



T.C.
NİĞDE ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ
ELEKTRİK ELEKTRONİK MÜHENDİSLİĞİ ANABİLİM DALI

ALAN PROGRAMLANABİLİR KAPI DİZİLERİ (FPGA) ÜZERİNDE
MANYETİK REZONANS GÖRÜNTÜLERİN BÖLÜTLENMESİ

SALİM ÇINAR

Temmuz 2011

T.C.
NİĞDE ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ
ELEKTRİK ELEKTRONİK MÜHENDİSLİĞİ ANABİLİM DALI

ALAN PROGRAMLANABİLİR KAPI DİZİLERİ (FPGA) ÜZERİNDE
MANYETİK REZONANS GÖRÜNTÜLERİN BÖLÜTLENMESİ

SALİM ÇINAR

Yüksek Lisans Tezi

Danışman

Yrd. Doç. Dr. Mehmet Nadir KURNAZ

Temmuz 2011

Salim ÇINAR tarafından **Yrd. Doç. Dr. Mehmet Nadir KURNAZ** danışmanlığında hazırlanan “**Alan Programlanabilir Kapı Dizileri (FPGA) Üzerinde Manyetik Rezonans Görüntülerin Bölütlenmesi**” adlı bu çalışma jürimiz tarafından Niğde Üniversitesi Fen Bilimleri Enstitüsü **Elektrik Elektronik Mühendisliği** Anabilim Dalında Yüksek Lisans tezi olarak kabul edilmiştir.

Başkan : Yrd. Doç. Dr. Mahmut TOKMAKÇI (Erciyes Üniversitesi)

Üye : Yrd. Doç. Dr. Fuat KARAKAYA (Niğde Üniversitesi)

Üye : Yrd. Doç. Dr. Mehmet Nadir KURNAZ (Niğde Üniversitesi)

ONAY:

Bu tez, Fen Bilimleri Enstitüsü Yönetim Kurulunca belirlenmiş olan yukarıdaki jüri üyeleri tarafından/....../20.... tarihinde uygun görülmüş ve Enstitü Yönetim Kurulu’nun/....../20.... tarih ve sayılı kararıyla kabul edilmiştir.

...../...../20...

Doç. Dr. Nurettin ACIR

MÜDÜR

ÖZET

ALAN PROGRAMLANABİLİR KAPI DİZİLERİ (FPGA) ÜZERİNDE MANYETİK REZONANS GÖRÜNTÜLERİN BÖLÜTLENMESİ

ÇINAR, Salim

Niğde Üniversitesi

Fen Bilimleri Enstitüsü

Elektrik-Elektronik Mühendisliği Anabilim Dalı

Danışman : Yrd. Doç. Dr. Mehmet Nadir KURNAZ

Temmuz 2011, 50 sayfa

Görüntü bölütleme, medikal görüntü işleme uygulamalarında sıkça kullanılan işlemlerden biridir. Manyetik Rezonans (MR) görüntüler, tanı koyma amacıyla görüntüdeki ayırt edici dokuların ortaya çıkarılması için bölütlenmektedir. MR görüntülerin yüksek çözünürlüklü olması ve matematiksel yöntemlerdeki hesap yükünün fazla olması nedeniyle, MR görüntü bölütleme işlemi aşırı hesaplama karmaşıklığına sahiptir. Son yıllarda paralel işlem yapabilme yeteneğinden dolayı alan programlanabilir kapı dizileri (FPGA) yüksek hesaplama zamanı gerektiren çalışmalarda geniş uygulama alanı bulmuştur. Tez çalışmasında, MR görüntülerdeki dokusal özniteliklerin çıkarılması için komşu benek yoğunluğu temelli öznitelik çıkarma yöntemi, bölütleme işlemi için k -NN (k -en yakın komşu) sınıflayıcı ve GAL (Büyük ve Öğren) ağı önerilmiştir. Çalışmanın ilk aşamasında, MR görüntülerin bölütlenmesi amacıyla önerilen yöntemlerin MATLAB ve Microsoft Visual C# programları kullanılarak bilgisayarla benzetimi gerçekleştirilmiştir. İkinci aşamada, aynı yöntemler FPGA donanım platformu üzerinde VHDL kodlama ile gerçekleştirilmiştir. Hem bilgisayarla benzetim hem donanımla gerçekleştirilen bölütleme sonuçları karşılaştırmalı olarak incelenmiştir.

Anahtar sözcükler: MR görüntü, Bölütleme, k -NN sınıflayıcı, GAL ağı, FPGA.

SUMMARY

SEGMENTATION OF MAGNETIC RESONANCE IMAGES ON FIELD PROGRAMMABLE GATE ARRAYS (FPGAs)

ÇINAR, Salim

Nigde University

Graduate School of Natural and Applied Science

Department of Electrical-Electronics Engineering

Supervisor : Assistant Professor Dr. Mehmet Nadir KURNAZ

July 2011, 50 pages

Image segmentation is one of the commonly used procedures in the medical image processing applications. Magnetic resonance (MR) images are segmented for extracting distinguishing tissues in the image on the purpose of diagnosis. Due to the high resolution characteristics of the MR images and a large amount of computational load in mathematical methods, MR image segmentation process has an excessive computational complexity. Recently, field programmable gate array (FPGA) implementation capable of performing many complex computations in parallel has been applied in many areas needed for high computation. In this thesis, neighbor-pixel-intensity based feature extraction method for extraction of the textural features in medical images, k-NN (k-Nearest Neighbor) classifier and GAL (Grow and Learn) network for segmentation process are proposed. In the first stage of the study, computer simulation of the proposed methods for segmentation of MR images is separately implemented by using MATLAB and Visual C# programming. In the second stage, the same methods are implemented by using VHDL coding on FPGA hardware platform. Segmentation results implemented with both computer simulation and hardware are comparatively examined.

Keywords: Medical image, Segmentation, k-NN classifier, GAL network, FPGA.

TEŞEKKÜR

Bu tezin hazırlanması sürecinde beni yönlendiren, teşvik eden ve gerek teknik bilgi gerekse de doküman konusunda desteğini esirgemeyen değerli danışmanım Sayın Yrd. Doç. Dr. Mehmet Nadir KURNAZ'a; yardımlarını benden esirgemeyen Yrd. Doç. Dr. Fuat KARAKAYA'ya; yüksek lisans eğitimim boyunca beraber çalıştığımız Mehmet Muzaffer KÖSTEN ve diğer çalışma arkadaşlarıma; hayatım boyunca maddi manevi her anlamda desteğini benden esirgemeyen canım aileme çok teşekkür ederim.

Ayrıca bu tez çalışmasının, Niğde Üniversitesi Bilimsel Araştırma Projeleri Birimi tarafından desteklenen FEB 2010/24 nolu proje kapsamında gerçekleştirilmesi nedeniyle NİĞDE Üniversitesine teşekkürlerimi sunarım.

İÇİNDEKİLER

ÖZET	iii
SUMMARY	iv
TEŞEKKÜR.....	v
İÇİNDEKİLER DİZİNİ	vi
ÇİZELGELER DİZİNİ	vii
ŞEKİLLER DİZİNİ	viii
KISALTMALAR ve SİMGELER	x
BÖLÜM I. GİRİŞ	1
1.1 Çalışmanın Amacı ve Kapsamı	1
BÖLÜM II. PROGRAMLANABİLİR LOJİK ELEMANLAR	6
2.1 Programlanabilir Lojik Elemanların Gelişim Süreci	6
2.2 Alan Programlanabilir Kapı Dizileri (FPGAs; Field Programmable Gate Arrays)	9
2.3 Donanım Tanımlama Dili (VHDL; VHSIC Hardware Description Language)	12
BÖLÜM III. YÖNTEM	17
3.1 Öznitelik Çıkarma Yöntemi	17
3.2 k -En Yakın Komşu (k -NN; k -Nearest Neighbor) Sınıflayıcısı	19
3.2.1 k -NN sınıflayıcısının FPGA üzerinde gerçekleştirilmesi	21
3.3 Büyü ve Öğren (GAL; Grow and Learn) Ağı	26
3.3.1 Geri beslemeli lineer kaydırmalı kaydedici (LFSR)	27
3.3.2 GAL ağının FPGA üzerinde gerçekleştirilmesi	30
BÖLÜM IV. MR GÖRÜNTÜLERİNİN BİLGİSAYAR ve FPGA ÜZERİNDE BÖLÜTLENMESİ	33
4.1 k -NN Sınıflayıcısı Kullanılarak Elde Edilen Bölütleme Sonuçları	34
4.2 GAL Ağı Kullanılarak Elde Edilen Bölütleme Sonuçları	38
BÖLÜM V. SONUÇLAR ve TARTIŞMA	43
KAYNAKLAR	47

ÇİZELGELER DİZİNİ

Çizelge 3.1	Bileşenlerin işlem sonuçlarını elde etmek için kullandıkları saat darbesi sayısı	19
Çizelge 4.1	k -NN sınıflayıcısının bilgisayarla benzetim ve FPGA sonuçları	37
Çizelge 4.2	GAL ağının bilgisayarla benzetim ve FPGA sonuçları	41
Çizelge 4.3	k -NN sınıflayıcı ve GAL ağı kullanılarak gerçekleştirilen bölütleme sisteminin FPGA’da kullandığı alan	42
Çizelge 5.1	GAL ağı ve k -NN sınıflayıcısının karşılaştırılması.....	45

ŞEKİLLER DİZİNİ

Şekil 1.1	Medikal görüntülerin bilgisayar destekli bölütlenmesi	2
Şekil 2.1	PLA yapısı.....	7
Şekil 2.2	PAL yapısı.....	7
Şekil 2.3	CPLD blok diyagramı	8
Şekil 2.4	FPGA blok diyagramı	9
Şekil 2.5	VHDL temel kod yapısı	12
Şekil 2.6	VHDL kütüphanesinin temel yapısı	13
Şekil 3.1	a) İlgilenilen benek ve 1-komşuluğu b) Ortalama bulma işlemi blok diyagramı	18
Şekil 3.2	k -NN sınıflayıcı algoritmasının blok diyagramı	20
Şekil 3.3	FPGA’da gerçekleştirilen k -NN yapısının blok diyagramı	22
Şekil 3.4	İşlem elemanının FPGA’da gerçekleştirilmesi	23
Şekil 3.5	Bilgisayar arayüzü	25
Şekil 3.6	CRT monitörde gösterilen bölütleme sonucu.....	25
Şekil 3.7	GAL ağının yapısı	26
Şekil 3.8	10 bitlik LFSR yapısı	28
Şekil 3.9	FPGA’da gerçekleştirilen GAL ağının blok diyagramı.....	30
Şekil 4.1	Orijinal MR-1 görüntüsü	34
Şekil 4.2	k -NN sınıflayıcısı kullanılarak; a) MATLAB platformu üzerinde bölütlenmiş MR-1 görüntüsü b) Microsoft Visual C# platformu üzerinde bölütlenmiş MR-1 görüntüsü c) FPGA üzerinde platformu üzerinde bölütlenmiş MR-1 görüntüsü.....	34
Şekil 4.3	Orijinal MR-2 görüntüsü	35
Şekil 4.4	k -NN sınıflayıcısı kullanılarak; a) MATLAB platformu üzerinde bölütlenmiş MR-2 görüntüsü b) Microsoft Visual C# platformu üzerinde bölütlenmiş MR-2 görüntüsü c) FPGA üzerinde platformu üzerinde bölütlenmiş MR-2 görüntüsü.....	35
Şekil 4.5	Orijinal MR-3 görüntüsü	36
Şekil 4.6	k -NN sınıflayıcısı kullanılarak; a) MATLAB platformu üzerinde bölütlenmiş MR-3 görüntüsü b) Microsoft Visual C# platformu üzerinde bölütlenmiş MR-3 görüntüsü c) FPGA üzerinde platformu üzerinde bölütlenmiş MR-3 görüntüsü.....	36

Şekil 4.7 Orijinal MR-1 görüntüsü	38
Şekil 4.8 GAL ağı kullanılarak;	
a) MATLAB platformu üzerinde bölütlenmiş MR-1 görüntüsü	
b) Microsoft Visual C# platformu üzerinde bölütlenmiş MR-1 görüntüsü	
c) FPGA üzerinde platformu üzerinde bölütlenmiş MR-1 görüntüsü.....	38
Şekil 4.9 Orijinal MR-2 görüntüsü	39
Şekil 4.10 GAL ağı kullanılarak;	
a) MATLAB platformu üzerinde bölütlenmiş MR-2 görüntüsü	
b) Microsoft Visual C# platformu üzerinde bölütlenmiş MR-2 görüntüsü	
c) FPGA üzerinde platformu üzerinde bölütlenmiş MR-2 görüntüsü.....	39
Şekil 4.11 Orijinal MR-3 görüntüsü	40
Şekil 4.12 GAL ağı kullanılarak;	
a) MATLAB platformu üzerinde bölütlenmiş MR-3 görüntüsü	
b) Microsoft Visual C# platformu üzerinde bölütlenmiş MR-3 görüntüsü	
c) FPGA üzerinde platformu üzerinde bölütlenmiş MR-3 görüntüsü.....	40

KISALTMALAR ve SİMGELER

KISALTMA/SİMGE

k -NN	k -En yakın komşuluk
GAL	Büyü ve öğren
FPGA	Alan programlanabilir kapı dizileri
IEEE	Institute of Electrical and Electronics Engineers
CPU	Merkezi işlem birimi
SIMD	Tek komut çoklu veri
PROM	Programlanabilir salt okunur bellek
PLA	Programlanabilir mantık dizileri
PAL	Programlanabilir dizi mantığı
SPLD	Basit programlanabilir lojik elemanlar
CPLD	Karmaşık programlanabilir lojik elemanlar
MPGA	Maskeli programlanabilir kapı dizileri
CLB	Yapılandırılabilir mantık blokları
ASIC	Uygulamaya özgü tümleşik devre
RAM	Rastgele erişimli bellek
EPROM	Silinebilir programlanabilir salt okunur bellek
EEPROM	Elektriksel etkiyle silinebilen programlanabilir salt okunur bellek
SRAM	Durağan rastgele erişimli bellek
SSRAM	Eşzamanlı durağan rastgele erişimli bellek
CMOS	Tümleyici metal oksit yarıiletken
VHSIC	Çok yüksek hızlı tümleşik devreler
VHDL	VHSIC donanım tanımlama dili
BT	Bilgisayarlı tomografi
ÖV	Öznitelik vektörü
ÖV1	Öznitelik vektörünün birinci elemanı
ÖV2	Öznitelik vektörünün ikinci elemanı
$d(x_i, x_l)$	Öklid mesafesi
x_i	Sınıfı bilinmeyen giriş beneğine ait öznitelik vektörü
x_l	Sınıfa ait öznitelik vektörü
p	Beneğe ait öznitelik sayısı

k	Komşuluk değeri
VGA	Video grafik dizisi
UART	Evrensel eşzamansız alıcı verici
İE	İşlem elemanı
LFSR	Geri beslemeli lineer kaydırmalı kaydedici
LSB	En önemsiz bit
MSB	En önemli bit

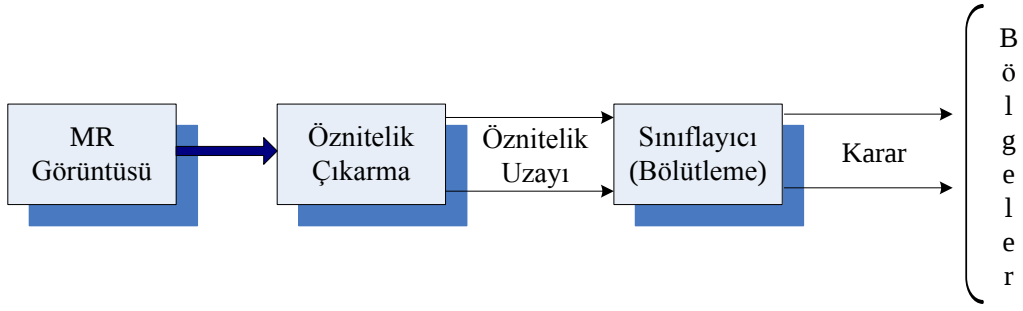
BÖLÜM I

GİRİŞ

1.1 Çalışmanın Amacı ve Kapsamı

Medikal görüntüler hastalıkların teşhisinde oldukça önemlidir. Röntgen, bilgisayarlı tomografi (BT), manyetik rezonans (MR) ve ultrason görüntüleme bu amaçla kullanılan görüntüleme yöntemlerinin başında gelmektedir. Bunlardan MR görüntüleme en çok çalışma ve araştırmanın yapıldığı görüntüleme yöntemidir [1]. MR görüntüleme tekniği ile vücudun değişik bölgelerinde oldukça yüksek çözünürlükte görüntüler elde edilebilmektedir. Özellikle yumuşak doku kontrastının çok yüksek olması nedeniyle, beyin gibi organlardaki patolojik dokular (tümör vs.) kolaylıkla saptanabilmektedir. Uzmanlar diğer yöntemlerle belirleyemedikleri hastalıkların neredeyse kesin tanısını bu görüntüleme yöntemi ile elde edilmiş görüntüleri inceleyerek koyarlar veya hastalık hakkındaki nihai kararlarını bu incelemeden sonra verirler.

