
GAZİ ÜNİVERSİTESİ
Gazi
1926

GÖRÜNTÜ ÇÖZÜNÜRLÜĞÜ İYİLEŞTİRME YÖNTEMLERİNİN FPGA ÜZERİNDE GERÇEKLENMESİ

Furkan DOĞAR

YÜKSEK LİSANS TEZİ
ELEKTRİK-ELEKTRONİK MÜHENDİSLİĞİ ANA BİLİM DALI

GAZİ ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

OCAK 2026

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.

Furkan DOĞAR

20 / 01 / 2026

GÖRÜNTÜ ÇÖZÜNÜRLÜĞÜ İYİLEŞTİRME YÖNTEMLERİNİN FPGA ÜZERİNDE GERÇEKLENMESİ

(Yüksek Lisans Tezi)

Furkan DOĞAR

GAZİ ÜNİVERSİTESİ

FEN BİLİMLERİ ENSTİTÜSÜ

Ocak 2026

ÖZET

Tek görüntü süper çözünürlük (SISR), özellikle gömülü görüntü işleme, tıbbi görüntüleme, uydu tabanlı gözlem ve otonom sistemlerde yüksek çözünürlüklü çıktılara ihtiyaç duyulan uygulamalarda önemli bir araştırma alanı hâline gelmiştir. Bu çalışmada, literatürde yüksek performansı ve düşük parametre sayısı ile bilinen FSRCNN (Hızlı Süper Çözünürlüklü Evrişimli Sinir Ağı) modeli, modern FPGA (Sahada Programlanabilir Kapı Dizileri) mimarilerine uyarlanmak üzere XCZU7EV-2FBVB900I Zynq UltraScale+ MPSoC'si (Çok İşlemcili Yonga üstü Sistem) üzerinde Vitis 2025.1 geliştirme ortamı kullanılarak Vitis Vision kütüphanesi fonksiyonlarıyla HLS (Yüksek Düzeyli Sentez) tasarımı yapılarak modernize edilmiştir. Mevcut FSRCNN FPGA uygulamasının yalnızca mavi renk kanalı üzerinde işlem yapması, xfOpenCV kütüphanesine bağımlı olması ve sınırlı pragma optimizasyonları nedeniyle verimsiz çalışması; bu çalışmanın temel motivasyonunu oluşturmuştur. Çalışma kapsamında FSRCNN mimarisi YCbCr (Parlaklık, Mavi renk kanalının parlaklıktan farkı, Kırmızı renk kanalının parlaklıktan farkı) renk uzayına uyarlanmış ve büyütme aşamasında ters konvolüsyonun yol açtığı periyodik artefaktlar yeniden boyutlandırmalı konvolüsyon yöntemi kullanılarak giderilmiştir. Önerilen tasarım RTL (Kayıt Aktarım Düzeyi) seviyesinde sentezlenmiş; C/RTL kosimülasyon (C ve RTL Düzeyinde Birlikte Doğrulama) başarıyla tamamlanmış ve ortaya çıkan IP (Fikri Mülkiyet) çekirdeği 32x32 giriş görüntüsü kullanılarak kaynak kullanımı, gecikme süresi ve PSNR (Tepe Sinyal-Gürültü Oranı) performansı bakımından xfOpenCV tabanlı uygulama ile karşılaştırılmıştır. Elde edilen sonuçlar; önerilen tasarımın LUT (Arama Tablosu) ve BRAM (Blok Rastgele Erişimli Bellek) kullanımını anlamlı ölçüde azalttığını, gecikmeyi yaklaşık üç kata kadar iyileştirdiğini ve yüksek frekans ayrıntılarında daha kararlı bir görsel kalite sunduğunu göstermektedir. Bu çalışma, hem FSRCNN mimarisinin güncel FPGA ekosistemine uygun biçimde yenilenmesi hem de büyütme aşamasındaki yapısal bozulmaların donanım farkındalıklı çözümlerle giderilmesi açısından literatüre özgün ve yeniden üretilebilir bir katkı sunmaktadır.

Bilim Kodu : 93417

Anahtar Kelimeler : Gömülü sistemler, derin öğrenme tabanlı görüntü işleme, FPGA, sayısal tasarım, HLS, FSRCNN

Sayfa Adedi : 55

Danışman : Prof. Dr. Hasan Şakir BİLGE

IMPLEMENTATION OF IMAGE RESOLUTION ENHANCEMENT METHODS ON FPGA

(M. Sc. Thesis)

Furkan DOĞAR

GAZİ UNIVERSITY

GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES

January 2026

ABSTRACT

Single Image Super-Resolution (SISR) has become an essential research field in applications that require high-resolution outputs under strict computational constraints, including embedded vision systems, medical imaging, satellite-based observation, and autonomous platforms. In this study, the FSRCNN (Fast Super-Resolution Convolutional Neural Network) model, which is known in the literature for its high performance and low parameter count, was modernized for modern FPGA (Field-Programmable Gate Array) architectures and implemented on the XCZU7EV-2FBVB900I Zynq UltraScale+ MPSoC (Multi-Processor System on Chip) using the Vitis 2025.1 development environment and Vitis Vision library functions through a High-Level Synthesis (HLS) workflow. The primary motivation for this study arises from limitations observed in the existing FSRCNN FPGA implementation, which operates solely on the blue color channel, depends on the deprecated xfOpenCV library, and exhibits inefficient performance due to restricted use of pragma-based optimization. In this study, the FSRCNN pipeline was adapted to the YCbCr (Luma, Blue-difference Chroma, Red-difference Chroma) color space, and periodic artifacts introduced by transposed convolution in the upsampling stage were eliminated through a resize-convolution approach. The proposed design was synthesized at the Register Transfer Level (RTL), successfully passed C/RTL co-simulation (combined verification of C-level and RTL-level models), and the resulting Intellectual Property (IP) core was evaluated using a 32x32 input image in comparison with an xfOpenCV-based reference design in terms of resource utilization, latency, and Peak Signal-to-Noise Ratio (PSNR). The results demonstrate that the redesigned architecture significantly reduces LUT (Look-Up Table) and BRAM (Block Random Access Memory) usage, improves latency by nearly a factor of three, and provides more stable visual quality in high-frequency regions. Overall, this study offers a reproducible and hardware-aware enhancement of the FSRCNN architecture for contemporary FPGA ecosystems, contributing a structurally robust solution to upsampling artifacts while modernizing a widely used super-resolution model for efficient hardware deployment.

Science Code : 93417

Key Words : Embedded systems, deep learning-based image processing, FPGA, digital design, HLS, FSRCNN

Page Number : 55

Supervisor : Prof. Dr. Hasan Şakir BİLGE

TEŞEKKÜR

Lisans mezuniyetimle başlayan çalışma hayatı, beni fiziksel olarak akademiden uzaklaştırsa da uzun süre geçmeden yüksek lisansla birlikte öğrenmeye devam edebildiğim için çok şanslı hissediyorum. Engin tecrübesi, samimi yaklaşımı ve derin bilgisiyle sadece tez çalışmam sırasında değil tüm yüksek lisans eğitimim boyunca beni hiç yalnız bırakmayan sevgili Hocam Prof. Dr. Hasan Şakir BİLGE'ye teşekkür ediyorum. Ders dönemimde kendisiyle tanışabildiğim ve tez dönemimde beraber çalışabildiğim için kendimi çok ayrıcalıklı sayıyorum. Çalıştığım kurum Aselsan A.Ş.'ye; enstitümüz ve ana bilim dalımızla koordinasyonumuzu sağlayan Aselsan Akademi çalışanlarına; eğitimimi çalışma hayatımla uyum içinde devam ettirmemi sağlayan ve destekleyen bölüm müdürüm Ümit BÜYÜKKELEŞ'e ve ekip liderim Osman Ulaş ŞAHİN'e; gerçekleştirdiğim çalışma boyunca yardımlarını aldığım meslektaşlarım Mehmet Emin YALÇIN, Yunus ESERGÜN ve Bahadır YILDIRIM'a teşekkürü borç bilirim. Hayatımın her anında, fedakarlık göstererek yanımda olan canım annem Gürzida DOĞAR ve kıymetli babam Mehmet DOĞAR'a müteşekkirim. Bu çalışma TÜBİTAK ARDEB 122E140 numaralı 1001 Araştırma Projesi kapsamında desteklenmiştir.

İÇİNDEKİLER

Sayfa

ÖZET iv

ABSTRACT v

TEŞEKKÜR vi

İÇİNDEKİLER vii

ÇİZELGELERİN LİSTESİ ix

ŞEKİLLERİN LİSTESİ x

RESİMLERİN LİSTESİ xi

SİMGELER VE KISALTMALAR xiii

1. GİRİŞ 1

2. ÇÖZÜNÜRLÜK İYİLEŞTİRME YÖNTEMLERİ 9

2.1. Bikübik Enterpolasyon 9

2.2. Süper Çözünürlüklü Evrişimli Sinir Ağı (SRCNN) 10

2.3. Hızlı Süper Çözünürlüklü Evrişimli Sinir Ağı (FSRCNN) 11

2.4. Verimli Alt-Piksel Evrişimli Sinir Ağı (ESPCN) 12

3. TASARIM ORTAMLARI 13

3.1. Vitis ve Kütüphaneleri 13

3.2. OpenCV Kütüphaneleri ve Adaptasyonu 14

3.3. Vivado XSIM Simülasyon Ortamı 16

4. HLS TASARIMI VE DENEYSEL SONUÇLAR 19

4.1. Yöntem Seçimi ve Model Parametrelerinin Belirlenmesi 19

4.2. Vitis Vision Kütüphanesi ile Modernizasyon 20

4.3. FSRCNN Mimarisi 22

4.3.1. Öznitelik çıkarımı 23

Sayfa

4.3.2. Daraltma 24
4.3.3. Haritalama 24
4.3.4. Genişletme 24
4.3.5. Ters konvolüsyon 25
4.4. FSRCNN Mimarisinin Son Katmanındaki İyileştirme Çalışmaları 28
4.4.1. Yeniden boyutlandırmalı konvolüsyon yöntemi 28
4.4.2. Çok fazlı ters konvolüsyon ve artefakt optimizasyonu 32
4.5. Deneysel Sonuçlar 34
4.5.1. Görsel kalite ve performans metrikleri analizi 34
4.5.2. Kaynak kullanımı ve performans analizi 42
5. TARTIŞMA 45
6. SONUÇ VE ÖNERİLER 49
KAYNAKLAR 53
ÖZGEÇMİŞ 55

ÇİZELGELERİN LİSTESİ

Çizelge **Sayfa**

Çizelge 1.1. Literatürdeki güncel FPGA/Donanım tabanlı görüntü işleme çalışmalarının performans metrikleri 5

Çizelge 3.1. FSRCNN testbench aşamasında kullanılan başlıca OpenCV fonksiyonları..... 15

Çizelge 4.1. FSRCNN tasarımında kullanılan fonksiyonların karşılaştırılması 21

Çizelge 4.2. İterasyon 1-2 tasarım kaynak kodlarının çıktılarının PSNR kalite metrikleri..... 31

Çizelge 4.3. İterasyon 1, 2 ve 3 tasarım kaynak kodlarının çıktılarının PSNR kalite metrikleri..... 33

Çizelge 4.4. Çeşitli çalışmalar ve önerilen tasarımının çıktılarının görsel kalite performans metrikleri..... 36

Çizelge 4.5. Farklı çözünürlük iyileştirme yöntemlerinin ultrason görüntüleri üzerindeki performans metrikleri ortalamaları 41

Çizelge 4.6. xfOpenCV ve önerilen tasarımının kaynak kullanımı ve gecikme verileri..... 42

Çizelge 4.7. Çeşitli çalışmalar ve önerilen tasarımının çıktılarının kaynak kullanımları 43

ŞEKİLLERİN LİSTESİ

Şekil **Sayfa**

Şekil 4.1. Önerilen FSRCNN ağ yapısı (x2 yükseltme) 23

Şekil 4.2. Testbench kodunun akış diyagramı 26

RESİMLERİN LİSTESİ

Resim **Sayfa**

Resim 4.1. Tasarım kaynak kodunun çıktısının görsel analizi a. 32x32 giriş görseli, b. Giriş görselinin 64x64 zemin gerçeği versiyonu, c. Bikübik enterpolasyonla elde edilmiş 64x64 çıkış görseli, d. Tasarım kaynak kodunun 64x64 çıkış görseli..... 27

Resim 4.2. İterasyon 1-2 tasarım kaynak kodlarının çıktılarının görsel analizi a. İterasyon-1 (ters konvolüsyon yöntemi) 64x64 çıkış görseli, b. İterasyon-2 (yeniden boyutlandırmalı konvolüsyon yöntemi) 64x64 çıkış görseli..... 31

Resim 4.3. İterasyon-3 tasarım kaynak kodunun çıktısının karşılaştırmalı görsel analizi a. 32x32 giriş görseli, b. Bikübik enterpolasyonla elde edilmiş 64x64 çıkış görseli, c. İterasyon-2 64x64 çıkış görseli, d. İterasyon-3 64x64 çıkış görseli..... 33

Resim 4.4. xfOpenCV uygulamasının ve önerilen tasarımının çıktılarının görsel analizi a. 32x32 giriş görseli, b. Giriş görselinin 64x64 zemin gerçeği versiyonu, c. xfOpenCV uygulamasının 64x64 çıkış görseli, d. Önerilen tasarımın (İterasyon-3) 64x64 çıkış görseli..... 35

Resim 4.5. Ultrason veri setindeki fetal baş görüntüsünün farklı çözünürlük iyileştirme yöntemleri altındaki çıkış görüntüleri a. Veri setinden alınan Ultrason görüntüsünün özgün hali, b. Ultrason görüntüsünün 32x32 boyutuna dönüştürülmüş hali, c. Bikübik enterpolasyonla elde edilmiş 64x64 çıkış görüntüsü, d. xfOpenCV uygulamasının 64x64 çıkış görüntüsü, e. Önerilen tasarımın (İterasyon-3) 64x64 çıkış görüntüsü..... 37

Resim 4.6. Ultrason veri setindeki trans-serebellar görüntüsünün farklı çözünürlük iyileştirme yöntemleri altındaki çıkış görüntüleri a. Veri setinden alınan Ultrason görüntüsünün özgün hali, b. Ultrason görüntüsünün 32x32 boyutuna dönüştürülmüş hali, c. Bikübik enterpolasyonla elde edilmiş 64x64 çıkış görüntüsü, d. xfOpenCV uygulamasının 64x64 çıkış görüntüsü, e. Önerilen tasarımın (İterasyon-3) 64x64 çıkış görüntüsü..... 38

Resim 4.7. Ultrason veri setindeki trans-talamik düzlem görüntüsünün farklı çözünürlük iyileştirme yöntemleri altındaki çıkış görüntüleri a. Veri setinden alınan Ultrason görüntüsünün özgün hali, b. Ultrason görüntüsünün 32x32 boyutuna dönüştürülmüş hali, c. Bikübik enterpolasyonla elde edilmiş 64x64 çıkış görüntüsü, d. xfOpenCV uygulamasının 64x64 çıkış görüntüsü, e. Önerilen tasarımın (İterasyon-3) 64x64 çıkış görüntüsü..... 39

Resim **Sayfa**

Resim 4.8. Ultrason veri setindeki trans-ventriküler kesit görüntüsünün farklı çözünürlük iyileştirme yöntemleri altındaki çıkış görüntüleri a. Veri setinden alınan Ultrason görüntüsünün özgün hali, b. Ultrason görüntüsünün 32x32 boyutuna dönüştürülmüş hali, c. Bikübik enterpolasyonla elde edilmiş 64x64 çıkış görüntüsü, d. xfOpenCV uygulamasının 64x64 çıkış görüntüsü, e. Önerilen tasarımın (İterasyon-3) 64x64 çıkış görüntüsü..... 40

SİMGELER VE KISALTMALAR

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

Simgeler	**Açıklamalar**
1x1	1×1 konvolüsyon çekirdeği
2x	2 kat büyütme oranı
2x2	2×2 blok yapısı
32x32	32×32 piksel boyutunda giriş görüntü bloğu
3x3	3×3 konvolüsyon çekirdeği
64x64	64×64 piksel boyutunda çıkış görüntü bloğu
dB	Desibel
GB	Gigabayt
GHz	Gigahertz
us, µs	Mikrosaniye

Kısaltmalar	**Açıklamalar**
AMD	Advanced Micro Devices
AMIQ	AMIQ Consulting
AGC	Animation-based Generative Codec (Animasyon Tabanlı Üretken Kodek)
BRAM	Block Random Access Memory (Blok Rastgele Erişimli Bellek)
CNN	Convolutional Neural Network (Evrişimli Sinir Ağı)
CPU	Central Processing Unit (Merkezi İşlem Birimi)
ESPCN	Efficient Sub-Pixel Convolutional Neural Network (Verimli Alt-Piksel Evrişimli Sinir Ağı)
FPGA	Field Programmable Gate Array (Sahada Programlanabilir Kapı Dizileri)
FSRCNN	Fast Super-Resolution Convolutional Neural Network (Hızlı Süper Çözünürlüklü Evrişimli Sinir Ağı)

Kısaltmalar	Açıklamalar
GPU	Graphics Processing Unit (Grafik İşlem Birimi)
HLS	High-Level Synthesis (Yüksek Düzeyli Sentez)
IP	Intellectual Property (Fikri Mülkiyet)
LUT	Look-Up Table (Arama Tablosu)
MPSoC	Multiprocessor System-on-Chip (Çok İşlemcili Yonga Üstü Sistem)
PSNR	Peak Signal-to-Noise Ratio (Tepe Sinyal-Gürültü Oranı)
RAM	Random Access Memory **(**Rastgele Erişimli Bellek)
RTL	Register Transfer Level (Kayıt Aktarım Düzeyi)
SISR	Single Image Super-Resolution (Tek Görüntü Süper Çözünürlük)
SoC	System-on-Chip (Yonga Üstü Sistem)
SRCNN	Super-Resolution Convolutional Neural Network (Süper Çözünürlüklü Evrişimli Sinir Ağı)
SSIM	Structural Similarity Index Measure (Yapısal Benzerlik İndeksi)
YCbCr	Luma, Chroma Blue, Chroma Red (Parlaklık, Mavi Renk Kanalının Parlaklıktan Farkı, Kırmızı Renk Kanalının Parlaklıktan Farkı)

1. GİRİŞ

Problem durumu

Tek görüntü süper çözünürlük (Single Image Super-Resolution, SISR), düşük çözünürlüklü bir görüntüden daha yüksek çözünürlüklü bir görüntünün tahmin edilmesine yönelik dönüşümsel bir problem olarak tanımlanmaktadır. Bu problem, düşük çözünürlüklü bir görüntüden, o görüntüye ait yüksek frekanslı ayrıntıların istatistiksel veya model tabanlı yöntemlerle tahmin edilerek yüksek çözünürlüklü bir çıktının oluşturulmasını hedefler. SISR kapsamında amaç, kaybolan yüksek frekans bileşenlerini geri kazandırmak ve giriş görüntüsünün uzaysal çözünürlüğünü bilgi kaybını en aza indirecek biçimde artırmaktır. Bununla birlikte, eldeki bilgi miktarı sınırlı olduğundan tek bir düşük çözünürlüklü görüntüden tekil bir yüksek çözünürlüklü görüntünün üretilmesi deterministik değildir; bu nedenle modellerin başarımı büyük ölçüde istatistiksel çıkarım kapasitesine, öğrenilmiş filtrelere ve hesaplama süreçlerinin doğruluğuna bağlıdır. SISR'nin uygulama alanları, gömülü görüntü işleme, tıbbi görüntüleme, uydu tabanlı gözlem ve otonom sistemler gibi yüksek çözünürlüklü çıktılara ihtiyaç duyulan senaryolarda belirginleşmektedir.

