



**PİRİNÇ YAPRAK HASTALIKLARININ
SINIFLANDIRILMASINDA DERİN EVRİŞİMLİ
SİNİR AĞI MODELLERİNİN
KARŞILAŞTIRMALI ANALİZİ**

Erkan VEZİROĞLU

Danışman: Dr. Öğr. Üyesi Ahmet COŞKUNÇAY

Yüksek Lisans Tezi

Bilgisayar Mühendisliği Ana Bilim Dalı

2023

(Her hakkı saklıdır.)

T.C.
ATATÜRK ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ
BİLGİSAYAR MÜHENDİSLİĞİ ANA BİLİM DALI

**PİRİNÇ YAPRAK HASTALIKLARININ SINIFLANDIRILMASINDA DERİN
EVRIŞİMLİ SİNİR AĞI MODELLERİNİN KARŞILAŞTIRMALI ANALİZİ**
(Comparative Analysis of Deep Convolutional Neural Network Models for Classification of
Rice Leaf Diseases)

YÜKSEK LİSANS TEZİ

Erkan VEZİROĞLU

Danışman: Dr. Öğr. Üyesi Ahmet COŞKUNÇAY
İkinci Danışman: Dr. Öğr. Üyesi İshak PAÇAL

Erzurum
Nisan, 2023



FEN BİLİMLERİ ENSTİTÜSÜ
Graduate School of Natural and
Applied Sciences

T.C.
ATATÜRK ÜNİVERSİTESİ
Fen Bilimleri Enstitüsü Müdürlüğü
TEZ KABUL VE ONAY TUTANAĞI

**PIRİNÇ YAPRAK HASTALIKLARININ SINIFLANDIRILMASINDA DERİN EVRİŞİMLİ
SİNİR AĞI MODELLERİNİN KARŞILAŞTIRMALI ANALİZİ**

Dr. Öğr. Üyesi Ahmet COŞKUNÇAY ve Dr. Öğr. Üyesi İshak PAÇAL danışmanlığında, Erkan VEZİROĞLU tarafından hazırlanan bu çalışma, 17/04/2023 tarihinde aşağıdaki jüri tarafından Bilgisayar Mühendisliği Anabilim Dalı Bilgisayar Mühendisliği Bilim Dalı'nda Yüksek Lisans tezi olarak **oybirliği (5/0)** ile kabul edilmiştir.

Jüri Başkanı:	Doç Dr. Barış ÖZYER <i>Atatürk Üniversitesi</i>	Aslı Islak İmzalıdır
Danışman:	Dr. Öğr. Üyesi Ahmet COŞKUNÇAY <i>Atatürk Üniversitesi</i>	Aslı Islak İmzalıdır
Jüri Üyesi:	Dr. Öğr. Üyesi Mete YAĞANOĞLU <i>Atatürk Üniversitesi</i>	Aslı Islak İmzalıdır
Jüri Üyesi:	Dr. Öğr. Üyesi Serhat KILIÇARSLAN <i>Bandırma Onyedli Eylül Üniversitesi</i>	Aslı Islak İmzalıdır
İkinci Tez Danışmanı	Dr. Öğr. Üyesi İshak PAÇAL <i>Iğdır Üniversitesi</i>	Aslı Islak İmzalıdır

Enstitü Yönetim
Kurulunun/.../.... tarih
ve sayılı kararı.

Bu tezin Atatürk Üniversitesi Lisansüstü Eğitim ve Öğretim Yönetmeliği'nin ilgili maddelerinde belirtilen şartları yerine getirdiğini onaylarım.

Prof.Dr. Saltuk Buğrahan CEYHUN

Enstitü Müdürü

Aslı Islak İmzalıdır

ETİK BİLDİRİM VE İNTİHAL BEYAN FORMU

Yüksek Lisans Tezi olarak *Dr. Öğr. Üyesi Ahmet COŞKUNÇAY* danışmanlığında sunulan “Pirinç Yaprak Hastalıklarının Sınıflandırılmasında Derin Evrişimli Sinir Ağı Modellerinin Karşılaştırmalı Analizi” başlıklı çalışmanın tarafımızdan bilimsel etik ilkelere uyularak yazıldığını, yararlanılan eserlerin kaynakçada gösterildiğini, Fen Bilimleri Enstitüsü tarafından belirlenmiş olan Turnitin Programı benzerlik oranlarının aşılmadığını ve aşağıdaki oranlarda olduğunu beyan ederiz.

Tez Bölümleri	Tezin Benzerlik Oranı (%)	Maksimum Oran (%)
Giriş	0	30
Kuramsal Temeller	0	30
Materyal ve Metot	1	35
Araştırma Bulguları ve Tartışma	1	20
Sonuçlar	3	20
Tezin Geneli	5	25

Not: Yedi kelimeye kadar benzerlikler ile Başlık, Kaynakça, İçindekiler, Teşekkür, Dizin ve Ekler kısımları tarama dışı bırakılabilir. Yukarıdaki azami benzerlik oranları yanında tek bir kaynaktan olan benzerlik oranlarının %5'den büyük olmaması gerekir.

Beyan edilen bilgilerin doğru olduğunu, aksi halde doğacak hukuki sorumlulukları kabul ve beyan ederiz.

Tez Yazarı (Öğrenci)	Tez Danışmanı
Erkan VEZİROĞLU	Dr. Öğr. Üyesi Ahmet COŞKUNÇAY
3.4.2023	3.4.2023
İmza: Aslı Islak İmzalıdır	İmza: Aslı Islak İmzalıdır

* Tez ile ilgili YÖKTEZ’de yayınlamasına ilişkin bir engelleme var ise aşağıdaki alanı doldurunuz.

☐ Tezle ilgili patent başvurusu yapılması / patent alma sürecinin devam etmesi sebebiyle Enstitü Yönetim Kurulunun/....../.... tarih ve sayılı kararı ile teze erişim 2 (iki) yıl süreyle engellenmiştir.

☐ Enstitü Yönetim Kurulunun/....../.... tarih ve sayılı kararı ile teze erişim 6 (altı) ay süreyle engellenmiştir.

TEŐEKKÖR

Öncelikle, eğitim sürecim boyunca her zaman bilgi ve tecrübeleriyle destekçim olan danışmanım Sayın Dr. Öğr. Üyesi Ahmet COŐKUNÇAY'a,

Tez çalışmasında bana sağladığı rehberlik, yönlendirme ve destek için 2. danışmanım Sayın Dr. Öğr. Üyesi İshak PAÇAL'a,

Her zaman yardıma hazır olan ve değerli önerileriyle beni motive eden Dr. Öğr. Üyesi Mete YAĞANOĞLU'na,

Ve son olarak beni her zaman destekleyen, bana inanan ve her türlü desteklerini esirgemeyen aileme ve sevdiklerime

Özel teşekkürlerimi sunarım.

Erkan VEZİROĞLU

ÖZET

YÜKSEK LİSANS TEZİ

PİRİNÇ YAPRAK HASTALIKLARININ SINIFLANDIRILMASINDA DERİN EVRİŞİMLİ SINIR AĞI MODELLERİNİN KARŞILAŞTIRMALI ANALİZİ

Erkan VEZİROĞLU

Danışman: Dr. Öğr. Üyesi Ahmet COŞKUNÇAY

İkinci Tez Danışmanı: Dr. Öğr. Üyesi İshak PAÇAL

Amaç: Bu çalışmanın amacı, pirinç bitkilerindeki yaprak hastalıklarını tespit etmek için farklı evrişimli sinir ağı (CNN) modellerinin performansını araştırmak ve karşılaştırmaktır. Çalışma, VGG, ResNet, DenseNet, Xception ve Inception gibi popüler CNN modellerinin performansını karşılaştırmaya ve Paddy Doctor veri kümesini kullanarak on adet pirinç yaprak hastalığının teşhisinde en etkili CNN modelini belirlemeye odaklanmaktadır.

Yöntem: Bu çalışmada çeltik yaprak hastalıklarının tespit edilmesi ve sınıflandırılması için derin öğrenme tabanlı bir yaklaşım geliştirilmiştir. Çalışmanın temel bileşenleri, veri seti, veri işleme aşaması ve öğrenme yaklaşımlarının olduğu aşamadır. 10.407 görüntüden oluşan veri seti rastgele eğitim (%70), doğrulama (%15) ve test (%15) verileri olarak üç kısma ayrılmıştır. Veri setine temel veri artırma teknikleri uygulanmış ve öğrenme aktarımı ile 20 CNN modeli test edilmiştir.

Bulgular: Nesne sınıflandırması için literatürde yaygın olarak kullanılan CNN modelleri eğitilmiş ve değerlendirilmiştir. Tüm modellerin %95'in üzerinde doğruluğa sahip olduğu görülmüştür. En yüksek doğruluğa sahip model EfficientNetv2-Small modelidir (%98.01 doğruluk, %97.99 F1-Skor) ve en düşük doğruluğa sahip model ise Inception-v3'tür (%95.77 doğruluk, %96.12 F1-Skor). Bu çalışma, çeltik yaprağı hastalıklarının erken tespiti için CNN tabanlı modellerin kullanımının potansiyel olarak yararlı olabileceğini göstermektedir.

Sonuç: Bu çalışma, çeltik yaprak hastalıklarının tespiti için birçok derin öğrenme yaklaşımı kullanılarak oluşturulan modellerin test edilmesiyle gerçekleştirilmiştir. Çalışma sonuçlarına göre, EfficientNetv2-Small modeli tüm çeltik hastalıklarını %98.01 test doğruluğu ile doğru bir şekilde teşhis edebilmiştir. Bu sonuçlar, bilgisayar destekli teşhis sistemlerinin çeltik hastalıklarının erken teşhisi için etkili bir yöntem olabileceğini göstermektedir. Ayrıca, bu çalışmanın literatürdeki diğer çalışmalardan daha kapsamlı ve etkili olduğu belirtilmektedir. Bu çalışmanın sonuçları, çiftçiler ve ziraat mühendislerinin gelecekteki tarımsal problemlerde bilgisayar destekli teşhis sistemlerine daha fazla güvenebileceklerini göstermektedir.

Anahtar Kelimeler: Bitki hastalığı sınıflandırması, Pirinç hastalığı tespiti, Evrişimli sinir ağları, Derin öğrenme, Öğrenme aktarımı

Nisan 2023, 109 sayfa

ABSTRACT

MASTER'S THESIS

COMPARATIVE ANALYSIS OF DEEP EVRİŞİMLİ NEURAL NETWORK MODELS FOR CLASSIFICATION OF RICE LEAF DISEASES

Erkan VEZİROĞLU

Supervisor: Assist Prof. Dr. Ahmet COŞKUNÇAY

Co-supervisor: Assist Prof. Dr. İshak PAÇAL

Purpose: The aim of this study is to investigate and compare the performance of different convolutional neural network (CNN) models for detecting leaf diseases in rice plants. The study focuses on comparing the performance of popular CNN models such as VGG, ResNet, DenseNet, Xception, and Inception, and determining the most effective CNN model for diagnosing ten rice leaf diseases using the Paddy Doctor dataset.

Method: In this study, a deep learning-based approach was developed for the detection and classification of rice leaf diseases. The main components of the study are the Dataset unit, the Data Processing stage, and the Learning Approaches unit. The dataset, consisting of 10,407 images, was divided into three parts as random training (70%), validation (15%), and test (15%) data. Basic data augmentation techniques were applied to the dataset, and 20 CNN models were tested using transfer learning.

Findings: Commonly used CNN models for object classification have been trained and evaluated in this study. It was found that all models had an accuracy of over 95%. The model with the highest accuracy is EfficientNetv2-Small (98.01% accuracy, 97.99% F1-Score), and the model with the lowest accuracy is Inception-v3 (95.77% accuracy, 96.12% F1-Score). This study demonstrates the potential usefulness of using CNN-based models for early detection of rice leaf diseases.

Results: This study was conducted by testing models created using various deep learning approaches for detecting rice leaf diseases. According to the results, the EfficientNetv2-Small model was able to accurately diagnose all rice diseases with a 98.01% test accuracy. These results demonstrate that computer-aided diagnosis systems can be an effective method for early diagnosis of rice diseases. Additionally, this study is noted to be more comprehensive and effective than other studies in the literature. The results of this study indicate that farmers and agricultural engineers can rely more on computer-aided diagnosis systems for future agricultural problems.

Keywords: Crop disease classification, Rice disease detection, Convolutional neural networks, Deep learning, Transfer learning

April 2023, 109 page

İÇİNDEKİLER

KABUL VE ONAY TUTANAĞI	i
ETİK BİLDİRİM VE İNTİHAL BEYAN FORMU.....	ii
TEŞEKKÜR.....	iii
ÖZET	iv
ABSTRACT.....	v
İÇİNDEKİLER	vi
TABLolar DİZİNİ.....	ix
ŞEKİLLER DİZİNİ	x
KISALTMALAR VE SİMGELER DİZİNİ	xii
GİRİŞ	1
Tezin Amacı.....	2
Tezin Katkıları	2
KURAMSAL TEMELLER	4
Temel Kavramlar	6
Yapay zeka.....	7
Makine öğrenmesi.....	9
Denetimli Öğrenme (Supervised Learning).....	9
Denetimsiz Öğrenme (Unsupervised Learning)	10
Yarı denetimli öğrenme	10
Pekiştirmeli Öğrenme (Reinforcement Learning).....	11
Derin Öğrenme.....	12
Derin öğrenme nedir?.....	12
Derin öğrenmenin tarihçesi	13
Yapay sinir ağları	14
İleri beslemeli sinir ağları (FNN).....	14
Tekrarlayan sinir ağları (RNN).....	15
Uzun kısa süreli hafıza ağları (LSTM)	15
Sınırlı boltzmann makineleri (RBM)	15
Derin inanç ağları (DBN).....	16
Oto-kodlayıcılar (AEs).....	16
Evrişimli sinir ağları (CNNs).....	16
Giriş görüntüsü	17

<i>Evrişimli katman</i>	18
<i>Havuzlama katmanı</i>	22
<i>Aktivasyon katmanı</i>	23
<i>Tam bağlantı katmanı</i>	25
<i>CNN yapılarının düzenlenmesi</i>	26
<i>Birkaç CNN mimarisi</i>	28
MATERYAL VE METOT	33
Veri Seti	33
Bakteriyel yaprak yanıklığı (Bacterial Leaf Blight (BLB))	35
Bakteriyel yaprak lekesi (Bacterial Leaf Streat (BLS))	36
Bakteriyel başak yanıklığı (Bacterial Panicle Blight (BPB)).....	36
Blast	36
Kahverengi leke (Brown Spot)	37
Ölü kalp (Dead Heart).....	37
Tüylü küf (Downy Mildew).....	37
Hispa	37
Tungro.....	39
Ziraat Alanında Derin Öğrenme	39
Derin Öğrenme Mimarileri	40
Derin Öğrenme Modelleri	40
DenseNet.....	41
EfficientNetv2	42
Inception.....	45
MobileNetv3	47
ResNet.....	49
ResNeXt.....	50
VGG	52
VovNet	54
Xception	56
CNN modellerinin genel görünümü.....	57
Veri Ön İşleme ve Artırma Teknikleri	58
Görüntü ön işleme	58
<i>Görüntü yeniden boyutlandırma</i>	59
<i>Kontrast artırma</i>	59
Veri artırma	59

Öğrenme Aktarımı	60
Önerilen Yaklaşım	62
ARAŞTIRMA BULGULARI.....	63
Uygulama Detayları	63
Değerlendirme Metrikleri	63
Deneysel Sonuçlar.....	64
Bulgular.....	65
Önerilen Yaklaşımın Diğer Çalışmalarla Karşılaştırılması	80
SONUÇLAR VE ÖNERİLER.....	82
KAYNAKLAR	84
ÖZGEÇMİŞ	95



TABLÖLAR DİZİNİ

Tablo 1. Farklı Popüler CNN Mimarileri Arasında Bir Karşılaştırma.....	30
Tablo 2. Veri Setindeki Çeltik Hastalıklarına İlişkin İstatistiksel Veriler	35
Tablo 3. DenseNet Modellerinin Karşılaştırılması	42
Tablo 4. EfficientNetv2 Modellerinin Karşılaştırılması	45
Tablo 5. Inception Modellerinin Karşılaştırılması	46
Tablo 6. MobileNetv3 Modellerinin Karşılaştırılması	48
Tablo 7. ResNet Modellerinin Karşılaştırılması.....	50
Tablo 8. ResNeXt Modellerinin Karşılaştırılması	52
Tablo 9. VGG Modellerinin Karşılaştırılması	53
Tablo 10. VovNet Modellerinin Karşılaştırılması	55
Tablo 11. Modellerinin Genel Karşılaştırılması.....	58
Tablo 12. CNN Modellerinin Test Sonuçları	65
Tablo 13. Önerilen Yaklaşımın Diğer Çalışmalarla Karşılaştırılması	80

ŞEKİLLER DİZİNİ

Şekil 1. Yapay zeka, makine öğrenmesi ve derin öğrenme süreçleri	7
Şekil 2. Yapay zeka alt alanları	9
Şekil 3. Makine öğrenimi ana kategorileri	9
Şekil 4. Denetimli öğrenme süreci	9
Şekil 5. Denetimsiz öğrenme süreci	10
Şekil 6. Yarı denetimli öğrenme süreci	10
Şekil 7. Pekiştirmeli öğrenme süreci	11
Şekil 8. Makine öğrenimi türleri.....	11
Şekil 9. Derin öğrenme ve makine öğrenimi türlerinin gösterimi	13
Şekil 10. CNN mimarisi	16
Şekil 11. CNN bileşenleri.....	17
Şekil 12. Temel hesaplamaların görsel bir temsili	18
Şekil 13. Farklı evrişimli matrislerin etkileri.....	20
Şekil 14. Filtre pencerelerinin her bağlantı için hareketi.....	20
Şekil 15. Çıktı üzerinde adım büyüklüğünün etkisi	21
Şekil 16. Sıfır dolgu (Zero-padding)	22
Şekil 17. Havuzlama katmanı.....	23
Şekil 18. Aktivasyon fonksiyonu.....	24
Şekil 19. Tam bağlantılı katman.....	25
Şekil 20. Gizli CNN katmanlarındaki ileri ve geriye yayılım süreçler	26
Şekil 21. Aşırı uydurmaya karşı düzenleme tekniklerinin CNN'e uygulanması.....	27
Şekil 22. Veri toplama ve ek açıklama iş akışı.....	33
Şekil 24. Veri setinde bir hastalık sınıfına ait görüntüler	38
Şekil 23. Her sınıf için veri yüzdeleri.....	39
Şekil 25. DenseNet mimarisi	41
Şekil 26. Üç yoğun bloğa sahip derin bir DenseNet.....	42
Şekil 27. EffigentNetV2 mimarileri	43
Şekil 28. Inception-v3 modelinin görünümü	46
Şekil 29. MobileNetV3 modülü.....	47
Şekil 30. ResNet mimarisi	49
Şekil 31. ResNext yapısı.....	51
Şekil 32. VGG sinir ağı mimarisi	52

Şekil 33. VoVNet mimarisi	54
Şekil 34. Xception modül yapısı.....	56
Şekil 35. Kahverengi lekesi hastalığına ait artırılmış görüntüler	60
Şekil 36. Öğrenme aktarımı.....	61
Şekil 37. Çeltik yaprağı hastalığı tespiti için genel yaklaşım.....	62
Şekil 38. Tüm modellere ait doğruluk değerleri.....	66
Şekil 39. Tüm modellere ait F1-Skor değerleri	67
Şekil 40. EfficientNetv2-Small modeline ait karışıklık matrisi	68
Şekil 41. Inception-v3 modeline ait karışıklık matrisi	69
Şekil 42. Modellere ait karışıklık matrisleri (1)	72
Şekil 43. Modellere ait karışıklık matrisleri (2)	73
Şekil 44. Modellere ait karışıklık matrisleri (3)	74
Şekil 45. Çoklu sınıflandırma ROC eğrileri	75
Şekil 46. Modellere ait Doğruluk, Eğitim Kaybı ve Doğrulama Kaybı grafikleri (1) ...	76
Şekil 47. Modellere ait Doğruluk, Eğitim Kaybı ve Doğrulama Kaybı grafikleri (2) ...	77
Şekil 48. Modellere ait Doğruluk, Eğitim Kaybı ve Doğrulama Kaybı grafikleri (3) ...	78
Şekil 49. Modellere ait Doğruluk, Eğitim Kaybı ve Doğrulama Kaybı grafikleri (4) ...	79

KISALTMALAR VE SİMGELER DİZİNİ

AEs	: Otomatik Kodlayıcılar (Autoencoders)
AI	: Yapay Zeka (Artificial Intelligence)
AutoML	: Otomatik Makine Öğrenimi ()
BLB	: Bakteriyel Yaprak Yanıklığı (Bacterial Leaf Blight)
BLS	: Bakteriyel Yaprak Lekesi (Bacterial Leaf Streat)
BPB	: Bakteriyel Başak Yanıklığı (Bacterial Panicle Blight)
CNN	: Evrişimli Sinir Ağı (Convolution Neural Network)
CPU	: Central Processing Unit (Merkezi İşlem Birimi)
DAE	: Gürültü Giderici Otokodlayıcılar (Denoising Autoencoders)
DBN	: Derin İnanç Ağları (Deep Belief Networks)
DL	: Derin Öğrenme (Deep Learning)
DNN	: Derin Sinir Ağları (Deep Neural Networks)
FNN	: İleri Beslemeli Sinir Ağları (Feedforward Neural Networks)
GAP	: Küresel Ortalama Havuzlama
GNDVI	: Green NDVI
GPU	: Grafik İşlem Birimi (Graphics Processing Unit)
LSTM	: Uzun Kısa Süreli Bellek (Long Short-Term Memory)
MAE	: Ortalama Hata (Mean Average Error)
ML	: Makine Öğrenmesi (Machine Learning)
MLP	: Çok Katmanlı Algılayıcı (Multi-Layer Perceptron)
nNDVI	: Yapay Sinir Ağı Temelli NDVI (Neural network based NDVI)
PdNP	: Desene Bağımlı Gürültü Tahmini
RBM	: Sınırlı Boltzmann Makineleri (Restricted Boltzmann Machines)
ReLU	: Doğrultulmuş Lineer Birim (Rectifier Linear Unit)
RNN	: Rekürsif/Tekrarlayan Sinir Ağları (Recurrent Neural Networks)
YSA	: Yapay Sinir Ağı (Artificial Neural Network)

GİRİŞ

Tarım, yaşamın temel gereksinimlerini sağlayan önemli bir küresel ekonomi alanıdır (Malhi vd 2021). Tarımsal faaliyetler insan hayatının sürdürülmesi için hayati öneme sahiptir ve dünya nüfusunun 2050 yılına kadar 9 milyarı aşmasıyla birlikte gıda güvenliği ve sürdürülebilir tarım, dünya genelinde büyük bir endişe kaynağı haline gelmiştir (Tripathi vd 2019). Bitki hastalıkları, haşereler ve diğer çevresel faktörlerin ürün verimi ve kalitesi üzerindeki etkisi sürdürülebilir ve akıllı tarım yöntemleri için önemli bir zorluk teşkil etmektedir. Bu nedenle, tarımda otomatik algılama ve tanımlama yöntemleri kullanmak, bir dizi önemli avantaj sağlayabilir (Udotalapally vd 2020).

Tarım uzmanları ve uygulayıcıları, bitki hastalıklarını daha etkili bir şekilde tanımlayarak ve tedavi ederek yüksek kaliteli mahsul üretimini sürdürmeye yardımcı olan derin öğrenme yöntemlerini kullanabilirler. Bu yöntemlerin kullanılması, mahsul yönetimi uygulamalarının geliştirilmesi, tedavilerin ve müdahalelerin hedefe yönelik uygulanması, pestisit kullanımının azaltılması ve sonuç olarak mahsul veriminin ve kalitesinin artırılması gibi birçok fayda sağlayabilir. Doğru ve etkili modellerin geliştirilmesi, bitki hastalıklarının tespiti ve sınıflandırılması için hayati öneme sahiptir. Bu nedenle, bitki hastalıklarının tanımlanması ve tedavisinde derin öğrenme yöntemlerinin kullanımı, tarım endüstrisi için önemli bir araçtır (Waheed vd 2020; An vd 2022).

Çeltik, yetiştiricilikte en önemli ve pratik ürünlerden biridir. Bir ülkenin ekonomisini büyütmek için tarımsal hasat verimi önemlidir. Mahsul büyümesi ve verimi üzerinde, hastalıklar, haşereler ve çevre koşulları gibi değişkenler etkili olabilir. Mahsul hastalığı, ürün kalitesini önemli ölçüde düşüren bir faktördür (Martinelli vd 2015; van Eeuwijk vd 2019; Kaur ve Gautam 2021). Bitkilerdeki yaprak enfeksiyonlarının tespiti, ürün verimindeki kayıpları azaltabilir. Hastalıkların erken ve doğru teşhisi, başlangıç aşamasında kontrol etmenin en etkili yoludur. Yaprak dokusu temel alarak, bitki yaprak hastalıklarının manuel kontrol yoluyla teşhisi yapıldı (Sethy vd 2020b).

Daha önce, hastalık tespiti görevi için eğitimli ve uzman kişilere ihtiyaç duyuluyordu ve bu görevin tamamlanması uzun zaman almaktaydı. Bu sebeple daha verimli bir hastalık tespit yöntemi aranmaktaydı. Ancak son yıllarda, araştırmacılar, verim tahmini, besin eksikliklerinin tanınması, ürün boyutunun ölçülmesi ve yabancı ot tespiti gibi farklı sorunları ele almak için bilgisayar görüşü ve görüntü işleme teknolojilerini kullanmaktadırlar.

Derin öğrenme teknolojisi, yaprak problemlerini tanımlamada yüksek bir başarı oranına sahiptir. Pirinç tarlalarında çalışılan hastalıklar arasında pirinç yaprak patlaması, yanılıcı küf,

boyun patlaması, kılıf yanıklığı, bakteriyel yaprak çizgisi ve kahverengi leke bulunur. Bu çalışmanın amacı, pirinç hastalıklarını sınıflandırmak için derin öğrenme tekniklerini kullanmaktır. Özellikle, Inception, Xception EfficientNet, MobileNet, VGG ve ResNet gibi en yeni aktarma öğrenme modelleri, pirinç hastalıklarını doğru bir şekilde sınıflandırmak için kullanılmıştır.

Çalışmamız, bitki hastalıklarının sınıflandırılması için derin öğrenme yöntemlerinin kullanımını hakkındaki artan bilgi birikimine katkıda bulunur ve sürdürülebilir ve akıllı tarım uygulamaları için doğru hastalık teşhisinin önemini vurgular. Bu araştırmanın motivasyonu, çiftçilere ve tarım uzmanlarına pirinç hastalıklarını doğru teşhis etmek ve sınıflandırmak için pratik ve etkili araçlar sağlamaktır. Sağlanan araçlar, ürün yönetimini geliştirebilir ve verimliliği artırabilir.

Tezin Amacı

Bu tez, pirinç bitkilerindeki yaprak hastalıklarını tespit etmek için farklı evrişimli sinir ağı (CNN) modellerinin performansını araştırmayı ve karşılaştırmayı amaçlamaktadır. Bu doğrultuda, VGG, ResNet, DenseNet, EfficientNet, Xception ve Inception gibi popüler CNN modellerinin performansını karşılaştırarak, Paddy Doctor (Kiruba ve Arjunan 2023) veri kümesinde bulunan on adet pirinç yaprak hastalığının teşhisinde en etkili CNN modelini belirlemeye odaklanmaktadır. Ayrıca, bu çalışma, CNN tabanlı mimarilerin pirinç yaprak hastalıklarının erken teşhisinde ve gelecekteki diğer tarımsal problemlerde tarım mühendisleri ve çiftçiler için değerli bir araç olabileceğini göstermeyi ve tarım sektöründe önemli bir soruna çözüm sunmayı hedeflemektedir.

Tezin Katkıları

Bu tez, derin öğrenme tekniklerini kullanarak pirinç bitkilerindeki yaprak hastalıklarının etkili bir şekilde tespiti için bir yöntem önererek tarım mühendisliği ve bilgisayar görüşü alanlarına katkı sağlamaktadır.

Çalışma, en popüler evrişimli sinir ağı modellerini kullanarak on farklı pirinç yaprak hastalığı sınıfının teşhis performansını karşılaştırmaktadır. Araştırma, %98.01 doğruluğu ve %97.99 F1-Skor değerleri ile EfficientNetv2-Small modelinin diğer tüm modellere göre daha iyi performans gösterdiğini bulmuştur.

Bu yöntem, geleneksel teşhis yöntemlerine göre daha verimli ve düşük maliyetli bir çözüm sunarak, bilgisayar destekli sistemlerin çiftçilere ve tarım mühendislerine erken dönemde bitki hastalıklarının teşhisinde kolaylık sağlamaktadır. Ayrıca çalışma, bitki

hastalıklarının tespitinde derin öğrenme modellerinin performansını artırmada görüntü dönüştürücü modellerinin önemini vurgulamaktadır.

Çalışmanın bulguları, tarım mühendisliği alanının ilerlemesine katkı sağlayarak, bitki hastalıklarının teşhisi için gelecekteki bilgisayar destekli sistemlerin geliştirilmesine ve tarım uygulamalarının iyileştirilmesine yardımcı olacaktır.



KURAMSAL TEMELLER

Bitki yaprak hastalıklarının tanınması (Xu vd 2011; Baresel vd 2017; Tao vd 2020), tarım açısından önemli bir yarar sağlamaktadır. Ancak, bu görev, tarım uygulamaları için yeterli sayıda yapay zeka aracının olmaması nedeniyle hala sorunlu olmaktadır (Strange ve Scott 2005; Li vd 2009; Islam vd 2018; W. Zhu vd 2018; Gautam 2020). Bu sorunu ele almak için birçok araştırmacı, makine öğrenimi (Gautam 2020), derin öğrenme ve hibrit teknikler (W. Yang vd 2013; Ozguven ve Adem 2019) ve öğrenme aktarımı teknikleri (Szegedy vd 2015; Szegedy vd 2016; Chen vd 2020; Peng vd 2020; Uğuz ve Uysal 2021; Sharma vd 2022) gibi çeşitli yapay zeka tekniklerini kullanmıştır.

Yapılan çalışmalarda, yapay sinir ağı (YSA) tasarımı ile hastalıkların tanımlanmasının en etkili yöntem olduğu görülmüştür. Tasarlanan YSA, dört giriş seviyesi, beş gizli seviye ve bir çıkış seviyesi nörondan oluşmaktadır. Yapılan testlerde, bu mimari, çeltik hastalıklarını tespit etmede %66.3 doğruluk oranına sahiptir (Gunawan vd 2021).

Ozguven ve Adem (2019) adlı araştırmacılar, şeker pancarı yapraklarındaki hastalıkları tespit etmek için güncellenmiş bir Hızlı R-Evrişimli Sinir Ağı (CNN) kullanarak modeli 155 görüntü ile eğitmişler ve %95.48 doğruluk elde etmişlerdir. Soya yapraklarındaki hastalıkların tespiti için, Karlekar ve Seal (Karlekar ve Seal 2020) tarafından iki modüllü bir yaklaşım olan SoyNet kullanılmış, bu yaklaşımda görüntülerin gizli kısımları segmente edilmiş ve daha sonra derin öğrenme tabanlı el yapımı modeller kullanılmıştır. Yapılan başka araştırmalar, PlantVillage veri kümesi kullanılarak domates yaprak hastalığını makul doğrulukla tespit etmek için önceden eğitilmiş derin öğrenme modellerini kullanmışlardır (Rangarajan vd 2018; Agarwal vd 2020; Trivedi vd 2021). Manyok yaprak hastalığını teşhis etmek için Sambasivam ve Opiyo (2021) adlı araştırmacılar bir derin öğrenme yöntemi kullanmışlardır. Lale yapraklarındaki hastalıkları tespit etmek için derin bir evrişimli ağ kullanan başka bir çalışma yapılmıştır (Polder vd 2019).

Başka bir çalışmada lale yapraklarındaki hastalıkları tespit etmek için derin bir evrişimsel ağ kullanılmıştır (Polder vd 2019). Coulibaly vd (Coulibaly vd 2019), mısır bitkilerinde hastalık tespiti için bir yaklaşım önerilmiştir. Tespit oranını artırmak için, MobileNet ile birleştirilmiş derin evrişimli sinir ağı geliştirilmiştir. Yaprak hastalıkları, Exodus bilgisine sahip gelişmiş klasik modeller kullanılarak %92 doğrulukla tespit edilmiştir (Kamal vd 2019). Pirinç yaprak hastalıklarının tespitinde iki aşamalı bir CNN mimarisi kullanılmış ve önerilen bu yaklaşım ile %93.3 doğruluk oranı elde edilmiştir. (Hossain vd 2020). Üç farklı CNN, bir veri kümesi kullanarak bağlamsal olmayan görüntü meta verilerini içermek üzere

önerilmiş ve 79 farklı hastalığa sahip 14 bitkiyi dikkate alan ve daha iyi doğruluk sağlayan yaprak hastalıklarını belirlemek için görüntü sınıflandırması temelli bir yöntem önerilmiştir (Arnal Barbedo 2019). Ek olarak, PlantVillage veri kümesi elma siyah çürüklük hastalığının dört önemli aşamasını tespit etmek için kullanılmıştır (Kovalskaya ve Hammond 2014; G. Wang vd 2017).

Üç farklı evrişimli ve max-pooling katmanının birleştirilmesiyle oluşan bir sinir ağı (CNN), domates tarımında hastalıkların tespiti için geliştirilmiştir. Deneysel sonuçlar, önerilen modelin önceden eğitilmiş modeller olan VGG16, MobileNet ve InceptionV3 ile kıyaslandığında rekabetçi bir performans gösterdiğini göstermiştir. Modelin ortalama doğruluğu %91.2 olarak hesaplanmıştır. Ayrıca, çeltik yapraklarında hastalıkların tespiti için bir görüntü işleme algoritması kullanılmıştır (Sladojevic vd 2016). Yaprakların ve lezyonların boyutu, lezyonların miktarı ve tipi, lezyonların renk özellikleri ve bozulmamış kesitler girdileri kullanarak lezyonları tanımlamak ve kategorize etmek için derin evrişim tabanlı bir CNN modeli kullanmıştır (Fuentes vd 2017). Bu model %96.43'lük bir doğruluk elde etmiştir. Derin öğrenme modelini kullanarak görüntüleri işleyen ve hastalıktan etkilenecek çeltik alanlarını belirlemek için bir çalışmanın özeti sunulmuştur (Narmadha ve Arulvadivu 2017; Atole ve Park 2018; Chawathe 2020). Özellikle çeltik çiftçileri için yapılan bu çalışmada, görüntü sınıflandırmadaki etkinliği sağlamak için CNN'ler kullanılmıştır (Velasca vd 2020). İlave olarak, CNN modellerinden biri, hastalık sınıflandırması için eğitilerek ince ayar ve iki aşamalı yeni bir eğitim yaklaşımı için kullanılmıştır (Masood vd 2020). Ancak, hastalık tahmini için sınıflandırma algoritmalarının doğruluğu farklı girdi parametreleri ile değiştiğinden, bunları kullanmak zordur. Bu yöntemler son yıllarda guava (Srinivas vd 2021), çay (Karmokar vd 2015) ve elma (Baranwal vd 2019) gibi ürünlerde lezyonları görselleştirmek için kullanılmıştır. Ayrıca, Goluguri vd (2021), rüzgar hızı, sıcaklık, yağış ve bağıl nem gibi meteorolojik parametreleri kullanarak çeltik patlama hastalığını tahmin etmek için bir sinir ağı geliştirilmiştir.

Çok sayıda uygulama yüksek performans sonuçları elde etmek için derin öğrenmeyi kullanmıştır. Çeltik yaprak hastalığının tespitinde %74.21 doğruluk elde edilen bulanık sinir ağı tabanlı bir yaklaşım kullanılmıştır (Kahar vd 2015). Hibrit bir yaklaşım kullanılarak çeltik yaprak hastalığı tanımlanmış ve %90 doğruluk sağlanmıştır (Htun ve Htwe 2018; Ganesan ve Chinnappan 2022). İleri beslemeli sinir ağı, yaprak hastalığını %88 doğrulukla tanımlamak için kullanılmıştır (Akila ve Deepan 2018). Optimize edilmiş bir derin sinir ağı ve Jaya algoritması çeltik yaprak hastalıklarını tespit etmek için kullanılmış ve ortalama %93.5 doğruluk elde edilmiştir (Ramesh ve Vydeki 2020). ESforRPD2 adlı bir uzman sistem, çeltik yaprak

hastalıklarını tespit etmek için geliştirilmiş ve sınıflandırma doğruluğu oranı %87.5 olarak elde (Agus vd 2019). Yaprak hastalığının tanınmasında %83,34'lük bir doğruluk oranı elde eden bir başka yaklaşım, radyal fonksiyonlar aracılığıyla çıkarılan ve işlenen hibrit özelliklerin kullanılmasını içermektedir (Verma ve Dubey 2018).

Xiao vd (2018), birincil bileşen analizi (PCA) ve sinir ağı tabanlı bir yaklaşım önermişlerdir ve bu yaklaşım, bakteriyel hastalıkların %95.83 doğrulukla başarılı bir şekilde tanımlanmasını sağlamıştır. Sengupta ve Das (2017), pirinç yapraklarında bakteri ve mantar enfeksiyonlarını tespit etmek için parçacık sürü optimizasyonu (PSO) tabanlı artımlı bir sınıflandırıcı önermişler ve bu yöntem, %84.02 doğruluk oranı ile sonuçlanmıştır. Pirinçte mantar enfeksiyonlarını tespit etmek için hiperspektral veriler kullanılmış ve bu yöntem, %82 doğruluk oranı elde etmiştir (J. Huang vd 2012). Son olarak, desene bağımlı gürültü tahmini (PdNP) sistemi bakteriyel ve fungal enfeksiyonları tespit etmek için kullanılmış ve bu yaklaşım %85 doğruluk oranı elde etmiştir (W. Yang vd 2013).

Öğrenme aktarımı, önceden öğrenilmiş bilginin kullanımıyla önceden eğitilmiş modellerin performansını başka bir görevde iyileştirmeye dayanır. Öğrenme aktarımına dayalı derin öğrenme modelleri, önceden eğitilmiş bir ağdan eğitim verileri kullanarak yaprak hastalığı tahminini %91.50 doğrulukla başarmıştır (Chen vd 2020). Bitki yaprak hastalıklarını tahmin etmek için büyük bir pirinç veri kümesi öğrenme aktarımında kullanılmıştır. Zeytin yaprağı hastalığı tespiti için öğrenme aktarımına dayalı bir derin sinir ağı kullanılarak %88 doğruluk elde edilmiştir (Uğuz ve Uysal 2021). Pirinç yaprağı veri seti kullanılarak yapılan başka bir çalışmada, öğrenme aktarımına dayalı bir makine öğrenimi yaklaşımı kullanılarak yaprak hastalığı tespiti için daha yüksek bir doğruluk oranı elde (Sharma vd 2022). Yaprak hastalığı tespiti için bilgisayar görüşü teknikleri arasında, incelemeli bir ağ kullanılmıştır (Szegedy vd 2015; Szegedy vd 2016) ve öğrenme aktarımına dayalı ResNet kullanılarak %84.3 doğruluk elde edilmiştir (Peng vd 2020)

Temel Kavramlar

Teknolojinin hızla gelişmesiyle birlikte, yapay zekâ kavramı giderek daha önemli hale gelmiştir. Bu yeni teknoloji, insanların uzun sürelerde yapabildiği işleri daha kısa sürede ve daha güvenilir sonuçlarla gerçekleştirebilmektedir. Tarım gibi alanlarda da yapay zeka destekli çözümlerin kullanımı zaman tasarrufu sağlayarak toprağın verimini artırmakta ve işleri büyük ölçüde kolaylaştırmaktadır.

Bu bölümde, çalışmada kullanılan yöntemin daha iyi anlaşılabilmesi için, yapay zeka ile ilgili temel kavramlar hakkında genel bilgiler verilmiştir. Şekil 1'de genelden özele doğru

bir sıralama ile yapay zeka, makine öğrenmesi ve derin öğrenme konuları ayrıntılı bir şekilde açıklanmıştır.

İlk olarak, yapay zekanın ortaya çıkışı, yaygınlaşması, genel kullanım alanları ve kullanılan yöntemler hakkında bilgi verilmiştir. Daha sonra, makine öğrenmesi tanımlanmış ve yöntemleri ile algoritmaları açıklanmıştır. Yapay sinir ağları hakkında ayrıntılı bilgiler verilerek, bu bağlamda kullanılan algoritmaların detaylarına yer verilmiştir. Çalışmada kullanılan CNN yöntemi de ayrıntılı bir şekilde ele alınmıştır. Son olarak, derin öğrenme kavramı ve kullanılan teknikler hakkında bilgi verilmiştir.



Şekil 1. Yapay zeka, makine öğrenmesi ve derin öğrenme süreçleri (mechtekno, 2022)

Yapay zeka

Yapay zeka (Artificial Intelligence), bilgisayar programları aracılığıyla insan zekasını taklit ederek akıllı makinelerin oluşturulmasını sağlayan bir disiplindir. Yapay zeka temel olarak, insanların yaptıklarını taklit ederek hedefe yönelik bir sistem geliştirir ve büyük verileri girdi olarak kullanarak verilen görevleri en ideal performansla yerine getirir (V. Gautam vd 2022).