Gelişen teknoloji ve yazılımlar ile birlikte görüntülerin analizi bilgisayar desteği ile otomatik ya da yarı-otomatik yapılabilmektedir [2, 3]. Günümüzde bilgisayar destekli bölütleme sistemleri doktora tanı koymada yardımcı olmak üzere geliştirilmiş sistemlerdir. Bu sistemler görüntüdeki farklı dokuların ayırt edilmesine olanak sağlayarak görüntülerin incelenmesinde doktora kolaylık sağlarlar. Bilgisayar destekli bölütleme sistemleri en genel haliyle iki aşamada gerçekleştirilmektedir; 1) öznitelik çıkarma, 2) bölütleme. Öznitelik çıkarma işlemi, görüntünün sahip olduğu farklı dokusal özelliklerin belirlenmesi için yapılır. Bölütleme ise görüntüde var olan ayırt edici dokuların ortaya çıkarılması için yapılan işlemidir. Şekil 1.1'de bilgisayar destekli bölütlemenin genel yapısı görülmektedir.



Şekil 1.1 Medikal görüntülerin bilgisayar destekli bölütlenmesi

Bölütleme işlemi için görüntüdeki dokulara ait özniteliklerin çıkarılması gerekmektedir. Öznitelik çıkarma işlemi bölütleme başarımını önemli ölçüde etkilemektedir. Literatürde dokusal özniteliklerin çıkarılması için birçok yöntem önerilmiştir. Bunlar; 1) istatistiksel, 2) model temelli (yapısal), 3) filtreleme temelli (spektral) yöntemlerdir [4, 5]. İstatistiksel yöntemler bir benneğin, çevresindeki diğer beneklerle olan istatistiksel ilişkilerini temel alır [6-8]. Model temelli (yapısal) yöntemler, dokuyu birkaç parametre ile tanımlayabilecek stokastik (rastgele) bir işlemin gerçekleştirilmesi esasına dayanır [9, 10]. Filtreleme temelli (spektral) yöntemler ise görüntüdeki dokuların farklı frekans ve yön bileşenlerine ayrıştırılarak incelenmesi ve çözülmesi biçiminde gerçekleştirilir [11-13].

Öznitelik çıkarma işlemi ile belirlenen alt kümedeki örnekler öznitelik uzayında farklı şekilde dağılım gösterirler. Sınıflayıcıların görevi, bu dağılımı en iyi temsil edecek şekilde sınıf sınırlarını oluşturmaktır. Sınıflayıcı yapısını karar fonksiyonları belirlemektedir. Yapı, karar fonksiyonlarının fiziksel olarak gerçekleşmesinden ibarettir. Sınıflayıcı yapısını tanımlayan karar fonksiyonlarının parametreleri, alt kümedeki örnekler (öznitelikler) kullanılarak belirlenmektedir. Literatürde çeşitli karar mekanizmalarına sahip sınıflayıcıların görüntü bölütleme amacıyla sıkça kullanıldığı gözlenmektedir [13-15].

Öznitelik çıkarma ve bölütleme işlemlerinde kullanılan algoritmalar genellikle karmaşık ve hesap yükü fazla olan işlemlerdir. Ayrıca, görüntülerin yüksek boyutlu ve yüksek çözünürlüklü oldukları ve çoğu bilgisayar yazılımlarının seri olarak işlem yaptıkları göz önüne alındığında, seri programlamayı kullanan bilgisayar yazılımlarının gerçek zamanlı uygulamalarda yeterince hızlı sonuçlar üretemeyecekleri kabul edilir. Öte yandan, bu yazılımlar gerçekleştirdikleri algoritmaya bağlı olarak, bulundukları

bilgisayarın işlemci ve bellek kapasitelerini aşırı derecede kullanabilmektedirler. Bu durum, bilgisayarın performansının düşmesine, arka planda gerçekleşen diğer işlemlerin etkilenmesine ve işlemcisinin oldukça ısınmasına neden olabilir.

Son yıllarda görüntü tanıma ve görüntü işleme uygulamalarında da kullanım olanağı bulan özel yongalar ve yeniden yapılandırılabilir elemanların kullanıldığı çeşitli platformlar geliştirilmektedir. Bu platformlar üzerinde gerçekleştirilen sistemler bilgisayar işlemcisine göre çok daha kısa sürede işlem yapabilmektedirler. Alan programlanabilir kapı dizileri (FPGAs; Field programmable gate arrays), herhangi bir sayısal fonksiyonu mantık kapıları ile gerçekleştirebilmek için kullanıcı tarafından programlanabilen tümleşik devrelerdir. FPGA'lar paralel işlem yeteneğine sahip, yeniden düzenlenerek hızlı ve esnek tasarım yapabilme olanağı tanıyan donanım platformlarıdır. Günümüzde FPGA'lara olan ilgi her geçen gün artmaktadır. Literatürde çeşitli görüntü işleme uygulamalarının FPGA donanım platformu üzerinde gerçekleştirildiği gözlenmektedir [16, 18-22].

MR görüntülerin yarı-otomatik veya tam otomatik bölütlendiği bilgisayarla benzetim sistemlerindeki hesap yoğunluğu ve MR görüntülerin yüksek çözünürlüklü olması, yüksek hızlı hesaplama yeteneğine sahip sistemlerin tasarlanmalarını gerektirir. Ayrıca, bilgisayarla benzetim uygulamalarında işlemler sıralı bir şekilde (seri programlama) yapıldığından, öznetelik çıkarma ve bölütleme aşamaları yüksek hesaplama zamanlarına sahip olmaktadır. Bu nedenle, hesap yükü fazla olan bu işlemlerin hesaplama zamanlarını düşürmek için paralel işlem yapabilme (paralel programlama) yeteneğine sahip FPGA'ları kullanmak MR görüntülerin bölütlenmesi için oldukça uygun bir tercih olacaktır.

Tahir ve diğerleri [16], prostat ve göğüs kanserine ait görüntülerin sınıflandırılması amacıyla k -NN sınıflayıcı kullanılarak yaptıkları çalışmada, FPGA'nın bilgisayarla gerçeklemeye göre aynı sınıflama başarımından ortalama 33 kat daha hızlı sonuç ürettiğini gözlemişlerdir. Çalışmada, k -NN algoritmasında mesafe ölçütü olarak kullanılan Öklid mesafesinin matematiksel ifadesi basitleştirilmiş ve işlem yükü azaltılmıştır.

Özdemir ve diğerleri [17], 8x2x1 katmanlı ileri beslemeli yapay sinir ağını FPGA üzerinde donanımsal olarak gerçekleştirmiş ve EKG işaretlerini sınıflandırmışlardır. IEEE 754 32 bit kayan noktalı sayı formatının kullanılması sayesinde yüksek doğrulukta sonuçlar elde edilmiştir. FPGA ile gerçekleştirilen sistemin sonuçları ile MATLAB kullanılarak gerçekleştirilen sistemin sonuçları incelendiğinde başarımların hemen hemen aynı olduğu gözlemlenmiştir.

Rakvic ve diğerleri [18], RED (Ridge Energy Direction) algoritmasının kullanıldığı bir iris tanıma uygulamasını, FPGA donanım platformu ve PlayStation3 oyun konsolunun mikroişlemcisi üzerinde ayrı ayrı gerçekleştirmişler ve FPGA sisteminin PlayStation3 işlemcisinden 2.5 kat daha hızlı olduğunu gözlemişlerdir.

Koo ve diğerleri [21] 3 boyutlu MR görüntüsünü sınıflandırmak için kısmi hacim kestirimi (PVE; Partial Volume Estimation) algoritmasını FPGA üzerinde ve Itanium-2 CPU üzerinde gerçekleştirmişlerdir. İki sistemin performansı karşılaştırıldığında FPGA'nın CPU'dan 5.1 kat daha hızlı işlem yaptığı sonucuna ulaşılmıştır. Doğruluk yönünden karşılaştırıldığında ise çok küçük farklılıkların olduğu tespit edilmiştir. Yapılan çalışmada PVE algoritmasını gerçekleştirmek için 4 adet FPGA platformu kullanılmıştır.

Cloutier ve diğerleri [22] görüntü işleme uygulamaları için sanal görüntü işlemcisi (VIP; Virtual Image Processor) geliştirmişlerdir. Bu yapıda 2x2 matris şeklinde yerleştirilmiş 4 adet FPGA ve çevresel bileşen bağlantı (PCI; Peripheral Component Interconnection) protokolü için 1 adet FPGA bulunmaktadır. Sistemde tek komut çoklu veri (SIMD; Single Instruction Multiple Data) yapısı mevcuttur. Bu yapıda FPGA'lar birbirlerine "2D Torus Bağlantı" topolojisiyle bağlanmışlardır. Yapılan çalışmada VIP, 2 adet uygulama ile diğer donanımsal görüntü işleme yapılarıyla karşılaştırılmıştır. Birinci uygulamada ikili (binary) görüntünün konvolüsyonu alınmıştır. VIP, NET32K bord [23] ve 90 MHz Pentium bilgisayarla karşılaştırılmıştır. Konvolüsyon işleminin sürelerine bakıldığında küçük şablon boyutu olan uygulamada VIP diğer iki sistemden oldukça hızlıdır. Şablon boyutu arttıkça VIP, NET32K borduna göre biraz yavaş kalmaktadır. Buna rağmen bilgisayara göre çok daha az işlem süresine sahiptir. İkinci uygulamada ise el yazısını tanıma hedeflenmiştir. El yazısı tanıma işlemi ileri yönlü yapay sinir ağı kullanılarak gerçekleştirilmiştir. Bu uygulama VIP, ANNA bord [23] ve

çift işlemcili SPARC10 [23] üzerinde gerçekleştirilmiştir. İşlem süreleri karşılaştırıldığında VIP'nin, ANNA bord'dan 1.36 kat ve SPARC10'dan 16 kat daha hızlı olduğu sonucuna ulaşılmıştır.

Liu ve diğerleri [24], Gazula ve Kabuka'nın önerdiği [25] yapay sinir ağı (YSA) tabanlı doğru veya yanlış değerleri alan k -en yakın komşu (BKNN-Boolean K-Nearest Neighbor) sınıflayıcısını FPGA'da gerçekleştirmişlerdir. BKNN ikili giriş ve çıkışlara, tam sayı ağırlık katsayılarına, hızlı öğrenme ve sınıflama süresine sahiptir. Ayrıca yakınsama oranı iyidir. Yapılan çalışma 64 adet ikili öznitelige sahip olan veri kümeleri üzerinde denenmiştir ve %95 oranında sınıflama başarımı gözlenmiştir.

Ölmez ve diğerleri [26], kalp seslerini sınıflamak amacıyla yaptıkları çalışmada, öznitelik vektörleri dalgacık dönüşümü kullanılarak çıkarılmıştır. Sınıflama işlemi için GAL ağı kullanılmış ve sınıflama sonuçları literatürdeki diğer sınıflayıcılarla karşılaştırılarak incelenmiştir.

Tez çalışmasında, MR görüntülerindeki dokusal özniteliklerin çıkarılması için komşu beneklerin gri seviye değerlerine dayanan öznitelik çıkarma yöntemi ve bölütleme işlemi için Fix ve Hodges [27] tarafından önerilen k -NN (k -en yakın komşu) sınıflayıcı ile Alpaydın [28] tarafından önerilen Büyü ve öğren (GAL; Grow and Learn) ağı kullanılmıştır. Çalışmanın ilk aşamasında, MR görüntülerin bölütlenmesi amacıyla önerilen yöntemlerin, MATLAB ve Microsoft Visual C# programları kullanılarak bilgisayarla benzetimi gerçekleştirilmiştir. İkinci aşamada, aynı yöntemler FPGA donanım platformu üzerinde VHDL kodlama ile gerçekleştirilmiştir. Hem bilgisayarla benzetim hem donanımla gerçekleştirilen bölütleme sonuçları karşılaştırmalı olarak incelenmiştir. Tez çalışmasında kullanılan öznitelik çıkarma ve bölütleme yöntemlerinin tamamı FPGA üzerinde donanımsal olarak gerçekleştirilmiştir. GAL ağı donanımsal olarak FPGA üzerinde *ilk kez* bu çalışmada gerçekleştirilmiştir.

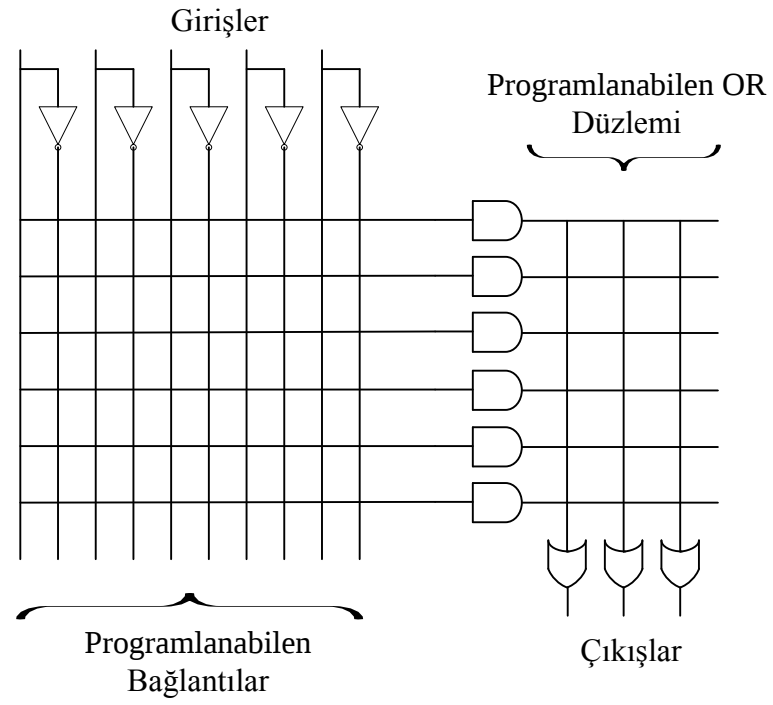
BÖLÜM II

PROGRAMLANABİLİR LOJİK ELEMANLAR

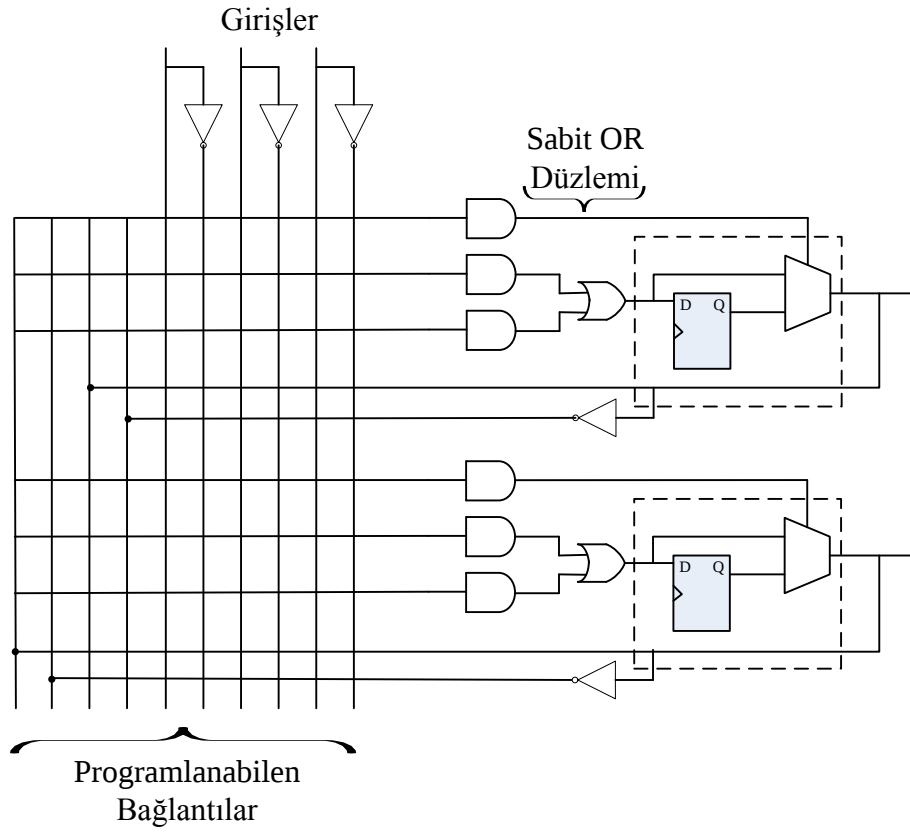
2.1 Programlanabilir Lojik Elemanların Gelişim Süreci

Programlanabilir lojik elemanlar (PLD; Programmable logic device) sayısal bir sistemi donanımsal olarak gerçekleştirmek için kullanılan tümleşik devrelerdir [29]. PLD'ler, farklı sayısal sistemlerin gerçekleştirilmesi için kullanıcı tarafından programlanabilir [29]. PLD'ler temel olarak üç bölüme ayrılır. Bunlar sırasıyla; basit programlanabilir lojik elemanlar (SPLD; Simple Programmable Logic Device), karmaşık programlanabilir lojik elemanlar (CPLD; Complex Programmable Logic Device) ve alan programlanabilir kapı dizileri (FPGA; Field Programmable Gate Array) olarak verilebilir.

