



**DONANIM TABANLI HIZLANDIRILMIŐ EVRİŐİMSEL SINİR AĐI İLE  
ULTRASON GÖRÜNTÜLERİNİN ÇAKIŐTIRILMASI**

**Seda GÜZEL AYDIN**

**DOKTORA TEZİ  
ELEKTRİK-ELEKTRONİK MÜHENDİSLİĐİ ANA BİLİM DALI**

**GAZİ ÜNİVERSİTESİ  
FEN BİLİMLERİ ENSTİTÜSÜ**

**AĐUSTOS 2023**

## ETİK BEYAN

Gazi Üniversitesi Fen Bilimleri Enstitüsü Tez Yazım Kurallarına uygun olarak hazırladığım bu tez çalışmasında;

- Tez içinde sunduğum verileri, bilgileri ve dokümanları akademik ve etik kurallar çerçevesinde elde ettiğimi,
- Tüm bilgi, belge, değerlendirme ve sonuçları bilimsel etik ve ahlak kurallarına uygun olarak sunduğumu,
- Tez çalışmasında yararlandığım eserlerin tümüne uygun atıfta bulunarak kaynak gösterdiğimi,
- Kullanılan verilerde herhangi bir değişiklik yapmadığımı,
- Bu tezde sunduğum çalışmanın özgün olduğunu,

bildirir, aksi bir durumda aleyhime doğabilecek tüm hak kayıplarını kabullendiğimi beyan ederim.

Seda GÜZEL AYDIN

09/08/2023

# DONANIM TABANLI HIZLANDIRILMIŞ EVRİŞİMSEL SİNİR AĞI İLE ULTRASON GÖRÜNTÜLERİNİN ÇAKIŞTIRILMASI

(Doktora Tezi)

Seda GÜZEL AYDIN

GAZİ ÜNİVERSİTESİ  
FEN BİLİMLERİ ENSTİTÜSÜ

Ağustos 2023

## ÖZET

Bu tez çalışmasında Ultrason-Ultrason (US-US) görüntü çiftleri arasındaki katı dönüşüm parametrelerinin tahmini için eğitilen Evrişimsel Sinir Ağı (ESA) çıkarım aşaması alanda programlanabilir kapı dizileri (FPGA) donanımı üzerinde gerçekleştirilmiştir. Çalışma iki aşamalı olarak gerçekleştirilmiştir. Birinci aşamada US-US B-mod görüntü çiftleri arasındaki katı dönüşüm parametrelerinin tahminini yapan ağı eğitimi yazılım tabanlı gerçekleştirilmiştir. Farklı hiper parametreler denenerek eğitimi tamamlanan ağı ağırlık parametreleri kaydedilmiştir. İkinci aşamada ESA modelini oluşturan evrişim, havuzlama, aktivasyon ve tam bağlantılı katmanlar teker teker donanım tabanlı tasarlanmıştır. Daha sonra yazılım tabanlı uygulamada kullanılan hiper parametre ve parametreler kullanılarak aynı model FPGA üzerinde çıkarım aşaması için katmanların birleştirilmesi ile gerçekleştirilmiştir. FPGA üzerinde tek duyarlıklı kayan noktalı sayı ve 32-, 28-, 20-, 18- ve 16-bit uzunluklarına sahip sabit noktalı sayı gösterim yöntemleri kullanılarak donanım kaynak kullanımı, hız ve hassasiyet parametreleri incelenmiştir. Hassasiyet parametresi yazılım tabanlı uygulamadan elde edilen değerlerin, donanım tabanlı elde edilen değerlerle ortalama kare hatası (Mean squared error, MSE), ortalama mutlak hata (mean absolute error, MAE), R-kare ve kök ortalama kare hatası (root mean squared error, RMSE) metriklerine göre değerlendirilmiştir. Bu değerlendirmeler sonunda en iyi hassasiyet, kaynak kullanımı ve hız dengesini sağlayan 20-bit sabit noktalı sayı gösterim yönteminin kullanılmasına karar verilmiştir. Katmanları ayrı ayrı tasarlanan ESA modelinde katmanlar arası ve katmanlar içi işlemlerin eş zamanlı yürütülmesi için eniyileme yöntemleri kullanılmıştır. Katmanlar arası işlemlerin kısmi eşzamanlı yapılabilmesi için veri akışı paralellliği, katman içlerinde ise döngü açma, döngü boru hattı ve dizi bölme eniyileme teknikleri kullanılmıştır. Uygulanan eniyileme tekniklerinden sonra FPGA tabanlı hızlandırılmış US-US B-mod görüntü çakıştırma parametrelerini tahmin eden ESA ağı (FUIR-CNN) ihmal edilebilecek bir miktarda hassasiyet kaybıyla beraber, yazılım tabanlı uygulamadan yaklaşık 139 kat daha hızlı sonuç ürettiği aynı zamanda da daha az güç tüketerek daha üstün bir performans sergilediği gösterilmiştir.

Bilim Kodu : 90520

Anahtar Kelimeler : Gömülü sistemler, derin öğrenme, ultrason, evrişimsel sinir ağları

Sayfa Adedi : 132

Danışman : Prof. Dr. Hasan Şakir BİLGE

REGISTRATION OF ULTRASOUND IMAGES USING HARDWARE BASED  
ACCELERATED CONVOLUTIONAL NEURAL NETWORK

(Ph. D. Thesis)

Seda GÜZEL AYDIN

GAZİ UNIVERSITY

GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES

August 2023

ABSTRACT

In this thesis, the extraction phase of the Convolutional Neural Network (CNN), which is trained for the estimation of rigid transformation parameters between Ultrasound-Ultrasound (US-US) image pairs, is implemented on the field programmable gate array (FPGA) hardware. The study was carried out in two stages. In the first stage, the training of the network, which estimates the rigid transformation parameters between the US-US B-mode image pairs, is carried out using software. The weight parameters of the network, whose training was completed by trying different hyper parameters, were recorded. In the second stage, The CNN model's convolution, pooling, activation, and fully connected layers are each uniquely designed based on hardware. Then, using the hyperparameters and parameters used in the software-based application, the same model was implemented on the FPGA by combining the layers for the extraction phase. Hardware resource utilization, speed and precision parameters were investigated by using single precision floating point number and fixed-point number representation methods with 32-, 28-, 20-, 18- and 16-bit lengths on the FPGA. The sensitivity parameter was evaluated according to the values obtained from the software-based application, with the values obtained from the hardware-based application, according to the MSE (Mean squared error), MAE (mean absolute error), R-squared and RMSE (root mean squared error) metrics. At the end of these evaluations, it was decided to use the 20-bit fixed-point number display method, which provides the best balance of precision, resource usage and speed. Optimization methods are used for simultaneous execution of operations between and within layers in the CNN model, in which layers are designed separately. Data flow parallelism is used to perform partial parallel operations between layers, while loop unrolling, loop pipelining and array partitioning optimization techniques are used within layers. After the applied optimization techniques, FPGA-based accelerated US-US image registration CNN network (FUIR-CNN) has been shown to exhibit superior performance by producing results approximately 139 times faster than the software-based application, with a negligible loss of precision, while consuming less power.

Science Code : 90520

Key Words : Embedded systems, deep learning, ultrasound, convolutional neural networks

Page Number : 132

Supervisor : Prof. Dr. Hasan Şakir BİLGE

## TEŞEKKÜR

Tez çalışmamın konusuyla ilgili önemli yönlendirmelerde bulunan, benden desteğini esirgemeyen ve bana örnek olan kıymetli danışman hocam Prof. Dr. Hasan Şakir BİLGE'ye, çalışmalarım sırasında beni destekleyerek araştırmaya yönelten değerli hocam Prof. Dr. Fırat HARDALAÇ'a, tez izleme komitesinde bulunan ve fikirleriyle bana yol gösteren kıymetli hocam Prof. Dr. Gökçe Nur YILMAZ'a çok teşekkür ederim.

Hayatım boyunca başardığım her şeyin temelinde ailemin sevgisi, desteği ve anlayışı yatmaktadır. Yoğun süren tez çalışmam süresinde bana katlanan, yanımda olan ve bana her koşulda güvenen aileme, özellikle sevgili anneme, kardeşlerime ve eşime teşekkürü borç bilirim. Doktora tezimi hazırlayıp bitirmemde en büyük paya sahip olan, arkamdaki en kıymetli ve en büyük destekçim, kızlarım için bir teyzeden daha fazlası olan canım kardeşim Sema GÜZEL'e ayrıca çok teşekkür ederim. Bu zorlu süreçte hayatıma yeni bir soluk katan, bana güç ve ilham kaynağı olan en kıymetli varlıklarım kızlarım Fatma Sena ve Ela'ya yürekten teşekkür ederim.

Bu doktora tezi Türkiye Bilimsel ve Teknolojik Araştırma Kurumu (TÜBİTAK) tarafından 122E393 numaralı 1002 Hızlı Destek Programı ile desteklenmiştir. Ayrıca bursiyer olarak görev aldığım 122E140 numaralı 1001 Bilimsel ve Teknolojik Araştırma Projelerini Destekleme Programının da doktora tezimin gerçekleşmesinde büyük katkısı olmuştur. Destek ve katkılarından dolayı TÜBİTAK'a çok teşekkür ederim.

## İÇİNDEKİLER

	<b>Sayfa</b>
ÖZET.....	iv
ABSTRACT .....	v
TEŞEKKÜR .....	vi
İÇİNDEKİLER .....	vii
ÇİZELGELERİN LİSTESİ.....	x
SİMGELER VE KISALTMALAR.....	xiv
1. GİRİŞ .....	1
2. LİTERATÜR ÇALIŞMASI.....	9
3. EVRİŞİMLİ SİNİR AĞLARI .....	15
4. ULTRASON GÖRÜNTÜLEME.....	23
4.1. Akustik Dalgaların Özellikleri .....	24
4.2. US Doku Etkileşimi.....	26
4.3. US Gri Tonlamalı Görüntüleme Modları.....	30
4.3.1. A mod .....	30
4.3.2. B mod .....	30
4.3.3. M mod.....	31
4.4. US Görüntü Kalitesi .....	31
4.4.1. Uzamsal çözünürlük .....	32
4.4.2. Zamansal çözünürlük (frame rate, çerçeve hızı).....	34
4.4.3. Kontrast çözünürlüğü .....	35
5. MEDİKAL GÖRÜNTÜ ÇAKIŞTIRMA .....	37
5.1. Medikal Görüntü Çakıştırma Sınıflandırması.....	38
5.2. Derin Öğrenme Tabanlı Medikal Görüntü Çakıştırma.....	40
5.3. Kullanılan Dönüşüm Yöntemi .....	43

	<b>Sayfa</b>
5.4. Medikal Görüntü Çakıştırma Değerlendirme Ölçütleri .....	44
5.5. Kullanılan Veri Setleri .....	46
6. FPGA .....	49
6.2. FPGA Programlama .....	54
6.2.1. Yüksek seviyeli sentezleme .....	54
7. FPGA TABANLI ESA GERÇEKLEMESİ .....	57
7.1. Kullanılacak Sayı Formatı .....	57
7.2. Tasarım Modeli .....	58
7.3. FPGA Tabanlı ESA Hızlandırma Yöntemleri .....	60
7.3.1. Veri alışverişi ve tampon bellek kullanımı .....	62
7.3.2. Verim optimizasyon yöntemleri .....	63
7.3.3. Paralellik yöntemleri .....	68
7.4. FPGA Tabanlı ESA Hızlandırıcı Modelinin Değerlendirme Ölçütleri .....	71
8. US-US KATI GÖRÜNTÜ ÇAKIŞTIRMA PARAMETRE TAHMİNİ İÇİN FPGA TABANLI HIZLANDIRILMIŞ ESA UYGULAMASI....	73
8.1. Yazılım Tabanlı Uygulama .....	75
8.2. Donanım Tabanlı Uygulamalar .....	78
8.2.1. Evrişim katmanına verilerin hazırlanması .....	79
8.2.2. Veri gösterim yöntemi ve veri gösterim bit uzunluğu .....	82
8.2.3. Hesaplama birimi iyileştirmeleri .....	90
8.3. Farklı Platformların Aynı Uygulama İçin Güç Tüketimlerinin Karşılaştırılması .....	99
9. DEĞERLENDİRME VE SONUÇLAR .....	103
KAYNAKLAR .....	107
EKLER .....	117
EK-1. Yazılım Tabanlı Uygulamada Farklı Hiper Parametreler İçin Yapılan Denemelere Ait Eğitim Doğrulama Grafikleri .....	118

**Sayfa**

EK-2. ESA Ađını Oluřturan Evriřim, Havuzlama ve Tam Bađlantılı Katmanların HLS Kullanılarak Tasarlanması, Dođrulanması, Modelsim RTL İzleme Dosyasının Oluřturulması ve IP Haline Getirilmesi.....	122
EK-3. RTL Blok Diyagramları .....	128
ÖZGEÇMİŐ .....	131

## ÇİZELGELERİN LİSTESİ

<b>Çizelge</b>	<b>Sayfa</b>
Çizelge 4.1. Tıbbi ultrasonla ilgili dokular ve malzemeler için yoğunluk, ses hızı ve akustik empedans.....	26
Çizelge 5.1. Görüntü dönüşüm yöntemleri ve matris formları.....	43
Çizelge 6.1. GPU, CPU ve FPGA'nın farklı parametrelere göre kıyaslanması .....	50
Çizelge 8.1. Eğitimi tamamlanan ağın hiper parametreleri.....	75
Çizelge 8.2. Uygulama sonuçlarının farklı değerlendirme metrikleriyle ölçülmesi.....	86
Çizelge 8.3. FPGA donanımlarında bulunan toplam kaynak miktarları.....	87
Çizelge 8.4. İki farklı FPGA platformlarının aynı uygulamalar için kaynak tüketimler ve gecikmeler açısından karşılaştırılması .....	88
Çizelge 8.5. Sıralı işlem yapan evrişim katmanlarında bir evrişim işlemi için kullanılan DSP kaynak miktarları .....	89
Çizelge 8.6. Hesaplama birimlerinde yapılan işlemlerin karşılaştırılması için kullanılan modeller .....	90
Çizelge 8.7. Karşılaştırma için kullanılan üç modelin gerçekleşmesinde kullanılan toplam kaynak miktarları ve gecikme miktarları.....	93
Çizelge 8.8. Yazılım donanım hız karşılaştırılması .....	95
Çizelge 8.9. Literatürde bulunan bazı çalışmaların kıyaslanması .....	95

## ŞEKİLLERİN LİSTESİ

Şekil	Sayfa
Şekil 3.1. Çok boyutlu bir görüntü için evrişim işlemi.....	17
Şekil 3.2. Maksimum ve ortalama havuzlama işlemi.....	18
Şekil 3.3. Tam bağlantılı katman.....	19
Şekil 3.4. Aktivasyon fonksiyonları .....	20
Şekil 3.5. Seyreltme yönteminin eğitim ve doğrulama kayıp grafiği üzerindeki etkisi, seyreltme (a) kullanılan ve (b) kullanılmayan ağı kayıp grafikleri	21
Şekil 4.1. US görüntüleme aşamaları .....	23
Şekil 4.2. Ses dalgalarının ortamda ilerlemesi ve sinüs dalga ile karakterize edilmesi .....	24
Şekil 4.3. Akustik dalgalarda yansıma ve kırılma olayı .....	27
Şekil 4.4. Akustik dalgaların farklı ortamlarda yansımaya ve saçılmaya uğraması....	29
Şekil 4.5. US görüntülerinde çözünürlük.....	31
Şekil 4.6. Dönüştürücü ve üretilen darbelerin özellikleri (a) düşük frekanslı dönüştürücü ve uzun SPL, (b) yüksek frekanslı dönüştürücü ve kısa SPL.	33
Şekil 4.7. US görüntüleme eksenel ve yanal çözünürlükler.....	34
Şekil 4.8. US görüntüleme çözünürlükleri ve bu çözünürlükleri etkileyen faktörler	36
Şekil 5.1. Görüntü karşılaştırma işlemi .....	38
Şekil 5.2. Medikal görüntü karşılaştırma sınıflandırılması .....	39
Şekil 5.3 Görüntü karşılaştırma için geometrik dönüşüm işlemleri.....	40
Şekil 5.4. Danışmanlı öğrenme yöntemi kullanılarak yapılan görüntü karşılaştırma modelinin genel yapısı .....	42
Şekil 5.5. Danışmansız öğrenme yöntemi kullanılarak yapılan görüntü karşılaştırma modelinin genel yapısı .....	43
Şekil 5.6. Veri setlerinden örnek görüntüler .....	47
Şekil 5.7. Veri artırma ve katı karşılaştırma uygulaması için yeni veri seti oluşturma...	48
Şekil 6.1. GPU, FPGA ve CPU nun CNN/RNN gerçeklemede kullanılan güç/verim grafiği.....	52
Şekil 6.2. Zedboard kartının genel görüntüsü .....	53

<b>Şekil</b>	<b>Sayfa</b>
Şekil 6.3. Xilinx FPGA ları için HLS akış tasarımı .....	55
Şekil 7.1. Donanım tabanlı ESA yöntemleri (a) akış tasarımı, (b) tek hesaplama birimi .....	59
Şekil 7.2. Evrişim işlemi sözde kodu .....	61
Şekil 7.3. Raster tarama düzeni .....	62
Şekil 7.4. Çok boyutlu özellik haritası için raster tarama düzeni.....	63
Şekil 7.5. Boru hattı eniyileme yönteminin döngü üzerindeki etkisi .....	64
Şekil 7.6. Döngü açma eniyileme yönteminin döngü üzerindeki etkisi .....	65
Şekil 7.7. HLS tabanlı dizi bölme yöntemleri.....	66
Şekil 7.8. Veri akış paralelliğinin katmanlara uygulanması .....	67
Şekil 7.9. Eniyileme yapılmadan önce evrişim işlemi ve karşılık gelen donanım yapısı .....	68
Şekil 7.10. L2 döngüsünün açılması ve karşılık gelen donanım yapısı.....	69
Şekil 7.11. L3 döngüsünün açılması ve karşılık gelen donanım yapısı.....	69
Şekil 7.12. L4 döngüsünün açılması ve karşılık gelen donanım yapısı.....	70
Şekil 7.13. L3 ve L4 döngülerinin beraber açılması ve karşılık gelen donanım yapısı	70
Şekil 7.14. L2, L3 ve L4 döngülerinin beraber açılması ve karşılık gelen donanım yapısı .....	71
Şekil 8.1. Uygulamanın genel yapısı .....	74
Şekil 8.2. Rot çıkış katmanının eğitim ve doğrulama kayıp grafiği .....	76
Şekil 8.3. dx çıkış katmanının eğitim ve doğrulama kayıp grafiği.....	77
Şekil 8.4. dy çıkış katmanının eğitim ve doğrulama kayıp grafiği.....	77
Şekil 8.5. Ağın eğitim ve doğrulama kayıp grafiği .....	78
Şekil 8.6. Satır ve pencere tampon bellek birimleri .....	80
Şekil 8.7. Donanım tabanlı gerçekleştirme genel yapısı.....	81
Şekil 8.8. Test görüntüleri rot çıkış katmanı için yazılım ve donanım tahmin sonuçları .....	83

<b>Şekil</b>	<b>Sayfa</b>
Şekil 8.9. Test görüntüleri dx çıkış katmanı için yazılım ve donanım tahmin sonuçları .....	84
Şekil 8.10. Test görüntüleri dy çıkış katmanı için yazılım ve donanım tahmin sonuçları .....	85
Şekil 8.11. Farklı sayı gösterim yöntemlerine göre hesaplanan MAE .....	87
Şekil 8.12. Farklı sayı gösterim yöntemlerine göre kullanılan BRAM ve DSP kaynak miktarları .....	89
Şekil 8.13. Eşzamanlı Hesaplama birimi .....	91
Şekil 8.14. Karşılaştırma için kullanılan üç modelin katmanlara göre kullandıkları DSP kaynak miktarları .....	92
Şekil 8.15. Karşılaştırma için kullanılan üç modelin katmanlara göre hesaplanan gecikme miktarları (döngü sayısı olarak).....	94
Şekil 8.16. Yazılım tabanlı ve donanım tabanlı oluşturulan karşılaştırılmış görüntüler...	96
Şekil 8.17. RTL doğrulama izleme dosyası .....	98
Şekil 8.18. VPE (a) Model 2 ve (b) FUIR-CNN için KCU hedef donanımı için yapılan güç tüketimi ölçüsü .....	100
Şekil 8.19. ZedBorad FPGA geliştirme kartı-dizüstü bilgisayar bağlantısı ve UT230B-EU güç ölçüm cihazı kullanılarak FPGA için yapılan güç ölçümü.....	101
Şekil 8.20. Yazılım çıkarım aşamasında dizüstü bilgisayar için yapılan güç ölçümü ..	102
Şekil 8.21. CPU ve GPU güç ölçümleri (a) GPU'nun görünümü, (b) çıkarım aşamasında GPU kullanılarak yapılan güç ölçümü, (c) i5 işlemcili CPU kullanılarak yapılan güç ölçümü.....	102

## SİMGELER VE KISALTMALAR

Bu çalışmada kullanılmış simgeler ve kısaltmalar, açıklamaları ile birlikte aşağıda sunulmuştur.

<b>Simgeler</b>	<b>Açıklamalar</b>
<b>c</b>	Ultrasonik dalgası ilerleme hızı
<b>cm</b>	Santimetre
<b>db</b>	Desibel
<b>GOPs</b>	Saniyede giga işlem oranı (Giga operation per second)
<b>f</b>	Frekans
<b>Hz</b>	Hertz
<b>kHz</b>	Kilo hertz
<b>MHz</b>	Mega hertz
<b>ms</b>	Milisaniye
<b>Rayl</b>	Akustik direnç
<b>RF</b>	Radyo frekans
<b>s</b>	Saniye
<b>Z</b>	Akustik empedans
<b>B</b>	Bulk modülü
<b><math>\theta</math></b>	Açı
<b><math>\lambda</math></b>	Dalga boyu
<b><math>\rho</math></b>	Ortam yoğunluğu
<b>Kısaltmalar</b>	<b>Açıklamalar</b>
<b>ASIC</b>	Uygulamaya özel tümleşik devreler (Application specific integrated circuits)
<b>B-Mod</b>	Parlaklık modu (Brightness mode)

<b>Kısaltmalar</b>	<b>Açıklamalar</b>
<b>BRAM</b>	Blok rastgele erişim belleği (Block random access memory)
<b>CPU</b>	Merkezi işlem birimi (Central processing unit)
<b>CT</b>	Bilgisayarlı tomografi (Computed tomography)
<b>DOF</b>	Serbestlik derecesi (Degree of freedom)
<b>DÖ</b>	Derin öğrenme
<b>DRAM</b>	Dinamik rastgele erişim belleği (Dynamic random access memory)
<b>DSE</b>	Tasarım alanı keşfi (Design space exploration)
<b>DSP</b>	Sayısal sinyal işleme (Digital signal processing)
<b>DTF</b>	Darbe tekrarlama frekansı (Pulse repetition frequency)
<b>ESA</b>	Evrişimli sinir ağları
<b>FF</b>	Flip-Flop
<b>FIFO</b>	İlk giren ilk çıkar (First in first out)
<b>FM</b>	Özellik haritası (Feature map)
<b>FP</b>	Kayan noktalı (Floating point)
<b>FPGA</b>	Alanda programlanabilir kapı dizileri (Field programmable gate arrays)
<b>FXP</b>	Sabit noktalı (Fixed point)
<b>GÇ</b>	Görüntü çakıştırma
<b>GPU</b>	Grafik işlemci birimleri (Graphics processing unit)
<b>HLS</b>	Yüksek seviye sentezleme (High level synthesis)
<b>ICP</b>	Yinelemeli en yakın nokta (Iterative closest point)
<b>IFM</b>	Giriş özellik haritası (Input feature map)
<b>II</b>	Başlatma aralığı (Initiation interval)
<b>LR</b>	Öğrenme oranı (Learning rate)
<b>LUT</b>	Arama çizelgeleri (Look up table)

<b>Kısaltmalar</b>	<b>Açıklamalar</b>
<b>MAE</b>	Ortalama mutlak hata (Mean absolute error)
<b>MAC</b>	Çarpma-biriktirme (Multiply-accumulator)
<b>MAD</b>	Çarpma-toplama (Multiply-add)
<b>MI</b>	Ortak bilgi (Mutual information)
<b>MR</b>	Manyetik rezonans
<b>MSE</b>	Ortalama kare hatası (Mean squared error)
<b>NCC</b>	Normalize edilmiş çapraz korelasyon (Normalized cross correlation)
<b>NMI</b>	Normalize edilmiş ortak bilgi (Normalized mutual information)
<b>OFM</b>	Çıkış özellik haritası (Output feature map)
<b>PET</b>	Pozitron emisyon tomografisi (Positron emission tomography)
<b>PLC</b>	Programlanabilir mantıksal denetleyiciler (Programmable logic controller)
<b>PSNR</b>	Tepe sinyal gürültü oranı (Peak signal to noise ratio)
<b>RBM</b>	Boltzmann makineleri (Restricted boltzmann machines)
<b>RAM</b>	Rastgele erişim belleği (Random access memory)
<b>RMSE</b>	Kök ortalama kare hatası (Root mean squared error)
<b>ROM</b>	Salt okunur bellek (Read only memory)
<b>RTL</b>	Kayıt düzeyinde transfer (Register transfer level)
<b>SPL</b>	Uzamsal darbe uzunluğu (Spatial pulse length)
<b>SRAM</b>	Durağan Rastgele Erişimli Bellek (Static random access memory)
<b>SSD</b>	Farkın karelerinin toplamı (Sum of squared differences)
<b>SSIM</b>	Yapısal benzerlik ölçütü (Structural similarity)
<b>TRE</b>	Hedef çakıştırma hatası (Target registration error)

<b>Kısaltmalar</b>	<b>Açıklamalar</b>
<b>US</b>	Ultrason
<b>VHDL</b>	Yüksek hızlı tümleşik devreler için donanım tanımlama dili (Very high speed integrated circuit hardware description language)
<b>VPE</b>	Vivado güç tüketimi tahmini (Vivado power estimation)
<b>2B</b>	İki boyutlu
<b>3B</b>	Üç boyutlu



## 1. GİRİŞ

Son birkaç yılda, derin öğrenme (DÖ) yöntemleri özellikle görüntü tanıma, sınıflandırma ve nesne tespiti gibi bilgisayarla görme alanında çok iyi performans sergileyen ve makine öğrenmesinde altın standart olarak kabul edilen yöntemlerinden bir tanesidir [1]. Klasik sinir ağlarından türetilen DÖ yöntemleri birçok alanda temel aldığı yapılardan daha üstün performanslar sergilemektedir [2]. Makine öğrenimi alanında giderek yaygın kullanılan hesaplama yaklaşımı haline gelen DÖ yöntemleri birkaç karmaşık bilişsel görevde, insan performansı tarafından sağlananlarla eşleşen ve bu performansları geride bırakan sonuçlar elde ettiği yapılan çalışmalarda ifade edilmektedir [1]. Büyük miktarda veri öğrenme yeteneğine sahip olan DÖ yöntemlerinin uygulama alanının siber güvenlik [3], doğal dil işleme [4], biyoinformatik [5], robotik ve kontrol [6], biyomedikal bilgi işleme [7] gibi birçok alan başta olmak üzere hızla bir şekilde genişlediği görülmektedir.

Klasik makine öğrenmesi yöntemlerinin başarısı büyük ölçüde girdi verilerinin iyi temsil edilmesine bağlıdır. Uygun verilerle beslenen bir makine öğrenmesi yöntemi, zayıf verilerle beslenen yöntemlerle kıyaslandığında daha üstün performanslar sergilediği gösterilmiştir [2]. Bu nedenle, makine öğrenimi araştırmaları, girdi olarak verilen verilerden uygun özelliklerin çıkarılmasına kaymıştır. Ham verilerden özniteliklerin elde edilmesini amaçlayan çalışmaların en önemli zorluğu, verilerin uygun bir şekilde temsil edilmesi için çoğunlukla uzman bilgilerine ihtiyaç olması, bazı durumlarda alana özgü olması ve büyük ölçüde insan çabasına bağlı olmasıdır. DÖ yöntemleri sayesinde öznitelik çıkarma işlemi büyük ölçüde ortadan kalkmıştır. Bu yöntemler verilerden öznitelikleri otomatik olarak çıkarmakta, böylece uygun veri temsili için bahsedilen dezavantajları önemli oranda azaltmaktadır.

DÖ algoritmaları çok katmanlı bir veri temsil mimarisine sahiptir. Katmanların derinliği arttıkça çıkarılan özniteliklerin de soyutluk seviyesi artmaktadır. İlk katmanlarda düşük seviye özellikleri çıkarılmakta iken son katmanlara doğru yüksek seviye özellikleri çıkarılmaktadır. Böylece farklı öznitelikler çıkartılarak otomatik olarak veri temsili sunulmaktadır [1].

DÖ yaklaşımında kullanan en önemli modeller; otomatik kodlayıcı (autoencoder) [8], kısıtlı Boltzmann makineleri (restricted Boltzmann machines, RBM) [9] ve Evrimsel Sinir

Ağlarıdır (ESA) [10]. Kodlayıcı ve kod çözücü (encoder-decoder) olarak adlandırılan iki kısımdan oluşan otomatik kodlayıcılar, girdi verilerini otomatik olarak kodlara dönüştürüp sıkıştırmak ve ardından yeniden oluşturulan girdinin orijinaline olabildiğinde benzer şekilde az kayıpla tekrar üretmek için tasarlanan sinir ağları türüdür. Denetimsiz öğrenme için kullanılan ve Boltzmann makinelerinin özel bir türü olan RBM, bir dizi girdi verisi üzerinden bir olasılık dağılımını öğrenebilen, görünür ve gizli katmanlar (visible and hidden) olarak adlandırılan iki katmandan oluşan kısıtlı bir sinir ağı türüdür.

Son zamanlarda yoğun bir şekilde kullanılmaya başlanan ESA modeli, DÖ yöntemi kullanan ağlar içerisinde en popüler olanlardan bir tanesidir. Nesne sınıflandırma, nesne takibi [11, 12], borsa fiyat tahmini [13], görüntü bölütleme [14, 15], medikal görüntü çakıştırma [16], dil işleme [17] gibi birçok alana uygulanmış ve başarılı performans sergilediği ifade edilmiştir. ESA modeli, her nöronun öğrenilebilir bir ağırlığı (weights) ve yanlılığı (bias) olan nöronlardan oluşmaktadır. Temel olarak giriş katmanı, gizli katmanlar ve çıkış katmanından oluşmaktadır. Gizli katmanlar genellikle evrişim katmanları, havuzlama katmanları, aktivasyon katmanları ve tam bağlantılı katmanlardan oluşmaktadır. Evrişim katmanında evrişim işlemi kullanılarak katmana gelen verilerden öznitelikler çıkartılır. Havuzlama işleminde çıkarılan öznitelik haritalarının boyutu azaltılarak işlem yükleri düşürülmeye çalışılır. Tam bağlantılı katmanlar ise klasik sinir ağlarındaki gibi her nöronu kendisinden sonraki tüm nöronlarla bağlar. Adını katmanlarından bir tanesi olan evrişim katmanlarında yapılan evrişim işleminden alan ESA farklı uygulamalara uyarlanabilen farklı kombinasyonlara sahiptirler. Sınıflandırma problemi için tasarlanan, evrişim, havuzlama ve tam bağlantılı katmanlara sahip olan LeNet-5 ile trend haline gelen ESA kullanma süreci [18], şimdilerde yığın evrişim modülleri [19], ters evrişim [20], ileri atlamalı bağlantılara sahip katmanlar gibi daha karmaşık modellerle devam etmektedir.

DÖ yöntemleri her ne kadar klasik makine öğrenmesi yöntemlerine karşı başarılı performanslar sergilese de bazı dezavantajları bulunmaktadır. Bunlardan ilki bu yapıları beslemek için çok sayıda veriye ihtiyaç duyulmasıdır. Diğer bir tanesi çok fazla işlem yapılması gerektiği için genel amaçlı işlemciler ile bu yapıların uygulanmasının zor olmasıdır. Bahsedilen ikinci dezavantajdan dolayı, DÖ yapılarına özel donanım ihtiyacı doğmuştur. DÖ uygulamaları için özel donanım kullanımının artmasıyla beraber ortaya çıkan önemli bir problem, tüketilen enerji miktarının artmasıdır.

Literatür incelendiğinde DÖ üzerine yapılan çalışmaların büyük çoğunluğu, problemin çözümlerinin eniyilemesine yönelik araştırmaları içermektedir. Fakat giderek karmaşıklaşan ve hesaplama yükü artan ESA uygulamaları genel amaçlı işlemciler üzerinde hesaplama yükünü de artırdığı için bu yapıların uygulanması sorun haline gelmiştir. Ayrıca, Merkezi İşlem Birimi (Central Processing Unit, CPU) genel amaçlı işlemler için tasarlanmış olup, işlemleri sıralı yürütmektedir. ESA yapılarında katmanlar arası ve katmanlar içinde yapılan işlemlerin kısmî veya tam olarak paralel bir şekilde yürütülmesi performansı artıracaktır. CPU'ların sıralı işlem yapma özelliğinden dolayı, CPU tabanlı yapılan ESA uygulamalarında, ulaşılabilecek maksimum işlem hızına ulaşamamaktadır ve performans yetersiz kalmaktadır [21]. Bundan dolayı, ESA mimarisine özel ve uyarlanmış donanım destek birimlerine ihtiyaç vardır.

İlk etapta, endüstri ve akademik çevrelerdeki araştırmacılar tarafından ESA uygulamaları için Grafik İşlemci Birimleri (Graphics Processing Unit, GPU) tercih edilmekteydi. Günümüzde büyük ve derin ağlar, GPU'lar üzerinde başarılı ve hızlı bir şekilde uygulanabilmektedir. GPU donanımları veri hesaplama için optimize edilmiş yapılar olup büyük ölçekli matris hesaplamalarının hızlı yapılması için tasarlanmışlardır. GPU'ların DÖ yöntemlerinin araştırılıp geliştirilmesi alanındaki katkısı çok büyüktür. Özellikle görüntü, metin ve video gibi büyük miktarlarda veriler üzerinde paralel hesaplama gerçekleştiren makine öğrenimi uygulamalarında yüksek performans sunmaktadır. Fakat ağlardaki katman sayısının artması, beraberinde donanımların işlem yükünü artırmış ve tüketilen güç miktarının artmasına neden olmuştur. GPU'lar güç tüketimi konusunda istenen sonuçları yeterince verememektedir [10, 22-24]. Günümüzde, tabletlerde hatta cep telefonlarında bile DÖ algoritmalarının kullanılmasıyla beraber güç tüketimi göz ardı edilemeyecek bir sorun haline gelmiştir [25]. Bunun üzerine, araştırmacıların ilgisi, kullanılan donanımların hızlandırılmasına ve minimum güç tüketimi için yapılması gereken araştırmalara yönelmiştir [26].

Son zamanlarda en çok kullanılan gömülü sistemler içinde Alanda Programlanabilir Kapı Dizileri (Field Programmable Gate Arrays, FPGA) teknolojisi öne çıkmaktadır [27-31]. Literatür incelendiğinde yapılan çalışmalar, GPU güç tüketimi konusunda FPGA'ların gerisinde kaldığını göstermektedir [32]. Bundan dolayı, ESA yapılarının donanım tabanlı gerçekleştirilmesi ile ilgili araştırmalar GPU kullanımından FPGA kullanımına yönelmiştir [33-35].

Donanım yapısı kullanıcı tarafından değiştirilebilen entegre devreler olan FPGA'lar üretimden sonra istenen fonksiyona göre uyarlanabilmektedir [36, 37]. İşlemlerin paralel yapılabilmesi, yapısının üretimden sonra değiştirebilmesi, tekrar tekrar programlanabilmesi, güç tüketiminin az olması gibi avantajlarından dolayı, savunma sanayi, endüstriyel alan, pazarlama gibi birçok alanda kullanımı giderek artmaktadır [38-41]. Derin öğrenme uygulamalarının cep telefonları, tabletler gibi günlük sürekli kullanılan elektronik aletlerde kullanılmasıyla, bu aletlerin enerji tüketimleri önemli bir konu haline gelmiştir [42]. Düşük güç tüketimleri sayesinde, FPGA tabanlı yapılacak olan çalışmalarla bu sorunun önemli ölçüde çözülebileceği öngörülmektedir [43, 44]. Ayrıca eş zamanlı işlem yapabilme yeteneği sayesinde FPGA platformu özellikle hızlı işlem yapılması gereken uygulamalar için önemli katkı sağlayacaktır. FPGA'ların belirli bir göreve optimize edilebilen ve programlanabilen yapısı sayesinde, ESA modellerinin özel olarak tasarlanıp donanım seviyesinde verimli bir şekilde çalıştırılmasına imkân sağlanmaktadır. GPU'lar ise genel amaçlı grafik işlemciler olarak tasarlanmış olup, geniş bir uygulama yelpazesine yönelik esneklik sunan yapılardır. Ayrıca FPGA'lar, belirli işlemleri optimize edilmiş donanım düzeyinde gerçekleştirdikleri için çoğunlukla daha düşük güç tüketimiyle çalışmaktadırlar. FPGA'ların özel tasarımlara uygun yapısı ve düşük güç tüketimi genel amaçlı işlemci olarak tasarlanan ve güç tüketimleri yüksek olan GPU'lara karşı avantaj sağlamaktadır.

Ancak, belirtilen avantajların yanında bazı dezavantajlar da bulunmaktadır. Özellikle FPGA programlama, tasarım ve yapılandırma süreçlerinin GPU platformlarına göre daha karmaşık olması ve özelleştirilmiş çözüm gerektirmesi FPGA'larda karşılaşılan başlıca problemlerdir. Ayrıca, GPU'lar, geniş bir programlama ve yazılım çeşitliliğine olanak tanıdıkları için çeşitli uygulamalarda daha rahatlıkla kullanılabilirler.

Bu çalışmanın amacı, ESA modelinin FPGA üzerinde gerçekleştirilmesinin yapılması ve uygulama problemi olarak Ultrason-Ultrason (US-US) katı görüntü çakıştırma parametrelerinin tahmininin FPGA tabanlı hızlandırılmış ESA modelleriyle daha az güç tüketerek yapılmasıdır.

Teşhis amaçlı kullanılan görüntüleme sistemleri, zamanında teşhis, hastalıkların evrenmesi, tedavi seçimi, planlama, rehberlik ve takip için sağlık hizmetlerinde kritik role sahiptir. Tanısal görüntüleme modaliteleri arasında olan US görüntüleme sistemi gerçek

zamanlı görüntüleme olanağı sunmasından dolayı yaygın olarak kullanılan modalitedir. Manyetik rezonans (MR), Bilgisayarlı tomografi (Computed Tomography, CT) gibi modalitelere kıyasla taşınabilir olması, gerçek zamanlı olması ve daha az maliyetli olması nedeniyle, özellikle kırsal kesimlere taşınabilmesi ve buralarda gerçek zamanlı görüntüleme için kullanılabilmesi, US modalitesini eşsiz kılmaktadır. Ultrasonografi, kardiyolojiden onkolojiye kadar birçok tıbbi uzmanlık alanında giderek daha fazla kullanılmaktadır ve pazar payı küresel olarak artmaktadır [45]. Özellikle gelişmekte olan ülkeler için US tanı amacıyla en sık kullanılan modalitelerden olduğu için, bu modalite üzerinde yapılan çalışmalar büyük önem arz etmektedir [46, 47]. Gerçek zamanlı görüntü oluşturması avantajından dolayı, cerrahi operasyonlar esnasında takip sistemi olarak kullanılmaktadır [46, 47]. Cerrahi müdahaleden önce alınmış MR, CT veya US görüntüleriyle cerrahi müdahale sırasında alınan US görüntülerinin üst üste çakıştırılması önemli bir problemdir [48]. Çakıştırma işlemi birçok uygulamada doğrudan veya dolaylı olarak kullanılmaktadır. Ultrason panoramik görüntü oluşturma, 3D görüntü oluşturma, US görüntülerinde çözünürlüğün artırılması gibi birçok problemde kullanılmaktadır [49-52]. Bu tür uygulamalarda çakıştırma hassasiyeti kadar hız da önemli bir parametredir.

Bu çalışmada, hem ESA modeli FPGA üzerinde gerçekleştirilmiş hem de yazılım uygulanasından daha hızlı cevap üreten ve daha az güç tüketen hızlı ve yüksek tahmin yeteneğine sahip US-US B-mod görüntü çakıştırma uygulaması FPGA üzerinde gerçekleştirilmiştir.

Bu çalışmadaki önemli aşamalar ve çalışmanın bilimsel katkıları şunlardır:

- US-US görüntü çiftleri arasındaki katı dönüşüm parametrelerinin tahmini için farklı hiper parametreler denenerek, katman sayısı az, tahmin oranı yüksek ESA ağının eğitimi ve doğrulaması yazılım tabanlı gerçekleştirilmiştir. Eğitimi tamamlanan ağ sadece evrişim ve havuzlama katmanlarından oluşmaktadır. Çıkış katmanları dışında tam bağlantılı katmanlar kullanılmayarak hesaplamalarda kullanılacak parametre sayısı azaltılmıştır.
- ESA modelinin evrişim, havuzlama ve tam bağlantılı katmanları FPGA üzerinde ayrı ayrı kodlanarak tasarlanmıştır. Bu katmanların doğrulukları hem C düzeyinde hem de C/RTL co-simulasyon düzeyinde doğrulanmıştır.

- FPGA üzerinde yapılacak ESA tasarımı için akış modeli kullanılmıştır. Böylece her katman için ayrı ayrı eniyileme yöntemleri uygulanabilmiştir.
- Tasarlanan katmanlar yazılım aşamasında belirlenen hiper parametrelere göre düzenlenerek arka arkaya oluşturulmuştur. Yazılım tabanlı ESA ağının eğitimi tamamlandıktan sonra dosyada saklanan ağırlık ve bias değerleri FPGA üzerinde yapılacak uygulama için uygun formata getirilmiştir.
- Donanım için tasarlanan ESA ağının gerçekleştirilmesi için kullanılacak sayı gösterim formatı ve uzunluğu, donanım kaynak kullanımı, hassasiyet ve hız parametrelerinin incelenmesi sonucu belirlenmiştir.
- Kullanılacak sayı gösterim formatı ve uzunluğu belirlendikten sonra, hesaplama birimlerinde işlemlerin eş zamanlı gerçekleştirilmesi için döngü açma (loop unrolling), veri akış boru hattı (dataflow), boru hattı (pipelining) ve dizi bölme (array partitioning) eniyileme yöntemleri kullanılmıştır.
- Katmanlar arası, katmalar içi ve ağ düzeyinde yapılan optimizasyonlar neticesinde, donanım tabanlı hızlandırılmış US-US katı dönüşüm parametre tahmini yapan ESA modeli yazılım tabanlı modelle kıyaslandığında, ihmal edilebilecek tahmin hassasiyet kaybına karşılık 139 kat daha hızlı sonuç ürettiği ve aynı zamanda daha az güç tükettiği gözlenmiştir.

### Araştırmanın önemi

Yapılan bu tez çalışması yazılım ve donanım aşaması olmak üzere iki aşamadan oluşmaktadır. Bu nedenle, bu tez çalışması medikal görüntü çakıştırma problemine hem yazılımsal bir çözüm sunması açısından, US görüntüleme sistemlerinin dezavantajı olan düşük çözünürlük probleminin iyileştirilmesine imkân sağlayacaktır, hem de donanımsal uyarlamaları teşvik edecek ve donanım seviyesinde hızlandırma uygulamalarına imkân sağlayacaktır. Görüntü çakıştırma uygulaması medikal alan [53] başta olmak üzere, panoramik görüntü oluşturma [54], uydu görüntülerinin çakıştırılması [55, 56], uzaktan kontrol uygulamaları [57, 58], güneş panellerinin izlenmesi [59], askeri alanda hedef tespitleri [60] gibi birçok farklı alanda kullanılmaktadır. Bunun yanı sıra ESA yapıları da birçok alanda farklı problemlerin çözümü için kullanılmaktadır. Bu tez çalışmasında ESA yapısını oluşturan katmanlar donanım üzerinde teker teker tasarlandığı için, bu temel yapıları kullanan farklı uygulamalar için donanım tabanlı gerçekleştirme yapmak mümkündür. Yazılım üzerinde farklı problemler için eğitildikten sonra hiper parametre ve parametreleri belirlenen

ağ, donanım kaynaklarını aşmayacak şekilde ESA'yı oluşturan evrişim, havuzlama ve tam bağlantılı katmanlar çoğaltılarak kullanılabilir. Bu da yapılan bu çalışmanın ölçeklenebilirliğini ve genişletilebilirliğini göstermektedir. Tez çalışmasında yapılan ESA yapılarının FPGA'ya uyarlanıp hızlandırılması sadece medikal alanda değil, savunma ve sivil endüstri gibi birçok alanda kullanılabilir olacaktır.

### Varsayımlar / Sayıtlar

FPGA tabanlı yapılan ESA hızlandırma uygulamalarında katman sayısının artması hatanın da artmasına neden olduğu varsayılmaktadır. Çünkü hata her katmanda katlanarak artmaktadır. Bundan dolayı katman sayısı hatanın artmasını engellemek amacıyla da düşük tutulmuştur.

### Sınırlılıklar

FPGA gerçeklemede 32-bit tek duyarlılık kayan noktalı (single precision floating point, FP) sayı gösterim sisteminin kullanılması, gerçekleştirme sırasında kullanılacak donanım kaynak sayısını artırmaktadır. Bunun yanında, hızlandırma amacıyla işlemlerin eşzamanlı yapılmasından dolayı da kullanılacak kaynak sayısı artmaktadır. Katman sayısının artması, ağın genişliğinin artması da kullanılacak donanım kaynak sayısının artmasına neden olacaktır. FPGA platformunun en büyük dezavantajlarından biri sınırlı donanım kaynaklarına sahip olmasıdır. ESA uygulamasının gerçek zamanlı yapılması için kullanılan ZedBoard FPGA geliştirme kartında sınırlı sayıda donanım kaynakları bulunmaktadır. Bundan dolayı, kullanılan FPGA kartı daha geniş bir ağın gerçekleştirilmesi için yetersiz kalmıştır ve çalışmayı sınırlandırmıştır. Ayrıca web ortamında herkese açık erişime sahip olan görüntü veri setleri genellikle görüntü sınıflandırma veya görüntü bölütleme gibi problemler için oluşturulan verileri içermektedir. Fakat bu tez çalışmasının yazılım aşamasında US görüntülerin katı çakıştırma parametrelerinin tahmini yapıldığı için, bu probleme özgü veri seti bilgimiz dahilinde web ortamında bulunamamıştır. Görüntülerin azlığı ve görüntü çakıştırma için yeterli veri setinin olmaması da yapılan bu çalışmayı sınırlandıran bir faktör olmuştur.

Bu tez çalışmasının geri kalan bölümleri şu şekilde düzenlenmiştir. İkinci bölümde yapılan literatür taraması özetlenmiştir. Literatür taraması hem FPGA tabanlı ESA uygulamaları

hem de medikal görüntü çakıştırma uygulamaları için ayrı ayrı taranmıştır. Üçüncü bölümde, ESA yapılarından kısaca bahsedilmiştir. Dördüncü bölümde ultrason görüntülemenin temelleri incelenerek, sesin genel özellikleri, ultrason görüntüleme modları ve ultrason görüntüleme elemanlarından bahsedilmiştir. Beşinci bölümde medikal görüntü çakıştırma sınıflandırılması ve genel yapısı ele alınmıştır. Altıncı bölümde, FPGA donanımı hakkında bilgiler verilmiştir. Yedinci bölüde, FPGA tabanlı ESA uygulamalarının genel hatları incelenmiştir. Ayrıca FPGA tabanlı ESA hızlandırma yöntemleri de bu bölümde ele alınmıştır. Sekizinci bölümde, US-US katı görüntü çakıştırma ağı yazılım ve donanım üzerinde gerçekleştirme uygulamaları yapılmıştır. Son bölüm olan dokuzuncu bölümde ise, uygulamaların değerlendirilmesi ve sonuçlar tartışılarak tez çalışması tamamlanmıştır.

## 2. LİTERATÜR ÇALIŞMASI

Bu çalışmada literatür taraması üç aşamalı yapılmıştır. İlk aşamada FPGA üzerinde ESA uygulamaları incelenmiştir. Kullanılan optimizasyon yöntemleri ve FPGA programlama yöntemleri araştırılmıştır. İkinci aşamada ise medikal görüntü çakıştırma uygulamaları araştırılmıştır. Son aşama olarak ise FPGA temelli yapılan görüntü işleme çalışmaları araştırılmıştır.

Son zamanlarda, belirli ESA modellerine FPGA tabanlı hızlandırma yöntemleri uygulanmaktadır. Donanım tasarımlarında yüksek performans elde etmek için, paralel hesaplama yöntemleri, boru hattı, veri akış boru hattı, dizi bölme, sayıları daha az bit ile temsil etme gibi yöntemler kullanılmaktadır. Li ve diğerleri [61] çalışmalarında tam bağlantılı katmanlar için iki farklı hesaplama yöntemi önererek uygulama için gereken yongayla tümleşik (on chip) tampon (buffer) bellek miktarını azalttıklarını belirtmişlerdir. Doğrulama için AlexNet modelini Xilinx VC709 platformu üzerinde gerçekleştirmişlerdir. Önerdikleri model 156 MHz çalışma frekansıyla 565,94 GOPs (saniyede yapılan gigaişlem sayısı, Giga operation per second) performansını ulaştığını ifade etmişlerdir. Katmanları teker teker oluşturup birbirine bağlamışlardır. FPGA'ların dezavantajı olan sınırlı bellek bant genişliğinden dolayı, tam bağlantılı katmanların ağırlıklarını yongayla tümleşik hafıza birimlerine depolamışlardır.

Yapılan çalışmalarda dikkat edilen bir önemli noktada Tasarım Alanı Keşfi (Design space exploration, DSE) çalışmalarıdır. DSE, geçici tasarım noktaları alanından istenen tasarım gereksinimlerini en iyi şekilde karşılayan bir tasarımı, çözümü veya çözümleri bulma sürecidir [62]. Bu keşif, kendiliğinden karmaşıktır, çünkü arama süreci, çeşitli soyutlama seviyelerinde kod dönüşümleri ve derleyici optimizasyonları uygulamaktan, belirli parametre değerlerini seçmekten (örneğin, yapılandırmaları) ve hatta algoritmik alternatifleri belirlemekten kaynaklanan geçici tasarımları içerebilir [62]. Mevcut gömülü sistemlerin geniş çeşitliliği göz önüne alındığında, bu araştırma alanının büyüklüğü, alternatif tasarım seçeneklerini ve ilgili tasarım üretimini otomatikleştirmeyi zorunlu kılar.

Zhang ve diğerleri [63] yaptıkları çalışmalarında Roofline modelini kullanarak DSE çalışmasına odaklanmışlardır. Roofline modeli nicel bir analiz yöntemidir. Kullanılan platformun teorik olarak sınırlarının belirlenmesinde kullanılabilir. Her platform için

(örneğin, her farklı FPGA modeli için) kullanılacak kaynak sayıları ve bant-genişliği farklı olduğu için, Roofline modelinde sınırlar farklı belirlenmektedir. AlexNet ağını, iki farklı CPU üzerinde ve VC707 FPGA kartı üzerinde yaptıkları uygulamalarda, gecikme süresi CPU'lar için 376,50 ms ve 103,48 ms çıkarken, FPGA için 21,61ms olarak çıkmıştır. Sistem performansları ise CPU'lar için 3,54 GFLOPS ve 12,87 GFLOPS, FPGA için 61,62 GFLOPS olarak hesaplanmıştır. Sistem performansı açısından yaklaşık 17 kat yüksek performansla DÖ ağının gerçekleştirildiği görülmektedir.

