücut ısısı,

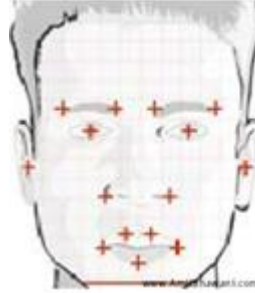
Davranışsal:

Ses,
El yazısı,
İmza,
Yürüyüş,
Dudak hareketleri,

2.1 Yüz tanıma

Yüz Tanıma sistemi, bazı yüzsel karakteristiklerin analizini, bunların veritabanına kaydedilmesini ve kimlik saptamasında canlı yüzle kayıdı karşılaştırılmasını ve uyumluysa kullanıcının erişimine izin verilmesi işlemini içerir. Yüz karakteristiği tanımlanırken göz çukurlarının saptanması, elmacık kemiğini çevreleyen bölgelerin taranması, ağız kenarlarının belirlenmesi, kulak memesinin analizi gibi çeşitli metotlar kullanılır (Şekil 2.2) (Hong ve Jain 1998).

Birçok yüz tanıma sisteminde saç stili, saçın uzunluğu veya kısalığı gibi belirleyicilere dikkat edilmez. Örnek imaj oluşturulurken en genel yöntem, kişinin kamera karşısında sabit dururken çeşitli açılardan yüzün görüntülerinin alınmasıdır. Bunun yanı sıra, kızılötesi algılayıcılarla da yüzün farklı açılardan görüntüleri alınabilir. Alınan görüntü işlenerek daha önce alınmış olan görüntü bilgisiyle karşılaştırılır. Görüntünün öznitelikleri çıkarılırken göz çukuru, burun, kulak gibi organların konumuna, bunlar arasındaki çapraz ilişkilere ve yüzün genel durumuna dikkat edilir.



Şekil 2.2. Yüz tanıma

Yüz tanıma sistemleri fiziksel bir temas gerektirmediğinden daha çok tercih edilirler. Ancak yüz tanımlama sistemleri uygulamalarda sınırlı başarı sağlamışlardır. Çünkü bıyık gelişimi, kilo alma-verme, ikizinin olması gibi bazı tanıma problemleri yüz tanıma sistemlerinde çok sık karşılaşılan durumlardır (Koçer, H. E., 2007, Jain ve ark. 1999).

2.2 Parmak izi tanıma

Parmak izi tanıma sistemleri günümüzde en yaygın kullanılan biyometrik tanıma sistemidir. Parmak izi sisteminin bulunuşu çok eski tarihlere dayanır, fakat bu izden istifade etmek oldukça yenidir. Eski literatürde parmak izi konusunda bazı kayıtlar varsa da bu kayıtlarda parmak izinin kullanılması hususunda herhangi bir bilgiye rastlanmamaktadır. İlk önce, Nehemiah Grew (1684), Marcello Malpighi (1686) ve J. E. Purkinje (1823) gibi anatomistler insanların parmaklarındaki kıvrımların bazı özellikleri bulunduğuna dikkat çekmekle beraber, bu izlerden faydalanma metotlarını belirtmemişlerdir (Şekil 2.3).



Şekil 2.3. Parmak izi tanıma (Vizyotek, 2011)

Oldukça uzun bir geçmişe sahip olan bu biyometrik çeşidi 20. yüzyılın başlarında özellikle kriminal uygulamalarda tercih edilmiştir. Parmak izi taranırken çoğunlukla optik algılayıcılar kullanılmaktadır. Ancak son yıllarda teknolojideki gelişmelerle birlikte termal ve kapasitif ölçüm yapan tarayıcı cihazlar kullanılmaya başlanmıştır. Optik tarayıcılar bir ışık kaynağından gönderilen ışınların tarayıcı yüzey üzerine yerleştirilmiş parmakta bulunan çukurlar ve çıkıntılara göre yansiyarak CCD algılayıcılara gönderilmesi prensibine göre çalışır. Diğer tip tarayıcılar ise yalnızca parmaktaki izleri taramakla kalmaz aynı zamanda parmaktaki statik etkileri ölçerek taranan parmağın canlı bir parmak olup olmadığını tarar (Inglis ve ark. 1998, Lee ve ark. 1999, Dickinson ve ark. 2000). Termal tarayıcılar ise parmakta bulunan girinti ve çıkıntıların arasındaki ısı farklılıklarını tespit ederek parmak izini oluşturur. Bir parmak izindeki tüm izler dikkate alınmaz. Bunun yerine izdeki bazı özel noktalar; bunlar, girinti ve çıkıntı arası uzaklıklar, hat sonları, çatallaşmalar, bazı kesişim bölgeleri v.b. işlenir ve daha sonra çeşitli görüntü işleme yöntemleriyle sayısal hale getirilir ve veritabanına kaydedilir.

Parmak izi tanıma sistemleri kullanım kolaylığı, düşük fiyatı ve küçük ebatları nedeniyle ev, ofis gibi yerlerde daha çok tercih edilirler. Ancak tarama sırasında optik tarayıcıyla fiziksel temasın olması, parmağın kirli, yağlı veya ıslak olması gibi görüntü kalitesini düşürebilecek etkenlerin bulunması, çabuk kopyalanabilir olması ve aşırı sürtünmeden izlerin bozulması gibi faktörler parmak izi tanıma sistemlerini diğerlerinden daha az güvenli yapmaktadır (Koçer, H. E., 2007, Uludağ ve ark. 2001).

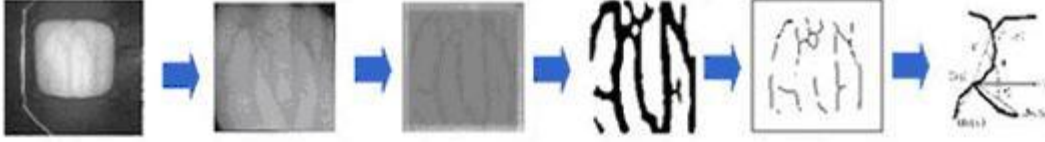
2.3 Yürüyüş tanıma

Davranışsal biyometriklerde yürüyüş tanıma düşüncesi yirminci yüzyılın sonlarında ortaya atılmıştır. Genellikle video görüntüleri kullanılarak elde edilen yürüyüş şekli kişilerin tanınmasında kullanılabilir. Bu alanda çalışmalar devam etmekte olup şuan bazı havaalanlarında suçluların bulunmasında kullanılmaya başlanılmıştır. Yürüyüş şeklinin zamanla değişken olması ve giyilen kıyafetin aşırı değişmesinin yürüyüşü etkilemesi bu tür bir tanıma sistemini güvensiz hale getirmektedir.

2.4 El ve Damar Tanıma

El & Damar tanıma el üzerindeki el & damar deseni yapısına göre tanıma yapmaktadır. Her kişinin el & damar deseni birbirinden farklıdır.

Damar tanıma işlemi el üzerinde bulunan deri altı damar görüntüsünün alınması işlemi ile başlamaktadır (Şekil 2.4).



Şekil 2.4. Damar tanıma

Alınan görüntü üzerinden ana damar hatları ayrılmakta ve özellik çıkarma algoritması uygulanarak her kişi için belirleyici ID kod oluşturularak veri tabanına kaydedilmektedir. (Vizyotek, 2011)

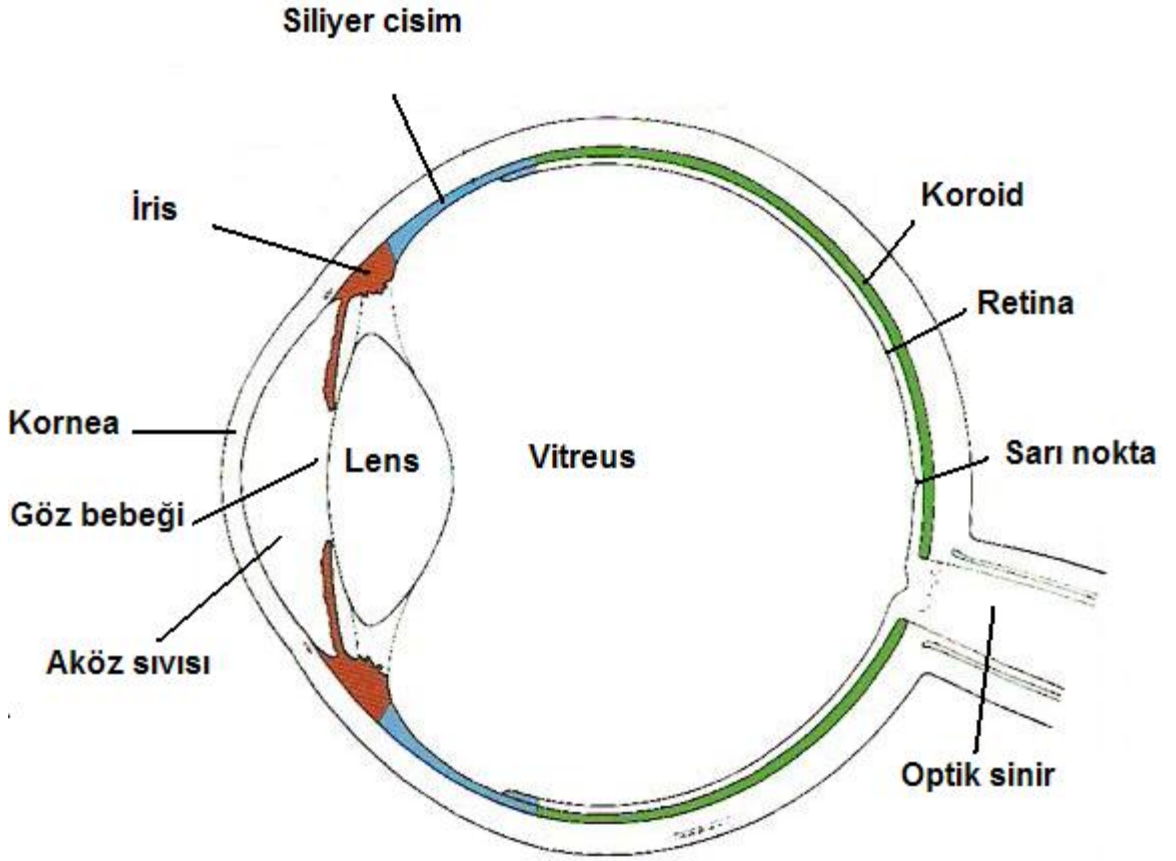
2.5 Retina tanıma

Retina göz yuvarlağının iç kısmında arka tarafta yer alan ince sinirlerin ve damarların bulunduğu ağ tabakadır. Bu tabakada yer alan ışığa duyarlı sinirler ışığı optik sinirler vasıtasıyla beyne iletir. Retina tarayıcı cihazlar gözbebeği içerisinden tarama yaparlar. Bir optik algılayıcı retinanın yapısını düşük yoğunluklu ışınlar kullanarak tarar. Bu işlem sırasında kullanıcı yaklaşık 1cm'lik bir delikten kıyımdamadan bakar. Tarayıcı cihaz tarama sırasında yaklaşık altı tur döner ve her turda belirlenen noktaları kaydeder. Daha sonra bu bilgiler dijitalleştirilerek kaydedilir (Hill 1999). Ancak tarama sırasında gözün tarayıcıya fiziksel teması, gözde oluşabilecek ve retina yapısına zarar verebilecek travmaların olması, tarama işleminin oldukça zahmetli olması ve uzun sürmesi gibi faktörler retina tanıma sistemlerini daha az tercih edilir hale getirmiştir. (Koçer, H. E., 2007)

2.6 İris Tanıma

İris, gözün ön bölümünde saydam katmanın arkasında bulunan ve göze rengini veren damarlı bölgedir. Ortasında bulunan ve göz bebeği denen delik, ışığın az ya da çok olmasına göre genişleyip büzülen bir çeşit diyaframdır. Saydam katman ile göz billuru arasındaki boşluğu ön oda ve art oda olmak üzere iki kısma ayıran iris, içten dışa doğru 3 parçaya ayrılır (Şekil 2.5):

- Renksiz hücreli ön endotelyum
- Asıl iris dokusu
- Arka epitelyum



Şekil 2.5. Göz anatomisi (İris, 2011)

İrisin genel özellikleri:

- Dünya üzerinde bulunan her insan gözü eşsizdir.

- Dünyada aynı irisin olma olasılığı 1/10 üzeri 78 'dir.
- Tek yumurta ikizleri aynı DNA yapısına fakat farklı iris yapısına sahiptir.
- Göz irisi genetik oluşumlardan en az derecede etkilenir.
- Göz irisi kalıtsal hastalıklardan etkilenmez.
- Irk, cinsiyet, renk gibi demografik özellikler irisi etkilemez.
- Gözle görülebilen ve hassasiyet ile ölçülebilen bir organdır.
- Ömür boyu değişmeyen tek organdır.
- İnsanın doğumun 16. ayından itibaren ölüme kadar değişmez.
- Göz oluşumu itibari ile muhafazası (göz kapağı) bulunan bir organdır.
- Doğal içgüdüler ile refleks olarak korunan yegâne vücut parçası olan başımızda yer alır.
- Göz, insanın yaşamını yitirmesinin ardından canlılığını en çabuk (3 sn) kaybeden organdır (Ergosis, 2010).

1936 yılında Frank Burch isimli göz doktoru bir bireyi tanımak için iris desenlerinin kullanılabileceği fikrini öne sürdü. 1985 yılında Dr. Leonard Flom ve Dr. Aran Safir tüm irislerin eşsiz olduğunu ispatlayarak 1987 iris tanıma ile ilgili patentlerini aldılar. Dr. Flom, Dr. John Daugman'dan iris tanımayı otomatik hale getirecek bir algoritma geliştirmesini istemiştir. 1993 yılında ABD'de Defense Nuclear Agency bu işlemi yapacak bir prototip üstünde çalışmalara başlamıştır. 1995 yılında tamamlanan sistem Dr. Daugman'a bu dalda bir patent kazandırmıştır. 1995 yılında otomatik iris tanıma yapabilen cihazlar piyasaya çıkmaya başlamıştır.

Özellikle Dr. John Daugman'ın geliştirdiği yöntem, Iris tanıma konusunda ilk çalışmadır (Daugman, J. 1993, 2001, 2004). İlk olarak 1993 yılında önerilmiş ve yıllar içinde geliştirilmeye devam edilmiştir. Bu çalışma, faz temelli olup iris tanıma yöntemlerinin çoğuna temel teşkil etmektedir. Ancak bu çalışma patentli bir ürün haline getirilmiş olduğundan çalışma sistemi tam olarak bilinmemektedir (Cui, J., 2004, Proença, H., 2007). Özellikle parametrelerin değerleri nasıl ayarlandığı konusunda bir bilgi yoktur. Bu çalışmanın en önemli kısıtlarından birisi de kullanıcının işbirliğine ihtiyaç duymasıdır. Yani bir kullanıcı, göz taraması yapacak cihaza belli bir mesafe yaklaşacak, belli bir süre belli bir açı ile bakacak, elde edilen kaydın kalitesi (odak, çözünürlük, vs) belli bir eşik değerinden yukarıda olacak. Böyle bir uygulamadan bir kesit Şekil 2.6'da gösterilmektedir. Günümüzde ise daha çok kullanıcının işbirliğine ihtiyaç duymayan Iris tanıma yöntemleri geliştirilmeye çalışılmaktadır. Bu durumda kullanılan iris resimleri daha fazla gürültü barındırmaktadır. Bu gürültülerin bazıları şunlardır: alt ve üst kapak, kirpikler, odaklanamama, göz hareketinden dolayı bulanıklaşma (blurring), gözlük veya kontak lensler, vb (Proença, H. ve ark, 2007).

Yukarıda bahsedilen gürültüler, iris tanıma kalitesini negatif yönde etkilemektedir. Oluşabilecek hataları iki başlıkta toplayabiliriz: Yanlış Tanıma Oranı (YTO), bir test irisin veri tabanındaki irislerden birisi ile eşlenmiş olmasına rağmen gerçekte o irisin veri tabanında kayıtlı olmamasıdır. YRO ise, veri tabanında kayıtlı bir irise ait bir örneğin karşılaştırma sırasında veri tabanında bulunamamasıdır. Karşılaştırmada her yöntem bir eşik değerine bakarak karar vermektedir. Eğer eşik değeri düşük tutulursa YTO yükselir. Eğer eşik değeri yükseltirse YRO yükselir. Bu nedenle de her yöntem için seçilecek eşik değeri, bir parametre ayarlaması sonrası tespit edilmektedir.

Burada Iris tanıma konusunda bir çok yöntemin de temelini oluşturan ve artık bir standart kabul edilen yöntem, YTO'nun 200 milyarda 1 olduğunu söylemektedir (Daugman, J., 2005). Ancak bu iddianın çok idealist olduğunu söylemek gerekir (Xie, L., 2007). Bu, patent dolayısı ile kullanılan resimlerden test koşullarına kadar birçok bilginin gizlenerek verildiği bir sonuçtur. Bilimsel anlamda bağımsız denetimden uzak olması nedeniyle de eleştiriye açıktır. Son zamanlarda yapılan üç adet bağımsız kuruluş tarafından gerçekleştirilen denetim raporları da böyle bir sonucu ortaya koymaktadır. Her üç değerlendirme de benzeri şekilde sonuç vermiştir. Burada YTO'nun 0.001 olduğu durumlarda YRO 0.01 ile 0.03 aralığında gerçekleşmektedir (Newton, E.M. ve ark 2007, Phillips, P., 2007, Authenti-Corp, 2007).

İris tanıma işlemi temelde beş kısımdan oluşmaktadır. Bunlar, resim alınması, ön işlemler, öznitelik vektörü çıkartılması, sınıflandırma ve tanımadır (Gupta, P., 2006). Bu çalışmada öznitelik vektörü çıkartılması ve sınıflandırma adımlarında çalışmalar yer alacaktır.



Şekil 2.6. Schiphol havaalanında iris tanıma uygulaması (Daugman, J.)

İris tanımanın dezavantajları:

- Tanımanın yapılabilmesi için çekilen resmin çözünürlüğünün ve kalitesinin çok iyi olması gerekir. Aksi halde beklenen sonuçlar elde edilmeyebilir.

- Yüksek çözünürlüklü iris resimleri ve üzerine iris deseni basılmış lensler ile varolan sistemleri aşmak mümkün olabilir.

2.7 Biyometrik Sistemlerin Karşılaştırılması

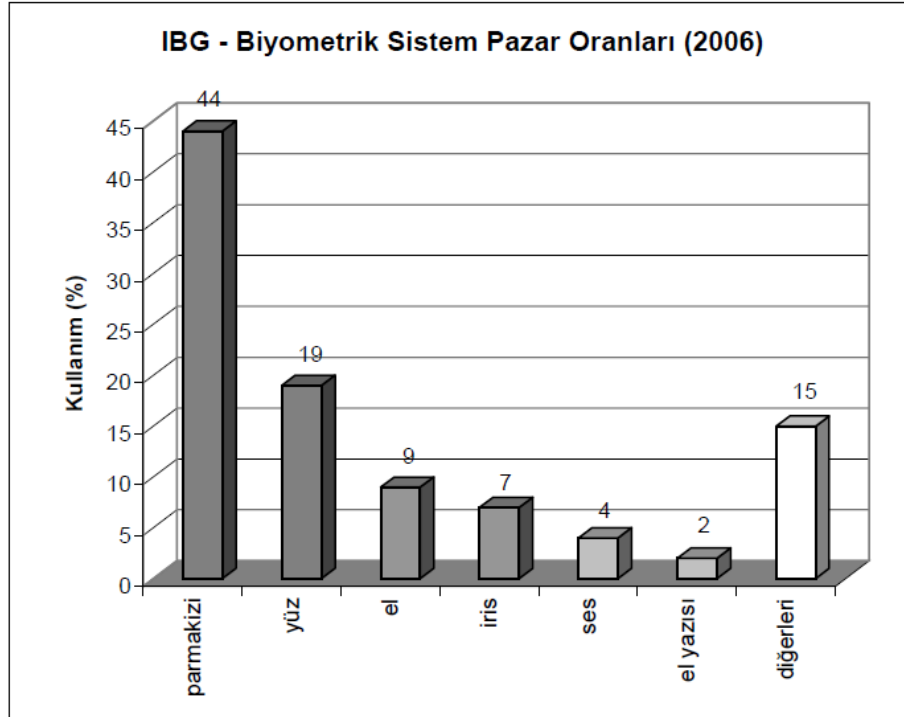
En sık kullanılan biyometrik sistemlerden bazıları; güvenlik seviyesi, kullanım kolaylığı, kullanıcıya uygunluk, dayanıklılık, sistemdeki hata oranı ve maliyet açısından karşılaştırılmış ve değerlendirme sonuçları tablo halinde gösterilmiştir (Tablo 2.1).

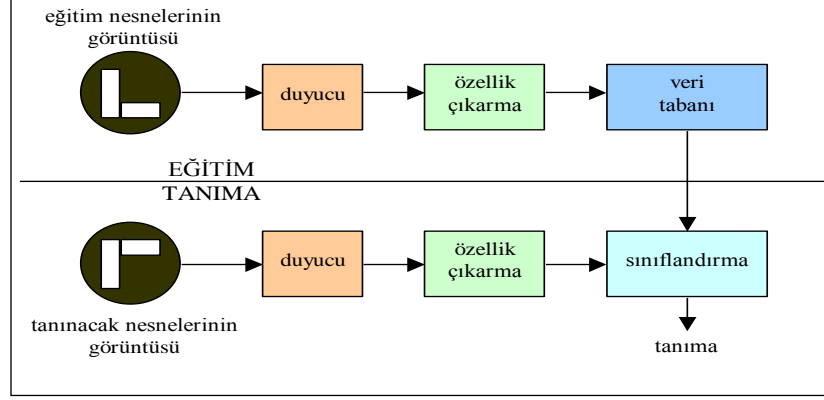
Tablo 2.1. Biyometrik Sistemlerin Karşılaştırılması

	Yüz	Parmakizi	El	Ses	İris	El Yazısı (İmza)
Güvenlik Seviyesi	Orta	Yüksek	Orta	Orta	Çok Yüksek	Düşük
Kullanım Kolaylığı	Orta	Yüksek	Yüksek	Yüksek	Yüksek	Yüksek
Kullanıcıya Uygunluk	Orta	Yüksek	Orta	Yüksek	Orta	Yüksek
Dayanıklılık	Yüksek	Yüksek	Yüksek	Yüksek	Çok Yüksek	Yüksek
Hatayı artıran etkenler	Yaşlanma Saç, sakal Gözlük	Kirlilik Nemlilik Yaşlanma	Kaza Yaşlanma	Gürültü Hastalık Yaşlanma	Düşük aydınlatma	El yazısının değişmesi
Maliyet (tarayıcı)	< \$200	< \$100	< \$500	< \$5	< \$300	< \$300

Biyometrik sistem seçilirken bazı önemli hususlar dikkate alınmalıdır. Öncelikle biyometrik sistemle çözümlenecek olan problem iyi analiz edilmelidir. İhtiyaçlara göre sistemin seçimi yapılmalıdır. Örneğin güvenlik seviyesi ve biyometrik özelliğin dayanıklılığı çok yüksek olması gereken bir yere biyometrik sistem uygulanacaksa o halde iris tanıma sistemi kullanmak daha uygun olacaktır. Kullanım kolaylığı ve maliyet ön planda ise, o halde parmak izi tanıma sistemini seçmek daha doğru olacaktır. Ayrıca hatayı artırıcı etkenlere de dikkat edilmelidir. Kişinin yaşlanması ile biyometrik özellikleri de değişime uğrayacaktır. Dolayısıyla sistem kişiyi tanımakta güçlük çekecektir. İrisin bir ömür boyu değişmez bir özelliğe sahip olması iris tanıma sistemlerini daha cazip hale getirmektedir. Buna karşılık ses tanıma sisteminde kullanılan kişinin sesi hastalanma gibi bir sebepten dolayı çok kısa aralıklarda değişime uğrayabilir. Bu da sistemi güvensiz kılmaktadır. Uluslararası Biyometrik Grup (International Biometric Group – IBG) tarafından 2006 yılında biyometrik sistemlerin karşılaştırılması üzerine yapılan araştırma sonuçları aşağıdaki grafikte verilmiştir (Tablo 2.2, IBG 2006) (Koçer, H. E., 2007).

Tablo 2.2. Biyometrik sistem pazar oranları (Karlık, B. 2009)





Şekil 2.7. Bir nesne tanıma sisteminin genel blok diyagramı

3 ÖZ NİTELİKLERİN ÇIKARILMASI

Bir görüntüye ait öznitelikler görüntü tanıma sistemlerinde görüntülerin karşılaştırılması ve tanınması için kullanılır. Herhangi bir görüntünün belirgin ve ayırt edici özelliklerinin sayısal olarak ifade edilmesi işlemine, özellik çıkarma (feature extraction) ve bu işlemi gerçekleştiren sisteme, özellik çıkarıcı (feature extractor) adı verilmektedir.

Şekil veya nesnelere belirlemede kullanılan her bir özelliğin veya ölçülecek büyüklüğün, ölçüm sonucunu veren gerçel sayıya özellik adı verilmekte ve bileşenleri bu özelliklerden oluşmuş vektöre, özelliklerin alındığı nesneye ilişkin öznitelik vektörü denilmektedir. Pratikte uygun özellik seçimi, bir nesne tanıma probleminin en güç ve en önemli bölümünü oluşturur. Eğer seçilen özellikler nesnelere bütün özelliklerini tam olarak taşıyorsa, tanıma ve sınıflandırma işlemlerinde yapılacak hatalar büyük olacaktır (Karlık, B., 2009).

3.1 Ayrık Kosinüs Dönüşümü

Ayrık Kosinüs Dönüşümü (DCT) Fourier Dönüşümü gibi sinüzoidal tabanlı bir dönüşümdür. Aralarındaki en önemli fark; DCT'nin yalnızca kosinüs terimlerini kullanması sinüs terimleri olmaması, bundan dolayı kompleks bileşenler içermemesidir. DCT yüksek iniltiye sahip bilgiyi, iniltisi düşük bilgiye dönüştürür.

İki boyutlu DCT ise normal DCT'nin sırasıyla boyutlara uygulanmasıdır. İlk önce bir boyuta uygulanır daha sonra diğer boyuta uygulanır. Giriş resmi A ve çıkış resmi B için iki boyutlu DCT'nin tanımı:

$$B_{pq} = \alpha_p \alpha_q \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} A_{mn} \cos \frac{\pi(2m+1)p}{2M} \cos \frac{\pi(2n+1)q}{2N}, \quad 0 \leq p \leq M-1, \quad 0 \leq q \leq N-1 \quad (3.1)$$

Burada

$$\alpha_p = \begin{cases} \frac{1}{\sqrt{M}}, & p=0 \\ \sqrt{\frac{2}{M}}, & 1 \leq p \leq M-1 \end{cases} \quad (3.2)$$

ve

$$\alpha_q = \begin{cases} \frac{1}{\sqrt{N}}, & q=0 \\ \sqrt{\frac{2}{N}}, & 1 \leq q \leq N-1 \end{cases} \quad (3.3)$$

M ve N satır ve kolon uzunluğunu göstermektedir. DCT gerçek sayılara uygulandığında sonuçta gerçek sayı olur. DCT bilgiyi daha konsantre hale getirmeye çalışır bu yönüyle DCT resim sıkıştırma işlemleri için uygun bir yöntemdir.

İki boyutlu ters Ayırık Kosinüs Dönüşümü de aşağıdaki gibi tanımlanabilir:

$$A_{mn} = \sum_{p=0}^{M-1} \sum_{q=0}^{N-1} \alpha_p \alpha_q B_{pq} \cos \frac{\pi(2m+1)p}{2M} \cos \frac{\pi(2n+1)q}{2N}, \quad \begin{matrix} 0 \leq m \leq M-1 \\ 0 \leq n \leq N-1 \end{matrix} \quad (3.4)$$

Burada

$$\alpha_p = \begin{cases} \frac{1}{\sqrt{M}}, & p=0 \\ \sqrt{\frac{2}{M}}, & 1 \leq p \leq M-1 \end{cases} \quad (3.5)$$

ve

$$\alpha_q = \begin{cases} \frac{1}{\sqrt{N}}, & q=0 \\ \sqrt{\frac{2}{N}}, & 1 \leq q \leq N-1 \end{cases} \quad (3.6)$$

Genel olarak görüntülerin pikselleri kendi aralarında yüksek iniltiye sahiptir. Eger resimde gereksiz bilgiler varsa küçük hatalarla, yüksek sıkıştırma oranları elde etmek mümkündür. Teorik olarak aynı bilgi daha az sayıda bilgi ile temsil edilebilir (Gunlu, G. ve ark.).

Diğer dönüşümlerle karşılaştırıldığında DCT bazı avantajlara sahiptir. Dikgen bir dönüşümdür ve karmaşık bileşenler içermez. Diğer bir avantajı ise görüntüleri küçük büyüklük (min-max) değerlerine çevirmesidir.

Var olan dönüşüm yöntemlerinde hiçbiri için en ideal tanıma dönüşümü denilemeyeceği için ve DCT'nin bu tarz iyi yönlerinin olması, özellikle yoğun enerji sıkıştırma özelliğinin olması nedeniyle DCT'yi özellik çıkarma işlemi için kullandığımız yöntemlerden biri olarak seçtik.

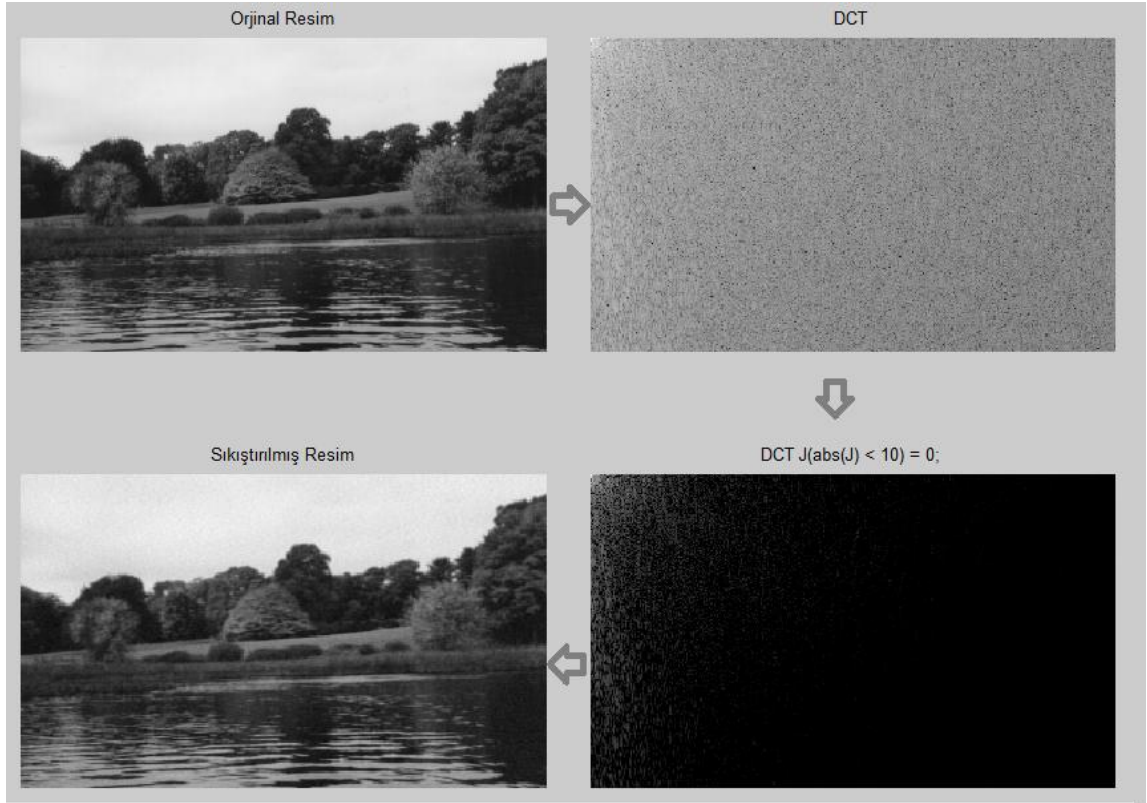
Matlab uygulamasında DCT2 metodu kullanılarak aşağıdaki gibi sıkıştırma işlemi yapılabilir (Şekil 3.1-3.2).

