



T.C.
SELÇUK ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

KONVOLÜSYONEL SİNİR AĞLARINI KULLANARAK ÜZÜM VE ÜZÜM YAPRAKLARININ HASTALIKLARININ TESPİT EDİLMESİ

Shekofa GHOURY

YÜKSEK LİSANS TEZİ

Bilgisayar Mühendisliği Anabilim Dalı

Ağustos-2019
KONYA
Her Hakkı Saklıdır

ÖZET

YÜKSEK LİSANS TEZİ

KONVOLÜSYONEL SİNİR AĞLARINI KULLANARAK ÜZÜM VE ÜZÜM YAPRAKLARININ HASTALIKLARININ TESPİT EDİLMESİ

Shekofa GHOURY

Selçuk Üniversitesi Fen Bilimleri Enstitüsü
Bilgisayar Mühendisliği Anabilim Dalı

Danışman: Prof. Dr. Cemil SUNGUR

2019, 74 Sayfa

Jüri
Prof. Dr. Cemil SUNGUR
Doç. Dr. İSMAİL BABAOĞLU
Dr.Öğr.Üyesi Mehmet HACIBEYOĞLU

Bağcılık, insanoğlunun en eski tarımsal faaliyetlerinden biri olup, ve halen de önemini kayıp etmeden ve iklimi uygun ülkelerde en yaygın tarımsal üretim alanlarından biridir. Bitki hastalıkları tarım sektöründe ürün kaybına ve kalitesinin düşmesine ve dolayısıyla büyük ekonomik kayıplara sebep olmaktadır. Bitkilerde hastalıkların ve zararlıların doğru ve hızlı şekilde tespit edilmesi, ekonomik kayıpları büyük ölçüde azaltırken, erken bir tedavi tekniğinin geliştirilmesine yardımcı olabilir. Asmalardaki hastalıkların manuel olarak (elle) takip etmek ve test etmek çok zor ve zaman alıcıdır. Bazı üzüm hastalıklarında kapsamlı işlem süresi, muazzam miktarda prosedür ve uzmanlık gerektirir. Asmalardaki hastalıkları tespit etmek için hızlı otomatik bir işleme sistemi gerekmektedir.

Bu çalışmada, asmada (Vitis vinifera) hastalıklarını tespit etme çalışması gerçekleştirilmiştir. Amaç üzüm meyvesi ve yaprağı üzerinde oluşacak hastalıklarının sınıflandırılmasıdır. Bu tez çalışması, veri büyütme (data agumentation) tekniklerini, önceden eğitilmiş olan iki derin öğrenme modelini (SSD_MobileNet v1 ve Faster R-CNN Inception v2) ve bu çalışma için özel olarak farklı kaynaklar kullanarak hazırlanmış olan iki veri setini kullanılarak yeniden eğitilmesi gerçekleştirilmiştir.

Birinci veri seti için, Faster R-CNN Inception v2 modeli %78 ile %99 aralığında sınıflandırma doğruluğu ile bu veri setinin tüm test görüntülerinin %95,57'sini doğru şekilde sınıflandırılabilmiştir. Bu modelin doğruluk seviyesi daha yüksek çıkmıştır ancak işlem süresi daha uzun sürmüştür. SSD_MobileNet v1, daha iyi işleme süresiyle tüm test görüntülerinin yalnızca %59,29'unu doğru bir şekilde sınıflandırılabilmiştir. İkinci veri seti için, Faster-R-CNN Inception v2 modeli, bu veri setinin tüm test görüntülerinin %91,61'sini %70 – 99 aralığında doğru şekilde sınıflandırılabilmiştir. Bu modelin doğruluk seviyesi daha yüksek olmuştur ancak daha uzun işlem süresine sahiptir. SSD_MobileNet v1, tüm test görüntülerinin sadece %18'ini doğru ve çok daha iyi işlem süresiyle sınıflandırılabilmiştir.

Elde edilen sonuçlara göre, SSD_MobileNet v1 modelinin üzüm meyvesi ve yaprağı hastalıklarının sınıflandırmanın işlemi için yararlı olamayacağı ve başarısız olduğu sonucuna varılmıştır. Diğer tarafından, Faster-R-CNN Inception v2 modeli oldukça iyi bir sonuçlar sergilemiş ve daha başarılı bir model olduğu sonucuna varılmıştır.

Anahtar Kelimeler: Derin Öğrenme, Faster R-CNN, Inception modeli, Konvolüsyonel Sinir Ağları, MobileNet, Tek Çekim Çok Kutulu Dedektör (SSD), Üzüm Hastalığı Tespiti.

ABSTRACT

MS THESIS

GRAPE AND GRAPE LEAVES DISEASES DETECTION USING CONVOLUTIONAL NEURAL NETWORKS

Shekofa GHOURY

THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCE OF SELÇUK UNIVERSITY
THE DEGREE OF MASTER OF SCIENCE IN COMPUTER ENGINEERING

Advisor: Prof. Dr. Cemil SUNGUR

2019, 74 Pages

Jury
Prof. Dr. Cemil SUNGUR
Doç. Dr. İSMAİL BABAOĞLU
Dr.Öğr.Üyesi Mehmet HACIBEYOĞLU

Viticulture is one of the oldest agricultural activities of mankind and it is still one of the most widespread agricultural production areas without losing its importance in the countries which has suitable climate for it. Plant diseases cause loss of crops and decrease quality of agricultural sector and consequently cause major economic losses. Accurate and rapid detection of diseases and pests in plants can help to develop an early treatment technique while greatly reducing economic losses. It is very difficult and time-consuming to follow and test the grape disease manually and requires extensive processing time, a tremendous amount of procedures and expertise in some grape diseases. A fast automated processing system is needed to detect grape disease.

In this thisis work, the task of grape and grape leaf (Vitis vinifera) disease detection is performed. The goal is to classify grapes and grape leaves diseases. This work is done using image agumanetation techniques and transfer learning by retraining two pre-trained deep learning models, SSD_MobileNet v1 and Faster R-CNN Inception v2 on two datasets (named first dataset and second dataset in this work) which had been prepared using different methods specially for this thesis work.

For the first dataset, Faster-R-CNN Inception v2 model classified 95.57% of all the testing images of this dataset correctly with a classification accuracy of 78% to 99%. The accuracy level of this model was higher but with longer processing time. SSD_MobileNet v1 only classified 59.29% of all the testing images correctly but with better processing time. For the second dataset, Faster-R-CNN Inception v2 model was able to classify 91.61% of all the testing images of this dataset correctly with a classification accuracy of 70% to 99%. The accuracy level of this model was higher but with longer processing time. SSD_MobileNet v1 only classified 18% of all the testing images correctly but with better processing time

Based on these results, it was concluded that the SSD_MobileNet v1 model could not be useful for the task of grape and grape leaves diseases classifications and was unsucessful model for this task. On the other hand, the Faster-R-CNN Inception v2 model showed very good results and was concluded to be a more successful model for this task.

Keywords: Convolutional Neural Networks, Deep Learning, Faster - RCNN, Grape Disease Detection, Inception model, MobileNet, Single Shot Multibox Detector (SSD).

ÖNSÖZ

Bu çalışma kapsamında aktarımlı öğrenme yöntem ile önceden eğitilmiş olan iki farklı konvolüsyonel sinir ağlarını ve bu çalışma için özel olarak hazırlanmış olan iki veri setlerini kullanılarak asmada (Vitis vinifera) önemli kayıplara neden olan hastalıkların tespit edilmesi gerçekleştirilmiştir.

Shekofa GHOURY
KONYA-2019

TEŞEKKÜRLER

Bilgisayar Mühendisliğinin lisansüstü eğitiminin kademelerden biri olan yüksek lisans eğitimimin sonuna gelmiş bulunuyorum. Yüksek lisans süreci boyunca yardımlarını esirgemeyen danışmanım Prof. Dr. Cemil SUNGUR'a rehberlik etmesi, konsültasyonları için harcanan zaman, düzeltme ve bu çalışmanın başlamasına ilham vermesi için teşekkür etmek istiyorum.

Bu süreçte çalışmama ilgi gösteren, çalışmama yardımcı olan fikirleri ve zaman harcamaları için Dr. Akif DURDU'a da sonsuz teşekkürlerimi sunarım.

Ayrıca tezimi yazarken yaşadığım dil sıkıntılarını gidermede yardımcı olan Almıla ÖZCAN'a da teşekkürlerimi sunarım.

Tüm eğitim ve tez çalışmalarım boyunca sabır, anlayış ve sevgileri ile her zaman yanımda olan, maddi-manevi desteklerini hiçbir zaman esirgemeyen ve aldığım kararlarda her zaman yanımda olan sevgili aileme tüm kalbimle teşekkür ederim.

İÇİNDEKİLER

ÖZET ..... **iv**

ABSTRACT ..... **v**

ÖNSÖZ ..... **vi**

TEŞEKKÜRLER ..... **vii**

İÇİNDEKİLER ..... **viii**

ŞEKİLLER DİZİNİ ..... **x**

SİMGELER VE KISALTMALAR ..... **xii**

1. GİRİŞ **1**

2. ÜZÜM MEYVESİ VE YAPRAKLARI HASTALIKLARI **3**

2.1. Tüylü Küf (Downy Mildew) 3
2.2. Külleme (Powdery Mildew) 4
2.3. Siyah Çürük (Black Rot) 4
2.4. Botrytis Çürüklük (Botrytis Bunch Rot) 5
2.5. Blister Akarı (Blister Mite) 6
2.6. Esca 7
2.7. Yaprak Yanıklığı (Leaf blight) 7

3. İLGİLİ ÇALIŞMALAR **9**

4. DERİN ÖĞRENME **18**

4.1. Yapay Zeka 18
4.2. Makine Öğrenmesi 19
4.3. Derin Öğrenme 20
4.3.1. Makine öğrenmesi ile derin öğrenme aralarındaki fark 20
4.3.2. Makine öğrenmesi ve derin öğrenme arasında seçim 21
4.3.3. Derin öğrenme modellerinin oluşturulması ve eğitilmesi 22
4.3.3.1. Sıfırdan eğitim (Training from scratch) 22
4.3.3.2. Aktarımlı öğrenme (Transfer learning) 22
4.3.4. Derin sinir ağının oluşturulması 23
4.3.4.1. Biyolojik nöron 23
4.3.4.2. Tek katmanlı perseptron 24
4.3.4.3. Yapay sinir ağı 25
4.3.4.4. Derin sinir ağları 26

5. KONVOLÜSYONEL SİNİR AĞLARI **28**

5.1. Konvolüsyonel Sinir Ağlarının Çalışması 28
5.2. Konvolüsyonel Sinir Ağların Mimarisi 28
5.2.1. Özellik öğrenme 29
5.2.1.1. Konvolüsyon (Convolution) katmanı 29

5.2.1.2. Doğrultulmuş doğrusal birim (ReLU) 31
5.2.1.3. Havuzlama (Pooling) katmanı 31
5.2.1.4. Tam bağlı (Fully connected) katmanı 32
5.2.2. Sınıflandırma katmanlar 32
5.3. Derin KSA Öğrenmesi 32

6. MATERYAL VE YÖNTEM 34

6.1. Sistem Konfigürasyonları 34
6.2. SSD MobileNet 35
6.2.1. MobileNet 35
6.2.2. SSD (single shot multibox detector) 36
6.3. Faster-RCNN Inception V2 37
6.3.1. Faster–RCNN 37
6.3.2. Inception v2 38
6.4. Veri Büyütme (Data Augmentation) 39
6.5. Veri Setleri 39
6.5.1. Birinci veri seti 40
6.5.2. İkinci veri seti 40

7. DENEYSEL SONUÇLAR VE TARTIŞMALAR 43

7.1. Modellerin Eğitilmesi 43
7.1.1. Birinci veri setinin üzerinde modellerin eğitilmesi 43
7.1.1.1. Faster-R-CNN Inception v2 modelin eğitilmesi 43
7.1.1.2. SSD MobileNet v1 modelin eğitilmesi 44
7.1.2. İkinci veri setinin üzerinde modellerin eğitilmesi 45
7.1.2.1. Faster-R-CNN Inception v2 modelin eğitilmesi 45
7.1.2.2. SSD MobileNet v1 modelinin eğitilmesi 45
7.2. Modellerin Test Edilmesi 46
7.2.1. Birinci veri seti için modellerin test sonuçları 46
7.2.1.1. Faster-RCNN Inception v2 modelin test sonuçları 46
7.2.1.2. SSD_MobileNet v1 modelin test sonuçları 47
7.2.2. İkinci veri seti için modellerin test sonuçları 48
7.2.2.1. Faster-RCNN Inception v2 modelin test sonuçları 49
7.2.2.2. SSD_MobileNet v1 modelin test sonuçları 50
7.3. Tartışmalar 52
7.3.1. Birinci veri setinin sonuçları için tartışma 52
7.3.2. İkinci veri setinin sonuçları için tartışma 53

8. SONUÇLAR VE ÖNERİLER 54

8.1. Sonuçlar 54
8.2. Öneriler 55

KAYNAKLAR 56

ÖZGEÇMİŞ 61

ŞEKİLLER DİZİNİ

Şekil 2.1. (a, b) Tüylü Küf Üzüm yaprakların üzerinde (Westover, 2013), (c) Tüylü Küf Üzüm meyvelerin üzerinde (Western Australian Agriculture Authority, 2019) 3

Şekil 2.2. (a) Külleme Üzüm yaprağı üzerinde (Homegrown, 2013), (b) Külleme Üzüm meyveleri üzerinde (Nelson, 2015) 4

Şekil 2.3. (a) Üzüm meyvesinde Siyah Çürük (Demchak, 2017), (b) Üzüm Yaprağında Siyah Çürük (Schilder, 2011) 5

Şekil 2.4. Botrytis Çürüklük (Ellis, 2008a) 5

Şekil 2.5. Blister Akarı (The Royal Horticultural Society, 2019) 6

Şekil 2.6. Esca hastalığı (Mugnai ve ark., 1999) 7

Şekil 2.7. Yaprak yanıklığı hastalığı (Grabowski, 2018) 8

Şekil 4.1. Derin öğrenme, Makine öğrenme ve Yapay zeka (Chollet, 2017) 18

Şekil 4.2. Makine öğrenmesi: yeni bir programlama paradigması (Chollet, 2017) 19

Şekil 4.3. Araçları kategorilere ayırmak için Derin öğrenme yaklaşımının Makine öğrenmesi yaklaşımı ile karşılaştırılması (Karunakaran, 2018a) 21

Şekil 4.4. Derin öğrenme modellerinin eğitilme yaklaşımları (Willingham, 2017) 22

Şekil 4.5. Biyolojik nöron (Zhang ve ark., 2018) 23

Şekil 4.6. Perseptron (Sharma, 2017) 24

Şekil 4.7. Tek Katmanlı Yapay Sinir Ağı (Amini, 2019) 25

Şekil 4.8. Derin Sinir Ağı (Amini, 2019) 26

Şekil 4.9 Mevcut olan bir kaç Yapay Sinir Ağların Mimarisi (Veen, 2016) 27

Şekil 5.1 Birçok konvolüsyon katmanları olan KSA örneği (MathWorks, 2018a) 29

Şekil 5.2. Konvolüsyon İşlemi (Goodfellow ve ark., 2016) 30

Şekil 5.3. Havuzlama İşlemi (Mishra, 2019) 31

Şekil 5.4. Derin KSA'nın verileri öğrenmesi (Amini, 2019) 33

Şekil 6.1. MobileNet'in Genel Mimarisi 36

Şekil 6.2. Genel Nesne Algılama İşlem için SSD'nin Üst Düzey Diyagramı (L. Liu ve ark., 2018) 37

Şekil 6.3. Genel Nesne Algılama İşlemi için Faster R-CNN'nin Üst Düzey Diyagramı (L. Liu ve ark., 2018) 38

Şekil 7.1. Birinci veri seti için Faster-RCNN Inception v2 modelin Toplam Kayıp değerlerinin grafiği 44

Şekil 7.2. Birinci veri seti için SSD_MobileNet v1 modelin Toplam Kayıp değerlerinin grafiği 44

Şekil 7.3. İkinci veri seti için Faster-RCNN Inception v2 modelin Toplam Kayıp değerlerinin grafiği 45

Şekil 7.4. İkinci veri seti için SSD_MobileNet v1 modelin Toplam Kayıp değerlerinin grafiği 46

Şekil 7.5. Birinci veri seti için Faster-RCNN Inception v2 modelin test sonuçlar grafiği 47

Şekil 7.6. Birinci veri seti için Faster-RCNN Inception v2 modelin test doğruluk öreği 47

Şekil 7.7. Birinci veri seti için SSD_MobileNet v1 modelin test sonuçlar grafiği..... 48

Şekil 7.8. Birinci veri seti için SSD_MobileNet v1 modelin test doğruluk örneği 48

Şekil 7.9. İkinci veri seti için Faster-RCNN Inception v2 modelin test sonuçlar grafiği 49

Şekil 7.10. İkinci veri seti için Faster-RCNN Inception v2 modelin test doğruluk öreği 50

Şekil 7.11. İkinci veri seti için SSD_MobileNet v1 modelin test sonuçlar grafiği 51

Şekil 7.12. İkinci veri seti için SSD_MobileNet v1 modelin test doğruluk öreği 51

