T.C.

SELÇUK ÜNİVERSİTESİ

FEN BİLİMLERİ ENSTİTÜSÜ

ESNEK HESAPLAMA YÖNTEMLERİ İLE OTOMATİK PARMAKİZİ TANIMA

Adem Alpaslan ALTUN

DOKTORA TEZİ

ELEKTRİK-ELEKTRONİK MÜHENDİSLİĞİ ANABİLİM DALI

Konya, 2007

T.C.

SELÇUK ÜNİVERSİTESİ

FEN BİLİMLERİ ENSTİTÜSÜ

ESNEK HESAPLAMA YÖNTEMLERİ İLE OTOMATİK PARMAKİZİ TANIMA

Adem Alpaslan ALTUN

DOKTORA TEZİ

ELEKTRİK-ELEKTRONİK MÜHENDİSLİĞİ ANABİLİM DALI

Bu tez 06 / 04 / 2007 tarihinde aşağıdaki jüri tarafından oybirliği / oyçokluğu ile kabul edilmiştir.

Prof. Dr. Novruz ALLAHVERDİ
(Danışman)

Prof. Dr. Bekir KARLIK
(Üye)

Prof. Dr. Ahmet ARSLAN
(Üye)

Yrd. Doç. Dr. Yüksel ÖZBAY
(Üye)

Yrd. Doç. Dr. Mehmet ÇUNKAŞ
(Üye)

ÖZET

Doktora Tezi

ESNEK HESAPLAMA YÖNTEMLERİ İLE OTOMATİK PARMAKİZİ TANIMA

Adem Alpaslan ALTUN

Selçuk Üniversitesi Fen Bilimleri Enstitüsü

Elektrik-Elektronik Mühendisliği Anabilim Dalı

Danışman : Prof. Dr. Novruz ALLAHVERDİ

2007, 228 Sayfa

Jüri : Prof. Dr. Novruz ALLAHVERDİ

Prof. Dr. Bekir KARLIK

Prof. Dr. Ahmet ARSLAN

Yrd. Doç. Dr. Yüksel ÖZBAY

Yrd. Doç. Dr. Mehmet ÇUNKAŞ

Bu çalışma, kişilere ait parmakizi görüntülerini kullanarak esnek hesaplama yöntemleri yardımıyla bu kişileri tanımak için yeni bir yöntem geliştirmek amacıyla yapılmıştır. Bu amaçla NIST-4 veritabanından 10 kişiye ait toplam 1000 adet parmakizi görüntüsü kullanılmıştır. Parmakizi tanıma aşamasında kullanılan görüntünün kalitesi tanıma aşamasındaki performansı etkilemektedir. Bu parmakizi görüntüleri elde edilirken dış etkenlerden dolayı gürültü içerirler. Bu yüzden bu parmakizi görüntülerine öncelikle görüntü iyileştirme işlemi gerçekleştirilir. Bunun

için görüntü iyileştirme teknikleri araştırılmış ve geleneksel yöntem olan lineer, medyan ve uyarlamalı filtrelerle birlikte dalgacık dönüşümü ve dalgacık dönüşümün geliştirilmişi olan çevritsel dönüşüm kullanılarak parmakizi görüntüleri gürültüden arındırılmaya çalışılarak iyileştirilmiştir. İyileştirilen bu görüntülerin en büyük sinyal-gürültü oranına göre performansları ölçülmüştür. En iyi ön iyileştirme performansını çevritsel dönüşümün verdiği gözlemlenmiştir.

Çalışmanın ikinci aşamasında iyileştirilmiş parmakizi görüntülerinden FingerCode adı verilen özellik vektörleri geliştirilen bir yöntemle elde edilmiştir. Elde edilen bu özellik vektörleri Yapay Sinir Ağı ile eğitilerek parmakizi tanıma işlemi gerçekleştirilmiştir. Bunun için toplam 1000 parmakizi görüntüsünden 700 tanesi YSA'nın eğitimi aşamasında, 150 tanesi doğrulama aşamasında ve 150 tanesi test aşamasında kullanılmıştır. YSA ile hatanın geriye yayılması algoritmaları kullanılarak yapılan eğitme sonucunda en iyi performansı %100 eğitim doğruluğu ve %99,6 genel tanıma oranı ile Eşleştirmeli Eğim Azaltımlı (Conjugate Gradient Descent) olarak adlandırılan algoritma vermiştir.

Çalışmanın son aşamasında elde edilen özellik vektörlerinin büyük boyutlu olmasının YSA'daki eğitim süresini etkilediği gözlemlenmiştir. Dolayısı ile özellik vektörlerinin boyutunun azaltılması gerektiği ortaya çıkmıştır. Bunun için özellik azaltımı, veri azaltımı ve boyut azaltımı konuları araştırılmıştır. Özellik azaltım yöntemleri arasından en uygun olanının bu alanda sıklıkla kullanılan Genetik Algoritmalar olduğu gözlemlenmiştir. Bu yüzden parmakizi görüntülerine ait özellik vektörleri boyutunun azaltılması için Genetik Algoritmalar kullanılmıştır. Böylece gürültü içeren ve daha az etkin özellikler çıkarılarak etkin özelliklerin seçimi yapılmıştır. Seçilmiş özellik vektörleri yeniden YSA'da eğitilerek tanıma işlemi gerçekleştirilmiştir. Elde edilen sonuçlara göre, GA kullanılarak boyutu azaltılan özellik vektörleri kullanılarak yapılan tanıma işlemi daha kısa sürede gerçekleştirilmiş ve %99,5 genel tanıma oranı ile özellik vektörleri boyutlarının azaltılmamış durumuna çok çok yakın bir tanıma oranı elde edildiği gözlemlenmiştir.

Anahtar Kelimeler: Parmakizi tanıma, yapay sinir ağları, genetik algoritmalar, görüntü iyileştirme, çevritsel dönüşüm, esnek hesaplama.

ABSTRACT

PhD Thesis

AUTOMATIC FINGERPRINT RECOGNITION BY USING SOFT COMPUTING TECHNIQUES

Adem Alpaslan ALTUN

Selcuk University

Graduate School of Natural and Applied Sciences

Department of Electrical and Electronics Engineering

Supervisor : Prof. Dr. Novruz ALLAHVERDİ

2007, 228 pages

Jury : Prof. Dr. Novruz ALLAHVERDİ

Prof. Dr. Bekir KARLIK

Prof. Dr. Ahmet ARSLAN

Assist. Prof. Dr. Yüksel ÖZBAY

Assist. Prof. Dr. Mehmet ÇUNKAŞ

This study is realized to recognize the people by a new generated method with the assistance of soft computing methods. 1000 fingerprint images that belong to 10 people are used from the NIST-4 database. The quality of the images of the fingerprints that are used in the process of the recognition affects the performance of recognition. These fingerprint images include interferences due to the external effects. For this reason the images of fingerprints are processed initially for image

enhancement. Because of all of these , image enhancement methods are investigated and in addition to the conventional methods that are named as linear , median and adaptive filters, wavelet transform and the improved version of wavelet transform called as “Contourlet transform” are used to separate the fingerprint images from the interferences. The fingerprint images are enhanced by using these techniques. The performances of these images are measured according to the peak signal to noise ratio. The best performance is obtained from Contourlet transform.

At the second stage of the study, the feature vector called as Fingercode is obtained by means of a generated method. The obtained feature vectors are trained by artificial neural networks and the fingerprint recognition is performed. For this purpose, from totally 1000 fingerprints, 700 fingerprints are used in the process of ANN training , 150 fingerprints are used in the process of validation and 150 fingerprints are used in the process of testing.

After performing the training by using the back propagation algorithm the best performance is obtained from the algorithm named as Conjugate Gradient Descent with %100 training accuracy and %99,6 general recognition rate .

In the last period of the study, the size of the feature vector is taken into account and it was observed that the size affects the time of training. Thus, the reduction of the size is needed. Because of this reason the topics of reduction of feature, data and size are investigated. It is observed that the most suitable method is Genetic Algorithm that is used frequently in this field. For this reason, genetic algorithms are used for the reduction of the size of the feature vector that belongs to the images of the fingerprints. Thus, by eliminating the features with interference, selection of the features are made. The selected features are retrained and recognition process is performed by using ANN. According to the derived results, the training of the selected features in the ANN took less time. The recognition rate is obtained as well as the previous application.

Keywords: Fingerprint recognition, artificial neural networks, genetic algorithms, image enhancement, contourlet transform, soft computing.

TEŞEKKÜR

Bu çalışma, Selçuk Üniversitesi, Fen Bilimleri Enstitüsü, Elektrik-Elektronik Mühendisliği Anabilim Dalı'nda ve Öğretim Üyesi Sayın Prof. Dr. Novruz ALLAHVERDİ yönetiminde hazırlanarak, Selçuk Üniversitesi, Fen Bilimleri Enstitüsüne Doktora Tezi olarak sunulmuştur.

Bu çalışmanın gerçekleşmesinde bana yol gösteren ve her türlü bilimsel katkıyı sağlayan değerli hocam ve danışmanım Sayın Prof. Dr. Novruz ALLAHVERDİ'ye teşekkür ederim. Kendileri ile çalıştığım altı yıl boyunca bana her türlü desteği sağlayan, çalışmanın önemli noktalarında yol gösteren ve algoritmaların seçiminde yardımcı olan Tez İzleme Komitesi üyeleri S.Ü. Bilgisayar Mühendisliği Bölüm Başkanı Sayın Prof. Dr. Ahmet ARSLAN'a ve S.Ü. Elektrik-Elektronik Mühendisliği Öğretim Üyesi Sayın Yrd. Doç. Dr. Yüksel ÖZBAY'a teşekkür ederim.

Ayrıca bu çalışmam sırasında büyük katkıları bulunan ve çalışma esnasında karşılaştığım sıkıntılar konusunda sabır ve desteğini esirgemeyen mesai arkadaşlarıma ve özellikle kullandığım bazı algoritmaların seçiminde bana yol gösteren Sayın Yrd. Doç. Dr. Mehmet ÇUNKAŞ'a minnetlerimi sunarım.

Bunun yanında akademik çalışmalarımı ve bu çalışma ile ilgili yurtdışında sunacağım bildirileri daima destekleyen Selçuk Üniversitesi Bilimsel Araştırma Projeleri Koordinatörlüğü ve değerli hocalarına teşekkürlerimi sunarım.

Bu meşakkatli çalışmanın başından sonuna kadar bana her türlü sabrı, fedakârlığı ve hoşgörüyü gösteren ve her türlü sıkıntıyı benimle paylaşan eşim Türkan ALTUN'a ve biricik kızım Aybüke Ceyda ALTUN'a en derin şükran duygularımı belirtmek istiyorum.

Adem Alpaslan ALTUN

Konya, 2007

İÇİNDEKİLER

ÖZET……..... i

ABSTRACT……...… iii

TEŞEKKÜR……... v

İÇİNDEKİLER……..… vi

ŞEKİLLER……..... viii

TABLOLAR……...… xii

SİMGELER ve KISALTMALAR……...… xiv

1. GİRİŞ……. 1
1.1. Tezin Organizasyonu……... 5
1.2. Çalışmanın Amacı ve Önemi…….. 8
1.3. Kaynak Araştırması…… 10

2. BİYOMETRİK SİSTEMLER ve PARMAKİZİ…… 21
2.1. Genetik Faktörler…… 23
2.2. Çevresel Faktörler……... 27
2.3. Parmakizinin Bireyselliği……..... 29
2.4. Parmakizinin Karakteristik Özellikleri…… 32
2.5. Parmakizi Tanıma Sistemleri…….. 33
2.6. Parmakizi Eşleştirme……... 37
2.7. Sonuç…… 39

3. PARMAKİZİ ÖN İŞLEME……. 40
3.1. Filtreleme Teknikleri Kullanarak Görüntü İyileştirme……... 41
3.2. Kontrast İyileştirme……. 42
3.3. Gürültü Azaltımı için Filtreleme……. 44
3.4. Dalgacık Dönüşüm ve Çevritsel Dönüşüm ile Parmakizi Görüntüsü İyileştirme……... 49
3.4.1. Dalgacık dönüşüm ile parmakizi görüntüsü iyileştirme…….. 49
3.4.2. Çevritsel dönüşüm ile parmakizi görüntüsü iyileştirme…….. 54
3.5. Sonuç……..…….. 59

4 PARMAKİZİ ÖZELLİK VEKTÖRÜ…… 61
4.1. Parmakizi Referans Noktasının Bulunması……. 64
4.2. Parmakizinin Sektörizasyonu (Mozaikleme)…….. 71
4.3. Parmakizi Filtreleme……... 74
4.4. Parmakizi Özellik Vektörünün Elde Edilmesi…… 77
4.5. Sonuç……..…….. 80

5. YAPAY SİNİR AĞLARI İLE PARMAKİZİ TANIMA……..... 82
5.1. Yapay Sinir Ağları…….... 84
5.2. Parmakizi Özellik Vektörlerinin YSA'da Eğitimi ve Testi…… 84
5.3. Parmakizi Tanımada YSA Yapısı……..... 89
5.4. Yapay Sinir Hücresi…… 90
5.5. Yapay Sinir Ağı Yapıları…… 92
5.5.1. Parmakizi tanımada YSA tasarımı……... 92
5.5.2. YSA yapısının ve öğrenme algoritmasının seçimi……... 93
5.5.3. Ara katman sayısının ve nöron sayısının belirlenmesi…… 95
5.5.4. Parmakizlerine ait özellik vektörlerinin normalizasyonu…… 98
5.5.5. Performans fonksiyonun seçimi……..... 100
5.6. Yapay Sinir Ağlarının Sınıflandırılması……..... 102
5.7. Parmakizi Tanımada Kullanılan YSA Öğrenme Algoritmaları……..... 103
5.7.1. Hızlı yayılım algoritması……..... 107
5.7.2. Yığın geri yayılım…… 109
5.7.3. Artışlı geri yayılım……..... 110
5.7.4. Eşleştirmeli eğim algoritması……..... 111
5.8. Parmakizi Tanımada Çok Katmanlı Ağların Performansı…….. 113
5.9. YSA Metotlarının Parmakizi Tanıma Performansları ve Analizi…… 114
5.10. Sonuç……..... 131

6. GENETİK ALGORİTMALAR KULLANILARAK PARMAKİZİ ÖZELLİK VEKTÖRLERİNİN AZALTILMASI……. 135
6.1. Veri Azaltma Teknikleri ile Parmakizi Özellik Vektörü Boyutunun Azaltılması…….. 136
6.2. Genetik Algoritmalar…….. 139
6.3. Özellik Vektörlerinin Genetik Algoritma için Kodlanması……. 141
6.4. İlk populasyonun oluşturulması…….. 144
6.5. Uygunluk Teknikleri……..... 144
6.5.1. Olasılıksal sinir ağı…….. 146
6.6. Genetik Operatörler……..... 148
6.6.1. Seleksiyon……... 149
6.6.2. Çaprazlama…….. 150
6.6.3. Mutasyon……..... 152
6.6.4. Yeni nesilin üretilmesi ve özellik seçimi……. 153
6.7. Özellik Vektörü Azaltmada GA'nın Performansını Etkileyen Etmenler. 154
6.8. Azaltılmış Özellik Vektörlerinin YSA ile Parmakizi Tanımada Uygulama Sonuçları……. 156
6.9. Sonuç……..... 165

7. SONUÇLAR...……... 168

8. TARTIŞMA ve ÖNERİLER……. 172

9. KAYNAKLAR……..... 179

EKLER……..... 200

ŞEKİLLER

Şekil 2.1 Çeşitli biyometrikler……. 21

Şekil 2.2 Digital Biometrics (2004) şirketine ait optik tarayıcı kullanarak elde edilen iki ikiz kız kardeşe ait parmakizi görüntüleri……..... 24

Şekil 2.3 Jain ve ark. (1997)'nın uyguladığı ayrıntı çıkarım algoritması kullanılarak elde edilmiş ikizlerin parmakizi ayrıntıları…….. 25

Şekil 2.4 Ayrıntı eşleştirme……. 26

Şekil 2.5 Aynı parmakizinin farklı zamanlarda alınmış örneklerinden elde edilmiş ayrıntıların eşleştirilmesi……. 26

Şekil 2.6 Sağ kement tipli bir parmakizi görüntüsü ve ayrıntı noktaları……. 27

Şekil 2.7 Parmakizi sınıflandırma ve tanıma işleminin genel aşamaları……. 29

Şekil 2.8 Parmakizlerinin karakteristik özellikleri…….. 32

Şekil 2.9 Parmakizi özelliklerinin belirlenmesi aşamaları…….. 36

Şekil 2.10 İki farklı parmakizine ait elde edilen yönsel histogram……... 37

Şekil 2.11 Parmakizi okuma sensörü eklenmiş bilgisayar çevre aygıtları……..... 38

Şekil 3.1 Filtreleme tabanlı parmakizi görüntüsü iyileştirme aşamaları……. 41

Şekil 3.2 a) Orijinal parmakizi görüntüsü ve b) histogramı; c) lokal histogram eşitleme sonucu elde edilen parmakizi görüntüsü ve d) yeni histogram skalası…….. 43

Şekil 3.3 İki boyutlu görüntüye medyan filtre uygulanması……. 46

Şekil 3.4 Filtreleme teknikleri uygulanarak iyileştirilmiş parmakizi görüntüleri... 47

Şekil 3.5 İki boyutlu dalgacık analiz filtre seti ve ayrışım sonuçları…….. 51

Şekil 3.6 Dalgacık dönüşümü uygulanmış parmakizi görüntüsü…… 54

Şekil 3.7 a) İki boyutlu dalgacık dönüşümünün ve b) çevritsel dönüşümün temel fonksiyon sonuçları…… 55

Şekil 3.8 a) Dalgacık dönüşüme ait ve b) çevritsel dönüşüme ait görüntü yakalama performansları……..... 55

Şekil 3.9 Eğri ölçekleme ilişkisini sağlayan çevritsel tabanlı görüntülerin gösterilmesi……..... 56

Şekil 3.10 a) Dalgacık dönüşüm sonucu elde edilen parmakizi görüntüsü, b) Çevritsel dönüşüm uygulanması sonucu elde edilmiş parmakizi görüntüsü……... 57

Şekil 3.11 NIST-4 parmakizi veritabanında bulunan parmakizlerine çevritsel dönüşüm uygulanarak iyileştirilmiş parmakizleri…… 58

Şekil 3.12 Farklı parmakizi görüntülerine uygulanmış filtreleme metotları sonucu elde edilmiş PSNR değerleri…….. 59

Şekil 4.1 Parmakizi görüntüsünün akış deseni…….……... 62

Şekil 4.2 Benzer özelliklere sahip olmalarına rağmen iki farklı parmakizi görüntüsü…….. 62

Şekil 4.3 Parmakizleri için doku-tabanlı simgelenişinin çıkarım şeması……..... 64

Şekil 4.4 Parmakizi görüntüsündeki içbükey ve dışbükey hatlar……..... 65

Şekil 4.5 Referans noktasının belirlenmesi……. 67

Şekil 4.6 À(i,j)'yi hesaplamak için ε'daki piksel yoğunluk bölgeleri……. 69

Şekil 4.7 Referans nokta tayini algoritmasının sonuçları……..... 70

Şekil 4.8 a) -45^o döndürülmüş, b) +45^o döndürülmüş ve c) Normal parmakizi görüntülerine ait sektörizasyon işleminin başlangıç yönleri…..……... 72

Şekil 4.9 İlgili bölgenin referans noktası…… 72

Şekil 4.10 Normalize edilmiş, filtrelenmiş ve iyileştirilmiş parmakizi görüntüleri.. 75

Şekil 4.11 Gabor filtre……... 77

Şekil 4.12 İki farklı parmakizine ait 640-boyutlu özellik vektörleri……. 79

Şekil 5.1 a) Genelleme eğilimi gösteren YSA, b) Ezberleme eğilimli YSA……... 86

Şekil 5.2 a) Verileri ezberleyen ve b) genellemeye ulaşan ağlardaki hata eğrileri. 86

Şekil 5.3 Parmakizi özellik vektörlerinden oluşan YSA eğitim, doğrulama ve test veri kümeleri…….. 88

Şekil 5.4 Yapay Sinir Ağı ve Yapay Sinir Hücresi (nöron) Modeli…….…….... 90

Şekil 5.5 Tecrübeye dayalı arama metodu ile belirlenen en uygun YSA yapıları.. 97

Şekil 5.6 Parmakizi özellik vektörlerinin normalizasyon sonucu yeni değerleri… 100

Şekil 5.7 Geri yayılım algoritmasının akış şeması…….. 104

Şekil 5.8 YSA eğitim ve doğrulama veri kümeleri doğru sınıflandırma oranları... 119

Şekil 5.9 a) Eğitim, b) test ve c) tüm veri kümelerine ait ortalama doğruluk oranları ve eşleştirme matrisleri……... 119

Şekil 5.10 Sekiz çıkışlı YSA eğitim ve doğrulama kümelerine ait doğru sınıflandırma oranları……... 123

Şekil 5.11 Sekiz çıkışlı YSA'ya uygulanan a) eğitim, b) test ve c) tüm veri kümelerine ait ortalama doğruluk oranları ve eşleştirme matrisleri…… 123

Şekil 5.12 Sekiz çıkışlı YSA'da eğitme aşamasına dahil edilmeyip test kümesine dahil edilen parmakizlerine ait test kümesi ve tüm kümeye ait eşleştirme matrisleri……. 124

Şekil 5.13 Diğer iki kişinin parmakizlerine ait eşleştirme oranlarının eşik değerinden geçirildikten sonra elde edilen test kümesi ve tüm kümeye ait eşleştirme matrisleri…… 125

Şekil 5.14 Dokuz çıkışlı YSA eğitim ve doğrulama veri kümeleri için doğru sınıflandırma oranları……... 126

Şekil 5.15 Dokuz çıkışlı YSA için a) eğitim, b) test ve c) tüm veri kümelerine ait ortalama doğruluk oranları ve eşleştirme matrisleri……..... 126

Şekil 5.16 Eğitme aşamasına dahil edilmeyip test kümesine dahil edilen dokuzuncu kişinin parmakizi bilgilerine ait test kümesi ve tüm kümeye ait eşleştirme matrisleri…… 128

Şekil 5.17 Eğitim aşamasına dahil edilmeyip test kümesine dahil edilen dokuzuncu kişinin parmakizlerine ait eşleştirme oranlarının eşik değerinden geçirildikten sonra elde edilen test kümesi ve tüm kümeye ait eşleştirme matrisleri…….……... 129

Şekil 5.18 Dokuz çıkışlı YSA eğitim ve doğrulama veri kümeleri için doğru sınıflandırma oranları……. 130

Şekil 5.19 Dokuz çıkışlı YSA için a) eğitim, b) test ve c) tüm veri kümelerine ait ortalama doğruluk oranları ve eşleştirme matrisleri……..... 130

Şekil 6.1 İleri yönlü sezgisel seçim kullanılarak en iyi özellikleri seçme işlemi… 137

Şekil 6.2 Geri yönlü sezgisel seçim kullanılarak en iyi özellikleri seçme işlemi… 138

Şekil 6.3 Karar ağaçları seçim tekniğini kullanarak en iyi özellikleri seçme işlemi……..... 138

Şekil 6.4 Genetik algoritmanın akış şeması……..... 141

Şekil 6.5 Ağaç kodlamalı kromozomlar örneği…… 143

Şekil 6.6 Olasılıksal sinir ağı yapısı……. 147

Şekil 6.7 Tek noktalı çaprazlama sonucu elde edilen yeni kromozomlar…….. 151

Şekil 6.8 GA ile en uygun kromozom yapısının belirlenmesi…… 154

Şekil 6.9 Özellik vektörleri azaltılmış sekiz çıkışlı YSA eğitim ve doğrulama veri kümeleri için doğru sınıflandırma oranları……..... 158

Şekil 6.10 Özellik vektörleri azaltılmış sekiz çıkışlı YSA için a) eğitim, b) test ve c) tüm veri kümelerine ait ortalama doğruluk oranları ve eşleştirme matrisleri…….. 158

Şekil 6.11 Özellik vektörleri azaltılmış dokuz çıkışlı YSA eğitim ve doğrulama veri kümeleri için doğru sınıflandırma oranları……..... 160

Şekil 6.12 Özellik vektörleri azaltılmış dokuz çıkışlı YSA için a) eğitim, b) test ve c) tüm veri kümelerine ait ortalama doğruluk oranları ve eşleştirme matrisleri……..... 160

Şekil 6.13 Özellik vektörleri azaltılmış on çıkışlı YSA eğitim ve doğrulama veri kümeleri için doğru sınıflandırma oranları……..... 161

Şekil 6.14 Özellik vektörleri azaltılmış on çıkışlı YSA için a) eğitim, b) test ve c) tüm veri kümelerine ait ortalama doğruluk oranları ve eşleştirme matrisleri……. 162

TABLOLAR

Tablo 2.1 Araştırmacıların kullandıkları parmakizi ayrıntıları……..……... 28

Tablo 3.1 Filtrelemeler sonucu elde edilen görüntülerin lokal histogram eşitleme uygulanmış görüntüyle karşılaştırma sonuçları……... 48

Tablo 3.2 İki boyutlu ölçekleme fonksiyonu ve dalgacık fonksiyonları…….. 51

Tablo 4.1 Bant sayısına göre belirlenen bant genişlikleri…….……..... 73

Tablo 5.1 YSA öğrenme algoritmaları ve uygulama alanları……..... 94

Tablo 5.2 Parmakizi tanımada uygulanan YSA yapıları ve parametre değerleri…. 106

Tablo 5.3 Tecrübeye dayalı arama yöntemi ile elde edilen en uygun YSA yapıları…….…….…….…….…….…. 115

Tablo 5.4 Görüntü iyileştirme metotlarına göre YSA yöntemleri eğitim ve test sonuçları…….…….…….…….……... 116

Tablo 5.5 Parmakizi özellik vektörlerinin YSA'da eğitildikten sonra test edilmesi sonucu elde edilen eşleştirme oranları…….…… 120

Tablo 5.6 Eğitim kümesine dahil olmayan parmakizlerinin YSA'daki test sonuçları…….…….…….…….……..... 122

Tablo 5.7 Parmakizi özellik vektörlerinin YSA'da eğitildikten sonra test edilmesi sonucu hatalı olarak eşleştirilen özellik vektörleri için elde edilen eşleştirme oranları…….…….…….……. 122

Tablo 5.8 Dokuz çıkışlı YSA'da "diğer" çıkışına dahil edilen parmakizlerinin yeniden eğitilerek test edilmesi sonucu elde edilen eşleştirme oranları…….…….…….……. 127

Tablo 6.1 İkili dizi kodlama…….…….…….…….……. 142

Tablo 6.2 Permutasyon kodlamalı kromozom örnekleri…….……. 143

Tablo 6.3 Değer kodlamalı kromozom örnekleri…….…….……..... 143

Tablo 6.4 Biyolojik çaprazlama…….…….…….…… 151

Tablo 6.5 İkili kodlama sistemine uygulanmış mutasyon operatörü……... 153

Tablo 6.6 Tecrübeye dayalı arama yöntemi kullanılarak elde edilen en uygun YSA yapıları…… 163

Tablo 6.7 Görüntü iyileştirme metotlarına göre YSA yöntemleri eğitim ve test sonuçları……..... 163

Tablo 7.1 YSA eğitim süresinin test edildiği bilgisayar konfigürasyonları……. 170

Tablo 7.2 İşlemci frekansına göre YSA eğitimi ve GA ile özellik azatlımı süresi……... 171

Tablo 8.1 Eşleştirme matrisi……. 173

Tablo 8.2 NIST-4 kullanılarak değerlendirilen sınıflandırma algoritmasının özeti……. 173

Tablo 8.3 Parmakizi sınıflandırma algoritmalarının doğruluk oranları……... 174

Tablo 8.4 Parmakizi tanıma algoritmalarının doğruluk oranları……. 175

SİMGELER ve KISALTMALAR

r_k	k'ıncı piksel gri seviyesi yoğunluğu, $0<r_k<1$, $k=0,1,2,...,255$
n_k	r_k gri seviyesindeki piksellerin sayısı
n	Toplam piksel sayısı
p_r	Yoğunluk fonksiyonu
A	İki boyutlu görüntü matrisi
$[m\ n]$	Medyan filtrenin pencere boyutu
B	Filtrelenmiş görüntü matrisi
$f(n_1, n_2)$	Orijinal Parmakizi Görüntüsü, $1\leq n_1\leq N$, $1\leq n_2\leq M$
$\hat{f}(n_1, n_2)$	İyileştirilmiş Parmakizi Görüntüsü
a	Ölçekleme/yayılma katsayısı
b	Dönüşüm/öteleme katsayısı
$f(t)$	İşaret
ψ	Dalgacık fonksiyonu
$W(a,b)$	İşaretin sürekli dalgacık dönüşümü
$\psi^Y(x,y)$	Yatay Dalgacık Fonksiyonu
$\psi^D(x,y)$	Dikey Dalgacık Fonksiyonu
$\psi^K(x,y)$	Köşegensel Dalgacık Fonksiyonu
φ	Ölçekleme fonksiyonu
s	Orijinal görüntü
sg	Gürültülü görüntü
$e(n)$	Gaussian beyaz gürültüsü
σ	Gürültü seviyesi
t	Eşikleme noktası

$\grave{A}(x_c,y_c)$	Referans noktası
F	Parmakizi görüntüsü
O	Yönelim alanı
$O(i,j)$	Bölgesel hat yönelimi
w	Blok boyutu
$\partial_x(i,j)$	Yatay eğim
$\partial_y(i,j)$	Dikey eğim
O'	Düzeltilen yönelim alanı
S_i	Sektör
θ_0	Referans nokta bloğunun açısal değeri
b	Her bir bandın genişliği
k	Her bir bantta hesaba katılan sektör sayısı
B	Eşmerkezli bantların sayısı
M_i	Ortalama değer
V_i	Varyans değeri
$N_i(x,y)$	Normalize edilmiş gri seviye değeri
$G(x,y;f,\theta)$	Gabor filtresi
f	x ekseninden θ açısı boyunca uzanan sinüs dalgasının frekansı
K	Hatlar-arası ortalama mesafe
$V_{i\theta}$	Parmakizi özellik vektörü
$f(v)$	Aktivasyon fonksiyonu
w	Ağırlıklar matrisi
x	Giriş vektörü
v	Ağ girişi
y	Çıkış fonksiyonu
θ	Bias
n	Parametre sayısı

e_i^2	Hesaplanan gözlem ve hesap sonuçları arasındaki farkın varyansı
X_{yeni}	Ölçeklendirme fonksiyonu
X_{min}	Veri sütununun en düşük değer
X_{max}	Veri sütununun en yüksek değer
SF	Ölçeklendirme faktörü
X	Özellik vektör değeri
SR_{min}	En küçük ölçekleme değeri
SR_{max}	En büyük ölçekleme değeri
X_p	Ölçeklenme sonucu elde edilen yeni değer
y	Gerçek çıkış değeri
y'	Hesaplanan çıkış değeri
η	Öğrenme katsayısı
α	Momentum
$\Delta w_{ij}(t)$	Ağırlık değişim fonksiyonu
t	İterasyon sayısı
ε	Öğrenme katsayısı
μ	Momentum büyüme faktörü
$h(t)$	Eğim fonksiyonu
$\Delta q(t)$	Minimum adım miktarı
δ	Hızlandırma katsayısı
κ	Ağırlık kırpma faktörü
P	Performans oranı
D	Doğru örnek sayısı
T	Toplam örnek sayısı
V_w	Objektif değeri
f_i	Uygunluk değeri
d	Eksilme oranı

σ	Yumuşatma faktörü
i	Sınıf sayısı
j	Örüntü sayısı
x_{ij}	i. sınıf j. popülasyonun eğitim vektörü
f_i	Gauss toplam fonksiyonu
T	Transpoze
P_i	i. bireyin olasılık değeri
F_i	i. bireyin uygunluk değerini
P_s	Populasyondaki birey sayısı
L	Gen sayısı

YSA	Yapay Sinir Ağları
GA	Genetik Algoritmalar
PIN	Personel kimlik saptama numarası
NIST	Ulusal Standartlar ve Teknoloji Enstitüsü
QP	Hızlı Geriyayılım
CGD	Eşleştirmeli Eğim Azaltımlı
IBP	Artışlı Geriyayılım
BBP	Yığın Geriyayılım
GMA	Gizli Markov Ağacı
ROC	İşlem Karakteristiği Eğrisi
ANFIS	Adaptif Sinirsel-Bulanık Çıkarım Ağı
AFIS	Otomatik Parmakizi Tanıma Sistemi
AFAS	Otomatik Parmakizi Doğrulama Sistemi
PSNR	En-uç Sinyal Gürültü Oranı
SNR	Sinyal Gürültü Oranı
DFB	Yönsel Filtre Kümesi

LP	Laplas Piramidi
PDFB	Piramidal Yönsel Filtre Kümesi
TRN	Eğitim veri kümesi
TST	Test veri kümesi
VLD	Doğrulama veri kümesi
AIC	Akaike Bilgi Kriteri
MSE	Karesel Ortalama Hata
SSE	Toplam Karesel Hata
RMS	Karesel Ortalama Hatanın Karekökü
İE	İşlemci Elemanlar
MLP	Çok Katmanlı Perseptron
LVQ	Lineer Vektör Kuantizasyonu
SOM	Özdüzenleyici Haritalar
CCR	Doğru Sınıflandırma Oranı
PNN	Olasılıksal Sinir Ağları
PDF	Olasılıksal Yoğunluk Fonksiyonu
AIC	Akaike bilgi kriteri

1. GİRİŞ

Güvenilir bir şekilde izin ve kimlik denetimi günlük yaşantımızın her anındaki olaylar ya da uygulamalar için gerekli olmaktadır. İzin ve yetkilendirme, tek bir kimliğe bürünmüş veya özel gruplara tanınan bir hak olmaktadır. Kimlik doğrulama, yüksek doğruluk gerektiren veya ret oranı düşük olması istenen yerlerde zor bir görev olmaktadır. Kullanıcı bu işlem esnasında reddedilmek istemez ve kimlik tespiti için bir yığın işlemden de geçmek istemez. Son zamanlarda birey tanımlama, toplumun vazgeçilmez unsurlarından biri olmaktadır. Biyometrik tanımlama ya da kısa adıyla biyometrik ise kişilerin kendilerine has fizyolojik özelliklerini ölçen ve bu özellikleri teşhis ve doğrulama uygulamalarında kullanan genel bir tekniktir (Jain ve Pankanti 2001). Bu tür sistemler, insan beyninin bir kişiyi tanıması ve diğer insanlardan ayırt etmesi yöntemi ile aynı şekilde çalışmaktadır. Biyometrik sistemler; kart, şifre veya pin numarası (personal identification number/personel kimlik saptama numarası) kullanan diğer tanıma yöntemlerine oranla daha çok tercih edilmektedirler. Bu durumun başlıca sebepleri arasında kullanıcının, kimlik saptama yapılacak yerde bizzat bulunma gerekliliği, yanında kendini tanıtmak için kimlik kartı benzeri tanıtıcılar taşımak zorunda olmayışı ve şifre/pin numarası gibi, gizli olması gereken bilgileri ezberlemek zorunda olmaması sayılabilir.

Şifreler veya pin numaraları ağ korsanları tarafından yardımcı programlar kullanılarak birçok kez kırılabileceği için ya da kullanıcılar tarafından sık sık unutulabildiklerinden dolayı yerlerini akıllı kartlar ve biyometrik cihazlar gibi yeni teknolojilere bırakmaktadırlar. Biyometrik cihazlar parmakizi, göz retinası gibi insanların benzeri olmayan karakteristiklerini kullanarak kullanıcıların şifre kullanmaksızın bilgisayar sistemleri, veri bankaları ve benzer güvenlik gerektiren ortamlara giriş için kimlik doğrulaması yapılmasını sağlarlar.

Bilgi teknolojisi araçları olarak bilgisayarların ve internetin her geçen gün etkin bir şekilde kullanılmasıyla birlikte, bu teknolojiyi kullanan kişi veya firmaların kişisel bilgilerine ve gizli verilerine yetkisi olmayan kişi ve kuruluşların ulaşımının engellenmesi zorunlu hale gelmiştir. Bilinen ve yaygın olarak kullanılan erişim kontrol sistemleri, kullanıcıları tanımlamak yerine kullanıcının sunduğu tanıtıcılara onay vermektedir. Biyometrik teknolojiler ise kişileri doğrudan tanıdıkları için yetkisi olmayan kişilerin kişisel bilgilere erişiminin engellenmesi ve ATM, cep telefonu, akıllı kart, masaüstü bilgisayar, iş istasyonu ve bilgisayar ağları gibi sistemlerin uygunsuz kullanımının engellenmesi için en çok başvurulan yöntem olmaktadırlar. Günümüzde çeşitli biyometrik sistemler, eşzamanlı tanımlama uygulamalarında yaygın olarak kullanılmaktadır. Bunlar içerisinde en bilinen sistemler; parmakizi eşleştirme, iris tanıma, retina tarama, ses ve konuşma tanıma, yüz tanıma ve el tanıma olarak sıralanabilir (Jain ve ark. 1999a). Bu sistemlerin her birisi yeteri derecede karmaşık olup, çoğunlukla halen de araştırma ve geliştirme safhalarında bulunmaktadırlar.

Bu çalışmada biyometrik sistemlerden olan ve insanın fiziksel bir özelliği olarak güvenlik ve tanıma alanında kullanılan parmakizi tanıma üzerinde durulmuştur. Bunun için Yapay Zeka Yöntemlerinden Genetik Algoritma (GA) ve Yapay Sinir Ağları (YSA) yöntemleri araştırılmış ve bunların otomatik parmakizi tanıma performansına etkileri değerlendirilmiştir. Parmakizi tanıma işleminin gerçekleştirilmesi için yapılması gereken genel işlem adımları aşağıda şekilde sıralanabilir:

- Parmakizinin elde edilmesi,
- Elde edilen parmakizi görüntüsünün ön işlenmesi,
- Ön işlenmiş parmakizi görüntüsünden özellik vektörünün elde edilmesi,
- Elde edilen özellik vektörü boyutunun azaltılması,
- Seçilmiş özellik vektörleri kullanılarak parmakizi görüntüsünün tanınmasıdır.

Parmakizi görüntüsünü elde etmek için çeşitli yöntemler mevcuttur. Adli uygulamalarda parmakizi görüntüsü mürekkeplenmiş parmakların on parmak

formuna bastırılmasıyla elde edilir. Olay yerleri parmakizleri genelde gözle görülemeyen görüntüler oldukları için bunların elde edilmesi işlemi çeşitli kimyasal metotlar kullanılarak gerçekleştirilir. Günümüzde ise parmakizi görüntüsünü elde etmek için CCD kameralar veya çeşitli tarayıcı aygıtlar kullanılmaktadır. Belirli bir yöntem geliştirmek ve geliştirilen bu yöntemi diğer yöntemlerle kıyaslamak için hazır parmakizlerine ihtiyaç duyulmaktadır. Bu açıdan, çalışmada genelde yaygın olarak ABD'deki Ulusal Standartlar ve Teknoloji Enstitüsü (National Institute of Standards and Technology (NIST))'nin parmakizi veritabanlarından olan ve literatürde sıklıkla üzerinde araştırma sonuçları bulunan NIST-4 veritabanı kullanılmıştır (NIST 1991, 1994). Bu veritabanında her bir görüntü 512x512 piksel ve 8 bit gri seviyeli olmak üzere toplam 2000 çift parmakizi görüntüsü mevcuttur. Çalışmada bu veritabanına ait 10 kişiden farklı zamanlarda alınmış toplam 1000 adet parmakizi görüntüsü üzerinde araştırmalar yapılmıştır.

Parmakizi tanımada elde edilen parmakizlerinin tam netlikte olmaması bir problem teşkil eder. Parmakizi görüntüsü elde edilirken dış ortamdan veya tarama aygıtından kaynaklanan gürültüler görüntü üzerine biner ve görüntüyü bozar. Bu yüzden parmakizi görüntülerine ait özellik vektörleri elde edilmeden önce ön iyileştirilme işlemi uygulanması gerekir. Bu ön iyileştirme işlemi için çeşitli yöntemler mevcuttur. Bu yöntemlerin başında geleneksel yöntemler olarak ifade edilen histogram eşitleme, filtreleme (uyarlamalı, medyan, ortalama v.b.) veya üzerinde hala çalışmalar yapılmakta olan dönüşüm işlemleri gelir (Greenberg ve ark. 2000). Bunların en bilinen yöntemi ise dalgacık dönüşümüdür, fakat bu dönüşüm parmakizi görüntüleri için yetersiz kalmaktadır. Bu yüzden parmakizi görüntülerine daha uygun olan, parmakizi üzerindeki tüm hatların işlenmesini gerçekleştiren ve dalgacık dönüşümünün geliştirilmişi olan çevritsel dönüşüm (Do ve Vetrelli 2002, 2003) kullanılarak görüntünün daha da iyileştirildiği sonucuna varılmıştır.

Parmakizi tanıma işlemi parmakizi görüntülerinden elde edilecek özellik vektörlerinin karşılaştırılması ile gerçekleştirilir. Bu özellikler genelde parmakizi hatlarındaki hat sonları ve çatallaşma noktaları ve bunların açısal bilgileridir. Bu bilgilerden başka, parmakizi özelliklerini belirlemek üzere yapılan çalışmalar sonucunda parmakizlerinden FingerCode (Jain ve ark. 1999b) adı verilen özel bir

kodlama sistemi geliştirilerek parmakizi özellik vektörleri üretilmiş ve bu kodlama kullanılarak parmakizlerinin tanınması amaçlanmıştır. Bu çalışmada FingerCode kodlama sistemi geliştirilerek üretilen kodlar, YSA metotları kullanılarak eğitilmiş ve tanıma işlemi için kullanılmıştır. Eğitme işlemi, YSA'nın geriye yayılım yöntemlerinden Hızlı Geri Yayılım, Eşleştirmeli Eğim Azaltımlı, Artışlı Geri Yayılım, Yığın Geri Yayılım yöntemleri (Wilson ve ark. 1997a, 1997b) kullanılarak yapılmıştır. Parmakizi özellik vektörlerini YSA ile eğitme aşamasında en iyi başarım oranı %100 ve tüm veri kümesi için ise %99,6 olarak belirlenmiştir. Bu başarımlar dikkate alındığında literatürde belirtilen parmakizi tanıma yöntemleri kadar hatta bazı durumlarda daha verimli bir sonuç elde edildiği, fakat YSA tek başına kullanıldığında eğitim süresinin yeteri kadar azalmadığı da görülmüştür.

Diğer bir problem, YSA'ya giriş olarak verilen ve FingerCode kodlama sistemi ile elde edilen özellik vektörlerinin boyutlarının belirli parametrelerin değişmesiyle artmakta olduğu gözlemlenmiştir. Dolayısı ile büyük boyutlu özellik vektörleri ile YSA'yı eğitme aşaması çok zaman almaktadır. Bu problemleri aşmak, yani YSA eğitim süresini azaltmak ve özellik vektörlerini gürültülü veriden arındırmak amacıyla girişte kullanılan özellik vektörlerinin azaltılması, dolayısı ile özellik seçimi yapılması gerektiği görülmüştür. Bunu yapabilmek için çalışmada çeşitli özellik seçimi yöntemleri araştırılmıştır. Bu yöntemler içerisinde Genetik Algoritmalar (GA)'ın özellik seçme aşamasında diğer yöntemlere göre daha verimli olduğu ve YSA ile eğitme başarısını da artırdığı gözlemlenmiştir. GA kullanılarak azaltılan özellik vektörlerinin YSA'ya giriş olarak verilmesi sonucu YSA, eğitme aşamasında %100 ve tüm veri kümesi için %99,6 başarım göstermiştir. Eğitme süresi ise elde edilen en iyi sonuçlar dikkate alındığında YSA ile tanıma ortalama 54 dakikada gerçekleşirken GA kullanılarak özellik vektörlerinin azaltılma süresi dahil YSA ile tanımada ortalama 10 dakika, hatta bazı durumlarda 5 dakikaya kadar düştüğü gözlemlenmiştir. Sonuç olarak esnek hesaplama yöntemlerinden olan GA ve olasılıksal sinir ağı kullanılarak azaltılmış parmakizi özellik vektörlerinin YSA ile eğitiminde tanıma süresinin 5 kat ile 10 kat arasında düşmesiyle birlikte en iyi doğruluk oranının da elde edildiği gözlemlenmiştir.

1.1. Tezin Organizasyonu

Bu tez çalışması toplam dokuz bölüm ve Ekler kısmından oluşmaktadır:

Birinci bölüm giriş bölümü olup konunun genel tanımı yapılmış, çalışmanın amacı ve önemi üzerinde durulmuştur. Konu ile ilgili önceden yapılmış olan benzer çalışmalar hakkında araştırmalar yapılmış, bilgiler verilmiş ve kısa bir analiz yapılmıştır.

İkinci bölümde, güvenlik alanlarında kullanılan ve kişinin kendine has özellikleri ile tanınabilmesini amaçlayan biyometrik sistemler ile birlikte güvenlik sistemleri ve kriminoloji araştırmalarında kişiye özel olan ve bireysellik gösteren parmakizinin bireyselliği hakkında bilgiler verilmiştir. Parmakizinin karakteristik özellikleri açıklanmıştır. Parmakizinin değişmez ve değiştirilemez, benzemez ve benzetilemez ve tasnif edilebilir özellikleri araştırılmıştır.

Üçüncü bölümde, parmakizi görüntüsünden gürültüleri azaltmak ve görüntüyü iyileştirmek için ön işleme aşamaları ayrıntılı bir şekilde incelenmiştir. Görüntü üzerinde ön işleme teknikleri konusunda bilgiler verilmiştir. Histogram eşitleme, uyarlamalı eşikleme, kuantalama ve kontrast iyileştirme konuları incelenmiştir. Ayrıca geleneksel yöntemlerle birlikte dijital filtreler konusuna yer verilerek dalgacık dönüşümü (wavelet transform) ve dalgacık dönüşümün geliştirilmişi olarak çevritsel dönüşümün (contourlet transform) görüntü ön iyileştirmedeki performansları araştırılmıştır. Burada çevritsel dönüşümün parmakizi hat yapısına uygun bir dönüşüm olduğu, dolayısı ile diğer filtreleme ve dönüşüm tekniklerine göre parmakizi görüntüsünü iyileştirmede daha verimli olduğu tespit edilmiştir.

Dördüncü bölümde, yönlü doku olarak parmakizinin simgelenişi, sınıflandırılması ve tanıma işlemleri açıklanmıştır. Parmakizindeki tekil nokta olan referans noktasının elde edilme aşamaları belirtilmiştir. Parmakizlerini tanıma aşamasında, parmakizindeki referans noktasının tüm aşamalarda aynı yerde belirlenmesi gerektiği belirtilmiştir. Parmakizi kalıpları ve parmakların çizgi desenleri sınıflandırılmış

ve bu kalıplara göre parmakizlerindeki referans noktaların ne tipte olabileceği gösterilmiştir. Referans noktası elde edilen parmakizinin sektörizasyon (mozaikleme) aşamasının işlem adımları sıralanmıştır. Gabor filtre 8 açısal yönde (0^o, 22.5^o, 45^o, 67.5^o, 90^o, 112.5^o, 135^o ve 157.5^o) kullanılarak parmakizi görüntülerine ait yönsel filtreler elde edilmiştir. Bu yönsel filtreler kullanılarak Jain ve ark. (1999b)'nın önerdiği FingerCode adı verilen parmakizine ait özellik vektörü çıkarılmıştır. Elde edilen FingerCode'ların vektör haline getirilmesinde parmakizinin döndürmeden bağımsız olarak özelliklerinin çıkarılarak vektör haline getirilmesi için yeni bir yöntem önerilmiştir.

Beşinci bölümde, parmakizi tanıma için Yapay Sinir Ağları yöntemlerine yer verilmiştir. Görüntü işleme, görüntü sınıflandırma ve tanıma işlemlerinde kullanılan Yapay Sinir Ağları (YSA) hakkında bilgiler verilmiştir. Çalışmada dört farklı YSA modeli uygulanmış ve bu modellerin parmakizi tanımadaki performansları araştırılmıştır. Bu modellere göre parmakizi tanıma aşaması için sonuçlar elde edilmiştir. Kullanılan YSA modelleri aşağıda sıralanmıştır:

1. Hızlı Geri Yayılım (Quick Propagation (QP));
2. Eşleştirmeli Eğim Azaltımlı (Conjugate Gradient Descent (CGD));
3. Artışlı Geri Yayılım (Incremental Backpropagation (IBP));
4. Yığın Geri Yayılım (Batch Backpropagation (BBP)).

Parmakizlerinden elde edilen özellik vektörleri, YSA'da eğitme, test ve doğrulama işlemine tabi tutulmuştur. Sonuç olarak YSA ile eğitme aşamasında %100 başarım ve tüm veri kümesi için %99,6 başarım gözlemlenmiştir.

Altıncı bölümde, parmakizlerinden elde edilen özellik vektörlerinin boyutunun büyük olduğu ve dolayısı ile YSA ile eğitme aşamasında çok fazla zaman harcandığı gözlemlenmiştir. Bundan dolayı özellik vektörlerini azaltmak için veri azaltma, boyut azaltma ve özellik seçimi metotları değerlendirilmiştir. Bu metotlar içerisinde Genetik Algoritmalar, YSA ile birlikte esnek bir yapı oluşturmak üzere geliştirilmesiyle en iyi performansın elde edildiği gözlemlenmiştir. Genetik algoritmalar kullanılarak azaltılan özellik vektörlerinin yine YSA metotları

kullanılarak eğitme işlemine tabi tutulmasından sonra elde edilen sonuçlar, normal sonuçlarla karşılaştırılmıştır. Çalışmada elde edilen sonuçlar göz önünde bulundurularak bu konu üzerinde çalışma yapmak isteyen araştırmacılara bazı önerilerde bulunulmuştur.

Yedinci bölümde, çalışmada geliştirilen yöntemlere ait elde edilen sonuçlar literatürde elde edilen sonuçlar ile kıyaslanmıştır. Ayrıca farklı bilgisayar konfigürasyonları kullanılarak sistem ve zaman performansı karşılaştırılmıştır. Bilgisayar konfigürasyonu arttıkça sistemin çok kısa sürede parmakizi tanıma işlemini gerçekleştirdiği gözlemlenmiştir.

Sekizinci bölüm, tez çalışmasında elde edilen sonuçların tartışıldığı ve bu konu üzerinde çalışma yapmak isteyen araştırmacılara önerilerde bulunulduğu tartışma ve önerilerden oluşmaktadır.

Dokuzuncu bölüm ise bu tez çalışmasında yararlanılan kaynaklar ve referanslardan oluşmaktadır.

Ekler kısmında, parmakizi tanımada elde edilen tüm sonuçların ayrıntılı çizelge ve grafiklerine yer verilmiştir.

1.2. Çalışmanın Amacı ve Önemi

Günlük yaşantımızın her anında kimlik sorgulama ve kişi tanımlama işlemleri ile karşılaşmaktayız. Önceleri insanların kimlik bilgilerini kontrol ederek yapılan doğrulama işlemi, daha sonraları bilgisayarların ve internetin de yaygınlaşması ile yerini şifre ve pin numaralarına bırakmıştır. Kimlik bilgileri, şifre ve pin numaraları gibi tanımlama araçlarının çalınma ve kaybolma ihtimalinin olmasından dolayı, araştırmacılar daha güvenilir sistem arayışı içerisine girmişlerdir. Kişi tanımlama ve doğrulamada kullanılabilecek en iyi sistemin yine kişinin kendisi olabileceğinden yola çıkan araştırmalar, kişileri ayırt edebilecek özelliklere yönelmişlerdir. Kişinin kendine has davranışsal ve/veya fiziksel özellikleri olan ve biyometrik olarak adlandırılan ses, imza, parmakizi, el geometrisi, iris ve hatta DNA'nın kişileri tanımlamada kullanılabileceği düşünülmüştür (Steen 1996).

Bu çalışmanın temel amacı, parmakizlerini kullanarak kişilerin doğrulanmasını sağlamak için güvenli yöntem veya yöntemler geliştirmektir. Bu nedenle mevcut çalışmalar da dikkate alınarak parmakizi tanımada daha güvenilir ve hata oranı daha düşük bir sistem tasarımı amaçlanmış ve parmakizlerine ait FingerCode adı verilen özellik vektörleri, Yapay Sinir Ağları (YSA) metotları kullanılarak eğitilmiştir. Daha sonra yine özellik vektörleri kullanılarak sistem test edilmiştir. Eğitim ve test sonuçları diğer çalışmalarda elde edilmiş sonuçlarla karşılaştırıldığında düşük hata ve yüksek bir doğrulamanın elde edildiği görülmüştür. Fakat parmakizlerinden elde edilen özellik vektörlerinin büyük boyutlu olmasından dolayı YSA'daki eğitim ve doğrulamanın uzun sürdüğü gözlemlenmiştir. Çalışmanın bu aşamasında sistem performansından kaybetmeden YSA eğitim süresinin azaltılması gerektiği görülmüştür. Bunun için veri kümelerinde boyut azaltımı ve özellik seçimi gibi metotlar araştırılmıştır. Bu metotlar içerisinde GA'nın özellik vektörleri içerisindeki gürültülü verileri arındırarak özellik vektörü boyutlarını azalttığı, böylece uygun bir özellik seçimi işlemi gerçekleştirdiği görülmüştür.

Bu tez çalışmasında temel olarak aşağıdakiler amaçlanmıştır:

- Parmakizlerinin bir tarayıcıdan elde edilmesi ve/veya literatürde kullanılan bir parmakizi veritabanının elde edilmesi,

- Elde edilen parmakizlerinin ön işleme yöntemleri kullanılarak gürültüden arındırılması ve ön iyileştirilmesi,

- Ön işlenmiş parmakizlerine ait tekil nokta olan referans noktalarının bulunması,

- Bulunan referans noktası etrafında ilgili bölgenin belirlenmesi,

- İlgili bölgenin sektörize edilmesi (mozaikleme işlemi),

- Parmakizinin döndürmeden bağımsız olarak parçalara ayrılan her bir sektörüne ait özelliklerin çıkarılması ve tüm sektörlerden elde edilen özellikler kullanılarak parmakizine ait özellik vektörünün elde edilmesi,

- Parmakizlerini tanıma aşaması için elde edilen özellik vektörlerinin YSA metotları ile eğitilmesi,

- Parmakizi tanıma işlemlerine ait eğitim, test, doğrulama ve tüm veri kümeleri için doğruluk ve hata oranlarının tespiti,

- Özellik vektörleri boyutunun GA kullanılarak azaltılması sonucu elde edilen seçilmiş özellik vektörleri kullanılarak YSA metotları ile parmakizlerinin tanıma işlemi için yeniden eğitilmesi ve veri kümelerine ait doğruluk ve hata oranlarının tespitidir.

Sonuç olarak kişi doğrulamada kullanılacak olan parmakizlerinin tanıma işleminden geçirilerek elde edilen doğruluk ve hata oranları dikkate alındığında gerçekleştirilen sistemin daha güvenilir ve daha düşük hata oranına sahip olduğu görülmektedir. Ayrıca, kullanılan parmakizi sayısı dikkate alındığında, 1000 adet parmakizinin YSA'da eğitilmesi sonucu yüksek doğruluk değerlerine ulaşılması açısından da önemlidir.

1.3. Kaynak Araştırması

Po ve Do (2003), çevritsel dönüşüm kullanarak görüntülerin yönsel çok ölçekli modellemesini gerçekleştirmişlerdir. Dalgacık dönüşüm ve çevritsel dönüşümün görüntü üzerindeki değişimlerini kıyaslamışlardır. Sisteme Gizli Markov Ağacı (GMA) (Senior 1997) yapısının da eklenmesiyle performansın arttığını göstermişlerdir. Çevritsel dönüşümlü GMA ve dalgacık dönüşümlü GMA'yı görüntüden gürültüleri uzaklaştırmak için kullanarak elde ettikleri sonuçları diğer klasik yöntemlerle sinyal-gürültü oranına göre kıyaslamışlardır. Uygulamayı parmakizi haricinde literatürde bilinen görüntüler üzerinde kullanmışlardır.

Eslami ve Radha (2003), düşük bit-oranlı görüntü kodlama için çevritsel dönüşümü kullanmışlardır. Çevritsel dönüşüm alanında bir görüntünün anlamlı haritalanması olarak kuantalanmış dönüşüm katsayıları üretmişlerdir. Çevritsel dönüşümün dalgacık dönüşümünden üstün olduğunu göstermişlerdir. Uygulamayı parmakizi haricinde literatürde bilinen bazı görüntüler üzerinde kullanmışlardır.

Do ve Vetterli (2002), çevritsel olarak tanımlanan yeni bir esnek, çok ölçekli, bölgesel ve yönsel görüntü simgeleniş modeli önermişlerdir. Çevritsel dönüşümün, görüntüye benzeyen iki boyutlu parça parça düzleştirilmiş işaretler için seyrek bir simgeleniş ortaya çıkardığını göstermişlerdir. Uygulamayı parmakizi haricinde literatürde bilinen bazı görüntüler üzerinde kullanmışlardır.

Greenberg ve ark. (2000), parmakizi sınıflandırma ve tanıma işleminin önemli bir aşaması olarak parmakizi görüntüsünden özellik çıkarımı için görüntünün ön işlenmesi gerektiğini belirtmişlerdir. Bunun için iki metot önermişlerdir. Birincisi, lokal histogram eşitleme, uyarlamalı (wiener) filtreleme ve görüntünün ikileştirilmesidir. İkinci metot ise, Gabor filtre tabanlı doğrudan gri ölçekli görüntü iyileştirmedir. NIST veritabanından alınan 10 parmakizi için gerçekleştirilen uygulamada yeniden düzenlenmiş Gabor filtrenin diğerlerine göre daha iyi ön iyileştirme gerçekleştirdiği gösterilmiştir. Uygulanan yöntemler haricinde diğer

güçlü görüntü iyileştirme yöntemleri uygulanmamıştır. Günümüzde klasik yöntemler olarak adlandırılan yöntemler kullanılmıştır.

Zhang ve ark. (2002), otomatik parmakizi tanımlama sistemlerinin performansının kaliteli parmakizi görüntüsü ile arttığını belirterek parmakizi görüntülerinin ön iyileştirilmesinin bu sistemler için önemli olduğu vurgulanmıştır. Bu çalışmada dalgacık dönüşüm ile doku filtreleme birleştirilerek daha etkili bir iyileştirme gerçekleştirilmiştir. Ayrıntı çıkarımı üzerine yapılan çalışmada dalgacık dönüşümün parmakizi görüntü iyileştirme için kullanılabileceği belirtilmiştir, fakat çevritsel dönüşüm gibi diğer dönüşüm teknikleri ile karşılaştırma yapılmamıştır.

Do ve Vetterli (2003), görüntüler için yeni iki boyutlu bir simgeleniş önermişlerdir. Piramitsel yönlü filtre kümesi olarak adlandırılan çift filtre kümesi ile yönlü filtre kümeli Laplas piramidinin birleştirilmesinden oluşan bu sistemi çevritsel dönüşüm olarak adlandırmışlardır. Çevritsel dönüşümün eğrisel tabanlı görüntüler için yüksek ön işleme performansı gösterdiğini belirterek parmakizi haricinde çeşitli görüntüler üzerinde yapmış oldukları uygulamaların sonuçlarında bunu göstermişlerdir.

Eslami ve Radha (2005), dönüşümle değişmeyen çevritsel dönüşüm tasarlamışlardır. Bu yöntemin ön işleme için uygulandığı görüntülerde daha yüksek sinyal-gürültü oranı elde ettiklerini belirtmişlerdir. Görüntüdeki gürültüleri azaltmada dönüşümle değişmeyen çevritsel dönüşümün dalgacık dönüşümüne göre daha etkili olduğunu parmakizi haricinde çeşitli görüntüler üzerinde yapmış oldukları uygulama sonuçları ile göstermişlerdir.

Jain ve ark. (1999c), parmakizlerini sınıflandırma ve tanıma işlemi için filtre tabanlı bir yaklaşım öne sürmüşlerdir. Buna göre ön işlenmiş parmakizi görüntüsü, Gabor filtre kümesinden geçirilerek her bir filtre kümesi için FingerCode adı verilen özellik vektörleri çıkartılmıştır. Özellik vektörleri çıkarımında sadece 5 bant ve 16 sektör kullanılmıştır. Bant ve sektör sayısının değişmesiyle performansın nasıl etkileneceği ile ilgili bir çalışma yapılmamıştır. Elde edilen özellik vektörleri k-NN YSA sınıflandırıcısında iki aşamalı olarak sınıflandırılmıştır. NIST-4 veritabanına ait

toplam 800 parmakizinin kullanıldığı uygulamada %90 sınıflandırma başarısına ulaşılmıştır.

Sha ve ark. (2003), FingerCode'un parmakizinin sınırlı ve ayrıntılı bilgilerini göstermede etkili olduğunu belirtmişlerdir. FingerCode'un performansını artırmak için referans noktasının doğru elde edilmesi gerektiğini söylemişlerdir. Fingercode özellik vektörü elde edilirken bant sayısı 5 ve 7 olarak seçilmiştir. Bant sayısı ve sektör sayısının değişmesiyle performansın nasıl etkileneceği ile ilgili bir çalışma yapılmamıştır. NIST-4 veritabanını kullanarak elde edilen FingerCode özellik vektörünün parmakizi eşleştirmedeki performansını işlem karakteristiği eğrisi (receiver operating characteristics (ROC)) üzerinde göstermişlerdir.

Jain ve ark. (1999b), Jain ve ark (2000), 640 byte boyutunda FingerCode olarak adlandırılan parmakizindeki sınırlı ve ayrıntılı detayları elde etmek için Gabor filtre kullanan bir filtre-tabanlı algoritma tasarlamışlardır. Ön işlenmiş parmakizi görüntüleri Gabor filtre kümesinden geçirilerek her bir filtre kümesi için FingerCode adı verilen özellik vektörleri elde edilmiştir. Özellik vektörleri çıkarımında sadece 5 bant ve 16 sektör kullanılmıştır. Bant ve sektör sayısının değişmesiyle performansın nasıl etkileneceği ile ilgili bir çalışma yapılmamıştır. Uygun FingerCode'lar arasındaki Euclidean mesafesi bulunarak parmakizi eşleştirme gerçekleştirilmiştir. 25 farklı parmaktan toplam 250 parmakizi örneği kullanarak %1 hata kabul oranına karşılık %12 hata ret oranı elde etmişlerdir.

Prabhakar (2001), parmakizlerinin yeni filtre tabanlı bir simgelenişini göstermiştir. Bu simgelenişi parmakizi tanımada ve sınıflandırmada kullanmıştır. Deneysel sonuçları güncel ayrıntı tabanlı eşleştirmelerle kıyaslamıştır. Özellik vektörlerinin elde edilmesinde 5 bant ve 16 sektör kullanılmış ve dolayısı ile 640 byte'lık bir özellik vektörü elde edilmiştir. Euclidean mesafesi kullanılarak tanıma işlemi yapılan çalışmada YSA ile ilgili bir çalışma yapılmamıştır. Euclidean mesafesi ile yapılan tanıma işleminde özellik vektörleri her işlem başlangıcında döndürülerek vektör elemanları arası mesafeler elde edilmektedir. Bu da işlem süresinin uzamasına sebep olmaktadır.

Jain ve ark. (1999d), Gabor filtre kümesi ile parmakizinin merkezi bölümünün filtrelenerek dört yöndeki (0^o, 45^o, 90^o ve 135^o) hat sayılarını ayıran algoritmayı önermişlerdir. Bu algoritmadan elde ettikleri bilgiyi Fingercode özellik vektörüne dönüştürmüşlerdir. Fingercode vektörünü kullanarak parmakizlerini sarmal, sağ kement, sol kement, kemer ve çadırsal kemer olmak üzere beş sınıfa ayırmışlardır. İki aşamalı bir sınıflandırıcı kullanmışlardır. Birinci aşamada k-en yakın komşuluk sınıflandırıcısı ve ikinci aşamada ise bir YSA metodunu kullanmışlardır. Beş sınıf için %90 başarı elde etmişlerdir. Kemer ve çadırsal kemer tek bir sınıfa birleştirilip dört sınıf üzerinden sınıflandırma işlemi uygulanınca başarı oranı %94.8'e kadar çıkmıştır.

Khmanee ve Nguyen (2004), yönsel bant geçiren Gabor filtre kümesinin parmakizi görüntüsünü iyileştirmek için çok etkili bir teknik olduğunu belirtmişlerdir. Filtre çıkışının, hat yönelimine ve Gaussian dağılımının uzaysal dağılımına çok duyarlı olduğu gösterilmiştir. Ayrıntı çıkarımı tabanlı sistemler için geliştirilen bu uygulamada özellik vektörleri kullanılmamıştır.

Hatami ve ark. (2005), otomatik parmakizi tanımlama sistemleri için sınırlı ayrıntıların kalitesini artırmayı amaçlamışlardır. Önerdikleri algoritma iki aşamalı işlemden oluşmaktadır; düzleştirme ve Gabor dalgacık filtrelemesi. İyileştirme algoritmasının düzleştirme kısmı, Gaussian filtreleme tabanlı yeni bir teknik ile gürültülerin azaltılmasını sağlamaktadır. Gabor dalgacık ise düzleştirilen görüntünün kalitesini artırmak için kullanılmıştır. 256x256 boyutlu 50 parmakizi görüntüsü için uygulanan bu algoritma diğer benzer çalışmalarla karşılaştırılarak daha performanslı olduğu gösterilmiştir. Dalgacık dönüşümün kullanıldığı uygulamada çevritsel dönüşüm ile karşılaştırma yapılmamıştır.

Hong ve Jian-Hua (2003), parmakizi tanıma işlemi için uygun Gabor filtre tasarlamışlardır. Adaptif Sinirsel-Bulanık Çıkarım Ağı (ANFIS) kullanarak eşleştirme algoritması önermişlerdir. 20 parmaktan alınan toplam 80 parmakizine ait hatalı kabul oranı ve hatalı ret oranını deneysel olarak bulmuşlardır.

Horton ve ark. (2002), parmakizi eşleştirme sistemi için 2-boyutlu Gabor filtreyi önermişlerdir. 2-boyutlu Gabor filtre aynı zamanda özellik çıkarım işlemini

de gerçekleştirmektedir. Bu filtre kullanılarak ön işlenmiş parmakizi görüntüsünden çeşitli açılarda elde edilen özellik vektörleri FingerCode adı verilen özellik vektörleri toplamını oluşturmaktadır. Özellik vektörleri çıkarımında sadece 5 bant ve 16 sektör kullanılmıştır. Bant ve sektör sayısının değişmesiyle performansın nasıl etkileneceği ile ilgili bir çalışma yapılmamıştır. Elde edilen FingerCode'lar arasında Euclidean mesafesi bulunarak eşleştirme işlemi gerçekleştirilmektedir. 500 örnek parmakizi üzerinde yapılan araştırmada hatalı kabul oranı %2,49 ve hatalı ret oranı ise %5,3 olarak elde edilmiştir.

Jiang (2001), parmakizi tanıma için parmakizi iyileştirmenin önemli aşamalardan birisi olduğunu vurgulayarak Gabor filtrenin sahte hat bilgilerini ortaya çıkardığını söylemiştir. Bu ise özellik çıkarımı, dolayısı ile otomatik parmakizi tanıma için istenmeyen bir durumdur. Bu çalışmada parmakizi görüntüsü iyileştirmede uygulanan Gabor filtrenin özellikleri araştırılmıştır. Sinüzoidal olmayan şekilli hat yapısı, hat frekans yaklaşım hatası ve küçük filtre boyutu, sahte hat yapılarının oluşmasına neden olmaktadır. Çözüm olarak otomatik parmakizi tanımada istenmeyen yan etkilerden kurtulmak için Gabor filtre yerine uyarlanır, yönlü alçak geçiren filtreyi önermiştir. Parmakizi görüntüsünde sahte hat yapılarının oluşmaması için görüntü üzerinde bir ön iyileştirme işlemi uygulanmamıştır.

Leung ve ark. (1990), geri yayılım ağları ve Gabor filtreleri kullanarak parmakizi görüntüsündeki özellik çıkarımı için bir sistem tasarlamışlardır. Parmakizi görüntüsünde sahte hat yapılarının oluşmaması için görüntü üzerinde bir ön iyileştirme işlemi uygulanmamıştır. Parmakizi görüntüleri önce karmaşık Gabor filtre ile evriştirilmiş, daha sonra ayrıntı bölgelerini tanımak için istenen evre ve işaretler ağda eğitilmiştir. Yüksek belirleme oranı ve düşük hata oranı elde edilmiştir. Dokusal görüntü işlemede Gabor filtrenin ve görüntü ayrıntı çıkarımında YSA'nın etkili olduğu gösterilmiştir.

Liu ve Li (2004), Gabor filtre uygun bağlantı çözünürlüğü sunmasından dolayı düşük kalitedeki parmakizi görüntülerinin gürültülerini filtrelemede kullanmışlardır. Parmakizi görüntülerinin yön bilgilerinden hat yapısını elde edilen bir hat seçme algoritması önermişlerdir. Bu yön bilgisinin kullanımıyla görüntü iyileştirme işlemini gerçekleştirmişlerdir. Karu ve Jain (1996)'in sınıflandırma

modelini kullanarak parmakizindeki merkez ve delta noktalarını sınıflandırmışlardır. 500 parmakizi örneği için sınıflandırma doğruluk oranı %97 olarak belirlenmiştir.

Wei ve ark. (2004), yeni bir parmakizi iyileştirme algoritması önermişlerdir. İki aşamalı olarak parmakizi görüntülerini iyileştirmek için Gabor filtre ve yönsel filtre kullanmışlardır. Birinci adımda birbirine yapışan hatları ayırmak için dinamik Gabor filtresini kullanmışlardır. Daha sonra kopan hatları birleştirmek için ve vadileri düzeltmek için yönsel filtreyi kullanmışlardır. Bu metot, parmakizi görüntüsünün döndürülmesine duyarlı değildir.

Kim ve ark. (2002), parmakizi görüntüsünü iyileştirmek için, görüntü normalizasyonu ve Gabor filtre kullanmışlardır. Parmakizi görüntülerini iyileştirmek için blok işleme tabanlı etkili bir normalizasyon önermişlerdir. Ayrıca Gabor filtrenin hat yönü ve hat frekansı olan iki önemli parametresinin seçimi için yeni bir teknik öne sürmüşlerdir. Parmakizi görüntüsünde sahte hat yapılarının oluşmaması için görüntü üzerinde bir ön iyileştirme işlemi uygulanmamıştır.

Saatci ve Tavsanoglu (2002), dış etkenlerden, yaralanmalardan veya parmakizini elde ederken mürekkebin hatları birleştirmesinden dolayı parmakizlerinin kötü kalitede olduğunu belirtmişlerdir. Parmakizi tanımada bu etkenler hatalara sebep olmakta, bundan dolayı doğru bir hat deseni ile parmakizi karşılaştırmanın temelinde parmakizi görüntüsünün iyileştirilmesi gelmektedir. Hat ayrıntılarını her bir noktada eşleştiren doğru filtre parametreli yönsel bir filtre tasarımıyla etkili bir parmakizi hat iyileştirilmesi yapılabilmektedir. Bunun için CNN-Gabor filtresi tasarlanmıştır. Parmakizi görüntüsünde sahte hat yapılarının oluşmaması için görüntü üzerinde bir ön iyileştirme işlemi uygulanmamıştır.

Jain ve ark. (2000), biyometrik sistemlerin personel tanımlamada güvenilir sistemler olduklarını belirterek elektronik ticaret, kredi kartı güvenliği, smart kart güvenliği ve oylama sistemlerinde biyometrik sistemlerinin kullanılabilirliğini vurgulamıştır. Biyometrik sistemlerin, evrensellik, doğruluk, performans, maliyet, güvenlik ve özellikleri bakımından karşılaştırmasını yapmıştır. Bu sistemler hakkında bilgi vermiş, parmakizi ile tanımlama sisteminin doğrulama ve tanımlama

işleminde nasıl kullanıldığını anlatmıştır. Sonuç olarak biyometrik sistemlerin şifre, PIN gibi geleneksel tanımlama metotlarının yerini alacağını belirtmiştir.

Vargas (2001) ve Prabhakar (2001), biyometrik sistemlerin kayıt, doğrulama ve tanımlama işlemlerinde nasıl kullanıldığını anlatmıştır. Biyometrik sistemlerin ticari alanda bilgisayar ağlarına girişte, elektronik veri güvenliğinde, elektronik ticarette, internete erişimde, ATM'lerde, kredi kartlarında, hücresel telefonlarda, fiziksel erişim kontrollerinde, tıbbi kayıt yönetimlerinde ve uzaktan eğitimde; devlet ile ilgili, ulusal kimlik kartı, doğrulama aracı, ehliyet, pasaport, güvenilir harcama olarak; ve adli işlerde, ceset tanımlamada, suçlu araştırmada, terörist belirlemede ve kayıp çocuklarda kullanılabilirliğini açıklamıştır. Sonuç olarak güvenlik ve gizliliği yüksek olan sistemlerde biyometrik sistemlerin iş dünyasında etkileşimli olarak kullanılabileceğini belirtmiştir.

Ratha ve Bolle (2003), birçok biyometrik tanıma ve doğrulama sistemlerinin olduğundan bahsederek terör tehditlerinden dolayı bu tür sistemlere ihtiyacın arttığını belirtmiştir. Biyometrik tanıma ve doğrulamanın geleneksel yöntemlere olan üstünlüğünü ele almıştır. Parmakizinin özellikleri ve personel tanıma işleminde nasıl kullanıldığı hakkında bilgi vermiştir.

DigitalPersona (2003)'e göre, yaşantımızda PC, dizüstü bilgisayar, ATM makineleri, hücresel telefonlar, fiziksel erişim ürünleri gibi birçok elektronik aygıtın olmasından dolayı güvenlik ve kullanılabilirlik ihtiyacı ile personel tanıma gerekli hale gelmiştir. Smart kart, PIN numaraları ve şifreler gibi güvenlik aygıtlarının olumsuzluklarından bahsederek hileli tanımlamaları önlemek, hızlı ve çabuk bir şekilde personel tanımanın gerekli olduğunu belirtmiştir. Bundan dolayı insan karakteristiklerini ve ayırt edici özelliklerini kullanarak tanımlama işleminin yapılması gerektiğini vurgulamıştır. Bilgisayar endüstrisinin kısa zamanda gelişmesine paralel olarak biyometrik teknolojilerin de zamanla önemli bir teknoloji olacağını belirtmiştir.

Bjorn (2002), kullanıcı doğrulama için kullanılan verilerin güvenilir şifreleme yöntemleri ile kripto edilmesine rağmen genelde basit şifreler verilmesi dolayısı ile bu tür şifrelerin kolaylıkla kırılabileceğini belirtmiştir. Şifre kullanım süresinin

dolması, karmaşık şifre belirleme kuralları ve güvenlik gerektiren ortamların fazlalığıyla paralel olarak birden çok şifrenin ezberlenmesi yüzünden birçok kullanıcı tarafından şifrelerin unutulduğunu belirtmiştir. Artan bu tür problemlere ideal bir cevap olarak biyometrik sistemlerin kullanılabilirliğini anlatmıştır. Parmakizi ile sistem güvenliği ve kullanıcı onaylama için gerekli sistem adımlarını açıklamıştır.

Bowman (2000), biyometrik doğrulama sistemlerinin ilk yıllardan birkaç yıl öncesine kadar çok pahalı sistemler olduğunu, bu yüzden uygulama imkanının kısıtlı olmasına rağmen günümüzde çok ucuz fiyatlara biyometrik sistemlerin kurulup uygulamaya geçilebileceğini belirtmiştir. İnsanın fiziksel ve davranışsal karakteristikleri kullanılarak tanımlama ve doğrulama işlemlerinin gerçekleştirilebileceğini vurgulamıştır. Biyometrik sistemler ve çalışma prensipleri hakkında ve bu sistemlerin maliyetleri konusunda bilgiler vermiştir.

Dugelay ve ark. (2002), biyometrik sistemlerin sinyal işlemenin önemli konularından olduğunu, ses, görüntü gibi teknolojilerin oldukça fazla çalışılmasına rağmen doğrulama sistemlerinin tamamlayıcı parçaları olduğunu belirtmiştir. Ses, parmakizi ve yüz doğrulama algoritmaları hakkında bilgiler vermiştir. Gelecekte biyometriklerin ve smart kartların erişim kontrol güvenliklerinde ve kişiselleştirilmiş elektronik ticaret servislerinde güvenliği artırmaya yönelik anahtar roller oynayacaklarını vurgulamıştır.

BenAbdelkader ve ark. (2002), biyometrik sistem çeşitlerinden olan ve insanların yürümesi göz önüne alınarak onların yürümesinden tanınmasını amaç edinmiştir. İki farklı yürüyüş tanıma; yürüyüş adımlarının özellik sınıflandırılması yapılarak parametrik olmayan yaklaşım ve adım uzunluğu gibi parametrelerin sınıflandırılması yapılarak parametrik yaklaşım ile tanıma gerçekleştirilmiştir. İnsan yürüyüşü ile tanıma işleminin %85 başarı sağladığı belirtilmiştir.

Liu ve Silverman (2001), ses tanıma gibi biyometrik sistemlerin tarihin en eski dönemlerinde bile bir tanımlayıcı unsur olarak kullanıldığını, günümüzde ise araştırmalar sonucu kişileri tanımlamada daha fazla karakteristiklerin olduğunun tespit edildiğini belirtmiştir. Bu tespit edilen karakteristikler arasında insanların parmakizi yapısı, DNA yapısı gibi özellikleri sıralanabilmektedir. Bütün bunlara

rağmen kanuni uygulamalarda ve adli durumlarda en çok kullanılan pozitif tanımlayıcının parmakizi yapısı olduğunu vurgulamıştır. Bu, aynı zamanda güvenilir sistem olmasının yanında genel olarak kullanılması en kolay olan sistemdir. Dolayısı ile gelişmiş parmakizi biyometrik sistemlerin personel takibinde, elektronik ticarette ve özellikle mobil elektronik ticarette finansal işlem sistemlerinin de entegrasyonu ile daha güvenilir ve özel bir şekilde kullanılabileceğini belirtmiştir.

Blomme'ye (2003) göre, normalde kullanıcı kimliğinin doğruluğu PIN kodları veya ID kartlarla yapılmaktadır. Bireysel vücut özelliklerini tanımlayan biyometrik sistemler ise kod veya kartlara alternatif yaklaşımlardır. Günümüzde parmakizi tarama en çok kullanılan biyometrik tanımlama metodudur. Kullanımının basit ve tanımlamanın hızlı olmasından dolayı retina tarama, imza doğrulama gibi diğer biyometrik tanımlama metotlarından popülerdir. Burada biyometrik tanıma sistemlerinden parmakizi tanıma incelenmiş, gerçek parmakizinin kopyasının kullanılarak çeşitli sistemler üzerine giriş yapılmaya çalışılmış ve bunların yapay parmakizini kabul edip etmedikleri araştırılmıştır.

Lawson'a (2003) göre, biyometrik sistemler güvenlik gerektiren ve özel alanlar için kullanışlı platformlardır. Sağlık hizmetlerinde hastaların bilgilerini genel bir veri tabanında tutmaları, hastalıkların sürekli kontrolü açısından önemlidir. Bu bilgilere erişmek için hastalar, parmakizi özelliklerini kullanarak kendi bilgilerine güvenilir bir şekilde erişebilirler.

Walner'a (2002) göre, manyetik şeritli kredi kartları ile biyometrik tanımlamanın birleşimi, artan hile olaylarını kontrol etmek ve gelişen doğrulama alanındaki işlem kapasitesini artırmak için uygundur. Çalınan, kaybolan, adres değişikliğinden işlemi durdurulan, değiştirilen, taklit edilen ve yanlış kullanılan kredi kartlarından dolayı milyonlarca dolar zarara uğrayan bu sektörde parmakizi tanıma yoluyla gerçekleştirilebilecek bir sistem ile güvenilir ve emin bir şekilde kredi kartı kullanımı gerçekleştirilir. Online veritabanı ağına bağlı bir sistemle, kredi kartı bilgileri kontrolü yapıldıktan sonra kontrolü yapılan kredi kartı sahibinin kendine ait karakteristik özellikleri kullanılarak aynı sistem tarafından kontrol edilmesi yukarıda sayılan olumsuzlukların önüne geçilir. Dolayısı ile ekonomideki zarar minimuma indirgenecektir.

Bormann'a (2005) göre çevrimiçi biyometrik doğrulama servisi, web üzerinden işlemlere veya herhangi bir veriye erişmek isteyen çevrimiçi kullanıcıları tanımlamak için tasarlanır. Şirketler büyüdükçe, bunlara üye olan çevrimiçi kullanıcılar da artar. Dolayısı ile bunlar web üzerinden çevrimiçi parmakizi doğrulama sistemi ile güvenilir ve hızlı bir şekilde sisteme giriş yapabilirler. Visa, Verisign gibi çevrimiçi para transferi gerçekleştiren şirketlerle bağlantılı çalışan şirketler ise web üzerinden parmakizi doğrulama sistemi sayesinde doğru kişilerin sisteme giriş yapıp istenen işlemleri gerçekleştirmesini sağlarlar.

Ham ve Atkinson'a (2002) göre, biyometrik sistemler hayatımızın güvenlik ve tanımlama gerektiren her alanında kullanılabilir hale gelmiştir. Bunlardan birisi de sürücülerin ehliyetleri yerine parmakizi bilgilerini kullanmalarıdır. Burada sürücülerin bilgileri bir sürücüler veritabanında tutulur. Sürücülere verilecek bir akıllı karta sürücünün parmakizi bilgisi gömülür. Sürücü ihtiyaç olduğu durumlarda sisteme önce akıllı kart daha sonra da parmakizi bilgisi ile giriş yapar. Sistem tarafından okunan akıllı karttaki parmakizi bilgisi veritabanındaki bilgi ile karşılaştırılır. Sonuç olumlu ise sürücü için gerekli işlem gerçekleştirilir. Böylece sistem hileli işlemlerden arındırılmış olur.

Adams'a (2000) göre, akıllı kart çipleri biyometrik şablonları depolamak ve eşleştirme yapmak amacıyla kullanımı her geçen gün artmaktadır. Yakın gelecekte güvenlik seviyesi, akıllı kartlar üzerine gömülen parmakizi bilgileri sayesinde artacaktır. Böylece akıllı kart ve biyometrik sistemlerin birleştirilmesi ile güvenilir ve kolay personel tanıma işlemi gerçekleşir.

Günlük yaşantımızın her anındaki olaylar ya da uygulamalar için güvenilir bir şekilde izin ve kimlik denetimi zorunlu hale gelmektedir. Son zamanlarda kişi tanımlama, vazgeçilmez unsurlardan olmaktadır. Biyometrik tanımlama ya da kısa adıyla biyometrik, kişilerin kendilerine has fizyolojik özelliklerini kullanarak tanıma ve doğrulama uygulamalarında kullanılan genel bir tekniktir. Bunlar içerisinde parmakizi eşleştirme, iris tanıma, retina tarama, ses ve konuşma tanıma, yüz tanıma ve el tanıma en bilinen sistemler olarak sıralanabilir. Bu çalışmada biyometrik sistemlerden olan ve insanın fiziksel bir özelliği olarak güvenlik ve tanıma alanında kullanılan parmakizi tanıma üzerinde durulmuştur. Parmakizi tanıma işlem adımları

genel olarak parmakizinin elde edilmesi, elde edilen parmakizi görüntülerinin ön işlenmesi ve ön işlenmiş parmakizi görüntülerinin tanınması şeklinde söylenebilir.

Literatürde parmakizi tanıma aşamasında elde edilen parmakizlerinin tam netlikte olmaması bir problem olarak görülmektedir. Bu yüzden parmakizi görüntülerine ait özellik vektörleri elde edilmeden önce ön iyileştirilmesi zorunludur. Parmakizi görüntülerinin ön iyileştirilmesi için literatürde kullanılan çeşitli yöntemler mevcuttur.

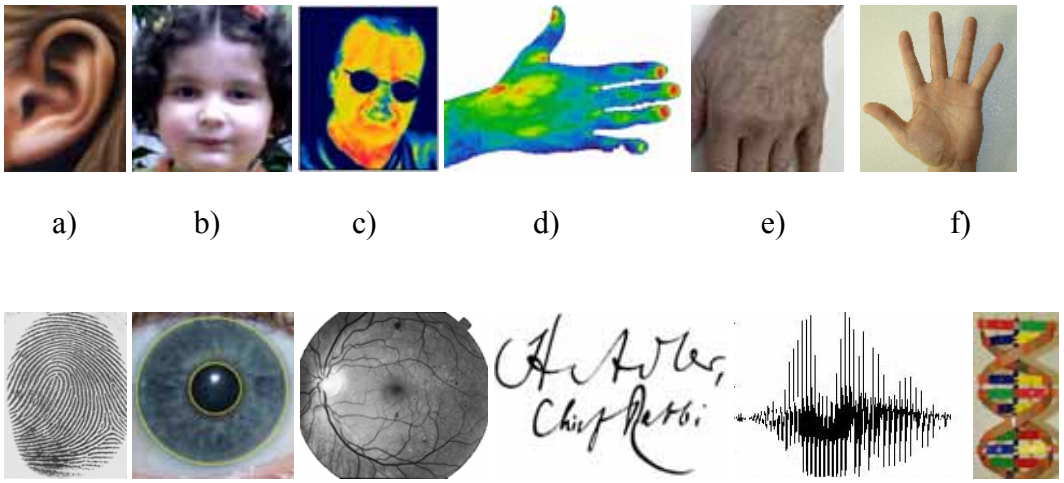
Parmakizi tanıma işlemi parmakizi görüntülerinden elde edilecek özellik vektörleri kullanılarak gerçekleştirilir. Daha önce yapılan çalışmalar için parmakizi özelliklerini belirlemek üzere hat yapısındaki hat sonları ve çatallaşmalar ile bunların açısal bilgileri ele alınmakla birlikte parmakizini simgeleyici FingerCode adı verilen özel bir kodlama sistemi de kullanılmıştır. Yapılan çalışmalarda kullanılan parmakizi sayısı çeşitli olmakla birlikte, parmakizlerinin elde edildiği kişi sayısı da genellikle az olarak görülmektedir.

Tanıma veya sınıflandırma yöntemleri kullanılarak elde edilen parmakizi tanıma veya sınıflandırma %50'ler oranında olabildiği gibi %100 tanıma veya sınıflandırma oranının elde edildiği çalışmalar da vardır. Bu oranlar elde edilirken sistemin bütününün göz önünde bulundurulması gerekir. Bazı çalışmalarda parmakizi ön iyileştirme yapılmaması tanıma ve sınıflandırma aşamasını da etkilemiştir. Buna benzer olarak bazı çalışmalarda parmakizi özelliklerinin belirlenmesi aşamasında seçilen yöntemin de başarısız oluşu sonucunda parmakizi tanıma veya sınıflandırma oranının %50 olarak elde edildiği görülmüştür.

