



**KONVOLÜSYONEL SİNİR AĞLARINI KULLANARAK BİTKİ TÜRLERİNİN
SINIFLANDIRILMASI VE BİTKİ HASTALIKLARININ TANISI**

FUAT KARA

Bilgisayar Mühendisliği Anabilim Dalı

Yüksek Lisans Tezi

Danışman: Doç. Dr. Pınar TÜFEKÇİ

2022

T.C.
TEKİRDAĞ NAMIK KEMAL ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ



KONVOLÜSYONEL SİNİR AĞLARINI KULLANARAK BİTKİ TÜRLERİNİN
SINIFLANDIRILMASI VE BİTKİ HASTALIKLARININ TANISI

FUAT KARA

ORCID: 0000-0002-5498-0361

BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI
YÜKSEK LİSANS TEZİ
Danışman: Doç. Dr. Pınar TÜFEKÇİ

NİSAN-2022

Her hakkı saklıdır.

ÖZET

KONVOLÜSYONEL SINİR AĞLARINI KULLANARAK BİTKİ TÜRLERİNİN SINIFLANDIRILMASI VE BİTKİ HASTALIKLARININ TANISI

Fuat KARA

Bilgisayar Mühendisliği Anabilim Dalı

Yüksek Lisans Tezi

Danışman: Doç. Dr. Pınar TÜFEKÇİ

Görüntü sınıflandırma, başta tarım, sağlık, ulaşım olmak üzere birçok alana hizmet sunmakla birlikte, insan gücü ile tahmin edilmesi güç problemlerde kullanmış olduğu yapay zeka mimarileriyle birlikte çok kısa sürede ve çok daha az maliyet ile gerçekleştirmektedir. Tarım alanında ise, bitki türlerinin belirlenmesi, bitki hastalıklarının tahminleri gibi süreçler insan gücü ile laboratuvar ortamlarında çok uzun süreçlerde gerçekleştirmekte olup, alınacak tedavide geç kalılabilmektedir. Derin öğrenme algoritmalarından olan konvolüsyonel sinir ağı, bitkilerin sınıflandırmaları ve sahip oldukları hastalıkların teşhislerinde çok kısa sürelerde çok başarılı sonuçlar sunmaktadır. Bu çalışmada, popüler bir veri seti olan Plantvillage veri seti ele alınmıştır ve bu veri setinden, büyük ölçekli ve küçük ölçekli durumlarının yanı sıra, çok sınıflı ve az sınıflı olmak üzere toplamda 8 adet yeni veri seti oluşturulmuştur. Oluşturulan bu veri setleri kullanılarak, bitki türlerinin sınıflandırılması, bitki hastalık tahmini ve bitki hastalıklarının sınıflandırılması için modelleme çalışmaları gerçekleştirilmiştir ve her bir sınıflandırma problemi için, geniş kapsamlı deneysel çalışmalar sonucunda en iyi modeller bulunmaya çalışılmıştır. İlk modelleme çalışmalarında, bitki türlerinin sınıflandırılması problemine yönelik hazırlanmış olan 6 farklı dengeli veri seti kullanılarak, Alexnet mimarisi uygulanmıştır ve en iyi model, % 99,88 test doğruluk başarısı elde edilmiştir. İkinci modelleme çalışmalarında, bitki hastalık tahmini problemine yönelik hazırlanmış olan, 2 sınıftan oluşan dengeli veri seti kullanılarak, literatür sonuçlarında gözlemlenen CNN'in en popüler 9 farklı derin öğrenme modelleri uygulanmıştır. Bitki hastalık tahmini için en iyi model, GoogleNet/Inception mimarisinin kullanıldığı model olmuştur ve 3 kez çapraz doğrulama ile %99,91 doğruluk başarısı elde edilmiştir. Son olarak, bitki hastalıklarının sınıflandırılması problemine yönelik, kullanılan veri seti ile DenseNet modeli en iyi model olarak bulunmuştur ve 10 kez çapraz doğrulama ile %99,14 doğruluk başarısı elde edilmiştir.

Anahtar Kelimeler: Görüntü Sınıflandırma, Bitki Hastalıkları Tanısı, CNN, Akıllı Tarım Uygulama Alanları, Konvolüsyonel Sinir Ağları.

ABSTRACT

CLASSIFICATION OF PLANT SPECIES AND DIAGNOSIS OF PLANT DISEASES USING CONVOLUTIONAL NEURAL NETWORKS

Fuat KARA

Department of Computer Engineering

MSc. Thesis / PhD Thesis

Supervisor: Doç. Dr. Pınar TÜFEKÇİ

Image classification provides services in many fields, especially agriculture, health, transportation, and it performs in a very short time and with much less cost, with the artificial intelligence architectures used in man-powered problems. In the field of agriculture, processes such as determination of plant species and estimates of plant diseases are carried out in human lab in a very long period of time, and the treatment to be taken can be delayed. Convolutional neural network, which is one of the deep learning algorithms, offers very successful results in very short periods in the classification of plants and in the diagnosis of their diseases. Within the scope of the study, a total of 8 data sets, including large-scale and small-scale cases, as well as multi-class and low-class, were created using the Plantvillage dataset, which is open to the public. Modelling studies were carried out in which it was used as a classifier in 3 phases: classification of plant species, prediction of plant diseases and diagnosis of plant diseases. In the first phase, Alexnet's comparisons were made on 6 different balanced datasets prepared for the classification of plant species and a better model was tried to be obtained with the changes in padding values. In the second phase, the balanced data set consisting of 2 classes prepared for the plant disease prediction problem was applied as a classifier in the most popular 9 different deep learning models of CNN observed in the literature, and the results were compared and 99 for the 3 cross-validation (CV-3) part. With 91 test accuracy, the GoogleNet/Inception model was chosen as the most optimal model. In the last phase, it has been observed that there is a DenseNet model with the dataset for the plant diseases diagnosis problem and a DenseNet model with a test accuracy of 99.14 for the 10 cross-validation (CV-10) parts. In this study, the literature of which was also carried out, comparisons of different CNN models and the performance of the models rearranged with transfer learning were evaluated.

Keywords: Image Classification, CNN, Smart Agriculture Application Areas, Convolutional Neural Network.

İÇİNDEKİLER

ÖZET	i
ABSTRACT	ii
İÇİNDEKİLER	iii
ÇİZELGELER DİZİNİ	v
ŞEKİLLER DİZİNİ	vi
1. GİRİŞ	10
1.1 Literatür Özeti	10
1.2 Çalışmanın Amacı ve Kapsamı	21
2. MATERYEL VE YÖNTEM.....	22
2.1 Orjinal Veri Seti Tanıtımı	22
2.1.1 Bitki Türü Sınıflandırılması İçin Yeni Oluşturulan Veri Setleri.....	25
2.1.2 Bitki Hastalık Tahmini İçin Yeni Oluşturulan Veri Setleri.....	26
2.1.3 Bitki Hastalıklarının Sınıflandırılması İçin Yeni Oluşturulan Veri Setleri.....	27
2.2 Veri Setinin Bölünmesi	28
2.3 Görüntü Sınıflandırma.....	29
2.4 Derin Öğrenme.....	29
2.5 Konvolüsyonel Sinir Ağ (CNN)	30
2.6 Popüler CNN Mimarileri	32
2.6.1 AlexNet Mimarisi.....	33
2.6.2 VggNet Mimarisi	34
2.6.2.1 VGG16Net Mimarisi	34
2.6.2.2 Vgg19Net Mimarisi	35
2.6.3 GoogleNet Mimarisi	36
2.6.4 ResNet50 Mimarisi.....	37
2.6.5 MobileNet Mimarisi	37
2.6.6 DenseNet Mimarisi.....	38
2.6.7 Xception Mimarisi	39
2.6.8 NasNetMobile Mimarisi	39
2.7 Transfer Öğrenme	39
2.8 Değerlendirme Ölçütleri	41
2.9 Çalışmada Kullanılan Kütüphaneler	42
3. ARAŞTIRMA BULGULARI	43

3.1 Bitki Türlerinin Sınıflandırılması	43
3.2 Faz 2 - Bitki Hastalık tahmini	57
3.1 Bitki Hastalıklarının Sınıflandırılması.....	66
4.1 Model Sonuçları ve Sonuçların Değerlendirilmesi	75
4. SONUÇ VE ÖNERİLER	91
KAYNAKLAR	93
ÖZGEÇMİŞ.....	Error! Bookmark not defined.



ÇİZELGELER DİZİNİ

Çizelge 2.1. Literatür çalışması.....	11
Çizelge 3.1. Plantvillage Veri Seti detaylı gösterim.....	24
Çizelge 3.2. Bitki Türü Sınıflandırması için oluşturulan veri setlerinin detaylı gösterimi.....	25
Çizelge 3.3. Bitki Hastalık Tahmin problemi için oluşturulan dengeli dataset (D7-BHT).....	26
Çizelge 3.4. Bitki Hastalıklarının Sınıflandırılması problemi için oluşturulan dataset (D8-BHS).....	27
Çizelge 3.5. Çalışmada Kullanılan Kütüphaneler.....	43
Çizelge 4.1. Detaylı Orjinal AlexNet Model Mimarisi	44
Çizelge 4.2. D1-BTS veri seti için Transfer Öğrenme Uygulanarak Geliştirilmiş AlexNet Model Mimarisi.....	46
Çizelge 4.3. D2-BTS veri seti için Transfer Öğrenme Uygulanarak Geliştirilmiş AlexNet Model Mimarisi	48
Çizelge 4.4. D3-BTS veri seti için Transfer Öğrenme Uygulanarak Geliştirilmiş AlexNet Model Mimarisi.....	50
Çizelge 4.5. D4-BTS veri seti için Transfer Öğrenme Uygulanarak Geliştirilmiş AlexNet Model Mimarisi.....	51
Çizelge 4.6. D5-BTS veri seti için Transfer Öğrenme Uygulanarak Geliştirilmiş AlexNet Model Mimarisi.....	53
Çizelge 4.7. D6-BTS veri seti için Transfer Öğrenme Uygulanarak Geliştirilmiş AlexNet Model Mimarisi.....	55
Çizelge 4.8. Bitki türlerinin sınıflandırılması problemi için 3 kat çapraz doğrulama sonuçları.....	75
Çizelge 4.9. Bitki türlerinin sınıflandırılması problemi için 5 kat çapraz doğrulama sonuçları.....	76
Çizelge 4.10. Bitki türlerinin sınıflandırılması problemi için 10 kat çapraz doğrulama sonuçları.....	77
Çizelge 4.11. Bitki hastalık tahmini problemi için 3 kat çapraz doğrulama sonuçları.....	78
Çizelge 4.12. Bitki hastalık tahmini problemi için 5 kat çapraz doğrulama sonuçları.....	78
Çizelge 4.13. Bitki hastalık tahmini problemi için 10 kat çapraz doğrulama sonuçları.....	79
Çizelge 4.14. Bitki hastalıklarının sınıflandırılması problemi için 3 kat çapraz doğrulama sonuçları.....	82
Çizelge 4.15. Bitki hastalıklarının sınıflandırılması problemi için 5 kat çapraz doğrulama sonuçları.....	83
Çizelge 4.16. Bitki hastalıklarının sınıflandırılması problemi için 10 kat çapraz doğrulama sonuçları.....	83

ŞEKİLLER DİZİNİ

Şekil 3.1. Plantvillage veri seti her sınıf örnek gösterimi.....	21
Şekil 3.2. 3 Kez Çapraz Doğrulama.....	27
Şekil 3.3. 5 Kez Çapraz Doğrulama.....	27
Şekil 3.4. 10 Kez Çapraz Doğrulama.....	28
Şekil 3.5. Konvolüsyonel Sinir Ağı yapısı.....	23
Şekil 3.6. AlexNet Mimari Yapısı.....	33
Şekil 3.7. VggNet mimari yapısı.....	34
Şekil 3.8. Vgg16Net mimari yapısı.....	34
Şekil 3.9. Vgg19Net mimari yapısı.....	35
Şekil 3.10. GoogleNet Mimari Yapısı.....	36
Şekil 3.11. ResNet50 Mimari Yapısı.....	37
Şekil 3.12. MobileNet Mimari Yapısı.....	37
Şekil 3.13. DenseNet Mimari Yapısı.....	38
Şekil 3.14. Örnek Transfer Öğrenme Modeli.....	39
Şekil 4.1. Zero Padding Örneği.....	44
Şekil 4.2. D1-BTS veri seti için geliştirilmiş AlexNet model parametresi.....	45
Şekil 4.3. D2-BTS Veri Seti için Geliştirilmiş AlexNet Model Parametresi.....	47
Şekil 4.4. D3-BTS Veri Seti için Geliştirilmiş AlexNet Model Parametresi.....	48
Şekil 4.5. D4-BTS Veri Seti için Geliştirilmiş AlexNet Model Parametresi.....	50
Şekil 4.6. D5-BTS Veri Seti için Geliştirilmiş AlexNet Model Parametresi.....	52
Şekil 4.7. D6-BTS Veri Seti için Geliştirilmiş AlexNet Model Parametresi.....	54
Şekil 4.8. D7-BHT Veri Seti ile Eğitilen VGGNET16 Modeli.....	56
Şekil 4.9. D7-BHT Veri Seti ile Eğitilen VGGNET19 Modeli.....	57
Şekil 4.10. D7-BHT Veri Seti ile Eğitilen AlexNet modeli.....	58
Şekil 4.11. D7-BHT Veri Seti ile Eğitilen ResNet50 Modeli Parametre Gösterimi.....	59
Şekil 4.12. D7-BHT veri seti ile eğitilen DenseNet modeli parametre gösterimi.....	60
Şekil 4.13. D7-BHT veri seti ile eğitilen GoogleNet / InceptionV3 modeli parametre gösterimi.....	61
Şekil 4.14. D7-BHT veri seti ile eğitilen MobileNet modeli parametre gösterimi.....	62
Şekil 4.15. D7-BHT veri seti ile eğitilen NasNetMobile modeli parametre gösterimi.....	63
Şekil 4.16. D7-BHT veri seti ile eğitilen Xception modeli parametre gösterimi.....	64
Şekil 4.17. D8-BHS veri seti ile eğitilen VGGNET16 modeli.....	65
Şekil 4.18. D8-BHS veri seti ile eğitilen VGGNET19 modeli.....	66
Şekil 4.19. D8-BHS veri seti ile eğitilen AlexNet modeli.....	67
Şekil 4.20. D8-BHS veri seti ile eğitilen ResNet50 modeli parametre gösterimi.....	68
Şekil 4.21. D8-BHS veri seti ile eğitilen DenseNet modeli parametre gösterimi.....	69
Şekil 4.22. D8-BHS veri seti ile eğitilen GoogleNet / InceptionV3 modeli parametre gösterimi.....	70
Şekil 4.23. D8-BHS veri seti ile eğitilen MobileNet modeli parametre gösterimi.....	71
Şekil 4.24. D8-BHS veri seti ile eğitilen NasNetMobile modeli parametre gösterimi.....	72
Şekil 4.25. D8-BHS veri seti ile eğitilen Xception modeli parametre gösterimi.....	73
Şekil 4.26. 3 Kat Çapraz Doğrulama ile Bitki Türlerinin Sınıflandırılması için AlexNet Modelleri	74
Şekil 4.27. 5 Kat Çapraz Doğrulama ile Bitki Türlerinin Sınıflandırılması için AlexNet Modelleri.....	75
Şekil 4.28. 10 Kat Çapraz Doğrulama ile Bitki Türlerinin Sınıflandırılması için AlexNet Modelleri	76

Şekil 4.29. Bitki Hastalık Tahmini Problemi için D7-BHT Veri Seti Kullanılarak 3 Kat Çapraz Doğrulama Model Sonuçları.....	79
Şekil 4.30. Bitki Hastalık Tahmini Problemi için D7-BHT Veri Seti Kullanılarak 5 Kat Çapraz Doğrulama Model Sonuçları.....	80
Şekil 4.31. Bitki Hastalık Tahmini Problemi için D7-BHT Veri Seti Kullanılarak 10 Kat Çapraz Doğrulama Model Sonuçları.....	80
Şekil 4.32 Bitki Hastalıklarının Sınıflandırılması Problemi için D8-BHS Veri Seti Kullanılarak 3 Kat Çapraz Doğrulama Model Sonuçları.....	83
Şekil 4.33 Bitki Hastalıklarının Sınıflandırılması Problemi için D8-BHS Veri Seti Kullanılarak 5 Kat Çapraz Doğrulama Model Sonuçları.....	84
Şekil 4.34 Bitki Hastalıklarının Sınıflandırılması Problemi için D8-BHS Veri Seti Kullanılarak 10 Kat Çapraz Doğrulama Model Sonuçları.....	85
Şekil 4.35. Verilerin rastgele teyit işlemleri.....	86
Şekil 4.36. Verilerin rastgele teyit işlemleri.....	86
Şekil 4.37. CNN model tahmin sonuç analiz örneği.....	87



KISALTMALAR

CNN	: Convolutional Neural Network
CV	: Cross Validation / apraz dođrulama
Acc	: Accuracy / Dođruluk
EtoE	: End to end model
Dataset	: Veri seti
TL	: Transfer Learning (Transfer ğrenme)
Fine-tuning	: İnce Ayar



TEŐEKKÜR

Yüksek lisans eğitimimde ve bu tez çalışmamda gösterdiği her türlü destek, ilgi ve yardımlarından dolayı çok kıymetli danışman hocam Doç. Dr. Pınar TÜFEKÇİ'ye en içten duygularıyla teşekkürlerimi sunarım. Yüksek lisans ders aşamasında dersini aldığı Tekirdağ Namık Kemal Üniversitesi'ndeki tüm hocalarıma teşekkürlerimi sunuyorum. Eğitim öğretim hayatım boyunca her zaman bana destek olan, maddi ve manevi yardımlarını esirgemeyen aileme, arkadaşlarıma ve bugünlere gelmemde üzerimde emeđi bulunan tüm öğretmenlerime çok teşekkür ederim.

Fuat KARA

Veri Mühendisi



1. GİRİŞ

Tarımsal üretim için ekonomik kayıpların önlenmesinde, bitkilerin sınıflandırılması ve bitki hastalıklarının tanısı önem arz etmektedir. Günümüzde bitkilerde yaşanan çeşitli hastalıkların tespiti ziraat mühendisleri tarafından laboratuvar ortamlarında yapılmaktadır. Ancak bu konudaki işlemler uzun zaman almakla birlikte, maliyetli ve düşük performanslı sonuçlar verebilmektedir. Bundan dolayı, öncelikle bitki türlerinin ve sonra da bitki hastalıklarının otomatik bir şekilde tespit edilip sınıflandırılmasına yönelik, makine öğrenmesi ve bilgisayar görmesi yöntemlerine dayalı çalışmaların önem kazandığı görülmektedir (Türkoğlu & Hanbay, 2019).

Son zamanlarda, oldukça gelişim sağlayan ve hayatımıza giren bilgisayar destekli tanı sistemleri, gelişmekte olan teknoloji ile birlikte büyük verilerin oluşması ve büyük ölçekli hafızalara sahip daha hızlı bilgisayarların üretilmesiyle, derin öğrenme algoritmalarına dayalı modeller geliştirilmesinde büyük rol almaktadır ve bitki hastalıklarının tespitine yönelik birçok derin öğrenme tabanlı çalışmaların yapıldığı görülmektedir (TÜRKOĞLU et al., 2020).

Derin öğrenme, özellikle kolay uygulanabilirliğinden dolayı, başta görüntü sınıflandırma konusunda olmak üzere birçok alanda çalışmalara ışık tutmaktadır. Özellikle karmaşık durumlardaki hastalıkların bile tanısında insan gücünden çok daha hızlı sürede ve çok daha az çabalar sonucunda ulaşılması konusunda büyük kolaylıklar sunduğuna birçok kez şahit olunmuştur (Sharma et al., 2019). Buna ek olarak derin öğrenme modellerinin, modelin kendi içini nasıl değiştirmesi gerektiğini gösteren yinelemeli algoritmalarının oluşu, geri yayılım kullanarak büyük veri kümelerini her katmandaki özellik temsilcilerini öğrenmek için farklı parametreleri barındırmasından gibi özelliklerinden dolayı yetenekli modeller olarak bilinmektedir (LeCun vd., 2015).

Bu çalışma kapsamında, derin öğrenme yaklaşımlarının kullanıldığı, görüntü sınıflandırmaya dayanan, başta bitki türleri olmak üzere, bitkilerin hastalıklı mı yoksa sağlıklı mı oldukları, eğer hastalıklı ise hangi tür hastalığa sahip olduklarını otomatik bir şekilde tespit etmeye çalışan, üç aşamalı bir çalışma sunulmuştur.

1.1 Literatür Özeti

Bu bölümde bitki türü sınıflandırma ve bitki hastalık tahminleriyle ilgili yapılmış olan önceki çalışmalara yer verilmiştir. Tablo-1'de, her bir çalışmada kullanılan veri seti, her bir veri

setindeki toplam örnek sayısı, model mimarisi ve modelin doğruluk başarı oranları görülmektedir.

Çizelge 1.1. Literatür Çalışması

Yıl	Yazar	Problem	Veri Seti	Toplam Örnek Sayısı	Model	Başarı (Acc. %)
2016	SharadaP.Mohanty ve ark.	Bitki hastalıkları Tahmini	Plantvillage	54.306	GoogleNet:TransferLearning	99,35
2016	Mads Dyrmann ve ark.	Bitki sınıflandırma	6 farklı tel ile çekilen dataseti	10.413	Ayarlanmış CNN-Fully Connected ve Softmax Akt. Fn	86.20
2017	Sue Han Lee ve ark.	Yaprak özellikleriyle bitki sınıflandırma	Malayakew Dataset	2816	D1-CNN model – ReLu Aktivasyon fonksiyonu Accuracy	96,30
2017	Wang-Su ve ark.	Yaprak özellikleriyle bitki sınıflandırma	Flevia Dataset	3867	Ayarlanmış GoogleNet modeli accuracy	98,40
2018	Jayme Garcia Arnal Barbedo	Bitki hastalıkları Tahmini	Barbedo	1383	CNN(GoogLeNet) - Tranfer Learning	84,00
2018	Aravind Krishnaswamy Rangarajan ve Ark.	Domates yaprağı hastalık tahmini	Plantvillage	13262	AlexNet, minibatch size of 32 with a learning rate of 40	97.49
2018	Konstantinos P. Ferentinos	Bitki hastalıklarının tespiti ve teşhisi	Plantvillage	87848	VGG'net ile Bitki ve Hasyalık kombinasyonlarında en yüksek başarıım sağlanmıştır.	99.53
2019	Parul Sharma ve ark.	Bitki hastalıkları Tahmini	Plantvillage, Yerel Çiftlik, WebSearch	17929	S-CNN modeli ve Softmax Aktivasyon Fonksiyonu	98,60

Çizelge 1.1 devamı. Literatür Çalışması

2019	Artzai Picon ve ark.	Bitki hastalıkları Tahmini	Barbedo	121955	RESNET-MC-1	98,00
2019	Ashraf Darwish ve ark.	Bitki hastalıkları tahmini	CIFAR10	15408	Geliştirilmiş VGG16 ve VGG19 mimarileri	98,20
2019	Aigerim KAIROLDAYEVA	Bitki Türü Sınıflandırma	TARBIL	9000	MobiLeNet version 2	97,72
2019	Zhendong ve ark.	Bitki hastalıkları Tahmini	Plantvillage	82161	Reduced-MobileNet Accuracy	98,34
2019	Wei Xue ve ark.	Tarım atık tahmini	composted dataset	Around 3000	ResNet Modeli	99,70
2019	Aydin Kaya ve ark.	Bitki Sınıflandırma	Flavia, Swedish Leaf, UCI Yaprak veri kümesi Plantvillage	57781	2T2 CNN Transfer Learning	98,70
2020	Muhammad Hammad Saleem ve ark.	Bitki hastalıkları tahmini	Plantvillage	54306	18 Different Model's comparison (Xception best)	99,90
2020	Ahmet Saygılı ve ark.	Domates yaprağı hastalık tahmini	Plantvillage	2000	Sequential Modeli	97,05
2020	Shreyas Rajesh Labhsetwar ve Ark.	Bitki hastalıkları Sınıflandırma	Plantvillage	30000	The proposed 28-layer Sequential CNN model by	98,00

					Adam optimizer, with the maximum validation accuracy being 98%	
--	--	--	--	--	--	--

Çizelge 1.1 devamı. Literatür Çalışması

2020	Herve ve ark.	Bitki hastalıkları Tahmini	Plantvillage, PlantCLEF2015, IPM, Bing	82161	Transfer Learning Modeli-VGG16 fine-tuning on ImageNet pre-trained CNN (FTIN)	99,95
2021	Amren Abbas ve ark.	Domates yaprağı hastalık tahmini	Plantvillage, Synthetic images	16012	Yeniden düzenlenmiş DenseNet121, 10-Class sınıflandırma modeli	99,51
2021	Punam Bedi ve ark.	Bitki hastalıkları Tahmini	Plantvillage	9914	Hybrid Convolutional Autoencoder network	98,38

Sharada P. Mohanty ve arkadaşları (Mohanty et al., 2016), yaptıkları çalışma kapsamında bitkisel hastalıkların tanısı, akıllı telefon destekli hastalık tanısıyla çok daha kısa sürelerde gerçekleştirilmesi amaçlanmışlardır. Plantvillage (54.306) veri seti kullanılmış olup, içerisinde 38 sınıf etiketi içerecek bitki türleri mevcuttur. Her sınıf etiketi bir bitki hastalığı çiftidir ve sadece bitki yaprağının görüntüsünde verilen bitki hastalığı çiftini tahmin etmeye çalışılmıştır. Bu makalede açıklanan tüm yaklaşımlarda, görüntüler 256×256 piksele yeniden boyutlandırılmış ve bu küçültülmüş görüntüler üzerinde hem model optimizasyonu hem de tahminleri gerçekleştirilmiştir. Bu dataseti üzerinde yer alan renkli yaprakları grileştirilme işlemi de yapılmıştır. AlexNet ve GoogLeNet Derin öğrenme yaklaşımları karşılaştırılmış ve GoogLeNet'in, AlexNet'e oranla çok daha yüksek bir başarı elde edildiği kaydedilmiştir. Çalışmanın son aşamasında farklı Train Test bölmeleri sonucunda GoogLeNet ile %99.35'lik bir başarı elde edilmiştir.

Mads, Henrik ve arkadaşlarının (Dyrmann et al., 2016), evrişimli bir sinir ağı kullanarak renkli görüntülerdeki bitki türlerini tanıyabilecek bir yöntem sunmaktadır. Ağ, erken büyüme aşamalarında 22 yabancı ot ve bitki türü içeren toplam 10.413 görüntü üzerinde eğitilmiş ve test edilmiştir. Bu görüntüler, aydınlatma, çözünürlük ve toprak türüne göre farklılık gösteren altı farklı veri setinden kaynaklanmaktadır. 22 farklı bitki türünü veya familyasını kapsayan fidelerin dikey olarak fotoğraflanan görüntüleri üzerinde gerçekleştirilmiştir. Geliştirilen Model Ağ, 128 x 128 RGB giriş olarak alınmış ve her sınıf için bir giriş ile her görüntü için bir vektör çıkmaktadır. Bu, filtre sayısını 64'ten 32'ye düşüren bir 1x1 evrişimsel katman halinde haritalanmıştır. Daha sonra, her biri 2x5 maksimum evrimleşme katmanından oluşan iki artık

blok gelmiş ve bunu bir 2x2 maksimum havuzlama katmanı izlemiştir. Bundan sonra, bir tane daha 5x5 evrişimsel katman ve 2x2 max-havuzlama katmanı kullanılmıştır. Ek olarak Python için Theano tabanlı Lasagne kütüphanesi kullanılmıştır. Bu 22 tür için, ağ %86.2'lik bir sınıflandırma doğruluğu elde edebilmektedir.

Sue Han Lee ve arkadaşları (Lee et al., 2017), bu çalışmada yaprak görüntülerinden ayırt edici özellikleri öğrenerek derin öğrenme ile birlikte bitki tanımlama için sınıflandırıcılar olarak uygulaması amaçlanmıştır. Deneysel kapsamında gerçekleştirilen bu çalışmada, özellikle DN ve CNN kıyaslaması yapılmıştır. Malayakew Dataset kullanılmış olup, yaprak görüntülerini bir bütün olarak değerlendirilmiştir, burada her yaprak görüntüsü, ön plan pikselleri HSV renk alanı bilgileri kullanılarak çıkarılmıştır. D1 veri kümesini büyütme için her yaprak görüntüsünü 7 farklı yönde döndürülmüştür, ör. 45°, 90°, 135°, 180°, 225°, 270° ve 360°. Daha sonra test için 528 yaprak görüntüsünü ve eğitim için 2288 görüntüyü rastgele seçmişlerdir. Oluşturulan CNN modeli 11 katmandan oluşmakta olup Aktivasyon Fonksiyonu olarak ta ReLu kullanılmıştır. Model eğitilirken Caffe framework'u kullanılmıştır. Özellikle üzerinde durulan DN ve CNN karşılaştırmasının yanı sıra MLP, SVM gibi sınıflandırıcılarla da kıyaslandığında CNN %98,10'luk bir doğruluk başarısı sağlamıştır.

Wang-Su Jeon ve Sang YongRhee adlı arkadaşların (Jeon & Rhee, 2017), bitki türlerinin insanlar tarafından kolayca ve kısa sürede sınıflandırılmamasının sonucu olarak GoogleNet mimarisini kullanarak bir CNN model oluşturmuşlardır. İlk model sonrasında 2. model olan düzenlenmiş GoogleNet modeli oluşturulma ihtiyacı hissetmişlerdir. İki model arasında kıyaslamalar yapılmıştır. Flavia adlı data seti kullanılarak 3767'den oluşan train verisi ve 100 adet test verisi ile test edilmiştir. Deneyde kullanılan her bir görüntünün boyutu, modele uyacak şekilde 1600×1200 ila 229×229 arasında ayarlanmıştır. Ayrıca doğada yaygın olduğu gibi rastgele kesilmiş veya oyulmuş yapraklar oluşturularak yaprakların renk değiştirmesini veya deforme olmasını test edilmiştir. İki modelde de renkleri değiştirilen ve oyulan yapraklar modellerin öğrenmesinde zorlanmasının yanı sıra iki model de başarılı sonuçlar vermektedir, 2. model 1. modele kıyasla çok küçük fark ilk daha yüksek başarı elde etmektedir. CNN GoogleNet'in derinliğinin ayarlanmış olduğu bu model sonucu %94'lük bir başarı elde edilmiştir.

Jayme (Barbedo, 2018), her bir örnek sayısı, hastalık sayısı ve çeşitli koşullar açısından çok farklı özellikler gösteren 12 bitki türü içeren bir görüntü veritabanına dayanan bir veri seti üzerinde çalışmıştır. Barbedo (2016) benzeyen bir veriseti kullanılmış olup, sadece

yapraklarından değil sapı ve gövdesinden de oluşmaktadır. Bitki hastalıklarının tanınması bağlamındaki üstün performansı nedeniyle GoogleNet ile birlikte CNN modeli oluşturulmuştur. Her durumda, örneklerin %80'i eğitim ve %20'si doğrulama için kullanılmıştır. Tüm görüntüler, GoogLeNet'in giriş boyutu gereksinimini (224×224x3 piksel) olarak sağlamıştır. Yani kısaca bu makale CNN ve Transfer learning kullanılarak bitki 16 hastalıklarının sınıflandırılmasının zor olduğu problemleri çözmek adına çok fayda sağlamıştır. Gelecek çalışmaları kolaylaştırılmak adına teorik bilgiler sunmaktadır.

Aravind ve Arkadaşları (Rangarajan & Purushothaman, 2018) tarafından bu çalışma kapsamında domates mahsulündeki geniş çaplı hastalık prevalansı, üretim kalitesini ve miktarının etkilenmesi gibi konuları ele almışlardır. Bu amaçla hızlı, güvenilir ve tahribatsız bir yöntem kullanarak sorunu erken teşhis etmeyi hedefleyerek çiftçilere kolaylık sağlamak istemektedirler. Bu çalışmada Plantvillage veri setinden elde edilen domates yapraklarının (6 hastalık ve bir sağlıklı sınıf) görüntüleri AlexNet ve VGG16 net olmak üzere iki derin öğrenme tabanlı mimariye girdi olarak sağlanmıştır. Sayısının rolü hiperparametrelerin görüntüleri ve önemi, yani minibatch boyutu, ağırlık ve önyargı öğrenme oranı sınıflandırma doğruluğu ve uygulama süresi analiz edilmiştir. Bu analiz sonucunda ise AlexNet ve VGG16 net kullanılarak elde edilen sınıflandırma doğruluğu sırasıyla % 97.49 ve % 97.23 olarak bulgulanmıştır. 13.262 görüntüden oluşan araştırmanın ikinci bölümünde modelin performansı, sayıları değiştirilerek değerlendirilmiştir. Görüntüler ve hiperparametre üzerinde, Hiperparametre minibatch boyutu, ağırlığı ve önyargı öğrenme oranı performans doğruluğunu gibi bazı değişiklikler yapılarak, 10 epoch doğrultusunda bu analizlerin sonuçları analiz edilmiştir.

