

ANKARA ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

YÜKSEK LİSANS TEZİ

GRIÖLÇEK GÖRÜNTÜLERDE DERİN ÖĞRENME TABANLI SÜPER
ÇÖZÜNÜRLÜK ALGORİTMALARININ KARŞILAŞTIRILMASI

Muhammet DABAK

ELEKTRİK ELEKTRONİK MÜHENDİSLİĞİ ANABİLİM DALI

ANKARA
2024

Her hakkı saklıdır

ÖZET

Yüksek Lisans Tezi

GRIÖLÇEK GÖRÜNTÜLERDE DERİN ÖĞRENME TABANLI SÜPER ÇÖZÜNÜRLÜK ALGORİTMALARININ KARŞILAŞTIRILMASI

Muhammet DABAK

Ankara Üniversitesi
Fen Bilimleri Enstitüsü
Elektrik -Elektronik Mühendisliği Anabilim Dalı

Danışman: Dr. Öğr. Üyesi Ahmet AKBULUT

Görüntü süper çözünürlüğü, düşük çözünürlüklü görüntüden yüksek çözünürlüklü görüntüye dönüşüm ve iyileştirme işlemidir. Görüntü süper çözünürlüğü gözetleme, medya ve medikal alanlarda çeşitli amaçlar doğrultusunda kullanılabilir. Askeri alanlarda kullanılan Elektronik Harp (EH) tekniklerinde veri olarak görünür bandın dışında kalan elektromanyetik spektrumun farklı bölgelerindeki griölçek görüntüler kullanılmaktadır. Griölçek görüntüler içerisinde farklı ışık/hava koşullarında ve yüksek menzilde görüntüleme sağlayabilen Orta-Dalga Kızılötesi (MWIR) görüntüleri bulunmaktadır. MWIR bölgesinde çalışan kamera ve ilgili optik bileşenleri soğutmalı detektörlerle kullanıldığından ve bu sebeple maliyetli olduğundan, yüksek çözünürlüklü kameralar tercih edilmemektedir. Düşük çözünürlüklü kamera kullanarak yüksek çözünürlüklü görüntü elde etme ihtiyacı süper çözünürlük teknikleri ile karşılanabilir. Klasik bir yöntem olan biküçük interpolasyon ve 8 derin öğrenme tabanlı algoritma, farklı kayıtlardan tekil olarak alınan 640x512 çözünürlüğünde 164 eğitim ve 41 test görüntüsü üzerinde x4 aşağı örnekleme ölçeğiyle biküçük interpolasyon yapıldıktan sonra x4 yukarı örnekleme ölçeğiyle çalıştırılmıştır. Kullanılan veri kümesinde eğitimi 24 dakika süren ve saniyede ortalama 8.81 görüntü işleyebilen DCSCN algoritmasının mimarisi ve hiper-parametreleri değiştirilerek MSE, PSNR ve SSIM performans metrikleri açısından en başarılı sonuç elde edilmiştir. Literatürde, mevcut olan derin öğrenme tabanlı süper çözünürlük tekniklerinin MWIR görüntüler üzerinde uygulanması ve karşılaştırılmasına dair çok sınırlı sayıda çalışma bulunmaktadır. Tez çalışmasında elde edilen bulgular ve çıkarımlar, derin öğrenme tabanlı süper çözünürlük algoritmalarının klasik yöntemler yerine MWIR görüntülerin yukarı örneklemeinde kullanılabilmesini göstermiştir.

Temmuz 2024, 64 sayfa

Anahtar Kelimeler: DCSCN, elektronik harp, griölçek, hiper-parametre, MSE, Orta-Dalga Kızılötesi, PSNR, SSIM, süper çözünürlük, yukarı örnekleme

ABSTRACT

Master Thesis

COMPARISON OF DEEP LEARNING BASED SUPER RESOLUTION ALGORITHMS ON GRAYSCALE IMAGES

Muhammet DABAK

Ankara University
Graduate School of Natural and Applied Sciences
Department of Electrical and Electronics Engineering

Supervisor: Asst. Prof. Dr. Ahmet AKBULUT

Image super-resolution is the process of transforming and enhancing a low-resolution image into a high-resolution image. Super-resolution imaging can be used for various purposes in surveillance, media, and medical fields. In electronic warfare (EW) techniques used in military fields, grayscale images from different regions of the electromagnetic spectrum outside the visible band are used as data. Among these grayscale images, there are Mid-Wave Infrared (MWIR) images that can provide imaging under different light/weather conditions and at high ranges. Since camera and related optic parts works in MWIR band are used with cooled detectors and therefore they are costly, high-resolution cameras are not preferred. The need to obtain high-resolution images using low-resolution cameras can be addressed through super-resolution techniques. A classical method, bicubic interpolation, and 8 deep learning-based algorithms were run on 164 training and 41 test images with a resolution of 640x512, taken individually from different records, with a x4 downsampling scale using bicubic interpolation and then a x4 upsampling scale. By modifying the architecture and hyper-parameters of the DCSCN algorithm, which took 24 minutes to train and could process an average of 8.81 images per second, the most successful results were achieved in terms of MSE, PSNR, and SSIM performance metrics. In the literature, there are very few studies on the application and comparison of existing deep learning-based super-resolution techniques on MWIR images. The findings and conclusions of the thesis have shown that deep learning-based super-resolution algorithms could be used to upsample MWIR images instead of classic methods.

July 2024, 64 pages

Key Words: DCSCN, electronic warfare, grayscale, hyper-parameter, MSE, mid-wave infrared, PSNR, SSIM, super resolution, upsampling

TEŐEKKÜR

Çalıőmalarımı yönlendiren, bilgi ve önerileriyle katkıda bulunan danışmanım Dr. Öğr. Üyesi Ahmet AKBULUT'a (Ankara Üniversitesi Elektrik ve Elektronik Mühendisliđi Anabilim Dalı Öğretim Üyesi), TÜBİTAK BİLGEM İLTAREN'deki yardımlarını esirgemeyen tüm meslektaşlarıma ve eşim Sevda DABAK'a teşekkür ederim.

Muhammet DABAK
Ankara, Temmuz 2024

İÇİNDEKİLER

ÖZET.....	i
ABSTRACT.....	ii
TEŞEKKÜR.....	iii
KISALTMALAR DİZİNİ.....	vi
ŞEKİLLER DİZİNİ.....	vii
ÇİZELGELER DİZİNİ.....	viii
1. GİRİŞ	1
2. KURAMSAL TEMELLER.....	5
2.1 Yapay Sinir Ağları	5
2.2 Sinir (Neuron) Yapısı.....	5
2.3 Derin Sinir Ağ Yapısı.....	6
2.4 Evrişimsel (Convolutional) Sinir Ağları	7
2.5 Aktivasyon Fonksiyonu	8
2.5.1 Eşik (threshold) aktivasyon fonksiyonu	8
2.5.2 Sigmoid aktivasyon fonksiyonu	9
2.5.3 Hiperbolik tanjant aktivasyon fonksiyonu	10
2.5.4 RELU aktivasyon fonksiyonu	10
2.5.5 Leaky RELU aktivasyon fonksiyonu.....	11
2.5.6 Softmax aktivasyon fonksiyonu	12
2.6 Geri Yayılım (Back Propagation)	13
2.7 Öğrenme Oranı.....	13
2.8 Öğrenme Yöntemleri	14
2.8.1 Denetimli öğrenme (supervised learning)	14
2.8.2 Denetimsiz öğrenme (unsupervised learning)	15
2.8.3 Yarı-denetimli öğrenme (semi-supervised learning).....	15
2.8.4 Kendi kendine denetimli öğrenme (self-supervised learning).....	16
2.8.5 Zayıf denetimli öğrenme (weakly-supervised learning)	16
2.9 Epok (Epoch)	16
2.10 Kayıp Fonksiyonu (Loss Function)	17
2.10.1 Ortalama kare hata (mean squared error (MSE))	17
2.10.2 Ortalama mutlak hata (mean absolute error (MAE))	18
2.10.3 Logaritmik kayıp (logarithmic loss / log loss)	18
2.11 Aşırı Öğrenme (Overfitting).....	19
2.12 Eksik Öğrenme (Underfitting).....	21
2.13 Yiğın Boyutu (Batch Size)	22
3. MATERYAL VE YÖNTEM.....	27
3.1 Süper Çözünürlük Algoritmaları	27
3.1.1 Bikübik (bicubic) interpolasyon	27
3.1.2 SRCNN (super resolution convolutional neural network)	28
3.1.3 VDSR (very deep convolutional networks).....	30
3.1.4 ESPCN (efficient sub-pixel convolutional neural network)	31
3.1.5 EDSR (enhanced deep residual networks).....	32
3.1.6 DCSCN (deep convolutional neural network with skip connection and network in network).....	34

3.1.7 LapSRN (deep laplacian pyramid networks)	35
3.1.8 RCAN (residual channel attention networks)	36
3.1.9 IDN (information distillation network).....	38
3.2 Veri Kümesi	40
3.3 Bilgisayar Konfigürasyonu.....	41
3.4 Performans Metrikleri.....	41
3.4.1 MSE (mean square error)	41
3.4.2 PSNR (peak signal-to-noise ratio).....	42
3.4.3 SSIM (structural similarity index measure)	43
3.5 Uygulama Detayları	43
4. ARAŞTIRMA BULGULARI.....	45
5. TARTIŞMA VE SONUÇ.....	60
KAYNAKLAR.....	63

KISALTMALAR DİZİNİ

CNN	Convolutional Neural Network
CUDA	Compute Unified Device Architecture
DCSCN	Deep CNN with Skip Connection and Network in Network
EDSR	Enhanced Deep Residual Networks
EH	Elektronik Harp
ESPCN	Efficient Sub-Pixel Convolutional Neural Network
GAN	Generative Adversarial Network
GB	Gigabyte
GPU	Graphical Processing Unit
HR	High-Resolution
IDN	Information Distillation Network
LapSRN	Laplacian Pyramid Super Resolution Network
LR	Low-Resolution
LWIR	Longwave Infrared
MSE	Mean Square Error
MWIR	Mid-wave Infrared
PSNR	Peak Signal-to-Noise Ratio
RCAN	Residual Channel Attention Networks
ReLU	Rectified Linear Unit
SR	Super Resolution
SRCNN	Super Resolution Convolutional Neural Network
SSIM	Structural Similarity Index Measure
SWIR	Shortwave Infrared
VDSR	Very Deep Super Resolution

ŞEKİLLER DİZİNİ

Şekil 2.1 Sinir (neuron) modeli.....	5
Şekil 2.2 Derin sinir ağ yapısı.....	7
Şekil 2.3 Evrişim işlemi.....	8
Şekil 3.1 SRCNN ağ yapısı.....	30
Şekil 3.2 VDSR ağ yapısı.....	31
Şekil 3.3 Önerilen kalan değerli ağ bloğu.....	33
Şekil 3.4 Önerilen tek ölçekli süper çözünürlük mimarisi.....	33
Şekil 3.5 Önerilen DCSCN mimarisi.....	34
Şekil 3.6 Önerilen LapSRN mimarisi.....	36
Şekil 3.7 Önerilen RCAN mimarisi.....	37
Şekil 3.8 Önerilen IDN mimarisi.....	38
Şekil 3.9 IDN iyileştirme bloğu.....	39
Şekil 3.10 Örnek veri kümesi görüntüsü (askeri araç).....	40
Şekil 4.1 Kalibrasyon nesnesi için 3 algoritmanın karşılaştırılması.....	52
Şekil 4.2 Askeri araç görüntüsü için 3 algoritmanın karşılaştırılması.....	53
Şekil 4.3 Sivil araç görüntüsü için 3 algoritmanın karşılaştırılması.....	54
Şekil 4.4 Arazi görüntüsü için 3 algoritmanın karşılaştırılması.....	55
Şekil 4.5 İnsan görüntüsü için 3 algoritmanın karşılaştırılması.....	56
Şekil 4.6 Orijinal askeri araç görüntüsü (640x512).....	58
Şekil 4.7 Bikübik interpolasyon ile yukarı örneklenmiş görüntü (1280x1024).....	58
Şekil 4.8 DCSCN algoritması ile yukarı örneklenmiş görüntü (1280x1024).....	59

ÇİZELGELER DİZİNİ

Çizelge 4.1 SRCNN deneme çizelgesi.....	45
Çizelge 4.2 VDSR deneme çizelgesi	45
Çizelge 4.3 ESPCN deneme çizelgesi.....	46
Çizelge 4.4 LapSRN deneme çizelgesi	47
Çizelge 4.5 DCSCN deneme çizelgesi.....	47
Çizelge 4.6 EDSR deneme çizelgesi.....	48
Çizelge 4.7 RCAN deneme çizelgesi	48
Çizelge 4.8 IDN deneme çizelgesi	49
Çizelge 4.9 Algoritmaların performans metriklerine göre karşılaştırılması	49
Çizelge 4.10 Algoritmaların eğitim süreleri	50
Çizelge 4.11 Algoritmaların test süreleri	51
Çizelge 4.12 Parametrelere göre DCSCN performans çizelgesi.....	57

1. GİRİŞ

Sezgisel olarak uzamsal çözünürlük, bir görüntüdeki ayırt edilebilir en küçük detayı ölçen ölçüttür. Görüntü çözünürlüğü ise yaygın olarak birim mesafe başına ayırt edilebilen en fazla çizgi çifti (siyah/beyaz) sayısı olarak tanımlanmaktadır (Gonzalez 2018). Görüntü süper çözünürlüğü, düşük çözünürlüklü görüntüden yüksek çözünürlüklü görüntüye dönüşüm ve iyileştirme işlemidir. Yapay sinir ağlarının ve öğrenme tabanlı yapıların kullanılmaya başlanması ile beraber görüntü süper çözünürlük teknikleri, klasik yöntemler aksine öğrenme tabanlı yapılar ile geliştirilmektedir. Derin öğrenme tabanlı süper çözünürlük tekniği ilk defa SRCNN (Super Resolution Convolutional Neural Network) algoritması ile Dong vd.(2014) tarafından geliştirilmiş ve diğer çalışmalar için öncü olmuş, 6 binden fazla atıf almıştır. Dong vd., ImageNet üzerinden alınan farklı veri kümeleri üzerinde, geleneksel yöntemler içerisinde bulunan biküçük interpolasyon ve seyrek kodlama tabanlı yöntemler ile SRCNN algoritmasını karşılaştırmıştır. İçerisinde Tepe Sinyal Gürültü Oranı (PSNR (Peak Signal-to-Noise Ratio)) ve Yapısal Benzerlik Göstergesi Ölçütü (SSIM (Structural Similarity Index Measure)) yer alan 6 performans metriği ile kıyaslama yapmış, SRCNN algoritmasının başarısını ortaya koymuştur. 2014 yılından günümüze dek sunulan süper çözünürlük mimarileri farklılaşmakta ve daha başarılı sonuçlar vermektedir. Derin öğrenme tabanlı süper çözünürlük algoritmaları tekrarlamalı (recursive), çoklu-dal (multi-branch), kalan değerli (residual), yoğun bağlı (densely connected), doğrusal (linear), aşamalı yeniden inşa (progressive reconstruction), ilgi-tabanlı (attention-based), çoklu alçalma işleme (multiple degradation handling) ve üretken çekişmeli ağ tabanlı (GAN-based) modeller olarak sınıflandırılmaktadır. Lepcha vd. (2023) derin öğrenme tabanlı süper çözünürlük algoritmalarının sınıflandırılması, uygulama alanları, performans metrikleri ve kullanılan veri kümelerine ilişkin çok detaylı bir çalışma sunmuştur. Süper çözünürlük literatürde farklı yukarı örnekleme ölçekleri ile çalışılmaktadır (x2, x3, x4, x8, x16). Yukarı örnekleme ölçeği, düşük çözünürlüklü görüntünün yüksek çözünürlüklü görüntüye dönüştürülürken dikey ve yatay yönde piksel sayısının kaç katına çıkacağını gösteren bir değerdir. Derin öğrenme tabanlı süper çözünürlük, literatürde halen aktif bir konu olarak karşımıza çıkmaktadır.

Görüntü süper çözünürlüğü, uydu görüntüleme, biyometrik bilgi tanıma, mikroskop görüntüleme, gözetleme, medya ve medikal alanlarda çeşitli amaçlar doğrultusunda kullanılabilir. Gözetleme sistemlerinde düşük maliyetli kamera kullanma ve nesne tanıma, medikal alanlarda Röntgen, MR (Manyetik Rezonans), Tomografi filmlerinin iyileştirilmesi, medya sektöründe transfer yükünü düşürerek maliyet azaltma gibi çeşitli amaçlar doğrultusunda kullanılabilir. Literatürde süper çözünürlük teknikleri genellikle yüz tanıma problemlerine çözüm olarak kullanılmaktadır (Lepcha vd. 2023). Bunun yanı sıra doğa, hayvan ve uydu görüntülerinin bulunduğu veri kümeleri ile çalışmalar yapılmaktadır.

Askeri alanlarda kullanılan Elektronik Harp (EH) tekniklerinde veri olarak görünür bant görüntülerinin yanı sıra elektromanyetik spektrumun farklı bölgelerinden alınan gri ölçekli görüntüler kullanılmaktadır. Bu görüntüler gözetleme ve EH sistemlerinde kullanılmaktadır. Elektromanyetik spektrumun farklı bölgelerinde çalışan kamera ve ilgili optik bileşenleri maliyetli olduğundan, yüksek çözünürlüklü kameralar tercih edilmemektedir. Özellikle Orta Dalga Kızılötesi (MWIR(Midwave Infrared)) elektromanyetik spektrum aralığında çalışan sistemler, askeri alanlarda sıkça kullanılan Kısa Dalga Kızılötesi (SWIR(Shortwave Infrared)) ve Uzun Dalga Kızılötesi (LWIR(Longwave Infrared)) bandında çalışan sistemlere göre daha maliyetlidir. Bunun en temel sebebi MWIR bandında genellikle soğutmalı detektörlerin kullanılmasıdır. MWIR bandında, yüksek çözünürlüklü detektörler yerine maliyet etkin şekilde derin öğrenme tabanlı süper çözünürlük tekniklerinin kullanılması kritik bir öneme sahiptir.