1940'larda dijital bilgisayarların geliştirilmesiyle, karmaşık görevlerin üstesinden gelmek mümkün hale gelmiştir. Bu görevler arasında, satranç oynamak ve matematiksel modelleri kanıtlamak gibi işlemler bulunur. Yapay zeka terimi ise ilk olarak 1956 yılında Dartmouth Konferansı'nda McCarthy, Minsky, Shannon ve Rochester tarafından kullanılmıştır. Ancak, Turing 1950 yılında yayınladığı makalesinde dijital bilgisayar kavramından bahsetmiş ve Turing Machine ile makinelerin insanların yaptığı tüm işleri yapabileceği olasılığı üzerinde

alışılması gereken bir konu olarak yapay zeka kavramından bahsetmiştir (Hoffmann 2022). McCarthy ve ekibi, konferans sonrasında yayınladıkları bildiride, yapay zeka ve ilgili parametreler üzerinde iki ay boyunca alışacak on kişilik bir ekip oluşturulmasını önermiştir (Alan Mathison Turing 1950). Bu proje, genel kabul görmüş bir uygulama olmaması ve katılımcıların kendi projelerine odaklanmaları nedeniyle beklentilerin altında kalsa da yeni ortaya atılan kavram üzerinde farklı hükümetlerin desteğiyle alışmalar yapılmaya başlanmıştır. Ancak birkaç eleştirel rapor, yapay zeka ve bu konudaki alışmaları eleştirdiği için "AI Winter" veya "Yapay Zeka Kışı" olarak adlandırılan bir dönemi başlatmıştır (McCarthy vd 2006; Yılmaz 2021).

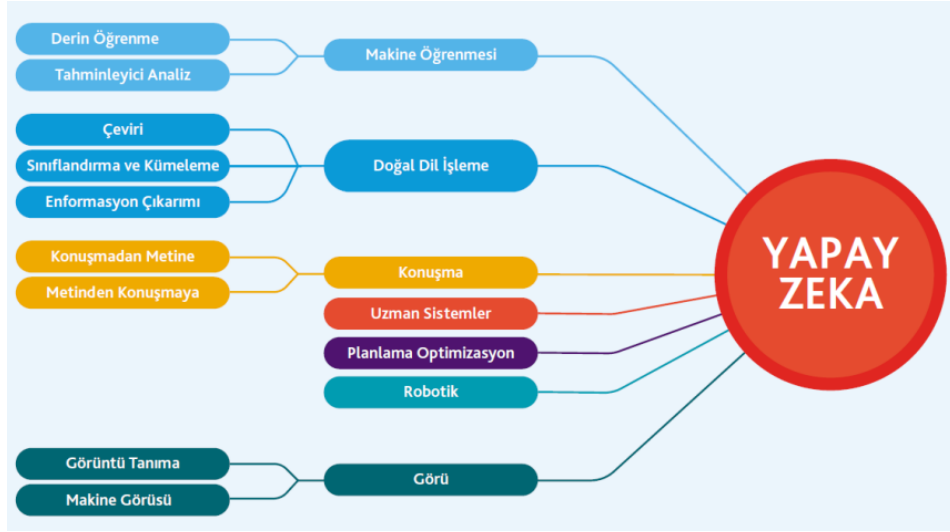
1950'li yıllarda başlayan ilk alışmalar daha ok sembolik yöntemler ve matematiksel problem özme üzerine odaklanmıştı. Ancak 1960'larda ABD Savunma Bakanlığı, olası durumlarda insan hareketlerini tahmin edebilecek sistemler üzerinde alışarak yeni teknolojinin askeri kullanımını geliştirmeye başladı. DARPA, 1970'lerde sokak haritalama projesi ile GPS sisteminin temellerini attı ve 2003'te akıllı kişisel asistanları üreterek alanında bir ilke imza attı. IBM'in 1997 yılında ürettiği Deep Blue isimli bilgisayar, Garry Kasparov'u yenerek tarihe geçen ilk satran şampiyonunu yendi. Bu erken alışmalar, bugün her bilgisayarda bulunan akıllı arama ve sesli komut özellikleri gibi birçok uygulamanın öncüsü oldu.

Teknoloji çağında veri hacminin artması, analiz süreçlerinde yapay zeka teknolojilerinin kullanımını zorunlu hale getirmiştir. Yapay zeka, normal bir insanın sınırlı zaman içinde tamamlayamayacağı işleri daha kısa sürede ve daha doğru sonuçlarla tamamladığı birçok uygulamada, performans, verimlilik ve üretkenlik açısından artış sağlamıştır.

Yapay zeka teknolojisi, tarım, sağlık ve finans gibi birçok endüstride kullanılabilecek özümmler sunarak uygulama alanını genişletmiştir. Özellikle sağlık sektöründe, ilaç dozajı ayarlaması ve cerrahi prosedürlerin test edilmesi gibi eşitli alanlarda yapay zekâ özümleri kullanılmaktadır (Pacal 2022)

Otomotiv endüstrisinde, otonom araçlarda kullanılan eşitli sensör verilerinin yorumlanması için yapay zeka özümleri kullanılırken, bankacılık sektörü de yapay zeka teknolojisinden yararlanarak potansiyel dolandırıcılık işlemlerinin tespiti için olağan dışı hesap hareketlerini izlemektedir (McCarthy vd 2006).

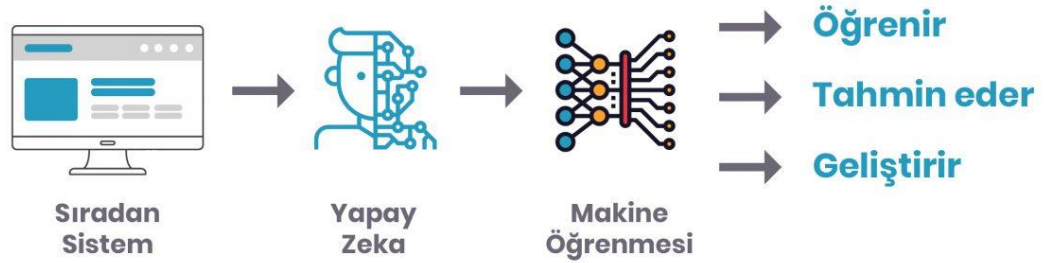
Yapay zeka, farklı alanlarda özelleştirilmiş yöntemler kullanarak görüntü işleme, doğal dil işleme ve optimizasyon gibi işlemleri gerçekleştirmektedir (McCarthy 1980; McCarthy vd 2006). Şekil 2'de yapay zekanın temel alt dalları ve bu dalların örnek kullanımları yer almaktadır



Şekil 2. Yapay zeka alt alanları (Villanueva ve Salenga 2018)

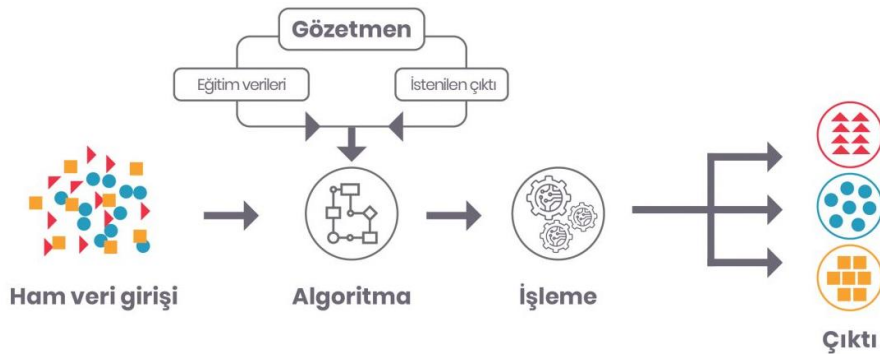
Makine öğrenmesi

Makine öğrenmesi (Machine Learning), Şekil 3’te görüldüğü gibi bir bilgisayar programının, verilerden doğrudan öğrenerek performansını ölçülebilir şekilde artırmasıdır. Bu öğrenme, insanların verileri anlamlandırarak yaptığı gibi doğrudan programlama yapmadan gerçekleştirilir. Makine öğrenimi üç ana kategoriye ayrılabilir.



Şekil 3. Makine öğrenimi ana kategorileri (turhost, 2021)

Denetimli Öğrenme (Supervised Learning)

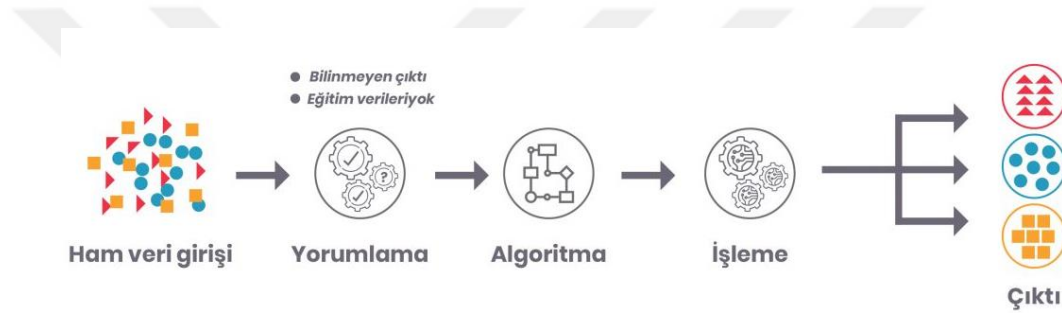


Şekil 4. Denetimli öğrenme süreci (turhost, 2021)

Denetimli öğrenme, belirli bir çıktıya karşılık gelen doğru bir girdi-çıkı ilişkisi öğrenmeye çalışır. Bu tür öğrenme algoritmaları, önceden etiketlenmiş verileri kullanır ve bu verilerden yararlanarak Şekil 4'te görüldüğü gibi yeni verileri sınıflandırır veya tahmin eder. Örnek olarak, bir resim tanıma uygulaması için, belirli bir nesneyi tanımlamak için kullanılan önceden etiketlenmiş resimler kullanılabilir (Hastie vd 2009).

Denetimsiz Öğrenme (Unsupervised Learning)

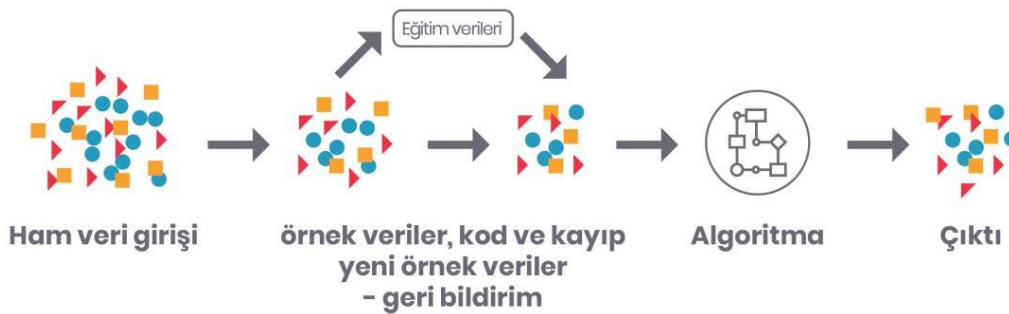
Denetimsiz öğrenme, verilerdeki doğal yapıları keşfetmeye ve öğrenmeye çalışır. Bu tür öğrenme algoritmaları, Şekil 5'te görüldüğü gibi etiketlenmemiş verileri kullanarak benzer özelliklere sahip verileri bir araya getirir veya verileri gruplar. Örnek olarak, bir müşteri segmentasyon uygulaması için, etiketlenmemiş müşteri verileri kullanarak benzer özelliklere sahip müşterileri bir araya getirebilir (Dike vd 2018).



Şekil 5. Denetimsiz öğrenme süreci (turhost, 2021)

Yarı denetimli öğrenme

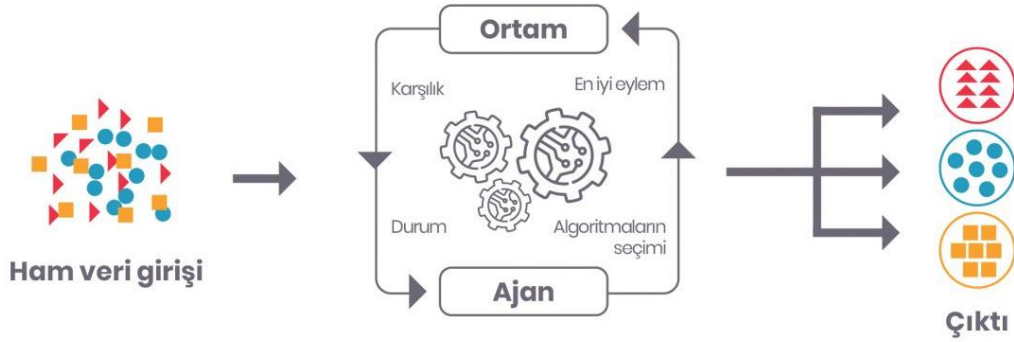
Denetimli ve denetimsiz öğrenme tekniklerini birleştiren yarı denetimli öğrenme türü, bir kısmı etiketlenmiş (yani çıktıları belirlenmiş) ve diğer kısmı etiketlenmemiş (yani çıktıları belirlenmemiş) veri kümesini kullanarak tahmin yapar. Model, etiketlenmiş verileri kullanarak bir ilişki öğrenir ve ardından etiketlenmemiş veriler üzerinde Şekil 6'da görüldüğü gibi tahmin yapmak için bu ilişkiyi kullanır (X. J. Zhu 2005).



Şekil 6. Yarı denetimli öğrenme süreci (turhost, 2021)

Pekiştirmeli Öğrenme (Reinforcement Learning)

Pekiştirmeli öğrenme, bir sistem, doğru olmayan bir sonuca ulaştığında cezalandırıldığı veya doğru bir sonuca ulaştığında ödüllendirildiği bir ortamda bir görevi öğrenmeye çalışır. Bu tür öğrenme algoritmaları, bir karar ağacı oluşturarak, Şekil 7’de görüldüğü gibi belirli bir durumda hangi eylemin gerçekleştirilmesi gerektiğini belirleyebilir. Örnek olarak, bir oyun oynama uygulaması için, doğru hareketler ödüllendirilirken yanlış hareketler cezalandırılabilir (Sutton ve Barto 2018).



Şekil 7. Pekiştirmeli öğrenme süreci (turhost, 2021)

Makine öğrenimi algoritmaları, verileri işlemek için çeşitli teknikler kullanır. Bu teknikler arasında karar ağaçları, doğrusal regresyon, lojistik regresyon, yapay sinir ağları, destek vektör makineleri ve kümeleme algoritmaları yer alır.



Şekil 8. Makine öğrenimi türleri (Kanwal 2020)

Sonuç olarak, makine öğrenimi, verileri kullanarak belirli bir görevi gerçekleştirmeye çalışan bir bilgisayar programının öğrenme sürecidir. Denetli, denetimsiz ve pekiştirmeli olmak üzere üç ana kategorisi vardır ve verileri işlemek için farklı teknikler kullanır. Makine öğrenimi, birçok farklı uygulama alanında kullanılabilir, örneğin:

1. Görüntü İşleme: Makine öğrenimi, görüntü işleme alanında sıklıkla kullanılır. Görüntüleri analiz eden modeller, nesneleri tanımlama, yüz tanıma ve karakter tanıma gibi birçok farklı işlem yapabilir (Karaman vd 2023).
2. Doğal Dil İşleme: Doğal dil işleme (NLP) alanında da yaygın olarak kullanılan makine öğrenimi modelleme teknikleri, metinleri analiz etmek ve anlamak için kullanılır. Örnek uygulamalar arasında konuşma tanıma, metin sınıflandırma ve makine çevirisi yer alır (Chowdhary ve Chowdhary 2020).
3. Tıp ve Sağlık: Makine öğrenimi, tıp ve sağlık alanlarında da kullanılmaktadır. Örneğin, kanser teşhisi, hastalık tahmini ve tedavi seçimleri gibi birçok farklı uygulama bu alanda yer almaktadır (Pacal vd 2020; Yağanoğlu 2022).
4. Finans: Finans alanında, makine öğrenimi, yatırım önerileri yapmak, dolandırıcılık tespiti ve kredi riski değerlendirmesi gibi birçok farklı işlemde kullanılmaktadır (Dixon vd 2020).
5. Robotik: Robotik alanında da yaygın olarak kullanılan makine öğrenimi, otonom araçlar, insansız hava araçları (İHA'lar) ve endüstriyel robotlar gibi alanlarda oldukça popülerdir (Siau ve Wang 2018; Öztürk vd 2022).

Makine öğrenimi, günümüzde pek çok farklı endüstride ve uygulama alanında kullanılmaktadır ve gelecekte de bu kullanım alanlarının genişlemesi beklenmektedir. Makine öğrenimi algoritmaları, veri analizi ve işleme süreçlerinde önemli bir rol oynamakta ve verilerin daha hızlı ve doğru bir şekilde analiz edilmesine yardımcı olmaktadır (Mitchell 2007).

Derin Öğrenme

Derin öğrenme nedir?

Derin öğrenme (Deep Learning), yapay sinir ağları (neural networks) gibi makine öğrenimi modellerinin daha büyük ve karmaşık veri kümelerini işlemek için tasarlanmış bir alt kategorisidir. Derin öğrenme modelleri, yüksek seviyede özelliklerin öğrenilmesi, görevlerin otomatik olarak tanınması ve çıkarılması için tasarlanmıştır. Bu teknikler, özellikle görüntü, ses ve doğal dil işleme gibi büyük veri kümeleriyle çalışırken, geleneksel makine öğrenimi yöntemlerinden daha başarılı sonuçlar elde edebilirler (Karaman vd 2022).

Derin öğrenme yaklaşımları, yapay zeka (AI) ve makine öğrenimi (ML) alanlarında son zamanlarda hızlı bir gelişme kaydetmiştir. Derin öğrenme teknikleri, özellikle büyük veri kümeleriyle çalışırken, daha geleneksel makine öğrenimi yöntemlerine göre daha yüksek başarı oranlarına sahiptirler. Bu teknikler, özellikle görüntü işleme, ses işleme ve doğal dil işleme gibi uygulamalarda önemli bir rol oynamaktadır. Derin öğrenme algoritmaları, örneğin yüz tanıma ve konuşma tanıma gibi uygulamalarda daha yüksek doğruluk oranları elde etmek için kullanılır (Yağanoğlu 2021).



Şekil 9. Derin öğrenme ve makine öğrenimi türlerinin gösterimi (turhost, 2021)

Derin öğrenmenin tarihçesi

Derin öğrenmenin tarihçesi, yapay zeka ve makine öğrenimi gibi konuların tarihçesiyle de yakından bağlantılıdır. Derin öğrenme, yapay sinir ağlarından yola çıkmaktadır ve bu nedenle de yapay sinir ağlarının tarihçesi, derin öğrenmenin tarihçesine önemli bir zemin hazırlamıştır.

1943 yılında, Warren McCulloch ve Walter Pitts, insan beynindeki sinir hücrelerinin basit bir modelini geliştirdiler ve bu modeli "Yapay Sinir Ağı" olarak adlandırdılar (McCulloch ve Pitts 1943). Bu model, insan beynindeki sinir hücrelerinin çalışma biçimini taklit ettiği için oldukça ilgi çekiciydi ve yapay zeka araştırmacıları arasında popüler hale geldi.

1950'lerde, yapay sinir ağlarına olan ilgi, Alan Turing'in "Bilgisayar Makineleri ve Zeka" adlı makalesi ile arttı. Turing, makalesinde, bir yapay zeka sistemine sahip bir bilgisayarın nasıl çalışabileceği konusunu ele aldı ve bu sistemlerin, insan beynindeki sinir ağlarına benzer şekilde çalışması gerektiğini öne sürdü (Alan M Turing 1950).

1960'larda, yapay sinir ağlarının daha da geliştirilmesi için birçok araştırma yapıldı. Ancak, bu araştırmaların çoğu, verimli sonuçlar vermedi ve yapay sinir ağları, yapay zeka ve makine öğrenimi araştırmaları için bir kenara bırakıldı.

1980'lerde, yapay sinir ağlarına olan ilgi yeniden arttı. Bu dönemde, yeni algoritmalar geliştirildi ve sinir ağları, nesne tanıma, konuşma tanıma ve doğal dil işleme gibi uygulamalarda kullanılmaya başlandı.

2000'lerde, yapay sinir ağları, Derin Öğrenme adı altında daha da geliştirildi. Bu dönemde, daha büyük ve daha karmaşık sinir ağı modelleri geliştirildi ve bu modeller, daha yüksek doğruluk oranlarına sahip oldu. Derin öğrenme teknikleri, otonom sürüş, robotik ve yapay zeka gibi birçok alanda kullanılmaya başlandı.

Günümüzde, derin öğrenme teknikleri, yapay zeka ve makine öğrenimi alanlarında en popüler ve etkili tekniklerden biridir (Bozkurt ve Yağanoğlu 2021). Yapay sinir ağları ve derin öğrenme, büyük veri kümelerini işlemek, nesne tanımak, ses ve görüntü işlemek, doğal dil işlemek ve daha birçok alanda kullanılmaktadır (Pichler ve Hartig 2022).

Yapay sinir ağları

Yapay sinir ağları (YSA), insan beyninin çalışma şeklini taklit etmek için tasarlanmış bir bilgisayar algoritması türüdür (Jain vd 1996). YSA'lar, çeşitli katmanlardan oluşan bir ağ yapısına sahiptir. Bu katmanlar, girdi ve çıktı katmanı ile gizli katmanlardır. Girdi katmanı, YSA'ya verilen verileri alırken, çıktı katmanı sonuçları sağlar. Gizli katmanlar, girdi ve çıktı katmanları arasında yer alır ve YSA'nın işlevselliğini belirleyen ağırlıkları içerir (Pacal vd 2022).

YSA'lar, verileri işlemek ve öğrenmek için kullanılır. Öğrenme süreci, YSA'ya verilen verilerin, doğru çıktıları üretmesi için ağırlıkların ayarlanması ile gerçekleştirilir. Bu işlem, geriye doğru yayılım algoritması olarak adlandırılır (Cilimkovic 2015). YSA'lar, çeşitli uygulamalarda kullanılabilirler. Örneğin, resim tanıma, konuşma tanıma ve doğal dil işleme gibi alanlarda kullanılabilirler. Genel olarak, YSA'lar, karmaşık problemleri çözmek için kullanılabilen güçlü bir makine öğrenimi aracıdır.

İleri beslemeli sinir ağları (FNN)

İleri Beslemeli Sinir Ağları (Feedforward Neural Networks - FNN), girdilerin yalnızca bir yönde (ileriye) geçtiği ve çıktıların hesaplandığı bir yapay sinir ağı tipidir. Bu ağlar, sınıflandırma, tahmin ve örüntü tanıma gibi çeşitli uygulamalarda yaygın kullanılır. FNN'ler, bir girdi ve bir çıktı katmanı ile bir veya daha fazla gizli katman içerir. Her katman bir dizi

nöron içerir ve aktivasyon fonksiyonları ve ağırlıklar kullanılarak hesaplamalar yapılır. FNN'lerin öğrenme süreci, geriye doğru yayılım algoritması kullanılarak gerçek çıktılarla karşılaştırılarak ağırlıkların ayarlanmasıyla gerçekleştirilir. FNN'ler, el yazısı tanıma, yüz tanıma, spam filtreleme, hisse senedi fiyat tahmini, arabaların otomatik sürüşü, tıbbi teşhis ve daha birçok alanda kullanılmaktadır. Bu nedenle, FNN'ler, verileri işlemek ve öğrenmek için güçlü bir makine öğrenimi aracıdır (Hatcher ve Yu 2018).

Tekrarlayan sinir ağları (RNN)

Tekrarlayan Sinir Ağları (RNN'ler), geri besleme yoluyla girdileri döngüsel olarak işleyen yapay sinir ağı türüdür. Bu geri besleme sayesinde RNN'ler, daha önce hesaplanan çıktıları gelecekteki adımlarda girdi olarak kullanarak, doğal dil işleme, dil modelleme ve zaman serisi analizinde özellikle faydalı hale gelir. RNN'ler, gizli katmanlardan, bir veya daha fazla hücreden ve her adımda çıktıyı hesaplamak için kullanılan bir bellek durumundan oluşur. RNN'lerin öğrenme süreci, geriye doğru yayılım kullanarak ağırlıkların ayarlanmasını içerir ve uzun zaman serileriyle çalışırken karşılaşılan sorunları ele almak için birkaç algoritma geliştirilmiştir. RNN'ler, konuşma tanıma, metin oluşturma ve müzik besteleme gibi çeşitli uygulamalarda kullanılmıştır. Farklı boyutlardaki girdilerle çalışabilir ve aralarındaki ilişkileri öğrenebilirler, ancak ağıın döngüsel yapısı nedeniyle eğitim zor olabilir. RNN'lerin LSTM ve Geçitli Tekrarlayan Birim (GRU) gibi alt tipleri, uzun zaman serilerinde daha iyi performans gösterir.

Uzun kısa süreli hafıza ağları (LSTM)

Uzun Kısa Süreli Hafıza Ağları (Long Short-Term Memory-LSTM), uzun süreli bağımlılıkları modellemede etkili olan bir RNN türüdür. Bilgi akışını kontrol etmek için hafıza hücreleri ve öğrenilebilir bileşenler olan kapıları kullanır (Graves 2013). LSTM, doğal dil işleme, konuşma tanıma ve müzik tanıma gibi çeşitli uygulamalarda kullanılabilir (Hochreiter ve Schmidhuber 1997). LSTM, daha yüksek performanslı ağlar oluşturmak için başka bir RNN türü olan GRU ile birleştirilebilir. Genel olarak, LSTM, uzun süreli bağımlılıkları modellemek için hafıza hücreleri ve kapı mekanizmalarını kullanan güçlü bir RNN türüdür.

Sınırlı boltzmann makineleri (RBM)

Sınırlı Boltzmann Makineleri (Restricted Boltzmann Machines-RBM), verilerin özniteliklerini çıkarmak için kullanılan bir yapay sinir ağı türüdür. RBM'ler, gizli ve görünür katmanlardan oluşan, çift yönlü bağlantılara sahip bir enerji modeli kullanır (Salakhutdinov vd 2007). ve verilen girdinin olasılık dağılımını hesaplar (Hinton 2012). RBM'ler, özellik çıkarımı, örüntü tanıma, gürültü azaltma ve boyut azaltma gibi çeşitli görevler için kullanılır. RBM'ler,

verilen girdinin olasılık dağılımını kullanarak yeni örnekler üretebilen bir çıkarım algoritmasına sahiptir. RBM'ler, özbellekli kodlayıcılar gibi derin öğrenme algoritmalarının temel yapı taşlarından biri olarak kabul edilir (Fischer ve Igel 2012).

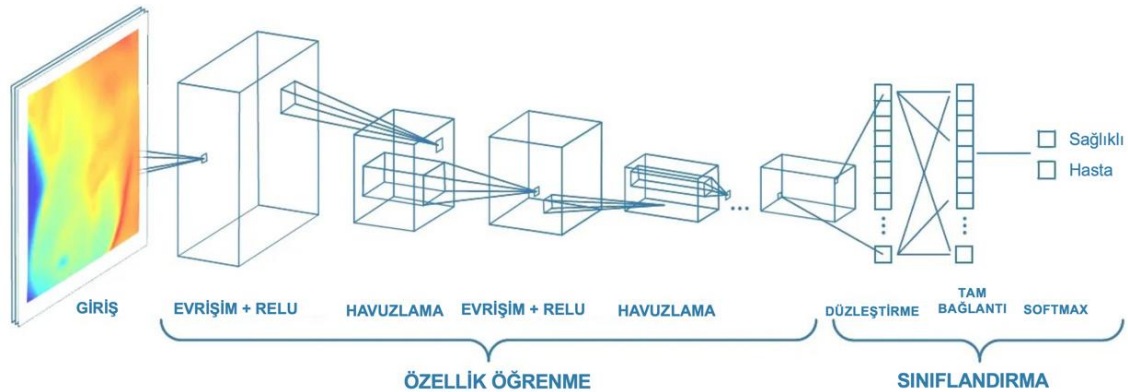
Derin inanç ağları (DBN)

Derin İnanç Ağları (Deep Belief Networks-DBN) yapay sinir ağları ailesinin bir alt dalıdır. RBM'lerin bir araya getirilmesi ile oluşturulan bir modeldir (Hinton vd 2006). DBN'ler, gizli yapıları öğrenmek için denetimsiz öğrenme kullanılarak eğitilirler ve özellikle veri boyutunu azaltma, öznitelik çıkarımı ve kalıp tanıma gibi görevler için etkilidirler (H. Lee vd 2009).. Sınıflandırma, regresyon, görüntü ve doğal dil işleme gibi birçok alanda yüksek performans göstermişlerdir (L. Deng vd 2013).

Oto-kodlayıcılar (AEs)

Oto-kodlayıcılar (AEs), öznitelik çıkarımı ve boyut indirgeme için kullanılan yapay sinir ağı alt kümeleridir (Vincent Pascalvincent vd 2010). AE'ler, bir veya daha fazla gizli katmanı olan bir girdi katmanı ve bir çıktı katmanından oluşur. Girdi verisi sıkıştırılmış bir temsillemeye dönüştürülür ve çıktı katmanında yeniden oluşturulur (Bengio 2009). Geriye doğru yayılım kullanarak eğitilirler ve gürültü giderici otokodlayıcılar (DAE'ler) gibi daha karmaşık versiyonları vardır. DAE'ler, gürültülü girdiler kullanarak eğitilir ve denetimsiz öğrenme yöntemleriyle yetersiz veya gürültülü verilerle çalışırken daha etkilidirler. İşlem görüntüleme, konuşma işleme, doğal dil işleme ve sınıflandırma gibi alanlarda kullanılırlar (Vincent Pascalvincent vd 2010).

Evrişimli sinir ağları (CNNs)



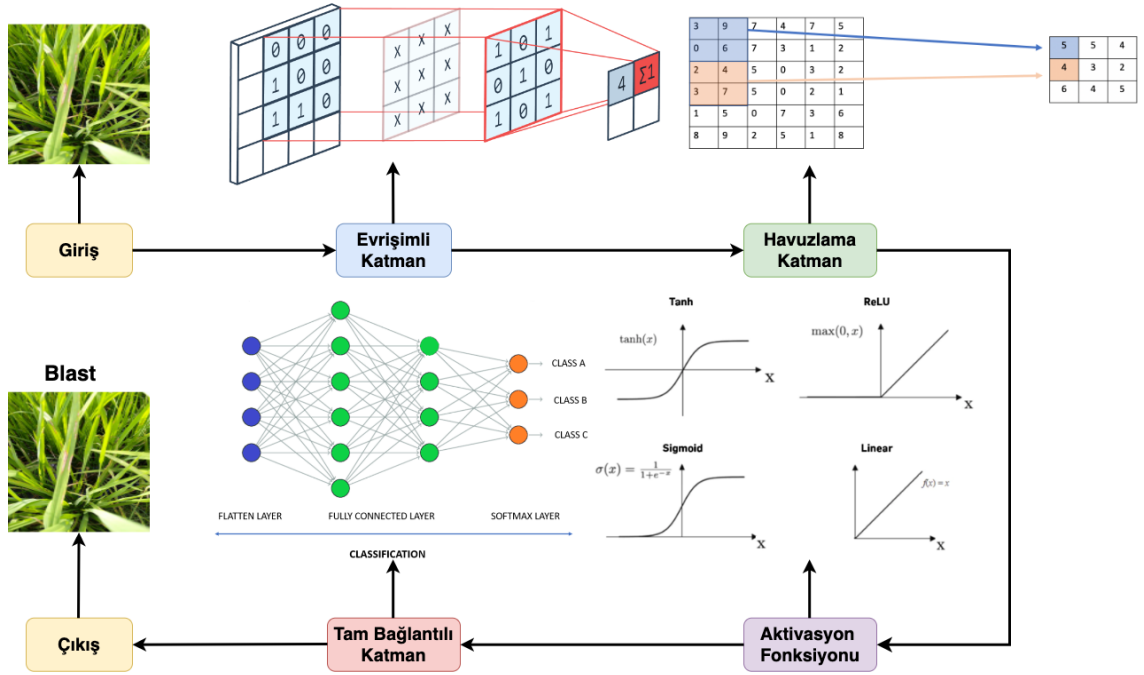
Şekil 10. CNN mimarisi

Evrişimli sinir ağları (CNN'ler), çok katmanlı sinir ağlarını kullanarak görüntülerdeki nesneleri tanımlamak, tanımak ve sınıflandırmak, ayrıca farklı nesneleri algılamak ve ayırmak için kullanılan yapay zeka sistemlerinden biridir. CNN'ler, manuel yönlendirmeye ihtiyaç

duymadan doğrudan giriş nesnelerinden öğrenilebilen popüler derin öğrenme mimarileridir ve CNN'lerin bu özelliği çeşitli çalışmalarla gösterilmiştir (Koushik 2016; Bezdan ve Džakula 2019).

CNN'ler, çeşitli 2D şekilleri ele almak için özel olarak tasarlandığından, genellikle görsel tanıma, tıbbi görüntü yorumlama, görüntü bölümlenme, doğal dil işleme ve diğer alanlarda kullanılmaktadır. Girdiden önemli özellikleri insan müdahalesine gerek duymadan tanıyabilmesi nedeniyle düzenli bir ağın verimliliğini aşmaktadır.

CNN mimarisiyle ilgili gelişmeleri anlamak için farklı CNN bileşenlerinin uygulamalarını anlamak çok önemlidir. Şekil 11, çeşitli CNN parçalarını göstermektedir. Temel bir CNN, dört çeşit katmandan oluşmaktadır. Bu katmanlar, Evrişimli (convolutional), havuzlama (pooling), aktivasyon fonksiyonu (function of activation) ve tam bağlantı (fully connected) katmanıdır.



Şekil 11. CNN bileşenleri

Giriş görüntüsü

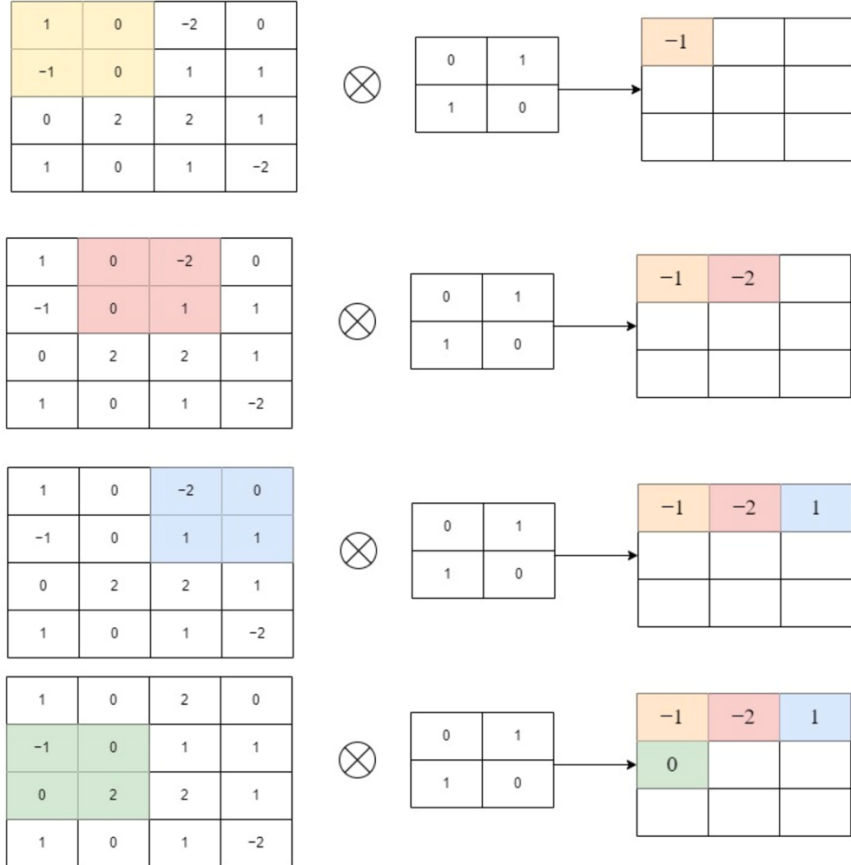
Pikseller, görsel verilerin ikili temsili olarak hizmet eden dijital bir görüntünün temel unsurlarıdır. Görüntü içinde matris benzeri bir düzenleme içinde sıralı olarak 0-255 aralığında yer alırlar. Piksel değeri tarafından her pikseldeki parlaklık ve ton belirlenir (Koushik 2016; Bezdan ve Džakula 2019). Bir görüntüyü görüntülemek için insan beyni büyük miktarda veriyi işler. Aynı şekilde, CNN katmanları önce çizgi ve eğri gibi daha temel desenleri belirlemek için tasarlanmıştır. Daha sonra yüzler ve nesneler gibi daha karmaşık desenlere belirlemeye

geçilmiştir. Bu nedenle, bir CNN kullanmanın bilgisayarlara görme yeteneği kazandırabileceği savunulabilir (Koushik 2016; Bezdan ve Džakula 2019).

Evrişimli katman

Evrişimli katman, CNN'in genel yapısında önemli bir rol oynar. Bu katman, giriş verilerine uygulanan bir kısım filtre veya çekirdek içerir. Her bir çekirdek, yükseklik, genişlik ve ağırlık özellikleri çıkarmak için kullanılır. Çekirdek ağırlıkları ilk olarak rastgele atanır, ancak ağırlıklar eğitim verileri tarafından daha fazla bilgilendirilerek ayarlandıkça anlamlı özellikler çıkarmayı öğrenir. Başka bir ifadeyle, özellik haritası, giriş görüntüsünün N-boyutlu metrikleriyle birlikte bir dizi filtrenin birleştirilmesiyle oluşturulur. Bu filtreler, kesirli sayılar veya tamsayılarla oluşturulur (Koushik 2016; Bezdan ve Džakula 2019).

Her bir tam sayıya, başlangıçta rastgele olarak atanmış bir ağırlık atanır. Bu ağırlıklar, çeşitli başlangıç yöntemleri kullanılarak eğitim sürecinde ayarlandıkça çekirdek anlamlı özellikler çıkarmayı öğrenir. Çekirdek, CNN'lerin yüksek boyutlu, dolaylı özellik uzayında işlem yapmasını sağlar ve verilerin o uzaydaki koordinatlarını hesaplamak zorunda kalmadan, özellik uzayındaki tüm veri çiftlerinin görüntülerinin iç çarpımını hesaplar. Doğrusal bir modeli doğrusal olmayan bir modele dönüştürebilen bir teknik olarak çekirdek hilesi kullanılabilir.



Şekil 12. Temel hesaplamaların görsel bir temsili

Evriřim iřlemi bařlamadan nce, CNN giriř formatı belirtilir. Klasik sinir aęları, verileri vektr formatında alırken, CNN'ler oklu kanallı grntleri alır. RGB grntleri  farklı renk kanalından oluřurken, gri tonlamalı grntler sadece tek bir kanala sahiptir.

Evriřim iřlemini anlamak iin, 4x4 boyutlarında gri tonlamalı bir grnt ve 2x2 boyutlarında rastgele bařlatılmıř bir ekirdek kullanabiliriz. Bu ekirdek, grntnn tamamı boyunca yatay ve dikey olarak hareket ettirilir ve bu sırada grnt ile ekirdek arasındaki nokta arpımı eřzamanlı olarak hesaplanır. Bu iřlem iin, grnt ile ekirdek arasındaki karřılık gelen deęerler arpılarak ve elde edilen sonular eklenerek tek bir sayısal deęer elde edilir. Bu iřlem, kaydırma yapılamayacak kadar kk bir boyuta gelinceye kadar tekrar edilir.

Ana resim (K), filtre (L).

ıktı matrisi, denklem temel alınarak hesaplanır:

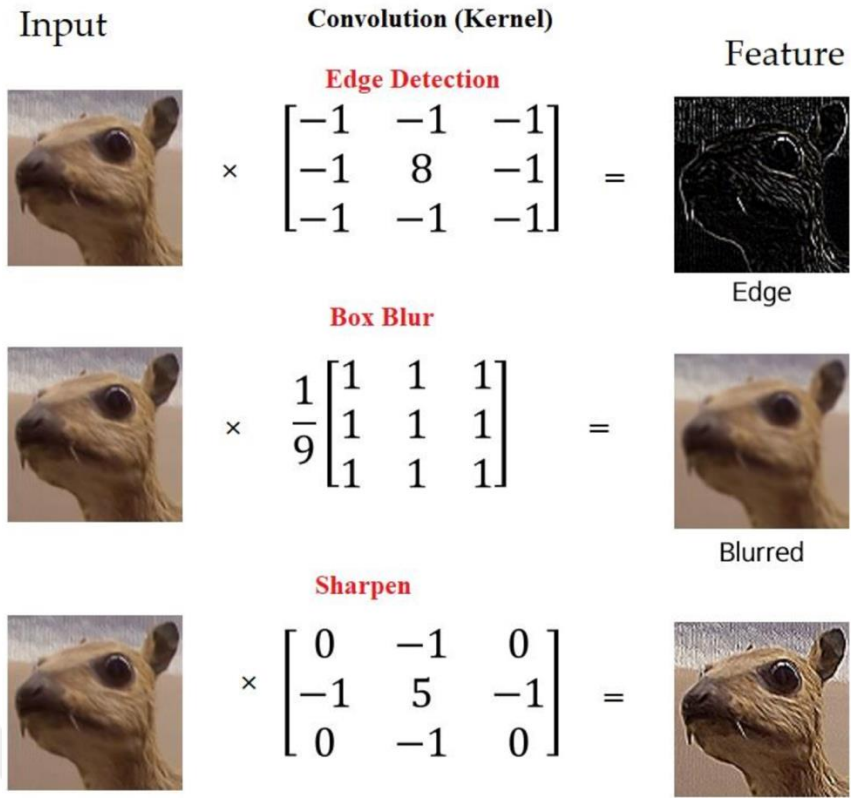
$$(K - L + 1) \tag{1}$$

4 - 2 + 1 = 3, bu nedenle ıktı matrisi 3 × 3'tr.