Konstantinos P. Ferentinos (Ferentinos, 2018) tarafından yapılan bu çalışmada ise, derin öğrenme metodolojileri aracılığıyla sağlıklı ve hastalıklı bitkilerin basit görüntülerini kullanarak Evrişimli sinir ağı modelleri, bitki hastalıklarının tespit ve teşhisi için geliştirilmiştir. Modellerin eğitimi, sağlıklı bitkiler de dahil olmak üzere 58 farklı bitki ve hastalık kombinasyonundan oluşan bir sette 25 farklı bitki içeren 87,848 görüntüden oluşan açık bir veritabanı kullanılarak gerçekleştirilmiştir. Birkaç model mimarileri eğitilmiş ve en iyi performans, karşılık gelen [bitki, hastalık] kombinasyonu gözlemlenmiştir. Dataset, %70'e %30 olacak şekilde, eğitim ve test olacak biçimde ayrılmıştır. Geliştirilmiş olan bu özel derin öğrenme modelleri, belirli evrişimli sinir ağları mimarilerine dayalı olarak, bitki hastalıklarının basit yaprak sağlıklı görüntülerle tanımlanması veya hastalıklı bitkiler üzerinde eğitime başlanmıştır. Modellerin eğitimi, bir ekim alanlarında hem laboratuvar koşullarında hem de

gerçek koşullarda çekilmiş 87,848 fotoğrafın açık veri tabanıyla sağlanmıştır. En başarılı bir VGG evrişimli sinir ağı olan model mimarisi, % 99.53 başarı oranında sonuç veren model olmuştur. Önemli ölçüde sağlanan bu yüksek başarı oranı, gerçek yetiştirme koşullarında çalışmak için entegre bir bitki hastalığı tanımlama sistemi geliştirilebilmesi açısından gelecekteki çalışmalara kaynaklı oluşturmaktadır.

Gıda güvenliği ve bitkisel hastalıkların tanısı üzerinde durulan bu makale çalışmasında, Parul ve diğer iki arkadaşı (Sharma et al., 2019), S-CNN modeli eğitmek için çalışmalar ve araştırmalar yapmışlardır. Makale daha önce yapılmış olan %93'lük bir accuracy'ye sahip olan çalışmayı daha da iyi hale getirilmesi hedeflenmiştir. Bu makale kapsamı için geliştirilen model okutulan görüntülerin 256x256 piksel rastgele parçaları çıkarılması ve gürültü, bozulma, çevirme veya döndürme dönüşümleri uygulanmasıyla oluşturulmuştur. Adım uzunluklarını kontrol ederek, maskelerin boyutları, çoklu kıvrımlar ve havuzlama adımları uygulanmıştır. Model Python ve TensorFlowTM kullanılarak uygulanmaktadır. Oluşturulan bu S-CNN modelinin performansını test etmek için 10 sınıflı domates bitki tipi seçilmiştir. Eğitim ve sınıflandırma görüntüleri (toplam 17929) yerel olarak çiftliklerden (637) kaynaklanmış, Plantvillage veritabanından (16579) alınmış ve İnternet'ten (713) taranmıştır. Plantvillage Veri setindeki görüntüler için görüntü segmentasyonu, parlaklık / kontrast / bulanıklığı değiştirmek için çıkarılan görüntülere rastgele dönüşümler uygulamak ve arka planı aydınlatmak gibi bazı işlemler uygulanmıştır. Sonuç olarak eğitilen bu model, 10 hastalık sınıfına sahip modellerde bile daha önce görülmeyen bağımsız veriler üzerinde test edildiğinde performansta iki kattan fazla %98.6 hassasiyete kadar eğitilmiş ve doğru sonuçlar alınmıştır.

Artzai ve arkadaşları (Picon et al., 2019) tarafından yapılan bu çalışma kapsamında gerçek alanlardan telefon kameralarıyla çekilmiş yüzbinden fazla veriseti ile birlikte CNN'in görüntü sınıflandırma üzerindeki üstün yeteneği kullanmışlardır. Kullanılan bu veriseti, aynı resimde çeşitli hastalıkların olabileceği on yedi hastalığın ve beş mahsulün (buğday, arpa, mısır)neredeys e eşit olarak dağılmış hastalık aşamalarını içermektedir. Bu çalışma kapsamında, görüntü tabanlı Konvolüsyonel Sinir Ağına (CNN) kırpma bilgisi gibi bağlamsal görüntü olmayan meta veriler içeren üç farklı CNN mimarisi önerilmiştir. Bu, hastalık sınıflandırma görevlerinin karmaşıklığını azaltırken, tüm çoklu mahsul veri kümesinden aynı anda öğrenmenin avantajlarını birleştirmektedir. Tüm mahsul ve hastalıkları içeren tek bir modelin kullanılmasının, her mahsul için bağımsız sınıflandırma modelleri üretmekten biraz daha iyi sonuçlar elde ettiğini kanıtlamışlardır. Önceki tüm yöntemleri iyileştirerek ve önceki

yöntemlerin yanlış sınıflandırmalarının %71'ini kaldırarak %98'lik dengeli bir doğruluk elde edilmiştir.

Darwish ve Ezzat (Darwish et al., 2020), Sağlıklı ve sağlıksız yaprak görüntülerini sınıflandırarak görev bitkisi hastalığı teşhisi için VGG16 ve VGG19 olmak üzere önceden eğitilmiş iki kıvrımlı sinir ağının (CNN) bir topluluk modelinin geliştirilmesi hedeflenmiştir. 17 Bu makale kapsamında kullanılan Dataseti, domates, portakal ve mısır gibi farklı bitkilerin enfekte ve sağlıklı yapraklarının yaklaşık 174.000 görüntüsünü içermektedir. Bu yazıda sadece mısır bitkilerinin görüntüleri kullanılmıştır. Bu çalışma kapsamında sadece mısır bitkilerinin kullanılmasından ötürü 12.332 veri train için, 3076 veri de test için kullanılmıştır. Enfekte mısır yapraklarını temsil eden üç kategori ve sağlıklı yaprakları temsil eden bir kategori sınıf etiketi oluşturulmuştur. Manuel deneme ve hata yöntemi gibi geleneksel yöntemler kullanmak yerine bu hiperparametreler için en uygun değerleri bularak bu hiperparametrelerin bir kısmını optimize etmek için dikey öğrenme parçacık sürüsü optimizasyonu (OLPSO) algoritması kullanılmıştır. CNN'lerin yerel minimum seviyeye düşmesini önlemek ve verimli bir şekilde eğitmek için katlanarak azalan bir öğrenme oranı (EDLR) şeması kullanılmaktadır. Sonrasında ise önceden oluşturulmuş olan VGG16 ve VGG19 modelleri arasında yapılan kıyaslamalarda çok daha iyi sonuçlar verdiği gözlenmiştir.

Kairoldayeva (Kairoldayeva et al., 2019), Tarbil adlı veritabanı tarafından sağlanan gerçek zamanlı fotoğraf verilerinden oluşan dataset üzerinde çalışılmakta olup, arpa, buğday, ayçiçek, mısır, pamuk ve nohut olmak üzere toplamda 9000 adet veri yer alan 6 tür bitkinin sınıflandırılması hedeflenmektedir. Diğer bir amaç ise, Bitkinin sadece türünü bulmak değil, aynı zamanda o bitkinin hangi fenoloji aşamasında olduğunu tanımlamaktır. Bu çalışma kapsamında CNN derin öğrenme mimarisi, normal doku, renk, şekil vb. özelliklerine dayanan nesne tanıma ya da algılama için kullanılan ortak tekniklerden daha iyi performans gösterme potansiyeli olduğu tespit edilmiş ve CNN de model geliştirilmesine karar verilmiştir. CNN mimarisinin yoğun dataya ihtiyaç duymasından ötürü data augmentation işlemi yapılarak datalar çoğaltılmıştır. Mobilenet sürüm 1, Mobilenet sürüm 2 ve VGG-16 modelleri üzerinde kıyaslamalar yapılmıştır. TARBIL veri setinden 6 bitki sınıfı kullanılarak CNN'lerin ince ayar ve özellik çıkarımları yapılmıştır. Her görüntü büyük parçalara bölünüp ve her yama için özellik çıkarımı yapılmıştır. İlk olarak, görüntü veri üretici kullanmadan sonuçlar elde edilmiş olup, doğruluk%79.69 ve kayıp 0.4685 olarak elde edilmiştir. Özel histogram eşitlemesi

eklendiğinde, görüntü veri üreticisine karşılık olarak stretching işlevleri, sonuçları MobileNet v2 ile %87.5 ve 0.3798'e yükselmiştir.

WeiXue ve arkadaşları (W. Xue et al., 2019), Tarımsal alanda oluşan atığın kompostolanmasının kaynakları geri dönüştürmenin en iyi yollarından birisi olduğu savunulmaktadır. Problem tanımı olarak bunların belirlenmesi sağlayan biyokimyasal testlerin uzun zaman alması olarak gösterilmiştir. Görüntülerin analiz edilmesi işlemiyle, bunları çok 18 daha kısa sürede yapılması amaçlanmaktadır. Bu çalışma kapsamında kullanılmak üzere yaklaşık 30.000 görüntü içeren 4 veri seti oluşturmak için 3 farklı kompost malzemesinin görüntüleri toplanmıştır. Veri kümelerindeki her görüntünün boyutu 300×300 piksel olarak ayarlanmış ve sonrasında ise bu data seti için ResNet modeli geliştirilmiş olup diğer yöntemler olan, Alex, VGG16, VGG19 ve gibi modellerde ayrı ayrı kıyaslanmıştır. 3000 epoch kullanılan bu train işleminde, hazırlanan 4 data seti içinde en yüksek başarıyı ResNet sağlamıştır.

Aydın, Ali ve arkadaşları (Kaya et al., 2019) tarafından, Bitki türlerinin sınıflandırmasını derin öğrenme yaklaşımları kullanarak otomatize etmek istemektedir. Bu 14 çalışma içerisinde halka açık dört farklı dataset kullanılmıştır. Flavia, SwedishLea, UCI Yaprak veri kümesi ve diğer çalışmalarda da kullanılan Plantvillage data seti kullanılmıştır. Toplamda 57.781 adet dataya sahip olan bu veri setleri, diğer derin öğrenme yöntemleri olan AlexNet, VGG16 network gibi modellerin incelenmesinin ardından bir model önermişlerdir. 100 epochtan oluşan bu modelin eğitim aşaması, %70 train, %30 test olarak ayrılmıştır. Aynı zamanda bu çalışma bir sonraki çalışmalara faydası olması amaçlandığından dolayı diğer yöntemler tarafından da tek tek test edilmiştir. Derin öğrenme içerisinde Transfer öğrenmeden de deneyim sağlanmıştır. Uçtan uca CNN modeli üzerinde denenen bu transfer learning kavramının modele başarılı olduğu gözlemlenmiştir.

Muhammad ve arkadaşları (Saleem & Potgieter, 2020) tarafından yapılan bu çalışmada; Halka açık oluşturulan veri kümelerindeki (DL) mimarileri, Bitki hastalıklarının iki aşamada sınıflandırılması için derin öğrenmeye dayalı karşılaştırmalı değerlendirme yapılması önerildi. İlk olarak, en iyi evrişimli sinir ağı (CNN), bir karşılaştırma yapılarak elde edildi. İyi bilinen CNN mimarileri ile geliştirilmiş ve hibrit versiyonlar arasında analizler gerçekleştirilmiştir. İkinci olarak, performansı en iyi elde edilen model, çeşitli derin öğrenme optimize ediciler aracılığıyla eğitim yoluyla geliştirilmeye çalışılmıştır. Çeşitli CNN'ler arasındaki karşılaştırma, doğrulama gibi performans ölçütlerine dayanmaktaydı ve doğruluk / kayıp, F1 skoru ve gerekli dönem sayısı gibi değerlendirme metrikleri üzerinde duruldu. Seçilen tüm DL mimarileri 14'e

ait 26 farklı hastalığı ve bitki türleri içeren Plantvillage veri setinde eğitilmiştir. TensorFlow arka uçlu Keras, derin öğrenme mimarilerini eğitmek için kullanılmıştır. Bu Adam optimizer ile eğitilen Xception mimarisinin en yüksek seviyeye ulaştığı sonucuna varılmıştır. Doğrulama doğruluğu ve F1-skoru sırasıyla %99.81 ve 0.9978, ki bu nispeten daha iyi önceki yaklaşımlardan daha fazla ve etkili olduğunu kanıtlamıştır.

Ahmet Saygılı (Irmak & Saygili, 2020) tarafında yapılan bu çalışmada hastalıklı ve sağlıklı domates bitkisi yaprakları üzerinde sınıflandırma çalışması gerçekleştirilmiştir. Çalışmada Plantvillage datasetinde yer alan domates bitkilerinden yeni bir dataset oluşturulmuştur. Bu oluşturulan veri setinde, 9 çeşit domates hastalığı ile sağlıklı olma durumu her classta veri setinde her bir sınıf için yaklaşık 2000 görüntü, toplamda 18345 görüntü yer almaktadır. Çalışmamızda her bir sınıftan 200 örnek seçilerek toplam 2000 örnek görüntü ile sistem gerçekleştirilmiştir. Uygulanan modelde bu veri setinin %20'si test için geri kalan kısmı da eğitim için kullanılmıştır. Eğitim için CNN'in Sequential modeli kullanılmış olup, model sonucunda; eğitim için seçilmiş örnek sayısı 1600, test için seçilmiş örnek sayısı 400 olmuştur. Her bir epoch (eğitim turu) ortalama 21 saniye sürmüştür. Test Kaybı: %9.33, Test Doğruluğu: %97.05 olarak hesaplanmıştır.

Shreyas ve arkadaşları (Labhsetwar et al., n.d.) yaptıkları araştırma sonucunda, zararlılar ve hastalıklar nedeniyle mahsul kıtlığı Hint tarımının doğasında olup, yıllık % 15-25 oranında verimlilik kaybı, ekonomik kayıp gibi konuları ele almışlardır. Bu araştırma tesisin hastalıklar tahmine dayalı analizi için çeşitli optimize edicilerden derin öğrenme yaklaşımları ile performansları analiz etmektedir. Araştırma kapsamında Evrişimli Sinir Ağlarını kullanılmaktadır. 3 farklı mahsulün bitki yaprak örneklerinin sınıflandırılması yer almaktadır ve toplamda 15 sınıftan oluşmaktadır. Bu çalışmada RMSprop, Adam ve AMSgrad gibi çeşitli optimize ediciler kullanılmıştır. En iyi performans Adam optimizer kullanılarak 28 katmandan oluşan CNN Sequential modeli ile % 98' doğruluk olarak elde edilmiştir. İkinci sırada ise RMSprop optimizer %95 doğruluk ile yer almaktadır. Bu makale çalışmasının amacı daha çok araştırmaya odaklanmaktır. Bitki hastalıklarının öngörülebileceğini kanıtlayan analiz ve derin öğrenme metodolojisi kullanılarak önceden boşaltılmış uydu, drone tabanlı veya mobil tabanlı yardım mahsul hatalarının azalmasına neden olan görüntüler ve tarımsal kayıplar gibi konular için gelecekteki çalışmalara kaynaklık edeceği hedeflenmiştir.

Sue Han Lee ve arkadaşları (Lee et al., 2020), Öncelikle genel bir nesne tanıma görevi yerine bir bitki tanımlama görevinde önceden eğitilmiş bir modelin ince ayarının yapılmasının

15 daha uygun olup olmadığını incelemiştir. Özellikle, görselleştirme teknikleri aracılığıyla, öğrenilen özelliklerin benimsenen yaklaşıma göre farklılaştığını ve mutlaka hastalıktan etkilenen kısma odaklanmadığını göstermektedir. Bu nedenle, hastalıkları ekinlerden bağımsız olarak ele alan daha sezgisel bir yöntem sunmuşlardır ve özellikle eğitim veritabanında gösterilmeyen ekinler içeren hastalıklarla uğraşırken klasik ekin-hastalık çifti yaklaşımından daha etkili olduğunu gösterilmiştir. Plantvillage adlı popüler veri seti kullanılmış ve tek tip bir arka plan üzerinde izole sağlıklı ve enfekte yapraklar içermektedir. Bu data seti 14 ekin bitkisi için 26 ekin hastalığı kategorisine sahip 38 ekin hastalığı çiftine sahiptir. Dataseti %80 eğitim ve %20 test verileri olarak ayrılmıştır. Transfer Learning modeli geliştirilerek VGG16, GoogleNetBN gibi mimarilerle karşılaştırılmıştır.

Amren Abbas ve arkadaşları (Abbas et al., 2021) yaptıkları bu çalışmada bitki hastalıkları ve zararlı böceklerin tarım üzerindeki etkilerini değerlendirerek erken tanı sayesinde faydasına kullanabilmeyi amaçlamışlardır. Çalışmaya başlamadan önce Plantvillage halka açık datasetinde yer alan domates türlerine ait yaprak verilerinin yanı sıra sentetik verilerle bunları çoğaltarak eğitime hazır hale getirmişlerdir. Sentetik verileri de C-Gan koşullu üretken ağı sayesinde gerçekleştirmişlerdir. Eğitim aşamasında kullanmak üzere transfer öğrenme yöntemiyle DenseNet121 modelini yeniden düzenlemişlerdir. Eğitim ve test aşamasında 5 sınıf, 7 sınıf ve 10 sınıf olmak üzere verisetini eğitime sokmuşlar ve sırasıyla, %99,51, %98,65 ve %97,11'lik başarımlar elde etmişlerdir.

Punam Bedi ve arkadaşları (Bedi & Gole, 2021), diğer literatürlerde olduğu gibi hızlı teşhisin önemini baz alarak bitki hastalıklarının tahminlenmesindeki hayati önem üzerinde yoğunlaşmıştır. Bu çalışmada, Otomatik bitki hastalık tespiti için Evrişimli Sinir Ağı (CNN) ve Convolutional Autoencoder (CAE) ağına dayalı yeni bir hibrit model önerilmektedir. Bilindiği üzere, bitki hastalıklarını otomatik olarak tespit etmek için CAE ve CNN'ye dayalı bir hibrit sistem, literatürde bulunan en son teknoloji sistemlerde önerilmemektedir. Bu çalışmada, önerilen hibrit model, Bakterileri tespit etmek için uygulanmıştır. Şeftali bitkilerinin yaprak görüntülerini kullanarak leke hastalığı saptanmaktadır ve diğer herhangi bir bitki hastalığı tespiti için de kullanılabilir. Bu yazıda gerçekleştirilen deneyler, yaprağı elde etmek için Plantvillage adlı halka açık bir veri kümesinde yer alan şeftali bitkilerinin görüntüleri kullanılmasıyla gerçekleştirilmiştir. Önerilen sistem 9,914 eğitim parametresi kullanarak %99,35 eğitim doğruluğu ve %98,38 test doğruluğu elde etmiştir

1.2 Çalışmanın Amacı ve Kapsamı

Bu tez çalışması kapsamında amacımız, Plantvillage halka açık veriseti aracılığı ile yeni oluşturduğumuz büyük ölçekli ve çoklu sınıflı veri setlerini kullanarak, başta bitki türlerinin sınıflandırılması olmak üzere, hastalık tahmini ve bitki hastalıklarının tanısı gibi problemlerin çözümüne yönelik derin öğrenme algoritmalarının kullanıldığı modeller oluşturmaktır. Bu amaçla, derin öğrenme algoritmalarından, AlexNet, VGG16, VGG19, GoogleNet, DenseNet, ResNet50, MobileNet, Xception ve NasNetMobile modelleri kullanılmıştır.

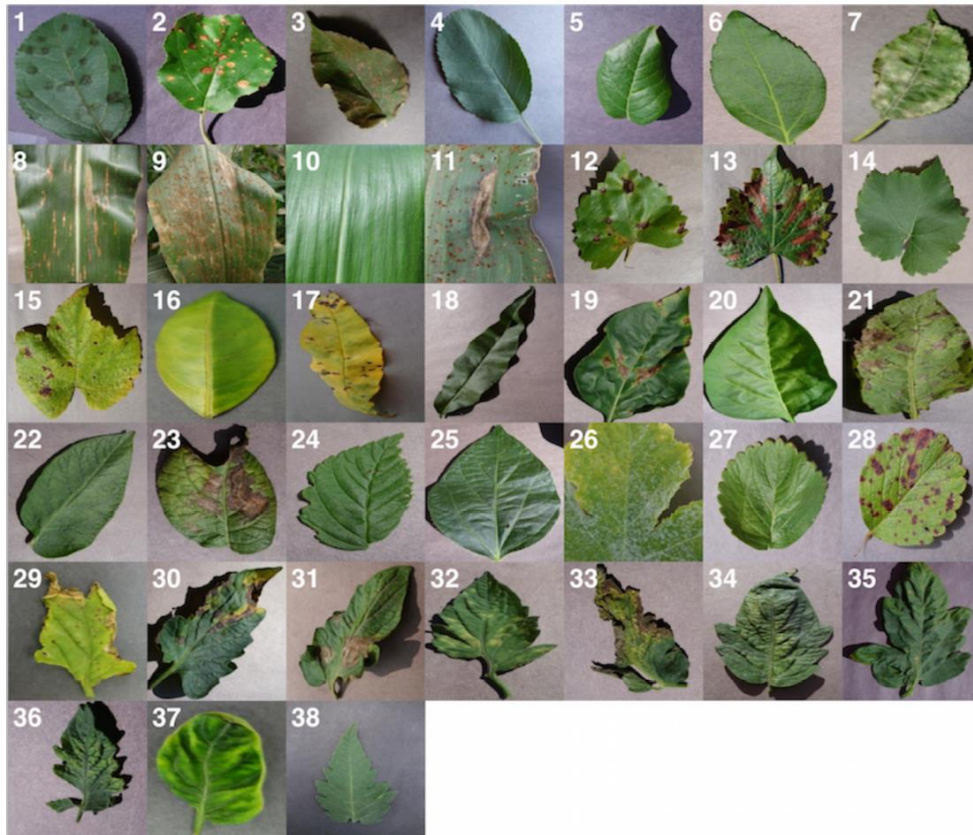
Bu bölümünün devamında bahsi geçen konuların detaylı açıklanmasının ardından, ikinci bölümünde, bitki türlerinin sınıflandırılması, hastalık tahmini ve bitki hastalıklarının tanısı ile ilgili kaynak özetleri, üçüncü bölümünde çalışmada kullanılan veri setleri, bu veri setlerinin oluşturulması, modellerin belirlenmesi, sınıflandırma algoritmaları, değerlendirme ölçütleri, çalışmada kullanılan kütüphaneler, dördüncü bölümünde araştırma bulguları ve beşinci bölümünde ise tartışma ve sonuçlar ele alınmıştır.

2. MATERYEL VE YÖNTEM

Bu bölümde, çalışmada kullanılan orijinal veri setinin (Plantvillage) tanıtımına, ve bu veri setinin, bitki türlerinin sınıflandırılması, bitki hastalık tahmini ve bitki hastalıklarının sınıflandırılması problemleri için yeniden düzenlenmesiyle oluşturulan yeni veri setlerinin detaylarına, ve yeni oluşturulan bu veri setlerinin derin öğrenme mimarilerine uygulanmasıyla oluşturulan modelleme çalışmalarına ve bu modellerin sonuçlarının değerlendirilmesine yer verilmiştir.

2.1 Orjinal Veri Seti Tanıtımı

Bu tez çalışmasında, halka açık olarak paylaşılan Plantvillage veri seti referans alınmıştır (Hughes & Salathe, 2015). Bitki yapraklarının görüntülerinden oluşan bu veri tabanı 14 bitki türü için sağlıklı ve hastalıklı olmak üzere toplam 38 sınıftan oluşmakta olup toplam 54.305 yaprak görüntüsü içermektedir. Görüntüler RGB renk alanında ve 256×256 piksel boyutlarındadır (Hernández & López, 2020). Her bir sınıfa ait örnek görüntülere Şekil 2.1'de gösterilmektedir (Mohanty et al., 2016).



Şekil 2.1. Plantvillage veri seti her sınıf örnek gösterimi

Şekil 2.1’de gösterilen her bir sınıf şu şekilde isimlendirilmektedir: (1) Elma kara leke hastalığı, (2) Elma siyah rot hastalığı, (3) Elma pas mantar hastalığı, (4) Elma sağlıklı, (5) Yaban mersini sağlıklı, (6) Kiraz sağlıklı, (7) Kiraz küllenme hastalığı, (8) Mısır gri yaprak lekesi hastalığı, (9) Mısır pas hastalığı, (10) Mısır sağlıklı, (11) Kuzey mısır yaprak yanıklığı hastalığı, (12) Üzüm siyah çürük hastalığı, (13) Üzümlerde siyah kızamık hastalığı, (14) Üzüm sağlıklı, (15) Üzüm yaprak yanıklığı hastalığı, (16) Portakal sağlıklı, (17) Şeftali bakteri lekesi, (18) Şeftali sağlıklı, (19) Biber bakteri lekesi, (20) Biber sağlıklı, (21) Patates yaprak yanıklığı, (22) Patates sağlıklı, (23) Patates geç yaprak yanıklığı, (24) Ahududu sağlıklı, (25) Soya sağlıklı, (26) Kabak küflenme hastalığı, (27) Çilek sağlıklı, (28) Çilek yaprağı kavrulması, (29) Domates bakteri lekesi, (30) Domates yaprak yanıklığı, (31) Domates geç yaprak yanıklığı, (32) Domates yaprak küfü, (33) Domates Septorya yaprak lekesi hastalığı, (34) Domates iki noktalı kırmızı örümcek hastalığı, (35) Domates yaprak lekesi hastalığı, (36) Domates Mozaik virüsü, (37) Domates sarı yaprak kıvrıcıklığı virüsü, (38) Domates sağlıklı (Mohanty et al., 2016).

Çizelge 2.1’de Plantvillage veri setinin içerik bilgileri yer almaktadır. Bu çizelgede de belirtildiği gibi, veri setinde ‘Elma’, ‘Yabanmersini’, ‘Kiraz’, ‘Mısır’, ‘Üzüm’, ‘Portakal’, ‘Şeftali’, ‘Biber’, ‘Patates’, ‘Ahududu’, ‘Soya’, ‘Kabak’, ‘Çilek’ ve ‘Domates’ olmak üzere 14 bitki türüne ait sağlıklı ve hastalıklı örnek görüntüler bulunmaktadır. Veri setinde, ‘Elma’ için, 1 sağlıklı ve 3 hastalıklı sınıf; ‘Yabanmersini’ için sadece 1 sağlıklı sınıf; ‘Kiraz’ için, 1 sağlıklı ve 1 hastalıklı sınıf; ‘Mısır’ için 1 sağlıklı ve 3 hastalıklı sınıf; ‘Üzüm’ için 1 sağlıklı ve 3 hastalıklı sınıf; ‘Portakal’ için sadece 1 sağlıklı sınıf; ‘Şeftali’ için 1 sağlıklı ve 1 hastalıklı sınıf; ‘Biber’ için 1 sağlıklı ve 1 hastalıklı sınıf; ‘Patates’ için 1 sağlıklı ve 2 hastalıklı sınıf; ‘Ahududu’ için sadece 1 sağlıklı sınıf; ‘Soya fasülyesi’ için sadece 1 sağlıklı sınıf; ‘Kabak’ için sadece 1 hastalıklı sınıf; ‘Çilek’ için, 1 sağlıklı ve 1 hastalıklı sınıf; ‘Domates’ için de 1 sağlıklı ve 9 hastalıklı sınıf olmak üzere toplam 38 sınıf bulunmaktadır. Her bitki türünün, alt sınıfları sağlıklı ve hastalık türü olmak üzere farklı sayıda örnek içermektedir ve her sınıftaki örnek sayılarının da birbirinden farklı olduğu Çizelge 2.1’de görülmektedir. Her sınıftaki örneklerin farklı sayıda olması, bu veri setinin dengesiz bir veri seti olduğu anlamına gelmektedir.

Çizelge 2.1. Plantvillage Veri Seti detaylı gösterim

Bitki Türü No	Bitki Türü	Sınıf Sayısı	Sınıf Türü		Örnek Sayısı		Toplam Örnek Sayısı
1	Elma	4	Sağlıklı		1.645		3.171
			Hastalıklı	Kara leke hastalığı	621	1.526	
				Pas mantar hastalığı	275		
				Siyah rot hastalığı	630		
2	Yabanmersini	1	Sağlıklı		1.502		1.502
3	Kiraz	2	Sağlıklı		854		1.906
			Hastalıklı	Küllenme hastalığı	1.052		
4	Mısır	4	Sağlıklı		1.162		3.852
			Hastalıklı	Gri yaprak lekesi hastalığı	513	2.690	
				Pas hastalığı	1.192		
				Yaprak yanıklığı hastalığı	985		
5	Üzüm	4	Sağlıklı		423		4.062
			Hastalıklı	Siyah çürük hastalığı	1.180	3.639	
				Siyah kızamık hastalığı	1.383		
				Yaprak yanıklığı hastalığı	1.076		
6	Portakal	1	Sağlıklı		5.507		5.507
7	Şeftali	2	Sağlıklı		360		2.657
			Hastalıklı	Bakteri lekesi hastalığı	2.297		
8	Biber	2	Sağlıklı		1.478		2.475
			Hastalıklı	Bakteri lekesi hastalığı	997		
9	Patates	3	Sağlıklı		152		2.152
			Hastalıklı	Yaprak yanıklığı hastalığı	1.000	2.000	
				Geç yaprak yanıklığı hastalığı	1.000		
10	Ahududu	1	Sağlıklı		371		371
11	Soya Fasulyesi	1	Sağlıklı		5.090		5.090
12	Kabak	1	Küflenme hastalığı		1.835		1.835
13	Çilek	2	Sağlıklı		456		1.565
			Hastalıklı	Yaprağı kavrulması hastalığı	1.109		
14	Domates	10	Sağlıklı		1.591		18.160
			Hastalıklı	Bakteri lekesi hastalığı	2.127	16.569	
				Yaprak yanıklığı hastalığı	1.000		
				Geç yaprak yanıklığı hastalığı	1.909		
				yaprak küfü hastalığı	952		
				Mozaik virüsü	373		
				Septorya yaprak lekesi hastalığı	1.771		
				İkinoktalı kırmızı örümcek hastalığı	1.676		
				Yaprak lekesi hastalığı	1.404		
Sarı yaprak kıvrıcıklığı virüsü	5.357						
Toplam		38					54.305

2.1.1 Bitki Türü Sınıflandırılması İçin Yeni Oluşturulan Veri Setleri

Bitki türü sınıflandırılması için, orijinal veri setindeki sağlıklı bitki türlerine ait örnekler seçilmiştir. Çizelge 2.1'deki orijinal veri setinin her bir bitki türüne ait, sağlıklı bitki sınıfları, eşit sayıda örnek içerecek şekilde yeniden düzenlenerek 6 tane yeni veri seti oluşturulmuştur. D1-BTS, D2-BTS, ..., D6-BTS diye adlandırılan bu yeni oluşturulmuş dengeli veri setlerine ait, sınıf ve örnek sayıları Çizelge 2.2'de gösterilmiştir.

Çizelge 2.2. Bitki Türü Sınıflandırılması için oluşturulan veri setlerinin detaylı gösterimi

Veri Seti (Sağlıklı)	Sınıf Sayısı	Sınıf Adı	Sınıf Başına Örnek Sayısı	Toplam Örnek
D1-BTS	7	Apple	1.162	8.134
		Blueberry	1.162	
		Corn	1.162	
		Orange	1.162	
		Pepper	1.162	
		Soybean	1.162	
		Tomato	1.162	
D2-BTS	6	Apple	1.478	8.868
		Blueberry	1.478	
		Orange	1.478	
		Pepper	1.478	
		Soybean	1.478	
		Tomato	1.478	
D3-BTS	5	Apple	1.502	7.510
		Blueberry	1.502	
		Orange	1.502	
		Soybean	1.502	
		Tomato	1.502	
D4-BTS	4	Apple	1.591	6.364
		Orange	1.591	
		Soybean	1.591	
		Tomato	1.591	
D5-BTS	3	Apple	1.645	4.935
		Orange	1.645	
		Soybean	1.645	
D6-BTS	2	Orange	5.090	10.180
		Soybean	5.090	

Çizelge 2.2’de görüldüğü gibi, D1-BTS veri seti, 7 sınıf ve her bir sınıfta 1.162 örnek olmak üzere, toplam 8.134 örnek içermektedir. D2-BTS veri seti, 6 sınıf ve her bir sınıfta 1.478 örnek olmak üzere, toplam 8.868 örnek içermektedir. D3-BTS veri seti, 5 sınıf ve her bir sınıfta 1.502 örnek olmak üzere, toplam 7.510 örnek içermektedir. D4-BTS veri seti, 4 sınıf ve her bir sınıfta 1.591 örnek olmak üzere, toplam 6.364 örnek içermektedir. D5-BTS veri seti, 3 sınıf ve her bir sınıfta 1.645 örnek olmak üzere, toplam 4.935 örnek içermektedir. D6-BTS veri seti, 2 sınıf ve her bir sınıfta 5.090 örnek olmak üzere, toplam 10.180 örnek içermektedir.