2. BİYOMETRİK SİSTEMLER ve PARMAKİZİ

Biyometrik, kişinin fiziksel ve davranışsal özelliklerini tanıyarak kimlik saptamak üzere geliştirilmiş bilgisayar kontrollü, otomatik sistemler için kullanılan genel bir terimdir. Biyometrik sistemler temelde, kişinin sadece kendisinin sahip olduğu, diğer kişilerden ayırt edici özelliğiyle kendisi olduğunu kanıtlamaya yarayan, değiştiremediği, fiziksel veya davranışsal bir özelliğinin tanınması prensipleri ile çalışan sistemlere denir (Koçsistem 2003).

Araştırmacıların üzerinde çalışma yaptıkları ve halen de üzerinde araştırma yapılan biyometrik sistemler genel olarak Şekil 2.1'de gösterilmiştir.



a) b) c) d) e) f)

g) h) i) j) k) l)

Şekil 2.1 Çeşitli biyometrikler:

a) kulak, b) yüz, c) yüz termogramı, d) el termogramı,
e) el damar geometrisi, f) el geometrisi, g) parmakizi,
h) iris, i) retina, j) imza, k) ses, l) DNA

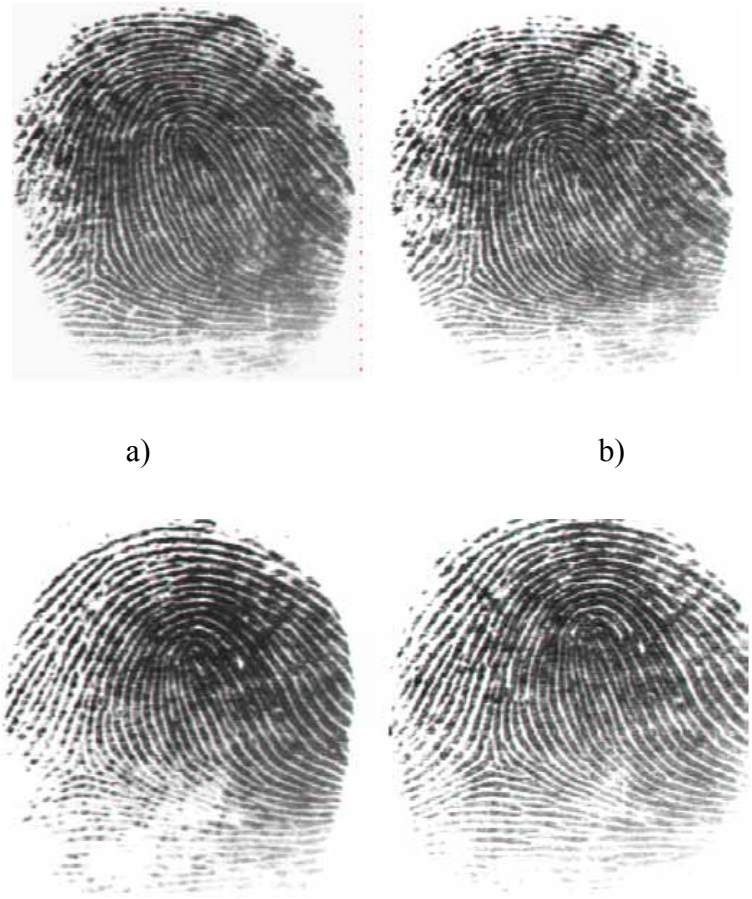
Biyometrik tanıma sistemleri için öncelikle bir görüntü kaydedilir. Kaydedilen bu görüntü sayısal koda çevrilir. Bu kod da gerekirse yapılan işleme göre şifrelenir ve bilgisayara kaydedilir. Daha sonra kullanıcı herhangi bir cihaz kullanarak kendini sisteme tanıtır. O anda tanıttığı kod ile kayıtlı olan kodun birebir tutma olasılığı yoktur. Bunda birçok faktör etkili olabilir. Görüntü alınan ortamın ışıklandırması, bakış açısı, parmağı veya eli sisteme tanıtma açısı, cihazın veya parmağın kirliliği veya nem gibi etkenlerden dolayı birebir aynı kod hiçbir zaman üretilemez. Dolayısıyla iki kod yan yana konulur. Belli bir yüzde tutuncaya kadar karşılaştırılır. Sistemin güvenilirliği için düzenlenen algoritmaya bağlı olarak istenilen oran yakalandığında şahıs tanımlanmış olur ve işlem için onay verilir (Altun 2002). Farklı uygulamalar farklı biyometrik sistemleri gerektirmesinden dolayı, en iyi biyometrik teknolojisi diye bir şey yoktur (Soutar 2002, Woodward 1999). Biyometrik sistemler, evrenselliklerine, ayırt edebilme özelliklerine, sürekli oluşlarına, depolanabilirliğine, performanslarına, kullanıcıya uygunluğuna ve kullanım kolaylığı gibi özelliklerine göre karşılaştırılabilirler (Bolle ve ark. 2004). Buna göre parmakizi, istenen özellikleri sağlayan bir biyometrik sistem olmaktadır. Parmakizleri oldukça ayırt edici özelliklere sahiptirler, bazı yaralanmalardan veya kesilmelerden dolayı parmakizi geçici olarak hasara uğrasa bile parmakizi detayları belirlidir. Bu yüzden parmakizinin güvenilir ve kullanıcı dostu olması, kişi tanımada tercih edilme sebeplerindendir.

Parmakizi tanımanın iki temel açıklaması yapılabilir: (i) Süreklilik: Parmakizinin temel karakteristikleri zamanla değişmez; (ii) Bireysellik: Parmakizi bireyler için eşsizdir. Parmakizinin sürekliliği, anatomi ve derinin biçimsel yapısıyla ilgilidir. Parmakizinin bireyselliği ise deneysel sonuçlar ile kabul edilmesine rağmen altında yatan bilimsel gerçekler günümüzde araştırma konusu olarak önemini korumaktadır. Kanunlar, parmakizinin bireyselliğini tespit etmek için kullanılan bilimsel temellerin mahkemelerde parmakizi ile kimlik tespitinin kabul edilebilirliğine karar vermektedir. Sonuçta, parmakizi mahkemelerde delil olarak kullanılmaktadır.

2.1. Genetik Faktörler

Bir kişinin ayırt edici fiziksel karakteristikleri, embriyo gelişimini etkileyen rasgele süreçler olduğu gibi insan popülasyonu içerisindeki doğuştan kişisel genetik farklılıktan dolayıdır (Maio ve Maltoni 1997). İki bireyin mesela tek yumurta ikizlerinin genetik oluşumu birbirine yakındır. Bu yüzden biyometrik tabanlı kimlik ayırımı, gelişimindeki birtakım farklılıklardan dolayı gerçekleştirilebilir. Tek yumurta ikizleri en yakın genetik tabanlı ilişkiye sahip insanlar oldukları için bunların parmakizleri arasında maksimum benzerlik beklenebilir (Şekil 2.2, Şekil 2.3 ve Şekil 2.4). Şekil 2.2(a) ve (b)'deki parmakizleri arasındaki benzerlik ölçütü olan ayrıntı tabanlı eşleştirme skoru 48, (c) ve (d)'deki parmakizleri arasındaki ayrıntı tabanlı eşleştirme skoru ise 51'dir (Prabhakar 2001). Şekil 2.2(a) ve (c) arasındaki eşleştirme skoru 2 ve (b) ve (d) arasındaki eşleştirme skoru ise 4'tür. İkizlerin her ikisinin de parmakizi yapıları aynı olup sağ kement şeklindedir. Ne var ki otomatik parmakizi tanıma sistemleri kadar parmakizi uzmanları da ikizlerin parmakizlerinin farklılığını kolayca elde edebilirler.

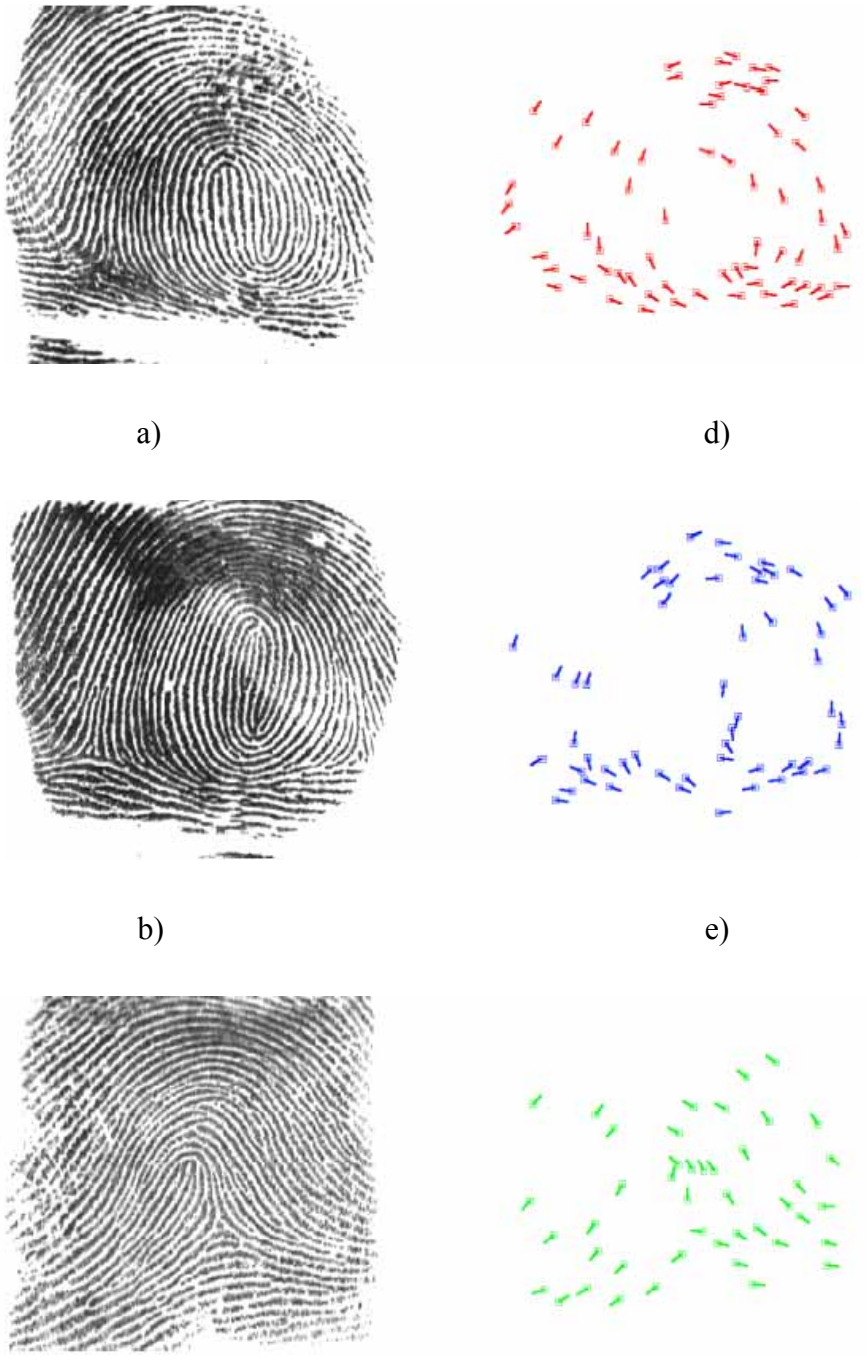
Tek yumurta ikizlerinin DNA yapıları bile aynı olmasına karşın parmakizi görüntüleri belirli farklılıklar içerir (Blue ve ark. 1993). Bu da tek yumurta ikizlerinin bile ayırt edilebilmesi için yeterlidir. Hatta bir kişinin farklı zamanlarda alınan aynı parmağına ait parmakizi görüntüsünden elde edilen özellikler bile değişebilmektedir (Şekil 2.5).



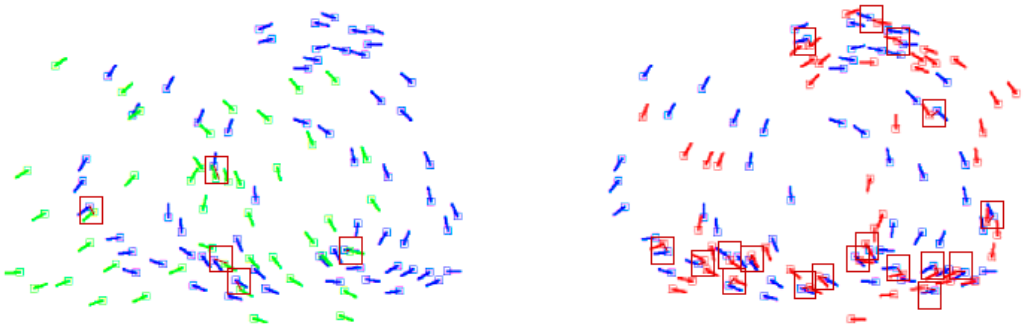
a) b)

c) d)

Şekil 2.2 Digital Biometrics (2004) şirketine ait optik tarayıcı kullanarak elde edilen iki ikiz kız kardeşe ait parmakizi görüntüleri; a) ve b) ikizlerden birinin aynı parmağına ait iki farklı parmakizi görüntüsü, c) ve d) diğer ikiz kardeşin aynı parmağa ait parmakizi görüntüleri

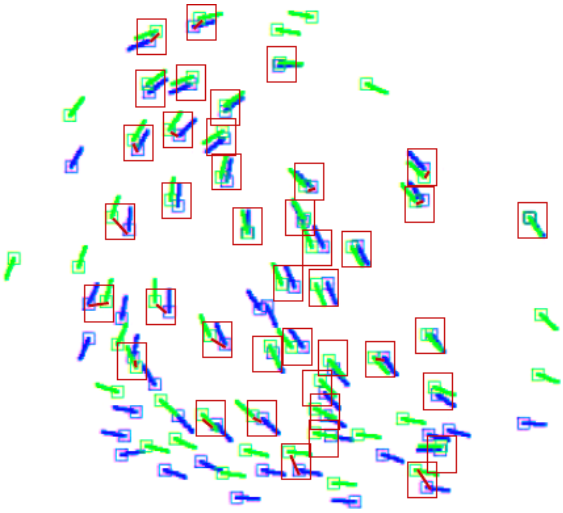


Şekil 2.3 Jain ve ark. (1997)'nın uyguladığı ayrıntı çıkarım algoritması kullanılarak elde edilmiş ikizlerin parmakizi ayrıntıları. a) ve b) ikizlere ait parmakizleri, c) ise başka bir kişiye ait parmakizi. d), e) ve (f) ise bunlardan elde edilmiş parmakizi ayrıntıları



a) b)

Şekil 2.4 Ayrıntı eşleştirme. a) Şekil 2.3(e) ve (f)'deki ikiz ve ikiz olmayan kişilerin parmakizi ayrıntılarına ait eşleştirme, eşleştirme skoru 5, b) Şekil 2.3(d) ve (e)'deki ikizlerin parmakizi ayrıntılarına ait eşleştirme, eşleştirme skoru 18 (Eşleştirme sonucu eşleşen ayrıntılar kutu içerisinde gösterilmiştir)



Şekil 2.5 Aynı parmakizinin farklı zamanlarda alınmış örneklerinden elde edilmiş ayrıntıların eşleştirilmesi, eşleştirme skoru 40 (Eşleştirme sonucu eşleşen ayrıntılar kutu içerisinde gösterilmiştir)

2.2. Çevresel Faktörler

Parmakizinin bireyselliği problemine ilgi iki türlüdür (Jain ve ark. 1997): Birincisi, bir kimlik delili olarak mahkemelerde parmakizinin kabul edilebilirliğini belirleyebilmektir. İkincisi ise otomatik parmakizi tanıma sistemlerinin performansını artırmaktır. Bu çalışmada, geliştirilen parmakizi tanıma sisteminin performansını artırıcı araştırmalar yapılmış ve performans artırıcı yenilikler yapılmıştır.

Parmakizlerinin şekillenmesi esnasındaki çevresel faktörler parmakizlerinin ayırt edici özelliğinde önemli rol oynarlar. Parmakizleri, tüm kabartı deseni, kabartı frekansı, lokasyonu ve tekil noktaların (merkez ve delta noktaları) pozisyonu, ayrıntı noktalarının lokasyonu, ayrıntı çiftleri arasındaki kabartı sayıları ve gözeneklerin lokasyonunu içeren geniş bir ayrıntı sayısı ile simgelenirler (Şekil 2.6). Tüm bu ayrıntılar parmakizinin bireyselliğini gösterir.



Şekil 2.6 Sağ kement tipli bir parmakizi görüntüsü ve ayrıntı noktaları

Parmakizinin bireyselliği problemi ilk defa Galton (1892) tarafından ifade edilmiştir. Farklı parmakizi tanıma ve doğrulama modellerinde kullanılan parmakizi ayrıntıları Tablo 2.1'de gösterilmiştir (Pankanti ve ark. 2002).

Tablo 2.1 Araştırmacıların kullandıkları parmakizi ayrıntıları

Araştırmacılar	**Kullandıkları Parmakizi Ayrıntıları**
Galton (1892)	Hatlar, merkez nokta ve delta noktası
Henry (1900)	Ayrıntı konumları, tipleri ve merkez ve delta nokta sayısı
Balthazard (1911)	Ayrıntı konumları, tipleri ve merkez ve delta nokta sayısı
Bose (1917)	Ayrıntı konumları, tipleri
Wentworth ve Wilder (1918)	Ayrıntı konumları
Pearson (1930)	Hatlar, merkez nokta ve delta noktası
Roxburgh (1933)	Ayrıntı konumları, tipleri, yönleri, merkez ve delta nokta sayısı ve parmakizi kalitesi
Cummins ve Midlo (1943)	Ayrıntı konumları, tipleri ve merkez ve delta nokta sayısı
Amy (1948)	Ayrıntı konumları, tipleri ve yönleri
Trauring (1963)	Ayrıntı konumları, tipleri ve yönleri
Kingston (1964)	Ayrıntı konumları, sayıları ve tipleri
Gupta (1968)	Ayrıntı konumları, tipleri ve hat sayısı
Osterburg ve ark. (1977)	Ayrıntı konumları ve tipleri
Stoney ve Thornton (1986)	Ayrıntı konumları, dağılımları, tipleri ve yönleri, hat sayıları, aynı kaynaktan alınan parmakizleri arasındaki varyasyon

2.3. Parmakizinin Bireyselliği

Güvenlik sistemleri ve kriminoloji araştırmalarında kişiye özel ve bireysellik gösteren yüz şekli, parmakizi, el ve yüz yapısı, iris, imza ve ses gibi örüntüler (pattern) kullanılmaktadır (Altun ve Koçer 2003). Biyometrik ile gerçekleştirilen tanıma işlemi, Şekil 2.7'de gösterildiği gibi genel olarak aşağıdaki aşamaları içerir (Nabiyev 2003).

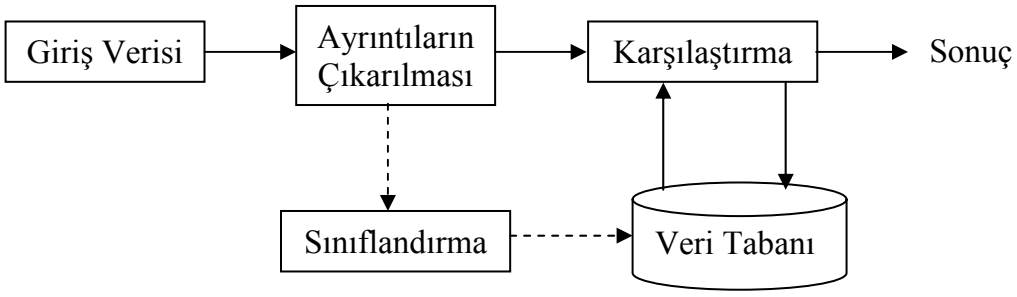
• Girişin Algılanması: Günümüzde bu iş için yapılmış özel algılayıcılar ve tarayıcılar mevcuttur. Biyometrik bilgi bu donanımsal araçlar kullanılarak elde edilir.

• Özelliklerin Çıkarılması: Bu aşamada, kullanılan karakteristiğe göre özellikler çıkarılır. Parmakizinde kişiye özgü bir özellik söz konusu olmaktadır. Pratikte parmakizine ait çatallaşma ve hat sonu özelliklerinden genellikle 12 tanesi parmakizi tanıma için yeterli görülmekte ve kullanılmaktadır.

• Karşılaştırma ve sonuç: Veri tabanı ile karşılaştırma yapılarak sonuç kararı verilir. Kullanılan özelliklerden tek birinin farklı olması sonucu etkilediğinden dolayı burada birebir eşleştirme algoritması kullanılmaktadır.

• Düzeltme ve Depolama: Bu kısımda karşılaştırma sonucunda bulunamayan kayıtlar veri tabanına eklenir ve güncelleme yapılmış olur.

• Sınıflandırma: Bu kısımda izler, yapısına ve özelliklerine göre sınıflandırılmaktadır.



Şekil 2.7 Parmakizi sınıflandırma ve tanıma işleminin genel aşamaları

Parmakizi kelimesi bireysellik ile eşanlamlı olarak algılanmasına rağmen parmakizinin eşsiz oluşu deneysel gözlem olmadan tespit edilemez. Parmakizinin yaygın kullanımı ile bireyselliği altında yatan neden, bilimsel temellerle ilgili olarak sürekli artan bir genelliğinin söz konusu olmasıdır. Toplumda artan hileli kimliklerden dolayı parmakizleri güvenilirliği, verimliliği ve uygunluluğu gibi durumlardan dolayı kişi tanımada oldukça yaygın bir şekilde kullanılmaktadır (Altun 2002).

Parmakizi, kişilerin kimlik saptanmasında en eski ve güvenilir biyometrik tanıma yöntemlerinden birisidir. FBI (1984, 1991), parmakizini suçlu tespitinde kullanan ilk otomatik sistemi kurmuştur. Daha önceden elle kağıt üzerinde yapılan işlemler günümüzde bilgisayar teknolojisinin yaygınlaşmasıyla otomatikleştirilmiştir. Kişinin sahip olduğu benzersiz özellikleri kullanan bu tür sistemler önemini her geçen gün arttırmaktadır.

Parmakizlerinin bireyselliğini aşağıdaki şekilde sıralayabiliriz:

1. Değişmez ve Değiştirilemez: İnsan elinin deri yüzeyinin özellikleri, burada kendine özgü olan tepe ve vadilerin olmasıdır. Burada kabartılı çizgilere papil denir (Gülver 1999). Papil hatlarını meydana getiren sinir ucu yumakları ve ter bezlerinin parmak uçlarındaki alt deri tabakasında olması, üst deri tabakasına alt deri görünümü kazandırır. Bunların merkez kısımları sarmal şekildedir. Parmakizleri, bu çizgilerin uzunluk, tür ve şekline göre sınıflandırılır (Henry 1900). Fakat her bir papil çizgisi kendine özgü olan kesilmeler, kalınlıklar, parçalanma ve birleşmeler, nokta ve boşluklar içermektedir. Bunlar sınıflandırılmış parmakizlerinin ayrıntılarını veya alt özelliklerini içermektedir. Dolayısıyla bu karmaşık çizgiler, tekrarlanmaz bir iz oluşturmaktadırlar. Yanma, sıyrılma ve aşınma gibi sebeplerden dolayı üst deri tabakası tahrip olsa bile, birkaç gün sonra yeniden meydana gelen üst deri tahrip olmadan önceki halini alır. Dolayısıyla üst deri tabakasındaki bir tahribat sebebiyle parmakizinin değişmesine imkân yoktur (İnandık 1998). Ancak alt deri tabakası tahrip olduğu takdirde parmakizi şekli de bozulmuş olur. Fakat bu defa yeni bir parmakizi şekli ortaya çıkmayıp, tahrip olan yerde papil hattı olmayan bir yara izi meydana gelir. Parmakizinin değişmesi mümkün olmadığı gibi, sahte parmakizi de yapmak hemen hemen imkânsızdır.

2. Benzemez ve Benzetilemez: İkiz kardeşlerin parmaklarındaki papil şekilleri, bir elde bulunan yan yana iki parmaktaki papil hatları, iki eldeki aynı cins parmakların izleri de hiçbir zaman birbirlerine benzememektedirler.

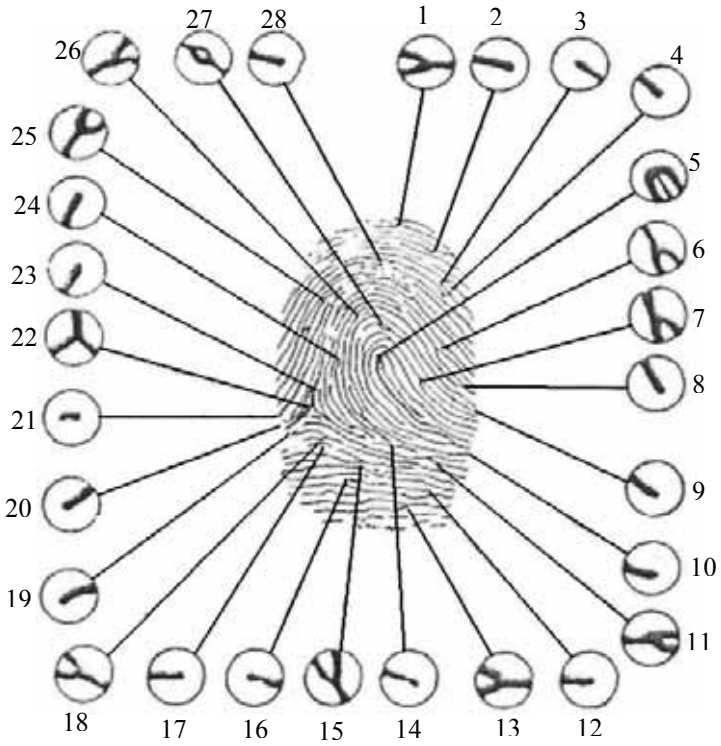
3. Tasnif Edilebilir: Parmakizi, üzerinde bulunan bazı özelliklerinden dolayı sınıflandırılır. Hat yapıları, hatların yönleri, hatlar üzerindeki sonlanmalar ve çatallaşmalar gibi ayırt edici özellikler sınıflandırmanın temel elemanlarıdır. Ayrıca günümüzde parmakizlerini tanıma amacıyla çok değişik özellik çıkarım algoritmalarıyla parmakizine ait farklı özellikler de elde edilebilmektedir. Bunlardan en çok bilineni ve bu tez çalışmasının da temelini oluşturan Fingercode adı verilen parmakizi özellik vektörüdür (Jain ve ark. 1999b).

Ayrıca parmakizini kesme, yakma vb. metotlar kullanarak değiştirmeye çalışanlar olmuş fakat bu izler daha da çok dikkat çekici hale gelmişlerdir. Parmağın bir kısmı yansa veya kesilse bile, etrafı aynı kalmakta ve bu da iz belirlemeye yetmektedir. Bu izlere benzer özellikte ve yine kişiler için ayırt edici özellikte olduğu düşünülerek araştırma konusu olmuş avuç içi ve ayak tabanında da benzer izler bulunmaktadır. Parmakizi, kimlik tespitinde yanılgıya imkân vermeyen delillerden birisidir. Veri tabanını yanıltacak sahte bir parmakizi yapılamaz. Dikkatle incelendiğinde parmakizlerindeki bazı hatların ani olarak sonlandığı veya ortadan ikiye ayrılıp bir çatal oluşturduğu görülür (Pernus ve ark. 1980). Bu karakteristik noktalar "ayrıntı" olarak tanımlanır. Bu noktalara, sırasıyla hat sonu ve çatal denir (Maio ve Maltoni 1997). Parmakizleri için esas ayırt edici özellik, ayrıntıların parmakizi içerisinde bulunduğu yerler ve yönleri ile birlikte belirli bir işlemle elde edilen gri seviye yoğunluk değerleridir (Altun 2002). Elimizdeki tüm parmakizlerimizi dikkatlice karşılaştırırsak, ana yapı olarak birbirine benzeseler de, nitemler göz önüne alındığında aslında çok farklı olduklarını görürüz. Bu farklılıklar öyle ayırt edicidir ki, yapılan çalışmalarda yeryüzündeki iki farklı insanın aynı parmakizine sahip olma olasılığı 64 milyarda bir olarak saptanmıştır (Nabiyev 2003).

Kriminolojik açıdan bu çizgilerin özelliklerine göre bir suçlunun cinsiyetinin ve yaşının belirlenmesi mümkündür. Kriminolojide suçluyu parmakizine göre tanıma Galton (1892), Henry (1900) ve Herschel (1918) tarafından önerilmiştir (Galton 2003).

2.4. Parmakizinin Karakteristik Özellikleri

Parmakizi üzerinde 6 temel karakteristik özellik mevcuttur. Bunlar Şekil 2.8'de görüldüğü gibi; (1) noktalar, (2) çatallar, (3) set sonları, (4) gözler veya adalar, (5) kısa setler ve (6) çubuklu göbektir. Aşağıdaki şekilde normal bir parmakizindeki karakteristik özellikler gösterilmiştir (Ballan 2000). Parmakizini tanımada kullanılan bu özelliklerin dışında bu çalışmada kullanılan FingerCode özellik vektörleri ve elde edilme aşamaları detaylı olarak anlatılmıştır. Parmakizini tanıma işleminin ilk aşamasında aynı referans noktasının bulunması gerekmektedir. Şekil 2.8'deki (5) nolu özellik bu parmakizinin referans noktasını göstermektedir.



Şekil 2.8 Parmakizlerinin karakteristik özellikleri

Çatal: (1, 6, 7, 11, 13, 15, 18, 25, 26); Çubuklu göbek: (5);
Delta, 3'lü delta: (22); Göz: (27); Kısa set: (21)
Set sonu: (2, 3, 4, 8, 9, 10, 12, 14, 16, 17, 19, 20, 23, 24, 28)

Parmakizi tipleri, parmakizlerine ait referans noktaların elde edilmesinde anahtar öneme sahiptirler. Parmakizleri, çizgi desenlerinin içerdiği delta ve göbek notlarının var olup olmadıklarıyla birlikte buralarda yapmış oldukları kavislere göre gruplandırılırlar. Parmakların çizgi desenleri, kemerler, sarmallar ve kementler olmak üzere temelde üç gruba ayrılırlar (Dönmez 2001). Parmakizi tiplerinde belirleyici özellik olan delta ve göbek noktaları, parmakizinin referans noktasının belirlenebileceği bölgelerdir. Bu bölgelerden delta noktası, genellikle parmakizi kenarına yakın yerdedir. Buna karşın göbek noktası ise parmakizinin orta kısmına yakın bölgededir. Dolayısı ile referans nokta elde edilirken göbek noktasının belirlenmesi gerekir. Böylece parmakizine ait geniş bir bölge, özellik vektörü elde edebilmek için işaretlenebilecektir. Referans nokta olarak hem delta hem de göbek noktasının elde edildiği durumlarda delta noktası elenir ve göbek noktası, referans nokta olarak belirlenir. Bu çalışmada, parmakizi özellik vektörünün elde edilmesi için göbek noktası sistem tarafından referans nokta olarak belirlenmiştir.

Parmakizi tanımada kullanılacak önemli aşamalardan biri, parmakizine ait referans noktasının şablonda elde edilen nokta ile aynı bölge olarak bulunmasıdır. Bu tez çalışmasında parmakizlerinin referans noktasının bulunması için yeni bir yöntem geliştirilmiştir. Geliştirilen bu yöntem ile elde edilen referans noktasındaki açı bilgisi, parmakizine ait özellik vektörü elde edilirken referans olarak ele alınmıştır. Böylece elde edilen özellik vektörünün yapısı şablon yapısı ile örtüştürülmüştür. Bunun sonucunda eşleştirme aşamasında büyük bir hız ve zaman kazancı ile birlikte yüksek doğruluk oranının da elde edildiği gözlemlenmiştir.

2.5. Parmakizi Tanıma Sistemleri

Parmakizi tanıma amacıyla birçok sistem kullanılmaktadır. Bu kullanılan sistemlere örnek olarak Henry (1900) sistemini, Otomatik Parmakizi Tanıma Sistemi (AFIS) ve Otomatik Parmakizi Doğrulama Sistemi (AFAS)'ni söyleyebiliriz (Gülver 1999). E ıtan Galton (1892) özellikleri ve Fingercode (Jain ve ark. 1999b) kullan

Henry Sistemi, ilk sistemlerden olup, günümüzde FBI (1984, 1991) gibi kuruluşlarda kullanılmaktadır (Banner ve Stock 1974, Brislawn ve ark. 1996). Bu sistemde kişinin 10 parmak bilgisi kullanılmaktadır. En iyi performansı elde etmek için oldukça zor aşamaların gerçekleştirilmesi gerekmektedir. Henry (1900) sınıflandırması aşağıdaki şekilde gerçekleştirilir:

Temel sınıflandırmada, parmaklar ve parmaklardaki desenlere farklı indis numaraları verilir. En yüksek puan sağ el başparmağı, en düşük ise sol el serçe parmağına aittir (Özdemir 1999). İkincil sınıflandırmada, her parmağın desen türüne bakılmaktadır. İkincil alt sınıflandırmada ise orta ve yüzük parmağı için göbek noktası olan merkez (core) ve delta arasındaki çizgi sayısı hesaplanır (Kaygısız 1995). Son olarak sonuç sınıflandırmada ise sağ elin küçük parmağında döngülü desenlerdeki tepecik sayısına bakılır. Sağ elde bu desen olmadığı durumlarda sol el küçük parmağına bakılır (Nabiyev 2003).

Parmakizlerinin sınıflandırılmasında kullanılan yöntemlerden bazıları aşağıda verilmiştir:

- Yakın Komşuluk Sınıflayıcı (Nearest-Neighbour Classifier)
- Doğrusal Sınıflayıcı (Linear Classifier)
- Yapay Sinir-Ağı Sınıflayıcı (The Artificial Neural-Net Classifier)
- Korelasyon Sınıflayıcı (Correlation Classifier)
- Gizli Markov Ağacı (Hidden Markov Tree)
- Öz Düzenlemeli Özellik Haritası (Self Organizing Feature Map)

Maio ve Maltoni (1997, 1998b), gürültüler de içerebilen parmakizinin yapısal sınıflandırılması için bir yöntem önermektedir. Bu yöntemin işlem adımları aşağıdaki gibi sıralanabilir (Nabiyev 2003):

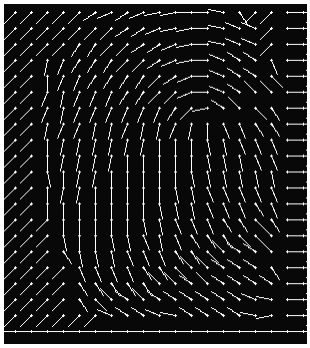
- Görüntüdeki kenarların elde edilmesi; kenar algılama için Roberts (1965), Sobel (Davis 1975), Prewitt (1970), Robinson (1977), Kirsch (1971) vb. gibi birçok yöntem mevcuttur.
- Yönlü görüntünün hesaplanması,
- Yönlü görüntünün bölümlenmesi,

- Önplan ve arkaplan segmentasyonu,
- Parmakizine ait hatların elde edilmesi,
- Küçük yapıların elenmesi,
- İnceltme ve morfoloji,
- İlişkisel grafın oluşturulması,
- Kesin olmayan graf eşleştirmesi,
- Sınıflandırma.

Parmakizinin uygun bir özeti, yönlü parmakizi görüntüsü tarafından elde edilir. Gürültülü görüntülerden de parmakizinin güvenilir bir şekilde elde edilebilmesi, parmakizi tanımada neden parmakizlerinin kullanımına önem verildiğinin bir göstergesidir. Parmakizi üzerine yapılan çalışmalar, genellikle görüntüdeki yön bilgisini kullanırlar. Parmakizinin özelliklerinin belirlenmesinde gerekli olan işlem adımları Şekil 2.9'da gösterilmiştir (Hong 1998, Hong ve ark. 1996, Pankanti 2002). Parmakizine ait iyileştirilmiş görüntüde, eleman yönleri arasındaki uyuşmazlıkları minimize etmek için uygun alan, bölgelere ayrılır. Her bölge ayrı ayrı işlenerek bu bölgelerin yönleri bulunur. Bu amaçla maskeleme yapılacağı gibi ilk ve son noktalar bulunarak basit bir açı bulma işlemi de gerçekleştirilebilir. Eğimdeki en büyük değişiklik, parmakizi yayının hangi bölgeye dahil olacağını belirler. Sonuçta yönlü görüntü elde edilmiş olur (Şekil 2.10). Parmakizi görüntüsüne ait bölgelerin açı bulma işlemi Bölüm 4'te detaylı bir şekilde açıklanmıştır.

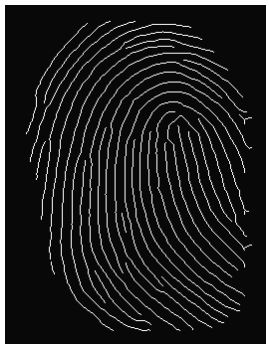


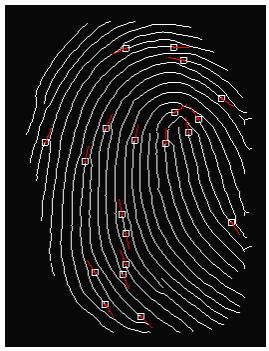
a)



b)

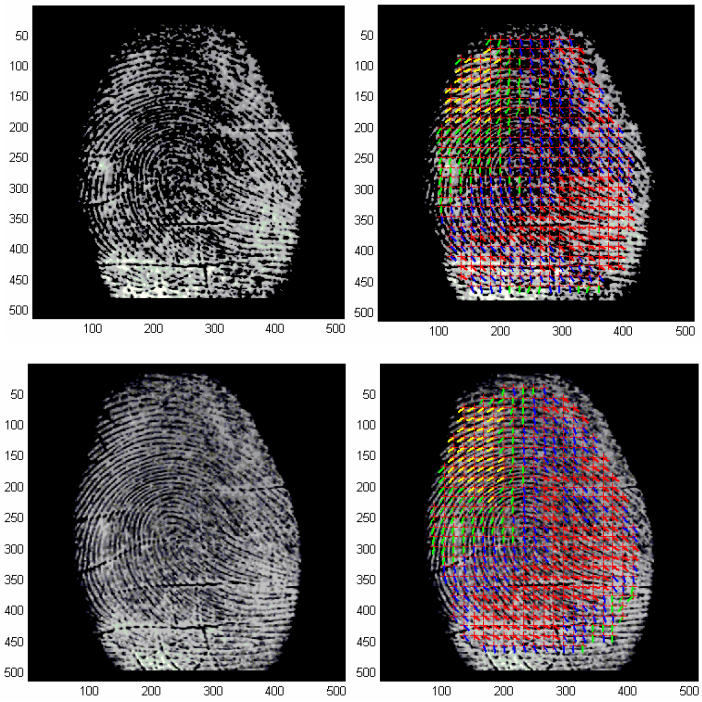






e)

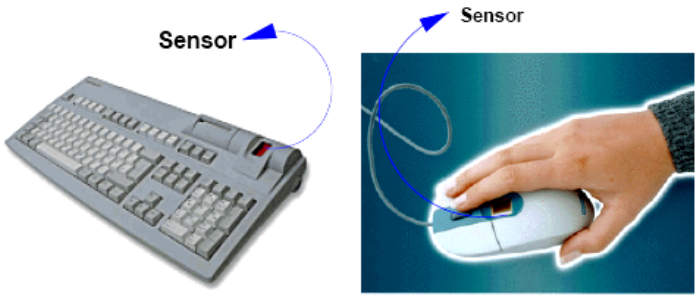
Şekil 2.9 Parmakizi özelliklerinin belirlenmesi aşamaları: a) gürültülü parmakizi görüntüsü; b) Parmakizi görüntüsüne ait yönsel bilginin elde edilmesi, c) bilgileri güçlendirilmiş parmakizi görüntüsü; d) ikileştirilmiş ve inceltilmiş parmakizi, e) parmakizi görüntüsünün özelliklerinin belirlenmesi



Şekil 2.10 İki farklı parmakizine ait elde edilen yönsel histogram

2.6. Parmakizi Eşleştirme

Tüm biyometrik özelliklerin arasında parmakizleri en yüksek güvenilirlik seviyesine sahip olan sistemlerden birisidir (Jain ve ark. 1997). Ayrıca kriminal olayları aydınlatmak amacıyla adli uzmanlar tarafından oldukça sık bir şekilde kullanılırlar (CJIS 1999; Allahverdi ve ark. 2004). Bir parmakizi, parmaktaki kabartı desenlerinin akışını gösterir. Geleneksel olarak parmakizi kalıpları, parmağın mürekkeplenerek kağıt üzerine bastırılması ile çıkarılırlar (Geçgil 2000). Günümüzde ise bu görüntüleri sayısal olarak elde etmek amacıyla tasarlanmış sensörler mevcuttur. Bu sensorlar, bilgisayara yetkili kişilerin erişimini sağlamak üzere klavye veya mouse gibi bilgisayar çevre aygıtlarına entegre edilebilirler (Şekil 2.11). Parmakizi tanımada donanımsal aygıtların geliştirilmesiyle birlikte sivil yaşamda ve adli uygulamalarda otomatik parmakizi tabanlı doğrulama sistemlerinin kullanımı yaygınlaşmıştır.



Şekil 2.11 Parmakizi okuma sensörü eklenmiş bilgisayar çevre aygıtları

Parmakizinin bireyselliği, topografik kabartı yapısıyla ve ayrıntı noktaları olarak adlandırılan belirli kabartı anomalileriyle belirlenir. Ayrıntı noktalarının dağılımı, eşleşmede ve iki parmakizi arasındaki benzerliği tespit etmede (Jain ve ark. 1997a, Kovács-Vajna 2000) kullanılmasına rağmen kabartı yapısı olarak tanımlanan ayrıntılı konfigürasyon, parmakizi sınıfını (Karu ve Jain 1996, Senior 2001) belirlemede kullanılır. Otomatik parmakizi tanıma sistemleri, parmakizi eşleştirmede ve parmakizi indekslemede bu yapıları kullanmaktadır.

Parmakizi eşleştirme teknikleri, ayrıntı tabanlı veya korelasyon tabanlı olarak sınıflandırılabilir (Prabhakar ve Jain 2002). Ayrıntı tabanlı teknikler, eşleşen ayrıntıların toplam sayısını hesaplamayı ve iki ayrıntı noktaları kümesini düzenlemeyi amaçlar (Pernus ve ark. 1980, Mehtre ve Murthy 1986). Korelasyon tabanlı teknikler ise eğer iki parmakizindeki kabartılar hizalı ise kabartıların ayrıntılı desenlerini ve görünen izleri karşılaştırmayı amaçlar (Sibbald 1997, Bazen ve ark. 2000). Korelasyon tabanlı tekniklerde, ayrıntı tabanlı tekniklerde karşılaşılan bazı güçlükler aşılmıştır. Korelasyon tabanlı teknikler, referans noktası etrafında kesin bir bölgeyi kapsar ve parmakizi görüntüsünün döndürülmesinden veya ötelenmesinden etkilenir. Bu tez çalışmasında parmakizlerini tanımak amacıyla korelasyon tabanlı teknik geliştirilerek kullanılmış ve parmakizinin döndürülmesinden ve ötelenmesinden etkilenmeyecek bir sistem tasarlanmıştır. Sonuç olarak parmakizinin döndürülmesinden ve ötelenmesinden bağımsız bir yapı oluşturularak parmakizlerinin tanınması gerçekleştirilmiştir. Geliştirilen bu yapı ileriki bölümlerde açıklanmıştır.

2.7. Sonuç

Temel karakteristiklerinin zamanla değişmemesinden dolayı parmakizi, kişilerin tanınması için eşsiz bir biyometrik bilgidir. İki bireyin mesela tek yumurta ikizlerinin genetik oluşumu birbirine çok yakındır, bu yüzden biyometrik tabanlı kimlik ayırımı, gelişimindeki birtakım farklılıklardan dolayı gerçekleştirilebilir. Tek yumurta ikizlerinin DNA yapıları bile aynı olmasına karşın parmakizi görüntüleri belirli farklılıklar içerir. Bu da tek yumurta ikizlerinin bile ayırt edilebilmesi için yeterlidir. Parmakizleri, kabartı desenleri, kabartı frekansları, konumları ve tekil noktalar olarak adlandırılan merkez ve delta noktaların pozisyonları, ayrıntı noktalarının lokasyonu, ayrıntı çiftleri arasındaki kabartı sayıları ve gözeneklerin lokasyonunu içeren geniş bir ayrıntı sayısı ile simgelenirler. Tüm bu ayrıntılar parmakizinin bireyselliğini gösterir. Parmakizinin değişmesi mümkün olmadığı gibi, sahte parmakizi de yapmak hemen hemen imkânsızdır. Böylece, kimlik tanıma problemi için en iyi yöntem olarak parmakizi yöntemi ön plana çıkmaktadır.

3. PARMAKİZİ ÖN İŞLEME

Parmakizi karşılaştırması için kullanılan otomatik sistemlerin birçoğu ayrıntı eşleştirme üzerinedir. Parmakizi görüntüsündeki kabartı yapılarının kalitesi, önemli bir karakteristiktir. Kabartılar, ayrıntı çıkarımı için gerekli olan karakteristik ayrıntı bilgilerini içerirler. İyi tanımlanmış bir parmakizi görüntüsünde kabartı ve vadiler birbirlerini paralel olarak takip ederler. Bir mürekkeplenmiş parmakizinin sayısallaştırılması veya doğrudan sensörden taranmasıyla elde edilen parmakizi görüntüsünden ayrıntıları çıkarmak için öncelikle bir hat yapısı elde edilmelidir. Elde edilen görüntünün kalitesi gibi parmakizi görüntüsündeki hat yapıları her zaman iyi tanımlanmamış olabilir (Hollingum 1992). Bundan dolayı elde edilen görüntünün ön işlenmesi gibi bazı iyileştirme işlemleri daha güvenilir ayrıntı çıkarımı elde etmek için gereklidir.

Günlük hayatta parmakizi görüntüleri, gürültülü elemanlardan dolayı iyi elde edilemeyebilir ve tanımlanmayabilir. Gürültü, parmakizi görüntüsünde gri seviye kaymalarına ve kabartı yapısındaki netliğin bozulmasına sebep olur. Bu bozukluk, derideki değişimden, kir, nem ve yaralardan ve parmakizi tarama cihazındaki düzgün olmayan dağılımdan kaynaklanabilir (Jain ve ark. 2001). Bu yüzden gürültüleri azaltmak ve vadilere karşı kabartıların daha iyi tanımlanmasını gerçekleştirmek amacı ile görüntü iyileştirme teknikleri kullanılır (Altun ve ark. 2005).

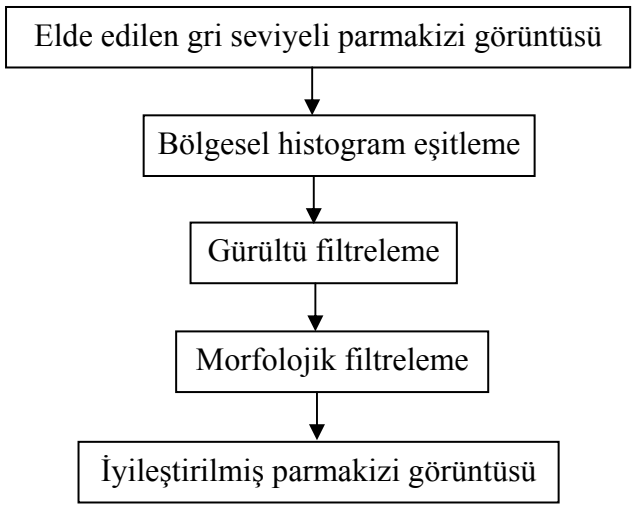
Güvenilir olarak ayrıntı çıkarımı, parmakizi sınıflandırması ve tanınmasında önemli adımlardan birisidir. Parmakizi görüntüsündeki bozuk hat yapıları için görüntü iyileştirme yöntemleri kullanılarak hat yapıları daha açık hale getirilir. Literatürde önerilen özellik belirleme metotlarının çoğu, doğrudan gri seviyeli görüntülerden özellik çıkarma üzerinedir (Hong ve ark. 1998, Maio ve Maltoni 1998a, 1999a,b). Bu yaklaşımla ilgili olarak bu tez çalışmasında gri seviyeli parmakizi görüntülerini iyileştirmek için iki metot araştırılmıştır. Birinci metotta bölgesel histogram eşitleme ve gürültü azaltma filtreleri kullanılmıştır. İkinci metotta

dalgacık dönüşüm (Meyer 1992) ve parmakizi görüntüsünün yapısına uygun olduğunu düşündüğümüz iki boyutlu dalgacık dönüşümün geliştirilmişi olan çevritsel dönüşüm (Do ve Vetterli 2002) parmakizi için geliştirilerek uygulanmıştır.

3.1. Filtreleme Teknikleri Kullanılarak Görüntü İyileştirme

Bu çalışmada ön işleme için filtreleme ve gürültü azaltma tekniklerinin birleştirildiği bir iyileştirme işlemi kullanılmıştır. Parmakizi görüntüsünü ve görüntüdeki hatları iyileştirmek için önerilen iyileştirme işleminin temel aşamaları Şekil 3.1'de gösterilmiştir.

Bölgesel histogram eşitleme, görüntü kontrastını yaymak için kullanılır. Parmakizi görüntüsündeki gürültüleri azaltmak için filtreleme yapılır. Morfolojik filtreleme ise gürültülü bölgelerdeki bozuklukları ortadan kaldırmak ve doğru hatlar üzerindeki bazı boşlukları doldurmak için kullanılır (Altun ve ark. 2005).



Şekil 3.1 Filtreleme tabanlı parmakizi görüntüsü iyileştirme aşamaları

3.2. Kontrast İyileştirme

Histogram, görüntü üzerindeki piksellerin değerlerinin grafiksel ifadesidir. Grafikte, yansıma değerleri (genellikle 0-255) *x*-ekseni üzerinde ve bu değerlerin görüntüde tekrar etme sıklığı ise *y*-ekseninde görülür (Maillet ve Sharaiha 1999). Buna görüntü histogramı veya gri-seviye histogramı denir (Castelman 1996). Görüntü histogramı, görüntünün her bir noktasındaki piksellerin tespiti ile bu piksellerin sayısının ne olduğunu gösterir. Bu sayede histogram üzerinden görüntü ile ilgili çeşitli bilgilerin çıkartılması sağlanır.

Histogram eşitleme, görüntüdeki düşük karşıtlığı iyileştirmek amacıyla kullanılır. Histogramdaki frekans bilgisine bağlı olarak yapılan, doğrusal olmayan eşleşmedir. Yüksek frekanslı piksel seviyesi geniş piksel alanına, düşük frekanslı piksel seviyesi ise dar piksel alanına yerleştirilir. Böylece, çok kullanılan piksel seviyeleri belirgin hale getirilir.

Histogram eşitleme, *q* gri seviye içerisine düzenli dağılımlı bir *p* gri seviye eşlemesini tanımlar. Bu eşleme, maksimum histogramın yanında gri seviye aralığını genişleterek kontrastı yayar (Greenberg ve ark. 2000). Kontrast, görüntü piksellerinin çoğu için genişletildiğinden dolayı dönüşüm, görüntü ayrıntılarının ortaya çıkarılabilirliğini artırır. Bir piksel yoğunluğu seviyesi r_k için muhtemel yoğunluk fonksiyonu aşağıdaki gibidir (Yaman ve ark. 2001):

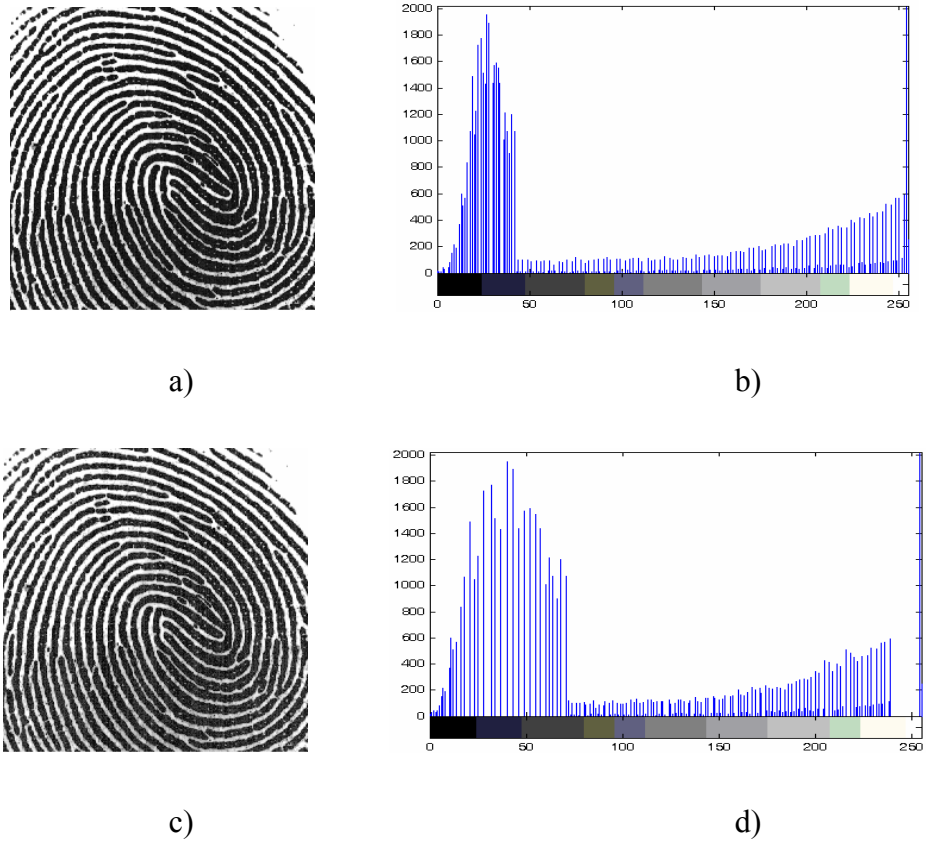
$$p_r(r_k) = \frac{n_k}{n} \tag{3.1}$$

Burada r_k, k'ıncı gri seviye, $0<r_k<1$; k=0,1,2,...,255; n_k, r_k gri seviyesindeki piksellerin sayısı ve n, toplam piksel sayısıdır. Histogram, r_k'ya karşı $p_r(r_k)$'nın çizilmesiyle elde edilir. k seviyesinin yeni yoğunluk değeri s_k aşağıdaki gibi tanımlanır:

$$s_k = \sum_{j=0}^{k} \frac{n_j}{n} = \sum_{j=0}^{k} p_r(r_j) \tag{3.2}$$

Bu çalışmada parmakizi görüntüsündeki hatların genişliği dikkate alınarak 11×11 piksellik sınırlı bir pencere kullanılmak suretiyle histogram eşitleme uygulanmıştır. Sonuçta bölgesel olarak kontrast genişlemesi gerçekleştirilmiş ve her bir pikselin komşu piksellere göre yoğunluğu değişmiştir.

Şekil 3.2, lokal histogram eşitleme ile elde edilen görüntü kontrastını gösterir. Burada parmakizi görüntüsünün belli dağılıma sahip yansıma değerleri üzerinde işlemler yapılarak görüntüde istenen özellikler daha belirgin hale getirilmiştir. Eşikleme işlemi ile parmakizi görüntüsü üzerindeki piksel değerlerinin belirli bir değere göre atılması ve yerine diğer değer/değerlerin yerleştirilmesi gerçekleştirilmiştir.



Şekil 3.2 a) Orijinal parmakizi görüntüsü ve b) histogramı, c) lokal histogram eşitleme sonucu elde edilen parmakizi görüntüsü ve d) yeni histogram skalası

3.3. Gürültü Azaltımı İçin Filtreleme

Dürtü gürültüsü görüntü işlemede sıklıkla ortaya çıkan bir problemdir. Gürültünün oluşmasına iletim kanal hataları, sensor hataları, kenar keskinleştirme prosedürleri, AC güç parazitleri ve atmosferdeki elektriksel emisyonlar neden olmaktadır (Yılmaz 2005). Dürtü gürültüsünün güçlü genliği nedeniyle, insan görsel algısı çok hassaslaşmakta ve gürültünün azaltılması görüntü işlemenin önemli bir noktası olmaktadır (Lin ve Wilson 1988). Görüntüden etkili bir şekilde gürültü azaltma işleminin başarısı, bazı uygulamalarda oldukça önemlidir. Bu işlemdeki zorluk, gürültü kaldırılırken görüntü içindeki detayların korunabilmesidir. Çoğu geleneksel dürtü gürültü azaltma filtreleri, bozulan bölgeleri onarırken, görüntüdeki bozulmamış bölgeleri değiştirmektedir. Medyan filtre ve onun türevleri bu gürültüyü azaltmak için kullanılan metotların en çok bilinenlerinden birisidir. Bununla birlikte medyan filtreleme işlemi görüntüdeki çok ince detayları da ortadan kaldırabilmektedir. Görüntü üzerinde dış etkenlerden kaynaklanan gürültüyü azaltmak için kullanılan filtreler ise lineer filtre, ortanca (median) filtre ve uyarlamalı (wiener) filtre olarak sayılabilir. Lineer filtreler belirli tip gürültüleri azaltmak için kullanılır. Gaussian filtreler lineer filtre sınıfındandır. Medyan filtreler de lineer filtrelemedeki işlemlerin yanı sıra sonuç pikselin değeri, komşu piksellerin ortalama değeri ile belirlenir. Uyarlamalı filtreler, bölgesel görüntü varyansının uygun hale getirilmesinde kullanılır. Uyarlamalı filtre, görüntüdeki kenar veya diğer bölgeleri bulmada lineer filtrelerden daha seçicidir.

Uzaysal medyan filtre, tek sayılı bir pencerenin parmakizi görüntüsü üzerinde hareket etmesi ve pencere içerisindeki piksellerin parlaklıklarına göre sıralanması ve görüntünün orta pikseli ile pencere içerisindeki orta pikselin değiştirilmesi prensibine dayanır. Medyan filtrenin performansı pencere boyutuna bağlıdır. Pencere boyutu büyük seçildiğinde görüntünün önemli kenar detaylarında hasarlar oluşur. Pencere boyutu küçük seçildiğinde ise medyan filtre gürültüyü yeterince bastıramaz.

Ancak pencere boyutu ve gürültü bastırımı arasında bir optimizasyon yapılabilir. Uzaysal medyan filtre yinelemeli olarak kullanıldığında, kök görüntü

yinelemenin erken safhasında ortaya çıktığından ve ileri gürültü azaltımı elde edilemeyeceğinden dolayı görüntü pekiştirme işlemi durur (Erçelebi ve Koç 2005).

Bir boyutlu, vektörel bir işaret medyan filtre kullanılarak işlenebilir. Burada filtrenin derecesi önem kazanır. Medyan filtre doğrusal bir filtre değildir. Komşu piksellerin sıralanarak medyan değerinin alınması ile elde edilir. Vektörel bir işaret için medyan filtre aşağıdaki şekilde ifade edilir:

$$f' = Medyan_filtre(f, n) \tag{3.3}$$

Burada f vektörü, f' sonuç vektörü ve n ise medyan filtrenin derecesidir. Eğer n değeri tek sayı ise ortanca değer $f'(k)$, $f(k-\frac{n-1}{2}:k+\frac{n-1}{2})$, çift olması durumunda ise $f(k-\frac{n}{2}), f(k-\frac{n}{2}+1),....., f(k+\frac{n}{2}-1)$ olarak belirlenir.

Örneğin f vektörü tek boyutlu bir işaret olsun.

$$f = [1, 3, 2, 4, 7, 11, 10, 6, 4, 5, 2] \tag{3.4}$$

ve f işareti 3. dereceden tek boyutlu bir uzaysal medyan filtre tarafından filtrelensin.

$$f' = Medyan_filtre(f, 3) = [1, 2, 3, 4, 7, 10, 10, 6, 5, 4, 2] \tag{3.5}$$

Eğer f' işareti aynı filtre ile tekrar filtrelenirse elde edilen yeni işaret aşağıdaki gibi ifade edilir:

$$f'' = Medyan_filtre(f', 3) = [1, 2, 3, 4, 7, 10, 10, 6, 5, 4, 2] \tag{3.6}$$

f' işareti f işaretinin kök işareti olarak tanımlanır, çünkü $f'' = f'$ olacak ve dolayısıyla f' işareti aynı filtre ile filtrelenmeye devam edildiğinde değişmeyecektir (Erçelebi ve Koç 2005).

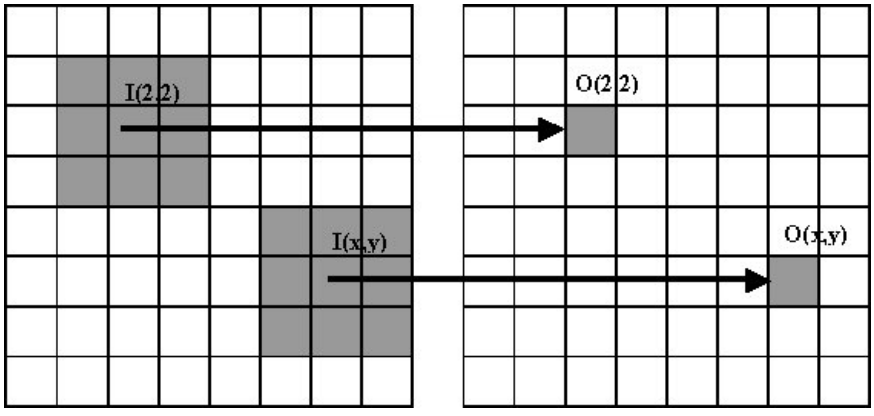
Görüntüden gürültüleri arındırmak ve böylece görüntüyü iyileştirmek amacıyla 2 boyutlu medyan filtre kullanılır. Amaç görüntüdeki gürültüleri azaltmakla birlikte görüntünün kenar bilgilerini de korumaksa medyan filtre evrişimden daha etkilidir. Medyan filtre aşağıdaki şekilde ifade edilir:

$$B = Medyan_filtre(A,[m\ \ n]) \tag{3.7}$$

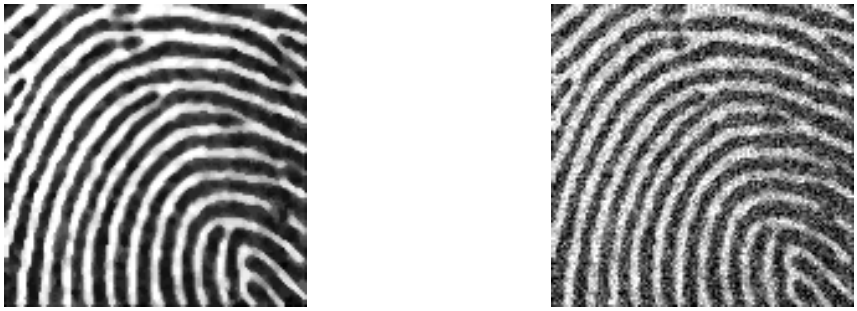
Burada A, iki boyutlu görüntü matrisini; $[m\ \ n]$ medyan filtrenin pencere boyutunu ve B, filtrelenmiş görüntü matrisini simgeler. Her bir çıkış pikseli, giriş görüntüsündeki uygun piksel etrafındaki $m \times n$ boyut komşuluğundaki ortalama değeri içerir. Bir görüntü matrisine 3×3 boyutlu bir medyan filtre uygulanması sonucu elde edilen yeni görüntü matrisi Şekil 3.3'te gösterilmiştir (Baxes 1994). Burada I, giriş görüntüsünü; O ise medyan filtrenin uygulanması sonucunda elde edilen sonuç görüntüsünü simgeler.

Filtrelerin performanslarını belirlemek için orijinal parmakizi görüntüsüne Gaussian gürültüsü (Ortiz ve ark. 2004) uygulanmıştır. Gaussian gürültüsü, ortalama ve varyans değerlere göre görüntüye beyaz gürültüler eklemektedir. Gürültü için ortalama değeri 0 ve varyans değeri 0,02 olarak belirlenmiştir.

Bu çalışmada Matlab programı kullanarak her biri 3×3 boyutlu lineer filtre, medyan filtre ve uyarlamalı filtre gürültülü parmakizi görüntülerine uygulanmıştır. Gürültüden arındırılmış parmakizi görüntüleri Şekil 3.4'te gösterilmiştir (Altun ve ark. 2005).



Şekil 3.3 İki boyutlu görüntüye medyan filtre uygulanması



a) orijinal parmakizi görüntüsü

b) gürültü eklenmiş parmakizi görüntüsü



c) lineer filtre sonrası parmakizi görüntüsü

d) medyan filtre sonrası parmakizi görüntüsü

e) uyarlamalı filtre sonrası parmakizi görüntüsü

Şekil 3.4 Filtreleme teknikleri uygulanarak iyileştirilmiş parmakizi görüntüleri

Filtrelerin karşılaştırılması farklı hata metotları ile birlikte "en uç sinyal-gürültü oranı" (PSNR)'na göre gerçekleştirilmiştir (Tablo 3.1). En uç sinyal-gürültü oranı, doğru verinin elde edilme oranı veya saptanmak istenen sinyal ile bunu etkileyen gürültüler arasındaki oran olarak tanımlanır (Muresan ve Parks 2003). PSNR aşağıdaki formül kullanılarak hesaplanır:

$$PSNR = 10\log_{10}\{\frac{255^2}{\frac{1}{NM}\sum_{n_1=1}^{N}\sum_{n_2=1}^{M}\left[f(n_1,n_2)-\hat{f}(n_1,n_2)\right]^2}\} \tag{3.8}$$

Burada $f(n_1,n_2)$ ve $\hat{f}(n_1,n_2)$, $1 \le n_1 \le N$, $1 \le n_2 \le M$, sırasıyla $N \times M$ boyutlu orijinal parmakizi görüntüsü ve iyileştirilmiş parmakizi görüntüsünü ifade eder.

Gürültülü görüntünün PSNR değeri 12,18 olarak elde edilmiştir. Filtreleme teknikleri uygulanmış parmakizi görüntüsünde bu oranın artması beklenir. Çünkü gürültüden arındırılmış bir parmakizi görüntüsünün piksel değerleri, orijinal parmakizi görüntüsünün piksel değerlerine yaklaşmalıdır. Dolayısı ile ortalama ölçüm hatasının düşmesi beklenirken PSNR değeri de artar.

Tablo 3.1'den görüleceği üzere parmakizi görüntüsündeki gürültü azaltımında ve gri seviyeli iyileştirilmiş parmakizi görüntüsü elde edilmesinde en iyi performansı medyan filtre sağlamıştır. Medyan filtre uygulanmış gürültülü parmakizi görüntüsü incelendiğinde parmakizindeki hat yapılarının daha belirgin ve orijinal görüntüye yakın bir görüntünün elde edildiği görülmektedir. Diğer hata oranları olan ortalama ölçüm hatası, normalize edilmiş Euclidean mesafesi ve sinyal-gürültü oranı da dikkate alındığında gürültülü parmakizi görüntüsünden gürültüleri en iyi şekilde çıkaran filtrenin medyan filtre olduğu görülmektedir.

Tablo 3.1 Filtrelemeler sonucu elde edilen görüntülerin lokal histogram eşitleme uygulanmış görüntüyle karşılaştırma sonuçları

Hata Metotları	**Gürültülü görüntülere uygulanan filtreler**		
	lineer	**medyan**	uyarlamalı
Maksimum Mutlak Fark	194,00	**255,00**	137,00
Ortalama Ölçüm Hatası	48,74	**24,17**	34,14
Normalize Edilmiş Euclidean Mesafesi	0,25	**0,16**	0,16
Sinyal-Gürültü Oranı	14,43	**15,97**	15,64
Tepe Sinyal-Gürültü Oranı	18,92	**20,46**	20,15

3.4. Dalgacık Dönüşüm ve Çevritsel Dönüşüm ile Parmakizi Görüntüsü İyileştirme

Dalgacık kuramında meydana gelen ilerlemeler, gürültülü görüntülerden gerçek işareti elde etmek için kullanılabilecek yeni bir yöntem olan dalgacık eşikleme yönteminin gelişmesini sağlamıştır. Dalgacık dönüşümü, gürültü temizleme (Donoho ve Johnston 1994) ve sıkıştırma (Villasenor ve ark. 1995) gibi işaret işleme uygulamalarında başarılı sonuçlar veren bir işaret dönüşüm tekniğidir. Bir işaretin dalgacık dönüşümü, o işareti yüksek geçiren ve alçak geçiren süzgeçlerden geçirdikten sonra çıktıyı seyreltmek ve alçak geçiren süzgecin çıktısını bu şekilde belirli sayıda işlemek yoluyla elde edilir (Bala ve Ertüzün 2002). Yüksek geçiren süzgeçlerin çıktıları dalgacık katsayılarını, en son basamaktaki alçak geçiren süzgecin çıktıları ise ölçek katsayılarını oluşturmaktadır. İki boyutlu işaretler için, aynı işlem ayrılabilir şekilde yapılabilmektedir. Benzer şekilde çevritsel dönüşüm de görüntü iyileştirme amaçlı kullanılmak üzere dalgacık dönüşümün geliştirilmiş halidir (Altun ve ark. 2005).

3.4.1. Dalgacık dönüşüm ile parmakizi görüntüsü iyileştirme

Dalgacık teorisi, işaret işleme, görüntü işleme, haberleşme, matematik, fizik, biyomedikal, astronomi vb. gibi birçok alanda yaygın olarak kullanılmaktadır. Dalgacık dönüşümü matematiksel olarak aşağıdaki şekilde tanımlanır:

$$W(a,b) = \frac{1}{\sqrt{a}} \int_{-\infty}^{\infty} f(t).\overline{\psi\left(\frac{t-b}{a}\right)}.dt \tag{3.9}$$

Bu denklemlerde $a > 0,\ b \in \Re$ 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 (Vatansever ve ark. 2005).

Eğer $f(t)$ işareti; iki boyutlu ise, dalgacık analizi için bir tane iki boyutlu $\varphi(x,y)$ ölçekleme fonksiyonu ve üç tane de iki boyutlu $\psi^Y(x,y)$ (yatay), $\psi^D(x,y)$ (dikey) ve $\psi^K(x,y)$ (köşegensel) dalgacık fonksiyonları gereklidir (Tablo 3.2). Bunların her biri, bir boyutlu φ ölçekleme fonksiyonu ve ilgili ψ dalgacık fonksiyonundan elde edilir.

Tablo 3.2'deki ölçekleme ve dalgacık fonksiyonlarından ölçeklenmiş ve ötelenmiş temel fonksiyonları $i = \{Yatay, Dikey, Köşegensel\}$ için

$$\begin{aligned}\varphi_{j,m,n}(x,y) &= 2^{j/2}\varphi\left(2^j x-m, 2^j y-n\right)\\ \psi^i_{j,m,n}(x,y) &= 2^{j/2}\psi^i\left(2^j x-m, 2^j y-n\right)\end{aligned} \tag{3.10}$$

elde edilerek MxN boyutundaki $s(x,y)$ görüntüsünün 2 boyutlu ayrık dalgacık dönüşümü $i = \{Yatay, Dikey, Köşegensel\}$ olmak üzere aşağıdaki şekilde hesaplanır:

$$\begin{aligned}W_\varphi(j_0,m,n) &= \frac{1}{\sqrt{MN}}\sum_{x=0}^{M-1}\sum_{y=0}^{N-1}s(x,y)\varphi_{j_0,m,n}(x,y)\\ W^i_\psi(j,m,n) &= \frac{1}{\sqrt{MN}}\sum_{x=0}^{M-1}\sum_{y=0}^{N-1}s(x,y)\psi^i_{j,m,n}(x,y)\end{aligned} \tag{3.11}$$

ve 2 boyutlu ters dalgacık dönüşümü;

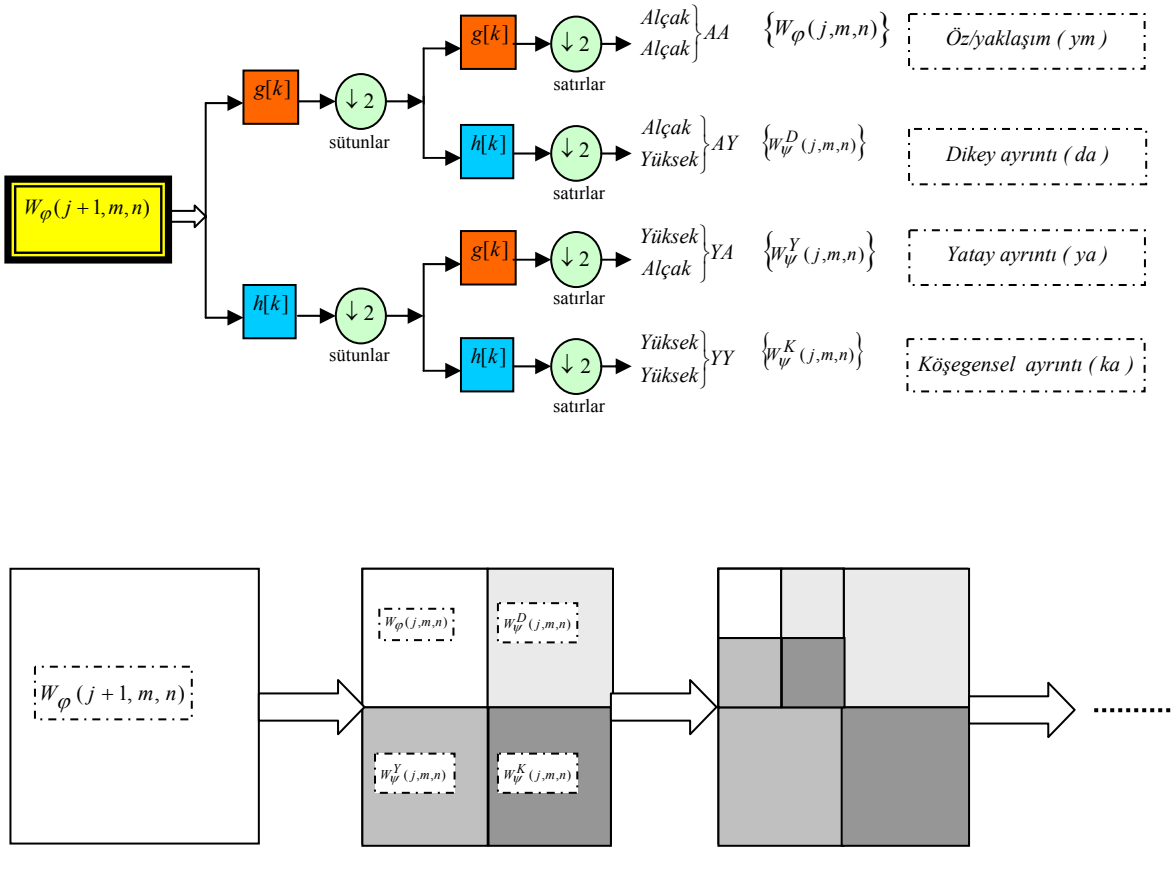
$$\begin{aligned}s(x,y) &= \frac{1}{\sqrt{MN}}\sum_m\sum_n W_\varphi(j_0,m,n)\varphi_{j_0,m,n}(x,y) +\\ &+\frac{1}{\sqrt{MN}}\sum_{i=Y,D,K}\sum_{j=j_0}^{\infty}\sum_m\sum_n W^i_\psi(j,m,n)\psi^i_{j,m,n}(x,y)\end{aligned} \tag{3.12}$$

ile hesaplanır (Wickerhauser 1994).

İki boyutlu dalgacık dönüşümü için Şekil 3.5'te görülen analiz filtre seti kullanılır. Burada her bir seviyedeki ayrışımda çıkış olarak elde edilen öz/yaklaşım alt görüntüsü, diğer analiz filtre setine giriş olarak uygulanır.

Tablo 3.2 İki boyutlu ölçekleme fonksiyonu ve dalgacık fonksiyonları

Fonksiyon	**İfadesi**	**Ölçü/değerlendirme**
Ölçekleme	$\varphi(x,y)=\varphi(x)\varphi(y)$	Düşük çözünürlüklü görüntü
Dalgacık	$\psi^{Y}(x,y)=\psi(x)\varphi(y)$	Sütunlardaki değişimler (yatay kenar)
	$\psi^{D}(x,y)=\varphi(x)\psi(y)$	Satırlardaki değişimler (dikey kenar)
	$\psi^{K}(x,y)=\psi(x)\psi(y)$	Köşegenlerdeki değişimler



Şekil 3.5 İki boyutlu dalgacık analiz filtre seti ve ayrışım sonuçları

$M \times N$ boyutundaki $s(x, y)$ görüntüsü; Şekil 3.5'teki analiz filtre seti kullanılarak bir seviye ayrıştırılırsa, dört tane $\frac{M}{2} \times \frac{N}{2}$ boyutunda alt görüntü elde edilir (Vatansever ve ark. 2005).

$$s \rightarrow \begin{Bmatrix} ym^1 & | & da^1 \\ - & & - \\ ya^1 & | & ka^1 \end{Bmatrix} \tag{3.13}$$

Yukarıdaki ifadede yer alan;

ym^1 alt görüntüsü; s ana görüntüsünün satır ve sütunlarındaki eğilimlerin hesaplanmasıyla elde edilir. Bu görüntü, ana görüntünün düşük çözünürlüklü (öz/yaklaşım) halidir.

ya^1 alt görüntüsü, ana görüntünün yatay değişimlerini/ayrıntılarını gösterir.

da^1 alt görüntüsü, ana görüntünün dikey değişimlerini gösterir.

ka^1 alt görüntüsü, ana görüntünün köşegensel değişimlerini gösterir.

Bir sonraki adımda; $\frac{M}{2} \times \frac{N}{2}$ boyutundaki ym^1 öz/yaklaşım alt görüntüsü, analiz filtre setine uygulanarak $\frac{M}{4} \times \frac{N}{4}$ boyutunda dört alt görüntü daha elde edilir ve bu şekilde iterasyon sürdürülür.

$$ym^1 \rightarrow \begin{Bmatrix} ym^2 & | & da^2 \\ - & & - \\ ya^2 & | & ka^2 \end{Bmatrix} \tag{3.14}$$

Şekil 3.6'da eşitlik (3.13) ve eşitlik (3.14)'te belirtilen matris yapısına uygun parmakizi görüntüsü ve bu görüntüye ait birinci ve ikinci seviye ayrışımlar görülmektedir. Öz ve ayrıntılardan görüntünün tekrar sentezlenmesi, ayrışım algoritmasındaki işlemlerin ters sırada uygulanmasıyla yapılır.

Herhangi bir gürültülü görüntü aşağıdaki eşitlikle ifade edilebilir:

$$sg(i,j) = s(i,j) + \sigma.e(i,j) \tag{3.15}$$

Burada; s orijinal görüntüyü, $e(n)$ Gaussian beyaz gürültüyü ve σ de gürültü seviyesini göstermektedir. sg görüntüsündeki $\sigma.e$ gürültüsünü bastırıp s görüntüsünü elde etmeyi amaçlayan gürültü temizleme işlemi, üç adımdan oluşur (Vatansever ve ark. 2002):

I. Ayrışım: sg görüntüsü; seçilen dalgacık ile N. seviyeye kadar ayrıştırılır.

II. Ayrıntı katsayılarını eşikleme: $1-N$ arasındaki her bir seviye için ayrıntı katsayıları (ayrıntı alt görüntüleri) eşiklenir. t eşikleme noktası olmak üzere, $f(x)$ işareti için eşiklemeler:

$$\text{Sert: } \begin{cases} |x| > t \Rightarrow f(x) = x \\ |x| \le t \Rightarrow f(x) = 0 \end{cases} \tag{3.16}$$

$$\text{Yumuşak: } \begin{cases} |x| > t \Rightarrow f(x) = Sgn(x)(|x| - t) \\ |x| \le t \Rightarrow f(x) = 0 \end{cases} \tag{3.17}$$

III. Sentez: N. seviyedeki yaklaşım alt görüntüsü ile $1-N$ seviyeleri arasındaki eşiklenmiş ayrıntı alt görüntülerinden, yeni görüntü sentezlenir.

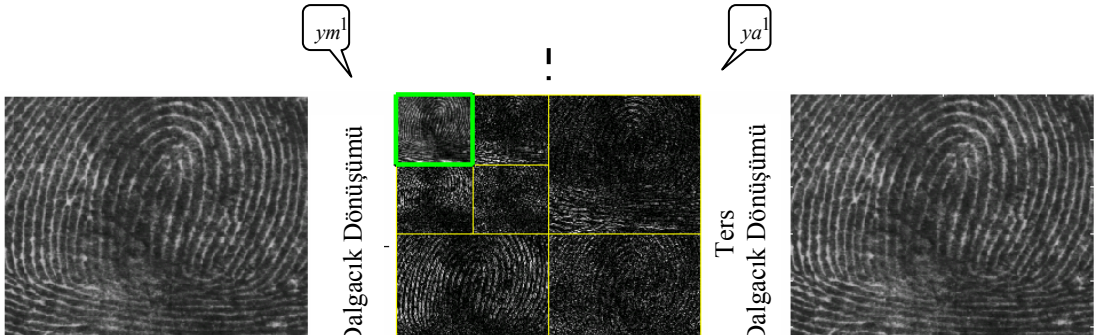
Verilen işlem adımları izlenerek temizlenen bir görüntü için, aşağıdaki parametreler tanımlanır (Walker 1999):

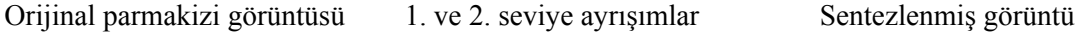
2-norm arasındaki göreceli fark ise aşağıdaki formda hesaplanır:

$$\Delta(s,sg) = \frac{\sqrt{(s_{11} - sg_{11})^2 + \ldots + (s_{MN} - sg_{MN})^2}}{\sqrt{s_{11}^2 + s_{12}^2 + \ldots + s_{MN}^2}} \tag{3.18}$$

Son olarak gürültülü görüntünün iyileştirme oranını gösteren veri olarak işaret-gürültü oranı (SNR) ise aşağıdaki formül kullanılarak belirlenir:

$$SNR = 20 \times \log_{10} \frac{1}{\Delta(s,sg)} \tag{3.19}$$



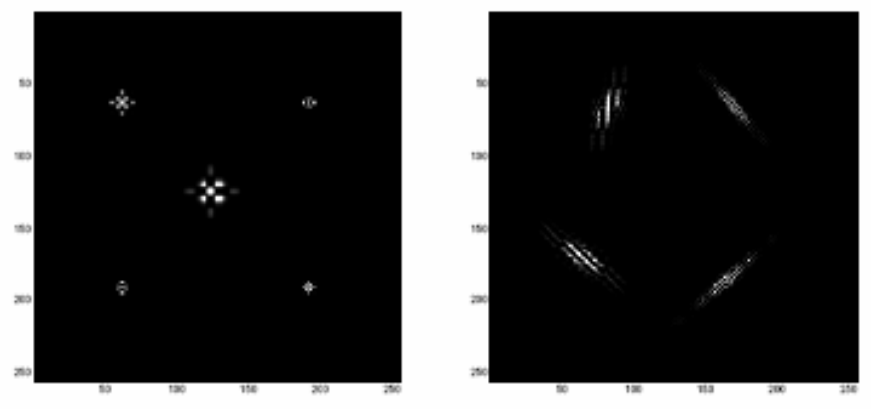


Şekil 3.6 Dalgacık dönüşümü uygulanmış parmakizi görüntüsü

3.4.2. Çevritsel dönüşüm ile parmakizi görüntüsü iyileştirme

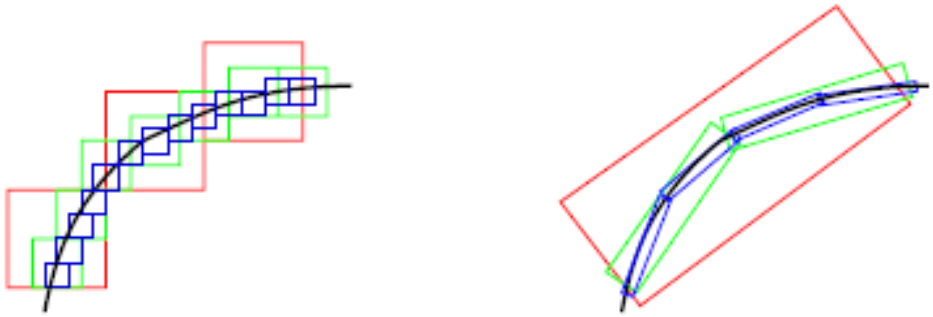
Çevritsel dönüşüm (contourlet transform), çok ölçekli ve yönsel filtre kümeleri kullanılarak iki boyutlu dalgacık dönüşümünün geliştirilmişidir. Çevritsel dönüşüm, esnek olarak çoklu ölçeklemelerde çeşitli yönlere yönlendirilmiş temel görüntülerin düzenlenmiş durumudur. Bu zengin temel görüntü kümeleri ile çevritsel dönüşüm, doğal görüntüdeki baskın ayrıntılara sahip olan pürüzsüz konturları etkili bir şekilde elde eder. İki boyutlu dalgacık dönüşümü, yönselliğe gereksinim duyar ve bununla süreksiz noktalar yakalanmasına rağmen konturların geometrik düzgünlüğü elde edilemez. Bu durum Şekil 3.7'de gösterilmiştir. Çevritsel dönüşüm bu eksikliği ortadan kaldırmak için geliştirilmiştir. Özellikle bu dönüşüm esnek olduğundan dolayı dalgacık dönüşümünden daha az katsayı ile ifade edilebilmektedir (Duncan ve ark. 2004).