Nokta arpımı deęerleri, aslında ıkıřın zellik haritasını gsterir. Her ařamada gerekleřen ana hesaplamaları gsteren řekil 12 ile bu iřlem grselleřtirilmiřtir. Diyagramda, ekirdeęi temsil etmek iin daha kk bir kare (2 x 2) ve girdi resmini temsil etmek iin daha byk bir kare (4 x 4) kullanılır. ekirdek ve girdi resmi arpıldıęında, ortaya ıkan rnler bir araya getirilerek tek bir sayı retilir. Bu toplam, ıkıř zellik haritası iin bir giriř deęeri olarak kullanılır ve iřleme dahil edilir (Bezdan ve Dřakula 2019).

nceki rnekte, ekirdek dikey veya yatay konumlarda adım boyutunu belirten 1 adım deęerine sahipti, ancak girdi grntsne herhangi bir dolgu yapılmamıřtı. Ancak istenirse farklı bir adım deęeri seilebilir. Adım deęerini artırmak, sonuta oluřan zellik haritasının boyutunu da azaltabilir. Dolgu, saęlanan grntnn kenar boyutunu byk lde etkiler. Dięer yandan, kenar zellikleri zamanla nemli lde deęiřir. Dolgu ekleyerek, girdi grnts daha byk hale gelir ve bu da zellik haritasının boyutunu artırır.

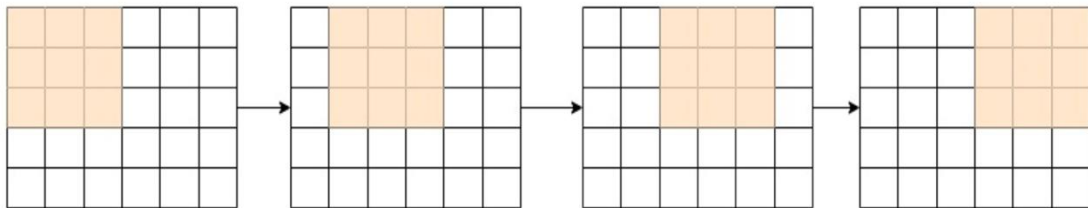
CNN'de, her filtre bir zellięi temsil edebilir, ancak grnt zerinde hareket ettike bir eřleřme tespit ettięinde yalnızca etkinleřir. Bu yntem, CNN'lerin nesne filtrelerini belirlemesine yardımcı olur. řekil iřleme yntemlerinde kullanılan geleneksel filtreler gibi davranan matrislerin, resimlerde kenarları tespit etmek iin nasıl yapılandırılabileceęini řekil 13 gstermektedir. Bu matrislere filtreler denir. Ancak CNN'de, eęitim srecinden nce uygulanan bu filtreler, grev iin daha uygun olan řekil filtrelerinden nce uygulanır.



Şekil 13. Farklı evrişimli matrislerin etkileri (Fang 2017)

Ağırlık Paylaşımı (Weight Sharing): CNN, girdi matrisindeki tüm pikseller için aynı ağırlıkları uygular, bu da bitişik katmanlardaki nöronlar arasında belirli ağırlıklar atanmadığı anlamına gelir. Tüm girdi için tek bir ağırlık seti öğrenme yaklaşımı, her nöron için ek ağırlıkların öğrenilmesi gereksinimini ortadan kaldırdığı için eğitim süresini ve çeşitli maliyetleri önemli ölçüde azaltır.

Adım (Stride): CNN, ayarları daha da iyileştirmek ve istenmeyen etkileri azaltmak için çeşitli seçenekler sunar. Bunlardan biri, bir sonraki katmandaki bitişik düğümler arasındaki örtüşmeyi değiştiren "stride" olarak adlandırılan bir seçenektir. Örneğin, 6 x 6 bir resimde 2'lik bir adım kullanılırsa, en fazla 3 x 3 çıkış boyutuna ulaşılabilir ve bitişik matrisler arasındaki toplam örtüşme azaltılabilir. Bu, Şekil 14'te sol, orta ve sağ matrislerin çıkışının örtüştüğü görülebilir. Ancak 2'lik bir adım kullanarak örtüşmeyi ve toplam çıkış boyutu azaltılabilir.

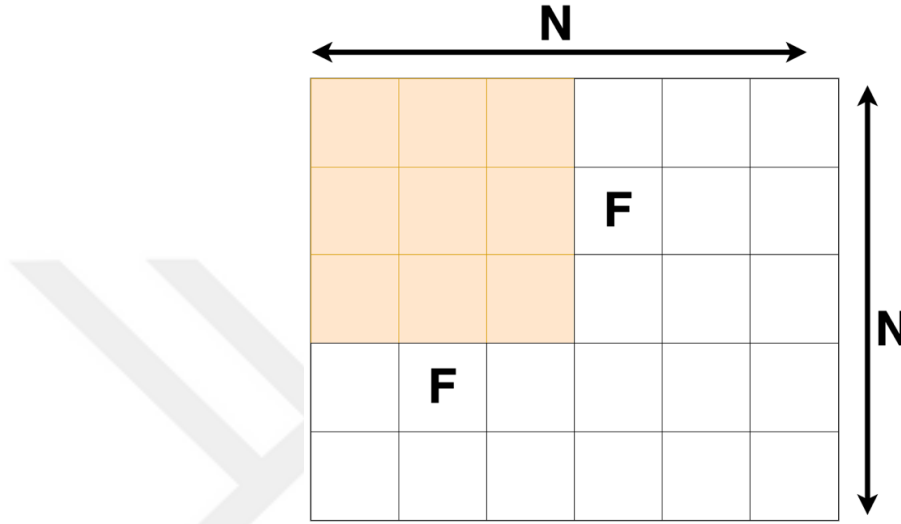


Şekil 14. Adım 1: Filtre pencereleri her bağlantı için yalnızca bir kez hareket eder.

İkinci denklem (2), bu fikri nesnel bir şekilde ifade eder. Şekil 15'te çıktı boyutu O'yu gösterir. Denklem, N x N boyutlarındaki girdi resminin ve F x F boyutundaki filtrenin boyutunu dikkate alır.

$$O = 1 + (N - F) / S \quad (2)$$

Değişkenler N, F ve S sırasıyla girdinin boyutunu, filtre boyutunu ve adım boyutunu temsil eder.



Şekil 15. Çıktı üzerinde adım büyüklüğünün etkisi

Dolgu (Padding): Filtreleme adımının dezavantajlarından biri, filtre hareket ettiğinde yalnızca yakalanan ancak tam olarak görülmeyen görüntünün kenarlarında detay kaybına neden olabilmesidir. Bu duruma basit bir çözüm, çıktı boyutunu yönetmek için kullanılabilecek sıfır dolgu kullanmaktır.

Örneğin, Şekil 15'te N = 6, F = 3 ve adım = 1 ise, çıktı boyutu 6x6 giriş boyutundan daha küçük olan 4x4 olacaktır.

Bununla birlikte, bir sıfır dolgusu dahil edildiğinde sonuç 6×6 olacaktır, bu da ilk girdiyle aynıdır (gerçek N artık 9'dur). Verilen formül, değiştirilmiş sıfır dolgulu (3) bir formüldür.

$$O = 1 + \frac{N + 2P - F}{S} \quad (3)$$

P, sıfır doldurmanın katmanlarının sayısıdır (3). Şekil 16'da gösterildiği gibi derin evrişimli ağların sayısını istediğimiz kadar artırabilmemizi sağlayan bir teknik olan sıfır dolgu katmanları ekleyerek ağ çıkış boyutunun derinlikle azalması önlenabilir (Fang 2017).

0	0	0	0	0	0	0	0
0							0
0							0
0							0
0							0
0							0
0							0
0	0	0	0	0	0	0	0

Şekil 16. Sıfır dolgu (Zero-padding)

CNN'lerin özelliği: Yukarıda bahsedilen ağırlık dağılımı, modelin çevirilere karşı duyarsız olmasına neden olur. Öğrenme işlevi, herhangi bir ortamda performansı iyileştirmek için filtrelenerek kullanılabilir. Filtreler başlangıçta rastgele değerlere ayarlanırsa, kenarları tespit etmeyi öğreneceklerdir (Şekil 10'da gösterildiği gibi). Girdinin uzamsal önemini değerlendirmek için paylaşılan ağırlıkların kullanılması önerilmez.

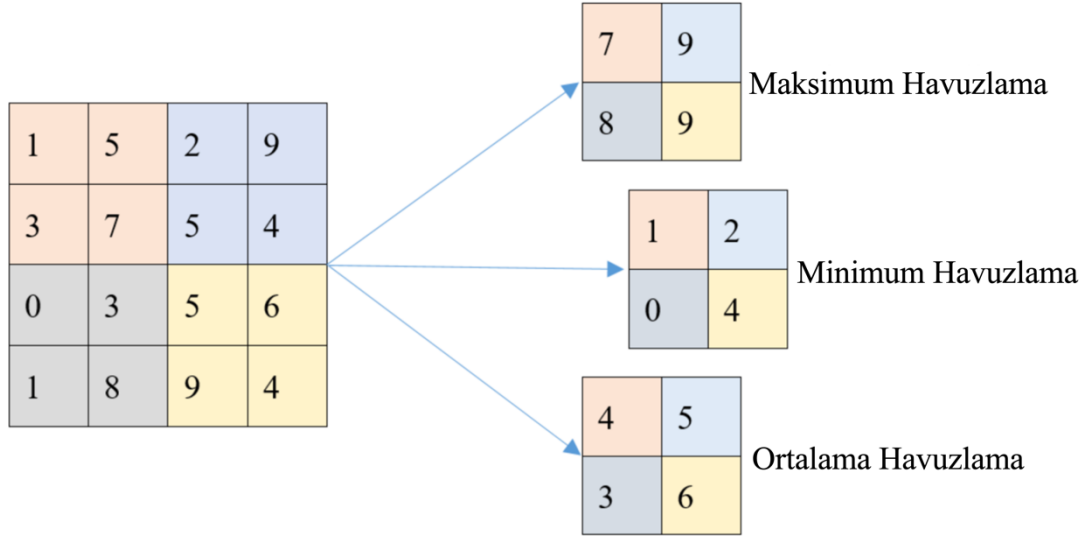
Havuzlama katmanı

Aşağı örnekleme katmanı olarak da adlandırılan Havuzlama katmanı, özellik haritalarının boyutunu azaltırken önemli verileri korumak için kullanılan bir tekniktir. Havuzlama katmanında, giriş verileri üzerinde kayan bir filtre kullanılır ve havuzlama işlemi uygulanır. Bunlar maksimum, minimum veya ortalama havuzlama olabilir. Literatüre göre en sık olarak maksimum havuzlama kullanılır.

Örnekleme, havuzlamanın üst katmanlarının karmaşıklığını azaltmaya yardımcı olan temel bir parçadır. Resim işlemede çözünürlüğü azaltmaya benzer. Ancak, filtre sayısı havuzlamadan etkilenmez. Maksimum havuzlama, genellikle 2x2 boyutlarında dikdörtgen alt bölgelere bölerek her bir alt bölgedeki en büyük değeri döndürerek resimlerdeki boyut azaltma işlemi için kullanılan bir yöntemdir.

Şekil 17'de, havuzlama sol üst köşedeki 2x2 bloklara uygulandığında, odak sağ üst köşeye kayar ve iki adım atılır. Bu nedenle, havuzlama için adım 2 kullanılır. Alışılmadık bir

durum olmasına rağmen, örneklemeden kaçınmak için adım 1 kullanılabilir. Fakat, örnekleme verilerin pozisyonunu korumaz.



Şekil 17. Havuzlama katmanı.

Global ortalama havuzlama (GAP), global maksimum havuzlama, ortalama havuzlama, minimum havuzlama ve kapılı havuzlama gibi farklı havuzlama teknikleri, çeşitli havuzlama seviyelerinde uygulanabilir. Şekil 17, bu üç havuzlama tekniğini (Du ve Swamy 2013; Alzubaidi vd 2021) göstermektedir.

Havuzlama katmanının temel sorunu, CNN'nin girdi resminde bir özelliğin var olup olmadığını belirlemesine yardımcı olmaması, ancak yalnızca özelliğin nerede bulunduğu hakkında bilgi sağlamasıdır. Bu sorun, önemli bilgilerin atlanması nedeniyle CNN modelinin genel doğruluğunda azalmaya neden olmaktadır. Ancak, bu gerekli bilgi, CNN modelinden çıkarılmıştır.

Aktivasyon katmanı

Evrişimli katmanından sonra, çıktıyı değiştiren veya durduran bir doğrusal olmayan katman bulunur. Bu katman, çıktıyı modüle etmek veya sınırlamak için kullanılabilir. Sinir ağlarındaki aktivasyon fonksiyonu, girdiyi çıktıya eşleştirir. Bir nöronun belirli bir girdiye yanıt olarak etkinleştirilip etkinleştirilmeyeceğini belirlemek için, nöron girdisi ve çekirdek ağırlığı (varsa) çarpılıp toplanır.

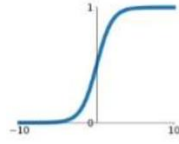
CNN mimarisinde, evrişim ve tam bağlantılı katmanlar gibi ağırlıklı katmanlarla birlikte doğrusal olmayan aktivasyon katmanları da kullanılır. Doğrusal olmayan aktivasyon katmanları, girdi-çıkı eşlemesini doğrusal olmayan hale getirerek, CNN'lerin karmaşık özellikleri öğrenmesine olanak tanır (Du ve Swamy 2013; Q. Zhang vd 2019; Alzubaidi vd

2021). Aktivasyon fonksiyonları, hata geri yayılımı ve ağı eğitimi için farklı olabilme özelliğine sahip olmalıdır.

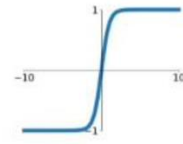
CNN'ler ve derin sinir ağlarında yaygın olarak kullanılan aktivasyon fonksiyonları arasında sigmoid, ReLU ve tanh bulunur. Sigmoid aktivasyon fonksiyonu, çıktı değerlerini 0 ile 1 aralığına sınırlar ve gerçel sayıları girdi olarak kabul eder (Lin,Goyal, vd 2017; S. Zhang vd 2018; Bhatt vd 2021; Prakash vd 2021). Tanh, sigmoid fonksiyonuna benzer ancak çıktı değerlerini -1 ile 1 aralığına sınırlar. CNN'ler için en sık kullanılan aktivasyon fonksiyonu, ReLU'dur. Tüm giriş değerlerini pozitif aralığa dönüştürür ve hesaplama verimliliği açısından avantaj sağlar.

Uzun bir süre boyunca Tanh ve sigmoid popülerdi, ancak ReLU hesaplama avantajları nedeniyle giderek popüler hale geldi. Farklı tipte non-lineer aktivasyon fonksiyonları Şekil 18'de gösterilmektedir.

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



tanh
$$\tanh(x)$$



ReLU
$$\max(0, x)$$



Şekil 18. Aktivasyon fonksiyonu

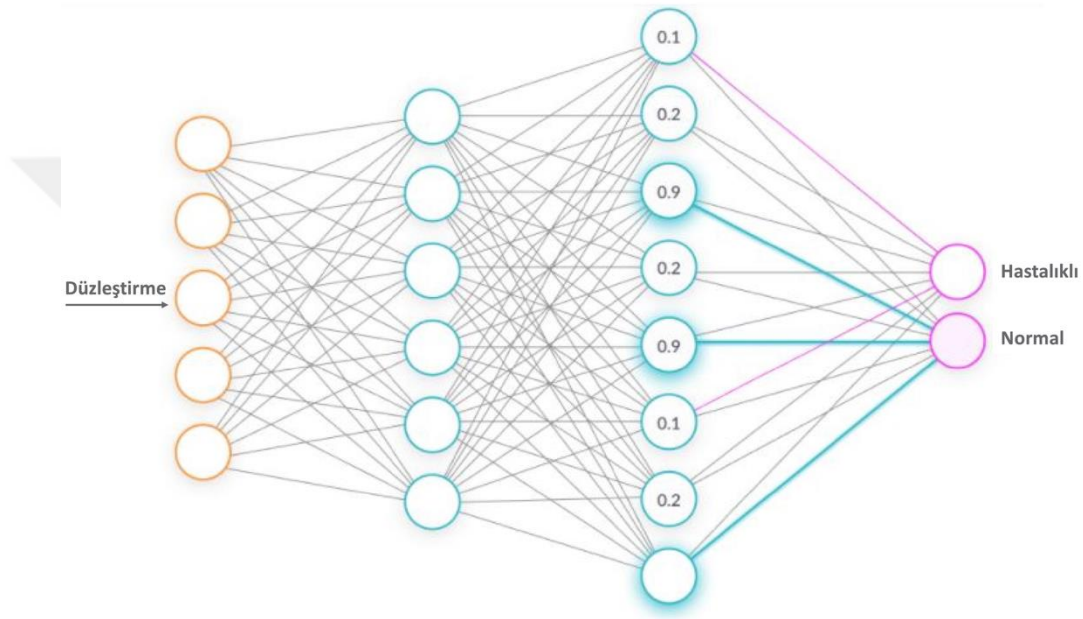
ReLU işlevi ve gradyan tanımlamaları oldukça basit bir yapıya sahiptir. Geriye yayılım, sigmoid ve tanh gibi doygun fonksiyonlarda sorunlu olabilir, çünkü bu fonksiyonların merkezinin her iki yanında gradyan sıfıra yakındır ve sinir ağının derinleştikçe gradyan sinyali azalır. ReLU, pozitif girdiler için sabit bir gradyana sahiptir ve fonksiyonu 0'da ayırt edilmezse uygulama sırasında ihmal edilebilir. ReLU, gradyanı sıfır olan tam bir sıfır oluşturduğundan daha seyrek bir temsil oluşturur. Sigmoid ve tanh hiçbir zaman sıfır gradyan üretmez, bu da eğitim sırasında ters etki yapabilir (Du ve Swamy 2013; G. Huang vd 2017; Lin,Goyal, vd 2017; S. Zhang vd 2018; Q. Zhang vd 2019; Alzubaidi vd 2021; Prakash vd 2021).

ReLU kullanırken bazı önemli sorunlar ortaya çıkabilir. Örneğin, hata geri yayılımı için daha büyük bir gradyan akışı gerektiren bir yöntem düşünelim. Bu gradyanı ağırlık güncellemesi için ReLU fonksiyonundan geçirerek, nöronun tekrar uyarılmamasına neden olabilir, bu da Ölen ReLU "Dying ReLU" sorununa yol açar. Bu sorunları çözmek için birkaç ReLU alternatifi bulunmaktadır. Ölü ReLU sorununu çözmek için alternatif aktivasyon

fonksiyonlarından biri olan Sızdıran (Leaky) ReLU kullanılabilir. Sızdıran ReLU negatif girdileri göz ardı etmek yerine onları küçültmek suretiyle gözardı etmez.

Tam bağlantı katmanı

CNN'deki tam bağlantılı nöron katmanı, her bir düğümün doğrudan üst ve alt katmanlardaki her düğüme bağlandığı geleneksel sinir ağlarına benzer. Bu katman, Şekil 19'da son havuzlama katmanının en son kareleri üst katmana bir vektör olarak bağlandığında görülebilir. Bununla birlikte, bu katmanlar eğitim süresince çok fazla zaman gerektirirler (Du ve Swamy 2013; Q. Zhang vd 2019; Alzubaidi vd 2021).



Şekil 19. Tam bağlantılı katman (Prasad 2021).

Tam bağlantılı katmanın en temel dezavantajı, yoğun hesaplama gerektiren ve eğitim sırasında büyük sayıda parametreye sahip olmasıdır. Bunu çözmek için, bağlantı ve düğüm sayısını azaltmak ve bazılarını Dropout ile ortadan kaldırılmakla mümkündür. LeNet ve AlexNet gibi ağlar, hesaplama verimliliğini korurken büyük ve karmaşık ağlar oluşturmuştur (Du ve Swamy 2013; Alzubaidi vd 2021).

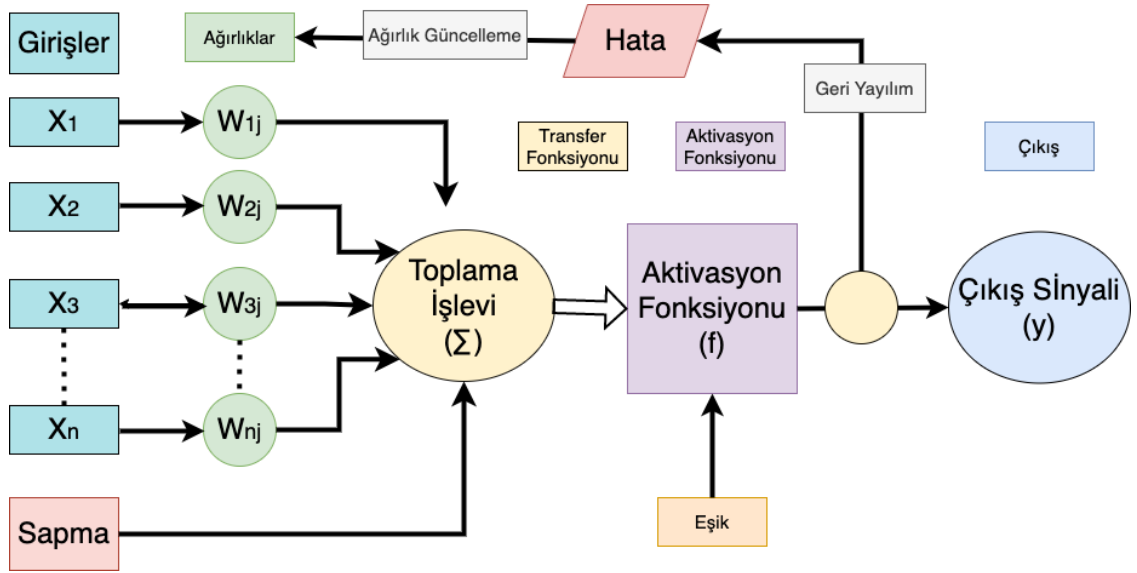
CNN ağının temel unsuru olan evrişimli, doğrusal olmayan katman ile havuzlama katmanlarının eklenmesiyle ortaya çıkar. Şekil 19'de gösterilen nihai CNN çıkışına uygun olarak, Tam Bağlantılı katmanın çıkışı uyumlu hale getirilir. Önceki bölümde, CNN tasarımında kullanılan çeşitli katman türleri tartışılırken, bu bölüm ise kayıp fonksiyonlarına odaklanmaktadır.

Son sınıflandırma, farklı kayıp fonksiyonları kullanarak eğitim verileri üzerinde tahmin edilen hatayı hesaplayan çıkış katmanı kullanılarak elde edilir. Bu işlem, gerçek ve tahmini çıktı arasındaki farkı vurgular ve ardından CNN öğrenme yaklaşımı kullanılarak iyileştirilir.

CNN'lerde kullanılan kayıp fonksiyonu, hatanın kaynağını belirlemek için iki giriş kullanır. İlk giriş tahmini çıktı veya çıktı tahmini olarak adlandırılırken, ikinci giriş gerçek çıktı veya etiket olarak adlandırılır. Farklı problemler için, farklı türde kayıp fonksiyonları kullanılır (Bhatt vd 2021).

Farklı türdeki kayıp fonksiyonlarının basit bir açıklaması aşağıda verilmiştir:

Eğitim (Training): Bir CNN modelinin eğitim sürecinde, sınıflar, sınırlayıcı kutular ve maskeler gibi görüntüler ve etiketler içeren bir eğitim veri kümesi kullanılır. Bir önceki katmanın çıktısını kullanarak bir hata değeri hesaplayan ve o katmandaki her bir nöronun ağırlığını güncelleyerek geriye doğru ilerleyen bir yöntem olarak Geriye Yayılım, sinir ağlarında yaygın olarak kullanılan bir algoritmadır (Prakash vd 2021). Şekil 20'de gösterildiği gibi yanlış değeri hesaplamak ve önceki ağırlıkları ayarlamak için yeni ağırlıklar kullanılır.



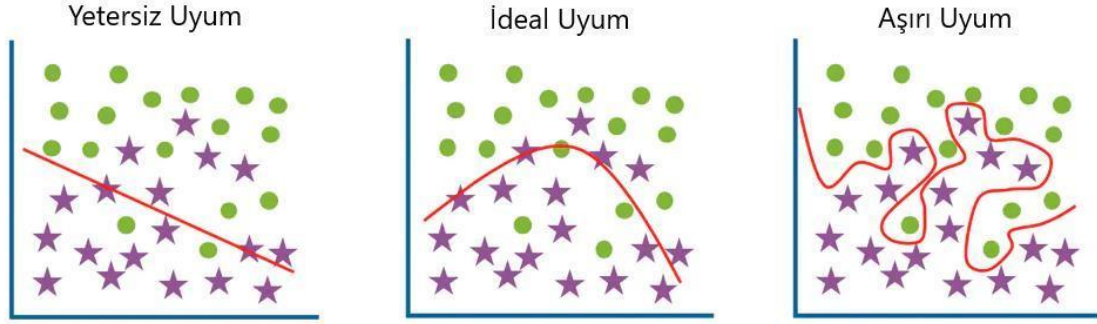
Şekil 20. Gizli CNN katmanlarındaki ileri ve geriye yayılım süreçler

Algoritma, bu işlemi birinci katmana ulaşana kadar tekrarlar. Tüm girdiler, sapma birimi de dahil olmak üzere etkinleştirme birimi tarafından özetlenir ve ardından etkinleştirme fonksiyonu hesaplanmak üzere kullanılır. Bu işlem, sinir ağlarının temel işleyişini oluşturur. Ağ, maliyet fonksiyonunu hesaplar ve ağırlıkları güncelleyerek maliyeti en aza indirmeye çalışır.

CNN yapılarının düzenlenmesi

CNN modelleri için doğru ve güvenilir genellemeler üretmedeki ana zorluk aşırı uydurmadır. Bir modelin eğitim verilerinde iyi performans gösterirken test verilerinde kötü performans göstermesi durumuna aşırı uydurma denir. Bu, modelin eğitim verilerinden fazla öğrenme yapması ve öğrendiği bilgileri genelleme yeteneğinin zayıf olması nedeniyle ortaya çıkar. Diğer bir durumda ise model, eğitim verilerinden yeterince bilgi alamadığında test

verilerinde de kötü performans gösterir. Bu duruma ise az uydurma denir (Lin,Goyal, vd 2017; S. Zhang vd 2018; Prakash vd 2021). Eğitim ve test verilerinde iyi performans gösteren bir model uygun şekilde uydurulmuş olarak kabul edilir. Şekil 21’de yetersiz, ideal ve aşırı uyum durumu görülmektedir.



Şekil 21. Aşırı uydurmaya karşı düzenleme tekniklerinin CNN'e uygulanması

Aşırı uydurma ve az uydurma önlemek için çeşitli teknikler kullanılır. Bu teknikler daha sonra daha detaylı olarak tartışılacaktır. Aşırı uydurmaya önlemek için popüler bir yöntem, modelin farklı özellikler öğrenmesini teşvik etmek için eğitim sırasında rastgele nöronların çıkarılması olan dropout'tur. Çıkarılan nöronlar eğitim sırasında kullanılmaz ancak test sırasında yeniden etkinleştirilir. Dropout ile benzer olan drop-weight eğitimi, nöronları bırakmak yerine nöronlar arasındaki bağlantıları ortadan kaldırır. Büyük miktarda eğitim verisi mevcut olduğunda aşırı uyumun önlenmesi için veri artırma yararlı bir tekniktir. Eğitim veri setinin boyutunu yapay olarak artırmanın farklı yöntemleri vardır.

Yığın normalleştirme (batch normalization), bir modeldeki son aktivasyonların etkili olmasını sağlamak için bir yöntemdir (F. Yang vd 2019; Zhou vd 2019). Her katmandaki çıktılar normalleştirilmesi işlemi, tek birimli Gauss dağılımı tarafından iyi bir şekilde tanımlanabilir. Bu işlem, verilerin normalleştirilmesi için ortalamanın çıkarılması ve standart sapma ile bölünmesini içerir. Normalizasyon genellikle bir ağı her katmanı için ön işleme adımı olarak görülse de başka ağlarla birleştirilebilir. Normalizasyon ayrıca aktivasyon katmanlarındaki içsel kovaryans kaymasını azaltmaya yardımcı olabilir, bu da her katmandaki aktivasyon dağılımındaki değişkenliğe bağlıdır.

Eğitim sırasında ağırlıklar sürekli olarak güncellendiğinde, özellikle eğitim verileri farklı kaynaklardan geldiğinde (örneğin, gece ve gündüz görüntüleri gibi), bu içsel kovaryans kaymasını artırabilir. Bu sorunu çözmek için, bir toplu normalleştirme katmanı CNN tasarımına eklenebilir, bu da eğitim süresini azaltmaya ve yakınsamayı iyileştirmeye yardımcı olabilir (Lin,Dollar, vd 2017; Ghiasi vd 2018; Kong vd 2020).

Toplu normalleştirme kullanmanın birkaç faydası vardır. Bunlar kaybolan gradyan sorununun oluşmasını engellemek, zayıf ağırlık başlatma durumlarını etkili bir şekilde yönetmek ve büyük veri kümeleri için özellikle faydalı olan ağ yakınsama süresini azaltmaktır. Bununla birlikte, çeşitli hiperparametrelere olan ağ bağımlılığını azaltmak zor olmaktadır. Bu zorluklara rağmen, toplu normalizasyon, aşırı uyum önlemede etkili bir yöntem olup, düzenleme için faydalı bir araçtır.

Birkaç CNN mimarisi

CNN, VGG, AlexNet, Xception, Inception ve ResNet gibi bir dizi mimariye sahip derin öğrenme modelidir. Bu mimariler öğrenme yeteneklerine göre farklı uygulama alanlarında kullanılır. CNN, giriş, evrişimli, havuzlama ve tam bağlantı katmanları gibi çeşitli katmanlardan oluşur ve mevcut ağların çoğu CNN geliştirmelerine dayanır. İlk CNN, el yazısı rakamları tanımlamak için LeCun tarafından 1998'de oluşturulan LeNet'ti. Ancak, sınırlı boyutu nedeniyle diğer veri türlerine genelleme yapamadı ve bilgisayar görüşünde ilerlemeyi yavaşlattı.

2012 yılında, Alex Krizhevsky AlexNet'i önerdi ve bilgisayar görüşünde önemli bir ilerleme sağlandı. AlexNet, özellik öğrenmek için beş evrişimli katman ve iki tam bağlı katman içerir ve derin öğrenmenin bilgisayar görüşü uygulamalarında kullanımını sergileyen ilk örnektir (Swapna vd 2020).

VGG-16, daha hassas bir karar verme süreci oluşturan küçük 3×3 kernel boyutlu birçok evrişimli katman kullanarak tasarlandı.

ZFNet (Zeiler ve Fergus 2014), 2013 yılında AlexNet mimarisinde birkaç değişiklik yaptı, bunlar arasında dekonvolüsyon yoluyla ara katman özelliklerinin görselleştirilmesine olanak tanıyan yeni bir görselleştirme tekniği bulunuyordu. Bu, önceden CNN'lerin optimizasyon ve geliştirme süreci için herhangi bir teorik veya metodolojik açıklama olmaksızın "siyah kutu" olarak görüldüğü için önemli bir ilerleme idi.

VGG modeli, Simonyan ve Zisserman tarafından 2014 yılında tanıtıldı (Simonyan ve Zisserman 2014). Bu model, ağ derinliğinin doğruluk üzerindeki etkisini inceledi ve büyük filtreleri küçük 3×3 evrişimli katmanlarıyla değiştirmek için birçok evrişimli katmanı kullandı. VGG modeli basit ve etkili olsa da parametreleri büyük olduğu için ayarlamak zor olabiliyordu. Bununla birlikte, araştırmacılar VGG modelini çeşitli alanlarda başarılı bir şekilde uygulamışlardır.

2014 yılında GoogLeNet (Swapna vd 2020), doğruluk üzerindeki derinliğin etkisini değerlendirmenin yanı sıra ağın genişliğini de dikkate aldı. Son tam bağlantı katmanını kaldırdı

ve boyutsallığı azaltmak ve ağı parametrelerinin aşırı büyüklüğünden kaynaklanan aşırı uydurmayı önlemek için 1x1 evrişimli işlemi kullandı.

2015 yılında, ResNet (Swapna vd 2020), artık bağlantılar ve ağı kullanan ve ağı derinliğini 152 katmana kadar çıkaran bir yöntem olan kalıntı bağlantılarını kullandı. Ağı, verimli bir şekilde ilerlemeyi teşvik etmek için havuzlama katmanlarının sayısını azaltıp, örnekleme tekniklerinin sayısını artırdı. Bu yaklaşım, o zamanki en üstün görüntü tanıma sonuçlarını elde etti ve kalıntı bağlantılarının uygulanabilirliğini gösterdi.

Liu ve diğerleri tarafından 2017 yılında tanıtılan DenseNet (Duan vd 2019; Bochkovski vd 2020; Swapna vd 2020), model doğruluğunu korurken ağı derinliği ve genişliğini artırmak için ResNet ağına benzer bir yaklaşım kullandı. Bu, tüm özellik haritalarının blok içinde birleştirilmesiyle gerçekleştirildi ve bu da parametre sayısının azalmasına ve özelliklerin birden fazla evrişimli katmanı boyunca yeniden kullanılabilirliğinin artmasına yol açtı.

2018 yılında, BoostCNN, en küçük kareler temelli bir amaç fonksiyonuna sahip derin öğrenme mimarisi kullandı ve modeli eğitmek için yükseltme ağırlıklarını ekledi. Bu model, önerilen yükseltme çerçevesi içinde çeşitli ağı mimarileri kullandı ve BoostCNN her tekrarda en etkili ağı mimarisini seçti.

2019 yılında, EfficientNet, modelleri etkili ve verimli bir şekilde ölçeklendirmek için bileşik katsayı yöntemini kullandı. Eşit ölçekleme katsayıları kümesini kullanarak, bileşik ölçekleme tekniği rastgele bir değer seçmek yerine görüntünün genişliğini, derinliğini veya çözünürlüğünü artırır. Çalışmanın yazarları bu ölçekleme tekniğini ve Otomatik Makine Öğrenimi (AutoML) kullanarak, boyutları değişen yedi model geliştirdiler ve bu modeller, en son CNN'lerden daha etkili ve başarılı oldu.

SVM, K-NN ve Rastgele Orman gibi diğer sınıflandırma algoritmalarının aksine, CNN'ler, görüntüdeki nesneleri temsil etmek için en önemli özellikleri öğrenirler, bu da onlara bir avantaj sağlar. Tablo 1, çeşitli popüler CNN mimarilerinin karşılaştırmasını sağlar.

Tablo 1. Farklı Popüler CNN Mimarileri Arasında Bir Karşılaştırma

Mimari Adı	Katmanlar	Ana Katkı	Öne Çıkanlar	Kuvvet	Boşluklar
LeNet LeNet-5 [1998]	Toplam 7 katmandan oluşur (5 Evrişimli 2 Tam Bağlı).	CNN mimarilerinin ilk popüler modelidir.	Küçük ölçekli nesneleri tanımada oldukça hızlı ve etkilidir.	Hesaplama ve parametre sayısını azaltmak için mekânsal korelasyon kullanılmıştır.	Farklı görüntü sınıflarına ölçeklendirme, çok büyük filtreler kullanma ve zayıf özellikler çıkarma gibi sınırlamalar vardır.
AlexNet [2012]	Toplam 8 katmandan oluşur (5 Evrişimli + 3 Tam Bağlı).	LeNet'ten daha fazla derinlik ve genişlik-Relu, bırakma ve örtüşme kullanır Havuzlama-NVIDIA GTX 580 GPU'lar Bırakma ve ReLU'dan yararlanır	AlexNet , daha karmaşık olması, katman başına daha fazla filtreye sahip olması ve yığılmış evrişimli katmanlar kullanması dışında LeNet-5 ile karşılaştırılabilir .	Başlangıç (5x5 ve 11x11) ve son katmanlarda (5x5 ve 11x11) ve ayrıca 3x3 katmanda hem büyük hem de küçük boyutlu filtreleri kullanarak düşük, orta ve yüksek seviyelerde özellik çıkarımı gerçekleştirir. •Karmaşık mimarilerle başa çıkmak için paralel GPU'lar hızlandırıcı olarak kullanılmaktadır.	Başlangıç ve ara katmanlarda hareketsiz nöronlar bulunabilir. • Büyük bir filtre boyutunun kullanılması, öğrenilen özellik haritalarında bozulmaya veya hatalara neden olabilir.
ZfNet [2014]	8	Orta seviyelerin kavramsallaştırılması		Resimli parametreleri ayarlamak için ara katmanların çıktısını görüntüler. •AlexNet'nin ilk iki katmanında filtre boyutu ve adım küçültülmüştür.	Görselleştirme için bilgilerin daha fazla işlenmesi gereklidir.
VGG [2014]	Toplam 16-19 katmandan oluşur (13-16 Evrişimli + 3 Tam Bağlı)	-Homojen bir yapı elde etmek için küçük tane boyutları kullanılır. -Filtre boyutunu küçültür ve derinliği artırarak modelin performansı iyileştirilir.	Modelin doğruluğunu artırmak için, her evrişim katmanı 3x3 boyutlarında küçük filtreler kullanır.	•Basit ve homojen bir topoloji kavramı sunuldu	Hesaplama açısından maliyetli tam bağlantılı katmanların uygulanması

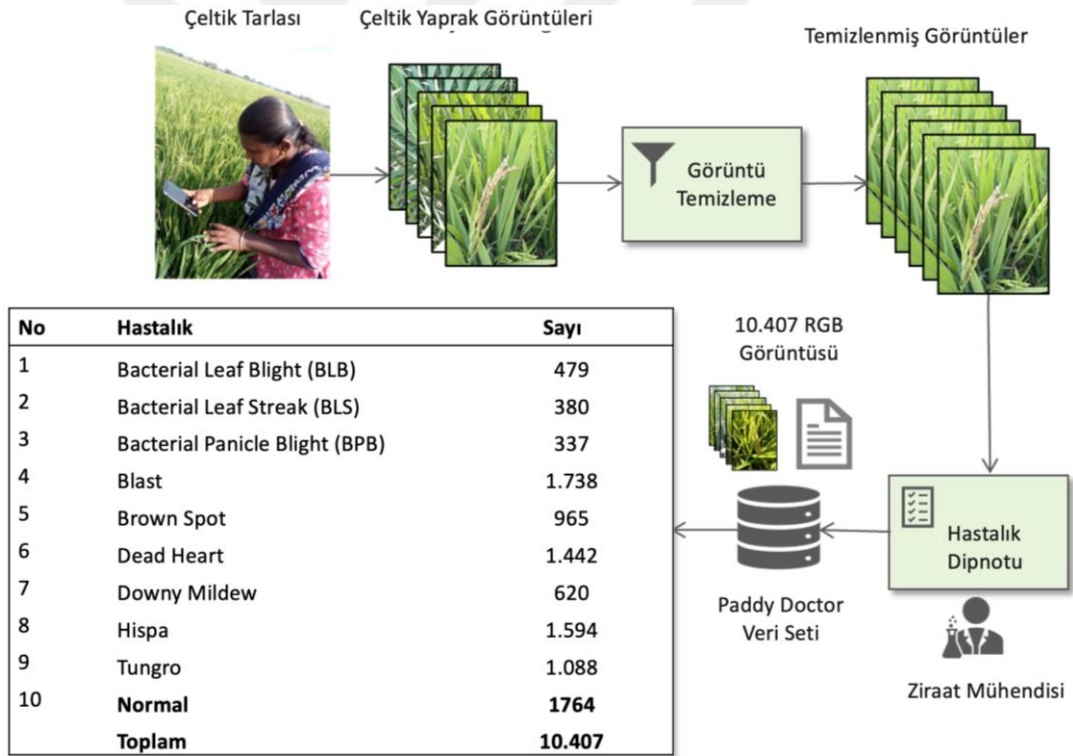
Mimari Adı	Katmanlar	Ana Katkı	Öne Çıkanlar	Kuvvet	Boşluklar
GoogLeNet [2015]	22 evrişim katmanı ve 9 başlangıç modülünden oluşur	Dönüştürme ve birleştirme işlemlerini ayıran blok kavramı tanıtıldı ve derinlik arttırdı, farklı filtre boyutları ve birleştirme kavramı kullanıldı.	Farklı alıcı alan boyutları ve son derece küçük evrişim türleri ile daha derin ve geniş bir mimari.	Katmanlar için çok ölçekli filtrelerin uygulanması ile bölme, dönüştürme ve birleştirme kavramı tanıtılmıştır. •Parametre sayısı, darboğaz katmanı kullanılarak, nihai katmanda global ortalama havuzlama ve seyrek bağlantılarla azaltılmıştır.	Modelin çeşitli topolojileri nedeniyle, parametre değiştirme zor ve zaman alıcıdır. • Tıkanıklık, yararlı bilgi kaybına neden olabilir.
Inception-v3 [2015]	Model, 42 evrişim katmanı ve 10 başlangıç modülünden oluşur.	-Temsili tıkanıklık sorununu çözer -Büyük boyutlu filtreleri küçük boyutlu filtrelerle değiştirir. -Küçük bir filtre boyutu ile gelişmiş özellik gösterimi kullanır.	Büyük boyutlu filtreler yerine daha küçük boyutlu filtreler kullanılarak temsilde tıkanıklık sorunu ele alındı. Model, daha küçük filtreleri kullanarak özellik temsili ve verimliliği artırmaktadır.	•Eğitimi hızlandırmak için Batch Normalization kullanarak verimliliği artırırken, daha büyük derinliğe ulaşmak için Inception-building öğelerini etkili bir şekilde kullanmıştır.	• Derin tasarımların hesaplama masraflarını azaltmak için tıkanıklık katmanları ile asimetrik filtreler kullanılmıştır. • Karmaşık bir mimari yapısına sahiptir. • Modelde bir birlik yoktur.
ResNet [2016]	ResNet-50, 50 katmana sahip, ResNet-101, 101 katmana sahip ve ResNet-152, 152 katmana sahiptir.	- Model, uzun süreli bilgi saklama için bağlantılar aracılığıyla kimlik eşleme kullanır. - Simetrik eşleme tabanlı atlama bağlantıları sayesinde model	Model, bağlantı atlamalara ve kapsamlı genel normalleştirmeye sahiptir.	•Daha derin ağların hata oranını azaltan model, artıklık öğrenme kavramını tanıtarak kayıp gradyan sorununu ele alır.	• Modelin biraz karmaşık bir yapısı vardır ve ileri besleme sırasında özellik haritası bilgisini bozabilir. • Benzer modüllerin bir araya getirilmesi, belirli bir görev için hiperparametrelerin

Mimari Adı	Katmanlar	Ana Katkı	Öne Çıkanlar	Kuvvet	Boşluklar
		aşırı öğrenmeye karşı dirençlidir.			aşırı uyum sağlamasına neden olabilir.
DenseNet DenseNet-121 [2017]	Model toplam 121 katmandan oluşur (117 Evrişimli, 3 Geçiş katmanı ve 1 Sınıflandırma)	Model, birbirine bağlı katman blokları kullanarak katmanlar arasında bilgi iletimine olanak tanır.	Tüm katmanlar birbirine ileri besleme yöntemiyle bağlıdır. Bu durum, kaybolan gradyan sorununu ele almaya yardımcı olur ve gereksinim duyulan parametre sayısını azaltır.	•Çapraz katman boyutu veya derinlik eklenerek, model için maksimum veri akışı sağlanmıştır. •Model, gereksiz özellik haritalarının yeniden öğrenilmesini önler. •Düşük ve yüksek seviye özelliklerine erişmeyi sağlayan karar katmanlarına sahiptir.	Her bir katman için özellik haritası artırılarak parametrelerde önemli bir artışa sebep olur.