Entegre devre üretimindeki gelişmeler ve bir yonga içinde bulunan transistör sayısının artması programlanabilir elemanlara olan ilgiyi artırmıştır. Programlanabilir elemanların gelişim sürecine bakıldığında ilk yapı olarak programlanabilir salt okunur bellek (PROM; Programmable Read Only Memory) verilebilir [30]. PROM'da adres girişleri ve bu adres girişlerine karşılık çıkış bilgisi mevcuttur. Bu yapısından dolayı lojik devre gerçeklenmesinde pek sık kullanılmamaktadır. Programlanabilir mantık dizileri (PLA; Programmable Logic Array) programlanabilir lojik devre gerçeklenmesinde geliştirilen ilk yapıdır. Şekil 2.1'de PLA yapısı görülmektedir [31]. PLA yapısında lojik fonksiyonlar çarpımların toplamı şeklinde gerçekleştirilir [30]. PLA üretiminin pahalı olması ve hız performansındaki zayıflık dezavantaj olarak verilebilir [30]. Bu dezavantajlar programlanabilir dizi mantığı (PAL; Programmable Array Logic) yapısının geliştirilmesi ile aşılmıştır. PAL yapısında PLA'dan farklı olarak tek seviyeli programlanabilirlik özelliği vardır. Programlanabilen AND bağlantılarının çıkışında sabit OR kapıları vardır. PAL yapısının çıkışında bulunan OR kapıları genellikle Flip-Flop'lara bağlıdır. Şekil 2.2'de PAL yapısı gösterilmiştir [31].



Şekil 2.1 PLA yapısı

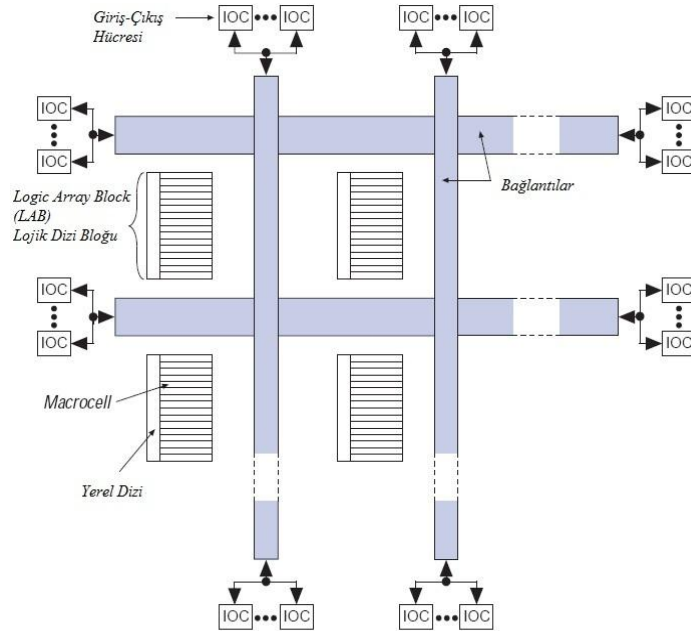


Şekil 2.2 PAL yapısı

PAL yapısı sayısal devre tasarımında derin bir etkiye sahiptir ve yeni gelişmeler için temel teşkil etmektedir [30]. PLA, PAL ve buna benzer yapılar SPLD kategorisinde yer almaktadır.

Gelişen teknoloji ile birlikte daha büyük kapasitede SPLD üretmek mümkündür. Yüksek kapasiteli üretimi olumsuz yönde etkileyen en önemli faktör giriş sayısının artmasına bağlı olarak programlanabilen alanın hızlı bir şekilde artmasıdır [30]. Bu problemin giderilmesi için CPLD yapısı geliştirilmiştir [30]. Bu yapıda birçok SPLD tek bir yapıda birleştirilmiştir. SPLD yapıları arasında programlanabilen bağlantılar mevcuttur. Şekil 2.3'de CPLD blok yapısı görülmektedir [32]. CPLD yapısı yaklaşık olarak 50 adet SPLD içermektedir [30]. SPLD sayısı artırılarak daha yüksek kapasiteli CPLD oluşturmak zordur.

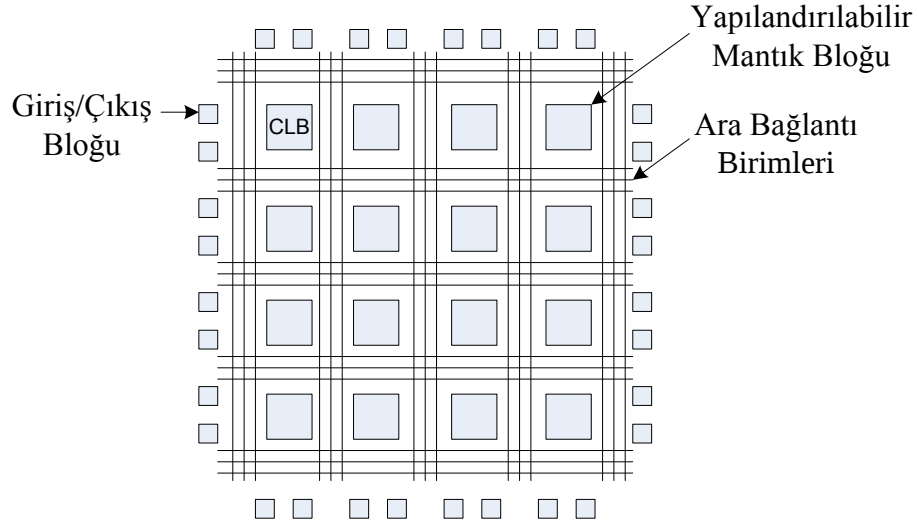
Maskeli programlanabilir kapı dizileri (MPGA; Mask Programmable Gate Array) yüksek kapasiteli genel amaçlı lojik entegre yapısıdır [30]. Bu yapıda önceden yerleştirilmiş transistörler mevcuttur. Bu transistörlerin bağlantıları kullanıcı tarafından gerçekleştirilir. MPGA'nın üretim süresinin uzunluğu ve yüksek maliyeti dezavantaj olarak verilebilir.



Şekil 2.3 CPLD blok diyagramı

2.2 Alan Programlanabilir Kapı Dizileri (FPGAs; Field Programmable Gate Arrays)

FPGA'lar; giriş/çıkış bloğu, matris şeklinde yerleşmiş yapılandırılabilir mantık bloklar (CLB) ve bu blokların birbirleriyle bağlantılarından oluşan programlanabilir bir sayısal tümleşik devredir [33]. FPGA blok diyagramı Şekil 2.4'de gösterilmiştir [30].



Şekil 2.4 FPGA blok diyagramı

FPGA'lar son kullanıcı tarafından elektriksel olarak programlanabilirler. Böylece uygulamaya yönelik üretilen, uygulamaya özel tümleşik devre (ASIC; Application Specific Integrated Circuit) entegrelerine göre daha esnek bir yapıya sahiptir. ASIC entegrelerin üretim süreci uzun ve pahalıdır. FPGA'lar ise kısa bir sürede programlanabilir ve tasarlanan sistemde hata olduğunda kolayca düzeltme yapılabilir. FPGA, ASIC yapılarına göre 20 ila 35 kat daha fazla alana ihtiyaç duyarlar. Ayrıca hız performansı yaklaşık 3 kat daha düşük ve yaklaşık 10 kat daha fazla güç tüketir. Bu dezavantajlara rağmen FPGA sayısal devre tasarımında önemli bir rol oynamaktadır [31].

FPGA temel olarak giriş/çıkış blokları, CLB ve bunların bağlantılarından oluşur [33]. Bugünkü teknolojiye FPGA'lar dış dünya ile iletişim için düzinelerce giriş/çıkış bağlantılarına (pin) sahiptir. Bu bağlantılar bank denilen bir yapının içinde kümelenmişlerdir. CLB, FPGA'nın temel lojik ünitesidir [33]. Her CLB 4 veya 6 girişe sahip yapılandırılabilir anahtarlama matrisi, seçme elemanları ve flip-floplardan oluşur

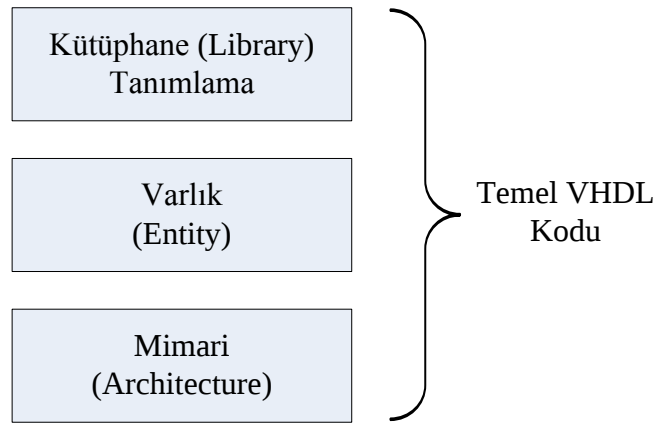
[33]. Anahtarlama matris; tümleşik lojik, kaydırmalı kaydedici ve rastgele erişimli bellek (RAM) yapılarını işletmede kullanılır. Programlanabilen bağlantılar ise CLB'lerin kendileri ve CLB'lerin giriş/çıkış kapıları arasındaki sinyalleri yönlendirir [33]. CLB'ler iki farklı temel yapıdan oluşmaktadır. Bunlar çoklayıcı (MUX; Multiplexer) ve doğruluk tablosu yapılarıdır (LUT; Look Up Table) [34]. MUX tabanlı yapı lojik fonksiyonları gerçekleştirmek için çoklayıcıları kullanır. LUT yapısında ise giriş sinyalleri doğruluk tablosunun adresini belirler. Bu adreslere karşılık gelen değer fonksiyonunun çıkışı olarak belirlenir. LUT yapısının içinde SRAM hücreleri mevcuttur. Bu hücrelerin içine, girişlere karşılık gelen fonksiyon çıkışları kaydedilir [34].

FPGA'ların çeşitli programlama teknolojileri vardır [31]. Bu yapılar silinebilir programlanabilir salt okunur bellek (EPROM; Erasable Programmable Read Only Memory), elektriksel etkiyle silinebilir programlanabilir salt okunur bellek (EEPROM; Electrically Erasable Programmable Read Only Memory), Flash, Sabit hafıza ve Karşıt sigorta (Anti-fuse) olarak sıralanabilir. Flash, Sabit hafıza ve Anti-fuse yapıları modern FPGA'larda geniş bir kullanım alanına sahiptir. Sabit hafıza yapısı durağan rastgele erişimli bellek (SRAM; Static Random Access Memory) temelli teknolojiyi esas almıştır. SRAM programlama teknolojisinin kullanımı FPGA'larda derin bir etkiye sahiptir. Bu teknolojinin iki temel avantajı vardır. Bunlar yeniden programlanabilmesi ve tümleyici metal oksit yarıiletken (CMOS; Complementary Metal Oxide Semiconductor) teknolojisini kullanmasıdır. Bunun yanında ise bazı dezavantajları da vardır. Programlanabilir elamanı oluşturmak için 5 veya 6 transistöre ihtiyaç duyulur. Bu da kullanılan alanın artmasına neden olur. Diğer bir dezavantaj ise enerji kesildiğinde içindeki bilgilerin kaybolmasıdır. Enerji kesilip yeniden verildiğinde FPGA'nın yeniden programlanması gerekir. Flash/EEPROM teknolojisinde ise enerji kesilse bile program silinmez. Böylece programın saklandığı harici hafızalara ihtiyaç duyulmaz. Enerji geldiği anda program çalışmaya başlar. Bu teknolojinin bazı dezavantajları vardır. Donanım tanımlama diliyle yazılan programın FPGA'ya yüklenmesi veya FPGA içindeki mevcut programın silinmesi işlemlerinin kaç kez yapılacağı sınırlıdır. Diğer bir önemli dezavantaj ise standart olmayan CMOS işlemine ihtiyaç duymaktadır [31]. Karşıt-sigorta teknolojisinde ise normal bağlantılar arasında yüksek direnç özelliği vardır. Programlama işleminden sonra bu bağlantılar düşük direnç özelliği göstermektedir [31]. Karşıt-sigorta yapısı için özel bir teknoloji

gerekir ve programlama işlemi yapıldıktan sonra yeniden programlanamaz. Program enerji kesilse dahi silinmez.

2.3 Donanım Tanımlama Dili (VHDL; VHSIC Hardware Description Language)

VHDL donanım tanımlama dilidir. VHDL elektronik devre veya sistemin davranışını tanımlar. VHDL, VHSIC donanım tanımlama dili anlamına gelmektedir [35]. VHSIC (Very High Speed Integrated Circuits), çok yüksek hızlı tümleşik devrelerin kısaltılmasıdır. Bu dil 1980 yılında Birleşik Devletler Savunma Bakanlığının finansmanı ile gerçekleştirilmiştir [35]. İlk versiyonu VHDL 87'dir. Sonra VHDL 93 olarak geliştirilmiştir. VHDL, IEEE (Institute of Electrical and Electronics Engineers) tarafından standartlaştırılmış ilk donanım tanımlama dilidir (IEEE 1076). Ek olarak IEEE 1164 standardı getirilmiştir. VHDL devre benzetimine benzer şekilde devre sentezlemesini hedeflemiştir. Şekil 2.5'de VHDL kodunun temel yapısı görülmektedir [35].



Şekil 2.5 VHDL temel kod yapısı

Kütüphane (Library) tanımlama: Tasarımda kullanılacak tüm kütüphaneler listelenir. Örneğin ieee, std, work gibi.

Varlık (Entity): Devrenin I/O pinleri tanımlanır.

Mimari (Architecture): Devrenin nasıl davranacağını belirttiği kodlar burada yazılır.

➤ Kütüphane (Library):

Kütüphane sıkça kullanılan kod parçalarının toplandığı yerdir. Bu kod parçalarının kütüphane içine konması sayesinde tekrar tekrar kullanılması ve diğer tasarımların bu kodları kullanması sağlanmıştır. Kütüphanenin temel yapısı Şekil 2.6'da gösterilmiştir

[35]. Kodlar genellikle fonksiyon, prosedür veya bileşen (component) formunda yazılır. Bu kodlar paket içinde tanımlanır ve hedef kütüphane içine derlenirler.



Şekil 2.6 VHDL kütüphanesinin temel yapısı

Kütüphaneyi belirtmek için 2 komut satırı kullanılır [35]. Bu satırlardan bir tanesinde kütüphane ismi belirtilir diğesinde ise “use” tanımı kullanılır.

```
LIBRARY kütüphane_ismi;  
USE kütüphane_ismi.paket_ismi.paket_parcalari;
```

Tasarım için en az 3 farklı kütüphaneden 3 tane paket kullanılır. Bunlar şu şekilde sıralanabilir.

- ieee.std_logic_1164 (ieee kütüphanesinden)
- standart (std kütüphanesinden)
- work (work kütüphanesinden)

Örnek:

```
Library ieee;  
Use ieee.std_logic_1164.all;  
Library std;  
Use std.standart.all;  
Library work;  
Use work.all;
```

std ve work kütüphanelerinin başlangıçta tanımlanmasına gerek yoktur. IEEE kütüphanesi başlangıçta tanımlanmalıdır [35]. IEEE kütüphanesi içindeki “std_logic_1164” paketi çok-seviyeli lojik sistemleri belirler. “std” kaynak kütüphanesidir (veri tipleri, text giriş/çıkışlar, vb.). “work” kütüphanesi ise tasarımın kaydedildiği yerdir (.vhd uzantılı dosya, derleyici ve simülatör tarafından oluşturulan dosyalar buradadır).

IEEE kütüphanesi birçok paket içerir bunlar şu şekilde sıralanabilir [35].

- std_logic_1164: std_logic (8 seviyeli) ve std_ulogic (9 seviyeli) çok değerli lojik sistemleri belirler.
- std_logic_arith: signed and unsigned veri tiplerini ve bu tiplerle alakalı aritmetik ve karşılaştırma işlemlerini belirler.
- std_logic_signed: std_logic_vector tipindeki değişkenlerle yapılan işlemleri sanki “işaretili” tipindeki değişkenle yapıyormuş gibi gerçekleştiren fonksiyon içerir.
- Std_logic_unsigned: std_logic_vector tipindeki değişkenlerle yapılan işlemleri sanki “işaretsiz” tipindeki değişkenle yapıyormuş gibi gerçekleştiren fonksiyon içerir.

➤ **Varlık (Entity):**

Varlık devrenin tüm giriş ve çıkışlarının belirlendiği yerdir [35]. Kullanımı şu şekildedir:

```
ENTITY entity_adı IS
PORT(
    Port_adı: sinyal_modu sinyal_tipi;
    Port_adı: sinyal_modu sinyal_tipi;
    ...
);
END entity_adı;
```

Sinyal modu IN, OUT, INOUT veya BUFFER şeklinde tanımlanır. IN ve OUT tek yönlü pinlerdir. INOUT iki yönlüdür. BUFFER ise çıkış sinyalinin içten kullanılması gerektiğinde kullanılır. Sinyal tipi BIT, STD_LOGIC, INTEGER gibi olabilir. Entity_ismi VHDL için ayrılmış kelimeleri dışında isteğe bağlıdır.

Örnek: AND kapısına ait VHDL kodu aşağıdaki gibi tanımlanmıştır.

```
ENTITY and_kapi IS
    PORT(m , n: IN BIT;
          y: OUT BIT);
END and_kapi;
```

Bu örnekte AND kapısı tanımlanmıştır. Bu kapsamda iki giriş (m,n) ve bir çıkış (y) oluşturulmuştur. Bütün sinyallerin tipi BIT olarak belirtilmiştir.