2.1.2 Bitki Hastalık Tahmini İçin Yeni Oluşturulan Veri Setleri

Çizelge 2.1’deki orijinal veri setinin sağlıklı ve hastalıklı bitki sınıflarından, 10 bitki türü için, her bir bitki türüne ait, sağlıklı ve hastalıklı olmak üzere iki sınıftan oluşan ve her bir sınıfta eşit sayıda 7.510 örnek görüntü bulunan, toplam 15.020 adet görüntü içeren dengeli bir veri seti D7-BHT adıyla oluşturulmuştur. Yeni oluşturulan bu veri setinin içerik detayları Çizelge 2.3’de belirtilmiştir.

Çizelge 2.3. Bitki Hastalık Tahmini için oluşturulan dengeli D7-BHT veri seti

Bitki Türleri	Sağlıklı Örnek Sayısı	Sağlıksız Örnek Sayısı
Elma	1.045	1.045
Kiraz	854	854
Mısır	962	962
Üzüm	423	423
Şeftali	360	360
Dolma Biber	1.077	1.077
Patates	152	152
Soya	1.090	1.090
Çilek	456	456
Domates	1.091	1.091
Toplam Örnek Sayısı = 15.020		

2.1.3 Bitki Hastalıklarının Sınıflandırılması İçin Yeni Oluşturulan Veri Setleri

Çizelge 2.1'deki orijinal veri setinin hastalıklı bitki sınıflarından 25 sınıf için, her bir sınıfta 513 adet örnek görüntü seçilerek, toplam 12.825 görüntü içeren dengeli bir veri seti D8-BHS adıyla oluşturulmuştur. Yeni oluşturulan bu veri setinin içerik detayları Çizelge 2.4'de belirtilmiştir.

Çizelge 2.4. Bitki Hastalıklarının Sınıflandırılması için oluşturulan D8-BHS veri seti

Sınıf Adı	Her Bir Sınıftaki Örnek Sayısı
Elma kara leke hastalığı	630
Elma siyah rot hastalığı	621
Elma pas mantar hastalığı	550
Kiraz küllenme hastalığı	1.052
Mısır gri yaprak lekesi hastalığı	513
Mısır pas hastalığı	1.192
Kuzey mısır yaprak yanıklığı hastalığı	985
Üzüm siyah çürük hastalığı	1.180
Üzümlerde siyah kızamık hastalığı	1.383
Üzüm yaprak yanıklığı hastalığı	1.076
Şeftali bakteri lekesi	1.297
Biber bakteri lekesi	997
Patates yaprak yanıklığı	1.000
Patates geç yaprak yanıklığı	1.000
Kabak küflenme hastalığı	1.835
Çilek yaprağı kavrulması	1.109
Domates bakteri lekesi	1.127
Domates yaprak yanıklığı	1.000
Domates geç yaprak yanıklığı	908
Domates yaprak küfü	952
Domates Septorya yaprak lekesi hastalığı	1.771
Domates ikinoktalı kırmızı örümcek hastalığı	1.676
Domates yaprak lekesi hastalığı	1.404
Domates Mozaik virüsü	746
Domates sarı yaprak kıvrıcıklığı virüsü	1.357
Toplam Sınıf = 25	27.361

2.2 Veri Setinin Bölünmesi

Bu çalışmada, bitki türlerinin sınıflandırılması, bitki hastalık tahmini ve bitki hastalıklarının sınıflandırılması için yeni oluşturulan veri setlerine, 3, 5 ve 10 kez çapraz doğrulama uygulanmıştır.

3 kez çapraz doğrulama ile veri seti, rastgele olarak 3 eşit parçaya bölünmekte ve sonrasında 3 farklı aşamada sırasıyla bu parçalardan biri test veri seti, diğer 2 parça da eğitim veri seti olarak kullanılmaktadır. Bu sayede her bir alt küme en az bir kez test amacıyla kullanılmış olmaktadır (Kalaycı, 2018). Sınıflandırma işlemi de ayrılan her bir eğitim ve test verisi için 3 kez tekrarlanmış olmaktadır. Bu işlem aynı şekilde 5 ve 10 kez çapraz doğrulama için de yapılmıştır. Şekil 2.2, 2.3 ve 2.4'te sırasıyla 3, 5 ve 10 kez çapraz doğrulama işleminin nasıl yapıldığı gösterilmiştir.

1. Kez	Test Seti	Eğitim Seti	Eğitim Seti
2. Kez	Eğitim Seti	Test Seti	Eğitim Seti
3. Kez	Eğitim Seti	Eğitim Seti	Test Seti

Şekil 2.2. 3 Kez Çapraz Doğrulama

1. Kez	Test Seti	Eğitim Seti	Eğitim Seti	Eğitim Seti	Eğitim Seti
2. Kez	Eğitim Seti	Test Seti	Eğitim Seti	Eğitim Seti	Eğitim Seti
3. Kez	Eğitim Seti	Eğitim Seti	Test Seti	Eğitim Seti	Eğitim Seti
4. Kez	Eğitim Seti	Eğitim Seti	Eğitim Seti	Test Seti	Eğitim Seti
5. Kez	Eğitim Seti	Eğitim Seti	Eğitim Seti	Eğitim Seti	Test Seti

Şekil 2.3. 5 Kez Çapraz Doğrulama

1. Kez	Test Seti	Eğitim Seti	Eğitim Seti	Eğitim Seti	Eğitim Seti	Eğitim Seti	Eğitim Seti	Eğitim Seti	Eğitim Seti	Eğitim Seti
2. Kez	Eğitim Seti	Test Seti	Eğitim Seti	Eğitim Seti	Eğitim Seti	Eğitim Seti	Eğitim Seti	Eğitim Seti	Eğitim Seti	Eğitim Seti
3. Kez	Eğitim Seti	Eğitim Seti	Test Seti	Eğitim Seti	Eğitim Seti	Eğitim Seti	Eğitim Seti	Eğitim Seti	Eğitim Seti	Eğitim Seti
4. Kez	Eğitim Seti	Eğitim Seti	Eğitim Seti	Test Seti	Eğitim Seti	Eğitim Seti	Eğitim Seti	Eğitim Seti	Eğitim Seti	Eğitim Seti
5. Kez	Eğitim Seti	Eğitim Seti	Eğitim Seti	Eğitim Seti	Test Seti	Eğitim Seti	Eğitim Seti	Eğitim Seti	Eğitim Seti	Eğitim Seti
6. Kez	Eğitim Seti	Eğitim Seti	Eğitim Seti	Eğitim Seti	Eğitim Seti	Test Seti	Eğitim Seti	Eğitim Seti	Eğitim Seti	Eğitim Seti
7. Kez	Eğitim Seti	Eğitim Seti	Eğitim Seti	Eğitim Seti	Eğitim Seti	Eğitim Seti	Test Seti	Eğitim Seti	Eğitim Seti	Eğitim Seti
8. Kez	Eğitim Seti	Eğitim Seti	Eğitim Seti	Eğitim Seti	Eğitim Seti	Eğitim Seti	Eğitim Seti	Test Seti	Eğitim Seti	Eğitim Seti
9. Kez	Eğitim Seti	Eğitim Seti	Eğitim Seti	Eğitim Seti	Eğitim Seti	Eğitim Seti	Eğitim Seti	Eğitim Seti	Test Seti	Eğitim Seti
10. Kez	Eğitim Seti	Eğitim Seti	Eğitim Seti	Eğitim Seti	Eğitim Seti	Eğitim Seti	Eğitim Seti	Eğitim Seti	Eğitim Seti	Test Seti

Şekil 2.4. 10 Kez Çapraz Doğrulama

2.3 Görüntü Sınıflandırma

Görüntü sınıflandırma, bilgisayarlı görme ve örüntü sınıflandırmanın en önemli dallarından birisidir. Özellikle son yıllarda bu kavram başta derin öğrenme olmak üzere birçok alanda daha da önem kazanmaktadır (Qin et al., 2019).

Tanımının da bilindiği üzere, resim verileri nesnelere gözle görülür görünümüdür. İnsanlar, nesnelere karşılık gelen yüksek çözünürlüklü fotoğrafları tercih etmektedirler. Özellikle de görmek istedikleri detaya bağlı olarak bu tercih meselesi önem kazanmaktadır. Neredeyse her zaman insanlar yüksek çözünürlüklü görüntüleri tercih etmektedir. Bu gibi faktörler sayesinde, insanlar yüksek performanslı kameralar ve görüntü işleme yazılımları sayesinde çok çabalar vermektedir (Feng et al., 2018).

2.4 Derin Öğrenme

Hinton'un yapmış olduğu çalışmalarla yayınlamış olduğu makalede yapay sinir ağlarına yeni bir yaklaşım getirmiştir. Bu yaklaşım derin öğrenme (Deep Convolution Neural Network) olarak adlandırılmıştır (Geoffrey E. Hinton et al., 2006).

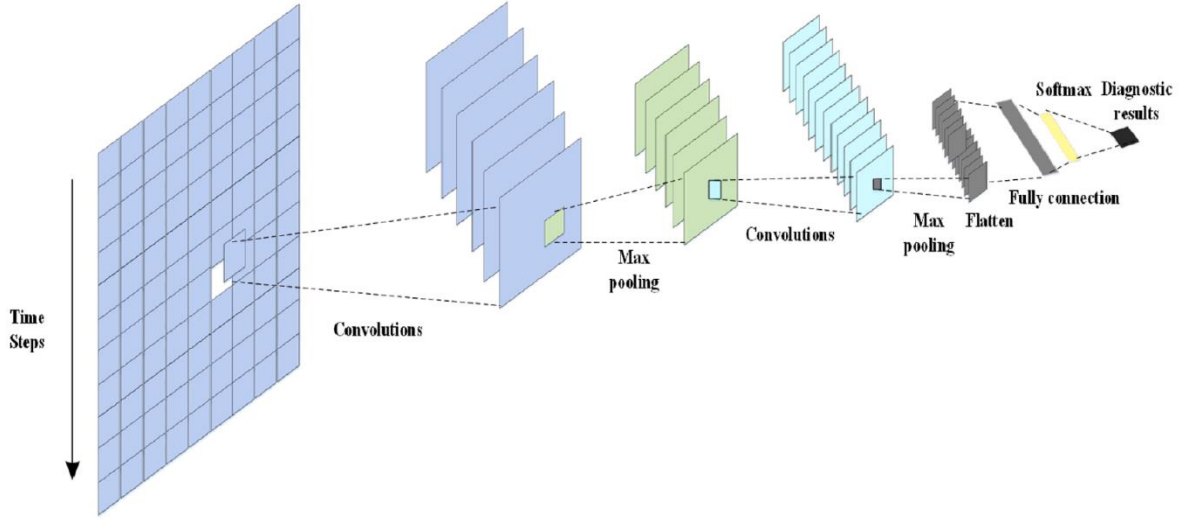
Derin öğrenme ile daha önce yapılan pek çok işlem bir arada yürütülerek sonuca gidilmektedir. Burada özellikle ön işlem ve özellik çıkarımı gibi yapılar göz ardı edilmekte ve sinir ağı içerisinde bu işlemler otomatik olarak yapılmaktadır. Derin konvolüsyonel sinir ağında özellik çıkarımı ağı içerisinde belirlenmekte ve katmanlar içerisinde tespit edilmesi istenen yapıya ait özellikler belirlenmektedir. Alt katman ile üst katman arasında bağlantılı hiyerarşik bir yapı bulunmaktadır. Özellik çıkarımı için özel bir safha bulunmamaktadır. Katmanlar içerisindeki yapısında nesne olaya ait belirgin özellikler belirlenmektedir (G. E. Hinton & Salakhutdinov, 2006).

2.5 Konvolüsyonel Sinir Ağ (CNN)

Yoğun nöral ağlardan oluşan etkili bir derinlik sınıfı olarak bilinmektedir. Bu ağlar esas olarak, görüntü verilerini tanıma, sınıflandırma gibi amaçlarla ortaya atılmış vebu amaçla modeller uygulanmıştır (Brodrick et al., 2019), (Krizhevsky et al., 2012). Son yıllarda ise etkili zaman serisi sınıflandırmaları gibi uygulamalarda da karşımıza çıkmaktadır (Zhao et al., 2017).

Çok katmanlı bu CNN ağı içerisinde birden fazla konvolüsyon katmanı, tam bağlı katman, aktivasyon katmanı, sınıflandırıcı katman, havuzlama katmanı ve bunlara ek katmanlar yer almaktadır. Her katman kendi işlevini yürüterek bir sonraki katmana 6 aktarmaktadır ve sınıflandırıcı katmanda da sonuç üretilmektedir. Derin öğrenme yapıları içerisinde en çok kullanılan sinir ağı konvolüsyonel sinir ağlarıdır. Daha çok sınıflandırma ve tespit işlemleri için kullanılmaktadır. Sinir ağı içerisindeki katmanlarla sınıflandırılacak öğelere ait öznitelikler belirlenerek sınıflandırıcı katmanı ile öğeler sınıflandırılır (DOĞAN & TÜRKÖĞLU, 2019).

Giriş verisine uygulanan her filtre sonucunda öznitelik değerleri elde edilmektedir. Havuz katmanı geçişinde öznitelik haritalarına ReLU aktivasyon fonksiyonu uygulanmaktadır. ReLU aktivasyon fonksiyonu diğer aktivasyon fonksiyonlarına göre daha iyi hesaplama süresi sağladığından diğerleriyle karşılaştırıldığında sınıflandırma problemlerinde daha çok tercih edilmektedir. Elde edilen öznitelik haritaları havuz katmanında bölgesel doku birleştirme işlemine tabi tutularak öznitelik haritalarının boyutları düşürülmektedir. Havuz katmanından sonra elde edilen öznitelikler tam bağlı katmana giriş olarak iletilmektedir. Tam bağlı katmanının çıkışında ise elde edilen sınıflandırmanın etiketlerine sahip olunmaktadır (Salur & Aydın, 2018).



Şekil 2.5. Konvolüsyonel Sinir Ağı yapısı

Şekil 2.5’de görüldüğü gibi CNN, birden fazla düzlem içeren çoklu bir katman yapısına sahiptir. Konvolüsyon (Convolution), Havuzlama (Pooling) ve Tam bağlı (Fully- Connected) katman olarak sıralanabilmektedirler. Konvolüsyon katmanında, her bir giriş için belirli özellikleri çıkaran bir dizi filtre uygulanmaktadır. Havuzlama katmanı, doğrusal olmayan alt örnekleme sağlarken, ağın öğrenmesi gereken parametre sayısını azaltarak çıkışı basitleştirmektedir. Bu sayede sadece aktive edilen özellikler bir sonraki katmana iletilmektedir. Bu özelliklerin öğrenilmesinin ardından sınıflandırma aşamasına geçilmektedir. Son katman c boyutlu tam bağlı bir katmandır ve her bir sınıfın olasılığı bu vektörde tutulmaktadır (Gu et al., 2006).

Giriş katmanı normalleştirilmiş girdileri içermektedir. Bir katmandaki her düzlem, önceki katman düzleminin küçük bir girdisini alır. Her bir düzlem, bir önceki katmandaki düzlem üzerinden yerel bir pencere ile taranarak elde edilen sabit bir özellik haritası olarak düşünülebilir. Böylece çoklu özellikler tespit edilebilir. Bir özellik tespit edildikten sonra, kesin konumu daha az öneme sahiptir. Bu nedenle, konvolüsyon katmanları tipik olarak bir yerel ortalama ve alt örnekleme işlemi yapan başka bir katman tarafından takip edilir. Sonunda, tüm piksel değerlerinin kombinasyonu bir vektöre dönüştürülür (LeCun et al., 1998). l katmanı ve k özellik haritasındaki (i, j) lokasyonunun değeri aşağıdaki şekilde hesaplanmaktadır.

$$Z_{i,j,k}^l = w_k^{lT} x_{i,j}^l + b_k^l \quad (1)$$

Burada w ve b sırasıyla l katmanı k özellik haritasındaki ağırlık vektörü ve bias’ı ifade etmektedir. Ortada yer alan x katmanı ise, (i, j) lokasyonunun ortalananmış giriş değeridir. w

ağırlık vektörünün ağ genelinde paylaşıldığı bilinmektedir. Böylece model karmaşıklığı azaltılırken, ağın daha kolay eğitilmesi gerçekleştirilmektedir. Bütün bunlarla birlikte aktivasyon fonksiyonu ile CNN'nin doğrusal olmayan özellikleri tanınması sağlanabilmektedir (Yıldız, 2019). Konvolüsyon özellik için aktivasyon değeri aşağıdaki şekilde hesaplanmaktadır.

$$a_{i,j,k}^l = a(Z_{i,j,k}^l) \quad (2)$$

CNN'de sıklıkla kullanılan tipik aktivasyon fonksiyonları; Sigmoid, Tanh ve ReLU'dur. CNN'lerin son katmanı çıkış katmanıdır. Çıkış katmanında sınıflandırma görevleri için softmax operatörü yaygın olarak kullanılmaktadır (Gu et al., 2006).

2.6 Popüler CNN Mimarileri

Çalışma kapsamında görüntü sınıflandırma uygulamalarında kullanılmakta olan Konvolüsyonel Sinir Ağı mimarileri bu başlık altında incelenecektir. Yapılan literatür çalışması doğrultusunda görüntü sınıflandırma uygulamalarında kullanılmak üzere 9 farklı CNN model mimarisi üzerinde durulmuştur.

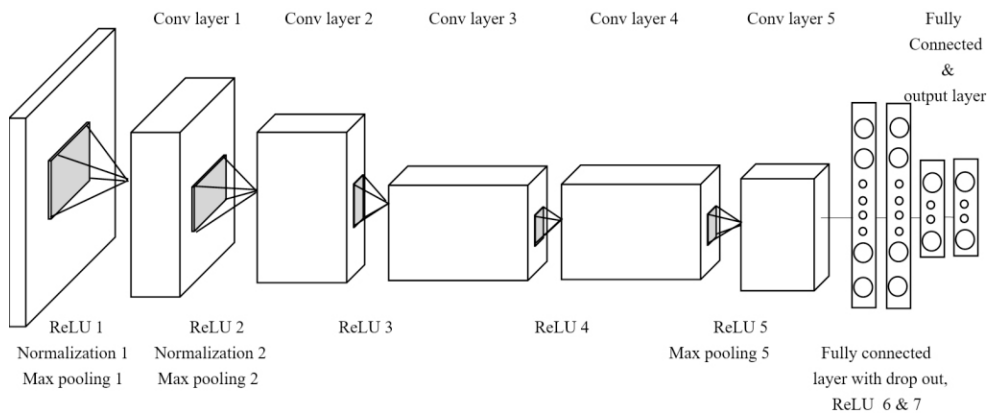
Alexnet modeli, 2012 yılında yapılan görsel tanıma ve sınıflandırma yarışmalarındaki başarımlarıyla, bu yıldan sonra görsel alanlarda yapılan çalışmalarda popülerliğini sağlamıştır (You et al., 2017). Vgg16Net modeli, AlexNet'e benzer yapısı ve daha derin evrişimli katmanları sayesinde 2014 yılında yapılmış olan görsel çalışmada en düşük hata oranı ile görsel alandaki başarısını kanıtlamıştır (Simonyan & Zisserman, 2015). Vgg19Net modeli, Vgg16Net modeline kıyasla 3 fazla evrimşimli katmanı sayesinde görsel alanlarda yapılan çalışmalarda yüksek başarımlar sağlamaktadır (Gatys et al., 2015). GoogleNet modelinde ise, literatür çalışmalarında görüldüğü üzere %5.7 gibi düşük bir hata oranı ile yüksek bir başarımlar elde etmiştir. Google tarafından geliştirilen bu model, Softmax'e sahip son katmandaki fonksiyon, toplam 42 katmandan ve giriş katmanından oluşmaktadır ve giriş değeri olarak 299×299 piksellik resimleri destekleyebilmektedir (Atila et al., 2021). Bir sonraki model, Microsoft firması tarafından geliştirilen ResNet modelinin 50 olarak adlandırılan modeli olup, ILSVRC-2015 yarışmasını kazanmasıyla popülerliğini duyurmuştur (Atila et al., 2021). MobileNet modeli ise, adından da anlaşılacağı üzere mobil sistemlerde kullanılmak için geliştirilmiş olup, diğer derin algoritmalara kıyasla daha az parametreye sahip olmasına rağmen ağlar, başarılı sonuçlar elde etme kapasitesine sahiptir (Ayan et al., 2020). DenseNet modeli ise, ResNet

modelinden esinlenerek yaratılmış bir model olarak bilinmesinin yanı sıra, gradyan kaybolmasını hafifletme, özellik yayılımını geliştirme ve parametre sayısını azaltma gibi daha yenilikçi yaklaşımlarla sunulmuştur (Liu et al., 2020). Xception model, Inception modelden esinlenmiş olup, tersine derinlemesine ayrılabilir evrişim katmanları kullanmasıyla dikkat çekmektedir (Ayan et al., 2020). NasNetMobile modeli, 2017 yılında bulunmuş olan ve En uygun sinir ağı mimarilerini bulabilmek için bir algoritma sağlayan Sinir Mimarisi Arama (NAS) çerçevesi kullanan yeni bir model olarak bilinmektedir (Cogan et al., 2019).

Modellere ait detaylı açıklamalar ve model mimarileri aşağıda incelenmektedir.

2.6.1 AlexNet Mimarisi

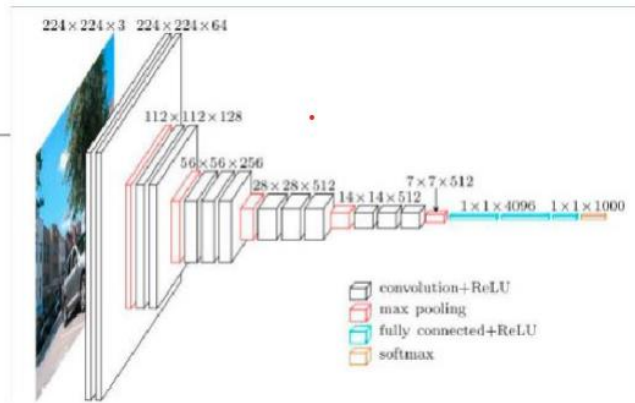
2012 yılında yapılan büyük ölçekli görsel tanıma yarışmasında AlexNet derin öğrenme mimarisi, örüntü tanımda hata oranını %26'lardan %15'lere indirgeyerek birinci olmuştur. Bu başarımla AlexNet'i en çok bilinen derin öğrenme mimarilerinden biri haline getirmiştir. Doğrusal olmayan fonksiyonlar için aktivasyon fonksiyonu (Relu) kullanılmaktadır. Bu aktivasyon fonksiyonu klasik tanh fonksiyonundan daha hızlı olduğundan eğitim süresini kısaltmak için kullanılmaktadır. Eğitim sürecinde aşırı öğrenme ve takılmaları önlemek için dropout katmanı kullanılmaktadır. Ağırlık gecikmeleri ve momentum değerleri için gradyan iniş modeli kullanılmaktadır. Şekil 2.6'da da görüldüğü gibi, 25 katmandan oluşan AlexNet, derin öğrenme ağı içerisinde 5 temel konvolüsyon katmanı yer almakta ve genellikle her konvolüsyon katmanından sonra bir aktivasyon katmanı olan relu katmanı kullanılmaktadır. Bununla beraber giriş katmanı, normalizasyon katmanı, havuzlama katmanı, drop out kamını, tam bağlı (full-connected) katman, yumuşak bağlı (SoftMax) katman ve çıkış katmanı yer almaktadır (Mahadeokar & Pesavento, 2016), (You et al., 2017).



Şekil 2.6. AlexNet Mimari Yapısı

2.6.2 VggNet Mimarisi

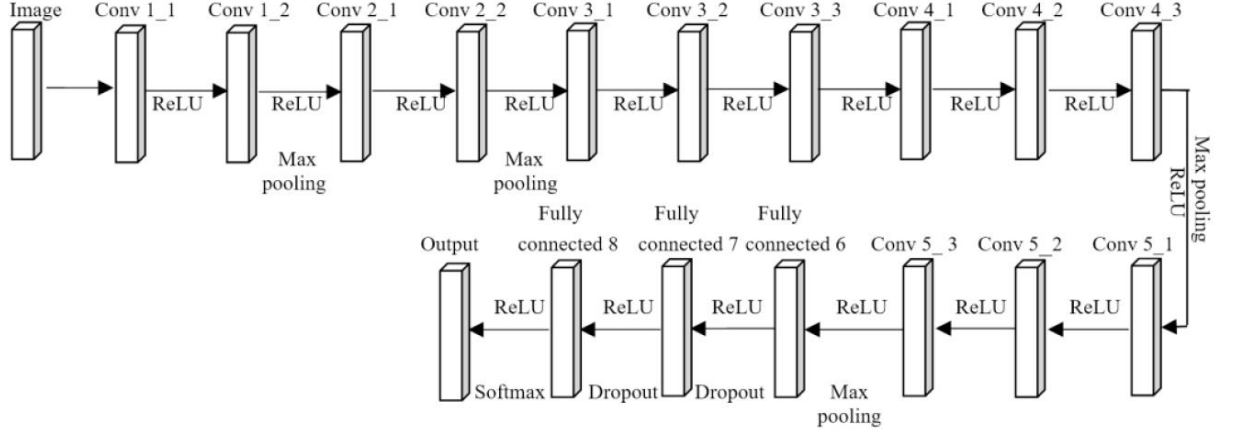
ImageNet 2014 yarışmasında %7.3 hata oranı ile oldukça başarılı bir performans sergileyen derin öğrenme algoritmasıdır. Simonyan ve Zisserman tarafından oxford üniversitesinde tasarlanan bu mimari ile 6 farklı mimari ortaya koymuşlardır. Bu 6 farklı 9 model de 11, 13, 16, 19 konvolüsyon katmandan oluşmaktadır (Simonyan & Zisserman, 2015). Önceki derin öğrenme algoritmalarıyla karşılaştırıldığında, içerisinde yer alan konvolüsyonlardan farklı olarak burada 2x2 ve 3x3'lük filtreler uygulanmıştır. Bu mimaride, Şekil 2.7'de de görüldüğü gibi art arda bağlı bulunan 3 tam bağlı (FC-FullConnected) katman yer almaktadır. Son tam bağlı katmanda 1000 nöron yer almakta olup çıkış için üretilen sınıflandırma katmanında ise yumuşak bağlı (SoftMax) katmanı bulunmaktadır. Bu modellerden VGG-16 (Ren et al., 2015); (S. Han et al., 2016) ve VGG-19 (Gatys et al., 2015) modelleri literatürde daha fazla yer almaktadır.



Şekil 2.7. VggNet mimari yapısı

2.6.2.1 VGG16Net Mimarisi

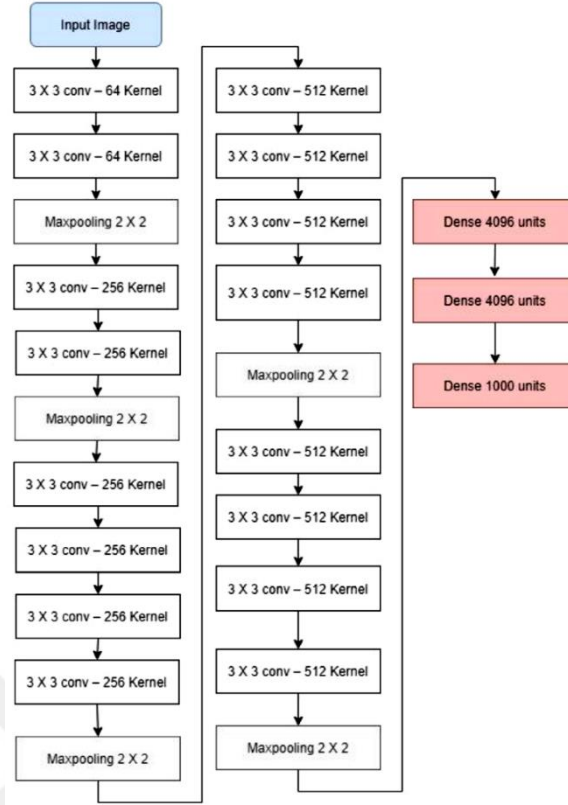
Bu model daha fazla sayıda evrişimli AlexNet'in yığılmış mimarisine dayanmaktadır. Şekil 3.8'de de görüldüğü gibi, giriş katmanı sonrası 13 evrişim katmanından oluşmaktadır ve her katman arasında ReLU katmanı kullanılmaktadır. Bazı evrişim katmanlarını, Alexnet'e benzer boyutu azaltmak için maxpooling katmanlarıyla takip etmektedir. AlexNet'te daha büyük boyutlar kullanılmasına kıyasla, Evrişim katmanlarında 3 x 3 boyutlu daha küçük filtreler kullanılmaktadır. Daha küçük filtreler, parametre sayısını azaltmak ve doğrusal olmayışı arttırmak için her filtreden sonra ReLU katmanı eklenmektedir (Simonyan & Zisserman, 2015).



Şekil 2.8. Vgg16Net mimari yapısı

2.6.2.2 Vgg19Net Mimarisi

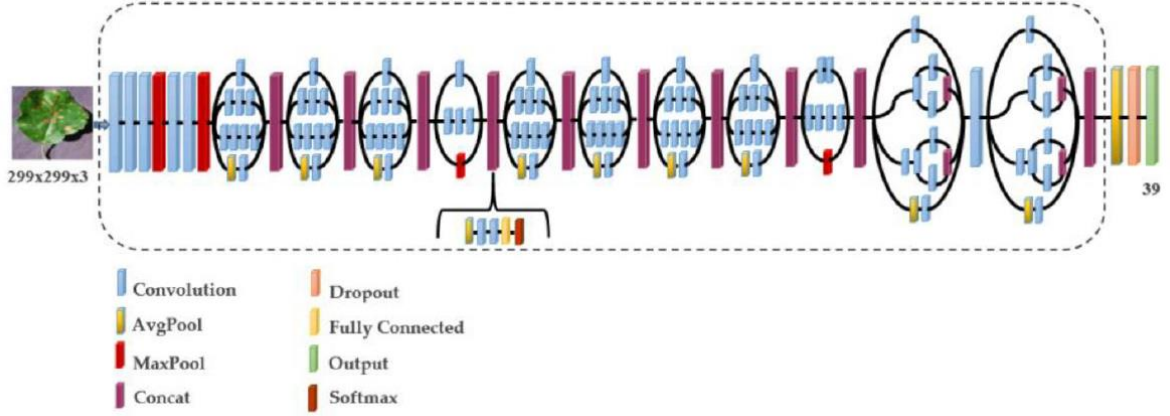
VGG 19, farklı sistemlerden oluşan VGG mimarisinden türetilmiştir. Şekil 2.9’da da görüldüğü gibi, evrişim katmanları gibi katmanlar 16, Tam bağlantılı katmanlar 3, MaxPool katmanları 5 ve SoftMax katmanı gibi katmanlardan oluşmaktadır. Büyük ölçekli görsel problemlerin çözümleri için tasarlanmıştır. Bu algoritmanın avantajı, kodunun açık kaynak olması ve ağın çalışmasını sağlayabilir diğer mimariler için aktarım öğrenimini kolayca uygulayabilir olmasıdır. Bu mimari ilk 5 puan tablosunda yer almaktadır ve başarımlar olarak görsel problemlerde yüksek başarımlar sunmaktadır (Subetha et al., 2021). VGG 19’un genel mimarisi aşağıda gösterilmektedir.



Şekil 2.9. Vgg19Net mimari yapısı

2.6.3 GoogleNet Mimarisi

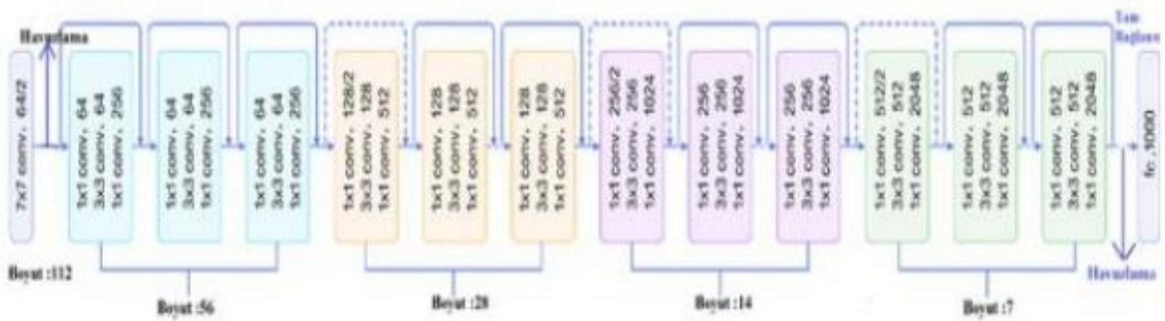
Komplex bir yapıya sahiptir. %5.7 gibi düşük bir hata oranı ile yüksek bir başarımla elde etmiştir. Şekil 3.10’da da görüldüğü gibi, 22 katman derinliğe sahip olup GoogleNet toplamda 144 katmandan oluşan bir yapısı bulunmaktadır. Inception modülü ile farklı boyutlarda filtreleme yaparak daha önce ortaya çıkan derin öğrenme mimarilerinden farklı bir oluşum ortaya koymaktadır. Bu filtrelemeler boyut azaltmak için kullanılan filtrelerdir. AlexNet’ten 12 kat daha az parametre içermektedir ve kullanılmakta olan katman sayısı bağımsız yapı taşlarına göre farklılık gösterebilmektedir. Inception modülünde yer alan filtre elemanları 1x1, 3x3, 5x5 şeklindedir. Diğer derin öğrenme mimarilerinde yer alan art arda yer alan katmansal yapıdan 10 farklılaşarak derinlikli bir yapı oluşturulmuştur. Bu derinliği oluşturan modüller bir filtreleme mantığı getirilmiştir (DOĞAN & TÜRKOĞLU, 2019).