Ayat vd. [64] yaptıkları çalışmalarında Roofline model gösterimine frekans parametresi ekleyerek oluşturdukları yeni modellerle AlexNet gerçekleştirmesini ZYNQ ZC706 FPGA üzerinde yapmışlardır. Kullandıkları modele göre tiling parametresini optimum seçmeye çalışmışlardır. Çalışmalarının sonunda donanım performansının 52,11 GFLOPS olarak ve işlem başına tüketilen güç miktarının 10,02 GFLOPS/W olarak ölçmüşlerdir. Roofline modeli kullanılarak yapılan çalışmalarda kullanılan parametrelerin belirlenmesinde optimizasyon yöntemlerinin kullanılmadığı, deneme yanılma yoluyla veya yazarların kendi geliştirdiği algoritmalara göre parametrelerin belirlendiği gözlenmiştir.

Ma ve diğerleri [65], optimum bir tasarım elde etmek için, farklı tasarım yöntemlerinin niteliksel analizi için akış diyagramı oluşturmuşlardır. Böylece, hafıza kullanımı, döngü açma veya döngü döşeme (loop tiling) gibi tekniklerin, DSE üzerindeki etkisi önceden tahmin edilebilir hale gelmiştir. Önerdikleri hızlandırma modellerini NiN, VGG-16 ve ResNet-50/152 ağları için çıkarım aşamasında uygulamışlardır. Ayrıca iki farklı FPGA üzerinde gerçekleştirme yaptıkları çalışmalarında, VGG-16 ağı için ulaşılan performans Intel Stratix V FPGA sını için 348 GOPs iken, Arria 10 FPGA sını için 715 GOPs hesaplanmıştır. Stratix V te Arria 10 FPGA'sına göre daha az DSP kaynakları olduğu için, bu FPGA ile yapılan çalışmada paralel işlem sayısı daha az tutulmuştur. Arria 10 da yapılan paralel işlem sayısı daha fazla olduğu için, performansı da yüksek olmuştur.

Khan ve diğerleri [66], önceki yapılan çalışmalardan farklı olarak 2D ESA yerine yeni bir inception-tabanlı 3 boyutlu ESA (I3D) modeli için tasarım parametrelerini araştırmak üzerine bir çalışma yapmışlardır. 70 katmana sahip olan bu yapı, video görüntülerindeki aksiyonları tahmin etmek için tasarlanmış bir ağ modelidir. Bu ağın karmaşıklığını azaltmak amacıyla ilk önce, eğitimden önce, ağırlıkların ve kullanılacak giriş verilerinin uzunluğu kısaltılmıştır. İkinci olarak, döngü döşeme tekniğini kullanarak giriş verilerinin hepsini bir

kerede almak yerine parçalara bölerek almışlardır. Böylece dinamik rastgele erişim belleği (dynamic random access memory, DRAM) alış-veriş trafiği azaltılmış ve hafıza bant-genişliği iyileştirilmiştir. Yaptıkları bu optimizasyonlar sonucu 32-bit kayan noktalı sayı gösterimi kullanarak VC709 FPGA tabanlı gerçekledikleri model 684 GOPs performansı yakalamıştır. Doğruluk oranındaki düşüşün ise sadece %2 kadar olduğu belirtilmiştir.

Messaoud ve diğerleri [34], trafik işaretlerinin tanınması için eğittikleri ESA ağını, iki farklı tasarım için üç farklı şekilde PYNQ-Z1 üzerinde gerçekleştirmişlerdir. İlk tasarım olan tek birim kullanarak katmanları sırayla çalıştırdıkları yapı 3,55 saniyede tamamlanmıştır. İkinci tasarımda ise katmanların teker teker tasarlanıp arka arkaya bağlandığı yapı 2,13 saniyede çalışmasını tamamlamıştır. Lenet-5 ağının ARM işlemci üzerinde çalışması ve iki tasarımın FPGA üzerinde çalışmasını kıyasladıkları zaman, FPGA üzerinde yapılan işlemlerin daha az güç tükettiği ve daha hızlı sonuç ürettiği tespit edilmiştir. Ayrıca Lenet-5 ağının son katmanlarında bulunan tam bağlantılı katmanlar yerine 1x1 filtre boyutlarına sahip evrişim katmanları kullanmışlardır. Böylece ağın karmaşıklığını azaltmışlardır.

Medikal görüntü çakıştırma (GÇ) problemi için literatür incelendiğinde derin öğrenme tabanlı yapılan medikal GÇ çalışmalarının klasik yöntemlere göre daha iyi sonuçlar verdiği görülmektedir. Sloan vd. [67] ESA kullanarak katı (rigid) görüntü çakıştırma uygulaması yapmışlardır. ESA modellerini hem mono-model hem de çoklu-model için eğitmişlerdir. Mono-model çalışması için OASIS veri tabanından aldıkları MR görüntülerini, çoklu-model çalışması için ise IXI veri tabanından aldıkları T1 ve T2 ağırlıklı MR görüntülerini kullanmışlardır. Çalışmanın sonunda geliştirdikleri yapıyı klasik çakıştırma yöntemleriyle kıyaslamışlardır. Değerlendirmelerini ağın çıktısı olan döndürme (rotation), x eksenine göre öteleme (x translation) ve y eksenine göre öteleme (y translation) miktarları olmak üzere üç parametre üzerinde yapmışlardır. Değerlendirme, bu üç parametredeki ortalama hata miktarı hesaplanarak yapılmıştır. Farklı veri setleri kullanarak yaptıkları değerlendirmede, klasik yöntem kullanılarak yapılan çalışma her üç parametre için en kötü sonucu verirken, önerdikleri ESA tabanlı yöntem her üç parametre için en iyi sonucu vermiştir.

Eppehof ve Pluim [68] danışmanlı öğrenme yöntemi kullanan ESA modeli kullanarak medikal CT 2D ve 3D görüntü çakıştırma uygulaması yapmışlardır. Çakıştırma için kullandıkları görüntü çiftlerini sentetik olarak oluşturmuşlardır. Performans değerlendirme ölçütleri olarak, RMSD (root mean squared distance) ve nitel değerlendirme yöntemlerini

kullanmışlardır. Önerdikleri ESA, dört evrişim katmanından, 2 adet havuzlama katmanından ve 3 adet de tam bağlantılı katmandan oluşmaktadır. Girdi olarak hareketli-sabit görüntü çiftini alan ağ, çıktı olarak bu iki görüntü arasındaki hatayı tahmin etmektedir. Klasik yöntemlerle yaptıkları kıyaslama sonucunda, önerdikleri ESA'nın daha başarılı sonuçlar verdiğini ifade etmişlerdir.

De-Vos vd. [69], önerdikleri derin öğrenme tabanlı yöntem, danışmansız eğitim modelini kullanmaktadır. Hem affin hem de deforme dönüşüm modeline sahip görüntüler için yaptıkları çalışma, klasik yöntemlere göre daha iyi sonuçlar vermiştir. Çalışmalarında MR ve CT görüntüler kullanmışlardır. Çalışmalarında iki yapı önermişlerdir. İlk yapı iki görüntü çifti arasındaki döndürme, öteleme, ölçekleme ve kırpma parametrelerini tahmin eden yapı iken, ikinci yapı ise görüntü çifti arasındaki deforme olan alanları piksel piksel hesaplayan bir modeldir. Değerlendirme ölçütü olarak, hedef çakıştırma hatası (Target registration error, TRE), Dice skoru ve Hausdorff mesafesi metriklerini kullanmışlardır. Kullandıkları ESA yapısı girdi olarak hareketli ve sabit görüntüleri ayrı ayrı iki koldan almaktadır. Birleştirme katmanı kullanarak daha sonra iki koldan gelen özellik haritaları birleştirilerek çıkış katmanı oluşturulmuştur.

Wei vd. [70], karaciğer tümör ameliyatlarına yardımcı olmak için yaptıkları çalışmada, 2D US görüntüsü ile 3D CT/MR görüntü çakıştırma uygulaması yapmışlardır. Medikal operasyon öncesinde çekilen CT/MR görüntülerinin operasyon sırasında 2D US görüntülerle çakıştırılmasını hedefleyen çalışmalarında, görüntü bölütleme kullanarak başarılı bir yöntem geliştirmişlerdir. Kullandıkları U-net benzeri yapının performansını, Dice skoru kullanarak ve döndürme, öteleme parametreleri için hata hesaplayarak karşılaştırmışlardır.

FPGA teknolojisi kullanılarak yapılan uygulamalar incelendiğinde, özellikle görüntü işleme alanında FPGA'nın sıklıkla kullanıldığı görülmektedir. Stratakos vd. [71] çalışmalarında FPGA tabanlı sistemlerde medikal uygulamalar için görüntü işleme problemine odaklanmışlardır. Genel amaçlı hesaplama birimi olarak CPU ve FPGA platformlarını sistem üzerinde çip (System on Chip, SoC) birleştirerek heterojen bir yapı kullanmışlardır. Çalışmalarında altı tane affin dönüşüm parametresinin tahminini yapmışlardır. Çalışmalarında üç uygulama yaparak bunların hız açısından kıyaslamasını göstermişlerdir. Kullandıkları platformlar Zynq-7000 SoC, Intel i5 4200U CPU ve Zynq üzerinde bulunan ARM CPU'dur. Görüntü boyutu 256 x 256 olarak alınmıştır. Tek duyarlıklı kayan noktalı

sayı gösterimi kullanarak yaptıkları uygulamada, heterojen uygulamanın 67 kat daha hızlı sonuç ürettiğini ifade etmişlerdir.

White [72], 640x480 görüntü boyutu kullanarak yaptığı çalışmada 100 MHz frekans altında FPGA tabanlı çalışmayı, kayan noktalı sayı kullanarak yaptığı 2,4 GHz Intel Core 2 Quad CPU üzerinde Matlab sonuçlarıyla kıyaslayınca, önerdiği FPGA tabanlı çalışmanın yaklaşık 400 kat daha hızlı olduğunu ifade etmiştir.

Chen vd [73] US görüntü oluşturmada kullanılan bir aşama olan hüzmeye şekillendirme (beamforming) işlemi için FPGA, GPU ve CPU kullanarak bu üç platformun hız karşılaştırılmasını yapmışlardır. Araştırmalarının sonuçlarına göre hem FPGA temelli çözümler hem de GPU tabanlı çözümler, saniyede 80 kareyi aşan gerçek zamanlı performans sağlamış ve temel CPU uygulamalarına kıyasla 38 katın üzerinde bir performans artışı elde etmişlerdir. Ayrıca, GPU tabanlı geliştirmenin FPGA alternatifine göre daha hızlı tamamlandığını, ancak FPGA çözümünün, uçtan uca gerçek zamanlı yeniden yapılandırılabilir bir görüntü oluşturma sistemi için gerekli olan I/O bant genişliğini sağlamada daha etkili olduğunu belirtmişlerdir.

Dandekar vd. [74] klasik yöntemler kullanarak, ortak bilgi tabanlı bilgisayarlı tomografi CT ve pozitron emisyon tomografisi (positron emission tomography, PET) görüntüleri kullanarak yaptıkları deforme çakıştırma uygulamasını hızlandırmak için FPGA tabanlı yaklaşım kullanmışlardır. Çalışmanın sonucunda, ortak bilgiye (mutual information, MI) dayalı klasik yöntem çakıştırma işleminin FPGA üzerinde uygulaması, klasik tabanlı uygulamayla yaklaşık olarak aynı TRE değerine sahipken, hız açısından MI tabanlı deforme kayıt işleminin süresini saatlerden birkaç dakikaya düşürdüğünü belirtmişlerdir.

FPGA tabanlı gömülü sistemlerin yüksek performanslı görüntü işleme uygulamaları için etkili ve güç verimli bir çözüm sunabileceğini vurgulamak için yapılan çalışmalardan önemli bir tanesi de Siddiqui vd [75] çalışmasıdır. Görüntü işleme ve diğer yoğun hesaplama uygulamaları için 337 MHz'e kadar çıkabilen yüksek seviyeli Xilinx FPGA platformunu kullanarak yaptıkları çalışmalarda k-means kümeleme ve trafik işareti tanıma uygulamalarını gerçekleştirmişlerdir. k-means kümeleme işleminde uygulamasının, ARM Cortex-A7 CPU, nVIDIA GeForce GTX980 GPU ve ARM Mali-T628 gömülü GPU'ya göre sırasıyla 57, 28 ve 1,7 kat daha verimli bir güç tüketimi (fps/W) sunduğunu ifade etmişlerdir.



### 3. EVRİŞİMLİ SINIR AĞLARI

Derin öğrenme tabanlı yöntemlerin hızlı ilerlemesinin merkezinde yer alan evrişimli sınır ağları, pratik uygulamalarda son derece başarılı olmuştur. Derin öğrenme kullanan yöntemler içerisinde en çok kullanılan ve en çok bilinen algoritma ESA yöntemleridir [2]. Özellikle görsel veri analizi alanında büyük bir başarı elde etmiştir ve genellikle tercih edilen sınır ağı türüdür. ESA modelinin en büyük avantajı, girdi olarak verilen verilerden uygun öznitelikleri otomatik olarak oluşturması ve bu aşamada insan yardımına veya denetimine ihtiyaç duyulmaması olarak görülmektedir [2]. Evrişimli sınır ağı adı, ağda evrişim işleminin kullanılmasından gelmektedir. ESA katmanlarından en az birinde genel matris çarpımı yerine doğrusal bir işlem olan evrişimi kullanan basit sınır ağlarıdır [76]. Klasik ESA modeli genellikle evrişim, aktivasyon, havuzlama ve tam bağlantı katmanları olmak üzere dört önemli katmandan oluşmaktadır. Klasik sınır ağları, genellikle katman katman veri üzerinde ilerleyerek özelliklerin çıkarımını yapar. Her katman, önceki katmanın çıktıları üzerinden bağımsız olarak işlem yapar. ESA yapılarında ise evrişim ve havuzlama katmanları aracılığıyla özellik çıkarımını gerçekleştirir. Evrişim katmanları, veri üzerindeki örüntüleri tespit ederken havuzlama katmanları, özellik haritalarının boyutunu azaltır ve önemli özellikleri vurgulanmasını sağlar.

Farklı özniteliklerin ortaya çıkmasını sağlayan matematiksel bir işlem olan evrişim işlemi, farklı filtre veya kernel olarak adlandırılan matrislerin giriş özellik haritalarının üzerinde kaydırılarak denk gelen alanlarla matris çarpılmasıyla yapılmaktadır. Giriş bilgisi bir görüntü, ses, metin veya başka verilerden oluşabilir. Her verinin yapısı farklıdır, fakat evrişim işleminde bazı ortak noktalar vardır [77]. Bunlar genel olarak şunlardır:

- Veriler çok-boyutlu dizilerde (tensörlerde) saklanırlar.
- Her veri bir veya daha fazla eksen içerir. Görüntü verileri genişlik, yükseklik eksenleri, ses verileri ise zaman eksenini içerir.
- Verilerin farklı görünümüne erişmek için kanal eksenini kullanılır. Görüntülerde kırmızı, yeşil, mavi, ses verisinde sağ ve sol kanallar.

ESA modelinde kullanılan ve bazı kaynaklarda ayrık evrişim işlemi (discrete convolution operation) [77] olarak geçen evrişim işlemi doğrusal bir işlemdir. ESA da uygulanan evrişim işleminin üç önemli özelliği makine öğrenmesine büyük katkı sağlamıştır [76]. Bunlar;

seyrek etkileşim (sparse interaction), parametre paylaşımı (parameter sharing) ve eşdeğer temsil veya eş değişken kullanımı [78] (equivarant representation) özellikleridir.

Klasik sinir ağı ve ESA'larda bulunan nöronların birbiriyle bağlantı kurma yapıları farklıdır. Klasik sinir ağları katmanlarında bulunan nöronlar kendinden önce ve sonra bulunan katmanlardaki bütün nöronlara bağlıdır. Fakat ESA yapılarının önemli bir özelliği olan seyrek bağlantı yapısı biraz farklıdır. Seyrek bağlantı, ESA da kullanılan filtreler giriş verisinin tamamında aynı anda değil yalnızca belirli bir bölgesinde etkili olurlar ve bu belirli bölgeyle bağlantı kurar. Seyrek bağlantı sayesinde ağın hafıza gereksinimi azalır, daha az bağlantıdan dolayı ağın işlem hızı artar.

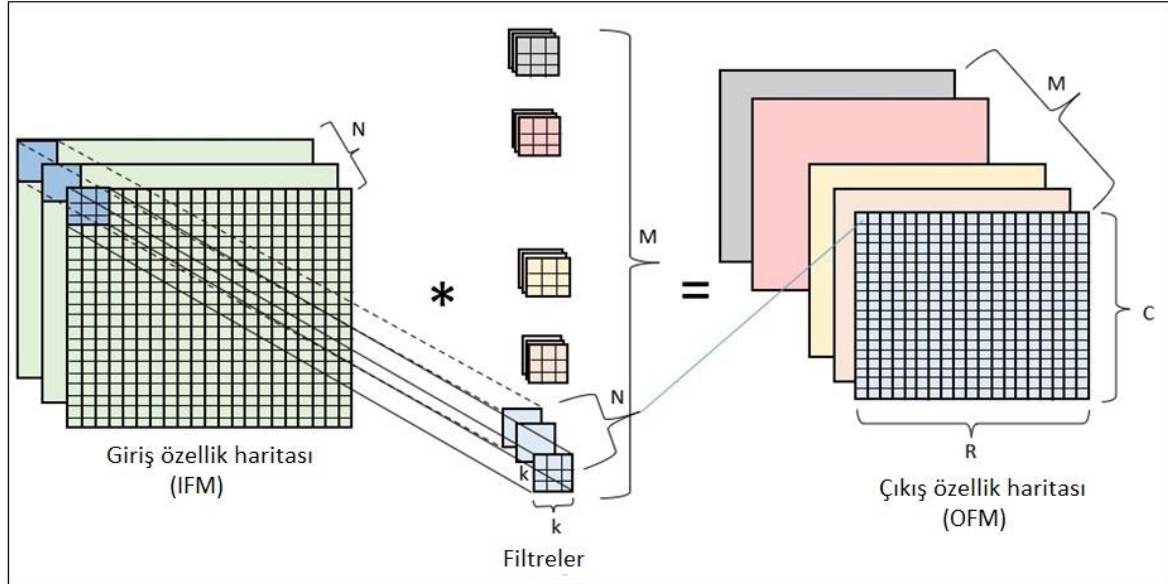
Klasik sinir ağlarında her bir ağırlık değeri yalnızca bir kez kullanılır. Bu da sinir ağının toplam parametre sayısını artırmakta ve genellikle daha fazla veri gerektirmektedir. ESA'larda ise ağırlık değerleri filtreler şeklinde kullanılır ve bu filtreler veri üzerinde gezdirilerek farklı bölgelerde paylaşılır. Bu parametre paylaşımı, sinir ağındaki toplam parametre sayısını azaltır ve daha az veriyle daha iyi genelleştirme yapabilir.

Eşdeğer gösterim evrişim katmanlarının kullanımına dayanmaktadır. Evrişim katmanları, filtrelerin veri üzerinde gezdirilmesiyle özellik haritalarını oluşturur. Bu filtreler, farklı pozisyonlardaki örüntüleri algılar ve bu sayede verinin yer değiştirmelerine karşı eş değer gösterim elde edilir. Örneğin, bir yüz tanıma modeli, farklı pozisyonlarda bulunan bir yüzü tanıyabilir ve aynı yüzü farklı yerlerde bulunduğu anda bile doğru sonuçları üretebilir. Eş değer gösterim, sinir ağlarının veri üzerindeki dönüşümlere duyarlı olmasını sağlar ve verinin yer değiştirmelerine, ölçek değişimlerine veya diğer dönüşümlere karşı genelleşme yeteneğini artırır. Bu, evrişimli sinir ağlarının özellikle görüntü işleme ve video analizi gibi yapısal olarak düzenli verileri işlerken etkili olmasının temel nedenlerinden biridir.

Evrişim katmanı özneliklerin çıkarıldığı en önemli katmandır. Evrişim işlemi, özel bir matematiksel işlem olan matris çarpımını temel alır. Temelde, filtre adı verilen küçük bir matris, girdi veri üzerinde belirli bir pencere boyutu boyunca hareket ettirilir ve bu pencere boyunca filtre ile verilerin elemanları arasında matris çarpımı yapılır. Daha sonra, elde edilen sonuçlar toplanarak çıktı oluşturulur. Evrişim işlemi 3B tensörler üzerinde işlem yapar.

Nitelik haritası olarak adlandırılan bu tensörler genişlik ve yükseklik diye adlandırılan iki uzamsal eksene ve derinlik veya kanal olarak adlandırılan üçüncü boyuta sahiptirler.

Evrişim işleminde 3B çarpma-biriktirme (multiply-accumulator, MAC) ve çarpma-toplama (multiply-add, MAD) işlemleri yapılmaktadır. Çok boyutlu bir görüntü için evrişim işlemi Şekil 3.1’de gösterilmiştir. H, W ve N sırasıyla giriş öznelik haritasının (Input feature map, IFM) yüksekliği, genişliği ve kanal sayısını göstermektedir. C, R ve M sırasıyla çıkış öznelik haritasının (Output feature map, OFM) yüksekliği, genişliği ve kanal sayısını göstermektedir. k ise kullanılan filtrelerin boyutunu temsil etmektedir.

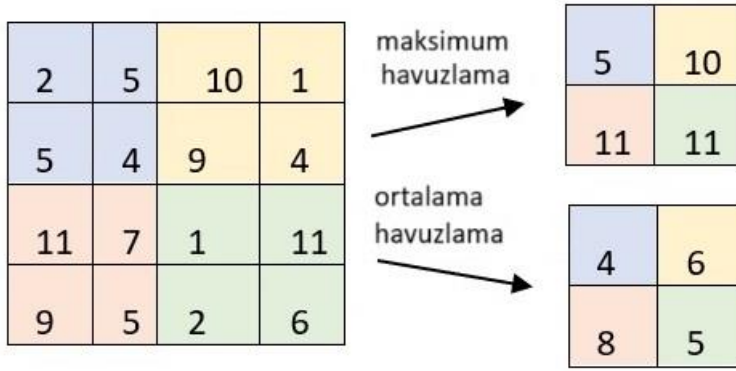


Şekil 3.1. Çok boyutlu bir görüntü için evrişim işlemi

Evrişim işlemi iki önemli parametre ile tanımlanır. Bunlar; girdi verisinden çıkarılacak parçaların boyutu ve çıkarılacak nitelik haritasının derinliğidir. Filtreler genellikle 3×3, 5×5 veya 7×7 boyutlarında seçilmektedir. Şekil 3.1’de gösterilen filtre boyutu 3×3 olarak gösterilmiştir.

Evrişim katmanına ek olarak havuzlama katmanı da ESA modelinin en önemli yapı taşıdır. Genellikle evrişim katmanının arkasında yer alan, aşağı örnekleme (down-sampling) yöntemi kullanarak sonraki katmana giriş verisi olarak kullanılan ve önceki katmanın çıkış verisi olan özellik haritalarının (Feature map, FM) genişliğini-yüksekliğini azaltmayı ve giriş haritalarını özetlemeyi sağlayan katmandır. Evrişim işleminden sonra elde edilen çıktı matrisinin boyutunu küçültürken, aynı zamanda elde edilen önemli özellikleri koruyarak

işlemi hızlandırmaya ve aşırı uyum (overfitting) riskini azaltmaya yardımcı olur. Aynı zamanda verilerde oluşan çeşitli dönüşüm ve kayma değişikliklerine karşı da invaryans (invariance) sağlamaya yardımcı olmaktadır [76]. Havuzlama katmanlarından sonra özellik haritalarının kanal sayısı değişmez, sadece yükseklik-genişlikleri değişir. Sık kullanılan iki havuzlama yöntemi bulunmaktadır. Bunlar; Şekil 3.2’de gösterildiği gibi maksimum havuzlama ve ortalama havuzlama yöntemleridir.

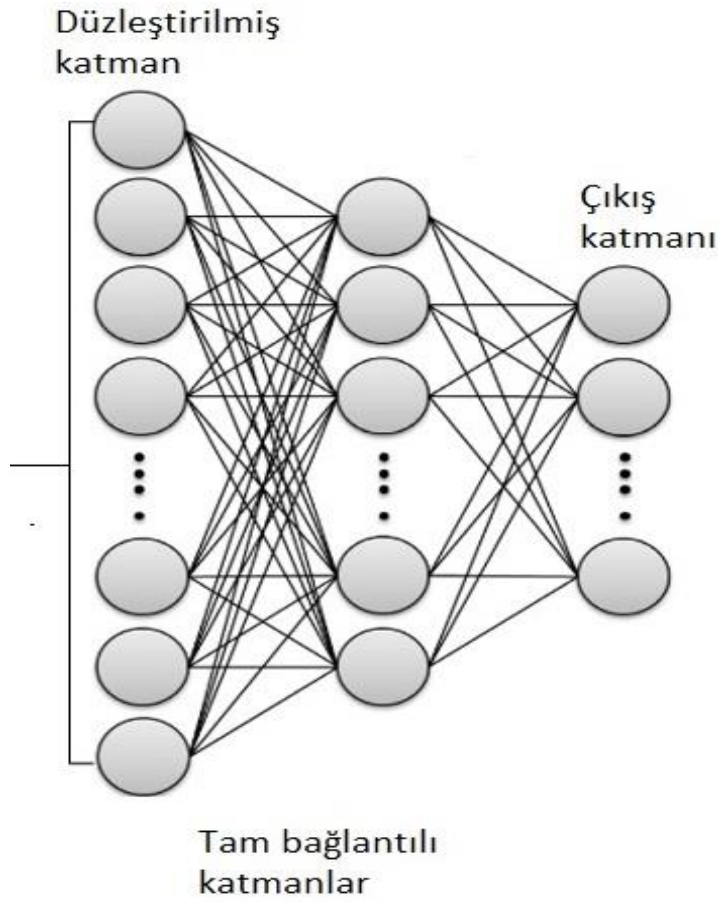


Şekil 3.2. Maksimum ve ortalama havuzlama işlemi

Ortalama havuzlama, bir bölge içindeki değerlerin ortalamasını alan havuzlama türüdür. Maksimum havuzlama ise her bir bölge içindeki en büyük değeri seçerek yeni özellik haritalarını oluşturmaktadır.

Havuzlama işlemi, ağıın eğitim sürecinde öğrenilen parametrelere sahip değildir. Yani, havuzlama katmanı sadece sabit bir işlem yapar ve girdi veri üzerinde öğrenme işlemi gerçekleştirmez.

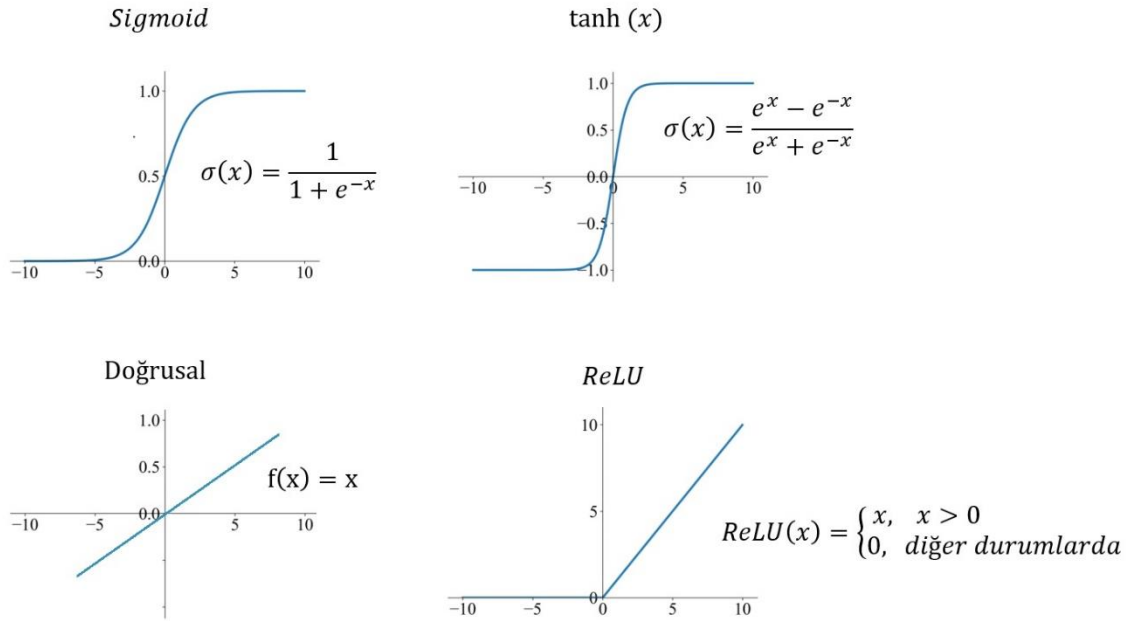
Genellikle ESA'nın sonunda yer alan tam bağlantılı katmanlarda, kendisinden önce gelen bütün nöronlar sonraki katmanda bulunan bütün nöronlara bağlıdır (Şekil 3.3). Bu yapı klasik sinir ağlarındaki yapıya benzemektedir. Bu katmanlar, evrişim ve havuzlama katmanlarının çıktılarını düzleştirerek ve son katmana doğru ilerler, sınıflandırma veya başka bir çıkarım görevi için kullanılırlar (Şekil 3.3). Tam bağlantı katmanları, mimarinin bir katmandaki bütün düğüm ve nöronlarının bir sonraki katmanlara bağlandığı bir yapay sinir ağ türüdür. Bu ağ türü hesaplama açısından karmaşık olmasının yanı sıra aşırı yüklenmeye meyillidir. Elde edilen verilere uygulanan düzleştirme işleminden sonra sinir ağları ile öğrenme işlemi uygulanır.



Şekil 3.3. Tam bağlantılı katman

Tam bağlantılı katman girdi haritalarından global örüntüleri öğrenmekte iken, evrişim katmanı bölgesel örüntüler öğrenmektedir [79].

Sinir ağlarında önemli bir etkiye sahip olan katmanlardan bir tanesi de aktivasyon katmanıdır. Evrişim ve havuzlama katmanları gibi diğer katmanlardan gelen çıktıları alır ve bu çıktılara doğrusal olmayan bir dönüşüm uygulayarak, nöronların doğrusal olmayan özellikler öğrenmesine ve ağı daha karmaşık veri dağılımlarını modellemesine yardımcı olurlar. Şekil 3.4'te en sık kullanılan aktivasyon fonksiyonları gösterilmiştir.



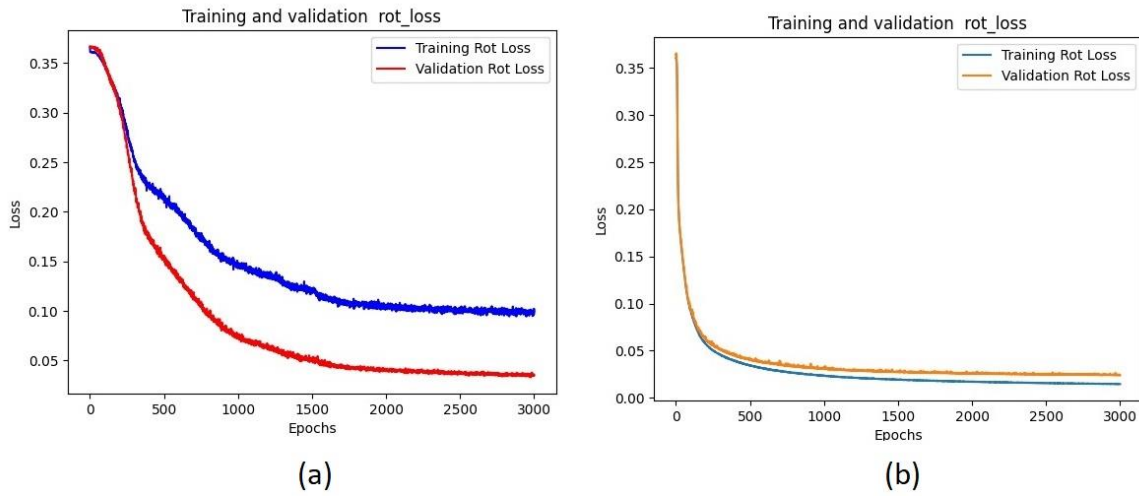
Şekil 3.4. Aktivasyon fonksiyonları

Sigmoid fonksiyonu girdi olarak aldığı değerlere karşılık 0 ile 1 arasında değerler üretir. Tanh fonksiyonu, sigmoid fonksiyonuna benzer ancak çıkış değerleri -1 ile 1 arasında olacak şekilde genişletilmiştir. En yaygın kullanılan aktivasyon fonksiyonu doğrultulmuş lineer birimdir (Rectified Linear Unit, ReLU). Matematiksel olarak, ReLU fonksiyonu, giriş değeri 0'dan küçükse 0, 0'dan büyükse giriş değerini direkt olarak çıktı olarak verir. Yani, negatif değerlerde hiçbir eşikleme yoktur ve pozitif değerler doğrudan aktarılır. Bu fonksiyon, hesaplamaları hızlandırır ve aşırı uyumu (overfitting) azaltmaya yardımcı olur. Özellikle donanım tabanlı çalışmalarda, sigmoid ve tanh gibi eksponensiyel terim içeren fonksiyonlara göre gerçekleşmesi daha kolay ve gerçekleşirken kullanılan kaynak sayısı daha az olduğu için tercih edilmektedir.

Makine öğrenmesinde temel konu, eniyileme ile genelleştirme arasındaki çekişmedir [79]. Kullanılan ağ eğitim veri seti üzerinde en iyi performansı yakalayabilmesi için eğitilerek eniyileme yapılmaktadır. Modelin ilk defa göreceği veriler üzerinde performansının değerlendirilmesi ise genelleştirme kabiliyetini göstermektedir. Eğitilen modeller aşırı uydurma (overfitting) veya yetersiz uydurma (underfitting) durumlarıyla karşılaşabilirler. Yetersiz uydurma eğitim veri setini yeterince öğrenememesi, eğitim ve test veri setlerinde yüksek hataya neden olması durumudur. Aşırı öğrenme ise, modelin eğitim veri setini iyi öğrenmesine rağmen, test veri seti üzerinde yüksek hata yapması durumudur. Her iki

durumun iyileştirilmesi için kullanılan bazı yöntemler vardır. Bunlar, verilerin artırılıp çeşitlendirilmesi, ağın karmaşıklığının artırılması veya azaltılması, seyreltme (dropout) kullanımı, ağırlıkların düzenlenmesi gibi yöntemler denenerek aşırı öğrenme ve yetersiz öğrenme arasında bir optimum çözüm bulunabilir.

Seyreltme yöntemi uygulandığı katmanda, eğitim sırasında ağı öğrendiği bilgilerin bir kısmı rastgele olarak sıfırlanır [79]. Seyreltme yöntemi seyreltme oranı kullanılarak uygulanır. Seyreltme oranı genellikle 0,2 ile 0,5 arasında bir değer seçilir. Bu değer sıfırlanacak elemanların toplam elemanlara oranını ifade eder. Seyreltme eğitim sırasında uygulanır. Doğrulama ve test aşamalarında uygulanmaz. Bundan dolayı, seyreltme uygulanan ağlarda eğitim kayıp değeri yüksek çıkabilir fakat doğrulama ve test kayıp değerleri düşük çıkabilir. Şekil 3.5'te seyreltme katmanının eğitim ve doğrulama kayıp değeri üzerindeki etkisi gösterilmiştir.



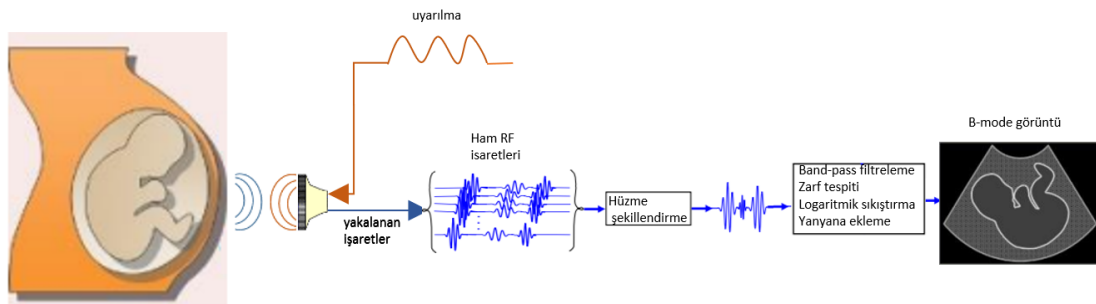
Şekil 3.5. Seyreltme yönteminin eğitim ve doğrulama kayıp grafiği üzerindeki etkisi, seyreltme (a) kullanılan ve (b) kullanılmayan ağı kayıp grafikleri

Ağın performansını belirleyen birçok etken vardır. Bunlar genelde hiper parametre olarak adlandırılırlar. ESA modeli için kullanılacak katmanlar ve bu katmanların sayısı, genişliği, kayıp fonksiyonu, eniyileme yöntemi, devir sayısı gibi birçok hiper parametre bulunmaktadır.



## 4. ULTRASON GÖRÜNTÜLEME

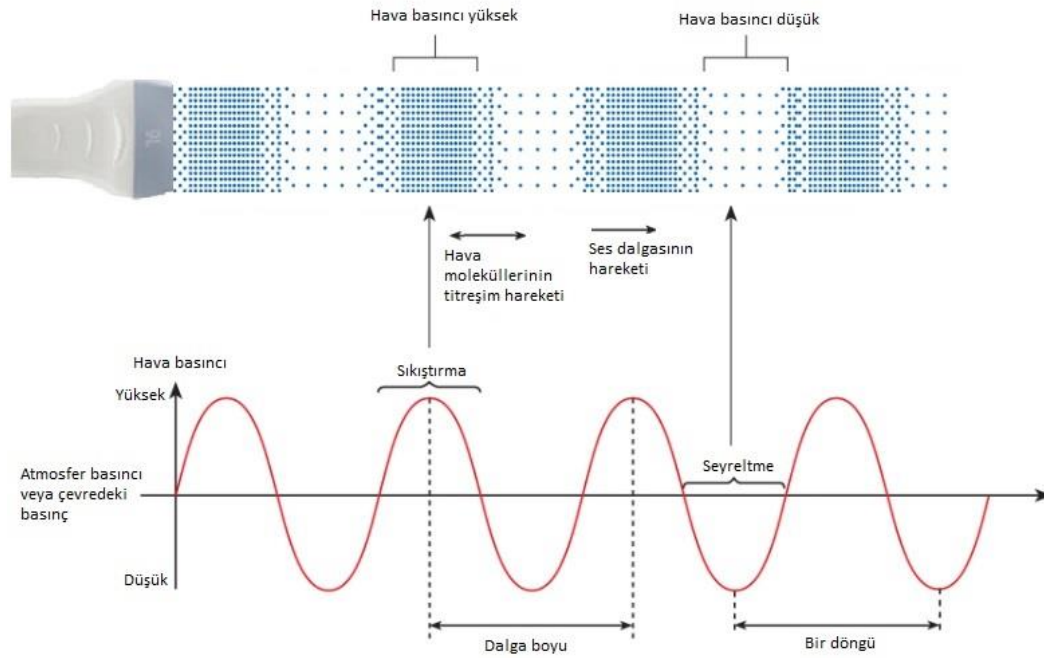
Ultrason görüntüleme klinik uygulamalarda yarım yüzyıldan fazla bir süredir kullanılmaktadır. Özellikle taşınabilir olması, gerçek zamanlı olması ve manyetik rezonans görüntüleme, bilgisayarlı tomografi gibi modalitelere kıyasla daha az maliyetli olması nedeniyle kırsal alanlarda da tercih edilen görüntüleme modalitesidir. Tanısal görüntüleme modaliteleri arasında yer alan US görüntüleme non-invaziv, iyonlaştırıcı olmayan, ağrısız ve gerçek zamanlı bir görüntüleme tekniğidir. X-ray ışınları kullanılmadığı için zararsız bir yöntem olarak görülmektedir. Tıbbi bir görüntüleme yöntemi olarak ultrason görüntüleme vücudun iç kısımlarında bulunan organ ve dokuların genel hatlarıyla görüntülenmesini ses dalgaları yardımıyla sağlayan bir görüntüleme teknolojidir. Ultrason görüntülemenin temel prensibi, dönüştürücüler tarafından üretilen insan kulağının duyamayacağı yüksek frekanslı ses dalgasının görüntülenmek istenen bölgeye iletilmesi ve farklı dokuların arayüzeylerinden yansıyan dalgaların dönüştürücüler tarafından yakalanarak genliklerinin zamana karşı kayıt altına alınmasıdır. US görüntülemenin temel çalışma prensibi Şekil 4.1'de gösterilmiştir. Bu şekilde nihai görüntü B-modundadır. US görüntüleme işlemi problemlerin içerisinde bulunan dönüştürücülerin (transducer), elektriksel olarak üretilen işaretleri yüksek frekanslı ses dalgalarına dönüştürüp görüntüleme alanına göndermesiyle başlamaktadır. Gönderilen ses hüzmelerinin bir kısmı önünde bulunan yüzeylere çarparak geri yansımakta, geri kalan dalgalar ise yoluna devam etmektedir. Devamında geri yansıyan (echo) dalgaların tekrar dönüştürücüler yardımıyla yakalanıp, elektriksel radyo frekans (RF) sinyallerine dönüştürülmesiyle devam eder. Ham RF sinyalleri hüzmeye şekillendirme (beamforming), zarf alma, logaritmik sıkıştırma gibi birçok aşamadan geçtikten sonra son aşama olan B-mod görüntülerine dönüştürülür.



Şekil 4.1. US görüntüleme aşamaları

#### 4.1. Akustik Dalgaların Özellikleri

Ses, kendisini oluşturan parçacıkların sıkıştırılması (yüksek basınç) ve seyreltilmesi (düşük basınç) yoluyla sürekli, elastik bir ortamda yayılan mekanik bir enerjidir. Ses dalgaları uzunlamasına (longitudinal waves) yayılan dalgalardır [80]. Uzunlamasına dalgalar için ortamda bulunan parçacıkların hareket yönü dalganın yönüne paraleldir. Sıkıştırma ve seyreltme, parçacıkların yerel olarak yer değiştirmesiyle meydana gelmektedir. Sıkıştırma bölgeleri ve seyrekleşme bölgeleri sırasıyla yüksek ve düşük basınç alanlarına karşılık gelir.



Şekil 4.2. Ses dalgalarının ortamda ilerlemesi ve sinüs dalgası ile karakterize edilmesi

Şekil 4.2 de gösterilen sinüs dalga formu, genlik ve dalga boyunun dalga parametreleri ile karakterize edilir. Belirli bir andaki genlik, parçacığın o belirli zamandaki ortalama konumundan yer değiştirmesinin büyüklüğüdür. Tepe genliği, maksimum parçacık yer değiştirmesini temsil eder. Bu, basınçtaki maksimum değişimle ilişkilendirilir ve ultrasonik hızının yoğunluğu ile bağlantılı olan titreşimlerin gücünü yansıtır. Genlik, mesafeye göre çizildiğinde, bir sinüs dalgası üzerinde ardışık iki konum arasındaki uzaklığı ifade eder. Bu, tam bir dalga dögüsü sırasında basınç dalgasının kat ettiği mesafeyi temsil eder.

Ultrason sinyallerinin enerjisi boşlukta hızla zayıflar, bu nedenle yayılması için bir ortam gerekmektedir. Bir ortamla temas halinde olan bir ultrason kaynağı, ortamın parçacıklarında

titreşimleri başlatarak mekanik enerjiyi ortama aktarır. Titreşen bir parçacık, ortam içindeki ortalama konumu etrafında mikroskobik bir ileri-geri hareket gerçekleştirir. Küçük titreşim hareketleri sayesinde, yakındaki diğer parçacıklar da benzer şekilde etkilenir ve enerji doğrudan bir parçacıktan diğerine aktarılır, bu da onları hareketlendirir. Her titreşen parçacık enerjiyi çevresindeki yakın parçacıklara ileterek, ortamın büyük bir bölümü hızla mekanik titreşimlerle dolar.

Ses, bir ortamda yayılma hızını, o maddeye özgü sıkıştırılabilirlik, sertlik ve yoğunluk özellikleri tarafından belirlenir. Yayılma hızı, bir ortamın sertleştikçe artar, yoğunluğu azaldıkça ise azalır. Örneğin, sıkıştırılabilir ve düşük yoğunluğa sahip hava, sesin yayılma hızı bakımından daha düşüktür; ancak sert ve yoğun kemik, sesin daha hızlı yayılmasını sağlar. Tıp alanında, sesin yayılma hızı (örneğin 1540 m/s) çeşitli dokular için sabit kabul edilir, bu da zaman içindeki derinlik etkileşimlerini hesaba katar. Sesin yayılma hızı, sesin frekansı veya dalga boyuyla ilgili değildir, sadece ortama bağlıdır.

Sesin yayılma hızı, bir ortamdaki sıkıştırma direnci ve yoğunluğun bir ölçüsü olan yığın modülü (Bulk Modulus) tarafından belirlenir. Bu modül, bir maddeye özgüdür ve sesin hızını belirler. Ses hızındaki değişiklikler, tıbbi ultrason cihazları gibi uygulamalarda önemlidir, çünkü beklenmeyen sonuçlara yol açabilir. Belirli bir malzeme için ses hızı, malzemenin yoğunluğu ( $\rho$ ) ve yığın modülü (B) ile ilişkilidir..

$$c = \sqrt{\frac{\beta}{\rho}} \quad (4.1)$$

Ses spektrumu genellikle düşük frekanslı ses (infrasound), işitilebilir ses ve ultrason olmak üzere üç aralığa ayrılır. İşitilebilir sesler, insan kulağının algılayabildiği seslerdir. Çoğu insanda işitilebilir frekans aralığı yaklaşık 20 Hz - 20.000 Hz'dir. Frekansı insan kulağının algılayabildiğinin altında olan seslere düşük frekanslı ses (<20 Hz), insan kulağının algılayabileceğinden daha yüksek frekanstaki seslere ise ultrason (>20kHz) denir. Bu nedenle ultrason, 20 kilohertz'den (20 kHz) daha yüksek frekanstaki ses olarak tanımlanabilir [81]. Tıbbi görüntüleme için ultrason, megahertz (MHz) seviyesindeki frekansları kullanır. Teşhis uygulamaları için kullanılan ses frekansları genel olarak 2 MHz-15 MHz arasındadır, ancak 50 MHz'e kadar yüksek frekanslar belirli özel görüntüleme

uygulamaları için de arařtırmalar yapılmaktadır. Çizelge 4.1’de tıbbi ultrasonla ilgili dokular ve malzemeler için yoğunluk, ses hızı ve akustik empedans deęerleri verilmiřtir.

Çizelge 4.1. Tıbbi ultrasonla ilgili dokular ve malzemeler için yoğunluk, ses hızı ve akustik empedans [82]

Materyal	Yoęunluk (kg/m <sup>3</sup> )	Hız (m/s)	Z (rayls)	dB/cm (1MHz için)
Hava	1,2	330	$3,96 \times 10^2$	12
Akcięer	300	600	$1,80 \times 10^3$	
Yaę	924	1450	$1,34 \times 10^6$	0,52
Su	1000	1480	$1,48 \times 10^6$	0,0022
Yumuřak doku	1050	1540	$1,62 \times 10^6$	
Böbrek	1041	1565	$1,63 \times 10^6$	1
Kan	1058	1560	$1,65 \times 10^6$	0,17
Karacięer	1061	1555	$1,65 \times 10^6$	0,69
Kas	1068	1600	$1,71 \times 10^6$	1,2
Kafatası kemięi	1912	4080	$7,8 \times 10^6$	11,3
Piezoelektrik	7500	4000	$3,0 \times 10^7$	

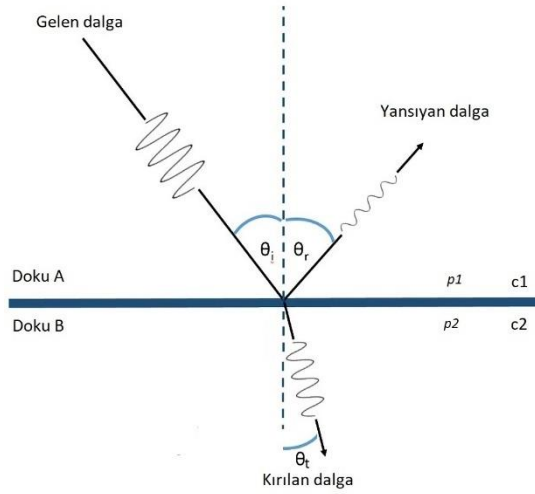
#### 4.2. US Doku Etkileřimi

US dalgaları homojen olmayan dokulardan geęerken, bir kısmı daha derin yapılara iletilir, bir kısmı yankı olarak dönüřtürücüye geri yansıtılır, bir kısmı saçılır ve bir kısmı da ısıya dönüřür [83]. Geri yansıyan ultrason dalgalarına eko (echo) denir. Pürüzsüz ve büyük yüzeylerde ultrason dalgaları yansıma ve kırılmaya uğrarlar. Fakat dokuların içinde homojen olmayan ve farklı yoğunlukta ve küçük kitleler mevcuttur. Ultrason hüzmeleri bu tür kitlelere çarptığında saçılmaya uğrar ve yön deęiřtirir. Görüntüleme ve teřhis amacıyla, çoęunlukla dönüřtürücüye geri yansıyan ekolar önemlidir. Doku arayüzüne çarptıktan sonra geri dönen yankı miktarı, akustik empedans tarafından belirlenir. Ultrason dalgalarının dokulardan yayılmasını engelleyen empedansa akustik empedans (Z) denir. Akustik empedans her doku için farklıdır. Akustik empedans, dokudaki sesin yoğunluęunun ve hızının ürünüdür.

$$Z = \rho c \quad (4.2)$$

Eřitlikteki  $\rho$  ortamın yoğunluęunu,  $c$  ise akustik dalğanın ortamda ilerleme hızını göstermektedir. Temel birimleri rayl olarak adlandırılmıř birimdir. Akustik empedans,

belirli bir ortamda ve ortamlar arasında ultrasonun yayılma özellikleriyle doğrudan ilgilidir. Akciğer gibi hava içeren organlar düşük akustik empedansa sahipken, kemik gibi yoğunluğu yüksek yapılar çok yüksek akustik empedansa sahiptir (Çizelge 4.1). Bu nedenle, bu yapıların ardında daha derin yapıların görüntülenmesi zor olabilir. Yansıtılan bir yankının yoğunluğu, iki ortam arasındaki akustik empedans farkıyla doğru orantılıdır. Eğer iki dokunun akustik empedansı birbirine yakınsa, yankı genellikle oluşmaz. Benzer akustik empedanslara sahip yumuşak dokular arasındaki sınırlar genellikle düşük yoğunluklu yansımalar üretir. Ancak, yumuşak doku ile kemik veya akciğer gibi farklı akustik empedansa sahip bölgeler arasındaki sınırlar, büyük bir akustik empedans farkı nedeniyle çok güçlü yankılar üretebilir.



