
GAZİ ÜNİVERSİTESİ
Gazi
1926

UYDU GÖRÜNTÜLERİNDE DERİN ÖĞRENME İLE SÜPER ÇÖZÜNÜRLÜK

Gamze DANIŞMAN

YÜKSEK LİSANS TEZİ
ELEKTRİK-ELEKTRONİK MÜHENDİSLİĞİ ANA BİLİM DALI

GAZİ ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

KASIM 2023

ETİK BEYAN

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

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

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

Gamze DANIŞMAN

12/11/2023

UYDU GÖRÜNTÜLERİNDE DERİN ÖĞRENME İLE SÜPER ÇÖZÜNÜRLÜK

(Yüksek Lisans Tezi)

Gamze DANIŞMAN

GAZİ ÜNİVERSİTESİ

FEN BİLİMLERİ ENSTİTÜSÜ

Kasım 2023

ÖZET

Yüksek çözünürlüklü görüntü, uzaktan nesne algılama kapsamında ihtiyaç haline gelmiştir. Yüksek çözünürlüklü görüntü çeşitli uygulamalarda kritik olabilecek daha fazla ayrıntıyı içermektedir. Çoğu uydu aynı sahneye ait yüksek çözünürlüklü pankromatik ve daha düşük çözünürlüklü multispektral görüntü sunabilmektedir. Araştırmacılar multispektral görüntülerin çözünürlüğünün arttırılması sürecinde, pankromatik görüntünün içerdiği bilgilerden yararlanarak çeşitli geleneksel pankeskinleştirme yöntemi sunmuşlardır. Fakat bu teknikler önemli ölçüde spektral ve uzamsal bozulmadan mustariptir. Bu dezavantajların üstesinden gelebilmek adına günümüzün trendi olan derin öğrenme mimarilerine bu problemin araştırılması sunulmuştur. Bu kapsamda birçok derin öğrenme tabanlı pankeskinleştirme yöntemi tasarlanmıştır. Gerçekleştirdiğimiz çalışmada hem geleneksel hem de derin öğrenme tabanlı pankeskinleştirme yöntemlerini uygulayarak en iyi sonucu elde eden modeli araştırılmıştır. Uygulamış olduğumuz bu yöntemlerden en iyi sonucu elde ettiğimiz modelin çıktı performansını artırmak adına süper çözünürlük için tasarlanmış artık yoğun ağ katmanını derin öğrenme modeline kazandırılmıştır. Bu yaklaşımımız ile Landsat-8 veri setinde 33,291'den 33,521 PSNR skoruna, WorldView-3 veri setinde 33,534'den 33,544 PSNR skoruna ulaşılmıştır. Sunulan yöntemlerde ağ çıkışının doğruluk sonuçlarına ek olarak modelle ilişkili hesaplama süresinin de önemli bir parametre olduğu fark edilmiştir. Bu kapsamda daha hızlı hesaplama süresi maliyeti sunan FPGA ortamında derin öğrenme tabanlı pankeskinleştirme yöntemlerini sunabilmek adına Xilinx Kria KV260 geliştirme kartı kullanılmıştır. Derin öğrenme tabanlı pankeskinleştirme yöntemlerini FPGA ortamına taşırken Vitis AI aracından faydalanılmıştır. Sonuçlar, hesaplama süresi maliyeti açısından, FPGA'nın CPU ve GPU ortamlarına kıyasla üstün işleme yeteneğini doğrulamaktadır. Önerilen model ile FPGA ortamında Landsat-8 veri setinde 33,331 PSNR skoruna, WorlView-3 veri setinde 31,051 PSNR skoruna ulaşılmıştır. Bu derin öğrenme modeli ile CPU ortamında 0,693 saniyede ve GPU ortamında 0,117 saniyede çıktı elde edilirken, FPGA ortamında 5,601 W güç tüketimine sahip donanım ile 0,053 saniye de model çıktısı elde edilmiştir.

Bilim Kodu	:	90542
Anahtar Kelimeler	:	Uzaktan algılama görüntüsü, pankeskinleştirme, süper çözünürlük, derin öğrenme, FPGA
Sayfa Adedi	:	68
Danışman	:	Prof. Dr. Hasan Şakir BİLGE

SUPER RESOLUTION WITH DEEP LEARNING ON SATELLITE IMAGES

(M. Sc. Thesis)

Gamze DANISMAN

GAZİ UNIVERSITY

GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES

November 2023

ABSTRACT

Images with higher resolutions have become a significant necessity within the scope of remote object detecting. A high-resolution image offers more detail which can be vital for various applications. Most of the satellites can provide high-resolution panchromatic and lower-resolution multispectral images of the same scene. During the process of increasing the resolution of multispectral images, researchers have presented various conventional pansharpening methods by taking advantage of the information contained in the panchromatic image. Nevertheless, mentioned techniques suffer significantly due to spectral and spatial distortions. To overcome these disadvantages, the investigations of this problem has been introduced to trending deep learning architectures. In scope of this situation, many deep learning-based pansharpening methods have been proposed. In this study, the model that achieved the best results by applying both conventional and deep learning-pansharpening methods was investigated. To increase the output performance of the model in which we achieved the best results from these methods we applied, we added the intense network layer designed for super resolution to the deep learning model. As the result of this approach, we obtained a PSNR score from 33,291 to 33,521 in the Landsat-8 dataset and a PSNR score from 33,534 to 33,544 in the WorldView-3 dataset. It has been noticed that in the proposed methods, in addition to the accuracy results of the network output, the computation time associated with the model is also an important parameter. In this context, Xilinx Kria KV260 development card was used to offer deep learning-based pan-sharpening methods in the FPGA environment that is capable of offering faster computation time costs faster. Vitis AI tool was used when transferring deep learning-based pansharpening methods to the FPGA environment. The results confirm the superior processing ability of FPGA compared to CPU and GPU environments in terms of computation time cost. With the proposed model, a PSNR score of 33.331 and 31.051 were achieved in the FPGA environment based on Landsat-8 and WorldView-3 datasets respectively. In addition, output was obtained in 0.693 seconds in the CPU environment and 0.117 seconds in the GPU environment, while model output was obtained in 0.053 seconds in the FPGA environment with hardware with a power consumption of 5.601 W.

Science Code	:	90542
Key Words	:	Remote sensing image, pansharpening, super resolution, deep learning, FPGA
Page Number	:	68
Supervisor	:	Prof. Dr. Hasan Şakir BİLGE

TEŞEKKÜR

Yüksek lisans eğitimimde ve tez çalışma süreci boyunca bana destek olan ve yardım eden akademik danışmanım sayın Prof. Dr. Hasan Şakir BİLGE'ye, yüksek lisans eğitimim boyunca üzerimde emeği olan bölüm akademisyenlerine, hayatımda karşılaştığım her zorlukta bana destek olan biricik annem Aynur ÖZCAN ve kardeşim Gaye DANIŞMAN'a, ve bu süreçte desteklerini her zaman hissettiğim kıymetli arkadaşlarıma teşekkürü borç bilirim.

Yüksek lisans eğitimim boyunca, TUSAŞ Bilimsel Araştırma Programı ile desteklerini esirgemeyen TUSAŞ' a sonsuz teşekkür ederim.

İÇİNDEKİLER

Sayfa

ÖZET iv

ABSTRACT v

TEŞEKKÜR vi

İÇİNDEKİLER vii

ÇİZELGELERİN LİSTESİ x

ŞEKİLLERİN LİSTESİ xi

RESİMLERİN LİSTESİ xii

SİMGELER VE KISALTMALAR xiii

1. GİRİŞ 1

2. MATERYAL VE YÖNTEM 11

2.1. Bikübik Enterpolasyon 11

2.2. Gaussian Filtresi 11

2.3. Veri Seti 12

2.4. Geleneksel Pankeskinleştirme Yöntemleri 13

2.5. Bileşen Değiştirme Pankeskinleştirme Yöntemleri 14

2.5.1. Brovey dönüşümü 14

2.5.2. Yoğunluk renk doyum dönüşümü 15

2.5.3. Temel bileşen analizi 16

2.5.4. Gram-Schmidt dönüşümü 16

2.5.5. Kılavuzlu filtre temel bileşen analizi 17

2.5.6. Adaptif Gram-Schmidt 17

2.6. Çoklu Çözünürlük Analizi Pankeskinleştirme Yöntemleri 18

2.6.1. Yumuşatma filtresi tabanlı yoğunluk modülasyonu 19

Sayfa

2.6.2. Modülasyon transfer fonksiyonu bazlı genelleştirilmiş laplasyen piramitleri 19

2.6.3. Yüksek geçiren filtre ile modülasyon transfer fonksiyonu bazlı genellestirilmiş laplasyen piramitleri 19

2.6.4. Dalgacık dönüşümleri 20

2.7. Derin Öğrenme Tabanlı Pankeskinleştirme 20

2.7.1. Yapay zekâ temelleri 20

2.7.2. Yapay sinir ağları 21

2.7.3. Aktivasyon fonksiyonları 22

2.7.4. Evrişimli sinir ağları 23

2.7.5. Artık ağ 24

2.7.6. Yoğun evrişimli ağ 25

2.7.7. Süper çözünürlüklü evrişimsel sinir ağları 25

2.7.8. Süper çözünürlük için RDN 26

2.7.9. PNN 27

2.7.10. PanNet 28

2.7.11. MSDCNN 29

2.7.12. SRPPNN 31

2.8. Vitis AI 32

2.8.1. Derin öğrenme işlem birimi 36

2.8.2. Vivado tasarımı 38

2.8.3. Petalinux kurulumu 39

2.8.4. Modelin FPGA ortamına entegrasyonu 40

3. DENEYSEL SONUÇLAR 45

3.1. Metrikler 45

Sayfa

3.1.1. PSNR ve MSE 45
3.1.2. RMSE 46
3.1.3. SSIM 46
3.1.4. ERGAS 46
3.1.5. CC 47
3.1.6. UIQI 47
3.1.7. BRISQUE 48
3.2. Deneysel Sonuçlar 48
3.3. Yöntemlerin Karşılaştırılması 48
4. TARTIŞMA 57
5. SONUÇ VE ÖNERİLER 63
KAYNAKLAR 65
ÖZGEÇMİŞ 68

ÇİZELGELERİN LİSTESİ

Çizelge **Sayfa**

Çizelge 2.1. WorldView-3 veri seti 13

Çizelge 2.2. PNN modeline ait özellikler 28

Çizelge 2.3. PanNet modeline ait özellikler 29

Çizelge 2.4. MSDCNN modeline ait özellikler 31

Çizelge 2.5. SRPPNN modelinin özellikleri 34

Çizelge 2.6. DPU IP mimarileri 37

Çizelge 2.7. DPU mimarileri özellikleri 37

Çizelge 2.8. Model eğitimi için kullanılan hiperparametreler 41

Çizelge 3.1. Landsat-8 veri setinde geleneksel pankeskinleştirme yöntemlerinin sonuçları 53

Çizelge 3.2. Landsat-8 veri setinde derin öğrenme yöntemlerinin sonuçları 53

Çizelge 3.3. WorldView-3 veri setinde geleneksel pankeskinleştirme yöntemlerinin sonuçları 54

Çizelge 3.4. WorldView-3 veri setinde derin öğrenme yöntemlerinin sonuçları 54

Çizelge 3.5. Yöntemlerin test süreleri 55

ŞEKİLLERİN LİSTESİ

Şekil **Sayfa**

Şekil 2.1. Bikübik enterpolasyon komşuluğu 11

Şekil 2.2. WorldView-3 veri setinde düşük çözünürlüklü görüntü elde edimi..... 12

Şekil 2.3. Landsat-8 veri setinde düşük çözünürlüklü görüntü elde edimi..... 13

Şekil 2.4. Dalgacık dönüşümü ile pankeskinleştirme modeli..... 20

Şekil 2.5. Yapay sinir hücresi 21

Şekil 2.6. Yapay sinir ağları..... 21

Şekil 2.7. Sigmoid (mavi eğri) ve tanh (kırmızı eğri) aktivasyon fonksiyonu 22

Şekil 2.8. Relu (mavi eğri) ve leaky relu (kırmızı eğri) aktivasyon fonksiyonu 23

Şekil 2.9. Resnet ağı..... 24

Şekil 2.10. DenseNet ağı..... 25

Şekil 2.11. Artık yoğun ağ modeli..... 26

Şekil 2.12. PNN modeli 27

Şekil 2.13. PanNet modeli 28

Şekil 2.14. MSDCNN modeli..... 30

Şekil 2.15. SRPPNN mimarisi..... 33

Şekil 2.16. Önerilen modelin ilk iki fazı..... 35

Şekil 2.17. Pankeskinleştirme yöntemlerinin FPGA'ya entegrasyon akışı 41

RESİMLERİN LİSTESİ

Resim **Sayfa**

Resim 2.1. Veri setine ait görüntüler 14

Resim 2.2. Model eğitimi ve doğrulama sonuçları 43

Resim 3.1. WorldView-3 veri setinde geleneksel pankeskinleştirme yöntemlerinin sonuçları 49

Resim 3.2. WorldView-3 veri setinde derin öğrenme yöntemlerinin sonuçları 50

Resim 3.3. Landsat-8 veri setinde geleneksel pankeskinleştirme yöntemlerinin sonuçları 51

Resim 3.4. Landsat-8 veri setinde derin öğrenme yöntemlerinin sonuçları 52

SİMGELER VE KISALTMALAR

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

Simgeler	**Açıklamalar**
ε	Epsilon
%	Yüzde
w	Ağırlık katsayısı
μ	Ortalama
σ	Varyans

Kısaltmalar	**Açıklamalar**
AI	Yapay Zekâ
APU	Uygulama İşlem Birimi
BSP	Kart Destek Paketi
CNN	Evrişimsel Sinir Ağları
CPU	Merkezi İşlemci Birimi
CS	Bileşen Değiştirme
DPU	Derin Öğrenme İşlem Birimi
DWT	Ayrık Dalgacık Dönüşümü
ERGAS	Erreur Relative Globale Adimensionnelle de Synthése
FPGA	Sahada Programlanabilir Kapı Dizisi
GFPCA	Kılavuzlu Filtre Temel Bileşen Analizi
GLP	Genelleştirilmiş Laplasyen Piramidi
GPU	Grafik İşlemci Birimi
GS	Gram-Schmidt
GSA	Adaptif Gram-Schmidt
H	Renk Tonu

Kısaltmalar	Açıklamalar
HH	Çapraz
HL	Dikey
HLS	Yüksek Düzey Sentez
HPF	Yüksek Geçiş Filtresi
I	Yoğunluk
IDWT	Ters Dalgacık Dönüşümü
IHS	Yoğunluk Renk Doyum
IP	Fikri Mülkiyet
LH	Yatay
LL	Yaklaşık
MRA	Çoklu Çözünürlük Analizi
MS	Multispektral
MTF	Modülasyon Transfer Fonksiyonu
NLV	Yerel Olmayan Değişken
PAN	Pankromatik
PCA	Temel Bileşen Analizi
PC_1	Birinci Temel Bileşen
PE	İşlem Motoru
PNN	Evrişimli Sinir Ağı ile Pankeskinleştirme
PSNR	Tepe Sinyal-Gürültü Oranı
Q	Evrensel Görüntü Kalitesi İndeksi
QNR	Referanssız İndeksli Kalite
RDN	Artık Yoğun Ağ
RGB	Kırmızı, Yeşil, Mavi
RMSE	Ortalama Karekök Hata
S	Doygunluk
SAM	Spektral Açı Eşleyici
SCC	Uzaysal Korelasyon Katsayısı
SFIM	Yumuşatma Filtresi Tabanlı Yoğunluk Modülasyonu
SGD	Stokastik Gradyan Azalma
SRCNN	Süper Çözünürlüklü Evrişimsel Sinir Ağları

Kısaltmalar	Açıklamalar
SSD	Katı Hal Sürücü
SSIM	Yapısal Benzerlik İndeksi Ölçümü
SVM	Destek Vektör Makineleri
VAI	Vitis Yapay Zekâsı
VHDL	Doğrudan Donanım Tabanlı Kodlama Dili
W	Watt
XRT	Xilinx Çalışma Zamanı

1. GİRİŞ

Uydu görüntülerinin ilk kullanımı askeri amaçlı uygulamalar içindi. Askeri uygulamalı kullanımın yanı sıra navigasyon, şehir planlama, ormancılık, tarım ve jeoloji gibi birçok farklı alanda da kullanılmaktadır. Bu nedenle uydudan alınan görüntülerin taşıdığı bilgiler büyük önem taşımaktadır. Sensörlerin teknik sınırlamaları nedeniyle, yüksek uzaysal ve spektral çözünürlüklü uzaktan algılama görüntülerinin doğrudan elde edilmesi mümkün değildir [1]. Bu sorunu çözmek için araştırmacılar, yüksek çözünürlüklü multispektral (MS) görüntüleri elde etmek amacıyla pankromatik (PAN) bant özelliklerini kullanan çeşitli görüntü işleme ve derin öğrenme yöntemlerini araştırmaya başlamışlardır.

Multispektral görüntülere göre daha yüksek çözünürlüğe sahip olan pankromatik görüntüler yalnızca siyah beyaz keskinlik bilgisi içermektedir. Pankromatik (PAN) görüntünün keskinlik bilgisinden yararlanılarak multispektral görüntünün çözünürlüğünün arttırılmasına pankeskinleştirme yöntemi denir. Bu yöntem uydu görüntülerine değer katmanın bir yoludur. Geleneksel pankeskinleştirme yöntemleri tipik olarak Bileşen Değiştirme (Component Substitution, CS) ve Çoklu Çözünürlük Analizi (Multi-resolution Analysis, MRA)'dir.

Bileşen Değiştirme, MS görüntünün uzaysal bileşenlerinin, pankromatik görüntüye ait uzaysal bileşenler ile yer değiştirildiği yöntemlerdir. Bileşen Değiştirme yaklaşımını benimseyen yöntemler; Brovey yöntemi, Yoğunluk Renk Doyum (Intensity Hue Saturation, IHS) yöntemi, Temel Bileşen Analizi (Principal Component Analysis, PCA), Kılavuzlu Filtre Temel Bileşen Analizi (Guided Filter Principal Component Analysis, GFPCA), Gram-Schmidt (GS) yöntemi ve Adaptif Gram-Schmidt (Adaptive Gram-Schmidt, GSA) yöntemidir. Bileşen değiştirme yöntemleri, yöntem algoritmaları basit ve hesaplama yükü hafif [2] iken, önemli ölçüde spektral bozulmadan muzdariptir.

Çoklu Çözünürlük Analizi, multispektral ile pankromatik görüntünün, uygulanan filtreler veya dönüşümler aracılığıyla birleştirilmesidir. Bu yöntemlerde genellikle Laplasyen Piramidi veya Dalgacık Dönüşümü kullanılmaktadır. Çoklu çözünürlüklü analizinde, pankromatik görüntüden multispektral görüntüye bol miktarda yüksek frekanslı ayrıntı enjekte edilmesine rağmen, görüntünün görsel kalitesini düşüren uzaysal bozulma meydana

gelmektedir. Çoklu çözünürlüklü Analizi yaklaşımını benimseyen yöntemler; Yumuşatma Filtresi Tabanlı Yoğunluk Modülasyonu (Smoothing Filter-based Intensity Modulation, SFIM), Modülasyon Transfer Fonksiyonu Bazlı Laplasyen Piramitleri (Modulation Transfer Function, MTF), Modülasyon Transfer Fonksiyonu Bazlı Genellestirilmiş Laplasyen Piramitleri (Generalized Laplacian Pyramid Based Modulation Transfer Function, MTF GLP) ve Dalgacık Dönüşümleri (Wavelet Transform)'dir.

Bu yöntemlerin yanı sıra yeni nesil pankeskinleştirme yöntemleri de derin öğrenmeye dayanmaktadır. Pankeskinleştirme sorunu için kullanılan derin öğrenme tabanlı yeni nesil yöntem, süper çözünürlüklü derin öğrenme yaklaşımlarından yararlanmaktadır. Derin öğrenme tabanlı pankeskinleştirme yönteminin süper çözünürlük yönteminden en önemli farkı, modelde yüksek kalitede uzaysal bilgi taşıyan pankromatik bant kullanılmasıdır. Literatürde kullanılan yöntemlerden bazıları PNN (Pansharpening by Convolution Neural Network) , PanNet, MSDCNN (Multi-Scale and Multi-Depth Convolutional Neural Network, MSDCNN) ve SRPPNN (Super-Resolution-Guided Progressive Pansharpening Based on a Deep Convolutional Neural Network, SRPPNN)'dir [3-6].

PNN (Pansharpening by Convolution Neural Network, PNN) modeli, Süper Çözünürlüklü Evrişimsel Sinir Ağları (Super-Resolution Convolutional Neural Network, SRCNN) yaklaşımından ilham alan bir pankeskinleştirme tekniğidir. SRCNN (Super-Resolution Convolutional Neural Network) yöntemi düşük çözünürlüklü bir görüntüden yüksek çözünürlüklü görüntü elde eden bir modeldir [7]. SRCNN (Super-Resolution Convolutional Neural Network) modelinde belirli evrişimsel katmanlar kullanılarak düşük çözünürlüklü görüntülerden yüksek çözünürlüklü görüntüler elde edilmeye çalışılır. PNN modelinin SRCNN (Super-Resolution Convolutional Neural Network) modelinden farkı sadece renkli görüntülerin (R, G ve B bantları) değil, pankromatik bantların da model girişinde kullanılmasıdır. Buna ek olarak, model karmaşıklığı artırılmadan pankromatik bandı modele girdi olarak vererek bu bantta yer alan radyometrik indekslerin çeşitli özellik haritalarını girdiye dahil etmişlerdir. Düşük çözünürlüklü MS görüntülerle birlikte pankromatik bant kullanılarak çıkışta pankeskinleştirilmiş bir yüksek çözünürlüklü görüntü elde edilir. Buradaki amaç sadece pankeskinleştirme yöntemi kullanılarak yüksek çözünürlüklü görüntü elde edilmesi değil, ayrıca modeli eğiterek geleneksel pankeskinleştirme yöntemlerinden

daha iyi bir pankeskinleştirme yöntemi elde etmektir. Bu durumu modele eğitmek için referans görüntülerde yüksek çözünürlüklü görüntüler kullanılır.

PanNet yöntemi pankeskinleştirmenin iki amacı olan hem spektral hem de uzaysal özelliklerin korunmasına odaklanmıştır. Uzaysal yapıyı korumak için hem PAN görüntüsünden hem de düşük çözünürlüklü MS görüntüsünden yüksek geçiren filtre çıktıları kullanılmıştır. Spektral yapıyı korumak için, MS görüntüsünün doğrudan çözünürlüğünün yükseltilmiş hali (Yüksek çözünürlüklü referans görüntüsünün çözünürlüğüne ve boyutlarına) ağ çıkışına eklenir. Bu işlem için farklı katmanlar kullanılır.

MSDCNN ve SRPPNN modellerinde ise, MSDCNN modelinde farklı çekirdek boyutları kullanılarak hem MS hem de PAN görüntülerinden özellik çıkarımını içermekte iken, SRPPNN modelinde PAN görüntüsünün özelliklerinin modele aşamalı olarak enjekte edilmesini içerir. Her iki model de oldukça kompleks ve kendi içinde katmanlara ayrılan modellerdir. Bu durumdan ötürü PNN ve PanNet modellerinde eğitilmesi gereken parametre sayısı, MSDCNN ve SRPPNN modellerindeki parametre sayılarından daha az sayıdadır.

Masi ve ark. [3] sundukları PNN modelinde Ikonos, GeoEye-1, ve WorldView-2 uydularından alınan görüntülerle bir veri seti oluşturmuşlardır. Veri setindeki test görüntüleri hariç tüm görüntülerin boyutları 33 x 33 boyutlarındadır. Her bir uydudan eğitim için 14400 görüntü, doğrulama için 7168 görüntü toplanmıştır. Model eğitimi için optimizasyon fonksiyonu olarak Stokastik Gradyan Azalma (Stochastic Gradient Descent, SGD) ($\mu = 0{,}9$ ve $\alpha = 10^{-3}$) kullanılmıştır. Veri seti için toplam iterasyon sayısı 1.12×10^{6}'dır. Test veri seti için ise Ikonos uydusundan 50, GeoEye-1 uydusundan 70 ve WorldView-2 uydusundan 30 test görüntüsü toplanmıştır. Test görüntüleri 320 x 320 boyutlarına sahiptir. Sonuç metrikleri olarak Spektral Açı Eşleyici (Spectral Angle Mapper, SAM), Erreur Relative Globale Adimensionnelle de Synthése (ERGAS), Uzaysal Korelasyon Katsayısı (Spatial Correlation Coefficient, SCC), Evrensel Görüntü Kalitesi İndeksi (Universal Image Quality Index, Q) ve Q değerinin x-bandı uzantısı olmak üzere referans görüntü gerektiren metrikler kullanılmıştır. Bunun yanı sıra Referanssız İndeksli Kalite (Quality with no-Reference Index, QNR) ve QNR değerlerinin spektral ve uzaysal bileşenleri referans görüntü gerektirmeyen görüntü kalite metrikleri olarak kullanılmıştır. En çok kullanılan ölçüm metriklerinden olan Q metriği ile WorldView-2 veri setinde 0,9442 değerini, Ikonos veri setinde 0,9006 değerini ve GeoEye-1 veri setinde 0,9402 değerlerini

elde etmişlerdir. Referanssız görüntü kalite metriği olan QNR değerinde ise WorldView-2 veri seti için 0,9326, Ikonos veri seti için 0,8796 ve GeoEye-1 veri seti için ise 0,9084 değerleri elde edilmiştir. Çalışmalardaki eğitim derin öğrenme çerçevesi olan Caffe aracılığıyla Grafik İşlemci Birimi üzerinden gerçekleşmiştir (CUDA 7.5 ve CuDNN v3 ile Nvidia Tesla K20c). Test çalışması ise 2GHz Intel Xeon işlemcili ve 64 Gb'a sahip Merkezi İşlem Birimi (Central Processing Unit, CPU) aracılığıyla gerçekleştirilmiştir.