Geleneksel yöntemler arasında yer alan bikübik enterpolasyon, düşük hesaplama maliyeti sunmasına karşın görüntüdeki ayrıntıları yumuşatma eğilimi göstermekte ve yüksek frekanslı dokuları yeterli düzeyde geri üretememektedir (Dong, Loy, He ve Tang, 2014). Derin öğrenme yöntemlerinin görüntü işleme alanında yaygınlaşmasıyla birlikte süper çözünürlük modelleri, hesaplama performansı ve görsel kalite açısından önemli ilerlemeler kaydetmiştir. İlk derin öğrenme tabanlı model olarak kabul edilen SRCNN (Süper Çözünürlüklü Evrişimli Sinir Ağı), bikübik enterpolasyonla büyütülmüş görüntü üzerinde üç aşamalı bir CNN mimarisi kullanarak iyileştirme gerçekleştirmiştir (Dong ve diğerleri, 2014). Bu yaklaşım, öğrenilmiş filtreler sayesinde klasik enterpolasyonun sınırlamalarını aşmış olsa da işlem uzayının yüksek çözünürlüklü bölgede olması nedeniyle hesaplama maliyetini artırmaktadır. Bu sınırlılığı gidermek amacıyla geliştirilen FSRCNN (Hızlı Süper Çözünürlüklü Evrişimli Sinir Ağı) ise tüm işlemleri düşük çözünürlüklü uzayda gerçekleştiren beş aşamalı (öznitelik çıkarımı, daraltma, haritalama, genişletme ve ters konvolüsyon) kompakt bir yapı sunarak hem parametre sayısı hem de işlem süresi açısından daha verimli bir performans sergilemiştir (Dong, Loy ve Tang, 2016). FSRCNN

sonrasında literatürde, özellikle FPGA veya benzeri hızlandırıcı donanımlara uygun, hesaplama maliyeti düşük modeller üzerinde yoğunlaşılmış; bu kapsamda ESPCN (Verimli Alt-Piksel Evrişimli Sinir Ağı) gibi büyütme işlemini düşük çözünürlüklü uzayda alt piksel yeniden düzenleme mekanizmasıyla gerçekleştiren yaklaşımlar, donanım uyumu açısından öne çıkmıştır (Shi ve diğerleri, 2016).

Derin öğrenme tabanlı SR modellerinin yüksek doğruluğu, özellikle kaynak kısıtlı uç cihazlarda gerçek zamanlı çalıştırma hedefiyle birlikte değerlendirildiğinde; bellek bant genişliği, işlem gücü ve donanım kaynak tüketimi bakımından önemli bir darboğaz ortaya çıkarmaktadır. Bu noktada Sahada Programlanabilir Kapı Dizileri (FPGA), yeniden yapılandırılabilirlik, enerji verimliliği ve paralel işlem yetenekleri sayesinde Grafik İşlem Birimi (GPU) ve Merkezi İşlem Birimi (CPU) tabanlı çözümlere kıyasla güçlü bir alternatif olarak konumlanmaktadır (Gao, Luo ve Wang, 2025). Literatürde FPGA üzerinde süper çözünürlük uygulamalarına yönelik güncel çalışmaların, donanım kaynaklarının (LUT, DSP, BRAM) verimli kullanımı ile görsel kalite arasındaki dengeyi kurmaya odaklandığı görülmektedir. Örneğin donanım maliyetini azaltmak için ikili sinir ağları yaklaşımıyla geliştirilen ResBinESPCN benzeri mimariler bellek gereksinimini düşürmeyi hedeflerken, ikili ağırlıkların doğası gereği PSNR gibi piksel tabanlı doğruluk metriklerinde kayıplar gözlenebilmektedir (Su, Seng, Smith ve Ang, 2024). Buna karşılık, nicelenmiş Winograd gibi hesaplama azaltıcı yöntemleri hibrit mimari kurgularla birleştiren yaklaşımlar, FPGA üzerinde yüksek işlem hacmi (throughput) ile görsel kaliteyi birlikte hedeflemektedir (Shi ve diğerleri, 2023).

Bu çalışmalara paralel olarak Joseph ve Jayakumar (2024), donanım verimliliğini artırmak adına FSRCNN mimarisinde PReLU yerine ReLU aktivasyon fonksiyonunu kullanmak ve ters konvolüsyon katmanını enterpolasyon ile değiştirmek gibi yapısal sadeleştirmelere gitmişlerdir. Yazarlar, 8-bit niceleme (quantization) tekniklerini kullanarak geliştirdikleri bu optimize mimari (QFSRCNN) ile ZCU104 FPGA kartı üzerinde, görüntü kalitesinden ödün vermeden 1080p'den 4K'ya gerçek zamanlı süper çözünürlük elde edilebileceğini göstermişlerdir. Benzer bir çizgide Ahmed, Ali, Javed ve Iqbal (2024) ise maliyet odaklı Zynq-7000 serisi SoC tabanlı FPGA'lar üzerinde özelleştirilmiş bir FSRCNN hızlandırıcısı tasarlayarak, düşük maliyetli donanımlarda dahi derin öğrenme tabanlı süper çözünürlüğün uygulanabilirliğini araştırmışlardır. Alternatif bir çizgide, çift yönlü ve yeniden yapılandırılabilir enterpolasyon mimarisi öneren BiDSRS+ gibi sistemler ise derin

öğrenme temelli çözümlere kıyasla çok düşük güç tüketimiyle yüksek çözünürlüklü ölçekleme başarısı raporlamaktadır (Amin ve Obermaisser, 2025). Bu çeşitlilik, SR'nin donanım üzerinde gerçekleştirilmesinde model seçimi, kaynak kullanımı, güç tüketimi, gecikme ve görsel kalite eksenlerinde çok boyutlu bir optimizasyon problemi bulunduğunu göstermektedir.

Süper çözünürlük modellerinde büyütme işlemi sırasında karşılaşılan en önemli yapısal sorunlardan biri, ters konvolüsyon katmanının belirli koşullarda düzenli örüntüler üretmesidir. Literatürde "dama tahtası bozulması" olarak adlandırılan bu görsel bozulma, çoğunlukla filtre boyutu ile adım (stride) uzunluğu arasındaki uyumsuzluk ve düzensiz örtüşme nedeniyle ortaya çıkmaktadır (URL-1). Farklı büyütme stratejilerini karşılaştıran çalışmalar, ters konvolüsyon tabanlı yöntemlerin belirli konfigürasyonlarda yapısal bozulmalar üretebildiğini; buna karşılık ardışık yeniden boyutlandırmalı konvolüsyon temelli yaklaşımların daha tutarlı çıktılar sağlayabildiğini göstermiştir (Sugawara, Shiota ve Kiya, 2019). Öte yandan, FPGA üzerinde ters konvolüsyon katmanlarının gerçeklemesine odaklanan çalışmalar, bu katmanların standart konvolüsyona kıyasla daha karmaşık bellek erişim desenleri gerektirdiğini ve yeterince optimize edilmediğinde ciddi donanım verimsizliğine yol açabildiğini vurgulamaktadır (Sestito, Perri ve Stewart, 2023). Buna ek olarak, düşük güçlü sistemlerde iş yükünü görsel algı önceliklerine göre yeniden dağıtan fovea tabanlı değişken çözünürlük işleme gibi teknikler de hesaplama yükünü azaltmak ve enerji verimliliğini artırmak amacıyla kullanılmaktadır (Spagnolo, Corsonello, Frustaci ve Perri, 2023). Bu bulgular, büyütme aşamasının yalnızca görsel kaliteyi değil, aynı zamanda donanım mimarisinin bellek ve paralellik düzenini doğrudan etkilediğini göstermesi bakımından kritik bir problem alanı oluşturmaktadır.

Süper çözünürlük uygulamalarında bir diğer belirleyici boyut, renk uzayı seçimi ve öğrenmenin hangi bileşen üzerinde yürütüldüğüdür. İnsan görme sisteminin parlaklık (luma) bileşenindeki uzaysal değişimlere daha duyarlı olduğu kabulü, birçok yaklaşımda yalnızca parlaklık (Y) kanalına odaklanılmasına neden olmuştur. Süper çözünürlük modellerinin yalnızca Y bileşeninde eğitilmesinin hesaplama maliyetini azalttığı ve PSNR performansının çok kanallı işleme kıyasla kayda değer bir düşüş göstermeyebildiği deneysel olarak raporlanmıştır (Dong ve diğerleri, 2014). Bu bulguyu destekleyen daha güncel analizler, tek kanallı (Y) SR modellerinin çok kanallı modellere göre daha düşük hesaplama gereksinimi ile benzer veya daha yüksek PSNR elde edebildiğini ve bunun

özellikle kaynak kısıtlı sistemlerde performans/verimlilik dengesi açısından avantaj sunduğunu göstermektedir (Chang, Chen, Lin Y.-W., Lin Y.-J. ve Pan, 2024). Dolayısıyla Y kanalına odaklanmak, donanım tabanlı SR tasarımlarında hem hesaplama verimliliği hem de algısal kalite hedefleriyle tutarlı bir problem çerçevesi sunmaktadır.

FPGA tabanlı süper çözünürlük araştırmaları, derin öğrenme modellerinin düşük gecikme süreleriyle donanım üzerinde çalıştırılması açısından giderek önem kazanan bir araştırma alanı hâline gelmiştir. Veri akışı (dataflow) temelli CNN hızlandırmanın FPGA üzerinde yüksek performans sağlayabildiği ve gerçek zamanlı SR elde etmede bellek erişim düzeni ile paralel hesaplama stratejilerinin kritik rol oynadığı literatürde vurgulanmaktadır (Aarrestad ve diğerleri, 2021; Manabe, Shibata ve Oguri, 2016). Süper çözünürlük dışındaki diğer bilgisayarlı görü görevlerinde de FPGA tabanlı hızlandırmayı amaçlayan çalışmalar mevcuttur; örneğin Saidani ve diğerleri (2024) nesne tespiti için YOLOv5 algoritmasını Xilinx Zynq platformunda hızlandırırken, Yanamala ve Pullakandam (2025) uykululuk tespiti için geliştirdikleri hibrit CNN-LSTM sistemiyle uç cihazlarda yüksek enerji verimliliği sunmuşlardır. Örneklendirilen ikinci çalışma, karmaşık ağ yapısı nedeniyle ortalama 73 ms bir gecikme süresine sahiptir ve bu yönüyle gerçek zamanlı video işleme gibi çok düşük gecikme gerektiren uygulamalarda daha hafif ve optimize mimarilere duyulan ihtiyacı göstermektedir (Yanamala ve Pullakandam, 2025). İncelenen bu çalışmalar, FPGA tabanlı tasarımların gecikme ve güç tüketimi metriklerindeki potansiyelini ortaya koymakta ve bu tezin performans değerlendirmeleri için önemli birer kıyas noktası oluşturmaktadır.

Bununla birlikte, FSRCNN'nin FPGA üzerinde gerçekleştirilmesine ilişkin literatürde öne çıkan uygulamalardan biri, Vivado HLS ve xfOpenCV kütüphanesi kullanılarak geliştirilen referans tasarımdır (URL-2). Bu tasarım FSRCNN'nin donanım potansiyelini göstermesi açısından değerli olsa da, yalnızca tek bir renk kanalını işlemesi, eski nesil kütüphane fonksiyonlarına bağımlı olması ve ters konvolüsyon kaynaklı bozulmaları gidermeye yönelik bir optimizasyon içermemesi nedeniyle güncel ihtiyaçlara tam yanıt verememektedir (URL-2). Ayrıca son dönem çalışmalar, görüntü kalitesinin değerlendirilmesinde yalnızca PSNR değil, LPIPS (Öğrenilmiş Algısal Görüntü Yaması Benzerliği) gibi algısal metriklerin de giderek önem kazandığını göstermektedir (URL-3).

Bu çerçevede problem, yalnızca tek görüntüden yüksek çözünürlük üretmek değil; aynı zamanda bu işlemi sınırlı donanım kaynakları altında, uygulanabilir gecikme süreleri ve enerji/kaynak verimliliği ile gerçekleştirebilmektir. Mevcut literatürde FSRCNN'nin güncel AMD araç zinciri (Vitis) ve Vitis Vision kütüphanesiyle, YCbCr renk uzayında ve büyütme aşamasındaki bozulmaları hedefleyen yapısal iyileştirmelerle yeniden yapılandırılmasına yönelik güncel ve yeniden üretilebilir bir uygulama boşluğu bulunduğu değerlendirilmektedir (URL-2). Bu tezdeki çalışma, FSRCNN mimarisinin modern Vitis Vision ekosistemine uyarlanması, büyütme aşamasında yeniden boyutlandırmalı konvolüsyon yaklaşımıyla görsel bozulmaların azaltılması ve sonuçların gecikme, kaynak kullanımı, güç tüketimi ve PSNR gibi metriklerle çok boyutlu biçimde raporlanması üzerinden bu boşluğu kapatmayı hedeflemektedir (URL-1; Sugawara ve diğerleri, 2019; Chang ve diğerleri, 2024).

Literatürdeki bu çalışmalar, odaklandıkları problemlere göre farklı başarı ölçütleri kullanmaktadır. Çizelge 1.1'de; değinilen güncel çalışmaların kullandıkları yöntemler, hedefledikleri donanım platformları ve performanslarını ölçümledikleri temel kalite ve donanım metrikleri özetlenmiştir.

Çizelge 1.1. Literatürdeki güncel FPGA/Donanım tabanlı görüntü işleme çalışmalarının performans metrikleri

Çalışma	Yöntem ve Hedef Platform	Performans Kalite Metrikleri	Donanım Kalite Metrikleri
Spagnolo ve diğerleri (2023)	FSRCNN, XC7K410T	$PSNR_{max@Set5} = 39{,}03\ dB$ $SSIM_{max@Set5} = 0{,}9631$	İşlem $Hacmi = 753{,}04\ Mp/s$ Güç $Tüketimi = 3{,}7\ W$
Shi ve diğerleri (2023)	FSRCNN, ZCU102	$PSNR_{max@Set5} = 36{,}40\ dB$ $SSIM_{max@Set5} = 0{,}9784$	İşlem $Hacmi$ $= 4K@120{,}1fps, 8K@30{,}0fps$
Su ve diğerleri (2024)	ResBinESPCN, Z7P	$PSNR_{ort.@Set5} = 29{,}82\ dB$	$\frac{BOPs}{MACs} = 36{,}38 * 10^9$ Güç $Tüketimi = 4{,}452\ W$
Amin ve Obermaisser (2025)	BiDSRS+, ZCU102	$PSNR_{ort.@Set5} = 34{,}03\ dB$ $SSIM_{ort.@Set5} = 0{,}9560$	İşlem $Hacmi = 4K@60fps$ Güç $Tüketimi = 1{,}6\ W$
Joseph ve Jayakumar (2024)	FSRCNN, ZCU104	$PSNR_{ort.@Set5} = 36{,}92\ dB$ $SSIM_{ort.@Set5} = 0.94$	İşlem $Hacmi = 4K@24{,}12fps$
URL-3	FPGA odaklı AGC, PYNQ-Z1	$PSNR_{max} = 26{,}48\ dB$ $MS - SSIM_{max} = 0{,}894$ $LPIPS_{min} = 0{,}066$	İşlem $Hacmi$ $= 256x256@2{,}38fps$ Güç $Tüketimi = 1{,}83\ W$
Yanamala ve Pullakandam (2025)	Hibrit CNN-LSTM, ZCU104	Doğruluk Oranı: %77,76	İşlem $Hacmi = 6{,}22\ GOP/s$ Güç $Tüketimi = 0{,}777\ W$

Çizelge 1.1 incelendiğinde, görüntü kalitesini ölçmek için standart olarak PSNR ve SSIM kullanıldığı görülmektedir. PSNR piksel bazlı sadakati ölçerken, SSIM algısal benzerliğe odaklanmaktadır. URL-3'te yapılan çalışmada, üretken modellerin başarısını ölçmek için insan algısıyla daha uyumlu olan LPIPS metriğini kullanmıştır. Donanım performansı tarafında ise sistemin işlem hacmi ve güç tüketimi temel kıyaslama ölçütleridir. Tablodaki veriler, süper çözünürlük probleminin çözümünde yalnızca görsel kalitenin değil, donanım kısıtları altında optimum dengenin sağlanmasının kritik olduğunu göstermektedir.

Araştırmanın amacı

Bu çalışmanın temel amacı, FSRCNN mimarisinin modern Vitis Vision kütüphanesi kullanılarak FPGA üzerinde yeniden yapılandırılması, büyütme aşamasında gözlemlenen dama tahtası bozulmasının yeniden boyutlandırmalı konvolüsyon yöntemiyle giderilmesi ve renk işleme sürecinin YCbCr (Parlaklık, Mavi Renk Kanalının Parlaklıktan Farkı, Kırmızı Renk Kanalının Parlaklıktan Farkı) uzayına taşınarak hesaplama verimliliğinin artırılmasıdır. Bu kapsamda çalışma, FSRCNN bloklarının Vitis Vision fonksiyonları ile yeniden modellenmesini, karmaşık konvolüsyon işlemlerinin donanım açısından optimize edilmesini, yalnızca parlaklık bileşeninin CNN ile işlenmesine yönelik bir veri akışı mimarisi oluşturulmasını, tasarımın Vitis HLS ortamında doğrulanmasını ve elde edilen sonuçların xfOpenCV tabanlı referans model ile karşılaştırılmasını amaçlamaktadır. Böylece hem hesaplama verimliliği hem de algısal kalite bakımından daha kararlı bir mimari ortaya konulması hedeflenmektedir.

Araştırmanın önemi

Bu çalışma, FSRCNN mimarisinin Vitis Vision kütüphanesi ile yeniden uygulanmasına yönelik literatürdeki eksikliği gidermekte ve güncel FPGA araçları ile uyumlu bir süper çözünürlük tasarımı sunmaktadır. Özellikle ters konvolüsyonun neden olduğu bozulmaların donanım bağlamında giderilmesi ve büyütme işleminin yeniden boyutlandırma temelli bir mimariyle ele alınması, hem literatüre kuramsal bir katkı sağlamakta hem de gömülü görüntü işleme uygulamalarına yönelik pratik bir çözüm üretmektedir. Çalışmanın sonuçları, süper çözünürlük modellerinin modern FPGA mimarilerine uyarlanmasında izlenebilecek yöntemlere ilişkin somut veriler sunarak akademik ve endüstriyel uygulamalar için referans niteliği taşımaktadır.

Varsayımlar / sayıltılar

Bu araştırmada FSRCNN modelinin ImageNet veri kümesi üzerinde eğitilmiş ağırlık ve bias parametrelerinin doğal sahneler üzerinde yeterli genelleştirme başarısı sunduğu varsayılmıştır. Ayrıca Vitis Vision kütüphanesinde yer alan fonksiyonların AMD tarafından sağlanan resmî belgelerde tanımlanan işlevsel doğruluğa uygun şekilde çalıştığı kabul edilmiştir. XSIM tabanlı C/RTL kosimülasyon çıktılarının donanım davranışını temsil ettiği varsayılmış; insan görme sisteminin parlaklık bileşenine olan duyarlılığı nedeniyle süper çözünürlük işleminin yalnızca Y kanalında gerçekleştirilmesinin algısal kalite açısından yeterli olduğu kabul edilmiştir.

Sınırlılıklar

Bu çalışma, yalnızca 2× büyütme (upscaling) faktörüne yönelik olarak optimize edilmiştir; daha yüksek ölçekler için ağın son katmanlarının yeniden eğitilmesi ve tasarımın donanıma tekrar uyarlanması gerekmektedir. Bunun yanı sıra gerçek zamanlı sistem entegrasyonu ile donanım yerleştirme ve yönlendirme (place & route) süreçleri araştırma kapsamı dışında tutulmuştur.

Tanımlar

Süper çözünürlük, düşük çözünürlüklü görüntülerden yüksek çözünürlüklü görüntülerin elde edilmesini amaçlayan bir görüntü işleme problemidir. Tek görüntü süper çözünürlük, yalnızca bir giriş görüntüsünden daha yüksek çözünürlüklü bir çıktı üretir. Evrişimli Sinir Ağları (CNN), görüntülerdeki yerel örüntüleri modelleyen derin öğrenme yapılarıdır. SRCNN, büyütülmüş görüntü üzerinde CNN işlemleri uygulayarak iyileştirme sağlar. FSRCNN, bu yapıyı düşük çözünürlüklü uzayda çalışacak biçimde genişleten daha hızlı bir modeldir. Ters konvolüsyon, görüntüyü genişletmek için kullanılan bir işlem olup düzenli bozulmalara yol açabilir. Yeniden boyutlandırmalı konvolüsyon yöntemi, bu bozulmaları azaltmak amacıyla büyütme işlemini iki aşamalı olarak gerçekleştirir. AXI-Stream, FPGA tabanlı sistemlerde yaygın olarak kullanılan bir veri akışı protokolüdür. Pipeline, dataflow ve döngü açma (unroll) direktifleri, HLS tasarımlarında paralelliği artırmak ve gecikmeyi azaltmak amacıyla kullanılan optimizasyon teknikleridir.

2. ÇÖZÜNÜRLÜK İYİLEŞTİRME YÖNTEMLERİ

Bu bölümde modern çözünürlük iyileştirme yöntemleri, teorik temelleri ve matematiksel modelleri çerçevesinde ele alınmaktadır. Literatürde çözünürlük artırma yöntemleri temelde iki kategori altında incelenmektedir: (i) klasik enterpolasyon tabanlı yöntemler ve (ii) derin öğrenme tabanlı yöntemler. Bu kapsamda bikübik enterpolasyon yöntemi referans bir temel yaklaşım olarak sunulmakta; ardından derin öğrenme tabanlı modeller olan SRCNN, FSRCNN ve ESPCN mimarileri matematiksel yönleriyle ayrıntılı biçimde açıklanmaktadır.

