

**YILDIZ TEKNİK ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ**

**PATERN TANIMA İÇİN OPTİMUM PERFORMANSLI
YENİ YSA YAKLAŞIMLARI ve DONANIMSAL
TASARIMI**

Elektronik ve Haberleşme Yük. Müh. Övünç POLAT

**FBE Elektronik ve Haberleşme Anabilim Dalı Elektronik Programında
Hazırlanan**

DOKTORA TEZİ

Tez Savunma Tarihi : 28.06.2007

Tez Danışmanı : Prof. Dr. Tülay YILDIRIM (YTÜ)

**Jüri Üyeleri : Prof. Dr. Aydın AKAN (İÜ)
Yrd. Doç. Dr. Lale ÖZYILMAZ (YTÜ)
Prof. Dr. Fikret GÜRGEN (BÜ)
Prof. Dr. Oruç BİLGİÇ (YTÜ)**

İÇİNDEKİLER

	Sayfa
SİMGE LİSTESİ.....	iv
KISALTMA LİSTESİ	v
ŞEKİL LİSTESİ.....	vi
ÇİZELGE LİSTESİ	vii
ÖNSÖZ	ix
ÖZET	x
ABSTRACT.....	xi
1. GİRİŞ	1
2. YAPAY SİNİR AĞLARI	4
2.1 Genelleştirilmiş Regresyon Sinir Ağı	6
2.2 Olasılıksal Sinir Ağı.....	8
2.3 Çok Katmanlı Algılayıcı Sinir Ağları	9
2.4 Radyal Temelli Fonksiyon Sinir Ağları.....	9
3. GENETİK ALGORİTMALAR	11
3.1 Genetik Algoritmaların Çalışma Prensipleri	11
3.2 Genetik Algoritmalarda Kullanılan Temel Operatörler	12
3.2.1 Kromozom Gösterimi	12
3.2.2 Uygunluk Fonksiyonu.....	12
3.2.3 Seçim.....	13
3.2.4 Çaprazlama	14
3.2.5 Mutasyon.....	14
3.3 Genetik Algoritmalarındaki Kontrol Parametreleri	14
4. ALAN PROGRAMLANABİLİR KAPI DİZİLERİ (FPGA–Field Programmable Gate Arrays).....	16
4.1 Programlanabilir Devre Elemanları	16
4.1.1 Programlanabilir Lojik.....	16
4.1.1.1 SPLD.....	16
4.1.1.2 CPLD	17
4.1.1.3 FPGA	17
4.2 FPGA Programlama Teknolojileri.....	18
4.3 FPGA ile gerçekleştirilecek sistemlerin tasarım aşamaları.....	19
5. YAPAY SİNİR AĞLARI İLE PATTERN TANIMA.....	21

5.1	Giriş Görüntülerinden Özellik Çıkarmaksızın 3-Boyutlu Nesne ve El Yazısı Rakam Tanıma	22
5.1.1	3-Boyutlu Nesne Tanıma	22
5.1.1.1	Metot	23
5.1.1.2	Simülasyon Sonuçları	24
5.1.2	El Yazısı Rakam Tanıma	26
5.2	3-Boyutlu Benzer Nesne Tanıma	28
5.2.1	Metot	28
5.2.2	Simülasyon Sonuçları	29
6.	PATERN TANIMA SİSTEMİNİN GENETİK ALGORİTMA İLE OPTİMİZASYONU	31
6.1	Metot	32
6.2	Simülasyonlar	34
6.2.1	3-boyutlu nesne tanıma sisteminin optimizasyonu	36
6.2.2	El yazısı rakam tanıma sisteminin optimizasyonu	37
7.	GRSA KULLANILARAK EL GEOMETRİSİ GÖRÜNTÜLERİNDEN BİOMETRİK PATERN TANIMA	41
7.1	Metot	41
7.2	GRSA için el geometrisi tanıma işlemindeki aşamalar ve sistemin doğrulama başarısı	42
7.3	Simülasyon Sonuçları	43
8.	GENELLEŞTİRİLMİŞ REGRESYON SİNİR AĞI'NIN FPGA ile DONANIMSAL GERÇEKLEMESİ	47
8.1	GRSA'nın Sayısal Donanım Tasarımı	48
8.1.1	Üs alma bloğunun tasarımı	49
8.1.2	Bölme bloğunun tasarımı	50
8.2	Simülasyon ve Sentez Sonuçları	51
	SONUÇLAR	57
	KAYNAKLAR	59
	ÖZGEÇMİŞ	66

SİMGE LİSTESİ

b_j	Bias değeri
c_i	Merkez vektörü
D	Öklid uzaklığı
F	Olasılık yoğunluk fonksiyonu
L	Yanlış sınıflama oranı
P	Sınıfların görülme olasılığı
W	Ağırlıklar
w_{ij}	i. gizli nöron ve j. çıkış nöronu arasındaki ağırlık katsayısı
x	Giriş vektörü
y	Yapay sinir ağı çıkış değeri
σ	Sigma değeri (Gauss fonksiyonunun genişliği)
ϕ	Aktivasyon fonksiyonu

KISALTMA LİSTESİ

ANN	Artificial neural network (Yapay sinir ağı)
CLB	Configurable logic block (Konfigüre edilebilir lojik blok)
CPLD	Complex Programmable Logic Device (Kompleks programlanabilir lojik eleman)
EEPROM	Electrically erasable programmable read-only memory (Elektriksel olarak silinip programlanabilen yalnızca okunur hafıza)
EER	Equal error rate (Eşit hata oranı)
EPROM	Erasable programmable read-only memory (Silinebilir programlanabilir yalnızca okunur hafıza)
FAR	False acceptance ratio (Hatalı kabul etme oranı)
FPAA	Field programmable analog array (Alan programlanabilir analog dizi)
FPGA	Field programmable gate array (Alan programlanabilir kapı dizi)
FRR	False rejection ratio (Hatalı reddetme oranı)
GA	Genetik algoritma
GRNN	General regression neural network
GRSA	Genelleştirilmiş regresyon sinir ağı
IOB	Input/output block (Giriş/çıkış bloğu)
LUT	Look-up table (Arama tablosu)
LVQ	Learning vector quantization (Vektör Kuantalamalı Öğrenme)
MLP	Multilayer Perceptron (Çok katmanlı algılayıcı)
OSA	Olasılıksal sinir ağı
PAL	Programmable array logic (Programlanabilir dizi lojik)
PC	Personel computer (Kişisel bilgisayar)
PLD	Programmable logic device (Programlanabilir lojik eleman)
PNN	Probabilistic neural network
PROM	Programmable read-only memory (Programlanabilir yalnızca okunur hafıza)
RAM	Random access memory (Rastgele erişilebilir hafıza)
RBF	Radial basis function (Radyal temelli fonksiyon)
SOM	Self organizing map (Kendi kendine organize olan harita)
SPLD	Simple Programmable Logic Device (Basit programlanabilir lojik eleman)
VHDL	Very high speed integrated circuit hardware description language (Çok hızlı tümleşik devre donanım tanımlama dili)
VLSI	Very large-scale integrated (Büyük ölçekli tümleşik devre)
YSA	Yapay Sinir Ağları

ŞEKİL LİSTESİ

Sayfa

Şekil 2.1	Genelleştirilmiş regresyon sinir ağı yapısı.	8
Şekil 2.2	Çok katmanlı algılayıcı yapay sinir ağı yapısı.....	9
Şekil 3.1	Genetik Algoritmanın Akış Diyagramı	12
Şekil 3.2	Rulet çarkı.....	13
Şekil 4.1	FPGA genel mimarisi.	18
Şekil 5.1	YSA'lar ile nesne tanıma sisteminin blok diyagramı.	23
Şekil 5.2	Nesne tanıma işleminde GRSA'nın katmanlarındaki işlemler.	24
Şekil 5.3	Tanıma sisteminin simülasyonunda kullanılan 10 nesne.....	24
Şekil 5.4	Veri kümesindeki 6. nesnenin 0° den 25° ye kadar olan poz görüntüleri.	25
Şekil 5.5	NIST veri setindeki el yazısı rakamlardan bazıları.....	27
Şekil 5.6	Nesne tanıma sistemi.	28
Şekil 5.7	Simülasyon için kullanılan benzer 8 nesne.....	29
Şekil 5.8	Veri kümesindeki 2 numaralı oyuncak arabanın 0° deki pozunun RGB görüntüleri.....	29
Şekil 6.1	Sınıflandırma amacıyla kullanılan GRSA'nın GA ile optimizasyonunun akış diyagramı.....	33
Şekil 6.2	GRSA ile tiroid verisi sınıflandırma uygulamasında genetik algoritma ile yapılan optimizasyon sonuçları.....	35
Şekil 6.3	El yazısı rakam tanıma uygulaması için genetik algoritma ile yapılan optimizasyon sonuçları.	38
Şekil 7.1	El görüntü paternlerinden bazı örnekler.	42
Şekil 7.2	GRSA ile el geometrisi tanıma sistemi.....	42
Şekil 7.3	Farklı σ değerleri için FAR ve FRR değişimi.	45
Şekil 7.4	Yetkili kullanıcı sayısının artırılması durumunda farklı σ değerleri için FAR ve FRR değişimi.....	45
Şekil 8.1	GRSA'nın işlemsel akış diyagramı	49
Şekil 8.2	İris verisi için yapılan örnek bir test işleminin zaman diyagramı.....	54
Şekil 8.3	Tiroid verisi için yapılan örnek bir test işleminin zaman diyagramı.	54

ÇİZELGE LİSTESİ

Sayfa

Çizelge 2.1 Farklı tip YSA'ların görüntü işleme ve tanıma uygulamalarındaki kullanımı.	6
Çizelge 5.1 Nesne tanıma uygulaması için farklı YSA'nın eğitime ve test doğruluk oranları.	25
Çizelge 5.2 OSA ve GRSA için referans görüntülerin sayısı ile test setinin tanıma oranının değişimi.	26
Çizelge 5.3 OSA ile herbir nesnenin tanıma oranı.	26
Çizelge 5.4 GRSA için herbir rakamın tanıma oranı.	27
Çizelge 5.5 OSA için herbir rakamın tanıma oranı.	27
Çizelge 5.6 Benzer nesne tanıma uygulamasında GRSA ve OSA için referans görüntülerin sayısı ile test setinin tanıma oranının değişimi.	30
Çizelge 6.1 Genetik algoritmadaki parametrelerin sayısı	34
Çizelge 6.2 Test aşamasında her bir sınıf için doğru sınıflanan örnek sayıları.	36
Çizelge 6.3 Ortalama test başarımları.	36
Çizelge 6.4 3-Boyutlu nesne tanıma uygulaması için GRSA ve GRSA+GA'nın test başarımları.	37
Çizelge 6.5 3-Boyutlu nesne tanıma uygulaması için OSA ve OSA+GA'nın test başarımları.	37
Çizelge 6.6 GRSA ile herbir rakamın tanınma oranı.	38
Çizelge 6.7 GRSA'nın GA ile optimizasyonu sonucunda herbir rakamın tanınma oranı.	38
Çizelge 6.8 Optimizasyon aşamasında kullanılmayan farklı rakamlar için GRSA optimize edilmeden önce herbir rakamın tanınma oranları.	39
Çizelge 6.9 Optimizasyon aşamasında kullanılmayan farklı rakamlar için GRSA optimize edildikten (GRSA+GA) sonra herbir rakamın tanınma oranları.	39
Çizelge 6.10 OSA ile herbir rakamın tanınma oranı.	39
Çizelge 6.11 OSA'nın GA ile optimizasyonu sonucunda herbir rakamın tanınma oranı.	40
Çizelge 6.12 Optimizasyon aşamasında kullanılmayan farklı rakamlar için OSA optimize edilmeden önce herbir rakamın tanınma oranları.	40
Çizelge 6.13 Optimizasyon aşamasında kullanılmayan farklı rakamlar için OSA optimize edildikten (OSA+GA) sonra herbir rakamın tanınma oranları.	40
Çizelge 7.1 Farklı YSA yapıları için eğitime, test ve doğrulama performansları.	43
Çizelge 7.2 Farklı σ değerleri için GRSA'nın eğitime, test ve doğrulama performansları	44
Çizelge 7.3 Farklı σ değerleri için FAR ve FRR sonuçları.	44
Çizelge 7.4 Yetkili kullanıcı sayısının artırılması durumunda farklı σ değerleri için GRSA'nın eğitime, test ve doğrulama performansları.	46

Çizelge 8.1 İris verisi için Matlab ve FPGA gerçeklemelerinin test performansları.....	51
Çizelge 8.2 Tiroid verisi için Matlab ve FPGA gerçeklemelerinin eğitime ve test performansları.	52
Çizelge 8.3 3-boyutlu benzer nesne tanıma uygulaması için Matlab ve FPGA gerçeklemelerinin eğitime ve test performansları.	53
Çizelge 8.4 30 eğitime vektörü ve 4 giriş için donanım gereksinim sonuçları.....	55

ÖNSÖZ

Bu tez çalışmasında, gelişen teknolojiyle birlikte önemi giderek artan patern tanıma sistemleri, yapay sinir ağları ile hem yazılımsal hem donanımsal olarak tasarlanmıştır.

Başta, bu çalışmanın hazırlanmasında destek ve yardımlarını esirgemeyen, beni doğru yönlendiren ve yapay sinir ağları konusunda eğitmenim olan değerli danışman hocam Prof. Dr. Tülay Yıldırım'a, çalışmamın gerçekleştirilmesinde öneri ve desteklerini esirgemeyen değerli hocalarım Prof. Dr. Aydın Akan ve Yrd. Doç. Dr. Lale Özyılmaz'a, destek ve katkılarından dolayı elektronik yüksek mühendisi olan arkadaşım Abid Üveys Danış'a ve araştırma görevlisi arkadaşlarım Nurhan Türker Tokan, Nergis Tural, Evren Cesur ve Ahmet Serbes'e içtenlikle teşekkür ederim. Ayrıca Elektronik ve Haberleşme Mühendisliği Bölümü'nün bütün öğretim eleman ve çalışanlarına teşekkürlerimi sunarım.

Ayrıca bugünlerimi borçlu olduğum aileme sonsuz teşekkürler ederim.

ÖZET

Bu çalışmada yapay sinir ağları kullanılarak patern tanıma uygulamalarına farklı bir yaklaşım sunulmaktadır. Çalışmada özellikle Radyal Temelli Fonksiyon yapay sinir ağlarının bir çeşidi olan Genelleştirilmiş Regresyon Sinir Ağları ve Olasılıksal Sinir Ağları yapıları kullanılarak farklı tür patern tanıma uygulamaları gerçekleştirilmiş ve tasarlanan patern tanıma sistemlerinin test performansları Genetik Algoritma ile optimize edilerek arttırılmıştır. Daha sonra da tanıma sistemlerinde oldukça başarılı sonuçlar veren Genelleştirilmiş Regresyon Ağ Yapısının “Alan programlanabilir kapı dizileri” ile donanımsal tasarımı gerçekleştirilmiştir.

Tezde patern tanıma uygulamaları olarak; giriş görüntülerine herhangi bir ön işleme ve özellik çıkarma işlemi uygulamadan 3-boyutlu nesne tanıma, el yazısı rakam tanıma ve el geometrisi paternlerinden kişi tanıma ve 3-boyutlu benzer nesnelerin özellik çıkarılması suretiyle tanınması işlemleri gerçekleştirilmiş ve yüksek test başarıları elde edilmiştir. Patern tanıma sistemlerinin optimizasyonu aşamasında ise literatürde sınıflayıcı uygulamalarında yaygın olarak kullanılan iris ve tiroid verileri sınıflandırma, 3-boyutlu nesne tanıma ve el yazısı rakam tanıma problemleri ele alınmış ve bütün uygulamalarda sistemlerin test performanslarının arttığı gözlemlenmiştir. Tezde ayrıca Genelleştirilmiş Regresyon Ağ Yapısı bir donanım tanımlama dili olan VHDL (Very High Speed Integrated Circuit Hardware Description Language) yardımıyla tasarlanmış ve sistem iris ve tiroid verileri ile 3-boyutlu benzer nesnelerin sınıflandırılması problemleri için denenmiş ve yüksek sınıflama başarıları elde edilmiştir.

Anahtar Kelimeler: Yapay sinir ağları, genelleştirilmiş regresyon sinir ağları, olasıksal sinir ağları, patern tanıma, nesne tanıma, genetik algoritmalar, optimizasyon, alan programlanabilir kapı dizileri.

ABSTRACT

This study presents an approach to recognize patterns using artificial neural networks (ANN). Pattern recognition applications are particularly realized using the General Regression Neural Networks (GRNN) which is a type of radial basis function networks and the Probabilistic Neural Networks (PNN). The test performances of the designed pattern recognition systems were optimized using genetic algorithm. Then, a GRNN hardware based on Field Programmable Gate Arrays has been designed.

In this study, a model was improved for pattern recognition without preprocessing and feature extraction by using ANNs and the system was tested for 3D object recognition, handwritten digit recognition and hand geometry identification. In addition, another model was improved for pattern recognition with feature extraction by GRNN and PNN and this system was tested for similar 3D object recognition. High recognition rate has been obtained for these applications. For optimization of pattern classification system, different datasets were used to test the system. The effectiveness of design was examined by the well known iris and thyroid data classification, and then 3D objects and handwritten digit recognition. Experimental results show that the genetically optimized GRNN and PNN provides higher recognition ability compare with that of unoptimized GRNN and PNN for all of applications. In hardware implementation of GRNN, the parts of the structure of GRNN designed using VHDL (Very High Speed Integrated Circuit Hardware Description Language). The developed system was carried out for different datasets such as iris plant, thyroid disease and similar-3D object recognition and high classification rate has been obtained.

Keywords: Artificial neural networks, general regression neural networks, probabilistic neural networks, pattern recognition, object recognition, genetic algorithms, optimization, field programmable gate arrays.

1. GİRİŞ

Genel anlamda Yapay Sinir Ağları (YSA), beynin bir işlevi yerine getirme yöntemini modellemek için tasarlanan sistemler olarak tanımlanabilir. YSA'lar, paralel dağılmış yapısı, öğrenebilme ve genelleme yapma özellikleri sayesinde karmaşık ve çözümlenmesi güç problemleri çözebilme yeteneğine sahiptir. Patern sınıflandırma, işaret işleme, sistem tanımlama ve denetimi gibi birçok mühendislik alanında YSA'lar oldukça başarılı sonuçlar vermektedir.

Patern tanıma sistemleri, günümüzde teknolojinin hızla ilerlemesiyle birlikte ağırlıklı olarak endüstri, tıp ve askeri alanlarda oldukça sık olarak kullanılmakta ve tanıma sistemlerinin önemi de giderek artmaktadır. Bu uygulamalara örnek olarak nesne, el yazısı karakter, yüz, parmak izi ve ses tanıma, biyomedikal veri sınıflama ve hedef tanıma çalışmaları verilebilir. Günümüzde patern tanıma sistemlerinin gerçek zamanlı uygulamalarda doğru bir şekilde çalışmasının sağlanması için çalışmalar yapılmaktadır. Örneğin 2 boyutlu patern tanıma uygulamaları için gerçekleştirilecek sistemlerde, görüntü işleme algoritmalarının uygulanması için çok hızlı ve çok büyük bellekli bilgisayarlara ihtiyaç duyulmaktadır. YSA'lar paralel yapısı nedeniyle büyük ölçekli tümleşik devre (VLSI) teknolojisi ile gerçekleştirilebilmektedir. Bu özellik, YSA'nın hızlı veri işleme yeteneğini artırmakta ve gerçek zamanlı uygulamalarda tercih edilmesine sebep olmaktadır.

Yapay sinir ağlarının yukarıda bahsedilen yetenek ve avantajlarından dolayı, bu çalışmada özellikle Genelleştirilmiş Regresyon Sinir Ağı (GRSA) ve Olasılıksal Sinir Ağı (OSA) yapıları ile patern tanıma uygulamaları gerçekleştirilmiş ve tasarlanan patern tanıma sistemlerinin performansları Genetik Algoritma ile optimize edilerek arttırılmıştır. Daha sonra ise tanıma sistemlerinde oldukça başarılı sonuçlar veren GRSA yapısının, "Alan programlanabilir kapı dizileri" ile donanımsal tasarımı gerçekleştirilmiştir.

Bu çalışmada gerçekleştirilmiş olan 2 boyutlu patern tanıma uygulamalarının çoğu giriş görüntülerine herhangi bir ön işleme ve özellik çıkarma işlemi uygulamadan gerçekleştirilmiştir. Elde edilen sonuçlar, önerilen sistemin görüntülerden özellik çıkarmanın karmaşık ve güç olduğu tanıma uygulamaları için kullanılabileceğini göstermektedir.

Tezin ikinci, üçüncü ve dördüncü bölümlerin de sırasıyla Yapay Sinir Ağları, Genetik Algoritma ve Alan Programlanabilir Kapı Dizileri ile ilgili ön bilgiler verilmiştir.

Çalışmanın beşinci bölümünde öncelikle 3 boyutlu nesne tanıma ve el yazısı rakam tanıma uygulamaları giriş görüntülerine herhangi bir ön işleme ve özellik çıkarma işlemi

uygulamadan YSA'lar kullanılarak gerçekleştirilmiştir. Daha sonra ise 3 boyutlu benzer nesnelerin tanınması uygulaması giriş görüntülerinden özellik çıkarılması suretiyle GRSA ve OSA kullanılarak gerçekleştirilmiştir.

Altıncı bölümde, gerçekleştirilmiş patern tanıma uygulamalarında kullanılan GRSA ve OSA yapılarının optimizasyonları genetik algoritma ile gerçekleştirilmiştir. Bu aşamada özellikle tasarlanan patern tanıma sistemlerinde optimum referans kümenin seçimi işlemi yapılmıştır.

Yedinci bölümde, biometrik bir patern olan el geometrisi görüntülerinden kişi tanıma uygulaması paternlere herhangi bir ön işleme ve özellik çıkarma işlemi yapmadan GRSA ile gerçekleştirilmiştir ve sistemin performansı False Acceptance Ratio (FAR) ve False Rejection Ratio (FRR) değerleri hesaplanarak gösterilmiştir.

Son bölümde ise önceki bölümlerde yazılımsal olarak gerçekleştirilmiş ve patern tanıma uygulamalarında oldukça başarılı sonuçlar veren GRSA'nın donanımsal tasarımı, Alan programlanabilir kapı dizileri (FPGA–Field Programmable Gate Arrays) ile gerçekleştirilmiştir. Bu aşamada tasarlanacak sistem içerisindeki birimler bir donanım tanımlama dili olan VHDL yardımıyla ifade edilmiştir. Sistemin tanımlanma aşamasını takiben derleme ve davranışsal benzetim adımları gerçekleştirilmiştir. Sistemden beklenen cevapların elde edilmesiyle beraber sentezleme işlemi gerçekleştirilmiş ve GRSA'nın sayısal olarak gerçekleştirmesindeki donanım gereksinimleri bulunmuştur.

Çalışmadaki orjinallikler şu şekilde özetlenebilir;

3 boyutlu nesne tanıma uygulamasında giriş görüntülerine herhangi bir ön işleme ve özellik çıkarma işlemi uygulamadan ve az sayıda referans veri kullanılarak GRSA ve OSA ile yüksek başarı elde edilmektedir. Gerçek zamanlı patern tanıma uygulamalarında karşılaşılan en büyük güçlüklerden biri olan özellik çıkarma işlemine gerek kalmadan ve özellikle nesne tanıma uygulamasının yüksek tanıma oranı elde edilerek gerçekleştirilebilmesi önerilen sistemin en büyük avantajı olmaktadır.