Şekil 3.8'den görüleceği üzere dalgacık dönüşümleri, sadece süreksiz noktaları yakalayabilen karesel dönüşümlerdir. Çevritsel dönüşüm ise konturların lineer bölümlerini yakalayabilen genişletilmiş dönüşümlerdir ve bu yüzden de daha az katsayı ile etkili bir şekilde pürüzsüz bir konturu simgeleyebilir.



a) b)

Şekil 3.7 a) İki boyutlu dalgacık dönüşümünün ve b) çevritsel dönüşümün temel fonksiyon sonuçları



(a) (b)

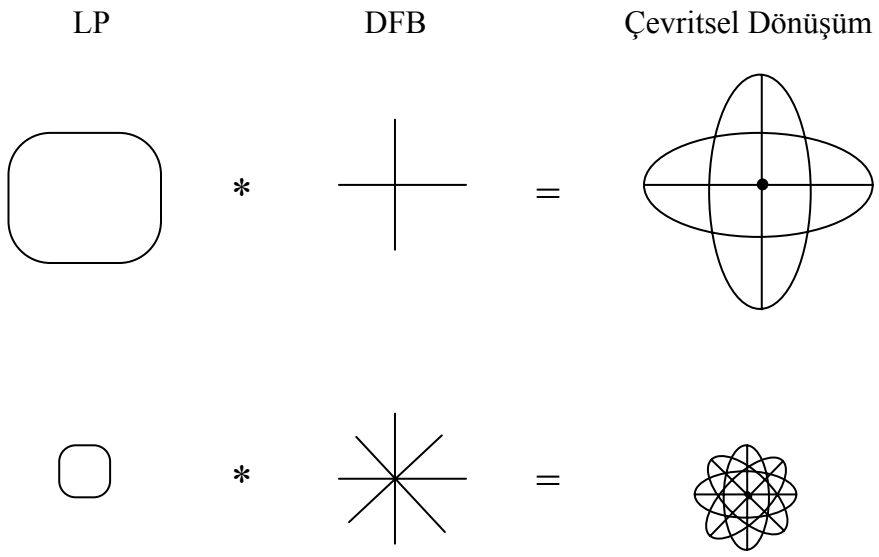
Şekil 3.8 a) Dalgacık dönüşüme ait ve b) çevritsel dönüşüme ait görüntü yakalama performansları

Çevritsel dönüşümün uygulamasında yönsel filtre kümesi (DFB) ve Laplas piramidi (LP) kullanılır. DFB, görüntünün yüksek frekans bileşenlerini yakalamak için tasarlanmıştır. Bundan dolayı küçük frekanslı bileşenler DFB tarafından zayıf olarak elde edilir (Do ve Vetterli 2005). Bu durumu düzeltmek için düşük frekanslar DFB işleminden önce çıkarılmalıdır. Bunun için LP, bant geçiren görüntüde uygulanarak daha ileri bir alt-bant ayrıştırmasına izin verir. Yönsel bilgi yeterli şekilde yakalanabildiği için bu bant geçiren görüntüler DFB içerisine gömülebilir.

Sonuç, çok yönlü ölçeklerde yönsel alt-bantlardaki görüntüleri ayrıştıran piramidal yönsel filtre kümesi (PDFB) olarak adlandırılan iki kat yinelemeli filtre kümesidir (Meyer 1992).

Çevritsel dönüşüm, çok yönlü ölçeklemelerde yönsel alt bantlardaki görüntüleri ayrıştıran piramidal yönsel filtre kümesinden geçerek uygulanır. Şekil 3.9, her ölçekteki yönsel filtre kümesini ve PDFB'nin bir basamağı olan Laplas piramidini gösterir (Meyer 1992). Şekil 3.9'da üst sıradan alt sıraya doğru yön sayısı iki kat artmasına rağmen ölçek dört kat azalmıştır.

Gaussian gürültüsü eklenmiş parmakizi görüntüsüne dalgacık dönüşümü ve çevritsel dönüşüm uygulanmıştır. Elde edilen parmakizi görüntüleri Şekil 3.10'da gösterilmiştir. Uygulanan dönüşümler sonucunda dalgacık dönüşümü için en büyük sinyal-gürültü oranı (PSNR) 25,2178 db olarak tespit edilmesine rağmen çevritsel dönüşüm için bu oran 28,6581 db olarak belirlenmiştir.



Şekil 3.9 Eğri ölçekleme ilişkisini sağlayan çevritsel tabanlı görüntülerin gösterilmesi

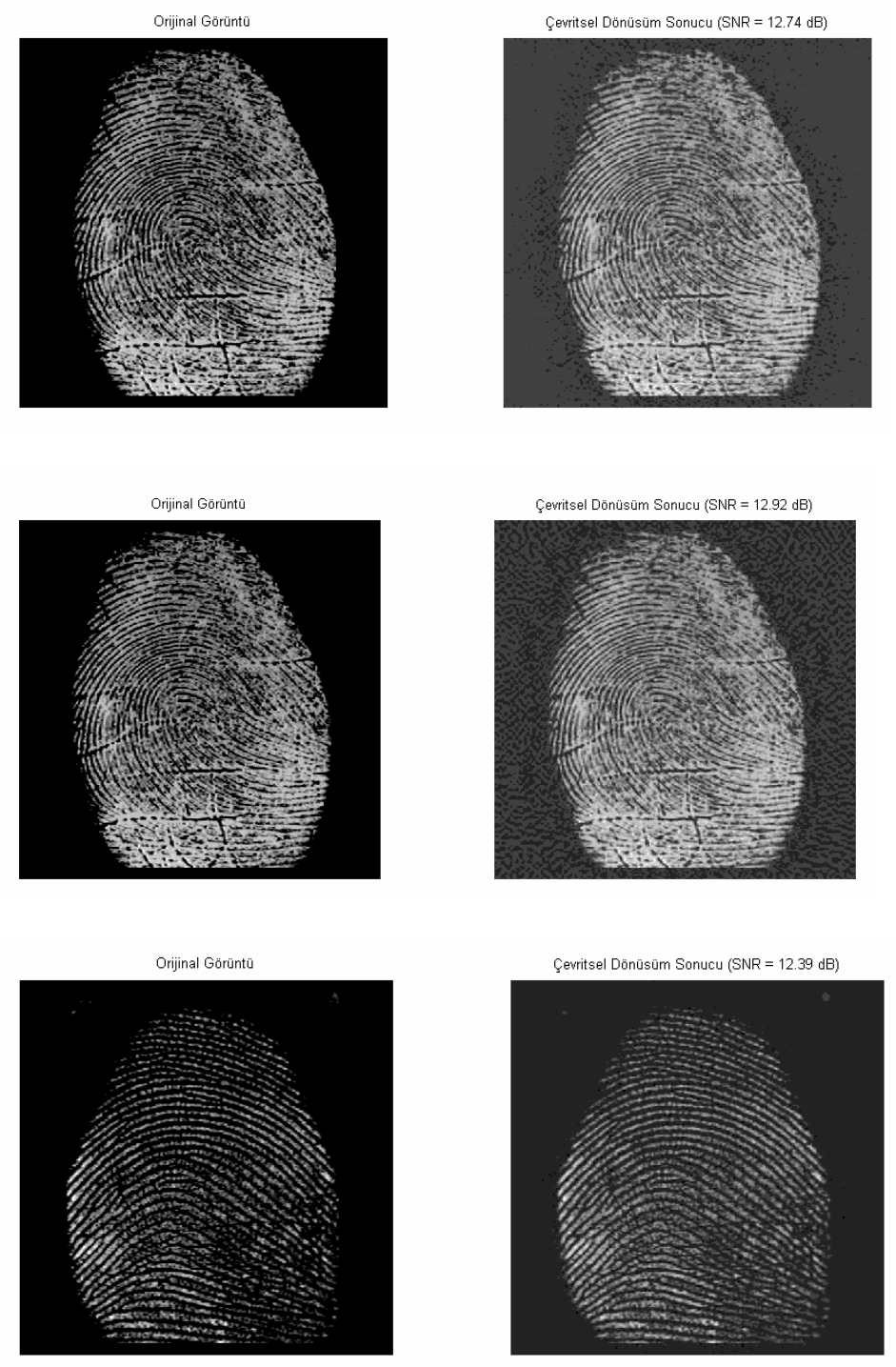
Parmakizinin kavisli yapıya sahip olması ve çevritsel dönüşümünün kavisli yapıya sahip görüntülere uygun olması dolayısı ile bu dönüşüm parmakizi görüntüsündeki gürültü azaltımında dalgacık dönüşümüne göre daha iyi performans göstermiştir (Altun ve ark. 2005). NIST-4 parmakizi veritabanında bulunan farklı parmakizlerinden bazılarına gürültü eklemeksizin çevritsel dönüşüm uygulanarak iyileştirilmiş parmakizi görüntüleri Şekil 3.11'de gösterilmektedir.

Ayrıca Şekil 3.12'de aynı parmakizi veritabanında bulunan bazı parmakizi görüntülerine Gaussian gürültüsü ekledikten sonra filtreleme teknikleri ve dalgacık dönüşümü ve çevritsel dönüşüm kullanılarak elde edilmiş en büyük sinyal-gürültü oranları gösterilmiştir.

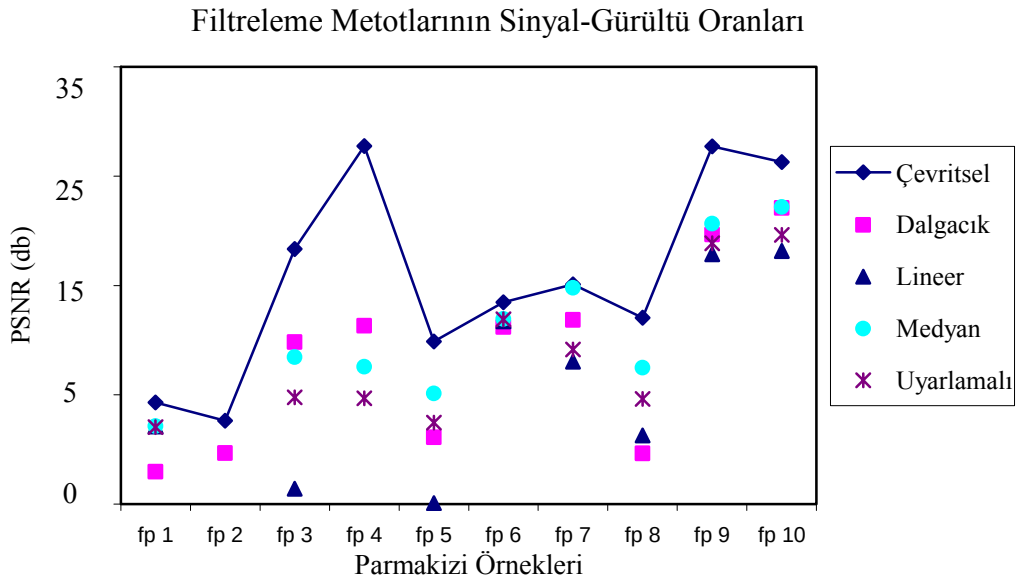


a) b)

Şekil 3.10 a) Dalgacık dönüşüm sonucu elde edilen parmakizi görüntüsü, b) Çevritsel dönüşüm uygulanması sonucu elde edilmiş parmakizi görüntüsü



Şekil 3.11 NIST-4 parmakizi veritabanında bulunan bazı parmakizlerine çevritsel dönüşüm uygulanarak iyileştirilmiş parmakizleri



Şekil 3.12 Farklı parmakizi görüntülerine uygulanmış filtreleme metotları sonucu elde edilmiş PSNR değerleri

3.5. Sonuç

Parmakizi görüntüsündeki hat yapılarının kalitesi, parmakizinin tanınmasında önemli bir karakteristiktir. Parmakizi görüntüsü üzerine derideki değişimden ve parmakizi tarama cihazındaki düzgün olmayan dağılımdan, kir, nem ve yaralardan dolayı binen gürültü, görüntüde gri seviye kaymalarına ve hat yapılarındaki netliğin bozulmasına sebep olur. Bu yüzden görüntüdeki gürültüleri azaltmak ve hat yapılarını iyileştirmek amacı ile görüntü iyileştirme teknikleri kullanılır.

Bu yaklaşımla ilgili olarak bu çalışmada parmakizi görüntü iyileştirme işlemi için iki metot uygulanmıştır. Birinci metotta bölgesel histogram eşitleme ve gürültü azaltma filtreleri kullanılır. İkinci metotta dalgacık dönüşüm ve parmakizi görüntüsünün yapısına uygun olduğunu düşündüğümüz iki boyutlu dalgacık dönüşümün geliştirilmişi olan çevritsel dönüşüm geliştirilerek Gaussian gürültüsü eklenmiş parmakizi görüntüsüne uygulanmıştır.

İyileştirme tekniklerinin gürültülü görüntüyü iyileştirme performanslarının bir ölçüsü, gürültülü görüntü ile iyileştirilmiş görüntünün karşılaştırılmasıyla ölçülen sinyal-gürültü oranıdır. Bu tez çalışmasında iyileştirme performansı en büyük sinyal-gürültü oranı (PSNR) temel alınarak ölçülmüştür. PSNR değerinin en yüksek olarak elde edildiği metodun, görüntüyü en iyi şekilde iyileştirdiği görülmektedir. Gürültülü görüntünün PSNR değeri 12,18 db iken uygulanan dönüşümler sonucunda dalgacık dönüşümü için 25,2178 db ve çevritsel dönüşüm için bu oran 28,6581 db olarak elde edilmiştir.

Parmakizinin kavisli yapıya sahip olması ve çevritsel dönüşümünün kavisli yapılı görüntülere uygun olması dolayısı ile çevritsel dönüşüm parmakizi görüntüsündeki gürültü azaltımında dalgacık dönüşümüne göre daha iyi performans gösterdiği gözlemlenmiştir. Bu, parmakizi tanıma aşamasında yapılan uygulamalar sonucunda da gözlemlenmiştir. Böylece, bir sonraki bölümde de görüleceği üzere YSA ile parmakizi tanıma aşamasından önce parmakizi görüntülerinin ön iyileştirilmesi, parmakizi tanıma performansını etkilediği görülmüştür.

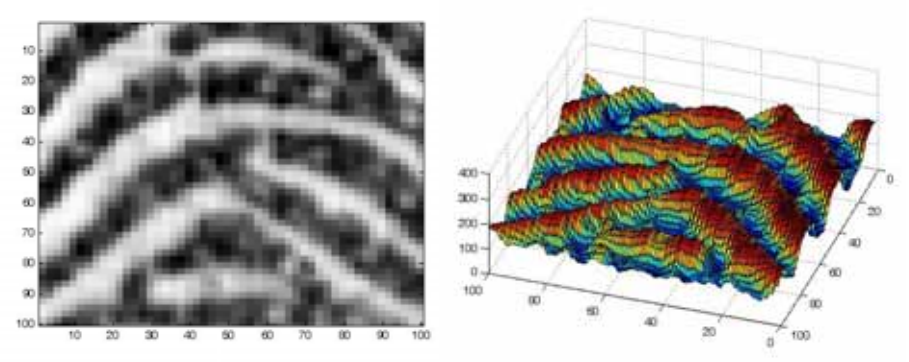
4. PARMAKİZİ ÖZELLİK VEKTÖRÜ

Genelde parmakizinde iki ana tip ayrıntı mevcuttur: (i) parmakizinin merkezi bölgesinde özel bir desen şeklinde olan ayrıntılı kabartılar ve iz yapısı ve (ii) sınırlı kabartı ve izlerle ilişkilendirilmiş özellik detaylarıdır. Bir parmakizi, birinci tip özelliğe göre sınıflandırılır ve ikinci tip özelliğe göre eşsiz olduğu tanımlanır (Prabhakar 2001). Bu özellik tabanlı gösterim, görsel olarak parmakizlerini eşleştiren adli uzmanlar tarafından uzun bir geçmişten beri popüler bir şekilde kullanılmaktadır. Ayrıca tanımlama amacıyla özellik çiftleri arasındaki kabartı sayısı da kullanılır. Bununla birlikte parmakizini otomatik eşlemede, parmakizleri arasındaki benzerliği elde etmek için kartezyen koordinatları ve Euclidean mesafesi kullanılır (Sha ve ark. 2003). Yapay zekâ yöntemlerinin gelişmesiyle birlikte parmakizini sınıflandırma ve tanıma işlemleri için YSA da kullanılır hale gelmiştir.

Şekil 4.1(a) ve (b)'den de görüldüğü üzere parmakizindeki kabartı ve vadilerin düzgün akışlı deseni, yönlü doku alanı olarak görülebilir (Graselli 1969). Parmakizi görüntülerindeki görüntü yoğunluk yüzeyi, yönlü bir doku oluşturan ve yönleri sürekli bir şekilde değişen kabartılardan oluşur. Karşılıklı belirgin dokular, bunların baskın frekanslarında önemli bir şekilde farklılıklar gösterir (Hong ve Jain 1999, Hong ve ark. 1999). 500 dpi'da taranmış bir parmakizi görüntüsü için farklı parmakizleri arasında kabartılar arası mesafe ile belirlenmiş uzaysal frekanslarda çok küçük değişimler mevcuttur. Bu çözünürlük, parmakizi dokusunu analiz etmek için uygun bir ölçektir. Bu çalışmada kullanılan parmakizi örneklerinin çözünürlüğü de 500 dpi olarak belirlenmiştir.

Yönlü dokuların analizi ve modellemesi, pratik uygulamalarda önemli bir araştırma sorunudur (Graselli 1969). Daha önce yapılan çalışmalarda yönlü dokuyu tanımlamak için ya yalnızca sınırlı ya da çoğunlukla ayrıntılı özellikler kullanılmıştır. Sınırlı gösterime Poincaré indeksi, sarmal sayısı ve anomaliler ile ilgili bilgi sıralanabilir (Prabhakar 2001). Yönlü bir dokunun gösterimine bölgesel yaklaşımların temel sınırlaması, bütün ayrıma dayalı bilgiyi etkili bir şekilde yakalayamamasıdır. Sınırlı bilgi kararsız ve gürültü eğilimli olabilir. Ayrıntılı

gösterimin örnekleri olarak yönsellik matrisi, yönelim alanları ve otomatik ilişki metotları sayılabilir. Jain ve Farrokhnia (1991), gabor filtre kümesini kullanarak farklı frekans ve yönelim bileşenleri içerisindeki giriş görüntüsünü ayrıştırarak dokunun ayrıntılı gösterimini elde etmişlerdir. Bu gösterimi, dokulu görüntüyü başarılı bir şekilde sınıflandırmada ve kısımlara ayırmada kullanmışlardır. Verimli ve hızlı olmasına rağmen ayrıntılı gösterim, parmakizine ait tüm ayırt edici bilgileri elde edemeyebilir. Şekil 4.2'de gösterilen iki parmakizinin ayrıntılı konfigürasyonu benzer olmasına rağmen bölgesel anomalilerin farklı görünümünden dolayı bu izlerin farklı olduğu gözlemlenir.



a) b)

Şekil 4.1 Parmakizi görüntüsünün akış deseni. a) Parmakizi görüntüsünün bir bölümü, b) Parmakizi görüntüsünün bir bölümüne ait 3-boyutlu görüntüsü



Şekil 4.2 Benzer özelliklere sahip olmalarına rağmen iki farklı parmakizi görüntüsü

Daugman (1993) gözle görülebilir iris dokusunu, çok ölçekli dörtlü Gabor dalgacık katsayılarının sıralı bir dizisi ile Iriscode olarak adlandırılan dönmeden ve ölçeklemeden bağımsız bir gösterimini elde etmiştir. Daugman'ın iris doku gösterimi döndürmeden bağımsızdır. Zaten insan gözünün irisinde büyük dönmeler, başın sınırlı hareketinden dolayı meydana gelemez. Küçük döndürmeler olursa, eşleştirme işlemi esnasında Iriscode'un kendi içerisinde döndürülmesiyle meydana gelen değişiklik eşleştirmeye imkân verir. Bu tez çalışmasında parmakizinin yönlü doku gösterimi için Daugman (1999)'ın iris tanımasından ve Jain ve Farrokhnia (1991)'nın Gabor filtre kümesinden esinlenilmiştir. Çalışmanın temelini oluşturan bu yapının ilk aşamasında öncelikle değişmez bir referans noktası elde edilir. Bu referans nokta etrafında belirlenmiş ilgili bölge hücrelere bölündükten sonra her bir hücre için farklı bir veya daha fazla açısal dönüşüme göre özellikler elde edilir. Her bir hücre için elde edilen özelliklerin sıralı bir dizisi, parmakizinin simgelenişi olarak kullanılır. Parmakizinin özellik vektörünün elde edilmesi aşamaları Şekil 4.3'te gösterilmektedir.

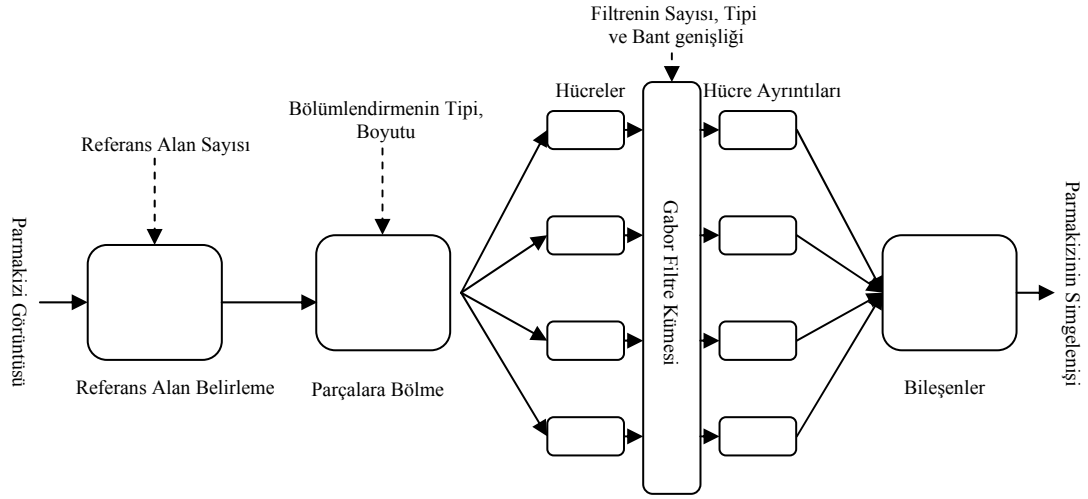
Gabor filtre kümesi, uzaysal frekansa göre parmakizi bilgisini dikey bileşenlerine ayrıştırması kadar, özel bant geçiren kanallardaki faydalı bilgiyi yakalamada da en iyi bilinen tekniklerden birisidir (Prabhakar 2001). Bu çalışmada parmakizi görüntüsüne ait özelliklerin çıkarım algoritmasının 4 temel adımı aşağıdaki gibi sıralanır (Altun ve ark. 2005a):

1. Parmakizi görüntüsündeki referans noktası tayin edilmiştir.

2. Referans noktası etrafındaki bölge parçalara ayrılmıştır.

3. Gabor filtre kümesi kullanılarak elde edilen bölge sekiz farklı yönde filtrelenir. Sekiz yön, parmakizindeki sınırlı ayrıntı karakteristiklerini tam olarak elde etmede dört yön uygulanmasından daha etkilidir (Jain ve ark. 1999d). Bu çalışmada 0^o'den başlayarak $22,5^o$ derece artırımla $157,5^o$'ye kadar Gabor filtre uygulanarak özellik vektörü elde edilme aşamasına geçilmiştir..

4. Daugman (1993) tarafından elde edilen IrisCode'una benzer FingerCode (Jain ve ark. 1999b) olarak adlandırılan ve parmakizinin özellik vektörünü belirleyen bu simgeleniş, filtrelenmiş görüntünün her bir sektöründeki gri değerlerin ortalamasından ortalama mutlak hatanın hesaplanmasıyla elde edilmiştir.

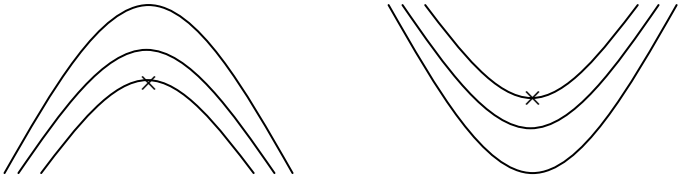


Şekil 4.3 Parmakizleri için doku-tabanlı simgelenişinin çıkarım şeması

4.1. Parmakizi Referans Noktasının Bulunması

Parmakizleri, dikkat çeken birçok sınır işaretlerine sahiptir ve bunların kombinasyonu bir referans noktasını tespit etmek için kullanılabilir. Parmakizindeki referans noktası, parmakizi görüntüsündeki içbükey hatların maksimum eğrilik noktası olarak tanımlanır (Şekil 4.4).

Referans noktası (x_c,y_c)'nin tayini için yapılan önceki yaklaşımlar, Poincaré indeks (Cappelli ve ark. 2001a,b) gibi sınırlı ayrıntılara veya yönelim alanının diğer benzer özelliklerine bağlıdır. Bu metotlar kaliteli parmakizi görüntüleri için iyi sonuç verse de çatlak, yara izi gibi kötü hatlara sahip parmakizlerinin referans nokta tayininde başarısız olmaktadırlar. Hong ve Jain (1996), parmakizi sınıflandırması için mevcut hat detayları ile yönelim alan bilgisinin mantıklı bir şekilde birleştirilmesini araştırmıştır. Fakat yönelim alanı çok gürültülü olduğundan ve parmağın zayıf yapıdaki derisi üzerindeki bere ve kesikten dolayı referans noktası elde edilemediği için bunların metodu iyi kalitede olmayan parmakizlerinde sağlıklı sonuç vermemiştir.



İçbükey hatlar Dışbükey hatlar

Şekil 4.4 Parmakizi görüntüsündeki içbükey ve dışbükey hatlar (Referans noktası × ile gösterilmiştir)

Bir referans nokta algoritmasının düşük kalitedeki parmakizleri üzerindeki bölgesel gürültüyü elde edebilmesi için parmakizi görüntüsündeki geniş bir bölgede komşulukların dikkate alınması gereklidir (Ross 2003). Yani referans noktasının doğru lokalizasyonu için kullanılan yöntem, küçük komşuluklardaki bölgesel değişimlere duyarlı olmalıdır. Bundan dolayı, Hong ve ark. (1998)'nın önerdiği algoritmadan daha duyarlı ve esnek bir sistem geliştirilmiştir.

$M \times N$ boyutlu bir parmakizi görüntüsü F ve yönelim alanı O olmak üzere $P \leq M$, $Q \leq N$ için (i, j) pikselindeki bölgesel hat yönelimi $O(i, j)$ ile ifade edilir ve $P \times Q$ görüntüsü olarak tanımlanır. Bölgesel hat yönelimi, F görüntüsündeki her bir pikselden ziyade genellikle bir blok için belirtilir. Bu yüzden, Şekil 4.5(a)'da da görüldüğü üzere parmakizi görüntüsü örtüşmeyen $w \times w$ boyutlu bloklara bölünür ve $P = \left[\frac{M}{w} \right]$, $Q = \left[\frac{N}{w} \right]$ olmak üzere her bir blok için tek bir yönelim tanımlanır.

Parmakizi yönelimi, π'nin bir faktörü için belirsizdir. Mesela $\frac{\pi}{2}$'de yönlü bölgesel hatlar ve $\frac{3\pi}{2}$'de yönlü bölgesel hatlar, birbirlerinden farklı olmayabilirler. Bu şekildeki metotlar (Rao 1990, Kass ve Witkin 1987, Kawagoe ve Tojo 1984, Chang 1980) parmakizi görüntüsündeki yönelim alanını belirlemek için geliştirilmiştir.

Aşağıdaki algoritma en küçük ortalama kare yönelimini (Hong ve ark. 1998) hesaplamada kullanılır:

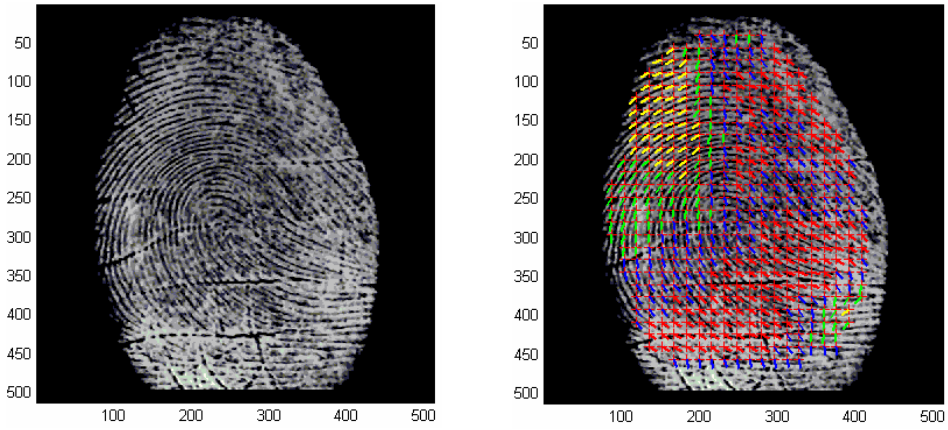
1. Parmakizi görüntüsü F, örtüşmeyen $w \times w$ boyutlu bloklara bölünür.
2. Her bir (i, j) pikselinde $\partial_x(i, j)$ ve $\partial_y(i, j)$ eğimleri bulunur. Hesaplamaya bağlı olarak basit Sobel operatörlerinden karmaşık Marr-Hildreth (1980) operatörlerine kadar birçok eğim operatörü kullanılabilir.
3. Aşağıdaki eşitlik kullanılarak (i, j) merkez olmak üzere her bir blok için bölgesel yönelim hesaplanır (Rao 1990):

$$V_x(i,j) = \sum_{u=i-\frac{w}{2}}^{i+\frac{w}{2}} \sum_{v=j-\frac{w}{2}}^{j+\frac{w}{2}} 2\partial_x(u,v)\partial_y(u,v) \tag{4.1}$$

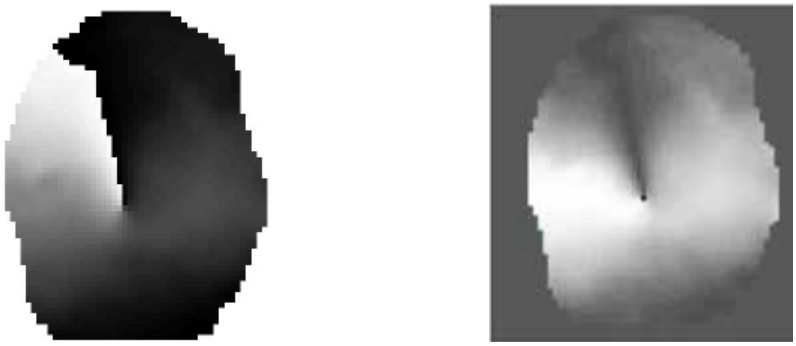
$$V_y(i,j) = \sum_{u=i-\frac{w}{2}}^{i+\frac{w}{2}} \sum_{v=j-\frac{w}{2}}^{j+\frac{w}{2}} (\partial_x^2(u,v) - \partial_y^2(u,v)) \tag{4.2}$$

$$O(i,j) = \frac{1}{2}\tan^{-1}\left(\frac{V_y(i,j)}{V_x(i,j)}\right) \tag{4.3}$$

Burada $O(i, j)$, (i, j) merkezli bloğun bölgesel hat yöneliminin en küçük kareler hesabıdır. Matematiksel olarak bu, $w \times w$ boyutlu pencerenin Fourier spektrumunun baskın yönüne dik olan yön olarak ifade edilir (Şekil 4.5). Şekil 4.5(d)'de görüntünün merkezindeki en koyu piksel, referans nokta olarak belirlenmiştir. Parmakizini görüntülemek için 0 - 255 aralığına ölçeklendirilmiştir.



a) b)



c) d)

Şekil 4.5 Referans noktasının belirlenmesi. a) Orijinal parmakizi görüntüsü, b) Parmakizi görüntüsü üzerine bindirilmiş yönelim alanı, c) $w = 10$ için yoğunluk dağılımı ve arka plandaki gri seviyenin bölümlenmesi, d) Yönelim alanının sinüs bileşeni

Referans nokta tayininde kullanılan algoritma (Prabhakar 2001), aşağıdaki şekilde geliştirilerek ifade edilir:

1. $w \times w$ boyutlu pencere kullanılarak yukarıda elde edilen O yönelim alanı hesaplanır.

2. Bölgesel komşuluklardaki yönelim alanı düzeltilir. Düzeltilen yönelim alanı O' olarak ifade edilir. Alçak-geçiren filtre kullanılarak düzeltmeyi gerçekleştirmek için yönelim görüntüsü aşağıdaki formül ile sürekli vektör alanına dönüştürülür:

$$\Phi_x(i, j) = \cos(2O(i, j)) \qquad (4.4)$$

$$\Phi_y(i,j) = \sin(2O(i,j)) \tag{4.5}$$

Burada Φ_x ve Φ_y, vektör alanının x ve y bileşenleridir. Elde edilen vektör alanının alçak geçiren filtreden geçirilmesi işlemi aşağıdaki gibidir:

$$\Phi'_x(i,j) = \sum_{u=-w_\Phi/2}^{w_\Phi/2} \sum_{v=-w_\Phi/2}^{w_\Phi/2} W(u,v)\Phi_x(i-uv, j-vw) \tag{4.6}$$

$$\Phi'_y(i,j) = \sum_{u=-w_\Phi/2}^{w_\Phi/2} \sum_{v=-w_\Phi/2}^{w_\Phi/2} W(u,v)\Phi_y(i-uv, j-vw) \tag{4.7}$$

Burada W, $w_\Phi \times w_\Phi$ boyutlu pencereye ve birim parçaya sahip alçak geçiren filtredir. Bu çalışmada 5×5 boyutlu blok ile ortalama filtre kullanılmıştır. (i,j) pikselinde düzeltilen yönelim alanı O', aşağıdaki formülle hesaplanır:

$$O'(i,j) = \frac{1}{2}\tan^{-1}\left(\frac{\Phi'_y(i,j)}{\Phi'_x(i,j)}\right) \tag{4.8}$$

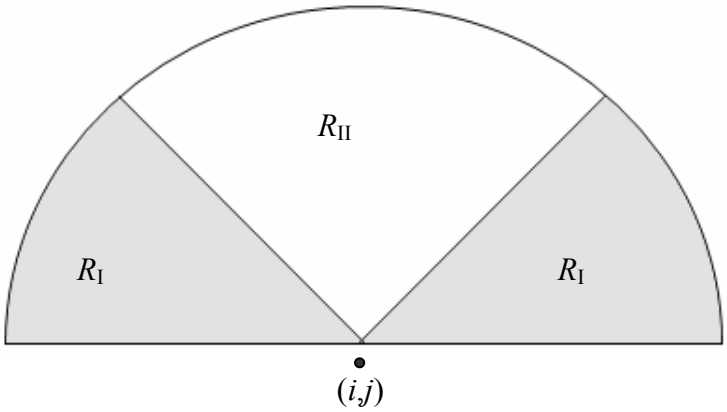
3. O'''ın yalnızca sinüs bileşenini içeren bir görüntü olarak ε hesaplanır:

$$\varepsilon(i,j) = \sin(O'(i,j)) \tag{4.9}$$

4. ε'daki her bir (i,j) pikseli için Şekil 4.6'da gösterilen R_I ve R_{II} bölgelerindeki piksel yoğunluğu hesaplanır ve bunların fark değeri, referans noktası $\grave{A}$'daki uygun piksel değerleri olarak bulunur:

$$\grave{A}(i,j) = \sum_{R_I} \varepsilon(i,j) - \sum_{R_{II}} \varepsilon(i,j) \tag{4.10}$$

Şekil 4.6'da gösterilen R_I ve R_{II} bölgeleri, geniş bir veritabanı üzerinde referans nokta tayini algoritmasının uygulanması ile elde edilir. Yarım dairesel bölgenin yarıçapı, w pencere boyutuna eşittir. R_I ve R_{II} bölgelerinin geometrisi, Şekil 4.4'te gösterilen içbükey hatlardaki maksimum eğriliği yakalamak için tasarlanır. Bu yaklaşım ile çift kementli parmakizlerindeki referans noktası çoğu durumda başarılı bir şekilde elde edilmesine rağmen (Şekil 4.7(a)), kemer tip parmakizlerinde kesin ve tutarlı değildir. Çünkü bu parmakizlerinde yüksek eğriliğin olduğu noktaların yerlerini belirlemek zordur (Prabhakar ve Jain 2002).

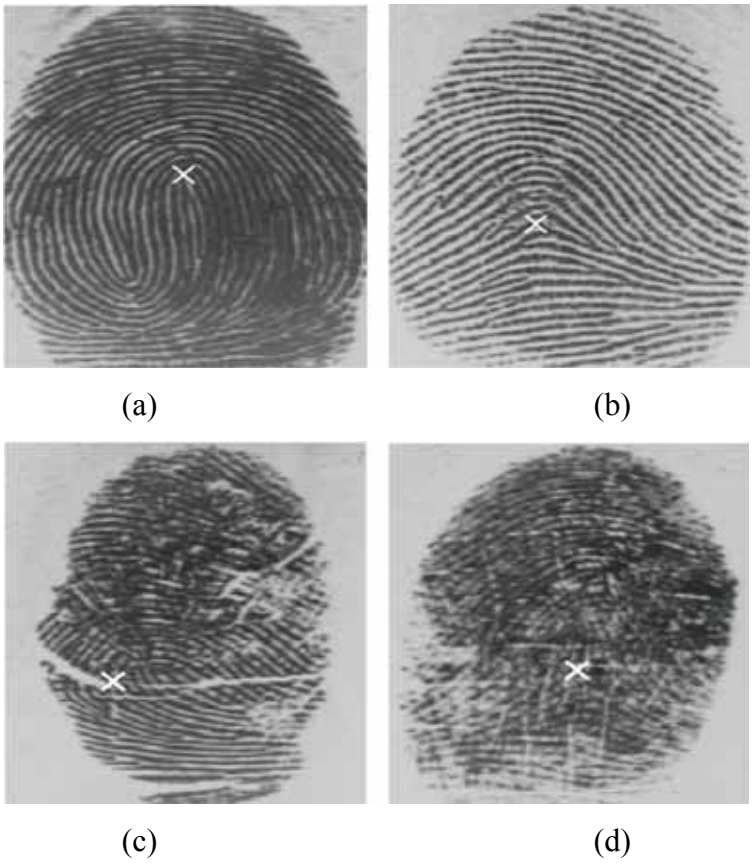


Şekil 4.6 $\grave{A}(i,j)$'yi hesaplamak için ε'daki piksel yoğunluklarını hesaplama bölgeleri

5. $\grave{A}$'nın maksimum değeri bulunur ve bu koordinat noktası referans noktası olarak işaretlenir.

6. $\hat{w}<w$ olmak üzere $\hat{w}\times\hat{w}$ pencere boyutu kullanılarak işlem adımları belirli defa tekrar uygulanır ve 5. adımda belirlenen referans noktasının bölgesel komşuluğundaki referans noktası için arama sınırlandırılır. Genelde deneysel tecrübelerde göz önünde bulundurulduğunda, bu tez çalışmasında da w = 15, 10 ve 5 piksel olarak seçilmiş ve böylece referans nokta tayininin hassasiyeti de 5 piksel olarak belirlenmiştir.

Şekil 4.7'de dört farklı parmakizi görüntüsü için referans noktaları elde edilmiştir. Bu yöntem ile sarmal, sol kement, sağ kement ve kemer tip parmakizleri için iyi sonuçlar elde edilmiştir. Kemer tip parmakizlerinde tek bir referans noktası olmamasından dolayı bu parmakizindeki referans nokta tayininde hatalar olabilmektedir. Bu tür parmakizlerinde de delta noktasında belirlenmiş referans noktaları göz ardı edilmiştir. Ayrıca düşük kalitedeki parmakizleri için de yönelim alanının hatalı elde edilmesinden dolayı referans nokta tayininde hatalar oluşabilmektedir. Referans nokta tayinindeki en önemli unsurlardan birisi, parmakizi tipinin belirlendiği özellik olan sarmal, kement veya kemer gibi özelliklerin parmakizi görüntüsünde belirgin bir şekilde var olması, referans noktasının elde edilmesinde en önemli unsurdur.



(a) (b)

(c) (d)

Şekil 4.7 Referans nokta tayini algoritmasının sonuçları

Eğer parmakizi tipi herhangi bir sebeple belirlenemezse ve parmakizi görüntüsü bozuksa, referans noktasının elde edilme başarısı da o derece azalır. Bu çalışmada, parmakizleri için referans noktası büyük oranda elde edilmiş ve referans noktası tayin edilemeyen parmakizleri diğer çalışmalarda olduğu gibi göz ardı edilmiştir. Şekil 4.7 (a) ve (b)'de görüntülenen parmakizi görüntüleri çift sarmal ve çadır kement tipi parmakizleri olduğu açıkça görüldüğü için algoritma bu parmakizlerine ait referans nokta tayininde başarılı olmuştur. Buna karşın Şekil 4.7 (c) ve (d)'de görüntülenen parmakizi görüntüleri, düşük kalitede oldukları için hangi sınıfa dahil edilecekleri belirlenememiştir. Bu parmakizleri için referans nokta tayini algoritması da başarısız olmuştur. Böylece parmakizlerinin görüntü kalitesi, bunların referans nokta tayininde büyük rol oynamaktadır. Görüntü kalitesi düşük olan bir parmakizi için herhangi bir referans noktası tayin edilememekte, dolayısı ile özellik vektörü de çıkarılamadığı için parmakizi tanıma işlemi gerçekleştirilememektedir.

4.2. Parmakizinin Sektörizasyonu (Mozaikleme)

$M \times N$ boyutlu bir parmakizi görüntüsünün (x, y) piksel koordinatına ait gri seviye değeri $F(x, y)$ ve referans noktası (x_c, y_c) olarak tanımlansın. Parmakizindeki referans bölgesi, tüm S_i sektörlerinin koleksiyonu olarak tanımlanır. Burada S_i sektörü, referans nokta bloğunun açısal vektörü başlangıç noktası kabul edilerek aşağıdaki formülle hesaplanır:

$$S_i = \{(x,y) \mid b(T_{i+1}) \le r < b(T_{i+2}),\ \theta_i \le \theta < \theta_{i+1},\ 1 \le x \le N,\ 1 \le y \le M\} \quad (4.11)$$

Burada $T_i = i / k$ (4.12)

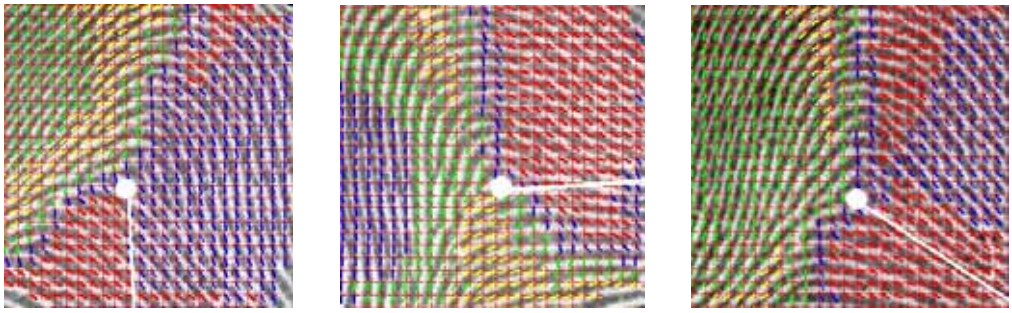
$$\theta_i = ((i \bmod k) \times (2\pi / k)) + \theta_0 \quad (4.13)$$

$$r = \sqrt{(x - x_c)^2 + (y - y_c)^2} \quad (4.14)$$

$$\theta = \tan^{-1}((y\text{-}y_c) / (x\text{-}x_c)) + \theta_0 \quad (4.15)$$

Burada θ_0, referans nokta bloğunun açısal değeri; b, her bir bandın genişliği; k, her bir bantta hesaba katılan sektör sayısı ve B, $i = 0,\ldots,(B \times k\text{-}1)$ olmak üzere özellik çıkarımı için referans noktası etrafında hesaba katılan eşmerkezli bantların sayısıdır (Altun ve ark. 2005a). Şekil 4.8'de orijinal parmakizi ve döndürülmüş parmakizi görüntüleri için elde edilen referans noktaya ait bloğun açısal değerine göre sektörizasyon işleminin başlangıç kabul edileceği yönler beyaz çizgi ile gösterilmiştir.

B parametresi, görüntülenen parmağın alanına bağlıdır. 500 dpi çözünürlükte 320x320 boyutlu görüntüden daha geniş bir alana sahip parmak görüntüsü 640x480 pikselde yakalanır. Bu yüzden B parametresi, görüntü boyutuna ve sensorun dpi çözünürlüğüne bağlıdır. Eşmerkezli bantların genişliği b parametresi tarafından tanımlanır. Bantların genişliği ortalama bir hat ve vadi çiftini yakalamalıdır. Bu yüzden 500 dpi çözünürlükte elde edilmiş bir parmakizi görüntüsü için b genelde 20 olarak seçilir (Şekil 4.9).



a) b) c)

Şekil 4.8 a) -45^{o} döndürülmüş, b) +45^{o} döndürülmüş ve c) Normal parmakizi görüntülerine ait sektörizasyon işleminin başlangıç yönleri



Şekil 4.9 İlgili bölgenin referans noktası (×), 80 sektör ($B = 5,\ k = 16$)

Bu çalışmada bant genişliği değeri olan *b* parametresi, bant sayısını gösteren *B* parametresine bağlı olarak değişik değerlerde seçilmiştir. 20 piksel genişliğinde bir bant, bir sektördeki tek bir özellikten oluşan bölgesel bir bilgiyi yakalamak için yeterlidir. Eğer piksel genişliği 20 pikselden büyük olursa sınırlı bilgi, daha ayrıntılı bilgiye dönüşür. Referans noktasının da bulunduğu en içteki bant, özellik çıkarımı için kullanılmaz, çünkü referans noktasına yakın bölgedeki sektörler çok az piksel içerirler. Bundan dolayı özellik çıkarımı bu bölgede güvenilir olmaz. Parçalara bölme (mozaikleme) işlemi dairesel olarak seçilebilir, çünkü parmakizinin döndürülmesi, ayrılmış parçaların döndürülmesine uygundur. *k* değeri, parmakizindeki sınırlı bilgiye karşı ayrıntılı özelliklerin elde edilmesini sağlar. Parmakizi sınıflandırma algoritması için daha ayrıntılı bir bilgi gerekli olduğundan daha küçük bir *k* değeri seçilir.

Parmakizi doğrulama uygulamaları ise parmakizindeki daha fazla bölgesel bilgiyi yakalama ihtiyacı duyduklarından dolayı daha yüksek bir *k* değeri seçilmelidir. Bundan dolayı *B*, *b* ve *k* parametreleri için değerler, parmakizi sınıflandırma ve eşleştirme uygulamaları için en iyi performansı elde edecek şekilde belirlenirler. Fingercode tabanlı sınıflandırma ve eşleştirme algoritmalarında, performansta belirgin bir bozulma olmaksızın bu parametrelerde küçük değişikliklere izin verilir. *B* değeri, parmakizi görüntüsünün büyük bir kısmını reddetmeksizin maksimum hat ve vadi detaylarını elde etmede kullanılır. *k* değeri, belirli bir uygulama için gereken ayrıntılı bilgi ile sınırlı bilgi arasındaki ilişkiye bağlı olarak seçilebilir. *b* değeri ise sensorların dpi çözünürlüğüne ve parmakizi görüntüsündeki hatlar arası ortalama mesafe göz önünde bulundurularak seçilir.

Parametre değerleri bir uygulama için seçildiğinde bu değerler sabit kalmalıdır. Bu tez çalışmasında her bir parmakizi görüntüsü için bant sayısı 3, 4, 5, 6, 7, 8 ve sektör sayısı da 8, 12, 16, 20, 24 olmak üzere toplam 30 farklı kombinasyondan oluşan sektörize edilmiş dairesel alan belirlenerek en uygun değerler araştırılmıştır. Burada bant genişlikleri, seçilen bant sayısına göre belirlenmiştir. Tablo 4.1'de seçilen bant genişlikleri belirtilmiştir. Parmakizi görüntülerinin seçilen bant ve sektör sayılarına göre sektörizasyonu sonucu elde edilen dairesel alanlardaki en az sektör sayısı, 3 bant ve 8 sektör olmak üzere 24 (3×8) ve en fazla sektör sayısı ise 8 bant ve 192 sektör olmak üzere 192 (8×24)'dir.

Tablo 4.1 Bant sayısına göre belirlenen bant genişlikleri

Bant sayısı	**Bant genişliği (piksel)**
3	24
4	18
5	15
6	12
7	11
8	9

4.3. Parmakizi Filtreleme

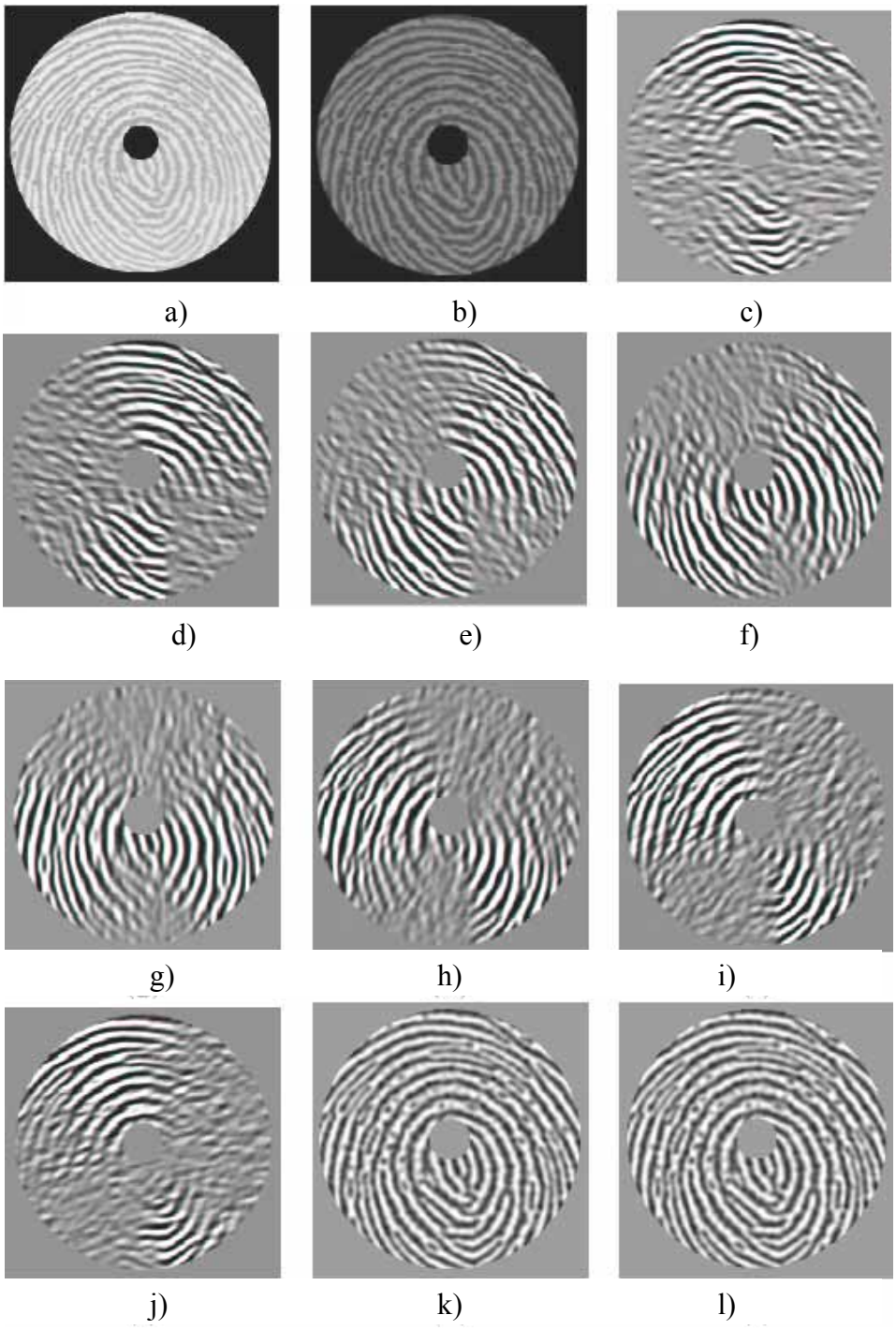
Parmakizleri, bölgesel paralel hatlara, vadilere ve iyi tanımlanmış bölgesel frekans ve yönelime sahiptirler (Şekil 4.10). Uygun bir şekilde ayarlanan Gabor filtreleri (FBI 1984, Bigun ve Granlund 1987, Bigun ve ark. 1997) parmakizlerinin doğru hat ve vadi yapılarını korumakla birlikte görüntüdeki gürültüyü ayıklar ve görüntüde belirli bir yönelimdeki bilginin elde edilmesini sağlar. Bir ayrıntı noktası, paralel hatlarda bozuk olarak görülebilir, bu yüzden bu bilgi Gabor filtre kullanılarak elde edilmeye çalışılır.

Parmakizi görüntüsü filtrelenmeden önce her bir sektördeki ilgili bölgede gri seviye yoğunluğu sabit bir ortalama ve varyans değerine normalize edilir. Normalizasyon işlemi, görüntüye karışan sensör gürültülerini ve parmağın basım farklılığından dolayı oluşan arka plan görüntüsünü çıkarmak için kullanılır. (x,y) pikselindeki gri seviye $F(x,y)$ olmak üzere, S_i sektöründeki gri seviyelerin ortalama ve varyans değeri sırasıyla M_i ve V_i olsun. (x,y) pikselindeki normalize edilmiş gri seviye değeri ise $N_i(x,y)$ olarak ifade edilsin (Ballan ve ark. 1997).

Buna göre S_i sektöründeki tüm pikseller için normalize edilmiş parmakizi görüntüsü aşağıdaki şekilde tanımlanır:

$$N(x,y)=\begin{cases} O_0+\sqrt{\dfrac{V_0\times(P(x,y)-O)^2}{V}} & , P(x,y)>M \\ O_0-\sqrt{\dfrac{V_0\times(P(x,y)-O)^2}{V}} & , \text{diğer durumlar} \end{cases} \tag{4.16}$$

Burada M_0 ve V_0 değerleri istenen ortalama ve varyans değerleridir. Normalizasyon işlemi, hat ve vadi yapısının belirgin bir şekilde değişmediği piksel tabanlı bir işlemdir. Bu işlem, görüntünün tamamında gerçekleştirilirse parmağın basım farklılığından dolayı görüntünün farklı bölgelerindeki yoğunluk varyasyonları dengelenemez (Jain ve ark. 2004). Her bir sektörün ayrı ayrı normalizasyonu bu problemi azaltır. Şekil 4.10, bu normalizasyon sonucunu gösterir. Bu çalışmada istenen M_o ortalama değeri ve V_o varyans değeri, 100 olarak belirlenmiştir.



Şekil 4.10 Normalize edilmiş, filtrelenmiş ve iyileştirilmiş parmakizi görüntüleri a) Parmakizi görüntüsüne ait elde edilmiş ilgili bölge, b) Normalize edilmiş görüntü, c)-j) 0^o, 22.5^o, 45^o, 90^o, 112.5^o, 157.5^o yönlerde filtrelenmiş görüntüler, k) Dört filtre ile iyileştirilmiş parmakizi görüntüsü, ve l) Sekiz filtre ile iyileştirilmiş parmakizi görüntüsü

Simetrik Gabor filtresi uzaysal düzlemde aşağıdaki formül ile hesaplanır:

$$G(x,y;f,\theta)=\exp\left\{\frac{-1}{2}\left[\frac{x'^2}{\delta_x^2}+\frac{y'^2}{\delta_y^2}\right]\right\}\cos(2\pi f x'), \tag{4.17}$$

$$x' = x\sin\theta + y\cos\theta \tag{4.18}$$

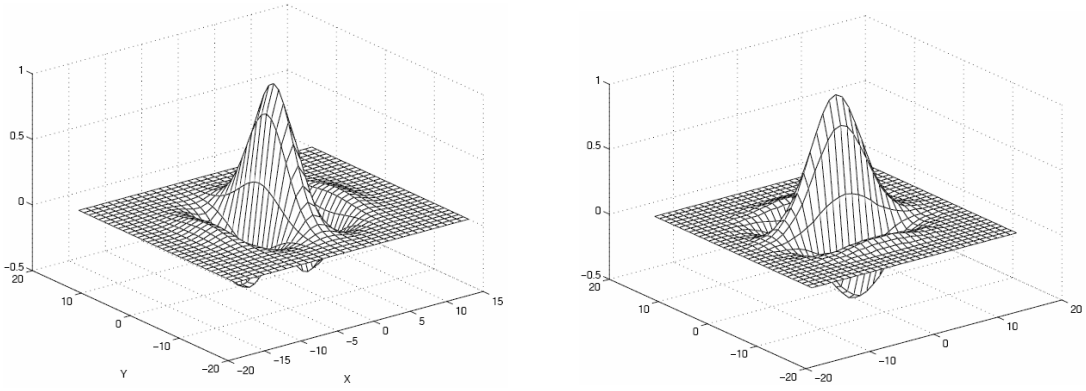
$$y' = x\cos\theta - y\sin\theta \tag{4.19}$$

Burada f, x ekseninden θ yönü boyunca uzanan sinüs dalgasının frekansıdır. δ_x ve δ_y ise sırasıyla x ve y eksenleri boyunca Gaussian dağılımı'nın mesafe sabitleridir (Almansa ve Lindeberg 1997, 2000). Gabor filtresinin uzaysal karakteristikleri Şekil 4.11'de gösterilmiştir.

Filtreleme işlemi 33×33 boyutlu bir maskeli uzaysal düzlemde gerçekleştirilir. Şekil 4.11'de 33×33 boyutlu maskenin dış kenarına yakın filtre değerlerinin sıfıra yakın olduğu görülmektedir.

Çalışmamızda filtre frekansı f, ortalama hat frekansı $(1/K)$'ya eşittir. Burada K değeri, hatlar-arası ortalama mesafedir. 500 dpi bir parmakizi görüntüsünde hatlar arası ortalama mesafe yaklaşık 10 pikseldir. f değeri çok büyük bir değer seçilirse filtrelenmiş görüntüde sahte hatlar oluşur. Benzer şekilde f değeri küçük bir değer seçilirse yakın hatlar birbiri içerisine karışır. x eksenine göre 0^o, 22.5^o, 45^o, 67.5^o, 90^o, 112.5^o, 135^o ve 157.5^o olarak farklı filtre yönleri uygulanabilir. Parmakizi görüntüsünde normalize edilmiş ilgili bölge, sekiz filtrelenmiş görüntü kümesi üretilerek bu sekiz filtrenin her biri ile evriştirilir. 0^o yönelimli filtre ile evriştirilen parmakizi görüntüsündeki hatlar, açısal yönün x eksenine paralel olduğu ve diğer sekiz yöndeki hatlar düzleştiği için baskın hale gelir.

Filtreleme işlemi aynı şekilde diğer yönler için de uygulanır. Bu sekiz yönelimli filtreler ile parmakizindeki sınırlı hat karakteristikleri gibi ayrıntılı hat bilgisinin bir çoğu da elde edilir. Bu sekiz filtrelenmiş görüntünün tümünün birleştirilmesiyle yeniden düzenlenmiş parmakizi görüntüsü elde edilir. Yeniden düzenlenmiş görüntü, önemli bir bilgi kaybı olmaksızın orijinal görüntüye benzerdir (Şekil 4.10(l)).



a) 0^o yönelim b) 90^o yönelim

Şekil 4.11 Gabor filtre (maske boyutu=33×33; f=0,1; δ_x=4,0; δ_y=4,0)

Çalışmamızda sınırlı hat karakteristiklerini elde etmek için sekiz yönsel filtre kullanılmasına karşın ayrıntılı hat bilgisini elde etmek için dört yönsel filtre yeterli olmuştur (Şekil 4.10(k)). Hem sınırlı hem de ayrıntılı hat bilgisi elde edilirken sekiz filtrelenmiş görüntü arasında bazı bilgiler tekrarlı olmasına rağmen gerçekleme doğruluğu artırılmıştır. Eğer Gaussian dağılımının standart sapma değerleri olan δ_x ve δ_y çok büyük seçilirse, filtre gürültüye karşı daha kuvvetli olur, fakat parmakizindeki hat ve vadi yapılarını düzleştirerek istenen görüntüden uzaklaşmak olasıdır. Eğer δ_x ve δ_y çok küçük seçilirse, filtre gürültüyü ayıklamada etkili olamaz. Bu yüzden δ_x ve δ_y değerleri yaklaşık hatlar arası ortalama mesafenin yarısı olan 4,0 değeri olarak seçilmiştir.

4.4. Parmakizi Özellik Vektörünün Elde Edilmesi

Parmakizi görüntülerinde özellikle de iyi kalitede olmayan görüntülerde yapısal ayrıntıların çıkarılması zordur. Görüntülerin istatistiksel özellikleri ile ilgili ayrıntılar, görüntü kalitesinin kötüleşmesi ile azalabilir. Bu tez çalışmasında gri seviyeli parmakizi görüntüleri üzerine çalışılmıştır. Bir gri seviyeli görüntü

sektöründeki ortalama değerden elde edilecek ortalama mutlak sapma, parmakizi sınıflandırma ve tanıma için gerekli olan sektördeki tüm hat hareketinin göstergesidir. Benzer ayrıntılar, doku sınıflandırması ve segmentasyonu için Jain ve Farrokhnia (1991) tarafından başarılı bir şekilde kullanılmıştır.

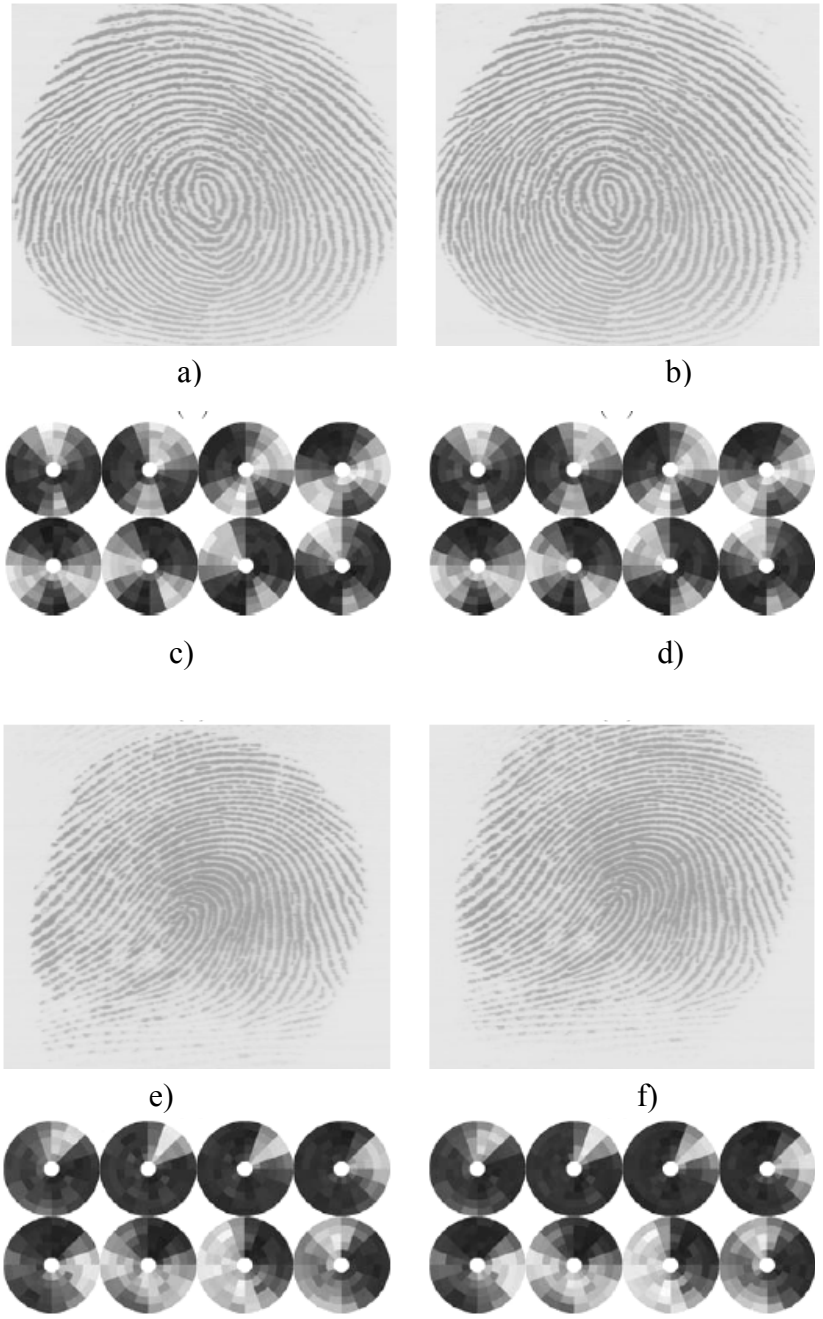
$FI_{i\theta}(x,y)$, S_i sektörü için θ-yönlü filtrelenmiş görüntü olsun. $\forall\, i \in \{0, 1,..., 79\}$ ve $\theta \in \{0^o, 22.5^o, 45^o, 67.5^o, 90^o, 112.5^o, 135^o, 157.5^o\}$ olmak üzere parmakizi özellik vektörü $V_{i\theta}$, aşağıdaki şekilde hesaplanır:

$$V_{i\theta} = \frac{1}{n_i}\left(\sum_{n_i} \left| FI_{i\theta}(x,y) - P_{i\theta} \right|\right) \tag{4.20}$$

Burada n_i, S_i sektöründeki piksellerin sayısıdır. $P_{i\theta}$ ise S_i sektöründeki $FI_{i\theta}(x,y)$ piksel değerlerinin ortalamasıdır. Sekiz filtrelenmiş her bir görüntüdeki sektörlerin her birisi için ortalama mutlak sapma, 640 boyutlu özellik vektörünü gösterir. NIST-4 veritabanındaki bazı örnek parmakizi görüntüleri için elde edilen özellik vektörleri Şekil 4.12'de gösterilmiştir.

Parmakizi tanıma uygulamalarında sekiz yönelim filtresi kullanmak dört yönelim filtresi kullanmaktan daha iyi sonuç vermiştir. Ayıca filtre sayısını artırmanın doğrulama performansında herhangi bir artışa sebep olmadığı gözlemlenmiştir.

NIST-4 veritabanından alınmış iki farklı parmağın parmakizi görüntülerinden elde edilmiş 640-boyutlu özellik vektörü (FingerCode), filtrelenmiş bir görüntüye uygun her bir disk ile toplam sekiz diskli gri seviyeli görüntüsü Şekil 4.12'de gösterilmiştir. Bir diskteki bir sektörün gri seviye değeri, aynı filtrelenmiş görüntüdeki sektörün özellik değerini gösterir. Şekil 4.12 (c) ve (d) görüntüleri, (g) ve (h) görüntülerine benzemesine rağmen bu iki parmak için aynı diskler çok farklı görülmektedir.



Şekil 4.12 İki farklı parmakizine ait 640-boyutlu özellik vektörleri. a) ve b) Birinci parmağa ait farklı zamanlarda alınmış iki parmakizi görüntüsü, c) ve d) Birinci parmakizi görüntülerine ait FingerCode diskleri, e) ve f) İkinci parmağa ait farklı zamanlarda alınmış iki parmakizi görüntüsü, (g) ve (h) İkinci parmakizi görüntülerine ait FingerCode diskleri

4.5. Sonuç

Parmakizi tanımadaki önemli aşamalardan biri, parmakizine ait referans noktasının şablonda elde edilen nokta ile aynı şekilde bulunmasıdır. Bu çalışmada referans noktasının bulunması için yeni bir yöntem geliştirilmiştir. Geliştirilen bu yöntem ile elde edilen referans noktasındaki açı bilgisi, parmakizine ait özellik vektörü elde edilirken referans olarak ele alınmıştır. Böylece elde edilen özellik vektörünün yapısı şablon yapısı ile örtüştürülmüştür. Bunun sonucunda eşleştirme aşamasında büyük bir hız ve zaman kazancı ile birlikte yüksek doğruluğun da elde edildiği gözlemlenmiştir. Ayrıca referans noktasının elde edilmesinde parmakizinin çizgi desenleri büyük rol oynamaktadır. Genellikle parmakizlerinin göbek noktaları, referans nokta olarak işaretlenmektedir.

Parmakizi eşleştirme teknikleri, ayrıntı tabanlı veya korelasyon tabanlı olarak sınıflandırılabilir. Ayrıntı tabanlı teknikler, eşleşen ayrıntıların toplam sayısını hesaplamayı ve iki ayrıntı noktaları kümesini düzenlemeyi amaçlar. Korelasyon tabanlı teknikler ise eğer iki parmakizindeki kabartılar hizalı ise kabartıların ayrıntılı desenlerini ve görünen izleri karşılaştırmayı amaçlar. Parmakizine ait iyileştirilmiş görüntüde, eleman yönleri arasındaki uyuşmazlıkları minimize etmek için uygun alan, bölgelere ayrılır. Her bölge ayrı ayrı işlenerek bu bölgelerin yönleri bulunur. Böylece ayrıntı tabanlı tekniklerde karşılaşılan bazı güçlükler korelasyon tabanlı tekniklerde aşılmıştır. Korelasyon tabanlı teknikler, referans noktası etrafında kesin bir bölgeyi kapsar ve parmakizi görüntüsünün döndürülmesinden veya ötelenmesinden etkilenir. Parmakizinin uygun bir özeti, yönlü parmakizi görüntüsü tarafından elde edilir. Bu çalışmada parmakizlerini tanımak amacıyla korelasyon tabanlı teknik geliştirilerek kullanılmış ve parmakizinin döndürülmesinden ve ötelenmesinden etkilenmeyecek bir sistem tasarlanmıştır. Sonuç olarak parmakizinin döndürülmesinden ve ötelenmesinden bağımsız bir yapı oluşturularak parmakizlerinin tanınması gerçekleştirilmiştir.

Parmakizinin yönlü doku gösterimi için Daugman (1999)'ın iris tanımasından ve Jain ve Farrokhnia (1991)'nın Gabor filtre kümesinden

esinlenilmiştir. Çalışmamızın temelini oluşturan ve parmakizi özellik vektörlerinin elde edilmesi için geliştirilen bu yöntemin aşamaları aşağıdaki şekilde sıralanır:

- Öncelikle değişmez bir referans noktası elde edilir.
- Bu referans nokta etrafında belirlenmiş ilgili alan bölgelere ayrılır.
- Sektörlere bölündükten sonra her bir sektör için Gabor filtre kullanılarak farklı bir veya daha fazla açısal dönüşüme göre özellikler elde edilir.
- Her bir hücre için elde edilen özellikler referans nokta alanının açısal yönünden başlanarak sıralanır.
- Bu sıralı dizi, parmakizinin simgelenişi olarak kullanılır.

5. YAPAY SİNİR AĞLARI İLE PARMAKİZİ TANIMA

Çağdaş dünyanın vazgeçilmez bir parçası haline gelen bilgisayarlar ve bilgisayar sistemleri, günümüzde hem olaylar hakkında karar verebilmekte hem de olaylar arasındaki ilişkileri öğrenebilmektedir. Matematiksel olarak ifade edilemeyen ve çözülmesi mümkün olmayan problemler tecrübeye dayalı yöntemler kullanılarak bilgisayarlar tarafından çözülebilmektedir (Doğan ve Alp 2002). Bilgisayarları bu özelliklerle donatan ve bu yeteneklerinin gelişmesini sağlayan çalışmalar "Yapay Zekâ" çalışmaları olarak bilinmektedir (Nabiyev 2004). Yapay zekâ, bilgi edinme, algılama, görme, düşünme ve karar verme gibi insan zekâsına özgü kapasitelerle donatılmış bilgisayarlardır (Whitby 2005). Yapay zekâ çalışmaları değişik teknolojilerin doğmasına neden olmuştur:

• Uzman sistemler: Bir uzmanın problemleri çözmesi gibi problemlere çözümler üreten bilgisayar programlarıdır (Allahverdi 2002).

• Yapay sinir ağları: Örneklerden olaylar arasındaki ilişkileri öğrenerek, daha sonra hiç görmediği örnekler hakkında öğrendikleri bilgileri kullanan ve insan beyin hücrelerinin çalışma prensibini taklit ederek karar veren sistemlerdir (Erler ve ark. 2003).

• Genetik algoritmalar: Geleneksel optimizasyon teknolojisi ile çözülemeyen problemleri çözmek üzere, biyolojideki genetik kurallarını uygulayarak daha iyi çözümler üretmek felsefesine dayanmaktadır (Şen 2004a).

• Bulanık mantık: Belirsiz bilgileri işleyebilme ve kesin rakamlar ile ifade edilemeyen durumlarda karar vermeyi kolaylaştıran bir teknolojidir (Şen 2001).

• Esnek hesaplama: Değişik yapay zekâ tekniklerini kullanabilen ve bağımsız olarak çalışabilen sistemler olup esnek bir şekilde programlanırlar (Görz ve Nebel 2006). Esnek hesaplama, insan aklının verdiği sezgi ve düşüncelerin gerçekleştirilmesini kullanır. Geleneksel hesaplama yöntemleri ile çözülemeyen veya

oldukça karmaşık olup tam modelleri tanımlanamayan gerçek dünya problemlerinin birçoğunda insan sezgilerinin ve tecrübelerinin kullanılması çok faydalı olmaktadır. Böylece esnek hesaplama, insan karar verme prosedüründe başarılı, basit, gerçekleştirilebilir ve düşük maliyetli çözümlerle belirsizlikleri ortadan kaldırır.

Esnek hesaplamanın temel bileşenlerinden olan yapay sinir ağları, bulanık mantık, kaos teorisi, genetik algoritmalar ve olasılıksal çıkarım kombine bir şekilde kullanılarak daha etkili olurlar (Aliev ve Aliev 2001). Esnek hesaplama bileşenlerinin temel kombinasyonları aşağıdaki şekilde sıralanabilir:

(1) Bulanık mantık + Genetik algoritmalar;

(2) Bulanık mantık + Kaos teorisi;

(3) Yapay sinir ağları + Genetik algoritmalar;

(4) Yapay sinir ağları + Kaos teorisi;

(5) Bulanık mantık + Yapay sinir ağları + Genetik algoritmalar;

(6) Yapay sinir ağları + Bulanık mantık + Genetik algoritmalar;

(7) Bulanık mantık + Olasılıksal çıkarım.

Esnek hesaplama yöntemlerinden olan yapay sinir ağları ve genetik algoritmaya dayalı modelleme yöntemleri bu yüzyılın son çeyreğinde bilim dünyasına sunulmuş ve kısa zamanda birçok farklı disiplin tarafından benimsenmiş ve yaygın olarak kullanılmış yöntemlerdir. Desen tanıma çalışmalarında kullanımı ise henüz çok yeni olmakla birlikte giderek artacağı düşünülmektedir. Parmakizi tanıma problemleri, YSA yöntemleri ve GA ile çözümlenebilmektedir. YSA ile parmakizi tanımada ortaya çıkan problemlerin analizi ve çözüm önerileri bu bölümde yapılmaktadır.

5.1. Yapay Sinir Ağları

Beynin üstün özellikleri, bilim adamlarını üzerinde çalışmaya zorlamış ve beynin nörofiziksel yapısından esinlenerek matematiksel modeli çıkarılmaya çalışılmıştır. Beynin bütün davranışlarını modelleyebilmek için fiziksel bileşenlerinin doğru olarak modellenmesi gerektiği düşüncesi ile çeşitli yapay hücre ve ağ modelleri geliştirilmiştir. Böylece, günümüz bilgisayarların algoritmik hesaplama yöntemlerinden farklı bir bilim dalı olarak "yapay sinir ağları" ortaya çıkmıştır (Karlık 1994).

Yapay sinir ağları (YSA), beyindeki sinirlerin çalışmasını taklit ederek sistemlere öğrenme, genelleme yapma, hatırlama gibi yetenekler kazandırmayı amaçlayan bilgi işleme metodudur (Öztemel 2003). Bir YSA, yapay sinir hücrelerinin birbirleri ile çeşitli şekillerde bağlanmasından oluşur. YSA'nın en önemli özelliklerinden birisi de onun öğrenme becerisinin olmasıdır. YSA'lar öğrenme algoritmaları ile öğrenme sürecinden geçtikten sonra, bilgiyi toplama, hücreler arasındaki bağlantı ağırlıkları ile bu bilgiyi saklama ve genelleme yeteneğine sahip olurlar. YSA'lar yapılarına göre farklı öğrenme yaklaşımları kullanırlar.

5.2. Parmakizi Özellik Vektörlerinin YSA'da Eğitimi ve Testi

Geleneksel bilgisayar uygulamalarının geliştirilmesinde bilgisayarın belli bilgisayar dilleri aracılığıyla ve kesin yazılım algoritmalarına uygun ifadelerle programlanması gerekir. Bu oldukça zaman alan, uyumluluk konusunda zayıf, teknik personel gerektiren ve pahalı bir süreçtir. Oysa biyolojik temele dayalı yapay zekâ teknolojilerinden biri olan yapay sinir ağlarının geliştirilmesinde programlama, yerini büyük ölçüde "eğitim" denilen sürece bırakmaktadır. İşlem elemanlarının bağlantılı

ağırlık değerlerinin belirlenmesi işlemine "ağın eğitilmesi" denir (Öztemel 2003). YSA'nın eğitilmesinde kullanılan parmakizi özellik vektörlerinden giriş ve çıkış dizileri çiftinden oluşan verilerin tümüne "eğitim seti" adı verilir.

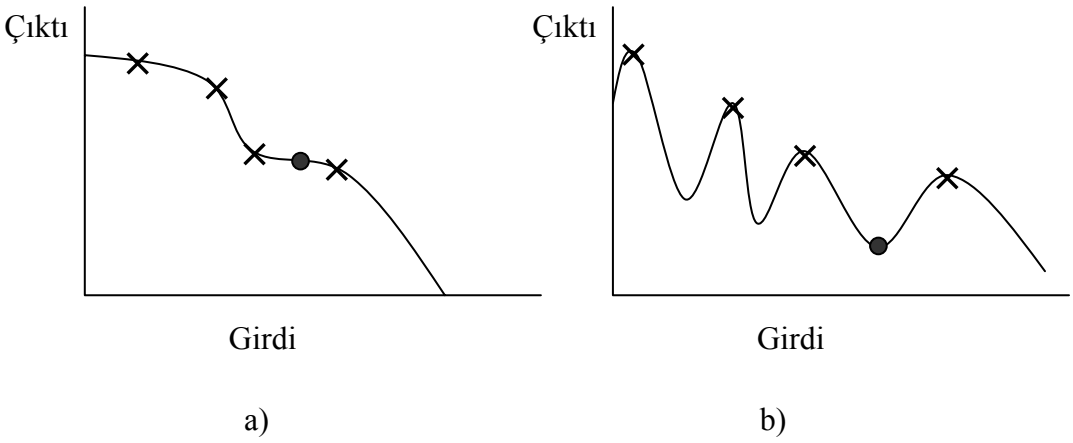
YSA öğrenme sürecinde, gerçek hayattaki problem alanına ilişkin veri ve sonuçlardan, bir başka deyişle örneklerden yararlanır. Gerçek hayattaki problem alanına ilişkin değişkenler YSA'nın giriş dizisini, bu değişkenlerle elde edilmiş gerçek hayata ilişkin sonuçlar ise YSA'nın ulaşması gereken hedef çıktıların dizisini oluşturur. Bu çalışmada parmakizi görüntülerine ait özellik vektörleri YSA'nın giriş dizisini ve parmakizlerine sahip kişiler ise YSA'nın hedef çıktılar dizisini oluşturmaktadır.

Öğrenme süresinde, seçilen öğrenme yaklaşımına göre ağırlıklar güncelleştirilir. Ağırlık değişimi, öğrenmeyi ifade eder. YSA'da ağırlık değişimi yoksa öğrenme işlemi de durur (Elmas 2003). Başlangıçta bu ağırlık değerleri rasgele atanır. YSA'lar kendilerine parmakizi örnekleri gösterildikçe, bu ağırlık değerlerini değiştirirler. Amaç, ağa gösterilen örnekler için doğru çıktıları üretecek ağırlık değerlerini bulmaktır. Ağın doğru ağırlık değerlerine ulaşması, örneklerin temsil ettiği olay hakkında genellemeler yapabilme yeteneğine kavuşması demektir. Bu genelleştirme özelliğine kavuşması işlemine "ağın öğrenmesi" denir (Efe ve Kaynak 2005).

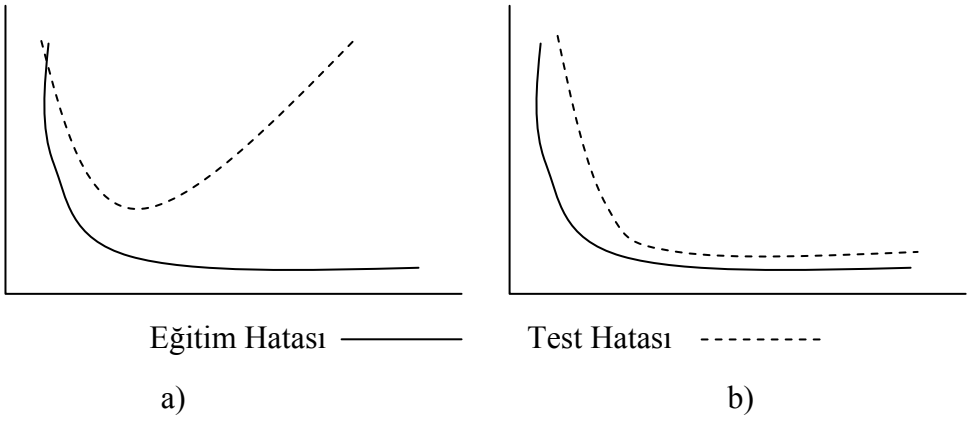
Eğitim süreci sonucunda YSA'da hesaplanan hatanın kabul edilebilir bir hata oranına inmesi beklenir. Ancak hata kareleri ortalamasının düşmesi her zaman için YSA'nın genellemeye ulaştığını göstermez. Yapay sinir ağının gerçek amacı giriş-çıkış örnekleri için genellemeye ulaşmaktır (Başaran 2003).

Genelleme, YSA'nın eğitimde kullanılmamış ancak benzer girdi-çıktı örneklerini ağın doğru bir şekilde sınıflandırabilme yeteneğidir (Elmas 2003). Şekilde (×) ile görülen noktalar eğitim verileri olmak üzere genellemenin nasıl gerçekleştiği Şekil 5.1(a)'da görülmektedir. Bunların arasında kalan eğri ise ağ tarafından oluşturulmaktadır. Bu eğri üzerindeki farklı bir girdi değeri için üretilen doğru çıktı değeri, ağın iyi bir genelleme yaptığını gösterir. Ancak ağ gereğinden fazla girdi-çıktı ilişkisini öğrendiğinde "ağ verileri ezberlemektedir" denir (Anderson

ve McNeill 1992). Bu durum genellikle gereğinden fazla gizli katman kullanıldığında verilerin sinaptik bağlantılar üzerinde saklanmasından veya gereğinden fazla veri kullanılarak eğitilmesinden kaynaklanmaktadır. Ezberleme, genellemenin iyi gerçekleşmediğini ve girdi-çıktı eğrisinin düzgün olmadığını gösterir (Şekil 5.1(b)). Verilerin ezberlenmiş olması YSA için istenmeyen bir durumdur. Bundan dolayı verileri ezberleyen ağ gerçek hayattaki örüntüyü iyi temsil edemeyeceği için kullanılamaz. Şekil 5.2(a)'da ağ verileri ezberlediği için eğitim hatası azalma, test hatası ise artma eğilimi göstermektedir. Şekil 5.2(b)'de ise ağ kabul edilebilir bir genellemeye ulaşmıştır (Ergezer ve ark. 2003).



Şekil 5.1 a) Genelleme eğilimi gösteren YSA, b) Ezberleme eğilimli YSA



Şekil 5.2 a) Verileri ezberleyen ve b) genellemeye ulaşan ağlardaki hata eğrileri

En uygun öğrenme seviyesi, öğrenme fonksiyonunun önceden amaçlanan bir değere ulaşması ile sağlanamayabilir. Uygulamalarda eğitim süreci boyunca performans fonksiyonu izlenmelidir (Rao ve Srinivas 2003). Ayrıca sık sık genelleme testleri gerçekleştirilerek en uygun öğrenme seviyesi elde edilebilir. Eğer en uygun öğrenme seviyesine, performans fonksiyonunun öngörülerinden önce ulaşılmış ise eğitim süresi daha erken sona erdirilebilir.

YSA sistemlerinin öğrenme başarısı, test edilerek sınanmalıdır. YSA geliştirme sürecinde veriler üçe ayrılır; bir bölümü ağın eğitilmesi için kullanılır ve "eğitim kümesi" adını alır, diğer bölümü ise ağın eğitim verileri dışındaki performansını ölçmede kullanılır ve "test kümesi" olarak adlandırılır (Ripley 1996). Verilerin bir kısmı da mesela YSA'nın gizli katmanındaki nöron sayısını belirlemek gibi YSA'nın ağırlık katsayılarından başka mimari parametrelerini düzenlemek için eğitme aşamasında kullanılır. Bu tür verilere "doğrulama veri kümesi" adı verilir (Reed 1993). Doğrulama seti, uygun YSA mimarisini belirlemede kullanılır.

Parmakizi özellik vektörlerinden oluşan eğitim ve test kümeleriyle ilgili temel sorun, yeterli eğitim ve test verisinin miktarının ne olacağıdır. Sınırsız sayıda verinin bulunabildiği durumlarda, YSA mümkün olan en çok veriyle eğitilmelidir. Eğitim verisinin yeterli olup olmadığı konusunda emin olmak için eğitim verisinin miktarı arttırılarak ağın performansında bir değişiklik oluşturup oluşturmadığına bakılır (Sağıroğlu ve ark. 2003). Ancak bunun mümkün olmadığı durumlarda YSA'nın eğitim ve test verileri üzerindeki performansının yakın olması da verilerin sayıca yeterli olduğuna ilişkin bir gösterge olarak kabul edilebilir.

Test kümesi, eğitim seti haricindeki verilerden seçilir. Bu kümedeki girdiler eğitilen YSA modeline verilir ve YSA'nın hesaplanan çıkış değeri ile istenilen çıkış değeri karşılaştırılır. Amaç, YSA modelinin yeterli bir genelleme yapıp yapamadığını görmektir. Eğitim ve test aşamalarında istenilen genelleme başarısı elde edilirse YSA modeli kullanılabilir. Eğitim ve test ile çapraz doğrulama (cross validation) kümesinin %25 ile %90 arasında değişen miktarı eğitim kümesi olarak seçilir (Setiono 2001). Geri kalan kısım ise test kümesi ve doğrulama kümesi olarak ayrılır.

Parmakizi özellik vektörleri, parmakizi görüntüsünden gri seviye değerleri olarak, yani 0 ile 255 arasında değerlerden oluşmaktadır. 0 değeri elde edilen bilginin siyah olduğunu, 255 değeri ise beyaz bir bilginin elde edildiğini göstermektedir. Bu çalışmada tüm özellik vektörleri kümesinin %70'i eğitim, %15'i doğrulama ve geri kalan %15 ise test veri kümesi olarak belirlenmiştir. Eğitim veri kümesi daha az seçildiği durumlarda doğru sınıflandırma oranının da düştüğü gözlemlenmiştir. Eğitim (TRN), doğrulama (VLD) ve test (TST) veri kümelerini oluşturan parmakizi özellik vektörleri Şekil 5.3'te gösterilmiştir. Burada ilk 384 sütun YSA'nın giriş verilerini, son 10 sütun ise YSA'nın çıkış verilerini oluşturmaktadır. Veri kümesi içerisinden eğitim, doğrulama ve test veri kümeleri rasgele seçilmiştir.