SİMGELER VE KISALTMALAR

Simgeler

$\boldsymbol{b}$	Bias	Bias
$\boldsymbol{w}$	Ağırlık	Weight
$\boldsymbol{g}$	Aktivasyon fonksiyonu	Activation function
$\boldsymbol{x}$	Girdiler	Inputs
$\boldsymbol{z}$	Perseptronun Çıktısı	Output of Perceptron
$\hat{\boldsymbol{z}}$	Perseptronun Nihai Çıktısı	Final Output of Perceptron
$\boldsymbol{y}$	YSA'nın Çıktılar	Outputs of ANN
$\hat{\boldsymbol{y}}$	YSA'nın Nihai Çıktısı	Final Outputs of ANN
%	Yüzde	Percentage

Kısaltmalar

KSA	Konvolüsyonel Sinir Ağılar	Convolutional Neural Network
BİT	Bilgi ve İletişim Teknolojileri	Information and Communication Technologies
YSA	Yapay Sinir Ağılar	Artificial Neural Network
GPU	Grafik İşleme Birimi	Graphic Processing Unit
SVM	Destek Vektör Makinesi	Support Vector Machine
ReLU	Doğrultulmuş Doğrusal Birimler	Rectified Linear Units
VGG	Görsel Geometri Grubu	Visual Geometry Group
RGB	Kırmızı Yeşil Mavi	Red Green Blue
ResNet	Artık Ağı	Residual Network
SMO	Sıralı Minimal Optimizasyon	Sequential Minimal Optimization
DR	Doğrusal Regresyon	Linear Regression

RO	Rastgele Orman	Random Forest
MLP	Çok Katmanlı Perceptron	Multilayer Perceptron
RNN	Özyinelemeli Sinir Ağı	Recurrent Neural Network
LSTM	Uzun Kısa Süreli Bellek	Long Short-Term Memory
GRU	Kapılı Özyinelemeli Birim	Gated Recurrent Unit
İHA	İnsansız Hava Aracı	Unmanned Aerial Vehicle
SSD	Tek Atış Çok Kutulu Dedektörü	Single Shot Multibox Detector
R-FCN	Bölge Tabanlı Tamamen Konvolüsyonel Ağı	Region-based Fully Convolutional Network
R-CNN	Bölge Tabanlı Konvolüsyonel Sinir Ağı	Region Based Convolutional Neural Network
fps	Saniyede Akan Kare	Frames Per Second
CUDA	Birleştirilmiş Aygıt Hesaplama Mimarisi	Compute Unified Device Architecture
GLCM	Gri Seviye Eşzamanlılık Matrisi	Gray Level Co-occurrence Matrix

1. GİRİŞ

Bağcılık, insanoğlunun en eski tarımsal faaliyetlerinden biri olup, ve halen de önemini kayıp etmeden ve iklimi uygun ülkelerde en yaygın tarımsal üretim alanlarından biridir. Bitki hastalıkları tarım sektöründe ürün kaybına ve kalitesinin düşmesine ve dolayısıyla büyük ekonomik kayıplara sebep olmaktadır. Bitkilerde hastalıkların ve zararlıların doğru ve hızlı şekilde tespit edilmesi, ekonomik kayıpları büyük ölçüde azaltırken, erken bir tedavi tekniğinin geliştirilmesine yardımcı olabilir.

Küresel nüfus sürekli artmakta olduğundan, sürdürülebilir tarım prosedürleri kullanılarak doğal ekosistemleri bozmadan dünya genelinde hem mevcudiyetini ve hem de yüksek besin kalitesini koruyarak, gıda üretiminde büyük bir artış sağlanmalıdır. Bu zorlukları ele almak için, karmaşık, çok değişkenli ve öngörülemeyen tarımsal ekosistemlerin sürekli olarak çeşitli fiziksel yönleri ve fenomenleri izleyerek, ölçerek ve analiz ederek daha iyi anlaşılması gerekmektedir. Bu, tarımda büyük veri analizi yapılmasına olanak verir ve hem yıllık mahsul/ çiftlik yönetimi hem de daha kapsamlı ekosistemlerin gözlemleri için, mevcut yönetim ve süreç oluşturma görevlerini, duruma göre iyileştiren yeni Bilgi ve İletişim Teknolojilerin (BİT) kullanılması anlamına gelmektedir.

Derin öğrenme, umut veren sonuçları ve büyük potansiyeli ile görüntü işleme ve veri analizi için en güncel ve kullanışlı bir tekniktir. Derin öğrenme çeşitli alanlarda başarıyla uygulanmış ve aynı zamanda tarım alanına da girmiştir. Akıllı tarım, tarımsal üretimdeki zorlukları verimlilik, çevresel etki, gıda güvenliği ve sürdürülebilirlik açısından önemlidir. Derin öğrenme, Yapay Sinir Ağlarının (YSA) yapısından ve çalışmasından ilham alan makine öğrenmesi alt alanıdır. Bununla birlikte, derin öğrenme, çeşitli konvolüsyonlar yoluyla verilerin hiyerarşik temsilini sağlayan “daha derin” yapay sinir ağları olarak betimlenebilir. Bu, daha büyük öğrenme yeteneklerine ve dolayısıyla daha yüksek performans ve sınıflandırma doğruluğa olanak tanımaktadır.

Tarım, memleketim Afganistan'da en önemli ve eski mesleklerden biridir. Afganistan nüfusunun yaklaşık %80'i tarım ve hayvancılıkla uğraşmaktadır, bu nedenle Afganistan'ın milli gelirin %31'i tarımsal üretim kaynaklıdır ve gıda üretimine azami özen gösterilmektedir. Üzüm bitki tarım alanında büyük ölçüde üretilen bir üründür. Üzüm yetiştiriciliğinin Afganistan'da sosyal ve ekonomik önemi vardır.

Son birkaç yılda dünya çapında üzüm kalitesi birçok nedenden ötürü bozulmuştur (Wilcox ve ark., 2015). En önemli nedenlerden biri hastalıklardır. Hastalıkları önlemek için çiftçilerin büyük miktarda böcek ilaçların püskürtmeleridir, ancak bu da üretim maliyetinin artmasına neden olmaktadır. Ayrıca çiftçiler hastalıkları el ile tanımlayamamaktalardır. Bitkilerdeki hastalıklar sadece bitkilere bulaşıp yayılmaya başlayınca teşhis edilebilir. Ancak tespit edilmesi zaman alır ve bu süre zarfında hastalıklar bağ üzerinde olumsuz etkileyebilir ve ürün kaybına ve kalitesinin düşmesine sebep olabilir.

Üzüm meyvesi ve yaprağı hastalıklarının tanımlanması, tarım ürününün verim ve kalitesindeki kayıpları önlemenin temel amacıdır. Üzüm hastalığını elle takip ve tespit etmek çok zor ve zaman alıcı olmaktadır ve aşırı işlem süresi, muazzam miktarda işlem gerekir. Üstelik bazı hastalıkları tanımada uzmanlık gerektirmektedir. Asma hastalığını tespit etmek için hızlı otomatik işleme ihtiyacımız vardır.

Tezin amacı üzüm çifçiliğin verimliliğini artabilmek için yapay zeka, derin öğrenme tekniklerini ve özellikle Konvolüsyonel Sinir Ağlarını (KSA) kullanarak üzüm meyvesi ve üzüm yaprağı (Asma) hastalıklarını tespit edebilen hızlı ve otomatik bir sistem geliştirmektir. Asma hastalıklarının sınıflandırma çalışması, Tensorflow nesne algılama program arayüzü üzerinde önceden eğitilmiş olan iki derin öğrenme modeli (SSD MobileNet v1 ve Faster R-CNN Inception v2) aktarımlı öğrenme yöntemiyle yeniden eğitilerek gerçekleştirilmiştir. Bu modeller, internetten ve PlantVillage (Hughes ve Salathe, 2015) veri seti gibi kaynaklardan toplanmış ve bu çalışma için özel olarak hazırlanan iki veri setinde yeniden eğitilmişlerdir.

Tez kapsamında yapılan çalışmalar anlatılırken, ikinci bölümde bu çalışma için hedeflenmiş olan asma hastalıklar hakkında genel ve kısa bilgi verilmiştir. Bu çalışmayı daha iyi bir şekilde tamamlayabilmek amacıyla yararlı materyallar ve yöntemlerini seçebilmek için diğer araştırmacıların bilimsel çalışmalarının literatür taraması üçüncü bölümde değinilmiştir. Dördüncü bölümde derin öğrenme kavramından ve yöntemlerinden bahsedilmiş ve makine öğrenmesi ile farkı açıklanmıştır. Beşinci bölümde konvolüsyonel sinir ağları kısaca anlatılmıştır. Altıncı bölümde bu tezde kullanılmış olan materyallar ve yöntemler anlatılmıştır. Yedinci bölümde ise bu çalışmanın deneysel sonuçları ve tartışmaları içermektedir. Son olarak, bu çalışmanın sonucu ve ileride bu çalışmanın geliştirilmesi için yapılabilecek işlemler sekizinci bölümde sunulmuştur.

2. ÜZÜM MEYVESİ VE YAPRAKLARI HASTALIKLARI

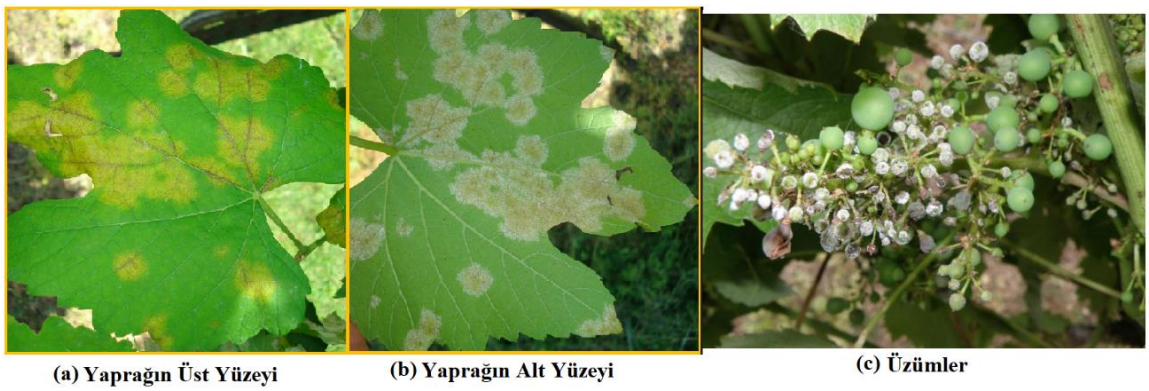
Çiftçilerin ve üzüm piyasası için, asma hastalıkları çok yıkıcı olabilir. Ne yazık ki, her koşulda gelişebilen birçok asma hastalığı türü vardır. Bakteriler ve mantarlar en sık rastlanılan üzüm meyvesi ve yaprağı hastalıklarına neden olurlar. Ayrıca böcekler de üzüm hastalıklarını yayabilir ve köklere zarar verebilirler. Çevresel koşullar da üzüm bağlarında hasarlar yaratan mantar gelişimini tetikleyebilir. Bu tezde çalışılan en yaygın 7 üzüm meyvesi ve yaprağı hastalıkları aşağıda açıklanmıştır.

2.1. Tüylü Küf (Downy Mildew)

Plasmopara viticola'nın neden olduğu tüylü küf, dünyanın birçok yerinde ekonomik açıdan önemli bir asma hastalığıdır. Hem yaprakları hem de meyveyi etkileyen hastalık, bol yağış, yüksek bağıl nem ve üzüm bağlarında uzun süre nem olan iklimlerde özellikle etkilidir (Madden ve ark., 2000).

Bu hastalık üzüm bitkilerini tahrip edebilir. Bu hastalık Mayıs ayının ortasından sonbaharın sonlarına kadar, bitkinin üst kısımları, özellikle sıcaklık 18°C'nin altındayken yayılır. Belirtileri, yağ lekeleri ve küflü bir örtü veya yapraklarda bulunan parlak yeşil lekelerdir. Yaprağın arkasında beyaz küflü bir örtü da bulunur. Enfeksiyon aynı zamanda meyvelere, özellikle ilk büyümeye başlayan tanelere yayılır, üzümün kurumasına ve kuru üzüm gibi görünmesine neden olabilir. Ayrıca yumuşatıp menekşe rengine dönüşebilir (Doman, 2015).

Tüylü Küf hastalığına yakalanmış olan yaprakların ve meyvenin örnek resimleri Şekil 2.1'de gösterilmiştir.



Şekil 2.1. (a, b) Tüylü Küf Üzüm yaprakların üzerinde (Westover, 2013), (c) Tüylü Küf Üzüm meyvelerin üzerinde (Western Australian Agriculture Authority, 2019)

2.2. Külleme (Powdery Mildew)

Külleme, dünya çapında önemli bir asma hastalığıdır. Yağmur, rüzgar gibi doğal hava şartları bu hastalığın yayılması için uygun çevresel şartlardır ve üzüm için yıkıcı olabilir. Külleme mantarının bitkiye bulaşabilmesi için bitki doku yüzeyinde serbest suya ihtiyaç yoktur. Külleme hastalığı üzüm verimin azalmasına, meyvenin kalitesinin zarar görmesine neden olabilir (Ellis, 2008b).



Şekil 2.2. (a) Külleme Üzüm yaprağı üzerinde (Homegrown, 2013), (b) Külleme Üzüm meyveleri üzerinde (Nelson, 2015)

Külleme, asmanın tüm yeşil dokusuna yayılabilir. Küçük, beyaz veya grimsi beyaz fungal büyüme yamaları, şekil 2.2'de gösterildiği gibi yaprağın üst ya da alt yüzeyinde bulunabilir. Bu yamalar genellikle, yaprağın üst yüzeyinin tamamı tozlu, beyaz ila gri olarak kaplanana kadar yayılırlar (Ellis, 2008b).

Şekil 2.2'deki fotoğraflarda görülen etkilenmiş olan üzüm meyveleri, yapraklardakilere benzer yüzeyleri üzerinde mantar büyüme lekelerine sahip olabilir veya tüm meyvenin beyaz, toz halinde büyümesi ile kaplanması mümkündür. Bu hastalığa yakalanmış olan üzüm meyvesinde genellikle şekil bozuklukları veya yüzeyinde paslanmış demir renginde lekeler oluşur (Ellis, 2008b).

2.3. Siyah Çürük (Black Rot)

Siyah çürük en ciddi asma hastalıklarından biridir. Mahsul kayıpları, bağdaki hastalık miktarına, hava durumuna ve çeşitli duyarlılığa bağlı olarak yüzde 5 ile 80 arasında değişir. Guignardia bidwelli mantarı, üzüm bitkisinin tüm yeşil kısımlarına

bulaşır. Daha sonra meyveye bulaşırsa, salkımdaki üzüm tanelerinin büyük kısmını hatta mahsulün tamamını yok edebilir (Demchak, 2017).

Hastalık bulaşmış olan yaprakların üst yüzeyinde kırmızımsı kahverengi, dairesel lekeler bulunur. Hastalık büyüdükçe, yaprağın oratasında kahverengi küçük ve siyah noktalar görünebilir. Hastalığın yayıldığı meyveler ise koyu kahverengiye döner ve yüzeyi çok sayıda siyah mantarla kaplanılır. Meyveler sonunda koflaşır (Demchak, 2017).

Siyah çürük bulaşmış olan yaprağın ve meyvenin örnek resmileri şekil 2.3'de gösterilmiştir.



(a) Üzüm (b) Üzüm Yaprağı

Şekil 2.3. (a) Üzüm meyvesinde Siyah Çürük (Demchak, 2017), (b) Üzüm Yaprağında Siyah Çürük (Schilder, 2011)

2.4. Botrytis Çürüklük (Botrytis Bunch Rot)

Botrytis çürüklüğüne Botrytis cinerea mantarı neden olur. Bu mantar doğada çok yaygındır ve alakasız bitkilerde hastalığa neden olur. Botrytis çürüklüğü, çok hassas üzüm çeşitlerinde ciddi kayıplara neden olur (Ellis, 2008a). Botrytis çürüklük hastalığı bulaşmış olan yaprağın ve üzüm meyvenin örnek resmileri şekil 2.4'te gösterilmiştir.



Şekil 2.4. Botrytis Çürüklük (Ellis, 2008a)

Olgun meyvelerin enfeksiyonu bu hastalığın en yaygın ve yıkıcı aşamasıdır. Bu üzüm meyveleri önce yumuşak ve sulu görünürler. Beyaz çeşitlerinde üzümler kahverengileşip büzüşürler, siyah çeşitlerin kırmızımsı bir rengi olurlar. Yüksek bağıl nem ve rutubet altında, üzüm meyveleri genellikle mantar miselyumunun gri büyümesiyle kaplanırlardır. Salkım içindeki birkaç tane ya da salkımın tamamı etkilenebilir. Genellikle, hastalıklı meyvelere dokunan sağlıklı meyveler de enfekte olurlar (Ellis, 2008a).

2.5. Blister Akarı (Blister Mite)

Blister akarı, sadece asmalarda ortaya çıkan bir tür akardır. Asma blisteri akarı gall akarın türlerinden birisidir. Gall akarları, çeşitli bitkilerde çeşitli anormal büyümelere neden olan minik (0,3mm'den küçük) sap emici asalaklardır (The Royal Horticultural Society, 2019).



