T.C.

YILDIZ TEKNİK ÜNİVERSİTESİ

FEN BİLİMLERİ ENSTİTÜSÜ

DERİN ÖĞRENME İLE YEŞİL ALANLARIN ÇIKARIMI

Yusuf Yekta GÜZEL

YÜKSEK LİSANS TEZİ

Harita Mühendisliği Anabilim Dalı

Uzaktan Algılama ve CBS Programı

Danışman

Dr. Öğr. Üyesi Anime Melis UZAR DİNLEMEK

Haziran, 2021

T.C.

YILDIZ TEKNİK ÜNİVERSİTESİ

FEN BİLİMLERİ ENSTİTÜSÜ

DERİN ÖĞRENME İLE YEŞİL ALANLARIN ÇIKARIMI

Yusuf Yekta GÜZEL tarafından hazırlanan tez çalışması 06.07.2021 tarihinde aşağıdaki jüri tarafından Yıldız Teknik Üniversitesi Fen Bilimleri Enstitüsü Harita Mühendisliği Anabilim Dalı, Uzaktan Algılama ve CBS Programı **YÜKSEK LİSANS TEZİ** olarak kabul edilmiştir.

Dr. Öğr. Üyesi Anime Melis UZAR
DİNLEMEK
Yıldız Teknik Üniversitesi
Danışman

Jüri Üyeleri

Dr. Öğr. Üyesi Anime Melis UZAR DİNLEMEK, Danışman
Yıldız Teknik Üniversitesi _____

Doç. Dr. Saygın ABDİKAN, Üye
Hacettepe Üniversitesi _____

Dr. Öğr. Üyesi Uğur ACAR, Üye
Yıldız Teknik Üniversitesi _____

Danışmanım Dr. Öğr. Üyesi Anime Melis UZAR DİNLEMEK sorumluluğunda tarafımca hazırlanan Derin Öğrenme ile Yeşil Alanların Çıkarımı başlıklı çalışmada veri toplama ve veri kullanımında gerekli yasal izinleri aldığımı, diğer kaynaklardan aldığım bilgileri ana metin ve referanslarda eksiksiz gösterdiğimi, araştırma verilerine ve sonuçlarına ilişkin çarpıtma ve/veya sahtecilik yapmadığımı, çalışmam süresince bilimsel araştırma ve etik ilkelerine uygun davrandığımı beyan ederim. Beyanımın aksinin ispatı halinde her türlü yasal sonucu kabul ederim.

Yusuf Yekta GÜZEL

Aileme

ve

sevdiceğime

TEŞEKKÜR

Değerli görüş ve önerileri ile çalışmama katkıda bulunan tez danışmanım Dr. Öğr. Üyesi Anime Melis UZAR DİNLEMEK'e ve Doç. Dr. Nursu TUNALIOĞLU'na,

Çalışmalarım boyunca bana birikimlerini aktaran, tavsiyelerini aldığım hocam Arş. Gör. Onur Can BAYRAK'a,

Maddi ve manevi desteklerini hiçbir zaman esirgemeyen, her an yanımda olan sevgili ailem ve biricik Burcum'a sonsuz teşekkürlerimi sunarım.

Yusuf Yekta GÜZEL

İÇİNDEKİLER

ŞEKİL LİSTESİ **ix**

TABLO LİSTESİ **xi**

ÖZET **xii**

ABSTRACT **xiv**

1 GİRİŞ **1**

1.1 Literatür Özeti 2

1.2 Tezin Amacı 7

1.3 Hipotez 7

2 DERİN ÖĞRENME **8**

2.1 Tanım 8

2.2 Tarihçe 8

2.3 Yapay Sinir Ağları 9

2.4 Derin Öğrenme Ağlarının Katmanları 11

2.5 Katman Türleri 12

2.6 Evrişimli Sinir Ağı Algoritmaları 16

2.7 Veri İşleme 21

2.8 Yapay Sinir Ağı Eğitimi 21

2.9 Optimum Ağırlıkların Bulunması 22

2.10 Ağırlıkları Düzenleme 23

2.11 Öğrenme Oranı 25

2.12 Uygulamada Kullanılan Algoritmalar 25

2.13 Doğruluk Ölçütleri 32

3 UYGULAMA **35**

3.1 Çalışma Alanı ve Kullanılan Veriler 35

3.2 Eğitim Verilerinin Hazırlanması 37

3.3 Kullanılan Donanım ve Yazılımlar 38

3.4 Ağ Parametreleri 38

3.5 Yeşil Alanların Bölütleme Doğruluğu 39

4 SONUÇ VE ÖNERİLER **50**

KAYNAKÇA **52**

TEZDEN ÜRETİLMİŞ YAYINLAR **56**

SİMGE LİSTESİ

A, B	Veri seti
W	Ağırlık
x	Girdi vektörü
b	Bias (eğilim)
z	Ürün (çıktı)
i, j	Ağda yer alan gizli katmanın sırası
I	Matris girdi boyutu
F	Matris boyutu
S	Adım aralığı
O	Matris çıktı boyutu
K	Filtre sayısı
P	Sıfır dolgulama miktarı
p	Eşikleyici olasılık vektörü
x_i	Eşikleyici skoru
γ, β	Küme normalleştirme hiper-parametreleri
μ_B	Küme ortalaması
σ^2_B	Küme varyansı
$L(z, y)$	Yitim fonksiyonu
$d/dw, d/dx$	Türev
α, η	Öğrenme oranı

KISALTMA LİSTESİ

İHA	İnsansız Hava Aracı
YSA	Yapay Sinir Ağları
UAV	Unmanned Aerial Vehicle
KSC	Kennedy Space Center
GPU	Graphics Processing Unit
ESA	Evrişimli Sinir Ağı
CNN	Convolutional Neural Network
MNIST	Modified National Institute of Standards and Technology
ReLU	Rectified Linear Unit
ResNET	Residual Network
NIN	Network in Network
SGD	Stochastic Gradient Descent
DenseNet	Densely Connected Convolutional Networks
LiDAR	Light Imaging Detection and Ranging
DBK	Light Imaging Detection and Ranging
KKÇD	K-Katlı Çapraz Doğrulama
OpenCV	Open Source Computer Vision Library
NumPy	Numerical Python
CBS	Coğrafi Bilgi Sistemleri
TP	True Positive
TF	True False
FP	False Positive
FN	False Negative

ŞEKİL LİSTESİ

Şekil 1.1 Yapay zeka, makine ve derin öğrenme ilişkisi [1]. 2
Şekil 2.1 İnsana ait bir sinir hücresi [23]. 9
Şekil 2.2 Yapay sinir ağı modeli [24]. 10
Şekil 2.3 Gizli katman sayısına göre YSA modelleri [24]. 11
Şekil 2.4 ESA mimarisindeki katmanlar [27]. 12
Şekil 2.5 YSA mimarisi [29]. 12
Şekil 2.6 Girdiye filtre uygulanarak yapılan evrişim işlemi [27]. 13
Şekil 2.7 Filtrelerin yükseklik, genişlik ve kanal sayısı gösteren model [27]. 13
Şekil 2.8 Filtrenin adım aralığı [27]. 13
Şekil 2.9 Sıfır dolgulama [27]. 14
Şekil 2.10 Evrişim katmanı girdi ve çıktısının matris boyutları [27]. 14
Şekil 2.11 Yaygın kullanılan aktivasyon fonksiyonları [29].....15
Şekil 2.12 Tam bağlantı katmanı modeli [27]. 16
Şekil 2.13 LeNet-5 mimarisi.....17
Şekil 2.14 AlexNet mimarisi [31].....17
Şekil 2.15 VGG16 mimarisi.....18
Şekil 2.16 ResNet mimarisi [33]. 19
Şekil 2.17 Network in Network (NIN) mimarisi [34] 20
Şekil 2.18 GoogLeNet inception modülü.....20
Şekil 2.19 GoogLeNet inception ağı [35]. 20
Şekil 2.20 İleri ve geri yayılım yönlerini gösteren model [36]. 23
Şekil 2.21 Ağırlıkların güncelleme aşamaları [36]. 23
Şekil 2.22 Seyreltme işlemi [36].....24
Şekil 2.23 Erken durdurma tekniğini gösteren uyarlanmış grafik [36].....24
Şekil 2.24 U-Net mimarisi [39].....26
Şekil 2.25 VGG19 mimarisi [40].....27
Şekil 2.26 ResUNet mimarisi [41]. 27
Şekil 2.27 Dense bloğu [42]. 29
Şekil 2.28 DenseUNet mimarisi [42]. 30
Şekil 2.29 Basit bir Inception modeli [44]. 31
Şekil 2.30 MultiRes Bloğu oluşturmak için hazırlanan 3x3'lük evrişim işlemleri [44]. 31

Şekil 2.31 MultiRes Bloğu [44].....31
Şekil 2.32 MultiResUNet'te yer alan Res yolu [44]. 32
Şekil 2.33 MultiResUNet mimarisi [44]. 32

Şekil 3.1 Davutpaşa Kampüsü, İstanbul (Google Earth).....35
Şekil 3.2 İHA ile alım yapılan hava fotoğraflarının yoğunluk raporu. 36
Şekil 3.3 (a) Etiketlenmiş veri ve (b) yeşil alanı temsil eden siyah-beyaz görüntü. 37
Şekil 3.4 Akış şeması. 37
Şekil 3.5 Beş mimari için eğitim ve test doğrulukları. 40
Şekil 3.6 Bölütlenmiş bir görüntüye ait ortofoto görüntüsü, operatör etiketi ve mimarilerin yeşil alan tahminlerini gösteren çıktı görüntüler.....43
Şekil 3.7 (a), (b), (c), (d) ile gösterilen alanlar UNet-VGG19 mimarisinin sık alanlardaki yeşil alan tespitini göstermektedir. 44
Şekil 3.8 Gösterilen alan UNet-VGG19 mimarisinin gölgedeki yeşil alan tespitini göstermektedir.....45
Şekil 3.9 (a) ve (b) ile gösterilen alan UNet-VGG19 mimarisinin ağaç gölgeleri ile yeşil alanları karıştırmadığını göstermektedir. 45
Şekil 3.10 (a), (b), (c), (d) ile gösterilen alanlar UNet-VGG19 mimarisinin seyrek ağaçları tespitini göstermektedir.....46
Şekil 3.11 (a), (b), (c), (d) ile gösterilen alanlar UNet-VGG19 mimarisinin zeminde yer alan yeşil alanların tespitini göstermektedir.....46
Şekil 3.12 Gösterilen alan UNet-VGG19 mimarisinin çatıda yer alan malzemenin üstüne düşen gölgeyi yeşil alan olarak tespit ettiğini göstermektedir. .47
Şekil 3.13 (a), (b), (c), (d) ile gösterilen alanlar UNet-VGG19 mimarisinin yeşil alan olmadığı halde yeşil alan olarak tespit ettiğini nesneleri göstermektedir.....47

TABLO LİSTESİ

Tablo 1.1 Literatürde İncelenen Çalışmaların Tablosu 5
Tablo 2.1 Karşılaştırma Matrisi 33
Tablo 3.1 Ağ Eğitimindeki Parametreler 38
Tablo 3.2 Mimarilere Ait Parametre Sayıları 39
Tablo 3.3 Karşılaştırma Matrisinden Elde Edilen Sonuçlar 41
Tablo 3.4 Yanlış veya Doğru Tespit Edilen Yeşil Alan Çıkarım Çıktıları 42
Tablo 3.5 Yeşil Alan Sınıfının Mimari ve Sınıf Analizlerine Göre Çıkarım Değerleri 48

ÖZET

Derin Öğrenme ile Yeşil Alanların Çıkarımı

Yusuf Yekta GÜZEL

Harita Mühendisliği Anabilim Dalı

Yüksek Lisans Tezi

Danışman: Dr. Öğr. Üyesi Anime Melis UZAR DİNLEMEK

Dünyada, doğal habitat ve sağlıklı çevreler oluşturmaları nedeniyle yeşil alanlar küresel iklim değişikliklerinin izlenmesi, hava tahminleri gibi konularda önemli bir yerdedir. Ayrıca, kentsel ve kırsal alanların dönüşümü, afet ve kriz yönetimi gibi konularda zamansal değişim analizi yapılması, ormanlık alan çıkarımı gibi sürdürülebilir kaynakların takibi ve envanter oluşturulması açısından önem teşkil etmektedir. Son yıllarda, küresel iklim değişikliği açısından önemli olan bu alanların izlenmelerine yönelik yürütülen çalışmalarda, mevcut yöntemler ile derin öğrenme yöntemlerinin entegrasyonu gündeme gelmektedir. Derin öğrenme, temelinde YSA (Yapay Sinir Ağı) prensibini kullanarak, tek katmanlı sinir ağlarından çok katmanlı sinir ağlarına geliştirilen ve evrimleşerek yöntemin temellerini oluşturan bir mimariye sahiptir. Bu yöntem, el yazısıyla yazılan rakamları tanıma, fotoğraflardan bireylerin ruh halini tespit etme, trafikte seyreden taşıtları ve yayaları tanıma, LiDAR (Light Imaging Detection and Ranging) sensörleriyle 3 boyutlu tespit, uydu görüntüleri ile yeşil alanların zamansal değişimi veya farklı organlarda gözlemlenebilen kanserli hücrelerin röntgen görüntülerinden tespiti gibi konular üzerine yapılan çalışmalarda kullanılmıştır. Son yıllardaki derin öğrenmeye dair bu ivmelenme, derin öğrenme yöntemlerinin sunduğu avantajların Harita Mühendisliği'nin çalışma konularında

da uygulanmasını sağlamıştır. Uzaktan algılama ve fotogrametri verilerinin derin öğrenme yöntemi ile entegrasyonu farklı kırsal ve kentsel çalışma alanlarının segmentasyonu ve kıyı çizgisi çıkarımı gibi sulak çalışma alanlarında hız ve doğruluk bakımından avantaj sağlamaktadır. Özellikle obje çıkarımı konusunda derin öğrenme algoritmalarının kullanımı ve makine eğitimi ile operatör müdahalesine ihtiyaç duymadan otomatik çıkarım gerçekleştirilmektedir. Böylece objeyi en iyi şekilde temsil edebilecek hiyerarşik bir yapı oluşturulabilmektedir. Tez çalışması kapsamında; 159 adet İHA (İnsansız Hava Aracı) görüntüsünden üretilen, Yıldız Teknik Üniversitesi Davutpaşa Kampüsü'ne ait ortofoto kullanılarak kampüsteki yeşil alanlar U-Net, UNet-VGG19, ResUNet, DenseUNet ve MultiResUNet kullanılarak çıkarılmıştır. En yüksek doğruluk oranını %94 DBK (Dice Benzerlik Katsayısı) oranı ile U-Net mimarisinde, %93 doğruluk oranına ise UNet-VGG19, ResUNet, DenseUNet mimarilerinde ulaşılmıştır. Elde edilen sonuçlar doğrultusunda, derin öğrenme yöntemlerinin İHA görüntülerinden yeşil alanların çıkarımında kullanılabileceği sonucuna varılmıştır.

Anahtar Kelimeler: Derin Öğrenme, Yeşil Alanların Çıkarımı, Otomatik Sınıflandırma, İHA, UNet.

ABSTRACT

Exraction of Green Areas with Deep Learning

Yusuf Yekta GÜZEL

Department of Geomatic Engineering

MSc Thesis

Advisor: Asst. Prof. Anime Melis UZAR DİNLEMEK

Green areas have an important place in the world in terms of monitoring global climate changes and weather forecasts, since they provide natural habitats and healthy environments. In addition, conducting temporal change analysis in subjects such as transformation of urban and rural areas, disaster and crisis management is important in terms of monitoring the sustainable resources and creating inventory such as forestry extraction. In recent years, the integration of existing methods and deep learning methods has come to the fore in studies conducted to monitor the areas mentioned that are important for global climate change. Deep learning, which is a method of computer vision, has an architecture that is developed from single-layer neural networks to multi-layer neural networks using the artificial neural network principle, and evolves to form the basis of the method. This method is widely used today in recognizing handwritten numbers, detecting the mood of individuals from photographs, recognizing vehicles and pedestrians in traffic, 3-dimensional detection with LiDAR (Light Imaging Detection and Ranging) sensors, temporal variation of green areas with satellite images, or x-rays of cancerous cells that can be observed in different organs. This acceleration of deep learning in recent years has enabled the advantages of deep learning methods to be applied in the study areas of Geomatics Engineering.

Integration of remote sensing and photogrammetric data with deep learning method provides advantages in terms of speed and accuracy in many study areas. With the use of deep learning algorithms and machine training, especially in object extraction, automatic extraction is performed without the need for operator intervention. Thus, a hierarchical structure can be created that can best represent the object. In this study, it is aimed to extract green areas in campus were extracted using U-Net, UNet-VGG19, ResUNet, DenseUNet and MultiResUNet, using the orthophoto of Yıldız Technical University Davutpaşa Campus produced from 159 UAV (Unmanned Aerial Vehicle) images. The highest accuracy rate was achieved in the U-Net architecture with 94% DBK (Dice Similarity Coefficient) and 93% accuracy in UNet-VGG19, ResUNet, DenseUNet architectures. In line with the results obtained, it was concluded that deep learning methods can be used to extract green areas from UAV images.

Keywords: Deep Learning, Extraction of green areas, Automatic classification, UAV, UNet.

1
GİRİŞ

Modern teknoloji kullanılarak yerel ve özel yönetimlerdeki işlemlerin operatör ihtiyacını en düşük seviyeye indirerek hızlı, doğru ve kontrollü bir şekilde yapılması sağlanmaktadır. Bu doğrultuda, gelişen teknolojiler ile bilgiyi algılayan, bunları depolayan ve bilgiyi güncelleyerek sistematik bir şekilde uyum sağlayabilen ve öğrenen bir sistem tasarımı hedeflenmektedir. Bilgisayarla görü bu noktada geleceğe açılan bir kapıdır. Bu terim, fotoğraf veya video gibi görsel bir ürünü girdi olarak kullanarak bu ürün üzerinde anlamlandırma çalışmalarını ifade eder. Görsel bir ürünü anlamlandırma çalışması, makinenin veriden öğrenmesini, otomatik bir biçimde bilgiye ulaşmasını ve depolaması sayesinde gerçekleşir. Bu unsurların bütününü kapsayan bir sistem esasında yapay zekayı ifade eder (Şekil 1.1). Eğitilen bu algoritmalar yapay zekanın bir alt kümesi olan makine öğrenmesi olarak ifade edilir. Makine öğrenmesi, gerçekleştirilirken kullanılan en küçük birimler YSA' dır. Çeşitli katmanlardan oluşan ve ismini esinlendiği insan sinir ağlarından alan bu sistemler, makinenin öğrenmesini sağlayan en önemli unsurlardır. Eğer ki kullanılan YSA çok katmanlı ve gizli ağlardan oluşan bir modelse bu durumda bu sistem, makine öğrenmesi yerine derin öğrenmeyle ifade edilir. Derin öğrenme, makine öğrenmesinin bir alt kümesi olmasına rağmen daha otomatikleştirilmiş, manuel müdahaleye gerek duymayan derin YSA yapısına sahip bir modeldir (Şekil 1.1). Yapay zekanın öğrenme metodu verilerden özellikler çıkarılarak eğitilen algoritmalardır.



Şekil 1.1 Yapay zeka, makine ve derin öğrenme ilişkisi [1].

Derin öğrenme mimarileri geriye yayılım algoritmalarını kullanarak sığ mimarilere göre daha yüksek başarı gösterir. Geri yayılım algoritmaları, türevlenebilir fonksiyonlarla öğrenilen bilgileri peş peşe düzeltmeye imkan sağlarken derin öğrenme algoritmalarında kullanılan seyreltme adımı ile makinenin gereksiz bağlantılar kurarak bilgi tekrarlamasını veya ezber yapmasını önler. Bununla birlikte derin algoritmalar, çıkartılmak istenen bilgilerin niteliğine göre özelleştirmeye imkan tanımaktadır. Bu sebepler bütünü, derin öğrenme yönteminin günümüzde popüler olmasının önünü açmıştır.