GRSA ve OSA kullanılarak geliştirilen özellik çıkarmaksızın patern tanıma sistemlerinin performanslarını artırmak için genetik algoritma ile optimum referans kümenin seçimi ve ağ yapılarının içerisindeki Gauss fonksiyonun genişliğinin belirlenmesi işlemi yapılmıştır. Literatürde yapılan GRSA ve OSA gibi radyal temelli fonksiyona sahip ağ yapılarının optimizasyonun da genel olarak gerçekleştirilen işlem çoğunlukla optimum Gauss fonksiyon genişliğinin belirlenmesidir. Bu tez çalışmasında sınıflandırma için kullandığımız ve yüksek test başarısı elde ettiğimiz GRSA ve OSA yapılarının test performanslarını maksimum

yapacak, radyal fonksiyon genişliğinin değeri ile birlikte veri kümelerinde mevcut olan özellik vektörlerinin hangilerinin referans olarak alınacağını belirlenmektedir. Mevcut merkezlerden farklı optimal merkezler bulma işlemi, özellikle sınıflandırma problemleri için çok fazla özelliğin çıkartıldığı uygulamalarda veya bizim çalışmamızdaki gibi iki boyutlu verilerden hiçbir özellik çıkarma işlemi yapmadan gerçekleştirilecek olan tanıma uygulamalarında çok fazla miktarda optimize edilecek parametre çıkmasına neden olacak ve bu da algoritmada hız problemi yaratacaktır. Bundan dolayı gerçekleştirilen sınıflandırma uygulamalarında optimize edilecek olan parametreler; mevcut veri kümesinde bulunan bütün özellik vektörlerinin içinden seçilecek olan ve her bir sınıf için bizim belirleyeceğimiz referans veri sayısı ile Gauss fonksiyonun genişliği olarak seçilmiştir. Önerilen algoritma; hem farklı veri kümeleri hem de farklı ağ yapıları için denenmiş ve bütün uygulamalarda test performansını arttırdığı gözlenmiştir.

Gerçekleştirilen el geometrisi paternlerinden kişi tanıma uygulaması da, yine paternlere herhangi bir ön işleme ve özellik çıkarma işlemi yapmadan GRSA ile gerçekleştirilmiştir ve tasarlanan sistemin yüksek bir performans gösterdiği bulunmuştur. Aynı uygulama farklı YSA'lar ile denenmiş olup sadece GRSA için yüksek doğrulama başarısı gözlemlenmiştir. Literatürde yapılan el paterni tanıma çalışmalarında giriş görüntülerine filtreleme, kenar belirleme gibi ön işlemler yapılmakta ve daha sonra da parmakların uzunluğu, genişliği gibi pek çok özellik çıkarılarak tanıma işlemi gerçekleştirilmektedir. Buna göre gerçekleştirdiğimiz yöntemin en büyük avantajı bu işlemlere gerek kalmadan yüksek performans gösterebilmesidir.

Gerçekleştirdiğimiz patern tanıma uygulamalarında oldukça başarılı sonuçlar veren GRSA'nın, alan programlanabilir kapı dizileri ile donanımsal tasarımı gerçekleştirilmiştir. Literatürde, GRSA'nın FPGA gerçeklemesine yönelik sadece bir çalışmaya rastlanmıştır. Çalışmada VHDL kodu içerisine, Matlab yazılımı içindeki GRSA kodunun çevirisi gerçekleştirilmiştir. Ayrıca yapılan çalışma gerçek bir problem üzerine denenmemiş olup, sadece Matlab ve VHDL simülasyonları ile elde edilen sonuçlar arasındaki fark karşılaştırılması yapılmıştır. Bu tez çalışmasında; GRSA yapısı içerisindeki; Öklid mesafesi, Gauss fonksiyonu ve bölme blokları, sabit-noktalı aritmetik kullanılarak sentezlenebilir VHDL kodu ile gerçekleştirilmiş ve simülasyon aşamasında tasarlanan sistemin performansı literatürde sınıflayıcı uygulamalarında sıklıkla kullanılan farklı veri kümeleri için denenmiştir.

2. YAPAY SİNİR AĞLARI

Yapay Sinir Ağları (YSA), temel olarak biyolojik sinir sisteminin çalışma şeklini simüle etmek için tasarlanan sistemlerdir. Simüle edilen sistem sinir hücreleri (nöronlar) içerir ve bu nöronlar çeşitli şekillerde birbirlerine bağlanarak ağı oluştururlar. Bu ağlar öğrenme, hafızaya alma ve giriş verileri arasındaki ilişkiyi ortaya çıkarma yeteneğine sahiptirler. Bu yetenekleri sayesinde YSA'lar bir insan hafızasının düşünme ve gözlemlene gibi doğal yeteneklerinin gerekeceği problemlere çözüm üretebilmektedirler.

Biyolojik sistemlerde öğrenme işlemi, nöronlar arasındaki sinaptik bağlantıların ayarlanması ile olmaktadır. Yani, insanlar doğumlarından başlayarak bir “yaşayarak öğrenme” süreci içerisine girerler ve bu süreç içinde beyin sürekli bir gelişme göstermektedir. Yaşayıp tecrübe ettikçe sinaptik bağlantılar ayarlanır ve hatta yeni bağlantılar oluşur. Bu sayede öğrenme gerçekleşmektedir. Bu durum YSA'lar için de geçerlidir. Öğrenme, giriş/çıkış verilerinin işlenmesiyle, yani eğitme algoritmasının bu verileri kullanarak sinapsların bağlantı ağırlıklarını bir yakınsama sağlanana kadar iteratif olarak ayarlamasıyla gerçekleşmektedir (Haykin, 1994).

YSA'lar birbirlerine bağlanmış ve bağlantıları ağırlıklandırılmış çok sayıda işlem elemanından (nöronlardan) oluşan matematiksel sistemlerdir ve geleneksel işlemcilerden farklı şekilde işlem yapmaktadırlar. Geleneksel işlemcilerde, tek bir merkezi işlem elemanı her işlemi sırasıyla gerçekleştirir. YSA'lar da ise en basit şekilde, bir işlem elemanı, bir girdiyi bir ağırlık kümesi ile ağırlıklandırır, doğrusal olmayan bir şekilde dönüşümünü sağlar ve bir çıkış değeri oluşturur. İlk bakışta, işlem elemanlarının çalışma şekli basit gözükabilir. Ancak YSA'lardaki sinirsel hesaplamanın gücü, toplam işlem yükünü paylaşan işlem elemanlarının birbirleri arasındaki yoğun bağlantı yapısından gelmektedir.

Mimari, öğrenme yöntemi, bağlantı stratejisi vb. farklılıklara sahip çok sayıda YSA yapısı vardır. Genel olarak, YSA'lar üç ana kritere göre sınıflandırılmaktadırlar. Bu kriterlerden biri öğrenme yöntemidir. Temel olarak iki çeşit öğrenme algoritması vardır, bunlar: Eğiticiyi öğrenme ve eğiticiyi öğrenmedir. Ancak her yöntemin kullandığı öğrenme kuralı değişebilmektedir. İkinci bir sınıflandırma, ağı kullandığı veriye göre yapılmaktadır ve temel olarak; *niteliksel* ve *niceliksel* olmak üzere iki tür veri vardır. Diğer bir sınıflandırma kriteri ise ağı yapısıdır. Bazı ağlar ileri besleme şeklinde yapılandırılırken, bazı ağlar ise geri beslemeli bir yapıdadır. İleri beslemeli ağlarda, işlem elemanları arasındaki bağlantılar bir

döngü oluşturmazlar ve bu ağlar giriş verisine genellikle hızlı bir şekilde karşılık üretirler. Geri beslemeli ağlarda ise bağlantılar döngü içerirler ve döngü sebebiyle girdinin karşılığı yavaş bir şekilde oluşur, ayrıca her seferinde yeni veri kullanabilmektedirler. Literatürde, hem ileri beslemeli hem de geriye yayılma özelliğine sahip ağ yapıları da mevcuttur. (Haykin, 1994), (Zurada, 1992)

YSA çeşitleri arasında en çok bilinen ve kullanılan ağlar geri yayılma ile eğitilen Çok Katmanlı Algılayıcı (Multilayer Perceptron), Radyal Tabanlı Fonksiyon (Radial Basis Function) ve Hopfield ağları sayılabilir.

Literatürde mevcut olan 100'den fazla yapay sinir ağı modellerinin bazıları, görüntü işleme, nesne tanıma, el yazısı karakter tanıma, parmak izi ve yüz tanıma gibi uygulamalarda sık olarak kullanılmaktadır. Çizelge 2.1'de farklı tip YSA'ların görüntü işleme ve tanıma uygulamalarındaki kullanım alanlarını göstermektedir (M.Egmont vd, 2002).

Çok fazla çeşit ve yoğun bir literatür olması nedeniyle, burada tüm ağ çeşitleri hakkında bilgi verilmemektedir. Bunun yerine, bu çalışmada en çok kullanılan radyal temelli fonksiyon ağlarının bir çeşidi olan Genelleştirilmiş Regresyon Sinir Ağları ve Olasılıksal Sinir Ağları ile, gerçekleştirilen simülasyonlarda GRSA ve OSA ile performans karşılaştırmalarının yapıldığı Çok Katmanlı Algılayıcı Sinir Ağları ve Radyal Temelli Fonksiyon Sinir Ağları hakkında bilgi verilecektir.

Çizelge 2.1 Farklı tip YSA'ların görüntü işleme ve tanıma uygulamalarındaki kullanımı.

YSA tipi	Görüntü Restorasyonu	Görüntü İyileştirme	Görüntü Sıkıştırma	Özellik Çıkarma	Piksel Temelli Segmentasyon	Özellik Temelli Segmentasyon	Piksel Temelli Obje Tanıma	Özellik Temelli Obje Tanıma	Optimizasyon
İleri beslemeli regresyon	*	*	*	*		*			*
İleri beslemeli oto-çağrışım			*	*			*		
İleri beslemeli sınıflandırma		*			*	*	*	*	
Algılayıcı (Perceptron)				*					
Radyal Tabanlı Fonksiyon			*		*	*			
Kendi kendine organize olan harita (SOM-Self organizing map)			*	*	*	*	*	*	
Vektör Kuantalamalı Öğrenme (Learning vector quantization-LVQ)			*			*			
Hopfield	*	*		*	*	*	*	*	*
Hücreli Sinir Ağları (Cellular Neural Network-CNN)	*				*	*			
Adaptif rezonans teorisi (Adaptive resonance theory-ART)		*					*		
Bulanık (Fuzzy) Nöral/Nöral Bulanık	*			*	*		*	*	
Diğer	*	*	*	*	*	*	*	*	*

2.1 Genelleştirilmiş Regresyon Sinir Ağı

Radyal Tabanlı Fonksiyon (Radial Basis Function-RBF) sinir ağına bir çeşidi olan GRSA modeli, Specht (D.F. Specht, 1991a) tarafından geliştirilmiştir. Ağı eğitme işlemi oldukça hızlıdır. Ağ yapısının basitliği nedeniyle görüntü işlemenin de dahil olduğu pek çok uygulamada geniş kullanım alanı bulunmaktadır. Genelleştirilmiş Regresyon Sinir Ağları özellikle sistem modelleme ve kestirimi, fonksiyon yaklaşırma ve sınıflandırma uygulamalarına oldukça başarılı sonuçlar veren ileri beslemeli yapay sinir ağlarıdır.

GRSA şekil 2.1'de gösterildiği gibi dört katmandan oluşur. İlk katman giriş katmanıdır ve ikinci katmana tam olarak bağlanmıştır. Giriş katmanı, n boyutlu x değişkenlerinden oluşan

giriş vektörünü örnek katmanına (ikinci katmana) iletir. İkinci katmanda her bir giriş örneğini temsil eden bir nöron vardır ve p adet $1 \times n$ boyutundaki eğitme yani referans vektörlerini depolar. Bu katmanda RBF ağlarındaki ilk katmanın işlevine benzer bir işlev gerçekleştirir; girişler ve depolanmış örnekler arasındaki uzaklık ölçütünü hesaplayarak çıkış üretirler. İkinci katmanda σ genişlik katsayısına sahip Gauss fonksiyonu kullanılır. Bu katmanda yeni giriş vektörü ile her bir referans vektör arasındaki Öklid uzaklığı (D) ve W ağırlıkları aşağıdaki şekilde hesaplanmaktadır.

$$D(x_1, x_2) = \sum_{k=1}^n \left(\frac{x_{1k} - x_{2k}}{\sigma_k} \right)^2 \quad (2.1)$$

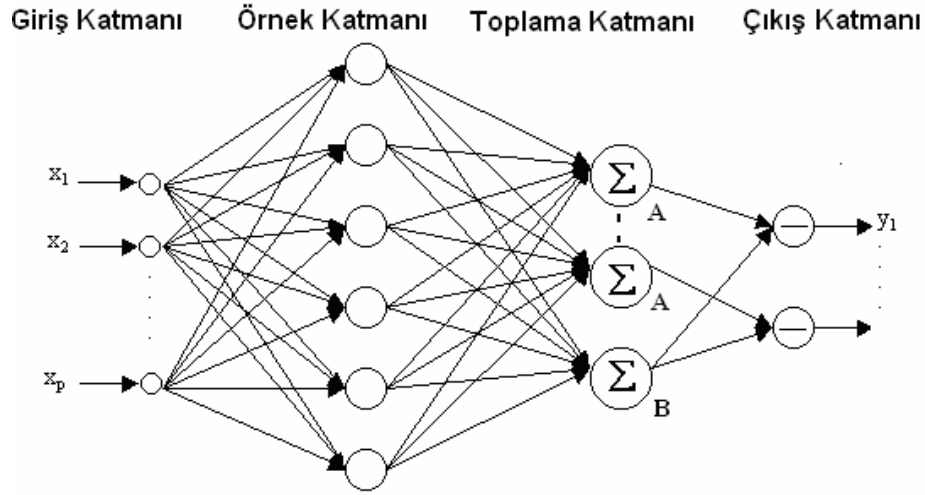
$$W(x, x^i) = e^{-D(x, x^i)} \quad (2.2)$$

Burada; sigma değeri (σ_k) ayarlanabilir bir parametre olarak Gauss fonksiyonun genişliğini, x; yeni giriş vektörünü, x^i ise depolanan i. referans vektörü göstermektedir.

GRSA de üçüncü katman toplama katmanıdır, A-toplama ve B-toplama nöronlarından oluşur. A-toplama nöronları ikinci katmanın çıkışlarının ağırlıklı (her bir depolanmış örnek için verilen hedef değerle ikinci katmanın çıkışının çarpılması) toplamını gerçekleştirir. B-toplama nöronları ise bu çıkış değerlerini ağırlıksız olarak toplar. Dördüncü katmandaki çıkış nöronlarının her birine bağlı sadece bir A-toplama nöronu ve hepsine ortak olarak bağlantılı bir adet B-toplama nöronu vardır. Çıkış katmanında ise her bir A-toplamı B-toplamına bölünerek çıkışlar elde edilir. Ağırlık çıkış değeri ise aşağıdaki gibi hesaplanır (Heimes vd, 1998), (Zhanwei vd, 2003).

$$y_j = \frac{\sum_{i=1}^p y_{ij} W(x, x^i)}{\sum_{i=1}^p W(x, x^i)} \quad j = 1, 2, \dots, m \quad (2.3)$$

Burada p; eğitme vektörlerinin sayısını, y_{ij} ; her bir depolanmış örnek için verilen hedef değeri göstermektedir.



Şekil 2.1 Genelleştirilmiş regresyon sinir ağı yapısı.

2.2 Olasılıksal Sinir Ağı

Olasılıksal sinir ağları Bayes-Parzen kestiriciler olarak da bilinir. K_1 ve K_2 sınıflarından birine ait, m -boyutlu bir x vektörü olsun. K_1 ve K_2 sınıflarına ait olasılık yoğunluk fonksiyonları $F_1(x)$ ve $F_2(x)$ olsun. Bayes Teoremi'ne göre x vektörü,

$$\frac{F_1(x)}{F_2(x)} > \frac{L_1 P_2}{L_2 P_1} \quad (2.4)$$

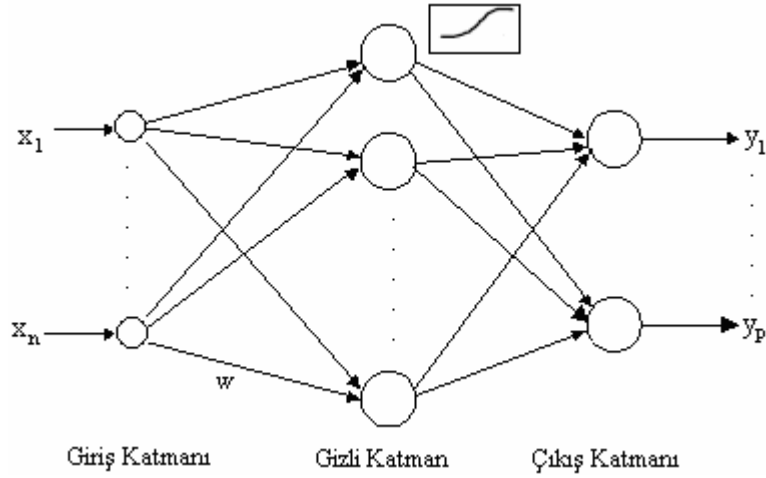
eşitsizliği doğru ise K_1 , eşitsizliğin tersi doğru ise K_2 sınıfına aittir. Burada P_1 ve P_2 , K_1 ve K_2 sınıflarının görülme olasılığıdır. L_1 , x vektörünün K_1 sınıfına ait iken K_2 olarak yanlış sınıflama oranı; L_2 ise x vektörünün K_2 sınıfına ait iken K_1 olarak yanlış sınıflama oranıdır ve maliyet fonksiyonu olarak adlandırılır. Buradan görüleceği gibi, $F_1(x)$, $F_2(x)$, L_1 ve L_2 'nin bilinmesi durumunda x vektörünün en yüksek olasılıkla hangi sınıfa ait olduğu tespit edilebilir (Goh, 2002). OSA'larda sınıflara ait yoğunluk fonksiyonları Parzen pencereleri (Parzen, 1962) kullanılarak aşağıdaki şekilde bulunur:

$$F(x) = \frac{1}{(2\pi)^{m/2} \sigma^m n} \sum_{j=1}^n \exp \left[-\frac{(x - x_j)^T (x - x_j)}{2\sigma^2} \right] \quad (2.5)$$

Burada n eğitme (referans) vektörlerinin sayısı, m giriş uzayının boyutu, j referans vektörün numarası ve σ ise ayarlanabilir bir yumuşatma terimidir.

2.3 Çok Katmanlı Algılayıcı Sinir Ağları

Çok sayıda katman içeren yapay sinir ağlarıdır, yani birimler katmanlar şeklinde düzenlenir. Bir katmandaki hücrelerin çıkışları bir sonraki katmana ağırlıklar üzerinden giriş olarak verilir. Giriş katmanı, dış ortamlardan aldığı bilgileri değişikliğe uğratmadan gizli katmandaki birimlere iletir ve bilgi gizli katmanda ve çıkış katmanında işlenerek ağın çıkışı belirlenir. Geriye yayılım öğrenme algoritmaları, bu tip YSA'ların eğitiminde etkin olarak kullanılmaktadır. Şekil 2.2'de giriş katmanı, gizli katman ve çıkış katmanı olmak üzere 3 katmanlı yapay sinir ağı yapısı gösterilmektedir (Park vd, 2002).



Şekil 2.2 Çok katmanlı algılayıcı yapay sinir ağı yapısı.

Ağın eğitiminde; ağırlıkların belirli veya rasgele değerlerle başlatıldıktan sonra en uygun değerleri alması önemli bir adımdır. Eğitime işlemde bir örnekler kümesi ve hedefler bulunması gerekmektedir. Eğitim süresince, ağın performans fonksiyonunu ki bu ileri beslemeli ağlar için en küçük karesel hatadır, minimize edecek şekilde ağırlıklar iteratif olarak ayarlanır. En küçük karesel hata, girişler ve hedefler arasındaki ortalama karesel hata olarak hesaplanır (Lera vd, 1998), (Haykin, 1994).

2.4 Radyal Temelli Fonksiyon Sinir Ağları

Radyal temelli fonksiyon ağları, çok katmanlı algılayıcı sinir ağı yapılarına benzer şekilde giriş katmanı, gizli katman ve çıkış katmanından oluşur. Giriş katmanı giriş vektör uzayı, çıkış katmanı da örüntü sınıfları ile ilişkilidir. Böylelikle tüm yapı, gizli katmanın yapısı ve gizli katman ile çıkış katmanı arasındaki ağırlıkların belirlenmesine indirgenir. Gizli katmandaki nöronların aktivasyon fonksiyonları bir c_i merkezi ve σ_i bant genişliği ile belirlenir. Aktivasyon fonksiyonu,

$$\varphi_i(x) = \exp\left[-\frac{\|x - c_i\|^2}{2\sigma_i^2}\right] \quad (2.5)$$

eşitliği ile tanımlanan bir Gauss eğrisidir. Çıkış katmanındaki j. nöronun çıkışı için genel eşitlik ise şu şekildedir:

$$y_j(x) = \sum_{i=1}^N w_{ij} \varphi_i(x) + b_j \quad (2.6)$$

Burada w_{ij} i. gizli nöron ve j. çıkış nöronu arasındaki ağırlık katsayısıdır (Verleysen vd, 1994), (Park vd, 2002).

3. GENETİK ALGORİTMALAR

Genetik algoritma (GA), doğadaki evrim mekanizmasını örnek alan bir arama metodudur ve bir veri grubu içerisinde özel bir veriyi bulmak için kullanılır. Genetik algoritmalar doğada geçerli olan en iyinin yaşaması kuralına dayanarak sürekli iyileşen çözümler üretir. Bunun için “iyi”nin ne olduğunu belirleyen bir uygunluk fonksiyonu ve yeni çözümler üretmek için çaprazlama ve değiştirme (mutasyon) gibi operatörleri kullanır. Genetik algoritmaların bir grup çözümle uğraşması özelliği sayesinde, çok sayıda çözümün içinden iyileri seçilip kötülerini elenebilir. Genetik algoritmaları diğer algoritmalarından ayıran en önemli özelliklerden biride seçmedir. Seçim işlemi de ilk grubun oluşturulması gibi rasgeledir ancak bu rasgele seçimde, seçilme olasılıklarını çözümlerin uygunluğu belirler (Goldberg, 1989).

3.1 Genetik Algoritmaların Çalışma Prensipleri

Genetik algoritmaların çalışmasını aşağıdaki gibi özetleyebiliriz;

Adım 1. Bir başlangıç popülasyonu (kromozomların bir nesli) oluşturulur, yani olası çözümlerin kodlandığı bir çözüm grubu oluşturulur.

Adım 2. Her bir kromozomun ne kadar iyi olduğu belirlenir (uygunluk fonksiyonunun belirlenmesi).

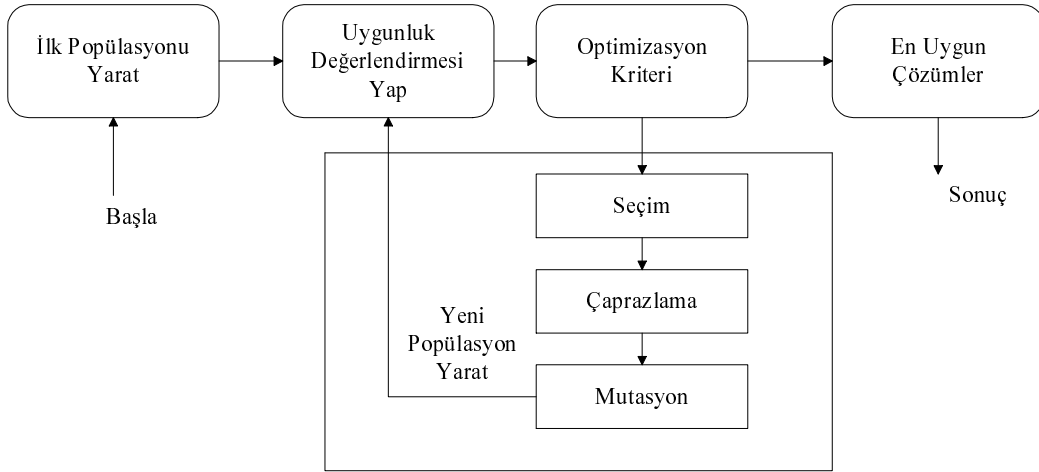
Adım 3. Mevcut kromozomlara eşleme işlemi ve bundan sonra ortaya çıkan yeni kromozomların uygunluk değerine göre yeniden kopyalama ve değiştirme operatörleri uygulanır. Bu sayede yeni bir toplum oluşturulur.

Adım 4. Nesilden arzu edilmeyen kromozomlar ortadan kaldırılır.

Adım 5. Tüm kromozomların uygunlukları tekrar hesaplanır ve 3. adımdan sonraki adımlar jenerasyon sayısı bitinceye kadar yeniden gerçekleştirilir

Bu adımlar sonunda o ana kadar bulunmuş en iyi kromozom algoritmanın sonucunu göstermektedir (Goldberg, 1989), (Karaboğa, 2004).