2.1. Bikübik Enterpolasyon

Bikübik enterpolasyon, hedef piksel değerini tahmin etmek için iki boyutlu ayrılabilir bir model kullanır. Yöntem, 4×4 komşuluk penceresindeki 16 pikselin ağırlıklı toplamı üzerinden çalışır. Genel enterpolasyon modeli aşağıdaki gibi ifade edilir (Eş. 2.1.):

$$I'(x,y) = \sum_{m=-1}^{2} \sum_{n=-1}^{2} I(i+m, j+n) * w(m,n) \quad \text{(Eş. 2.1.)}$$

Burada taban koordinat $(i,j) = (\lfloor x \rfloor, \lfloor y \rfloor)$ olup kullanılan çekirdek fonksiyonu iki ayrı bir boyutlu kernelin çarpımından oluşur. Bu ayrılabilir ağırlık yapısı Eş. 2.2 ile tanımlanır:

$$w(m,n) = k\big(x - (i+m)\big) * k\big(y - (j+n)\big) \quad \text{(Eş. 2.2.)}$$

Katsayı fonksiyonu Catmull–Rom çekirdeği olup aşağıdaki gibi Eş. 2.3 ile tanımlanır:

$$k(t) = \begin{cases} (a+2) * |t|^3 - (a+3) * |t|^2 + 1, & |t| < 1 \\ a * |t|^3 - 5 * a * |t|^2 + 8 * a * |t| - 4 * a, & 1 \le t < 2 \\ 0, & 2 \le t \end{cases} \quad \text{(Eş. 2.3.)}$$

Bu eşitlikler (Eş. 2.1, Eş. 2.2 ve Eş. 2.3), bikübik enterpolasyon modelinin temel matematiksel yapısını oluşturmaktadır. Bikübik fonksiyonlar uzamsal süreklilik sağlar; ancak yüksek frekans bileşenlerini yeniden oluşturamadığı için özellikle kenar bölgelerde ayrıntı kaybı oluşturur. Bu yönüyle FPGA uygulamalarında boru hattı (pipeline) açısından

uyumlu olsa da, hesaplama 16 komşu piksele bağlı olduğundan bellek erişim maliyetini artırır ve çözünürlük iyileştirme performansı derin öğrenme tabanlı modellere kıyasla sınırlıdır.

2.2. Süper Çözünürlüklü Evrişimli Sinir Ağı (SRCNN)

SRCNN modeli, düşük çözünürlüklü görüntüyü önce büyütmek için bikübik enterpolasyon kullanmakta, ardından üç ardışık konvolüsyon işlemi ile iyileştirme gerçekleştirmektedir. İlk işlem olan öznitelik çıkarımı Eş. 2.4 ile gerçekleştirilir:

$$F_1 = \sigma(W_1 * I_{HR} + b_1) \quad \text{(Eş. 2.4.)}$$

Bu aşama, yüksek çözünürlüklü uzaydaki düşük seviyeli dokuların çıkarılmasını sağlar. Ardından çizgisel olmayan haritalama işlemi Eş. 2.5 uygulanır:

$$F_2 = \sigma(W_2 * F_1 + b_2) \quad \text{(Eş. 2.5.)}$$

Bu dönüşüm, giriş özelliklerini daha yüksek çözünürlük için anlamlı bir gösterim uzayına taşır. Son aşama (Eş. 2.6.), yeniden yapılandırma işlemini gerçekleştirir:

$$I_{SR} = W_3 * F_2 + b_3 \quad \text{(Eş. 2.6.)}$$

Bu üç işlem; bir başka deyişle öznitelik çıkarımı, çizgisel olmayan haritalama ve yeniden yapılandırma işlemleri; SRCNN'in hesaplama yapısını belirler. Model büyütmeyi işlem öncesinde bikübik enterpolasyonla (Eş. 2.1, Eş. 2.2 ve Eş. 2.3) tamamladığı için tüm konvolüsyonlar büyütülmüş görsel uzayında yapılır. Bu durum FPGA üzerinde paralel hesaplama gereksinimini artırır, DSP tüketimini yükseltir ve maliyeti ölçek faktörünün karesiyle büyütür. Bununla birlikte, modelin katman yapısının sade olması HLS arayüzleriyle uyumlu şekilde sentezlenmesini kolaylaştırır; dolayısıyla teorik olarak uygulanabilir bir yöntemdir ancak kaynak kullanım verimliliği düşüktür.

2.3. Hızlı Süper Çözünürlüklü Evrişimli Sinir Ağı (FSRCNN)

FSRCNN modeli, hesaplamayı düşük çözünürlük uzayına taşıyarak SRCNN'nin yüksek hesaplama maliyetini azaltmayı amaçlayan bir "kum saati" mimarisi sunar (Dong ve diğerleri, 2016). İlk işlem öznitelik çıkarımıdır ve Eş. 2.7 ile ifade edilir:

$$F_f = \sigma\left(W_f * I_{LR} + b_f\right) \quad \text{(Eş. 2.7.)}$$

Çıkarımı yapılan öznitelikler daha sonra daraltma katmanında (Eş. 2.8.) daha düşük kanal boyutuna indirgenir:

$$F_s = \sigma\left(W_s * F_f + b_s\right) \quad \text{(Eş. 2.8.)}$$

Bu işlem, FSRCNN'nin hesaplama yükünü yönetilebilir kılmasının temel mekanizmasıdır. Ardışık haritalama katmanları (Eş. 2.9.), düşük çözünürlük uzayında konvolüsyon işlemlerinin dört katman boyunca yoğun biçimde uygulanmasını sağlar:

$$F_m^{(i+1)} = \sigma\left(W_m^{(i)} * F_m^{(i)} + b_m^{(i)}\right) \quad \text{(Eş. 2.9.)}$$

Genişletme katmanı (Eş. 2.10.), kanal derinliğini ters konvolüsyona uygun hâle getirir:

$$F_e = \sigma\left(W_e * F_m^{(M)} + b_e\right) \quad \text{(Eş. 2.10.)}$$

Modelin çözünürlük iyileştirme ve görsel büyütme işlemi, ters konvolüsyon katmanıyla (Eş. 2.11.) gerçekleştirilir. Eş. 2.11'de geçen "$\star$" işlemi, ters konvolüsyonu ifade etmektedir.

$$I_{SR} = W_d \star F_e + b_d \quad \text{(Eş. 2.11.)}$$

Bu eşitlikler (Eş. 2.7, Eş. 2.8, Eş. 2.9, Eş. 2.10 ve Eş. 2.11), FSRCNN'nin düşük çözünürlüklü hesaplama yapısını ayrıntılı şekilde tanımlar. Bu mimarinin FPGA açısından en belirgin avantajı, tüm konvolüsyon işlemlerinin LR uzayında yapılmasıdır; bu durum DSP kullanımını ciddi ölçüde azaltır ve boru hattı gecikmesini en aza indirir. Kanal

daraltma–genişletme yapısı bellek bant genişliği kullanımını düşürerek yüksek paralellik sağlar. Ancak ters konvolüsyon adımı (Eş. 2.11.), uygun filtre-adım ilişkisi sağlanmazsa düzenli örüntülere neden olabilir; bu nedenle literatürde yeniden boyutlandırmalı konvolüsyon yönteminin değerlendirildiği çalışmalar da mevcuttur. Ters konvolüsyon yerine yeniden boyutlandırmalı konvolüsyon yönteminin benimsenmesi, FSRCNN'i FPGA açısından son derece verimli ve esnek bir modele dönüştürmektedir (URL-1; Sugawara ve diğerleri, 2019).

2.4. Verimli Alt-Piksel Evrişimli Sinir Ağı (ESPCN)

ESPCN, çözünürlüğü artırma işlemini alt piksel yeniden düzenleme algoritmasıyla (pixel shuffle) gerçekleştiren, düşük çözünürlük uzayında çalışan bir modeldir. Alt piksel dönüşümü şu şekilde tanımlanır:

$$I_{SR}(x, y, c) = T\left(F\left(\frac{x}{r}, \frac{y}{r}, c'\right)\right), \qquad c' = c * r^2 \qquad \text{(Eş. 2.12.)}$$

Bu eşitlik (Eş. 2.12.), büyütme işlemi öncesi kanal sayısının r^2 katsayısıyla arttığını göstermektedir. Bu yönüyle kanal genişliğinin doğrudan büyümesi, FPGA üzerinde hem DSP kullanımını hem de BRAM erişimini artırır. Alt piksel yeniden düzenleme yöntemi hesaplama açısından hafif olsa da, bellek düzeni üzerinde yoğun bir yeniden yapılandırma gerektirdiğinden donanım akışını karmaşıklaştırır. Bu nedenle ESPCN genel olarak yazılım tabanlı uygulamalarda verimli olsa da, donanım hızlandırmalı ortamlarda veri akışı sürekliliğini bozduğu için FSRCNN kadar avantajlı değildir.

3. TASARIM ORTAMLARI

Bu bölümde önerilen FSRCNN tabanlı çözünürlük iyileştirme mimarisinin FPGA üzerinde gerçekleştirilmesine olanak sağlayan yazılım ve donanım araçları açıklanmaktadır. Tasarım sürecinde AMD/Xilinx'in modern geliştirme ekosistemi olan Vitis ortamı, Vitis Vision kütüphanesi, OpenCV uyarlama süreçleri ve C/RTL kosimülasyon için kullanılan Vivado XSIM aracı değerlendirilmiştir. Her bir araç, tasarımın doğrulanması, optimize edilmesi ve donanım davranışı ile tutarlılığının sağlanması açısından kritik rol oynamaktadır.

3.1. Vitis ve Kütüphaneleri

AMD/Xilinx tarafından sunulan Vitis geliştirme ortamı, Vivado HLS'nin önceki nesil derleme modelini önemli ölçüde genişleterek donanım hızlandırıcılarının modern FPGA mimarileri üzerinde daha verimli şekilde gerçekleştirilmesini amaçlayan kapsamlı bir yazılım ekosistemi sunmaktadır. Vivado döneminde kullanılan xfOpenCV kütüphanesi, OpenCV fonksiyonlarının donanım uyumlu karşılıklarını sağlamayı hedefleyen ilk girişimlerden biri olmakla birlikte, yapısal sınırlılıkları nedeniyle zaman içerisinde modern görüntü işleme uygulamalarının gereksinimlerini karşılayamaz hâle gelmiştir. xfOpenCV'nin temel problemi, fonksiyonlarının çoğunun CPU'da karşılığı bulunan OpenCV API'sinin doğrudan donanım eşlemesi mantığıyla tasarlanmış olmasıdır. Bu durum, kütüphanenin karmaşık veri akışları gerektiren modern görüntü işleme boru hatlarında esnek biçimde kullanılmasını zorlaştırmış; özellikle yüksek çözünürlüklü görüntülerin işlenmesinde bellek yönetimi, ardışık işleme ve paralel veri aktarımı gibi kritik gereksinimleri karşılamakta yetersiz kalmasına yol açmıştır. Ayrıca xfOpenCV, modüler olarak genişletilemeyen ve OpenCV'nin güncel sürümleriyle sınırlı etkileşim kurabilen bir tasarım felsefesine sahip olduğundan, AMD'nin geliştirme yol haritası için uzun vadede sürdürülebilir bir çözüm olmaktan çıkmıştır.

Bu bağlamda Vitis Vision kütüphanesi, yalnızca xfOpenCV'nin yerini alan bir araç değil, FPGA üzerinde görüntü işleme uygulamalarının gerektirdiği veri akışı odaklı paradigmayı temel alan yeni bir tasarım anlayışını temsil etmektedir. Vitis Vision, hesaplamaların CPU-benzeri matris işlemleriyle değil, FPGA mimarisinin doğasına uygun olan satır-satır akan

veri yapılarıyla modellenmesini sağlar. Kütüphane, dahili olarak line buffer ve window buffer gibi donanım için optimize edilmiş tamponlama mekanizmalarını kullanarak yüksek çözünürlüklü görüntülerin ardışık ve düşük gecikmeli biçimde işlenmesini mümkün kılar. Özellikle konvolüsyon tabanlı işlemlerin uygulanmasında, piksel akışının AXI4-Stream protokolüyle uyumlu olarak işlenmesi, yüksek paralellik derecelerinin elde edilebilmesini sağlayan önemli bir özellik olarak ortaya çıkmaktadır. Vitis Vision fonksiyonları, FPGA üzerinde donanım sentezi sırasında kaynak kullanımını ve yürütme gecikmesini optimize edecek biçimde tasarlanmış çekirdek modüller içerir; bu nedenle konvolüsyon, renk dönüşümü ve yeniden boyutlandırma gibi temel işlemler, hem düşük kaynak tüketimiyle hem de yüksek veri verimliliğiyle gerçekleştirilebilir.

xfOpenCV'nin aksine Vitis Vision, OpenCV'nin API yapısını taklit etmeye çalışmak yerine, doğrudan FPGA veri akışı mimarisine uygun yüksek verimli çekirdekler sunan bir kütüphane olarak tasarlanmıştır. Bu yaklaşım, hem tasarımcıya daha öngörülebilir bir sentez davranışı sağlar hem de pipeline, unroll ve dataflow optimizasyonları gibi HLS direktiflerinin daha etkin biçimde uygulanmasına imkân verir. Böylece, FSRCNN gibi derin öğrenme tabanlı görüntü iyileştirme modellerinde katmanlar arası veri akışının kesintisiz biçimde işlenmesi sağlanarak donanım performansı önemli ölçüde artırılabilmektedir. Bu nedenle, bu çalışma kapsamında önerilen mimarinin geliştirilmesinde xfOpenCV yerine Vitis Vision'ın tercih edilmesi, hem modern AMD araç zinciri ile uyumluluk hem de donanım kaynaklarının etkin kullanımı açısından zorunlu bir tercih hâline gelmiştir.

3.2. OpenCV Kütüphaneleri ve Adaptasyonu

Bu çalışmada HLS tasarımının doğrulanması ve FPGA üzerinde gerçeklenecek FSRCNN mimarisinin yazılım modeliyle tutarlı çıktılar ürettiğinin test edilmesi için OpenCV kütüphanesinden yararlanılmıştır. FSRCNN'nin FPGA üzerinde gerçeklenmesi doğrudan OpenCV fonksiyonları gerektirmese de, yazılım tabanlı referans modelin oluşturulması ve C-simülasyon aşamasında giriş/çıkış görüntülerinin işlenebilmesi için OpenCV'nin temel görüntü işleme fonksiyonlarına ihtiyaç duyulmuştur. Özellikle görüntü okuma, renk uzayı dönüşümü, yeniden boyutlandırma ve referans karşılaştırma işlemleri yalnızca yazılım ortamında yapılabilir olduğundan, bu fonksiyonlar FPGA sentez sürecinde değil yalnızca testbench tarafında kullanılmıştır. Bu kapsamda kullanılan fonksiyonlar Çizelge 3.1'de

açıklanmıştır.

Çizelge 3.1. FSRCNN testbench aşamasında kullanılan başlıca OpenCV fonksiyonları

Fonksiyon	Açıklama	Kullanım Amacı
`cv::imread()`	Diskten görüntü okur	Test görüntülerinin C-simülasyon aşamasında yüklenmesi
`cv::imwrite()`	Görüntüyü dosyaya kaydeder	FPGA çıktısının kaydedilerek görsel doğrulama yapılması
`cv::cvtColor()`	Renk uzayı dönüşümü yapar (BGR ↔ YCbCr)	Y kanalının ayrılması ve FPGA'ya yalnızca luma bilgisinin aktarılması
`cv::resize()`	Görüntüyü yeniden boyutlandırır	Referans sonuçların üretilmesi ve değerlendirme metriklerinin hesaplanması
`cv::absdiff()`	Piksel başına fark hesaplar	Donanım çıktısının yazılım referansıyla karşılaştırılması
`cv::Mat` yapısı	CPU tabanlı görüntü temsilidir	C-simülasyon aşamasında görüntü manipülasyonu

OpenCV'nin bu çalışmadaki rolü, donanım tasarımının doğruluk kontrolünde bir referans mekanizması oluşturmakla sınırlıdır. Vitis HLS tarafından gerçekleştirilen donanım sentezi OpenCV fonksiyonlarını işleyemeyeceğinden, bu fonksiyonların tümü koşullu derleme yapısı kullanılarak sentez aşamasının dışında tutulmuştur. Bu yaklaşım literatürde yaygın olup Xilinx/AMD tarafından da önerilen yöntemdir; OpenCV yalnızca C-simülasyon aşamasında etkin olur ve donanıma aktarılmayan bir yazılım katmanı görevi üstlenir.

FSRCNN tasarımının geliştirilmesi sürecinde OpenCV'nin kullanılabilmesi için öncelikle AMD tarafından sağlanan rehber niteliğindeki iki dokümanda tariflenen yapılandırma adımlarının uygulanması gerekmiştir (URL-4, URL-5). Bu iki doküman, OpenCV'nin Windows üzerinde MinGW tabanlı bir derleyici kullanılarak kaynak koddan derlenmesi, kütüphane dosyalarının doğru dizinlere yerleştirilmesi, çevresel değişkenlerin güncellenmesi ve son olarak derlenen OpenCV kütüphanelerinin Vitis HLS ile uyumlu hâle getirilmesi için gereken temel işlemleri tanımlamaktadır. Bu süreç, özellikle OpenCV'nin 4.4.0 sürümü gibi varsayılan Vitis kurulumunda bulunmayan fonksiyon setlerinin kullanılabilmesini sağlar.

Dokümanlarda önerilen CMake yapılandırma adımları ve MinGW derleme işlemleri, Windows ortamında OpenCV'nin Vitis tarafından erişilebilir bir kütüphane hâline getirilmesi için zorunludur. Vitis Vision projeleri OpenCV'yi yalnızca C-simülasyon aşamasında çağırabildiğinden, OpenCV'nin doğru biçimde derlenmesi ve proje dosyasına include/library yollarının eklenmesi gerekmektedir. AMD'nin örnek projelerde gösterdiği bu yapılandırma akışı, testbench'in cv::Mat üzerinden görüntüyü okuması, AXI-Stream formatına dönüştürmesi ve Vitis Vision fonksiyonlarıyla birlikte çalışması için gerekli altyapıyı sağlamaktadır. OpenCV'nin doğru kurulumu yapılmadan, C/RTL kosimülasyon aşamasında yazılım–donanım karşılaştırması gerçekleştirilemez.

Özetle, OpenCV'nin kaynak koddan derlenmesi, MinGW tabanlı derleyiciye uygun hâle getirilmesi ve Vitis Vision'ın örnek uygulamalarıyla bütünleştirilmesi; FSRCNN tasarımında kullanılan testbench fonksiyonlarının güvenilir şekilde çalışmasını sağlayan temel altyapıyı oluşturmuştur. Bu yapılandırma sayesinde FSRCNN mimarisinin yazılım ve donanım modellerinin çıktıları birbiriyle karşılaştırılabilmiştir.

3.3. Vivado XSIM Simülasyon Ortamı

Vivado XSIM, Xilinx tabanlı yüksek seviyeli sentez (HLS) akışının temel doğrulama bileşenlerinden biri olup, yazılımla tanımlanan algoritmik model ile sentez sonrasında üretilen RTL tabanlı donanım mimarisi arasındaki işlevsel tutarlılığın sınanmasını sağlayan bütüncül bir simülasyon altyapısı sunmaktadır. XSIM bu yönüyle yalnızca bir fonksiyonel doğrulama aracı değil, aynı zamanda tasarımın mikro-mimari davranışlarını görünür kılan analitik bir inceleme ortamı olarak değerlendirilebilir. Özellikle derin konvolüsyon zincirleri barındıran FSRCNN gibi modellerde, zamanlama ilişkilerinin, veri akışı örüntülerinin ve hesaplama ardışıklığının dalga biçimi üzerinde ayrıştırılabilir hâle gelmesi; tasarım optimizasyonlarının etkisini nicel ve yapısal düzeyde inceleme olanağı sağlamaktadır.

Pragma direktiflerinin tasarım üzerindeki etkileri XSIM ortamında özellikle belirginleşmektedir. Pragma direktifi kullanılmayan bir HLS tasarımı, Vivado sentezleyicisinin doğal olarak ürettiği düşük paralellik düzeyine sahip, ardışıl yürütme ağırlıklı bir RTL yapısına karşılık gelir. Döngü iterasyonlarının birbirini tamamlama esasına dayalı bu yapı, özellikle derin CNN katmanlarında çok uzun yürütme

gecikmelerine yol açmakta; XSIM üzerinde yapılan simülasyonun tamamlanmasını güçleştirmektedir. Nitekim AMIQ Consulting tarafından yayımlanan referans FSRCNN uygulamasında, pragma direktifi kullanımı zayıf tasarımın bir günlük süre içerisinde dahi C/RTL kosimülasyonu tamamlayamaması, bu mimari sınırlılığın pratik bir örneğidir (URL-2).