Literatürde çalışmalar içerisinde Droege vd. (2012) atmosferik türbülansı azaltacak şekilde bir yöntem geliştirmiş ve x3 ölçeğinde bir yukarı örnekleme yapabilmıştır. Mendez-Rial (2018) birden fazla açıklık (aperture) kullanarak MWIR ve LWIR görüntüler üzerinde bir süper çözünürlük metodu geliştirerek x2.5 ölçeğinde bir yukarı örnekleme yapabilmıştır. Wu ve Wang (2019) öğrenme tabanlı olmayan bir süper çözünürlük metodu geliştirmiş, bunu dijital mikro-ayna cihazı (digital micromirror device (DMD)) kullanarak sanal sahneler üzerinde denemiştir. Yapılan bu çalışmalar derin öğrenme tabanlı olmayan metotlar içermektedir. Lee vd. (2023) MWIR görüntülerde son derece küçük hedef olan (16x16 piksel) tank, kamyon ve insansız hava aracı LR

görüntülerini kullanarak GAN tabanlı bir stil dönüşüm ağı tasarlayarak uygulamıştır. x4 yukarı örnekleme ölçeği ile daha önce sunulan 2 adet GAN tabanlı ağ ile karşılaştırma yapılmış ve daha iyi sonuçlar edilmiştir. Literatürde mevcut olan derin öğrenme tabanlı süper çözünürlük tekniklerinin MWIR görüntüler üzerinde uygulanması ve karşılaştırılmasına dair çok sınırlı sayıda çalışma bulunmaktadır.

Tez çalışmasında 8 derin öğrenme tabanlı SR algoritması veri kümesi ile çalıştırılacak ve elde edilen sonuçlara göre algoritma mimarilerinde ve hiper-parametrelerinde değişiklik yapıp daha iyi sonuç alınması hedeflenecektir. Hiper-parametreler, derin öğrenme tabanlı yapılarda öğrenme sürecini kontrol eden, öğrenme sürecinin ne zaman sonlanacağını belirleyen ve model parametre değerlerini belirleyen parametrelerdir. Tez, derin öğrenme tabanlı süper çözünürlük tekniklerinin bahsi geçen görüntüleme sistemleri ile kullanılması ile beraber askeri alanda kullanılan uygulamaların başarımının artmasını amaçlamaktadır. Buna ek olarak, askeri alanlarda kullanılan düşük çözünürlüklü görüntülerinin, gerçeğe yakın şekilde yüksek çözünürlüklü görüntülere dönüştürülüp dönüştürülemeyeceği sorusu bu tez ile beraber cevaplanabilecektir. Tezin çıktıları, konunun askeri uygulamalarını araştıran ve derin öğrenme tabanlı algoritmalar üzerinde çalışan araştırmacılara fayda sağlayacaktır. Yapılacak olan çalışma literatüre katkı sağlamakla birlikte, makine öğrenmesi, derin öğrenme, yapay sinir ağları gibi güncel teknik konuların savunma sanayii içerisinde kullanımının bir örneği olacak ve ileri çalışmaları teşvik edecektir.

Günümüzde, bilgisayarlı görü görevlerinde yapay sinir ağları sıkça kullanılmaktadır. Yapay sinir ağları ve ilişkin terimler 2. Bölüm olan Kuramsal Temeller içerisinde sunulmuştur.

SRCNN, VDSR (Very Deep Convolutional Networks), ESPCN (Efficient Sub-Pixel Convolutional Neural Network), EDSR (Enhanced Deep Residual Networks), DCSCN (Deep CNN with Skip Connection and Network in Network), LapSRN (Deep Laplacian Pyramid Networks), RCAN (Residual Channel Attention Networks), IDN (Information Distillation Network) ve klasik bir yöntem olan biküçük interpolasyon algoritmaları, tez çalışmasında kullanılan algoritmalar. MWIR bandında alınmış görüntüleri içeren veri

kümesi %80 eğitim kümesi ve %20 test kümesi olacak şekilde ayrılmıştır. Veri kümesi insan, sivil ve askeri araç içeren doğa görüntülerinden oluşmaktadır. Algoritmalar grafik işlemci birimi (GPU (Graphical Processing Unit)) içeren bir bilgisayar ile eğitilmiş ve testler gerçekleştirilmiştir. Her bir algoritma özelinde birden fazla eğitim/test gerçekleştirilerek hiper-parametreler ayarlanmıştır. Performans metrikleri arasında Ortalama Kare Hata (MSE (Mean Square Error)), PSNR ve SSIM yer almaktadır. Algoritma detayları, veri kümesi içeriği, bilgisayar konfigürasyonu, uygulama detayları ve kullanılan performans metrikleri, 3. Bölüm olan Materyal ve Yöntem içerisinde sunulmuştur.

Eğitilen algoritmalar test kümesi üzerinde çalıştırılmış ve görüntü çıktıları kullanılarak performans kıyaslamaları yapılmıştır. Elde edilen görüntüler, performans sonuçları ve algoritma eğitim/test süreleri 4. Bölüm olan Araştırma Bulguları içerisinde sunulmuştur.

Tez çalışması sonuçları ve ilgili tartışmalar, 5. Bölüm olan Tartışma ve Sonuç kısmında sunulmuştur.

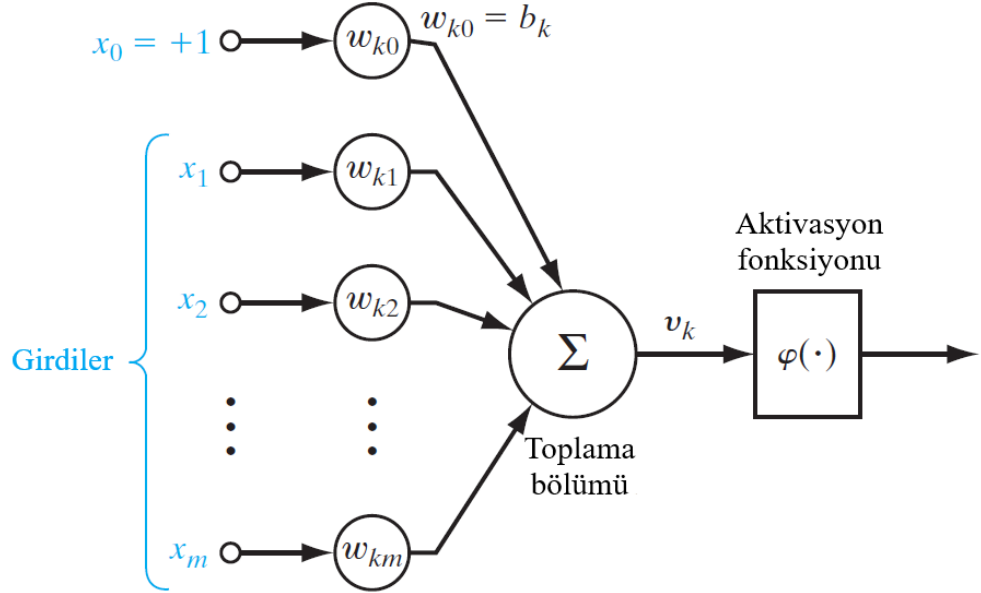
2. KURAMSAL TEMELLER

2.1 Yapay Sinir Ağları

Yapay sinir ağları başka bir deyişle sinir ağları, insan beyninin geleneksel dijital bilgisayardan tamamen farklı bir şekilde hesaplama yaptığı bilinerek hazırlanan birbirine sinyal gönderen bağlı birimlerin oluşturduğu gruplar olarak ifade edilebilir (Haykin 2009). Yapay sinir ağları insan beyni örnek alınarak öğrenme sürecinin matematiksel olarak ifade edilmesiyle oluşturulmuştur.

2.2 Sinir (Neuron) Yapısı

Sinir, sinir ağı operasyonunun temelini oluşturan bir bilgi işleme birimidir. Sinirler farklı ağırlıkları olan bağlantılar ile bağlıdır. Sinir modeli şekil 2.1 ile verilmiştir.



Şekil 2.1 Sinir (neuron) modeli

Girdilerin toplama bölümündeki karşılığı u_k eşitlik (2.1) ile verilmiştir. x girdi değerlerini, w ağırlıkları temsil etmekte olup m adet bağlantı mevcuttur.

$$u_k = \sum_{j=1}^m w_{kj} x_j \quad (2.1)$$

u_k tüm girdilerin ağırlıkları ile çarpımlarının toplamına eşittir. Yanlılık değeri, x_0 sabiti ile yanlılık ağırlığının çarpımına eşit olup eşitlik (2.2) ile verilmiştir.

$$b_k = +1 * w_{k0} \quad (2.2)$$

Toplama bölümü çıktısı v_k , u_k değeri ile yanlılık değerinin toplamına eşit olup eşitlik (2.3) ile verilmiştir.

$$v_k = u_k + b_k \quad (2.3)$$

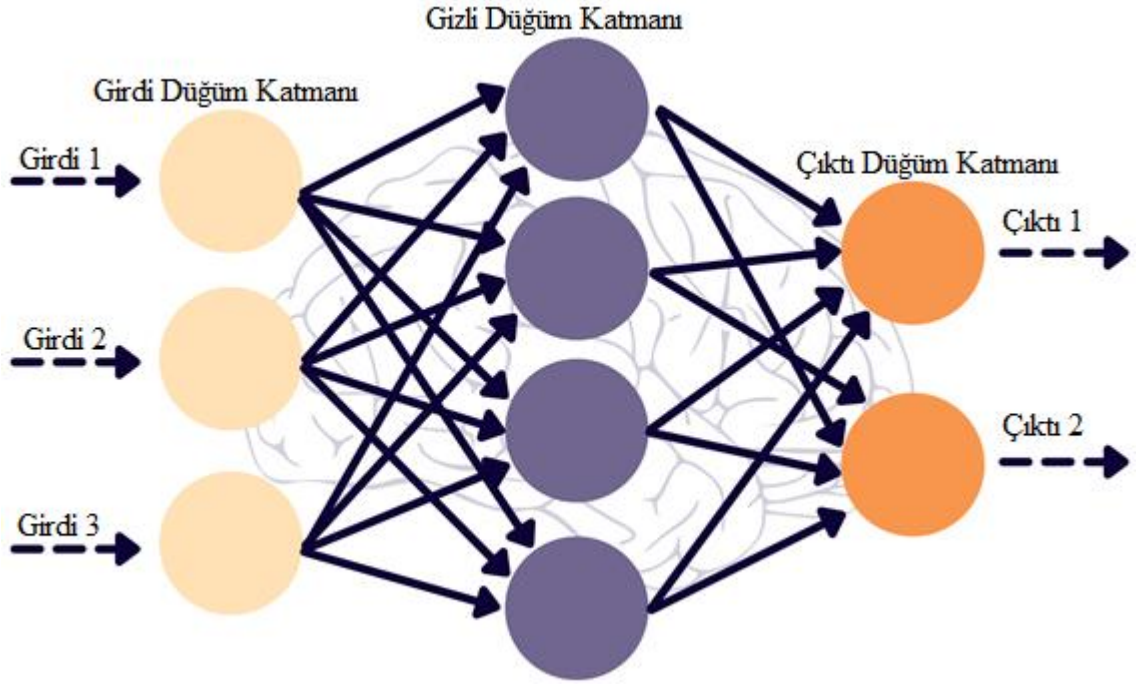
Çıktı sinyali, toplama bölümü çıktısının aktivasyon fonksiyonu (φ) ile işlemi sonucu elde edilmekte olup eşitlik (2.4) ile verilmiştir.

$$y_k = \varphi(u_k + b_k) \quad (2.4)$$

2.3 Derin Sinir Ağ Yapısı

Derin sinir ağları, girdi katmanı, gizli katmanlar ve çıktı katmanından oluşmaktadır. Derin sinir ağlarında eğitim sürecinin başlangıcında girdiler girdi katmanına giriş yaparlar. Bu katmanda girdiler gerekli ağırlıklarla çarpılıp, sapma değerleri eklenerek sonraki katmana aktarılırlar. Benzer matematiksel işlemler gerçekleştirilerek çıktı katmanında sonuçlar elde edilir. Elde edilen sonuçlar, gerçek değerler ile kayıp fonksiyonu kullanılarak karşılaştırılır. Elde edilen sonuçlar geri yayılım metodu ile ağa beslenir. Bu metot kısmi türevlerin kullanıldığı ve çeşitli matematiksel hesaplamaların yapıldığı bir işlemler bütünüdür. Son katmandan ilk katmana doğru yayılım sürdürülür ve ağırlıklar güncellenir. Ardından farklı bir girdi verilerek işlem tekrarlandıktan sonra ağırlıkların yakınsadığı ya da iterasyon adedinin sonlandığı zamana kadar bu işlem devam ettirilir. Böylece eğitim tamamlanarak sinir ağının parametreleri yakınsamış olur.

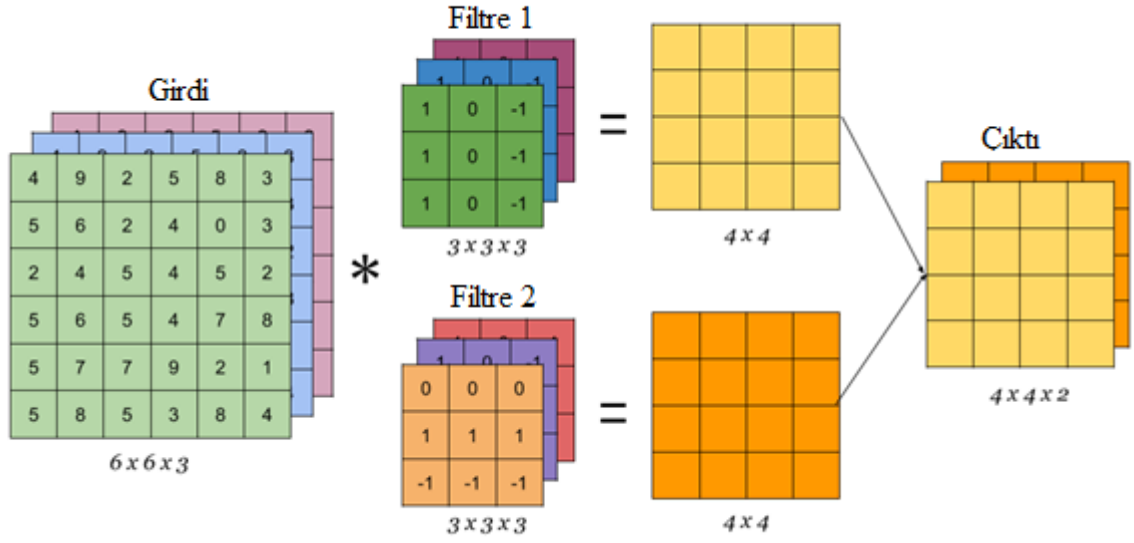
Derin sinir ađ genel yapısı Őekil 2.2 ile verilmiŐtir.



Őekil 2.2 Derin sinir ađ yapısı

2.4 EvriŐimsel (Convolutional) Sinir Ađları

Bir evriŐimsel (convolutional) sinir ađı, özellikle iki boyutlu Őekilleri çevirme, ölçekleme, eğme ve diđer bozulma türlerine karŐı yüksek derecede deđiŐmezlikle tanımak için tasarlanmış çok katmanlı bir algılayıcıdır (Haykin, 2009). EvriŐimsel sinir ađları, frekans bölgesinde çarpma işlemine denk gelen evriŐim işlemi ile genellikle görüntü üzerinde kullanılırlar. Nesne tespiti, nesne takibi, görüntü bölütleme, görüntü üretme ve otonom sistemler gibi birçok bilgisayarlı görü alanında sıkça kullanılmaktadır. Girdi görüntüleri ile filtreler evriŐim sonrası çıktı vermekte ve kanal sayısı kadar çıktı alınabilmektedir. Siyah-beyaz görüntüler 1, renkli görüntüler 3 kanala sahiptir. EvriŐim işlemi, filtrelerin görüntü üzerinde gezdirilerek her bir piksel için toplama işleminin yapıldıđı işlemdir. 3 kanallı bir görüntü için evriŐim işlemi örneđi Őekil 2.3 ile verilmiŐtir.



Şekil 2.3 Evrişim işlemi

2.5 Aktivasyon Fonksiyonu

Çok katmanlı algılayıcının her bir nöronunun yerel (local) gradyanının (gradient) hesaplanması, o nöronla ilişkilendirilen aktivasyon fonksiyonunun türevini bilmeyi gerektirir. Bu türevin var olması için, aktivasyon fonksiyonunun sürekli olması gerekir. Bu sebeple farklılanabilirlik (differentiability), bir aktivasyon fonksiyonunun karşılaması gereken tek gereksinimdir (Haykin 2009). Literatürde çeşitli aktivasyon fonksiyonları kullanılmakta olup, bazıları alt başlıklar altında verilmiştir.

2.5.1 Eşik (threshold) aktivasyon fonksiyonu

Eşik (threshold) aktivasyon fonksiyonu, genellikle ikili sınıflandırma problemlerinde kullanılan basit bir aktivasyon fonksiyonudur. Bu fonksiyon, bir eşik değeri belirler ve giriş değeri bu eşik değerinden büyükse, çıktıyı 1'e, aksi takdirde 0'a ayarlar. Bu nedenle, eşik fonksiyonu, çıkışı bir eşik değerine göre sınıflandırır. Eşik aktivasyon fonksiyonu eşitlik (2.5) ile verilmiştir.

$$f(x) = \begin{cases} 1, & \text{eğer } x \geq \text{eşik değeri} \\ 0, & \text{eğer } x < \text{eşik değeri} \end{cases} \quad (2.5)$$

Eşik aktivasyon fonksiyonunun basitliği ve doğrudan doğruya karar verme yeteneği, onu bazı temel sınıflandırma problemlerinde kullanışlı hale getirir. Ancak, çoğu zaman daha karmaşık problemlerde kullanılan diğer aktivasyon fonksiyonları, özellikle gradyanlarının hesaplanması ve geri yayılım algoritmasının etkinliği açısından, eşik fonksiyonundan daha iyi performans gösterir. Bu fonksiyon, genellikle basit yapısı nedeniyle daha karmaşık derin öğrenme modellerinde kullanılmamaktadır.