Şekil 4.3. Akustik dalgalarda yansıma ve kırılma olayı

Şekil 4.3'te bir akustik dalganın yoğunluğu farklı iki ortamda hareket ederken uğradığı yansıma ve kırılma olayı gösterilmiştir. Yansıyan ve kırılan dalgaların frekansı gelen dalganın frekansı ile aynıdır. Kırılma ve yansıma açıları Eş. 4.3'te gösterildiği gibi belirlenmektedir. Eşitlikteki  $c_1$  ve  $c_2$  ultrason hızının sırasıyla doku A ve doku B ile isimlendirilen dokulardaki ilerleme hızlarını,  $\theta_i$ ,  $\theta_r$  ve  $\theta_t$  sırasıyla ultrason hüzmelerinin geliş, yansıma ve kırılma açılarını ifade etmektedir.

$$\frac{\sin \theta_i}{c_1} = \frac{\sin \theta_r}{c_1} = \frac{\sin \theta_t}{c_2} \quad (4.3)$$

$Z_1$  ve  $Z_2$  olarak ifade edilen farklı akustik empedanslara sahip iki bitişik doku arasında oluşan yansıma miktarını ölçmek için yansıma katsayısı ( $\alpha_R$ ) kullanılır (Eş. 4.4). Ses dalgasının ortamda ilerleme enerjisi ve yansıma enerjisi sırasıyla  $I_i$  ve  $I_r$  olarak gösterilmiştir.

$$\alpha_R = \frac{I_r}{I_i} = \frac{\left(\frac{Z_2}{\cos\theta_t} - \frac{Z_1}{\cos\theta_i}\right)^2}{\left(\frac{Z_2}{\cos\theta_t} + \frac{Z_1}{\cos\theta_i}\right)^2} \quad (4.4)$$

Akustik yansıma olayı, US hüzmesinin akustik empedans farkı olan iki dokunun sınırına dik hareket etmesi sonucu oluşur. Dokular arasındaki akustik empedans farkı ne kadar yüksek ise, yansıma miktarı da o kadar yüksek olacaktır. Örneğin deri ve kemik arasındaki akustik empedans farkı yüksek olduğu için, kemiğe gelen ses dalgalarının büyük bir oranı geri yansıtılmaktadır. Kemik arkasında bulunan yapıları görüntülemeye yetecek kadar ses dalgası kalmadığı için bu bölgeler görüntülenememektedir. Yansıtılan ses dalgalarının oranı ne kadar yüksek ise, o kadar az ses dalgası dokunun geri kalanının görüntülemeye ya da daha derin yapıları görüntülemeye yetecek kadar kalır. Özellikle hava dolu olan akciğer ve kemik gibi sert yüzeylerin diğer dokularla olan akustik empedans farklarından dolayı arkalarında kalan yapıların görüntülenebilmesi kullanıcıya bağlıdır. Kullanıcının probu farklı açılarla yerleştirip en iyi görüntüleme alanının yaratması görüntüleme kalitesine etki etmektedir. US hüzmesinin geliş açısına kıyasla değişmeyen ya da çok az bir farkla yansıtılabilmesi için, ses dalgasını yansıtan yüzeyin gelen dalgaya dik ve pürüzsüz olması gerekir. Geri yansıyan dalganın yansıma açısı genellikle küçüktür. Bu nedenle yansıyan ses dalgası dönüştürücüye çarpmaktadır. Bu tür yansımaya ayna yansıma denir.

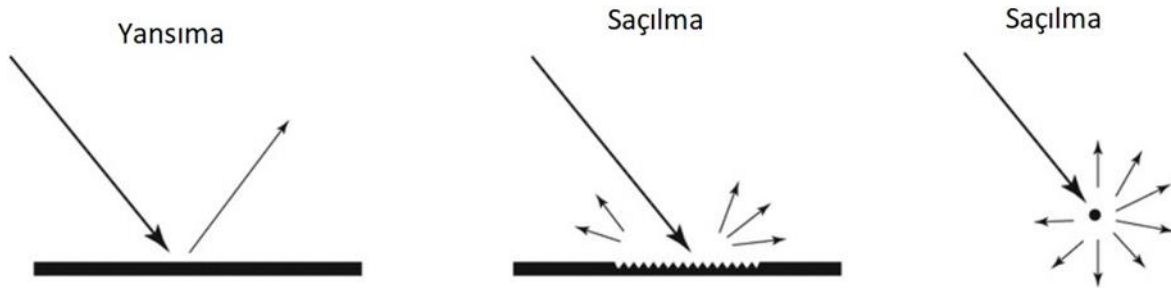
İletilen ultrason dalgasının doku yüzeyine geliş açısı önemlidir. İletilen ultrason dalgası doğrusal arayüze  $90^\circ$ 'ye yaklaşırsa, üretilen yankının neredeyse tamamı dönüştürücüye geri döner. Yüzeyle olan açısı büyük ise geri yansıyan ultrason dalgaları dönüştürücü tarafından tespit edilemeyecektir.

US hüzmelerinde kırılma olayı, hüzmelerin doku sınırına gelmemesi ve ses iletim hızlarının farklı olduğu iki dokuya iletilen ultrason darbesinin yönündeki değişikliktir. Tıpkı yansıma da olduğu gibi kırılma durumunda da ses hüzmesinin frekansı değişmez. Ses iletim hızı farklı olan bu iki ortamda hüzmelerin yayılma yönü ve genliği de değişmektedir. Ultrason

darbesinin düz bir hat yayılımı varsayıldığından, kırılma meydana geldiğinde görüntü oluşumu sırasında anatominin yanlış yerleşimi meydana gelebilir.

Akustik saçılma, ultrason dalgalarının bir dokuda bulunan nesnelere veya sınırlar tarafından yakalanması ve yansıtılması sonucu ortaya çıkar. Bu saçılma olayı, nesnelere ultrason dalgalarının dalga boyundan daha küçük veya ona yakın boyutlarda olduğu durumlarda meydana gelir. Düşük frekanslı ultrason (1-5 MHz) kullanıldığında, dalga boyları genellikle büyük olduğundan, dokular arasındaki sınırlar genellikle düz ve net bir şekilde görünür. Ancak daha yüksek frekansta (5-15 MHz), dalga boyları daha küçüldüğünden, sınırlar daha karmaşık hale gelir ve birçok yönde yansımalar oluşur. Her organın kendine özgü bir saçılma karakteristiği vardır ve dokudan dokuya geçişlerde saçılma genliği farklılıkları, ultrason görüntülerinde parlaklık değişikliklerine yol açar.

Saçılma olayı, ultrason frekansının artmasıyla güçlü bir şekilde değişir. Bu saçılma nedeniyle ultrason görüntülerinde benek benzeri gürültüler oluşabilir. Ancak özel filtreleme yöntemleri kullanılarak bu gürültüler azaltılabilir ve görüntü kalitesi artırılabilir. Şekil 4.4'te, akustik dalganın pürüzsüz bir yüzeye temas etmesi sonucu nasıl yansıdığını, pürüzlü bir yüzeye temas etmesi sonucu nasıl saçıldığını ve noktasal bir küçük yüzeye temas etmesi sonucu nasıl saçıldığını gösteren örnekler bulunmaktadır.



Şekil 4.4. Akustik dalgaların farklı ortamlarda yansımaya ve saçılmaya uğraması

Yansıma, geri dönüş sinyali için kullanılırken, saçılma, yansıyan sinyalin dağılımı hakkında kullanılır. Akustik zayıflama, gelen ışının saçılması ve soğurulmasından kaynaklanan, katedilen mesafeyle enerjinin kaybedilmesidir. Soğurma (absorpsiyon), doku yapısının ısınmasına veya mekanik olarak bozulmasına neden olan enerjinin dokulara aktarılmasıyla gerçekleşir. Saçılma ve absorpsiyonun birleşik etkileri, artan frekansın bir fonksiyonu olarak katedilen mesafe ile ultrason yoğunluğunun katlanarak zayıflamasına neden olur.

Logaritmik bir yoğunluk ölçüsü olan desibel (dB) cinsinden ifade edildiğinde, dB/cm cinsinden zayıflama, ultrason frekansı ile doğrusal olarak artar. Yumuşak dokuda ultrason zayıflama ortalaması için yaklaşık olarak MHz cinsinden frekansın 0,5 dB/cm katıdır. 1 MHz'lik bir hüzmeye ile karşılaştırıldığında, 2 MHz'lik bir hüzmeye yaklaşık olarak iki kat zayıflamaya sahip olacak, 5 MHz'lik bir hüzmeye beş kat zayıflamaya sahip olacak ve 10 MHz'lik bir hüzmeye kat edilen birim mesafe başına on kat zayıflamaya sahip olacaktır. Bu nedenle, daha yüksek frekanslı ultrason ışınları hızla azalan bir penetrasyon derinliğine sahiptir.

### **4.3. US Gri Tonlamalı Görüntüleme Modları**

Ekoların görüntülenmesi üç şekilde yapılmaktadır.

#### **4.3.1. A mod**

Darbenin görüntüleme alanına iletilmesinden hemen sonra dönüştürücü alıcı olarak görev yapar. Dönüştürücü tarafından yakalanan ekolar şiddetlerine göre zamanın bir fonksiyonu olarak kaydedilir. Ekoların şiddeti piklerin tepe değerlerini belirlerken, iki tepe arasındaki mesafe ise ekonun geri dönüş zamanını göstermektedir. Geri dönüş zamanı aynı zamanda yapıların vücut içerisindeki uzaklığı hakkında bilgi vermektedir. Yakalanan sinyaller genellikle radyo frekans (RF) sinyalleri olarak adlandırılır. Çünkü yakalanan sinyaller genellikle elektromanyetik spektrumda MHz aralığındadır. Günümüzde kullanım alanı diğer görüntüleme yöntemlerine kıyasla azalmıştır. Genellikle gözün incelenmesinde kullanılmaktadır.

#### **4.3.2. B mod**

İki boyutlu görüntüleme imkânı sunan parlaklık modu (brightness mode, B-mod) farklı dokulardan dönen ekoların şiddetine göre ekranda parlaklığı değişen noktalardan oluşan görüntüler oluşturmaktadır. Günümüzde en yaygın kullanılan görüntüleme modlarından biridir. Özellikle gebelik takibinde sıklıkla kullanılmaktadır.

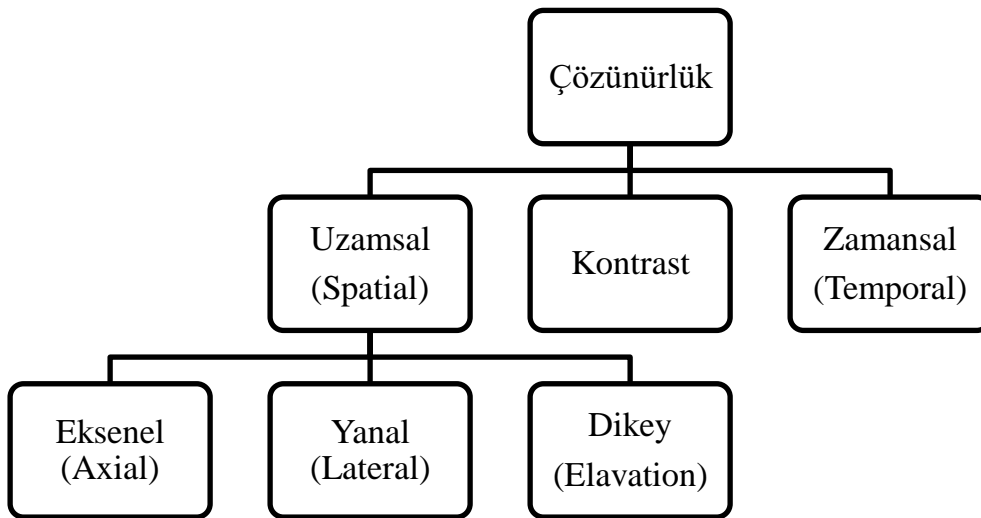
### 4.3.3. M mod

M mod görüntüleme (Motion, M), hareketli organların zaman içerisindeki hareketlerini incelemek için kullanılmaktadır. Gebelikte cenin hareketlerinin incelenmesi veya kalp kapakçığının hareketlerinin incelenmesi gibi durumlarda kullanılmaktadır.

## 4.4. US Görüntü Kalitesi

Ultrason görüntü kalitesi, yalnızca görüntüleme sistemine ait olmayan, aynı zamanda operatörün becerisi ve performansı gibi birçok faktörün birleşiminden kaynaklanan nihai bir sonuçtur [84-87]. Bu kaliteyi etkileyen temel faktörler, dönüştürücü, görüntü işleme süreci, görüntüleme ve kayıt cihazları gibi görüntüleme sisteminin tüm bileşenlerini içerir.

Ultrason görüntü kalitesi, genellikle yüksek bir çözünürlükle ilişkilendirilir. US görüntüleme çözünürlük, görüntülenen organ ve yapıların ayırt edilebilme veya ayrıntıların iyi gösterilmesi yeteneğini ifade eder. US görüntüleme çözünürlüğün tanımlanmasında farklı parametreler kullanılmaktadır. Bir ultrason görüntüsünün kalitesini etkileyen temel faktörler; uzamsal çözünürlük (spatial), zamansal (temporal) çözünürlük ve kontrast (contrast) çözünürlükleridir. Bunların dışında, görüntülenen dokunun yapısı, operatörün cihazı kullanma yeteneği ve çeşitli artefaktlar da nihai görüntülerin kalitesini etkilemektedir. Şekil 4.5'te US görüntü çözünürlüğüne etki eden faktörler gösterilmiştir.



Şekil 4.5. US görüntülerinde çözünürlük

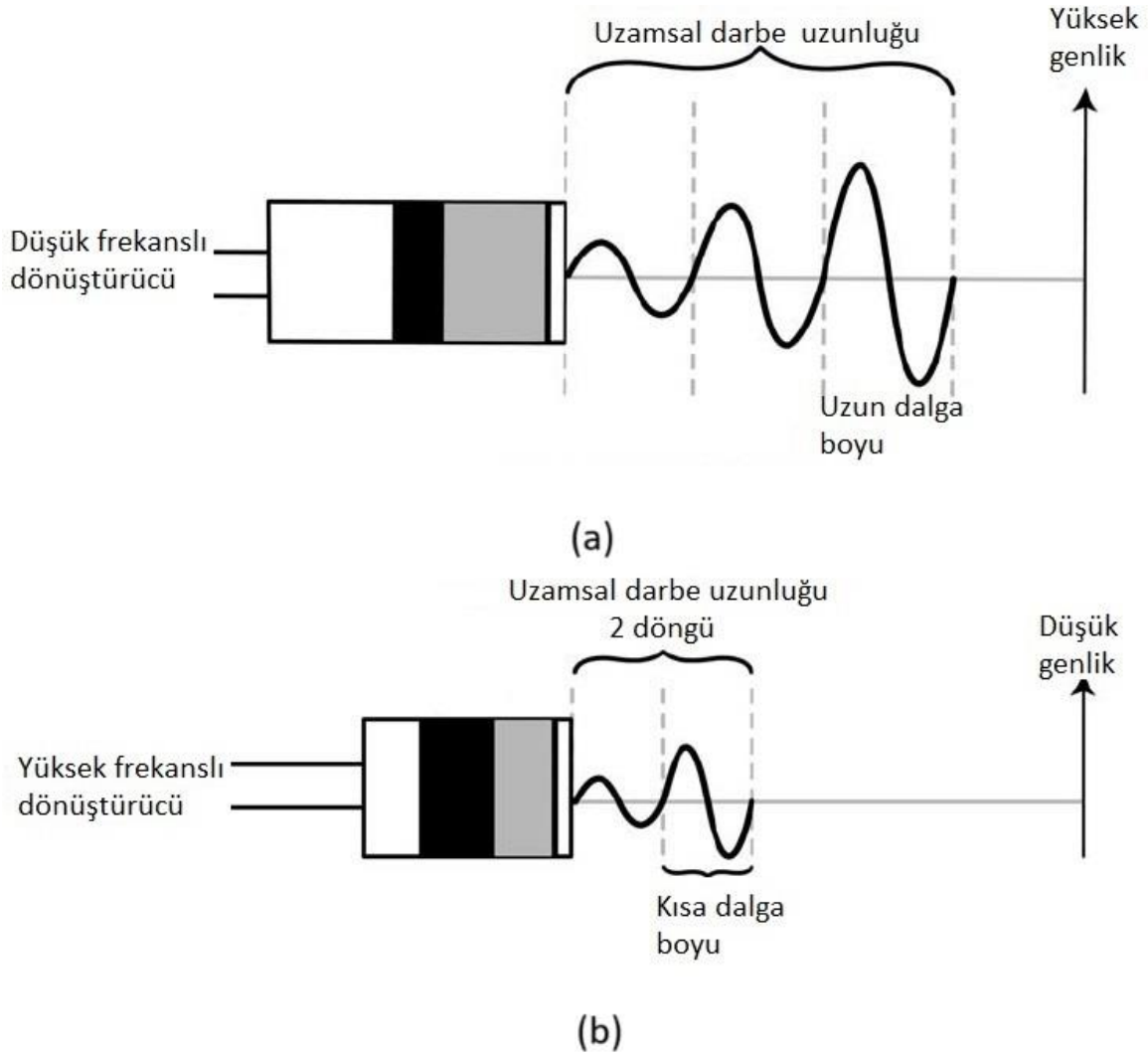
#### 4.4.1. Uzamsal çözünürlük

Birbirine yakın konumdaki iki nesneyi farklı yapılar olarak ayırt etme yeteneğini ifade eder. Uzamsal çözünürlük ne kadar yüksek olursa, algılanabilecek mesafe de o kadar küçük olur ve birbirine yakın nesnelere ayırt etme yeteneği yüksek olur. Uzamsal çözünürlük, her biri için farklı çözünürlük belirleyicileri olan üç düzlemden oluşur. Bunlar aksel (axial), yanal (lateral veya azimuthal çözünürlüğü de denir) ve dikey (slice thickness veya elevational resolution) çözünürlüklerdir.

Eksenel çözünürlük, ultrason hüzmesinin ekseni boyunca olan çözünürlüğünü ifade eder. Hüzme yayılma yönü boyunca yakın aralıklı nesnelere ayırt etme yeteneğini ifade eder. Matematiksel olarak aksel çözünürlük, uzamsal darbe uzunluğunun (spatial pulse length, SPL) yarısı kadardır. Yani uzamsal darbe uzunluğunun kısa olması, aksel çözünürlüğün yüksek olması anlamına gelir.

Kristale sürekli bir elektrik sinyali uygulamak yerine, uzamsal bilgi elde etmek için darbeler kullanılır. Bu yapıda olan dönüştürücülere darbeli ultrason (pulsed US) denir. Darbe tekrarlama frekansı (DTF), dönüştürücü tarafından birim zamanda yayılan darbelerin sayısıdır. Tıbbi görüntüleme cihazları için DTF, 1 ila 10 kHz arasında değişmektedir.

Ultrason görüntülemesinde SPL, bir ultrason darbesinin uzayda kapladığı sürenin uzunluğunu belirtmektedir. SPL'nin yarısından daha yakın mesafede bulunan dokuların birbirinden ayrılması zor olabilir. Eksenel çözünürlük derinlikten etkilenmez, ancak daha yüksek frekans ve daha kısa atım uzunluğu ile iyileştirilir. Matematiksel olarak, bir darbedeki döngü sayısının ve dalga boyunun ürünüdür. Daha kısa uzamsal darbe uzunluğu, daha yüksek aksel çözünürlükle sonuçlanır. SPL'nin azaltılabilmesi için daha ağır sönümlenme veya daha yüksek frekanslı bir dönüştürücü kullanılabilir. Piezoelektrik materyali tarafından belirlenen döngü sayısı her darbede genellikle iki ya da üçtür. Yüksek aksel çözünürlük Eş. 4.6'da gösterildiği gibi kısa dalgaboyu/yüksek frekansla sağlanabilir. Şekil 4.6'da düşük frekanslı ve yüksek frekanslı dönüştürücülerin ürettikleri uzamsal darbeler gösterilmiştir.



Şekil 4.6. Dönüştürücü ve üretilen darbelerin özellikleri (a) düşük frekanslı dönüştürücü ve uzun SPL, (b) yüksek frekanslı dönüştürücü ve kısa SPL [55]

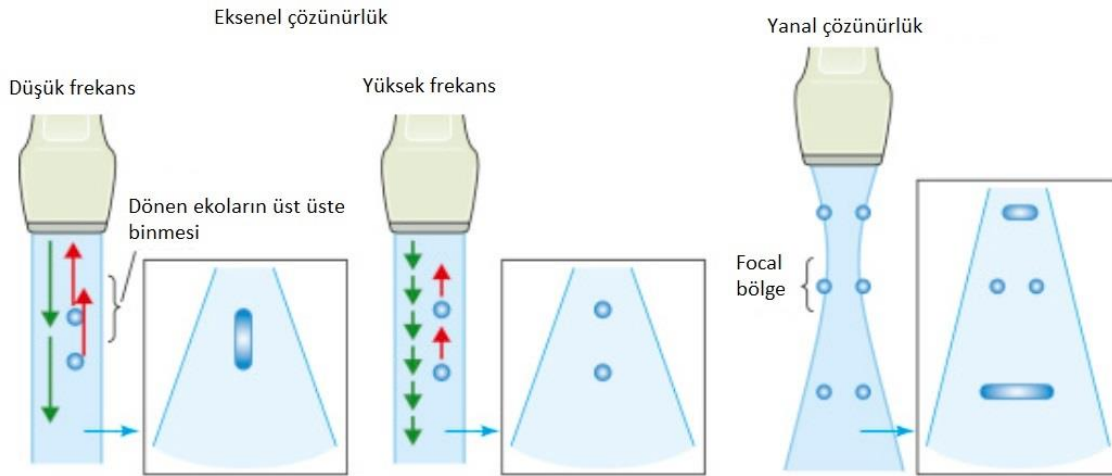
$$\text{Uzamsal darbe uzunluğu} = \text{dalgaboyu} \times \text{bir darbedeki döngü sayısı} \quad (4.5)$$

$$\text{Eksenel çözünürlük} = \frac{1}{2} \text{uzamsal darbe uzunluğu} \quad (4.6)$$

Uzamsal çözünürlüğü belirleyen ikinci önemli faktör olan ve bazen azimut çözünürlüğü olarak adlandırılan yanal uzamsal çözünürlük, ultrason eksenine dik iki noktayı yan yana ayırt etme yeteneğini ifade eder. Yanal çözünürlük, piezoelektrik kristallerin çapı ve frekansından etkilenen ultrason ışınının genişliği ile belirlenir. Yüksek frekanslı darbeler üreten küçük çaplı kristaller, dar ultrason ışınları üretir ve böylece yanal çözünürlüğü artırır. Ultrason hüzmeleri, dönüştürücü yüzeyinden yayılma ortamına uzunlamasına bir dalga

olarak yayılır ve ses kaynağına olan uzaklığa bağlı olarak ses hüzmeleri farklı davranış sergilemektedir. Bu davranışlar genellikle iki farklı bölgeye ayrılarak incelenmektedir. Bunlar; yakın alan (near-field) ve uzak alan (far-field) [86]. Dönüştürücünün şekli ve frekansı tarafından belirlenen ve hüzmelerin yakınsadığı bölgeye yakın-alan denilmektedir. Hüzmelerin belirli bir noktadan sonra ıraksadığı bölgeye ise uzak-alan denilmektedir. Her iki alanda ses kaynağına olan fiziksel uzaklıkla ilgili terimlerdir.

Ultrason ışınının en dar kısmı veya odak bölgesi en yüksek yanal çözünürlüğe sahiptir. Ses hüzmesi uzak alanda ıraksadığından, yanal çözünürlük artan derinlikle azalır. Yani yanal çözünürlük aksel çözünürlüğün aksine, nesneye olan mesafeye bağlıdır. Ayrıca, daha geniş bir hüzmeye de (örneğin daha geniş prop) yanal çözünürlükte bir azalmaya yol açacaktır. Ultrason hüzmesinin genişliği dar olduğunda yanal çözünürlük yüksektir. Aksel çözünürlükle karşılaştırıldığında, yanal çözünürlük daha az güvenilirdir. Görüntüleme derinliğinden ve ultrason hüzmesinin genişliğinden büyük ölçüde etkilenir. Şekil 4.7’de aksel ve yanal çözünürlüklerin etkisi gösterilmiştir.



Şekil 4.7. US görüntüleme de frekansın aksel çözünürlüğe etkisi ve hüzmeye şeklinin yanal çözünürlüğe etkisi [88]

#### 4.4.2. Zamansal çözünürlük (frame rate, çerçeve hızı)

Vücutta görüntülenmek istenen bölgeler, ultrason makinesinin ekranında zaman içinde ardışık kareler halinde iki veya üç boyutlu olarak görüntülenir. Zamansal çözünürlük, bir nesnenin konumunu zamanın belirli bir anında tam olarak belirleme yeteneğini ifade eder. Çözünürlüğün diğer iki alt kategorisinden farklı olarak, hertz cinsinden ölçülür ve genellikle

çerçeve hızı olarak da isimlendirilir. Ultrason dönüştürücüler, ardışık birden çok kareyi taramak ve bir nesnenin zaman içindeki hareketini gözlemlemek için zamansal çözünürlüğü kullanır. Özellikle hareketli organlar gibi kalp gibi yapıların incelemelerinde kapakların ve hızlı atışlı kalp yapısının daha iyi görselleştirilmesi için daha hızlı kare hızlarına ihtiyaç vardır. Zaman içindeki ayrıntıları daha iyi yakalamak ve görüntüdeki çizgi yoğunluğunu azaltmak ve sektör açısını azaltmak gibi yöntemler kullanılarak, zamansal çözünürlük derinliği artırılabilir. Bu durumda alınan çerçeve oranı artacaktır. Eş. 4.7’de c; sesin dokulardan geçiş hızını, N; 2 boyutlu B-mod görüntü oluşturmak için kullanılan A çizgileri tarama sayısını, D ise tarama derinliğini ifade etmektedir. Ultra ses hüzmesi gidip döndüğü için bu mesafenin hesaplanmasında x2 kullanılmıştır.

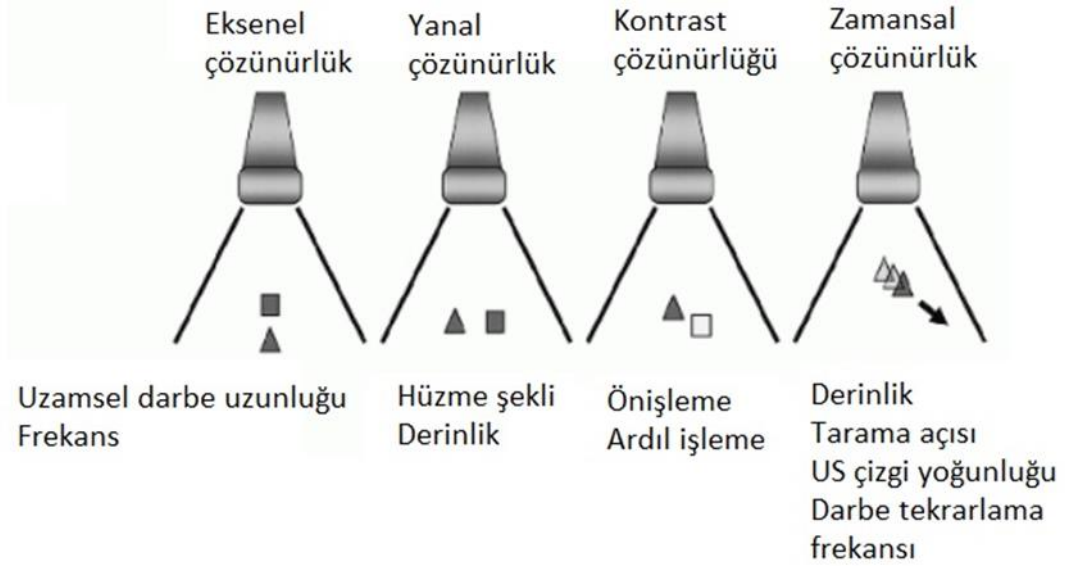
$$\text{Çerçeve hızı} = \frac{c}{N_{(A_{\text{lines}})} \times 2 \times D_{\text{cm}}} \quad (4.7)$$

Eş. 4.7’te göre yüksek zamansal çözünürlük A tarama çizgi sayısının ve tarama derinliğinin azaltılması ile artmaktadır.

Eksenel, yanal ve zamansal çözünürlüklerin tümü, kullanılan dönüştürücü dizisinin tipine ve yapısına bağlıdır. Dönüştürücü dizisinin frekansı düşükçe aksenal ve yanal çözünürlük azalır. Bununla birlikte, ultrason hüzmesinin penetrasyonu artar. Çözünürlük, sistem üzerindeki kullanıcı kontrolleri ile bir ölçüde artırılabilir. Ancak mutlak aksenal, yanal ve zamansal çözünürlük her zaman dönüştürücü dizisinin yapısına bağlıdır.

#### 4.4.3. Kontrast çözünürlüğü

Kontrast çözünürlüğü, farklı yoğunluktaki nesnelere veya bitişik yapıların farklı yankı genliklerini ayırt etme yeteneğini ifade eder. Diğer bir deyişle, ekranda farklı boyutlardaki yankıları temsil eden noktaların yoğunlukları arasında ayırım yapmak anlamına gelir Bu, yansıma gücüne (geri dönen sinyal) ve görüntü gürültüsüne bağlıdır. Görüntüleme sistemindeki elektronik aksamlar, görüntüleme, kayıt cihazlarının doğal kontrast özellikleri kontrast çözünürlüğünü büyük ölçüde etkiler. Diğer US çözünürlük modlarından farklı olarak kontrast çözünürlüğü sayısal görüntü işleme tekniklerinin kullanımı US görüntülerinin kontrastını iyileştirmek mümkündür. Şekil 4.8’de US görüntü çözünürlüğünün değerlendirilmesinde kullanılan dört tane çözünürlük faktörü ve bunlara etki eden parametreler özetlenmiştir [81].



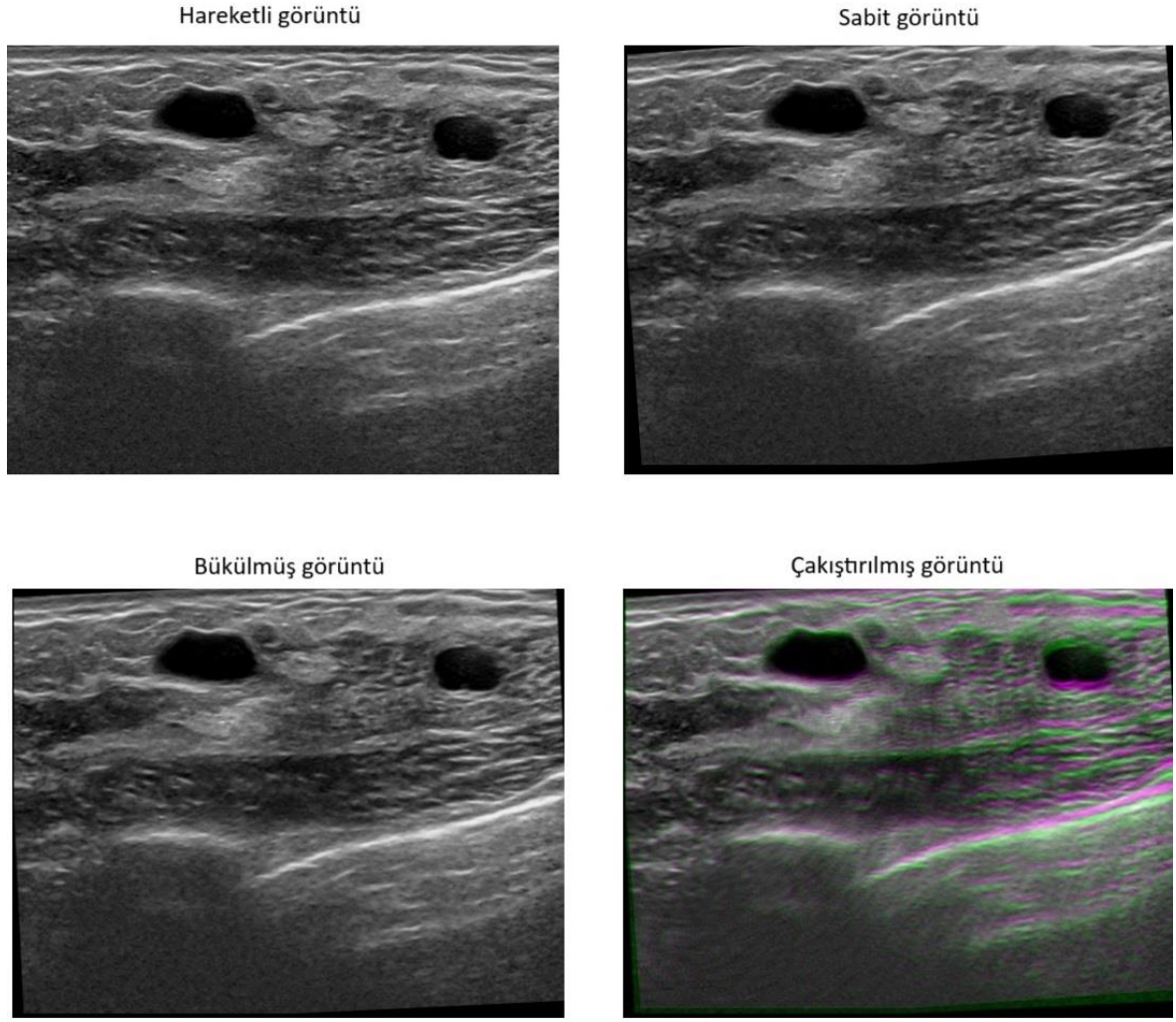
Şekil 4.8. US görüntüleme çözünürlükleri ve bu çözünürlükleri etkileyen faktörler [81]

## 5. MEDİKAL GÖRÜNTÜ ÇAKIŞTIRMA

Medikal görüntüler özellikle tıbbi uygulamalarda hayati önem taşıyabilen birçok uygulamada kullanılmaktadırlar. Genel uygulamalar olarak hastalık teşhisi, cerrahi ve radyoterapi prosedürlerinin planlanması, yürütülmesi ve değerlendirilmesi alanlarında yapılmaktadır [89-93]. Klinik takiplerde bir yerine birden fazla görüntüden elde edilen bilgiler tamamlayıcı nitelikte olduğundan, genellikle ayrı görüntülerden elde edilen faydalı verilerin uygun şekilde birleştirilmesi istenir. Bu süreç için ilk adım olarak genellikle, görüntü çakıştırma denilen görüntüleri uzamsal olarak hizalandığı yöntemdir. Görüntü çakıştırmadan sonraki adım ise birleştirme (fusion) işlemidir.

Görüntü çakıştırma (GÇ), görüntü işleme alanındaki en temel konulardan biridir [94, 95]. GÇ, ilgilenilen bölgenin görüntülerinin farklı yöntemlerle elde edilmesi, farklı açılardan veya mesafelerden elde edilmesi, farklı yöntemlerle elde edilmesi, farklı zamanlarda elde edilmesi gibi nedenlerle yapılmaktadır. GÇ işlemi başta tıp ve radyoloji olmak üzere savunma sanayi, uzaktan algılama, haritacılık, bilgisayarlı görü gibi birçok alanda kullanılmaktadır. Medikal GÇ, hastalık teşhisi ve izleme ve görüntü kılavuzluğunda operasyon gibi tıbbi görüntü işleme alanında çok yaygın bir uygulamadır [96, 97].

GÇ işlemi aynı sahnenin iki veya daha fazla görüntüsünü belirli bir görüntüye göre hizalama işlemidir. Giriş görüntülerinden biri referans veya sabit görüntü olarak tanımlanır. Diğer görüntüler ise hareketli görüntü olarak tanımlanır. Hareketli görüntü, katı, afın veya deforme gibi çeşitli dönüşümler uygulanarak hedefe uyacak şekilde uzamsal olarak dönüştürülür. GÇ işleminde amaç, hareketli ve sabit görüntü çiftlerini tek bir koordinat sistemine dönüştürmektir. Şekil 5.1’de sabit, hareketli görüntü, hareketli görüntüye dönüşüm matrisi uygulandıktan sonra elde edilen bükülmüş görüntü ve bükülmüş görüntü-sabit görüntünün üst üste çakıştırılmasıyla elde edilen görüntü gösterilmiştir.

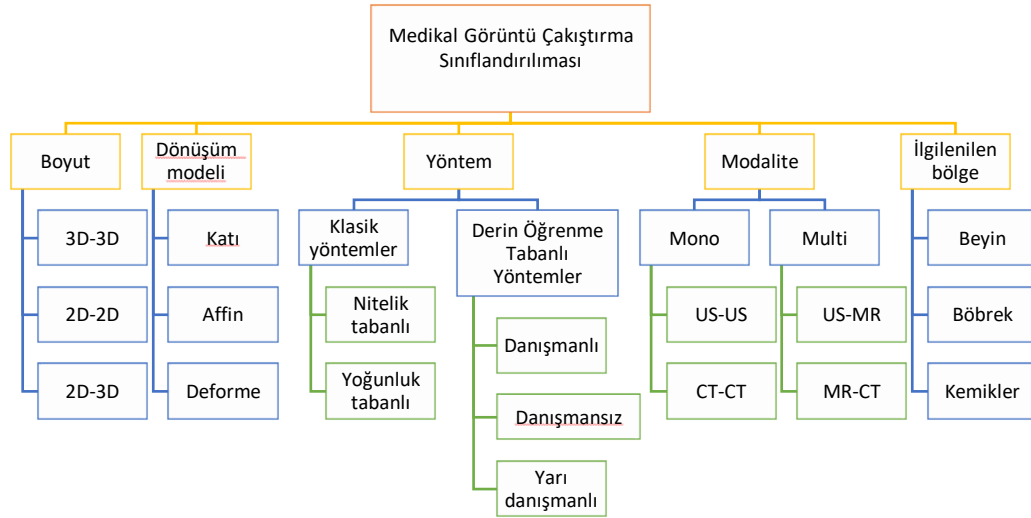


Şekil 5.1. Görüntü çakıştırma işleminde kullanılan sabit, hareketli, bükülmüş ve çakıştırılmış görüntüler

### 5.1. Medikal Görüntü Çakıştırma Sınıflandırması

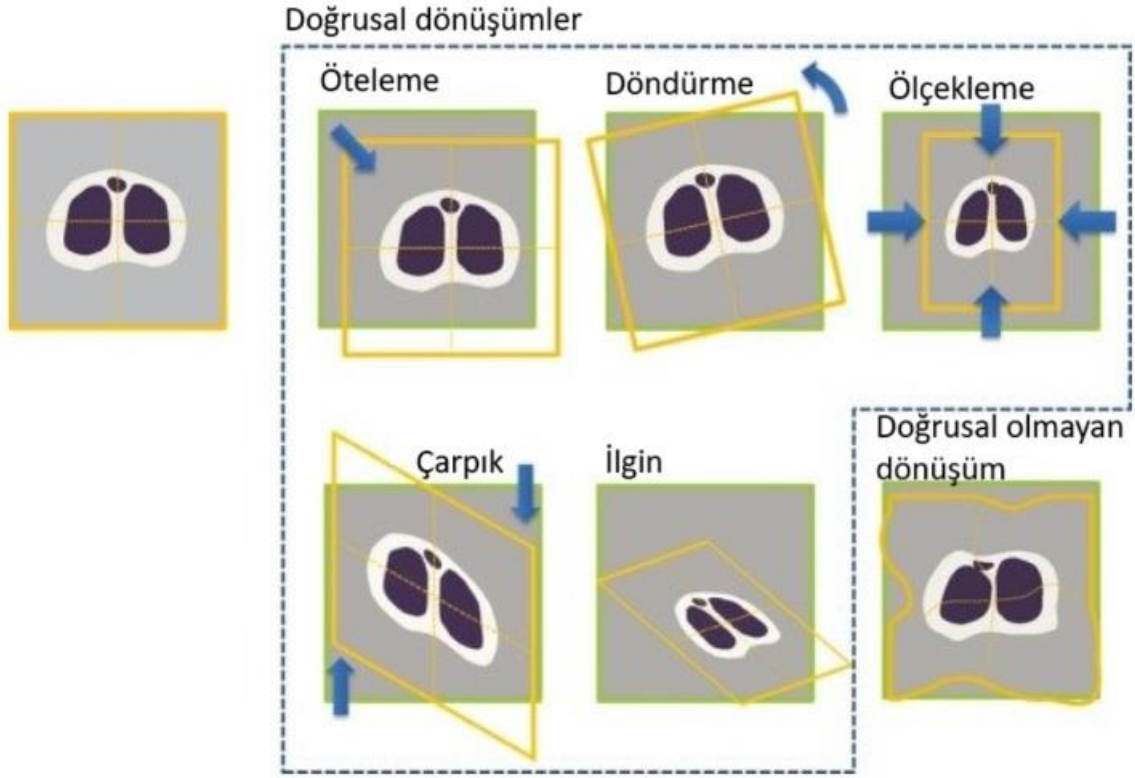
Medikal görüntü çakıştırma yöntemleri birçok şekilde kategorize edilebilir. En sık kullanılan sınıflandırma türleri Şekil 5.2'de gösterilmektedir. Görüntü çakıştırma yaklaşımlarında kullanılan geometrik dönüşümler; katı (rigid) dönüşüm (döndürme, öteleme), afin dönüşüm (makaslama, ölçekleme gibi) ve esnek (non-rigid ya da deformable) dönüşümlerdir [100]. Katı çakıştırmada temel olarak belirli bir açıyla döndürme ( $\theta$ ) ve x-y eksenlerinde öteleme ( $dx$ ,  $dy$ ) hareketleri söz konusudur. Farklı dönüşüm yöntemlerinin uygulaması Şekil 5.3'te gösterilmiştir.

Medikal GÇ, görüntü modalitesine göre mono-modalite veya multi-modalite olabilir. Mono-modalite aynı kaynaktan aynı yöntemle alınan görüntüler için kullanılır. Örneğin, US-US veya MR T1- MR T1. Multi-modalite ise farklı kaynaklardan veya aynı kaynaktan farklı yöntemlerle alınan görüntüler için kullanılır. Örneğin, US-MR veya MR T1-T2 görüntülerinin karşılaştırılması. Mono-modalite GÇ işlemi genellikle, aynı bölgenin farklı açılardan alınan görüntülerini birleştirerek veya yan yana getirerek gözlemlenen alan hakkında ayrıntılı veya daha yüksek çözünürlüklü bilgi elde etmek için kullanılır. Bu yöntem, özellikle US görüntüleme, ilgilenilen bölgenin farklı açılardan alınan görüntülerin birleştirilip daha ayrıntılı bilgi elde etmek için kullanılabilir. Çakıştırılacak görüntüler 2D-2D, 3D-3D veya 2D-3D çiftlerinden oluşabilir. Özellikle bazı cerrahi işlemler öncesi MR ile elde edilmiş 3D görüntüler ile cerrahi müdahale sırasında elde edilen US görüntülerinin karşılaştırılması için 2D-3D karşılaştırma yöntemleri kullanılır. GÇ, çakıştırılacak bölgeye göre de sınıflandırmak mümkündür. Örneğin, beyindeki bir tümörün veya göğüste bulunan bir kitlenin ilerleme/televi süreçlerinin kontrolü için beyin MR görüntü karşılaştırma veya US görüntü karşılaştırma yöntemleri kullanılabilir.



Şekil 5.2. Medikal görüntü çakıştırma sınıflandırılması

Çakıştırma işlemi bütün görüntüye uygulanıyorsa global, görüntünün parçalarının her birine ayrı ayrı dönüşüm modeli tanımlanıyorsa yerel çakıştırma denir. Yerel görüntü çakıştırma nadir kullanılmaktadır. Genellikle global görünüş çakıştırma kullanılmaktadır.



Şekil 5.3. Görüntü çakıştırma için kullanılan farklı geometrik dönüşüm yöntemleri [99]

## 5.2. Derin Öğrenme Tabanlı Medikal Görüntü Çakıştırma

Görüntü çakıştırma uygulamaları ilk zamanlarda klasik yöntemler kullanılarak yapılmaktaydı [100]. Klasik yöntemde öznitelik tabanlı ve yoğunluk tabanlı olmak üzere iki yöntem kullanılmaktadır. Yoğunluk tabanlı yöntemde, görüntülerde bulunan piksellerle işlem yapılmaktadır. Öznitelik tabanlı yöntemde ise, pikseller yerine görüntülerde seçilen özniteliklerle (noktalar, köşeler, çizgiler veya kontürler) işlemler yapılmaktadır. Çakıştırma işlemi bu özniteliklerin eşleştirilmesine dayanmaktadır.

Klasik görüntü çakıştırma dört adımdan oluşmaktadır. Bunlardan ilki, özellik algılama işlemidir. Çakıştırma için kullanılacak özellikler için iki yöntem vardır. Birincisi yoğunluk tabanlı, ikincisi ise görüntüdeki belirli noktaların seçilmesi ve bu noktaların kaydedilmesidir. İkinci adım özellik eşleştirmedir. Üçüncü adım, dönüşüm modeli tahminidir ve son adım, görüntü dönüşümü veya görüntünün yeniden örneklenmesidir. Bu adımda hareketli görüntü, dönüştürme parametrelerine göre dönüştürülür. Genel olarak, klasik görüntü çakıştırma yinelemeli bir deneme yanılma

prosedürü olarak tanımlanabilir. Her yinelemede görüntüler arasında bir benzerlik veya mesafe ölçüsü hesaplanır ve yeterince hizalanıp hizalanmadıklarını belirlemek için kullanılır.

Klasik görüntü çakıştırma üç önemli bileşenden oluşur. Bunlar benzerlik metrikleri, optimizasyon algoritmaları ve geometrik dönüşüm modelleridir. Benzerlik metrikleri kullanılan giriş bilgisinin öznitelik tabanlı olması veya yoğunluk tabanlı olmasına göre ikiye ayrılır. Bazen bu iki yöntem için kullanılan ortak değerlendirme metrikleri de kullanılmaktadır. Ayrıca çakıştırma yönteminin katı veya katı olmayan yöntemler olmasına göre de kullanılan benzerlik metrikleri değişmektedir. Yoğunluk tabanlı yöntemlerde kullanılan başlıca benzerlik metrikleri karelerin farkının toplamı (sum of squared difference, SSD), çapraz korelasyon (cross correlation) ve bunların türevleridir. Özellikle mono-model çakıştırma problemleri için bu metrikler kullanılmaktadır. Bu yöntemlerin yanı sıra, ortak bilgi (mutual information, MI) değerlendirme metriği ve bu metriğin türevi olan normalize MI (NMI) metrikleri de klasik tabanlı yöntemlerde sıklıkla kullanılmaktadır. Öznitelik tabanlı çakıştırma yöntemlerinde kullanılan metrikler ise SSD, yinelemeli en yakın nokta (iterative closest point, ICP) ve MI metrikleridir.

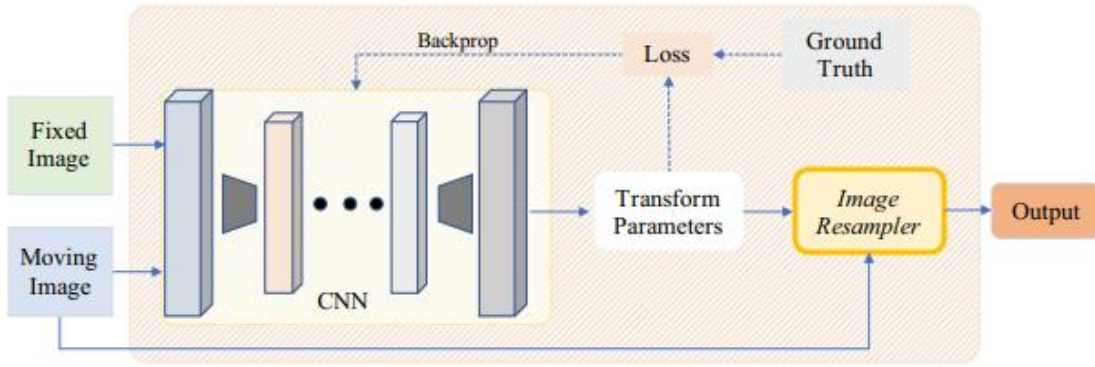
Eniyileme algoritmalarının kullanım amacı, benzerlik metriğini maksimum veya minimum edecek değerlerin bulunmasıdır. Eşitlik 5.1, görüntü çakıştırma işleminin bir optimizasyon problemi olarak ele alınabileceğini göstermektedir. Eşitlikteki  $I_M$  ve  $I_F$  sırasıyla hareketli görüntüyü ve sabit görüntüyü temsil etmektedir.  $T$ ,  $I_F$  ve  $I_M$ 'yi hizalayan bir dönüştürme işlevini göstermektedir.  $T(I_M)$ ,  $T$  dönüşümü kullanılarak  $I_M$ 'den oluşturulan bükülmüş görüntüyü göstermektedir.  $S()$  fonksiyonu ise,  $I_F$  ve  $T(I_M)$  arasındaki benzerliği değerlendirmekte kullanılan benzerlik fonksiyonunu temsil etmektedir. Sonuç olarak görüntü çakıştırmanın amacı, maliyeti hesaba katarak görüntü benzerliğini maksimize eden  $T$ 'yi bulmaktır.

$$\hat{T} = \operatorname{argmax}_T S(I_F, T(I_M)) \quad (5.1)$$

Klasik yöntemin dezavantajı iteratif bir süreç olmasıdır. Bu nedenle, özellikle gerçek zamanlı uygulamalar için klasik yöntemlerin uygulanabilirliği pek olası değildir [101]. Bu nedenle alternatif olarak, derin öğrenme tabanlı yöntemler araştırılmaya başlanmıştır. Klasik yöntemlerin yanı sıra son zamanlarda derin öğrenme yöntemleri de görüntü çakıştırma

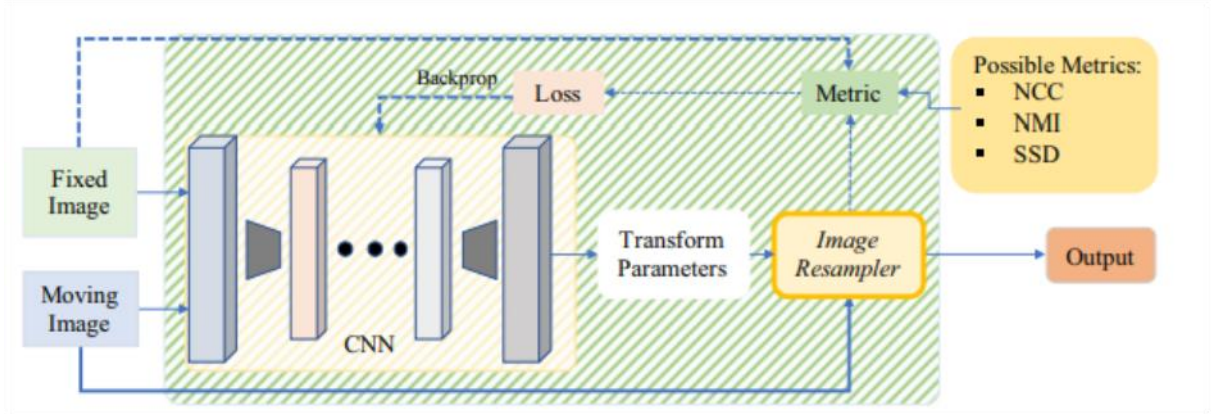
problemlerinde başarıyla uygulanmaktadır. ESA tabanlı yaklaşımlar danışmanlı ve danışmansız olarak iki gruba ayrılabilir.