PanNet modelini öneren Yang ve ark. [4] WorldView-3 uydusundan 425 görüntü toplamışlardır. Deneyleri için 64 x 64 boyutunda 18000 adet yama (bölge) çıkarmışlardır ve bu yamaları eğitim ve doğrulama veri seti için kullanmışlardır. Bunun %90'ı eğitim için, %10'u ise doğrulama için kullanılmıştır. Veri seti kullanılarak eğitim SGD optimizasyon fonksiyonu kullanılarak gerçekleştirilmiştir. SGD için ağırlık azalması 10^{-7}'ye, momentum 0,9'a ayarlanmıştır. Örnek kümesi boyutu 16 ve öğrenme katsayısı 0,001'lik ile başlayıp, 10^5 ve 2×10^5 yinelemelerde 10'a bölünerek toplamda $2,5 \times 10^5$ iterasyon kullanılmıştır. 425 görüntünün 225 adedi daha düşük ölçekli test değerlerinde 200 adedi ise orijinal ölçekli test değerlerinde kullanılmıştır. Kendi veri setlerini sadece kendi modellerinde değil başka modellerde de kullanarak karşılaştırmalar yapmışlardır. Bir önceki çalışmada bahsedilen Masi ve ark. [3] kullandığı PNN modelinde orijinal ölçekte kullanılan 200 adet WorldView-3 veri setinde 0,88 QNR değeri elde ederlerken, kendi sundukları PanNet modelinde 0,908 QNR değeri elde etmişlerdir. Buna ek olarak, düşük ölçekli 200 adet test görüntüleri için ise PNN modeli 0,882 QNR değeri verirken, kendi modelleri 0,925 QNR değeri vermiştir. Referans görüntü gerektiren metrik olan Q değerinde ise orijinal ölçekli 200 adet görüntüde PNN modeli 0,938 değerini, kendi modelleri 0,972 değerini verirken düşük ölçekli 225 adet görüntülerde ise PNN modeli 0,891 Q değerini ve kendi modelleri 0,928 Q değerlerini vermiştir. Eğitim ve test çalışması GPU donanımı ile gerçekleştirilmiştir. Geleneksel pankeskinleştirme yöntemlerinin aşamaları CPU üzerinde gerçekleştirilmiştir.

Yuan ve ark. [5] pankeskinleştirme için Çok Ölçekli Özellik Çıkarma Bloğu ve Bağlantı Atlama (Multi-Scale and Multi-Depth Convolutional Neural Network, MSDCNN) adlı bir model önermişlerdir. Veri setlerinde Quickbird, IKONOS ve WorldView-2 uydularından alınan görüntüler kullanılmıştır. Quickbird uydusundan elde edilen görüntülerden 41 x 41 x 5 boyutlarında 51648 adet görüntü eğitim fazında, 250 x 250 x 5 boyutlarında 160 görüntü test fazında kullanılmıştır. Worldview-2 uydusundan elde edilen görüntülerden 41 x 41 x 9 boyutlarında 59840 adet görüntü eğitim fazında, $250 \times 250 \times 9$ boyutlarında 80 adet görüntü

test fazında ve gerçek test görüntüsü olarak 800 x 800 x 9 boyutlarında 28 adet görüntü kullanılmıştır. IKONOS uydusundan elde edilen 400 × 400 × 5 boyutlarında 112 görüntü gerçek test görüntüsü olarak kullanılmıştır. Her veri seti için, 64 örnek veri kümesi 300 iterasyonda SGD ($\mu = 0.9$) ile CM (Classic Momentum) ($\varepsilon = 0.1$) optimizasyon fonksiyonları kullanılarak eğitim gerçekleştirildi. Daha önceki çalışmalara benzer olarak, PSNR, ERGAS, SAM, Q ve QNR gibi metrikler kullanılmıştır. Quickbird uydusundan elde edilen veri setinde elde edilen test sonuçlarından olan Q değeri PNN modelinde 0,9206 iken önerilen modelde 0,9303 olarak elde edilmiştir. Worldview-2 uydu veri setinde ise Q değeri PNN modelinden MSDCNN modeline 0,0181 artış elde edilmiştir. Son olarak IKONOS veri setinde ise PNN modelinde QNR değeri 0,8606 iken önerilen modelde 0,8797 olarak elde edilmiştir. Derin öğrenmeye dayalı yöntemlerin eğitim aşaması GPU (NVIDIA Quadro M4000) ortamında CUDA 7.5 tarafından desteklenen Caffe derin öğrenme çerçevesi kullanılarak gerçekleştirilmiştir. Tüm evrişimli ağların testi Intel CPU (Xeon E5-2620 v3 @ 2,40 GHz) Dell Tower 7810 iş istasyonunda gerçekleştirilmiştir.

SRPPNN modelini öneren Cai ve ark. [6] QuickBird, WorldView-3 ve Landsat-8 olmak üzere üç farklı uydudan alınan görüntüleri kullanmışlardır. Çiftler oluşturmak için 64 × 64 piksel boyutunda 7000 MS yama ve buna karşılık gelen 256 × 256 piksel boyutunda PAN yaması çıkarılmış ve tüm çiftler, eğitim/doğrulama veri setleri için %90/%10 oranında bölünmüştür. Test verileri, 256 × 256 piksel boyutunda 200 MS görüntüsünden ve bunlara karşılık gelen 1024 × 1024 piksel boyutunda PAN görüntülerinden oluşmaktadır. Metrik olarak SAM, ERGAS, Q, QNR gibi metrikler kullanmışlardır. Modelleri iki farklı stratejiden oluştuğu için kullandıkları ilk stratejiden 0,8326 Q değerine ve 0,8773 QNR değerlerine ulaşmışlardır. Kullandıkları ikinci strateji için ise Q ve QNR değerleri olarak sırasıyla 0,8357 ve 0,8516 değerlerini elde etmişlerdir. Bu çalışma kendi modelini sırasıyla PNN, MSDCNN, PanNet ve MSDCNN modelleri ile karşılaştırmıştır. Quickbird uydusundan elde edilen test veri setinde SAM değeri PNN için 6,5547, MSDCNN için 6,2311, PanNet için 6,8387 iken önerilen yöntemde 5,8732 olarak elde edilmiştir. Worldview-3 uydusundan elde edilen test veri setinde ise PNN, MSDCNN, PanNet için sırasıyla SAM değerleri 7,6080, 6,9759 ve 7,1270 iken önerilen yöntemde 5,7653 ile daha iyi bir sonuca ulaşıldığı tespit edilmiştir. Son olarak Landsat-8 uydusu için SAM sonuçları PNN, MSDCNN ve PanNet için 2,0458, 1,8695, ve 3,0754 iken önerilen yöntemde 1,4016 değeri ile daha yüksek bir skor elde edilmiştir. Geleneksel pankeskinleştirme yöntemlerinin aşamaları MATLAB R2019a aracılığıyla Intel Core i7-2600 CPU @ 3,40 GHz üzerinde gerçekleştirilmiştir. Derin

öğrenmeye dayalı yöntemlerin eğitim ve test aşamaları, TensorFlow'daki açık bir derin öğrenme çerçevesi aracılığıyla GPU'da (NVIDIA GeForce RTX 2080Ti) uygulanır.

Mamatha ve ark. [8] tarafından geleneksel pankeskinleştirme yöntemlerinden biri olan dalgacık dönüşüm tekniği FPGA ortamında uygulanmıştır. Gerçekleştirilen çalışma da Haar, Daubechies 3 (db3) ve Cohen Daubechies Feauveau (CDF) 9/7 filtreleri kıyaslanmıştır. Pankeskinleştirme teknikleri Xilinx ISE 13.1'de sentezlenerek ML605 Virtex 6 FPGA kiti üzerinde uygulanmıştır. Çalışma da PAN'ın dalga boyu (0,5 μm ila 0,8 μm) RGB'den Yoğukluk (Intensity, I) dönüştürücü blok kullanılarak yoğunluğa dönüştürülmüştür. Daha sonra 256 x 256 boyutlarındaki MS görüntüler FPGA'deki bellek kısıtlamaları nedeniyle 128 x 128 olarak yeniden boyutlandırılmıştır. Yeniden boyutlandırılan bu görüntüler, iki boyutludan MATLAB Simulink blok setleri kullanılarak tek boyutlu bit akışına dönüştürülmüştür. Daha sonra FPGA modeline girdi olarak sunulmuştur. MATLAB Simulink R2010b'de, düşük çözünürlüklü multispektral ve yüksek çözünürlüklü pankromatik görüntüleri Haar, daubechies 3 ve CDF 9/7 filtreleriyle birleştirmek için ortalama, toplamsal ve ikamesel füzyon kuralları kullanılmıştır. Sonuçlara bakıldığında CDF 9/7 filtresinin Haar ve db3 filtrelerine göre tüm füzyon kurallarında daha iyi performans gösterdiği ortaya çıkmıştır. Bu nedenle CDF 9/7, FPGA uygulaması için en iyi filtre olarak seçilmiştir. Önerilen yöntem FPGA entegrasyonu 4,3 Watt güç tüketimine sahiptir.

Viel ve ark. [9] hiperspektral ve pankromatik görüntüleri birleştiren bir pankeskinleştirme algoritması ve algoritmanın FPGA ortamına entegrasyonunu sunmuşlardır. FPGA üzerinde çalışacak olan yerel olmayan değişken (Non-local Variant, NLV) algoritması bir görüntü birleştirme tekniğidir. Algoritma donanımda, VHDL dili kullanılarak görüntü işlemedeki sayısal işlemler için IEEE 754 tek duyarlıklı ikili kayan nokta formatını kullanmışlardır. Xilinx Vivado Design Suite 2017.4 aracı kullanılarak sentezlenen model Xilinx Zynq-7000 SoC ZC702 Değerlendirme Kitinde mevcut olan XC7Z020-CLG484-1 FPGA'da gerçekleştirilmiştir. SoC 800 MHz'de çalışan iki ARM Cortex-A9 sabit çekirdeğe sahiptir. Gömülü uygulamada, Xilinx'in SW geliştirme kiti (SDK) ve Petalinux 2017.4 araçlarını kullanarak Ubuntu 19.10 dosya sistemiyle oluşturulmuş bir Linux dağıtımı kullanılmıştır. Test görüntülerinin okunması ve yazılması sürecinde Geospatial Data Abstraction Library'den (GDAL) yararlanılmıştır. Test verileri, 48 × 48 piksel boyutunda Moffett Field veri setinden oluşmaktadır. Her bir görüntü 10 banttan oluşmaktadır. Model için 100 iterasyonda hesaplama süresi 23,97 milisaniye olarak ölçülmüştür. Çalışma sonuçları CC

ölçütünde 0,97432 değeri elde edilmiştir. Önerilen yöntem FPGA entegrasyonu 10,01 Watt güç tüketimine sahiptir.

Akkin [10] tarafından Süper çözünürlüklü evrişimsel sinir ağlarına ait dört yöntem FPGA ortamında gerçekleştirilmiştir. İlk yöntem EDSR'dir, Div2K veri seti üzerinden eğitim eğitilmiştir. Div2K veri setinin %20'si eğitim için, kalan %80'i doğrulama için kullanılmış ve öğrenme oranı 0,001 olarak alınmıştır. İkinci yöntem ise ESPCN'dir, Div2K veri seti üzerinden elde edilen görüntülerin tamamı eğitim için kullanılmış, öğrenme oranı 0,01 olarak alınmıştır. Üçüncü yöntem ise FSRCNN'dir. T91 veri setinin %20'si eğitim için, %80'i doğrulama için kullanılmış ve öğrenme oranı 0,001 olarak alınmıştır. Son yöntem LapSRN'dir, Div2K veri seti tamamı üzerinden eğitim gerçekleştirilmiş, öğrenme oranı 0,01 olarak alınmıştır. Uygulamada Kria KV260 kartı, DPU (Deep Learning Processing Unit) kullanılmıştır. Metrik olarak PSNR ve SSIM gibi metrikler kullanmışlardır. Test görüntüsü 172 x179 çözünürlüğünde olan kelebek görselidir. Görüntünün çözünürlüğünün 2 katına arttırıldığı durum için SSIM metriği EDSR için 0,95, LapSRN için 0,94, ESPCN için 0,94 ve FSRCNN için 0,95 olarak elde edilmiştir. Diğer yandan PSRN metriği EDSR için 33,69 dB, LapSRN için 32,04, ESPCN için 32,53 dB ve FSRCNN için 32,88 olarak elde edilmiştir. Modellerin tek bir görüntü için hesaplama süresi ESDR için 12,72 saniye, LapSRN için 0,16 saniye, ESPCN için 0,021 saniye ve FSRCNN için 0,023 saniye olarak ölçülmüştür.

Verilen literatüre göre literatür eksiklikleri şu şekilde özetlenebilir:

- Literatürde yer alan derin öğrenme tabanlı pankeskinleştirme yöntemleri, yoğunlukta geleneksel evrişimli ağ mimarisini ve artık ağ mimarisini kullanmakta, ancak yoğun evrişimli ağ mimarisini kullanan çok az çalışma bulunmaktadır.
- Derin öğrenme tabanlı pankeskinleştirme yöntemlerine ait performans sonuçları CPU ve/veya GPU ortamı için sunulmuş fakat FPGA (Field Programmable Gate Array, FPGA platformuna taşınmadığı fark edilmiştir.
- FPGA ortamı hem paralel işlem yapabilme kabiliyeti hem de GPU ortamına göre daha az güç tüketmesi ile uydular gibi sınırlı enerji kaynağına ihtiyaç duyan platformlar için önemli bir kriter haline geldiği fark edilmiştir.

Verilen literatür eksikliklerine göre bu çalışmanın amacı şu şekildedir:

- Derin öğrenme mimarisine yoğun evrişimli ağ entegre edilirken, süper çözünürlük problemi için Zhang ve ark. [11] tarafından sunulmuş olan Artık Yoğun Ağ (Residual Dense Net, RDN) modelinden faydalanılmış ve bu yaklaşım en iyi sonuç elde eden SRPPNN modeline uygulanmıştır.
- Yeni nesil teknoloji ürünü olan Vitis AI aracı kullanılarak PNN, PanNet, MSDCNN, SRPPNN ve RDN uyarlaması için SRPPNN derin öğrenme tabanlı pankeskinleştirme modelleri FPGA platformuna taşınmıştır.
- Sahada Programlanabilir Kapı Dizisi (Field Programmable Gate Array, FPGA) platformunda derin öğrenme modellerinin çalıştırılması için üç farklı tasarım farklı kaynak kullanımı ve performans sonuçları ile birlikte sunulmuştur.

Gerçekleştirilen derin öğrenme tabanlı modelleri içeren çalışmalarda modelin performansı modelin işletilme süresi ile birlikte sunulmaktadır. Modelin işletilme süresi, modelin kendi karmaşıklık seviyesi ve modelin çalıştırıldığı platforma bağlı olarak değişmektedir. Model platformu olarak CPU ve GPU ortamı birçok çalışmada işletme süresi açısından değerlendirilmiş ve kendi çalışmamızda da bu değerlendirme yapılmıştır. CPU ortamı derin öğrenme modeli için gereken matematiksel hesaplamaları yaparken çok uzun işletme süresi ihtiyacı bulunmakta iken GPU ortamı CPU ortamına göre daha hızlı işlem yapabilme kabiliyeti sunmuştur. Fakat GPU ortamı hem pahalı hem de aşırı güç tüketen bir platformdur. Bu nedenle bu dezavantajların üstesinden gelebilmek adına paralel işlem yapabilme kabiliyeti ile FPGA ortamı önemli bir seçenek haline gelmiştir.

Derin öğrenme çalışmaları FPGA ortamına taşımanın birçok yöntemi bulunmaktadır. Bu yöntemler doğrudan donanım tabanlı kodlama dili (Very high speed integrated circuit Hardware Description Language, VHDL) ile modeli oluşturmak ve derlemek, MATLAB ortamında oluşturulan modelin FPGA ortamında çalışabilecek şekilde modelin derlenmesi, Xilinx Vivado HLS aracı kullanılarak C dilinde yazılmış kodun donanımda derlenecek şekilde dönüştürülüp derlenmesidir. Bu yöntemler önemli derecede donanım tabanlı kodlama diline hakimiyet ihtiyacını beklemektedir.

Gerçekleştirdiğimiz çalışmada yeni nesil bir araç olan Xilinx Vitis aracı öğrenilerek donanım katmanından uygulama katmanına kadar tüm ortam kurulmuş ve derin öğrenme tabanlı pankeskinleştirme modelleri Kria KV260 geliştirme kartında koşulmuştur. Donanım katmanında mimari Derin Öğrenme İşlem Birimi (Deep Learning Processing Unit, DPU)

‘nin konfigürasyonunu değiştirerek 3 farklı mimari tasarımı gerçekleştirilmiştir. İlk donanım mimarisinde 8 piksel paralelliği, 12 giriş kanalı paralelliği ve 12 çıkış kanalı paralelliği sunan B2304 mimarisi kullanılmıştır. Bu mimari 38631 LUT kullanırken, 4,531 W güç tüketimine sahiptir. İkinci donanım 8 piksel paralelliği, 14 giriş kanalı paralelliği ve 14 çıkış kanalı paralelliği sunan B3136 mimarisini kullandık. Bu mimari 42938 LUT kullanırken, 4,906 W güç tüketimine sahiptir. Son donanım mimarimizde 8 piksel paralelliği, 16 giriş kanalı paralelliği ve 16 çıkış kanalı paralelliği sunan B4096 mimarisini kullandık. Bu mimari 48280 LUT kullanırken, 5,601 W güç tüketimine sahiptir. Nihai durumda paralel işlem yapabilme kabiliyeti arttıkça, kaynak kullanımı ve güç tüketiminin arttığı gözlemlenmiştir. Diğer yandan RDN uyarlaması olan SRPPNN derin öğrenme modelinin üç farklı donanımda görüntü başına derin öğrenme mimarisinin test süresi ilk donanımda 0,072 saniye iken, ikinci donanımda 0,064 saniye ve üçüncü donanımda 0,044 saniye olarak ölçülmüştür. Sonuçlara göre paralel işlem yapabilme kabiliyeti ile test süresi iyileşmektedir. Diğer yandan CPU ve GPU ortamı ile kıyasladığımız durumda, derin öğrenme mimarisinin test süresi CPU ortamında 0,693 saniye ve GPU ortamında 0,117 saniye olarak ölçülmüştür. FPGA ortamında hesaplanan test süreleri CPU ve GPU ortamından da daha kısa sürmüştür.

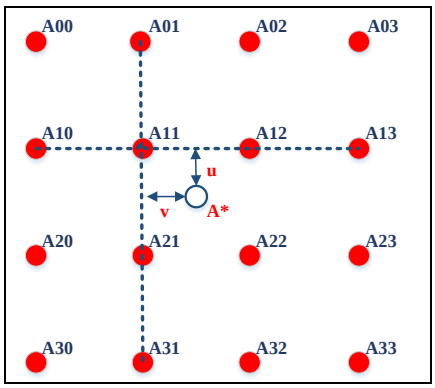
Gerçekleştirdiğimiz çalışmada 4 farklı derin öğrenme tabanlı pankeskinleştirme yöntemlerinden en iyi metrik sonuçlarını elde eden model için, model çıktı performansını arttırmak adına özellik çıkarımı ve yeniden yapılandırma için kullanılan Artık Ağ bloklarından bazıları Süper Çözünürlük için etkileyici sonuçlar veren Artık Yoğun Ağ blokları ile yer değiştirilmiştir. Bu durumda FPGA ortamında elde edilen sonuçlara göre WorldView-3 uydu görüntülerinden elde edilen sonuçlar için PSNR değeri 33,29’dan 33,52’ye yükseltilmiş iken Lansdat-8 uydu görüntülerinde ERGAS değeri 8,58’den 8,16’ya düşürülmüştür. Diğer yöntemler ile FPGA platformunda adaletli bir kıyaslama sunmak adına bütün derin öğrenme tabanlı yöntemler, FPGA platformuna taşınmış ve en iyi sonucu neredeyse tüm metriklerde önerdiğimiz yöntem elde etmiştir.

Gerçekleştirilen tez çalışmasına ait akış şu şekilde olmuştur: çalışmada kullanılan yöntemler 2. Bölümde açıklanmıştır. Deneysel sonuçlar Bölüm 3’te yer almaktadır. Tez çalışması kapsamında tartışma ve literatürde yer alan yöntemlerin karşılaştırılması Bölüm 4’te yer almış ve tez çalışmasına ait sonuca Bölüm 5’te yer verilmiştir.

2. MATERYAL VE YÖNTEM

2.1. Bikübik Enterpolasyon

Pankeskinleştirme yöntemlerini gerçekleştirirken görüntülerin ölçeklendirilmesi için bikübik enterpolasyon yönteminden faydalanılmıştır. Bu yöntem, Şekil 2.1'de gösterildiği gibi A* pikselinin değeri hesaplanırken kırmızı noktalar ile işaretlenen pikselin on altı en yakın komşusu dikkate alınarak hesaplanmaktadır.



Şekil 2.1. Bikübik enterpolasyon komşuluğu

Bilinmeyen piksele ait değerler hesaplanırken Eş. 2.1'den faydalanılır. Eş. 2.1'de R çıktı görüntüsünü, i pikselin satır konumunu ve j değeri ise pikselin sütun konumunu ifade etmektedir. Bilinmeyen değerler A katsayılarıdır. Bu değerler (i,j) konumuna ait en yakın 16 piksel değerine göre belirlenerek hesaplanır.

$$R(x\,,y) = \sum_{m=0}^{3}\sum_{n=0}^{3} A_{mn}\, X\, i^m\, j^n \qquad (2.1)$$

2.2. Gaussian Filtresi

Gaussian filtresi görüntü içerisinde yer alan yüksek frekanslı bilgileri azaltarak görüntünün bulanıklaşmasını sağlayan alçak geçiren bir filtredir. Gaussian filtresi x giriş değerleri için Eş. 2.2'de yer alan formül ile R(x) çıkış değerleri hesaplanmaktadır.

$$R(x) = \frac{1}{2*\pi*\sigma} * e^{-\frac{x^2}{2*\sigma}} \qquad (2.2)$$

Değişken x değerleri için Eş (2.2)'ye göre elde edilen çıktı değerleri ile bir çan eğrisi formu elde edilir. Gaussian filtresi, görüntü üzerinde konvolüsyon matrisi kullanarak hesaplanır. Gauss filtresindeki σ standart sapmadır. Çan eğrisinin formu σ katsayısına bağlı olarak değişmekte, σ değeri azaldıkça çan eğrisi daha dik olmakta ve konvolüsyon işleminde komşu piksellerin etkisi daha az olacağından görüntüdeki bulanıklaşma etkisi azalmaktadır.

Pankeskinleştirme yöntemlerini gerçekleştirirken görüntülerin yüksek frekanslı bileşenlerini ön plana çıkarmak için Gaussian filtresinden faydalanılmıştır. Yüksek geçiren filtre, görüntünün Gaussian filtresi (konvolüsyon çekirdek boyutu=5x5 , σ=1) çıkışı, orijinal görüntüden çıkarılarak elde edilmiştir.

2.3. Veri Seti

Deneysel sonuçları elde edebilmek için WorldView-3 [12] ve Landsat-8 [13] uydularından elde edilen görüntüler kullanılmıştır. WorldView-3 uydusundan elde edilen multispektral görüntü kırmızı, yeşil ve mavi renk bantlarından oluşmaktadır. WorldView-3 uydusu pankromatik görüntünün çözünürlüğü multispektral görüntü çözünürlüğünün 4 katı büyüklüğündedir. Uydunun düşük çözünürlüklü görüntüsünü simüle etmek için Wald Protokolü [14] kullanılmıştır. Şekil 2.2'deki gibi tüm orijinal görüntüler 7 x 7 Gauss filtresi çekirdeği tarafından filtrelenir ve sonrasında 4 faktörü ile alt örneklenir.



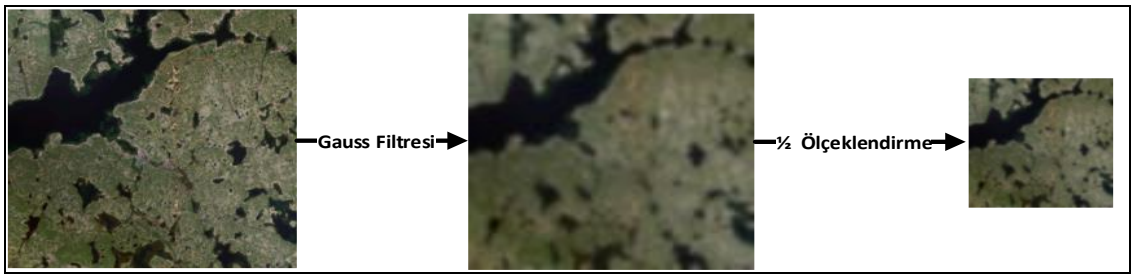
Şekil 2.2. WorldView-3 veri setinde düşük çözünürlüklü görüntü elde edimi

Derin öğrenme tabanlı modellerin eğitimi için WorldView-3 uydusuna ait veri seti Çizelge 2.1'de verilmiştir. Düşük çözünürlüklü görüntünün çözünürlük bilgisi 56 x 56 x 3,

pankromatik görüntünün çözünürlük bilgisi 224 x 224 x 1 ve yüksek çözünürlüklü görüntünün çözünürlük bilgisi 224 x 224 x 3'tür.