	(N) Sütun #374	(N) Sütun #375	(N) Sütun #376	(N) Sütun #377	(N) Sütun #378	(N) Sütun #379	(N) Sütun #380	(N) Sütun #381	(N) Sütun #382	(N) Sütun #383	(N) Sütun #384	(C10) Sütun #385
TRN	187	167	181	71	67	61	31	135	162	153	238	51
TRN	234	255	198	112	142	89	89	178	192	151	255	51
TRN	227	255	168	110	151	77	75	149	182	254	252	51
TRN	220	255	167	115	145	94	141	119	142	124	117	51
TRN	125	182	171	88	49	63	62	142	172	166	209	51
TRN	215	255	180	112	151	66	71	146	195	150	195	51
VLD	151	230	187	92	61	51	44	152	165	217	255	51
TRN	177	152	172	82	58	74	50	155	146	156	208	51
TRN	190	155	189	71	38	33	46	133	224	179	253	51
TST	174	153	171	83	56	72	40	155	150	153	207	51
TST	192	164	175	72	57	61	30	133	161	161	232	51
VLD	230	255	172	106	148	77	86	145	179	229	255	51
TRN	65	255	195	60	36	31	35	154	255	207	52	52
TRN	62	255	179	73	60	66	50	150	255	176	53	52
TRN	88	255	170	64	65	61	74	156	244	180	66	52
TRN	86	255	179	60	45	41	45	157	255	225	54	52
TRN	97	255	167	68	49	43	44	151	255	227	57	52
TRN	81	238	195	83	47	51	54	166	255	234	55	52
TRN	87	255	176	59	50	45	46	153	255	226	56	52
TST	84	255	170	68	67	55	62	152	250	177	65	52
VLD	82	235	177	73	38	45	33	165	255	210	65	52
TRN	83	231	195	89	49	51	53	155	255	239	59	52
TRN	81	255	186	67	64	67	65	154	243	178	68	52
TRN	92	255	173	58	54	45	45	149	255	226	54	52
TRN	79	255	185	71	61	67	61	159	242	171	66	52
TRN	64	255	185	68	50	58	47	159	255	205	51	52
TST	82	227	176	65	39	38	36	158	255	202	68	52
TRN	90	255	190	70	50	45	48	155	255	227	52	52
TRN	65	255	179	75	59	66	54	149	255	177	50	52
VLD	83	255	180	70	48	43	44	152	255	219	52	52
TRN	63	255	176	62	36	32	38	153	255	178	54	52
TST	88	255	193	68	39	34	50	155	255	226	53	52
VLD	80	237	176	79	37	48	36	162	255	194	62	52
TRN	84	255	178	59	44	43	49	153	255	219	50	52
TRN	84	238	192	90	53	54	55	173	255	236	55	52
TRN	90	255	180	60	43	42	41	154	255	224	54	52
TRN	89	255	179	57	45	41	44	150	255	229	53	52
TRN	66	255	179	56	46	39	45	145	255	242	59	52
TRN	88	255	169	66	58	58	65	157	241	187	68	52
TST	81	237	177	77	35	49	34	163	255	201	63	52
TRN	87	255	168	65	60	58	69	159	236	183	65	52
VLD	66	255	186	60	53	49	50	154	255	183	51	52
TRN	86	229	187	77	45	50	53	169	255	239	62	52
TRN	93	255	162	72	45	45	48	159	255	219	55	52
TRN	82	255	180	72	66	68	73	164	218	177	67	52

Şekil 5.3 Parmakizi özellik vektörlerinden oluşan YSA eğitim, doğrulama ve test veri kümeleri

5.3. Parmakizi Tanımada YSA Yapısı

Sinir hücreleri bir grup halinde işlev gördükleri için ağ olarak adlandırılırlar ve böyle bir grupta binlerce nöron bulunur. Yapay nöronların birbirleriyle bağlantılar yolu ile bir araya gelmeleri YSA'yı oluşturmaktadır (Haykin 1994). YSA ile aslında biyolojik sinir ağının bir modeli oluşturulmak istenmektedir. Nöronların aynı doğrultu üzerinde bir araya gelmeleri YSA'nın katmanlarını oluşturur. Katmanların değişik şekilde birbirleriyle bağlanmaları değişik ağ mimarilerini oluşturur. YSA'lar genel olarak üç katmandan oluşur (Şekil 5.4). Bu katmanlar sırasıyla;

- Giriş katmanı: Bu katmandaki işlem elemanları parmakizlerine ait giriş bilgilerini alarak ara katmanlara transfer ederler. Bazı ağlarda giriş katmanında herhangi bir bilgi işleme olmaz.

- Ara katman: Giriş katmanından gelen bilgiler burada işlenerek çıktı katmanına gönderilir. Bir ağ içinde birden fazla ara katman olabilir.

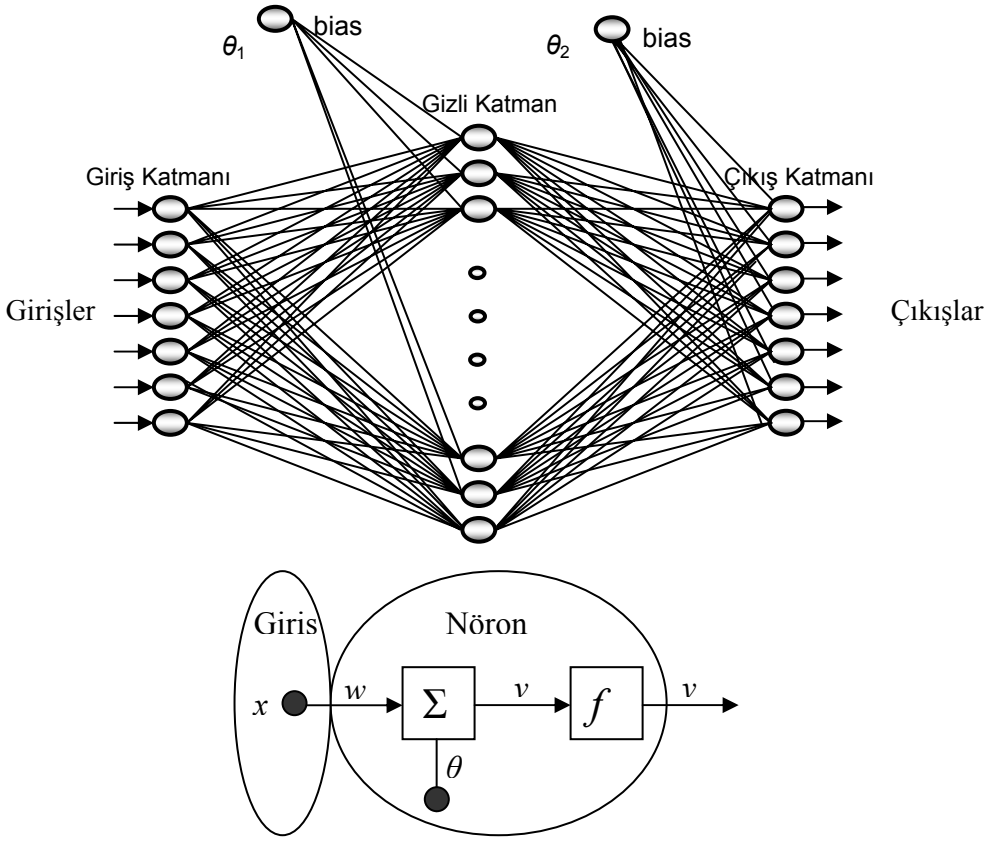
- Çıkış katmanı: Bu katmandaki işlem elemanları ara katmandan gelen bilgileri işleyerek ağın giriş katmanından sunulan ve giriş seti için üretmesi gereken çıkış verilerini üretirler.

- Bias girişi: Bias, kendisinden sonraki katmanın tüm nöronları ile bağlı olan bir nörondur. Genelde 1 değerini üretir. Bazı durumlarda YSA yapısı oluşturulurken rasgele değer üretir. Eğitim aşamasında ağdaki tüm ağırlıklarla birlikte bias ağırlığı da güncellenir.

5.4. Yapay Sinir Hücresi

YSA'da da biyolojik sinir ağlarında olduğu gibi temel unsur, yapay sinir hücresidir. Yapay sinir hücresi, YSA'nın çalışmasına esas teşkil eden en küçük ve temel bilgi işleme birimidir Ağ içerisinde yer alan tüm nöronlar bir veya birden fazla giriş bilgisi alırlar ve tek bir çıkış verirler (Ripley 1996). Bu çıkış, YSA'nın dışına verilen çıkışlar olabileceği gibi başka nöronlara giriş olarak da kullanılabilir.

Şekil 5.4'ten de görüldüğü üzere geliştirilen hücre modellerinde bazı farklılıklar olmakla birlikte genel özellikleri ile bir yapay hücre modeli 5 bileşenden oluşur (Çikoğlu 2003).



$$y = f(xw+\theta)$$

Şekil 5.4 Yapay sinir ağı ve yapay sinir hücresi (nöron) modeli

Bu bileşenler;

- Girdiler: Girdiler, diğer hücrelerden ya da dış ortamlardan hücreye giren bilgilerdir.

- Ağırlıklar: Bilgiler, bağlantılar üzerindeki ağırlıklar üzerinden hücreye girer ve ağırlıklar, ilgili girişin hücre üzerindeki etkisini belirler. Ağırlıklar bir nöronda girdi olarak kullanılacak değerlerin göreceli kuvvetini (matematiksel katsayısını) gösterir. YSA içerisinde girdilerin nöronlar arasında iletimini sağlayan tüm bağlantıların farklı ağırlık değerleri bulunmaktadır. Böylelikle ağırlıklar her işlem elemanının tüm girdileri üzerinde etki yaparlar.

- Birleştirme fonksiyonu: Birleştirme fonksiyonu, bir hücreye gelen net girdiyi hesaplayan bir fonksiyondur ve genellikle net girdi, girişlerin ilgili ağırlıkla çarpımlarının toplamı olarak hesaplanır. Birleştirme fonksiyonu, ağ yapısına göre maksimum, minimum ya da çarpım fonksiyonu olabilir.

$$v = \sum x_i w_i + \theta, \quad y = f(v) \tag{5.1}$$

Burada w, hücrenin ağırlıklar matrisini; x, hücrenin giriş vektörünü; v, hücrenin net girişini; y, hücre çıkışını; θ, bias girişini ve f, nöron çıkışını izin verilen aralıkta tutan aktivasyon fonksiyonunu simgeler.

- Aktivasyon fonksiyonu: Transfer fonksiyonu olarak da adlandırılan aktivasyon fonksiyonu, birleştirme fonksiyonundan elde edilen net girdiyi işlemden geçirerek hücre çıktısını belirleyen ve genellikle doğrusal olmayan bir fonksiyondur (Psaltis ve ark. 1988). Hücre modellerinde, hücrenin gerçekleştireceği işleve göre çeşitli tipte aktivasyon fonksiyonları kullanılabilir. Aktivasyon fonksiyonları sabit parametreli ya da uyarlanabilir parametreli seçilebilir. Aktivasyon fonksiyonunun seçimi büyük ölçüde YSA'nın verilerine ve ağın neyi öğrenmesinin istendiğine bağlıdır. Geçiş fonksiyonları içinde en çok kullanılanı sigmoid ve hiperbolik tanjant fonksiyonlarıdır. Örneğin eğer ağın bir modelin ortalama davranışını öğrenmesi isteniyorsa sigmoid fonksiyon, ortalamadan sapmanın öğrenilmesi isteniyorsa hiperbolik tanjant fonksiyon kullanılması önerilir (Anderson ve McNeill 1992). Aktivasyon fonksiyonları olarak eşik aktivasyon, sigmoid aktivasyon, doğrusal

aktivasyon, tanjant hiperbolik aktivasyon fonksiyonlarından başka basamak fonksiyonu, kutuplamalı basamak fonksiyonu ve parçalı doğrusal fonksiyon da mevcuttur.

- Çıktı: En basit şekilde, bir işlem elemanı, bir girdiyi bir ağırlık kümesi ile ağırlıklandırır, doğrusal olmayan bir şekilde dönüşümünü sağlar ve bir çıktı değeri oluşturur (Yurtoğlu 2005). Aktivasyon fonksiyonundan geçirildikten sonra elde edilen değer, çıktı değeridir.

5.5. Yapay Sinir Ağı Yapıları

YSA'daki sinir hücrelerinin sayısı, bunların birbirlerine göre konumu ve birbirleri arasındaki sinyallerin akış yönleri YSA yapısını belirlemektedir (Haykin 1998). Yapısı belirlenmiş bir YSA eğitilerek YSA ile ilgili uygulamalarda kullanılır. Bu tez çalışmasında parmakizi özellik vektörleri kullanılarak YSA'da parmakizi tanıma uygulanmıştır.

YSA yapıları, performans ve karakteristik özellikleri bakımından birbirlerinden farklıdırlar. YSA yapıları, özellikle ağın modelleme yeteneğini belirledikleri için oldukça önemlidirler (Sağıroğlu ve ark. 2003). YSA'nın tasarım aşamasında bu ağ yapıları arasından uygulamaya en elverişlisi seçilir.

5.5.1. Parmakizi tanımada YSA tasarımı

YSA uygulamasının başarısı, uygulanacak olan yaklaşımlar ve deneyimlerle yakından ilgilidir. YSA'nın parmakizi tanımadaki başarısında uygun metodolojiyi belirlemek büyük önem taşır. YSA'nın geliştirilmesi sürecinde ağın yapısına ve işleyişine ilişkin aşağıdaki karakteristiklerin belirlenmesi gereklidir (Algil 2003):

- Ağ mimarisinin seçilmesi ve yapı özelliklerinin belirlenmesi (katman sayısı, katmandaki nöron sayısı v.b.)
- Nörondaki fonksiyonların karakteristik özelliklerinin belirlenmesi,
- Öğrenme algoritmasının seçilmesi ve parametrelerinin belirlenmesi,
- Eğitim ve test verilerinin oluşturulması

Bu kararların doğru verilememesi durumunda, YSA'ların sistem karmaşıklığı artar. Sistem karmaşıklığı, yapısal ve toplam hesaplama karmaşıklığının bir fonksiyonudur. Toplam hesaplama karmaşıklığı ise, genellikle yapısal karmaşıklığın bir fonksiyonu olarak ortaya çıkar ve bu hesaplamanın en aza indirilmesi amaçlanır (Anderson ve McNeill 1992). Bu hesaplama karmaşıklığının ölçülmesinde de genellikle YSA sisteminin toplam tepki süresi veya sisteme ait bir işlemci elemanın tepki süresi değeri temel alınır. Bunun yanında kapladığı hafıza ve zaman karmaşıklığı da bazı uygulamalarda hesaplanmaktadır.

Bir YSA'nın uygun parametrelerle tasarlanması durumunda YSA sürekli olarak kararlı ve istikrarlı sonuçlar üretecektir (Öztemel 2003). Ayrıca sistemin tepki süresinin yeterince kısa olabilmesi için de ağ büyüklüğünün yeterince küçük olması gerekir. İhtiyaç duyulan toplam hesaplama da bu sayede sağlanmış olacaktır.

5.5.2. YSA yapısının ve öğrenme algoritmasının seçimi

YSA'nın tasarımı sürecinde ağ yapısının seçilmesi, uygulama problemine bağlı olarak seçilmelidir. Hangi problem için hangi ağın daha uygun olduğunun bilinmesi önemlidir. Kullanım amacı ve o alanda başarılı olan ağ türleri Tablo 5.1'de verilmiştir.

Uygun YSA yapısının seçimi, büyük ölçüde ağda kullanılması düşünülen öğrenme algoritmasına da bağlıdır. Ağda kullanılacak öğrenme algoritması

seçildiğinde, bu algoritmanın gerektirdiği mimari de zorunlu olarak seçilmiş olacaktır. Örneğin geriyayılım algoritması ileri beslemeli ağ mimarisi gerektirir. YSA'nın geliştirilmesinde kullanılacak çok sayıda öğrenme algoritması bulunmaktadır (Erler ve ark. 2003). Bunlar içinde bazı algoritmaların bazı tip uygulamalar için daha uygun olduğu bilinmektedir. Bu algoritmalar eğer uygun oldukları uygulama alanlarına göre sınıflandırılacak olursa, gruplar ve içinde yer alacak öğrenme algoritmalarından bazıları Tablo 5.1'deki gibi sıralanabilir.

Tablo 5.1 YSA öğrenme algoritmaları ve uygulama alanları

Uygulama tipi	**YSA algoritması**	**Kullanımı**
Öngörü Tanıma ve Tahmin	Geriyayılım (Backpropagation) Delta Bar Delta Geliştirilmiş Delta Bar Delta Geriyayılım içinde Öz Düzenleyici Özellik Haritası (Self Organizing Feature Map in Backpropagation) Yüksek Dereceli Sinir Ağları (Higher Order Neural Networks) İleri Beslemeli Regresyon (Feed-forward Regression)	Giriş değerlerini kullanarak istenen çıkış bilgileri tahmin edilir. Mesela bir hastaya ait laboratuar bulguları kullanılarak kanser riskli kişilerin tespiti veya önceki yıllarda meydana gelen hava olayları dikkate alınarak mevcut hava durumunun tahmin edilmesidir.
Sınıflandırma	Öğrenmeli Vektör Kuantalama (Learning Vector Quantization) Sayıcı-Yayılımlı (Counter-Propagation) Olasılıksal Yapay Sinir Ağları (Probabilistic Neural Network) Adaptif Rezonans Ağları (Adaptive Resonance Network)	Giriş değerlerini kullanarak belirli özelliklere sahip bilgilerin sınıflandırılması için kullanılır. Mesela alfabedeki harflerin sınıflandırılması veya parmakizi tiplerine göre parmakizlerinin sınıflandırılmasıdır.
Veri İlişkilendirme	Hopfield Ağları Boltzmann Makinesi Radyal Tabanlı Fonksiyon (Radial Basis Function) İki yönlü Çağrışımlı Bellek (Bidirectional Associative Memory)	Sınıflandırmaya benzer fakat burada tarayıcıdan düzgün olarak taranmayan bilgiler gibi hata içeren veriler de tanımlanır.

5.5.3. Ara katman sayısının ve nöron sayısının belirlenmesi

YSA'nın tasarımı sırasında yapılması gereken diğer işlem de ağdaki katman sayısına karar vermektir. Çoğu problem için iki veya üç katmanlı bir ağ beklenen sonuçları üretebilir (Şen 2004b). Nöronların aynı doğrultu üzerinde bir araya gelmeleriyle katmanlar oluşmaktadır. Katmanların değişik şekilde birbirleriyle bağlanmaları değişik ağ yapılarını oluşturur. Giriş ve çıkış katmanlarının sayısı, problemin yapısına göre değişir (Freeman ve Skapura 1991). Bu çalışmada değişik sayılarda ara katman kullanılmıştır, fakat en iyi sonuçlar bir ara katman kullanılmasıyla elde edilmiştir.

Ağın yapısal özelliklerinden birisi her bir katmandaki nöron sayısıdır. Katmandaki nöron sayısının tespitinde geleneksel yöntem olarak deneme-yanılma yöntemi kullanılır. Bunun için izlenecek yol, başlangıçtaki nöron sayısını istenilen performansa ulaşıncaya kadar arttırmak veya tersi şekilde istenen performansın altına inmeden azaltmaktır. Bir katmanda kullanılacak nöron sayısı olabildiğince az olmalıdır. Nöron sayısının az olması yapay sinir ağının "genelleme" yeteneğini arttırırken, gereğinden fazla olması ağın verileri ezberlemesine neden olur (Saraç 2004). Ancak gereğinden az nöron kullanılması, verilerdeki örüntünün ağ tarafından öğrenilememesi gibi bir sorun yaratabilir.

Bu çalışmada Neurointelligence (2003) programı kullanılarak deneme-yanılma yolunun aksine tecrübeye dayalı (heuristic) metot ile nöron sayısı belirlenmiştir. Burada gizli katman nöron sayısı belirli aralıkta aranır. Tecrübeye dayalı arama metodu, her bir adımda başlangıç arama aralığını sınırlar ve arama doğruluğunu son adımda denenen mimari ile bir önceki arasındaki durumda belirler (Breiman 1994). İterasyon sayısı ve yeniden eğitme sayısı dikkate alınarak eğitme işlemi uygulanan YSA için uygunluk kriteri hesaplanır. Uygunluk kriterine göre en iyi performansı veren YSA yapısı bu problem için seçilir. Burada uygunluk kriteri olarak Akaike'nin bilgi kriteri kullanılmıştır.

Akaike Bilgi Kriteri (AIC), Hirotsugu Akaike (1974) tarafından geliştirilmiş ve önerilmiş istatistiksel bir modeldir. Bu kriter, gizli katmanda farklı ağırlıklara sahip farklı ağlar için kullanılır. YSA yapısını belirleme esnasında uygunluk kriteri olarak kullanılan bu ölçüt, basit modellerde tercih edilir. Nöron sayısı ne olursa olsun bir gizli katmanlı YSA yapısı için en uygun nöron sayısını tanımlamada iyi sonuçlar elde edilir. Gizli katman sayısı arttıkça ağırlık sayısı da artacağından dolayı ağ yapısı karmaşık bir hale gelecektir. Bu tür durumlarda AIC ile nöron sayısı belirleme tercih edilmez.

Akaike bilgi kriteri, yapay sinir ağındaki nöronların en uygun sayısını belirler. Bu yöntemde farklı sayıda parametreye sahip olan farklı modeller arasından seçim yapılmaktadır. Bu yöntem, her bir modelin ayrı ayrı en çok olabilirlik fonksiyonunu bulup olabilirlik fonksiyonunun değeri en küçük olan modeli en uygun model olarak seçmektir (Schwarz 1978).

Akaike bilgi kriteri, aşağıdaki eşitlik ile hesaplanır:

$$\mathrm{AIC} = \ln\left(\frac{\sum e_i^2}{n}\right) + \frac{2k}{n} \tag{5.2}$$

Burada n, modeldeki parametrelerin sayısı ve e_i^2 ise hesaplanan gözlem ve hesap sonuçları arasındaki farkın varyansıdır. AIC, $\sum e_i^2$ ile tahmin edilecek parametre sayısı olan k'ya bağlıdır (Burnham ve Anderson 2002). Burada yeni açıklayıcı değişken ilave edildikten sonra $\sum e_i^2$ 'deki düşüş, AIC kriterinde bir düşüşe neden olmadıkça modele değişken ilave edilmesi gerekli değildir. Çünkü her ilave değişken tahmin edilecek parametre sayısında artışa neden olur. Eğer $\sum e_i^2$ 'deki düşme k'daki artışın etkisinden daha önemli ise burada AIC'nin değeri düşer.

Buna göre AIC'nin bazı özellikleri söyle sıralanır (Ucal 2006):

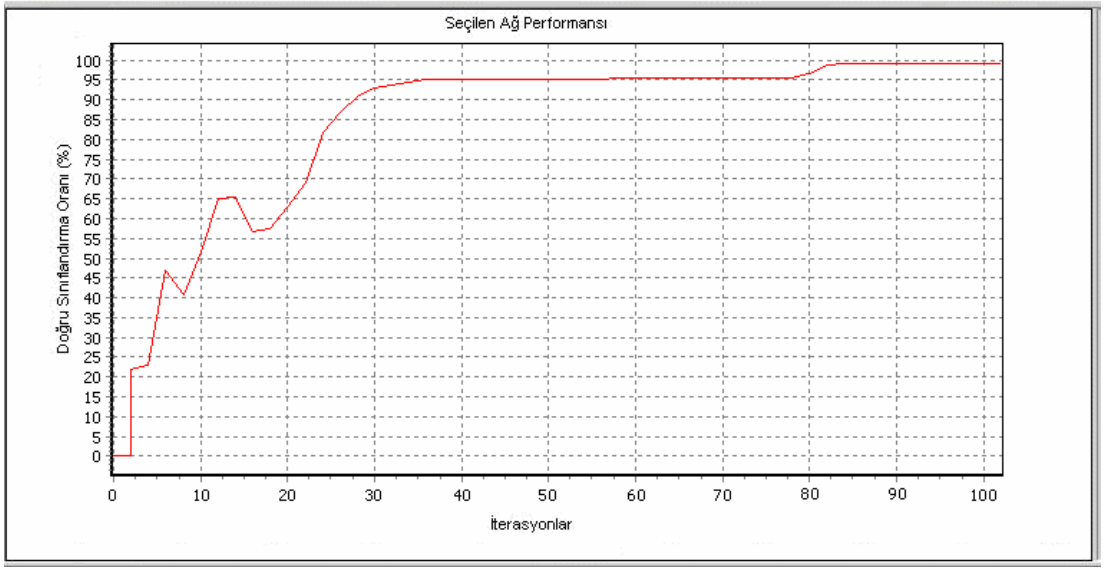
- Lineer modellerde rahatlıkla kullanılabilir. Model karşılaştırmalarında her zaman en düşük AIC değerini veren model tercih edilir.
- AIC sadece seçili örnek büyüklüğü içinde değil aynı zamanda seçili örnek büyüklüğü dışındaki gelecek tahmini içinde geçerlidir.

YSA yapısı da dikkate alındığında lineer bir model söz konusu olduğundan dolayı uygunluk olarak seçilebilen AIC'nin en düşük olduğu yapı en uygun yapı olarak tercih edilir. Bu tez çalışmasında arama, belirlenen aralıktaki nöron sayılarına göre oluşturulan YSA yapısının AIC'ye göre göre en düşük oranı veren YSA yapısının belirlenmesiyle sonuçlanır.

Şekil 5.5'te tecrübeye-dayalı gizli katman nöron sayısı belirleme işlemi gösterilmiştir. Burada 4 bant ve 12 sektörden elde edilen parmakizi özellik vektörleri için 384 giriş ve 10 çıkışa sahip YSA yapısı ele alınmıştır. Bir gizli katman için nöron sayısının belirleme aralığı 2 ile 80 olarak seçilmiştir. Gerçekleştirilen belirleme işlemi neticesinde AIC'nin en düşük oranına göre gizli katman nöron sayısı 58 olarak belirlenmiş ve YSA yapısı da 384:58:10 olarak elde edilmiştir.

Böylece tüm bant ve sektör sayılarına göre elde edilmiş özellik vektörü kümeleri için en uygun YSA yapısı belirlenmiştir. Tüm yapılarda bir gizli katman olmak üzere belirleme işlemi gerçekleştirilmiştir.

No	Mimari	Ağırlık Sayısı	Uygunluk	Eğitim Hatası	Doğrulama Hatası	Test Hatası	Akaike Bilgi Kriteri	Durma Nedeni
1	[384-2-10]	800	1.898734	0.498571	0.533333	0.473333	21468.962113	İterasyonlar Tamamlandı
2	[384-50-10]	19760	21.428571	0.997286	0.991667	0.973334	31318.893584	İterasyonlar Tamamlandı
3	**[384-58-10]**	**22255**	**30**	**0.995714**	**0.986667**	**0.966667**	**16107.516133**	**İterasyonlar Tamamlandı**



Şekil 5.5 Tecrübeye dayalı arama metodu ile belirlenen en uygun YSA yapıları

5.5.4. Parmakizlerine ait özellik vektörlerinin normalizasyonu

YSA'ların en belirgin özelliklerinden olan doğrusal olmama özelliğini anlamlı kılan yaklaşım, YSA'ya uygulanacak olan verilerin bir normalizasyona tabii tutulmasıdır. Verilerin normalizasyonu için seçilen yöntem YSA'nın eğitimi sırasındaki performansı doğrudan etkiler. Verilerin normalize edilmesi ile işlemci elemanlarında verilerin kümülatif toplamlarının oluşturacağı olumsuzluklar engellenir. Genel olarak veriler [0 1] veya [-1 +1] aralıklarına normalize edilir. Normalizasyon ile veriler geçerli eksen sisteminde sıkıştırıldığından dolayı bu durum, veri kalitesi aşırı salınımlar içeren problemlerin YSA modellerini olumsuz yönde etkileyebilir (Saraoğlu 1998). Bu olumsuzluk, kullanılacak öğrenme fonksiyonunu da başarısız kılabilir.

Örneğin, bir transfer fonksiyonuna uygulanan girişler ile ağırlıkların çarpım toplamının 50 ve 100 olduğunu farz edelim. Bu toplamı bir tanjant hiperbolik fonksiyonundan geçirdiğimizde;

$$y = tanh(50) = (e^{50} - e^{-50}) / (e^{50} + e^{-50}) = 1{,}0 \tag{5.3}$$

ve

$$y = tanh(100) = (e^{100} - e^{-100}) / (e^{100} + e^{-100}) = 1{,}0 \tag{5.4}$$

sonuçları elde edilir. Burada, 50 ve 100 değerlerine karşılık gelen fonksiyon sonuçları arasında fark yoktur. Bu durumda da, birbirinden oldukça farklı değerler sistemde sanki aynı değere sahipmiş gibi ele alınacak ve hem uygulama hem de öğrenme algoritması açısından olumsuz sonuçlar ortaya çıkacaktır.

Bu olumsuz sonuçlardan etkilenmemek için, X veri kümesi [-1 +1] ya da [0 1] aralığına normalize edilmelidir. Veri kümesinin [0 1] aralığına normalizasyonu için öncelikle o kümenin X_{min} ve X_{max} değerleri bulunur ve aşağıdaki formüle göre normalizasyon yapılır.

$$X_{yeni} = \frac{X - X_{\min}}{X_{\max} - X_{\min}} \tag{5.5}$$

Bu çalışmada parmakizi özellik vektörünü oluşturan 0 ile 255 arasındaki değerler YSA girişine verilmeden önce [-1 +1] aralığına normalize edilmiştir. Normalizasyon işlemi özellik vektörlerinin her bir sütunundaki en düşük ve en yüksek değerler dikkate alınarak yapılır. Özellik vektör değerleri aşağıdaki formül kullanılarak normalize edilmiştir:

$$SF = \frac{SR_{\max} - SR_{\min}}{X_{\max} - X_{\min}},\ X_p = SR_{\min} + (X - X_{\min}) \times SF \tag{5.6}$$

Bu eşitliklerde X, özellik vektör değeri; X_{min}, ilgili özellik vektör sütununun en küçük değeri; X_{max}, ilgili özellik vektör sütununun en büyük değeri; SR_{min}, en küçük ölçekleme değeri; SR_{max}, en büyük ölçekleme değeri; SF, ölçekleme faktörü, X_p ise ölçeklenme sonucu elde edilen yeni değerdir.

Mesela özellik vektöründe en küçük değer 0 ve en büyük değer 255 olsun. Özellik vektörü içerisinde 187 ve 255 olan değerlerin [-1 +1] aralığına normalizasyon sonucu elde edilen yeni değerleri aşağıdaki şekilde bulunur:

$$SF = \frac{1-(-1)}{255-0},\ X_p = (-1) + (187-0) * \frac{2}{255} = 0{,}4667$$

$$SF = \frac{1-(-1)}{255-0},\ X_p = (-1) + (255-0) * \frac{2}{255} = 1$$

Şekil 5.6'da parmakizi özellik vektörlerinin [-1 +1] aralığına normalizasyonu sonucu elde edilen yeni değerler gösterilmiştir. Böylece parmakizi özellik vektörlerinin 0 ile 255 arasında değerler aldığı göz önünde bulundurulursa, özellik vektörü 0 olan bir verinin Normalizasyon sonucu yeni değeri -1 ve 255 olan bir verinin de +1 olduğu görülür.

Parmakizi özellik vektörleri [0 1] aralığına normalize edilerek YSA ile eğitme aşamasına hazır hale getirilmiştir. Burada sistem performansı ile birlikte YSA eğitim süresinin de çok fazla değişmediği gözlemlenmiştir.

Tüm bu işlemler gerçekleştirilerek normalize edilmiş parmakizi özellik vektörleri, YSA eğitimine hazır hale getirilir.

Sütun #1	Sütun #2	Sütun #3	Sütun #4	Sütun #5	Sütun #6	Sütun #7	Sütun #8	Sütun #9	Sütun #10	Sütun #11	Sütun #12	Sütun #13	Sütun #14	Sütun #15	Sütun #16	Sütun #17
1	0.709091	0.412281	-0.06087	-0.116883	-0.296137	-0.175966	0.159292	0.784483	1	1	1	0.208	1	-0.345133	-0.422222	-0.45045
0.808696	0.663636	0.342105	-0.286957	-0.402597	-0.613734	-0.313305	0.017699	0.732759	1	1	1	0.248	0.783333	-0.345133	-0.484444	-0.072072
0.704348	0.2	0.719298	0.313043	0.878788	1	0.648069	1	1	-0.099174	0.891213	1	0.336	0.558333	0.327434	0.777778	1
1	0.736364	0.385965	0.017391	0.004329	-0.313305	-0.253219	0.035398	0.810345	1	1	1	0.304	0.858333	-0.300885	-0.342222	-0.468468
0.852174	0.845455	0.087719	-0.191304	0.056277	-0.381974	-0.339056	-0.097345	0.5	1	1	1	0.208	0.5	-0.300885	-0.44	-0.342342
0.913043	0.518182	0.578947	0.53913	0.965368	1	0.579399	1	0.698276	0.297521	0.924686	1	0.376	0.458333	0.274336	0.555556	0.873874
0.747826	0.4	0.631579	0.556522	0.922078	1	0.502146	1	0.672414	0.289256	0.90795	1	0.376	0.483333	0.318584	0.573333	0.864865
0.756522	0.245455	0.77193	0.330435	0.878788	1	0.587983	1	0.922414	-0.214876	0.941423	1	0.328	0.541667	0.39823	0.724444	1
1	0.790909	0.631579	-0.052174	-0.134199	-0.175966	-0.072961	0.185841	0.827586	1	1	1	0.216	0.941667	-0.371681	-0.431111	-0.432432
1	0.436364	-0.157895	0.104348	-0.160173	-0.381974	-0.656652	-0.115044	0.465517	1	1	0.931915	0.512	0.508333	-0.566372	-0.546667	-0.558559
1	0.445455	-0.017544	0.034783	-0.151515	-0.330472	-0.545064	-0.00885	0.525862	1	1	1	0.4	0.708333	-0.415929	-0.466667	-0.540541
0.643478	0.027273	0.745614	0.426087	0.82684	1	0.133047	1	0.905172	0.231405	0.941423	1	0.328	0.516667	0.327434	0.688889	1
0.956522	0.545455	-0.061404	-0.095652	-0.125541	-0.330472	-0.459227	0	0.491379	1	1	0.923404	0.304	0.541667	-0.530973	-0.564444	-0.495495
0.730435	0.563636	0.692982	-0.208696	-0.419913	-0.467811	-0.227468	0.19469	0.741379	1	1	1	0.384	0.775	-0.336283	-0.448889	0.045045
0.747826	0.418182	0.631579	0.547826	0.922078	1	0.613734	1	0.663793	0.347107	0.90795	1	0.384	0.458333	0.353982	0.564444	0.873874
0.669565	0.127273	0.657895	0.608696	0.705628	1	0.167382	1	0.681034	0.338843	0.90795	1	0.304	0.508333	0.318584	0.582222	1
0.678261	0.172727	0.596491	0.6	0.904762	1	0.253219	1	0.758621	0.289256	0.916318	1	0.376	0.5	0.327434	0.635556	0.864865
0.791304	0.272727	0.754386	0.556522	0.948052	1	0.648069	1	0.62931	0.793388	0.832636	0.87234	0.28	0.466667	0.362832	0.582222	0.837838
1	0.827273	0.587719	-0.008696	-0.125541	-0.296137	-0.158798	0.159292	0.801724	0.917355	1	1	0.144	0.958333	-0.336283	-0.395556	-0.423423
0.834783	0.2	0.666667	0.565217	0.887446	1	0.347639	1	0.793103	0.446281	0.974895	1	0.304	0.541667	0.247788	0.555556	1
0.878261	0.518182	0.526316	-0.565217	-0.367965	-0.433476	-0.193133	0.256637	0.913793	1	1	1	0.52	0.916667	-0.318584	-0.422222	-0.054054
0.521739	0.563636	0.745614	-0.286957	-0.385281	-0.347639	-0.064378	0.433628	0.939655	1	1	1	0.504	1	-0.168142	-0.475556	-0.108108
0.756522	0.736364	0.596491	-0.121739	-0.333333	-0.596567	-0.347639	0.061947	0.62069	1	1	1	0.2	0.716667	-0.318584	-0.395556	0
0.652174	0.145455	0.605263	0.66087	0.904762	1	0.107296	1	0.681034	0.380165	0.891213	1	0.32	0.541667	0.345133	0.626667	1
-0.33913	0.190909	0.078947	-0.06087	0.575758	0.639485	-0.502146	-0.460177	-0.12069	-0.22314	-0.58159	-0.625532	-0.392	0.466667	-0.876106	-0.288889	-0.513514
0.66087	0.209091	0.754386	0.313043	0.91342	1	0.579399	1	0.939655	-0.206612	0.924686	1	0.336	0.55	0.389381	0.733333	1
0.730435	0.454545	0.526316	0.46087	0.904762	1	0.519313	1	0.698276	0.553719	0.933054	1	0.384	0.466667	0.389381	0.6	0.81982
0.930435	0.527273	0.640351	0.417391	0.948052	1	0.699571	1	0.508621	0.46281	1	1	0.36	0.458333	0.247788	0.537778	0.954955
1	0.345455	0.122807	-0.113043	-0.212121	-0.321888	-0.459227	-0.026549	0.525862	1	1	0.940426	0.408	0.65	-0.610619	-0.6	-0.630631
0.92913	0.018182	0.649123	0.365217	0.904762	1	0.682403	1	0.344828	0.586777	0.782427	1	0.424	0.641667	0.389381	0.822222	0.810811
1	0.7	0.394737	-0.113043	-0.134199	-0.399142	-0.27897	0.053097	0.775862	1	1	1	0.224	1	-0.318584	-0.466667	-0.486486
0.643478	0.227273	0.614035	0.669565	0.852814	1	0.304721	1	0.75	0.347107	0.949791	1	0.4	0.541667	0.345133	0.635556	0.900901
1	0.727273	0.412281	-0.104348	-0.056277	-0.390558	-0.201717	0.141593	0.775862	1	1	1	0.216	1	-0.300531	-0.431111	-0.441441
0.956522	0.536364	0.54386	0.652174	1	1	0.527897	1	0.715517	0.421488	1	0.974468	0.36	0.466667	0.292035	0.52	0.927928
1	0.836364	0.561404	-0.052174	-0.194805	-0.390558	-0.175966	0.150442	0.75	0.884298	1	1	0.192	0.975	-0.362832	-0.422222	-0.396396
1	0.845455	0.342105	-0.017391	-0.082251	-0.373391	-0.253219	0.053097	0.698276	1	1	1	0.184	0.875	-0.265487	-0.377778	-0.414414
1	0.445455	0.587719	0.6	0.887446	1	0.622318	1	0.327586	0.38843	1	1	0.36	0.5	0.318584	0.546667	0.864865
1	0.727273	0.421053	-0.026087	0.030303	-0.330472	-0.236052	0.061947	0.793103	0.975207	1	1	0.32	0.875	-0.318584	-0.431111	-0.423423
-0.173913	-0.218182	-0.280702	-0.130435	-0.064935	0.021459	0.175966	-0.584071	-0.025862	-0.479339	-0.439331	0.004255	-0.44	-0.508333	-0.610619	0.057778	-0.45045
1	0.836364	0.412281	-0.078261	-0.056277	-0.347639	-0.175966	0.044248	0.681034	1	1	1	0.208	0.858333	-0.292035	-0.422222	-0.396396
1	0.781818	0.491228	-0.095652	-0.125541	-0.27897	-0.133047	0.088496	0.75	1	1	1	0.176	0.883333	-0.292035	-0.413333	-0.405405
0.869565	0.763636	0.570175	-0.121739	-0.272727	-0.570815	-0.347639	-0.017699	0.681034	1	1	1	0.24	0.733333	-0.221239	-0.431111	-0.153153
0.808696	0.881818	0.587719	-0.147826	-0.341991	-0.622318	-0.356223	0.026549	0.62069	1	1	1	0.256	0.766667	-0.309735	-0.395556	-0.045045
0.956522	0.790909	0.596491	-0.104348	-0.238095	-0.587983	-0.287554	0.044248	0.560345	1	1	1	0.28	0.8	-0.256637	-0.386667	-0.063063

Şekil 5.6 Parmakizi özellik vektörlerinin normalizasyon sonucu yeni değerleri

5.5.5. Performans fonksiyonun seçimi

YSA'nın öğrenme performansını etkileyen önemli hususlardan birisi de performans fonksiyonudur (Rumelhart ve ark. 1985). İleri beslemeli ağlarda performans fonksiyonu olarak genellikle karesel ortalama hata (MSE) kullanılır. Karesel ortalama hata, istenen sonuç ile hesaplanan sistem çıkışı arasındaki farkın kareleri toplamının ortalaması olup aşağıdaki formülle hesaplanır:

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(y_i^{'} - y_i)^2 \qquad (5.7)$$

Burada y, gerçek çıkış değerini; $y^{'}$ ise sistem tarafından hesaplanan çıkış değerini ifade eder.

İleri beslemeli ağlarda kullanılan performans fonksiyonlarından bir diğeri de toplam karesel hata (SSE)'dır. Toplam karesel hata, aşağıdaki eşitlik kullanılarak hesaplanır:

$$SSE = \sum_{i=1}^{N} (y_i^{'} - y_i)^2 \tag{5.8}$$

Bu ağlarda kullanılan diğer bir performans fonksiyonu da karesel ortalama hatanın karekökü (RMS) fonksiyonudur ve aşağıdaki formül ile hesaplanır:

$$RMS = \sqrt{\frac{1}{N}} \sqrt{\sum_{i=1}^{N} (y_i^{'} - y_i)^2} \tag{5.9}$$

Performans fonksiyonuna ağın biaslarının ve ağırlıklarının kareleri toplamının ortalama değerini ifade eden bir terim eklenerek genelleştirme iyileştirilebilir (Sağıroğlu ve ark. 2003). Böylece ağ daha hassas hale getirilir ve ağın, eğitim kümesi haricindeki veriler için sağlıklı bir genelleme yapması engellenebilir.

Genelleştirmeyi iyileştirmek için kullanılan yöntemlerden birisi de öğrenmeyi erken sonlandırmaktır. Bu yöntemde, YSA'ya uygulanacak veri kümesi eğitim, doğrulama ve test olmak üzere üç bölüme ayrılır. İlk küme YSA'nın eğitiminde kullanılarak ağırlıkların, biasın ve eğimin güncelleştirilmesinde kullanılır. İkinci bölüm veri kümesi ise doğrulama olarak adlandırılan ve YSA'nın eğitimi sırasında YSA yapısının performansını hesaplamak üzere kullanılan doğrulama veri kümesidir. Eğitim süresince, eğitim verilerinden hesaplanan hata oranı düştüğü gibi test veri kümesine ait hata da normal şartlarda düşer. YSA, eğitim veri kümesi üzerinde bir aşırı öğrenme eğilimine girerse, doğrulama veri kümesinden hesaplanan hata değeri artmaya başlar (Sağıroğlu ve ark. 2003). Bu çalışmada doğrulama veri kümesinden hesaplanan hata değeri belirlenmiş bir eşik değerini aştığı durumlarda eğitim durdurulmuştur ve ağırlık ve bias değerleri, doğrulama veri kümesinden elde edilen hataların minimum olduğu andaki değerlerine geri döndürülmüştür.

5.6. Yapay Sinir Ağlarının Sınıflandırılması

YSA'lar yapılarına ve öğrenme algoritmalarına göre sınıflandırılırlar (Şen 2004b). Yapay sinir ağları, yapılarına göre, ileri beslemeli (feedforward) ve geri beslemeli (feedback) ağlar olmak üzere iki şekilde sınıflandırılırlar.

İleri beslemeli bir ağda nöron olarak ifade edilen işlemci elemanlar (İE) genellikle katmanlara ayrılmışlardır (Tosun 1997). İleri beslemeli YSA'da, bir katmandaki hücrelerin çıkışları bir sonraki katmana ağırlıklar üzerinden giriş olarak verilir (Özbay 2004, Özbay ve Karlık 2001). Buradaki İE sayısı tamamen uygulanan problemlerin giriş sayısına bağlıdır (Göknar ve Yıldırım 2005). İE'ler bir katmandan diğer bir katmana bağlantı kurarlarken, aynı katman içerisinde bağlantıları bulunmaz. Giriş katmanı, dış ortamlardan aldığı bilgileri hiçbir değişikliğe uğratmadan ara (gizli) katmandaki hücrelere iletir. Bilgi, ara ve çıkış katmanında işlenerek ağ çıkışı belirlenir (Türkoğlu ve Arslan 1996).

İleri beslemeli ağlara örnek olarak çok katmanlı idrak (Multi Layer Perceptron-MLP) (Bishop 1995) ve lineer vektör kuantizasyon (Learning Vector Quantization-LVQ) ağları verilebilir (Kohonen ve ark. 1988). Herhangi bir problemi çözmek amacıyla kullanılan YSA da, katman sayısı ve orta katmandaki hücre sayısı gibi kesin belirlenememiş bilgilere rağmen nesne tanıma ve sinyal işleme gibi alanların yanı sıra, ileri beslemeli YSA, sistemlerin tanılanması ve denetiminde de yaygın olarak kullanılmaktadır (Nabiyev 2004).

Bir geri beslemeli sinir ağı, çıkış ve ara katmanlardaki çıkışların, giriş birimlerine veya önceki ara katmanlara geri beslendiği bir ağ yapısıdır (Öztemel 2003). Böylece, girişler hem ileri yönde hem de geri yönde aktarılmış olur. Bu çeşit sinir ağlarının dinamik hafızaları vardır ve bir andaki çıkış hem o andaki hem de önceki girişleri yansıtır (Saraoğlu 1998). Bundan dolayı, özellikle önceden tahmin uygulamaları için uygundurlar (Grossberg 1969). Geri beslemeli ağlar çeşitli tipteki zaman serilerinin tahmininde oldukça başarı sağlamışlardır. Bu ağlara örnek olarak Hopfield (Tank ve Hopfield 1987), SOM (Self Organizing Map) (Kohonen 1984), Elman (1990) ve Jordan (1986) ağları verilebilir.

5.7. Parmakizi Tanımada Kullanılan YSA Öğrenme Algoritmaları

Öğrenme algoritmalarının çoğunluğu matematik tabanlı olup YSA ağırlıklarının güncelleştirilmesi için kullanılırlar. Bu algoritmaların birçoğu Hebb (1949) kuralından türetilmiştir. Bu kuralın temelinde "bir nöron diğer bir nörondan giriş alıyorsa ve her iki nöron da aktif ise yani matematiksel olarak aynı işarete sahip ise nöronlar arasındaki ağırlıklar kuvvetlendirilir" ilkesi bulunmaktadır (Sağıroğlu ve ark. 2003).

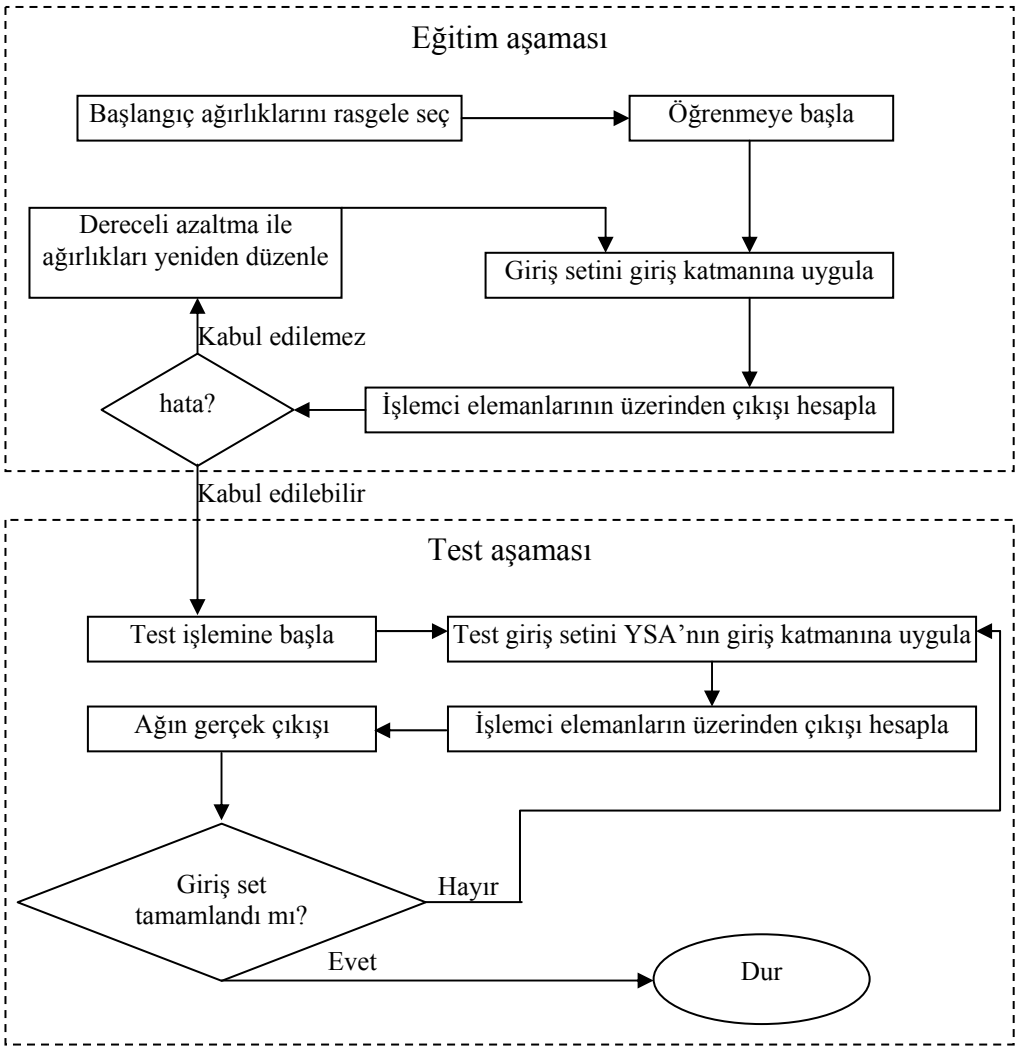
Hopfield kuralında eğer istenilen çıkış ve girişin her ikisi aktif veya her ikisi de pasif ise, öğrenme oranı tarafından bağlantı ağırlığı arttırılır, diğer durumlarda azaltılır.

Delta kuralı, nöronun gerçek çıkışı ile istenilen çıkış değerleri arasındaki farkı azaltma, giriş bağlantılarını güçlendirme ve sürekli olarak değiştirme yani ortalama karesel hatayı, bağlantı ağırlık değerlerinin değiştirilmesiyle düşürme prensibine dayanır. Bu kural geri yayılım, Widrow-Hoff (1960) öğrenme kuralı olarak da bilinir.

Kohonen öğrenme kuralında biyolojik sistemlerdeki öğrenmede olduğu gibi nöronlar öğrenmek için yarışırlar, kazanan nöronun ağırlıkları güncellenir. Bu kural kazanan tamamını alır (winner-takes-all) olarak ta bilinir (Sağıroğlu ve ark. 2003). En büyük çıkışa sahip işlemci nöron kazanır, bu nöron komşularını uyarma ve yasaklama kapasitesine sahiptir. Bu çalışmada öğrenme kuralı olarak kazanan tamamını alır öğrenme kuralı uygulanmıştır. Kazanan tamamını alır ağının amacı, diğer birimlerin katmanlarından gelen en yüksek olasılıklar ile kategoriyi seçmektir (Türkoğlu 2003). En büyük çıkış birimi takviye edilir ve diğer çıkışlar bastırılır, böylece yalnız bir tek çıkış kazandırılır. Ağda hesaplanan birimlere, çıkış ünitelerinin birbiri arasındaki ilişkilere bakılarak ve her bir bireysel çıkış birimi kullanılarak bu işlem başarılır.

En çok tahmin problemlerinde kullanılan YSA için en yaygın olarak kullanılan danışmanlı öğrenme algoritması ise geri yayılım algoritmasıdır.

Geriyayılım algoritması, ileri beslemeli ve çok katmanlı bir ağ mimarisini gerektirmektedir. Bu algoritmada hata, ağdaki ağırlıkların bir fonksiyonu olarak görülür ve hataların kareleri ortalaması, eğim azaltma (gradient descent) yöntemi kullanılarak minimize edilmeye çalışılır (Golden 1986). Bu algoritma, hataları çıkıştan girişe (geriye) doğru azaltmaya çalışmasından dolayı geriyayılım ismini almıştır (Parker 1985, Werbos 1974). Geriyayılım algoritması, çok katmanlı ağları eğitmede en çok kullanılan temel bir öğrenme algoritmasıdır (McClelland ve Rumelhart 1986). Eğitme işlemi ve eğitimden sonraki test işlemi bu akışa göre gerçekleştirilir (Şekil 5.7).



Şekil 5.7 Geri yayılım algoritmasının akış şeması

Geri yayılım algoritması, danışmanlı öğrenme yöntemini kullandığı için öncelikle parmakizi özellik vektörleri ağa öğretilir ve ağa hedef değeri verilir. Öğrenme, her özellik vektörü için ağın çıktı değeri ile hedef değerin karşılaştırılmasıdır. Hata değeri, ağa tekrar geri besleme şeklinde verilir. Özellik vektörü setindeki hata kareleri toplamını azaltmak için nöronlar arasındaki bağlantı ağırlıkları güncellenir (Curram ve Minger 1994).

Burada, gizli katman sayısını tespit etmek zor bir işlemdir. Gizli katmanlar, lineer olmayan ya da değişkenler arasında etkileşim olduğu durumlarda kullanılır. Bu etkileşim ne kadar karmaşıksa, o kadar çok gizli katmana ihtiyaç duyulur. Eğer az sayıda gizli katman kullanılırsa ağ öğrenmeyi başaramayabilir. Gereğinden fazla sayıda gizli katman bulunması halinde ise, ağ ezberler. Bu da ağın, yeni örnekler için genelleme yeteneğini azaltır (Saraç 2004). Amaç, ağın genelleme yeteneğini optimum yapabilecek en az sayıda katmanı kullanmaktır.

Öğrenme algoritması olarak geriyayılım algoritması seçildiğinde iki parametre önem kazanmaktadır (McClelland ve Rumelhart 1986). Bunlar öğrenme katsayısı η ve momentum α'dır. Bu algoritma ile i ve j katman işlem elemanları arasındaki ağırlıklar değişimi olan $\Delta w_{ij}(t)$ hesaplanır. Bu ifade aşağıdaki eşitlik ile gösterilir:

$$\Delta w_{ij}(t) = \eta \delta_j x_i + \alpha \Delta w_{ij}(t-1) \tag{5.10}$$

Burada η, öğrenme katsayısı olup ağırlıkların bir sonraki düzeltmede hangi oranda değiştirileceğini gösterir. Küçük öğrenme katsayıları, ağın sonuca ulaşmasını yavaşlatır. Büyük öğrenme katsayıları ise ağın sonuca daha kısa sürede ulaşmasını sağlar. Öğrenme katsayısı için tipik değerler 0,01 ile 0,9 arasında değişir. Karmaşık ve zor çalışmalar için küçük öğrenme katsayıları seçilebilir.

α ise momentum katsayısını ifade eder. Momentum terimi, ağdaki salınımları engellemeye ve ağın hata yüzeyindeki bölgesel minimum noktalardan kaçarak, daha dip noktalara ulaşmasına yardımcı olur.

δ_j, ara veya çıkış katmanındaki herhangi bir j nöronuna ait bir faktördür. Çıkış katmanı için bu faktör aşağıdaki eşitlikle verilir.

$$\delta_j = \frac{\partial f}{\partial net_j}(y_j^{(t)} - y_j) \tag{5.11}$$

Burada, $net_j = \sum x_j w_{ji}$ ve $y_j^{(t)}$ ise j işlemci elemanının hedef çıkışıdır. Ara katmanlardaki nöronlar için herhangi bir hedef çıkış olmadığından bu eşitlik faktörü aşağıdaki şekilde ifade edilir:

$$\delta_j = \frac{\partial f}{\partial net_j}\sum w_{qi}\delta_q \tag{5.12}$$

Bu duruma bağlı olarak çıkış katmanından başlayarak δ_j faktörü, bütün katmanlardaki işlemci elemanlar için hesaplanır. Daha sonra eşitlik (5.12)'deki formüle bağlı olarak, bütün bağlantılar için ağırlıkların güncelleştirilmesi gerçekleştirilir.

Bu çalışmada danışmanlı öğrenme, çok katmanlı perseptron modeli ve geri yayılım algoritmalarından olan, hızlı yayılım (QP), artışlı geri yayılım (IBP) (Bertsekas 1995), yığın geri yayılım (BBP) (Bertsekas ve Tsitsiklis 1996), eşleştirmeli eğim (CGD) (Shewchuk 1994) algoritmaları kullanılmıştır. Oluşturulan YSA yapıları ve belirlenen parametre değerleri Tablo 5.2'deki gibidir.

Tablo 5.2 Parmakizi tanımada uygulanan YSA yapıları ve parametre değerleri

YSA Yapısı	**Parametre ve değerleri**
Hızlı Geriyayılım	Hızlı yayılım katsayısı = 1,75 Öğrenme oranı = 0,1
Yığın Geriyayılım	Öğrenme oranı = 0,1 Momentum = 0,9
Artışlı Geriyayılım	Öğrenme oranı = 0,1 Momentum = 0,9
Eşleştirmeli Eğim	-

Genel parametre ve ayarlar	
Eğitimi durdurma koşulları	
Eğitim kümesi hata değerine göre	Doğru sınıflandırma oranı: %100
İterasyon sayısına göre	100
Sınıflandırma modeli	Kazanan-tamamını-alır
Ağırlıkları rasgeleleştirme metodu	Otomatik rasgeleleştirme dizisi

5.7.1. Hızlı yayılım algoritması

Hızlı yayılım (Quick propagation) algoritması, Scott Fahlman (1988) tarafından geliştirilen ve Newton metoduna dayanan, çok katlı algılayıcıların eğitilmesi için kullanılan bir sezgisel öğretme algoritmasıdır.

Sezgisel algoritmalar, tecrübeye ve eğitilmiş tahmine dayalı kurallar içermektedir (Moallemi 1991). Bu algoritmalar optimum çözümü garanti edemeyip sadece ona yakın çözümü bulmaktadırlar.

Sezgisel metotları kullanmanın sağladığı bazı avantajlar şöyle özetlenebilir (Sağıroğlu ve ark. 2003):

- Sezgisel metotlar karar verici mekanizma için sadeleştirici olabilir.
- Sezgisel metot, herhangi bir tam yöntemin parçası olarak öğrenme amacıyla kullanılabilir.
- Gerçek dünya problemleri için her zaman matematiksel formülasyon kurmak çok kolay olmayabilir. Bu basitleştirme sonucu oluşan hata, bir sezgisel metodun sağladığı optimale yakın çözümün sahip olduğu hatadan daha büyük olabilir.

Problem uzayında mümkün olan en kısa sürede, bir çözüm bulmak için ağırlık uzayında eğimi son derece azaltmak gerekebilir (Fausett 1994). Bunun yerine çözümden fazla uzaklaşmaksızın mümkün olan en büyük adımlarla kısa sürede çözüme ulaşmak arzu edilir. Hata fonksiyonunun eğim derecesi biliniyorsa, güvenli bir şekilde ağırlık uzayında daha büyük adımla çözüme ulaşılabilir. Bu algoritmada iki geleneksel yaklaşım birleştirilerek çözülmüştür. Bunlar;

- Hesaplamanın geçmişteki durumu hakkında genel veya her bir ağırlık için ayrı sezgilere dayanan dinamik olarak ağırlıkların ayarlanması,
- Her bir ağırlığa göre hatanın ikinci türevinin belirgin olarak kullanımıdır.

Hızlı yayılım algoritması bir ikinci dereceli metottan daha sezgiseldir ve çok az da olsa Newton metoduna dayanan bu metotta aşağıdaki kabuller kullanılır (Sağıroğlu ve ark. 2003):

- Her bir ağırlık için ağırlık hata eğrisi kolları, yukarı doğru açık olan bir parabol ile yaklaştırılabilir.
- Hata eğrisinin eğilimindeki değişim, diğer tüm ağırlıkların aynı andaki değişiminden etkilenmez.

Yukarıda bahsedilen parabol, birbirinden bağımsız her bir ağırlık için mevcut ve önceki hata eğimleri ve bu eğimlerin ölçüldüğü noktalar arasındaki değişim kullanılarak belirlenir. Daha sonra, algoritma doğrudan bu parabolün minimum noktasına dallanır.

Genellikle hızlı yayılım algoritmasının performans testleri diğer tekniklerle karşılaştırıldığında oldukça iyidir (Sağıroğlu ve ark. 1999). Hızlı yayılım algoritması gürültü seviyesi fazla olan verilerin kullanıldığı problemlerde çok iyi çalışır.

Bu algoritmada ağırlık hızlandırma ve kırpılması ihmal edilerek t-1 den t'ye kadar ağırlıklardaki değişim aşağıdaki eşitlik ile ifade edilir (Fausett 1994).

$$\Delta w(t) = \varepsilon L(t) + \alpha Q(t) \tag{5.13}$$

Burada ε, öğrenme katsayısıdır.

$$L(t)=\begin{cases} h(t) & \text{eğer } h(t)h(t\text{-}1)\geq 0 \\ 0 & \text{diğer durumlarda} \end{cases} \tag{5.14}$$

ve

$$Q(t)=\begin{cases} \mu\Delta w(t-1) & \text{eğer } h(t)\left(h(t)-\left(\frac{\mu}{\mu+1}\right)h(t-1)\right)\geq 0 \\ \Delta q(t) & \text{diğer durumlarda} \end{cases} \tag{5.15}$$

eşitlikleri ile verilir.

Bu eşitliklerde μ, momentum büyüme faktörünü; $h(t)=\frac{\partial E}{\partial w(t)}$, eğimi ve $\Delta q(t)=\frac{\Delta w(t)h(t)}{h(t-1)h(t)}$ ise minimum adım miktarını göstermektedir (Haykin 1998). Buradan ağırlık fonksiyonunu güncelleştirmek için, delta ağırlık fonksiyonu ve ağırlık hızlandırma katsayısı işleme dahil edilirse;

$$w(t)=(1-\delta)w(t-1)+\Delta w(t) \tag{5.16}$$

eşitliği elde edilir. Burada δ, hızlandırma katsayısıdır. Son olarak ağırlık çok küçük ise, 0 alınarak kırpılır. Yani, $|w(t)|<\kappa$ ise $w(t)$=0 alınır. Burada κ, ağırlık kırpma faktörüdür.

5.7.2. Yığın geri yayılım

Geri yayılım algoritmasında bütün örnekler gösterildikten sonra eşikler ve ağırlıklar güncellenirse bu işleme "yığın güncelleme" adı verilir (Han ve Kamber 2000). Teoride geri yayılıma ait matematiksel türevim, yığın güncellemeyi kullanır. Bu, geri yayılım algoritmasının gelişmiş bir çeşidi olarak ağ ağırlıklarının her epoktan sonra güncellendiği bir yapıdır. Epok, tüm eğitim verilerinin ağdan geçmesi işlemidir.

Yığın geri yayılım prosedürü aşağıdaki şekilde ifade edilebilir:

1. Ağırlıkları başlangıç durumunda ayarla,
2. Aşağıdaki adımlara devam et,
3. Tüm eğitim verisini eğit,
4. Ağırlıkları güncelle
5. Hata oranı beklenenin altında ise dur, yoksa 2. adıma dön.

Yığın geri yayılım algoritmasını kullanan YSA'nın eğitimi, aşağıdaki durumlarda sonlandırılır:

- Bir önceki epoktaki bütün Δw_{ij}'ler belirli bir eşik seviyesinin altında olduğunda,
- Bir önceki epokta sınıflandırılamamış örnek sayısına ait yüzde, belirli bir eşik seviyesinin altında olduğunda,
- Önceden belirlenmiş bir epok sayısı aşıldığında eğitim sonlandırılır.

5.7.3. Artışlı geri yayılım

Geri yayılım algoritmasının bir çeşidi olarak ağ ağırlıklarının her iterasyondan önce ve mevcut her bir durumdan sonra eğitim kümesinden güncellendiği bir yapıdır (Anthony ve Bartlett 1999). Geniş veri kümeleri için çok tercih edilir. Genellikle artışlı geri yayılım, yığın geri yayılım algoritmasına göre daha performanslıdır. Bu algoritmada ağırlıkları güncelleme kuralı aşağıdaki gibidir:

$$\Delta w_{ji}(t) = -\eta \frac{\partial E(t)}{\partial w_{ji}(t)} + \alpha \Delta w_{ji}(t-1) \qquad (5.17)$$

Burada η, öğrenme oranı ve α, momentum terimidir.

Artışlı geri yayılım prosedürü aşağıdaki şekilde ifade edilebilir:

1. Ağırlıkları başlangıç durumunda ayarla,
2. Aşağıdaki adımlara devam et,
3. Bir eğitim durumunu eğit,
4. Ağırlıkları güncelle
5. Hata oranı beklenenin altında ise dur, yoksa 2. adıma dön.

5.7.4. Eşleştirmeli eğim algoritması

Eşleştirmeli eğim algoritması, çok katmanlı perseptron eğitiminin gelişmiş bir metodudur (Bishop 1995, Shepherd 1997). Geri yayılım algoritmasından daha iyi performans gösterir ve bu algoritmanın kullanıldığı tüm işlemler için kullanılabilir. Çok büyük sayıda ağırlıkları içeren ve çok çıkış nöronlu ağlar için gerekli bir tekniğe sahiptir.

Geriyayılım algoritmaları ağırlık değerlerini eğimin negatifi yönünde dengelemeye çalışırlar. Bu doğrultu eğimin hızla düştüğü doğrultu olup performans fonksiyonu da bu doğrultuda düşer. Performans fonksiyonundaki hızlı düşüş, ağın hızlı yakınsamasına izin vermez. Eşleştirmeli eğim algoritmalarında eğim azaltım yöntemindeki doğrultulardan genellikle daha hızlı yakınsayan eşleştirme doğrultularında bir arama işlemi yapılır (Hagan ve ark. 1996).

Eşleştirmeli eğim algoritması, bir yığın güncelleştirme algoritmasıdır. Geri yayılım algoritması, her bir durumdan sonra ağın ağırlıklarını ayarlar. Buna karşın eşleştirmeli eğim algoritmasında, epokun sonunda ağırlıkları güncellemeden önce tüm durumlardaki ortalama hata eğimi hesaplanır (Luenberger 2003).

Eşleştirmeli eğim algoritması için öğrenim ve momentum oranı belirlemeye ihtiyaç yoktur, bu yüzden kullanımı geri yayılım algoritmasından daha kolaydır.

Eşleştirmeli eğim algoritması önce en dik iniş yönünde çalışır ve yığın tabanlı bir yapı olduğu için her bir eğitim durumunda hata eğimlerinin toplamı hesaplanarak sonuca ulaşılır.

Başlangıç arama yönü $d_0 = -g_0$ şeklindedir. Daha sonra arama yönü Polak-Rebiere formülü kullanılarak güncellenir (Ellsworth ve Moran 2003):

$$D_{j+1} = -g_{j+1} + \beta_j d_j \qquad \beta_j = \frac{g_{j+1}^T (g_{j+1} - g_j)}{g_j^T g_j} \tag{5.18}$$

Arama yönü inişli değilse, algoritma en dik iniş hattını kullanarak yeniden başlar. Burada eğim, bir önceki iterasyondaki eğim değişiminin mevcut iterasyondaki eğim değeriyle olan iç çarpımının önceki eğimin norm karesiyle bölümü ile elde edilir (Fletcher ve Revees 1964).

Eşleştirmeli eğim YSA eğitim algoritması diğer eğim azaltma algoritmalarıyla karşılaştırıldığında ölçüt işlevinin en küçük değerine daha çabuk ve daha kararlı ulaşmasıyla tanınır (Fine 1999). YSA üzerinde bulunan nöronlar arasındaki bağlantıların *t*. iterasyondaki ağırlıklarına $w(t)$ dersek, ağırlıkların (t+1). iterasyondaki değerine aşağıdaki eşitlikle ulaşırız:

$$w(t+1) = w(t) + \eta(t) d(t) \tag{5.19}$$

Burada $\eta(t)$, *t*. iterasyondaki adım büyüklüğü, $d(t)$ ise en küçüklemenin yapılacağı doğrultuyu veren yön vektörüdür. $\eta(t)$'nin değerine $d(t)$ yönünde çizgi en küçüklemesi (line minimization) yapılarak ulaşılır ve $d(t)$, $d(t+1)$ Hessian matrisine göre birbirine dik yön vektörleridir (Bishop 1992).

5.8. Parmakizi Tanımada Çok Katmanlı Ağların Performansı

Bir YSA'nın parmakizi tanıma performansı, öğrenme yeteneği ile anlaşılır. Ağın eğitimi sırasında kendisine gösterilen bütün parmakizi örneklerine doğru cevapları üretmesi, performansının iyi olduğu anlamına gelmez. Bu, ağın öğrenip öğrenmediğini gösterebilir ama iyi öğrenip öğrenmediğini göstermez. Bu nedenle, öğrenen ağların daha önce görmedikleri parmakizi örnekleri karşısında da beklenen performansı gösterip göstermeyeceklerinin ölçülmesi gerekmektedir. Bunun için genellikle ağın eğitildiği problem üzerinden hem eğitimde kullanılacak hem de test esnasında kullanılacak örnekler seçilir (Diamantaras ve Kung 1996). Eğitim sırasında ağa eğitim setindeki örnekler ile birlikte doğrulama seti olarak ayrılan örnekler de gösterilir. Ağ, eğitim setindeki örnekleri öğrenirken hepsi için veya kabul edilebilir oranda hepsi için doğru cevaplar üretmeye başladığında ağa doğrulama setindeki örnekler gösterilir. Ağın performansı bu doğrulama setindeki örnekler karşısında ürettiği doğru cevaplar oranı ile aşağıdaki şekilde ölçülür:

$$P = \frac{D}{T} \times 100 \tag{5.20}$$

Burada D, doğrulama setinden doğru olarak cevaplandırılan örnek sayısını, T ise doğrulama setinde bulunan toplam örnek sayısını, P ise performans oranını göstermektedir. Eğer performans oranı P, istenilen düzeyde veya kabul edilebilir bir değerde değil ise ağ eğitim setindeki bütün örnekleri doğru cevaplasa bile iyi öğrenmiştir denemez (Sağıroğlu ve ark. 2003). O zaman ağı eğitmek için iterasyona devam etmek gerekebilir. Ağın eğitimi için iterasyona devam edilmesine rağmen ağın performansı artmıyor ise o zaman örneklerin problem uzayını iyi temsil edemedikleri veya ağın parametrelerinin veya topolojisinin iyi seçilemediği anlaşılır. Bu yüzden daha önce belirtilen değişiklikleri yaparak ağı yeniden eğitmek gerekebilir.

Ağın eğitilmesi sırasında öğrenme katsayısı ve momentum katsayılarının belirlenmesinde dikkat edilmelidir. Başlangıç değerleri kadar öğrenme ve momentum

katsayılarının belirlenmesi de ağın öğrenme performansı ile yakından ilgilidir. Öğrenme katsayısı ağırlıkların değişim miktarını belirlemektedir. Eğer büyük değerler seçilirse o zaman yerel çözümler arasında ağın dolaşması ve osilasyon yaşaması söz konusu olmaktadır. Küçük değerler seçilmesi ise öğrenme zamanını artırmaktadır. Tecrübeler genellikle 0,1 ile 0,4 arasındaki değerlerin kullanıldığını göstermektedir. Fakat bu tamamen ilgili probleme bağlıdır. Bu değerler iyidir demek de doğru olmaz. Bazı uygulamalarda öğrenme katsayısının 0,6 değerini aldığı zaman en başarılı sonuçları verdiği görülmektedir. Bu çalışmada öğrenme katsayısı 0,1 olduğunda ağın en iyi öğrenme performansını gösterdiği gözlemlenmiştir.

Benzer şekilde momentum katsayısı da öğrenmenin performansını etkiler. Momentum katsayısı bir önceki iterasyondaki değişimin belirli bir oranının yeni değişim miktarına eklenmesi olarak görülmektedir. Bu özellikle yerel çözümlere takılan ağların bir sıçrama ile daha iyi sonuçlar bulmasını sağlamak amacı ile önerilmiştir. Bu değerin küçük olması yerel çözümlerden kurtulmayı zorlaştırabilir. Çok büyük değerler için ise tek bir çözüme ulaşmada sorunlar yaşanabilir. Tecrübeler bu değerin 0,6 ile 0,8 arasında seçilmesinin uygun olacağını göstermektedir. Fakat bu da kesin denilemez. Problemin niteliğine göre kullanıcının belirlemesinde fayda vardır. Daha küçük değerler ile başarılı sonuçların alındığını gösteren örnekleri görmek de mümkündür. Bu çalışmada ilgili metotlar için optimum momentum değeri 0,9 olarak belirlenmiştir.

5.9. YSA Metotlarının Parmakizi Tanıma Performansları ve Analizi

Elde edilen parmakizi özellik vektörleri YSA'nın girişlerini oluşturur. Çıkış olarak parmakizi görüntülerinin kime ait olduğu bilgisi verilmiştir. Bu çalışmada 10 kişiye ait parmakizi görüntüleri kullanılmıştır. Eğitim aşamasında literatürde kullanılan NIST-4 veritabanından 10 kişiye ait 100'er adet parmakizi görüntülerinden elde edilen toplam 1000 adet parmakizi görüntüsü kullanılmıştır. Bu görüntülerden 700'ü eğitim veri kümesi, 150'si doğrulama veri kümesi ve 150'si test

veri kümesi olarak belirlenmiştir. Tecrübeye dayalı arama yöntemi (Ghada ve John 2005) kullanılarak gizli katman nöron sayısı belirlenmiştir. Arama işlemi belirlenen aralıktaki nöron sayılarına göre oluşturulan YSA ağının, test kümesine göre en düşük hata oranını veren yapının belirlenmesiyle sonuçlanır. Bant ve sektör sayılarına göre belirlenen en uygun YSA yapılarından bazıları Tablo 5.3'te gösterilmiş olup elde edilen tüm sonuçlar Ekler kısmında yer almaktadır. Bant ve sektör sayısı arttıkça, bantlar arası mesafe düşük seçilmekte ve parmakizlerinden elde edilen özellik vektörü boyutu artmaktadır. Buna karşılık düşük bant ve sektör sayısı seçildiğinde bantlar arası mesafe artmakta ve parmakizi özellik vektörünün boyutu azalmaktadır. Bu çalışmada bant sayısı 3, 4, 5, 6, 7 ve 8 olarak, sektör sayısı da 8, 12, 16, 20 ve 24 olarak belirlenmiştir. Böylece 30 farklı kombinasyon için en uygun YSA yapısı belirlenmiştir.

Tablo 5.3 Tecrübeye dayalı arama yöntemi ile elde edilen en uygun YSA yapıları

Bant sayısı	**Sektör sayısı**	**En uygun YSA yapıları**
3	12	288:44:10
3	16	384:58:10
4	12	384:58:10
4	16	512:78:10
4	20	640:96:10
5	16	640:96:10
5	20	798:120:10
6	12	576:86:10
6	16	768:116:10
7	8	448:68:10
7	20	1118:168:10
8	16	1023:154:10
8	24	1536:236:10

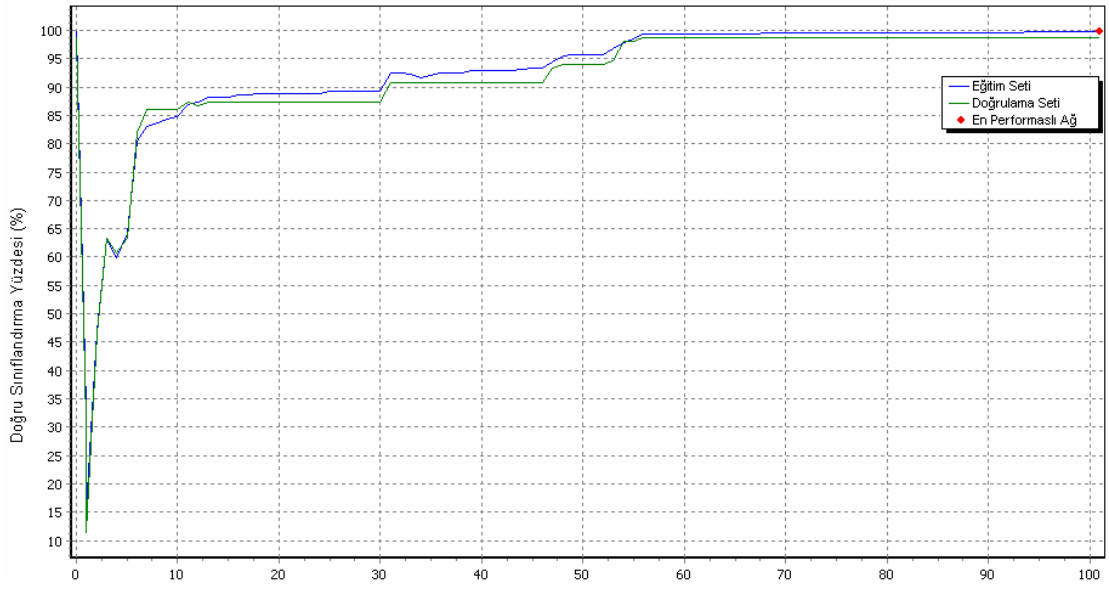
Parmakizi görüntülerine uygulanan görüntü iyileştirme yöntemlerinden olan medyan filtre, dalgacık dönüşümü ve dalgacık dönüşümün geliştirilmişi olan çevritsel dönüşüm uygulanarak iyileştirilmiş parmakizi görüntülerinden elde edilen özellik vektörleri, danışmanlı eğitim ve çok katmanlı perseptron algoritmaları kullanılarak YSA'da eğitilmiştir. Eğitim aşamasına başlamadan önce YSA algoritmalarının parametrelerini belirlemek gerekmektedir. YSA eğitimine başlanmadan önce ağırlık değerleri dizisi rasgele belirlenmiştir. Bu durum ağ girişlerinin dağılımı (normal dağılım, üniform dağılım v.b.) bilinmiyorsa etkilidir. Artışlı geri yayılım ve yığın geri yayılım algoritmaları için her bir iterasyondan sonra öğrenme oranı ve momentum değeri ayarlanmıştır. YSA eğitim ve test sonuçları Tablo 5.4'te gösterilmiştir.

Tablo 5.4 Görüntü iyileştirme metotlarına göre YSA yöntemleri eğitim ve test sonuçları

Görüntü İyileştirme Yöntemi	**Bant ve sektör sayısı**	**Eğitim süresi (ss:dd:ss)**	**YSA yöntemleri**	**YSA yapısı**	**Eğitim veri kümesi %**	**Doğrulama veri kümesi %**	**Test veri kümesi %**	**Tüm veri kümesi %**
Çevritsel	3x16	00:10:06	QP	384:58:10	99,9	97,3	98,0	99,2
Dalgacık	3x16	00:11:06	QP	384:58:10	99,6	96,7	97,3	98,8
Normal	3x16	00:09:39	QP	384:58:10	96,7	92,0	93,3	95,5
Çevritsel	3x20	00:08:21	QP	480:72:10	99,9	97,3	96,0	98,9
Dalgacık	3x20	00:18:45	QP	480:72:10	99,3	96,0	96,7	98,4
Normal	3x20	00:12:51	QP	480:72:10	99,3	94,0	94,0	97,7
Çevritsel	**4x12**	**00:27:02**	**CGD**	**384:58:10**	**100,0**	**98,7**	**98,7**	**99,6**
Dalgacık	4x12	01:12:00	CGD	386:58:10	100,0	96,7	95,3	98,8
Normal	4x12	01:27:59	CGD	385:58:10	99,4	95,3	98,0	98,6
Çevritsel	4x16	00:19:54	BBP	512:78:10	99,1	98,7	95,3	98,5
Normal	4x16	00:16:09	BBP	512:78:10	97,3	94,7	96,0	96,7
Dalgacık	4x16	00:15:20	BBP	512:78:10	96,9	95,3	94,7	96,3
Çevritsel	4x16	01:49:58	CGD	512:78:10	100,0	98,7	95,3	99,1
Dalgacık	4x16	01:30:35	CGD	512:78:10	99,6	96,7	96,7	98,7
Normal	4x16	01:46:38	CGD	512:78:10	99,4	95,3	96,0	98,3
Çevritsel	4x16	00:17:59	QP	512:78:10	99,9	98,7	96,0	99,1

Dalgacık	4x16	00:23:31	QP	512:78:10	99,7	96,7	98,7	99,1
Normal	4x16	00:14:41	QP	512:78:10	99,6	98,0	96,7	98,9
Çevritsel	6x16	00:43:26	BBP	768:116:10	99,1	95,3	98,7	98,2
Normal	6x16	00:37:49	BBP	768:116:10	97,7	97,3	93,3	97,0
Dalgacık	6x16	00:35:49	BBP	768:116:10	92,4	94,0	93,3	92,8
Çevritsel	6x20	00:34:13	IBP	959:144:10	100,0	97,3	98,0	99,3
Normal	6x20	00:11:37	IBP	959:144:10	100,0	96,0	95,3	98,7
Dalgacık	6x20	00:29:04	IBP	959:144:10	100,0	95,3	94,7	98,5
Çevritsel	6x24	01:07:27	CGD	1150:196:10	100,0	96,0	96,7	98,1
Normal	6x24	07:52:09	CGD	1149:196:10	93,4	88,0	88,0	91,8
Dalgacık	6x24	04:51:22	CGD	1149:196:10	89,4	88,7	86,7	88,9
Çevritsel	7x16	04:15:24	CGD	894:134:10	99,9	99,3	98,7	99,6
Dalgacık	7x16	06:06:41	CGD	896:134:10	99,0	94,7	96,7	98,0
Normal	7x16	06:11:04	CGD	896:134:10	95,5	97,3	97,3	95,9
Çevritsel	7x20	00:44:38	BBP	1118:168:10	99,4	98,0	96,7	98,8
Normal	7x20	01:04:24	BBP	1120:168:10	98,9	96,0	96,7	98,1
Dalgacık	7x20	00:56:17	BBP	1120:168:10	98,6	98,0	96,0	98,1
Çevritsel	7x24	02:06:56	QP	1341:224:10	99,9	98,7	97,3	99,3
Dalgacık	7x24	03:09:02	QP	1344:224:10	99,7	97,3	96,7	98,9
Normal	7x24	01:16:21	QP	1344:224:10	99,4	98,7	94,7	98,6
Çevritsel	8x12	00:32:04	IBP	768:116:10	100,0	97,3	98,7	99,4
Dalgacık	8x12	00:18:19	IBP	767:116:10	100,0	96,0	98,7	99,2
Normal	8x12	00:05:02	IBP	767:116:10	100,0	96,0	94,0	98,5
Çevritsel	8x16	00:42:28	BBP	1023:154:10	99,3	96,7	98,7	98,8
Dalgacık	8x16	00:56:52	BBP	1023:154:10	98,0	94,0	96,0	97,1
Normal	8x16	00:52:38	BBP	1023:154:10	97,6	94,0	93,3	96,4
Çevritsel	8x16	00:18:37	IBP	1023:154:10	100,0	97,3	98,7	99,4
Dalgacık	8x16	00:15:40	IBP	1023:154:10	100,0	94,7	96,7	98,7
Normal	8x16	00:12:14	IBP	1023:154:10	99,9	95,3	94,7	98,4
Çevritsel	8x24	00:58:55	BBP	1536:236:10	99,0	98,7	97,3	98,7
Dalgacık	8x24	01:54:47	BBP	1535:236:10	93,6	95,3	93,3	93,8
Normal	8x24	01:46:53	BBP	1535:236:10	82,3	84,7	80,7	82,4
Çevritsel	8x24	00:54:29	IBP	1536:236:10	100,0	98,0	96,7	99,2
Çevritsel	8x24	00:38:20	QP	1536:236:10	99,9	98,7	97,3	99,3
Dalgacık	8x24	03:23:34	QP	1535:236:10	99,9	98,7	94,7	98,9
Normal	8x24	01:15:00	QP	1535:236:10	99,6	97,3	94,7	98,5
Dalgacık	8x24	00:36:51	IBP	1535:236:10	100,0	98,0	95,3	99,0
Normal	8x24	00:30:31	IBP	1535:236:10	100,0	96,0	95,3	98,7

Elde edilen sonuçlara göre en iyi performansı çevritsel dönüşüm kullanılarak iyileştirilmiş parmakizi görüntüleri sağlamıştır. En iyi tanıma oranının göz önüne alındığı bu durumda parmakizi özellik vektörleri, 4 bant ve 12 sektör ile yapılan sektörizasyon sonucu elde edilmiştir. Bu sektörizasyon sonucu bir parmakizine ait 384 adet özellik vektörü elde edilmiştir. Elde edilen 384 özellik vektörü için 384 nöronlu giriş katmanına, 58 nöronlu bir gizli katmana ve 10 nöronlu bir çıkış katmanına sahip YSA yapısına eşleştirmeli eğim öğrenme algoritması uygulanmıştır. Burada YSA eğitim ve test aşamasını toplam 27 dakika 2 saniyede tamamlarken, eğitim için 100 iterasyon gerçekleştirilmesi sonucu eğitim veri kümesi için %100 başarı elde edilmiştir. Test veri kümesi dikkate alındığında başarımın %98,67 ve tüm veri kümesi için ise %99,6 olduğu belirlenmiştir. Şekil 5.8'de en iyi performansı gösteren YSA'nın eğitimi aşamasında eğitim ve doğrulama veri kümelerine ait doğru sınıflandırma oranları (CCR)'na ait grafik gösterilmiştir. YSA'da eğitme sonucunda eğitim veri kümesi, test veri kümesi ve tüm veri kümelerine ait elde edilen eşleştirme matrisleri sırasıyla Şekil 5.9 (a), (b) ve (c)'de gösterilmiştir. Ayrıca YSA ile eğitim sonucu tüm giriş kümeleri için elde edilen eşleştirme oranlarının en yüksek değer olan 1 değerine çok yakın bir değerde, hatta bazı özellik vektörleri için 1 değerinde olduğu gözlemlenmiştir. YSA eğitim sonucu elde edilen eşleştirme oranlarının bir kısmı Tablo 5.5'te gösterilmiştir. Sonuçların geleneksel yöntemlere göre oldukça iyi olduğu gözlemlenmiştir. Parmakizi tanıma aşamasında YSA tarafından elde edilen sonuçlar Ekler kısmında verilmiştir.