Danışmanlı sistemlere örnek olarak Şekil 5.4'te bir model gösterilmiştir [102]. Bu modelde girişler sabit ve hareketli olmak üzere iki görüntüdür. Ağın eğitimi sırasında gerçek referans değerler (ground truth) ile hata fonksiyonu hesaplanmaktadır. Ağın çıkışında üretilen dönüşüm parametreleri ile hareketli görüntüye ters dönüşüm uygulanmakta ve böylece çıkış görüntüsü elde edilmektedir. Danışmanlı öğrenme yöntemi kullanılmasından dolayı gerçek referans değerlere ihtiyaç vardır. Gerçek referans değerlerin klinik verileri modellenmesi, geliştirilen ağın performansını da etkilemektedir. Bu açıdan gerçek referans değerlerinin tüm olasılıkları kapsamaması ağın performansının iyi olmamasına neden olabilir.



Şekil 5.4. Danışmanlı öğrenme yöntemi kullanılarak yapılan görüntü çakıştırma modelinin genel yapısı [102]

Danışmansız sistemlere örnek olarak Şekil 5.5'te bir derin ağ modeli gösterilmiştir [102]. Bu ağ modelinde gerçek referans değerler kullanılmamaktadır. Burada hata fonksiyonu hesaplanması metriklere dayalı olarak yapılmaktadır. Ağın tahmin ettiği parametreler hareketli görüntüye uygulandıktan sonra bükülmüş görüntü elde edilmektedir. Bükülmüş görüntü ile sabit görüntü arasındaki benzerlik belirlenen bir benzerlik metriği ile yapılmaktadır. Ağın eğitimi sırasında kayıp fonksiyonunun hesaplanması, bu seçilen metriği maksimum veya minimum yapmaya dayalı olarak yapılmaktadır. En çok kullanılan metrikler: ortak veya karşılıklı bilgi (mutual information, MI), normalize edilmiş çapraz korelasyon (Normalized Cross Correlation, NCC), normalize edilmiş ortak bilgi (normalized mutual information, NMI), farkın karelerinin toplamı (sum of squared differences, SSD) metrikleridir. Bu yöntemde seçilecek benzerlik metriği ağın performansın büyük ölçüde belirleyen bir parametredir.



Şekil 5.5. Danışmansız öğrenme yöntemi kullanılarak yapılan görüntü çakıştırma modelinin genel yapısı [102]

### 5.3. Kullanılan Dönüşüm Yöntemi

Dönüşüm yöntemleri, hareketli görüntünün noktalarını dönüştürülmüş görüntüdeki yeni konumlara eşleştirme için kullanılmaktadır. Çakıştırma problemine göre, dönüşüm işlemi katı veya deforme olabilir. Doğrusal dönüşümler katı, afin ve izdüşümdür. Katı dönüşüm yalnızca öteleme ve döndürmeyi içerirken, afin dönüşüm aynı zamanda ölçekleme ve kesmeyi de içermektedir [71]. Genel olarak kullanılan bazı önemli dönüşümler Çizelge 5.1’de gösterilmiştir. Tabloda gösterilen serbestlik derecesi (degree of freedom, DOF), çakıştırma işleminde dönüşüm matrisini tanımlayan bağımsız parametrelerin sayısıdır. Öteleme dönüşüm parametreleri iki tane parametreye sahiptir. Bunlar dx ve dy’dir. Dolayısıyla DOF bu işlem için iki olacaktır.

Çizelge 5.1. Görüntü dönüşüm yöntemleri ve matris formları

Görüntü modeli ve DOF	Matris Formu	Meydana gelen değişimler
Öteleme (translation) 2 DOF (dx ve dy)	$\begin{bmatrix} 1 & 0 & dx \\ 0 & 1 & dy \\ 0 & 0 & 1 \end{bmatrix}$	Sadece öteleme işlemi
Döndürme 1 DOF ( $\theta$ )	$\begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$	Sadece döndürme işlemi
Öklid (Euclidean) veya katı 3 DOF (öteleme için dx ve dy, döndürme için $\theta$ )	$\begin{bmatrix} \cos\theta & -\sin\theta & dx \\ \sin\theta & \cos\theta & dy \\ 0 & 0 & 1 \end{bmatrix}$	Hem öteleme hem de döndürme işlemi
Benzeşim (Similarity) 4 DOF ( Ölçekleme için ölçek faktörü (s), döndürme için ( $\theta$ ), öteleme işlemi için dx ve dy)	$\begin{bmatrix} s \cos\theta & -s \sin\theta & dx \\ s \sin\theta & s \cos\theta & dy \\ 0 & 0 & 1 \end{bmatrix}$	Öteleme, döndürme ve ölçekleme
İlgün (Affine) 6 DOF (2 ölçek, 2 dönüklük ve 2 öteleme)	$\begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ 0 & 0 & 1 \end{bmatrix}$	Öteleme, döndürme, ölçekleme ve çarpıklık

Çizelge 5.1. (devam) Görüntü dönüşüm yöntemleri ve matris formları

İzdüşümsel (Projective) 8 DOF	$\begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix}$	İlgin dönüşümde kullanılan dönüşümlere ek olarak eğme işlemi
----------------------------------	--	--

#### 5.4. Medikal Görüntü Çakıştırma Değerlendirme Ölçütleri

Bu bölümde çalışmada kullanılan değerlendirme ölçütleri gösterilmiştir. Medikal görüntü çakıştırma uygulamalarında genellikle TRE, Dice, HD vb. değerlendirme metrikleri kullanılmaktadır. Fakat henüz yapılan bütün çalışmaları kapsayıcı bir standart belirlenmemiştir. Bunun nedeni, medikal görüntü çakıştırma uygulamalarında kullanılan yöntemlerin, görüntülerin, modalitelerin farklı olmasından kaynaklanmaktadır [100]. Bu çalışmanın amacı donanım tabanlı ESA algoritmasının gerçekleşmesi olduğu için, bahsedilen metrikler yerine aşağıda gösterilen metrikler kullanılmıştır. Amaç, yazılım tabanlı uygulamadan elde edilen değerlerin donanım tabanlı uygulama ile kıyaslamak olduğu için, uygulamaların sonuçlarını kıyaslayan metrikler kullanılmıştır.

##### 5.4.1. Ortalama kare tatası (mean squared error, MSE):

Eş. 5.2’de gösterilen MSE, tahmin edilen değer ile hedef değer arasındaki kare farkının ortasını verir. Burada  $y_i$  tahmin edilen değer,  $y_i$  de hedef değeri göstermektedir. N kullanılan örnek sayısını göstermektedir.

##### 5.4.2. Ortalama hata kareleri kökü (root mean square error, RMSE):

Eş. 5.3’te gösterilen RMSE, MSE değerinin karekökünün alınmasıyla hesaplanmaktadır.

##### 5.4.3. Ortalama mutlak hata (mean absolute error, MAE):

Eş. 5.4’te gösterilen MAE, tahmin edilen değer ile hedef değer arasındaki farkın mutlak değerinin, kullanılan örnek sayısına oranıyla bulunur. Burada  $y_i$  tahmin edilen değer,  $x_i$  hedef değer n ise örnek sayısıdır.

##### 5.4.4. R-kare (R-squared):

R-kare, Eş. 5.5'te gösterildiği gibi regresyon çizgisinin orijinal verilere ne kadar iyi uyduğunu değerlendirir. R-kare değerleri 0 ile 1 arasında değişmektedir. Yüksek bir R-kare değeri, iyi bir regresyon modeli uyumuna işaret etmektedir.

İkinci değerlendirme yönteminde ise sabit-bükülmüş görüntü çifti arasındaki benzerlik ölçülmektedir. Ağın çıkışında oluşturulan geometrik dönüşüm değerlerinin hareketli görüntüye uygulanmasıyla elde edilen bükülmüş görüntü ile sabit görüntü arasındaki benzerliği ölçmek için şu değerlendirme ölçütleri kullanılmıştır:

#### 5.4.5. Yapısal benzerlik ölçütü (structural similarity, SSIM):

Eş. 5.7'de gösterilen SSIM, benzerliği parçaya ayırarak yapar. Bunlar; parlaklık ( $l$ ), kontrast ( $c$ ) ve yapıdır ( $s$ ).  $x$  ve  $y$  yi birer görüntü kabul edersek, iki görüntü arasındaki karşılaştırma, eşitlikteki gibi ifade edilir. Parlaklık  $I(x, y)$ , kontrast  $c(x, y)$  ve yapı  $s(x, y)$  fonksiyonları ile karşılaştırılmaktadır. Fonksiyondaki  $\alpha$ ,  $\beta$  ve  $\gamma$  karşılaştırmaların etki oranlarını ifade etmektedir.

#### 5.4.6. Tepe sinyal gürültü oranı (peak signal to noise ratio, PSNR):

İki görüntü arasındaki benzerliğin bir kalite ölçümü olarak kullanılır. Maksimum piksel değerinin piksellerin kalitesini etkilen gürültüye oranıdır ve genellikle logaritmik bir ölçek olan desibel cinsinden ifade edilir. Eş. 5.5'te gösterilen PSNR değeri ne kadar yüksekse, çıktı görüntüsün kalitesi o kadar iyidir.

$$\text{MSE} = \frac{1}{N} \sum_{i=1}^N (y_i - \hat{y}_i)^2 \quad (5.2)$$

$$\text{RMSE} = \sqrt{\frac{1}{N} \sum_{i=1}^N (y_i - \hat{y}_i)^2} \quad (5.3)$$

$$\text{MAE} = \frac{1}{N} \sum_{i=1}^N |(y_i - \hat{y}_i)| \quad (5.4)$$

$$R - \text{squared} = 1 - \frac{\sum_{i=1}^N (y_i - \hat{y}_i)^2}{\sum_{i=1}^N (y_i - \bar{y}_i)^2} \quad (5.5)$$

$$\text{PSNR} = 10 \log_{10} \frac{\text{MAX}_I^2}{\text{MSE}} \quad (5.6)$$

$$\text{SSIM} (I_F, I_W) = [I(I_F, I_W)]^\alpha \cdot [c(I_F, I_W)]^\beta \cdot [s(I_F, I_W)]^\gamma \quad (5.7)$$

### 5.5. Kullanılan Veri Setleri

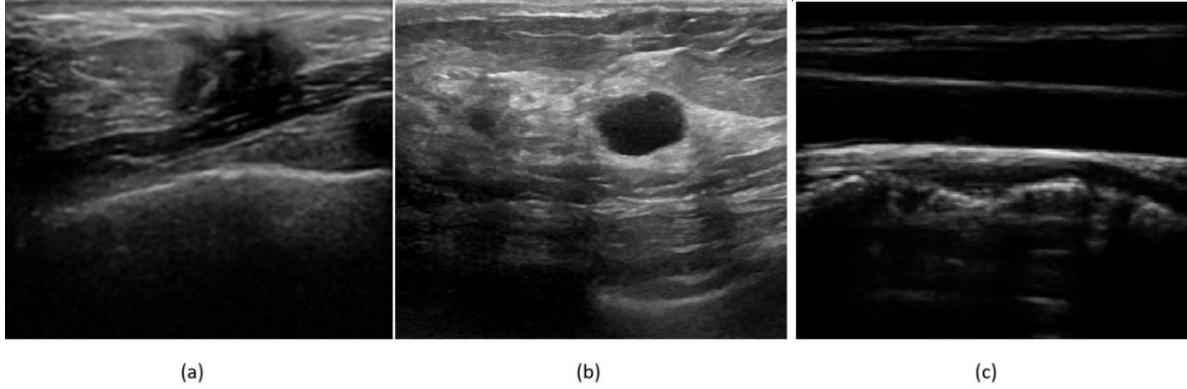
Bu tez çalışmasında ağ eğitimi, doğrulama ve test işlemleri için 3 farklı veri seti kullanılmıştır. Eğitim ve doğrulama CPU üzerinde yapılmıştır. Test işlemi ise hem CPU hem de FPGA üzerinde yapılarak belirlenen parametrelere göre bu sistemlerin performansları kıyaslanmıştır. Bu tezde kullanılan görüntüler, sınıflandırma problemleri için hazırlanan veri tabanlarından alınan US görüntüleridir.

**BUSI:** BUSI veri seti, normal, iyi huylu ve kötü huylu olmak üzere 3 sınıfa ait US B-mod görüntüleri içermektedir [103]. Görüntüler 2018 yılında, 25-75 yaşları arasındaki 600 kadın hastadan toplanmıştır. Bu veri kümesi, PNG görüntü formatında ortalama görüntü boyutu  $500 \times 500$  piksel olan 780 adet US görüntüsünü içermektedir.

**Dataset B:** Bu veri setinde toplamda 163 US B-mod görüntüsü bulunmaktadır [104]. 2012 de toplanan bu görüntüler, ortalama  $760 \times 570$  piksel boyutuna sahiptir.

**B modu karotid arter (CCA) ultrason görüntü veri tabanı:** Bu veritabanı, on gönüllünün ortak karotid arterinin (CCA) US B-mod görüntülerini içermektedir [105].

Şekil 5.6'da çalışmada kullanılan üç veri setine ait örnek görüntüler gösterilmiştir.

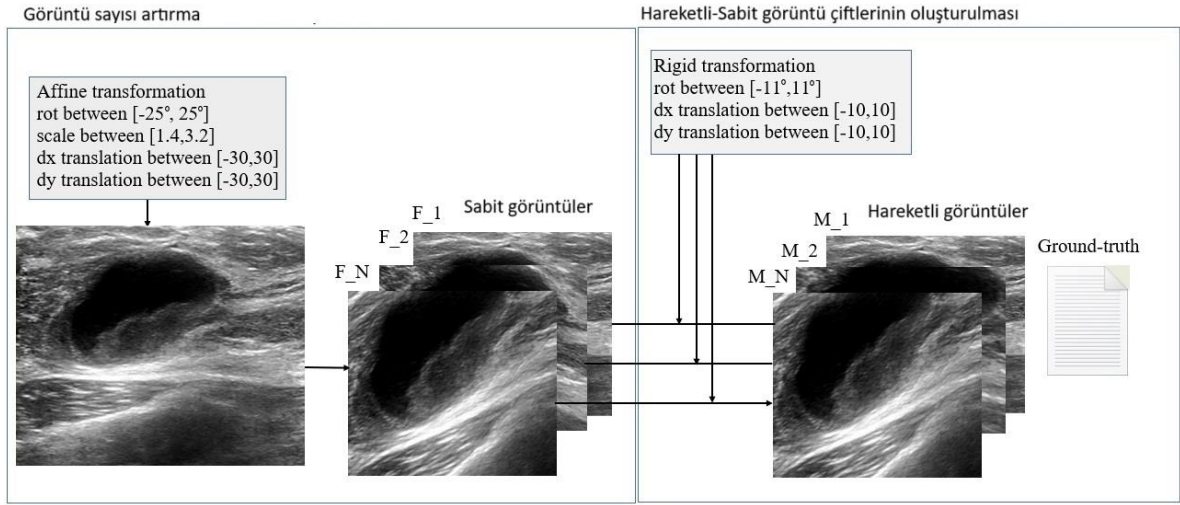


Şekil 5.6. Veri setlerinden örnek görüntüler (a) BUSI, (b) Dataset B ve (c) CCA

### 5.6. Veri Artırma ve Veri Seti Hazırlama

Görüntü çakıştırma probleminde hareketli-sabit görüntü çiftlerine ihtiyaç vardır. Web ortamında yukarıda belirtilen görüntü kümelerinin doğrudan görüntü çakıştırma işleminde kullanılması mümkün değildir. Bu nedenle ilk olarak yukarıda bahsedilen görüntülere afin dönüşüm yöntemleri uygulanarak veri sayısı artırılmış, daha sonra artırılan veri setlerine katı dönüşüm işlemleri uygulanarak görüntü çiftleri oluşturulmuştur (Şekil 5.7).

ESA eğitimi sırasında çeşitliliği artırmak ve ağın ezberlemesinin önüne geçmek için belirlenen veri kümeleri kullanılarak veri artırma işlemi yapılmıştır. Böylece ağı eğitimi için kullanılacak görüntü sayısı artırılmıştır. İlk olarak, yukarıda belirtilen üç US veri setine döndürme, ölçekleme ve öteleme gibi afin dönüşüm yöntemleri rastgele uygulanarak görüntü sayısı 10.230'a çıkarılmıştır. Veri sayısını artırmak için alınan görüntülere  $[-25^\circ, 25^\circ]$  arasında döndürme,  $[1,4 ; 3,2]$  oranları arasında ölçekleme,  $[-30, 30]$  piksel sağa-sola ve aşağı-yukarı öteleme işlemleri yapılmıştır.



Şekil 5.7. Veri artırma ve katı çakıştırma uygulaması için yeni veri seti oluşturma

Danışmanlı öğrenme yöntemi kullanan ESA temelli GÇ işlemi için sabit ve hareketli görüntü çiftleri ve aralarındaki dönüşüm matrisine ihtiyaç vardır. Bu nedenle ikinci adım olarak sabit-hareketli görüntü çiftleri ve aralarında dönüşüm matrisini içeren gerçek referans oluşturulmuştur. Bunun için ise üç parametre kullanılarak görüntü çiftleri oluşturulmuştur. Bunlar, döndürme (rot), x ekseninde öteleme (dx) ve y ekseninde ötelemedir (dy). Sabit görüntülere döndürme ve dx, dy öteleme işlemleri uygulanarak hareketli görüntüler oluşturulmuştur. Döndürme işlemi  $[-11^\circ, 11^\circ]$  aralığında saat yönünde (negatif döndürme değeri) ve saat yönünün tersine (pozitif döndürme değeri) kullanılmıştır. Öteleme işlemleri için, x ekseninde  $[-10, 10]$  piksel aralığında dx öteleme ve y ekseninde  $[-10, 10]$  piksel aralığında dy öteleme sabit görüntülere uygulanmıştır. Uygulanan değerler gerçek referans değerleri olarak hareketli-sabit görüntü çiftleri için kaydedilmiştir.

## 6. FPGA

DÖ gelişen teknolojiye paralel olarak büyük bir gelişme kaydetmiştir. Özellikle yüksek hesaplama yapma yeteneğine sahip ASIC, FPGA, ve GPU gibi farklı platformlara bu yapıların uygulanabilmesi ilerlemenin odağını oluşturmaktadır. CPU üzerinde yapılan uygulamalarla başlayan bu süreç, şimdilerde performans-güç parametrelerinin araştırılmasıyla ivmeyi daha çok GPU ve FPGA'lara kaydırmıştır. DÖ modelleri, yüksek hız ve performans gerektiren uygulamalardır. Günümüzde makine öğrenmesi uygulamalarının geliştirilmesine en fazla katkı sağlayan platformlardan olan GPU platformları binlerce çekirdeğe sahip olabilmektedir. GPU'lar derin öğrenme modellerini eğitmek için optimize edilmiştir ve birden çok paralel görevi bir CPU'dan üç kata kadar daha hızlı işleyebilir. Fakat GPU'lar güç tüketimi konusunda hem CPU'ların hem de FPGA'ların gerisinde kalmaktadır. Özellikle taşınabilir elektronik cihazlar için pil ömrü, çalışma süresi, performans ve verimlilik, ısı oluşumu ve soğutma gibi faktörler çok önemlidir. Bunların hepsi de cihazların güç tüketimi ile doğrudan ilgilidir. Bundan dolayı son yıllarda düşük güç tüketimi ve aynı zamanda yüksek performanslarından dolayı FPGA ile ilgili yapılan çalışmaların arttığı söylenebilir.

Yarı iletken teknolojisine sahip olan FPGA'lar istenilen sayısal devrelerin oluşturulması ve elektriksel olarak programlanabilmesini sağlayan platformlardır. FPGA'lar, üretimden sonra istenen uygulama veya işlevsellik gereksinimlerine göre yeniden programlanabilir [106, 107]. Bu özellik FPGA'ları belirli tasarım görevleri için özel olarak üretilmiş Uygulamaya Özel Tümüleşik Devrelerden (Application Specific Integrated Circuits, ASIC) ayırmaktadır. Tek seferlik programlanabilir FPGA'lar mevcut olsa da yaygın olarak kullanılan FPGA'lar yeniden programlanabilen SRAM tabanlı olanlardır. FPGA'lar, programlanabilir mantık bloklarının bir dizi ve bu bloklar arasında bağlantıları kurmaya imkân tanıyan tekrar yapılandırılabilir bir ara bağlantı hiyerarşisini içerir. Mantık blokları, karmaşık kombinasyonel işlevleri gerçekleştirmek veya temel mantık kapıları olarak işlev görmek üzere yapılandırılabilir. FPGA'lar, gömülü sistem geliştirme alanında önemli bir role sahiptir, çünkü sistem yazılımının geliştirilmesini donanım geliştirmesiyle aynı anda başlatabilir, sistem performans simülasyonlarını erken aşamalarda etkinleştirebilir ve sistem mimarisini tamamlamadan önce çeşitli sistem denemelerine ve tasarım düzeltmelerine izin verir. Yeniden programlanabilir yapısı ve düşük güç tüketimi gibi avantajları sayesinde FPGA'lar birçok farklı alanda kullanılmaktadır. Bunlardan bazıları:

- Dijital Tasarım ve Prototipleme: FPGA'lar, karmaşık dijital devrelerin tasarım ve prototiplemesinde yaygın olarak tercih edilir, çünkü programlanabilir yapıları sayesinde tasarım sürecinin hızlı ve tekrarlanabilir şekilde geliştirilmesine ve test edilmesine imkan sağlar.
- Görüntü ve Sinyal İşleme: FPGA'lar, görüntü ve sinyal işleme uygulamalarında yüksek hızlı veri işleme ihtiyacını karşılamak için ideal bir tercihtir. Özellikle görüntü işleme, video kodlama/çözme, filtreleme ve ses işleme gibi uygulamalarda kullanımları yaygınlaşmaktadır.
- Ağ ve İletişim: FPGA'lar, yüksek hızlı veri iletişimi gerektiren ağ altyapılarında, veri paketleme, açma, veri şifreleme ve çözme gibi işlemlerde etkin bir şekilde kullanılırlar.
- Endüstriyel Otomasyon ve Kontrol: Endüstriyel otomasyon sistemlerinde, paralel işlem ve hızlı tepki süreleri gerektiren uygulamalarda FPGA'lar, PLC'lerin (Programmable Logic Controllers) yanı sıra kullanılarak etkin bir şekilde destek sağlar.
- Yapay Zeka ve Derin Öğrenme: FPGA'lar, yapay zeka ve derin öğrenme uygulamalarında özellikle model eğitiminde hızlandırma ve optimize etme amacıyla kullanılır.
- Uzay ve Havacılık: FPGA'lar, uzay ve havacılık uygulamalarında, yüksek güvenilirlik, hafiflik ve dayanıklılık gerektiren işlemler için tercih edilir.
- Tıp ve Biyoteknoloji: Tıp alanında, FPGA'lar, görüntüleme cihazları, tıbbi cihazlar ve biyoenformatik işlemleri gibi çeşitli uygulamalarda kullanılır.

FPGA'lar, sadece bahsedilen uygulamalarla sınırlı kalmayarak birçok farklı alanda kullanılabilirler. Özellikle düşük güç tüketimi, yüksek hız ve özelleştirilebilirlik gerektiren uygulamalarda, FPGA'lar diğer donanım platformlarına kıyasla önemli avantajlar sunmaktadır. Çizelge 6.1'de farklı parametreler açısından FPGA, CPU ve GPU kıyaslaması yapılmıştır.

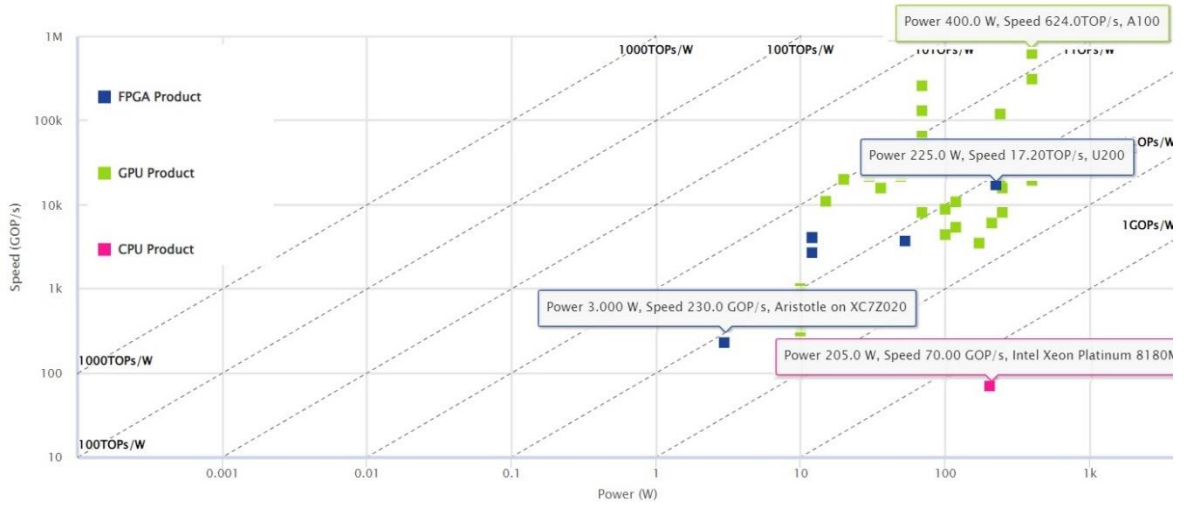
Çizelge 6.1. GPU, CPU ve FPGA'nın farklı parametrelere göre kıyaslanması

Parametre	Değerlendirme	En iyi platform
DÖ eğitim aşaması	GPU kayan noktalı sayılarla işlem yapma performansı çok iyi	GPU
DÖ çıkarım aşaması	FPGA özelleştirilebilir olmasından dolayı gecikme daha az	FPGA

Çizelge 6.1. (devam) GPU, CPU ve FPGA'nın farklı parametrelere göre kıyaslanması

Gecikme (latency)	FPGA üzerinde gerçekleştirilen algoritmalar özelleştirilebilir optimizasyonlar uygulanabileceği için gecikme ayarlanabilir	FPGA
İşlem/Watt	FPGA özelleştirilebilir olmasından dolayı daha avantajlı	FPGA
İşlem/Maliyet	GPU yüksek işlem yapma kapasitesinden dolayı, FPGA ise geniş uygulama alanı avantajından dolayı	GPU, FPGA
Arayüz	FPGA birçok arayüz bağlantısını desteklemekte	FPGA
Algoritma geliştirme süreci	CPU ve GPU'ların ikisi de FPGA'dan daha kolay algoritma geliştirme sürecine sahip	CPU, GPU
Boyut	FPGA ve CPU platformlarının ikisi de daha küçük boyuta sahip	FPGA, CPU

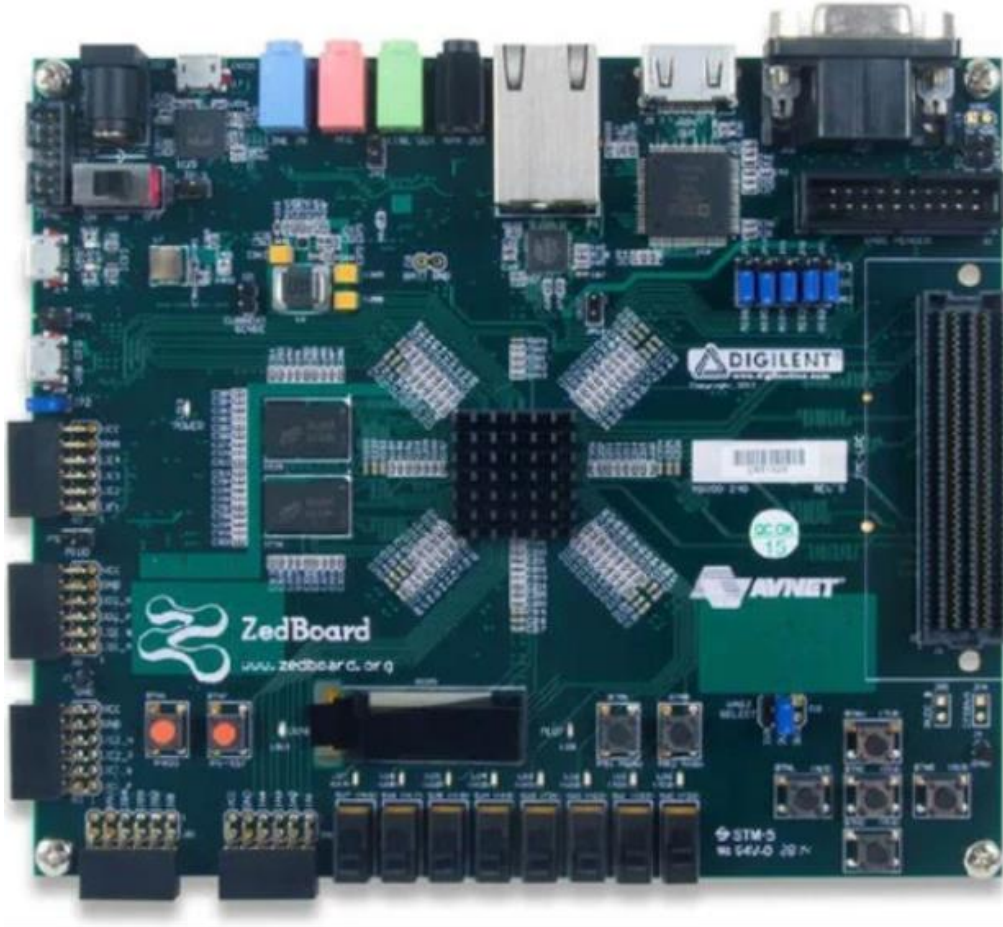
Guo vd. [108] 2010-2021 yılları arasında hem akademik çalışmalarda hem de endüstriyel alanlarında derin öğrenme algoritmalarının GPU, CPU, FPGA ve ASIC platformları üzerinde farklı parametrelere göre incelenmesini grafik üzerinde göstermişlerdir. Şekil 6.1'de yapılan çalışmanın bir kısmı filtrelenerek gösterilmiştir. Bu çalışmaya göre, en düşük güç tüketiminin 28 nm teknolojisine sahip XC7Z020 FPGA platformu kullanılarak yapılan çalışmada olduğu görülmektedir. CNN gerçekleştirilmesi için yapılan uygulama 76,67 GOP/s/W performansına sahiptir. En düşük performansın ise CNN/RNN uygulamasınının 14 nm teknolojisine sahip Intel Xeon Platinum 8180M üzerinde 0,3415 GOP/s/W ile CPU uygulamasınının olduğu görülmektedir. En yüksek performansın ise 7 nm teknolojisine sahip NVIDIA A100 GPU üzerinde 1,560 TOP/s/W olarak hesaplanmıştır.



Şekil 6.1. GPU, FPGA ve CPU'nun CNN/RNN gerçeklemlerinde kullanılan güç/verim grafiği [108]

Yapılan bu çalışmaya bakılarak, CPU'ların birçok senaryoda işlem hızlarının en düşük olduğu, en yüksek işlem performansına sahip platformun GPU olduğu ve güç tüketimlerinin genel olarak yapılan uygulamaya bağlı olarak değiştiği fakat GPU'ların birçok senaryoda FPGA'ların gerisinde kaldığı söylenebilir.

Bu tez çalışmasında gerçek zamanlı uygulama için FPGA platformu olarak Xilinx Zynq-7000 All Programmable SoC'yi içeren ZedBoard geliştirme kartı kullanılmıştır. Zedboard, farklı uygulamaların gerçekleştirilebilmesi için gerekli birçok arabirimleri ve destekleyici işlevleri de içermektedir. Kartın genişletilebilirlik özellikleri sayesinde hızlı prototip oluşturulmasına imkân sağlar. ZedBoard geliştirme kitinin görünümü Şekil 6.2'de gösterilmiştir.



Şekil 6.2. Zedboard kartının genel görüntüsü

ZedBoard geliştirme kartının bazı teknik özellikleri:

- Xilinx Zynq-7000 AP SoC XC7Z020-CLG484
- Dual-core ARM Cortex™-A9
- 512 MB DDR3
- 256 MB Quad-SPI Flash
- 4 GB SD card
- On-board USB-JTAG Programming
- 10/100/1000 Ethernet
- USB OTG 2.0 and USB-UART
- Dijital Ses işlemcisi ADAU1761 SigmaDSP® Stereo, 96 kHz, 24-Bit Audio Codec
- HDMI/DVI verici ADV7511 225 MHz HDMI verici (1080p HDMI, 8-bit VGA, 128x32 OLED)
- PS and PL I/O expansion (FMC, Pmod, XADC)

## 6.2. FPGA Programlama

FPGA platformunun programlanması için öncelikle donanım davranışları donanım tanımlama dilleri olan yüksek hızlı tümleşik devreler için donanım tanımlama dili (Very high speed integrated circuit Hardware Description Language, VHDL) veya Verilog kullanılarak yapılmaktadır [109, 110]. Ancak, bu diller, diğer programlama dillerine kıyasla öğrenmesi daha zor ve zaman alıcı olabilir. Diğer programlama dillerinde tek satırda yapılan işlemler bazen VHDL ve Verilog'da daha karmaşık ve birden çok kod satırına denk gelebilmektedir. Vivado platformu blok tasarım yapmayı, tasarlanan blokların paketlenmesini ve yazılım donanım olarak ortak tasarım yapılmasını oldukça basit yapılmasına imkân sağlamıştır.

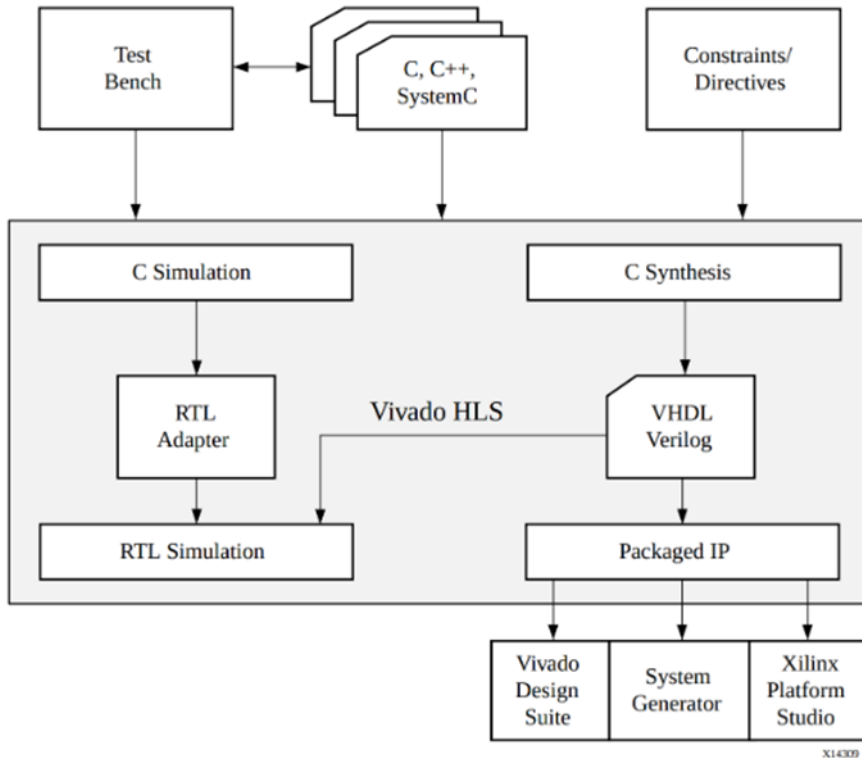
Bu tez çalışmasında oluşturulan paketlerin FPGA'ya gömülmesi için Vivado 2017.4 ve ESA uygulamasının yüksek seviyeli dillerle tasarlanması, doğrulanması ve sentezlenmesi için ise Vivado yüksek seviyeli sentezleme (high level synthesis, HLS), 2017.4 versiyonunun ücretsiz yayımları olan WebPack Edition kullanılmıştır. Vivado Artix®-7, Kintex®-7, Kintex UltraScale™, Zynq®-7000 All Programmable SoC gibi FPGA kartlarını ücretsiz olarak desteklemektedir.

### 6.2.1. Yüksek seviyeli sentezleme

Vivado HLS (AMD Adaptive Computing Documentation Portal, 2023), Xilinx tarafından geliştirilmiş bir araçtır [111, 112]. Vivado HLS, kullanıcıların C/C++ kodunu kullanarak karmaşık FPGA tabanlı algoritmaları kolayca oluşturmasına olanak tanır. C/C++ kodları ve tanımlamaları saklayıcı aktarım seviyesine (register transfer level, RTL) dönüştürerek, Xilinx FPGA'larında gerçekleşmesini sağlar. HLS ile yapılan tasarım VHDL ve Verilog kodlarına dönüştürülerek, daha sonra FPGA içine gömülebilen IP paketlerine dönüştürülmektedir. C dilleriyle yazılan programların FPGA'da paralel çalışabilmesi, gerçekleştirme esnasında kaynak kullanımının optimizasyonu ve modelin donanım üzerindeki performansını artırmak için çeşitli imkanlar sunmaktadır. Ayrıca HLS, yazılım ve donanım arasındaki bağlantıyı da sağlamaktadır. Donanım tasarımcıları için yüksek soyutlama seviyesinde üretkenliği artırmakta, yazılım tasarımcıları için ise geliştirilen modellerin FPGA donanımları üzerinde performans artırma imkanları sunmaktadır [112]. Oluşturulan algoritmalar hem C seviyesindeki diller için fonksiyonel olarak doğrulanabilmekte, hem de

donanım seviyesinde RTL olarak doğrulanabilmektedir. C seviyesinde fonksiyonel doğrulama sayesinde yapılan uygulamalara RTL'e dönüştürülmeden hızlı bir şekilde kontrol edilebilmektedir. Böylece zaman kayıplarının önüne geçilebilir. Vivado HLS tasarımları basitçe şu şekilde gerçekleşmektedir:

- İlk olarak algoritmalar C seviyesindeki dillerle oluşturulmaktadır. Böylece, donanım tanımlama dilleri kullanılırken, dikkat edilmesi gereken gerçekleştirme detaylarından kurtulmaktadır.
- İkinci olarak yazılan C kodları C seviyesinde doğrulanabilmektedir. Donanım üzerinde yapılan gerçeklemlerin test edilmesi uzun sürmektedir. C seviyesinde kodların doğrulanması sayesinde, geliştirilen algoritmaların fonksiyonel olarak doğru çalışıp çalışmadığı kısa süre içinde test edilebilmektedir.
- C seviyesinde fonksiyonel olarak doğru çalıştığı test edilen algoritmalar daha sonra istenen optimizasyonlar uygulanarak yüksek performanslı tasarımlar gerçekleştirilebilmektedir.
- Farklı optimizasyonlar uygulanarak oluşturulan tasarımlar, kaynak kullanımı, performans, gecikme gibi çeşitli değerlendirme parametreleriyle kıyaslanabilmektedir.



Şekil 6.3. Xilinx FPGA ları için HLS akış tasarımı [112]

Şekil 6.3'te Xilinx FPGA ları için HLS akış tasarımının genel yapısı gösterilmiştir. Tasarım girdileri genel olarak üç dosyadan oluşmaktadır. Dosyalardan bir tanesi C, C++ veya SystemC dilinde yazılmış kodların bulunduğu dosyalardır. İkinci dosya ise, C dilleriyle yazılmış algoritmaların fonksiyonel doğruluğunun test edilmesi için kullanılan testbench dosyalarıdır. Üçüncü girdi dosya grubu ise, kullanılan donanımı, hedef çalışma frekansını gibi bilgileri içeren Kısıtlar (Constraints) ve tasarımlara uygulanacak olan optimizasyonları içeren direktifler (Directives) dosyalarıdır. HLS ile yazılan algoritmalar FPGA'ya gömülmeden önce doğru çalışıp çalışmadığı iki şekilde kontrol edilebilir. Bunlara sentezden önce geçerlilik analizi (SÖD, pre-synthesis validation) ve sentezden sonra doğrulama (SSG, post-synthesis verification) denir. SÖD ile C düzeyinde algoritmaların çalışması, kodlar Verilog, VHDL'e dönüştürülmeden önce kontrol edilmektedir. Bu aşamadan, girdi olarak kullanılan C, C++, SystemC dosyalarından ilgili olanı ve test-bench dosyası kullanılır. Fonsiyonel olarak doğru çalıştığı onaylanan kodun ikinci aşama olarak Verilog veya VHDL kodlarına dönüştürülmesini sağlayan sentezleme işlemi yapılır. Sentezlenen kodun doğru çalışıp çalışmadığının kontrolü ise, C/RTL co-simulasyon ile yapılır. C/RTL co-simulasyonu işlemi, sentezlenen VHDL, Verilog dosyalarını ve test-bench dosyasını kullanır. Bu iki doğrulama işlemi isteğe bağlıdır. C test-bench dosyasının yazılması isteğe bağlıdır. Fakat daha hızlı bir tasarım için, sentezlemeden önce işlemlerin fonksiyonel olarak doğruluklarının test edilmesi için kullanılması önerilmektedir.

Yazılan C düzeyindeki kodların FPGA içine gömülmeye hazır hale gelmesi süresi için Constraints/Directive dosyalarına ihtiyaç vardır. Sentezleme için kullanılacak hedef FPGA, optimizasyonların olduğu dosyalar sentezleme işlemi için kullanılmaktadır. Sentezlenen kodlar Verilog veya VHDL'e dönüştürüldükten sonra bunların IP paketlere dönüştürülüp FPGA içine gömülmeye hazır hale getirilmesi gerekir. HLS'de oluşturulan modüller test edilip sentezlendikten sonra IP-Core olarak paketlenerek Vivado ortamına aktarılır. Böylece, Vivado ortamında oluşturulacak olan FPGA projelerinde kütüphane dosyası olarak kullanılabilir. RTL IP paketleri önceden hazır olan diğer IP paketlerle entegre hale getirilebilir. RTL IP paketler birbirleriyle sıralı, boru hattı paralelliği veya veri akış paralelliği yöntemleriyle de bağlanabilir ve işlem yapabilirler. IP paketleri üç şekilde oluşturulabilir. Bunlar, Vivado design suit, System generator ve Xilinx platform Studio seçenekleridir.

## 7. FPGA TABANLI ESA GERÇEKLEMESİ

ESA uygulamasının FPGA üzerinde gerçekleştirilmesi için Vivado FPGA tabanlı yapılacak olan ESA uygulamaları için dikkat edilmesi gereken pek çok nokta vardır. Bunlardan bazıları:

- Kullanılacak sayı gösterim formatı ve uzunluğu
- Tasarım şekli
- Paralellik yöntemi
- Veri ulaşım şekli

### 7.1. Kullanılacak Sayı Formatı

Yapılan FPGA tabanlı çalışmalar için kullanılan sayı gösterim formatları ve sayı gösterim uzunluklarının doğru bir şekilde seçilmesi çok önemlidir. HLS uygulamaları için C dilinde yazılan algoritmalar genellikle 8, 16, 32 veya 64 bit gibi sekizin katları uzunluğunda olmaktadır. C veri tiplerini kullanmak, donanım gerçekleştirilmesinde verimsiz kaynak kullanımına neden olabilmektedir. Vivado HLS isteğe bağlı uzunlukta tasarlanabilen sayı gösterim şeklini kullanma imkanı sunar. Veriler isteğe bağlı uzunlukta seçilerek, gereksiz kaynak kullanımı ve gecikmenin önlenmesi sağlanabilir. Bu gösterim şekline “arbitrary precision data type” denir. Bu gösterim hem tam sayılar için (arbitrary precision integer data type) hem de kesirli sayılar için (arbitrary precision fixed-point data type) kullanılabilir.

#### 7.1.1. Sabit noktalı sayı formatı

Donanım platformlarında, sayıları kesirli bileşenlerle temsil etmek için iki temel yaklaşım vardır: sabit nokta ve kayan nokta gösterimi [113, 114]. Donanım tabanlı sistem tasarımlarında kullanılan veri türlerinin formatları, kaynak kullanımı, doğruluk oranı ve gecikme süresi gibi birçok yönden sistemin performansını etkiler. Sabit nokta temsili, keyfi kesinlik veri türleri olarak da adlandırılır. Tamsayı kısmı ve kesir kısmı için bit genişliği hassasiyete göre istenildiği gibi seçilebilir.

$$X = \langle M, I \rangle, M = I + F \quad (7.1)$$

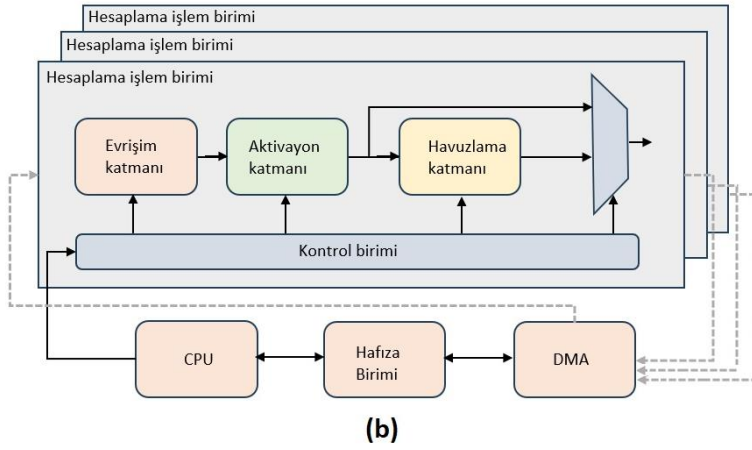
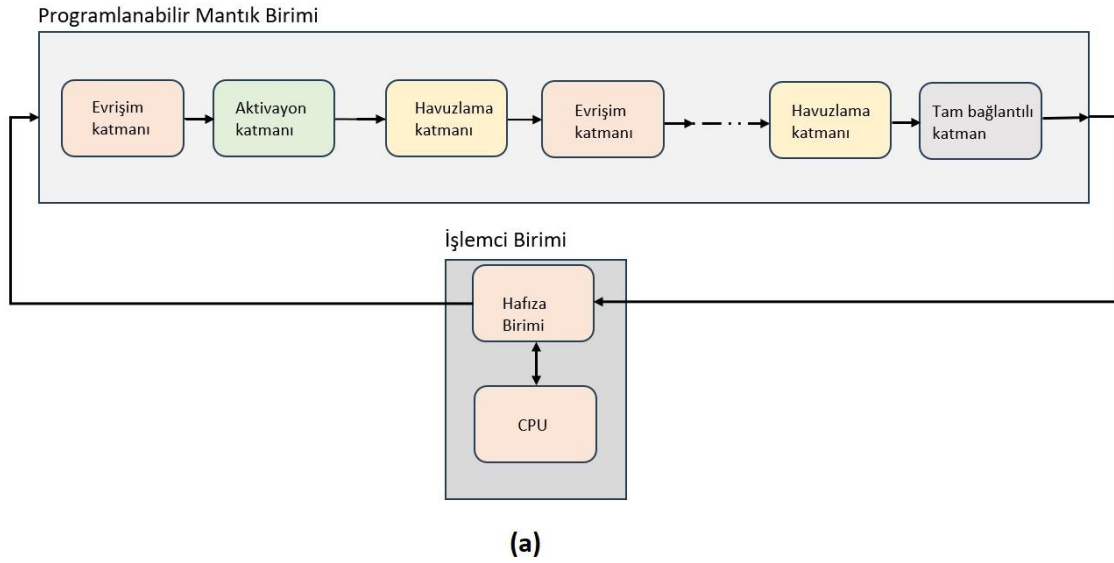
M, I, F sırasıyla M-bit deęişkenini, ikili noktanın üzerindeki sayıyı ve ondalık virgölün altındaki deęeri temsil eder. Doğru veri gösterim türünün kullanılması, daha hızlı çalışmanın ve daha az donanım kaynaęı kullanmanın yanı sıra, neredeyse aynı doğrulukta donanım uygulaması için daha iyi bir performans sağlar. Bu çalışmada, sayı görüntüleme formatı olarak keyfi duyarlıklılı sabit noktalı veri tipi kullanılmıştır.

### **7.1.2. Tek duyarlıklılı kayan noktalı sayı formatı**

Kayan noktalı sayılar, geniş sayı gösterim aralığına sahip olduęu için en çok tercih edilen gösterim şeklidir [115]. IEEE 754 standardına göre tek duyarlıklılı kayan noktalı sayılar ikili tabanda 32 bit olarak gösterilmektedir. Bu gösterim şekli, mantissa, exponent ve işaret biti olmak üzere üç temel bileşene sahiptir. Kayan noktalı sayılarla çalışmanın en büyük avantajı, hesaplama hassasiyetlerinin çok yüksek olmasıdır [116, 117]. Böylece aęın ürettięi sonuçların doğruluk oranı da aynı şekilde yüksek olacaktır. Fakat donanım tabanlı yapılan çalışmalarda, yüksek hassasiyete sahip olan bu sayı formatı, beraberinde çok fazla kaynak tüketimine de neden olmaktadır. Çok fazla kaynaęın tüketilmesi de daha fazla güç kullanımına neden olacaktır. Donanımlar için aęın performansı, tüketilen kaynak sayısı ve harcanan güç arasındaki dengeyi sağlamak oldukça önemlidir.

### **7.2. Tasarım Modeli**

Donanım tabanlı ESA uygulamaları için Şekil 7.1'de gösterildięi gibi iki farklı tasarım modeli vardır [118]. Bunlar; akış tasarımı ve tek birimli tasarım modelidir. Akış tasarımı modelinde her katman ayrı ayrı tasarlanır ve donanımda sentezlenir. Tek birimli tasarım modelinde ise bir birim tasarlanır ve sırası gelen katman bu birimi hesaplama için kullanır.



Şekil 7.1. Donanım tabanlı ESA yöntemleri (a) akış tasarımı, (b) tek hesaplama birimi

Akış mimarisinde katmanları kontrol eden kontrol birimlerine ihtiyaç yoktur. Fakat tek hesaplama birimli tasarımlarda katmanların sırasını kontrol eden kontrol birimi bulunmaktadır. Her iki mimarinin de bazı avantaj ve dezavantajları bulunmaktadır. Akış modelinde katmanlar teker teker tasarlanıp sentezlendiği için her katmana farklı optimizasyon yöntemleri uygulanabilir. Böylece paralel işlemler yürütülebilir ve tasarımın verimi (throughput) artırılabilir. Ayrıca katmanlar arka arkaya bağlandığı için katmanlar arasında eş zamanlı çalışma mümkündür. Böylece bir katmanın bütün çıktıları üretilmeden, sonraki katman çalışmaya başlayarak gecikme (latency) parametresi iyileştirilebilir. Ayrıca katmanlar eş zamanlı çalışabileceği için katmanlardan oluşan ara çıktılar FIFO, ping-pong veya tampon bellek (buffer) gibi saklama birimlerinde depolanarak çip dışı (off-chip) bellek ihtiyacı büyük oranda azaltılabilir. Bu yapının dezavantajı ise, sınırlı donanım kaynaklarından dolayı çok geniş ve derin ağların katmanlarının ayrı ayrı tasarlanıp FPGA

üzerinde gerçekleşmesi mümkün olmayabilir. Tek birimli tasarım modelinde ise tek bir birim tasarlandığı için büyük ve geniş ağlar FPGA üzerinde gerçekleştirilebilir. Farklı boyutlardaki ESA modellerinin bu yapıya uygulanabilmesi daha kolaydır. Fakat tek bir birim olduğu için bütün katmanlar için aynı optimizasyon işlemleri uygulanır. Katmanlar aynı birimi kullandıkları için eş zamanlı çalışmaları mümkün değildir. Katmanların ara çıktılarının hepsinin oluşması ve hafıza birimlerinde saklanması gerekir. Bu da parametrelerin saklanması için ihtiyaç olan hafıza birimlerinde artışa neden olur.