Derin öğrenmenin temel düzeydeki örnekleri el yazısıyla yazılan rakamları tanıma ve fotoğraflardan bireylerin ruh halini tespit etmedir. Trafikte seyreden taşıtları ve yayaları tanıma uydu görüntüleri ile yeşil alanların zamansal değişimi gibi uygulamalar ise çevresel değişiklikleri izlemeye dayalı uygulamalardır. Farklı organlarda gözlemlenebilen kanserli hücrelerin röntgen görüntülerinden tespiti ise tıbbın konusuna giren uygulama örneği olarak öne çıkar.

1.1 Literatür Özeti

Canlıların yaşadıkları topografik, klimatik, edafik, biyolojik, tarihi, sosyal ve ekonomik koşulların oluşturduğu çevre ekolojik bir alandır. Bu alanda söz konusu koşulların toplum yararına dengede tutulması hedeflenir. Bu sebeple yaşam alanı planlanırken yeşil alanların düzenlenmesine önem verilmelidir [2]. Yeşil alanlar, dünya üzerinde doğal habitat ve sağlıklı çevreleri oluşturur. Yeşil alanlar, yaşam döngüsünü sağlarken doğa, hava ve atmosfer dengesinin koruyucularıdır. Küresel

iklim değişikliklerinin takibi, kentsel ve kırsal alanların dönüşümü, afet ve kriz yönetimi, sürdürülebilir kaynakların tespiti ve takibi açısından önem teşkil etmektedir.

İnsanoğlunun geleceği açısından yeşil alanların tespiti, sınıflandırılması ve bunlara ait zenginliklerin ortaya koyulmasının önemi açıktır. Bu doğrultuda 2013'te M. Li ve diğerleri makine öğrenmesi yöntemi olan Karar Ağaçları, Rassal Orman ve Destek Vektör Makinelerini kullanarak çok zamanlı Landsat tematik harita görüntüleriyle orman sınıflandırma ve değişim analizi yapmışlardır. %77 - %88 arasında değişen doğruluk oranlarıyla elde edilen sonuçlara bakıldığı zaman en yüksek başarıya sahip olan yöntemin Destek Vektör Makineleri olduğu görülmüştür [3]. 2015'te T. Kavzoğlu ve diğerleri Karar Ağaçları, En Yakın Komşuluk ve Destek Vektör Makinelerini kullanarak yüksek çözünürlüklü uydu görüntüleriyle obje tabanlı yaklaşımda makine öğrenme algoritmalarının sınıflandırma performansının analizini yapmışlardır. %90 - %97 arasında değişen doğruluk oranında en yüksek başarıyı Destek Vektör Makineleri göstermiştir. Kendi içerisinde de kırsal alandaki test verisindeki başarı, kentsel alana göre daha yüksek olmuştur [4]. 2017'de A. Şeker ve diğerleri Derin Öğrenme Yöntemleri ve Uygulamaları Hakkında Bir İnceleme çalışmasıyla derin öğrenme konusundaki gelişmeleri ortaya koyan bir araştırma makalesi yayımlamışlardır [5]. 2018'de M. Wessel ve diğerleri obje ve piksel tabanlı Karar Ağacı ve Destek Vektör Makinelerini kullanarak Sentinel-2 uydu görüntülerini kullanarak sadece üç ağaç türünü sınıflandırmaya çalışmışlardır. Kullanılan sınıflandırıcı türüne ve diğer değişkenlere bağlı olarak %82 - %97 arasında değişen doğruluk oranlarına ulaşmışlardır [6]. F. Doğan ve İ. Türkoğlu'nun çalışmasında popüler derin öğrenme mimarileriyle yaprak sınıflandırma başarıları karşılaştırılmıştır. Bu çalışmada UCI-Machine Learning Repository verileri kullanılarak %97.77 - %99.72 arasında değişen doğruluk oranında en yüksek başarıyı AlexNet sergilemiştir [7]. 2018'deki başka bir çalışma M. A. Kızrak ve B. Bolat'ın Derin Öğrenme ile Kalabalık Analizi Üzerine Detaylı Bir Araştırma isimli araştırma makalesidir. YSA'dan derin öğrenmeye kadar modeller detaylı bir biçimde açıklanmıştır. Derin öğrenmenin aşamalarını ve matematiksel modelini açıkça ortaya koymuşlardır [8]. 2019'a gelindiğinde makine öğrenmesine dayalı kullanıcı

müdahalesi gerektiren sınıflandırıcılar yerini tamamen derin öğrenme mimarilerine bırakmıştır. O. Yıldız'ın C4Net derin öğrenme mimarisiyle tasarladığı ISIC doku görüntülerinin kullanıldığı dermoskopi görüntülerinden melanom tespiti çalışması %96.94 doğruluk oranı elde eden başarılı bir yaklaşım olmuştur [9]. Ben G. Weinstein ve diğerleri RetinaNet mimarisiyle RGB görüntü ve LiDAR verisi kullanılarak yarı gözetimli bir şekilde ağaç tacı tanıma çalışmasında %83 başarıya ulaşmışlardır. Buradaki başarı tek başına ve çevresi açık olan büyük ağaçlara aittir [10]. A. Wang ve diğerleri tarafından Rassal Orman, Destek Vektör Makineleri ve derin öğrenme mimarisiyle KSC (Kennedy Space Center) ve Indian Pines veri setlerine ait hiperspektral görüntüler kullanılarak %89 başarıya ulaşılmıştır. İlgili çalışmada en yüksek başarı, rassal orman yöntemi ve derin öğrenme algoritmasının birlikte kullanıldığı uygulama olmuştur [11]. A. Ostovar ve diğerlerinin VGG19 mimarisi kullanarak sayısal RGB görüntülerle Avrupa ladin türünün köklerindeki çürüme miktarının tespiti ve kademeli sınıflandırması amaçlı çalışmasında %90 doğruluk oranıyla hedefe ulaşılmıştır. Patojenik çürüğün tespitinde başarı yüksek iken çürük miktarlarının kendi içinde kademeli sınıflandırmasında başarının azaldığı sonucuna ulaşılmıştır [12]. 2019'daki C. Sothe ve diğerlerinin yaptığı başka bir çalışmada ise kendi geliştirdikleri evrişimli sinir ağı modelleriyle LiDAR verisi ve WorldView-2 görüntülerini kullanarak subtropikal orman bölgesinin arazi yapısı sınıflandırmışlardır. Ağaç, su ve toprak gibi sınıflarda hatasız sınıflandırma başarı elde edilmiş olmasına rağmen genel doğruluk oranı %92 olmuştur [13]. J. D. Sylvain ve diğerleri VGG16 mimarisini kullanarak sayısal hava kamerasıyla toplanan verileri kullanarak %76 - %95 arasında değişen doğruluk oranlarında başarılar elde etmişlerdir. Zamansal değişime dayalı hata modellerine bağlı olarak üretilen sonuçlar oldukça değişkenlik göstermiştir [14]. J. F. Abrams ve diğerleri U-Net mimarisine dayalı Habitat-Net'i geliştirerek orman taç örtüsü ve alt bitki örtüsü segmentasyonu amaçlı geliştirmişlerdir. %94 - %95 doğruluk oranıyla hızlı ve güvenilir segmentasyon işlemi yapılmıştır [15]. J. Liu ve diğerleri VGG16 mimarisini kullanarak dijital fotoğraf makinesi görüntülerini kullanarak 3 ağaç türü ve bunlara ait stok hacim kestirimi %96.03 doğrulukla ağaç türleri doğru saptanmış ve miktarları doğru öngörülmüştür [16]. Yine 2019 yılında Z. Zhang ve diğerleri

AMTNet mimarisiyle C-DCGAN ağında geliştirilmiş görüntüler kullanılarak tarımsal makineleri otomatik tanıma amaçlı uygulamaları %97.83 doğruluk oranıyla çevresel faktörlere rağmen yüksek bir başarı göstermiştir [17]. Tablo 1.1'de literatürde araştırılan çalışmalardan önemli görülen çalışmalardan bazılarından yılı, yazarı, veri seti ve yöntemi örnek olarak verilmiştir.

Tablo 1.1 Literatürde İncelenen Çalışmaların Tablosu

No.	Çalışma Başlığı	Yıl	Yazar	Veri seti	Yöntem	Kaynak No.
1	Forest Classification and Change Analysis Using Multi-Temporal Landsat TM Images over Huntington Wildlife Forest	2013	Manqi Li ve diğerleri	Landsat TM görüntüleri	Rassal Orman ve Destek Vektör Makineleri	[3]
2	Obje Tabanlı Yaklaşımda Makine Öğrenme Algoritmalarının Sınıflandırma Performansının Analizi	2015	T. Kavzoğlu ve diğerleri	Yüksek çözünürlüklü uydu görüntüleri	Destek Vektör Makineleri	[4]
3	Derin Öğrenme Yöntemleri ve Uygulamaları Hakkında Bir İnceleme	2017	Abdulkadir Şeker ve diğerleri	--	Derin öğrenme konusundaki gelişmeler	[5]
4	Derin Öğrenme Algoritmalarının Yaprak Sınıflandırma Başarımlarının Karşılaştırılması	2018	Ferdi Doğan ve İbrahim Türkoğlu	--	AlexNet, VGG16/19, ResNet50 ve GoogleNet	[7]
5	Evaluation of Different Machine Learning Algorithms for Scalable Classification of Tree Types and Tree Species Based on Sentinel-2 Data	2018	Mathias Wessel ve diğerleri	--	Destek Vektör Makineleriyle obje tabanlı görüntü analizi	[6]

Tablo 1.1 Literatürde İncelenen Çalışmaların Tablosu (devamı)

6	Derin Öğrenme ile Kalabalık Analizi Üzerine Detaylı Bir Araştırma	2018	Merve Ayyüce Kızrak ve Bülent Bolat	UCSD, WorldExpo10 ve UCF_CC_50	Derin öğrenme yaklaşımları kullanılmıştır	[8]
7	Derin Öğrenme Yöntemleriyle	2019	Oktay Yıldız	ISIC doku görüntüleri	C4Net	[9]

	Dermoskopi Görüntülerinden Melanom Tespiti: Kapsamlı Bir Çalışma					
8	Habitat-Net: Segmentation of Habitat Images Using Deep Learning	2019	Jesse F. Abrams, Anand Vashishtha ve diğerleri	Canopy closure and understory vegetation	UNet	[15]
9	Classification of Tree Species and Stock Volume Estimation in Ground Forest Images Using Deep Learning	2019	Jiazheng Liu ve diğerleri	RGB görüntü	VGG16	[16]
10	Deep Learning-Based Automatic Recognition Network of Agricultural Machinery Images	2019	Ziqiang Zhang ve diğerleri	C-DCGAN ağında geliştirilmiş görüntüler	AMTNet	[17]
11	Individual Tree-Crown Detection in RGB Imagery Using Semi-Supervised Deep Learning Neural Networks	2019	Ben G. Weinstein ve diğerleri	RGB görüntü ve LiDAR verisi	RetinaNet	[10]
12	A Comparison of Machine and Deep-Learning algorithms applied to multisource data for a subtropical forest area classification	2019	C. Sothe ve diğerleri	LiDAR verisi ve WorldView-2 görüntüleri	CNN	[13]
13	Detection and Classification of Root and Butt-Rot (RBR) in Stumps of Norway Spruce Using RGB Images and Machine Learning	2019	Ahmad Ostovar ve diğerleri	RGB görüntü	VGG19	[12]
14	Hyperspectral Image Classification Based on Convolutional Neural Network and Random Forest	2019	Aili Wang ve diğerleri	KSC (Kennedy Space Center) ve Indian pines	Rassal Orman ve Destek Vektör Makineleri	[11]

Tablo 1.1 Literatürde İncelenen Çalışmaların Tablosu (devamı)

15	Mapping Dead Forest Cover Using a Deep Convolutional Neural Network and Digital Aerial Photography	2019	Jean-Daniel Sylvain ve diğerleri	Sayısal hava kamerası	VGG16	[14]
16	Deep Residual Learning for Image Recognition	2015	Kaiming He ve diğerleri	ImageNet and COCO dataset	VGG16, ResNet, GoogLeNet, NIN	[33]

1.2 Tezin Amacı

Geçtiğimiz yıllarda oldukça popüler hale gelen ve gelişim gösteren derin öğrenme mimarileri görüntü işleme ve sınıflandırma başarıları sebebiyle Harita Mühendisliği, Uzaktan algılama ve fotogrametri alanında da tercih edilebilecek sistem olarak ön plana çıkmaktadır. Derin öğrenme sayesinde otomatikleştirilmiş ve kullanıcı müdahalesine gerek kalmaksızın nesne tanıma, obje çıkarımı ve sınıflandırma gibi işlemler; Harita Mühendisleri için zamandan ve iş gücünden tasarruf sağlayarak sonuç ürün doğruluğunu arttırması beklenen bir alan sunmaktadır. Bu sebeple, yeşil alanların tespit edilmesi amacıyla derin öğrenme algoritmalarının ne denli katkı sağlayabilecekleri, otomatik ve hızlı bir işlem mümkün olup/olmayacağının tespiti amacıyla bu çalışma gerçekleştirilmiştir.

1.3 Hipotez

Çalışma, “Davutpaşa kampüsünün ortofoto görüntüleri üzerinde, derin öğrenme algoritmaları kullanılarak yeşil alanların çıkarımı mümkündür.” hipotezi üzerine kurulmuştur.

2
DERİN ÖĞRENME

2.1 Tanım

Klasik YSA (Yapay Sinir Ağı)'nın temeli 1958'lere uzanır [18]. Tek nöronlu veya tek katmanlı olarak ifade edilen YSA, Yüksek Hızlı Grafik İşlem Birimi'nin (GPU (Graphics Processing Unit)) ve bilgisayar işlemcilerinin işlem güçleri geliştikçe yerini çok katmanlı sinir ağlarına bırakmıştır [19]. Bu sayede veriye ait öznitelikler, derin öğrenmeden güç alan makine tarafından otomatik olarak sınıflandırılıp öğrenilebilir olmuştur. Makine öğrenmesi, yoğun miktarda veriyle eğitilen algoritmalar sayesinde bilginin makine tarafından anlamlandırılması, öğrenilmesi ve sınıflandırılmasıdır. Derin öğrenme ise makine öğrenmesinin bir alt kümesidir [20]. Derin öğrenmeyi makine öğrenmesinden ayıran özellik YSA'ların kullanılıyor olmasıdır. Bu sayede öznitelikler otomatik çıkarılmaktadır. Evrişim katmanını takip eden aktivasyon fonksiyonu, kullanılan matematiksel model ile öğrenme işlemini doğrusal olmaktan uzaklaştırır. Doğrusal olmayan katmanlar, ileri ve geri yayılım fonksiyonlarıyla öğrenmenin başarısını arttıran derin öğrenme mimarilerinin temel unsurlarıdır.

Öğrenme işlemi bir görüntünün matrislerle ifade edilmesine dayanır. Veriler vektör ve matris gibi dizi ifadeleriyle gösterilir. Kenar ve görüntü özellikleri adeta bir filtre uygulanması işlemine benzer bir işlem sonucu değerler kazanarak anlamlandırılır. Bu gösterim veriyi temsil eden özelliklerin saptanmasını ve bunların çıkarımını sağlar. Bu süreç katman sayısı değiştirilerek veya gizli katmanlar eklenerek iyileştirilebilir.

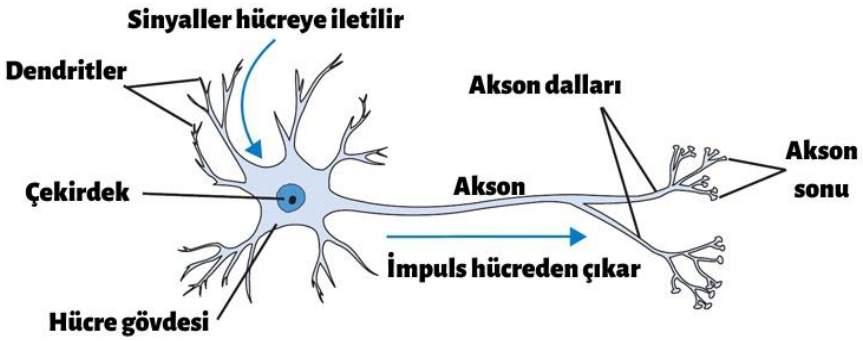
2.2 Tarihçe

Geçmişi 1940'lı yılların başına dayanan derin öğrenme algoritması, 1960'larda çok katmanlı sinir ağlarının ortaya atılıp 1980'lerin ikinci yarısından itibaren paralel dağıtık işlem olarak uygulanmaya başlaması ile değişim kazanmış ve bu tarih derin öğrenmenin temellerinin atıldığı zaman dilimi olarak kabul edilmiştir. "Derin

öğrenme" ifadesi YSA bağlamında ilk kez 2000 yılında Igor Aizenberg ve diğerleri tarafından tanıtılmıştır [22]. 2006'ya gelindiğinde Geoffrey Hinton çok katmanlı bir sinir ağını önce her katmanı bir Boltzmann makinesiyle eğitmiş ve daha sonra geri yayılım yöntemiyle hassas iyileştirmeler yapılabileceğini göstermişlerdir [21]. 2010'lardan itibaren yüksek kapasiteli bellekler, sanal veri havuzları ve hızlı grafik işlem birimlerinin gelişmesiyle popüler olmaya başlayarak farklı uygulama alanlarında kullanımı artmıştır. Hinton, Krizhevsky ve Sutskever 2012'de birbirine benzer mimariler tasarlamıştır. Çalışmalarında GPU'dan güç alarak "dropout" adı verilen makinenin ezber yapmasını engelleme amacıyla katmanlarda seyreltme yapmışlardır [22]. Seyreltme işleminin başarısı 2012'de ImageNet yarışmasında tescillenmiştir. Bu tarihten itibaren sosyal medya şirketleri, teknoloji devleri ve işlemci üreticileri derin öğrenme üzerine çalışmaya başlamışlardır.

2.3 Yapay Sinir Ağları

İnsan sinir hücreleri yani nöronlar: dendrit, hücre gövdesi ve aksondan oluşur. Dendritler, çevreden veya başka bir nörondan aldıkları sinyalleri (impuls) hücre gövdesine getirir (Şekil 2.1). Hücre gövdesindeki çekirdekler bu bilgileri işleyerek aksonlar yoluyla sinyal başka bir nöronun dendritine taşınır. Miyelin kılıf adı verilen yağ dokudan oluşan koruyucu kılıf, sinyallerin daha hızlı iletilmesini sağlar. Bir nöronun akson ucuyla diğer nöronun dendritini bağlayan noktaya ise sinaps denir [23].



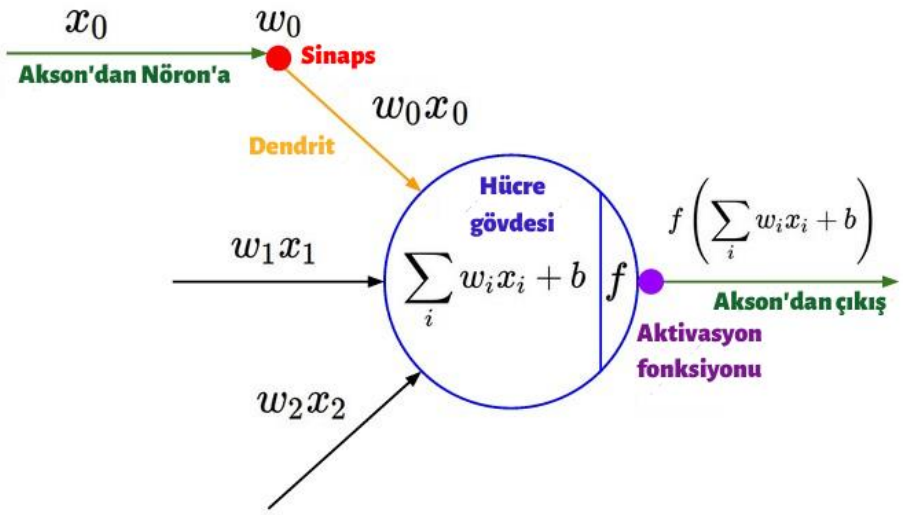
Şekil 2.1 İnsana ait bir sinir hücresi [23].