Çizelge 2.1. WorldView-3 veri seti

Görüntü Tanımı	Görüntü sayısı		
	Eğitim	Doğrulama	Test
Düşük Çözünürlüklü	2000	500	250
Pankromatik	2000	500	250
Yüksek Çözünürlüklü	2000	500	250

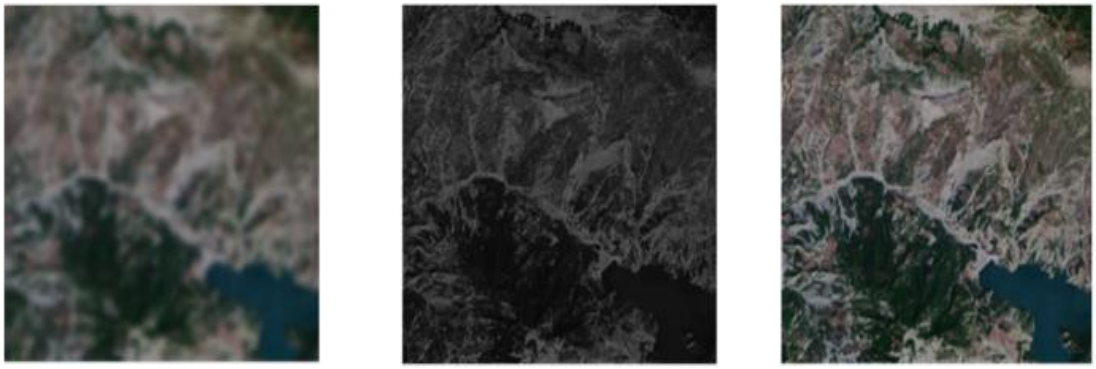


Şekil 2.3. Landsat-8 veri setinde düşük çözünürlüklü görüntü elde edimi

Landsat-8 uydusundan elde edilen multispektral görüntü kırmızı, yeşil ve mavi renk bantlarından oluşmaktadır. Landsat-8 uydusu pankromatik görüntünün çözünürlüğü multispektral görüntünün çözünürlüğünün 2 katı büyüklüğündedir. Uydunun düşük çözünürlüklü görüntüsünü simüle etmek için Wald Protokolü [14] kullanılmıştır. Şekil 2.3'te verildiği gibi tüm orijinal görüntüler 7 × 7 Gauss filtresi çekirdeği tarafından filtrelenir ve sonrasında 2 faktörü ile alt örneklenir. Landsat-8 veri setine ait görüntü sayısı WorldView-3 veri seti ile aynıdır ve Çizelge 2.1'de verilmiştir.

2.4. Geleneksel Pankeskinleştirme Yöntemleri

Uzaktan algılama verileri tipik olarak yüksek çözünürlüklü pankromatik ve düşük çözünürlüklü multispektral (çok bantlı) görüntülerden oluşur. Multispektral görüntülerin çözünürlüğünü artırma sürecinde, pankromatik görüntünün içerdiği bilgilerden yararlanan yöntemlere pankeskinleştirme yöntemleri denir. Pankeskinleştirme modelinin girişinde Resim 2.1. (a)'da yer alan multispektral düşük çözünürlüklü görüntü ve Resim 2.1. (b)'de yer alan aynı sahneye ait pankromatik görüntü bulunur. Modelin çıktısında Resim 2.1 (c)'deki gibi ayrıntıları ön plana çıkarılmış pankeskinleştirilmiş görüntü elde edilir.



(a) (b) (c)

Resim 2.1. Veri setine ait görüntüler: (a) Multispektral düşük çözünürlüklü görüntü, (b) Pankromatik görüntü, (c) Pankeskinleştirilmiş Multispekral Görüntü [13]

Geleneksel pankeskinleştirme yöntemleri, Bileşen Değiştirme (Component Substitution-CS) ve Çoklu Çözünürlük Analizi (MultiResolution Analysis-MRA) olmak üzere iki sınıfa ayrılmıştır.

2.5. Bileşen Değiştirme Pankeskinleştirme Yöntemleri

Bileşen Değiştirme, multispektral görüntünün uzaysal bileşenlerinin, pankromatik görüntüye ait uzaysal bileşenler ile yer değiştirildiği yöntemlerdir. Bileşen Değiştirme yaklaşımını benimseyen yöntemler; Brovey yöntemi, Yoğunluk Renk Doyum (Intensity Hue Saturation-IHS) yöntemi, Temel Bileşen Analizi (Principal Component Analysis-PCA), Kılavuzlu Filtre Temel Bileşen Analizi (Guided Filter Principal Component Analysis-GFPCA), Gram-Schmidt (GS) yöntemi ve Adaptif Gram-Schmidt (Adaptive Gram-Schmidt-GSA) yöntemidir. Bileşen Değiştirme yöntemleri, yöntem algoritmaları basit ve hesaplama yükü hafif [2] iken, önemli ölçüde spektral bozulmadan muzdariptir.

2.5.1. Brovey dönüşümü

Brovey dönüşümünde, Kırmızı-Yeşil-Mavi (RGB) olmak üzere üç kanaldan oluşan multispekral görüntü kanal bileşenlerine ayrılır. Sırasıyla her bir kanal bileşeni Eş. 2.3, Eş 2.4 ve Eş. 2.5'teki gibi pankromatik görüntü ile çarpılırken, kanal bileşenlerinin toplamına bölünür. Tüm kanallarda bu işlemler yapıldıktan sonra, bileşenler tekrardan tek bir görüntü formu olan Kırmızı-Yeşil-Mavi (RGB)'ye dönüştürülerek pankeskinleştirilmiş görüntü elde edilir.

$$R\sim = \frac{R}{(R+G+B)/3} * PAN \tag{2.3}$$

$$G\sim = \frac{G}{(R+G+B)/3} * PAN \tag{2.4}$$

$$B\sim = \frac{B}{(R+G+B)/3} * PAN \tag{2.5}$$

2.5.2. Yoğunluk renk doyum dönüşümü

Multispektral görüntü, Kırmızı-Yeşil-Mavi (RGB) renk uzayından Yoğunluk-Renk-Doyum (IHS) renk uzayına dönüştürülür. Bu dönüşüm işlemi sırasıyla yoğunluk (I) bileşeni ile v1 ve v2 değerleri için Eş. 2.6'daki gibi hesaplanır. Renk tonu bileşeni (H) için Eş. 2.7'deki gibi, doygunluk bileşeni (S) için Eş. 2.8'deki gibi hesaplanır.

$$\begin{pmatrix} I \\ v1 \\ v2 \end{pmatrix} = \begin{bmatrix} \frac{1}{\sqrt{3}} & \frac{1}{\sqrt{3}} & \frac{1}{\sqrt{3}} \\ \frac{1}{\sqrt{6}} & \frac{1}{\sqrt{6}} & \frac{-1}{\sqrt{3}} \\ \frac{1}{\sqrt{2}} & \frac{-1}{\sqrt{2}} & 0 \end{bmatrix} \begin{pmatrix} R \\ G \\ B \end{pmatrix} \tag{2.6}$$

$$H = tan^{-1}(\frac{v1}{v2}) \tag{2.7}$$

$$S = \sqrt{v_1^2 + v_2^2} \tag{2.8}$$

Multispektral görüntünün yoğunluk bileşeni (I) yerine, histogram eşitlemesi yapılmış ve ölçeklendirilmiş olan pankromatik (PAN) görüntü alınır. Nihai durumda sırasıyla Eş. 2.9, Eş. 2.10 ve Eş 2.11'da yer alan ters dönüşüm fonksiyonları ile tekrar Kırmızı-Yeşil-Mavi (RGB) renk uzayına geçilerek pankeskinleştirilmiş görüntü elde edilir.

$$v_1 = \mathrm{S} * \cos(H) \tag{2.9}$$

$$v_2 = \mathrm{S} * \sin(H) \tag{2.10}$$

$$\begin{pmatrix} R \\ G \\ B \end{pmatrix} = \begin{bmatrix} \frac{1}{\sqrt{3}} & \frac{1}{\sqrt{6}} & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{3}} & \frac{1}{\sqrt{6}} & \frac{-1}{\sqrt{2}} \\ \frac{1}{\sqrt{3}} & \frac{-1}{\sqrt{3}} & 0 \end{bmatrix} \begin{pmatrix} PAN \\ v_1 \\ v_2 \end{pmatrix} \tag{2.11}$$

2.5.3. Temel bileşen analizi

Multispektral görüntü renk bileşenlerine kırmızı (R), yeşil (G) ve mavi (B) olacak şekilde ayrılır. Kırmızı (R), yeşil (G) ve mavi (B) bileşenlere temel bileşen analizi uygulanarak Eş. 2.12'de yer alan temel bileşenleri elde edilir. Birinci temel bileşen (PC_1) ile histogram eşitlemesi yapılmış olan pankromatik (PAN) görüntü yer değiştirilir. Yer değiştirilmiş olan bileşen ile Eş. 2.13'te yer alan ters dönüşüm fonksiyonu ile tekrar Kırmızı-Yeşil-Mavi (RGB) renk uzayına geçilerek pankeskinleştirilmiş görüntü elde edilir.

$$\begin{pmatrix} PC_1 \\ PC_2 \\ PC_3 \end{pmatrix} = \begin{bmatrix} \frac{1}{3} & \frac{1}{3} & \frac{1}{3} \\ \frac{1}{2} & 0 & \frac{-1}{2} \\ \frac{-1}{4} & \frac{1}{2} & \frac{-1}{4} \end{bmatrix} \begin{pmatrix} R \\ G \\ B \end{pmatrix} \tag{2.12}$$

$$\begin{pmatrix} R\sim \\ G\sim \\ B\sim \end{pmatrix} = \begin{bmatrix} 1 & 1 & \frac{-2}{3} \\ 1 & 0 & \frac{4}{3} \\ 1 & -1 & \frac{-2}{3} \end{bmatrix} \begin{pmatrix} PAN \\ PC_2 \\ PC_3 \end{pmatrix} \tag{2.13}$$

2.5.4. Gram-Schmidt yöntemi

Gram-Schmidt yönteminde, ilk olarak pankromatik görüntünün çözünürlüğü, multispektal görüntü çözünürlüğüne düşürülür. Yöntem, bir tür ortogonelleştirme işlemi üzerine odaklanmıştır. Eş. 2.14'te yer alan projeksiyon fonksiyonunda, <u, v>, u ile v vektörünün iç çarpımıdır ve <u, v> = 0 olduğunda, iki vektörün ortogonal olduğu söylenir.

$$proj_u(v) = \frac{<u,v>}{<v,v>} u \tag{2.14}$$

Düşük çözünürlüklü pankromatik görüntü, Eş. 2.15'te k = 1 için u_1 vektörü olarak karşımıza çıkar. Gram-Schmidt ortonormalizasyonu Eş. 2.16'daki gibi n_k vektör setini oluşturur.

Multispektral bantlar, u_1 ile ortogonalleştirilir. Daha sonra düşük çözünürlüklü pankromatik görüntü ile orijinal pankromatik görüntü yer değiştirir [15]. Bu yer değiştirme ortogonelleştirme ilişkisi kuracak şekilde pankeskinleştirilmiş görüntünün oluşmasını sağlar. Gram-Schmidt spektral dönüşümünün tersini uygulayarak elde edilen temel bileşenleri tekrar Kırmızı-Yeşil-Mavi (RGB) renk uzayına geçilerek pankeskinleştirilmiş görüntü elde edilir.

$$u_k = v_k - \sum_{i=1}^{k-1} proj_{ui}(v_k) \tag{2.15}$$

$$n_k = \frac{u_k}{||u_k||} \tag{2.16}$$

2.5.5. Kılavuzlu filtre temel bileşen analizi

Kılavuzlu Filtre Temel Bileşen Analizi (GFPCA), Gram-Schmidt yöntemi ile Temel Bileşen Analizi (PCA) yöntemlerini kullanır. Yöntem ilk olarak Kırmızı-Yeşil-Mavi (RGB) renk bileşenlerine Gram-Schmidt yöntemi uygulanarak spektral uzay ortama geçilir. Spektral uzay ortamındaki bileşenlere temel bileşen analizinden elde edilen temel bileşenleri, pankromatik görüntü ile çarpma işlemi gerçekleştirilir. Gram-Schmidt spektral dönüşümünün tersini uygulayarak elde edilen temel bileşenleri tekrar Kırmızı-Yeşil-Mavi (RGB) renk uzayına geçilerek pankeskinleştirilmiş görüntü elde edilir.

2.5.6. Adaptif Gram-Schmidt

Gram-Schmidt yönteminden farklı olarak Adaptif Gram-Schmidt yönteminde, multispektral ve pankromatik bantları arasında doğrusal bir regresyon gerçekleştirilir. Pankromatik görüntüye alçak geçiren filtre uygulanarak, çözünürlüğü multispektral görüntünün çözünürlüğüne düşürülmesi sağlanır. Bu görüntü P olarak gösterilir. Kırmızı-Yeşil-Mavi (Red-Green-Blue) renk bileşenlerine ayrılarak Eş. 2.17'deki gibi sentez yoğunluk (I) değeri hesaplanır.

$$I = w_1 * B + w_2 * G + w_3 * R \tag{2.17}$$

P ve I arasındaki ortalama kare hatasını en aza indirmek için regresyon algoritması [16] ile w_1, w_2, w_3 ve b katsayıları tahmin edilir. Ağırlıkların amacı orijinal pankromatik görüntüsüne benzer bir yoğunluk görüntüsü (I) sentezlemektir. Sentezlenen yoğunluk ve multispektral kanallar arasındaki kovaryans değerine yakınsanarak g_1, g_2 ve g_3 katsayıları hesaplanır [17].

$$g_1 = \frac{cov(I,G)}{var(I)} \tag{2.18}$$

$$g_2 = \frac{cov(I,G)}{var(I)} \tag{2.19}$$

$$g_3 = \frac{cov(I,R)}{var(I)} \tag{2.20}$$

Mavi, yeşil ve kırmızı kanallar için g_1, g_2 ve g_3 sırasıyla Eş. 2.18, Eş. 2.19 ve Eş. 2.20'deki gibi hesaplanarak kanallara eklenecek olan ayrıntı miktarını belirlenir.

2.6. Çoklu Çözünürlük Analizi Pankeskinleştirme Yöntemleri

Çoklu Çözünürlük Analizi, multispektral ile pankromatik görüntünün, uygulanan filtreler veya dönüşümler aracılığıyla birleştirilmesidir. Bu yöntemlerde genellikle Laplasyen Piramidi veya Dalgacık Dönüşümü kullanılmaktadır. Çoklu Çözünürlüklü Analizi'nde, pankromatik görüntüden multispektral görüntüye bol miktarda yüksek frekanslı ayrıntı enjekte edilmesine rağmen, görüntünün görsel kalitesini düşüren uzaysal bozulma meydana gelmektedir. Çoklu çözünürlüklü Analizi yaklaşımını benimseyen yöntemler; Yumuşatma Filtresi Tabanlı Yoğunluk Modülasyonu (Smoothing Filter-based Intensity Modulation, SFIM), Modülasyon Transfer Fonksiyonu Bazlı Laplasyen Piramitleri (Modulation Transfer Function, MTF), Modülasyon Transfer Fonksiyonu Bazlı Genellestirilmiş Laplasyen Piramitleri (Generalized Laplacian Pyramid Based Modulation Transfer Function, MTF GLP) ve Dalgacık Dönüşümleri (Wavelet Transform)'dir. Sırasıyla bu teknikler kullanılan matematiksel fonksiyonları ile birlikte sırasıyla açıklanacaktır. Çoklu Çözünürlük Analizi, Bileşen Analizi yöntemlerine göre daha karmaşık ancak daha iyi sonuç sunabilmektedir.

2.6.1. Yumuşatma filtresi tabanlı yoğunluk modülasyonu

HPF yöntemine benzer bir yöntemdir. Bu yöntemde yüksek geçiren bir filtre yerine bulanıklaştırma (alçak geçiren) filtreleri kullanılır. Alçak geçiren filtresi pankromatik görüntü ile konvolüsyona sokulur. Ardından aşağıda verilen ifade uygulanır ve pankeskinleştirilmiş görüntü elde edilmiş olur [18].

$$O_i = MS_i * \frac{PAN}{PAN_{lpf}} \tag{2.21}$$

Verilen Eş. 2.21'de MS, multispektral görüntüyü, PAN, pankromatik görüntüyü, PAN_{lpf} ise bulanıklaştırma filtresi ile filtrelenmiş görüntüyü ifade eder.

2.6.2. Modülasyon transfer fonksiyonu bazlı genelleştirilmiş laplasyen piramitleri

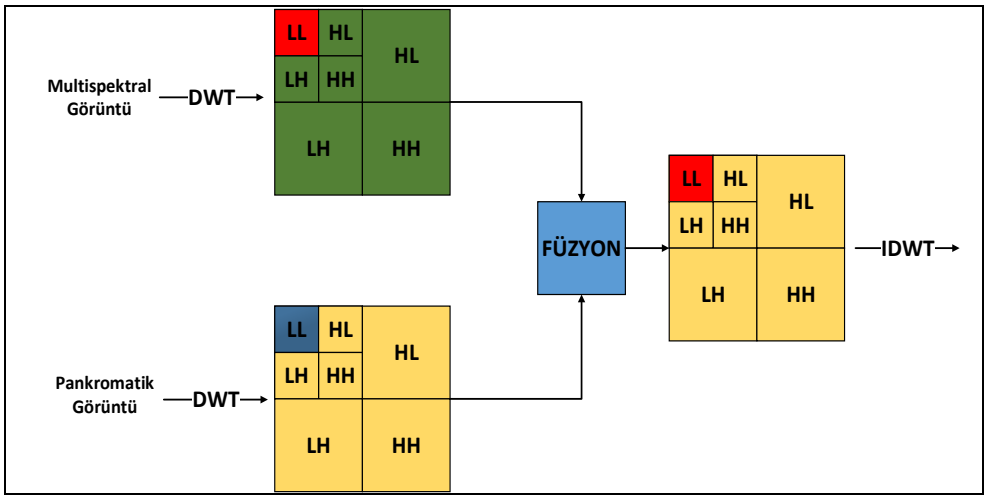
Bu füzyon yöntemi, çok ölçekli ve aşırı örneklenmiş bir yapı olan genelleştirilmiş laplasyen piramidine (GLP) dayanır. Amaç, daha kaba ayrıntıların spektral bilgisini koruyarak, bir görüntüden diğerine uzaysal bilginin seçici olarak enjekte edilmesidir [19]. GLP indirgeme filtresi, modülasyon transfer fonksiyonlarının (MTF) bir modelini kullanarak, pankromatik görüntüde yer alan yüksek uzaysal frekans bileşenlerini multispektral görüntüye seçici geçirgenlikte enjekte eder. MTF'ye ait modelleme görüntüyü yakalayan sensörlerin teknolojisine bağlı olarak değişir, bu nedenle MTF karakteristiği uydudan uyduya farklılık gösterir.

2.6.3. Yüksek geçiren filtre ile modülasyon transfer fonksiyonu bazlı genelleştirilmiş laplasyen piramitleri

Modülasyon transfer fonksiyonu bazlı genelleştirilmiş laplasyen piramitleri yönteminden farklı olarak bu yöntem pankromatik görüntüden ayrıntıları çıkarmak için yüksek geçiş filtresinin (HPF) uygulanmasından faydalanmaktadır [20]. Yöntem, yüksek geçişli filtre tasarımında Gaussian filtresi kullanmaktadır. Uzaysal bilgi, yukarı örneklenmiş olan multispektral görüntüye enjekte edilmesi, hem sensörün karakteristiğine ait olan modülasyon transfer fonksiyonunun tamamlayıcı frekanslarından faydalanarak, hem de pankromatik görüntünün yüksek geçiren filtre çıktısından faydalanarak gerçekleşir.

2.6.4. Dalgacık dönüşümleri

Dalgacık yöntemi, multispektral ve pankromatik görüntüleri ayrıştırmak için Ayrık Dalgacık Dönüşümlerini (DWT) kullanır. DWT yönteminde, dalgacık analizi için iki, dalgacık sentezi için iki olacak şekilde toplamda dört adet FIR filtre kullanılır. Bu dört adet filtre öncelikle satırlara, ardından da sütunlara uygulanır. İşlem sonucunda dört adet bileşen elde edilir. Bu bileşenler, Şekil 2.4'teki gibi yaklaşık (LL), yatay (LH), dikey (HL), ve çapraz (HH) bileşenlerini oluşturur.



Şekil 2.4. Dalgacık dönüşümü ile pankeskinleştirme modeli

DWT, Şekil 2.4'teki gibi hem multispektral hem de pankromatik görüntüyü bileşenlere ayrıştırır. Bir bileşen (LL) düşük çözünürlüklü ve spektral bilgi içerirken diğerleri daha ayrıntılı uzaysal bilgi içerir. Pankromatik görüntünün düşük çözünürlüklü bileşeninin yerini düşük çözünürlüklü multispektral bileşeni alır. Bileşen değimi yapılmış forma ters dalgacık dönüşümü (IDWT) gerçekleştirilerek pankeskinleştirilmiş görüntü elde edilir.

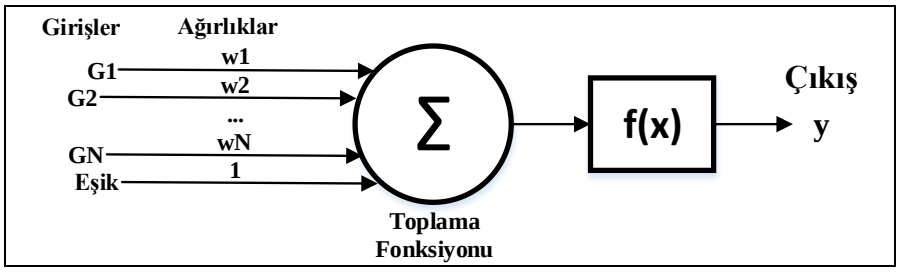
2.7. Derin Öğrenme Tabanlı Pankeskinleştirme

2.7.1. Yapay zekâ temelleri

Derin öğrenme en az bir adet yapay sinir ağının kullanıldığı bir yapay zekâ tekniğidir. Derin öğrenme Çok Katmanlı Perceptron, Evrişimli Sinir Ağı ve Yinelgeli Sinir ağı olmak üzere üç ana modele sahiptir. Derin öğrenme modelleri ses veya görüntü verileri için üretme, tespit etme ve iyileştirme olmak üzere birçok amaç için kullanılmaktadır.

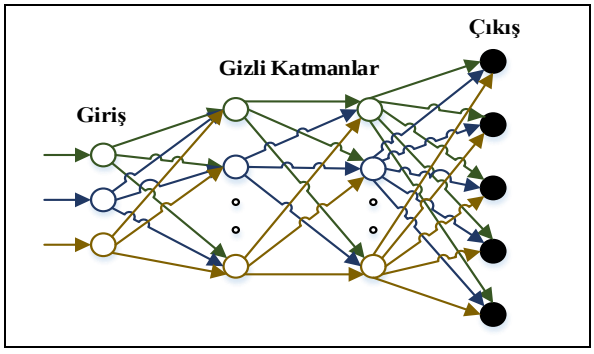
2.7.2. Yapay sinir ağları

Yapay sinir ağları insan kabiliyetlerini öğrenerek oluşturulur. Bu öğrenme farklı kabiliyetler öğrenme yeteneklerine sahiptir. Yapay sinir ağları birbirleri ile bağlı nöronlardan oluşur. Nöronlarda uçtan uca bilgi taşımı gerçekleşir. Yapay sinir ağlarında amaç, modelin en iyi doğruluk değerinin vereceği ağırlık değerinin hesabını yapmaktır [21]. Bir yapay sinir hücresi Şekil 2.5'te verilmiştir. Şekil 2.5'te girişler, ağırlıklar, toplama fonksiyonu, aktivasyon fonksiyonu ve çıkış olmak üzere 5 bölümden oluşur.



Şekil 2.5. Yapay sinir hücresi

Girişlerden gelen bilgiler toplanmak üzere toplama fonksiyonuna gönderilir. Toplama fonksiyonuna yönlendirilen bilgiler geldikleri bağlantıların ağırlığıyla çarpıldıktan sonra iletilir. Bunun sebebi girişlerin çıktı üzerindeki etkisinin ayarlanabilir olmasıdır. Toplama fonksiyonuna gelen veriler toplanarak aktivasyon fonksiyonu girdisi olan değeri hesaplar. Aktivasyon fonksiyonu, ağırlıklı toplama sonucunu doğrusal olmayan bir forma dönüştürür. Tipik olarak ReLU, sigmoid olmak üzere fonksiyonlar kullanılır. Aktivasyon fonksiyonun çıktısı nihai çıkış değerini oluşturmaktadır. Bir yapay sinir hücresinin çıktısı başka bir yapay sinir hücresi ile bağlanabilir. Bu bağlantı sayesinde yapay sinir ağları oluşur. Yapay sinir ağlarına ait örnek Şekil 2.6'da verilmiştir.



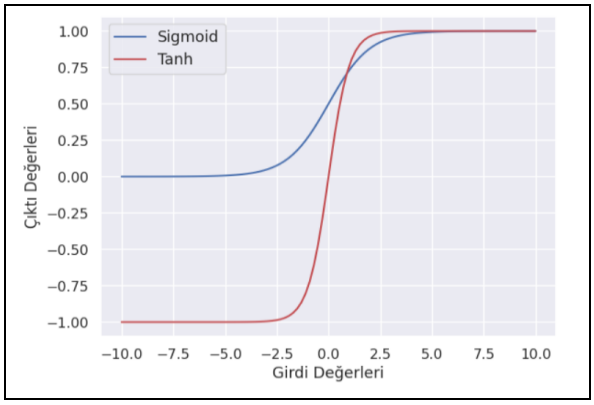
Şekil 2.6. Yapay sinir ağları

Yapay sinir ağı modeli Şekil 2.6'daki gibi ilk katman olan girişten, son katman olan çıkıştan oluşur. Aradaki katmanlara ise gizli katmanlar denir. Her yapay sinir hücresi bir sonraki katmandaki yapay sinir hücresi ile bağlantı halindedir. Yapay sinir ağları, ileri beslemeli ağlar ve geri beslemeli ağlar olmak üzere ikiye ayrılır. İleri beslemeli yapay sinir ağlarında, dış ortamlardan alınan bilgiler doğrudan gizli katmana iletilir. Gizli katmanında işletilen bilgiler, çıkış katmanına yönlendirilir. Nihai çıktı değeri çıkış katmanında bilginin işletilmesi ile elde edilir. Geri beslemeli yapay sinir ağlarında, bir yapay sinir hücresi çıktısı sadece kendinden sonra gelen yapay sinir hücresi kendinden önceki katmanda veya kendi katmanında bulunan herhangi bir yapay sinir ağına girdi olarak bağlanabilir. Bu yapısı nedeni ile doğrusal olmayan bir yapı oluşturur [22].