2.5.2 Sigmoid aktivasyon fonksiyonu

Sigmoid aktivasyon fonksiyonu eşitlik (2.6) ile verilmiştir.

$$\sigma(x) = \frac{1}{1 + e^{-x}} \quad (2.6)$$

Sigmoid aktivasyon fonksiyonu, bir giriş değerini alarak, bu değeri 0 ile 1 arasında bir çıktıya dönüştürür. Sigmoid fonksiyonunun S-şeklindeki eğrisi, küçük ve büyük giriş değerlerinde doygunluk (saturation) sağlarken, orta büyüklükteki giriş değerleri için daha belirgin bir gradyan sunar. Bu sayede, modelin belirli aralıklardaki değişimlere duyarlı olması sağlanır. Fonksiyonun çıktısının 0 ile 1 arasında olması, özellikle sınıflandırma problemlerinde, çıktıların olasılık olarak yorumlanmasını mümkün kılar. Bununla birlikte, sigmoid fonksiyonunun büyük ve küçük giriş değerleri için türevlerinin çok küçük olması, geri yayılım algoritması sırasında gradyanların kaybolmasına (vanishing gradient) yol açabilir. Bu durum, derin ağların eğitimi sırasında öğrenmeyi zorlaştırabilir. Sigmoid fonksiyonu, özellikle doğrusal olmayan dönüşümler gerektiren durumlarda, giriş verilerinin etkili bir şekilde işlenmesine yardımcı olur. Ancak, modern derin öğrenme uygulamalarında, kayıp gradyan problemini aşmak için genellikle ReLU gibi alternatif aktivasyon fonksiyonları tercih edilmektedir.

2.5.3 Hiperbolik tanjant aktivasyon fonksiyonu

Hiperbolik tanjant aktivasyon fonksiyonu eşitlik (2.7) ile verilmiştir.

$$\tanh(x) = \frac{2}{1 + e^{-2x}} - 1 \quad (2.7)$$

Hiperbolik tanjant (tanh) aktivasyon fonksiyonu, yapay sinir ağlarında sıkça kullanılan bir aktivasyon fonksiyonudur ve sigmoid fonksiyonuna benzer şekilde doğrusal olmayan dönüşümler sağlar. Ancak, tanh fonksiyonu giriş değerlerini -1 ile 1 arasında bir aralığa dönüştürür, bu da onu bazı durumlarda sigmoid fonksiyonundan daha avantajlı hale getirir. Hiperbolik tanjant fonksiyonunun grafiği, S-şeklinde olup, negatif ve pozitif giriş değerlerine karşı simetrik bir davranış sergiler. Bu özellik, tanh fonksiyonunun sıfır merkezi etrafında çalışmasını sağlar ve çıktılarının negatif, sıfır veya pozitif olabilmesi, sinir ağının öğrenme dinamiklerine olumlu katkıda bulunur. Tanh fonksiyonu, büyük pozitif giriş değerleri için 1'e, büyük negatif giriş değerleri için ise -1'e yaklaşır. Bu sayede, tanh fonksiyonu, verilerdeki büyük değişimleri daha etkili bir şekilde yakalar ve özellikle daha derin sinir ağlarında daha kararlı bir öğrenme süreci sunar. Ancak, sigmoid fonksiyonunda olduğu gibi, tanh fonksiyonu da giriş değerleri çok büyük veya çok küçük olduğunda gradyanların kaybolmasına neden olabilir. Yine de, sıfır merkezi özelliği nedeniyle tanh, genellikle sigmoid fonksiyonuna kıyasla daha iyi performans gösterir.

2.5.4 RELU aktivasyon fonksiyonu

RELU aktivasyon fonksiyonu eşitlik (2.8) ile verilmiştir.

$$RELU(x) = \max(0, x) \quad (2.8)$$

ReLU (Rectified Linear Unit) aktivasyon fonksiyonu, modern yapay sinir ağlarında en çok kullanılan aktivasyon fonksiyonlarından biridir. ReLU, giriş değerinin pozitif olup olmadığını kontrol eder ve pozitif ise giriş değerini değiştirmezken, negatif ise sıfıra

eşitler. Bu basit işleyiş ile ReLU fonksiyonu, diğer aktivasyon fonksiyonlarına göre birkaç önemli avantaja sahiptir. Öncelikle, ReLU fonksiyonu doğrusal olmayan bir fonksiyondur, bu da sinir ağının karmaşık ilişkileri öğrenebilmesini sağlar. Ancak, ReLU'nun sigmoid ve tanh fonksiyonlarından farkı, gradyanlarının büyük değerler alabilmesidir. Bu durum, geri yayılım (backpropagation) sırasında gradyanların kaybolma problemini (vanishing gradient) büyük ölçüde azaltır, böylece derin ağların daha hızlı ve etkili bir şekilde öğrenmesini sağlar. ReLU fonksiyonunun bir diğer avantajı, hesaplama açısından çok basit ve hızlı olmasıdır. Sadece giriş değerinin pozitif ve negatifliğini kontrol ederek hesaplama maliyetini düşürür ve eğitim süresini hızlandırır. Bununla birlikte, ReLU'nun bazı dezavantajları bulunmaktadır. Örneğin, giriş değeri negatif olduğunda çıkış her zaman sıfır olduğundan, bazı nöronlar hiç etkinleşmeyebilir ve bu durum ölü nöron problemi (dying ReLU) olarak adlandırılır. Bu sorunu hafifletmek için Leaky ReLU ve Parametric ReLU gibi ReLU'nun çeşitli varyantları geliştirilmiştir. Sonuç olarak, ReLU aktivasyon fonksiyonu, basitliği ve etkinliği nedeniyle derin öğrenme modellerinde yaygın olarak kullanılır. Özellikle, derin sinir ağlarının eğitimi sırasında hızlı öğrenmeyi ve gradyanların kaybolma probleminin azaltılmasını sağlar, bu da daha derin ve daha karmaşık modellerin başarılı bir şekilde eğitilmesine olanak tanır.

2.5.5 Leaky RELU aktivasyon fonksiyonu

Leaky ReLU aktivasyon fonksiyonu eşitlik (2.9) ile verilmiştir.

$$\text{Leaky RELU}(x) = \max(0.01x, x) \quad (2.9)$$

Leaky ReLU (Leaky Rectified Linear Unit) aktivasyon fonksiyonu, standart ReLU fonksiyonunun bir varyantıdır ve temel amacı, standart ReLU'nun ölü nöron (dying ReLU) problemine çözüm getirmektir. Standart ReLU fonksiyonunda, giriş değeri negatif olduğunda çıktı sıfır olur. Bu durum, bazı nöronların sürekli sıfır değer üretmesi ve dolayısıyla eğitim sırasında hiç etkinleşmemesi anlamına gelir. Leaky ReLU, bu sorunu aşmak için negatif giriş değerleri için küçük bir pozitif eğim kullanır. Leaky ReLU, giriş değeri pozitif olduğunda standart ReLU gibi davranır, başka bir deyişle çıktı giriş

değerine eşittir. Ancak, giriş değeri negatif olduğunda, çıktı, negatif giriş değerinin küçük bir katsayı ile çarpılmış halidir. Bu katsayı genellikle küçük bir pozitif sayı olarak seçilir. Böylece, negatif giriş değerleri sıfıra eşitlenmez ve küçük bir değerle ifade edilir. Bu küçük değişiklik, Leaky ReLU'nun birkaç önemli avantaja sahip olmasını sağlar. İlk olarak, negatif giriş değerleri için küçük de olsa bir gradyan sağlandığından, ölü nöron problemi büyük ölçüde azalır. Bu, ağına daha fazla nöronu etkin bir şekilde kullanmasını ve öğrenme sürecinin daha verimli olmasını sağlar. İkinci olarak, Leaky ReLU, negatif değerler için bile bilgi akışını koruyarak daha dengeli bir öğrenme süreci sunar. Leaky ReLU'nun matematiksel tanımı, giriş değeri pozitif olduğunda giriş değerini, negatif olduğunda ise negatif giriş değerinin küçük bir pozitif katsayı ile çarpılmış halini alır. Bu, hesaplama açısından oldukça basit ve hızlıdır, bu nedenle modern derin öğrenme uygulamalarında sıkça tercih edilir. Negatif giriş değerleri için küçük bir pozitif eğim kullanarak, nöronların etkinliğini artırır ve öğrenme sürecini daha dengeli hale getirir.

2.5.6 Softmax aktivasyon fonksiyonu

Softmax aktivasyon fonksiyonu eşitlik (2.10) ile verilmiştir.

$$\text{Softmax}(x_i) = \frac{e^{x_i}}{\sum_j e^{x_j}} \quad (2.10)$$

Softmax aktivasyon fonksiyonu, yapay sinir ağlarında yaygın olarak kullanılan bir aktivasyon fonksiyonudur. Bu fonksiyon, genellikle çok sınıflı sınıflandırma problemlerinde çıktı katmanında kullanılır. Softmax, her bir giriş değerini sınıflar arasında bir olasılık dağılımına dönüştürür. Özellikle, her bir giriş değeri üzerinde üstel fonksiyon uygulanarak elde edilen çıktılar, daha sonra bu üstel değerlerin toplamına bölünerek her bir sınıfın olasılığı hesaplanır. Böylece, her bir sınıfın olasılığı 0 ile 1 arasında bir değer alır ve tüm sınıf olasılıklarının toplamı 1'e eşittir. Softmax fonksiyonunun bu özelliği, sınıflar arasında doğru bir olasılık dağılımı sağlar ve modelin en olası sınıf tahminini yapmasını sağlar. Bu sayede, Softmax fonksiyonu, sinir ağlarının sınıflandırma problemlerindeki çıktıları yorumlamayı kolaylaştırır ve modelin güven seviyesini değerlendirmeye olanak tanır.

2.6 Geri Yayılım (Back Propagation)

Geri yayılım (back propagation), yapay sinir ağlarının eğitiminde kullanılan temel bir optimizasyon algoritmasıdır. Bu algoritma, ağın verilen bir eğitim örneğini işledikten sonra, tahminler ile gerçek etiketler arasındaki hatayı hesaplayarak, ağı geriye doğru hareket ettirir ve ağırlıkların güncellenmesine yardımcı olur. Geri yayılım, bir gradyan iniş algoritmasıdır ve hedef, ağın tahminlerini gerçek etiketlere mümkün olduğunca yaklaştırmaktır.

İlk olarak, bir eğitim örneği ağı beslenir ve ağın çıktıları hesaplanır. Ardından, bu çıktılar gerçek etiketlerle karşılaştırılarak bir hata değeri belirlenir. Hata, genellikle bir kayıp (loss) fonksiyonu kullanılarak ölçülür. Daha sonra, bu hata değeri ağın içinden geriye doğru yayılır ve her katmandaki ağırlıkların ve yanlıkların hataya katkısını hesaplamak için gradyanlar hesaplanır.

Geri yayılımın ana prensibi, zincir kuralı kullanarak gradyanları geriye doğru hesaplamaktır. Her bir katmanın gradyanı, bir sonraki katmandan gelen gradyanın ve katmanın aktivasyon fonksiyonunun türevinin çarpımıdır. Bu şekilde, ağın her bir parçasının hataya katkısı belirlenir ve bu bilgi, ağırlıkların güncellenmesinde kullanılır.

Geri yayılım, ağın her bir ağırlığının ve yanlılığının hataya olan katkısını hesaplamak için gerçekleştirilen bir hesaplama işlemidir. Bu işlem, ağın optimize edilmesini ve eğitim verilerine uyum sağlamasını sağlar. Genellikle, gradyan iniş gibi optimizasyon algoritmalarıyla birlikte kullanılır ve ağın parametrelerini güncellemek için gradyanları kullanır.

2.7 Öğrenme Oranı

Öğrenme oranı (learning rate), sinir ağ yapılarında kullanılan bir hiper-parametredir. Bu parametre, modelin her bir yinelemede (iteration) ağırlıklarının ne kadar değiştirileceğini belirler. Doğru öğrenme oranı seçimi modelin hızlı ve etkili bir şekilde öğrenmesini

sağlarken, yanlış bir öğrenme oranı seçimi eğitim sürecini yavaşlatabilir veya başarısız olmasına neden olabilir.

Öğrenme oranının sabit olduğu durumda, $w(n)$ n numaralı ağırlık ve $x(n)$ n numaralı girdi olarak kabul edildiğinde öğrenme oranı $\eta(n)$ eşitlik (2.11) ve (2.12) ile ifade edilebilir (Haykin 2009).

$$w(n + 1) = w(n) - \eta(n)x(n) \text{ eğer } \omega^T(n)x(n) > 0 \quad (2.11)$$

$$w(n + 1) = w(n) + \eta(n)x(n) \text{ eğer } \omega^T(n)x(n) \leq 0 \quad (2.12)$$

Sabit öğrenme oranının yanı sıra azalan öğrenme oranı ve uyarlanabilir (adaptive) öğrenme oranı, literatürdeki çalışmalarda yaygın olarak kullanılmaktadır.

2.8 Öğrenme Yöntemleri

Her bir öğrenme yöntemi, farklı türde veri ve belirli bir öğrenme hedefi için uygun olmaktadır. Makine öğrenmesi uygulamalarında, genellikle bu farklı metotların kombinasyonları kullanılarak daha karmaşık problemler çözülmektedir. Öğrenme metotları alt başlıklar ile verilmiştir.

2.8.1 Denetimli öğrenme (supervised learning)

Denetimli öğrenme etiketli örneklerle model kurma yöntemidir. Verileri etiketlemek, verileri farklı kategorilere ayırmak anlamına gelir. Bu etiketleme esas olarak denetimli öğrenmede gerçekleşir. Bu nedenle denetimli öğrenmenin amacı, bir veri örneği ve istenen çıktılar verildiğinde, girdi ve çıktı arasındaki ilişkiyi en iyi şekilde tahmin eden bir model oluşturmaktır.

Avantajları:

- En kolay ve yaygınlıkla kullanılan öğrenme yöntemi türüdür.
- Eğitim verileri yalnızca modeli eğitirken kullanılmaktadır. Hafızada yer kaplamasına karşın, eğitim tamamlandıktan sonra hafızadan silinebilmektedir.

Dezavantajları:

- Kendi etiketlerini oluşturmadığından, denetimsiz öğrenme gibi verileri kendi başına öğrenmemektedir.
- Mevcut sınıflara ait bir veri sunulmadığı takdirde yanlış sonuç elde edilmesi olasıdır.
- Eğitim için daha güçlü bilgisayarlara ihtiyaç duyulmaktadır.

2.8.2 Denetimsiz öğrenme (unsupervised learning)

Denetimsiz öğrenmede genellikle etiketlenmemiş verilerle çalışılmaktadır. Bu yöntem, veri kümesindeki gizli yapıları ve desenleri keşfetmeyi amaçlamaktadır. Eğitim verilerinin girdi ve çıktıları birlikte sunulmadığı için model, veri kümesindeki doğal yapıları öğrenmek için kendiliğinden düzenleme yapmaktadır. Etiketleme yapılmadan sınıflandırma yapılması, etiketlerin veriler sınıflandırıldıktan sonra eklenebilmesi ve işlenmemiş verilerin anlaşılması için daha kullanışlı olması avantajları arasında yer alırken daha yavaş bir öğrenme süreci gerçekleşmektedir.

2.8.3 Yarı-denetimli öğrenme (semi-supervised learning)

Yarı denetimli öğrenme, hem etiketlenmiş hem de etiketlenmemiş verilerin birlikte kullanıldığı bir makine öğrenmesi yaklaşımıdır. Bu yöntem, genellikle etiketlenmiş verilerin az olduğu durumlarda kullanılırken, etiketlenmemiş verilerin fazla olmasından yararlanarak modelin performansını artırmayı amaçlamaktadır.

Yarı denetimli öğrenme, denetimli ve denetimsiz öğrenme arasında bir köprü görevi görür. Bir kısmı etiketlenmiş olan veri kümesi, genellikle zaman alıcı bir şekilde toplanırken, etiketlenmemiş veriler daha kolay toplanabilmektedir. Yarı denetimli öğrenme, hem etiketlenmiş hem de etiketlenmemiş verileri bir araya getirerek modelin daha iyi bir şekilde eğitilmesini sağlamaktadır.

2.8.4 Kendi kendine denetimli öğrenme (self-supervised learning)

Kendi kendine denetimli öğrenme, modelin büyük miktarda etiketlenmemiş veriyi kullanarak öğrenmesini sağlar. Bu yöntem, etiketsiz verilerden yararlanarak modelin kendisini eğitmesi için otomatik olarak oluşturulan etiketler veya görevler kullanır. Kendi kendine denetimli öğrenmenin temel ilkesi, veride mevcut olan içsel yapıları ve desenleri kullanarak modelin öğrenmesini sağlamaktır. Model, verinin bir kısmını tahmin etmeye çalışırken diğer kısmını girdi olarak kullanır. Bu, modelin güçlü temsiller öğrenmesine ve daha sonra bu temsilleri denetimli öğrenme görevlerinde kullanmasına olanak tanır.

2.8.5 Zayıf denetimli öğrenme (weakly-supervised learning)

Zayıf denetimli öğrenme, etiketli verilerin sınırlı, hatalı veya belirsiz olduğu durumlarda kullanılan bir makine öğrenimi yaklaşımıdır. Bu yöntem, denetimli öğrenmenin gerektirdiği yoğun ve maliyetli etiketleme sürecini azaltmayı hedefler. Bu yöntemin başarılı bir şekilde uygulanması, modelin gerçek dünyadaki karmaşık ve çeşitli veri kaynaklarından daha iyi öğrenmesini sağlar. Örneğin, tıbbi görüntüleme, doğal dil işleme ve nesne tanıma gibi alanlarda yaygın olarak kullanılır. Zayıf denetimli öğrenme, makine öğrenimi modellerinin geniş çapta veri kaynaklarından öğrenmesini sağlayarak, daha genel ve uyarlanabilir modellerin geliştirilmesine katkıda bulunur.

2.9 Epok (Epoch)

Epok, bir yapay sinir ağının tüm eğitim veri kümesini bir kez ileri ve geri yayılım işlemiyle işlemesi olarak ifade edilir. Başka bir deyişle, eğitim veri kümesindeki her bir

örneğin sinir ağı tarafından bir kez işlenmesi tamamlandığında bir epok gerçekleşmiş olur. Epok, modelin eğitim sürecinde kullanılan bir parametre olup, modelin ne kadar süre ve ne derinlikte eğitileceğini belirlemektedir. Epok sayısının düşük olması eksik öğrenmeye (underfitting) sebep olurken, fazla olması aşırı öğrenmeye (overfitting) sebep olur. Her bir epok sonrası modelin performansı bir kayıp fonksiyonu (loss function) veya bir doğruluk metriği kullanılarak değerlendirilmektedir.

2.10 Kayıp Fonksiyonu (Loss Function)

Kayıp fonksiyonu (loss function), makine öğrenmesi ve derin öğrenme modellerinde modelin tahminlerinin gerçek değerlerden ne kadar uzak olduğunu ölçen bir metriktir. Modelin performansını değerlendirmek ve optimize etmek için kullanılır. Kayıp fonksiyonları, modelin eğitim sürecinde kullanılan geri yayılım (back-propagation) algoritması ile ağırlıkların güncellenmesini sağlar ve çıktısını 0 değerine yaklaştırarak modeli düzeltmeye çalışır.