Şekil 5.8 YSA eğitim ve doğrulama veri kümeleri doğru sınıflandırma oranları

Ortalama doğruluk oranları (%)

a) Eğitim veri kümesi: % 100

output:	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10
S1	70	0	0	0	0	0	0	0	0	0
S2	0	73	0	0	0	0	0	0	0	0
S3	0	0	71	0	0	0	0	0	0	0
S4	0	0	0	72	0	0	0	0	0	0
S5	0	0	0	0	72	0	0	0	0	0
S6	0	0	0	0	0	71	0	0	0	0
S7	0	0	0	0	0	0	71	0	0	0
S8	0	0	0	0	0	0	0	66	0	0
S9	0	0	0	0	0	0	0	0	68	0
S10	0	0	0	0	0	0	0	0	0	66

b) Test veri kümesi: % 98,67

output:	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10
S1	15	0	0	0	0	0	0	0	0	0
S2	0	14	0	0	0	0	0	0	0	0
S3	0	0	14	0	0	0	0	0	0	0
S4	0	0	0	14	0	0	0	0	0	0
S5	0	0	0	0	14	0	0	0	0	0
S6	0	0	0	0	0	13	0	0	0	1
S7	0	0	0	0	0	0	14	0	0	1
S8	0	0	0	0	0	0	0	17	0	0
S9	0	0	0	0	0	0	0	0	16	0
S10	0	0	0	0	0	0	0	0	0	17

c) Tüm veri kümesi: % 99,6

output:	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10
S1	99	0	0	0	1	0	0	0	0	0
S2	0	100	0	0	0	0	0	0	0	0
S3	0	0	100	0	0	0	0	0	0	0
S4	0	0	0	100	0	0	0	0	0	0
S5	0	0	0	0	100	0	0	0	0	0
S6	0	0	0	0	0	99	0	0	0	1
S7	0	0	0	0	0	0	99	0	0	1
S8	0	0	0	0	0	0	0	100	0	0
S9	0	0	0	0	0	0	0	0	100	0
S10	0	0	0	0	0	0	0	1	0	99

Şekil 5.9 a) Eğitim, b) test ve c) tüm veri kümelerine ait ortalama doğruluk oranları ve eşleştirme matrisleri

Tablo 5.5 Parmakizi özellik vektörlerinin YSA'da eğitildikten sonra test edilmesi sonucu elde edilen eşleştirme oranları

Hedef	Çıkış	Eşleştirme	1: S1	2: S2	3: S3	4: S4	5: S5	6: S6	7: S7	8: S8	9: S9	10: S10
S1	S1	DOĞRU	0,9987	0,0018	0	0	0	0,002	0	0	0,0028	0
S1	S1	DOĞRU	1	0,0019	0	0	0,001	0,0021	0	0	0,0015	0
S1	S1	DOĞRU	0,9972	0,0103	0	0	0,0051	0	0	0	0,0013	0
S1	S1	DOĞRU	0,9973	0,0118	0	0	0,0051	0	0	0	0,0013	0
S1	S1	DOĞRU	1	0,0021	0	0	0	0,0017	0	0	0,0015	0
S1	S1	DOĞRU	0,999	0,0017	0	0	0	0,0021	0	0	0,0022	0
S1	S1	DOĞRU	1	0,0016	0	0	0	0,0029	0	0	0	0
S1	S1	DOĞRU	0,9973	0,0123	0	0	0,0055	0	0	0	0,0013	0
S1	S1	DOĞRU	1	0,0024	0	0	0	0	0	0	0,0021	0
S1	S1	DOĞRU	1	0,0021	0	0	0	0,0073	0	0	0	0
S1	S1	DOĞRU	0,9989	0,0019	0	0	0	0,0013	0	0	0,0025	0
S2	S2	DOĞRU	0	0,9879	0	0	0	0,0014	0,0014	0	0	0
S2	S2	DOĞRU	0	0,9887	0	0	0	0,0014	0,0013	0	0	0
S2	S2	DOĞRU	0	0,9888	0	0	0	0,0013	0,0013	0	0	0
S2	S2	DOĞRU	0	0,9884	0	0	0	0,0014	0,0014	0	0	0
S2	S2	DOĞRU	0	0,9889	0	0	0	0,0013	0,0013	0	0	0
S3	S3	DOĞRU	0,0011	0,0045	0,9887	0,0073	0,0012	0,0012	0	0	0	0
S3	S3	DOĞRU	0,0015	0,0037	0,9898	0,01	0,002	0,0011	0	0	0	0
S3	S3	DOĞRU	0,0015	0,0037	0,9904	0,0114	0,0021	0,0011	0	0	0	0
S3	S3	DOĞRU	0,0014	0,0038	0,9903	0,0113	0,0019	0,001	0	0	0	0
S4	S4	DOĞRU	0	0	0	0,9964	0	0	0,0037	0	0,0019	0
S4	S4	DOĞRU	0	0	0	0,9931	0	0	0	0	0,0092	0
S4	S4	DOĞRU	0	0	0	0,9964	0	0	0,0037	0	0,002	0
S4	S4	DOĞRU	0	0	0	0,9964	0	0	0,0037	0	0,002	0
S4	S4	DOĞRU	0	0	0	0,9965	0	0	0,0039	0	0,0019	0
S5	S5	DOĞRU	0	0	0,0028	0	0,9986	0	0	0	0,0067	0,08
S5	S5	DOĞRU	0	0	0	0	0,9981	0	0	0	0,0015	0
S5	S5	DOĞRU	0	0,0015	0	0	0,9981	0	0	0	0,0018	0
S5	S5	DOĞRU	0	0,0012	0	0	0,9979	0	0	0	0,0019	0
S6	S6	DOĞRU	0	0,0011	0	0	0	0,9956	0	0,0058	0	0
S6	S6	DOĞRU	0	0,0011	0	0	0	0,9965	0	0,004	0	0
S6	S6	DOĞRU	0	0	0	0	0	0,9946	0	0,0026	0	0
S6	S6	DOĞRU	0	0	0	0	0	0,9949	0	0,0026	0	0
S6	S6	DOĞRU	0	0	0	0	0	0,9958	0	0,0048	0	0

S7	S7	DOĞRU	0	0	0	0,0028	0	0	0,9717	0	0,0028	0
S7	S7	DOĞRU	0	0	0	0,0056	0	0	0,968	0	0,0022	0
S7	S7	DOĞRU	0	0	0	0,0074	0	0	0,9583	0	0,0085	0
S7	S7	DOĞRU	0	0	0	0,0072	0	0	0,9637	0	0,0074	0
S7	S7	DOĞRU	0	0	0	0,0047	0	0	0,9539	0	0,0069	0
S8	S8	DOĞRU	0,0051	0	0	0	0	0	0	0,978	0,0495	0
S8	S8	DOĞRU	0	0	0	0	0,00165	0,0012	0	1	0	0
S8	S8	DOĞRU	0,0028	0,0096	0	0,0017	0	0	0,0163	0,9861	0,001594	0
S8	S8	DOĞRU	0,0025	0,0058	0	0	0	0	0,0016	0,9931	0,0129	0
S8	S8	DOĞRU	0	0	0	0	0,001514	0,0011	0	1	0	0
S8	S8	DOĞRU	0	0	0	0,0029	0,0195	0	0	0,9949	0	0
S8	S8	DOĞRU	0,0011	0	0	0,0028	0	0	0	0,9948	0	0

Parmakizi tanımada sisteme dahil olmayan kişilerin sistem tarafından reddedilmesi ve sisteme dahil olan kişilerin de mümkün olduğu kadar sistem tarafından tanınabilirliği, tasarlanan sistemin performansının bir ölçütüdür. Bundan dolayı iki kişinin parmakizi görüntülerine ait özellik vektörleri parmakizi veritabanından çıkarılarak sistem eğitilmiş ve çıkarılan iki kişiye ait parmakizi görüntüleri test aşamasında kullanılmıştır. Elde edilen YSA eğitim ve test sonuçları Tablo 5.6'da verilmiştir.

Burada sekiz çıkışlı YSA'da toplam 7 dakika 46 saniyede eğitilen parmakizi özellik vektörleri için YSA'nın eğitim aşamasında 100 iterasyon gerçekleştirilmesi sonucu elde edilen doğru sınıflandırma oranları, eğitim veri kümesi için %100 ve doğrulama veri kümesi için ise %99,17 oranında elde edilmiştir. Test veri kümesi dikkate alındığında doğru sınıflandırma oranının %97,5 ve tüm veri kümesi için ise %99,6 oranında belirlenmiştir. Şekil 5.10'da YSA'nın eğitim aşamasında eğitim ve doğrulama veri kümelerine ait doğru sınıflandırma oranları (CCR)'na ait grafik gösterilmiştir. YSA'nın eğitim aşamasında eğitim veri kümesine ait tüm parmakizi özellik vektörlerini doğru olarak sınıflandırdığı gözlemlenmiştir. Buna rağmen test aşamasında tüm veri kümesine ait özellik vektörlerinden sadece üçünü hatalı kabul yaparak farklı bir parmakizi olarak eşleştirdiği görülmüştür. Hatalı olarak eşleştirilen parmakizi özellik vektörlerine ait eşleştirme oranları Tablo 5.7'de gösterilmekle

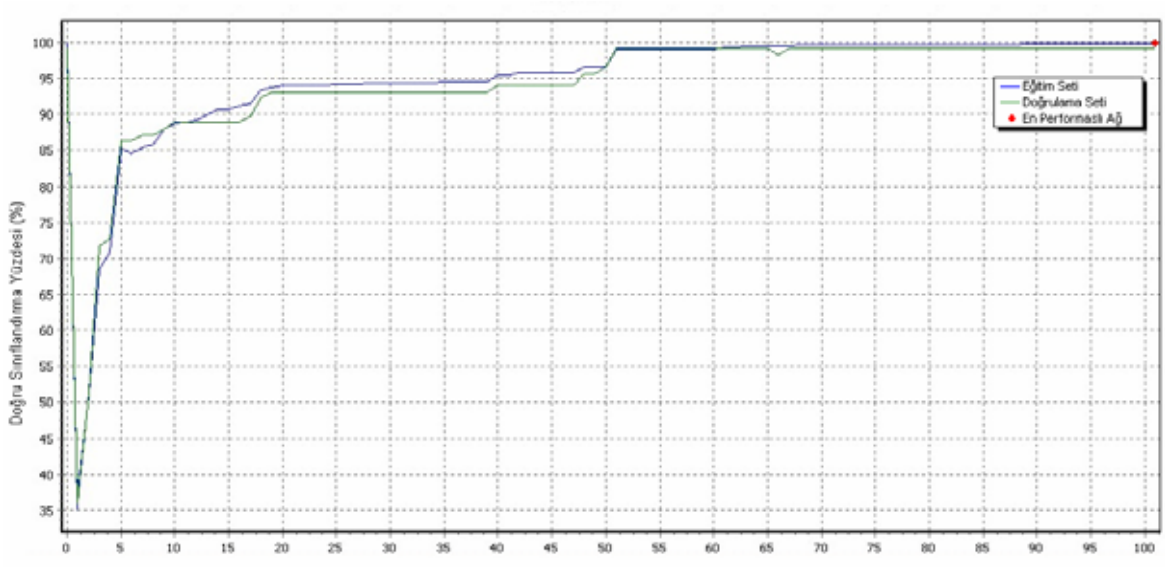
birlikte, bu parmakizi bilgilerinin çok düşük eşleştirme oranları ile hatalı olarak eşleştirildiği gözlemlenmiştir. YSA'da eğitme sonucunda eğitim veri kümesi, test veri kümesi ve tüm veri kümelerine ait elde edilen eşleştirme matrisleri sırasıyla Şekil 5.11 (a), (b) ve (c)'de gösterilmiştir.

Tablo 5.6 Eğitim kümesine dahil olmayan parmakizlerinin YSA'daki test sonuçları

Görüntü İyileştirme Yöntemi	Bant ve sektör sayısı	Eğitim süresi (ss:dd:ss)	YSA yöntemleri	YSA yapısı	Eğitim veri kümesi %	Doğrulama veri kümesi %	Test veri kümesi %	Tüm veri kümesi %
Çevritsel	4x12	00:07:46	CGD	384:38:8	100	99,17	97,5	99,6

Tablo 5.7 Parmakizi özellik vektörlerinin YSA'da eğitildikten sonra test edilmesi sonucu hatalı olarak eşleştirilen özellik vektörleri için elde edilen eşleştirme oranları

Hedef	Çıkış	Eşleştirme	1: S1	2: S2	3: S3	4: S4	5: S5	6: S6	7: S7	8: S8
S1	S5	HATALI	0,006139	0	0	0	0,21774	0	0	0,000038
S6	S4	HATALI	0,001637	0	0	0,414233	0,002534	0,195327	0	0,085778
S7	S4	HATALI	0	0,000015	0	0,000605	0	0,000035	0	0,000023



Şekil 5.10 Sekiz çıkışlı YSA eğitim ve doğrulama kümelerine ait doğru sınıflandırma oranları

Ortalama doğruluk oranları (%)

a) Eğitim veri kümesi: % 100

output:	S1	S2	S3	S4	S5	S6	S7	S8
S1	70	0	0	0	0	0	0	0
S2	0	67	0	0	0	0	0	0
S3	0	0	71	0	0	0	0	0
S4	0	0	0	72	0	0	0	0
S5	0	0	0	0	70	0	0	0
S6	0	0	0	0	0	70	0	0
S7	0	0	0	0	0	0	71	0
S8	0	0	0	0	0	0	0	69

b) Test veri kümesi: % 97,5

output:	S1	S2	S3	S4	S5	S6	S7	S8
S1	15	0	0	0	1	0	0	0
S2	0	14	0	0	0	0	0	0
S3	0	0	14	0	0	0	0	0
S4	0	0	0	14	0	0	0	0
S5	0	0	0	0	16	0	0	0
S6	0	0	0	1	0	15	0	0
S7	0	0	0	1	0	0	14	0
S8	0	0	0	0	0	0	0	15

c) Tüm veri kümesi: % 99,6

output:	S1	S2	S3	S4	S5	S6	S7	S8
S1	99	0	0	0	1	0	0	0
S2	0	100	0	0	0	0	0	0
S3	0	0	100	0	0	0	0	0
S4	0	0	0	100	0	0	0	0
S5	0	0	0	0	100	0	0	0
S6	0	0	0	2	0	98	0	0
S7	0	0	0	1	0	0	99	0
S8	0	0	0	0	0	0	0	100

Şekil 5.11 Sekiz çıkışlı YSA'ya uygulanan a) eğitim, b) test ve c) tüm veri kümelerine ait ortalama doğruluk oranları ve eşleştirme matrisleri

Eğitim kümesine dahil edilmeyen parmakizlerinin test kümesine dahil edilmesinden dolayı bu parmakizleri için sisteme 0,7 değerinde bir eşik değeri tanımlanmıştır. Böylece eşik değerinin üstündeki eşleştirme sonuçları doğru olarak belirtilmiş, eşik değerinin altında kalan eşleştirme sonuçları için ise YSA'nın çıkışına "diğer (S9)" adı altında bir çıkış eklenmiştir. Eğitim kümesine dahil edilmeyen parmakizlerine ait özellik vektörlerinin test kümesine dahil edilerek test edilmesi sonucu elde edilen test kümesi ve tüm kümeye ait eşleştirme matrisleri Şekil 5.12'deki gibidir.

Eşleştirme matrisi dikkate alındığında eğitim kümesine dahil olmayan iki kişiye ait parmakizi bilgileri, test aşamasında farklı kişilerin parmakizleri olarak tanımlanmıştır. Böylece sistemde olmayan parmakizi bilgileri sistemdeki kişilere ait olarak eşleştirildiğinden dolayı sistem, hatalı kabul yapmıştır. Fakat parmakizlerini eşleştirmede eşleştirme oranı dikkate alındığında bu kişilere ait parmakizi bilgilerinin eşleştirme oranlarının, eşik değeri olan 0,7'nin altında olduğu görülmüştür. Dolayısı ile bu kişilere ait parmakizi bilgilerinin tamamı "diğer" olarak tanımlanan YSA çıkışı olarak tanımlandığı görülmüştür. Bu parmakizlerine ait eşleştirme oranlarına eşik değeri uygulandıktan sonra elde edilen eşleştirme matrisi Şekil 5.13'te gösterilmiştir.

output:	S1	S2	S3	S4	S5	S6	S7	S8
S1	15	0	0	0	1	0	0	0
S2	0	14	0	0	0	0	0	0
S3	0	0	14	0	0	0	0	0
S4	0	0	0	14	0	0	0	0
S5	0	0	0	0	16	0	0	0
S6	0	0	0	1	0	15	0	0
S7	0	0	0	1	0	0	14	0
S8	0	0	0	0	0	0	0	15
S9	63	0	0	16	21	0	0	0
S10	4	0	0	34	2	60	0	0

output:	S1	S2	S3	S4	S5	S6	S7	S8
S1	99	0	0	0	1	0	0	0
S2	0	100	0	0	0	0	0	0
S3	0	0	100	0	0	0	0	0
S4	0	0	0	100	0	0	0	0
S5	0	0	0	0	100	0	0	0
S6	0	0	0	2	0	98	0	0
S7	0	0	0	1	0	0	99	0
S8	0	0	0	0	0	0	0	100
S9	63	0	0	16	21	0	0	0
S10	4	0	0	34	2	60	0	0

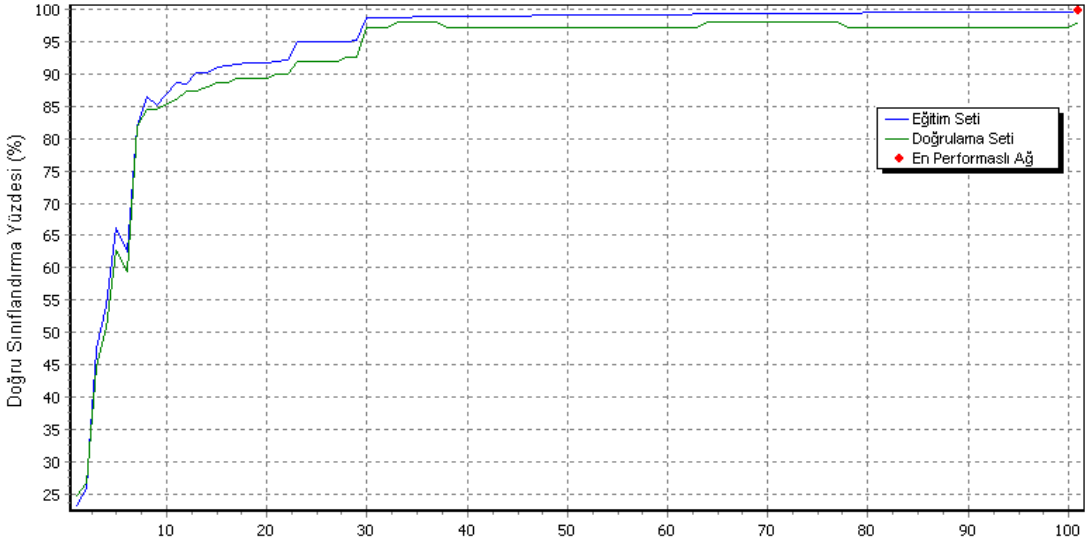
Şekil 5.12 Sekiz çıkışlı YSA'da eğitme aşamasına dahil edilmeyip test kümesine dahil edilen parmakizlerine ait test kümesi ve tüm kümeye ait eşleştirme matrisleri

output:	S1	S2	S3	S4	S5	S6	S7	S8	Diğer
S1	15	0	0	0	1	0	0	0	0
S2	0	14	0	0	0	0	0	0	0
S3	0	0	14	0	0	0	0	0	0
S4	0	0	0	14	0	0	0	0	0
S5	0	0	0	0	16	0	0	0	0
S6	0	0	0	1	0	15	0	0	0
S7	0	0	0	1	0	0	14	0	0
S8	0	0	0	0	0	0	0	15	0
S9	0	0	0	0	0	0	0	0	100
S10	0	0	0	0	0	0	0	0	100

output:	S1	S2	S3	S4	S5	S6	S7	S8	Diğer
S1	99	0	0	0	1	0	0	0	0
S2	0	100	0	0	0	0	0	0	0
S3	0	0	100	0	0	0	0	0	0
S4	0	0	0	100	0	0	0	0	0
S5	0	0	0	0	100	0	0	0	0
S6	0	0	0	2	0	98	0	0	0
S7	0	0	0	1	0	0	99	0	0
S8	0	0	0	0	0	0	0	100	0
S9	0	0	0	0	0	0	0	0	100
S10	0	0	0	0	0	0	0	0	100

Şekil 5.13 Diğer iki kişinin parmakizlerine ait eşleştirme oranlarının eşik değerinden geçirildikten sonra elde edilen test kümesi ve tüm kümeye ait eşleştirme matrisleri

Eşleştirme matrisi elde edilirken 0,7 olarak belirlenen eşik değeri dikkate alınmıştır. Buna göre sistem son olarak elde edilen bilgilere göre yeniden eğitilmiştir. Burada dokuz çıkışlı YSA 00:09:13 sürede eğitimi tamamlarken, eğitim aşamasında 100 iterasyon gerçekleştirilmesi sonucu eğitim veri kümesi için %100 ve doğrulama veri kümesi için ise %98,67 oranında doğru sınıflandırma oranları elde edilmiştir. Test veri kümesi için bu oran %96,67 ve tüm veri kümesi için ise %99,3 olarak elde edilmiştir. Şekil 5.14'te dokuz çıkışlı YSA'nın eğitimi aşamasında eğitim ve doğrulama veri kümelerinin doğru sınıflandırma oranlarına ait grafik gösterilmiştir. YSA'da parmakizi bilgilerini eğitme sonucunda eğitim veri kümesi, test veri kümesi ve tüm veri kümelerine ait elde edilen eşleştirme matrisleri sırasıyla Şekil 5.15 (a), (b) ve (c)'de gösterilmiştir.



Şekil 5.14 Dokuz çıkışlı YSA eğitim ve doğrulama veri kümeleri için doğru sınıflandırma oranları

Ortalama doğruluk oranları (%)

a) Eğitim veri kümesi: % 100

output:	S1	S2	S3	S4	S5	S6	S7	S8	S9
S1	70	0	0	0	0	0	0	0	0
S2	0	73	0	0	0	0	0	0	0
S3	0	0	71	0	0	0	0	0	0
S4	0	0	0	72	0	0	0	0	0
S5	0	0	0	0	72	0	0	0	0
S6	0	0	0	0	0	71	0	0	0
S7	0	0	0	0	0	0	71	0	0
S8	0	0	0	0	0	0	0	66	0
S9	0	0	0	0	0	0	0	0	134

b) Test veri kümesi: % 96,67

output:	S1	S2	S3	S4	S5	S6	S7	S8	S9
S1	15	0	0	0	1	0	0	0	0
S2	0	14	0	0	0	0	0	0	0
S3	0	0	14	0	0	0	0	0	0
S4	0	0	0	14	0	0	0	0	0
S5	0	0	0	0	16	0	0	0	0
S6	0	0	0	1	0	15	0	0	0
S7	0	0	0	1	0	0	14	0	0
S8	0	0	0	0	0	0	0	15	0
S9	0	0	0	0	0	0	1	1	28

c) Tüm veri kümesi: % 99,3

output:	S1	S2	S3	S4	S5	S6	S7	S8	S9
S1	99	0	0	0	1	0	0	0	0
S2	0	100	0	0	0	0	0	0	0
S3	0	0	100	0	0	0	0	0	0
S4	0	0	0	100	0	0	0	0	0
S5	0	0	0	0	100	0	0	0	0
S6	0	0	0	2	0	98	0	0	0
S7	0	0	0	1	0	0	99	0	0
S8	0	0	0	0	0	0	0	100	0
S9	0	0	0	0	0	0	2	1	197

Şekil 5.15 Dokuz çıkışlı YSA için a) eğitim, b) test ve c) tüm veri kümelerine ait ortalama doğruluk oranları ve eşleştirme matrisleri

Daha önce YSA'da eğitime dahil edilmeyen parmakizlerinin test aşamasında eşik değerinden geçirildikten sonra "diğer" çıkışına dahil edilerek yeniden eğitilmesi sonucu bu parmakizi bilgileri için elde edilen doğruluk oranlarından bazıları Tablo 5.8'de gösterilmiştir. Buna göre daha önce bu parmakizleri için elde edilen eşleştirme oranının, eşik değeri olan 0,7 değerinin altında olduğu gözlemlenmişken, "diğer" çıkışına dahil edilerek yeniden eğitilmesi sonucu elde edilen eşleştirme oranlarının eşik değerinin üstüne çıktığı görülmüştür. Bu da sisteme dahil olmayan parmakizi bilgilerinin farklı bir çıkış olarak belirtilmesi ile yeniden eğitilmesi sonucu YSA'nın bu parmakizi bilgilerini ayırt etmede iyi olduğunu göstermektedir.

Tablo 5.8 Dokuz çıkışlı YSA'da "diğer" çıkışına dahil edilen parmakizlerinin yeniden eğitilerek test edilmesi sonucu elde edilen eşleştirme oranları

Hedef	Çıkış	Eşleştirme	1: S1	2: S2	3: S3	4: S4	5: S5	6: S6	7: S7	8: S8	9: S9
S9	S9	DOĞRU	0	0	0	0,0022	0	0	0	0,0016	0,9957
S9	S9	DOĞRU	0	0	0	0,0017	0	0	0	0	0,991
S9	S9	DOĞRU	0	0	0	0,0065	0	0	0	0	0,9505
S9	S9	DOĞRU	0	0	0	0,0037	0	0	0	0	0,9975
S9	S9	DOĞRU	0	0	0,002	0	0	0	0,0026	0	0,9981
S9	S9	DOĞRU	0	0	0	0,002	0	0	0	0,0018	0,9956
S9	S9	DOĞRU	0,0012	0	0,0014	0	0	0	0	0	0,9916
S9	S9	DOĞRU	0	0	0	0	0	0	0	0	0,9944
S9	S9	DOĞRU	0	0	0	0,003	0	0	0	0	0,997
S9	S9	DOĞRU	0	0	0	0,0023	0	0	0	0,0012	0,9956
S9	S9	DOĞRU	0	0	0	0	0	0	0	0	0,9887
S9	S9	DOĞRU	0	0	0,0024	0	0	0	0,0015	0	0,9963
S9	S9	DOĞRU	0	0	0	0,0035	0	0	0	0	0,9954
S9	S9	DOĞRU	0	0	0	0	0	0	0	0	0,9917
S9	S7	HATALI	0	0,0099	0	0,0517	0	0	0,0622	0	0,0261
S9	S9	DOĞRU	0,0012	0	0,0011	0,0011	0	0	0	0	0,9929
S9	S9	DOĞRU	0	0	0	0	0	0	0,002	0	0,9876
S9	S9	DOĞRU	0,0011	0	0	0,0172	0	0	0	0	0,9929
S9	S9	DOĞRU	0	0	0	0,0023	0	0	0	0	0,9904
S9	S9	DOĞRU	0	0	0	0,0015	0	0	0	0	0,9916

Burada sisteme dahil edilmeyen iki kişiye ait parmakizlerinin test edilmesi sonucu YSA tarafından doğru bir şekilde tanınması sistem açısından önemlidir. Fakat oluşturulan sistemin adaptif ve esnek bir yapı haline getirilmesi için kişilere ait parmakizlerinin kişi bazında sisteme dahil edilmesi gerekir. Bu nedenle tez çalışmasının bu aşamasında YSA eğitimine dahil edilmeyen iki kişiye ait parmakizleri sisteme tek tek dahil edilmiştir. Her yeni kişiye ait parmakizi özellik vektörlerinin sisteme tanıtılması durumunda sistemin göstereceği tepki gözlemlenmiştir. Daha önce yukarıda da bahsedildiği üzere sekiz kişinin parmakizi özellik vektörlerinin eğitildiği YSA'da dokuzuncu kişinin test edilmesi sonucu elde edilen test kümesi ve tüm kümeye ait eşleştirme matrisleri Şekil 5.16'daki gibidir.

Eşleştirme matrisi, YSA eğitimi sonucu eğitim kümesine dahil olmayan dokuzuncu kişiye ait parmakizi bilgilerinin test aşamasında farklı kişilerin parmakizleri olarak tanımlandığını göstermektedir. Böylece YSA eğitimine dahil olmayan parmakizi bilgileri, sanki sistemdeki kişilere aitmiş gibi eşleştirildiğinden dolayı sistem hatalı kabul yapmıştır. Fakat dokuzuncu kişiye ait parmakizlerini eşleştirmede eşleştirme oranlarının, eşik değeri olan 0,7'nin altında olduğu görülmüştür. Dolayısı ile bu kişilere ait parmakizi bilgilerinin tamamı "S9" YSA çıkış değeri olarak tanımlanmıştır. Bu parmakizlerine ait eşleştirme oranlarına eşik değeri uygulandıktan sonra elde edilen eşleştirme matrisi Şekil 5.17'de gösterilmiştir.

output:	S1	S2	S3	S4	S5	S6	S7	S8
S1	15	0	0	0	1	0	0	0
S2	0	14	0	0	0	0	0	0
S3	0	0	14	0	0	0	0	0
S4	0	0	0	14	0	0	0	0
S5	0	0	0	0	16	0	0	0
S6	0	0	0	1	0	15	0	0
S7	0	0	0	1	0	0	14	0
S8	0	0	0	0	0	0	0	15
S9	63	0	0	16	21	0	0	0

output:	S1	S2	S3	S4	S5	S6	S7	S8
S1	99	0	0	0	1	0	0	0
S2	0	100	0	0	0	0	0	0
S3	0	0	100	0	0	0	0	0
S4	0	0	0	100	0	0	0	0
S5	0	0	0	0	100	0	0	0
S6	0	0	0	2	0	98	0	0
S7	0	0	0	1	0	0	99	0
S8	0	0	0	0	0	0	0	100
S9	63	0	0	16	21	0	0	0

Şekil 5.16 Eğitme aşamasına dahil edilmeyip test kümesine dahil edilen dokuzuncu kişinin parmakizi bilgilerine ait test kümesi ve tüm kümeye ait eşleştirme matrisleri

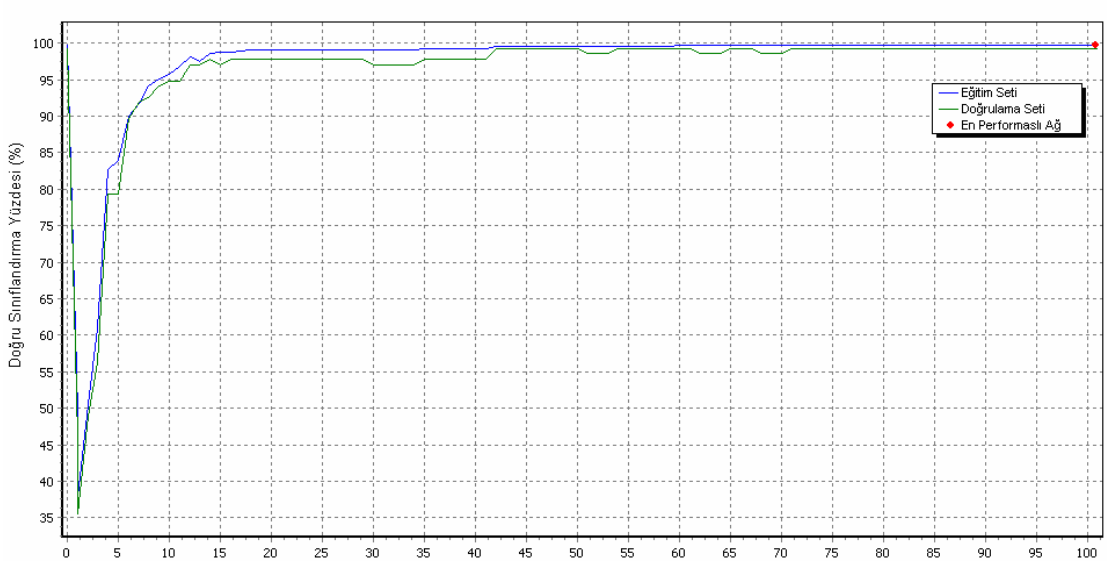
output:	S1	S2	S3	S4	S5	S6	S7	S8	S9
S1	15	0	0	0	1	0	0	0	0
S2	0	14	0	0	0	0	0	0	0
S3	0	0	14	0	0	0	0	0	0
S4	0	0	0	14	0	0	0	0	0
S5	0	0	0	0	16	0	0	0	0
S6	0	0	0	1	0	15	0	0	0
S7	0	0	0	1	0	0	14	0	0
S8	0	0	0	0	0	0	0	15	0
S9	0	0	0	0	0	0	0	0	100

output:	S1	S2	S3	S4	S5	S6	S7	S8	S9
S1	99	0	0	0	1	0	0	0	0
S2	0	100	0	0	0	0	0	0	0
S3	0	0	100	0	0	0	0	0	0
S4	0	0	0	100	0	0	0	0	0
S5	0	0	0	0	100	0	0	0	0
S6	0	0	0	2	0	98	0	0	0
S7	0	0	0	1	0	0	99	0	0
S8	0	0	0	0	0	0	0	100	0
S9	0	0	0	0	0	0	0	0	100

Şekil 5.17 Eğitim aşamasına dahil edilmeyip test kümesine dahil edilen dokuzuncu kişinin parmakizlerine ait eşleştirme oranlarının eşik değerinden geçirildikten sonra elde edilen test kümesi ve tüm kümeye ait eşleştirme matrisleri

Buna göre son olarak dokuz kişiden elde edilen özellik vektörlerine göre sistem yeniden eğitilmiştir. Burada 384:43:9 yapısındaki YSA, 00:07:48 sürede eğitimi tamamlarken, eğitim aşamasında 100 iterasyon gerçekleştirilmesi sonucu eğitim veri kümesi için %100 ve doğrulama veri kümesi için ise %99,3 oranında doğru sınıflandırma oranları elde edilmiştir. Tüm veri kümesi için ise bu oran %99,6 olarak belirlenmiştir. Şekil 5.18'de dokuz çıkışlı YSA'nın eğitimi aşamasında eğitim ve doğrulama veri kümelerinin doğru sınıflandırma oranlarına ait grafik gösterilmiştir. YSA'da parmakizi bilgilerinin eğitilmesi sonucunda eğitim veri kümesi, test veri kümesi ve tüm veri kümelerine ait elde edilen eşleştirme matrisleri sırasıyla Şekil 5.19 (a), (b) ve (c)'de gösterilmiştir.

Eğitim kümesine dahil olmayan kişilerin parmakizi özellik vektörleri test edildiğinde eşleştirme oranlarının 0,7 değerinin altında olduğu görülmektedir. Eşik değeri 0,7 olarak seçilerek bu kişiler sisteme başarılı bir şekilde yeni bir kişi olarak dahil edilmiştir. YSA'ya yeni bir kişi eklenerek yeniden eğitilmesi sonucu tüm verilerin doğruluk oranlarının %100'e yakın çok yüksek değerlerde olduğu gözlemlenmiştir. Böylece YSA'nın adaptif ve esnek bir şekilde yeni kişileri sisteme dahil ederek tanıyabildiği görülmüştür. Sonuç olarak YSA'nın parmakizi tanımada geleneksel yöntemler kadar, hatta daha yüksek oranda başarılı sonuçlar üretmesi bu metodun parmakizi ile kişi tanımada kullanılabilirliğini göstermektedir.



Şekil 5.18 Dokuz çıkışlı YSA eğitim ve doğrulama veri kümeleri için doğru sınıflandırma oranları

Ortalama doğruluk oranları (%)

a) Eğitim veri kümesi: %100

output:	S1	S2	S3	S4	S5	S6	S7	S8	S9
S1	70	0	0	0	0	0	0	0	0
S2	0	67	0	0	0	0	0	0	0
S3	0	0	71	0	0	0	0	0	0
S4	0	0	0	72	0	0	0	0	0
S5	0	0	0	0	72	0	0	0	0
S6	0	0	0	0	0	71	0	0	0
S7	0	0	0	0	0	0	71	0	0
S8	0	0	0	0	0	0	0	66	0
S9	0	0	0	0	0	0	0	0	70

b) Test veri kümesi: % 97,78

output:	S1	S2	S3	S4	S5	S6	S7	S8	S9
S1	15	0	0	0	1	0	0	0	0
S2	0	14	0	0	0	0	0	0	0
S3	0	0	14	0	0	0	0	0	0
S4	0	0	0	14	0	0	0	0	0
S5	0	0	0	0	16	0	0	0	0
S6	0	0	0	1	0	15	0	0	0
S7	0	0	0	1	0	0	14	0	0
S8	0	0	0	0	0	0	0	15	0
S9	0	0	0	0	0	0	0	0	15

c) Tüm veri kümesi: % 99,6

output:	S1	S2	S3	S4	S5	S6	S7	S8	S9
S1	99	0	0	0	1	0	0	0	0
S2	0	100	0	0	0	0	0	0	0
S3	0	0	100	0	0	0	0	0	0
S4	0	0	0	100	0	0	0	0	0
S5	0	0	0	0	100	0	0	0	0
S6	0	0	0	2	0	98	0	0	0
S7	0	0	0	1	0	0	99	0	0
S8	0	0	0	0	0	0	0	100	0
S9	0	0	0	0	0	0	0	0	100

Şekil 5.19 Dokuz çıkışlı YSA için a) eğitim, b) test ve c) tüm veri kümelerine ait ortalama doğruluk oranları ve eşleştirme matrisleri

5.10 Sonuç

Yapay sinir ağları (YSA), insan beynindeki sinirlerin çalışma prensibini taklit ederek bilgi sistemlerine öğrenme, genelleme yapma, hatırlama, tahmin yürütme gibi yetenekler kazandırmayı amaçlayan bilgi işleme metodudur. YSA yöntemlerinin geliştirilmesiyle birlikte desen tanıma olarak parmakizi tanıma problemlerinin YSA yöntemleri ile çözümlenmesi mümkün olduğu görülmüştür. Bu çalışmada parmakizi görüntülerine ait özellik vektörleri YSA'nın giriş setini ve parmakizlerine sahip kişiler ise YSA'nın hedeflenen çıkış setini oluşturmaktadır.

YSA'daki sinir hücrelerinin sayısı, bunların birbirlerine göre konumu ve birbirleri arasındaki sinyallerin akış yönleri YSA yapısını belirlemektedir. Yapısı belirlenen YSA sistemlerinin öğrenme başarısı, test edilerek sınanmalıdır. Bunun için YSA geliştirme sürecinde parmakizi özellik vektörlerinden oluşan veriler üç gruba ayrılır; birinci grup "eğitim kümesi" adını alır ki ağın eğitilmesi için kullanılır, diğer bölümü ise ağın eğitim verileri dışındaki performansını ölçmede kullanılır ve "test kümesi" olarak adlandırılır. Verilerin bir kısmı da mesela YSA'nın gizli katman nöron sayısını belirlemek gibi YSA'nın ağırlık katsayılarından başka mimari parametrelerini düzenlemek için eğitme aşamasında kullanılır. Bu tür verilere "doğrulama kümesi" adı verilir. Doğrulama seti, uygun YSA mimarisini belirlemede kullanılır.

YSA eğitildikten sonra test kümesindeki veriler eğitilen YSA modeline verilir ve YSA'nın hesaplanan çıkış değeri ile istenilen çıkış değeri karşılaştırılır. Amaç, eğitilen YSA modelinin iyi bir genelleme yapıp yapamadığını görmektir. Eğitim ve test aşamalarında istenilen genelleme başarısı elde edilirse YSA modeli kullanılabilir.

YSA eğitim ve test kümesi için elde edilen parmakizi özellik vektörleri, parmakizi görüntüsünden gri seviye değeri olarak elde edildiğinden dolayı 0 ile 255 arasında değerlerden oluşmaktadır. Bu çalışmada 1000 adet parmakizine ait tüm özellik vektörleri kümesinin %70'i eğitim, %15'i doğrulama ve geri kalan %15'i ise

test veri kümesi olarak belirlenmiştir. Bu belirleme işlemi literatürde geçen optimum kriterler göz önünde bulundurularak belirlenmiş ve eğitim veri kümesinin daha az seçildiği durumlarda doğru sınıflandırma oranının da düştüğü gözlemlenmiştir.

Parmakizi özellik vektörlerinden oluşan verilerin normalizasyonu için seçilen yöntem YSA'nın eğitimi sırasındaki performansı doğrudan etkiler. Verilerin normalize edilmesi ile işlemci elemanlarda verilerin kümülatif toplamlarının oluşturacağı olumsuzluklar engellenir. Bu çalışmada özellik vektörleri YSA girişine verilmeden önce [-1 +1] aralığına ölçeklendirilmiştir. Ölçeklendirme işlemi özellik vektörlerinin her bir sütunundaki en düşük ve en yüksek değerler dikkate alınarak yapılmıştır.

Verilerin normalize edilmesi aşamasından sonra YSA yapısının oluşturulması gerekir. YSA'nın giriş katmanındaki giriş nöron sayısı ve çıkış katmanındaki çıkış nöron sayısı parmakizi ve özelliklerine bağlı olduğundan dolayı, burada gizli katman sayısı ve gizli katmandaki nöron sayısı belirlenmelidir. Bunun için deneme-yanılma ile nöron sayısı belirlemenin yanı sıra tecrübeye dayalı arama metodu da iyi sonuçlar vermiştir. Buna göre tüm YSA yapıları için gizli katman sayısı tek olarak belirlenmiş, fakat gizli katmandaki nöron sayısı YSA yapısına göre farklı değerlerde belirlenmiştir.

Bu çalışmada danışmanlı öğrenme, çok katmanlı ağ modeli ve geri yayılım (BP) algoritmalarından olan, hızlı yayılım (QP), artışlı geri yayılım (IBP), yığın geri yayılım (BBP), eşleştirmeli eğim (CGD) algoritmaları kullanılmıştır. Bu algoritmaların her birisi için parametre değerleri optimum olarak belirlenmiştir. Çalışmada kullanılan görüntü iyileştirme yöntemlerinden olan medyan filtre, dalgacık dönüşümü ve çevritsel dönüşüm uygulanarak iyileştirilmiş parmakizi görüntülerinden özellik vektörleri elde edilirken 6 farklı değerde bant sayısı ve 5 farklı değerde sektör sayısı dikkate alınarak toplam 30 farklı kombinasyonda elde edilen parmakizi özellik vektörleri, 4 YSA metoduna uygulanmıştır. Böylece toplam 120 adet parmakizi tanıma işlemi gerçekleştirilmiştir. 3.2 GHz işlemci saat frekanslı, 1 GB DDR hafızaya sahip bir bilgisayar konfigürasyonunda gerçekleştirilen bu parmakizi tanıma işlemlerinden en iyi başarımın %100 YSA eğitim oranı ile 4 bant ve 12 sektörden oluşan parmakizi özellik vektörlerinin 384:58:10 YSA yapısındaki

eşleştirmeli eğim (CGD) metodunda yaklaşık 27 dakika süren eğitim ve test sonucunda elde edildiği gözlemlenmiştir.

Ayrıca YSA eğitim sonucu tüm giriş kümeleri için elde edilen eşleştirme oranlarının en yüksek değer olan 1 değerine çok yakın bir değerde, hatta bazı özellik vektörleri için 1 değerinde olduğu belirlenmiştir.

Parmakizi tanımada sisteme dahil olmayan kişilerin sistem tarafından reddedilmesine karşın sisteme dahil olan kişilerin de sistem tarafından reddedilmemesi, tasarlanan sistemin performansının bir ölçütüdür. Bunun tespiti için iki kişinin parmakizlerine ait özellik vektörleri parmakizi veritabanından çıkarıldıktan sonra mevcut veriler kullanılarak sekiz çıkışlı YSA eğitilmiştir. Bu eğitim aşamasında 100 iterasyon gerçekleştirilmesi sonucu elde edilen doğru sınıflandırma oranları, eğitim veri kümesi için %100 ve doğrulama veri kümesi için ise %99,2 oranında elde edilmiştir. Test veri kümesi dikkate alındığında doğru sınıflandırma oranının %98,3 ve tüm veri kümesi için ise %99,6 oranında belirlenmiştir. Böylece YSA'nın eğitim aşamasında eğitim veri kümesine ait tüm parmakizi özellik vektörlerini doğru olarak sınıflandırdığı gözlemlenmiştir. Buna rağmen test aşamasında tüm veri kümesine ait özellik vektörlerinden sadece 3'ünü hatalı bir şekilde farklı bir parmakizi olarak eşleştirdiği görülmüştür.

Eğitim kümesine dahil edilmeyen parmakizlerinin test kümesine dahil edilmesinden dolayı sistemde olmayan parmakizleri için sisteme bir eşik değeri tanımlanmıştır. Böylece eşik değerinin üstündeki eşleştirme sonuçları doğru olarak belirtilmiş, eşik değerinin altında kalan eşleştirme sonuçları için sistemin hatalı kabul yaptığı belirlenerek bunlar, YSA'nın çıkışına "diğer" adı altında bir çıkış olarak sisteme dahil edilmişlerdir. Parmakizi eşleştirmede eşleştirme oranı dikkate alındığında bu kişilere ait parmakizlerinin eşleştirme oranlarının, eşik değeri olan 0,7'nin çok altında olduğu görülmüştür. Dolayısı ile bu kişilere ait parmakizi bilgilerinin tamamı "diğer" olarak tanımlanan YSA çıkış değerine eşitlendiği görülmüştür.

Daha önce YSA'da eğitime dahil edilmeyen parmakizlerinin test aşamasında eşik değerinden geçirildikten sonra "diğer" çıkışına dahil edilerek yeniden eğitilmesi

sonucu bu parmakizi bilgileri için elde edilen doğruluk oranlarının eşik değerinin üstüne çıktığı görülmüştür. Bu da sisteme dahil olmayan parmakizi bilgilerinin farklı bir çıkış olarak belirtilmesi ile yeniden eğitilmesi sonucu YSA'nın bu parmakizi bilgilerini ayırt etmede iyi olduğunu göstermektedir.

Kişilerden elde edilen parmakizi sayısı kadar bunlara ait parmakizi özellik vektörlerinin boyutu da parmakizi tanıma için önemlidir. YSA'ya giriş olarak verilen ve FingerCode kodlama sistemi ile elde edilen özellik vektörlerinin boyutlarının büyük olmasından dolayı büyük boyutlu özellik vektörleri ile YSA'yı eğitme aşaması çok zaman almaktadır. Bu problemleri aşmak, yani YSA eğitim süresini azaltmak ve özellik vektörlerini gürültülü veriden arındırmak amacıyla özellik seçimi yapılması gerektiği görülmüştür. Bunu yapabilmek için çeşitli özellik seçimi yöntemleri araştırılmıştır. Bu yöntemler içerisinde Genetik Algoritmalar (GA)'ın özellik seçme aşamasında diğer yöntemlere göre daha verimli olduğu, bundan dolayı YSA ile eğitme başarısını da artırdığı Bölüm 6'da gösterilmiştir.

6. GENETİK ALGORİTMALAR KULLANILARAK PARMAKİZİ ÖZELLİK VEKTÖRLERİNİN AZALTILMASI

Verilerin kalitesi, verilerin kullanılabilirliği açısından anahtar öneme sahiptir. Hatalı ve gürültülü veriler güvenilirliğin artırılması için ön işleme tabi tutulur. Aksi halde hatalı girdi verileri, hatalı çıktılara da sebep olacaktır. Verilerin ön işlenmesi, çoğu durumlarda yarı otomatik olan ve zaman isteyen bir aşamadır. Verilerin sayısındaki artış ve buna bağlı olarak çok büyük sayıdaki verilerin ön işlemeden geçirilmesinin gerekliliği, otomatik veri ön işleme için etkin teknikleri önemli hale getirmiştir. Verilerin ön işleme tabi tutulması, verilen bir veri kümesinden daha anlamlı bilginin çıkarılmasından dolayı uygulanır (Oğuzlar 2003).

Uygulamaların birçoğunda, hangi veri ön işleme tekniklerinin kullanılması gerektiği ve bu tekniklerin belirlenmesi önemli bir aşamadır (Famili ve ark. 1997). Çok sayıda veri ön işleme tekniği vardır. Bunlardan biri olan "veri temizleme", verilerdeki gürültünün giderilmesi ve tutarsızlıkların düzeltilmesi için uygulanır. Bir diğer teknik olan "veri birleştirme" ise farklı kaynaklı verileri uygun bir veri kümesinde birleştirilmesinde kullanılır. Veri kümesini normalleştirme amacıyla "veri dönüştürme" yöntemleri uygulanabilir. Fazla olan bazı değişkenlerin atılması ve birleştirilmesi veya kümeleme yolu ile veri büyüklüğünün azaltılması "veri indirgeme" tekniği ile gerçekleştirilir. Burada adı geçen veri ön işleme tekniklerinin uygulanması ile hem elde edilen sonuçların kalitesi artırılmış hem de harcanacak zamandan kazanılmış olur (Han ve Kamber 2000). Bu çalışmada iyileştirilmiş parmakizi görüntülerinden elde edilen özellik vektörlerinin boyutunun indirgenmesi ve özellik seçimi amacıyla veri temizleme ve veri indirgeme teknikleri kullanılmıştır.

6.1. Veri Azaltma Teknikleri ile Parmakizi Özellik Vektörü Boyutunun Azaltılması

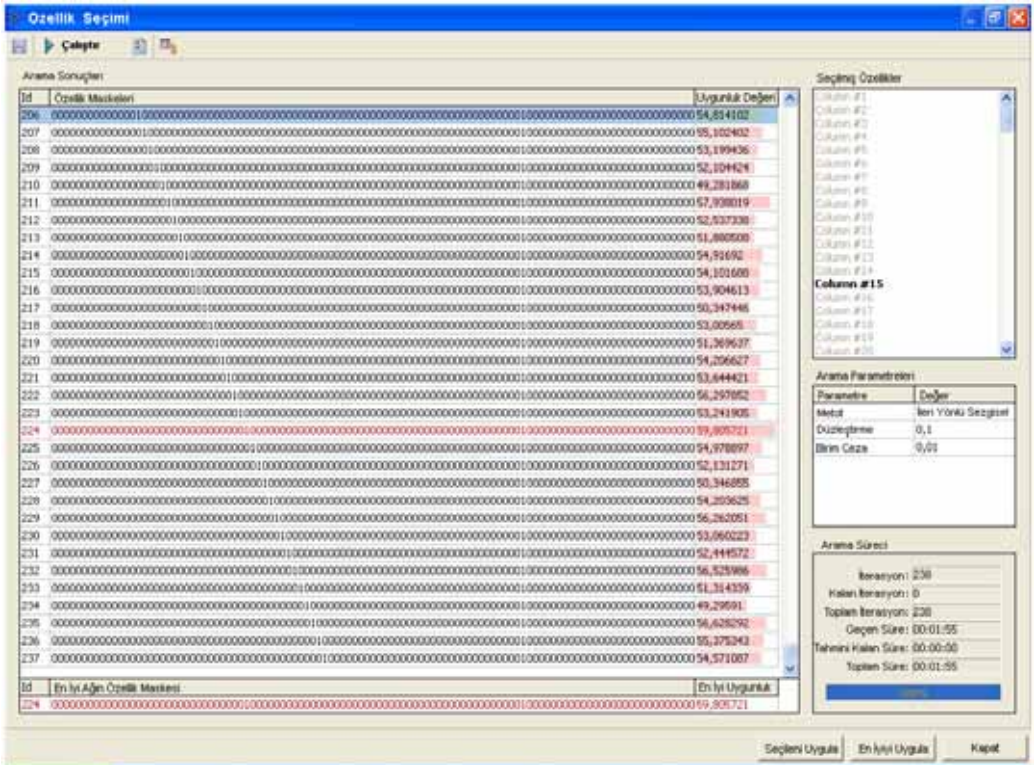
Veri azaltma ya da başka bir deyişle veri indirgeme teknikleri, daha küçük hacimli olarak veri kümesinin indirgenmiş bir örneğinin elde edilmesi amacıyla uygulanır. Bu teknikler ile daha etkin sonuçlar elde edilebilir. Veri indirgeme yöntemleri aşağıdaki biçimde sıralanabilir:

1. Veri birleştirme
2. Boyut indirgeme
3. Veri sıkıştırma
4. Kesikli hale getirme

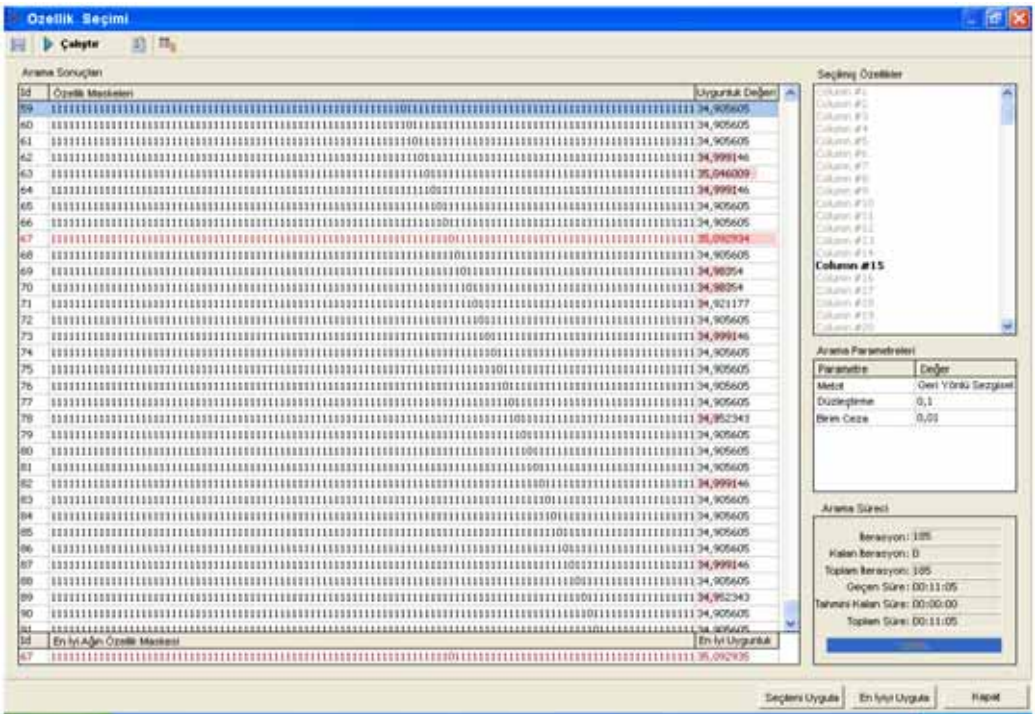
Veri kümesi bazen gereksiz olarak yüzlerce değişken içerebilir. Bu tür gereksiz değişkenler elde edilecek örüntüleri kalitesizleştirir. Bu yüzden gereksiz değişkenlerin elenmesi amacıyla ileri veya geri yönlü olarak sezgisel seçimler yapılabilir. İleri yönlü sezgisel seçimde orijinal değişkenleri en iyi temsil edecek değişkenler belirlenir. Ardından her bir değişken veya değişkenler grubunun, bu kümeye dahil edilip edilmeyeceği sezgisel olarak belirlenir (Şekil 6.1). Geri yönlü sezgisel seçimde ise öncelikle değişkenlerin tüm kümesi ele alınır. Daha sonra gereksiz bulunan değişkenler kümeden çıkarılarak, en iyi değişken kümesi elde edilmeye çalışılır (Şekil 6.2). Boyut indirgeme amacıyla kullanılacak bir diğer yöntem ise karar ağaçlarıdır (Şekil 6.3). Karar ağaçları ele alınacak çıktı değişkenini en iyi temsil edecek değişken kümesini belirler.

Bu metotlardan başka veri azaltma tekniği olarak Genetik Algoritmalar (GA) da kullanılmaktadır. GA, stokastik ve iteratif bir süreçten oluşur (Coşkun 2006). Algoritmanın başlangıcında çözüm uzayı rasgele veya sezgisel bir şekilde belirlenmektedir. GA'nın en önemli özelliği olan rasgelelik, algoritmanın başlangıcından itibaren genetik operatörlerin kullanılmasına kadar sürdürülür. GA

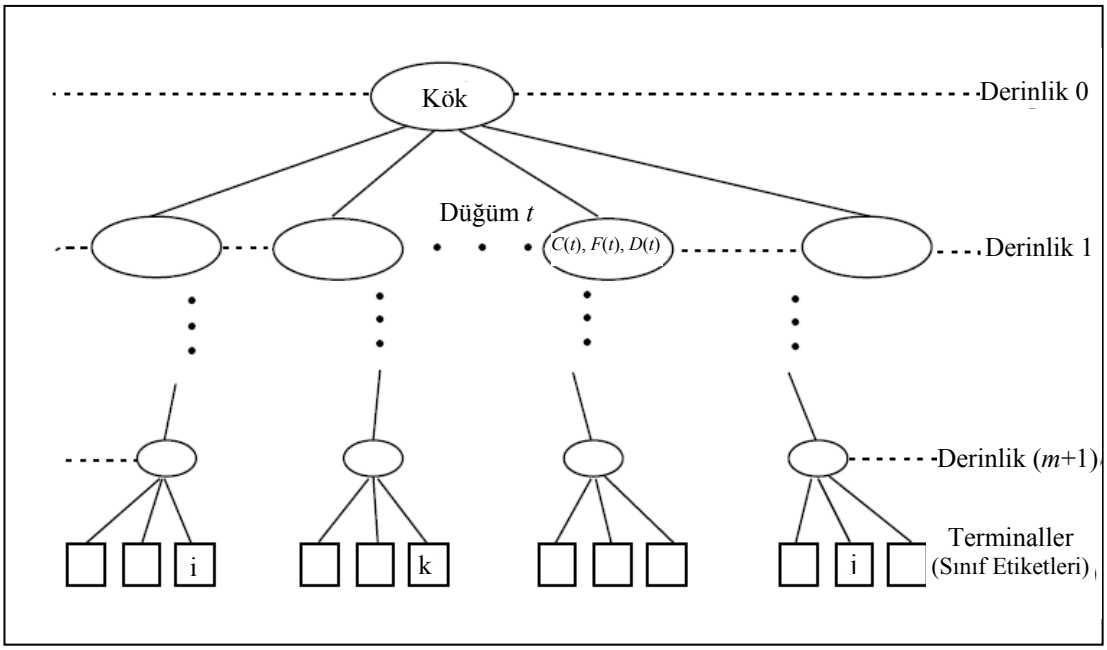
tekrarlanan bir süreçten oluşması özelliği, uygun çözümün bulunmasına kadar algoritmanın işletilmesiyle görülmektedir. Bu çalışmada ileri yönlü sezgisel ve geri yönlü sezgisel özellik seçimiyle birlikte GA kullanılarak veri azaltma işlemi gerçekleştirilmiştir. Elde edilen sonuçlar karşılaştırılmıştır.



Şekil 6.1 İleri yönlü sezgisel seçim kullanılarak en iyi özellikleri seçme işlemi



Şekil 6.2 Geri yönlü sezgisel seçim kullanılarak en iyi özellikleri seçme işlemi



Şekil 6.3 Karar ağaçları seçim tekniğini kullanarak en iyi özellikleri seçme işlemi

$C(t)$, düğüm t'den erişilen sınıfların alt kümesi; $F(t)$, düğüm t'de kullanılan özellik alt kümesi; $D(t)$, düğüm t'de kullanılan karar kuralı

6.2. Genetik Algoritmalar

Genetik algoritmalar (GA), doğadaki evrim yöntemlerini kullanan bir arama yöntemidir (Holland 1975). GA, doğada geçerli olan en iyinin yaşaması kuralına dayanarak sürekli iyileşen çözümler üretir (Goldberg 1985). Bunun için iyinin ne olduğunu belirleyen bir uygunluk fonksiyonu ve yeni çözümler üretmek için yeniden kopyalama, değiştirme gibi operatörleri kullanır. GA'nın bir diğer önemli özelliği de bir grup çözümle uğraşmasıdır. Bu sayede çok sayıda çözümün içinden iyileri seçilip kötüleri elenebilir.

GA'ları diğer algoritmalardan ayıran en önemli özelliklerden biri de seçmedir. GA'da çözümün uygunluğu onun seçilme şansını arttırır, fakat bunu garanti etmez (Goldberg 1989). Seçim de ilk grubun oluşturulması için rasgele olarak belirlenir, ancak bu rasgele seçimde seçilme olasılıklarını çözümlerin uygunluğu belirler. GA'ları diğer metodlardan ayıran noktalar aşağıdaki şekilde sıralanabilir (LeGrand 2004):

- GA, sadece bir arama noktası değil, bir grup arama noktaları üzerinde çalışır. Yani arama uzayında, yerel değil global, bir tek yerden değil bir grup çözüm içinden arama yaparak sonuca ulaşmaya çalışır.

- GA, arama uzayında bireylerin uygunluk değerini bulmak için sadece nesnel uygunluk fonksiyonu ister. Böylelikle sonuca ulaşmak için türev ve diferansiyel işlemler gibi başka bilgi ve kabul kullanmaya gerek duymaz.

- Bireyleri seçme ve birleştirme aşamalarında deterministik kurallar değil rasgele kuralları kullanır.

- Diğer metodlarda olduğu gibi doğrudan parametreler üzerinde çalışmayıp optimize edilecek parametreleri kodlar ve bu kodlar üzerinde işlem yapar.

- Genetik algoritma ne yaptığı konusunda bilgi içermez, nasıl yaptığını bilir.

- Olasılık kurallarına göre çalışırlar. Programın ne kadar iyi çalıştığı önceden kesin olarak belirlenemez. Ama olasıklıkla hesaplanabilir.

- GA, kombinezonsal bir atama mekanizmasıdır.

Bir çok alanda uygulama imkanı ve uygulamaları olan GA'nın işlem adımları aşağıdaki şekilde sıralanabilir (Engin 2001):

- Arama uzayındaki mümkün tüm çözümler dizi olarak kodlanır.

- Başlangıç populasyonu rasgele olarak üretilir.

- Her bir dizi için amaç fonksiyonuna göre uygunluk değeri hesaplanır ve bulunan uygunluk değerleri, populasyonun çözüm kalitesini gösterir.

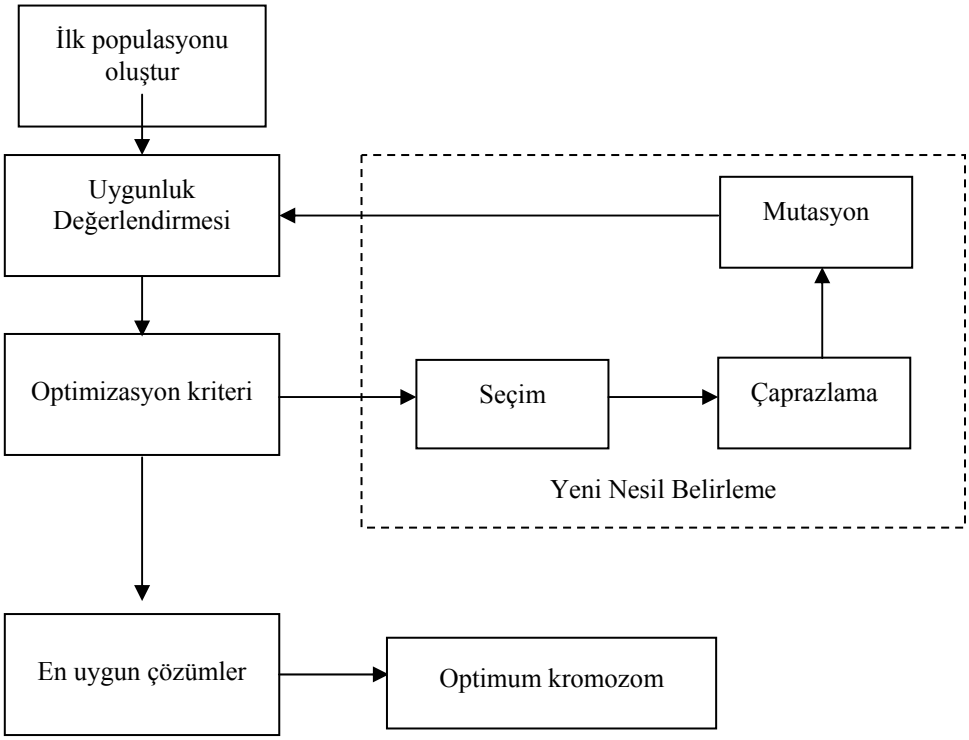
- Bir grup dizi, belirli bir olasılık değerine göre rasgele seçilip çoğalma işlemi gerçekleştirilir.

- Elde edilen yeni bireylerin uygunluk değerleri hesaplanarak, çaprazlama ve mutasyon işlemleri uygulanır.

- Önceden belirlenen nesil sayısı kadar yukarıdaki işlemler devam ettirilir.

- Belirlenen nesil sayısına ulaşan iterasyon sonucu işlem durdurulur ve amaç fonksiyonuna göre en uygun olan dizi seçilir.

Buna göre GA'nın işlem adımlarını gösteren akış şeması Şekil 6.4'te gösterilmiştir. GA, başlangıçta bilinmeyen bir arama uzayından topladığı bilgileri yığıp, daha sonraki aramaları alt arama uzaylarına yönlendirmek için kullanır (Haupt ve Haupt 2004). Böylece belirlenen nesil sayısı veya uygunluk değerine göre durdurulan işlem sonucunda elde edilen en uygun birey seçilir. Arama uzayı büyük seçildiği durumlarda uygun bireyi belirleme işlemi zaman almaktadır. Benzer şekilde nesil sayısı da çok büyük değerlerde seçildiği durumlarda bazen uygunluk fonksiyonu, sürekli aynı değeri üreterek sonsuz döngüye girebilir. Bu yüzden GA için uygun parametrelerin belirlenmesi gerekir.



Şekil 6.4 Genetik algoritmanın akış şeması

6.3. Özellik Vektörlerinin Genetik Algoritma için Kodlanması

Bir problemin çözümü için GA geliştirmenin ilk adımı, tüm çözümlerin aynı boyutlara sahip olacak şekilde kodlanmasıdır. Çünkü bu kodlama, bilginin çerçevesini kesin olarak sınırlayabilir. Bu sayede probleme özgü bilginin bir kromozomsal gösterimle temsili sağlanır (Yeniay 2001). Kromozom, genellikle problemdeki değişkenlerin belli bir düzende sıralanmış halidir. Kromozomun, temsil ettiği çözüm hakkında bilgiyi herhangi bir yolla içermesi gerekir. Kromozomu oluşturmak için sıralanmış her bir değişkene "gen" adı verilir. Buna göre bir gen, kendi başına anlamlı bilgiyi taşıyan en küçük genetik yapıdır.

Kodlama sistemleri, uygulanacak probleme göre belirlenir (Jang 1997). En çok kullanılan ve bu tez çalışmasında da kullanılan kodlama sistemi ikili kodlama sistemidir. İkili kodlamadan başka, permütasyon kodlama, değer kodlama ve ağaç kodlama gibi kodlama sistemleri de mevcuttur.

İkili kodlama: Her kromozom dizisi ikili kodlardan oluşur. Bu kromozomdaki her gen, çözümün belli bir karakteristiğini temsil eder veya tüm kromozom bir sayıyı temsil eder (Mitchell 1996). Bu kodlama sisteminin özellik seçim problemlerinde kullanılması durumunda, kromozomun 0 değerini alması hali "seçilmemiş özellik" olarak ifade edilir. Buna karşın kromozom 1 değerini almışsa bu kromozom "seçilmiş özellik" olarak tanımlanır. Bu kodlama ile ilgili örnek Tablo 6.1'de gösterilmiştir.

Permutasyon kodlama: Burada her kromozom sayıları bir sırada temsil eden bir sayılar dizisidir (Tablo 6.2). Permutasyon kodlama sadece sıralama problemleri için kullanışlı bir yapı oluşturur (Man ve ark. 1999).

Değer kodlama: Reel sayılar gibi karmaşık değerlerin kullanıldığı problemlerde değer kodlanması kullanılabilir (Haupt ve Haupt 2004). Ayrıca karakterlerden veya benzer isimlerden oluşan kümeleri kodlamak amacıyla da kullanılabilir (Tablo 6.3). Bu tip kodlama için probleme özel yeni bazı mutasyon ve çaprazlama işlemleri geliştirmek gerekebilir.

Ağaç kodlama: Bu, gelişen programlar veya ifadeler için kullanılır. Ağaç kodlamada her kromozom, fonksiyonlar ya da programlama dilindeki komutlar gibi bazı nesnelerin bir ağacıdır (Şekil 6.5). Genetik programlamada ve LISP programlama dilinde bu tip kodlama sistemi kullanılır (Langdon ve Poli 2002).

Tablo 6.1 İkili dizi kodlama

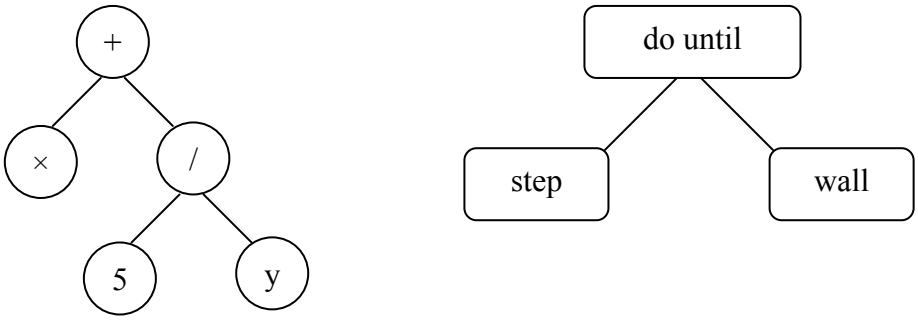
Kromozomlar	**İkili dizi kodlar**
Kromozom 1	1101100100110110
Kromozom 2	1101111000011110

Tablo 6.2 Permutasyon kodlamalı kromozom örnekleri

Kromozomlar	Permutasyon kodlama
Kromozom A	1 5 3 2 6 4 7 9 8
Kromozom B	8 5 6 7 2 3 1 4 9

Tablo 6.3 Değer kodlamalı kromozom örnekleri

Kromozomlar	Değer kodlama
Kromozom A	1,2324 5,3243 0,4556 2,3293 2,4545
Kromozom B	ABDJEIFJDHDIERJFDLDFLFEGT
Kromozom C	(geri), (geri), (sağ), (ileri), (sol)



Kromozom A
(+ x (/ 5 y))

Kromozom B
(do_until step wall)

Şekil 6.5 Ağaç kodlamalı kromozomlar örneği

6.4. İlk Populasyonun Oluşturulması

Özellik vektörlerinin azaltılması amacıyla ilk olarak başlangıç populasyonu oluşturulur. Bunun için öncelikle, olası çözümlerin kodlandığı bir çözüm grubu oluşturulur. Elde edilen çözüm grubu, populasyon ve bu çözümlerin kodları da kromozom olarak adlandırılır. İkili dizi kodlamanın kullanıldığı kromozomların gösteriminde, ilk popülasyonun oluşturulması için rassal sayı üreticileri kullanılabilir. Rastsal sayı üreticisi çağrılır ve değer 0,5'ten küçükse konum 0'a, değilse 1 değerine ayarlanır (Yeo ve Agyel 1996). Bu aşamada populasyonda bulunacak birey sayısı da belirlenir. Bu sayı için bir standart yoktur. Genel olarak önerilen 50 ile 300 aralığında bir büyüklüktür. Bu tez çalışmasında populasyondaki birey sayısı farklı değerlerde seçilerek optimum değer elde edilmeye çalışılmıştır. Gerçekleştirilen araştırmalar sonucunda populasyon boyutunun 100 olarak belirlendiği durumda en uygun sonuç elde edilmiştir. Büyüklük seçiminde yapılan işlemlerin karmaşıklığı ve aramanın derinliği önemlidir (Haupt ve Haupt 2004). Populasyon bu işlemden sonra rasgele bireyler oluşturularak elde edilir.

6.5. Uygunluk Teknikleri

Başlangıç populasyonu bir kez oluşturulduktan sonra populasyondaki her kromozomun uygunluk değeri hesaplanır. Kromozomların ne kadar iyi olduğunu bulan fonksiyona uygunluk fonksiyonu denir. Bu fonksiyonun işletilmesiyle kromozomların uygunluklarının bulunmasına evrim adı verilir. Yüksek ihtimalle uygun olan bu kromozomlar tekrar üreme, çaprazlama ve mutasyon işlemlerine tabi tutulurlar.

Bazı problemler için bireyin uygunluğu, bireyden elde edilen sonuç ile tahmin edilen sonuç arasındaki hatadan bulunabilir. Daha iyi bireylerde bu hata sıfıra

yakın olur. Bu hata genellikle, girişe tekrar sunulacak kombinezonların ortalaması veya toplamıyla hesaplanır. Beklenen ve üretilen değer arasındaki korelasyon etkeni, uygunluk değerini hesaplamak için kullanılabilir (Koza 1992).

Uygunluk fonksiyonu, her bir kromozomun durumunu değerlendirmek için kullanılan bir kaynaktır. Bu, GA ve sistem arasında önemli bir bağlantıdır. Fonksiyon, giriş olarak kodu çözülmüş şekilde kromozomu alır ve kromozomun performansına bir ölçü olarak bir objektif değer üretir. Bu diğer kromozomlar için de yapıldıktan sonra bu değerler kullanılarak, uygun değerler uygunluk fonksiyonuyla hesaplanıp belli bir düzende planlanır. Bu planlamayı sağlayan ve uygunluk teknikleri olarak bilinen birçok yöntem vardır (Kozek ve ark. 1993). Çoğu ortak kullanılan bu yöntemler şunlardır:

1. Pencereleme: Populasyonda en kötü kromozomun objektif değerinin V_w olduğu kabul edilirse her bir i kromozomuna en kötü kromozom arasındaki farkla orantılı bir f_i uygunluk değeri atanabilir. Bu durum matematiksel olarak aşağıdaki şekilde ifade edilir (Tang ve ark. 1996):

$$f_i = c \pm (V_i - V_w) \tag{6.1}$$

Burada V_i, i. kromozomun objektif değeri ve c ise uygunluğun negatif çıkmamasını sağlayacak kadar büyük sabit bir sayıdır. Eğer bir maksimizasyon problemiyle karşılaşılırsa eşitlik (6.1)'deki pozitif işaret kabul edilir. Aksi takdirde minimizasyon gerekliyse negatif işaret kabul edilir.

2. Lineer Normalizasyon: Objektif fonksiyonun maksimize veya minimize durumuna göre kromozomlar, objektif değerinin artma veya azalma düzenine göre sıralanır. En iyi kromozoma rasgele en iyi bir uygunluk f_{best} atanarak sıralanmış düzende i. kromozomun uygunluğu aşağıdaki eşitlik gibi lineer bir fonksiyonla bulunur (Tang ve ark. 1996).

$$f_i = f_{best} - (i-1)d \tag{6.2}$$

Burada *d*, eksilme oranıdır. Bu teknik, populasyonun ortalama objektif değerini ortalama uygunluk içerisinde ayrıntılarıyla planlamayı sağlar.

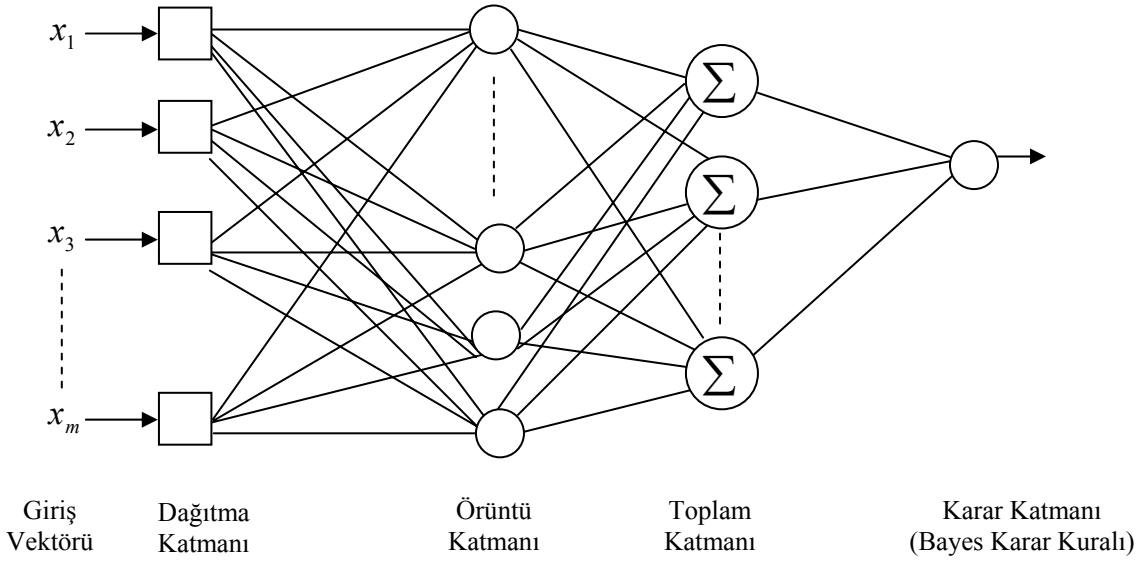
Bu çalışmada uygunluk değerlerinin belirlenmesi amacıyla uygunluk fonksiyonu olarak YSA yöntemlerinden olasılıksal sinir ağı uygulanmıştır.

6.5.1. Olasılıksal sinir ağı

Olasılıksal Sinir Ağları (PNN) ya da olasılık tabanlı sinir ağı (Sağıroğlu ve ark. 2003), ileri beslemeli ağ yapısı olarak standart bir Bayes-Parzen sınıflandırıcısının paralel bir uygulamasıdır (Rao ve Srinivas 2003). Giriş katmanına ek olarak iki gizli katman ve bir çıkış katmanı bulunmaktadır. Hatanın geriye yayılımı algoritmasıyla eğitilen ileri beslemeli bir ağdan en temel farkı, PNN'in eğitim setini sadece bir eğitme adımında kullanmasıdır (Coşkun ve Yıldırım 2003). Diğer ANN metotlarının aksine PNN, Bayes karar stratejisinden ve olasılıksal yoğunluk fonksiyonunun parametrik olmayan çekirdek tabanlı yaklaşımlarından elde edilen sağlam istatistiksel temellere dayanır (Low ve Togneri 1998).

PNN'de bir nörondaki aktivasyon fonksiyonu, eğitim örneklerine dayanan olasılıksal yoğunluk fonksiyonundan (PDF) hesaplanarak türetilmiştir. PNN'in topolojisini oluşturan katmanlar Şekil 6.6'da görüldüğü üzere aşağıdaki gibi şekillendirilirler:

Giriş katmanı *m* boyutlu giriş vektörü uygulanmak üzere toplam *m* nörona sahiptir. İlk katman her giriş örneği için bir model oluşturur. Her bir örüntü katmanı birimi, test vektörü ve eğitim vektörü arasındaki öklit uzaklığını hesaplar. Toplama katmanı, her sınıf için bir toplama işlemi gerçekleştirir. Her toplama işlemi bir önceki katmandaki çıkışlardan bir toplam belirlemek amacıyla kullanılır. Çıkış katmanı, toplamın çıkışlarından en yüksek olasılığı seçerek karar kuralını uygulamak için oluşturulan katmandır. Bu karar katmanı, kazanan alır ağları ile kurulabilir (Avcı ve Yıldırım 2002).



Şekil 6.6 Olasılıksal sinir ağı yapısı

PNN'nin temeli, Parzen pencerelerini kullanarak bilinmeyen durum yoğunluklarını tahmin etmektir (Specht 1990). Parzen'in tekniği, verilen bir kategorinin her bir eğitim vektörü çevresinde *n*-boyutlu bir Gauss fonksiyonunu merkezlemektir. *n* (M_i) artırımlar ile tahmin edilen PDF yaklaşımları, doğru PDF'i verir. Matematiksel olarak, tahmin edilen PDF aşağıdaki şekildedir;

$$f_i(x) = \frac{1}{(2\pi)^{m/2}\sigma^m M_i} \sum_{j=1}^{M_i} \exp\left[-\frac{(x - x_{ij})^T (x - x_{ij})}{2\sigma^2} \right] \tag{6.3}$$

Burada σ yumuşatma faktörü (standart sapma), i sınıf sayısı, j örüntü sayısı, x_{ij} i. sınıfın j. popülasyonun eğitim vektörü, f_i i. sınıf için hesaplanan olasılıksal yoğunluk fonksiyonu için x eğitim vektörünün her birisinde merkezli Gauss toplamı, x test vektörü, M_i i. sınıfın eğitim vektörü sayısı, m x vektörünün boyutu, T transpozeyi simgeler (Low ve Togneri 1998).

Böylece sınıflandırma kararı, Bayes stratejisine uygun olarak verilir ki C_i, *i.* sınıfı göstermek üzere aşağıdaki şekilde ifade edilir:

$$d(x) = C_i \text{ ,eğer } C_i : f_i(x) > f_k(x) \qquad k \neq i \tag{6.4}$$

PNN'de eğitme, parmakizi görüntü vektörlerinin görüntü katmanı birimlerine doğrudan pay edildiği andır. Kodlamanın meydana gelmediği basit durumdur. Bir test vektörü PNN'e verildiğinde, kararda güvenilir bir tahmin kadar iyi olan kazanan kategori çıkışı oluşturur (Türkoğlu 2003).

Bu tez çalışmasında parmakizi özellik vektörlerinin seçilmesi ve seçilmemesi durumu, GA kullanılarak rasgele oluşturulan kromozomun olasılıksal yoğunluk fonksiyonundan geçirildikten sonra elde edilen uygunluk değerine göre belirlenmiştir. Çıkış katmanında elde edilen en iyi uygunluk değerine sahip kromozomun işaret ettiği parmakizi özellik vektörleri, YSA ile eğitimde iyi bir sonuca ulaşabilecek kadar yeterlidir.

6.6. Genetik Operatörler

Kullanılan genetik operatörler, varolan populasyon üzerine uygulanan işlemlerdir. Bu işlemlerin amacı, daha iyi özelliğe sahip yeni nesiller üretmek ve arama algoritmasının alanını genişletmektir. GA'da genel olarak kullanılan 3 tip genetik operatör vardır (Holland 1992). Bunlar:

1. Seleksiyon (Seçim)
2. Çaprazlama
3. Mutasyon'dur.

6.6.1. Seleksiyon (seçim)

Seçim operatörü, varolan nesilden uygun olan bireylerin seçilmesi ve bunların bir sonraki topluluğa kopyalanarak hayatta kalmalarıyla ilgilidir. Seçim işleminde bireyler, bunların uygunluk fonksiyonlarına göre kopya edilirler. Uygunluk fonksiyonundan elde edilen değer, mümkün olduğu kadar yükseltilmesi gereken iyi bir ölçüttür. Topluluk uzayındaki her bir bireyin uygunlukları temel alınarak ne kadar sayıda kopyasının olacağına karar verilir. En iyi bireylerden daha fazla kopya alınır, en kötü bireylerden kopya alınmaz. Bu hayatta kalmak için uygunluk stratejisinin GA'ya sağladığı avantajdır (Karahan ve Ceylan 2004).

GA tarafından üretilen yavruların sayısını belirlemede birkaç yol vardır. Birbirine yakın parametrelerden kaçınmak için uygun bir seçim metodu kullanılmalıdır (Er ve ark. 2005). Seçim başlangıcında basit bir yöntem olan rulet tekerleğiyle bireylerin uygunluk değerleri, bir rulet tekerleğinde hazırlanır. Tekerleğin rasgele döndürülmesinden sonra bireyin bir sonraki nesil için seçilmesi, tekerlek üzerinde kapladığı alan ile doğrudan bağlantılıdır (Engin 2001). Bu yöntem düşük uygunluğa sahip bireylere de seçilme hakkı verir. Bu yöntemde öncelikle eşitlik (6.5)'e göre populasyondaki her bir bireyin olasılıkları hesaplanır ve bu değerler sırayla toplanır. Populasyon boyutuna göre rasgele veriler üretilir ve bu sayılarla olasılık değerleri karşılaştırılarak sonuç populasyonu elde edilir.

$$P_i = \frac{F_i}{\sum_{j=1}^{Ps} F_j} \tag{6.5}$$

Burada P_i *i*. bireyin olasılık değerini, F_i *i*. bireyin uygunluk değerini, P_s ise populasyondaki bireylerin sayısını göstermektedir. Ebeveynler uygunluklarına göre seçilirler. Kromozomlar ne kadar iyiyse, o kadar seçilme şansları fazladır (İşçi ve Korukoğlu 2003). Eğer uygunluklar arası oran çok fazla değişiyorsa, rulet tekerleği ile seçim yöntemi bazı problemlere yol açacaktır. Mesela, en iyi kromozom

uygunluğu tüm rulet tekerleğinin %90'ı ise diğer kromozomların seçilme şansları oldukça az olacaktır.

Bir diğer seçme yöntemi olan rank (sıralama) seçimde ise önce populasyon sıralanır ve her kromozom uygunluğu bu sıralamadan sonra belirlenir. En kötüsü 1 uygunluk değerini almak üzere en iyisi N uygunluk değerini alır ki N de populasyondaki kromozom sayısıdır. Bundan sonra her kromozomun seçilme hakkı olacaktır. Fakat bu metotta en iyi kromozomlar diğerlerinden fazla değişiklik göstermediğinden dolayı daha yavaştır.

Bir başka seçme metodu olan kararlı hal metodu, turnuva seçim metodu olarak da adlandırılır (Davis 1994). Bu seçimin temeli, kromozomların büyük kısmının bir sonraki nesilde hayatta kalmak zorunda olmasıdır. Bu seçme işleminde GA, yeni çocuklar oluşturmak için her nesilde iyi uygunluk değerine sahip birkaç kromozom seçer. Sonra düşük uygunluk değerli kromozomlar atılır ve yeni çocuklar bunların yerine yerleştirilir. Populasyonun geri kalan kısmı yeni nesilde hayattadır. Yani kısaca bu yöntemde alt populasyon oluşturulduktan sonra uygunluklar hesaplanır, en kötü kromozomlar yerlerini başlangıç populasyonundaki en iyi kromozomlara terk ederler.

Elitizm en iyi kromozomun ya da birkaç en iyi kromozomların yeni nesile aynen kopyalanması metodunun adıdır (Weile ve Michielssen 1997). Elitizm, en iyi bulunan çözümü kaybetmeyi önlediğinden dolayı GA'nın performansını arttırır (DeJong 1975). Bu çalışmada rank seçim ile birlikte elitizm kullanıldığından populasyon seçiminde daha performanslı bir yapı elde edildiği gözlemlenmiştir.