2.7.3. Aktivasyon fonksiyonları

Aktivasyon fonksiyonları, derin öğrenme modelinin hedeflenen amaca yönelik başarılı çıktı üretmesini desteklemektedir. İyi bir aktivasyon fonksiyonundan beklenen doğrusal olmayan bir davranış, türevlenebilir bir fonksiyon ve orijin noktasına göre yakınsamadır [23]. Veriye ve problemlere göre kullanılan birçok aktivasyon fonksiyonu bulunmaktadır. Doğrusal olmayan aktivasyon fonksiyonları, sigmoid, hiperbolik tanjant, ReLU ve Leaky ReLU'dur. Sigmoid fonksiyonunun grafiksel gösterimi Şekil 2.7'da fonksiyonel gösterimi Eş. 2.22'de gösterilmiştir. Sigmoid fonksiyonu değer aralığı [0,1] arasındadır. Sigmoid fonksiyonu uç giriş değerleri için aynı çıktı değerlerini verir. Bu noktalarda değişim sıfıra yakınsamaktadır. Bu durum sigmoid fonksiyonu için bir dezavantaj haline gelir.

$$y = \frac{1}{1+e^{-x}} \tag{2.22}$$

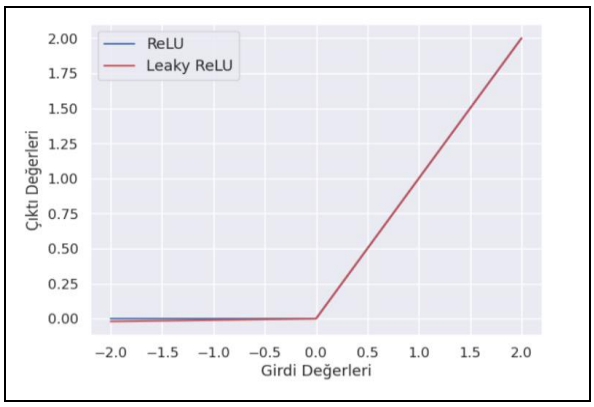


Şekil 2.7. Sigmoid (mavi eğri) ve tanh (kırmızı eğri) aktivasyon fonksiyonu

$$y = \frac{e^{x}-e^{-x}}{e^{x}+e^{-x}} \tag{2.23}$$

Hem sigmoid hem de hiperbolik tanjant fonksiyonu için hesaplama yükü fazladır. Bu nedenle literatürde daha çok ReLU fonksiyonu daha az hesaplama yükü olduğu için tercih edilmektedir. ReLU fonksiyonunun grafiksel gösterimi Şekil 2.8'de, fonksiyonel gösterimi Eş. 2.24'te gösterilmiştir. Sigmoid fonksiyonu değer aralığı [0, +x] arasındadır, x pozitif girdi değerini temsil etmektedir.

$$y = \begin{cases} 0, x < 0 \\ x, x \geq 0 \end{cases} \tag{2.24}$$



Şekil 2.8. Relu (mavi eğri) ve leaky relu (kırmızı eğri) aktivasyon fonksiyonu

ReLU fonksiyonunda tüm negatif girdi değerleri için çıktı değerleri aynıdır. Bu durum ReLU fonksiyonu için dezavantaj haline gelmektedir. Bu dezavantajın üstesinden gelmek için farklı negatif girdi değerleri için farklı çıktı değerleri üreten Leaky ReLU fonksiyonu geliştirilmiştir Leaky ReLU fonksiyonunun grafiksel gösterimi Şekil 2.8'de, fonksiyonel gösterimi Eş. 2.25'te gösterilmiştir.

$$y = \begin{cases} (0.01) * x, x < 0 \\ x \quad\quad , x \geq 0 \end{cases} \tag{2.25}$$

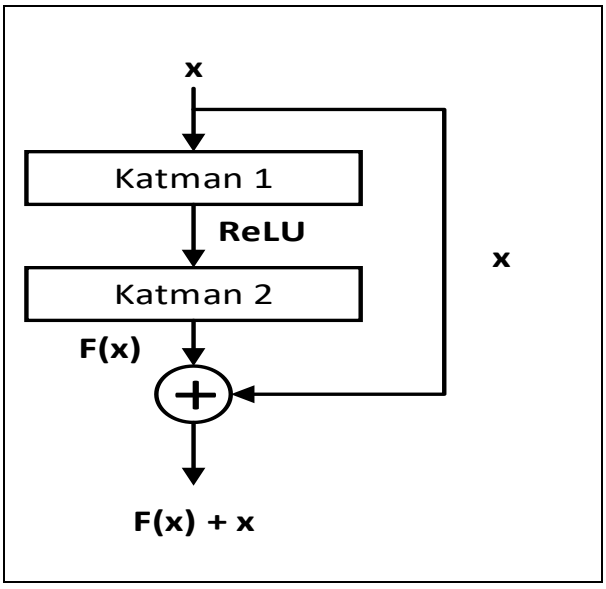
2.7.4. Evrişimli sinir ağları

Evrişimsel Sinir Ağları (Convolutional Neural Networks, CNN), Evrişimsel sinir ağları, derin öğrenmenin bir alt dalıdır ve genellikle görsel bilginin analiz edilmesinde kullanılan bir derin öğrenme yöntemidir. CNN evrişimsel katman, havuzlama katmanı ve tam bağlantı

katmanı olmak üzere üç katmana sahiptir. Evrişim katmanı, evrişimsel sinir ağlarının temel katmanıdır. Evrişim katmanının amacı, girdi verilerine uygulanan filtre matrisleri ile girdiye ait özellik haritasının çıkarılmasıdır. Daha sonra çıkarılan bu özellik haritaları kullanılarak model eğitimi gerçekleşir. Ortaklama katmanı, evrişim katmanından elde edilen özelliklerden bazılarının ön plana çıkartılması için azaltılmasıdır. Ağın eğitimi gerçekleşirken modelin ortaklama katmanı sayesinde yalnızca model için önemli olan özelliklere odaklanması model performansını arttırmaktadır. Ortaklama katmanlarından maksimum ortaklama ve ortalama ortaklama olmak üzere iki çeşit ortaklama katmanı bulunmaktadır. Maksimum ortaklama katmanı literatürde daha çok tercih edilmektedir. CNN modellerinde evrişimli katman ve ortaklama katmanları özellik çıkarımı için kullanılırken, çıkarılan özelliklerin sınıflandırılması için tam bağlantı katmanı kullanılmaktadır.

2.7.5. Artık ağ

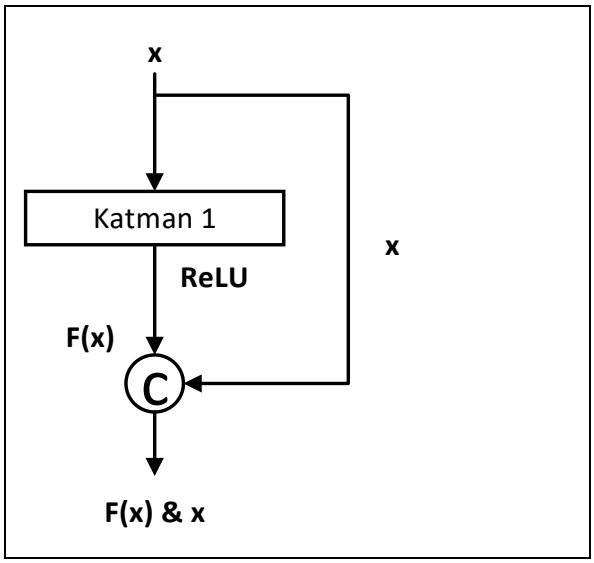
Artık Ağ (ResNet) atlamalı bağlantılardan oluşan bir mimaridir. Geleneksel derin sinir ağlarında her katman doğrudan bir önceki katmana bağlanmaktadır. ResNet mimarisinde ise artık bağlantılar aracılığı ile iki veya daha fazla katman atlanarak bağlantı kurulur. Geleneksel derin sinir ağlarında yaşanan artan gradyan problemine çözüm olarak ResNet mimarisi geliştirilmiştir. ResNet mimarisi Şekil 2.9'da verilmiştir.



Şekil 2.9. Resnet ağı

2.7.6. Yoğun evrişimli ağ

Yoğun Evrişimli Ağ (DenseNet)'da her katman, kendinden önceki katmanlardan birleştirme fonksiyonu ile bilgiler alır. Bu ağda önceki katmanlardan taşınan kolektif bilgi sayesinde özellik haritası daha zengin desenlere sahip olma eğilimindedir. Şekil 2.10'da DenseNet ağ mimarisi verilmiştir.



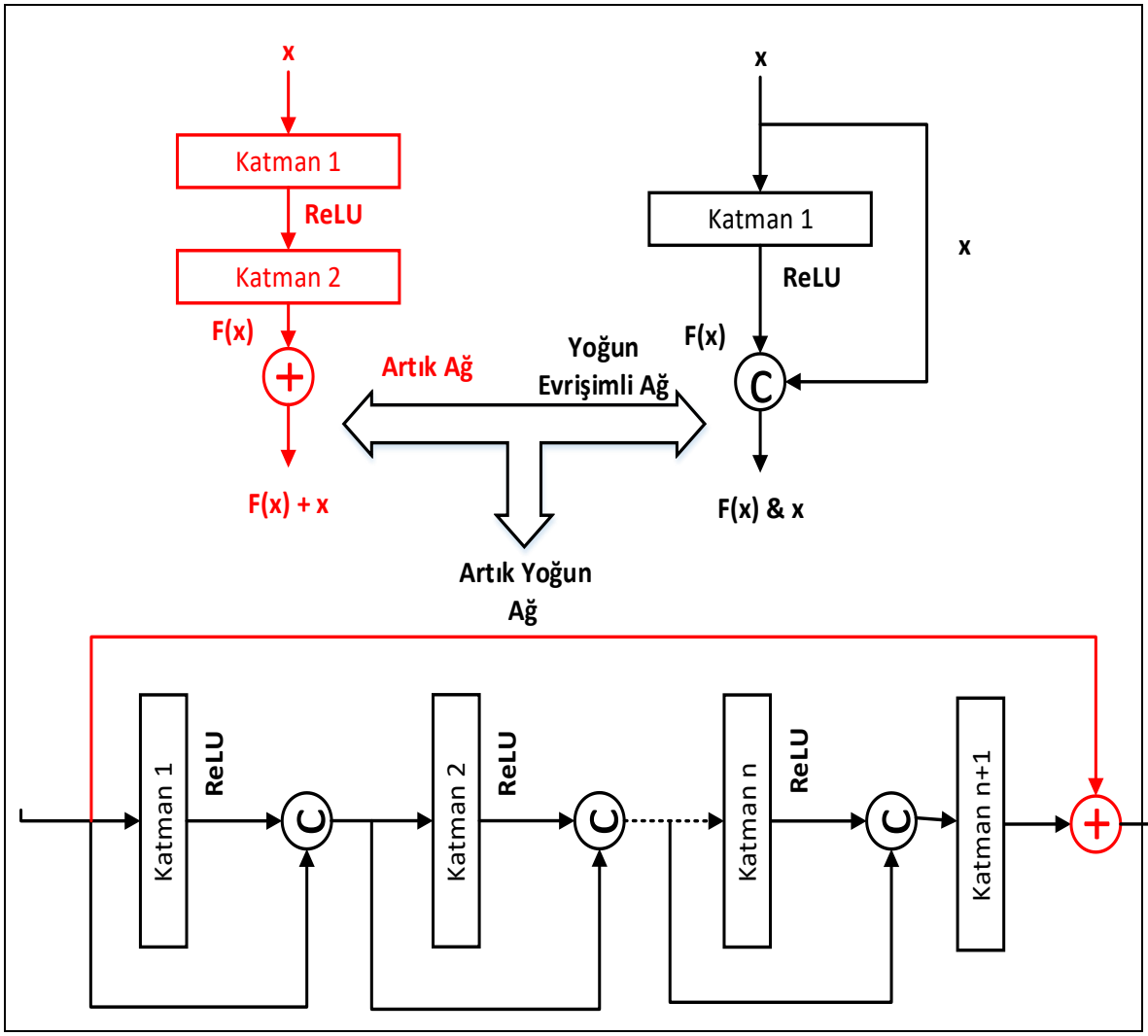
Şekil 2.10. DenseNet ağı

2.7.7. Süper çözünürlüklü evrişimsel sinir ağları

Süper çözünürlüklü evrişimsel sinir ağları düşük çözünürlüklü görüntülerden daha yüksek çözünürlüklü görüntüler elde etmeyi amaçlar. Literatürde derin öğrenme tabanlı süper çözünürlüklü model tasarımını ilk olarak Dong ve ark.'nın [7] SRCNN modelleri ile gerçekleştirmişlerdir. Daha sonrasında birçok süper çözünürlüklü çıktı üretmeyi hedefleyen modeller geliştirilmiştir. Popüler olan modellerden bazıları şunlardır; SRCNN (Super Resolution Convolutional Neural Network), FSRCNN (Fast Super Resolution Convolutional Neural Network) [24], EDSR (Enhanced Deep Super-resolution Network) [25] ve RDN (Residual Dense Network) [11]'dir. Geliştirilen her model bir önceki modelin sahip olduğu dezavanjların üstesinden gelmeyi hedeflemektedir. FSRCNN, hesaplama yükü ve daha iyi doğruluk elde etmesi ile SRCNN'in önüne geçmiştir. EDSR ve RDN ise FSRCNN'den daha iyi sonuç elde etmiştir. EDSR'de ResNet mimarisi kullanılırken, RDN'de DenseNet mimarisi kullanılmaktadır.

2.7.8. Süper çözünürlük için RDN

Artık ağ mimarisi ile yoğun evrişimli ağ mimarisi bir araya getirilerek Şekil 2.11'deki gibi artık yoğun ağ modeli elde edilmiştir. Tüm evrişim katmanlarındaki hiyerarşik özelliklerden tam olarak yararlanmaktadır. Yoğun yerel özellikleri tamamen elde edildikten sonra, küresel hiyerarşik özellikleri bütünsel bir şekilde ortaklaşa ve uyarlanabilir bir şekilde öğrenmek için artık ağ kullanılmaktadır.

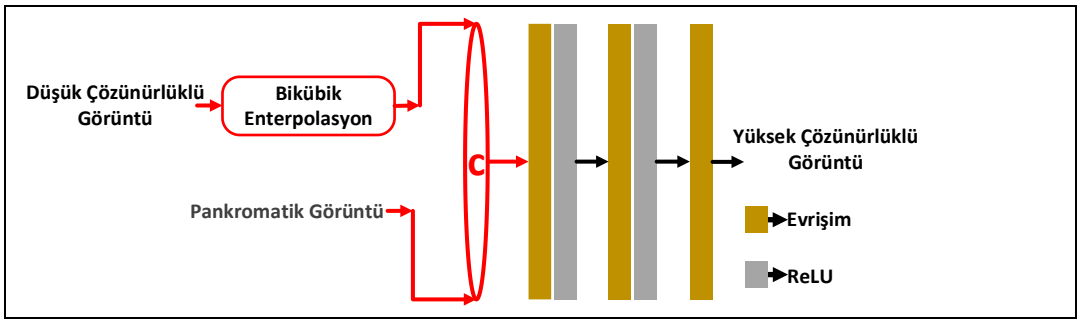


Şekil 2.11. Yoğun artık ağ modeli

Şekil 2.11'de olduğu gibi gerçekleştirilen çalışmada n değeri 8 olarak alınmıştır. Şekilde yer alan Katman 1, Katman 2 ve Katman n, 3 x 3 evrişimli katman, küresel artık öğrenmeye yönelik özellikleri çıkarmak için kullanılırken, 1 x 1 evrişimli katman, farklı seviyelerdeki özellikleri birleştirmek için kullanılmaktadır.

2.7.9. PNN

PNN (Pansharpening by Convolution Neural Network, PNN) modeli, Süper Çözünürlüklü Evrişimsel Sinir Ağları (Super-Resolution Convolutional Neural Network, SRCNN) yaklaşımından ilham alan bir pankeskinleştirme tekniğidir. SRCNN yöntemi düşük çözünürlüklü bir görüntüden yüksek çözünürlüklü görüntü elde eden bir modeldir [7]. SRCNN modelinde belirli evrişimsel katmanlar kullanılarak düşük çözünürlüklü görüntülerden yüksek çözünürlüklü görüntüler elde edilmeye çalışılır. PNN modelinin SRCNN modelinden farkı sadece renkli görüntülerin (R, G ve B bantları) değil, pankromatik bantların da model girişinde kullanılmasıdır. Buna ek olarak, model karmaşıklığı artırılmadan pankromatik bandı modele girdi olarak vererek bu bantta yer alan radyometrik indekslerin çeşitli özellik haritalarını girdiye dahil etmişlerdir. Düşük çözünürlüklü MS görüntülerle birlikte pankromatik bant kullanılarak çıkışta pankeskinleştirilmiş bir yüksek çözünürlüklü görüntü elde edilir. Buradaki amaç sadece pankeskinleştirme yöntemi kullanılarak yüksek çözünürlüklü görüntü elde edilmesi değil, ayrıca modeli eğiterek geleneksel pankeskinleştirme yöntemlerinden daha iyi bir pankeskinleştirme yöntemi elde etmektir. Bu durumu modele eğitmek için referans görüntülerde yüksek çözünürlüklü görüntüler kullanılır. PNN modeli Şekil 2.12'de verilmiştir.



Şekil 2.12. PNN modeli

PNN modeli Şekil 2.12'deki gibi katmanlardan oluşmaktadır. Düşük çözünürlüklü MS görüntü PNN derin öğrenme modeline gönderilmeden önce çözünürlüğü bikübik enterpolasyon yöntemi ile pankromatik görüntünün çözünürlük seviyesine getirilmektedir. Daha sonra aynı çözünürlükte olan MS görüntü ile PAN görüntü birleştirme (C, concatanate) işlemi ile bir araya getirilmektedir. MS görüntü (3 bantlı) ve PAN (1 bantlı) görüntü birleştirildiğinde 4 bantlı bir görüntü uzayı oluşmaktadır. Birleştirilmiş olan 4 bantlı görüntü derin öğrenme modeline gönderilmektedir. İlk katmanda Çizelge 2.2'de olduğu gibi 64 adet

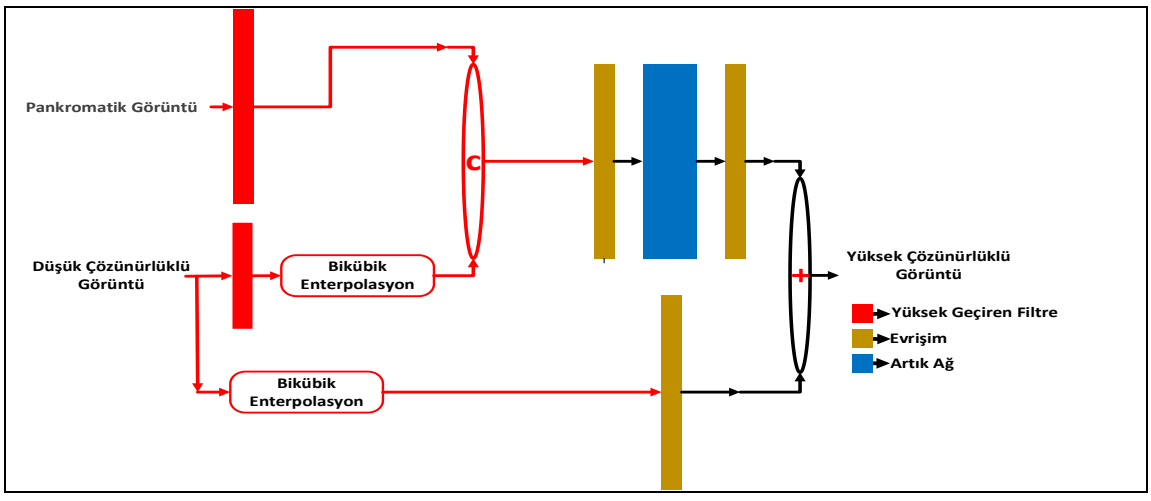
filtreden oluşan 9x9 çekirdek boyutlu evrişim katmanı bulunmaktadır. Evrişim katmanının çıkışına modelin ezberlemesini önlemek adına aktivasyon fonksiyonu olan ReLU fonksiyonu uygulanmaktadır. İkinci ve üçüncü katman bilgileri Çizelge 2.2'deki gibidir. Üçüncü katmanın çıkışı çözünürlüğü yükseltilmiş RGB renk uzayında görüntü olmaktadır. PNN modeli basit bir modeldir, bu nedenle toplam parametre sayısı 74435 olacak şekilde düşüktür. PNN modeline ait özellikler Çizelge 2.2'de verilmiştir.

Çizelge 2.2. PNN modeline ait özellikler

Yöntem	Katman	Filtre Sayısı	Çekirdek Boyutu	Toplam Parametre Sayısı
PNN	Evirişim, ReLU	64	9x9	74435
	Evirişim, ReLU	32	5x5	
	Evirişim	3	5x5	

2.7.10. PanNet

PanNet yöntemi pankeskinleştirmenin iki amacı olan hem spektral hem de uzaysal özelliklerin korunmasına odaklanmıştır. Uzaysal yapıyı korumak için hem PAN görüntüsünden hem de düşük çözünürlüklü MS görüntüsünden yüksek geçiren filtre çıktıları kullanılmıştır. Spektral yapıyı korumak için, MS görüntüsünün doğrudan çözünürlüğünün yükseltilmiş hali (Yüksek çözünürlüklü referans görüntüsünün çözünürlüğüne ve boyutlarına) ağ çıkışına eklenir. Bu işlem için farklı katmanlar kullanılır. Şekil 2.13'te PanNet modeli verilmiştir.



Şekil 2.13. PanNet modeli

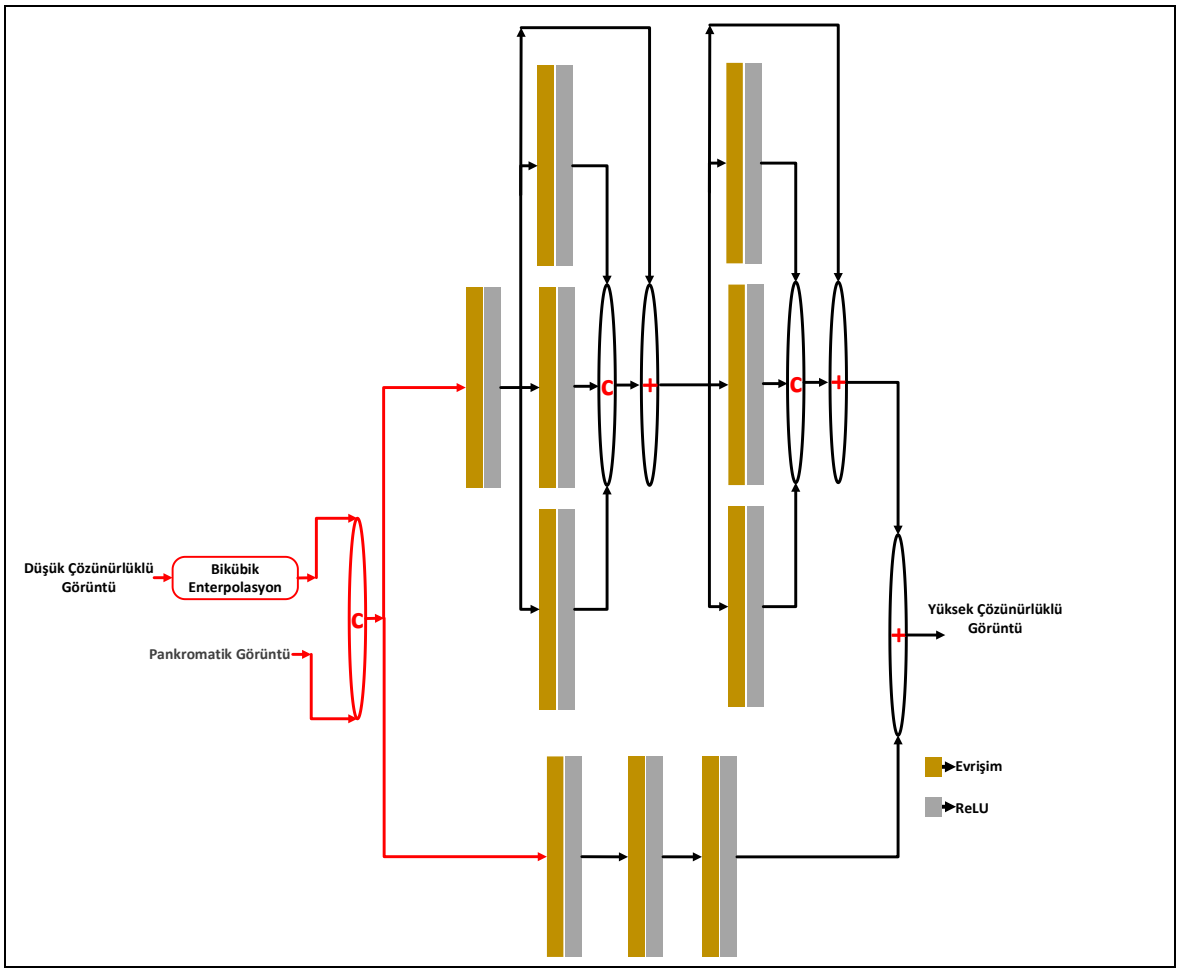
PanNet modeli Şekil 2.13'te verilen katmanlardan oluşmaktadır. Veri seti görüntüleri modele gönderilmeden önce hem MS görüntüye hem de PAN görüntüye yüksek geçiren filtre uygulanmaktadır. Yüksek geçiren filtre, görüntünün Gaussian filtresi (konvolüsyon çekirdek boyutu=5x5, σ=1) çıkışı, orijinal görüntüden çıkarılarak elde edilmiştir. Düşük çözününürlüklü MS görüntünün çözünürlüğü PAN görüntü ile eş seviyeye bikübik enterpolasyon ile getirilir. Daha sonra MS görüntü ile PAN görüntü C fonksiyonu ile birleştirilerek 4 bantlı görüntü model girişi olarak elde edilir. Yüksek frekanslı bileşenlerin bir araya getirildiği bu model girişine Çizelge 2.3'teki gibi sırasıyla Evrişim, 4 adet Artık Ağ katmanı ve Evrişim katmanı uygulanmaktadır. Son evrişim katmanı ile PanNet modelinin bir diğer model girişi düşük çözünürlüklü MS görüntünün PAN görüntü ile eş seviyedeki çözünürlüğe yükseltilmiş formu toplama fonksiyonu ile bir araya getirilir. Bu şekilde nihai model çıktısı olan yüksek çözünürlüklü MS görüntü elde edilir. Kullanılan artık ağ sayısına paralel olarak toplam parametre sayısı artmaktadır. Deney sonuçları için seçmiş olduğumuz 4 artık ağ'dan oluşan model için toplam parametre sayısı 57539 olarak elde edilmiştir.