Şekil 2.5. Blister Akarı (The Royal Horticultural Society, 2019)

Belirtiler ilkbaharın sonlarında ortaya çıkmaya başlar. Şekil 2.5'te gösterildiği gibi bu hastalığın bulaştığı yaprağın üst yüzeyi yukarıya doğru çıkıntı yapar ve yaprak bombeli bir görünüm alır. Bombeli alanların altında, akarların yaşadığı ve beslendiği yoğun bir ince tüy kaplaması gelişir. Tüyler genellikle beyaz kremsidir, ancak bazı üzüm çeşitlerinde, tüyler pembemsi kırmızı olabilir. Yaz sonunda tüyler kurumaya başlar ve kahverengimsi hale gelir (The Royal Horticultural Society, 2019).

2.6. Esca

Esca asmaların gövdesinde (trunk) gelişen hastalıkların türlerinden biridir. Bu hastalıklar tipik olarak uzun süre sonunda asmalarda hasara yol açar. Bu patojenler daha "sessiz bir katil" gibi hareket eder ve mahsulun ömrünü 5-10 yıl kısaltabilir (Mugnai ve ark., 1999).



Şekil 2.6. Esca hastalığı (Mugnai ve ark., 1999)

Belirtileri, yaprak yeşilinde açık yeşil veya klorotik, yuvarlak veya düzensiz lekeler oluşur. Başlangıçta küçük ve yaprakların üzerine saçılan lekeler, yavaş yavaş genişler ve birleşir, kısmen nekrotik hale gelir ve sonuçta etkilenmemiş yeşil yaprakta dar şeritler halinde yayılır. Şekil 2.6'de göründüğü gibi yaprak sarı-kahverengi veya kırmızı-kahverengiye döndüğü zaman, hastalıklı yapraklar bir "kaplan çizgileri" şeklini alır. Bazen yaprağın nekrotik alanları kurur ve yaprağın kenar boşlukları bırakarak ayrılırlardır (Mugnai ve ark., 1999).

2.7. Yaprak Yanıklığı (Leaf blight)

Alternaria yaprak yanıklığı olarak da bilinir. Mantar Alternaria cucumerina, Yaprak Yanıklığına neden olur. Bu mantar mevsimden mevsime bitki kalıntılarında yaşayabilmektedir. Rüzgar akımları Alternaria cucumerina'yı uzak bir mesafeye taşır. Alternaria cucumerina, su sıçratarak alana da yayılabilir. Islak, yağışlı hava bu hastalığı desteklemektedir. Sıcak, ıslak koşullarda hasar çok ağır olabilir (Grabowski, 2018).



Şekil 2.7. Yaprak yanıklığı hastalığı (Grabowski, 2018)

Şekil 2.7'de göründüğü gibi küçük sarımsı lekeler önce yaprak kenarlarında görünür, bu lekeler yavaş yavaş büyür ve konsantrik halkalarla kahverengimsi lekelere dönüşür. Ciddi derecede enfekte olmuş yapraklar kahverengiye döner, yukarı kıvrılır, büzüşür ve ölürler. Meyveler yaprak kaybından dolayı güneş yanarak zarar görebilirler. Bitki canlılığının azalması ve maruz kalan meyvelerin güneşten yanması yüzünden verim ve kalite düşebilir (Grabowski, 2018).

3. İLGİLİ ÇALIŞMALAR

İkinci bölümde anlatılan asma hastalıklarını tespit edebilen ve tanıyabilen bir sistem geliştirilmesine başlamadan önce, bir literatür taraması yapılması gerekmektedir. Literatür taramasından amaç, diğer araştırmacıların by çalışmayla benzer çalışmalarını değerlendirilmesidir. Araştırmacıların tarafından kullanılan yöntem ve teknikleri, elde edilen sonuçları ve önerilen yöntemleri öğrenilmesi gerekmektedir. Bu, çalışmanın hedefine ulaşılabilmek için en iyi yöntemlerin seçilmesine yardımcı olacaktır.

Bitki hastalıklarının ve zararlıların araştırılması dahil olmak üzere tarım alanında çeşitli yaklaşımlar kullanılmaktadır. Görüntü işleme teknikleri yanı sıra derin öğrenme de uygulanmıştır. Tarım alanında en sık kullanılan makine öğrenme algoritmaları yaygın olarak benimsenmiştir.

Birçok araştırmacı, farklı teknikleri kullanarak farklı bitki hastalıklarını tespit edebilmek için çalışmış ve önerilen yöntemleri konusunda oldukça iyi sonuçlar elde etmişlerdir. Bu çalışmanın konusuyla ilgili son yıllardaki bazı çalışmalardan aşağıda bahsedilmiştir:

Kakade ve Ahire'nin çalışmasında üzüm yaprağındaki hastalıkları tespit etmek için görüntü işlemeye dayalı yeni bir yaklaşım geliştirilmiştir. Amaçları, hastalığın ilk belirtilerini tespit etmek, tanımlamak ve doğru bir şekilde ölçmek olmuştur. Yapılan çalışmada hastalığı tanıma aşamaları; görüntü elde etme, görüntü ön işleme, özellik çıkarıma ve sinir ağlar temelli sınıflandırmadır. Geliştirilen algoritmanın sonuçları, incelenen örneklerde %92.94 başarıyla hastalığı tespit edip sınıflandırabildiğini göstermiştir (Kakade ve Ahire, 2015).

Mohanty ve ark. tarafından, bitki yapraklarındaki hastalıklarının analizi için derin KSA kavramı kullanılmıştır. Bu çalışmada derin öğrenme temelli iki tane model (AlexNet ve GoogLeNet) seçilmiştir. Hastalıklı ve sağlıklı bitki yapraklarının 54,306 görüntüsünden oluşan veri setini kullanılarak, 14 bitki türü ve 26 hastalığı içeren 38 farklı sınıfın mahsul türlerini ve hastalık durumunu sınıflandırmak için bu iki KSA eğitilmiş ve %99'un üzerinde başarı elde edilmiştir (Mohanty ve ark., 2016).

Sladojevic ve ark. tarafından, yaprak görüntü işleme dayanan ve KSA kullanılarak bitkinin hastalıklarını tanıyabilen bir model geliştirmek için yeni bir yaklaşım sunulmuştur. Daha verimli sonuçlar elde etmek için ince ayar (fine tuning) tekniği modele

eklenmiştir. Geliştirilen model, bitki yapraklarını çevrelerinden ayırt etme yeteneği ile, 14 farklı bitki hastalık türünü sağlıklı yapraklardan ayırt edebilmiştir. Deneysel sonuçlar, %91 ile %98 arasında olmuştur, ayrı sınıf testleri için, eğitilen modelin nihai genel başarısı %96.3 olarak belirtilmiştir. İnce ayar, genel başarıda önemli değişiklikler göstermemiştir, ancak veri büyütme (Data Augmentation) süreci, önemli sonuçların elde edilmesinde daha büyük bir rol oynamıştır (Sladojevic ve ark., 2016).

Hanson ve ark. tarafından, derin öğrenme teknikleri kullanılarak yapraklardan hastalığı tespit etmek ve sınıflandırmak için yeni bir yaklaşım sunulmuştur. İlk önce bir veritabanı oluşturmak için, tarımsal uzmanlar tarafından değerlendirilmiş olan görüntüleri toplamakla başlanmış, görüntü ön işleme, büyütme ve son olarak derin konvolüsyonel sinir ağları ve ince ayar eğitim prosedürleri kullanılmıştır. Sonuç olarak, ayrı sınıf testleri için %91 ile %98 arasında başarı elde edilmiştir. Eğitilen modelin nihai genel başarısı %95 olmuştur. İnce ayar, genel başarıda önemli değişiklikler göstermemiştir, ancak büyütme süreci, önemli sonuçlara ulaşmada daha etkili olmuştur (Hanson ve ark., 2017).

Tiwari ve Gupta tarafından, bitki yaprağındaki hastalıkların tespit edilebilmesi ve sınıflandırılması için değiştirilmiş SVM-Cuckoo aramasının özerk (autonomous) bir yaklaşım önerilmiştir. SVM'nin sınıflandırma parametrelerini optimize etmek için Destek Vektör Makinesi (SVM) ile Cuckoo arama konsepti birleştirilmiştir. Deneyler için Alternaria Alternata, Cercospora Yaprak Noktası, Antrasnose ve Bakteriyel Blight ile zarar görmüş yaprakların ve bazı sağlıklı yaprakların görüntülerini içeren bir veri kümesi kullanılmıştır. Değiştirilmiş SVM-Cuckoo araması için analiz edilmiş başarı seviyesi %96.5 ile %98.5 arasında olmuştur. Son olarak önerilen kavramın, yapraktaki hastalıkları tespit ettiği için çok verimli olduğu özetlenmiştir (Tiwari ve Gupta, 2017).

Fuentes ve ark. tarafından, Domates bitkisindeki hastalıkları tespit etmek için, çeşitli çözünürlüklere sahip kameralar ile çekilen görüntüler kullanılarak sağlam bir derin öğrenme temelli bir yaklaşım önerilmiştir. Üç ana detektör ailesi kullanılmıştır: Daha hızlı Bölge tabanlı konvolüsyonel sinir ağları (Faster R-CNN), Bölge Tabanlı Tam KSA (R-FCN) ve Tek Atış Çok Kutulu Dedektörü (SSD). Bu üç algoritma “meta mimarileri” olarak isimlendirilmiştir. Bu meta-mimarilerin her birini Görsel Geometri Grubu (VGG net) ve Residual Network (ResNet) gibi “derin özellik çıkarıcılar” ile birleştirilmiştir. Önerilen sistem, 9 farklı hastalık türünü etkili bir şekilde tanıyabilmiş ve bir bitkinin çevresindeki alandan karmaşık senaryolarla başa çıkabilmiştir. Yetersiz örnek sayısı

nedeniyle, yüksek desen varyasyonuna sahip olan bazı sınıflar diğer sınıflarla karışmış ve yanlış pozitif veya daha düşük ortalama hassasiyet ile sonuçlar elde edilmiştir. Buna ek olarak, teknik temelli veri ek açıklaması (data annotation) ve veri büyütme yöntemlerin kullanılması daha iyi performansla sonuçlandığı bulunmuştur (Fuentes ve ark., 2017).

Jeon ve Rhee'nin çalışmasında, KSA modelini kullanarak yaprakların sınıflandırılması için yeni bir yöntem önerilmiştir. Bunun için iki KSA model (GoogleNet ve GoogleNet'in türevleri) seçilmiş ve test edilmiştir. Bu çalışmada Flavia veri kümesi kullanılmıştır. Bu veri kümesi sarelanceolate, lightoval, akupunktur, lineer, uzun oval, uzatılmış, kalp ve uzun yaprak olmak üzere 8 yaprak türünü içermektedir. Her modelin performansı, yaprakların renginin bozulmasına veya hasar görmesine göre değerlendirilmiştir. Sonuç olarak elde edilen tanıma oranı, yaprak hasarının %30 olduğu durumlara rağmen %94'ten daha fazla olmuştur (Jeon ve Rhee, 2017).

Y. Lu ve ark. tarafından, derin KSA tekniklerine dayanan pirinç hastalıklarını tanımlamak için yeni bir yöntem önerilmiştir. KSA'nın daha hızlı bir yakınsamaya sahip olmasını sağlamak ve üstün değişmez özellikleri işlerken genelleme yeteneğini geliştirmek için, önerilen KSA modelinde stokastik havuzlama (stochastic pooling) yöntemi uygulanmıştır. Daha sonra KSA'yı eğitmek için gradyan inişi (gradient-descent) algoritması kullanılmıştır. Hastalıklı ve sağlıklı pirinç yapraklarının ve pirinç tarlalarından alınan sapların 500 doğal görüntüsü kullanılarak, KSA en yaygın 10 pirinç hastalığının belirlenmesi için eğitilmiştir. 10 kat çapraz doğrulama stratejisi kapsamında, önerilen KSA tabanlı model %95.48'lik bir başarı elde etmiştir (Y. Lu ve ark., 2017).

Durmuş ve ark tarafından, PlantVillage veri kümesi kullanılarak sağlıklı görüntüler de dahil olmak üzere 10 farklı domates hastalık sınıfının belirlenmesi ve sınıflandırılması için iki farklı derin öğrenme ağ mimarisi (AlexNet'i kullandıktan sonra SqueezeNet'i) kullanılmıştır. Derin öğrenme algoritmasının robot üzerinde gerçek zamanlı olarak çalışması hedeflenmiştir. Bu derin öğrenme ağlarının her ikisi için Nvidia Jetson TX1'de eğitim ve doğrulama yapılmıştır. AlexNet, %95.65'lik bir başarıyla sonuçlanırken, SqueezeNet, daha düşük hesaplama yükünde %94.3'lük bir verim sağlamıştır (Durmuş ve ark., 2017).

McCool ve ark. tarafından, AgBot II (automated weed management) gibi robotik platformlarda uygun hafif modellerin öğrenilmesi için karmaşıklık ve doğruluktan kurtulmaya olanak veren derin KSA'ı eğitmek için yeni bir yaklaşım önerilmiştir. İlk

olarak, en gelişmiş performansı sağlamak için yüksek hesaplama karmaşıklığı maliyetine göre önceden eğitilmiş olan bir model (Inception-v3) uyarlanmıştır. İkincisi, 25M parametreleri ile hafif bir derin KSA öğrenmek için uyarlanmış model (adapted model) üzerinde sıkıştırma teknikleri kullanılmıştır. Üçüncü olarak, değişken K hafif modeli, hafif modellerin performansını daha da artırmak için bir karışım modeli olarak birleştirilmiştir. Ot segmentasyonunun zorlu görevine uygulandığında, önceden eğitilmiş olan bir derin KSA'ı uyarlayarak başarı %85.9'dan %93.9'a çıkartılmıştır. Ancak bu uyarlanmış model sadece saniyede 0.12 resim (fps) işleyebilmiştir. Bu yaklaşımı başarısını koruyarak hızlandırmak için, K=4 hafif derin KSA öğrenir, bu da birleştirildiğinde, 1.07 ve 1.83fps arasında işleyebilen çok daha az sayıda parametre kullanılırken kombine edildiğinde % 90'dan daha yüksek bir başarı elde etmiştir (McCool ve ark., 2017).

Ienco ve ark. tarafından, bir zaman serisindeki uydu görüntülerinden elde edilmiş çoklu zamansal mekansal verileri göz önüne alınarak, Özyinelemeli Sinir Ağı (RNN), özellikle de Uzun Kısa Süreli Bellek (LSTM) modelinin, arazi örtüsü sınıflandırma yeteneğinin değerlendirilmesi yapılmıştır. Hem piksel tabanlı (Reunion Island) hem de nesne tabanlı (Thau basin) veri setlerini sınıflandırma çalışmaları gerçekleştirilmiştir. Elde edilen sonuçlara göre, RNN, en gelişmiş sınıflayıcılarla karşılaştırıldığında rekabetçi olduğu ve düşük temsil edilen veya çok karışık sınıfların varlığında klasik yaklaşımlardan daha iyi performansa sahip olduğunu göstermiştir. Ayrıca, LSTM tarafından oluşturulan alternatif özellik gösteriminin standart sınıfların performansını artırabileceği gösterilmiştir (Ienco ve ark., 2017).

H. Lu ve ark. tarafından derin KSA ve Aktarımlı öğrenme'ye (DTCLE) dayalı, İnsansız Hava Aracı (İHA) görüntülerinde (insansız hava aracı, düşük yükseklikte uzaktan algılama tekniği, esnek, düşük maliyetle ve yüksek çözünürlükle verimli olan) ekili arazi bilgileri için yeni bir çıkarma yöntemi önerilmiştir. İlk olarak, Derin KSA'ya dayalı doğrusal özellikler (yollar ve köprüler vb.) çıkartılmıştır. Daha sonra, derin KSA'dan öğrenilmiş olan özellik çıkarma yöntemi ve aktarımlı öğrenme mekanizmasını kullanarak arazi bilgi çıkarma işlemleri gerçekleştirilmiştir. Son olarak, ekili arazi bilgileri çıkarma sonuçları DTCLE ve eCognition tarafından tamamlanmıştır. Sichuan eyaletinin Pengzhou ve Guanghan ilçeleri deneysel amaçlar için seçilmiştir. Elde edilen deneysel sonuçlarda, (görüntü 1, 2 ve 3 ekili alanın çıkarılması) genel hassasiyetleri DTCLE yöntemiyle sırasıyla %91.7, %88.1 ve %88.2 olduğu ve ECLE'nin %90.7, %90.5

olduğu gösterilmiştir. Sonuç olark, DTCLE'in doğruluğunun ECLE'ninkine eşdeğer olduğu ve ayrıca bütünlük ve süreklilik açısından ECLE'den daha iyi sonuçlandığı özetlenmiştir (H. Lu ve ark., 2017).