6.6.2. Çaprazlama

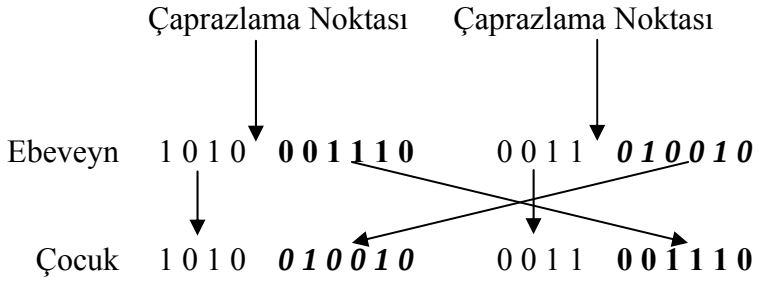
Çaprazlama operatörü, GA'lardaki en önemli operatördür. Amaç, ana kromozom genlerinin yerini değiştirerek çocuk kromozomlar üretmek ve böylece varolan uygunluk değeri yüksek olan kromozomlardan uygunluk değeri daha yüksek

olan kromozomlar elde etmektir (Emel ve Taşkın 2002). Çaprazlama genellikle, belirlenen bir çaprazlama oranına eşit bir olasılıkla seçilen ebeveyn çeşitlerine uygulanır (Jang 1997).

GA'nın performansını etkileyen önemli parametrelerden biri olan çaprazlama operatörü, doğal popülasyondaki çaprazlamaya karşılık gelmektedir (Tablo 6.4). Çoğalma işlemi sonucunda elde edilen yeni populasyondan rastsal olarak iki kromozom seçilir ve çaprazlama işlemine tabi tutulur. Çaprazlama işleminde dizideki gen sayısı L olmak üzere $1 \leq k \leq L-1$ aralığında bir k tamsayısı seçilir. Bu tamsayı değerine göre dizi çaprazlanır. Farklı uygulamalarda farklı kodlama yöntemleri kullanıldığı için farklı çaprazlama yöntemlerinin kullanılması da olasıdır (Poon ve Carter 1995). En basit çaprazlama yöntemi, her iki kromozomun da aynı gen uzunluğunda olmasını öngören tek noktalı çaprazlama yöntemidir (Şekil 6.7). İki noktalı çaprazlamada ise kromozom iki noktadan kesilir ve parçalar karşılıklı olarak yer değiştirilir (Fığlalı 2002).

Tablo 6.4 Biyolojik çaprazlama

Ebeveynler			**Elde edilen yavrular**
AA	BB	→	AABB
aa	bb	→	aabb



Şekil 6.7 Tek noktalı çaprazlama sonucu elde edilen yeni kromozomlar

Çaprazlamadan başka tersinme denilen bir üreme yöntemi daha vardır. Holland (1975) bunu tanımlayarak kromozom uzunluğu çok olan bireylerde çaprazlama yerine bunun kullanılmasını performans açısından önermiştir. Bu tez çalışmasında da kullanılan tersinme, bir kromozomu oluşturan genlerden ardışık bir grubun kendi içerisinde birbirleriyle yer değiştirerek ters dizilmeleri işlemidir. Mesela 011110101 kromozomunda 5. ve 8. genler tersindiğinde yeni elde edilen çocuk kromozom 011101011'dir. Tersinme genellikle kromozom uzunluğu fazla olan populasyonlara uygulanır. Bu çalışmada en düşük kromozom uzunluğu 192 ve en yüksek kromozom uzunluğu 1536'dır. Kromozom uzunluğunun fazla olmasından dolayı çaprazlama işlemi olarak tersinme metodu kullanılmıştır.

6.6.3. Mutasyon

Populasyon gerekli tüm kodlanmış bilgiyi içermez ise çaprazlama tatmin edici çözüm üretmede yetersiz kalabilir. Bundan dolayı mevcut kromozomlardan yeni kromozomlar üretme yeteneğine sahip bir operatöre ihtiyaç duyulur (Braysy 2001). Mutasyon GA'lardaki operasyonda karar verici olarak ikinci derecede rol oynar. Amaç, varolan bir kromozomun genlerinin bir ya da birkaçının yerlerini değiştirerek yeni kromozom oluşturmaktır. Yeniden ve sürekli yeni nesil üretiminde belirli bir süre sonra nesildeki kromozomlar birbirlerini tekrarlayabilir ve bunun sonucunda farklı kromozom üretimi durabilir veya azalabilir (İşçi ve Korukoğlu 2003). İşte bu nedenle nesildeki kromozomlarının çeşitliğini artırmak için kromozomlardan bazıları mutasyona uğratılır. Öyle ki burada ilk populasyon, mümkün olan tüm alt çözümlerin küçük bir alt kümesi olabilir ve ilk populasyondaki tüm kromozomların önemli biti sıfır olabilir. Halbuki o bitin problemin çözümü için 1 olması gerekebilir ve bunu da çaprazlama düzeltemeyebilir. Bu durumda o bit için mutasyon kaçınılmazdır. Bu işlem çaprazlama aşamasından sonra gelir. İkili kodlama sisteminin kullanıldığı problemlerde mutasyon, düşük bir olasılık değeri altında bir bit değerini (0 veya 1) diğer bit değerine (1 veya 0) dönüştürür (Tablo 6.5).

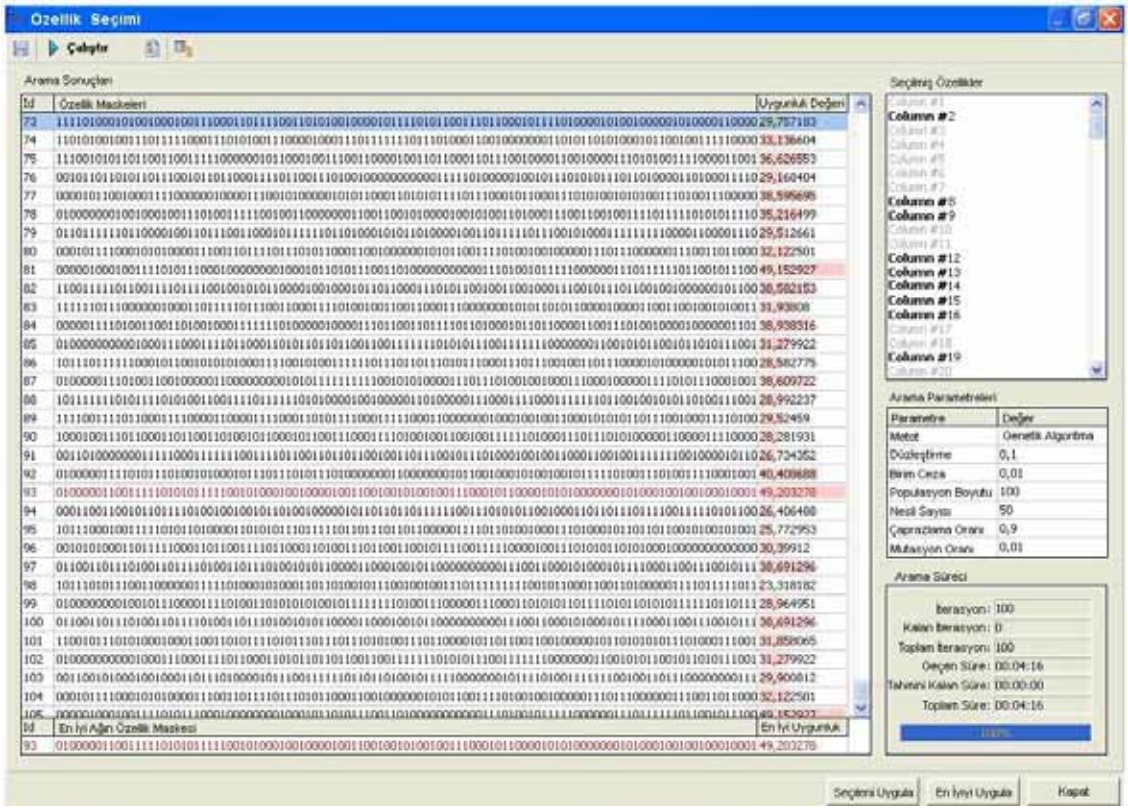
Tablo 6.5 İkili kodlama sistemine uygulanmış mutasyon operatörü

Orijinal çocuk		Mutasyonlu çocuk
110**1**111000011110	⟶	110**0**111000011110
110110**0**100110**1**0	⟶	110110**1**100110**0**0

6.6.4. Yeni nesilin üretilmesi ve özellik seçimi

Yeni nesil, yeniden üretme, çaprazlama ve mutasyon işlemlerinden sonra tanımlanmakta ve bir sonraki nesilin ebeveynlerini oluşturmaktadır. Süreç, yeni nesille çoğalma için belirlenen uygunluk ile devam eder. Bu süreç, önceden belirlenen nesil sayısı kadar veya bir hedefe ulaşılıncaya kadar ya da başka bir durdurma kriteri sağlanana kadardır (Yeo ve Agyel 1996). İstenen hassasiyet derecesine göre de maksimum iterasyon sayısı belirlenebilmekte ve iterasyon bu sayıya ulaştığında döngü durdurulabilmektedir. Durdurma kriteri iterasyon sayısı olabileceği gibi hedeflenen uygunluk değeri de olabilmektedir (Fung ve ark. 2001). Bu çalışmada uygun iterasyon sayısı 100 olarak belirlenmiş ve iterasyon sayısına göre GA'yı durdurma kriteri olarak da kullanılmıştır. Daha sonra elde edilen uygunluk değerlerinin tümüne bakılarak en iyi uygunluk değerine sahip kromozoma göre parmakizlerine ait yeni özellik vektörleri belirlenmiştir. Bu durum Şekil 6.8'de gösterilmiştir.

Tüm bu parametreler belirlenerek parmakizi özellik vektörlerinin azaltılması için GA uygulanmıştır. Şekil 6.8'den de görüldüğü üzere en iyi ağın özellik maskesi 010000011001111……. şeklinde devam eden bir dizi olarak belirlenmiştir. Burada 0 olarak belirlenen özellik vektörü sütunları çıkarılarak 1 değerini içeren özellik vektörleri seçilmiştir. Bu durum şekildeki seçilmiş özellikler alanında da görülmektedir. GA uygulanması sonucu elde edilen en uygun ve azaltılmış özellik vektörleri değerlerinin YSA'da tekrar eğitme işlemine tabi tutularak parmakizi performansı test edilmiştir.



Şekil 6.8 GA ile en uygun kromozom yapısının belirlenmesi

6.7. Özellik Vektörü Azaltmada GA'nın Performansını Etkileyen Etmenler

Parametreler, GA'nın performansı üzerinde önemli etkiye sahiptirler. Kontrol parametreleri olarak adlandırılan bu parametrelerin optimum değerlerini bulmak için bir çok çalışma yapılmıştır, fakat tüm problemler için genel olarak kullanılabilecek parametreler bulunamamıştır (Altıparmak ve ark. 2000). Kontrol parametreleri populasyon büyüklüğü, çaprazlama olasılığı, mutasyon olasılığı, nesil aralığı, seçim stratejisi ve fonksiyon ölçeklemesi olarak sayılabilir. Bu parametreler aşağıda açıklanmıştır (Yeniay 2001, Sinreich ve Samakh 1999):

- Populasyon Büyüklüğü: GA kullanıcısı tarafından verilen en önemli kararlardan birisi olup bu değer çok küçük seçilirse, genetik algoritma yerel bir optimuma takılabilmektedir. Populasyonun çok büyük olması ise çözüme ulaşma zamanını

arttırır. Bu konuda Goldberg (1985), yalnızca kromozom uzunluğuna bağlı bir populasyon büyüklüğü hesaplama yöntemi önermiştir. Ayrıca Schaffer ve ark. (1989), çok sayıda test fonksiyonları üzerinde yaptıkları araştırmalar sonucunda, 20-30 arası bir populasyon büyüklüğünün iyi sonuçlar verdiğini belirtmişlerdir. Bu çalışmada populasyon büyüklüğü 20 ile 640 arasındaki tüm değerler için araştırılmış ve en uygun populasyon büyüklüğü 100 olarak belirlenmiştir.

- Çaprazlama Olasılığı: Çaprazlamanın amacı, mevcut iyi kromozomların özelliklerini birleştirerek daha uygun kromozomlar elde etmektir. Kromozom çiftleri, $P(c)$ olasılığı ile çaprazlamaya uğramak üzere seçilirler. Çaprazlamanın artması, yapı bloklarının artmasına neden olmakta fakat aynı zamanda bazı iyi kromozomların bozulma olasılığını da artırmaktadır. Bu çalışmada çaprazlama olasılığı 0 ile 1 arasındaki değerler sırasıyla 0,1 oranında artırılarak araştırılmış ve en uygun çaprazlama olasılığı olarak 0,9 değeri belirlenmiştir.

- Mutasyon Olasılığı: Mutasyon işleminin uygulanmasıyla populasyondaki genetik çeşitlilik korunur. Mutasyon, $P(m)$ olasılığı ile bir kromozomdaki her bitte meydana gelebilir. Kromozomlar birbirine benzemeye başladığında hala çözüm noktalarının uzağında bulunuyorsa mutasyon işlemi, GA'nın sıkıştığı yerden kurtulmak için tek yoldur. Ancak yüksek bir değer vermek GA'yı kararlı bir noktaya ulaşmaktan alıkoyar. Yani mutasyon olasılığı artarsa, genetik arama rassal bir aramaya dönüşür (Janikow 1993). Fakat bu aynı zamanda kayıp genetik malzemeyi tekrar bulmada yardımcı olabilir. Bu çalışmada mutasyon olasılığı 0 ile 1 arasındaki değerler 0,01 artırılarak araştırılmış ve en uygun mutasyon olasılığı 0,01 olarak belirlenmiştir. Parmakizlerine ait özellik vektörlerinin büyük boyutlu olmasının çaprazlama olasılığının büyük, mutasyon olasılığının ise küçük belirlenmesinde etkili olduğu görülmüştür.

- Nesil Aralığı: Her nesildeki yeni kromozom oranına nesil aralığı denmektedir. Genetik operatörler için kaç tane kromozomun seçildiğini gösterir. Yüksek bir değer birçok kromozomun yer değiştirdiği anlamına gelmektedir.

- Seçim Stratejisi: Eski nesili yenilemenin çeşitli yöntemleri mevcuttur. Nesilsel stratejide, mevcut populasyondaki kromozomlar tamamen yavrular ile yer değiştirir.

Populasyonun en iyi kromozomu da yenilendiğinden dolayı bir sonraki nesile aktarılamaz ve bu yüzden bu strateji, elitizm stratejisiyle beraber kullanılır (Emel ve Taşkın 2002). En uygun stratejisinde, populasyondaki en iyi kromozomlar hiçbir zaman yenilenmemektedir, bundan dolayı çoğalma için en iyi çözüm her zaman elverişlidir. Denge durumu stratejisinde ise, her nesilde yalnızca birkaç kromozom yenilenmektedir. Genellikle, yeni kromozomlar populasyona katıldığında en kötü kromozomlar yenilenir.

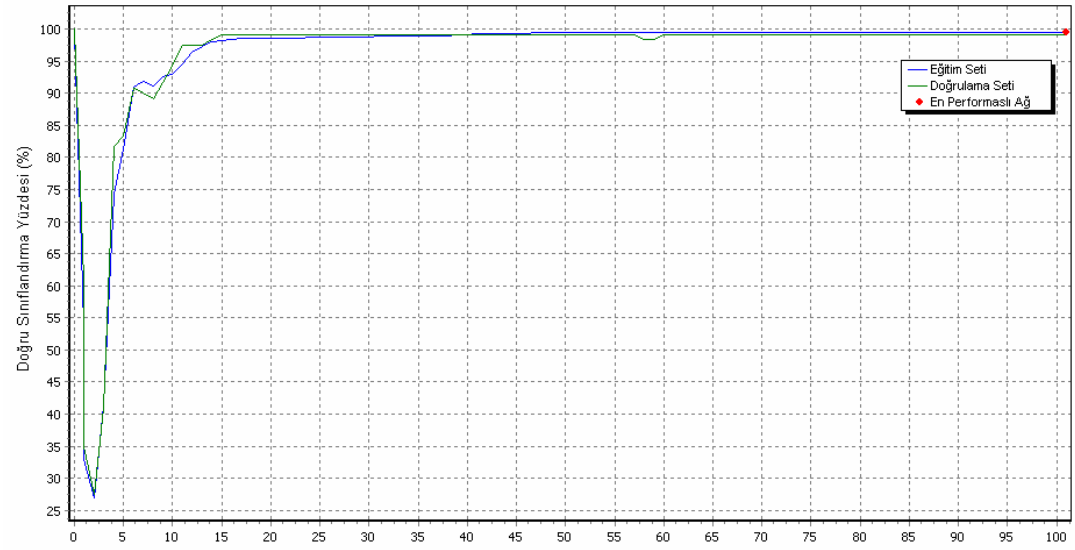
6.8. Azaltılmış Özellik Vektörlerinin YSA ile Parmakizi Tanımada Uygulama Sonuçları

İyileştirilmiş parmakizi görüntülerinden elde edilen parmakizi özellik vektörleri oldukça fazla boyuta sahiptir. Özellik vektörlerinin çok büyük boyutlu olmasından dolayı YSA'da eğitim aşamasında harcanan süre de fazla olmakta ve dolayısı ile YSA yapısı da karmaşık hale gelmektedir. YSA'nın giriş katman nöron sayısını etkileyen özellik vektörü boyutu gizli katmandaki nöron sayısını da dolaylı olarak etkilemekte ve çok fazla nöron kullanılmasını gerektirmektedir. Tüm bu olumsuzluklardan dolayı bu tez çalışmasında parmakizlerine ait özellik vektörlerinin boyutu GA kullanılarak azaltılmıştır.

Boyutu azaltılmış, başka bir ifadeyle seçilmiş özellik vektörlerinin YSA'da eğitilerek parmakizi tanımadaki performansları araştırılmıştır. Çıkış olarak yine parmakizi görüntülerinin kime ait olduğu bilgisi verilmiştir. Tez çalışmasında, eğitim aşamasında literatürde kullanılan NIST-4 veritabanından 10 kişiye ait 100'er adet parmakizi görüntülerinden elde edilen toplam 1000 adet parmakizi görüntüsü kullanılmıştır. Bu görüntülerden yine 700'ü eğitim veri kümesi, 150'si doğrulama veri kümesi ve 150'si test veri kümesi olarak belirlenmiştir.

5. bölümde parmakizi tanımada en iyi sonucun alındığı YSA yapısı, 4 bant ve 12 sektörden oluşan parmakizi özellik vektörleri için 384:58:10 olarak belirlenmiştir.

Bu bölümde, iki kişiye ait parmakizi özellik vektörleri YSA eğitimine katılmayarak test aşamasında uygulanmıştır. YSA eğitimi öncelikle sekiz kişiye ait parmakizlerinin seçilmiş özellik vektörleri için gerçekleştirilmiştir. Burada sekiz kişiye ait toplam 800 adet parmakizi görüntüsünden 560'ı eğitim veri kümesini, 120'si doğrulama veri kümesini ve 120'si test veri kümesini oluşturmuştur. Parmakizi özellik vektörlerinin seçilmesi için GA uygulanması sonucunda daha önce 384 olarak belirlenen giriş vektörü sayısı 174 olarak elde edilmiştir. Böylece özellik vektörlerinde yaklaşık %54 oranında bir azalma gerçekleştirilmiştir. Parmakizi özellik vektörleri [-1 +1] aralığına normalize edildikten sonra tecrübeye dayalı (Heuristic) arama yöntemi (Ghada ve John 2005) kullanılarak gizli katman düğüm sayısı belirlenmiştir. Elde edilen 174 giriş vektörü sayısına göre belirlenen en uygun gizli katman düğüm sayısı 23 olarak elde edilmiştir. Böylece en uygun YSA yapısı 174:23:8 olarak belirlenmiştir. Buna göre sekiz kişiden elde edilen parmakizi özellik vektörlerine göre sistem yeniden eğitilmiştir. Burada 174:23:8 yapısındaki YSA toplam 28 saniyede eğitimi tamamlarken, eğitim aşamasında 100 iterasyon gerçekleştirilmesi sonucu eğitim veri kümesi için %100 ve doğrulama veri kümesi için ise %99,1 oranında doğru sınıflandırma oranları elde edilmiştir. Tüm veri kümesi için ise bu oran %99,6 olarak elde edilmiştir. Şekil 6.9'da sekiz çıkışlı YSA'nın eğitimi aşamasında eğitim ve doğrulama veri kümelerine ait doğru sınıflandırma oranlarına ait grafik gösterilmiştir. YSA'da seçilmiş parmakizi özellik vektörlerini eğitme sonucunda eğitim veri kümesi, test veri kümesi ve tüm veri kümelerine ait elde edilen eşleştirme matrisleri sırasıyla Şekil 6.10 (a), (b) ve (c)'de gösterilmiştir.



Şekil 6.9 Özellik vektörleri azaltılmış sekiz çıkışlı YSA eğitim ve doğrulama veri kümeleri için doğru sınıflandırma oranları

Ortalama doğruluk oranları (%)

a) Eğitim veri kümesi: %100

output:	S1	S2	S3	S4	S5	S6	S7	S8
S1	70	0	0	0	0	0	0	0
S2	0	67	0	0	0	0	0	0
S3	0	0	71	0	0	0	0	0
S4	0	0	0	72	0	0	0	0
S5	0	0	0	0	72	0	0	0
S6	0	0	0	0	0	71	0	0
S7	0	0	0	0	0	0	71	0
S8	0	0	0	0	0	0	0	66

b) Test veri kümesi: % 98,3

output:	S1	S2	S3	S4	S5	S6	S7	S8
S1	15	0	0	0	0	0	0	0
S2	0	17	0	0	0	0	0	0
S3	0	0	14	0	0	0	0	0
S4	0	0	0	15	0	0	0	0
S5	0	0	0	0	14	0	0	0
S6	0	0	0	0	1	12	0	0
S7	0	0	0	0	0	1	14	0
S8	0	0	0	0	0	0	0	17

c) Tüm veri kümesi: % 99,6

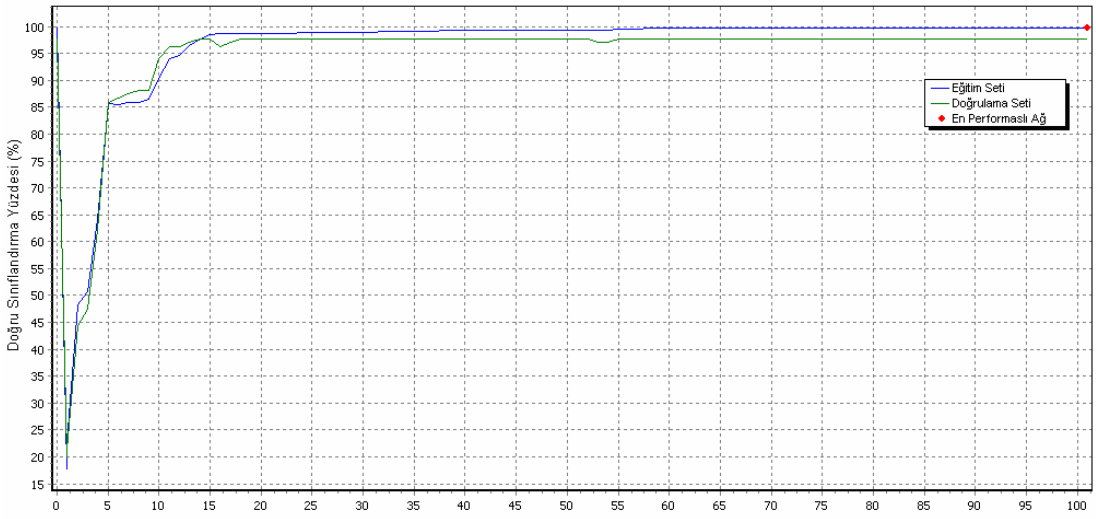
output:	S1	S2	S3	S4	S5	S6	S7	S8
S1	99	0	0	0	1	0	0	0
S2	0	100	0	0	0	0	0	0
S3	0	0	100	0	0	0	0	0
S4	0	0	0	100	0	0	0	0
S5	0	0	0	0	100	0	0	0
S6	0	0	0	0	1	99	0	0
S7	0	0	0	0	0	1	99	0
S8	0	0	0	0	0	0	0	100

Şekil 6.10 Özellik vektörleri azaltılmış sekiz çıkışlı YSA için a) eğitim, b) test ve c) tüm veri kümelerine ait ortalama doğruluk oranları ve eşleştirme matrisleri

Daha sonra diğer iki kişiye ait parmakizleri YSA'da test edilmiştir. Bu parmakizlerinin test sonucu elde edilen doğruluk oranlarının, eşik değeri olan 0,7'nin altında bir oranla sisteme kayıtlı parmakizleri olarak hatalı bir şekilde eşleştirildiği gözlemlenmiştir. Dolayısı ile sistem bu iki parmakizi için hatalı kabul yapmıştır.

Bir sonraki aşamada dokuzuncu kişiye ait parmakizlerinin seçilmiş özellik vektörleri YSA'ya "S9" çıkışı olarak eklenmiştir. Bu dokuz kişiye ait toplam 900 adet parmakizi görüntüsünden 630'u eğitim veri kümesini, 135'i doğrulama veri kümesini ve 135'i de test veri kümesini oluşturur. Özellik vektörlerinin seçilmesi için GA uygulanması sonucunda daha önce sekiz kişiye ait parmakizleri için 174 olarak belirlenen giriş vektörü sayısı 176 olarak elde edilmiştir. Giriş vektör sayısının bir önceki aşamadan farklı elde edilmesinin sebebi YSA eğitimi için en uygun özellik vektörlerinin belirlenmesidir. Her bir kişiye ait parmakizi bilgilerinin daha önce elde edilen seçilmiş parmakizi özellik vektörlerine göre YSA'ya eklenmesinde, YSA eğitiminin başarısız olduğu gözlemlenmiştir. Dolayısı ile her yeni kişiye ait parmakizi bilgilerinin sisteme ilavesinde yeniden GA ile özellik seçimi yapılması gerekmektedir. Bu her yeni kişi ilavesinde YSA'nın çıkış sayısını da etkilemesinden dolayı gizli katman düğüm sayısı için tecrübeye dayalı arama metodunun da yeniden uygulanmasına benzer bir durumdur.

Dokuz kişiye ait parmakizi özellik vektörleri normalize edildikten sonra gizli katman düğüm sayısı belirlenmiştir. 176 giriş vektörü sayısına göre belirlenen en uygun gizli katman düğüm sayısı 27 olarak elde edilmiştir. Böylece en uygun YSA yapısı 176:27:9 olarak belirlenmiştir ve sistem yeniden eğitilmiştir. Burada 176:27:9 yapısındaki YSA toplam 42 saniyede eğitimi tamamlanırken, eğitim aşamasında 100 iterasyon gerçekleştirilmesi sonucu eğitim veri kümesi için %100 ve doğrulama veri kümesi için ise %99,2 oranında doğru sınıflandırma oranları elde edilmiştir. Tüm veri kümesi için ise bu oran %99,6 olarak elde edilmiştir. Şekil 6.11'de dokuz çıkışlı YSA'nın eğitimi aşamasında eğitim ve doğrulama veri kümelerine ait doğru sınıflandırma oranlarına ait grafik gösterilmiştir. YSA'da seçilmiş parmakizi özellik vektörlerini eğitme sonucunda eğitim veri kümesi, test veri kümesi ve tüm veri kümelerine ait elde edilen eşleştirme matrisleri sırasıyla Şekil 6.12 (a), (b) ve (c)'de gösterilmiştir.



Şekil 6.11 Özellik vektörleri azaltılmış dokuz çıkışlı YSA eğitim ve doğrulama veri kümeleri için doğru sınıflandırma oranları

Ortalama doğruluk oranları (%)

a) Eğitim veri kümesi: %100

output:	S1	S2	S3	S4	S5	S6	S7	S8	S9
S1	70	0	0	0	0	0	0	0	0
S2	0	67	0	0	0	0	0	0	0
S3	0	0	71	0	0	0	0	0	0
S4	0	0	0	72	0	0	0	0	0
S5	0	0	0	0	72	0	0	0	0
S6	0	0	0	0	0	71	0	0	0
S7	0	0	0	0	0	0	71	0	0
S8	0	0	0	0	0	0	0	66	0
S9	0	0	0	0	0	0	0	0	70

b) Test veri kümesi: % 98,5

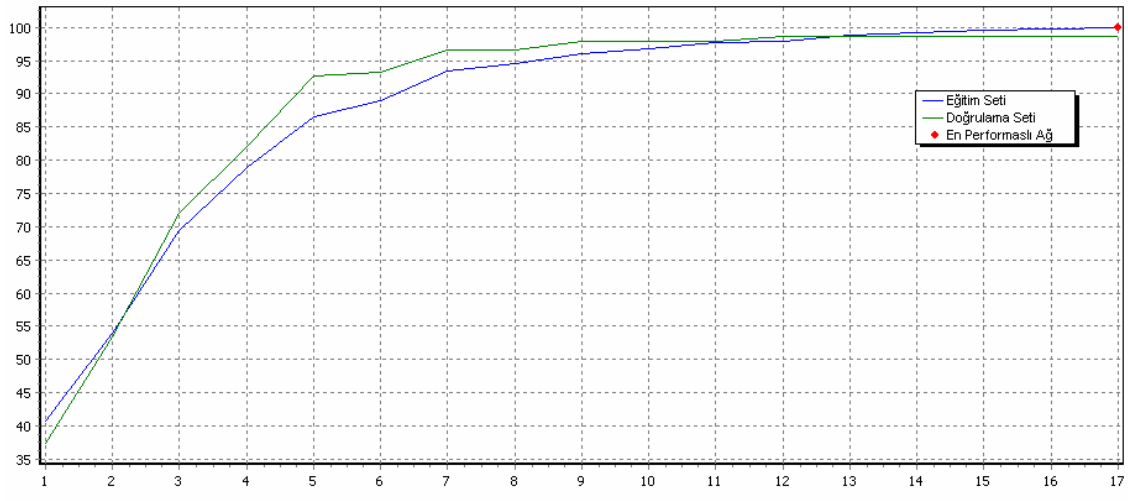
output:	S1	S2	S3	S4	S5	S6	S7	S8	S9
S1	15	0	0	0	0	0	0	0	0
S2	0	17	0	0	0	0	0	0	0
S3	0	0	14	0	0	0	0	0	0
S4	0	0	0	15	0	0	0	0	0
S5	0	0	0	0	14	0	0	0	0
S6	0	0	0	0	1	12	0	0	0
S7	0	0	0	0	0	1	14	0	0
S8	0	0	0	0	0	0	0	17	0
S9	0	0	0	0	0	0	0	0	15

c) Tüm veri kümesi: % 99,6

output:	S1	S2	S3	S4	S5	S6	S7	S8	S9
S1	99	0	0	0	1	0	0	0	0
S2	0	100	0	0	0	0	0	0	0
S3	0	0	100	0	0	0	0	0	0
S4	0	0	0	100	0	0	0	0	0
S5	0	0	0	0	100	0	0	0	0
S6	0	0	0	0	1	99	0	0	0
S7	0	0	0	0	0	1	99	0	0
S8	0	0	0	0	0	0	0	100	0
S9	0	0	0	0	0	0	0	0	100

Şekil 6.12 Özellik vektörleri azaltılmış dokuz çıkışlı YSA için a) eğitim, b) test ve c) tüm veri kümelerine ait ortalama doğruluk oranları ve eşleştirme matrisleri

Son olarak YSA kümesine onuncu kişinin parmakizi özellik vektörleri "S10" çıkışı olarak eklenmiştir. Böylece on kişiye ait toplam 1000 adet parmakizi görüntüsünden 700'ü eğitim veri kümesini, 150'si doğrulama veri kümesini ve 150'si de test veri kümesini oluşturur. Parmakizi özellik vektörlerinin seçilmesi için yeniden GA uygulanması sonucunda daha önce 384 olarak belirlenen giriş vektörü sayısı 182 olarak elde edilmiştir. Parmakizi özellik vektörleri normalize edildikten sonra gizli katman düğüm sayısı belirlenmiştir. 182 giriş vektörü sayısına göre belirlenen en uygun gizli katman düğüm sayısı 32 olarak elde edilmiştir. Böylece en uygun YSA yapısı 182:32:10 olarak belirlenmiştir. Burada 182:32:10 yapısındaki YSA, 00:00:51 sürede eğitimi tamamlarken, eğitim aşamasında 100 iterasyon gerçekleştirilmesi sonucu eğitim veri kümesi için %100 ve doğrulama veri kümesi için %99,2 ve tüm veri kümesi için ise %99,6 doğru sınıflandırma oranları elde edilmiştir. Şekil 6.13'te on çıkışlı YSA'nın eğitimi aşamasında eğitim ve doğrulama veri kümelerine ait doğru sınıflandırma oranlarına ait grafik gösterilmiştir. YSA'da seçilmiş parmakizi özellik vektörlerini eğitme sonucunda eğitim veri kümesi, test veri kümesi ve tüm veri kümelerine ait elde edilen eşleştirme matrisleri sırasıyla Şekil 6.14 (a), (b) ve (c)'de gösterilmiştir.



Şekil 6.13 Özellik vektörleri azaltılmış on çıkışlı YSA eğitim ve doğrulama veri kümeleri için doğru sınıflandırma oranları

Ortalama doğruluk oranları (%)

a) Eğitim veri kümesi: %100

output:	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10
S1	70	0	0	0	0	0	0	0	0	0
S2	0	67	0	0	0	0	0	0	0	0
S3	0	0	71	0	0	0	0	0	0	0
S4	0	0	0	72	0	0	0	0	0	0
S5	0	0	0	0	72	0	0	0	0	0
S6	0	0	0	0	0	71	0	0	0	0
S7	0	0	0	0	0	0	71	0	0	0
S8	0	0	0	0	0	0	0	66	0	0
S9	0	0	0	0	0	0	0	0	70	0
S10	0	0	0	0	0	0	0	0	0	70

b) Test veri kümesi: % 98,5

output:	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10
S1	15	0	0	0	0	0	0	0	0	0
S2	0	17	0	0	0	0	0	0	0	0
S3	0	0	14	0	0	0	0	0	0	0
S4	0	0	0	15	0	0	0	0	0	0
S5	0	0	0	0	14	0	0	0	0	0
S6	0	0	0	0	1	12	0	0	0	1
S7	0	0	0	0	0	1	14	0	0	0
S8	0	0	0	0	0	0	0	17	0	0
S9	0	0	0	0	0	0	0	0	16	0
S10	0	0	0	0	0	0	0	0	0	13

c) Tüm veri kümesi: % 99,6

output:	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10
S1	99	0	0	0	1	0	0	0	0	0
S2	0	100	0	0	0	0	0	0	0	0
S3	0	0	100	0	0	0	0	0	0	0
S4	0	0	0	100	0	0	0	0	0	0
S5	0	0	0	0	100	0	0	0	0	0
S6	0	0	0	0	1	98	0	0	0	1
S7	0	0	0	0	0	1	99	0	0	0
S8	0	0	0	0	0	0	0	100	0	0
S9	0	0	0	0	0	0	0	0	100	0
S10	0	0	0	0	1	0	0	0	0	99

Şekil 6.14 Özellik vektörleri azaltılmış on çıkışlı YSA için a) eğitim, b) test ve c) tüm veri kümelerine ait ortalama doğruluk oranları ve eşleştirme matrisleri

Ayrıca on kişiye ait parmakizlerinden farklı bant ve sektör sayısına göre elde edilen özellik vektörlerine GA uygulanarak özellik seçimi gerçekleştirilmiştir. Parmakizleri için özellik vektörü seçiminden sonra YSA'da eğitim aşaması gerçekleştirilmiştir. Tecrübeye dayalı arama yöntemi kullanılarak gizli katman düğüm sayısı belirlenmiştir. Bant ve sektör sayılarına göre belirlenen en uygun YSA yapılarından bazıları Tablo 6.6'da gösterilmiş olup elde edilen sonuçlar Ekler kısmında yer almaktadır.

Çalışmada kullanılan görüntü iyileştirme yöntemlerinden olan medyan filtre, dalgacık dönüşümü ve çevritsel dönüşüm uygulanarak iyileştirilmiş parmakizi

görüntülerinden elde edilen özellik vektörlerinin GA kullanılarak azaltılması sonucu elde edilen seçilmiş özellik vektörlerinin belirli algoritmalar kullanılarak YSA'da eğitilmesi ve test edilerek elde edilen eğitim ve test sonuçları Tablo 6.7'de gösterilmiştir.

Tablo 6.6 Tecrübeye dayalı arama yöntemi kullanılarak elde edilen en uygun YSA yapıları

Bant sayısı	Sektör sayısı	En uygun YSA yapısı
3	16	172:30:10
3	20	203:36:10
4	12	182:32:10
4	16	219:40:10
6	16	359:60:10
6	20	441:72:10
6	24	542:98:10
7	16	415:68:10
7	20	503:84:10
8	24	712:118:10

Tablo 6.7 Görüntü iyileştirme metotlarına göre YSA yöntemleri eğitim ve test sonuçları

Görüntü İyileştirme Yöntemi	Bant ve sektör sayısı	Eğitim süresi (ss:dd:ss)	YSA yöntemleri	GA'dan sonra YSA yapısı	Eğitim veri kümesi %	Doğrulama veri kümesi %	Test veri kümesi %	Tüm veri kümesi %
Çevritsel	3 x 16	00:00:07	QP	172:30:10	100	95,33	96,00	98,70
Normal	3 x 16	00:00:13	QP	159:30:10	99,00	96,00	98,00	98,40
Dalgacık	3 x 16	00:00:06	QP	155:30:10	99,86	96,00	96,67	98,80
Çevritsel	3 x 20	00:00:21	QP	203:36:10	100	95,33	96,00	98,70
Normal	3 x 20	00:00:23	QP	201:36:10	99,86	94,67	92,67	98,00
Dalgacık	3 x 20	00:00:27	QP	218:36:10	100	95,33	96,67	98,80

Çevritsel	**4 x 12**	**00:00:51**	**CGD**	**182:32:10**	**100**	**99,23**	**98,50**	**99,60**
Çevritsel	4 x 16	00:02:07	CGD	219:40:10	100	98,67	96,00	99,20
Çevritsel	4 x 16	00:00:31	QP	219:40:10	99,71	98,67	95,33	98,90
Normal	4 x 16	00:01:27	BBP	238:40:10	99,71	96,00	98,00	98,90
Normal	4 x 16	00:01:33	CGD	238:40:10	100	95,33	98,00	99,00
Dalgacık	4 x 16	00:00:22	QP	241:40:10	100	96,00	97,33	99,00
Çevritsel	6 x 16	00:02:11	BBP	359:60:10	100	95,33	97,33	98,90
Çevritsel	6 x 16	00:02:27	CGD	359:60:10	100	94,00	98,67	98,90
Normal	6 x 16	00:02:28	BBP	354:60:10	100	98,00	94,00	98,80
Normal	6 x 16	00:03:12	CGD	354:60:10	100	98,00	93,33	98,70
Çevritsel	6 x 24	00:13:18	CGD	542:98:10	100	96,67	96,67	99,00
Normal	6 x 24	00:14:42	CGD	551:98:10	99,29	97,33	92,67	98,00
Dalgacık	6 x 24	00:14:17	CGD	517:98:10	99,86	98,00	96,00	99,00
Çevritsel	7 x 16	00:01:35	CGD	415:68:10	99,71	99,33	98,00	99,40
Normal	7 x 16	00:03:21	CGD	434:68:10	95,88	99,11	98,21	96,50
Dalgacık	7 x 16	00:02:51	CGD	418:68:10	99,43	95,33	96,00	98,30
Çevritsel	7 x 20	00:02:03	BBP	503:84:10	100	96,67	96,67	99,00
Çevritsel	7 x 20	00:05:05	IBP	503:84:10	100	97,33	96,67	99,10
Normal	7 x 20	00:05:16	BBP	530:84:10	100	96,67	96,67	99,00
Normal	7 x 20	00:10:55	IBP	530:84:10	100	96,00	95,33	98,70
Dalgacık	7 x 20	00:05:10	BBP	519:84:10	100	98,00	96,00	99,10
Dalgacık	7 x 20	00:06:56	IBP	519:84:10	100	97,33	94,67	98,80
Çevritsel	7 x 24	00:02:37	QP	611:112:10	100	97,33	97,33	99,20
Normal	7 x 24	00:02:38	QP	617:112:10	100	98,67	94,67	99,00
Dalgacık	7 x 24	00:02:52	QP	626:112:10	100	97,33	96,67	99,10
Çevritsel	**8 x 12**	**00:01:11**	**IBP**	**359:58:10**	**100**	**98,00**	**98,67**	**99,50**
Dalgacık	8 x 12	00:01:06	IBP	360:58:10	100	95,33	98,67	99,10
Çevritsel	8 x 16	00:05:21	BBP	476:78:10	100	98,00	98,67	99,50
Çevritsel	8 x 16	00:03:14	IBP	476:78:10	100	96,67	97,33	99,10
Normal	8 x 16	00:05:57	BBP	477:78:10	97,57	95,33	92,67	96,50
Çevritsel	**8 x 24**	**00:11:00**	**BBP**	**712:118:10**	**100**	**98,67**	**98,00**	**99,50**
Çevritsel	8 x 24	00:18:04	IBP	712:118:10	100	97,33	95,33	98,90
Çevritsel	8 x 24	00:01:51	QP	712:118:10	99,57	98,67	96,67	99,00
Normal	8 x 24	00:12:28	BBP	709:118:10	99,00	98,67	96,67	99,30
Normal	8 x 24	00:08:40	IBP	709:118:10	100	96,67	96,67	99,00
Normal	8 x 24	00:02:05	QP	709:118:10	99,86	98,67	97,33	99,30
Dalgacık	8 x 24	00:15:35	BBP	738:118:10	100	98,00	95,33	99,00
Dalgacık	8 x 24	00:15:10	IBP	738:118:10	100	97,33	95,33	98,90
Dalgacık	8 x 24	00:01:10	QP	738:118:10	100	98,67	95,33	99,10

Elde edilen sonuçlara göre en iyi performansı özellik vektörleri seçilmeden yapılan tanıma işleminde olduğu gibi çevritsel dönüşüm kullanılarak iyileştirilmiş parmakizi görüntüsü sağlamıştır. Burada en iyi performans gösteren bant ve sektör sayıları 4 bant ve 12 sektör, 8 bant ve 12 sektör ve 8 bant ve 24 sektör olmuştur. Bu bant ve sektör sayılarına göre sektörizasyonlar yapıldıktan sonra elde edilen özellik vektörlerinin GA kullanılarak seçilmesi sonucu bir parmakizine ait sırasıyla 182, 359 ve 712 özellik vektörü elde edilmiştir. Bu özellik vektörlerinden 182 düğümlü giriş katmanına, 32 düğümlü bir gizli katmana ve 10 düğümlü bir çıkış katmanına sahip bir YSA yapısı belirlenmiştir. Bu YSA yapısına uygulanan eşleştirmeli eğim azaltımlı öğrenme algoritması (CGD) ile en iyi parmakizi tanıma performansı elde edilmiştir. Aynı performansı yukarıda adı geçen özellik vektörleri de vermiştir. Burada YSA eğitim aşamasında 100 iterasyon gerçekleştirilmesi sonucu eğitim veri kümesi için %100 başarı elde edilmiştir. Test veri kümesi dikkate alındığında başarımın %98,5 ve tüm veri kümesi için ise %99,6 olduğu belirlenmiştir. Bu çalışmada geliştirilen yöntemlerin geleneksel yöntemler kadar iyi sonuç verdiği gözlemlenmiş ve Sonuçlar kısmında karşılaştırılmıştır.

6.9. Sonuç

Kişilerden elde edilen parmakizi sayısı kadar bunlara ait parmakizi özellik vektörlerinin boyutu da parmakizi tanıma için önemlidir. İyileştirilmiş parmakizi görüntülerinden elde edilen parmakizi özellik vektörleri oldukça fazla boyuta sahiptir. Özellik vektörlerinin çok büyük boyutlu olmasından dolayı YSA'da eğitim aşamasında harcanan süre de fazla olmakta ve dolayısı ile YSA yapısı da karmaşık hale gelmektedir. YSA'nın giriş katman nöron sayısını etkileyen özellik vektörü boyutu gizli katmandaki nöron sayısını da dolaylı olarak etkilemekte ve çok fazla nöron kullanılmasını gerektirmektedir. Tüm bu olumsuzluklardan dolayı bu tez çalışmasında parmakizlerine ait özellik vektörlerinin boyutu GA kullanılarak azaltılmıştır. Boyutu azaltılmış, başka bir ifadeyle seçilmiş özellik vektörlerinin

YSA'da eğitilerek parmakizi tanımadaki performansları araştırılmıştır. Çıkış olarak yine parmakizi görüntülerinin kime ait olduğu bilgisi verilmiştir.

Daha önce parmakizi tanıma için en iyi sonucun alındığı YSA yapısı, 4 bant ve 12 sektörden oluşan parmakizi özellik vektörleri için 384:58:10 olarak belirlenmiştir. YSA eğitimi öncelikle sekiz kişiye ait parmakizlerinin seçilmiş özellik vektörleri için gerçekleştirilmiştir. Parmakizi özellik vektörlerinin seçilmesi için GA uygulanması sonucunda daha önce 384 olarak belirlenen giriş vektörü sayısı 174 olarak elde edilmiştir. Böylece özellik vektörlerinde yaklaşık %54 oranında bir azalma gerçekleştirilmiştir. Parmakizi özellik vektörleri [-1 +1] aralığına normalize edildikten sonra gizli katman düğüm sayısı belirlenmiştir. Elde edilen 174 giriş vektörü sayısına göre belirlenen en uygun gizli katman düğüm sayısı 23 olarak elde edilerek en uygun YSA yapısı da 174:23:8 olarak belirlenmiştir. Buna göre sekiz kişiden elde edilen parmakizi özellik vektörlerine göre sistem yeniden eğitilmiştir. Burada 174:23:8 yapısındaki YSA toplam 28 saniyede eğitimi tamamlarken, eğitim aşamasında 100 iterasyon gerçekleştirilmesi sonucu eğitim veri kümesi için %100 ve doğrulama veri kümesi için ise %99,1 oranında doğru sınıflandırma oranları elde edilmiştir. Tüm veri kümesi için ise bu oran %99,6 olarak belirlenmiştir.

Bir sonraki aşamada YSA kümesine dokuzuncu kişiye ait parmakizlerinin seçilmiş özellik vektörleri "S9" çıkışı olarak eklenmiştir. Parmakizi özellik vektörlerinin seçilmesi için yeniden GA uygulanması sonucunda daha önce sekiz kişiye ait parmakizleri için 174 olarak belirlenen giriş vektörü sayısı 176 olarak elde edilmiştir. Giriş vektör sayısının bir önceki aşamadaki sayıdan farklı elde edilmesinin sebebi YSA eğitimi için en uygun özellik vektörlerinin belirlenmesidir. Her bir kişiye ait parmakizi bilgilerinin YSA kümesine eklenmesinde, daha önce elde edilen seçilmiş parmakizi özellik vektörlerinin bu parmakizi özellik vektörleri için de kullanılması sonucunda YSA eğitiminin başarısız olduğu gözlemlenmiştir. Dolayısı ile her yeni kişiye ait parmakizi bilgilerinin sisteme ilavesinde yeniden GA ile özellik seçimi yapılması gerekmektedir. Burada 176:27:9 yapısındaki YSA için 00:00:42 sürede eğitim tamamlanırken, eğitim aşamasında 100 iterasyon gerçekleştirilmesi sonucu eğitim veri kümesi için %100 doğru sınıflandırma oranı elde edilmiştir. Tüm veri kümesi için ise bu oran yaklaşık %99,6 olarak elde edilmiştir.

YSA kümesine onuncu kişinin parmakizi özellik vektörleri “S10” çıkışı olarak eklenmiştir. Burada 182:32:10 yapısındaki YSA, 51 saniyede eğitilerek, eğitim aşamasında 100 iterasyon gerçekleştirilmesi sonucu eğitim veri kümesi için %100 ve tüm veri kümesi için ise %99,6 doğru sınıflandırma oranları elde edilmiştir.

Ayrıca on kişiye ait parmakizlerinden farklı bant ve sektör sayısına göre elde edilen özellik vektörlerine GA uygulanarak özellik seçimi gerçekleştirilerek adaptif bir sistem gerçekleştirilmiştir. Parmakizleri için özellik vektörü seçiminden sonra YSA’da eğitim aşaması gerçekleştirilmiştir. Tecrübeye dayalı arama yöntemi kullanılarak gizli katman düğüm sayısı belirlenmiştir.

Çalışmada, görüntü iyileştirme yöntemleri uygulanarak iyileştirilmiş parmakizi görüntülerinden elde edilen özellik vektörleri GA kullanılarak azaltılmıştır. Seçilmiş özellik vektörlerinin belirli algoritmalar kullanılarak YSA’da eğitilmesi ve test edilerek elde edilen eğitim ve test sonuçlarına göre en iyi performansı, özellik vektörleri seçilmeden yapılan tanıma işleminde olduğu gibi çevritsel dönüşüm ile iyileştirilmiş görüntüler sağlamıştır. Burada en iyi performans gösteren bant ve sektör sayıları 4 bant ve 12 sektör, 8 bant ve 12 sektör ve 8 bant ve 24 sektör olmuştur. Bu bant ve sektör sayılarına göre sektörizasyonlar yapıldıktan sonra elde edilen özellik vektörlerinin GA kullanılarak seçilmesi sonucu bir parmakizine ait sırasıyla 182, 359 ve 712 özellik vektörü elde edilmiştir. Elde edilen bu özellik vektörleri için 182 düğümlü giriş katmanına, 32 düğümlü bir gizli katmana ve 10 düğümlü bir çıkış katmanına sahip YSA yapısına eşleştirmeli eğim azaltımlı öğrenme algoritması (CGD) ile iyi bir parmakizi tanıma elde edilmiştir. Aynı performans yukarıda adı geçen özellik vektörleri tarafından da elde edilmiştir. Burada YSA eğitim aşamasında 100 iterasyon gerçekleştirilmesi sonucu eğitim veri kümesi için %100 başarı elde edilmiştir. Test veri kümesi dikkate alındığında başarımın %98,5 ve tüm veri kümesi için ise %99,6 olduğu belirlenmiştir.

Sonuçların geleneksel yöntemler kadar iyi olduğu gözlemlenmiş olup elde edilen sonuçlarla geleneksel yöntemlerle elde edilmiş sonuçlar, bir sonraki bölümde karşılaştırılmıştır.

7. SONUÇLAR

Bu çalışma parmakizi tabanlı biyometrik tanıma sistemlerinde önemli bir yer tutan parmakizi özellik çıkarımı, GA ile özellik vektörü boyutunun azaltılması ve YSA kullanımının incelenmesi ve gerçekleştirilmesi problemleri üzerine yapılmıştır. Parmakizi şablonunun güvenliği, sahte parmakizlerinin belirlenmesi, şifreleme ve güvenlik şirketlerini içeren parmakizi tanıma sistemleri üzerine yapılmış çok önemli çalışmalar vardır. Ancak bunlar, güvenilir bir parmakizi tanıma sistemi geliştirmede sistematik bir yol haritası çizerler. Bu tez çalışmasında ise amaç ve yöntem farklıdır. Parmakizi özellik çıkarımı, sınıflandırma ve tanıma gibi parmakizi tanıma teknolojisi, ilginç problemler ve bunlarla uğraşan birkaç ticari firma olmasına rağmen oldukça önemlidir. Bu teknolojilerde ulaşılması gereken nokta tanıma doğruluk oranının geliştirilmesidir. Bu tez çalışmasında Alyuda firmasına ait Neurointelligence ve MATLAB programı ile gerçekleştirilerek parmakizi tanıma performansını artırıcı sonuçlar elde edilmiştir. Bu sonuçlardan önemli olanları aşağıda sıralanmıştır:

1. Parmakizi doğrulama sistemlerinde temel sorunlardan birisi, parmakizlerinde ayırt edilebilir bilginin ne olduğudur. Bu tez çalışmasında parmakizlerinin ayrıntı tabanlı simgelenişinde bilgi içeriğine tesir eden genetik ve çevresel faktörler nicel olarak analiz edilmiştir. Bu analiz ile ayrıntı tabanlı simgelenişte mevcut sınırlı bilgi miktarından dolayı otomatik ayrıntı tabanlı parmakizi tanımanın sınırlı bir performans göstereceği sonucuna varılır. Otomatik parmakizi tanıma sistemleri tasarlanırken ayrıntı tabanlı parmakizi simgeleniş haricinde bir yapı üzerine araştırmalar yapılmalıdır. Bundan dolayı parmakizleri için yeni bir filtre kümesi tabanlı simgeleniş geliştirilmiştir. Bu simgeleniş, az yer kaplar ve parmakizleri arasında fark edilecek derecede güçlüdür. Bu simgeleniş kullanılarak literatürde ifade edilen en iyi doğruluklu parmakizi sınıflandırma ve doğrulama oranı başarılmıştır. Bu yaklaşımın temel avantajı, hesaplanabilir olarak cazip eşleştirme kapasitesine sahip olmasıdır. Mesela, tüm kayıtlı parmakizlerinin dönüşüm ve yönelim olarak normalize edilmiş “FingerCode” özellik vektörleri şablon olarak kayıt edilirse, YSA

ile tanımada etkin olarak bitsel tabanlı eşleştirme yapılır. Sonuç olarak, tanıma süresi diğerlerine göre veritabanı boyutuna duyarsız olacaktır. Ayrıca parmakizi özellik çıkarımı ve tanıma için bu tez çalışmasındaki yaklaşım, donanım uygulamasından çok string tabanlı bir ayrıntı eşleştiricidir.

2. Görüntü iyileştirme yöntemlerinin YSA ile parmakizi tanımaya etkisi araştırılmıştır. Geleneksel yöntem olan lineer, medyan ve uyarlamalı filtrelerle birlikte dalgacık dönüşümü ve dalgacık dönüşümün geliştirilmişi olan çevritsel dönüşüm kullanılarak parmakizi görüntüleri gürültüden arındırılmaya çalışılarak iyileştirilmiştir. İyileştirilen bu görüntülerin en büyük sinyal-gürültü oranına göre performans analizi yapılmıştır. En iyi parmakizi iyileştirme performansını, çevritsel dönüşümün verdiği gözlemlenmiştir.

3. İyileştirilmiş parmakizi görüntülerinden FingerCode adı verilen özellik vektörleri geliştirilen bir yöntemle elde edilmiştir. Elde edilen bu özellik vektörleri YSA ile eğitilerek parmakizi tanıma gerçekleştirilmiştir. Burada 10 kişiye ait toplam 1000 parmakizi görüntüsünden 700'ü YSA ile eğitim aşamasında, 150 tanesi YSA'nın eğitimi sırasında YSA yapısını doğrulamada ve 150'si eğitilen YSA yapısını test etmede kullanılmıştır. Bir YSA algoritması olan "hatanın geriye yayılması" metotları ile eğitme sonucunda en iyi performans %100 eğitim doğruluğu ve %99,6 genel tanıma oranı, eşleştirmeli eğim azaltım algoritması ile elde edilmiştir.

4. İyileştirilmiş parmakizi görüntülerinden elde edilen parmakizi özellik vektörleri çok büyük boyutludur. Özellik vektörlerinin çok büyük boyutlu olmasına paralel olarak, bunların YSA ile eğitilmesi aşamasında harcanan süre de fazla olmakta ve aynı zamanda YSA yapısı da karmaşık hale gelmektedir. YSA'nın giriş katman nöron sayısını etkileyen özellik vektörü boyutu gizli katmandaki nöron sayısını da dolaylı olarak etkilemektedir. Tüm bu olumsuzluklardan dolayı parmakizlerine ait özellik vektörlerinin boyutu GA kullanılarak azaltılmıştır. GA uygulanarak azaltılan parmakizi özellik vektörlerinin YSA ile eğitilmesi aşamasında 100 iterasyon gerçekleştirilmesi sonucu eğitim veri kümesi için %100 başarı elde edilmiştir. Test veri kümesi dikkate alındığında başarımın %98,5 ve tüm veri kümesi için ise %99,6 olduğu belirlenmiştir.

5. 4 bant ve 12 sektör kullanılarak elde edilen özellik vektörlerinin ve GA ile azaltılmış özellik vektörlerinin YSA ile eğitimi aşamasında süre, Intel Pentium 4 3.2 GHz (FSB800) saat frekanslı (1024 KBayt önbellekli) işlemci ve 2×512 MBayt DDR 400 (3.2 GB/s) RAM'a sahip bir bilgisayar (I) konfigürasyonu ile Intel Pentium Duo 2.8 (FSB800) GHz saat frekanslı (2048 MBayt önbellekli) işlemci ve 2×512 MBayt Dual DDR2 400 (6.4 GB/s) RAM'a sahip çift çekirdekli bir bilgisayar (II) konfigürasyonu kullanılarak kıyaslanmıştır. Bilgisayar (I) ile bilgisayar (II) konfigürasyonu Tablo 7.1'de gösterilmiştir. Burada normal sayıdaki özellik vektörleri ve GA ile azaltılmış özellik vektörleri ile her iki konfigürasyonda YSA ile parmakizi tanıma gerçekleştirilmiştir. Her iki durum için tanıma ve zaman performansı karşılaştırıldığında, tanıma performansı benzer çıkmasıyla birlikte YSA ile eğitme aşaması bilgisayar (I) konfigürasyonu için yaklaşık 27 dakika sürer iken GA ile azaltılan özellik vektörlerinin YSA ile eğitme aşaması aynı konfigürasyonda 51 saniye sürmüştür. (II) nolu bilgisayar konfigürasyonunda gerçekleştirilen YSA ile eğitme aşaması ise yaklaşık 7 dakika sürer iken, bu konfigürasyonda GA ile azaltılan özellik vektörlerinin YSA ile eğitme aşaması 15 saniye gibi çok kısa bir sürede gerçekleştirilmiştir. GA ile parmakizi özellik vektörlerinin azaltılması aşaması da bilgisayar (I) konfigürasyonu ile yaklaşık 4 dakika ve (II) nolu bilgisayar konfigürasyonu ile yaklaşık 1 dakika gibi bir sürede gerçekleştirilmiştir. Böylece zaman performansı olarak özellik vektörlerinin azaltılmasının işlemci performansıyla orantılı olarak daha iyi sonuç verdiği görülmektedir. İşlemci frekansına göre elde edilen zaman performansı Tablo 7.2'de gösterilmiştir.

Tablo 7.1 YSA eğitim süresinin test edildiği bilgisayar konfigürasyonları

Özellikler	**Bilgisayar (I)**	**Bilgisayar (II)**
İşlemci	Intel Pentium 4	Intel Pentium Duo
İşlemci Saat Frekansı	3.2 GHz	2 × 2.8GHz
Sistem Saat Frekansı	800 MHz	800 MHz
Ön Bellek	1 Mega Bayt	2 Mega Bayt
Bellek	2 × 512 DDR	2 × 512 Dual DDR2
Bellek En İyi Bant Genişliği	3.2 GB/s	6.4 GB/s

Tablo 7.2 İşlemci frekansına göre YSA eğitimi ve GA ile özellik azaltımı süresi

YSA Eğitiminde Kullanılan Özellik Vektörü	YSA Eğitim Süresi (dakika:saniye)		Eğitme kümesi Doğrulama Oranı
	Bilgisayar (I)	Bilgisayar (II)	
Normal	27:02	07:20	%100
GA ile azaltılmış	00:51	00:15	%100
GA ile özellik azaltımı	04:12	00:58	

Böylece çalışmada, parmakizi tanıma probleminin daha hassas ve hızlı çözümü için Genetik Algoritmalar ile birlikte Yapay Sinir Ağlarından oluşan Esnek Hesaplama Yöntemleri'nin kullanımının iyi sonuçlar verdiği gözlemlenmiştir.

8. TARTIŞMA ve ÖNERİLER

Elde edilen sonuçların tartışılabilmesi için literatürdeki diğer uygulama sonuçları ile karşılaştırılmıştır.

Bilindiği üzere parmakizi sınıflandırma ve tanıma sonuçlarını göstermenin bir yolu eşleştirme matrisidir. Eşleştirme matrisi, tahmin edilen her bir sınıf için satırlara ve gerçek her bir sınıf için sütunlara sahiptir (Karu ve Jain 1996). Diyagonal üzerindeki sayılar, doğru olarak sınıflandırılmış parmakizi sayılarını, diyagonal haricindeki sayılar ise hatalı olarak eşleştirilen parmakizi sayılarını belirtmektedir. Karu ve Jain (1996)'da yapılan çalışmada elde edilen sınıflandırma sonuçları Tablo 8.1'te gösterilmiştir. Burada 197 adet kemer tip parmakizinin çadır kemer olarak hatalı sınıflandırıldığı görülmektedir.

Yager ve Amin (2004)'in yapmış olduğu çalışmada da farklı parmakizi veritabanlarının sistem sonuçlarını değerlendirmek üzere kullanıldığı görülmektedir. Parmakizi görüntülerinin kalitesinin sistem performansını açık olarak etkilemekte olduğu ve görüntü kalitesinin parmakizi veritabanları arasında farklılıklar gösterdiği gözlemlenmiştir. Bu çalışmada ABD'de Standartlar ve Teknoloji Ulusal Enstitüsü (NIST) tarafından oluşturulan NIST-4 veritabanı 1000 adet 8-bit gri ölçekli parmakizi görüntüsü kullanılmıştır. Bu veritabanında parmakizleri kişilere ait olarak sınıflandırılmasının yanı sıra parmakizi tipleri olan kemer, çadır kemer, sarmal, sağ kement, sol kement'e göre de etiketlenmiştir. Yapılan çalışmalarda sınıflandırma parmakizi tiplerine, parmakizi tanıma ise parmakizlerinin sahibi olduğu kişiler dikkate alınarak gerçekleştirilmiştir. Tablo 8.2, NIST-4 veritabanı için parmakizi sınıflandırma sistemlerini göstermektedir (Yager ve Amin 2004).

Tablo 8.1 Eşleştirme Matrisi

	Sarmal	**Sol Kement**	**Sağ Kement**	**Kemer**	**Çadır Kemer**
Sarmal	731	35	30	1	10
Sol Kement	33	780	6	10	79
Sağ Kement	23	3	672	7	7
Kemer	5	36	37	912	197
Çadır Kemer	4	11	45	5	321

Tablo 8.2 NIST-4 kullanılarak değerlendirilen sınıflandırma algoritmasının özeti

Araştırmacı, Yıl	**Parmakizi Özellikleri**	**Sınıflandırma**
Wilson ve ark. (1997)	Yönelim alanı	NN
Karu ve Jain (1996)	Ayrıntılar	Tecrübeye dayalı
Senior (1997)	Hat yapısı	HMM
Jain ve ark. (1999)	FingerCode	kNN, NN
Hong ve Jain (1999)	Ayrıntılar ve hat yapısı	Tecrübeye dayalı
Cappelli ve ark. (2000)	Yönelim alanı	Çoklu-mesafeli KL
Cappelli ve ark. (2000)	İlişkisel graf	Hatalı graf eşleştirme
Senior (2001)	Hat yapısı	HMM, Karar ağacı, NN
Zhang ve ark. (2002)	Ayrıntılar ve hat izleme	Tecrübeye dayalı
Jain ve ark. (2001)	Parmakizi referansları	Referans uydurma

Tablo 8.2'de listelenen algoritmalar aynı veritabanı üzerinde test edilse bile, bunlar arasında doğrudan ve açık bir karşılaştırma yapmaya bazı engeller vardır. YSA gibi bazı algoritmalar eğitime ihtiyaç duyarlar ve eğitmede kullanılan görüntüler sistemden sisteme çeşitlilik gösterirler. Bu algoritmalar veritabanının bir kısmını eğitim için, bir kısmını da test için kullanırlar. NIST-4 veritabanı üzerinde

yapılan çalışmalar sonucunda önerilen algoritmaların doğrulama sonuçları Tablo 8.3'te gösterilmiştir.

Tablo 8.3'teki "sınıf" sütunu, sarmal, kement ve kemer gibi parmakizi tiplerine göre sınıflandırma yapmak için kullanılan sınıf sayısını gösterir. Beş sınıf; kemer, çadır kemer, sarmal, sağ kement, sol kement tipini gösterir. Bunlardan kemer ve çadır kemeri ayırt etmek zordur ve nadir olarak bulunurlar. Bu yüzden bunları tek bir sınıf olarak ele alıp toplam dört sınıf olarak ta sınıflandırma yapılabilir. "Doğruluk" sütunu ise sınıflandırma sistemi tarafından doğru bir şekilde etiketlenmiş veritabanı yüzdesini gösterir.

Tablo 8.3 Parmakizi sınıflandırma algoritmalarının doğruluk oranları

Araştırmacı, Yılı	**Sınıf Sayısı**	**Doğruluk Oranı**
Wilson ve ark. (1997)	5	81,0
Karu ve Jain (1996)	5	85,4
Jain ve ark. (1999)	5	90,0
Hong ve Jain (1999)	5	87,5
Cappelli ve ark. (2000)	5	92,2
Cappelli ve ark. (2000)	5	87,1
Zhang ve ark. (2002)	5	84,0
Wilson ve ark. (1997)	4	86,0
Karu ve Jain (1996)	4	91,4
Senior (1997)	4	81,6
Jain ve ark. (1999)	4	94,8
Hong ve Jain (1999)	4	92,3
Senior (2001)	4	88,5
Jain ve ark. (2001)	4	91,3

Parmakizi tanıma aşamasında kullanılan parmakizi sayısı önemlidir. Farklı parmakizi veritabanları üzerinde farklı yöntemler kullanılarak gerçekleştirilen parmakizi tanıma, kimlik saptama ve eşleştirme doğruluk ve/veya hata oranları da Tablo 8.4'te gösterilmiştir. Burada CR, doğrulama oranı; FRR, hatalı ret oranı; FAR, hatalı kabul oranı; GAR, gerçek kişi kabul oranı; EER ise eşit hata oranını simgeler. Literatürde elde edilen bu sonuçlar göz önüne alındığında bazı araştırmacıların çok az sayıda parmakizi kullanarak araştırma yaptığı görülmüştür. Parmakizi sayısı arttıkça araştırmacıların kullandıkları yöntemlerin de hata oranlarının arttığı görülmektedir. Buna karşın bu çalışmada geliştirilen yöntemle, parmakizi sayısı arttıkça parmakizi tanıma oranı çok etkilenmemektedir. Böylece adaptif ve esnek bir yapı elde edilmiştir.

Tablo 8.4 Parmakizi tanıma algoritmalarının doğruluk oranları

Araştırmacı, Yılı	**Tanıma ve Eşleştirme Algoritmaları**	**Doğruluk Oranı**
Wang ve Lee (1999)	LVQ	%99,6 CR
Wei ve ark. (2006)	Graf tabanlı	%1,42 EER
Jiang ve Yau (2000)	Graf tabanlı	%2,82 EER
Tong ve ark. (2006)	Hough dönüşümü	%98,26 CR
Huang ve ark. (2005)	Ayrıntı tabanlı Euclidean mesafesi	%25,4 FRR %0,83 FAR
Kwon ve ark. (2006)	Ayrıntı sınıflandırma, k-DNN	%0,92 EER
Qi ve ark. (2005)	Ayrıntı tabanlı, Euclidean mesafesi	%3,01 EER
Jea ve Govinderaju (2005)	Ayrıntı indeksleme	%1,09 EER
Nanni ve Lumini (2006)	FingerCode, Parzen pencere sınıflandırma	%0,39 EER
Kulkarni ve ark. (2006)	Yönelim alanı	%97 GAR (GAR=100-FRR)
Bazen ve Gerez (2000)	Ayrıntı tabanlı, Genetik programlama	%1 FRR %0,01 FAR
Tan ve Bhanu (2003)	Ayrıntı tabanlı, üçgenler	%83 GAR

Lee ve ark. (2002)	Ayrıntı tabanlı	%99,58 CR
Deriche (2001)	Dalgacık dönüşümlü	%57,56 CR
Bazen ve ark. (2000)	Korelasyon tabanlı	%7,98 EER
Hao ve ark. (2002)	Ayrıntı tabanlı	%98 CR
Sujan, Mulqueen (2001)	Fourier dönüşümü, YSA	%95 CR
Luo ve ark. (2000)	Ayrıntı tabanlı	%99,4 CR
Jiang ve ark. (2004)	Referans nokta tabanlı	%94,79 CR
Lee ve Wang (1999)	Gabor filtre, k-NN	%97,2 CR
Bazen ve Gerez (2003)	Spline modelleme	%2,5 EER
Zorita ve ark. (2001)	Ayrıntı tabanlı	%4,28 EER
Gao ve ark. (2003)	Sinerjetik	%94,1 CR
Guo (2005)	Gizli Markov modeli	%7,1 EER
Zhu ve ark. (2005)	Ayrıntı tabanlı	%1,75 EER
Lee ve ark. (2001)	Gabor temelli fonksiyon	%97,9 CR
Melin ve ark. (2004)	Modüler YSA	%96,4 CR
Lynch ve Gaunt (1995)	Lineer ağırlıklı YSA	%84 CR
Patil ve ark. (2005)	Karesel FingerCode	%93 CR
Tan ve Bhanu (2002)	Ayrıntı tabanlı, Genetik algoritma	%99,1 CR
Tico ve Kuosmanen (2001)	Dalgacık dönüşümü, k-NN	%85,71 CR
Jain ve ark. (2000)	Filtre kümesi tabanlı (FingerCode)	%1,07 FAR %7,87 FRR
Tachaphetpiboon ve Amornraksa (2005)	Ayrık kosinüs dönüşümü, k-NN	%95,6 CR
Tico ve ark. (2001)	Dalgacık ayrıntılar	%79,23 CR
Tachaphetpiboon ve Amornraksa (2006)	FFT Ayrıntılar, k-NN	%98,9 CR
Melin ve ark. (2005)	Fuzzy Sugeno'lu YSA	%95 CR
He ve ark. (2006)	Hat tabanlı en yakın komşu	%2,7 EER
Bu tez çalışması	**ANN, GA**	**%100 ve %99,6**

Tüm bu sonuçlar dikkate alındığında Genetik algoritmalar ile azaltılmış özellik vektörlerinin Yapay Sinir Ağı kullanılarak eğitilmesi ile parmakizi tanıma sistemi mevcut yöntemler kadar, hatta diğer birçok yöntemden elde edilen sonuçlara göre daha iyi bir başarım elde edildiği gözlemlenmiştir.

Çalışmanın ilk aşamasında FingerCode özellik vektörü çıkarımı için referans noktasının belirlenmesi amacıyla kayıt işlemi gerçekleştirilir. Referans nokta tayini algoritması parmakizleri üzerinde doğru bir şekilde çalışsa, hatta bununla düşük kalitedeki parmakizi görüntülerinden referans noktası elde edilse bile çok düşük kalitedeki görüntülerde referans nokta tayini belirlemek başarısız olmaktadır. Çok iyi bir özellik çıkarımı algoritması, tek referans noktasına güvenmeyebilir. Mümkün çözüm olarak, çoklu referans nokta adayları belirlenebilir ve bu referans noktaların tümü çoklu şablonlarda depolanabilir. FingerCode özellik çıkarımı için genelde tek referans noktanın elde edilmesi beklenir. Bu yüzden çoklu referans nokta tayininde merkeze yakın olan nokta esas alınır.

Filtre kümesi simgelenişinin mevcut uygulamaları dönmeden bağımsız değildir. Dönme, FingerCode elde edilirken önemlidir. Parmakizi özellik vektörleri elde edilirken bir başlangıç yeri ve yönünden başlamak gerekir. Bu problemi aşmak için referans noktasının elde edildiği noktanın yönsel histogramından elde edilen açısal bilgisi başlangıç yönü olarak alınır. Böylece parmakizinin tekrar sisteme tanıtıldığında dönmesinden kaynaklanan hatalı FingerCode elde edilmesi önlenmiş olur.

Fingercode simgelenişi, parmağın kuru ya da kirli oluşundan dolayı parmakizi görüntüsündeki gürültüyü göz önünde bulunduran herhangi açık bir prosedüre sahip değildir. Sektörler sabit bir ortalama ve varyans değerine normalize edilmesine ve daha sonra Gabor filtre kümeleri ile filtrelenmesine rağmen, çok büyük miktardaki gürültü gri-seviyeli görüntü karakteristiklerini değiştirir ve sektörlerdeki ayırıcı bilginin ölçülmesinde problemlere neden olur. Bu tez çalışmasında önerilen basit varyans tabanlı özellikler, iyi ayırıcı güce sahip olmakla birlikte parmakizi görüntüsündeki gürültü ile ayrıntı tabanlı özelliklerden daha iyi sonuç almaya imkan vermektedir. Ne var ki filtrelenmiş görüntülerdeki sektörlerden daha zengin ve daha ayırıcı özellikler elde edilecek ve eşleştirme performansını

artıracak yöntemler geliştirilmelidir. Mevcut eşleştirme algoritması oldukça basit olmasına rağmen daha çabuk bir eşleştirme algoritması ile doğrulama performansı artırılabilir. Mesela, her bir sektörden elde edilen eşleştirme sonuçları görüntü kalitesine göre farklı olabilir. Bu yüzden doğrulama sistemi, parmakizlerindeki belirsiz bilgiyi tutabilen bir eşleştiriciden faydalanabilir.

Parmakizinden özellik vektörü elde edilirken sektörizasyon aşamasında kullanılacak bant ve sektör sayısının seçimi önemlidir. Bant sayısının fazla seçilmesi durumunda görüntüden taşmalar olabilmekte ve dolayısı ile parmakizine ait olmayan arka plan görüntü bilgileri de özellik vektörü olarak elde edilebilmektedir. Buna karşın bant sayısının az seçilmesi durumunda parmakizine ait yeterince bilginin elde edilemediği gözlemlenmiştir. Sektör sayısının fazla seçilmesi durumunda, parmakizinin sınırlı ve ayrıntılı bilgisi tam olarak elde edilemeyebilir. Bu sayının az seçilmesi durumunda ise parmakizinin sınırlı ve ayrıntılı bilgisi, parmakizini simgelemeyecek kadar az olacaktır. Bu da YSA ile parmakizini tanımada problem oluşturacaktır. Parmakizlerine ait çok büyük boyutlu özellik vektörü elde edilmesi durumunda ise YSA ile tanıma işlemi uzun zaman alacaktır. Dolayısı ile özellik vektörlerinin veri azaltma, boyut indirgeme gibi metotlarla seçilerek azaltılması gerekir. Bunun için bu tez çalışmasında kullanılan ve iyi sonuçların alındığı Genetik Algoritmalardan başka yöntemler de kullanılabilir.

9. KAYNAKLAR

Adams (2000) Adams J., Survey: Biometrics, Smart Cards, BBT, pp.8-11, 2000.

Akaike (1974) Akaike H., A new look at statistical model identification, IEEE Transactions on Automatic Control, vol. 19, pp. 716-723, 1974.

Algil (2003) Algil M., Dalgacık dönüşümü ile yapay sinir ağlarını kullanarak ses dalgalarının tanımlanması, Erciyes Üniversitesi, Fen Bilimleri Enstitüsü, Yüksek Lisans, 2003.

Aliev ve Aliev (2001) Aliev R.A, Aliev R.R., Soft Computing and its Application, World Scientific Publishing Co. Pte. Ltd., 2001.

Allahverdi (2002) Allahverdi N., Uzman Sistemler: Bir Yapay Zeka Uygulaması, Atlas Yayın Dağıtım, ISBN: 9756574119, 2002.

Allahverdi ve ark. (2004) Allahverdi N., Altun A.A., Koçer H.E., Kriminalistikte Parmakizi İncelemesi ve Otomatik Parmakizi Teşhis Sistemi, Polis Dergisi, sayı: 41, s. 121-125, 2004.

Almansa ve Lindeberg (1997) Almansa A., Lindeberg T., Enhancement of Fingerprint Images Using Shape-Adapted Scale-Space Operators, in Gaussian Scale-Space Theory, I Sporring, M. Nielsen, L, Florack, P. Johansen (Eds.), pp. 21-30, Kluwer, New York, 1997.

Almansa ve Lindeberg (2000) Almansa A., Lindeberg T., Fingerprint Enhancement by Shape Adaptation of Scale-Space Operators with Automatic Scale Selection, IEEE Transactions on Image Processing, vol. 9, no. 12, pp. 2027-2042, 2000.

Altıparmak ve ark. (2000) Altıparmak F., Dengiz B., Smith A.E., An Evolutionary Approach For Reliability Optimization in Fixed Topology Computer Networks, Transactions On Operational Research, Vol. 12, Number: 1-2, s. 57-75, 2000.

Altun (2002) Altun A.A., Biyometrik Tanıma Sistemleri ve Bir Örnek : Parmak İzi, Selçuk Üniversitesi, Fen Bilimleri Enstitüsü, Doktora Semineri, 2002.

Altun ve Allahverdi (2006) Altun A.A., Allahverdi N., Recognition of Fingerprints Enhanced by Contourlet Transform with Artificial Neural Networks, 28th International Conference on Information Technology Interfaces, pp. 167-172, Croatia, 2006.

Altun ve ark. (2005) Altun A.A., Allahverdi N., Kocer H.E., Yilmaz T., Alan S., Fingerprint image enhancement using filtering techniques, Proceedings of the IEEE 13th Signal Processing, Communications Applications Conference, pp. 229-232, 2005.

Altun ve ark. (2005a) Altun A.A., Allahverdi N., Koçer H.E., 2 Boyutlu Gabor Filtre Kullanarak Yapay Sinir Ağları ile Parmakizi Analizi, 4. Uluslararası İleri Teknolojiler Sempozyumu, 1.cilt, s. 10-14, 2005.

Altun ve Koçer (2003) Altun A.A., Koçer H.E., Biyometrik Sistemlerin Güvenlik Alanında Kullanılması ve Otomatik Parmakizi Tanıma Sistemleri, 1. Polis Bilişim Sempozyumu, Ankara, 2003.

Amy (1948) Amy L., Recherches sur Lidentification des Traces Papillaries, Annales de Medecine Legale, vol. 28, no. 2, pp. 96-101, 1948.

Anderson ve McNeill (1992) Anderson D., McNeill G., Artificial Neural Networks Technology, A DACS State-of-the-Art Report, Contract Number F30602-89-C-0082, (Data & Analysis Center for Software) , 1992.

Anthony ve Barlett (1999) Anthony M., Bartlett P.L., Neural Network Learning: Theoretical Foundations, Cambridge University Press, ISBN: 052157353X, 1999.

Avcı ve Yıldırım (2002) Avci M., Yildirim T., Classification of escherichia coli bacteria by artificial neural networks, First International IEEE Symposium of Intelligent Systems, Vol. 3, pp: 13- 16, 2002.

Bala ve Ertüzün (2002) Bala E., Ertüzün A., Çoklu-Dalgacik (Multiwavelet) Teknikleri ile İmgelerden Gürültü Temizleme Uygulamaları (in Turkish), SIU 2002 (10th National Signal Processing and Applications Conf.), pp. 1232-1237, 2002.

Ballan (2000) Ballan M., Parmakizi Tanıma Algoritmalarına İki Yeni Yaklaşım, Yüksek Lisans Tezi, GYTE, 2000.

Ballan ve ark. (1997) Ballan M., Sakarya F.A., Evans B.L., A fingerprint classification technique using directional images, Thirty-First Asilomar Conference on Signals, Systems & Computers, Vol. 1, pp. 101-104, 1997.

Balthazard (1911) Balthazard V., De lidentification par les Empreintes Ditalis,Comptes Rendus, des Academies des Sciences, vol. 1862, no. 152, 1911.

Banner ve Stock (1974) Banner C.B., Stock R.M., Finder, the FBIs Approach to Automatic Fingerprint Identification, in Proc. Conf, on Science of Fingerprints, pp. 15-49, 1974.

Başaran (2003) Başaran B., Dağılım tahmini algoritmaları kullanarak en uygun yapay sinir ağı yapısının bulunması, Boğaziçi Üniversitesi, Fen Bilimleri Enstitüsü, Yüksek Lisans, 2003.

Baxes (1994) Baxes G.A., Digital Image Processing. Principles & Applications, Wiley & Sons Pub., 1994.

Bazen ve ark. (2000) Bazen A.M., Verwaaijen G.T.B., Gerez S.H., veelenturf L.P.J., van der Zwaag B.J., A Correlation-Based Fingerprint Verification System, in Proc. Workshop on Circuits Systems, Signal Processing (PruRISC 2000), pp. 205-213, 2000.

BenAbdelkader ve ark. (2002) BenAbdelkader C., Cutler R., Davis L., Motion-based recognition of people in EigenGait space, Fifth IEEE International Conference on Automatic Face, Gesture Recognition, pp: 254-259, 2002.

Bertsekas (1995) Bertsekas D. P., Nonlinear Programming, Belmont, MA: Athena Scientific, ISBN 1-886529-14-0, 1995.

Bertsekas ve Tsitsiklis (1996) Bertsekas D. P., Tsitsiklis J. N, Neuro-Dynamic Programming, Belmont, MA: Athena Scientific, ISBN 1-886529-10-8., 1996.

Bigun ve ark. (1997) Bigun E.S., Bigun J., Due B., Fischer S., Expert Conciliation for Multi Modal Person Authentication Systems by Bayesian Statistics, in Proc. Int. Conf. on Audio-, Video-Based Biometric Personal Authentication (1st), pp. 291-300, 1997.

Bigun ve Granlund (1987) Bigun J., Granlund G.H., Optimal Orientation Detection of Linear Symmetry, in Proc. Int. Conf. on Computer Vision (1st), pp. 433-438, 1987.

Bishop (1992) Bishop, C. M., Exact calculation of the Hessian matrix for the multilayer perceptron. Neural Computation 4(4), 494–501, 1992.

Bishop (1995) Bishop C., Neural Networks for Pattern Recognition, Oxford Press, Oxford, UK, 1995.

Bjorn (2002) Bjorn V., Enterprise Security Architecture for Biometric User Authentication Systems, DigitalPersona Inc., 2002.

Blomme (2003) Blomme J., Evaluation of biometric security systems against artificial fingers, Masters thesis LITH-ISY-EX-3514-2003, Department of Electrical Engineering, Linkoping University, Linkoping, Sweden, 2003.

Blue ve ark. (1993) Blue J.L., Candela G.T., Grother PJ., Chellappa R., Wikinson R., Wilson C, Evaluation of Pattern Classifiers for Fingerprint, OCR Applications, Tech. Report NIST TR 3162, 1993.

Bolle ve ark. (2004) Bolle R. M., Connell J. H., Pankanti S., Ratha N. K., Andrew W. Senior, Guide To Biometrics, Springer-Verlag New York, Inc, 2004.

Bormann (2005) Bormann F., Manteau L., Linke A., Integrated Secure Platform for Interactive Trusted Personal Device: The Future of Smart Cards, SIT Smart Card Workshop, Darmstadt, 2005.

Bose (1917) Pankanti S., Prabhakar S., Jain A.K., On the Individuality of Fingerprints, IEEE Transactions on Pattern Analysis, Machine Intelligence, vol. 24, no. 8, pp. 1010-1025, 2002.

Bowman (2000) Bowman E., Everything You Need to Know About Biometrics, from http: //www.ibia.org, (Erişim Tarihi: 2000).

Braysy (2001) Braysy O., Local Search, Variable Neighborhood Search Algorithms for The Vehicle Routing Problem with Time Windows, PHD Thesis, Department of Mathematics, Statistics, University of Vaasa, Finland, 2001.

Breiman (1994) Breiman L., Heuristics of instability in model selection, Technical report, Department of Statistics, University of California at Berkeley, 1994.

Brislawn ve ark. (1996) Brislawn C.M., Bradley J.N., Onyshczak RJ., Hopper T., The FBI Compression Standard for Digitized Fingerprint Images, Proc. of SPIE (Applications of Digital Image Processing XIX), vol. 2847, 1996.

Burnham ve Anderson (2002) Burnham K.P., Anderson D.R., Model Selection and Multimodel Inference: A Practical-Theoretic Approach, 2nd Edition. Springer, New York. 2002.

Cappelli ve ark. (2000a) Cappelli R., Maio D., Maltoni D., Combining Fingerprint Classifiers, in Proc. Int. Workshop on Multiple Classifier Systems (1st), pp. 351-361, 2000.

Cappelli ve ark. (2000b) Cappelli R., Maio D., Maltoni D., Indexing Fingerprint Databases for Efficient 1: N Matching, in Proc. Int. Conf. on Control Automation Robotics, Vision (6th), 2000.

Cappelli ve ark. (2001a) Cappelli R., Maio D., Maltoni D., Modelling Plastic Distortion in Fingerprint Images, in Proc. Int. Conf. on Advances in Pattern Recognition (2nd), pp. 369-376, 2001.

Cappelli ve ark. (2001b) Cappelli R., Maio D., Maltoni D., Multi-Space KL for Pattern Representation, Classification, IEEE Transactions on Pattern Analysis, Machine Intelligence, vol. 23, no. 9, pp. 977-996, 2001.

Castelman (1996) Castelman, R. K., Digital Image Processing, Prentice Hall, Englewood Cliffs, New. Jersey, USA, 1996.

Chang (1980) Chang T., Texture Analysis of Digitized Fingerprints for Singularity Detection, in Proc. Int. Conf. on Pattern Recognition (5th), pp. 478-480, 1980.

CJIS (1999) Criminal Justice Information Services, Electronic Fingerprint Transmission Specification, Int. Report. CJIS-RS-0010 (V7), available at: http: //www.fbi.gov/hq/cjisd/iafis/efts70/covei.hlui, 1999.

Coşkun (2006) Coşkun A., Bir Optimizasyon Yöntemi: Genetik Algoritma "Molga Programı ile İterasyon Sayılarının Değerlendirilmesi", G.Ü. Endüstriyel Sanatlar Eğitim Fakültesi Dergisi, Sayı 18, s. 49-55, 2006.

Coşkun ve Yıldırım (2003) M. Avci, T. Yıldırım , Classification of Escherichia Coli Bacteria by Artificial Neural Networks, IEEE International Symposium on Intelligent Systems, Vol III, Varna, Bulgaristan, sf. 16-20, 2002.

Cummins ve Midlo (1943) Cummins H., Midlo C, Fingerprints, Palms, Soles, Dover, New York, 1943.

Curram ve Minger (1994) Curram S.P., Minger J., Neural Networks, Decision .Tree Induction, Discriminant Analysis: Embrical Comprasion, p.441, 1994.

Çikoğlu (2003) Çikoğlu S., Yapay sinir ağları ile imza tanıma, Sakarya Üniversitesi, Fen Bilimleri Enstitüsü, Yüksek Lisans, 2003.