Şekil 2.10. GoogleNet Mimari Yapısı

2.6.4 ResNet50 Mimarisi

2015 yılında gerçekleştirilen ILSVRC ImageNet yarışmasında Microsoft ResNet kazanan mimari olmuştur. Bu mimari, yarışmada %3.6 gibi bir hata oranı yakalamıştır. İnsanlar ortalama %5 - %10 hata oranı ile görüntüyü sınıflandırırken, ResNet %3.6'lık bir hata oranı ile insandan daha iyi görsel tanıma yaptığını kanıtlamıştır. Diğer mimariler ile karşılaştırıldığında ResNet mimarisi 152 katmandan oluşan çok yoğun katman sayısına sahiptir (Russakovsky et al., 2015). Microsoft ResNet mimarisinde iki RELU ile doğrusal katman arasında bir kez kalıntı değeri (Residual Value) beslenen bir bloktan (ResidualBlock) oluşturulmuştur. Bu yapı ile öğrenme daha hızlı şekilde gerçekleşeceği düşünülmüştür. Şekil 2.11'de bu kalıntı modülü görülmektedir (He et al., 2016).

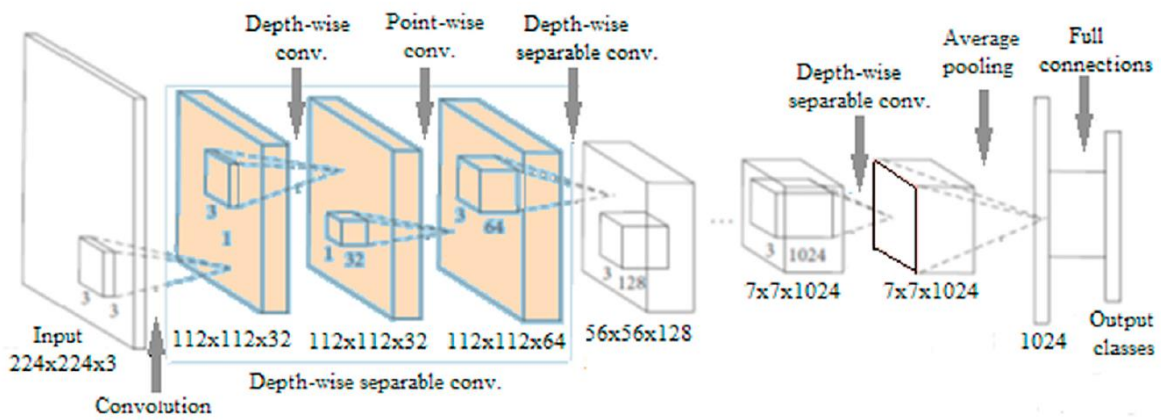


Şekil 2.11. ResNet50 Mimari Yapısı

2.6.5 MobileNet Mimarisi

MobileNet, mobil ve gömülü sistemlerde kullanılmak üzere geliştirilmiştir (Howard & Wang, 2012). Diğer derinlere göre daha az parametreye sahip olmasına rağmen ağlar, başarılı

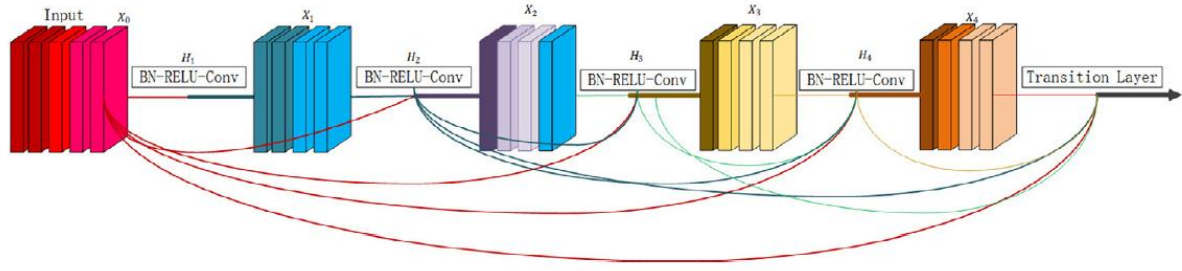
sonular elde etme kapasitesine sahip olması dikkat ekmektedir. Bu model gcn derinlemesine ayrılabilir evriřim katmanlarından almaktadır. Derinlemesine ayrılabilir evriřim katmanları, uzamsal korelasyonları haritalandırır ve giriř grntlerinin zellik haritalarında kanallar arası korelasyonlar. Őekil 2.12’de de grldđ gibi, MobileNet, derinlik ve noktasal dnřmler ayrı katmanlar olarak deđerlendirildiđinde toplam 28 katmandan oluřmaktadır. Modelin giriř katmanı, 224×224×3 boyutlu grntlerden oluřmaktadır. Modelin deđeritirilmiş versiyonu bir kresel ortalama havuzlama katmanı, 512 nronlu tamamen bađlantılı bir katman, ve sınıflandırma yapılacak grnt sınıflarına gre bir ıktı katmanından oluřmaktadır (Ayan et al., 2020).



Őekil 2.12. MobileNet Mimari Yapısı

2.6.6 DenseNet Mimarisi

Bu model, Evriřim Sinir Ađı (CNN) bir tr ileri beslemedir derin bir yapıya sahip sinir ađıdır. Evriřim hesaplama, derin đrenme algoritmalarının temsilcilerinden biridir. ResNet' ten sonra, DenseNet modeli ile grsellerin erveleri art arda katmanlar arası iletim sađlanıp, grnt alma alanında byk ilerleme sađlandı. AlexNet ve VGG19 modellerinin ıkarılan zellikleri 4096-dim, InceptionV3 ve ResNet50 modellerinin ıkarılan zellikleri 2048-dim ve DenseNet121 modelinin ıkarılan zellikleri ise 1024-dim'dir. Boyut kldke, zelliklerin fazlalıđı da azalır. Ancak, zelliklerin kullanım oranı daha iyi bir performans elde edilecek Őekilde artması sađlanmıřtır. Őekil 2.13’de de grldđ gibi, DenseNet, ResNet’in tm buluřlarından sonra ve en son geliřtirilmiř ađlardan biridir. ResNet'in paraları ve gradyan kaybolmasını hafifletme, zellik yayılımını geliřtirme ve parametre sayısını azaltma gibi daha yeniliki alıřmalar zerine kurulmuřtur (Liu et al., 2020).



Şekil 2.13. DenseNet Mimari Yapısı

2.6.7 Xception Mimarisi

Bu mimari Chollet (2017) tarafından tanıtılmıştır. Model Inception mimarisinden esinlenmiştir. Xception modeli, tersine derinlemesine ayrılabilir evrişim katmanları kullanmaktadır. Yani, ilk başta, 1×1 noktasal evrişim işlemi girdiye uygulanırsa, derinlemesine evrişim işlemi gerçekleşmektedir. Xception neredeyse aynı sayıda parametreye sahip olmasına rağmen, ImageNet veri kümesinde Inception-V3'ten daha iyi başarı elde etmiştir. Xception mimarisi 14 modülden ve 36 evrişim katmanından oluşmaktadır. Modelin girdi görüntü boyutu $299 \times 299 \times 3$ tür ve bir genel ortalama içerecek şekilde değiştirilmiştir. Havuzlama katmanı olmak üzere, bir 512 tam bağlı katman ve sınıflandırma yapılacak olan dataya ait sınıf sayısına göre bir çıktı katmanından oluşmaktadır (Ayan et al., 2020).

2.6.8 NasNetMobile Mimarisi

NASNet diğer sinir ağları ile karşılaştırıldığında biraz daha yeni bir ağıdır, Zoph ve ark. (ayrıca Google mühendisleri) 2017'de bulunmuştur. NASNet, En uygun sinir ağı mimarilerini bulabilmek için bir algoritma sağlayan Sinir Mimarisi Arama (NAS) çerçevesi kullanmaktadır. Diğer bir deyişle, NASNet mimarisi doğrudan insanlar tarafından tasarlanmamıştır. NASNet ILSVRC 2012 veri kümesinde olduğundan daha yüksek performans sunmaktadır. Sırasıyla Inception-v4 ve Inception-ResNet-v2, % 96,2'ye karşılık % 95,2 ve % 95,3 doğruluklar sunmaktadır. Ancak, performanstaki bu artış, ağ boyutunun maliyeti ve gerekli olan çoğaltma-biriktirme işlemleri gibi sorunları da beraberinde getirmektedir (Cogan et al., 2019).

2.7 Transfer Öğrenme

Transfer öğrenme, bir modelin herhangi bir amaç için, küçük veri kümelerinde ilgili bazı amaçlarla eğitim ve test için yeniden kullanılabilen büyük veri kümesi için tekrar eğitildiği bir süreçtir. T1 “önceden eğitilmiş model yaklaşımı” genellikle uygulanmaktadır. Bu

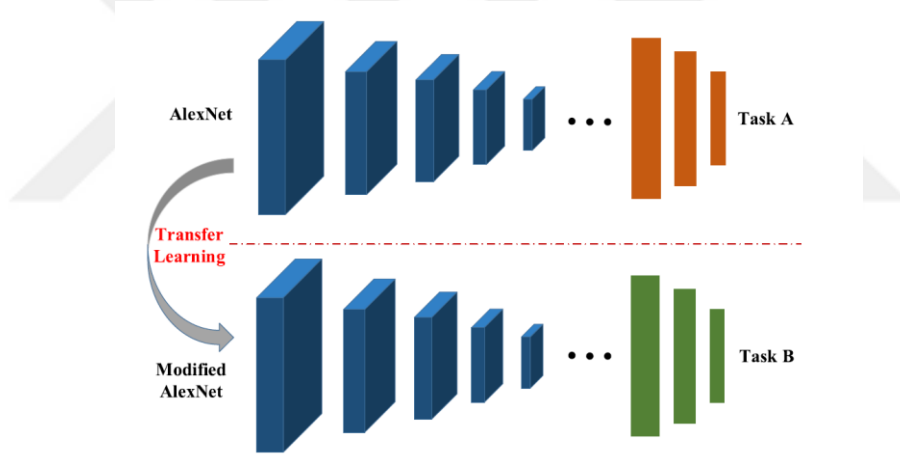
yaklaşımında önce bir model seçilir ve bu model daha önce küçük veri setlerinde uygulanmış olur ve bu kez büyük veri kümeleri üzerinde eğitilmek ve sonra tekrar kullanılabilmek için bu modelin gereksinimlerine göre ince ayar yapılmaktadır (Kumari & Seeja, 2020).

İlgili görevlerden öğrenmeyi amaçlayan transfer öğrenimi, son zamanlarda giderek daha fazla ilgi görmektedir. Bu alanın araştırması, daha önce öğrenilen bilginin genellemede oynadığı rol hakkında ilk tartışma olan ve özellikle eğitim verilerinin kıt olduğu dönemlerde (Thrun, n.d.) ile başlamıştır. Daha sonra (Ando, 2005) tarafından yapılan çalışmalarla, bilginin denetimli öğrenmeden denetimsiz öğrenmeye aktarılması amacıyla çoklu görevler tarafından paylaşılan hipotez alanlarının ortak bir yapısını keşfetmeye çalışılmıştır. (G. Xue et al., 2008), bu diller arası sınıflandırma problemini ele almak için bilgi darboğazına dayalı bir yaklaşım önermektedir. (Yang & Chen, 2009), veriler farklı özelliklerde olabildiğinde öğrenme performansını artıran yeni bir öğrenme senaryosu olan heterojen aktarım öğrenmeyi sunmuştur. Uygun bir haritalama alanı bulmak için seyrek kodlama yöntemine başvuran, alanlar arası yazışmanın neden olduğu önyargıyı azaltmak için kaynak alanı ve alanı hedeflemeyi yeni bir alana eşleyerek transfer öğrenmeyi uygulamaya çalışmıştır. Bu yaklaşım, iki alanın örnek dağılımlarının çok benzer olduğu şeklindeki güçlü hipoteze dayanmaktadır. CNN tabanlı transfer öğrenmeye odaklanan bazı araştırmalar da vardır. (Yosinski et al., n.d.), diğer veri kümelerinde farklı ince ayar stratejileri kullanarak ImageNet veri kümesinden önceden öğrenilen özelliklerin aktarılabilirliğini kapsamlı bir şekilde incelemektedir. (Jia et al., 2014), ağ katmanını katman katman ince ayarlayarak önceden eğitilmiş bir AlexNet'in her katman parametresinin aktarılabilirliğini göstermiştir. Bu ti modellerinin hepsi, bir milyondan daha fazlasını içeren ImageNet veritabanı üzerinde önceden eğitilmiştir (D. Han et al., 2018).

Önceden öğrenilen bilgi ve becerileri uygulayarak yeterli eğitim verisine sahip görevlerde edinilmiş verilerin, sonrasında ise sınırlı eğitim verisine sahip yeni görevler üzerine öğrenmeyi transfer eden modellerin performansını artırmak için olası bir çözüm sağlamaktadır. Transfer öğrenme bir benzerlikten yararlanan önemli bir makine öğrenimi dalı olup, önceden öğrenilen bilgileri aktararak veriler ve alanlar arasında ilgili bir alandaki yeni problemler için çözümler sunmaktadır. Bu yaklaşım ustaca derin öğrenme algoritmalarının gereksinimlerini çözerek büyük veri kümelerini işlemek için hesaplama gücüne duyulan ihtiyaç öğrenme özellikleri sağlamaktadır. Çünkü derin öğrenme yöntemleri doğrudan öğrenebilir ham verilerden, derin olmayan verilerle karşılaştırıldığında iki avantajı bulunmaktadır. Otomatik

olarak daha etkileyici özellikleri çıkarabilmektedirler ve pratik uygulamaların uçtan uca gereksinimlerini karşılayabilmektedirler (Y. Zhang et al., 2021).

Aktarım öğrenme yaklaşımları genellikle iyi eğitilmiş bir modelin ilk n katmanını, hedeflenen yeni öğrenecek modelin m katmanını kadarını aktarmasıyla gerçekleşir ve arasındaki ilişki ($m > n$) 'dir. Başlangıçta, hedef ağdaki ilk n katman dondurulur. Çünkü ilk birkaç katman tipik olarak en genel özellikleri öğrenen katmanlardır ve bunları doğrudan aktarmak çok daha iyidir. Ardından, hedef ağın son (m-n) katmanları, yeni göreve özgü örnek verilir. Ağ katmanları haline geldikçe daha derinde, giderek daha spesifik özelliklerin öğrenilmesi vurgulanır. Bazı durumlarda eklenen model, geçişi sırasında fine-tuning yapmak kaynak ve hedef veri alanları arasında oluşabilecek farkların üstesinden gelmekte ve ağın etkisinin iyileştirmesi sağlanabilmektedir (Y. Zhang et al., 2021). Şekil 2.14'de Orijinal Alexnet mimarisinden transfer öğrenme yardımı ile geliştirilen geliştirilmiş Alexnet modelinin bir örneği görülmektedir (Fatih et al., 2021).



Şekil 2.14. Örnek Transfer Öğrenme Modeli

2.8 Değerlendirme Ölçütleri

Bu çalışmada, her bir modelin başarısını değerlendirmek için doğruluk (accuracy), metriği kullanılmıştır. Doğruluk, sınıflandırma algoritmalarının başarısını ölçmede en çok kullanılan değerlendirme ölçütlerinden birisidir. Genel olarak, değerlendirilen toplam örnek sayısına göre doğru tahminlerin oranını ölçer (Hossin, 2016). Doğruluk değeri denklem 2.3 kullanılarak hesaplanır.

$$\text{Doğruluk} = \frac{TP + TN}{TP + TN + FN + FP} \quad (2.3)$$

2.9 Çalışmada Kullanılan Kütüphaneler

Bu tez çalışmasında problemlerin çözümüne yönelik tasarlanmış olan modellerin kodlaması, Python programlama dilinin 3. versiyonu kullanılarak geliştirilmiştir. Kodlar 12 gigabayta kadar ücretsiz GPU desteği sunan Google Colab platformu kullanılarak çalıştırılmıştır (Bisong, 2019). İlk olarak Colab'ın drive erişimi üzerinden verisetine ulaşabilmesi için drive colab kütüphanesi eklenmiştir. Sonrasında ise özellikle Python ile standart bir şekilde gelen ve kurulum gerektirmeyen, klasör yönetimi gibi işlemlerle kolaylık sağlayan bir kütüphane olan OS kullanılmıştır (Irmak & Saygılı, 2020). CNN algoritmaları için ise Keras Kütüphanesi kullanılmıştır (Chollet, 2019). Derin öğrenme algoritmalarındaki matris işlemleri için Numpy kütüphanesi (Oliphant, 2014). Modellerin sonuçlarının görselleştirilebilmesi için Matplotlib ve Seaborn Kütüphaneleri eklenmiştir (Thiruvathukal & Hunter, 2007). Çizelge 2.5'de çalışmada kullanılan kütüphaneler ve bu kütüphanelerin hangi aşamalarda kullanıldığı belirtilmiştir.

Çizelge 2.5. Çalışmada Kullanılan Kütüphaneler

Kütüphane	Açıklama	Aşama
Drive	Google Colab Drive erişimi	Veri setinin okunması ve işlenmesi
Time	Kod satır zaman sayacı	
numpy	Matris işlemleri	
OS	Klasör yönetimi işlemleri	
Shuffle	Veri setinin karışık işlenmesi	
StratifiedKFold	Kfold çapraz geçerlilik işlemi	
tensorflow.keras.applications	CNN Algoritmalarının çalıştırılması	Modellenmesi
Keras	CNN Algoritmalarının çalıştırılması	
matplotlib.pyplot	Görselleştirme	Modelin değerlendirilmesi
Seaborn	Görselleştirme	

3. ARAŞTIRMA BULGULARI

Bu bölümde, bitki türlerinin sınıflandırılması, bitki hastalık tahmini ve bitki hastalıklarının sınıflandırılması problemleri için yeniden düzenlenerek elde edilmiş yeni veri setlerinin kullanılmasıyla, her bir problem için, derin öğrenme mimarileri ile detaylı bir şekilde gerçekleştirilen modelleme çalışmalarına yer verilmiştir.

3.1 Bitki Türlerinin Sınıflandırılması

Bitki türlerinin sınıflandırılması problemi için, bölüm 3'te detaylı olarak anlatılmış olan 6 farklı dengeli veri seti (D1-BTS, D2-BTS, ..., D6-BTS), orijinal AlexNet mimarisi ve transfer öğrenme uygulanarak geliştirilmiş AlexNet mimarileri ile eğitilerek, her bir veri seti için ayrı modeller oluşturulmaya çalışılmıştır.

Orijinal AlexNet mimarisi ile oluşturulan modelde, giriş değeri 227x227x3 olarak seçilmiştir. AlexNet modeli, toplamda 5 konvolüsyonel katmandan oluşmakta olup 'padding' olarak 'same' değeri verilerek, modelin görüntüler üzerindeki tarama boyutu standart tutulmuştur. Katmanlar arasında kullanılan aktivasyon fonksiyonları ReLu'dur. Bu modelde, 3 tane 'Max pooling' katmanı, 5 konvolüsyonel katmanı ve son olarakta 3 'Fully connected' katmanı kullanılmıştır. Çıkış katmanı olarak Alexnet'in orijinal nöron boyutu sabit tutulmuş, Softmax aktivasyon fonksiyonuyla desteklenmiştir. Bu modelde optimizör olarak 'adam', 'batch_size' parametresi olarak 64, 'loss' fonksiyonu olarak 'categorical_crossentropy' seçilmiş ve model sırasıyla 3, 5 ve 10 kez çapraz doğrulama uygulanarak, 20 'epoch' için çalıştırılmıştır. Çizelge 3.1'de orijinal AlexNet mimarisine ait her bir katman detayları belirtilmiştir.

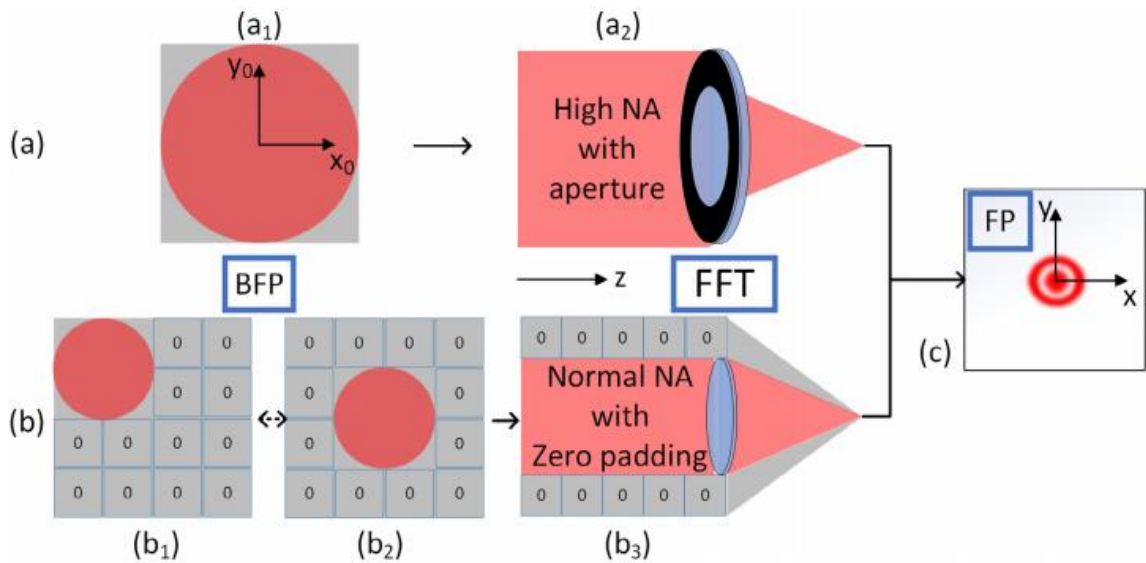
Çizelge 3.1. Detaylı Orijinal AlexNet Model Mimarisi

Katman	Katman İsmi	Katman Türü	Katman Detayları
1	'input'	Image Input	227x227x3 images with 'zerocenter' normalization
2	'conv1'	Convolution	96 11x11x3 convolutions with stride [4 4] and padding [0 0 0 0]
3	'relu1'	ReLU	ReLU
4	'norm1'	Cross Channel Normalization	cross channel normalization with 5 channels per element
5	'pool1'	Max Pooling	3x3 max pooling with stride [2 2] and padding [0 0 0 0]
6	'conv2'	Convolution	256 5x5x48 convolutions with stride [1 1] and padding [2 2 2 2]
7	'relu2'	ReLU	ReLU
8	'norm2'	Cross Channel Normalization	cross channel normalization with 5 channels per element

Çizelge 3.1 devamı. Detaylı Orjinal AlexNet Model Mimarisi

9	'pool2'	Max Pooling	3x3 max pooling with stride [2 2] and padding [0 0 0 0]
10	'conv3'	Convolution	384 3x3x256 convolutions with stride [1 1] and padding [1 1 1 1]
11	'relu3'	ReLU	ReLU
12	'conv4'	Convolution	384 3x3x192 convolutions with stride [1 1] and padding [1 1 1 1]
13	'relu4'	ReLU	ReLU
14	'conv5'	Convolution	256 3x3x192 convolutions with stride [1 1] and padding [1 1 1 1]
15	'relu5'	ReLU	ReLU
16	'pool5'	Max Pooling	3x3 max pooling with stride [2 2] and padding [0 0 0 0]
17	'fc6'	Fully Connected	4096 fully connected layer
18	'relu6'	ReLU	ReLU
19	'fc7'	Fully Connected	4096 fully connected layer
20	'relu7'	ReLU	ReLU
21	'fc8'	Fully Connected	1000 fully connected layer
22	prob'	Softmax	Softmax
23	classificationLayer'	Classification Output	crossentropyex

Orijinal AlexNet modelinde konvolüsyonel katman, 'max pooling' gibi katmanlarda verilmiş olan 'padding' değerlerinin 'same' olduğu gözlemlenmiştir. 'Same', 'zero padding' olarak bilinmesinin yanı sıra çerçeve dışına çıkılmadan sadece o aralıktaki görüntüsünün taranması ve çıktı boyutu girdi boyutuyla aynı olması anlamına gelmektedir (B. Zhang et al., 2019). Şekil 3.1'de 'Zero Padding' için bir örnek gösterilmiştir.



Şekil 3.1. Zero Padding Örneği

Transfer öğrenme uygulanmasıyla geliştirilmiş AlexNet modelinde ‘Padding’ değeri, ‘Valid’ yapılarak ‘filter_size’, 1 verilmiştir. Böylece çerçevelerde taranan alanın artması sonucu parametre sayısının arttığı gözlemlenmiştir. Bu modelde, giriş değeri 227x227x3 olarak verilmiştir. Bu model toplamda 5 konvolüsyonel katmandan oluşmaktadır ve yukarıda açıklandığı üzere padding olarak valid değeri verilerek, modelin görüntüler üzerindeki tarama boyutu maksimum seviyeye çıkartılmıştır. Katmanlar arasında kullanılan aktivasyon fonksiyonları ReLu’dur. 3 tane Max pooling katmanları yer almakta ve 5 konvolüsyonel katmandan sonra da 3 Fully connected katmanları kullanılmaktadır. Çıkış katmanı olarak datasetlerin sınıf sayıları sırasıyla nöron boyutu olarak verilmiş ve Softmax aktivasyon fonksiyonuyla desteklenmiştir. Bu modelde optimizör olarak adam, batch_size parametresi olarak 64, loss fonksiyonu olarak categorical_crossentropy seçilmiş ve model sırasıyla 3, 5 ve 10 kez çapraz doğrulama uygulanmıştır.

D1-BTS veri setinde, çıkış katmanının nöron sayısı 7 olarak belirlenmiş ve Şekil-15’de görüldüğü gibi 91,787,779 parametre sayısı ile model eğitilmiştir. Çizelge 3.2’de ise transfer öğrenme ile geliştirilmiş AlexNet mimarisine ait her bir katmana ait detaylar belirtilmiştir.

dense_1 (Dense)	(None, 4096)	16781312
batch_normalization_6 (Batch Normalization)	(None, 4096)	16384
activation_6 (Activation)	(None, 4096)	0
dropout_1 (Dropout)	(None, 4096)	0
dense_2 (Dense)	(None, 1000)	4097000
batch_normalization_7 (Batch Normalization)	(None, 1000)	4000
activation_7 (Activation)	(None, 1000)	0
dropout_2 (Dropout)	(None, 1000)	0
dense_3 (Dense)	(None, 7)	7007
batch_normalization_8 (Batch Normalization)	(None, 7)	28
activation_8 (Activation)	(None, 7)	0
=====		
Total params: 91,787,779		
Trainable params: 91,766,629		
Non-trainable params: 21,150		

Şekil 3.2. D1-BTS veri seti için geliştirilmiş AlexNet model parametresi

Çizelge 3.2. D1-BTS veri seti için Transfer Öğrenme Uygulanarak Geliştirilmiş AlexNet Model Mimarisi

Katman	Katman İsmi	Katman Türü	Katman Detayları
1	input	Image Input	227x227x3 images with 'zerocenter' normalization
2	conv2d	Convolution	96 11x11x3 convolutions with stride [4 4] and padding [1 1 1 1]
3	batch_normalization	Cross Channel Normalization	cross channel normalization with 5 channels per element
4	activation	Activation	ReLU
5	max_pooling2d	Max Pooling	3x3 max pooling with stride [2 2] and padding [1 1 1 1]
6	conv2d_1	Convolution	256 5x5x48 convolutions with stride [1 1] and padding [2 2 2 2]
7	batch_normalization_1	Cross Channel Normalization	cross channel normalization with 5 channels per element
8	activation_1	Activation	ReLU
9	max_pooling2d_1	Max Pooling	3x3 max pooling with stride [2 2] and padding [1 1 1 1]
10	conv2d_2	Convolution	384 3x3x256 convolutions with stride [1 1] and padding [1 1 1 1]
11	batch_normalization_2	Cross Channel Normalization	cross channel normalization with 5 channels per element
12	activation_2	Activation	ReLU
13	conv2d_3	Convolution	384 3x3x192 convolutions with stride [1 1] and padding [1 1 1 1]
14	batch_normalization_3	Cross Channel Normalization	cross channel normalization with 5 channels per element
15	activation_3	Activation	ReLU
16	conv2d_4	Convolution	256 3x3x192 convolutions with stride [1 1] and padding [1 1 1 1]
17	batch_normalization_4	Cross Channel Normalization	cross channel normalization with 5 channels per element
18	activation_4	Activation	ReLU
19	max_pooling2d_2	Max Pooling	3x3 max pooling with stride [2 2] and padding [1 1 1 1]
20	dense	Fully Connected	4096 fully connected layer
21	batch_normalization_5	Cross Channel Normalization	cross channel normalization with 5 channels per element
22	activation_5	Activation	ReLU
23	dense_1	Fully Connected	4096 fully connected layer
24	batch_normalization_6	Cross Channel Normalization	cross channel normalization with 5 channels per element
25	activation_6	Activation	ReLU

Çizelge 3.2 devamı. D1-BTS veri seti için Transfer Öğrenme Uygulanarak Geliştirilmiş AlexNet Model Mimarisi

26	dense_2	Fully Connected	1000 fully connected layer
27	batch_normalization_7	Cross Channel Normalization	cross channel normalization with 5 channels per element
28	activation_7	Activation	ReLU
29	dense_3	Classification Output	7
30	batch_normalization_8	Cross Channel Normalization	cross channel normalization with 5 channels per element
31	activation_8	Classification Activation	Softmax
Total Parameter: 91,787,779			

D2-BTS veri seti için transfer öğrenme uygulanarak AlexNet mimarisini geliştirmek istediğimizde, çıkış katmanının nöron sayısı 6 olarak belirlenmiştir ve Şekil 3.3’de görüldüğü gibi, 91,786,774 parametre sayısı ile model eğitilmiştir. Çizelge 3.3’de ise transfer öğrenme ile geliştirilmiş AlexNet mimarisine ait her bir katmana ait detaylar belirtilmiştir.

dense_13 (Dense)	(None, 4096)	16781312
batch_normalization_33 (Batch Normalization)	(None, 4096)	16384
activation_33 (Activation)	(None, 4096)	0
dropout_10 (Dropout)	(None, 4096)	0
dense_14 (Dense)	(None, 1000)	4097000
batch_normalization_34 (Batch Normalization)	(None, 1000)	4000
activation_34 (Activation)	(None, 1000)	0
dropout_11 (Dropout)	(None, 1000)	0
dense_15 (Dense)	(None, 6)	6006
batch_normalization_35 (Batch Normalization)	(None, 6)	24
activation_35 (Activation)	(None, 6)	0
=====		
Total params: 91,786,774		
Trainable params: 91,765,626		
Non-trainable params: 21,148		

Şekil 3.3. D2-BTS Veri Seti için Geliştirilmiş AlexNet Model Parametresi

Çizelge 3.3. D2-BTS veri seti için Transfer Öğrenme Uygulanarak Geliştirilmiş AlexNet Model Mimarisi

Katman	Katman İsmi	Katman Türü	Katman Detayları
1	input	Image Input	227x227x3 images with 'zerocenter' normalization
2	conv2d	Convolution	96 11x11x3 convolutions with stride [4 4] and padding [1 1 1 1]
3	batch_normalization	Cross Channel Normalization	cross channel normalization with 5 channels per element
4	activation	Activation	ReLU
5	max_pooling2d	Max Pooling	3x3 max pooling with stride [2 2] and padding [1 1 1 1]
6	conv2d_1	Convolution	256 5x5x48 convolutions with stride [1 1] and padding [2 2 2 2]
7	batch_normalization_1	Cross Channel Normalization	cross channel normalization with 5 channels per element
8	activation_1	Activation	ReLU
9	max_pooling2d_1	Max Pooling	3x3 max pooling with stride [2 2] and padding [1 1 1 1]
10	conv2d_2	Convolution	384 3x3x256 convolutions with stride [1 1] and padding [1 1 1 1]
11	batch_normalization_2	Cross Channel Normalization	cross channel normalization with 5 channels per element
12	activation_2	Activation	ReLU
13	conv2d_3	Convolution	384 3x3x192 convolutions with stride [1 1] and padding [1 1 1 1]
14	batch_normalization_3	Cross Channel Normalization	cross channel normalization with 5 channels per element
15	activation_3	Activation	ReLU
16	conv2d_4	Convolution	256 3x3x192 convolutions with stride [1 1] and padding [1 1 1 1]
17	batch_normalization_4	Cross Channel Normalization	cross channel normalization with 5 channels per element
18	activation_4	Activation	ReLU
19	max_pooling2d_2	Max Pooling	3x3 max pooling with stride [2 2] and padding [1 1 1 1]
20	dense	Fully Connected	4096 fully connected layer
21	batch_normalization_5	Cross Channel Normalization	cross channel normalization with 5 channels per element
22	activation_5	Activation	ReLU
23	dense_1	Fully Connected	4096 fully connected layer