Amara ve ark. tarafından, muz yaprakları sınıflandırma sürecini otomatik hale getiren ve muzun (muz sigatoka ve muz benekleri) iki yaygın hastalığının gerçek zamanlı tanımasını sağlayan, derin öğrenme temelli bir yaklaşım önerilmiştir. Görüntülerin sınıflandırması için bir KSA olarak LeNet mimarisi, açık kaynaklı derin bir öğrenme kütüphanesi olarak deeplearning4j ve PlantVilage veri setinden elde edilen veriler kullanılmıştır. Sonuç olarak, RGB görüntülerin yaklaşık %99'u ve gri tonlamalı görüntüler için yaklaşık %90'lık bir başarı elde edilmiştir. Ön sonuçlar, aydınlanma, karmaşık arka plan, farklı çözünürlük, boyut, poz ve gerçek sahne görüntülerinin yönlendirilmesi gibi zorlu koşullar altında bile önerilen yaklaşımın etkinliğini göstermiştir (Amara ve ark., 2017).

Rahnemoonfar ve Sheppard tarafından KSA ve Inception-ResNet'in değiştirilmiş bir versiyonunu dayanarak meyveleri saymak (nesneleri algılamadan saymak) için yeni bir derin öğrenme mimarisi geliştirilmiştir. Bu yöntem, simüle edilmiş veriler (üretilen sentetik domates görüntüleri) üzerinde eğitilmiş, ancak gerçek veriler üzerinde test edilmiş, simülasyon temelli bir öğrenme yöntemidir. Rasgele seçilmiş 100 tane gerçek görüntü için %91 doğruluk gözlemlenmiştir. Algoritma kötü koşullar altında da doğru çalışmıştır, domatesler gölgede olsa, yeşillik, dallar tarafından gizlenmiş olsa ya da domatesler arasında bir dereceye kadar örtüşüyorsa da doğru sayabilmiştir. Bu algoritma domatesleri saymak için eğitilmiş olsa da, diğer meyvelerde de uygulanabilir olduğu iddia edilmiştir. Olgunlaşmış ve yarı olgunlaşmış meyveleri sayabilir; Ancak, yeşil meyveler saymakta başarısız olmuştur, çünkü bu amaç için eğitilmemiştir (Rahnemoonfar ve Sheppard, 2017)

Chen ve ark. tarafından, yöntemlerinin yapılandırılmamış ortamlarda meyveyi doğru olarak saydığını iddia ederek, derin öğrenmeye dayalı bir meyve sayımı akış hattı önerilmiştir. Büyük veri kümelerini hızlı bir şekilde etiketlemek için özel bir kitle kaynak platformu kullanan akış hattı, tamamen KSA'ya dayalı bir blob dedektörü görüntülerinde olası bölgeleri çıkartmıştır. Daha sonra ikinci bir konvolüsyon ağına dayanan bir sayım algoritması her bir bölgedeki meyve sayısını tahmin etmiştir. Son olarak, doğrusal regresyon modeli, meyve sayımının son meyve sayımına yakın sonuçlar bulunduğunu göstermiştir. Akış hattının performansı gün ışığında iki farklı renkteki portakal veri

kümesinde ve gece yeşil elma üzerinde analiz edilmiştir. Bu sistem insan kaynaklı etiketlemenin yerine kullanılmaya başlamıştır. Son olarak da akış hattının kısa bir eğitim süresine sahip olduğu ve sınırlı bir veri seti boyutuyla iyi performans gösterdiği ortaya konulmuştur. Yöntem, her iki veri setinde de genelleştirilmiş ve insan etiketleyicilerinin açıklama yazmasını zorlaştıran gizlenmiş meyvelerde bile iyi performans gösterebilmiştir (Chen ve ark., 2017).

Mude ve ark. tarafından, görüntü işleme teknikleri ve YSA'ı kullanarak pamuk bitkisinin yaprağında gelişen hastalıkları tespit etmek için bir yöntem önerilmiştir. Dört çeşit hastalık (Rendening, Rust, Bakteriyel, Fussarium) tespit edilmeye çalışılmıştır. Elde edilen sonuçlarda Rendening ve Rust hastalıklarının pamuk bitkisinde en çok görülen hastalıklar olduğunu gösterilmiştir. Önerilen yöntem yaklaşık %84 oranında doğruluk sağlamıştır (Mude ve ark., 2017).

Namin ve ark. tarafından KSA ve LSTM algoritmalarının avantajını kullanılarak ve bu algoritmaları birleştirilerek bitki fenotipini tanımak için yeni bir derin öğrenme yapısı önerilmiştir. Geliştirilen algoritmaya göre, bir dizi model oluşturmak için, KSA resimlerden derin özellikleri çıkarılmış ve çıktı bir LSTM birimine yüklenmiştir. Deney sonuçları, KSA kullanıldığında el işi (handcrafted) özellikleri uygulandığında dizi modelinin doğruluk oranını %68'den %76.8'e önemli ölçüde arttırdığını ve LSTM ile zamansal bilginin ilave edilmesinin de doğruluğu %93'e çıkardığını göstermiştir (Namin ve ark., 2018).

Wang ve ark. tarafından, yaban mersini hiperspektral görüntülerini sağlam ve hasarlı gruplara ayırmak için derin KSA uygulanmıştır. Hiperspektral verilere dayanarak yaban mersinindeki iç mekanik hasarlarını tespit etmek için ResNet ve ResNeXt derin KSA kullanılmıştır. Sınıflandırma görevini tamamlamak için, temel modelleri olarak toplam beş makine öğrenme algoritması (Sıralı Minimal Optimizasyon (SMO), Doğrusal Regresyon (DR), Rastgele Orman (RO), Bagging, ve Çok Katmanlı Perceptron (MLP)) kullanılmıştır. Çalışmalarında, hassas ayarlı ResNet/ ResNeXt için sırasıyla, 0.8844/ 0.8784 ortalama başarısı ve 0.8952 /0.8905 ve F1-skoru elde edilmiştir. SMO/ DR/ RO/ Bagging / MLP sınıflandırıcılarda, sıralarıyla, 0.8082/ 0.7606/ 0.7314/ 0.7113/ 0.7827 ortalama doğruluk ve 0.8268/ 0.7796/ 0.7529/ 0.7339/ 0.7971 ve F1-skoru elde edilmiştir. Sonuç olarak bu iki derin öğrenme modelinin geleneksel makine öğrenme yöntemlerinden daha iyi sınıflandırma performansı elde edebilecekleri özetlenmiştir. Her bir test numunesi için sınıflandırma, ResNet ve ResNeXt için sırasıyla 5.2ms ve 6.5ms

zaman almıştır. Bu süreler de derin öğrenme çerçevesinin anlık meyve sınıflandırması için büyük bir potansiyele sahip olduğunu göstermiştir. Çalışmanın sonuçları, meyvelerin iç mekanik hasarlarının analizinde derin KSA uygulamasının potansiyelini göstermiştir (Wang ve ark., 2018).

Rangarajan ve ark. tarafından, AlexNet ve VGG16 net adları verilen önceden eğitilmiş derin öğrenme mimarilerini kullanılarak PlantVillage veri seti görüntülerinden domates bitkisi hastalıklarının sınıflandırması yapılmıştır. Çalışmaları sonucunda, 13.226 görüntü kullanarak sınıflandırma başarısı VGG16 net için %97.29 ve AlexNet için %97.49 olmuştur. Modelin performansı, görüntü sayısı değiştirilerek, çeşitli mini-batch boyutları ayarlanarak, ağırlık ve bias öğrenme oranı değiştirilerek değerlendirilmiştir. Görüntülerin sayısı, modellerin performansını önemli ölçüde etkilediği göstermiştir. Görüntülerin sayısı 373 olduğunda maksimum başarı elde edilmiştir. AlexNet'teki mini-batch boyutunun ince ayarlanması, sınıflandırma başarısı ile net bir korelasyon göstermemiştir, ancak VGG16net'teki başarı, mini-batch boyutu arttıkça azalmıştır. Benzer şekilde, ince ayarlı ağırlık ve bias öğrenme oranı kullandıktan sonra, AlexNet'teki başarısı, öğrenme oranı 30'a kadar azalmış ve daha sonra keskin bir şekilde artmıştır. VGG16 ağında başarı, ağırlık ve bias oranı arttınca düşmüştür. Hesaplama yükü açısından, AlexNet, derin VGG16 ağıyla karşılaştırıldığında minimum uygulama süresiyle iyi bir başarı oranı sağlanmıştır (Rangarajan ve ark., 2018).

Minh ve ark. tarafından, iki RNN tabanlı sınıflandırıcı, bir LSTM ağı ve bir Kapılı Özyinelemeli Birim (GRU) ağı eğitilmiş ve onları tek-zamansal modellerle birlikte kış bitki örtüsünün kalite kapsamını haritalandırmada uygulanmıştır. Klasik makine öğrenme yöntemlerine (RO ve SVM) göre RNN bazlı sınıflandırma yaklaşımları kullanılarak önemli bir performans artışı gözlemlenmiştir. Elde edilen sonuçlar, GRU modelinin 5 kat çapraz doğrulama (5-fold cross validation) veri seti için % 99.05, LSTM % 98.83, SVM % 91.25 ve RO'nın %91.79 doğruluğu ile diğer tüm modellerden üstün olduğunu göstermiştir (Minh ve ark., 2018).

Kamilaris ve Prenafeta-Boldu'nun inceleme makalesinde, çeşitli tarım ve gıda üretim zorluklarına uygulanan derin öğrenme tekniklerini kullanan 40 çalışma araştırılmıştır. Çalışmalarında belirli tarımsal sorunları, kullanılan spesifik modelleri ve çerçeveleri, kullanılan verilerin kaynakları, doğası ve ön işlemleri, her çalışmada kullanılan ölçütlere göre elde edilen genel performansları incelenmiştir. Bu incelemenin ardından sınıflandırma veya regresyon performansındaki farklılıklar bakımından, diğer

mevcut popüler tekniklerle derin öğrenmenin karşılaştırılması yapılmıştır. Sonuç olarak, derin öğrenmenin, yaygın olarak kullanılan görüntü işleme tekniklerinden daha iyi performans göstererek yüksek başarı sağlayabildiği özetlenmiştir. Bu araştırmanın amacı, daha fazla araştırmacılarının derin öğrenmeyi denemeye teşvik etmek, sınıflandırma veya tahminleme ile ilgili çeşitli tarımsal problemleri çözmede, bilgisayarlı görme ve görüntü analiziyle ya da daha genel olarak veri analiziyle uygulanmalarını sağlamak olmuştur (Kamilaris ve Prenafeta-Boldu, 2018).

Hunt ve Daughtry'nin çalışmasında, tarımın uzaktan algılaması ve hassas tarım için insansız hava araçları ne işe yaradığı incelenmiştir. Tarımsal amaçlar için İHA'ların potansiyel ve belki de en önemlisi mevcut sınırlamalarına genel bir şeklinde inceleyip sunulmuş ve USDA Ekonomik Araştırma Raporu olarak, ABD'deki çiftçilerin sadece %20'sinin teknolojiyi benimsedikleri gösterilmiştir (Hunt ve Daughtry, 2018).

Kothavale ve Barbade'nin çalışmasında, üzüm yaprağı hastalıklarının tespit edilmesinde ve sınıflandırılmasında fayda sağlayacak bir yöntem önerilmiştir. Önerilen yöntem dört aşamayı (görüntü elde etme, histogram eşitleme, Gri Seviye Eşzamanlılık Matrisi (GLCM) özelliği ve SVM sınıflandırması) içermektedir. Gelecekte bu yöntemin iyi sonuçlar vereceği iddia edilmiştir (Kothavale ve Barbade, 2018).

Kothawale ve ark. tarafından, üzüm yaprağı hastalıklarını tespit etmek ve sınıflandırmak için görüntü işleme teknikleri ve SVM sınıflandırıcı teknikleri kullanılmıştır. Modelleri, 90 adet sağlıklı ve hastalıklı üzüm yaprağı görüntüsü üzerinde eğitilmiştir. Sonuç olarak % 89.90 doğruluk elde edilmiştir (Kothawale ve ark., 2018).

Horea ve Mihai'nin çalışmasında, 81 farklı sınıfa sahip olan 55244 tane meyve görüntüsü içeren yeni, yüksek kaliteli bir veri seti hazırlanmıştır ve ayrıca derin sinir ağı kullanılarak meyveleri tespit etme işlemi de denenmiştir. Önce meyveleri bir motor tarafından döndürülürken meyvelerin filmi çekilmiş ve sonra çekilen filmden gürüntüleri çıkartılmıştır. Veri setlerinde farklı ağ konfigürasyonları için farklı sonuçlar elde ediltmiştir. Sonuç olarak %99,42 maksimum eğitim doğruluk ve %96,52 maksimum test doğruluk elde edilmiştir (Horea ve Mihai, 2018).

Too ve ark. tarafından, farklı derin öğrenme modellerinin kullanımı, VGG ağı 16, Inception V4, 50, 101 ve 152 katmanlara içeren ResNet ve 121 katmanı olan DenseNet, plantVillage veri kümesinin 14 farklı bitkinin sağlıklı ve hastalıklı yaprak görüntüleri dahil olmak üzere 38 farklı sınıfın sınıflandırılması için değerlendirilmiştir. DenseNets

modelin, dönme sayıları arttınca giderek başarısı da artmış ve aşırı eğitme ve performans bozulma belirtisi de olmamıştır. DenseNets, son teknoloji performanslarını elde etmek için oldukça az sayıda parametre ve makul hesaplama süresi kullanmıştır. Bu model %99.75'lik bir test başarısı göstererek diğer modelleri geride bırakmıştır. Modeller Theano ve Keras arka uçları kullanılarak eğitilmiştir (Too ve ark., 2019).

4. DERİN ÖĞRENME

Son yıllarda, yapay zeka yoğun medya yutturmacasına konu olmuştur. Makine öğrenmesi, derin öğrenme ve yapay zeka'dan sayısız makelede bahsedilmiş hatta çoğu zaman teknoloji fikirli yayınların dışında da gündeme taşınmıştır (Chollet, 2017).

Yapay zeka, makine öğrenmesi ve derin öğrenmesi kavramlarının ilişkisi şekil 4.1'de gösterilmiştir.



Şekil 4.1. Derin öğrenme, Makine öğrenme ve Yapay zeka (Chollet, 2017)

4.1. Yapay Zeka

Yapay Zeka, çevrelerini algılayan ve belirli bir hedefe ulaşma şansını en üst düzeye çıkaracak bir eylem planı tanımlayan cihazların çalışmasıdır (Karunakaran, 2018a).

Yapay zeka kavramı, 1950'lerde ortaya çıkmıştır, bilgisayar bilimleri alanından gelen bir çok öncü tarafından bugüne kadar sonuçları araştırılan sorunun, bilgisayarların "düşünebilme" yeteneğinin verilip verilmeyeceğinin sorulmasıyla başlamıştır. Alanın kısa ve öz bir tanımı şöyle yapılmıştır: normalde insanlar tarafından gerçekleştirilen entelektüel görevleri otomatikleştirme çabasıdır. Bu nedenle, yapa zeka, makine öğrenmeyi ve derin öğrenmeyi kapsayan genel bir alandır. Aynı zamanda herhangi bir öğrenmeyi içermeyen daha birçok yaklaşımı da içerir. Örneğin, eski satranç programları, yalnızca programcılar tarafından hazırlanmış kodlanmış kuralları içermekteydiler ve makine öğrenmesi olarak nitelendirilmemiştir. Oldukça uzun bir süre boyunca, birçok uzman, insan düzeyindeki yapay zekanın, programcıların bilgiyi manipüle etmek için yeterince geniş bir açık kurallar setine sahip olmasıyla sağlanabildiğine inanıyorlardı. Bu yaklaşım sembolik yapay zeka olarak bilinir ve 1950'lerden 1980'lerin sonlarına kadar yapay zekada baskın paradigma olmuştur (Chollet, 2017).

Her ne kadar sembolik yapay zeka, satranç oynamak gibi iyi tanımlanmış, mantıksal sorunları çözmek için uygun görünse de, görüntü sınıflandırma, konuşma tanıma ve dil çevirisi gibi daha karmaşık, bulanık problemleri çözmek için açık kurallar bulmak zordur. Sembolik yapay zekanın yerini almak için makine öğrenmesi yaklaşımı ortaya çıkmıştır (Chollet, 2017).

4.2. Makine Öğrenmesi

Makine öğrenmesi makinelerin açık bir şekilde programlamaya gerek kalmadan belirli bir görevi nasıl tamamlayacaklarını öğrendikleri yapay zekanın bir alt kümesidir (Karunakaran, 2018a).

Makine öğrenmesi, "Bir makinenin işlemlerini yerine getirtmesi hakkında ne biliyoruz" un ötesine geçip "kendi başına belirli bir görevi nasıl gerçekleştireceğini öğrenebilir mi?", "Bir bilgisayar bizi şaşırtabilir mi?", "Programcıların veri işleme kurallarını elle oluşturması yerine, bir bilgisayar verilere bakarak bu kuralları otomatik olarak öğrenebilir mi?" ve benzeri sorulardan geliştirilmiştir (Chollet, 2017).

Bu sorular yeni bir programlama paradigmasına yol açmıştır. Şekil 4.2'de gösterildiği gibi, klasik programlamada, sembolik yapay zeka paradigması, insanlar kuralları (bir program) ve bu kurallara göre işlenecek veriler girdi olarak veriliyordu ve sonuç olarak cevaplar elde ediliyordu. Makine öğrenmesinde verilerden beklenen cevaplar verilerle birlikte girdi olarak verilir ve sonuç olarak kurallar elde edilir. Bu kurallar daha sonra orijinal cevapları üretmek için yeni verilere uygulanabilir.