Organizma bağlamında bir tutumun soyutlanma işlemi, canlılara ait olan bir özelliğin yapay bir biçimde ifade şeklidir. YSA'nın (ANN (Artificial Neural Networks)) da ilham alındığı insan sinir hücrelerinden yola çıkarak makinelerin

öğrenmesini amaçlanır. İnsan öğrenme modeli Şekil 2.2'deki algoritma ile gösterilebilir.

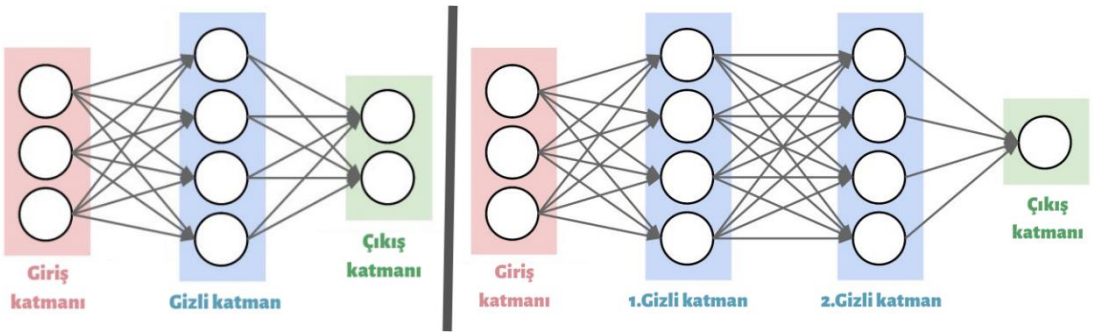
YSA en küçük parçası "Perseptron" adı verilen 'Algılayıcı'lardır. Buradan itibaren Perseptron yerine Algılayıcı terimi kullanılacaktır. Basit nöronlardan oluşan bir Algılayıcı ağı ise çok katmanlı ağları oluşturur. Algılayıcı'nın tanımı 1958'de Rosenblatt tarafından yapılmıştır. Algılayıcı, giriş ağırlıklarına göre doğrusal bir fonksiyon oluşturup çıktıyı doğrusal olmayan bir aktivasyon fonksiyonu kullanarak, çok değerli bir girişten tek bir çıktı olarak hesaplar. Frank Rosenblatt'in doğrusal olarak tanımladığı Algılayıcı denklemi [25]:

$$y = W \times x + b \tag{2.1}$$



Şekil 2.2 Yapay sinir ağı modeli [24].

Bu denklemde W ağırlık, x girdi vektörü ve b değeri bias'i ifade eder. Buradaki bias, modelin tahminlerindeki sistematik hatayı ifade eder. YSA ya da derin öğrenme modelindeki en iyi başarı değerine ulaşmak için en uygun W ve b parametresini seçmek oldukça önemlidir.



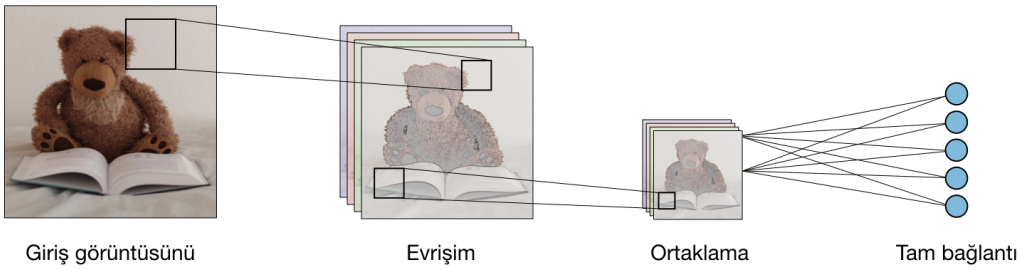
Şekil 2.3 Gizli katman sayısına göre YSA modelleri [24].

Şekil 2.3'te ilk kısımda yer alan YSA modeli, tek gizli katman içerir. Nöron sayıları hesap edilirken giriş katmanları dahil edilmez. Buna göre ilk modelde 4+2 olmak üzere 6 nöron vardır. İlk YSA modeli için öğrenilmesi gereken parametre sayısını hesaplamak istersek: $[3\times4]+[4\times2]=20$ ağırlık ve $4+2=6$ adet bias değerinden 26 adet parametreye ulaşılır. Şekil 2.3'te ikinci kısımda yer alan YSA modelinde ise $4+4+4+1=9$ nöron vardır. $[3\times4]+[4\times4]+[4\times1]=32$ ağırlık ve $4+4+1=9$ bias değeri olmak üzere 41 adet öğrenilmesi gereken parametre vardır. Çok katmanlı yapıya 1960 yılında Widrow ve Hoff'un ilk çalışmasıyla geçilmiştir. Modelde katmanlarda yer alan nöronların sistemin performansını etkilediği doğrudur fakat, nöronların doğrudan birbirleriyle ilişkileri yoktur. Bu sebeple katman sayısı arttıkça öğrenme performansı artmaz. Bu sistemi etkileyen birden çok hiper-parametre mevcuttur. Bunun yanında ileri ve geriye yayılım da başarıyı etkilemektedir [26].

2.4 Derin Öğrenme Ağlarının Katmanları

Evrişimli sinir ağları matematiksel formülleri temel alan katmanlardan oluşur. Geleneksel bir ESA (Evrişimli Sinir Ağı) mimarisinde (Şekil 2.4) üç katman türü vardır: evrişim, ortaklama (örnekleme/ havuzlama) ve tam bağlantı. İlk iki katmanda hiper-parametreler kullanarak ince ayar (fine-tuned) yapılabilir [27]. Hiper-parametre olarak ifade edilen parametreler, modeli tasarlayan kişinin inisiyatifine bırakılmış değiştirilip, güncelleştirilebilen değerlerdir. Bu değişkenleri: veri seti boyutu, mini-batch boyutu (işleme tabi tutulacak küme boyutu), öğrenme hızı (learning rate), momentum katsayısı, optimizasyon algoritması, eğitim tur (epoch) sayısı, başlangıç ağırlık (W) değerleri, aktivasyon fonksiyonu, seyreltme değeri, katman sayısı, gizli katmanlardaki nöron sayısı,

evrişimli sinir ağı filtre boyutları ve ortaklama yöntemi (maksimum, ortalama) olarak ifade edilebilir [28].

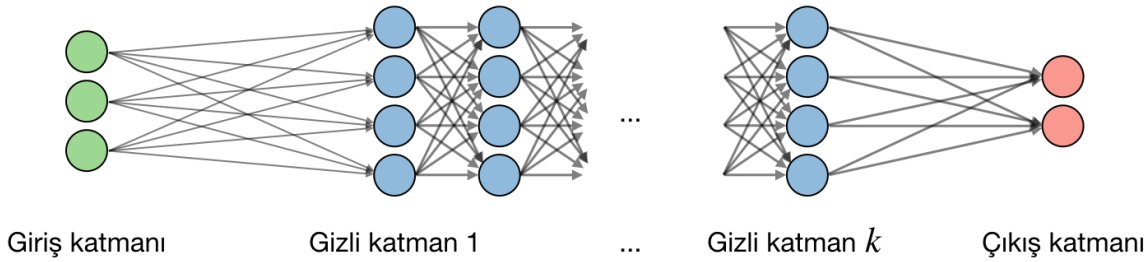


Şekil 2.4 ESA mimarisindeki katmanlar [27].

2.5 Katman Türleri

W, b, z sırasıyla ağırlık, bias (eğilim) ve ürünü temsil eder. Ağın i. sıradaki katmanda yer alan j. sıradaki gizli katmanların (Şekil 2.5) denklemi aşağıdaki gibidir [29].

$$z_j^{[i]} = w_j^{[i]T} x + b_j^{[i]} \tag{2.2}$$

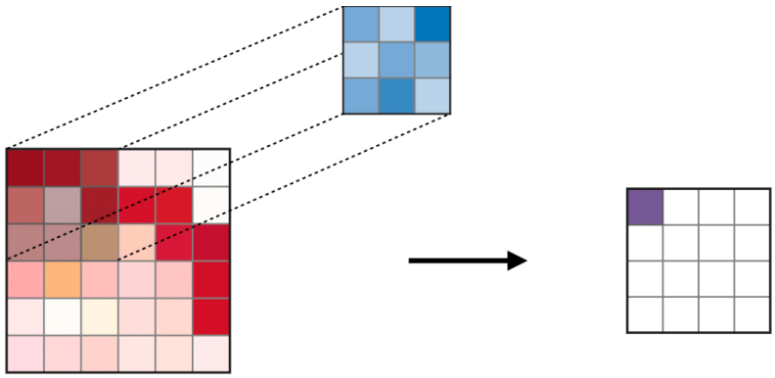


Şekil 2.5 YSA mimarisi [29].

2.5.1 Evrişim Katmanı

Evrişim (Convolution) işlemini gerçekleştiren filtre (Şekil 2.6), I girdisini F boyutlu ve S adım aralıklı (filtre, girdi matrisinde S adım kaydırılmak suretiyle) tarar. Elde edilen O çıktısı, öznitelik haritası olarak adlandırılır [27]. Evrişim

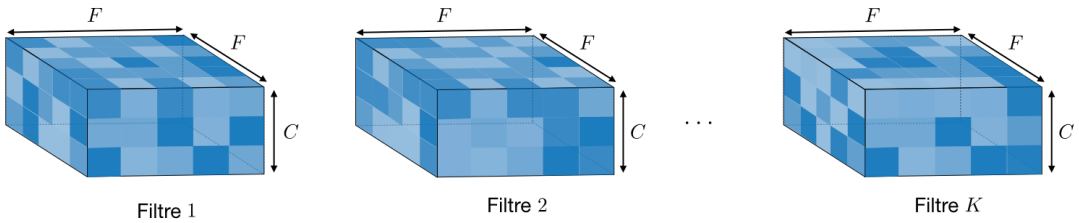
katmanının hiper-parametreleri ve aktifleştirme fonksiyonları alt başlıklar olarak sıralanmıştır.



Şekil 2.6 Girdiye filtre uygulanarak yapılan evrişim işlemi [27].

2.5.1.1 Filtre Boyutları

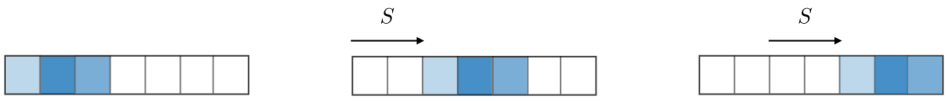
$I \times I \times C$ boyutlu bir girdi matris için C kanallı ve $F \times F$ boyutlu bir filtre (Şekil 2.7) uygulanırken $O \times O \times 1$ boyutlu bir öznitelik haritası üretir (K tane filtre uygulanırsa 1 yerine K yazılır) [27].



Şekil 2.7 Filtrelerin yükseklik, genişlik ve kanal sayısı gösteren model [27].

2.5.1.2 Adım Aralığı

Evrişim ve ortaklama işleminde filtre gezdirilirken (Şekil 2.8) her bir işlemden sonra S adımı (stride) filtre penceresinin kaç piksel hareket edeceğini ifade eder [27].

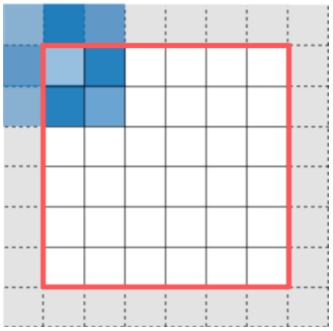


Şekil 2.8 Filtrenin adım aralığı [27].

2.5.1.3 Sıfır Dolgulama

Zero padding olarak da bilinir. Girdi genellikle çıktı katmanına ulaşana kadar boyut kaybeder, sisteme yük getirmeyecek 0 değerleri matrise eklenirse daha çok

bilgi depolanmış ve sonraki katmanlara aktarılmış olur. Şekil 2.9'da kırmızı hattın dışında kalan gri pikseller sıfır doldurmayla eklenmiş alanları, mavi matris de evrişim işleminde gezdirilen filtreyi temsil eder [27].



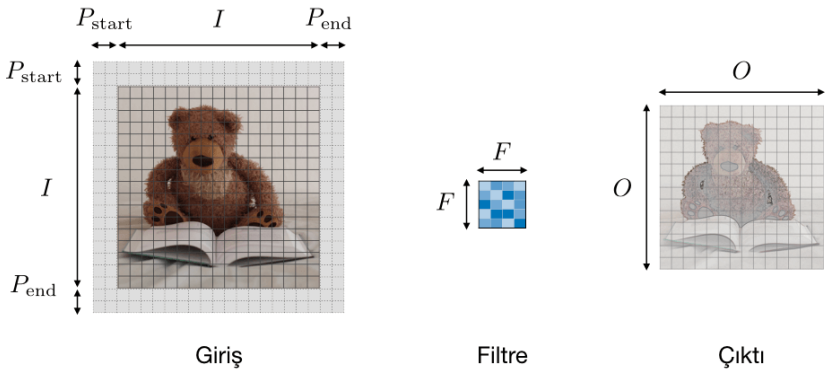
Şekil 2.9 Sıfır dolgulama [27].

2.5.1.4 Evrişim Katmanı Çıktı Boyut Hesabı

Girdinin boyutları I x I, P sıfır dolgulama miktarı (çoğunlukla $P_{start} = P_{end} \triangleq P$), S adım aralığı, F filtre boyutu, O değeri ise öznitelik haritasının çıktı boyutunu (Şekil 2.10) temsil eder [27].

Nöron sayısı hesap edilirken I değerinin yerini W, O değerini ise N alır [27].

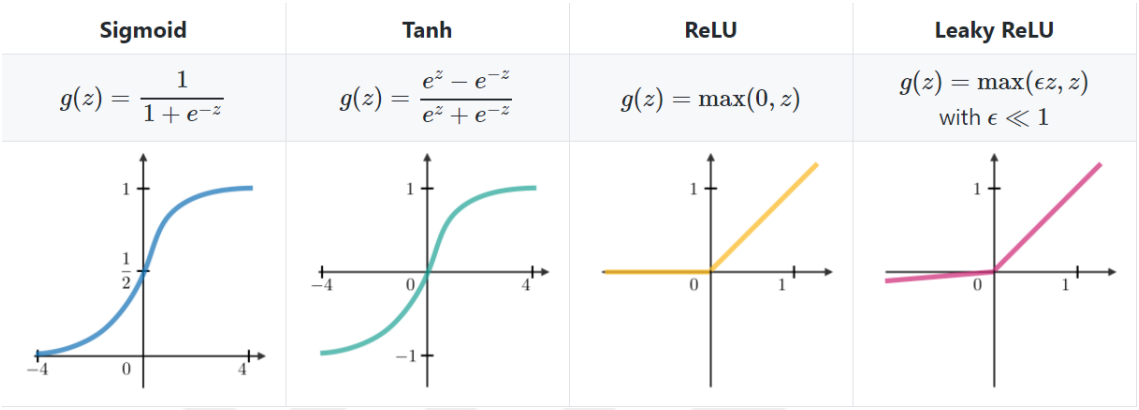
$$O = \frac{I - F + P_{start} + P_{end}}{S} + 1 \tag{2.3}$$



Şekil 2.10 Evrişim katmanı girdi ve çıktısının matris boyutları [27].

2.5.1.5 Aktivasyon (Etkinleştirme) Fonksiyonları

Aktifleştirme fonksiyonları, gizli katmanların sonunda kullanılarak modele doğrusal olmamayı sağlarken ağın öğrenmesini amaçlar. Sigmoid, Tanh, ReLU ve Leaky ReLU en yaygın kullanılan aktivasyon fonksiyonlarıdır (Şekil 2.11). Sigmoid fonksiyonu 0 ile 1 arasında değerler alırken, Tanh fonksiyonu -1 ile 1 arasında değerler alır. ReLU fonksiyonları ise pozitif değerler için doğrusal bir artış gösterir.



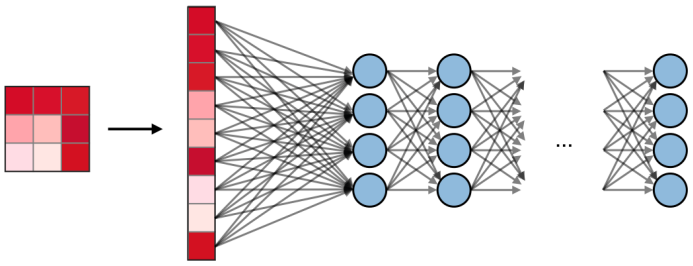
Şekil 2.11 Yaygın kullanılan aktivasyon fonksiyonları [29].

2.5.2 Ortaklama (Havuzlama/Örnekleme) Katmanı

Ortaklama (pooling) katmanı, genellikle bir evrişim işlemi takip eden örnekleme işlemidir. Burada amaç özniteliklerin algılanarak korunmasıdır. İşlem sırasında evrişim işlemi görmüş matriste gezdirilen filtre, hücrelere düşen en büyük değeri tutuyorsa maksimum ortaklama, bu değerlerin ortalamasını alma prensibine dayanıyorsa ortalama ortaklama olarak isimlendirilir. Maksimum ortaklama, algılanan özelllikleri koruduğu için en çok kullanılan ortaklama yöntemidir. Ortalama ortaklama, LeNet ve NIN mimarisinde kullanılmış olup boyut azaltarak örneklenmiş öznitelik haritası çıkarılır [27].

2.5.3 Tam Bağlantı Katmanı

Tam bağlantı (full connected) katmanı genellikle ESA mimarisinin son katmanında bulunur. Her girişin tüm nöronlara bağlı olduğu bir ağ düzeninde çalışır (Şekil 2.12). Sınıf skorları gibi hedefleri optimize etmek için kullanılabilir [27].



Şekil 2.12 Tam bağlantı katmanı modeli [27].

2.5.3.1 Eşikleyici

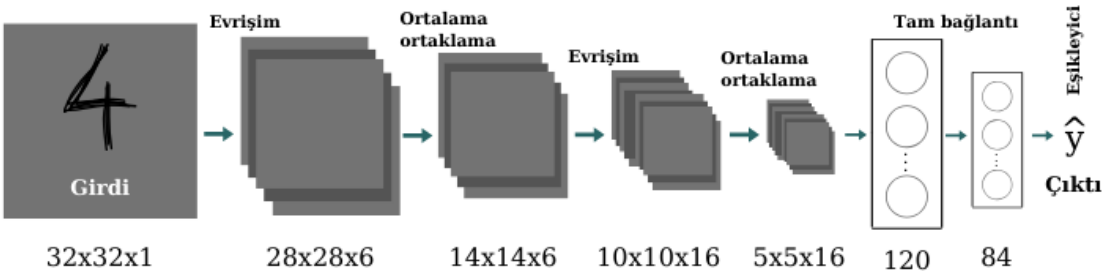
Eşikleyici (softmax) fonksiyonu esasen bir aktivasyon fonksiyonu türüdür. $x \in R^n$ skorlarının bir vektörünü girdi olarak alıp, $p \in R^n$ çıktılarının olasılık vektörünü oluşturan, sınıfların arasındaki farkların yorumlanmasını sağlayan lojistik bir fonksiyondur. Denklem (2.4) olarak tanımlanır [27].

$$p_i = \frac{e^{X_i}}{\sum_{j=1}^{n} e^{X_i}} \qquad p = \begin{pmatrix} p_1 \\ \vdots \\ p_n \end{pmatrix} \tag{2.4}$$

2.6 Evrişimli Sinir Ağı Algoritmaları

Derin öğrenme kendi içinde denetimli, yarı denetimli ve denetimsiz öğrenme olmak üzere kollara ayrılmaktadır. ESA (CNN (Convolutional Neural Networks)) denetimli öğrenme yöntemlerinden bir tanesidir. Yapılan çalışmalarda obje tanıma, çıkarma ve sınıflandırma görevlerinde başarısını kanıtlamış olması bu derin öğrenme yönteminin ön plana çıkmasını sağlamıştır. Bu doğrultuda incelenebilecek yaygın olarak kullanılan evrişimli sinir ağı algoritmaları mevcuttur.