Matlab Kodu:

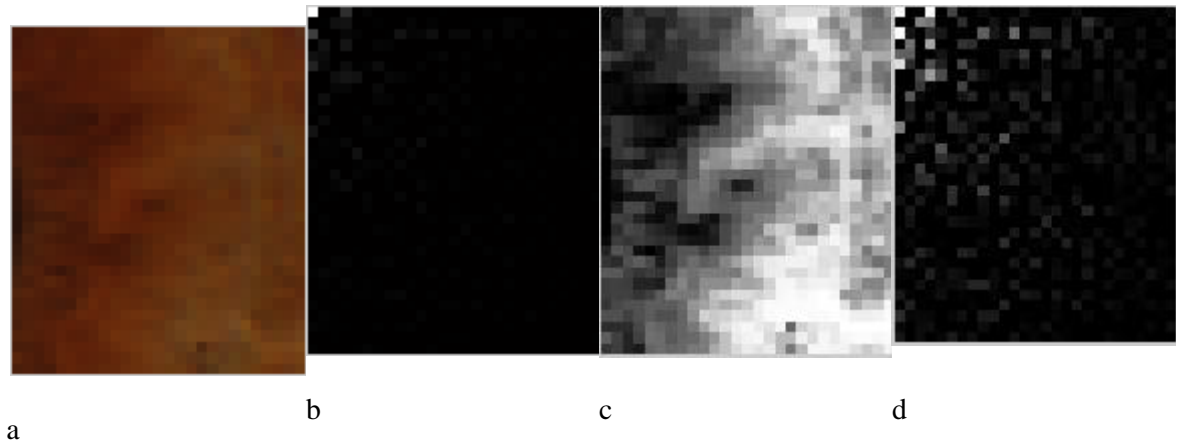
```
RGB = imread('autumn.tif');  
I = rgb2gray(RGB);  
J = dct2(I);  
J2 = J;  
J(abs(J) < 10) = 0;  
K = idct2(J);
```

figure

```
subplot(2,2,1), imshow(I), title('Orjinal Resim');  
subplot(2,2,2), imshow(log(abs(J2)),[]), title('DCT');  
subplot(2,2,3), imshow(K,[0 255]), title('Sıkıştırılmış Resim');  
subplot(2,2,4), imshow(log(abs(J)),[]), title('DCT J(abs(J) < 10) = 0;');
```



Şekil 3.1. DCT ile özellik çıkarma ve resim sıkıştırma



Şekil 3.2. DCT ile özellik çıkarma ve Histogram Eşitleme a) Orjinal parçanın küçültülmüş hali, b) a'nın DCT uygulanmış hali, c) a'nın histogram eşitlenmiş hali, d) c'nin DCT uygulanmış hali

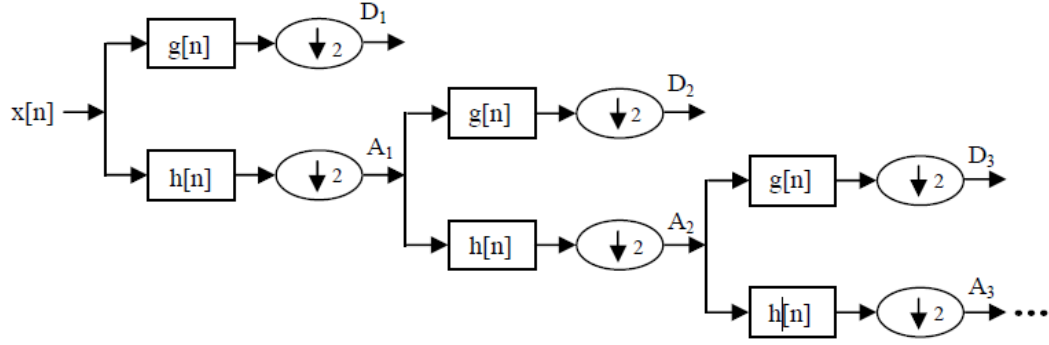
3.2 Dalgacık (Wavelet) Dönüşümü

Dalgacık ifadesi olarak 1909 yılında Alfred Haar tarafından ortaya atılmıştır. Zaman içerisinde Jean Morlet ile Y. Meyer ve arkadaşları metodu geliştirmişler ve 1988 yılında Stephane Mallat önemli katkılar sağlamıştır. Daha sonra Ingrid Daubechies, Ronald Coifman gibi uluslar arası araştırmacılar yöntemi geliştirerek bugünkü şekline getirmişlerdir.

Dalgacık dönüşümü Fourier dönüşümüne benzeyen ve fourier dönüşümün durağan olmayan sinyallerdeki eksiklerini gideren farklı bir dönüşüm yöntemidir. Bu analiz yöntemi gürültüye karşı daha az hassasiyet göstermekte ve durağan olmayan sinyallere rahatlıkla uygulanabilmektedir. Durağan olmayan işaretlerin spektral analizi için uygun olmasından dolayı dalgacık dönüşümünün diğer spektral analiz metotlarına göre üstünlükleri vardır. Dalgacık dönüşümünün en önemli avantajı, düşük frekanslar için geniş, yüksek frekanslar için dar olacak şekilde değişen pencere boyutlarının olmasıdır. Böylece, bütün frekans aralıklarında optimum zaman-frekans çözünürlüğü sağlanabilmektedir (Güler, İ. ve ark, 2004, Daubechies, I., 1990, Zhang, Y. ve ark 2001).

Dalgacık dönüşümü sürekli ve ayrık olmak üzere iki farklı şekilde incelenir. Sürekli dalgacık dönüşümünde ölçeklendirme ve dönüşüm parametrelerinin sürekli olarak değişiminden dolayı her bir ölçek için dalgacık katsayılarının hesaplanması zor ve zaman alıcı olmaktadır. Bu nedenle ayrık dalgacık dönüşümü daha sık kullanılmaktadır.

Dalgacık dönüşümü ile işaret belli sayıda ölçeklere ayrılır. Çoklu çözünürlük ayrışımı olarak isimlendirilen bu işlem, $x(n)$ işareti için Şekil 3.3'de gösterilmektedir. Şekil 3.3'de, ilk yüksek geçiren filtreye ($g[\cdot]$) ve alçak geçiren filtreye ($h[\cdot]$) ait olan örneklenmiş çıkışlar sırası ile ayrıntılı D_1 ve yaklaşık A_1 alt bantlarını oluşturur. A_1 yaklaşım bandı tekrar ayrıştır ve bu işlem Şekil 3.3'de görüldüğü gibi devam eder.



Şekil 3.3. Ayırık dalgacık dönüşümünün gerçekleştirilmesinde alt bantlara ayırışım; $g[n]$ yüksek geçiren filtre, $h[n]$ alçak geçiren filtre (Güler, İ. ve ark, 2004)

Dalgacık dönüşümü aşağıda verilen şartı sağlayan alçak geçiren filtre (h) ile belirtilebilir:

$$H(z)H(z^{-1}) + H(-z)H(-z^{-1}) = 1 \quad (3.7)$$

Burada $H(z)$, h filtresinin z -dönüşümüdür. Bu filtrenin tamamlayıcı yüksek geçiren filtresi (g) şu şekilde tanımlanabilir:

$$G(z) = zH(-z^{-1}) \quad (3.8)$$

Artan uzunluklar ile filtre dizisi (i indeksi ile) şu şekilde elde edilir:

$$\begin{aligned} H_{i+1}(z) &= H(z^{2^i})H_i(z) \\ G_{i+1}(z) &= G(z^{2^i})H_i(z), \quad i = 0, \dots, I-1 \end{aligned} \quad (3.9)$$

Burada başlangıç şartı $H_0(z) = 1$ 'dir. Bu, zaman domeninde şu şekilde ifade edilir:

$$\begin{aligned} h_{i+1}(k) &= [h]_{\uparrow 2^i} * h_i(k) \\ g_{i+1}(k) &= [g]_{\uparrow 2^i} * h_i(k) \end{aligned} \quad (3.10)$$

Burada, $[\cdot]_{\uparrow m}$ indeksi m faktörü ile yukarı örnekleme yapıldığını gösterir ve k eşit olarak örneklenmiş ayırık zamanı belirtir.

Normalize edilmiş dalgacık ve ölçek temelli fonksiyonlar $\phi_{i,l}(k)$, $\psi_{i,l}(k)$ şu şekilde tanımlanır:

$$\begin{aligned} \phi_{i,l}(k) &= 2^{i/2} h_i(k - 2^i l) \\ \psi_{i,l}(k) &= 2^{i/2} g_i(k - 2^i l) \end{aligned} \quad (3.11)$$

Burada, $2^{i/2}$ faktörü iç çarpım normalizasyonudur, i ölçeklendirme parametresi, l dönüşüm parametresidir. Ayırık dalgacık dönüşüm ayırışımı şu şekilde belirtilir:

$$\begin{aligned}
a_{(i)}(l) &= x(k) * \varphi_{i,l}(k) \\
d_{(i)}(l) &= x(k) * \psi_{i,l}(k)
\end{aligned}
\tag{3.12}$$

Burada $a_{(i)}(l)$ ve $d_{(i)}(l)$ sırası ile i çözünürlüğündeki yaklaşık katsayılar ve ayrıntılı katsayılarıdır (Güler, İ. ve ark, 2004, Daubechies, I., 1990, Zhang, Y. ve ark 2001).

Bu çalışmada biyometrik data Matlab kullanılarak dalgacık dönüşümüne tabi tutulmuşlardır. Dalgacık olarak Daubechies-1 (Haar wavelet) kullanılmıştır. Bu dönüşüm sonucu elde edilen katsayılar ağ eğitimi ve sınıflandırmada kullanılmışlardır.

Yine elde edilen sınıflandırma başarısını mukayese etmek için biyometrik veriye Ayrık Kosinüs Dönüşümü uygulanmış ve elde edilen katsayılar yine ağ eğitimi ve sınıflandırmada kullanılmışlardır.

Dalgacık dönüşümünün matematiksel tanımı;

$$W(a,b) = \frac{1}{\sqrt{a}} \int_{-\infty}^{\infty} f(t) \cdot \overline{\psi\left(\frac{t-b}{a}\right)} \cdot dt
\tag{3.13}$$

Bu denklemlerde $a > 0$, $b \in \mathbb{R}$ olmak üzere a , ölçekleme/yayılma katsayısını;

b dönüşüm/öteleme katsayısını;

$f(t)$ işareti;

ψ , dalgacık fonksiyonunu (ana dalgacığı);

$W(a,b)$ ise işaretin sürekli dalgacık dönüşümünü belirtir.

4 YAPAY SİNİR AĞLARI

YSA, biyolojik sinir ağlarından esinlenerek modellenmiş olup, onlardan çok daha basit bir yapıya sahiptir. O nedenle de anlaşılması daha kolaydır. Geliştirilen birçok YSA, biyolojik sinir ağlarının bilinen temel birkaç özelliğini (öğrenme kabiliyeti gibi) benzetmek (simulation) üzere geliştirilmiştir. Birtakım özellikler ise nörofizyolojik yaklaşımlar yerine mühendislik yaklaşımı ile geliştirilmektedir. Bir YSA'nın yapısını belirleyen bazı faktörler vardır. Yapay nöronlar veya mühendislik deyişiyle işlem elemanları, sinir ağının yapısal dağılışı şekli, ağın sahip olduğu öğrenme kuralı ve stratejisi bunların başında gelir. YSA mimarilerinin çeşitliliğine rağmen, benzer kısımlardan oluşurlar. Bir sinir ağının genel formunda, nörona benzeyen birime "işlem elemanı" denilmektedir. Ağ, bu elemanların bir araya gelmeleriyle meydana gelmektedir. Bu elemanlar genellikle tamamen veya kısmen bağlı sıralı katmanlar şeklinde düzenlenmişlerdir.

YSA, paralel dağılmış bir bilgi işleme sistemidir. Bu sistem genellikle tek yönlü işaret kanalları (bağlantılar) ile birbirine bağlanan işlem elemanlarından oluşur. Çıkış işareti bir tane olup isteğe göre çoğaltılabilir. YSA yaklaşımının temel düşüncesiyle, insan beyninin fonksiyonları arasında benzerlik vardır. Bu yüzden YSA sistemine "insan beyninin modeli" denilebilir. YSA çevre şartlarına göre davranışlarını şekillendirebilir. Girişler ve istenen çıkışların sisteme verilmesi ile kendisini farklı cevaplar verebilecek şekilde ayarlayabilir. Ancak son derece karmaşık bir içyapısı vardır.(Karlık, B., 2009)

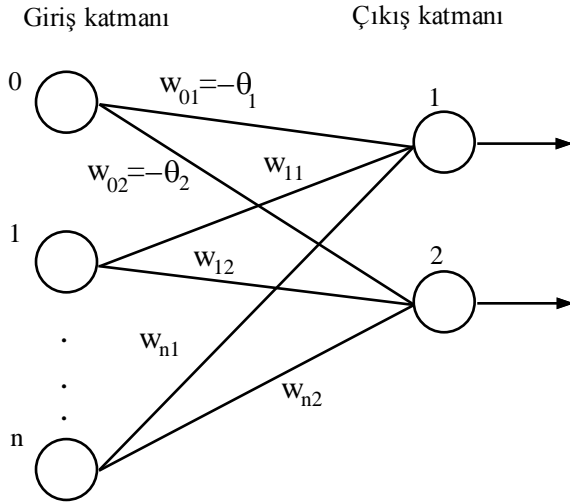
4.1 YSA nın Tarihsel Gelişimi

1940’larda Mc Culloch ve Pitts, bir biyolojik nöronun temel fonksiyonlarının basit bir eşik cihazı olarak modellenebileceğini göstermiştir (Mc Culloch, W.S. ve ark, 1943). YSA alanında yapılan ilk çalışmalar McCulloch ve Pitts tarafından yapılmıştır (McCulloch ve Pitts 1943). Bu araştırmacıların yayınladığı “Sinir Aktivitesindeki Düşüncelere Ait Bir Mantıksal Hesap” başlıklı makalede YSA konusunda ilk adım atılmıştır. Hızlı hesaplamaya yönelik ilk YSA çalışmaları 1950’li yıllarda başlamıştır. Basit nöron modellerine dayalı bir hesaplama modeli, 1950’lerde Frank Rosenblatt tarafından önerilmiş (Rosenblatt, F., 1959) ve ardından perceptron diye bilinen tek katmanlı ilk YSA modeli (Şekil 4.1) ortaya çıkmıştır (Rosenblatt, F. ,1962). 1960’lı yıllarda Widrow ve Hoff, bu basit nöron modellerini kullanarak öğrenebilen ilk adaptif sistemler üzerinde çalışmış ve delta kuralı diye bilinen; gerçek çıkış ile istenen çıkış arasındaki farka eşit bir hata terimi kullanarak bağ ağırlıklarının değiştirildiği bir öğrenme kuralını ortaya koymuşlardır (Widrow, B. ve ark, 1960).

1969-1982 yılları arasındaki çalışmalarda ise teori artık oturmuş ve 1982’de J.J. Hopfield tarafından yayınlanan “Neural Networks and Physical Systems” adlı çalışma ile çağdaş YSA devri başlamıştır (Hopfield, J.J., 1982). Bu çalışmada Hopfield, nöronların karşılıklı etkileşimlerine dayanan bir nöral hesaplama modeli önermiştir. Bu model, bir enerji fonksiyonunu, alabileceği en az değere indiren 1. mertebe lineer olmayan diferansiyel denklemlerden oluşmuştur. Hopfield; ağ seviyesinde, tek tek nöron seviyesinde var olmayan hesaplama kapasitesinin bulunduğunu öne sürmüştür. Bu tür YSA ya, “Hopfield Ağı” denilmektedir. Hopfield’in geri beslemeli YSA modelini ortaya atması (Hopfield, J.J., 1987) ve bunun pratik optimizasyon problemlerinde kullanılabilirliğini göstermesi (Hopfield 1985), YSA konusundaki çalışmaları yeniden hızlandırmıştır.

1986’da Grossberg, ART yani Uyarlanabilir Rezonans Teorisi (Adaptive Resonance Theory) adında bir YSA yapısını geliştirmiştir (Grossberg, S. 1986). O sıralarda Kohonen’de “kendi kendini düzenleyen nitelik haritasını (self-organizing maps) geliştirmiştir (Kohonen, T. 1984). Bu YSA modeli, nümerik aerodinamik akış hesaplamaları için çoğu algoritmik yöntemden daha etkili olmuştur.

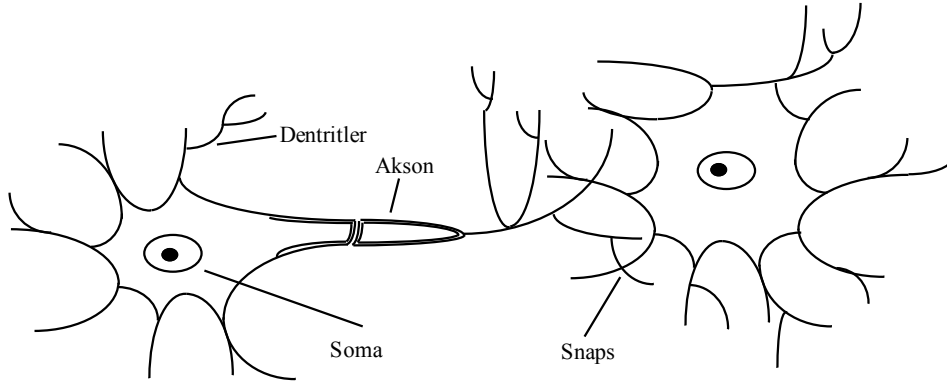
1986'da Rumelhart ve arkadaşları "Parallel Distributed Processing" (paralel dağılımlı işleme) adlı kitaplarında, ileri beslemeli (feed-forward) ağlarda yeni öğrenme modeli olan hatanın geriye yayılması algoritmasını (backpropagation algorithm) geliştirerek, daha önce bu konuda Minsky ve arkadaşları tarafından iddia edilen aksaklıkların aşılabileceğini göstermişlerdir (Rumelhart, D.E. ve ark, 1986). Bugün endüstride birçok YSA uygulamasında bu öğrenme yöntemi ile bunun değişik varyasyonları kullanılmaktadır. Back-propagation algoritması, kullanımı çok yaygın olan ve öğrenilmesi kolay bir ağıdır. Ayrıca önceki bölümde anlatıldığı gibi biyolojik işaretlerin sınıflamalarında tercih edilen bir ağ yapısıdır. İç içe girmiş konveks yapıdaki kümelerin sınıflamasını rahatlıkla yapmaktadır. Bundan dolayı bu tez çalışmasında da bu algoritma kullanıldı. (Karlık, B., 2009)



Şekil 4.1. Perceptron modeli

4.2 Biyolojik Sinir Ağları ve Nöronun Biyolojik Yapısı

Biyolojik sinir ağları başlangıçta sadece nörofizyolojistler tarafından araştırılmıştır. Araştırmaların konusu, genel olarak her bir nöronun yapısı ve birbirleri ile nasıl ilişkide buldukları üzerine olmuştur. Yapay sinir ağlarının oluşturulmasında bu çalışmaların faydası inkar edilemez. Sinir sisteminin en basit yapısı olan bir nöronun analizi bu nedenle önem arz eder. Nöron, sinir ağlarının en temel elemanlarından birisi olup sinir sistemi içerisindeki fonksiyon ve görevlerine göre değişik şekil ve büyüklükte olabilir. Bütün nöronların ortak bazı özellikleri bulunmaktadır (Şekil 4.2). Nöron, soma adı verilen hücre gövdesi, bir ucunda bir grup liflere benzer “dentrit” adı verilen ve hücreye diğer hücrelerden veya dış dünyadan bilgiler (sinyaller) getiren bağlantı elemanları, diğer ucunda ise tek bir life benzer “akson” adı verilen ve hücrelerden diğerlerine veya dış dünyaya bilgiler taşıyan bağlantı elemanından müteşekkildir. Bu akson daha sonra diğer hücrelerle birleşme esnasında dağınık dallara ayrılmaktadır. Bu iki uçtaki bağlantı noktalarının elektrofizyolojik olarak hücrelerdeki bilgileri işlemede önemli yeri vardır. Dentrit tarafından alınan sinyaller hücrede birleştirilerek bir çıkış darbesi üretilip üretilmeyeceğine karar verilir. Eğer bir iş yapılacaksa üretilen çıkış darbesi aksonlar tarafından taşınarak diğer nöronlarla olan bağlantılara veya terminal organlara iletilir. (Karlık, B., 2009)



Şekil 4.2. Biyolojik nöron/sinir hücresinin şematik yapısı

Nöronların birbirleri ile elektrik sinyalleri şeklinde haberleştiği belirlenmiştir. Sinyaller bir nöronun aksonundan diğerinin dentritine gönderilir. Bir akson birden fazla dentrit ile ilişkiye girebilir. Bu bağlantının yapıldığı yere “snaps” (Şekil 4.2) denir. Nöronlar elektrik sinyalini hücre duvarındaki voltajı değiştirerek üretirler. (Karlık, B., 2009)

Bir hücre diğer bir hücreye elektrik enerjisini bu kimyasal iyonlar vasıtasıyla transfer eder. Bazı iyonlar elektrik ve manyetik kutuplaşmaya sebep olurken bazıları kutuplaşmadan kurtulup hücre zarını açarak iyonların somaya geçmesini sağlar. Zaten sinyallerin bir hücreden diğerine akmasını sağlayan da bu kutuplaşmanın azalması olayıdır. Sinyaller hücrenin etkinliğini (dürtüsünü) belirler. Bir hücrenin etkinliği, hücreye gelen snaps sayısı, snapslardaki iyonların konsantrasyonu, snapsın sahip olduğu güç olmak üzere üç faktöre bağlıdır. Bir nöron sahip olduğu dürtü miktarınca diğer hücreleri etkiler. Bazı hücreler diğerlerinin dürtülerini pozitif yönde, bazı hücreler de negatif yönde etkiler. İnsan beyni bu şekilde çalışan milyonlarca nöronun bir araya gelmesinden oluşur. Beyinde korteks denilen bölgede her nöronun bir karşılığı vardır. Bir nöronun çıkışı ona bağlı olan bütün nöronlara iletilir. Fakat korteks, işin yapılabilmesi için hangi nöron harekete geçecekse sadece ona komut gönderir.(Karlık, B., 2009)

Biyolojik beynin en önemli özelliklerinden birisi de öğrenme olayıdır. Öğrenilen her yeni bilgi, hemen beynin fonksiyonlarını etkileyerek davranışlarda da kendini gösterir. YSA'nın geliştirilmesinde bu özellik esas teşkil eder.

4.3 YSA'nın Yapısı ve İşlem Elemanı

Beynin düşünme ve karar verme özellikleri, bilim adamlarını üzerinde çalışmaya zorlamış ve beynin nörofiziksel yapısından esinlenerek matematiksel modeli çıkarılmaya çalışılmıştır. Beynin bütün davranışlarını tam olarak modelleyebilmek için fiziksel bileşenlerinin doğru olarak modellenmesi gerektiği düşüncesi ile çeşitli yapay hücre ve ağ modelleri geliştirilmiştir. Beynin tüm fonksiyonları bilinmediği için belli oranlarda benzetim yapılarak bir karar verme algoritması geliştirilmiştir. Böylece Yapay Sinir Ağları (YSA) denen yeni ve günümüz bilgisayarlarının algoritmik hesaplama yönteminden farklı bir bilim alanı ortaya çıkmıştır. Yapay sinir ağları; yapısı, bilgi işleme yöntemindeki farklılık ve uygulama alanları nedeniyle çeşitli bilim dallarının da kapsam alanına girmektedir.

Genel anlamda YSA, beynin bir işlevi yerine getirme yöntemini modellemek için tasarlanan bir sistem olarak tanımlanabilir. YSA, yapay sinir hücrelerinin birbirleri ile çeşitli şekillerde bağlanmasından oluşur ve genellikle katmanlar şeklinde düzenlenir. Donanım olarak elektronik devrelerle ya da bilgisayarlarda yazılım olarak gerçekleştirilebilir. Beynin bilgi işleme yöntemine uygun olarak YSA, bir öğrenme sürecinden sonra bilgiyi toplama, hücreler arasındaki bağlantı ağırlıkları ile bu bilgiyi saklama ve genelleme yeteneğine sahip paralel dağılmış bir işlemcidir. Öğrenme süreci, arzu edilen amaca ulaşmak için YSA ağırlıklarının yenilenmesini sağlayan öğrenme algoritmalarını ihtiva eder.

Örüntü tanıma/sınıflandırılmasında YSA kullanılmasının nedeni, sonuçta iyi kalitede bir dağılıma ulaşması ve işlemin doğrusal olmayışıdır (Merényi, E. ve ark, 1996). Yine paralel programlama yapısında uygulanabilirliği test edilmiş olduğu için büyük boyutlu verilerde çalışabilmektedir (özellikle uygun donanım ile de desteklendiği zaman). YSA'nın örüntü tanıma özellikle karmaşık ve gürültülü örüntülerde başarıyla uygulanabilirliği ispatlanmıştır (Abd-El-Reheim M.A-E-A., 1999).

4.4 Öğrenme Kuralları

Bilginin kurallar şeklinde açıklandığı klasik uzman sistemlerin tersine, YSA gösterilen örnekten öğrenerek kendi kurallarını oluşturur. Öğrenme; giriş örneklerine veya (tercihen) bu girişlerin çıkışlarına bağlı olarak ağırlık bağlantı ağırlıklarını değiştiren veya ayarlayan öğrenme kuralı ile gerçekleştirilir. Ağ, ağırlık bileşendeki değerler değiştirilerek örneklere göre eğitilir ve herhangi bir örnek ile yeniden karşılaştırıldığında uygun cevabı üretir. Öğrenmede iyi bir model kullanıp, ağırlıkların bu modele göre değiştirilmesi esastır. Bu ağırlıkların değiştirilmesi belli bir hata oranına kadar itere edilir. En son ağırlık değerleri test etmede kullanmak için kaydedilir. Danışmansız öğrenmede her giriş işareti için istenen çıkış sisteme tanıtılır ve YSA giriş/çıkış ilişkisini gerçekleştirene kadar kademe kademe kendini ayarlar. Günümüzde kullanılan birçok öğrenme kuralı vardır. Bilinen ve en çok kullanılan öğrenme kuralları şunlardır:

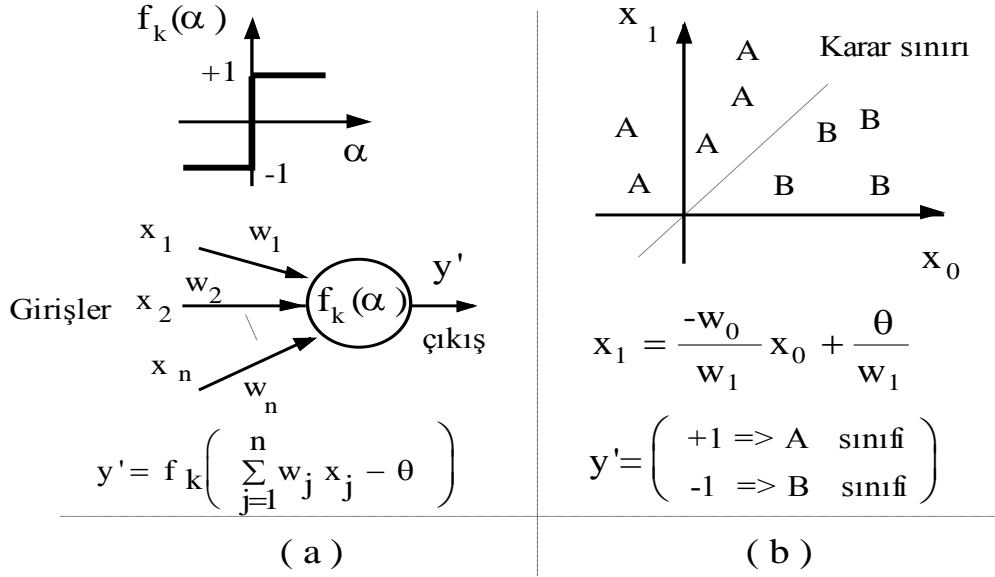
- Rastlantısal (Hebb) öğrenme kuralı
- Performans (Widrow ve *ADALINE*) öğrenme kuralı
- Yarışmalı (Kohonen) öğrenme
- Filtreleme (Grossberg - Volan)
- Spotitemporal öğrenme
- Genelleştirilmiş delta kuralı öğrenme

4.4.1 Perceptron (İdrak-algılayıcı)

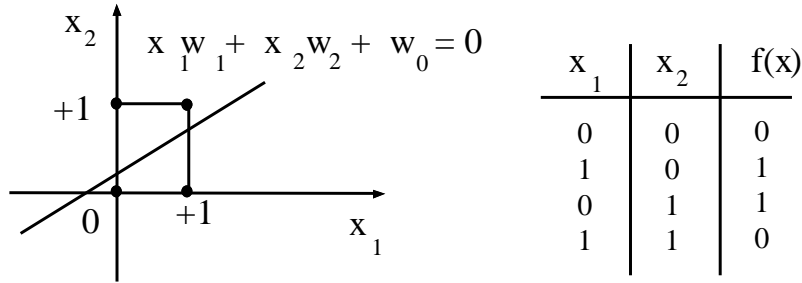
Perceptron (idrak-algılayıcı) ağı, ilk defa 1943 yılında McCulloch ve Pitts tarafından ortaya atılmıştır (McCulloch ve Pitts 1943). Şekil 4.3-a'da tek katmanlı idrak (perceptron) yapısı görülmektedir.

$$\sum_{j=1}^n w_j x_j - \theta = 0; \quad n \text{ boyutlu uzayda, } n-1 \text{ boyutlu bir düzlem belirler.}$$

Ağın ürettiği çıkış istenen çıkıştan farklı ise, hata azaltılacak yönde hareket edilmektedir. Hata sıfır olduğunda w değişimi durmaktadır. Bu basit ağ, ilk defa geliştirildiğinde basit örnekleri öğrenme kabiliyetlerinden ötürü oldukça ilgi toplamıştır.(Karlık, B., 2009)



Şekil 4.3. (a) Tek katmanlı perceptron yapısı, (b) Bir analog giriş vektörünü A ve B ile gösterilen iki sınıfa ayıran tek katmanlı bir perceptron


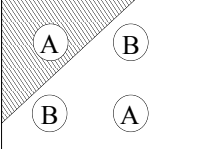
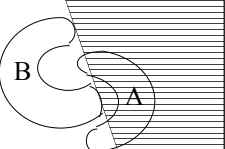
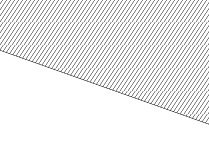

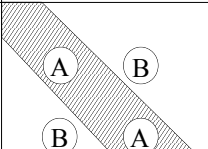
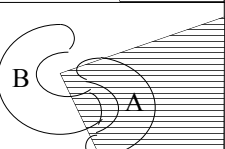
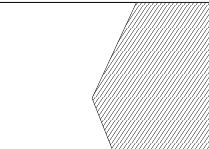

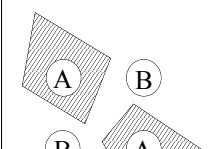

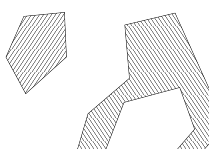


Şekil 4.4. Lineer ayrılabilirliğin gösterimi

Şekil 4.3-b’de bir girişin iki sınıftan (A veya B) hangisine ait olduğuna karar veren bir perceptron görülmektedir. Bir tek düğüm giriş elemanlarının ağırlıklı toplamını hesaplar, bir eşik θ ’yı çıkarır ve y' çıkışı +1 veya -1 olacak şekilde sonucu bir süreksizlikten geçirir. Karar verme kuralı ise; çıkış +1 ise A sınıfı ile, -1 ise B sınıfı ile gösterilir. Yani bu ilk perceptron modeli, giriş bilgisinin mevcut iki sınıftan hangisine eşit olabileceğini bulacak şekilde eğitilen basit bir ağıdır. (Karlık, B., 2009)

Daha sonra 1960 yıllarında F. Rosenblatt yukarıdaki ağ tipini biraz daha geliştirmiştir (Rosenblatt 1958). Fakat, Minsky ve Papert, bu tek katmanlı perceptronun XOR işlemini gerçekleştiremediğini ispatlamışlardır. Şekil 4.3 ve 4.4'den anlaşılacağı gibi 0'ların bir tarafta, 1'lerin bir tarafta ayrılacağı şekilde bir bölge oluşmamaktadır. XOR gibi 3 veya daha fazla sınıfa ihtiyaç duyulan problemleri çözmek için yapılması gereken işlem ise tek katmanlı ağa yeni katmanlar eklemektir. Eşik bağlarıyla oluşturulan karar bölgesi şeklinin karmaşıklığı, sadece eklenmiş olan katmanların sayısı ile sınırlıdır. Şekil 4.5'de gösterildiği gibi içbükey ayrılabilir fonksiyonlar gerçekleştirilebilir (Karlık, B., 2009).

Şekil 4.5'de tek ve çok katmanlı perceptronların sınırlayabileceği (ayırabileceği) karar bölgeleri görülmektedir. Taranmış bölgeler A sınıfına ait karar bölgelerini göstermektedir. Tek katmanlı perceptron, bir düzlemi ikiye ayıran karar bölgeleri oluşturur. İki katmanlı bir perceptron, açık ve kapalı konveks bölgeler oluşturabilir. Üç katmanlı perceptron ise, keyfi derecelerde kompleks karar bölgelerini ayırabilir. Buradan anlaşılmalıdır ki; perceptron gibi ileri yayımlı ağlarda, iki adet gizli katmandan fazlasına gerek yoktur.