Şekil 4.2. Makine öğrenmesi: yeni bir programlama paradigması (Chollet, 2017)

Bir makine öğrenme sistemi programlanmak yerine eğitilir. Bir görevle ilgili birçok örnek sunulur ve bu örneklerle sistemin sonunda görevi otomatikleştirmek için kurallar oluşturmasına izin veren istatistiksel yapıyı bulur (Chollet, 2017).

4.3. Derin Öğrenme

Derin Öğrenme, işlemlerin parçalandığı ve ardışık katmanlarda düzenlenen makine öğrenme algoritmalarına dağıtıldığı bir makine öğrenmenin alt kümesidir. Her katman önceki katmandan çıktıyı almaktadır. Katmanlar birlikte, insan beynindeki nöronlar tarafından gerçekleştirilen problem çözme için dağıtılmış yaklaşımı taklit eden yapay bir sinir ağı oluşturmaktadır (Karunakaran, 2018a).

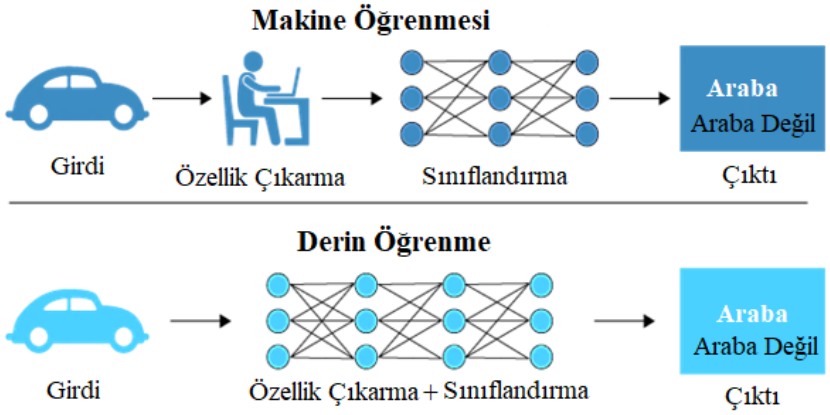
Derin öğrenmede "derin" kelimesi, yaklaşımın başardığı hiçbir derinlemesine anlayış demek değildir; bunun yerine, ardışık temsil katmanları fikrini ifade eder. Bir veri modeline kaç katmanın katkıda bulunduğu, modelin derinliği olarak adlandırılır (Chollet, 2017). Derin öğrenme, YSA olarak adlandırılan beynin yapısından ve işlevinden esinlenen algoritmalarla uğraşan bir makine öğrenme alt-alanıdır (Zhang ve ark., 2018). Başka bir deyişle, beynimizin işleyişini yansıtmaktadır. Derin öğrenme algoritmaları, her nöronun birbirine bağlandığı ve bilgi aktarıldığı insan sinir sisteminin yapılandırılmasına benzemektedir (Karunakaran, 2018a).

Son yıllarda, büyük verilere ve güçlü donanıma bağlı olarak, derin öğrenme, metin, görüntü ve ses sinyalleri gibi karmaşık yüksek boyutlu verilerin işlenmesi için giderek daha çok tercih edilmektedir (Zhang ve ark., 2018). Derin öğrenmede, bir bilgisayar modeli, sınıflandırma görevlerini doğrudan görüntülerden, metinlerden veya sesten gerçekleştirmeyi öğrenir. Derin öğrenme modelleri, bazen insan seviyesinde performansı aşan, son teknoloji hassasiyetini elde edebilir. Modeller, çok sayıda katmanı içeren büyük bir etiketlenmiş veri seti ve sinir ağı mimarileri kullanılarak eğitilir (MathWorks, 2018b).

4.3.1. Makine öğrenmesi ile derin öğrenme aralarındaki fark

Derin öğrenme, özel bir makine öğrenme biçimidir. Bir makine öğrenme işlem akışı (pipeline) , görüntülerden manuel olarak çıkarılmış olan ilgili özellikler ile başlar. Özellikler daha sonra görüntüdeki nesneleri kategorilere ayıran bir model oluşturmak için kullanılır. Derin öğrenme'nin işlem akışıyla, ilgili özellikler otomatik olarak görüntülerden çıkarılır. Buna ek olarak, derin öğrenme bir ağa ham verilerin verildiği ve sınıflandırma gibi bir görevi yerine getirdiği "uçtan uca öğrenmeyi" gerçekleştirir ve bunu otomatik olarak nasıl yapacağını öğrenir. Diğer bir önemli farklılık ise derin öğrenme algoritmaları veri ile ölçeklendirilirken, sığ öğrenme yakınsamadır. Sığ

öğrenme, daha fazla eğitim verileri ağa eklendiğinde belirli bir performans düzeyine sahip olan makine öğrenme yöntemlerini ifade etmektedir. Derin öğrenme ağlarının önemli bir avantajı, verilerin boyutu arttıkça genellikle geliştirilmeye devam edilebilmesidir (MathWorks, 2018b).



Şekil 4.3. Araçları kategorilere ayırmak için Derin öğrenme yaklaşımının Makine öğrenmesi yaklaşımı ile karşılaştırılması (Karunakaran, 2018a)

Şekil 4.3'te gösterildiği gibi, makine öğrenmesi ile derin öğrenme modelleri arasındaki farklardan biri, özellik çıkarma alanıdır. Özellik çıkarma, makine öğrenmede insan tarafından yapılırken, derin öğrenme modelleri otomatik olarak yapmaktadır (Karunakaran, 2018a).

4.3.2. Makine öğrenmesi ve derin öğrenme arasında seçim

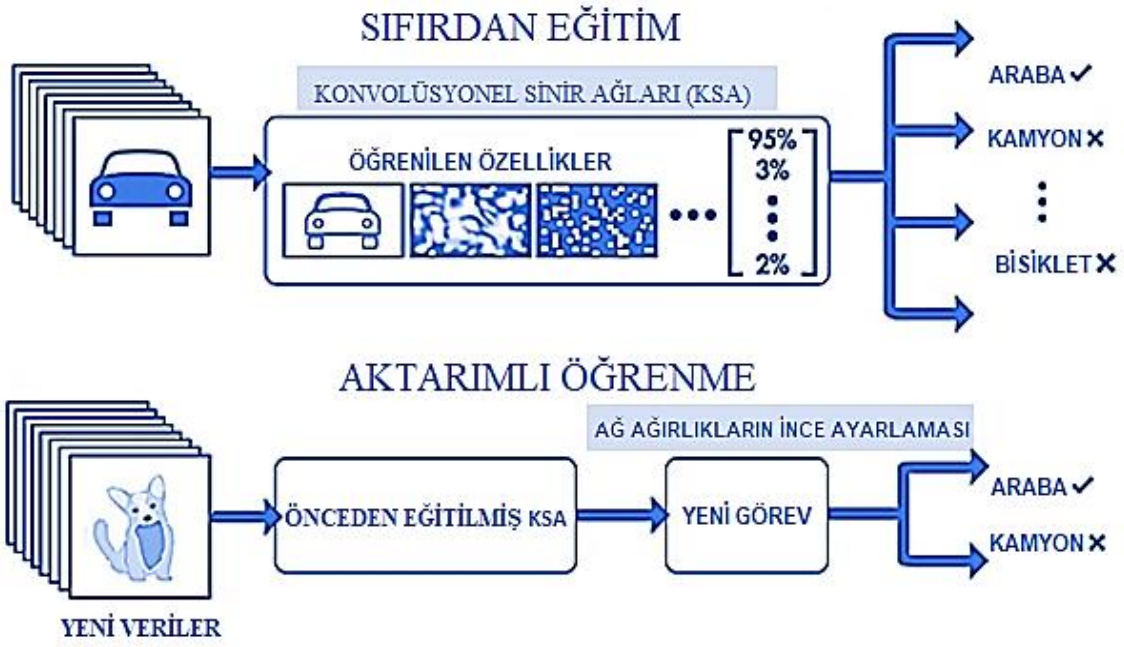
Makine öğrenmesi, uygulamalarda, işlenildiği veri boyutuna ve çözmek istendiği sorunun türüne göre seçim yapılabileceği çeşitli teknikler ve modeller sunar. Başarılı bir derin öğrenme uygulaması, verileri hızlı bir şekilde işlemek için modeli, Grafik İşleme Birimi (GPU) veya grafik işlem ünitelerini eğitmek için çok büyük miktarda veri (binlerce resim) gerektirmektedir (MathWorks, 2018b).

Makine öğrenmesi ve derin öğrenme arasında seçim yaparken, yüksek performanslı bir GPU ve çok sayıda etiketli veri olup olmadığını göz önünde bulundurulmalıdır. Eğer bunlardan herhangi biri mevcut değilse, derin öğrenme yerine makine öğrenmesi kullanmak daha avantajlı olabilmektedir. Derin öğrenme genellikle daha karmaşıktır, bu sebepten dolayı güvenilir sonuçlar elde etmek için en az birkaç bin veriye ihtiyaç duymaktadır. Yüksek performanslı bir GPU'ya sahip olmak, modelin tüm

bu görüntüleri (verileri) analiz etmek için daha az zaman alacağı anlamına gelir (MathWorks, 2018b).

4.3.3. Derin öğrenme modellerinin oluşturulması ve eğitilmesi

Nesne sınıflandırması yapmak için insanların derin öğrenmeyi kullandığı en yaygın iki tane yaklaşım vardır, bu yaklaşımların genel mantığı şekil 4.4'te gösterilmiştir:



Şekil 4.4. Derin öğrenme modellerinin eğitilme yaklaşımları (Willingham, 2017)

4.3.3.1. Sıfırdan eğitim (Training from scratch)

Derin bir ağı sıfırdan eğitmek için çok büyük etiketli veri seti toplamak ve özellikleri ve modeli öğrenecek bir ağ mimarisi tasarlanmak gerekmektedir. Bu yeni uygulamalar veya çok sayıda çıktı kategorisine sahip uygulamalar için iyidir. Bu çok yaygın bir yaklaşım değildir, çünkü büyük miktarda veri ve öğrenme oranıyla, bu ağları tipik olarak eğitmek günler hatta haftalar sürebilmektedir (MathWorks, 2018b).

4.3.3.2. Aktarımlı öğrenme (Transfer learning)

Çoğu derin öğrenme uygulaması, önceden eğitilmiş bir modelin ince ayarını içeren bir süreç olan aktarımlı öğrenme yaklaşımını kullanmaktadır. AlexNet veya GoogLeNet gibi mevcut bir ağ ile başlar ve önceden bilinmeyen sınıfları içeren yeni

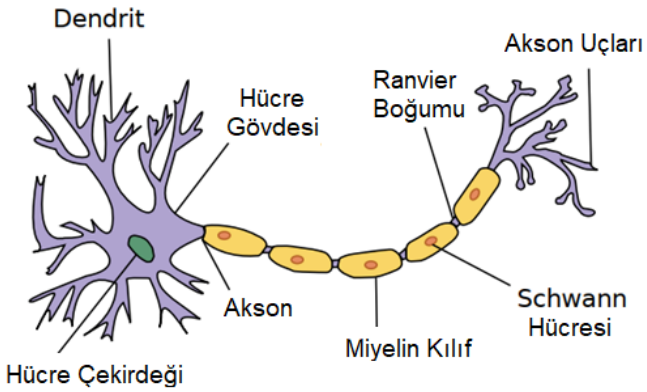
verilerle beslenilir. Ağa bazı ince ayarlamalar yaptıktan sonra, artık 1000 farklı nesne yerine yalnızca belirli bir kaç nesneyi kategorize etmek gibi yeni bir görev gerçekleştirebilir. Bu aynı zamanda çok daha az veriye (milyonlarca görüntü yerine binlerce görüntü işleme) ihtiyaç duyma avantajına sahiptir, bu nedenle hesaplama süresi dakikalara veya saatlere düşer. Aktarımlı öğrenme, önceden var olan ağın iç yapısına bir arayüz gerektirir, bu yüzden yeni işlem için parametreleri değiştirilebilir ve geliştirilebilir (MathWorks, 2018b).

4.3.4. Derin sinir ağının oluşturulması

Derin öğrenmenin esas yapısı, yapay sinir ağındaki tek bir perseptrondur. Açıkçası, YSA ismini Neuroscience'dan almıştır. Bir ağ yapısının nasıl icat edildiğini daha iyi anlamak için, bir nöronun temel yapısını göz önünde bulundurmaya gerekmektedir.

4.3.4.1. Biyolojik nöron

Şekil 4.5 gerçek bir nöronu (biyolojik nöronu) temsil etmektedir, analoji amacıyla dendritleri (giriş uçları), çekirdeği (CPU), aksonu (çıkış teli) ve diğer nöronlara sinapslar yoluyla bağlanan akson uçları (çıkış terminalleri) dikkate alınması yeterlidir (Zhang ve ark., 2018).



Şekil 4.5. Biyolojik nöron (Zhang ve ark., 2018)

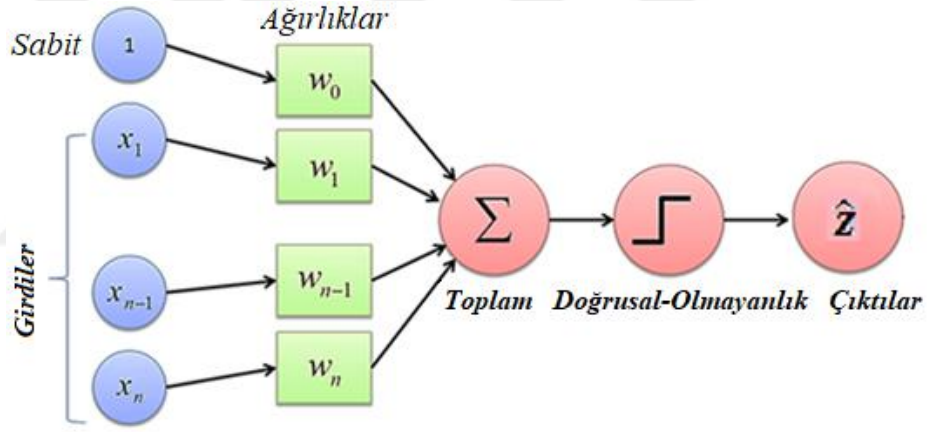
Diğer nöronlardan (veya retina gibi çevresel sensörlerden) gelen $\boldsymbol{x_i}$ bilgisi, dendritler tarafından girdi olarak alınır. Girdilere verilecek cevapları belirleyen $\boldsymbol{w_i}$ sinaptik ağırlıklarıyla bu $\boldsymbol{x}$ bilgileri ağırlıklandırılır (örneğin, $\boldsymbol{x_i w_i}$ yoluyla aktivasyon veya inhibisyon). Sonra bu bilgilerin toplamına $\boldsymbol{b}$ eğim (bias) ağırlığı eklenir.

$$z = b + \sum_{i=1}^{n} x_i w_i \tag{4.1}$$

Bunların hepsi, denklem 4.1'e gösterildiği gibi çekirdekte toplanır ve bu bilgiler daha sonra, denklem 4.2'de gösterildiği gibi tipik olarak $\boldsymbol{g(z)}$ bazı doğrusal olmayan işlemlerden sonra, z aksonuna daha fazla işlem için gönderilir. Oradan ya hedefine (örneğin bir kas) ulaşır ya da dendritleri vasıtasıyla başka bir nörona gönderilir (Zhang ve ark., 2018).

4.3.4.2. Tek katmanlı perseptron

Perseptron, tek katmanlı bir sinir ağıdır ve çok katmanlı perseptron YSA oluşturur. Perseptron doğrusal bir sınıflandırıcıdır. Ayrıca, denetimli öğrenmede de kullanılır. Girdi verilerinin sınıflandırılmasına yardımcı olur (Sharma, 2017).



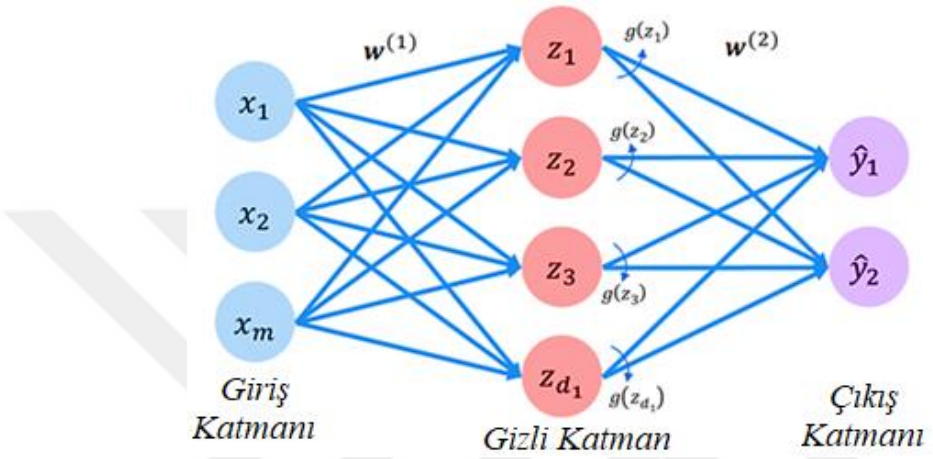
Şekil 4.6. Perseptron (Sharma, 2017)

$$\hat{\boldsymbol{z}} = \boldsymbol{g}(b + \sum_{i=1}^{n} x_i w_i) \tag{4.2}$$

Şekil 4.6'de gösterildiği gibi bir perseptron aynen bir biyolojik nöron gibi verileri alır ve öğrenir. İlk önce giriş katmanındaki perceptronlara beslenen girdiler (x_i) alınır, ağırlıkları (w_i) ile çarpılır ve toplamı hesaplanır. Denklem 4.1'de gösterildiği gibi, elde edilen toplama eğim (bias) ağırlığı ($b = 1 * w_0$) eklenir. Bu, her bir perseptronun çıkış fonksiyonu sayı grafiğinde yukarı, aşağı, sola ve sağa hareket ettirmeyi mümkün kılan teknik bir adımdır. Toplamı doğrusal olmayan aktivasyon fonksiyonundan geçirilir. Aktivasyon fonksiyonunun sonucu çıktı olarak elde edilir (Sharma, 2017).