MATERYAL VE METOT

Veri Seti

Paddy Doctor (Kiruba ve Arjunan 2023) veri kümesi, 13 sınıfa ait 16.225 etiketli pirinç yaprağı görüntüsünden oluşan, uzmanlar tarafından etiketlenmiş büyük ölçekli bir görsel görüntü veri kümesidir. Bu veri kümesinin bir alt kümesi olan ve eğitim verilerini içeren 10 sınıfa ait 10.407 görüntü, Kaggle (2022) yarışmasında kullanılmak üzere halka açık olarak paylaşılmıştır. Bu veri seti, uzmanlar tarafından etiketlenmiş büyük ölçekli bir görsel görüntü veri kümesidir. Veri kümesi, verim kaybını önlemek ve pestisit kullanımını azaltmak için önemli olan doğru ve erken çeltik hastalıklarının tespiti için bilgisayar görüşü algoritmalarının verimli ve sağlam bir şekilde deney yapılabilmesi için tasarlanmıştır. Şekil 22’te veri toplama ve ek açıklama iş akışı gösterilmektedir.



Şekil 22. Veri toplama ve ek açıklama iş akışı

Veri kümesi, yüksek çözünürlüklü RGB görüntüleri içerir ve bu görüntüler, 2021 yılının Şubat ve Nisan ayları arasında Tamilnadu, Hindistan'daki Tirunelveli bölgesine yakın bir köydeki gerçek çeltik tarlalarından yüksek çözünürlüklü bir akıllı telefon kamerası (CAT S62 Pro) kullanılarak toplanmıştır. Paddy bitkisinin yaşı, veri toplama sürecinde 40 ila 80 gün

arasında deęişkenlik göstermiştir. Toplanan görüntüler titizlikle temizlenmiş, kötü ve yinelenen görüntüler çıkarılarak son veri kümesi 16.225 görüntüye indirgenmiştir.

Her görüntü, ziraat mühendisleri yardımıyla hastalık semptomlarının varlığına göre manuel olarak etiketlenmiş ve karşılık gelen bir etiket atanmıştır. Veri seti, 10 sınıfa ayrılan 9 farklı çeltik hastalığı ve sağlıklı yapraklar için uygun olan verileri içermektedir. RGB görüntülerinin yanı sıra, veri kümesi her yaprak görüntüsü için çeşitlilik ve görüntülerin toplandığı çeltik bitkisinin yaşı gibi meta verilerini içerir.

Veri kümesinin kullanılabilirliğini artırmak için dört farklı varyantı yayınlanmıştır. İlk varyant, orijinal yüksek çözünürlüklü (1080x1440) görüntüleri içerirken, ikinci ve üçüncü varyantlar sırasıyla 480x640 piksel ve 256x256 piksel çözünürlükte yeniden boyutlandırılmış görüntüleri içerir. Dördüncü varyant, sınıf etiketlerine ve çeltik çeşidine göre eğitim ve test setleri içeren 256x256 piksel görüntülerden oluşur. Veri kümesi yüksek kaliteli veri toplama teknikleri kullanılarak toplanmıştır ve her görüntü elle işaretlenmiştir. Görüntülerin yanı sıra, her bir yaprak görüntüsü için çeşitlilik ve yaşı gibi meta veriler de sağlanmaktadır. Ayrıca, veri kümesinin kullanılabilirliğini artırmak için, farklı boyutlarda dört farklı varyantı bulunmaktadır. Bunlar, orijinal yüksek çözünürlüklü görüntülerin yanı sıra, daha düşük çözünürlüklü ve yeniden boyutlandırılmış görüntülerin de bulunduğu farklı boyutlardaki varyantlardır.

Paddy Doctor veri kümesi, bilgisayarlı görü işleme algoritmalarının verimli ve sağlam pirinç hastalığı tanı sistemleri için kıyaslama oluşturulması ve deney yapılması için tasarlanmıştır. Ayrıca, doğru ve erken bir şekilde hastalık tespiti yapmak, önemli verim kaybını önlemek ve pestisit kullanımını azaltmak için gereklidir.

Sonuç olarak, Paddy Doctor veri kümesi pirinç hastalığı teşhisi için uzman tarafından işaretlenmiş mevcut olan en büyük görüntü veri kümesidir. Veri kümesi, verimli ve sağlam pirinç hastalığı tanı sistemlerinin geliştirilmesine imkan sağlar ve farklı bilgisayarlı görü işleme algoritmalarının değerlendirilmesi ve karşılaştırılması için bir kıyaslama görevi görebilir.

Çeltik yaprak hastalıklarını teşhis etmek için kullandığımız veri seti, Paddy Doctor (Kiruba ve Arjunan 2023) veri setidir. Bu veri seti, sağlıklı ve hastalıklı örnekler de dahil olmak üzere toplamda 10 hastalık sınıfına ait 10.407 yaprak görüntüsü içermektedir. Güncel ve yüksek çözünürlüklü Paddy Doctor veri seti, bakteriyel yaprak yanıklığı, bakteriyel yaprak çizgisi, bakteriyel başak yanıklığı, blast, kahverengi benek, dead heart, tüylü küf, hispa, tungro ve normal sınıflarından oluşmakta ve bu sınıflara ait görüntü sayıları Tablo 2'de listelenmektedir. Ayrıca her sınıf için veri yüzdeleri Şekil 23'te gösterilmektedir.

Tablo 2. Paddy Doctor Veri Seti İçerisindeki Çeltik Hastalığı Sınıflarına İlişkin İstatistiksel Veriler

Sınıf Adı (Çeltik Hastalığı)	Toplam (%100)	Eğitim (%70)	Doğrulama (%15)	Test (%15)
Bacterial Leaf Blight (BLB)	479	335	72	72
Bacterial Leaf Streak (BLS)	380	266	57	57
Bacterial Panicle Blight (BPB)	337	237	50	50
Blast	1738	1218	260	260
Brown Spot	965	675	145	145
Dead Heart	1442	1010	216	216
Downy Mildew	620	434	93	93
Hispa	1594	1114	240	240
Normal	1764	1234	265	265
Tungro	1088	762	163	163

Kaggle yarışmasında kullanılmak üzere Paddy Doctor veri setinin sadece 10 hastalık sınıfına ait 10.407 eğitim görüntüleri halka açık olarak paylaşılmıştır. Veri setindeki tüm görüntülerin tek bir klasörde toplanmış olması nedeniyle, modellerin genelleme yeteneklerini ölçmek için ayrı bir doğrulama veya test verisi mevcut değildi. Bu yüzden Tablo 2’de görüldüğü gibi CNN modellerin performans analizi için veri setindeki her sınıf için %70’i eğitim, %15’i doğrulama ve kalan %15’i ise test verisi olarak rastgele atanmıştır. Bu oran daha küçük veri setleri için tercih edilmektedir. Bu sayede modeller eğitilirken aşırı uydurma (overfitting) problemi önlenerek gerçek dünya verilerinde model performansları test edilebilir. Şekil 23’te her bir sınıf için veri yüzdeleri gösterilmektedir.

Bakteriyel yaprak yanıklığı (Bacterial Leaf Blight (BLB))

Bakteriyel Yaprak Yanıklığı (BLB), öncelikle pirinç bitkilerini etkileyen *Xanthomonas oryzae* bakterisi tarafından neden olan bir bitki hastalığıdır. Hastalık, düşük verim, kötü tahıl kalitesi ve ciddi vakalarda tam bir mahsul başarısızlığına neden olabilir.

BLB'nin belirtileri, hastalığın ilerlemesiyle sararmaya ve ardından kahverengileşmeye başlayan yapraklarda su emmiş lezyonlardır. Lezyonlar ayrıca, lezyonun etrafındaki doku sağlıklı dokunun etrafındaki dokudan daha açık renkte görüldüğü "hale" etkisi de olabilir.

Enfekte bitkilerde, gelişme geriliği de görülebilir ve hastalık yüksek nem ve sıcaklığın olduğu uygun koşullarda hızla yayılabilir (CM Vera Cruz 2023).

Bakteriyel yaprak lekesi (Bacterial Leaf Streat (BLS))

Bakteriyel Yaprak Lekesi (BLS), *Xanthomonas translucens* pv. *undulosa* bakterisi tarafından oluşturulan bir bitki hastalığıdır. Pirinç, mısır, buğday ve arpa dahil olmak üzere birkaç çim türünü etkiler.

BLS, enfekte bitkilerin yapraklarında uzun, dar, kahverengimsi gri çizgilerin ortaya çıkmasıyla karakterize edilir. Çizgiler birkaç inç uzunluğa kadar çıkabilir ve genellikle çizgilerin etrafındaki yaprak dokusunun sararmasıyla birlikte görülür. Hastalık ilerledikçe, çizgiler daha belirgin hale gelebilir ve nihayetinde birleşerek tüm yaprağı kahverengiye döndürüp ölümüne neden olabilir (Sparks 2023a).

Bakteriyel başak yanıklığı (Bacterial Panicle Blight (BPB))

Bakteriyel Başak Yanıklığı (BPY), pirinç bitkilerini etkileyen bir bitki hastalığıdır. *Burkholderia glumae* bakterisi tarafından oluşturulur ve pirinç başaklarında (bitkinin çiçeklenme bölümü) kahverengi ile siyah arasında değişen lekelerin varlığıyla karakterizedir. Hastalık, özellikle Asya ve Güney Amerika Birleşik Devletleri'nde sıcak ve nemli ortamlarda yaygın olarak görülür.

BPY, pirinç üretiminde önemli bir etkiye sahip olabilir ve %50'ye kadar verim kaybına neden olabilir. Hastalık ayrıca, pirinç tanesi kalitesini de azaltarak, tüketim veya işleme için uygun olmayacak hale getirebilir (Ham vd 2011).

Blast

Blast hastalığı, pirinç bitkilerini etkileyen bir mantar hastalığıdır. *Magnaporthe oryzae* adı verilen bir mantar tarafından oluşturulur. Mantar, yapraklar, saplar ve taneler de dahil olmak üzere pirinç bitkisinin tüm bölümlerine bulaşabilir. Blast hastalığı pirinç ürünlerinde önemli hasarlara neden olabilir ve verim kaybına yol açabilir.

Pirinç bitkilerindeki blast hastalığı belirtileri, yapraklar, saplar ve tanelerde kahverengi ile gri lezyonlar ve etkilenen bitki kısımlarının renk değişimini içerir. Enfekte yapraklar ayrıca küçük, koyu lekeler oluşturabilir ve zamanla düzensiz şekilli lekeler haline gelebilir (Dean vd 2012).

Kahverengi leke (Brown Spot)

Kahverengi leke hastalığı, ayrıca *Helminthosporium* yaprak lekesi olarak da bilinen, pirinç, mısır, buğday ve soya fasulyesi gibi çeşitli tarım ürünlerini etkileyen bir fungal hastalıktır. Bitki artıklarında ve toprakta hayatta kalabilen *Bipolaris oryzae* fungusu (önceden *Helminthosporium oryzae* olarak bilinirdi) tarafından oluşturulur.

Kahverengi leke hastalığı belirtileri, enfekte bitkilerin yapraklarında, saplarında ve tahıllarında sarımsı sınırlı küçük oval kahverengi lekelerdir. Lekeler, daha büyük ölü doku alanlarını oluşturacak şekilde birleşebilir ve bu durumda verim kaybına neden olabilir (Zeigler ve Barclay 2008).

Ölü kalp (Dead Heart)

BLB, BLS, BPB ve Blast gibi bir dizi hastalık, pirinç bitkilerinin büyüme noktalarını etkiler ve genç yapraklar ile gövde öldüğünde, ölü bir kalbi andıran kahverengi ve solmuş doku bırakır. Ölü Kalp adı verilen bu durum, verim ve büyümenin azalmasına neden olabilir ve genellikle mantar ve bakteriyel hastalıklardan kaynaklanır. (Jena ve Kim 2010).

Tüylü küf (Downy Mildew)

Tüylü küf, *Peronosclerospora oryzae* mantar patojeni tarafından oluşturulan yaygın bir pirinç hastalığıdır. Bu hastalık, pirinç bitkilerinin herhangi bir büyüme aşamasını etkileyebilir ve önemli verim kayıplarına neden olabilir.

Tüylü küfün belirtileri, yapraklarda sarıdan kahverengi lekeler kadar değişen lezyonlar ve bazen yaprakların alt tarafında gri veya beyaz tüylü bir büyüme ile görülür. Hastalık ilerledikçe, lezyonlar nekrotik hale gelebilir ve yapraklar önceden kuruyup ölebilir. Şiddetli durumlarda, hastalık gövde ve başakları da etkileyebilir, bu da azalan başak boyutlarına ve tane verimine neden olabilir. Hastalık, yüksek nem ve 20-30°C arası sıcaklıklarda görülür (Sujithra ve Ukrit 2020).

Hispa

Halk arasında pirinç yaprak böceği olarak bilinen bir haşere olan pirinç hispa, pirinç yapraklarının üst yüzeyini kazıyarak ve yaprak dokularında tünel açarak %20'ye varan verim kayıplarına neden olan zararlı bir böcektir. Yağış, yetersiz yağış, minimum gece-gündüz sıcaklık farkı, yüksek nem ve çeltik tarlalarındaki yabani otlar gibi faktörler bu zararlının büyümesini artırır. Aşırı gübrelemeden kaçınmak ve mahsulleri sık aralıklarla dikmek gibi kültürel kontrol yöntemleri etkili olabilir. Küçük eşek arıları, reduviid böcekler ve üç tür mantar kullanımı gibi biyolojik kontrol yöntemleri de kullanılabilir. Kimyasal kontrol yöntemleri de

kullanılabilir, ancak diğer kontrol yöntemleriyle birlikte entegre bir haşere yönetimi yaklaşımı uygulanmalıdır. Düzenli saha denetimleri ve erken müdahale, haşere zararının önlenmesi için önemlidir (Catindig 2023). Şekil 24’te Paddy Doctor veri setinden her bir hastalığa ait rastgele seçilmiş görüntüler gösterilmektedir.

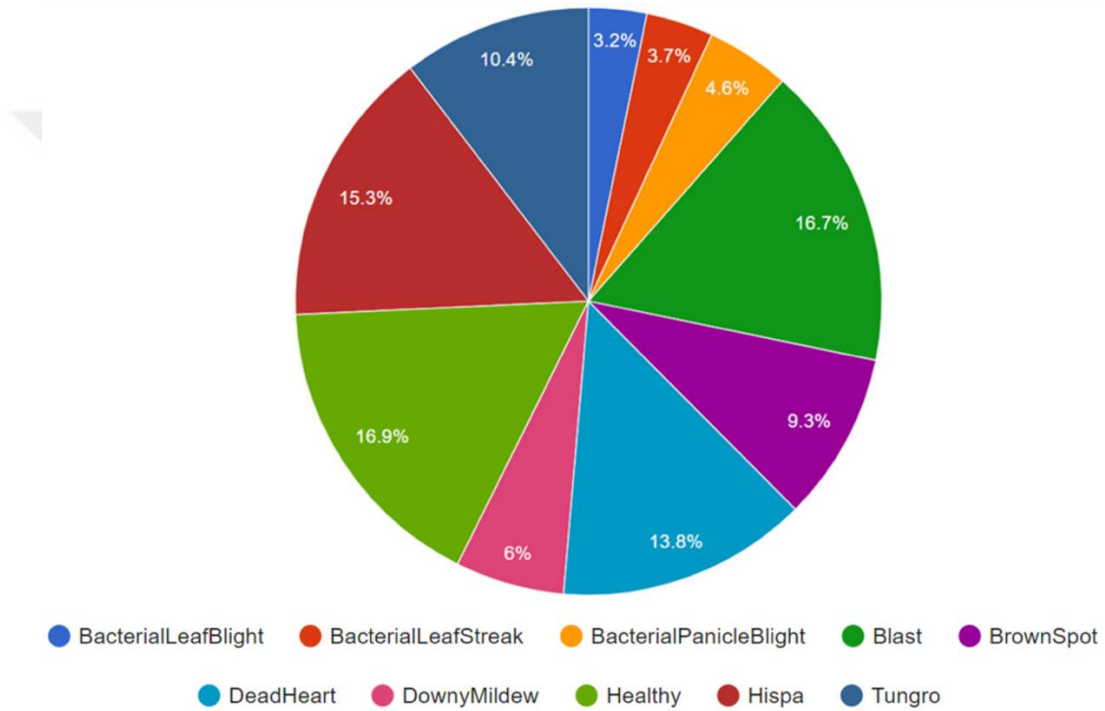


Şekil 23. Paddy Doctor veri setinden her bir hastalık sınıfına ait rastgele seçilmiş görüntüler (1 Sağlıklı - 9 Hastalıklı)

Tungro

Tungro, önemli verim kayıplarına neden olabilen bir pirinç virüs hastalığıdır. Hastalık, yeşil yaprak hopper (*Nephotettix virescens*) adlı bir böcek taşıyıcısı aracılığıyla iletilen iki virüs tarafından oluşturulur. Bu hastalıklar, pirinç tungro küresel virüsü (RTSV) ve pirinç tungro çubuk şekilli virüsüdür (RTBV).

Tungro belirtileri enfeksiyonun şiddetine, bitkinin yaşı ve büyüme evresine bağlı olarak değişebilir. Genç fideler, cüceleşme, sararma ve solgunluk gösterebilir. Enfekte olmuş daha büyük bitkiler, yaprakların sararması veya kızarması, azalmış başak boyutu gösterebilir. Enfekte bitkilerin yaprakları ayrıca dikleşebilir ve şekilsizleşebilir (Sparks 2023b).



Şekil 24. Her sınıf için veri yüzdeleri

Ziraat Alanında Derin Öğrenme

Son yıllarda, derin öğrenme mimarilerinin ziraat alanında kullanımı hızla artmaktadır. Bu teknikler, büyük miktarda veri toplama ve analiz etme yeteneği sayesinde, bitki hastalıklarının teşhisinde ve tahmininde, toprak verimliliğinin artırılmasında ve bitki büyümesinin izlenmesinde başarıyla kullanılmaktadır.

Bitki hastalıklarının teşhisinde ve tahmininde, derin öğrenme teknikleri, bitki görüntüleri üzerinde çalışarak hastalıkları tespit etmek ve hastalık türlerini sınıflandırmak için kullanılmaktadır (Mohanty vd 2016). Bu teknikler, insan gözünün algılayamayacağı ince ayrıntıları tespit edebilir ve bu sayede hastalıkların erken teşhis edilmesi ve önlenmesi sağlanabilir.

Toprak verimliliğinin artırılmasında, derin öğrenme teknikleri, toprak özelliklerini analiz ederek bitki yetiştiricilerine hangi gübreleri ve ne zaman kullanmaları gerektiği konusunda tavsiyelerde bulunabilir (Zhang vd 2017). Bu sayede, topraktaki besin maddelerinin optimum seviyede tutulması ve bitki büyümesinin sağlıklı bir şekilde devam etmesi sağlanabilir.

Bitki büyümesinin izlenmesinde, derin öğrenme teknikleri, bitki görüntülerini analiz ederek bitki büyümesinin gelişimini takip edebilir ve bitkilerin ihtiyaç duydukları su, gübre ve diğer faktörlerin belirlenmesinde yardımcı olabilir (Krizhevsky vd 2012). Bu sayede, bitki yetiştiricileri bitkilerin ihtiyaçlarını daha iyi anlayabilir ve bitki büyümesini optimize ederek verimliliği artırabilir.

Sonuç olarak, derin öğrenme mimarileri, ziraat alanında kullanıldığında büyük bir potansiyele sahiptir. Bu teknikler, bitki hastalıklarının teşhisinde ve tahmininde, toprak verimliliğinin artırılmasında ve bitki büyümesinin izlenmesinde başarıyla kullanılabilir.

Derin Öğrenme Mimarileri

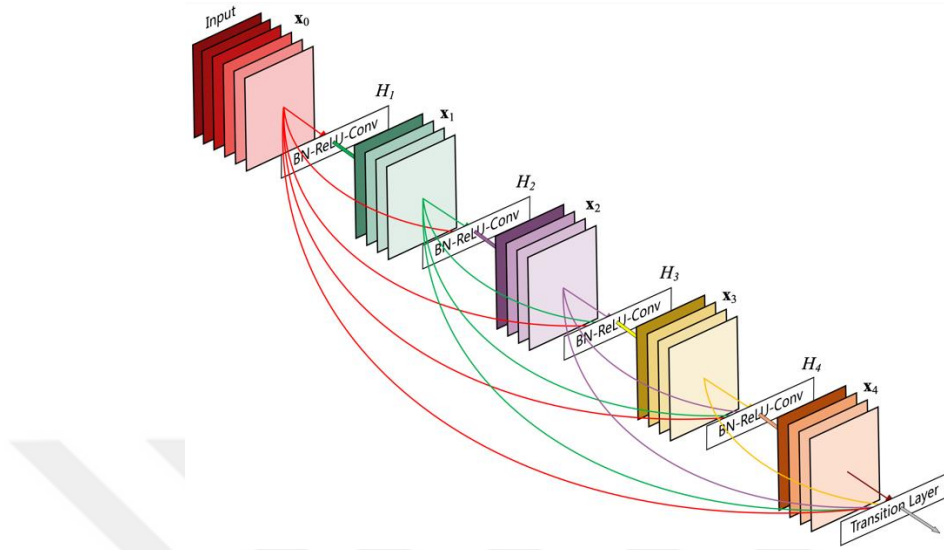
Derin öğrenme mimarileri, makine öğrenimi alanında çok katmanlı sinir ağlarının eğitime odaklanan bir dal olarak bilinmektedir (Karaman vd 2023). Doğal dil işleme, karar verme, görüntü tanıma ve konuşma tanıma gibi çeşitli görevler için kullanılan sinir ağları, derin öğrenme mimarilerinin temel yapı taşlarını oluşturur. Amaçları, insan beyninin bilgi işleme şeklini taklit etmektir ve büyük eğitim veri setleri ve güçlü bilgisayar kaynakları kullanarak performanslarını zamanla artırmaktır (Pacal vd 2020). Evrişimli sinir ağları (CNN'ler) ve tekrarlayan sinir ağları (RNN'ler), otonom araçlar, konuşma tanıma, doğal dil işleme ve bilgisayar görüşü gibi uygulamalarda yaygın olarak kullanılan derin öğrenme mimarileridir. Derin öğrenme modelleri, büyük veri kümelerini analiz etmek için yapay sinir hücreleri kullanır ve geniş veri kümeleri üzerinde eğitilirler (Gündüz ve Işık 2023). Performanslarını sürekli olarak iyileştirmek için daha fazla veri eklenir. TensorFlow, Keras ve PyTorch, popüler derin öğrenme çerçeveleridir. CNN'ler, her katmanda girdi verilerinde giderek daha karmaşık desenleri tanımaya çalışan çok katmanlı sinir ağlarıdır. Evrişimli katmanlar, girdi verilerini taramak ve özellikleri çıkarmak için filtreler kullanır ve özetleme katmanları, verinin mekânsal boyutunu azaltır. Tam bağlantılı katmanlar, çıkarılan özellikleri birleştirerek tahminler yapar.

Derin Öğrenme Modelleri

Derin öğrenme modelleri, yapay sinir ağı mimarileri kullanarak karmaşık veri setlerindeki desenleri öğrenen makine öğrenmesi algoritmalarıdır. En popüler derin öğrenme

modellerinden biri, evrişimli sinir ağlarıdır (CNN). Bu araştırmada, eğitim ve değerlendirme için kullanılan en popüler CNN modelleri aşağıda açıklanmaktadır:

DenseNet



Şekil 25. DenseNet mimarisi (G. Huang vd 2017).

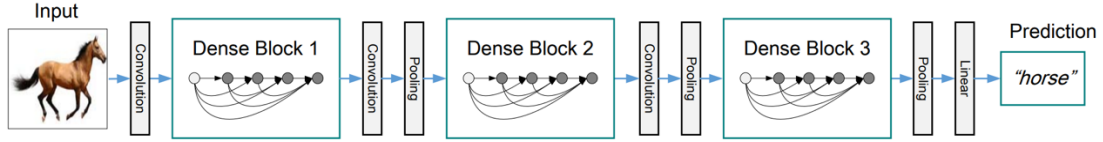
Tabanı Yoğun Bloklar (Dense Blocks) olan bir Sinir Ağı Mimarisi olan DenseNet (G. Huang vd 2017), düşük boyutlu ağlarda yüksek doğruluk elde etmek için tasarlanmıştır. 2016 yılında Kaiming He, Xiangyu Zhang, Shaoqing Ren ve Jian Sun tarafından yayınlanan bir makale, Şekil 25'te görülen DenseNet mimarisini tanımlamaktadır.

DenseNet mimarisi, ResNet mimarisindeki (Residual Network) gibi bir "skip connection" (atlayış bağlantısı) kullanır, ancak ResNet'teki gibi birimleri (blocks) eklemek yerine, her katmanda önceki tüm katmanların çıktılarını birleştirir. Bu, ağın öğrenme kapasitesini artırarak, vanishing gradient sorununu azaltır ve daha derin ağlar için bile etkili sonuçlar verir.

DenseNet blokları, evrişimli (convoluion) ve havuzlama (pooling) katmanlarından oluşur. Yoğun katmanlar (dense layers), bloklarda yer alır ve blok boyunca her katmanda birleştirilir. Yoğun bağlantılar (dense connections), her bloktaki yoğun katmanların çıktılarını birleştirir. Ayrıca, her blok sonunda bir geçiş katmanı (transition layer) yer alır, bu katman boyutu azaltmak için evrişimli ve havuzlama katmanlarından oluşur.

DenseNet, "growth rate" olarak adlandırılan bir parametre ile özelleştirilebilir. Ağın öğrenme kapasitesini yöneten Growth rate, katmanlara eklenecek filtre sayısını belirler. Bu parametre, ağın performansını etkiler ve farklı veri setleri için ayarlanabilir.

DenseNet mimarisi, ImageNet, CIFAR-10 ve CIFAR-100 veri setleri gibi birçok farklı veri setinde kullanılmış ve diğer sınıflandırma problemlerinde de başarıyla uygulanmıştır. Ayrıca, transfer öğrenme ile de oldukça başarılıdır ve özellik çıkarımı için kullanılabilir. Üç yoğun bloğa sahip derin bir DenseNet mimarisi Şekil 26’da gösterilmiştir.



Şekil 26. Üç yoğun bloğa sahip derin bir DenseNet (G. Huang vd 2017).

Çalışmaya dahil edilen DenseNet-121, DenseNet-169 ve DenseNet-201, hepsi de farklı sayıda katmanlardan oluşan farklı boyutlarda derin sinir ağı mimarileridir. Tablo 3’te, bu üç mimarinin ana özellikleri karşılaştırılmıştır.

Tablo 3. DenseNet Modellerinin Karşılaştırılması

Mimari	Toplam Katman Sayısı	Yoğun Blok Sayısı	Top-1 Doğruluk	Top-5 Doğruluk
DenseNet-121	121	4	74.9%	92.2%
DenseNet-169	169	4	76.0%	93.3%
DenseNet-201	201	23	77.6%	93.8%

DenseNet-121, en az sayıda katmana sahip olanıdır ve ImageNet veri setinde 1000 sınıf üzerinde eğitilmiştir. DenseNet-169, daha fazla sayıda katmana sahiptir ve bu nedenle daha derin bir mimariye sahiptir. DenseNet-201 ise, en yüksek sayıda katmana ve yoğun bloğa sahip olanıdır ve en büyük boyuta sahip olanıdır.

Bunların yanı sıra, DenseNet-201 daha büyük veri setleri ve daha karmaşık görevler için tasarlanmıştır. DenseNet-121 ve DenseNet-169 ise, daha küçük veri setleri ve daha az karmaşık görevler için uygundur. Ancak, bu mimarilerin performansı, veri setinin boyutuna ve özelliklerine, ağın hiperparametrelerine ve diğer faktörlere bağlı olarak değişir.

EfficientNetv2

EfficientNetV2 (Tan ve Le 2019), Google tarafından geliştirilen yeni bir derin sinir ağı mimarisidir. Bu mimari, önceki EfficientNet mimarisinden geliştirilmiştir ve daha yüksek performans ve daha az hesaplama maliyeti sağlar. Bu mimari, EfficientNetB0’dan EfficientNetB7’ye kadar farklı boyutlarda 8 farklı varyantı içermektedir.

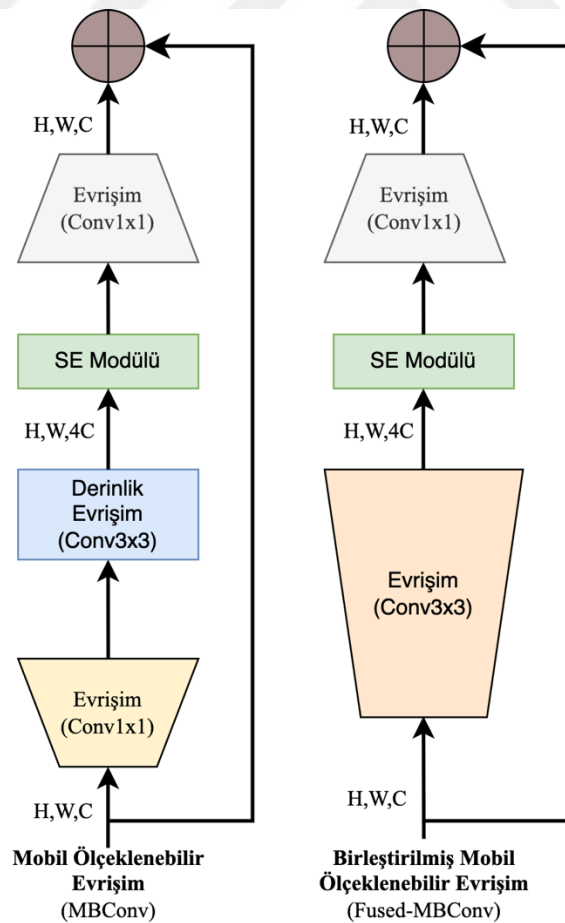
EfficientNetV2’nin ana bileşenleri, önceki EfficientNet mimarisindeki gibi, çeşitli

boyutlarda evrişimli, derinlemesine evrişimli ve darboğaz (bottleneck) bloklarından oluşur. Ancak, yeni mimaride bloklar daha fazla paralelleştirilir ve aktivasyon fonksiyonları Swish yerine Mish olarak kullanılır. Bunun yanı sıra, modelin optimizasyonu için daha iyi bir ölçüt olan Softmax-CE yerine JSD (Jensen-Shannon Divergence) kullanılır (Tan ve Le 2019).

Mimarideki en önemli değişiklik, önceki EfficientNet mimarisindeki her blok için farklı ölçeklendirme, genişlik ve derinlik katsayılarının kullanılması yerine, her varyant için tek bir katsayı takımının kullanılmasıdır. Bu, mimarinin daha hızlı eğitilmesine ve daha düşük hesaplama maliyetine neden olur (Tan ve Le 2019).

EfficientNetV2 mimarisinde, aynı zamanda bir "stem" bloğu da kullanılır. Bu blok, giriş görüntülerinin boyutunu azaltır ve daha sonra ilk evrişimli bloğu besler. Bu durum daha az sayıda parametre ile daha yüksek doğruluk sağlamaya yardımcı olur.

Ayrıca, EfficientNetV2'nin performansı, bileşik ölçekleme (compound scaling) adı verilen bir teknik kullanılarak artırılmıştır. Bu teknik, EfficientNet'in önceki sürümlerinde kullanılan çeşitli hiperparametrelerin, ölçeklendirme, genişlik ve derinlik katsayılarının tek bir parametre takımı olarak birleştirilmesini içerir (Tan ve Le 2019).



Şekil 27. EffigientNetV2 mimarileri (Tan ve Le 2019).

Sonuç olarak, EfficientNetV2, daha yüksek doğruluk ve daha az hesaplama maliyeti sağlayan yenilikçi bir derin sinir ağı mimarisidir. Bu mimari, özellikle mobil ve gömülü cihazlar için daha verimli ve kullanışlı bir yapıya sahiptir.

EfficientNetv2 mimarisi, özelleştirilmiş bileşik ölçekleme (compound scaling) yaklaşımı kullanarak, farklı boyutlardaki modellerin aynı temel mimariyi kullanarak ölçeklendirilmesine olanak tanır. Bu yaklaşım, farklı boyutlardaki modellerin aynı anda eğitilmesine izin verir ve verimli bir şekilde hiperparametrelerin ayarlanmasına yardımcı olur. EfficientNetV2 modellerinde kullanılan MBConv ve Birleştirilmiş MBConv mimarileri Şekil 27'de gösterilmiştir.

Çalışmada kullanılan EfficientNetv2 modellerinden EfficientNetv2-Small, EfficientNetv2-Medium ve EfficientNetv2-Large arasındaki fark, boyutları ve içerdikleri katmanların sayısıdır. EfficientNetv2-Small en küçük boyutlu modele sahipken, EfficientNetv2-Large en büyük boyutlu modele sahiptir.

EfficientNetv2-Small, 7M parametre içeren en küçük modeldir. Giriş görüntülerinin boyutunu 224x224'ten 128x128'e kadar indirger. EfficientNetv2-Small, 7 adet "MBConv" bloğu içerir. MBConv blokları, önceki EfficientNet mimarilerinde kullanılan "Inverted Residual Block" bloklarının geliştirilmiş versiyonudur. Her bir MBConv bloğu, önce 1x1 boyutunda bir expansion convolution, sonra bir depthwise convolution ve en sonunda bir 1x1 boyutunda bir projection convolution içerir. Bu bloklar, daha düşük hesaplama maliyetleri ile daha yüksek doğruluk sağlar.

EfficientNetv2-Medium, 21M parametre ile daha büyük bir modeldir. Giriş görüntülerinin boyutunu 240x240'tan 144x144'e kadar azaltır. EfficientNetv2-Medium, EfficientNetv2-Small'da olduğu gibi 7 adet MBConv bloğu içerir, ancak daha büyük boyutlu bloklar kullanır.

EfficientNetv2-Large, 480M parametre ile en büyük boyutlu modele sahiptir. Giriş görüntülerinin boyutunu 384x384'ten 96x96'ya kadar indirger. EfficientNetv2-Large, EfficientNetv2-Medium'da olduğu gibi 7 adet MBConv bloğu içerir, ancak daha büyük boyutlu bloklar kullanır ve daha fazla depthwise convolution katmanına sahiptir.

Bu farklı boyutlu modeller, farklı hesaplama maliyetlerinde ve farklı doğruluklarda farklı uygulama senaryoları için kullanılabilir. EfficientNetv2-Small, mobil cihazlar veya işlem gücü kısıtlı cihazlar için daha uygunken, EfficientNetv2-Large daha büyük ve daha güçlü bilgisayarlarda kullanılabilir. Tablo 4, EfficientNetv2 modellerinin özelliklerini ve farklarını özetlemektedir:

Tablo 4. EfficientNetv2 Modellerinin Karşılaştırılması

Model	Parametre Sayısı	Giriş Boyutu	MBConv Blok Sayısı
EfficientNetv2-Small	7M	128x128	7
EfficientNetv2-Medium	21M	144x144	7
EfficientNetv2-Large	480M	96x96	7

Tabloda da görüldüğü gibi, EfficientNetv2-Small en küçük boyutlu modele sahiptir ve en az parametre sayısına sahiptir. EfficientNetv2-Large ise en büyük boyutlu modele sahiptir ve en fazla parametre sayısına sahiptir. Her üç model de 7 adet MBConv bloğu içerir, ancak boyutları ve içerdikleri katmanların sayısı farklıdır.

Inception

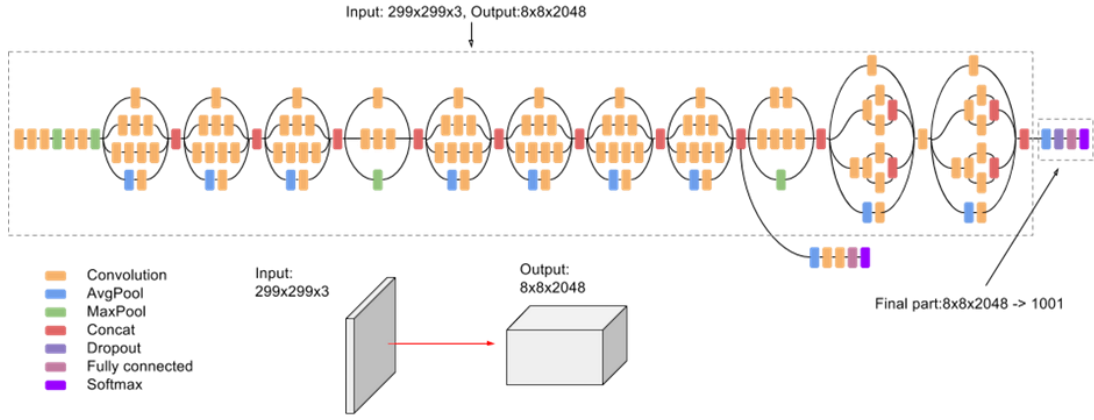
Inception (Szegedy vd 2015), Google tarafından 2014 yılında tanıtılan ve derin sinir ağlarının daha verimli hale getirilmesi amacıyla tasarlanmış bir mimaridir. Bu mimari, ağırlık farklı boyutlu filtrelerle çalışarak hem hesaplama maliyetini düşürmek hem de ağırlık doğruluğunu artırmak için tasarlanmıştır.

Inception mimarisi, geleneksel olarak kullanılan küçük filtrelerle çalışan ağlara kıyasla daha büyük filtrelerle çalışır. Bununla birlikte, büyük filtrelerin kullanımı, ağırlık derinliğinin artmasıyla birlikte hesaplama maliyetini artırır. Bu sorunu çözmek için Inception mimarisi, farklı boyutlarda filtrelerin ağırlık farklı yerlerinde kullanılmasını önerir.