Çizelge 2.3. PanNet modeline ait özellikler

Yöntem	Blok	Katman	Filtre Sayısı	Çekirdek Boyutu	Toplam Parametre Sayısı
PanNet		Evirişim	32	3x3	57539
	Artık Ağ x4	Evirişim, ReLU	32	3x3	
		Evirişim	32	3x3	
		Toplama			
		Evirişim	3	3x3	
		Toplama			

2.7.11. MSDCNN

MSDCNN (Multi-Scale Feature Extraction Block and Skip Connection) modelinde farklı çekirdek boyutları kullanılarak hem MS hem de PAN görüntülerinden özellik çıkarımını içermektedir. Model iki kola ayrılmaktadır, bir kolda görüntü özellikleri detay seviyede elde edilerek derinlemesine özellik çıkarımı hesaplanırken, diğer kolda görüntü özellikleri kaba seviyede elde edilerek sığ bir özellik çıkarımı elde edilir. Nihai durumda hem ince detaylar hem de kaba detaylar bir araya getirilip toplanarak yüksek çözünürlüklü MS görüntü elde edilir. Şekil 2.14'te MSDCNN modeli verilmiştir.



Şekil 2.14. MSDCNN modeli

MSDCNN modeli Şekil 2.14'te verilen katmanlardan oluşmaktadır. Düşük çözünürlüklü MS görüntü MSDCNN derin öğrenme modeline gönderilmeden önce çözünürlüğü bikübik enterpolasyon yöntemi ile pankromatik görüntünün çözünürlük seviyesine getirilmektedir. Daha sonra aynı çözünürlükte olan MS görüntü ile PAN görüntü birleştirme (C, concatanate) işlemi ile bir araya getirilmektedir. MS görüntü (3 bantlı) ve PAN (1 bantlı) görüntü birleştirildiğinde 4 bantlı bir görüntü uzayı oluşmaktadır. Birleştirilmiş olan 4 bantlı görüntü derin öğrenme modeline gönderilmektedir. Model iki koldan oluşmaktadır. İlk koldaki ilk katman İlk katmanda Çizelge 2.4'te olduğu gibi 60 adet filtreden oluşan 7x7 çekirdek boyutlu evrişim katmanı bulunmaktadır. Evrişim katmanının çıkışına modelin ezberlemesini önlemek adına aktivasyon fonksiyonu olan ReLU fonksiyonu uygulanmaktadır. Buradaki çıktısı 3 kola ayrılmaktadır. Bu 3 kol sırasıyla ince detaylardan kalın detaylara doğru odaklanan 3x3, 5x5 ve 7x7 çekirdek boyutlarından oluşan 20 adet filtre içeren evrişim katmanlarından oluşmaktadır. Bu 3 koldan elde edilen filtre çıktıları birleştirilerek giriş modeli olan 60 adet filtreden oluşan özellik haritası ile toplanmaktadır. Benzer şekilde 30

adet filtre, 3 kola ayrılarak birleştirilip tekrar giriş ile toplanmaktadır. İkinci kolda ise sırasıyla geleneksel evirişim yaklaşımı uygulanarak sığ özellik çıkarımı elde edilmiştir. Nihai durumda her iki kolun çıktısı toplanarak yüksek çözünürlüklü MS görüntüsü elde edilmektedir. MSDCNN modeli birçok filtreden oluşan bir model olduğu için parametre sayısı 163376 olarak elde edilmiştir.

Çizelge 2.4. MSDCNN modeline ait özellikler

Yöntem	Katman	Filtre Sayısı	Çekirdek Boyutu	Toplam Parametre Sayısı
MSDCNN	Evirişim, ReLU	60	7x7	163376
		20	3x3	
		20	5x5	
		20	7x7	
		Birleştirme		
		Toplama		
		30	3x3	
		10	3x3	
		10	5x5	
		10	7x7	
		Birleştirme		
		Toplama		
		3	5x5	
		64	9x9	
		32	1x1	
	Evirişim	3	5x5	
	Toplama			

2.7.12. SRPPNN

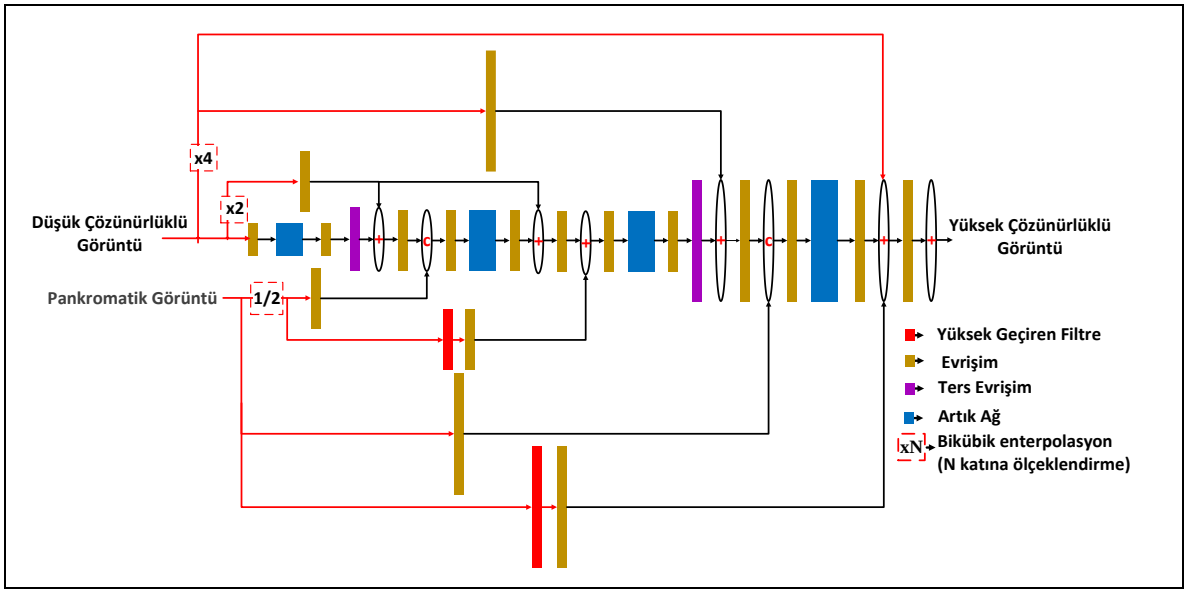
Derin öğrenme tabanlı pankeskinleştirme modeli, Cai, J., ve Huang, B'nin sunduğu "Derin Evrişimli Sinir Ağına Dayalı Süper Çözünürlük Kılavuzlu Aşamalı Paneskinleştirme" [6] yöntemi referansında uyarlanmıştır. Bu derin öğrenme yöntemi düşük çözünürlüklü görüntünün kendisi, bikübik enterpolasyon ile 2 ve 4 katına ölçeklendirilmiş halini giriş olarak almaktadır. Diğer yandan pankromatik bandın kendisi ile yüksek geçiren filtre çıktısını ve bikübik enterpolasyon ile ½ katına ölçeklendirilmiş halini ve yüksek geçiren filtre çıktısını giriş olarak almaktadır. Yüksek geçiren filtre, görüntünün Gaussian filtresi

(konvolüsyon çekirdek boyutu=5x5, σ=1) çıkışı, orijinal görüntüden çıkarılarak elde edilmiştir. Yöntemde özellik çıkarımı ve yeniden yapılandırma için sıralı Artık Ağ (ResNet) blokları kullanılmıştır. Önerdiğimiz yöntemde ise Artık Ağ mimarisinin avantajlarının yanı sıra Artık Yoğun Ağ katmanı mimarisinin avantajlarını ekleyebilmek adına, Şekil 2.15.a'da yer alan iki adet Artık Ağ katmanının yerini Şekil 2.15.b'deki gibi Artık Yoğun Ağ katmanı almıştır. Artık Yoğun Ağ katmanı, Süper Çözünürlük için etkileyici sonuçlar veren Zhang, Y. ve arkadaşlarının sunduğu 'Süper Çözünürlük için Artık Yoğun Ağ' [15] mimarisi ayrıntılarının Bölüm 2.5.8'de yer aldığı gibi kullanılmıştır. Artık Yoğun Ağ katmanında her katman, önceki tüm katmanlardan ek girdiler alır ve kendi özellik haritalarını sonraki tüm katmanlara iletir. Bu yapı sayesinde özellik haritası zenginleştirilmiş olur. Çizelge 2.5'te SRPPNN modelinin katmanları verilmiştir. Modele ait toplam parametre sayısı 467811'dir.

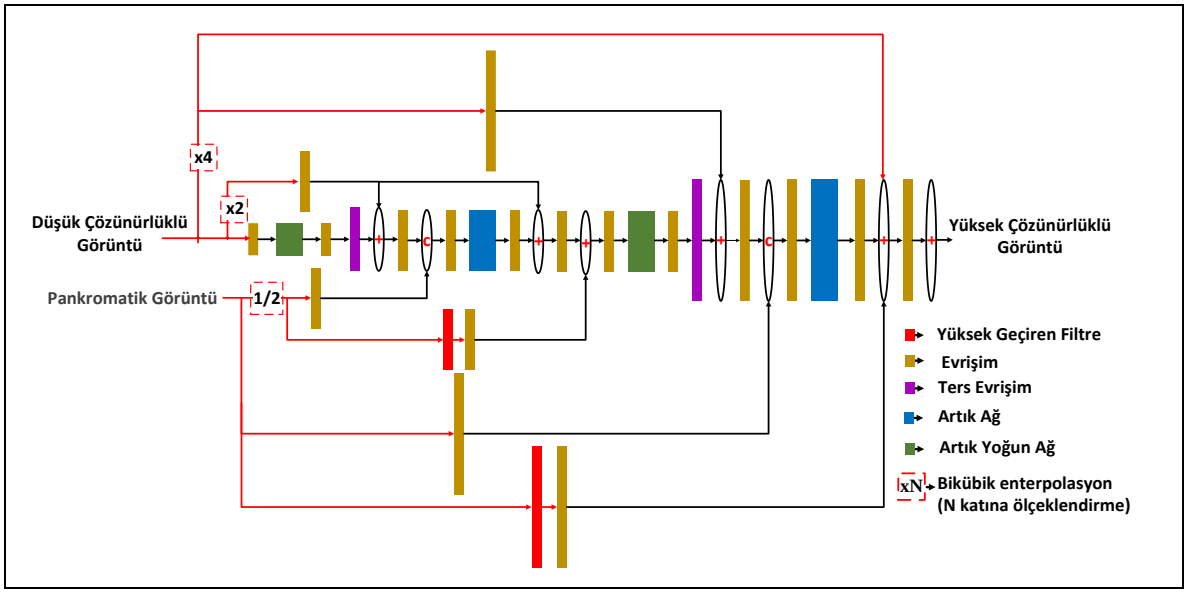
Landsat-8 uydusu için multispektral görüntü ile pankromatik görüntü arasında iki katı çözünürlük oranı bulunmaktadır. Pankromatik görüntü ile eş çözünürlük seviyesine taşıyabilmek adına multispektral görüntünün çözünürlüğünün iki katına çıkarılması gerekmektedir. Bu nedenle Landsat-8 uydu görüntüleri için önerilen modelin Şekil 2.16'daki gibi ilk 2 fazı uygulanmıştır.

2.8. Vitis AI

FPGA ortamı paralel işlem yapabilme kabiliyeti nedeniyle CPU ve GPU ortamına göre daha hızlı işlem sonuçları sunabilmektedir. Derin öğrenme modelleri birçok matematiksel hesap ihtiyacı doğurduğundan, işlem süresi uzamaktadır. Derin öğrenme modelleri için işlem süresi, modelin çıktı doğruluğunun yanı sıra önemli bir kriterdir. Bu ihtiyacı karşılamak için modelin hesaplama süresine ait yükü hafifletmek adına FPGA platformu tercih edilmektedir. Modellerin FPGA ortamına entegre edilebilmesi, önemli derecede FPGA bilgisi gerektirmektedir. FPGA bilgisine sahip olmak uzun süreli tecrübe ve araştırma ihtiyacı anlamına gelmektedir. Bu soruna cevap bulabilmek adına tasarımcılar AMD FPGA'ları için özelleştirilmiş bir Vitis AI (Artificial intelligence) ortamı sunmaktadır. Vitis AI geliştirme ortamı, detaylı FPGA bilgisi olmayan tasarımcıların bile ortamın avantajlarından faydalanabilmesini sağlamaktadır. Vitis AI Docker platformu gerekli olan tüm bağımlı dosyaların kurulabilmesi için hazır bir platform olarak sunulmaktadır. Docker platformu hem CPU hem de GPU kullanımı için Tensorflow ve PyTorch kütüphanelerini ve modelin FPGA ortamına entegre edilebilmesi için gerekli olan araçların kurulumunu sağlar. Kurulu



(a)



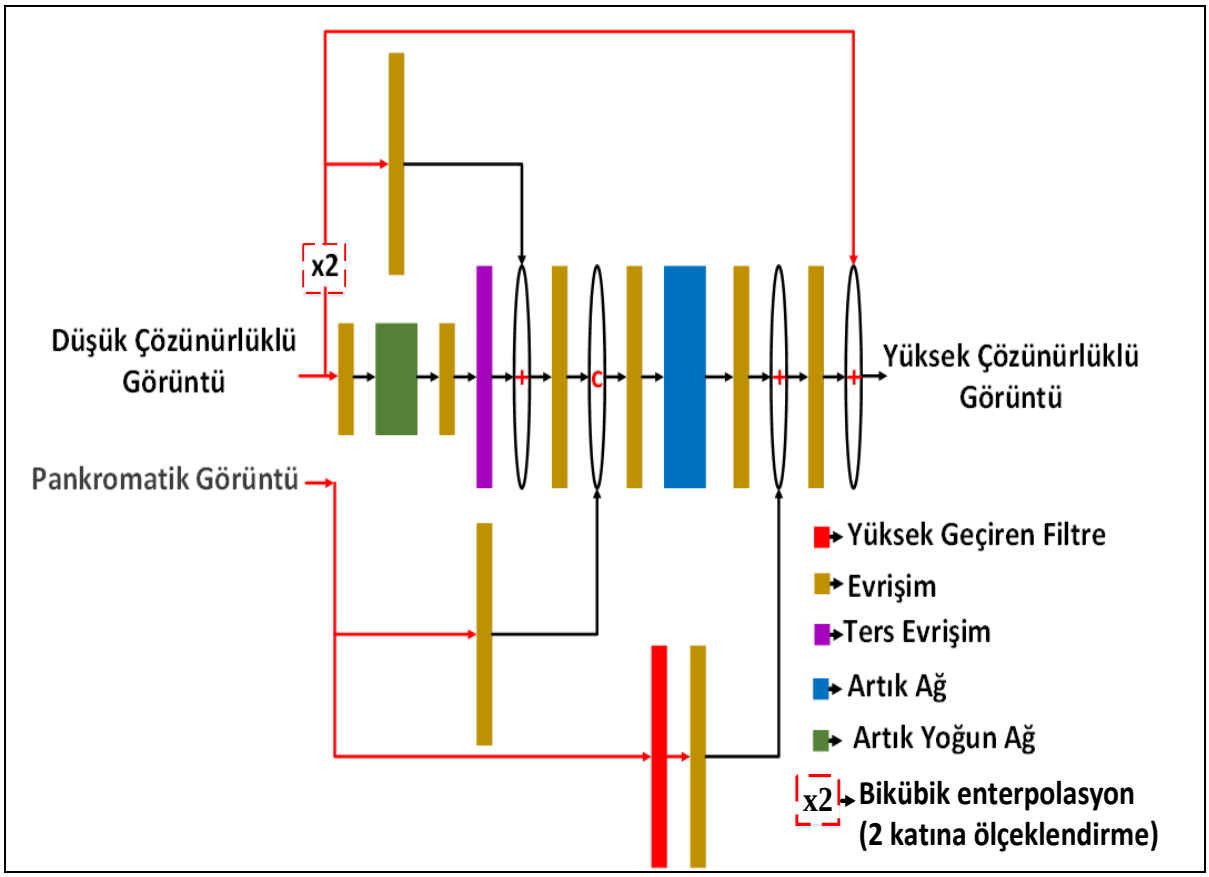
(b)

Şekil 2.15. SRPPNN mimarisi: (a) SRPPNN modeli, (b) SRPPNN-RDN Modeli

olan kütüphaneler sayesinde model eğitimleri Docker platformunda gerçekleştirilebilmektedir. Vitis AI, optimize edilmiş IP (Intellectual Property) çekirdeklerinden, araçlardan, kitaplıklardan ve modellerden oluşmaktadır. Modelin kuantalanma işlemi Vitis AI Kuantalayıcı ile yapılır. Vitis AI Kuantalayıcı, 32 bit kayan nokta ağırlıklarını, 8 bitlik sabit noktalı tamsayıya dönüştürür. Sabit noktalı ağ modeli daha hızlı ve daha yüksek güç verimliliği sağlamaktadır [26].

Çizelge 2.5. SRPPNN modelinin özellikleri

Yöntem	Blok	Katman	Filtre Sayısı	Çekirdek Boyutu	Toplam Parametre Sayısı
SRPPNN	Evirişim x3	Evirişim	32	1x1	467811
	Evirişim x3	Evirişim	32	3x3	
	Artık Ağ x4	Evrişim, ReLU			
		Evirişim			
		Toplama			
		Evirişim	32	3x3	
		Ters Evrişim	32	8x8	
		Toplama			
		Evirişim	32	3x3	
		Birleştirme			
	Artık Ağ x4	Evirişim	32	3x3	
		Evrişim, ReLU			
		Evirişim			
		Toplama			
		Evirişim	32	3x3	
		Toplama			
		Evrişim	32	3x3	
		Toplama			
	Artık Ağ x4	Evirişim	32	3x3	
		Evirişim, ReLU			
		Evirişim			
		Toplama			
		Evirişim	32	3x3	
		Ters Evirişim	32	8x8	
		Toplama			
		Evirişim	32	3x3	
		Birleştirme			
	Artık Ağ x4	Evirişim	32	3x3	
		Evirişim, ReLU			
		Evirişim			
		Toplama			
		Evrişim	32	3x3	
		Toplama			
		Evrişim	3	3x3	
		Toplama			



Şekil 2.16. Önerilen modelin ilk iki fazı

Modelin derlenmesi VAI (Vitis Artificial intelligence) Derleyici ile yapılır. VAI (Vitis Artificial intelligence) Derleyici, kuantalanmış modeli giriş olarak alarak, hedef platform olan FPGA'de tanımlı Derin Öğrenme İşlem Birimi (Deep Learning Processing Unit, DPU)'de koşacak şekilde "xmodel" uzantılı bir model çıktısı üretir. Derleyici araç, 'vai_quantizer' ile kuantalanmış giriş modelini dönüştürerek grafik elde eder. Grafiğe çeşitli optimizasyon teknikleri uygulayarak ve işlemin DPU üzerinde yürütülebilme desteğine bağlı olarak alt grafiklere böler. DPU alt grafikleri için derleyici talimat akışı oluşturularak VART (Vitis AI Run Time) için ihtiyaç duyulan bilgiler eklenerek derlenmiş 'xmodel' dosyası elde edilir.

Vitis AI kitaplığı, DPU'ların kullanımı için geliştirilmiş, karmaşık üst düzey kitaplıkların kullanımını basitleştirilen bir dizi kitaplık setidir. Bu kitaplık seti Xilinx kartlarında çalışmak üzere Xilinx Çalışma Zamanı (XRT) desteklemektedir. Vitis AI çalışma zamanı ise C++ ve Python dilin yazılmış uygulamalar için çoklu iş parçacığı ve işlem yürütme desteği sunan uygulama programlama ara yüzüdür.

2.8.1. Derin öğrenme işlem birimi

Derin Öğrenme İşlem Birimi, FPGA üzerinde çalışması için geliştirilmiş ve evrişimli sinir ağlar için optimize edilmiş olan konfigüre edilebilir bir IP çekirdeğidir. Xilinx KRIA KV260 kartı için tasarlanan DPU ise DPUCZDX8G' [27] dir. DPU, yapılandırılabilir 8 adet farklı mimariye sahiptir. Bu mimariler; B512, B800, B1024, B1152, B1600, B2304, B3136 ve B4096'dır. Mimarilerin isimlendirilmesi DPU saat döngüsü başına toplam işlem sayısını gösterir. DPU, konvolüsyon, ReLU, ters evrişimli konvolüsyon ve çarpma gibi temel operatörleri desteklemektedir.

DPU, işlemci ile FPGA arasında AXI ara yüzü ile birbirine bağlanmıştır. DPU, derlenmiş sinir ağı grafiğinden oluşmuş modeli (xmodel uzantılı) çalıştırır ve giriş ve çıkış görüntülerini saklamak için harici geçici belleğe ihtiyaç duyar. Ayrıca DPU içerisinde, dahili bellek bulunmaktadır. Dahili bellek, yüksek verimlilik elde etmek amacıyla giriş aktivasyonlarını ve ara özellik haritalarını tamponlamak için kullanılır. Veri aktarımının koordine edilmesi Uygulama İşlem Birimi (APU) tarafından sağlanır. Konvolüsyon motoru, İşlem Motoru (PE) ve genel amaçlı işlemler yapan hesaplayıcıdan (Misc Engines) oluşmaktadır. İşlem Motoru (PE), model ağırlıklarının hesaplandığı yerdir ve birçok çarpan ve toplayıcıdan oluşmaktadır. Yüksek hızlı boru hattı, İşleme Motoru tarafından kullanılmaktadır. FPGA içerisindeki kaynak kullanımını optimize etmek ve model için özgünleştirmek adına yapılandırılabilen parametreleri kullanmak gerekmektedir. Bu parametreler sırasıyla açıklanacaktır.

DPU Çekirdek Sayısı kavramı açıklanırsa, bir DPU IP'si en fazla dört çekirdeğe kadar çalıştırılabilir. Çekirdek sayısının arttırılmasının nedeni model performansını arttırmak adına bir yöntem olabilir. Fakat çekirdek sayısı arttıkça DPU FPGA'da daha fazla kaynak kullanımına ihtiyaç duyacaktır.

DPU mimari seçimi için DPU IP kullanılmıştır. DPU IP, evrişim işlemi paralelliği ile ilgili çeşitli evrişim mimarilerine sahiptir. Bu mimariler B512, B800, B1024, B1152, B1600, B2304, B3136 ve B4096'dır. DPUCZDX8G evrişim mimarisinde paralelliğin üç boyutu vardır: piksel paralelliği, giriş kanalı paralelliği ve çıkış kanalı paralelliğidir. Her saat döngüsünde, evrişim dizisi iki işlem olarak sayılan bir çarpma ve bir toplama işlemi gerçekleştirir. Dolayısıyla döngü başına en yüksek işlem sayısı (Piksek Paralelliği)*(Giriş

Kanalı Paralelliği)*(Çıkış Kanalı Paralelliği)*2'ye eşittir. DPU IP mimari çeşitleri Çizelge 2.6'da verilmiştir.

Kaynak kullanımı KV260 kartı için Çizelge 2.7'de verilmiştir. DPU konfigürasyonu, DPU çekirdek sayısı '1', 'düşük' RAM kullanımı, kanal büyütme 'aktif', DepthwiseConv (ALU) özelliği 'aktif', ElementWise Çarpma özelliği 'pasif', AvaragePool özelliği 'aktif', ReLU tipi 'ReLU + LeakyReLU + ReLU6' ve SoftMax çekirdek sayısı 'sıfır' olacak şekilde seçilmiştir.

Çizelge 2.6. DPU IP mimarileri [27]

DPUCZDX8G Mimarisi	Piksel Paralelliği	Giriş Kanalı Paralelliği	Çıkış Kanalı Paralelliği	En Yüksek İşlem Sayısı
B512	4	8	8	512
B800	4	10	10	800
B1024	8	8	8	1024
B1152	4	12	12	1152
B1600	8	10	10	1600
B2304	8	12	12	2304
B3136	8	14	14	3136
B4096	8	16	16	4096

Çizelge 2.7. DPU mimarileri özellikleri

DPU Mimarisi	LUT		LUTRAM		FF		BRAM		URAM		DSP	
B512	25933	117120	1810	57600	33681	234240	1,50	144	18	64	110	1248
B800	28127		2142		40691		1,50		40		157	
B1024	31844		2358		47733		1,50		26		218	
B1152	30560		2498		47196		2,00		44		212	
B1600	35200		2818		59063		15,00		50		312	
B2304	38611		3278		68799		29,50		50		422	
B3136	42938		3780		79950		47,50		50		548	
B4096	48280		5598		98550		69,50		50		690	

RAM kullanımı için, ağırlıklar, önyargı ve ara özellik haritaları çip üzerinde dahili belleğe alınır. Çip üstü bellek, blok RAM ve UltraRAM olarak iki RAM'den oluşur. RAM kullanım seçeneği, çip üzerindeki belleğin toplam miktarını belirler. Yüksek RAM kullanımı, daha yüksek bellek bloğuna sahip olduğu için ara verileri işlemede daha dazla esneklik sağladığı için yüksek performans anlamına gelir. Kanal artırma ise CNN'deki ilk katmanın kanal

sayısı kanal paralelliğinden daha büyük ise kullanılmaktadır. Kanal artırımı kaynak kullanımını arttırmaktadır.

Derinlemesine ayrılabilir evrişim işlemi derinlemesine evrişim ve noktasal evrişim olmak üzere iki adımda gerçekleşir. Her bir özellik haritası için ayrı ayrı derinlemesine evrişim gerçekleştirildikten sonra çekirdek boyutu 1x1 noktasal evrişimi gerçekleştirilmektedir. Piksel paralelliğinin yarısı kadar derinlemesine evrişimin paralelliği elde edilir. ElementWise Çarpma, iki giriş özelliği haritasının Hadamard çarpımını sağlar. Ortalama havuz seçeneği, ortalama havuzlama işleminin gerçekleşme durumunu belirler. DPU ile kullanılabilecek belirli ReLU etkinleştirme işlevlerini belirlemektedir. Bunlardan biri ReLU ve ReLU6 diğeri 'ReLU + LeakyReLU + ReLU6'dur. LeakyReLU katsayısı, 26/256 = 0,1015625 olarak tahmin edilen 0,1'e sabitlenmiştir. Softmax operatörünün donanım uygulamasını etkinleştirilmesi, donanım softmax hızlandırıcısını etkinleştiriir. Softmax'ın donanım uygulaması, MPSoC yazılım uygulamasından 160 kat daha hızlı olabilmektedir.