Daugman (1993) Daugman J. G., High Confidence Recognition of Persons by a Test of Statistical Independence, IEEE Trans. Pattern Anal., Machine Intell., Vol. 15, No. 11, pp. 1148-1161, 1993.

Daugman (1999) Daugman J., Recognizing Persons by Their Iris Patterns, in Biometrics: Personal Identification in a Networked Society, A.K. Jain, R. Bolle, S. Pankanti (Eds.), Kluwer Academic, New York, 1999.

Davis (1975) Davis L. S., A survey of edge detection techniques, Comput. Graphics Image Processing, vol. 4, pp. 248-270, 1975.

Davis (1994) Davis L., Genetic Algorithms, Financial Applications, Deboeck GJ. (ed) Trading on the Edge. Wiley, New York, 133–147, 1994.

DeJong (1975) DeJong K. A., An analysis of the behavior of a class of genetic adaptive systems, PhD thesis, Univ. Michigan, Ann Arbor, 1975.

Diamantaras ve Kung (1996) Diamantaras K. I., Kung S. Y., Principal Component Neural Networks : Theory, Applications (Adaptive, Learning Systems for Signal Processing, Communications, Control Series), Wiley-Interscience, ISBN: 0471054364, 1996.

Digital Biometrics (2004) http://www.identix.com (Erişim Tarihi: 2004)

DigitalPersona (2003) http:www.digitalpersona.com (Erişim Tarihi: 2003)

Do ve Vetterli (2002) Do M.N., Vetterli M., Contourlets: a directional multiresolution image representation, International Conference on Image Processing, Vol. 1, pp. I-357-I-360, 2002.

Do ve Vetterli (2003) Do M. N., Vetterli M., Contourlets, Beyond Wavelets, G. V. Welland ed., Academic Press, 2003.

Do ve Vetterli (2005) Do M.N., Vetterli M., The contourlet transform: an efficient directional multiresolution image representation, IEEE Transactions on Image Processing, vol 14, Issue 12, pp. 2091- 2106, 2005.

Doğan ve Alp (2002) Doğan A., Alp K., Yapay Zeka, Kariyer Yayıncılık, ISBN: 9758515373, 2002.

Donoho ve Johnston (1994) Donoho D. L., Johnston I. M., Ideal Spatial Adaptation via Wavelet Shrinkage, Biometrika, Vol. 81, pp-425-455, 1994.

Dönmez (2001) Dönmez M., Polis Akademisi ders notları, http://www.geocities.com/mdonmez1/ (Erişim Tarihi: 2001)

Dugelay ve ark. (2002) Dugelay J.-L., Junqua J.-C., Kotropoulos C., Kuhn R., Perronnin F., Pitas I., IEEE International Conference on Acoustics, Speech, Signal Processing, pp: IV-4060- IV-4063, vol.4, 2002.

Duncan ve ark. (2004) Duncan D., Po Y., Do M. N., Directional Multiscale Modeling Of Images Using The Contourlet Transform, IEEE Transactions On Image Processing, 2004.

Efe ve Kaynak (2005) Efe Ö., Kaynak O., Yapay Sinir Ağları, Uygulamaları, Boğaziçi Üniversitesi Yayınevi(696) , 2005.

Ellsworth ve Moran (2003) Ellsworth D., Moran P.J., Accelerating Large Data Analysis By Exploiting Regularities, 14th IEEE Visualization 2003, pp. 561-568, 2003.

Elman (1990) Elman J.L., Finding structure in time, Cognitive Science, 14, pp. 179-211, 1990.

Elmas (2003) Elmas Ç., Yapay Sinir Ağları (Kuram, Mimari, Uygulama), Seçkin Yayıncılık, ISBN: 9753476124, 2003.

Emel ve Taşkın (2002) Emel G. G., Taşkın Ç., Genetik Algoritmalar, Uygulama Alanları, Uludağ Üniversitesi İ.İ.B.F. Dergisi, Cilt 21, Sayı 1, s. 129-152, 2002.

Engin (2001) Engin O., Akış tipi çizelgeleme problemlerinin genetik algoritma ile çözüm performansının artırılmasında parametre optimizasyonu, İstanbul Teknik Üniversitesi, Fen Bilimleri Enstitüsü, Doktora, 2001.

Er ve ark. (2005) Er H., Çetin M.K., Çetin E.İ., Finansta Evrimsel Algoritmik Yaklaşımlar: Genetik Algoritma Uygulamaları, Akdeniz Üniversitesi İ.İ.B.F. Dergisi, vol.10, pp. 73-94, 2005.

Erçelebi ve Koç (2005) Erçelebi E., Koç S., Image enhancement by wavelet domain median filter, 13th Signal Processing and Communications Applications Conference, pp. 123-126, 2005.

Ergezer ve ark. (2003) Ergezer H., Dikmen M., Özdemir E., Yapay sinir ağları ve tanıma sistemleri, Pivolka, 2(6), s. 14-17, 2003.

Erler ve ark. (2003) Erler M., Sağıroğlu Ş., Beşdok E., Mühendislikte Yapay Zeka Uygulamaları 1 / Yapay Sinir Ağları, Ufuk Kitap-Kayseri, ISBN: 9759594854, 2003.

Eslami ve Radha (2003) Eslami R., Radha H., On low bit-rate coding using the contourlet transform, Conference Record of the Thirty-Seventh Asilomar Conference on Signals, Systems, Computers, Vol. 2, pp. 1524–1528, 2003.

Eslami ve Radha (2005) Eslami R., Radha H., Image Denoising Using Translation-Invariant Contourlet Transform, IEEE International Conference on Acoustics, Speech, Signal Processing, Volume 4, pp. 557-560, 2005.

Fahlman (1988) Fahlman S.E., An Empirical Study of Learning Speed in Back-Propagation Networks, Proceedings of the 1988 Connectionist Models Summer School, Morgan Kaufmann, 1988.

Famili ve ark. (1997) Famili A., Shen W, Weber R., Simoudis E., Data Preprocessing, Intelligent Data Analysis, Intelligent Data Analysis, 1, USA, pp.3-23., 1997.

Fausett (1994) Fausett L.V., Fundamentals of Neural Networks, Prentice Hall; 1st edition, ISBN: 0133341860, 1994.

FBI (1984) Federal Bureau of Investigation, The Science of Fingerprints: Classification, Uses, U.S. Government Publication, Washington, DC, 1984.

FBI (1991) Federal Bureau of Investigation, The FBI Fingerprint Identification Automation Program: Issues, Options, U.S. Government Publication, U.S. Congress, Office of Technology Assessment, Washington, DC, 1991.

Fığlalı (2002) Fığlalı A., Engin O., Genetik Algoritmalarla Akış Tipi Çizelgelemede Üreme Yöntemi Optimizasyonu, İTÜ Dergisi, s. 1-6, 2002.

Fine (1999) Fine T. L., Feedforward Neural Network Methodology (Springer Series in Statistics), Springer, ISBN: 0387987452, 1999.

Fletcher ve Revees (1964) Fletcher R., Reeves C. M., Function minimization by conjugate gradients, Computer Journal, 7, 149–154, 1964.

Freeman ve Skapura (1991) Freeman J. A., Skapura D. M., Neural Networks: Algorithms, Applications, Programming Techniques, Addison Wesley Pub. , 1991.

Fung ve ark. (2001) Fung R.Y.K., Tang J., Wang D., Extension Of A Hybrid Genetic Algorithm For Nonlinear Programming Problems with Equality, Inequality Constraints, Computers & Operations Research, Vol. 29, Issue: 3, s. 261-274, 2001.

Galton (1892) Galton F., Finger Prints, McMillan, London, 1892.

Galton (2003) www.galton.org (Erişim tarihi: 2003)

Geçgil (2000) Geçgil E., Parmakizinin Ayırt Edici Özelliklerinin Kullanılması Suretiyle Analiz Edilmesi ve Otomatik Ön İşleme Yöntemlerinin Programlanması, Yüksek Lisans Tezi, Gazi Üniversitesi F.B.E., 2000.

Ghada ve John (2005) Ghada B., John B. O., Enhancing trie-based syntactic pattern recognition using AI heuristic search strategies, Lecture Notes in Computer Science, v 3686, n PART I, Pattern Recognition, Data Mining: Third International Conference on Advances in Pattern Recognition, ICAPR 2005. Proceedings, p 1-17, 2005.

Goldberg (1985) Goldberg D. E., Optimal initial population size for binary-coded genetic algorithms, Technical Report TCGA-850001, University of Alabama, 1985.

Goldberg (1989) Goldberg D., Genetic Algorithms in Search, Optimization, Machine Learning, Addison-Wesley, Reading, MA, 1989.

Golden (1986) Golden R. M., Brain-State-in-a-Box Neural Model is a Gradient Descent Algorithm, Journal of Mathematical Psychology, 1986.

Göknar ve Yıldırım (2005) Göknar G. Yıldırım T., Hand gesture recognition using artificial neural networks, IEEE 13th Signal Processing, Communications Applications Conference, pp. 210- 213, 2005.

Görz ve Nebel (2006) Görz G., Nebel B., Yapay Zeka, İnkılap Kitabevi, ISBN: 9751024056, 2006.

Grasselli (1969) Grasselli A., On the Automatic Classification of Fingerprints, in Methodologies of Pattern Recognition, S. Watanabe (Ed.), Academic, New York, 1969.

Greenberg ve ark. (2000) Greenberg S., Aladjem M., Kogan D., Dimitrov L, Fingerprint Image Enhancement Using Filtering Techniques, in Proc. Int. Conf. on Pattern Recognition (15th), vol. 3, pp. 326-329, 2000.

Grossberg (1969) Grossberg S., Some Networks That Can Learn, Remember, Reproduce any Number of Complicated Space-Time Patterns, Journal of Mathematics, Mechanics, Volume 19, 1969.

Gupta (1968) Gupta S.R., Statistical Survey of Ridge Characteristics, Int. Criminal Police Review, vol. 218, no. 130, 1968.

Gülver (1999) Gülver M., Parmak izi klasifikasyonu, karşılaştırılması, AFİS (Automated Fingerprint Idendification Systems) yazılımlarının değerlendirilmesi, İstanbul Üniversitesi, Adli Tıp Enstitüsü, Yüksek Lisans, 1999.

Hagan ve ark. (1996) Hagan M., Demuth H., Beale M., Neural Network Design, Boston, MA: PWS Publishing, 1996.

Ham ve Atkinson (2002) Ham S., Atkinson R. D., Modernizing the State Identification System, An Action Agenda, Progressive Policy Institute, 2002.

Han ve Kamber (2000) Han J., Kamber M., Data Mining: Concepts, Techniques (The Morgan Kaufmann Series in Data Management Systems), Morgan Kaufmann, ISBN: 1558604898, 2000.

Hatami ve ark. (2005) Hatami S., Hosseini R., Kamarei M., Ahmadi H., Wavelet Based Fingerprint Image Enhancement, IEEE International Symposium on Circuits, Systems, Pp.: 4610-4613, 2005.

Haupt ve Haupt (2004) Haupt R.L., Haupt S.E., Practical Genetic Algorithms, Wiley-IEEE Publisher, 272 p., 2004.

Haykin (1994) Haykin, S., Neural Networks: A Comprehensive Foundation, Macmillan College Publishing, New York, 1994.

Haykin (1998) Haykin S., Neural Networks: A Compreensive Foundation, Prentice Hall, 2.ed., 1998.

Hebb (1949) Hebb D. O., The Organization of Behavior. New York: John Wiley, 1949.

Henry (1900) Henry E., Classification, Uses of Finger Prints, Routledge, London, 1900.

Herschel (1918) Herschel W., Archiv Fur Kriminologie, vol. 70, 1918.

Hirotugu (1974) Hirotugu A., A new look at the statistical model identification, IEEE Transactions on Automatic Control 19 (6): 716–723, 1974.

Holland (1975) Holland J. H., Adaptation in Natural, Artificial Systems. University of Michigan Press, Ann Arbor., 1975.

Holland (1992) Holland J.H., Genetic Algorithms, Scientific American, 267, (1). 4450., 1992.

Hollingum (1992) Hollingum J., Automated Fingerprint Analysis Offers Fast Verification, Sensor Review, vol. 12, no. 3, pp. 12, 1992.

Hong (1998) Hong L., Automatic Personal Identification Using Fingerprints, Ph. D. Thesis, Department of Computer Science, Engineering, Michigan State University, East Lansing, 1998.

Hong ve ark. (1996) Hong L., Jain A.K., Pankanti S., Bolle R., Fingerprint Enhancement, in Proc. Workshop on Applications of Computer Vision, pp. 202-207, 1996.

Hong ve ark. (1998) Hong L., Wan Y., Jain A.K., Fingerprint Image Enhancement: Algorithms, Performance Evaluation, IEEE Transactions on Pattern Analysis, Machine Intelligence, vol. 20, no. 8, pp. 777-789, 1998.

Hong ve ark. (1999) Hong L., Jain A.K., Pankanti S., Can Multibiometrics Improve Performance?, Workshop on Automatic Identification Advanced Technologies, pp. 59-64. 1999.

Hong ve Jain (1996) Hong L., Jain A.K., On-Line Fingerprint Verification, in Proc. Int. Conf. on Pattern Recognition (15th), pp. S96-6OU, 1996.

Hong ve Jain (1999) Hong L., Jain A.K., Classification of Fingerprint Images, in Proc. Scandinavian Conf. on Image Analysis (1 Ith), 1999.

Hong ve Jian-Hua (2003) Hong H., Jian-Hua L., Fingerprint matching using ANFIS, Systems, IEEE International Conference on Man, Cybernetic, Pp.: 217-222 vol.1, 2003.

Horton ve ark. (2002) Horton, M.; Meenen, P.; Adhami, R.; Cox, P., The costs, benefits of using complex 2-D Gabor filters in a filter-based fingerprint-matching system, Proceedings of the Thirty-Fourth Southeastern Symposium on System Theory, Pp.: 171-175, 2002.

İnandık (1998) İnandık Ö., Öznitelik Tabanlı Otomatik Parmakizi Eşleme, Yüksek Lisans Tezi, İTÜ, 1998.

İşçi ve Korukoğlu (2003) İşçi Ö., Korukoğlu S., Genetik Algoritma Yaklaşımı, Yöneylem Araştırmasında Bir Uygulama, Celal Bayar Üniversitesi Yönetim, Ekonomi Dergisi, 2, 191/208, 2003.

Jain ve ark. (1997) Jain A.K., Hong L., Pankanti S., Bolle R., An Identity Authentication System Using Fingerprints, Proceedings of the TEEE, vol. 85, no. 9, pp. 1365-1388, 1997.

Jain ve ark. (1997a) Jain A.K., Hong L., Bolle R., On-line Fingerprint Verification, IEEE Transactions on Pattern Analysis, Machine Intelligence, vol. 19, no. 4, pp. 302-313, 1997.

Jain ve ark. (1999a) Jain L.C., Halici U., Hayashi I., Lee S.B., Intelligent Biometric Techniques in Fingerprint, Face Recognition, CRC Press, Boca Raton, FL, 1999.

Jain ve ark. (1999b) Jain A.K., Prabhakar S., Hong L., Pankanti S., FingerCode: a filterbank for fingerprint representation, matching, IEEE Computer Society Conference on Computer Vision, Pattern Recognition Volume 2, 1999.

Jain ve ark. (1999c) Jain A. K., Prabhakar S., Pankanti S., A Filterbank-based Representation for Classification, Matching of Fingerprints, International Joint Conference on Neural Networks, Pp.: 3284-3285, vol.5, 1999.

Jain ve ark. (1999d) Jain A.K., Prabhakar S., Hong L., A Multichannel Approach to Fingerprint Classification, IEEE Transactions on Pattern Analysis, Machine Intelligence, vol. 21, no. 4, pp. 348-359, 1999.

Jain ve ark. (2000) Jain A.K., Prabhakar S., Hong L., Pankanti S., Filterbank-Based Fingerprint Matching, IEEE Transactions on Image Processing, vol. 9, pp. 846-859, 2000.

Jain ve ark. (2001) Jain A.K., Pankanti S., Prabhakar S., Ross A., Recent Advances in Fingerprint Verification, in Proc. Int. Conf. on Audio-, Video-Based Biometric Person Authentication (3rd), -pp. 182-191, 2001.

Jain ve ark. (2004) Jain A. K., Ross A., Prabhakar S., An Introduction to Biometric Recognition, IEEE Transactions On Circuits, Systems For Video Technology, Vol. 14, No. 1, pp. 4-20, 2004.

Jain ve Farrokhnia (1991) Jain A.K., Farrokhnia F., Unsupervised Texture Segmentation Using Gabor Filters, Pattern Recognition, vol. 24, no. 12, pp. 1167-1186, 1991.

Jain ve Pankanti (2001) Jain A.K., Pankanti S., Biometrics Systems: Anatomy of Performance, IEICE Transactions on Information, Systems (Special Issue on Biometrics), vol. E84-D, no. 7, pp. 788-799, 2001.

Jang (1997) Jang J. S. R., Neuro-Fuzzy, Soft Computing: A Computational Approach To Learning, Machine Intelligence, Chapter 7: Derivative-Free Optimization, Prentice-Hall, USA, s. 173-196., 1997.

Janikow (1993) Janikow C. Z., A knowledge-intensive genetic algorithm for supervised learning, Machine Learning, 1(13):169-228, 1993.

Jiang (2001) Jiang X., A Study of Fingerprint Image Filtering, in Proc. Int. Conf. on Image Processing, vol. 3, pp. 238-241, 2001.

Jordan (1986) Jordan M.I., Serial order: A parallel distributed processing approach. Technical report, Institute for Cognitive Science. University of California, 1986.

Karahan ve Ceylan (2004) Karahan H., Ceylan H., Akış serilerinin genetik algoritma ile tahmini, 6th International Conference Advances in Civil Engineering, pp. 1289-1298, İstanbul, 2004.

Karlık (1994) Karlık B., Çok fonksiyonlu protezler için yapay sinir ağları kullanılarak miyoelektrik kontrol, Yıldız Teknik Üniversitesi Fen Bilimleri Enstitüsü, Doktora Tezi, 1994.

Karu ve Jain (1996) Karu K., Jain A.K., Fingerprint Classification, Pattern Recognition, vol. 29, no. 3, pp. 389-404, 1996.

Kass ve Witkin (1987) Kass M., Witkin A., Analyzing Oriented Patterns, Computer Vision, Graphics, Image Processing, vol. 37, no. 3, pp. 362-385, 1987.

Kawagoe ve Tojo (1984) Kawagoe M., Tojo A., Fingerprint Pattern Classification, Pattern Recognition, vol. 17, pp. 295-303, 1984.

Kaygısız (1995) Kaygısız M., Kriminalistikte Parmakizi İncelemesi, Yüksek Lisans Tezi, İstanbul Ü. Adli Tıp Ens., 1995.

Kim ve ark. (2002) Kim B.G.. Kim H.J., Park D.J., New Enhancement Algorithm for Fingerprint Images, in Proc. Int. Conf. on Pattern Recognition (16th), vol. 3, pp. 879-892, 2002.

Kingston (1964) Kingston C, Probabilistic Analysis of Partial Fingerprint Patterns, Ph.D. Thesis, University of California, Berkeley, 1964.

Kirsch (1971) Kirsch R., Computer determination of the constituent structure of biological images, Comput. Biomed., Res., 4, pp. 315-328, 1971.

Khmanee ve Nguyen (2004) Khmanee, C.; Nguyen, D., 2004, On the design of 2D Gabor filtering of fingerprint images, First IEEE Consumer Communications, Networking Conference, Pp.: 430-435, 2004.

Koçsistem (2003) http://www.kocsistem.com.tr/tr/ch_pro_bio.asp (Erişim Tarihi: 2003)

Kohonen (1984) Kohonen T., Self-Organization, Associative Memory, volume 8 of Springer Series in Information Sciences, Springer-Verlag, New York, 1984.

Kohonen ve ark. (1988) Kohonen T., Barna G., Chrisley R.; Statistical pattern recognition with neural networks: benchmarking studies, Proc. of ICNN, vol. 1, pp. I-61–I-68, 1988.

Kovacs-Vajna (2000) Kovacs-Vajna Z.M., A Fingerprint Verification System Based on Triangular Matching, Dynamic Time Warping, IEEE Transactions on Pattern Analysis, Machine Intelligence, vol. 22, pp. 1266-1276, 2000.

Koza (1992) Koza, J. R., A Genetic Approach to Econometric Modelling, in Bourgine, P., B. Walliser (eds.), Economics, Cognitive Science, Pergamon Press, s. 57-75, 1992.

Kozek ve ark. (1993) Kozek T., Roska T.ve Chua L. O., Genetic Algorithm for CNN Template Learning. IEEE Trans. Circuits, Syst., 40(I): 392--402, 1993.

Langdon ve Poli (2002) Langdon W. B., Poli R., Foundations of Genetic Programming, Springer-Verlag, 260 p., 2002.

Lawson (2003) Lawson W. J., Enhancing Assistive Technologies: Through the Theoretical Adaptation of Biometric Technologies to People of Variable Abilities, Phd Thesis, Kennedy-Western University, 2003.

Lee ve ark. (1999) Lee J.W.. Min PI., Kim I., Kim W.. A 600 dpi Capacitive Fingerprint Sensor Chip, Image Synthesis Technique, IEEE Journal of Solid-State Circuits, vol. 34, no. 4, pp. 469-475, 1999.

LeGrand (2004) LeGrand B. T., Using a Genetic Algorithm for Means-Based Path-Planning in a Weighted Environment, Washington University, Department of Computer Science, Msc Thesis, 2004.

Leung ve ark. (1990) Leung M, Engeler W., Frank P., Fingerprint Image Processing Using Neural Network, in Proc. IEEE Region 10 Conf. on Computer, Comm. Systems, 1990.

Lin ve Wilson (1988) Lin H. M., Wilson A. N., Median filters with adaptive length, IEEE Trans. Circuits Svst., vol. 35, no. 6, 1988.

Liu ve Li (2004) Liu Y.-N., Li Z.-H., Research on fingerprint enhancement, classification algorithm, The Fourth International Conference on Computer, Information Technology, Pp.: 199-204, 2004.

Liu ve Silverman (2001) Liu S., Silverman M., A practical guide to biometric security technology, IT Professional, Volume 3, Issue 1, pp. 27-32, 2001.

Low ve Togneri (1998)

Luenberger (2003) Luenberger D. G., Linear, Nonlinear Programming, Second Edition, Springer, ISBN: 1402075936, 2003.

Maillet ve Sharaiha (1999) Maillet S.M., Sharaiha Y. M., Binary Digital Image Processing, 3nd edition, Portland OR, 1999.

Maio ve Maltoni (1997) Maio D., Maltoni D., Direct Gray-Scale Minutiae Detection in Fingerprints, IEEE Transactions on Pattern Analysis, Machine Intelligence, vol. 19, no. 1, 1997.

Maio ve Maltoni (1998a) Maio D., Maltoni D., Ridge-Line Density Estimation in Digital Images, in Proc. Int. Conf. on Pattern Recognition (14th), pp. 534-538, 1998.

Maio ve Maltoni (1998b) Maio D., Maltoni D., Neural Network Based Minutiae Filtering in Fingerprints, in Proc. Int. Conf. on Pattern Recognition (14th), pp. 1654-1658, 1998.

Maio ve Maltoni (1999a) Maio D., Maltoni D., A Secure Protocol for Electronic Commerce Based on Fingerprints, Encryption, in Proc. World Conf. on Systems Cybernetics, Informatics; vol. 4, pp. 519-525, 1999.

Maio ve Maltoni (1999b) Maio D., Maltoni D., Minutiae Extraction, Filtering from Gray-scale Images, in Intelligent Biometric Techniques in Fingerprint & Face Recognition, L.C. Jain, U. Halici, L Hayashi, S.B. Lee (Eds.), CRC Press, Boca Raton, FL, 1999.

Maltoni ve ark. (2003) Maltoni D., Maio D., Jain A.K., Prabhakar S., Handbook of Fingerprint Recognition, Springer Verlag, New York, Inc., 2003.

Man ve ark. (1999) Man K.F., Tang K.S., Kwong S., Genetic Algorithms, Springer Verlag, 350 p., 1999.

Marr ve Hildreth (1980) Marr D., Hildreth E.C., Theory of Edge Detection, Proc. Royal Society London, B 207, pp. 187-217, 1980.

McClelland ve Rumelhart (1986) McClelland J., Rumelhart D., Explorations in Parallel Distributed Processing. MIT Press, Cambridge, MA, 1986.

Mehtre ve Murthy (1986) Mehtre B.M., Murthy N.N., A Minutia Based Fingerprint Identification System, in Proc. Int. Conf. on Avances in Pattern Recognition, Digital Techniques (2nd), 1986.

Meyer (1992) Meyer Y., Wavelets: Algorithms, Applications, SIAM, 1992.

Mitchell (1996) Mitchell M., An Introduction to Genetic Algorithms, Complex Adaptive Systems. MIT Press, Cambridge, MA, 1996.

Moallemi (1991) Moallemi, C., Classifying cells for cancer diagnosis using neural networks, IEEE Intelligent Systems, Their Applications, vol. 6/6, pp. 8-12, 1991.

Muresan ve Parks (2003) Muresan, D.D. Parks, T.W., Adaptive principal components, image denoising, International Conference on Image Processing, pp.: I- 101-4 vol.1, 2003.

Nabiyev (2003) Nabiyev V. V., Yapay Zeka, Seçkin yayımcılık, Ankara, 717p., 2003.

Nabiyev (2004) Nabiyev V.V., Yapay Zeka, Problemler-Yöntemler – Algoritmalar, Seçkin Yayıncılık, ISBN: 9753476876, 2004.

Neurointelligence (2003) http://www.alyuda.com (Erişim Tarihi: 2003)

NIST (1991) National Institute of Standards, Technology, Data Format for the Interchange of Fingerprint Information, U.S. Government Publication, Washington, DC, 1991.

NIST (1994) National Institute of Standards, Technology, Guideline for the Use of Advanced Audientication Technology Alternatives, Federal Information Processing Standards Publication 190, 1994.

Oğuzlar (2003) Oğuzlar A. veri Ön İşleme, Erciyes Üniversitesi İktisadi, İdari Bilimler Fakültesi Dergisi, Sayı: 21, sayfa: 67-76, 2003.

Ortiz ve ark. (2004) Ortiz, F. Torres, F. Gil, P., Gaussian noise elimination in colour images by vector-connected filters, 17th International Conference on ICPR 2004., pp: 807-810 Vol.4, 2004.

Osterburg ve ark. (1977) Osterburg J., Parthasarathy T., Raghaven T., Sclove S., Development of a Mathematical Formula for the Calculation of Fingerprint Probabilities Based on Individual Characteristic, Journal American Statistic Association, vol. 72, pp. 772-778, 1977.

Özbay (2004) Özbay Y., YSA Kullanarak EKG Aritmilerinin Tanınması, Sınıflandırılması, Biyomedikal Mühendisliği Ulusal Toplantısı, Boğaziçi Üniv., İstanbul , 2004.

Özbay ve Karlık (2001) Özbay Y., Karlık B., A Recognition of ECG Arrhythmias Using Artificial Neural Networks, Proceedings of the 23 rd Annual International Conference of the IEEE Engineering in Medicine, Biology Society, October 25-28 , 2001, Istanbul, 2001.

Özdemir (1999) Özdemir Ş., Manuel Parmakizi Sisteminin Bilgisayar Sistemine Adaptasyonu ve Karşılaştırılması, Yüksek Lisans Tezi, Ankara Ü. Adli Tıp Ens., 1999.

Öztemel (2003) Öztemel E., Yapay Sinir Ağları, Papatya Yayıncılık, ISBN: 9756797398, 2003.

Pankanti (2002) http://www.research.ibm.com/ecvg/biom/fp-steps.html
(Erişim Tarihi: 2002)

Pankanti ve ark. (2002) Pankanti S., Prabhakar S., Jain A.K., On the Individuality of Fingerprints, IEEE Transactions on Pattern Analysis, Machine Intelligence, vol. 24, no. 8, pp. 1010-1025, 2002.

Parker (1985) Parker D. B., Learning logic. Technical Report TR-47, Center for Computational Research in Economics, Management Science, MIT, Cambridge, MA, 1985.

Pearson (1930) Pearson K., The Life, Letters of Francis Galton, vol. HJA, University Press, Cambridge, 1930.

Pernus ve ark. (1980) Pernus F., Kovacic S., Gyergyek L., Minutiae Based Fingerprint Recognition, in Proc. Int. Conf. on Pattern Recognition (5th), pp. 1380-1382, 1980.

Po ve Do (2003) Po D.D.-Y., Do M.N., Directional multiscale modeling of images using the contourlet transform, Statistical Signal Processing, 2003 IEEE Workshop, pp. 262-265, 2003.

Poon ve Carter (1995) Poon P. W., Carter J. N., Genetic algorithm crossover operators for ordering applications, Computers & Operations Research 22(1), pp. 135-148, 1995.

Prabhakar (2001) Prabhakar S., Fingerprint Classification, Matching Using a Filterbank, Phd Thesis, Michigan State University, 2001.

Prabhakar ve Jain (2002) Prabhakar S., Jain A.K., Decision-Level Fusion in Fingerprint Verification, Pattern Recognition, vol. 35, pp. 861-874, 2002.

Prewitt (1970) Prewitt J.M.S., Object enhancement and extraction, in Picture Processing and Physhopictorics, Linkin B. and Rosenfeld A., Eds. NewYork:Academic, pp. 75-149, 1970.

Psaltis ve ark. (1988) Psaltis D., Sideris A., Yamamura A.A., A multilayered neural network controller, IEEE Control Systems Magazine, vol. 8/2, pp. 17-21, 1988.

Rao (1990) Rao A.R., A Taxonomy for Texture Description, Identification, Springer-Verlag, New York, 1990.

Rao ve Srinivas (2003) Rao M.A., Srinivas J., Neural Networks, Algorithms and Applications, Alpha Science International, 2003.

Ratha ve Bolle (2003)

Reed (1993) Reed R., Pruning algorithms-a survey, IEEE Transactions on Neural Networks, vol. 4/5, pp. 740-747, 1993.

Ripley (1996) Ripley, B.D., Pattern Recognition, Neural Networks, Cambridge: Cambridge University Pres, 1996.

Robinson (1977) Robinson G. S., Edge detection by compass gradient masks, Comput. Graphics Image Processing,'vol. 6, pp. 492-501, 1977.

Roberts (1965) Roberts L.G., Machine perception of three-dimensional solids, in Optical arid Electro-optical Information Processing, J. Tippett et al.. Eds. Cambridge, MA: MIT Press, pp. 159-197, 1965.

Ross (2003) Ross A.A., Information Fusion in Fingerprint Authentication, PhD Thesis, Michigan State University, 2003.

Roxburgh (1933) Roxburgh T., On Evidential Value of Fingerprints, Sankhya: Indian Journal of Statistics, vol. 1, pp. 189—214, 1933.

Rumelhart ve ark. (1985) Rumelhart D.E., Hinton G.E., Williams R.J., Learning Internal Representations by Error Propagation, Institute for Cognitive Science Report 8506, San Diego, University of California, 1985.

Saatci ve Tavsanoglu (2002) Saatci, E.; Tavsanoglu, V., 2002, Fingerprint image enhancement using CNN Gabor-Type filters, 7th IEEE International Workshop on Cellular Neural Networks, Their Applications Pp.: 377-382, 2002.

Sağıroğlu ve ark. (1999) Sagiroğlu S., Güney K., Erler M., Calculation of bandwidth for electrically thin, thick rectangular microstrip antennas with the use of multilayered perceptrons, International Journal of RF, Microwave Computer-Aided Engineering, vol. 9/3, pp. 277-286, 1999.

Sağıroğlu ve ark. (2003) Sağıroğlu Ş., Beşdok E., Erler M., Mühendislikte Yapay Zeka Uygulamaları-I: Yapay Sinir Ağları, Ufuk Kitap Kırtasiye-Yayıncılık Tic. Ltd., 2003.

Saraç (2004) Saraç T., Yapay Sinir Ağları, Gazi Üniversitesi Endüstri Mühendisliği Bölümü, Seminer Projesi, 2004.

Saraoğlu (1998) Saraoğlu H.M., Beyin sinyallerinin geri yaymalı yapay sinir ağı ile eğitilmesi, haritalanması, Sakarya Üniversitesi, Fen Bilimleri Enstitüsü, Doktora, 1998.

Schaffer (1989) Schaffer J. D.(editor), Proceedings of the Third International Conference on Genetic Algorithms, San Mateo, Ca: Morgan Kaufmann Publishers Inc., 1989.

Schwarz (1978) Schwarz G., Estimating the Dimension of a Model, Annals of Statistics, vol. 6, no. 2, pp. 461-464, 1978.

Scott Fahlman (1988)

Senior (1997) Senior A., A Hidden Markov Model Fingerprint Classifier, in Proc. Asilomar Conf. on Signals Systems, Computers (3lst), pp. 306-310, 1997.

Senior (2001) Senior A., A Combination Fingerprint Classifier, IEEE Transactions on Pattern Analysis, Machine Intelligence, vol. 23, no. 10, pp. 1165-1174, 2001.

Setiono (2001) Setiono R., Feedforward Neural Network Construction Using Cross Validation, Neural computation, 13: 2865-2877, MIT Press, 2001.

Sha ve ark. (2003) Sha, L., Zhao, F., Tang, X., Improved fingercode for filterbank-based fingerprint matching, International Conference on Image Processing, , pt. 3, p II-895-8, vol.3, 2003.

Shepherd (1997) Shepherd A. J., Second-Order Methods for Neural Networks, Springer-Verlag New York, Inc., Secaucus, NJ, 1997.

Shewchuk (1994) Shewchuk J.R., An Introduction to the Conjugate Gradient Method Without the Agonizing Pain, Carnegie Mellon University, School of Computer Science, Pittsburgh, PA 15213, 1994.

Sibbald (1997) Sibbald A., Method, Apparatus for Fingerprint Characterization, Recognition Using Auto-correlation Pattern, US Patent 5633947, 1997.

Sinreich ve Samakh (1999) Sinriech D., Samakh E., A Genetic Approach to the Pickup/Delivery Station Location Problem in Segmented Flow Based Material Handling Systems, Journal of Manufacturing Systems, Volume: 18, Number: 2, s.81-99, 1999.

Soutar (2002) Soutar C, Biometric System Security, Secure — the Silicon Trust Magazine, no. 5, pp. 46-49, 2002.

Specht (1990) Specht, D. E., Probabilistic Neural Networks, Neural Networks, Vol. 3, pp. 109-118, 1990.

Steen (1996) Steen R. G., DNA, Destiny: Nature, Nurture in Human Behavior, New York: Plenum Press, 1996.

Stoney ve Thornton (1986) Stoney D.A., Thornton J.I., A Critical Analysis of Quantitative Fingerprints Individuality Models, Journal of Forensic Sciences, vol. 31, no. 4, pp. 1187-1216, 1986.

Şen (2001) Şen Z., Bulanık Mantık, Modelleme İlkeleri, Bilge Kültür Sanat, ISBN: 9758509233, 2001.

Şen (2004a) Şen Z., Genetik Algoritmalar, En İyileme Yöntemleri, Su Vakfı Yayınları, ISBN: 9756455128, 2004.

Şen (2004b) Şen Z., Yapay Sinir Ağları İlkeleri, Su Vakfı Yayınları, ISBN: 9756455136, 2004.

Tang ve ark. (1996) Tang K.S., Man K.F., Kwong S., He Q.H., Genetic algorithms, their applications, IEEE Signal Processing Magazine, vol.13, no.6, pp.22-37, Nov 1996.

Tank ve Hopfield (1987) Tank D. W., Hopfield J. J.. Collective computation in neuronlike circuits. Scientific American, 257(6): 104-114, 1987.

Tosun (1997) Tosun M., Geriyansımalı yapay sinir ağının görüntü sıkıştırmada kullanılması, Dumlupınar Üniversitesi, Fen Bilimleri Enstitüsü, Yüksek Lisans, 1997.

Trauring (1963) Trauring M., Automatic Comparison of Finger-Ridge Patterns, Nature, pp. 938-940, 1963.

Türkoğlu (2003) Türkoğlu İ., Örüntü Tanıma Ders Notları, Fırat Üniversitesi, 2003. (Erişim Tarihi: 10.10.2004)

Türkoğlu ve Arslan (1996) Türkoğlu, İ., Arslan, A., Yapay Sinir Ağları ile Bozuk Örüntü Tanıma, F.Ü. Fen, Mühendislik Bilimleri Dergisi, 8, , 147 – 158, 1996.

Ucal (2006) Ucal M.Ş., Ekonometrik Model Seçim Kriterleri Üzerine Kısa Bir İnceleme, C.Ü. İktisadi ve İdari Bilimler Dergisi, Cilt 7, Sayı 2, s. 41-57, 2006

Vargas (2001) http://www.biometricsview.com/doc/bio_eng.pdf (Erişim Tarihi: 2001)

Vatansever ve ark. (2002) Vatansever, F., Uysal, F., Uzun, A., Ayrık Dalgacık Dönüşümü ile Gürültü Süzme, ELECO2002, Bursa, s.338-342, 2002.

Vatansever ve ark. (2005) Vatansever F., Çankaya İ., Akgün D. , Dalgacık Tabanlı Görüntü Temizleme, 4. Uluslararası İleri Teknolojiler Sempozyumu, Konya, sf. 63-68, 2005.

Villasenor ve ark. (1995) Villasenor J.D., Belzer B., Liao J., Wavelet filter evaluation for image compression, IEEE Transactions on Image Processing, vol. 4/8, pp. 1053-1060, 1995.

Walker (1999) Walker, J.S., A Primer on Wavelets, their Scientific Applications, Chapman&Hall/CRC, 1999.

Walner (2002) Walner G., A biometric solution is ideal at the point of sale, In Biometric Technology Today, Vol. 10, No. 3, pp. 7-8, 2002.

Wei ve ark. (2004) Wei L., Pu-liu Y., De-ling X., Cong Z., 2004, An approach to dynamic fingerprint image enhancement, Third International Conference on Image, Graphics, Pp.: 294-297, 2004.

Weile ve Michielssen (1997) Weile D.S., Miclielssen E., Genetic Algorithm optimization applied to electromagnetics: A review, IEEE Trans. Antennas Propag., vol.45, pp.343-353, 1997.

Wentworth ve Wilder (1918) Wentworth B., Wilder H.H., Personal Identification, R.G. Badger, Boston, 1918.

Werbos (1974) Werbos P., Beyond Regression: New Tools for Prediction, Analysis in the Behavioral Sciences. PhD thesis, Harvard, Cambridge, MA, 1974.

Whitby (2005) Whitby B., Yapay Zeka, İletişim Yayınları, ISBN: 9750503791, 2005.

Wickerhauser (1994) Wickerhauser, M.V., Adapted Wavelet Analysis from Theory to Software, A.K. Peters Ltd., USA, 1994

Widrow-Hoff (1960) Widrow G., Hoff M.E., Adaptive switching circuits. Institute of radio engineers, Western Electronic show & convention, Convention record, Part 4 p96-104, 1960.

Wilson ve ark. (1997a) Wilson C. L., Blue J. L., Omidvar O. M., Training Dynamics, Neural Network Performance, Neural Networks, Vol. 10, No. 5, pp. 907-923, 1997.

Wilson ve ark. (1997b) Wilson C. L., Blue J. L., Omidwar O. M., Neurodynamics of Learning, Network Performance, Journal of Electronic Imaging, Vol. 6, No. 3, pp. 379-385, 1997.

Woodward (1999) Woodward J., Biometrics: Identifying Law, Policy Concerns, in Biometrics: Personal Identification in a Networked Society, A.K. Jain, R. Bolle, S. Pankanti (Eds.), Kluwer, New York, 1999.

Yager ve Amin (2004) Yager N., Amin A., Fingerprint classification: a review, Pattern Anal. Applic. vol. 7, pp. 77–93, 2004.

Yaman ve ark. (2001) Yaman K., Sarucan A., Atak M., Aktürk N., Dinamik Çizelgeleme için Görüntü İşleme ve ARIMA Modelleri Yardımıyla Veri Hazırlama, Gazi Üniversitesi Müh.Mim. Fak. Dergisi, Cilt. 16, Sayı 1, s. 19-40, 2001.

Yeniay (2001) Yeniay Ö., An Overview of Genetic Algorithms, Anadolu Üniversitesi Bilim, Teknoloji Dergisi, Cilt: 2, Sayı: 1, s. 37-49, 2001.

Yeo ve Agyel (1996) Yeo M. F., Agyel E. O., Optimising Engineering Problems Using Genetic Algorithms, Engineering Computations, Volume: 15, Number: 2, s. 268-280., 1996.

Yılmaz (2005) Yılmaz N., Yapay Zeka Yöntemleriyle Nesne Tanıma, Selçuk Üniversitesi, Fen Bilimleri Enstitüsü, Doktora Tezi, 2005.

Yurtoğlu (2005) Yurtoğlu H., Yapay Sinir Ağlari Metodolojisi İle Öngörü Modellemesi: Bazi Makroekonomik Değişkenler İçin Türkiye Örneği, DPT Uzmanlık Tezi, Yayın No: 2683, 2005.

Zhang ve ark. (2002) Zhang W.-P., Wang Q.-R., Tang Y.Y., A wavelet-based method for fingerprint image enhancement, International Conference on Machine Learning, Cybernetics, Volume 4, pp. 1973–1977, 2002.

EK. PARMAKİZİ TANIMA İÇİN ELDE EDİLEN SONUÇLAR

Bu Ek bölümünde, tez çalışmasına ait elde edilen sonuçlar detaylı olarak gösterilmektedir. Bu bölümde sonuçların gösterimi standart bir çizelge kullanılarak sunulmuştur. Böylece sonuçlar arasında karşılaştırmanın kolay bir şekilde gerçekleştirilmesi amaçlanmıştır. Parmakizi tabanlı biyometrik tanıma sistemleri üzerine odaklı bu çalışmada aşağıdaki işlem adımları gerçekleştirilmiş ve sonuçlar alınmıştır:

- Parmakizi görüntülerinin ön iyileştirilmesi,
    - Lineer filtre, Uyarlamalı filtre, Medyan filtre
    - Dalgacık dönüşümü, Çevritsel dönüşüm
- Parmakizi özellik vektörlerinin elde edilmesi,
    - FingerCode
- Özellik vektörü boyutunun azaltılması
    - Genetik Algoritmalar
- Normal boyutlu ve azaltılmış özellik vektörleri ile parmakizi tanıma,
    - Yapay Sinir Ağları

Parmakizi görüntülerinin iyileştirilme aşamasında geleneksel yöntemlerin yanı sıra güncel dönüşüm yöntemleri de kullanılarak bunlar arasında görüntü iyileştirme performansı olarak kıyaslama yapılmıştır. Bu aşamada en iyi görüntü iyileştirme başarımı, dalgacık dönüşümün gelişmişi olan çevritsel dönüşüm ile elde edilmiştir. Sinyal-gürültü oranı dikkate alınarak gerçekleştirilen parmakizi iyileştirme işlemlerinde en yüksek sinyal-gürültü oranına çevritsel dönüşüm ile ulaşılmıştır. Bu bölümde kullanılan görüntü iyileştirme yöntemi belirtilmiştir.

Parmakizi tanıma aşamasında kullanılmak üzere iyileştirilen parmakizi görüntülerinden özellik vektörleri elde edilmelidir. Bunun için daha önce literatürde kullanılan FingerCode yöntemi geliştirilerek özellik vektörleri elde edilmiştir. Ayrıca parmakizinin yönsel histogramına göre elde edilen referans noktasına ait açısal bilginin, parmakizinin özellik vektörünün başlangıç referansı olarak alınması bu çalışmada orijinaldir. Bu, özellik vektörlerinin Yapay Sinir Ağları ile parmakizi tanımaya uyumu için gereklidir.

Parmakizi tanıma aşamasında giriş sayısı olan normal özellik vektörü boyutu ve Genetik Algoritmalar ile azaltılmış özellik vektörü boyutuna uygun gizli katman düğüm sayısı belirlenmiştir. Bu bölümde özellik vektörü boyutuna göre elde edilen en uygun Yapay Sinir Ağı (YSA) yapıları gösterilmiştir. Yine parmakizi tanıma aşamasında YSA'na ait çeşitli algoritmalar kullanılmıştır. Kullanılan algoritmalar bu bölümde belirtilmiştir. YSA ile özellik vektörlerinin eğitilmesi sonucunda iterasyon sayısına göre ağın doğru sınıflandırma oranı grafik ile gösterilmiştir. Tüm YSA algoritmaları için eğitim kümesi, doğrulama kümesi, test kümesi ve tüm veri kümelerine ait test sonuçlan eşleştirme matrisi şeklinde gösterilmiştir. Eşleştirme matrisi, gerçek çıkış bilgisiyle ağın eğitilmesi sonucu elde edilen çıkış değerinin karşılaştırıldığı matristir. Matrisin diyagonal ekseni doğru eşleştirilen parmakizi sayılarını göstermektedir. Diyagonal eksen haricindeki sayılar ise hatalı olarak farklı bir kişiye aitmiş gibi eşleştirilen parmakizi sayılarını ifade etmektedir.

Tüm bu sonuçlar detaylı bir şema içerisinde gösterilmiştir. Bu şema aşağıdaki temel kısımlardan oluşmaktadır:

- Ön işleme yöntemi
- Band ve sektör sayısı
- YSA metodu, yapısı ve eğitim süresi
- YSA parametreleri, eğitim ve doğrulama sonuçları
- YSA eğitim ve doğrulama veri kümesi doğru sınıflandırma grafiği
- Eğitim, doğrulama, test ve tüm veri kümelerine ait doğru sınıflandırma oranları ve eşleştirme matrisleri

Şema ve kısımların açıklaması aşağıdaki şekildedir.

<table>
<tr><td>Ön işleme yöntemi:

Parmakizi görüntülerini iyileştirmek için kullanılan yöntem</td><td>Band ve sektör sayısı:

Özellik vektörüne ait band ve sektör sayıları</td><td>YSA eğitim süresi:

Parmakizi tanıma için YSA’nın toplam eğitim süresi</td></tr>
<tr><td colspan="3">YSA metodu ve yapısı: Eğitmede kullanılan YSA algoritması ve YSA’nın giriş katmanı, gizli katmanı ve çıkış katmanı düğüm sayılarından oluşan YSA yapısı</td></tr>
<tr><td colspan="2">YSA parametreleri:
Eğitim ve doğrulama sonuçları:</td><td>YSA ile eğitme ve doğrulama veri kümelerine ait doğru sınıflandırma oranları, ağın hata oranları, iterasyon sayısı, eğitim hızı (saniyedeki iterasyon sayısına göre), YSA yapısı, YSA eğitim algoritması</td></tr>
<tr><td colspan="3">YSA veri kümesi hata grafiği

YSA’nın eğitimi sırasında iterasyon sayısına göre eğitim veri kümesi ve doğrulama veri kümesine ait YSA’nın doğru sınıflandırma grafiği
Her iki veri kümesi için de %100 ve %100’e yakın değerler olmak üzere YSA’nın başarılı bir eğitme yaptığını göstermektedir.</td></tr>
<tr><td colspan="3">Ortalama doğruluk oranları (%)</td></tr>
<tr><td colspan="2">Eğitim veri kümesi:</td><td>Doğrulama veri kümesi:</td></tr>
<tr><td colspan="2">YSA eğitimi sonucunda ağın elde ettiği çıkışlar ile eğitim veri kümesine ait gerçek çıkışların karşılaştırıldığı eşleştirme matrisi ve doğru sınıflandırma oranı</td><td>YSA eğitimi sonucunda ağın elde ettiği çıkışlar ile doğrulama veri kümesine ait gerçek çıkışların karşılaştırıldığı eşleştirme matrisi ve doğru sınıflandırma oranı</td></tr>
<tr><td colspan="2">Test veri kümesi:</td><td>Tüm veri kümesi:</td></tr>
<tr><td colspan="2">YSA eğitimi sonucunda ağın elde ettiği çıkışlar ile test veri kümesine ait gerçek çıkışların karşılaştırıldığı eşleştirme matrisi ve doğru sınıflandırma oranı</td><td>YSA eğitimi sonucunda ağın elde ettiği çıkışlar ile tüm veri kümesine ait gerçek çıkışların karşılaştırıldığı eşleştirme matrisi ve doğru sınıflandırma oranı</td></tr>
</table>

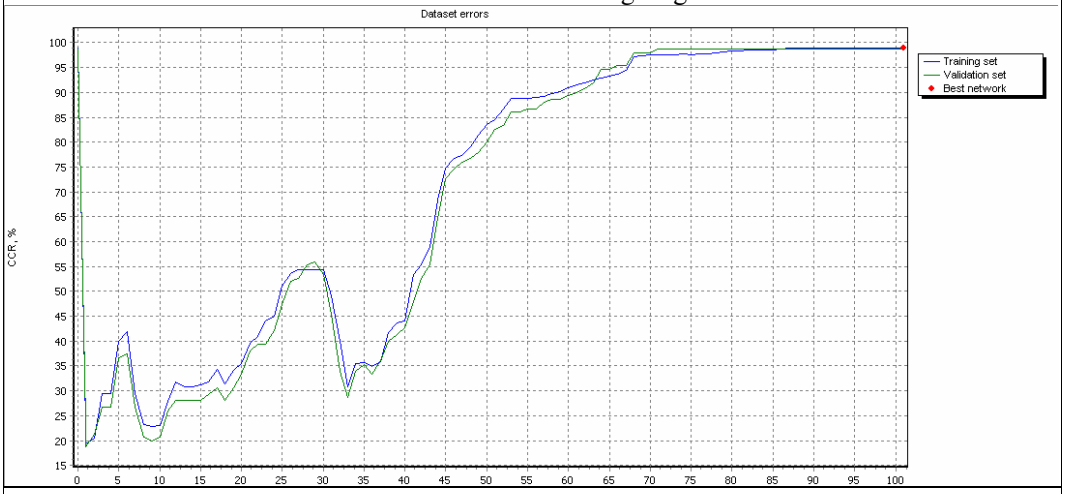
Ön işleme yöntemi: *Çevritsel dönüşüm*	Band ve sektör sayısı: 8 x 24	YSA eğitim süresi: 00:58:55

YSA metodu ve yapısı: *Batch Back Propagation* 1536:236:10

YSA parametreleri:
Eğitim ve doğrulama sonuçları:

	Training	Validation
CCR, %:	99	98,666667
Network error:	0,001679	0
Error improvement:	0,000037	
Iteration:	101	
Training speed, iter/sec:	0,02892	
Architecture:	[1536-236-10]	
Training algorithm:	Batch Back Propagation	
Training stop reason:	All iterations done	

YSA veri kümesi hata grafiği



Ortalama doğruluk oranları (%)

Eğitim veri kümesi: 99

output:	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10
S1	70	0	0	0	0	0	0	0	0	0
S2	0	72	0	0	0	0	0	0	0	0
S3	0	0	71	0	0	0	0	0	0	0
S4	0	0	0	71	0	0	0	0	0	1
S5	0	0	0	0	71	0	0	0	0	0
S6	0	0	1	0	0	68	0	0	1	1
S7	0	0	0	0	0	0	71	0	0	1
S8	0	0	0	0	0	0	0	67	0	0
S9	0	0	0	0	1	0	0	0	65	0
S10	0	0	0	0	1	0	0	0	0	67

Doğrulama veri kümesi: 98,66667

output:	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10
S1	15	0	0	0	0	0	0	1	0	0
S2	0	13	0	0	0	0	0	0	0	0
S3	0	0	15	0	0	0	0	0	0	0
S4	0	0	0	14	0	0	0	0	0	0
S5	0	0	0	0	15	0	0	0	0	0
S6	0	0	0	0	0	14	0	0	0	0
S7	0	0	0	0	0	0	14	0	0	0
S8	0	0	0	0	0	0	0	16	0	0
S9	0	0	0	0	0	0	0	0	18	0
S10	0	0	0	0	1	0	0	0	0	14

Test veri kümesi: 97,333333

output:	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10
S1	14	0	0	0	0	0	0	0	0	0
S2	0	15	0	0	0	0	0	0	0	0
S3	0	0	14	0	0	0	0	0	0	0
S4	0	0	0	13	0	0	0	0	0	1
S5	0	0	0	0	14	0	0	0	0	0
S6	0	0	0	0	0	14	1	0	0	0
S7	0	0	0	0	0	0	14	0	0	0
S8	0	0	0	0	0	0	0	17	0	0
S9	0	0	0	0	0	0	0	0	15	1
S10	0	0	0	1	0	0	0	0	0	16

Tüm veri kümesi: 98,7

output:	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10
S1	99	0	0	0	0	0	0	1	0	0
S2	0	100	0	0	0	0	0	0	0	0
S3	0	0	100	0	0	0	0	0	0	0
S4	0	0	0	98	0	0	0	0	0	2
S5	0	0	0	0	100	0	0	0	0	0
S6	0	0	1	0	0	96	1	0	1	1
S7	0	0	0	0	0	0	99	0	0	1
S8	0	0	0	0	0	0	0	100	0	0
S9	0	0	0	0	1	0	0	0	98	1
S10	0	0	0	1	2	0	0	0	0	97

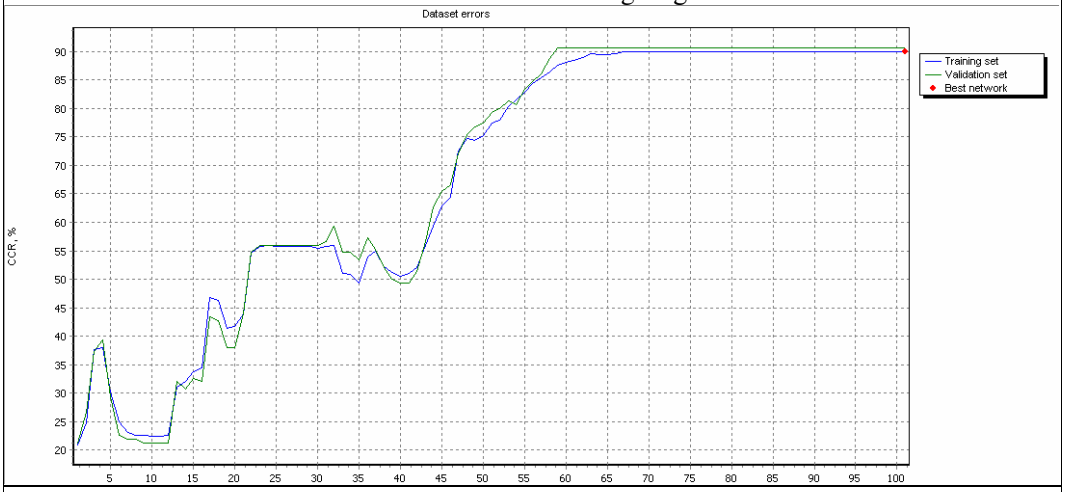
Ön işleme yöntemi: *Çevritsel dönüşüm*	Band ve sektör sayısı: 8 x 20	YSA eğitim süresi: 00:17:40

YSA metodu ve yapısı: *Batch Back Propagation* 1280:192:10

YSA parametreleri:
Eğitim ve doğrulama sonuçları:

	Training	Validation
CCR, %:	90	90,666667
Network error:	0,021028	0
Error improvement:	0,000009	
Iteration:	101	
Training speed, iter/sec:	0,096463	
Architecture:	[1280-192-10]	
Training algorithm:	Batch Back Propagation	
Training stop reason:	All iterations done	

YSA veri kümesi hata grafiği



Ortalama doğruluk oranları (%)

Eğitim veri kümesi: 90

output:	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10
S1	66	0	1	0	4	0	0	0	0	0
S2	0	71	0	0	0	0	0	0	0	0
S3	0	0	72	0	0	0	0	0	0	0
S4	0	0	0	71	0	0	0	0	0	1
S5	3	0	0	0	54	0	3	3	8	0
S6	0	0	0	0	1	68	2	0	0	0
S7	2	0	0	0	0	1	66	1	0	2
S8	0	0	0	0	5	0	0	62	0	0
S9	0	0	0	3	0	0	0	1	62	0
S10	2	0	0	8	2	0	0	0	17	38

Doğrulama veri kümesi: 96,66667

output:	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10
S1	15	0	0	0	0	0	0	0	0	0
S2	0	14	0	0	0	0	0	0	0	0
S3	0	0	14	0	0	0	0	0	0	0
S4	0	0	0	13	0	0	0	1	0	0
S5	1	0	0	0	12	0	0	0	1	0
S6	0	0	0	0	0	15	0	0	0	0
S7	0	0	0	0	0	0	13	0	0	1
S8	0	0	0	0	1	0	0	15	0	0
S9	0	0	0	2	0	0	0	0	14	1
S10	1	0	0	1	0	0	0	0	4	11

Test veri kümesi: 90

output:	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10
S1	13	0	1	0	0	0	0	0	0	0
S2	0	15	0	0	0	0	0	0	0	0
S3	0	0	14	0	0	0	0	0	0	0
S4	0	0	0	14	0	0	0	0	0	0
S5	1	0	0	0	12	0	1	0	1	0
S6	0	0	0	0	0	13	0	0	0	1
S7	2	0	0	0	0	0	12	0	0	0
S8	0	0	0	0	1	0	0	16	0	0
S9	0	0	0	1	0	0	1	0	15	0
S10	0	0	0	1	0	0	0	0	4	11

Tüm veri kümesi: 90,1

output:	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10
S1	94	0	2	0	4	0	0	0	0	0
S2	0	100	0	0	0	0	0	0	0	0
S3	0	0	100	0	0	0	0	0	0	0
S4	0	0	0	98	0	0	0	1	0	1
S5	5	0	0	0	78	0	4	3	10	0
S6	0	0	0	0	1	96	2	0	0	1
S7	4	0	0	0	0	1	91	1	0	3
S8	0	0	0	0	7	0	0	93	0	0
S9	0	0	0	6	0	0	1	1	91	1
S10	3	0	0	10	2	0	0	0	25	60

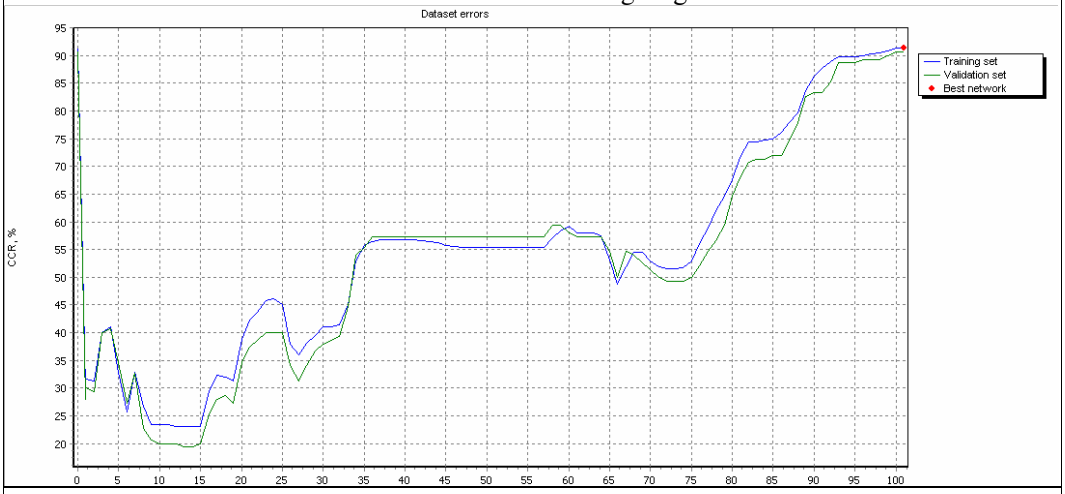
Ön işleme yöntemi: *Çevritsel dönüşüm*	Band ve sektör sayısı: 7 x 24	YSA eğitim süresi: 01:21:23

YSA metodu ve yapısı: *Batch Back Propagation* 1341:224:10

YSA parametreleri:
Eğitim ve doğrulama sonuçları:

	Training	Validation
CCR, %:	91,428571	90,666667
Network error:	0,022533	0
Error improvement:	0,001659	
Iteration:	101	
Training speed, iter/sec:	0,020833	
Architecture:	[1341-224-10]	
Training algorithm:	Batch Back Propagation	
Training stop reason:	All iterations done	

YSA veri kümesi hata grafiği



Ortalama doğruluk oranları (%)

Eğitim veri kümesi: 91,428571

output:	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10
S1	67	0	0	0	3	0	0	1	0	0
S2	0	72	0	0	0	0	0	0	0	0
S3	0	0	72	0	0	0	0	0	0	0
S4	1	0	0	69	0	0	0	1	0	0
S5	2	0	0	0	57	0	2	2	9	0
S6	0	0	0	0	0	69	0	1	0	0
S7	5	0	0	0	0	1	63	3	0	0
S8	0	0	0	0	0	0	0	67	0	0
S9	0	0	0	0	0	0	0	1	64	1
S10	3	0	0	2	1	0	0	0	21	40

Doğrulama veri kümesi: 90,66667

output:	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10
S1	15	0	0	0	0	0	0	0	0	0
S2	0	13	0	0	0	0	0	0	0	0
S3	0	0	15	0	0	0	0	0	0	0
S4	0	0	0	14	0	0	0	0	0	0
S5	2	0	0	0	9	0	0	0	3	0
S6	0	0	0	0	0	13	2	0	0	0
S7	0	0	0	0	0	0	14	0	0	0
S8	0	0	0	0	0	0	0	16	0	0
S9	0	0	0	1	0	0	0	0	16	0
S10	0	0	0	1	0	0	0	0	5	11

Test veri kümesi: 90,66667

output:	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10
S1	13	0	1	0	0	0	0	0	0	0
S2	0	15	0	0	0	0	0	0	0	0
S3	0	0	13	0	0	0	0	0	0	0
S4	0	0	0	15	0	0	0	0	0	0
S5	1	0	0	0	11	0	0	1	1	0
S6	0	0	0	0	0	14	0	0	0	1
S7	0	0	0	0	0	0	14	0	0	0
S8	0	0	0	0	1	0	0	16	0	0
S9	0	0	0	0	0	0	1	0	16	0
S10	1	0	0	0	1	0	0	0	5	9

Tüm veri kümesi: 91,2

output:	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10
S1	95	0	1	0	3	0	0	1	0	0
S2	0	100	0	0	0	0	0	0	0	0
S3	0	0	100	0	0	0	0	0	0	0
S4	1	0	0	98	0	0	0	1	0	0
S5	5	0	0	0	77	0	2	3	13	0
S6	0	0	0	0	0	96	2	1	0	1
S7	5	0	0	0	0	1	91	3	0	0
S8	0	0	0	0	1	0	0	99	0	0
S9	0	0	0	1	0	0	1	1	96	1
S10	4	0	0	3	2	0	0	0	31	60

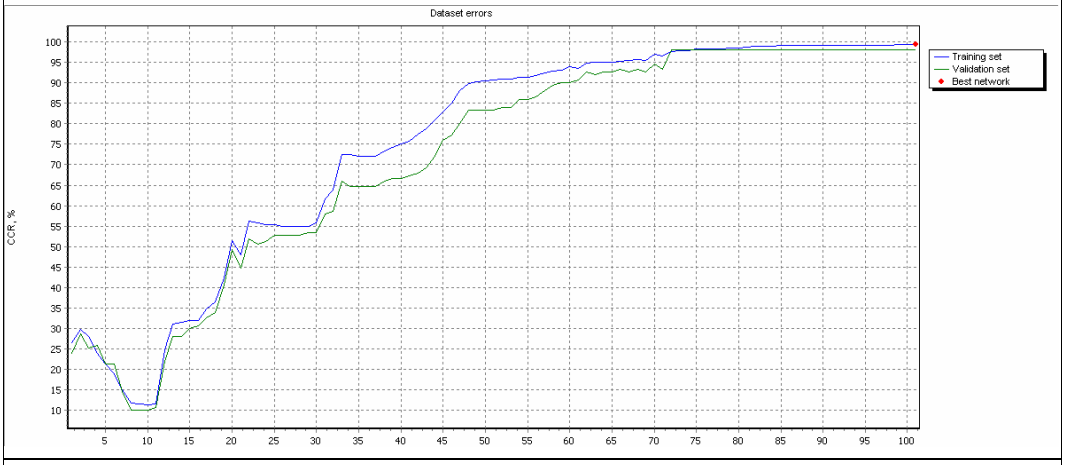
Ön işleme yöntemi: *Çevritsel dönüşüm*	Band ve sektör sayısı: 7 x 20	YSA eğitim süresi: 00:44:38

YSA metodu ve yapısı: *Batch Back Propagation* 1118:168:10

YSA parametreleri:
Eğitim ve doğrulama sonuçları:

	Training	Validation
CCR, %:	99,428571	98
Network error:	0,001508	0
Error improvement:	0,000043	
Iteration:	101	
Training speed, iter/sec:	0,038049	
Architecture:	[1118-168-10]	
Training algorithm:	Batch Back Propagation	
Training stop reason:	All iterations done	

YSA veri kümesi hata grafiği



Ortalama doğruluk oranları (%)

Eğitim veri kümesi: 99,428571

output:	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10
S1	71	0	0	0	0	0	0	0	0	0
S2	0	73	0	0	0	0	0	0	0	0
S3	0	0	70	0	0	0	0	0	0	0
S4	0	0	0	72	0	0	0	0	0	0
S5	0	0	0	0	71	0	0	0	0	0
S6	0	0	0	0	0	67	2	0	0	2
S7	0	0	0	0	0	0	72	0	0	0
S8	0	0	0	0	0	0	0	67	0	0
S9	0	0	0	0	0	0	0	0	67	0
S10	0	0	0	0	0	0	0	0	0	66

Doğrulama veri kümesi: 98

output:	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10
S1	15	0	0	0	0	0	0	0	0	0
S2	0	13	0	0	0	0	0	0	0	0
S3	0	0	15	0	0	0	0	0	0	0
S4	0	0	0	13	0	0	0	0	0	1
S5	0	0	0	0	14	0	0	0	1	0
S6	0	0	0	0	0	14	0	0	0	0
S7	0	0	0	0	0	0	14	0	0	0
S8	0	0	0	0	0	0	0	17	0	0
S9	0	0	0	0	0	0	0	0	15	1
S10	0	0	0	0	0	0	0	0	0	17

Test veri kümesi: 96,66667

output:	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10
S1	13	0	0	0	1	0	0	0	0	0
S2	0	14	0	0	0	0	0	0	0	0
S3	0	0	15	0	0	0	0	0	0	0
S4	0	0	0	13	0	0	0	0	0	1
S5	0	0	0	0	14	0	0	0	0	0
S6	0	0	0	0	0	15	0	0	0	0
S7	0	0	0	0	0	0	12	0	0	2
S8	0	0	0	0	0	0	0	16	0	0
S9	0	0	0	0	1	0	0	0	16	0
S10	0	0	0	0	0	0	0	0	0	17

Tüm veri kümesi: 98,8

output:	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10
S1	99	0	0	0	1	0	0	0	0	0
S2	0	100	0	0	0	0	0	0	0	0
S3	0	0	100	0	0	0	0	0	0	0
S4	0	0	0	98	0	0	0	0	0	2
S5	0	0	0	0	99	0	0	0	1	0
S6	0	0	0	0	0	96	2	0	0	2
S7	0	0	0	0	0	0	98	0	0	2
S8	0	0	0	0	0	0	0	100	0	0
S9	0	0	0	0	1	0	0	0	98	1
S10	0	0	0	0	0	0	0	0	0	100

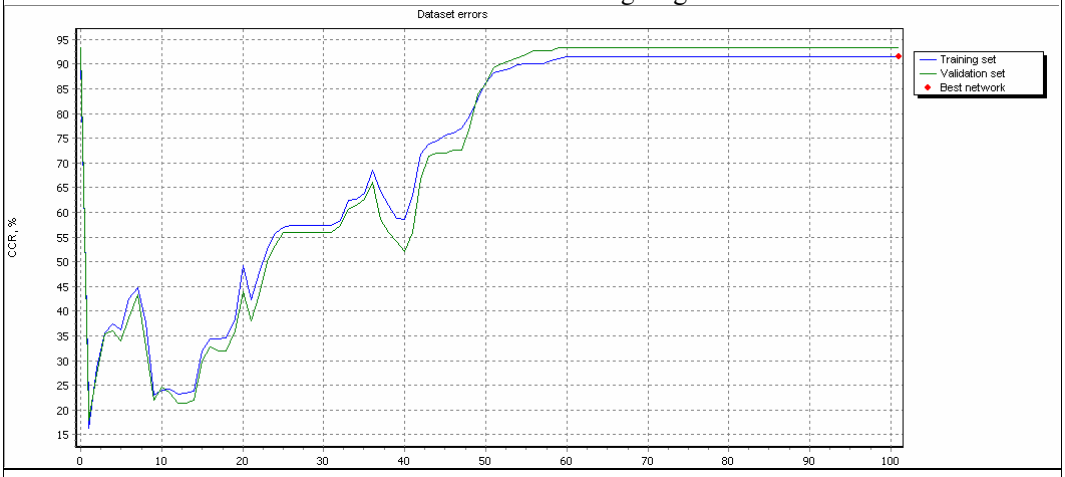
Ön işleme yöntemi: *Çevritsel dönüşüm*	Band ve sektör sayısı: 7 x 16	YSA eğitim süresi: 00:50:29

YSA metodu ve yapısı: *Batch Back Propagation* 894:134:10

YSA parametreleri:
Eğitim ve doğrulama sonuçları:

	Training	Validation
CCR, %:	91,428571	93,333333
Network error:	0,018936	0
Error improvement:	0,000008	
Iteration:	101	
Training speed, iter/sec:	0,033531	
Architecture:	[894-134-10]	
Training algorithm:	Batch Back Propagation	
Training stop reason:	All iterations done	

YSA veri kümesi hata grafiği



Ortalama doğruluk oranları (%)

Eğitim veri kümesi: 91,428571

output:	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10
S1	67	0	0	0	3	0	0	1	0	0
S2	0	73	0	0	0	0	0	0	0	0
S3	0	0	70	0	0	0	0	0	0	0
S4	0	0	0	70	0	0	0	1	0	1
S5	3	0	0	0	55	0	1	2	9	0
S6	0	0	0	0	0	71	1	0	0	1
S7	3	0	0	0	0	1	63	1	0	3
S8	0	0	0	0	1	0	0	65	0	0
S9	0	0	0	0	0	0	0	0	67	1
S10	2	0	0	10	1	0	0	0	14	39

Doğrulama veri kümesi: 96

output:	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10
S1	15	0	0	0	0	0	0	0	0	0
S2	0	13	0	0	0	0	0	0	0	0
S3	0	0	15	0	0	0	0	0	0	0
S4	0	0	0	14	0	0	0	0	0	0
S5	1	0	0	0	13	0	0	0	1	0
S6	0	0	0	0	0	13	0	0	0	0
S7	0	0	0	0	0	0	15	0	0	0
S8	0	0	0	0	0	0	0	17	0	0
S9	0	0	0	0	0	0	0	1	15	0
S10	0	0	0	4	0	0	0	0	3	10

Test veri kümesi: 91,33333

output:	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10
S1	14	0	0	0	0	0	0	0	0	0
S2	0	14	0	0	0	0	0	0	0	0
S3	0	0	15	0	0	0	0	0	0	0
S4	0	0	0	14	0	0	0	0	0	0
S5	1	0	0	0	11	0	0	1	2	0
S6	0	0	0	0	0	12	1	1	0	0
S7	1	0	0	0	0	0	13	0	0	0
S8	0	0	0	0	0	0	0	17	0	0
S9	0	0	0	1	0	0	0	0	15	0
S10	2	0	0	1	0	0	0	0	2	12

Tüm veri kümesi: 91,7

output:	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10
S1	96	0	0	0	3	0	0	1	0	0
S2	0	100	0	0	0	0	0	0	0	0
S3	0	0	100	0	0	0	0	0	0	0
S4	0	0	0	98	0	0	0	1	0	1
S5	5	0	0	0	79	0	1	3	12	0
S6	0	0	0	0	0	96	2	1	0	1
S7	4	0	0	0	0	1	91	1	0	3
S8	0	0	0	0	1	0	0	99	0	0
S9	0	0	0	1	0	0	0	1	97	1
S10	4	0	0	15	1	0	0	0	19	61

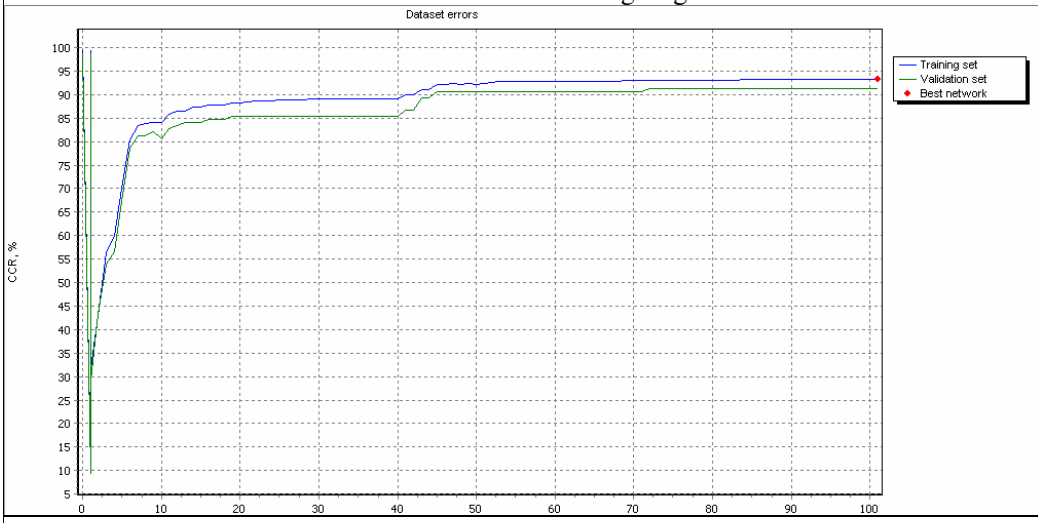
Ön işleme yöntemi: *Çevritsel dönüşüm*	Band ve sektör sayısı: 6 x 12	YSA eğitim süresi: 02:06:02

YSA metodu ve yapısı: *Conjugate Gradient Descent* 576:86:10

YSA parametreleri:
Eğitim ve doğrulama sonuçları:

	Training	Validation
CCR, %:	93,142857	91,333333
Network error:	0,006857	0
Error improvement:	2,20E-09	
Iteration:	101	
Training speed, iter/sec:	0,013369	
Architecture:	[576-86-10]	
Training algorithm:	Conjugate Gradient Descent	
Training stop reason:	All iterations done	

YSA veri kümesi hata grafiği



Ortalama doğruluk oranları (%)

Eğitim veri kümesi: 99,428571

output:	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10
S1	71	0	0	0	0	0	0	0	0	0
S2	0	72	0	0	0	0	0	0	0	0
S3	0	0	71	0	0	0	0	0	0	0
S4	0	0	0	72	0	0	0	0	0	0
S5	0	0	0	0	71	0	0	0	0	0
S6	0	1	0	1	0	69	0	0	0	0
S7	0	0	0	0	0	0	71	1	0	0
S8	0	0	0	0	0	0	0	66	0	0
S9	0	0	0	0	0	0	0	0	66	1
S10	0	0	0	0	0	0	0	0	0	67

Doğrulama veri kümesi: 98,666667

output:	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10
S1	14	0	0	0	0	0	0	0	0	0
S2	0	15	0	0	0	0	0	0	0	0
S3	0	0	14	0	0	0	0	0	0	0
S4	0	0	0	14	0	0	0	0	0	0
S5	0	0	0	0	15	0	0	0	0	0
S6	0	0	0	0	0	13	0	0	1	0
S7	0	0	0	0	0	0	14	0	0	0
S8	0	0	0	0	0	0	0	17	0	0
S9	0	0	0	0	0	0	0	0	16	0
S10	1	0	0	0	0	0	0	0	0	16

Test veri kümesi: 98,666667

output:	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10
S1	15	0	0	0	0	0	0	0	0	0
S2	0	13	0	0	0	0	0	0	0	0
S3	0	0	15	0	0	0	0	0	0	0
S4	0	0	0	14	0	0	0	0	0	0
S5	0	0	0	0	14	0	0	0	0	0
S6	0	0	0	0	0	14	0	0	0	1
S7	0	0	0	0	0	0	14	0	0	0
S8	0	0	0	0	0	0	0	17	0	0
S9	0	0	0	0	0	0	0	0	16	1
S10	0	0	0	0	0	0	0	0	0	16

Tüm veri kümesi: 99,2

output:	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10
S1	100	0	0	0	0	0	0	0	0	0
S2	0	100	0	0	0	0	0	0	0	0
S3	0	0	100	0	0	0	0	0	0	0
S4	0	0	0	100	0	0	0	0	0	0
S5	0	0	0	0	100	0	0	0	0	0
S6	0	1	0	1	0	96	0	0	1	1
S7	0	0	0	0	0	0	99	1	0	0
S8	0	0	0	0	0	0	0	100	0	0
S9	0	0	0	0	0	0	0	0	98	2
S10	1	0	0	0	0	0	0	0	0	99

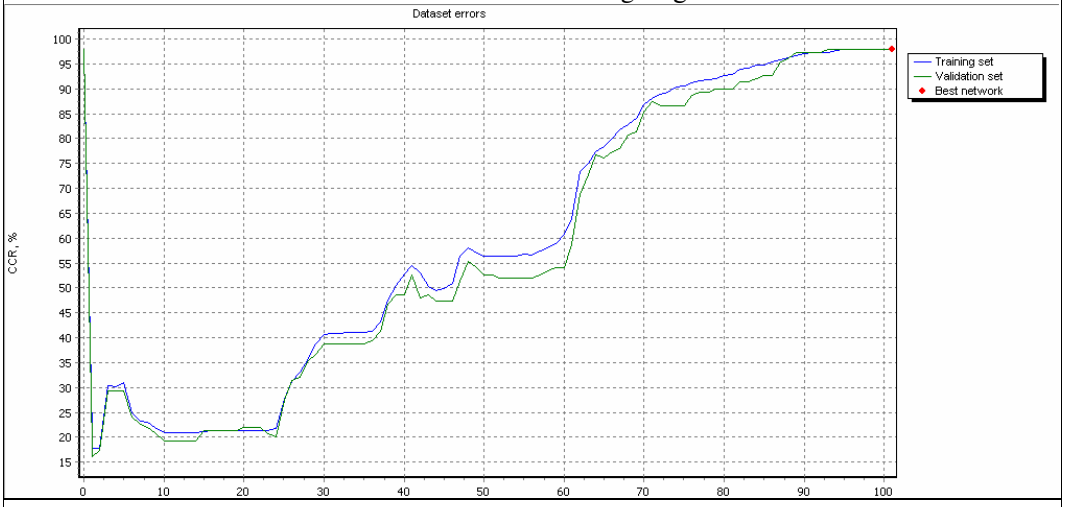
Ön işleme yöntemi: *Çevritsel dönüşüm*	Band ve sektör sayısı: 6 x 12	YSA eğitim süresi: 00:15:58

YSA metodu ve yapısı: *Batch Back Propagation* 576:86:10

YSA parametreleri:
Eğitim ve doğrulama sonuçları:

	Training	Validation
CCR, %:	97,857143	98
Network error:	0,005302	0
Error improvement:	0,000233	
Iteration:	101	
Training speed, iter/sec:	0,106066	
Architecture:	[576-86-10]	
Training algorithm:	Batch Back Propagation	
Training stop reason:	All iterations done	

YSA veri kümesi hata grafiği



Ortalama doğruluk oranları (%)

Eğitim veri kümesi: 98

output:	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10
S1	69	0	0	0	2	0	0	0	0	0
S2	0	72	0	0	0	0	0	0	0	0
S3	0	0	71	0	0	0	0	0	0	0
S4	0	0	0	70	0	0	0	0	0	2
S5	0	0	0	0	70	0	0	1	0	0
S6	0	0	0	1	0	69	1	0	0	0
S7	0	0	0	0	0	0	69	1	0	2
S8	0	0	0	0	0	0	0	66	0	0
S9	0	0	0	0	0	0	0	0	66	1
S10	1	0	0	0	2	0	0	0	0	64

Doğrulama veri kümesi: 98

output:	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10
S1	13	0	0	0	0	0	0	0	0	1
S2	0	15	0	0	0	0	0	0	0	0
S3	0	0	14	0	0	0	0	0	0	0
S4	0	0	0	14	0	0	0	0	0	0
S5	0	0	0	0	15	0	0	0	0	0
S6	0	0	0	0	0	13	1	0	0	0
S7	0	0	0	0	0	0	14	0	0	0
S8	0	0	0	0	0	0	0	17	0	0
S9	0	0	0	0	0	0	0	0	16	0
S10	1	0	0	0	0	0	0	0	0	16

Test veri kümesi: 98,666667

output:	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10
S1	15	0	0	0	0	0	0	0	0	0
S2	0	13	0	0	0	0	0	0	0	0
S3	0	0	15	0	0	0	0	0	0	0
S4	0	0	0	14	0	0	0	0	0	0
S5	0	0	0	0	14	0	0	0	0	0
S6	0	0	0	0	0	14	0	0	0	1
S7	0	0	0	0	0	0	14	0	0	0
S8	0	0	0	0	0	0	0	17	0	0
S9	0	0	0	0	1	0	0	0	16	0
S10	0	0	0	0	0	0	0	0	0	16

Tüm veri kümesi: 98,1

output:	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10
S1	97	0	0	0	2	0	0	0	0	1
S2	0	100	0	0	0	0	0	0	0	0
S3	0	0	100	0	0	0	0	0	0	0
S4	0	0	0	98	0	0	0	0	0	2
S5	0	0	0	0	99	0	0	1	0	0
S6	0	0	0	1	0	96	2	0	0	1
S7	0	0	0	0	0	0	97	1	0	2
S8	0	0	0	0	0	0	0	100	0	0
S9	0	0	0	0	1	0	0	0	98	1
S10	2	0	0	0	2	0	0	0	0	96

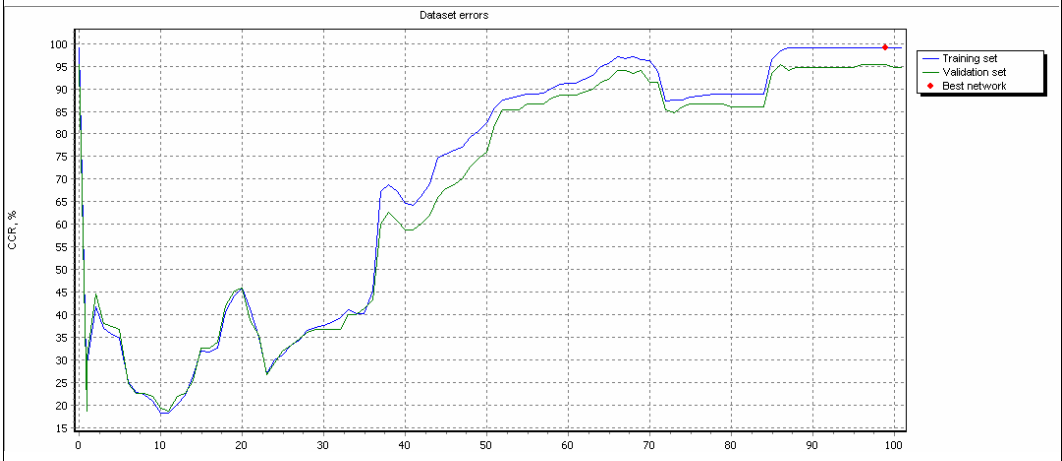
Ön işleme yöntemi: *Çevritsel dönüşüm*	Band ve sektör sayısı: 6 x 16	YSA eğitim süresi: 00:43:26

YSA metodu ve yapısı: *Batch Back Propagation* 768:116:10

YSA parametreleri:
Eğitim ve doğrulama sonuçları:

	Training	Validation
CCR, %:	99,142857	94,666667
Network error:	0,001795	0
Error improvement:	0,00005	
Iteration:	101	
Training speed, iter/sec:	0,038956	
Architecture:	[768-116-10]	
Training algorithm:	Batch Back Propagation	
Training stop reason:	All iterations done	

YSA veri kümesi hata grafiği



Ortalama doğruluk oranları (%)

Eğitim veri kümesi: 99,142857

output:	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10
S1	72	0	0	0	0	0	0	0	0	0
S2	0	70	0	0	0	0	0	0	0	0
S3	0	0	73	0	0	0	0	0	0	0
S4	0	0	0	71	0	0	0	0	0	0
S5	1	0	0	0	70	0	0	0	0	0
S6	0	0	0	1	0	69	1	0	0	0
S7	0	0	0	0	0	0	70	1	0	1
S8	0	0	0	0	0	0	0	66	0	0
S9	0	0	0	0	0	0	0	0	65	1
S10	0	0	0	0	0	0	0	0	0	68

Doğrulama veri kümesi: 95,33333

output:	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10
S1	14	0	0	0	0	0	0	0	0	0
S2	0	15	0	0	0	0	0	0	0	0
S3	0	0	13	0	0	0	0	0	0	0
S4	0	0	0	14	1	0	0	0	0	0
S5	0	0	0	0	14	0	0	0	0	0
S6	0	0	0	0	0	14	0	0	0	1
S7	0	0	0	1	0	0	13	0	0	0
S8	0	0	0	0	0	0	0	17	0	0
S9	0	0	0	0	1	0	0	0	16	0
S10	1	0	0	1	1	0	0	0	0	13

Test veri kümesi: 98,666667

output:	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10
S1	14	0	0	0	0	0	0	0	0	0
S2	0	15	0	0	0	0	0	0	0	0
S3	0	0	14	0	0	0	0	0	0	0
S4	0	0	0	13	0	0	0	0	0	1
S5	0	0	0	0	14	0	0	1	0	0
S6	0	0	1	0	0	13	0	0	0	0
S7	0	0	0	0	0	1	13	0	0	0
S8	0	0	0	0	0	0	0	17	0	0
S9	0	0	0	0	0	0	0	0	17	0
S10	0	0	0	1	0	0	0	0	0	15

Tüm veri kümesi: 98,2

output:	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10
S1	100	0	0	0	0	0	0	0	0	0
S2	0	100	0	0	0	0	0	0	0	0
S3	0	0	100	0	0	0	0	0	0	0
S4	0	0	0	98	1	0	0	0	0	1
S5	1	0	0	0	98	0	0	1	0	0
S6	0	0	1	1	0	96	1	0	0	1
S7	0	0	0	1	0	1	96	1	0	1
S8	0	0	0	0	0	0	0	100	0	0
S9	0	0	0	0	1	0	0	0	98	1
S10	1	0	0	2	1	0	0	0	0	96