Çizelge 3.3 devamı. D2-BTS veri seti için Transfer Öğrenme Uygulanarak Geliştirilmiş AlexNet Model Mimarisi

24	batch_normalization_6	Cross Channel Normalization	cross channel normalization with 5 channels per element
25	activation_6	Activation	ReLU
26	dense_2	Fully Connected	1000 fully connected layer
27	batch_normalization_7	Cross Channel Normalization	cross channel normalization with 5 channels per element
28	activation_7	Activation	ReLU
29	dense_3	Classification Output	6
30	batch_normalization_8	Cross Channel Normalization	cross channel normalization with 5 channels per element
31	activation_8	Classification Activation	Softmax
Total Parameter: 91,786,774			

D3-BTS veri seti için transfer öğrenme uygulanarak AlexNet mimarisini geliştirmek istediğimizde, çıkış katmanının nöron sayısı 5 olarak belirlenmiştir ve Şekil 3.4de görüldüğü gibi, 91,785,769 parametre sayısı ile model eğitilmiştir. Çizelge 3.4’de ise transfer öğrenme ile geliştirilmiş AlexNet mimarisine ait her bir katmana ait detaylar belirtilmiştir.

dense_17 (Dense)	(None, 4096)	16781312
batch_normalization_42 (Batch Normalization)	(None, 4096)	16384
activation_42 (Activation)	(None, 4096)	0
dropout_13 (Dropout)	(None, 4096)	0
dense_18 (Dense)	(None, 1000)	4097000
batch_normalization_43 (Batch Normalization)	(None, 1000)	4000
activation_43 (Activation)	(None, 1000)	0
dropout_14 (Dropout)	(None, 1000)	0
dense_19 (Dense)	(None, 5)	5005
batch_normalization_44 (Batch Normalization)	(None, 5)	20
activation_44 (Activation)	(None, 5)	0
=====		
Total params: 91,785,769		
Trainable params: 91,764,623		
Non-trainable params: 21,146		

Şekil 3.4. D3-BTS Veri Seti için Geliştirilmiş AlexNet Model Parametresi

Çizelge 3.4. D3-BTS veri seti için Transfer Öğrenme Uygulanarak Geliştirilmiş AlexNet Model Mimarisi

Katman	Katman İsmi	Katman Türü	Katman Detayları
1	input	Image Input	227x227x3 images with 'zerocenter' normalization
2	conv2d	Convolution	96 11x11x3 convolutions with stride [4 4] and padding [1 1 1 1]
3	batch_normalization	Cross Channel Normalization	cross channel normalization with 5 channels per element
4	activation	Activation	ReLU
5	max_pooling2d	Max Pooling	3x3 max pooling with stride [2 2] and padding [1 1 1 1]
6	conv2d_1	Convolution	256 5x5x48 convolutions with stride [1 1] and padding [2 2 2 2]
7	batch_normalization_1	Cross Channel Normalization	cross channel normalization with 5 channels per element
8	activation_1	Activation	ReLU
9	max_pooling2d_1	Max Pooling	3x3 max pooling with stride [2 2] and padding [1 1 1 1]
10	conv2d_2	Convolution	384 3x3x256 convolutions with stride [1 1] and padding [1 1 1 1]
11	batch_normalization_2	Cross Channel Normalization	cross channel normalization with 5 channels per element
12	activation_2	Activation	ReLU
13	conv2d_3	Convolution	384 3x3x192 convolutions with stride [1 1] and padding [1 1 1 1]
14	batch_normalization_3	Cross Channel Normalization	cross channel normalization with 5 channels per element
15	activation_3	Activation	ReLU
16	conv2d_4	Convolution	256 3x3x192 convolutions with stride [1 1] and padding [1 1 1 1]
17	batch_normalization_4	Cross Channel Normalization	cross channel normalization with 5 channels per element
18	activation_4	Activation	ReLU
19	max_pooling2d_2	Max Pooling	3x3 max pooling with stride [2 2] and padding [1 1 1 1]
20	dense	Fully Connected	4096 fully connected layer
21	batch_normalization_5	Cross Channel Normalization	cross channel normalization with 5 channels per element
22	activation_5	Activation	ReLU
23	dense_1	Fully Connected	4096 fully connected layer
24	batch_normalization_6	Cross Channel Normalization	cross channel normalization with 5 channels per element

Çizelge 3.4 devamı. D3-BTS veri seti için Transfer Öğrenme Uygulanarak Geliştirilmiş AlexNet Model Mimarisi

25	activation_6	Activation	ReLU
26	dense_2	Fully Connected	1000 fully connected layer
27	batch_normalization_7	Cross Channel Normalization	cross channel normalization with 5 channels per element
28	activation_7	Activation	ReLU
29	dense_3	Classification Output	5
30	batch_normalization_8	Cross Channel Normalization	cross channel normalization with 5 channels per element
31	activation_8	Classification Activation	Softmax
Total Parameter: 91,785,769			

D4-BTS veri seti için transfer öğrenme uygulanarak AlexNet mimarisini geliştirmek istediğimizde, çıkış katmanının nöron sayısı 4 olarak belirlenmiştir ve Şekil 3.5’de görüldüğü gibi, 91,784,764 parametre sayısı ile model eğitilmiştir. Çizelge 3.5’de ise transfer öğrenme ile geliştirilmiş AlexNet mimarisine ait her bir katmana ait detaylar belirtilmiştir.

dense_17 (Dense)	(None, 4096)	16781312
batch_normalization_42 (Batch Normalization)	(None, 4096)	16384
activation_42 (Activation)	(None, 4096)	0
dropout_13 (Dropout)	(None, 4096)	0
dense_18 (Dense)	(None, 1000)	4097000
batch_normalization_43 (Batch Normalization)	(None, 1000)	4000
activation_43 (Activation)	(None, 1000)	0
dropout_14 (Dropout)	(None, 1000)	0
dense_19 (Dense)	(None, 4)	4004
batch_normalization_44 (Batch Normalization)	(None, 4)	16
activation_44 (Activation)	(None, 4)	0
=====		
Total params: 91,784,764		
Trainable params: 91,763,620		
Non-trainable params: 21,144		

Şekil 3.5. D4-BTS Veri Seti için Geliştirilmiş AlexNet Model Parametresi

Çizelge 3.5. D4-BTS veri seti için Transfer Öğrenme Uygulanarak Geliştirilmiş AlexNet Model Mimarisi

Katman	Katman İsmi	Katman Türü	Katman Detayları
1	input	Image Input	227x227x3 images with 'zerocenter' normalization
2	conv2d	Convolution	96 11x11x3 convolutions with stride [4 4] and padding [1 1 1 1]
3	batch_normalization	Cross Channel Normalization	cross channel normalization with 5 channels per element
4	activation	Activation	ReLU
5	max_pooling2d	Max Pooling	3x3 max pooling with stride [2 2] and padding [1 1 1 1]
6	conv2d_1	Convolution	256 5x5x48 convolutions with stride [1 1] and padding [2 2 2 2]
7	batch_normalization_1	Cross Channel Normalization	cross channel normalization with 5 channels per element
8	activation_1	Activation	ReLU
9	max_pooling2d_1	Max Pooling	3x3 max pooling with stride [2 2] and padding [1 1 1 1]
10	conv2d_2	Convolution	384 3x3x256 convolutions with stride [1 1] and padding [1 1 1 1]
11	batch_normalization_2	Cross Channel Normalization	cross channel normalization with 5 channels per element
12	activation_2	Activation	ReLU
13	conv2d_3	Convolution	384 3x3x192 convolutions with stride [1 1] and padding [1 1 1 1]
14	batch_normalization_3	Cross Channel Normalization	cross channel normalization with 5 channels per element
15	activation_3	Activation	ReLU
16	conv2d_4	Convolution	256 3x3x192 convolutions with stride [1 1] and padding [1 1 1 1]
17	batch_normalization_4	Cross Channel Normalization	cross channel normalization with 5 channels per element
18	activation_4	Activation	ReLU
19	max_pooling2d_2	Max Pooling	3x3 max pooling with stride [2 2] and padding [1 1 1 1]
20	dense	Fully Connected	4096 fully connected layer
21	batch_normalization_5	Cross Channel Normalization	cross channel normalization with 5 channels per element
22	activation_5	Activation	ReLU
23	dense_1	Fully Connected	4096 fully connected layer
24	batch_normalization_6	Cross Channel Normalization	cross channel normalization with 5 channels per element

Çizelge 3.5 devamı. D4-BTS veri seti için Transfer Öğrenme Uygulanarak Geliştirilmiş AlexNet Model Mimarisi

25	activation_6	Activation	ReLU
26	dense_2	Fully Connected	1000 fully connected layer
27	batch_normalization_7	Cross Channel Normalization	cross channel normalization with 5 channels per element
28	activation_7	Activation	ReLU
29	dense_3	Classification Output	4
30	batch_normalization_8	Cross Channel Normalization	cross channel normalization with 5 channels per element
31	activation_8	Classification Activation	Softmax
Total Parameter: 91,784,764			

D5-BTS veri seti için transfer öğrenme uygulanarak AlexNet mimarisini geliştirmek istediğimizde, çıkış katmanının nöron sayısı 3 olarak belirlenmiştir ve Şekil 3.6’da görüldüğü gibi, 91,783,759 parametre sayısı ile model eğitilmiştir. Çizelge 3.6’da ise transfer öğrenme ile geliştirilmiş AlexNet mimarisine ait her bir katmana ait detaylar belirtilmiştir.

dense_1 (Dense)	(None, 4096)	16781312
batch_normalization_6 (Batch Normalization)	(None, 4096)	16384
activation_6 (Activation)	(None, 4096)	0
dropout_1 (Dropout)	(None, 4096)	0
dense_2 (Dense)	(None, 1000)	4097000
batch_normalization_7 (Batch Normalization)	(None, 1000)	4000
activation_7 (Activation)	(None, 1000)	0
dropout_2 (Dropout)	(None, 1000)	0
dense_3 (Dense)	(None, 3)	3003
batch_normalization_8 (Batch Normalization)	(None, 3)	12
activation_8 (Activation)	(None, 3)	0
=====		
Total params: 91,783,759		
Trainable params: 91,762,617		
Non-trainable params: 21,142		

Şekil 3.6. D5-BTS Veri Seti için Geliştirilmiş AlexNet Model Parametresi

Çizelge 3.6. D5-BTS veri seti için Transfer Öğrenme Uygulanarak Geliştirilmiş AlexNet Model Mimarisi

Katman	Katman İsmi	Katman Türü	Katman Detayları
1	input	Image Input	227x227x3 images with 'zerocenter' normalization
2	conv2d	Convolution	96 11x11x3 convolutions with stride [4 4] and padding [1 1 1 1]
3	batch_normalization	Cross Channel Normalization	cross channel normalization with 5 channels per element
4	activation	Activation	ReLU
5	max_pooling2d	Max Pooling	3x3 max pooling with stride [2 2] and padding [1 1 1 1]
6	conv2d_1	Convolution	256 5x5x48 convolutions with stride [1 1] and padding [2 2 2 2]
7	batch_normalization_1	Cross Channel Normalization	cross channel normalization with 5 channels per element
8	activation_1	Activation	ReLU
9	max_pooling2d_1	Max Pooling	3x3 max pooling with stride [2 2] and padding [1 1 1 1]
10	conv2d_2	Convolution	384 3x3x256 convolutions with stride [1 1] and padding [1 1 1 1]
11	batch_normalization_2	Cross Channel Normalization	cross channel normalization with 5 channels per element
12	activation_2	Activation	ReLU
13	conv2d_3	Convolution	384 3x3x192 convolutions with stride [1 1] and padding [1 1 1 1]
14	batch_normalization_3	Cross Channel Normalization	cross channel normalization with 5 channels per element
15	activation_3	Activation	ReLU
16	conv2d_4	Convolution	256 3x3x192 convolutions with stride [1 1] and padding [1 1 1 1]
17	batch_normalization_4	Cross Channel Normalization	cross channel normalization with 5 channels per element
18	activation_4	Activation	ReLU
19	max_pooling2d_2	Max Pooling	3x3 max pooling with stride [2 2] and padding [1 1 1 1]
20	dense	Fully Connected	4096 fully connected layer
21	batch_normalization_5	Cross Channel Normalization	cross channel normalization with 5 channels per element
22	activation_5	Activation	ReLU
23	dense_1	Fully Connected	4096 fully connected layer

Çizelge 3.6. D5-BTS veri seti için Transfer Öğrenme Uygulanarak Geliştirilmiş AlexNet Model Mimarisi

24	batch_normalization_6	Cross Channel Normalization	cross channel normalization with 5 channels per element
25	activation_6	Activation	ReLU
26	dense_2	Fully Connected	1000 fully connected layer
27	batch_normalization_7	Cross Channel Normalization	cross channel normalization with 5 channels per element
28	activation_7	Activation	ReLU
29	dense_3	Classification Output	3
30	batch_normalization_8	Cross Channel Normalization	cross channel normalization with 5 channels per element
31	activation_8	Classification Activation	Softmax
Total Parameter: 91,783,759			

D6-BTS veri seti için transfer öğrenme uygulanarak AlexNet mimarisini geliştirmek istediğimizde, çıkış katmanının nöron sayısı 2 olarak belirlenmiştir ve Şekil-20’de görüldüğü gibi, 91,782,754 parametre sayısı ile model eğitilmiştir. Tablo-13’de ise transfer öğrenme ile geliştirilmiş AlexNet mimarisine ait her bir katmana ait detaylar belirtilmiştir.

dense_1 (Dense)	(None, 4096)	16781312
batch_normalization_6 (Batch Normalization)	(None, 4096)	16384
activation_6 (Activation)	(None, 4096)	0
dropout_1 (Dropout)	(None, 4096)	0
dense_2 (Dense)	(None, 1000)	4097000
batch_normalization_7 (Batch Normalization)	(None, 1000)	4000
activation_7 (Activation)	(None, 1000)	0
dropout_2 (Dropout)	(None, 1000)	0
dense_3 (Dense)	(None, 2)	2002
batch_normalization_8 (Batch Normalization)	(None, 2)	8
activation_8 (Activation)	(None, 2)	0
=====		
Total params: 91,782,754		
Trainable params: 91,761,614		
Non-trainable params: 21,140		

Şekil 3.7. D6-BTS Veri Seti için Geliştirilmiş AlexNet Model Parametresi

Çizelge 3.7. D6-BTS veri seti için Transfer Öğrenme Uygulanarak Geliştirilmiş AlexNet Model Mimarisi

Katman	Katman İsmi	Katman Türü	Katman Detayları
1	input	Image Input	227x227x3 images with 'zerocenter' normalization
2	conv2d	Convolution	96 11x11x3 convolutions with stride [4 4] and padding [1 1 1 1]
3	batch_normalization	Cross Channel Normalization	cross channel normalization with 5 channels per element
4	activation	Activation	ReLU
5	max_pooling2d	Max Pooling	3x3 max pooling with stride [2 2] and padding [1 1 1 1]
6	conv2d_1	Convolution	256 5x5x48 convolutions with stride [1 1] and padding [2 2 2 2]
7	batch_normalization_1	Cross Channel Normalization	cross channel normalization with 5 channels per element
8	activation_1	Activation	ReLU
9	max_pooling2d_1	Max Pooling	3x3 max pooling with stride [2 2] and padding [1 1 1 1]
10	conv2d_2	Convolution	384 3x3x256 convolutions with stride [1 1] and padding [1 1 1 1]
11	batch_normalization_2	Cross Channel Normalization	cross channel normalization with 5 channels per element
12	activation_2	Activation	ReLU
13	conv2d_3	Convolution	384 3x3x192 convolutions with stride [1 1] and padding [1 1 1 1]
14	batch_normalization_3	Cross Channel Normalization	cross channel normalization with 5 channels per element
15	activation_3	Activation	ReLU
16	conv2d_4	Convolution	256 3x3x192 convolutions with stride [1 1] and padding [1 1 1 1]
17	batch_normalization_4	Cross Channel Normalization	cross channel normalization with 5 channels per element
18	activation_4	Activation	ReLU
19	max_pooling2d_2	Max Pooling	3x3 max pooling with stride [2 2] and padding [1 1 1 1]

Çizelge 3.7 devamı. D6-BTS veri seti için Transfer Öğrenme Uygulanarak Geliştirilmiş AlexNet Model Mimarisi

20	dense	Fully Connected	4096 fully connected layer
21	batch_normalization_5	Cross Channel Normalization	cross channel normalization with 5 channels per element
22	activation_5	Activation	ReLU
23	dense_1	Fully Connected	4096 fully connected layer
24	batch_normalization_6	Cross Channel Normalization	cross channel normalization with 5 channels per element
25	activation_6	Activation	ReLU
26	dense_2	Fully Connected	1000 fully connected layer
27	batch_normalization_7	Cross Channel Normalization	cross channel normalization with 5 channels per element
28	activation_7	Activation	ReLU
29	dense_3	Classification Output	2
30	batch_normalization_8	Cross Channel Normalization	cross channel normalization with 5 channels per element
31	activation_8	Classification Activation	Softmax
Toplam Parametre Sayısı: 91,782,754			

3.2 Faz 2 - Bitki Hastalık tahmini

Bitki hastalık tahmini problemi için, bölüm 3'te detaylı olarak anlatılmış olan, 'Sağlıklı' ve 'Hastalıklı' diye 2 sınıftan oluşan, D7-BHT veri seti kullanılarak 9 farklı model geliştirilmeye çalışılmıştır. Her bir model, aşağıda anlatılmış olup, model mimarileri, 'Imagenet' veri kümesi için önceden eğitilmiş ağırlık değerleriyle yüklenmiştir, ve herhangi bir değişiklik yapılmamıştır.

- 1- VGG16 Modeli:** Bu model, 13 konvolüsyon, 3 tam bağlı katman olmak üzere 16 katmandan oluşmaktadır. Modele girdi olarak, 224 x 224 x 3 boyutlarındaki resimler verilmiştir. Tam bağlı katman öncesi özellik çıkarma işlemleri yapılarak boyutları 7 x 7 x 512 olarak değiştirilmiştir. Bağlı katmanlar sonrasında, modelin çıktı katmanı olarak veri setimizin sınıf sayısı olan 2 değeri nöron olarak verilmiş ve aktivasyon

fonksiyonu olarak Softmax belirlenmiştir. Modelin veri setiyle eğitimleri sırasında; batch_size değeri 64, optimizer'ı Adam ve loss fonksiyon olarak da sparse_categorical_crossentropy kullanılmıştır. Sırasıyla, 3 kez çapraz doğrulama için 7 epoch, 5 kez çapraz doğrulama için 4 epoch ve 10 kez çapraz doğrulama için de 3 epoch değerleri uygulanmıştır. Şekil 3.16'da görüldüğü gibi, 14,764,866 parametre ile model eğitilmiştir.

Layer (type)	Output Shape	Param #
input_1 (InputLayer)	[(None, 224, 224, 3)]	0
block1_conv1 (Conv2D)	(None, 224, 224, 64)	1792
block1_conv2 (Conv2D)	(None, 224, 224, 64)	36928
block1_pool (MaxPooling2D)	(None, 112, 112, 64)	0
block2_conv1 (Conv2D)	(None, 112, 112, 128)	73856
block2_conv2 (Conv2D)	(None, 112, 112, 128)	147584
block2_pool (MaxPooling2D)	(None, 56, 56, 128)	0
block3_conv1 (Conv2D)	(None, 56, 56, 256)	295168
block3_conv2 (Conv2D)	(None, 56, 56, 256)	590080
block3_conv3 (Conv2D)	(None, 56, 56, 256)	590080
block3_pool (MaxPooling2D)	(None, 28, 28, 256)	0
block4_conv1 (Conv2D)	(None, 28, 28, 512)	1180160
block4_conv2 (Conv2D)	(None, 28, 28, 512)	2359808
block4_conv3 (Conv2D)	(None, 28, 28, 512)	2359808
block4_pool (MaxPooling2D)	(None, 14, 14, 512)	0
block5_conv1 (Conv2D)	(None, 14, 14, 512)	2359808
block5_conv2 (Conv2D)	(None, 14, 14, 512)	2359808
block5_conv3 (Conv2D)	(None, 14, 14, 512)	2359808
block5_pool (MaxPooling2D)	(None, 7, 7, 512)	0
flatten (Flatten)	(None, 25088)	0
dense (Dense)	(None, 2)	50178
Total params: 14,764,866		
Trainable params: 14,764,866		
Non-trainable params: 0		

Şekil 3.8. D7-BHT Veri Seti ile Eğitilen VGGNET16 Modeli

2- VGG19 Modeli: Bu model 16 konvolüsyon, 3 tam bağlı katman olmak üzere 19 katmandan oluşmaktadır. Modele 224 x 224 x 3 boyutlarındaki resimler veri girişi olarak verilmiştir. Tam bağlı katman öncesi özellik çıkarma işlemleri yapılarak boyutları 7 x 7 x 512 olarak değiştirilmiştir. Bağlı katmanlar sonrasında modelin çıktı katmanı olarak verisetimizin sınıf sayısı olan 2 değeri nöron olarak verilmiş ve aktivasyon fonksiyonu olarak Softmax belirlenmiştir. Modelin verisiyle eğitimleri sırasında; batch_size değeri 64, optimizer'ı Adam ve loss fonksiyon olarak da sparse_categorical_crossentropy kullanılmıştır. Sırasıyla, 3 kez çapraz doğrulama için 7 epoch, 5 kez çapraz doğrulama için 4 epoch ve 10 kez çapraz doğrulama için de 3

epoch değerleri uygulanmıştır. Şekil 3.17’de görüldüğü gibi, 20,074,562 parametre ile model eğitilmiştir.

Layer (type)	Output Shape	Param #
input_1 (InputLayer)	[(None, 224, 224, 3)]	0
block1_conv1 (Conv2D)	(None, 224, 224, 64)	1792
block1_conv2 (Conv2D)	(None, 224, 224, 64)	36928
block1_pool (MaxPooling2D)	(None, 112, 112, 64)	0
block2_conv1 (Conv2D)	(None, 112, 112, 128)	73856
block2_conv2 (Conv2D)	(None, 112, 112, 128)	147584
block2_pool (MaxPooling2D)	(None, 56, 56, 128)	0
block3_conv1 (Conv2D)	(None, 56, 56, 256)	295168
block3_conv2 (Conv2D)	(None, 56, 56, 256)	590880
block3_conv3 (Conv2D)	(None, 56, 56, 256)	590880
block3_conv4 (Conv2D)	(None, 56, 56, 256)	590880
block3_pool (MaxPooling2D)	(None, 28, 28, 256)	0
block4_conv1 (Conv2D)	(None, 28, 28, 512)	1180160
block4_conv2 (Conv2D)	(None, 28, 28, 512)	2359808
block4_conv3 (Conv2D)	(None, 28, 28, 512)	2359808
block4_conv4 (Conv2D)	(None, 28, 28, 512)	2359808
block4_pool (MaxPooling2D)	(None, 14, 14, 512)	0
block5_conv1 (Conv2D)	(None, 14, 14, 512)	2359808
block5_conv2 (Conv2D)	(None, 14, 14, 512)	2359808
block5_conv3 (Conv2D)	(None, 14, 14, 512)	2359808
block5_conv4 (Conv2D)	(None, 14, 14, 512)	2359808
block5_pool (MaxPooling2D)	(None, 7, 7, 512)	0
flatten (Flatten)	(None, 25088)	0
dropout (Dropout)	(None, 25088)	0
dense (Dense)	(None, 2)	50178
Total params: 20,074,562		
Trainable params: 20,074,562		
Non-trainable params: 0		

Şekil 3.9. D7-BHT Veri Seti ile Eğitilen VGGNET19 Modeli

3- AlexNet Modeli: Bu modelin giriş değeri 227x227x3 olarak verilmiştir. AlexNet modeli toplamda 5 konvolüsyonel katmandan oluşmakta olup, ‘padding’ olarak ‘same’ değeri verilerek, modelin görüntüler üzerindeki tarama boyutu standart tutulmuştur. Katmanlar arasında kullanılan aktivasyon fonksiyonları ReLu’dur. 3 adet ‘Max pooling’ katmanı yer almakta ve 5 konvolüsyonel katmandan sonra da 3 ‘Fully connected’ katmanı kullanılmıştır. Çıkış katmanı olarak veri setinin sınıf sayısı olan 2 nöron sayısı verilmiştir, ve Softmax aktivasyon fonksiyonuyla desteklenmiştir. Modelin veri setiyle eğitimleri sırasında; ‘batch_size’ değeri 64, ‘optimizer’, ‘Adam’ ve ‘loss’ fonksiyonu olarak da ‘sparse_categorical_crossentropy’ kullanılmıştır. Sırasıyla, 3 kez çapraz doğrulama için 27 epoch, 5 kez çapraz doğrulama için 48 epoch ve 10 kez çapraz doğrulama için de 48 epoch değerleri uygulanmıştır. Şekil 3.18’de görüldüğü gibi, 28,845,514 parametre ile model eğitilmiştir.

Layer (type)	Output Shape	Param #
conv2d_20 (Conv2D)	(None, 55, 55, 96)	34944
max_pooling2d_12 (MaxPooling)	(None, 27, 27, 96)	0
batch_normalization_24 (Batch Normalization)	(None, 27, 27, 96)	384
conv2d_21 (Conv2D)	(None, 23, 23, 256)	614656
max_pooling2d_13 (MaxPooling)	(None, 11, 11, 256)	0
batch_normalization_25 (Batch Normalization)	(None, 11, 11, 256)	1024
conv2d_22 (Conv2D)	(None, 9, 9, 384)	885120
conv2d_23 (Conv2D)	(None, 7, 7, 384)	1327488
conv2d_24 (Conv2D)	(None, 5, 5, 256)	884992
max_pooling2d_14 (MaxPooling)	(None, 2, 2, 256)	0
batch_normalization_26 (Batch Normalization)	(None, 2, 2, 256)	1024
flatten_4 (Flatten)	(None, 1024)	0
dense_16 (Dense)	(None, 4096)	4198400
dropout_12 (Dropout)	(None, 4096)	0
batch_normalization_27 (Batch Normalization)	(None, 4096)	16384
dense_17 (Dense)	(None, 4096)	16781312
dropout_13 (Dropout)	(None, 4096)	0
batch_normalization_28 (Batch Normalization)	(None, 4096)	16384
dense_18 (Dense)	(None, 1000)	4097000
dropout_14 (Dropout)	(None, 1000)	0
batch_normalization_29 (Batch Normalization)	(None, 1000)	4000
dense_19 (Dense)	(None, 2)	2002
Total params: 28,865,114		
Trainable params: 28,845,514		
Non-trainable params: 19,600		

Şekil 3.10. D7-BHT Veri Seti ile Eğitilen AlexNet modeli

4- ResNet50 Modeli: Bu model 177 katmandan oluşmaktadır ve giriş değeri, 224 x 224 x 3 olarak verilmiştir. Bu modelin ResNet50 diye bilinmesinin sebebi içerisinde 50 katmandan oluşan konvolüsyon katmanı olmasıdır. Modelin çıkış değeri olarak veri setinin 2 sınıflı olmasından ötürü, 2 nöron olarak belirlenmiştir. Modelin veri setiyle eğitimleri sırasında ‘batch_size’ değeri olarak 64, ‘optimizer’ olarak ‘Adam’ ve ‘loss’ fonksiyonu olarak da ‘sparse_categorical_crossentropy’ kullanılmıştır. Sırasıyla, 3 kez çapraz doğrulama için 10 epoch, 5 kez çapraz doğrulama için 5 epoch ve 10 kez çapraz doğrulama için de 3 epoch değerleri uygulanmıştır. Şekil 3.19’da görüldüğü gibi, 23,735,298 parametre ile model eğitilmiştir.

conv5_block2_2_relu (Activation)	(None, 7, 7, 512)	0	conv5_block2_2_bn[0][0]
conv5_block2_3_conv (Conv2D)	(None, 7, 7, 2048)	1050624	conv5_block2_2_relu[0][0]
conv5_block2_3_bn (BatchNormali)	(None, 7, 7, 2048)	8192	conv5_block2_3_conv[0][0]
conv5_block2_add (Add)	(None, 7, 7, 2048)	0	conv5_block1_out[0][0] conv5_block2_3_bn[0][0]
conv5_block2_out (Activation)	(None, 7, 7, 2048)	0	conv5_block2_add[0][0]
conv5_block3_1_conv (Conv2D)	(None, 7, 7, 512)	1049088	conv5_block2_out[0][0]
conv5_block3_1_bn (BatchNormali)	(None, 7, 7, 512)	2048	conv5_block3_1_conv[0][0]
conv5_block3_1_relu (Activation)	(None, 7, 7, 512)	0	conv5_block3_1_bn[0][0]
conv5_block3_2_conv (Conv2D)	(None, 7, 7, 512)	2359808	conv5_block3_1_relu[0][0]
conv5_block3_2_bn (BatchNormali)	(None, 7, 7, 512)	2048	conv5_block3_2_conv[0][0]
conv5_block3_2_relu (Activation)	(None, 7, 7, 512)	0	conv5_block3_2_bn[0][0]
conv5_block3_3_conv (Conv2D)	(None, 7, 7, 2048)	1050624	conv5_block3_2_relu[0][0]
conv5_block3_3_bn (BatchNormali)	(None, 7, 7, 2048)	8192	conv5_block3_3_conv[0][0]
conv5_block3_add (Add)	(None, 7, 7, 2048)	0	conv5_block2_out[0][0] conv5_block3_3_bn[0][0]
conv5_block3_out (Activation)	(None, 7, 7, 2048)	0	conv5_block3_add[0][0]
flatten (Flatten)	(None, 100352)	0	conv5_block3_out[0][0]
dense (Dense)	(None, 2)	200706	flatten[0][0]
=====			
Total params: 23,788,418			
Trainable params: 23,735,298			
Non-trainable params: 53,120			

Şekil 3.11. D7-BHT Veri Seti ile Eğitilen ResNet50 Modeli Parametre Gösterimi

5- DenseNet Modeli: Bu modelde girdi olarak 224 x 224 x 3 değeri verilmiştir. 5 blok konvolüsyon katmanı, 3 blok ‘transition’ katmanı (6, 12, 32), 1 blok sınıflandırma katmanı (32’li) ve son olarak da 2 blok ‘denseBlok’ katmanı bulunmaktadır. Modelin çıktı katmanı olarak veri setinin sınıf sayısı olan 2 değeri nöron olarak verilmiş ve aktivasyon fonksiyonu olarak ‘Softmax’ belirlenmiştir. Tüm bu bloklar içerisinde yer alan katmanlar toplandığı zaman, DenseNet169’un katman sayısı olan 169 elde edilmektedir. Modelin ver isetiyle eğitimleri sırasında; ‘batch_size’ değeri 64, ‘optimizer’ olarak Adam ve ‘loss’ fonksiyon olarak da ‘sparse_categorical_crossentropy’ kullanılmıştır. Sırasıyla, 3 kez çapraz doğrulama için 5 epoch, 5 kez çapraz doğrulama için 4 epoch ve 10 kez çapraz doğrulama için de 3 epoch değerleri uygulanmıştır. Şekil 3.20’de görüldüğü gibi, 12,810,626 parametre ile model eğitilmiştir.

conv5_block31_concat	(Concatena (None, 7, 7, 1632))	0	conv5_block30_concat[0][0] conv5_block31_2_conv[0][0]
conv5_block32_0_bn	(BatchNormal (None, 7, 7, 1632))	6528	conv5_block31_concat[0][0]
conv5_block32_0_relu	(Activatio (None, 7, 7, 1632))	0	conv5_block32_0_bn[0][0]
conv5_block32_1_conv	(Conv2D) (None, 7, 7, 128)	208896	conv5_block32_0_relu[0][0]
conv5_block32_1_bn	(BatchNormal (None, 7, 7, 128))	512	conv5_block32_1_conv[0][0]
conv5_block32_1_relu	(Activatio (None, 7, 7, 128))	0	conv5_block32_1_bn[0][0]
conv5_block32_2_conv	(Conv2D) (None, 7, 7, 32)	36864	conv5_block32_1_relu[0][0]
conv5_block32_concat	(Concatena (None, 7, 7, 1664))	0	conv5_block31_concat[0][0] conv5_block32_2_conv[0][0]
bn	(BatchNormalization) (None, 7, 7, 1664)	6656	conv5_block32_concat[0][0]
relu	(Activation) (None, 7, 7, 1664)	0	bn[0][0]
flatten_1	(Flatten) (None, 81536)	0	relu[0][0]
dropout_1	(Dropout) (None, 81536)	0	flatten_1[0][0]
batch_normalization	(BatchNorma (None, 81536))	326144	dropout_1[0][0]
dense_1	(Dense) (None, 2)	163074	batch_normalization[0][0]