2.8.2. Vivado tasarımı

Deney sonuçlarını elde etmek için kullanılan Vivado tasarımı, Vivado 2021.1 üzerinde gerçekleştirilmiştir. Vivado tasarımında işlemci olarak 'Zynq UltraScale+ MPSoC' IP (Intellectual Property)'si eklenmiştir. ZynqMP işlemci IP bloğu, KV260 PetaLinux BSP (Board Support Package) için gerekmektedir. Vivado tasarımına, derin öğrenme modellerinin üzerinde çalışacağı Derin Öğrenme İşlem Birimi (Deep Learning Processing Unit, DPU) eklenmiştir. Tasarımın saat (clock) ihtiyacı için 'Clocking Wizard' IP bloğu, sıfırlama (reset) ihtiyacı için 'Processor System Reset' IP'si eklenmiş ve gereken bağlantılar yapılmıştır. İşlemci Birimi ile FPGA içerisinde yer alan DPU bloğu AXI arayüzleri ile haberleşmektedir. DPU frekansı 275 Mhz'dir. DPU IP'sine ait konfigürasyon seçimi ise DPU çekirdek sayısı '1', DPU mimarisi seçimi 'B4096', ram kullanımı, 'düşük', kanal artırma, 'aktif', DepthwiseConv (ALU) özelliği 'aktif', ElementWise Çarpma özelliği 'pasif', AvaragePool özelliği 'aktif', ReLU tipi 'ReLU + LeakyReLU + ReLU6' ve SoftMax çekirdek sayısı 0 olacak şekilde konfigürasyon gerçekleştirilmiştir. Ayrıca DPU mimarisi seçimi için 'B2304' ve 'B3136' içinde donanım oluşturulmuştur.

Bir Vivado projesinin donanım tasarımını PetaLinux projesinde kullanmak için donanım platformu olarak Xilinx'e özel bir XSA (Xilix support archive) içerinde paketlenmesi

gerekmektedir. Bu dosya vivado tasarımı sentezlenip bit dosyası elde edildikten sonra donanım dışarı aktarılarak oluşturulmuştur.

2.8.3. Petalinux kurulumu

KV260 kartı, birincil ve ikincil önyükleme cihazlarını kullanarak iki adımlı bir önyükleme işlemi sağlamaktadır. Birincil önyükleme cihazı Sistem Üstü Modül (System On Module, SOM) üzerindeki bir QSPI (Quad serial peripheral interface) belleğidir ve ikincil önyükleme cihazı ise taşıyıcı kart üzerindeki bir SD (Secure Digital) bellek karttır [28]. Birincil önyükleme cihazı, birinci aşama önyükleyici ürün yazılımı (FSBL), platform yönetim birimi ürün yazılımı (PMU), arm güvenilir ürün yazılımı (ATF) ve ikinci aşama önyükleme yükleyicisi (U-BOOT) dahil gerekli tüm modülleri içeren bir BOOT.BIN dosyası içermektedir. İkincil önyükleme cihazı, yani SD (Secure Digital) bellek kartı, genellikle iki bölüme sahiptir. İlk bölüm Linux imajı (Image) ve cihaz üç dosyasını (Device Tree Binary, DTB) içerir, ikinci bölüm ise kök dosya sistemlerini ve uygulamalarını içerir. Birincil ve ikincil ön yükleme cihazını kullanarak KV260 kartı başlatılmaktadır.

Petalinux kurulumu Petalinux 2021.1 aracı ile gerçekleştirilmiştir. Petalinux aracı Ubuntu 18.04.05 terminali üzerinden başlatılmıştır. Bu fonksiyon için ‘source <kurulu olan Petalinux_2021_1 yolu>settings.sh’ komut satırı terminal üzerinden yazılmıştır. Xilinx’in sunmuş olduğu KV260 kartına ait BSP (Board Support Package) dosyası kullanılarak bir petalinux projesi oluşturulmuştur. Bu fonksiyon için ‘petalinux-create -t project -s xilinx-k26-som-v2021.1-updated-final.bsp’ komut satırı terminal üzerinden çalıştırılmıştır. Oluşturulan petalinux projesine, Vivado tasarımından elde edilen ‘xsa’ uzantılı dosyanın bulunduğu konum yazılarak, oluşturulan donanım içe aktarılmıştır. Bu fonksiyon için ‘petalinux-config --get-hw-description=../VIVADO-DPU/ --silentconfig’ komut satırı terminal üzerinden yazılmıştır. İlerleyen adımlarda konfigürasyon kapsamında ‘petalinux-config –c rootfs’ komut satırı ile ihtiyaç duyulan paketler seçilmiştir. Son olarak ‘petalinux build’ komut satırı ile petalinux işletim sistemi inşa edilmiştir. Bu işlem bilgisayar performansı ve internet hızına bağlı olarak yaklaşık 1 saat sürmüş ve petalinux inşa edilirken yaklaşık olarak 60 GB’lık dosya oluşmuştur. Oluşturulan dosyalardan ‘../images/linux’ dizininde önyükleyici için ihtiyaç duyulan ‘BIN’ uzantılı ikili dosya sistemi oluşturulmuştur. Bunun için ‘petalinux-package --boot --fsbl zynqmp_fsbl.elf --u-boot u-boot.elf --pmufw pmufw.elf --fpga system.bit’ komut satırı girilerek BOOT.BIN önyükleme dosyası elde

edilmiştir. BOOT.BIN dosyası KV260 kartı için varsayılan olarak QSPI (Quad serial peripheral interface) belleğinden çekilerek başlatılacaktır. Nihai durumda SD karta yazmak için hazırlanacak olan ‘wic’ uzantılı dosya ‘petalinux-package --wic --bootfiles ‘ramdisk.cpio.gz.u-boot BOOT.BIN boot.scr Image system.dtb’ komut satırı girilerek yaklaşık 6,5 GB’lık bir işletim sistemi dosyası elde edilmiştir. Elde edilen dosya SD karta taşınarak, işletim sistemi kurulumu tamamlanmıştır.

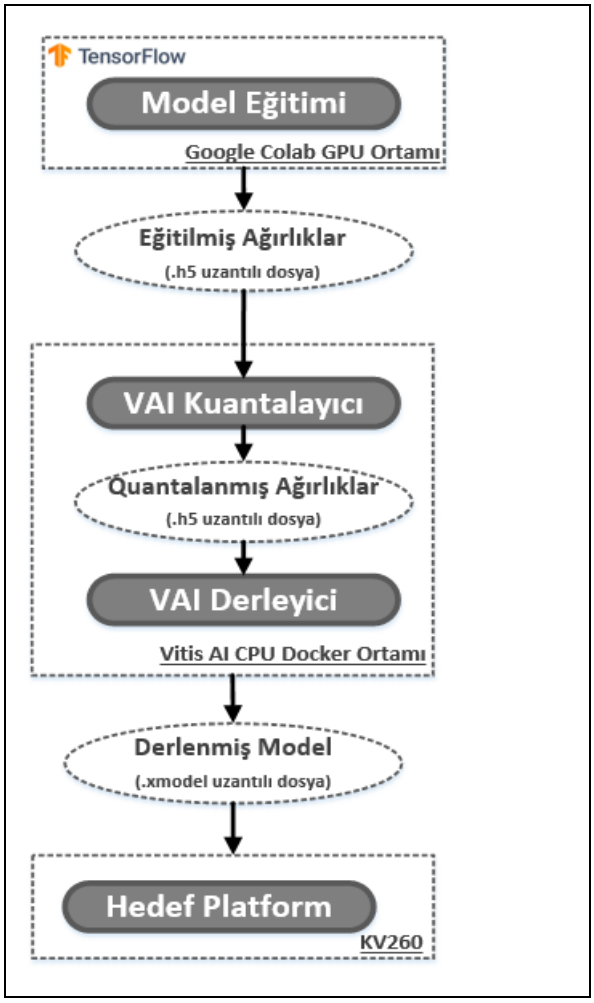
2.8.4. Modelin FPGA ortamına entegrasyonu

Önerilen model ve 4 farklı derin öğrenme tabanlı pankeskinleştirme yöntemi Şekil 2.17’deki akış kullanılarak ilk olarak Google Colab GPU ortamında modellerin eğitimi gerçekleştirilmiştir. Eğitilmiş ağırlıklar Vitis AI CPU Docker ortamı kullanılarak kuantalanmış ve derlenmiş model elde edilmiştir.

Modelin eğitimi Tensorflow’un kütüphanesi kullanılarak Google Colab ortamında gerçekleştirilmiştir. Tensorflow kütüphanesinin 2.10 versiyonu kullanılarak derin öğrenme tabanlı pankeskinleştirme yöntemleri kodlanmıştır. Google Colab ortamı Tesla T4 GPU, 12 GB RAM ve 107.71 GB disk alanı sunmaktadır. Model eğitimi için kullanılan hiperparemetreler Çizelge 2.8’deki gibidir.

Makine öğrenimi bağlamında örnek kümesi boyutu, model eğitimi sırasında tek bir gradyan iniş veya gradyan iniş algoritmasında kullanılan eğitim örneklerinin sayısını ifade eder [29]. Bir makine öğrenimi modelini eğitirken, veri kümesi genellikle tek seferde işlenemeyecek kadar büyüktür. Bunun yerine, veriler daha küçük kümelere veya kümelere bölünür ve her küme model parametrelerini güncellemek için kullanılır. Uygun örnek kümesi boyutu, veri kümesinin boyutuna ve karmaşıklığının yanı sıra mevcut donanım ve hesaplama kaynaklarına bağlıdır. Genellikle model seçim sürecinde ayarlanması gereken bir hiperparametredir. Örnek kümesi boyutu değerinin 1 yerine 16 seçilmesi eğitimin hızını arttırmak ve ara sonuçları depolamak için daha fazla bellek gereksinimini ortadan kaldırmaktır. Aksi takdirde bu durum daha yavaş yakınsamaya ve hatta fazla uydurmaya neden olabilmektedir.

Makine öğrenimi bağlamında, öğrenme hızı, eğitim sırasında bir modelin parametrelerinin güncellendiği adım boyutunu kontrol eden bir hiperparametredir.



Şekil 2.17. Pankeskinleştirme yöntemlerinin FPGA'ya entegrasyon akışı

Çizelge 2.8. Model eğitimi için kullanılan hiperparametreler

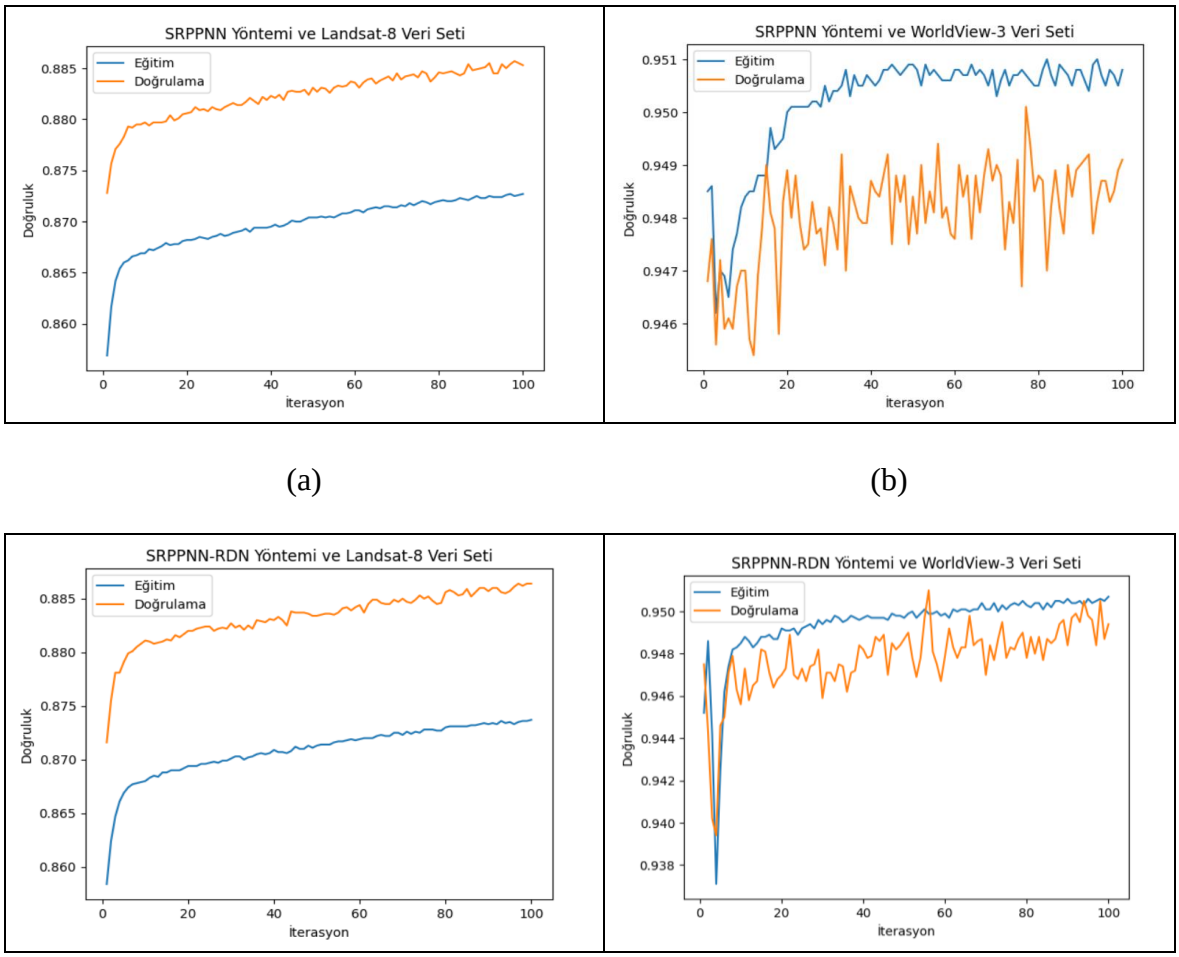
Parametre	Değer
Optimizasyon yöntemi	ADAM
Örnek kümesi boyutu	16
Iterasyon sayısı	100
Öğrenme oranı	0.0001

Bir modelin minimum kayıp fonksiyonuna ne kadar hızlı veya yavaş yakınsadığını belirler. Eğitim sırasında model, eğitim verileri üzerinde tahminler yapar, bunları gerçek etiketlerle karşılaştırır ve kaybı hesaplamaktadır. Öğrenme oranı daha sonra, kaybı en aza indirmek için model parametrelerini güncellemek için kullanılmaktadır. Öğrenme oranı çok küçükse, modelin yakınsaması uzun zaman alacaktır ve çok büyükse, model minimumu aşabilir ve yakınsamayabilir. Bu nedenle, verimli ve etkili model eğitimi için uygun bir öğrenme oranı belirlemek çok önemlidir. Optimum derin öğrenme oranı deneme yanılma yoluyla veya

modelin performansına bağlı olarak seçilmektedir. Makine öğrenimi bağlamında, optimize edici, modelin tahmin edilen çıktıları ile gerçek çıktılar arasındaki hatayı en aza indirmek amacıyla bir makine öğrenimi modelinin parametrelerini ayarlamak için kullanılan bir algoritmadır. Optimizasyon süreci tipik olarak model parametrelerine göre bir kayıp fonksiyonunun gradyanlarının hesaplanmasını ve bu gradyanların parametreleri kaybı azaltacak şekilde güncellemek için kullanılmasını içerir. Optimizasyon fonksiyonu olarak Adam algoritması seçilmiştir. Adam optimizasyon fonksiyonundaki yaklaşım, her parametre için farklı gradyan türlerine ve farklı güncelleme büyüklüklerine uyum sağlamasına olanak tanımasıdır. Verimli stokastik optimizasyon sağlayan ADAM [31] optimizasyon yöntemi az bellek gereksinimi ile yalnızca birinci dereceden gradyanlar gerektirir. Literatürde en çok kullanılan ve modelin doğru sonuca daha hızlı yakınsaması nedeniyle model eğitimi için ADAM optimizasyon yöntemi seçilmiştir.

Bir iterasyon, bir modelin eğitimi sırasında tüm veri kümesi boyunca bir tam döngüyü ifade etmektedir. Bir modeli eğitmek için gereken iterasyon sayısı, sorunun karmaşıklığına ve veri kümesinin boyutuna bağlıdır.

WorldView-3 uydu görüntüleri kullanarak giriş MS görüntüsünü 4 katı çözünürlüğe taşıdığımız SRPPNN yönteminde 100 iterasyon için toplam eğitim süresi 88 dakika 2 saniye, RDN uyarlaması için SRPPNN yönteminde 117 dakika 29 saniye sürmüştür. Landsat-8 uydu görüntüleri kullanarak giriş MS görüntüsünü 2 katı çözünürlüğe taşıdığımız SRPPNN yönteminde 100 iterasyon için toplam eğitim süresi 67 dakika 5 saniye, RDN uyarlaması için SRPPNN yönteminde 96 dakika 22 saniye sürmüştür. Modele kazandırılan artık yoğun ağ katmanı model eğitim süresini uzatmıştır. Bunun sebebi artık yoğun ağ katmanlarında önceki özellik haritasının sonraki katmanlara taşınması ile daha çok özellik haritası sonuçları kullanılarak işlem yapılmasıdır. Resim 2.2'de 100 iterasyon için eğitim ve doğrulama aşamalarına ait doğruluk sonuçları yer almaktadır. Eğitim ve doğrulama veri seti sürekli artış göstermektedir. Doğrulama veri seti düşüşe geçmediği için modelin eğitim veri setini ezberlediği söylenemez yani ezberleme problemi yoktur. Resim 2.2.a'da ve Resim 2.2.c'de doğrulama sonuçları eğitim sonuçlarından daha iyi doğruluk değerine sahiptir. Diğer yandan Resim 2.2.b ve Resim 2.2.d'de ise doğrulama sonuçları eğitim sonuçlarının altında kalmıştır. Bunun sebebi Landsat-8 veri setinin daha iyi kalitede olması ile ilgilidir. Ayrıca Resim 2.2.b'de SRPPNN yöntemi için doğrulama veri seti ile daha karmaşık bir eğri elde edilmiştir.



Resim 2.2. Model eğitimi ve doğrulama sonuçları: (a) Landsat-8 veri setinde SRPPNN yöntemi, (b) WorldView-3 veri setinde SRPPNN yöntemi, (c) Landsat-8 veri setinde SRPPNN yöntemi, (d) WorldView-3 veri setinde SRPPNN yöntemi

Fakat önerdiğimiz model Resim 2.2.d'deki gibi hem eğitim hem doğrulama için daha doğru ve düzgün bir şekilde artmakta olduğu gözlemlenmiştir.

Modelin hedef platforma hazırlanması için Linux Ubuntu 18.04.05 terminali üzerinden TensorFlow-2 kütüphanesi CPU ortamı için './docker_run.sh xilinx/vitis-ai-tf2-cpu' komutu çalıştırılmıştır. Model ağırlıklarının kuantalanması için ilk olarak kaydedilmiş olan kayan noktalı ağırlıklar yüklenerek, derin öğrenme modeli derlenmiştir. Derlenen model VitisQuantizer aracı kullanarak 8 bit tam sayı tabanlı sabit noktalı ağırlıklarla ile model kaydedilmiştir. Kuantalanmış model VAI Derleyicisi kullanılarak 'vai_c_tensorflow2 -m ${QUANT} \ -a ${ARCH}\ -o ${COMPILE_KV260} \-n ${NET_NAME} }' komutu çalıştırılmıştır. Komutta verilen '-m' değeri kuantalanmış ağırlıklı modeli içeren

kaydedilmiş model dosyasıdır. ‘-a’ değeri mimari tanımını içeren ‘json’ uzantılı dosyadır. Bu durumda KV260 için DPUCZDX8G olur. ‘-n’ ise kuantalanmış modelin son bağlantı katmanının adıdır. Son olarak ‘-o’ derlenmiş model çıktı dosyaların oluşturulduğu dosya adıdır. Dosya içerisinde derlenmiş model için hazırlanmış ‘meta.json’ dosyası ve ‘xmodel’ uzantılı derlenmiş model çıktısı oluşmaktadır. Derlenmiş model çıktıları ve test görüntüleri ethernet veri yolu üzerinden KV260 geliştirme kartına taşınarak, her bir model için özel olarak Python dilinde yazılmış olan uygulama kodları ile çalıştırılarak sonuçlar elde edilmiştir.

3. DENEYSEL SONUÇLAR

3.1. Metrikler

WorldView-3 ve Landsat-8 uydularından elde edilen görüntüler kullanılarak geleneksel ve derin öğrenme tabanlı pankeskinleştirme yöntemleri adına karşılaştırmanın yapılabilmesi için referanslı metrikler olan PSNR, RMSE, SSIM, CC, ERGAS, UIQI ve referanssız metrik olan BRISQUE kullanılmıştır.

3.1.1. PSNR ve MSE

En Büyük İşaret Gürültü Oranı (PSNR, Peak Signal-to- Noise Ratio) tepe sinyal-gürültü oranını temsil etmektedir. PSNR değerinin yüksek olması, sinyalin gürültüden daha güçlü olduğunu ve daha iyi bir görüntü kalitesini sunduğunu belirtmektedir. PSNR formülü Eş. 3.1'de verilmiştir.

$$PSNR = 10\log_{10}\frac{255}{\sqrt{MSE}} \tag{3.1}$$

Eş. 3.1'de L, bir görüntüdeki mümkün olan maksimum yoğunluk seviyelerinin sayısıdır (minimum yoğunluk seviyesinin 0 olduğu varsayılır). MSE ise ortalama karesel hatayı temsil eder ve formülü Eş 3.2'de verilmiştir.

$$MSE = \frac{1}{mn}\sum_{i=0}^{m-1}\sum_{j=0}^{n-1}\big(R(i,j) - G(i,j)\big)^2 \tag{3.2}$$

Eş. 3.2'de yer alan 'R', referans görüntüye ait matris verilerini temsil etmektedir. 'G', gözlemlenen görüntünün matris verilerini temsil etmektedir. 'm', piksel satırlarının sayısını 'i' ise görüntünün o anki satırına ait indeksini temsil etmektedir. 'n', piksel sütunlarının sayısını 'j' ise görüntünün o anki sütununun indeksini temsil etmektedir. MSE değeri ile PSNR değeri birbiri ile ters orantılıdır. MSE değerinin artması görüntünün referans görüntüden uzaklaştığı anlamına gelirken, PSNR değerinin de azalması anlamına gelmektedir. Diğer yandan MSE değerinin azalması görüntünün referans görüntüye yaklaştığı ve PSNR değerinin artması anlamına gelmektedir.

3.1.2. RMSE

Hata Kökü aslında Ortalama Kare Hatasının kareköküdür. RMSE Eş. 3.3'te verildiği gibi hesaplanmaktadır. Ortalama Karekök Hata (RMSE), referans veri noktaları ile tahmin edilen veri noktaları arasındaki hataların standart sapmasıdır . Başka bir deyişle, tahmin edilen veri noktalarının referans noktalarından ne kadar uzakta olduğunun bir ölçüsüdür. Buna ek olarak, tahmin edilen verilerin, referans verilerin oluşturduğu ideal çizgi etrafında ne kadar yoğunlaştığını gösterir.

$$RMSE = \sqrt{MSE} \tag{3.3}$$

3.1.3. SSIM

Yapısal benzerlik indeksi, (Structural Similarity Index Measurement, SSIM) referans görüntü ile gözlemlenen görüntü arasındaki benzerliği ölçmek için kullanılan bir metriktir. SSIM indeksi, piksellerin özellikle mekânsal olarak yakın olduklarında güçlü bir karşılıklı bağımlılığa sahip olduğu fikrini taşıyan yapısal bilgidir. Eş. 3.4'te verilen SSIM formülünde μ_R referans görüntünün ortalaması, μ_G gerçek görüntünün ortalaması, σ_R referans görüntünün standart sapması, σ_G gözlemlenen görüntünün standart sapması, c_1, c_2 yapısal benzerlik karşılaştırmasındaki kararlılığı sağlayan sabitlerdir. SSIM indeksine ait değer '0' ile '1' arasında değişmektedir. SSIM indeksinin '1' olması gözlemlenen görüntünün referans görüntüye mükemmel benzediğini göstermektedir.

$$SSIM = \frac{(2\mu_R\mu_G + C_1)(2\sigma_{RG} + C_2)}{(\mu_R^2 + \mu_G^2 + C_1)(\sigma_R^2 + \sigma_G^2 + C_2)} \tag{3.4}$$

3.1.4. ERGAS

Boyutsuz Küresel Göreli Sentez Hatası, (Erreur Relative Globale Adimensionnelle de Synthèse, ERGAS) Eş 3.5'te verilmiştir. Eş. 3.5'te yer alan 'h', PAN görüntüsüne ait olan uzaysal çözünürlüktür, 'l' ise MS görüntüsünün çözünürlüğüdür, 'B', bantların sayısıdır ve M_b^2, b bandının ortalamasına aittir. ERGAS değeri, RMSE değeri ile doğru orantılıdır. RMSE değerinin küçük olması gözlemlenen görüntünün referans görüntüye yaklaştığı anlamına geldiği gibi ERGAS değerinin de küçük olması istenir.

$$ERGAS = 100.\frac{h}{l}.\sqrt{\frac{1}{B}\sum_{b=1}^{B}\frac{(RMSE(b))^2}{M_b^2}} \quad (3.5)$$

3.1.5. CC

Korelasyon Katsayısı (Correlation Coefficient, CC) görüntü metriği, iki sinyal arasındaki benzerliği ölçmek için kullanılan matematiksel bir işlem olan çapraz korelasyona dayanmaktadır. Görüntü işleme bağlamında çapraz korelasyon, karşılık gelen konumlardaki piksel değerlerini karşılaştırarak iki görüntü arasındaki benzerliği ölçmek için kullanılmaktadır [32]. Eş. 3.6'da yer alan CC ifadesi, 'R' referans görüntü ve 'μR' referans görüntünün ortalaması, 'G' gözlemlenen görüntü ve 'μG' gözlemlenen görüntünün ortalamasıdır. CC metriğinin 1 değeri mükemmel benzerliği, 0 değeri korelasyonun olmadığını göstermektedir.

$$CC = \frac{1}{B}\sum_{b\in B}\frac{\sum_{i=1}^{N}\left(R_b-\mu_{R_b}\right)(G_b-\mu_{G_b})}{\sqrt{\sum_{i=1}^{N}(R_b-\mu_{R_b})^2}\sqrt{\sum_{i=1}^{N}(G_b-\mu_{G_b})^2}} \quad (3.6)$$

3.1.6. UIQI

Evrensel Görüntü Kalitesi İndeksi (Universal Image Quantity Index, UIQI) referanslı bir görüntü metriğidir. Referans görüntü ile test görüntüsü arasındaki korelasyon kaybı, parlaklık bozulması ve kontrast bozulmasını ölçmektedir [32]. UIQI skorunun '1' olması gözlemlenen görüntünün referans görüntüye mükemmel benzediğini göstermektedir.

$$Q = \frac{\sigma_{RG}}{\sigma_R\sigma_G}\cdot\frac{2\,\bar{R}\,\bar{G}}{(\bar{R})^2+(\bar{G})^2}\cdot\frac{2\sigma_R\sigma_G}{\sigma_R^2+\sigma_G^2} \quad (3.7)$$

Eş. 3.7'de yer alan 'R', referans görüntüye ait matris verilerini temsil etmektedir. 'G', gözlemlenen görüntünün matris verilerini temsil etmektedir. UIQI skorunun çıktısı Eş 3.7'de yer alan Q çıktısı olarak hesaplanmaktadır. Bu Eş üç bileşenden oluşmaktadır. İlk bileşen, R ile G arasındaki doğrusal korelasyonun derecesini ölçen, korelasyon katsayısıdır. [0, 1] değer aralığına sahip ikinci bileşen, ortalama parlaklığın R ile G arasında ne kadar yakın olduğunu ölçmektedir. σ_R ve σ_G, R ve G'nin kontrastının tahmini olarak değerlendirilir ve üçüncü bileşen, görüntülerin kontrastlarının ne kadar benzer olduğunu ölçmektedir.