Genetik algoritmaların temel akış diyagramı Şekil 3.1’de gösterilmiştir, buradaki standart GA’nın çalıştırılabilmesi için tespit edilmesi gereken temel noktalar; kromozom gösterimi, seçim stratejisi, çoğalma operatörleri, ilk popülasyonun yaratılması, optimizasyonu sonlandırma kriteri ve değerlendirme fonksiyonudur (Hatanaka vd, 2003), (Goldberg, 1989).



Şekil 3.1 Genetik Algoritmanın Akış Diyagramı

3.2 Genetik Algoritmelerde Kullanılan Temel Operatörler

3.2.1 Kromozom Gösterimi

Genetik algoritma ile optimize edilmeye çalışılan problemin muhtemel çözümü bir dizi parametre ile temsil edilmektedir ve buna göre öncelikle algoritmada parametre sayısının belirlenmesi gerekir. Parametreler gen olarak adlandırılırlar ve birleşerek kromozomları oluştururlar. Kromozomların gösteriminde belirli bir alfabe kullanılır. Bu alfabe sembollerden, iki tabanlı sayılardan, tamsayılardan ve gerçel sayılardan oluşabilir. GA'nın yaygınlaşmaya başladığı ilk zamanlarda, parametre değerleri için iki tabanlı gösterim daha uygun görülüp yaygın olarak kullanılmıştır. Michalewicz, çalışmalarıyla arama uzayının doğasıyla uyumlu gösterimlerinin daha iyi sonuçlar sağladığını göstermiştir. Bu nedenle, fonksiyon optimizasyonunda, kromozomların iki tabanlı gösterimine kıyasla gerçel sayılarla ifade edilmesi daha verimlidir (Michalewicz, 1994).

3.2.2 Uygunluk Fonksiyonu

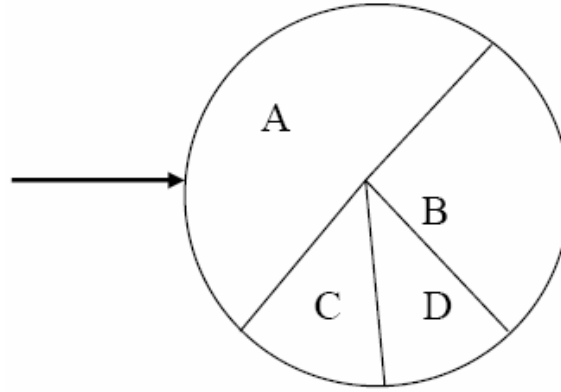
Algoritma içerisinde çözümü istenen her problem için bir uygunluk fonksiyonu belirlenmelidir. Bu fonksiyon sayesinde, belirli bir kromozomun çözüme yakınlığının göstergesi olan uygunluk değeri hesaplanmaktadır. Popülasyon içerisinde uygunluk değeri yüksek olan bireyler seçilerek, en yüksek uygunluklu birey, uygunluğu düşük olan bireyin yerini almaktadır. Böylelikle bireyler uygunluk değerlerinin yüksekliğine göre daha büyük bir olasılıkla seçilmektedir. (Duvivier vd, 1998), (Goldberg, 1989).

3.2.3 Seçim

Genetik algortimaların içerisindeki *seçim* işleminin amacı, uygunlukları düşük olan kromozomların elenip uygunlukları yüksek olan kromozomların ağırlıklarını hissettirecek yeni bir nesil oluşturmak ve böylece nesil içerisindeki kalitesi düşük bireylerin sayılarının azaltılmasıdır. Seçim işlemi, uygunlukları büyük kromozomların, uygunlukları küçük kromozomların üzerine tekrar yazılıp veya rulet çarkı seçimi (roulette wheel selection) ve turnuva seçimi (Tournament Selection) gibi seçme yöntemleri kullanılarak yapılmaktadır (Jinghui vd, 2005), (Kharbat vd, 2005).

Bu çalışmada kullanılan rulet tekerleği seçimi yöntemi aşağıda açıklanmıştır.

- 1- Popülasyondaki tüm bireylerin uygunluk değerlerinin toplamı hesaplanır.
- 2- Tüm bireylerin uygunluk değerleri bu toplam değere bölünür ve buradan bireylerin seçilme olasılıkları elde edilir.
- 3- Elde edilen bu seçilme olasılıkları değerleri, her bir bireyin bir daire (rulet tekerleği) içinde kaplayacağı alanı göstermektedir. Çark döndürülerek herhangi bir dilimde durması beklenir (Şekil 3.2) ve okun en fazla dilime sahip olan kromozomu gösterme olasılığı en yüksek olacaktır (Yuanping vd, 2006), (Jinghui vd, 2005).



Şekil 3.2 Rulet çarkı.

3.2.4 Çaprazlama

Çaprazlama işlemi, doğal sistemlerde meydana gelen veya genetik çaprazlama olayıyla ortaya çıkan melez (hibrid) yapıların üretilmesine eşdeğer bir özelliği genetik algortimaya kazandırır ve algoritmada kromozomların genlerini birbirleriyle değiştirmelerini sağlamaktadır. Önce çaprazlama yapılacak kromozom çifti rasgele seçilir ve çaprazlama operatörü bu kromozom çiftlerinden yeni iki yapı meydana getirir. Buradaki amaç mevcut nesilden daha farklı nesiller elde etmektir. (Karaboğa, 2004)

3.2.5 Mutasyon

Genetik algoritmalarda popülasyonda çeşitlilik yaratabilmek ve genel bir en iyiye ulaşabilmek için bireylerdeki kodlar belli bir olasılık ile değişime (mutasyona) uğratılmaktadır. GA'daki bu işlem önceden belirlenen bir durdurma koşulu sağlanıncaya kadar devam etmektedir. Rasgele seçilen kromozomdaki bir veya birkaç geni rasgele değişikliğe uğratan mutasyon operatörünün nesil başına uygulanma oranının düşük olması önerilir. Mutasyon işlemiyle, çaprazlama işlemiyle üretilmeyen uygunluk değeri yüksek kromozomları üretmek mümkün olmaktadır. Ancak uygunluk değeri oldukça yüksek olan kromozomları bozma ihtimali de bulunmaktadır. (Goldberg, 1989), (Yingjie vd, 2006)

Mutasyon işlemi olmayan bir genetik algoritmada, elemanların seçim ve çaprazlama işlemlerinde değişmediği varsayılırsa, optimal çözüm ancak çözüm elemanlarının başlangıç popülasyonunda bulunması halinde gerçekleşebilir (Karaboğa, 2004).

3.3 Genetik Algoritmalarındaki Kontrol Parametreleri

Yukarıda bahsedilen ve kullanıcı tarafından belirlenen operatörlerin yanında algortimanın kontrol parametrelerinin değerlerinin seçimi de algoritmanın performansını önemli bir biçimde etkilemektedir. Bir genetik algoritmanın temel kontrol parametreleri şunlardır (Karaboğa, 2004):

- a) Popülasyon Büyüklüğü: Popülasyon büyüklüğü için seçilen değer, algortimanın performansını iki şekilde etkilemektedir. İlki, popülasyon büyüklüğünün aşırı küçülmesi araştırma uzayının yetersiz örneklenmesine neden olacağından kontrollü ıraksamayı sağlamak zorlaşacak ve araştırma belirli bir alt optimum noktaya doğru sürüklenecektir. İkincisi, popülasyon büyüklüğü aşırı yüksek seçildiğinde bir jenerasyonluk gelişim oldukça uzun süreye ihtiyaç duyacaktır.

- b) **Çaprazlama Oranı:** Popölasyon bireylerine uygulanacak çaprazlama operatörünün frekansını belirleme amacıyla kullanılan parametredir. Düşük çaprazlama oranı yeni kuşağa çok az sayıda yeni yapının girmesine neden olacaktır. Çaprazlama oranı aşırı yüksek ise daha iyi yapılar üretmeden kuvvetli olan yapılar çok hızlı olarak bozulacağından algortimanın performansı düşecektir.
- c) **Mutasyon oranı:** Mutasyon işlemi popölasyona yeni bireylerin girmesini sağlamaktadır. Yüksek mutasyon oranı poölasyona aşırı bir rastgelelik kazandıracak ve tahribata neden olacaktır. Çok düşük mutasyon oranının kullanılması ise araştırma uzayının tamamen araştırılmasını engelleyecektir.

4. ALAN PROGRAMLANABİLİR KAPI DİZİLERİ (FPGA–Field Programmable Gate Arrays)

Alan programlanabilir kapı dizileri, programlanabilir lojik bloklar ve bu bloklar arasında programlanabilir ara bağlantılar içeren genel amaçlı sayısal tümleşik devrelerdir. Tasarımcıların farklı ihtiyaçları için gerekli olan lojik fonksiyonları gerçekleştirebilmesi amacıyla alanda programlanabilir olarak üretilmiş ve adlandırılmıştır. Tümdevre üreticisi tarafından sağlanan bir yazılım ile lojik bloklar ve aralarındaki bağlantılar kullanıcıların tasarladığı lojik devreye göre programlanır. Elektronik sistemlerin tasarımında, gerek uygulamalarda kullanıcıya sağladığı büyük kapasite ve esneklik, gerekse yer ve enerji sorunlarının yoğun olarak yaşandığı alanlarda kullanılabilme ve düşük maliyet özelliklerinden dolayı FPGA tümleşik devreleri tercih edilmektedir.

4.1 Programlanabilir Devre Elemanları

4.1.1 Programlanabilir Lojik

Programlanabilir lojik temel olarak programlanabilir bağlantılar ile birbirine bağlanabilen ve programlanabilen lojik blokları içermektedir. Programlanabilir cihazların 3 ana kategorisi şunlardır (Brown vd, 1996):

4.1.1.1 SPLD

Programlanabilir Lojik Eleman (Programmable Logic Device-PLD) olarak adlandırılan çipler bu alanda büyük bir dönüm noktasını oluşturmuştur. Bu çipler mantıksal devreler oluşturmak için kullanılır. PLD içinde bir AND kapıları dizisi ile bir OR kapıları dizisi içerir. Programlanabilir Dizi Lojik (Programmable Array Logic-PAL) olarak isimlendirilen cihazlar ise; sabit bir OR kapıları dizisi düzlemi ile programlanabilir AND kapıları dizisi içerir. PAL ufak mantıksal devreler için tasarlanmıştır. Bu cihazlar basit programlanabilir lojik elemanlar (Simple Programmable Logic Device-SPLD) olarak adlandırılır. SPLD'ler programlanabilir lojik ailesinin en ufak ve en ucuz üyesidir. SPLD aygıtların programlanabilmesi anti-fuse veya EPROM, EEPROM, Flash gibi kalıcı bellek kullanımı ile gerçekleştirilmektedir. [1], (Brown vd, 1996)

4.1.1.2 CPLD

Kompleks programlanabilir lojik elemanlar (Complex Programmable Logic Device - CPLD), SPLD'lere nazaran daha fazla kapasiteye sahip cihazlardır. Bir CPLD tipik olarak 2 ila 64 SPLD kapasitesine sahiptir. Bir CPLD, yüzlerce hücre içerir. Sekizden onaltıya kadar hücreler gruplanarak daha büyük bir fonksiyon bloğu oluşturur. Genel olarak bir fonksiyon bloğu içerisinde yer alan hücreler birbirlerine tamamen bağlıdır. Hangi blokların nasıl bağlandığı üretici firmaya göre değişiklik gösterir.[1],[2]

4.1.1.3 FPGA

FPGA'lar SPLD ve CPLD'den daha yoğun bir yapıya sahiptirler. Bir FPGA, lojik blok dizisi ve lojik blok dizisini çevreleyen giriş çıkış birimlerinden oluşur. Birbirleri ile programlanabilen ara bağlantılar ile bağlıdır. Tipik bir FPGA altmış dörtten onbinlerceye kadar lojik blok ve flip-flop içerir. Çoğu FPGA, lojik bloklar arasında tam bağlantı içermez. Bunun için yazılmış bir yazılım ile PCB'de görülen otomatik yönlendirme gibi lojik bloklar arasındaki yönlendirmeyi yapar.

FPGA çipi genel bir ifadeyle “Bir lojik blok dizisi, bu dizinin çevresinde bir halka oluşturan giriş çıkış birimleri ve bütün bu birimleri bağlayan programlanabilir ara bağlantılardan oluşan aygıt” olarak tanımlanabilir. Bu grupta değişik alt mimariler bulunmaktadır. Bu çiplerin içerdiği lojik blokların yapabildiği lojik fonksiyonlar ve yönlendirmenin performansı, yoğunluğu ve performansı belirlemektedir.

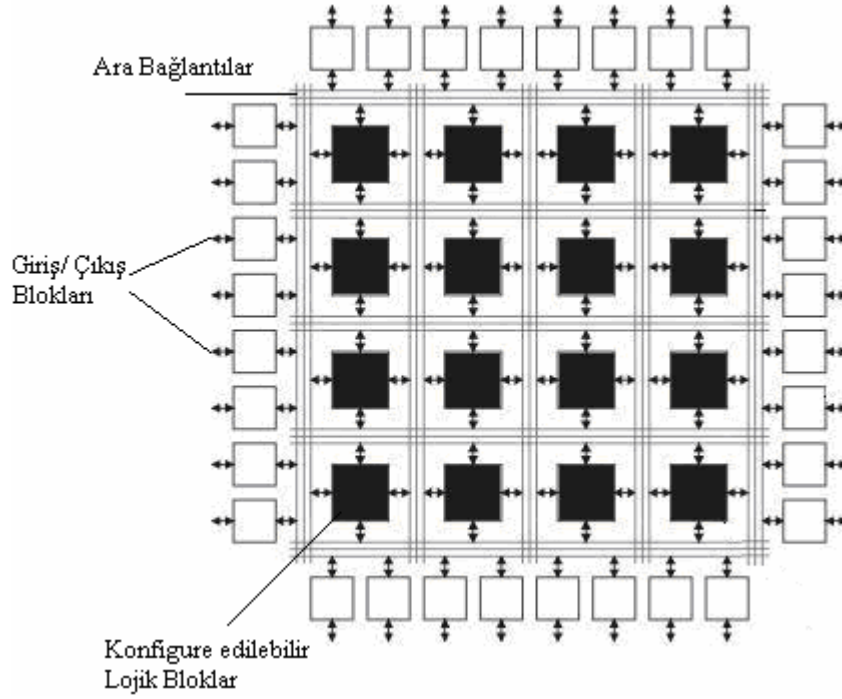
Bir FPGA, ayarlanabilir üç bileşen içerir:

- Konfigüre edilebilir lojik bloklar (Configurable Logic Block-CLB),
- Giriş/çıkış blokları (Input/Output Block-IOB),
- Ara bağlantılar.

CLB blokları kullanıcının oluşturmak istediği lojik devre için fonksiyonel elemanlar sağlar. IOB blokları tümleşik devrenin iç sinyal hatları ile pinleri arasında bir arabirim sağlar. Programlanabilir ara bağlantı kaynakları ise, CLB ve IOB giriş çıkışları arasındaki yönlendirmeyi tayin eder.

FPGA çipleri hafızaya ayar verisinin yüklenmesi vasıtasıyla özelleştirilebilir. FPGA ayar verisi dışarıdan, ister seri olarak ister paralel byte olarak bir PROM'dan yüklenebilir (master

mode), ya da FPGA çipe dışarıdan yazılabilir (slave mode). FPGA çipin programlanma süresi çipin boyutlarına ve dizayna bağlı olarak değişir. FPGA çip sınırsız olarak tekrar programlanabilir ve 50MHz saat hızlarına çıkabilir. Şekil 4.1’de lojik bloklardan oluşan iki boyutlu bir FPGA yapısı gösterilmiştir. (Brown vd, 1996), [1], [2]



Şekil 4.1 FPGA genel mimarisi.

4.2 FPGA Programlama Teknolojileri

FPGA’ler, yapısında içerdikleri programlanabilir anahtarlar ve lojik bloklardan dolayı programlanabilme özelliklerine sahiptirler. FPGA’lerin programlanması için farklı teknolojiler kullanılmaktadır, bunlar; statik RAM, Anti-fuse, EPROM tranzistörleri, EEPROM tranzistörleri’dir. Uygulama türüne göre bu teknolojilerin bazı üstün özellikleri olabilir. Bundan dolayı uygulamaya göre en uygun olan teknoloji seçilir (Brown vd, 1996), [1].

a) Statik RAM Teknolojisi

Statik RAM programlanabilir FPGA teknolojisinde bağlantılar, SRAM hücreleri tarafından kontrol edilen geçiş tranzistörleri, iletim kapıları ve çoklayıcılar vasıtasıyla sağlanır. Bu teknolojinin en büyük avantajı hızlı bir şekilde tekrar programlanabilmesidir. Dezavantajı ise kullandığı RAM teknolojisi sebebiyle çipin ihtiyaç duyduğu fiziksel alandır. FPGA çiplerde

kullanılan SRAM hücreleri hafızalarda kullanılan hücrelerin değiştirilmiş halidir. Hafıza hücrelerinde en kısa yazma okuma zamanı önem kazanırken FPGA çipte verilerin stabil olması önemlidir. SRAM kalıcı bir hafıza tipi olduğu için, cihaz kapatılsa bile bilgiler kaybolmaz. Bu nedenle FPGA cihaz açıldığında çalışmaya başlar. SRAM teknolojisinde FPGA çip çalışırken bile ayarları değiştirilebilir. Fonksiyonları devamlı değişen bir sistem için çok uygundur.[1], (Brown vd, 1996)

b) Anti-Fuse Teknolojisi

Bir anti-fuse yüksek empedans (açık devre) durumundadır. “fused” konumuna getirilerek düşük empedans konumunda programlanır. (Brown vd, 1996)

c) EPROM / EEPROM Teknolojisi

Bu metod EPROM veya EEPROM hafızalarda kullanılan metod ile aynıdır. Bu teknolojinin bir avantajı ise, dış kaynaklara ihtiyaç duymadan programlanabilmesidir. Fakat devre içinden programlanmasına izin vermez. [1], (Brown vd, 1996)

4.3 FPGA ile gerçekleştirilecek sistemlerin tasarım aşamaları

Dizayn edilecek sistemlerin tasarım süreci, gerçekleştirilecek lojik fonksiyonların, sözle veya şematik olarak tanımlanması ile başlar. Sözle tanımlamada, genellikle VHDL (Very High Speed Integrated Circuits (VHSIC) Hardware Description Language) veya Verilog gibi yüksek seviyeli donanım tanımlama dilleri kullanılır. Bu çalışmada VHDL dili vasıtası ile GRSA'nın donanım katmanlarını tanımlanmaktadır. Simülasyonlar için Modelsim, sentezleme işlemi için ise Xilinx ISE programı kullanılmıştır.

Donanım tanımlaması yapıldıktan sonra, derleme işlemini takiben tüm tanımlamalar, *standart bağlantı listesi* biçimine çevrilir. Yapılan tanımlamaların, istenilen fonksiyonları yerine getirip getirmediği davranışsal benzetim yapılarak test edilir ve alınan sonuca göre, tanımlamalarda gerekli değişiklikler yapılabilir.

Yapılan tanımlamanın doğruluğu test edildikten sonra lojik sentezleme aşamasına geçilir. Lojik sentezleme sürecinde, tanımlanmış olan fonksiyonların en uygun şekilde gerçekleştirilebilmesi için gerekli lojik iyileştirmeler yapılır ve bu işlemin ardından elde edilen lojik fonksiyonların FPGA içindeki lojik hücrelere eşleştirilmesi işlemi yapılır (Skahill, 1996), (Hwang vd, 1994). Teknoloji haritalaması (Technology Mapping) denen bu işlem esnasında, haritalama işlemini gerçekleştiren program, kullanılacak olan lojik hücrelerin

sayısını azaltmayı amaçlamaktadır (Safarpour vd, 2006).

Sentezleme işlemi sonrasında yerleştirme ve yönlendirme işlemleri gerçekleştirilir. Bu aşamada devre fonksiyonları ile eşleştirilmiş lojik bloklar FPGA içine yerleştirilir ve bu bloklar arasındaki bağlantılar oluşturulur. Yerleştirme ve yönlendirme işleminin tamamlanması ile beraber, tasarlanan devrenin gerçek kapı ve bağlantı gecikmeleri belirlenmiş olur (Skahill, 1996).

Büyük veya karmaşık uygulamalarda, yazılım başarılı bir şekilde yerleştirme ve yönlendirme yapamayabilir. Bazı yazılım paketleri tam yönlendirme yapabilmek için farklı seçenekleri deneyen yazılımlar içerir. Genel olarak, aygıt üzerindeki kaynakların %85'inden azının kullanılması tavsiye edilir. Bu teknik yazılıma, rahat yönlendirme yapabilmesi için fazladan kaynaklar sunar. Bazı üreticiler, fiziksel yerleştirmeye müdahale etmek için araçlar sağlar. Yerleştirme büyük FPGA'ler için oldukça önemlidir. Çünkü bazı yazılımlar dizayn yapısını kavramayabilir. İyi bir yerleştirme yazılımı dizayn yapısını yerleştirip yönlendirmeyi yapabilmesini sağlar [1]. Yerleştirme ve yönlendirme adımları sonrasında simülasyonlarda istenilen sonuçların elde edilmesine bağlı olarak FPGA'in programlaması aşamasına geçilir.

5. YAPAY SİNİR AĞLARI İLE PATTERN TANIMA

Pattern terimi, ilgilenilen nesnelere ait gözlenebilir veya ölçülebilir bilgiler olarak tanımlanabilir. Pattern tanıma çalışmalarının amacı ise bu nesneleri bir kategoriye koymak veya sınıflamaktır. Patternler, gerçekleştirilecek uygulamaya göre farklı boyutlarda yani görüntü, ses veya sınıflandırılması istenen başka bir işaret olabilir. Pattern tanıma uygulamaları mühendislik, tıp ve askeri alanlarda sık olarak kullanılmaktadır. Bu uygulamalara örnek olarak; nesne, el yazısı karakter, yüz, parmak izi ve ses tanıma, EEG (Elektroensefalografi) ve EMG (elektromiyografi) verisi sınıflama ve hedef tanıma çalışmaları verilebilir.

Bir pattern tanıma sistemi genel olarak 3 kısımdan oluşmaktadır; ilk blok patternlere önışlemenin yapıldığı blok, ikinci blok patternlerden özelliklerin çıkarıldığı blok ve son kısım sınıflandırma bloğudur.

Bu çalışmada ilk olarak; 3 boyutlu nesne tanıma ve el yazısı rakam tanıma uygulamaları giriş görüntülerine boyut azaltma dışında herhangi bir önışleme ve özellik çıkarma işlemi uygulamadan, YSA'lar ile gerçekleştirilmiştir. Literatürde yüz tanıma, el yazısı karakter tanıma gibi farklı pattern tanıma uygulamalarında kullanılan ve giriş görüntülerinden spesifik bir özelliğin çıkarılmadığı, yani doğrudan giriş görüntülerinin ağına girişine verilmesiyle tanıma işleminin gerçekleştirildiği *konvolüsyonel yapay sinir* ağlarının uygulamaları mevcuttur. Bu sistemlerde sadece giriş görüntüleri ile belirli maskelerin konvolüsyonu işlemi ile özellik haritaları çıkarılmakta ve ağına belirli katmanlarında boyut azaltma işlemi olmakta ve ağına son katmanında Multilayer perceptron ağ yapıları ile sınıflandırma işlemi gerçekleştirilmektedir (Nebauer, 1998), (Fasel, 2002), (Simard vd, 2003). Bu tezde önerilen modelin en büyük avantajı, gerçek zamanlı pattern tanıma uygulamalarında karşılaşılan en büyük güçlüklerden biri olan önışleme ve özellik çıkarma işlemlerine gerek kalmadan ve özellikle nesne tanıma uygulamasında yüksek tanıma oranı elde edilerek gerçekleştirilebilir olmasıdır. Önerilen yöntem görüntülerden özellik çıkarmanın karmaşık ve güç olduğu tanıma uygulamaları için kullanılabilir.

Tezin nesne tanımaya yönelik bölümünde ikinci olarak; 3 boyutlu benzer nesnelerin tanınması uygulaması giriş görüntülerinden özellik çıkarılması suretiyle GRSA ve OSA kullanılarak gerçekleştirilmiştir.