Yapı	Karar bölgeleri tipi	XOR problemi	Bölgelere dayalı sınıflar	En iyi ayırdığı bölge şekilleri
Tek katman  (a)	Doğrusal ayrıştırabilir, XOR işlevini gerçekleyemez			
İki katman  (b)	Rastgele bölgeler Konveks açık veya kapalı bölgeler			
Üç katman  (c)	Komplekslik derecesi düğüm sayısına bağlı olarak sınırlıdır			

Şekil 4.5. Tek ve çok katmanlı perceptronların sınırlayabileceği karar bölgeleri

Üç katmanlı perceptronlarda (giriş katmanı dahil değil) kullanılacak düğüm sayılarının belirlenmesine gelince; karar bölgeleri arasında bağlantı yoksa veya bu bölgeler bir konveks alan ile oluşturulamıyorsa, ikinci katmandaki düğüm sayısı bir'den büyük olmalıdır. En kompleks durum için, ikinci katmandaki düğümlerin sayısı, giriş dağılımlarındaki ayrı bölgelerin sayısı kadar olmalıdır (Karlık, B., 2009). Benzer şekilde birinci katmandaki düğüm sayıları da her bir ikinci katman düğümünce oluşturulan her bir konveks bölge için üç veya daha fazla kenar sağlanacak şekilde yeterli sayıda olmalıdır. Bununla beraber, Gutierrez ve arkadaşları, değişik perceptron ağlarının ihtiyacı olan düğüm sayıları hakkında araştırma yapmışlar ve çok fazla düğümün de, çok az sayıda olduğu gibi zararlı etkisi olduğunu bulmuşlardır (Karlık, B., 1994).

Tek katmanlı perceptron uygulanan her eğitim seti modelinin en önemli özelliği, lineer biçimde dağılmak zorunda olmasıdır. Eğer bu doğrudur; Rosenblatt (Rosenblatt, F., 1962), perceptron ağının kararlı olacağını göstermiştir (Örneğin ağırlıklar iterasyonla yakınsar).

4.4.2 Perceptron Öğrenme Algoritması

Verilen giriş bilgisinin mevcut iki sınıftan hangisine eşit olabileceğini bulacak şekilde eğiten en basit YSA yapısı olan perceptronun öğrenme algoritması şöyledir (Pao, Y.H. 1989, Karlık, B. 2009):

ADIM 1: öncelikle, başlangıç ağırlık değerleri ile eşik seviyesi (θ) değerleri rastgele (random) olarak belirlenir

$$W(i)(0), \quad 0 \leq i \leq N-1$$

ADIM 2: Ağa öğretilmesi gereken istenen (hedef) bilgi değerleri verilir. Ayrıca öğrenme oranı (α) set edilir.

$$(X,D), [X: x(0), x(1), \dots, x(n)] \text{ ve } [D: d(0), d(1), \dots, d(n)]$$

$$0 < \alpha < 1 \text{ (1'den küçük olmalı)}$$

Sınıflandırma problemi için istenen çıkışlar, lojik değerler; 1 (küme içinde) ve 0 (küme dışında) olarak alınır.

ADIM 3: İlk adımdaki rastgele ağırlık değerleri, eşik seviyesi ve mevcut giriş değerleri kullanılarak gerçek çıkış değeri (ağ çıkışı) belirlenir.

$$y(n)=f[w(i)(n)*x(i)(n)- \theta]$$

ADIM 4: Eğer gerçek çıkış değerleri istenen çıkış değerlerine uymuyorsa, istenen çıkış değerinin uyumu amacıyla ağırlık değerleri şu şekilde değiştirilir ve 3. ADIM'a dönülür. (Aynı işlem varsa eşik değer için de tekrarlanır).

$$W(i)(n+1)=w(i)(n)+\alpha*[d(n)-y(n)]*x(i)(n)$$

Eğer gerçek çıkış değerleri istenen çıkış değerine uyuyorsa (hata yoksa), ağ istenen bilgiyi öğrenmiş demektir. Bu durumda ağırlık değerlerinin değişimi durdurulur.

OR (veya) işlemini basit idrak algoritmasını kullanarak çözümü (a) ve Aynı algoritmayı kullanarak XOR problemini çözümü (b) aşağıda verilmiştir.

OR		
A	B	Q
0	0	0
0	1	1
1	0	1
1	1	1

(a)

EXOR		
A	B	Q
0	0	0
0	1	1
1	0	1
1	1	0

(b)

$$0*W(1)+0*W(2)< \theta$$

$$0*W(1)+1*W(2)> \theta$$

$$1*W(1)+0*W(2)> \theta$$

$$1*W(1)+1*W(2)> \theta$$

Ağırlık değerlerinden her ikisi de pozitif (+) ise öğrenme işlemi tamamlanmış olur.

$$0*W(1)+0*W(2)< \theta$$

$$0*W(1)+1*W(2)> \theta$$

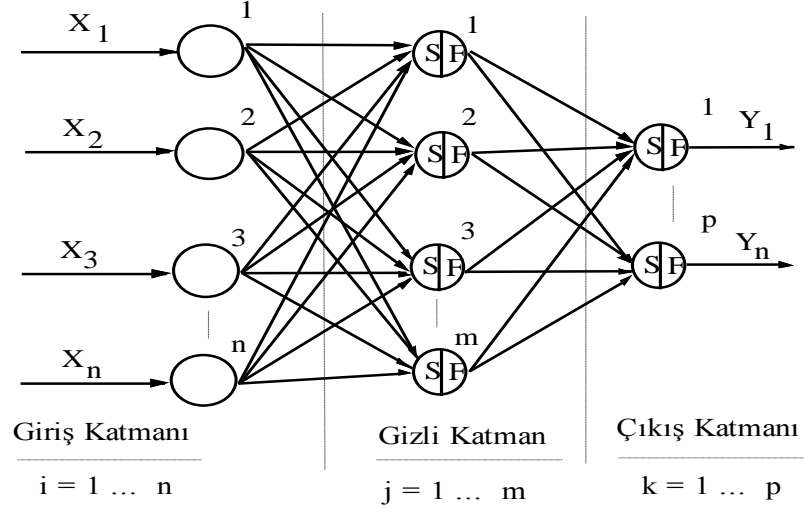
$$1*W(1)+0*W(2)> \theta$$

$$1*W(1)+1*W(2)< \theta$$

Girişlere bağlı olarak, ağırlık değerleri tek-tek eşik değerinden büyük olurken aynı anda eşik değerinden küçük olması imkânsızdır. Sonuç olarak; tek katmanlı idrak ağı EXOR problemini çözmede yetersiz kalmaktadır.

4.4.3 Çok Katmanlı İdrak

Çok katmanlı Perceptron (idrak-algılayıcı) giriş ve çıkış katmanları arasında birden fazla katmanın kullanıldığı ileri yayımlı (feed-forward) YSA sistemidir. Gizli katman (hidden layer) olarak isimlendirilen bu ara katmanlarda, düğümleri giriş ve çıkış katmanlarına doğrudan bağlı olmayan işlem elemanları vardır. Şekil 4.6’da çok katmanlı perceptronun genel yapısı verilmiştir. Çok katmanlı perceptronlar, tek katmanlı perceptronlara ait sınırlamaların birçoğunu ortadan kaldırmalarına rağmen, önceleri etkin eğitime algoritmalarının mevcut olmamasından ötürü uygulamaları yaygın değildi. Ancak yeni eğitime algoritmalarının geliştirilmesi ile bu durum değişmiştir (Rumelhart, D.E., ve ark. 1986).

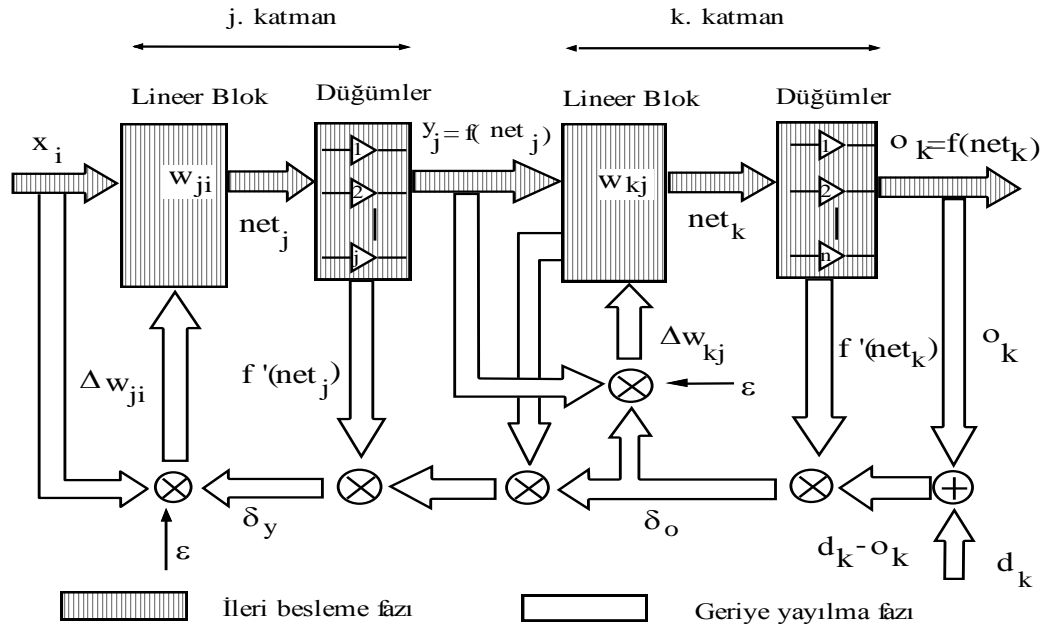


- F** : Nöron çıkış değerini veren Transfer/Eşik Fonksiyonu
S : Nöron NET değeri için, (Giriş x Ağırlık) değerlerinin toplamı
○ : Sadece veri girişinin sağlandığı işlem elemanı
⊕ : S & F hesaplamalarının gerçekleştirildiği işlem elemanı

Şekil 4.6. (n:m:p) mimarisine sahip bir çok katmanlı perceptronun yapısı

Çok katmanlı ağlarda, veriler giriş katmanı tarafından kabul edilirler. Ağ içinde yapılan işlemler sonucunda çıkış katmanında oluşan sonuç değer, istenen cevap ile karşılaştırılır. Bulunan cevap ile istenen cevap arasında herhangi bir ayrılık varsa, ağırlıklar bu farkı azaltacak şekilde yeniden düzenlenir. Girişteki değer, ağırlıklar uygun noktaya ulaşana kadar değişmez. Hesaplanan çıkışlar, istenilen cevaplarla karşılaştırılarak sonuçta gerekirse hata belirtilir. Hata işareti gizli işlem elemanlarından çıkış birimine olan ağırlıkları değiştirmekte kullanılır. Ama bunu yaparken giriş katmanından gizli katmana gelenin değiştirilip değiştirilemediğini düşünmek gerekir. Gizli birimlerden ne tür bir çıkış istendiği bilinmeyeceği için gizli birimlerin çıkışında hata işareti verilmesi kolay bir şey değildir. Bunun yerine her birimin çıkış biriminin hatalarına olan etkisi bilinmelidir. Bu ise hatalı birim için gizli birime bağlı olan çıkış birimlerinin hata işaretlerinin toplamı alınarak yapılır.

Çok sayıda gizli katmana sahip sistemlerde, her sistemin hata işaretleri, bir önceki katmanın düzeltilmiş işlemlerinden çıkartılarak işlem tekrarlanır. Sonuç olarak, ağırlık düzeltme işlemi çıkış seviyesine bağlı ağırlıklardan başlar ve işlem ters yönde, giriş seviyesine varana kadar devam eder. Sonuçta sistem hatalar yapar, ama bu hatalardan bir şeyler öğrenip isteneni bulana kadar işleme devam eder. Bu yönteme “hatanın geriye yayılması algoritması” (back-propagation algorithm) denilir (Şekil 4.7).



Şekil 4.7. Hatanın geriye yayılması algoritmasının blok diyagramı

4.4.4 Hatanın Geriye Yayılması Algoritması ve Genelleştirilmiş Delta Kuralı

Hatanın geriye yayılması (Back-propagation) eğitme algoritması, çok katmanlı, ileri yayımlı bir perceptrondan elde edilen çıkışlar ile eldeki hedef çıkışlar arasındaki hataların karesinin ortalamasını minimum yapmak için geliştirilmiş iteratif bir gradyan algoritmadır ve eğitme işlemi için genelleştirilmiş delta kuralını (Generalized Delta Rule) kullanır.

Şekil 4.7’de blok diyagramı gösterilen algoritma, ana hatlarıyla şöyledir: Ağ mimarisi tanımlanır ve ağırlıklar bazı rastgele küçük sayılar ile başlatılarak, ağa ilk giriş sunulur. Burada m-boyutlu giriş örnekleri girildiğinde; $x_i = [x_1, x_2, \dots, x_m]^T$ dir. Benzer şekilde istenilen n-boyutlu çıkış örneklerini ise; $d_k = [d_1, d_2, \dots, d_n]^T$ belirtir. x_i değerleri i katmanındaki nöronların çıkış değerleri ise, j katmanındaki bir nörona gelecek olan toplam giriş,

$$net_j = \sum_{i=1}^m w_{ji} \cdot x_i \quad (\text{i. düğümünden j. düğüme}) \quad (4.1)$$

olur. Gizli katmandaki j nöronunun çıkışı (transfer fonksiyonu çıkışı);

$$y_j = f_j (net_j) , \quad j=1,2,\dots,J \quad (4.2)$$

dir ve burada f_j transfer fonksiyonudur. Çıkış katmanındaki k nöronuna gelecek olan toplam giriş ise,

$$net_k = \sum_{j=1}^J w_{kj} \cdot y_j \quad (4.3)$$

dir. Yine çıkış katmanındaki bir k nöronun lineer olmayan çıkışı;

$$o_k = f_k (net_k) , \quad k = 1, 2, \dots, n \quad \text{olur.} \quad (4.4)$$

Ağdan elde edilen çıkış, asıl çıkış ile karşılaştırılır ve e_k hatası aşağıdaki gibi hesaplanır;

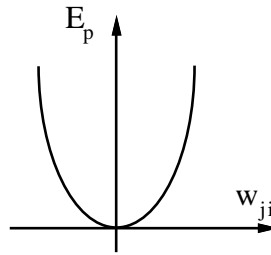
$$e_k = (d_k - o_k) \quad (4.5)$$

Burada d_k ve o_k sırasıyla çıkış katmanındaki herhangi bir k nöronunun hedef (istenilen) ve ağdan elde edilen (asıl) çıkışlardır. Dikkat edilmesi gereken husus, böyle bir karşılaştırmanın sadece ağın çıkış katmanı için mümkün olmasıdır. Böylece çıkış katmanıyla olan bağlar için ağırlık ayarlaması öncelikle göz önüne alınır. Her bir örnek (pattern) için toplam karesel hata;

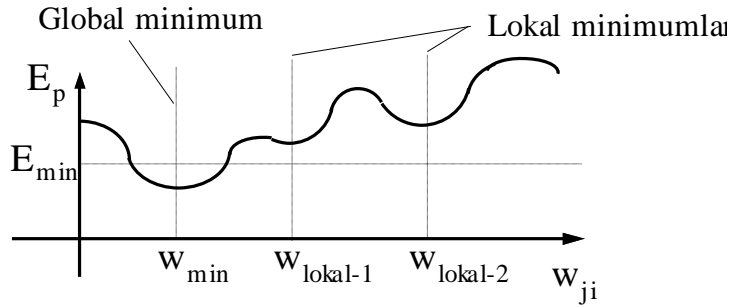
$$E = \frac{1}{2} \sum_k (d_k - o_k)^2 \quad (4.6)$$

olur. Hata fonksiyonu Şekil 4.8 ve Şekil 4.9’de gösterildiği gibidir. Gizli katman olduğu zaman; hata düzeyi Şekil 4.8’de olduğu gibi sadece bir minimumdan oluşmaz, Şekil 4.9’daki gibi çeşitli minimumlar oluşur. Bu minimumlar lokal ve global olarak sınıflandırılırlar. Lokal minimumlar birden fazla olabilir. Global minimum, hatanın en küçük olduğu minimumdur. Öğrenme sırasında global minimuma ulaşmak amaçlanır. Ağırlıkların değişimi;

$$\Delta w_{kj} = -\varepsilon \frac{\partial E}{\partial w_{kj}} \quad (4.7)$$



Şekil 4.8. Gizli katmanı olmayan ağırlık hata fonksiyonu



Şekil 4.9. Gizli katmana ait ağırlık hata fonksiyonu

Burada ε , öğrenme oranı katsayısı adı verilen küçük değerde pozitif bir sayıdır (çoğunlukla 0.01 ile 0.9 arasında seçilir). Bu öğrenme oranı, bir sayısal optimizasyon algoritmasındaki adım boyutu parametresine benzemektedir. (4.7)’de eşitliğin sağ tarafı açılırsa (Rumelhart, D.E., McClelland, J.L., 1986), şu sonuç elde edilir:

$$\frac{\partial E}{\partial w_{kj}} = \frac{\partial E}{\partial net_k} \frac{\partial net_k}{\partial w_{kj}} \quad (4.8)$$

$$\frac{\partial net_k}{\partial w_{kj}} = \frac{\partial}{\partial w_{kj}} \sum_j w_{kj} \cdot y_j = \sum_j \frac{\partial w_{kj} \cdot y_j}{\partial w_{kj}} = y_j \quad (4.9)$$

Amaç, uygun w seçimiyle, $E = \sum_p E_p$ toplam hatayı yeterince küçük yapmaktır. Bu amacı gerçekleştirmek için, bir $p \in P$ örneksi art arda ve rastgele biçimde seçilir. k . nöronda oluşan ve “delta” adı verilen hata işareti,

$$\delta_o = -\frac{\partial E}{\partial \text{net}_k} \quad (4.10)$$

(4.19) ve (4.20)'yi (4.18)'de yerine konulursa,

$$\frac{\partial E}{\partial w_{kj}} = -\delta_o \cdot y_j \quad (4.11)$$

ve, (4.21) ifadesi (4.17)'de yerine konulursa,

$$\Delta w_{kj} = \varepsilon \delta_o y_j \quad (4.12)$$

elde edilir. E_p değerini düşürmek demek, ağırlığı $\delta_o y_j$ 'ye bağlı olarak değiştirmek demektir. Buna “Delta kuralı” denir. Önceki gizli katmandaki tüm nöronlar ile çıkış katmanındaki k . nöron arasındaki bağ mukavemetleri, (4.12)'deki Δw_{kj} miktarı ile ayarlanır. (4.10) denklemini kısmi türevlerine ayrılırsa,

$$\delta_o = -\frac{\partial E}{\partial \text{net}_k} = -\frac{\partial E}{\partial o_k} \frac{\partial o_k}{\partial \text{net}_k} \quad (4.13)$$

$$\frac{\partial E}{\partial o_k} = -(d_k - o_k) \quad (4.14)$$

olup; k . nöron çıkışının lokal hatasını verir.

$$\frac{\partial o_k}{\partial \text{net}_k} = f'_k(\text{net}_k) \quad (4.15)$$

Son iki formül (4.13)'de yerine konulursa,

$$\delta_o = (d_k - o_k) f'_k(\text{net}_k) \quad (4.16)$$

bulunur. Bu son terim (4.12)'de yerine konulduğunda k nöronu için;

$$\Delta w_{kj} = \varepsilon (d_k - o_k) f'_k(\text{net}_k) y_j \quad (4.17)$$

olur. Eğer ağırlıklar çıkış nöronlarını doğrudan etkilemiyorsa (arada gizli katman varsa), (4.12)'e benzer biçimde delta kuralı bu gizli katman için de şu şekilde uygulanır.

$$\begin{aligned}
\Delta w_{ji} &= -\varepsilon \frac{\partial E}{\partial w_{ji}} \\
&= -\varepsilon \frac{\partial E}{\partial net_j} \frac{\partial net_j}{\partial w_{ji}} = -\varepsilon \frac{\partial E}{\partial net_j} x_i
\end{aligned} \tag{4.18}$$

$$\begin{aligned}
&= -\varepsilon \left(-\frac{\partial E}{\partial y_j} \frac{\partial y_j}{\partial net_j} \right) x_i = \varepsilon \left(\frac{\partial E}{\partial y_j} \right) f'_j(net_j) x_i \\
\Delta w_{ji} &= \varepsilon \delta_y x_i
\end{aligned} \tag{4.19}$$

Bununla birlikte, $\partial E / \partial y_j$ faktörü doğrudan geliştirilemez. Özel olarak çıkış katmanına etkisi,

$$-\frac{\partial E}{\partial y_j} = -\sum_k \frac{\partial E}{\partial net_k} \frac{\partial net_k}{\partial y_j} = \sum_k \left(-\frac{\partial E}{\partial net_k} \right) \frac{\partial}{\partial y_j} \sum_m w_{km} y_j = \sum_k \left(-\frac{\partial E}{\partial net_k} \right) w_{kj} = \sum_k \delta_o w_{kj}$$

şeklinde bulunur. Bu durumda;

$$\delta_y = f'_j(net_j) \sum_k \delta_o w_{kj} \tag{4.20}$$

olur. Kısaca özetlenecek olursa; (4.19) formülü şayet j. nöron, çıkış katmanı nöronuysa (3.16)'e benzer biçimde,

$$\delta_y = (d_y - o_y) f'_j(net_j) \tag{4.21}$$

olur. Eğer j nöronları gizli katmana ait nöronlar ise, o zaman (4.21) denklemi kullanılır. Transfer (eşik) fonksiyonu olarak sigmoid fonksiyonu kullanıldığında,

$$f(net_j) = y_j = \frac{1}{1 + e^{-net_j}} \tag{4.22}$$

olur. Bu ifadenin türevi alınıp, gerekli sadeleştirme yapıldığında,

$$f'(net_j) = \frac{1}{1 + e^{-net_j}} \frac{1 + e^{-net_j} - 1}{1 + e^{-net_j}} \tag{4.23}$$

$$\frac{\partial y_j}{\partial net_j} = y_j(1 - y_j) \tag{4.24}$$

bulunur. Benzer işlem k katmanı için de yapılırsa;

$$\frac{\partial o_k}{\partial net_k} = f'_k(net_k) = o_k(1 - o_k) \tag{4.25}$$

elde edilir. (4.25) denklemi (4.16)'da, (4.24) denklemi de (4.20)'da yerine konursa, (4.16) ve (4.20)'deki delta ifadeleri aşağıdaki gibi olur. (δ_o çıkış katmanı, δ_y : gizli katman elemanları içindir);

$$\delta_o = (d_k - o_k) o_k (1 - o_k) \quad (4.26)$$

$$\delta_y = y_j (1 - y_j) \sum_k \delta_o w_{kj} \quad (4.27)$$

w_{ji} yi öğrenme durumunda her eğitime örneklerinin seti için Δw_{ij} yi hesaplamak gerekir. Öğrenme oranı ε , hızlı öğrenmeyi sağlar; fakat dalgalanmalara sebep olabilir. Rumelhart ve McClelland (Karlık, B. ve ark, 2002) (4.17) ve (4.19) ifadelerine bir tür momentum terimi dahil ederek bir değişim öngörmüşlerdir. Bu durumda;

$$\Delta w_{ji}(n+1) = \varepsilon \delta_y x_i + \alpha \Delta w_{ji}(n) \quad (4.28)$$

$$\Delta w_{kj}(n+1) = \varepsilon \delta_o y_j + \alpha \Delta w_{kj}(n) \quad (4.29)$$

olarak yazılabilir. Burada n , öğrenme döngülerinin (iterasyon) sayısını gösterir. Momentum terimi olan α , küçük değerde pozitif bir sayıdır.

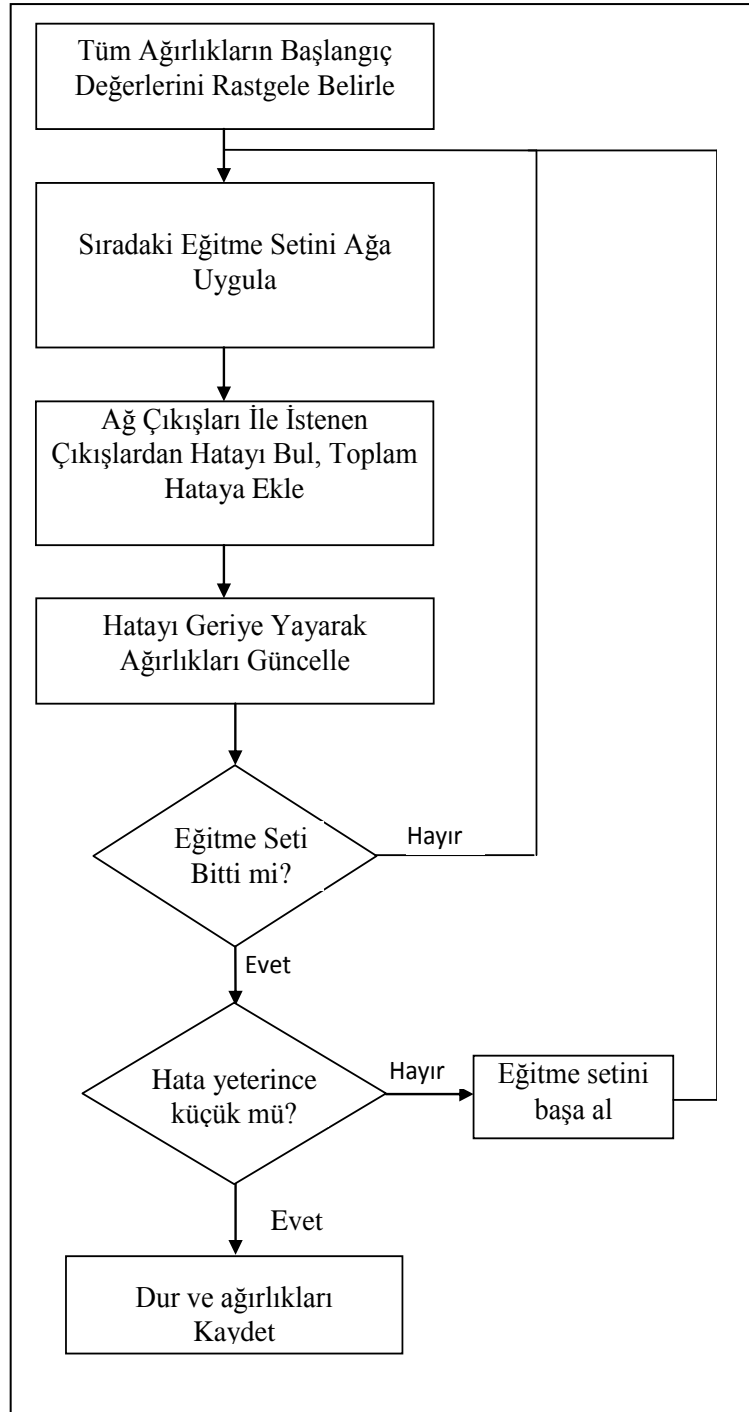
4.4.5 Öğrenme ve Momentum Sabitleri

YSA ile ilgili bir başka sorun da, düzgün bir öğrenme katsayısının (ε) ayarlanmasıdır. Ağırlıkları çok yüksek tutmak davranışın bozulmasına sebep olabilir. Öğrenme katsayısını böyle bir davranışı önlemek için küçük tutmak gereklidir. Öğrenme katsayısı, $0.01 < \varepsilon < 1$ aralığında seçilen sabit bir sayıdır. Öte yandan, çok küçük bir öğrenme oranında, öğrenme işleminin yavaşlamasına yolaçar.

Momentum (α) fikri bu noktadan hareketle ortaya atılmıştır. Momentum mevcut delta ağırlığı üzerinden önceki delta ağırlığının belirli bir kısmını besler. Böylece daha düşük öğrenme katsayısı ile daha hızlı öğrenme elde edilir. Momentum katsayısı genellikle $0 < \alpha < 1$ aralığında değişen sabit bir sayıdır.

Hatanın geriye yayılması algoritması genellikle iyi bir performans göstermesine rağmen; bu algoritma, bir gradyent arama tekniği olduğu için, global minimum yerine, en küçük kareler (Least Mean Square) fonksiyonunun bir lokal minimumunu bulabilmektedir. Performansı artırmak ve lokal minimum problemini azaltmak için; gizli düğümler ilave edilmesi, ağırlıkların ayarlanması için kullanılan öğrenme katsayısının azaltılması ve farklı rastgele ağırlık setlerinden başlayarak birçok defa eğitme işleminin tekrarlanması öneriler arasındadır (Rumelhart, D.E., ve ark. 1986.).

Hatanın geriye yayılması algoritması, karesi alınmış hata fonksiyonunu minimize eden kodlu bir algoritma olup ve genelleştirilmiş delta kuralını eğitme için kullanılmaktadır. Bu YSA modelinin eğitme işleminin akış diyagramı, Şekil 4.10 da verildiği gibidir.



Şekil 4.10. Hatanın Geriye Yansımaları Algoritmasının akış diyagramı (Karlık, B. 2009)

5 BULANIK KÜMELEME

5.1 Bulanık Mantığa Giriş

Bulanık mantık (Fuzzy Logic) kavramı ilk kez 1965 yılında California Berkeley Üniversitesinden Prof. Lotfi A.Zadeh'in bu konu üzerinde ilk makalelerini yayınlamasıyla duyuldu. O tarihten sonra önemi gittikçe artarak günümüze kadar gelen bulanık mantık, belirsizliklerin anlatımı ve belirsizliklerle çalışılabilmesi için kurulmuş katı bir matematik düzen olarak tanımlanabilir. Bilindiği gibi istatistikte ve olasılık kuramında, belirsizliklerle değil kesinliklerle çalışılır ama insanın yaşadığı ortam daha çok belirsizliklerle doludur. Bu yüzden insanoğlunun sonuç çıkarabilme yeteneğini anlayabilmek için belirsizliklerle çalışmak gereklidir.

Bulanık mantık, İngilizcesiyle fuzzy logic, adından anlaşılacağı gibi mantık kurallarının esnek ve bulanık bir şekilde uygulanmasıdır. Klasik (boolean) mantıkta bildiğiniz gibi, "doğru" ve "yanlış" ya da "1" ve "0"lar vardır, oysa bulanık mantıkta, ikisinin arasında bir yerde olan önermeler ve ifadelere izin verilebilir ki, gerçek hayata baktığımızda hemen hemen hiçbir şey kesinlikle doğru veya kesinlikle yanlış değildir. Gerçek hayatta önermeler genelde kısmen doğru veya belli bir olasılıkla doğru şeklinde değerlendirilir. Bulanık mantığa da zaten klasik mantığın gerçek dünya problemleri için yeterli olmadığı durumlar dolayısıyla ihtiyaç duyulmuştur. Bulanık mantığın sistemi şu şekildedir. Bir ifade tamamen yanlış ise klasik mantıkta olduğu gibi 0 değerindedir, yok eğer tamamen doğru ise 1 değerindedir. (Ancak bulanık mantık uygulamalarının çoğu bir ifadenin 0 veya 1 değerini almasına izin vermezler veya sadece çok özel durumlarda izin verirler.) Bunların dışında tüm ifadeler 0'dan büyük 1 den küçük reel değerler alırlar. Yani değeri 0.32 olan bir ifadenin anlamı %32 doğru %68 yanlış demektir.(Yapay Zeka, 2011)

5.2 Bulanık Kümeleme (BK) Yaklaşımı

Bulanık kümeleme yöntemi, kümeler birbirinden belirgin bir şekilde ayrılmıyorsa ya da üyeliklerinde bazı birimler küme üyeliğinde kararsızsa uygun bir yöntem olarak ortaya çıkmaktadır. Bulanık kümeler kümedeki birimin üyeliği olarak tanımlanan 0 ile 1 arasındaki her birimi belirleyen fonksiyonlardır. Birbirine çok benzeyen birimler aynı kümede yüksek üyelik ilişkisine göre yer alırlar. Bundan dolayı Bulanık Kümeleme Yöntemi, birimlerin kümeye ya da kümelere ait olabilme katsayılarını hesaplar. Üyelik katsayılarının toplamı daima 1'e eşittir. Böylelikle birim en yüksek üyelik katsayısına sahip olduğu kümeye atanır. Üyelik fonksiyonları, kümedeki elemanlar sürekli veya süreksiz olsun bir bulanık kümedeki bulanıklığı karakterize fonksiyonlardır. Klasik kümeleme yöntemlerinde ise her bir birim sıfır olmayan sadece bir üyelik katsayısına sahiptir ve bu değer daima 1'dir. Dolayısıyla klasik kesin kümeleme yöntemleri, bulanık çözümlemenin sınırlı bir durumudur. (Bezdek ve ark., 1992, Kaufman ve ark., 1990)

BK algoritmaları, veri uzayındaki bireylerin farklı üyelik dereceleri ile her kümeye girmesine izin veren bölünme-optimizasyon teknikleridir. En yaygın kullanılan bulanık kümeleme algoritması Bulanık C-Ortalamlar (BCO) algoritmasıdır. Bu algoritma, klasik C-ortalamlar algoritmasının bulanık uygulamalar için genişletilmiş halidir. (Bezdek ve ark., 1984) BCO algoritması, kümenin merkezi (center) olarak adlandırılan, her küme içindeki kümeyi en iyi temsil eden noktayı bulmaya çalışır. Ayrıca küme içindeki her bir birey bu kümeye aitlik derecesini gösteren bir değere sahiptir. Buradan anlaşılacağı gibi bulanık kümeleme algoritmalarının bu özelliği, her bir bireyin farklı derecelerle her kümeye ait olmasını sağlar.