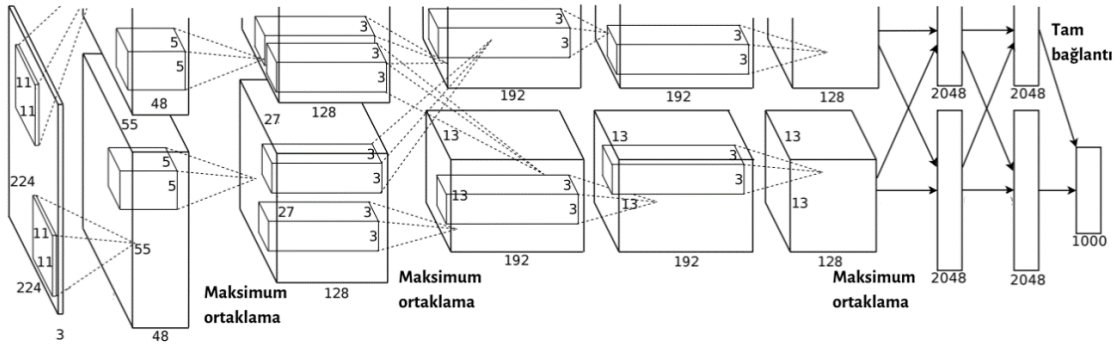
LeNet-5, ilk başarılı ESA modelidir. LeCun ve diğerleri 1998 yılında banka çekleri ve posta numaralarındaki rakamların otomatik olarak tanınması için MNIST (Modified National Institute of Standards and Technology) veri kümesi kullanılarak geliştirilmiştir. Ortalama ortaklama metodu ile birlikte aktivasyon fonksiyonu olarak da Sigmoid ve Tanh kullanılmıştır [30].



Şekil 2.13 LeNet-5 mimarisi.

Şekil 2.13'teki modelde ilk iki çarpan giriş görüntüsünün boyutunu, üçüncü çarpan ise giriş görüntüsünün kanal sayısı değerini verir. Ağ boyunca görüntü boyutu azalırken derinlik değeri artar.

Şekil 2.14'te gösterilen AlexNet, evrişimli sinir ağı modellerinin yeniden popüler hale gelmesini sağlayan çalışma olmuştur. 2012 yılında Alex Krizhevsky, Ilya Sutskever ve Geoffrey Hinton tarafından geliştirilmiş olup temelde LeNet modeline benzemektedir. 5 evrişim ve 3 maksimum ortaklama katmanı vardır. Aktivasyon fonksiyonu olarak da ReLU (Rectified Linear Unit) kullanılmıştır [31].



Şekil 2.14 AlexNet mimarisi [31].

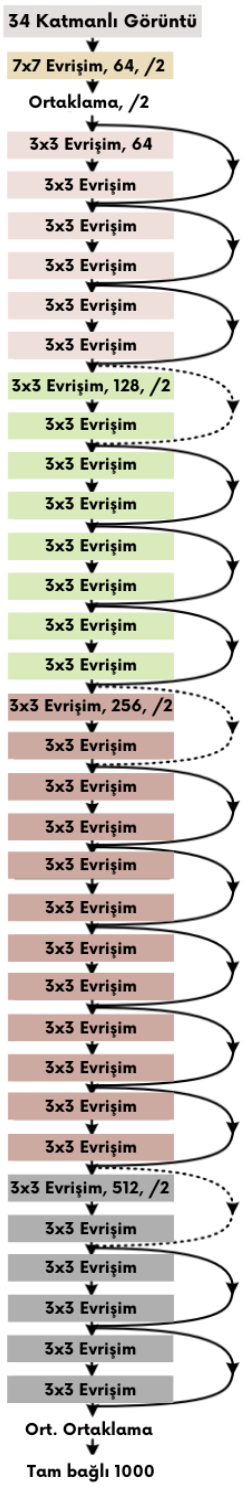
Şekil 2.15'te verilen VGG16 mimarisinin, önceki mimarilerden farkı peş peşe yapılan evrişim işlemidir. Diğer modellerde genel itibariyle bir evrişimden sonra bir ortaklama işlemi yer alırken burada bir evrişim katmanından sonra bir veya birden çok evrişim katmanı daha yer alır. İki tam bağlantı katmanında $6\times6\times256=9216$ nöronlu öznitelik vektörü elde edilir. Çıktı 1000 sınıflı eşikleyici ile sınıflandırılır. Matris yükseklik ve genişlik boyutları azalırken derinlik boyut değeri artmaktadır [32].



Şekil 2.15 VGG16 mimarisi.

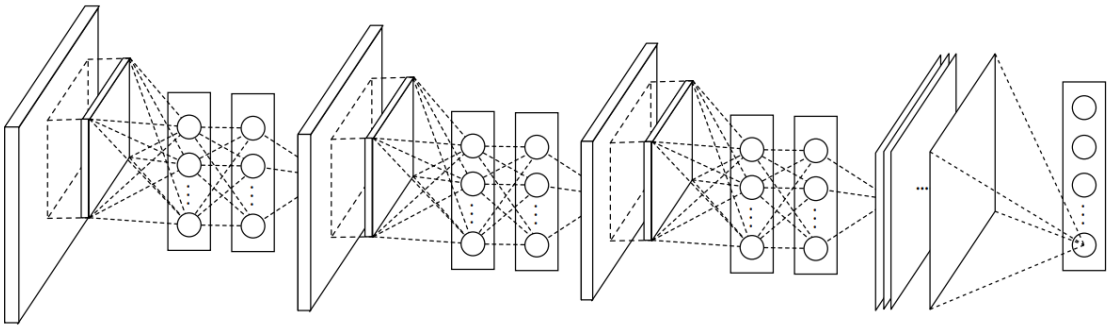
ResNet (Residual Network), Artık Ağ mimarisi olarak isimlendirilir [27]. ResNet ESA modelinde, peş peşe gelen katmanların sırayla çalışması yerine, bir katmanı es geçmek suretiyle işlem yapılır [33]. Bu yönüyle diğer modellerden ayrılır. Ancak katman atlarken yeni katman hesabında ağırlık değerlerinin sıfıra eşit olduğu durumda türev sıfıra eşit olur (vanishing gradient). Bu durumun önüne geçmek için artık öğrenme (residual learning) işlemi yapılır. Artık (residual) değerden öğrenme işlemi besleme artık değerinin, bir sonraki katmana (artık bloğa) eklenmesidir. Bu modelde ReLU aktivasyon fonksiyonu kullanılır. Bu durum derin bir model oluşmasını sağlamış olur ve ağın daha hızlı eğitilmesine imkan tanır [33]. Bu mimari sayesinde Eğitim hataları azaltılmış olur [27]. Şekil 2.16'da gösterilen mimaride giriş görüntüsünü 7x7'lik Evrişim katmanını

Ortaklama katmanı izler. Bu katmanı da peş peşe 3x3'lük Evrişim katmanıyla boyut arttırımı izler. Ardından Ortalama Ortaklama katmanıyla Tam Bağlantı sağlanır [33].



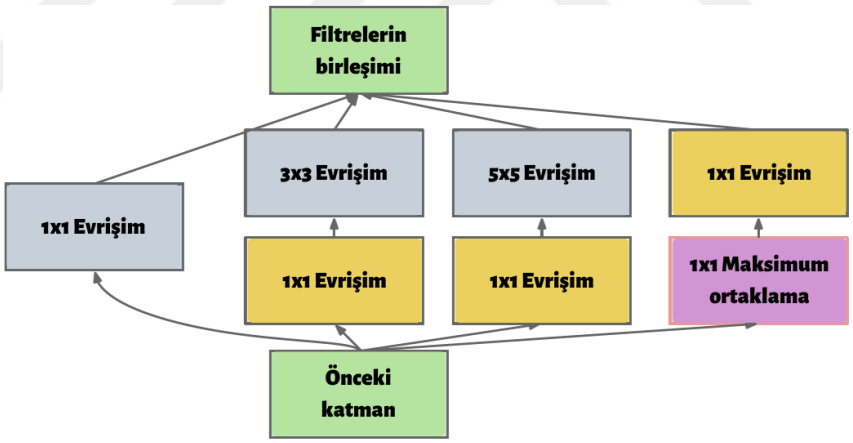
Şekil 2.16 ResNet mimarisi [33].

GoogLeNet'in temeli Min Lin ve diğerlerinin 2014'teki NIN (Network in Network) çalışmasına dayanır (Şekil 2.17). Mimaride üç tane çok katmanlı algılayıcı içeren evrişim işlemleri ve ortalama ortaklama işlemi vardır [34].

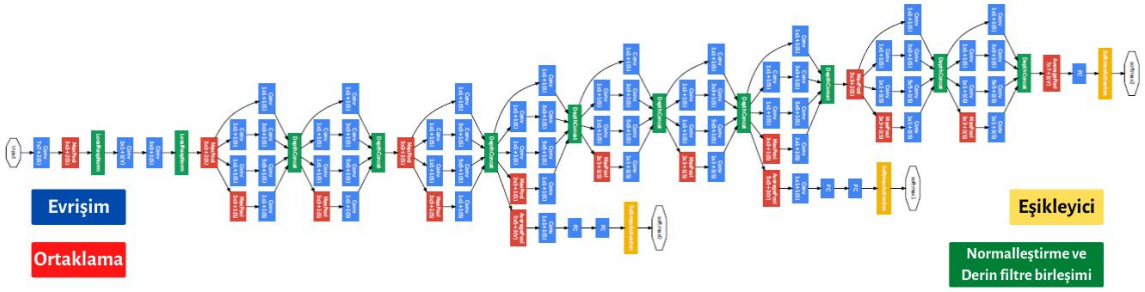


Şekil 2.17 Network in Network (NIN) mimarisi [34]

2015'te Google, NIN mimarisini kullanarak ve aynı zamanda LeNet'e de atıfta bulunarak geliştirdiği modele GoogLeNet (Şekil 2.19) ismini vermiştir. İç içe ağlardan oluşan yapı modeline ise Inception modülü (Şekil 2.18) adını vermiştir [35].



Şekil 2.18 GoogLeNet inception modülü.



Şekil 2.19 GoogLeNet inception ağı [35].

2.7 Veri İşleme

Derin Öğrenmedeki veri işleme (data processing) parametreleri, veri arttırma (data augmentation) ve küme normalleştirme (batch normalization) dir.

2.7.1 Veri Arttırma

Derin öğrenme algoritmaları, eğitim aşamasında çok miktarda veriye ihtiyaç duyar. Bu doğrultuda var olan veriyi arttırmak, makine öğrenimi açısından yarar sağlar. Çevirme, döndürme, rastgele kırpma, görüntü bant kombinasyonlarıyla renkleri değiştirme, gürültü ekleme, bilgi kaybına neden olacak bazı pikselleri silme işlemi ve kontrast değişimi gibi veri arttırma teknikleri kullanılabilir [36].

2.7.2 Küme Normalleştirme

γ, β hiper-parametrelerinin, $\{x_i\}$ kümesini normalleştiren bir adımıdır. Bu işlem genellikle tam bağlantılı bir katmandan sonra ve doğrusal olmayan bir katmandan önce yapılır. Bu işlem öğrenme oranını arttırmak ve başlangıç durumuna olan bağımlılığı azaltmayı amaçlar. μ_B normalleştirmek istenen kümenin ortalamasını, σ_B^2 ise düzeltmek istenen kümenin varyansını ifade eder [36, 29].

$$X_i \leftarrow \gamma \frac{X_i - \mu_B}{\sqrt{\sigma_B^2 + \epsilon}} + \beta \tag{3.1}$$

2.8 Yapay Sinir Ağı Eğitimi

YSA eğitim parametrelerini eğitim tur sayısı, gradyan azalma algoritması, yitim fonksiyonu ve çapraz-entropi kaybı olarak kategorize etmek mümkündür.

2.8.1 Eğitim Tur Sayısı

Eğitim tur sayısı, epok veya dönem olarak da ifade edilir. Bir model ilk kez eğitildiğinde modele ait ağırlıklar oluşur. Peş peşe eğitim işlemi yapılarak en uygun ağırlık değerlerine ulaşıncaya dek ağrılıkları güncellemeyi ifade eden hiper-parametredir.

2.8.2 Küçük Küme Gradyan Azalma Algoritması

Mini-Batch Gradient Descent olarak da ifade edilir. Veri setindeki verileri aynı anda işlemek öğrenme, zaman ve bellek açısından maliyetli bir iştir. Küçük küme hiper-parametresi sayesinde bu problemi çözmek daha kolay hale gelir. Veriler kümelere ayırılarak optimizasyon sağlanır. Bu sayede öğrenme işlemi bu küçük kümelerde gerçekleşir [28].

2.8.3 Yitim Fonksiyonu

L yitim fonksiyonu (loss function), bir modelin göstereceği başarıyı ölçmek için y gerçek çıktıların, z model çıktıları tarafından tahmin doğruluğunun değerlendirilmesi için kullanılır [36].

2.8.4 Çapraz-Entropi Kaybı

Sınıflandırıcı olarak kullanılan bir yitim fonksiyonu türüdür. Çapraz-entropi kaybı, YSA' da yitim fonksiyonunu $L(z, y)$ şeklinde optimizasyon problemine dönüştürür.

$$L(z, y) = -[\, y \log(z) + (1-y) \log(1-z)] \tag{3.2}$$

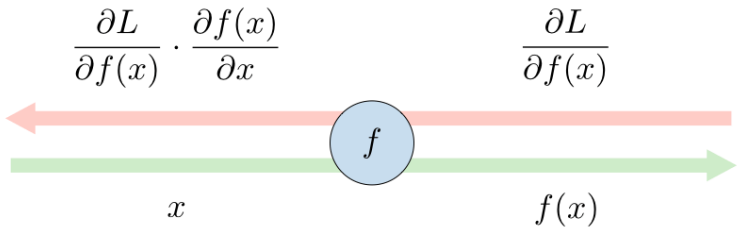
2.9 Optimum Ağırlıkların Bulunması

Veri seti en başta üçe ayrılır: eğitim, doğrulama ve modelin görmediği test veri kümesi. Eğitim ve test gruplarının karşılaştırılması sonucu yitim fonksiyonuna ulaşılmaktadır. Yitim fonksiyonunda kullandığımız test grubu, veriyi ifade eder. Eğitim grubu da tüm veriyle çalışarak zaman kaybını yerine, veriyi en doğru temsil edecek temsil grubunu ifade eder. Ağırlık ortalamaları alınarak bunlar hesaplanır.

2.9.1 Geri Yayılım Algoritması

Geri yayılım algoritması, ileri beslemeli (Şekil 2.20) bir sinir ağının eğitmek için kullanılır. Ağırlık güncellemek için sağdan itibaren her bir ağırlık için türev hesabı yapılır. Denklem (3.3)'e göre göre hesap edilir.

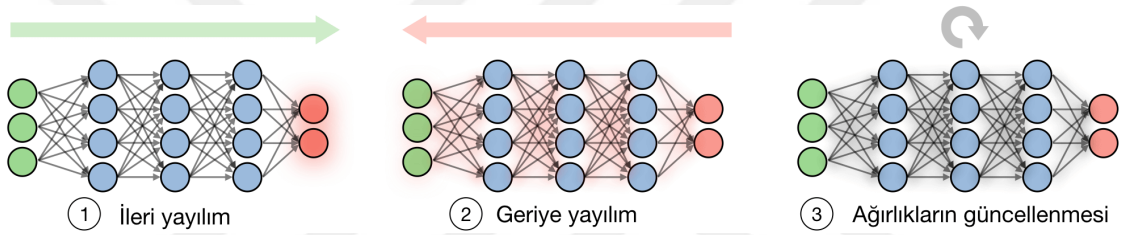
$$w \longleftarrow w - \alpha \frac{\partial L(z,y)}{\partial w} \tag{3.3}$$



Şekil 2.20 İleri ve geri yayılım yönlerini gösteren model [36].

2.9.2 Ağırlık Güncellenmesi

İlk adımda seçilen bir küme, eğitim verisiyle ileriye yayılımla hesaplanır. Yitim fonksiyonuna ulaşmak için her bir ağırlık için geriye yayılım işlemi yapılır. Gradyan azalma algoritmaları kullanarak ağırlık güncellemeleri yapılır (Şekil 2.21).



Şekil 2.21 Ağırlıkların güncelleme aşamaları [36].

2.9.3 Başlangıç Ağırlığının Ayarlanması

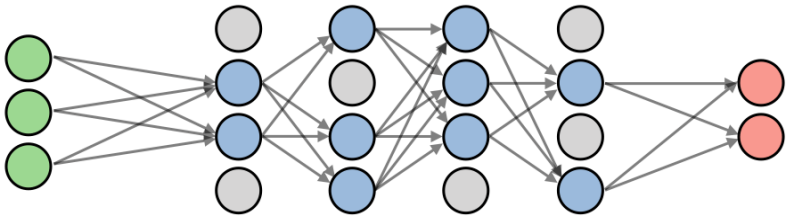
Başlangıç ağırlığı için Xavier başlangıcı ve transfer öğrenme yöntemleri mevcuttur. Başlangıç ağırlığı da bir hiper-parametredir. Xavier başlangıcındaki ağırlıklar, mimariye özgü özelliklere sahip ilk ağırlıklardır. Transfer öğrenme yönteminde ise büyük veri setleri üzerinde önceden eğitilmiş ağırlıklardan yararlanılır. Çünkü bir derin öğrenme modelinin eğitilmesi çok fazla veri ve uzun zaman gerektirir. Eğitimin boyutuna göre üç ayrı metot mevcuttur. Alt seviye bir eğitim yapılacaksa sadece çıktı katmanındaki ağırlıklar eğitilir. Orta seviye bir eğitim yapılacaksa çıktı katmanıyla birlikte son katmanlardan birkaçı eğitilir. Eğer üst seviyeli bir eğitim hedefleniyorsa önceki eğitimden elde edilen tüm ağırlıklar tekrar eğitilebilir [36].

2.10 Ağırlıkları Düzenleme

Düzenleme (regularization), makinenin gereksiz öğrenme yapmasına engel olmak veya makine eğitiminin duracağı yeri tespit etmek için kullanılır. Bu sayede gereksiz ilişkilendirmelere engel olunarak işlem yükü azaltılabilir.

2.10.1 Seyreltme

Seyreltme (dropout) tekniği sinir ağındaki $p > 0$ olasılıklı nöronları silerek makinenin eğitim verisine tamamen bağlı kalmasını engeller. Bu sayede makinenin ezber yapmasına izin verilmez. Şekil 2.22'de gri renkte gösterilen nöronlar, seyreltilerek düzenli hale getirilir [36].