4.3.4.3. Yapay sinir ağı

YSA preseptronlardaki nöronların ve katmanların arttırılmasıyla oluşturulur. Henüz derin ağ değildir. Şekil 4.7'de görüldüğü gibi ortadaki düğümler gizlenmiştir, bilgiler girdi katmandan gizli katmana ve gizli katmandan çıktı katmanına geçerken bir dönüşüm oluşur. Burada giriş katmanından gizli katmanına $w^{(1)}$ ve gizli katmandan çıkış katmanına $w^{(2)}$ olmak üzere iki tane ağırlık matrisine ihtiyaç duyulur (Amini, 2019).



Şekil 4.7. Tek Katmanlı Yapay Sinir Ağı (Amini, 2019)

Daha önce belirtildiği gibi, şekil 4.7'de, en soldaki katman giriş katmanı olarak adlandırılır ve katman içindeki nöronlara giriş nöronları denir. En sağdaki veya çıkış katmanı, çıkış nöronlarını veya bu şekilde olduğu gibi iki çıkış nöronunu içerir. Orta katmana gizli katman denir, çünkü bu katmandaki nöronlar ne girdi ne de çıktıdır. "Gizli" terimi, "giriş veya çıkış değil" anlamına gelmektedir. Şekil 4.7'deki ağın yalnızca tek bir gizli katmanı vardır, ancak bazı ağların birden çok gizli katmanları olabilir (Nielsen, 2019).

$$\boldsymbol{z_i} = w_{0,i}^{(1)} + \sum_{j=1}^{m} x_i w_{j,i}^{(1)} \tag{4.3}$$

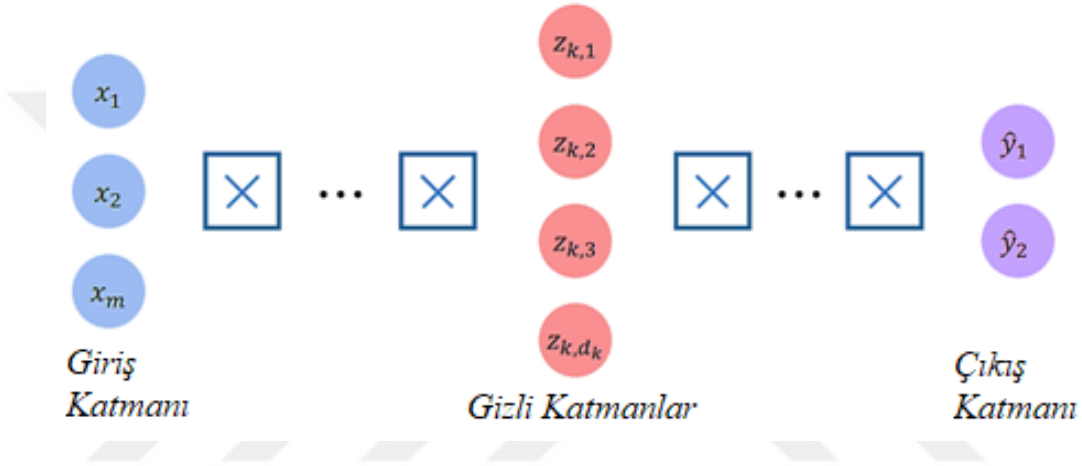
$$\boldsymbol{\hat{y}_i} = g(w_{0,i}^{(2)} + \sum_{j=1}^{d_1} z_i w_{j,i}^{(2)}\ ) \tag{4.4}$$

Bir YSA girilen verileri öğrenmek için ilk olarak, denkelm 4.3'te gösterildiği gibi girdi katmanının değişkenlerinin ağırlıklı toplamı, karşılık gelen ağırlık matrisi ($w^{(1)}$) kullanılarak ve ilgili bais eklenmesiyle gizli katmanda hesaplamalıdır ve sonucu doğrusal olmaması için bazı aktivasyon fonksiyonlardan geçirir. Denklem 4.4 gösterildiği gibi gizli

katmanın son çıktısı bir sonraki katmana (bu durumda çıkış katmana) giriş olarak kullanılacak, aynı işlemler hesaplanacaktır ve son sonuç elde edilecektir (Amini, 2019).

4.3.4.4. Derin sinir ağları

Derin sinir ağları YSA'nın gizli katmanlarının sayısının arttırılmasıyla oluşurlar (bkz. Şekil 4.8). Çoğu derin öğrenme yöntemleri, sinir ağ mimarilerini kullanır, bu nedenle derin öğrenme modellerine genellikle derin sinir ağları denir. "Derin" terimi sinir ağındaki gizli katmanların sayısını ifade eder.



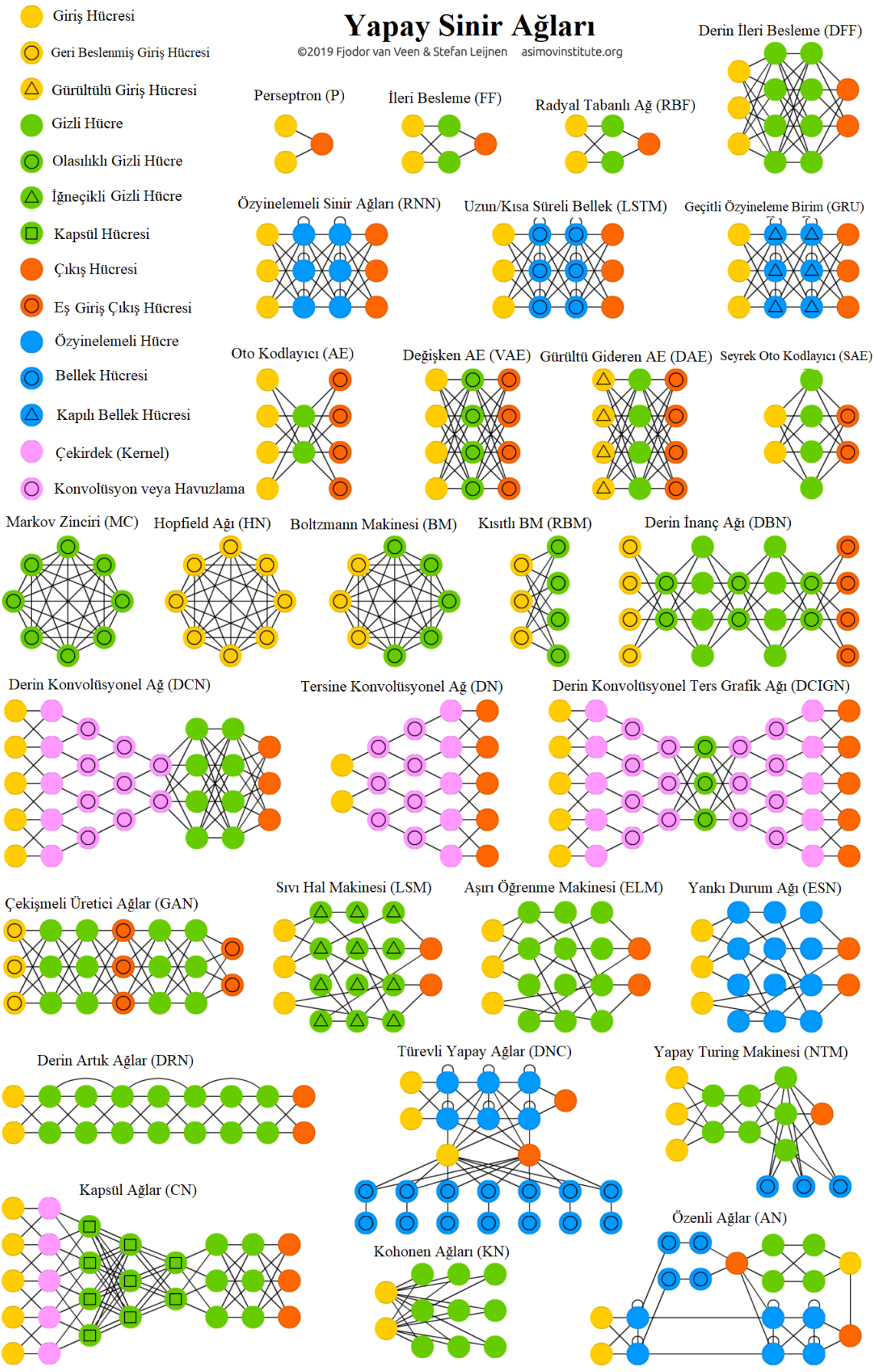
Şekil 4.8. Derin Sinir Ağı (Amini, 2019)

Derin Sinir Ağı'nda, daha derin modeller oluşturmak için gizli katman sayısı arttırılır. Denklem 4.5'te gösterildiği gibi ağın çıktısı daha derin bir şekilde hesaplanır ve ağırlıklı toplamları tekrar tekrar uygulayarak, aktivasyon fonksiyonlarıyla bir çok defa hesaplanır (Amini, 2019).

$$\boldsymbol{z_{k,i}} = w_{0,i}^{(k)} + \sum_{j=1}^{d_{k-1}} g(z_{k-1,j}) w_{j,i}^{(k)} \tag{4.5}$$

Geleneksel sinir ağları sadece 2-3 gizli katman içerirken, derin sinir ağlar yüzden fazla katmana sahip olabilirlar. Derin öğrenme modelleri, elle özellik çıkarma gerekmeden veriyi doğrudan öğrenen büyük etiketli veri setleri ve yapay ağ mimarileri kullanılarak eğitilir (MathWorks, 2018b).

Şekil 4.9'de mevcut olan yaklaşık tüm yapay sinir ağların mimarisi gösterilmiştir.



Şekil 4.9 Mevcut olan bir kaç Yapay Sinir Ağların Mimarisi (Veen, 2016)

5. KONVOLÜSYONEL SİNİR AĞLARI

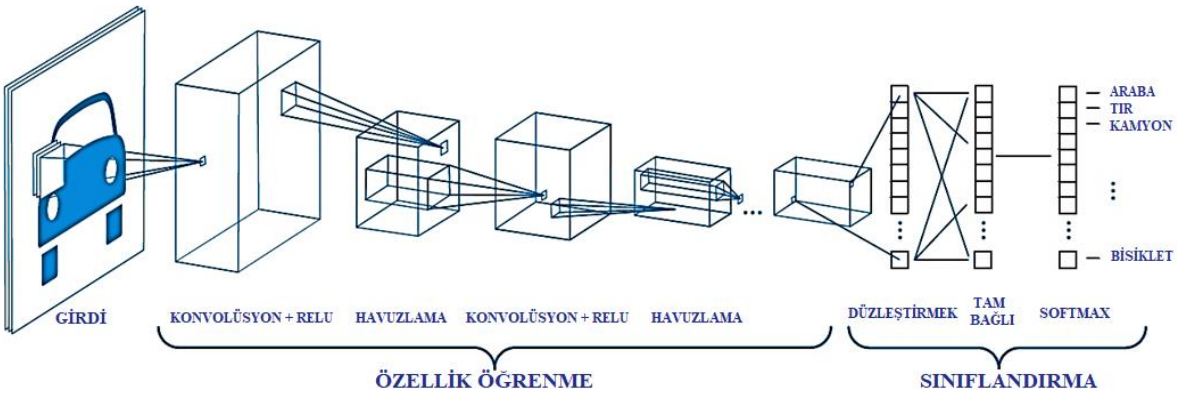
Konvolüsyonel sinir ağları (KSA veya ConvNet), doğrudan görüntülerden, videolardan, metinlerden veya seslerden sınıflandırma işlemlerini gerçekleştirmeyi öğrenebilen bir makine öğrenme türü olan derin öğrenmenin en popüler algoritmalarından birisidir. KSA özellikle nesneleri, yüzleri ve sahneleri tanımak ve görüntülerde kalıp bulmak için kullanışlıdır. Doğrudan görüntü verilerini öğrenir, görüntüleri sınıflandırmak için desenler kullanır ve manuel özellik çıkarma ihtiyacını ortadan kaldırır. Kendi kendini süren araçlar ve yüz tanıma uygulamalar gibi nesne tanıma ve bilgisayar vizyonu gerektiren uygulamalar, büyük ölçüde KSA'ya dayanır. Uygulamalara bağlı olarak, sıfırdan bir KSA oluşturulabilir veya veri setiyle önceden eğitilmiş bir model kullanılabilir (MathWorks, 2018a).

5.1. Konvolüsyonel Sinir Ağlarının Çalışması

Bir KSA, her bir görüntünün farklı özelliklerini algılamayı öğrenen onlarca veya yüzlerce katmana sahip olabilir. Filtreler, farklı çözünürlüklerdeki her eğitim görüntüsüne uygulanır ve her bir resmin konvole edilen çıkışı, sonraki katmana girdi olarak kullanılır. Filtreler, parlaklık ve kenarlar gibi çok basit özellikler olarak başlar ve her bir katmanda nesneyi benzersiz şekilde tanımlayan özelliklerin karmaşıklığını arttırır. KSA, görüntülerin, metinlerin, seslerin ve videoların özelliklerin tanımlamalarını ve sınıflandırılmalarını gerçekleştirebilmektedir (MathWorks, 2018a).

5.2. Konvolüsyonel Sinir Ağların Mimarisi

Diğer sinir ağları gibi, bir KSA'da bir girdi katmanı, bir çıkış katmanı ve birçok gizli katmandan oluşur. Bu katmanlarda, verileri, verilere özgü özelliklerin öğrenme amacı ile değiştiren işlemler gerçekleştirilmektedir. Bir KSA tipik olarak üç katmandan oluşur: konvolüsyon (convolution) katmanı, havuzlama (pooling) katmanı ve tam bağlı (fully connected) katmanı (MathWorks, 2018a).



Şekil 5.1 Birçok konvolüsyon katmanları olan KSA örneği (MathWorks, 2018a)

Şekil 5.1'de birçok konvolüsyon katmanı olan bir KSA örneği gösterilmektedir. Filtreler, her eğitim görüntüsüne farklı çözünürlüklerde uygulanır ve her bir konvule edilen resmin çıktısı, bir sonraki katmana girdi olarak kullanılır (MathWorks, 2018a).

Şekil 5.1'de görüldüğü gibi ağırlıkları ve eğimleri (bias) öğrenilebilir olan KSA içinde birçok nöron vardır. Her bir nöron, girdileri kabul eder, bir kaç işlemlerini gerçekleştirir ve sonrası doğrusal olmayan fonksiyonu verir. Ve tam bağlı katmanda bir kayıp fonksiyonu (ör. SVM / Softmax) mevcuttur, ve normal bir YSA'nın öğrenmesi için kullanılan tüm işlemler ve püf noktaları burada da geçerlidir (Karunakaran, 2018b).

5.2.1. Özellik öğrenme

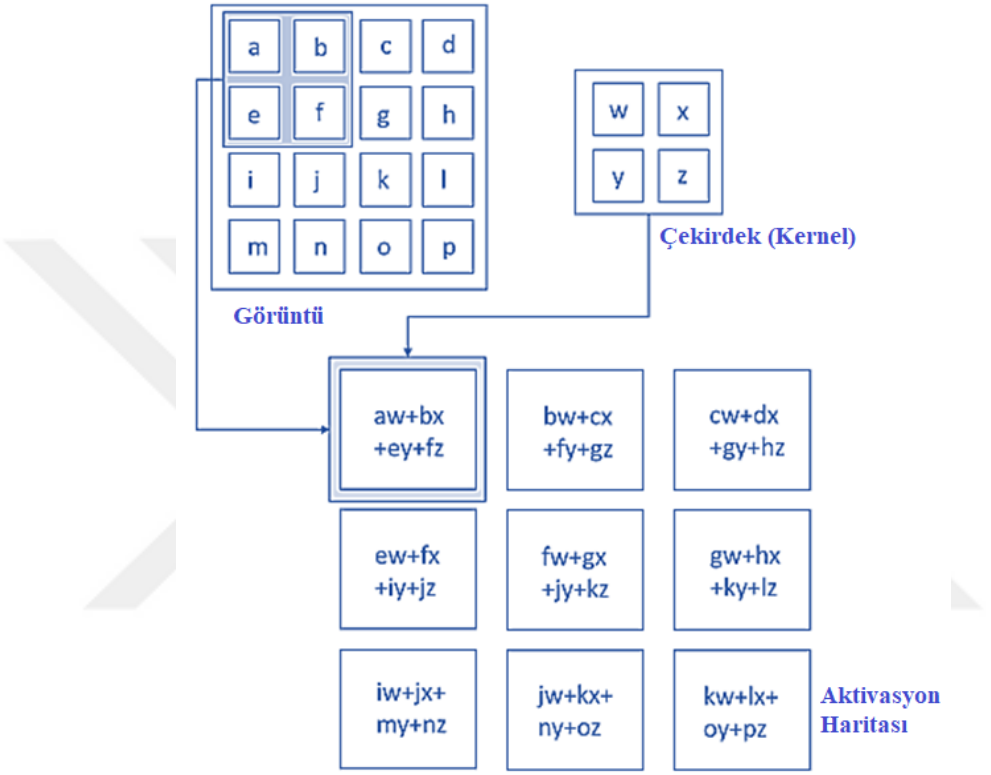
Her bir katman farklı özellikleri tanımlamayı öğrenirken, bu işlemler onlarca veya yüzlerce katman üzerinde tekrarlanmaktadır (MathWorks, 2018a):

5.2.1.1. Konvolüsyon (Convolution) katmanı

Girdi görüntüleri, her biri görüntülerden belirli özellikleri etkinleştiren bir dizi iç içe filtreden geçirilir (MathWorks, 2018a). Konvolüsyon katmanlarının amacı sunulan resimlerdeki özellikleri tespit etmektir. Her biri belirli özellikleri tanıyan çoklu özellikli haritalardan oluşur.