5.3 Bulanık Kümeler

Klasik küme teorisinde, bir eleman bir kümeye kesin olarak ya girer ya da girmez. İkisinin ortasında bir durumdan söz edilemez. Bulanık küme teorisi ise, elemanların farklı üyelik dereceleriyle birden fazla kümeye girmesini sağlayan, klasik küme teorisinin genişletilmiş versiyonudur. Üyelik fonksiyonları bir elemanın bir kümeye ne kadar ait olduğunu gösteren değerlerdir. 0 olması durumu, elemanın kümeye ait olmadığını, 1 olması durumu ise kesin olarak ait olduğunu gösterir. 1'ye yakın değerler elemanın yüksek derecede kümeye ait olduğunu, 0 yakın değerlerde ise düşük derecede ait olduğunu gösterir.

Bu şekilde tanımlanan üyelik derecelerinin her bir bulanık söz için üç temel özelliği sağlaması tanım olarak gerekmektedir. Bunlar şöyle sıralanabilir; Bulanık kümenin normal olmasıdır ki; bunun için en azından o küme de bulunan elemanlardan bir tanesinin en büyük üyelik derecesi olan 1'e sahip bulunması gerekliliğidir.

Bulanık kümenin monoton olması gerekir ki bunun anlamı üyelik derecesi 1'e eşit olan elemana yakın sağda ve soldaki elemanların üyelik derecelerinin de 1'e yakın olmasıdır. Üyelik derecesi 1'e eşit olan elemandan sağa ve sola eşit mesafede hareket edildiği zaman bulunan elemanların üyelik derecelerinin birbirine eşit olmasıdır ki buna da bulanık kümenin simetrik özelliği adı verilir.

Klasik kümelerle bulanık kümelerin arasındaki önemli farklardan bir tanesi, klasik kümelerin sadece bir tane dikdörtgen üyelik derecesi fonksiyonu bulunmasına karşılık, bulanık kümenin yukarıdaki üç şarttan ilk ikisini mutlaka sağlayacak biçimde değişik üyelik derecesi fonksiyonlarına sahip olmasıdır. Buradan üyelik derecesi fonksiyonlarının mutlaka simetrik olma özelliğini sağlaması gerekmediği anlaşılabilir.(Şen, Z. 2001)

Diğer kümeleme yöntemlerine benzer olarak bulanık kümeleme de uzaklık ölçümüne dayanır. Bu yöntemin kullandığı uzaklık ölçümlerinin arasında Euclidian ve Mahalanobis uzaklık ölçümleri yer alır. Bu uzaklık ölçümlerinden hangisinin seçileceği küme yapısına ve kullanılan algoritmaya bağlıdır.

Bulanık kümelemenin geleneksel kümeleme yöntemlerine göre avantajı, veri hakkında daha detaylı bilgi vermesidir. Diğer taraftan dezavantajları da vardır. Çok sayıdaki birey ve küme durumunda çok fazla küme elemanı olacağından, özetlemek ve bilgiyi tasnif etmek zordur. Ayrıca bulanık kümeleme algoritmaları genellikle karmaşıktır ve daha çok belirsizlik söz konusu olduğunda kullanılır. (Şahinli, F. 1999)

5.4 Bulanık Kümelerin Gösterimi

Verilen bir U evrensel kümesi için karakteristik fonksiyon f_A bulanık olmayan bir A kümesi için aşağıdaki şekilde tanımlanır.

$$\begin{aligned} f_A : U &\rightarrow \{0,1\} \\ f_A(x) &= 1 \quad x \in A \\ f_A(x) &= 0 \quad x \notin A \end{aligned} \quad (5.1)$$

Bulanık kümeler ise $[0,1]$ aralığında değişen üyelik derecelerini kullanır.

$$\mu_A : U \rightarrow [0,1]$$

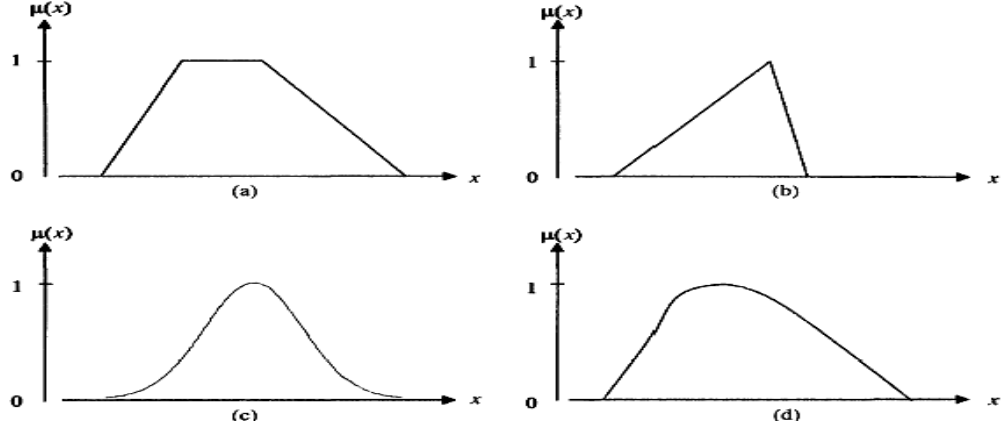
Bulanık kümeler 4 yakın sayılar, siyaha yakın renkler, büyük gezegenler, yüksek sıcaklık, kısa uzaklık gibi kesin olmayan kavramları temsil eder. Bulanık küme alanları kesikli veya sürekli olabilir. Sonlu alana sahip bulanık küme A için yaygın olarak matematiksel notasyon $\{x_1, x_2, \dots, x_n\}$ ve üyelik fonksiyonu μ_A ise:

$$\mu_A(x_1)/x_1 + \mu_A(x_2)/x_2 + \dots + \mu_A(x_n)/x_n \quad (5.2)$$

Örnek olarak 4' yakın sayılar için üyelik fonksiyonu

$$0.2/1+0.5/2+0.8/3+1/4+0.8/5+0.5/6+0.2/7$$

Çoğu durumunda, bulanık kümeler sürekli alanlardan oluşur. Bu durumda üyelik fonksiyonları için matematiksel fonksiyonlar ve grafikler tanımlanır. Çoğu durumda, üçgen, yamuk, s-şekilli ve gauss fonksiyonları tercih edilir.



Şekil 5.1. Yaygın olarak kullanılan üyelik fonksiyonlar (a)Yamuk Şekilli (b) Üçgen Şekilli (c)Gauss Üyelik Fonksiyonu (d)Çan Şekilli Not: Tüm üyelik fonksiyonları, sürekli, normal ve konvektistir.

Üçgen şekilli bulanık küme aşağıdaki gibi tanımlanabilir.

$$\begin{aligned} \mu_A(x) &= m_l x + h_l & x \leq a \\ \mu_A(x) &= -m_r x + h_r & x > a \end{aligned} \quad (5.3)$$

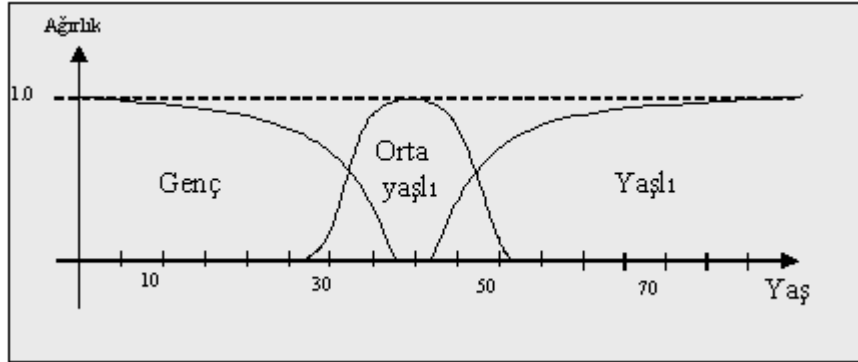
Üçgen şekilli üyelik fonksiyonu düşünüldüğünde; üyelik değeri 1 olan elemanın olduğu kısım fonksiyonun **özü**, üyelik derecesi 0 ile 1 arasında değişen elemanların olduğu kısım fonksiyonun **kısımları** üyelik fonksiyonu 0 den büyük olan elemanların oluşturduğu kısım ise fonksiyonunun **dayanağı** olarak adlandırılır. Normal olma özelliğinden dolayı üyelik fonksiyonunun değeri en az bir eleman için 1 olmalıdır. Ayrıca üyelik fonksiyonları sürekli artan ve sürekli azalan özelliğine sahip olmalıdır. Üyelik değeri 0.5 olan kısım **geçiş noktası**, en yüksek üyeliğe sahip olan kısım ise **yükseklik** olarak adlandırılır.

Bulanık küme kavramı için önemli olan bir diğer kavram ise dilsel değişken kavramıdır. Cebir değişkenleri sayısal değerler alırken, dilsel değişken kelime veya cümle şeklindeki metin değerler almaktadır. Bu değerlerin kümesi terim kümesi olarak adlandırılır. Terim kümesi içindeki her bir değer, temel değişkenlere dayanarak tanımlanmış bulanık bir değişkendir. Kısaca hiyerarşik olarak bakıldığında dilsel değişken \rightarrow bulanık değişken \rightarrow temel değişken

Dilsel deęişkenin “yaş ” olarak etiketlendięi varsayılırsa; bu dilsel deęişkenin her biri bir bulanık kümeye karşılık gelen, terimleri, “yaşlı”, “çok yaşlı”, “orta yaşlı”, “biraz genç”, “ genç”, “çok genç” şeklinde olabilir. Her bir terim, 0 ile 100 arasında ölçeklendirilebilen temel deęişkenlerin üzerinde tanımlanan bulanık bir deęişkendir (Güler, N., 2006).

5.5 Üyelik Fonksiyonları

BK, her biri n tane elemandan oluşan bir sıralı eleman gruplarıdır. Bu tür kümelerde, üyelik ağırlıklarını deęerlendiren ve karakteristik fonksiyonlarını genelleştiren üyelik fonksiyonları (fuzzy membership) adı verilen fonksiyonlar kullanılır. İşte bulanık kümeler, Şekil 5.2'de gösterildięi üzere böyle bir düşünüşe olanak tanır. Kümelerin birbirinden keskin çizgilerle ayrılmamış olması, aralarında belirli bir örtüşüm olması, 35 yaşın bir oranda hem orta yaşlı, hem de genç olarak düşünülmesine olanak tanır.

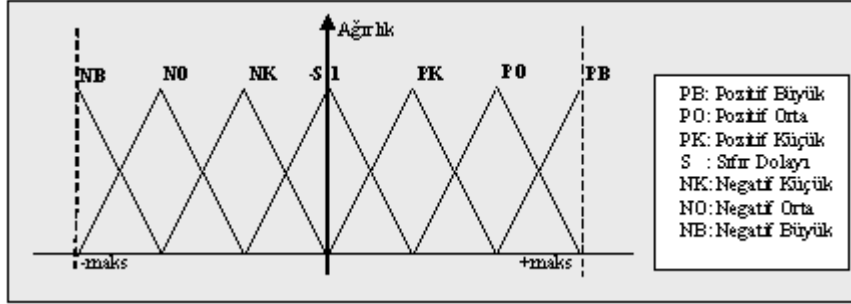


Şekil 5.2. Bulanık Kümeler: Yaşın üç gruba ayrıldığı bulanık kümeleme

Üyelik fonksiyonu: $0 \leq \mu_A \leq 1$ aralığında yazılabilen bir doğal sayıdır. Eğer ($\mu_A = 0$) ise negatif nesnenin kümeye hiç ait olmadığını, ($\mu_A = 1$) ise kümeye tam anlamıyla ait olduğunu gösterir (giarratano). Üyelik fonksiyonunun kısmi deęerleri ise (0.5 gibi) objenin kümeye kısmen ait olmasını ifade eder ve bu deęerlere Üyelik fonksiyonun ağırlıkları denir. Bulanık mantık kullanan bütün sistemlerde, üyelik fonksiyonun üretimi çok önemlidir, çünkü, sistemin kararlılığı, üyelik fonksiyonun seçimine baęlıdır ve bu seçimi uzman bir kiři gerçekleştirir. Bu yüzden üyelik fonksiyonu üretimindeki ana problemlerden biri de, uzman kiřiden elde edilen sisteme ilişkin bilgilerin eksikliğidir.

Üyelik fonksiyonları, denetlenen sürecin özelliklerine göre değişik şekillerde olabilir. Genel olarak, sıkça kullanılan üçgen, çan ve yamuk şekilleridir.

Üyelik fonksiyonlarında; 3 (küçük, orta, büyük), 5 (küçük, orta derece küçük, orta, orta derece büyük, büyük), ve 7 (pozitif büyük, pozitif orta, pozitif küçük, sıfır dolayı, negatif küçük, negatif orta, negatif büyük) ayrı etiket tanımlanır. En yaygın olanı 7 etiket kullanılmasıdır. Şekil 5.3'te böyle bir durum gösterilmiştir (Karlık, B., 2009 c).



Şekil 5.3. Yedi ayrı etiketli üyelik (Karlık, B., 2009 c)

5.6 Bulanık C-Ortalamalar Algoritması

Bulanık kümeleme analizi, desen tanıma, görüntü işleme ve bulanık modelleme gibi uygulamalar için etkili bir şekilde kullanılmaktadır. En iyi bilinen bulanık kümeleme yaklaşımı ilk olarak Dunn (1974) tarafından önerilen ve Bezdek tarafından geliştirilen Bulanık C-Ortalamalar(BCM) algoritmasıdır.

Bulanık C-Ortalamalar algoritması, amaç fonksiyonuna dayanan bütün kümeleme tekniklerinin temelini oluşturmaktadır. İlk olarak Duda ve Hart sert küme bölünmesini (Hard Cluster Partition) hesaplamıştır. Dunn ise (Dunn, 1974) bir bireyin birden fazla kümeye girebilmesini sağlamak için bu algoritmanın bulanık versiyonunu tanıtmıştır. Son olarak, Bezdek bulanıklık indeksini(m) dahil ederek algoritmanın son halini geliştirmiştir.

BCO algoritması sonuçlandığında, p- boyutlu uzaydaki noktalar küresel bir şekil alır. Bu kümelerin yaklaşık olarak aynı boyutta olduğu varsayılır. Her bir kümeyi, küme merkezleri (center) temsil eder ve bunlara prototip denir. Uzaklık ölçüsü olarak, veriler ile küme merkezi arasındaki Euclidian uzaklığını kullanır.

$$d^2(v_i, x_k) = D_{FCM} = \|x_k - v_i\|^2 = \sum_{v=1}^p (x_k^{(v)} - v_i^{(v)})^2 \quad (5.4)$$

Burada $p \in N, D = R^p, x = \{x_1, x_2, \dots, x_n\} \subseteq D, C = R^p, c \in N, R = P_c(C), m \in R_{>1}$ ve

$$d: D \times C \rightarrow R, (x, p) \rightarrow \|x - p\| \quad V = \{v_1, v_2, \dots, v_c\} \in R$$

Kümeleme yönteminde küme elemanlarının küme merkezine olan uzaklıkları hesaplanır ve bu hesaplanan değer bir önceki iterasyonda hesaplanan değere çok yakınsa algoritma doğru çalışıyor demektir. Bu kümeleme yöntemi için kullanılan amaç fonksiyonu şu şekildedir:

$$J(\mu, v) = \sum_{i=1}^N \sum_{j=1}^C \mu_{i,j}^m \|x_i - v_j\|^2 \quad (5.5)$$

J amaç fonksiyonudur – bir önceki değeri.

N Resimdeki nokta sayısı.

C oluşturulacak küme sayısı.

μ üyelik değerleri tablosu —Nx C tane girişin üyelik değerlerini tutar.

m bulanıklık indeksi (1’den büyük bir değer).

x_i N içinde i. nokta.

v_j C içindeki j. küme merkezi.

$\|x_i - v_j\|$ x_i ve v_j arasındaki Euclidean uzaklığı.

Bu fonksiyon en küçük kareler fonksiyonudur. Bu fonksiyonun değeri her bir iterasyonda tekrar hesaplanır ve yeni hesaplanan değer ile bir önceki değer arasındaki fark belli bir değere eşit ya da küçükse iterasyon durdurulur, kümeleme işlemi tamamlanmıştır.

Eğer J(μ, v) fonksiyonu C’nin her değeri için minimize edilecek olursa, diğer bir deyişle v_i ’lere göre 1.dereceden türevi alınıp 0’a eşitlenirse BCO Algoritmasının Prototipi şu şekilde olacaktır;

$$v_j = \frac{\sum_{i=1}^N \mu_{ij}^m x_i}{\sum_{i=1}^N \mu_{ij}^m} \quad (5.6)$$

Burada bulanıklaştırıcı m BCO algoritması için önemli bir parametredir. Kaç tane kümenin üst üste geleceğini kontrol eder. m=1(bulanıklık indeksi) seçildiği zaman BCO algoritması, Sert C-Ortalamlar(SCO) algoritmasının genelleştirilmiş bir biçimi olacaktır. Burada küme prototipleri yine aynı formül ile hesaplanır ve üyelikler ya 0 olacaktır ya 1 olacaktır. Veriler en küçük uzaklığa sahip olduğu kümeye girer. Ancak BCO algoritması 0'a bölüm hatasına yol açabileceğinden m değeri olarak 1 kullanılmasına izin vermez. (Höppner, 1999) Eğer m değeri çok büyük seçilirse bireylerin kümelere etkileri çok küçük olacaktır ve bireylerin kümelere üyelik dereceleri yaklaşık olarak 1/c olacaktır.

BCO Algoritması için Gerekli Adımlar:

Adım1: Başlangıç Değerlerinin Girilmesi: Üyelik dereceleri matrisi μ rastgele (0,1) aralığında girilir. Küme sayısı C, bulanıklık indeksi (m) ve hassasiyet (ϵ) atamaları yapılır.

Adım 2: J değeri aşağıdaki formüle göre hesaplanır.

$$J(\mu, v) = \sum_{i=1}^N \sum_{j=1}^C \mu_{i,j}^m \|x_i - v_j\|^2 \quad (5.7)$$

Adım 3: Küme merkezleri (v_j) hesaplanır.

$$v_j = \frac{\sum_{i=1}^N \mu_{ij}^m x_i}{\sum_{i=1}^N \mu_{ij}^m} \quad (5.8)$$

Adım 4: V küme prototiplerinin rastgele üretildiği varsayılırsa bu değerler kullanılarak üyelik dereceleri matrisi hesaplanır.

$$\mu_{ik} = \sum_{l=1}^C \left(\frac{\|x_i - v_j\|}{\|x_i - v_l\|} \right)^{-2/m-1} \quad (5.9)$$

Adım 5: (5.6) eşitliğine göre V küme prototipleri güncellenir.

Adım 6: J değerini tekrar hesapla, J değeri ϵ den küçükse, ($\|V^{(t)} - V^{(t-1)}\| < \epsilon$) iterasyon durdurulur aksi takdirde **Adım 3**'e geri dönülür.

BCO algoritması uygulandıktan sonra hangi bireyin hangi kümeye gireceğine karar vermek için üyelik dereceleri kullanılır. Her bir bireyin hani kümeye olan üyeliğinin en büyük olduğuna bakılır ve bu bireyler o kümeye dahil edilir. Ancak her bir birey diğer kümelere de belli bir üyelikle girerler.

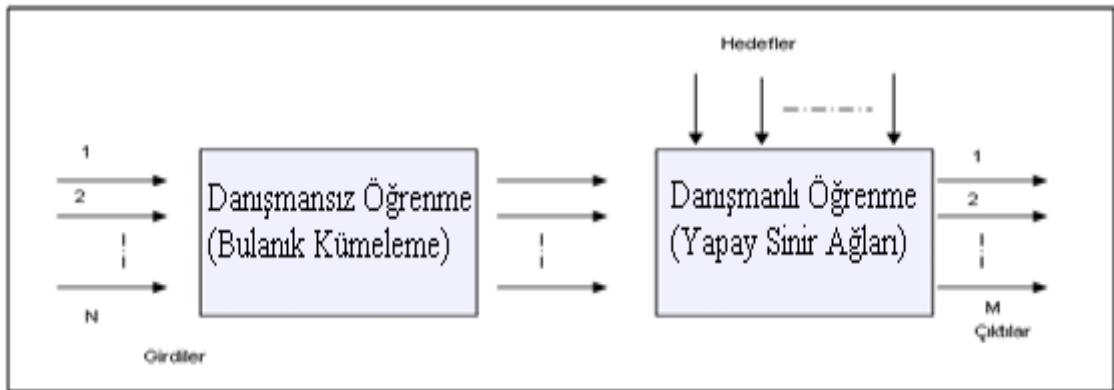
BCO algoritmasının sonucu başlangıçta rastgele üretilen değerlere oldukça bağlıdır. Bu yüzden bu rastgelelikten kaynaklanan problemleri ortadan kaldırmak için bir takım algoritmalar geliştirilmiştir.

6 BULANIK KÜMELEMELİ YAPAY SİNİR AĞLARI

6.1 BKYSA'nın İris Tanımda Uygulanması

BKYSA, şu ana kadar daha çok bir boyutlu (1-D) işaretlerde uygulanmıştır (Karlık, B. ve ark 2002 - 2003, Özbay, Y. ve ark 2006). BKYSA yapılan çalışmada iki boyutlu işaretlerde (2-D) kullanılmıştır. BK işleminin böyle bir işarete kullanılması için bir modelleme gerekmektedir. Bu çalışmada YSA ile sınıflandırma yapılmadan önce eğitim (örnek) girdisi BK işleminden geçirilmektedir. Bu işlemin çıktısını da YSA'ya girdi olarak verilmekte ve sınıflandırılmaktadır. BK, girdinin içinden, tanımda fazla önem içermeyen (gereksiz) özellikleri öznelik vektöründen çıkartarak sadece tanımda fayda sağlayacak olanları ortaya çıkartmaktadır. Ayrıca bu boyut düşürmesi, tanımın da kalitesini artırmaktadır. Çünkü gereksiz ve tanımda hiçbir faydası olmayacak, hatta tanımayı olumsuz etkileyebilecek olan özellikler (features) hem YSA'nın eğitim süresini uzatır, hem de eğitimin, dolayısı ile tanımın başarısını negatif yönde etkilemektedir. Şekil 6.1'de BKYSA modelinin blok diyagramı görülmektedir.

Aşağıda alt başlıklar halinde bu çalışmada BKYSA ile Iris tanıma işleminin nasıl gerçekleştirildiğini ve elde edilen ilk sonuçlar daha detaylı bir şekilde anlatılmıştır.



Şekil 6.1. BKYSA blok diyagramı

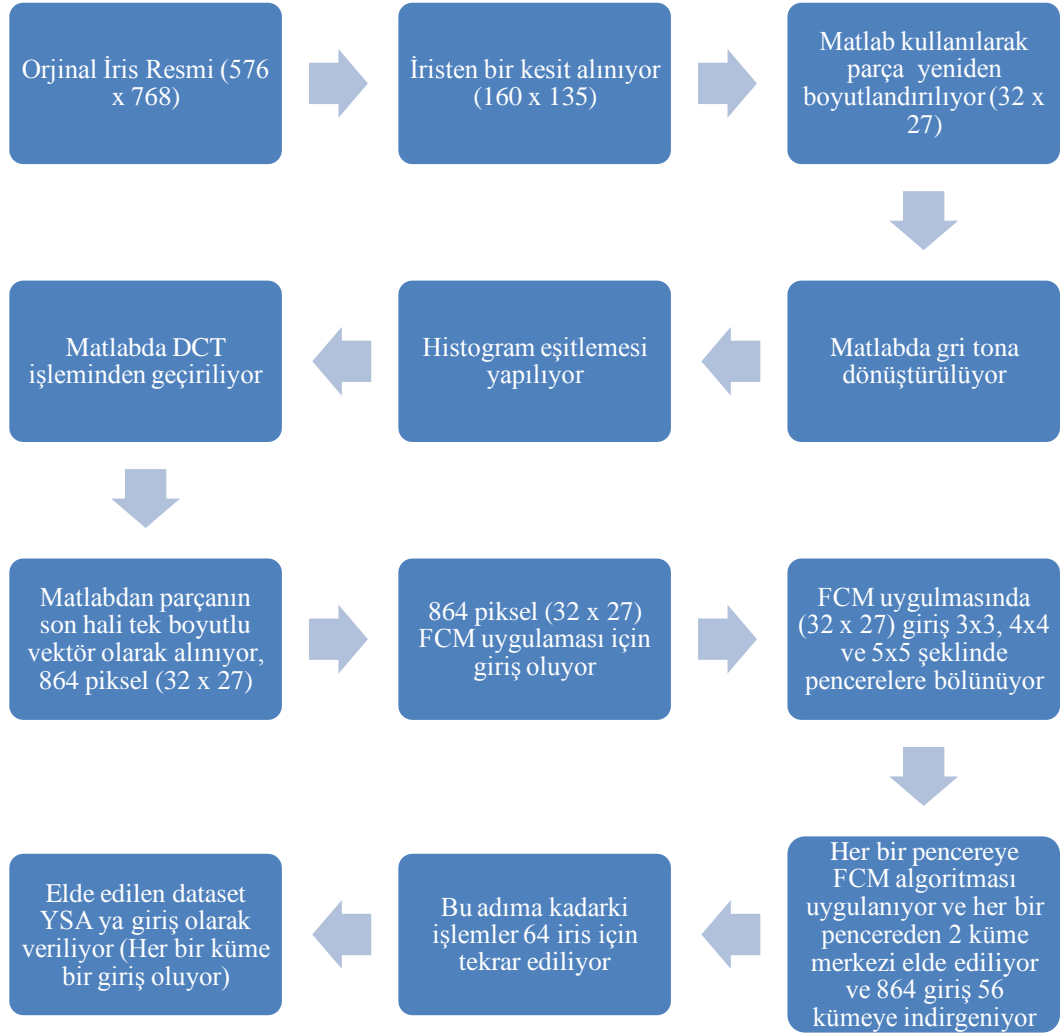
Tez çalışmasında BKYSA iki boyutlu biyometrik üzerine uygulanmış ve elde edilen sonuçlar klasik YSA ile karşılaştırılmıştır. Bu amaçla veritabanımızda ki iris resimlerinden 160 x 135 boyutlarında parçalar çıkarılmış ve çıkarılan parçalar Matlabda 'resize' methodu ile 32 x 27 boyutlarına indirgenmiş sonra özellik çıkarma işlemleri uygulanmıştır.

Özellik çıkarma işlemlerinden sonra özellik vektörü pencerelemlere bölünerek her bir pencereye BCO algoritması uygulanmıştır. BCO işleminden sonra küme merkezleri elde edilmiştir. Elde edilen küme merkezleri YSA uygulamasına giriş olarak verilmiştir. Bu sayede YSA uygulamasının giriş sayısında azalma olmuştur. Özellik çıkarma işleminde ayrı ayrı DCT ve DWT kullanılmıştır. Aşağıdaki akışta (Şekil 6.2) DCT kullanılan yöntem baştan sona kadar çizilmiştir.

BKYSA'nın iki boyutlu bir veriye uygulanmasında en önemli nokta pikseller arasındaki komşulukların kaybedilmemesidir. Komşuluk değerlerini kaybetmemek/en üst seviyede tutmak için pencereleme yöntemi kullanılmıştır. Pencereleme yöntemi sayesinde bir birine yakın olan noktalar kendi aralarında kümeleme yapılmış ve sadece o pencereyi temsil eden küme merkezleri elde edilmiş olur. Bir birine olan komşulukları düşük olan ancak piksel değerleri birbirine yakın olan noktalar aynı kümeye konulmamıştır. Bu sayede iris resmi üzerinde ki özellikler kaybolmamış olur.

Özellik vektörü üzerinde kümeleme yapılırken satır satır kümeleme yapılmış olsaydı, sadece x ekseninde komşuluklar tutulmuş olacaktı. Satır başındaki piksel ile satır sonundaki piksel arasındaki komşuluk çok düşük oranda olduğu için küme elemanları arasındaki komşuluklarda düşük olacaktı. Aynı şekilde pencereleme yapmadan tüm özellik vektörü üzerinde kümeleme yapılmış olsaydı da komşulukları çok düşük olan pikseller aynı kümeye dahil edilmiş olacaktı.

Tanıma işlemi iris resimleri üzerinde özellikler/desen sayesinde olmaktadır ve her bir kişiye ait iris farklı desenlere sahiptir. İris üzerindeki özellikler piksel grupları halinde tanımlanmaktadır. Bu nedenle pencereleme yapıldıktan bu özelliklerin kaybolması veya birbirine karışması engellenmiş olmaktadır. Bulanık kümeleme için BCO algoritması kullanılmış ve bu algoritmayı kullanan bir uygulama geliştirilmiştir. Geliştirilen uygulama ile küme sayıları ve pencere boyutları ayarlanarak irise ait küme merkezleri elde edilmiştir. (Şekil 6.7)

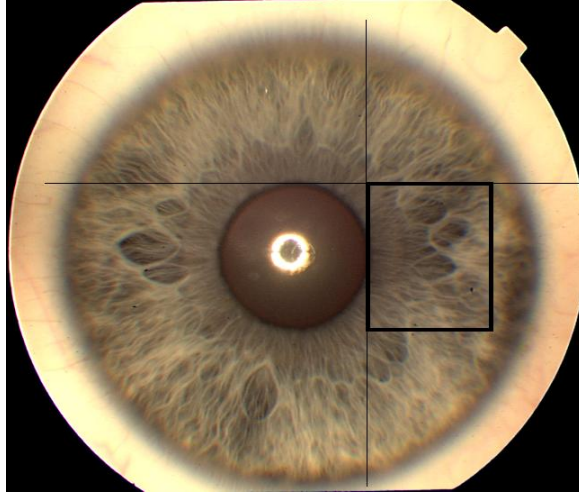


Şekil 6.2. BKYSA ile biyometrik tanıma akış diyagramı (DCT)

6.1.1 Ön işlemler

Tez çalışmasında biyometrik olarak iris seçilmiştir. İris; tanıma için kullanılabilir çok zengin bir desene sahiptir. Ancak iris desenlerini gürültülerden arındırmak her zaman kolay değildir. Bu nedenle birçok çalışmada irisin gürültüden en az etkilenen bölgeleri kullanılarak iris tanıma işlemleri yapılmıştır (Du, Y. ve ark, 2005, Bolle, R. M. ve ark, 2004). Gözün üst ve alt kısmının göz kapağı ve kirpiklerin etkisinde kalma ihtimali oldukça yüksektir. Bu nedenle sağ ve sol kısımlar ile irisin gözbebeğine yakın olan iç kısımları gürültüden en az etkilenmesi beklenen kısımlardır. Bu çalışmada da gözün tamamı yerine belli bir kısmını kullanarak tanımayı gerçekleştirmeye çalışılmıştır.

Bu çalışmada iris parçası çıkarılırken irisin gözbebeği referans noktası olarak alınmıştır. Gözbebeğinin üstünden ve sağ yanından teğet geçecek şekilde iki doğru çizilmiş ve bu iki doğrunun oluşturduğu düzlemden (-1,1) bölgesi üzerinde çalışma yapılmıştır. Bu bölgeden de gürültünün olma olasılığı en düşük olan bölümü çıkarılmıştır. Çıkarılan bölümün boyutu 160x135 pikseldir. Bu boyutlar genel olarak iris veritabanı incelendikten sonra en uygun değer boyut olarak kabul edilmiştir. Seçilen alan, göz kapaklarının ve kirpiklerin etkisinden teorik olarak en az etkilenecek olan kısımlardandır. Seçilen bölgeyi Şekil 6.3’de görebilirsiniz.



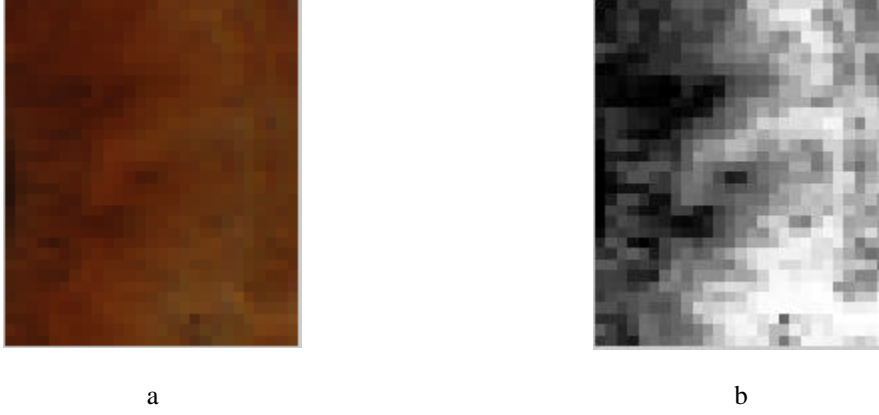
Şekil 6.3. İrisin seçilen kısmı

Eski ve patentli ürün haline dönüştürülmüş olan yöntemlerde, resimlerin belli bir mesafeden, belli bir süre, belli bir noktaya bakarak alınan kayıtlara dayanmaktadır. Bu, kullanıcının göz resminin alındığının farkında olmasına ve bunu gerçekleştiren sistemle belli bir işbirliği yapmasını zorunlu kılmaktadır. Günümüzde iris tanıma konusunda çalışmalar daha çok kullanıcının işbirliğini gerektirmeyecek (hatta göz resminin alındığının farkında olmayacak) yöntemlerin geliştirilmesine doğru kaymaktadır. Bazı resimlerde iris ve gözbebeği arasında yeterli kontrast olmaması, iris bölgesinin doğru bir şekilde bulunmasını zorlaştırmaktadır. Odaklanma problemi ve kullanılan göz merceği veya gözlüğün oluşturduğu gürültü de irisin bulunmasını zorlaştıran önemli bir faktördür.