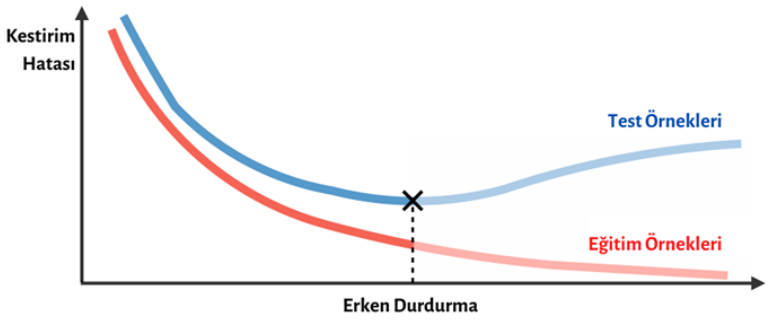
Şekil 2.22 Seyreltme işlemi [36].

2.10.2 Ağırlıkları Düzenleme

Ağırlık düzenleme, modelin eğitim setine uygun olup olmadığının araştırması amacıyla ağırlıkların denetime tabi tutulması işlemidir. Model ağırlıklarında düzenleme sağlayan temel teknikler: LASSO, Ridge ve ElasticNet'tir. Bunlar değişken seçimi yaparak katsayıları daha küçük değerlere getirirler [36].

2.10.3 Erken Durdurma

Erken durdurma (early stopping), eğitimin biteceği noktayı işaret eder. Kestirim hatasının en az olduğu yerin tespit edilmesini sağlar. Şekil 2.23'te kestirim hatasının en az olduğu nokta gösterilmiştir.



Şekil 2.23 Erken durdurma tekniğini gösteren uyarlanmış grafik [36].

2.11 Öğrenme Oranı

Öğrenme oranı (learning rate), genellikle α (nadiren η) ile gösterilir. Yakınsamayı optimize eden, yani ağırlıkların güncellendiğini hızı belirleyen bir hiper-parametredir [36]. SGD (Stokastik Gradyan Azalma Algoritması) bir Gradyan Azalma Varyantıdır. Momentum, RMSprop, Adam, Adadelta ve Adagrad ise Gradyan Azalma Optimizasyon Algoritmalarıdır [37]. Genellikle tercih edilen optimizasyon yöntemi Adam'dır. Bu yöntemler modelin eğitimi esnasında öğrenme oranın değişmesine izin vererek eğitim süresinin kısaltılmasına veya matematiksel işlemlerin iyileştirmesine olanak sağlar [36].

Adam optimizasyon yöntemine ait w (ağırlık) ve b (bias) güncelleme denklemleri aşağıda verilmiştir.

$$w - \alpha \frac{v_{dw}}{\sqrt{s_{dw}} + \in} \qquad b \leftarrow b - \alpha \frac{v_{db}}{\sqrt{s_{db}} + \in} \tag{3.4}$$

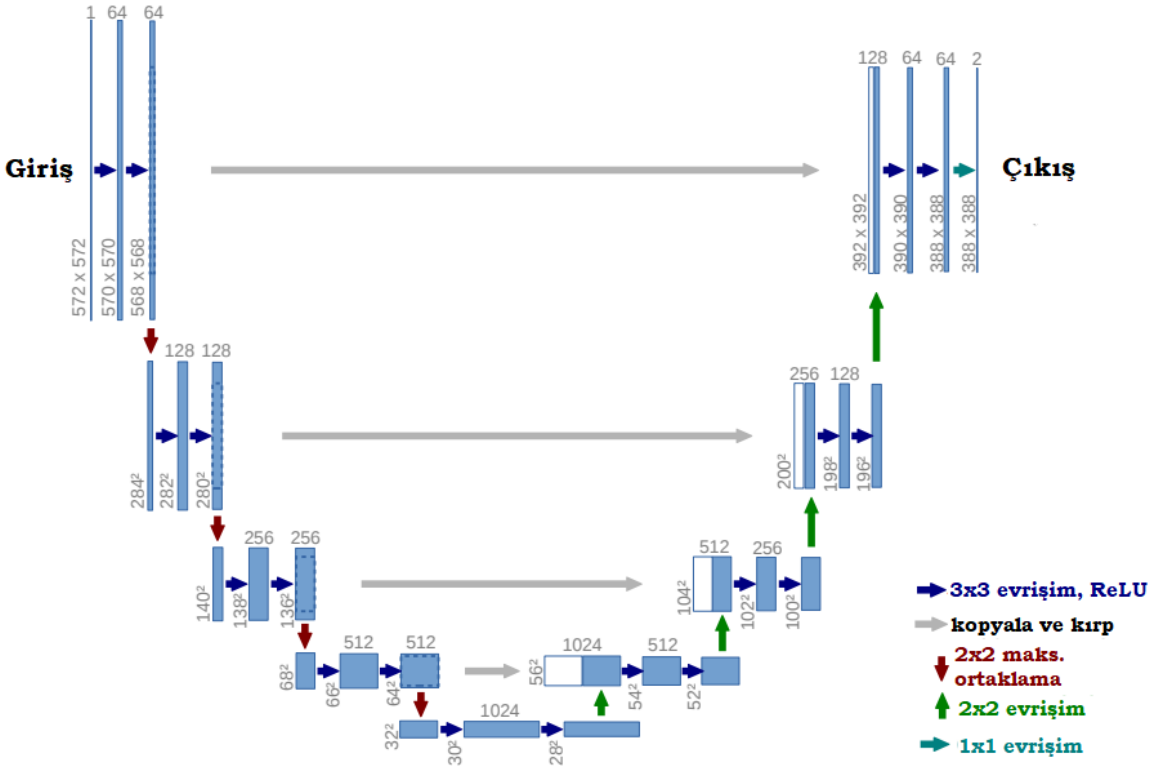
2.12 Uygulamada Kullanılan Algoritmalar

Evrişimli Sinir Ağı Algoritmaları, Derin Öğrenme'nin kronolojik gelişimine genel bir bakış olarak değerlendirilebilir. Bu başlık altında yer alan mimariler ise sadece uygulamada kullanılmış algoritmalardır. Daha önceki bölümlerde ifade edilen mimariler irdelendiğinde medikal veya coğrafi görüntü analiz etmek için U-Net mimarilerinin uygun olabileceği görülmüştür. Uygulamada kullanılan 5 adet mimari kendi içinde kıyaslanarak hipoteze en uygun algoritmanın tespiti hedeflenmektedir.

2.12.1 U-Net Mimarisi

Şekil 2.24'te U harfine benzeyen bu mimaride ilk kısım Kodlayıcı, diğer kısım ise Kod Çözücü katman olarak isimlendirilir. Her bir seviyede 3x3 iki adet Evrişim işlemi ve Evrişim işlemini takip eden ReLU aktivasyon fonksiyonu yer alır. Bir sonraki seviyeye geçerken Ortaklama katmanı ile boyut azaltılır. Kodlayıcı katmandan Kod Çözücü katmana geçerken veri kaybının engellenmesi için öznitelik haritaları kopyalanır. Kod Çözücü katmanda Evrişim ve aktivasyon

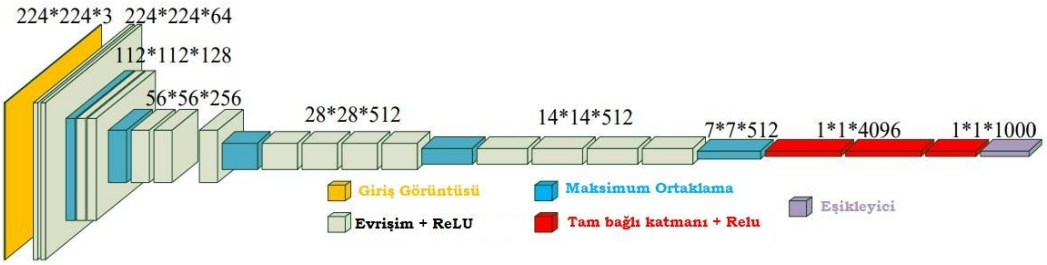
fonksiyonu kullanılarak yukarı örnekleme işlemi yapılır. Bu işlem esnasında her bir piksele ait olasılık değeri üretilir [38]. Yüksek çözünürlüklü öznitelikler ile örneklenen çıkış verisi birleştirilerek daha az kayıplı bir çıktı hedeflenir. Girdi olarak verilen giriş görüntüsü, çıktıda bölütlenmiş olarak elde edilir. Bu mimari diğer algoritmalardan tam bağlantı katmanına sahip olmaması yönüyle ayrılır.



Şekil 2.24 U-Net mimarisi [39].

2.12.2 VGG19 Mimarisi

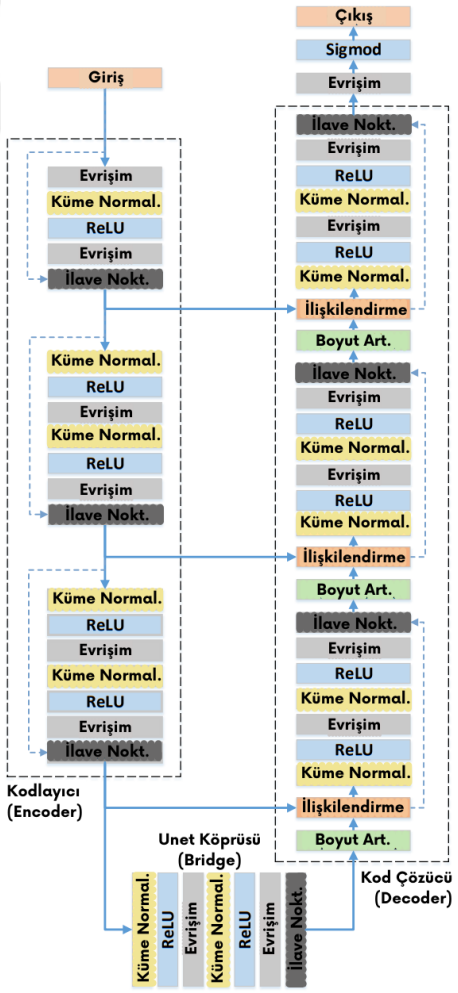
Evrişimli Sinir Ağları başlığı altında VGG16 mimarisi verilmiştir. VGG16 ile VGG19 mimarisi, peş peşe gelen Evrişim katmanları ve bunu izleyen Maksimum ortaklama katmanı yönünden birbirine benzemektedir. Şekil 2.25'te görüldüğü üzere Maksimum ortaklama katmanıyla boyut azaltma işlemi yapılır. VGG16 mimarisine oldukça benzer olan VGG19 mimarisi, birkaç Evrişim katmanı ve tam bağlı katman ile VGG16 mimarisinden ayrılır.



Şekil 2.25 VGG19 mimarisi [40].

2.12.3 ResUNet Mimarisi

Evrişimli Sinir Ağları başlığı altında ResNet mimarisi verilmiştir. ResNet mimarisinde kullanılan Artık Öğrenme metodu bu mimaride de kullanılmaktadır. ResUNet mimarisi ise U-Net'e gömülü bir ResNet mimarisi şeklinde ifade edilebilir. Şekil 2.26'da verilen mimaride kodlayıcı ve kod çözücü katmanları U-Net köprüsüyle birbirine bağlanır.

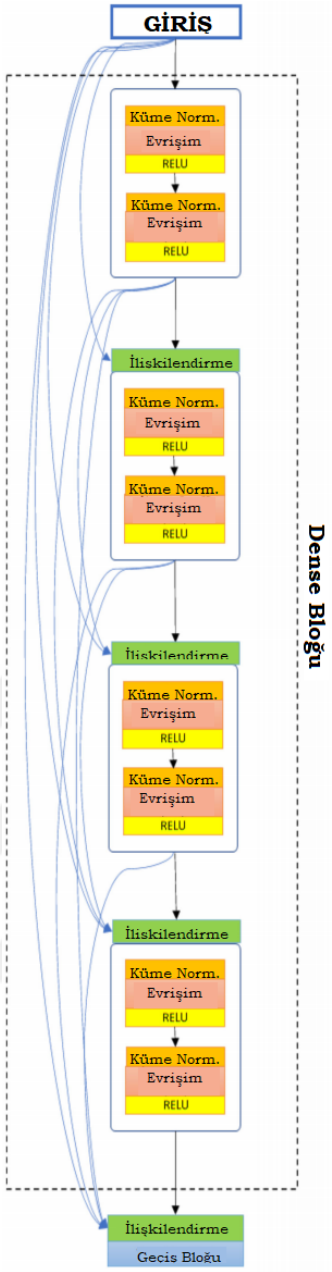


Şekil 2.26 ResUNet mimarisi [41].

Artık Öğrenme metodunun daha derin ağ oluşturma yeteneği vardır. Bunun yanında zengin atlama ilişkileri (ilave noktaları/ilişkilendirme) farklı katmanlar arasında daha iyi bir bilgi akışı sağlar. Ayrıca eğitim aşamasında daha başarılı bir geri yayılım imkanı sunar [41].

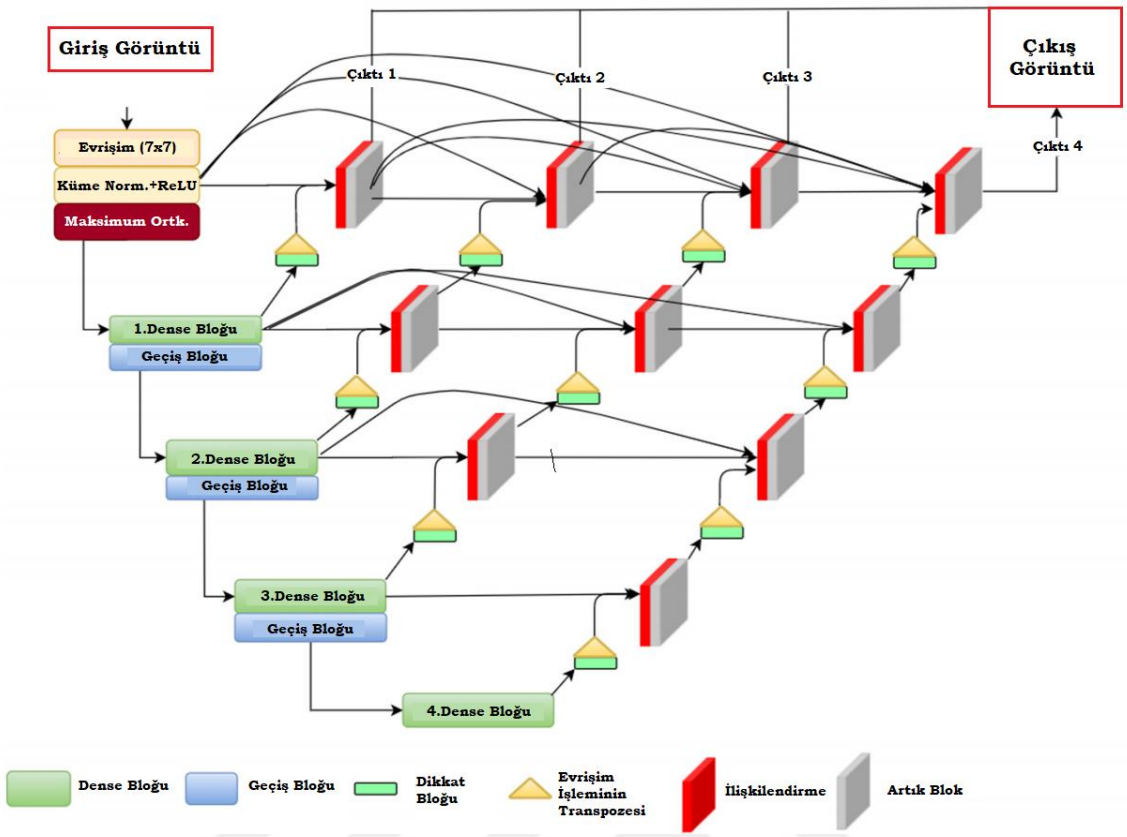
2.12.4 DenseNet ve DenseUNet Mimarisi

DenseNet'in açılımı Densely Connected Convolutional Networks'tür. Türkçe karşılığı 'Yoğun Bağlantılı Evrişimli Ağlar' şeklinde ifade edilebilir. Bu mimaride tüm katmanlar yoğun bir biçimde birbirine bağlanır. Her katman kendinden önceki tüm katmanları girdi alır ve kendinden sonraki tüm katmanlara iletir.



Şekil 2.27 Dense bloğu [42].

Şekil 2.27'de yer alan Dense Bloğu: Küme normalleştirmesi, Evrişim katmanı ve ReLU aktivasyon fonksiyonun peş peşe yer aldığı bir mimaridir. Bu katmanlar tekrar etmektedir ve tüm katmanlar yoğun bir biçimde birbirine bağlıdır.



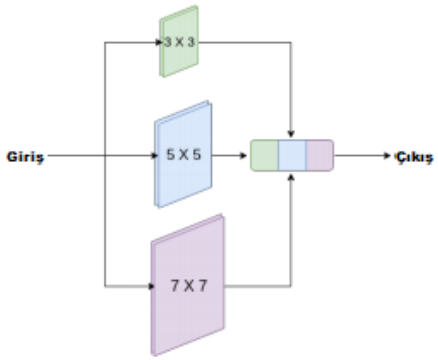
Şekil 2.28 DenseUNet mimarisi [42].

Şekil 2.28'de yer alan DenseUNet mimarisi, temel yapı taşını DenseNet'teki Dense bloklarından alan ve geçiş bloklarıyla U-Net mimarisi formuna sahip yoğun derin bir mimaridir. Diğer mimarilerden farklı olarak bu mimaride yer alan dikkat bloğu (attention block), anlamsal bölütlemede piksel bazlı bir tahmin modeli olarak kullanılmıştır [43].

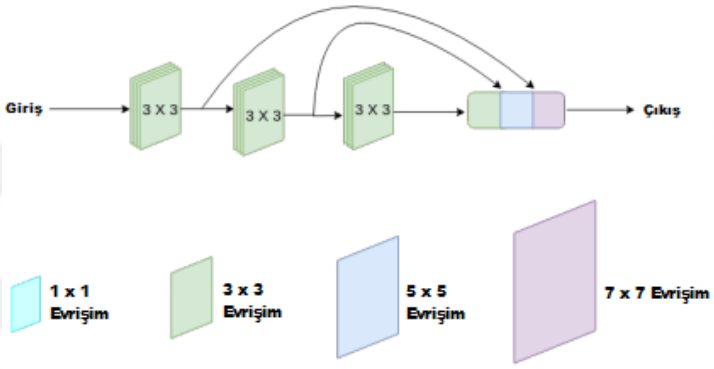
2.12.5 MultiResUNet Mimarisi

MultiResUNet mimarisi, ResNet ve U-Net mimarilerinin özelleştirilmiş halidir. Bu mimari gücünü MultiRes bloğu (MultiRes block) ve Res yollarının (Res path) U-Net mimarisi formunda tasarlanmasına borçludur.

Şekil 2.29'da 3x3, 5x5, 7x7 evrişim filtreleri peş peşe gelecek şekilde birleştirilerek bir Inception modeli kullanılır. Şekil 2.30'da 3x3, 5x5, 7x7 evrişim filtreleri peş peşe kullanmak yerine 3x3 filtrelerle daha küçük ve işlem yükü az olanlar kullanılmış olur.

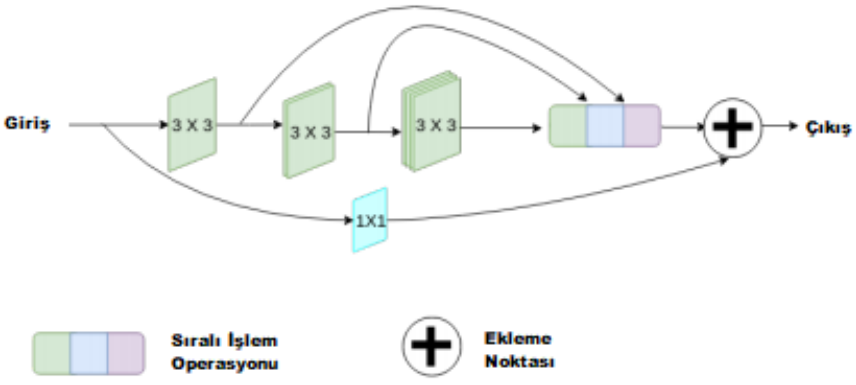


Şekil 2.29 Basit bir Inception modeli [44].