Inception mimarisi, modül denilen temel yapı bloklarından oluşur. Bu modüller, farklı boyutlarda filtreleri bir araya getirerek, girdi verilerini paralel olarak işler. Her bir modül, birkaç farklı filtre boyutuna sahip evrimsel katmanlarının bir arada kullanımıyla oluşturulur.

Inception mimarisi, geleneksel olarak kullanılan evrimsel katmanlarının yanı sıra, 1x1 boyutunda filtreleri kullanan evrimsel katmanlarını da kullanır. Bu 1x1 boyutundaki filtreler, girdi verilerinin boyutunu azaltmak veya artırmak için kullanılır ve hesaplama maliyetini düşürür.

Inception mimarisi ayrıca, havuzalam katmanlarının yanı sıra, girdi verilerindeki özellikleri bir araya getirmek için birleştirme (concatention) işlemlerini de kullanır. Bu işlem, farklı boyutlarda filtrelerin kullanımını birleştirerek, ağırlık doğruluğunu artırır. Inception mimarisi, çeşitli boyutlarda filtreleri kullanarak hesaplama maliyetini düşürürken ağırlık doğruluğunu artırır. Bu nedenle, birçok görev için başarılı bir şekilde kullanılmıştır.



Şekil 28. Inception-v3 modelinin görünümü (Szegedy vd 2016).

Çalışmaya dahil edilen Inception-v3 ve Inception-v4, Inception mimarisinin gelişmiş sürümleridir. İki mimari arasındaki ana fark, Inception-v4'ün daha derin ve daha karmaşık bir ağa sahip olmasıdır. İşlem yükü ve hesaplama gücü açısından daha yüksek bir yapıya sahip olan Inception-v4, birçok benzersiz teknik ve bileşen içerir.

Tablo 5, Inception-v3 ve Inception-v4'ün farklı özelliklerini özetleyerek, modeller arasındaki farkları açıklamaktadır.

Tablo 5. Inception Modellerinin Karşılaştırılması

Özellikler	Inception-v3	Inception-v4
Ağ yapısı	Derinlik: 159	Derinlik: 300
Modüller	3x3 ve 5x5 Conv, Max Pooling	1x1, 3x3 ve 5x5 Conv, Max Pooling
Doğrusal birimler	Kullanılmadı	Kullanıldı
Sıralı strateji	Standart	Geliştirilmiş
Gözetimli eğitim	Doğrusal olmayan bileşenlerle	Doğrusal ve gözetimli
Çok ölçekli özellik haritası	Standart	Kullanıldı

Inception-v3 ve Inception-v4 arasındaki bazı farklılıklar:

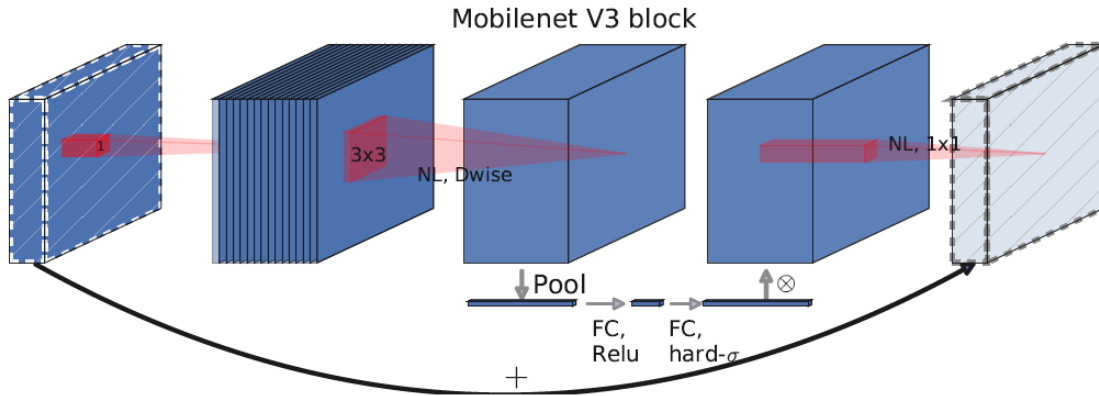
1. Daha derin ağ: Inception-v4, Inception-v3'ten daha derin bir ağa sahiptir. Inception-v4'te, 4 ekstra Inception modülü ve 1 ekstra Reduction modülü vardır.
2. Doğrusal birimler: Inception-v4'te, doğrusal birimler kullanılarak hesaplama gücü artırılmıştır. Bu, ağın daha iyi sonuçlar vermesine yardımcı olabilir.
3. Geliştirilmiş sıralı strateji: Inception-v4, Inception-v3'ten farklı bir sıralı strateji kullanır. Bu strateji, ağın daha iyi performans göstermesini sağlar.
4. Gözetimli eğitim: Inception-v4, gözetimli eğitim ile eğilirken, Inception-v3 sadece doğrusal olmayan bileşenlerle eğilmiştir. Bu, Inception-v4'ün daha doğru sonuçlar

vermesine yardımcı olabilir.

5. Çok ölçekli özellik haritası: Inception-v4, çeşitli ölçeklerde özellik haritalarını birleştirerek, nesnelerin farklı boyutlarına daha iyi uyum sağlar.

MobileNetv3

MobileNetV3 (Howard vd 2017), mobil cihazlarda ve diğer sınırlı ortamlarda gerçek zamanlı olarak çalışabilen yüksek doğruluklu bir derin öğrenme modelidir. Bu çalışmada, MobileNetV3 mimarisinin iki farklı sürümü olan "Small" ve "Large" incelenmiştir. MobileNetV3 mimarisi Şekil 29'da gösterilmiştir.



Şekil 29. MobileNetV3 modülü (Howard vd 2017).

Her iki MobileNetV3 mimarisi de ImageNet veri kümesi üzerinde eğitilmiştir ve diğer mobil modellerle karşılaştırıldığında daha yüksek doğruluklar elde edilmiştir. MobileNetV3 ayrıca, mobil cihazlar için yüksek performanslı ve hafif modellerin gerektirdiği sınırlamalara uygun olması nedeniyle gerçek zamanlı uygulamalarda da kullanışlıdır.

Çalışmada kullanılan MobileNetv3-Small ve MobileNetv3-Large, aynı yapıya sahip birçok özellik paylaşmasına rağmen farklı boyut ve hesaplama gücüne sahiptir.

MobileNetv3-Small, daha küçük boyutlu mobil cihazlar için optimize edilmiş bir mimaridir. Toplam 2.9 milyon parametreye ve 66.6 MOPS'a sahiptir. Özellikle, küçük çekirdek boyutları, azaltma katmanlarının daha az parametreye sahip olması ve mobil cihazlarda kullanılan çeşitli optimizasyon teknikleri, küçük boyutunu ve hesaplama gücünü mümkün kılar.

MobileNetv3-Large, daha büyük ve güçlü cihazlar için optimize edilmiş bir mimaridir. Toplam 5.4 milyon parametreye ve 219 MOPS'a sahiptir. Daha büyük boyutu, daha büyük çekirdek boyutları, daha fazla özellik haritası ve daha derin ağ yapısı ile daha yüksek doğruluk sağlar. MobileNetv3-Small'a göre daha yüksek hesaplama gücüne sahiptir ve daha yüksek doğruluk elde etmek için tasarlanmıştır.

Tablo 6’de görüldüğü gibi, MobileNetv3-Large mimarisi MobileNetv3-Small mimarisine göre daha yüksek hesaplama gücüne, daha yüksek parametre sayısına ve daha yüksek yapılandırmaya sahiptir. Ayrıca, MobileNetv3-Large mimarisi, SE modülü kullanarak, MobileNetv3-Small mimarisine göre daha yüksek yapay nesne algılama performansına sahiptir. Bununla birlikte, MobileNetv3-Small mimarisi MobileNetv3-Large mimarisine göre daha yüksek hesaplama verimliliğine sahiptir.

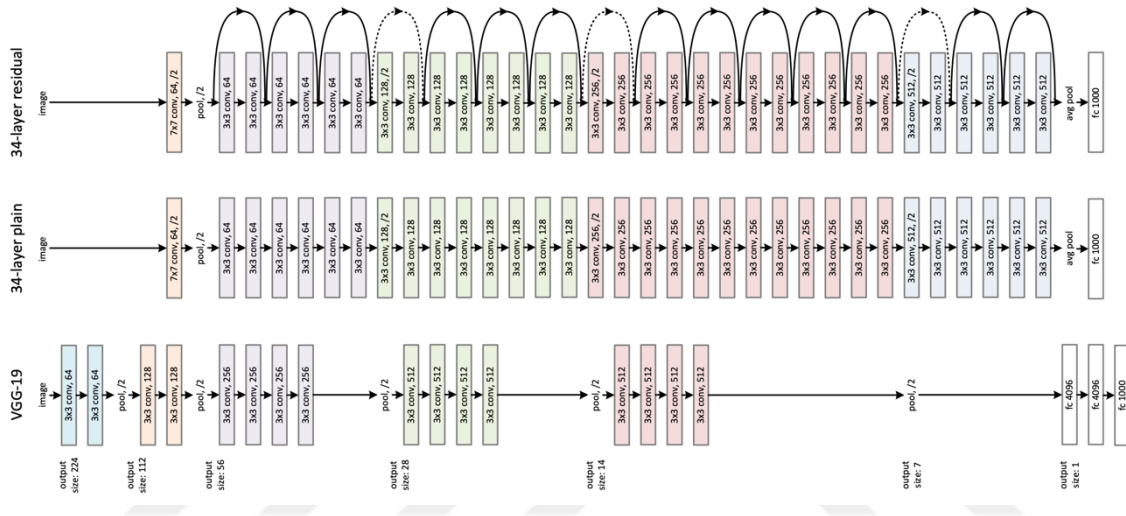
Tablo 6. MobileNetv3 Modellerinin Karşılaştırması

Özellikler	MobileNetv3-Small	MobileNetv3-Large
Girdi Boyutu	224x224	224x224
İşlem Sayısı	66M	219M
Parametre Sayısı	2.9M	5.4M
Hesaplama Gücü (GFLOPs)	27.4	155.1
Yapılandırma	1.0, 0.75,	1.4, 1.17, 1.0
Derinlik Çarpanı	5	5
Blok Sayısı	7	17
Aktivasyon Fonksiyonu	ReLU6	h-swish
Dropout Oranı	0.2	0.2
Genişleme Faktörü	4	6
Depthwise Aktivasyon Fonk.	ReLU6	h-swish
Depthwise Conv. Padding	same	same
Pointwise Aktivasyon Fonk.	ReLU6	h-swish
Pointwise Conv. Padding	same	same
SE Modülü	-	var
Hesaplama Verimliliği	Yüksek	Orta
Yapay Nesne Algılama Perform.	-	Yüksek

ResNet

ResNet (Residual Neural Network) (He vd 2016), derin sinir ağlarında karşılaşılan aşırı öğrenme (overfitting) sorununu çözmek için tasarlanmış bir mimardır.

ResNet, "skip connection" olarak adlandırılan bağlantılarla CNN katmanları arasına eklenen öğrenilebilir ek bağlantılar kullanır. Bu bağlantılar, önceki katmanlardan gelen girdileri, sonraki katmanlara doğrudan iletir. Bu sayede, gradyanların daha iyi akması ve daha az kaybolması sağlanır.



Şekil 30. ResNet mimarisi (He vd 2016).

VGG-19 modeli, 19,6 milyar FLOP hesaplama kapasitesine sahiptir (Alt):. Sadece 34 parametre katmanı (3,6 milyar FLOP) içeren bir ağ yapısı kullanılmıştır (Orta). Artık olarak kabul edilen bir ağ yapısıdır ve yine 34 parametre katmanı (3,6 milyar FLOP) içermektedir (Üst).

Şekil 30'da gösterilen ResNet mimarisi, bir dizi küçük öğrenme bloğu ile başlar. Bu bloklar, girdi verilerini işlemek için birkaç konvolüsyon ve aktivasyon katmanını içerir. Ardından, öğrenme blokları eklenir. "Skip connection" bağlantıları, önceki katmanın girdilerini doğrudan bir sonraki katmana iletme yöntemiyle bu blokların işlevini tamamlarlar. Bu sayede, ağın öğrenme yeteneği artar ve daha derin mimarilerin eğitilmesi mümkün hale gelir.

ResNet'in farklı versiyonları, öğrenme bloklarındaki konvolüsyon ve aktivasyon katmanlarının farklı sayılarını kullanarak farklı derinlik ve karmaşıklık seviyelerine sahiptir. ResNet mimarisi, farklı boyutlardaki ResNet modellerinden oluşur ve isimleri sırasıyla ResNet-18, ResNet-34, ResNet-50, ResNet-101 ve ResNet-152'dir.

ResNet-50, önceki versiyonlara göre daha derin ve karmaşıktır ve daha iyi performans

gösterir. Bununla birlikte, daha fazla öğrenme bloğu eklenerek ResNet-101 ve ResNet-152 gibi daha derin modeller elde edilebilir.

Araştırmaya dahil edilen ResNet34, ResNet50 ve ResNet101, ResNet ailesinin üyeleri olarak benzer bir yapıya sahiptir, ancak farklı sayıda katman içerirler. ResNet34, 34 katmandan oluşurken, ResNet50 ve ResNet101 sırasıyla 50 ve 101 katmandan oluşur.

ResNet50 ve ResNet101, ResNet34'e göre daha derin ve daha fazla parametreye sahip olmaları nedeniyle daha yüksek doğruluk oranlarına sahip olabilirler. Ancak, daha derin modeller daha fazla hesaplama gücü gerektirir ve daha az veriye sahip durumlarda aşırı uydurmaya (overfitting) neden olabilirler.

Ayrıca, ResNet50 ve ResNet101'de kullanılan darboğaz (bottlebeck) blokları, daha küçük boyutlu filtreler kullanarak daha az parametre ile daha etkili bir özellik çıkarımı sağlayarak, ResNet34'te kullanılan basit bloklardan farklılık gösterir. Bu nedenle, Tablo 7'de görüldüğü gibi ResNet50 ve ResNet101, ResNet34'e göre daha yüksek bir hesaplama verimliliği ve daha yüksek bir doğruluk oranı sağlar.

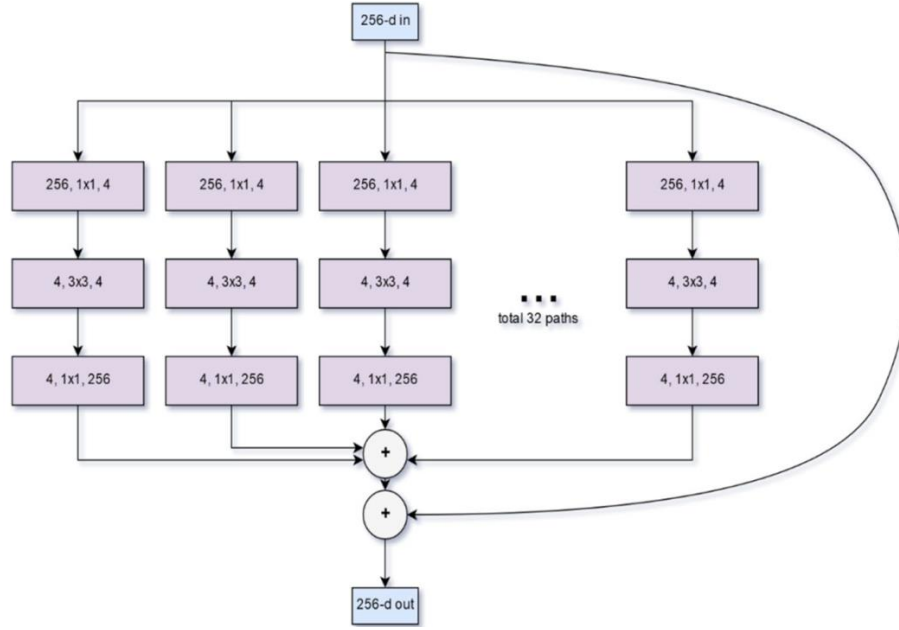
Tablo 7. ResNet Modellerinin Karşılaştırması

Model	Katman Sayısı	Parametre Sayısı	Giriş Boyutu	Top-1 Doğruluk	Top-5 Doğruluk
ResNet34	34	21.8 M	224x224	73.3%	91.6%
ResNet50	50	25.6 M	224x224	76.2%	93.1%
ResNet101	101	44.6 M	224x224	77.4%	93.8%

ResNeXt

ResNeXt (Xie vd 2017) mimarisi, ResNet mimarisine dayanmaktadır. Ancak, ResNet'teki blokların birden fazla yol içermesine dayanan "cardinality" adlı yeni bir parametre eklenerek, modelin performansı artırılmıştır.

ResNeXt'in blokları, gruplar olarak adlandırılan alt-ağlar içerir. Her bir grup içindeki paralel alt-ağlar, aynı boyutta filtreler kullanır, ancak farklı bağlantı ağlarına sahiptir. Grupların sayısı arttıkça, modelin performansı artar. ResNeXt mimarisi, ResNet'in yanı sıra daha küçük bir boyutlu veri seti olan CIFAR-10 ve CIFAR-100 üzerinde de eğitilebilir.



Şekil 31. ResNext yapısı (Dreizin vd 2021).

ResNeXt ayrıca, ImageNet veri kümesinde ResNet ve Inception modelleriyle rekabet edebilen ve bazı durumlarda onları aşan sonuçlar elde etmiştir. Görüntü sınıflandırma, nesne algılama ve görüntü segmentasyonu gibi farklı görevlerde oldukça başarılı sonuçlar ortaya koyan bir mimari olarak kabul edilen ResNeXt, sektörde dikkat çeken bir performans sergilemektedir. Şekil 31’de ResNeXt mimarisine ait yapı gösterilmektedir.

ResNeXt modellerinden olan ResNeXt34 ve ResNeXt50 modelleri arasındaki temel fark, modellerin derinliği (katman sayısı) ve genişlemesi (width)dir. ResNeXt34, 34 katman içeren bir modeldir. ResNeXt50 ise 50 katmana sahip bir modeldir. Dolayısıyla, ResNeXt50, ResNeXt34’ten daha derindir.

Diğer bir fark, genişlemeleridir. ResNeXt50, ResNeXt34’e göre daha genişletilmiştir. Bu, daha fazla parametreye sahip olduğu ve daha yüksek doğruluk elde etme potansiyeline sahip olduğu anlamına gelir. Ancak, daha fazla parametreye sahip olduğu için daha yüksek hesaplama maliyeti de gerektirir.

Ayrıca, ResNeXt50, daha yüksek boyutlu veri setleri için daha uygun olabilir. Çünkü daha fazla parametreleri, daha fazla veri üzerinde eğitim yapabilmesi ve daha yüksek doğruluk elde edebilmesi anlamına gelir. Tablo 8’de ResNet34, ResNet50, ResNet101, ResNeXt34 ve ResNeXt50 mimarilerinin özellikleri ve performansları görülmektedir. Depth sütunu, her bir modelin kaç katman içerdiğini göstermektedir. Parametre Sayıları sütunu, modelin öğrenilebilir parametre sayısını ifade etmektedir. Top-1 Doğruluk sütunu, modelin en iyi tahmin yapabilme başarısını, Top-5 Doğruluk sütunu ise en iyi tahminlerinin içinde doğru sınıfı bulma başarısını göstermektedir.

Tablo 8. ResNeXt Modellerinin Karşılaştırılması

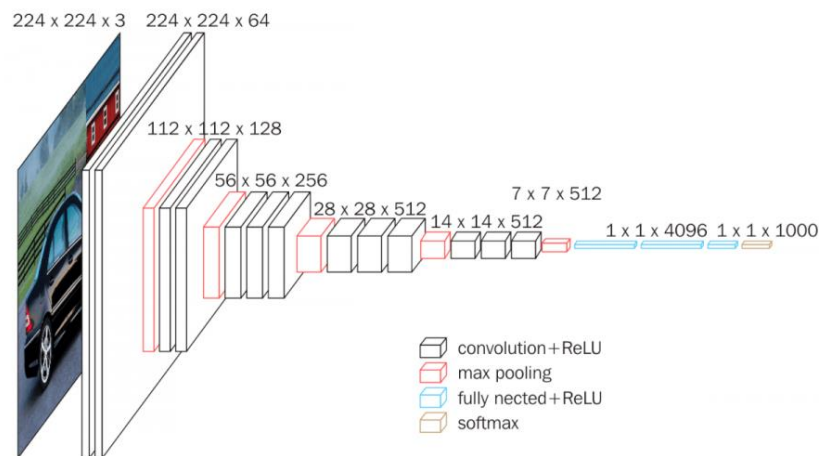
Model	Depth	Parametre Sayıları (M)	Top-1 Doğruluk (%)	Top-5 Doğruluk(%)
ResNet34	34	21.8	74.6	92.9
ResNet50	50	25.6	76.2	93.8
ResNet101	101	44.5	77.4	94.4
ResNeXt34	34	22.8	75.3	93.3
ResNeXt50	50	27.6	77.2	93.9

ResNet50, daha fazla katman ve parametreye sahip olmasına rağmen, ResNet34'ten daha yüksek bir doğruluk elde ederken, ResNet101 daha da yüksek bir doğruluk elde etmektedir. Benzer şekilde, ResNeXt50, ResNeXt34'ten daha fazla parametreye sahip olmasına rağmen, daha yüksek bir doğruluk elde etmektedir.

VGG

VGG (Visual Geometry Group) mimarisi (Simonyan ve Zisserman 2014), Microsoft Research ve University of Oxford'da geliştirilmiştir. ResNeXt mimarisi, ImageNet Large Scale Visual Recognition Challenge 2014 yarışmasında ikinci sırayı alarak önemli bir başarıya imza atmıştır.

VGG mimarisi, Şekil 32'de de görüldüğü gibi sıralı katmanlardan oluşan bir sinir ağıdır. Bir sonraki katmanın girdisi olarak, her katmanın çıktısı kullanılır. Mimaride 3x3 boyutlu filtreler kullanılır. Filtrelerin boyutları, Conv katmanları arasındaki maksimum havuzlama katmanlarından sonra yarıya indirilir. Bu sayede, derin bir ağ oluşturulurken, boyut sayısı kontrol altında tutulur. VGG, birbirine bağlı iki katman ile sonlanır.

**Şekil 32.** VGG sinir ağı mimarisi (Alshehri vd 2021).

VGG mimarisinde farklı boyutlarda üç farklı model kullanılmaktadır: VGG11, VGG16 ve VGG19. CNN filtrelerinin sayısını belirtmek için, bu modellerdeki sayılar kullanılır. VGG11 modelinde 11 katman, VGG16 modelinde 16 katman ve VGG19 modelinde 19 katman vardır. VGG mimarisi, özellikle görüntü sınıflandırma problemleri için kullanılmaktadır.

VGG modellerinden olan VGG13, VGG16 ve VGG19 mimarileri, ağırlık (weights) ve katman sayısı açısından farklılık gösterir. Tüm bu modeller temel olarak VGG mimarisine dayanır ve evrişimli sinir ağı (CNN) tabanlıdır. Model farklılıklarının detayları aşağıda verilmiştir.

1. Ağırlıklar (weights): VGG13, VGG16 ve VGG19 arasındaki temel fark, ağırlıklarıdır. VGG13, daha az sayıda parametreye sahipken VGG19, daha fazla parametreye sahiptir. Bu, VGG19'un daha derin bir model olduğu anlamına gelir.
2. Katman sayısı: VGG13, 13 adet katmana sahiptir, VGG16 16 adet katmana sahiptir VGG19 19 adet katmana sahiptir. Bu ek katmanlar, daha karmaşık modeller oluşturmak için kullanılır.
3. Genelleme yeteneği: VGG19, VGG16 ve VGG13'e kıyasla daha derin olduğundan, daha iyi bir genelleme yapma yeteneğine sahip olabilir. Ancak, daha fazla parametreye sahip olduğundan, daha fazla hesaplama gücü gerektirir ve daha fazla eğitim verilerine ihtiyaç duyar.
4. Performans: Genellikle daha derin modeller daha yüksek performans gösterir. Ancak, daha büyük modeller daha fazla hesaplama gücü gerektirdiğinden, performansla hesaplama arasında bir denge oluşturulması gereklidir.

Özetle, VGG13, VGG16 ve VGG19 modelleri arasındaki temel farklar, ağırlıklar, katman sayısı, genelleme yeteneği ve performans açısından farklılıklardır. Bu farklılıklar, farklı görevler için farklı model seçenekleri sunar.

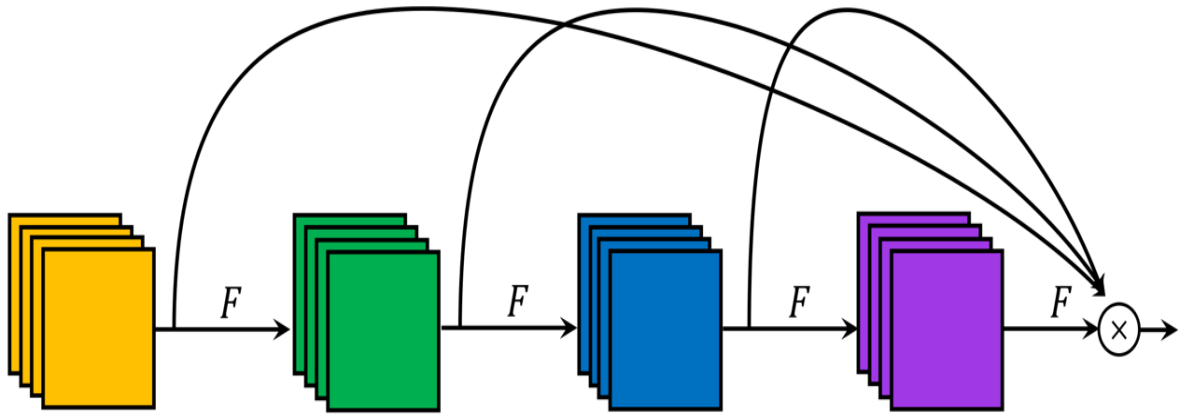
Tablo 9. VGG Modellerinin Karşılaştırılması

Model	Katman	Parametre	Top-1	Top-5
	Sayıları	Sayıları	Doğruluk	Doğruluk
VGG13	13	14.72M	71.09%	90.39%
VGG16	16	138.34M	71.59%	90.45%
VGG19	19	143.67M	72.38%	90.97%

Tablo 9'daki veriler, ImageNet veri kümesi üzerindeki doğruluk sonuçlarını göstermektedir. Modelin öğrenilebilir parametre sayısı ve doğruluk oranı, katman sayısı arttıkça artar; bu nedenle, kullanılan katmanların sayısı ile doğrudan ilişkilidir. VGG13, VGG16 ve VGG19 modelleri arasındaki en önemli fark, katman sayısı ve parametre sayısındaki artıştır. Ancak doğruluk oranları arasındaki farklar oldukça küçüktür.

VovNet

"An Energy and GPU-Computation Efficient Backbone Network for Real-Time Object Detection" makalesinde, 2019 yılında tanıtılan VovNet (Voting Block Network) mimarisi tanıtılmıştır (Y. Lee vd 2019). VovNet, sıradan konvolüsyon bloklarından (evrişimli block) ziyade, çeşitli ölçeklerdeki filtrelerin birleşiminden oluşan bloklar (voting block) kullanır. Bu bloklar, her bir ölçekteki filtrenin en iyi özellikleri ortak bir sonuç çıkarmak için birleştirir.



Şekil 33. VoVNet mimarisi (Y. Lee vd 2019).

Şekil 33'te gösterilen VovNet mimarisi, birkaç farklı boyuttaki filtreleri kullanarak farklı ölçeklerdeki özellikleri yakalar. Özellik çıkarmanın ilk aşamasında, özellik haritasını birleştirerek büyüten "expansion convolution" katmanları kullanılır. Daha sonra, bu genişletilmiş özellik haritaları, her bir ölçekteki filtreye uygulanan Vov blokları ile işlenir. Her Vov bloğu, çeşitli ölçeklerdeki filtrelerin çıktılarını birleştirir ve bunları ardışık bir şekilde birbirine ekler.

VovNet mimarisi, büyük ve küçük nesnelerin algılanmasında aynı anda yüksek performans gösterme kabiliyetine sahiptir. Bu, filtre boyutlarındaki farklılıklar sayesinde elde edilir. Ayrıca, VovNet mimarisi, yüksek hassasiyetli ve düşük hesaplama maliyeti nedeniyle gerçek zamanlı uygulamalar için de idealdir.

Tablo 10. VovNet Modellerinin Karşılaştırılması

Özellikler	VovNet39a	VovNet57a
Toplam Katman Sayısı	227	376
Toplam Parametre Sayısı	22,616,430	38,667,790
Stem Blk	3x3 Conv, stride=2	3x3 Conv, stride=2
	3x3 Conv, stride=1	3x3 Conv, stride=1
	3x3 Conv, stride=1	3x3 Conv, stride=1
	Max Pool, stride=2	Max Pool, stride=2
Artık VovNet Blok Sayısı	18	30
Lokal Olmayan Blok Sayısı	1	2
Sıkıştırma ve Uyarma Blok Sayısı	3	4
Sınıflandırıcı Blok	Global Average Pooling	Global Average Pooling
	Fully Connected (4096)	Fully Connected (4096)
	Fully Connected (1000)	Fully Connected (1000)
	Softmax	Softmax
En Yüksek Doğruluk	80.9%	81.9%

VovNet39a ve VovNet57a, aynı VovNet mimarisi ailesine ait evrişimli sınır ağı modelleridir, ancak farklı boyutlara ve hesaplama kapasitelerine sahiptirler. İşte bu iki model arasındaki bazı farklar:

1. Model Boyutu: VovNet39a, 27.6 milyon parametreye sahipken, VovNet57a 45.2 milyon parametreye sahiptir. Bu nedenle, VovNet57a daha büyük bir modeldir.
2. Blok Sayısı: VovNet39a modeli, 5 blok tipi ile 39 evrişimli katmana sahipken, VovNet57a modeli, aynı 5 blok tipiyle 57 evrişimli katmana sahiptir. Bu, VovNet57a'nın daha derin bir modele sahip olduğu anlamına gelir.
3. Derinlik: VovNet39a, giriş görüntüsünü işlemek için 3 stem katmanına sahipken, VovNet57a 4 stem katmanına sahiptir. Bu, VovNet57a'nın daha derin bir model olduğunu gösterir.

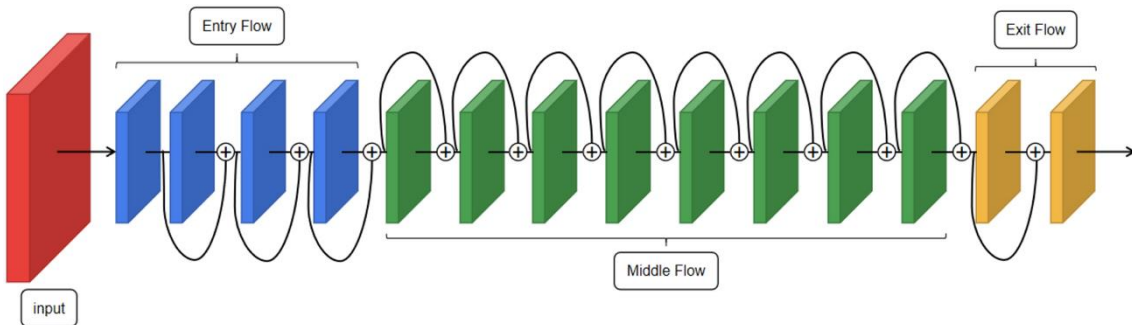
4. Hesaplama Kapasitesi: VovNet57a, VovNet39a'dan daha büyük olduğundan, daha yüksek bir hesaplama kapasitesine sahiptir.
5. Performans: VovNet39a ve VovNet57a modelleri, çeşitli bilgisayar görüşü görevlerinde yüksek performans sergilemek için tasarlanmıştır. Ancak, VovNet57a, daha büyük boyutu ve daha derin modeli nedeniyle, VovNet39a'dan daha yüksek performans gösterir.

Tablo 10'da da görüldüğü gibi VovNet57a, daha fazla katman ve parametreye sahipken VovNet39a'dan daha büyük bir modeldir. VovNet57a ayrıca daha fazla Lokal Olmayan blok ve Sıkıştırma ve Uyarma bloğu içerir. Bu nedenle VovNet57a, VovNet39a'ya göre daha yüksek bir doğruluk elde eder. Ancak, bu artan doğruluk, VovNet57a'nın daha büyük boyutu ve daha yüksek hesaplama gücü gerektirdiği anlamına gelir.

Xception

Xception, temel olarak Geleneksel Derinlemesine Ayrılabilir Evrişim (Conventional Depthwise Separable Convolution (CDSC)) adı verilen bir yapıyı kullanarak mimarisi optimize edilmiş bir evrişimli sinir ağıdır (Szegedy vd 2015; Chollet 2017). Bu yapıda, evrişimli işlem iki ayrı parçada gerçekleştirilir: önce profillendirme (depthwise convolution), sonra birleştirme (pointwise convolution). Bu yapı, geleneksel evrişimli tabanlı mimarilere benzer, ancak CDSC, katman başına daha az parametre kullanarak daha yüksek doğruluk elde etmek için tasarlanmıştır.

Xception mimarisi, bu CDSC bloklarını birbirine bağlayan bir dizi derin bloktan oluşur. Her bir blok, birkaç CDSC bloğunu içerir ve her bir CDSC bloğu, birden fazla evrişimli katmanını yerine getiren bir seyrek yapıya sahiptir. Bu yapının avantajı, parametre sayısını ve hesaplama yoğunluğunu azaltmak için gereksiz hesaplamaları ortadan kaldırmaktır. Bunun sonucunda, Xception, daha yüksek doğruluk için daha az parametre kullanır. Şekil 34'te Xception yapısı gösterilmektedir.



Şekil 34. Xception modül yapısı (Liu vd 2022).

Xception ayrıca, çıkış özellik haritalarının boyutunu azaltmak için kullanılan çift ölçekli havuzlama işlemi gibi diğer optimize edilmiş teknikler de içerir. Xception mimarisi, ImageNet veri kümesinde üstün performans göstermiş ve çeşitli görsel sınıflandırma görevlerinde de kullanılmıştır.

CNN modellerinin genel görünümü

Tablo 11'teki veriler, karşılaştırma amacıyla ImageNet (J. Deng vd 2009) veri kümesi üzerindeki sonuçlardan alınmıştır (Szegedy vd 2016; G. Huang vd 2017; Sandler vd 2018; Tan ve Le 2019).

Genel olarak, DenseNet121, DenseNet169 ve DenseNet201 modellerinde yoğun bloklar kullanılır. Yoğun bloklar, her katmandaki özellik haritalarını birleştirerek daha derin bir ağı sahip olunmasına olanak tanır.

EfficientNetv2-Small, EfficientNetv2-Medium ve EfficientNetv2-Large modelleri, önceki EfficientNet modellerine dayanarak geliştirilmiştir ve otomatik çevrimiçi hiperparametre optimizasyonu kullanılarak daha verimli bir ağ tasarlanmıştır.

Inception-v3 ve Inception-v4 modellerinde paralel yan yana çeşitli evrimsel filtreler sahip Inception modülleri kullanılır. Bu modüller, farklı ölçeklerdeki nesneleri tespit edebilen çoklu filtrelerin birleşiminden oluşur.

MobileNetv3-Small ve MobileNetv3-Large modelleri, mobil cihazlar gibi düşük kaynaklı cihazlarda çalışması için tasarlanmıştır. Derinlemesine Ayrılabilir Evrişim (Depthwise Separable Convolution (DSConv)) adı verilen bir yapı kullanarak daha hafif bir model elde edilir.

ResNet34, ResNet50 ve ResNet101 modelleri, artık (residual) blokları adı verilen blokları kullanarak ağı derinliğini artırır. Artık blokları, bir önceki katmandan gelen çıktıyı, bir sonraki katmana ekleyerek ağı derinliğini artırır.

ResNeXt34 ve ResNeXt50 modelleri, ResNet'teki artık blokları kullanırken, birçok paralel filtreye sahip evrimsel katmanları bir araya getiren ayrı bir blok yapısı kullanır.

VGG13, VGG16 ve VGG19 modellerinde birçok küçük filtreli evrimsel katmanlar kullanılır. Bu katmanlar, daha büyük filtrelerin yerine kullanılarak ağı daha derin ve daha öğrenilebilir olmasını sağlar.

Tablo 11. Modellerinin Genel Karşılaştırılması

Model	Yıl	Parametreler	Top-1 Doğruluk	Top-5 Doğruluk
VGG13	2014	133M	69.6%	89.4%
VGG16	2014	138M	71.5%	90.2%
VGG19	2014	144M	72.4%	90.8%
ResNet34	2015	21M	74.3%	91.4%
ResNet50	2015	25M	76.2%	93.1%
ResNet101	2015	44M	77.4%	93.8%
ResNeXt34	2017	22M	75.3%	92.2%
ResNeXt50	2017	26M	77.9%	93.8%
DenseNet-121	2016	8M	74.6%	91.8%
DenseNet-169	2016	14M	76.0%	93.2%
DenseNet-201	2016	20M	77.3%	93.9%
Inception-v3	2015	23M	77.9%	94.4%
Inception-v4	2016	43M	80.2%	95.3%
MobileNetv3-Small	2019	2M	67.4%	87.6%
MobileNetv3-Large	2019	5M	75.2%	92.2%
Xception	2017	22M	79.0%	94.0%
EfficientNetv2-Small	2021	9M	78.4%	94.1%
EfficientNetv2-Medium	2021	26M	81.2%	95.3%
EfficientNetv2-Large	2021	56M	82.6%	96.0%

Veri Ön İşleme ve Artırma Teknikleri

Görüntü ön işleme

Gerçek resim işlemeye başlamadan önce, görüntülerin kalitesi ve berraklığına yönelik gürültüleri azaltmak önemlidir. Bu işlem, görüntülerin çekildiği ortamda bulunan toz, su damlacıkları, gölgeler gibi gürültüleri çıkararak başlar. Bu gürültüler, sonraki analizlerde yanlış

sonuçlara neden olabilir. Bu sebeple, veri setindeki resimlerin önceden işlenmesi gereklidir (Nixon ve Aguado 2019).

Önceden işleme adımlarından biri, görüntü segmentasyonu ve özellik çıkarmadır. Görüntü segmentasyonu, görüntüdeki farklı nesne ve alanları tanımlayarak ayırıştırır. Özellik çıkarma, görüntüden anlamlı özellikleri (örneğin kenarlar, köşeler, doku) çıkararak daha kolay işlenebilir hale getirir. Bununla birlikte, bazı durumlarda, yakalanan fotoğraflar düşük kontrasta ve arka plan gürültüsüne sahip olabilir. Ayrıca, bulanık görüntüler de analizlerde yanıltıcı sonuçlara neden olabilir. Bu nedenle, görüntüler bu aşamada ön işleme tabi tutulur.

Ön işleme adımları arasında, görüntüleri netleştirmek, kontrastı artırmak, bulanıklığı azaltmak ve gürültüyü gidermek yer alır. Bu işlemler, görüntülerin daha doğru analiz edilmesini ve sonuçların daha güvenilir olmasını sağlar (Ebrahimi vd 2017; Thomas ve Raj 2021).

Görüntü yeniden boyutlandırma

Görüntü yeniden boyutlandırma, makine öğrenimi ve derin öğrenme algoritmalarının verimli çalışması için önemlidir (Brownlee 2019). Bu algoritmalar genellikle farklı boyut ve çözünürlüklerde görüntüler içeren büyük miktarda veri üzerinde çalışırlar. Ancak, algoritmalar tüm görüntülerin aynı boyutta olduğunu varsayarlar, bu nedenle görüntü yeniden boyutlandırma işlemi, tüm görüntülerin standart bir boyuta ölçeklenmesini sağlayarak bellek kullanımını ve işlem gücü ihtiyacını azaltır. Ancak, yeniden boyutlandırma işlemi, görüntü kalitesi veya çözünürlüğü kaybına neden olabilir, bu nedenle özellikle nesne tanıma görevleri için görüntü kalitesinin korunması önemlidir.

Kontrast artırma

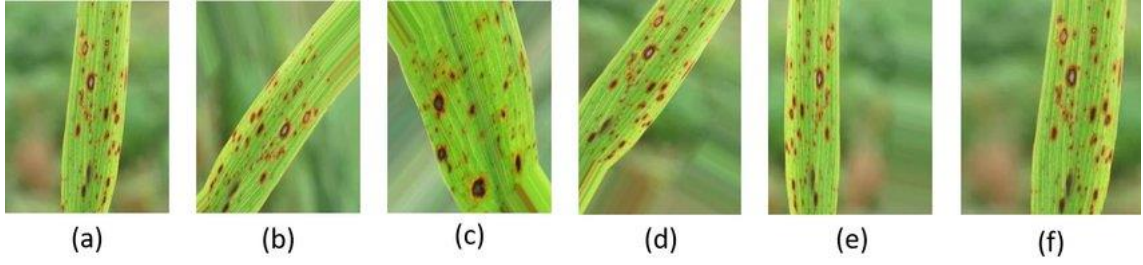
Kontrast artırma, fotoğraflardaki farklı bölümlerin daha iyi ayırt edilmesini sağlayan bir görüntü işleme adımıdır. Bu işlem, histogram eşitleme yöntemiyle gerçekleştirilir ve adaptif histogram eşitleme (Barbedo 2016)) gibi daha gelişmiş teknikler de kullanılabilir. Kontrast artırma işlemi, görüntüdeki ayrıntıların daha belirgin hale gelmesini sağlar (Athiraja ve Vijayakumar 2021) ve sonraki analizler için daha iyi bir veri seti elde edilmesine yardımcı olur.