➤ **Mimari (Architecture):**

Mimari devrenin nasıl davranacağını tanımlar. Yazımı şu şekildedir.

```
ARCHITECTURE architecture_ismi OF entity_ismi IS
[tanımlamalar: Sinyal tanımlamaları vs.]
BEGIN
(kodlar BEGIN-END bloğunun içine yazılır)
END architecture_ismi;
```

Mimari iki kısımdan meydana gelir. 1) tanımlama kısmı (isteğe bağlı); sinyal ve sabitler tanımlanır. 2) kod kısmı (Begin-End arası); kodlar yazılır. Mimari ismi verilirken VHDL için ayrılmış kelimeler verilmemeye dikkat edilmelidir.

Örnek: Bir önceki örnekte and_kapi için varlık tanımlandı. Bu örnekte ise and_kapi'nin yapısı belirtilecektir.

```
ARCHITECTURE mimari OF and_kapi IS
```

```
BEGIN
```

```
y<= m AND n;
```

```
END mimari;
```

Bu yapıda iki giriş sinyali AND kapısının girişlerine uygulanıyor ve sonuç çıkış pinine aktarılıyor. Atama işlemi için “<=” operatörü kullanılmıştır. İsim olarak “mimari” kullanılmıştır. Bu yapıda tanımlama kısmı kullanılmamıştır ve kod kısmında tek bir atama işlemi gerçekleştirilmiştir.

BÖLÜM III

YÖNTEM

Tez çalışmasında, MR görüntülerindeki dokusal özniteliklerin çıkarılması için komşu beneklerin gri seviye değerlerine dayanan öznitelik çıkarma yöntemi ve bölütleme işlemi için k -NN sınıflayıcı ile GAL ağı kullanılmıştır. Çalışmanın ilk aşamasında, önerilen yöntemlerin MATLAB ve Microsoft Visual C# programları kullanılarak bilgisayarla benzetimi gerçekleştirilmiştir. İkinci aşamada, aynı yöntemler FPGA donanım platformu üzerinde VHDL kodlama ile gerçekleştirilmiştir. Hem bilgisayarla benzetim hem donanımla gerçekleştirilen bölütleme sonuçları karşılaştırmalı olarak incelenmiştir.

3.1 Öznitelik Çıkarma Yöntemi

Öznitelik vektörleri ilgilenilen beneğin 1-komşuluğundaki benek şiddetleri göz önüne alınarak Eşitlik 3.1'deki gibi oluşturulmuştur. Şekil 3.1'de 1-komşuluktaki benekler gösterilmiştir.

$$\vec{OV} = [OV1, OV2]$$

$$OV1 = I5$$

$$OV2 = (I1 + I2 + I3 + I4 + I6 + I7 + I8 + I9)/8 \quad (3.1)$$

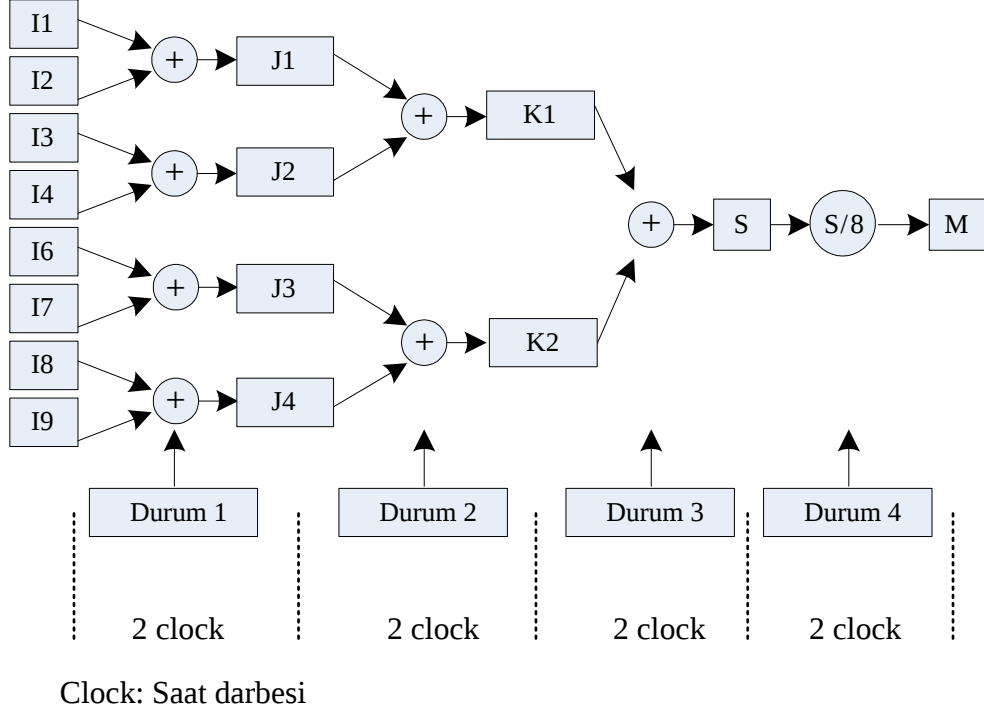
Burada $\vec{OV}$, öznitelik vektörünü; $OV1$, öznitelik vektörünün 1. elemanını, $OV2$, öznitelik vektörünün 2. elemanını göstermektedir.

İlgilenilen beneğe ilişkin iki adet öznitelik çıkarılmıştır. Bunlardan ilki seçilen beneğin kendi gri seviye değeridir. İkincisi ise merkez beneğin (ilgilenilen benek) etrafındaki komşu beneklerin gri seviye değerlerinin ortalamasıdır.

Şekil 3.1(a)'da ilgilenilen benek ve komşu benekler gösterilmiştir. FPGA'da yapılan ortalama bulma işleminin blok diyagramı Şekil 3.1(b)'de gösterilmiştir.

I1	I2	I3
I4	I5	I6
I7	I8	I9

(a)



(b)

Şekil 3.1 a) İlgilenilen benek ve 1-komşuluğu b) Ortalama bulma işlemi blok diyagramı

FPGA işlemlerini gerçekleştirirken sistemde bulunan ana saat darbesini (clock) referans alır. Gelen saat darbesinin yükselen veya düşen kenarına göre işlem yapılır. Matematiksel veya lojik fonksiyonların sonucuna ulaşmak için gereken saat darbesinin sayısı sistem tasarımı için önemlidir. Tasarlanan sistemde işlemlerin minimum saat darbesi kullanılarak yapılması hedeflenmektedir. Aynı veya benzer işlemlerin seri bir şekilde tekrar tekrar yapılması sistemin yavaşlamasına neden olmaktadır. Benzer işlemlerin aynı saat darbesinde paralel olarak yapılması hız performansında kayda değer bir avantaj sağlamaktadır. Ancak sistemde kullanılan paralelleştirme, FPGA içindeki programlanabilir alan tüketiminin artmasına neden olur.

Öznitelik çıkarma ve bölütleme yöntemlerinin matematiksel ifadelerinde bulunan toplama, çıkarma, bölme, çarpma ve karekök alma işlemleri için Xilinx firmasının ISE Design Suite 12.4 derleyicisinde bulunan 32 bit floating point bileşenler (component)

kullanılmıştır. Bileşenlerin işlem sonuçlarını elde etmek için kullandıkları saat darbesi sayısı çizelge 3.1’de verilmiştir.

Çizelge 3.1 Bileşenlerin işlem sonuçlarını elde etmek için kullandıkları saat darbesi sayısı

Bileşen (Component)	İşlemini tamamlaması için gereken saat darbesi
Toplama	2
Çıkarma	2
Bölme	2
Çarpma	2
Karekök alma	2

FPGA üzerinde gerçekleştirilen öznitelik çıkarma işleminde bazı iyileştirmeler yapılmıştır. İkinci öznitelik çıkarılırken ortalama alınması gerekmektedir. Ortalama bulma işlemi 4 durumdan oluşmaktadır. İlk iki durumda bazı paralelleştirmeler yapılmıştır. Durum 1’de I1 ve I2, I3 ve I4, I6 ve I7, I8 ve I9 değerleri aynı anda toplanarak J1, J2, J3, J4 sonuçları üretilir. Durum 2’de ise J1 ve J2, J3 ve J4 değerleri aynı anda toplanarak K1, K2 sonuçları üretilir. Durum3 ve Durum4 de paralel işlem yapılmamaktadır. Bu paralelleştirmeler hesaplama süresini azaltmaktadır. Ortalama bulma işlemi toplam 8 saat darbesinde gerçekleşmektedir.

3.2 *k*-En Yakın Komşu (*k*-NN; *k*-Nearest Neighbor) Sınıflayıcısı

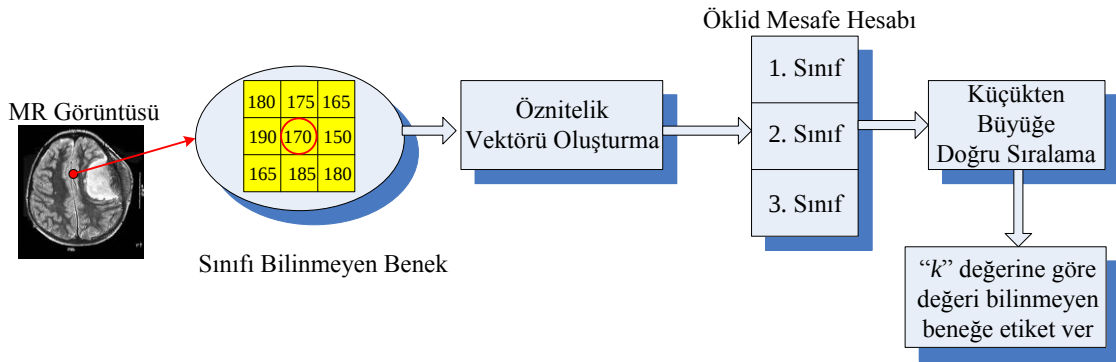
k-NN sınıflayıcısında, karar kuralı sınıflandırılacak giriş vektörüne *k* yakınındaki komşularının sınıf etiketlerine dayanarak bir etiket vermektedir. Bu karar kuralının tercih edilmesinin sebebi, getirdiği hesap basitliği ve öznitelik vektörü sayısının az olduğu birçok problemde iyi sonuçlar vermesidir.

k-NN sınıflayıcısında, eğitim işlemi gerçekleştirilmez. Sınıflara ait öznitelik vektörlerinden ve bu vektörlerin etiketlerinden oluşan bir küme vardır. Bu küme kullanıcı tarafından oluşturulur ve sınıflama süresince sabit kalır. En yakın komşuluğu belirlemede kullanılan ölçüt Öklid mesafesinin hesaplanmasıdır. Bu mesafe hesabı sınıfı bilinmeyen bir beneğin hangi sınıfın özelliklerine daha yakın olduğunu belirlemede kullanılır. En yakın

komşu öznitelik vektörünü hesaplamada kullanılan Öklid mesafe ölçütü Eşitlik 3.2’de verilmiştir [36].

$$d(x_i, x_l) = \sqrt{(x_{i1} - x_{l1})^2 + (x_{i2} - x_{l2})^2 + \dots + (x_{ip} - x_{lp})^2} \quad (3.2)$$

Burada, x_i , sınıfı bilinmeyen giriş beneğine ait öznitelik vektörünü; x_l , sınıfa ait öznitelik vektörünü; p , beneğe ait öznitelik vektörünün içindeki eleman sayısını göstermektedir. Şekil 3.2’de k -NN sınıflayıcı algoritmasının blok diyagramı gösterilmiştir.



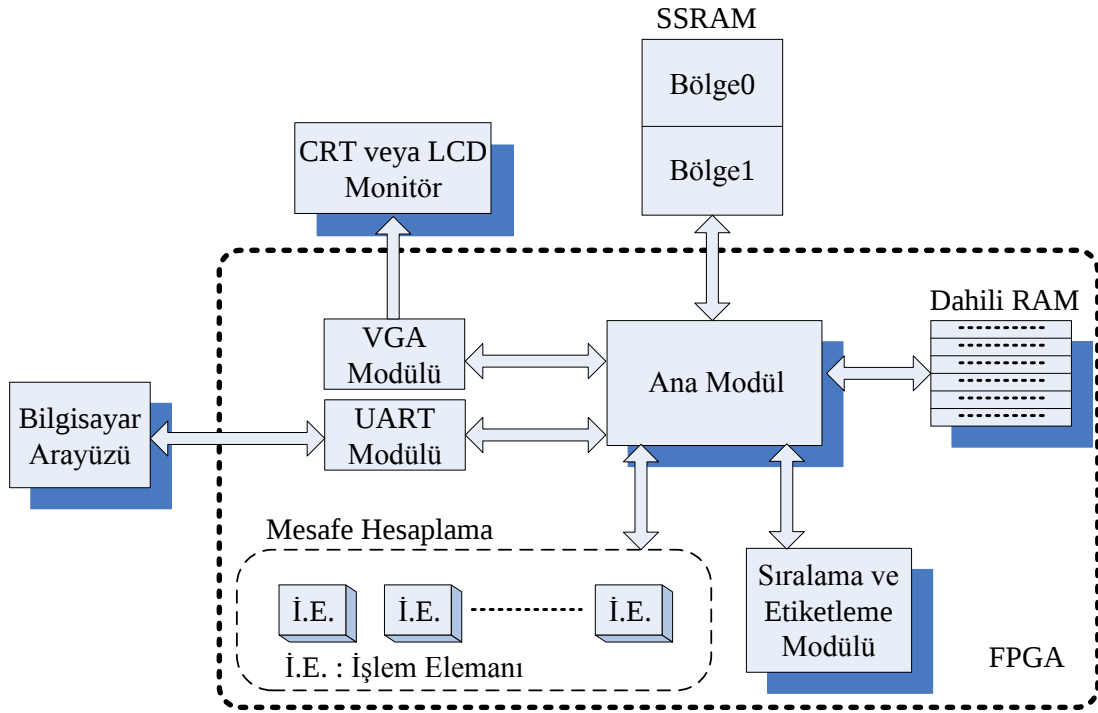
Şekil 3.2. k -NN sınıflayıcı algoritmasının blok diyagramı

Sınıfı bilinmeyen bir beneğin hangi sınıfa ait olduğunu belirlemek için öncelikle sınıflanacak beneğe ait öznitelik vektörü ile sınıflara ait öznitelik vektörleri arasındaki mesafe Eşitlik 3.2 ile hesaplanır. Hesaplanan bu mesafeler etiketleri ile beraber küçükten büyüğe doğru sıralanır. En küçük “ k ” adet mesafenin etiketleri incelenir. Her bir sınıfın kaç tane etiketi olduğu hesaplanır. Hangi sınıfın etiketi daha fazla ise o sınıfın etiketi sınıfı bilinmeyen beneğin etiketi olarak atanır. “ k ” değeri belirlenirken k ’nın çift sayı olmamasına dikkat edilmelidir. “ k ” değeri çift sayı alınırsa; en küçük “ k ” adet mesafenin etiketlerini sayma işlemi sonunda; sınıflara ait etiket sayıları eşit olabilir. Bu durumda karar mekanizması sınıfı bilinmeyen beneğe yanlış etiket değeri verebilmektedir.

3.2.1 *k*-NN sınıflayıcısının FPGA üzerinde gerçekleştirilmesi

MR görüntülerin bölütleme sonuçlarının yüksek doğrulukta elde edilmesi doğru tanı için en önemli unsurlardan birisidir. Bu nedenle, *k*-NN sınıflayıcı algoritmasının FPGA üzerinde gerçekleştirilmesinde Öklid mesafesi hesaplanırken Eşitlik 3.2’de herhangi bir sadeleştirme yapılmamıştır. Hem sadeleştirme yapılmaması hem de kayan noktalı sayı formatının kullanılması işlemlerin doğruluğunu artırmaktadır. Buna karşın, hesaplamalardaki bu tercih FPGA üzerinde kullanılan alanın ve işlem süresinin de artmasına yol açmaktadır.

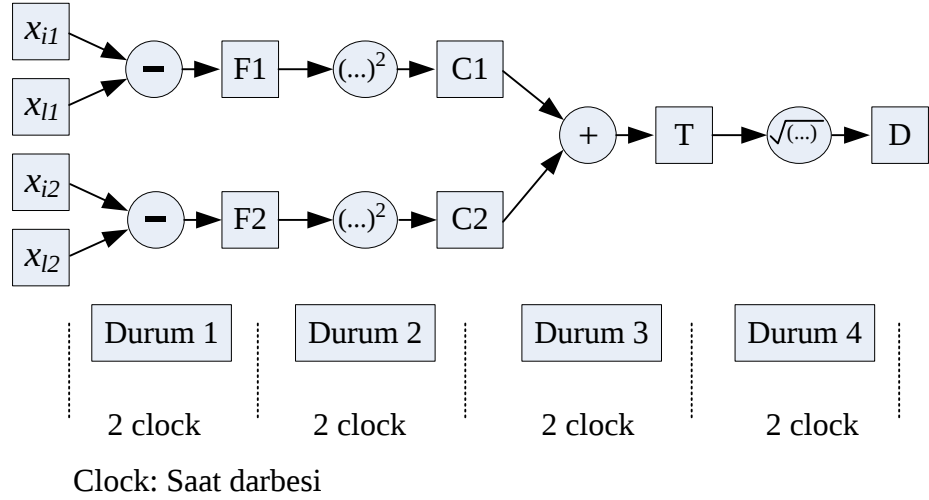
Çalışmada kullanılan MR görüntüler için 3 sınıf belirlenmiş ve her bir sınıfa ait 100’er adet öznitelik vektörleri seçilmiştir. Sınıfı bilinmeyen beneğe ait öznitelik vektörü ile sınıflara ait öznitelik vektörleri arasında Öklid mesafeleri tek tek hesaplanır. Klasik programlamada bu mesafeler sırayla hesaplanır. Mesafe hesaplama işlemi 300 kez çalıştırılır. Buda zaman kaybına neden olur. Zaman kaybını azaltmak için FPGA’nın paralel işlem yapabilme yeteneği kullanılmıştır. Çalışmada 10 adet işlem elemanından oluşan mesafe hesaplama bloğu tasarlanmıştır. İşlem elemanları Öklid mesafesini hesaplamak için kullanılır. Mesafe hesaplama bloğunda, işlem elemanları sayesinde 10 adet Öklid mesafesi paralel olarak aynı anda hesaplanabilmektedir. Sınıfı bilinmeyen beneğin öznitelik vektörü ile sınıflara ait öznitelik vektörleri arasındaki Öklid mesafelerinin hesaplanması için *mesafe hesaplama* bloğu 30 kez kullanılır. FPGA’da oluşturulan paralel yapılar hafızanın yoğun bir şekilde kullanılmasına neden olur. Hafızanın verimli kullanılması için Öklid mesafe hesabını gerçekleştiren işlem elemanından 10 adet oluşturulmuştur.