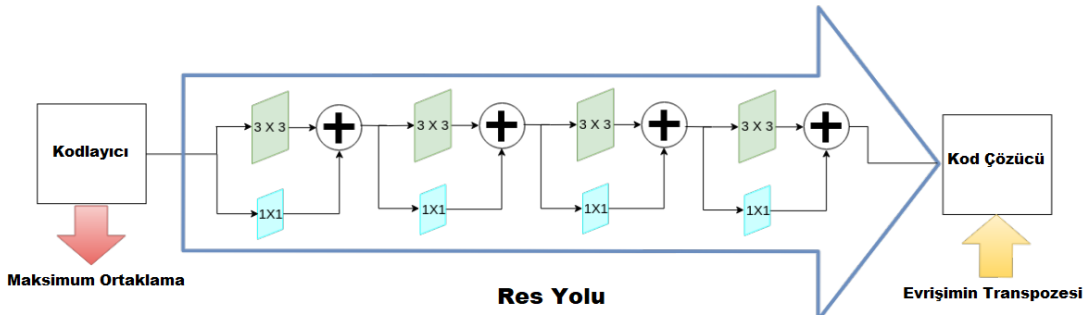
Şekil 2.30 MultiRes Bloğu oluşturmak için hazırlanan 3x3'lük evrişim işlemleri [44].

Şekil 2.31 ardışık üç katmandaki filtre sayısının kademeli olarak arttığı ve artık bir bağlantıya sahip MultiRes bloğunu gösterir. Turkuaz renkte gösterilen 1x1 filtre, artık bağlantının boyutlarını korumak için kullanılır.



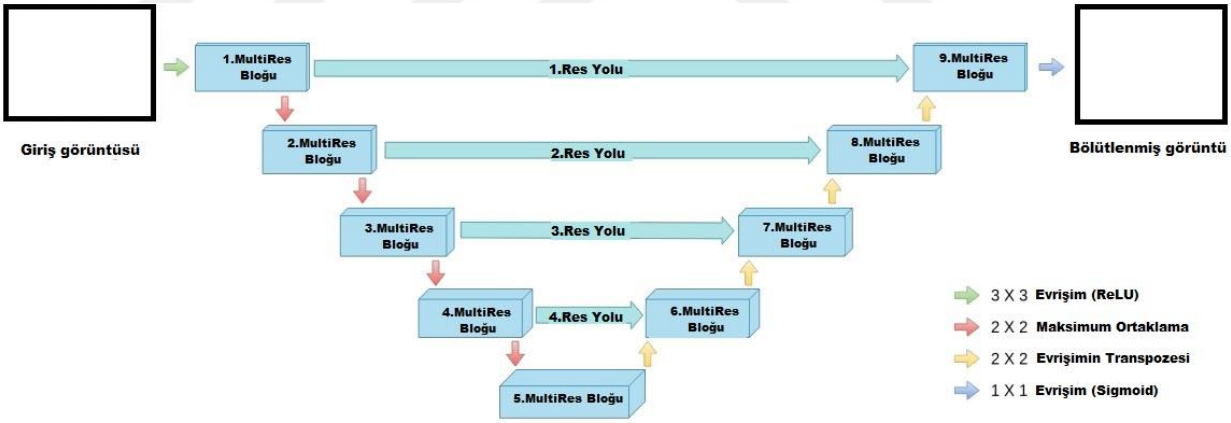
Şekil 2.31 MultiRes Bloğu [44].

Şekil 2.32'de MultiResUNet'in Res yolu (Res path) gösterilmektedir. MultiRes blokları arasındaki bu işlem 3x3 evrişim filtrelerini ve artık öğrenme metodunu kullanır. Artık öğrenme neticesinde boyutların değişmemesi adına 1x1 evrişim filtreleri burada da kullanılır.



Şekil 2.32 MultiResUNet'te yer alan Res yolu [44].

Şekil 2.33'te MultiRes blokları ve Res yollarıyla birlikte U-Net formuna sahip MultiResUNet mimarisi yer almaktadır. Evrişim, maksimum ortaklama gibi katmanlar kullanılarak giriş görüntüsünden bölütlenmiş görüntü elde edilir.



Şekil 2.33 MultiResUNet mimarisi [44].

2.13 Doğruluk Ölçütleri

Doğruluk ölçütü, değerlendirme kıstasına göre şekillenir. Bölütlenmiş görüntüleri kıyaslayarak her bir görüntü için kullanılan yöntemin başarısı ölçülülebilir. Referans görüntü ve hipotezi sınayacak olan görüntü, iki ayrı veri seti olarak düşünülecek olursa; yapılacak değerlendirme, iki küme arasındaki benzerliğin

hesaplanması görevine dönüşür. Dice Benzerlik Katsayısı kullanılarak yöntem başarısı ölçülebilir.

2.13.1 Dice Benzerlik Katsayısı

A ve B iki farklı veri setini ifade eden kümeler olmak üzere,

$$dice\ (A,B) = \ 2\ \frac{A \cap B}{A+B} \tag{4.1}$$

A ve B arasındaki benzerlik katsayısı, iki küme kesişiminin birleşimine oranının iki katı alınarak bulunmaktadır [45].

Görüntüler tek boyutlu diziye dönüştürülüp iki görüntü arasındaki benzerlik katsayısı hesaplanarak, her bir görüntü için DBK (Dice Benzerlik Katsayısı) elde edilmiştir. DBK ortalamaları ile uygulamada kullanılan tüm mimarilerin performansı incelenmiştir.

2.13.2 Karşılaştırma Matrisi

Tablo 2.1'de TP, olumlu ve doğru tahmin edilen; TN, olumsuz ve doğru tahmin edilen; FP, olumlu ve yanlış tahmin edilen; FN, olumsuz ve yanlış tahmin edileni ifade etmektedir [46].

Tablo 2.1 Karşılaştırma Matrisi

Karşılaştırma	Pozitif	Negatif
Pozitif	TP	FP
Negatif	FN	TN

Karşılaştırma matrisi, Kalite, Doğruluk, Bütünlük ve F-Skoru gibi doğruluk değerlerinin hesaplanmasını sağlar. Kalite, negatif tahminin referans doğruluğunu; Doğruluk, pozitif tahmin doğruluğunu; Bütünlük, pozitif tahminin

referans doğruluğunu; F-Skoru ise Kalite ve Bütünlük arasındaki ilişkiyi tanımlar [46, 47, 48, 49].

$$Kalite = \frac{TN}{TN+FP} \qquad Doğruluk = \frac{TP}{FP+TP}$$

$$Bütünlük = \frac{TP}{FN+TP} \qquad F - Skoru = \frac{2\times TP}{(2\times TP+FP+FN)} \tag{4.2}$$

2.13.3 K-Katlı Çapraz Doğrulama

Makine öğrenmesinde kullanılan algoritmaların değerlendirilebilmesi için birden çok test kümesine ihtiyaç vardır. Tek bir test kümesinin varlığı verinin doğru şekilde değerlendirilmesini engelleyebilir. Bu durumun önüne geçmek adına veri seti, K adet eşit parçaya bölünür. Her bir parça hem test hem de eğitim aşamasında kullanılır. KKÇD (K-Katlı Çapraz Doğrulama) metodu ile kullanılan mimarilerin doğruluğu tespit edilmektedir [50]. Bu tez çalışması kapsamında 5-Katlı Çapraz Doğrulama kullanılmıştır.

3
UYGULAMA

3.1 Çalışma Alanı ve Kullanılan Veriler

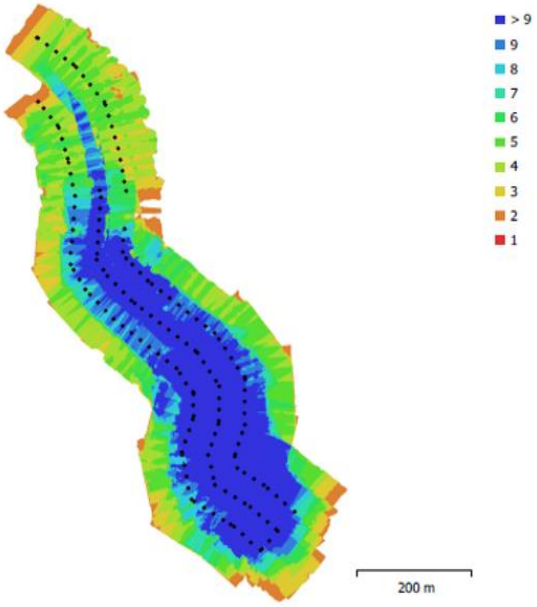
Bu çalışma Şekil 3.1'de gösterilen İstanbul, Esenler, Yıldız Teknik Üniversitesi, Davutpaşa Kampüsüne ait olup, 2019 yılının ikinci yarısında İHA ile elde edilen veriler kullanılarak gerçekleştirilmiştir. Çalışma alanı 25 hektarlık bir alandır. İHA alımı ile elde edilen 159 adet hava fotoğrafından üretilen ortofoto kullanılmıştır. Bu ortofoto'da yer alan yol orta kaldırımları ve çevresindeki yeşil alanlar bu çalışmanın konusu kapsamındadır.



Şekil 3.1 Davutpaşa Kampüsü, İstanbul (Google Earth).

3.1.1 İHA Özellikleri

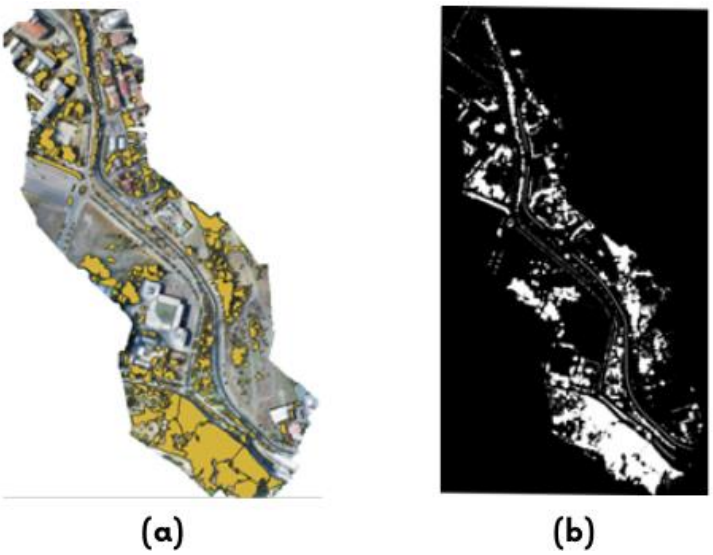
DJI Phantom marka kamera ve FC6310R (f=9mm) lens kullanılmıştır. Fotoğraf çözünürlüğü 5472x3648 piksel ve diyafram açıklığı f/4.5 olup 1/120 sn. poz süresiyle, 103 metre uçuş yüksekliğiyle ve 2.53 cm/piksel yer çözünürlüğü ile çekilmiştir. Veri seti bu özelliklere sahip 159 adet hava fotoğrafından meydana gelmektedir. Şekil 3.2'de İHA ölçümü ile elde edilen verilerin raporunda, alım esnasındaki hava fotoğrafının göstermektedir. 1'den 9'a kadar numaralandırılan ve bu numaralara göre yapılan renklendirme, çalışma alanındaki hava fotoğraflarının yoğunluğunu göstermektedir. Peş peşe gelen hava fotoğrafları arasındaki toplam nokta sayısı 182914 ve piksel hatası 0.568 px'dir.



Şekil 3.2 İHA ile alım yapılan hava fotoğraflarının yoğunluk raporu.

3.1.2 Veri Setinin Hazırlanması

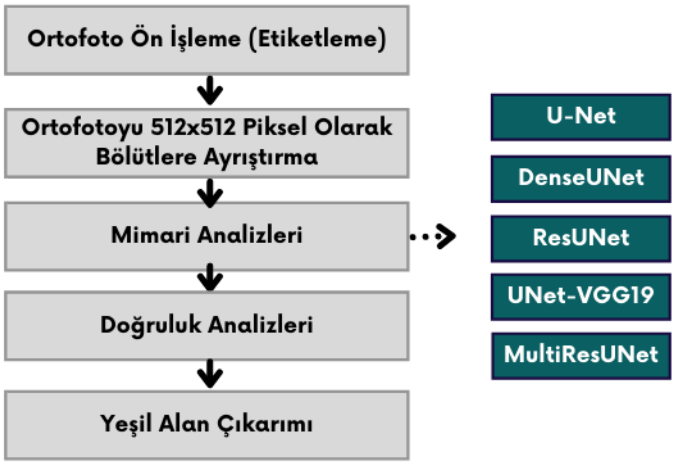
159 adet hava fotoğrafından elde edilen ortofoto, QGIS uygulaması kullanılarak etiketlenmiştir (Şekil 3.3, (a)). Etiketleme işlemi yeşil alanların vektör veri olarak hazırlanmasını ifade etmektedir. Ardından ArcMap uygulaması kullanılarak vektör veriden siyah-beyaz (binary) görüntü verisi elde edilmiştir (Şekil 3.3, (b)). Burada beyaz renkte görünen alanlar, operatör tarafından etiketlenen yeşil alanları ifade etmektedir. Bu görüntüler bölütlenerek verilerin daha hızlı işlenmesi amaçlanmıştır. Veri setinde yer alan görüntülerin 1000 tanesi eğitim, 217 tanesi de test aşamasında kullanılmıştır. Tez çalışması kapsamında "Yeşil Alan" ve "Yeşil Alan Değil" sınıfı üzerinden yapılan işlemler sonucunda ortofotodaki yeşil alanların otomatik çıkarımı amaçlanmıştır.



Şekil 3.3 (a) Etiketlenmiş veri ve (b) yeşil alanı temsil eden siyah-beyaz görüntü.

3.2 Eğitim Verilerinin Hazırlanması

Davutpaşa kampüsüne ait ortofoto, derin öğrenme algoritmalarına uygun olacak şekilde bölütlenmiştir. Yeşil alanların çıkarımı için 23.452x41.718 boyutundaki siyah-beyaz ortofoto görüntüsünden 512x512 boyutunda 61,09163 adet alt görüntü parçası elde edilmiştir. Bu bölütlenmiş görüntüler U-Net, UNet-VGG19, ResUNet, DenseUNet ve MultiResUNet mimarilerinde ayrı ayrı eğitilmiştir. Görüntüleri sisteme yüklerken ve alt görüntülere ayrıştırırken OpenCV (Open Source Computer Vision Library) kütüphanesi kullanılmıştır. Kesilen görüntüler NumPy dizisine dönüştürülürken hem OpenCV hem de NumPy (Numerical Python) kütüphanesi kullanılmıştır. Akış şeması Şekil 3.4'te verilmiştir.



Şekil 3.4 Akış şeması.

3.3 Kullanılan Donanım ve Yazılımlar

Yeşil alanların tespitinde,

- NVIDIA GeForce RTX 2080 Super, 8 GB RAM Ekran Kartı Sürücüsü
- Intel® Core™ i9-10900K CPU @3.70 GHz İşlemci
- 64 GM RAM, 64 bit İşletim Sistemi

Konfigürasyonundan faydalanılmıştır.

Derin öğrenme algoritmaları kullanılırken Python - 3.7.0 programlama dili, Tensorflow - 1.15.0, Keras - 2.3.1, OpenCV - 3.4.2 ve NumPy - 1.19.2 kütüphaneleri kullanılmıştır.

3.4 Ağ Parametreleri

Yeşil alanlara ait görüntülerin U-Net, UNet-VGG19, ResUNet, DenseUNet ve MultiResUNet mimarileri ile eğitiminde kullanılan parametreler Tablo 3.1'de ve bu mimarilere ait parametre sayıları Tablo 3.2'de gösterilmiştir.

Tablo 3.1 Ağ Eğitimindeki Parametreler

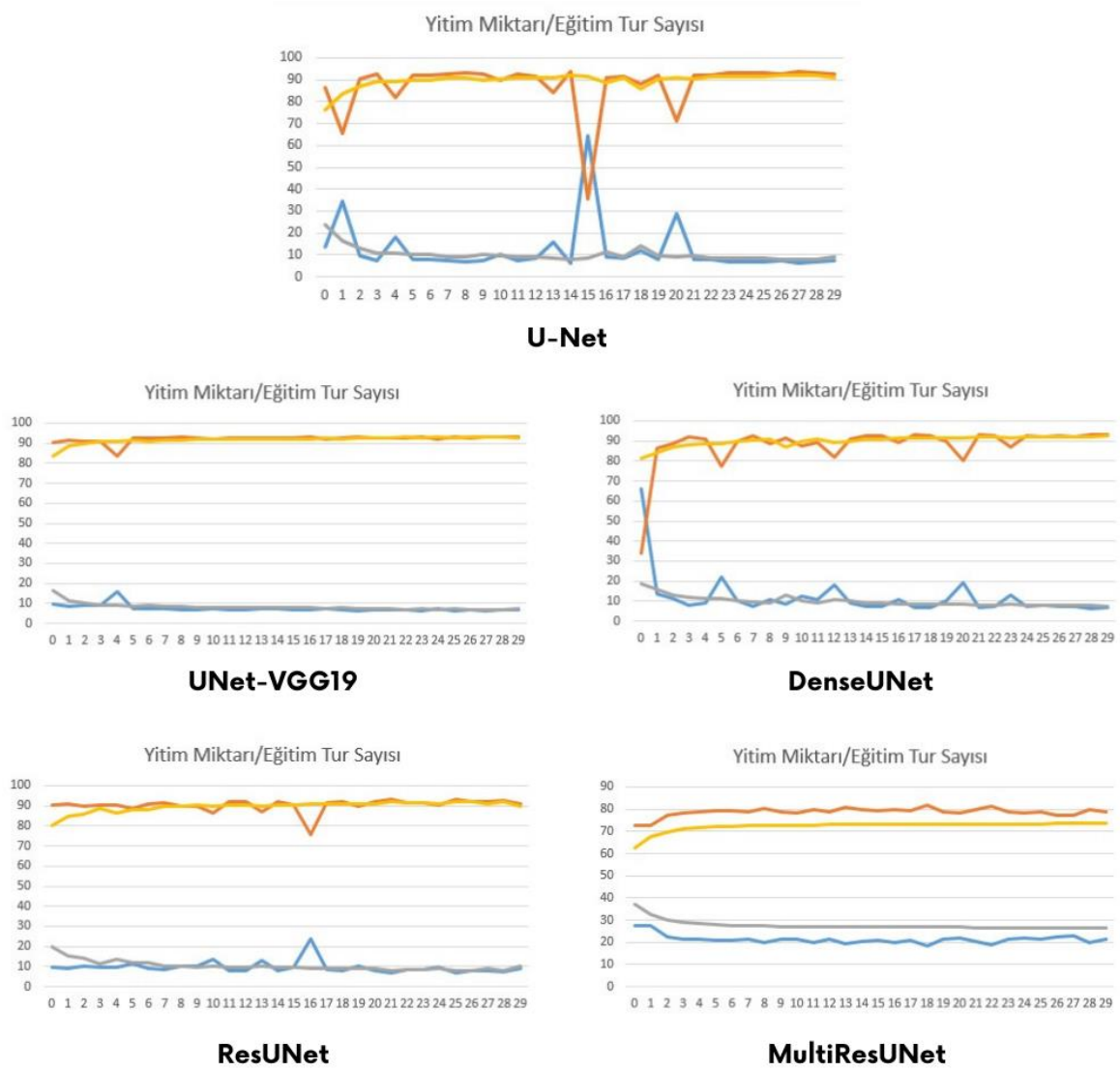
Parametreler	**Değerler**
Eğitim Tur Sayısı	30
Optimizasyon Algoritması	Adam
Öğrenme Oranı	0.001
Küme Boyutu	4
Görüntü Boyutu	512x512

Tablo 3.2 Mimarilere Ait Parametre Sayıları

Mimariler	Parametre Sayısı
U-Net	34.632.770,00
UNet-VGG19	24.339.281,00
ResUNet	24.459.299,00
DenseUNet	12.145.122,00
MultiResUNet	7.262.804,00

3.5 Yeşil Alanların Bölütleme Doğruluğu

Beş adet farklı mimari için yeşil alanların bölütlemesine dair KKÇD eğitim ve test doğruluğu, Şekil 3.5'te eğitim tur sayısı boyunca meydana gelen yitim miktarı gösterilmektedir. Şekil' de sarı eğitim doğruluğunu, turuncu test doğruluğunu ifade eder. Gri ile gösterilen yitim miktarını, mavi ile gösterilen ise doğrulama kaybını ifade eder.