Veri artırma

Görüntü artırma işlemleri, derin öğrenme modellerinin performansını artırmak için oldukça yaygın bir yöntemdir (Mikołajczyk ve Grochowski 2018; Shorten ve Khoshgoftaar 2019). Bu yöntem, sınırlı sayıda görüntü ile aşırı öğrenmeyi önlemek ve daha genelleme yeteneğine sahip modeller elde etmek için kullanılır (Nagaraju ve Chawla 2020).

Görüntü artırma işlemleri, özellikle görüntü sınıflandırması, nesne tespiti ve yüz algılama gibi görevler için sıklıkla tercih edilir. Bu işlemler, görüntüleri rastgele olarak çevirme, döndürme veya kırpma işlemlerini içerirler (Khirade ve Patil 2015; Nagaraju ve Chawla 2020). Bu işlemler, ayrıca görüntülerin renk tonunu, parlaklığını veya kontrastını değiştirerek farklı veri varyasyonları oluşturulmasına yardımcı olurlar.

Şekil 35'te orijinal (a), döndürülmüş (b), yakınlaştırılmış (c), kaydırılmış (d), ters çevrilmiş (e) ve ölçekli (f) görüntüler gösterilmiştir.



Şekil 35. Kahverengi lekesi hastalığına ait artırılmış görüntüler (Hossain vd 2020)

Görüntü artırma işlemleri, derin öğrenme modelleri için uygun veri kümesi boyutu elde etmek için de kullanılır. Derin öğrenme modelleri, yeterli miktarda veriye sahip olmadığında genellikle aşırı öğrenme sorunu ile karşılaşır. Ancak, görüntü artırma işlemleri ile daha fazla varyasyonlu veri oluşturularak, bu sorun önlenir.

Veriler toplanırken görüntü ön işlem yapılarak verideki siyah köşeler, kenarlar çıkartılır. Ama aldığımız Paddy Doctor veri setinde bu işlemler otomatik olarak geldiği için bu veriler üzerinde herhangi bir işlem yapılmadan veriler doğrudan kullanılmıştır.

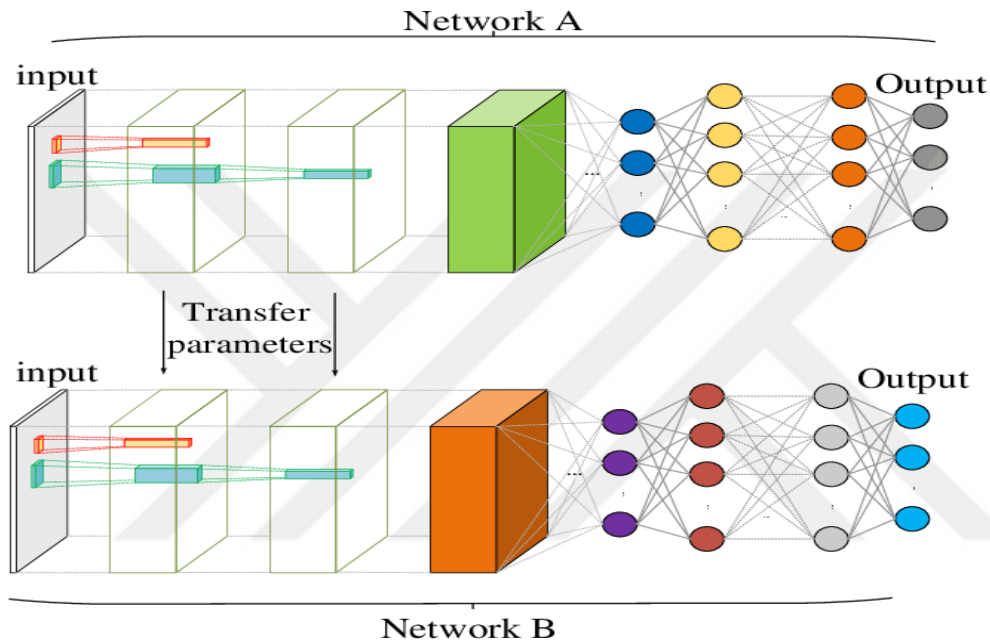
Öğrenme Aktarımı

Öğrenme aktarımı (Transfer Learning), bir önceki eğitilmiş modelin performansını geliştirmek için kullanılan bir makine öğrenmesi tekniğidir (Pacal 2022). Geleneksel olarak, makine öğrenmesi modelleri bağımsız olarak çalışacak şekilde tasarlanır ve yeni özellikler ve veriler sunulduğunda yeniden oluşturmak için büyük çaba gerektirir. Öğrenme aktarımı, edinilen bilgi ve modellerin yeniden kullanımıyla bu çabayı azaltmayı ve böylece geliştirme süresini kısaltmayı ve performansı artırmayı hedefler. Şekil 36, A ve B ağlarındaki ilk katmanların aynı olduğu transfer öğrenme kavramının gösterimidir.

Öğrenme aktarımı, yazılım hata tahmini (Nam ve Kim 2015), duygu sınıflandırması (C. Wang ve Mahadevan 2011) ve aktivite tanıma (Cook vd 2013) gibi çeşitli alanlarda uygulanmaktadır. Özellikle, bilgisayar görüşü alanında, önceden eğitilmiş ImageNet veri kümesindeki modellerin kullanılması yaygın bir uygulamadır. ImageNet (J. Deng vd 2009),

görsel nesne tanıma araştırmaları için kullanılan geniş bir resim koleksiyonudur ve birçok derin öğrenme modeli, ImageNet'te eğitildikten sonra görüntü sınıflandırma alanında en iyi sonuçları elde etmiştir.

Araştırmacılar, benzer görsel özelliklere sahip küçük veri kümeleri için önceden eğitilmiş ImageNet modellerini kullanarak zaman ve kaynak tasarrufu yaparken daha iyi sonuçlar elde edebilirler (Kılıçarslan 2022). Önceden eğitilmiş ImageNet modellerini kullanmak için araştırmacılar, modeli yeni bir veri kümesinde ince ayarlayabilir veya özellik çıkarıcısı olarak kullanabilirler.



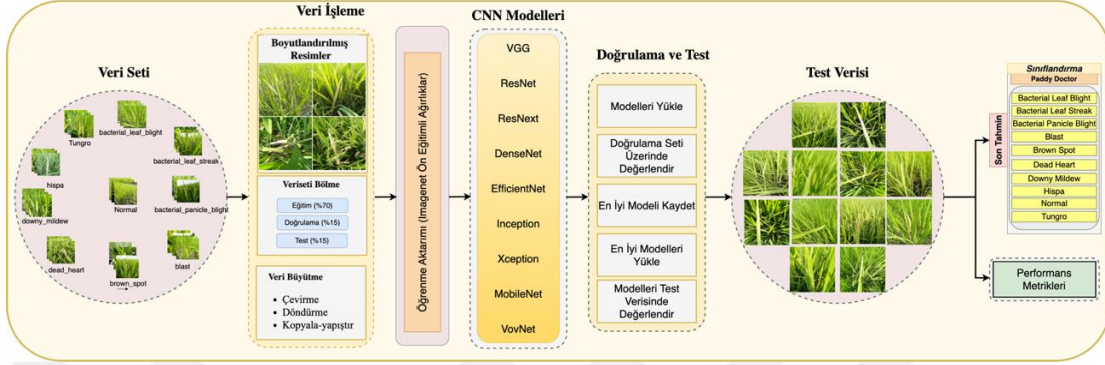
Şekil 36. Öğrenme aktarımı (Lemley vd 2017).

Öğrenme aktarımı, birçok avantaj sağlar. Öncelikle, öğrenme süresi ve veri toplama maliyetleri azalır. İkincisi, daha az sayıda eğitim örneği ile daha iyi sonuçlar elde edilebilir. Üçüncüsü, farklı veri kümelerinde önceden eğitilmiş modeller kullanılarak, genel özelliklerin öğrenilmesi ve daha iyi genelleme yapılması sağlanabilir (Pan ve Yang 2010; Torrey ve Shavlik 2010; Cook vd 2013).

Ancak, öğrenme aktarımı da bazı zorluklarla karşı karşıyadır. Öncelikle, önceden eğitilmiş modellerin farklı veri kümeleri üzerindeki performansları değişebilir. İkincisi, veri kümelerindeki farklı dağılımlar, öğrenme aktarımını zorlaştırabilir. Üçüncüsü, önceden eğitilmiş modellerin özellikleri yeni veri kümesine uygun değilse, öğrenme aktarımı etkili olmayabilir.

Önerilen Yaklaşım

Çeltik yaprak hastalıklarının tespit edilmesi ve sınıflandırılması için yapılan çalışmanın temel bileşenleri Şekil 37’de gösterilmektedir. Derin öğrenme tabanlı bu çalışmada ilk aşamayı veri seti birimi oluşturmaktadır. Veri seti, 10 hastalık sınıfına ait 10.407 görüntüden oluşmuş olup derin öğrenme yaklaşımları için uygun bir veri setidir.



Şekil 37. Çeltik yaprağı hastalığı tespiti için genel yaklaşım

Veri işleme aşaması, veri setinden sonraki ikinci adımdır. Bu aşamada, temel amaç veri setindeki görüntüleri aynı boyuta indirgemek, bölmek ve bazı temel veri artırma tekniklerini uygulamaktır. Çeltik yaprak hastalıklarının 10.407 görüntüsünden oluşan Paddy Doctor veri seti, derin öğrenme yaklaşımlarında kullanılmak üzere eğitim (%70), doğrulama (%15) ve test (%15) verileri olmak üzere rastgele üç bölüme ayrılmıştır. Daha sonra, sınıf dağılımlarını dengelemek için veri setine eğitim sırasında çevirme, döndürme ve kaydırma gibi temel veri artırma teknikleri uygulanmıştır. Veri büyütme, büyük ölçekli veri kümelerinde çok etkili olmayabilir, ancak küçük ölçekli ve sınırlı çeşitliliğe sahip veri kümelerinde oldukça etkili bir yaklaşımdır.

Üçüncü aşama öğrenme yaklaşımlarının yer aldığı kısımdır. Bu kısımda öğrenme aktarımı ve sınıflandırma için CNN modelleri kullanılmaktadır. Transfer öğrenme olarak da bilinen öğrenme aktarımı, bir alanda eğitilmiş bir modelin ağırlıklarının farklı bir alan için kullanılması sürecini ifade eder. Başka bir deyişle, ağırlıkların yeni bir alana aktarılmasını içerir. Bu çalışmada, ImageNet veri seti (Karaman vd 2022) üzerinde eğitilen CNN model ağırlıkları Paddy Doctor veri setine uygulanmıştır.

Öğrenme aktarımı, özellikle küçük ölçekli veri kümeleriyle çalışırken önemli performans avantajları sağlar. Bu veri setinde, sıfırdan eğitime kıyasla daha hızlı performans ve yakınsama elde etmek için öğrenme aktarımı kullanılmıştır. Son olarak, öğrenim aktarımının ardından en popüler 20 CNN modeli, sınıflandırma işlemi için doğrulama ve test süreçlerine tabi tutulmuştur. Her modelin performans metrikleri, test verileri üzerinde test edilerek belirlenmiştir. Son olarak, sınıflandırma işlemi için bu modellerin çıktıları gözlemlenmiştir.

ARAŞTIRMA BULGULARI

Uygulama Detayları

Bu çalışma, Linux tabanlı bir bilgisayar üzerinde Ubuntu 22.04 işletim sistemi kullanılarak yürütülmüştür. Eğitim ve test aşamalarında kullanılan bilgisayar, Intel Core i9 9900X işlemci (10 çekirdek, 3.50 GHz hız), 19.25 MB Intel Smart Cache bellek, 16GB DDR4 RAM ve tek bir NVIDIA RTX 3090 (24 GB GDDR6X, 10496 CUDA çekirdeği ve 328 tensör çekirdeği) ile donatılmıştır. Deneyler, PyTorch framework'ünün en son kararlı sürümü ile NVIDIA CUDA kullanılarak gerçekleştirilmiştir. Tüm mimariler için aynı ortam kullanılmış ve tüm modeller aynı parametrelerle eğitilmiştir.

Değerlendirme Metrikleri

Derin öğrenme modellerinin etkililiğini ölçmek ve bilinçli kararlar vermek için performans metrikleri oldukça önemlidir. Bu metrikler, bir modelin performansını değerlendirmek, optimizasyon sürecini yönlendirmek, sonuçları raporlamak, hataları ve önyargıları belirlemek, referans noktaları oluşturmak ve aşırı uyum tespit etmek gibi görevler için kullanılır.

Bu çalışmada, özellikle çeltik hastalıklarının tanımlanmasında etkili olan yaygın olarak kullanılan metrikler incelenmiştir. Derin öğrenmede yaygın olarak kullanılan metrikler arasında doğruluk, kesinlik, duyarlılık ve F1-Skor değeri bulunmaktadır.

Doğruluk, doğru yapılan tahminlerin tüm tahminlere oranını temsil eder ve genel bir model performansını gösterir. Bir modelin genel doğruluğunu değerlendirmek için bu metrik kullanılır.

Kesinlik, modelin tüm pozitif tahminlerine oranla başarılı tahminlerin oranını belirtir. Bu metrik, bir modelin ne kadar doğru pozitif tahmin yaptığını ölçmek için kullanılır.

Duyarlılık, doğru pozitif tahminlerin tüm pozitif tahminlerine oranını ölçer ve modelin kaç gerçek pozitif örneğin doğru bir şekilde tanımlandığını gösterir. Bu metrik, bir modelin ne kadar iyi bir şekilde gerçek pozitif örnekleri tespit ettiğini ölçmek için kullanılır.

F1-Skor değeri, kesinlik ve duyarlılık arasında bir denge kurmak için kullanılan harmonik ortalamadır. Hem kesinlik hem de duyarlılığı dikkate alarak bir modelin performansını ölçmek için bu metrik kullanılır.

Bu çalışmada, derin öğrenme modellerinin etkililiğini değerlendirmek için yaygın olarak kullanılan metrikler incelenmiştir. Bu metrikler, modellerin performansını değerlendirmek, optimizasyon sürecini yönlendirmek, sonuçları raporlamak, hataları ve

önyargıları belirlemek, referans noktaları oluşturmak ve aşırı uyum tespit etmek gibi görevler için önemlidir.

Tüm metrikler aşağıdaki denklemlerle verilen matematiksel formüllere sahiptirler:

$$\text{Doğruluk} = \frac{\text{Doğru tahmin sayısı}}{\text{Toplam tahmin sayısı}} \quad (4)$$

$$\text{Kesinlik} = \frac{\text{Doğru pozitif}}{\text{Doğru pozitif} + \text{Yanlış pozitif}} \quad (5)$$

$$\text{Duyarlılık} = \frac{\text{Doğru pozitif}}{\text{Doğru pozitif} + \text{Yanlış negatif}} \quad (6)$$

$$F1 - \text{skor} = \frac{2 * \text{Kesinlik} * \text{Duyarlılık}}{\text{Kesinlik} + \text{Duyarlılık}} \quad (7)$$

Deneysel Sonuçlar

Derin öğrenme modellerinin hız ve doğruluğunu artırmak için birçok teknik ve parametre kullanılabilir. Öğrenme aktarımı ve veri artırma, performansı en çok artıran teknikler arasındadır. Öğrenme aktarımı, önceden eğitilmiş bir modelin ağırlıklarını yeni bir modelde kullanarak eğitim zamanını azaltır ve performansı artırır. Veri artırma ise, öğrenme veri setini çeşitlendirmek ve modelin genelleştirme yeteneğini artırmak için kullanılır. Veri artırma teknikleri arasında, ölçeklendirme, yumuşatma, karıştırma, renk değiştirme ve yansıtma bulunur.

Diğer parametreler arasında giriş görüntüsü boyutu, küme boyutu, epok sayısı, optimizasyon algoritması, öğrenme oranı, momentum vb. bulunur. Giriş görüntüsü boyutu, modelin verimliliğini etkiler. Küme boyutu, eğitim süresini ve bellek kullanımını etkiler. Epok sayısı, modelin ne kadar süreyle eğitildiğini belirtir. Optimizasyon algoritması, modelin güncellenmesinde kullanılan matematiksel yöntemdir. Öğrenme oranı, modelin ne kadar hızlı öğrendiğini belirtir. Momentum ise modelin ne kadar hızlı hareket ettiğini belirtir.

Model performansı, bu gibi birçok değişken ve teknik yardımıyla artırılabilir. Ancak, bu parametrelerin seçimi, veri kümesine ve modelin amaçlarına bağlıdır. Bu nedenle, bu çalışmada, tüm modeller için aynı parametreler ve teknikler kullanılmıştır. Bu yaklaşım, modellerin birbirleriyle daha nesnel bir şekilde karşılaştırılmasını sağlamaktadır. Modellerin eğitimi ve testi, Ubuntu 22.04 işletim sistemiyle çalışan bir Linux bilgisayarda gerçekleştirilmiş ve NVIDIA CUDA ile PyTorch framework'ünün en son kararlı sürümü kullanılmıştır. Tüm mimariler için eğitim ve test için aynı ortamlar kullanılmış ve tüm modeller aynı parametrelerle eğitilmiştir.

Bulgular

Bu bölüm, CNN'nin deneysel sonuçlarını inceleyerek performans ölçütlerini ve karışıklık matrisinin sonuçlarını tartışır. Ayrıca, bu modellerin Paddy Doctor veri kümesi üzerindeki performansını değerlendirir ve hangi modelin diğerlerine göre daha üstün olduğunu inceler.

Tablo 12. CNN Modellerinin Test Sonuçları

No	Model	Doğruluk	Kesinlik	Duyarlılık	F1-Skor
1	ResNet34	0,9725	0,9714	0,9677	0,9695
2	ResNet50	0,9680	0,9683	0,9651	0,9667
3	ResNet101	0,9622	0,9612	0,9532	0,9572
4	VGG13	0,9705	0,9708	0,9666	0,9687
5	VGG16	0,9718	0,9718	0,9683	0,9700
6	VGG19	0,9654	0,9666	0,9604	0,9635
7	DenseNet-121	0,9725	0,9713	0,9657	0,9685
8	DenseNet-169	0,9725	0,9734	0,9697	0,9715
9	DenseNet-201	0,9680	0,9691	0,964	0,9665
10	Xception	0,9699	0,9702	0,9649	0,9675
11	Inception-v3	0,9577	0,9627	0,9597	0,9612
12	Inception-v4	0,9712	0,9714	0,9677	0,9695
13	ResNeXt50	0,9680	0,9712	0,9649	0,9680
14	VovNet39a	0,9731	0,9722	0,9678	0,9700
15	VovNet57a	0,9699	0,9694	0,9632	0,9663
16	MobileNetv3-Small	0,9731	0,9744	0,9676	0,9710
17	MobileNetv3-Large	0,9731	0,9752	0,9698	0,9725
18	EfficientNetv2-Small	0,9801	0,9800	0,9799	0,9799
19	EfficientNetv2-Medium	0,9737	0,9735	0,9724	0,9729
20	EfficientNetv2-Large	0,9782	0,9779	0,9764	0,9771

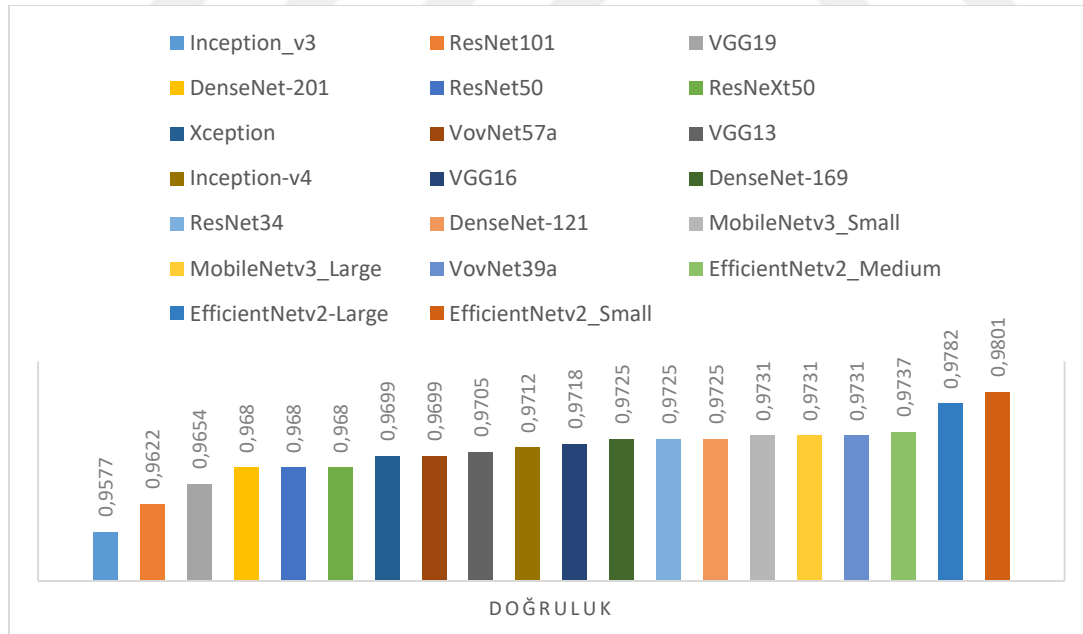
Paddy Doctor, bitki hastalıklarının teşhisinde kullanılan yeni bir veri kümesidir. CNN son zamanlarda bilgisayarlı görü alanında popüler hale gelmiştir ve özellikle tarımsal görüntü analizinde yüksek doğruluk ve az parametreler nedeniyle başarılıdır. Bu çalışmada, literatürde nesne sınıflandırması için yaygın olarak kullanılan CNN modelleri, Paddy Doctor veri setindeki çeltik hastalıklarına ait görüntüler üzerinde eğitilmiş ve değerlendirilmiştir. CNN'in parametre ve hiperparametre değerleri kullanılarak, modellerin performans analizi yapılmıştır.

Çok sayıda bitki hastalığının erken teşhisi, üretim için hayati önem taşır ve çeltik yaprağı hastalıklarının erken tespiti, çiftçilerin hastalığın yayılmasını kontrol altına almalarına ve ürün kaybını en aza indirmelerine yardımcı olabilir. Tablo 13, çeltik yaprağı görüntülerinden elde edilen Paddy Doctor veri seti kullanılarak çeltik yaprağı hastalıklarının sınıflandırılmasının deneysel sonuçlarını göstermektedir.

Tablo 12, tüm modellerin doğruluk, kesinlik, duyarlılık ve F1-Skor değerlerine ait performans metriklerini göstermektedir. Tablo 13'e bakıldığında, tüm derin öğrenme modellerinin %95'in üzerinde yüksek bir doğruluk oranına sahip olduğu görülmektedir.

Modellere ait performans metriklerinin karşılaştırmalı analizi için Şekil 38 ve Şekil 39'da her bir modele ait doğruluk ve F1-Skor değerleri görselleştirilmiştir.

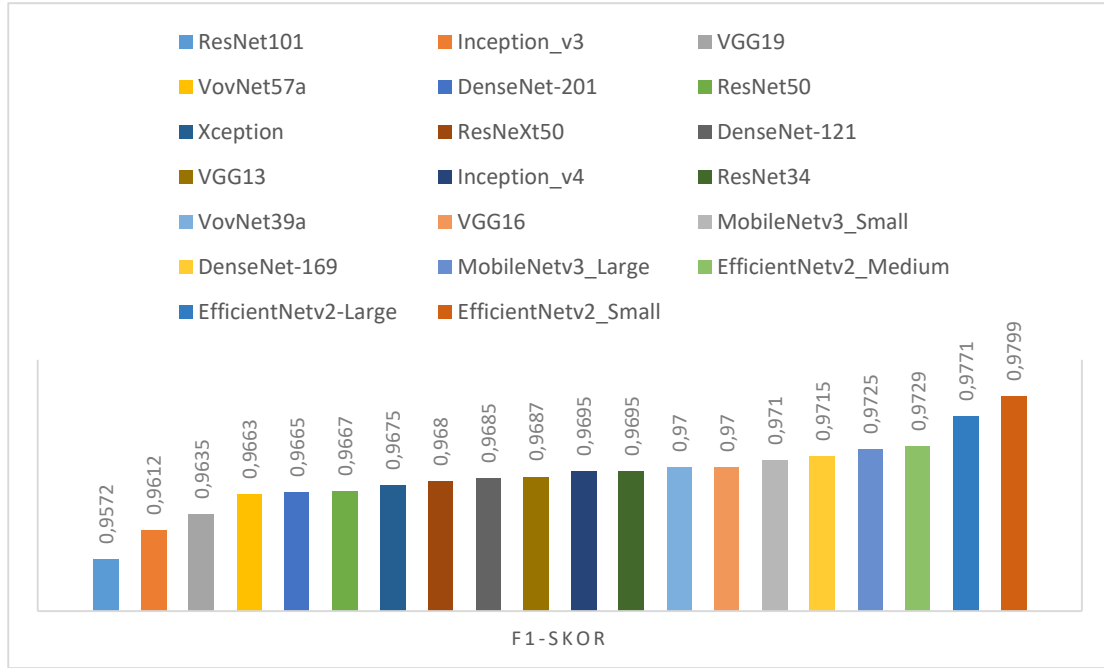
Şekil 38'deki doğruluk değerlerine bakıldığında, DenseNet-201 ResNet50 ve ResNeXt50 modelleri %96,80; Xception ve VovNet57a modelleri %96,99; ResNet34, DenseNet-121 ve DenseNet-169 modelleri %97,25; son olarak MobileNetv3_Small, MobileNetv3_Large ve VovNet39a modelleri %97,31 doğruluk değerleri ile kendi aralarında aynı sonucu vermiştir. En yüksek doğruluk performansını EfficientNetv2 modelleri göstermiştir. En yüksek doğruluk değerine %98,01 ile EfficientNetv2-Small modeli sahip olurken en düşük doğruluk değerine %95,77 ile Inception-v3 modeli sahip olmuştur.



Şekil 38. Tüm modellere ait doğruluk değerleri

Şekil 39'daki F1-Skor değerlerine bakıldığında, CNN tabanlı modeller arasında Xception ve VovNet57a modelleri %96,99 F1-Skor değerleri ile aynı sonucu vermiştir. Doğruluk değerlerinde olduğu gibi F1-Skor değerlerinde de yine en yüksek performansı EfficientNetv2 modelleri göstermiştir. En yüksek F1-Skor değerine %97,99 ile EfficientNetv2-Small modeli sahip olurken en düşük değere %95,72 ile ResNet101 modeli sahip olmuştur.

Şekil 39. Tüm modellere ait F1-Skor değerleri

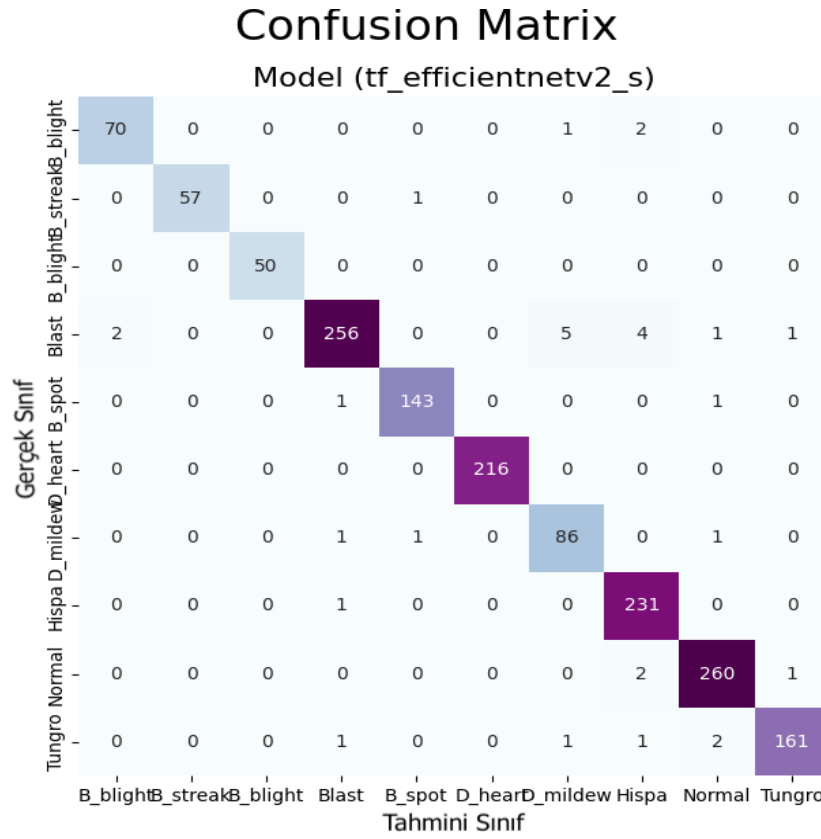


Genel olarak, doğruluk ve F1-Skor değeri göz önünde bulundurularak yapılan değerlendirmede çalışmaya dahil edilen CNN tabanlı modeller arasında en yüksek performansı EfficientNetv2-Small modeli gösterirken en düşük performansı Inception-v3 modeli göstermiştir.

Bulgulara göre, Paddy Doctor veri kümesindeki çeltik yapraklarından alınan görüntülerle yapılan teşhislerde, EfficientNetv2-Small modeli diğer modellere göre daha yüksek bir performans sergilerken, Inception-v3 modelinin performansı daha düşük seviyelerde kalmaktadır. Karışıklık matrisleri, hemen hemen tüm modellerin %96'dan daha yüksek bir doğruluk oranına sahip olduğunu göstermektedir. Çalışma sonucunda en yüksek ve en düşük performansa sahip EfficientNetV2 modeli ile Inception-v3 modelinin karışıklık matrisleri hakkında detaylı bilgi verilmiştir.

Modeller arasında en iyi sonucu veren EfficientNetV2 modeline ait Şekil 40'taki karışıklık matrisi, bu modelin 10 farklı bitki hastalığı sınıfını sınıflandırma performansını göstermektedir. Matrisin satırları, bitki hastalıklarının gerçek sınıflarını temsil ederken, sütunlar tahmin edilen sınıfları temsil etmektedir. Matrisin hücrelerindeki sayılar, her

kategoriye giren örnek sayısını gösterir. Sağlanan bilgilere dayanarak, aşağıda her hastalık sınıfı için ayrıntılı bir açıklama yer almaktadır:



Şekil 40. EfficientNetV2-Small modeline ait karışıklık matrisi

1. Bakteriyel Yaprak Yanıklığı (BLB): Model, 70 örneği BLB olarak doğru tahmin ederken, 1 örneği Tüylü Küf ve 2 örneği Hispa olarak yanlış sınıflandırdı. Bu sınıf için yanlış pozitif veya yanlış negatif yoktur.

2. Bakteriyel Yaprak Çizgisi (BLS): Model, 57 örneği BLS olarak doğru tahmin ederken, bu sınıf için yanlış pozitif veya yanlış negatif olmamıştır.

3. Bakteriyel Salkım Yanıklığı (BPB): Model, 50 örneği doğru bir şekilde BPB olarak tahmin ederken, bu sınıf için yanlış pozitif veya yanlış negatif olmamıştır.

4. Blast: Model, 256 örneği Blast olarak doğru tahmin ederken, 2 örneği Dead Heart, 5 örneği Bakteriyel Yaprak Yanıklığı, 4 örneği Hispa, 1 örneği Normal ve 1 örneği Tungro olarak yanlış sınıflandırdı. Bu sınıf için yanlış negatif yoktur.

5. Kahverengi Leke: Model, 143 numuneyi Kahverengi Leke olarak doğru tahmin ederken, 1 numuneyi Bakteriyel Yaprak Yanıklığı ve 1 numuneyi Tungro olarak yanlış sınıflandırdı. Bu sınıf için yanlış pozitif veya yanlış negatif yoktur.

6. Ölü Kalp: Model, 216 örneği Ölü Kalp olarak doğru bir şekilde tahmin ederken, bu sınıf için yanlış pozitif veya yanlış negatif yoktur.

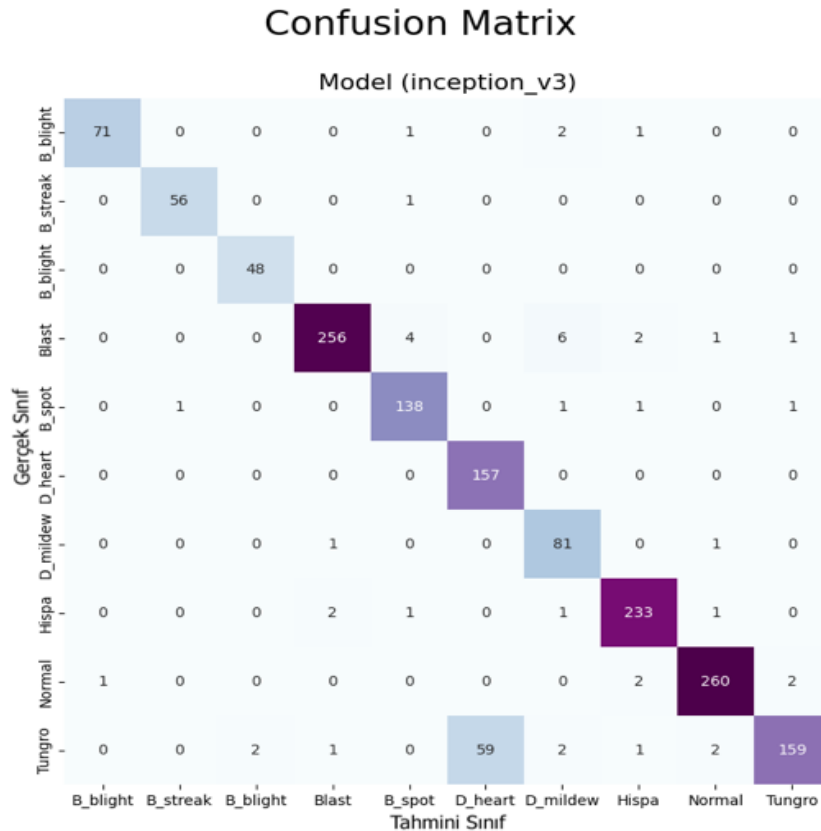
7. Tüylü Küf: Model, 86 örneği Tüylü Mildew olarak doğru tahmin ederken, 1 örneği Bakteriye Yaprak Yanıklığı ve 1 örneği Normal olarak yanlış sınıflandırmıştır. Bu sınıf için yanlış pozitif veya yanlış negatif yoktur.

8. Hispa: Model, 231 numuneyi Hispa olarak doğru tahmin ederken, 2 numuneyi Bakteriye Yaprak Yanıklığı ve 1 numuneyi Tungro olarak yanlış sınıflandırmıştır. Bu sınıf için yanlış pozitif veya yanlış negatif olmamıştır.

9. Normal: Model, 260 numuneyi Normal olarak doğru tahmin ederken, 1 numuneyi Bakteriye Yaprak Yanıklığı ve 1 numuneyi Tungro olarak yanlış sınıflandırmıştır. Bu sınıf için yanlış pozitif veya yanlış negatif yoktur.

10. Tungro: Model, 161 örneği Tungro olarak doğru tahmin ederken, 1 örneği Blast ve 2 örneği Hispa olarak yanlış sınıflandırmıştır. Bu sınıf için yanlış pozitif veya yanlış negatif yoktur.

Sonuç olarak, EfficientNetV2 modeli genel olarak iyi bir performans sergilemektedir ve birçok hastalığı doğru şekilde sınıflandırmaktadır. Ancak, bazı hatalar da bulunmaktadır ve modelin daha da geliştirilmesi gerekebilir.



Şekil 41. Inception-v3 modeline ait karışıklık matrisi

Modeller arasında en düşük sonucu veren Inception-v3 modeline ait Şekil 41'deki karışıklık matrisi, bu modelin 10 farklı bitki hastalığı sınıfını sınıflandırma performansını göstermektedir. Sağlanan bilgilere dayanarak, aşağıda her hastalık sınıfı için ayrıntılı bir açıklama yer almaktadır:

1. Bakteriyel Yaprak Yanıklığı (BLB): Model, 71 örneği BLB olarak doğru tahmin ederken, 1 örneği Blast olarak yanlış sınıflandırmıştır. Bu sınıf için yanlış pozitif veya yanlış negatif yoktur.

2. Bakteriyel Yaprak Çizgisi (BLS): Model, 56 örneği BLS olarak doğru bir şekilde tahmin ederken, bu sınıf için yanlış pozitif veya yanlış negatif olmamıştır.

3. Bakteriyel Başak Yanıklığı (BPB): Model, 48 numuneyi doğru bir şekilde BPB olarak tahmin ederken, bu sınıf için yanlış pozitif veya yanlış negatif olmamıştır.

4. Blast: Model, 256 numuneyi Blast olarak doğru tahmin ederken, 4 numuneyi BLS, 6 numuneyi BPB, 2 numuneyi Ölü Kalp, 1 numuneyi Tungro ve 1 numuneyi Normal olarak yanlış sınıflandırmıştır. Bu sınıf için yanlış negatif yoktur.

5. Kahverengi Leke: Model, 138 numuneyi Kahverengi Leke olarak doğru tahmin ederken, 1 numuneyi Ölü Kalp ve 1 numuneyi Tungro olarak yanlış sınıflandırmıştır. Bu sınıf için yanlış pozitif veya yanlış negatif yoktur.

6. Ölü Kalp: Model, 157 numuneyi Ölü Kalp olarak doğru tahmin ederken, 1 numuneyi Bakteriyel Salkım Yanıklığı ve 1 numuneyi Kahverengi Leke olarak yanlış sınıflandırmıştır. Bu sınıf için yanlış pozitif veya yanlış negatif yoktur.

7. Tüylü Küf: Model, 81 örneği Tüylü Küf olarak doğru tahmin ederken, 1 örneği Hispa ve 1 örneği Normal olarak yanlış sınıflandırmıştır. Bu sınıf için yanlış pozitif veya yanlış negatif yoktur.

8. Hispa: Model, 233 örneği Hispa olarak doğru tahmin ederken, 2 örneği Ölü Kalp ve 1 örneği Tungro olarak yanlış sınıflandırmıştır. Orada bu sınıf için yanlış pozitif veya yanlış negatif olmamıştır.

9. Normal: Model, 260 örneği Normal olarak doğru tahmin ederken, 1 örneği Blast ve 2 örneği Tungro olarak yanlış sınıflandırdı. Bu sınıf için yanlış pozitif veya yanlış negatif yoktur.

10. Tungro: Model, 159 numuneyi Tungro olarak doğru tahmin ederken, 2 numuneyi Kahverengi Leke ve 1 numuneyi Hispa olarak yanlış sınıflandırmıştır. Bu sınıf için yanlış pozitif veya yanlış negatif yoktur.

Özetle, Inception-v3 modeli için verilen karışıklı matrisi, farklı hastalık sınıfları için farklı performans sonuçları göstermektedir. Bazı sınıflar doğru bir şekilde tahmin edilirken, diğer sınıflar diğer sınıflarla karıştırılmıştır. Modelin performansını iyileştirmek için, özellikle Ölü Kalp (Dead Heart) ve Bakteriyel Yaprak Lekesi sınıfları için daha fazla veri toplanması ve modelin bu sınıfları daha iyi öğrenmesi gerekebilir.