5.1 Giriş Görüntülerinden Özellik Çıkarmaksızın 3-Boyutlu Nesne ve El Yazısı Rakam Tanıma

5.1.1 3-Boyutlu Nesne Tanıma

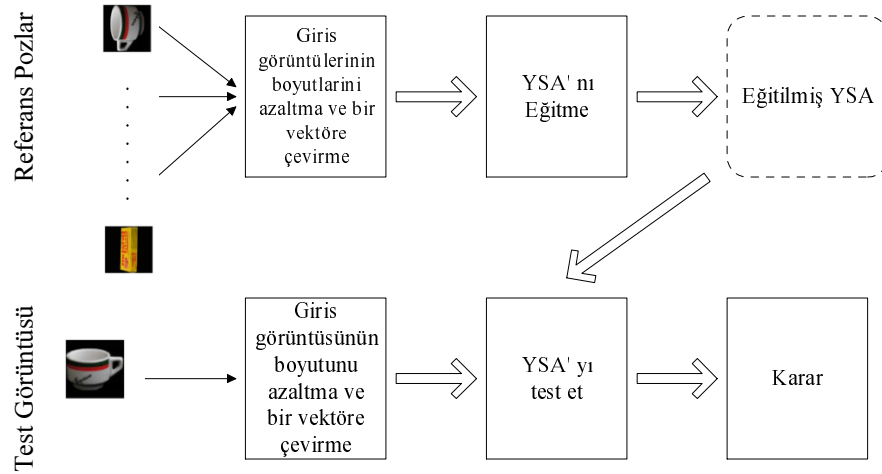
Günümüzde teknolojinin hızla ilerlemesiyle birlikte ağırlıklı olarak endüstri ve güvenlik uygulamalarında kullanılan nesne tanıma çalışmalarında da gelişmeler kaydedilmektedir. Tanıma sisteminde mevcut nesnelerden elde edilecek bir veri tabanı üzerinden otomatik olarak yapılacak tanımlar, nesnelerin tanınması sırasında yetkililere büyük kolaylık sağlamaktadır. Bu çalışmada bir nesne veri kümesi kullanılarak nesnelerin otomatik olarak tanınmasını sağlayacak bir yaklaşım sunulmaktadır.

Literatürde bizim kullandığımız nesne veri kümesi kullanılarak yapılan farklı çalışmalar mevcuttur (Daewon vd, 2007), (Lian vd, 2004), (Okamoto vd, 1998). (Daewon vd, 2007) çalışmasında Destek vektör domeni tanımlaması (support vector domain description) kullanılarak domen tanımlama bilgisi oluşturulmakta ve daha sonra Bayesian karar kuralına göre sınıflama yapılmaktadır. Uygulama 20 nesne için denenmiş ve bütün veri kümesinin %70'i eğitime için kullanılmış geriye kalan %30 luk kısım test edildiğinde, % 3,24 lük bir tanıma hatası oluşmuştur. (Lian vd, 2004) çalışmasında temel bileşen analizi (principal component analysis-PCA) uygulayarak giriş görüntülerinden özellik çıkarılması suretiyle nesnelerin sınıflandırılması gerçekleştirilmiştir. Uygulama 20 nesne için denenmiş ve bütün veri kümesinin %16,6'sı eğitime için kullanılmış geriye kalan %83,4 luk kısım test edildiğinde, % 96,45 lik bir tanıma oranı elde edilmiştir. (Okamoto vd, 1998) çalışmasında ise 5 tane 3 boyutlu nesnenin dönerken yaptığı optik akışın hesabından 13 tane özellik çıkarmak suretiyle nesnelerin sınıflandırılması işlemi gerçekleştirilmiştir. Yapılan bu çalışmalarda görüldüğü üzere, sınıflandırma işleminden önce giriş görüntülerinden özellik çıkarma işlemleri gerçekleştirilmektedir.

Bu tez çalışmasında tanıma sisteminde 3 boyutlu her bir nesnenin 2 boyutlu poz görüntüleri için boyut azaltma dışında herhangi bir ön işleme veya spesifik bir özellik çıkarma işlemi yapılmamaktadır. Nesnelerin sınıflandırılması için Çok Katmanlı Algılayıcı, Radyal Temelli Fonksiyon, Genelleştirilmiş Regresyon ve Olasılıksal Sinir Ağları olmak üzere dört farklı YSA çeşidi kullanılmıştır. GRSA ve OSA 10 farklı nesne için simule edilmiştir ve yüksek tanıma oranı elde edilmiş ve MLP ve RBF yapılarına göre daha hızlı bir simülasyon zamanına ve tanıma oranına sahip oldukları gözlemlenmiştir. (Polat ve Yıldırım, 2007), (Polat ve Yıldırım, 2006a)

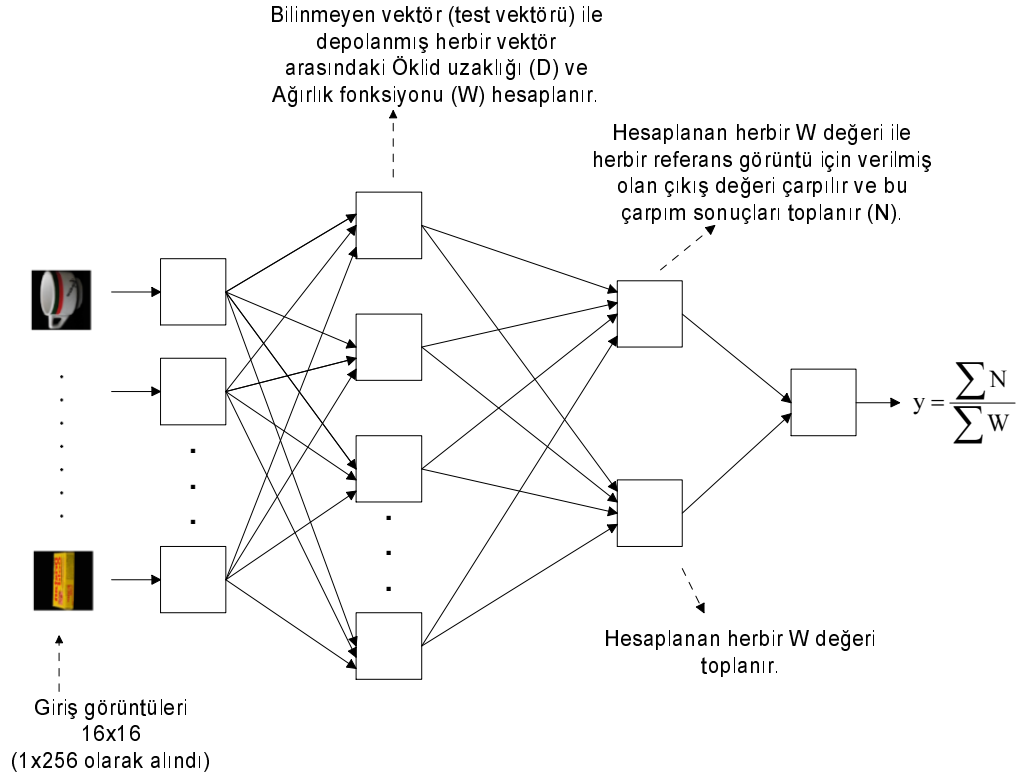
5.1.1.1 Metot

Çalışmada tanınması istenen her bir nesneden döndürülmek suretiyle elde edilmiş pozlardan oluşan ve her bir nesne için toplam 72 adet görüntünün bulunduğu bir küme kullanılmıştır. İlk olarak 128x128 lik görüntüler 16x16 lık olarak yeniden boyutlandırıldı. 72 pozun her biri için aynı işlem tekrarlandı ve bu pozların bir kısmı referans olarak alınarak yapay sinir ağı simüle edildi. Geri kalan kısım ise test edildiğinde GRSA ve OSA ile yüksek tanıma oranı elde edildi. Tasarlanan sistemin blok diyagramı şekil 5.1'deki gibidir.



Şekil 5.1 YSA'lar ile nesne tanıma sisteminin blok diyagramı.

Nesne tanıma işleminde yüksek test başarısı elde edilen GRSA ağıнын katmanlarındaki aşamalar ise şekil 5.2'de gösterilmiştir.



Şekil 5.2 Nesne tanıma işleminde GRSA'nın katmanlarındaki işlemler.

5.1.1.2 Simülasyon Sonuçları

Simülasyon için Columbia görüntü veri kümesi kullanılmıştır. Columbia Obje Görüntü Kütüphanesinde (COIL-100) 100 objenin renkli görüntüleri mevcuttur (Sameer vd, 1996). Veri kümesi oluşturulurken nesneler siyah bir arka planla dönen bir masa üzerine yerleştirilmiş ve sabit bir kamera ile her bir nesneden döndürülmek suretiyle pozlar elde edilmiştir. Çalışmamızda bu veri kümesinden şekil 5.3'de gösterilen 10 nesneyi seçtik. 10 nesne için toplam 720 resim bulunmaktadır. Şekil 5.4'de simülasyonda kullanılan 6. nesnenin 0° den 25° ye kadar olan poz görüntüleri gösterilmektedir.



Şekil 5.3 Tanıma sisteminin simülasyonunda kullanılan 10 nesne.



Şekil 5.4 Veri kümesindeki 6. nesnenin 0° den 25° ye kadar olan poz görüntüleri.

Veri kümesindeki her bir renkli görüntü önce gri seviyeye çevrildi, sonra 16x16 şeklinde boyutlandırılan görüntü matrisleri 1x256'lık bir vektöre dönüştürülerek bir veri kümesi oluşturuldu. Bu veri kümesinden alınan referans vektörler YSA'nın girişine uygulandı. Ağa hedef değerler olarak her bir nesne için bir kod verildi ve ağ bu şekilde simule edildi. Veri kümesinin kalan kısımları ile ağ test edildi.

Çizelge 5.1'de kullanılan YSA yapıları için elde edilmiş maksimum doğruluk değerleri ve ağların eğitime süreleri gösterilmiştir. Bu tablodan görüleceği üzere GRSA ve OSA için test başarısı oldukça yüksektir. GRSA 'nın eğitime süresi P4 3GHz, 256 MB RAM özelliklerine sahip PC de, MATLAB çalışma ortamında 0,07 sn olmaktadır. 16x16 olarak boyutlandırılan nesne tanıma uygulamasında 1 gizli katmanlı ve gizli katmandaki nöron sayısının 10 olduğu MLP ağının minimum hataya ulaşmak için geçen eğitime süresi yaklaşık olarak 8 dakika sürmektedir.

Çizelge 5.1 Nesne tanıma uygulaması için farklı YSA'nın eğitime ve test doğruluk oranları.

	Eğitime Başarısı (%) (Veri kümesinden her bir nesne için 12 örnek referans olarak alındığında)	Test Başarısı (%) (Geriye kalan 60 poz ile ağ test edildiğinde)	Eğitime Zamanı (saniye)
MLP	%100	%75	480
RBF	%90	%60	1,1
OSA	%100	%96,83	0,06
GRSA	%100	%95,83	0,07

Çizelge 5.2’de orijinal veri kümesinden sırasıyla 60° ve 30° dönme aralıkları ile çekilmiş pozlar yani toplamda 6 ve 12 poz referans olarak alındığında, kalan pozlarla da GRSA test edildiğinde elde edilen tanıma oranları gösterilmiştir. OSA kullanılarak 12 referans görüntünün kullanıldığı tanıma işleminde her bir nesne için elde edilmiş test sonuçları ise Çizelge 5.3’de verilmiştir.

Çizelge 5.2 OSA ve GRSA için referans görüntülerin sayısı ile test setinin tanıma oranının değişimi.

Referans Pozların Sayısı	Test seti için tanıma oranı	
	OSA	GRSA
Her bir nesnenin 60° lik açı farkına sahip görüntüleri için ağ simule edildiğinde (toplam 6 örnek ile simule, geriye kalan 66 poz ile ağ test edildiğinde)	%85,76	%85,15
Her bir nesnenin 30° lik açı farkına sahip görüntüleri için ağ simule edildiğinde (toplam 12 örnek ile simule, geriye kalan 60 poz ile ağ test edildiğinde)	%96,83	% 95,83

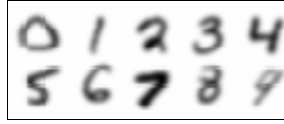
Çizelge 5.3 OSA ile herbir nesnenin tanıma oranı.

	Obj1	Obj2	Obj3	Obj4	Obj5	Ort.
Test Başarım Oranları	86,6%	100%	93,3%	100%	100%	%96,83
	Obj6	Obj7	Obj8	Obj9	Obj10	
	98,3%	100%	90%	100%	100%	

Bu tablolardan görüleceği üzere, eğitime aşamasında farklı sayıda referans veri alınarak sistemlerin performansı denenmiştir. Veri kümesinden seçilen 10 nesne için bulunan bütün poz görüntülerinin %16,6’sı eğitime için alındığında geriye kalan % 83,4 lük kısmın test performansı OSA için %96,83, GRSA için % 95,83 olmaktadır. Şayet bütün poz görüntülerinin %8,33’sı eğitime için alındığında geriye kalan % 91,66 lık kısmın test performansı OSA için %85,76, GRSA için % 85,15 olmaktadır.

5.1.2 El Yazısı Rakam Tanıma

Şekil 5.1’deki tasarlanan yapının performansı el yazısı rakamların otomatik olarak tanınmasını sağlamak amacı ile de denenmiştir. Simülasyon için NIST [4] el yazısı karakter kümesi kullanılarak her bir rakamdan 20’şer adet alınmıştır. Veri kümesindeki rakamlardan bazıları şekil 5.5’de gösterilmiştir.



Şekil 5.5 NIST veri setindeki el yazısı rakamlardan bazıları.

Tanıma sisteminde veri kümesindeki rakamlar için nesne tanıma uygulamasına benzer şekilde herhangi bir önışleme veya bir özellik çıkarma işlemi yapılmamaktadır. El yazısı karakterlerin sınıflandırılması için GRSA ve OSA yapıları denenmiştir. Sistem 0–9 arası her bir rakamdan toplam 10’ar adet (toplam 100 karakter) kullanılarak eğitilmiş ve 10’ar adet (toplam 100 karakter) ile de test edilmiştir. (Polat ve Yıldırım, 2007), (Polat ve Yıldırım, 2006a)

El yazısı tanıma işleminde GRSA ağının katmanlarındaki aşamalar şekil 5.2’de gösterilen yapı ile aynıdır. 28x28 gri seviye giriş görüntüsünde bir önışleme veya özellik çıkarma işlemi yapılmamaktadır. 28x28 lik görüntüler 1x784 lük bir giriş vektörüne dönüştürüldükten sonra bu veri kümesinden elde edilen referans vektörler GRSA ve OSA ağının girişine uygulandı. Ağa hedef değerler olarak her bir rakam için bir kod verildi ve ağ bu şekilde simule edildi. Veri kümesinin kalan kısımları ile ağ test edildi. GRSA ile elde edilen test sonuçları aşağıdaki Çizelge 5.4’de, OSA için elde edilen test sonuçları ise çizelge 5.5’de gösterilmektedir. Ağların eğitme başarısı ise %100 dür.

Çizelge 5.4 GRSA için herbir rakamın tanıma oranı.

Rakamlar	0	1	2	3	4	5	6	7	8	9	Ort.
Test sonuçları	%100	%70	%70	%80	%20	%60	%50	%60	%60	%80	%65

Çizelge 5.5 OSA için herbir rakamın tanıma oranı.

Rakamlar	0	1	2	3	4	5	6	7	8	9	Ort.
Test sonuçları	100%	80%	80%	80%	20%	50%	50%	70%	60%	80%	%67

Bu sonuçlara göre el yazısı rakam tanıma uygulamasında, GRSA ve OSA yapıları ile yapılan sınıflandırmada 4, 5, 6 gibi rakamlarda tanıma başarısı düşük kalmaktadır, ortalama olarak ise dikkate değer bir test başarısı elde edilmiştir. El yazısı rakam tanıma uygulaması RBF ağları ile gerçekleştirildiğinde yaklaşık olarak %10 test başarısı elde edildi, MLP uygulamasında ise

256 MB RAM sahip bilgisayar için bellek problemi oluşmakta ve ağ eğitilememektedir.

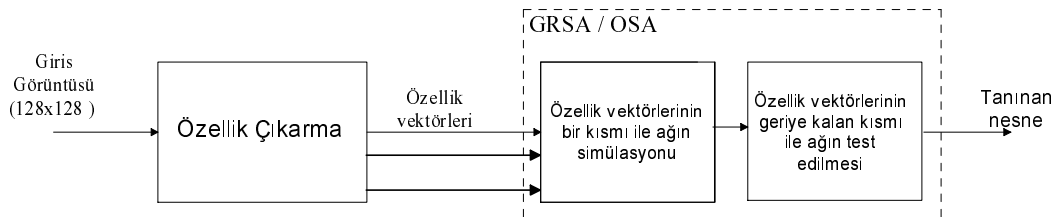
Buna göre önerilen metot ile herhangi bir önışleme ve özellik çıkarma işlemi yapmadan özellikle nesne tanıma uygulamasında az sayıda referans küme ile yüksek başarı oranı elde edildiği, el yazısı rakam tanıma uygulamasında ise dikkate değer başarı oranı gösterdiği gözlemlenmiştir. Önerilen bu topoloji özellikle bazı uygulamalardaki karmaşık ve zor bir işlem gerektiren önışleme ve özellik çıkarma işlemlerinden kaçınılması istendiği durumlarda kullanılabilir. Ayrıca imza ve el yazısı harf tanıma uygulamaları gibi farklı pattern tanıma uygulamalarında da denenebilir. Bu uygulamalarda GRSA ve OSA yapılarının hız ve doğruluk açısından diğer MLP ve RBF ağ yapılarından daha yüksek performans gösterdiği gözlenmiştir.

5.2 3-Boyutlu Benzer Nesne Tanıma

Çalışmanın bu aşamasında benzer şekile sahip nesnelerin otomatik olarak tanınmasını sağlayacak bir yaklaşım sunulmaktadır. Tanıma sisteminde 3 boyutlu her bir nesnenin 2 boyutlu poz görüntülerinden renk özellikleri çıkarılmış ve bu özellik vektörleri kullanılarak GRSA ve OSA yapılarıyla nesnelerin sınıflandırılması işlemi gerçekleştirilmiştir. Sistem, şekilleri benzer 8 farklı nesne ile simule edilmiştir ve yüksek bir tanıma oranı elde edilmiştir. Ayrıca ağın çok az sayıda örnekle eğitildikten sonra çok sayıda bilinmeyen nesneyi tanıyabilmesi de sistemin önemli bir özelliğidir.(Polat ve Yıldırım, 2006b)

5.2.1 Metot

Tanıma işleminde kullanılan sistem şekil 5.6’da gösterildiği gibi iki kısımdan oluşmaktadır, ilk kısım özelliklerin çıkarıldığı blok, diğer kısım sınıflama amaçlı kullanılan YSA bloğudur. Veri kümesindeki 128x128’lik her bir görüntüden R-G-B (red/green/blue-kırmızı/yeşil/mavi) tonlarının ortalama değerleri alınarak her bir poz için 3 farklı özellik değeri elde edildi. Bütün pozlar için aynı işlem tekrarlandı ve pozların bir kısmı referans kümesi olarak alındı ve GRSA ve OSA simule edildi ve geri kalan kısım ile de test edildi.



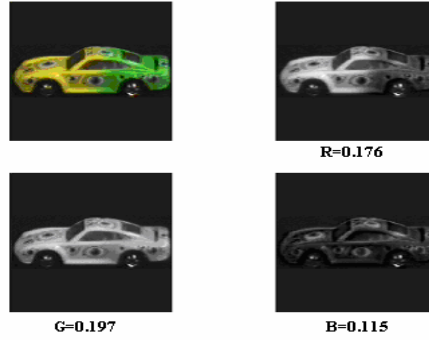
Şekil 5.6 Nesne tanıma sistemi.

5.2.2 Simülasyon Sonuçları

Simülasyonlar için yine Columbia görüntü veri kümesi kullanıldı. Çalışmamızda bu veri kümesinden şekil 5.7’de gösterilen benzer 8 nesneyi seçtik. Nesnelerin 5° lik açılarla çekilmiş pozlarından oluşan ve her bir nesne için toplam 72 adet görüntü bulunmaktadır, yani 8 nesne için toplam 576 resim bulunmaktadır (Sameer vd, 1996). RGB giriş görüntülerinin her biri için R, G ve B renk matrislerinin ortalama değerleri alındı. Bu şekilde o giriş görüntüsü için 3 farklı özellik vektörü oluşturulmuş oldu. Şekil 5.8’de 2 numaralı oyuncak arabanın 0° deki pozunun R-G-B görüntüleri ve ortalama piksel değerleri gösterilmiştir.



Şekil 5.7 Simülasyon için kullanılan benzer 8 nesne.



Şekil 5.8 Veri kümesindeki 2 numaralı oyuncak arabanın 0° deki pozunun RGB görüntüleri.

Daha sonra 8 nesnenin 72 farklı açıdan çekilmiş görüntüleri için elde edilen özellik vektörleri kullanılarak YSA’lar simule ve test edildi. Çıkarılan özelliklerden elde edilen referans vektörler sinir ağlarının girişine uygulandı. Ağa hedef değerler olarak her bir nesne için bir kod verilerek ağlar simule edildi. Veri kümesinin kalan kısımlarından elde edilen özellik vektörleri ile ağ test edildi. Çizelge 5.6’da orijinal veri kümesinden sırasıyla 60° ve 30° dönme aralıkları ile çekilmiş pozlar yani toplamda 6 ve 12 poz referans olarak alındığında ve kalan pozlarla ağ test edildiğinde elde edilen tanıma oranları gösterilmiştir.

Çizelge 5.6 Benzer nesne tanıma uygulamasında GRSA ve OSA için referans görüntülerin sayısı ile test setinin tanıma oranının değişimi.

Referans Pozların Sayısı	Test Başarısı (%)	
	GRSA	OSA
Her bir nesnenin 60° lik açılı farkına sahip görüntüleri için ağ simüle edildiğinde (toplam 6 örnek ile simüle, geriye kalan 66 poz ile ağ test edildiğinde)	% 85,04	% 88,64
Her bir nesnenin 30° lik açılı farkına sahip görüntüleri için ağ simüle edildiğinde (toplam 12 örnek ile simüle, geriye kalan 60 poz ile ağ test edildiğinde)	%93,33	%93,13

Farklı verilerin eğitime ve test için kullanılması da sonucu etkilemektedir. Örneğin her bir nesnenin 35° lik açılı farkına sahip görüntüleri için elde edilmiş özellik vektörü ile GRSA ağı simüle edildiği zaman (toplam 10 örnek referans olarak alındığında), geriye kalan örnekler test edildiğinde sistemin test başarısı % 97 olmaktadır. Bu aşamada 30° lik açılı farkına sahip duruma göre 2'şer adet daha az sayıda referans veri kullanılmasına rağmen, ancak farklı verilerin referans olarak alınmasıyla test başarısının arttığı, yani referans olarak alınan verinin seçiminin sistemin başarısı etkilediği gözlemlenmiştir.

Bu uygulama benzer biçimli 8 nesne için yapılmış ve yüksek tanıma oranı sağlanmıştır. Ayrıca ağı çok az sayıda örnekle eğitildikten sonra çok sayıda bilinmeyen nesneyi tanıyabilmesi de önemli bir özelliktir. Burada nesne sayısı artırıldığında, giriş görüntülerinden farklı özellikler çıkarılarak tanıma oranı daha da artırılabilir. Alınan sonuçlar 3 boyutlu benzer nesnelerin tanınmasında GRSA ve OSA yapılarının başarılı bir şekilde kullanılabileceğini göstermiştir.

Bu aşamada benzer nesneler için GRSA yapısının performansı daha önceki kısımda anlatılan özellik çıkarmaksızın tanıma modeli ile de gerçekleştirilmiştir. Orjinal veri kümesinden sırasıyla 6 ve 12 poz referans olarak alındığında ve kalan pozlarla da ağ test edildiğinde % 59,1 ve % 79,6 test performansı elde edilmiştir. Sistemin eğitime başarısı ise % 100 olmaktadır. Bu sonuca göre benzer nesnelerin sınıflandırılması uygulamasında özellik çıkarma işlemi GRSA'ın sınıflandırma başarısını arttırmıştır.

6. PATERN TANIMA SİSTEMİNİN GENETİK ALGORİTMA İLE OPTİMİZASYONU

Günümüzde özellikle mühendislik uygulamalarının performanslarını arttırmak ve problemlere hızlı ve kolay çözüm verebilmesi bakımından optimizasyon tekniklerinin kullanımı önem kazanmaktadır. Bir optimizasyon tekniği ve evrimsel yaklaşımlardan olan genetik algoritmaların kullanımı da oldukça yaygınlaşmıştır.