### 7.3. FPGA Tabanlı ESA Hızlandırma Yöntemleri

FPGA tabanlı yapılan ESA çalışmaları genellikle ESA'nın çıkarım aşaması için yapılmaktadır [119, 120]. ESA mimarileri çok katmanlı yapılar olduğu için ve uygulama sırasında binlerce parametre kullandıkları için bu yapıların FPGA üzerinde optimizasyon yapmadan gerçekleşmesi sistemin performansını olumsuz yönde etkilemektedir. İşlemlerin sıralı yapılması toplam gerçekleştirme süresini uzatacak, saklama birimlerinde optimizasyon yapılmaması hem donanım verimini hem de gerçekleştirme süresini etkileyecek aynı zamanda, kullanılacak sayı formatlarının gereğinden fazla hassasiyet içermesi daha fazla kaynak kullanımına neden olacaktır [121]. Bundan dolayı, optimizasyon işlemlerinin yapılması gerekmektedir.

ESA'yı oluşturan temel katmanlar evrişim, aktivasyon, havuzlama ve tam bağlantılı katmanlardır. Evrişim işlemi için kullanılan sözde kod (pseudo code) Şekil 7.2'de gösterilmiştir. HLS kullanılarak yapılan donanım tabanlı ESA işleminde kullanılan başlıca yapılar vardır. Bunlar; döngüler, diziler ve veri tipleridir. Şekil 7.2'de görüldüğü gibi evrişim işlemi iç içe for döngülerinden oluşmaktadır. Verilerin tasarıma aktarılmasında ve depolanmasında ise diziler kullanılmaktadır.

Döngüler HLS tarafından sentezlenebilen ve farklı eniyileme yöntemleri uygulanarak tasarımın iyileştirilmesinde büyük role sahip yapılardır. Döngüler üzerinde boru hattı (pipeline), döngü açma, kısmen döngü açma, farklı döngülerin birleştirilmesi veya döngülerin parçalanması gibi eniyileme işlemleri yapılabilir. Döngü açma eniyilemesinde Şekil 7.2'de gösterilen en içteki döngü açılabilmesi gibi en sondaki döngü de açılabilir. Bütün döngüler kısmen veya tamamen açılabilirler. Yapılacak olan tasarımda performans ve tüketilen kaynak dengesi göz önünde bulundurularak farklı şekilde döngü açma işlemleri

uygulanabilir. Döngülerin paralel işlem yapması döngülerde kullanılan değişkenlerin diğer döngülere bağımlı olup olmamasına da bağlıdır. Bağlı değişken bulunan döngüler, diğer döngüden değer gelmesini beklediği için tam anlamıyla paralel çalışabilmesi mümkün olmayacaktır.

```

for(int no=0; no<M; no++) {
  for (int y = 0; y < R; y++){
    for (int x = 0; x < C; x++){
      for (int ni = 0; ni < N; ni++){
        for(int ky=0;ky<k;ky++) {
          for(int kx=0;kx<k;kx++){

            OBRAM[no][x][y]+=IBRAM[ni][S1*x+kx][S1*y+ky]*WBRAM[ni][no][kx][ky]; } } } } } }

```

Şekil 7.2. Evrişim işlemi sözde kodu

Döngülerin açılması hız parametresinin iyileştirilmesi için yeterli değildir. Aynı zamanda C dillerinde önemli bir yapı olan dizilerde de bazı eniyileme işlemleri uygulamak gerekmektedir. Eşzamanlı işlem yapan birimlere verilerin de eşzamanlı gelmesi gerekmektedir.

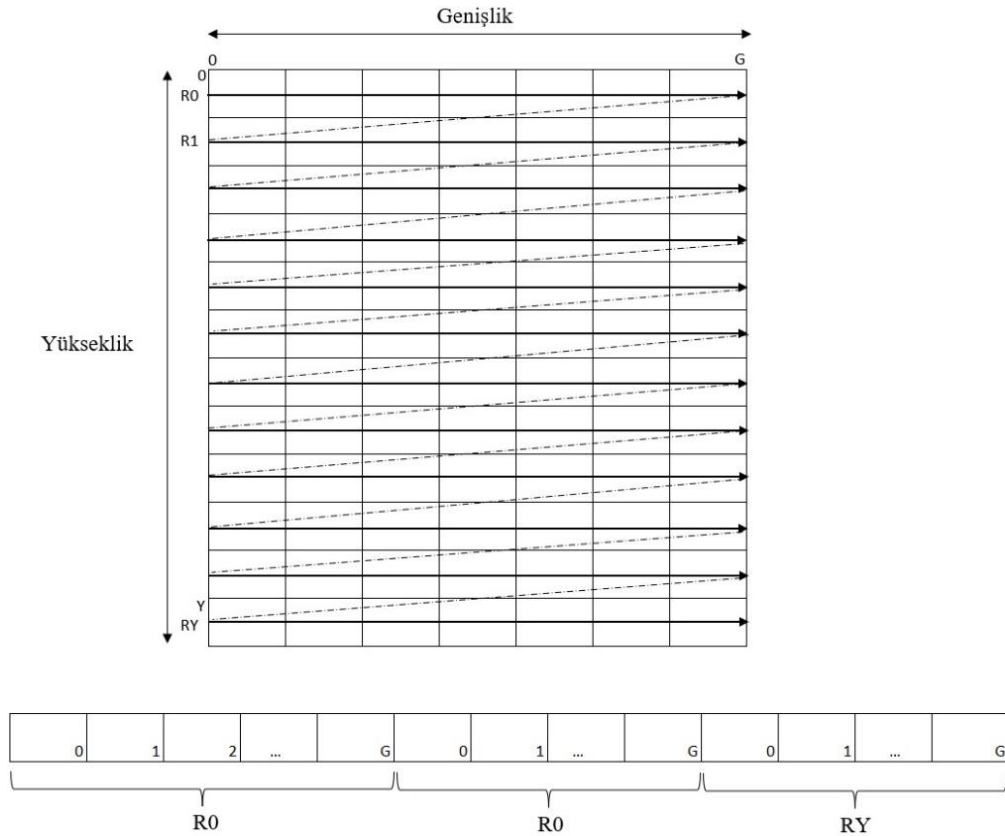
Diziler RTL sentezlemede hafıza birimleri olarak sentezlenirler. Bu hafıza birimleri RAM, ROM veya FIFO olabilir. C düzeyinde kodlamada ana fonksiyon (top-level) argüman olarak bulunan diziler RTL sentezlemede dışardan hafıza biriminde veri alabilen yapılar olarak gerçekleştirilir. 1024-bit boyutundan küçük olan diziler kaymalı/seri girişli kaydedicilere (shift register) depolanırlar. 1024-bitten büyük veriler ise RAM, LUTRAM veya UltraRAM hafıza birimlerine kullanılan FPGA platformuna ve yapılan eniyileme yöntemine bağlı olarak sentezlenirler. Dizilerde eniyileme yapılmaması darboğaz (bottleneck) sorununa neden olabilir. Hafıza birimlerinin port sayısı sınırlı olduğu için, paralel işlem yapan birimler kısıtlı sayıda veri iletilebilir. Bu da veri akışında darboğaza neden olacaktır. Eniyileme işlemi olarak dizi parçalama yöntemi kullanılmaktadır. Dizi parçalama yöntemi, RTL sentezleme sırasında yapılmaktadır. Büyük diziler daha küçük

RTL RAM yapılarına veya bağımsız elemanlar olarak saklayıcılara (register) parçalanabilirler. Böylece veri paralelliği sağlanmış olacaktır.

Dizilerdeki verilere ulaşmak için kullanılan özel ve önemli bir hafıza elemanı da FIFO yapısıdır. Özellikle veri akışı paralelliği (dataflow) uygulamalarında FIFO kullanmak modelin performansını artırırken kullanılan donanım kaynak sayısına da olumlu etkileyecektir. Verilerin sıralı alındığı durumlarda gereksiz adresleme yapılmasını önleyecektir.

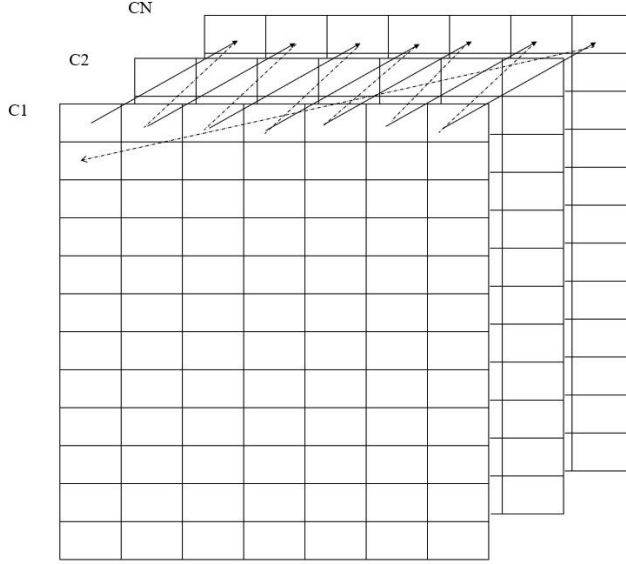
### 7.3.1. Veri alışverişi ve tampon bellek kullanımı

Donanım tabanlı görüntü işlemedeki yaygın darboğazlardan biri, görüntüyü bellekten okumak ve elde edilen görüntüyü belleğe yazmak için gereken zaman ve bant genişliğidir. Bunu iyileştirmek için kullanılan tekniklerden bir tanesi ızgara (raster) tarama tekniğidir. Izgara tarama tekniğinde pikseller belirli bir sırayla alınarak işlem yapılır. Donanım tabanlı yöntemde kullanılan ızgara tarama yöntemi Şekil 7.3'te gösterilmiştir.



Şekil 7.3. Izgara (raster) tarama düzeni

Bu tez çalışmasında kullanılan görüntüler birden fazla kanallardan oluşmaktadır. Ayrıca ara katmanların çıktıları da çok kanallı özellik haritalarıdır. Bu özellik haritalarının hafıza birimlerine aktarılması Şekil 7.4'te gösterilen şekilde yapılmıştır.



Şekil 7.4. Çok boyutlu özellik haritası için ızgara tarama düzeni

Bu tez çalışmasında çıkarımı yapılacak görüntü Şekil 7.4'te gösterilen tarama düzenine göre öncelikle BRAM birimlerine depolanmıştır. Uygulama esnasında ise BRAM bellekte bulunan veriler tampon belleklere aktarılmıştır. Verilerin tasarıma alınması HLS de bulunan akış işleme (streaming process) özelliği kullanılarak yapılmıştır. BRAM yapılarında bulunan veriler öncelikle satır tamponlara (line buffer) depolanmakta oradan da pencere tamponlara ayrılmaktadır.

FPGA tabanlı ESA tasarımlarının performansı farklı şekillerde iyileştirilmektedir. Genellikle birim zamanda gerçekleştirilen iş (verim, throughput), bir işin cevap süresi (latency) ve kullanılan donanım kaynak miktarının iyileştirilmesi için farklı optimizasyon yöntemleri kullanılmaktadır.

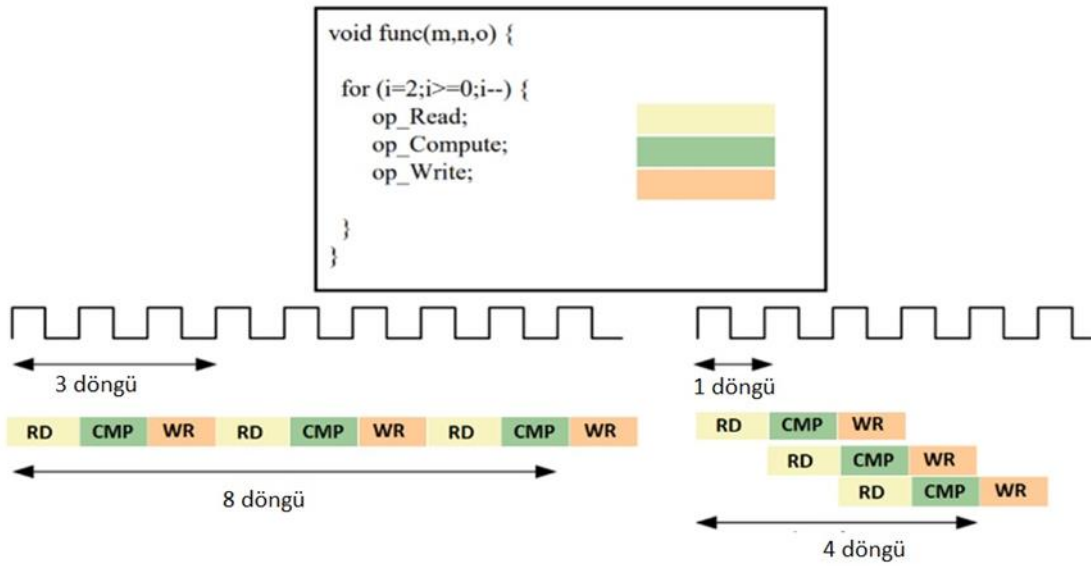
### 7.3.2. Verim optimizasyon yöntemleri

Verim, sistemin girdileri işleyebilme hızıdır. Belirli bir zamandaki ölçüm miktarıdır. Verim genellikle FPGA da yapılan ESA işlemleri için Giga Operation Per Second (GOPS/s) olarak ifade edilir. ESA tasarımında verimi artırmak için uygulanan yöntemler;

- Döngü ve fonksiyonların işlemlerinin ardışık yapılması (pipeline)
- Dizilerin parçalanması (array partitioning)
- Döngülerin açılması (loop unrolling)
- Görev seviyesinde paralellik gibi iyileştirme işlemleri uygulanır.

### Boru hattı paralelliği işlemi (PIPELINE)

Boru hattı, işlemlerin eşzamanlı olarak gerçekleşmesine imkân sağlar. Başlatma aralığını (II) iyileştirmek için kullanılır. Bu yöntemde, her yürütme adımının, bir sonraki işleme başlamadan önce tüm işlemleri tamamlaması beklenmez (UG902 Vivado). Boru hattı tekniği, fonksiyonlara ve döngülere uygulanabilir. Şekil 7.5'te 3 iterasyona sahip bir döngüye boru hattı tekniği uygulanmadan önce ve uygulandıktan sonra çalışma zamanında oluşan fark saat çevrimi (clock cycle, cc) olarak gösterilmiştir.



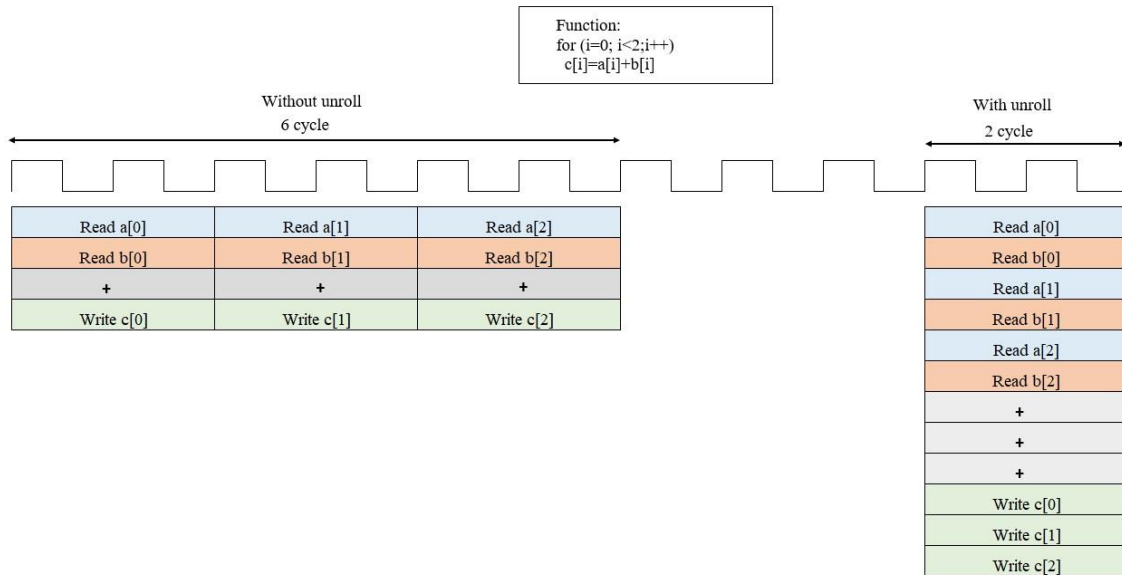
Şekil 7.5. Boru hattı eniyileme yönteminin döngü üzerindeki etkisi

Şekil 7.5'te, 3 iterasyona sahip bir for döngüsü görülmektedir. Döngü içinde üç işlem yapılmaktadır. Boru hattı tekniği uygulandığı zaman, her bir işlemin bir saat döngüsünde yapıldığı varsayılırsa, üç işle üç saat döngüsünde tamamlanmakta, döngünün üç defa tekrarlanması için ise toplamda 8 saat döngüsü gerekmektedir. Fakat döngüye boru hattı tekniği uygulandığında ise, döngü içindeki bir işlem bitmeden diğer işlem başlamakta böylece döngünün tamamlanabilmesi için 4 saat döngüsü gerekmektedir. Boru hattı kullanılmayan şekildeki tasarımda döngünün yeni iterasyonu başlatması için gereken saat

döngüsü (initiation interval, II) 3'tür, fakat boru hattı kullanılan tasarımda ise  $II=1$ 'dir. Ardışık düzen sayesinde döngü içindeki işlemlerin örtüşen bir şekilde (overlap) uygulanmasına izin verilir.

### Döngülerin açılması (loop unrolling)

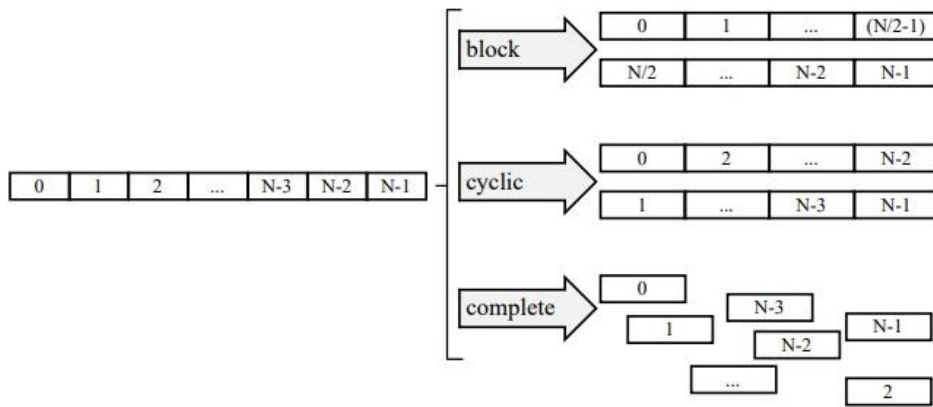
Herhangi bir optimizasyon uygulanmamış döngüler donanım tabanlı gerçekleştirilirken, her yinelemede kullanılmak üzere bir donanım kaynağı oluşturur ve sırayla bu kaynak kullanılır. Böylece donanım kaynakları verimli bir şekilde kullanılmış olur fakat bazen bir performans darboğazına neden olabilir. Döngülerin açılması tekniği for-döngülerinin tamamen açılmasını veya kısmen açılmasını sağlar. Şekil 7.6'da döngü açma işlemi gösterilmiştir. Şekil 7.6'daki for döngüsü 3 iterasyona sahip bir döngüdür. Her iterasyonda bir işlem yapılmaktadır. Döngü açılmadan önce, işlemler sırayla yapılmakta ve oluşturulan kaynak her iterasyon için sıra ile kullanılmaktadır. Fakat döngünün açılması ile döngü içerisindeki birbirinden bağımsız yapılabilen işlemler paralel bir şekilde yürütülmektedir. Bu durum işlemlerin daha kısa sürede tamamlanmasını sağlayacak, fakat daha fazla kaynak kullanılmasına neden olacaktır.



Şekil 7.6. Döngü açma eniyileme yönteminin döngü üzerindeki etkisi

### Dizilerin parçalanması

Diziler, FPGA tabanlı ESA hızlandırma yöntemlerinin en önemli ögesidir. Diziler, FPGA’da blok rasgele erişim belleği (BRAM) olarak gerçekleştirilir. BRAM yapıları aynı anda maksimum iki veriyi hesaplama birimlerine gönderebilir. Bu durumda hesaplama işlemleri paralel bir şekilde yapılsa bile, bu birimleri besleyen veriler paralel bir şekilde gelmediği için bir darboğaz sorunu oluşacaktır. Veri akışının paralel bir şekilde sağlanabilmesi için FPGA’da bulunan BRAM modülleri parçalanarak daha küçük BRAM birimlerine veya saklayıcı (register) birimlerine bölünebilirler. Bu olaya dizi parçalama (array partitioning) denmektedir. Böylece, RTL tasarımında, tek bir büyük bellek yerine birden çok küçük bellek ve birden çok saklayıcı (register) kullanılabilir.



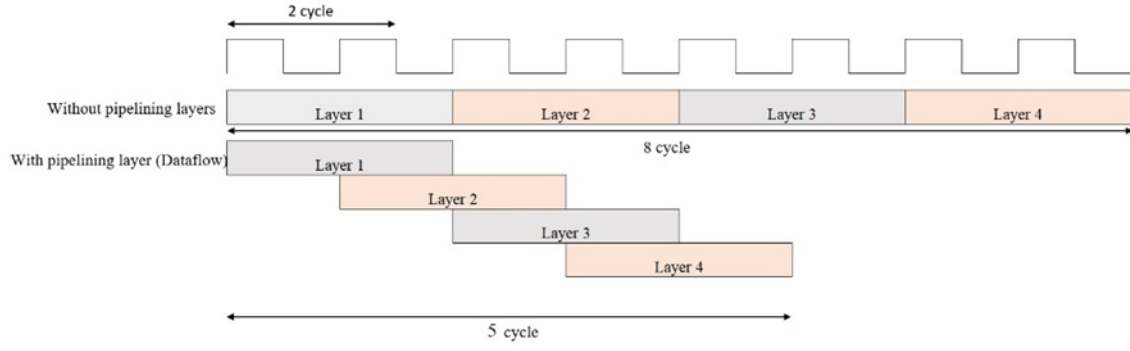
Şekil 7.7. HLS tabanlı dizi bölme yöntemleri

Vivado HLS, Şekil 7.7’de gösterildiği gibi üç farklı şekilde dizileri parçalara ayırma imkânı sunar. Bunlar:

- **block:** Orijinal dizi, orijinal dizinin ardışık öğelerinden oluşan eşit boyutlu bloklara bölünür.
- **cyclic:** Orijinal dizi, orijinal dizinin öğelerini serpiştirerek eşit boyutlu bloklara bölünür.
- **complete:** Dizinin bütün elemanlarını ayrı ayrı böler.

Dizilerin parçalanması donanım tabanlı uygulamalar için verimi artırır, veri akışında oluşacak darboğazı engeller ve işlemlerin paralel yapılma performansını artırır. Fakat kullanılan donanım kaynak miktarını da artırır.

### Veri akışı (katman düzeyinde paralellik)



Şekil 7.8. Veri akış paralelliğinin katmanlara uygulanması

Veri akışı sıralı döngülerin aynı anda çalışmasına izin verir. ESA uygulaması için veri akışı (dataflow) tekniği katmanlar arası kısmi paralelliği sağlar. Bir katmanın bütün çıktıları oluşmadan, sonraki katmanın çalışmasını sağlayarak kısmi paralellik sağlar. Şekil 7.8’de katmanlar arasında veri akış paralelliği uygulanıp uygulanmadığı durumlar gösterilmiştir.

### Gecikme optimizasyon yöntemleri

Gecikme, bir işlemi tamamlamak için geçen süredir. Zaman birimleriyle ölçülür ve genellikle milisaniye, mikro saniye ve nanosaniye birimlerinde bulunur. Gecikme iyileştirme yöntemleri işlemlerin paralel yapılmasıyla sağlanabilir. Yukarıda sayılan döngü açma, boru hattı paralelliği aynı zamanda gecikme iyileştirme işlemleridir.

### Donanım kaynak kullanımı optimizasyon yöntemleri

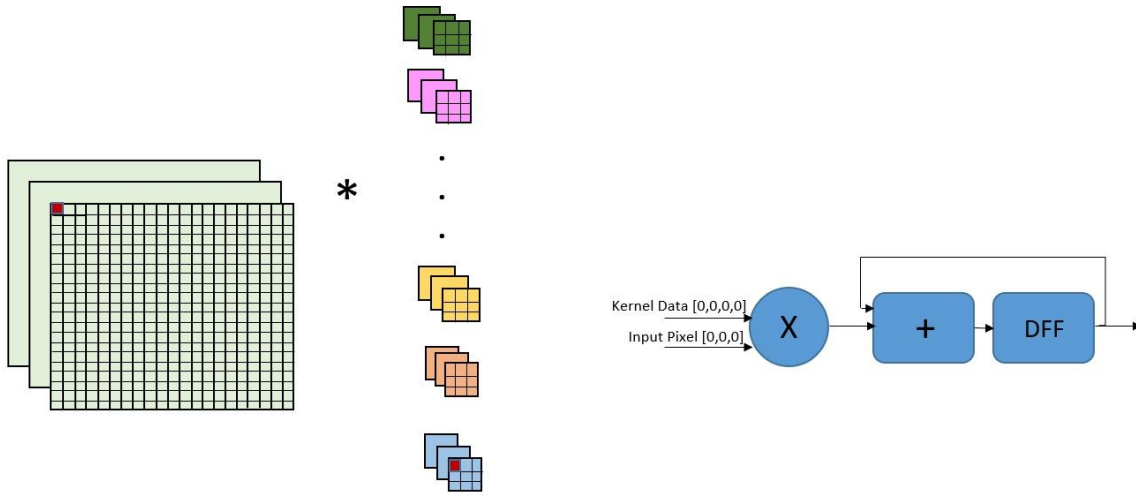
RTL uygulamasında kullanılan sayı gösterim biçimi ve sayı bit uzunluğu hem kullanılan depolama birimlerin miktarını hem de evrişim işlemi yapmak için kullanılacak DSP birimlerinin sayısını artıracaktır. Ayrıca işlemlerin daha yavaş yapılmasına ve dolayısıyla gecikmenin artmasına neden olacaktır. Kullanılacak sayı formatının doğru seçilmesi ile donanım kaynak kullanımı azaltılabilir.

### 7.3.3. Paralellik yöntemleri

ESA algoritmasının temel işlemi olan evrişim işlemi, IFM ve filtreler arasında gerçekleşen üç boyutlu çarpma ve biriktirme (multiply and accumulate, MAC) işlemlerini içermektedir. FPGA üzerinde yapılan gerçeklemelerin verimli bir şekilde yapılabilmesi için, evrişim işlemini yapan döngülerin optimize edilmesi gerekmektedir. Döngüler üzerinde yapılan en önemli üç tane optimizasyon tekniği bulunmaktadır. Bunlar döngü açma, döngü gruplama ve döngüler arası sıraların değiştirilmesi yöntemleridir.

Döngü açma yöntemi, donanım tabanlı ESA uygulamasının paralellik derecesini belirler. Döngü gruplama ise kullanılacak sistem üzerindeki bellek kapasitesini belirler. Döngüler arası sıraların değiştirilmesi ise 4 döngü arasındaki hesaplama sırasını belirler aynı zamanda da kullanılacak verilerin akışını da belirler. FPGA tabanlı ESA için farklı paralellik yöntemleri vardır. Bunlar döngüler üzerinde yapılan paralellik yöntemleri ve katmanlar arasında paralellik yöntemleri olarak ikiye ayrılabilir.

Döngüler açılmadan önce yapılan işlemler:

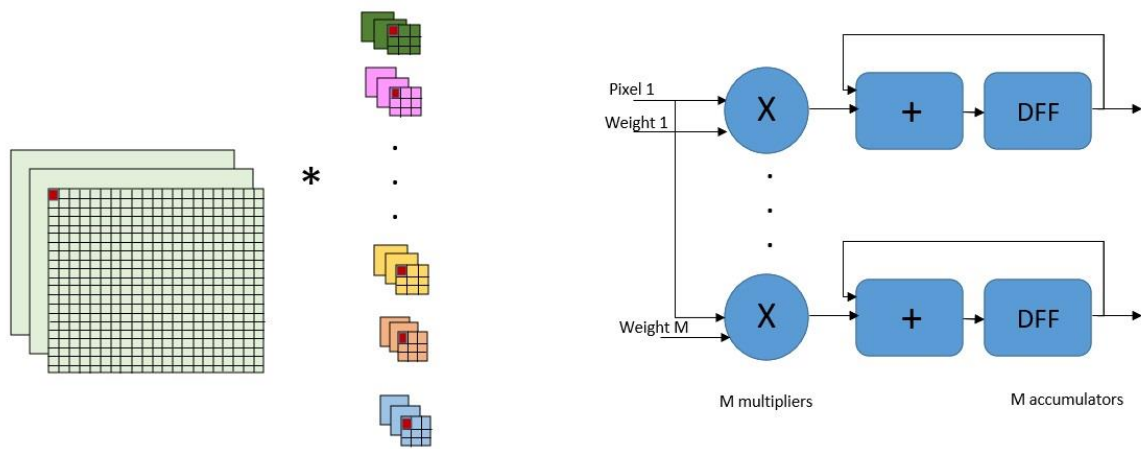


Şekil 7.9. Eniyileme yapılmadan önce evrişim işlemi ve karşılık gelen donanım yapısı

Şekil 7.9’da döngülerin açılmaması yani sıralı işlem yapmanın etkisi ve bu yapıya denk gelen hesaplama yapısı gösterilmiştir. Bu aşamada her işlem sıralı yapılmaktadır. Sırayla birinci filtrenin bir ağırlığı, IFM’nin birinci kanalının bir pikseli ile çarpılmaktadır. Son filtrenin son ağırlığı IFM’nin son kanalının son ağırlığıyla çarpıldıktan sonra evrişim işlemi tamamlanmaktadır. Bu yapı ile yapılan evrişim işleminin bir görüntü için bütün

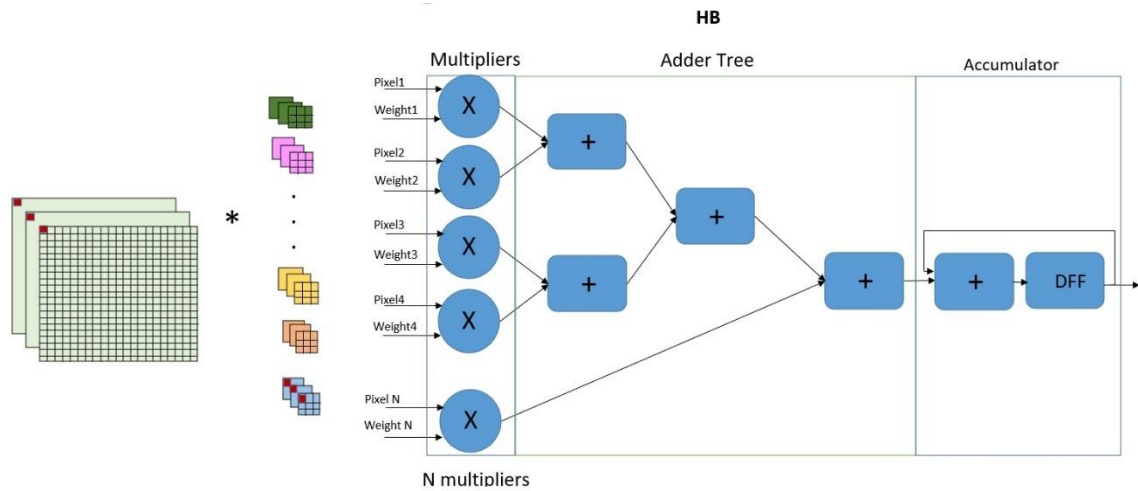
OFM'lerinin oluşması için gereken işlem sayısı  $R \times C \times M \times N \times k \times k$  dir. Kullanılacak MAC birimlerinin sayısı verilerin sayı gösterim yöntemine ve sayı gösterim uzunluğuna bağlıdır. Bu bölümde sayı gösterim yönteminin etkisi incelenmediği için, her işlemin yukardaki şekilde gösterilen bir birimi kullandığı varsayılmıştır. Buna göre açılmayan döngü sıralı işlemlerin yapıldığı evrişim işlemi için 1 tane çarpım, 1 tane toplam ve bir tane biriktirici kullanılmıştır.

Döngü seviyesinde yapılan paralellik işlemleri: L2 döngüsünün açılması (Şekil 7.10):



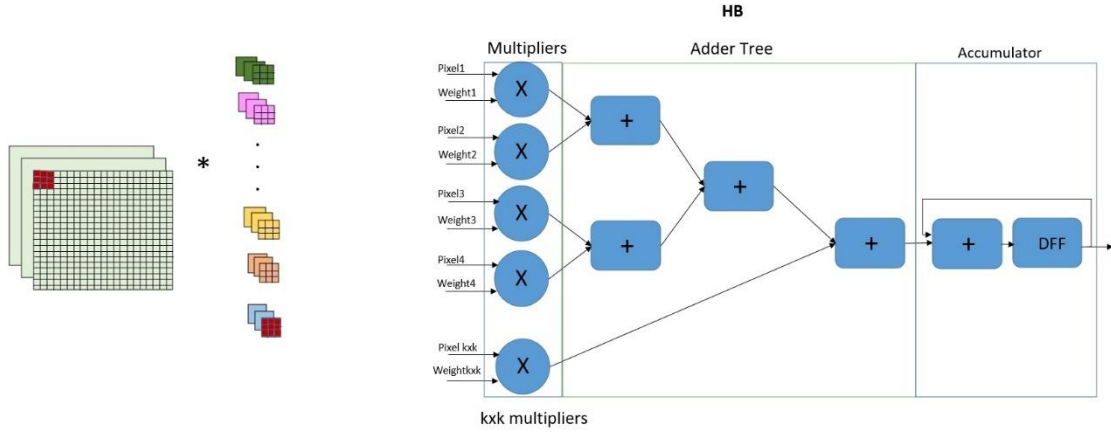
Şekil 7.10. L2 döngüsünün açılması ve karşılık gelen donanım yapısı

L2 döngüsünün açılması oluşturulacak OFM sayısı kadar L3 döngüsünün açılması (Şekil 7.11):



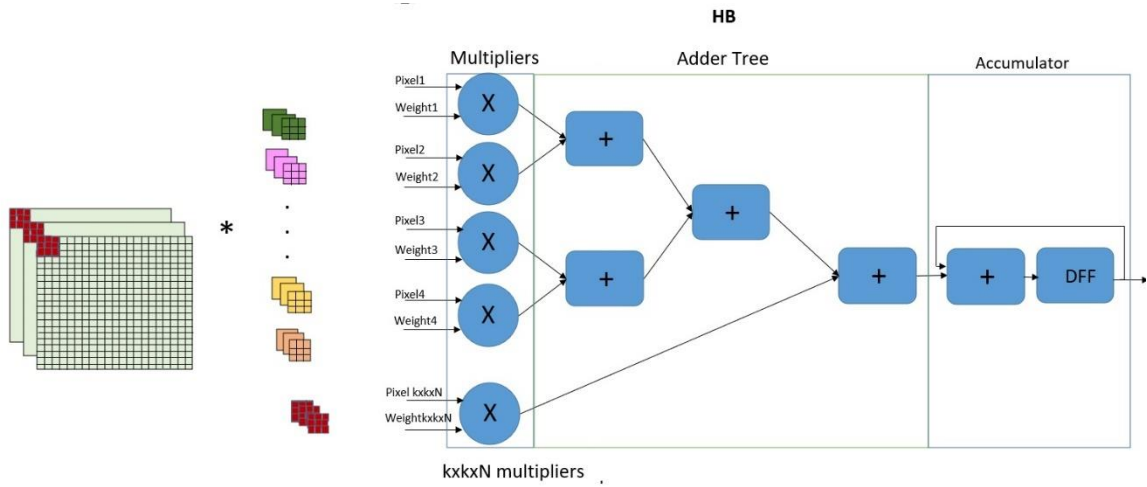
Şekil 7.11. L3 döngüsünün açılması ve karşılık gelen donanım yapısı

L4 döngüsünün açılması (Şekil 7.12):



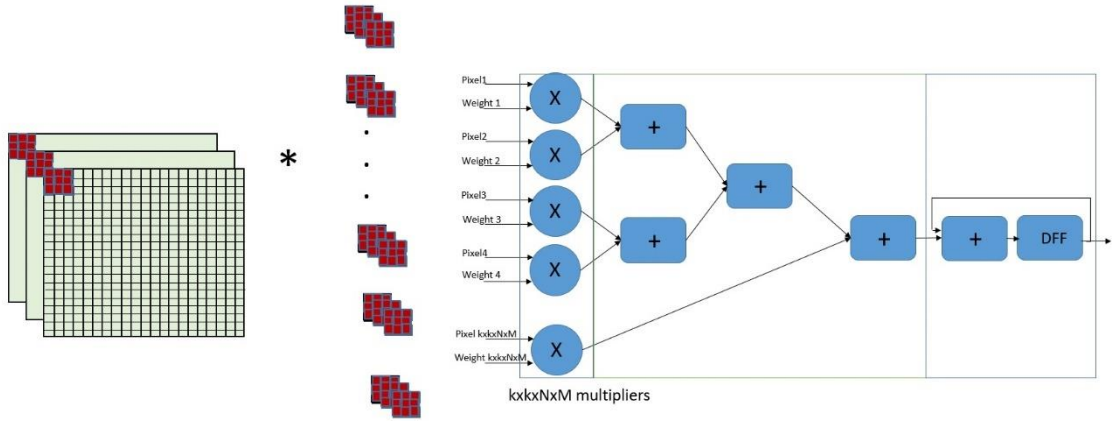
Şekil 7.12. L4 döngüsünün açılması ve karşılık gelen donanım yapısı

L4 ve L3 döngülerinin beraber açılması (Şekil 7.13):



Şekil 7.13. L3 ve L4 döngülerinin beraber açılması ve karşılık gelen donanım yapısı

L4, L3 ve L2 döngülerinin beraber açılması (Şekil 7.14):



Şekil 7.14. L2, L3 ve L4 döngülerinin beraber açılması ve karşılık gelen donanım yapısı

#### 7.4. FPGA Tabanlı ESA Hızlandırıcı Modelinin Değerlendirme Ölçütleri

Donanım kaynaklarının değerlendirme ölçütü olarak, arama çizelgeleri (LUT), saklayıcılar (flip-flops, FFs), blok RAM'ler ve DSP48'ler dahil olmak üzere FPGA'da bulunan kaynaklara dayalı olarak tasarımı uygulamak için gereken donanım kaynaklarının miktarı kullanılır.

DSP48 modülü üç farklı bloktan oluşan, FPGA'nın yapısında gömülü olarak bulunan aritmetik mantık birimidir. DSP48 bloğu temel olarak bir 18x18-bit ikiye tümleyen çarpandan ve ardından 48-bit genişletilmiş toplayıcı/-çıkarcı veya biriktirici birimden oluşur.

FPGA uygulamalarında hesaplama değerlerini okumak veya depolamak için bellek birimlerinin kullanılması gerekmektedir. FPGA'da bulunan gömülü bellek blokları, bu gibi durumlarda kullanılabilen çip üzerindeki belleklerdir. FPGA yapısı, rastgele erişim belleği (RAM), salt okunur bellek (ROM) veya kaydırma yazmaçları olarak kullanılabilen gömülü bellek öğeleri içerir. Bu öğeler, blok RAM'ler (BRAM'ler), LUT'lar ve kaydırma yazmaçlarıdır. BRAM, nispeten büyük bir veri kümesi için çip üzerinde depolama sağlamak üzere FPGA üzerinde gömülü olarak bulunan RAM modülüdür. İki tip BRAM modülü bulunmaktadır. Bunlar 18 K ve 36 K'dır. 18 Kbit ve 36 Kbit veri boyutlarına kadar depolayabilen bu yapılar farklı şekillerde konfigüre edilebilirler. Her 18 Kb BRAM;

- 1K × 18-bit

- $2K \times 9\text{-bit}$
- $4K \times 4\text{-bit}$
- $8K \times 2\text{-bit}$
- $16K \times 1\text{-bit}$
- $512 \times 36\text{-bit}$

olarak oluşturulabilir. Örneğin,  $512 \times 32\text{-bit}$  verinin depolanması için kullanılacak BRAM sayısının hesaplanması:

$512 \times 32 = 16.416$  bir veri yapmaktadır. Yukardaki seçeneklerden  $512 \times 36\text{-bit}$  yapısı seçildiğinde  $512 \times 32 = 18.432$  veri depolanabilir. Yukardaki yapıların kullanılması arka-arkaya (cascade) ve paralel mod olmak üzere iki şekilde yapılmaktadır. Arka-arkaya yapıda 2 tane  $512 \times 36$  bit yapısının kullanılması ile  $1024 \times 36\text{-bit}$  veri depolanabilir. Veya paralel mod kullanımı için 2 tane  $1K\text{-}18\text{-bit}$  yapısının kullanılması ile  $1K\text{-}36$  bit veri depolanabilir. Yani 36 bit gösterim kullanılarak  $1K$  veri depolanabilir.

$32 \times 32 \times 3 \times 32\text{-bit}$  için

$1K\text{-}18$  bit modu kullanılmak istenirse:

$32\text{-bit}$  yakalamak için iki tane paralel  $1K\text{-}18$  bit modülü kullanılır. Böylece  $1K\text{-}36$  bit veri depolama sağlanabilir. Fakat  $3072 > 1024$  ( $1K$ ) olduğu için bu yapı yeterli gelmeyecektir. O yüzden  $2K\text{-}9\text{-bit}$  yapısına bakılabilir. Burada da  $32$  biti yakalamak için 4 tane yapı kullanılması lazım. 4 tane  $2K\text{-}32$  bit ile  $2K\text{-}36$  bit veri saklanabilir. Bu da  $3072 > 2048$  olduğu için yeterli gelmeyecektir. Üçüncü olarak  $4K\text{-}4$  bit kullanılmak istenirse 8 tane gerekecektir.  $3072 < 41024$  olduğu için bu yapı uygundur.

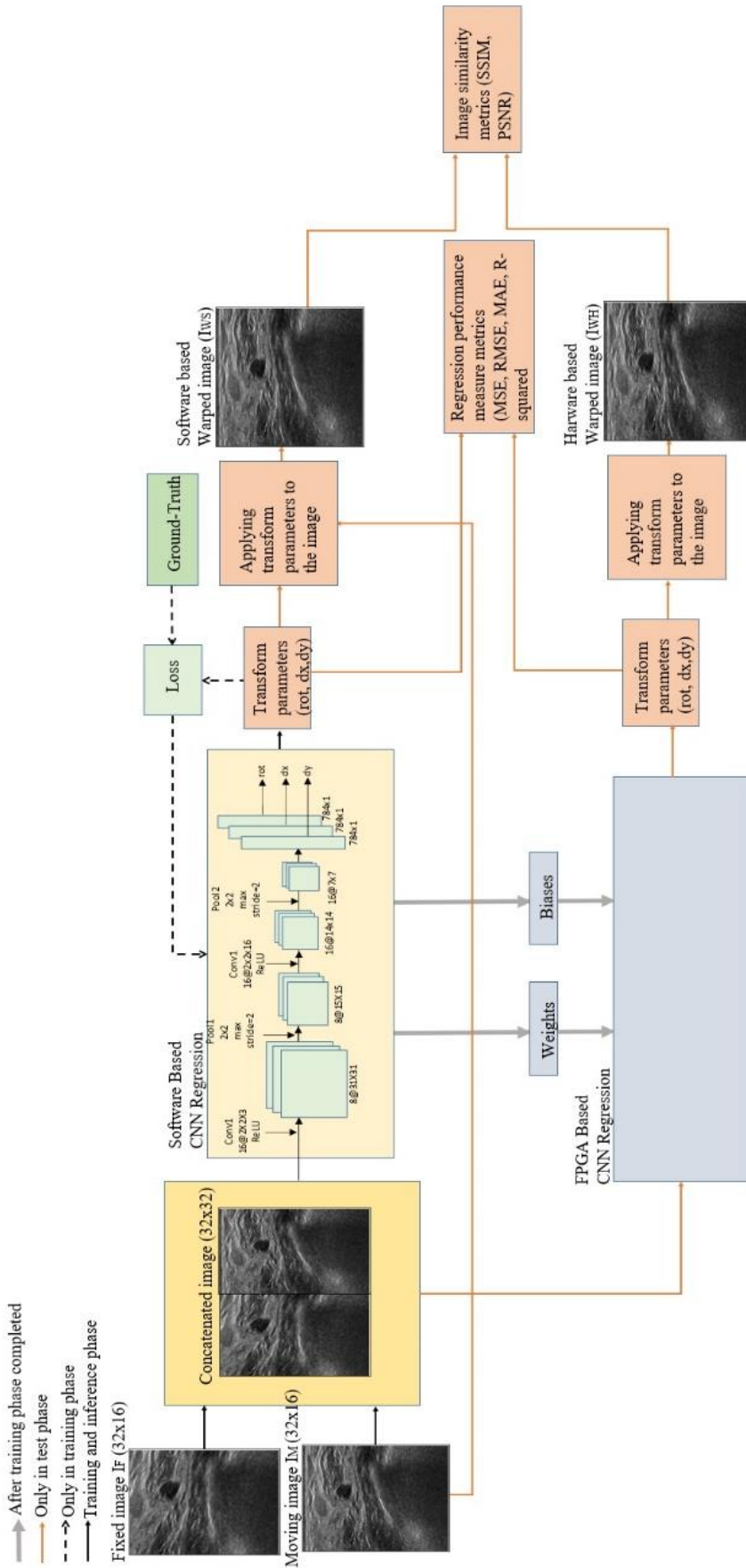
Gecikme (latency), fonksiyonun tüm çıkış değerlerini hesaplaması için gereken saat döngüsü sayısıdır.

Başlatma aralığı (Initiation interval, II), Fonksiyonun yeni giriş verilerini kabul etmesinden önceki saat döngüsü sayısıdır.

## **8. US-US KATI GÖRÜNTÜ ÇAKIŞTIRMA PARAMETRE TAHMİNİ İÇİN FPGA TABANLI HIZLANDIRILMIŞ ESA UYGULAMASI**

Bu tez çalışmasında yapılan çalışmalar yazılım ve donanım tabanlı yapılan çalışmalar olmak üzere iki kısımdan oluşmaktadır. Yazılım aşamasında US-US görüntülerinin çakıştırma parametrelerinin tahmini için evrişim katmanlarından oluşan ağı eğitimi yapılarak, en iyi tahmin sonuçlarını veren parametre ve hiper parametreler belirlenmiştir. Ağı eğitimi tamamlandıktan sonra eğitilen ağırlık ve bias değerleri donanım aşamasında kullanılmak üzere kaydedilmiştir. İkinci aşamada donanım üzerinde ESA modelini oluşturan evrişim, havuzlama ve tam bağlantılı çıkış katmanları teker teker kodlanarak C tabanlı ve C/RTL tabanlı doğrulaması yapılmıştır. Daha sonra yazılımda belirlenen modele göre bu katmanlar çoğaltılarak birleştirilmiştir. Birleştirme işleminden sonra donanımda kullanılacak sayı gösterim şeklinin belirlenmesi için tasarlanan ağı tahmin hassasiyeti-donanım kaynak kullanımı dikkate alınarak en uygun sayı gösteri şekli ve uzunluğu belirlenmeye çalışılmıştır. Daha sonra ise ağı hızlanması için farklı paralellik yöntemleri denenerek ağı hızlandırılma-kaynak kullanımı dikkate alınarak hızlandırma işlemi yapılmıştır. Tasarlanan ağ çıkarım aşaması için FPGA üzerinde gerçekleştirilmiştir. Son olarak da aynı ağı GPU, CPU ve FPGA üzerinde çıkarım aşamasında tükettikleri güç miktarları ölçülmüştür.

Uygulamanın genel hatları Şekil 8.1’de gösterilmiştir.



Şekil 8.1. Uygulamanın genel yapısı

## 8.1. Yazılım Tabanlı Uygulama

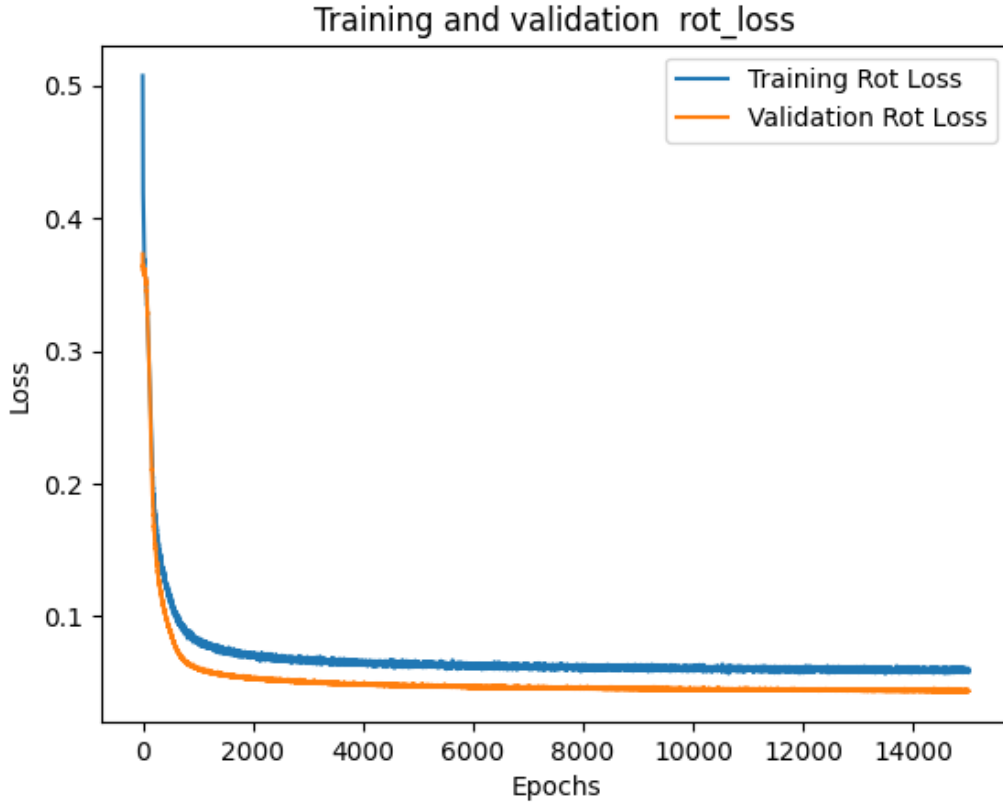
İlk aşama olarak yazılım tabanlı ESA ağının eğitimi tamamlanarak kullanılacak ağın parametreleri belirlenmiştir. Bu tez çalışmasında farklı veri setlerinde bulunan görüntüler kullanıldığı için bütün görüntülerin aynı boyuta getirilerek ağın beslenmesi sağlanmıştır. Veri artırma yöntemleriyle sayısı artırılan görüntüler 32x16x3 boyutuna getirilmiştir. Daha sonra sabit-hareketli ( $I_F-I_M$ ) görüntü çiftleri birleştirilerek 32x32x3 boyutuna getirilmiştir. Ağın girişi görüntü çiftinden oluşan yan yana birleştirilmiş tek bir görüntüyle beslenmiştir. Yığın sayısı ve devir (epok) sayısı yapılan farklı denemeler sonunda sırasıyla 50 ve 15.000 olarak belirlenmiştir. Ağ üç ayrı çıkış katmanından oluşmaktadır. Her katman bir tane çakıştırma parametresinin tahminini yapmaktadır. Bunlar döndürme (rotation, rot), x ekseninde kaydırma (dx) ve y ekseninde kaydırma (dy) değerlerinin tahmini olan üç ayrı katmandır. Döndürme miktarı  $[-11^\circ, 11^\circ]$  arasında, x ve y eksenlerinde kaydırma miktarı ise  $[-10, 10]$  piksel olarak sınırlandırılmıştır. Kullanılan görüntüler 0-1 arasında normalize edilerek ağa verilmiştir.