Birçok farklı kayıp fonksiyonu türü vardır ve seçilen kayıp fonksiyonu, modelin problem türüne ve veri dağılımına bağlı olarak değişebilir. Modelin doğru tahminler yapmasını sağlamak için, uygun bir kayıp fonksiyonunun seçilmesi ve bu fonksiyonun optimize edilmesi önemlidir. Bu sayede, modelin eğitimi daha etkili hale gelir ve daha iyi sonuçlar elde edilir. Literatürde çeşitli kayıp fonksiyonları kullanılmakta olup, bazıları alt başlıklar altında verilmiştir.

2.10.1 Ortalama kare hata (mean squared error (MSE))

Ortalama kare hata ağ çıktı değerleri ile gerçek değerler arasındaki farkların karesinin ortalamasını alır. MSE kayıp fonksiyonu eşitlik (2.13) ile verilmiştir. y_i gerçek değeri, \hat{y}_i ağ çıktısını ve n veri örnek sayısını ifade etmektedir.

$$MSE = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2 \quad (2.13)$$

2.10.2 Ortalama mutlak hata (mean absolute error (MAE))

Ortalama mutlak hata ağ çıktı değerleri ile gerçek değerler arasındaki farkların mutlak değerlerinin ortalamasını alır. MAE kayıp fonksiyonu eşitlik (2.14) ile verilmiştir. y_i gerçek değeri, \hat{y}_i ağ çıktısını ve n veri örnek sayısını ifade etmektedir.

$$MAE = \frac{1}{n} \sum_{i=1}^n |y_i - \hat{y}_i| \quad (2.14)$$

2.10.3 Logaritmik kayıp (logarithmic loss / log loss)

Log loss, aynı zamanda cross-entropy loss olarak da adlandırılan, sınıflandırma problemlerinde yaygın olarak kullanılan bir kayıp fonksiyonudur. Özellikle, tahmin edilen olasılıkların gerçek etiketlerle ne kadar uyumlu olduğunu değerlendirmek için kullanılır. Log loss, her bir örnek için, gerçek etiketlere (0 veya 1) karşılık gelen tahmini olasılığın negatif logaritması alınarak hesaplanır. Log Loss aktivasyon fonksiyonu eşitlik (2.15) ile verilmiştir. y_i gerçek değeri, \hat{y}_i ağ çıktısını ve n veri örnek sayısını ifade etmektedir.

$$Log Loss = -\frac{1}{n} \sum_{i=1}^n [y_i \log(\hat{y}_i) + (1 - y_i) \log(1 - \hat{y}_i)] \quad (2.15)$$

Formül her bir örneğin logaritmik hata miktarını ölçmekte ve küçük olması istenen bir kayıp değeridir. İdeal durumlarda değer 0'a yaklaşırken, yanlış tahminlerde değer artmaktadır. Log loss, modelin sınıflar arasındaki belirsizliği azaltmasına ve doğru sınıflandırma yapmasına yardımcı olmaktadır.

2.11 Aşırı Öğrenme (Overfitting)

Aşırı öğrenme (overfitting), makine öğrenmesi ve istatistikte sıkça karşılaşılan ve istenmeyen bir durumdur. Bu durum, bir modelin eğitim verileri üzerinde çok iyi performans göstermesi, ancak yeni ve görülmemiş veriler üzerinde yetersiz performans sergilemesi anlamına gelir. Model, eğitim verisindeki en küçük ayrıntıları, gürültüyü ve rastlantısal hataları öğrenir ve bu da onun genelleme yeteneğini azaltır.

Aşırı Öğrenmenin Belirtileri:

- Düşük Eğitim Hatası, Yüksek Test Hatası

Model, eğitim verisi üzerinde çok düşük hata oranlarına sahip olabilir, fakat test veya doğrulama verisi üzerinde performansı önemli ölçüde düşer. Bu durum, modelin eğitim verisine aşırı derecede uyum sağladığını ve yeni veriler üzerinde genelleme yapamadığını gösterir.

- Model Karmaşıklığı

Aşırı öğrenme genellikle çok karmaşık modellerde ortaya çıkar. Bu modeller, gereğinden fazla parametre içerir ve eğitim verisindeki gürültüleri bile öğrenir. Sonuç olarak, modelin kapasitesi gereğinden fazla olur.

- Yüksek Varyans

Aşırı öğrenmiş modeller, küçük veri değişikliklerine karşı bile çok hassastır. Bu durum, modelin varyansının yüksek olduğunu ve farklı veri kümeleri üzerinde çok farklı performans gösterebileceğini ifade eder.

Aşırı Öğrenmenin Çözümleri:

- Daha Fazla Eğitim Verisi Toplama

Daha fazla veri toplamak, modelin genelleme yeteneğini artırabilir. Daha büyük veri kümeleri, modelin eğitim verisindeki gürültüleri öğrenmesini zorlaştırır.

- Regularizasyon Teknikleri Kullanma

L1 ve L2 regularizasyonu gibi teknikler, modelin ağırlıklarını sınırlayarak aşırı öğrenmeyi önler. Bu teknikler, modelin parametrelerini küçültür ve böylece daha az karmaşık modeller elde edilir.

- Model Basitliğini Arttırma

Daha basit modeller, aşırı öğrenmeye daha az eğilimlidir. Örneğin, daha az katmanlı veya daha az parametrelili modeller tercih edilebilir.

- Çapraz Doğrulama (Cross-Validation)

Eğitim sürecinde çapraz doğrulama teknikleri kullanmak, modelin genelleme yeteneğini değerlendirmeye yardımcı olur. Bu yöntem, modelin farklı veri alt kümeleri üzerinde nasıl performans gösterdiğini analiz eder.

- Erken Durdurma (Early Stopping)

Model eğitimi sırasında doğrulama hatası belirli bir eşiğin üzerine çıkmaya başladığında eğitimi durdurmak, aşırı öğrenmeyi önleyebilir. Bu yöntem, modelin gereğinden fazla öğrenmesini engeller.

2.12 Eksik Öğrenme (Underfitting)

Eksik öğrenme (underfitting), bir makine öğrenmesi modelinin hem eğitim verisi hem de test verisi üzerinde düşük performans göstermesi durumudur. Bu, modelin eğitim verisindeki temel desenleri bile öğrenemediği anlamına gelir. Model, veriler arasındaki ilişkileri yeterince iyi kavrayamaz ve bu da hem eğitim hem de test verisinde yüksek hata oranlarına yol açar.

Eksik Öğrenmenin Belirtileri:

- Yüksek Eğitim ve Test Hatası

Model, hem eğitim verisi hem de test verisi üzerinde yüksek hata oranlarına sahiptir. Bu durum, modelin karmaşık desenleri öğrenemediğini ve yetersiz kaldığını gösterir.

- Model Basitliği

Eksik öğrenme genellikle çok basit modellerde ortaya çıkar. Bu modeller, yeterince parametre içermez veya verilerdeki karmaşık desenleri yakalayamaz.

- Düşük Varyans

Eksik öğrenmiş modeller, veri kümesi değişikliklerine karşı daha az hassastır. Bu durum, modelin varyansının düşük olduğunu ve her veri kümesi üzerinde benzer performans gösterdiğini ifade eder.

Eksik Öğrenmenin Çözümleri:

- Daha Karmaşık Modeller Kullanma

Daha karmaşık modeller veya daha fazla katman içeren derin öğrenme modelleri kullanılabilir. Bu modeller, verilerdeki karmaşık desenleri daha iyi öğrenebilir.

- Öznitelik Mühendisliği

Daha fazla ve daha iyi öznitelikler (features) oluşturulabilir. Öznitelik mühendisliği, verilerdeki önemli desenleri modelin daha iyi öğrenmesine yardımcı olur.

- Modelin Eğitim Süresini Uzatma

Modelin eğitim süresi uzatılarak, veriler üzerindeki desenleri daha iyi öğrenmesi sağlanabilir. Ancak bu, aşırı öğrenmeye de yol açabileceğinden dikkatli olunmalıdır.

- Hiper-parametre Ayarlamaları

Modelin hiper-parametreleri dikkatlice ayarlanarak, performansı artırılabilir. Örneğin, öğrenme oranı, düzenleme parametreleri ve katman sayısı gibi hiper-parametreler optimize edilebilir.

- Daha Fazla Öznitelik Ekleme

Daha fazla bağımsız öznitelik eklenerek modelin verileri daha iyi kavraması sağlanabilir. Bu, modelin kapasitesini artırabilir ve daha iyi genelleme yapmasına yardımcı olabilir.

2.13 Yığın Boyutu (Batch Size)

Yığın boyutu, bir makine öğrenme modelinin eğitimi sırasında modelin ağırlıklarını güncellemek için kullanılan eğitim örneklerinin sayısını ifade eder. Makine öğrenme

modelleri, genellikle eğitim verilerini bir kerede değil, yığınlar halinde işler ve her yığın için modelin parametreleri günceller. Bu işlem, eğitim sürecini hızlandırabilmekte ve daha büyük veri kümeleriyle çalışmayı mümkün kılabilir.

Yığın Boyutu Türleri:

- Tam Yığın İniş (Batch Gradient Descent)

Tüm eğitim verisi kullanılarak modelin ağırlıkları güncellenir. Başka bir deyişle, her iterasyonda tüm veri kümesi üzerinden geçilir. Teorik olarak en doğru ağırlık güncellemesini gerçekleştirme avantaj sağlamasına rağmen, büyük veri kümelerinde çok yavaş olabilmekte ve yüksek hesaplama gücü gerektirebilmektedir.

- Stokastik Gradyan İniş (Stochastic Gradient Descent)

Her iterasyonda yalnızca tek bir eğitim örneği kullanılarak modelin ağırlıkları güncellenir. Daha hızlı olması ve büyük veri kümeleriyle çalışırken daha az bellek gerektirmesi bir avantaj sağlamasına rağmen, ağırlık güncellemeleri çok gürültülü olabilmekte ve bu da daha dengesiz bir öğrenme sürecine yol açmaktadır.

- Mini Yığın Gradyan İniş (Mini-Batch Gradient Descent)

Eğitim verisi, daha küçük yığınlara (mini-batch) bölünür ve her yığın için modelin ağırlıkları güncellenir. Örneğin, yığın boyutu 32, 64 veya 128 olabilir. Stokastik ve tam yığın iniş yöntemlerinin avantajlarını birleştirir; daha hızlıdır ve daha dengeli ağırlık güncellemeleri sağlar ancak, en uygun yığın boyutunu bulmak zor olabilir ve belirli bir hesaplama gücü gerektirir.

Yığın Boyutunun Model Eğitimine Etkileri:

- Hız ve Bellek Kullanımı

Daha küçük yığın boyutları, her iterasyonda daha az veri işlediği için bellek kullanımını azaltır ve modelin daha hızlı eğitim almasını sağlar. Ancak çok küçük yığınlar, güncellemelerin gürültülü olmasına yol açabilir. Daha büyük yığın boyutları, ağırlık güncellemelerinin daha dengeli olmasını sağlar ancak daha fazla bellek gerektirir ve eğitim süresini uzatabilir.

- Ağırlık Güncellemeleri ve Öğrenme Dinamiği

Küçük yığın boyutları, modelin parametrelerini daha sık günceller ve daha hızlı bir öğrenme dinamiği sağlar. Ancak bu, potansiyel olarak daha gürültülü ve dengesiz güncellemelere neden olabilir. Büyük yığın boyutları ise daha az sıklıkla güncellenir, bu da daha kararlı ve dengeli bir öğrenme süreci sağlar.

- Genelleme Yeteneği

Araştırmalar, küçük yığın boyutlarının daha iyi genelleme performansı sağlayabileceğini göstermektedir. Bunun nedeni, gürültülü güncellemelerin modelin daha genel öznitelikler öğrenmesine yardımcı olmasıdır. Ancak çok küçük yığın boyutları, modelin aşırı öğrenme (overfitting) yapmasına neden olabilir.

Optimal Yığın Boyutu Seçimi:

Optimal yığın boyutunu seçmek, deneme-yanılma sürecini gerektirebilir. Bu seçim, eğitim veri kümesinin büyüklüğüne, modelin karmaşıklığına ve mevcut donanımın kapasitesine bağlı olarak değişmektedir. Optimal yığın boyutunu seçmek için belirtilen aşağıda verilen işlemler yapılabilir:

- Donanım Kapasitesi

Mevcut GPU veya CPU'nun bellek kapasitesi göz önünde bulundurulmalıdır. Daha büyük yığın boyutları daha fazla bellek gerektirir.

- Veri Kümesi Büyüklüğü:

Çok büyük veri kümeleri için mini-batch yöntemleri genellikle daha uygun olur.

- Model Karmaşıklığı

Daha karmaşık modeller, genellikle daha büyük yığın boyutları gerektirebilir.

Yığın Boyutu Ayarlamalarının Etkileri:

- Öğrenme Oranı ile İlişkisi

Yığın boyutunun değişmesi, öğrenme oranının da ayarlanmasını gerektirebilir. Örneğin, daha büyük yığın boyutları genellikle daha küçük öğrenme oranları ile birlikte kullanılır.

- Eğitim Süresi

Daha büyük yığın boyutları, her iterasyonda daha fazla veri işlediği için eğitim süresini uzatabilir. Ancak, her epokta daha az güncelleme yapılması nedeniyle toplam eğitim süresi daha kısa olabilir.

- Paralel İşleme ve Verimlilik

Modern GPU'lar, paralel işleme yetenekleri sayesinde daha büyük yığın boyutları ile daha verimli çalışabilir. Ancak bu, donanımın sınırlarına bağlıdır.

Eđitim Süreci:

Eđitim süreci boyunca yığın boyutunu ayarlamak, modelin performansını optimize etmek için önemlidir. Genellikle aşağıdaki stratejiler kullanılır:

- Dinamik Yığın Boyutu

Eđitim sürecinin başında küçük yığın boyutları kullanılarak modelin hızlı öğrenmesi sağlanabilir ve sonrasında yığın boyutu artırılarak daha kararlı bir öğrenme sağlanabilir.

- Öğrenme Oranı Ayarlanması

Yığın boyutunun yanı sıra öğrenme oranını da dinamik olarak ayarlamak, modelin daha iyi performans göstermesine yardımcı olabilir.

3. MATERYAL VE YÖNTEM

Bu bölümde tez çalışmasında kullanılan süper çözünürlük algoritmaları, materyal olarak kullanılan veri kümesi içeriği, algoritmaların çalıştığı bilgisayar konfigürasyonu, uygulama detayları ve kullanılan performans metriklerinden bahsedilecek olup, izlenen yönteme dair bilgiler verilecektir.

3.1 Süper Çözünürlük Algoritmaları

Bu başlık altında tez çalışmasında kullanılan ve kaynak kodları çalıştırılan süper çözünürlük algoritmaları ve mimarilerine kronolojik sırayla yer verilecektir.

3.1.1 Bikübik (bicubic) interpolasyon

Bikübik interpolasyon, bir görüntüdeki eksik veya bilinmeyen pikselleri tahmin etmek için kullanılan bir yöntemdir. Bikübik interpolasyon ile 4x4 boyutundaki 16 piksellik alanlar üzerinden hesaplama gerçekleştirilir. Görüntü parçalarının 4 köşesi dikkate alınarak ve 3. Derece bir polinom kullanılarak interpolasyon yapılır. Her bir köşedeki yatay, dikey ve çapraz türevler hesaplanarak işlem yapılır. İnterpolasyon uygulanan alan $f(x, y)$ 3. Derece bir polinom olup eşitlik (3.1) ile verilmiştir.

$$f(x, y) = \sum_{i=0}^3 \sum_{j=0}^3 a_{ij} x^i y^j \quad (3.1)$$

f_x , f fonksiyonu için x kısmi türevi, f_y , f fonksiyonu için y kısmi türevi ve f_{xy} , f fonksiyonu için x ve y cinsinden kısmi türevi olarak simgelenmekte olup, f fonksiyonu değerleri ve türevleri bir birim kare için bilindiğinde, a_{ij} katsayılarını veren eşitlik (3.2) ve (3.3) ile verilmiştir.

$$A = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -3 & 3 & -2 & -1 \\ 2 & -2 & 1 & 1 \end{bmatrix} \quad (3.2)$$

$$\begin{bmatrix} a_{00} & a_{01} & a_{02} & a_{03} \\ a_{10} & a_{11} & a_{12} & a_{13} \\ a_{20} & a_{21} & a_{22} & a_{23} \\ a_{30} & a_{31} & a_{32} & a_{33} \end{bmatrix} = A \begin{bmatrix} f(0,0) & f(0,1) & f_y(0,0) & f_y(0,1) \\ f(1,0) & f(1,1) & f_y(1,0) & f_y(1,1) \\ f_x(0,0) & f_x(0,1) & f_{xy}(0,0) & f_{xy}(0,1) \\ f_x(1,0) & f_x(1,1) & f_{xy}(1,0) & f_{xy}(1,1) \end{bmatrix} A^T \quad (3.3)$$

3.1.2 SRCNN (super resolution convolutional neural network)

SRCNN algoritması görüntüye bir ön-işleme metodu olarak yalnızca biküçük interpolasyon uygular ve görüntüyü arzu edilen boyuta getirir. İnterpolasyon uygulanmış görüntü Y , gerçek yüksek çözünürlüklü görüntü X ve görüntü haritalamasını gerçekleştirecek olan denklem F sembolü ile gösterilmektedir.

F denklemi 3 operasyondan oluşmaktadır.

- Parça çıkarma ve gösterim: Bu operasyonda Y görüntüsünden çakışan parçalar çıkarılır ve çıkarılan her bir parça yüksek boyutlu bir vektör haline getirilir.
- Doğrusal olmayan haritalama: Bu operasyonda her bir yüksek boyutlu vektör doğrusal olmayan haritalama ile başka bir yüksek boyutlu vektöre haritalanır. Her bir haritalanmış vektör kavramsal anlamda yüksek çözünürlüklü bir görüntü parçasını temsil eder.
- Yeniden İnşa: Bu operasyon yüksek çözünürlüklü görüntü parçalarını toplar ve nihai yüksek çözünürlüklü görüntüyü elde eder. Elde edilen görüntünün X görüntüsüne yaklaşması beklenir.