Şekil 3.3 FPGA’da gerçekleştirilen k -NN yapısının blok diyagramı

Şekil 3.3’de k -NN sınıflayıcısının FPGA üzerinde gerçekleştirilmesi blok diyagram olarak gösterilmiştir. Bu diyagramdaki blokların görevleri aşağıda belirtilmiştir.

- Ana modül: Diğer modülleri ve giriş çıkış işlemlerini kontrol eder.
- UART modülü: RS232 standardında haberleşmenin olduğu kısımdır.
- İşlem elemanı: Eşitlik 3.2’de verilen Öklid mesafesini hesaplamak için FPGA üzerinde oluşturulan yapıdır. Bu yapı Şekil 3.4’de gösterilmiştir. Öklid mesafesinin hesabı 4 durumdan oluşmaktadır. Hesaplama süresini kısaltmak için Durum 1 ve Durum 2’de paralelleştirmeler yapılmıştır. Durum 1’de giriş vektörünün elemanları ile sınıf vektörünün elemanlarının farkı alınmaktadır. Paralel yapı kullanılarak fark alma işlemleri aynı anda yapılmaktadır. Durum 2’de F1 ve F2 sonuçlarının karesi aynı saat darbesinde paralel olarak hesaplanmaktadır. Kare alma işlemi için F1 ve F2 sonuçları kendileri ile çarpılmıştır. Durum 3’de C1 ve C2 sonuçları toplanmıştır. Durum 3’de elde edilen toplam işleminin karekökü Durum 4’de alınmıştır. Öklid mesafesinin hesaplanması 8 saat darbesinde gerçekleşmektedir.

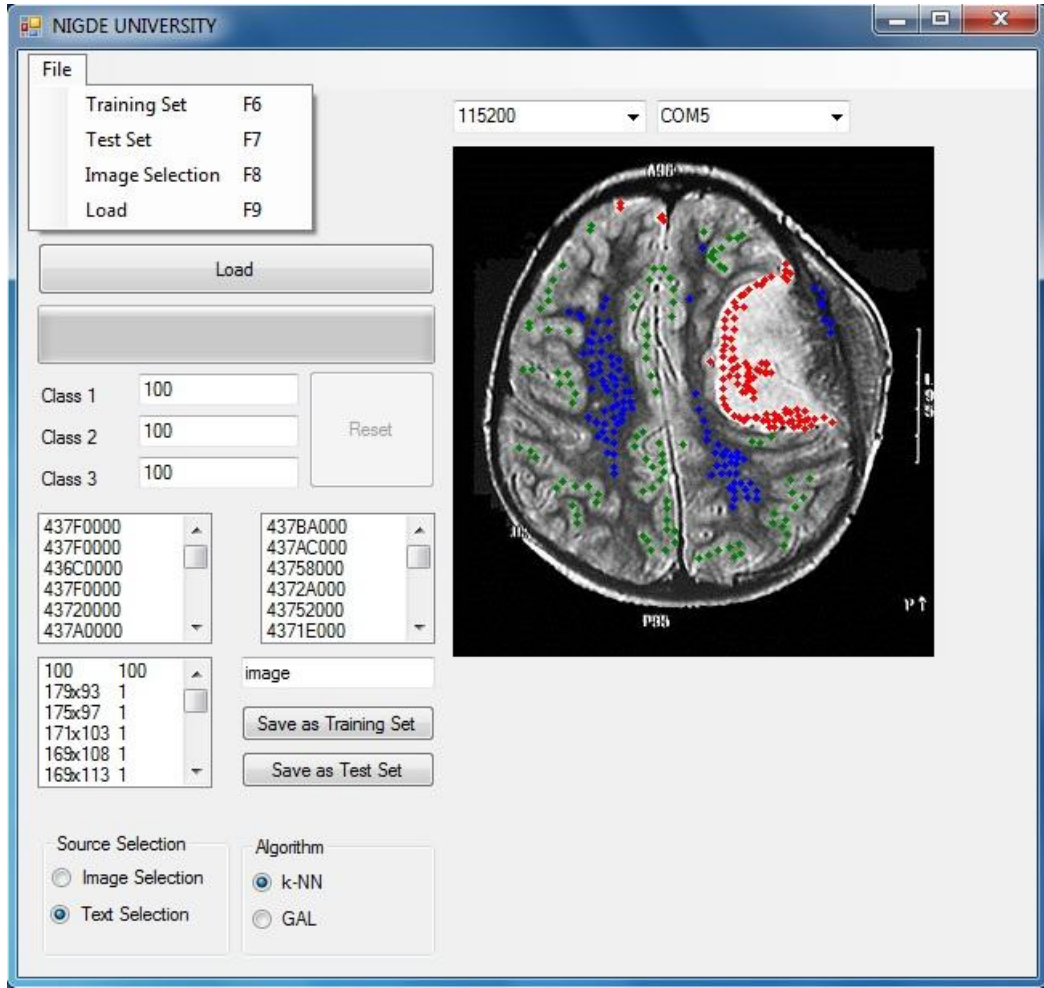


Şekil 3.4 İşlem elemanın FPGA’da gerçekleştirilmesi

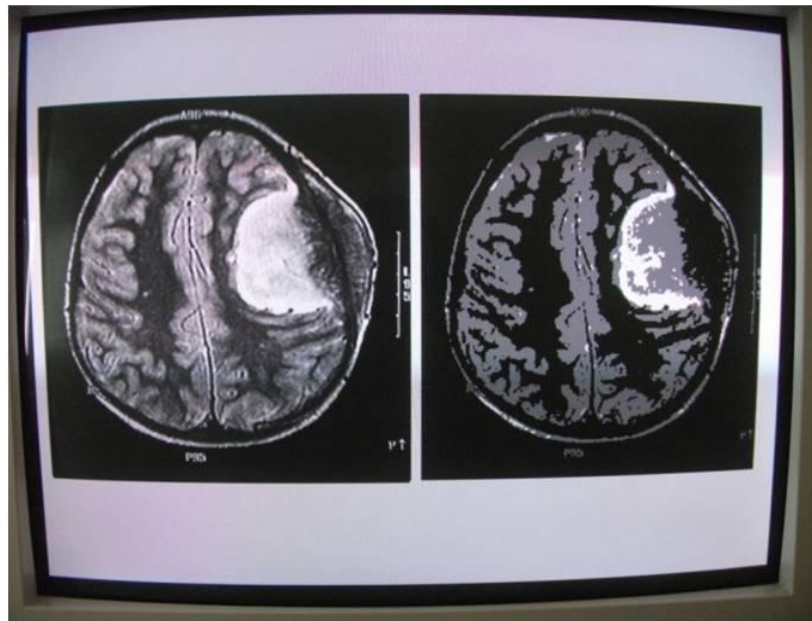
- Sıralama ve etiketleme modülü: Minimum mesafenin bulunması ve bu mesafeye göre etiket değerinin verilmesinde kullanılır.
- Dahili RAM: Eğitim kümesi, test kümesi ve sıralama işlemi için hesaplanan mesafeler burada saklanır.
- VGA modülü: CRT veya LCD monitöre görüntü aktarılmasını sağlar.
- SSRAM: Orijinal ve sınıflanmış görüntü burada saklanır. SSRAM içindeki hafıza iki bölüme ayrılmıştır. Bu bölümler sırasıyla Bölge0 ve Bölge1 olarak isimlendirilir. Bölge0’da orijinal görüntü Bölge1’de bölütlenmiş görüntü saklanmaktadır.
- CRT veya LCD Monitör: Bölütleme sonucu bu iki ekrandan birinde gösterilir.
- Bilgisayar Arayüz Programı: Bilgisayar ile FPGA arasındaki bağlantı bu program tarafından sağlanır.

Şekil 3.5’de gösterilen bilgisayar arayüzü FPGA ile bilgisayar arasındaki bağlantıyı sağlamaktadır. Bu arayüz Microsoft Visual C# dilinde yazılmıştır. Programın kullanımı kısaca şöyle özetlenebilir. Kullanıcı ilgilenilen görüntüyü seçerek (image selection) programa yükler. Programa sınıf sayıları girilir ve bu sayı kadar fare ile istenilen bölgelerden benekler seçilir. Seçilen bu benekler istenirse daha sonra kullanılmak üzere eğitim kümesi veya test kümesi olarak kaydedilebilir. Kaydedilen bu bilgiler kullanılmak istenirse eğitim kümesi (F7) ve test kümesi (F6) tıklanarak ilgili “txt” uzantılı metin dosyaları seçilir.

FPGA üzerinde gerçekleştirilen sistemin çalışması şu şekildedir. Şekil 3.5’de gösterilen bilgisayar arayüzü ile eğitim kümesi, test kümesi ve MR görüntüsü seçilir. Algoritma olarak “ k -NN” seçilir. Seçme işleminden sonra yükle (Load) butonuna basılır. Şekil 3.5’de görüldüğü gibi bilgisayarla haberleşme işlemi evrensel eşzamansız alıcı verici (UART-Universal asynchronous receiver/transmitter) modülü üzerinden gerçekleştirilmektedir. Bilgisayardan gelen veriler UART modülünden geçerek Ana modüle gelir. Ana modül ise; eğitim ve test kümelerini dâhili RAM bölgesine, MR görüntüsünü ise eşzamanlı durağan rastgele erişimli belleğin (SSRAM- Synchronous Static Random Access Memory) Bölge0 bölümüne kaydeder. Yapılan çalışmada her bir sınıftan 100 adet öznitelik vektörleri seçilmiştir. Bu vektörler dâhili RAM’de saklanmaktadır. Bunun sebebi ise dâhili RAM’in okuma-yazma süresinin SSRAM’in okuma-yazma süresine göre daha kısa olmasıdır. k -NN algoritmasında eğitim işlemi olmadığı için görüntü kaydedildikten sonra hemen bölütleme işlemine geçilir. Görüntüye ait ilk benek değeri SSRAM’den okunur ve bu beneğe ait öznitelik vektörü çıkarılır. Bu adımdan sonra dâhili RAM’deki 300 adet öznitelik vektörü ile sınıfı bilinmeyen beneğin arasındaki Öklid mesafesi hesaplanır. Mesafe hesaplamada İşlem elemanları (İE) kullanılmıştır. Mesafe hesaplama modülünde 10 adet İE bulunmaktadır. Bu elemanlar aynı anda Öklid mesafesi hesaplamaktadırlar. Böylece 300 adet öznitelik vektörünün bulunduğu bir kümede mesafe hesaplama modülünün 30 kere kullanılması yetmektedir. Hesaplanan mesafeler ve etiketler dâhili RAM’de saklanmaktadır. 300 adet mesafe hesaplandıktan sonra sıralama ve etiketleme işlemine geçilir. k değeri yapılan çalışmada 1 alınmıştır. Hesaplanan mesafelerin en küçüğü bulunur. Bulunan bu değerın etiketi sınıfı bilinmeyen beneğe verilir. Sınıfı belirlenen beneğin etiket değerine göre renk kodu belirlenir ve bu renk kodu SSRAM’in Bölge1 bölümüne kaydedilir. Çalışmada 3 sınıf olduğu için renk kodu olarak Beyaz, Gri ve Siyah (255, 128, 0) alınmıştır. Yukarıda anlatılan işlemler görüntüdeki tüm beneklere uygulanır. Bölütleme işlemi bittikten sonra bölütlenmiş görüntü iki şekilde gösterilebilir. Birincisi Bölge1’deki kayıtlı görüntü UART modülü ile tekrardan bilgisayara aktarılır. Bilgisayar arayüzü gelen bu bilgiyi “bmp; bitmap” uzantılı resim dosyası olarak kaydeder. Ayrıca FPGA üzerinde gerçekleştirilen video grafik dizisi (VGA-Video Graphics Array) modülü sayesinde katot ışınlı tüp (CRT-Cathode Ray Tube) veya sıvı kristal göstergeli (LCD-Liquid Crystal Display) monitörde hem orijinal görüntü hem de bölütlenmiş görüntü yan yana görüntülenir. Şekil 3.6’da bölütleme sonucunun CRT monitöre aktarılmış hali görülmektedir.



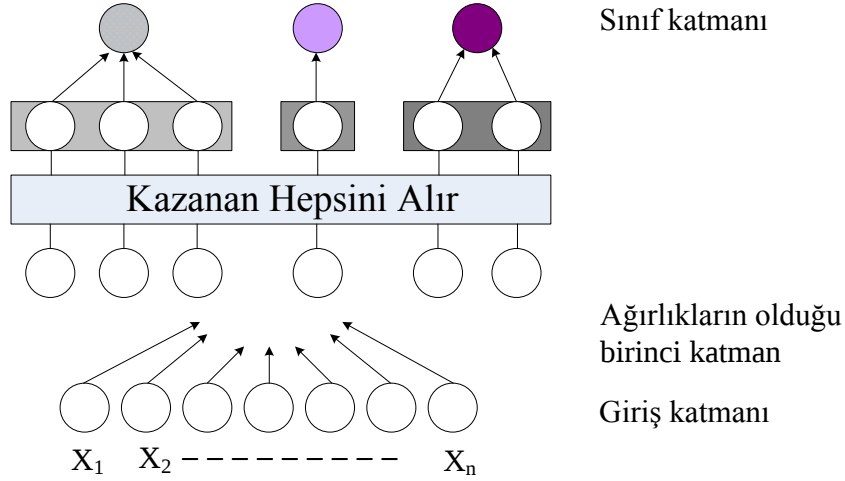
Şekil 3.5 Bilgisayar arayüzü



Şekil 3.6 CRT monitörde gösterilen bölütleme sonucu

3.3 Büyü ve Öğren (GAL; Grow and Learn) Ağı

GAL ağı, öznitelik uzayındaki sınıf sınırlarını en yakın mesafe ölçütüne göre belirler [37]. Şekil 3.7’de GAL ağının yapısı görülmektedir. GAL ağı dinamik bir yapıya sahiptir. Düğümler ve bağlantılar eğitim işlemi sırasında ihtiyaca göre eklenmektedir. GAL ağının temel avantajı öğrenme işlemini hızlı bir şekilde gerçekleştirebilmesidir.



Şekil 3.7 GAL ağının yapısı

GAL ağı denetimli öğrenmenin olduğu artımsal bir yapıya sahiptir [28]. GAL ağının çıkış düğümleri eğitim kümesinden öznitelik vektörleri seçilerek oluşturulur. Ağın eğitim algoritması aşağıda verilen adımları takip eder:

Adım 1. Her bir sınıftan birer adet öznitelik vektörü seç. Seçilen bu vektörleri GAL ağının çıkış katmanında bulunan başlangıç düğümleri olarak ayarla. Eğitim için gerekli iterasyon sayısını belirle ve başlangıç değerini sıfır olarak ayarla.

Adım 2. İterasyon sayısını bir artır. Eğer iterasyon sayısı belirlenen maksimum değere ulaşmış ise eğitim işlemini sonlandır. Aksi halde Adım 3’e git.

Adım 3. x_i olarak isimlendirilen bir adet vektörü eğitim kümesinden seç. Seçilen giriş vektörü ile GAL ağındaki mesafeleri Eşitlik 3.3’de verilen denklem ile hesapla ve minimum mesafeyi bul.

$$D_j = \sum_{i=1}^N (x_i - w_{ji})^2$$

$$D_m = \min_j \{D_j\} \quad (3.3)$$

Burada x_i ; giriş vektörünün i numaralı elemanını, w_{ji} ; j numaralı düğüm vektörünün i numaralı elemanını, N ; öznitelik vektörünün eleman sayısını temsil etmektedir. D_m ise giriş vektörü ile düğümler arasında hesaplanan mesafelerin minimum değerini belirtir. D_m değeri ayrıca m numaralı düğümün etiketine sahiptir. Eğer giriş vektörünün etiketi ile D_m değerinin etiketi aynı ise adım 2'ye git. Aksi takdirde adım 4'e git.

Adım 4. Giriş vektörünü GAL ağına yeni bir düğüm olarak ekle. Adım 2'ye git.