Konvolüsyon katmanı, KSA'nın esas yapısıdır. Ağın hesaplamalı yükünün ana bölümünü taşır. Bu katman, iki matris arasında bir nokta çarpımı gerçekleştirir, burada ilk matris, kernel olarak bilinen öğrenilebilir parametreler kümesidir ve diğer matris, gönderilen alanın kısıtlı kısmıdır (Mishra, 2019).

Şekil 5.2'de, kernel çevirme olmadan 2 boyutlu konvolüsyon örneği gösterilmiştir. Çıktı yalnızca kernelin tamamen görüntüde yer aldığı, bazı bağlamlarda "geçerli" konvolüsyon adı verilen konumlarla sınırlıdır. Şekil 5.2'de gösterildiği gibi çıkış tensörünün sol üst öğesinin kernelin giriş tensörünün karşılık gelen üst sol bölgesine uygulanarak nasıl oluşturulduğunu göstermek için oklu kutular çizilmiştir (Goodfellow ve ark., 2016).



Şekil 5.2. Konvolüsyon İşlemi (Goodfellow ve ark., 2016)

Çekirdek (Kernel), bir görüntüden uzamsal olarak daha küçüktür, ancak daha derindir. Eğer görüntü üç (RGB) kanallardan oluşuyorsa, kernel yüksekliğinin ve genişliğinin uzamsal olarak küçük olacağı, ancak derinliğinin üç kanalda da uzanacağı anlamına gelir. İleri geçiş sırasında, kernel, alıcı bölgenin görüntü temsilini üreten görüntünün yüksekliği ve genişliği boyunca kayar. Böylece, görüntünün her uzamsal pozisyonunda kernelin tepkisini/cevabını veren görüntünün iki boyutlu gösterimi olan aktivasyon haritasını üretir. Kernelin kayan büyüklüğü bir adım (stride) olarak adlandırılır (Mishra, 2019).

5.2.1.2. Doğrultulmuş doğrusal birim (ReLU)

ReLU fonksiyonu çok basit bir doğrusal olmayan dönüşüm sağlar. x verisi göz önüne alındığında, bu fonksiyon denklem 5.1'de tanımlanmıştır (Zhang ve ark., 2018).

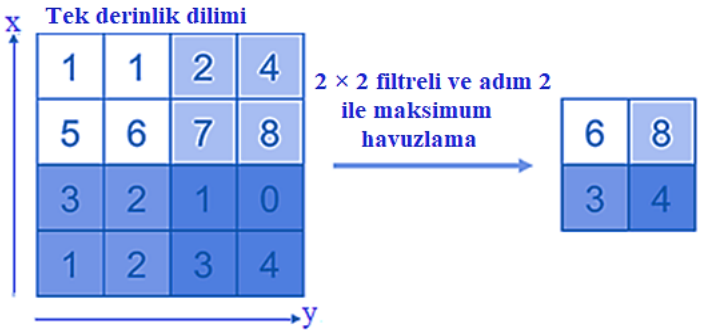
$$\boldsymbol{ReLu}(x) = max(x, 0) \tag{5.1}$$

Negatif değerleri sıfıra eşleyerek ve pozitif değerleri koruyarak, ağın daha hızlı ve etkili eğitilmesini sağlamaktadır. Bu bazen aktivasyon fonksiyonu olarak anılır, çünkü sadece aktive edilen özellikler bir sonraki katmana taşınır (MathWorks, 2018a). Aktivasyon işlemleri, düğüme girdiler verildiğinde, etkinleştirme işlemi gerçek çıktıya karar veren işlemlerdir (Karunakaran, 2018b).

5.2.1.3. Havuzlama (Pooling) katmanı

Havuzlama katmanı doğrusal olmayan verilerin çıktılarını basitleştirir, ağın öğrenmesi gereken parametrelerin sayısını azaltır (MathWorks, 2018a). Havuzlama katmanı, ağdaki hesaplamaların sayısını ve hesaplama miktarını azaltmak için gösterimin mekansal boyutunu azaltmaya yardımcı olur. Bu ayrıca aşırı uyumu kontrol eder (Karunakaran, 2018b).

Havuzlama katmanı, yakınlardaki çıktıların bir özet istatistiğini türetmek suretiyle belirli yerlerde ağ çıktısının yerini alır. Bu, gereken hesaplama ve ağırlık miktarını azaltan gösterimin uzamsal boyutunu azaltmada yardımcı olur. Havuzlama işlemi, temsilin her diliminde ayrı ayrı gerçekleştirilir (Mishra, 2019). 2×2



Şekil 5.3. Havuzlama İşlemi (Mishra, 2019)

Dikdörtgen alanın ortalaması, dikdörtgen alanın L2 normu ve merkez pikselden olan mesafeye göre ağırlıklı ortalama gibi çeşitli havuzlama fonksiyonları bulunmaktadır.

Bununla birlikte, en popüler işlem, şekil 5.3'te gösterildiği gibi alandan maksimum verimi veren maksimum havuzlamadır (max pooling) (Mishra, 2019).

5.2.1.4. Tam bağlı (Fully connected) katmanı

Konvolüsyonel Sinir Ağı'nın sonundaki katman tam bağlı nöron katmanıdır. Tam bağlı bir katmandaki nöronlar, normal Yapay Sinir Ağlarında görüldüğü gibi, önceki katmandaki tüm aktivasyonlarla tam bağlantılara sahiptir ve benzer şekilde çalışır. Bu nedenle, her zamanki gibi bir matris çarpımı ve ardından bir eğim etkisi ile hesaplanabilir. Tam bağlı katmanı giriş ile çıkış arasındaki gösterimi eşlemeye yardımcı olur (Mishra, 2019).

5.2.2. Sınıflandırma katmanlar

Birçok katmandaki özellikleri öğrendikten sonra, KSA mimarisi sınıflandırma işlemine geçer. Sonuncu katman, K boyutlu bir vektörü veren tam bağlı bir katmandır. Burada K ağın tahmin edeceği sınıfların sayısıdır. Bu vektör sınıflandırılmış herhangi bir verinin her sınıfı için olasılıkları içerir. KSA mimarisinin son katmanı, sınıflandırma çıktısını sağlamak için softmax gibi bir sınıflandırma katmanı kullanır (MathWorks, 2018a).

5.3. Derin KSA Öğrenmesi

Geleneksel makine öğrenme algoritmaları tipik olarak verilerden çıkarılması gereken kural veya özellik setini tanımlar, genellikle bunlar elle işlenen özelliklerdir ve pratikte son derece kırılgan olma eğilimindedir (Amini, 2019).

Derin öğrenmenin temel fikri ya da temel kavrayışı, bu özelliklerin elle işlenmeden, doğrudan veriden öğrenilebilmesidir. Özellikleri öğrenmek için derin bir öğrenme algoritması veya KSA, önce düşük seviye özelliklerini tespit eder, orta seviye özelliklerini tespit etmek için bu düşük seviye özelliklerini birleştirip kullanır ve son olarak daha yüksek seviye özelliklerini tespit etmek / öğrenmek için orta seviye özellikleri birleştirip kullanır (Amini, 2019).

Örneğin şekil 5.4'te gösterildiği gibi, bir yüz tanıma işlemini öğrenmek için bir derin öğrenme model veya algoritma önce resimdeki kenarları algılamaya/ öğrenmeye

çalışır, bu kenarları göz, burun ve ağız gibi şeyleri algılamak/ öğrenmek için birleştirir ve daha sonra bu özellikleri, daha yüksek seviyeli yapıları (yüz) öğrenmek için oluşturur. Bunların hepsi hiyerarşik bir şekilde gerçekleştirilir (Amini, 2019).



Şekil 5.4. Derin KSA'nın verileri öğrenmesi (Amini, 2019)

6. MATERYAL VE YÖNTEM

Bu çalışmada aktarımlı öğrenmeyi (4.3.3.2) kullanılarak üzüm meyvesi ve üzüm yaprağı hastalıklarının tespit edilmesi için iki derin KSA modeli kullanılmıştır. Genelde (4.3.3)'de belirtildiği gibi, derin bir sinir ağını eğitmek için iki seçenek vardır. Bunlardan biri, sıfırdan derin bir sinir ağı oluşturmak ve onu istenen veri kümesi üzerinde sıfırdan eğitmektir. Diğeri ise bu çalışmada da kullanılmış olan önceden eğitilmiş bir ağı seçip, yeni işlem için parametreleri değiştirmek ve geliştirme işleminden sonra istenen veri seti üzerinde yeniden eğitmektir.

Derin sinir ağlarının milyonlarca parametreyle eğitilmesi, optimizasyonunun uygun şekilde sınırlandırılması için büyük miktarda veri gerekmektedir. Ayrıca bu ağların eğitilmesi için yüksek hesaplama kaynaklarına ihtiyaç vardır. Bu durum araştırmacıları önceden eğitilmiş olan ağların bilgilerini kullanarak çalışmalarını yapmalarına yönlendirmiştir (Mehdipour-Ghazia ve ark., 2017).

Yeterli donanımsal altyapının sağlanamaması ve bu ağı eğitmek için büyük veri kümesinin olmamasından dolayı, bu çalışma için transfer öğrenme yöntemi kullanılarak önceden eğitilmiş olan iki tane derin KSA modeli (SSD_MobileNet v1 ve Faster-R-CNN Inception v2) seçilmiştir. Bu iki model COCO-Tensorflow'un (Rathod ve ark., 2019) önceden eğitilmiş olan nesne tanıma modellerindendir.

6.1. Sistem Konfigürasyonları

Derin öğrenme modelleri çok güçlü GPU ve Birleştirilmiş Aygıt Hesaplama Mimarisi (CUDA) gerektirir (Rangarajan ve ark., 2018). Bu çalışma CUDA 9.0, 2G video belleği ve 4G Rasgele Erişim Belleği (RAM) ile NVIDIA GeForce 740M (sürüm 417.35) GPU özelliklerine sahip bir makinede gerçekleştirilmiştir. Ayrıca uygulama, Python programlama dilinde, Windows 10 işletim sistemi altında ve Tensorflow kütüphanesi kullanılarak geliştirilmiştir. TensorFlow Google tarafından geliştirilmiş açık kaynaklı bir kütüphanedir. TensorFlow görüntülerdeki nesneleri otomatik olarak algılamak ve sınıflandırmak için bir YSA'nın geliştirilmesi ve eğitilmesi amacıyla kullanılır.

6.2. SSD MobileNet

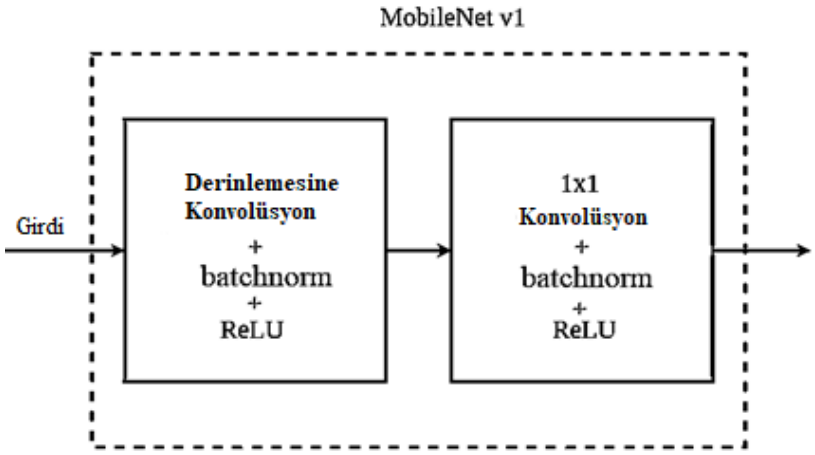
MobileNet sınıflandırıcısı ve Tek Çekim Çoklu Kutu (SSD) modelleri birleştirilerek oluşturulmuş ve COCO (Common Objects in Context) veri seti üzerinde önceden eğitilmiş bir mimaridir. COCO veri kümesi, 328 bin görüntü ve 2,5 milyon etiketli örnekten oluşmakta olup 91 nesne kategorisini içermektedir (Lin ve ark., 2014).

6.2.1. MobileNet

Bu model, mobil ve gömülü görme uygulamaları için tanıtılmış olup derinlik açısından ayrılabilir konvolüsyonlara dayanmaktadır (Howard ve ark., 2017).

MobileNet, diğer birçok popüler modelden çok daha küçük boyutta ve performans açısından daha hızlı olan, basit bir derin konvolüsyonel sinir ağı modelidir. Derinlik açısından ayrılabilir konvolüsyonlarda her bir girdi kanalına tek bir filtre uygulanır. Ardından nokta bazlı konvolüsyonda 1×1 konvolüsyonlar kullanılarak derinlik bazlı katmanların çıktısının doğrusal kombinasyonu oluşturulur (Howard ve ark., 2017).

Derinlik açısından ayrılabilir katmanlar, tipik konvolüsyon katmanlarının işlemini neredeyse taklit eder, ancak çok daha hızlı ve küçük bir farkla (tipik konvolüsyon, çıktı özelliklerine hem filtre uygular hem de birleştirir, ancak bu işlem derinlik açısından ayrılabilir konvolüsyonda iki katmana ayrılır, filtreleme için ayrı bir katman ve birleştirme için ayrı bir katman) işlem gerçekleştirilir. Bu yaklaşım modelin boyutunu en aza indirir ve hesaplama güç taleplerini azaltır. Çıktıyı sınıflandırmak için softmax katmanını besleyen tam bağlantılı katman dışında, tüm katmanlardan sonra bir batchnorm ve ReLU doğrusal olmayan aktivasyon katmanı gelir. MobileNet, derinlikli ve noktasal konvolüsyonlar haricinde 28 katmana sahiptir (Howard ve ark., 2017). MobileNet'in genel mimarisi Şekil 6.1'de görsel olarak sunulmuştur.



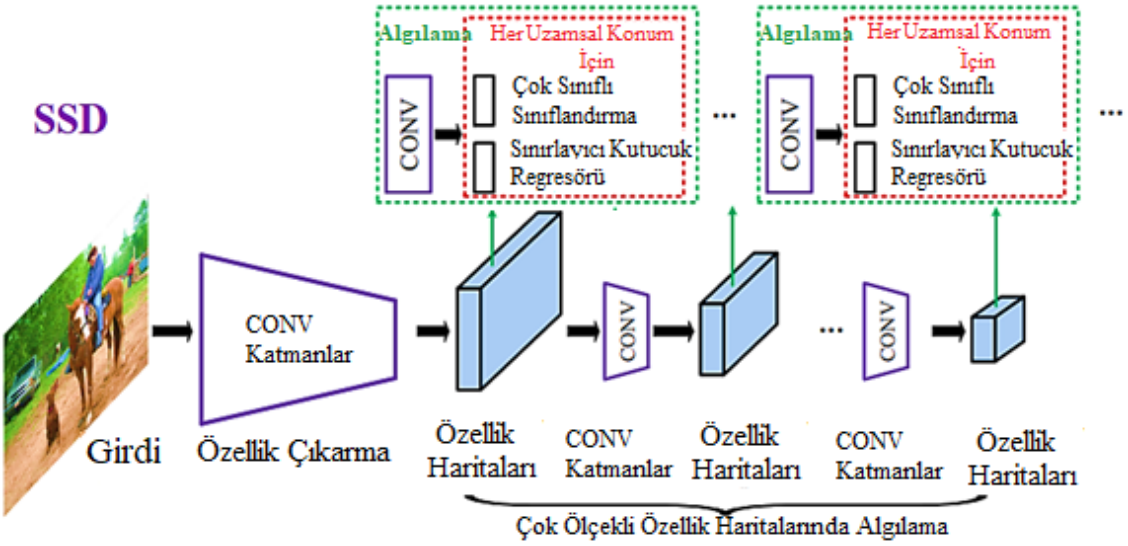
Şekil 6.1. MobileNet'in Genel Mimarisi

6.2.2. SSD (single shot multibox detector)

SSD yöntemi W. Liu ve arkadaşlarının 2016'da yaptığı çalışmada nesne tespit işlemleri için önerilmiştir. SSD'de yüksek algılama kalitesini koruyup kaliteli algılama hızı elde etmek için daha Fast R-CNN, YOLO ve çok ölçekli KSA gibi birçok ağın arkasındaki fikirler dikkate alınmıştır (W. Liu ve ark., 2016, L. Liu ve ark., 2018).