ESA modelinin eğitimi Intel(R) Core(TM) i7-10750H 2,60 GHz CPU ve 16,0 GB RAM üzerinde TensorFlow arka ucuna sahip Keras kullanılarak bir Python ortamında yapılmıştır. Optimizasyon yöntemi olarak Adam iyileştirici kullanılmış ve öğrenme oranı (LR) yapılan denemeler sonucunda 0,00001 olarak belirlenmiştir. Kayıp fonksiyonu olarak MSE kullanılmıştır. Bu çalışmada toplam 20.460 US B-mod görüntüsü kullanılmıştır. Görüntüler daha sonra 10.230 sabit-hareketli çifte bölünmüştür. Eğitim için toplam 8.184 çift, doğrulama için 1.842 görüntü çifti ve test için 204 görüntü çifti kullanılmıştır. Test için kullanılan görseller ağ eğitimi sırasında kullanılmamıştır. Eğitim sonunda belirlenen hiper parametreler Çizelge 8.1'de verilmiştir.

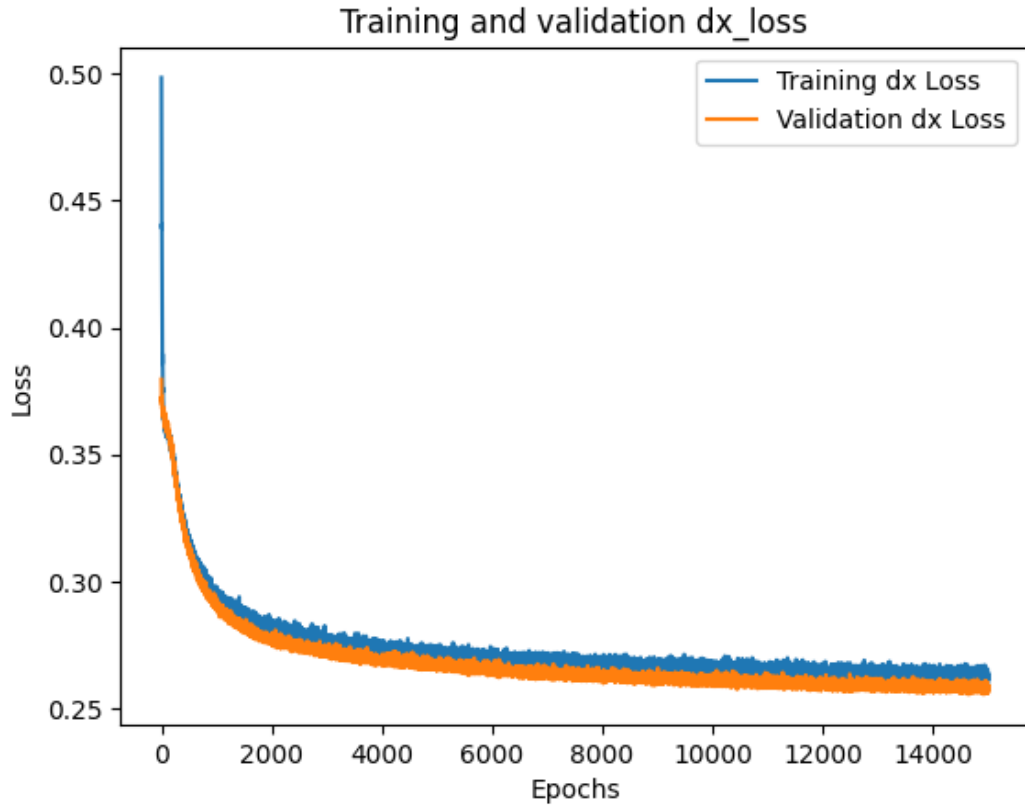
Çizelge 8.1. Eğitimi tamamlanan ağın hiper parametreleri

Katmanlar	Çıkış boyutları	Filtre boyutu	Filtre sayısı	Adım	Parametreler
Input	(32,32,3)				
Conv2D	(31,31,8)	(2,2)	8	1	104
MaxPooling2D	(15,15,8)	(2,2)		2	0
Conv2D	(14,14,16)	(2,2)	16	1	528
MaxPooling2D	(7,7,16)	(2,2)		2	0
Flatten	784				0
Dropout	784				0
rot	1				784
dx	1				784
dy	1				784
Toplam ve eğitilebilir parameter sayısı					2984

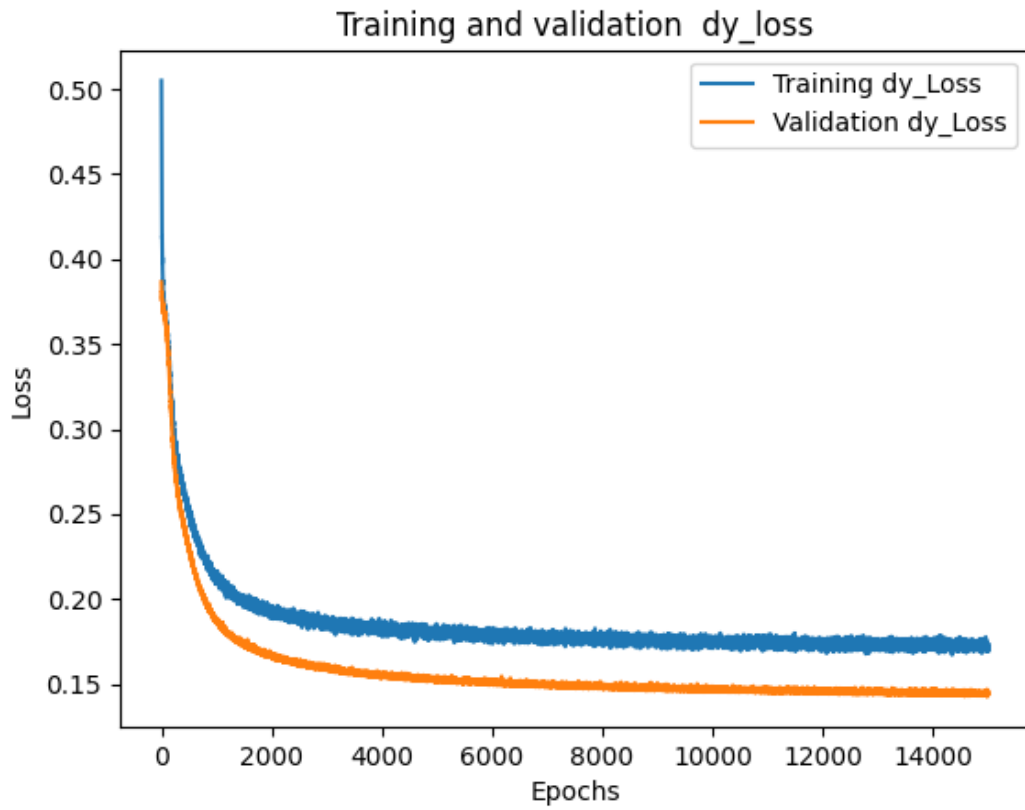
Hem aşırı öğrenmeyi engellemek hem de parametre sayısını azaltmak için katman sayısı az tutulmuştur. Farklı denemeler yapıldıktan sonra evrişim katmanı sayısı iki olarak belirlenmiştir. İki tane de maksimum havuzlama yöntemini kullanan havuzlama katmanı evrişim katmanlarından hemen sonra yerleştirilmiştir. Çıkış olarak rot, dx ve dy değerlerini tahminini veren üç ayrı katman bulunmaktadır. Evrişim katmanlarında aktivasyon fonksiyonu olarak ReLU kullanılmıştır. Çıkış katmanlarında aktivasyon fonksiyonu kullanılmamıştır. Ağın tahmin performansının ölçülmesi için her üç çıkış katmanı için de MSE fonksiyonu kullanılmıştır. Her katmanın ve ağın genel eğitim, doğrulama kayıp fonksiyonları Şekil 8.2, 8.3, 8.4 ve 8.5'te sırasıyla gösterilmiştir. Belirlenen ağın test görüntülerinde katı parametreleri tahmin performansı rot, dx ve dy için sırasıyla 0,0551 , 0,2802 ve 0,1374 olarak hesaplanmıştır.



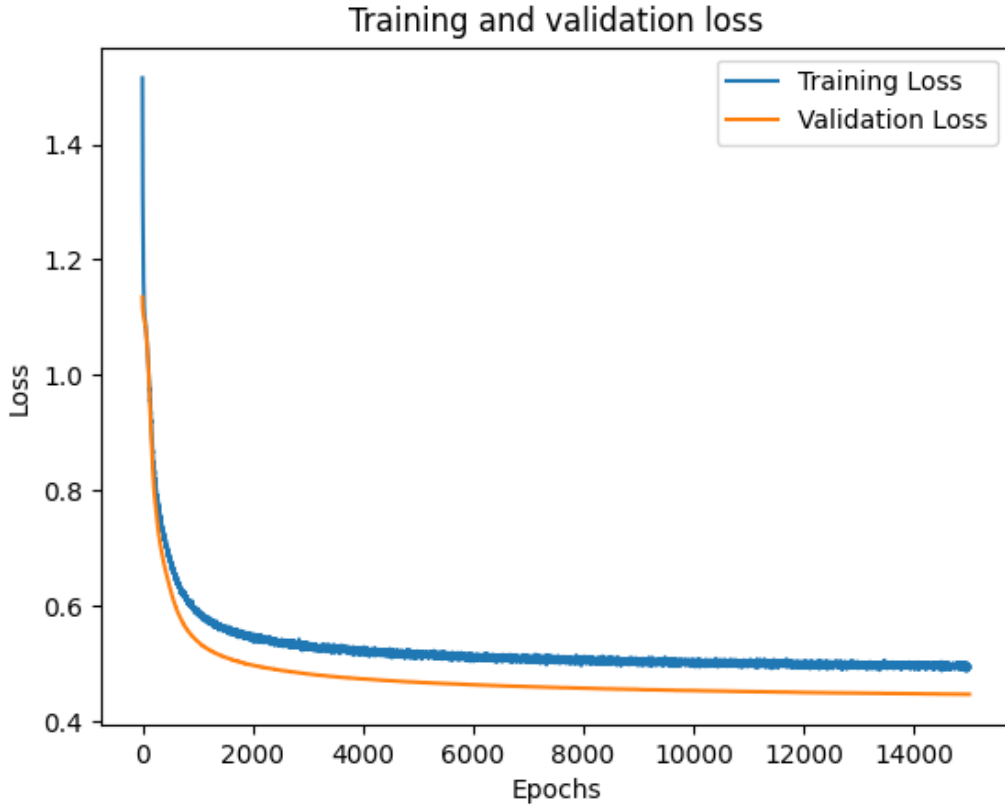
Şekil 8.2. rot çıkış katmanının eğitim ve doğrulama kayıp grafiği



Şekil 8.3. dx çıkış katmanının eğitim ve doğrulama kayıp grafiği



Şekil 8.4. dy çıkış katmanının eğitim ve doğrulama kayıp grafiği



Şekil 8.5. Ağın eğitim ve doğrulama kayıp grafiği

Şekillerde yatay eksen devir sayısını, dikey eksen ise MSE kayıp fonksiyonuna göre hesaplanan hataları göstermektedir. Farklı devir sayısı ve hiper parametreler için yapılan denemeler EK-1’de verilmiştir. Giriş görüntü boyutunun artırılması, seyreltme (dropout) katmanının kullanılması, evrişim katman sayısının artırılması, evrişim katmanlarında kullanılan filtre sayılarının artırılması ağın kayıp değerini azaltmaktadır. Fakat parametre sayısı yüksek olan ağın bu tez çalışmasında kullanılan FPGA üzerinde gerçekleştirilebilmesi FPGA kaynakları yetersiz kalmıştır. Bu nedenle parametre sayıları az olan yapı tercih edilmiştir. Ağın ezber yapmasını engellemek için seyreltme katmanı da kullanılmıştır.

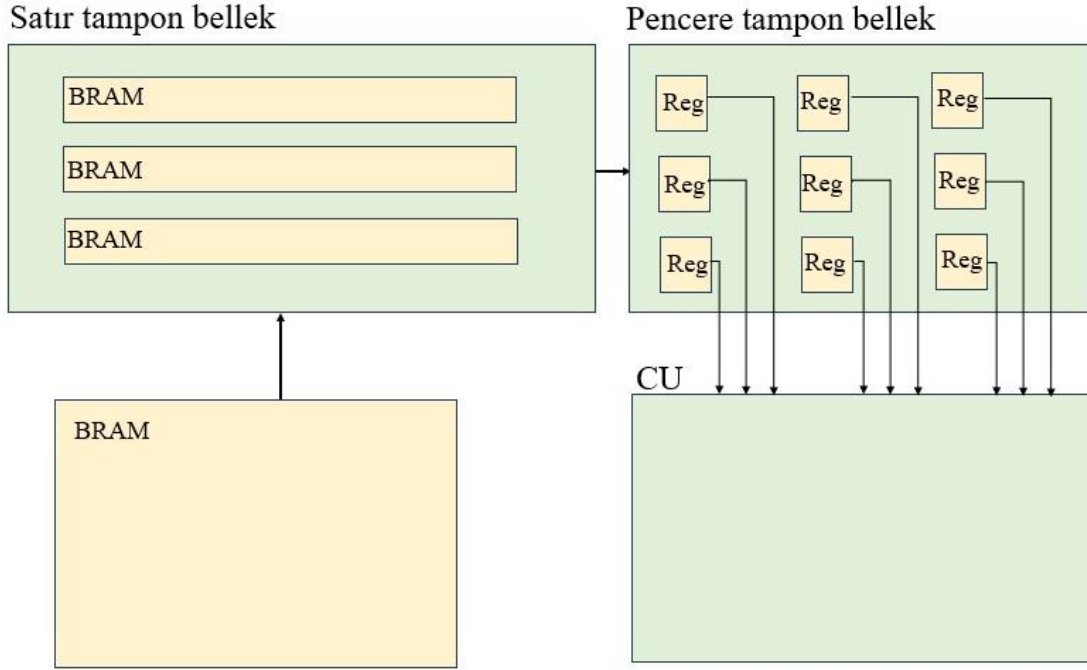
## 8.2. Donanım Tabanlı Uygulamalar

Yazılım aşamasında eğitimi tamamlanan ağın ağırlık ve bias değerleri noktadan sonra 6 basamak olacak şekilde kaydedilmiştir. FPGA gerçekleştirilmesinde kullanılacak katmanlar teker teker tasarlanarak arka arkaya bağlanmıştır. Her katmanın doğrulaması teker teker yapılmıştır. Tasarlanan katmanlara ait doğrulama sonuç ekran görüntüleri EK-2’de verilmiştir. EK-2’de verilen doğrulamalar esnasında sayı gösterim yöntemi olarak 32-bit FP kullanılmıştır. Evrişim katmanı, havuzlama katmanı ve tam bağımlı katmanların

tasarımları C++ dilinde yapıldıktan sonra bu katmanlar arka arkaya bağlanmıştır. İlk olarak kullanılacak sayı gösterim şekli ve sayı gösterim uzunluğu belirlenmiştir. Sonraki aşamada ise ağın işlemleri eşzamanlı yapabilmeye için döngü açma, dizi bölme eniyileme teknikleri uygulanarak ağın hızlandırılması sağlanmıştır. Modeller C++ dilinde geliştirilmiştir. Öncelikle C seviyesinde doğrulama, C/RTL co-simulasyonu ve RTL doğrulaması yapılmıştır. Doğrulan tasarım IP blokları haline getirilmiştir. Vivado 2018 kullanılarak IP paketler ve gerekli bağlantılardan sonra tasarım ZedBoard içerisine başarılı bir şekilde gömülmüştür. Belirlenen tasarımlar FPGA içerisinde gömülerek gerçek-zamanlı tasarım elde edilmiştir.

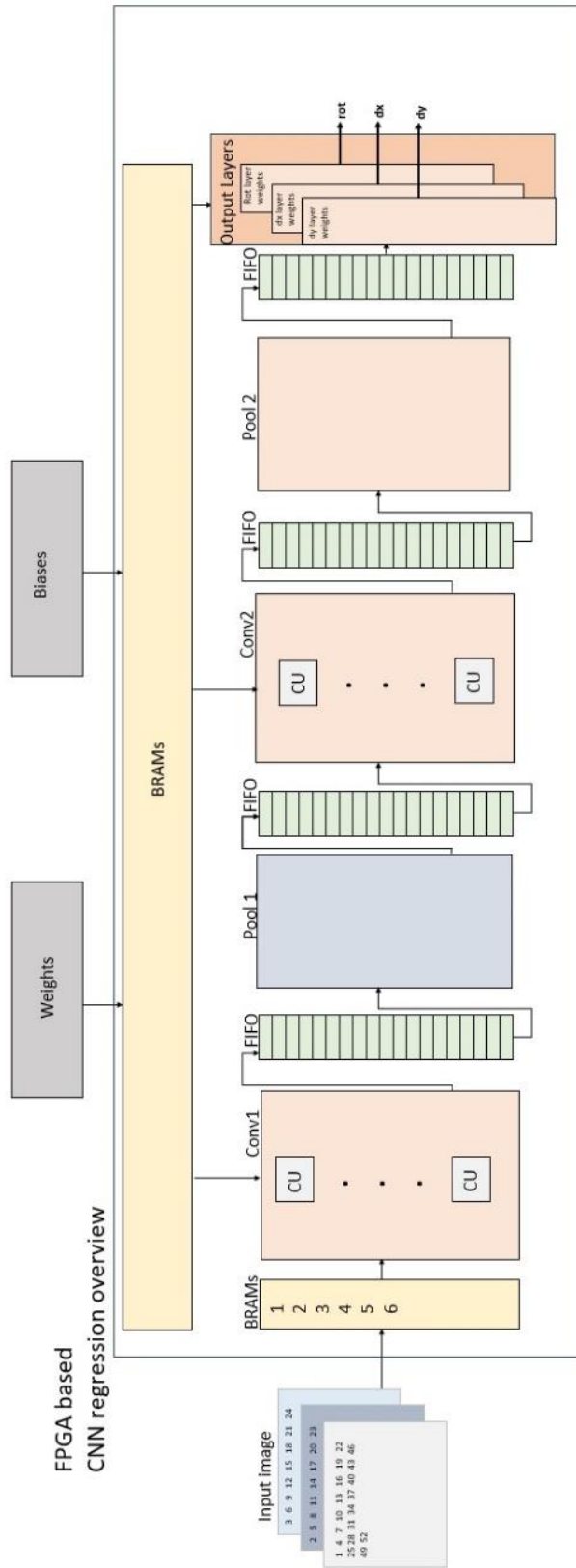
### **8.2.1. Evrişim katmanına verilerin hazırlanması**

Çıkarımı yapılan görüntü öncelikle BRAM hafıza birimine raster tarama düzenine göre depolanmıştır. BRAM birimlerinden satır tampon bellek birimlerine, paralel işlem yapılacak kadar veriler aktarılmaktadır. Uygulamalar göre yapılan paralel işlem sayısı değiştiği için kullanılan tampon bellek miktarı da değişmektedir. Satır tampon belleklere aktarılan veriler dizi parçalama eniyileme yöntemi kullanılarak bağımsız saklayıcı birimlerine ayrılmaktadır. Böylece eşzamanlı işlem yapmak için gereken verilerde işlem birimlerine paralel bir şekilde iletilebilecektir. Satır ve pencere tampon bellek birimlerinin genel yapısı Şekil 8.6'da gösterilmiştir.



Şekil 8.6. Satır ve pencere tampon bellek birimleri

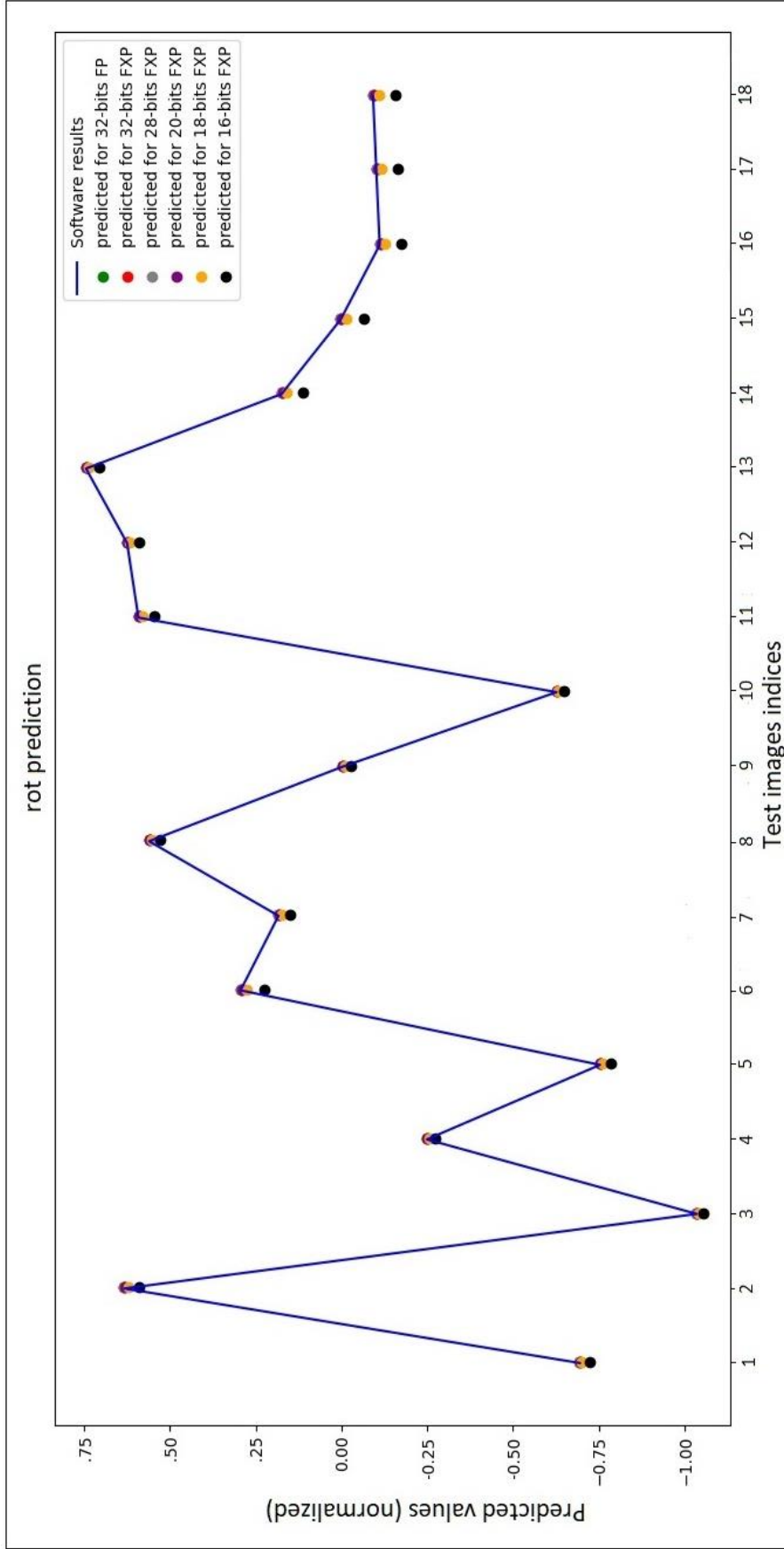
Şekil 8.7’de donanım tabanlı uygulamanın genel yapısı gösterilmiştir. Bu yapıda çıkarım yapılan görüntü, ağırlık ve bias değerleri BRAM hafıza birimlerine depolanmaktadır. İşlem yapılan katmandan çıkan yeni özellik haritaları FIFO birimlerine depolanmaktadır. Yazılım tabanlı yapılan uygulamada üç tane çıktı katmanı bulunmaktadır. Pool2 katmanından çıkan veriler üç tane tam bağlantılı katmana iletilmekte buradan da çıktılar oluşturulmaktadır. Pool2 katmanından çıkan veriler FIFO birimine depolandığı için, üç katman aynı anda verilere erişememektedir. Yani donanım tabanlı uygulamada aynı anda bu üç katmanın çıktısı elde edilememektedir. Bu sorun, FIFO’da bulunan verilerin tek bir katmana iletilmesi ve bu katmandan da üç çıkış vermesi şeklinde düzenlenerek çözülmüştür. Donanım tabanlı sonraki aşamaları eniyileme yöntemleri kullanılarak performans artırmaya yönelik devam etmektedir.



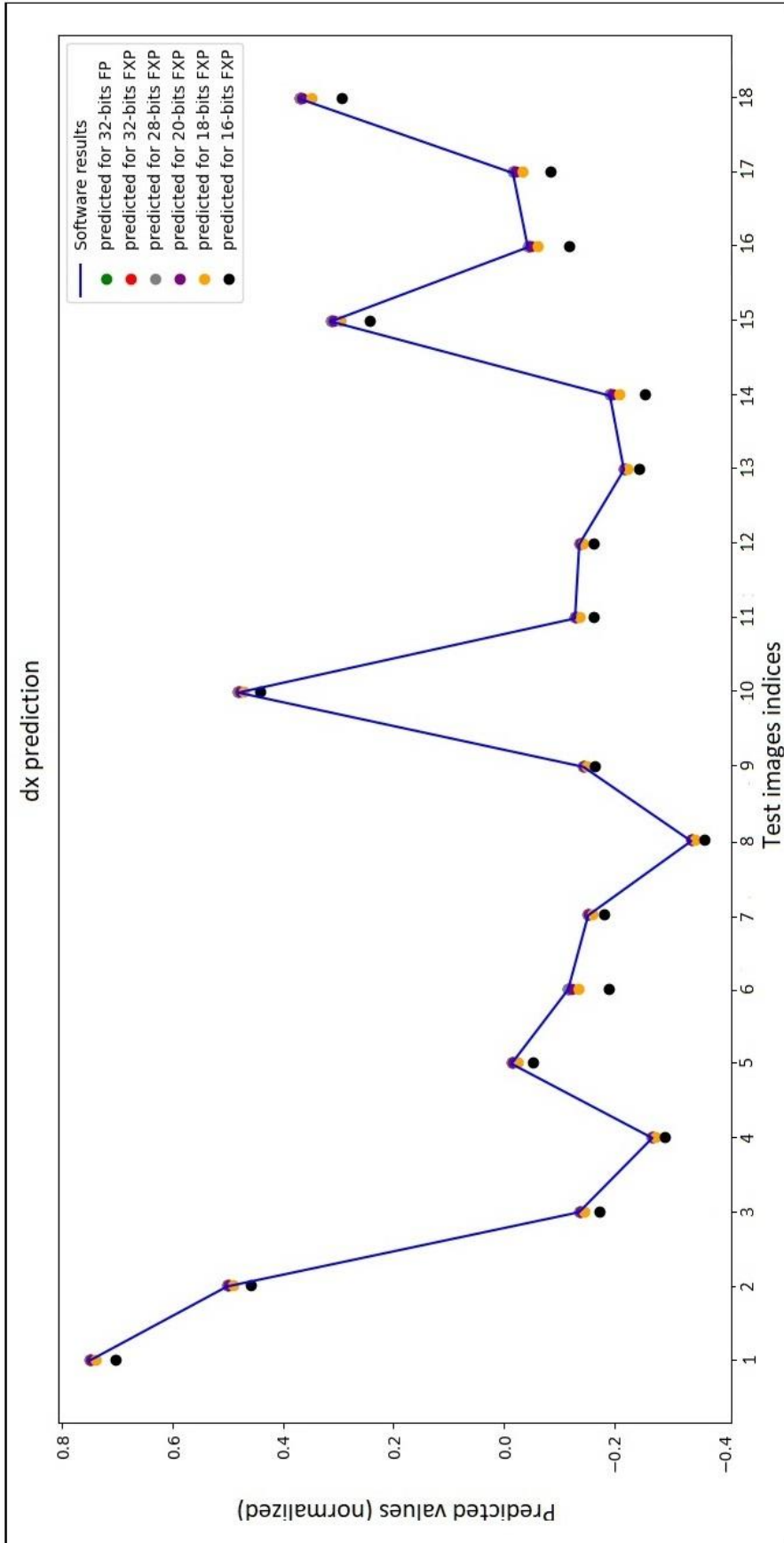
Şekil 8.7. Donanım tabanlı gerçekleştirme genel yapısı

### 8.2.2. Veri gösterim yöntemi ve veri gösterim bit uzunluğu

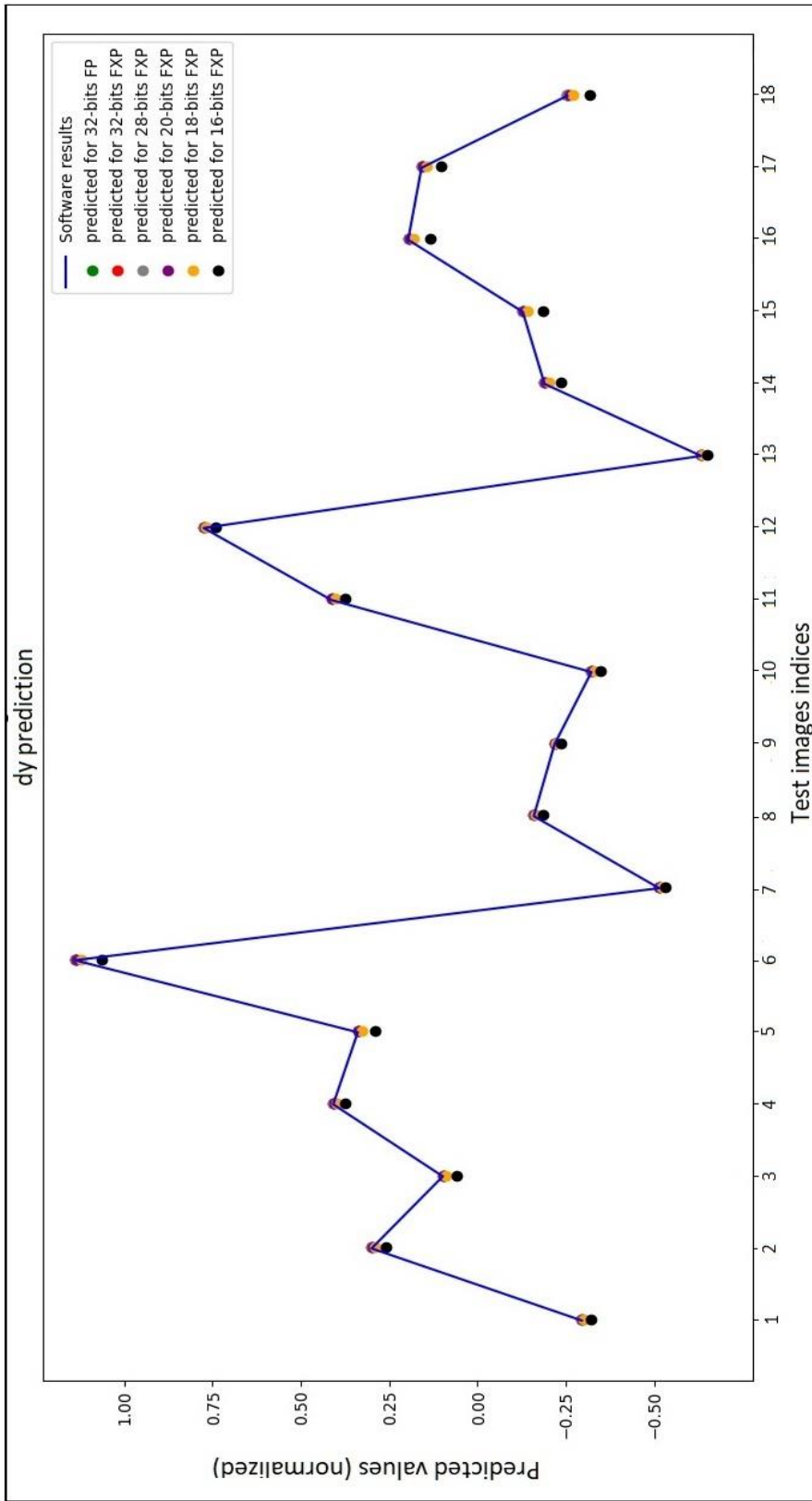
Veri gösterim yöntemi olarak 32-bit tek duyarlıklı kayan noktalı say gösterimi ve sabit noktalı sayı gösterim şekli olmak üzere iki farklı yöntem kullanılmıştır. Sabit noktalı sayı gösterim yöntemi için 32-, 28-, 20-, 18- ve 16- bit uzunlukları kullanılarak kaynak kullanımı, hız ve doğruluk parametreleri kıyaslanarak en uygun gösterim uzunluğu tespit edilmeye çalışılmıştır. Test işlemi için 204 I<sub>F</sub>-I<sub>M</sub> US B-mod görüntü çifti kullanılmıştır. Test başarısı hem yazılım hem de donanım tabanlı uygulamalar için karşılaştırılmıştır. 18 tane US B-mod görüntü için tahmin edilen yazılım ve donanım değerleri Şekil 8.8, 8.9 ve 8.10'da rot, dx ve dy çıkış katmanları için sırasıyla görselleştirilmiştir.



Şekil 8.8. Test görüntüleri rot çıkış katmanı için yazılım ve donanım tahmin sonuçları



Şekil 8.9. Test görüntüleri dx çıkışı katmanı için yazılım ve donanım tahmin sonuçları



Şekil 8.10. Test görüntüleri dy çıkış katmanı için yazılım ve donanım tahmin sonuçları

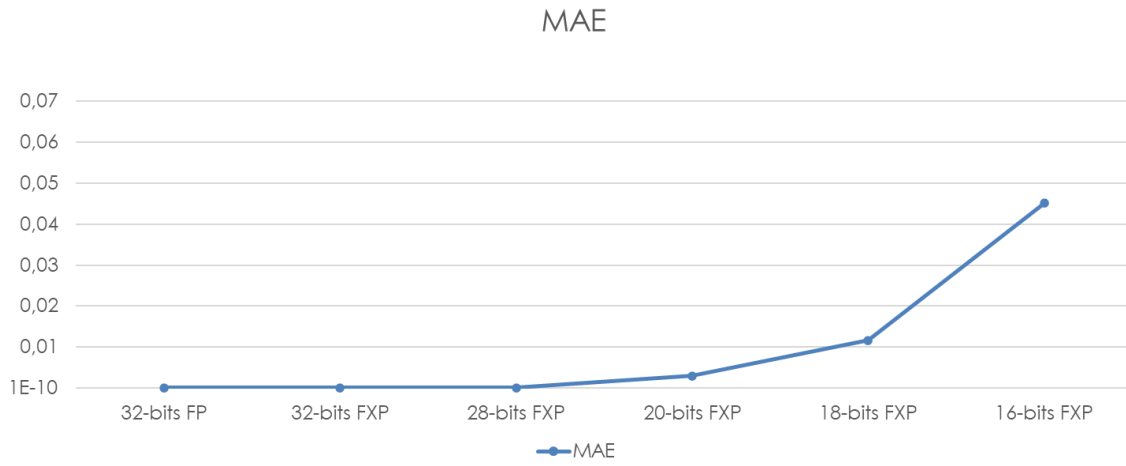
Şekil 8.8, 8.9 ve 8.10’da yatay eksen kullanılan US B-mod görüntü indisini, dikey eksenler ise tahmin edilen sonuçları göstermektedir. Sürekli mavi çizgi yazılım sonucundan tahmin edilen değerleri, yeşil, kırmızı, gri, mor, sarı ve siyah renkli noktalar ise sırasıyla 32-bit FP, 32-bit FXP, 28-bit FXP, 20-bit FXP, 18-bit FXP ve 16-bit FXP sayı gösterimleri kullanılarak elde edilen donanım tabanlı tahmin edilen sonuçları göstermektedir. Gerçek referans değerleri  $[-1,1]$  asarına normalize edildiği için tahmin edilen değerler de bu aralıkta çıkmıştır. Şekil 8.8, 8.9 ve 8.10 incelendiğinde sarı ve siyah noktalarla temsil edilen 18-bit FXP ve 16-bit FXP dışındaki bütün sayı gösterim yöntemlerinin yazılımla çok yakın değerler ürettiği görülmektedir. Sayıların gösterimi için kullanılan bit sayısı düştükçe hassasiyetin de azaldığı görülmektedir. Bu uygulama için özellikle 20-bitten daha az sayıda bitlerle temsil edilen değerlerin hassasiyeti büyük oranda düşürdüğü görülmüştür.

Çizelge 8.2. Uygulama sonuçlarının farklı değerlendirme metrikleriyle ölçülmesi

	MAE			MSE			RMSE			R-squared		
	rot	dx	dy	rot	dx	dy	rot	dx	dy	rot	dx	dy
32-bits FP	1,412e-06	1,758e-06	2,011e-06	2,997e-12	5,508e-12	2,474e-06	1,731e-06	2,346e-06	2,474e-06	0,9999	0,9999	0,9999
32-bits FXP	1,555e-06	1,599e-06	2,366e-06	3,370e-12	4,086e-12	7,311e-12	1,835e-06	2,021e-06	2,703e-06	0,9999	0,9999	0,9999
28-bits FXP	1,152e-05	1,089e-05	1,086e-05	1,450e-10	1,308e-10	1,299e-10	1,204e-05	1,144e-05	1,139e-05	0,9999	0,9999	0,9999
20-bits FXP	0,0026	0,0029	0,0025	8,303e-06	1,053e-05	7,772e-06	0,0021	0,0032	0,0027	0,9999	0,9998	0,9999
18-bits FXP	0,0106	0,0116	0,0099	0,0001	0,0001	0,0001	0,0114	0,0126	0,0106	0,9995	0,9982	0,9994
16-bits FXP	0,0425	0,0451	0,0399	0,0021	0,0024	0,0018	0,0452	0,0493	0,0434	0,9995	0,9982	0,9994

Çizelge 8.2’de FP ve FXP sayı gösterim teknikleri kullanılarak gerçekleştirilen FPGA tabanlı uygulamaların ve yazılımdan elde edilen değerler arasındaki farkın 204 US B-mod test görüntü çifti için farklı metriklerle değerlendirme sonuçları gösterilmiştir. Çizelge incelendiğinde yazılım uygulaması sonucu tahmin edilen değerlerle 32-bit FP sayı gösterimi kullanılarak gerçekleştirilen FPGA uygulaması arasındaki farkın her üç çıkış katmanı içinde düşük olduğu görülmektedir. MAE metriğine göre rot, dx ve dy çıkışları için sırasıyla yazılımla üretilen sonuçlar arasında 1,412e-06, 1,758e-06 ve 2,011e-06 fark olduğu görülmektedir. Yazılım aşamasında ağırlık eğitimi tamamlandıktan sonra ağırlık ve bias değerleri noktadan sonra 6 basamak alınarak kaydedilmiştir. Yazılım sonucu ve 32-bit FP donanım uygulamasının arasındaki farkın büyük bir oranı buradan kaynaklanmaktadır. Buna rağmen, iki uygulama arasındaki farkın çok az olduğu görülmektedir. Aynı şekilde MSE ve RMSE metriklerine göre de 32-bit FP donanım tabanlı uygulama ile yazılım tabanlı uygulamaların tahmin ettikleri değerler arasındaki farkın çok düşük olduğu görülmektedir.

32-bit FXP donanım tabanlı uygulama ile yazılım tabanlı uygulama kıyaslandığında yine tahmin edilen değerler arasındaki farkın çok az olduğu görülmektedir. 32-bit FP ile 32-bit FXP arasında da dikkate alınmayacak kadar az bir farkın olduğu görülmektedir. Bu sonuçtan da kayan noktalı sayılar kullanmanın sabit noktalı sayılar kullanmaya göre daha hassas olduğu görülmektedir. FXP sayı gösterim tekniği için kullanılan bit sayısı azaldıkça hassasiyetinde azaldığı görülmektedir. Özellikle 16-bit kullanıldığında yazılım ve donanım uygulamaları arasındaki tahmin farkının arttığı görülmektedir. R-squared metriği incelendiğinde ise 20-bit e kadar olan uzunluk kullanıldığında benzerliğin %99,99 olduğu, 18-bit ve 16-bit FXP içinde benzerliğin %99,95-%99,82 arasına düştüğü görülmektedir.



Şekil 8.11. Farklı sayı gösterim yöntemlerine göre hesaplanan MAE

Şekil 8.11 Farklı sayı gösterim yöntemlerine göre dx çıktı katmanı için hesaplanan MAE değerleri çizgi grafik üzerinde gösterilmiştir. Grafikten de anlaşılacağı üzere, sayı temsili için kullanılan bit sayısının azalması, hassasiyeti de azaltmaktadır. Oluşan hata miktarı da artmaktadır.

Hassasiyet analizi yapıldıktan sonra kullanılan donanım kaynak miktarları incelenmiştir. Tez çalışmasında gerçekleştirme için iki kullanılan iki FPGA kartında bulunan toplam donanım kaynak miktarları Çizelge 8.3'te gösterilmiştir.

Çizelge 8.3. FPGA donanımlarında bulunan toplam kaynak miktarları

	BRAM	DSP48E	FF	LUTs	URAM
ZedBoard	280	220	106400	53200	-
KCU 116	960	1824	433920	216960	64

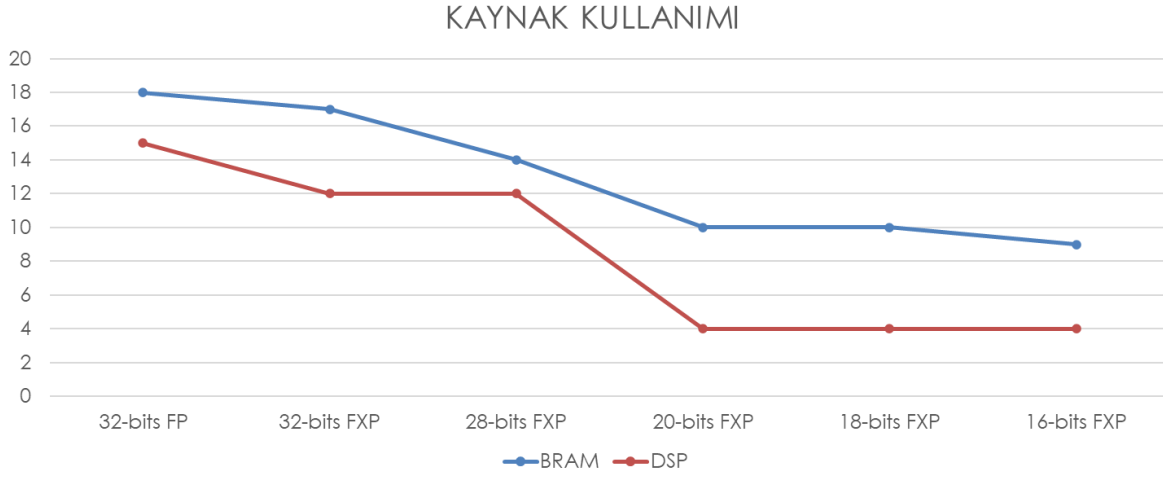
Çizelge 8.4 farklı uygulama ve farklı FPGA geliştirme kartları için kullanılan kaynak miktarları gösterilmiştir.

Çizelge 8.4. İki farklı FPGA platformlarının aynı uygulamalar için kaynak tüketimler ve gecikmeler açısından karşılaştırılması

	ZEDBOARD (XC7Z020CLG484-1)					KINTEX KCU116 (XCKU5P-FFVB676-2-E)				
	BRAM18k	DSP48E	FFs	LUTs	Latency(cc)	BRAM18k	DSP48E	FFs	LUTs	Latency(cc)
32-BITS FP	18	15	33069	12833	1006610	16	15	32555	10941	714466
32-BITS FXP	17	12	31452	5926	512657	14	12	30982	5884	320457
28-BITS FXP	14	12	27472	5893	420401	13	12	27166	5878	320457
20-BITS FXP	10	4	19730	5491	420401	10	4	19534	5460	320457
18-BITS FXP	10	4	17795	5417	420401	10	4	17626	5386	320457
16-BITS FXP	9	4	15871	5367	420401	8	4	15718	5328	320457

Çizelge 8.4'te kullanılan BRAM, DSP, FF, LUT kaynak sayıları ve sonucun üretilmesi için gereken toplam süre çevrim sayısı (clock cycle, cc) olarak verilmiştir. Aynı ESA modelinin 32-bit FP sayı gösterim tekniği kullanılarak ZedBoard ve Kintex KCU116 FPGA geliştirme donanımlarında gerçekleşmesi incelendiğinde, farklı kaynak miktarı kullanıldığı ve farklı gecikme miktarına neden olduğu görülmektedir. Bunun nedeni, farklı FPGA geliştirme kartlarının konfigürasyonunun farklı olmasından kaynaklanmaktadır. 32-bit FP yüksek hassasiyet sağlamakla beraber daha fazla kaynak kullanımına ve daha geç sonuç üretilmesine neden olduğu görülmektedir. ZedBoard için 32-bit FP 1.006.610 çevrim sonra sonuç üretmişken 32-bit FXP 512.657 çevrim sonra değer üretmiştir. Aradaki fark yaklaşık olarak 493.953 çevrim sayısı kadardır. Tek duyarlıklı kayan noktalı sayılarla işle yapmak sabit noktalı sayılara göre daha karmaşıktır. Bu nedenle kullanılan kaynak sayısı artmakta, aynı zamanda gecikme miktarının da artmasına neden olmaktadır. 28-bit, 20-bit, 18-bit ve 16-bit uzunluklarının kullanıldığı uygulamalara bakıldığında, gecikme miktarlarının aynı olduğu görülmektedir. 20-bit ve 18-bit FXP uygulamalarında kullanılan kaynak miktarlarının da aynı olduğu görülmektedir. Çizelge 8.2 ve Çizelge 8.3 beraber dikkate alındığında, kullanılan kaynak sayısı, hassasiyet ve gecikme miktarları açısından 20-bit FXP kullanılmasının bu tez çalışmasının devamı için daha uygun olduğu sonucuna varılmıştır. Çünkü tez çalışmasında gerçek zamanlı uygulama için kullanılacak FPGA kartı ZedBoard geliştirme kartıdır. ZedBoard geliştirme kartının üzerinde daha sonraki aşamalarda yapılacak olan optimizasyonlar yapıldıktan sonra gerekli DSP kaynağı

bulunmamaktadır. Bu nedenle kaynak kullanımının az olduğu ama hassasiyetinde makul bir düzeyde olduğu 20-bit FXP sayı gösteriminin kullanılmasına karar verilmiştir.



Şekil 8.12. Farklı sayı gösterim yöntemlerine göre kullanılan BRAM ve DSP kaynak miktarları

XC7Z020CLG484-1 gerçeeklemedesinde farklı sayı gösterim yöntemlerinin kullanılmasıyla tüketilen donanım kaynakları Şekil 8.12’de çizgi grafiği olarak gösterilmiştir. Grafikten görüldüğü üzere, bit kullanım oranıyla ve sayı gösterim yöntemleriyle kaynak kullanımı değişmektedir. Bit sayısının artması kullanılan kaynak sayısını da artırmaktadır.

Çizelge 8.5. Sıralı işlem yapan evrişim katmanlarında bir evrişim işlemi için kullanılan DSP kaynak miktarları

	32-bits FP	32-bits FXP	28-bits FXP	20-bits FXP	18-bits FXP	16-bits FXP
Conv1	5	3	3	1	1	1
Conv2	5	3	3	1	1	1

Çizelge 8.5’te de ESA de kullanılan iki evrişim katmanı için kullanılan DSP kaynak sayıları verilmiştir. Çarpma işlemleri yapıldığında, kullanılan DSP kaynak sayısı büyük oranda sayıları temsil etmek için kullanılan bit sayısıyla belirlenmektedir. Sıralı yapılan bir evrişim işleminde bir işlem yapmak için 32-bit FP uygulaması 5 tane DSP kaynağını aynı anda kullanmıştır. 32-bit ve 28-bit FXP uygulamalarında ise işlem başına üçer DSP kaynağı kullanılmıştır. 20-bit, 18-bit ve 16-bit uygulamalarında ise çarpma işlemi için birer tane DSP kaynağı kullanıldığı görülmektedir.

### 8.2.3. Hesaplama birimi iyileştirmeleri

FPGA tabanlı ESA hızlandırma işlemi için döngü açma, veri akış paralelliği, dizi bölme ve boru hattı teknikleri kullanılmıştır. Evrişim katmanında, havuzlama katmanında ve çıkış katmanında işlemlerin eşzamanlı yapılabilmesi için bu katmanların tasarımında kullanılan iç içe olan for döngülerine döngü açma tekniği uygulanmıştır. Böylece iç içe olan döngüler açılarak her iterasyonun ayrı ayrı yapılması sağlanmıştır. Bu döngüler varsayılan olarak sıralı işlem yapmaktadır. Döngü açma tekniğinde belirlenen derecede işlemler eş zamanlı yapılmaktadır.

Evrişim katmanında eşzamanlı çalışan birimlere verilerin de eş zamanlı gelmesi gerekmektedir. Bunu içinde BRAM kaynaklarında bulunan veriler dizi bölme yöntemi kullanılarak daha küçük saklama alanlarında depolanmıştır. BRAM kaynaklarından aynı anda en fazla iki veri alınabilmektedir. Verilerin birbirinden bağımsız daha küçük BRAM kaynaklarına depolanmasıyla beraber, donanımda bulunan kaynak sayısını geçmeyecek şekilde istenilen sayıda verinin eş zamanlı iletimi mümkündür.

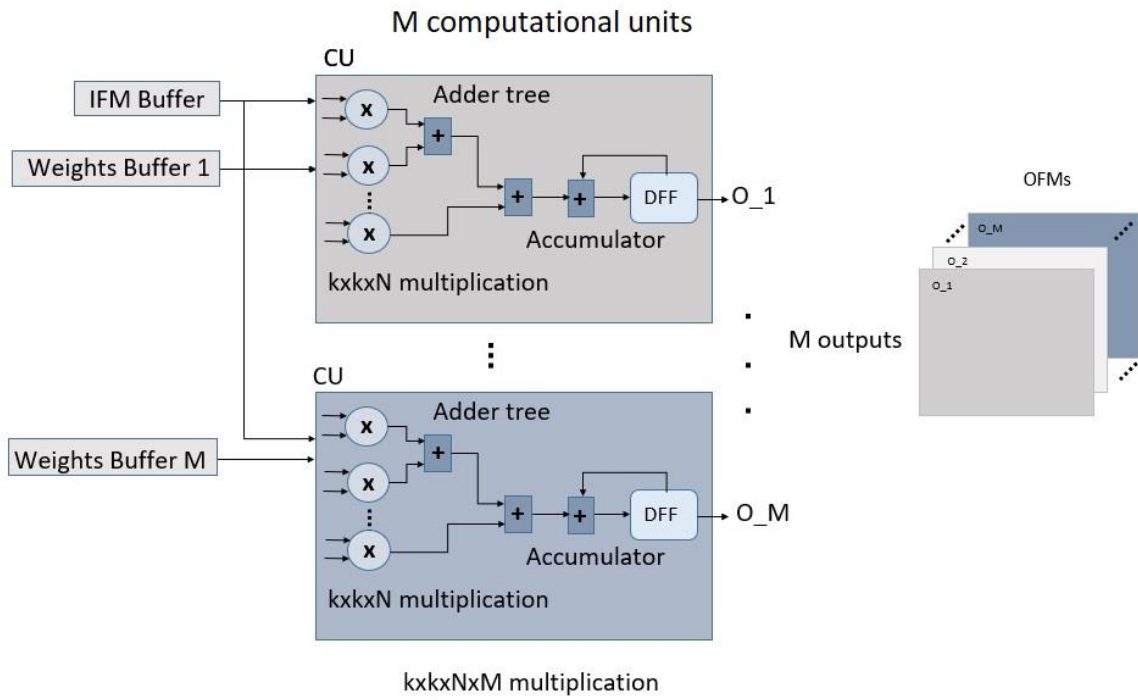
Hesaplama biriminde yapılan işlemlerin kıyaslanması için üç model kullanılmıştır. Kullanılan modeller Çizelge 8.6'da verilmiştir.