Total params: 13,132,098			
Trainable params: 12,810,626			
Non-trainable params: 321,472			

Şekil 3.12. D7-BHT veri seti ile eğitilen DenseNet modeli parametre gösterimi

6- GoogleNet / Inception Modeli: Bu modelin giriş katman değeri 224 x 224 x 3 olarak belirlenmiştir. 22 katman derinliğe sahip olup GoogleNet toplamda 144 katmandan oluşan bir yapısı bulunmaktadır. Modelin çıktı katmanını olarak veri setinin sınıf sayısı olan 2 değeri nöron olarak verilmiş ve aktivasyon fonksiyonu olarak ‘Softmax’ belirlenmiştir. Modelin veri setiyle eğitimleri sırasında; ‘batch_size’ değeri 32, ‘optimizer’ olarak Adam ve ‘loss’ fonksiyon olarak da ‘sparse_categorical_crossentropy’ kullanılmıştır. Sırasıyla, 3 kez çapraz doğrulama için 6 epoch, 5 kez çapraz doğrulama için 5 epoch ve 10 kez çapraz doğrulama için de 3 epoch değerleri uygulanmıştır. Şekil 3.21’de görüldüğü gibi, 21,973,154 parametre ile model eğitilmiştir.

activation_91 (Activation)	(None, 5, 5, 384)	0	batch_normalization_91[0][0]
activation_92 (Activation)	(None, 5, 5, 384)	0	batch_normalization_92[0][0]
batch_normalization_93 (BatchNo	(None, 5, 5, 192)	576	conv2d_93[0][0]
activation_85 (Activation)	(None, 5, 5, 320)	0	batch_normalization_85[0][0]
mixed9_1 (Concatenate)	(None, 5, 5, 768)	0	activation_87[0][0] activation_88[0][0]
concatenate_1 (Concatenate)	(None, 5, 5, 768)	0	activation_91[0][0] activation_92[0][0]
activation_93 (Activation)	(None, 5, 5, 192)	0	batch_normalization_93[0][0]
mixed10 (Concatenate)	(None, 5, 5, 2048)	0	activation_85[0][0] mixed9_1[0][0] concatenate_1[0][0] activation_93[0][0]
flatten (Flatten)	(None, 51200)	0	mixed10[0][0]
dropout (Dropout)	(None, 51200)	0	flatten[0][0]
batch_normalization_94 (BatchNo	(None, 51200)	204800	dropout[0][0]
dense (Dense)	(None, 2)	102402	batch_normalization_94[0][0]

Total params: 22,109,986			
Trainable params: 21,973,154			
Non-trainable params: 136,832			

Şekil 3.13. D7-BHT veri seti ile eğitilen GoogleNet / InceptionV3 modeli parametre gösterimi

7- MobileNet Modeli: Bu model, giriş katman değeri olarak 224 x 224 x 3 olarak verilmiştir. Giriş bölümünde detaylıca tanıtılan bu model içerisinde, derinlemesine ayrılabilir evrişim katmanları yer almakta ve uzamsal korelasyonları haritalandırabilmektedir. MobileNet, derinlik ve noktasal dönüşümler ayrı katmanlar olarak değerlendirildiğinde toplam 28 katman yer almaktadır. Modelin çıktı katmanı olarak veri setinin sınıf sayısı olan 2 değeri nöron olarak verilmiş ve aktivasyon fonksiyonu olarak ‘Softmax’ belirlenmiştir. Modelin veri setiyle eğitimleri sırasında; ‘batch_size’ değeri 32, ‘optimizer’ olarak Adam ve ‘loss’ fonksiyon olarak da ‘sparse_categorical_crossentropy’ kullanılmıştır. Sırasıyla, 3 kez çapraz doğrulama için 6 epoch, 5 kez çapraz doğrulama için 5 epoch ve 10 kez çapraz doğrulama için de 3 epoch değerleri uygulanmıştır. Şekil 3.22’de görüldüğü gibi, 3,407,682 parametre ile model eğitilmiştir.

conv_dw_12_relu (ReLU)	(None, 7, 7, 512)	0
conv_pw_12 (Conv2D)	(None, 7, 7, 1024)	524288
conv_pw_12_bn (BatchNormaliz	(None, 7, 7, 1024)	4096
conv_pw_12_relu (ReLU)	(None, 7, 7, 1024)	0
conv_dw_13 (DepthwiseConv2D)	(None, 7, 7, 1024)	9216
conv_dw_13_bn (BatchNormaliz	(None, 7, 7, 1024)	4096
conv_dw_13_relu (ReLU)	(None, 7, 7, 1024)	0
conv_pw_13 (Conv2D)	(None, 7, 7, 1024)	1048576
conv_pw_13_bn (BatchNormaliz	(None, 7, 7, 1024)	4096
conv_pw_13_relu (ReLU)	(None, 7, 7, 1024)	0
flatten_15 (Flatten)	(None, 50176)	0
dropout_14 (Dropout)	(None, 50176)	0
batch_normalization_1416 (Ba	(None, 50176)	200704
dense_15 (Dense)	(None, 2)	100354
=====		
Total params: 3,529,922		
Trainable params: 3,407,682		
Non-trainable params: 122,240		

Şekil 3.14. D7-BHT veri seti ile eğitilen MobileNet modeli parametre gösterimi

8- NasNetMobile Modeli: Bu modele giriş katman değeri olarak 224 x 224 x 3 olarak verilmiştir. Modelin çıktı katmanını olarak veri setinin sınıf sayısı olan 2 değeri nöron olarak verilmiş ve aktivasyon fonksiyonu olarak ‘Softmax’ belirlenmiştir. Modelin veri setiyle eğitimleri sırasında; ‘batch_size’ değeri 64, ‘optimizer’ olarak Adam ve ‘loss’ fonksiyon olarak da ‘sparse_categorical_crossentropy’ kullanılmıştır. Sırasıyla, 3 kez çapraz doğrulama için 8 epoch, 5 kez çapraz doğrulama için 8 epoch ve 10 kez çapraz doğrulama için de 4 epoch değerleri uygulanmıştır. Şekil 3.23’de görüldüğü gibi, 4,439,956 parametre ile model eğitilmiştir.

separable_conv_2_bn_normal_left (None, 7, 7, 176)	704	separable_conv_2_normal_left5_12[
normal_add_1_12 (Add)	(None, 7, 7, 176)	0	separable_conv_2_bn_normal_left1 separable_conv_2_bn_normal_right1
normal_add_2_12 (Add)	(None, 7, 7, 176)	0	separable_conv_2_bn_normal_left2 separable_conv_2_bn_normal_right2
normal_add_3_12 (Add)	(None, 7, 7, 176)	0	normal_left3_12[0][0] adjust_bn_12[0][0]
normal_add_4_12 (Add)	(None, 7, 7, 176)	0	normal_left4_12[0][0] normal_right4_12[0][0]
normal_add_5_12 (Add)	(None, 7, 7, 176)	0	separable_conv_2_bn_normal_left5_ normal_bn_1_12[0][0]
normal_concat_12 (Concatenate)	(None, 7, 7, 1056)	0	adjust_bn_12[0][0] normal_add_1_12[0][0] normal_add_2_12[0][0] normal_add_3_12[0][0] normal_add_4_12[0][0] normal_add_5_12[0][0]
activation_187 (Activation)	(None, 7, 7, 1056)	0	normal_concat_12[0][0]
flatten (Flatten)	(None, 51744)	0	activation_187[0][0]
dropout (Dropout)	(None, 51744)	0	flatten[0][0]
batch_normalization (BatchNorma	(None, 51744)	206976	dropout[0][0]
dense (Dense)	(None, 2)	103490	batch_normalization[0][0]
=====			
Total params: 4,580,182			
Trainable params: 4,439,956			
Non-trainable params: 140,226			

Şekil 3.15. D7-BHT veri seti ile eğitilen NasNetMobile modeli parametre gösterimi

9- Xception Modeli: Bu modele giriş katman değeri olarak 224 x 224 x 3 olarak verilmiştir. Modelde tersine derinlemesine ayrılabilir evrişim katmanları kullanılmaktadır. Giriş bölümünde detaylıca tanıtıldığı üzere, ilk başta, 1×1 noktasal evrişim işlemi girdiye uygulandığında, derinlemesine evrişim işlemi gerçekleşmektedir. Xception mimarisi 14 modülden ve 36 evrişim katmanından oluşmaktadır. Havuzlama katmanı olmak üzere, bir 512 tam bağlı katman ve sınıflandırma yapılacak olan dataya ait sınıf sayısına göre bir çıktı katmanından oluşmaktadır. Modelin çıktı katmanı olarak veri setinin sınıf sayısı olan 2 değeri nöron olarak verilmiş ve aktivasyon fonksiyonu olarak ‘Softmax’ belirlenmiştir. Modelin veri setiyle eğitimleri sırasında; ‘batch_size’ değeri 16, ‘optimizer’ olarak ‘Adam’ ve ‘loss’ fonksiyon olarak da ‘sparse_categorical_crossentropy’ kullanılmıştır. Sırasıyla, 3 kez çapraz doğrulama için 5 epoch, 5 kez çapraz doğrulama için 4 epoch ve 10 kez çapraz doğrulama için de 3 epoch değerleri uygulanmıştır. Şekil 3.24’de görüldüğü gibi, 21,208,362 parametre ile model eğitilmiştir.

block13_sepconv2_act (Activatio	(None, 14, 14, 728)	0	block13_sepconv1_bn[0][0]
block13_sepconv2 (SeparableConv	(None, 14, 14, 1024)	752024	block13_sepconv2_act[0][0]
block13_sepconv2_bn (BatchNorma	(None, 14, 14, 1024)	4096	block13_sepconv2[0][0]
conv2d_3 (Conv2D)	(None, 7, 7, 1024)	745472	add_10[0][0]
block13_pool (MaxPooling2D)	(None, 7, 7, 1024)	0	block13_sepconv2_bn[0][0]
batch_normalization_4 (BatchNor	(None, 7, 7, 1024)	4096	conv2d_3[0][0]
add_11 (Add)	(None, 7, 7, 1024)	0	block13_pool[0][0] batch_normalization_4[0][0]
block14_sepconv1 (SeparableConv	(None, 7, 7, 1536)	1582080	add_11[0][0]
block14_sepconv1_bn (BatchNorma	(None, 7, 7, 1536)	6144	block14_sepconv1[0][0]
block14_sepconv1_act (Activatio	(None, 7, 7, 1536)	0	block14_sepconv1_bn[0][0]
block14_sepconv2 (SeparableConv	(None, 7, 7, 2048)	3159552	block14_sepconv1_act[0][0]
block14_sepconv2_bn (BatchNorma	(None, 7, 7, 2048)	8192	block14_sepconv2[0][0]
block14_sepconv2_act (Activatio	(None, 7, 7, 2048)	0	block14_sepconv2_bn[0][0]
flatten_1 (Flatten)	(None, 100352)	0	block14_sepconv2_act[0][0]
dropout_1 (Dropout)	(None, 100352)	0	flatten_1[0][0]
batch_normalization_5 (BatchNor	(None, 100352)	401408	dropout_1[0][0]
dense_1 (Dense)	(None, 2)	200706	batch_normalization_5[0][0]

Total params: 21,463,594			
Trainable params: 21,208,362			
Non-trainable params: 255,232			

Şekil 3.16. D7-BHT veri seti ile eğitilen Xception modeli parametre gösterimi

3.1 Bitki Hastalıklarının Sınıflandırılması

Bitki hastalıklarının sınıflandırılması problemi için, bölüm 3'te detaylı olarak anlatılmış olan, 25 sınıftan oluşan hastalıklı bitkilere ait resimler içeren, D8-BH8 veri seti kullanılarak 9 farklı model geliştirilmeye çalışılmıştır. Her bir model, aşağıda anlatılmış olup, model mimarileri, 'Imagenet' veri kümesi için önceden eğitilmiş ağırlık değerleriyle yüklenmiştir, ve herhangi bir değişiklik yapılmamıştır. Sadece, çıktı katmanları 25 sınıf nöronuna göre şekillendirilmiştir.

- 1- VGG16 Model:** Bu model, 13 konvolüsyon ve 3 tam bağlı katman olmak üzere 16 katmandan oluşmaktadır. Modelde, 224 x 224 x 3 boyutlarındaki resimler veri girişi olarak uygulanmıştır. Tam bağlı katman öncesi özellik çıkarma işlemleri yapılarak boyutları 7 x 7 x 512 olarak değiştirilmiştir. Bağlı katmanlar sonrasında, modelin çıktı katmanı olarak veri setinin sınıf sayısı olan 25 değeri, nöron olarak verilmiş ve aktivasyon fonksiyonu olarak 'Softmax' seçilmiştir. Modelin veri setiyle eğitimleri sırasında, 'batch_size' değeri 64, 'optimizer' olarak 'Adam' ve 'loss' fonksiyon olarak da 'sparse_categorical_crossentropy' kullanılmıştır. Sırasıyla, 3 kez çapraz doğrulama için 20 epoch, 5 kez çapraz doğrulama için 10 epoch ve 10 kez çapraz doğrulama için de 5 epoch değerleri uygulanmıştır. Şekil 3.25'de görüldüğü gibi, 15,341,913 parametre ile model eğitilmiştir.

Layer (type)	Output Shape	Param #
input_1 (InputLayer)	[(None, 224, 224, 3)]	0
block1_conv1 (Conv2D)	(None, 224, 224, 64)	1792
block1_conv2 (Conv2D)	(None, 224, 224, 64)	36928
block1_pool (MaxPooling2D)	(None, 112, 112, 64)	0
block2_conv1 (Conv2D)	(None, 112, 112, 128)	73856
block2_conv2 (Conv2D)	(None, 112, 112, 128)	147584
block2_pool (MaxPooling2D)	(None, 56, 56, 128)	0
block3_conv1 (Conv2D)	(None, 56, 56, 256)	295168
block3_conv2 (Conv2D)	(None, 56, 56, 256)	590080
block3_conv3 (Conv2D)	(None, 56, 56, 256)	590080
block3_pool (MaxPooling2D)	(None, 28, 28, 256)	0
block4_conv1 (Conv2D)	(None, 28, 28, 512)	1180160
block4_conv2 (Conv2D)	(None, 28, 28, 512)	2359808
block4_conv3 (Conv2D)	(None, 28, 28, 512)	2359808
block4_pool (MaxPooling2D)	(None, 14, 14, 512)	0
block5_conv1 (Conv2D)	(None, 14, 14, 512)	2359808
block5_conv2 (Conv2D)	(None, 14, 14, 512)	2359808
block5_conv3 (Conv2D)	(None, 14, 14, 512)	2359808
block5_pool (MaxPooling2D)	(None, 7, 7, 512)	0
flatten (Flatten)	(None, 25088)	0
dropout (Dropout)	(None, 25088)	0
dense (Dense)	(None, 25)	627225
Total params: 15,341,913		
Trainable params: 15,341,913		
Non-trainable params: 0		

Şekil 3.17. D8-BHS veri seti ile eğitilen VGGNET16 modeli

2- VGG19 Model: Bu model 16 konvolüsyon ve 3 tam bağlı katman olmak üzere 19 katmandan oluşmaktadır. Modelde, 224 x 224 x 3 boyutlarındaki resimler veri girişi olarak verilmiştir. Tam bağlı katman öncesi özellik çıkarma işlemleri yapılarak boyutları 7 x 7 x 512 olarak değiştirilmiştir. Bağlı katmanlar sonrasında modelin çıktı katmanı olarak veri setinin sınıf sayısı olan 25 değeri, nöron olarak verilmiş ve aktivasyon fonksiyonu olarak ‘Softmax’ seçilmiştir. Modelin veri setiyle eğitimleri sırasında; ‘batch_size değeri’ 64, ‘optimizer’ olarak, ‘Adam’ ve ‘loss’ fonksiyon olarak da ‘sparse_categorical_crossentropy’ kullanılmıştır. Sırasıyla, 3 kez çapraz doğrulama için 14 epoch, 5 kez çapraz doğrulama için 10 epoch ve 10 kez çapraz doğrulama için de 5 epoch değerleri uygulanmıştır. Şekil 3.26’da görüldüğü gibi, 20,651,609 parametre ile model eğitilmiştir.

Layer (type)	Output Shape	Param #
input_2 (InputLayer)	[(None, 224, 224, 3)]	0
block1_conv1 (Conv2D)	(None, 224, 224, 64)	1792
block1_conv2 (Conv2D)	(None, 224, 224, 64)	36928
block1_pool (MaxPooling2D)	(None, 112, 112, 64)	0
block2_conv1 (Conv2D)	(None, 112, 112, 128)	73856
block2_conv2 (Conv2D)	(None, 112, 112, 128)	147584
block2_pool (MaxPooling2D)	(None, 56, 56, 128)	0
block3_conv1 (Conv2D)	(None, 56, 56, 256)	295168
block3_conv2 (Conv2D)	(None, 56, 56, 256)	590800
block3_conv3 (Conv2D)	(None, 56, 56, 256)	590800
block3_conv4 (Conv2D)	(None, 56, 56, 256)	590800
block3_pool (MaxPooling2D)	(None, 28, 28, 256)	0
block4_conv1 (Conv2D)	(None, 28, 28, 512)	1180160
block4_conv2 (Conv2D)	(None, 28, 28, 512)	2359808
block4_conv3 (Conv2D)	(None, 28, 28, 512)	2359808
block4_conv4 (Conv2D)	(None, 28, 28, 512)	2359808
block4_pool (MaxPooling2D)	(None, 14, 14, 512)	0
block5_conv1 (Conv2D)	(None, 14, 14, 512)	2359808
block5_conv2 (Conv2D)	(None, 14, 14, 512)	2359808
block5_conv3 (Conv2D)	(None, 14, 14, 512)	2359808
block5_conv4 (Conv2D)	(None, 14, 14, 512)	2359808
block5_pool (MaxPooling2D)	(None, 7, 7, 512)	0
flatten_1 (Flatten)	(None, 25088)	0
dropout_1 (Dropout)	(None, 25088)	0
dense_1 (Dense)	(None, 25)	627225
Total params: 20,651,609		
Trainable params: 20,651,609		
Non-trainable params: 0		

Şekil 3.18. D8-BHS veri seti ile eğitilen VGGNET19 modeli

3- AlexNet Model: Bu modelin giriş değeri 227 x 227 x 3 olarak verilmiştir. AlexNet modeli toplamda 5 konvolüsyonel katmandan oluşmakta olup ‘padding’ olarak ‘same’ değeri verilerek, modelin görüntüler üzerindeki tarama boyutu standart tutulmuştur. Katmanlar arasında kullanılan aktivasyon fonksiyonları ReLu’dur. 3 adet ‘Max pooling’ katmanı yer almakta ve 5 konvolüsyonel katmandan sonra da 3 ‘Fully connected’ katman kullanılmaktadır. Çıkış katmanı olarak veri setinin sınıf sayısı olan 25 nöron sayısı verilmiştir, ve aktivasyon fonksiyonu olarak ‘Softmax’ seçilmiştir. Modelin veri setiyle eğitimleri sırasında; ‘batch_size’ değeri 16, ‘optimizer’ olarak ‘Adam’ ve ‘loss’ fonksiyon olarak da ‘sparse_categorical_crossentropy’ kullanılmıştır. Sırasıyla, 3 kez çapraz doğrulama için 100 epoch, 5 kez çapraz doğrulama için 80 epoch ve 10 kez çapraz doğrulama için de 60 epoch değerleri uygulanmıştır. Şekil 3.27’de görüldüğü gibi, 28,868,537 parametre ile model eğitilmiştir.

batch_normalization_13 (Batc	(None, 11, 11, 256)	1024
conv2d_12 (Conv2D)	(None, 9, 9, 384)	885120
conv2d_13 (Conv2D)	(None, 7, 7, 384)	1327488
conv2d_14 (Conv2D)	(None, 5, 5, 256)	884992
max_pooling2d_8 (MaxPooling2	(None, 2, 2, 256)	0
batch_normalization_14 (Batc	(None, 2, 2, 256)	1024
flatten_2 (Flatten)	(None, 1024)	0
dense_8 (Dense)	(None, 4096)	4198400
dropout_6 (Dropout)	(None, 4096)	0
batch_normalization_15 (Batc	(None, 4096)	16384
dense_9 (Dense)	(None, 4096)	16781312
dropout_7 (Dropout)	(None, 4096)	0
batch_normalization_16 (Batc	(None, 4096)	16384
dense_10 (Dense)	(None, 1000)	4097000
dropout_8 (Dropout)	(None, 1000)	0
batch_normalization_17 (Batc	(None, 1000)	4000
dense_11 (Dense)	(None, 25)	25025
=====		
Total params: 28,888,137		
Trainable params: 28,868,537		
Non-trainable params: 19,600		

Şekil 3.19. D8-BHS veri seti ile eğitilen AlexNet modeli

- 4- ResNet50:** Bu model 177 katmandan oluşmaktadır ve giriş değeri 224 x 224 x 3 olarak uygulanmıştır. Bu modelin ResNet50 diye bilinmesinin sebebi içerisinde 50 katmandan oluşan konvolüsyon katmanlarının olmasıdır. Modelin çıkış değeri olarak veri setinin 25 sınıflı olmasından ötürü, 25 nöron olarak belirlenmiştir. Modelin veri setiyle eğitimleri sırasında ‘batch_size’ değeri olarak 64, ‘optimizer’ olarak ‘Adam’ ve ‘loss’ fonksiyon olarak da ‘sparse_categorical_crossentropy’ kullanılmıştır. Sırasıyla, 3 kez çapraz doğrulama için 10 epoch, 5 kez çapraz doğrulama için 5 epoch ve 10 kez çapraz doğrulama için de 3 epoch değerleri uygulanmıştır. Şekil 3.28’de görüldüğü gibi, 26,244,121 parametre ile model eğitilmiştir.

conv5_block2_out (Activation)	(None, 7, 7, 2048)	0	conv5_block2_add[0][0]
conv5_block3_1_conv (Conv2D)	(None, 7, 7, 512)	1049088	conv5_block2_out[0][0]
conv5_block3_1_bn (BatchNormali	(None, 7, 7, 512)	2048	conv5_block3_1_conv[0][0]
conv5_block3_1_relu (Activation	(None, 7, 7, 512)	0	conv5_block3_1_bn[0][0]
conv5_block3_2_conv (Conv2D)	(None, 7, 7, 512)	2359808	conv5_block3_1_relu[0][0]
conv5_block3_2_bn (BatchNormali	(None, 7, 7, 512)	2048	conv5_block3_2_conv[0][0]
conv5_block3_2_relu (Activation	(None, 7, 7, 512)	0	conv5_block3_2_bn[0][0]
conv5_block3_3_conv (Conv2D)	(None, 7, 7, 2048)	1050624	conv5_block3_2_relu[0][0]
conv5_block3_3_bn (BatchNormali	(None, 7, 7, 2048)	8192	conv5_block3_3_conv[0][0]
conv5_block3_add (Add)	(None, 7, 7, 2048)	0	conv5_block2_out[0][0] conv5_block3_3_bn[0][0]
conv5_block3_out (Activation)	(None, 7, 7, 2048)	0	conv5_block3_add[0][0]
flatten (Flatten)	(None, 100352)	0	conv5_block3_out[0][0]
dropout (Dropout)	(None, 100352)	0	flatten[0][0]
batch_normalization (BatchNorma	(None, 100352)	401408	dropout[0][0]
dense (Dense)	(None, 25)	2508825	batch_normalization[0][0]
=====			
Total params: 26,497,945			
Trainable params: 26,244,121			
Non-trainable params: 253,824			

Şekil 3.20. D8-BHS veri seti ile eğitilen ResNet50 modeli parametre gösterimi

5- DenseNet: Bu modelde girdi olarak 224 x 224 x 3 değeri verilmiştir. 5 blok konvolüsyon katmanı, 3 blok ‘transition’ katmanı (6, 12, 32), 1 blok sınıflandırma katmanı (32’li) ve son olarak da 2 blok ‘DenseBlok’ katmanı bulunmaktadır. Modelin çıktı katmanı olarak veri setinin sınıf sayısı olan 25 değeri, nöron olarak verilmiş ve aktivasyon fonksiyonu olarak ‘Softmax’ seçilmiştir. Tüm bu bloklar içerisinde yer alanlar katmanlar toplandığı durumda, DenseNet169’un katman sayısı olan 169’u vermektedir. Modelin veri setiyle eğitimleri sırasında; ‘batch_size’ değeri 16, ‘optimizer’ olarak ‘Adam’ ve ‘loss’ fonksiyon olarak da ‘sparse_categorical_crossentropy’ kullanılmıştır. Sırasıyla, 3 kez çapraz doğrulama için 12 epoch, 5 kez çapraz doğrulama için 10 epoch ve 10 kez çapraz doğrulama için de 8 epoch değerleri uygulanmıştır. Şekil 3.29’da görüldüğü gibi, 14,685,977 parametre ile model eğitilmiştir.

conv5_block32_0_bn (BatchNormal	(None, 7, 7, 1632)	6528	conv5_block31_concat[0][0]
conv5_block32_0_relu (Activatio	(None, 7, 7, 1632)	0	conv5_block32_0_bn[0][0]
conv5_block32_1_conv (Conv2D)	(None, 7, 7, 128)	208896	conv5_block32_0_relu[0][0]
conv5_block32_1_bn (BatchNormal	(None, 7, 7, 128)	512	conv5_block32_1_conv[0][0]
conv5_block32_1_relu (Activatio	(None, 7, 7, 128)	0	conv5_block32_1_bn[0][0]
conv5_block32_2_conv (Conv2D)	(None, 7, 7, 32)	36864	conv5_block32_1_relu[0][0]
conv5_block32_concat (Concatena	(None, 7, 7, 1664)	0	conv5_block31_concat[0][0] conv5_block32_2_conv[0][0]
bn (BatchNormalization)	(None, 7, 7, 1664)	6656	conv5_block32_concat[0][0]
relu (Activation)	(None, 7, 7, 1664)	0	bn[0][0]
flatten (Flatten)	(None, 81536)	0	relu[0][0]
dropout (Dropout)	(None, 81536)	0	flatten[0][0]
batch_normalization (BatchNorma	(None, 81536)	326144	dropout[0][0]
dense (Dense)	(None, 25)	2038425	batch_normalization[0][0]
=====			
Total params: 15,007,449			
Trainable params: 14,685,977			
Non-trainable params: 321,472			

Şekil 3.21. D8-BHS veri seti ile eğitilen DenseNet modeli parametre gösterimi

6- GoogleNet / Inception: Bu modelin girdi katman değeri 224 x 224 x 3 olarak belirlenmiştir. 22 katman derinliğe sahip olup GoogleNet toplamda 144 katmandan oluşan bir yapısı bulunmaktadır. Modelin çıktı katmanı olarak veri setinin sınıf sayısı olan 25 değeri, nöron olarak verilmiş ve aktivasyon fonksiyonu olarak ‘Softmax’ seçilmiştir. Modelin veri setiyle eğitimleri sırasında; ‘batch_size’ değeri 64, ‘optimizer’ olarak ‘Adam’ ve ‘loss’ fonksiyon olarak da ‘sparse_categorical_crossentropy’ kullanılmıştır. Sırasıyla, 3 kez çapraz doğrulama için 10 epoch, 5 kez çapraz doğrulama için 8 epoch ve 10 kez çapraz doğrulama için de 5 epoch değerleri uygulanmıştır. Şekil 3.30’da görüldüğü gibi, 23,150,777 parametre ile model eğitilmiştir.

activation_467 (Activation)	(None, 5, 5, 384)	0	batch_normalization_471[0][0]
activation_468 (Activation)	(None, 5, 5, 384)	0	batch_normalization_472[0][0]
batch_normalization_473 (BatchN	(None, 5, 5, 192)	576	conv2d_469[0][0]
activation_461 (Activation)	(None, 5, 5, 320)	0	batch_normalization_465[0][0]
mixed9_1 (Concatenate)	(None, 5, 5, 768)	0	activation_463[0][0] activation_464[0][0]
concatenate_9 (Concatenate)	(None, 5, 5, 768)	0	activation_467[0][0] activation_468[0][0]
activation_469 (Activation)	(None, 5, 5, 192)	0	batch_normalization_473[0][0]
mixed10 (Concatenate)	(None, 5, 5, 2048)	0	activation_461[0][0] mixed9_1[0][0] concatenate_9[0][0] activation_469[0][0]
flatten_4 (Flatten)	(None, 51200)	0	mixed10[0][0]
dropout_4 (Dropout)	(None, 51200)	0	flatten_4[0][0]
batch_normalization_474 (BatchN	(None, 51200)	204800	dropout_4[0][0]
dense_4 (Dense)	(None, 25)	1280025	batch_normalization_474[0][0]
Total params: 23,287,609			
Trainable params: 23,150,777			
Non-trainable params: 136,832			

Şekil 3.22. D8-BHS veri seti ile eğitilen GoogleNet / InceptionV3 modeli parametre gösterimi

7- MobileNet: Bu model, girdi olarak 224 x 224 x 3 olarak uygulanmıştır. Giriş bölümünde detaylıca tanıtılan bu model içerisinde, derinlemesine ayrılabilir evrişim katmanları yer almakta ve uzamsal korelasyonları haritalandırabilmektedir. MobileNet, derinlik ve noktasal dönüşümler ayrı katmanlar olarak değerlendirildiğinde toplam 28 katman yer almaktadır. Modelin çıktı katmanı olarak veri setinin sınıf sayısı olan 25 değeri, nöron olarak verilmiş ve aktivasyon fonksiyonu olarak ‘Softmax’ seçilmiştir. Modelin veri setiyle eğitimleri sırasında; ‘batch_size’ değeri 64, ‘optimizer’ olarak ‘Adam’ ve ‘loss’ fonksiyon olarak da ‘sparse_categorical_crossentropy’ kullanılmıştır. Sırasıyla, 3 kez çapraz doğrulama için 9 epoch, 5 kez çapraz doğrulama için 5 epoch ve 10 kez çapraz doğrulama için de 3 epoch değerleri uygulanmıştır. Şekil 3.31’de görüldüğü gibi, 4,561,753 parametre ile model eğitilmiştir.

conv_dw_12_relu (ReLU)	(None, 7, 7, 512)	0
conv_pw_12 (Conv2D)	(None, 7, 7, 1024)	524288
conv_pw_12_bn (BatchNormaliz	(None, 7, 7, 1024)	4096
conv_pw_12_relu (ReLU)	(None, 7, 7, 1024)	0
conv_dw_13 (DepthwiseConv2D)	(None, 7, 7, 1024)	9216
conv_dw_13_bn (BatchNormaliz	(None, 7, 7, 1024)	4096
conv_dw_13_relu (ReLU)	(None, 7, 7, 1024)	0
conv_pw_13 (Conv2D)	(None, 7, 7, 1024)	1048576
conv_pw_13_bn (BatchNormaliz	(None, 7, 7, 1024)	4096
conv_pw_13_relu (ReLU)	(None, 7, 7, 1024)	0
flatten_14 (Flatten)	(None, 50176)	0
dropout_14 (Dropout)	(None, 50176)	0
batch_normalization_766 (Bat	(None, 50176)	200704
dense_14 (Dense)	(None, 25)	1254425
=====		
Total params: 4,683,993		
Trainable params: 4,561,753		
Non-trainable params: 122,240		

Şekil 3.23. D8-BHS veri seti ile eğitilen MobileNet modeli parametre gösterimi

- 8- NasNetMobile Model:** Bu modele girdi olarak 224 x 224 x 3 olarak uygulanmıştır. Modelin çıktı katmanı olarak veri setinin sınıf sayısı olan 25 değeri, nöron olarak verilmiş ve aktivasyon fonksiyonu olarak ‘Softmax’ seçilmiştir. Modelin veri setiyle eğitimleri sırasında; ‘batch_size’ değeri 16, ‘optimizer’ olarak ‘Adam’ ve ‘loss’ fonksiyon olarak da ‘sparse_categorical_crossentropy’ kullanılmıştır. Sırasıyla, 3 kez çapraz doğrulama için 11 epoch, 5 kez çapraz doğrulama için 9 epoch ve 10 kez çapraz doğrulama için de 10 epoch değerleri uygulanmıştır. Şekil 3.32’de görüldüğü gibi, 5,630,091 parametre ile model eğitilmiştir.