Şekil 3.5 Beş mimari için eğitim ve test doğrulukları.

Beş adet farklı mimari için eğitim oranları kıyaslandığında en kötü performansı sergileyen mimarinin %72 eğitim doğruluk oranıyla MultiResUNet olduğu görülmektedir. U-Net mimarisi, MultiResUNet mimarisine kıyasla daha iyi bir doğruluk oranı sunmasına karşın test doğruluk oranının istikrarlı olmadığı görünmektedir. Buna karşın U-Net mimarisi genellikle %90'ın üstünde bir eğitim doğruluğu sergilemektedir. DenseUNet mimarisi, U-Net'e göre daha istikrarlı sayılabilecek bir performans göstermektedir. Bunun yanında, bu mimaride eğitim tur sayısı arttıkça %90'ın üstünde bir başarıya ulaşmaktadır. ResUNet mimarisi, diğer analiz edilen mimarilerden daha iyi bir performans sergilemiştir. Fakat eğitim başarısı %90'a yakın değerlerde kaldığı görülmektedir. En iyi performansı

gösteren ve en istikrarlı olan mimari ise %95 gibi bir eğitim doğruluk oranına sahip UNet-VGG19 mimarisidir.

Tablo 3.3 Karşılaştırma Matrisinden Elde Edilen Sonuçlar

Mimariler	Doğruluk Değerleri					Karşılaştırma Matrisi Değerleri			
	DBK	Bütünlük	Kalite	Doğruluk	F-Skoru	TP	TN	FP	FN
U-Net	0.94	0.74	0.42	0.90	0.81	0.66	0.05	0.07	0.23
MultiResUNet	0.82	0.97	0.01	0.72	0.82	0.70	0.00	0.28	0.02
ResUNet	0.93	0.80	0.17	0.84	0.82	0.68	0.03	0.13	0.17
DenseUNet	0.93	0.75	0.33	0.89	0.82	0.66	0.04	0.08	0.22
UNet-VGG19	0.93	0.72	0.56	0.92	0.81	0.64	0.07	0.05	0.25

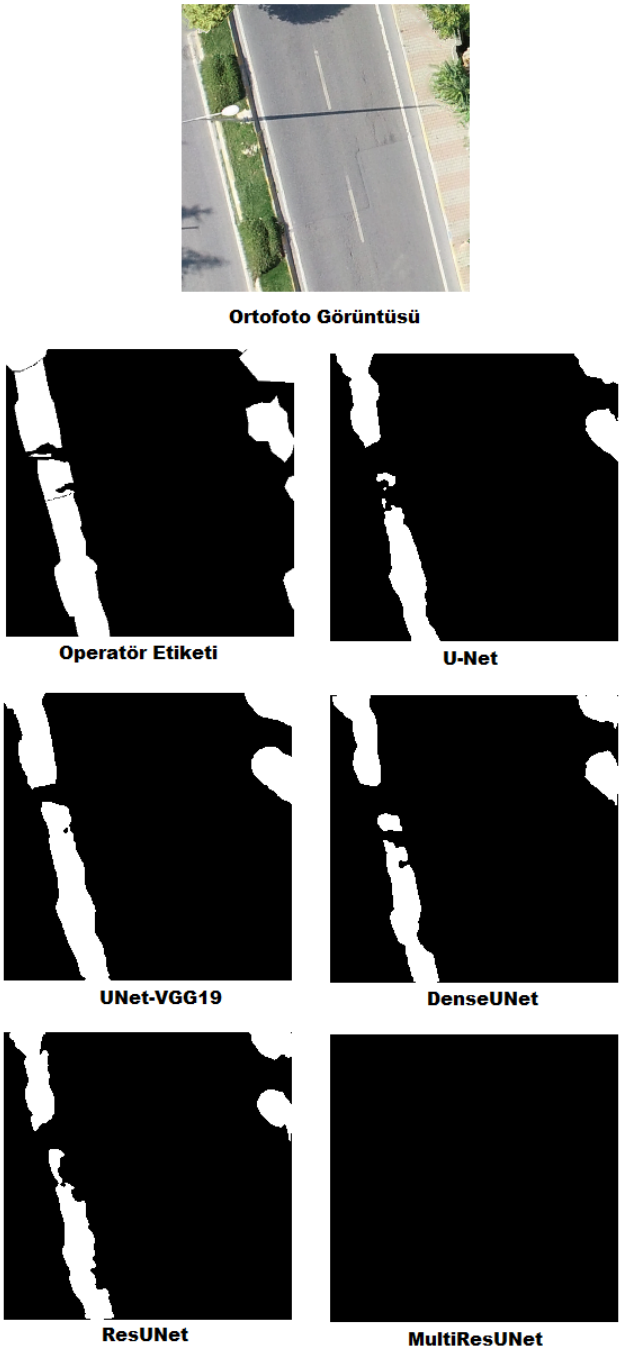
Bu çalışmada kullanılan mimariler U-Net, UNet-VGG19, ResUNet, DenseUNet ve MultiResUNet olmak üzere, karşılaştırma matrisinden elde edilen değerler Tablo 3.3'te doğruluk değerleri yer almaktadır. Karşılaştırmalar, bölütlenmiş görüntülerin referans etiketi baz alınarak yapılmıştır.

Tablo 3.3'te beş mimari için de F-Skoru benzerlik göstermektedir. En yüksek DBK'ya %94 oranı ile U-Net mimarisinde ulaşılmış olup ResUNet, DenseUNet ve UNet-VGG19 mimarileri de U-Net'e yakın sonuçlar alınmıştır. MultiResUNet mimarisi %97 oranında Bütünlük değerine ulaşmıştır. Bütünlük konusunda bu mimarinin diğer mimarilerden pozitif ayrıştığı görülmüştür. UNet-VGG19 mimarisinin Doğruluk değeri ise %92 olarak elde edilmiştir.

Tablo 3.4 Yanlış veya Doğru Tespit Edilen Yeşil Alan Çıkarım Çıktıları

Mimariler	Karşılaştırma Matrisi Değerleri			
	TP	TN	FP	FN
U-Net	37151704.89	13114844.20	2785963.73	3832737.49
MultiResUNet	39720835.90	1218609.17	170089.95	15775714.14
ResUNet	38416115.11	9464472.76	1474810.74	7529850.54
DenseUNet	37702128.23	12437716.72	2212169.43	4533236.20
UNet-VGG19	36385837	13978089.03	3668686.49	2852637.04

Tablo 3.4'te beş mimari için yeşil alanların çıktıları gösterilmektedir. Her bir test görüntüsü için elde edilen karşılaştırma matrisi elemanları U-Net, UNet-VGG19, ResUNet, DenseUNet ve MultiResUNet mimarileri için tabloda verildiği gibi bulunmuştur. TP yeşil alan olup yeşil alan olarak tahmin edilen alanı, FP yeşil alan olmadığı halde yeşil alan olarak tahmin edilen alanı, FN gerçekte yeşil alan olup tahminde arka plan olarak tahmin edilen alanı, TN referansta ve tahminde arka plan olan alanı ifade etmektedir. Yeşil alan olup yeşil alan olarak tahmin edilen alanı en iyi tespit eden mimari MultiResUNet, arka plan olup arka planı en iyi tespit eden mimari ise UNet-VGG19 olmuştur. Yeşil alan olan alanları, arka plan olarak tahmin eden MultiResUNet mimarisi, en fazla yanlış tahmini yapan mimari olmuştur.



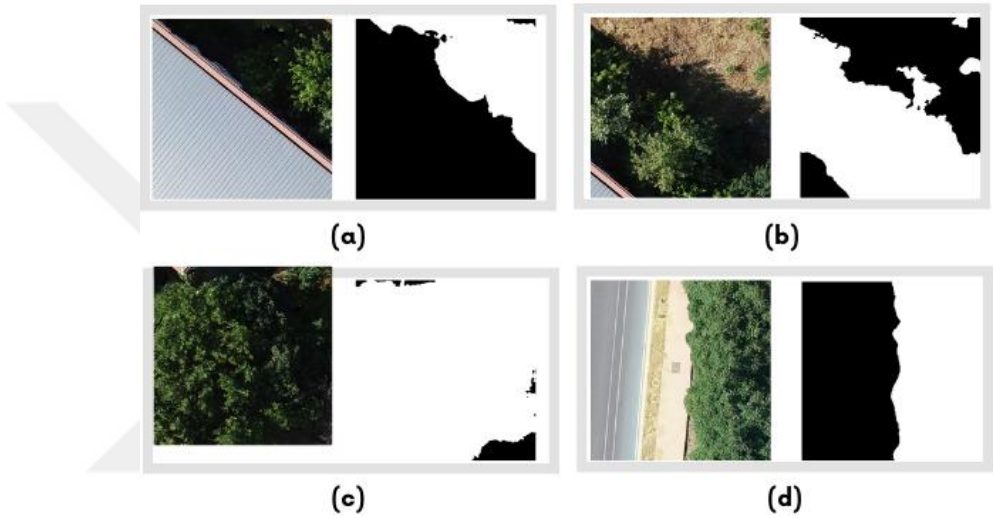
Şekil 3.6 Bölütlenmiş bir görüntüye ait ortofoto görüntüsü, operatör etiketi ve mimarilerin yeşil alan tahminlerini gösteren çıktı görüntüler.

Şekil 3.6'da yer alan Ortofoto görüntüsü, Davutpaşa kampüsünün orijinal ortofotodan bölütlenmiş olan görüntüyü ifade eder. Operatör etiketi, ortofoto üzerinden CBS (Coğrafi Bilgi Sistemi) programı kullanılarak etiketlenen verinin siyah-beyaz görüntüye dönüştürülmüş halini ifade eder. Şekil 3.5'te yer alan beş mimari, operatör etiketine uygun şekilde eğitilen algoritmalarla geliştirilen sistemlerin kendi tahminlerini göstermektedir. Mimariler, bu tahminleri Şekil

3.4'te yer alan doğruluk oranlarına benzer şekilde gerçekleştirmiştir. Söz konusu bölütlenmiş görüntü için MultiResUNet mimarisinin tamamen başarısız olduğu açıkça görülmektedir. Diğer mimariler de bu alanı tespit edebildiği halde en başarılı mimari UNet-VGG19'dur.

3.5.1 Yeşil Alanlar için Özel Durumlar

Bu bölümde bölütlenmiş görüntülerde yer alan yeşil alanların, en yüksek başarıyı gösteren UNet-VGG19 mimarisi tarafından tespit edilirken ortaya çıkan özel sonuçlar analiz edilmiştir.



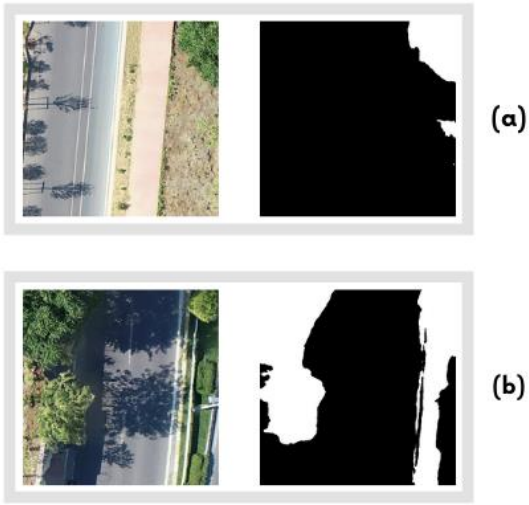
Şekil 3.7 (a), (b), (c), (d) ile gösterilen alanlar UNet-VGG19 mimarisinin sık alanlardaki yeşil alan tespitini göstermektedir.

Şekil 3.7'de yer alan dört görüntü UNet-VGG19 mimarisinin, sık alanda yeşil alanı tespit etme konusundaki başarısını göstermektedir. (a) ve (b) görüntüsünde yer alan gölgede kalan yeşil alanlar tespit edilmiştir. (c) görüntüsünde sadece yeşil örtüden oluşan alanda tespitler oldukça başarılı görünmektedir. (d) görüntüsünde de yol kenarında yer alan ağaçlar dallarının ve yapraklarının detayı nezdinde tespit edilebilmiştir.



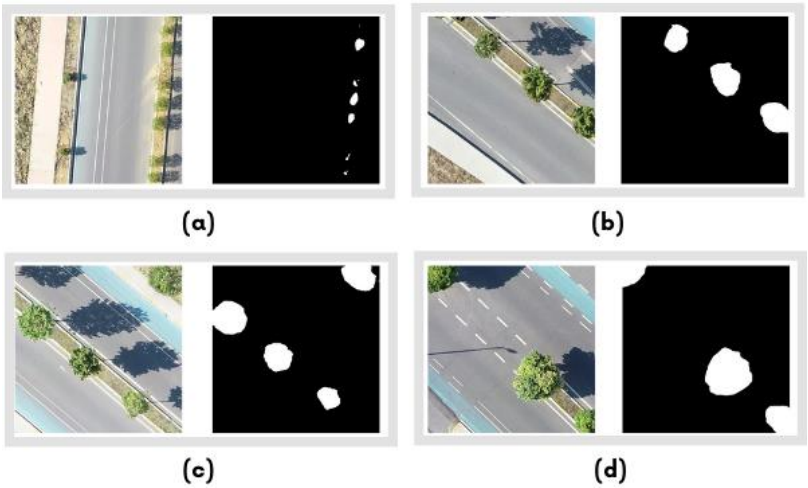
Şekil 3.8 Gösterilen alan UNet-VGG19 mimarisinin gölgedeki yeşil alan tespitini göstermektedir.

Şekil 3.8'de yer alan görüntü UNet-VGG19 mimarisinin, gölgedeki yeşil örtüyü tespit ettiğini fakat çatıya düşen ağaç yaprakları gölgesini tespit etmediğini göstermektedir. Bu durum, mimarinin yeşil alan tespitinde amacına uygun bir performans gösterdiğini doğrulamaktadır.



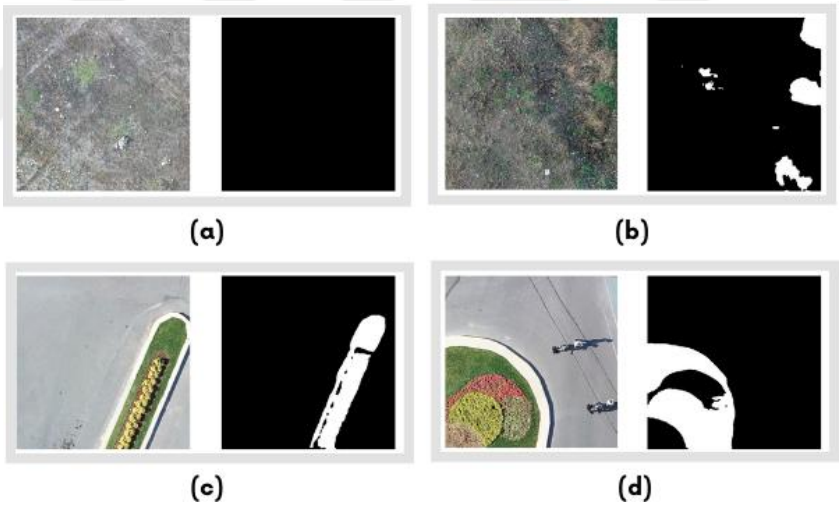
Şekil 3.9 (a) ve (b) ile gösterilen alan UNet-VGG19 mimarisinin ağaç gölgeleri ile yeşil alanları karıştırmadığını göstermektedir.

Şekil 3.9'da yer alan (a) ve (b) görüntüsünde UNet-VGG19 mimarisinin, ağaç yapraklarını ve yol orta kaldırımındaki yeşil alanları tespit ettiğini, yola düşen ağaç gölgelerini tespit etmediğini göstermektedir. Söz konusu gölgeler, amaca uygun olmadığı için mimarinin başarılı olduğunu göstermektedir.



Şekil 3.10 (a), (b), (c), (d) ile gösterilen alanlar UNet-VGG19 mimarisinin seyrek ağaçları tespitini göstermektedir.

Şekil 3.10'da yer alan (a), (b), (c) ve (d) görüntüsünde UNet-VGG19 mimarisinin, seyrek ağaçları tespit ettiğini, yola düşen ağaç gölgelerini tespit etmediğini göstermektedir. Mimari bu örneklerde de amacına uygun bir başarı göstermiştir.



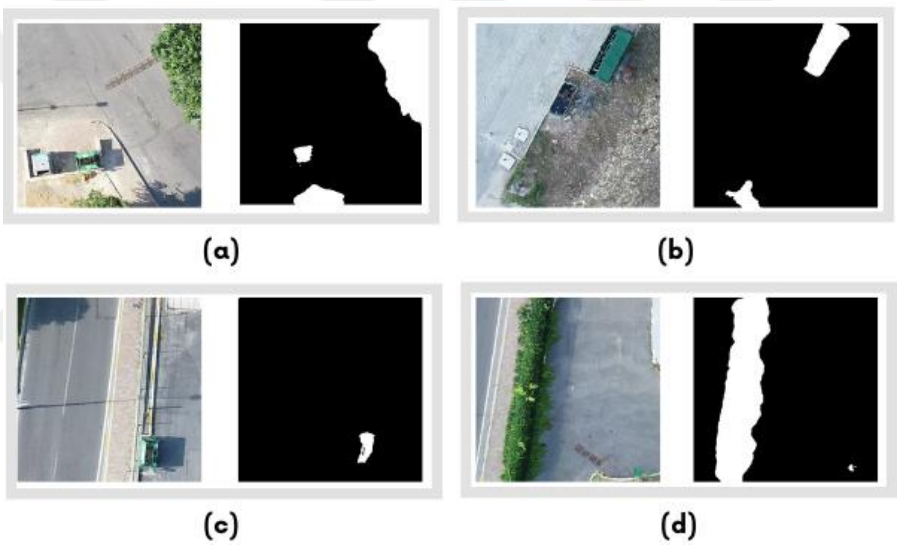
Şekil 3.11 (a), (b), (c), (d) ile gösterilen alanlar UNet-VGG19 mimarisinin zeminde yer alan yeşil alanların tespitini göstermektedir.

Şekil 3.11'de yer alan (a) görüntüsünde, zeminde yer alan açık tondaki çimleri yeşil alan olarak tespit etmediği, (b) görüntüsünde ise zeminde yer alan yeşil alanları tespit ettiği görülmektedir. (c) ve (d) görüntüsünde refüjde zeminde bulunan yeşil ve yeşil tonlarındaki örtüyü tespit edebildiği görülmüştür.