3.1.7. BRISQUE

BRISQUE, görüntüye ait kalite değerini gösteren referanssız bir metriktir. BRISQUE, görüntünün kalitesini tahmin etmek adına eğitilmiş olan bir Destek Vektör Makinesi (Support Vector Machines, SVM)'den [33] faydalanmaktadır. Görüntünün kalitesi ile BRISQUE değeri ters orantılıdır, düşük BRISQUE değeri daha kaliteli görüntü anlamına gelmektedir.

3.2. Deneysel Sonuçlar

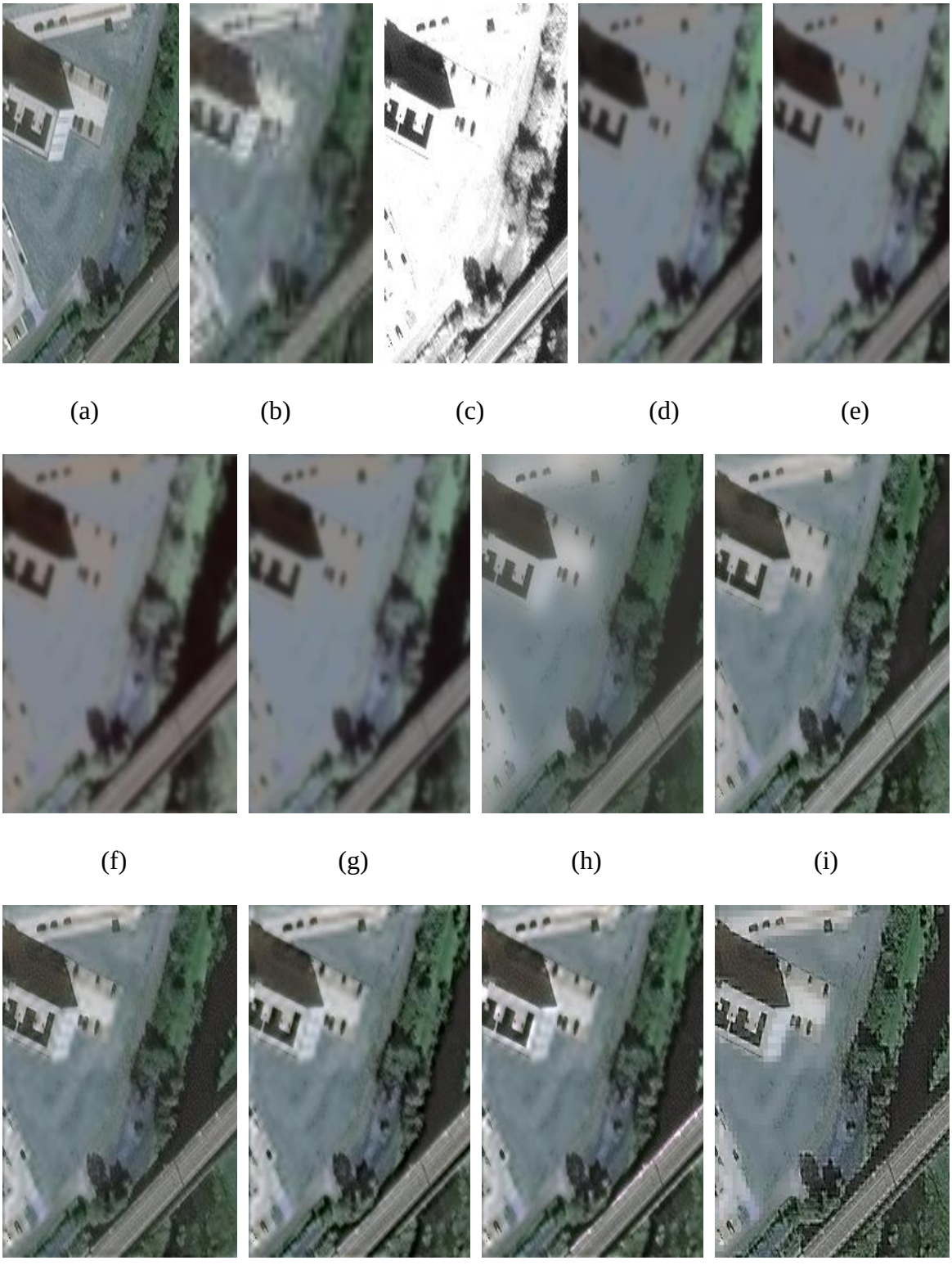
Sonuçlar için WorldView-3 veri seti ile Landsat-8 veri setine ait test görüntüler ayrı ayrı incelenmiştir. WorldView-3 veri setine ait geleneksel pankeskinleştirme yöntemlerinin sonuçları Resim 3.1'de, derin öğrenme yöntemlerinin sonuçları ise Resim 3.2'de verilmiştir. WorldView-3 veri setinde görüntülerin çözünürlüğü 4 kat artırılmıştır. Görüntülerin girdi boyutları 56 x 56 iken, çıkış boyutları 224 x 224 boyutlarındadır.

Landsat-8 veri setine ait geleneksel pankeskinleştirme yöntemlerinin sonuçları Resim 3.3'te verilmiştir. Landsat-8 veri setine ait derin öğrenme yöntemlerinin sonuçları ise Resim 3.4'te verilmiştir. Landsat-8 veri setinde görüntülerin çözünürlüğü 2 kat artırılmıştır. Görüntülerin girdi boyutları 112 x 112 iken, çıkış boyutları 224 x 224 boyutlarındadır.

3.3. Yöntemlerin Karşılaştırılması

Geleneksel pankeskinleştirme yöntemleri CPU (Intel(R) Xeon(R) CPU @ 2.00GHz) ortamında gerçekleştirilmiştir. CPU ortamında gerçekleştirilen Landsat-8 uydu görüntülerine ait geleneksel pankeskinleştirme yöntemlerinin sonuçları Çizelge 3.1'de verilmiştir.

Derin öğrenme yöntemleri ise Tesla T4 GPU (Ortam I) ve KV260 FPGA DPU (Ortam II) ortamlarında gerçekleştirilmiştir. Landsat-8 uydu görüntülerine ait derin öğrenme yöntemlerinin sonuçları Çizelge 3.2'de verilmiştir. SRPPNN* modeli değişiklik yapılan SRPPNN modelini temsil eder.

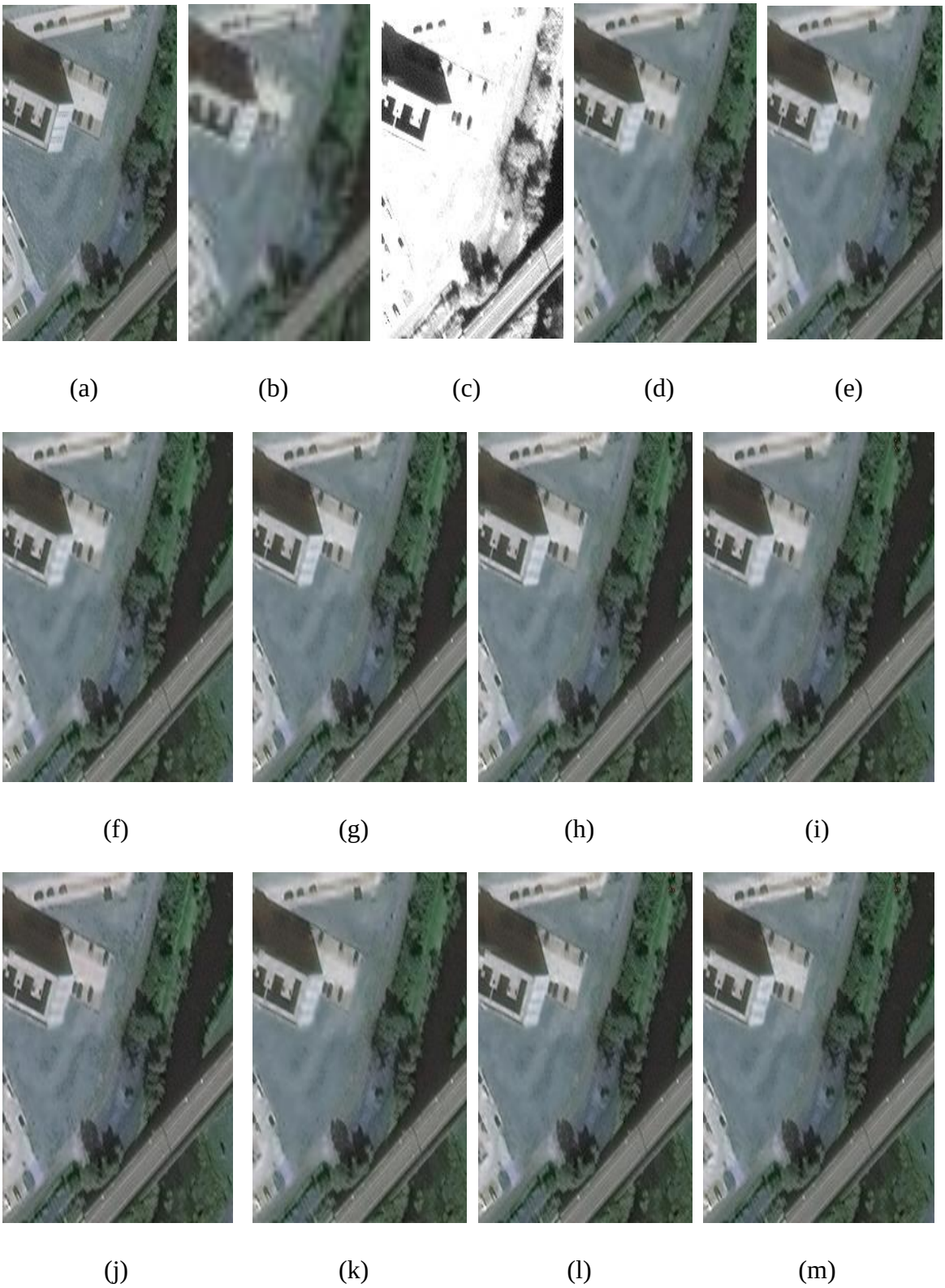


(a) (b) (c) (d) (e)

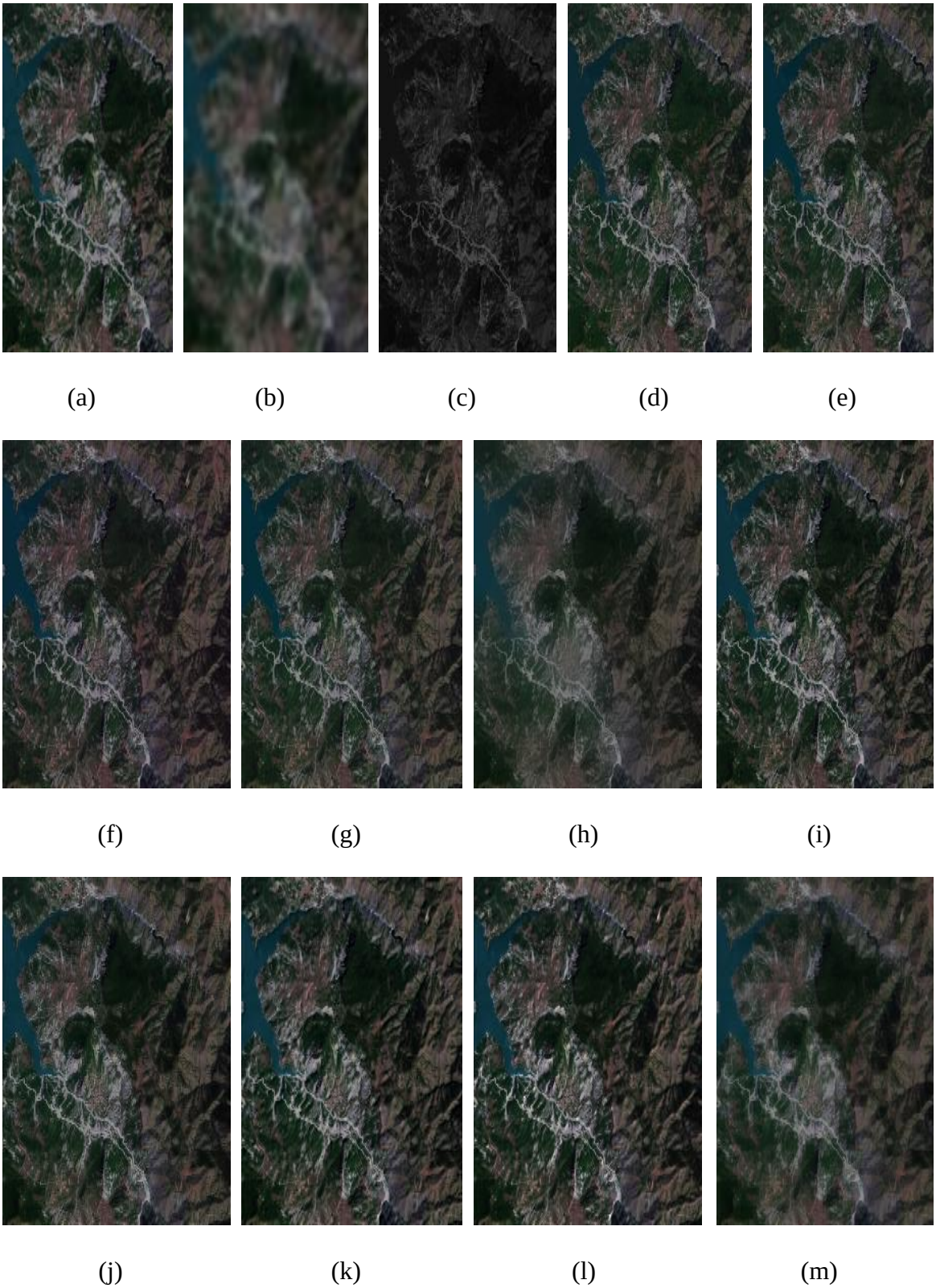
(f) (g) (h) (i)

(j) (k) (l) (m)

Resim 3.1. WorldView-3 veri setinde geleneksel pankeskinleştirme yöntemlerinin sonuçları: (a) Referans multispektral görüntü, (b) Düşük çözünürlüklü x4 yakınlaştırılmış multispektral görüntü, (c) Pankromatik görüntü, (d) Brovey, (e) IHS, (f) PCA, (g) GS, (h) GFPCA, (i) GSA, (j) SFIM, (k) MTF-GLP, (l) MTF-GLP-HPM, (m) DWT



Resim 3.2. WorldView-3 veri setinde derin öğrenme yöntemlerinin sonuçları: (a) Referans multispektral görüntü, (b) Düşük çözünürlüklü x4 yakınlaştırılmış multispektral görüntü, (c) Pankromatik görüntü, (d) PNN, (e) MSDCNN, (f) PanNet, (g) SRPPNN, (h) SRPPNN*, (i) PNN-FPGA, (j) MSDCNN-FPGA, (k) PanNet-FPGA, (l) SRPPNN-FPGA, (m) SRPPNN*-FPGA

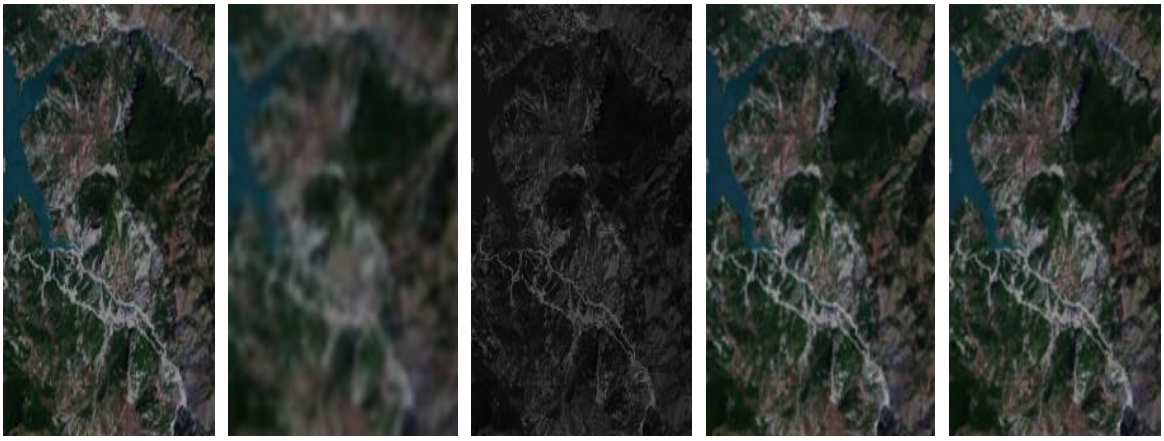


(a) (b) (c) (d) (e)

(f) (g) (h) (i)

(j) (k) (l) (m)

Resim 3.3. Landsat-8 veri setinde geleneksel pankeskinleştirme yöntemlerinin sonuçları: (a) Referans multispektral görüntü, (b) Düşük çözünürlüklü x2 yakınlaştırılmış multispektral görüntü, (c) Pankromatik görüntü, (d) Brovey, (e) IHS, (f) PCA, (g) GS, (h) GFPCA, (i) GSA, (j) SFIM, (k) MTF-GLP, (l) MTF-GLP-HPM, (m) DWT



(a) (b) (c) (d) (e)



(f) (g) (h) (i)



(j) (k) (l) (m)

Resim 3.4. Landsat-8 veri setinde derin öğrenme yöntemlerinin sonuçları: (a) Referans multispektral görüntü, (b) Düşük çözünürlüklü x4 yakınlaştırılmış multispektral görüntü, (c) Pankromatik görüntü, (d) PNN, (e) MSDCNN, (f) PanNet, (g) SRPPNN, (h) SRPPNN*, (i) PNN-FPGA, (j) MSDCNN-FPGA, (k) PanNet-FPGA, (l) SRPPNN-FPGA, (m) SRPPNN*-FPGA

Çizelge 3.1. Landsat-8 veri setinde geleneksel pankeskinleştirme yöntemlerinin sonuçları

Yöntemler	PSNR	RMSE	SSIM	ERGAS	CC	Q	BRISQUE
Referans	~	0	1	0	1	1	12
Brovey	28,169	10,101	0,874	14,194	0,956	0,977	20,741
GS	28,243	10,015	0,876	14,077	0,957	0,971	20,352
GSA	28,444	9,792	0,875	13,812	0,958	0,979	21,379
IHS	28,235	10,024	0,874	14,087	0,957	0,971	20,167
MTF GLP HPM	28,063	10,230	0,881	14,513	0,960	0,981	18,201
PCA	20,173	37,139	0,239	43,293	0,082	0,830	20,692
SFIM	28,046	10,252	0,863	14,540	0,954	0,978	18,394
Wavelet	27,615	10,754	0,840	14,932	0,951	0,971	18,962
GFPCA	27,130	11,398	0,814	15,913	0,951	0,963	13,662
MTF GLP	28,564	9,634	0,883	13,429	0,964	0,974	19,413

Çizelge 3.2. Landsat-8 veri setinde derin öğrenme yöntemlerinin sonuçları

Yöntemler	PSNR		RMSE		SSIM		ERGAS		CC		Q		BRISQUE	
Ortam	I	II	I	II	I	II	I	II	I	II	I	II	I	II
PNN	32,52	31,483	6,118	6,933	0,934	0,929	8,765	9,853	0,983	0,979	0,989	0,986	16,471	16,720
MSDCNN	32,77	32,199	5,933	6,359	0,938	0,930	8,466	9,060	0,984	0,982	0,990	0,988	13,792	16,001
PanNet	31,20	30,756	7,284	7,648	0,900	0,896	10,735	11,209	0,975	0,972	0,984	0,983	21,071	20,842
SRPPNN	33,29	32,219	5,582	6,333	0,942	0,930	8,013	8,889	0,985	0,982	0,990	0,988	11,271	7,802
SRPPNN*	33,52	32,331	5,431	6,265	0,946	0,938	7,703	8,931	0,987	0,984	0,991	0,987	10,842	11,201

CPU ortamında gerçekleştirilen WorldView-3 uydu görüntüleri ile elde edilmiş olan geleneksel pankeskinleştirme yöntemlerinin sonuçları Çizelge 3.3'te verilmiştir. Derin öğrenme tabanlı pankeskinleştirme yöntemlerinin sonuçları ise Çizelge 3.4'te verilmiştir. SRPPNN* ile gösterilen model tez çalışmasında önerilen RDN uyarlaması için SRPPNN modelini temsil eder.

Çizelge 3.3. WorldView-3 veri setinde geleneksel pankeskinleştirme yöntemlerinin sonuçları

Yöntem	PSNR	RMSE	SSIM	ERGAS	CC	Q	BRISQUE
Referans	~	0	1	0	1	1	37
Brovey	22,251	20,097	0,850	22,543	0,826	0,951	29,301
GS	22,273	20,045	0,851	22,482	0,826	0,950	29,292
GSA	22,292	20,004	0,851	22,435	0,827	0,950	29,391
IHS	25,174	14,340	0,846	15,949	0,935	0,973	45,265
MTF GLP HPM	22,00	20,728	0,840	23,339	0,804	0,945	29,301
PCA	22,543	19,217	0,723	21,494	0,899	0,946	41,652
SFIM	25,847	13,256	0,802	14,736	0,938	0,981	20,471
Wavelet	25,816	13,313	0,852	14,773	0,942	0,962	45,379
GFPCA	26,045	13,061	0,881	14,474	0,933	0,975	34,244
MTF GLP	27,899	10,457	0,879	11,632	0,957	0,986	33,292

Çizelge 3.4. WorldView-3 veri setinde derin öğrenme yöntemlerinin sonuçları

Yöntem	PSNR		RMSE		SSIM		ERGAS		CC		Q		BRISQUE	
Ortam	I	II	I	II	I	II	I	II	I	II	I	II	I	II
PNN	31,766	30,078	6,747	8,214	0,940	0,932	7,469	9,321	0,981	0,972	0,986	0,980	34,626	35,169
MSDCNN	32,204	30,472	6,411	7,947	0,944	0,935	7,100	9,132	0,983	0,979	0,988	0,983	33,251	36,903
PanNet	31,775	30,609	6,728	7,797	0,938	0,924	7,456	8,643	0,982	0,975	0,987	0,984	34,948	33,203
SRPPNN	33,534	31,007	5,494	7,476	0,954	0,940	6,082	8,584	0,988	0,972	0,992	0,985	32,496	32,766
SRPPNN*	33,544	31,051	5,482	7,321	0,954	0,940	6,070	8,157	0,988	0,974	0,992	0,985	32,204	32,269

Yöntemlerin bir görüntü için gereken test süreleri Çizelge 3.5'te verilmiştir. SRPPNN* değişiklik yapılan SRPPNN modelini temsil eder.

Çizelge 3.5. Yöntemlerin test süreleri

Veri Seti	Yöntem	Ortam	Test Süresi		
WV3/L8	Brovey	CPU	0,031		
	IHS		0,028		
	Wavelet		0,031		
	PCA		0,033		
	SFIM		0,039		
	GS		0,035		
	MTF_GLP		0,072		
	MTF_GLP_HPM		0,072		
	GSA		0,035		
	GFPCA		0,034		
	PNN	CPU	0,272		
		GPU	0,07		
		FPGA-DPU-I/II/III	0,037	0,034	0,027
	PanNet	CPU	0.280		
		GPU	0.078		
		FPGA-DPU-I/II/III	0,036	0,036	0,027
	MSDCNN	CPU	0.785		
		GPU	0.072		
		FPGA-DPU-I/II/III	0,066	0,06	0,05
WV3	SRPPNN	CPU	0.620		
		GPU	0.107		
		FPGA-DPU-I/II/III	0,063	0,062	0,042
	SRPPNN*	CPU	0.810		
		GPU	0.098		
		FPGA-DPU-I/II/III	0,084	0,081	0,053
L8	SRPPNN	CPU	0.519		
		GPU	0.103		
		FPGA-DPU-I/II/III	0,067	0,05	0,036
	SRPPNN*	CPU	0.693		
		GPU	0.117		
		FPGA-DPU-I/II/III	0,072	0,064	0,044

Çizelge 3.5'te yer alan Bölüm 2.6.2'de açıklanan Vivado tasarımında yer alan DPU mimarisi seçimi DPU-I için B2304, DPU-II için B3136, DPU-III için ise B4096'dır. DPU mimarisine göre oluşturulan donanımların kaynak kullanımı Bölüm 2.6.1 Çizelge 2.7'de verildiği gibidir. Donanımların güç tüketimleri ise DPU-I için 4,531 W, DPU-II için 4,906 W ve DPU-III için 5,601 W'dır. DPU mimarileri için test süreleri kısaldıkça paralel çalışma kabiliyeti için ihtiyaç duyduğu kaynak kullanımı artmaktadır. Kaynak kullanımının artması ile birlikte donanımların tükettiği güç miktarı da artmaktadır.