Bu çalışmada, geliştirilen patern tanıma uygulamalarında oldukça başarılı sonuçlar veren GRSA ve OSA yapılarının optimizasyonları genetik algoritma ile gerçekleştirilmiştir. Bu aşamada özellikle tasarlanan patern tanıma sistemlerinde optimum referans kümenin seçimi işlemi yapılmıştır. Çalışmada ilk olarak örnek bir sınıflandırma problemi için GRSA ağıının optimizasyonu genetik algoritma ile gerçekleştirilmiştir. Ağdan en yüksek test performansını elde etmek için ağa uygulanacak referans paternler ve Gauss fonksiyonunun genişliği genetik algoritma ile belirlenmeye çalışılmıştır (Polat ve Yıldırım, 2006c), (Yazıcı vd, 2006). Daha sonra ise önceki bölümlerde anlatılan 3 boyutlu nesne tanıma ve El yazısı rakam tanıma uygulamalarında kullanılan GRSA (Polat ve Yıldırım, kabul edildi, basımda-a) ve OSA yapılarının optimizasyonu gerçekleştirilmiştir.

Literatürde RBF ağlarının sınıflandırma ve fonksiyon yaklaştırma problemleri için Complexity ve Evrimsel Optimizasyon gibi farklı algoritmalar kullanılarak optimizasyonunun yapıldığı çalışmalar bulunmaktadır (Leonardis vd, 1996), (de Lacerda vd, 2000). Ayrıca farklı uygulamalar için RBF ağlarının genetik algoritma ile optimizasyonunun yapıldığı çalışmalar da mevcuttur (QingNian vd, 2003), (Hatanaka, 2003). RBF ağlarının optimizasyonu için yapılan bu çalışmalarda genel olarak optimize edilecek parametreler; nöronlar arasındaki bağlantı ağırlıkları, katman sayısı veya öğrenme oranı değerleri olmaktadır. Radyal temelli olasılıksal sinir ağlarının genetik algoritma ile optimizasyonu ile ilgili bir çalışmada ise ağın giriş katmanı için merkezlerin seçimi işlemi gerçekleştirilmektedir (Wen-Bo vd, 2004). Literatürde GRSA yapısının optimizasyonu; lokal minimum bulma problemi için Quasi-Newton, Genetik ve Conjugate Gradient gibi farklı algoritmalarla gerçekleştirilmiş ve optimal sigma (Gauss Kernel'nin genişliği) değeri belirlenmeye çalışılmıştır (Masters vd, 1997). (Zhanwei vd, 2003) de ise GRSA 'nin patern katmanında kullanılan Gauss fonksiyonunun genişliğinin dinamik olarak belirlenmesi amaçlanmaktadır. (Eric vd, 2006) de ise gürültülü veri sınıflandırma problemi için GRSA ve Bulanık Adaptif Rezonans Teorisi modelinin birlikte kullanıldığı hibrid bir yapı da Gauss Kernel'inin merkezleri ve genişliğinin kestirimi yapılmaktadır.

Patern tanıma uygulamaları için ise; Kombinasyonel (Combinatorial) Optimizasyon ile nesne tanıma uygulaması (Irie vd, 2003) veya Bulanık-Nöral yapılarının genetik algoritma ile optimizasyonu yapılarak yapay koku tanıma çalışmaları mevcuttur (Kusumoputro vd, 2002).

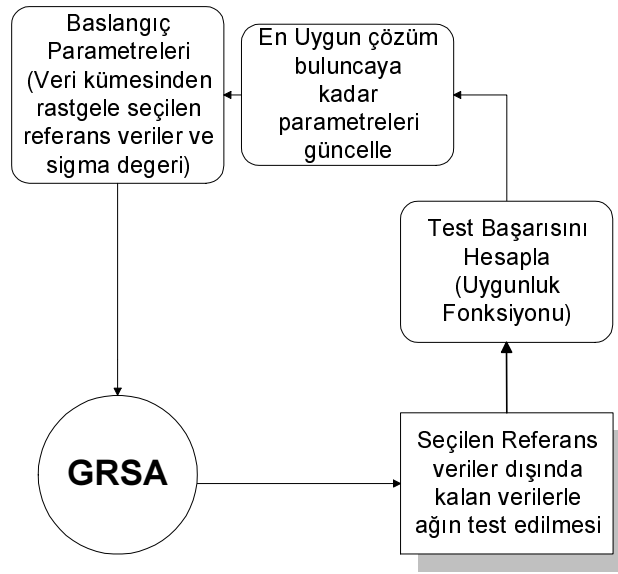
Gauss Kernel'inin kullanıldığı GRSA ve OSA gibi yapıların optimizasyonun da, literatürdeki çalışmalara bakıldığında genel olarak yapılan işlem çoğunlukla uygulanan probleme göre optimal kernel genişliğinin belirlenmesine çalışmaktır. Mevcut merkezlerden farklı bir optimal merkez bulma işlemi ise özellikle sınıflandırma problemleri için çok fazla özelliğin çıkartıldığı uygulamalarda veya bizim çalışmamızdaki gibi iki boyutlu verilerden hiçbir özellik çıkarma işlemi yapmadan gerçekleştirilecek olan tanıma uygulamalarında çok fazla miktarda optimize edilecek parametre çıkmasına neden olacak ve bu durumun da hız problemi yaratabileceği düşünülmektedir.

Bu çalışmada gerçekleştirilen sınıflandırma uygulamalarında mevcut veri kümeleri içinden optimal merkezler belirlenmeye çalışılmıştır yani önerilen metot; sınıflandırma için kullanılan GRSA ve OSA yapılarının test performanslarını maksimum yapacak, sigma (σ) genişliği değeri ve mevcut olan özellik vektörlerinin hangilerinin referans olarak alınacağını belirlemeyi amaçlamaktadır. Buna göre optimize edilecek parametre sayısı; sigma değeri ve mevcut veri kümesinde bulunan bütün özellik vektörlerinin içinden seçilecek olan ve her bir sınıf için bizim belirleyeceğimiz referans veri sayısı kadar olmaktadır. Simülasyonlarımızda veri kümelerinin küçük bir kısmını referans veri olarak aldığımız düşünüldüğünde (Örneğin tiroid verisi sınıflandırma uygulaması için tüm veri kümesinin sadece % 16,2 si referans veri olarak alındı) optimize edilecek parametrelerin sayısında buna bağlı olarak düşük olacaktır.

6.1 Metot

Yapılan uygulamada öncelikle farklı veri kümelerinin sınıflandırılması için kullanılan GRSA'nın optimizasyonu genetik algoritma ile gerçekleştirilmiştir. Ağdan en yüksek test performansını elde etmek için ağa uygulanacak örnek paternler ve gauss fonksiyonunun genişliği genetik algoritma ile belirlenmeye çalışılmıştır. Bu aşamada veri kümesinden simülasyon için minimum sayıda örnek alınarak, geriye kalan test kümesinin yüksek başarı oranı ile sınıflandırılması amaçlanmıştır. Daha sonra aynı optimizasyon uygulaması daha önceki çalışmalarda dizayn edilen patern tanıma sistemleri için denenmiştir. GRSA ve OSA'nın genetik algoritma ile optimizasyonunun ağların test performansını arttırdığını gözlemlenmiştir.

Çalışmamızda genetik algoritma için uygunluk fonksiyonu test başarıım oranıdır. Optimize edilecek parametrelerin (bunlar; σ değeri ve her bir sınıf için referans olarak belirlenecek özellik vektörlerinin veri kümesi içerisindeki yeridir) alabileceği maksimum ve minimum değerler yazılımda belirtilmektedir. Algoritma verilen sınır değerler arasından en uygun çözümü bulmaya çalışmaktadır. Algoritma istenen çözüm sağlanıncaya kadar devam ettirilir. GRSA ile gerçekleştirilen farklı patern tanıma uygulamalarının optimizasyonu için akış diyagramı şekil 6.1'deki gibidir. OSA için gerçekleştirilen optimizasyon işlemi de benzer şekildedir.



Şekil 6.1 Sınıflandırma amacıyla kullanılan GRSA'nın GA ile optimizasyonunun akış diyagramı

Yapılan uygulamalarda farklı sınıflandırma problemleri için GRSA ve OSA'nın genetik algoritma ile optimizasyonunda, genetik algoritma içerisinde oluşturulan popülasyon boyutları ve her bir jenerasyonda kaç kromozomun mutasyona uğramasına izin verildiğini gösteren çizelge 6.1'deki gibidir. Bu değerler; hız ve doğruluk açısından algoritma performansına bakılarak yaklaşık bir değerde belirlenmiştir.

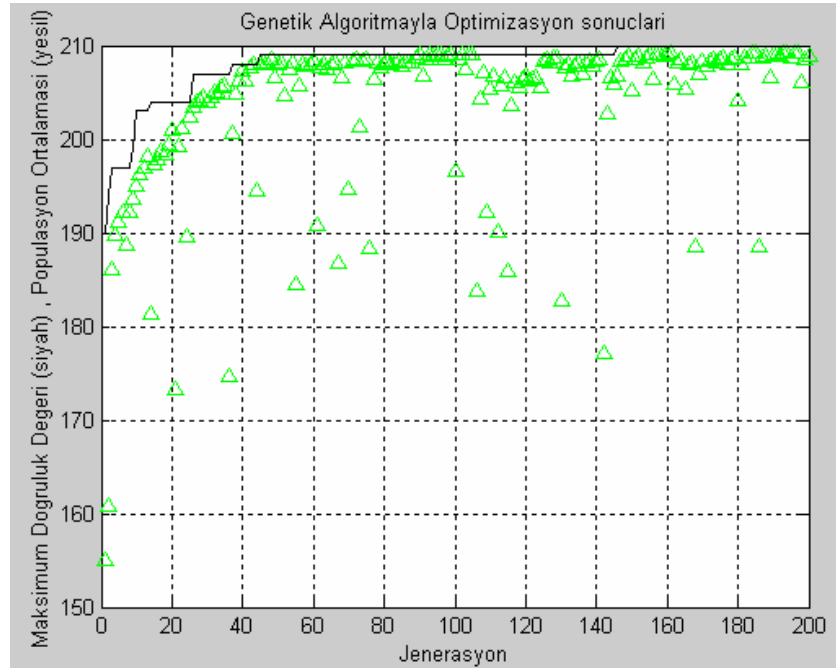
Çizelge 6.1 Genetik algorithmadaki parametrelerin sayısı

	GA içerisindeki popülasyon boyutları	Herbir jenerasyonda mutasyona uğramasına izin verilen kromozom sayısı	Jenerasyon Sayısı
Tiroid Verisi Sınıflandırma	40	2	200
Nesne Tanıma	140	4	10
El Yazısı Rakam Tanıma	240	4	10

6.2 Simülasyonlar

İlk olarak, örnek bir uygulama olarak 3 farklı Tiroit hastalığının verilen 5 özellik değerine göre sınıflandırılması problemi optimize edilmiştir. Veri kümesinde birinci sınıftan 150, ikinci sınıftan 35 ve üçüncü sınıftan 30 örnek bulunmaktadır. GRSA ağına örnek giriş paternlerini belirlemek için birinci sınıftan 150 değerden hangi 25'i, ikinci sınıftan 35 örnekten hangi 5'i ve üçüncü sınıftan 30 örnekten hangi 5'i seçilirse, veri kümesinde geriye kalan kısım için en yüksek test başarısı elde edilir sorusu genetik optimizasyon ile çözülmeye çalışılmıştır. Sonuçta veri kümesinde geriye kalan, birinci sınıftan 125 örnek, ikinci sınıftan 30 örnek ve üçüncü sınıftan 25 örnek ile ağ test edildiğinde ortalama %97,2 gibi bir test başarı oranı elde edilmiştir.

Şekil 6.2'de genetik algoritma ile yapılan optimizasyon işlemi sonucunda; simülasyon ve test doğruluk değerinin toplamının jenerasyon sayısına bağlı olarak değişimi gösterilmektedir. Grafikteki "maksimum doğruluk" değeri; jenerasyonda elde edilmiş maksimum toplam doğru sınıflanmış örnek sayısını, "popülasyon ortalaması" değeri ('Δ' sembol ile gösterilen) ise her bir jenerasyondaki ortalama toplam doğruluk değerini göstermektedir. Buna göre; her üç sınıftan ağı simüle etmek için alınan toplam örnek sayısı 35, test edilen örnek sayısı toplam 180'dir, yani referans ve test kümesinin toplam sayısı 215'dir. 200 jenerasyon sonucunda bu toplam 215 örnekten 210 tanesi doğru olarak sınıflandırılmıştır. Ağı referans verilerle simülasyonu sonucunda doğru sınıflama başarısı her 3 sınıf için %100 olarak elde edilmiştir. Test aşamasında birinci sınıftan test edilen 125 örnekten 124'ü, ikinci sınıftan test edilen 30 örnekten 28'i ve üçüncü sınıftan test edilen 25 örnekten 23'ü doğru olarak sınıflandırılmıştır.



Şekil 6.2 GRSA ile tiroid verisi sınıflandırma uygulamasında genetik algoritma ile yapılan optimizasyon sonuçları.

Yapılan optimizasyon sonucunda maksimum test başarısı elde etmek için seçilecek referans kümenin belirlenmesiyle birlikte ağ yapısında kullanılan Gauss fonksiyonunun genişliğinin değeri (σ) 1,22 olarak belirlenmiştir. Şayet σ bu değerde alınıp, giriş örnekleri rasgele (yani optimize edilmeden) seçilmezse, yani veri kümesindeki sırasından alınırsa, örneğin birinci sınıf için ilk 25 değer, ikinci sınıf için ilk 5 değer ve üçüncü sınıf için ilk 5 değer referans alınıp ağ simüle ve test edildiğinde ağın eğitime başarısı yine %100 olurken, test aşamasında birinci sınıftan test edilen 125 örnekten 123'ü, ikinci sınıftan test edilen 30 örnekten 19'u ve üçüncü sınıftan test edilen 25 örnekten 17'si doğru olarak sınıflanmaktadır, yani ortalama %88,3 gibi bir başarı elde edilmektedir.

Burada giriş paternleri optimize edilmeden (veri kümesindeki sırasından alınarak) sadece σ değeri değiştirildiğinde ise (yani sadece σ değeri optimize edilirse) yine maksimum %88,3'lük bir test başarısı elde edilmektedir. Çizelge 6.2'de test aşamasında her bir sınıf için GRSA'nın GA ile optimize edilerek ve optimize edilmeden doğru sınıfladığı örnek sayıları gösterilmektedir. Çizelge 6.3'de GRSA'nın optimize edilerek ve optimize edilmeden ulaştığı ortalama test performansları gösterilmektedir.

Çizelge 6.2 Test aşamasında her bir sınıf için doğru sınıflanan örnek sayıları.

$\sigma=1,22$	GRSA+GA	GRSA
1 Sınıf (toplam 125 örnek)	124	123
2 Sınıf (toplam 30 örnek)	28	19
3 Sınıf (toplam 25 örnek)	23	17
Toplam (toplam 180 örnek)	175	159

Çizelge 6.3 Ortalama test başarımları oranları.

$\sigma=1,22$	GRSA+GA	GRSA
Test başarımları	% 97,2	% 88,3

Buna göre optimizasyon sonucunda yaklaşık olarak %8,9 luk bir fark avantaj olarak elde edilebilmektedir. Bu avantaj özellikle veri kümesindeki örnek sayısının az olduğu sınıflar için (2. ve 3. sınıflar) önemli olmaktadır.

Benzer uygulama; iris ve Escherichia Coli verilerinin sınıflandırılması için de denenmiş olup bu uygulamalar için de GRSA'nın optimizasyonu ile sınıflandırma başarısının arttığı gözlemlenmiştir (Yazıcı vd, 2006).

6.2.1 3-boyutlu nesne tanıma sisteminin optimizasyonu

Bu kısımda, daha önceki bölümde anlatılan GRSA ve OSA kullanılarak özellik çıkarmaksızın 3 boyutlu nesne tanıma sisteminin performansını artırmak için optimizasyon işlemi gerçekleştirilmiştir.

GRSA ağıının örnek giriş paternlerini belirlemek için her bir sınıftan veri kümesinde mevcut olan 72 pozdan hangi 6'sı, seçilirse, veri kümesinde geriye kalan kısım için en yüksek test başarımları elde edilir sorusu genetik optimizasyon ile çözülmeye çalışılmıştır. GRSA'nın Genetik algoritma ile optimizasyonu sonucunda belirlenen referans paternler ile ağı eğitildiğinde sistemin eğitime başarımları %100 ve test başarımları ise %90,3 olmaktadır. GRSA (optimize edilmeden) her bir nesnenin 60° lik açı farkına sahip pozları ile simule edildiğinde (toplam 6 örnek ile simule, geriye kalan 66 poz ile ağı test edildiğinde) gösterdiği test performansı %85,1 dir. Buna göre sistemin test aşamasında optimizasyon sonucunda yaklaşık olarak %5,2 lik bir fark avantaj olarak elde edilebilmektedir. GRSA'nın GA ile optimize

edilince ve edilmeden gösterdiği test performansları çizelge 6.4’de gösterilmiştir.

Çizelge 6.4 3-Boyutlu nesne tanıma uygulaması için GRSA ve GRSA+GA’nın test başarımları oranları.

Referans Pozların Sayısı	Test seti için tanıma oranı	
Her bir nesnenin 60° lik açı farkına sahip görüntüleri için ağ simule edildiğinde (toplam 6 örnek ile simule, geriye kalan 66 poz ile ağ test edildiğinde)	GRSA	GRSA+GA
	% 85,1	% 90,3

Yapılan optimizasyon sonucunda maksimum test başarısı elde etmek için seçilecek referans kümenin belirlenmesiyle birlikte ağ yapısında kullanılan gauss fonksiyonun genişliğinin değeri (σ) 0.208 olarak belirlenmiştir.

Aynı uygulama OSA yapısı içinde denendi. OSA’nın GA ile optimize edildiğinde ve edilmeden gösterdiği test performansları çizelge 6.5’de gösterilmiştir.

Çizelge 6.5 3-Boyutlu nesne tanıma uygulaması için OSA ve OSA+GA’nın test başarımları oranları.

Referans Pozların Sayısı	Test seti için tanıma oranı	
Her bir nesnenin 60° lik açı farkına sahip görüntüleri için ağ simule edildiğinde (toplam 6 örnek ile simule, geriye kalan 66 poz ile ağ test edildiğinde)	OSA	OSA+GA
	% 85,7	% 89,8

6.2.2 El yazısı rakam tanıma sisteminin optimizasyonu

Çalışmanın bu aşamasında, nesne tanıma sisteminin optimizasyonu uygulamasına benzer şekilde, özellik çıkarmaksızın el yazısı rakam tanıma sisteminin performansını artırmak için genetik algortima ile optimizasyon işlemi gerçekleştirilmiştir.

Burada amaç, genetik algoritma kullanılarak GRSA ağının örnek giriş paternlerini belirlemek yani her bir sınıftan veri kümesinde mevcut olan 20 örnekten hangi 10’u seçilirse, veri kümesinde geriye kalan kısım için en yüksek test başarısının elde edilebileceğini belirlemektir. GRSA’nın genetik algoritma ile optimizasyonu sonucunda belirlenen referans paternler ile ağ eğitildiğinde sistemin eğitime başarısı %100 ve test başarısı ise %83 olmaktadır. GRSA’nın optimize edilmeden gösterdiği test performansı çizelge 6.6’da gösterilmiştir ve GRSA optimize edilmeden önceki ortalama test başarısı %65 dir. Buna göre optimizasyon sonucunda yaklaşık olarak %18 lik bir fark avantaj olarak elde edilebilmektedir.

GRSA'nın optimize edildiği zaman gösterdiği test performansı çizelge 6.7'de gösterilmiştir. Optimizasyon sonucunda GRSA için sigma değeri ise 57,18 olarak bulunmuştur.

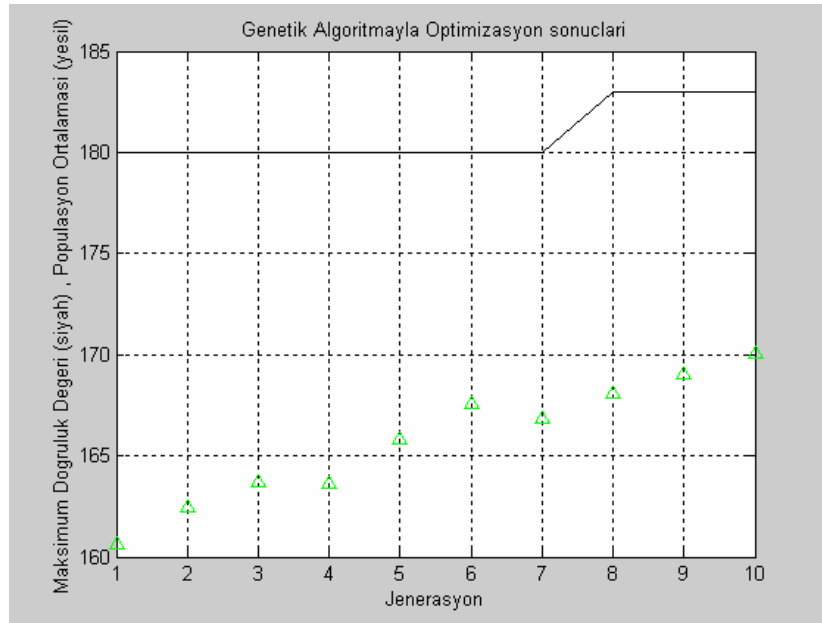
Çizelge 6.6 GRSA ile herbir rakamın tanınma oranı.

Rakamlar	0	1	2	3	4	5	6	7	8	9	Ort.
Test sonuçları	%100	%70	%70	%80	%20	%60	%50	%60	%60	%80	% 65

Çizelge 6.7 GRSA'nın GA ile optimizasyonu sonucunda herbir rakamın tanınma oranı.

Rakamlar	0	1	2	3	4	5	6	7	8	9	Ort.
Test sonuçları	%100	%80	%70	%100	%70	%90	%80	%90	%70	%80	% 83

Şekil 6.3'de el yazısı rakam tanıma uygulamasında GRSA'nın genetik algoritma ile yapılan optimizasyon işlemi sonucunda; simülasyon ve test doğruluk değerinin toplamının jenerasyon sayısına bağlı olarak değişimi gösterilmektedir. Buna göre 10 jenerasyon sonunda 100'ü eğitime datası ve 83'ü de test datası olmak üzere toplam 183 örnek doğru olarak sınıflandırılabilmiştir.



Şekil 6.3 El yazısı rakam tanıma uygulaması için genetik algoritma ile yapılan optimizasyon sonuçları.

Bu aşamadan sonra; el yazısı rakam tanıma uygulaması için GRSA'nın optimizasyonunda kullanılan her bir rakam için mevcut olan 20 örnekten farklı olarak NIST veri kümesinden her bir rakamdan 10'ar adet rakam daha sistemi doğrulama için alınmış ve optimize edilmiş patern tanıma sistemi yeniden test edilmiştir. Optimizasyon yapılmadan yani veri kümesindeki sırasından her bir rakam için 10'ar adet referans veri alınarak eğitilmiş GRSA'nın bu farklı 100 rakam için elde edilmiş doğru sınıflandırma başarıları çizelge 6.8'de gösterilmiştir. Çizelge 6.9'da ise yine optimizasyon aşamasında kullanılmayan farklı 10 rakam için GRSA'nın GA ile optimize edildikten sonraki doğrulama başarıları sonuçları gösterilmiştir. Bu sonuçlar yapılan optimizasyonun %10 luk bir avantaj sağladığını göstermektedir.

Çizelge 6.8 Optimizasyon aşamasında kullanılmayan farklı rakamlar için GRSA optimize edilmeden önce her bir rakamın tanınma oranları.

Rakamlar	0	1	2	3	4	5	6	7	8	9	Ort.
Doğrulama Başarısı	%100	%90	%70	%30	%50	%100	%100	%50	%50	%80	% 72

Çizelge 6.9 Optimizasyon aşamasında kullanılmayan farklı rakamlar için GRSA optimize edildikten (GRSA+GA) sonra her bir rakamın tanınma oranları.