GAL ağının eğitim işleminde üretilen düğümler giriş vektörlerine bağlıdır. Eğitim işlemi sonunda ağa, sınıf sınırlarını en iyi şekilde temsil eden yeni düğümlerin eklenmesiyle bazı düğümler işlevselliğini kaybedebilir. Kullanışsız düğümlerin ağdan çıkarılması sistem performansını etkilemeyecektir. Ayrıca ağdan çıkarılan kullanışsız düğümler ağ yapısının küçülmesini sağlar. Böylece hem kullanılan hafıza birimi azalmış olur hem de ağ daha kısa sürede sınıflama işlemini gerçekleştirir. Kullanışsız düğümleri ağdan çıkarmak için kullanılan unutma algoritması aşağıda verilen adımları takip eder:

Adım 1. İterasyon sayısını belirle ve başlangıç değeri olarak sıfır ver. Adım 2'ye git.

Adım 2. İterasyon sayısını bir artır. Eğer iterasyon sayısı belirlenen maksimum değere ulaşmış ise eğitim işlemini sonlandır. Aksi halde Adım 3'e git.

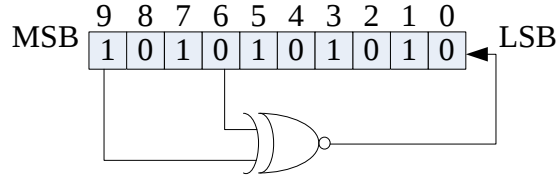
Adım 3. Sırayla ağdan bir düğümü çıkar. Çıkan düğümü ağa giriş olarak ver. Adım 4'e git.

Adım 4. Eşitlik 3.3'ü kullanarak mesafeleri hesapla. Minimum mesafe değerine sahip m numaralı düğümün etiketi ile giriş vektörünün etiketini karşılaştır. Eğer sınıf etiketleri eşit değil ise adım 5'e git. Aksi takdirde Adım 3'e git.

Adım 5. Giriş vektörünü ağa tekrardan ekle ve Adım 2'ye git.

3.3.1 Geri beslemeli lineer kaydırmalı kaydedici (LFSR)

k -NN algoritmasından farklı olarak GAL ağında eğitim işlemi gerçekleştirilmektedir. Başlangıç düğümlerinin seçiminde ve eğitim işleminde rastgele değerler kullanılmaktadır. Bilgisayar ortamında kullanılan programlama dilinin kendine ait rastgele sayı üreteç fonksiyonları veya kütüphaneleri vardır. Literatürde sayısal sistemlerde kullanılmak üzere rastgele sayı üretmek amacıyla çeşitli yöntemler geliştirilmiştir [38, 39]. Yapılan çalışmada FPGA üzerinde rastgele sayı üretmek için geri beslemeli lineer kaydırmalı kaydedici (LFSR; Linear Feedback Shift Register) kullanılmıştır.



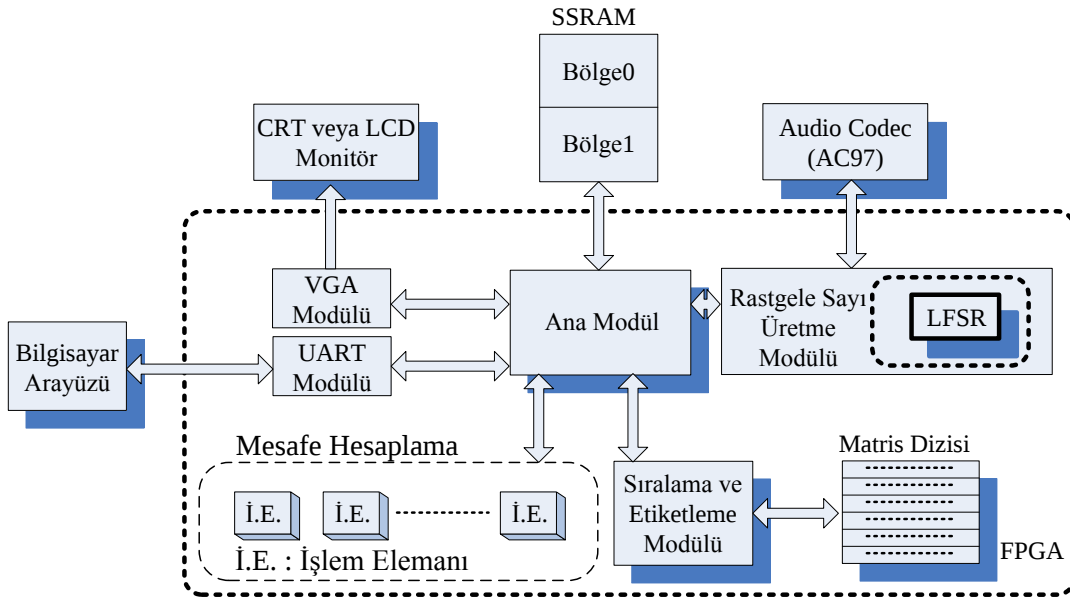
Şekil 3.8 10 bitlik LFSR yapısı

LFSR n-bitlik bir kaydırmalı kaydedicidir. Şekil 3.8’de 10 bitlik LFSR gözükmemektedir. LFSR’ın bazı bitleri XNOR lojik kapısına giriş olarak verilmiştir. XNOR işleminin sonucu ise kaydırmalı kaydedicinin en önemsiz bitine (LSB; Least Significant Bit) verilir. Sistemin çalışması kısaca şu şekildedir. Kaydırmalı kaydediciye başlangıç değeri (seed value) girilir. Değer girildikten sonra tüm bitler en önemsiz bitten (LSB) en önemli bite (MSB; Most Significant Bit) doğru bir bit kaydırılır. Kaydırma işleminden sonra LSB’nin yeni değeri olarak XNOR işleminin çıkışı verilir. 2^n-1 kere kaydırma işleminden sonra kaydırmalı kaydedici başlangıç değerine döner ve bir önceki çevrim sırasında sayı üretmeye devam eder. Görüldüğü gibi LFSR aslında n-bitlik bir sayıcıdır. 2^n-1 kaydırma işleminde (bir çevrim) sayıların hangi sırayla geleceği LFSR’ın başlangıç değerine bağlıdır. Her bir çevrimden sonra başlangıç değeri değiştirilerek üretilen sayıların sırası değişmiş olur. Böylece rastgelelik artmış olur. Bu açıklamalar doğrultusunda LFSR sayısal sistemlerde Sözcük-sayı üretici (Pseudo-Random Generator) olarak kullanılabilir [40-42].

LFSR’a bir başlangıç değeri verilerek bir çevrim süresi geçtiğinde (2^n-1 kere kaydırma işlemi yapıldığında) LFSR’ın değeri tekrardan başa dönmektedir. Her çevrimde üretilen sayıların sırası aynıdır. Bu durum istenilen rastgeleliği sağlamamaktadır. Bu çalışmada rastgeleliği artırmak için bazı iyileştirmeler yapılmıştır. Çevrim sonunda farklı bir başlangıç değeri verilerek farklı sırada sayılar üretilebilir. Başlangıç değeri, değişen bir analog sinyalin sayısal karşılığından alınırsa rastgele sayı üretmede istenilen iyileştirme elde edilir. Tez çalışmasında, rastgele sayı üretici için farklı başlangıç değerleri üretmek amacıyla gerçek zamanlı ses işareti kullanılmıştır. FPGA platformu üzerinde bulunan ses entegresi (AC97; Audio Codec) bu iş için kullanılmaktadır. AC97 girişine gelen ses işareti 20 bit çözünürlüğünde sayısal işarete dönüşmektedir. Bu sayısal işaretin 10 bit’lik kısmı LFSR’ın başlangıç değeri olarak verilir. LFSR’a başlangıç değeri verilirken bazı noktalara dikkat edilmelidir. Başlangıç değerinin tüm bitleri lojik 1 olmamalıdır. XNOR lojik kapısının girişlerine lojik 1 verilirse çıkış yine lojik 1 olur.

Kaydırma işleminden sonra LFSR'ın LSB bitine XNOR kapısının çıkışı verilir. Görüldüğü üzere LFSR'ın tüm bitleri yine lojik 1 olarak kalır ve böylece LFSR rastgele sayı üretemez. Yapılan çalışmada bu durumun önüne geçmek için bazı önlemler alınmıştır. AC97'den alınan başlangıç değerleri LFSR'a verilmeden önce kontrol edilmektedir. Eğer başlangıç değerinin tüm bitleri lojik 1 ise AC97'den yeni bir sayısal bilgi alınır.

3.3.2 GAL ağının FPGA üzerinde gerçekleştirilmesi



Şekil 3.9 FPGA’da gerçekleştirilen GAL ağının blok diyagramı

Şekil 3.9’da GAL ağının FPGA üzerinde gerçekleştirilmesi blok diyagram olarak gösterilmiştir. Bu diyagramdaki blokların görevleri aşağıda belirtilmiştir.

- Ana modül: Diğer modülleri ve giriş çıkış işlemlerini kontrol eder.
- UART modülü: RS232 standardında haberleşmenin olduğu kısımdır.
- İşlem elemanı: Mesafe hesaplamada kullanılır.
- Sıralama ve etiketleme modülü: Minimum mesafenin bulunması ve bu mesafeye göre etiket değerinin verilmesinde kullanılır.
- Matris Dizisi: Eğitim kümesi, test kümesi ve sıralama işlemi için hesaplanan mesafeler burada tutulur. Matris dizisi, matris formunda tanımlanmış sinyalden oluşmaktadır. Matris dizisinde okuma ve yazma işlemi çok hızlı gerçekleşmektedir. Kullanılan bu yapı sıralama ve etiketleme işlemlerindeki hız performansını artırmaktadır.
- Rastgele sayı üretme modülü: LFSR yapısı kullanılarak, GAL ağının eğitim algoritması için gerekli olan rastgele işlemleri gerçekleştirir.
- VGA modülü: CRT veya LCD monitöre görüntü aktarılmasını sağlar.
- SSRAM: Orijinal ve sınıflanmış görüntü burada saklanır. SSRAM içindeki hafıza iki bölüme ayrılmıştır. Bu bölümler sırasıyla Bölge0 ve Bölge1 olarak

isimlendirilir. Bölge0'da orijinal görüntü Bölge1'de bölütlenmiş görüntü saklanmaktadır.

- Audio Codec (AC97): Girişine gelen ses işaretini dijital bilgiye çevirir.
- CRT veya LCD Monitör: Bölütleme sonucu bu iki ekrandan birinde gösterilir.
- Bilgisayar Arayüz Programı: Bilgisayar ile FPGA arasındaki bağlantı bu program tarafından sağlanır.

Gerçeklenen sisteminin çalışması şu şekildedir. Bölüm 3.2.1'de açıklanan bilgisayar arayüzü ile eğitim kümesi, test kümesi ve MR görüntüsü seçilir. Algoritma olarak “GAL” seçilir. Seçme işleminden sonra yükle butonuna basılır. Gönderilen veriler UART modülünden geçerek Ana modüle gelir. Ana modül eğitim ve test kümesini matris dizisine kaydeder. İlgilenilen görüntü ise SSRAM'ın Bölge0 bölümüne kaydedilir. Eğitim işlemi için iterasyon sayısı belirlenir. Eğitim işlemine geçmeden önce Rastgele sayı üretme modülü iterasyon değeri kadar sayı üretir. Bu modül LFSR'dan oluşmaktadır. İterasyon değeri kadar rastgele sayı üretildikten sonra eğitim işlemine geçilir. Çalışmada eğitim kümesi için her sınıftan 100 adet öznitelik vektörü seçilmiştir. Eğitim sonunda yeni düğümler üretilir veya üretilmez. Başlangıç koşulları ve eğitim kümesinden elemanlar rastgele seçildiği için yapılan her yeni eğitimden sonra düğüm sayısı farklı olabilir. Eğitim işlemi bittikten sonra bölütleme işlemine geçilir. Görüntünün ilk beneği seçilir ve bu beneğe ait iki adet öznitelik çıkarılır. Öznitelik vektörleri ağa giriş olarak verilir. Mesafe hesaplama işlemi için şekil 3.8'de gösterilen İE'ler kullanılır. Mesafe hesaplama modülünde 10 adet İE vardır ve bu elemanlar paralel çalışarak aynı anda mesafe hesabı yaparlar. Düğüm sayısı 10 adetten fazla ise geri kalan düğümler için İE'ler tekrar kullanılır. İlgilenilen beneğin öznitelik vektörü ile ağın düğümlerindeki mesafe hesabı yapıldıktan sonra sonuçlar matris dizisinde saklanır. Sıralama ve etiketleme modülü ise öznitelik vektörlerini etiketleri ile birlikte küçükten büyüğe doğru sıralar ve en küçük mesafenin etiketini ilgilenilen beneğin etiketi olarak verir. Bu işlemler görüntüdeki tüm beneklere uygulanır ve bölütlenmiş görüntü SSRAM'ın Bölge1 bölümüne kaydedilir.

Bölütleme işlemi bittikten sonra bölütlenmiş görüntü iki şekilde gösterilebilir. Birincisi Bölge1'deki kayıtlı görüntü UART modülü kullanılarak tekrardan bilgisayara aktarılır. Bilgisayar arayüzü gelen bu bilgiyi “bmp; bitmap” uzantılı resim dosyası olarak

kaydeder. Ayrıca FPGA’da gerçekleştirilen VGA modülü sayesinde CRT veya LCD monitörde hem orijinal görüntü hem de bölütlenmiş görüntü yan yana görüntülenir.

BÖLÜM IV

MR GÖRÜNTÜLERİNİN BİLGİSAYAR ve FPGA ÜZERİNDE BÖLÜTLENMESİ

Tez çalışmasında beyinden alınmış üç farklı MR görüntüsü hem bilgisayar platformu hem de FPGA platformu üzerinde bölütlenmiştir. Bölütleme işlemi için k -NN sınıflayıcı ve GAL ağı ayrı ayrı kullanılmıştır. Her iki platform ve bölütleme yönteminde 1-komşuluklu gri seviye değerlerinin kullanılarak oluşturulduğu aynı öznitelik vektör kümeleri kullanılmıştır. Yöntemlerdeki tüm işlemler, bilgisayar platformunda MATLAB ve Microsoft Visual C# olmak üzere iki ayrı programlama dili, FPGA platformunda ise VHDL programlama dili kullanılarak gerçekleştirilmiştir. Her iki platform üzerinde gerçekleştirilen yazılımlarla elde edilen bölütleme performansları, hem bölütlenmiş görüntüler üzerinde görsel olarak hem de tablolar halinde nicel olarak karşılaştırmalı bir şekilde verilmiştir.

MATLAB ve Microsoft Visual C# programları 2.4 GHz işlemci hızına ve 3 GB'lık hafıza birimine sahip bilgisayar platformu üzerinde koşturulmuştur. FPGA platformu olarak Xilinx firmasının Virtex-5 XC5VLX110T FPGA serisi kullanılmıştır. FPGA işlemci hızı 100,5 MHz alınmıştır. Yöntemlerdeki sayısal işlemlerin doğruluğunu artırmak için matematiksel ifadelerde sadeleştirmeye gidilmemiş ve 32 bit kayan noktalı sayı formatı kullanılmıştır.

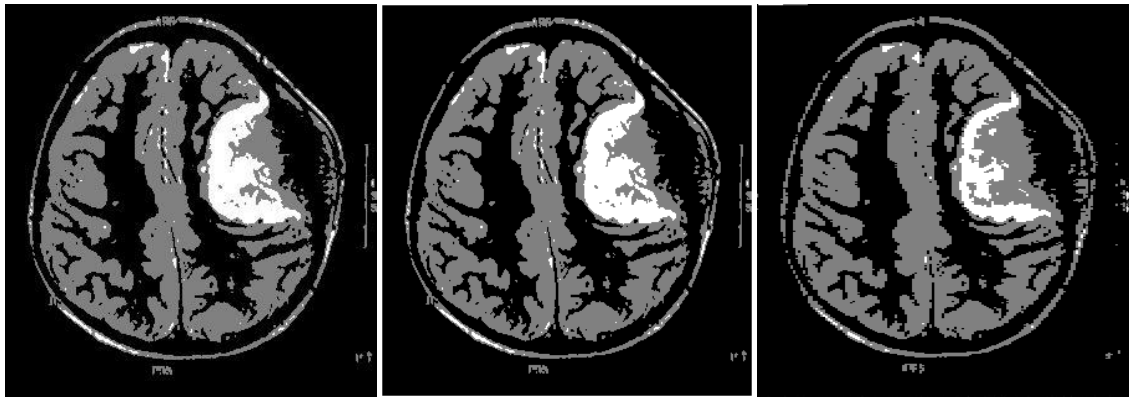
Çalışmada üç farklı gri seviyeli MR görüntüsü incelenmiştir ve her bir benek 8 bit çözünürlüğündedir. MR-1, MR-2 ve MR-3 görüntüleri, sırasıyla 300X319, 300X346 ve 256X256 boyutlarındadır. Hem k -NN sınıflayıcısı hem de GAL ağı için her sınıftan 100'er olmak üzere toplam 300 adet öznitelik vektörü kullanıcı tarafından seçilmiştir. Her iki yöntem için aynı öznitelik vektör kümesi kullanılmıştır. Yöntemlerin başarımını nicel olarak belirlemek amacıyla her sınıftan 50'ser olmak üzere toplam 150 adet öznitelik vektörü seçilerek test kümesi oluşturulmuştur. Çalışmada k -NN sınıflayıcısı için k değeri 1 alınmıştır. GAL ağının eğitim işleminde kullanılan iterasyon sayısı 1000 olarak seçilmiştir.