Buna karşılık, döngü ardışıklaması (pipeline), döngü açma (unroll), veri akışı yönetimi (dataflow) ve bellek bölütleme (array partitioning) gibi pragma direktiflerinin sistematik biçimde uygulanması, HLS sentezleyicisinin ürettiği RTL yapısında belirgin paralellik düzeyleri oluşturur. Pipeline direktifi, döngü gövdelerinin örtüşmeli şekilde yürütülmesini sağlayarak başlatma aralığını (II) düşürür; unroll ise belirli döngülerin gövdelerini donanım üzerinde çoğaltarak paralel işlem birimlerinin etkin hâle gelmesini sağlar. Dataflow direktifi katman bazlı görev paralelliği oluştururken, array partitioning ise bellek bant genişliğini artırarak eş zamanlı ağırlık/aktivasyon erişimini mümkün kılar. Bu dönüşümler, XSIM dalga biçimlerinde aynı saat çevriminde birden fazla hesaplama aşamasının yer almasıyla gözlemlenebilir; bu da tasarımın ardışık-paralel bir hesaplama modeline taşındığını gösterir.

Bu bulgular; Sestito ve diğerlerinin (2023) FPGA üzerinde ters konvolüsyon hızlandırıcılarının HLS ile gerçekleştirilmesine yönelik analizleriyle de örtüşmektedir. Söz konusu çalışmada pragma direktifi temelli optimizasyonların, hesaplama zincirinin farklı aşamalarını paralel yürütmeye olanak vererek gecikmeyi azaltmakla kalmadığı, aynı zamanda sistemin işlem hacmini (throughput) anlamlı şekilde yükselttiği gösterilmiştir (Sestito ve diğerleri, 2023). Aynı çalışma, paralelleştirmenin donanım kaynak tüketimini artırdığını; özellikle LUT, DSP ve BRAM kullanımında belirgin artışlara yol açtığını da ortaya koymaktadır. Bu ikili etki, bir başka deyişle gecikme azaltımı ile alan maliyeti artışı arasındaki denge, XSIM üzerinde gözlemlenebilir hâle gelmektedir. Pipeline ve unroll uygulandığında kaynakların yoğun biçimde çoğaldığı; buna karşılık simülasyonun belirgin biçimde hızlandığı dalga biçimlerinden izlenebilmektedir.

Dolayısıyla, XSIM yalnızca fonksiyonel doğrulama sağlayan bir simülasyon aracı değil; pragma direktifi kullanımının tasarımın zamanlama davranışı, veri akış düzeni ve kaynak paylaşım stratejileri üzerindeki etkilerini görünür kılan metodolojik bir inceleme ortamıdır. Bu çalışma kapsamında geliştirilen FSRCNN hızlandırıcısında da pragma direktifi

kullanımı zayıf tasarımın yüksek gecikme karakteristiği ile optimize edilmiş tasarımın ardışık-paralel hesaplama modeli arasındaki fark, XSIM aracılığıyla açık biçimde ortaya konmuştur. Bu nedenle XSIM, FPGA üzerinde derin öğrenme tabanlı görüntü işleme mimarilerinin değerlendirilmesinde yalnızca doğrulama aşamasının bir parçası değil, tasarım odaklı çıkarımlar için de merkezi bir analitik araç niteliği taşımaktadır.

4. HLS TASARIMI VE DENEYSEL SONUÇLAR

Bu bölümde çözünürlük iyileştirme yöntemlerinin FPGA üzerinde donanım hızlandırıcısı olarak gerçekleştirilebilmesine yönelik yüksek seviyeli sentez (HLS) tasarım süreci ayrıntılı biçimde ele alınmaktadır. Tasarım yaklaşımı, görüntü işleme adımlarını akış tabanlı bir mimariye dönüştürmeyi, model parametrelerini donanıma uygun yapılarla temsil etmeyi ve hesaplama yoğun katmanları Vitis Vision kütüphanesi aracılığıyla donanımsal olarak yeniden inşa etmeyi hedeflemektedir. Süreç boyunca hesaplama işlemleri, veri akışı ilkeleri doğrultusunda düzenlenmiş ve tüm katmanlar FPGA'nın paralel işlem altyapısıyla uyumlu bir boru hattı hâline getirilmiştir.

Bu amaçla yapılan tüm sentez, doğrulama ve kosimülasyon adımları AMD Zynq UltraScale+ MPSoC XCZU7EV-2FBVB900I hedef platformunda gerçekleştirilmiş; işlem hattının modellenmesi ve doğrulanması ise Intel Core i5-5200U CPU, 2,20 GHz ve 8 GB RAM'e sahip bir iş istasyonunda Vitis 2025.1 geliştirme ortamı kullanılarak yürütülmüştür. Böylece çözünürlük iyileştirme sürecini oluşturan tüm hesaplama adımları, FPGA mimarisinin doğal paralellik ilkeleri ile uyumlu bir doğruluk ve performans çerçevesi içinde yeniden yapılandırılmıştır.

4.1. Yöntem Seçimi ve Model Parametrelerinin Belirlenmesi

Bu çalışmada çözünürlük iyileştirme amacıyla kullanılacak yöntemin belirlenmesinde hem algoritmik başarı hem de donanım üzerinde uygulanabilirlik ölçütleri birlikte değerlendirilmiştir. Literatürde yer alan ve bu çalışmanın 2. bölümünde detaylandırılmış olan çözünürlük iyileştirme yöntemleri, FPGA üzerinde kaynak tüketimi, gecikme ve veri akışı düzeni bakımından farklı performans özellikleri sergilemektedir. Bu yöntemler arasında FSRCNN, mimari yapısının FPGA hızlandırma mantığıyla doğal bir uyum göstermesi nedeniyle tercih edilmiştir. Modelin hesaplamalarının düşük çözünürlüklü uzayda yürütülmesi, konvolüsyon işlemlerinin maliyetini önemli ölçüde azaltmakta; daraltma katmanı sayesinde haritalama aşamasındaki kanal sayısı düşürülerek ardışık konvolüsyon zincirinin donanım üzerinde daha verimli çalışması sağlanmakta; 3x3 filtrelerden oluşan haritalama katmanları FPGA'de satır tamponlama ve pencere kaydırma mekanizmalarıyla uyumlu bir işlevsellik sunmakta; genişletme katmanındaki 1x1

konvolüsyon hesaplamaları ise DSP blokları üzerinde düşük kaynak kullanımıyla yürütülebilmektedir. Mimarisinin sade ve düzenli olması, yüksek seviyeli sentez (HLS) araçlarının pipeline ve veri akışı (dataflow) optimizasyonlarını daha etkili uygulamasına da katkı sağlamaktadır. Bu nedenlerle FSRCNN, hem hesaplama verimliliği hem de donanımsal uygulanabilirlik açısından bu çalışma için uygun bir yöntem olarak değerlendirilmiştir.

Model mimarisinin FPGA üzerinde doğru biçimde yeniden oluşturulabilmesi için ağın tüm parametrelerinin; bir başka deyişle konvolüsyon ağırlıkları, bias değerleri ve PReLU katsayıları; özgün hâlleriyle tasarıma aktarılması gerekmektedir. Bu çalışma kapsamında kullanılan parametreler, Dong ve diğerleri (2016) tarafından önerilen FSRCNN modelinin eğitimli ağırlıklarıdır ve herhangi bir yeniden eğitim yapılmaksızın doğrudan donanım tasarımına dahil edilmiştir. Parametrelerin tümü wrapper.hpp isimli bir dosyada sabit diziler şeklinde tanımlanmıştır. Bu sayede ağırlıkların FPGA üzerinde ROM benzeri yapılarda depolanması ve konvolüsyon çekirdekleri tarafından düşük gecikmeyle erişilmesi sağlanmıştır. Ayrıca sabit katsayıların HLS sentezleyicisi tarafından DSP bloklarına optimize biçimde dağıtılması, hem hesaplama doğruluğunun korunmasına hem de filtreleme işlemlerinin verimli yürütülmesine katkıda bulunmuştur. PReLU katsayılarının FPGA üzerinde uygulanması ise yalnızca sabit çarpım ve maksimum-minimum tabanlı bir işlem gerektirdiğinden kontrol mantığına ek bir yük getirmemiştir.

4.2. Vitis Vision Kütüphanesi ile Modernizasyon

Bu çalışmada gerçekleştirilen çözünürlük iyileştirme tasarımının donanım üzerinde verimli biçimde çalışabilmesi için, geleneksel xfOpenCV tabanlı yaklaşımların modern Vitis Vision kütüphanesiyle yeniden yapılandırılması gerekli hâle gelmiştir. xfOpenCV, uzun yıllar Vivado HLS projelerinde standart görüntü işleme kütüphanesi olarak kullanılmış olsa da, hem bakımının durdurulmuş olması hem de çağdaş FPGA mimarilerinin sunduğu paralelleştirme olanaklarını tam olarak kullanamaması nedeniyle artık güncel tasarım ihtiyaçlarına yeterli yanıt verememektedir. Bu nedenle tasarımda yer alan tüm temel görüntü işleme adımları, AMIQ Consulting tarafından yayımlanan FSRCNN uygulamasında da görüldüğü üzere (URL-2), aynı işlevleri daha optimize ve donanım odaklı biçimde sunan Vitis Vision fonksiyonlarıyla yeniden düzenlenmiştir.

Bu dönüşüm sürecinin temel motivasyonu, Vitis Vision kütüphanesinin HLS akışına doğrudan entegre olabilen sınıf yapıları, optimize edilmiş pencereleme ve satır tamponlama mekanizmaları, veri akışını kesintisiz sürdürebilen boru hattı uyumlu fonksiyonları ve geniş kapsamlı donanım optimizasyonlarıdır. Özellikle satır tamponlamayı yöneten xf::cv::LineBuffer, pencere işlemeyi sağlayan xf::cv::Window, piksellerin doğru konumda yerleştirilmesini mümkün kılan insert_pixel() gibi metodlar ve tamamen optimize edilmiş konvolüsyon mantığı, FSRCNN tasarımının gerektirdiği yüksek yoğunluklu ardışık hesaplamayı çok daha düşük bir gecikme ile gerçekleştirmektedir. Böylelikle, hem veri akışı hızı yükselmiş hem de tasarımın FPGA kaynak kullanım profili daha tutarlı hâle gelmiştir.

Çizelge 4.1, AMIQ Consulting tarafından yayımlanan xfOpenCV tabanlı özgün tasarımda kullanılan fonksiyonların modern karşılıklarını ve aralarındaki yapısal farkları özetlemektedir. Bu eşleştirme, tasarımın kod seviyesinde güncellenmesi ve HLS sentezleyicisinin Vitis Vision yapıları üzerinden daha kararlı bir derleme ve optimizasyon yapabilmesi açısından kritik önemdedir.

Çizelge 4.1. FSRCNN tasarımında kullanılan fonksiyonların karşılaştırılması

İşlev	AMIQ Consulting Reposu (xfOpenCV / Legacy)	Modern Tasarım (Vitis Vision)
Arayüz Dönüşümü	cvMat2AXIvideo / AXIvideo2cvMat	cvMat2AXIvideoxf / AXIvideo2cvMatxf
Veri Tipi	hls::stream<ap_axiu<24,1,1,1>>	hls::stream<ap_axiu<24,1,1,1>> (Uyumlu)
Pencereleme	Manuel C++ döngüleri ve shift_pixels	xf::cv::Window ve insert_pixel metodları
Satır Tamponlama	Manuel array yönetimi	xf::cv::LineBuffer sınıfı
Konvolüsyon	İç içe döngülerle özel implementasyon	Optimize edilmiş Vitis Vision window çarpım yapısı

Tasarımın Vitis Vision kütüphanesine uyarlanması yalnızca fonksiyonel eş değişimi sağlamakla kalmamış, aynı zamanda tasarımın veri akış hatlarını daha yüksek donanım verimliliğiyle çalışabilir hâle getirmiştir. Testbench'in güncellenmesinin ardından, önceki çalışma (URL-2) yalnızca tek bir renk kanalının (mavi bileşeni) çözünürlüğünü artırmaya

yönelikken, modernize edilmiş sistem artık YCbCr renk uzayını tam olarak desteklemektedir. Bu kapsamda Y kanalı donanım üzerinde FSRCNN ağı tarafından işlenmekte, Cb ve Cr bileşenleri ise işlemci tarafında uygun yeniden boyutlandırma tekniğiyle ölçeklendirilerek birleşmekte ve sonuçta hem renkli hem de yüksek çözünürlüklü bir çıktı elde edilmektedir. Böylece tasarım hem FPGA üzerinde daha optimize bir hesaplama akışı sunmakta hem de tam renkli süper çözünürlük üretme kapasitesine sahip olmaktadır.

Bu modernizasyon yaklaşımı, çözünürlük iyileştirme algoritmasının donanım uyumluluğunu güçlendirmiş; veri akışı seviyesinde kararlılığı artırmış; optimize pencereleme, tamponlama ve konvolüsyon mekanizmalarıyla tasarımın toplam gecikmesini azaltmıştır. Aynı zamanda geliştirilen çözümün Vitis ekosistemiyle uyumlu hâle gelmesi, tasarımın uzun vadeli sürdürülebilirliğini de garanti altına almaktadır.

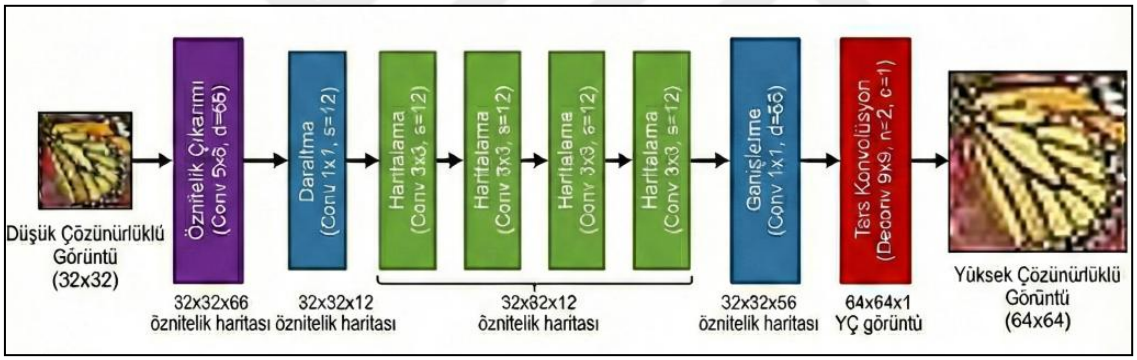
4.3. FSRCNN Mimarisi

Bu çalışmada kullanılan çözünürlük iyileştirme yaklaşımı, FSRCNN ağ mimarisinin hesaplama akışını temel almaktadır. FSRCNN, düşük çözünürlüklü giriş görüntüsünü hızlı ve verimli bir biçimde yüksek çözünürlüklü çıktıya dönüştürmek üzere tasarlanmış, ardışık konvolüsyon katmanlarından oluşan bir derin öğrenme modelidir. Model, Dong ve diğerleri (2016) tarafından geliştirilen özgün FSRCNN mimarisine dayanmakta olup; hesaplamaların düşük çözünürlük uzayında gerçekleştirilmesi, kanal boyutunun belirli katmanlarda daraltılması ve daha sonra yeniden genişletilmesi gibi yapısal özellikleri nedeniyle FPGA üzerinde uygulanmaya son derece elverişlidir. Dong ve diğerleri (2016), FSRCNN yapısını “kum saati (hourglass)” biçimli bir mimari olarak tanımlamıştır ve daraltma aşaması hesaplama yükünü azaltırken genişletme aşaması modelin yeniden yapılandırma kapasitesini güçlendirmektedir.

Bu çalışmada kullanılan tüm hesaplama süreci YCbCr renk uzayı üzerinden yürütülmüştür. Bu bağlamda giriş görüntüsü önce RGB’den YCbCr renk uzayına dönüştürülmekte; parlaklık bileşeni olan Y kanalı donanım üzerinde FSRCNN ağı tarafından işlenmektedir. İnsan görsel sistemi yüksek frekanslı ayrıntılara en çok Y bileşeninde duyarlı olduğundan, çözünürlük iyileştirme işleminin yalnızca bu kanalda uygulanması görüntü kalitesi açısından yeterli olmaktadır. Cb ve Cr bileşenleri ise işlemci tarafında yeniden

boyutlandırılarak Y kanalıyla birleştirilmekte ve sonuç görüntü yeniden RGB formatında elde edilmektedir. Bu yaklaşım, hem FSRCNN mimarisinin doğru şekilde uygulanması hem de FPGA üzerinde kaynak kullanımının optimize edilmesi açısından uygun bir yöntem sunmaktadır.

FSRCNN mimarisinin genel işlem akışı Şekil 4.1'de gösterilmiştir. Bu şematik yapı, giriş görüntüsünden özniteliklerin çıkarılması, kanal boyutunun daraltılması, haritalama katmanlarında dönüşümün gerçekleştirilmesi, kanalın yeniden genişletilmesi ve son olarak ters konvolüsyon aracılığıyla yüksek çözünürlüklü çıktının üretilmesi aşamalarından oluşmaktadır. Modelin ardışık yapısı, FPGA üzerinde boru hattı (pipeline) ve veri akışı (dataflow) ilkeleri ile uyumlu bir işlem dizisi oluşturmaktadır. FSRCNN mimarisini oluşturan tüm hesaplama aşamaları, HLS projesinde kullanılan kod bloklarıyla birlikte ayrıntılı biçimde sunulmuştur.



Şekil 4.1. Önerilen FSRCNN ağ yapısı (x2 yükseltme)

4.3.1. Öznitelik çıkarımı

Modelin ilk aşaması, düşük çözünürlüklü giriş görüntüsünden temel kenar, doku ve yapı bilgilerinin çıkarılmasıdır. Bu aşamada giriş görüntüsü, 5×5 boyutunda bir konvolüsyon filtresi ile işlenerek yüksek seviyeli olmayan ancak çözünürlük iyileştirme süreci için gerekli temel öznitelikler elde edilir. Bu aşamanın temel amacı, sonraki katmanlarda yapılacak dönüşümlere uygun bir öznitelik temsili oluşturmaktır. FPGA üzerinde bu işlemin uygulanması, sabit boyutlu pencerelerin sürekli görüntü akışı üzerinden kaydırılmasını ve her piksel için yerel komşuluk bilgilerinin ardışık döngülerde değerlendirilmesini gerektirir. Bu yapı, FSRCNN'nin hesaplama hattının ilk bileşenini oluşturmaktadır.

4.3.2. Daraltma

Öznitelik çıkarımı aşamasında elde edilen d kanal sayısındaki öznitelik haritaları, bu aşamada 1×1 boyutunda bir konvolüsyon filtresi uygulanarak s kanal sayısına indirgenir. Kanal boyutunun daraltılması, sonraki haritalama katmanlarında gerçekleştirilecek ardışık konvolüsyon işlemlerinin maliyetini önemli ölçüde azaltır. Shrink aşaması, kum saati benzetmesinin daralmaya başladığı bölgeyi temsil eder ve FSRCNN mimarisinin SRCNN'e göre çok daha hızlı çalışmasını mümkün kılan temel adımlardan biridir. FPGA üzerinde 1×1 konvolüsyon düşük hesaplama gereksinimiyle uygulanabildiği için bu aşama donanım performansına önemli bir katkı sağlamaktadır.

4.3.3. Haritalama

FSRCNN mimarisinin öğrenilmiş dönüşüm kapasitesinin en yoğun olduğu bölüm haritalama katmanlarıdır. Bu aşamada ardışık m adet 3×3 konvolüsyon uygulanarak, daraltılmış kanal boyutu üzerinde yüksek seviyeli özellik dönüşümleri gerçekleştirilir. Haritalama katmanları kum saatinin en dar noktasını temsil eder; burada ağ, giriş görüntüsünün çözünürlük artırımı için gerekli dönüşümleri soyut ve yüksek seviyeli bir temsil üzerinden hesaplar. Bu aşamanın FPGA üzerinde uygulanabilmesi için ardışık pencereleme, boru hattı akışı ve bellek erişiminin düzenli bir yapıda organize edilmesi gerekmektedir. Donanım üzerinde bu yapı, ardışık konvolüsyon çekirdeklerinin bir veri akışı hattı boyunca bağlanmasıyla sağlanmaktadır. Bir önceki Daraltma katmanından elde edilen 12 katmanlı öznitelik haritası FSRCNN veri akışı hattı boyunca toplamda dört Haritalama katmanında konvolüsyon işlemlerine tabi tutulacaktır.

4.3.4. Genişletme

Haritalama aşamasında daraltılan kanal boyutu, Genişletme aşamasında yeniden d kanal sayısına çıkarılır. 1x1 konvolüsyon tabanlı bu genişletme işlemi, kum saatinin genişleyen bölümünü temsil eder ve modelin yüksek çözünürlüklü detayları yeniden oluşturabilmesi için gerekli kanal boyutunu sağlar. Genişletme aşaması, hesaplama bakımından hafif bir yapıya sahiptir ve donanım üzerinde düşük maliyetle uygulanabilir. Böylece model, büyütme aşamasına uygun bir öznitelik yapısı oluşturur.