Çizelge 8.6. Hesaplama birimlerinde yapılan işlemlerin karşılaştırılması için kullanılan modeller

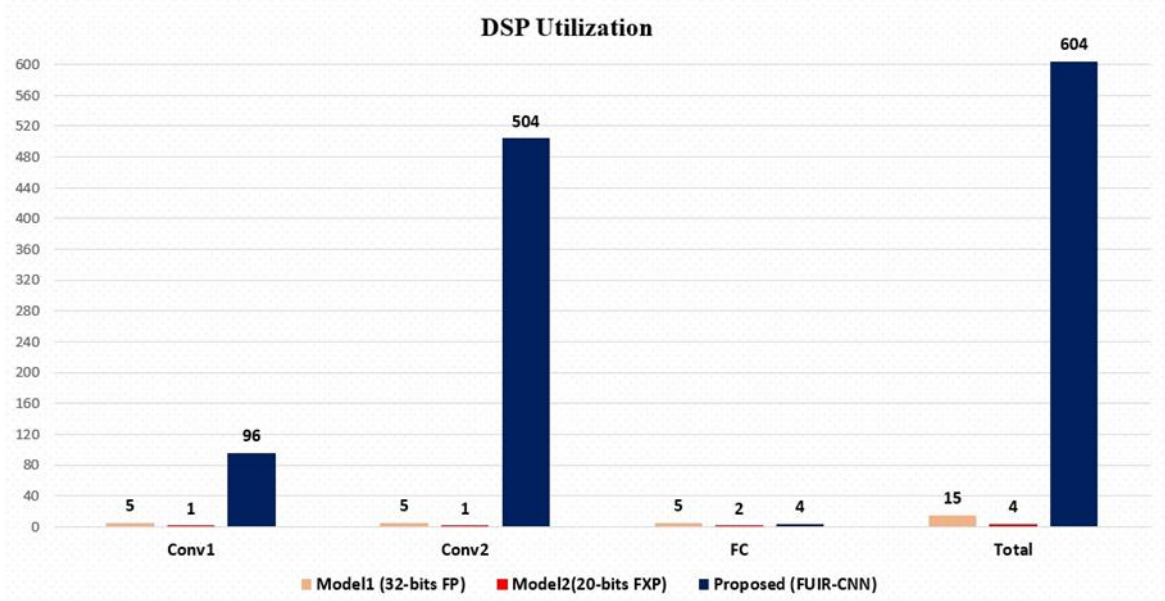
	Veri gösterme yöntemi	Sayı uzunluğu	Veri akışı	Döngü açma	Dizi parçalama
Model1	Kayan noktalı sayı	32 bit	+	-	-
Model2	Sabit noktalı	20 bit	+	-	-
Önerilen yöntem (FUIR-CNN)	Sabit noktalı	20 bit	+	+	+

Modellerde kullanılan eniyileme teknikleri + ile gösterilmiştir. Model1 olarak isimlendirilen model 32-bit FP sayı gösterim tekniği kullanılan modeli göstermektedir. Bu modelde veri akış paralelliği dışında herhangi bir iyileştirme yapılmamıştır. Katmanların içinde yapılan bütün işlemler sıralı olarak yapılmaktadır. Sadece katman çıkışlarında kullanılan FIFO saklayıcılarından dolayı veriler sıralı olarak değil kısmen eş zamanlı olarak sonraki katmanlara depolanmıştır. Model2 olarak isimlendirilen modelde ise 20-bit FXP sayı gösterim tekniği kullanılmıştır. Yine bu modelde de veri akış paralelliği dışında

iyileştirme uygulanmamıştır. Tez çalışmasında önerilen yöntem olan FUIR-CNN modelinde ise sayı gösterim tekniği 20-bit FXP yöntemidir. Bu modelde işlemlerin paralel yapılabilmesi için döngüler açılmış ve hafıza birimleri daha küçük hafızalara ayrılarak veri akışının paralel yapılması sağlanmıştır. Şekil 8.13'te önerilen FUIR-CNN eş zamanlı hesaplama birimi gösterilmiştir. Burada hesaplama birimlerine gelen IFM ler ortak, filtre değerleri ise farklıdır. Aynı anda M tane çıktı değeri üretilmektedir.



Şekil 8.13. Eşzamanlı hesaplama birimi



Şekil 8.14. Karşılaştırma için kullanılan üç modelin katmanlara göre kullandıkları DSP kaynak miktarları

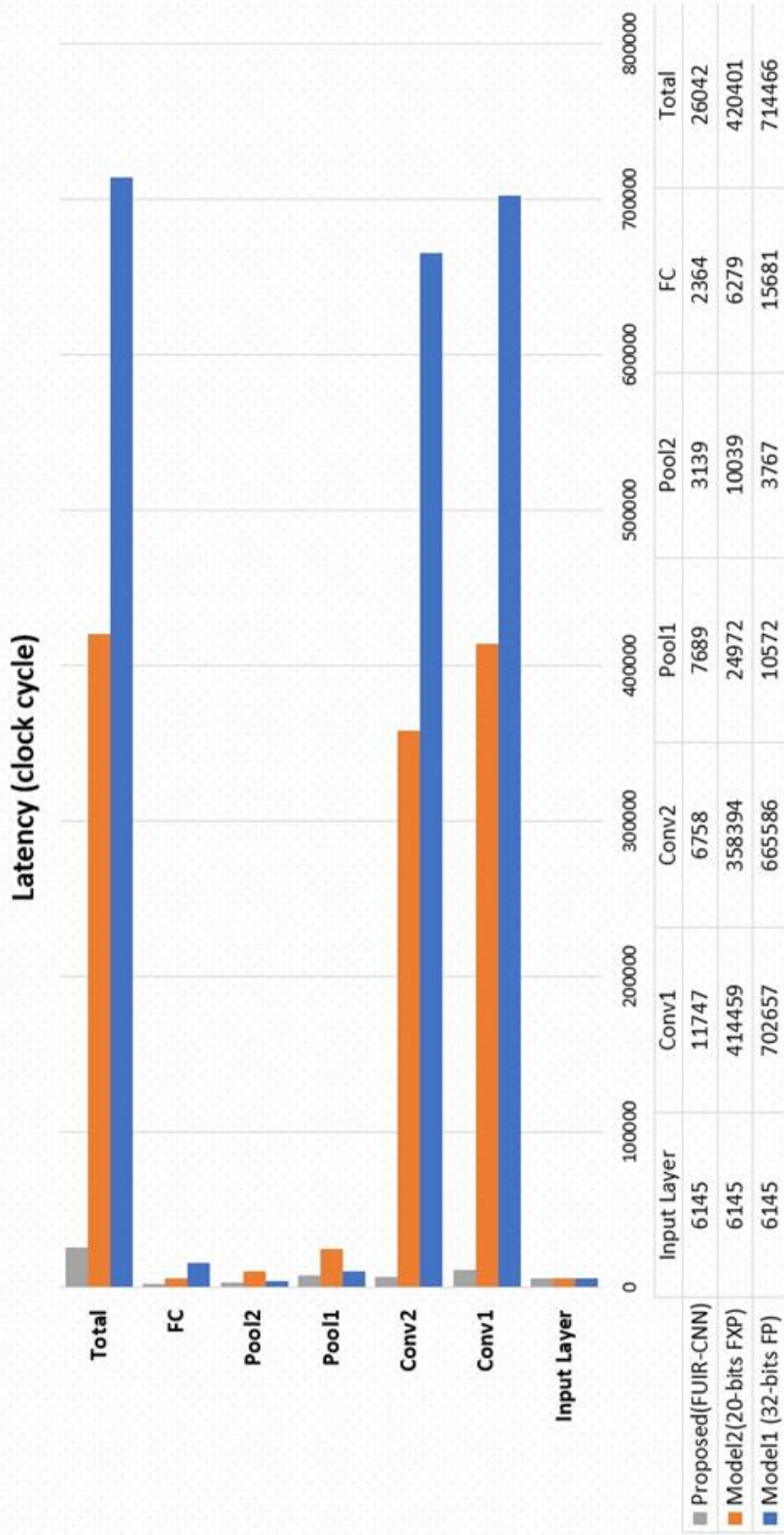
Şekil 8.14'te üç modelin gerçekleşmesi sırasında farklı katmanlarında kullanılan DSP48E kaynak sayılar gösterilmiştir. Birinci modelde evrişim işlemleri sıralı yapılmaktadır. Evrişim işleminde bir çarpma işlemi için 5 tane DSP kaynağı kullanılmaktadır. İşlemler sıralı yapıldığı için DSP'lerde sıralı olarak kullanılmaktadır. Bu nedenle her evrişim katmanında 5 tane tam bağımlı katmanda 5 tane olmak üzere toplamda 15 tane DSP kaynağı kullanılmıştır. Model2 için ise Model1'e benzer şekilde işlemler sıralı yapılmakta ve bir sayısı Model1'e kıyasla daha az olduğu için bir çarpma işlemi için sadece 1 tane DSP kaynağı kullanılmıştır. Toplamda ise 4 tane DSP kaynağı kullanılmıştır. Önerilen model olan FUIR-CNN modelinde ise kullanılan kaynak sayısının arttığı görülmektedir. Çünkü bu modelde işlemler eş zamanlı yapıldığı için kullanılan DSP kaynakları ortak değil, nerdeyse her işleme bir tane DSP kaynağının ayrıldığı görülmektedir. Birinci katman için  $2 \times 2 \times 8 \times 16$  paralel işlem yapılmakta ve işleme bir tane DSP kaynağının ayrıldığı görülmektedir. Toplamda 96 işlem için 96 tane DSP kaynağının kullanıldığı görülmektedir. İkinci evrişim işleminde ise  $2 \times 2 \times 8 \times 16$  işlem için 504 DSP kaynağının ayrıldığı görülmektedir. Toplam kullanılan kaynak sayısının 604 olduğu görülmektedir. Havuzlama katmanında sadece karşılaştırma işlemi yapıldığı için bu katmanlarda DSP kaynakları kullanılmamaktadır.

Çizelge 8.7. Karşılaştırma için kullanılan üç modelin gerçekleşmesinde kullanılan toplam kaynak miktarları ve gecikme miktarları

	BRAM_18K	DSP48E	FF	LUT	Gecikme (cc)
Model1	16	15	32555	10941	714466
Model2	10	4	19677	5494	420401
Proposed	74	604	43301	38735	26042

Çizelge 8.7’de FPGA üzerinde gerçekleştirilen üç modelin kullandığı donanım kaynak sayısı ve gecikme miktarları verilmiştir. Paralel işlem yapan birimlere verilerinde paralel bir şekilde aktarılması için BRAM kaynakları daha küçük BRAM kaynaklarına bölünmüştür. Bu nedenle önerilen modelde kullanılan BRAM kaynak sayısının da arttığı görülmektedir. Gecikme miktarlarına bakıldığı zaman, önerilen FUIR-CNN modelinin diğer iki modele göre daha hızlı sonuç ürettiği görülmektedir. İşlemlerin eşzamanlı yapılması sayesinde ESA regresyona modeli FPGA üzerinde Model1’e göre yaklaşık olarak x27, Model2’ye göre ise yaklaşık olarak x16 hızlandırılmıştır.

Şekil 8.15’te kıyaslanan üç modelin katmanlara göre gecikme miktarları verilmiştir. Giriş katmanında veriler depolandığı için bu katmanlarda modeller arasında bir fark yoktur. Evrişim katmanlarında önerilen modelin işlemlerin eşzamanlı yapılmasından dolayı daha hızlı sonuç ürettiği görülmektedir. Birinci evrişim katmanı için önerilen model Model1’e göre gecikme değerini 702.657 den 11.747 ye düşürmüştür. Bu da işlemlerin yaklaşık olarak 35 kat daha hızlı yapılması anlamına gelmektedir. İkinci evrişim katmanında ise önerilen hızlandırma yönteminin Model1’e göre yaklaşık 98 kat daha hızlı yapıldığı görülmektedir.



Şekil 8.15. Karşılaştırma için kullanılan üç modelin katmanlara göre hesaplanan gecikme miktarları (döngü sayısı olarak)

Çizelge 8.8. Yazılım donanım hız karşılaştırılması

	Yazılım	Model1	Model2	FUIR-CNN
Gecikme	0,018125 s	3,5723 ms	2,102 ms	0,13021 ms
Hız	1x	5x	8,6x	139x
Görüntü/saniye	55	280	476	7580

Çizelge 8.8’de yazılım ve donanım uygulamasında yapılan üç modelin hız parametresine göre karşılaştırılması yapılmıştır. Model1 yazılım tabanlı uygulamaya göre 5 kat daha hızlı sonuç üretmiştir. Bunun nedeni katmanlar arasında FIFO kullanılması dolayısıyla katmanlar arası kısmi paralellik sağlanmasından kaynaklanmaktadır. Önerilen yöntem ise yazılım uygulamasına göre 139 kat daha hızlı sonuç üretmiştir.

Yazılım tabanlı uygulama saniyede 55 görüntü çifti için işlem yapabilecekken, önerilen FPGA tabanlı hızlandırılmış ESA ise 200 MHz çalışma frekansı altında yaklaşık olarak saniyede 7.580 görüntü çifti için işlem yapabilme kapasitesine sahiptir. Yazılım tabanlı uygulama bir görüntü çiftinin arasındaki üç parametreyi 0,018125 saniyede hesaplarken, önerilen FPGA tabanlı yöntem 0,13021 milisaniyede hesaplamıştır.

Çizelge 8.9. Literatürde bulunan bazı çalışmaların kıyaslanması

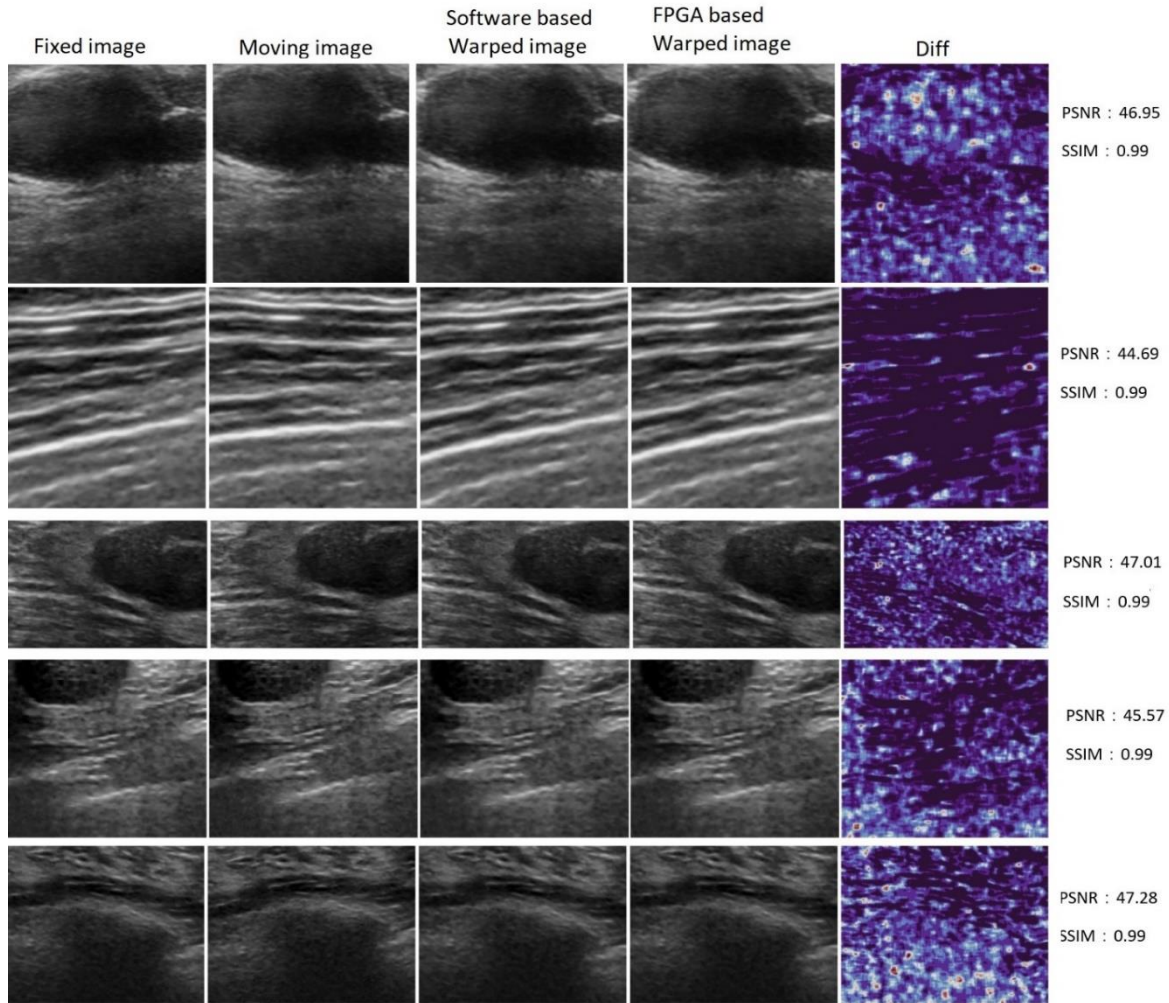
	[68]	[101]	[122]	[123]	[124]	[125]	Bu çalışma
Görüntü boyutu	128x128x128	64x64	512 × 512 × 323	512 × 512 × 150	256x256, 512x512, 400x400	512x512	32x32x3
Yöntem	Denetimli	Denetimsiz	Denetimsiz	Klasik tekrarlı yöntem	Denetimli	Denetimsiz	Denetimli
Dönüşüm modeli	Deforme	Katı	Deforme	Deforme	Katı	Deforme	Katı
Çaıştırılan görüntüler	CT	US	CT/US	CT	CT	CT	US
Çalışma zamanı	0,55 s	0,1 s	0,7 s	58 dk	0,1 s	1,67 s	0,13021 ms

Çizelge 8.9’da literatürde görüntü çaıştırma çalışmalarının ve bu çalışmanın özeti verilmiştir. Çizelge incelendiğinde, klasik tekrarlı yöntemle yapılan çalışmanın, derin öğrenme tabanlı yapılan çalışmalardan oldukça yüksek çalışma zamanına sahip olduğu görülmektedir. DÖ tabanlı çalışmalarda, görüntü boyutu, ağı katman sayısı ve karmaşıklığı gibi hiper parametreler cevap üretme süresini etkilemektedir. Çizelgede gösterilen yöntemler CPU veya GPU tabanlı gerçekledikleri için görüntü boyutları bu çalışmaya göre daha büyüktür. Donanım tabanlı çalışmaların en büyük dezavantajı,

donanımlarda sınırlı kaynak bulunmasıdır. Buda çalışmayı sınırlandırmış ve görüntü boyutunun düşük tutulmasına neden olmuştur. Fakat çizelge incelendiğinde, FPGA tabanlı önerilen bu çalışmanın, mili saniyeler hızında çıkarım aşamasında cevap ürettiği görülmektedir.

Şekil 8.16'da beş tane  $I_F-I_M$  görüntü çiftleri, yazılım tabanlı elde edilmiş parametrelere göre elde edilen  $I_{SW}$  ve donanım tabanlı önerilen optimizasyonlarla elde edilen parametrelere göre oluşturulan  $I_{HW}$ ,  $I_{SW}$  ve  $I_{HW}$  arasındaki farkın görselleştirilmesi verilmiştir. Ayrıca  $I_{SW}$  ve  $I_{HW}$  arasındaki benzerliklerin PSNR ve SSIM benzerlik metriklerine göre hesaplanması verilmiştir.

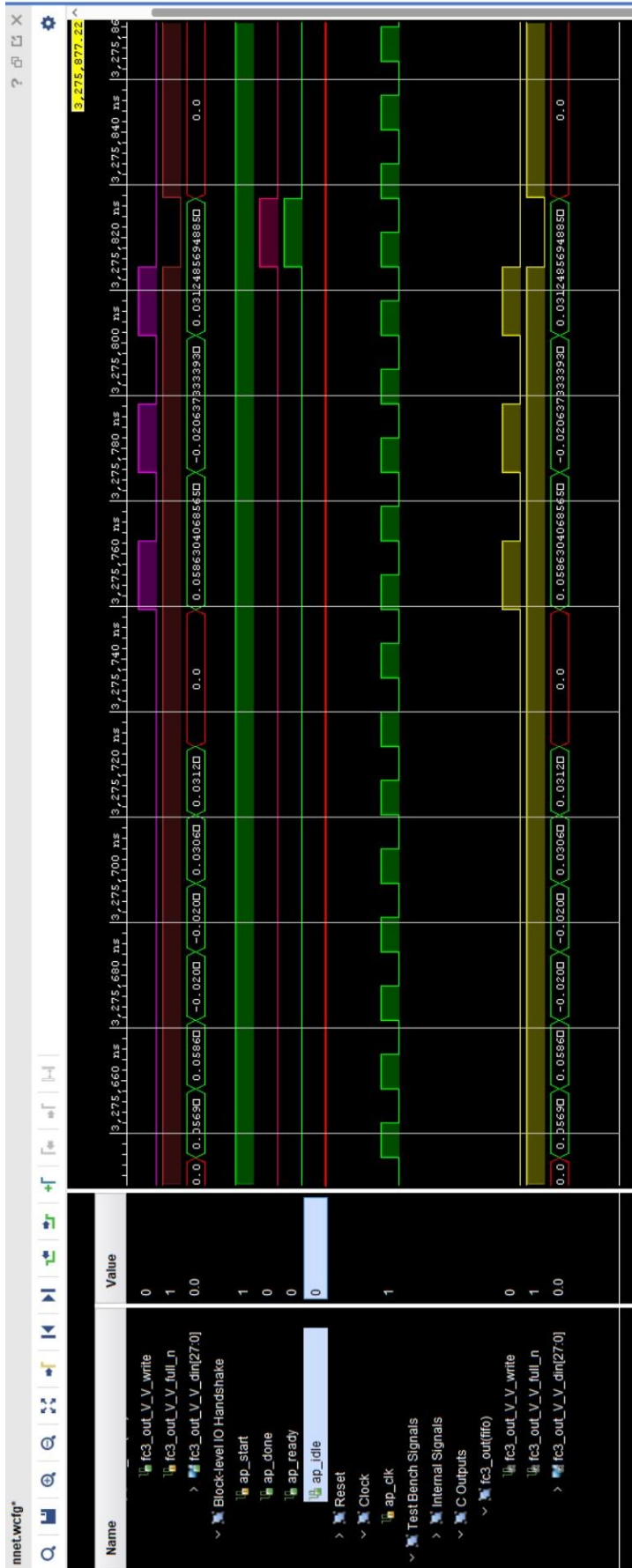
204 US B-mod test görüntü çifti için  $I_{SW}$  ve  $I_{HW}$  arasındaki benzerlik SSIM metriği için 0,99, PSNR metriği için ise 46 olarak hesaplanmıştır.



Şekil 8.16. Yazılım tabanlı ve donanım tabanlı oluşturulan karşılaştırılmış görüntüler

Şekil 8.16’da birinci sütun sabit görüntüleri, ikinci sütun sabit görüntülerin katı dönüşüm uygulanarak elde edilmiş hali olan hareketli görüntüleri, üçüncü sütun yazılım uygulaması sonucu elde edilmiş katı dönüşüm parametrelerin hareketli görüntüye uygulanmasıyla oluşturulan görüntüyü, dördüncü sütun ise hızlandırılmış FPGA tabanlı ESA’nın ürettiği katı dönüşüm parametrelerin hareketli görüntülere uygulanmasıyla elde edilen görüntüleri göstermektedir. Beşinci sütun ise  $I_{SW}$  ve  $I_{HW}$  arasındaki piksel farklarının gösterilmiş halini göstermektedir. PSNR ve SSIM görüntü benzerliğini ölçen değerlendirme metriklerine göre, donanım ve yazılım tabanlı uygulamalardan elde edilen  $I_{SW}$  ve  $I_{HW}$  görüntüler arasındaki benzerliğin çok yüksek çıkmaktadır.

Şekil 8.17’de benzetim sonucu izleme dosyası ekran görüntüsü verilmiştir.



Şekil 8.17. RTL doğrulama izleme dosyası

### 8.3. Farklı Platformların Aynı Uygulama İçin Güç Tüketimlerinin Karşılaştırılması

Güç tüketimi araştırması için kullanılan platformlar:

- Intel(R) Core(TM) i7-10750H 2,60 GHz CPU ve 16,0 GB RAM (dizüstü)
- Intel(R) Core(TM) i5-12400F CPU ve 16,0 GB RAM (Masaüstü)
- Nvidia RTX2080 Ti GPU
- Zedboard FPGA geliştirme kartı
- Kintex® UltraScale+™ FPGA KCU116 geliştirme kartı

Zedboard FPGA geliştirme kartı, masaüstü bilgisayar, dizüstü bilgisayar ve GPU donanımlarının kullandıkları güç miktarının ölçümü UT230B-EU güç ölçüm cihazı kullanılarak yapılmıştır. 0 ~ 3.680W arası güç ölçüm yapabilmekte olan bu cihazın kullanımı güç soketine takılarak yapılmaktadır. Gücü ölçülmek istenen elektrikli cihaz ise sokete takılan güç ölçerin üstünde bulunan sokete takılarak yapılmaktadır. Kintex® UltraScale+™ FPGA KCU116 geliştirme kartı elimizde bulunmadığı için, bu cihaz için yazılım üzerinden tahmini güç ölçümü yapılmıştır. Bunun için Vivado Power Estimation (XPE) aracı kullanılmıştır. Güç ölçümleri sadece çıkarım aşamasında US-US görüntü çakıştırma uygulamasında rot parametre tahmini için yapılmıştır. Şekil 8.18 (a)'da Kintex (baseline tasarım için) (b)'de Kintex optimize edilen tasarım için ve (c)'de Zedboard (baseline tasarım için) XPE tarafından hesaplanan güç değerleri görülmektedir. Kaynak kullanımının artmasının kullanılan güç miktarını da artırdığı görülmektedir. XPE tahmini güç tüketimi statik ve dinamik güç tüketimi olarak iki ayrı şekilde hesaplayarak raporlamaktadır.



Şekil 8.18. VPE (a) Kintex (baseline tasarım için) (b) Kintex optimize edilen tasarım için, (c) Zedboard (baseline tasarım için) XPE tarafından hesaplanan güç değerleri

Şekil 8.19’da Zedboard bağlantıları ve güç ölçüm sonucu gösterilmektedir. Şekil 8.20’de yazılım çıkarım aşamasında dizüstü bilgisayar için yapılan güç ölçümü gösterilmiştir. Şekil 8.21’de ise (a) Masaüstü bilgisayarda bulunan GPU, (b) RTX 2080 ti GPU kullanılarak yapılan çıkarım işlemi sırasında ölçülen güç, ve (c) de ise i5 işlemcili CPU kullanılarak yapılan çıkarım işlemi sırasında ölçülen güç gösterilmektedir. Zedboard için ölçülen güç

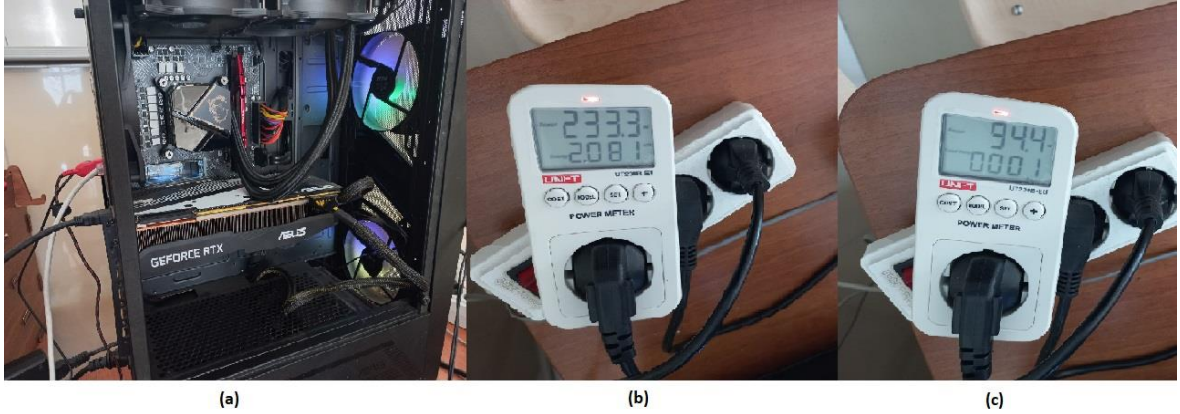
3W iken, i5 işlemcili bilgisayar için 94 W, GPU kullanılırken ölçülen güç ise ortalama 233W olduğu görülmektedir. GPU çıkarım aşamasında ölçülen gücün içinde CPU güç tüketimi de olduğu için GPU platformunun yaklaşık olarak 130 W tükettiği söylenebilir. i7 işlemcili dizüstü bilgisayar kullanılarak yapılan çıkarım aşamasında kullanılan güç ortalama 45 W olarak ölçülmüştür.



Şekil 8.19. ZedBoard FPGA geliştirme kartı-dizüstü bilgisayar bağlantısı ve UT230B-EU güç ölçüm cihazı kullanılarak FPGA için yapılan güç ölçümü



Şekil 8.20. Yazılım çıkarım aşamasında dizüstü bilgisayar için yapılan güç ölçümü



Şekil 8.21. CPU ve GPU güç ölçümleri (a) GPU'nun görünümü, (b) çıkarım aşamasında GPU kullanılarak yapılan güç ölçümü, (c) i5 işlemcili CPU kullanılarak yapılan güç ölçümü

## 9. DEĞERLENDİRME VE SONUÇLAR

Evrişimsel sinir ağlarının görüntü sınıflandırma, nesne tespiti, semantik görüntü bölütleme gibi birçok bilgisayarlı görüntü işleme alanlarında bulunan problemlerde elde ettiği başarıdan dolayı son zamanların en sık tercih edilen algoritmaların başında yer almaktadır. Fakat ESA algoritmaları derin öğrenme algoritmaları olduğu için yüksek hesaplama gücü gerektiren işlemleri içerirler. ESA'larda kullanılan yüksek matematiksel işlemler karmaşık hesaplamalara neden olabilmektedir. Bu da CPU ların sınırlı hesaplama gücünü zorlamakta ve işlemlerin uzamasına neden olabilmektedir. ESA'ların doğası ve genel işlemci birimlerinin mimarisi arasındaki uyumsuzluktan kaynaklanan uygulama zorluğu, bu yapılar için alternatif platformların öne çıkmasına neden olmuştur. Bu nedenle, büyük ve karmaşık ağları eğitmek ve çıkarımlar yapmak için GPU ve FPGA gibi özel amaçlı donanımların kullanılması önemlidir.

Evrişimsel sinir ağları, genellikle GPU ve ASIC gibi yüksek paralel hesaplama yeteneği olan donanımlarda yüksek performansla çalışırlar. Bunun nedeni, ESA'lardaki matris çarpımları, vektör işlemleri ve diğer yoğun hesaplama adımlarının paralel olarak yürütülebilir olmasıdır. GPU'lar, özellikle bu tür matris ve tensör işlemleri için optimize edilmiş mimarilere sahip olduklarından, evrişimsel sinir ağlarının eğitimi ve çıkarımı için yaygın olarak kullanılırlar. Çok sayıda çekirdek ve paralel işlem yapma birimleri sayesinde yüksek derecede eşzamanlı işlem yürütebilirler. Eş zamanlı işlemlerin yürütülmesi ve verilere ulaşılması, ESA uygulamalarının CPU üzerinde gerçeklemeye göre yüksek hıza ulaşmasını sağlayacaktır. Fakat yüksek performans ve eşzamanlı işlemlerin yürütülmesi gibi avantajlarına rağmen GPU uygulamaları için bazı önemli dezavantajlar da bulunmaktadır. Özellikle güç tüketimi konusu, GPU'lar istenen sonuçlardan şimdilik uzak görünmektedir. Yapılan çalışmalar GPU'ların güç tüketiminin yüksek olduğunu, dolayısıyla sinir ağlarının gerçek zamanlı işlenmesinin maliyetli olabileceğini göstermektedir. Ayrıca, güç tüketimine ek olarak, GPU platformları özelleştirilebilirlik açısından şimdilik uzak kalmaktadır. GPU'ların güç tüketimi FPGA'lardan fazla olması çalışmaların yönünü FPGA platformlarına kaydırmıştır.

FPGA'ların tekrar konfigüre edilebilir yapıda olmaları, paralel işlem yapma kabiliyetlerinin olması, güç tüketimlerinin az olması ve özellikle HLS araçlarının C ve C++ gibi dilleri kullanarak programlayabilmesi avantajlarından dolayı ESA mimarilerinin uygulanması için

tercih edilen donanımların başında yer almaktadır. Yeniden programlanabilir donanım olmalarından dolayı yapılan uygulama gereksinimlerine göre tekrar yapılandırılabilirler. Böylece, farklı uygulamalara göre optimize edilebilir ve yeniden yapılandırılabilir. Ayrıca, FPGA belirli ESA modelleri için optimize edilmiş donanımlar olarak kullanılabilir. Bu tür özel donanımlar, evrimsel sinir ağlarının verimli bir şekilde çalışmasını sağlayarak daha hızlı ve güç tasarruflu işlemler gerçekleştirmeye yardımcı olacaktır.

Yapılan bu tez çalışmasında ESA modelinin FPGA üzerinde gerçekleştirilmesi yapılmıştır. Bu tez çalışması iki aşamadan meydana gelmektedir. İlk aşama olarak medikal görüntü çakıştırma problemi için, girdi olarak görüntü çifti alan ve çıktı olarak görüntü çifti arasındaki 3 DOF parametresini tahmin eden ağın eğitimi yapılmış, donanımda gerçekleştirmek üzere ağın parametre ve hiper parametreleri belirlenmiştir. İkinci aşama olarak, belirlenen parametre ve hiper parametreler kullanılarak ESA modelinin FPGA üzerinde gerçekleştirilmesi yapılmıştır. ESA tasarımı için kullanılan katmanlar teker teker oluşturularak doğrulukları test edilmiştir. Katmanların birleştirilmesiyle ESA yapısı oluşturulmuş ve FPGA üzerinde gerçekleştirilmiştir. FPGA üzerinde gerçekleştirilen ESA ağının hızlandırılması, sabit noktalı sayı gösterim şekli, evrişim işlemlerin paralel yapılması ve donanım hafıza birimlerinin parçalanması yöntemleri kullanılarak sağlanmıştır. Katmanları ayrı ayrı tasarlanan ESA modelinde katmanlar arası ve katmanlar içi işlemlerin eş zamanlı yürütülmesi için eniyileme yöntemleri kullanılmıştır. Katmanlar arası işlemlerin kısmi eşzamanlı yapılabilmesi için veri akışı paralelliği, katman içlerinde ise döngü açma, döngü boru hattı ve dizi bölme eniyileme teknikleri kullanılmıştır. Uygulanan eniyileme tekniklerinden sonra FPGA tabanlı hızlandırılmış US-US B-mod görüntü çakıştırma parametrelerini tahmin eden ESA ağının (FUIR-CNN) ihmal edilebilecek bir miktarda hassasiyet kaybıyla beraber, yazılım tabanlı uygulamadan yaklaşık 139 kat daha hızlı sonuç ürettiği aynı zamanda da daha az güç tüketerek daha üstün bir performans sergilediği gösterilmiştir. CPU, GPU ve FPGA platformlarının gerçek zamanlı güç tüketimleri açısından yapılan incelemelerde, aynı modelin çıkarım aşamasında FPGA üzerinde daha az güç tükettiği gözlenmiştir.

Yapılan bu tez çalışması yazılım ve donanım aşaması olmak üzere iki aşamadan oluşması nedeniyle, medikal görüntü çakıştırma problemine hem yazılımsal bir çözüm sunması açısından, US görüntüleme sistemlerinin dezavantajlı olan düşük çözünürlük probleminin iyileştirilmesine imkân sağlayacaktır, hem de donanımsal uyarlamaları teşvik edecek ve

donanım seviyesinde hızlandırma uygulamalarına imkân sağlayacaktır. Görüntü çakıştırma uygulaması medikal alan başta olmak üzere birçok farklı alanda kullanılmaktadır. Bunun yanı sıra ESA yapıları da birçok alanda farklı problemlerin çözümü için kullanılmaktadır. Bu tez çalışmasında ESA yapısını oluşturan katmanlar teker teker tasarlandığı için, bu temel yapıları kullanan farklı uygulamalar için donanım tabanlı gerçekleştirme yapmak mümkündür. Yazılım üzerinde farklı problemler için eğitildikten sonra hiper parametre ve parametreleri belirlenen ağ, donanım kaynaklarını aşmayacak şekilde ESA'yı oluşturan evrişim, havuzlama ve tam bağlantılı katmanlar çoğaltılarak kullanılabilir. Bu da yapılan bu çalışmanın ölçeklenebilirliğini ve genişletilebilirliğini göstermektedir. Tez çalışmasında yapılan ESA yapılarının FPGA'ya uyarlanıp hızlandırılması sadece medikal alanda değil, savunma, endüstri gibi birçok alanda kullanılabilmesi açısından, birden fazla alana kolaylık sağlayacağı öngörülmektedir.



## KAYNAKLAR

1. LeCun, Y., Bengio, Y., and Hinton, G. (2015). Deep learning. *Nature*, 521(7553), 436-444.
2. Alzubaidi, L., Zhang, J., Humaidi, A. J., Al-Dujaili, A., Duan, Y., Al-Shamma, O., and Farhan, L. (2021). Review of deep learning: Concepts, CNN architectures, challenges, applications, future directions. *Journal of big Data*, 8, 1-74.
3. Berman, D. S., Buczak, A. L., Chavis, J. S., and Corbett, C. L. (2019). A survey of deep learning methods for cyber security. *Information*, 10(4), 122.
4. Young, T., Hazarika, D., Poria, S., and Cambria, E. (2018). Recent trends in deep learning based natural language processing. *Computational Intelligence Magazine*, 13(3), 55-75.
5. Min, S., Lee, B., and Yoon, S. (2017). Deep learning in bioinformatics. *Briefings in Bioinformatics*, 18(5), 851-869.
6. Pierson, H. A., and Gashler, M. S. (2017). Deep learning in robotics: a review of recent research. *Advanced Robotics*, 31(16), 821-835.
7. Zemouri, R., Zerhouni, N., and Racoceanu, D. (2019). Deep learning in the biomedical applications: Recent and future status. *Applied Sciences*, 9(8), 1526.
8. Li, P., Pei, Y., and Li, J. (2023). A comprehensive survey on design and application of autoencoder in deep learning. *Applied Soft Computing*, 1, 110176.
9. Hinton, G. E. (2012). *A practical guide to training restricted Boltzmann machines*. In *Neural Networks: Tricks of the Trade: Second Edition*. Berlin, Heidelberg: Springer Berlin Heidelberg, 599-619.
10. Dhillon, A., and Verma, G. K. (2020). Convolutional neural network: a review of models, methodologies and applications to object detection. *Progress in Artificial Intelligence*, 9(2), 85-112.
11. İnik, Ö., ve Ülker, E. (2017). Derin öğrenme ve görüntü analizinde kullanılan derin öğrenme modelleri. *Gaziosmanpaşa Bilimsel Araştırma Dergisi*, 6(3), 85-104.
12. Tan, F. G., Yüksel, A. S., Aydemir, E., ve Ersoy, M. (2021). Derin öğrenme teknikleri ile nesne tespiti ve takibi üzerine bir inceleme. *Avrupa Bilim ve Teknoloji Dergisi*, (25), 159-171.
13. Şişmanoğlu, G., Koçer, F., Önde, M. A., ve Sahingoz, O. K. (2020). Derin öğrenme yöntemleri ile borsada fiyat tahmini. *Bitlis Eren Üniversitesi Fen Bilimleri Dergisi*, 9(1), 434-445.
14. Kaya, B., ve Önal, M. (2021). COVID-19 Tespiti için Akciğer BT Görüntülerinin Bölütlenmesi. *Avrupa Bilim ve Teknoloji Dergisi*, (28), 1296-1303.

15. Karayeğen, G., and Akşahin, M. F. (2021, November). *Brain Tumor Prediction with Deep Learning and Tumor Volume Calculation*. In 2021 Medical Technologies Congress (TIPTEKNO), 1-4.
16. Cao, X., Yang, J., Zhang, J., Nie, D., Kim, M., Wang, Q., and Shen, D. (2017). *Deformable image registration based on similarity-steered CNN regression*. In Medical Image Computing and Computer Assisted Intervention– MICCAI 2017: 20th International Conference, Quebec City, QC, Canada, Proceedings, 303-308.
17. Dayan, A., ve Yılmaz, A. (2022). Doğal dil işleme ve derin öğrenme algoritmaları ile makine dili modellemesi. *Dicle Üniversitesi Mühendislik Fakültesi Mühendislik Dergisi*, 13(3), 467-475.
18. LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11), 2278-2324.
19. Huang, G., Liu, Z., Van Der Maaten, L., and Weinberger, K. Q. (2017). *Densely connected convolutional networks*. In Proceedings of the IEEE conference on computer vision and pattern recognition, 4700-4708.
20. Ronneberger, O., Fischer, P., and Brox, T. (2015). *U-net: Convolutional networks for biomedical image segmentation*. In Medical Image Computing and Computer-Assisted Intervention–MICCAI 18th International Conference, Munich, Germany, 234-241. Springer International Publishing.
21. Abdelouahab, K., Pelcat, M., Serot, J., and Berry, F. (2018). Accelerating CNN inference on FPGAs: A survey. *arXiv preprint arXiv:1806.01683*.
22. Nurvitadhi, E., Venkatesh, G., Sim, J., Marr, D., Huang, R., Ong Gee Hock, J., ... and Boudoukh, G. (2017, February). *Can FPGAs beat GPUs in accelerating next-generation deep neural networks?*. In Proceedings of the 2017 ACM/SIGDA international symposium on field-programmable gate arrays, 5-14.
23. Qasaimeh, M., Denolf, K., Lo, J., Vissers, K., Zambreno, J., and Jones, P. H. (2019, June). Comparing energy efficiency of CPU, GPU and FPGA implementations for vision kernels. In *2019 IEEE international conference on embedded software and systems (ICCESS)*, 1-8.
24. Sze, V., Chen, Y. H., Yang, T. J., and Emer, J. S. (2017). Efficient processing of deep neural networks: A tutorial and survey. *Proceedings of the IEEE*, 105(12), 2295-2329.
25. Horowitz, M. (2014, February). *1.1 computing's energy problem (and what we can do about it)*. In 2014 IEEE International Solid-State Circuits Conference Digest of Technical Papers (ISSCC), 10-14.
26. Nguyen, D. T., Nguyen, T. N., Kim, H., and Lee, H. J. (2019). A high-throughput and power-efficient FPGA implementation of YOLO CNN for object detection. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 27(8), 1861-1873.
27. Magdy Saady, M., and Hassan Essai, M. (2022). Hardware implementation of neural network-based engine model using FPGA. *Alexandria Engineering Journal*, 61(12), 12039–12050.

28. Inam, O., Basit, A., Qureshi, M., and Omer, H. (2020). FPGA-based hardware accelerator for SENSE (a parallel MR image reconstruction method). *Computers in Biology and Medicine*, 117, 103598.
29. Mani, V. R. S., Saravanaselvan, A., and Arumugam, N. (2022). Performance comparison of CNN, QNN and BNN deep neural networks for real-time object detection using ZYNQ FPGA node. *Microelectronics Journal*, 119, 105319.
30. Baba, A., and Bonny, T. (2023). FPGA-based parallel implementation to classify Hyperspectral images by using a Convolutional Neural Network. *Integration*, 92, 15-23.
31. Rawal, V., Prajapati, P., and Darji, A. (2023). Hardware implementation of 1D-CNN architecture for ECG arrhythmia classification. *Biomedical Signal Processing and Control*, 85, 104865.104865–104865.
32. Shawahna, A., Sait, S. M., and El-Maleh, A. (2018). FPGA-based accelerators of deep learning networks for learning and classification: A review. *Access*, 7, 7823-7859.
33. Sait, S. M., El-Maleh, A., Altakrouri, M., and Shawahna, A. (2022). Optimization of FPGA-based CNN accelerators using metaheuristics. *The Journal of Supercomputing*. 79(4), 4493-4533.
34. Messaoud, S., Bouaafia, S., Maraoui, A., Ammari, A. C., Khriji, L., and Machhout, M. (2022). Deep convolutional neural networks-based Hardware–Software on-chip system for computer vision application. *Computers and Electrical Engineering*, 98, 107671.
35. Khan, F. H., Pasha, M. A., and Masud, S. (2023). Towards designing a hardware accelerator for 3D convolutional neural networks. *Computers and Electrical Engineering*, 105, 108489.
36. İnternet: Gina R. Smith. FPGAs 101. <https://doi.org/10.1016/c2009-0-20154-1>, Son Erişim Tarihi: 26.06.2023.
37. Çavuşlu, M. A., and Kösten, M. M. (2015). VHDL ile Sayısal Tasarım ve FPGA Uygulamaları. İstanbul: Kodlab Yayınları.
38. Liu, S., Zhai, D., and Han, B. (2021). FPGA medical big data system and ischemic stroke rehabilitation nursing. *Microprocessors and Microsystems*, 83, 104014.
39. Jin, Y., Wan, Q., Wu, X., Fu, X., and Chen, J. (2023). FPGA-accelerated deep neural network for real-time inversion of geosteering data. *Geoenergy Science and Engineering*, 224, 211610.
40. Monmasson, E., Id-khajine, L., Merai, M., Naouar, M. W., and Ricco, M. (2022). *FPGA-based control for power electronics applications*. New York: Elsevier, 577–589.
41. Luo, Y., Cai, X., Qi, J., Guo, D., and Che, W. (2023). FPGA–accelerated CNN for real-time plant disease identification. *Computers and Electronics in Agriculture*, 207, 107715.

42. Jing-yu, S., Hong, C., Gang, W., Zi-bo, G., and Zhang, H. (2023). FPGA image encryption-steganography using a novel chaotic system with line equilibria. *Digital Signal Processing*, 134, 103889.
43. Elkholy, M. H., Elymany, M., Yona, A., Senjyu, T., Takahashi, H., and Lotfy, M. E. (2023). Experimental validation of an AI-embedded FPGA-based Real-Time smart energy management system using Multi-Objective Reptile search algorithm and gorilla troops optimizer. *Energy Conversion and Management*, 282, 116860.
44. Gadea-Gironés, R., Fe, J., and Monzo, J. M. (2023). Task parallelism-based architectures on FPGA to optimize the energy efficiency of AI at the edge. *Microprocessors and Microsystems*, 98, 104824.
45. Tang, C., Guss, D., Tanaka, M. J., and Lubberts, B. (2022). Portable ultrasound devices: A method to improve access to medical imaging, barriers to implementation, and the need for future advancements. *Clinical imaging*, 81, 147–149.
46. Buyukfirat, E., Binici, O., Duran, E., Pehlivan, B., Atlas, A., Erol, M. K., and Babacan, S. (2023). Can different positions facilitate block application in ultrasound-guided obturator nerve block? A prospective comparative study. *Electronic Journal of General Medicine*, 20(1), em426.
47. Vuorenmaa, A., Siitama, E., Hakulinen, U., and Eskola, H. (2023). Technical Performance Assessment and Quality Control of Ultrasound Device Monitors. *Ultrasound in Medicine and Biology*, 49(1), 380–387.
48. Chen, Y., Xing, L., Yu, L., Liu, W., Pooya Fahimian, B., Niedermayr, T., Bagshaw, H. P., Buyyounouski, M., and Han, B. (2021). MR to ultrasound image registration with segmentation-based learning for HDR prostate brachytherapy. *Medical physics*, 48(6), 3074–3083.
49. Yang, M., Ding, H., Zhu, L., and Wang, G. (2016). Ultrasound fusion image error correction using subject-specific liver motion model and automatic image registration. *Computers in Biology and Medicine*, 79, 99–109.
50. Li, Y., Wang, J., and Yao, K. (2022). Modified phase correlation algorithm for image registration based on pyramid. *Alexandria Engineering Journal*, 61(1), 709–718.
51. Liu, B., Bai, X., Zhou, F., Han, H., and Hou, C. (2013). Mutual information based three-dimensional registration of rat brain magnetic resonance imaging time-series. *Computers and Electrical Engineering*, 39(5), 1473–1484.
52. Aydin, S. G., Bilge, H. Ş., and Hardalac, F. (2022). *Supervised rigid image registration with CNN for MR brain images*. 2022 Innovations in Intelligent Systems and Applications Conference (ASYU), Antalya, 1-5.
53. Matsopoulos, G. K., Mouravliansky, N. A., Delibasis, K. K., and Nikita, K. S. (1999). Automatic retinal image registration scheme using global optimization techniques. *Transactions on Information Technology in Biomedicine*, 3(1), 47-60.
54. Wang, Y., Li, Y., Chen, Y., Peng, M., Li, H., Yang, B., and Dong, Z. (2022). Automatic registration of point cloud and panoramic images in urban scenes based on

- pole matching. *International Journal of Applied Earth Observation and Geoinformation*, 115, 103083.
55. Sommervold, O., Gazzea, M., and Arghandeh, R. (2023). A survey on sar and optical satellite image registration. *Remote Sensing*, 15(3), 850.
  56. Sen, M., Hemaraj, Y., Plishker, W., Shekhar, R., and Bhattacharyya, S. S. (2008). Model-based mapping of reconfigurable image registration on FPGA platforms. *Journal of Real-Time Image Processing*, 3, 149-162.
  57. Misra, I., Rohil, M. K., Manthira Moorthi, S., and Dhar, D. (2022). Feature based remote sensing image registration techniques: a comprehensive and comparative review. *International Journal of Remote Sensing*, 43(12), 4477-4516.
  58. Bentoutou, Y., Taleb, N., Kpalma, K., and Ronsin, J. (2005). An automatic image registration for applications in remote sensing. *Transactions on Geoscience and Remote Sensing*, 43(9), 2127-2137.
  59. Ying, Y., Ying, P., Men, H., and Joo, Y. H. (2023). Image registration based fault localization in panoramas of mountain-mounted PV plants. *Solar Energy*, 256, 16-31.
  60. Maslov, I. V. (2004, August). *Automatic image registration and target recognition with multi-resolution hybrid evolutionary algorithm*. In *Signal Processing, Sensor Fusion, and Target Recognition*, 180-187.
  61. Li, H., Fan, X., Jiao, L., Cao, W., Zhou, X., and Wang, L. (2016, August). *A high performance FPGA-based accelerator for large-scale convolutional neural networks*. In *2016 26th International Conference on Field Programmable Logic and Applications (FPL)*, 1-9.
  62. Cardoso, J. M. P., de Figueiredo Coutinho, J. G., and Diniz, P. C. (2017). *Embedded computing for high performance: Efficient mapping of computations using customization, code transformations and compilation*. Morgan Kaufmann. 255–280.
  63. Zhang, C., Li, P., Sun, G., Guan, Y., Xiao, B., and Cong, J. (2015, February). *Optimizing FPGA-based accelerator design for deep convolutional neural networks*. In *Proceedings of the 2015 ACM/SIGDA international symposium on field-programmable gate arrays*, 161-170.
  64. Ayat, S. O., Khalil-Hani, M., and Rahman, A. (2018). Optimizing fpga-based cnn accelerator for energy efficiency with an extended roofline model. *Turkish Journal of Electrical Engineering and Computer Sciences*, 26(2), 919-935.
  65. Ma, Y., Cao, Y., Vrudhula, S., and Seo, J. S. (2018). Optimizing the convolution operation to accelerate deep neural networks on FPGA. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 26(7), 1354-1367.
  66. Khan, F. H., Pasha, M. A., and Masud, S. (2023). Towards designing a hardware accelerator for 3D convolutional neural networks. *Computers and Electrical Engineering*, 105, 108489.