Görüntüden çıkartılan parçaların önceden eğitilmiş temeller kümesi olarak gösterilmesi görüntü onarma işleminde popüler bir yöntemdir. Bu yöntem, görüntü üzerinde bir filtre kümesi ile evrişim işlemine eşdeğerdir. Bu işleme karşılık gelen ve ilk katmanı oluşturan F_1 operasyonu eşitlik (3.4) ile ifade edilebilir.

$$F_1(Y) = \max(0, W_1 * Y + B_1) \quad (3.4)$$

W_1 filtreyi, B_1 taraflılığı temsil etmektedir. '*' işareti ise evrişim operasyonunu göstermektedir. W_1 filtresi, $c \times f_1 \times f_1$ ebadında n_1 adet filtreyi içermektedir. c görüntüdeki kanal sayısını gösterirken, f_1 ise uzamsal boyutu göstermektedir. F_1 operasyonu çıktısı n_1 adet öznetelik haritasından oluşmaktadır. Çıktıya RELU(Rectified Linear Unit) aktivasyon fonksiyonu uygulanır.

Doğrusal olmayan haritalama katmanında n_1 boyutlu vektörler n_2 boyutlu vektörlere çevrilir. Bu işlem n_2 adet 1×1 boyutlu filtre ile işlenmesine denk gelmektedir. Bu işleme karşılık gelen ve ikinci katmanı oluşturan F_2 operasyonu eşitlik (3.5) ile ifade edilebilir.

$$F_2(Y) = \max(0, W_2 * F_1(Y) + B_2) \quad (3.5)$$

W_2 filtreyi, B_2 taraflılığı temsil etmektedir. W_2 filtresi, $n_1 \times f_2 \times f_2$ ebadında n_2 adet filtreyi içermektedir. Her bir n_2 boyutlu vektör kavramsal anlamda yeniden inşa için kullanılacak olan yüksek çözünürlüklü görüntü parçasını temsil etmektedir.

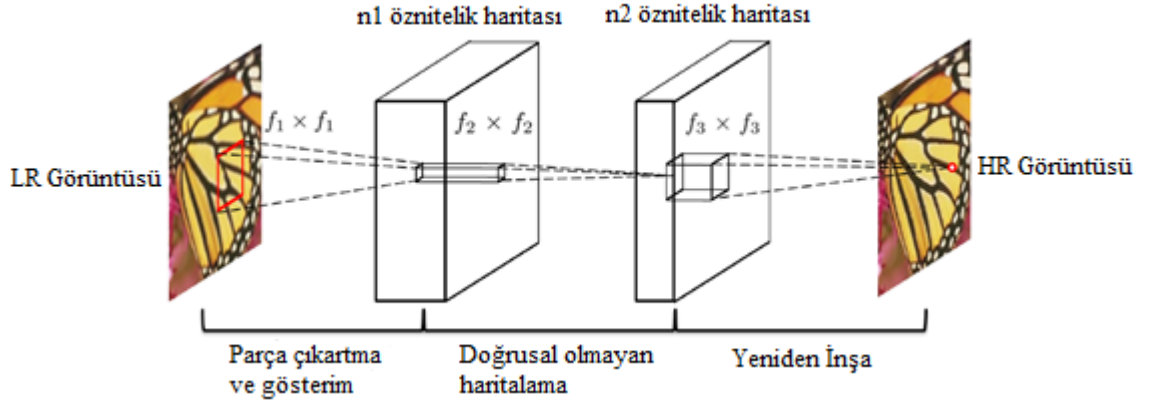
Algoritmaya daha fazla katman eklenerek doğrusallık (linearity) düşürülebilir ancak bu modelin karmaşıklığını artırıcı etki yapmaktadır.

Yeniden inşa katmanında nihai yüksek çözünürlüklü görüntü elde edilir. Bu işleme karşılık gelen ve üçüncü katmanı oluşturan F_3 operasyonu eşitlik (3.6) ile ifade edilebilir.

$$F(Y) = W_3 * F_2(Y) + B_3 \quad (3.6)$$

W_3 filtreyi, B_3 taraflılığı temsil etmektedir. W_3 filtresi, $n_2 \times f_3 \times f_3$ ebadında c adet filtreyi içermektedir. W_3 filtresi bir doğrusal filtre kümesidir.

3 katman bir araya getirilerek evrişimsel sinir ağı oluşturulur. Ağırlık ve yanlılık değerleri algoritma eğitilerek optimize edilir. SRCNN ağ yapısı şekil 3.1 ile verilmiştir.



Şekil 3.1 SRCNN ağ yapısı

3.1.3 VDSR (very deep convolutional networks)

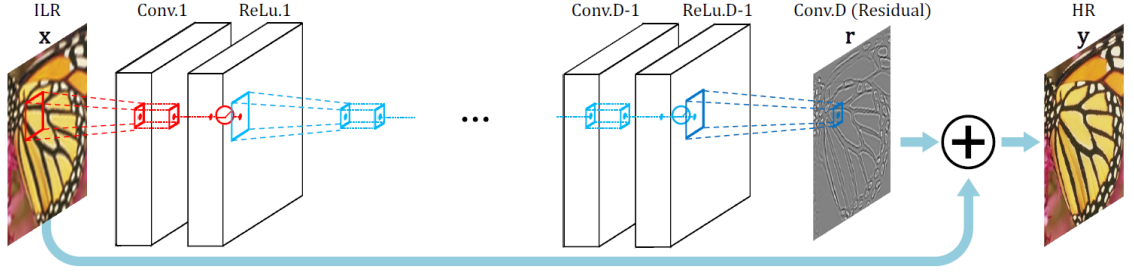
Kim vd. (2016) tarafından geliştirilen VDSR algoritması SRCNN algoritmasında mevcut olan 3 kısıtı ortadan kaldırmak ve daha iyi bir sonuç almak için hazırlanmıştır. Bu kısıtlar: küçük görüntü parçaları içeriğinin eksikliği, eğitimin yakınsama süresinin uzunluğu ve sinir ağının yalnızca bir ölçek faktörü ile çalışabilmesidir.

VDSR algoritmasında büyük görüntü parçaları kullanılarak bilgi çıkarılmakta, eğitim süresi kısaltılmakta ve birden çok ölçek faktörü kullanarak süper çözünürlük işlemini gerçekleştirilebilmektedir.

VDSR sinir ağında d adet katman bulunmaktadır. Birinci ve sonuncu katman haricindeki katmanlar $3 \times 3 \times 64$ boyutunda 64 filtreden oluşmaktadır. Birinci katman girdi görüntüsünü alırken, yeniden inşa katmanı olan son katman $3 \times 3 \times 64$ boyutunda tek filtreden oluşmaktadır.

SRCNN ağında olduğu gibi, VDSR ağı da interpolasyon işlemi uygulanmış düşük çözünürlüklü görüntüyü girdi olarak almaktadır.

VDSR ağı çok fazla katman içerdiğinden ve her katmanda evrişim işlemi uygulandığından, katman çıktılarının boyutu küçülmektedir. Bu durumu engellemek için evrişim işleminden önce görüntülerin kenarlarına sıfır doldurma (zero padding) işlemi uygulanmaktadır. VDSR ağ yapısı şekil 3.2 ile verilmiştir.



Şekil 3.2 VDSR ağ yapısı

3.1.4 ESPCN (efficient sub-pixel convolutional neural network)

Shi vd. (2016) daha önce sunulan süper çözünürlük tekniklerinin hesaplama ve hafıza açısından maliyetli olduğu ortaya koyup, daha etkili bir ağ olarak ESPCN sinir ağını sunmuşlardır. Özellikle girdi görüntüsünün ilk katmanda çözünürlüğünün yükseltilmesi hesaplama yükünü artırmaktadır. Shi vd. (2016) ayrıca bikübik interpolasyonun yeniden inşa problemine ek bir bilgi getirmediğini öne sürmüştür. ESPCN sinir ağında, hesaplama yükünün düşük olması için yukarı örnekleme işlemi ağın son katmanında gerçekleştirilmektedir. Bu yöntem daha önce yapılan çalışmalara göre tekniğin daha hızlı olmasını sağlamıştır.

L katmandan oluşan bir ağ için, $L - 1$ katmanı eşitlik (3.7) ve (3.8) ile ifade edilebilir.

$$f^1(\mathbf{I}^{LR}; W_1, b_1) = \phi(W_1 * \mathbf{I}^{LR} + b_1) \quad (3.7)$$

$$f^l(\mathbf{I}^{LR}; W_{1:l}, b_{1:l}) = \phi(W_l * f^{l-1}(\mathbf{I}^{LR}) + b_l) \quad (3.8)$$

W_l ağ ağırlıklarını, b_l yanlılık değerlerini temsil etmektedir. $W_l \ n_{l-1} \times n_l \times k_l \times k_l$ boyutlu evrişim tensörünü, n_l l katmanındaki öznelik sayısını, k_l ise l katmanındaki filtre büyüklüğünü göstermektedir. ϕ doğrusalsızlık (nonlinearity) fonksiyonunu göstermekte ve eleman başına (element-wise) hesaplama yapmaktadır. Son katman f^L düşük çözünürlüklü görüntüyü yüksek çözünürlüklü görüntüye (\mathbf{I}^{SR}) dönüştürmektedir.

Düşük çözünürlüklü görüntünün yüksek çözünürlüklü görüntüye dönüştürülmesi ters evrişim ile gerçekleştirilmekte ve eşitlik (3.9) ile ifade edilmektedir.

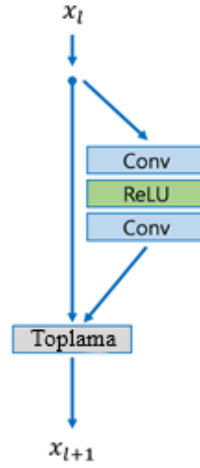
$$\mathbf{I}^{SR} = f^L(\mathbf{I}^{LR}) = PS(W_L * f^{L-1}(\mathbf{I}^{LR}) + b_L) \quad (3.9)$$

PS periyodik karıştırma operatörü olup, $H \times W \times C \cdot r^2$ tensörünü $rH \times rW \times C$ şekilli tensöre çevirmektedir.

3.1.5 EDSR (enhanced deep residual networks)

Lim vd. (2017) daha önce sunulan süper çözünürlük tekniklerinde ağ modellerinin yeniden inşa performansının mimarideki ufak değişikliklere karşı çok hassas olduğunu, farklı ölçekteki yukarı örnekleme problemlerinin bağımsız olarak ele alınması yaklaşımının doğru olmadığını öne sürmüştür. Bu problemleri gidermek için, gereksiz modüllerin ağ mimarisinden çıkarılması ve farklı ölçekteki görüntüler ile eğitim işleminin yapılması üzerinde çalışılmıştır.

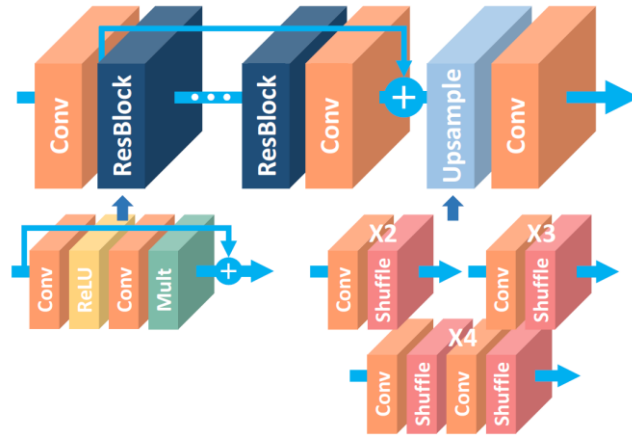
EDSR sinir ağında kalan değerli (residual) bloklar kullanılmış olup, Ledig vd. (2017) tarafından sunulan SRResNet çalışması üzerinde basit değişiklikler yapılarak sunulmuştur. Yapılan bu değişiklik mimarinin performansını artırıcı etki yapmıştır. Bu etkiye ek olarak, GPU hafıza kullanımını yaklaşık %40 düşmüştür. Lim vd. (2017) tarafından önerilen kalan değerli blok yapısı şekil 3.3 ile verilmiştir.



Şekil 3.3 Önerilen kalan değerli ağ bloğu

EDSR mimarisinde 0.1 faktörü ile kalan değerli ölçekleme uygulanmıştır. Her bir kalan değerli blokta son evrişim katmanından hemen sonra sabit ölçekleme katmanları konulmuştur. Eklenen bu katmanlar, özellikle yüksek sayıda filtre kullanıldığında eğitim yöntemini dengelemektedir.

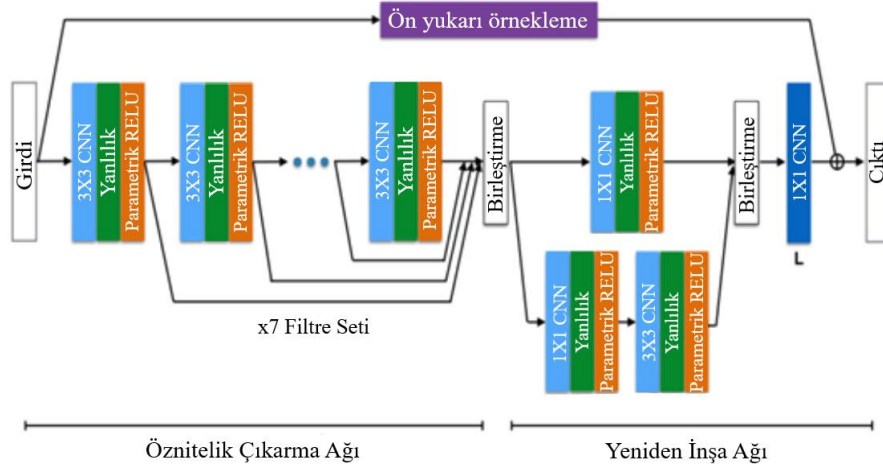
Her bir evrişim katmanı için 64 öznitelik haritası bulunmaktadır. Katman sayısı 32, öznitelik kanal sayısı 256 ve ölçekleme faktörü 0.1 olarak ayarlanmıştır. Model mimarisi şekil 3.4 ile verilmiştir.



Şekil 3.4 Önerilen tek ölçekli süper çözünürlük mimarisi

3.1.6 DCSCN (deep convolutional neural network with skip connection and network in network)

Yamanaka vd. (2017) süper çözünürlük çalışmalarında kullanılan derin ve tam bağlantılı evrişim ağlarının yüksek hesaplama gücü ve uzun işleme süresine ihtiyaç durduğunu belirtmiştir. Bu nedenle, ağ mimarilerinin daha uygun hale getirilip hafifletilmesi önerilmiştir. Genellikle 20-30 katmandan oluşan süper çözünürlük mimarilerine karşılık yalnızca 11 katmandan oluşan DCSCN mimarisi önerilmiştir. Önerilen mimari literatürdeki mimarilere göre 10 ile 100 kat arasında daha az bir hesaplama gücüne ihtiyaç duymaktadır. Önerilen DCSCN mimarisi şekil 3.5 ile verilmiştir.



Şekil 3.5 Önerilen DCSCN mimarisi

Öznitelik çıkarma (feature extraction) bloğunda düşük çözünürlüklü görüntü olduğu gibi girdi olarak kullanılmakta ve her bir katman çıktısı toplanarak yeniden inşa (reconstruction) ağına verilmektedir. Bu ağıdaki her bir katman çıktısının birleştirme katmanına aktarılması atlama bağlantısı (skip connection) olarak tanımlanmaktadır. Öznitelik çıkarma ağı 3x3 boyutunda 7 kademeli katmandan oluşmakta ve her bir katmanda yanılılık ve parametrik RELU birimleri bulunmaktadır. Yeniden inşa ağına ise paralel katmanların bulunduğu bir yapı tercih edilmiştir. Paralel 2 adet 1x1 boyutunda evrişimsel sinir ağı içeren katmanlar literatürde ağ içinde ağ (network in network) yapısı olarak tanımlanmaktadır. Bu yapı yalnızca önceki katmanın ebatlarını düşürmekle

kalmamış aynı zamanda düşük bir bilgi kaybıyla yüksek bir hesaplama yapabilme kabiliyeti kazandırmıştır. Ayrıca yeniden inşa ağındaki bu yapı daha fazla doğrusalsızlık kazandırmıştır. Doğrusalsızlık mimarinin eğitimini iyileştirerek daha iyi sonuçlar alınmasını sağlamaktadır. Son olarak, yeniden inşa ağı çıktısı ile bikübik interpolasyon uygulanan görüntü birbirine eklenerek nihai yüksek çözünürlüklü görüntü elde edilmektedir.

Aşırı öğrenmeyi (overfitting) önlemek için sönümlenme (dropout) hiper-parametresi kullanılmış olup, bu değer 0.8 olarak belirlenmiştir. Bu durum her bir eğitim yinelemesinde nöronların %20'sinin rasgele olarak sönümleneceği ve hesaplamalara dahil edilmeyeceği anlamına gelmektedir.

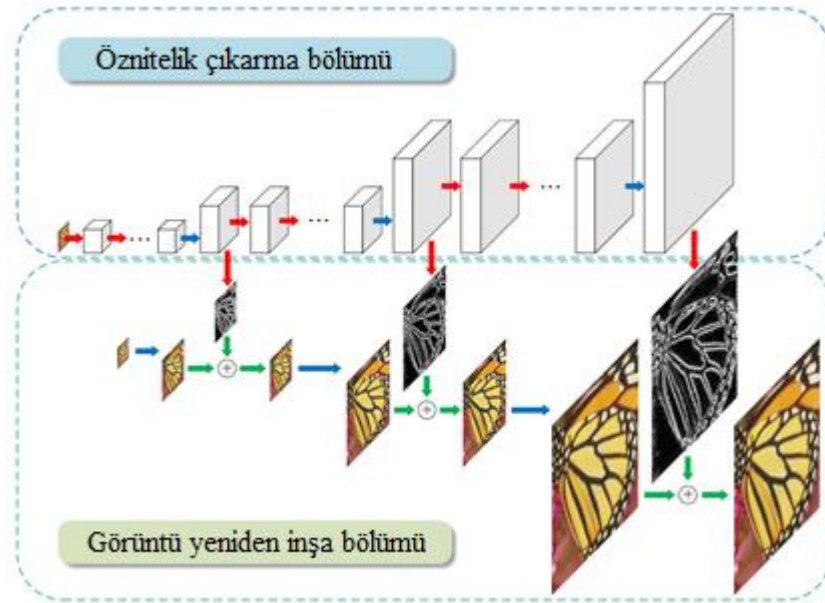
3.1.7 LapSRN (deep laplacian pyramid networks)

Lai vd. (2017) LapSRN mimarisini sunarak daha önce yapılan çalışmaların eksikliklerini ve bu eksikliklere karşın nasıl bir metot izlendiğini aktarmıştır. Birinci konu, girdi görüntüsü üzerinde uygulanan ön-yukarı örnekleme işleminin gereksiz bir hesaplama yükü oluşturduğu ve sıklıkla görünür yeniden inşa bozuklukları meydana getirdiğidir. Ters evrişim (transposed convolution) işleminin görece küçük ağlar kullanması ve karmaşık haritalamayı ağ kapasitesinden dolayı yapamamasıdır. İkinci konu, literatürde mevcut mimarilerin l_2 kayıp fonksiyonunu kullanması ve genellikle bulanık tahminler yapmasıdır. Oluşturan görüntüler bu sebeple aşırı-yumuşatılmış (overly-smooth) olmakta ve insan görsel algısına uzak kalmaktadır. LapSRN mimarisinde Charbonnier kayıp fonksiyonu kullanılmıştır. Üçüncü konu ise yüksek çözünürlüklü görüntülerin istenen ölçek için tek bir adımda gerçekleştirilmesidir. LapSRN mimarisi, görüntülerin farklı ölçekler için kademeli olarak büyütülmesine olanak sağlamaktadır. Tüm bu konulara getirilen çözümler LapSRN mimarisini daha isabetli ve hızlı yapmaktadır. Ayrıca, aşamalı yeniden inşa kabiliyeti sayesinde farklı ölçekteki görüntülerin sunulması mümkün kılınmaktadır.