Rakamlar	0	1	2	3	4	5	6	7	8	9	Ort.
Doğrulama Başarısı	%100	%90	%70	%70	%80	%80	%100	%70	%80	%80	% 82

El yazısı rakam tanıma uygulamasının optimizasyonu OSA yapısı içinde gerçekleştirildi. OSA'nın GA ile optimize edilince ve edilmeden gösterdiği test performansları çizelge 6.10 ve çizelge 6.11'de gösterilmiştir. Bu çizelgelerden görüldüğü üzere OSA'nın GA ile optimizasyonu sonucunda ortalama test başarısında %14 lük bir yükselme olmaktadır. Optimizasyon sonucunda OSA için sigma değeri ise 76,23 olarak bulunmuştur.

Çizelge 6.10 OSA ile her bir rakamın tanınma oranı.

Rakamlar	0	1	2	3	4	5	6	7	8	9	Ort.
Test sonuçları	100%	80%	80%	80%	20%	50%	50%	70%	60%	80%	67%

Çizelge 6.11 OSA'nın GA ile optimizasyonu sonucunda herbir rakamın tanınma oranı.

Rakamlar	0	1	2	3	4	5	6	7	8	9	Ort.
Test sonuçları	100%	80%	80%	90%	80%	80%	100%	80%	60%	60%	81%

Daha sonra GRSA için gerçekleştirilen doğrulama işleminin aynısı OSA için de uygulanmıştır. Optimizasyon yapılmadan yani veri kümesindeki sırasından herbir rakam için 10'ar adet referans veri alınarak eğitilen OSA'nın, yeni 100 örnek için elde edilmiş doğru sınıflandırma başarıları çizelge 6.12'de gösterilmiştir. Çizelge 6.13'de ise optimizasyon aşamasında kullanılmayan bu farklı rakamlar için, OSA'nın GA ile optimize edildikten sonraki tanıma başarı oranları gösterilmiştir.

Çizelge 6.12 Optimizasyon aşamasında kullanılmayan farklı rakamlar için OSA optimize edilmeden önce herbir rakamın tanınma oranları.

Rakamlar	0	1	2	3	4	5	6	7	8	9	Ort.
Doğrulama Başarısı	%100	%90	%70	%30	%50	%90	%100	%50	%60	%80	% 72

Çizelge 6.13 Optimizasyon aşamasında kullanılmayan farklı rakamlar için OSA optimize edildikten (OSA+GA) sonra herbir rakamın tanınma oranları.

Rakamlar	0	1	2	3	4	5	6	7	8	9	Ort.
Doğrulama Başarısı	%100	%90	%70	%40	%80	%70	%100	%80	%50	%80	% 76

Elde edilen bu sonuçlara göre farklı sınıflandırma ve patern tanıma uygulamaları için kullanılan GRSA ve OSA yapılarının genetik algoritma ile optimizasyonunun tanıma sistemlerinin test ve doğrulama başarısını önemli ölçüde arttırdığı gözlenmiştir. Ağlardan en yüksek test performansını elde etmek için ağa uygulanacak örnek paternler ve Gauss fonksiyonunun genişliği genetik algoritma ile belirlenmiştir. Veri kümesinden farklı sayıda örnek patern alınarak veya genetik algoritmadaki jenerasyon sayısı artırılarak test başarı oranı arttırılmaya çalışılabilir. Ayrıca bu örnek uygulama farklı problemler içinde denenebilir ve aynı optimizasyon algoritması farklı YSA yapıları içinde kullanılabilir.

7. GRSA KULLANILARAK EL GEOMETRİSİ GÖRÜNTÜLERİNDEN BİOMETRİK PATERN TANIMA

Biometrik özellik tanıma sistemleri insanların; parmak izi, iris, yüz, ses, imza, el geometrisi vs. gibi karakteristiklerini saptayarak bilgisayar sistemleri, veri bankaları gibi ortamlara giriş için kimlik doğrulaması yapmaktadırlar. Biometrik sistemler temelde, kişinin sadece kendisinin sahip olduğu, değiştiremediği ve diğerlerinden ayırt edici olan, fiziksel veya davranışsal bir özelliğinin tanınması amacıyla kullanılmaktadırlar. Ancak bu sistemlerin güvenilir olmalarının yanı sıra pratik olmaları da gerektiği için, kişileri hangi yöntemler ile saptadıkları da önemli bir etkidir. El geometrisi tanıma sistemleri, elin bir kısmının veya tamamının taranması suretiyle yapılmaktadır, kullanıcıların tarama işlemine geçmeden biometrik özelliklerin kayıtlı olduğu sisteme önceden girmeleri gerekmektedir. Kişinin kayıtlı biometrik özellikleri tarama sonuçlarıyla karşılaştırılarak kişinin yetkili olup olmadığı belirlenir. El geometrisi sistemleri, günümüzde girişleri kontrol etmede ve verileri onaylamada, birçok hava alanında, ofislerde, fabrikalarda, nükleer güç merkezlerinde ve yüksek güvenli binalarda kullanılmaktadır.(YingLiang vd, 2004), [8]

Literatürde yapılan el paterni tanıma çalışmalarında taranan giriş görüntülerine filtreleme, kenar belirleme gibi ön işlemler yapılmakta ve daha sonra da parmakların uzunluğu, genişliği gibi pek çok özellik çıkarılarak tanıma işlemi gerçekleştirilmektedir (YingLiang vd, 2004), (Sanchez vd, 2000b), (Faundez, 2005), (Golfarelli vd, 1997), (Gonzalez vd, 2003).

Bu çalışmada el geometrisi paternlerinden kişi tanıma uygulaması, paternlere herhangi bir ön işleme ve özellik çıkarma işlemi yapmadan GRSA ile gerçekleştirilmiştir ve sistemin performansı False Acceptance Ratio (FAR) ve False Rejection Ratio (FRR) değerleri hesaplanarak gösterilmiştir (Polat ve Yıldırım, kabul edildi, basımda-b). Elde edilen sonuçlara göre tasarlanan sistemin dikkate değer bir performans gösterdiği görülmektedir. Aynı uygulama farklı YSA'lar ile denenmiş olup sadece GRSA için yüksek doğrulama (verifikasyon) başarısı gözlemlenmiştir.

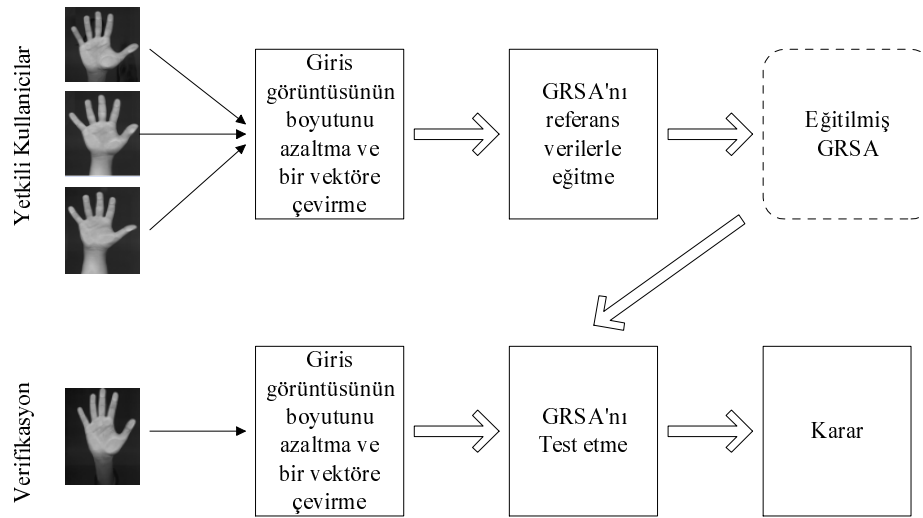
7.1 Metot

Kullanılan veri kümesinde 50 kişinin 10'ar adet el görüntüsü bulunmaktadır [5]. Çalışmada ilk olarak veri kümesindeki 10 kişi sistemde yetkili kişi olarak kabul edilmiş kalan 40 kişi ise davetsiz misafir olarak kabul edilmiştir. Yetkili 10 kullanıcının 7'ser adet el görüntüsü referans görüntü olarak alınarak sistem eğitilmiştir yani toplamda 70 görüntü referans olarak

alınmıştır. Yetkili kullanıcıların geriye kalan 3'er el görüntüsü ise sistemi test için, yani toplamda 30 görüntü test için kullanılmıştır. 40 davetsiz misafirin 2'şer adet el görüntüsü ise, yani toplamda 80 görüntü sistemi doğrulama için kullanılmıştır. Veri kümesindeki el görüntüleri 1021x1403 piksel 8 bitlik gri seviye görüntülerdir. Veri kümesindeki bazı el görüntüleri şekil 7.1'de gösterilmiştir. Ağın hızlı simülasyonu için giriş görüntülerinin her biri 50x71 piksel olarak yeniden boyutlandırıldı ve bu görüntü matrisi 1x3570 olarak vektöre dönüştürülerek sistemin girişine uygulandı. Tasarlanan sistemin blok diyagramı şekil 7.2'deki gibidir.



Şekil 7.1 El görüntü paternlerinden bazı örnekler.



Şekil 7.2 GRSA ile el geometrisi tanıma sistemi.

7.2 GRSA için el geometrisi tanıma işlemindeki aşamalar ve sistemin doğrulama başarısı

Ağın patern katmanında (2. katman) hesaplanan Öklid uzaklığı değerinin büyüklüğü, bilinmeyen görüntünün referans görüntülerden o oranda farklı olduğunu göstermektedir. Bu uzaklık değeri arttıkça denklem 2.2'deki ağırlık fonksiyonunun değeri (W) azalacak, hatta sıfıra yaklaşacaktır. Ağın üçüncü katmanında referans görüntüler için bilinen hedef değerlerle

ağırlıkların çarpımlarının toplamı (A) ve ayrıca sadece ikinci katmanın (yani ağırlıkların) çıkışlarının toplamı (B) işlemleri gerçekleştirilir. Çıkış katmanında ise A değeri B değerine bölünerek çıkış değeri elde edilir. A değerinin hesabında, bilinmeyen giriş görüntüsü ile herbir referans görüntü arasındaki W değeri ne kadar küçük olursa, o W değeri; o referans görüntü için verilen hedef değerle çarpılacağından, sistemin çıkışının hedef değere yakınsaması o derece uzak olacaktır. Tam tersine W değeri ne kadar büyük olursa sistemin çıkışı o referans değer için verilen hedef değere yakın bir değer alacaktır. Bu da bize bilinmeyen giriş görüntüsünün o referansa yakın bir görüntü olabileceğini göstermektedir. Doğrulama aşamasında davetsiz bir misafirin el görüntüsü ile referans görüntüler arasındaki fark büyük olduğundan bütün W değerleri sıfıra yakınsayacaktır. Bu da bize A 'nın sıfıra yakınsayacağını gösterir. Toplamın bu sıfıra yakınsaması da bizim için davetsiz bir misafirin geldiğinin bir göstergesi olmaktadır.

7.3 Simülasyon Sonuçları

Sistemde referans vektörler YSA'ların girişine uygulandı, ağa hedef değerler olarak her bir kişi için bir kod verildi ve ağ bu şekilde simule edildi. Çizelge 7.1'de kullanılan farklı YSA yapıları için elde edilmiş maksimum doğruluk değerleri ve ağların eğitime süreleri gösterilmiştir. Bu çizelgeden görüleceği üzere sadece GRSA davetsiz misafirleri belirlemede iyi bir performans göstermektedir. GRSA'nın eğitim süresi P4 3GHz, 256 MB RAM özelliklerine sahip PC'de ve MATLAB çalışma ortamında 0,14 sn olmaktadır. 50x71 olarak boyutlandırılan giriş görüntüleri için 1 gizli katmanlı ve gizli katmandaki nöron sayısının 10 olduğu MLP ağı için ise hafıza yetmeme uyarısı vermektedir.

Çizelge 7.1 Farklı YSA yapıları için eğitime, test ve doğrulama performansları.

	Eğitime Başarısı(%) (10 yetkili kullacıdan 7'ser adet eğitime görüntüsü alındığında)	Test Başarısı(%) (10 yetkili kullacının farklı 3'er adet test görüntüsü alındığında)	Doğrulama Başarısı (%) (40 davetsiz misafirden 2'ser adet doğrulama görüntüsü alındığında)	Eğitime zamanı (saniye)
RBF	100%	75%	0%	1,17
OSA	100%	93,3%	0%	0,12
GRSA	100%	90%	90%	0,14

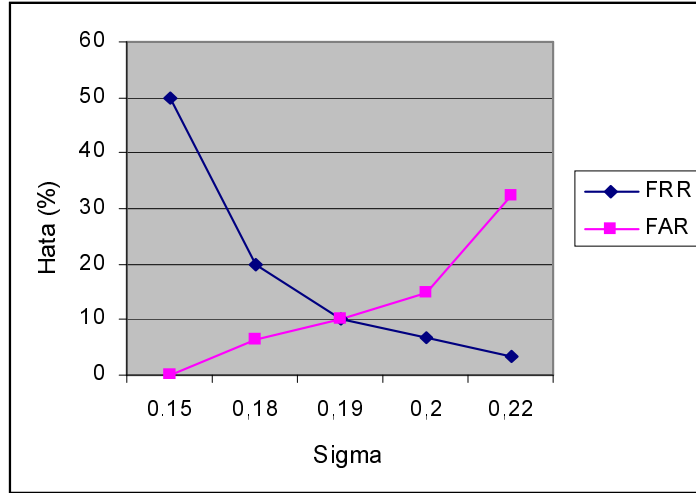
Verifikasyon sistemlerinin başarısı genelde FAR ve FRR değerleri ile gösterilmektedir. FRR: bir yetkili kullanıcının hatalı bir şekilde reddedilme oranı, FAR ise: bir davetsiz misafirin hatalı bir şekilde kabul edilme oranını göstermektedir. Bu iki oran birbirini tamamlayıcı değerlerdir ve bu iki oranın yaklaşık olarak eşit olduğu seviye (EER- Equal Error Rate) optimum seviye olarak kabul edilir ve istenen güvenlik seviyesine bağlı olarak kabul edilebilir bir değerde belirlenir (Sanchez vd, 2000a). Burada W'nın hesabında kullanılan Gauss fonksiyonunun genişliği (σ) sistem performansını önemli bir oranda etkilemektedir. Çalışmada farklı σ değerleri için eğitime, test ve doğrulama performansları belirlenmiştir. Alınan sonuçlar çizelge 7.2'de gösterilmiştir. Ayrıca farklı σ değerleri için FAR ve FRR değerleri hesaplanmıştır (çizelge 7.3'de gösterildiği gibi). Bu sonuçlara göre en yüksek performans FRR ve FAR'ın eşit olduğu $\sigma=0,19$ değerinde elde edilmiştir. Şekil 7.3'de σ değerine bağlı olarak FRR ve FAR değerlerinin değişimi gösterilmiştir. Bu grafikten görüleceği üzere optimum performans EER'nin %10 olduğu σ değerinde elde edilmektedir.

Çizelge 7.2 Farklı σ değerleri için GRSA'nın eğitime, test ve doğrulama performansları

σ (sigma değeri)	Eğitime Başarısı(%)	Test Başarısı(%)	Doğrulama Başarısı (%)
0,15	100%	50%	100%
0,18	100%	80%	93,75%
0,19	100%	90%	90%
0,20	100%	93,3%	85%
0,22	100%	96,6%	67,75%

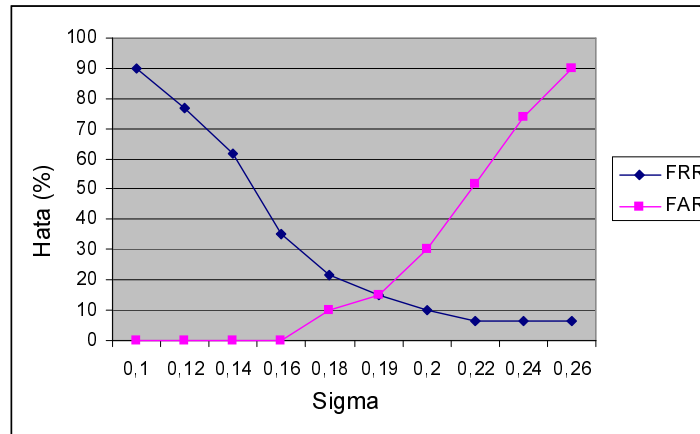
Çizelge 7.3 Farklı σ değerleri için FAR ve FRR sonuçları.

σ (sigma değeri)	FRR(%)	FAR(%)
0,15	50%	0%
0,18	20%	6,25%
0,19	10%	10%
0,20	6,7%	15%
0,22	3,4%	32,25%



Şekil 7.3 Farklı σ değerleri için FAR ve FRR değişimi.

Aynı uygulama sistemdeki yetkili kullanıcı sayısının artabileceği düşünülerek de GRSA kullanılarak yeniden denenmiştir. Buna göre veri kümesindeki 20 kişi sistemde yetkili kişi olarak kabul edilmiş kalan 30 kişi ise davetsiz misafir olarak kabul edilmiştir. Yetkili 20 kullanıcının 7'ser adet el görüntüsü referans görüntü olarak alınarak sistem eğitilmiştir yani toplamda 140 görüntü referans olarak alınmıştır. Yetkili kullanıcıların geriye kalan 3'er el görüntüsü ise sistemi test için, yani toplamda 60 görüntü test için kullanılmıştır. 30 davetsiz misafirin 2'ser adet el görüntüsü ise, yani toplamda 60 görüntü sistemi doğrulama için kullanılmıştır. Elde edilen sonuçlar çizelge 7.4'de gösterilmiştir. Şekil 7.4'de σ değerine bağlı olarak FRR ve FAR değerlerinin değişimi gösterilmiştir. Buna göre optimum performans EER'nin %15 olduğu $\sigma = 0,19$ değerinde elde edilmektedir.



Şekil 7.4 Yetkili kullanıcı sayısının artırılması durumunda farklı σ değerleri için FAR ve FRR değişimi.

Çizelge 7.4 Yetkili kullanıcı sayısının artırılması durumunda farklı σ değerleri için GRSA'nın eğitime, test ve doğrulama performansları.

σ (Sigma)	Eğitime Başarısı (20 yetkili kullacıdan 7'şer adet eğitime görüntüsü alındığında)	Test Başarısı (20 yetkili kullacının farklı 3'er adet test görüntüsü alındığında)	Doğrulama Başarısı (30 davetsiz misafirden 2'şer adet doğrulama görüntüsü alındığında)
0.1	% 100	% 10	% 100
0.12	% 100	% 23.3	% 100
0.14	% 100	% 38.3	% 100
0.16	% 100	% 65	% 100
0.18	% 100	% 78.3	% 90
0.19	% 100	% 85	% 85
0.20	% 100	% 90	% 70
0.22	% 100	% 93.3	% 48
0.24	% 100	% 93.3	% 26
0.26	% 100	% 93.3	% 10

Elde edilen bu sonuçlara göre, GRSA ile el geometrisi görüntülerinden özellik çıkarmaksızın yapılan kişi tanıma işleminin başarısı oldukça yüksektir. Ayrıca tasarlanan sistem ses veya imza gibi farklı biometrik tanıma uygulamalarında da kullanılabilir. Tanıma işleminde görüntülere boyut azaltma dışında herhangi bir ön işleme ve özellik çıkarma işlemi yapılmadan yüksek tanıma oranının elde edilebilmesi sistemin önemli bir özelliğidir.

8. GENELLEŞTİRİLMİŞ REGRESYON SİNİR AĞI'NIN FPGA ile DONANIMSAL GERÇEKLEMESİ

Yazılımsal olarak gerçekleştirilmiş olan patern tanıma uygulamalarında oldukça başarılı sonuçlar veren GRSA'nın, *alan programlanabilir kapı dizileri* ile donanımsal tasarımı gerçekleştirilmiştir. Bu aşamada tasarlanacak sistem içerisindeki birimler bir donanım tanımlama dili olan VHDL yardımıyla ifade edilmiştir. Sistemin tasarımında GRSA yapısı içerisindeki; Öklid mesafesi, üstel fonksiyon (e^{-x}), ve bölme blokları, sabit-noktalı aritmetik (fixed point arithmetic) kullanılarak sentezlenebilir VHDL kodu ile gerçekleştirilmiştir. Sistemin tanımlanma aşamasını takiben derleme ve simülasyon adımları gerçekleştirilmiştir. Sistemden beklenen cevapların elde edilmesiyle beraber sentezleme işlemi gerçekleştirilmiş ve GRSA'nın sayısal olarak gerçekleştirmesindeki donanım gereksinimleri bulunmuştur.

Genelde pattern sınıflandırma uygulamaları yazılımsal olarak gerçekleştirilmektedir. Bununla birlikte, yazılım temelli gerçeklemelerin işlemsel hızı düşük kalmaktadır ve ayrıca uygulama aşamasında bir bilgisayara ihtiyaç duyulacağından, bazı portatiflik istenen çalışmaların gerçekleştirilmesinde zorluklar yaşanabilmektedir (Dongsheng vd, 2004).

Literatürde, farklı yapay sinir ağı yapılarının FPGA'ler ile donanımsal gerçekleştirmesinin yapıldığı çalışmalar bulunmaktadır (Dongsheng vd, 2004), (Savich vd, 2007), (Minchin vd, 1999), (Karabıyık vd, 2006), (Yang vd, 2003), (Maeda vd, 2005).

(Savich vd, 2007) de çok katmanlı algılayıcı yapay sinir ağının FPGA gerçekleştirmesinde aritmetik gösterimlerin etkisi anlatılmıştır. (Minchin vd, 1999) çalışmasında FPGA ler ile olasılıksal sinir ağı dizaynı anlatılmaktadır. (Karabıyık vd, 2006) bir Wavelet yapay sinir yapısının donanımsal gerçekleştirmesini anlatmaktadır. (Yang vd, 2003) çalışmasında ise bir Radyal temelli fonksiyon sinir ağının gerçek zamanlı uygulamalar için donanımsal gerçekleştirilmesi anlatılmaktadır.

Literatürde, GRSA ağının FPGA gerçeklemesine yönelik sadece bir çalışmaya rastlanmıştır (Lazaro vd, 2004). Çalışmada VHDL kodu içerisine, Matlab yazılımı içindeki GRSA kodunun çevirisi gerçekleştirilmektedir. Ayrıca yapılan çalışma gerçek bir problem üzerine denenmemiş olup, sadece Matlab ve VHDL simülasyonları ile elde edilen sonuçlar arasındaki fark karşılaştırılması yapılmıştır. Ayrıca GRSA yapısı içindeki üstel işlem bloğu bir LUT (Look-up table) kullanılarak tasarlanmıştır. Ancak bu yaklaşım farklı uygulamalarda LUT'un bit sınırlaması yüzünden yeterli olmayabilir, yani uygulama içerisinde daha fazla bit uzunluğuna sahip üst alma işlemi gerekli olduğunda yeniden bir LUT oluşturmak

gerekecektir.

Bu çalışmada GRSA'nın donanımsal gerçeklemesi özellikle patern sınıflandırma uygulaması için tasarlanmıştır. Ancak GRSA'nın kullanıldığı diğer uygulamalar içinde kullanılabilme esnekliğine sahiptir. Simülasyonlar *iris bitkisi* ve *tiroid hastalığı* sınıflandırma ve *3-boyutlu benzer nesne* tanıma problemleri için gerçekleştirilmiştir ve yüksek sınıflandırma başarısı elde edilmiştir. Çalışmada üst alma işlemi ise Taylor serisi yaklaşıklığı kullanılarak gerçekleştirilmiştir. Üst alma işleminin hassasiyeti sistemden istenen doğruluk seviyesine göre belirlenebilmektedir. Örneğin üst alma (e^{-x}) bloğunun çıkış hassasiyeti iris verisi uygulamasında 2^{-26} , tiroid verisi uygulamasında ise 2^{-34} olarak sistem tasarlanmıştır.