Öncelikle bir resimde iris halkasının tespiti gerekmektedir. Bunun için literatürde çeşitli yöntemler vardır. Hough dönüşümü (HD) yöntemi bunlardan en yaygınlarından birisidir (Wildes, R. P., 1997). Yine “integrodifferential operator” bilinen başka bir yöntemdir (Daugman, J.G., 1993). Bu çalışmanın amacı mevcut yöntemlerden daha başarılı bir iris bulma yöntemi geliştirmek değildir. Amacımız, daha önce görüntü tanımada kullanılmamış olan BKYSA sistemini İris Tanıma’da uygulamaktır. Bu nedenle iris bölgesinin çıkartılmasıyla çok fazla zaman kaybedilmemiştir. Geliştirilen basit bir uygulama ile her bir göz resminden, belirtilen koordinatlar kullanılarak iris parçaları çıkarılmıştır. Genel bir yöntem ya da alternatif bir yöntem kullanılmamış onun yerine iris parçasından özellik çıkarma ve sınıflandırma bölümleri üzerine yoğunlaşmıştır. Çıkardığımız parçalar göz kapağı ve kirpik gibi gürültüler barındırmamaktır. Ancak bazı resimlerde odaklanma problemi ve kullanılan makinesinden kaynaklanan lekeler bulunmaktadır. Bu tür resimler gerçekçiliği korumak için veritabanından çıkarılmamıştır.

İris parçası resimden çıkarıldıktan sonra Matlab kullanılarak yeniden boyutlandırılmış gri tona çevrilmiş. İris parçasının boyutu 160 x 135’tan 32 x 27 ye indirgenilmiştir. Yeniden boyutlandırmanın en büyük nedeni YSA’nın eğitim sürelerini makul seviyelerde tutmaktır. Test için kullandığımız bilgisayarda YSA’nın giriş sayısı 1000’in üstüne çıktığında eğitim süresinde ciddi oranda artışlar olmuştur.

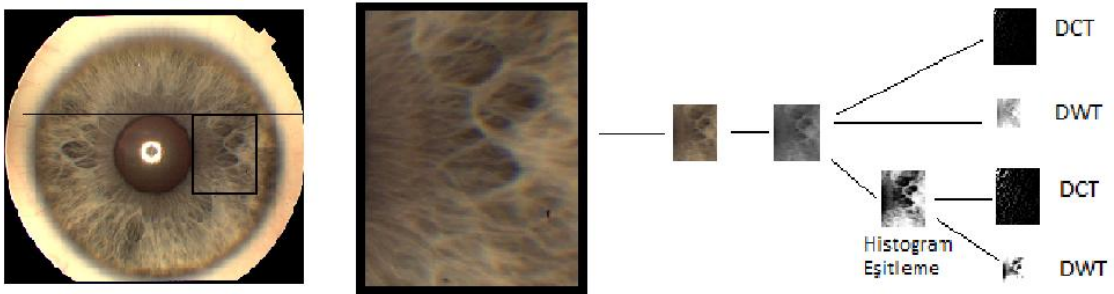
İris parçası çıkarıldıktan sonra, nispeten resim içinde koyu bölgeleri içerdiği için, özellik çıkarımından önce Matlab kullanılarak histogram eşitlemesi yapılmıştır. Bu şekilde elde edilen iris parçalarına belli bir açıklık kalitesine getirilmiştir. Bu, farklı ortamlarda, farklı ışık şartlarında çekilen resimlerde oluşabilecek bir tanıma zorluğunu giderecek bir yöntemdir. Şekil 6.4’de iris parçasının histogram eşitlemesi yapıldıktan sonraki hali görülmektedir. Şekil 6.5’de ise ön işlemler ve özellik vektörü çıkarma adımlarında iris resimleri görüntülenmektedir.



Şekil 6.4. Histogram Eşitleme a) Orjinal parçanın küçültülmüş hali, b) a'nın histogram eşitlenmiş hali

6.1.2 Öznitelik Vektörü Çıkarımı

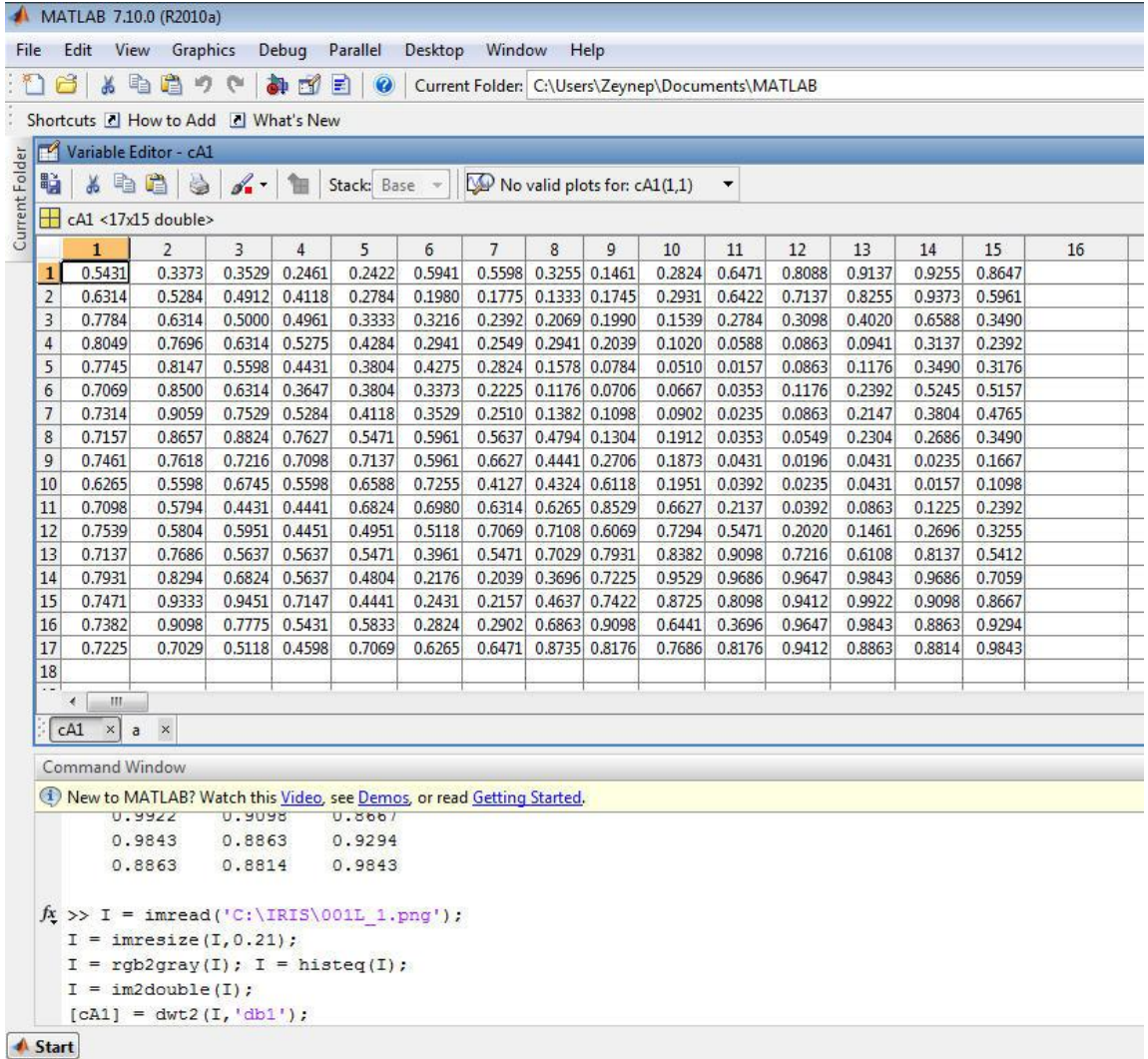
Ön işlemleri tamamlanmış iris parçalarının, sınıflandırma işleminde kullanılacak olan özellik vektörleri çıkarılmıştır. Özellik vektörü çıkarılırken Ayrık Kosinüs Dönüşümü ve Ayrık Sinüs Dönüşümü kullanılmıştır. Bu iki yöntem ayrı ayrı uygulanmış ve karşılaştırmaları yapılmıştır. Tablo 6.1'de DCT ve DWT ile elde edilen özellik vektörlerinin boyut değişimleri görülmektedir. Öz nitelik vektörleri Matlab kullanılarak çıkarılmıştır. DWT ve DCT uygulandıktan sonra iris resmi hala iki boyutludur (Şekil 6.6). Bu iki boyutlu veri yine Matlab kullanılarak tek boyuta indirgenilmiştir (Şekil 6.7). Bu şekilde DWT için 17×15 'lik matris, 1×255 e dönüştürülmüş olur.



Şekil 6.5. Özellik Çıkarma adımlarındaki iris görüntüleri

Tablo 6.1. Öznitelik vektörlerinin çıkarılması

	İris Parçasının Piksel Sayısı	Küçültülmüş Parçanın Piksel Sayısı	Öznitelik Vektöründeki Piksel Sayısı
DCT	21600(160x135)	864(32x27)	864(32x27)
DWT	21600(160x135)	986(34x29)	255(17x15)



Şekil 6.6. Matlab ile özellik vektörünün çıkarılması - DWT (2 Boyutlu)

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
1	0.5431	0.3373	0.3529	0.2461	0.2422	0.5941	0.5598	0.3255	0.1461	0.2824	0.6471	0.8088	0.9137	0.9255	0.8647	0.6314	0.5284	0.4912	0.4118	0.2784	0.1980	0.1775	0.1333	0.17
2																								
3																								
4																								

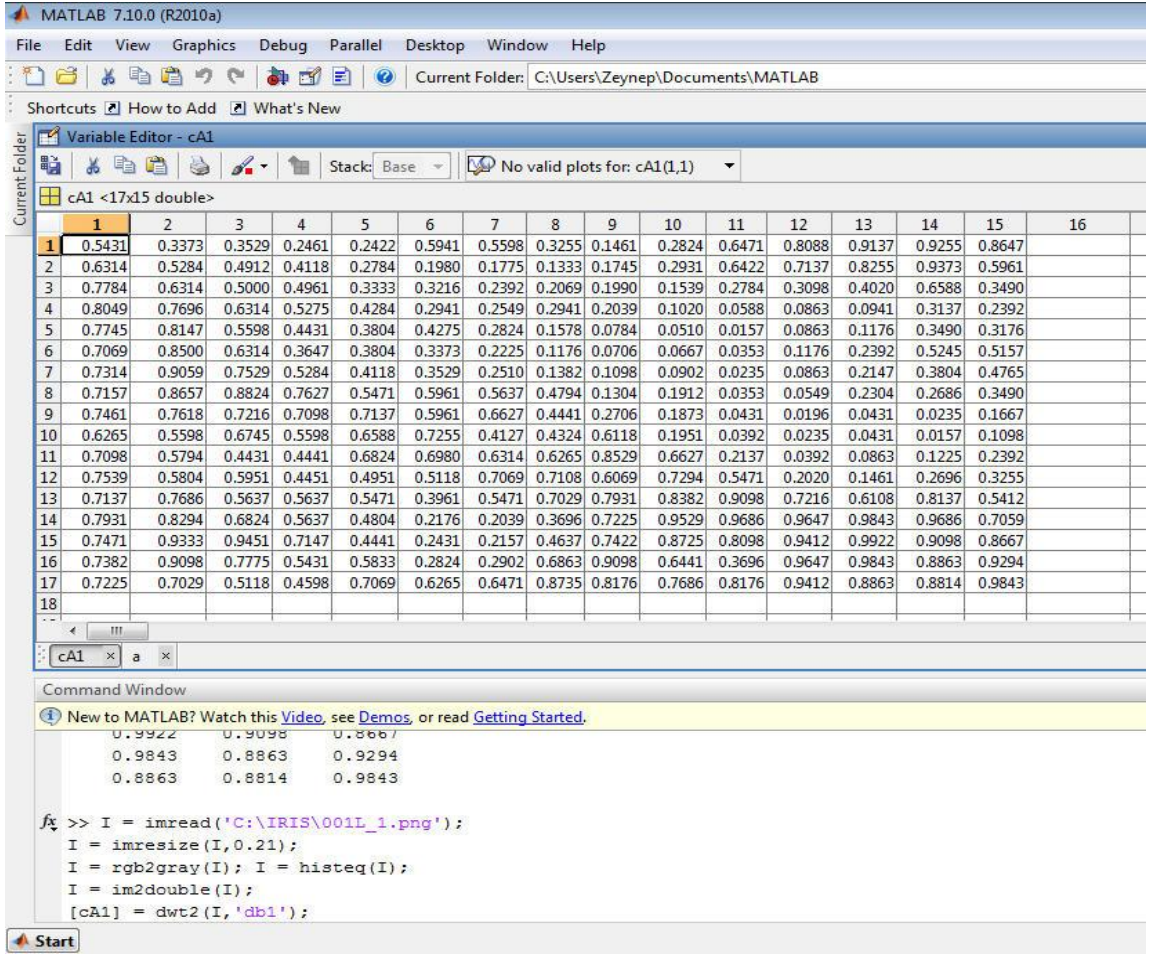
Şekil 6.7. İki boyutlu verinin bir boyuta çevrilmesi

6.1.3 Öznitelik Vektörüne BCO Uygulanması

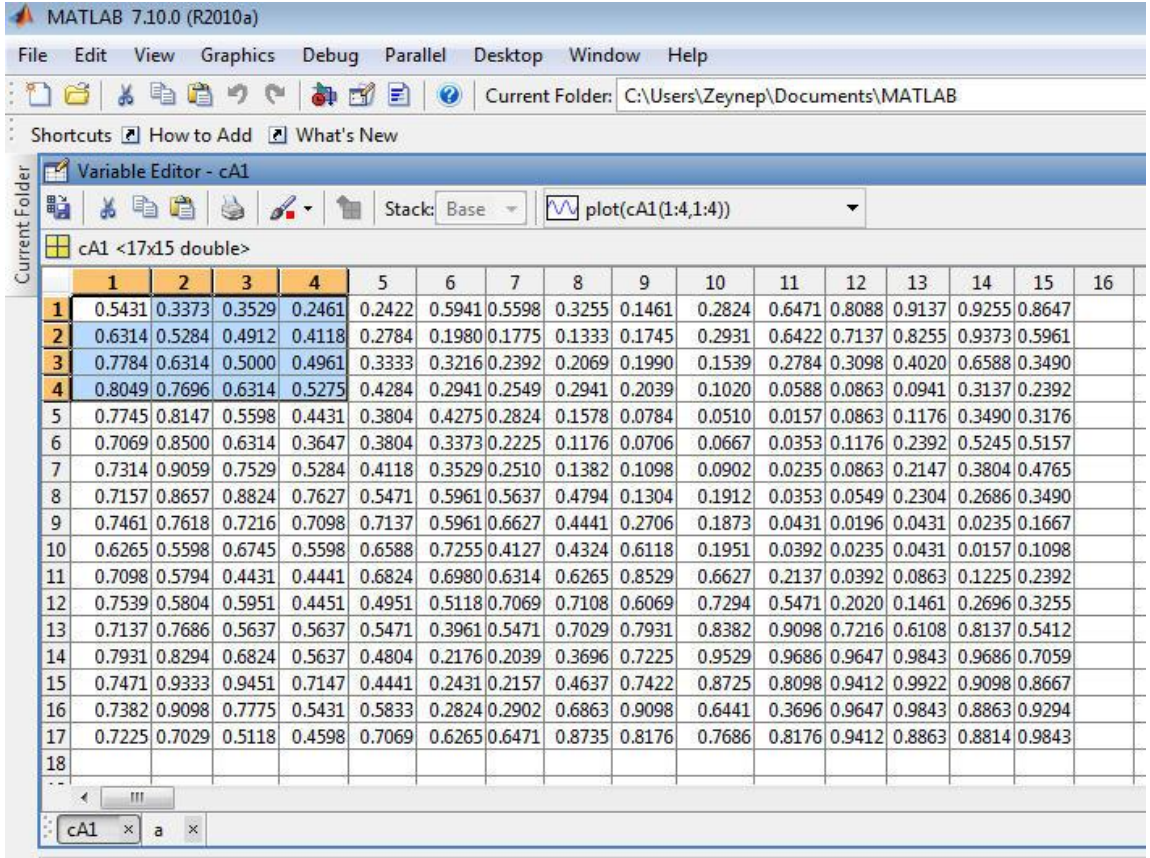
Bu çalışmanın asıl amacı, daha önce de belirtildiği gibi, iris tanımda (2-D) BKYSA yönteminin başarılı bir şekilde uygulanmasıdır. Tanınması istenen işaret iki boyutlu bir işarettir. Daha önce de işaret edildiği gibi, şu ana kadar olan çalışmalarda BKYSA sadece bir boyutlu işaretlerde kullanılmıştır. İris tanımda kullanılması için bir uyarılma gerektiği yine daha önce belirtmiştir. Bu kısımda öncelikle BKYSA bir boyutlu (1-D) işaretlerde nasıl uygulandığı açıklanacaktır. Sonra da iki boyutlu işaretlerde (2-D) bu sistemi kullanmak için yapılan uyarılmalardan bahsedilecektir.

DWT kullanarak elde ettiğimiz özellik vektörü 1x255 elemanlı bir işarettir. Bu işaretin toplam 255 elemanı vardır ve her bir $i \in N$ için bir değer tutulur. Tutulan bu değerler için bulanık kümelemeler yapılır ve her bir elemanın değerinin küme merkezlerine olan uzaklıklarına bağlı olarak bu indeksteki elemanın kümelere aitlik (üyelik) değeri bulunur. Bu değer, 0 ile 1 arasında değişen bir ağırlık değeridir. BK'yi iris tanıma probleminde tek boyutlu işaretlerdeki gibi direk olarak kullanmak mümkün değildir. DWT örneğinde, 17x15 boyutunda bir kartezyen koordinat sistemine çevrilmiş bir iris resmi elde edildi. Bu matris gerçekte iki boyutlu bir işaret tutmaktadır. Bu nedenle BKYSA, Iris tanıma işleminde kullanılmadan önce bu problemin doğasına uygun olacak bir şekilde uyarlanması gerekmektedir. Bu uyarılma sonunda tek boyutlu 17x15 adet özellik barındıran bir vektör elde edilmesi gerekmektedir. Burada en önemli şey, iris verisindeki *komşuluk bilgisinin* kaybolmaması veya en az kayıpla bu uyarılmanın gerçekleşmesidir.

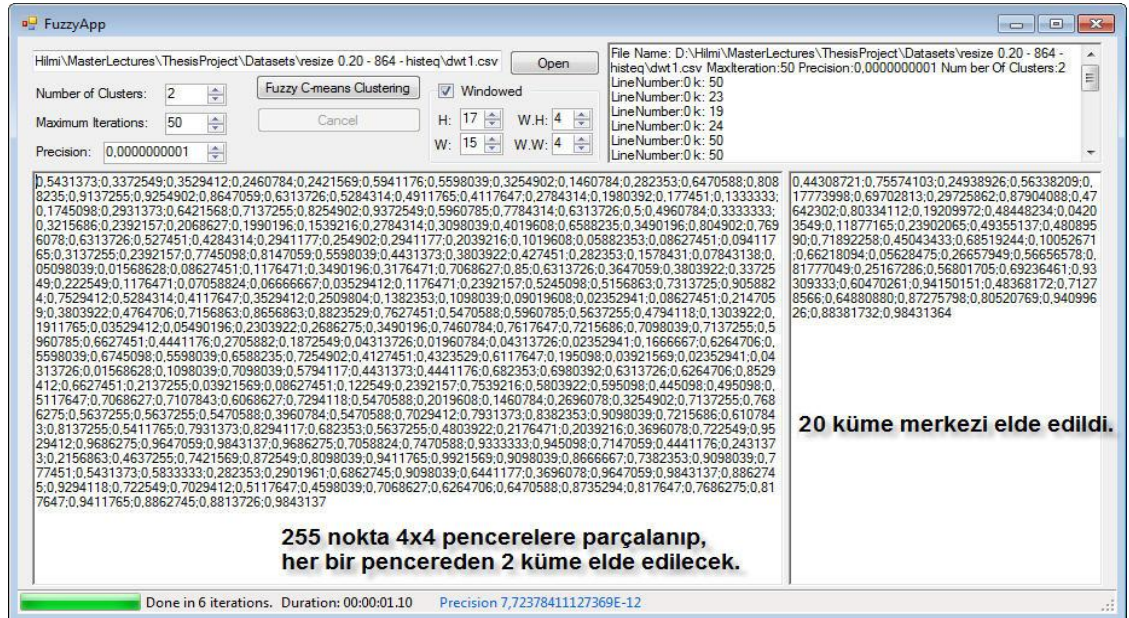
Öncelikle eldeki iki boyutlu işaretin bir şekilde tek boyutlu hale getirilmesi gerekiyor. Bunun için Şekil 6.6 ve Şekil 6.7’de olduğu gibi Şekil 6.8, Şekil 6.9, Şekil 6.10’da görülen model uygulanmıştır. Öncelikle kartezyen koordinat sistemine çevrilmiş olan iris resminden gürültüden arındırılmış (en aza indirgenilmiş) bir parça alınmıştır. Bu kısım, Şekil 6.9’da görüldüğü gibi dikdörtgen bölgelere ayrılır. Her bir dikdörtgen alan 4x4 eleman içerir. Sonra bu bölgedeki elemanlar 1x16 şekline dönüştürülür, yani tek boyutlu hale getirilmiş olur. BK işlemi, bu bölgelere uygulanacaktır. DWT örneğinde pencereler 4x4 seçildiğinde toplam 20 pencere oluşmakta ve her bir pencereden 2 küme merkezi alındığında toplam 40 küme merkezi elde edilmiştir. BK sonunda elde edilen küme merkezleri bir araya getirilerek 1x40 şeklinde YSA uygulamasına giriş olarak verilmiştir (Şekil 6.10). BCO algoritması Ek-1 de verilmiştir.



Şekil 6.8. Matlab ile özellik vektörünün çıkarılması - DWT (2 Boyutlu)



Şekil 6.9. DWT özellik vektöründen pencerelerin çıkarılması (Temsili gösterimdir, pencereler FCM uygulamasında bu şekilde elde edilmektedir.)



Şekil 6.10. Tek bir İris için BKYS'nın uygulaması

6.1.4 Sınıflandırma

Sınıflandırıcı olarak Yapay Sinir Ağları (YSA) kullanılmıştır. Öznitelik vektörü BK işleminden geçirildikten sonra YSA işlemine girdi olarak verilmektedir. YSA'da iki işlem yapılmaktadır. İlk işlemde tanınması istenen irislerin örneklerinin öznitelik vektörleri YSA'ya verilerek onun eğitilmesi sağlanır. Daha sonra ise test amacıyla bir iris öznitelik vektörü verilerek bu örneğin sahibi olan gözün YSA ile daha önceden öğrenilmiş olup olmadığına bakılır. Belli bir eşik değerini geçen benzeme oranına, tanıma gerçekleşmiştir denilir. Yaptığımız çalışmada genel olarak 40% oranını eşik değeri olarak kabul edip YTO ve YRO değerleri ona göre hesaplanmıştır. ROC eğrilerini çizerken ve AUC değeri hesaplanırken 0 ile 100 arasında değişen eşik değerleri kabul edilip ilgili hesaplamalar yapılmıştır.

Sınıflandırma işleminde hatanın geriye yayılması algoritması (back-propagation algorithm) kullanılmıştır. Transfer Fonksiyonu olarak Sigmoid kullanılmıştır.

Ağın giriş ve çıkış set sayıları vektörün boyutlarına bağlı olarak değişmektedir. Veritabanından alınan her bir kişiye ait 3 görüntüden bir tanesi test iki tanesi eğitim aşamasında kullanılmıştır. Örnek sayısı az olduğu için testleri artırma adına aynı görüntüler 1 tanesi eğitim 2 tanesi test aşamasında da kullanılmış ve sonuçlar karşılaştırılmıştır. Ağa verilen giriş sayısı özellik vektöründe kullanılan yöntem ve BK uygulanıp uygulanmamasına göre değişmektedir. Ağdaki giriş sayısı ve ağ katmanlarında sinir sayıları tablo 6.2'de detaylı olarak belirtilmiştir.

Tablo 6.2. YSA Ağ Yapısı

	BCO				YSA		
	Piksel Sayısı	Pencere Boyutu	Penceredeki Küme Sayısı**	Toplam Küme Sayısı	Momentum	Eğitim Oranı	Nöron Sayısı
DCT	864(32x27)	-	-	-	0,75	0,35	864-433-64
DCT*	864(32x27)	-	-	-	0,75	0,75	864-433-64
DWT	255(17x15)	-	-	-	0,75	0,75	255-129-64
DWT*	255(17x15)	-	-	-	0,75	0,5	255-129-64
DCT-BCO	864(32x27)	3x3	2	198	0,75	0,75	198-100-64
DCT-BCO*	864(32x27)	3x3	2	198	0,75	0,75	198-100-64
DCT-BCO	864(32x27)	4x4	2	112	0,75	0,75	112-57-64
DCT-BCO*	864(32x27)	4x4	2	112	0,65	0,85	112-57-64
DCT-BCO	864(32x27)	5x5	2	84	0,75	0,75	84-43-64
DCT-BCO*	864(32x27)	5x5	2	84	0,5	0,75	84-43-64
DWT-BCO	255(17x15)	3x3	2	60	0,01	0,9	60-33-64
DWT-BCO*	255(17x15)	3x3	2	60	0,01	0,9	60-33-64
DWT-BCO	255(17x15)	4x4	2	40	0,01	0,9	40-33-64
DWT-BCO*	255(17x15)	4x4	2	40	0,55	0,85	40-33-64
DWT-BCO	255(17x15)	5x5	2	24	0,85	0,85	24-33-64
DWT-BCO*	255(17x15)	5x5	2	24	0,15	0,95	24-33-64

*1 iris eğitim 2 iris test için kullanılmıştır. **Her bir penceredeki küme sayısıdır.

Eğitim doğrulama oranı tüm testlerde 100 % e kadar eğitime devam edilmiştir.

MLP sinir ağı modelinde gizli katman hücre sayısı belirlenirken tecrübeye dayalı yöntem kullanılmıştır. Bunun içinde giriş ve çıkış sayılarından büyük olanın yarısının bir fazla alınmıştır. Tablo 6.2’de bu değerleri görebilirsiniz.

Eğitim aşamasında öğrenme metoduna bağlı olarak öğrenme oranı ve momentum parametreleri belirlenmelidir. Bu parametreler ağın öğrenme hızını ve performansını etkilemektedir. Formül 6.1’de MLPBP Algoritmasının ana formülü görülmektedir. Bu formüldeki ε öğrenme oranı, α ise momentumu temsil etmektedir.

$$\Delta w_{kj}(n+1) = \varepsilon \delta_o y_j + \alpha \Delta w_{kj}(n) \quad (6.1)$$

Öğrenme oranı; ağırlıkların değişim miktarını belirlemektedir. Eğer büyük değerler seçilirse o zaman yerel çözümler arasında ağırların dolaşması ve dalgalanma yaşaması söz konusu olmaktadır. Küçük değerler seçilmesi ise öğrenme zamanını artırmaktadır. Öğrenme oranı için en ideal değer şeklinde bir tanımlamak mümkün değildir. Probleme göre farklı değerlerin iyi sonuçlar verdiği bilinmektedir. Bizim çalışmamızda da bir birinden farklı değerler kullanılmıştır.

Momentum katsayısı; bir önceki iterasyondaki değişimin belirli bir oranının yeni değişim miktarına eklenmesi olarak görülmektedir. Bu özellikle yerel çözümlere takılan ağırların bir sıçrama ile daha iyi sonuçlar bulmasını sağlamak amacı ile önerilmiştir. Bu değer küçük olması yerel çözümlerden kurtulmayı zorlaştırabilir. Çok büyük değerler için ise tek bir çözüme ulaşmada sorunlar yaşanabilir. Tecrübeler bu değer 0,6–0,8 arasında seçilmesinin uygun olacağını gösterse de farklı değerlerle de başarılı sonuçlar alınabilmektedir. Momentumun problemin niteliğine göre belirlenmesi daha doğrudur. Daha küçük değerler ile başarılı sonuçların alındığını gösteren örnekleri görmek de mümkündür (Koçer, H. E., 2007). Tablo 6.2’de bu çalışmadaki değerler gösterilmektedir. Tez çalışmasında kullanılan uygulamaların ekran görüntüleri Ek-2 de verilmiştir.

7 TEST

Neredeyse biyometrik sayısı kadar performans ölçü birimleri vardır. Ne yazık ki bir sistemin ne kadar iyi çalışacağını gösteren tek bir ölçü yoktur. Çoklu ölçü birimlerinin analizi her bir teknolojinin güçlü ve zayıf yönlerini tespit etmede gereklidir. Ayrıca testlerin gerçek dünyada var olan çevresel etkilerin ve uygulamaların olduğu bir ortamda test edilmesi de çok önemlidir. Yapılan çalışmada testler için kullandığımız ölçütler:

- Yanlış Tanıma Oranı (YTO),
- Yanlış Reddetme Oranı (YRO),
- Doğru Tanıma Oranı (DTO),
- ROC eğrisinin altında kalan alan (AUC),

Testte kullanılan bilgisayarın özellikleri: Intel core i5 - 460M işlemci, 3 GB RAM, windows 7 işletim sistemi.

Yapılan çalışmanın testleri de adım adım bu bölümde belirtilmiştir.

7.1 ROC

Sinyal algılama teorisinde, alıcı işletim karakteristiği (orijinal adıyla; Receiver Operating Characteristic - ROC) ya da sade biçimde ROC eğrisi olarak tanımlanmaktadır. ROC eğrisi, ikili sınıflandırma sistemlerinde ayırım eşik değerinin farklılık gösterdiği durumlarda, hassasiyetin kesinliliğe olan oranıyla ortaya çıkmaktadır. ROC daha basit anlamda doğru pozitiflerin, yanlış pozitiflere olan kesri olarak da ifade edilebilir. (Swets, 1996).

Her sınıflandırma işleminde yapıldığı gibi, metotlar kesinlik (yanlış pozitifleri eleme kabiliyeti) ve hassasiyet (doğru pozitifleri tespit etme kabiliyeti) arasındaki dengeyi kurmakla uğraşmaktadır. Veri kümesindeki pozitif ve negatif örnekler, eşit bir şekilde dağılım göstermediğinden dolayı, doğrudan kesinlik ve hassasiyet ölçütlerinden önce, ROC adı verilen eğri, kesinlik ve hassasiyet arasındaki dengeyi değerlendirmek için kullanılmıştır. ROC eğrisi altında kalan alan (AUC) ROC puanı olarak tanımlanır. ROC eğrisi değişen sınıflandırma eşik değerlerine göre doğru pozitiflerin sayısının, yanlış pozitiflerin bir fonksiyonu olarak çizilmesiyle oluşmaktadır. AUC 1 (bir) olduğunda anlamı, pozitifler mükemmel bir şekilde negatiflerden ayrılmıştır, olmaktadır. AUC 0 (sıfır) olduğunda ise herhangi bir pozitif bulunamadı anlamına gelir. Tablo 7.2’de örnek bir ROC analizi görülmektedir.

Geniş olarak tıp, radyoloji, psikoloji ve benzer alanlarda yüzyıllardır kullanılmaktadır, günümüzde makine öğrenme teknikleri ve veri madenciliği alanlarında da kullanılmaktadır (ROC, 2011).

Tablo 7.1. ROC analizi

Cutoff Value	True Positives	True Negatives	False Positives	False Negatives	Specificity	Sensitivity	Efficiency	Accuracy	Positive Predictive Value	Negative Predictive Value	False Positive Rate	Phi
0,00000	64	0	10	0	0,00000	1,00000	0,50000	0,86486	0,86486	1,00000	1,00000	0,00000
0,20000	63	9	1	1	0,90000	0,98438	0,94219	0,97297	0,98438	0,90000	0,10000	0,88594
0,80000	42	10	0	22	1,00000	0,65625	0,82813	0,70270	1,00000	0,31250	0,00000	0,45286
1,00000	0	10	0	64	1,00000	0,00000	0,50000	0,13514	1,00000	0,13514	0,00000	0,00000
0,40000	61	10	0	3	1,00000	0,95313	0,97656	0,95946	1,00000	0,76923	0,00000	0,85626
0,60000	59	10	0	5	1,00000	0,92188	0,96094	0,93243	1,00000	0,66667	0,00000	0,78395

ROC analizinde hesaplanan değerler şu şekildedir:

- Cutoff Value: Eşik Değeri
- Positives: Pozitif kümedeki eleman sayısı (P). Sisteme tanıtılan elemanlar.
- Negatives: Negatif kümedeki eleman sayısı (N).
- True Positives: Pozitif küme içindeki örneğin Pozitif küme içinde olarak tanınması (TP).
- True Negatives: Negatif küme içindeki örneğin Negatif küme içinde olarak tanınması (TN).
- False Positives: Negatif küme içindeki örneğin Pozitif küme içinde olarak tanınması (FP).