4.3.5. Ters konvolüsyon

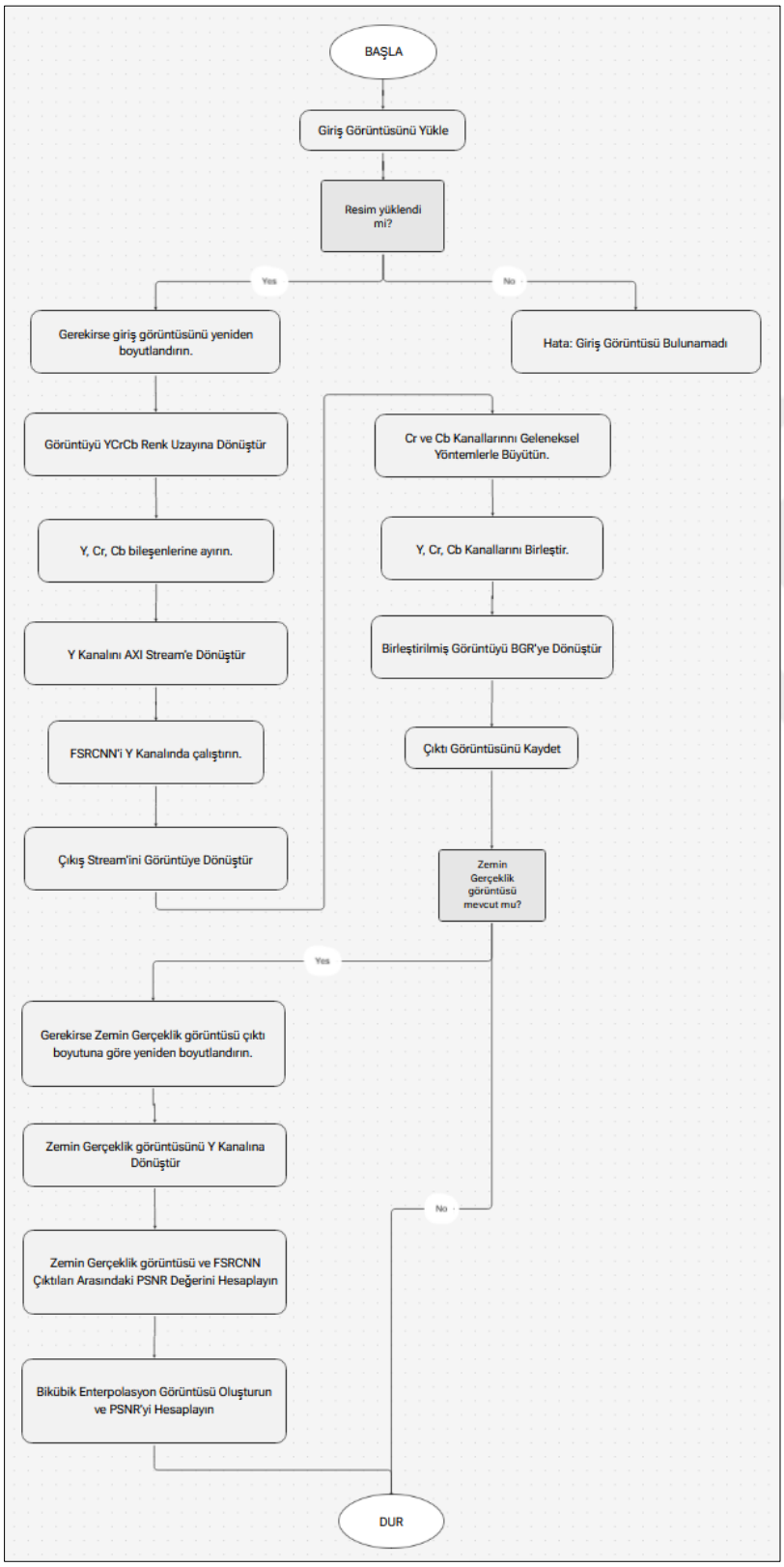
FSRCNN mimarisinin son aşaması, çözünürlüğü artırmak için kullanılan ters konvolüsyon (transposed convolution) katmanıdır. Dong ve diğerleri (2016), bu aşamada 9×9 boyutunda büyük bir filtre kullanmış ve ters konvolüsyonu, düşük çözünürlüklü öznitelik haritası üzerinde yeni pikselleri doğru geometrik konumlarına yerleştirerek yüksek çözünürlüklü görüntüyü yeniden kuran bir işlem olarak tanımlamıştır. Ters konvolüsyon, matematiksel olarak konvolüsyon işleminin uzamsal genişletme yapan bir tür ters eşleniği olup, temel mekanizması giriş öznitelik haritasına sıfır dolgulu örnekleme (zero-insertion) uygulamak ve ardından bu genişletilmiş yapıyı bir filtreyle konvolüsyon işleminden geçirmektir. Bu işlem yüksek frekans bileşenlerini yeniden oluşturarak çözünürlüğün artırılmasını sağlar.

Bu aşamanın donanım üzerinde uygulanması ise belirgin hesaplama ve bellek gereksinimleri doğurur. Çünkü ters konvolüsyon, standart konvolüsyona kıyasla daha büyük boyutlu filtreler ve daha karmaşık örnekleme düzenleri gerektirir. Ayrıca genişletilmiş öznitelik haritasının hesaplanması, adresleme ve bellek erişiminde yoğun bir kontrol yapısı gerektirir. Bu nedenle ters konvolüsyon FPGA üzerinde çalıştırıldığında hesaplama maliyetinin artmasına sebep olabilmektedir. Donanım üzerinde kaynak kullanımı ve gecikme süresinde fark edilir derecede maliyet artışına sebep olacak ters konvolüsyon aşaması işlemleri, oluşturulan HLS projesinde sentezlenen RTL'nin C/RTL kosimülasyon sonuçlarının alınabilmesi ve değerlendirilebilmesi için Vitis ortamının sağladığı pragma direktifleri etkin ve verimli kullanılarak HLS kodlarına dönüştürülmelidir.

Katmanları sırasıyla açıklanan FSRCNN mimarisi, düşük çözünürlüklü bir giriş görüntüsünden yüksek çözünürlüklü, daha net ve detaylı bir görüntü üretmeyi amaçlamaktadır. Modelin öznitelik çıkarımı ve haritalama katmanlarında belirgin biçimde öğrenilmiş dönüşümler gerçekleştirilmekte; ters konvolüsyon katmanı ise bu özellikleri yüksek çözünürlüklü piksel konumlarına taşımaktadır. Bu mimari uygulandığında, düşük çözünürlüklü görüntüdeki kenarların daha keskin, dokuların daha doğal ve genel görsel bütünlüğün daha yüksek bir algısal kaliteyle yeniden oluşturulması beklenmektedir.

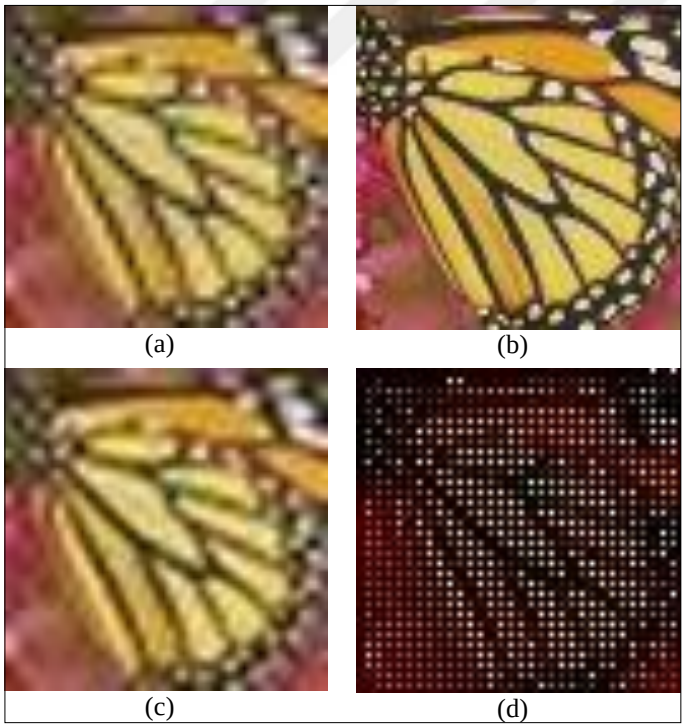
HLS projesinde üretilen bu tasarım kaynak kodu kullanılarak 32x32 büyüklüğünde giriş görseli okunup C-simülasyon ve RTL simülasyonunun sonucunun ayrı ayrı alınıp 64x64

büyüklüğünde çıktı görsellerinin üretilmesini sağlayacak testbench kod bloğu Şekil 4.2'deki akış diyagramı ile gösterilebilir.



Şekil 4.2. Testbench kodunun akış diyagramı

Bu testbench HLS projesinin ayarlarında dizin konumu belirtilmiş olan “input_image32x32.jpeg" ismindeki 32x32 büyüklüğündeki görseli okuduktan sonra çözünürlük iyileştirirken kullanılan yöntemin hesaplama maliyetini düşürmek amacıyla görseli RGB renk uzayından YCbCr renk uzayına dönüştürür. Devamında bu renk uzayından insan gözünün daha duyarlı olduğu Y kanalı verileri tasarım kaynak kodundaki işlemlere tabi tutulur. Testbench, Cb ve Cr kanal verilerini bikübik enterpolasyonla 2x büyüttükten sonra çözünürlüğü iyileştirilmiş Y kanalı verileriyle tekrar birleştirir. Son olarak görseli RGB renk uzayına dönüştürür ve "output_image_64x64_vvision_deconv.jpeg" ismiyle proje dizinine kaydeder. Bu çalışma için kullanılan “input_image32x32.jpeg" 32x32 giriş görseli Resim 4.1.a’da, giriş görselinin 64x64 zemin gerçeği versiyonu Resim 4.1.b’de, bikübik enterpolasyonla elde edilmiş 64x64 çıkış görseli Resim 4.1.c’de ve oluşturulan tasarım kaynak kodunda giriş görselinin Y kanalının işlenmesi sonucunda elde edilen "output_image_64x64_vvision_deconv.jpeg" 64x64 çıkış görseli ise Resim 4.1.d’de verilmiştir:



Resim 4.1. Tasarım kaynak kodunun çıktısının görsel analizi a. 32x32 giriş görseli, b. Giriş görselinin 64x64 zemin gerçeği versiyonu, c. Bikübik enterpolasyonla elde edilmiş 64x64 çıkış görseli, d. Tasarım kaynak kodunun 64x64 çıkış görseli

Ters konvolüsyon katmanı kullanılarak elde edilen yüksek çözünürlüklü çıktı incelendiğinde, özellikle ince dokuların ve yüksek frekanslı bölgelerin yer aldığı alanlarda düzenli bir gürültü örüntüsünün ortaya çıktığı gözlemlenmiştir. Bu durum, ters konvolüsyon işleminin doğasında bulunan düzensiz örnekleme ve dengesiz örtüşme (uneven overlap) yapısının bir sonucudur; filtre pencerelerinin uzamsal ızgarayı eşit biçimde kaplamaması, periyodik bozulmaların oluşmasına yol açmaktadır (URL-1). Bu bozulma, özellikle ince çizgilerin veya yarı saydam yapıların yakınında görüntünün "örgü dokusunun arkasından" görülüyormuş izlenimi veren periyodik bir artefakt oluşturmakta olup, CNN tabanlı üst örnekleme katmanlarında rastlanan tipik bir davranıştır. Sugawara ve diğerleri (2019), bu tür dama tahtası ("checkerboard") yapılarının hem ileri yayılım hem de geriye yayılım süreçlerinde ortaya çıkabileceğini ve özellikle yüksek frekanslı bölgelerde daha görünür hâle geldiğini ifade etmektedir. Bu nedenle burada gözlemlenen örüntüler, ters konvolüsyon tabanlı mimarilerin bilinen bir sınırlılığını doğrulamakta ve bir sonraki bölümde ele alınacak alternatif büyütme yapısının gerekliliğini açık biçimde ortaya koymaktadır.

4.4. FSRCNN Mimarisinin Son Katmanındaki İyileştirme Çalışmaları

4.4.1. Yeniden boyutlandırmalı konvolüsyon yöntemi

Dong ve diğerlerinin (2016) SRCNN'den daha az donanım yüküyle çözünürlüğü iyileştirilmiş görsele ulaşmak için ileri sürdükleri FSRCNN mimarisine bağlı kalarak oluşturulan tasarım kaynak kodu, çözünürlük iyileştirme ve görsel büyütme işlemlerini ters konvolüsyon yöntemleri ile yapmaktadır. Ancak bu tasarım kaynak kodu kullanılarak oluşturulan çıktı görseli incelendiğinde kullanılan filtre boyutu ile ölçekleme oranı arasında yapısal bir uyumsuzluk bulunduğundan görüntüde periyodik desenler şeklinde artefaktlar oluşabilmektedir. Daha önce yapılmış çalışmalar tarandığında bu problem, URL-1'de kapsamlı biçimde analiz edilmiş olup temel nedenin dengesiz örtüşme olduğu gösterilmiştir. Dengesiz örtüşme, görsel büyütme işlemi sırasında her çıkış pikselinin giriş uzayından aynı yoğunlukta filtre katkısı almaması durumunu ifade eder. Ters konvolüsyon yönteminde filtre konumlarının uzamsal olarak düzenli dağılmaması, bazı bölgelerin aşırı, bazılarının ise yetersiz örneklenmesine yol açarak periyodik artefaktların oluşmasına neden olur. Bu bağlamda FSRCNN'nin ilk iterasyonunda, başka bir deyişle 4.3 bölümünde çıktıları paylaşılan tasarım kaynak kodunda, elde edilen çıktıların belirli bölgelerinde,

özellikle ince yapısal detayların bulunduğu alanlarda örüntüsel bozulmaların ortaya çıkması beklenen bir davranıştır.

Bu çalışma kapsamında çözünürlük iyileştirme ve görsel büyütme katmanı olan son katman söz konusu görüntü bozulmalarını ortadan kaldırmak amacıyla ters konvolüsyon yerine yeniden boyutlandırmalı konvolüsyon (resize–convolution) yöntemi kullanılarak yeniden tasarlanmıştır. Yeniden boyutlandırmalı konvolüsyon yöntemi, çözünürlük iyileştirme problemini iki ayrı ve matematiksel olarak kararlı adıma ayırmaktadır. İlk adımda, düşük çözünürlüklü öznitelik haritası en yakın komşu (nearest-neighbor) yöntemine dayalı deterministik bir uzamsal genişletme ile 2x boyutuna çıkarılmaktadır. Bu ölçekleme işlemi, ters konvolüsyonda olduğu gibi düzensiz örnekleme içermez; her piksel tek biçimde çoğaltılarak yüksek çözünürlüklü ızgaraya yerleştirilir. İkinci adımda ise 3x3 boyutunda bir konvolüsyon uygulanmakta ve genişletilmiş görüntü üzerindeki geçiş bölgeleri düzenlenerek doğal ve artefaktsız bir yüksek çözünürlüklü görüntü elde edilmektedir. Bu iki aşamalı yapı, hem teorik olarak kararlı hem de donanımda uygulanması son derece verimli bir mimari sunmaktadır.

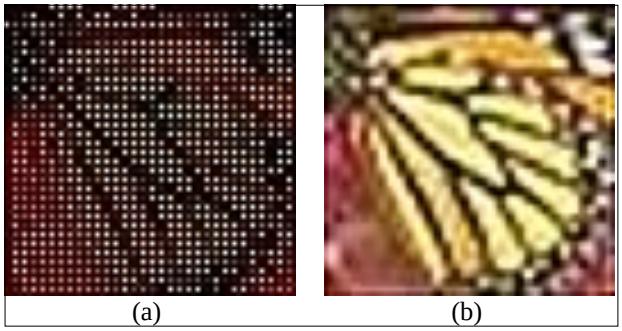
Söz konusu yaklaşımın FPGA üzerindeki uygulanabilirliği, basit pikselleme mantığına dayanan en yakın komşu yönteminin hiçbir ek çarpım gerektirmemesi ve 3x3 konvolüsyonun küçük pencere boyutu sayesinde DSP kullanımını optimize etmesiyle güçlenmektedir. Ayrıca Vitis Vision kütüphanesi fonksiyonları aktif bir şekilde kullanılarak filtre pencere kaydırması, satır tamponlama ve pipeline yapıları açısından düzenli bir hesaplama akışı sağlanmıştır. Vitis 2025.1'in pipeline ve dataflow gibi optimizasyon mekanizmaları sayesinde çözünürlük iyileştirme ve görsel büyütme işlemleri, ters konvolüsyona kıyasla donanım üzerinde çok daha verimli şekilde sentezlenebilmektedir. Sugawara ve diğerleri (2019), yeniden boyutlandırma tabanlı yöntemlerde hesaplama adımlarının uzamsal olarak tutarlı bir dağılım ürettiğini ve dolayısıyla dama tahtası etkilerinin oluşmadığını teorik olarak göstermiştir. Bu çalışma kapsamında da aynı sonuç doğrulanmıştır.

Yeniden boyutlandırmalı konvolüsyon yöntemi, HLS tasarımında genişletme katmanından elde edilen 56 kanallı öznitelik haritasının iki ardışık fonksiyon aracılığıyla işlenmesiyle gerçekleştirilmiştir. İlk fonksiyon her uzamsal konumda 56 giriş kanalını tek tek okuyarak, bu kanallar üzerinde ağırlıklandırılmış bir toplama işlemi gerçekleştirmektedir. Böylece

kanal boyutunun 56'dan 1'e indirildiği 1x1 konvolüsyon tabanlı bir projeksiyon üretmektedir. Aynı fonksiyon, elde edilen bu tek kanallı parlaklık (Y) temsilini, her piksel değerini yatay ve dikey yönde ikişer kez çoğaltarak en yakın komşu (nearest neighbor) yöntemiyle 2x büyütme ile yüksek çözünürlüklü uzaya taşımaktadır. Büyütülen bu tek kanallı öznitelik haritası, ikinci fonksiyona aktarılmaktadır. Bu fonksiyonda 3x3 boyutunda sabit bir çekirdek ile konvolüsyon uygulanarak uzamsal geçişler düzenlenmekte ve yüksek frekanslı bileşenler daha doğal bir biçimde yeniden yapılandırılmaktadır.

İkinci iterasyon çıktılarının görsel niteliğini daha iyi değerlendirebilmek amacıyla testbench içerisinde görüntüye ayrıca kenar keskinleştirme maskesi (unsharp masking) uygulanmıştır. Bu teknik, bulanıklaştırılmış bir görüntünün özgün görüntüden çıkarılmasıyla elde edilen yüksek frekans bileşenlerinin yeniden eklenmesine dayanan klasik bir keskinleştirme yöntemidir. Ayrıca, testbench karşılaştırmanın sağlıklı yapılabilmesi için bikübik enterpolasyonla büyütme uygulayabilecek ve 64x64 zemin gerçeği ve çıkış görsellerinin PSNR kalite metriğini hesaplayabilecek hale getirilmiştir.

Bu yöntem değişikliğin etkisi görsel çıktılar üzerinde açıkça görülmektedir. Ters konvolüsyon yöntemini gerçekleştiren ilk iterasyonda üretilen yüksek çözünürlüklü görüntü, belirli bölgelerde örgü dokusunu andıran düzenli bozulmalar içermekteyken, yeniden boyutlandırmalı konvolüsyon yöntemiyle elde edilen sonuç daha doğal, daha pürüzsüz ve dokusal olarak daha tutarlı bir yapı sergilemektedir. Resim 4.2.a'daki görsel ters konvolüsyon yönteminin uygulandığı tasarım kaynak kodunun 64x64 çıkış görseli iken Resim 4.2.b ikinci iterasyonla oluşturulan ve yeniden boyutlandırmalı konvolüsyon yönteminin gerçekleştirildiği tasarım kaynak kodunun 64x64 çıkış görselini göstermektedir. Burada ilk iterasyonda görülen periyodik örüntüler, ikinci iterasyonda yerini daha düzgün ve artefaktsız bir yeniden yapılandırmaya bırakmıştır. Bu gözlem, yeniden boyutlandırmalı konvolüsyon yönteminin özellikle algısal kalite üzerinde açık bir üstünlük sunduğunu göstermektedir.