8.1 GRSA'nın Sayısal Donanım Tasarımı

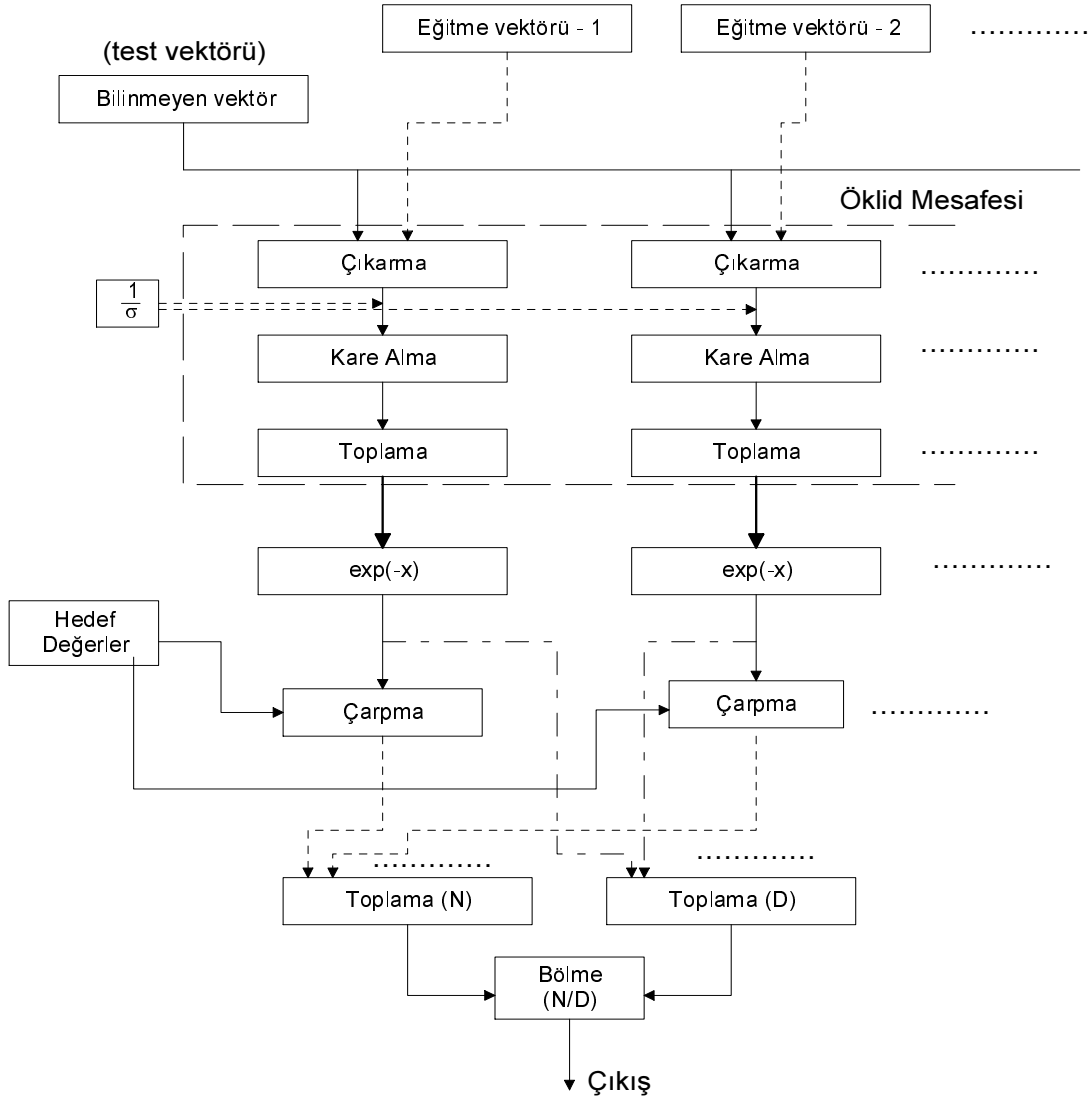
GRSA, sayısal olarak VHDL programı kullanılarak gerçekleştirilmiştir. Sentez işlemi Xilinx firmasının 2 milyon lojik kapıya sahip Spartan3 serisi çipi için gerçekleştirilmiştir. GRSA'nın donanımsal gerçeklemesindeki işlemsel akış diyagramı şekil 8.1'de gösterilmiştir.

Sistemde eğitme vektörlerinin uzunluğu ve sigma değeri 8 bit, hedef değerlerin boyutu ise 3 bit'dir ve bu değer sınıf sayısına bağlı olarak arttırılabilir veya azaltılabilir. Sistemde üstel alma bloğu ve çarpma bloklarında; bit değerlerinin çok büyümemesi için kesme işlemi uygulanmıştır. Sistemin çıkışı ise test edilen vektörün hangi sınıfa ait olduğuna karar vermemizi sağlamaktadır ve 3 bit uzunluğundadır.

Sistemde, iris verisi sınıflandırma uygulaması için her biri 4 (özellik değerlerinin sayısı) elemandan oluşan 30 eğitme vektörü ve herbir eğitme vektörü için hedef değerler VHDL kodu içerisinde sabit dizi (array constant) olarak tanımlanmaktadır. Bu şekilde tanımlama FPGA içerisinde alan etkisi (Area efficient) oluşturmadığından tercih edilmiştir. Tiroid verisi sınıflandırma uygulamasında her biri 5 elemandan oluşan 55 eğitme vektörü kullanılarak sistem simüle edilmiştir. 3-boyutlu benzer nesne tanıma uygulamasında ise her biri 3 elemandan oluşan 96 eğitme vektörü kullanılarak sistem simüle edilmiştir.

GRSA'nın donanımsal olarak gerçeklenmesindeki en büyük zorluklardan biri; her bir referans veri sayısı kadar sistemde Öklid mesafesi ve e^{-x} (üstel fonksiyon) hesabı yapılması ve bu hesapların birbirleriyle senkron bir şekilde gerçekleşmesinin gerekmesidir. Bunun için VHDL kodu içerisinde bu bloklar *fonksiyon* olarak tanımlanmıştır, herbir referans veri için üstel bloğunun çıkışında elde edilecek sinyaller şekil 8.1'den de görüleceği üzere, hem doğrudan toplanacağından hemde hedef değerlerle çarpılıp toplanacağından, bu toplam değerlerin

atanacağı sinyaller dizi olarak tanımlanarak sistem tasarlanmıştır. Buradaki dizilerin boyutu referans vektörlerin sayısı kadar olmaktadır.



Şekil 8.1 GRSA'nın işlemsel akış diyagramı

8.1.1 Üs alma bloğunun tasarımı

GRSA'nın örnek (patern) katmanında yer alan üstel fonksiyon işlemi FPGA gerçekleştirilmesinde Taylor serisi yaklaşıklığı kullanılarak tasarlanmıştır. Taylor serisi yaklaşıklığı aşağıdaki gibidir;

$$e^x = 1 + \frac{x}{1!} + \frac{x^2}{2!} + \frac{x^3}{3!} + \dots + \frac{x^n}{n!} \quad (8.1)$$

Burada n serideki derece sayısıdır. Tasarlanan yapıda ise $e^{-x} = \frac{1}{e^x}$ alınarak hesaplanmıştır.

Çünkü seçilen n değeri için olması gereken sonuca e^{-x} 'in Taylor serisi ($e^{-x} = 1 - \frac{x}{1!} + \frac{x^2}{2!} - \frac{x^3}{3!} + \dots + \frac{x^n}{n!}$) yaklaşımına göre daha iyi yakınsamaktadır. e^{-x} işlemi sonucunda 0 ila 1 arasında sayılar oluşmaktadır. İris verisi uygulaması için n değeri 4 seçilmiş, tiroid verisi için ise sınıflandırma başarısını daha da arttırdığı gözlenerek 5 olarak belirlenmiştir. Ancak bu belirleme sistemin donanımsal gereksinimlerini arttırmaktadır. Ayrıca donanımı küçültmek amacıyla işlemsel olarak sonucu yani sınıflandırma performansını etkilemediği için Taylor serisi yaklaşıkliğinde 3. ve 4. derecelerin paydasındaki $3! (=6)$ değeri yerine 4 ve $4! (=24)$ değeri yerinde 16 alınmıştır. Bunu yapmaktaki amaç, bu sayıların 2' nin kuvveti olmasıdır, böylece sadece öteleme yapılarak bu işlem gerçekleştirilebilmektedir.

Tasarımda iris verisi uygulaması için bu üstel işlemi sonucunda 2^{-26} ya kadar küçük sayılar elde edilebilmektedir ve bu değer istenen hassasiyete göre yeniden belirlenebilir. Tiroid verisi uygulaması için 2^{-34} , nesne tanıma uygulaması için ise 2^{-10} olarak bu hassasiyet değerleri belirlenmiştir. Buna sınıflandırma başarısı göz önüne alınarak karar verilmiştir. Farklı uygulamalarda e^{-x} 'in hesabı Öklid mesafesi değerinin büyük bir değer olmasına bağlı olarak e^{-50} gibi çok küçük değerlerde olabilmektedir. Bu işlem fixed point aritmetik kullanılarak rahatlıkla gerçekleştirilebilmektedir.

8.1.2 Bölme bloğunun tasarımı

Tasarımda $e^{-x} = \frac{1}{e^x}$ işleminde ve çıkış katmanında fixed point bölme işlemi mevcuttur.

Bölme bloğu çıkarıcı, karşılaştırıcı ve sayıcı kullanılarak tasarlanmıştır. Bölme işlemi; bölen sayı değerinin bölünen sayı değeri içinde kaç adet olduğu bulunarak gerçekleştirilmektedir. Bölme işleminde bölünen sayı bölen sayıdan küçük olabileceğinden, bölünen sayının LSB (least significant bit-düşük değerlikli bit) bitinin yanına istenen hassasiyete göre "0" biti ekleme işlemi yapılmaktadır. Örneğin iris verisi sınıflandırma uygulaması için üst alma kısmındaki bölme işleminde bölünen sayının yanına 26 adet "0" biti eklenmiştir.

8.2 Simülasyon ve Sentez Sonuçları

Tasarlanan sistemi test etmek için farklı veri kümeleri kullanılmıştır. Sistemin performansı ilk olarak literatürde (Karthikeyan vd, 2006), (Salvini vd, 2000) sınıflandırma problemlerinde sık kullanılan iris verisi [6] kullanılarak denenmiştir. Uygulamada üç farklı iris bitkisinin verilen 4 özellik değerine göre sınıflandırılması amaçlanmaktadır. Veri kümesinde her bir sınıftan 50 örnek olmak üzere toplam 150 örnek bulunmaktadır. GRSA'nı eğitmek amacıyla her bir sınıftan 10 örnek, yani toplamda 30 örnek alınmıştır. Uygulamada Matlab yazılımı ve FPGA simülasyonu sonucunda %100 eğitime başarısı elde edilmiştir. Test verileri için ise Matlab yazılımı kullanarak 120 örnekten 117'si, VHDL kodunun Modelsim simülasyonu sonucunda ise 120 örnekten 116'sı doğru olarak sınıflandırılmıştır. Çizelge 8.1'de Matlab ve Modelsim simülasyonları sonucu elde edilen test performansları gösterilmiştir. (iris verisi sınıflandırma uygulamasında bütün özellik vektörlerinin değeri 10 ile çarpılmıştır.)

Çizelge 8.1 İris verisi için Matlab ve FPGA gerçeklemelerinin test performansları.

	1. Sınıf	2.Sınıf	3.Sınıf	Toplam (Sınıflandırma Başarısı)
Matlab	40	39	38	117 (97.5 %)
FPGA	40	36	40	116 (96.6 %)

Bir başka patern sınıflandırma uygulaması olarak üç farklı tiroid hastalığının verilen 5 özellik değerine göre sınıflandırılması problemi için tasarlanan sistem test edilmiştir. Veri kümesinde [6] birinci sınıftan 150 örnek, ikinci sınıftan 35 ve 3. sınıftan 30 olmak üzere toplam 215 örnek bulunmaktadır. GRSA'nın eğitmek amacıyla birinci sınıftan 25 örnek, 2. sınıftan 15 örnek ve 3. sınıftan 15 örnek yani toplamda 55 örnek alınmıştır. Uygulamada Matlab simülasyonu sonucunda %100, FPGA simülasyonu sonucunda %85,4 eğitime başarısı elde edilmiştir. Test verilerinin sınıflandırılması için ise Matlab kullanarak test edilen toplam 160 örnekten 145'i, Modelsim simülasyonu sonucunda ise 160 örnekten 141'i doğru olarak sınıflandırılmıştır. Çizelge 8.2'de Matlab ve Modelsim simülasyonları sonucu elde edilen test performansları gösterilmiştir. Her iki uygulama için sigma değeri(σ) 1 olarak alınmıştır.

Çizelge 8.2 Tiroid verisi için Matlab ve FPGA gerçeklemelerinin eğitime ve test performansları.

		1. Sınıf	2.Sınıf	3.Sınıf	Toplam (Sınıflandırma Başarısı)
Eğitime	Matlab	25	15	15	55 (100 %)
	FPGA	25	10	12	47 (85.4 %)
Test	Matlab	115	18	12	145 (90.6 %)
	FPGA	118	14	9	141 (88.1 %)

Sistemin girişine gelen test verisinin, her bir *referans* veri ile arasındaki Öklid mesafesinin hesaplanması ve e^{-x} işleminin gerçekleştirilmesi için geçen süre 100 ns dir, iris verisi için toplam 30 adet referans vektör kullanıldığından hepsinin hesaplanması için geçen süre 3 μ s olmaktadır. Daha sonra her bir üstel değerin birbirleriyle toplanması ve aynı zamanda bu değerlerin hedef değerlerle çarpılıp toplanması ve çıkış katmanındaki bölme işlemi için geçen süre de 100 ns olmaktadır. Dolayısıyla toplam süre 3,1 μ s olmaktadır. Tiroid verisi sınıflandırma uygulamasında ise sistemin girişine gelen bir test verisinin hangi sınıfı ait olduğunu belirlenmesi için geçen süre; 55 adet referans vektör olmasına bağlı olarak 5,5 μ s de bütün e^{-x} lerin hesap edilmesi ve toplamda 5,6 μ s olmaktadır.

Tasarlanan GRSA donanımının simülasyonu, bölüm 5’de yazılımsal olarak gerçekleştirilmiş olan 3-boyutlu benzer nesnelerin tanınması uygulaması içinde gerçekleştirilmiştir. Tanıma işlemi için 3 boyutlu ve şekilleri benzer 8 nesnenin 2 boyutlu poz görüntülerinden (Şekil 5.7) renk özellikleri çıkarılmış ve bu özellik vektörleri kullanılarak nesnelerin sınıflandırılması gerçekleştirilmişti (R-G-B tonlarının ortalama değerleri alınarak her bir poz için 3 farklı özellik değeri çıkarılmıştı).

Yazılımsal olarak elde edilen sonuçlar çizelge 5.6’da gösterildiği üzere orijinal veri kümesinden 30° dönme aralıkları ile çekilmiş pozlar yani toplamda 12 poz referans olarak alındığında ve geriye kalan pozlarla ağ test edildiğinde elde edilen ortalama tanıma oranı %93,3, eğitime başarısı ise % 100 olarak bulunmuştu. Modelsim simülasyonları sonucunda ise tasarlanan sistemin test performansı ortalama %78,5, eğitime başarısı ise ortalama %97,9

olmaktadır. Çizelge 8.3’de Matlab ve Modelsim simülasyonları sonucu her bir nesne için elde edilen test ve eğitme performansları gösterilmiştir. Bu uygulamada bir test vektörünün sınıflandırılması için geçen süre ise 96 adet referans vektör olmasına bağlı olarak 9,7 μ s olmaktadır.

Çizelge 8.3 3-boyutlu benzer nesne tanıma uygulaması için Matlab ve FPGA gerçeklemelerinin eğitme ve test performansları.

	FPGA		MATLAB	
	Eğitme	Test	Eğitme	Test
Nesne–1	%100	%80	%100	%95
Nesne–2	%100	%76,6	%100	%88,3
Nesne–3	%91,6	%70	%100	%70
Nesne–4	%100	%80	%100	%100
Nesne–5	%100	%81,6	%100	%98,3
Nesne–6	%100	%85	%100	%100
Nesne–7	%100	%78,3	%100	%88,3
Nesne–8	%91,6	%76,6	%100	%90
Ortalama	%97,9	%78,5	%100	%93

Matlab çalışma ortamında GRSA’nın bir test vektörününün hangi sınıfa ait olduğunu belirleme süresi P4 3GHz, 256 MB RAM özelliklerine sahip bir PC de 0,20 sn olmaktadır. Bu sonuçlara göre GRSA’nın FPGA tasarımı, Matlab yazılımına göre oldukça yüksek işlemsel hız avantajına sahiptir.

Simülasyon aşamasında, iris verisi için yapılan bir test işleminde zaman diyagramı şekil 8.2 deki gibi olmaktadır. Bu deneme, ikinci sınıfa ait bir örneğin dört özellik değeri girilerek gerçekleştirilmiştir. Ağın çıkışında görülen ikilik tabandaki “010” değeri, test edilen bu verinin 2. sınıfa ait olduğunu göstermektedir. Tiroid verisi için örnek bir test işleminin zaman diyagramı ise şekil 8.3’deki gibi olmaktadır. Bu deneme de birinci sınıfa ait bir örneğin beş özellik değeri girilerek gerçekleştirilmiştir. Ağın çıkışında görülen “001” değeri, test edilen bu verinin 1. sınıfa ait olduğunu göstermektedir.

Uygulamalarda GRSA'nın donanımsal tasarımının Matlab'a göre sınıflandırma başarısı bir miktar düşük olmaktadır. Bunun nedeni üstel hesap kısmında yapılan Taylor serisi yaklaşıklığının kullanılmasıdır. Şayet yaklaşıklıkta, serinin daha yüksek bileşenleri alınırsa sınıflandırma performansı yükselecektir, ancak bu durumda donanımsal kaynak ihtiyacı da artacaktır.

İris verisi sınıflandırma problemi için tasarlanan GRSA'nın FPGA gerçeklemede Spartan3-xc3s2000 çipi için elde edilen sentez raporu çizelge 8.4'de gösterilmiştir. Sistemin donanım gereksinimleri, giriş konfigürasyonları ve eğitme vektörlerinin sayısı ile değişmektedir.

Aşağıdaki çizelge 8.4'de verilen sonuçlar 8 bitlik 4 giriş, her bir elemanı 8 bit olan 30 eğitme vektörü ve kontrol girişleri için elde edilmiştir. Tiroid verisi uygulaması için girişlerin sayısı 5 ve eğitme vektörlerinin sayısı 55 dir, ayrıca taylor serisinin derecesi 5 olarak alınmıştır. Bu sebeplerden dolayı tiroid verisi sınıflandırma uygulamasında seçilen çip için donanım kaynakları yetmemektedir. Bu uygulamada sentez işleminin başlangıç özet raporu dikkate alındığında donanım kaynaklarının yaklaşık olarak 2 katına çıktığı anlaşılmaktadır. Nesne tanıma uygulaması için ise yine sentez işleminin başlangıç özet raporu dikkate alındığında donanım gereksinim ihtiyacının iris uygulaması için gerekli olandan yaklaşık olarak 4 kat fazla olduğu anlaşılmaktadır. (Nesne tanıma uygulaması için girişlerin sayısı 3 ve eğitme vektörlerinin sayısı 96, taylor serisinin derecesi ise 5 olarak alınmıştır.)

Çizelge 8.4 30 eğitme vektörü ve 4 giriş için donanım gereksinim sonuçları.

Xilinx Spartan3 xc3s2000	Mevcut	Kullanılan	Kullanım Yüzdesi
Dilimlerin sayısı	20480	17616	% 86
Dilimlerdeki Flip-Flopların sayısı	40960	1322	% 3
4 girişli LUT'ların sayısı	40960	24493	% 59
IOB sayısı	489	33	% 6
18x18 Çarpıcıların sayısı	40	28	% 70
Saat (Clock)	8	1	%12

Sonuç olarak çalışmanın bu aşamasında özellikle patern sınıflandırma uygulamalarında kullanılabilecek olan GRSA'nın sayısal olarak tasarımı VHDL ile tanımlanarak gerçekleştirilmiştir. Geliştirilen sistem farklı veri kümesi sınıflandırma uygulamaları için denenmiş ve yüksek tanıma performansı elde edilmiştir. Büyük boyutlu veri kümeleri için eldeki donanım imkânlarının yetersizliği nedeniyle sentezleme işlemi gerçekleştirilememiştir. Tasarlanan sistem farklı sınıflandırma problemleri için kolaylıkla modifiye edilebilecek yapıdadır. Simülasyon sonuçları sistemin sınıflandırma başarısı yönünden yazılım temelli çözüme yakın sonuçlar verdiğini göstermektedir. Sistemin işlemsel hız avantajı ise bir PC üzerindeki yazılım temelli çözüme göre oldukça yüksektir. Teknolojinin ilerlemesiyle birlikte daha yüksek kapasiteli ve gelişmiş FPGA tümleşik devrelerin kullanılması durumunda tasarlanan bu sistem daha da avantajlı olarak gerçekleştirilebilecektir.

SONUÇLAR

Bu çalışma sonucunda, mevcut olan nesne tanıma ve el geometrisi paterni tanıma uygulamalarındaki önışleme ve özellik çıkarma işlemlerine gerek kalmadan yüksek tanıma performansı sağlanmıştır. Buna göre gerçekleştirilen topolojinin, özellikle 2 boyutlu paternlerden özellik çıkarmanın karmaşık ve güç olduğu tanıma uygulamaları için kullanılabileceğini göstermektedir.

Ayrıca çalışmada paternleri sınıflandırmada kullandığımız ve yüksek test başarısı elde ettiğimiz GRSA ve OSA yapılarının test performanslarını maksimum yapacak, radyal fonksiyon genişliğinin değeri ve veri kümelerinde mevcut olan özellik vektörlerinin hangilerinin referans olarak alınacağını genetik algoritma ile belirlenmiştir. Özellikle bu yöntem ile özellik çıkarmaksızın el yazısı rakam tanıma uygulamasında tanıma performansının oldukça arttırdığı bulunmuştur. Önerilen algoritma; hem farklı veri kümeleri hem de farklı ağ yapıları için de denenmiş ve bütün uygulamalarda test performansının arttığı gözlenmiştir. Buna göre önerilen bu optimizasyon yöntemi, özellikle önışleme ve özellik çıkarmanın güç olduğu patern tanıma uygulamalarında sistem performansını artırma avantajını yaratmaktadır.

Çalışmada gerçekleştirilen el geometrisi paternlerinden kişi tanıma uygulaması, yine paternlere herhangi bir ön işleme ve özellik çıkarma işlemi yapmadan GRSA ile gerçekleştirilmiş ve sistemin yüksek bir doğrulama başarısı gösterdiği bulunmuştur. Elde edilen sonuçlara göre tasarlanan sistem için; mevcut olan el paterni tanıma çalışmalarındaki patern görüntülerine filtreleme, kenar belirleme gibi ön işlemler ve daha sonra da parmakların uzunluğu, genişliği gibi pek çok özellik çıkarma işlemine gerek kalmamaktadır.

Çalışmada gerçekleştirilen patern tanıma uygulamalarında oldukça başarılı sonuçlar veren GRSA'nın, FPGA'ler ile donanımsal tasarımı gerçekleştirilmiştir. Literatürde mevcut olan GRSA'nın FPGA gerçeklemesine yönelik çalışmadan farklı olarak; GRSA yapısı içerisindeki Öklid mesafesi bloğu, Gauss fonksiyonu ve bölme blokları sabit-noktalı aritmetik kullanılarak sentezlenebilir VHDL kodu ile gerçeklenmiş ve simülasyon aşamasında dizayn edilen sistemin performansı sınıflayıcı uygulamalarında yaygın olarak kullanılan farklı veri kümeleri için denenmiştir. Ayrıca sistem farklı giriş sayısına ve istenen çıkış hassasiyetine göre yeniden konfigüre edilebilecek esnek bir yapıya sahiptir.

Bu çalışma boyunca edinilen tecrübelerle göre bu konuda yapılacak çalışmalar için şu önerilerde bulunulabilir: Tasarlanan özellik çıkarmaksızın patern tanıma sisteminin

performansı el yazısı harf, desen, koku tanıma veya parmak izi, imza, iris gibi gerçek zamanlı yapılması tercih edilen farklı patern tanıma uygulamaları için denenebilir. Ayrıca patern tanıma sisteminin optimizasyonu Tabu Araştırma, Karınca Koloni veya Diferansiyel Gelişim algoritmaları gibi farklı algoritmalar ile gerçekleştirilip Genetik Algoritma ile elde edilen sonuçlarla performans karşılaştırmaları yapılabilir. GRSA'nın sayısal olarak gerçekleştirilmesinde ihtiyaç duyulacak olan donanım gereksinimlerini ve işlemsel hızı azaltıp azaltmayacağını belirlemek için özellikle yapı içerisindeki bölme işlemleri, Newton-Raphson, Restoring veya Non-Restoring bölme algoritmaları gibi farklı metotlar ile gerçekleştirilebilir. Yine benzer şekilde donanım gereksinimlerini ve işlemsel hızı azaltıp azaltmayacağını belirlemek için sabit noktalı aritmetik yerine kayan noktalı aritmetik (floating point arithmetic) kullanılarak sistem dizayn edilebilir. Ayrıca GRSA'nın donanımsal gerçekleştirilmesi “alan programlanabilir analog diziler” (field programmable analog array-FPAA) ile gerçekleştirilip analog devrelerin sağladığı düşük güç tüketimi ve yüksek hız avantajları elde edilebilir.