- False Negatives: Pozitif küme içindeki örneğin Negatif küme içinde olarak tanınması (FN).
- Specificity: $SPC = TN / N = TN / (FP + TN) = 1 - FPR$
- Sensitivity: $TPR = TP / P = TP / (TP + FN)$
- Efficiency: Verimlilik.
- Accuracy: Doğruluk oranı. ($ACC = (TP + TN) / (P + N)$)
- Positive Predictive Value: Hassasiyet, $TPR = TP / P = TP / (TP + FN)$
- Negative Predictive Value: Hassasiyet, $NPV = TN / (TN + FN)$
- False Positive Rate: $FPR = FP / N = FP / (FP + TN)$

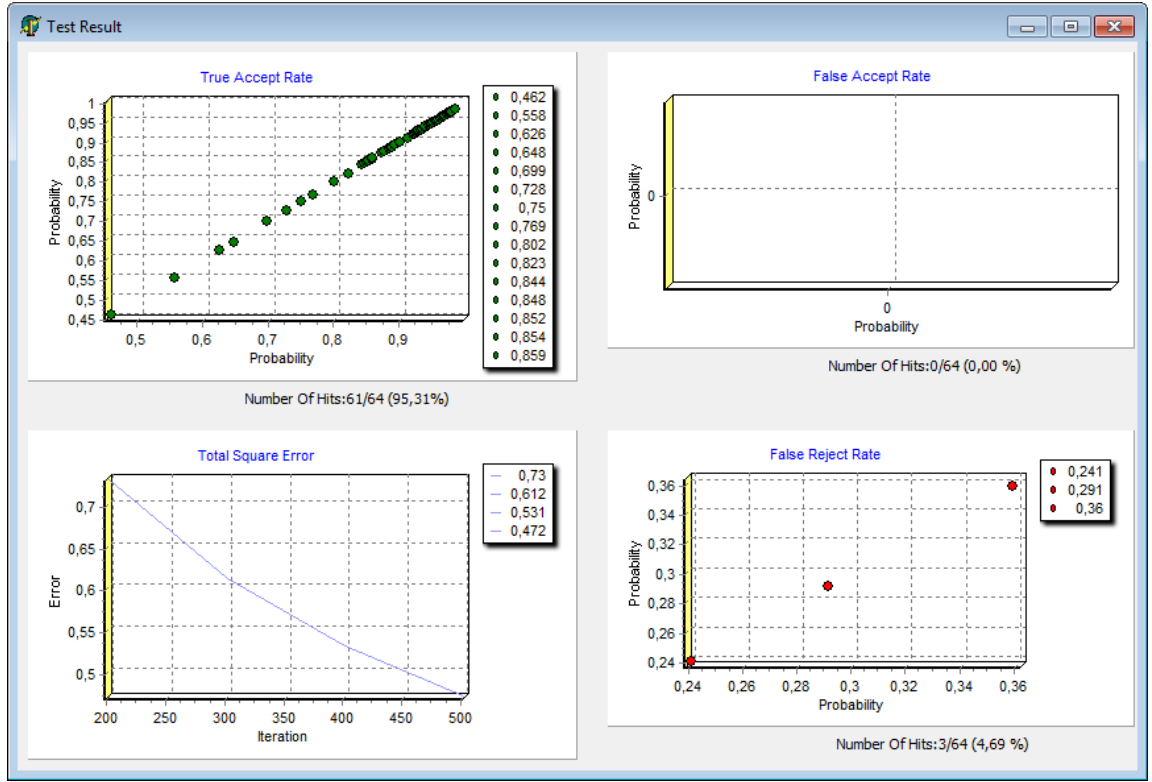
7.2 DCT-YSA Test Sonuçları

Yapılan ilk testte DCT ile özellik vektörü çıkarıp klasik YSA ile sınıflandırma yapılmıştır. Testlerde her bir kişi için 3 iris örneği ile çalışılmıştır. İki şekilde test yapılmıştır. İlkinde 2 iris eğitim 1 iris test için, diğerinde ise 1 iris eğitim 2 iris test için kullanılmıştır. 2 iris eğitim 1 iris test için kullanılarak yapılan testlerde (Tablo 7.2); eğitim süresi 5 dk 34 sn, YTO % 0, YRO % 4,69, DTO % 95,31, AUC ise 0,97 şeklinde gerçekleşmiştir. Aşağıdaki tablo ve şekillerde test sonuçları detaylı olarak verilmiştir.

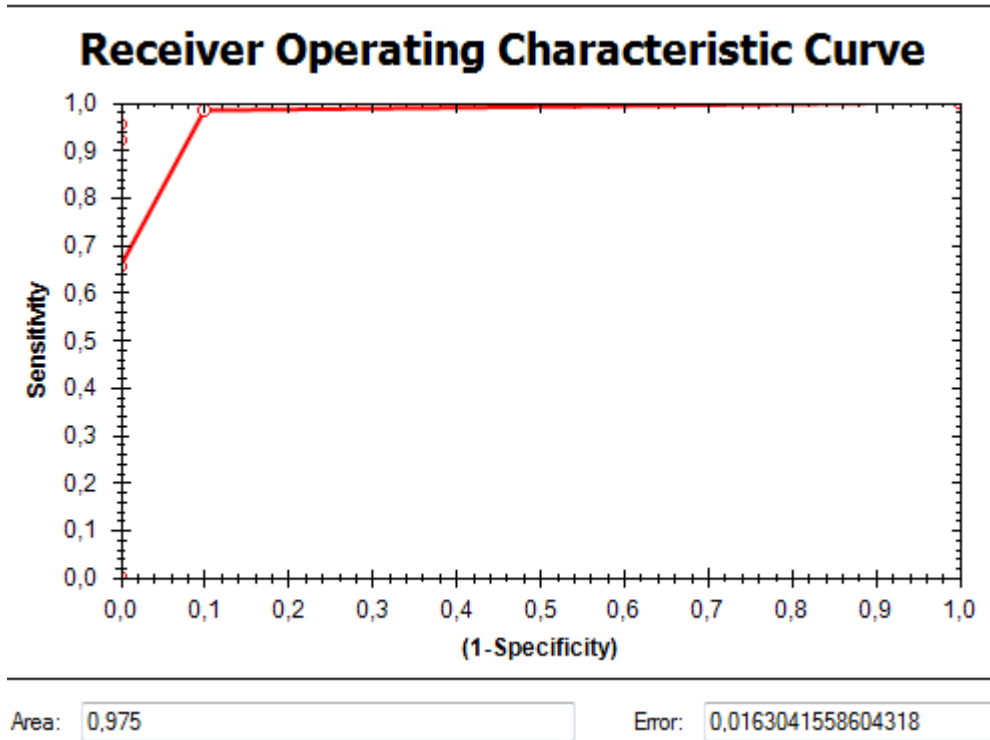
Tablo 7.2. DCT-YSA test sonuçları

2 iris eğitim 1 iris testte kullanıldı														
	Eğitim							Test				YSA		
	Piksel Sayısı	YTO (%)	YRO (%)	Süre (dk:sn)	DTO (%)	İterasyon	Hata	YTO (%)	YRO (%)	AUC	DTO (%)	Momentum	Eğitim Oranı	Nöron Sayısı
DCT	864 (32x27)	0,00	0,00	05:34	100,00	517	1,18	0,00	4,69	0,97	95,31	0,75	0,35	433

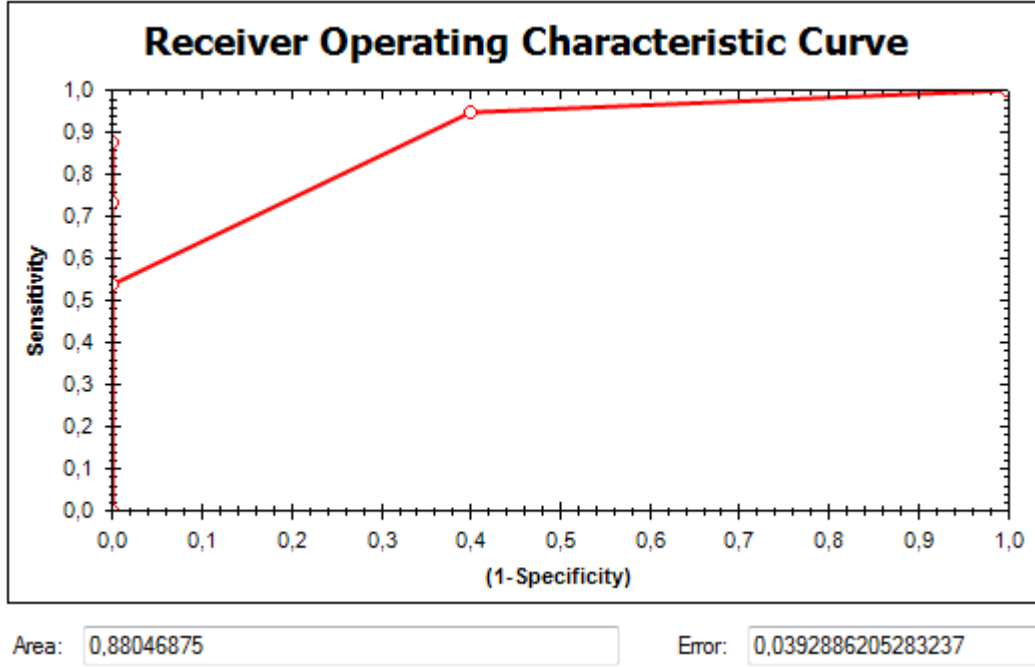
1 iris eğitim 2 iris testte kullanıldı														
	Eğitim							Test				YSA		
	Piksel Sayısı	YTO (%)	YRO (%)	Süre (dk:sn)	DTO (%)	İterasyon	Hata	YTO (%)	YRO (%)	AUC	DTO (%)	Momentum	Eğitim Oranı	Nöron Sayısı
DCT	864 (32x27)	0,00	0,00	04:40	100,00	750	0,49	0,00	10,94	0,88	89,06	0,75	0,75	433



Şekil 7.1. DCT-YSA – Test sonuçları ve hatanın kareleri toplamı (2 iris eğitim – 1 iris test)



Şekil 7.2. DCT-YSA ROC Eğrisi (2 iris eğitim – 1 iris test)



Şekil 7.3. DCT-YSA ROC Eğrisi (1 iris eğitim – 2 iris test)

Cutoff Value	True Positives	True Negatives	False Positives	False Negatives	Specificity	Sensitivity	Efficiency	Accuracy	Positive Predictive Value	Negative Predictive Value	False Positive Rate	Phi
0,00000	64	0	10	0	0,00000	1,00000	0,50000	0,86486	0,86486	1,00000	1,00000	0,00000
0,20000	63	9	1	1	0,90000	0,98438	0,94219	0,97297	0,98438	0,90000	0,10000	0,88594
0,80000	42	10	0	22	1,00000	0,65625	0,82813	0,70270	1,00000	0,31250	0,00000	0,45286
1,00000	0	10	0	64	1,00000	0,00000	0,50000	0,13514	1,00000	0,13514	0,00000	0,00000
0,40000	61	10	0	3	1,00000	0,95313	0,97656	0,95946	1,00000	0,76923	0,00000	0,85626
0,60000	59	10	0	5	1,00000	0,92188	0,96094	0,93243	1,00000	0,66667	0,00000	0,78395

Şekil 7.4. ROC Analiz Sonuçları (2 iris eğitim – 1 iris test)

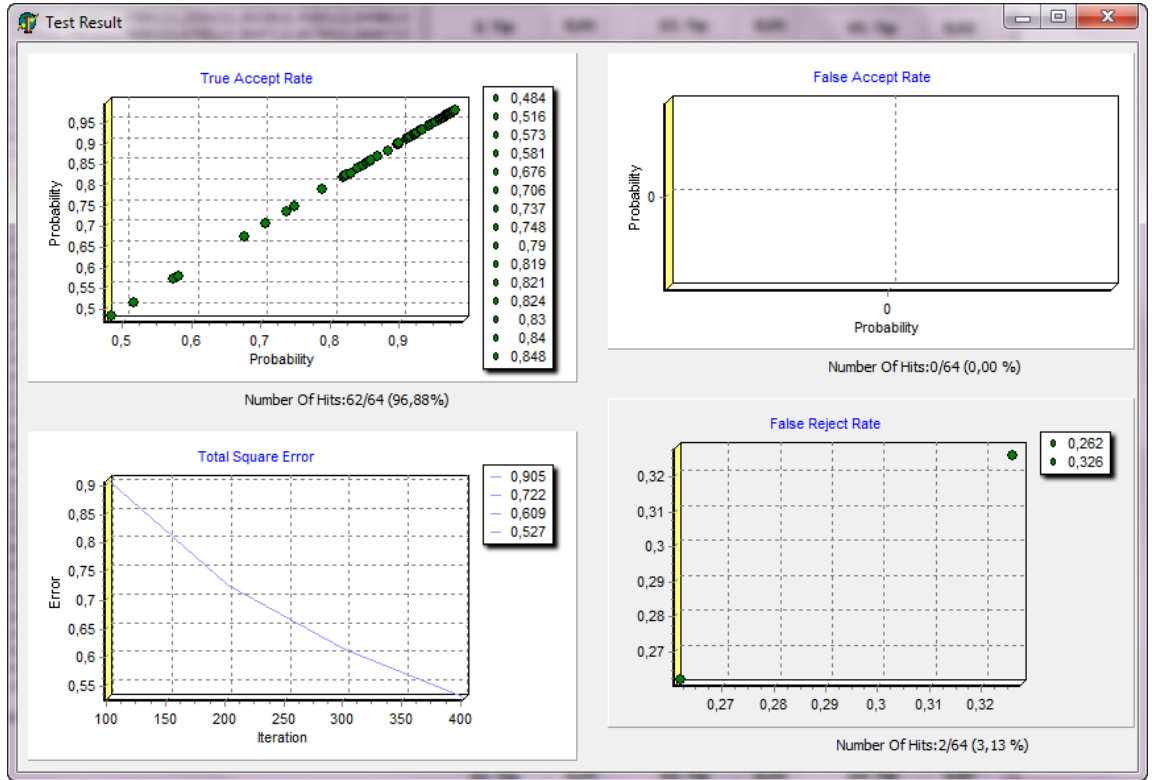
7.3 DWT –YSA Test Sonuçları

Yapılan testte DWT ile özellik vektörü çıkarıp (Şekil 6.6) klasik YSA ile sınıflandırma yapılmıştır. 2 iris eğitim 1 iris test için kullanılarak yapılan testlerde (Tablo 7.3); eğitim süresi 20 sn, YTO % 0, YRO % 3,13, DTO % 96,88, AUC ise 0,99 şeklinde gerçekleşmiştir. Aşağıdaki tablo ve şekillerde test sonuçları detaylı olarak verilmiştir.

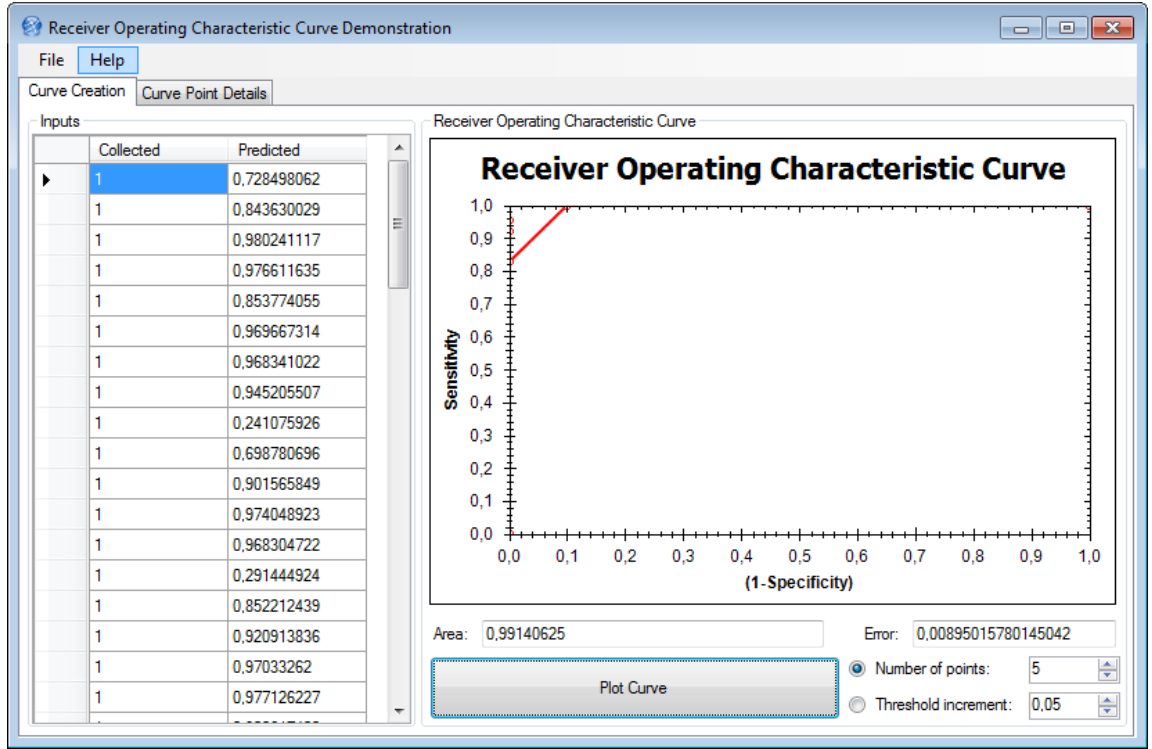
Tablo 7.3. DWT-YSA test sonuçları

2 iris eğitim 1 iris testte kullanıldı														
	Eğitim							Test				YSA		
	Piksel Sayısı	YTO (%)	YRO (%)	Süre (dk:sn)	DTO (%)	İterasyon	Hata	YTO (%)	YRO (%)	AUC	DTO (%)	Momentum	Eğitim Oranı	Nöron Sayısı
DWT	255 (17x15)	0,00	0,00	00:20	100,00	845	0,5	0,00	3,13	0,99	96,88	0,75	0,75	129

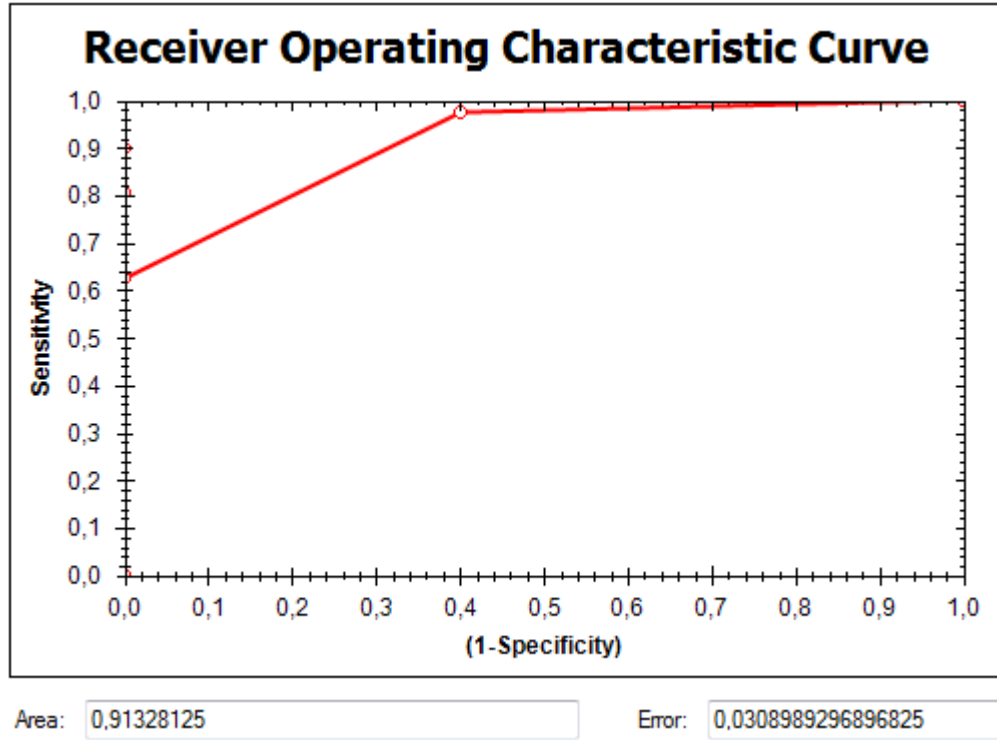
1 iris eğitim 2 iris testte kullanıldı														
	Eğitim							Test				YSA		
	Piksel Sayısı	YTO (%)	YRO (%)	Süre (dk:sn)	DTO (%)	İterasyon	Hata	YTO (%)	YRO (%)	AUC	DTO (%)	Momentum	Eğitim Oranı	Nöron Sayısı
DWT	255 (17x15)	0,00	0,00	01:04	100,00	2739	0,09	0,00	9,38	0,91	90,63	0,75	0,50	129



Şekil 7.5. DWT-YSA – Test sonuçları ve hatanın kareleri toplamı (2 iris eğitim – 1 iris test)



Şekil 7.6. DWT-YSA için ROC Grafiği (2 iris eğitim 1 iris test)



Şekil 7.7. DWT-YSA için ROC Grafiği (1 iris eğitim 2 iris test)

7.4 DCT-BKYSA Test Sonuçları

Yapılan testte DCT ile özellik vektörü çıkarıp BKYSA ile sınıflandırma yapılmıştır. BKYSA ile yapılan testlerde kümeleme yapılırken pencere boyutları 3x3, 4x4 ve 5x5 şeklinde ayarlanmış ve her 3 şekilde de testler yapılmıştır. 2 iris eğitim 1 iris test için kullanılan ve 3x3 pencereleme yapılan testlerde; eğitim süresi 38 sn, YTO % 23,44 YRO % 32,81 DTO % 43,75 AUC ise 0,67 şeklinde gerçekleşmiştir. 2 iris eğitim 1 iris test için kullanılan ve 4x4 pencereleme yapılan testlerde ise; eğitim süresi 6 dk 32 sn, YTO % 34,38 YRO % 34,38 DTO % 31,25 AUC ise 0,26 şeklinde gerçekleşmiştir. DCT-BKYSA yöntemi ile yapılan testlerde sonuçlar beklenilenden daha kötü çıkmıştır. Kötü çıkma nedenleri sonuç bölümünde yorumlanmıştır. Aşağıdaki tablo ve şekillerde test sonuçları verilmiştir.

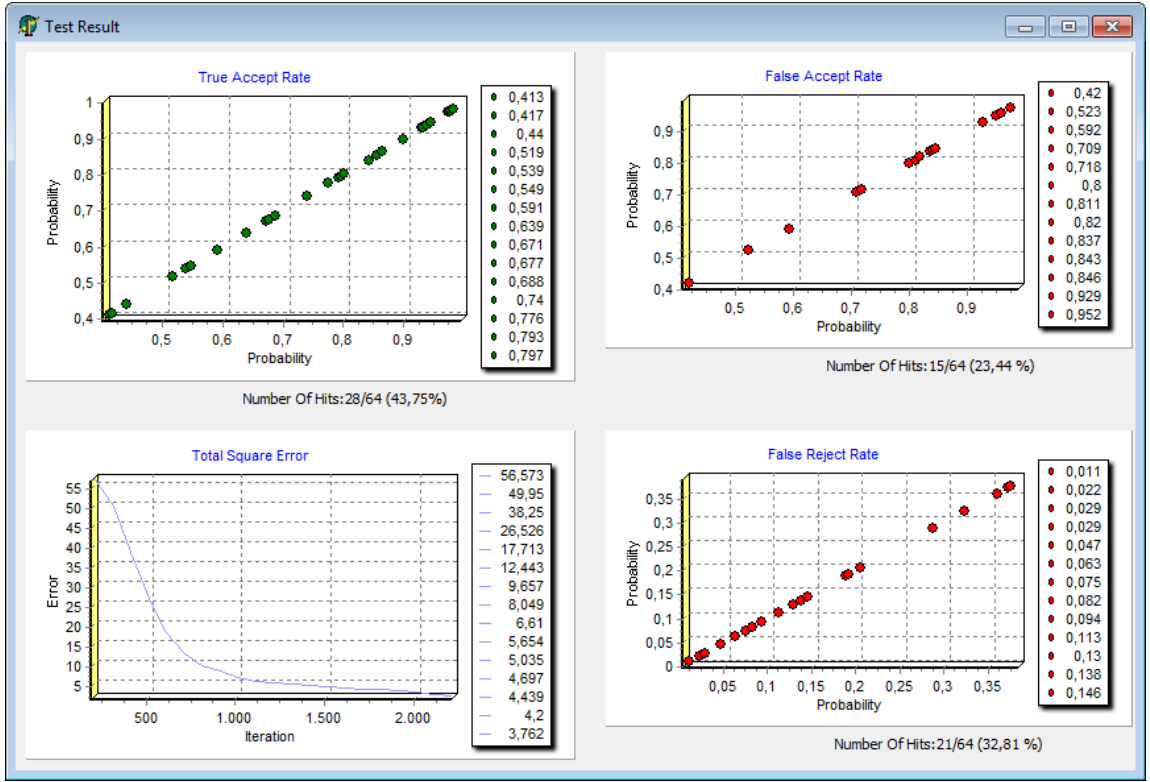
Tablo 7.4. DCT-BKYSA test sonuçları

2 iris eğitim 1 iris testte kullanıldı																		
	Piksel Sayısı	Eğitim					Test					YSA			BCO		Toplam Küme Sayısı	
		YTO (%)	YRO (%)	Süre (dk:sn)	DTO (%)	İterasyon	Hata	YTO (%)	YRO (%)	AUC	DTO (%)	Momentum	Eğitim Oranı	Nöron Sayısı	Pencere Boyutu	Pencere Küme Sayısı*		
DCT-BCO	864 (32x27)	0,00	0,00	01:06	100,00	2202	1,85	23,44	32,81	0,67	43,75		0,3	0,75	198-100-64	3x3	2	198
DCT-BCO	864 (32x27)	0,00	0,00	06:32	100,00	29782	2,23	34,38	34,38	0,26	31,25		0,75	0,75	112-57-64	4x4	2	112
DCT-BCO	864 (32x27)	0,00	0,00	08:10	100,00	53322	3,26	40,63	34,38	0,13	25		0,65	0,85	84-43-64	5x5	2	84

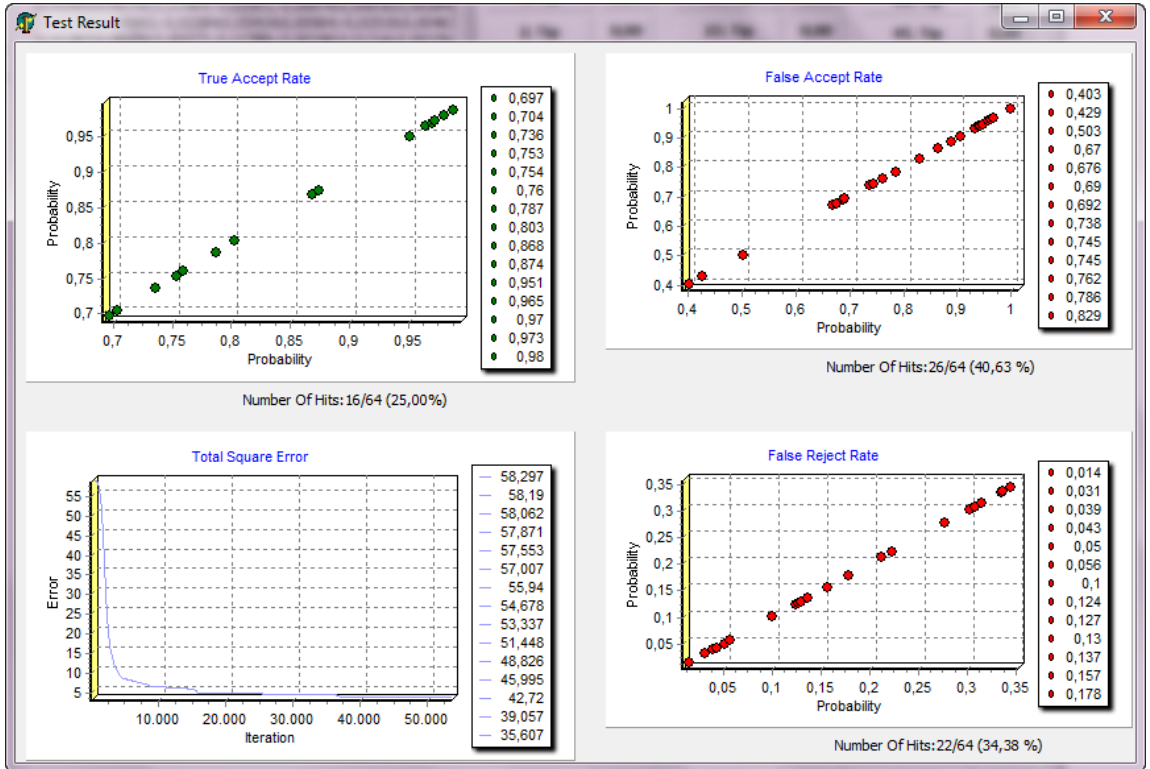
*Her bir penceredeki küme sayısı

1 iris eğitim 2 iris testte kullanıldı																		
	Piksel Sayısı	Eğitim					Test					YSA			BCO		Toplam Küme Sayısı	
		YTO (%)	YRO (%)	Süre (dk:sn)	DTO (%)	İterasyon	Hata	YTO (%)	YRO (%)	AUC	DTO (%)	Momentum	Eğitim Oranı	Nöron Sayısı	Pencere Boyutu	Pencere Küme Sayısı*		
DCT-BCO	864 (32x27)	0,00	0,00	00:38	100,00	2547	0,65	28,13	51,56	0,61	20,31		0,15	0,75	198-100-64	3x3	2	198
DCT-BCO	864 (32x27)	0,00	0,00	01:48	100,00	16300	0,58	44,53	39,84	0,44	15,63		0,75	0,75	112-57-64	4x4	2	112
DCT-BCO	864 (32x27)	0,00	0,00	03:30	100,00	45166	0,58	49,22	38,28	0,44	12,50		0,5	0,75	84-43-64	5x5	2	84

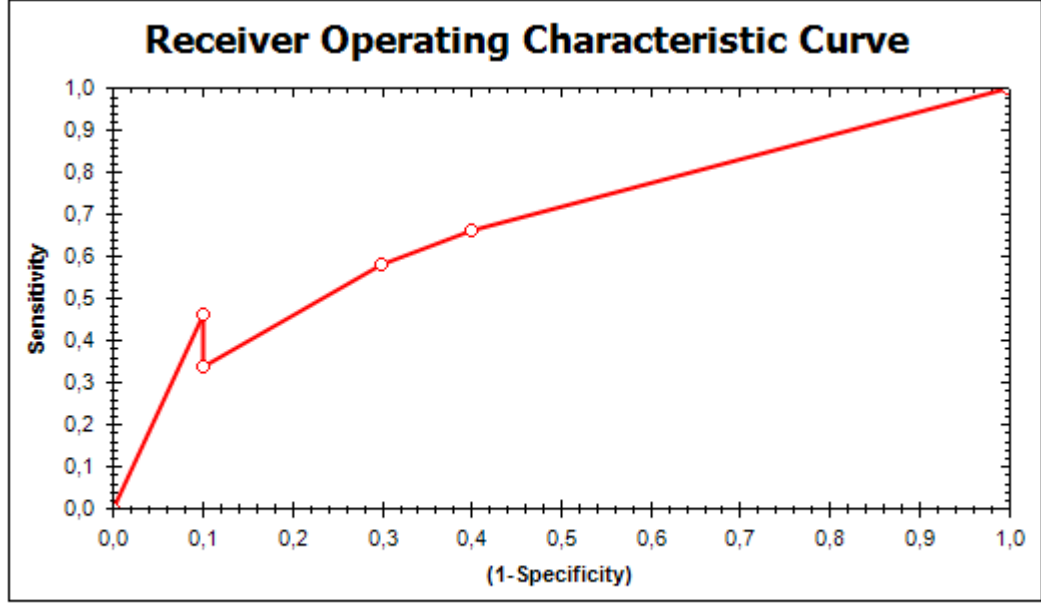
*Her bir penceredeki küme sayısı



Şekil 7.8. DCT-BKYSA – Test sonuçları ve hatanın kareleri toplamı (2 iris eğitim 1 iris test, 3 x 3 pencerelemeli)



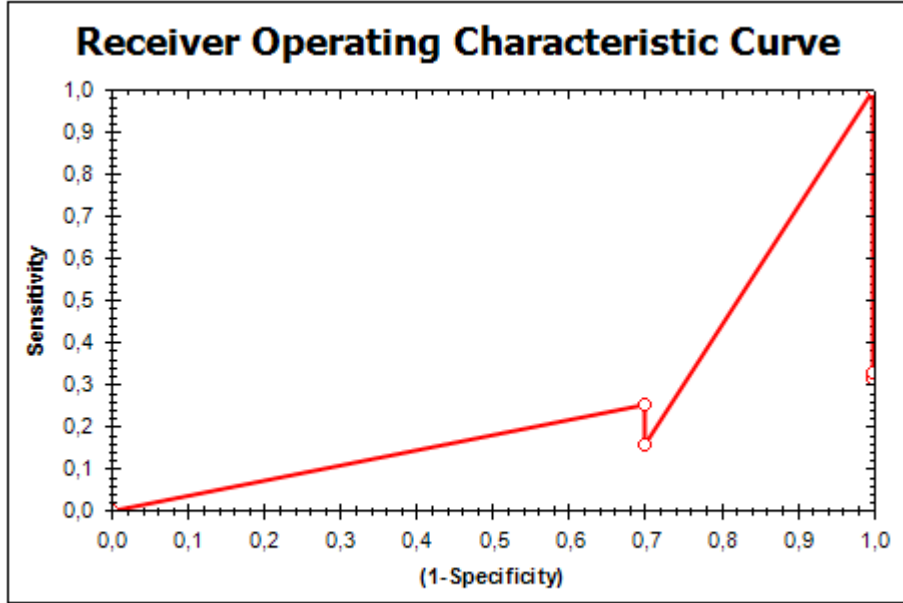
Şekil 7.9. DCT-BKYSA – Test sonuçları ve hatanın kareleri toplamı (2 iris eğitim 1 iris test, 4 x 4 pencerelemeli)



Area: 0,675675675675676

Error: 0,0816390285640348

Şekil 7.10. ROC Grafiği (2 iris eğitim 1 iris test için, 3 x 3 pencere)



Area: 0,2609375

Error: 0,094499371404374

Şekil 7.11. ROC Grafiği (2 iris eğitim 1 iris test için, 4 x 4 pencere)

7.5 DWT-BKYSA Test Sonuçları

Yapılan testte DWT ile özellik vektörü çıkarıp (Şekil 6.6) BKYSA ile sınıflandırma yapılmıştır. BKYSA ile yapılan testlerde kümeleme yapılırken pencere boyutları 3x3, 4x4 ve 5x5 şeklinde ayarlanmış ve her iki şekilde de testler yapılmıştır. 2 iris eğitim 1 iris test için kullanılan ve 3x3 pencereleme yapılan testlerde (Tablo 7.5); eğitim süresi 4 sn, YTO % 0 YRO % 1,56 DTO % 98,44 AUC ise 0,984 şeklinde gerçekleşmiştir. 2 iris eğitim 1 iris test için kullanılan ve 4x4 pencereleme yapılan testlerde (Tablo 7.5); eğitim süresi 8 sn, YTO % 0 YRO % 3,13 DTO % 96,88 AUC ise 0,89 şeklinde gerçekleşmiştir. DWT-BKYSA yöntemi ile yapılan testlerde sonuçlar beklenildiği gibi iyi çıkmıştır. Test sonuçları klasik YSA ile karşılaştırıldığında; süre kısalmış ve tanıma oranları artmıştır. Tezin amaçlarından biri olan BKYSA'nın 2 boyutlu biyometriklere uygulanması bu testte başarıyla gerçekleştirilmiştir. Aşağıdaki tablo ve şekillerde test sonuçları ayrıntılı olarak verilmiştir.