İki model için verilen karışıklık matrislerine dayanarak, performansları arasında birkaç karşılaştırma yapabiliriz:

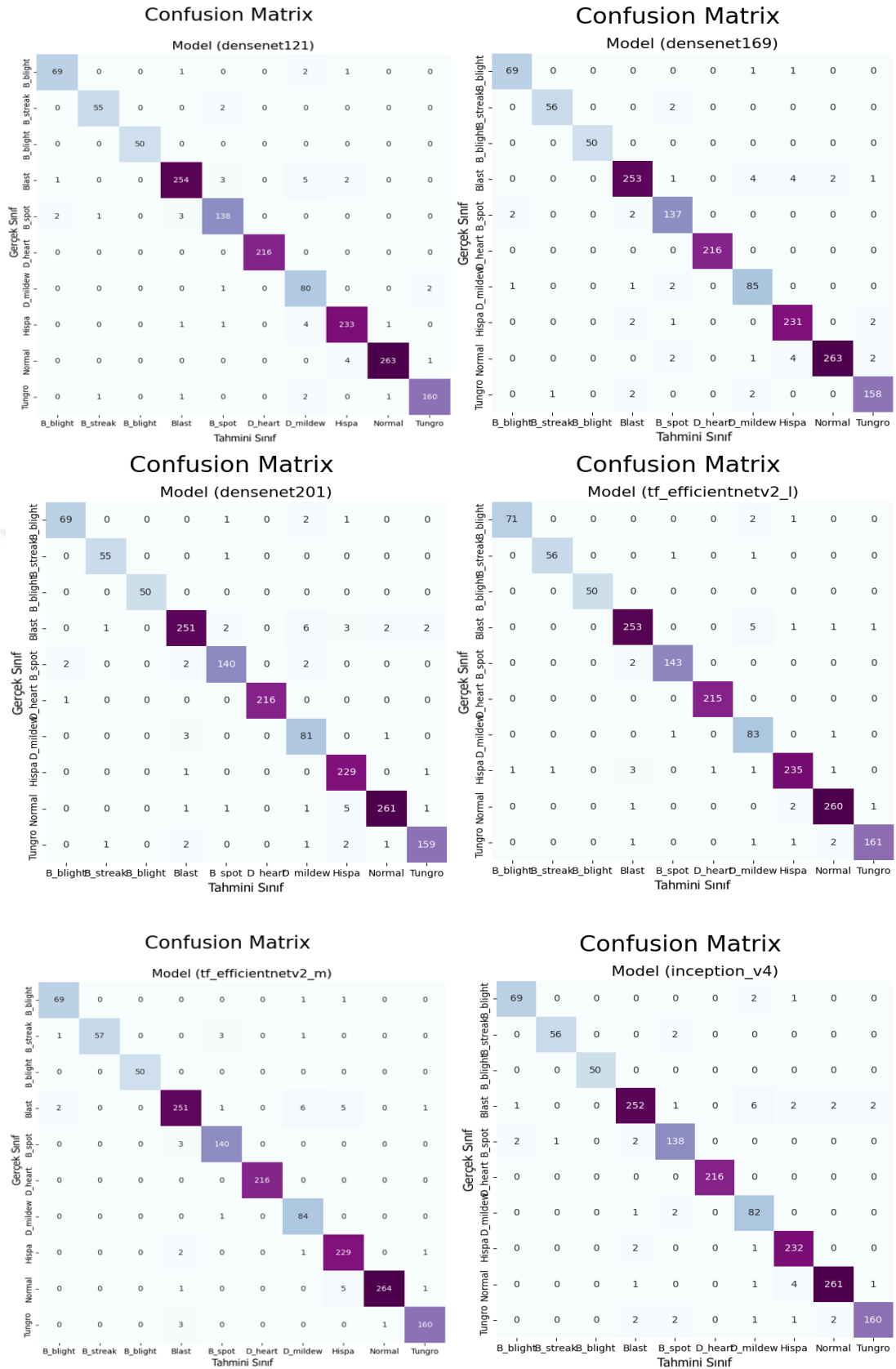
1. Genel Doğruluk: Her iki matrisin çaprazındaki doğru tahminlerin yüksek sayıya sahip olması bu iki modelin de yüksek genel doğruluğa sahip olduğunu gösterir. Ancak, EfficientNetV2 modeline ait matristeki doğru tahmin sayılarının daha büyük olması, bu modelin Inception_v3 modelinden biraz daha yüksek doğruluğa sahip olduğunu göstermektedir.

2. Yanlış Sınıflandırmalar: Her matrisin çaprazında olmayan hücrelerdeki sıfır olmayan değerlerin varlığı her iki modelin de bazı yanlış sınıflandırmalar yaptığını göstermektedir. Ancak, Inception_v3 modelinin matrisinde daha fazla yanlış sınıflandırma olduğu görülmektedir. Çünkü matrisindeki sıfır olmayan değerlerin sayısı daha fazladır.

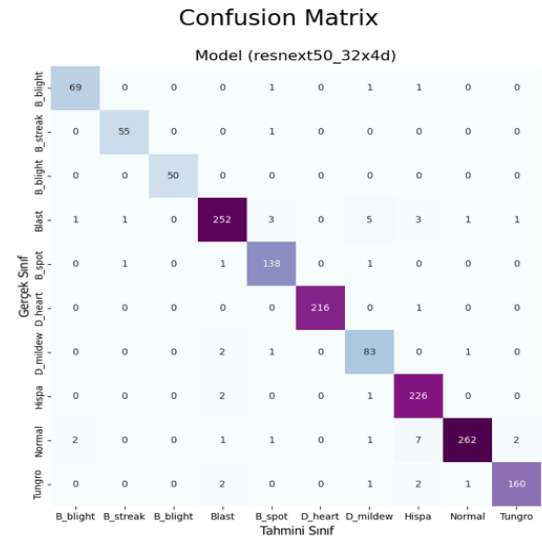
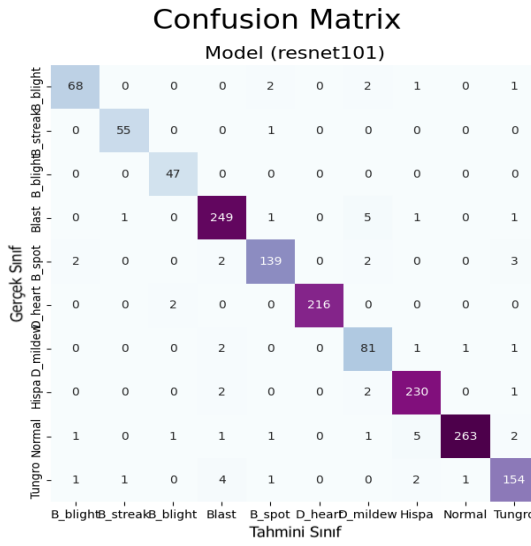
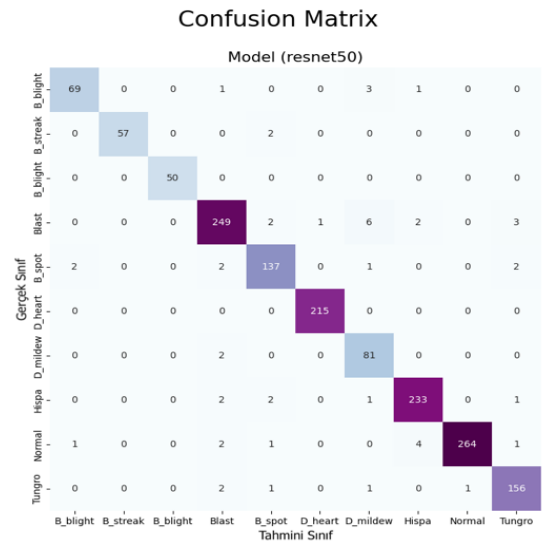
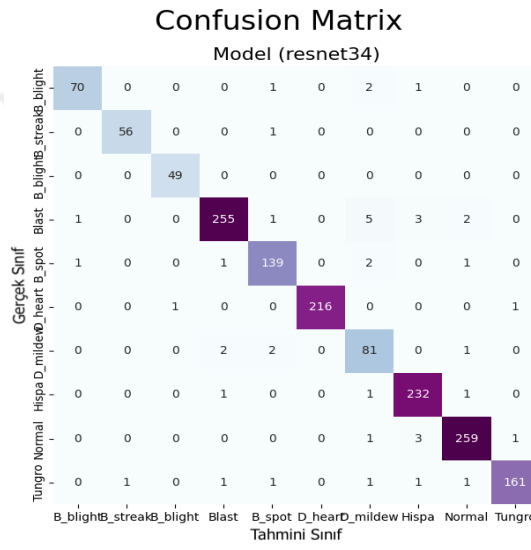
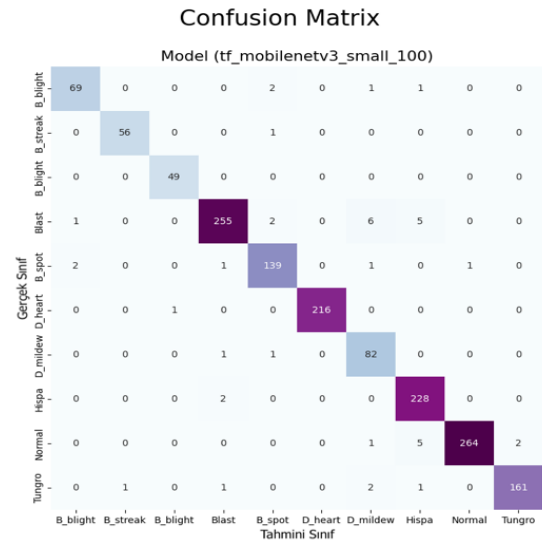
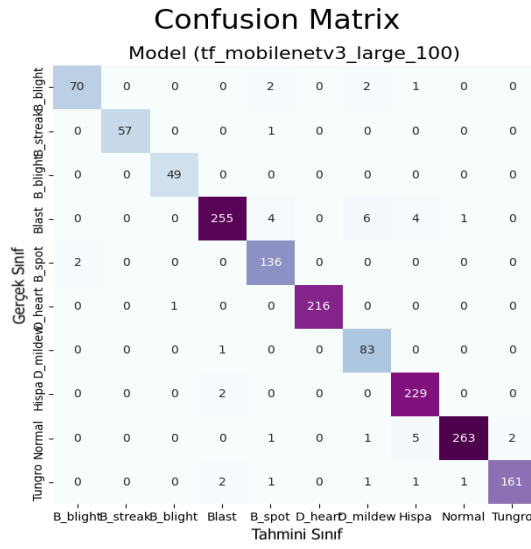
3. Belirli Sınıflar: Bazı hastalık sınıfları, her iki model için diğerlerinden daha zor görünmektedir. Örneğin, Blast sınıfı, her iki matriste de göreceli olarak yüksek sayıda yanlış sınıflandırma içermekte ve diğer hastalıklardan ayırt etmek zor olabileceğini düşündürmektedir. Öte yandan, Normal ve Kahverengi Leke sınıfları, her iki matriste de göreceli olarak az yanlış sınıflandırmaya sahiptir ve diğer hastalıklardan ayırt etmek daha kolay olabilir.

Genel olarak, yüksek genel doğruluk ve nispeten az yanlış sınıflandırmalarla birlikte her iki model de bitki hastalıklarını sınıflandırmada etkili görünmektedir. Ancak, EfficientNetV2 modeli, biraz daha yüksek doğruluk ve daha az yanlış sınıflandırma ile Inception-v3 modelinden daha etkili bir model olduğunu göstermektedir.

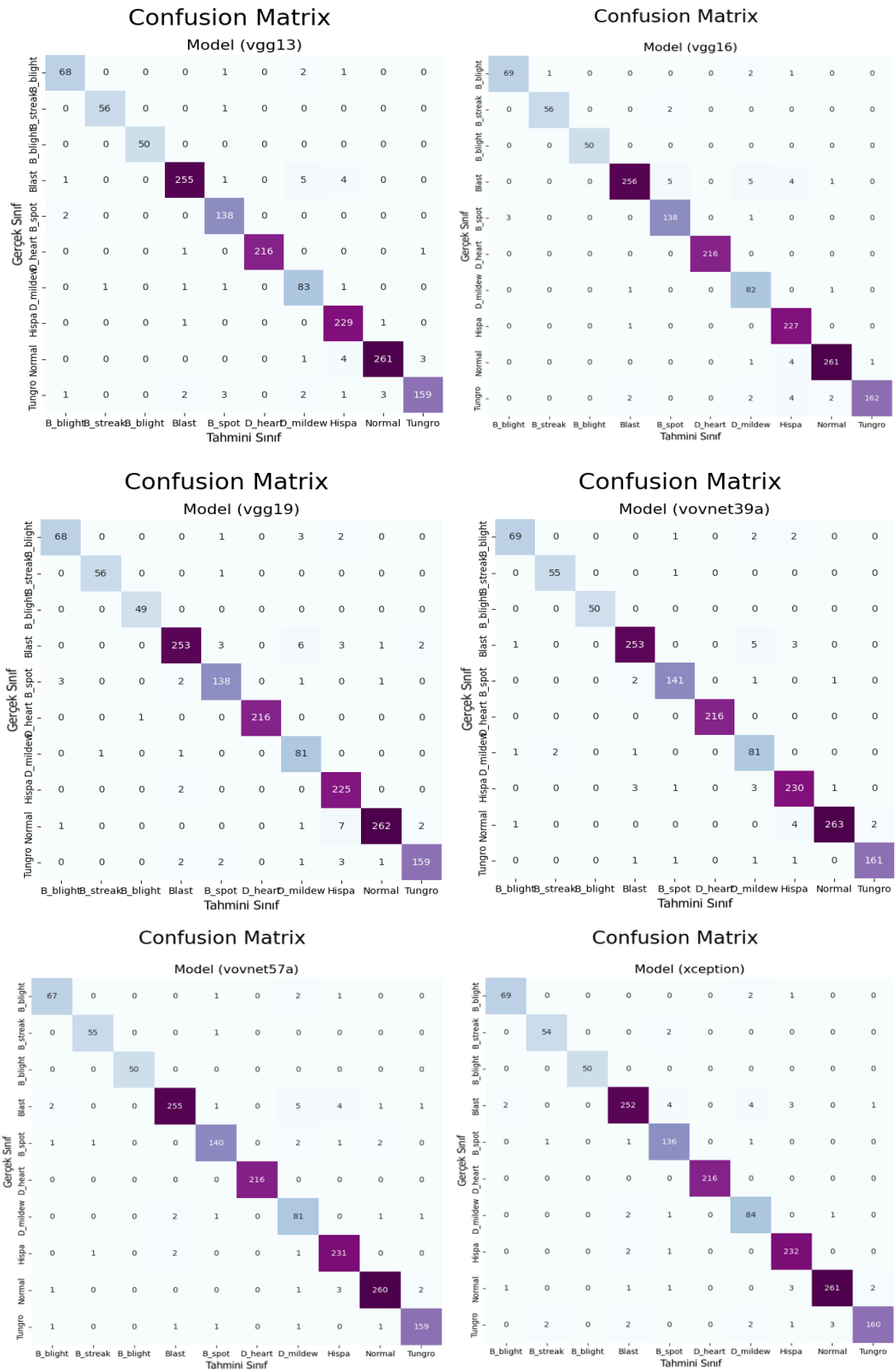
Şekil 42, Şekil 43 ve Şekil 44, çeltik yaprak hastalıklarının tespiti için kullanılan diğer CNN modellerinin karışıklık matrislerini göstermektedir.



Şekil 42. Modellere ait karışıklık matrisleri (1)



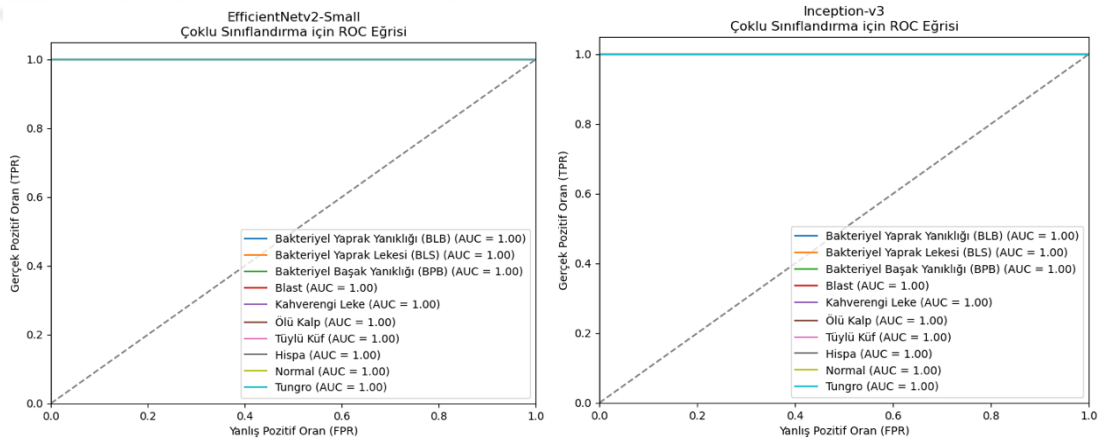
Şekil 43. Modellere ait karışıklık matrisleri (2)



Şekil 44. Modellere ait karışıklık matrisleri (3)

ROC eğrisi, sınıflandırma modellerinin performansını değerlendirmek için kullanılan bir metriktir. Bu eğri, bir sınıflandırıcının doğruluğunu ve yanlış pozitif oranını (false positive rate) gösterir. Yanlış pozitif oran, gerçekte negatif olan ancak model tarafından pozitif olarak tahmin edilen örneklerin oranıdır. ROC eğrisi, sınıflandırıcının hassasiyeti ve özgülüğü arasındaki ilişkiyi gösterir ve bu nedenle sınıflandırıcının performansını değerlendirmede kullanılır.

Tüm CNN modelleri %95'in üzerinde yüksek performans gösterdikleri için bu modellere ait ROC eğrilerine ait AUC (Eğri Altındaki Alan) değeri 1'e eşit olmaktadır. Bu yüzden 10 hastalık sınıfına ait sınıflandırma işleminde sadece en yüksek ve en düşük performansa sahip olan EfficientNetv2-Small ile Inception-v3 modelleri için ROC eğrileri çizilmiştir. Her iki modele ait ROC eğrileri Şekil 45'te gösterilmiştir.



Şekil 45. EfficientNetv2-Small ve Inception-v3 modelleri için çoklu sınıflandırma ROC eğrileri

Yapılan çalışmada her iki model de 10 sınıf için AUC=1.00 sonucu vermiştir. Bu sonuçlar, her iki modelin de yüksek performans sergilediğini göstermektedir. Çalışmaya dahil edilen diğer modeller, Inception-v3 modelinden daha yüksek performans metriklerine sahip oldukları için bu modellerin de aynı AUC değerlerine sahip olacakları sonucuna varılmıştır.

10. sınıf için elde edilen ROC eğrisi AUC=1.00 sonucu, modellerin mükemmel bir performans sergilediğini gösterir. AUC (Alan Altında Kalan Eğri) değeri, ROC eğrisinin altında kalan alanın ölçüsüdür ve 0 ile 1 arasında bir değer alır. AUC değeri 1'e ne kadar yakınsa, modellerin performansı o kadar iyidir. Bu durumda, AUC değeri 1 olduğundan, tüm modellerin neredeyse hiç yanlış sınıflandırma yapmadığı ve tüm görüntüleri %100'e yakın doğru bir şekilde sınıflandırdığı söylenebilir.

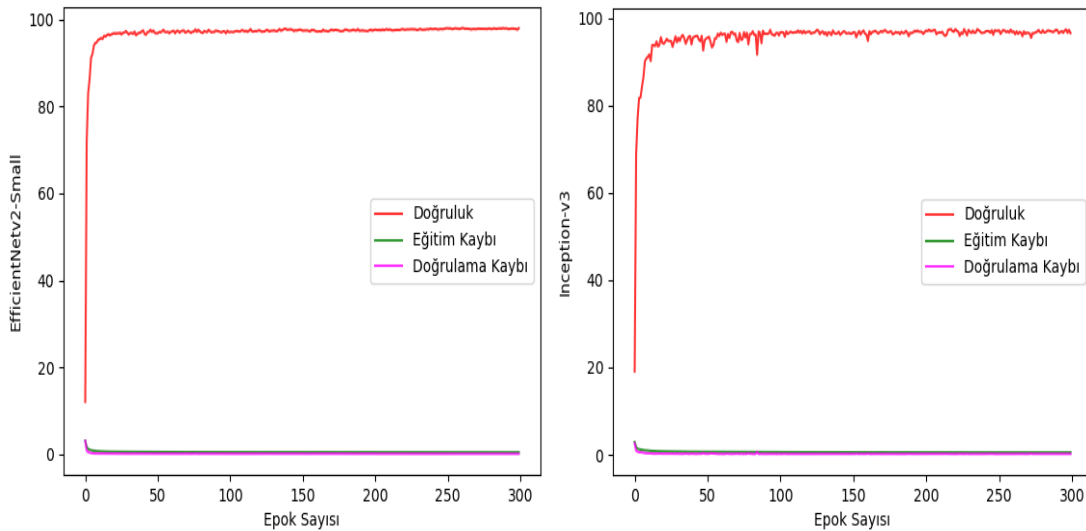
Eğitim kaybı (training loss) ve doğrulama kaybı (validation loss), makine öğrenmesi modellerinin performansını değerlendirmek için kullanılan iki önemli metriktir. Eğitim kaybı,

modelin eğitim verileri üzerinde ne kadar iyi performans gösterdiğini ölçerken, doğrulama kaybı, modelin yeni veriler üzerinde ne kadar iyi performans gösterdiğini ölçer.

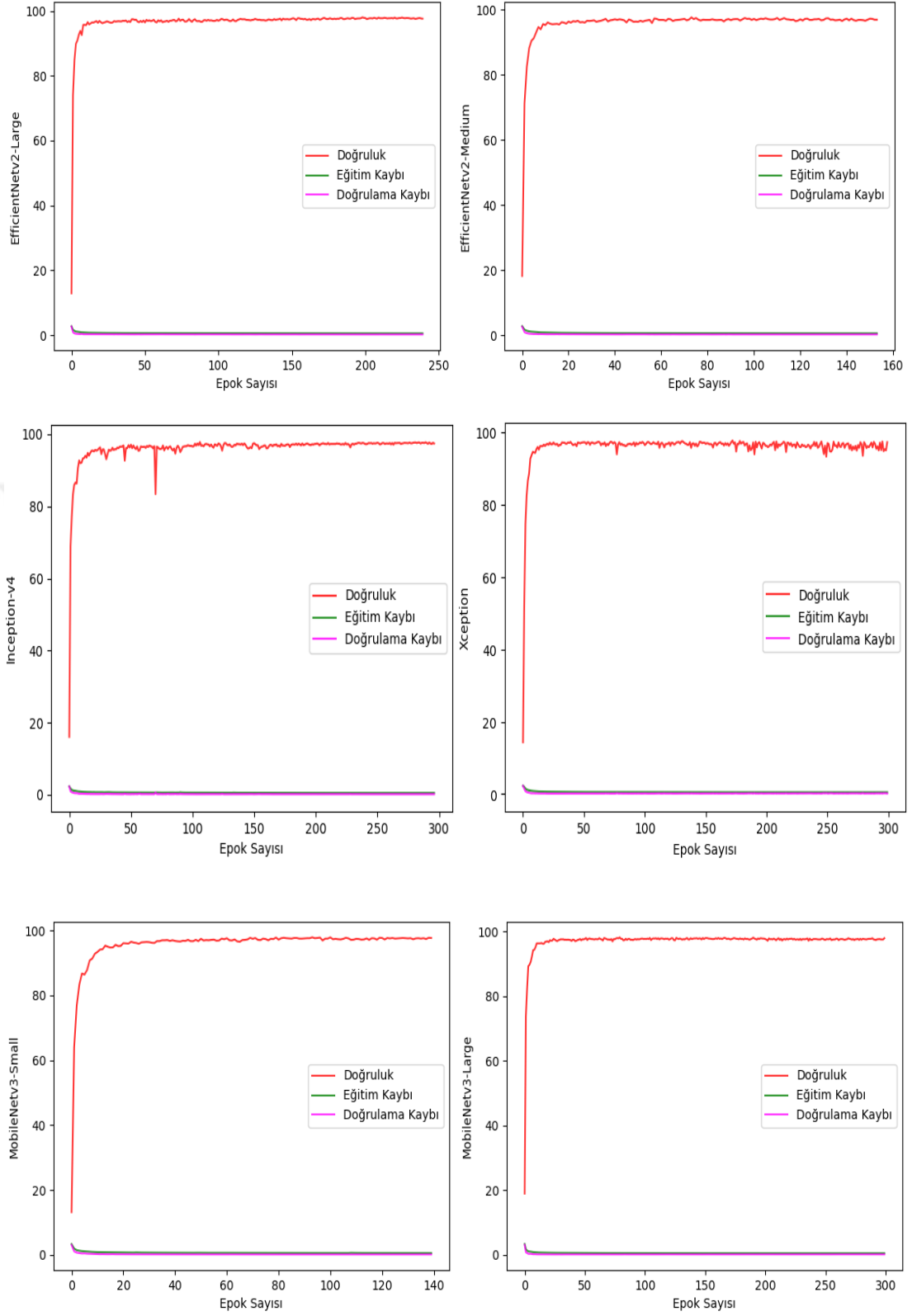
İdeal durumda, eğitim kaybı ve doğrulama kaybı birbirine yakın olmalıdır. Ancak, eğitim kaybı düşükken doğrulama kaybının yüksek olması, aşırı öğrenme (overfitting) sorununa işaret eder. Aşırı öğrenme, modelin eğitim verilerine çok iyi uyduğu ancak yeni veriler üzerinde kötü performans gösterdiği durumlarda ortaya çıkar. Bu durumda, modelin genelleme yapabilme yeteneği düşük olur.

Şekil 46, Şekil 47, Şekil 48 ve Şekil 49’da görüldüğü gibi tüm modellerin eğitim kaybı ve doğrulama kaybı değerlerinin birbirine çok yakın olması, bu modellerin aşırı öğrenme (overfitting) sorunu yaşamadığını göstermektedir. Ayrıca, eğitim kaybı ve doğrulama kaybının her ikisinin de %0'a yakın olması, modelin eğitim verilerine ve doğrulama verilerine çok iyi uyduğunu ve iyi bir performans sergilediğini göstermektedir.

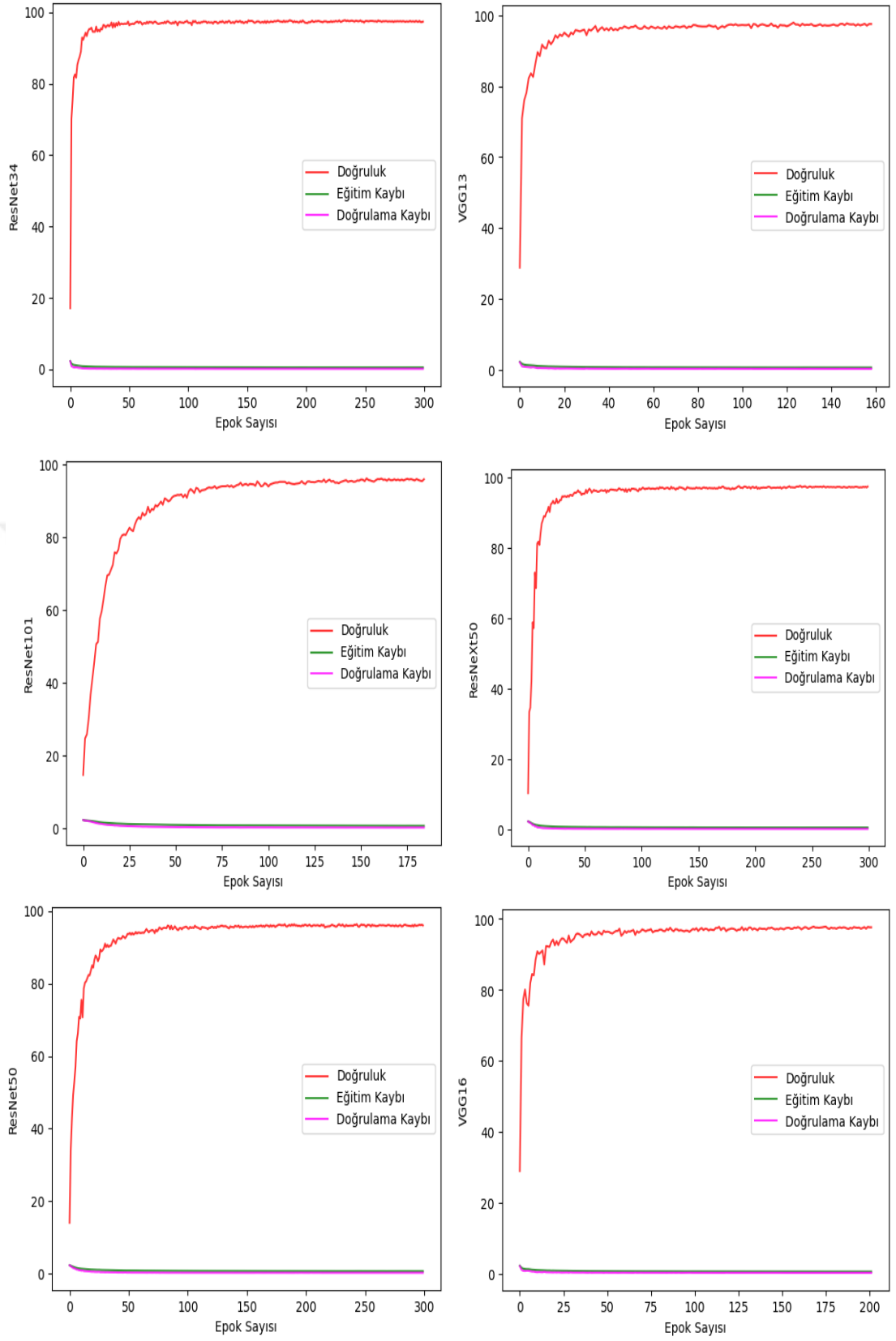
Doğruluk değerinin %100'e yakın olması ise, modellerin eğitim ve doğrulama verileri üzerinde neredeyse tüm örnekleri doğru bir şekilde sınıflandırdığını göstermektedir. Bu sonuçlar, tüm CNN modellerin çok iyi bir performans sergilediğini ve veriler üzerindeki sınıflandırma işlemini başarıyla yerine getirdiğini göstermektedir.



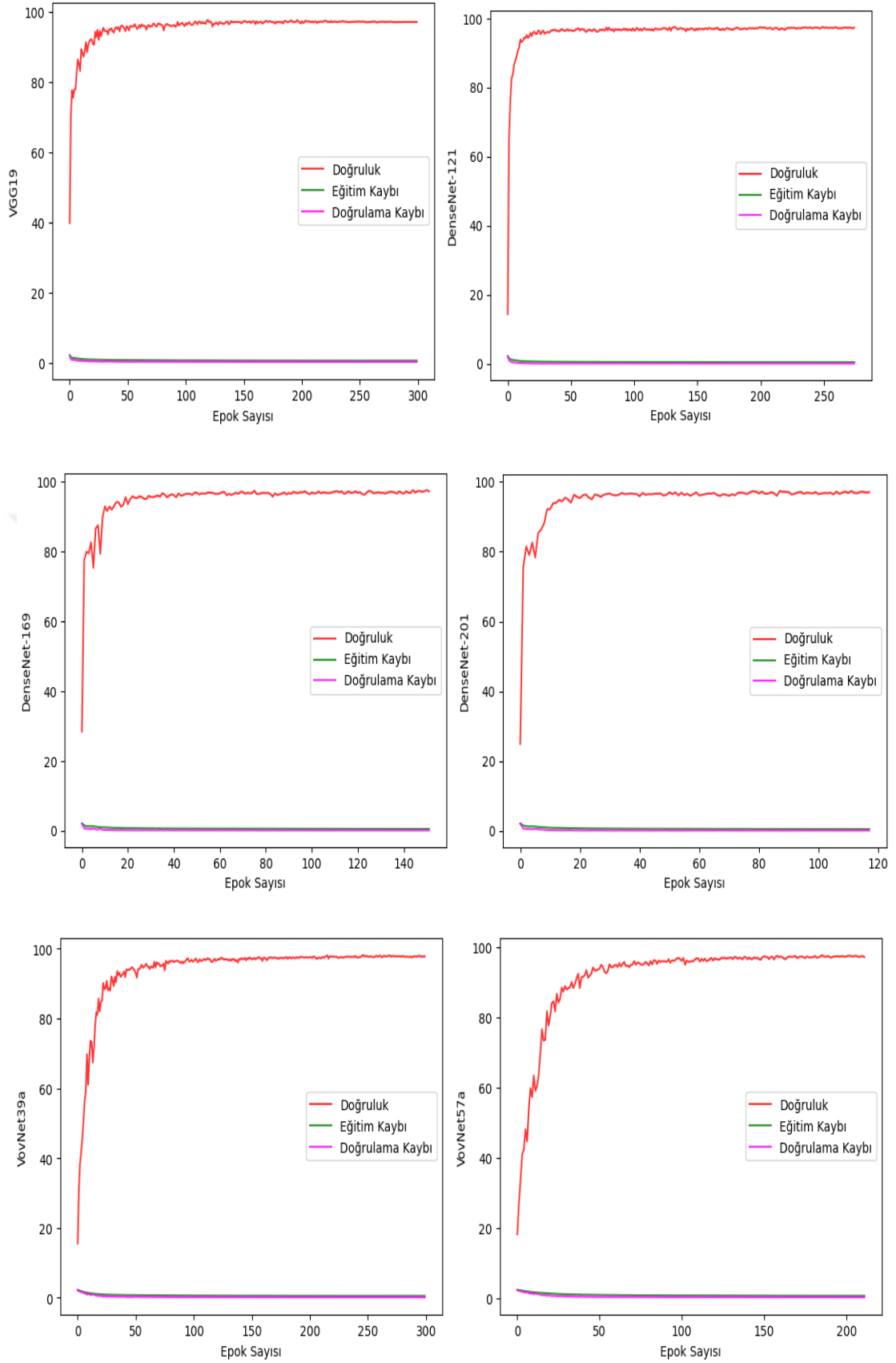
Şekil 46. Modellere ait Doğruluk, Eğitim Kaybı ve Doğrulama Kaybı grafikleri (1)



Şekil 47. Modellere ait Doğruluk, Eğitim Kaybı ve Doğrulama Kaybı grafikleri (2)



Şekil 48. Modellere ait Doğruluk, Eğitim Kaybı ve Doğrulama Kaybı grafikleri (3)



Şekil 49. Modellere ait Doğruluk, Eğitim Kaybı ve Doğrulama Kaybı grafikleri (4)

Görüntü analizinde en popüler yöntemlerden biri olan derin öğrenme, farklı ağ mimarileri kullanarak özellikle tarım görüntüleri alanında yaygın olarak kullanılmaktadır. Derin öğrenme modelleri için yeterli veriye erişememek ve matematiksel modellerdeki bazı problemler gibi zorluklar olsa da literatürde %90'ın üzerinde başarı oranına sahip birçok çalışma, derin öğrenme algoritmalarının bu alanda ne kadar umut vaat ettiğini göstermektedir. Bu çalışmada, pirinç yaprak hastalıklarının tespiti için elde edilen yaprak görüntülerine derin öğrenme yöntemleri uygulanarak, çeltik yaprak hastalığının türü belirlenmiştir. Sonuç olarak, CNN modelleri, çeltik yaprak hastalıklarının teşhisinde yüksek bir sınıflandırma başarısı göstermektedir.

Önerilen Yaklaşımın Diğer Çalışmalarla Karşılaştırılması

Yapılan çalışma, özellikle literatürdeki çeltik görüntülerinin kullanıldığı benzer çalışmalarla kıyaslanmıştır. Önerilen yaklaşımın performans değerlendirmesi Tablo 13'te sunulmuştur. Tablo 13'te, çeltik hastalıklarının doğru teşhis edilmesi için doğruluk ve F1-Skor değerleri ölçüt olarak kullanılmıştır.

Tablo 13. Önerilen Yaklaşımın Diğer Çalışmalarla Karşılaştırılması

Çalışma	Yöntem	Görüntü Sayısı	Veri Seti	Doğruluk (%)	F1-Skor (%)
Pandi vd (2022)	DCNN +GAP	5.932	Mendeley	96.50	94.32
Sethy vd (2020a)	Mobilenetv2	5.932	Mendeley	97.96	97.96
Gautam vd (2022)	InceptionV3	1.500	Mendeley ve Kaggle	96.47	96.47
Krishnamoorthy vd (2021)	InceptionResNetV2	5.200	Kaggle & Google	95.67	95.67
Kiruba ve Arjunan (2023)	Resnet34	16.225	Paddy Doctor	97.50	97.50
Nayem vd (2023)	Özel CNN	10.407	Paddy Doctor	88.18	-
Önerilen yaklaşım	EfficientNetv2-Small	10.407	Paddy Doctor	98.01	97.99

Kiruba ve Arjunan (2023) tarafından yapılan çalışma ile Nayem vd (2023) tarafından yapılan çalışmalar dışındaki diğer ilk dört çalışma farklı veri kümeleri üzerinde gerçekleştirilmiştir. Bu çalışmalardan Pandi vd (2022) tarafından evrişimli CNN modeli ve GAP (Global Average Pooling) birleştirilerek 5.932 görüntü üzerinde yapılan çalışmada %96.50'lik bir doğruluk değeri elde edilmiştir. Geleneksel CNN mimarilerini kullanan diğer

alışmalardan Sethy vd (2020a), 5.932 grnt kullanarak %97.96; Gautam vd (2022), 1.500 grnt kullanarak %96.47; Krishnamoorthy vd (2021), 5.200 grnt kullanarak %95.67 doėruluk deėeri elde etmiřlerdir.

Kiruba ve Arjunan (2023) tarafından yapılan alışmada ėrenme aktarımı yntemleri incelenmiř ve 16.225 eltik hastalık grntsne sahip Paddy Doctor veri seti kullanılmıřtır. İlgili alışmada, ResNet34 modeli en yksek doėruluk (%97.50) ve F1-Skor (%97.50) deėerleriyle en bařarılı model olmuřtur.

Nayem vd (2023) tarafından yapılan alışmada ise Paddy Doctor (2022) veri setine ait 10.407 grnt kullanılarak farklı CNN modelleri test edilmiřtir. Bu modeller arasından zel bir CNN modeli, %88.18 doėruluk deėeri ile en yksek sonucu vermiřtir.

Yaptığımız alışmada ise EfficientNetv2-Small modeli, %98.01 doėruluk ve %97.99 F1-Skor deėerleri ile hem Paddy Doctor veri setini kullanan her iki alışmadan hem de farklı veri setlerini kullanan diėer alışmalardan daha bařarılı olmuřtur.

alışmamızın daha efektif, genelleme yeteneėinin daha iyi olmasını saėlamak iin veri setini eėitim, doėrulama ve test verisi olmak zere  kısıma blerek geleneksel CNN mimarilerine uyguladık. Ayrıca ėrenme aktarımı, geliřmiř veri artırma tekniklerini uyguladık. Aynı zamanda modelleri ince ayar yaparak bařarımlarını artırmaya alıřtık.

Sonuçta nerdiğimiz yaklařım, Paddy Doctor veri setinin sınırlı verileri (10.407) olmasına raėmen eltik grntlerini temel alan diėer tm alışmalar arasından en bařarılı alışma olmuřtur.

Gelecekte, Paddy Doctor veri setindeki tm grntlerin kullanıldığı bir alışmanın daha yksek sonular vereceėi tahmin edilmektedir.

SONUÇLAR VE ÖNERİLER

Çeltik üretimi, dünya genelinde tarım sektöründe büyük bir öneme sahiptir. Ancak, çeltik üretiminde yapılan hatalar ve çeltik hastalıkları gibi faktörler, verimliliği azaltarak üretimi olumsuz etkilemektedir. Bu nedenle, çeltik hastalıklarının erken teşhisi, tarım sektöründe oldukça önemlidir. Geleneksel yöntemlerle hastalıkların tespit edilmesi oldukça zahmetli bir işlemdir ve zaman almaktadır. Ancak, son yıllarda gelişen bilgisayar destekli teşhis sistemleri, çeltik hastalıklarının tespiti ve teşhisi için oldukça etkili bir yöntem olarak kabul edilir.

Bu çalışmada, çeltik yaprak hastalıklarının tespiti için Paddy Doctor veri seti kullanılmıştır. Paddy Doctor veri seti, yaprak hastalıklarının tanımlanması için birçok yaprak görüntüsü içermektedir. Bu veri seti, çiftçiler ve ziraat mühendisleri tarafından çeltik hastalıklarının erken teşhisi için kullanılmaktadır.

Bu çalışma, literatürdeki çeltik hastalıkları üzerine yapılmış en geniş çalışmadır. Birçok farklı derin öğrenme yaklaşımı kullanılarak, çeltik yaprak hastalıklarının tespiti için birçok model oluşturulmuştur. Son zamanların en popüler derin öğrenme mimarilerinin birçok modelini bir araya getiren bu çalışma, çok yüksek bir test doğruluğu elde etmiştir.

Deneysel sonuçlar, EfficientNetv2-Small modelinin tüm çeltik hastalıklarını %98.01 test doğruluğu ve %97.99 F1-Skor değeri ile doğru bir şekilde teşhis ettiğini göstermiştir. Bu sonuç, çeltik hastalıklarının erken teşhisinde bilgisayar destekli teşhis sistemlerinin etkinliğini göstermektedir.

Ayrıca, bu çalışma, literatürdeki diğer çalışmalara kıyasla daha fazla sayıda çeltik hastalığı türünü kapsamaktadır. Bu durum yapılan çalışmanın, diğer çalışmalardan daha kapsamlı ve etkili olduğunu göstermektedir.

Bu çalışmanın sonuçları, çiftçilerin ve ziraat mühendislerinin çeltik hastalıklarının erken teşhisinde ve potansiyel olarak gelecekteki diğer tarımsal problemlerde bilgisayar destekli teşhis sistemlerinin kullanımına daha fazla güvenebileceklerini göstermektedir.

Bu çalışma, çeltik yaprak hastalıklarının tespiti için derin öğrenme yaklaşımlarının kullanımının etkisini gösteren önemli bir çalışmadır. Ancak, gelecekte bu alanda daha fazla çalışma yapılması planlanmaktadır. Bu çalışmada sadece Paddy Doctor veri seti kullanılarak gerçekleştirilmiştir. Gelecekte, farklı veri setleri de dahil edilerek daha geniş bir veri seti ile daha iyi sonuçlar elde edilmesi hedeflenmektedir. Ayrıca özellikle CNN tabanlı yaklaşımlar

üzerine odaklanılmıştır. Gelecekte, farklı derin öğrenme yöntemleri de dahil olmak üzere başka yaklaşımlar da incelenecektir.