Şekil 3.12 Gösterilen alan UNet-VGG19 mimarisinin çatıda yer alan malzemenin üstüne düşen gölgeyi yeşil alan olarak tespit ettiğini göstermektedir.

Şekil 3.12'de çatıda yer alan, yeşil örtü tonuna yakın olan malzemeye düşen gölgeyi mimari yeşil alan olarak tespit etmiştir. Fakat bu tespit doğru değildir.



Şekil 3.13 (a), (b), (c), (d) ile gösterilen alanlar UNet-VGG19 mimarisinin yeşil alan olmadığı halde yeşil alan olarak tespit ettiğini nesneleri göstermektedir.

Şekil 3.13'te (a) ve (c) görüntüsünde çöp kutusunun yeşil detaylarını, (b) görüntüsünde trafonun yeşil siperliğini, (d) görüntüsünde ise sağ alt köşede yer alan nesnenin turkuaz rengi itibariyle yeşil alan olarak tespit edilmiştir.

U-Net, UNet-VGG19, ResUNet, DenseUNet ve MultiResUNet mimarileri arasında en yüksek DBK oranını %94 ile U-Net elde etmiştir. UNet-VGG19, ResUNet ve DenseUNet mimarileri ise %93 ile U-Net mimarisine yakın bir performans sergilemiştir. Bütünlük analizi olarak değerlendirildiğinde MultiResUNet mimarisi %97 oranına ulaşmıştır. Kalite analizinde en yüksek başarıyı %56 oranıyla UNet-

VGG19 mimarisi göstermiştir. Doğruluk analizi açısından karşılaştırıldığında ise yine UNet-VGG19 mimarisi öne çıkmaktadır. %92 doğruluk oranı elde eden bu mimariyi %90 ile U-Net, %89 ile DenseUNet mimarisi izlemektedir. F-Skoru tüm mimariler için neredeyse aynıdır. Derin öğrenme uygulamalarının çıkarım başarısını kıyaslamanın diğer bir önemli kriteri ise eğitim doğruluklarıdır. Eğitim doğruluğu açısından bakıldığında %95 ile en yüksek performans UNet-VGG19'a aittir. U-Net, ResUNet ve DenseUNet mimarileri ise %90 eğitim doğruluk oranına ulaşabilmiştir. Eğitim doğruluğu açısından sınıfta kalan mimari MultiResUNet olmuştur.

Tablo 3.5 Yeşil Alan Sınıfının Mimari ve Sınıf Analizlerine Göre Çıkarım Değerleri

Mimari / Sınıf Analizleri	Sık Alan	Seyrek Alan	Gölge Alan	Zemin	Çatı	Refüj	Bina	Yol	Kaldırım
U-Net	4	4	3	1	5	4	5	5	5
ResUNet	4	3	2	2	2	2	3	3	3
DenseUNet	5	5	3	2	4	3	4	4	4
UNet-VGG19	5	5	5	5	4	5	5	5	5
MultiResUNet	3	2	2	0	0	0	0	0	2
Tabloda yer alan yeşil alan çıkarım değerleri: 0- başarısız, 1- çok kötü, 2- kötü, 3- orta, 4- iyi, 5- çok iyi.									

Eğitim doğruluğu, DBK, Bütünlük, Kalite, Doğruluk ve F-Skoru doğruluk ölçütleri bir bütün olarak değerlendirildiğinde Tablo 3.5'teki mimarilerin ilgili alanda yeşil alan çıkarım değerleri ortaya çıkmaktadır. Bu değerler 5 üzerinden verilmiş olup doğruluk ölçütleri ve bölütlenmiş sonuç görüntülerin kıyaslanmasına göre hazırlanmıştır. Burada yer alan bu değerler 0- başarısız, 1- çok kötü, 2- kötü, 3- orta, 4- iyi, 5- çok iyi gibi çıkarım değerlerini ifade etmektedir. Tablo 3.5'e göre en başarılı mimarinin UNet-VGG19, en başarısız olan mimarinin ise MultiResUNet olduğu görülmüştür. En başarılı mimari olan UNet-VGG19 mimarisinin çıkarımını yanlış yaptığı alanlar mevcuttur. Çıkarımı yanlış yapılan bu alanların toplamı ise 10 adetten az olduğu tespit edildiğinde ise bu çalışma için bu mimarinin başarısı kabul edilebilir düzeyde olduğu görülmüştür.

4
SONUÇ VE ÖNERİLER

Derin öğrenme ile yeşil alanların çıkarımının araştırıldığı bu çalışmada İstanbul, Esenler'de yer alan YTÜ Davutpaşa kampüsünün 159 adet İHA görüntüsünden ortofoto elde edilmiştir. Ortofoto üzerinden, QGIS uygulaması kullanılarak yol orta kaldırımda ve kampüste yer alan yeşil alanlar vektör veri olarak etiketlenmiştir. Etiketlenen bu veri, ArcGIS uygulamasında siyah-beyaz diğer bir deyişle ikili (binary) görüntüye dönüştürülmüştür. Bu görüntülerin algoritmalar tarafından daha hızlı işlenebilmesi için siyah-beyaz görüntü, 512x512 olarak bölütlenmiştir. Bölütlenmiş görüntüler, NumPy dizisine dönüştürülerek mimari eğitimlerine başlanmıştır. Bu çalışmada U-Net, UNet-VGG19, ResUNet, DenseUNet ve MultiResUNet mimarileri kullanılmıştır. Bu aşamadan sonra mimarilerin karşılaştırması ve değerlendirilmesi için doğruluk analizleri yapılmıştır.

1217 adet 512x512 piksellik görüntünün 1000 adedi eğitim verisi, 217 adedi ise test verisi olarak kullanılmıştır. Etiketleme süreci 15 saat, eğitim süreci ise bahsi geçen donanım konfigürasyonuyla her bir mimari için yaklaşık 30 dakika olmak üzere beş adet mimari için toplamda 2.5 saat sürmüştür. Bu tez çalışmasında kullanılan ekran kartı ile zamandan tasarruf sağlamıştır. Daha derin ağ mimarilerinin kullanılması söz konusu olduğunda daha performanslı ekran kartı grafik sürücülerinin kullanılabileceği görülmüştür.

Çalışma kapsamında kullanılan derin öğrenme algoritmalarından olan U-Net mimarisi ve varyasyonlarının genel olarak yeşil alanları tespit etmek konusunda başarılı olduğu tespit edilmiştir. En düşük başarıya sahip mimari olan MultiResUNet'in %70'in üstünde doğruluk oranı olduğu, en yüksek başarıya sahip mimarinin ise %95'e yakın bir doğruluk oranıyla UNet-VGG19 mimarisi olduğu görülmüştür.

En yüksek başarıyı yakalayan UNet-VGG19 mimarisinin gölgede kalan yeşil alanların yanında hem sık hem de seyrek yeşil alanları başarıyla tespit ettiği

görülmüştür. Bununla birlikte ağaç gölgelerini yeşil alan olarak tespit etmeyerek amacına uygun performans gösterdiği tespit edilmiştir. Ayrıca, bu mimari zemindeki yeşil örtünün de doğru bir şekilde tespit edilmesinde başarılı sonuçlar vermiştir.

UNet-VGG19 ve U-Net mimarisi kullanılırken hatalı/yanlış tespit edilen detaylar olduğu görülmüştür. Örneğin; turkuaz renkte olan çatı gölgesi, çöp kutusunun yeşil detayları ve kaldırım kenarında bulunan turkuaz renkte bir nesne yeşil alan olarak tespit edilmiştir. Bu durumun nedeni ortofoto'da yeşil alanı gösteren bazı bölgelerin bulanık olmasıdır. Bulanık bölgelere düşen alanlar, etiketleme dışı bırakılamadığı için yeşil alanı temsil eden spektral yansıma değerlerinin diğer objelerden ayırt edilemediği görülmüştür. Fakat doğru tespit edilemeyen nesnelerin ve detayların toplamının 10 adetten az olduğu görülmüştür. Bu bakımdan UNet-VGG19 mimarisinin doğruluğu %95'lere yaklaşmaktadır.

Yeşil alanların derin öğrenme ile tespitinde daha doğru sonuçlar alınması için daha yüksek çözünürlüklü görüntülerle ve görüntü sayısı olarak daha çok görüntüyle çalışılması gerektiği düşünülmektedir. Eğitim veri sayısının ve/veya mimari derinliğini arttırmanın yanlış tespit edilen görüntü sayısını azaltabileceği önerilmektedir.

Teknolojik ilerleme ve gelişmelerle daha performanslı donanımlarla birlikte yeni derin öğrenme mimarileri ile veri setlerinin daha doğru eğitilip, güncel mesleki çözüm arayışlarına hizmet edeceği düşünülmektedir. Örneğin kentsel dönüşüm ve değişim analizleri, akıllı şehirlerin planlanması, çevre ve iklim değişikliklerinin tespiti gibi önemli konularda mesleki disiplin ve derin öğrenme mimarileriyle karma bir yaklaşım geliştirilebilir. Böyle bir yaklaşımın hızlı, doğru ve amaca uygun hizmet verebileceği öngörülmektedir.

KAYNAKÇA

[1] Internet: Suman, https://ai.stackexchange.com/questions/15859/is-machine-learning-required-for-deep-learning, 24.01.2021.

[2] K. Yeang, DesignWith Nature, M. C. Graw Hill Company, London, 1992.

[3] M. Li, J. Im, and C. Beier, "Machine learning approaches for forest classification and change analysis using multi-temporal Landsat TM images over Huntington Wildlife Forest", GIScience & Remote Sensing, Aug 2013.

[4] T. Kavzoğlu, İ. Çölkesen, E. K. Şahin, "Obje Tabanlı Yaklaşımda Makine Öğrenme Algoritmalarının Sınıflandırma Performansının Analizi", TUFUAB VIII. Teknik Sempozyumu, May. 2015.

[5] A. Şeker, B. Diri, H. H. Balık, "Derin Öğrenme Yöntemleri ve Uygulamaları Hakkında Bir İnceleme", Gazi Mühendislik Bilimleri Dergisi, Mart 2017.

[6] M. Wessel, M. Brandmeier and D. Tiede, "Evaluation of Different Machine Learning Algorithms for Scalable Classifications of Tree Types and Tree Species Based on Sentinel-2 Data", Remote Sensing, Sep. 2018.

[7] F. Doğan, İ. Türkoğlu, "Derin Öğrenme Algoritmalarının Yaprak Sınıflandırma Başarımlarının Karşılaştırılması", Sakarya University Journal of Computer and Informations Sciences, Apr. 2018.

[8] M. A. Kızrak, B. Bolat, "Derin Öğrenme ile Kalabalık Analizi Üzerine Detaylı Bir Araştırma", Bilişim Teknolojileri Dergisi, Tem. 2018.

[9] O. Yıldız, "Melanoma detection from dermoscopy images with deep learning methods: A comprehensive study", Gazi Üniversitesi Mühendislik ve Mimarlık Fakültesi Dergisi, Mart 2019.

[10] B. G. Weinstein, S. Marconi, S. Bohlman, A. Zare, and E. White, "Individual Tree-Crown Detection in RGB Imagery Using Semi-Supervised Deep Learning Neural Networks", Remote Sensing, June 2019.

[11] A. Wang, Y. Wang, and Y. Chen, "Hyperspectral image classification based on convolutional neural network and random forest", Remote Sensing Letters, Jul 2019.

[12] A. Ostovar, B. Talbot, S. Puliti, R. Astrup, and O. Ringdahl, "Detection and classification of Root and Butt-Rot (RBR) in Stumps of Norway Spruce Using RGB Images and Machine Learning", Sensors, Apr. 2019.

[13] C. Sothe, C. M. De Almeida, M. B. Schimalski, V. Liesenberg, L. E. C. La Rosa, J. D. B. Castro and R. Q. Feitosa, "A comparison of machine and deep-learning algorithms applied to multisource data for a subtropical forest area classfication", International Journal Of Remote Sensing, Aug. 2019.

[14] J. D. Sylvain, G. Drolet, N. Brown, "Mapping dead forest cover using a deep convolutional neural network and digital aerial photography", ISPRS Journal of Photogrammetry and Remote Sensing, Oct. 2019.

[15] J. F. Abrams, A. Vashishtha, S. T. Wong, A. Nguyen, A. Mohamed, S. Wieser, A. Kuiper, A. Wilting, A. Mukhopadhyay, “Habitat-Net: Segmentation of habitat images using deep learning”, Ecological Informatics, May 2019.

[16] J. Liu, X. Wang, T. Wang, “Classification of tree species and stock volume estimation in ground forest images using Deep Learning”, Computers and Electronics in Agriculture, Nov. 2019.

[17] Z. Zhang, H. Liu, Z. Meng, and J. Chen, “Deep learning-based automatic recognition network of agricultural machinery images”, Computers and Electronics in Agriculture, Nov. 2019.

[18] F. Rosenblatt, “The Perceptron a Perceiving and Recognizing Automaton (Project Para)”, Cornell Aeronautical Laboratory, Inc., Jan. 1957.

[19] Internet: K. Grosser, “The Acceleration of AI”, https://blog.advantech.com/sites/iiot-us/the-acceleration-of-ai/, 15.06.2021.

[20] I. N. Aizenberg, N. N. Aizenberg, and J. Vandewalle, “Multiple-Valued Threshold Logic and Multi-Valued Neurons,” in Multi-Valued and Universal Binary Neurons, Boston, MA: Springer US, 2000, pp. 25–80.

[21] G. E. Hinton, “Learning multiple layers of representation,” Trends Cogn. Sci., 11(10), pp. 428–434, Oct. 2007.

[22] G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. R. Salakhutdinov, “Improving Neural Networks by Preventing Co-adaptation of Feature Detectors,” Neural Evol. Comput., Jul. 2012.

[23] Internet: “Sinir Hücresi (Nöron) ve Yapısı”, https://www.fenokulu.net/yeni/Fen-Konulari/Konu/Sinir-Hucresi-Noron-ve-Yapisi_0_1617.html, 17.02.2020.

[24] Internet: A. Karpathy, Stanford University, Stanford CS class CS231n: “Convolutional Neural Networks for Visual Recognition”, https://cs231n.github.io/, 17.02.2020.

[25] Internet: A. Honkela, University of Helsinki, Nonlinear Switching State-Space Models, “Multilayer perceptrons”, http://users.ics.aalto.fi/ahonkela/dippa/node41.html, 17.02.2020.

[26] Internet: A. Kızrak, “ŞU KARA KUTUYU AÇALIM: Yapay Sinir Ağları”, https://medium.com/@ayyucekizrak/%C5%9Fu-kara-kutuyu-a%C3%A7alim-yapay-sinir-a%C4%9Flar%C4%B1-7b65c6a5264a, 18.02.2020.

[27] Internet: A. Amidi, S.Amidi, Stanford University, “CS230 – Deep Learning: Convolutional Neural Networks cheatsheet”, https://stanford.edu/~shervine/teaching/cs-230/cheatsheet-convolutional-neural-networks, 03.06.2021.

[28] Internet: N. Çarkacı, “Derin Öğrenme Uygulamalarında En Sık kullanılan Hiper-parametreler”, https://medium.com/deep-learning-turkiye/derin-ogrenme-uygulamalarinda-en-sik-kullanilan-hiper-parametreler-ece8e9125c4, 21.02.2020.

[29] Internet: A. Amidi, S.Amidi, Stanford University, “CS229 – Machine Learning: Deep Learning cheatsheet”, https://stanford.edu/~shervine/teaching/cs-229/cheatsheet-deep-learning, 03.06.2021.

[30] Y. LeCun, L. Bottou, Y. Bengio ve P. Haffner, “Gradient-Based Learning Applied to Document Recognition”, Proceeding of the IEEE, 86 (11), 2278-2324, 1998.

[31] A. Krizhevsky, I. Sutskever, G. E. Hinton, “ImageNet Classification with Deep Convolutional Neural Networks”, Advances in Neural Information Processing Systems25 (NIPS’12), 2012.

[32] M. D. Zeiler, R. Fergus, “Visualizing and Understanding Convolutional Networks”, European Conference on Computer Vision (ECCV’14), 818-833, 2013.

[33] K. He, X. Zhang, S. Ren, J. Sun, “Deep Residual Learning for Image Recognition”, In IEEE Conference on Computer Vision and Pattern Recognition (CVPR’15), 770-778, 2015.

[34] M. Lin, Q. Chen, S. Yan, “Network in Network”, arXiv:1312.4400, 2014.

[35] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, A. Rabinovich, “Going Deeper with Convolutions”, In IEEE Conference on Computer Vision and Pattern Recognition (CVPR’15), 1-9, 2015.

[36] Internet: A. Amidi, S.Amidi, Stanford University, “CS230 – Deep Learning: Tips and Tricks cheatsheet”, https://stanford.edu/~shervine/teaching/cs-230/cheatsheet-deep-learning-tips-and-tricks, 03.06.2021.

[37] S. Ruder, “An Overview of Gradient Descent Optimization Algorithms”, arXiv:1609.04747, 2017.

[38] O. C. Bayrak, “Tıbbi Görüntülerden Derin Öğrenme Yöntemi ile Karaciğer ve Beyin Lezyonlarının Bölütlenmesi”, Kasım, 2019.

[39] O. Ronneberger, P. Fischer, T. Brox, “U-Net: Convolutional Networks for Biomedical Image Segmentation”, arXiv:1505.04597, 2015.

[40] J. Xiao, J. Wang, S. Cao and B. Li, “Application of a Novel and Improved VGG-19 Network in the Detection of Workers Wearing Masks”, Journal of Physics: Conferecence Series, 2, 2020.

[41] Z. Zhang, Q. Liu, “Road Extraction by Deep Residual U-Net”, arXiv:1711.10684, 2017.

[42] S. Safarov, T. K. Whangbo, “A-DenseUNet: Adaptive Densely Connected UNet for Polyp Segmentation in Colonoscopy Images with Atrous Convolution”, Sensors, Feb. 2021.

[43] H. Li, P. Xiong, J. Wang, “Pyramid attention network for semantic segmentation”, arXiv:1805.10180, 2018.

[44] N. Ibtehaz, M. S. Rahman, “MultiResUNet: Rethinking the U-Net Architecture for Multimodal Biomedical Image Segmentation”, arXiv:1902.04049, 2019.

[45] O. Ronneberger, P. Fischer, T. Brox, “U-Net: Convolutional Networks for Biomedical Image Segmentation,” in International Conference on Medical Image Computing and Computer-Assisted Intervention, Oct. 2015, pp. 234–241.

[46] Internet: S. Narkhede, “Understanding Confusion Matrix”, https://towardsdatascience.com/understanding-confusion-matrix-a9ad42dcfd62, 20.06.2021.

[47] M. Uzar Dinlemek, “Otomatik Bina Çıkarım Uygulamalarında Çoklu Algılama Sistemi Verilerinin Kullanım Olanaklarının Analizi”, Doktora tezi, 2012, s.49.

[48] F. Rottensteiner, J. Trinder, S. Clode ve K. Kubik, “Building Detection By Fusion of Airborne Laser Scanner Data and Multi Spectral Images: Performance Evaluation and Sensitivity Analysis”, ISPR Journal of Photogrammetry and Remote Sensing, 2007, 135-149.

[49] M. Rutzinger, F. Rottensteiner, F. Ve N. Pfeifer, “A Comparison of Evaluation Techniques for Building Extraction from Airborne Laser Scanning”, IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, 2009, 1:11-20.

[50] M. Khun, K. Johnson, “Applied Predictive Modeling”, New York: Springer.

TEZDEN ÜRETİLMİŞ YAYINLAR

Konferans Bildirileri

1. Y. Y. Güzel, A. M. Uzar Dinlemek, N. Tunalıoğlu, Derin Öğrenme ile Yeşil Alanların Çıkarımı, ISPEC 9.Uluslararası Mühendislik ve Fen Bilimleri Kongresi, s. 304-305, 2020.