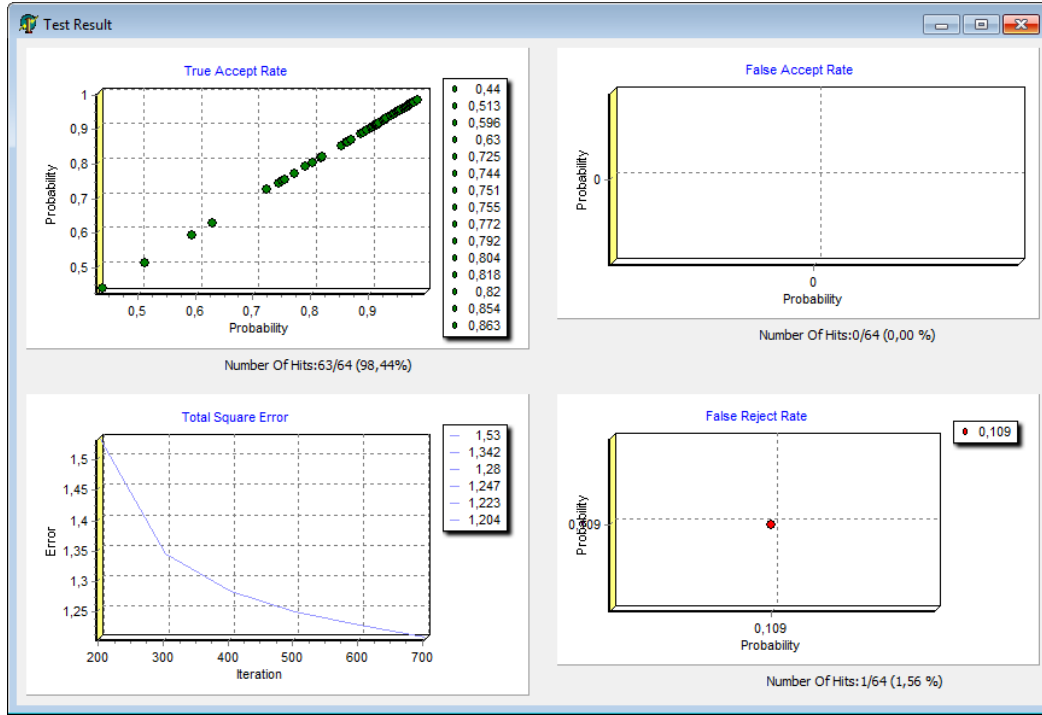
Tablo 7.5. DWT-BKYSA test sonuçları

	2 iris eğitim 1 iris testte kullanıldı																
	Piksel Sayısı	Eğitim					Test				YSA			BCO			
		YTO (%)	YRO (%)	Süre (dk:sn)	DTO (%)	İterasyon	Hata	YTO (%)	YRO (%)	AUC	DTO (%)	Momentum	Eğitim Oranı	Nöron Sayısı	Pencere Boyutu	Pencere Küme Sayısı*	Toplam Küme Sayısı
DWT-BCO	255 (17x15)	0,00	0,00	00:04	100,00	722	1,2	0	1,56	0,98	98,44	0,15	0,75	60-33-64	3x3	2	60
DWT-BCO	255 (17x15)	0,00	0,00	00:08	100,00	1363	1,2	0	3,13	0,89	96,88	0,01	0,90	40-33-64	4x4	2	40
DWT-BCO	255 (17x15)	0,00	0,00	00:06	100,00	1215	1,32	6,3	7,81	0,77	85,94	0,55	0,85	24-33-64	5x5	2	24

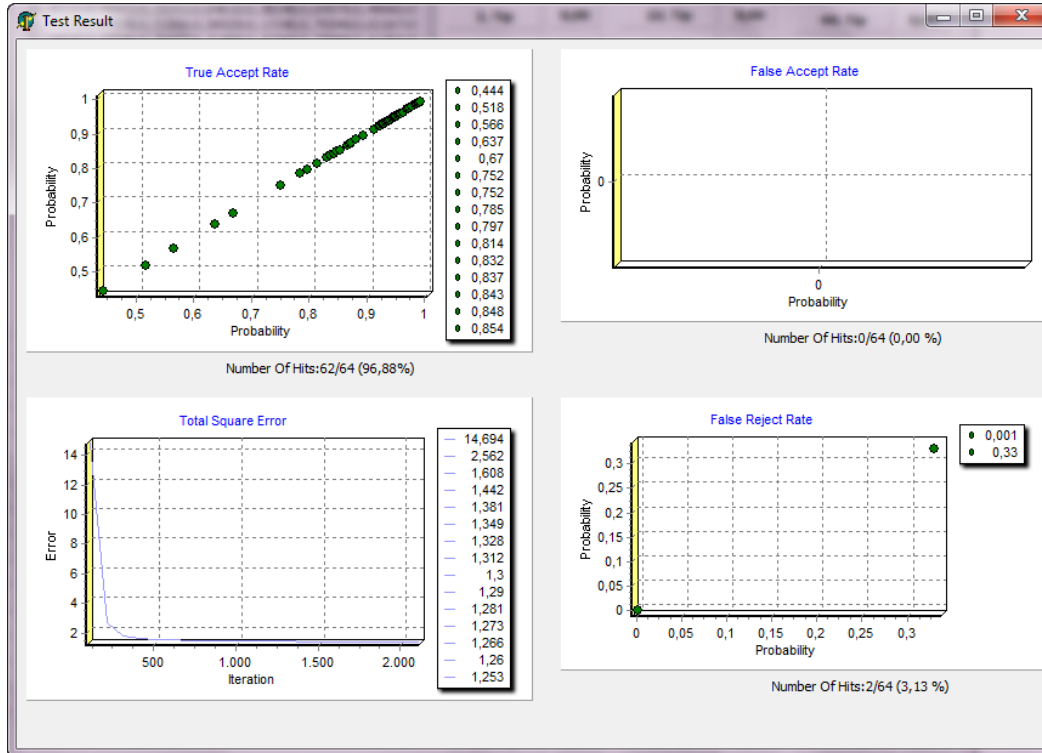
*Her bir penceredeki küme sayısı

	1 iris eğitim 2 iris testte kullanıldı																
	Piksel Sayısı	Eğitim					Test				YSA			BCO			
		YTO (%)	YRO (%)	Süre (dk:sn)	DTO (%)	İterasyon	Hata	YTO (%)	YRO (%)	AUC	DTO (%)	Momentum	Eğitim Oranı	Nöron Sayısı	Pencere Boyutu	Pencere Küme Sayısı*	Toplam Küme Sayısı
DWT-BCO	255 (17x15)	0,00	0,00	00:15	100,00	4481	0,4	2,3	3,13	0,91	94,53	0,15	0,75	60-33-64	3x3	2	60
DWT-BCO	255 (17x15)	0,00	0,00	00:05	100,00	1853	0,54	3,9	7,03	0,84	89,06	0,85	0,85	40-33-64	4x4	2	40
DWT-BCO	255 (17x15)	0,00	0,00	00:07	100,00	2845	0,6	12	13,3	0,82	75	0,15	0,95	24-33-64	5x5	2	24

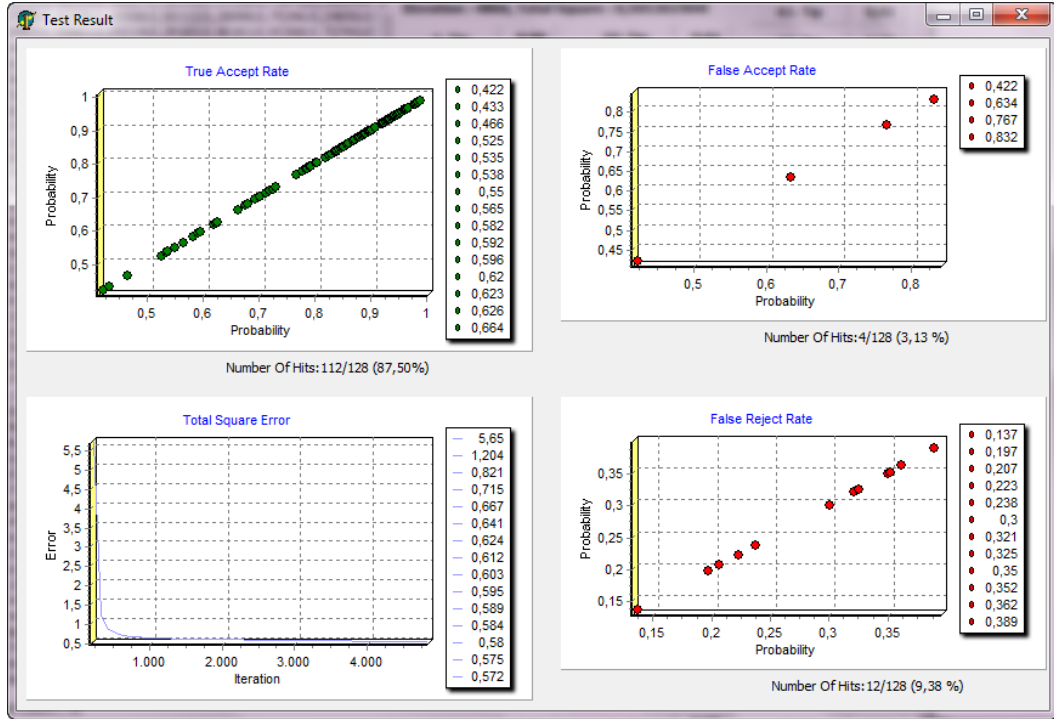
*Her bir penceredeki küme sayısı



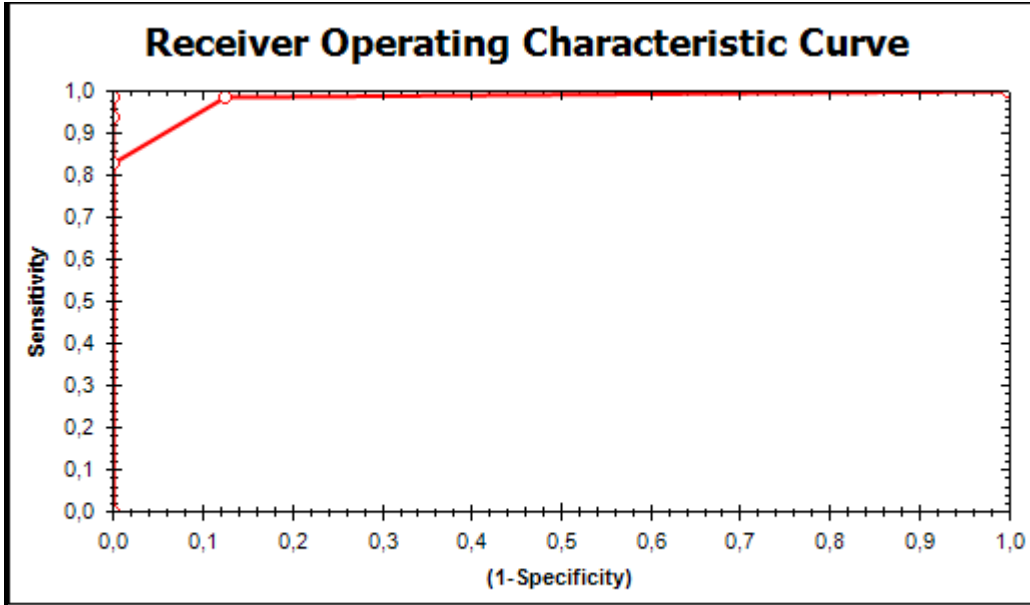
Şekil 7.12. DWT-BKYSA – Test sonuçları ve hatanın kareleri toplamı (2 iris eğitim 1 iris test için, 3x3 pencere)



Şekil 7.13. DWT-BKYSA – Test sonuçları ve hatanın kareleri toplamı (2 iris eğitim 1 iris test için, 4x4 pencere)

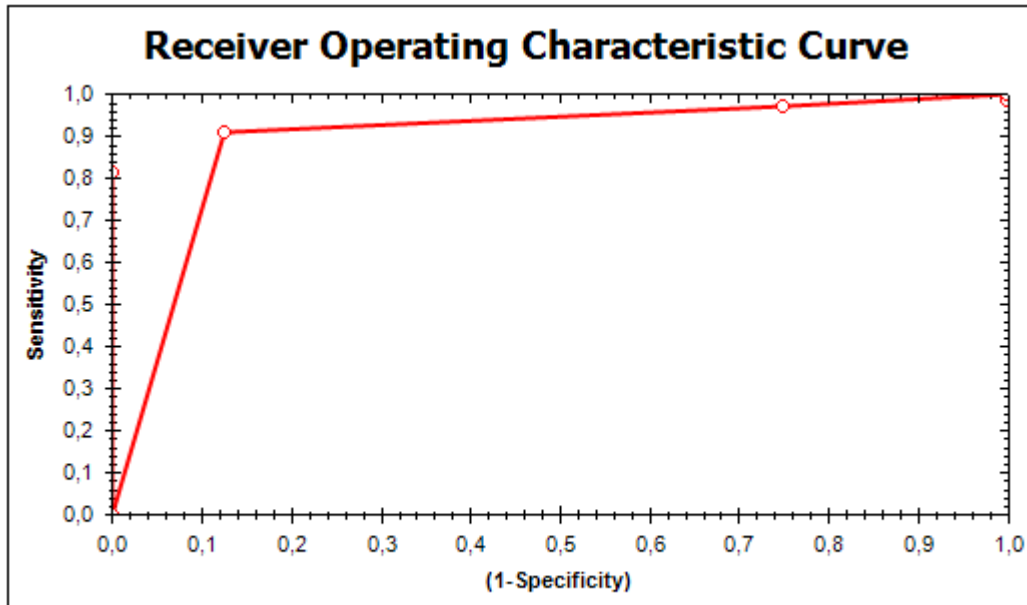


Şekil 7.14. DWT-BKYSA Hatanın kareleri toplamı ve test sonuçları (1 iris eğitim 2 iris test için, 4x4 pencere)



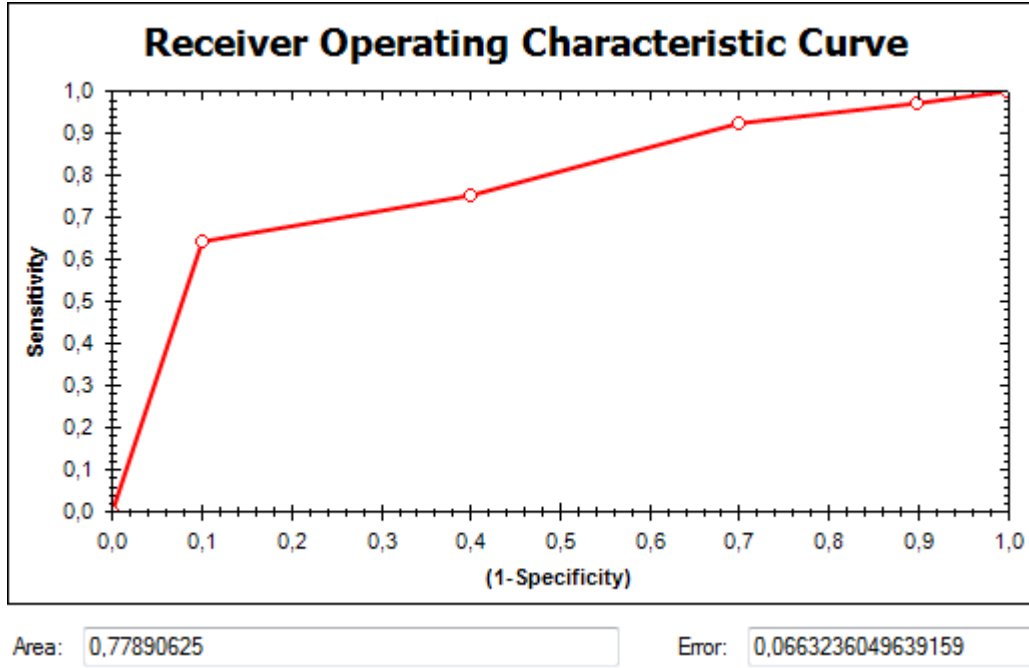
Area: Error:

Şekil 7.15. Roc Grafiği (2 iris eğitim 1 iris test için, 3x3 pencerelemeli)



Area: Error:

Şekil 7.16. Roc Grafiği (2 iris eğitim 1 iris test için, 4x4 pencerelemeli)



Şekil 7.17. Roc Grafiği (2 iris eğitim 1 iris test için, 5x5 pencerelemeli)

7.6 Genel Test Sonuçları

Tablo 7.6'da daha önceden yapılan çalışmalarda elde edilen sonuçlar belirtilmiştir. En alttaki 3 değer ise tez çalışmasında aldığımız DWT-YSA, DWT-BKYSA (3x3 pencerelemeli) ve DWT-BKYSA (4x4 pencerelemeli) test sonuçlarıdır. DWT-BKYSA ve DWT-YSA yöntemlerinde elde edilen sonuçlar daha önceden bu alanda yapılan çalışmalar ile karşılaştırıldığında en az onlar kadar başarılı olduğu görülmektedir. Test sonuçları klasik YSA ile karşılaştırıldığında; test sonuçları iyileşmiş, süre kısalmış ve tanıma oranları artmıştır. Tezin amaçlarından biri olan BKYSA'nın 2 boyutlu biyometriklere uygulanması DWT-BKYSA örneğinde başarıyla gerçekleştirilmiştir. Aşağıdaki tablo ve şekillerde yukarıda belirtilen test sonuçları birlikte aynı tablolarda özet şeklinde verilmiştir (Tablo 7.7, Tablo 7.8).

Tablo 7.6. Daha önce arařtırmacılar tarafından geliřtirilmiř iris tanıma sistemleri ve tanıma performans sonuçları (YTO – Yanlıř tanıma oranı / YRO – Yanlıř reddetme oranı / DTO - Dođru tanıma oranı)
(Koçer, H. E., 2007)

Arařtırmacı	Yıl	Tanıma Yöntemi	Veritabanı	Performans
Daugman	1993 2001 2003	Hamming Uzak. Hamming Uzak. Hamming Uzak.	592 iris gör. 2150 iris gör. 4258 iris gör.	<0.0001 (YTO) - <0.0001 (YRO) <0.0001(YTO) - <0.0001(YRO) <0.0001(YTO) - <0.0001(YRO)
Boles ve ark.	1998	Benzersizlik fonk.	4 iris gör.	100% (DTO)
Reillo ve ark.	2000 2002	Hamming Uzak. Hamming Uzak. Öklid Uzaklıđı	10 kiři 10 kiři	%98.3 (DTO) %98.7 (DTO) % 96.7 (DTO)
Tisse ve ark.	2002	Hamming Uzak.	300 iris gör.	% 1 (YTO) – % 3 (YRO)
Lim ve ark.	2001	YSA (LVQ)	200 kiři	% 94.4 (DTO)
Li Ma ve ark.	2002	NFL Öklid Uzaklıđı	109 kiři 25 kiři	% 0.1 (YTO) – % 0.83 (YRO) % 99.09 (DTO)
Huang ve ark.	2002	Öklid Uzaklıđı	10 kiři	% 100 (DTO)
Liam ve ark.	2002	YSA (SOM)	30 kiři	% 83 (DTO)
Alim ve ark.	2002	YSA (MLP)	159 kiři	% 96.1 (DTO)
Yu ve ark.	2002	Fourier Dönüřümü	10 kiři	% 0.26 (YTO) – % 1.79 (YRO)
Chen ve ark.	2003	K-Means	40 kiři	% 3.67 (YTO) – % 8.82 (YRO)
Noh ve ark.	2003	Öklid Uzaklıđı	16 kiři	% 93.8 (DTO)
Masek	2003	Hamming Uzak.	756 iris gör. 120 iris gör.	% 0.005 (YTO) – % 0.238 (YRO) % 0 (YTO) – % 0 (YRO)
Sun ve ark.	2005	Hamming Uzak.	306 kiři	% 0.01 (YTO) – % 0.01 (YRO)
Miyazawa ve ark.	2006	Faz tabanlı karř. (Phase Based)	108 kiři	% 0.0032 (YTO / YRO)
DWT-BKYSA – 3x3 Pencerelemeli*	2011	DWT-BKYSA	64 kiři	%98,44 (DTO) %0 YTO, % 1,56 YRO, 0,98 AUC
DWT-BKYSA – 4x4 Pencerelemeli*	2011	DWT-BKYSA	64 kiři	%96,88 (DTO) %0 YTO, % 3,13 YRO, 0,89 AUC
DWT-YSA*	2011	DWT-YSA	64 kiři	%96,88 (DTO) %0 YTO, % 3,13 YRO, 0,99 AUC
*Tez çalıřmasının sonuçları				

Tablo 7.7. Genel test sonuçları (Klasik YSA)

	2 iris eğitim 1 iris testte kullanıldı													
	Piksel Sayısı	Eğitim						Test				YSA		
		YTO (%)	YRO (%)	Süre (dk:sn)	DTO (%)	İterasyon	Hata	YTO (%)	YRO (%)	AUC	DTO (%)	Momentum	Eğitim Oranı	Nöron Sayısı
DCT	864 (32x27)	0,00	0,00	05:34	100,00	517	1,18	0,00	4,69	0,97	95,31	0,75	0,35	864-433-64
DWT	255 (17x15)	0,00	0,00	00:20	100,00	845	0,5	0,00	3,13	0,99	96,88	0,75	0,75	129

	1 iris eğitim 2 iris testte kullanıldı													
	Piksel Sayısı	Eğitim						Test				YSA		
		YTO (%)	YRO (%)	Süre (dk:sn)	DTO (%)	İterasyon	Hata	YTO (%)	YRO (%)	AUC	DTO (%)	Momentum	Eğitim Oranı	Nöron Sayısı
DCT	864 (32x27)	0,00	0,00	04:40	100,00	750	0,49	0,00	10,94	0,88	89,06	0,75	0,75	864-433-64
DWT	255 (17x15)	0,00	0,00	01:04	100,00	2739	0,09	0,00	9,38	0,91	90,63	0,75	0,50	129

Tablo 7.8. Genel test sonuçları (BKYSA)

	2 iris eğitim 1 iris testte kullanıldı																
	Piksel Sayısı	Eğitim						Test				YSA			BCO		
		YTO (%)	YRO (%)	Süre (dk:sn)	DTO (%)	İterasyon	Hata	YTO (%)	YRO (%)	AUC	DTO (%)	Momentum	Eğitim Oranı	Nöron Sayısı	Pencere Boyutu	Pencere Küme Sayısı*	Toplam Küme Sayısı
DCT-BCO	864 (32x27)	0,00	0,00	01:06	100,00	2202	1,85	23,44	32,81	0,67	43,75	0,3	0,75	198-100-64	3x3	2	198
DCT-BCO	864 (32x27)	0,00	0,00	06:32	100,00	29782	2,23	34,38	34,38	0,26	31,25	0,75	0,75	112-57-64	4x4	2	112
DCT-BCO	864 (32x27)	0,00	0,00	08:10	100,00	53322	3,26	40,63	34,38	0,13	25	0,65	0,85	84-43-64	5x5	2	84
DWT-BCO	255 (17x15)	0,00	0,00	00:04	100,00	722	1,2	0	1,56	0,98	98,44	0,15	0,75	60-33-64	3x3	2	60
DWT-BCO	255 (17x15)	0,00	0,00	00:08	100,00	1363	1,2	0	3,13	0,89	96,88	0,01	0,90	40-33-64	4x4	2	40
DWT-BCO	255 (17x15)	0,00	0,00	00:06	100,00	1215	1,32	6,25	7,81	0,77	85,94	0,55	0,85	24-33-64	5x5	2	24

	1 iris eğitim 2 iris testte kullanıldı																
	Piksel Sayısı	Eğitim						Test				YSA			BCO		
		YTO (%)	YRO (%)	Süre (dk:sn)	DTO (%)	İterasyon	Hata	YTO (%)	YRO (%)	AUC	DTO (%)	Momentum	Eğitim Oranı	Nöron Sayısı	Pencere Boyutu	Pencere Küme Sayısı*	Toplam Küme Sayısı
DCT-BCO	864 (32x27)	0,00	0,00	00:38	100,00	2547	0,65	28,13	51,56	0,61	20,31	0,15	0,75	198-100-64	3x3	2	198
DCT-BCO	864 (32x27)	0,00	0,00	01:48	100,00	16300	0,58	44,53	39,84	0,44	15,63	0,75	0,75	112-57-64	4x4	2	112
DCT-BCO	864 (32x27)	0,00	0,00	03:03	100,00	45166	0,58	49,22	38,28	0,44	12,50	0,5	0,75	84-43-64	5x5	2	84
DWT-BCO	255 (17x15)	0,00	0,00	00:15	100,00	4481	0,4	2,34	3,13	0,91	94,53	0,15	0,75	60-33-64	3x3	2	60
DWT-BCO	255 (17x15)	0,00	0,00	00:05	100,00	1853	0,54	3,91	7,03	0,84	89,06	0,85	0,85	40-33-64	4x4	2	40
DWT-BCO	255 (17x15)	0,00	0,00	00:07	100,00	2845	0,6	11,72	13,3	0,82	75	0,15	0,95	24-33-64	5x5	2	24

8 SONUÇ

BKYSA iki boyutlu bir örüntü tanıma problemi üzerine uygulanışı gerçekleştirilmiştir. Yapılan çalışma birçok biyometrik üzerine uygulanabilir şekilde geliştirilmiştir (iris, yüz, parmak izi vs.). Bu çalışma için biyometriklerden biri olan iris seçilmiş ve iris tanıma üzerine çalışma yapılmıştır. Bu yöntem ile biyometrik sistemin eğitim süresinde ciddi oranlarda kısalma olmuş ve tanıma kalitesinde de artış olmuştur.

Ön işlemler aşamasında irisin göz kapağı, kirpik gibi gürültülerden en az etkileneceği göz bebeğinin sağ ve sol yanındaki parçalar kullanılmıştır. İris parçası resimden çıkarıldıktan sonra diğer ön işlem adımları uygulanmıştır. Ön işlem adımlarından olan yeniden boyutlandırma işleminin en büyük nedeni YSA'nın eğitim sürelerini makul seviyelerde tutmaktır. Test için kullandığımız bilgisayarda YSA'nın giriş sayısı 1000'in üstüne çıktığında eğitim süresinde ciddi oranda artışlar olmuştur. Bu nedenle daha küçük boyutlarda veri ile çalışma yapılmıştır.

Öz nitelik vektörü elde edilirken DWT ve DCT yöntemlerinden yararlanılmıştır. Öz nitelik vektörü elde edildikten sonra sınıflandırma aşamasına geçilmiştir.

Sınıflandırma aşamasında iki yöntem kullanılmıştır. Bunlardan birincisi klasik YSA diğeri ise BKYSA yöntemidir. BKYSA yönteminde kümeleme işlemi yapılırken BK algoritmalarından bir tanesi olan BCO ile öz nitelik vektörü üzerinde kümeleme yapılmış ve YSA ile de sınıflandırma gerçekleştirilmiştir. İki yöntemle de yapılan sınıflandırmaların test sonuçları karşılaştırılmıştır. Testler için YTO, YRO, DTO ve AUC değeri kullanılmıştır.

BKYSA ile çalışıldığında tüm veri yerine pencerelerden elde edilen küme merkezleri kullanıldığı için veri boyutunda çok büyük oranlarda azalma olmuştur. Büyük boyutlardaki öz nitelik vektörü küçük boyutlardaki öz nitelik vektörleri ile temsil edilebilir hale gelmiştir. Bunun sayesinde gereksiz ve tanımada hiçbir faydası olmayacak, hatta tanımayı olumsuz etkileyebilecek olan özellikler (features) öz nitelik vektöründen arındırılmıştır. Yapılan bu işlem BKYSA'nın klasik YSA'na göre en önemli üstünlüklerinden biridir.

BKYSA yöntemi DCT ile kullanıldığında tanıma oranlarında düşüşler gözlemlenmiştir. DWT ile kullanıldığında olumlu sonuçlar alınmıştır. DCT de sonuçların kötü olmasının nedeni, DCT dönüşümünden sonra enerjinin belli bir bölgede toplanmasıdır. Bir diğer nedeni ise DCT katsayılarının bir birine çok yakın olmalarıdır. Bu katsayılar ile kümele yapıldığında sonuçlar bir birine çok yakın çıkmaktadır ve buda ağın testlerdeki doğruluk oranını çok fazla düşürmektedir. DCT işleminden sonra elde edilen görüntüde özelliklerin belli bölgelerde yoğunlaşması (Şekil 3.2) kümelemeyi olumsuz etkilediği tespit edilmiştir.

Öznitelik vektörüne BCO algoritması uygulanırken 5x5, 4x4, 3x3 gibi farklı boyutlarda pencereleme yapılmıştır. DWT – BKYSA yönteminde 5x5 boyutunda pencereleme yapıldığında klasik YSA ya nazaran test ölçütlerine göre iyi sonuçlar elde edilmemiştir. 4x4 boyutunda pencereleme yapıldığında klasik YSA ya benzer test sonuçları alınmıştır. 3x3 boyutunda pencereleme yapıldığında ise klasik YSA ya göre daha iyi test sonuçları alınmış, tanıma oranlarında da artma olmuş ve eğitim süresi de önemli derecede azalmıştır. DWT-BKYSA'da giriş sayıları azaldığı için YSA'nın giriş ve gizli katmandaki nöron sayıları ciddi derecede azalmıştır. Bu azalmada donanımsal bir avantaj sağlamıştır.