4.1 k -NN Sınıflayıcısı Kullanılarak Elde Edilen Bölütleme Sonuçları

Şekil 4.1’de orijinal MR-1 görüntüsü verilmiştir. Bu görüntü MATLAB ve Microsoft Visual C# programları kullanılarak k -NN sınıflayıcısı ile bölütlenmiştir. Benzer şekilde FPGA platformu üzerinde bölütleme işlemi gerçekleştirilmiştir. Şekil 4.2’de Bilgisayar ve FPGA üzerinde gerçekleştirilmiş bölütleme sonuçları verilmiştir.



Şekil 4.1 Orijinal MR-1 görüntüsü



(a)

(b)

(c)

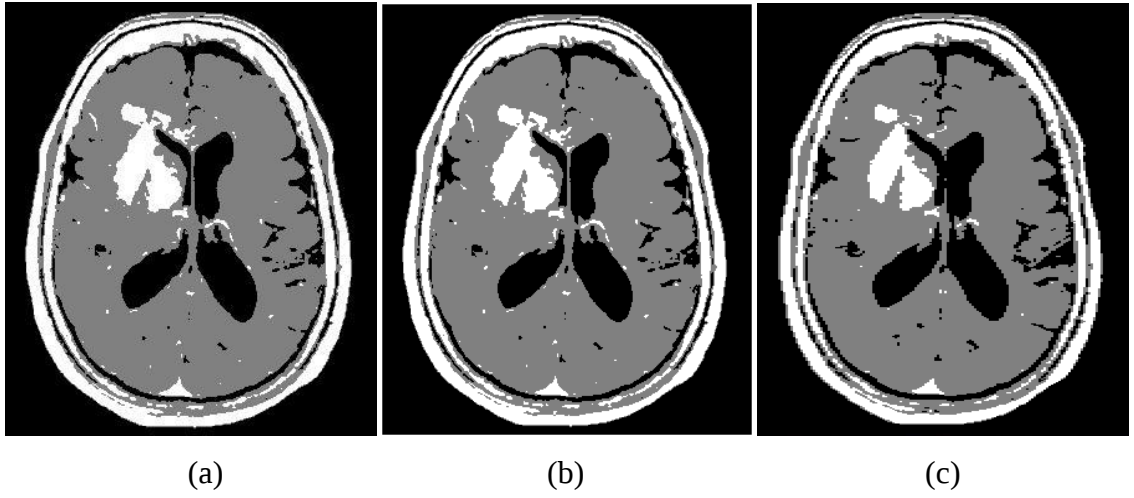
Şekil 4.2 k -NN sınıflayıcısı kullanılarak;

- a) MATLAB platformu üzerinde bölütlenmiş MR-1 görüntüsü
- b) Microsoft Visual C# platformu üzerinde bölütlenmiş MR-1 görüntüsü
- c) FPGA platformu üzerinde bölütlenmiş MR-1 görüntüsü

Şekil 4.3’de orijinal MR-2 görüntüsü verilmiştir. Bu görüntü MATLAB ve Microsoft Visual C# programları kullanılarak k -NN sınıflayıcısı ile bölütlenmiştir. Benzer şekilde FPGA platformu üzerinde bölütleme işlemi gerçekleştirilmiştir. Şekil 4.4’de Bilgisayar ve FPGA üzerinde gerçekleştirilmiş bölütleme sonuçları verilmiştir.



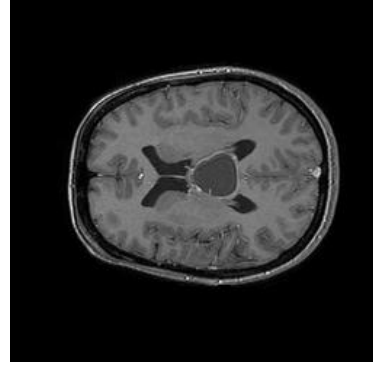
Şekil 4.3 Orijinal MR-2 görüntüsü



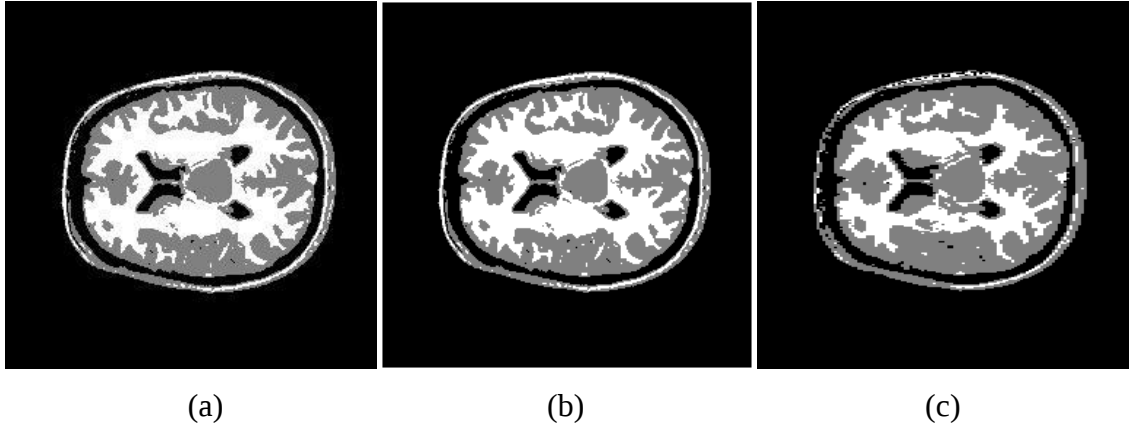
Şekil 4.4 k -NN sınıflayıcısı kullanılarak;

- a) MATLAB platformu üzerinde bölütlenmiş MR-2 görüntüsü
- b) Microsoft Visual C# platformu üzerinde bölütlenmiş MR-2 görüntüsü
- c) FPGA platformu üzerinde bölütlenmiş MR-2 görüntüsü

Şekil 4.5’de orijinal MR-3 görüntüsü verilmiştir. Bu görüntü MATLAB ve Microsoft Visual C# programları kullanılarak k -NN sınıflayıcısı ile bölütlenmiştir. Benzer şekilde FPGA platformu üzerinde bölütleme işlemi gerçekleştirilmiştir. Şekil 4.6’da Bilgisayar ve FPGA üzerinde gerçekleştirilmiş bölütleme sonuçları verilmiştir.



Şekil 4.5 Orijinal MR-3 görüntüsü



Şekil 4.6 k -NN sınıflayıcısı kullanılarak;

- a) MATLAB platformu üzerinde bölütlenmiş MR-3 görüntüsü
- b) Microsoft Visual C# platformu üzerinde bölütlenmiş MR-3 görüntüsü
- c) FPGA platformu üzerinde bölütlenmiş MR-3 görüntüsü

Çizelge 4.1’de k -NN sınıflayıcısı kullanılarak bilgisayarla benzetim ve FPGA üzerinde gerçekleştirilen bölütleme sonuçları karşılaştırmalı olarak verilmiştir.

Çizelge 4.1 k -NN sınıflayıcısının bilgisayarla benzetim ve FPGA sonuçları

Görüntü	Platform		İşlemci hızı (MHz)	İşlem süresi (sn)	Başarım (%)
MR-1	Bilgisayarla benzetim	MATLAB	2400	53,88	99,67
	Bilgisayarla benzetim	Microsoft Visual C#	2400	9	99,67
	FPGA	Virtex-5 LX110T	100,5	6	99,67
MR-2	Bilgisayarla benzetim	MATLAB	2400	61,42	99,67
	Bilgisayarla benzetim	Microsoft Visual C#	2400	9,64	99,67
	FPGA	Virtex-5 LX110T	100,5	6.5	99,67
MR-3	Bilgisayarla benzetim	MATLAB	2400	18,68	99,67
	Bilgisayarla benzetim	Microsoft Visual C#	2400	6,16	99,67
	FPGA	Virtex-5 LX110T	100,5	4,1	99,67

4.2 GAL Ağı Kullanılarak Elde Edilen Bölütleme Sonuçları

Şekil 4.7’de orijinal MR-1 görüntüsü verilmiştir. Bu görüntü MATLAB ve Microsoft Visual C# programları kullanılarak GAL ağı ile bölütlenmiştir. Benzer şekilde FPGA platformu üzerinde bölütleme işlemi gerçekleştirilmiştir. Şekil 4.8’de Bilgisayar ve FPGA üzerinde gerçekleştirilmiş bölütleme sonuçları verilmiştir.



Şekil 4.7 Orijinal MR-1 görüntüsü



(a)

(b)

(c)

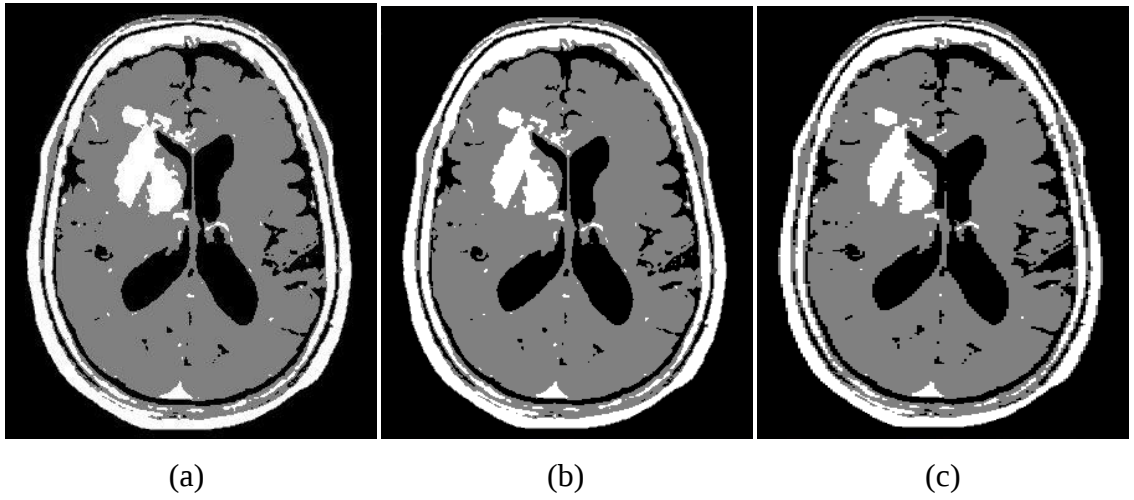
Şekil 4.8 GAL ağı kullanılarak;

- a) MATLAB platformu üzerinde bölütlenmiş MR-1 görüntüsü
- b) Microsoft Visual C# platformu üzerinde bölütlenmiş MR-1 görüntüsü
- c) FPGA platformu üzerinde bölütlenmiş MR-1 görüntüsü

Şekil 4.9’da orijinal MR-2 görüntüsü verilmiştir. Bu görüntü MATLAB ve Microsoft Visual C# programları kullanılarak GAL ağı ile bölütlenmiştir. Benzer şekilde FPGA platformu üzerinde bölütleme işlemi gerçekleştirilmiştir. Şekil 4.10’da Bilgisayar ve FPGA üzerinde gerçekleştirilmiş bölütleme sonuçları verilmiştir.



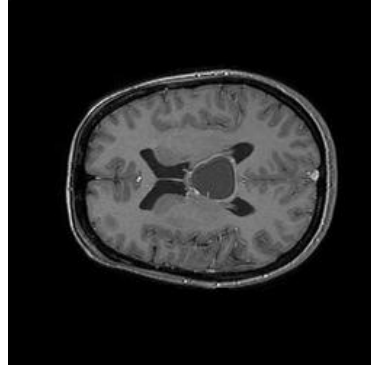
Şekil 4.9 Orijinal MR-2 görüntüsü



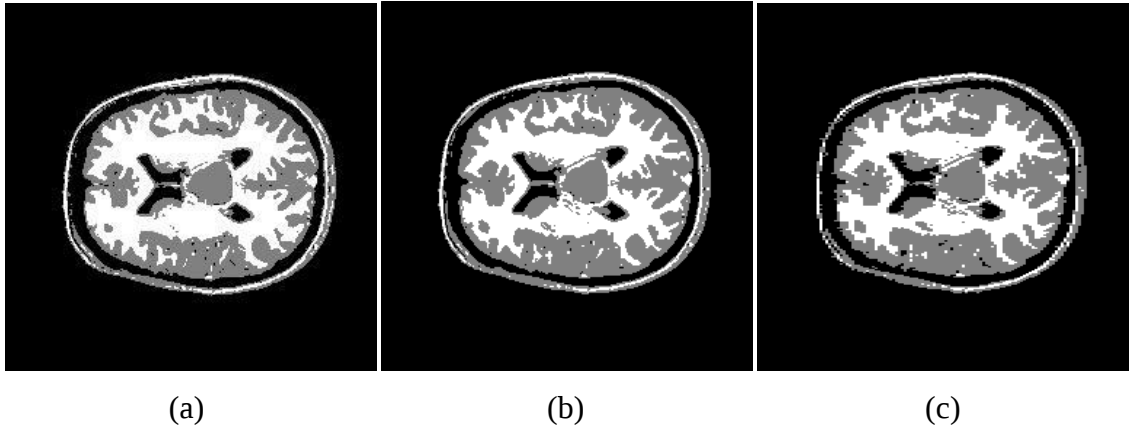
Şekil 4.10 GAL ağı kullanılarak;

- a) MATLAB platformu üzerinde bölütlenmiş MR-2 görüntüsü
- b) Microsoft Visual C# platformu üzerinde bölütlenmiş MR-2 görüntüsü
- c) FPGA platformu üzerinde bölütlenmiş MR-2 görüntüsü

Şekil 4.11’de orijinal MR-1 görüntüsü verilmiştir. Bu görüntü MATLAB ve Microsoft Visual C# programları kullanılarak GAL ağı ile bölütlenmiştir. Benzer şekilde FPGA platformu üzerinde bölütleme işlemi gerçekleştirilmiştir. Şekil 4.12’de Bilgisayar ve FPGA üzerinde gerçekleştirilmiş bölütleme sonuçları verilmiştir.



Şekil 4.11 Orijinal MR-3 görüntüsü



Şekil 4.12 GAL ağı kullanılarak;

- a) MATLAB platformu üzerinde bölütlenmiş MR-3 görüntüsü
- b) Microsoft Visual C# platformu üzerinde bölütlenmiş MR-3 görüntüsü
- c) FPGA platformu üzerinde bölütlenmiş MR-3 görüntüsü

Çizelge 4.2’de GAL ağı kullanılarak bilgisayarla benzetim ve FPGA üzerinde gerçekleştirilen bölütleme sonuçları karşılaştırmalı olarak verilmiştir.

Çizelge 4.2 GAL ağının bilgisayarla benzetim ve FPGA sonuçları

Görüntü	Platform		İşlemci hızı (MHz)	Düğüm sayısı (Ortalama)	Eğitim süresi (msn)	Bölütleme süresi (msn)	Başarım (%)
MR-1	Bilgisayarla benzetim	MATLAB	2400	5	122	9830	99,33
	Bilgisayarla benzetim	Microsoft Visual C#	2400	5	2,09	1110	99,33
	FPGA	Virtex-5 LX110T	100,5	5	0,3	72,61	99,33
MR-2	Bilgisayarla benzetim	MATLAB	2400	4	108	8830	99,33
	Bilgisayarla benzetim	Microsoft Visual C#	2400	4	2,01	1170	99,33
	FPGA	Virtex-5 LX110T	100,5	4	0,26	73,68	99,33
MR-3	Bilgisayarla benzetim	MATLAB	2400	6	133	8040	99,33
	Bilgisayarla benzetim	Microsoft Visual C#	2400	6	2,26	811	99,33
	FPGA	Virtex-5 LX110T	100,5	6	0,32	53,49	99,33

FPGA’lar yapılandırılabilir lojik elamanları kullanarak (CLB) istenilen fonksiyonu gerçekleştirmektedir. Bölüm 2.2’de anlatılan CLB’ler temelde iki yapıdan oluşmaktadır. Bunlar MUX ve LUT yapılarıdır. Çalışmada kullanılan FPGA’nın içinde bulunan CLB’lerde LUT yapısı mevcuttur. Çizelge 5.2’de k -NN sınıflayıcı ve GAL ağı kullanılarak FPGA üzerinde gerçekleştirilen bölütleme sisteminin kullandığı alan verilmiştir. Çizelge 4.3 incelendiğinde k -NN sınıflayıcı mevcut LUT’ların %56’sını,

GAL ağı ise LUT'ların %58'ini kullanmaktadır. *k*-NN sınıflayıcı algoritması oluşturulurken eğitim kümesi, test kümesi ve sıralama işlemleri için dağıtılmış hafıza yapısı (Distributed Memory) kullanılmıştır. Mevcut hafızanın %5'i tek portlu RAM olarak kullanılmıştır. GAL ağında ise toplam hafızanın %9'u kullanılmıştır. *k*-NN algoritmasında register'ların %13'ü, GAL ağında ise register'ların %15'i kullanılmıştır.

Çizelge 4.3 *k*-NN sınıflayıcı ve GAL ağı kullanılarak gerçekleştirilen bölütleme sisteminin FPGA'da kullandığı alan

	Register	LUT	Tek Portlu RAM	
<i>k</i>-NN	9492 (%13)	38711 (%56)	1024 (%5)	
	Register	LUT	Blok RAM	Toplam hafıza (KB)
GAL Ağı	10463 (%15)	40514 (%58)	14 (%9)	486 (%9)

BÖLÜM V

SONUÇLAR ve TARTIŞMA

Tez çalışmasında, MR görüntülerdeki dokusal özniteliklerin çıkarılması için komşu benek yoğunluğu temelli öznitelik çıkarma yöntemi, bölütleme işlemi için k -NN sınıflayıcı ve GAL ağı kullanılmıştır. Çalışmanın ilk aşamasında, MR görüntülerin bölütlenmesi amacıyla önerilen yöntemlerin MATLAB ve Microsoft Visual C# programları kullanılarak bilgisayarla benzetimi gerçekleştirilmiştir. İkinci aşamada, aynı yöntemler FPGA donanım platformu üzerinde VHDL kodlama ile gerçekleştirilmiştir.