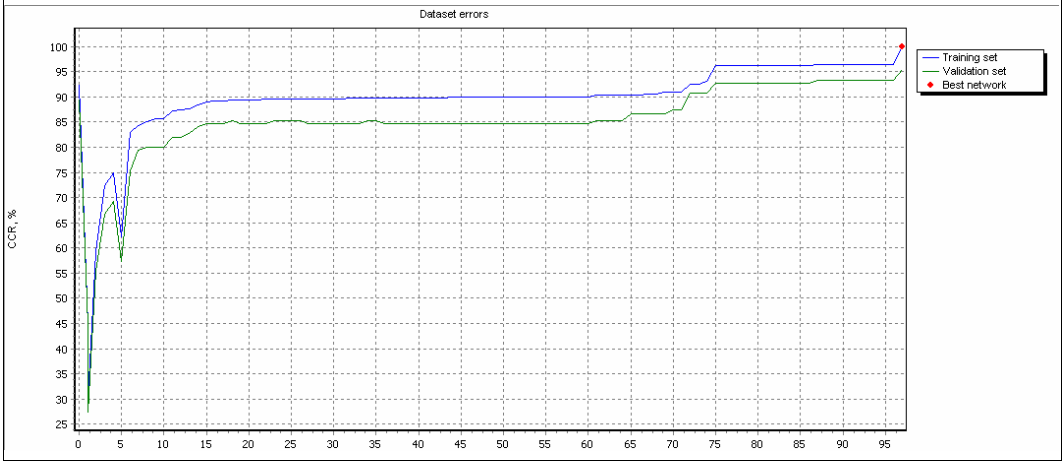
Ön işleme yöntemi: *Çevritsel dönüşüm*	Band ve sektör sayısı: 6 x 16	YSA eğitim süresi: 03:29:39

YSA metodu ve yapısı: *Conjugate Gradient Descent* 768:116:10

YSA parametreleri:
Eğitim ve doğrulama sonuçları:

	Training	Validation
CCR, %:	100	95,333333
Network error:	0,003453	0
Error improvement:	0,000118	
Iteration:	97	
Training speed, iter/sec:	0,007714	
Architecture:	[768-116-10]	
Training algorithm:	Conjugate Gradient Descent	
Training stop reason:	Desired error achieved	

YSA veri kümesi hata grafiği



Ortalama doğruluk oranları (%)

Eğitim veri kümesi: 100

output:	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10
S1	72	0	0	0	0	0	0	0	0	0
S2	0	70	0	0	0	0	0	0	0	0
S3	0	0	73	0	0	0	0	0	0	0
S4	0	0	0	71	0	0	0	0	0	0
S5	0	0	0	0	71	0	0	0	0	0
S6	0	0	0	0	0	71	0	0	0	0
S7	0	0	0	0	0	0	72	0	0	0
S8	0	0	0	0	0	0	0	66	0	0
S9	0	0	0	0	0	0	0	0	66	0
S10	0	0	0	0	0	0	0	0	0	68

Doğrulama veri kümesi: 95,33333

output:	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10
S1	14	0	0	0	0	0	0	0	0	0
S2	0	15	0	0	0	0	0	0	0	0
S3	0	0	13	0	0	0	0	0	0	0
S4	0	0	0	14	0	0	0	0	0	1
S5	0	0	0	0	14	0	0	0	0	0
S6	0	0	0	0	0	14	0	0	1	0
S7	0	0	0	1	0	0	13	0	0	0
S8	0	0	0	0	0	0	0	17	0	0
S9	0	0	0	0	0	1	0	0	16	0
S10	0	0	0	1	0	1	0	1	0	13

Test veri kümesi: 97,33333

output:	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10
S1	14	0	0	0	0	0	0	0	0	0
S2	0	15	0	0	0	0	0	0	0	0
S3	0	0	14	0	0	0	0	0	0	0
S4	0	0	0	13	0	0	0	0	0	1
S5	0	0	0	0	14	0	0	1	0	0
S6	0	0	0	0	0	14	0	0	0	0
S7	0	0	0	0	0	0	13	0	0	1
S8	0	0	0	0	0	0	0	17	0	0
S9	0	0	0	0	0	0	0	0	17	0
S10	0	0	0	1	0	0	0	0	0	15

Tüm veri kümesi: 98,9

output:	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10
S1	100	0	0	0	0	0	0	0	0	0
S2	0	100	0	0	0	0	0	0	0	0
S3	0	0	100	0	0	0	0	0	0	0
S4	0	0	0	98	0	0	0	0	0	2
S5	0	0	0	0	99	0	0	1	0	0
S6	0	0	0	0	0	99	0	0	1	0
S7	0	0	0	1	0	0	98	0	0	1
S8	0	0	0	0	0	0	0	100	0	0
S9	0	0	0	0	0	1	0	0	99	0
S10	0	0	0	2	0	1	0	1	0	96

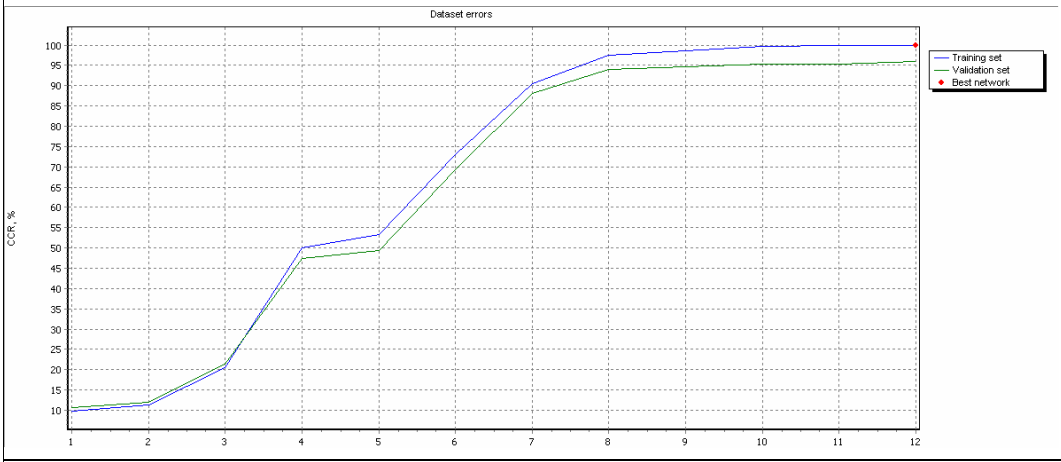
Ön işleme yöntemi: *Medyan Filtre*	Band ve sektör sayısı: 8 x 12	YSA eğitim süresi: 00:05:02

YSA metodu ve yapısı: *Online Back Propagation* 767:116:10

YSA parametreleri:
Eğitim ve doğrulama sonuçları:

	Training	Validation
CCR, %:	100	96
Network error:	0,000386	0
Error improvement:	0,000149	
Iteration:	12	
Training speed, iter/sec:	0,041783	
Architecture:	[767-116-10]	
Training algorithm:	Online Back Propagation	
Training stop reason:	Desired error achieved	

YSA veri kümesi hata grafiği



Ortalama doğruluk oranları (%)

Eğitim veri kümesi: 100

output:	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10
S1	71	0	0	0	0	0	0	0	0	0
S2	0	71	0	0	0	0	0	0	0	0
S3	0	0	74	0	0	0	0	0	0	0
S4	0	0	0	69	0	0	0	0	0	0
S5	0	0	0	0	73	0	0	0	0	0
S6	0	0	0	0	0	70	0	0	0	0
S7	0	0	0	0	0	0	72	0	0	0
S8	0	0	0	0	0	0	0	66	0	0
S9	0	0	0	0	0	0	0	0	66	0
S10	0	0	0	0	0	0	0	0	0	68

Doğrulama veri kümesi: 96

output:	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10
S1	13	0	0	1	0	0	0	0	0	1
S2	0	14	0	0	0	0	0	0	0	0
S3	0	0	13	0	0	0	0	0	0	0
S4	0	0	0	15	0	0	0	0	0	0
S5	0	0	0	0	14	0	0	0	0	0
S6	0	0	0	0	0	14	0	0	1	0
S7	0	0	0	0	0	0	14	0	0	0
S8	0	0	0	0	0	0	1	16	0	0
S9	0	0	0	0	0	0	0	0	17	0
S10	0	0	0	0	0	1	0	0	1	14

Test veri kümesi: 94

output:	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10
S1	14	0	0	0	0	0	0	0	0	0
S2	0	15	0	0	0	0	0	0	0	0
S3	0	0	13	0	0	0	0	0	0	0
S4	0	0	0	13	1	2	0	0	0	0
S5	0	0	0	0	11	1	0	1	0	0
S6	0	0	0	0	0	14	0	0	0	1
S7	0	0	0	0	0	0	14	0	0	0
S8	0	0	0	0	0	0	0	16	0	1
S9	0	0	0	0	0	1	0	0	16	0
S10	0	0	0	0	1	0	0	0	0	15

Tüm veri kümesi: 98,5

output:	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10
S1	98	0	0	1	0	0	0	0	0	1
S2	0	100	0	0	0	0	0	0	0	0
S3	0	0	100	0	0	0	0	0	0	0
S4	0	0	0	97	1	2	0	0	0	0
S5	0	0	0	0	98	1	0	1	0	0
S6	0	0	0	0	0	98	0	0	1	1
S7	0	0	0	0	0	0	100	0	0	0
S8	0	0	0	0	0	0	1	98	0	1
S9	0	0	0	0	0	1	0	0	99	0
S10	0	0	0	0	1	1	0	0	1	97

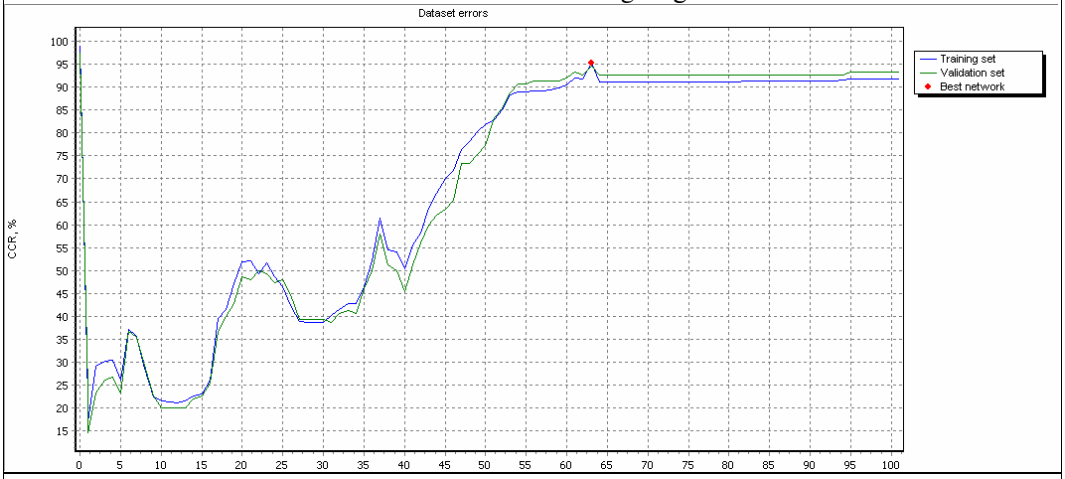
Ön işleme yöntemi: *Çevritsel dönüşüm*	Band ve sektör sayısı: 8 x 12	YSA eğitim süresi: 00:23:13

YSA metodu ve yapısı: *Batch Back Propagation* 768:116:10

YSA parametreleri:
Eğitim ve doğrulama sonuçları:

	Training	Validation
CCR, %:	91,714286	93,333333
Network error:	0,020369	0
Error improvement:	0,000138	
Iteration:	101	
Training speed, iter/sec:	0,073339	
Architecture:	[768-116-10]	
Training algorithm:	Batch Back Propagation	
Training stop reason:	All iterations done	

YSA veri kümesi hata grafiği



Ortalama doğruluk oranları (%)

Eğitim veri kümesi: 98,857143

output:	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10
S1	71	0	0	0	0	0	0	0	0	0
S2	0	73	0	0	0	0	0	0	0	0
S3	0	0	70	0	0	0	0	0	0	0
S4	0	0	0	70	0	0	0	0	0	2
S5	0	0	0	0	71	0	0	0	0	0
S6	0	0	0	0	0	69	2	0	0	1
S7	0	0	0	0	0	0	70	0	0	2
S8	0	0	0	0	0	0	0	65	0	0
S9	0	0	0	0	0	0	0	0	66	1
S10	0	0	0	0	0	0	0	0	0	67

Doğrulama veri kümesi: 97,33333

output:	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10
S1	13	0	0	0	1	0	0	0	0	0
S2	0	14	0	0	0	0	0	0	0	0
S3	0	0	15	0	0	0	0	0	0	0
S4	0	0	0	14	0	0	0	0	0	0
S5	0	0	0	0	14	0	0	0	0	0
S6	0	0	0	1	0	13	0	0	0	0
S7	0	0	0	0	0	1	13	0	0	0
S8	0	0	0	0	0	0	0	18	0	0
S9	0	0	0	0	1	0	0	0	16	0
S10	0	0	0	0	0	0	0	0	0	16

Test veri kümesi: 100

output:	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10
S1	15	0	0	0	0	0	0	0	0	0
S2	0	13	0	0	0	0	0	0	0	0
S3	0	0	15	0	0	0	0	0	0	0
S4	0	0	0	14	0	0	0	0	0	0
S5	0	0	0	0	15	0	0	0	0	0
S6	0	0	0	0	0	14	0	0	0	0
S7	0	0	0	0	0	0	14	0	0	0
S8	0	0	0	0	0	0	0	17	0	0
S9	0	0	0	0	0	0	0	0	16	0
S10	0	0	0	0	0	0	0	0	0	17

Tüm veri kümesi: 98,8

output:	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10
S1	99	0	0	0	1	0	0	0	0	0
S2	0	100	0	0	0	0	0	0	0	0
S3	0	0	100	0	0	0	0	0	0	0
S4	0	0	0	98	0	0	0	0	0	2
S5	0	0	0	0	100	0	0	0	0	0
S6	0	0	0	1	0	96	2	0	0	1
S7	0	0	0	0	0	1	97	0	0	2
S8	0	0	0	0	0	0	0	100	0	0
S9	0	0	0	0	1	0	0	0	98	1
S10	0	0	0	0	0	0	0	0	0	100

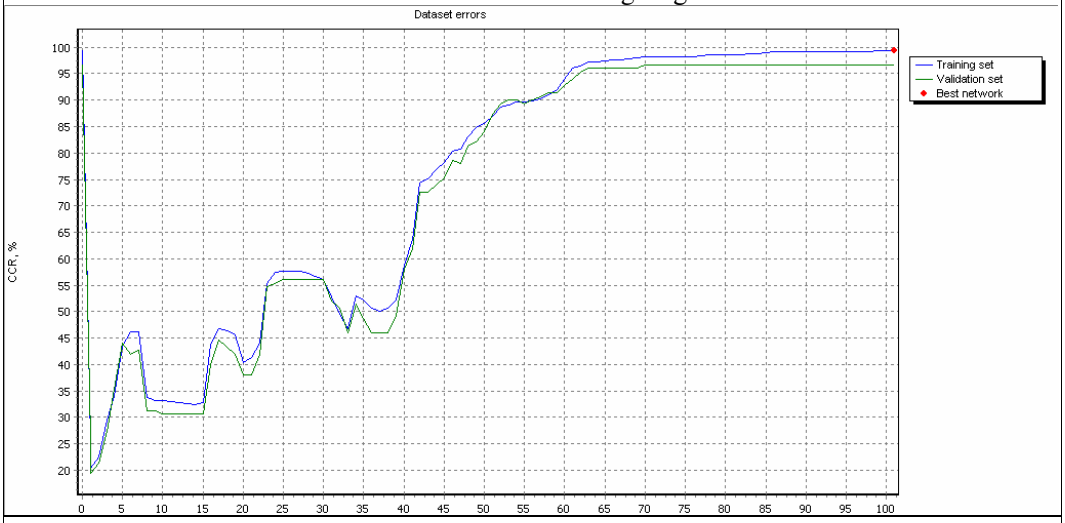
Ön işleme yöntemi: *Çevritsel dönüşüm*	Band ve sektör sayısı: 8 x 16	YSA eğitim süresi: 00:42:28

YSA metodu ve yapısı: *Batch Back Propagation* 1023:154:10

YSA parametreleri:
Eğitim ve doğrulama sonuçları:

	Training	Validation
CCR, %:	99,285714	96,666667
Network error:	0,001663	0
Error improvement:	0,000036	
Iteration:	101	
Training speed, iter/sec:	0,039813	
Architecture:	[1023-154-10]	
Training algorithm:	Batch Back Propagation	
Training stop reason:	All iterations done	

YSA veri kümesi hata grafiği



Ortalama doğruluk oranları (%)

Eğitim veri kümesi: 99,285714

output:	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10
S1	71	0	0	0	0	0	0	0	0	0
S2	0	72	0	0	0	0	0	0	0	0
S3	0	0	71	0	0	0	0	0	0	0
S4	0	0	0	72	0	0	0	0	0	0
S5	0	0	0	0	71	0	0	0	0	0
S6	0	0	0	2	0	69	1	0	0	0
S7	0	0	0	0	0	0	71	0	0	0
S8	0	0	0	0	0	0	0	67	0	0
S9	0	0	0	0	0	0	0	0	66	1
S10	0	0	0	0	0	0	0	1	0	65

Doğrulama veri kümesi: 96,66667

output:	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10
S1	14	0	0	0	0	0	0	1	0	0
S2	0	14	0	0	0	0	0	0	0	0
S3	0	0	14	0	0	0	0	0	0	0
S4	0	0	0	13	0	0	0	1	0	0
S5	0	0	0	0	13	0	0	1	0	0
S6	0	0	0	0	0	14	0	0	0	0
S7	0	0	0	1	0	0	14	0	0	0
S8	0	0	0	0	0	0	0	16	0	0
S9	0	0	0	0	0	0	0	0	17	0
S10	0	0	1	0	0	0	0	0	0	16

Test veri kümesi: 98,666667

output:	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10
S1	14	0	0	0	0	0	0	0	0	0
S2	0	14	0	0	0	0	0	0	0	0
S3	0	0	15	0	0	0	0	0	0	0
S4	0	0	0	14	0	0	0	0	0	0
S5	0	0	0	0	15	0	0	0	0	0
S6	0	0	0	0	0	14	0	0	0	0
S7	0	0	0	0	0	0	13	0	0	1
S8	0	0	0	0	0	0	0	17	0	0
S9	0	0	0	0	1	0	0	0	15	0
S10	0	0	0	0	0	0	0	0	0	17

Tüm veri kümesi: 98,8

output:	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10
S1	99	0	0	0	0	0	0	1	0	0
S2	0	100	0	0	0	0	0	0	0	0
S3	0	0	100	0	0	0	0	0	0	0
S4	0	0	0	99	0	0	0	1	0	0
S5	0	0	0	0	99	0	0	1	0	0
S6	0	0	0	2	0	97	1	0	0	0
S7	0	0	0	1	0	0	98	0	0	1
S8	0	0	0	0	0	0	0	100	0	0
S9	0	0	0	0	1	0	0	0	98	1
S10	0	0	1	0	0	0	0	1	0	98

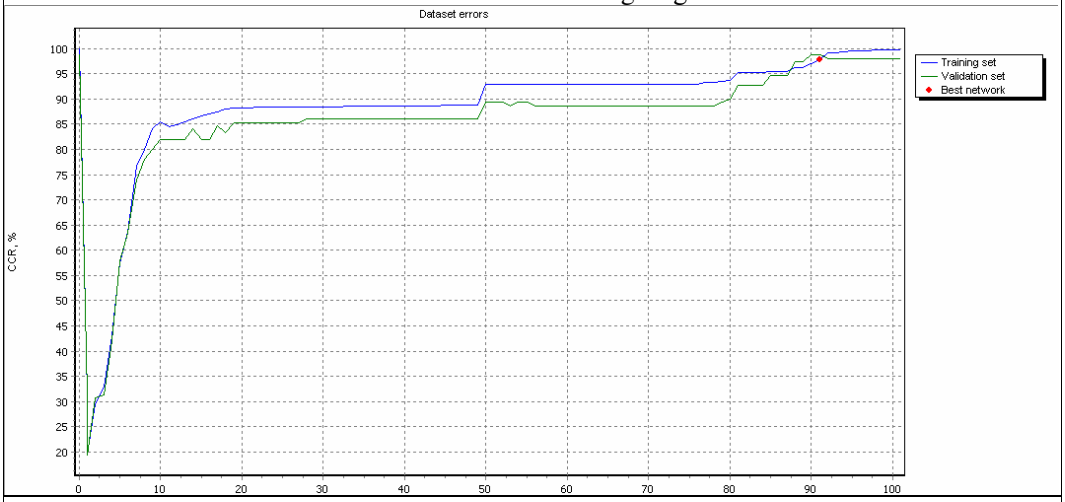
Ön işleme yöntemi: *Çevritsel dönüşüm*	Band ve sektör sayısı: 4 x 16	YSA eğitim süresi: 01:49:58

YSA metodu ve yapısı: *Conjugate Gradient Descent* 512:78:10

YSA parametreleri:
Eğitim ve doğrulama sonuçları:

	Training	Validation
CCR, %:	99,714286	98
Network error:	0,000289	0
Error improvement:	0,000003	
Iteration:	101	
Training speed, iter/sec:	0,015323	
Architecture:	[512-78-10]	
Training algorithm:	Conjugate Gradient Descent	
Training stop reason:	All iterations done	

YSA veri kümesi hata grafiği



Ortalama doğruluk oranları (%)

Eğitim veri kümesi: 100

output:	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10
S1	71	0	0	0	0	0	0	0	0	0
S2	0	72	0	0	0	0	0	0	0	0
S3	0	0	71	0	0	0	0	0	0	0
S4	0	0	0	72	0	0	0	0	0	0
S5	0	0	0	0	72	0	0	0	0	0
S6	0	0	0	0	0	70	0	0	0	0
S7	0	0	0	0	0	0	72	0	0	0
S8	0	0	0	0	0	0	0	66	0	0
S9	0	0	0	0	0	0	0	0	67	0
S10	0	0	0	0	0	0	0	0	0	67

Doğrulama veri kümesi: 98,66667

output:	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10
S1	13	0	0	1	0	0	0	0	0	0
S2	0	14	0	0	0	0	0	0	0	0
S3	0	0	15	0	0	0	0	0	0	0
S4	0	0	0	14	0	0	0	0	0	0
S5	0	0	0	0	14	0	0	0	0	0
S6	0	0	0	0	0	15	0	0	0	0
S7	0	0	0	0	0	0	14	0	0	0
S8	0	0	0	0	0	0	0	17	0	0
S9	0	0	0	0	0	0	0	0	16	0
S10	0	0	0	0	1	0	0	0	0	16

Test veri kümesi: 95,33333

output:	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10
S1	15	0	0	0	0	0	0	0	0	0
S2	0	14	0	0	0	0	0	0	0	0
S3	0	0	14	0	0	0	0	0	0	0
S4	0	0	0	14	0	0	0	0	0	0
S5	0	0	0	0	13	0	1	0	0	0
S6	0	0	0	0	0	13	2	0	0	0
S7	0	0	0	0	0	0	12	1	0	1
S8	0	0	0	0	0	0	0	17	0	0
S9	0	0	0	0	0	0	0	0	16	1
S10	0	0	0	0	1	0	0	0	0	15

Tüm veri kümesi: 99,1

output:	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10
S1	99	0	0	1	0	0	0	0	0	0
S2	0	100	0	0	0	0	0	0	0	0
S3	0	0	100	0	0	0	0	0	0	0
S4	0	0	0	100	0	0	0	0	0	0
S5	0	0	0	0	99	0	1	0	0	0
S6	0	0	0	0	0	98	2	0	0	0
S7	0	0	0	0	0	0	98	1	0	1
S8	0	0	0	0	0	0	0	100	0	0
S9	0	0	0	0	0	0	0	0	99	1
S10	0	0	0	1	1	0	0	0	0	98

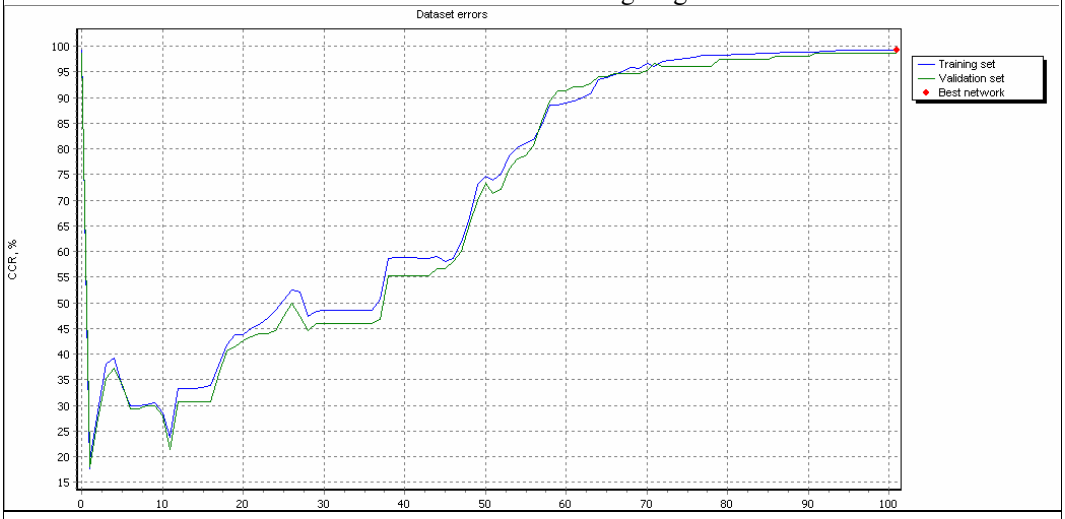
Ön işleme yöntemi: *Çevritsel dönüşüm*	Band ve sektör sayısı: 4 x 16	YSA eğitim süresi: 00:19:54

YSA metodu ve yapısı: *Batch Back Propagation* 512:78:10

YSA parametreleri:
Eğitim ve doğrulama sonuçları:

	Training	Validation
CCR, %:	99,142857	98,666667
Network error:	0,00279	0
Error improvement:	0,000088	
Iteration:	101	
Training speed, iter/sec:	0,085088	
Architecture:	[512-78-10]	
Training algorithm:	Batch Back Propagation	
Training stop reason:	All iterations done	

YSA veri kümesi hata grafiği



Ortalama doğruluk oranları (%)

Eğitim veri kümesi: 99,142857

output:	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10
S1	71	0	0	0	0	0	0	0	0	0
S2	0	72	0	0	0	0	0	0	0	0
S3	0	0	71	0	0	0	0	0	0	0
S4	0	0	0	71	0	0	0	0	0	1
S5	0	0	0	0	72	0	0	0	0	0
S6	0	0	0	0	0	68	1	0	0	1
S7	0	0	0	1	0	0	71	0	0	0
S8	0	0	0	0	0	0	0	66	0	0
S9	0	0	0	0	1	0	0	0	66	0
S10	0	0	0	1	0	0	0	0	0	66

Doğrulama veri kümesi: 98,66667

output:	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10
S1	13	0	0	1	0	0	0	0	0	0
S2	0	14	0	0	0	0	0	0	0	0
S3	0	0	15	0	0	0	0	0	0	0
S4	0	0	0	14	0	0	0	0	0	0
S5	0	0	0	0	14	0	0	0	0	0
S6	0	0	0	0	0	15	0	0	0	0
S7	0	0	0	0	0	0	14	0	0	0
S8	0	0	0	0	0	0	0	17	0	0
S9	0	0	0	0	0	0	0	0	16	0
S10	0	0	0	0	1	0	0	0	0	16

Test veri kümesi: 95,33333

output:	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10
S1	15	0	0	0	0	0	0	0	0	0
S2	0	14	0	0	0	0	0	0	0	0
S3	0	0	14	0	0	0	0	0	0	0
S4	0	0	0	14	0	0	0	0	0	0
S5	0	0	0	0	13	0	0	1	0	0
S6	0	0	1	1	0	13	0	0	0	0
S7	0	0	0	0	0	0	12	1	0	1
S8	0	0	0	0	0	0	0	17	0	0
S9	0	0	0	0	0	0	0	0	16	1
S10	0	0	0	0	1	0	0	0	0	15

Tüm veri kümesi: 98,5

output:	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10
S1	99	0	0	1	0	0	0	0	0	0
S2	0	100	0	0	0	0	0	0	0	0
S3	0	0	100	0	0	0	0	0	0	0
S4	0	0	0	99	0	0	0	0	0	1
S5	0	0	0	0	99	0	0	1	0	0
S6	0	0	1	1	0	96	1	0	0	1
S7	0	0	0	1	0	0	97	1	0	1
S8	0	0	0	0	0	0	0	100	0	0
S9	0	0	0	0	1	0	0	0	98	1
S10	0	0	0	1	2	0	0	0	0	97

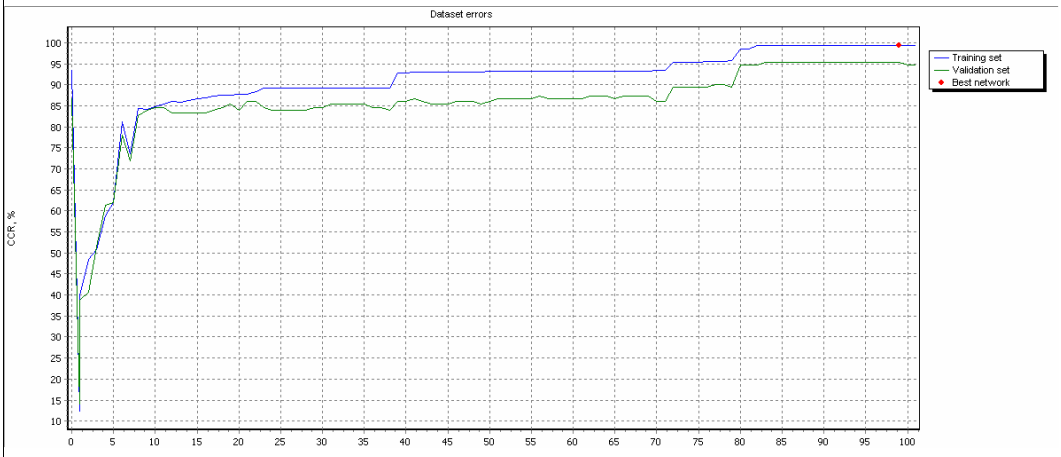
Ön işleme yöntemi: *Medyan Filtre*	Band ve sektör sayısı: 4 x 16	YSA eğitim süresi: 01:46:38

YSA metodu ve yapısı: *Conjugate Gradient Descent* 512:78:10

YSA parametreleri:
Eğitim ve doğrulama sonuçları:

	Training	Validation
CCR, %:	99,428571	94,666667
Network error:	0,000571	0
Error improvement:	1,95E-10	
Iteration:	101	
Training speed, iter/sec:	0,015794	
Architecture:	[512-78-10]	
Training algorithm:	Conjugate Gradient Descent	
Training stop reason:	All iterations done	

YSA veri kümesi hata grafiği



Ortalama doğruluk oranları (%)

Eğitim veri kümesi: 99,428571

output:	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10
S1	72	0	0	0	0	0	0	0	0	0
S2	0	72	0	0	0	0	0	0	0	0
S3	0	0	70	0	0	0	0	0	0	0
S4	0	0	0	71	0	0	0	0	0	0
S5	0	0	0	0	72	0	0	0	0	0
S6	1	0	0	0	0	71	1	0	0	0
S7	0	0	1	0	0	0	68	0	1	0
S8	0	0	0	0	0	0	0	66	0	0
S9	0	0	0	0	0	0	0	0	68	0
S10	0	0	0	0	0	0	0	0	0	66

Doğrulama veri kümesi: 95,333333

output:	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10
S1	14	0	0	0	0	0	0	0	0	0
S2	0	14	0	0	0	0	0	0	0	0
S3	0	0	14	0	0	0	0	0	1	0
S4	0	0	0	14	1	0	0	0	0	0
S5	0	0	0	0	13	0	0	0	0	0
S6	0	0	0	0	0	13	0	0	0	1
S7	0	0	0	0	1	0	12	0	2	0
S8	0	0	0	0	0	0	0	17	0	0
S9	0	0	0	0	0	0	0	0	16	0
S10	0	0	0	0	0	1	0	0	0	16

Test veri kümesi: 96

output:	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10
S1	14	0	0	0	0	0	0	0	0	0
S2	0	14	0	0	0	0	0	0	0	0
S3	0	0	15	0	0	0	0	0	0	0
S4	0	0	0	13	0	1	0	0	0	0
S5	0	0	0	0	14	0	1	0	0	0
S6	0	0	0	0	0	13	0	0	0	0
S7	0	0	0	0	1	0	12	0	2	0
S8	0	0	0	0	1	0	0	16	0	0
S9	0	0	0	0	0	0	0	0	16	0
S10	0	0	0	0	0	0	0	0	0	17

Tüm veri kümesi: 98,3

output:	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10
S1	100	0	0	0	0	0	0	0	0	0
S2	0	100	0	0	0	0	0	0	0	0
S3	0	0	99	0	0	0	0	0	1	0
S4	0	0	0	98	1	1	0	0	0	0
S5	0	0	0	0	99	0	1	0	0	0
S6	1	0	0	0	0	97	1	0	0	1
S7	0	0	1	0	2	0	92	0	5	0
S8	0	0	0	0	1	0	0	99	0	0
S9	0	0	0	0	0	0	0	0	100	0
S10	0	0	0	0	0	1	0	0	0	99

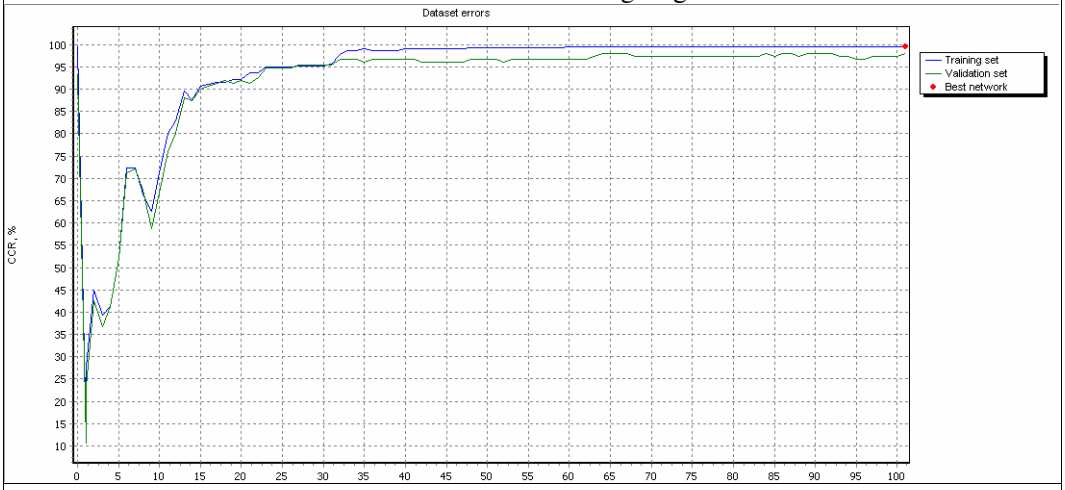
Ön işleme yöntemi: *Medyan Filtre*	Band ve sektör sayısı: 4 x 16	YSA eğitim süresi: 00:14:41

YSA metodu ve yapısı: *Quick Propagation* 512:78:10

YSA parametreleri:
Eğitim ve doğrulama sonuçları:

	Training	Validation
CCR, %:	99,571429	98
Network error:	0,000429	0
Error improvement:	5,41E-09	
Iteration:	101	
Training speed, iter/sec:	0,115187	
Architecture:	[512-78-10]	
Training algorithm:	Quick Propagation	
Training stop reason:	All iterations done	

YSA veri kümesi hata grafiği



Ortalama doğruluk oranları (%)

Eğitim veri kümesi: 99,571429

output:	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10
S1	72	0	0	0	0	0	0	0	0	0
S2	0	72	0	0	0	0	0	0	0	0
S3	0	0	70	0	0	0	0	0	0	0
S4	0	0	0	71	0	0	0	0	0	0
S5	0	0	0	0	72	0	0	0	0	0
S6	0	0	0	0	0	71	0	0	2	0
S7	0	0	0	0	0	0	70	0	0	0
S8	0	0	0	0	0	0	0	66	0	0
S9	0	0	0	0	0	0	0	0	67	1
S10	0	0	0	0	0	0	0	0	0	66

Doğrulama veri kümesi: 98

output:	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10
S1	14	0	0	0	0	0	0	0	0	0
S2	0	14	0	0	0	0	0	0	0	0
S3	0	0	14	1	0	0	0	0	0	0
S4	0	0	0	14	1	0	0	0	0	0
S5	0	0	0	0	13	0	0	0	0	0
S6	0	0	0	0	0	13	0	0	0	1
S7	0	0	0	0	0	0	15	0	0	0
S8	0	0	0	0	0	0	0	17	0	0
S9	0	0	0	0	0	0	0	0	16	0
S10	0	0	0	0	0	0	0	0	0	17

Test veri kümesi: 96,666667

output:	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10
S1	14	0	0	0	0	0	0	0	0	0
S2	0	14	0	0	0	0	0	0	0	0
S3	0	0	15	0	0	0	0	0	0	0
S4	0	1	0	13	0	0	0	0	0	0
S5	2	0	0	0	13	0	0	0	0	0
S6	0	0	0	0	0	13	0	0	0	0
S7	0	0	0	0	0	0	15	0	0	0
S8	0	0	0	0	2	0	0	15	0	0
S9	0	0	0	0	0	0	0	0	16	0
S10	0	0	0	0	0	0	0	0	0	17

Tüm veri kümesi: 98,9

output:	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10
S1	100	0	0	0	0	0	0	0	0	0
S2	0	100	0	0	0	0	0	0	0	0
S3	0	0	99	1	0	0	0	0	0	0
S4	0	1	0	98	1	0	0	0	0	0
S5	2	0	0	0	98	0	0	0	0	0
S6	0	0	0	0	0	97	0	0	2	1
S7	0	0	0	0	0	0	100	0	0	0
S8	0	0	0	0	2	0	0	98	0	0
S9	0	0	0	0	0	0	0	0	99	1
S10	0	0	0	0	0	0	0	0	0	100

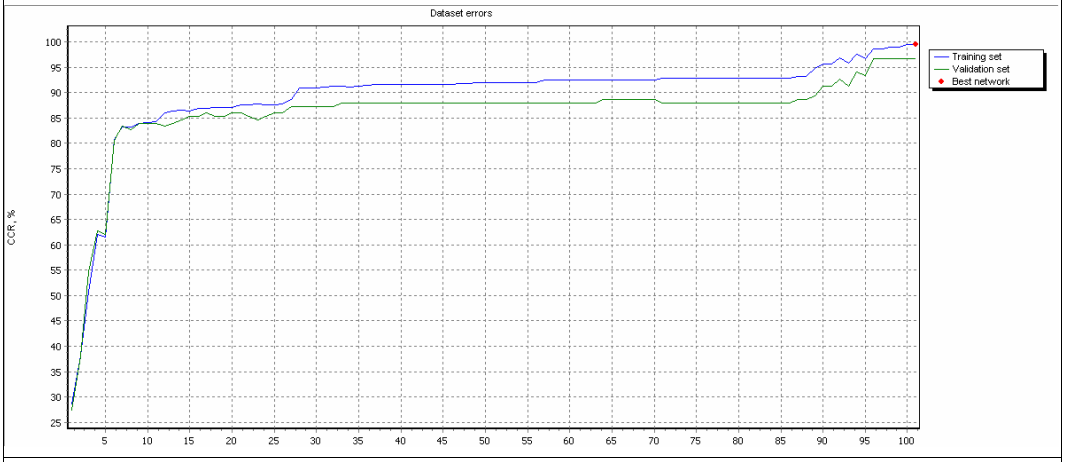
Ön işleme yöntemi: *Dalgacık dönüşüm*	Band ve sektör sayısı: 4 x 16	YSA eğitim süresi: 01:30:35

YSA metodu ve yapısı: *Conjugate Gradient Descent* 512:78:10

YSA parametreleri:
Eğitim ve doğrulama sonuçları:

	Training	Validation
CCR, %:	99,571429	96,666667
Network error:	0,000742	0
Error improvement:	0,000149	
Iteration:	101	
Training speed, iter/sec:	0,018617	
Architecture:	[512-78-10]	
Training algorithm:	Conjugate Gradient Descent	
Training stop reason:	All iterations done	

YSA veri kümesi hata grafiği



Ortalama doğruluk oranları (%)

Eğitim veri kümesi: 99,571429

output:	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10
S1	72	0	0	0	0	0	0	0	0	0
S2	0	70	0	0	0	0	0	0	0	0
S3	0	0	73	0	0	0	0	0	0	0
S4	0	0	0	71	0	0	0	0	0	0
S5	0	0	0	0	71	0	0	0	0	0
S6	0	0	0	0	0	72	0	0	0	0
S7	0	0	0	2	0	0	68	1	0	0
S8	0	0	0	0	0	0	0	67	0	0
S9	0	0	0	0	0	0	0	0	67	0
S10	0	0	0	0	0	0	0	0	0	66

Doğrulama veri kümesi: 96,66667

output:	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10
S1	13	0	0	0	1	0	0	0	0	0
S2	0	15	0	0	0	0	0	0	0	0
S3	0	0	13	0	0	0	0	0	0	0
S4	0	0	0	14	1	0	0	0	0	0
S5	0	0	0	0	14	0	0	0	0	0
S6	0	0	0	0	0	14	0	0	0	0
S7	0	0	0	0	0	1	14	0	0	0
S8	0	0	0	0	0	0	0	16	0	0
S9	0	0	0	0	1	0	0	0	16	0
S10	0	0	0	0	1	0	0	0	0	16

Test veri kümesi: 96,66667

output:	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10
S1	14	0	0	0	0	0	0	0	0	0
S2	0	15	0	0	0	0	0	0	0	0
S3	0	0	14	0	0	0	0	0	0	0
S4	0	0	0	14	0	0	0	0	0	0
S5	0	0	0	0	15	0	0	0	0	0
S6	0	0	0	0	0	14	0	0	0	0
S7	0	0	0	1	0	2	10	1	0	0
S8	0	0	0	0	0	0	0	17	0	0
S9	0	0	0	0	0	0	0	0	16	0
S10	0	0	0	0	1	0	0	0	0	16

Tüm veri kümesi: 98,7

output:	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10
S1	99	0	0	0	1	0	0	0	0	0
S2	0	100	0	0	0	0	0	0	0	0
S3	0	0	100	0	0	0	0	0	0	0
S4	0	0	0	99	1	0	0	0	0	0
S5	0	0	0	0	100	0	0	0	0	0
S6	0	0	0	0	0	100	0	0	0	0
S7	0	0	0	3	0	3	92	2	0	0
S8	0	0	0	0	0	0	0	100	0	0
S9	0	0	0	0	1	0	0	0	99	0
S10	0	0	0	0	2	0	0	0	0	98

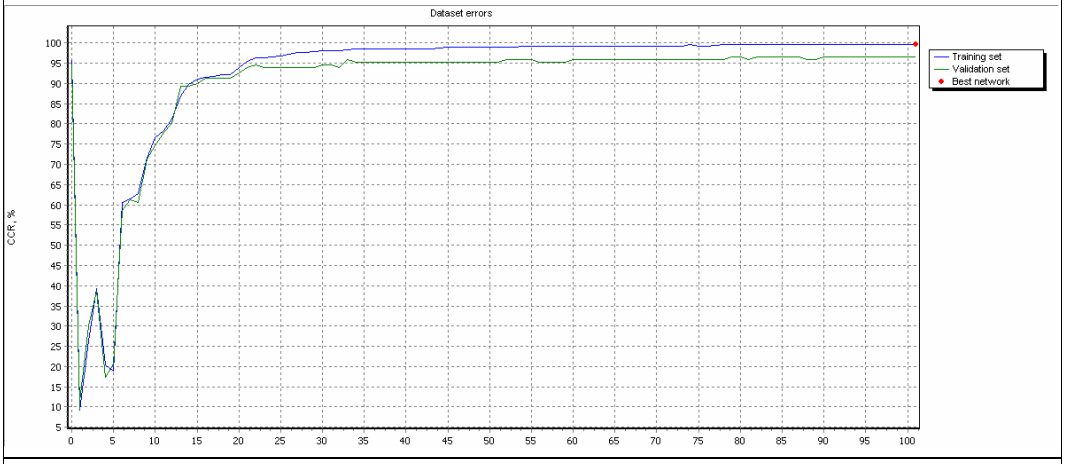
Ön işleme yöntemi: *Dalgacık dönüşüm*	Band ve sektör sayısı: 4 x 16	YSA eğitim süresi: 00:23:31

YSA metodu ve yapısı: *Quick Propagation* 512:78:10

YSA parametreleri:
Eğitim ve doğrulama sonuçları:

	Training	Validation
CCR, %:	99,714286	96,666667
Network error:	0,000286	0
Error improvement:	2,43E-09	
Iteration:	101	
Training speed, iter/sec:	0,072125	
Architecture:	[512-78-10]	
Training algorithm:	Quick Propagation	
Training stop reason:	All iterations done	

YSA veri kümesi hata grafiği



Ortalama doğruluk oranları (%)

Eğitim veri kümesi: 99,714286

output:	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10
S1	72	0	0	0	0	0	0	0	0	0
S2	0	70	0	0	0	0	0	0	0	0
S3	0	0	73	0	0	0	0	0	0	0
S4	0	0	0	71	0	0	0	0	0	0
S5	0	0	0	0	71	0	0	0	0	0
S6	0	0	0	0	1	70	0	0	0	1
S7	0	0	0	0	0	0	71	0	0	0
S8	0	0	0	0	0	0	0	67	0	0
S9	0	0	0	0	0	0	0	0	67	0
S10	0	0	0	0	0	0	0	0	0	66

Doğrulama veri kümesi: 96,66667

output:	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10
S1	13	0	0	0	0	0	0	0	0	1
S2	0	15	0	0	0	0	0	0	0	0
S3	0	0	13	0	0	0	0	0	0	0
S4	0	0	0	14	1	0	0	0	0	0
S5	0	0	0	0	14	0	0	0	0	0
S6	0	0	0	0	0	13	0	0	0	1
S7	0	0	0	0	0	0	15	0	0	0
S8	0	0	0	0	0	0	0	16	0	0
S9	0	0	0	0	1	0	0	0	16	0
S10	0	0	0	0	1	0	0	0	0	16

Test veri kümesi: 98,666667

output:	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10
S1	14	0	0	0	0	0	0	0	0	0
S2	0	15	0	0	0	0	0	0	0	0
S3	0	0	14	0	0	0	0	0	0	0
S4	0	0	0	14	0	0	0	0	0	0
S5	0	0	0	0	15	0	0	0	0	0
S6	0	0	0	0	0	14	0	0	0	0
S7	0	0	0	0	0	0	14	0	0	0
S8	0	0	0	0	0	0	0	17	0	0
S9	0	0	0	0	0	0	0	0	16	0
S10	0	0	0	0	0	0	0	2	0	15

Tüm veri kümesi: 99,1

output:	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10
S1	99	0	0	0	0	0	0	0	0	1
S2	0	100	0	0	0	0	0	0	0	0
S3	0	0	100	0	0	0	0	0	0	0
S4	0	0	0	99	1	0	0	0	0	0
S5	0	0	0	0	100	0	0	0	0	0
S6	0	0	0	0	1	97	0	0	0	2
S7	0	0	0	0	0	0	100	0	0	0
S8	0	0	0	0	0	0	0	100	0	0
S9	0	0	0	0	1	0	0	0	99	0
S10	0	0	0	0	1	0	0	2	0	97

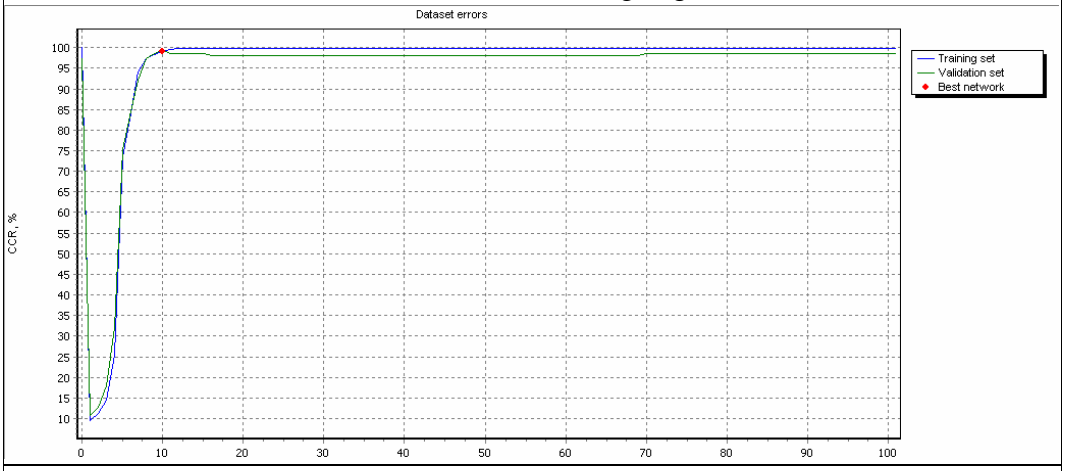
Ön işleme yöntemi: *Çevritsel dönüşüm*	Band ve sektör sayısı: 5 x 16	YSA eğitim süresi: 00:30:23

YSA metodu ve yapısı: *Online Back Propagation* 640:96:10

YSA parametreleri:
Eğitim ve doğrulama sonuçları:

	Training	Validation
CCR, %:	99,857143	98,666667
Network error:	0,000156	0
Error improvement:	1,90E-07	
Iteration:	101	
Training speed, iter/sec:	0,055695	
Architecture:	[640-96-10]	
Training algorithm:	Online Back Propagation	
Training stop reason:	All iterations done	

YSA veri kümesi hata grafiği



Ortalama doğruluk oranları (%)

Eğitim veri kümesi: 99,285714

output:	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10
S1	72	0	0	0	0	0	0	0	0	0
S2	0	70	0	0	0	0	0	0	0	0
S3	0	0	72	0	0	0	0	0	0	0
S4	0	0	0	71	0	0	0	0	1	0
S5	0	0	0	0	72	0	0	0	0	0
S6	0	0	0	0	0	68	0	0	1	1
S7	0	0	0	0	0	0	72	1	0	0
S8	0	0	0	0	0	0	0	66	0	0
S9	0	0	0	0	0	0	0	0	65	1
S10	0	0	0	0	0	0	0	0	0	67

Doğrulama veri kümesi: 99,333333

output:	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10
S1	14	0	0	0	0	0	0	0	0	0
S2	0	15	0	0	0	0	0	0	0	0
S3	0	0	14	0	0	0	0	0	0	0
S4	0	0	0	14	0	0	0	0	0	0
S5	0	0	0	0	14	0	0	0	0	0
S6	0	0	0	0	0	15	0	0	0	0
S7	0	0	0	0	0	0	12	0	0	1
S8	0	0	0	0	0	0	0	18	0	0
S9	0	0	0	0	0	0	0	0	17	0
S10	0	0	0	0	0	0	0	0	0	16

Test veri kümesi: 98

output:	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10
S1	13	0	0	0	1	0	0	0	0	0
S2	0	15	0	0	0	0	0	0	0	0
S3	0	0	14	0	0	0	0	0	0	0
S4	0	0	0	14	0	0	0	0	0	0
S5	0	0	0	0	13	0	0	0	0	1
S6	0	0	0	0	0	14	0	0	0	1
S7	0	0	0	0	0	0	14	0	0	0
S8	0	0	0	0	0	0	0	16	0	0
S9	0	0	0	0	0	0	0	0	17	0
S10	0	0	0	0	0	0	0	0	0	17

Tüm veri kümesi: 99,1

output:	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10
S1	99	0	0	0	1	0	0	0	0	0
S2	0	100	0	0	0	0	0	0	0	0
S3	0	0	100	0	0	0	0	0	0	0
S4	0	0	0	99	0	0	0	0	1	0
S5	0	0	0	0	99	0	0	0	0	1
S6	0	0	0	0	0	97	0	0	1	2
S7	0	0	0	0	0	0	98	1	0	1
S8	0	0	0	0	0	0	0	100	0	0
S9	0	0	0	0	0	0	0	0	99	1
S10	0	0	0	0	0	0	0	0	0	100

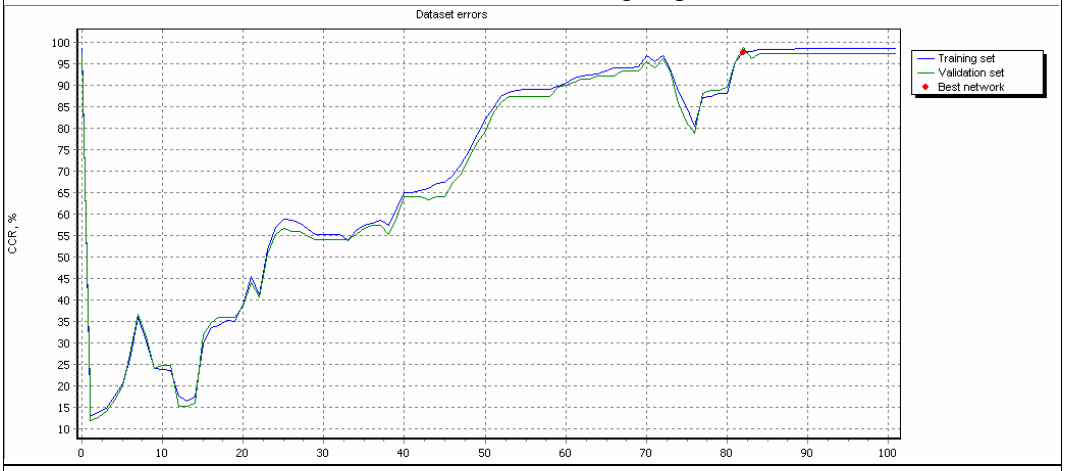
Ön işleme yöntemi: *Çevritsel dönüşüm*	Band ve sektör sayısı: 5 x 16	YSA eğitim süresi: 00:30:08

YSA metodu ve yapısı: *Batch Back Propagation* 640:96:10

YSA parametreleri:
Eğitim ve doğrulama sonuçları:

	Training	Validation
CCR, %:	98,571429	97,333333
Network error:	0,002852	0
Error improvement:	0,000059	
Iteration:	101	
Training speed, iter/sec:	0,056316	
Architecture:	[640-96-10]	
Training algorithm:	Batch Back Propagation	
Training stop reason:	All iterations done	

YSA veri kümesi hata grafiği



Ortalama doğruluk oranları (%)

Eğitim veri kümesi: 97,428571

output:	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10
S1	71	0	0	0	1	0	0	0	0	0
S2	0	70	0	0	0	0	0	0	0	0
S3	0	0	72	0	0	0	0	0	0	0
S4	0	0	0	66	0	0	0	1	0	5
S5	0	0	0	0	71	0	0	1	0	0
S6	0	1	0	0	0	68	1	0	0	0
S7	0	0	0	0	0	1	70	1	0	1
S8	0	0	0	0	0	0	0	66	0	0
S9	0	0	0	0	1	0	0	0	64	1
S10	0	0	1	0	1	0	0	0	1	64

Doğrulama veri kümesi: 98,66667

output:	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10
S1	14	0	0	0	0	0	0	0	0	0
S2	0	15	0	0	0	0	0	0	0	0
S3	0	0	14	0	0	0	0	0	0	0
S4	0	0	0	14	0	0	0	0	0	0
S5	0	0	0	0	14	0	0	0	0	0
S6	0	0	0	0	0	15	0	0	0	0
S7	0	0	0	0	0	0	12	0	0	1
S8	0	0	0	0	0	0	0	18	0	0
S9	0	0	0	0	0	0	0	0	17	0
S10	0	0	0	0	1	0	0	0	0	15

Test veri kümesi: 96,66667

output:	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10
S1	12	0	0	0	2	0	0	0	0	0
S2	0	15	0	0	0	0	0	0	0	0
S3	0	0	14	0	0	0	0	0	0	0
S4	0	0	0	14	0	0	0	0	0	0
S5	0	0	0	0	13	0	0	1	0	0
S6	0	0	0	0	0	13	1	0	0	1
S7	0	0	0	0	0	0	14	0	0	0
S8	0	0	0	0	0	0	0	16	0	0
S9	0	0	0	0	0	0	0	0	17	0
S10	0	0	0	0	0	0	0	0	0	17

Tüm veri kümesi: 97,5

output:	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10
S1	97	0	0	0	3	0	0	0	0	0
S2	0	100	0	0	0	0	0	0	0	0
S3	0	0	100	0	0	0	0	0	0	0
S4	0	0	0	94	0	0	0	1	0	5
S5	0	0	0	0	98	0	0	2	0	0
S6	0	1	0	0	0	96	2	0	0	1
S7	0	0	0	0	0	1	96	1	0	2
S8	0	0	0	0	0	0	0	100	0	0
S9	0	0	0	0	1	0	0	0	98	1
S10	0	0	1	0	2	0	0	0	1	96

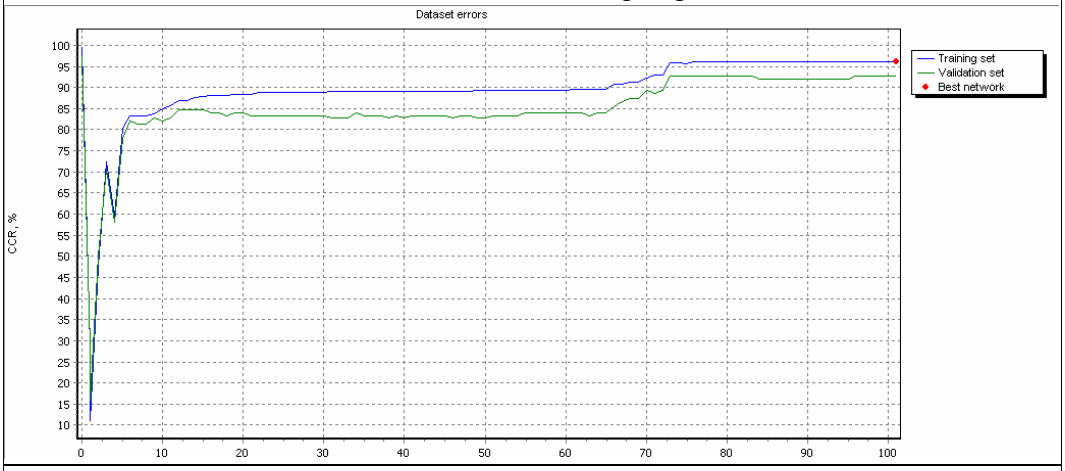
Ön işleme yöntemi: *Çevritsel dönüşüm*	Band ve sektör sayısı: 5 x 16	YSA eğitim süresi: 02:52:28

YSA metodu ve yapısı: *Conjugate Gradient Descent* 640:96:10

YSA parametreleri:
Eğitim ve doğrulama sonuçları:

	Training	Validation
CCR, %:	96,142857	92,666667
Network error:	0,003857	0
Error improvement:	6,84E-10	
Iteration:	101	
Training speed, iter/sec:	0,009768	
Architecture:	[640-96-10]	
Training algorithm:	Conjugate Gradient Descent	
Training stop reason:	All iterations done	

YSA veri kümesi hata grafiği



Ortalama doğruluk oranları (%)

Eğitim veri kümesi: 99,428571

output:	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10
S1	72	0	0	0	0	0	0	0	0	0
S2	0	70	0	0	0	0	0	0	0	0
S3	0	0	72	0	0	0	0	0	0	0
S4	0	0	0	72	0	0	0	0	0	0
S5	0	0	0	0	72	0	0	0	0	0
S6	0	0	0	0	0	68	0	0	0	2
S7	0	0	0	0	0	0	73	0	0	0
S8	0	0	0	0	0	0	0	66	0	0
S9	0	0	0	0	0	0	0	0	65	1
S10	0	0	0	1	0	0	0	0	0	66

Doğrulama veri kümesi: 97,333333

output:	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10
S1	14	0	0	0	0	0	0	0	0	0
S2	0	15	0	0	0	0	0	0	0	0
S3	0	0	14	0	0	0	0	0	0	0
S4	0	0	0	14	0	0	0	0	0	0
S5	0	0	0	0	14	0	0	0	0	0
S6	0	0	0	0	0	15	0	0	0	0
S7	0	0	0	0	0	0	12	0	0	1
S8	0	0	0	0	0	0	0	18	0	0
S9	0	0	0	1	0	0	0	0	16	0
S10	0	0	0	2	0	0	0	0	0	14

Test veri kümesi: 97,333333

output:	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10
S1	13	0	0	0	1	0	0	0	0	0
S2	0	15	0	0	0	0	0	0	0	0
S3	0	0	14	0	0	0	0	0	0	0
S4	0	0	0	14	0	0	0	0	0	0
S5	0	0	0	0	14	0	0	0	0	0
S6	0	0	1	0	0	13	0	0	0	1
S7	0	0	0	0	0	0	14	0	0	0
S8	0	0	0	0	0	0	0	16	0	0
S9	0	0	0	1	0	0	0	0	16	0
S10	0	0	0	0	0	0	0	0	0	17

Tüm veri kümesi: 98,8

output:	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10
S1	99	0	0	0	1	0	0	0	0	0
S2	0	100	0	0	0	0	0	0	0	0
S3	0	0	100	0	0	0	0	0	0	0
S4	0	0	0	100	0	0	0	0	0	0
S5	0	0	0	0	100	0	0	0	0	0
S6	0	0	1	0	0	96	0	0	0	3
S7	0	0	0	0	0	0	99	0	0	1
S8	0	0	0	0	0	0	0	100	0	0
S9	0	0	0	2	0	0	0	0	97	1
S10	0	0	0	3	0	0	0	0	0	97

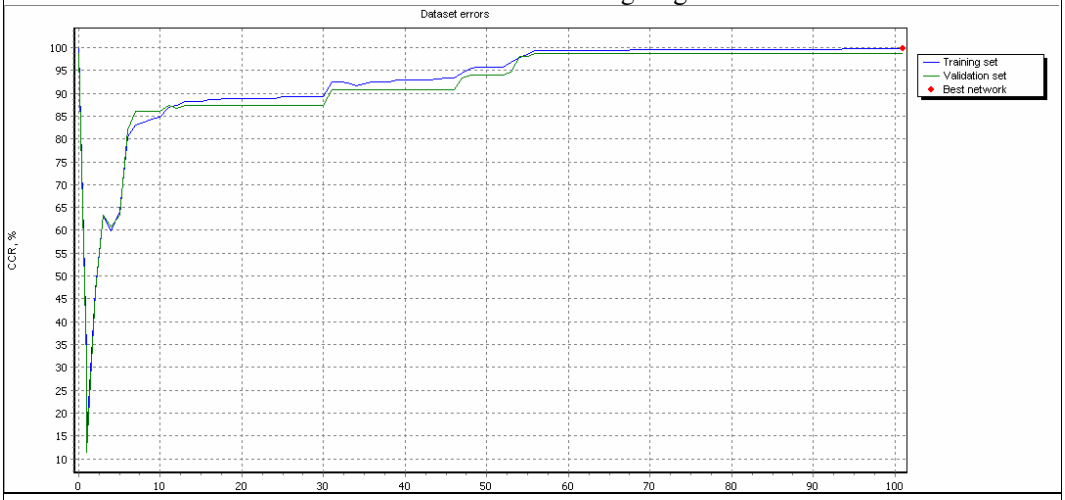
Ön işleme yöntemi: *Çevritsel dönüşüm*	Band ve sektör sayısı: 4 x 12	YSA eğitim süresi: 00:54:04

YSA metodu ve yapısı: *Conjugate Gradient Descent* 384:58:10

YSA parametreleri:
Eğitim ve doğrulama sonuçları:

	Training	Validation
CCR, %:	99,857143	98,666667
Network error:	0,000427	0
Error improvement:	7,90E-07	
Iteration:	101	
Training speed, iter/sec:	0,031164	
Architecture:	[384-58-10]	
Training algorithm:	Conjugate Gradient Descent	
Training stop reason:	All iterations done	

YSA veri kümesi hata grafiği



Ortalama doğruluk oranları (%)

Eğitim veri kümesi: 100

output:	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10
S1	70	0	0	0	0	0	0	0	0	0
S2	0	73	0	0	0	0	0	0	0	0
S3	0	0	71	0	0	0	0	0	0	0
S4	0	0	0	72	0	0	0	0	0	0
S5	0	0	0	0	72	0	0	0	0	0
S6	0	0	0	1	0	70	0	0	0	0
S7	0	0	0	0	0	0	71	0	0	0
S8	0	0	0	0	0	0	0	66	0	0
S9	0	0	0	0	0	0	0	0	68	0
S10	0	0	0	0	0	0	0	0	0	66

Doğrulama veri kümesi: 98,666667

output:	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10
S1	14	0	0	0	1	0	0	0	0	0
S2	0	13	0	0	0	0	0	0	0	0
S3	0	0	15	0	0	0	0	0	0	0
S4	0	0	0	14	0	0	0	0	0	0
S5	0	0	0	0	14	0	0	0	0	0
S6	0	0	0	0	0	15	0	0	0	0
S7	0	0	0	0	0	0	14	0	0	0
S8	0	0	0	0	0	0	0	17	0	0
S9	0	0	0	0	0	0	0	0	16	0
S10	0	0	0	0	0	0	0	1	0	16

Test veri kümesi: 98,666667

output:	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10
S1	15	0	0	0	0	0	0	0	0	0
S2	0	14	0	0	0	0	0	0	0	0
S3	0	0	14	0	0	0	0	0	0	0
S4	0	0	0	14	0	0	0	0	0	0
S5	0	0	0	0	14	0	0	0	0	0
S6	0	0	0	0	0	13	0	0	0	1
S7	0	0	0	0	0	0	14	0	0	1
S8	0	0	0	0	0	0	0	17	0	0
S9	0	0	0	0	0	0	0	0	16	0
S10	0	0	0	0	0	0	0	0	0	17

Tüm veri kümesi: 99,6

output:	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10
S1	99	0	0	0	1	0	0	0	0	0
S2	0	100	0	0	0	0	0	0	0	0
S3	0	0	100	0	0	0	0	0	0	0
S4	0	0	0	100	0	0	0	0	0	0
S5	0	0	0	0	100	0	0	0	0	0
S6	0	0	0	1	0	98	0	0	0	1
S7	0	0	0	0	0	0	99	0	0	1
S8	0	0	0	0	0	0	0	100	0	0
S9	0	0	0	0	0	0	0	0	100	0
S10	0	0	0	0	0	0	0	1	0	99

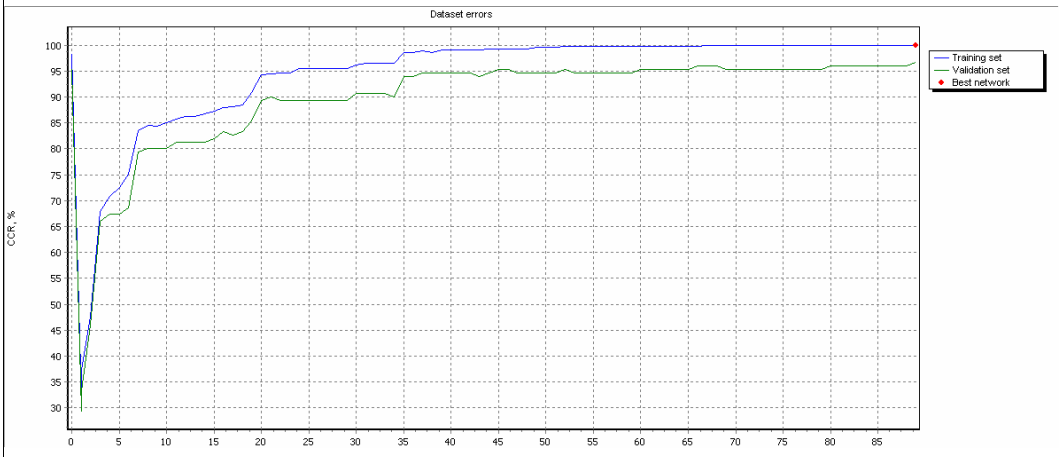
Ön işleme yöntemi: *Dalgacık dönüşüm*	Band ve sektör sayısı: 4 x 12	YSA eğitim süresi: 01:12:00

YSA metodu ve yapısı: *Conjugate Gradient Descent* 384:58:10

YSA parametreleri:
Eğitim ve doğrulama sonuçları:

	Training	Validation
CCR, %:	100	96,666667
Network error:	0,000046	0
Error improvement:	0,000097	
Iteration:	89	
Training speed, iter/sec:	0,020623	
Architecture:	[384-58-10]	
Training algorithm:	Conjugate Gradient Descent	
Training stop reason:	Desired error achieved	

YSA veri kümesi hata grafiği



Ortalama doğruluk oranları (%)

Eğitim veri kümesi: 100

output:	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10
S1	72	0	0	0	0	0	0	0	0	0
S2	0	71	0	0	0	0	0	0	0	0
S3	0	0	71	0	0	0	0	0	0	0
S4	0	0	0	72	0	0	0	0	0	0
S5	0	0	0	0	71	0	0	0	0	0
S6	0	0	0	0	0	71	0	0	0	0
S7	0	0	0	0	0	0	72	0	0	0
S8	0	0	0	0	0	0	0	67	0	0
S9	0	0	0	0	0	0	0	0	67	0
S10	0	0	0	0	0	0	0	0	0	66

Doğrulama veri kümesi: 96,666667

output:	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10
S1	13	0	0	0	1	0	0	0	0	0
S2	0	14	0	0	0	0	0	0	0	0
S3	0	0	15	0	0	0	0	0	0	0
S4	0	0	0	14	0	0	0	0	0	0
S5	0	0	0	0	13	0	0	0	0	1
S6	0	0	1	0	0	14	0	0	0	0
S7	0	0	0	0	0	0	14	0	0	0
S8	0	0	0	0	0	0	0	16	0	0
S9	0	0	0	0	0	1	0	0	16	0
S10	0	0	0	0	0	0	0	0	1	16

Test veri kümesi: 95,333333

output:	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10
S1	13	0	0	1	0	0	0	0	0	0
S2	0	15	0	0	0	0	0	0	0	0
S3	0	0	14	0	0	0	0	0	0	0
S4	0	0	0	13	1	0	0	0	0	0
S5	0	0	0	0	14	0	0	1	0	0
S6	0	0	1	0	0	13	0	0	0	0
S7	0	0	0	0	0	0	14	0	0	0
S8	0	0	0	0	1	0	0	16	0	0
S9	0	0	0	0	1	0	0	0	15	0
S10	0	0	1	0	0	0	0	0	0	16

Tüm veri kümesi: 98,8

output:	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10
S1	98	0	0	1	1	0	0	0	0	0
S2	0	100	0	0	0	0	0	0	0	0
S3	0	0	100	0	0	0	0	0	0	0
S4	0	0	0	99	1	0	0	0	0	0
S5	0	0	0	0	98	0	0	1	0	1
S6	0	0	2	0	0	98	0	0	0	0
S7	0	0	0	0	0	0	100	0	0	0
S8	0	0	0	0	1	0	0	99	0	0
S9	0	0	0	0	1	1	0	0	98	0
S10	0	0	1	0	0	0	0	0	1	98

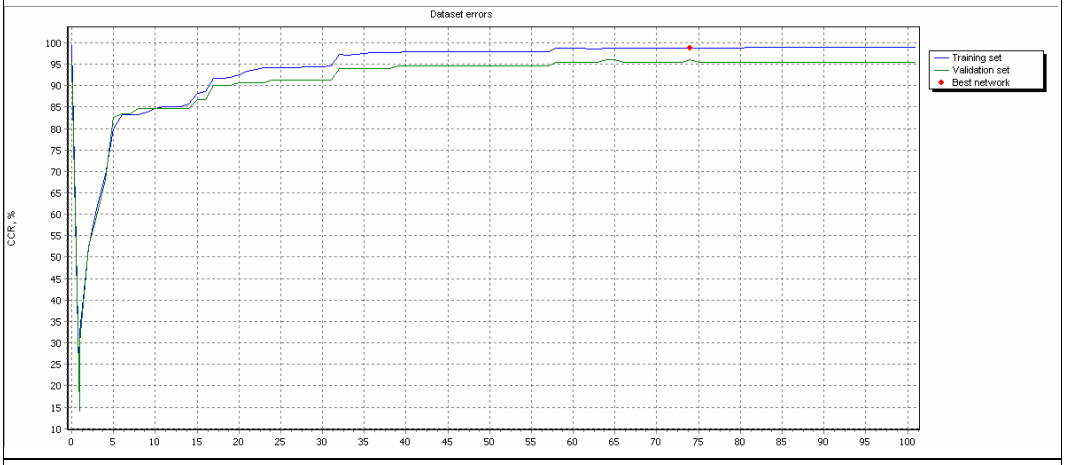
Ön işleme yöntemi: *Medyan Filtre*	Band ve sektör sayısı: 4 x 12	YSA eğitim süresi: 01:27:59

YSA metodu ve yapısı: *Conjugate Gradient Descent* 384:58:10

YSA parametreleri:
Eğitim ve doğrulama sonuçları:

	Training	Validation
CCR, %:	99	95,333333
Network error:	0,001	0
Error improvement:	0	
Iteration:	101	
Training speed, iter/sec:	0,019147	
Architecture:	[384-58-10]	
Training algorithm:	Conjugate Gradient Descent	
Training stop reason:	All iterations done	

YSA veri kümesi hata grafiği



Ortalama doğruluk oranları (%)

Eğitim veri kümesi: 99,428571

output:	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10
S1	68	0	0	0	0	0	1	0	0	3
S2	0	70	0	0	0	0	0	0	0	0
S3	0	0	73	0	0	0	0	0	0	0
S4	0	0	0	71	0	0	0	0	0	0
S5	0	0	0	0	71	0	0	0	0	0
S6	0	0	0	0	0	73	0	0	0	0
S7	0	0	0	0	0	0	70	0	0	0
S8	0	0	0	0	0	0	0	66	0	0
S9	0	0	0	0	0	0	0	0	67	0
S10	0	0	0	0	0	0	0	0	0	67

Doğrulama veri kümesi: 95,333333

output:	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10
S1	11	0	0	1	0	0	0	0	0	2
S2	0	15	0	0	0	0	0	0	0	0
S3	0	0	14	0	0	0	0	0	0	0
S4	0	0	0	14	0	0	0	0	0	0
S5	0	0	0	0	13	0	0	1	0	0
S6	0	0	0	0	0	14	0	0	0	0
S7	0	0	0	0	0	0	15	0	0	0
S8	0	0	0	0	0	0	0	17	0	0
S9	0	0	0	0	0	1	0	0	15	0
S10	0	0	0	0	0	1	0	0	1	15

Test veri kümesi: 98

output:	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10
S1	13	0	0	0	0	0	0	0	0	1
S2	0	15	0	0	0	0	0	0	0	0
S3	0	0	13	0	0	0	0	0	0	0
S4	0	0	0	15	0	0	0	0	0	0
S5	0	0	0	0	15	0	0	0	0	0
S6	0	0	0	0	0	13	0	0	0	0
S7	0	0	0	0	0	0	15	0	0	0
S8	0	0	0	0	0	0	0	17	0	0
S9	0	0	0	0	0	1	0	0	16	0
S10	1	0	0	0	0	0	0	0	0	15

Tüm veri kümesi: 98,6

output:	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10
S1	92	0	0	1	0	0	1	0	0	6
S2	0	100	0	0	0	0	0	0	0	0
S3	0	0	100	0	0	0	0	0	0	0
S4	0	0	0	100	0	0	0	0	0	0
S5	0	0	0	0	99	0	0	1	0	0
S6	0	0	0	0	0	100	0	0	0	0
S7	0	0	0	0	0	0	100	0	0	0
S8	0	0	0	0	0	0	0	100	0	0
S9	0	0	0	0	0	2	0	0	98	0
S10	1	0	0	0	0	1	0	0	1	97

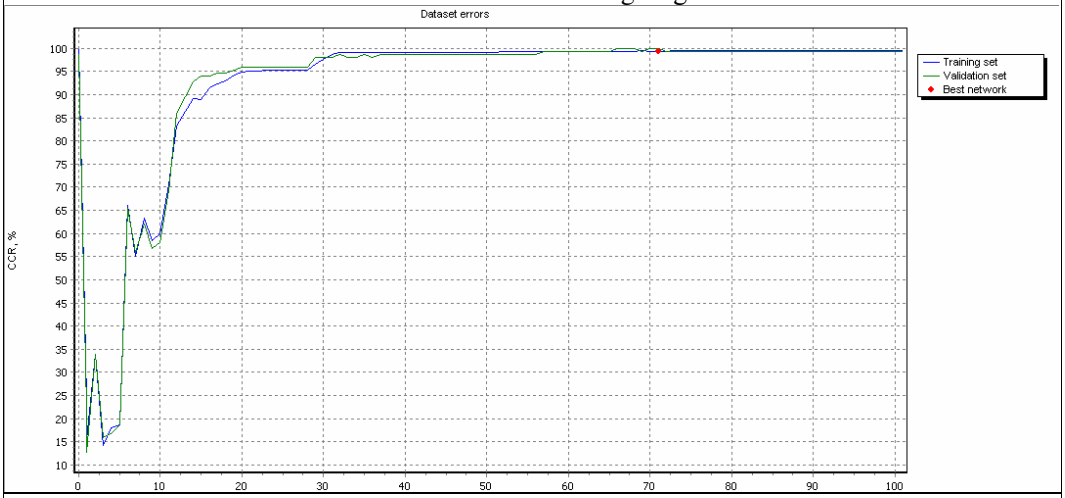
Ön işleme yöntemi: *Çevritsel dönüşüm*	Band ve sektör sayısı: 4 x 12	YSA eğitim süresi: 00:09:46

YSA metodu ve yapısı: *Quick Propagation* 384:58:10

YSA parametreleri:
Eğitim ve doğrulama sonuçları:

	Training	Validation
CCR, %:	99,571429	99,333333
Network error:	0,000433	0
Error improvement:	0,000006	
Iteration:	101	
Training speed, iter/sec:	0,173714	
Architecture:	[384-58-10]	
Training algorithm:	Quick Propagation	
Training stop reason:	All iterations done	

YSA veri kümesi hata grafiği



Ortalama doğruluk oranları (%)

Eğitim veri kümesi: 99,285714

output:	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10
S1	70	0	0	0	0	0	0	0	0	0
S2	0	73	0	0	0	0	0	0	0	0
S3	0	0	71	0	0	0	0	0	0	0
S4	0	0	0	71	0	0	0	0	0	1
S5	1	0	0	0	71	0	0	0	0	0
S6	0	0	0	1	0	70	0	0	0	0
S7	0	0	0	0	0	0	70	1	0	0
S8	0	0	0	0	0	0	0	66	0	0
S9	0	0	0	1	0	0	0	0	67	0
S10	0	0	0	0	0	0	0	0	0	66

Doğrulama veri kümesi: 100

output:	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10
S1	15	0	0	0	0	0	0	0	0	0
S2	0	13	0	0	0	0	0	0	0	0
S3	0	0	15	0	0	0	0	0	0	0
S4	0	0	0	14	0	0	0	0	0	0
S5	0	0	0	0	14	0	0	0	0	0
S6	0	0	0	0	0	15	0	0	0	0
S7	0	0	0	0	0	0	14	0	0	0
S8	0	0	0	0	0	0	0	17	0	0
S9	0	0	0	0	0	0	0	0	16	0
S10	0	0	0	0	0	0	0	0	0	17

Test veri kümesi: 96

output:	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10
S1	14	0	0	0	1	0	0	0	0	0
S2	0	14	0	0	0	0	0	0	0	0
S3	0	0	14	0	0	0	0	0	0	0
S4	0	0	0	14	0	0	0	0	0	0
S5	0	0	0	0	12	0	0	1	0	1
S6	0	0	0	0	0	13	0	0	0	1
S7	0	0	0	1	0	0	14	0	0	0
S8	0	0	0	0	0	0	0	17	0	0
S9	0	0	0	0	0	0	0	0	16	0
S10	0	0	0	0	1	0	0	0	0	16

Tüm veri kümesi: 98,9

output:	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10
S1	99	0	0	0	1	0	0	0	0	0
S2	0	100	0	0	0	0	0	0	0	0
S3	0	0	100	0	0	0	0	0	0	0
S4	0	0	0	99	0	0	0	0	0	1
S5	1	0	0	0	97	0	0	1	0	1
S6	0	0	0	1	0	98	0	0	0	1
S7	0	0	0	1	0	0	98	1	0	0
S8	0	0	0	0	0	0	0	100	0	0
S9	0	0	0	1	0	0	0	0	99	0
S10	0	0	0	0	1	0	0	0	0	99

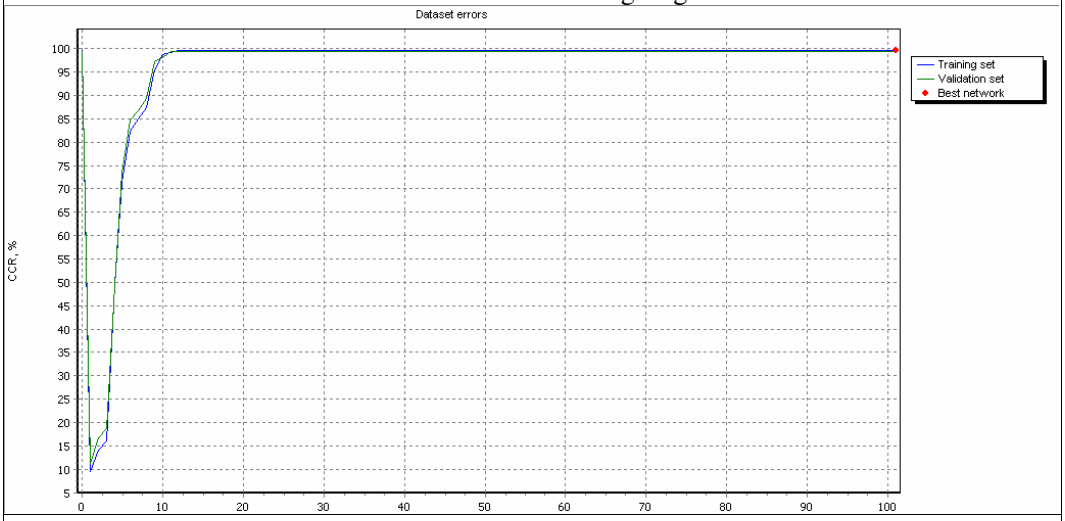
Ön işleme yöntemi: *Çevritsel dönüşüm*	Band ve sektör sayısı: 4 x 12	YSA eğitim süresi: 00:18:01

YSA metodu ve yapısı: *Online Back Propagation* 384:58:10

YSA parametreleri:
Eğitim ve doğrulama sonuçları:

	Training	Validation
CCR, %:	99,714286	99,333333
Network error:	0,000306	0
Error improvement:	2,68E-07	
Iteration:	101	
Training speed, iter/sec:	0,093707	
Architecture:	[384-58-10]	
Training algorithm:	Online Back Propagation	
Training stop reason:	All iterations done	

YSA veri kümesi hata grafiği



Ortalama doğruluk oranları (%)

Eğitim veri kümesi: 99,714286

output:	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10
S1	70	0	0	0	0	0	0	0	0	0
S2	0	73	0	0	0	0	0	0	0	0
S3	0	0	71	0	0	0	0	0	0	0
S4	0	0	0	72	0	0	0	0	0	0
S5	0	0	0	0	72	0	0	0	0	0
S6	0	0	0	0	0	70	0	0	0	1
S7	0	0	0	0	0	0	70	1	0	0
S8	0	0	0	0	0	0	0	66	0	0
S9	0	0	0	0	0	0	0	0	68	0
S10	0	0	0	0	0	0	0	0	0	66

Doğrulama veri kümesi: 99,333333

output:	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10
S1	14	0	0	0	1	0	0	0	0	0
S2	0	13	0	0	0	0	0	0	0	0
S3	0	0	15	0	0	0	0	0	0	0
S4	0	0	0	14	0	0	0	0	0	0
S5	0	0	0	0	14	0	0	0	0	0
S6	0	0	0	0	0	15	0	0	0	0
S7	0	0	0	0	0	0	14	0	0	0
S8	0	0	0	0	0	0	0	17	0	0
S9	0	0	0	0	0	0	0	0	16	0
S10	0	0	0	0	0	0	0	0	0	17

Test veri kümesi: 97,333333

output:	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10
S1	15	0	0	0	0	0	0	0	0	0
S2	0	14	0	0	0	0	0	0	0	0
S3	0	0	14	0	0	0	0	0	0	0
S4	0	0	0	14	0	0	0	0	0	0
S5	0	0	0	0	13	0	0	1	0	0
S6	0	0	0	1	0	13	0	0	0	0
S7	0	0	0	0	0	0	14	0	0	1
S8	0	0	0	0	0	0	0	17	0	0
S9	0	0	0	0	0	0	0	0	16	0
S10	0	0	0	0	0	0	1	0	0	16

Tüm veri kümesi: 99,3

output:	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10
S1	99	0	0	0	1	0	0	0	0	0
S2	0	100	0	0	0	0	0	0	0	0
S3	0	0	100	0	0	0	0	0	0	0
S4	0	0	0	100	0	0	0	0	0	0
S5	0	0	0	0	99	0	0	1	0	0
S6	0	0	0	1	0	98	0	0	0	1
S7	0	0	0	0	0	0	98	1	0	1
S8	0	0	0	0	0	0	0	100	0	0
S9	0	0	0	0	0	0	0	0	100	0
S10	0	0	0	0	0	0	1	0	0	99

ÖZGEÇMİŞ

Adem Alpaslan ALTUN, 1977 yılında Ankara'da doğdu. 1998 yılında İstanbul Üniversitesi Mühendislik Fakültesi Bilgisayar Mühendisliği Bölümünden mezun oldu. Aynı yıl Selçuk Üniversitesi Teknik Eğitim Fakültesi Elektronik ve Bilgisayar Eğitimi Bölümüne Araştırma Görevlisi olarak atandı. 2001 yılında Selçuk Üniversitesi Fen Bilimleri Enstitüsü Bilgisayar Mühendisliği Ana Bilim Dalında "Arızalı Hiperküpte sağlıklı Veri İletiminin Sağlanması" konulu çalışmasıyla yüksek lisans derecesi aldı. 2001 yılında Fen Bilimleri Enstitüsünün Elektrik-Elektronik Mühendisliği Ana Bilim Dalında doktora programına başladı.

2007 yılında Selçuk Üniversitesi Teknik Eğitim Fakültesi Elektronik ve Bilgisayar Eğitimi bölümünün Bilgisayar Sistemleri Eğitimi Ana Bilim Dalında Öğretim Görevlisi olarak görevine devam etmektedir.