Resim 4.2. İterasyon 1-2 tasarım kaynak kodlarının çıktılarının görsel analizi a. İterasyon-1 (ters konvolüsyon yöntemi) 64x64 çıkış görseli, b. İterasyon-2 (yeniden boyutlandırmalı konvolüsyon yöntemi) 64x64 çıkış görseli

Çizelge 4.2. İterasyon 1-2 tasarım kaynak kodlarının çıktılarının PSNR kalite metrikleri

Yöntem	Mutlak PSNR (dB)
Bikübik Enterpolasyon	19,3437 dB
İterasyon-1 (Ters Konvolüsyon)	5,9340 dB
İterasyon-2 (Yeniden Boyutlandırmalı Konvolüsyon)	15,7577 dB

Çizelge 4.2 ile kullanılan her iki yöntemin Tepe Sinyal-Gürültü Oranı (PSNR) kalite metriği değerleri bikübik enterpolasyon PSNR değeri referans alınarak karşılaştırılmaktadır. Ters konvolüsyonun ürettiği düşük PSNR değeri, dama tahtası etkisinden kaynaklı görüntü bozukluklarının gürültü oranını artırmasından kaynaklanırken; yeniden boyutlandırmalı konvolüsyon daha düzgün ve artefaktsız bir çıkış görüntüsü elde edebildiği için daha yüksek bir PSNR değeri üretmiştir. Bu sonuç, teorik beklenti ile deneysel bulguların tamamen uyumlu olduğunu ve yöntem geçişinin isabetli bir karar olduğunu doğrulamaktadır. Buna rağmen ikinci iterasyon ile elde edilen tasarım kaynak kodunun çıktısının PSNR değerinin bikübik enterpolasyon çıktısının PSNR değerinden daha düşük olması, testbench'te uygulanan ve yüksek frekanslı ayrıntıları iyileştirmeyi amaçlayan kenar keskinleştirme maskesinin piksel bazında hata oranını (MSE) artırmasından kaynaklanmaktadır. Derin öğrenme modeli kullanılarak uygulanan çözünürlük iyileştirme yönteminin PSNR kalite metriği kullanılarak bikübik enterpolasyon yöntemine göre PSNR metriğine göre daha iyi bir çıktı verebilmesi adına yeniden boyutlandırmalı konvolüsyon yöntemine geçmek yerine Dong ve diğerlerinin (2016) önerdiği FSRCNN mimarisine sadık kalınarak algoritmada iyileştirmeler yapılmalıdır.

4.4.2. Çok fazlı ters konvolüsyon ve artefakt optimizasyonu

FSRCNN mimarisinin son katmanında gerçekleştirilen görsel büyütme işlemi, ilk tasarım iterasyonunda pikseller arası yoğunluk dengesizliklerinden kaynaklanan periyodik artefaktlara yol açmıştır. Bu problemin donanım düzeyinde giderilmesi amacıyla, ters konvolüsyon işlemi çok fazlı bir filtre yapısına dönüştürülerek yeniden yapılandırılmıştır.

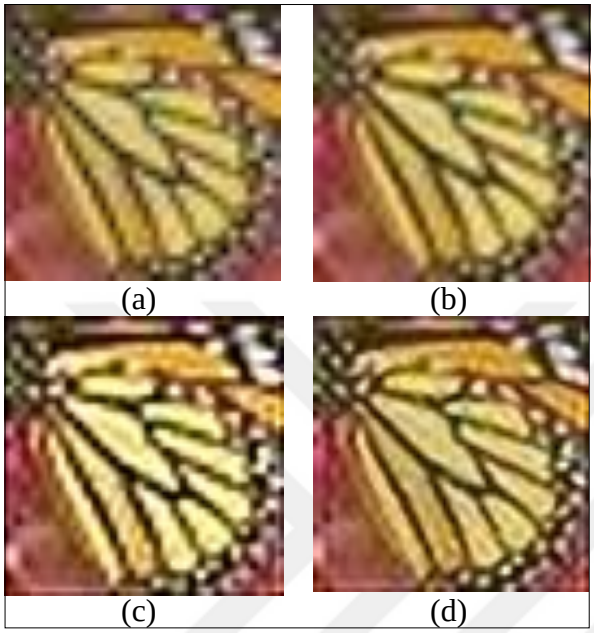
Ters konvolüsyonun matematiksel olarak genişletilmiş bir ızgara üzerinde standart konvolüsyon uygulama prensibi, HLS üzerinde doğrudan gerçeklendiğinde evrişimsel izdüşüm hatalarına neden olabilmektedir. Tasarımın üçüncü iterasyonunda, her bir giriş pikselinden eş zamanlı olarak 2×2 boyutunda bir çıktı bloğu üreten çok fazlı filtre bankası yaklaşımı benimsenmiştir. Bu yöntemde, 9×9 boyutundaki orijinal filtre çekirdeği dört farklı faza (üst-sol, üst-sağ, alt-sol, alt-sağ) bölünerek her bir fazın çıktı ızgarası üzerindeki mekânsal karşılığı hassas bir şekilde tanımlanmıştır.

Dama tahtası etkisinin temel nedeni olan dengesiz örnekleme yoğunluğunu ortadan kaldırmak için, filtre katsayılarının indeksleme mantığı revize edilmiştir. Filtre çekirdeğinin ters konvolüsyon karakteristiğine uygun şekilde aynalanması ve her bir fazın hesaplanmasında kullanılan ağırlıkların sınır denetimli bir yapıda ($j \geq 1$ ve $i \geq 1$ gibi) kurgulanması sağlanmıştır. Bu sayede, çıktı piksellerinin her birinin giriş uzayından eşit miktarda ve doğru fazda filtre katkısı alması temin edilerek periyodik desen bozulmaları tamamen elimine edilmiştir.

Ters konvolüsyon yöntemi işlemindeki bu iyileştirmeyle oluşturulan tasarım kaynak kodu, projenin üçüncü iterasyonu olarak kabul edilmiştir. Üçüncü iterasyon çıktılarının görsel niteliğini daha iyi değerlendirebilmek ve çıktıların PSNR kalite metriğini bikübik enterpolasyon yönteminin çıktılarına göre kıyaslayabilmek amacıyla 32x32 büyüklüğünde giriş görselinin okunmasını ve tasarım kaynak kodu tarafından işlendikten sonra oluşturulan 64x64 büyüklüğünde çıktı görsellerinin üretilmesini sağlayacak testbench kodu önceki iterasyonlardaki gibi performans metriklerini de ölçecek şekilde kapsamlı bir şekilde oluşturulmuştur.

Resim 4.3.a'da 32x32 giriş görseli, Resim 4.3.b'de bikübik enterpolasyonla elde edilmiş 64x64 çıkış görseli, Resim 4.3.c'de ikinci iterasyonla oluşturulan ve yeniden

boyutlandırmalı konvolüsyon yönteminin gerçekleştirildiği tasarım kaynak kodunun 64x64 çıkış görseli ve Resim 4.3.d'de ise üçüncü iterasyonla oluşturulan ve ters konvolüsyon işleminde gerçekleştirilen iyileştirmeler sonucunda elde edilen 64x64 çıkış görseli verilmiştir.



Resim 4.3. İterasyon-3 tasarım kaynak kodunun çıktısının karşılaştırmalı görsel analizi a. 32x32 giriş görseli, b. Bikübik enterpolasyonla elde edilmiş 64x64 çıkış görseli, c. İterasyon-2 64x64 çıkış görseli, d. İterasyon-3 64x64 çıkış görseli

Çizelge 4.3'te bikübik enterpolasyon ve öne sürülen üç tasarım kaynak kodunun, giriş görselinin 64x64 boyutlarındaki versiyonu zemin gerçeği olarak kullanılarak hesaplanan Tepe Sinyal-Gürültü Oranı (PSNR) kalite metriği değerleri gösterilmektedir.

Çizelge 4.3. İterasyon 1, 2 ve 3 tasarım kaynak kodlarının çıktılarının PSNR kalite metrikleri

Yöntem	Mutlak PSNR (dB)
Bikübik Enterpolasyon	19,3437 dB
İterasyon-1 (Ters Konvolüsyon)	5,9340 dB
İterasyon-2 (Yeniden Boyutlandırmalı Konvolüsyon)	15,7577 dB
İterasyon-3 (İyileştirilmiş Ters Konvolüsyon)	20,0715 dB

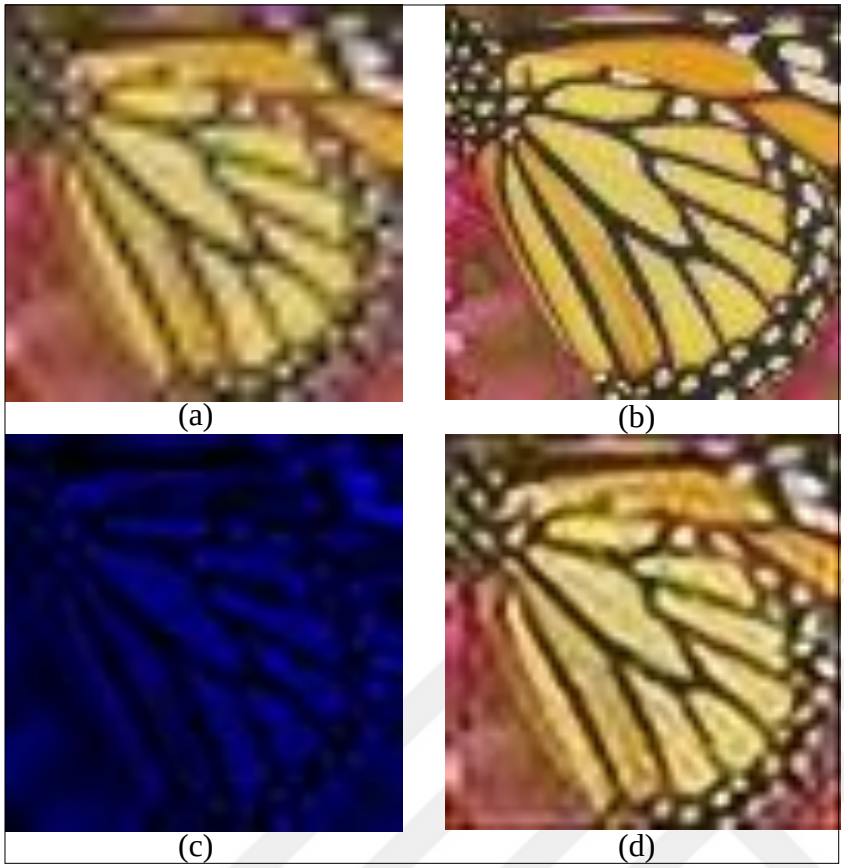
Dong ve diğerlerinin (2016) önerdiği FSRCNN mimarisine sadık kalınarak ve algoritmada iyileştirmeler yapılarak elde edilen tasarım kaynak kodu, PSNR kalite metriği değerleri incelendiğinde ve görsel olarak analiz edildiğinde önceki iterasyonların ve bikübik enterpolasyon yönteminin çıkışından daha başarılı bir çıkış sunmaktadır. Bu açıdan değerlendirildiğinde FSRCNN mimarisi kullanılarak alınan en başarılı sonuçların iterasyon-3 ile alınan sonuçlar olduğu söylenebilir.

4.5. Deneysel Sonuçlar

Bu bölümde, ortaya konan modern Vitis Vision kütüphanesi fonksiyonlarıyla inşa edilmiş HLS tasarımının, bir başka deyişle üçüncü iterasyon tasarım kaynak kodu ile oluşturulan HLS tasarımının, çıktıları görsel kalite (PSNR, SSIM) ve donanım performansı (Gecikme, Kaynak Kullanımı) açısından değerlendirilmiştir. Elde edilen sentez ve simülasyon sonuçları, literatürdeki benzer süper çözünürlük çalışmalarıyla karşılaştırılmıştır.

4.5.1. Görsel kalite ve performans metrikleri analizi

Donanım tarafından oluşturulan sonuçların kalitesini değerlendirmek için görsel incelemeler gerçekleştirilmiştir. Resim 4.4.a 32x32 giriş görselini, Resim 4.4.b giriş görselinin 64x64 zemin gerçeği versiyonunu, Resim 4.4.c xfOpenCV uygulamasının 64x64 çıktısını ve Resim 4.4.d ise önerilen tasarımın, yani İterasyon-3'ün 64x64 çıktısını göstermektedir. xfOpenCV uygulamasının FPGA üzerindeki önemli iş yükü ve kısıtlı donanım kaynakları nedeniyle tek bir renk kanalını (Mavi renk kanalı) işlediği göz ardı edilmemelidir. Öngörüldüğü gibi, bu tasarım kararıyla üretilen çıktı, ağırlıklı olarak mavi tonlar içeren bir görüntüdür. Buna karşılık; önerilen tasarım, mimarisi gereği Y (Parlaklık) kanalını donanım aracılığıyla işler ve renk kanallarını (Cr, Cb) birleşik bir yaklaşım kullanarak entegre eder. Bunun sonucunda oluşturulan çıkış görselindeki renk dengesi korunur.



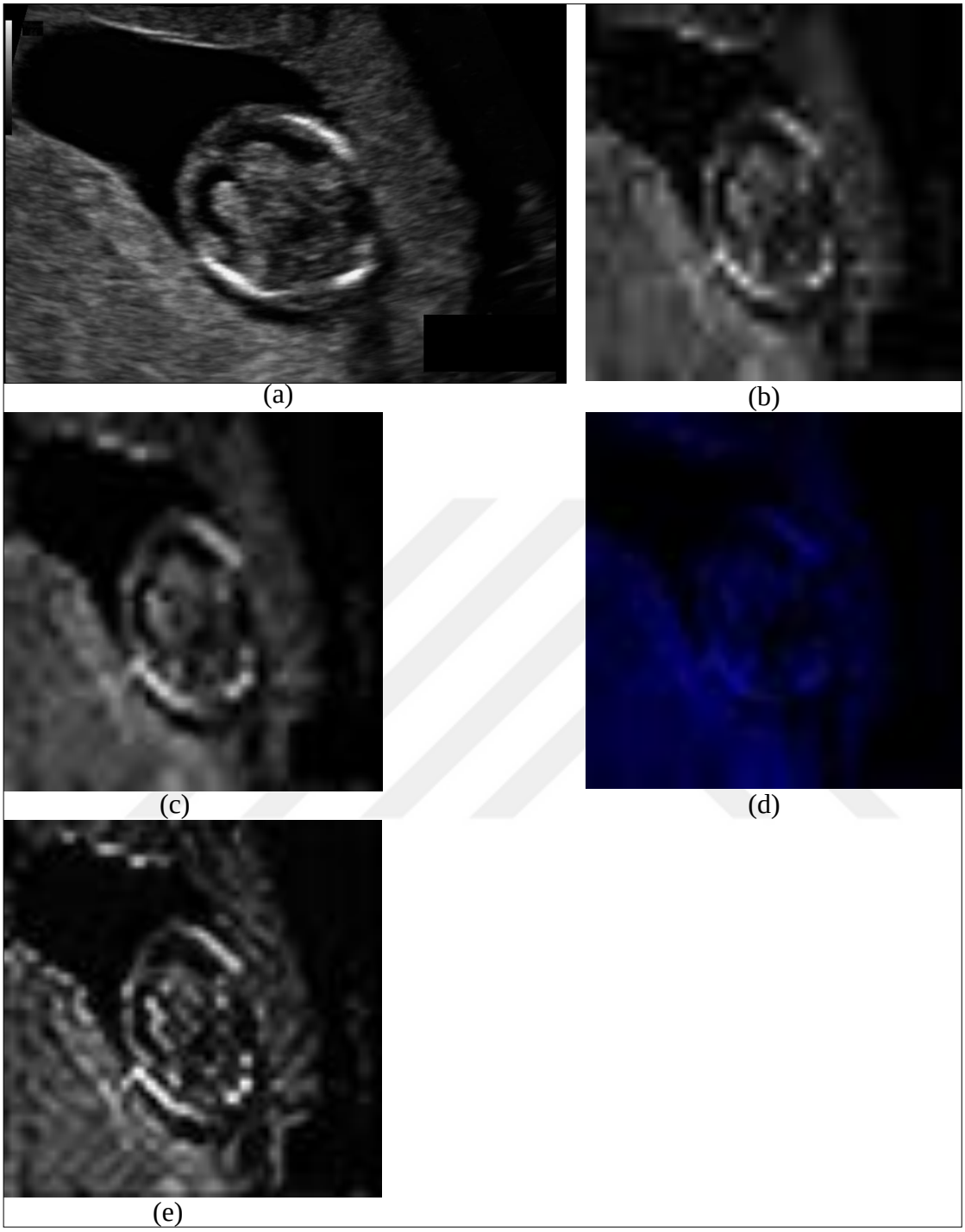
Resim 4.4. xfOpenCV uygulamasının ve önerilen tasarımının çıktılarının görsel analizi a. 32x32 giriş görseli, b. Giriş görselinin 64x64 zemin gerçeği versiyonu, c. xfOpenCV uygulamasının 64x64 çıkış görseli, d. Önerilen tasarımın (İterasyon-3) 64x64 çıkış görseli

Önerilen tasarımın görsel kalitesini ölçmek için PSNR (Tepe Sinyal-Gürültü Oranı) ve SSIM (Yapısal Benzerlik İndeksi) metrikleri kullanılmıştır. PSNR, sıkıştırma veya yeniden oluşturma sırasındaki piksel kayıplarını logaritmik ölçekte ifade ederken; SSIM, görüntünün yapısal bütünlüğünü insan görsel algısına daha yakın bir şekilde modeller. Çizelge 4.4; literatürde incelenen çalışmaların, tek renk kanalını işleyerek çıktı üreten xfOpenCV uygulamasının, bikübik enterpolasyon yönteminin ve önerilen HLS tasarımının PSNR ve SSIM kalite metrik değerlerini göstermektedir.

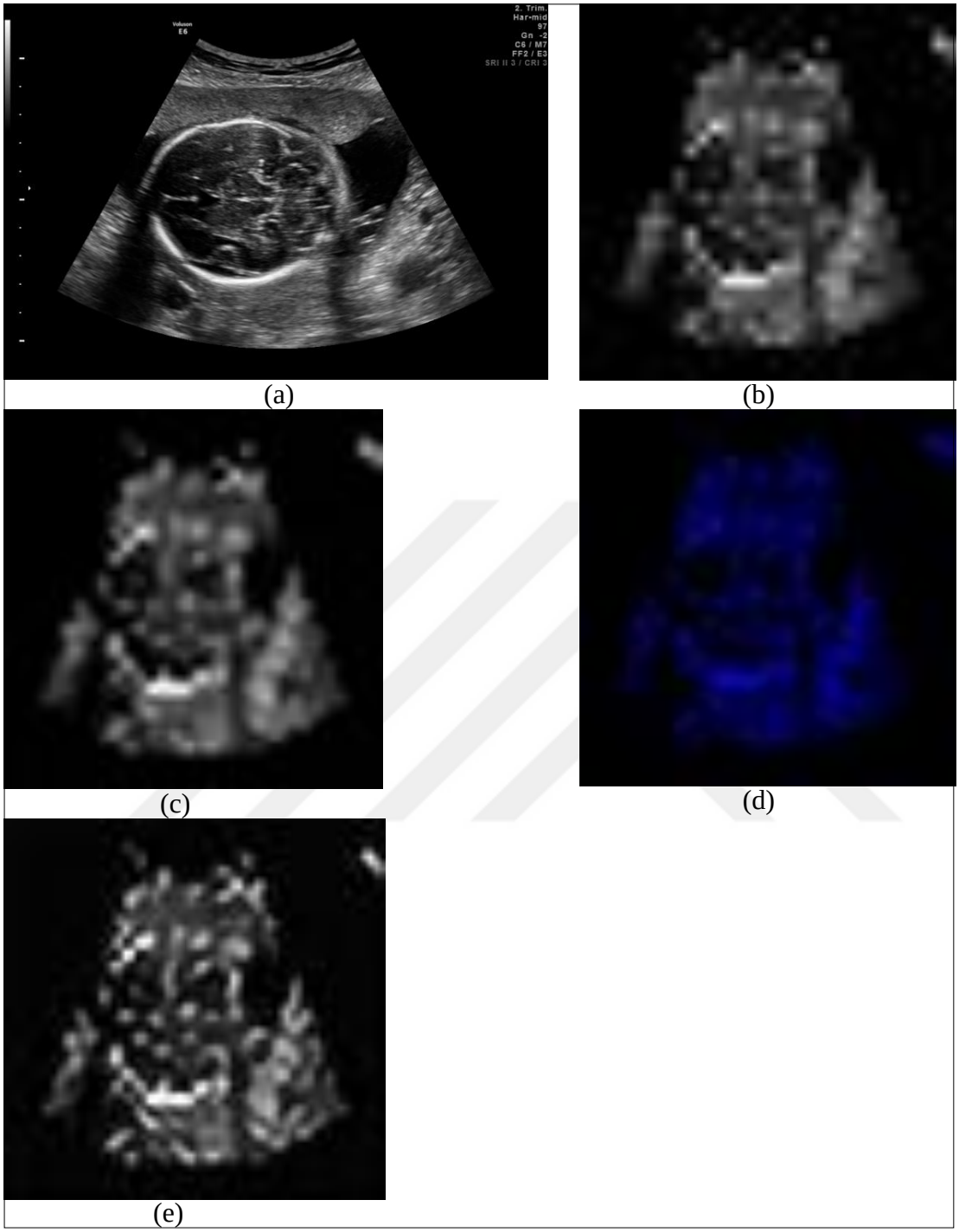
Çizelge 4.4. Çeşitli çalışmalar ve önerilen tasarımının çıktılarının görsel kalite performans metrikleri