Gerçekleştirilmiş bu çalışmanın farklı bir yaklaşım olarak gelecekteki çalışmalara ve bu konuda çalışanlara faydalı olmasını dilerim.

KAYNAKLAR

- B. Lee and N. Burgess, (2003) "Some Results on Taylor-series Function Approximation on FPGA," 37th Asilomar Conference on Sig. Sys. and Comps., Nov. 2003.
- Bennis, A.; Tull, M., (2005) "Redundant binary based fixed-point divider". Canadian Conference on Electrical and Computer Engineering, 1–4 May 2005, Page(s):1316 – 1319.
- Brown S., Rose J., (1996) "FPGA and CPLD Architectures: A Tutorial", IEEE Design and Test of Computers, vol. 13, no. 2, pp. 42–57.
- Chandni Patel, T. Srikanthan and Sangeetha Narayan, (2002) "A Rotation-invariant Embedded Pattern Recognition System", IEEE International Conference on Industrial Technology, Vol. 1, pp. 88–92.
- Comes,B.; Kelemen, A., (2003) "Probabilistic neural network classification for microarraydata". Proceedings of the International Joint Conference on Neural Networks, Volume 3, 20-24 July 2003 Page(s):1714 – 1717.
- D. F. Specht, (1991a) "A general regression neural network". IEEE Trans. Neural Networks, vol. 2, pp. 568–576.
- D. F. Specht, (1991b) "Enhancements to probabilistic neural network". Proc.of Int. Joint Conf. on Neural Network, vol. 1, pp. 761–768.
- D'Mello, D.R., Gulak, P.G., (1998) "Design Approaches to Field-Programmable Analog Integrated Circuits", Analog Integrated Circuits And Signal Processing, Vol. 17, pp. 7–34.
- Daewon L., Jaewook L.,(2007) "Domain described support vector classifier for multi-classification problems". Pattern Recognition, Volume 40, Issue 1, Pages 41–51.
- de Lacerda, E. G. M., de Carvalho, A. C. P. L. F., & Ludermir, T. B. (2000). "Evolutionary optimization of RBF networks". Sixth Brazilian Symposium on Neural Networks, 219–224.
- Deschamps, Jean-Pierre / Bioul, Gery J. A. / Sutter, Gustavo D.,(2006) "Synthesis of Arithmetic Circuits: FPGA, ASIC and Embedded Systems". 1st Edition–2006. 576 Pages, John Wiley & Sons.
- Dongsheng Shen, Lianwen Jin, and Xiaobin Ma, (2004) "FPGA Implementation of Feature Extraction and Neural Network Classifier for Handwritten Digit Recognition". Lecture Notes in Computer Science, vol 3173, pp 988–995.
- Duvivier, D.; Preux, P.; Fonlupt, C.; Robilliard, D.; Talbi, E.-G.,(1998) "The fitness function and its impact on local search methods". IEEE International Conference on Systems, Man, and Cybernetics. Volume 3, 11–14 Oct. 1998 Page(s):2478 – 2483.
- Eric W.M. Lee, Y.Y. Lee, C.P. Lim, C.Y. Tang, (2006) "Application of a noisy data classification technique to determine the occurrence of flashover in compartment fires". Advanced Engineering Informatics, Volume 20, Issue 2 , April 2006, Pages 213–222.
- Fasel, B., (2002) "Head-pose invariant facial expression recognition using convolutional neural networks". Proc. of Fourth IEEE International Conference on Multimodal Interfaces. 14–16 Oct. 2002, 529 – 534.
- Faundez-Zanuy, M., (2005) "Biometric Verification of Humans by Means of Hand Geometry". International Carnahan Conference on Security Technology, 11–14 Oct. 2005

Page(s):1 – 7.

Freeman, James A.; Skapura, David M., (1991) “Neural Networks: Algorithms, Applications and Programming Techniques.”, Addison-Wesley.

Funabiki, N.; Yoda, M.; Kitamichi, J.; Nishikawa, S., (1999) “A gradual neural network approach for FPGA segmented channel routing problems”. IEEE Transactions on Systems, Man and Cybernetics. Vol. 29, Issue 4, Aug. 1999 Page(s):481 – 489.

Gaborit, L.; Granado, B.; Garda, P., (1999) “Evaluating microprocessor multimedia extensions for the real-time simulation of RBF networks”. Proceedings of the Seventh International Conference on Microelectronics for Neural, Fuzzy and Bio-Inspired Systems. 7-9 April 1999 Page(s):217 – 221.

Goh, A.T.C. (2002), "Probabilistic neural network for evaluating seismic liquefaction potential", Canadian Geotechnical Journal, Vol. 39, pp.219–32.

Goldberg D. E., (1989) “Genetic algorithm in search, optimization, and machine learning”. Addison-Wesley.

Golfarelli, M.; Maio, D.; Malton, D., (1997) “On the error-reject trade-off in biometric verification systems”. IEEE Transactions on Pattern Analysis and Machine Intelligence, Volume 19, Issue 7, July 1997 Page(s):786 – 796.

Gonzalez, S.; Travieso, C.M.; Alonso, J.B.; Ferrer, M.A., (2003) “Automatic biometric identification system by hand geometry”. IEEE International Carnahan Conference on Security Technology, 14–16 Oct. 2003, Page(s):281 – 284.

Granado, B.; Pinna, A.; Baborit, L.; Garda, P., (1999) “MAHARADJA: an embedded system for the real time execution of GRBF networks”. Proceedings of the Seventh International Conference on Microelectronics for Neural, Fuzzy and Bio-Inspired Systems, MicroNeuro '99. 7-9 April 1999, Page(s):409 – 414.

Guoyu Zuo, Wenju Liu and Xiaogang Ruan, (2004) “Genetic Algorithm Based RBF Neural Network for Voice Conversion. Proc. of WCICA, vol.5, 2004, p 4215- 4218.

Hatanaka, T., Kondo, N., Uosaki, K., (2003), “Multi-Objective Structure Selection for Radial Basis Function Networks Based on Genetic Algorithm”, Evolutionary Computation, Volume 2, Issue , 8-12 Dec. 2003 Page(s): 1095 – 1100.

Haykin S., (1994) “Neural networks - A comprehensive foundation”, Macmillan.

Heimes, F.; van Heuveln, B., (1998) “The normalized radial basis function neural network”. IEEE International Conference on Systems, Man, and Cybernetics, Vol. 2, 11–14 Oct. 1998, Page(s):1609 – 1614.

Huang J., Tahoori M.B., Lombardi F., (2005) “Fault Tolerance of Switch Blocks and Switch Block Arrays in FPGA”. IEEE Trans. on VLSI Syst. 13(7): 794–807.

Hwang T-T.; Owens, R.M.; Irwin, M.J.; Kuo Hua Wang , (1994) “Logic synthesis for field-programmable gate arrays”. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, Volume 13, Issue 10, Oct. 1994, Page(s):1280 – 1287.

Irie, T.; Katou, H.; Maeda, H. , (2003) “Object recognition from stereo images using synergetics and combinatorial optimization”. SICE 2003 Annual Conference Volume 1, 2003 Page(s):20 – 25.

J  r  mie Detrey; Florent de Dinechin, (basımda, online kullanılabilir) “Parameterized floating-point logarithm and exponential functions for FPGAs”. Microprocessors and Microsystems.

Jiangjian Xiao and Mubarak Shah, (2004) “Automatic Target Recognition Using Multi-View Morphing”, Conference of SPIE Automatic Target Recognition IV, April 13–15, Orlando, USA.

Jinghui Zhong; Xiaomin Hu; Jun Zhang; Min Gu, (2005) “Comparison of Performance between Different Selection Strategies on Simple Genetic Algorithms”. Proceedings of the 2005 International Conference on Computational Intelligence for Modelling, Control and Automation, and International Conference on Intelligent Agents, Web Technologies and Internet Commerce. Vol. 2, 28–30 Nov. 2005 Page(s):1115 – 1121.

K. Basterretxea, J.M. Tarela, I. del Campo, (2006) “Digital Gaussian membership function circuit for neuro-fuzzy hardware”. Electronics Letters, Vol. 42, Issue 1, p. 44–46.

Karabiyik A., A. Savran., (2006) “Hardware Implementation of a Wavelet Neural Network Using FPGAs”. Lecture Notes in Computer Science. pp. 1095–1104.

Karabo  a, D., (2004) “Yapay Zeka Optimizasyon Algoritmaları”, Atlas Yayınevi, İstanbul.

Karthikeyan B., S. Gopal, S. Venkatesh ,(2006) “ART 2—an unsupervised neural network for PD pattern recognition and classification”. Expert Systems with Applications, Volume 31, Issue 2 , August 2006, Pages 345–350.

Kay-Soon Low, Vinitha Krishnan, Hualiang Zhuang, Wei-Yun Yau, (2006) “On-Chip Genetic Algorithm Optimized Pulse Based RBF Neural Network for Unsupervised Clustering Problem”. Lecture Notes in Computer Science, vol. 4222, page(s):851–860.

Kharbat, F.; Bull, L.; Odeh, M., (2005) “Revisiting genetic selection in the XCS learning classifier system”. The 2005 IEEE Congress on Evolutionary Computation. Volume 3, 2–5 Sept. 2005 Page(s):2061 – 2068.

Kusumoputro, B.; Irwanto, P.; Jatmiko, W., (2002) “Optimization of fuzzy-neural structure through genetic algorithms and its application in artificial odor recognition-system”. Asia-Pacific Conference on Circuits and Systems, 28–31 Oct. 2002. vol.2, Page(s):47 – 51.

Lazaro, J.; Arias, J.; Martin, J.L.; de Alegria, I.M.; Andreu, J.; Jimenez, J., (2004) “An implementation of a general regression network on FPGA with direct Matlab link”. IEEE International Conference on Industrial Technology, Volume 3, 8–10 Dec. 2004 Page(s):1150 – 1155.

Leonardis, A.; Bischof, H., (1996) “Complexity optimization of adaptive RBF networks”. Proceedings of the 13th International Conference on Pattern Recognition Volume 4, 25–29 Aug. 1996, Page(s): 654 – 658.

Lera, G.; Pinzolas, M.,(1998) “A quasi-local Levenberg-Marquardt algorithm for neural network training”. IEEE World Congress on Computational Intelligence. Vol.3, 4–9 May 1998 Page(s):2242 – 2246.

Lian-Wei Zhao; Si-Wei Luo; Ling-Zhi Liao, (2004) “3D object recognition and pose estimation using kernel PCA”. Proceedings of 2004 International Conference on Machine Learning and Cybernetics, Volume 5, 26–29 Aug. 2004 Page(s):3258 – 3262.

M. Egmont-Petersen, D. de Ridder, H. Handels., (2002) “Image processing with neural networks - a review,” The Journal of the Pattern Recognition Society, Vol. 35, No.10, pp.

2279- 2301.

Maeda, Y.; Wakamura, M., (2005) “Simultaneous perturbation learning rule for recurrent neural networks and its FPGA implementation”. IEEE Transactions on Neural Networks, Volume 16, Issue 6, Nov. 2005 Page(s):1664 – 1672.

Mahalingam, V.; Ranganathan, N., (2006) “An efficient and accurate logarithmic multiplier based on operand decomposition”. Proceedings of the 19th International Conference on VLSI Design. 3-7 Jan. 2006.

Masters, T.; Land, W., (1997) “A new training algorithm for the general regression neural network”. IEEE International Conference on Systems, Man, and Cybernetics, Volume 3, 12–15 Oct. 1997, Page(s):1990 – 1994.

Mclaren, D.J., (2003) “Improved Mitchell-based logarithmic multiplier for low-power DSP applications”. IEEE International [Systems-on-Chip] SOC Conference, 2003. 17–20 Sept. 2003 Page(s):53 – 56.

Michalewicz, Z., (1994), “Genetic Algorithms + Data Structure = Evolution Programs”, AI Series, Springer-Verlag, New York.

Minchin, G.; Zaknich, A., (1999) “A design for FPGA implementation of the probabilistic neural network”. Proceedings of 6th International Conference on Neural Information Processing, Volume 2, 16–20 Nov. 1999, Page(s):556 – 559.

ModelSim Advanced Verification and Debugging SE Tutorial, (2005). Version 6.0.c. January 21, 2005. Mentor Graphics Corporation 8005 S.W. Boeckman Road, Wilsonville, Oregon.

Nebauer, C., (1998) “Evaluation of convolutional neural networks for visual recognition”. IEEE Transactions on Neural Networks, Vol. 9, Issue 4. pp. 685 – 696.

Nguyen, D.; Halupka, D.; Aarabi, P.; Sheikholeslami, A.,(2006) “Real-time face detection and lip feature extraction using field-programmable gate arrays”. IEEE Transactions on Systems, Man and Cybernetics, Part B, Vol. 36, Issue 4, Aug. 2006 Page(s):902 – 912.

Okamoto, J., Jr.; Milanova, M.; Bueker, U., (1998) “Active perception system for recognition of 3D objects in image sequences”. 5th International Workshop on Advanced Motion Control, 29 June–1 July 1998 Page(s):700 – 705.

Ozyilmaz, L.; Yildirim, T., (2002) “Diagnosis of thyroid disease using artificial neural network methods”. Proceedings of the 9th International Conference on Neural Information Processing, Volume 4, 18–22 Nov. 2002 Page(s):2033 – 2036.

P.-Y. Chen, H.-M. Tsai, C.-J. Lin and C.-Y. Lee , (2005) “FPGA Realization of a Radial Basis Function Based Nonlinear Channel Equalizer,” Lecture Notes in Computer Science, Springer-Verlag, May, 2005, Vol. 3498, pp. 320–325.

Pakhira, M.K. De, R.K., (2004) “Function optimization using a pipelined genetic algorithm”. Proceedings of the Intelligent Sensors, Sensor Networks and Information Processing Conference, 2004. page(s): 253- 257.

Paredes R., Vidal E.; (2000) “A class-dependent weighted dissimilarity measure for nearest-neighbor classification problems”. Pattern Recognition Letters, 21(12):1027- 1036.

Park, J.W.; Harley, R.G.; Venayagamoorthy, G.K., (2002) “Comparison of MLP and RBF neural networks using deviation signals for on-line identification of a synchronous generator”.

Power Engineering Society Winter Meeting, IEEE. Vol.1, 27–31 Jan. 2002, Page(s):274 – 279.

Parzen E.,(1962) “On Estimation of a Probability Density Function and Mode”, Annals of Mathematical Statistics, Vol. 36, 1065–1076, 1962.

Polat, Ö. ve Yıldırım, T., (kabul edildi, basımda-a) “Genetic Optimization of GRNN for Pattern Recognition without Feature Extraction”, Elsevier Expert Systems with Applications.

Polat, Ö. ve Yıldırım, T., (2006c) “Genelleştirilmiş Regresyon Sinir Ağının Genetik Algoritma ile Optimizasyonu”, Akıllı Sistemlerde Yenilikler ve Uygulamaları Sempozyumu 2006 (ASYU INISTA 2006) bildiriler kitabı, sf. 224-226 , Yıldız Teknik Üniversitesi, İstanbul, 31 Mayıs-2 Haziran 2006.

Polat, Ö. ve Yıldırım, T., (kabul edildi, basımda-b) “Hand geometry identification without feature extraction by general regression neural network”, Elsevier Expert Systems with Applications.

Polat, Ö., Yıldırım, T., (2006a) “Pattern Recognition Without Feature Extraction Using Probabilistic Neural Network”. Lecture Notes in Control and Information Sciences, Volume 345, pp.402–409, Springer-Verlag.

Polat, Ö., Yıldırım, T., (2006b) “GRNN ile 3-boyutlu benzer nesnelerin tanınması”, IEEE 14. Sinyal İşleme ve İletişim Uygulamaları Kurultay kitabı, 17-19 Nisan 2006, Antalya, Türkiye.

Polat, Ö., Yıldırım, T., (2007) “Recognition of Patterns without Feature Extraction by GRNN”. Lecture Notes in Computer Sciences, Volume 4432, pp.161–168, Springer-Verlag.

QingNian Zhang; XiangYang He; JianQi Liu, (2003) “RBF network based on genetic algorithm optimization for nonlinear time series prediction”. ISCAS '03. Proceedings of the 2003 International Symposium on Circuits and Systems, Volume 5, 25-28 May 2003 Page(s):V-693 - V-696.

S. Oberman and M. Flynn, (1997) “Division algorithms and implementations,” IEEE Transactions on Computers, vol. 46, no. 8, pp. 833–854.

Safarpour, S.; Veneris, A.; Baeckler, G.; Yuan, R. ,(2006) “Efficient SAT-based Boolean matching for FPGA technology mapping”. 43rd ACM/IEEE Design Automation Conference, 24–28 July 2006 Page(s):466 – 471.

Salvini, R.L.; V. de Carvalho, L.A., (2000) “Elastic neural net algorithm for cluster analysis”. Proceeding of Sixth Brazilian Symposium on Neural Networks, 22–25 Nov. 2000 Page(s):191 – 195.

Sameer A. Nene, Shree K. Nayar, Hiroshi Murase, (1996) “Columbia Object Image Library (COIL–100)”. Technical Report No. CUCS–006–96, Department of Computer Science Columbia University New York, N.Y. 10027.

Sanchez-Reillo, R.; Gonzalez-Marcos, A., (2000a) “Access control system with hand geometry verification and smart cards”. Aerospace and Electronic Systems Magazine, IEEE, Volume 15, Issue 2, Feb. 2000 Page(s):45 – 48.

Sanchez-Reillo, R.; Sanchez-Avila, C.; Gonzalez-Marcos, A., (2000b) “Biometric identification through hand geometry measurements”. IEEE Transactions on Pattern Analysis and Machine Intelligence Vol. 22, Issue 10, Page(s):1168 – 1171.

- Savich, A. W.; Moussa, M.; Areibi, S., (2007) "The Impact of Arithmetic Representation on Implementing MLP-BP on FPGAs: A Study". IEEE Transactions on Neural Networks, vol.18, no.1, Page(s): 240–252.
- Shyh-Jye Jon; Hui-Hsuan Wang, (2000) "Fixed-width multiplier for DSP application". The Proceedings of the 2000 IEEE International Conference on Computer Design: VLSI in Computers & Processors. 17–20 Sept. 2000 Page(s):318 – 322.
- Simard, P.Y., Steinkraus, D., Platt, J.C., (2003) "Best practices for convolutional neural networks applied to visual document analysis". Proc. of the 2003 Seventh International Conference on Document Analysis and Recognition, 3–6 Aug. 2003, 958 – 963.
- Skahill, K., (1996) "VHDL for Programmable Logic". Addison Wesley, Reading, Massachusetts.
- Tomandl D., Schober A., (2001) "A Modified General Regression Neural Network (MGRNN) with new, efficient training algorithms as a robust 'black box'-tool for data analysis". Neural Networks, Volume 14, Issue 8 , October 2001, Pages 1023–1034.
- Verleysen M., Hlavackova K.,(1994) "An optimized RBF network for approximation of functions", European Symposium on Artificial Neural Networks, Bruxelles (Belgique), pp. 175–180.
- Wen-Bo Zhao, De-Shuang Huang, Ji-Yan Du, Li-Ming Wang, (2004) "Genetic Optimization of Radial Basis Probabilistic Neural Networks". IJPRAI 18(8): 1473–1499.
- Yang F.; Paindavoine, M., (2003) "Implementation of an RBF neural network on embedded systems: real-time face tracking and identity verification". IEEE Transactions on Neural Networks, Volume 14, Issue 5, Sept. 2003 Page(s):1162 – 1175.
- Yazıcı G., (2006), "Genetik algoritmalar ile Radyal Temelli Fonksiyon Ağlarının Optimizasyonu". Yıldız Teknik Üniversitesi, Fen Bilimleri Enstitüsü, Yüksek lisans tezi. İstanbul, Türkiye.
- Yazıcı, G., Polat Ö., Yıldırım, T., (2006) "Genetic Optimizations for Radial Basis Function and General Regression Neural Networks", Lecture Notes in Artificial Intelligence, Volume 4293, pp.348-356, Springer-Verlag, November 2006.
- Yingjie Xing; Zhuqing Wang; Jing Sun; Wanlei Wang, (2006) "An Improved Genetic Algorithm with Recurrent Search for the Job-Shop Scheduling Problem". The Sixth World Congress on Intelligent Control and Automation, Volume 1, 21–23 June 2006 Page(s):3386 – 3390.
- YingLiang Ma; Pollick, F.; Hewitt, W.T., (2004) "Using B-spline curves for hand recognition". ICPR 2004, Proceedings of the 17th International Conference on Pattern Recognition, Volume 3, 23–26 Aug. 2004 Page(s):274 – 277.
- Youssif, R.S.; Purdy, C.N., (2003) "Comparing hybrid versus single strategy intelligent systems in signal pattern classification". MWSCAS '03, Proceedings of the 46th IEEE International Midwest Symposium on Circuits and Systems. Vol. 2, Page(s): 912- 915.
- Yuanping Zou; Zhengkun Mi; Minghai Xu, (2006) "Dynamic Load Balancing Based on Roulette Wheel Selection". International Conference on Communications, Circuits and Systems Proceedings, Volume 3, June 2006 Page(s):1732 – 1734.
- Zhanwei Li; Jizhou Sun; Jiawan Zhang; Zzunce Wei, (2003) "Computing models based on

GRNNs". IEEE Canadian Conference on Electrical and Computer Engineering, Volume 3, 4–7 May 2003, Page(s):1853 – 1856.

Znamirowski, L., Palusinski, O.A., Vrudhula,S.B.K., (2004) "Programmable Analog/Digital Arrays in Control and Simulation". Analog Integrated Circuits and Signal Processing, Vol. 39, pp. 55–73.

Zurada, J.M. (1992), "Introduction to Artificial Neural Systems". Boston, PWS Publishing Company.

İNTERNET KAYNAKLARI

- [1] <http://www.bilesim.com.tr/tr/>
- [2] <http://www.elektrik.gen.tr/>
- [3] <http://www.gmvhdl.com/VHDL.html>
- [4] <http://www.yapay-zeka.org>
- [5] <http://www.gpds.ulpgc.es/download>
- [6] <http://www.ics.uci.edu/~mlearn/MLSummary.html>
- [7] <http://www.bromba.com/faq/biofaq.htm>
- [8] http://www.sensormatic.com.tr/bio_sis.htm
- [9] <http://research.cs.tamu.edu/prism/lectures.htm>
- [10] <http://www.teknohaber.net>
- [11] <http://www1.cs.columbia.edu/CAVE/>
- [12] <http://www.xilinx.com>
- [13] <http://www.mathworks.com/>

ÖZGEÇMİŞ

Doğum tarihi 25.09.1977

Doğum yeri Gaziantep

Lise 1991–1994 Gaziantep-Bahaattin Bayraktar Lisesi

Lisans 1995–1999 Süleyman Demirel Üniversitesi Mühendislik Mimarlık Fak. Elektronik ve Haberleşme Mühendisliği Bölümü

Yüksek Lisans 2001–2003 Süleyman Demirel Üniversitesi Fen Bilimleri Enstitüsü Elektronik ve Haberleşme Müh. Anabilim Dalı

Doktora 2003–2007 Yıldız Teknik Üniversitesi Fen Bilimleri Enstitüsü Elektronik ve Haberleşme Müh. Anabilim Dalı, Elektronik programı

Çalıştığı kurum(lar)

2000–2003 Süleyman Demirel Üniversitesi Mühendislik Mimarlık Fak. Elektronik ve Haberleşme Mühendisliği Bölümü Araştırma Görevlisi

2003-Devam ediyor Yıldız Teknik Üniversitesi Elektronik ve Haberleşme Mühendisliği Bölümü Araştırma Görevlisi (2547 Sayılı Kanun'un 35. maddesi uyarınca doktora eğitimi için görevlendirildi)