BKYSA ve klasik YSA ile yapılan çalışma sonuçlarında yüksek tanıma oranları elde edilmiştir. Elde edilen sonuçlar incelendiğinde ve daha önce yapılan çalışmalar ile de karşılaştırıldığında tanıma oranlarındaki başarı görülmektedir. Tez çalışmasının ana amaçlarından biri olan BKYSA yönteminin 2 boyutlu biyometriklere uygulanabilirliği; yöntemin DWT ile yapılan testlerde (3x3 ve 4x4 pencerelemelerde) başarılı bir şekilde ortaya konulmuştur. BKYSA yönteminin DCT ile kullanıldığında ise yukarıda belirttiğimiz nedenlerden dolayı aynı başarı elde edilememiştir.

Yapılan tez çalışmasının bir sonraki adımları olacağını düşündüğümüz geleceğe yönelik çalışmalar aşağıda belirtilmiştir:

- Farklı yöntemlerle alınmış ve fazla sayıda iris görüntüsü içeren veritabanlarının kullanılması.
- BKYSA yönteminin farklı biyometriklere uygulanması.
- DCT-BKYSA yönteminde yukarıda belirtilen nedenler doğrultusunda test sonuçlarını iyileştirmeye yönelik çalışmalar yapılması.

- Dięer 6znetelik ıkarma y6ntemlerinin BKYSA'ya uygulanabilirlięinin incelenmesi. Dięer geliřmiř k6meleme algoritmalarının kullanılması.
- YSA iin uygun momentum, 6ęrenme oranı deęerlerini otomatik olarak bulabilen bir algoritmanın geliřtirilmesi.

9 KAYNAKLAR

- Abd-El-Reheim M.A-E-A. 1999. Power Network Topology Recognition Using Neural Networks, *Electric Machines and Power Systems*, Volume 27, Number 2, 1 February 1999, pp. 195-208(14)
- Authenti-Corp. 2007. Standards-Based Performance and User Cooperation Studies of Commercial Iris recognition Products, *Iris Recognition Study 2006 Final Report*, March 31st, 2007
- Bezdek, J.C., Ehrlich R., Full, W., 1984. FCM, *Fuzzy C-Means Algorithm Computers and Geoscience* 10 (2-3), 191-203.
- Bezdek, J.C., Pal, S.K. 1992. Fuzzy Models For Pattern Recognition: Methods that Search for Structures in Data, *IEEE Pres*, New York
- Biyometrik Diagram, 2011,
http://commons.wikimedia.org/wiki/File:Biometric_system_diagram.svg
- Bolle, R. M., Pankanti, S., Connell, J. H. and Ratha, N. Iris individuality: A partial iris model. In *International Conference on Pattern Recognition*, pages II:927-930, 2004
- Cui, J., Wang, Y., Tan, T., Ma, L., Sun, Z., A Fast and Robust Iris Localization Method Based on Texture Segmentation, *Proceedings of SPIE*, 2004
- Daubechies, I., “The wavelet transform, time-frequency localization and signal analysis”, *IEEE Transactions on Information Theory*, 36(5), 961-1005, 1990.
- Daugman, J., High confidence visual recognition of persons by a test of statistical independence. *Trans. Pattern Analysis and Machine Intelligence*, (1993), 15(11): 1148-1161
- Daugman, J. 2004. How iris recognition works, *IEEE Trans*, CSVT 14, 21—30, 2004
- Daugman, J. 2001. Statistical richness of visual phase information: Update on recognizing persons by their iris patterns. *International Journal of Computer Vision*, (2001), 45(1): 25-38
- Daugman, J. 2005. Technical Report of Results from 200 Billion Iris Cross Comparisons. *University of Cambridge*, no. UCAM-CL-TR-635, June 2005

Dobeš, M., Martinek, J., Skoupil, D., Dobešová, Z., Pospíšil, J., Human eye localization using the modified Hough transform. *Optik*, Volume 117, No.10, p.468-473, Elsevier 2006, ISSN 0030-4026.

Du, Y., Ives, R., Bonney, B., Etter, D., M. Analysis of partial iris recognition. *Biometric Technology for Human Identification II*. Edited by Jain, Anil K.; Ratha, Nalini K. *Proceedings of the SPIE*, Volume 5779, pp. 31-40 (2005).

Ergosis Güvenlik Sistemleri, 2010, <http://www.ergosis.com.tr/>

Geraldo, X. *Fuzzy Logic*, Computing Science Department and Systems and Computing.

Grossberg, S. 1986. *The Adaptive Brain I: Cognition, Learning, Reinforcement, and Rhythm, and The Adaptive Brain II: Visions, Speech, Language, and Motor Control*, Elsevier/North-Holland, Amsterdam.

Gunlu, G., Bilge, H. Ş. Improving the Performance of the Image Compression by Global Discrete Cosine Transform.

Gupta, P., Mehrotra, H., Rattani, A., Chatterjee, A. and Kaushik, A.K., Iris Recognition using Corner Detection, 23rd International Biometric Conference, Montreal, Canada, 2006

Güler, İ., Übeyli, E., Dalgacık Dönüşümünün Kullanımı İle Teşhis Sistemleri İçin Öznitelik Çıkarma: İç Karotid Atardamar Doppler İşaretlerinin Durum Analizi, ASYU-INISTA 2004

Güler, N., Bulanık Kümeleme Analizi Ve Bulanık Modelleme Uygulamaları, Muğla, 2006

Hong, L. and Jain A.K. 1998. Integrating Faces and Fingerprints for Personal Identification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, no. 12, pp. 1295-1307.

Hopfield, J.J., 1982. Neural networks and physical systems with emergent collective computational abilities. *Proc.Nat.Acad.Sci* 79:2554-2558.

Hopfield, J.J., 1987. Learning algorithms and probability distributions in feed-forward and feed-back networks. *Proc.Nat.Acad.Sci* 88:8429-8433.

Iris, 2011, <http://www.banucosar.net/img/modul/1mo3su55mfeg1y2bl4efel55423521.jpg>

Iris Database, Dobeš, M. and Martinek, J., <http://www.inf.upol.cz/iris/>, 2010

- Jain, L.C., Halici U., Hayashi I., and Lee S.B. 1999a. Intelligent Biometric Techniques in Fingerprint and Face Recognition, CRC Press, Boca Raton, FL.
- Jang, J.-S. R., Sun, C.-T. and Mizutani, E., *Neuro-Fuzzy and Soft Computing: A Computational Approach to Learning and Machine Intelligence*, Prentice-Hall, Upper Saddle River, NJ, USA, 1997
- Karlik, B., Artificial Neural Networks, 2009 b, Lecture Notes
- Karlik, B., Fuzzy Logic, 2009 c, Lecture Notes
- Karlik, B., H. Tropi, and M. Alci, "A fuzzy-neural approach for the characterization of the active microwave devices," in *Proceedings of the 12th International Conference on Microwave and Telecommunication Technology (CriMiCo '02)*, pp. 114–117, Sevastopol, Ukraine, September 2002
- Karlik, B., M. O. Tokhi, and M. Alci, "A fuzzy clustering neural network architecture for multifunction upper-limb prosthesis," *IEEE Transactions on Biomedical Engineering*, vol. 50, no. 11, pp. 1255–1261, 2003
- Karlik, B., "Neural Models for the Active Microwave Devices", LFNM'2003, Alushta, Ukraine, Sep. 2003, pp. 281-286
- Karlik, B. 2009. Pattern Recognition and Computer Vision, 2009 a, Lecture Notes
- Karlik, B. 2002. The effects of fuzzy clustering on the backpropagation algorithm, in *International Conference on Computational and Applied Mathematics*, pp. 9–10, Kiev, Ukraine, September 2002
- Karlik, Bekir, Tokhi Osman, Alci Musa. 2003. A Fuzzy Clustering Neural Network Architecture for Multi-Function Upper-Limb Prosthesis, *IEEE Trans. on Biomedical Engineering*, Vol. 50, No. 11, Jan. 2003, pp. 1255-1261
- Karlik, B., Yuksek K. 2007. Fuzzy Clustering Neural Networks for Real Time Odor Recognition System, *Journal of Automated Methods and Management in Chemistry* , Dec. 2007
- Karlik, B. 1994. *Myoelectric Control of Multifunction Prosthesis Using Neural Networks*, Yildiz Technical University, PhD, Thesis, Istanbul, Turkey
- Karlik, B. 2011. YSA – MLP-BP algoritması Prof. Dr. Bekir Karlık'a aittir ve izni dahilinde kullanılmaktadır.

- Kaufman, L., ; Rousseuw, P., 1990, Finding Groups Data : *An Introduction to Cluster Analysis*, John Wiley and Sons, New York.
- Koçer, H. E. (2007). *İris Deseninin Yapay Zeka Yöntemleri İle Tanınması*, Yayınlanmamış Doktora Tezi. Konya: SÜ. Fen Bilimleri Enstitüsü.
- Kohonen, T. 1984. Self-Organization and Associative Memory, Springer-Verlag, Berlin.
- L., Tichavský, P., Pospíšil, J., Human Eye Iris Recognition Using the Mutual Information. Optik Volume 115, No.9, p.399-405, Elsevier 2004, ISSN 0030-4026.
- Masek, L., «Recognition of Human Iris Patterns for Biometric Identification», *School of Computer and Software Engineering, The University of Western Australia*, 2003
- Mc Culloch, W.S., Pitts, W.H. 1943. A logical calculus of the ideas immanent in neural nets. Bull. Math.Biophys. 5:115-133.
- Merényi, E., Taranik, J.V., Minor, T.B., and Farrand, W.H. (1996) Quantitative Comparison of Neural Network and Conventional Classifiers For Hyperspectral Imagery, In Summaries of the Sixth Annual JPL Airborne Earth Science Workshop, Pasadena, CA, March 4-8, 1996, Vol.1: AVIRIS Workshop. (Ed. R.O.Green).
- Naes, T.; Mevik, B., H., 1999, The Flexibility of Fuzzy Clustering Illustrated by Examples , Journal of Chemo metrics, 13: 435-444.
- Newton, E.M. and Phillips, P.J. August 2007. Meta-Analysis of Third-Party Evaluations of Iris Recognition. *Technical Report NISTIR 7440*, National Institute of Standards and Technology, Gaithersburg, MD.
- Özbay, Y., Ceylan, R., Karlik, B. 2006. A fuzzy clustering neural network architecture for classification of ECG arrhythmias, *Computers in Biology and Medicine*, Vol.36, No.4, pp. 376-388, 2006
- Pao, Y.H. 1989. Adaptive Pattern Recognition and Neural Networks, Addison-Wesley Publishing Co., Reading, Mass.
- Phillips, P. 2007. FRVT 2006 and ICE 2006 Large-Scale Results. *Technical Report NISTIR 7408*, National Institute of Standards and Technology, Gaithersburg, MD, March 2007
- Proença, H., Alexandre, L. 2007. Towards Non-Cooperative Iris Recognition: A Classification Approach Using Multiple Signatures. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Vol. 3, No. 1, pp. 1 - 12, July, 2007

Proença, H., *Towards Non-Cooperative Biometric Iris Recognition*, Universidade da Beira Interior, Phd Thesis, January, 2007, ISBN: 978-972-99648-2-4

ROC, 2011, <http://tr.wikipedia.org/wiki/ROC>

Rosenblatt, F. 1962. *Principles of Neuradynamics:Perceptrons and The Theory of Brain Mechanisms*, Spartan Books, Washington D.C.

Rosenblatt, F., 1959. *Principles of Neuradynamics*, Spartan Books, New York.

Rosenblatt, F. ,1962. *Principles of Neuradynamics:Perceptrons and The Theory of Brain Mechanisms*, Spartan Books, Washington D.C.

Rumelhart, D.E., Hinton, G.E., Williams, R.J. 1986. "*Learning Internal Representations by Error Propagation*", MIT Press, Cambridge, MA.

Rumelhart, D.E., McClelland, J.L., PDP Research Group 1986. *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*, v. 1:Foundations, MIT Press, Cambridge, MA.

Swets, 1996, *Signal detection theory and ROC analysis in psychology and diagnostics : collected papers*.

Şen, Z., 2001, *Bulanık Mantık ve Modelleme İlkeleri*, Bilge Sanat Yapım Evi, Ekim 2001, İstanbul.

Şahinli, F., *Kümeleme Analizine Fuzzy Set Teorisi Yaklaşımı*, Yüksek Lisans Tezi(İstatistik), Gazi Üniversitesi Fen bilimleri Enstitüsü , Ankara,1999.

Uludag, U., Gunsel, B., and Ballan M. 2001. A spatial method for watermarking of fingerprint images. Proc. First Intl. Workshop on Pattern Recognition in Information Systems, Setúbal, Portugal, pp. 26-33.

Vizyotek, 2011, <http://www.vizyotek.com/>

Yapay Zeka, 2011, <http://www.yapay-zeka.org/modules/icontent/index.php?page=33>

Widrow, B., Hoff, M. 1960. Adaptive switching circuits. IRE WESCON Convention Record 4:96-104.

Wildes, R. P., "Iris Recognition: An Emerging Biometric Technology", Proceedings of the IEEE, Vol. 85, No. 9, September 1997

Xie, L., *Algorithms for Recognition of Low Quality Iris Images*, M.A.Sc. (2007)

Zhang, H., 2005, *A Note Fuzzy Clustering*, Department of Computer Science and Engineering University Connecticut.

Zhang, Y., Wang, Y., Wang, W., Liu, B., “Doppler ultrasound signal denoising based on wavelet frames”, IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control, 48(3), 709-716, 2001.

10 EKLER

EK-1: Kaynak Kodu

DCT işlemini yapan Matlab kaynak kodu:

```
I = imread('C:\IRIS\001L_1.png');
I = imresize(I,0.20);
I = rgb2gray(I);
I = histeq(I);
I = im2double(I);
[cA1] = dct2(I);
[cA1] = mat2vec(cA1);
cA1 = single(cA1);
cA1 = cA1 / max(cA1);
```

DWT işlemini yapan Matlab kaynak kodu:

```
I = imread('C:\IRIS\001L_1.png');
I = imresize(I,0.21);
I = rgb2gray(I);
I = histeq(I);
I = im2double(I);
[cA1] = dwt2(I,'db1');
[cA1] = mat2vec(cA1);
cA1 = single(cA1);
cA1 = cA1 / max(cA1);
```

mat2vec fonksiyonu:

```
function vect= mat2vec(A)
[M,N]=size(A);
L=M*N;
for i=1:M
    vect(i*N-N+1:i*N)=A(i,:);
end
```

Bulanik C-Ortalamalar Algoritması Kaynak Kodu:

```
CmeansAlgorithm.cs
using System.Collections.Generic;
namespace System.Data.Clustering
{
    public sealed class CMeansAlgorithm
    {
        private double Accuracy = Math.Pow(10, -5);
        /// Array containing all points used by the algorithm
        public List<ClusterPoint> Points;
        /// Array containing all clusters handled by the algorithm
        public List<ClusterPoint> Clusters;
        /// Gets or sets membership matrix
        public double[,] U;
        /// Gets or sets the current fuzzyness factor
        private double Fuzzyness;
        /// Algorithm precision
        private double Eps = Math.Pow(10, -5);
        /// Gets or sets objective function
        public double J { get; set; }
        /// Gets or sets log message
        public string Log { get; set; }
        public CMeansAlgorithm(List<ClusterPoint> points, List<ClusterPoint> clusters)
            : this(points, clusters, 2.0)
        {}
        public CMeansAlgorithm(List<ClusterPoint> points, List<ClusterPoint> clusters,
double fuzzy)
        {
            if (points == null)
            {
                throw new ArgumentNullException("points");
            }
            if (clusters == null)
            {
                throw new ArgumentNullException("clusters");
            }
            this.Points = points;
        }
    }
}
```

```

this.Clusters = clusters;
U = new double[this.Points.Count, this.Clusters.Count];
this.Fuzzyness = fuzzy;
double diff;
// Iterate through all points to create initial U matrix
for (int i = 0; i < this.Points.Count; i++)
{
    ClusterPoint p = this.Points[i];
    double sum = 0.0;
    for (int j = 0; j < this.Clusters.Count; j++)
    {
        ClusterPoint c = this.Clusters[j];
        diff = this.CalculateEulerDistance(p, c);
        U[i, j] = (diff == 0) ? Eps : diff;
        sum += U[i, j];
    }
    double sum2 = 0.0;
    for (int j = 0; j < this.Clusters.Count; j++)
    {
        U[i, j] = 1.0 / Math.Pow(U[i, j] / sum, 2.0 / (Fuzzyness - 1.0));
        sum2 += U[i, j];
    }
    for (int j = 0; j < this.Clusters.Count; j++)
    {
        U[i, j] = U[i, j] / sum2;
    }
}
this.RecalculateClusterIndexes();
}
private CMeansAlgorithm()
{
}
private void RecalculateClusterIndexes()
{
    for (int i = 0; i < this.Points.Count; i++)
    {
        double max = -1.0;
        var p = this.Points[i];

```



```

    for (int j = 0; j < this.Clusters.Count; j++)
    {
        if (max < U[i, j])
        {
            max = U[i, j];
            p.ClusterIndex = (max == 0.5) ? 0.5 : j;
        }
    }
}
}
}
public void Step()
{
    for (int c = 0; c < Clusters.Count; c++)
    {
        for (int h = 0; h < Points.Count; h++)
        {
            double top = CalculateEulerDistance(Points[h], Clusters[c]);
            //if (top <= 1.0) top = Eps;
            if (top == 0) top = Eps;
            // sumTerms is the sum of distances from this data point to all clusters.
            double sumTerms = 0.0;
            for (int ck = 0; ck < Clusters.Count; ck++)
            {
                sumTerms += top / CalculateEulerDistance(Points[h], Clusters[ck]);
            }
            // Then the membership value can be calculated as...
            U[h, c] = (double)(1.0 / Math.Pow(sumTerms, (2 / (this.Fuzzyness - 1))));
        }
    }
    this.RecalculateClusterIndexes();
}
private double CalculateEulerDistance(ClusterPoint point, ClusterPoint centroid)
{
    double sum = 0.0;
    for (int i = 0; i < point.Dimention; i++)
    {
        sum += Math.Pow(point.Coords[i] - centroid.Coords[i], 2);
    }
}

```

```

    return Math.Sqrt(sum);
}
public double CalculateObjectiveFunction()
{
    double Jk = 0;
    for (int i = 0; i < this.Points.Count; i++)
    {
        for (int j = 0; j < this.Clusters.Count; j++)
        {
            Jk += Math.Pow(U[i, j], this.Fuzzyness) *
Math.Pow(this.CalculateEulerDistance(Points[i], Clusters[j]), 2);
        }
    }
    return Jk;
}
public void CalculateClusterCenters()
{
    ClusterPoint[] tempClusters = new ClusterPoint[this.Clusters.Count];
    this.Clusters.CopyTo(tempClusters, 0);
    for (int j = 0; j < this.Clusters.Count; j++)
    {
        ClusterPoint c = this.Clusters[j];
        //double uX = 0.0;
        //double uY = 0.0;
        double[] uC = new double[c.Dimention];
        double l = 0.0;
        for (int i = 0; i < this.Points.Count; i++)
        {
            ClusterPoint p = this.Points[i];
            double uu = Math.Pow(U[i, j], this.Fuzzyness);
            for (int k = 0; k < c.Dimention; k++)
            {
                uC[k] += uu * p.Coords[k];
            }
            l += uu;
        }
        for (int k = 0; k < c.Dimention; k++)
        {
            c.Coords[k] = (uC[k] / l);
        }
    }
}

```

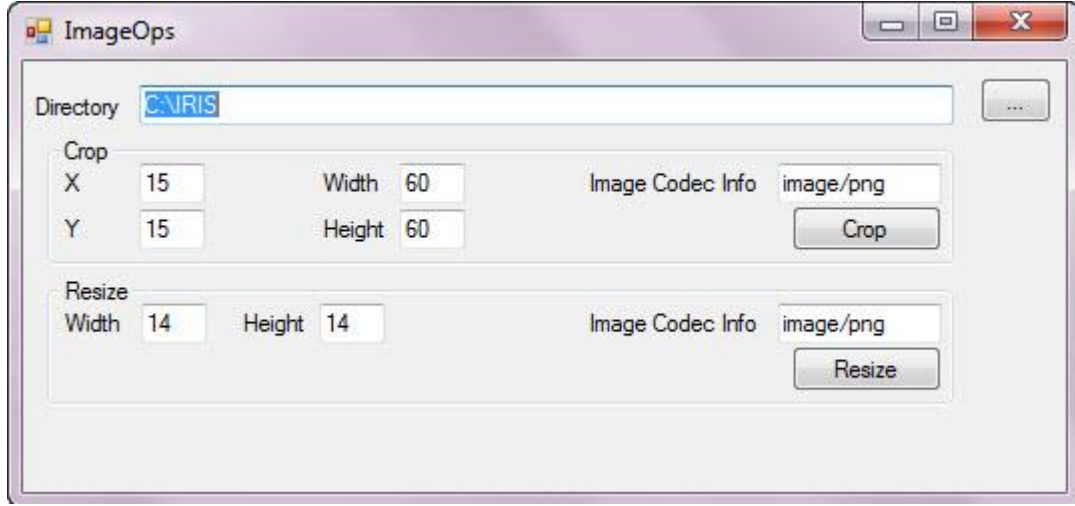
```

        }
        this.Log += string.Format("Cluster Centroid: ({0})" +
System.Environment.NewLine, c.Coords[0]);
    }
}
public int Run()
{
    return this.Run(this.Accuracy, 20);
}
public int Run(double accuracy, int maxIterations)
{
    int i = 0;
    do
    {
        i++;
        this.J = this.CalculateObjectiveFunction();
        this.CalculateClusterCenters();
        this.Step();
        double Jnew = this.CalculateObjectiveFunction();
        if (Math.Abs(this.J - Jnew) < accuracy) break;
    }
    while (maxIterations > i);
    return i;
}
}
}

```

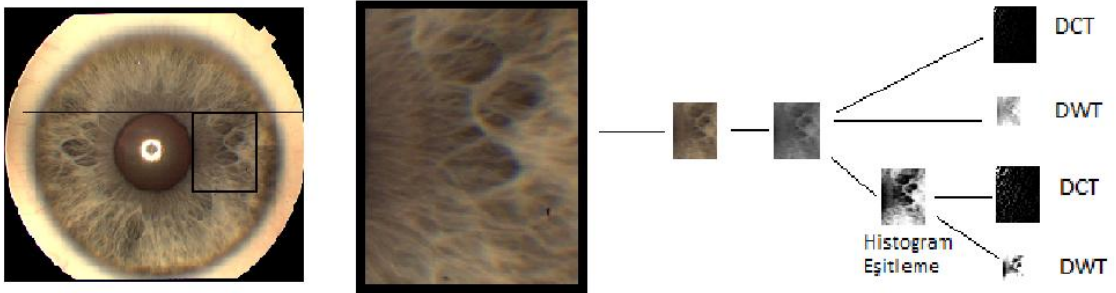
EK-2: Kullanıcı Arayüzü

Resim İşleme Uygulaması:



Şekil 10.1. Resim İşleme

Öz Nitelik Vektörünün Çıkartılması:

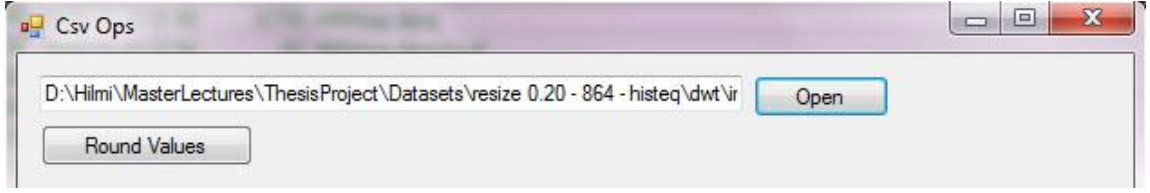


Şekil 10.2. Ön işlemler ve Öz nitelik vektörünün çıkarımı

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
1	0.5431	0.3373	0.3529	0.2461	0.2422	0.5941	0.5598	0.3255	0.1461	0.2824	0.6471	0.8088	0.9137	0.9255	0.8647	0.6314	0.5284	0.4912	0.4118	0.2784	0.1980	0.1775	0.1333	0.17
2																								
3																								
4																								

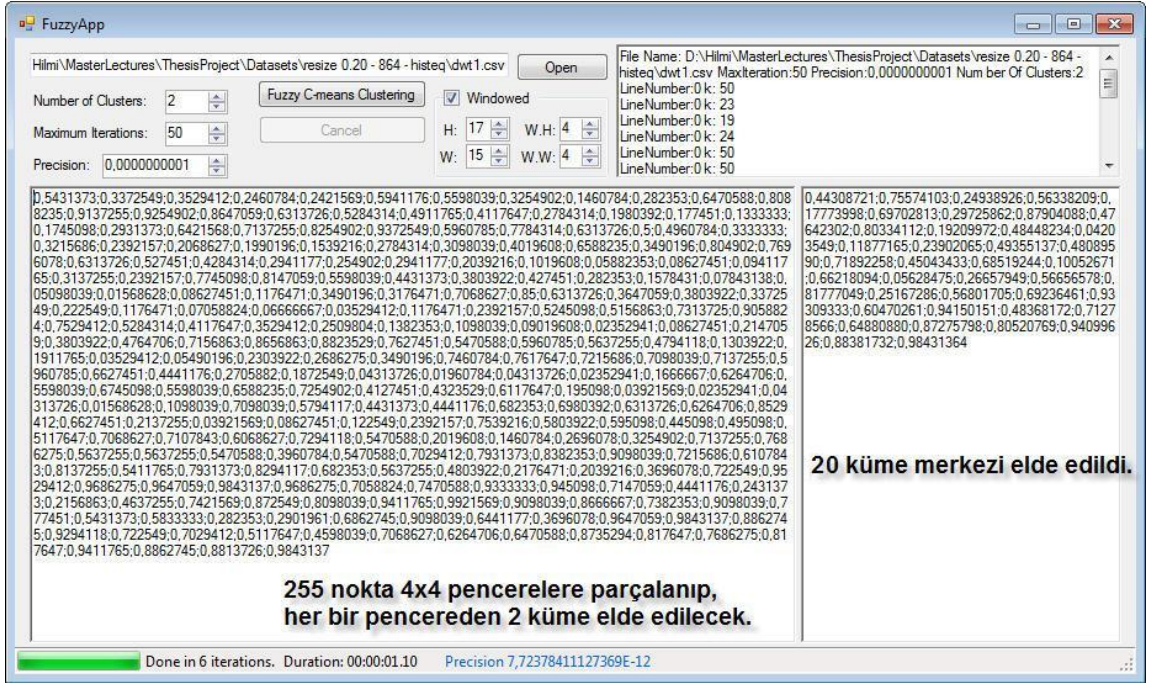
Şekil 10.3. Matlab ile iki boyutlu verinin tek boyuta indirgenmesi

Normalizasyon Uygulaması:



Şekil 10.4. Matlab ile elde edilen özellik vektörüne normalize eden program

FCM Uygulaması:



Şekil 10.5. BCO Uygulması

YSA Uygulaması:

ANNAp D:\Hilmi\MasterLectures\ThesisProject\Datasets\resize 0.20 - 864 - histeq\iris_dwt2_L1- Rounded.csv

Training Data Select File Reset Seperator ;

Threshold Error Learning Rate Momentum Neurons Count Chart Interval
0,40 0,1 0,5 0,75 129 100

Train Train End Test Auto Test

Duration : 00:01:04 Real Value 1 ROC Test

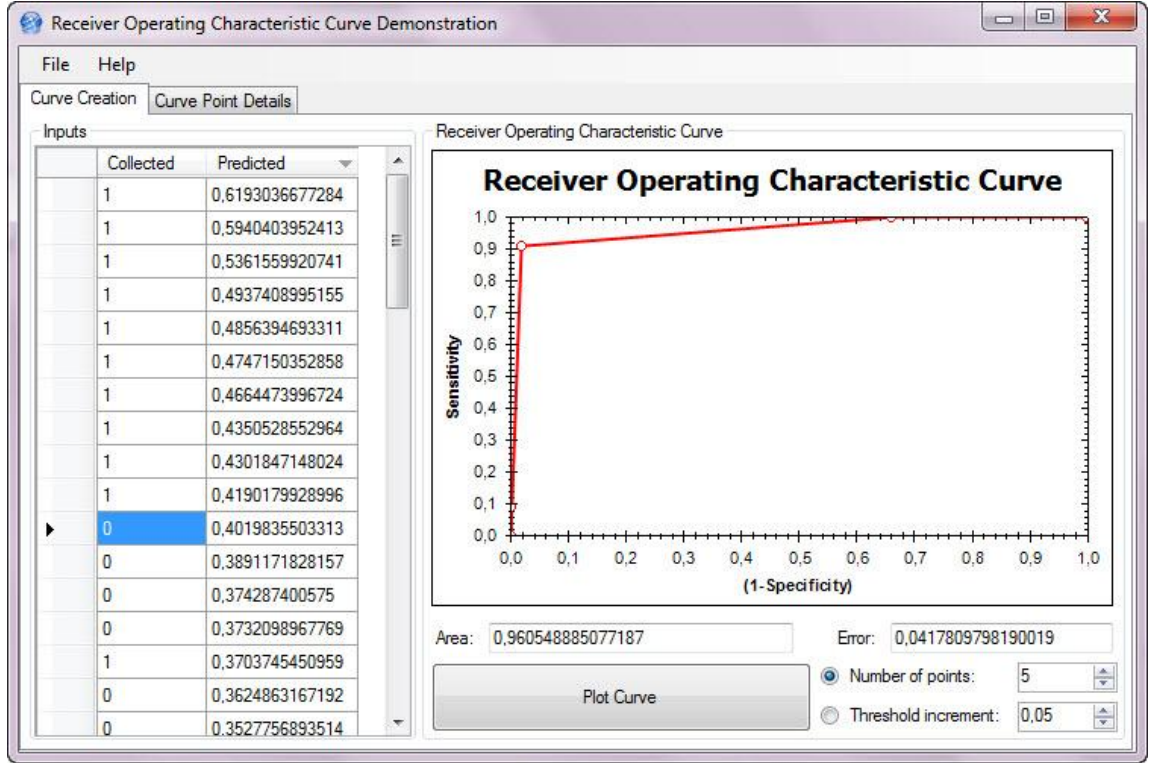
Iteration : 2739, Total Square : 0,099990260!

43. Tip	0,01
1. Tip	0,02
22. Tip	0,01
44. Tip	0,02
2. Tip	0,01
23. Tip	0,01
45. Tip	0,06
3. Tip	0,02
24. Tip	0,02
46. Tip	0,03
4. Tip	0,02
25. Tip	0,03
47. Tip	0,01
5. Tip	0,02
26. Tip	0,01
48. Tip	0,04
6. Tip	0,02
27. Tip	0,04
49. Tip	0,02
7. Tip	0,03
28. Tip	0,05
50. Tip	0,01
8. Tip	0,01
29. Tip	0,03
51. Tip	0,01
9. Tip	0,01
30. Tip	0,02
52. Tip	0,02
10. Tip	0,00
31. Tip	0,01
53. Tip	0,03
11. Tip	0,01
32. Tip	0,01
54. Tip	0,03
12. Tip	0,02
33. Tip	0,02
55. Tip	0,02
13. Tip	0,02
34. Tip	0,01
56. Tip	0,02
14. Tip	0,02
35. Tip	0,01
57. Tip	0,02
15. Tip	0,02
36. Tip	0,01
58. Tip	0,02
16. Tip	0,03
37. Tip	0,02
59. Tip	0,08
17. Tip	0,04
38. Tip	0,05
60. Tip	0,02
18. Tip	0,02
39. Tip	0,01
61. Tip	0,02
19. Tip	0,02
40. Tip	0,01
62. Tip	0,02
20. Tip	0,01
41. Tip	0,02
63. Tip	0,02
21. Tip	0,02
42. Tip	0,02
64. Tip	0,83

Input Count : 255

Şekil 10.6. YSA ekranı

ROC Analizi:



Şekil 10.7. ROC Analizi Uygulaması 1

The 'Point details' table is as follows:

Cutoff Value	True Positives	True Negatives	False Positives	False Negatives	Specificity	Sensitivity	Efficiency	Accuracy	Positive Predictive Value	Negative Predictive Value	False Positive Rate	Phi
0,00000	64	0	5	0	0,00000	1,00000	0,50000	0,32754	0,92754	1,00000	1,00000	0,00000
0,20000	62	2	3	2	0,40000	0,96875	0,68438	0,92754	0,95385	0,50000	0,60000	0,42989
0,40000	61	3	2	3	0,60000	0,95313	0,77656	0,92754	0,96825	0,50000	0,40000	0,52617
1,00000	0	5	0	64	1,00000	0,00000	0,50000	0,07246	1,00000	0,07246	0,00000	0,00000
0,80000	49	5	0	15	1,00000	0,76563	0,88281	0,78261	1,00000	0,25000	0,00000	0,43750
0,60000	58	5	0	6	1,00000	0,90625	0,95313	0,91304	1,00000	0,45455	0,00000	0,64182

Şekil 10.8. ROC Analizi Uygulaması 2