Çalışma	PSNR (dB)	SSIM
Spagnolo ve diğerleri (2023)	39,03	0,9631
Shi ve diğerleri (2023)	36,40	0,9784
Su ve diğerleri (2024)	29,82	-
Amin ve Obermaisser (2025)	34,03	0,9560
Joseph ve Jayakumar (2024)	36,92	0,94
URL-2	5,9701	0,376
URL-3	26,48	0,894
Bikübik Enterpolasyon	19,3437	0,784
Önerilen Tasarım (İterasyon-3)	20,0715	0,876

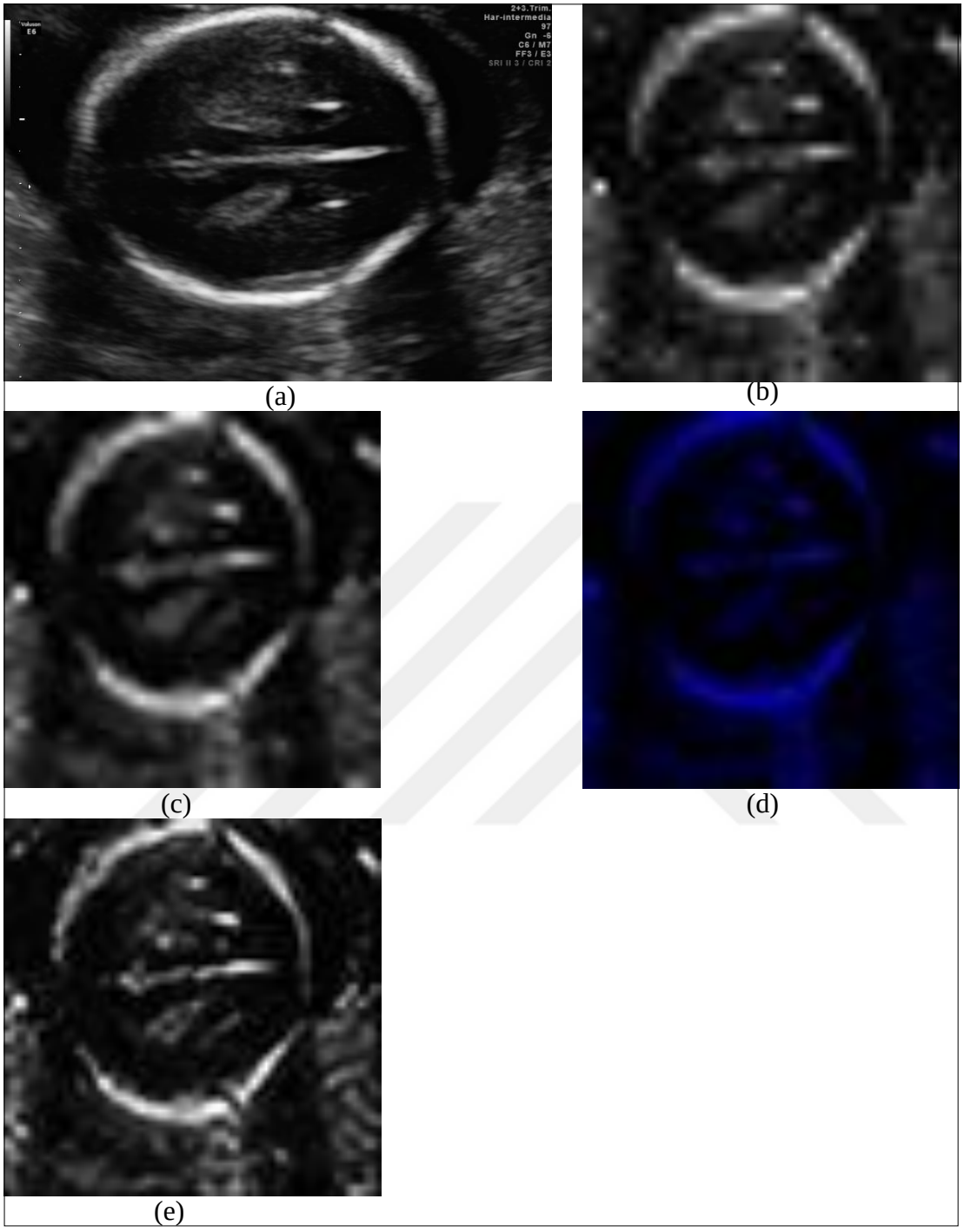
Önerilen HLS tasarımı, ultrason görüntülerinin çözünürlüklerinin iyileştirilmesi için de kullanılabilir. Alzubaidi ve diğerlerinin (2023) çalışmaları sonucunda inşa edilen 959x661 piksel çözünürlüğüne sahip 3832 görüntüden oluşan veri setinden çeşitli ultrason görüntüleri önerilen HLS tasarımının performansını gözlemleyebilmek amacıyla seçilmiş ve 32x32 boyutuna indirgenerek çözünürlükleri iyileştirilmiştir. Resim 4.5.a, Resim 4.6.a, Resim 4.7.a ve Resim 4.8.a veri setinden alınan görüntülerin özgün halini göstermekteyken Resim 4.5.b, Resim 4.6.b, Resim 4.7.b ve Resim 4.8.b veri setinden alınan görüntülerin HLS tasarımında kullanmak üzere indirgendiği 32x32 boyutundaki halini göstermektedir. Resim 4.5.c, Resim 4.6.c, Resim 4.7.c ve Resim 4.8.c'de her bir giriş görüntüsünün bikübik enterpolasyonla elde edilmiş 64x64 çıkış görüntüsü verilmiştir. Resim 4.5.d, Resim 4.6.d, Resim 4.7.d ve Resim 4.8.d görüntüleri xfOpenCV uygulamasının 64x64 çıkış görüntülerini göstermekteyken Resim 4.5.e, Resim 4.6.e, Resim 4.7.e ve Resim 4.8.e önerilen tasarımın, yani İterasyon-3'ün 64x64 çıkış görüntülerini göstermektedir.



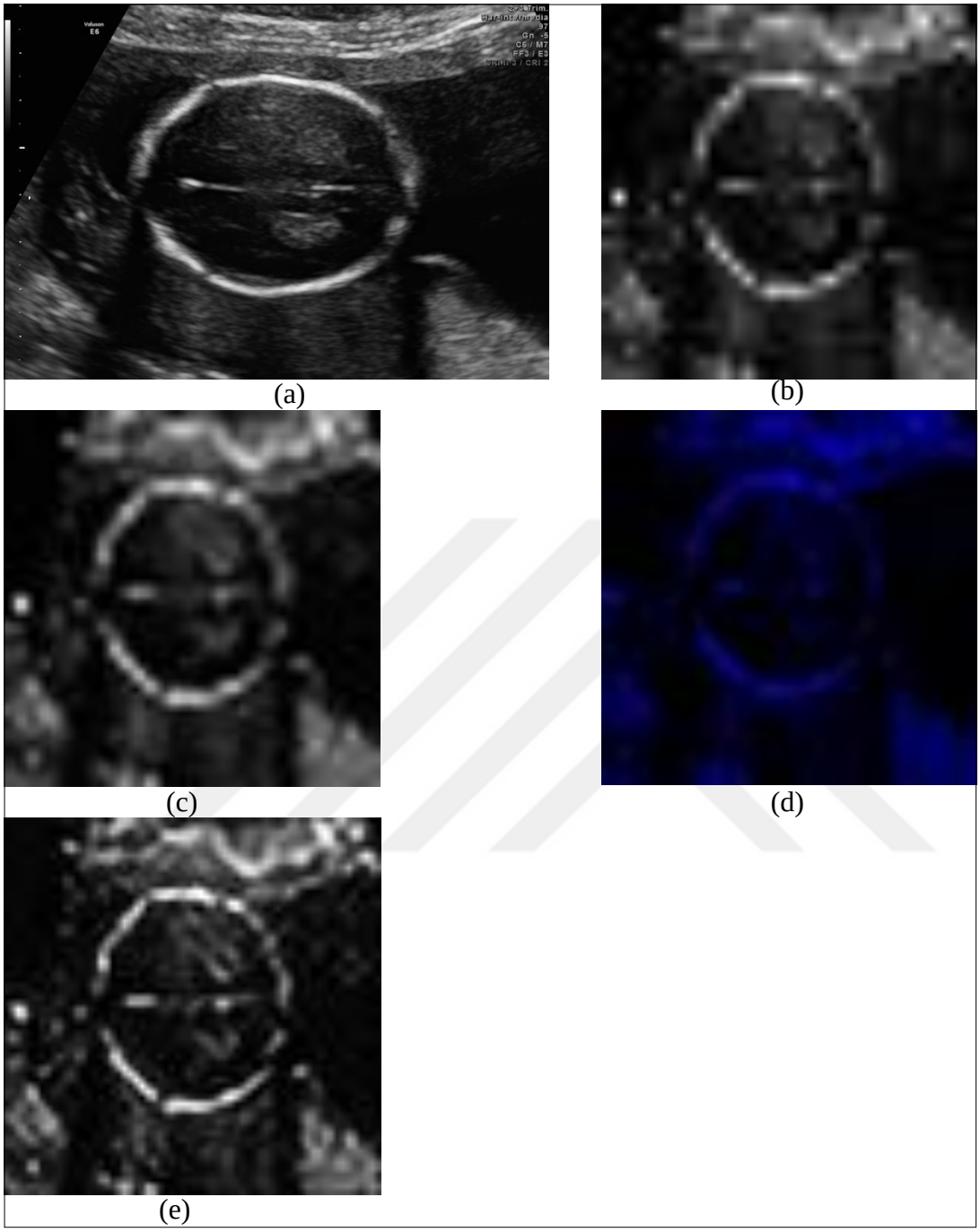
Resim 4.5. Ultrason veri setindeki fetal baş görüntüsünün farklı çözünürlük iyileştirme yöntemleri altındaki çıkış görüntüleri a. Veri setinden alınan Ultrason görüntüsünün özgün hali, b. Ultrason görüntüsünün 32x32 boyutuna dönüştürülmüş hali, c. Bikübik enterpolasyonla elde edilmiş 64x64 çıkış görüntüsü, d. xfOpenCV uygulamasının 64x64 çıkış görüntüsü, e. Önerilen tasarımın (İterasyon-3) 64x64 çıkış görüntüsü



Resim 4.6. Ultrason veri setindeki trans-serebellar görüntüsünün farklı çözünürlük iyileştirme yöntemleri altındaki çıkış görüntüleri a. Veri setinden alınan Ultrason görüntüsünün özgün hali, b. Ultrason görüntüsünün 32x32 boyutuna dönüştürülmüş hali, c. Bikübik enterpolasyonla elde edilmiş 64x64 çıkış görüntüsü, d. xfOpenCV uygulamasının 64x64 çıkış görüntüsü, e. Önerilen tasarımın (İterasyon-3) 64x64 çıkış görüntüsü



Resim 4.7. Ultrason veri setindeki trans-talamik düzlem görüntüsünün farklı çözünürlük iyileştirme yöntemleri altındaki çıkış görüntüleri a. Veri setinden alınan Ultrason görüntüsünün özgün hali, b. Ultrason görüntüsünün 32x32 boyutuna dönüştürülmüş hali, c. Bikübik enterpolasyonla elde edilmiş 64x64 çıkış görüntüsü, d. xfOpenCV uygulamasının 64x64 çıkış görüntüsü, e. Önerilen tasarımın (İterasyon-3) 64x64 çıkış görüntüsü



Resim 4.8. Ultrason veri setindeki trans-ventriküler kesit görüntüsünün farklı çözünürlük iyileştirme yöntemleri altındaki çıkış görüntüleri a. Veri setinden alınan Ultrason görüntüsünün özgün hali, b. Ultrason görüntüsünün 32x32 boyutuna dönüştürülmüş hali, c. Bikübik enterpolasyonla elde edilmiş 64x64 çıkış görüntüsü, d. xfOpenCV uygulamasının 64x64 çıkış görüntüsü, e. Önerilen tasarımın (İterasyon-3) 64x64 çıkış görüntüsü

Ultrason veri setindeki görüntüler, HLS tasarımlarının işleyebileceği hale dönüştürülmek üzere kendi özgün boyutlarından pi7 çevrimiçi görüntü aracı[1] kullanılarak 32x32 görsel

[1] Web: https://image.pi7.org/resize-image-pixel Son Erişim Tarihi: 10.11.2025.

boyutlarına indirgenmiştir. Sonrasında bu görüntüler sırasıyla bikübik enterpolasyon, tek renk kanalının işlendiği xfOpenCV uygulaması ve önerilen HLS tasarımında giriş görüntüsü olarak kullanılmış olup her biri için ayrı ayrı çıkış görüntüleri elde edilmiştir. Burada paylaşılan xfOpenCV uygulamasının çıkış görüntüleri, her bir giriş görüntüsünün sadece mavi renk kanalının çözünürlüğünün FSRCNN yöntemiyle iyileştirilmesiyle elde edilmiş görüntülerdir. Diğer renk kanallarının çözünürlükleri iyileştirilmeden bu sonuçların sunulma sebebi, belirlenen hedef platforma sığabilecek xfOpenCV HLS tasarımının ancak tek renk kanalını işlemeye yeterli kaynak içermesidir. Değerlendirilen farklı çözünürlük iyileştirme yöntemlerinin çıkış görüntülerinin adil karşılaştırılabilmesi amacıyla hedef platformun kaynaklarının yetebileceği RTL sentezleri gerçekleştirilmiştir. Bu doğrultuda xfOpenCV uygulaması için üretilen IP blok sadece tek renk kanalının çözünürlüğünü iyileştirecek şekilde optimize edilmiştir.

Farklı çözünürlük yöntemlerinin çıkış görüntüleri değerlendirildiğinde bikübik enterpolasyon yönteminin giriş görüntüsünü bulanık hale getirdiği söylenebilir. xfOpenCV uygulamasının çıkış görüntüleri, sadece mavi renk kanalının çözünürlüğünü iyileştirebilmiş olsa da bikübik enterpolasyon yönteminin çıktılarına kıyasla daha az bulanıktır. Buna karşılık önerilen İterasyon-3 HLS tasarımının çıkış görüntülerinin kontrastı daha yüksektir. Ayrıca bu tasarım, giriş görüntüleri üzerinde dalgalanma etkisini (ringing effect) andıran bir sonuç doğurmuştur.

Alzubaidi ve diğerlerinin (2023) oluşturduğu veri setinden alınan bu dört görüntü üzerinde çalışan farklı çözünürlük iyileştirme yöntemlerinin çıkış görüntülerinin her biri için ayrı ayrı PSNR ve SSIM metrik değerleri hesaplanmıştır. Çizelge 4.5; her bir çözünürlük iyileştirme yönteminin çıkış görüntülerindeki ortalama PSNR ve SSIM metrik değerini göstermektedir. Değerler incelendiğinde önerilen HLS tasarımının diğer iki yönteme kıyasla daha iyi performans gösterdiği söylenebilir.

Çizelge 4.5. Farklı çözünürlük iyileştirme yöntemlerinin ultrason görüntüleri üzerindeki performans metrikleri ortalamaları

Yöntem	Mutlak PSNR (dB)	SSIM
Bikübik Enterpolasyon	21,2247	0,248
xfOpenCV Uygulaması	14,0121	0,811
Önerilen Tasarım (İterasyon-3)	19,4309	0,834

4.5.2. Kaynak kullanımı ve performans analizi

Her iki uygulama da Vitis 2025.1 geliştirme ortamı kullanılarak ve Xilinx Zynq UltraScale+ XCZU7EV-2FBVB900I platformu merkezinde oluşturulmuştur. Çizelge 4.6, bu iki tasarım için kaynak kullanımı ve gecikme süresi (latency) parametrelerinin bir karşılaştırmasını sunmaktadır.

Çizelge 4.6. xfOpenCV ve önerilen tasarımının kaynak kullanımı ve gecikme verileri

Kaynak / Metrik	xfOpenCV Uygulaması		Önerilen Tasarım		Değişim
BRAM (18K)	657	%105	385	%61	Kullanımı azaldı.
DSP	219	%12	523	%30	Daha etkin kullanıldı.
Flip-Flop (FF)	97 316	%21	86 331	%18	Kullanımı azaldı.
LUT	325 672	%141	64 308	%27	Kullanımı azaldı.
Gecikme (Latency) (µs)	~12 612		~969		~13x Hızlanma

Sentez sonuçlarının analizi, xfOpenCV uygulamasının, mantık elemanları (LUT) ve bellek blokları (BRAM) açısından hedef FPGA'nın kaynak sınırlarını aştığını ortaya koymaktadır. xfOpenCV uygulaması, LUT kapasitesinin %141'ini ve BRAM kapasitesinin %105'ini kullandığından bu platformda uygulanması mümkün değildir. Öte yandan, Vitis Vision kütüphanesi fonksiyonlarıyla oluşturulan tasarımda kullanılan yeniden boyutlandırmalı konvolüsyon yöntemi kaynak kullanımını önemli ölçüde azaltmaktadır. LUT kullanımı %141'den %27'ye, BRAM tüketimi %105'ten %61'e, Flip-Flop kullanımı ise %21'den %18'e düşmüştür. Tasarımdaki DSP kullanımındaki artış (%12'den %30'a), hesaplamaların mantık kapıları yerine FPGA aritmetik bloklarına (DSP) kaydırıldığını ve bunun sonucunda donanım verimliliğinin arttığını göstermektedir. Ayrıca, önerilen tasarımın gecikme süresi, referans tasarıma kıyasla yaklaşık 13 kat azalmıştır (~12 612 µs'ye kıyasla ~969 µs).

Çizelge 4.7; literatürde incelenen çalışmaların, tek renk kanalını işleyerek çıktı üreten xfOpenCV uygulamasının ve önerilen HLS tasarımının kaynak kullanım değerlerini göstermektedir.

Çizelge 4.7. Çeşitli çalışmalar ve önerilen tasarımının çıktılarının kaynak kullanımları

Çalışma	DSP kullanımı	Flip-Flop (FF) kullanımı	LUT kullanımı
Spagnolo ve diğerleri (2023)	1 750	81 791	28 080
Shi ve diğerleri (2023)	1 064	-	153 356
Su ve diğerleri (2024)	-	70 154	64 015
Amin ve Obermaisser (2025)	30	-	3 110
Joseph ve Jayakumar (2024)	1 114	66 000	55 000
Saidani ve diğerleri (2024)	19	56 141	42 025
Yanamala ve Pullakandam (2025)	109	62 867	67 864
URL-2	219	97 316	325 672
URL-3	210	44 284	42 806
Önerilen Tasarım (İterasyon-3)	523	86 331	64 308

Çeşitli çalışmaların Çizelge 4.7'deki kaynak kullanım miktarları değerlendirilirken her bir çalışma için farklı boyutlu bir giriş görüntüsü ile çalışıldığı dikkate alınmalıdır. Değerler incelendiğinde, önerilen tasarımın (İterasyon-3) işlem gücü ile alan verimliliği arasında optimum bir denge kurduğu görülmektedir. Önerilen mimarinin 523 olan DSP kullanımı, Spagnolo ve diğerleri (2023) ile Joseph ve Jayakumar (2024) tarafından sunulan ve yoğun aritmetik tüketen tasarımlardan belirgin şekilde düşüktür. 86 331 olan FF kullanımı, tasarımın yazmaç yoğunluklu bir yapıya sahip olduğunu gösterse de; 64 308 olan LUT tüketimi, Shi ve diğerlerinin (2023) yanı sıra AMIQ Consulting'in URL-2'deki çalışmasında görülen aşırı mantık yükünün başarıyla önüne geçildiğini göstermektedir. Sonuç olarak önerilen HLS tasarımı, Su ve diğerleri (2024) ile Yanamala ve Pullakandam (2025) gibi güncel çalışmalarla rekabetçi bir çizgide kalarak uç noktalardaki alan maliyetlerinden arındırılmış optimize bir donanım çözümü sunmaktadır.

5. TARTIŞMA

Bu çalışma kapsamında gerçekleştirilen FSRCNN tabanlı çözünürlük iyileştirme mimarisinin FPGA üzerinde yeniden tasarlanması, literatürde süper çözünürlük probleminin hem algoritmik hem de donanımsal yönlerine ilişkin süregelen tartışmalarla doğrudan ilintilidir. Süper çözünürlük modellerinin gömülü sistemlerdeki uygulanabilirliği hâlen açık bir araştırma alanı olup özellikle düşük gecikme, sınırlı hesaplama kapasitesi ve enerji verimliliği gibi tasarım kısıtları bu konudaki kararı belirlemektedir. Bu çalışma, söz konusu kısıtların FSRCNN üzerinde nasıl yeniden yorumlanabileceğini ve modern FPGA mimarileri için nasıl optimize edilebileceğini analiz ederek literatüre hem teorik hem de uygulamalı düzeyde bir katkı sunmaktadır.