67. Sloan, J. M., Goatman, K. A., and Siebert, J. P. (2018). *Learning rigid image registration-utilizing convolutional neural networks for medical image registration*. In 11th International Joint Conference on Biomedical Engineering Systems and Technologies, Funchal, Madeira, Portugal.
68. Eppenhof, K. A., Lafarge, M. W., Veta, M., and Pluim, J. P. (2019). Progressively trained convolutional neural networks for deformable image registration. *IEEE Transactions on Medical Imaging*, 39(5), 1594-1604.
69. de Vos, B. D., Berendsen, F. F., Viergever, M. A., Sokooti, H., Staring, M., and Išgum, I. (2019). A deep learning framework for unsupervised affine and deformable image registration. *Medical Image Analysis*, 52, 128–143.
70. Wei, W., Haishan, X., Alpers, J., Rak, M., and Hansen, C. (2021). A deep learning approach for 2D ultrasound and 3D CT/MR image registration in liver tumor ablation. *Computer Methods and Programs in Biomedicine*, 206, 106117.
71. Stratakos, I., Gourounas, D., Tsoutsouras, V., Economopoulos, T., Matsopoulos, G., and Soudris, D. (2019, May). *Hardware acceleration of image registration algorithm on fpga-based systems on chip*. In Proceedings of the International Conference on omni-layer intelligent systems, 92-97.
72. White, B. A. (2009). *Using FPGAs to perform embedded image registration*. BSc. Major Thesis, Computer Engineering, University of Central Florida, Florida.
73. Chen, J., Alfred, C. H., and So, H. K. H. (2012, December). *Design considerations of real-time adaptive beamformer for medical ultrasound research using FPGA and GPU*. In 2012 International Conference on Field-Programmable Technology, 198-205.
74. Dandekar, O., & Shekhar, R. (2007). FPGA-accelerated deformable image registration for improved target-delineation during CT-guided interventions. *IEEE Transactions on Biomedical Circuits and Systems*, 1(2), 116-127.
75. Siddiqui, F., Amiri, S., Minhas, U., Deng, T., Woods, R., Rafferty, K., and Crookes, D. (2019). FPGA-based processor acceleration for image processing applications. *Journal of Imaging*, 5(1), 16.
76. Goodfellow, I., Bengio, Y., and Courville, A. (2016). *Deep learning* (Press 5). New York: MIT Press, 5.
77. Dumoulin, V., and Visin, F. (2016). A guide to convolution arithmetic for deep learning. *arXiv preprint arXiv:1603.07285*.
78. Kurt, F. (2018). *Evrışimli Sinir Ağlarında Hiper Parametrelerin Etkisinin İncelenmesi*, Yüksek Lisans Tezi, Hacettepe Üniversitesi, Fen Bilimleri Enstitüsü, Ankara.
79. Chollet, F. (2021). *Python ile derin öğrenme* (İkinci Baskı). Türkiye: Buzdağı Yayınevi.

80. Aldrich, J. E. (2007). Basic physics of ultrasound imaging. *Critical Care Medicine*, 35(5), S131-S137.
81. Wells, P. N. (2006). Ultrasound imaging. *Physics in Medicine and Biology*, 51(13), R83.
82. Temiz, H. (2020). *Ultrason Görüntülerinde Derin Öğrenme Yaklaşımıyla Süper Çözünürlük*, Doktora Tezi, Gazi Üniversitesi, Bilişim Enstitüsü, Ankara.
83. Chan, V., and Perlas, A. (2011). *Basics of ultrasound imaging. Atlas of ultrasound-guided procedures in interventional pain management*. New York: Springer, 13-19.
84. Ng, A., and Swanevelder, J. (2011). Resolution in ultrasound imaging. *Continuing Education in Anaesthesia Critical Care and Pain*, 11(5), 186-192.
85. Suetens, P. (2017). *Fundamentals of medical imaging*. Cambridge: Cambridge University Press.
86. Szasz, T. (2016). *Advanced beamforming techniques in ultrasound imaging and the associated inverse problems*. Doctoral dissertation, Institut de Recherche en Informatique de Toulouse, Medical Imaging Université Toulouse, France.
87. Bilge, H. Ş. (2003). *Delta-Sigma Örneklemeli Altdizilim İşlemeye Dayalı Bir Demetleme Yöntemi*, Doktora Tezi, Başkent Üniversitesi Fen Bilimleri Enstitüsü, Ankara, 5.
88. Internet: Transducers, <https://radiologykey.com/transducers-2/> Son Erişim Tarihi: 20.06.2023.
89. Hirotaki, K., Moriya, S., Akita, T., Yokoyama, K., and Sakae, T. (2022). Image preprocessing to improve the accuracy and robustness of mutual-information-based automatic image registration in proton therapy. *Physica Medica*, 101, 95–103.
90. Derek, Batchelor, P., Holden, M. A., and Hawkes, D. J. (2001). Medical image registration. *Physics in Medicine & Biology*, 46(3), R1–R45.
91. Laganà, M. M., Preti, M. G., Forzoni, L., D'Onofrio, S., De Beni, S., Barberio, A., and Baselli, G. (2013). Transcranial ultrasound and magnetic resonance image fusion with virtual navigator. *Transactions on Multimedia*, 15(5), 1039-1048.
92. Huang, Q., and Zeng, Z. (2017). A review on real-time 3D ultrasound imaging technology. *BioMed research international*, 2017. <https://pubmed.ncbi.nlm.nih.gov/28459067/> Son Erişim Tarihi: 20.06.2023.
93. Mohamed, F., and Siang, C. V. (2019). A survey on 3D ultrasound reconstruction techniques. *Artificial Intelligence—Applications in Medicine and Biology*, 1, 73-92.
94. Lai, K. W., and Dewi, D. E. O. (Eds.). (2015). *Medical imaging technology: reviews and computational applications*. New York: Springer.
95. Goshtasby, A. A. (2012). *Image registration: Principles, tools and methods*. Springer: Science and Business Media.

96. Goshtasby, A. A. (2017). *Theory and applications of image registration*. London: John Wiley and Sons.
97. He, K., Zhang, X., Ren, S., and Sun, J. (2016). *Deep residual learning for image recognition*. In Proceedings of the IEEE conference on computer vision and pattern recognition, 770-778. Las Vegas, NV, ABD.
98. İnternet: Approaches to Registering Images - MATLAB and Simulink. (2023). from Mathworks.com URL: <https://www.mathworks.com/help/images/approaches-to-registering-images.html>, Son Erişim Tarihi: 20.06.2023.
99. Uchida, S. (2013). Image processing and recognition for biological images. *Development, Growth & Differentiation*, 55(4), 523-549.
100. Nag, S. (2017). Image registration techniques: a survey. *arXiv preprint arXiv:1712.07540*.
101. Liu, H., Chi, Y., Mao, J., Wu, X., Liu, Z., Xu, Y., and Huang, W. (2021, November). *End to end unsupervised rigid medical image registration by using convolutional neural networks*. In 2021 43rd Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC), 4064-4067.
102. Haskins, G., Kruger, U., and Yan, P. (2020). Deep learning in medical image registration: a survey. *Machine Vision and Applications*, 31, 1-18.
103. Al-Dhabyani, W., Gomaa, M., Khaled, H., and Fahmy, A. (2020). Dataset of breast ultrasound images. *Data in Brief*, 28, 104863.
104. Yap, M.H., Pons, G., Marti, J., Ganau, S., Sentis, M., Zwigelaar, R., Davison, A.K. and Marti, R.(2017), Automated Breast Ultrasound Lesions Detection using Convolutional Neural Networks. *IEEE journal of biomedical and health informatics*, 22(4), 1218-1226.
105. İnternet: Ultrasound image database | SPLab. (2012). Splab.cz. URL: <http://splab.cz/en/download/databaze/ultrasound>, Son Erişim Tarihi: 20.06.2023.
106. Nasri Sulaiman, Z. A. O., Marhaban, M. H., and Hamidon, M. N. (2009). Design and implementation of FPGA-based systems-a review. *Australian Journal of Basic and Applied Sciences*, 3(4), 3575-3596.
107. Yang, H., Zhang, J., Sun, J., and Yu, L. (2014). Review of advanced FPGA architectures and technologies. *Journal of Electronics (China)*, 31(5), 371-393.
108. İnternet: Guo, K., Li, W., Zhong, K., Zhu, Z., Zeng, S., Han, S., and Wang, Y. (2020). Neural network accelerator comparison. *NICS Lab of Tsinghua University*. URL: <https://nicsefc.ee.tsinghua.edu.cn/projects/neural-network-accelerator/> Son Erişim Tarihi: 20.06.2023.
109. Aydin, S. G., and Bilge, H. S. (2020). *FPGA based implementation of sigmoid function using different approaches*. 2020 28th Signal Processing and Communications Applications Conference (SIU), Gaziantep, 1-4.

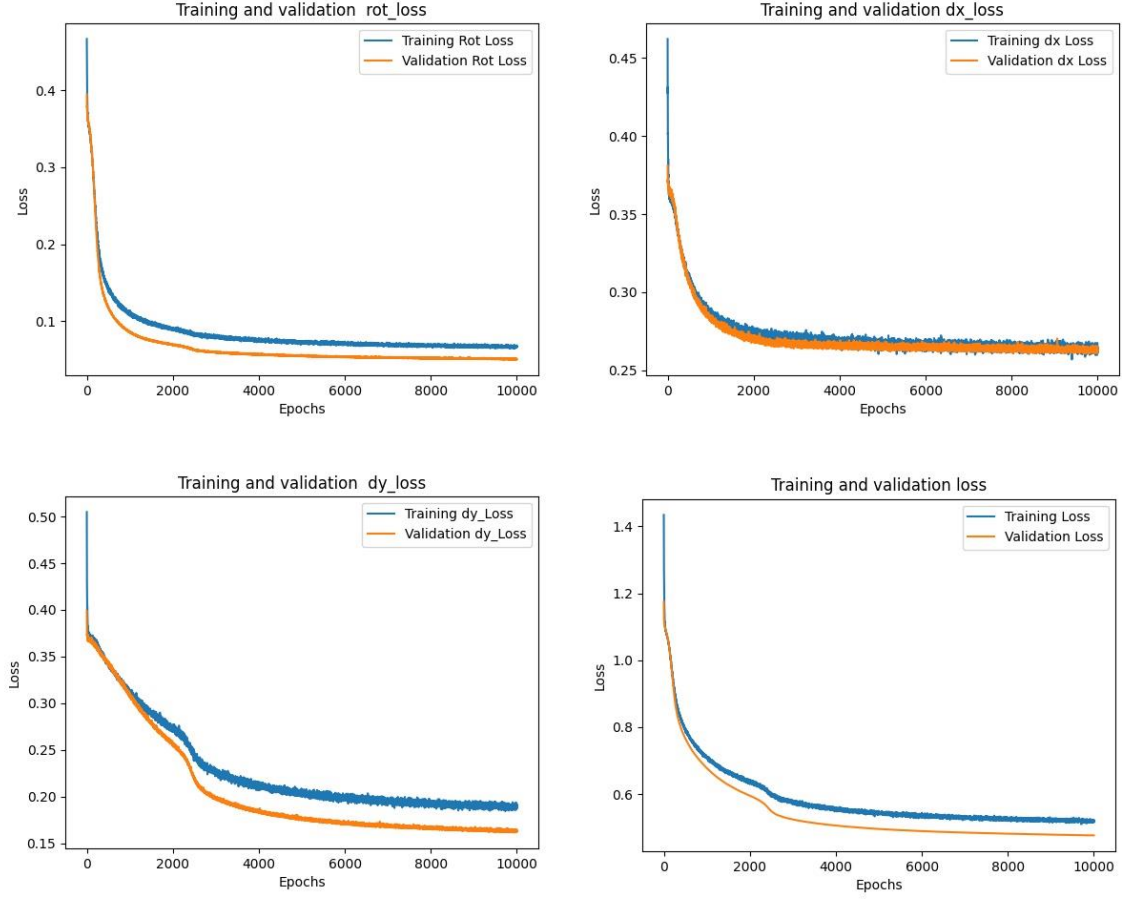
110. Çavuşlu, M. A., Tuğaç, S., and Öner, M. (2021). FPGA implementation of bearing tracking using passive array for underwater acoustics. *Microprocessors and Microsystems*, 87, 104366.
111. Internet: Neuendorffer, S., Li, T., and Wang, D. (2013). *Accelerating opencv applications with zynq-7000 all programmable soc using vivado hls video libraries*. Xilinx Inc., 1-14. URL: <https://xilinx.eetrend.com/files-eetrend-xilinx/download/201401/6326-10718-xapp1167.pdf>, Son Erişim Tarihi: 20.06.2023.
112. İnternet: AMD Adaptive Computing Documentation Portal. (2023). Xilinx.com URL: <https://docs.xilinx.com/v/u/en-US/ug902-vivado-high-level-synthesis>, Son Erişim Tarihi: 20.06.2023.
113. Peter W.F., and Wilson. (2015). *Fixed Point Arithmetic. Design Recipes for FPGAs*. USE: TOOD Green. doi: 10.1016/b978-0-08-097129-2.00023-4
114. Peter W.F., and Wilson. (2007). *Fixed Point Arithmetic in VHDL. Design Recipes for FPGAs*. USE: TOOD Green.
115. DiCecco, R., Sun, L., and Chow, P. (2017, December). *FPGA-based training of convolutional neural networks with a reduced precision floating-point library*. In 2017 International Conference on Field Programmable Technology (ICFPT), 239-242. Melbourne, VIC, Avustralya.
116. Chetan, S., Sourabh, K. S., Lekshmi, V., Sudhakar, S., and Manikandan, J. (2020). Design and Evaluation of Floating point Matrix Operations for FPGA based system design. *Procedia Computer Science*, 171, 959-968.
117. Patel, S. K., and Singhal, S. K. (2023). An area-delay efficient single-precision floating-point multiplier for VLSI systems. *Microprocessors and Microsystems*, 98, 104798–104798.
118. Arredondo-Velázquez, M., Diaz-Carmona, J., Torres-Huitzil, C., Padilla-Medina, A., and Prado-Olivarez, J. (2020). A streaming architecture for Convolutional Neural Networks based on layer operations chaining. *Journal of Real-Time Image Processing*, 17(5), 1715–1733.
119. Zhang, X. (2017). *A design methodology for efficient implementation of deconvolutional neural networks on an FPGA*. University of California, San Diego.
120. Liu, S., Fan, H., Niu, X., Ng, H. C., Chu, Y., and Luk, W. (2018). Optimizing CNN-based segmentation with deeply customized convolutional and deconvolutional architectures on FPGA. *ACM Transactions on Reconfigurable Technology and Systems (TRETS)*, 11(3), 1-22.
121. Sikka, P. (2023). High-level synthesis assisted, low-latency, area- and power-optimized FPGA implementation of MUSIC algorithm for direction-of-arrival estimation. *Sustainable Energy Technologies and Assessments*, 57, 103201–103201.
122. Dahman, B., and Dillenseger, J. L. (2020, Temmuz). *Deformable US/CT Image Registration with a Convolutional Neural Network for Cardiac Arrhythmia Therapy*.

In 2020 42nd Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC), 2011-2014.

123. Delmon, V., Rit, S., Pinho, R., and Sarrut, D. (2013). Registration of sliding objects using direction dependent B-splines decomposition. *Physics in Medicine & Biology*, 58(5), 1303.
124. Miao, S., Wang, Z. J., and Liao, R. (2016). A CNN regression approach for real-time 2D/3D registration. *Transactions on Medical Imaging*, 35(5), 1352-1363.
125. Ho, T. T., Kim, W. J., Lee, C. H., Jin, G. Y., Chae, K. J., and Choi, S. (2023). An unsupervised image registration method employing chest computed tomography images and deep neural networks. *Computers in Biology and Medicine*, 154, 106612.

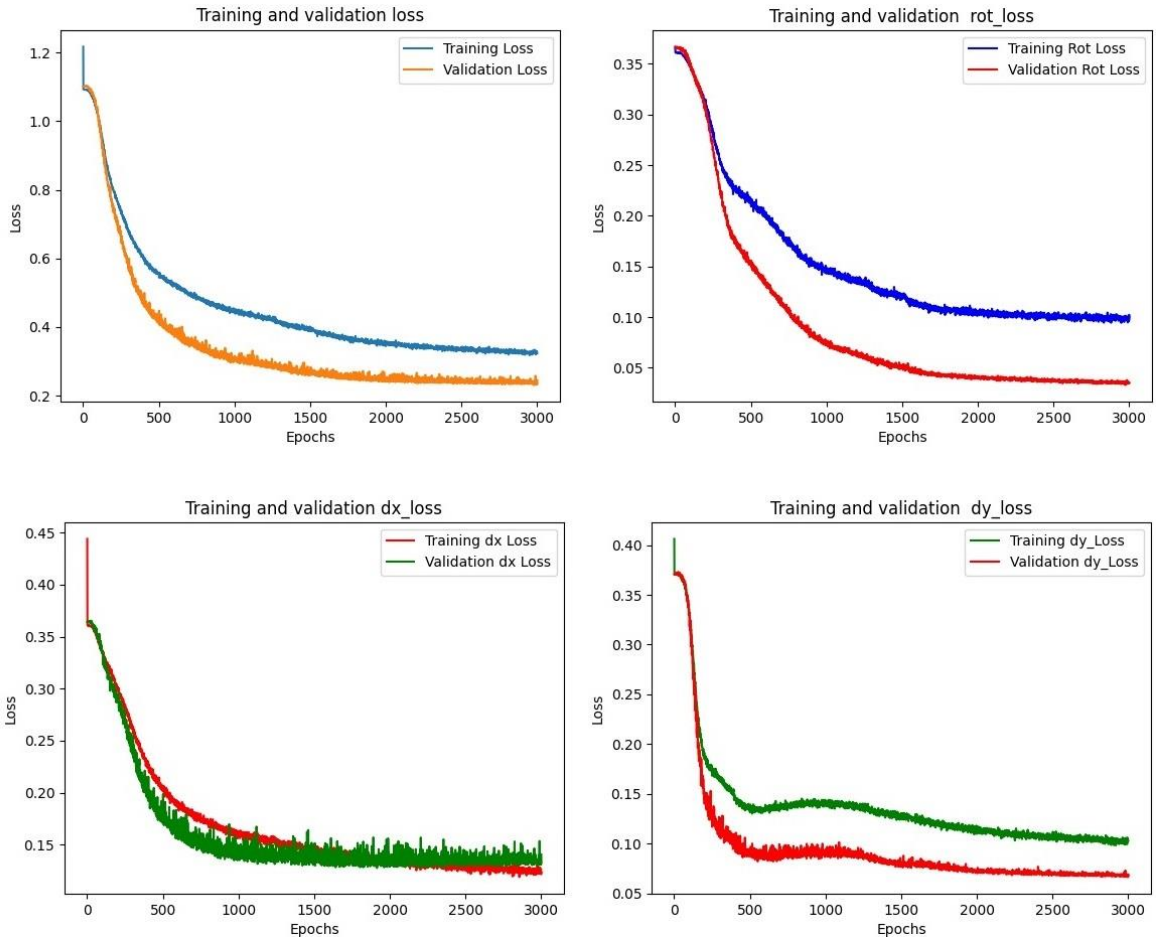
**EKLER**

## EK-1. Yazılım Tabanlı Uygulamada Farklı Hiper Parametreler İçin Yapılan Denemelere Ait Eğitim Doğrulama Grafikleri



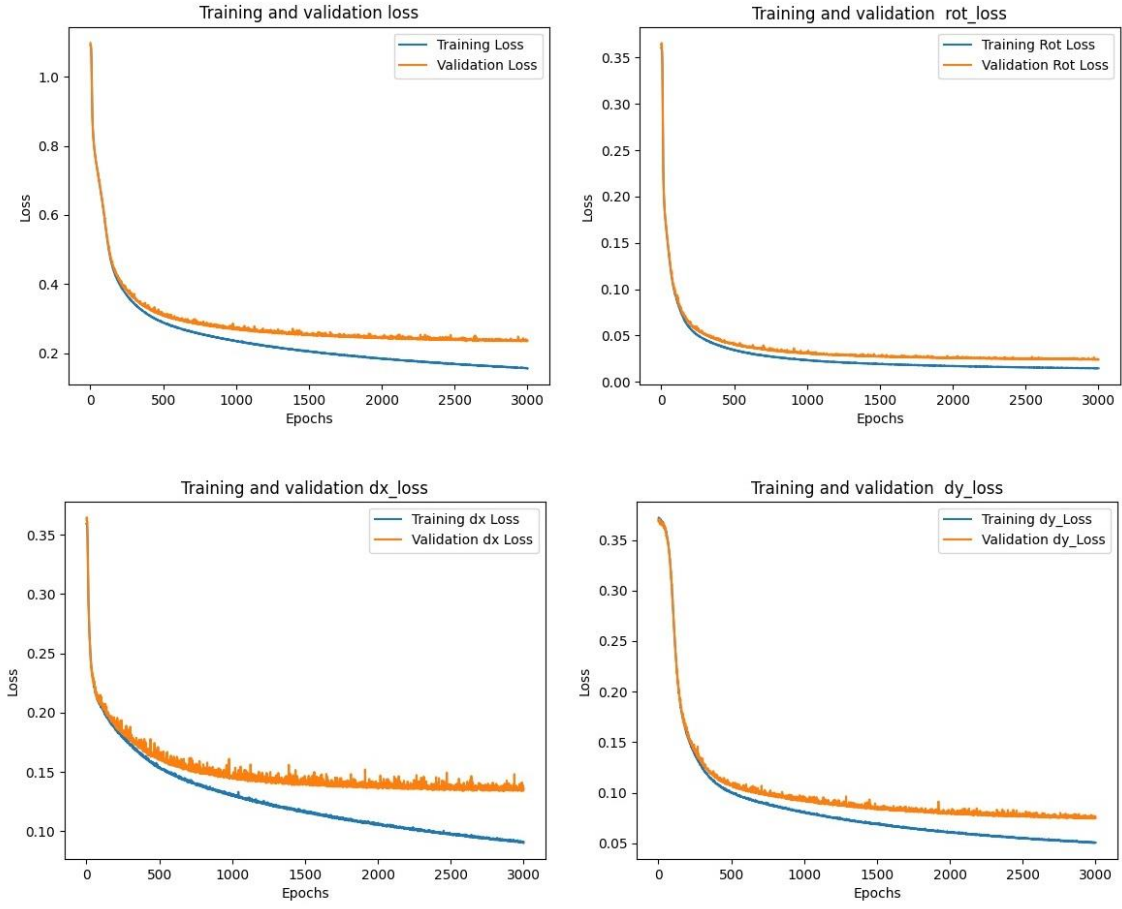
Şekil 1.1. Giriş görüntü boyutu 32x32x3, katmanlar: evrişim (2x2x8), maxpooling (2x2), evrişim (2x2x16), maxpooling (2x2), dropout (0.5) ve devir sayısı 10000 için eğitim ve doğrulama kayıp grafikleri

EK-1. (devam) Yazılım Tabanlı Uygulamada Farklı Hiper Parametreler İçin Yapılan Denemelere Ait Eğitim Doğrulama Grafikleri



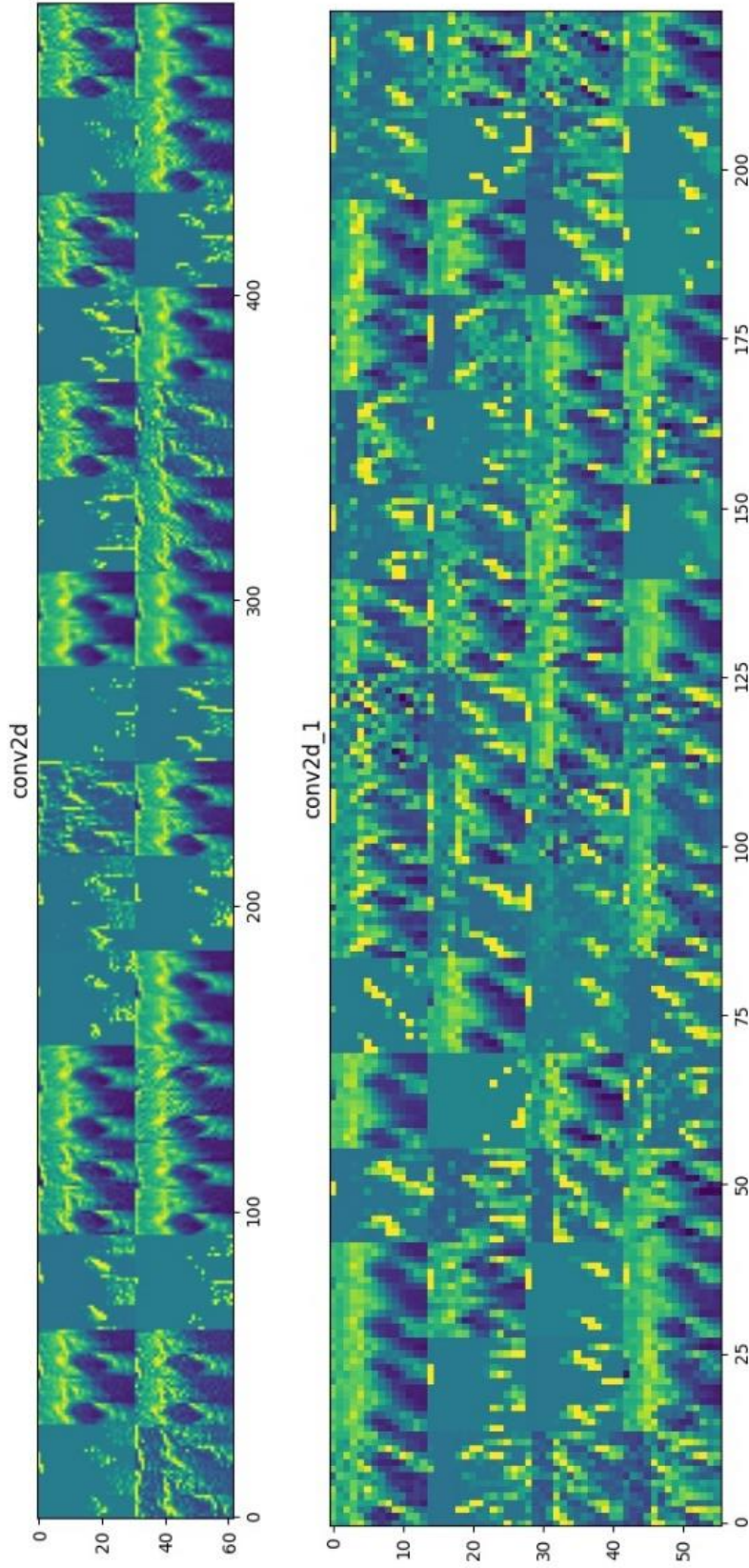
Şekil 1.2. Giriş görüntü boyutu 64x64x3, katmanlar: evrişim (3x3x32), maxpooling (2x2), evrişim (2x2x16), maxpooling (2x2), dropout (0.5) ve devir sayısı 3000 için eğitim ve doğrulama kayıp grafikleri

EK-1. (devam) Yazılım Tabanlı Uygulamada Farklı Hiper Parametreler İçin Yapılan Denemelere Ait Eğitim Doğrulama Grafikleri



Şekil 1.3. Giriş görüntü boyutu 64x64x3, katmanlar: evrişim (3x3x32), maxpooling (2x2), evrişim (2x2x16), maxpooling (2x2), devir sayısı 3000 için eğitim ve doğrulama kayıp grafikleri

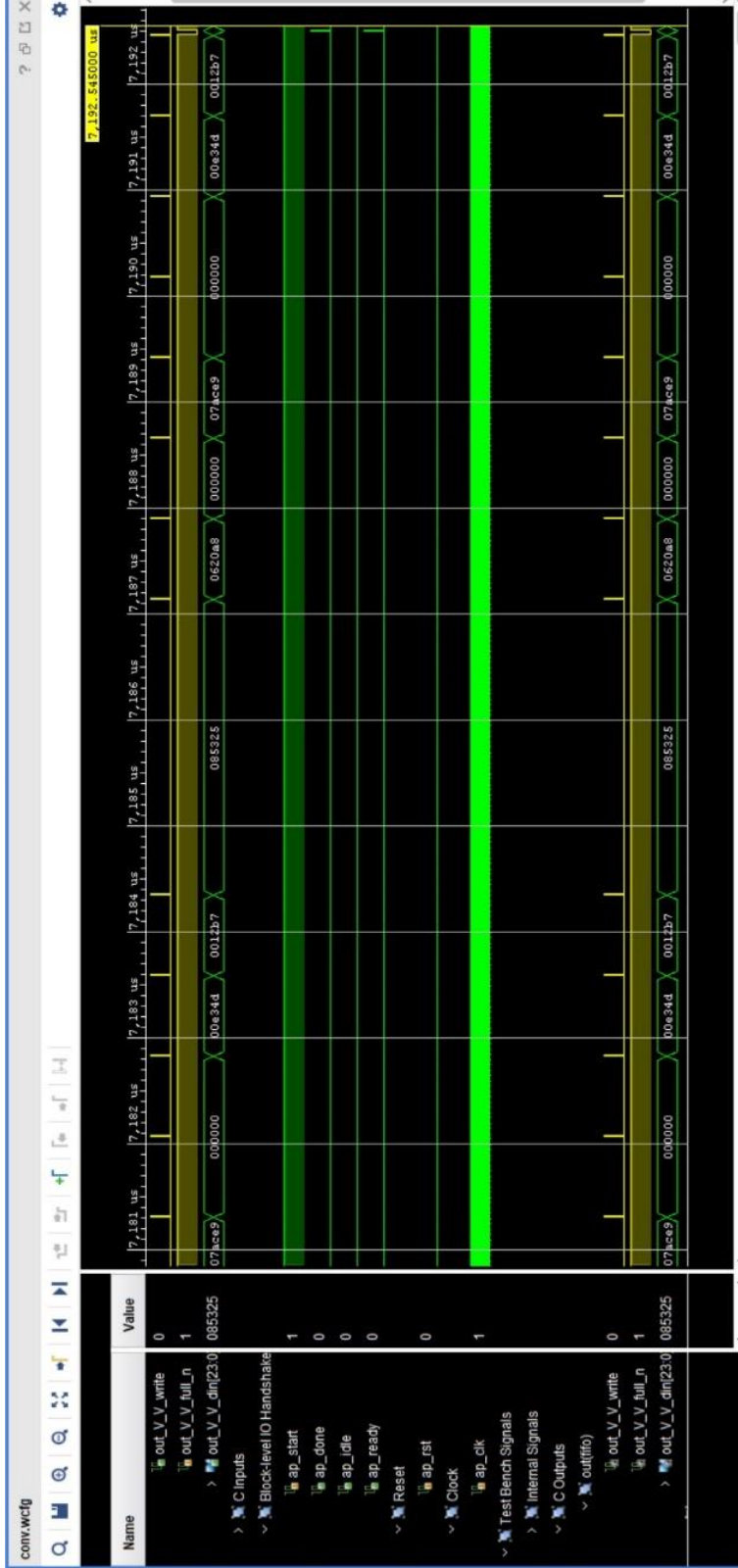
EK-1. (devam) Yazılım Tabanlı Uygulamada Farklı Hiper Parametreler İçin Yapılan Denemelere Ait Eğitim Doğrulama Grafikleri



Şekil 1.4. ESA ara çıktılarının görselleştirilmesi

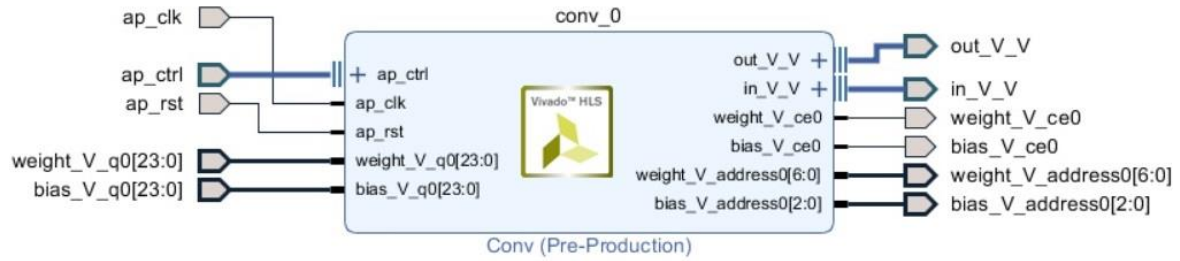


EK-2. (devam) ESA Ağını Oluşturan Evrişim, Havuzlama ve Tam Bağlantılı Katmanların HLS Kullanılarak Tasarlanması, Doğrulması, Modelsim RTL İzleme Dosyasının Oluşturulması ve IP Haline Getirilmesi



Şekil 2.2. Modelsim RTL simülasyon sonucunda zamanlama diyagramı

EK-2. (devam) ESA Ağını Oluşturan Evrişim, Havuzlama ve Tam Bağlantılı Katmanların HLS Kullanılarak Tasarlanması, Doğrulması, Modelsim RTL İzleme Dosyasının Oluşturulması ve IP Haline Getirilmesi



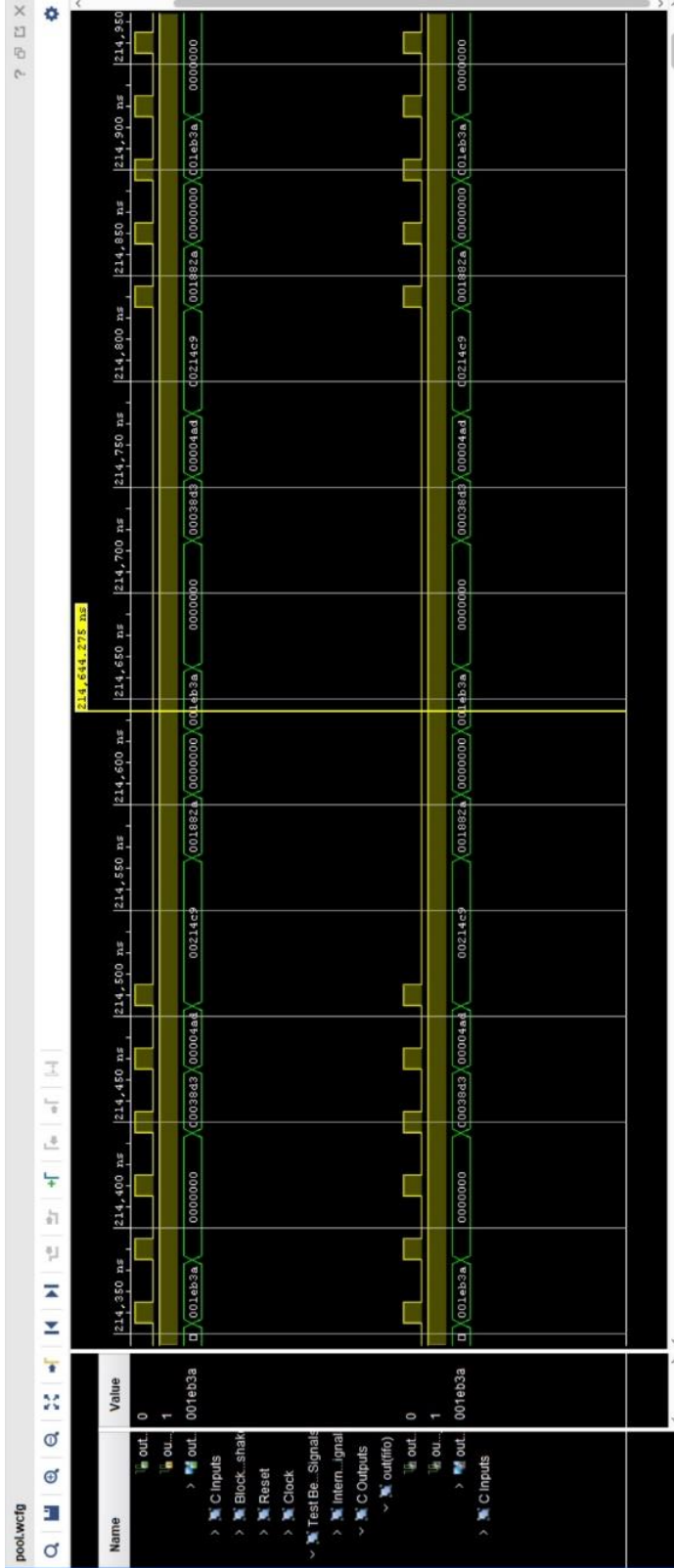
Şekil 2.3. Evrişim modülü blok diyagramı RTL portları

Çizelge 2.2. Havuzlama katmanında kullanılan değerler

Giriş özellik haritası boyutu	Max havuzlama boyutu	stride	Çıkış özellik harita boyutu
29x29x8	2x2	2	14x14x8=1568

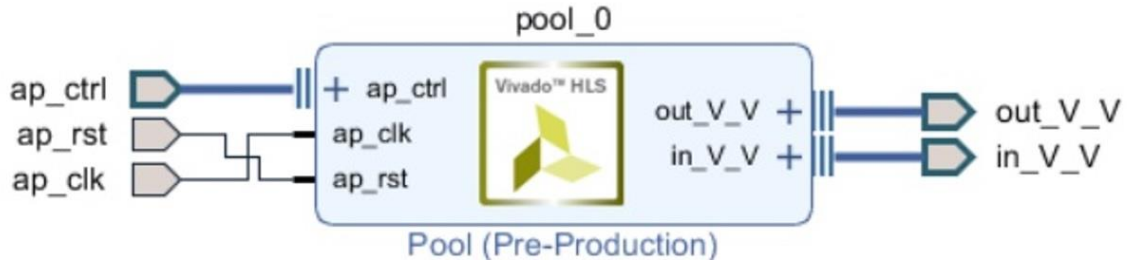


EK-2. (devam) ESA Ağını Oluşturan Evrişim, Havuzlama ve Tam Bağlantılı Katmanların HLS Kullanılarak Tasarlanması, Doğrulaması, Modelsim RTL İzleme Dosyasının Oluşturulması ve IP Haline Getirilmesi



Şekil 2.5. Havuzlama katmanı için Modelsim görüntüsü

EK-2. (devam) ESA Ağını Oluşturan Evrişim, Havuzlama ve Tam Bağlantılı Katmanların HLS Kullanılarak Tasarlanması, Doğrulması, Modelsim RTL İzleme Dosyasının Oluşturulması ve IP Haline Getirilmesi



Şekil 2.6. Havuzlama katmanı blok diyagramı RTL portları

```

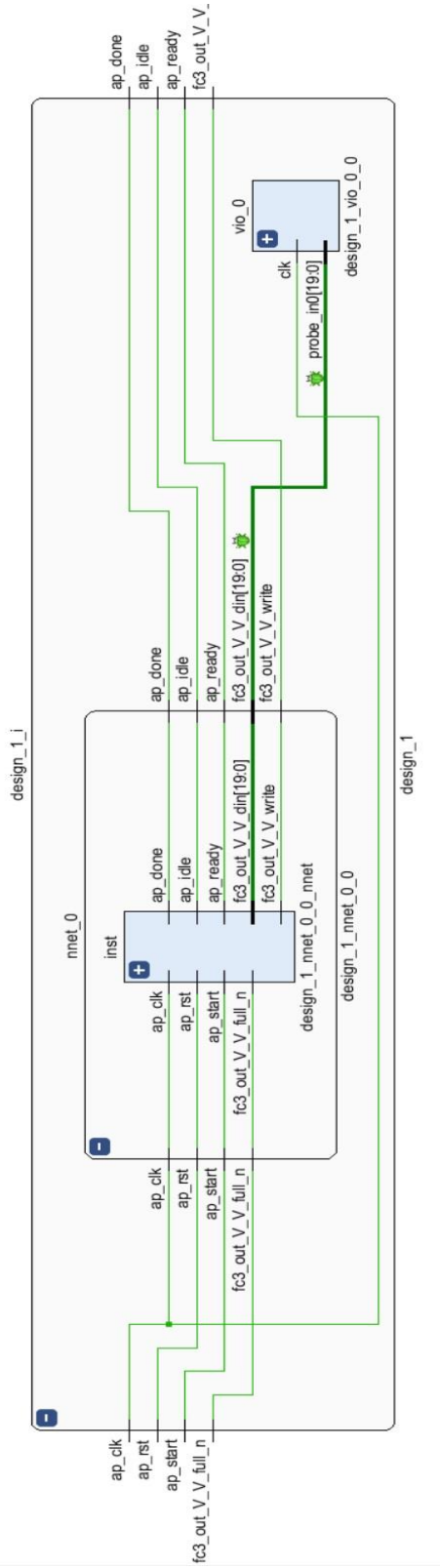
ret value= 0.000000      RTL_value= 0.000000
ref value= 0.200348      RTL_value= 0.200350
ref value= 0.219789      RTL_value= 0.219788
ref value= 0.945204      RTL_value= 0.945203
ref value= 0.000000      RTL_value= 0.000000
ref value= 0.769302      RTL_value= 0.769301
ref value= 0.087621      RTL_value= 0.087618
ref value= 0.734820      RTL_value= 0.734820
ref value= 0.889984      RTL_value= 0.889987
ref value= 0.047663      RTL_value= 0.047664
ref value= 0.000000      RTL_value= 0.000000
ref value= 0.000000      RTL_value= 0.000000
ref value= 0.000000      RTL_value= 0.000000
ref value= 0.127022      RTL_value= 0.127020
ref value= 0.000000      RTL_value= 0.000000
ref value= 0.000000      RTL_value= 0.000000
ref value= 0.441880      RTL_value= 0.441879
ref value= 0.000000      RTL_value= 0.000000
ref value= 0.000000      RTL_value= 0.000000
ref value= 0.000000      RTL_value= 0.000000
ref value= 0.000000      RTL_value= 0.000000
ref value= 0.000000      RTL_value= 0.000000
ref value= 0.000000      RTL_value= 0.000000
ref value= 0.411058      RTL_value= 0.411057
ref value= 0.148380      RTL_value= 0.148381
ref value= 0.141357      RTL_value= 0.141359
ref value= 0.000000      RTL_value= 0.000000
ref value= 0.039523      RTL_value= 0.039526
ref value= 0.000000      RTL_value= 0.000000
DONE: 120 out of 120 are correct

INFO: [SIM 211-1] Csim done with 0 errors.
INFO: [SIM 211-3] ***** CSIM finish *****
Finished C simulation.

```

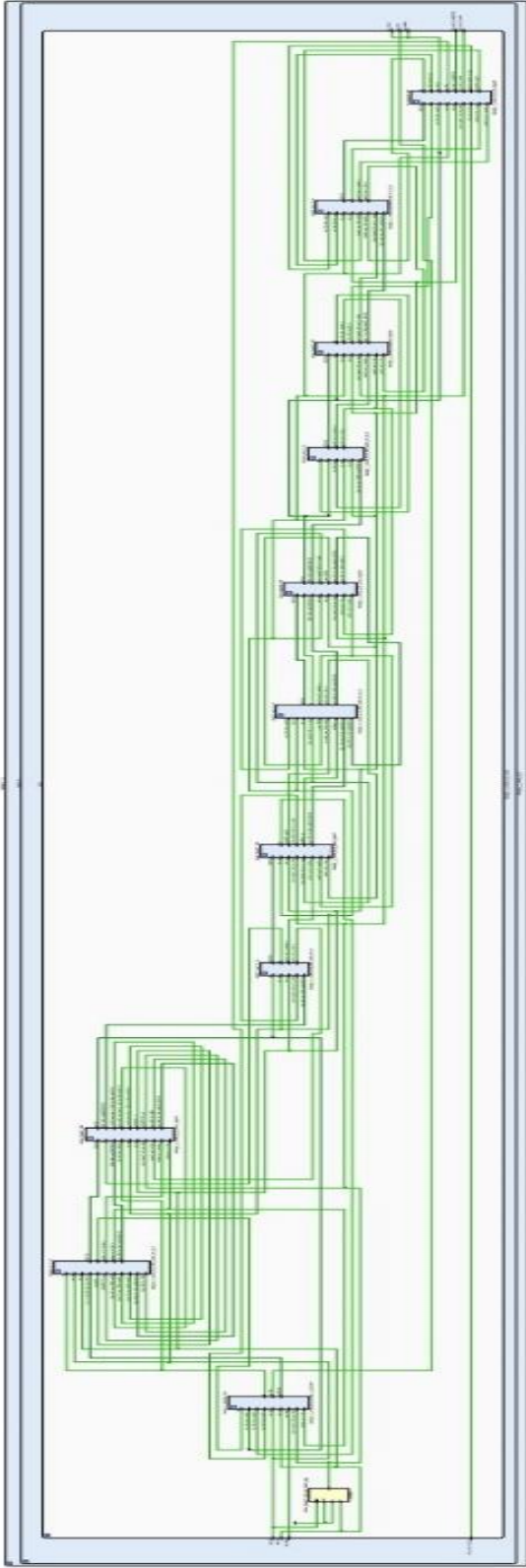
Şekil 2.7. Tam bağlantılı katman C benzetim sonucu

## EK-3. RTL Blok Diyagramları



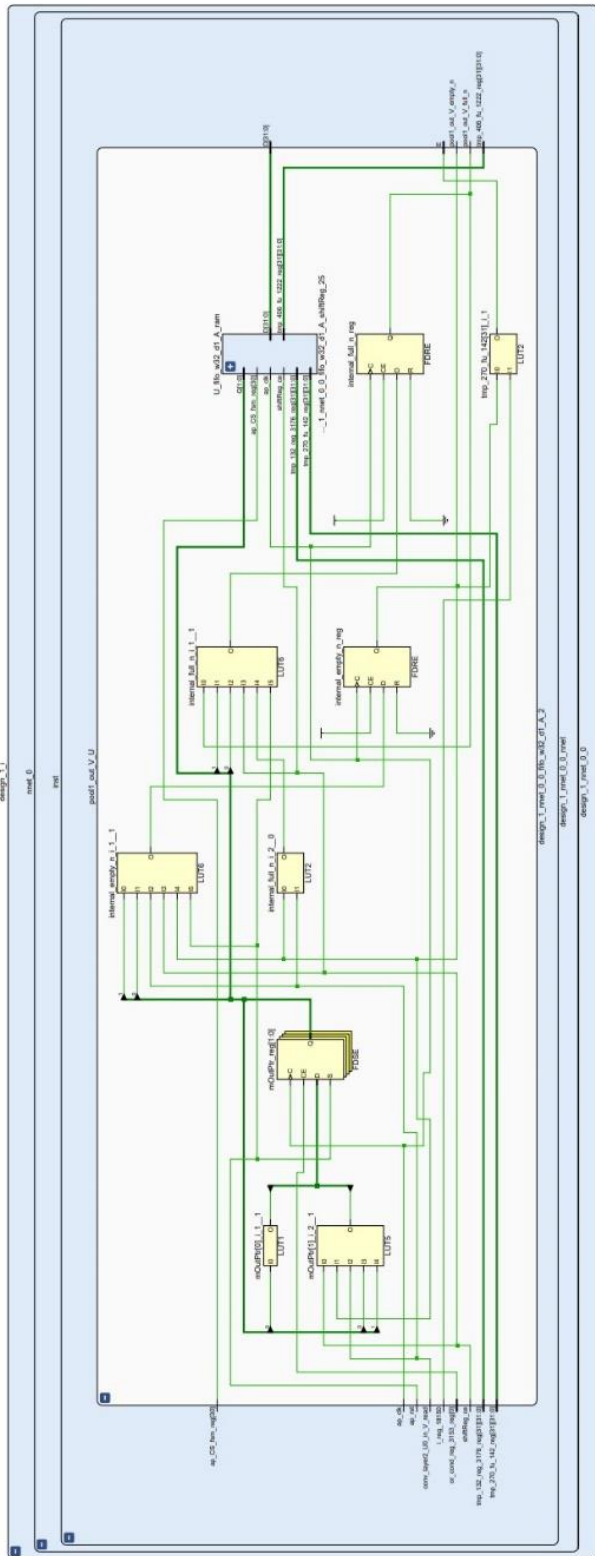
Şekil 3.1. Genel görünüm

## EK-3. (devam) RTL Blok Diyagramları



Şekil 3.2. Genel görünümün açılmış hali

## EK-3. (devam) RTL Blok Diyagramları



Şekil 3.3. Havuzlama katmanı blok diyagramı

2. Guzel Aydin, S., ve Bilge, H. Ş. (2023). *Optimal hardware implementation for end-to-end CNN-based classification*. 4th International Conference on Innovative Trends in Information Technology (ICITIIT), Kottayam, India, 1-6, doi: 10.1109/ICITIIT57246.2023.10068601.
3. Guzel Aydin, S., Bilge, H. Ş., ve Hardalaç, F. (2022). *Supervised rigid image registration with CNN for MR brain images*. In 2022 Innovations in Intelligent Systems and Applications Conference (ASYU), Antalya, Turkey, 1-5.
4. Guzel Aydin, S., ve Bilge, H. Ş. (2022). Efficient Hardware Optimization for CNN. *International Journal of Multidisciplinary Studies and Innovative Technologies*, 6(1), 38-44.
5. Guzel Aydin, S., ve Bilge, H. Ş. (2021). *FPGA-Based Implementation of Convolutional Layer Accelerator Part for CNN*. In 2021 Innovations in Intelligent Systems and Applications Conference (ASYU), Elazig, Turkey, 1-6.
6. Guzel Aydin, S., ve Bilge, H. Ş. (2020). *FPGA based implementation of sigmoid function using different approaches*. In 2020 28th Signal Processing and Communications Applications Conference (SIU), Gaziantep, Turkey, 1-4.
7. Guzel Aydin, S., Kaya, T., ve Guler, H. (2016). Heart rate variability (HRV) based feature extraction for congestive heart failure. *International Journal of Computer and Electrical Engineering*, 8(4), 275.
8. Guzel Aydin, S., Kaya, T., ve Guler, H. (2016). Wavelet-based study of valence–arousal model of emotions on EEG signals with LabVIEW. *Brain Informatics*, 3, 109-117.
9. Güzel, S., Kaya, T., ve Güler, H. (2015). *LabVIEW-based analysis of EEG signals in determination of sleep stages*. In 2015 23rd Signal Processing and Communications Applications Conference (SIU), Malatya, Turkey, 799-802.

## Projeler

1. TÜBİTAK ARDEB 1002 Projesi: 121E393, FPGA Üzerinde Evrişimsel Sinir Ağları Hızlandırma Yöntemlerinin İncelenmesi ve Etkin Bir Yöntemin Geliştirilmesi, Doktora Bursiyeri, 2022
2. TÜBİTAK ARDEB 1001 Projesi: 122E140, Ultrason RF İşaretlerinin Oluşturduğu Görüntülere Donanım Tabanlı Hızlandırılmış Derin Öğrenme Yöntemleri Uygulanarak B-Mod Görüntü Çözünürlüğünün Gerçek Zamanlı Artırılması, Doktora Bursiyeri, 2023.

## Hobiler

Kitap okumak, resim yapmak



*Gazili olmak ayrıcalıktır*