LapSRN mimarisi düşük çözünürlüklü (low resolution (LR)) görüntüsünü girdi olarak almakta ve S ölçek değeri için $\log_2 S$ adet seviye içermektedir. Örneğin ölçek değeri 8

için 3 alt ağ bulunmaktadır. s seviyesinde, öznetelik çıkarma katmanı d evrişim katmanından oluşmakta ve ters evrişim katmanı ile çıkarılan öznetelikler 2 değeri ile ölçeklenmektedir. Bu işlem, işlem yükünü düşürmek için her bir katmanda yalnız bir kez gerçekleştirilmektedir.

Yeniden inşa bölümünde (branch) ise görüntüler biküçük interpolasyon ile büyütülmekte ve öznetelik çıkarma bölümünden elde edilen yüksek çözünürlüklü (high resolution (HR)) ile toplanmaktadır. Böylece, tüm ağ her bir seviyede benzer mimari ile kademeli olarak oluşur. Mimaride her bir katman 3×3 boyutunda 64 filtreden oluşmaktadır. Ters evrişim filtre boyutu ise 4×4 olarak ayarlanmıştır. Önerilen LapSRN mimarisi şekil 3.6 ile verilmiştir.

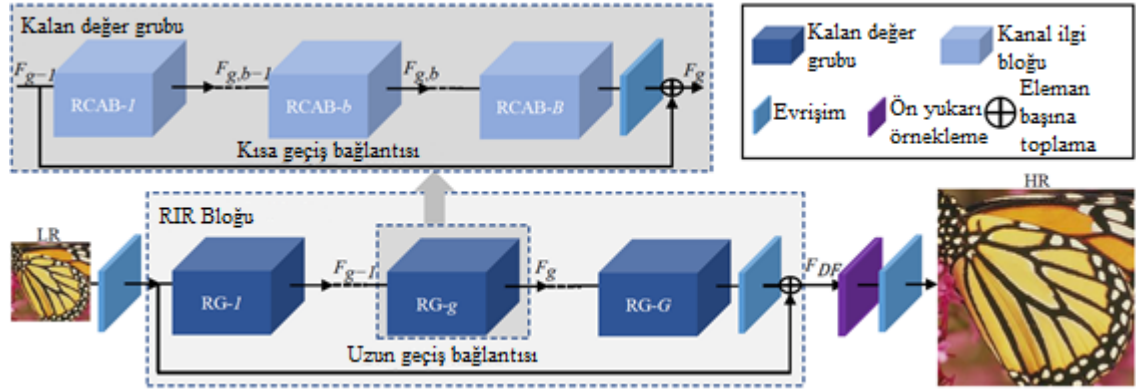


Şekil 3.6 Önerilen LapSRN mimarisi

3.1.8 RCAN (residual channel attention networks)

Zhang vd. (2018) önceki süper çözünürlük çözümlerine 3 katkıda bulunarak bir mimari sunmuştur. Önceki yöntemlerden daha derin bir mimari tasarlanarak daha iyi bir performans sağlanmıştır. Kalan değer içinde kalan değer (residual in residual (RIR))

yapısı ile çok daha derin bir ağ kurulabilmiştir. Kanal ilgi (channel attention) mekanizması ile öznitelik kanalları arasındaki farklar önemsiz olarak uyarlanabilir bir yaklaşım sunulmuştur. Önerilen RCAN mimarisi şekil 3.7 ile verilmiştir.



Şekil 3.7 Önerilen RCAN mimarisi

RCAN mimarisi temelde 4 kısımdan oluşmaktadır: sığ öznitelik çıkarma, RIR derin öznitelik çıkarma, yukarı örnekleme ve yeniden inşa.

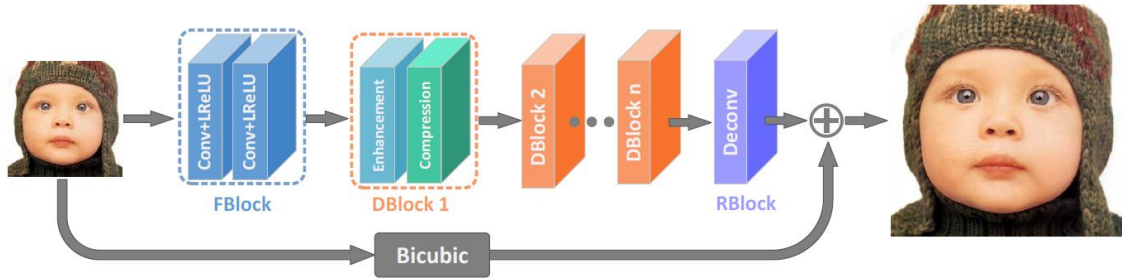
Sığ öznitelik çıkarma kısmında yalnızca 1 evrişim katmanı bulunur. Elde edilen öznitelikler G adet kalan değerli grup barındıran RIR derin öznitelik çıkarma bloğuna aktarılır. Elde edilen derin öznitelikler yukarı örnekleme katmanına aktarılır. Yukarı örnekleme yapılan görüntü yeniden inşa bloğuna aktarılır ve nihai HR görüntüsü elde edilir.

Önerilen RCAN mimarisinde $G = 10$ olarak ayarlanmış olup, kanal aşağı ve yukarı örnekleme katmanları hariç tüm katmanlar 3×3 boyutunda filtre içermektedir. Öznitelik çıkarma blokları 64 filtreden oluşmaktadır. Görüntü ebadını korumak için sıfır-doldurma metodu uygulanmıştır.

3.1.9 IDN (information distillation network)

Hui vd. (2018) literatürdeki süper çözünürlük mimarilerinin eksiklikleri olduğunu belirtmiş ve hesaplama yükü bakımından daha etkili bir mimari sunmuştur. Daha derin ya da geniş bir ağ tasarımı her ne kadar popüler olsa da hesaplama maliyeti ve hafıza kullanımının yüksek oluşu gerçek zamanlı sistemlerde kullanılabilirliği düşürmektedir.

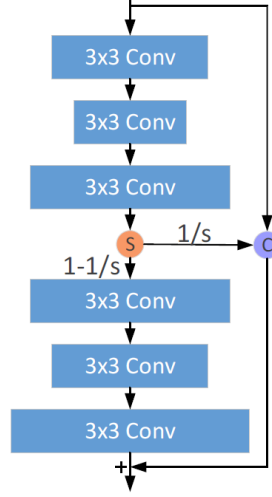
IDN mimarisinin ilk bloğu LR görüntüden öznitelik çıkarma bloğudur. Ardından gelen çoklu bilgi damıtma (information distillation) bloğu kalan değer bilgisini kademeli olarak biriktirmektedir. Son blokta ise yeniden inşa yapılarak HR görüntüsü elde edilmektedir. Önerilen IDN mimarisi şekil 3.8 ile verilmiştir.



Şekil 3.8 Önerilen IDN mimarisi

IDN mimarisinin kilit bileşeni bilgi damıtma bloğu olup, bu blok iyileştirme (enhancement) ve sıkıştırma biriminden oluşmaktadır. Şekil 3.9 ile verilen iyileştirme bloğu iki sığ evrişim ağından oluşmaktadır. İlk modülün öznitelik haritaları kısa bir yol (3 katmanlı) ile çıkarılmaktadır. Bu nedenle, yerel kısa yol öznitelikleri olarak kabul edilebilir. Derin ağların daha fazla ifade gücüne sahip olduğunu göz önünde bulundurarak, yerel kısa yol özniteliklerinin bir kısmını sıradaki modüle gönderilir. Bu şekilde, ikinci modülün öznitelik haritaları doğal olarak yerel uzun yol öznitelikleri haline gelir. Öznitelik haritalarını iki kısma ayırılır: İlk kısım, ayrılmış kısa yol özniteliklerini, ikinci kısım iyileştirilecek kısa yol özniteliklerini ifade eder. Uzun ve kısa yol öznitelik haritalarını elde ettikten sonra, daha zengin ve verimli bilgi elde etmek için bu iki tür

öznitelik birleştirilir. Sıkıştırma biriminde ise iyileştirme birimindeki özniteliklerin gereksiz bilgilerini sıkıştırmak için basit bir evrişim katmanı kullanılır.



Şekil 3.9 IDN iyileştirme bloğu

İyileştirme biriminin eşitliği (3.10) ile verilmiştir. P^k iyileştirme birim çıktısını, P_2^k yerel uzun-yol öznitelikleri, R^k kısa-yol özniteliklerini, C_b yığın (stacked) evrişim operasyonlarını, S birleştirme operasyonunu, C_a zincirlenmiş evrişim operasyonlarını, (B_{k-1}) k sırasındaki girdiyi, s bölme değerini, C birleştirme operasyonunu temsil etmektedir.

$$P^k = P_2^k + R^k = C_b(S(C_a(B_{k-1}), 1 - 1/s)) + C(S(C_a(B_{k-1}), 1/s), B_{k-1}) \quad (3.10)$$

Sıkıştırma birimi ise 1×1 boyutlu evrişim katmanı kullanmaktadır. Bu katman ile sonraki katman için ilişkili bilgiler çıkartılır. Sıkıştırma birimi eşitlik (3.11) ile ifade edilebilir. f_F^k evrişim katmanı fonksiyonunu, α_F^k aktivasyon fonksiyonunu ve W_F^k ağırlık parametrelerini ifade etmektedir.

$$B_k = f_F^k(P^k) = \alpha_F^k(W_F^k(P^k)) \quad (3.11)$$

3.2 Veri Kümesi

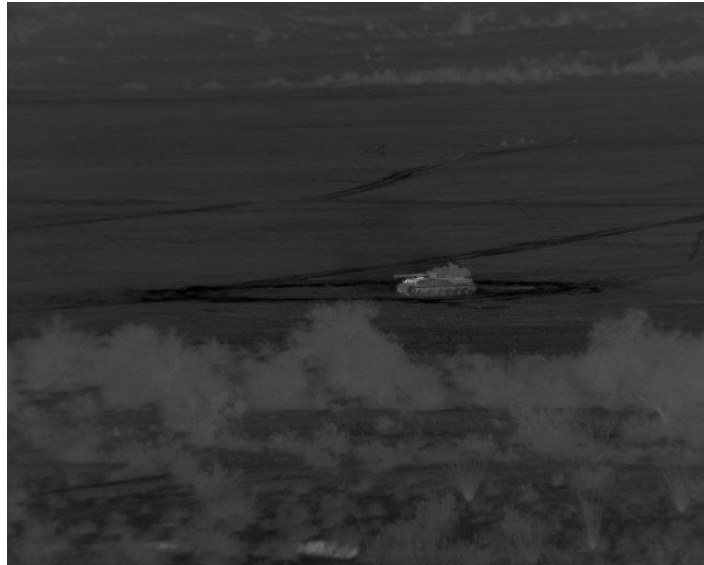
Birleşmiş Devletler Ordusu (US Army) ve Elektronik Sensör Direktörlüğü (NVESD) tarafından toplanan algoritma geliştirme görüntü veri kümesi, süper çözünürlük algoritmaları için eğitim ve test veri kümeleri olarak kullanılmıştır (Anonymous 2024).

Veri kümesi, insan, sivil/askeri araç ve doğa görüntülerinden oluşmaktadır. Farklı kayıtlardan alınan tekli görüntüler ile 205 adet görüntü seçilmiş olup, 164 görüntü (%80) eğitim, 41 görüntü (%20) test için ayrılmıştır.

MWIR görüntüleri alınan kamera, $3.5 - 5.0\mu m$ bandında, İndiyum-antimon (Indium-antimonide) odak düzlemine (focal plane), $28\mu m$ sensör mesafesine, $300mm$ odak uzunluğuna, 640×512 çözünürlüğe ve $3.4^\circ \times 2.6^\circ$ görüş alanına sahiptir.

Görüntüler 640×512 çözünürlüğünde, tek kanal (channel), normalizasyon uygulanmış ve 8 bit derinliğindedir.

Örnek veri kümesi görüntüsü şekil 3.10 ile verilmiştir.



Şekil 3.10 Örnek veri kümesi görüntüsü (askeri araç)

3.3 Bilgisayar Konfigürasyonu

Süper çözünürlük algoritmalarının eğitim ve test aşamaları masaüstü bir bilgisayarda gerçekleştirilmiştir. Bilgisayar özellikleri:

- Intel i7-10700 2.90 Ghz (16 CPU) işlemci
- 32 GB RAM
- NVIDIA GeForce RTX 2060 Super ekran kartı
- Windows 10 İşletim Sistemi

Algoritmalar, Anaconda paket ve ortam yönetim aracıyla, Python 3.7 sürümü ile tensorflow ve pytorch derin öğrenme kütüphaneleri kullanılarak çalıştırılmıştır.

ESPCN ve EDSR algoritmaları pytorch yapısı (framework) ile çalıştırılmış olup, diğer algoritmalar tensorflow yapısı ile çalıştırılmıştır.

3.4 Performans Metrikleri

Derin öğrenme tabanlı SR algoritmalarının karşılaştırılması için kullanılacak olan performans metrikleri MSE, PSNR ve SSIM olup, detayları alt başlıklarda verilmiştir.

3.4.1 MSE (mean square error)

Ortalama Kare Hata (MSE), iki görüntü için karşılık gelen piksel değer farkları karesinin ortalamasını veren bir performans metriğidir. Görüntü işleme algoritmalarında yaygın olarak kullanılmaktadır.

I_{HR} SR algoritması tarafından yukarı örneklenen görüntü, I_{GT} görüntüsü gerçek HR görüntüsü olarak temsil edildiğinde, aşağıdaki eşitlikler ile MSE değeri hesaplanır.

Piksel pozisyonu (i, j) iken görüntü farkını hesaplayan eşitlik (3.12) ile verilmiştir.

$$\Delta(i, j) = I_{HR}(i, j) - I_{GT}(i, j) \quad (3.12)$$

Elde edilen farkların karesini alan eşitlik (3.13) ile verilmiştir.

$$\Delta^2(i, j) = (I_{HR}(i, j) - I_{GT}(i, j))^2 \quad (3.13)$$

MXN boyutundaki görüntüler için kare farklarının toplamını hesap eden eşitlik (3.14) ile verilmiştir.

$$\sum_{i=0}^{M-1} \sum_{j=0}^{N-1} (I_{HR}(i, j) - I_{GT}(i, j))^2 \quad (3.14)$$

Kare farklarının ortalamasını (MSE) veren eşitlik (3.15) ile verilmiştir.

$$MSE = \frac{1}{MN} \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} (I_{HR}(i, j) - I_{GT}(i, j))^2 \quad (3.15)$$

3.4.2 PSNR (peak signal-to-noise ratio)

PSNR, oluşturulan görüntünün gerçek görüntü ile karşılaştırmasında kullanılan bir kalite ölçüğü olup birimi desibeldir. n bit bir görüntüde, PSNR değeri verilen eşitlik (3.16) ile hesaplanır. 8-bit bir görüntüdeki azami piksel değeri 255'tir.

$$PSNR = 10 \cdot \log_{10} \frac{(2^n - 1)^2}{MSE} \quad (3.16)$$

Yüksek PSNR oluşturulan görüntünün iyi kalitede olduğunu gösterirken, düşük PSNR yüksek bozukluk ya da gürültülü görüntüyü ifade etmektedir.

3.4.3 SSIM (structural similarity index measure)

Yapısal benzerlik gösterge ölçütü (SSIM), iki görüntü arasındaki benzerliği yapısal bilgi, luminans (luminance) ve zıtlık (contrast) faktörlerine göre değerlendiren bir performans metriğidir.

μ_x x görüntüsünün ortalama değerini, σ_x^2 x görüntüsünün varyansını, σ_{xy} x ve y görüntüsünün kovaryansını, C değerleri ise sabitleri göstermektedir. Luminans faktörü eşitlik (3.17), zıtlık faktörü eşitlik (3.18) ve yapı faktörü eşitlik (3.19) ile verilmiştir.

$$l(x, y) = \frac{2\mu_x\mu_y + C_1}{\mu_x^2 + \mu_y^2 + C_1} \quad (3.17)$$

$$c(x, y) = \frac{2\sigma_x\sigma_y + C_2}{\sigma_x^2 + \sigma_y^2 + C_2} \quad (3.18)$$

$$s(x, y) = \frac{\sigma_{xy} + C_2/2}{\sigma_x\sigma_y + C_2/2} \quad (3.19)$$

3 faktörün çarpımı SSIM metriğini vermekte olup, eşitlik (3.20) ile gösterilmiştir.

$$SSIM(x, y) = l(x, y) \cdot c(x, y) \cdot s(x, y) \quad (3.20)$$

SSIM metriği 0 ile 1 arasında değişmekte olup, 1 mükemmel benzerliği, 0 benzerlik olmadığını ifade etmektedir.