Görüntü işleme uygulamalarında kullanılan öznitelik çıkarma ve bölütleme algoritmaları işlem yükünü artırmaktadır. Genel itibariyle bilgisayar yazılımları işlemlerini sıralı bir şekilde (seri programlama) yürütmektedir. İşlemlerin seri olarak yerine getirilmesi de işlem yükünün fazla olduğu görüntü işleme uygulamalarında önemli ölçüde zaman kaybına yol açmaktadır. Günümüz işlemcilerinin hızlarının ve çekirdek sayılarının artması görüntü işleme algoritmalarının hızlı bir şekilde gerçekleştirilmesine olanak sağlamaktadır. Fakat hız ve çekirdek sayısındaki artış işlemci fiyatını, güç tüketimini ve soğutma maliyetini artırmaktadır. FPGA'lar genel işlemcilerle göre düşük saat hızında çalışmaktadırlar. Böylece güç tüketimi ve soğutma maliyeti önemli miktarda azalmış olur. Yeniden yapılandırılabilme ve paralel işlem yeteneği sayesinde karmaşık hesapların yoğun olduğu görüntü işleme uygulamalarında FPGA'lar sıklıkla kullanılmaktadır.

Öznitelik çıkarma işlemi bölütleme başarımını doğrudan etkilemektedir. İyi seçilmemiş öznitelik vektörleri bölütleme başarımını ve işlem süresini olumsuz yönde etkilemektedir. Yapılan çalışmada MR görüntülerindeki dokusal özniteliklerin çıkarılması için komşu beneklerin gri seviye değerlerine dayanan öznitelik çıkarma yöntemi kullanılmıştır. Bu yöntemle, ilgilenilen benek ve komşuluğundaki diğer beneklerin ortalaması alınarak iki adet öznitelik vektör elemanı elde edilmiştir. MR görüntülerdeki doku dağılımı belirgin olduğundan seçilen öznitelik çıkarma yöntemi yüksek başarılı bölütleme sonuçlarını elde etmek için tatmin edici olmuştur. Aynı zamanda bu yöntem, basitliği nedeniyle toplam işlem süresinin düşmesini sağlamıştır (Çizelge 5.1).

Çalışmada MR görüntülerini bölütlemek amacıyla k -NN sınıflayıcısı ve GAL ağı her bir platform üzerinde ayrı ayrı denenmiştir. Öncelikle, önerilen yöntemler bilgisayar üzerinde MATLAB ve Microsoft Visual C# yazılımları kullanılarak gerçekleştirilmiş, aynı yöntemler daha sonra FPGA üzerinde gerçekleştirilmiştir. Literatürde k -NN algoritmasının FPGA üzerindeki çeşitli uygulamaları mevcuttur. GAL ağının FPGA üzerindeki uygulaması bu tez çalışmasında *ilk kez* gerçekleştirilmiştir.

GAL ağının başlangıç düğümlerinin seçiminde ve eğitim işleminde rastgele sayılara ihtiyaç duyulmaktadır. Literatürde rastgele sayı üretme işlemi için LFSR sıklıkla tercih edilmektedir. Tezde, rastgeleliği artırmak için bazı iyileştirmeler yapılmıştır. Çalışmada tasarlanan rastgele sayı üretici için farklı başlangıç değerleri (seed value) üretmek amacıyla gerçek zamanlı ses işareti kullanılmıştır. Bu işlem için FPGA platformu üzerinde bulunan ses entegresi (AC97 – Audio Codec) kullanılmıştır. AC97 entegresi girişine gelen ses işaretini sayısal işarete dönüştürmekte ve elde edilen sayısal işaret LFSR'ın başlangıç değeri olarak belirlenmektedir.

Bölütleme amacıyla kullanılan her iki sınıflayıcı için aynı öznetelik çıkarma yöntemi kullanılmıştır. k -NN sınıflayıcısı kullanılarak FPGA üzerinde gerçekleştirilen bölütleme işlemi, MATLAB kullanılarak gerçekleştirilen bölütleme işlemine göre; MR-1 ve MR-2 görüntüleri için yaklaşık 9 kat, MR-3 görüntüsü için yaklaşık 4.5 kat daha hızlıdır. Diğer yandan, FPGA üzerinde gerçekleştirilen bölütleme işleminin Microsoft Visual C# tarafından gerçekleştirilen bölütleme işlemine göre yaklaşık 1.5 kat daha hızlı olduğu gözlenmiştir. MATLAB, Microsoft Visual C# ve FPGA platformlarında gerçekleştirilen bölütleme işlemlerinde aynı bölütleme başarımları elde edilmiştir (Çizelge 4.1).

Çizelge 4.2'de her bir MR görüntüsü için GAL ağının eğitim süreleri, üretilen düğüm sayıları, bölütleme süreleri ve bölütleme başarımları verilmiştir. Yapılan çalışmada eğitim işlemi birçok kez tekrarlanmıştır. GAL ağındaki rastgele işlemler nedeniyle, eğitim işlemi sonucunda farklı sayıda düğümler üretilebilmektedir. Üretilen her bir düğüm bölütleme başarımını ve işlem sürelerini etkilemektedir. Karşılaştırmanın doğru yapılabilmesi için, eğitim işlemi her bir platform için aynı sayıda düğüm üretecek şekilde gerçekleştirilmiş ve bu durumda elde edilen veriler kaydedilmiştir. Çizelge 5.1'de, tüm MR görüntüleri için elde edilen verilerin (Çizelge 4.2) ortalama değerleri verilmiştir. GAL ağının FPGA üzerinde gerçekleştirilen eğitim işleminin MATLAB

benzetim platformuna göre yaklaşık 417 kat daha hızlı olduğu gözlenmiştir. Benzer şekilde, GAL ağının FPGA üzerinde gerçekleştirilen eğitim işlemi Microsoft Visual C# platformuna göre yaklaşık 7 kat daha hızlı gerçekleşmiştir. Bölütleme süreleri karşılaştırıldığında ise FPGA üzerindeki gerçeklemenin, MATLAB platformuna göre yaklaşık 133 kat daha hızlı olduğu gözlenmiştir. Benzer şekilde FPGA, Microsoft Visual C# programına göre yaklaşık 15 kat daha hızlı bölütleme gerçeklemiştir (Çizelge 5.1).

Çizelge 5.1 GAL ağı ve k -NN sınıflayıcısının karşılaştırılması

Yöntem	Platform	İşlemci hızı (MHz)	Düğüm Sayısı (Ortalama)	Eğitim Süresi (ms)	Bölütleme Süresi (sn)	Başarım (%)
k -NN	MATLAB	2400	k -NN Sınıflayıcısında Eğitim İşlemi Yoktur.		44,66	99,67
	Microsoft Visual C#	2400			8,27	99,67
	Virtex-5 LX110T	100,5			5,53	99,67
GAL Ağı	MATLAB	2400	5	121	8,9	99,33
	Microsoft Visual C#	2400	5	2,12	1,03	99,33
	Virtex-5 LX110T	100,5	5	0,29	0,067	99,33

Donanımla gerçekleştirme başarımının bilgisayarla benzetim başarımına oldukça yakın olması ve işlem süresindeki büyük üstünlüğü, FPGA platformlarının gerçek zamanlı uygulanmalarda kullanılmasına olanak sağlamaktadır. Tasarımlarındaki esneklik ve FPGA'ların yeniden yapılandırılabilmesi, hem maliyet açısından ve hem de yeni geliştirilen algoritmaların uygulanabilmesi yönünden tasarımcıya büyük kolaylıklar sağlamaktadır.

Çalışmada kullanılan k -NN sınıflayıcısının ve GAL ağının minimum mesafe hesaplama ve sıralama blokları benzer özellikler göstermektedir. k -NN sınıflayıcısında, sınıfı

bilinmeyen benek ile öznitelik vektör kümesindeki örnekler arasında Öklid mesafesi hesaplanır ve minimum mesafedeki vektörün etiketi ilgilenilen benek için etiketi olarak verilir. Görüntüdeki her bir benek için bu işlem tekrarlanır. GAL ağında ise, bölütleme işlemi için gereken en uygun düğümler eğitim kümesinden seçilir. Böylece daha az düğüm ile yüksek başarımlı ve yüksek hız elde edilmiş olur. FPGA üzerinde gerçekleştirilen yöntemlerden GAL ağı k -NN algoritmasına göre yaklaşık 83 kat daha hızlı bölütleme işlemi gerçekleştirmiştir (Çizelge 5.1).

Yukarıda yapılan değerlendirmeler göz önüne alındığında ve düşük geliştirme maliyeti, esnek yapısı, paralel ve hızlı işlem yapabilme özellikleri gibi üstünlükleri nedeniyle, FPGA teknolojisinin özellikle savunma sanayi ve medikal sektördeki gerçek zamanlı görüntü işleme uygulamalarında kullanımının her geçen gün hızla artması beklenmektedir.

KAYNAKLAR

- [1] Konez, O., Manyetik Rezonans Görüntüleme: Temel Bilgiler, İstanbul, 1995.
- [2] O'Donnell, L., Semi-Automatic Medical Image Segmentation, MIT, Massachusetts, 2001
- [3] Ecabert, O., Peters, J. and Schramm, H., Automatic Model-Based Segmentation of the Heart in CT Images, IEEE Transactions On Medical Imaging, 9, 1189-1201, 2008.
- [4] Tuceryan M. and Jain A. K., Texture Analysis, In The Handbook of Pattern Recognition and Computer Vision (2nd Edition), World Scientific Publishing Co., 207-248,1998.
- [5] Kim, N.D., Texture representation using wavelet filter banks, PhD Thesis, Iowa State University, Ames, Iowa, 2000.
- [6] Haralick R.M. and Shapiro L.G., Computer and Vision, Addison-Wesley Publishing Company, 1, 453–507, 1993.
- [7] Layer, G., Zuna, I., Lorentz, A., et al., Computerized ultrasound B-scan texture analysis of experimental fatty liver disease: influence of total lipid content and fat deposit distribution, Ultrasonic Imaging, 12, 171-188, 1990.
- [8] Retico, A., Delogu, P., Fantacci, M.E. and Kasae, P., An automatic system to discriminate malignant from benign massive lesions on mammograms. Nuclear Instruments and Methods in Physics Research, 596–600. 2006.
- [9] Wang, L. and Liu, J., Texture segmentation based on MRMRF modeling, Pattern Recognition Letters, 21, 189-200, 2000.
- [10] Chen, D.R., Chang, R.F., Chen, C.J., Ho, M.F., Kuo, S.J., Chen, S.T., Hung, S.J. and Moon, W.K., Classification of breast ultrasound images using fractal feature, Journal of Clinical Imaging, 29, 235–245, 2005.
- [11] Feleppa, E.J., Kalisz, A., Melgar, J.B.S., Lizzi, F.L., Liu, T., Rosado, A.L., Shao, M.C., Fair, W.R., Wang, Y., Cookson, M.S., Reuter, V.E. and Heston, W.D.W., Typing of prostate tissue by ultrasonic spectrum analysis. IEEE Transactions on Ultrasonics Ferroelectrics and Frequency Control, 3(4), 609–619, 1996.
- [12] Chiu, B., Freeman, G.H., Salama, M.M.A. and Fenster, A., Prostate segmentation algorithm using dyadic wavelet transform and discrete dynamic contour, Phys. in Med. & Biol., 49, 4943–4960, 2004.

- [13] Kurnaz M.N., Dokur Z. and Ölmez, T., An incremental neural network for tissue segmentation in ultrasound images, *Computer Methods and Programs in Biomedicine*, 85(3), 187–195, 2007.
- [14] Clarke, L.P., Velthuizen, R.P., Camacho, M.A., Heine, J.J., Vaidyanathan, M., Hall, L.O., Thatcher, R.W. and Silbiger, M.L., MRI segmentation: methods and applications, *Magnetic Resonance Imaging*, 3, 343-368, 1995.
- [15] Kaus, M., Warfield, S. K., Nabavi, A., Black, P. M., Jolesz, F. A. and Kikinis, R., Automated segmentation of MRI of brain tumors. *Radiology*, 218, 586–591, 2001.
- [16] Tahir M.A. and Bouridane A., An FPGA Based Coprocessor for Cancer Classification Using Nearest Neighbour Classifier, *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2006.
- [17] Özdemir A.T., Danışman K. ve Asyali M.H., FPGA Tabanlı Aritmi Sınıflandırıcı, 14.Biyomedikal Mühendisliği Ulusal Toplantısı (BIYOMUT 2009), İzmir, 20-24 Mayıs, 2009.
- [18] Rakvic, R.N., Ngo, H., Broussard, R.P. and Ives, R.W., Comparing an FPGA to a Cell for an Image Processing Application, *EURASIP Journal on Advances in Signal Processing* Volume 2010, 2010.
- [19] Huitzil, C.T. and Estrada, M.A., Real-time image processing with a compact FPGA-based systolic architecture, *Real-Time Imaging*, Volume 10, Issue 3, 177-187, 2004.
- [20] Kalomiros, J.A. and Lygouras, J., Design and evaluation of a hardware/software FPGA-based system for fast image processing, *Microprocessors and Microsystems*, Volume 32, Issue 2, 95-106, 2008.
- [21] Koo, J.J., Evans, A.C. and Gross, W.J., 3-D Brain MRI Tissue Classification on FPGAs, *IEEE Transactions on Image Processing*, Volume. 18, 12, 2009
- [22] Cloutier, J., Cosatto, E., Pigeon, Steven., Boyer, F.R. and Simard, P.Y., VIP: An FPGA-based Processor for Image Processing and Neural Networks, *5th International Conference on Microelectronics for Neural Networks and Fuzzy Systems*, IEEE, 330-336, 1996.
- [23] Sackinger, E. and Graf H., A system for high-speed pattern recognition and image analysis. In *Proceedings of the Fourth International Conference on Microelectronics for Neural Networks and Fuzzy Systems*, 364-371, 1994.

- [24] Liu, J., Li, B. and Liang, D., Design and Implementation of FPGA-Based Modified BKNN Classifier, IJCSNS International Journal of Computer Science and Network Security, Volume 7, No.3, 2007
- [25] Gazula, S. and Kabuka, M.R., Design of Supervised Classifiers Using Boolean Neural Networks, IEEE Trans. On Pattern Analysis and Machine Intelligence, Volume 17, No. 12, 1995
- [26] Ölmez, T. and Dokur, Z., Classification of heart sounds using an artificial neural network, Pattern Recognition Letters, 24, 617–629, 2003
- [27] Fix, E. and Hodges, J.L., Discriminatory analysis, nonparametric discrimination: Consistency properties. Technical Report 4, USAF School of Aviation Medicine, Randolph Field, Texas, 1951.
- [28] Alpaydın, E., Neural Models of Incremental Supervised and Unsupervised Learning, Lausanne, Thesis PhD, Ecole Polytechnique Federale De Lausanne, 1990.
- [29] Brown, S., An Overview of Technology, Architecture and CAD Tools for Programmable Logic Devices, IEEE Custom Integrated Circuits Conference, 1994.
- [30] Brown, S. and Rose J., Architecture of FPGAs and CPLDs: A Tutorial, Department of Electrical and Computer Engineering University of Toronto.
- [31] Kuon, I., Tessier, R. and Rose J., FPGA Architecture: Survey and Challenges, Foundations and Trends in Electronic Design Automation, Volume 2, 2, 135–253, 2007.
- [32] Max 9000 Programmable Logic Family,
<http://www.altera.com/literature/ds/m9000.pdf>, 2003
- [33] Getting Started; What are FPGAs,
<http://www.xilinx.com/company/gettingstarted/index.htm>
- [34] Maxfield, C., The Design Warrior's Guide to FPGAs, Mentor Graphics Corporation and Xilinx, Inc., 2004
- [35] Pedroni V.A., Circuit Design with VHDL, MIT press, London, 2004
- [36] K-Nearest Neighbor, http://www.scholarpedia.org/article/K-nearest_neighbor, 2009.
- [37] Dokur, Z. ve Ölmez, T., Uzaktan Algılama Görüntülerinin Bölütlenmesi, URSI-Türkiye'2002, İTÜ, 2002.

- [38] Jun, D., Na, L., A high-performance pseudo-random number generator based on FPGA, International Conference on Wireless Networks and Information Systems, IEEE, 2009
- [39] Shen, H., A high-speed and long period combined pseudo-random number generator, Second International Symposium on Computational Intelligence and Design, IEEE, 2009
- [40] Poorghanad, A., Generating High Quality Pseudo Random Number Using Evolutionary Methods, International Conference on Computational Intelligence and Security, IEEE, 2008
- [41] Moghadam, Z. I., Designing a Random Number Generator with Novel Parallel LFSR Substructure for Key Stream Ciphers, International Conference On Computer Design And Applications (ICCD 2010), IEEE, 2010.
- [42] Chen, J., Design of a Random Testing Circuit Based on LFSR for the External Memory Interface, Solid-State and Integrated Circuit Technology (ICSICT), 10th IEEE International Conference, 2010.