normal_right4_12 (AveragePoolin	(None, 7, 7, 176)	0	adjust_bn_12[0][0]
separable_conv_2_bn_normal_left	(None, 7, 7, 176)	704	separable_conv_2_normal_left5_12[
normal_add_1_12 (Add)	(None, 7, 7, 176)	0	separable_conv_2_bn_normal_left1_ separable_conv_2_bn_normal_right1
normal_add_2_12 (Add)	(None, 7, 7, 176)	0	separable_conv_2_bn_normal_left2_ separable_conv_2_bn_normal_right2
normal_add_3_12 (Add)	(None, 7, 7, 176)	0	normal_left3_12[0][0] adjust_bn_12[0][0]
normal_add_4_12 (Add)	(None, 7, 7, 176)	0	normal_left4_12[0][0] normal_right4_12[0][0]
normal_add_5_12 (Add)	(None, 7, 7, 176)	0	separable_conv_2_bn_normal_left5_ normal_bn_1_12[0][0]
normal_concat_12 (Concatenate)	(None, 7, 7, 1056)	0	adjust_bn_12[0][0] normal_add_1_12[0][0] normal_add_2_12[0][0] normal_add_3_12[0][0] normal_add_4_12[0][0] normal_add_5_12[0][0]
activation_563 (Activation)	(None, 7, 7, 1056)	0	normal_concat_12[0][0]
flatten_2 (Flatten)	(None, 51744)	0	activation_563[0][0]
dropout_2 (Dropout)	(None, 51744)	0	flatten_2[0][0]
batch_normalization_2 (BatchNor	(None, 51744)	206976	dropout_2[0][0]
dense_2 (Dense)	(None, 25)	1293625	batch_normalization_2[0][0]
=====			
Total params: 5,770,317			
Trainable params: 5,630,091			
Non-trainable params: 140,226			

Şekil 3.24. D8-BHS veri seti ile eğitilen NasNetMobile modeli parametre gösterimi

9- Xception Model: Bu modele girdi olarak 224 x 224 x 3 olarak uygulanmıştır. Modelde tersine derinlemesine ayrılabilir evrişim katmanları kullanılmıştır. 1×1 noktasal evrişim işlemi girdiye uygulandığında, derinlemesine evrişim işlemi gerçekleşmektedir. Xception mimarisi 14 modülden ve 36 evrişim katmanından oluşmaktadır. Havuzlama katmanı olmak üzere, bir 512 tam bağlı katman ve sınıflandırma yapılacak olan veri setine ait sınıf sayısına göre bir çıktı katmanından oluşmaktadır. Modelin çıktı katmanı olarak veri setinin sınıf sayısı olan 25 değeri, nöron olarak verilmiş ve aktivasyon fonksiyonu olarak ‘Softmax’ seçilmiştir. Modelin veri setiyle eğitimleri sırasında; ‘batch_size’ değeri 16, ‘optimizer’ olarak ‘Adam’ ve ‘loss’ fonksiyon olarak da ‘sparse_categorical_crossentropy’ kullanılmıştır. Sırasıyla, 3 kez çapraz doğrulama için 5 epoch, 5 kez çapraz doğrulama için 4 epoch ve 10 kez çapraz doğrulama için de 3 epoch değerleri uygulanmıştır. Şekil 3.33’de görüldüğü gibi, 23,516,481 parametre ile model eğitilmiştir.

block13_sepconv2_bn (BatchNorma	(None, 14, 14, 1024)	4096	block13_sepconv2[0][0]
conv2d_3 (Conv2D)	(None, 7, 7, 1024)	745472	add_10[0][0]
block13_pool (MaxPooling2D)	(None, 7, 7, 1024)	0	block13_sepconv2_bn[0][0]
batch_normalization_3 (BatchNor	(None, 7, 7, 1024)	4096	conv2d_3[0][0]
add_11 (Add)	(None, 7, 7, 1024)	0	block13_pool[0][0] batch_normalization_3[0][0]
block14_sepconv1 (SeparableConv	(None, 7, 7, 1536)	1582080	add_11[0][0]
block14_sepconv1_bn (BatchNorma	(None, 7, 7, 1536)	6144	block14_sepconv1[0][0]
block14_sepconv1_act (Activatio	(None, 7, 7, 1536)	0	block14_sepconv1_bn[0][0]
block14_sepconv2 (SeparableConv	(None, 7, 7, 2048)	3159552	block14_sepconv1_act[0][0]
block14_sepconv2_bn (BatchNorma	(None, 7, 7, 2048)	8192	block14_sepconv2[0][0]
block14_sepconv2_act (Activatio	(None, 7, 7, 2048)	0	block14_sepconv2_bn[0][0]
flatten (Flatten)	(None, 100352)	0	block14_sepconv2_act[0][0]
dropout (Dropout)	(None, 100352)	0	flatten[0][0]
batch_normalization_4 (BatchNor	(None, 100352)	401408	dropout[0][0]
dense (Dense)	(None, 25)	2508825	batch_normalization_4[0][0]
=====			
Total params: 23,771,713			
Trainable params: 23,516,481			
Non-trainable params: 255,232			

Şekil 3.25. D8-BHS veri seti ile eğitilen Xception modeli parametre gösterimi

3.1 Model Sonuçları ve Sonuçların Değerlendirilmesi

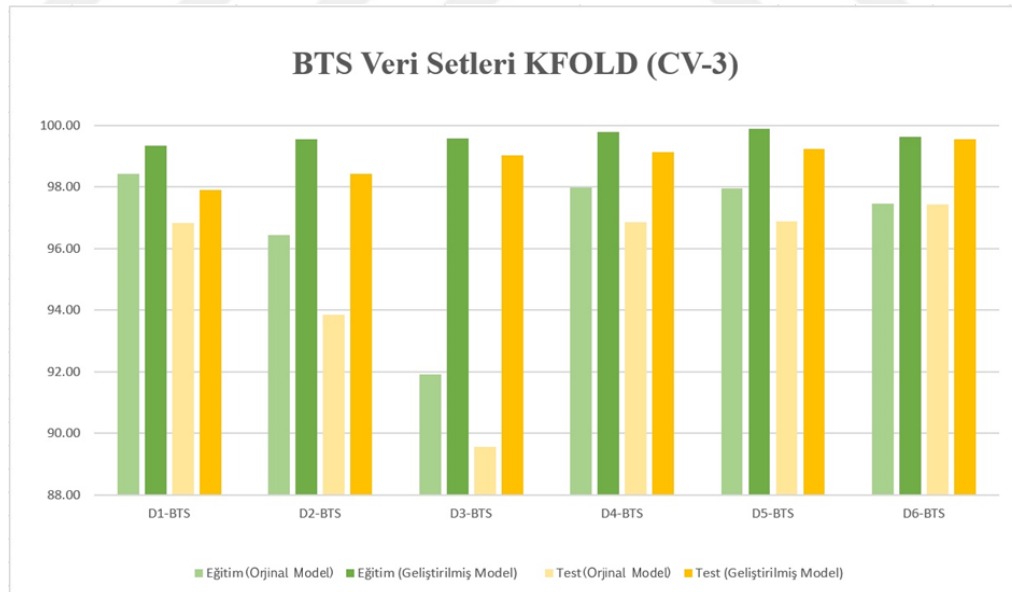
Bitki türlerinin sınıflandırılması, bitki hastalık tahmini ve bitki hastalıklarının sınıflandırılması problemleri için oluşturulan modellerin sonuçları sırasıyla aşağıdaki çizelgelerde verilmiştir. Çizelgelerde yer alan kırmızı renkli alanlar modellerdeki en yüksek başarıma sahip olan değerleri göstermektedir.

Bitki türlerinin sınıflandırılması problemi için 6 ayrı veri seti (D1, 2, 3, 4, 5, 6-BTS), orijinal AlexNet ve transfer öğrenme ile geliştirilmiş AlexNet modelleriyle eğitilmiştir ve 3, 5, ve 10 kat çapraz doğrulama için, sırasıyla Çizelge 3.8, 3.9 ve 3.10'daki sonuçlar elde edilmiştir.

Çizelge 3.8. Bitki türlerinin sınıflandırılması problemi için 3 kat çapraz doğrulama sonuçları

Modeller	Veri Seti (Sağlık)	3-Çapraz Doğrulama			
		Eğitim (Doğruluk, %)	Test (Doğruluk, %)	Epoch	Çalışma Süresi
Orjinal AlexNet	D1-BTS	98.42	96.82	30	7 dk 06s
	D2-BTS	96.44	93.86	30	7 dk 15s
	D3-BTS	91.93	89.57	30	7 dk 1s
	D4-BTS	97.97	96.86	30	5 dk 56s
	D5-BTS	97.96	96.88	30	4 dk 42s
	D6-BTS	97.47	97.43	30	8 dk 32s
Geliştirilmiş AlexNet	D1-BTS	99.35	97.91	26	13 dk 33s
	D2-BTS	99.54	98.42	26	14 dk 54s
	D3-BTS	99.58	99.03	26	9 dk 34s
	D4-BTS	99.78	99.12	50	19 dk 51s
	D5-BTS	99.89	99.23	26	8 dk 24s
	D6-BTS	99.62	99.54	10	6 dk 39s

Şekil 3.26’da bitki türlerinin sınıflandırılması probleminin, orijinal AlexNet modeli ve transfer öğrenme ile geliştirilmiş AlexNet modeline ait 6 veri setinin 3 kez çapraz doğrulama yöntemiyle elde edilmiş sonuçlarına ait grafik yer almaktadır. Bu şekilde görüldüğü üzere, transfer öğrenme ile geliştirilmiş AlexNet modellerinin, orijinal AlexNet modellerinden daha başarılı sonuçlar elde edilmiştir.



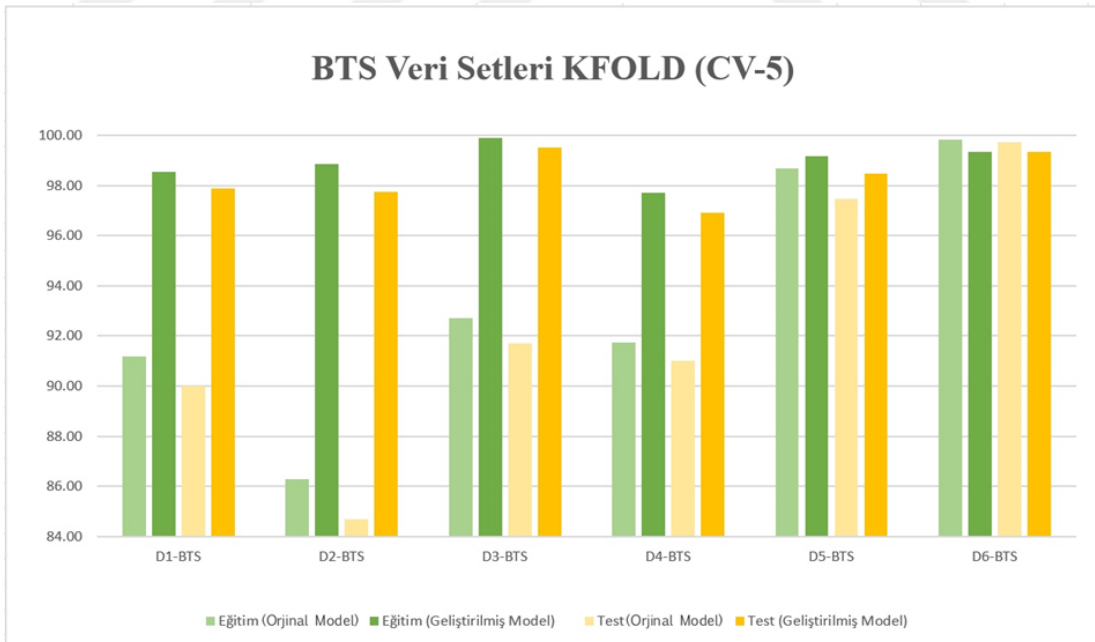
Şekil 3.26. 3 Kat Çapraz Doğrulama ile Bitki Türlerinin Sınıflandırılması için AlexNet Modelleri

Şekil 3.27 ve 3.28’de, bitki türlerinin sınıflandırılması probleminin orijinal AlexNet modeli ve transfer öğrenme ile geliştirilmiş AlexNet modellerine ait, 6 veri setinin 5 ve 10 kat çapraz doğrulama ile elde edilmiş sonuçları yer almaktadır.

Çizelge 3.9. Bitki türlerinin sınıflandırılması problemi için 5 kat çapraz doğrulama sonuçları

Modeller	Veri Seti (Sağlıklı)	5-Çapraz Doğrulama			
		Eğitim (Doğruluk, %)	Test (Doğruluk, %)	Epoch	Çalışma Süresi
Orjinal AlexNet	D1-BTS	91.17	90.00	30	14 dk 2s
	D2-BTS	86.28	84.69	30	15 dk 3s
	D3-BTS	92.70	91.70	30	12 dk 42s
	D4-BTS	91.75	91.00	30	10 dk 54s
	D5-BTS	98.67	97.45	30	8 dk 30s
	D6-BTS	99.82	99.72	30	15 dk 42s
Geliştirilmiş AlexNet	D1-BTS	98.53	97.87	23	22 dk 28s
	D2-BTS	98.87	97.74	6	4 dk 2s
	D3-BTS	99.89	99.52	50	45 dk 19s
	D4-BTS	97.72	96.92	20	15 dk 39s
	D5-BTS	99.16	98.48	21	12 dk 26s
	D6-BTS	99.34	99.34	26	28 dk 43s

Şekil 3.35’deki 5 kez doğrulama ile alınan sonuçlara bakıldığında, D-1,2,3,4,5 BTS veri setleri için, transfer öğrenme ile geliştirilmiş AlexNet modellerinin, orijinal AlexNet modellerine göre daha başarılı olduğu görülmektedir. D6-BTS veri seti içinse, orijinal AlexNet modeli daha başarılı sonuçlar elde edilmiştir.

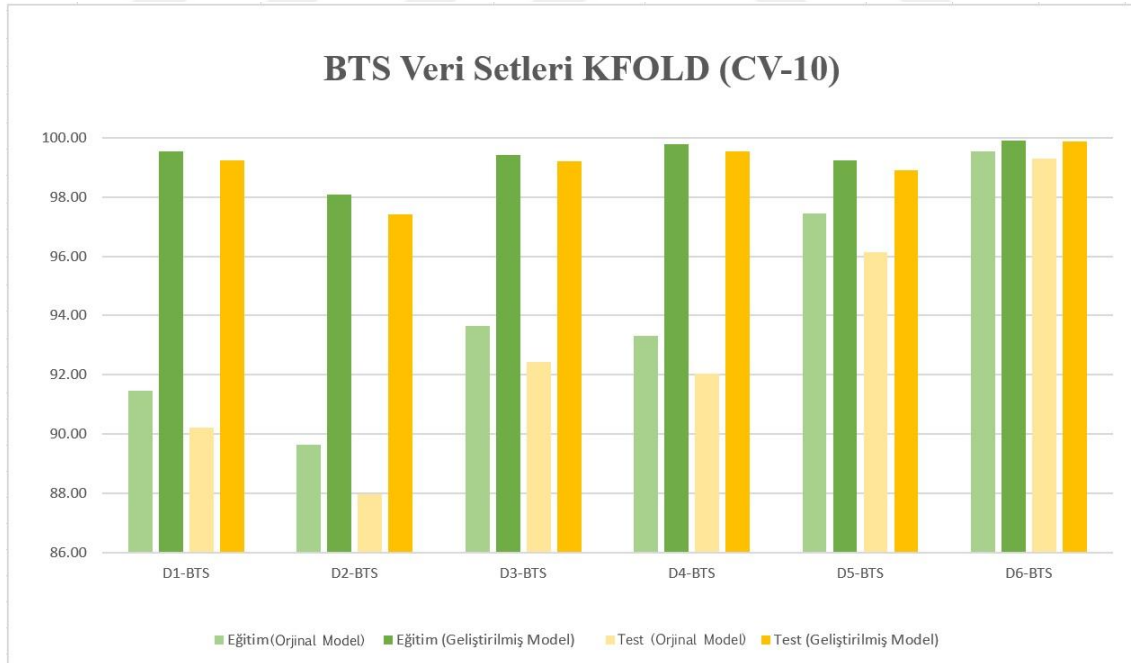


Şekil 3.27. 5 Kat Çapraz Doğrulama ile Bitki Türlerinin Sınıflandırılması için AlexNet Modelleri

Çizelge 3.10. Bitki türlerinin sınıflandırılması problemi için 10 kat çapraz doğrulama sonuçları

Modeller	Veri Seti (Sağlıklı)	10-Çapraz Doğrulama			
		Eğitim (Doğruluk, %)	Test (Doğruluk, %)	Epoch	Çalışma Süresi
Orjinal AlexNet	D1-BTS	91.46	90.22	30	19 dk 3s
	D2-BTS	89.63	87.95	30	31 dk 23s
	D3-BTS	93.64	92.42	30	26 dk 22s
	D4-BTS	93.30	92.03	30	23 dk 13s
	D5-BTS	97.44	96.13	30	39 dk 19s
	D6-BTS	99.56	99.31	30	1h 16 dk 5s
Geliştirilmiş AlexNet	D1-BTS	99.56	99.25	23	49 dk 11s
	D2-BTS	98.08	97.43	20	27 dk 28s
	D3-BTS	99.43	99.21	26	30 dk 7s
	D4-BTS	99.79	99.54	20	34 dk 9s
	D5-BTS	99.25	98.91	20	25 dk 47s
	D6-BTS	99.91	99.88	26	1h 8 dk 50s

Şekil 3.36’da ise 6 veri seti için de transfer öğrenme ile geliştirilmiş AlexNet modeline ait 10 kat çapraz doğrulama yöntemiyle elde edilmiş sonuçların, orijinal AlexNet modellerinden daha başarılı olduğu görülmektedir.



Şekil 3.28. 10 Kat Çapraz Doğrulama ile Bitki Türlerinin Sınıflandırılması için AlexNet Modelleri

Bitki hastalık tahmini problemi için D7-BHT veri seti kullanılarak, 9 mimari (VGG16, VGG19, AlexNet, ResNet, DenseNet, GoogleNet, MobileNet, NasNetMobile ve Xception) ile oluşturulan modellerin 3, 5 ve 10 kat çapraz doğrulama ile elde edilmiş sonuçları, Çizelge 3.11, 3.12 ve 3.13'te yer almaktadır.

Çizelge 3.11. Bitki hastalık tahmini problemi için 3 kat çapraz doğrulama sonuçları

Modeller	3 Kat Çapraz Doğrulama			
	Eğitim (Doğruluk, %)	Test (Doğruluk, %)	Epoch	Çalışma Süresi
VGG16	99.72	99.39	7	55 dk 12s
VGG19	99.89	99.67	7	1h 10 dk 54s
AlexNet	96.01	93.66	27	8 dk 38s
ResNet50	99.90	99.69	5	31 dk 37s
DenseNet	99.83	99.57	5	20 dk 55s
GoogLeNet / Inception	99.91	99.50	6	12 dk 49s
MobileNet	99.81	99.20	6	14 dk 32s
NASNetMobile	93.79	93.44	8	42 dk 48s
Xceptions	96.45	95.91	5	36 dk 12s

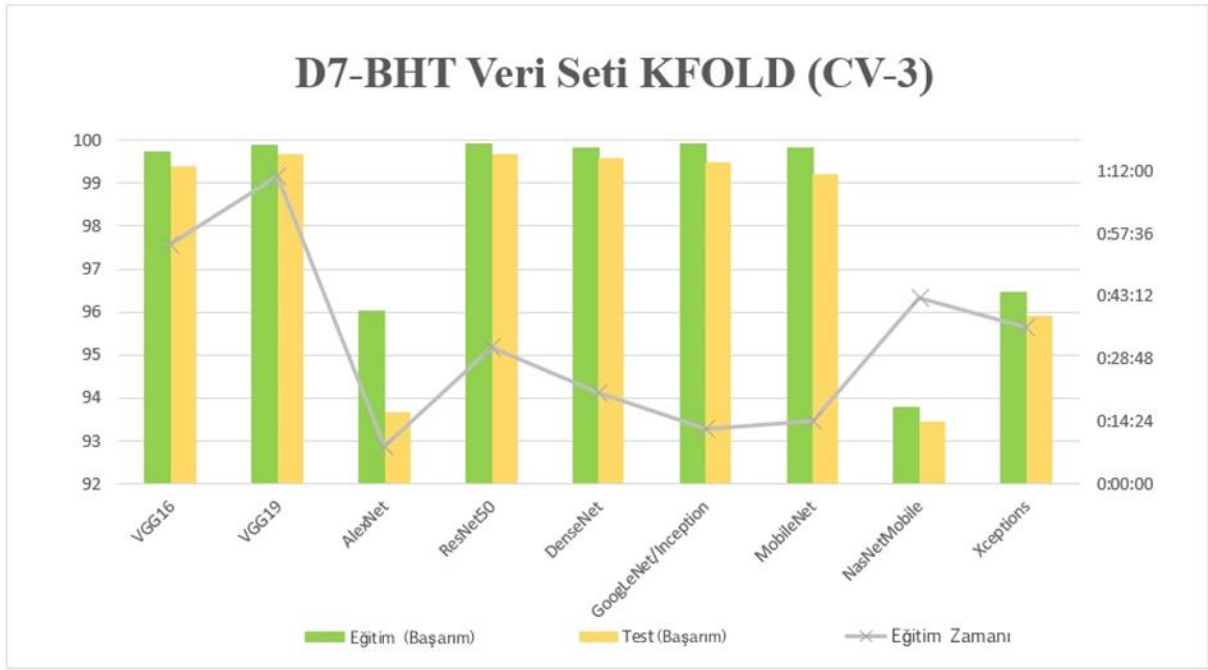
Çizelge 3.12. Bitki hastalık tahmini problemi için 5 kat çapraz doğrulama sonuçları

Modeller	5 Kat Çapraz Doğrulama			
	Eğitim (Doğruluk, %)	Test (Doğruluk, %)	Epoch	Çalışma Süresi
VGG16	99.84	99.53	4	28 dk 10s
VGG19	99.67	99.33	4	33 dk 58s
AlexNet	97.55	95.90	48	24 dk 43s
ResNet50	99.40	99.06	4	23 dk 3s
DenseNet	99.87	99.75	4	31 dk 16s
GoogLeNet / Inception	99.93	99.75	5	23 dk 1s
MobileNet	99.87	99.67	5	23 dk 23s
NASNetMobile	97.33	96.86	8	1h 23 dk 11s
Xceptions	99.68	99.47	4	56 dk 1s

Çizelge 3.13. Bitki hastalık tahmini problemi için 10 kat çapraz doğrulama sonuçları

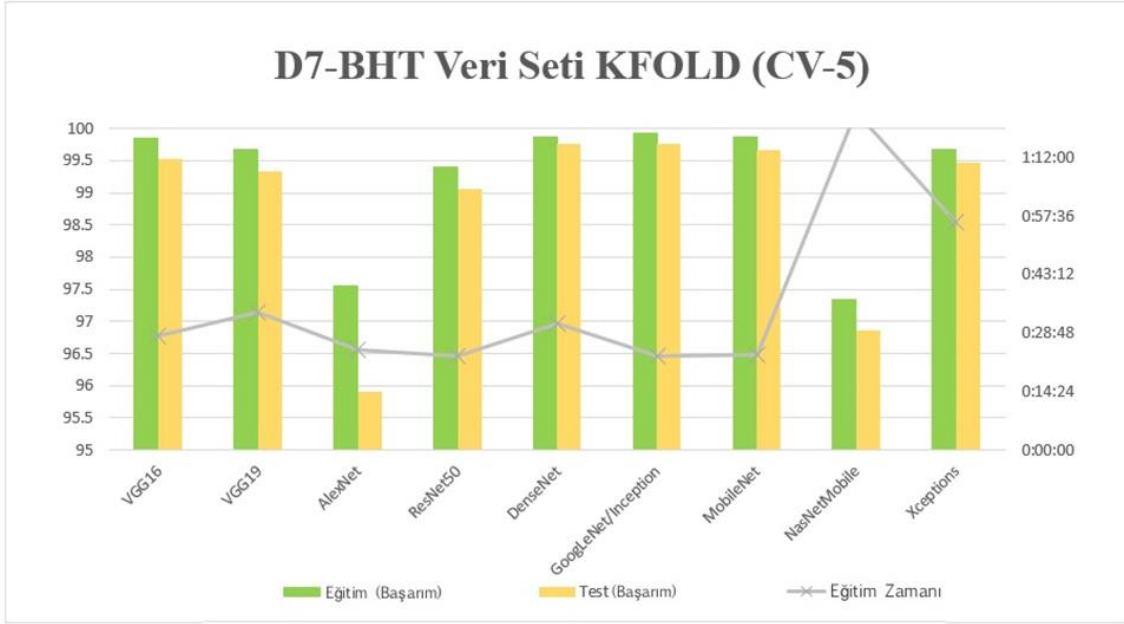
Modeller	10 Kat Çapraz Doğrulama			
	Eğitim (Doğruluk, %)	Test (Doğruluk, %)	Epoch	Çalışma Süresi
VGG16	99.45	99.23	3	46 dk 3s
VGG19	99.64	99.40	3	54 dk 46s
AlexNet	97.52	95.90	48	50 dk 57s
ResNet50	99.07	98.91	4	38 dk 37s
DenseNet	99.62	99.47	3	1h 4 dk 59s
GoogLeNet / Inception	99.70	99.48	3	28 dk 51s
MobileNet	99.64	99.39	3	34 dk 21s
NASNetMobile	95.15	95.22	4	1h 32 dk 51s
Xceptions	97.73	97.46	3	1h 32 dk 51s

Bitki hastalık tahmini problemi için D7-BHT veri seti kullanılarak, oluşturulan 9 modelin 3, 5 ve 10 kat çapraz doğrulama ile elde edilmiş sonuçları (doğruluk başarı ve modelleme süreleri), sırasıyla Şekil 3.37, 3.38 ve 3.39’da grafik olarak gösterilmiştir.



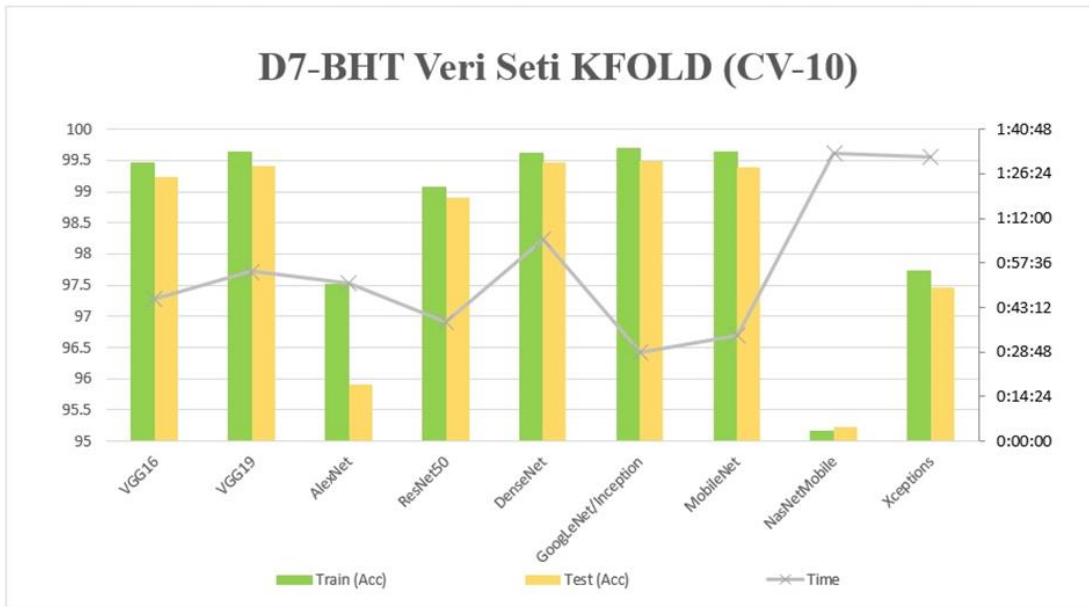
Şekil 3.29. Bitki Hastalık Tahmini Problemi için D7-BHT Veri Seti Kullanılarak 3 Kat Çapraz Doğrulama Model Sonuçları

Şekil 3.29’da görüldüğü üzere, D7-BHT veri seti için, 3 kez çapraz doğrulama ile oluşturulan ResNet50 ve GoogleNet modellerinin test sonuçlarının, %99,69 doğruluk başarıyla en iyi performans olarak elde edilmiştir. Bu modellerin, modelleme sürelerine bakıldığında, ResNet modelinin 31 dakika 37 saniyede, GoogleNet modelinin de 12 dakika 49 saniyede oluşturulduğunu görüyoruz. Bu sonuçlara göre, en yüksek performans ve en kısa modelleme süresi ile GoogleNet modeli, 3 kat çapraz doğrulama sonuçlarına göre, bitki hastalık tahmini yapabilen en iyi model olarak bulunmuştur.



Şekil 3.30. Bitki Hastalık Tahmini Problemi için D7-BHT Veri Seti Kullanılarak 5 Kat Çapraz Doğrulama Model Sonuçları

Şekil 3.30’da görüldüğü üzere, D7-BHT veri seti için, 5 kat çapraz doğrulama ile oluşturulan GoogleNet modelinden, %99,75 test doğruluk başarıyla ve 23 dakika 1 saniye modelleme süresi ile diğer modellere göre daha iyi performans elde edilmiştir.



Şekil 3.31. Bitki Hastalık Tahmini Problemi için D7-BHT Veri Seti Kullanılarak 10 Kat Çapraz Doğrulama Model Sonuçları

Şekil 3.31’de görüldüğü üzere, D7-BHT veri seti için, 10 kez çapraz doğrulama ile oluşturulan GoogleNet modelinden, %99,48 test doğruluk başarısıyla ve 28 dakika 51 saniye modelleme süresi ile diğer modellere göre daha iyi performans elde edilmiştir.

Şekil 3.29, 3.30 ve 3.31 kıyaslandığında, bitki hastalık tahmini yapabilen en iyi model, 5 kat çapraz doğrulama uygulanarak, en yüksek doğruluk başarısı (%99,75) ve en kısa modelleme süresi (23 dakika 1 saniye) ile GoogleNet modeli olarak bulunmuştur. Bitki hastalıklarının sınıflandırılması problemi için D8-BHS veri seti kullanılarak, 9 mimari (VGG16, VGG19, AlexNet, ResNet, DenseNet, GoogleNet, MobileNet, NasNetMobile ve Xception) ile oluşturulan modellerin 3, 5 ve 10 kat çapraz doğrulama ile elde edilmiş sonuçları, Çizelge 3.14, 3.15 ve 3.16’da yer almaktadır.