Kullanılan veri setinde bazı sınıflardaki görüntüler fazla iken bazıları düşüktür. Bu durum veri dengesizliğine sebep olabilmektedir. Bir sonraki çalışmada veri dengesizliğini çözmek için düşük görüntülü veriler artırarak veri setindeki her sınıf için görüntü sayıları eşitlenip CNN modellerinin bu yeni veri seti üzerindeki başarımı gözlemlenecektir.

Son zamanlarda CNN kadar başarımlı sağlayan ve popüler hale gelen görüş transformatörlerine (ViT) dayalı bir çalışma yapılarak ViT modelleri arasında performans analizinin yapılması düşünülmektedir. Ayrıca çalışmanın daha da genişletilerek hem CNN modellerinin hem ViT modellerinin hem de hibrit yapılarının Paddy Doctor veri seti üzerindeki başarımlarının incelenmesi planlanmaktadır.



KAYNAKLAR

- Agarwal, Mohit, Singh, Abhishek, Arjaria, Siddhartha, Sinha, Amit, & Gupta, Suneet. 2020. ToLeD: Tomato Leaf Disease Detection using Convolution Neural Network. *Procedia Computer Science*, 167, 293-301. doi:<https://doi.org/10.1016/j.procs.2020.03.225>
- Agus, Fahrul, Ihsan, Muh, Khairina, Dyna Marisa, & Candra, Krishna Purnawan. 2019. Expert System for Rice Plant Disease Diagnosis. In: vol.
- Akila, M, & Deepan, P. 2018. Detection and classification of plant leaf diseases by using deep learning algorithm. *International Journal of Engineering Research & Technology (IJERT)*, 6(7), 1-5.
- Alshehri, Mohammed, Kumar, Manoj, Bhardwaj, Akashdeep, Mishra, Shailendra, & Gyani, Jayadev. 2021. Deep learning based approach to classify saline particles in sea water. *Water*, 13(9), 1251.
- Alzubaidi, Laith, Zhang, Jinglan, Humaidi, Amjad J, Al-Dujaili, Ayad, Duan, Ye, Al-Shamma, Omran, . . . Farhan, Laith. 2021. Review of deep learning: Concepts, CNN architectures, challenges, applications, future directions. *Journal of big data*, 8, 1-74.
- An, Changcheng, Sun, Changjiao, Li, Ningjun, Huang, Bingna, Jiang, Jiajun, Shen, Yue, . . . Wang, Chunxin. 2022. Nanomaterials and nanotechnology for the delivery of agrochemicals: strategies towards sustainable agriculture. *Journal of Nanobiotechnology*, 20(1), 1-19.
- Arnal Barbedo, Jayme Garcia. 2019. Plant disease identification from individual lesions and spots using deep learning. *Biosystems Engineering*, 180, 96-107. doi:<https://doi.org/10.1016/j.biosystemseng.2019.02.002>
- Athiraja, A, & Vijayakumar, P. 2021. RETRACTED ARTICLE: Banana disease diagnosis using computer vision and machine learning methods. *Journal of Ambient Intelligence and Humanized Computing*, 12(6), 6537-6556.
- Atole, Ronnel R, & Park, Daechul. 2018. A multiclass deep convolutional neural network classifier for detection of common rice plant anomalies. *International Journal of Advanced Computer Science and Applications*, 9(1).
- Baranwal, Saraansh, Khandelwal, Siddhant, & Arora, Anuja. 2019. *Deep learning convolutional neural network for apple leaves disease detection*. Paper presented at the Proceedings of international conference on sustainable computing in science, technology and management (SUSCOM), Amity University Rajasthan, Jaipur-India.
- Barbedo, Jayme Garcia Arnal. 2016. A review on the main challenges in automatic plant disease identification based on visible range images. *Biosystems Engineering*, 144, 52-60.
- Baresel, Jörg Peter, Rischbeck, Pablo, Hu, Yuncai, Kipp, Sebastian, Hu, Yuncai, Barmeier, Gero, . . . Schmidhalter, Urs. 2017. Use of a digital camera as alternative method for non-destructive detection of the leaf chlorophyll content and the nitrogen nutrition status in wheat. *Computers and Electronics in Agriculture*, 140, 25-33. doi:<https://doi.org/10.1016/j.compag.2017.05.032>
- Bengio, Yoshua. 2009. Learning deep architectures for AI. *Foundations and trends® in Machine Learning*, 2(1), 1-127.
- Bezdan, Timea, & Džakula, N Baćanin. 2019. *Convolutional neural network layers and architectures*. Paper presented at the Data science and digital broadcasting systems, international scientific conference on information technology and data related research, DOI.

- Bhatt, Dulari, Patel, Chirag, Talsania, Hardik, Patel, Jigar, Vaghela, Rasmika, Pandya, Sharnil, . . . Ghayvat, Hemant. 2021. CNN variants for computer vision: history, architecture, application, challenges and future scope. *Electronics*, 10(20), 2470.
- Bochkovskiy, Alexey, Wang, Chien-Yao, & Liao, Hong-Yuan Mark. 2020. Yolov4: Optimal speed and accuracy of object detection. *arXiv preprint arXiv:2004.10934*.
- Bozkurt, Ferhat, & Yağanoğlu, Mete. 2021. Derin evrişimli sinir ağları kullanarak akciğer X-Ray görüntülerinden COVID-19 tespiti. *Veri Bilimi*, 4(2), 1-8.
- Brownlee, Jason. 2019. *Deep learning for computer vision: image classification, object detection, and face recognition in python*: Machine Learning Mastery.
- Catindig, Jo. 2023. Hispa. Retrieved from <http://www.knowledgebank.irri.org/training/fact-sheets/pest-management/insects/item/rice-hispa>
- Chawathe, Sudarshan S. 2020. *Rice disease detection by image analysis*. Paper presented at the 2020 10th Annual Computing and Communication Workshop and Conference (CCWC).
- Chen, Junde, Chen, Jinxiu, Zhang, Defu, Sun, Yuandong, & Nanekaran, Y. A. 2020. Using deep transfer learning for image-based plant disease identification. *Computers and Electronics in Agriculture*, 173, 105393. doi:<https://doi.org/10.1016/j.compag.2020.105393>
- Chollet, François. 2017. *Xception: Deep learning with depthwise separable convolutions*. Paper presented at the Proceedings of the IEEE conference on computer vision and pattern recognition.
- Chowdhary, KR, & Chowdhary, KR. 2020. Natural language processing. *Fundamentals of artificial intelligence*, 603-649.
- Cilimkovic, Mirza. 2015. Neural networks and back propagation algorithm. *Institute of Technology Blanchardstown, Blanchardstown Road North Dublin*, 15(1).
- CM Vera Cruz, I Ona, NP Castilla, and R Opulencia. 2023. Bacterial Leaf Blight Retrieved from <http://www.knowledgebank.irri.org/decision-tools/rice-doctor/rice-doctor-fact-sheets/item/bacterial-blight>
- Cook, Diane, Feuz, Kyle D, & Krishnan, Narayanan C. 2013. Transfer learning for activity recognition: A survey. *Knowledge and information systems*, 36, 537-556.
- Coulibaly, Solemane, Kamsu-Foguem, Bernard, Kamissoko, Dantouma, & Traore, Daouda. 2019. Deep neural networks with transfer learning in millet crop images. *Computers in Industry*, 108, 115-120. doi:<https://doi.org/10.1016/j.compind.2019.02.003>
- Dean, Ralph, Van Kan, Jan AL, Pretorius, Zacharias A, Hammond-Kosack, Kim E, Di Pietro, Antonio, Spanu, Pietro D, . . . Ellis, Jeff. 2012. The Top 10 fungal pathogens in molecular plant pathology. *Molecular plant pathology*, 13(4), 414-430.
- Deng, Jia, Dong, Wei, Socher, Richard, Li, Li-Jia, Li, Kai, & Fei-Fei, Li. 2009. *Imagenet: A large-scale hierarchical image database*. Paper presented at the 2009 IEEE conference on computer vision and pattern recognition.
- Deng, Li, Li, Jinyu, Huang, Jui-Ting, Yao, Kaisheng, Yu, Dong, Seide, Frank, . . . Williams, Jason. 2013. *Recent advances in deep learning for speech research at Microsoft*. Paper presented at the 2013 IEEE international conference on acoustics, speech and signal processing.

- Dike, Happiness Ugochi, Zhou, Yimin, Deveerasetty, Kranthi Kumar, & Wu, Qingtian. 2018. *Unsupervised learning based on artificial neural network: A review*. Paper presented at the 2018 IEEE International Conference on Cyborg and Bionic Systems (CBS).
- Dixon, Matthew F, Halperin, Igor, & Bilokon, Paul. 2020. *Machine learning in Finance* (Vol. 1170): Springer.
- Dreizin, David, Goldmann, Florian, LeBedis, Christina, Boscak, Alexis, Dattwyler, Matthew, Bodanapally, Uttam, . . . Unberath, Mathias. 2021. An automated deep learning method for tile AO/OTA pelvic fracture severity grading from trauma whole-body CT. *Journal of Digital Imaging*, 34, 53-65.
- Du, Ke-Lin, & Swamy, Madisetti NS. 2013. *Neural networks and statistical learning*: Springer Science & Business Media.
- Duan, Kaiwen, Bai, Song, Xie, Lingxi, Qi, Honggang, Huang, Qingming, & Tian, Qi. 2019. *Centernet: Keypoint triplets for object detection*. Paper presented at the Proceedings of the IEEE/CVF international conference on computer vision.
- Ebrahimi, MA, Khoshtaghaza, Mohammad Hadi, Minaei, Saeid, & Jamshidi, Bahareh. 2017. Vision-based pest detection based on SVM classification method. *Computers and Electronics in Agriculture*, 137, 52-58.
- Fang, Xing. 2017. Understanding deep learning via backtracking and deconvolution. *Journal of big data*, 4(1), 1-14.
- Fischer, Asja, & Igel, Christian. 2012. *An introduction to restricted Boltzmann machines*. Paper presented at the Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications: 17th Iberoamerican Congress, CIARP 2012, Buenos Aires, Argentina, September 3-6, 2012. Proceedings 17.
- Fuentes, Alvaro, Yoon, Sook, Kim, Sang Cheol, & Park, Dong Sun. 2017. A robust deep-learning-based detector for real-time tomato plant diseases and pests recognition. *Sensors*, 17(9), 2022.
- Ganesan, Gangadevi, & Chinnappan, Jayakumar. 2022. Hybridization of ResNet with YOLO classifier for automated paddy leaf disease recognition: An optimized model. *Journal of Field Robotics*, 39(7), 1085-1109.
- Gautam, V., Trivedi, N. K., Singh, A., Mohamed, H. G., Noya, I. D., Kaur, P., & Goyal, N. 2022. A Transfer Learning-Based Artificial Intelligence Model for Leaf Disease Assessment. *Sustainability*, 14(20). doi:ARTN 1361010.3390/su142013610
- Gautam, Vinay. 2020. Qualitative model to enhance quality of metadata for data warehouse. *International Journal of Information Technology*, 12, 1025-1036.
- Gautam, Vinay, Trivedi, Naresh K, Singh, Aman, Mohamed, Heba G, Noya, Irene Delgado, Kaur, Preet, & Goyal, Nitin. 2022. A Transfer Learning-Based Artificial Intelligence Model for Leaf Disease Assessment. *Sustainability*, 14(20), 13610.
- Ghiasi, Golnaz, Lin, Tsung-Yi, & Le, Quoc V. 2018. Dropblock: A regularization method for convolutional networks. *Advances in neural information processing systems*, 31.
- Goluguri, NV Raja Reddy, Devi, K Suganya, & Srinivasan, P. 2021. Rice-net: an efficient artificial fish swarm optimization applied deep convolutional neural network model for identifying the Oryza sativa diseases. *Neural Computing and Applications*, 33(11), 5869-5884.
- Graves, Alex. 2013. Generating sequences with recurrent neural networks. *arXiv preprint arXiv:1308.0850*.

- Gunawan, PA, Kencana, EN, & Sari, K. 2021. *Classification of rice leaf diseases using artificial neural network*. Paper presented at the Journal of Physics: Conference Series.
- Gündüz, Mehmet Şirin, & Işık, Gültekin. 2023. A new YOLO-based method for real-time crowd detection from video and performance analysis of YOLO models. *Journal of Real-Time Image Processing*, 20(1), 5.
- Ham, Jong Hyun, Melanson, Rebecca A, & Rush, Milton C. 2011. Burkholderia glumae: next major pathogen of rice? *Molecular plant pathology*, 12(4), 329-339.
- Hastie, Trevor, Tibshirani, Robert, Friedman, Jerome, Hastie, Trevor, Tibshirani, Robert, & Friedman, Jerome. 2009. Overview of supervised learning. *The elements of statistical learning: Data mining, inference, and prediction*, 9-41.
- Hatcher, William Grant, & Yu, Wei. 2018. A survey of deep learning: Platforms, applications and emerging research trends. *IEEE Access*, 6, 24411-24432.
- He, Kaiming, Zhang, Xiangyu, Ren, Shaoqing, & Sun, Jian. 2016. *Deep residual learning for image recognition*. Paper presented at the Proceedings of the IEEE conference on computer vision and pattern recognition.
- Hinton, Geoffrey E. 2012. A practical guide to training restricted Boltzmann machines. *Neural Networks: Tricks of the Trade: Second Edition*, 599-619.
- Hinton, Geoffrey E, Osindero, Simon, & Teh, Yee-Whye. 2006. A fast learning algorithm for deep belief nets. *Neural computation*, 18(7), 1527-1554.
- Hochreiter, Sepp, & Schmidhuber, Jürgen. 1997. Long short-term memory. *Neural computation*, 9(8), 1735-1780.
- Hoffmann, Christian Hugo. 2022. Is AI intelligent? An assessment of artificial intelligence, 70 years after Turing. *Technology in Society*, 68, 101893.
- Hossain, Syed Md Minhaz, Tanjil, Md Monjur Morhsed, Ali, Mohammed Abser Bin, Islam, Mohammad Zihadul, Islam, Md Saiful, Mobassirin, Sabrina, . . . Islam, SM Riazul. 2020. *Rice leaf diseases recognition using convolutional neural networks*. Paper presented at the Advanced Data Mining and Applications: 16th International Conference, ADMA 2020, Foshan, China, November 12–14, 2020, Proceedings 16.
- Howard, Andrew G, Zhu, Menglong, Chen, Bo, Kalenichenko, Dmitry, Wang, Weijun, Weyand, Tobias, . . . Adam, Hartwig. 2017. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*.
- Htun, Khaing War, & Htwe, Chit Su. 2018. Development of Paddy Diseased Leaf Classification System Using Modified Color Conversion. *International Journal of Software & Hardware Research in Engineering*, 6(8).
- Huang, Gao, Liu, Zhuang, Van Der Maaten, Laurens, & Weinberger, Kilian Q. 2017. *Densely connected convolutional networks*. Paper presented at the Proceedings of the IEEE conference on computer vision and pattern recognition.
- Huang, Jianrong, Liao, Huaijian, Zhu, Yubo, Sun, Jiayi, Sun, Qihua, & Liu, Xiangdong. 2012. Hyperspectral detection of rice damaged by rice leaf folder (Cnaphalocrocis medinalis). *Computers and Electronics in Agriculture*, 82, 100-107. doi:<https://doi.org/10.1016/j.compag.2012.01.002>
- Islam, Taohidul, Sah, Manish, Baral, Sudipto, & Choudhury, Rudra Roy. 2018. *A faster technique on rice disease detection using image processing of affected area in agro-field*. Paper presented at the 2018 Second International Conference on Inventive Communication and Computational Technologies (ICICCT).

- Jain, Anil K, Mao, Jianchang, & Mohiuddin, K Moidin. 1996. Artificial neural networks: A tutorial. *Computer*, 29(3), 31-44.
- Jena, Kshirod K, & Kim, Suk-Man. 2010. Current status of brown planthopper (BPH) resistance and genetics. *Rice*, 3(2), 161-171.
- Kahar, Mohd Adzhar A, Mutalib, Sofianita, & Rahman, Shuzlina A. 2015. Early Detection and Classification of Paddy Diseases with Neural Networks and Fuzzy Logic Recent Advances in Mathematical and Computational Method.
- Kamal, KC, Yin, Zhendong, Wu, Mingyang, & Wu, Zhilu. 2019. Depthwise separable convolution architectures for plant disease classification. *Computers and Electronics in Agriculture*, 165, 104948.
- Kanwal, Neel. 2020. *Dilated Convolution Networks for Classification of ICD-9 based Clinical Summaries*. Politecnico di Torino,
- Karaman, Ahmet, Karaboga, Dervis, Pacal, Ishak, Akay, Bahriye, Basturk, Alper, Nalbantoglu, Ufuk, . . . Sahin, Omur. 2022. Hyper-parameter optimization of deep learning architectures using artificial bee colony (ABC) algorithm for high performance real-time automatic colorectal cancer (CRC) polyp detection. *Applied Intelligence*, 1-18.
- Karaman, Ahmet, Pacal, Ishak, Basturk, Alper, Akay, Bahriye, Nalbantoglu, Ufuk, Coskun, Seymanur, . . . Karaboga, Dervis. 2023. Robust real-time polyp detection system design based on YOLO algorithms by optimizing activation functions and hyper-parameters with artificial bee colony (ABC). *Expert Systems with Applications*, 221, 119741. doi:<https://doi.org/10.1016/j.eswa.2023.119741>
- Karlekar, Aditya, & Seal, Ayan. 2020. SoyNet: Soybean leaf diseases classification. *Computers and Electronics in Agriculture*, 172, 105342.
- Karmokar, Bikash Chandra, Ullah, Mohammad Samawat, Siddiquee, Md Kibria, & Alam, Kazi Md Rokibul. 2015. Tea leaf diseases recognition using neural network ensemble. *International Journal of Computer Applications*, 114(17).
- Kaur, Prabhjot, & Gautam, Vinay. 2021. *Plant biotic disease identification and classification based on leaf image: A review*. Paper presented at the Proceedings of 3rd International Conference on Computing Informatics and Networks: ICCIN 2020.
- Khirade, Sachin D, & Patil, AB. 2015. *Plant disease detection using image processing*. Paper presented at the 2015 International conference on computing communication control and automation.
- Kılıçarslan, Serhat. 2022. Kurum Üzüm Tanelerinin Sınıflandırılması İçin Hibrit Bir Yaklaşım. *Mühendislik Bilimleri ve Araştırmaları Dergisi*, 4(1), 62-71.
- Kiruba, Briskline, & Arjunan, Pandarasamy. 2023. *Paddy Doctor: A Visual Image Dataset for Automated Paddy Disease Classification and Benchmarking*. Paper presented at the Proceedings of the 6th Joint International Conference on Data Science & Management of Data (10th ACM IKDD CODS and 28th COMAD).
- Kong, Tao, Sun, Fuchun, Liu, Huaping, Jiang, Yuning, Li, Lei, & Shi, Jianbo. 2020. Foveabox: Beyond anchor-based object detection. *IEEE Transactions on Image Processing*, 29, 7389-7398.
- Koushik, Jayanth. 2016. Understanding convolutional neural networks. *arXiv preprint arXiv:1605.09081*.
- Kovalskaya, Natalia, & Hammond, Rosemarie W. 2014. Molecular biology of viroid–host interactions and disease control strategies. *Plant Science*, 228, 48-60.

- Krishnamoorthy, N, Prasad, LV Narasimha, Kumar, CS Pavan, Subedi, Bharat, Abraha, Haftom Baraki, & Sathishkumar, VE. 2021. Rice leaf diseases prediction using deep neural networks with transfer learning. *Environmental Research*, 198, 111275.
- Lee, Honglak, Grosse, Roger, Ranganath, Rajesh, & Ng, Andrew Y. 2009. *Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations*. Paper presented at the Proceedings of the 26th annual international conference on machine learning.
- Lee, Youngwan, Hwang, Joong-won, Lee, Sangrok, Bae, Yuseok, & Park, Jongyoul. 2019. *An energy and GPU-computation efficient backbone network for real-time object detection*. Paper presented at the Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops.
- Lemley, Joseph, Bazrafkan, Shabab, & Corcoran, Peter. 2017. *Transfer Learning of Temporal Information for Driver Action Classification*. Paper presented at the MAICS.
- Li, Bo, Liu, Zhanyu, Huang, Jingfeng, Zhang, Lili, Zhou, Wan, & Shi, Jingjing. 2009. Hyperspectral identification of rice diseases and pests based on principal component analysis and probabilistic neural network. *Transactions of the Chinese Society of Agricultural Engineering*, 25(9), 143-147.
- Lin, Tsung-Yi, Dollár, Piotr, Girshick, Ross, He, Kaiming, Hariharan, Bharath, & Belongie, Serge. 2017. *Feature pyramid networks for object detection*. Paper presented at the Proceedings of the IEEE conference on computer vision and pattern recognition.
- Lin, Tsung-Yi, Goyal, Priya, Girshick, Ross, He, Kaiming, & Dollár, Piotr. 2017. *Focal loss for dense object detection*. Paper presented at the Proceedings of the IEEE international conference on computer vision.
- Liu, Yixin, Zhang, Lihang, Hao, Zezhou, Yang, Ziyuan, Wang, Shanjuan, Zhou, Xiaoguang, & Chang, Qing. 2022. An xception model based on residual attention mechanism for the classification of benign and malignant gastric ulcers. *Scientific Reports*, 12(1), 15365.
- Malhi, Gurdeep Singh, Kaur, Manpreet, & Kaushik, Prashant. 2021. Impact of climate change on agriculture and its mitigation strategies: A review. *Sustainability*, 13(3), 1318.
- Martinelli, F., Scalenghe, R., Davino, S., Panno, S., Scuderi, G., Ruisi, P., . . . Dandekar, A. M. 2015. Advanced methods of plant disease detection. A review. *Agronomy for Sustainable Development*, 35(1), 1-25. doi:10.1007/s13593-014-0246-1
- Masood, M Hammad, Saim, Habiba, Taj, Murtaza, & Awais, Mian M. 2020. Early disease diagnosis for rice crop. *arXiv preprint arXiv:2004.04775*.
- McCarthy, John. 1980. Defense Advanced Research Projects Agency.
- McCarthy, John, Minsky, Marvin L, Rochester, Nathaniel, & Shannon, Claude E. 2006. A proposal for the dartmouth summer research project on artificial intelligence, august 31, 1955. *AI magazine*, 27(4), 12-12.
- McCulloch, Warren S, & Pitts, Walter. 1943. A logical calculus of the ideas immanent in nervous activity. *The bulletin of mathematical biophysics*, 5, 115-133.
- Mikołajczyk, Agnieszka, & Grochowski, Michał. 2018. *Data augmentation for improving deep learning in image classification problem*. Paper presented at the 2018 international interdisciplinary PhD workshop (IIPhDW).
- Mitchell, Tom Michael. 2007. *Machine learning* (Vol. 1): McGraw-hill New York.
- Murugan, D. 2022. Paddy Doctor: A Visual Image Dataset for Paddy Disease Classification. *arXiv preprint arXiv:2205.11108*.

- Nagaraju, Mamillapally, & Chawla, Priyanka. 2020. Systematic review of deep learning techniques in plant disease detection. *International journal of system assurance engineering and management*, 11, 547-560.
- Nam, Jaechang, & Kim, Sunghun. 2015. *Heterogeneous defect prediction*. Paper presented at the Proceedings of the 2015 10th joint meeting on foundations of software engineering.
- Narmadha, RP, & Arulvadivu, G. 2017. *Detection and measurement of paddy leaf disease symptoms using image processing*. Paper presented at the 2017 International Conference on Computer Communication and Informatics (ICCCI).
- Nayem, Zihad Hossain, Jahan, Iqbal, Rakib, Abdul Aziz, & Mia, Solaiman. 2023. *Detection and Identification of Rice Pests Using Memory Efficient Convolutional Neural Network*. Paper presented at the 2023 International Conference on Computer, Electrical & Communication Engineering (ICCECE).
- Nixon, Mark, & Aguado, Alberto. 2019. *Feature extraction and image processing for computer vision*: Academic press.
- Ozguven, Mehmet Metin, & Adem, Kemal. 2019. Automatic detection and classification of leaf spot disease in sugar beet using deep learning algorithms. *Physica A: Statistical Mechanics and its Applications*, 535, 122537. doi:<https://doi.org/10.1016/j.physa.2019.122537>
- Öztürk, Satuk Buğrahan, Uğuz, Sezer, Özyer, Barış, & Coşkunçay, Ahmet. 2022. The task of picking up boxes using a robotic manipulator based on semantic knowledge.
- Pacal, Ishak. 2022. Deep Learning Approaches for Classification of Breast Cancer in Ultrasound (US) Images. *Journal of the Institute of Science and Technology*, 12(4), 1917-1927.
- Pacal, Ishak, Karaboga, Dervis, Basturk, Alper, Akay, Bahriye, & Nalbantoglu, Ufuk. 2020. A comprehensive review of deep learning in colon cancer. *Computers in Biology and Medicine*, 126, 104003. doi:<https://doi.org/10.1016/j.compbimed.2020.104003>
- Pacal, Ishak, Karaman, Ahmet, Karaboga, Dervis, Akay, Bahriye, Basturk, Alper, Nalbantoglu, Ufuk, & Coskun, Seymanur. 2022. An efficient real-time colonic polyp detection with YOLO algorithms trained by using negative samples and large datasets. *Computers in Biology and Medicine*, 141, 105031. doi:<https://doi.org/10.1016/j.compbimed.2021.105031>
- Pan, Sinno Jialin, & Yang, Qiang. 2010. A survey on transfer learning. *IEEE Transactions on knowledge and data engineering*, 22(10), 1345-1359.
- Peng, Jie, Kang, Shuai, Ning, Zhengyuan, Deng, Hangxia, Shen, Jingxian, Xu, Yikai, . . . Gong, Wuxing. 2020. Residual convolutional neural network for predicting response of transarterial chemoembolization in hepatocellular carcinoma from CT imaging. *European radiology*, 30, 413-424.
- Pichler, Maximilian, & Hartig, Florian. 2022. Machine Learning and Deep Learning--A review for Ecologists. *arXiv preprint arXiv:2204.05023*.
- Polder, Gerrit, Westeringh, Nick van de, Kool, Janne, Khan, Haris Ahmad, Kootstra, Gert, & Nieuwenhuizen, Ard. 2019. Automatic Detection of Tulip Breaking Virus (TBV) Using a Deep Convolutional Neural Network**This work was partially funded by the Dutch Product Board for Horticulture and the Dutch Ministry of Economic Affairs, Agriculture and Innovation. The research is part of the public private partnership BollenRevolutie 4.0 funded under the grant TKI-TU-1806. *IFAC-PapersOnLine*, 52(30), 12-17. doi:<https://doi.org/10.1016/j.ifacol.2019.12.482>

- Prakash, Kolla Bhanu, Kannan, Ramani, Alexander, S Albert, & Kanagachidambaresan, GR. 2021. *Advanced Deep Learning for Engineers and Scientists*: Springer.
- Prasad, SP. 2021. *Compressed MobileNet V3: An efficient CNN for resource constrained platforms*.
- Ramesh, S., & Vydeki, D. 2020. Recognition and classification of paddy leaf diseases using Optimized Deep Neural network with Jaya algorithm. *Information Processing in Agriculture*, 7(2), 249-260. doi:<https://doi.org/10.1016/j.inpa.2019.09.002>
- Rangarajan, Aravind Krishnaswamy, Purushothaman, Raja, & Ramesh, Anirudh. 2018. Tomato crop disease classification using pre-trained deep learning algorithm. *Procedia Computer Science*, 133, 1040-1047. doi:<https://doi.org/10.1016/j.procs.2018.07.070>
- Salakhutdinov, Ruslan, Mnih, Andriy, & Hinton, Geoffrey. 2007. *Restricted Boltzmann machines for collaborative filtering*. Paper presented at the Proceedings of the 24th international conference on Machine learning.
- Sambasivam, G., & Opiyo, Geoffrey Duncan. 2021. A predictive machine learning application in agriculture: Cassava disease detection and classification with imbalanced dataset using convolutional neural networks. *Egyptian Informatics Journal*, 22(1), 27-34. doi:<https://doi.org/10.1016/j.eij.2020.02.007>
- Sandler, Mark, Howard, Andrew, Zhu, Menglong, Zhmoginov, Andrey, & Chen, Liang-Chieh. 2018. *Mobilenetv2: Inverted residuals and linear bottlenecks*. Paper presented at the Proceedings of the IEEE conference on computer vision and pattern recognition.
- Sengupta, Shampa, & Das, Asit K. 2017. Particle Swarm Optimization based incremental classifier design for rice disease prediction. *Computers and Electronics in Agriculture*, 140, 443-451.
- Senthil Pandi, S., Senthilselvi, A., Gitanjali, J., ArivuSelvan, K., Gopal, Jagadeesh, & Vellingiri, J. 2022. Rice plant disease classification using dilated convolutional neural network with global average pooling. *Ecological Modelling*, 474, 110166. doi:<https://doi.org/10.1016/j.ecolmodel.2022.110166>
- Sethy, Prabira Kumar, Barpanda, Nalini Kanta, Rath, Amiya Kumar, & Behera, Santi Kumari. 2020a. Deep feature based rice leaf disease identification using support vector machine. *Computers and Electronics in Agriculture*, 175, 105527. doi:<https://doi.org/10.1016/j.compag.2020.105527>
- Sethy, Prabira Kumar, Barpanda, Nalini Kanta, Rath, Amiya Kumar, & Behera, Santi Kumari. 2020b. Image Processing Techniques for Diagnosing Rice Plant Disease: A Survey. *Procedia Computer Science*, 167, 516-530. doi:<https://doi.org/10.1016/j.procs.2020.03.308>
- Sharma, Mayuri, Kumar, Chandan Jyoti, & Deka, Aniruddha. 2022. Early diagnosis of rice plant disease using machine learning techniques. *Archives of Phytopathology and Plant Protection*, 55(3), 259-283.
- Shorten, Connor, & Khoshgoftaar, Taghi M. 2019. A survey on image data augmentation for deep learning. *Journal of big data*, 6(1), 1-48.
- Siau, Keng, & Wang, Weiyu. 2018. Building trust in artificial intelligence, machine learning, and robotics. *Cutter business technology journal*, 31(2), 47-53.
- Simonyan, Karen, & Zisserman, Andrew. 2014. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*.

- Sladojevic, Srdjan, Arsenovic, Marko, Anderla, Andras, Culibrk, Dubravko, & Stefanovic, Darko. 2016. Deep neural networks based recognition of plant diseases by leaf image classification. *Computational intelligence and neuroscience*, 2016.
- Sparks, Adam. 2023a. Bacterial Leaf Streak Retrieved from <http://www.knowledgebank.irri.org/training/fact-sheets/pest-management/diseases/item/bacterial-leaf-streak>
- Sparks, Adam. 2023b. Tungro. Retrieved from <http://www.knowledgebank.irri.org/training/fact-sheets/pest-management/diseases/item/tungro#:~:text=Rice%20tungro%20disease%20is%20caused,commonly%20found%20in%20rice%20paddies.>
- Srinivas, B, Satheesh, P, Rama Santosh Naidu, P, & Neelima, U. 2021. *Prediction of guava plant diseases using deep learning*. Paper presented at the ICCCE 2020: Proceedings of the 3rd International Conference on Communications and Cyber Physical Engineering.
- Strange, Richard N, & Scott, Peter R. 2005. Plant disease: a threat to global food security. *Annu. Rev. Phytopathol.*, 43, 83-116.
- Sujithra, J, & Ukrit, M Ferni. 2020. A review on crop disease identification and classification through leaf images. *European Journal of Molecular & Clinical Medicine*, 7(09), 2020.
- Sutton, Richard S, & Barto, Andrew G. 2018. *Reinforcement learning: An introduction*: MIT press.
- Swapna, M, Sharma, Yogesh Kumar, & Prasad, BMG. 2020. CNN architectures: Alex Net, Le Net, VGG, Google Net, Res Net. *Int. J. Recent Technol. Eng*, 8(6), 953-960.
- Szegedy, Christian, Liu, Wei, Jia, Yangqing, Sermanet, Pierre, Reed, Scott, Anguelov, Dragomir, . . . Rabinovich, Andrew. 2015. *Going deeper with convolutions*. Paper presented at the Proceedings of the IEEE conference on computer vision and pattern recognition.
- Szegedy, Christian, Vanhoucke, Vincent, Ioffe, Sergey, Shlens, Jon, & Wojna, Zbigniew. 2016. *Rethinking the inception architecture for computer vision*. Paper presented at the Proceedings of the IEEE conference on computer vision and pattern recognition.
- Tan, Mingxing, & Le, Quoc. 2019. *Efficientnet: Rethinking model scaling for convolutional neural networks*. Paper presented at the International conference on machine learning.
- Tao, Ming, Ma, Xu, Huang, Xunan, Liu, Chuang, Deng, Ruoling, Liang, Kaiming, & Qi, Long. 2020. Smartphone-based detection of leaf color levels in rice plants. *Computers and Electronics in Agriculture*, 173, 105431. doi:<https://doi.org/10.1016/j.compag.2020.105431>
- Thomas, Jeena, & Raj, Ebin Deni. 2021. *Effectual single image dehazing with color correction transform and dark channel prior*. Paper presented at the Data Science and Computational Intelligence: Sixteenth International Conference on Information Processing, ICInPro 2021, Bengaluru, India, October 22–24, 2021, Proceedings 16.
- Torrey, Lisa, & Shavlik, Jude. 2010. Transfer learning. In *Handbook of research on machine learning applications and trends: algorithms, methods, and techniques* (pp. 242-264): IGI global.
- Tripathi, Abhishek D, Mishra, Richa, Maurya, Kamlesh K, Singh, Ram B, & Wilson, Douglas W. 2019. Estimates for world population and global food availability for global health. In *The role of functional food security in global health* (pp. 3-24): Elsevier.

- Trivedi, Naresh K, Gautam, Vinay, Anand, Abhineet, Aljahdali, Hani Moaiteq, Villar, Santos Gracia, Anand, Divya, . . . Kadry, Seifedine. 2021. Early detection and classification of tomato leaf disease using high-performance deep neural network. *Sensors*, 21(23), 7987.
- Turing, Alan M. 1950. Can a machine think. *Mind*, 59(236), 433-460.
- Turing, Alan Mathison. 1950. Mind. *Mind*, 59(236), 433-460.
- Udotalapally, Venkanna, Mohanty, Saraju P, Pallagani, Vishal, & Khandelwal, Vedant. 2020. sCrop: A novel device for sustainable automatic disease prediction, crop selection, and irrigation in Internet-of-Agro-Things for smart agriculture. *Ieee Sensors Journal*, 21(16), 17525-17538.
- Uğuz, Sinan, & Uysal, Nese. 2021. Classification of olive leaf diseases using deep convolutional neural networks. *Neural Computing and Applications*, 33(9), 4133-4149.
- van Eeuwijk, F. A., Bustos-Korts, D., Millet, E. J., Boer, M. P., Kruijer, W., Thompson, A., . . . Chapman, S. C. 2019. Modelling strategies for assessing and increasing the effectiveness of new phenotyping techniques in plant breeding. *Plant Science*, 282, 23-39. doi:10.1016/j.plantsci.2018.06.018
- Velesaca, Henry O, Mira, Raúl, Suárez, Patricia L, Larrea, Christian X, & Sappa, Angel D. 2020. *Deep learning based corn kernel classification*. Paper presented at the Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops.
- Verma, Toran, & Dubey, Sipi. 2018. *Optimizing Rice Plant Diseases Recognition in Image Processing and Decision Tree Based Model*. Paper presented at the Smart and Innovative Trends in Next Generation Computing Technologies: Third International Conference, NGCT 2017, Dehradun, India, October 30-31, 2017, Revised Selected Papers, Part II 3.
- Villanueva, Marizel B, & Salenga, Ma Louella M. 2018. Bitter melon crop yield prediction using machine learning algorithm. *International Journal of Advanced Computer Science and Applications*, 9(3).
- Vincent Pascalvincent, P, Larocheh, Larochele, & Autoencoders, H Stacked Denoising. 2010. Learning Useful Representations in a Deep Network with a Local Denoising Criterion Pierre-Antoine Manzagol. *J. Mach. Learn Res*, 11, 3371-3408.
- Waheed, Abdul, Goyal, Muskan, Gupta, Deepak, Khanna, Ashish, Hassanien, Aboul Ella, & Pandey, Hari Mohan. 2020. An optimized dense convolutional neural network model for disease recognition and classification in corn leaf. *Computers and Electronics in Agriculture*, 175, 105456.
- Wang, Chang, & Mahadevan, Sridhar. 2011. *Heterogeneous domain adaptation using manifold alignment*. Paper presented at the IJCAI proceedings-international joint conference on artificial intelligence.
- Wang, G., Sun, Y., & Wang, J. 2017. Automatic Image-Based Plant Disease Severity Estimation Using Deep Learning. *Comput Intell Neurosci*, 2017, 2917536. doi:10.1155/2017/2917536
- Xiao, Maohua, Ma, You, Feng, Zhixiang, Deng, Ziang, Hou, Shishuang, Shu, Lei, & Lu, ZhiXiong. 2018. Rice blast recognition based on principal component analysis and neural network. *Computers and Electronics in Agriculture*, 154, 482-490. doi:<https://doi.org/10.1016/j.compag.2018.08.028>

- Xie, Saining, Girshick, Ross, Dollár, Piotr, Tu, Zhuowen, & He, Kaiming. 2017. *Aggregated residual transformations for deep neural networks*. Paper presented at the Proceedings of the IEEE conference on computer vision and pattern recognition.
- Xu, Guili, Zhang, Fengling, Shah, Syed Ghafoor, Ye, Yongqiang, & Mao, Hanping. 2011. Use of leaf color images to identify nitrogen and potassium deficient tomatoes. *Pattern Recognition Letters*, 32(11), 1584-1590. doi:<https://doi.org/10.1016/j.patrec.2011.04.020>
- Yağanoğlu, Mete. 2021. Real time wearable speech recognition system for deaf persons. *Computers & Electrical Engineering*, 91, 107026.
- Yağanoğlu, Mete. 2022. Hepatitis C virus data analysis and prediction using machine learning. *Data & Knowledge Engineering*, 142, 102087.
- Yang, Fan, Fan, Heng, Chu, Peng, Blasch, Erik, & Ling, Haibin. 2019. *Clustered object detection in aerial images*. Paper presented at the Proceedings of the IEEE/CVF International Conference on Computer Vision.
- Yang, WeiJuan, Chen, JingHua, Chen, GuoNan, Wang, ShiHua, & Fu, FengFu. 2013. The early diagnosis and fast detection of blast fungus, *Magnaporthe grisea*, in rice plant by using its chitinase as biochemical marker and a rice cDNA encoding mannose-binding lectin as recognition probe. *Biosensors and Bioelectronics*, 41, 820-826.
- Yılmaz, Atınç. 2021. *Yapay Zeka*: KODLAB YAYIN DAĞITIM YAZILIM LTD. ŞTİ.
- Zeigler, Robert S, & Barclay, Adam. 2008. The relevance of rice. In (Vol. 1, pp. 3-10): Springer.
- Zeiler, Matthew D, & Fergus, Rob. 2014. *Visualizing and understanding convolutional networks*. Paper presented at the Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part I 13.
- Zhang, Qianru, Zhang, Meng, Chen, Tinghuan, Sun, Zhifei, Ma, Yuzhe, & Yu, Bei. 2019. Recent advances in convolutional neural network acceleration. *Neurocomputing*, 323, 37-51.
- Zhang, Shifeng, Wen, Longyin, Bian, Xiao, Lei, Zhen, & Li, Stan Z. 2018. *Single-shot refinement neural network for object detection*. Paper presented at the Proceedings of the IEEE conference on computer vision and pattern recognition.
- Zhou, Xingyi, Wang, Dequan, & Krähenbühl, Philipp. 2019. Objects as points. *arXiv preprint arXiv:1904.07850*.
- Zhu, Wenjing, Chen, Hua, Ciechanowska, Izabela, & Spaner, Dean. 2018. Application of infrared thermal imaging for the rapid diagnosis of crop disease. *IFAC-PapersOnLine*, 51(17), 424-430. doi:<https://doi.org/10.1016/j.ifacol.2018.08.184>
- Zhu, Xiaojin Jerry. 2005. Semi-supervised learning literature survey.

ÖZGEÇMİŞ

Kişisel Bilgiler	
Adı Soyadı:	Erkan VEZİROĞLU
Doğum tarihi:	
Doğum Yeri:	
Uyruğu:	
Adres:	
Tel:	
E-mail:	
Eğitim	
Lise:	Erzincan Fen Lisesi
Lisans:	Sakarya Üniversitesi, Bilgisayar ve Bilişim Bilimleri Fakültesi, Bilgisayar Mühendisliği (2013-2018)
Yüksek lisans:	Atatürk Üniversitesi, Fen Bilimleri Enstitüsü, Bilgisayar Mühendisliği Anabilim Dalı (2020-2023)
Yabancı Dil Bilgisi	
İngilizce:	İyi
Üye Olunan Mesleki Kuruluşlar	
Tezden Üretilmiş Yayınlar	