3.5 Uygulama Detayları

Veri kümesindeki eğitim görüntüleri biküçük interpolasyon kullanılarak 4 ölçeği ile aşağı örneklenmiş ve mimarilere girdi görüntüleri olarak verilmiştir. Tüm algoritmalar görüntüyü 4 ölçeği ile yukarı örneklenmeye çalışmıştır. Gerçekleştirilen eğitim ve test

denemeleri göz önüne alınarak, hem algoritmaların mevcut bilgisayar konfigürasyonu ile çalışabilmesi hem de eğitim sürelerinin çok yüksek olmaması için yığın boyutu 16, parça boyutu 128 olarak seçilmiştir. Böylece bir epok gerçekleşirken aynı anda 16 görüntü parçası sinir ağına girmekte ve hesaplamalar yapılmaktadır. Yakınsama değerleri ve çalışma süresi gözetilerek epok sayısı 100 olarak belirlenmiştir.

Öğrenme oranı her bir algoritma için literatürdeki değerler göz önüne alınarak belirlenmiş, daha iyi sonuç verip vermediğine bakılarak aşağı/yukarı yönde değiştirilerek ayarlanmıştır. SRCNN algoritması ilk 50 epok için 10^{-4} , sonraki 25 epok için 10^{-5} , son 25 epok için 10^{-6} ; VDSR algoritması 10^{-4} ; ESPCN algoritması 10^{-3} ; LapSRN algoritması 10^{-4} ; DCSCN algoritması ilk 50 epok için 2.10^{-3} , sonraki 25 epok için 2.10^{-4} , son 25 epok için 2.10^{-5} ; EDSR algoritması 10^{-4} ; IDN algoritması 10^{-4} ve RCAN algoritması 10^{-4} öğrenme oranı değerini kullanmıştır.

4. ARAŞTIRMA BULGULARI

Tez kapsamında kullanılan derin öğrenme tabanlı süper çözünürlük algoritmaları farklı hiper-parametre konfigürasyonlarıyla çalıştırılarak eğitim ve test denemeleri gerçekleştirilmiştir.

Çizelge 4.1 SRCNN deneme çizelgesi

Yığın Boyutu / Parça Boyutu / Öğrenme Oranı	Eğitim Süresi (dk.)	MSE	PSNR	SSIM
16 / 64 / 1e-4->1e-6	12	14.409	39.418	0.9414
32 / 64 / 1e-4->1e-6	20	14.414	39.413	0.9412
16 / 128 / 1e-4->1e-6	40	14.324	39.452	0.9414

Çizelge 4.1 ile görüldüğü üzere SRCNN algoritması için farklı yığın ve parça boyutuyla (patch size) denemeler gerçekleştirilmiştir. Parça boyutu değeri, görüntü parçasının bir ekseninde yer alan piksel adedini temsil etmektedir. 3 performans metriği bakımından birbirine yakın sonuçlar (<1%) elde edilmesine karşın, yığın ya da parça boyutunun artırılması eğitim süresini uzatmaktadır.

Çizelge 4.2 VDSR deneme çizelgesi

Yığın Boyutu / Parça Boyutu / Öğrenme Oranı	Eğitim Süresi (dk.)	MSE	PSNR	SSIM
16 / 64 / 1e-4	25	13.644	39.778	0.9437
32 / 64 / 1e-4	50	13.448	39.846	0.9439
64 / 64 / 1e-4	100	13.346	39.829	0.9447
16 / 128 / 1e-4	90	13.151	39.897	0.9446

Çizelge 4.2 ile görüldüğü üzere VDSR algoritması için farklı yığın ve parça boyutuyla denemeler gerçekleştirilmiştir. Yığın ya da parça boyutunun artırılması ile daha iyi sonuçlar alınmasına karşın, eğitim süresi uzamaktadır.

Çizelge 4.3 ESPCN deneme çizelgesi

Yığın Boyutu / Parça Boyutu / Öğrenme Oranı	Eğitim Süresi (dk.)	MSE	PSNR	SSIM
16 / 64 / $1e-2 \rightarrow 1e-4$	40	18.204	37.417	0.9299
16 / 128 / $1e-2 \rightarrow 1e-4$	135	19.822	36.816	0.9272
16 / 32 / $1e-2 \rightarrow 1e-4$	15	19.622	36.841	0.9277
16 / 32 / $1e-3$	15	14.862	39.229	0.9401
16 / 64 / $1e-3$	40	14.900	39.267	0.9402
16 / 128 / $1e-3$	135	14.738	38.978	0.9404

Çizelge 4.3 ile görüldüğü üzere ESPCN algoritması için farklı parça boyutu ve öğrenme oranıyla denemeler gerçekleştirilmiştir. Denemelerde yığın boyutu 16 olarak kullanılmış olup, yüksek parça boyutu için eğitim süresi uzamaktadır. Parça boyutu 64 olarak seçildiğinde en iyi sonuçlar elde edilmektedir. İlk üç denemede öğrenme oranı ilk 50 epok için 10^{-2} , sonraki 25 epok için 10^{-3} ve son 25 epok için 10^{-4} olarak uygulanmış; son üç denemede ise öğrenme oranı 10^{-3} olarak kullanılmıştır. Öğrenme oranı 10^{-3} olarak seçildiğinde 3 performans metriği bakımından da belirgin ölçüde daha iyi sonuçlar elde edilmiştir.

Çizelge 4.4 LapSRN deneme çizelgesi

Yığın Boyutu / Parça Boyutu / Öğrenme Oranı	Eğitim Süresi (dk.)	MSE	PSNR	SSIM
16 / 64 / 1e-4	15	13.916	39.684	0.9428
16 / 128 / 1e-4	20	13.774	39.704	0.9434
64 / 128 / 1e-4	80	13.825	39.699	0.9436
16 / 128 / 1e-5	20	14.707	39.332	0.9301

Çizelge 4.4 ile görüldüğü üzere LapSRN algoritması için farklı yığın boyutu, parça boyutu ve öğrenme oranıyla denemeler gerçekleştirilmiştir. Parça boyutunun artırılması ile daha iyi sonuçlar alınabilmişken, yığın boyutunun artırılması ile çok yakın sonuçlar alınmış, buna ek olarak eğitim süresi belirgin ölçüde artmıştır. Öğrenme oranı 10^{-5} olarak seçildiğinde, 3 performans metriği açısından başarı düşmüştür.

Çizelge 4.5 DCSCN deneme çizelgesi

Yığın Boyutu / Parça Boyutu / Öğrenme Oranı	Eğitim Süresi (dk.)	MSE	PSNR	SSIM
16 / 64 / 1e-4	15	13.295	39.723	0.9443
16 / 128 / 1e-4	20	12.104	40.032	0.9466
16 / 128 / 1e-3	17	10.744	40.341	0.9489
16 / 128 / 1e-5	17	15.023	39.185	0.9379
16 / 128 / 2e-3->2e-5	17	10.515	40.567	0.9503

Çizelge 4.5 ile görüldüğü üzere DCSCN algoritması için farklı parça boyutu ve öğrenme oranıyla denemeler gerçekleştirilmiştir. Yığın boyutunun artırılması ile daha iyi sonuçlar alınabilmiş, eğitim süresi artmıştır. Öğrenme oranı 10^{-3} olarak seçildiğinde daha iyi, 10^{-5} olarak seçildiğinde daha kötü bir sonuç elde edilmiştir. Yamanaka vd. (2017) yaptığı çalışmada öğrenme oranını başlangıçta 2.10^{-3} , 50 epok ardından 2.10^{-4} ve son

25 epok için 2.10^{-5} olarak belirlemiştir. Çalışmadaki öğrenme oranı uygulandığında 3 performans metriği açısından en iyi sonuç elde edilmiştir.

Çizelge 4.6 EDSR deneme çizelgesi

Yığın Boyutu / Parça Boyutu / Öğrenme Oranı	Eğitim Süresi (dk.)	MSE	PSNR	SSIM
1 / 64 / $1e-4$	75	12.378	39.788	0.9469
16 / 128 / $1e-4$	4200	9.6165	40.847	0.9516

Çizelge 4.6 ile görüldüğü üzere EDSR algoritması için farklı yığın ve parça boyutuyla denemeler gerçekleştirilmiştir. Yığın ve parça boyutunun artırılması ile daha iyi sonuçlar elde edilmesine karşın çok yüksek (70 saat) bir eğitim süresi kullanılmıştır.

Çizelge 4.7 RCAN deneme çizelgesi

Yığın Boyutu / Parça Boyutu / Öğrenme Oranı	Eğitim Süresi (dk.)	MSE	PSNR	SSIM
16 / 64 / $1e-4$	20	14.627	39.284	0.9398
16 / 128 / $1e-4$	30	14.459	39.351	0.9407
16 / 128 / $1e-4 \rightarrow 2.5e-5$	30	14.631	39.226	0.9410

Çizelge 4.7 ile görüldüğü üzere RCAN algoritması için farklı parça boyutu ve öğrenme oranıyla denemeler gerçekleştirilmiştir. Daha yüksek parça boyutu (128) ile daha iyi sonuçlar elde edilebilmiş olup, eğitim süresi bu nedenle artmıştır. Zhang vd. (2018) yaptığı çalışmada öğrenme oranını başlangıçta 10^{-4} , 50 epok ardından 5.10^{-5} ve son 25 epok için $2.5.10^{-5}$ olarak belirlemiştir. Çalışmadaki öğrenme oranı uygulandığında 3 performans metriği açısından en iyi sonuç elde edilmesine karşın, artış oranı düşüktür.

Çizelge 4.8 IDN deneme çizelgesi

Yığın Boyutu / Parça Boyutu / Öğrenme Oranı	Eğitim Süresi (dk.)	MSE	PSNR	SSIM
16 / 64 / 1e-4	15	15.449	38.845	0.9360
16 / 128 / 1e-4	20	15.823	38.653	0.9336
64 / 128 / 1e-4	60	15.736	38.755	0.9342

Çizelge 4.8 ile görüldüğü üzere IDN algoritması için farklı yığın ve parça boyutuyla denemeler gerçekleştirilmiştir. Yığın ya da parça boyutunun artırılması daha fazla eğitim süresi gerektirmesine karşın, elde edilen sonuçlar yakındır.

Tez çalışmasında kullanılan algoritmaların nihai sonuçları çizelge 4.9 ile verilmiştir. Değerler test veri kümesindeki her bir görüntü için hesaplanan performans metriklerinin ortalamasını göstermektedir. Kırmızı renkli değerler en iyi, mavi renkli değerler ikinci en iyi değeri göstermektedir. EDSR algoritması MSE, PSNR ve SSIM bakımından en iyi, DCSCN algoritması ikinci en iyi algoritma olacak şekilde sonuç vermiştir. Derin öğrenme tabanlı tüm algoritmalar klasik yöntem olan bikübik interpolasyon karşısında 3 performans metriğine göre de daha başarılı olmuştur.

Çizelge 4.9 Algoritmaların performans metriklerine göre karşılaştırılması

	Bicubic	SRCNN	VDSR	ESPCN	EDSR	DCSCN	LapSRN	RCAN	IDN
MSE	18.539	14.324	13.151	14.738	9.616	10.515	13.774	14.459	15.823
PSNR	38.351	39.452	39.897	38.978	40.847	40.567	39.704	39.351	38.653
SSIM	0.9311	0.9414	0.9446	0.9404	0.9516	0.9503	0.9434	0.9407	0.9336

Tez çalışmasında kullanılan algoritmaların eğitim süreleri çizelge 4.10 ile verilmiştir. Kırmızı renkli değer en iyi, mavi renkli değer ikinci en iyi değeri göstermektedir. DCSCN algoritması eğitim süresi bakımından en düşük, LapSRN ve IDN algoritması ikinci en

düşük eğitim süresine sahiptir. Eğitim süresi düşük olan algoritmalar, tez çalışmasında kullanılan veri kümesinden daha büyük bir veri kümesi ile eğitilmek istendiğinde zaman maliyeti açısından tercih edilebilecek algoritmalarlardır. EDSR algoritması diğer algoritmalara göre çok daha uzun bir sürede (70 saat) eğitimini tamamlamıştır. EDSR algoritması için yığın sayısı 1 olarak verildiğinde diğer algoritmaların değerlerine yakın bir sürede eğitim tamamlanabilirken, aynı yığın sayısı (16) için çok daha uzun bir süreye ihtiyaç duyulmaktadır. Yığın sayısının 16 olduğunda, EDSR algoritması performans metrikleri açısından daha iyi sonuç vermiştir. Karşılaştırmanın daha doğru olması için algoritmalar aynı parça ve yığın boyutu ile çalıştırılmıştır.

Çizelge 4.10 Algoritmaların eğitim süreleri

	SRCNN	VDSR	ESPCN	EDSR	DCSCN	LapSRN	RCAN	IDN
Eğitim Süresi (dk.)	40	90	135	4200	17	20	30	20
Eğitim Süresi (1 görüntü) (dk.)	0.24	0.55	0.82	25.61	0.10	0.12	0.18	0.12
Eğitim Süresi (10000 görüntü) (gün)	1.7	3.81	5.71	177.85	0.72	0.85	1.27	0.85

Tez çalışmasında kullanılan algoritmaların test (çalışma) süreleri çizelge 4.11 ile verilmiştir. Kırmızı renkli değer en iyi, mavi renkli değer ikinci en iyi değeri göstermektedir. ESPCN algoritması en hızlı olup saniyede ortalama 14.71 görüntüyü 4 ölçeği ile yukarı örneklerken, LapSRN algoritması ikinci en hızlı algoritma olup saniyede ortalama 11.9 görüntüyü yukarı örneklemiştir. EDSR algoritması en yavaş algoritma olup saniyede ortalama 2.96 görüntüyü yukarı örneklemiştir. Geleneksel bir yöntem olan biküçük interpolasyon, kullanılan bilgisayar konfigürasyonu ile saniyede ortalama 20.89 görüntüyü yukarı örneklemiştir.

Çizelge 4.11 Algoritmaların test süreleri

	SRCNN	VDSR	ESPCN	EDSR	DCSCN	LapSRN	RCAN	IDN
Test Süresi (iterasyon/saniye)	8.33	5.93	14.71	2.96	8.81	11.9	7.98	8.12

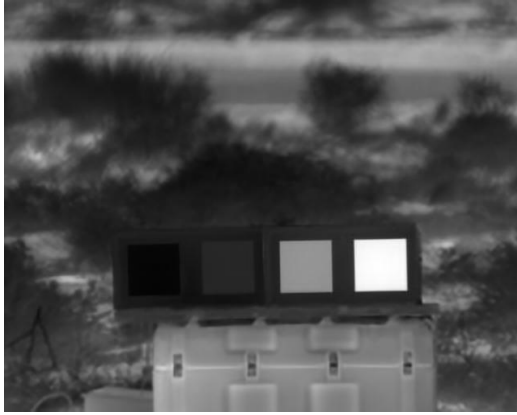
Test veri kümesi içindeki bazı görüntüler ile biküçük interpolasyon, EDSR, DCSCN algoritma çıktıları şekil 4.1, şekil 4.2, şekil 4.3, şekil 4.4 ve şekil 4.5 ile sunulmuştur. EDSR ve DCSCN algoritmalarının tercih edilme sebebi çizelge 4.9 ile verilen performans değerleri arasında en iyi 2 algoritma oluşudur. Şekil 4.2’de bulunan askeri araç HR görüntü ebadı 35X100, şekil 4.3’de bulunan sivil araç HR görüntüsü ebadı 25X50’dir. Görüntü veya görüntü parçaları orijinal görüntü ile PSNR metriği ile karşılaştırılmış olup değerler görüntülerin altına eklenmiştir.



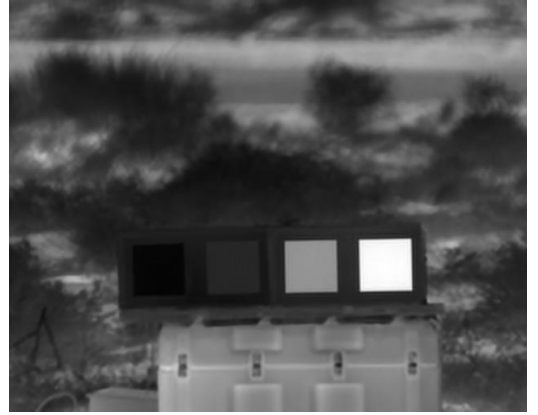
Orijinal Görüntü



Bikübik (PSNR: 36.9073)



EDSR (PSNR: 41.2718)

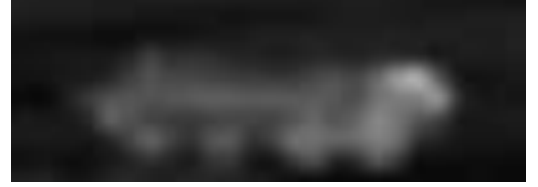


DCSCN (PSNR: 40.4890)

Şekil 4.1 Kalibrasyon nesnesi için 3 algoritmanın karşılaştırılması



Askeri araç LR görüntüsü



Orijinal Görüntü

Bikübik (PSNR: 26.4334)



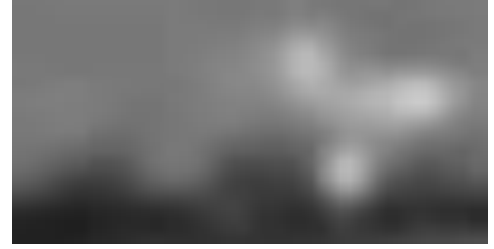
EDSR (PSNR: 27.9282)

DCSCN (PSNR: 28.0774)

Şekil 4.2 Askeri araç görüntüsü için 3 algoritmanın karşılaştırılması

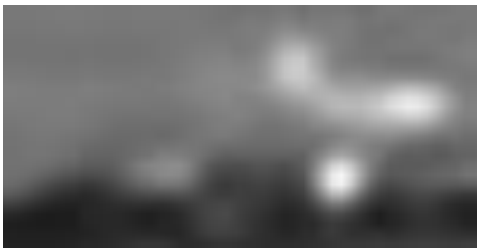


Sivil araç LR görüntüsü



Orijinal Görüntü

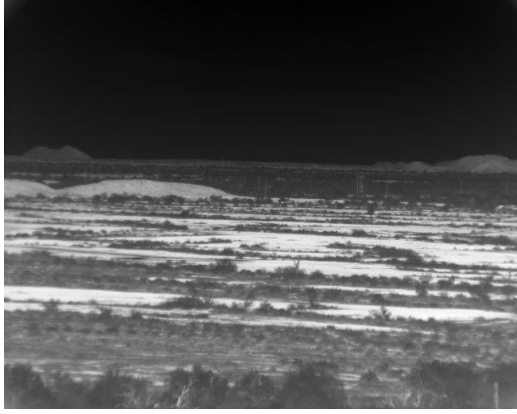
Bikübik (PSNR: 23.6701)