Çizelge 3.14. Bitki hastalıklarının sınıflandırılması problemi için 3 kat çapraz doğrulama sonuçları

Modeller	3 Kat Çapraz Doğrulama			
	Eğitim (Doğruluk, %)	Test (Doğruluk, %)	Epocs	Çalışma Süresi
VGG16	99.38	96.86	20	59 dk 9s
VGG19	99.33	96.52	14	50 dk 2s
AlexNet	99.68	92.24	100	43 dk 34s
ResNet50	97.23	94.44	10	56 dk 41s
DenseNet	98.26	96.49	12	55 dk 8s
GoogLeNet / Inception	99.19	95.55	10	19 dk 57s
MobileNet	99.30	96.91	9	18 dk 8s
NASNetMobile	93.58	90.54	11	1 s 2 dk 32s
Xceptions	98.40	96.23	5	30 dk 28s

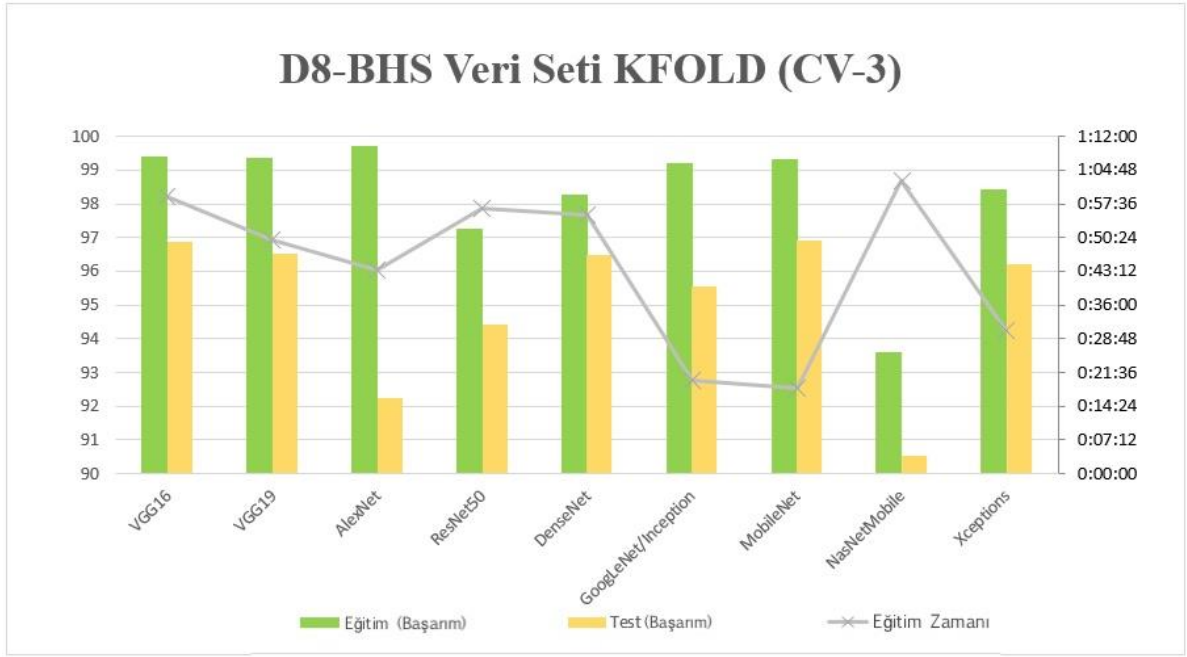
Çizelge 3.15. Bitki hastalıklarının sınıflandırılması problemi için 5 kat çapraz doğrulama sonuçları

Modeller	5 Kat Çapraz Doğrulama			
	Eğitim (Doğruluk, %)	Test (Doğruluk, %)	Epocs	Çalışma Süresi
VGG16	99.24	97.77	10	56 dk 38s
VGG19	98.99	97.29	10	1 s 6 dk 17s
AlexNet	97.41	89.12	80	1 s 3 dk 56s
ResNet50	97.51	95.63	5	24 dk 26s
DenseNet	99.41	98.50	10	1 s 18 dk 15s
GoogLeNet / Inception	97.95	95.95	8	30 dk 18s
MobileNet	98.84	97.01	5	19 dk 39s
NASNetMobile	91.69	89.57	9	1 s 39 dk 9s
Xceptions	99.29	97.94	4	47 dk 58s

Çizelge 3.16. Bitki hastalıklarının sınıflandırılması problemi için 10 kat çapraz doğrulama sonuçları

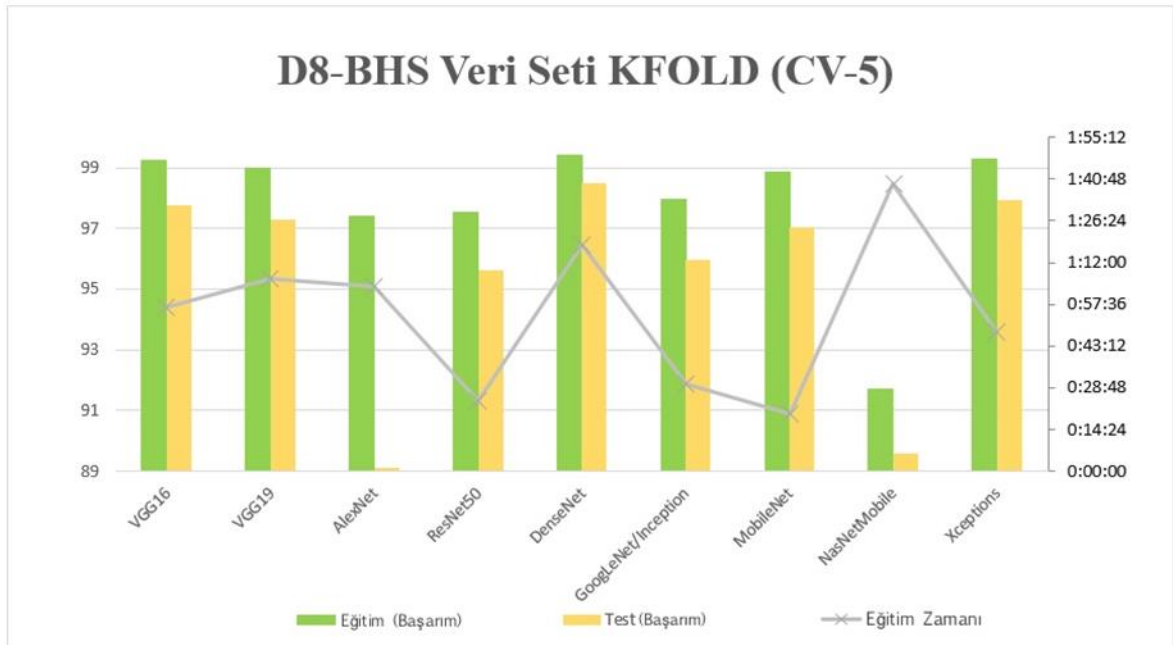
Modeller	10 Kat Çapraz Doğrulama			
	Eğitim (Doğruluk, %)	Test (Doğruluk, %)	Epos	Çalışma Süresi
VGG16	98.73	97.67	5	1 s 2 dk 52s
VGG19	98.33	97.29	5	1 s 14 dk 30s
AlexNet	96.74	89.22	60	1 s 42 dk 10s
ResNet50	97.21	95.92	3	33 dk 24s
DenseNet	99.57	99.14	8	2 s 30 dk 27s
GoogLeNet / Inception	98.98	97.95	5	42 dk 17s
MobileNet	98.62	97.52	3	27 dk 5s
NASNetMobile	91.15	89.92	10	3 s 8 dk 28s
Xceptions	99.07	98.25	3	1 s 20 dk 5s

Bitki hastalıklarının sınıflandırılması için, D8-BHS veri seti kullanılarak, oluşturulan 9 modelin 3, 5 ve 10 kat çapraz doğrulama ile elde edilmiş sonuçları (doğruluk başarı ve modelleme süreleri), sırasıyla Şekil 3.32, 3.33 ve 3.34'te grafik olarak gösterilmiştir.



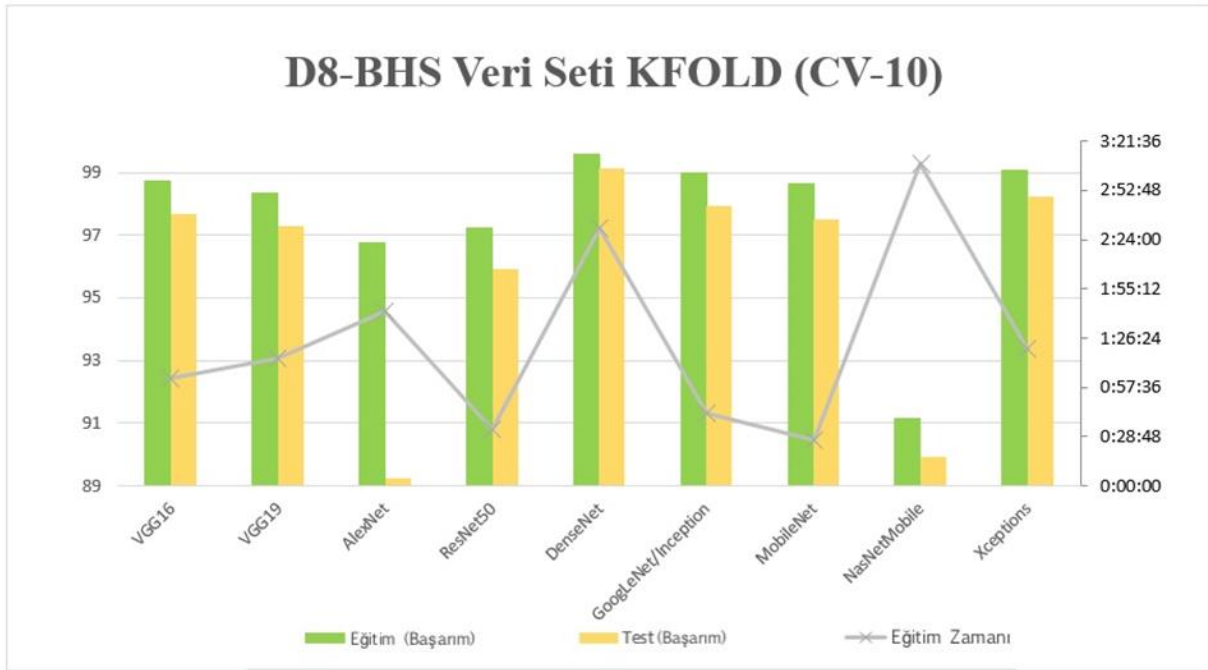
Şekil 3.32 Bitki Hastalıklarının Sınıflandırılması Problemi için D8-BHS Veri Seti Kullanılarak 3 Kat Çapraz Doğrulama Model Sonuçları

Şekil 3.32’de görüldüğü üzere, D8-BHS veri seti için, 3 kat çapraz doğrulama ile oluşturulan MobileNet modelinden, %96,91 test doğruluk başarıyla ve 18 dakika 8 saniye modelleme süresi ile diğer modellere göre daha iyi performans elde edilmiştir.



Şekil 3.33 Bitki Hastalıklarının Sınıflandırılması Problemi için D8-BHS Veri Seti Kullanılarak 5 Kat Çapraz Doğrulama Model Sonuçları

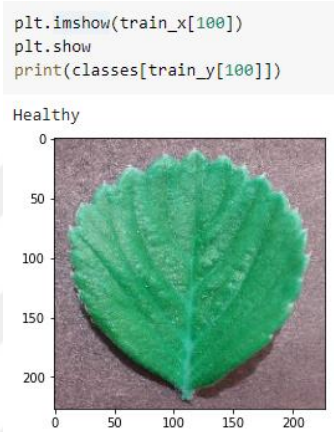
Şekil 3.33'te görüldüğü üzere, D8-BHS veri seti için, 5 kat çapraz doğrulama ile oluşturulan DenseNet modelinden, %98,5 test doğruluk başarısıyla en yüksek başarı elde edilmiştir. Bununla birlikte DenseNet modelinin oluşturulması için harcanan süre 1 saat 18 dakika 15 saniye olmuştur. Modelleme sürelerine bakıldığında, en kısa sürede oluşturulan model, 19 dakika 39 saniye ile MobileNet modeli olmuştur. MobileNet modelinin test doğruluk başarısı ise %97,01 olmuştur. Optimum model olarak, modelleme süresi önemsendiğinde, MobileNet modeli, sadece yüksek doğruluk başarısı önemli olduğu durumlarda ise DenseNet modeli seçilebilir.



Şekil 3.34 Bitki Hastalıklarının Sınıflandırılması Problemi için D8-BHS Veri Seti Kullanılarak 10 Kat Çapraz Doğrulama Model Sonuçları

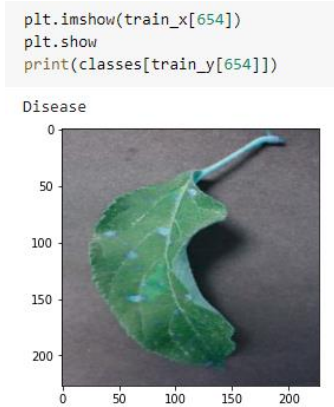
Şekil 3.34'te görüldüğü üzere, D8-BHS veri seti için, 10 kez çapraz doğrulama ile oluşturulan DenseNet modelinden, %99,14 test doğruluk başarısıyla en yüksek başarı elde edilmiştir. Bununla birlikte DenseNet modelinin oluşturulması için harcanan süre 2 saat 30 dakika 27 saniye olmuştur. Ancak, modelleme sürelerine bakıldığında, 5 kez çapraz doğrulama uygulanarak, en yüksek doğruluk başarısı (%99,14) ve en kısa modelleme süresi (23 dakika 1 saniye) ile GoogleNet modeli olarak bulunmuştur. Optimum model olarak, modelleme süresi önemsendiğinde, MobileNet modeli, sadece yüksek doğruluk başarısı önemli olduğu durumlarda ise DenseNet modeli seçilebilir.

Şekil 3.32, 3.33 ve 3.34 kıyaslandığında, bitki hastalıklarının sınıflandırılmasını yapabilen en iyi model, en yüksek test doğruluk başarısına göre 10 kat çapraz doğrulama uygulanarak, %98,5 doğruluk başarısı ile DenseNet modeli olarak bulunmuştur. Modelleme sürelerine bakıldığında, en kısa sürede oluşturulan model, 5 kat çapraz doğrulama uygulanarak, 19 dakika 39 saniye ile MobileNet modeli olmuştur. MobileNet modelinin test doğruluk başarısı ise %97,01 olmuştur. Optimum model olarak, modelleme süresi önemsendiğinde, MobileNet modeli, sadece yüksek doğruluk başarısı önemli olduğu durumlarda ise DenseNet modeli seçilebilir.

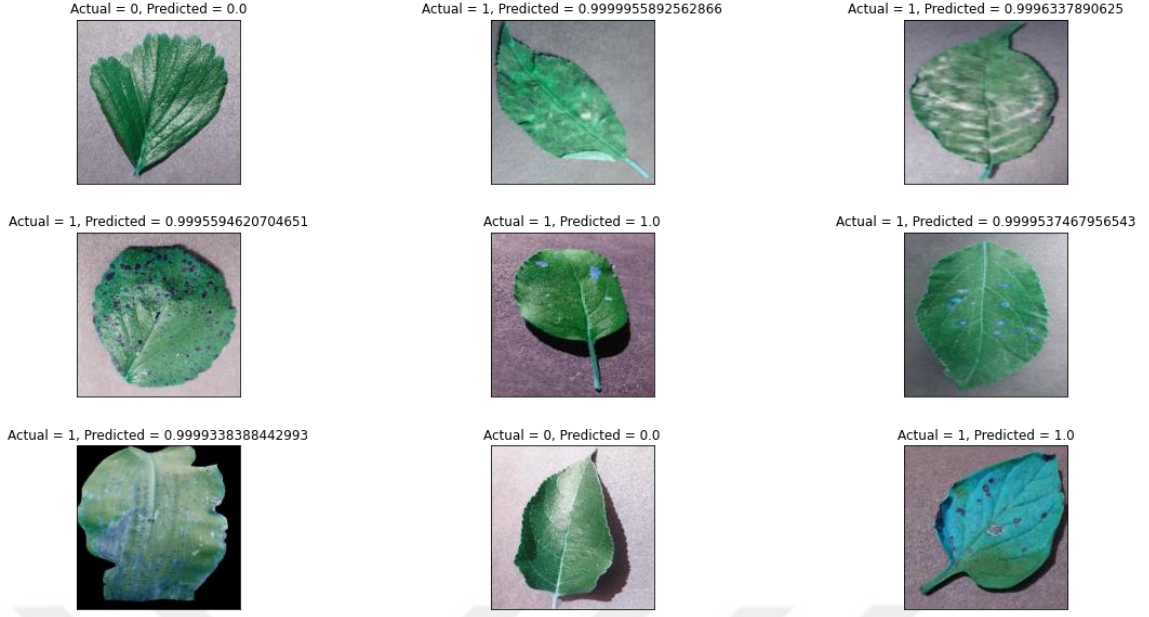


Şekil 3.35. Verilerin rastgele teyit işlemleri

Eğitilen CNN modellerinin yüksek başarımlar sağlamanın ardından veri setleri üzerinde bazı denemeler sağlanarak verilerin sıralı olup olmadıkları konusunda teyitler sağlanmıştır. Şekil 3.35’ te görüldüğü üzere veri setinin, eğitim kısmında yer alan 100 numaralı veri sağlıklı olarak görülmektedir. Şekil 4.36’ da ise veri setinin, eğitim kısmında yer alan 654 numaralı veri hastalıklı olarak görülmektedir.



Şekil 3.36. Verilerin rastgele teyit işlemleri



Şekil 3.37. CNN model tahmin sonuç analiz örneği

Tez çalışmasının kapsamında olan üç farklı problem için eğitilmiş olan konvolüsyonel modeller ele alındığında, problemlere çözüm olarak yüksek sonuçlar vermiş ve istenilen tahminleri çok yüksek oranda sağlamıştır. Şekil 3.37’de görüldüğü üzere, bir CNN modelinden alınmış olan model başarımlarını analiz örneğinde 9 farklı veriye ait gerçek değer ve modelin tahmin ettiği değerler gözlemlenmiş ve sonuçların çok yüksek doğrultuda doğru tahminlerde bulunduğu saptanmıştır.

4. SONUÇ VE ÖNERİLER

Bu çalışmada, derin öğrenme yaklaşımlarının kullanıldığı, görüntü sınıflandırmaya dayanan, başta bitki türleri olmak üzere, bitkilerin hastalıklı mı yoksa sağlıklı mı oldukları, eğer hastalıklı ise hangi tür hastalığa sahip olduklarını otomatik bir şekilde tespit etmeye çalışan, üç aşamalı bir çalışma sunulmuştur.

Popüler bir veri seti olan Plantvillage veri seti seçilerek, bu veri setinden çok sınıflı ve az sınıflı olmak üzere toplam 8 adet yeni veri seti oluşturulmuştur. Oluşturulan bu veri setleri kullanılarak, bitki türlerinin sınıflandırılması, bitki hastalık tahmini ve bitki hastalıklarının sınıflandırılması için, derin öğrenme algoritmalarına ait özel mimarilerden, AlexNet, VGG16, VGG19, GoogleNet, DenseNet, ResNet50, MobileNet, Xception ve NasNetMobile mimarileri uygulanarak, modelleme çalışmaları gerçekleştirilmiştir. Ve her bir sınıflandırma problemi için, geniş kapsamlı deneysel çalışmalar sonucunda en iyi modeller bulunmuştur.

Modelleme çalışmalarının ilk aşamasında, bitki türlerinin sınıflandırılması problemine yönelik hazırlanmış olan 6 farklı dengeli veri seti kullanılarak, Alexnet mimarisi uygulanmıştır ve en iyi model %99,88 test doğruluk başarısı ile elde edilmiştir.

Modellemenin ikinci aşamasında, bitki hastalık tahmini problemine yönelik hazırlanmış olan, 2 sınıftan oluşan dengeli veri seti kullanılarak, literatür sonuçlarında gözlemlenen CNN'in en popüler 9 farklı derin öğrenme mimarisi uygulanmıştır. Bunun sonucunda, bitki hastalık tahmini için en iyi model, GoogleNet/Inception mimarisinin kullanıldığı model olmuştur ve 3 kez çapraz doğrulama ile % 99,91 doğruluk başarısı elde edilmiştir.

Son olarak, bitki hastalıklarının sınıflandırılması problemine yönelik, kullanılan veri seti ile DenseNet modeli en iyi model olarak bulunmuştur ve 10 kez çapraz doğrulama ile % 99,14 doğruluk başarısı elde edilmiştir.

Bu tez çalışmasının sonucunda, birçok farklı CNN modeli çok farklı sınıf ve veri özelliklerine sahip veri setleri üzerinde modellenmiş olup, geniş kapsamda farklılıkları karşılaştırarak inceleme fırsatı sunmuştur. CNN modellerinin veri setinin daha az sınıftan oluştuğu durumlarda (D6-BTS ve D7-BHT) çok daha yüksek sonuçlar sergiledikleri gözlemlenmiştir. Bir diğer saptanan durum ise çapraz doğrulama yönteminin özellikle çok fazla sınıfa sahip olunan veri setlerinde 3 kat çapraz doğrulama yerine 10 kat çapraz doğrulama kullanılarak model başarımının daha yüksek seviyelere taşıma konusunda destek sağladığı

olmuştur. Elde edilen sonuçlar ışığında, derin öğrenme için veri miktarının çokluğunun hem modelin öğrenme başarısını hem de test başarısını arttırdığı ve özellikle görüntü verileri üzerinde sınıflandırma problemlerinde çok başarılı sonuçlar vererek kullanılabileceğini göstermiştir.



KAYNAKLAR

- Abbas, A., Jain, S., Gour, M., & Vankudothu, S. (2021). Tomato plant disease detection using transfer learning with C-GAN synthetic images. *Computers and Electronics in Agriculture*, 187. <https://doi.org/10.1016/j.compag.2021.106279>
- Ando, R. K. (2005). *A Framework for Learning Predictive Structures from Multiple Tasks and Unlabeled Data*. 6, 1817–1853.
- Atila, Ü., Uçar, M., Akyol, K., & Uçar, E. (2021). Ecological Informatics Plant leaf disease classification using EfficientNet deep learning model. *Ecological Informatics*, 61(September 2020), 101182. <https://doi.org/10.1016/j.ecoinf.2020.101182>
- Ayan, E., Erbay, H., & Varçın, F. (2020). Crop pest classification with a genetic algorithm-based weighted ensemble of deep convolutional neural networks. *Computers and Electronics in Agriculture*, 179(September), 105809. <https://doi.org/10.1016/j.compag.2020.105809>
- Barbedo, J. G. A. (2018). Impact of dataset size and variety on the effectiveness of deep learning and transfer learning for plant disease classification. *Computers and Electronics in Agriculture*. <https://doi.org/10.1016/j.compag.2018.08.013>
- Bedi, P., & Gole, P. (2021). Plant disease detection using hybrid model based on convolutional autoencoder and convolutional neural network. *Artificial Intelligence in Agriculture*, 5, 90–101. <https://doi.org/10.1016/j.aiia.2021.05.002>
- Brodrick, P. G., Davies, A. B., & Asner, G. P. (2019). Uncovering Ecological Patterns with Convolutional Neural Networks. *Trends in Ecology & Evolution*, 34(8), 734–745. <https://doi.org/10.1016/j.tree.2019.03.006>
- Cogan, T., Cogan, M., & Tamil, L. (2019). MAPGI : Accurate identification of anatomical landmarks and diseased tissue in gastrointestinal tract using deep learning. *Computers in Biology and Medicine*, 111(July), 103351. <https://doi.org/10.1016/j.combiomed.2019.103351>
- Darwish, A., Ezzat, D., & Hassanien, A. E. (2020). An optimized model based on convolutional neural networks and orthogonal learning particle swarm optimization algorithm for plant diseases diagnosis. *Swarm and Evolutionary Computation*. <https://doi.org/10.1016/j.swevo.2019.100616>
- DOĞAN, F., & TÜRKOĞLU, İ. (2019). Derin Öğrenme Modelleri ve Uygulama Alanlarına İlişkin Bir Derleme. *DÜMF Mühendislik Dergisi*. <https://doi.org/10.24012/dumf.411130>
- Dyrmann, M., Karstoft, H., & Midtiby, H. S. (2016). Plant species classification using deep convolutional neural network. *Biosystems Engineering*. <https://doi.org/10.1016/j.biosystemseng.2016.08.024>
- Fatih, M., Fahri, M., Sabanci, K., & Durdu, A. (2021). CNN-based transfer learning – BiLSTM network : A novel approach for COVID-19 infection detection. *Applied Soft Computing Journal*, 98, 106912. <https://doi.org/10.1016/j.asoc.2020.106912>
- Feng, H., Liu, P., Mu, N., & Wang, J. (2018). The high dynamic reconstruction of high-

- resolution image based on MapReduce. *Proceedings of 2017 IEEE International Conference on Unmanned Systems, ICUS 2017*.
<https://doi.org/10.1109/ICUS.2017.8278388>
- Ferentinos, K. P. (2018). Deep learning models for plant disease detection and diagnosis. Deep learning models for plant disease detection and diagnosis. *Computers and Electronics in Agriculture*, 145(February), 311–318.
<https://doi.org/10.1016/j.compag.2018.01.009>
- Gatys, L. A., Ecker, A. S., & Bethge, M. (2015). Texture synthesis using convolutional neural networks. *Advances in Neural Information Processing Systems*.
- Gu, J., Wang, Z., Kuen, J., Ma, L., Shahroudy, A., & Shuai, B. (2006). *Recent Advances in Convolutional Neural Networks*. 1–38.
- Han, D., Liu, Q., & Fan, W. (2018). A new image classification method using CNN transfer learning and web data augmentation. *Expert Systems With Applications*, 95, 43–56.
<https://doi.org/10.1016/j.eswa.2017.11.028>
- Han, S., Mao, H., & Dally, W. J. (2016). Deep compression: Compressing deep neural networks with pruning, trained quantization and Huffman coding. *4th International Conference on Learning Representations, ICLR 2016 - Conference Track Proceedings*.
- He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. <https://doi.org/10.1109/CVPR.2016.90>
- Hernández, S., & López, J. L. (2020). Uncertainty quantification for plant disease detection using Bayesian deep learning. *Applied Soft Computing Journal*, 96, 106597.
<https://doi.org/10.1016/j.asoc.2020.106597>
- Hinton, G. E., & Salakhutdinov, R. R. (2006). Reducing the dimensionality of data with neural networks. *Science*. <https://doi.org/10.1126/science.1127647>
- Hinton, Geoffrey E., Osindero, S., & Teh, Y. W. (2006). A fast learning algorithm for deep belief nets. *Neural Computation*. <https://doi.org/10.1162/neco.2006.18.7.1527>
- Hossin, M. (2016). *A REVIEW ON EVALUATION METRICS FOR*. March 2015.
<https://doi.org/10.5121/ijdkp.2015.5201>
- Howard, A. G., & Wang, W. (2012). *Applications*.
- Hughes, D. P., & Salathe, M. (2015). *An open access repository of images on plant health to enable the development of mobile disease diagnostics*.
- Irmak, G., & Saygili, A. (2020). *Evrışimli Sinir Ağları ile Domates Yaprağı Hastalık Tespiti ve Sınıflandırması Tomato Leaf Disease Detection and Classification using Convolutional Neural Networks*.
- Jeon, W. S., & Rhee, S. Y. (2017). Plant leaf recognition using a convolution neural network. *International Journal of Fuzzy Logic and Intelligent Systems*.
<https://doi.org/10.5391/IJFIS.2017.17.1.26>
- Jia, Y., Shelhamer, E., Donahue, J., Karayev, S., Long, J., Girshick, R., Guadarrama, S., Darrell,

- T., & Eecs, U. C. B. (2014). *Caffe : Convolutional Architecture for Fast Feature Embedding * Categories and Subject Descriptors*.
- Kalaycı, T. E. (2018). Comparison of Machine Learning Techniques for Classification of Phishing Web Sites. *Pamukkale University Journal of Engineering Sciences*, 24(5), 870–878. <https://doi.org/10.5505/pajes.2018.10846>
- Kaya, A., Keceli, A. S., Catal, C., Yalic, H. Y., Temucin, H., & Tekinerdogan, B. (2019). Analysis of transfer learning for deep neural network based plant classification models. *Computers and Electronics in Agriculture*. <https://doi.org/10.1016/j.compag.2019.01.041>
- Krizhevsky, B. A., Sutskever, I., & Hinton, G. E. (2012). *ImageNet Classification with Deep Convolutional Neural Networks*.
- Kumari, P., & Seeja, K. R. (2020). ScienceDirect ScienceDirect Periocular Biometrics for non-ideal images : with off-the-shelf Deep CNN & Transfer Learning approach Periocular Biometrics for images : with off-the-shelf Deep CNN & Transfer Learning approach. *Procedia Computer Science*, 167(2019), 344–352. <https://doi.org/10.1016/j.procs.2020.03.234>
- Labhsetwar, S. R., Mumbai, N., Deshpande, R., Mumbai, N., Haridas, S., Mumbai, N., Kolte, P. A., Mumbai, N., Panmand, R., Mumbai, N., Pati, S., & Mumbai, N. (n.d.). *Performance Analysis of Optimizers for Plant Disease Classification with Convolutional Neural Networks*.
- LeCun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*. <https://doi.org/10.1109/5.726791>
- Lee, S. H., Chan, C. S., Mayo, S. J., & Remagnino, P. (2017). How deep learning extracts and learns leaf features for plant classification. *Pattern Recognition*. <https://doi.org/10.1016/j.patcog.2017.05.015>
- Lee, S. H., Goëau, H., Bonnet, P., & Joly, A. (2020). New perspectives on plant disease characterization based on deep learning. *Computers and Electronics in Agriculture*. <https://doi.org/10.1016/j.compag.2020.105220>
- Liu, Q., Xiang, X., Qin, J., Tan, Y., Tan, J., & Luo, Y. (2020). Knowledge-Based Systems Coverless steganography based on image retrieval of DenseNet features and DWT sequence mapping ☆. *Knowledge-Based Systems*, 192, 105375. <https://doi.org/10.1016/j.knosys.2019.105375>
- Mahadeokar, J., & Pesavento, G. (2016). *Open Sourcing a Deep Learning Solution for Detecting NSFW Images*. Yahoo Engineering.
- Mohanty, S. P., Hughes, D. P., & Salathé, M. (2016). Using deep learning for image-based plant disease detection. *Frontiers in Plant Science*. <https://doi.org/10.3389/fpls.2016.01419>
- Oliphant, T. E. (2014). *Guide to NumPy. January 2006*.
- Picon, A., Seitz, M., Alvarez-Gila, A., Mohnke, P., Ortiz-Barredo, A., & Echazarra, J. (2019). Crop conditional Convolutional Neural Networks for massive multi-crop plant disease

- classification over cell phone acquired images taken on real field conditions. *Computers and Electronics in Agriculture*. <https://doi.org/10.1016/j.compag.2019.105093>
- Rangarajan, A. K., & Purushothaman, R. (2018). ScienceDirect ScienceDirect Tomato crop disease classification using pre-trained deep learning algorithm. *Procedia Computer Science*, 133, 1040–1047. <https://doi.org/10.1016/j.procs.2018.07.070>
- Ren, S., He, K., Girshick, R., & Sun, J. (2015). Faster R-CNN: Towards real-time object detection with region proposal networks. *Advances in Neural Information Processing Systems*.
- Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A. C., & Fei-Fei, L. (2015). ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision*. <https://doi.org/10.1007/s11263-015-0816-y>
- Saleem, M. H., & Potgieter, J. (2020). *Plant Disease Classification : A Comparative Evaluation of Convolutional Neural Networks and Deep Learning Optimizers*.
- Salur, M. U., & Aydin, I. (2018). Sentiment classification based on deep learning. *26th IEEE Signal Processing and Communications Applications Conference, SIU 2018*. <https://doi.org/10.1109/SIU.2018.8404471>
- Sharma, P., Berwal, Y. P. S., & Ghai, W. (2019). Performance analysis of deep learning CNN models for disease detection in plants using image segmentation. *Information Processing in Agriculture*. <https://doi.org/10.1016/j.inpa.2019.11.001>
- Simonyan, K., & Zisserman, A. (2015). Very deep convolutional networks for large-scale image recognition. *3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings*.
- Sparse Transfer Learning for Interactive Video Search Reranking*. (2012). 8(3), 1–19. <https://doi.org/10.1145/2240136.2240139>
- Subetha, T., Khilar, R., & Subaja, M. (2021). Materials Today : Proceedings A comparative analysis on plant pathology classification using deep learning architecture – Resnet and VGG19. *Materials Today: Proceedings*, xxxx. <https://doi.org/10.1016/j.matpr.2020.11.993>
- Thiruvathukal, E. G. K., & Hunter, B. J. D. (2007). *MATPLOTLIB : A 2D GRAPHICS ENVIRONMENT*. 90–95.
- Thrun, S. (n.d.). *Is Learning The -th Thing Any Easier Than Learning The First ?*
- Türkoğlu, M., & Hanbay, D. (2019). Apricot Disease Identification based on Attributes Obtained from Deep Learning Algorithms. *2018 International Conference on Artificial Intelligence and Data Processing, IDAP 2018*. <https://doi.org/10.1109/IDAP.2018.8620831>
- TÜRKOĞLU, M., HANBAY, K., SARAÇ SİVRİKAYA, I., & HANBAY, D. (2020). Derin Evrimsel Sinir Ağı Kullanılarak Kayısı Hastalıklarının Sınıflandırılması. *Bitlis Eren Üniversitesi Fen Bilimleri Dergisi*. <https://doi.org/10.17798/bitlisfen.562101>
- Xue, G., Jiang, Y., Yang, Q., & Yu, Y. (2008). *Can Chinese web pages be classified with English*

- data source ? Can Chinese Web Pages be Classified with English Data Source ? January.*
<https://doi.org/10.1145/1367497.1367628>
- Xue, W., Hu, X., Wei, Z., Mei, X., Chen, X., & Xu, Y. (2019). A fast and easy method for predicting agricultural waste compost maturity by image-based deep learning. *Bioresource Technology*. <https://doi.org/10.1016/j.biortech.2019.121761>
- Yang, Q., & Chen, Y. (2009). *Heterogeneous Transfer Learning for Image Clustering via the Social Web*.
- Yıldız, O. (2019). Derin öğrenme yöntemleriyle dermoskopi görüntülerinden melanom tespiti: Kapsamlı bir çalışma. *Gazi Üniversitesi Mühendislik-Mimarlık Fakültesi Dergisi*. <https://doi.org/10.17341/gazimmfd.435217>
- Yosinski, J., Clune, J., Bengio, Y., & Lipson, H. (n.d.). *How transferable are features in deep neural networks?*
- You, Y., Zhang, Z., Demmel, J., Keutzer, K., & Hsieh, C.-J. (2017). 100-epoch ImageNet Training with AlexNet in 24 Minutes. *Arxiv*.
- Zhang, B., Xiao, T., & Zhong, J. (2019). A simple determination approach for zero-padding of FFT method in focal spot calculation. *Optics Communications*, 451(July), 260–264. <https://doi.org/10.1016/j.optcom.2019.06.065>
- Zhang, Y., Zhao, Z., Deng, Y., Zhang, X., & Zhang, Y. (2021). Biomedical Signal Processing and Control Human identification driven by deep CNN and transfer learning based on multiview feature representations of ECG. *Biomedical Signal Processing and Control*, 68(December 2019), 102689. <https://doi.org/10.1016/j.bspc.2021.102689>
- Zhao, B., Lu, H., Chen, S., Liu, J., & Wu, D. (2017). *Convolutional neural networks for time series classification*. 28(1), 162–169. <https://doi.org/10.21629/JSEE.2017.01.18>
- Bisong, E. (2019). Google Colaboratory. In *Building Machine Learning and Deep Learning Models on Google Cloud Platform* (s. 59-64).
- Chollet, F. (2019). *Python ile Derin Öğrenme*. Ankara: Buzdağı Yayınevi.
- Kairoidayeva, A., (2019). Automatic determination of plan type and phenological stage with deep learning methods. *Istanbul Technical University, informatics institu*