Model seçimi açısından değerlendirildiğinde FSRCNN'nin tercih edilmesi daha önce tamamlanmış çalışmalarla güçlü bir paralellik sergilemektedir. Dong ve diğerlerinin (2016) FSRCNN mimarisi, düşük parametre sayısı ve hesaplama yoğunluğu ile gömülü donanımlara uygun modeller arasında sıklıkla anılmaktadır. Bu yapı, modern derin SÇ modellerinde görülen çok katmanlı ve geniş filtre yapılarının aksine FPGA üzerinde ardışık veri akışı ile uyumlu bir işlem düzeni sağlamaktadır. Bu yönüyle seçilen model hem literatürle uyumludur hem de FPGA tabanlı hızlandırıcı tasarımlarında pratik uygulanabilirliği önceleyen güncel eğilimlerle örtüşmektedir. Buna karşın çözünürlük iyileştirme alanında geliştirilmiş olan HLS projeleri tarandığında hâlen Vivado HLS platformunda xfOpenCV kütüphanesi temelli bir proje (URL-2) dışında bir HLS projesine rastlanmamıştır. Bu tasarım eski akışlara bağımlı kaldığı için Vitis Vision kütüphanesini merkeze alan bir çalışma ile FSRCNN modelinin tekrar ele alınmasına ihtiyaç duyulmaktadır.

Büyütme katmanının tasarımı bağlamında bu çalışma, literatürde uzun süredir bilinen bir yapısal sınırlılığı pratik düzeyde doğrulamıştır. URL-1'deki çalışma, ters konvolüsyon yönteminin çıkış görsellerini incelediğinde dengesiz örtüşmeye maruz bırakılan piksellerde periyodik artefaktlar oluştuğunu göstermiştir. Bu çalışmada FSRCNN'nin ilk iterasyonunda gözlemlenen örgü benzeri bozulmalar, bu kuramsal problemin uygulamadaki karşılığını oluşturmaktadır. Daha önce yapılmış çalışmalarla paralellik gösteren bu durum, ters konvolüsyon yönteminin FPGA üzerinde hem hesaplama

karmaşıklığı hem de örnekleme geometrisi açısından istikrarsız bir büyütme yöntemi olduğunu bir kez daha ortaya koymuştur.

Ters konvolüsyon yönteminin yapısal sınırlılıklarını gidermek amacıyla uygulanan yeniden boyutlandırmalı konvolüsyon yöntemi, büyütme işleminin uzamsal örnekleme adımlarını ayrıştırarak daha kararlı bir hesaplama geometrisi oluşturmakta ve bu sayede artefakt oluşumunu önlemektedir. Bu çalışmada kullanılan iki aşamalı yapı hem yüksek frekans bileşenlerinin daha düzenli bir biçimde yeniden oluşturulmasını sağlamış hem de FPGA üzerinde yürütülen operasyonların hesaplama yoğunluğunu önemli ölçüde azaltmıştır. Elde edilen görsel çıktılar, Sugawara ve diğerlerinin (2019) teorik düzeyde ortaya koyduğu düzensiz örnekleme sorunlarının uygulamada nasıl gerçekleştiğini doğrulamakta ve yeniden boyutlandırmalı konvolüsyon yönteminin bu sorunu donanım açısından uygulanabilir bir yöntemle ortadan kaldırdığını göstermektedir. Bu sebeple yapılan çalışma, hem artefaktların giderilmesine yönelik literatürdeki yaklaşımlarla paralellik göstermiş hem de bu yaklaşımların FPGA'ya özgü tasarım ilkeleriyle bütünleştirilmiş bir örneğini sunmuştur.

Ayrıca HLS tasarımı boyunca algoritmik iyileştirmeler ve veri akışı optimizasyonları, bu çalışmanın ayrı bir boyutunu ifade etmektedir. FSRCNN'nin genişletme katmanından sonra 56 kanallı öznitelik haritasının tek kanala indirgenmesi büyütme aşamasındaki işlemleri sadeleştirmiştir ve bu sayede hesaplama maliyetini düşürmüştür. Bu tasarım kararı, öğrenme tabanlı SÇ literatüründe yaygın kabul gören Y-kanalı işleme yaklaşımıyla da uyum içindedir. Chang ve diğerlerinin (2024), parlaklık bileşeni üzerinden gerçekleştirdikleri öğrenme ile çok kanallı modellere kıyasla PSNR performansını koruyarak hesaplama maliyetini önemli ölçüde azalttıkları anlaşılmaktadır. Bu çalışmada FSRCNN'nin genişletme katmanından çıkan 56 kanallı haritanın 1x1 konvolüsyonla tek kanala indirgenmesi ve büyütme işleminin yalnızca bu bileşen üzerinde yürütülmesi, literatürdeki bu bulgularla doğrudan uyumludur. Böylece bu çalışma, hem renk uzayı seçimlerinin SÇ performansına etkisine ilişkin tartışmaya katkı sunmuş hem de bu seçimin donanımsal açıdan nasıl avantaj sağladığını somut olarak göstermiştir.

HLS tasarımında pragma direktiflerinin rolü, bu tartışmanın önemli bir boyutunu oluşturmaktadır. Pipeline, unroll ve dataflow gibi pragma direktifleri teorik olarak sistemin gecikmesini azaltmak ve toplam veri işleme verimini artırmak için güçlü olanaklar sunsa

da bu iyileştirmelerin FPGA kaynak tüketimi üzerinde doğrudan bir maliyeti bulunmaktadır. Bu çalışma boyunca yapılan denemelerde, özellikle ters konvolüsyon yöntemini benimseyen ve xfOpenCV kütüphanesi temelli oluşturulan tasarım (URL-2) üzerinde pipeline, unroll ve dataflow pragma direktifleri uygulandığında gecikmenin kısaldığı ancak LUT ve BRAM kullanımının sistemin kapasitesini aşacak seviyelere çıktığı gözlemlenmiştir. Sestito ve diğerlerinin (2023) benzer hızlandırıcı mimarileri üzerine yaptığı çalışmada da belirtildiği gibi döngü paralelleştirme teknikleri performans kazanımını çoğu zaman kaynak kullanımındaki belirgin artışla birlikte getirmektedir. Buna karşın bu çalışmada uygulanan yeniden boyutlandırmalı konvolüsyon yöntemi hesaplama akışını sadeleştirdiği için yalnızca gerekli noktalarda sınırlı pragma direktifi kullanımıyla hem düşük gecikme değerleri elde edilmiş hem de tasarım hedef SoC'nin kaynak sınırları içinde kalmıştır. Bu sonuç, mimari karmaşıklığın azaltılmasının donanım verimliliği üzerinde çoğu zaman en etkili strateji olduğunu göstermektedir.

Sonuç olarak, bu çalışma literatürde süper çözünürlük modellerinin FPGA üzerinde gerçekleştirilmesine yönelik çabalara yöntemsel ve yapısal açıdan anlamlı bir katkı sunmaktadır. FSRCNN mimarisinin ters konvolüsyon tabanlı sınırlılıkları teorik ve deneysel olarak doğrulanmış, bunun yerine entegre edilen yeniden boyutlandırmalı konvolüsyon yönteminin hem görsel kaliteyi artırdığı hem de FPGA kaynaklarını daha verimli kullandığı ortaya konmuştur.

6. SONUÇ VE ÖNERİLER

Bu çalışma, çözünürlük iyileştirme algoritmalarının FPGA tabanlı donanım mimarilerine uyarlanması sürecinde ortaya çıkan yapısal ve hesaplamalı sorunları inceleyerek bu sorunlara yönelik bütüncül bir çözüm geliştirmeyi amaçlamıştır. Süper çözünürlük modellerinin gömülü sistemlerde etkin biçimde uygulanmasını zorlaştıran başlıca iki unsur, yani yüksek hesaplama yoğunluğu ve büyütme aşamasında ortaya çıkan yapısal artefaktlar, sistematik biçimde incelenmiştir. Bu bağlamda çalışmanın çıkış noktası, literatürde yer alan klasik FPGA uygulamalarının modern gereksinimleri karşılamada yetersiz kalmasıdır. Özellikle AMIQ Consulting tarafından sunulan FSRCNN HLS tasarımının (URL-2) yalnızca mavi renk kanalında işlem yapması, artık AMD tarafından desteklenmeyen xfOpenCV kütüphanesine bağımlı olması ve pragma direktiflerinin sınırlı kullanımı nedeniyle yüksek gecikme ve düşük paralellik sergilemesi, mevcut yaklaşımın hem işlevsel hem de yapısal düzeyde modern FPGA ekosistemine uygun olmadığını göstermektedir. Bu çalışma, söz konusu tasarımın yalnızca yeniden uygulanması değil, aynı zamanda mimari açıdan değişiklikler yapılarak ve güncel Vitis Vision kütüphanesi fonksiyonlarıyla modernize edilerek Vitis 2025.1 geliştirme ortamı kullanılarak optimize edilmesini amaçlamıştır.

Ters konvolüsyon tabanlı büyütme işleminin FPGA üzerinde düzenli örüntü bozulmalarına yol açtığı gözlemi, bu çalışmanın literatürle uyumlu en belirgin bulgularındandır. URL-1'de ayrıntılı biçimde açıklanan dengesiz örtüşme durumu, çalışma kapsamında yapılan ilk iterasyon deneyleriyle doğrulanmış ve örgü benzeri düzenli bozulmaların bulunduğu bir çıkış görseli gözlenmiştir. Bu doğrultuda benimsenen yeniden boyutlandırmalı konvolüsyon yöntemi sayesinde oluşturulan ikinci iterasyon tasarım kaynak kodu ve beraberinde kullanılan testbench kodu ile büyütme aşaması daha kararlı bir örnekleme geometrisi üzerinden yeniden tanımlanarak artefakt oluşumunun önüne geçilmiştir. İkinci iterasyon HLS tasarımının çıktısının yalnızca algısal kaliteyi yükseltmediği, aynı zamanda FPGA üzerinde daha düşük kaynak tüketen bir planlama sunduğu yapılan karşılaştırmalarla doğrulanmıştır. Yeniden boyutlandırmalı konvolüsyon yönteminin hesaplama yapısının sade olması, HLS optimizasyonlarının daha kontrollü kullanılmasına da olanak tanımış ve bu sayede hem gecikme süreleri hem de kaynak kullanımı açısından dengeli bir tasarım elde edilmiştir.

HLS tasarımında pragma direktiflerinin rolünün incelenmesi, bu çalışmanın önemli çıktılarından biridir. Özellikle pipeline, unroll ve dataflow pragma direktiflerinin ters konvolüsyon tabanlı xfOpenCV uygulaması üzerinde gecikmeyi düşürürken kaynak tüketimini hızla artırdığı gözlemlenmiştir. Karşılaşılan bu durum, literatürde döngü paralelleştirmenin ikili etkisini vurgulayan çalışmalarla da uyumludur. Buna karşılık bu çalışmada uygulanan yeniden boyutlandırmalı konvolüsyon yöntemi, algoritmik sadeliği sayesinde yalnızca gerekli noktalarda sınırlı pragma optimizasyonu gerektirmiştir. Böylece hedef SoC'nin kaynak sınırları aşılmadan düşük gecikme elde edilebilmiştir. Bu durum, HLS projelerinde optimizasyonlar ancak algoritmik düzenlemelerle desteklenirse ortaya verimli bir tasarım çıkarılabileceğini ortaya koymaktadır.

Renk uzayı seçiminin SÇ modellerinin donanımsal uygulanabilirliği açısından kritik olduğu da çalışma kapsamında doğrulanmıştır. Yalnızca parlaklık (Y) bileşeni üzerinde işlem yapılarak elde edilen çıktılar, gerek hesaplama maliyetinin düşürülmesi gerekse PSNR performansının korunması açısından literatürde önerilen yöntemlerle (Chang ve diğerleri, 2024) uyumluluk göstermektedir. Renk uzayı tercihinin etkisiyle genişletme katmanından çıkan 56 kanallı öznitelik haritası 1x1 konvolüsyon işlemi ile tek kanala indirgenmiş ve dolayısıyla görsel büyütme işlemleri sadeleştirmiştir. Bu sayede hesaplama maliyetini düşürülmüş olup FPGA kaynaklarının verimli kullanılması sağlanmıştır.

Teorik açıdan bakıldığında bu çalışma, süper çözünürlük modellerinin büyütme aşamasındaki yapısal sorunlarının donanım bağlamındaki karşılıklarını açık biçimde ortaya koymuştur. Uygulamalı açıdan ise Vitis Vision kütüphanesi ile uyumlu, yeniden üretilebilir bir tasarım akışı sunmuştur. Bununla birlikte, bu araştırmanın bazı sınırlılıkları mevcuttur. Çalışma sonunda elde edilen tasarım, RTL sentezinden sonra C/RTL kosimülasyon aşamasında doğrulanmış ve kullanılabilir bir IP blok olarak üretilmiş olsa da herhangi bir FPGA kartına gömülerek sistem seviyesinde test edilmemiştir. Bu durum, gecikme süresi, kaynak kullanımı ve güç tüketimi gibi metriklere bağlı performans ölçümlerinin sistem düzeyinde doğrulanmasını engellemekte ve gelecekte yapılması gerekli bir çalışma alanı oluşturmaktadır. Ayrıca tasarım 2x büyütme oranıyla sınırlıdır, daha yüksek ölçekleme faktörleri için mimari yapılandırma veya yeniden eğitilecek bir model gerekmektedir.

Bu çalışma sonucunda elde edilen çıktılar yalnızca FSRCNN bağlamında değil, ters konvolüsyon tabanlı tüm SÇ modellerinin FPGA üzerinde yeniden değerlendirilmesi gerekliliğini de ortaya koymaktadır. Genel olarak bu çalışma, çözünürlük iyileştirme probleminin donanımsal boyutuna hem kuramsal hem de pratik katkılar sunmuş olup süper çözünürlük modellerinin modern FPGA platformlarında uygulanabilirliğine yönelik bütüncül bir çözüm ortaya koymuştur.

Gelecek araştırmalarda, farklı büyütme stratejilerinin donanımsal verimlilik açısından karşılaştırılması, gelişmiş enterpolasyon tekniklerinin FPGA'ya uyarlanması, ESPCN gibi diğer modellerin Vitis Vision kütüphanesiyle yeniden yapılandırılması ve nihayetinde tasarlanan IP bloklarının gerçek FPGA kartları üzerinde kontrollü performans ölçümleriyle doğrulanması potansiyel araştırma alanları olarak öne çıkmaktadır.

KAYNAKLAR

Aarrestad, T.K., Loncar, V., Ghielmetti, N., Pierini, M., Summers, S.P., Ngadiuba, J., Petersson, C., Linander, H., Iiyama, Y., Di Guglielmo, G., Duarte, J.M., Harris, P.C., Rankin, D.S., Jindariani, S., Pedro, K., Tran, N.V., Liu, M., Kreinar, E., Wu, Z., and Hoang, D. (2021). Fast convolutional neural networks on FPGAs with hls4ml. *Machine Learning: Science and Technology*, 2(4), 045015.

Ahmed, O., Ali, H., Javed, Y., and Iqbal, N. (2024, 14-19 Temmuz). *Super-resolution CNN accelerator on low-cost Zynq-7000 series SoC-based FPGA*. 2024 IEEE INC-USNC-URSI Radio Science Meeting (Joint with AP-S Symposium), Floransa, 257-258.

Alzubaidi, M., Agus, M., Makhlouf, M., Anver, F., Alyafei, K., and Househ, M. (2023). Large-scale annotation dataset for fetal head biometry in ultrasound images. *Data in Brief*, 109708.

Amin, R. A., and Obermaisser, R. (2025). BIDSRS+: Resource efficient reconfigurable real time bidirectional super resolution system for FPGAs. *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, 15(1), 120-132.

Chang, C.-C., Chen, W.-P., Lin, Y.-W., Lin, Y.-J., and Pan, P.-J. (2024, 15-17 Kasım). *Real-time super resolution utilizing dilation and depthwise separable convolution*. 2024 IEEE 6th Eurasia Conference on IoT, Communication and Engineering, Yunlin, 1-7.

Dong, C., Loy, C. C., He, K., and Tang, X. (2014, 6-12 Eylül). *Learning a deep convolutional network for image super-resolution*. ECCV 2014, Zürih, 184-199.

Dong, C., Loy, C. C., and Tang, X. (2016, 11-14 Ekim). *Accelerating the super-resolution convolutional neural network*. ECCV 2016, Amsterdam, 391-407.

Gao, L., Luo, Z., and Wang, L. (2025). Convolutional neural network acceleration techniques based on FPGA platforms: Principles, methods, and challenges. *Information*, 16(10), 914.

İnternet: Odena, A., Dumoulin, and V., Olah, C. (2016). Deconvolution and checkerboard artifacts. *Distill*. URL-1: https://distill.pub/2016/deconv-checkerboard/ Son Erişim Tarihi: 16.06.2025.

İnternet: Leuștean, I. (2021). How to accelerate an image upscaling CNN on FPGA using HLS. *AMIQ Consulting Blog*. URL-2: https://www.consulting.amiq.com/2021/08/23/how-to-accelerate-an-image-upscaling-cnn-on-fpga-using-hls/ Son Erişim Tarihi: 10.10.2025.

İnternet: Wan, R., Zheng, Q., Zhang, R., Chen, B., Liu, J., Li, M., Jing, M., Zhou, J., and Fan, Y. (2025). GRACE: Designing generative face video CODEC via agile hardware-centric workflow. *arXiv*. URL-3: https://arxiv.org/abs/2511.09272 Son Erişim Tarihi: 10.11.2025.

İnternet: Gaines, R., and Morris, K. (November, 2023). Create and Run a Vitis Vision Library Example on Windows 10. *AMD Adaptive Computing Knowledge Base*. URL-4: https://adaptivesupport.amd.com/s/article/000035905 Son Erişim Tarihi: 12.05.2025.

İnternet: Gaines, R., and Morris, K. (2023). Install/Setup OpenCV on Windows 10. *AMD Adaptive Computing Knowledge Base*. URL-5: https://adaptivesupport.amd.com/s/article/000035890 Son Erişim Tarihi: 12.03.2025.

Joseph, G., and Jayakumar, E. P. (2024, 1-3 Eylül). *Quantized neural network architecture for hardware efficient real-time 4K image super-resolution*. 2024 28th International Symposium on VLSI Design and Test (VDAT), Vellore, 1-5.

Manabe, T., Shibata, Y., and Oguri, K. (2016, 7-9 Aralık). *FPGA implementation of a real-time super-resolution system using a convolutional neural network*. 2016 International Conference on Field-Programmable Technology (FPT), Xi'an, 249-252.

Saidani, T., Ghodhbani, R., Alhomoud, A., Alshammari, A., Zayani, H., and Ammar, M. B. (2024). Hardware acceleration for object detection using YOLOv5 deep learning algorithm on Xilinx Zynq FPGA platform. *Engineering Technology & Applied Science Research*, 14(1), 13066-13071.

Sestito, C., Perri, S., and Stewart, R. (2023). FPGA design of transposed convolutions for deep learning using high-level synthesis. *Journal of Signal Processing Systems*, 95, 1245-1263.

Shi, B., Zhang, J., He, Z., Wei, X., Li, S., Luo, G., Zheng, H., and Xie, Y. (2023). Efficient super-resolution system with block-wise hybridization and quantized winograd on FPGA. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 42(11), 3910-3924.

Shi, W., Caballero, J., Huszár, F., Totz, J., Aitken, A. P., Bishop, Rueckert, D., and Wang, Z. (2016, 27-30 Haziran). *Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network*. CVPR 2016, Las Vegas, 1874-1883.

Spagnolo, F., Corsonello, P., Frustaci, F., and Perri, S. (2023). Design of a low-power super-resolution architecture for virtual reality wearable devices. *IEEE Sensors Journal*, 23(8), 9009-9016.

Su, Y., Seng, K. P., Smith, J., and Ang, L. M. (2024). Efficient FPGA binary neural network architecture for image super-resolution. *Electronics*, 13(2), 266.

Sugawara, Y., Shiota, S., and Kiya, H. (2019). Checkerboard artifacts free convolutional neural networks. *APSIPA Transactions on Signal and Information Processing*, 8, e9.

Yanamala, R. M. R., and Pullakandam, M. (2025). FPGA-accelerated hybrid CNN-LSTM system for efficient EEG-based drowsiness recognition. *The Journal of Supercomputing*, 81, 453.



Gazili olmak ayrıcalıktır