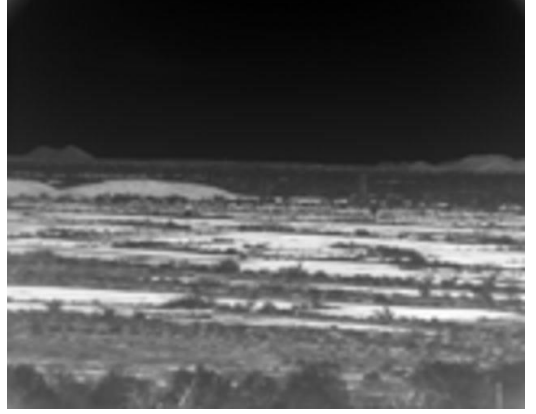
EDSR (PSNR: 24.8785)

DCSCN (PSNR: 25.1446)

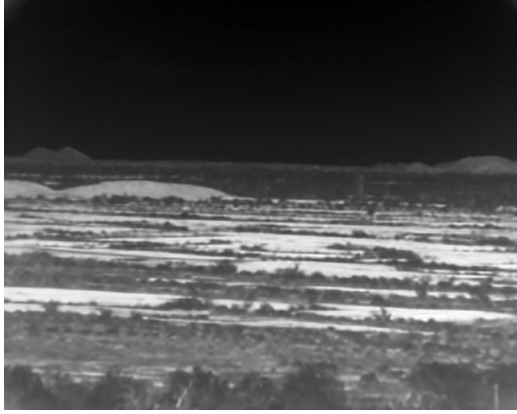
Şekil 4.3 Sivil araç görüntüsü için 3 algoritmanın karşılaştırılması



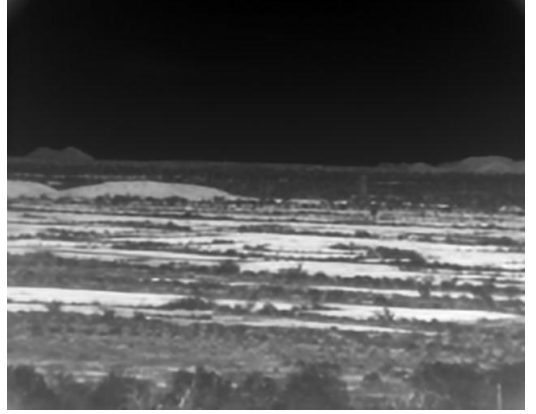
Orijinal Görüntü



Bikübik (PSNR: 28.7369)



EDSR (PSNR: 32.9924)



DCSCN (PSNR: 32.0020)

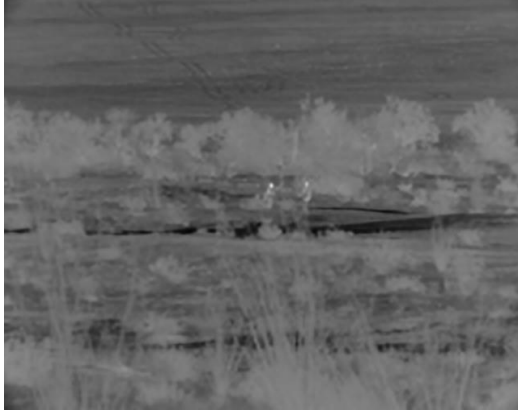
Şekil 4.4 Arazi görüntüsü için 3 algoritmanın karşılaştırılması



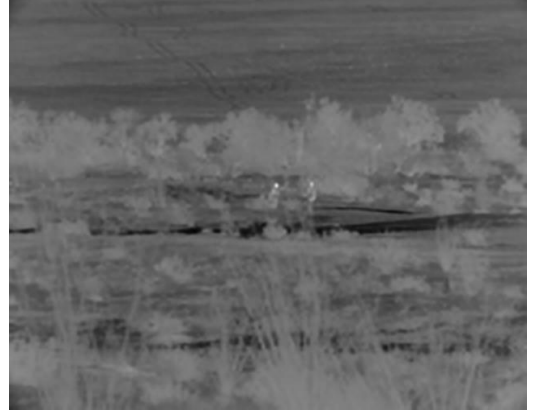
Orijinal Görüntü



Bikübik (PSNR: 36.4963)



EDSR (PSNR: 37.5226)



DCSCN (PSNR: 37.5045)

Şekil 4.5 İnsan görüntüsü için 3 algoritmanın karşılaştırılması

Hem görüntü kalitesi hem de eğitim süresi dikkate alındığında DCSCN algoritması veri kümesi için daha uygulanabilir durumdadır. Yamanaka vd. (2017) yaptığı çalışmada öznitelik çıkarma bloğu için 7 katman, yeniden inşa bloğu için ağ içinde ağ yapısı haricinde 1 katman kullanmıştır. Sönümlenme hiper-parametresi 0.8 olarak kullanılmış olup, her bir eğitim yinelemesinde nöronların %20'si rasgele olarak sönümlenmektedir.

2 bloktaki katman sayısı ve sönümlenme hiper-parametresi değiştirilerek eğitim ve test gerçekleştirilip, makalede yer alan mimari ile değiştirilen mimari veri kümesi üzerinde performans metrikleri ve eğitim süresi bakımından karşılaştırılmıştır. Sönümlenme oranı

p , öznitelik çıkarma katman sayısı n_1 , yeniden inşa katman sayısı n_2 olarak temsil edilmiş olup, performans çizelge 4.12 ile sunulmuştur.

Çizelge 4.12 Parametrelere göre DCSCN performans çizelgesi

	$p = 0.8$	$p = 0.9$	$p = 1.0$	$p = 1.0$	$p = 1.0$	$p = 1.0$	$p = 1.0$	$p = 1.0$
Metrik	$n_1 = 7$	$n_1 = 7$	$n_1 = 7$	$n_1 = 7$	$n_1 = 7$	$n_1 = 8$	$n_1 = 9$	$n_1 = 8$
	$n_2 = 1$	$n_2 = 1$	$n_2 = 1$	$n_2 = 2$	$n_2 = 3$	$n_2 = 3$	$n_2 = 3$	$n_2 = 4$
MSE	10.515	10.205	10.041	9.9421	9.7355	9.5826	10.007	9.7704
PSNR	40.567	40.648	40.702	40.727	40.835	40.875	40.707	40.806
SSIM	0.9503	0.9510	0.9513	0.9515	0.9521	0.9526	0.9514	0.9520
Süre	17dk.	17dk.	17dk.	20dk.	23dk.	24dk.	28dk.	32dk.

Çizelge 4.12 ile görüldüğü üzere her bir parametre değiştirilerek sonuçlar karşılaştırılmış, başarımın arttığı durumda parametre düzenlemeye devam edilmiş aksi durumda diğer parametre değiştirilmeye başlanmıştır. Sönümlenme etkisi ortadan kaldırıldığında, ilk blok için 8 ve ikinci blok için ağ içinde ağ yapısı haricinde 3 katman seçildiğinde en yüksek başarım elde edilmiştir. DCSCN algoritması verilen parametreler ile çalışılan veri kümesi üzerinde en başarılı sonucu vermekle beraber başarı bakımından en yakın EDSR algoritmasının eğitim süresinden çok daha kısa bir sürede eğitimini tamamlamaktadır.

Test veri kümesinde bulunan 640x512 boyutundaki görüntüler, mimarisi değiştirilen DCSCN algoritması ve biküçük interpolasyon yöntemi kullanılarak 2 yukarı örnekleme ölçeği ile yukarı örneklenmiştir. Test kümesinde yer alan ve şekil 4.6 ile verilen 640x512 boyutundaki görüntü yukarı örneklenmiş, elde edilen 1280x1024 boyutundaki görüntüler şekil 4.7 ve şekil 4.8 ile verilmiş olup görüntüdeki askeri araç görüntü parçası yeniden boyutlandırılarak görüntülerin sağ alt kısmına yerleştirilmiştir.



Şekil 4.6 Orijinal askeri araç görüntüsü (640x512)



Şekil 4.7 Bikübik interpolasyon ile yukarı örneklenmiş görüntü (1280x1024)



Şekil 4.8 DCSCN algoritması ile yukarı örneklenmiş görüntü (1280x1024)

5. TARTIŞMA VE SONUÇ

Literatürdeki çalışmalar, sundukları algoritmalar ile daha önceki çalışmalar arasında bir karşılaştırma yapabilmek için kullandıkları veri kümelerinin en az bir tanesini önceki çalışmalardan seçmiş ve daha iyi bir performans ortaya koymuştur. Literatürde kullanılan ve tez kapsamında uygulanan algoritmalar, makalelerinde 3 kanallı görüntüden oluşan veri kümeleri üzerinde test edilmiştir. Tez çalışmasında kullanılan görüntülerin ise tek kanal ve MWIR görüntüleri olmasından dolayı elde edilen sonuçlar literatürdeki algoritma karşılaştırmalarından farklıdır.

Derin öğrenme tabanlı algoritmaların hiper-parametreleri değiştirilerek daha iyi sonuçlar elde edilmeye çalışılmıştır. Değiştirilen hiper-parametreler içinde yığın boyutu, parça boyutu ve öğrenme oranı bulunmaktadır. Yığın boyutunun artırılması genellikle daha iyi sonuç alınmasını sağlarken, yüksek yığın boyutları daha gelişmiş bir bilgisayar konfigürasyonu gerektirmektedir. Yığın boyutunun düşük olması eğitim sürecinde aşırı uydurmaya sebep olurken, çok yüksek yığın boyutu eksik uydurmaya sebep olmaktadır. Bu sebeple en uygun değer deneme yapılarak belirlenmiştir. Süper çözünürlük mimarilerinde kullanılan parça boyutu hiper-parametresi ise görüntülerin kullanılan GPU'da çalışabilmesi için parçalara bölünmesi gerektiğinden oluşturulmuştur. Bilgisayar konfigürasyonunun yeterliliği ve eğitim süreleri göz önünde bulundurularak olabildiğince yüksek parça boyutu (128x128) kullanılmıştır. Daha düşük parça boyutları ile genellikle daha kötü sonuçlar elde edilmiştir. Birbiri ile kısmen örtüşen 2 adet görüntü parçası 1 adet LR görüntüsünü oluşturmaktadır. Mimarilere uygun olan görüntü parça yükseklik ve genişlikleri eşit olduğundan bu durum oluşabilmektedir. Her bir algoritma için makalelerinde yer alan öğrenme oranları ile denemeler yapılmış, aşağı/yukarı yönde değiştirmeler yapılarak en uygun değerler seçilmiştir. Karşılaştırmaların daha makul olması için yığın boyutu ve parça boyutu değerleri tüm algoritmalar için aynı seçilmiş, yalnızca öğrenme oranı değerleri farklı seçilmiştir.

Derin öğrenme tabanlı 8 süper çözünürlük algoritması, kullanılan veri kümesinde klasik yöntemden daha iyi bir sonuç vermiştir. Kalan değerli ağ mimarisine sahip EDSR ve DCSCN algoritmaları en iyi sonuçları vermiş, DCSCN mimarisi ve hiper-parametreleri

değiştirilerek MSE (9.5826), PSNR (40.875) ve SSIM (0.9526) metrikleri açısından en başarılı sonuç EDSR algoritma eğitim süresinden çok daha kısa bir sürede (24 dk.) eğitim yapılarak elde edilmiştir. 8-bit derinliğindeki görüntüler için 40.875 olan PSNR değeri başarılı kabul edilebilirken, daha yüksek bit derinliğine sahip görüntüler kullanıldığında daha yüksek PSNR değerleri hedeflenmelidir.

Literatürdeki her bir algoritma farklı yukarı örnekleme ölçeği için farklı performans vermektedir. Tezde kullanılan algoritmalar 4 ölçeği ile aşağı örneklenmiş 160x128 boyutundaki görüntüleri kullanarak 640x512 ebadındaki HR görüntüleri elde etmiştir. Ölçek faktörü görüntülerin yukarı örneklenmesinde önemli bir etken olup, daha düşük ölçekler ile daha iyi sonuçlar alınabileceği öngörülmektedir (ölçek: x2, x3). Daha yüksek ölçekler (x8, x16) ile çalışma yapılabilmesi için daha yüksek çözünürlüklü LR görüntülerinin kullanılması önerilmektedir.

Görüntülerin bütününe bakıldığında, algoritmaların çıktıları klasik yöntemden daha iyi sonuç vermekle beraber, görüntü içerisindeki ebat olarak küçük nesnelere düşük PSNR (< 30dB) değerine sahiptir. LR görüntülerindeki parçalar incelenirken boyut için bir alt sınır belirlenerek parça içerisinde yeterli bilgi olması sağlanabilir. Aksi durumda görüntü detaylarının çıkarılması ve başarılı bir sonuç elde edilmesi mümkün olmayabilir. Bu bulgular eşliğinde, algoritmaların daha başarılı sonuç vermesi için kullanılan eğitim veri kümesinin nicelik ve nitelik bakımından artırılması önerilmektedir.

Test veri kümesinde bulunan 640x512 boyutundaki görüntüler, mimarisi değiştirilen DCSCN algoritması ve biküçük interpolasyon yöntemi kullanılarak 2 yukarı örnekleme ölçeği ile yukarı örneklenmiş olup, 1280x1024 boyutunda görüntüler elde edilmiştir. Elde edilen görüntüler birbirine yakın görüntüler olmuştur. Daha yüksek yukarı örnekleme ölçekleri (x3, x4) ile deneme yapılmasına karşın ekran kartı sınırlılığı sebebiyle test gerçekleştirilememiştir. Bu durum, çözünürlüğün ve ölçek faktörünün artması durumunda daha güçlü ekran kartına ihtiyaç duyulduğunu göstermektedir.

Algoritmaların başarısının yanı sıra eğitim süresi, veri kümesinin büyüklüğüne göre değişkenlik göstermektedir. Daha fazla CUDA (Compute Unified Device Architecture)

ekirdeđi ieren GPU kullanımının eđitim suresini kısaltacađı ngrlmektedir. Algoritmaların test sureleri, elde edilecek yukarı rneklenmiř grntlerin evrimdiři olarak analiz edilmediđi ve gerek-zamanlı sistemler ile kullanıldıđı durumlarda byk nem arz etmektedir. Kullanılan kamera ereve hızı (frame rate) ve bađlı bilgisayar konfigrasyonu dřnlerek algoritma tercihi yapılması nerilmektedir.

Tez alıřmasında elde edilen bulgular ve ıkarımlar, derin đrenme tabanlı sper znrlk algoritmalarının klasik yntemler yerine MWIR grntlerin yukarı rneklenmesinde kullanılabileređini gstermiřtir.

KAYNAKLAR

- Anonymous. 2024. Web Sitesi: <https://dsiac.org/databases/atr-algorithm-development-image-database/>, Erişim Tarihi: 01.01.2024.
- Dong, C., Loy, C. C., He, K. and Tang, X. 2014. Learning a Deep Convolutional Network for Image Super-Resolution, European Conference on Computer Vision (ECCV), 6 September, 184-199, Zurich, Switzerland.
- Droege, D. R., Hardie, R. C., Allen, B. S., Dapore, A. J. and Blevins, J. C. 2012. A real-time atmospheric turbulence mitigation and super-resolution solution for infrared imaging systems. *Proceedings SPIE*, 8355(1); 234–250.
- Gonzalez, R. C. and Woods, R. E. 2018. *Digital Image Processing 4th Edition*. Pearson Education, 1022 p., New York
- Haykin, S. 2009. *Neural Networks and Learning Machines Third Edition*. Pearson Prentice Hall, 906 p., USA.
- Hui, Z., Wang, X., Gao, X. 2018. Fast and Accurate Single Image Super-Resolution via Information Distillation Network, *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 18-22 June, 723-731, Salt Lake City, Utah, USA.
- Kim, J., Lee, J. K. and Lee, M. 2016. Accurate Image Super-Resolution Using Very Deep Convolutional Networks, *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 26 June, 1646-1654, Las Vegas, NV, USA.
- Lai, W., Lai, S., Huang, J., Huang B., Ahuja, N., Yang, M. and Yang, H. 2017. Fast and Accurate Image Super-Resolution with Deep Laplacian Pyramid Networks, *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 21-26 July, 624-632, Honolulu, HI, USA.
- Ledig, C., Theis, L., Huszar, F., Caballero, J., Cunningham, A., Acosta, A., Aitken, A., Tejani, A., Totz, J., Wang, Z. 2017. Photo-Realistic Single Image Super-Resolution Using a Generative Adversarial Network, *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 21-26 July, 105-114, Honolulu, HI, USA.
- Lee, I. H., Chung, W. Y., Park, C. G. 2023. Style transformation super-resolution GAN for extremely small infrared target image, *Pattern Recognition Letters*, 174; 1-9.
- Lepcha, D. C., Goyal, B., Dogra, A., Goyal, V. 2023. Image super-resolution: A comprehensive review, recent trends, challenges and applications. *Information Fusion*, 91(1); 230-260.

- Lim, B., Son, S., Kim, H., Nah, S. and Lee, K. M. 2017. Enhanced Deep Residual Networks for Single Image Super-Resolution, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 21-26 July, 1132-1140, Honolulu, HI, USA.
- Mendez-Rial, R., Souto-Lopez, A. and Garcia-Diaz, A. 2018. Embedded video rate super-resolution in the infrared with a low-cost multi-aperture camera, Unconventional Optical Imaging, 22-26 April, Strasbourg, France.
- Shi, W., Caballero, J., Huszar, F., Totz, J., Aitken, A. P., Bishop, R., Rueckert, D., Wang, Z., Twitter 2016. Real-Time Single Image and Video Super-Resolution Using an Efficient Sub-Pixel Convolutional Neural Network, Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 26 June, 1874-1883, Las Vegas, NV, USA.
- Wu, Z. and Wang, X. 2019. Focal plane array-based compressive imaging in medium wave infrared: modeling, implementation, and challenges. Applied Optics, 58(31); 8433-8441.
- Yamanaka, J., Kuwashima, S. and Kurita, T. 2017. Fast and Accurate Image Super Resolution by Deep CNN with Skip Connection and Network in Network, 24th International Conference, ICONIP 2017, 14-18 November, 217-225, Guangzhou, China.
- Zhang, Y., Li, Ku., Li, K., Wang, L., Zhong, B. and Fu, Y. 2018. Image Super-Resolution Using Very Deep Residual Channel Attention Networks, Computer Vision and Pattern Recognition (CVPR), 8-14 September, 286-301, Munich, Germany.