4. TARTIŞMA

Bu bölümde, çalışmada tez çalışması gerçekleştirilirken araştırmacılara faydası olması adına karşılaşılan problemler ile çözümleri, seçilen yöntemler ile sebepleri ve çalışmanın neticelendirilmesi noktasına kadar elde edilen deneyimler paylaşılacaktır.

Tez çalışması AMD Ryzen 7 5800H Radeon Graphics 3.20 GHz ve 16 GB RAM donanımlarına sahip dizüstü bilgisayarda gerçekleştirilmiştir. Derin öğrenme tabanlı pankeskinleştirme yöntemlerinin FPGA platformuna taşınabilmesi için Vitis AI ortamı kullanılmıştır. Vitis AI ortamı Linux Ubuntu dağıtımında çalışabilmekte, Windows ortamında destek sunmamaktadır. Bu nedenle Ubuntu 18.04.05 dağıtımı kurulmuştur. Fakat derin öğrenme tabanlı pankeskinleştirme modellerinin eğitimi Vitis AI ortamında gerçekleştirilememiştir. Bu aşama da iki problem ile karşılaşılmıştır. İlk problem model eğitimi için ihtiyaç duyulan RAM ihtiyacının bilgisayar ortamından karşılanamamasıdır. Eğitim aşamasında modelden modele farklılık gösterecek şekilde ek olarak yaklaşık 32 GB'lık RAM ihtiyacı doğmuştur. Fakat bilgisayar ortamında yalnızca 13 GB'lık kullanılabilir RAM alanı bulunmaktadır. Bu sorunu çözmek adına dizüstü bilgisayarda takılı olan SSD (Solid-State Drive)'de takas alanı (swap area) tanımlanarak, RAM ihtiyacı SSD üzerinden sağlanmıştır. Yazma ve okuma hızı RAM'e kıyasla daha yavaş olsa da RAM ihtiyacını bu yöntem çözmüştür. İkinci problem kurulu olan Ubuntu dağıtımında dizüstü bilgisayardaki işlemcinin GPU desteğini sunmamasıdır. Bu nedenle model eğitimi CPU ortamında gerçekleştirilmesi adına deneme yapılmıştır. Fakat CPU ortamının derin öğrenme modellerinin eğitiminde ihtiyaç duyduğu matematiksel hesaplamaları uzun bir süre de yapması ve bir model eğitiminin 100 iterasyon için 1 haftaya yakın sürmesi çalışmaları zor duruma sokmuştur. Bu nokta da model eğitimi aşamasında GPU ihtiyacını karşılamak adına Google'ın sunmuş olduğu Colab ortamı değerlendirilmiştir. Fakat Google Colab sınırlı kaynak sunduğu için eğitim veri setinde yer alan görüntülerin tamamı modele gönderilmemekte, görüntü kümeleri şeklinde modele gönderilmesine karar verilmiştir. Bu kapsamda eğitim ve doğrulama için 2000 adet görüntüden oluşan veri seti 100 iterasyon için 4 parçaya bölünerek 500'er şekilde modele gönderilmekte ve her görüntü kümesi için model ağırlıkları kaydedilerek tekrar eğitilmektedir. Diğer yandan model eğitimi için iki farklı optimizasyon algoritması denenmiştir. SGD optimizasyon algoritması ile model eğitimi başlatılmış fakat daha düşük başarım sonucuna daha uzun sürede ulaşıldığı için zaman

kaybetmemek adına sonuçlandırılmıştır. ADAM optimizasyon algoritması ise hem değişen öğrenme oranı hem de gradyan azalma tekniğinde momentuma sahip olduğu için başarılı eğitim sonuçları elde edilmiş ve deneysel sonuçlara ulaşılmıştır.

Uydu görüntülerinde pankeskinleştirme işleminin yapılabilmesi için aynı sahneye ait pankromatik bant ve multispektral bantların yer aldığı veri setine ihtiyaç duyulmaktadır. Açık kaynak erişilebilir veri setlerinde yalnızca multispektral görüntülere ait sahneler yer almaktadır. Bu nedenle veri seti toplanması aşamasında zorluk yaşanmıştır. Tez çalışmamızda iki farklı uydudan elde edilen görüntüleri kullanarak deney sonuçları elde edilmiştir. Landsat-8 uydusundan elde edilen görüntülerden oluşan veri seti, WorldView-3 uydusundan elde edilen görüntülere göre daha kaliteli idi. Buradan kaliteliden kasıt hem görsel olarak fark edilmekte hem de BRISQUE skoru ile sunulmuştur. Bu nedenle pankromatik bantta yer alan bilgilerden önemli derecede yararlanan geleneksel yöntemlerde WorldView-3 uydusu veri seti için Landsat-8'e göre daha kötü sonuçlar sunmuştur.

Derin öğrenme tabanlı pankeskinleştirme yöntemleri adına özgün bir çalışma sunmak için geleneksel yöntemlerden olan dalgacık dönüşümü yönteminin derin öğrenme alanında kullanılması denenmiştir. Yaklaşım şu şekilde idi, multispektral görüntünün bikübik enterpolasyon yöntemi ile yükselterek pankromatik bant ile eş seviyeye taşınması ve sonrasında dalgacık dönüşümünün yapılmasıdır. Dalgacık dönüşümü yapıldıktan sonra modele ait düşük frekans bileşenlerinin yer aldığı siyah-beyaz görüntü ile pankromatik bantta yer alan düşük frekanslı bileşenlerin yer aldığı görüntünün yer değiştirilmesidir. Ardından tekrar tersine dalgacık dönüşümü yaparak evrişim katmanları ile model eğitiminin gerçekleştirilmesidir. Fakat bu yaklaşım multispektral görüntüde yer alan düşük frekanslı bileşenlerin kaybolmasına neden olmaktadır. Bu da model performansını düşürmektedir. Bu nedenle multispektral görüntüde yer alan tüm frekans bileşenlerinin yer aldığı görüntünün kademeli olarak çözünürlüğünü arttırılması aşamasında görüntüye pankromatik bandın enjekte edilmesi ile daha iyi sonuçlar elde edildiği fark edilmiştir. Pankromatik bandın enjekte edilmediği durumda ise derin öğrenme modelinde kullanılan filtreler nedeniyle bulanık bir çıktı verme eğiliminde olduğu fark edilmiştir. Fakat pankromatik bantta yer alan yüksek frekanslı bileşenlerin modele enjekte edilmesinin model performansını iyileştirdiği gözlemlenmiştir.

Modelin FPGA'de tanımlı DPU mimarisinde koşması için görüntülerin 0 ile 1 arasında normalleştirilmesi gerekmektedir. Görüntülerin 0 ile 1 arasında normalleştirilmesi çoğu derin öğrenme modelinde iyi sonuç sunmakta iken RAM ihtiyacını arttırmaktadır. Sınırlı olan RAM ihtiyacını kullanabilmek adına veri setinde yer alan görüntülerin boyutları 1920 x 1080'den 224 x 224'e düşürülmüştür. Görüntü boyutlarının düşürülmesi hem RAM ihtiyacını azaltmış hem de eğitim süresini düşürerek deney çalışmalarını hızlandırmıştır.

Tez çalışmasında 224 x 224 çözünürlüğündeki WorldView-3 veri setindeki 250 görüntü için ortalama test süresi PNN modeli adına CPU ortamında 0,272 saniye, GPU ortamında 0,07 saniye FPGA DPU-III ortamında 0,027 saniye sürmüştür. Cai ve ark. [6] tarafından WorlView-3 veri setindeki 1024 x 1024 çözünürlüğündeki 200 görüntü için ortalama test süresi yalnızca GPU ortamında verilmiştir, bu sürede 0,0676 saniye olarak ölçülmüştür. MSDCNN modeli için tez çalışmasında CPU ortamında 0,785 saniye, GPU ortamında 0,072 saniye ve FPGA DPU-III ortamında 0,05 saniye sürmüş iken Cai ve ark [6] tarafından GPU ortamında 0,0837 saniye sürmüştür. PanNet modeli için çalışmamızda gerçekleşen test süreleri CPU ortamında 0,280 saniye, GPU ortamında 0,078 saniye ve FPGA DPU-III ortamında ise 0,027 saniye sürmüştür. Cai ve ark. [6] tarafından PanNet için GPU ortamında 0,0880 saniye sürmüştür. Son olarak SRPPNN modeli için tez çalışmasında CPU ortamı için 0,620 saniye, GPU ortamı için 0,098 saniye ve FPGA-DPU-III ortamı için 0,042 saniye ölçüm alınmıştır. Cai ve ark. [6] tarafından ölçülen süre GPU ortamı için 0,0892 saniye olmuştur. Nihai durumda yapılan karşılaştırmada en hızlı model daha az sayıda derin öğrenme katmanına sahip olmasından dolayı PNN modeli olmuştur, en yavaş model ise daha fazla derin öğrenme katmanına sahip olmasından dolayı SRPPNN modeli olmuştur. Nihai durumda tüm modeller için FPGA ortamı hem CPU hem de GPU ortamı için daha hızlı işlem yapabilme kabiliyetini kanıtlamıştır.

Masi ve ark. [3] tarafından önerilen PNN yönteminde WorldView-2 veri setinde 320 x 320 boyutlarına sahip 30 test görüntüsü için Q skoru 0,9442 olarak ölçülmüştür. Benzer boyutlarda ve adette olan test görüntüleri ile sırasıyla Ikonos ve GeoEye-1 veri setinde 0,9006 ve 0,9402 Q skorları ölçülmüştür. Yang ve ark. [4] tarafından önerilen PanNet yöndeminde WorldView-3 veri setinden elde edilen 200 adet test görüntüsü için 0,938 Q skoru ölçülmüştür. Araştırmacılar tarafından aynı veri seti için PNN yöntemi 0,972 Q skoru ölçülmüştür. Önerdikleri yöntem PNN yönteminin önüne geçmiştir. Yuan ve ark. [5] tarafından önerilen MSDCNN yönteminde Quickbird veri setindenden elde edilen 250 x 250

boyutlarında 160 test görüntüsü için 0,9303 Q skoru ölçülürken PNN yöntemi için 0,9206 Q skoru ölçülmüştür. Diğer yandan WorldView-2 veri seti için 250 x 250 boyutlarında 80 görüntü için önerilen yöntem için Q skoru 0,9570 olarak elde edilirken PNN yöntemi için 0,9389 elde edilmiştir. Nihai durumda önerdikleri yöntem PNN'nin önüne geçmiştir. Cai ve ark. [6] tarafından önerilen SRPPNN yönteminde Quickbird, WorldView-3 ve Landsat-8 uydularından elde edilen 256 x 256 boyutlarında 200 test görüntüsü kullanılmıştır. Araştırmacıların önerdiği yönteme ait Q skoru Quickbird veri seti için 0,8326, PNN için 0,7841, MSDCNN için 0,8068 ve PanNet için 0,7762 olarak ölçülmüştür. WorldView-3 veri seti için önerilen yöntem olan SRPPNN ait sonuçlar 0,8340, PNN için 0,6654, MSDCNN için 0,8105 olarak ölçülmüştür. Landsat-8 veri seti için SRPPNN yönteminde Q skoru 0,9774, PNN için 0,9529, MSDCNN için 9600 ve PanNet için 0,9133 olarak ölçülmüştür. Nihai durumda önerilen yöntem tüm test veri seti için en iyi skoru elde etmiştir. Gerçekleştirdiğimiz çalışmada önerdiğimiz yöntem RDN uyarlaması için SRPPNN Landsat-8 veri setinde 224 x 224 boyutlarında 250 adet test görüntüsü için 0,991 Q skor değerine ulaşılmıştır. Diğer yandan SRPPNN yönteminde benzer Q skoru 0,990, PNN için 0,989, MSDCNN için 0,990, PanNet için 0,983 skoru elde edilmiştir. WorlView-3 veri setinde ise önerdiğimiz yöntem ile SRPPNN aynı Q skoruna ulaşırken, SRPPNN için 33,534 olarak ölçülen PSNR değeri önerdiğimiz yöntemde 33,534 olarak elde edilmiştir. Aynı veri seti için PNN yönteminde 0,986 olarak ölçülen Q skoru, MSDCNN yöntemi 0,988, PanNET yöntemi 0,987 olarak ölçülmüştür.

Tez çalışması kapsamında hem geleneksel pankeskinleştirme yöntemleri hem de derin öğrenme tabanlı pankeskinleştirme yöntemleri adil bir karşılaştırma olması açısından aynı test veri setleri kullanılarak referanslı metrik sonuçları ve referanssız metrik sonucu elde edilmiştir. Test veri seti Landsat-8 uydusundan alınan görüntüler için giriş MS görüntüsü 112 x 112 pankromatik görüntünün çözünürlüğü olan 224 x 224 ile aynı çözünürlüğe taşınacak şekilde yöntemler uygulanmıştır. Diğer yandan WorldView-3 uydusuna ait görüntüler için giriş MS görüntüsü 56 x 56 pankromatik görüntünün çözünürlüğü olan 224 x 224 ile aynı çözünürlüğe taşınacak şekilde yöntemler uygulanmıştır. Landsat-8 uydusundan elde edilen veri setinde yapılan karşılaştırmada (Bknz. Çizelge 3.1) MTF GLP yöntemi diğer geleneksel pankeskinleştirme yöntemlerinden 28,56 PSNR ve 13,43 ERGAS metrik değerleri ile daha iyi sonuç elde etmiştir. Diğer yandan WorlView-3 uydusundan elde edilen veri setinde yapılan karşılaştırmada (Bknz. Çizelge 3.3) MTF GLP yöntemi 27,90 PSNR ve 11,63 ERGAS metrikleri ile daha iyi sonuç elde etmiştir. Derin öğrenme tabanlı

pankeskinleştirme yöntemleri için yapılan karşılaştırmada hem GPU ortamı için hem de FPGA ortamı için elde edilen sonuçlar metrikler ile sunulmuştur (Bknz. Çizelge 3.2 ve Çizelge 3.4). Landsat-8 uydusundan elde edilen veri seti için GPU ortamında SRPPNN* önermiş olduğumuz yöntem 33,52 PSNR ve 7,70 ERGAS metrik değerleri ile diğer yöntemlerden daha iyi bir sonuç elde etmiştir (Bknz. Çizelge 3.2). Aynı veri seti için FPGA ortamında 32,33 PSNR değeri ile SRPPNN* yöntemi daha iyi sonuç elde etmiştir. WorldView-3 uydusundan elde edilen veri seti için GPU ortamında 33,54 PSNR ve 6,07 ERGAS metrik değerleri ile diğer yöntemlerden daha iyi sonuç elde edilmiştir. Aynı veri seti için FPGA ortamında 31,05 PSNR ve 8,16 metrik değerleri ile diğer yöntemlerden daha iyi bir sonuç elde edilmiştir. Nihai durumda aynı yöntemler FPGA ortamında GPU ortamına göre model çıktıları için performans düşüşü meydana gelmiştir bunun sebebi GPU ortamında kullanılan ondalık değerlerin FPGA ortamında tam sayı değerler ile hesap yapılmasından dolayı oluşan kayıptır.

5. SONUÇ VE ÖNERİLER

Uzaktan nesne algılama kapsamında görüntüye ait çözünürlüğün arttırılmasının bir ihtiyaç haline gelmesi nedeniyle görüntü çözünürlüğünü hem spektral hem de uzaysal alanda arttırmaya yönelik bir çalışma gerçekleştirilmiştir. Çoğu uydu görüntüleri düşük çözünürlüklü multispektral görüntü ve daha yüksek uzaysal çözünürlüğe sahip pankromatik görüntüden oluşmaktadır. Görüntü çözünürlüğünü arttırırken yalnızca multispektral görüntüde yer alan bilgilerden değil aynı zamanda pankromatik görüntüde yer alan bilgilerden de yararlanarak tasarlanmış birçok model incelenmiştir. Deneysel sonuçlarda elde edilen verilere göre süper çözünürlük tabanlı kademeli pankeskinleştirme yöntemi en iyi performansı göstermiştir. Model performansını daha da arttırmak adına bu yönteme süper çözünürlük problemi için tasarlanan artık yoğun ağ mimarisi entegre edilmiştir. Artık yoğun ağ modelinde her katman, önceki tüm katmanlardan ek girdiler alır ve kendi özellik haritalarını sonraki tüm katmanlara iletmesi ile özellik haritasını zenginleştirerek model performansını arttırmıştır. Model performansı PSNR skoru ile Landsat-8 veri setinde 33,291'den 33,521'e artış gösterirken, WorldView-3 veri setinde 33,534'den 33,544'e artış göstermiştir. Çalışma kapsamında incelenen bir diğer konu modellerin hesaplama süresi maliyetidir. Bu süre tasarlanan modelin karmaşıklık seviyesine ve hesaplamayı yapan ortama göre değişkenlik göstermektedir. Bu kapsamda CPU ve GPU ortamına göre daha kısa sürede hesaplama yapabilen FPGA ortamı çalışmaya dahil edilmiştir. FPGA ortamı paralel işlem yapabilme kabiliyeti sayesinde modellerin hesaplama süresi maliyetlerini iyileştirmiştir. Bu çalışma da, derin öğrenme tabanlı modellerde hızlı işleme yetenekleri için uyarlanmış olan FPGA platformundan faydalanmak için Xilinx'in ürünü olan Kria KV260 geliştirme kartı kullanılmıştır. Çalışma kapsamında PNN, PanNet, MSDCNN, SRPPNN ve RDN uyarlaması için SRPPNN derin öğrenme tabanlı pankeskinleştirme modelleri Vitis AI aracı kullanılarak FPGA platformuna taşınmıştır. Sonuçlar, hesaplama süresi maliyeti açısından, FPGA'nın CPU ve GPU ortamlarına kıyasla üstün işleme yeteneğini doğrulamaktadır. Önerilen model ile FPGA ortamında Landsat-8 veri setinde 33,331 PSNR skoruna, WorlView-3 veri setinde 31,051 PSNR skoruna ulaşılmıştır. Bu derin öğrenme modeli ile CPU ortamında 0,693 saniyede ve GPU ortamında 0,117 saniyede çıktı elde edilirken, FPGA ortamında 5,601 W güç tüketimine sahip donanım ile 0,053 saniye de model çıktısı elde edilmiştir.

Çalışmada önerilen derin öğrenme modelinin performansını daha da arttırmak adına literatürde yer alan süper çözünürlük modellerindeki katmanlar araştırılarak modele kazandırılabilir. Karşılaştırma yapabilmek adına FPGA ortamına diğer derin öğrenme tabanlı pankeskinleştirme yöntemleri benzer metrikler ve hesaplama süresi maliyetleri açısından sunulabilir. Gerçekleştirilen çalışmada Kria KV260 kartı kullanılmıştır, KV260 kartı üzerinde yer alan FPGA'in sınırlı kaynağı olması nedeniyle tek çekirdekli DPU IP'si B4096 mimarisi için sentezlenebilmiştir. Daha çok kaynağa sahip olan FPGA kullanılarak B4096 mimarisine sahip olan DPU IP'sinin tek çekirdekli yapısına ek olarak iki, üç ve dört çekirdekli donanımların hesaplama süresi maliyetleri sunulabilir.

KAYNAKLAR

1. Kaur, G., Saini, K. S., Singh, D., and Kaur, M. (2021). A comprehensive study on computational pansharpening techniques for remote sensing images. *Archives of Computational Methods in Engineering*, 28(6) 1-18.

2. Wang, P., Yao, H., Li, C., Zhang, G., and Leung, H. (2021). Multiresolution analysis based on dual-scale regression for pansharpening. *IEEE Transactions on Geoscience and Remote Sensing*, 60, 1-19.

3. Masi, G., Cozzolino, D., Verdoliva, L., and Scarpa, G. (2016). Pansharpening by convolutional neural networks. *Remote Sensing*, 8(7), 594.

4. Yang, J., Fu, X., Hu, Y., Huang, Y., Ding, X., and Paisley, J. (2017, October). *PanNet: A deep network architecture for pan-sharpening.* In Proceedings of the IEEE international conference on computer vision (ICCV), Venice, Italy, 5449-5457.

5. Yuan, Q., Wei, Y., Meng, X., Shen, H., and Zhang, L. (2018). A multiscale and multidepth convolutional neural network for remote sensing imagery pan-sharpening. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 11(3), 978-989.

6. Cai, J., and Huang, B. (2020). Super-resolution-guided progressive pansharpening based on a deep convolutional neural network. *IEEE Transactions on Geoscience and Remote Sensing*, 59(6), 5206-5220.

7. Dong, C., Loy, C.C., He, K., and Tang, X. (2016). Image Super-Resolution Using Deep Convolutional Networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 38(2), 295-307.

8. Mamatha, G., Sumalatha, V., and Lakshmaiah, M. V. (2015, July). *FPGA implementation of satellite image fusion using wavelet substitution method.* In 2015 Science and Information Conference (SAI), London, United Kingdom, 1155-1159.

9. Viel, F., Parreira, W. D., Susin, A. A., and Zeferino, C. A. (2020). A hardware accelerator for onboard spatial resolution enhancement of hyperspectral images. *IEEE Geoscience and Remote Sensing Letters*, 18(10), 1796-1800.

10. Akkin, M., and Yalcin, M.E. (2023). *Implementation and comparison of super resolution algorithms on embedded systems*. Yüksek Lisans Tezi. İstanbul Teknik Üniversitesi, Fen Bilimleri Enstitüsü, İstanbul, Türkiye.

11. Zhang, Y., Tian, Y., Kong, Y., Zhong, B., and Fu, Y. (2018, June). *Residual dense network for image super-resolution*. In Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR), Salt Lake City, UT, USA.

12. İnternet: USGS. Web: https://earthexplorer.usgs.gov, Son Erişim Tarihi: 20.04.2023.

13. İnternet: Kaggle. Web: https://www.kaggle.com , Son Erişim Tarihi: 20.06.2023.

14. Wald, L., Ranchin, T., and Mangolini, M. (1997). Fusion of satellite images of different spatial resolutions: Assessing the quality of resulting images. *Photogrammetric engineering and remote sensing*, 63(6), 691-699.

15. Maurer, T. (2013). How to pan-sharpen images using the gram-schmidt pan-sharpen method–A recipe. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 40, 239-244.

16. S. M. Ross. (2004). *Introduction to Probability and Statistics for Engineers and Scientists.* Burlington, MA: Elsevier Academic.

17. Aiazzi, B., Baronti, S., and Selva, M. (2007). Improving component substitution pansharpening through multivariate regression of MS + Pan data. *IEEE Transactions on Geoscience and Remote Sensing*, 45(10), 3230-3239.

18. Liu, J. G. (2000). Smoothing filter-based intensity modulation: A spectral preserve image fusion technique for improving spatial details. *International Journal of Remote Sensing*, 21(18), 3461-3472.

19. Aiazzi, B., Alparone, L., Baronti, S., Garzelli, A., and Selva, M. (2006). MTF-tailored multiscale fusion of high-resolution MS and Pan imagery. *Photogrammetric Engineering & Remote Sensing*, 72(5), 591-596.

20. Vivone, G., Restaino, R., Dalla Mura, M., Licciardi, G., and Chanussot, J. (2013). Contrast and error-based fusion schemes for multispectral image pansharpening. *IEEE Geoscience and Remote Sensing Letters*, 11(5), 930-934.

21. Zafari, A., Kianmehr, M. H., and Abdolahzadeh, R. (2013). Modeling the effect of extrusion parameters on density of biomass pellet using artificial neural network. *International Journal of recycling of Organic waste in Agriculture*, 2, 1-11.

22. Güntürkün, R. (2003*). İleri beslemeli ve elman geri beslemeli yapay sinir ağları kullanarak harmoniklerin kompanzasyonu*, Doktora Tezi, Sakarya Üniversitesi, Fen Bilimleri Enstitüsü, Sakarya, Türkiye.

23. Qian, H. and Ding, J. Z. B. (2018, June). *Activation Function and Their Characteristics in Deep Neural Networks*, Chinese Control and Decision Conference (CCDC), Shenyang, China.

24. Dong, C., Loy, C.C., and Tang, X. (2016, October). *Accelerating the Super-Resolution Convolutional Neural Network*. European Conference on Computer Vision (ECCV), Amsterdam, Netherlands.

25. Lim, B., Son, S., Kim, H., Nah, S., and Mu Lee, K. (2017, July). *Enhanced deep residual networks for single image super-resolution*. In Proceedings of the IEEE conference on computer vision and pattern recognition workshops (CVPRW), Honolulu, Hawaii.

26. İnternet: Vitis AI Library User Guide. Web: https://docs.xilinx.com/r/en-US/ug1354-xilinx-ai-sdk , Son Erişim Tarihi: 20.09.2023.

27. İnternet: Zynq UltraScale+ MPSoC: DPUCZDX8G. Web: https://docs.xilinx.com/r/en-US/ug1414-vitis-ai/Zynq-UltraScale-MPSoC-DPUCZDX8G , Son Erişim Tarihi: 20.09.2023.

28. İnternet: Kria SOMs & Starter Kits Confluence. Web: https://xilinx wiki.atlassian.net/wiki/spaces/A/pages/1641152513/Kria+SOMs+Starter+Kits , Son Erişim Tarihi: 20.09.2023.

29. Aldin, N. B. and Aldin, S. S. A. B. (2022, March). *Accuracy Comparison of Different Batch Size for a Supervised Machine Learning Task with Image Classification*. 2022 9th International Conference on Electrical and Electronics Engineering (ICEEE), 136-144, Alanya, Türkiye.

30. Bock, S., and Weiß, M. (2019, July). *A proof of local convergence for the Adam optimizer*. In 2019 international joint conference on neural networks (IJCNN), Budapest, Hungary.

31. Tsukamoto, N., Sugaya, Y., and Omachi, S. (2020). Spectrum correction using modeled panchromatic image for pansharpening. *Journal of Imaging*, 6(4), 20.

32. Wang, Z., and Bovik, A. C. (2002). A universal image quality index. *IEEE signal processing letters*, 9(3), 81-84.

33. Mittal, A., Moorthy, A. K., and Bovik, A. C. (2012). No-reference image quality assessment in the spatial domain. *IEEE Transactions on image processing*, 21(12), 4695-4708.



Gazili olmak ayrıcalıktır