SSD'de, her sınırlayıcı kutucuğun içindeki her nesne kategorisinin önceliği için bir puanlama tablosu oluşturulur, ardından nesnenin şekli ile daha iyi eşleşmesi için kutucuğa bir ayarlama aşaması (son tespit) uygulanır. KSA tamamen konvolüsyonludur ve temel ağı VGG-16 ağıdır. Ardından kademeli olarak boyut azaltan birçok yardımcı konvolüsyon katmanı bulunur (L. Liu ve ark., 2018).

Küçük nesneleri algılamak için, SSD daha yüksek çözünürlüklü daha sığ katmanları kullanır. Farklı boyutlardaki nesneler için SSD, çoklu konvolüsyon özellik haritalarında çalışarak çoklu ölçek algılamasını gerçekleştirir. Her biri uygun boyutta olan kutucukları sınırlamak için kategori puanlarını ve kutucuk konumunu (ofsetlerini) tahmin eder (L. Liu ve ark., 2018). Şekil 6.2'de genel nesne algılama işlem için SSD'nin üst düzey diyagramı gösterilmektedir.



Şekil 6.2. Genel Nesne Algılama İşlem için SSD'nin Üst Düzey Diyagramı (L. Liu ve ark., 2018)

6.3. Faster-RCNN Inception V2

Inception modeli Faster R-CNN model ile bileştirilmiş ve COCO veri seti üzerinde önceden eğitilmiştir.

6.3.1. Faster–RCNN

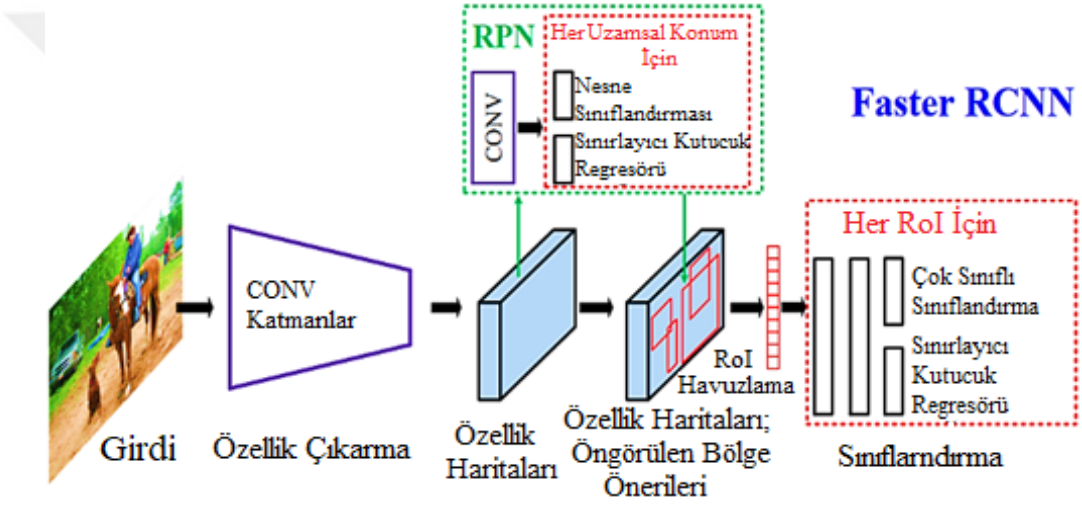
Bu mimari, (Ren ve ark., 2015)'de, nesne konumlarını tahmin etmek için bölge önerme algoritmalarına bağlı olan, nesne algılama amaçları için tanımlanmış bir bölge önerme ağıdır (RPN). RPN'de bölge önerme işlemi ile birlikte bu bölgelerin Fast R-CNN ile sınıflandırılmasını sağlayan tek bir ağ kullanılmıştır.

Faster R-CNN'de RPN, Fast R-CNN ile görüntünün tam konvolüsyon özelliklerini paylaşır, bu da neredeyse masrafsız bölge önerileri sağlar. RPN, eşzamanlı yani aynı anda nesnenin sınırlarını ve puanlarını tahmin eden tam konvolüsyonlu bir ağdır ve bu Faster R-CNN'i tamamen KSA tabanlı bir model yapmaktadır. Uçtan uca eğitildikten sonra RPN'ler tarafından üretilen yüksek kaliteli bölge önerileri, bölge tespiti için Fast R-CNN tarafından kullanılır (Ren ve ark., 2015).

Bir RPN, herhangi bir boyuta sahip olan bir girdi görüntüsünü alır ve her biri için çıktı olarak nesne puanına sahip bir dizi dikdörtgen bölgeler önerir. İlk $n \times n$ referans kutucukları (her kayan pencere), her konvolüsyon özellik harita konumunda farklı

ölçeklere ve en boy oranlarına sahip olacak şekilde RPN tarafından oluşturulur (Ren ve ark., 2015).

Her kayan pencere, 2 tane tam bağlı olan katmanları (kutucuk sınıflandırma katmanı ve bir kutucuk regresyon katmanı) besleyen daha düşük boyutlu bir vektöre (Zeiler & Fergus modeli için 256 ve VGG için 512 gibi) eşleştirilir. ReLU'lar $n \times n$ konvolüsyon katmanının çıktısına uygulanır. Faster R-CNN, çok derin VGG16 modeli ile her görüntü için 300 öneri bölgesi kullanarak PASCAL VOC 2007'de en gelişmiş nesne algılama doğruluğunu elde ederek, bir GPU'da 5fps'de (tüm adımlar dahil) test yapabilir (Ren ve ark., 2015). Genel nesne algılama için Faster R-CNN'nin üst seviye diyagramı Şekil 6.3'te görsel olarak sunulmuştur.



Şekil 6.3. Genel Nesne Algılama İşlemi için Faster R-CNN'nin Üst Düzey Diyagramı (L. Liu ve ark., 2018)

6.3.2. Inception v2

Inception modeli, çok yüksek işlem gücü olan bilgisayarlar kullanılarak ImageNet veri seti üzerinde önceden eğitilmiş milyonlarca parametreye sahip olan bir derin konvolüsyonel sinir ağıdır ve bu modelin eğitilmesi haftalarca sürebilir. Inception modelinin ilk sürümü (Inception-v1) (Szegedy ve ark., 2015)'de GoogleLeNet adıyla tanıtılmıştır. Daha sonra, toplu iş (batch) normalizasyon tekniği ve diğer farklı yöntemler kullanılarak Inception'in mimarisi (Szegedy ve ark., 2016)'da geliştirilmiş ve yeni sürüm (Inception -v2) ortaya çıkmıştır.

Hesaplama hızını arttırmak için çarpanlarına ayırma (faktorization) tekniğine dayanarak 7×7 konvolüsyonlar üç tane 3×3 konvolüsyona ayrılmıştır. Ağın başlangıç

(inception) bölümünde 3 geleneksel inception modülü vardır ve grid azaltma tekniği kullanılarak, 288 filtreli 35×35 grid, her biri 768 filtreli 17×17 grid'e indirgenmiştir. Ardından ayrılmış olan (faktorize edilmiş) inception modüllerinin 5 örneği gelmektedir. Grid azaltma tekniği kullanılarak $8 \times 8 \times 1280$ grid'e düşürülmüştür. En merkezi 8×8 seviyesinde iki Inception modülü vardır ve her desen (tile) için 2048 boyutunda bir çıktı filtre bankası birleştirilmiştir (Szegedy ve ark., 2016).

6.4. Veri Büyütme (Data Augmentation)

Veri büyütme teknolojisi, benzer fakat farklı eğitim örnekleri üretmek için eğitim görüntülerinde bir dizi rastgele değişiklik yaparak eğitim veri kümeleri ölçeğini genişletir. Örneğin, görüntüler farklı şekillerde kesilir, böylece ilgilenilen nesneler farklı konumlarda bulunabilir ve bu da modelin nesnelerin göründüğü konuma bağımlılığını azaltır. Modelin renge duyarlılığını azaltmak için parlaklığı, tonu ve diğer faktörleri de ayarlanır. Veri büyütme işleminin en eski ve en yaygın kullanılan yöntemlerinden biri de görüntü çevirme (image flip) yöntemidir. Görüntüyü 4 yöne (yukarı, aşağı, sola ve sağa) çevrilir, bu da genellikle nesnenin kategorisini değiştirmez (Zhang ve ark., 2018).

6.5. Veri Setleri

Modellerin sınıflandırma başarısını test edebilmek için bu çalışmada iki farklı veri seti hazırlanmıştır. Bunlardan ilki 656 adet görüntü içeren ve sağlıklı üzüm meyvesi (Healthy_Grape), hastalıklı üzüm meyvesi (Diseased_Grape), sağlıklı üzüm yaprağı (Healthy_Grape_Leaf) ve hastalıklı üzüm yaprağı (Diseased_Grape_Leaf) sınıfları olmak üzere 4 sınıfa ayrılmış olan veri setidir. Diğeri ise 5208 adet görüntü içeren ve 12 sınıfa ayrılmış veri setidir. Sağlıklı ve hastalıklı üzüm yaprağı görüntülerin sınıfları (GL_Healthy, GL_BlackRot, GL_BlisterMite, GL_DownyMildew, GL_Esca, GL_LeafBlight ve GL_PowderyMildew), sağlıklı ve hastalıklı üzüm meyvesi görüntülerin sınıfları (G_Healthy, G_BlackRot, G_BotrytisBunchRot, G_DownyMildew ve G_PowderyMildew) şekilde düzenlenmiştir.

Daha sonra LabelImg yazılımı (Tzutalin, 2018) kullanılarak her iki veri setinin tüm görüntüleri ait oldukları sınıflarına göre etiketlenmiştir. LabelImg yazılımı etiketlenmiş eğitim ve test görüntüleri için xml uzantılı dosyalar oluşturur. Bu dosyalar daha sonra Tensorflow tarafından okunabilmesi için csv uzantılı dosyalara

dönüştürülmüştür. Tüm görüntüler ve karşılık gelen sınıfları hakkında bilgiler içeren training.csv ve testing.csv adlı iki dosya oluşturulmuştur.

6.5.1. Birinci veri seti

Çizelge 6.1'de gösterildiği gibi, sağlıklı ve hastalıklı üzüm meyvesi ve üzüm yaprağı görüntüleri internetten toplanan 136 adet sağlıklı üzüm meyvesi ve 124 adet hastalıklı üzüm meyvesi görüntüsü elde edilmiştir. Üzüm yaprağı görüntülerinin bir kısmı internetten toplanmış, bir kısmı da PlantVillage (Hughes ve Salathe, 2015) veri setinden alınmıştır. Toplam 201 adet sağlıklı üzüm yaprağı ve 195 adet hastalıklı üzüm yaprağın görüntüsü elde edilmiştir. Sonuç olarak bu veri seti 656 adet görüntüden oluşmaktadır.

Çizelge 6.1 Birinci Veri Seti

Görüntü Sınıfları	Toplam	Test	Eğitim	Toplam	Test	Eğitim
Healthy_Grape	136	26	110	656	113	543
Healthy_Grape_Leaf	201	41	160			
Diseased_Grape	124	16	108			
Diseased_Grape_Leaf	195	30	165			

Veri setinin oluşturulması ile görüntüler eğitim ve test veri seti olarak ikiye bölünmüştür. Burada 543 adet görüntü eğitim, 113 adet görüntü test için olacak şekilde iki klasöre bölünmüştür. Daha sonra LabelImg yazılımı kullanılarak tüm bu görüntüler Healthy_Grape, Diseased_Grape, Healthy_Grape_Leaf ve Diseased_Grape_Leaf isimleriyle 4 sınıfa etiketlenmiştir.

6.5.2. İkinci veri seti

Bu veri setinin görüntüleri 3 farklı şekilde toplanmış ve veri büyütme (data agumentation) tekniklerini kullanarak veri sayısı arttırılmıştır. Bu çalışma için tüm görüntülerin boyutu (görüntünün genişliği + görüntünün yüksekliği = 500 piksel) içerisinde tutulmaya çalışılmıştır.

İlk olarak, GL_Healthy, GL_Esca, GL_BlackRot, GL_LeafBlight ve GL_PowderyMildew sınıflarının verileri PlantVilage veri kümesinden alınmıştır.

İkinci olarak, GL_BlisterMite, G_BotrytisBunchRot ve G_BlackRot sınıflarının verileri isimleri kullanılarak internetten toplanmış ve toplanan görüntüler bu çalışmaya göre çözünürlüklerini kaybetmeden yeniden boyutlandırılmıştır.

Üçüncü olarak, GL_DownyMildew, G_Healthy, G_DownyMildew ve G_PowderyMildew sınıflar ile ilgili videolar internetten indirilmiştir şu aşamalar takip edilmiştir:

- İndirilen bütün videolardan alakasız bölümler kaldırılmıştır.
- İşlenen videolar veri artırmak (data augmentation) için 4 yöne (yukarı, aşağı, sağa ve sola) döndürülmüştür.
- Her videodan kareler bu çalışma için istenen piksel boyutlarına çıkarılmıştır.
- Videodan elde edilen ve birbirine çok benzeyen görüntüler silinmiş ve sadece daha fazla farklılığa sahip olan görüntüler veri setine eklenmiştir.

Farklı şekillerde toplanmış olan tüm görüntüler bir klasöre kaydedilmiştir. Sonraki aşamada klasördeki tüm görüntüler karıştırılmış ve yeniden adlandırılmıştır. Toplam sağlıklı ve hastalıklı üzüm meyvesi ve üzüm yaprağının **5208** adet görüntüsü bu veri seti için elde edilmiştir.

Çizelge 6.2 İkinci Veri Seti

Görüntü Sınıfları	**Toplam**	**Test**	**Eğitim**	**Toplam**	**Test**	**Eğitim**
G_Healthy	396	79	317	5208	1037	4171
G_BlackRot	518	104	414			
G_BotrytisBunchRot	590	118	472			
G_DownyMildew	337	67	270			
G_PowderyMildew	299	59	240			
GL_Healthy	496	99	397			
GL_BlackRot	466	93	373			
GL_BlisterMite	459	92	367			
GL_DownyMildew	368	74	294			
GL_Esca	463	93	370			
GL_LeafBlight	403	81	322			
GL_PowderyMildew	413	78	335			

Çizelge 6.2'de gösterildiği gibi, bu veri seti, 396 tane G_Healthy görüntüsü, 337 tane G_DownyMildew görüntüsü, 299 tane G_PowderyMildew görüntüsü, 590 tane G_BotrytisBunchRot görüntüsü ve 518 tane G_BlackRot görüntüsü içeren 2140 adet sağlıklı ve hastalıklı üzümlerin görüntülerini içermektedir. 496 tane GL_Healthy görüntüsü, 368 tane GL_DownyMildew görüntüsü, 413 tane GL_PowderyMildew görüntüsü, 459 tane GL_BlisterMite görüntüsü, 463 tane GL_Esca görüntüsü, 403 tane GL_LeafBlight görüntüsü ve 466 tane G_BlackRot görüntüsü içeren 3068 adet sağlıklı ve hastalıklı yaprakların görüntülerini içermektedir.

Bu görüntülerin 4171 tanesi eğitim için, 1037 tanesi de test için olacak şekilde iki klasöre bölünmüştür. Daha sonra LabelImg yazılımı kullanılarak tüm bu görüntüler 12

sınıfa etiketlenmiştir. Bu sınıfların adları: GL_Healthy, GL_BlackRot, GL_BlisterMite, GL_DownyMildew, GL_Esca, GL_LeafBlight, GL_PowderyMildew, G_Healthy, G_BlackRot, G_BotrytisBunchRot, G_DownyMildew ve G_PowderyMildew olarak belirlenmiştir.

7. DENEYSEL SONUÇLAR VE TARTIŞMALAR

Daha önce de bahsedildiği gibi, bu çalışmada başka bir işlem için önceden eğitilmiş bir modeli kullanan aktarımlı öğrenme yaklaşımı kullanılmıştır. İlk olarak, sınıfın kimlik numaraları tanımlayarak her görüntünün ne içerdiğini sınıflayıcıya söyleyecek etiket haritası oluşturulmuştur. Sonra her iki modelin (SSD_MobileNet v1 ve Faster-R-CNN Inception v2) bazı parametreleri bu çalışmada faydalı olacak parametrelerle değiştirilmiştir.

Her iki modelin konfigürasyon dosyalarında eval_input_reader (test verisine işaret eden), num_examples (test görüntülerinin sayısı), num_class (çıktı sınıfı sayısı), train_input_reader (eğitim verilerine işaret eden) ve fine_tune_checkpoints (ince ayar noktaları) bölümleri bu çalışmaya göre değiştirilmiştir. Bu çalışma için kullanılmış olan bilgisayarın bellek büyüklüğünü aştığından dolayı, SSD_MobileNet v1 modelindeki batch büyüklüğü 24'ten 1'e indirilmiştir.

7.1. Modellerin Eğitilmesi

Bu çalışma için en iyi sonuçları elde etmek için ve en etkili yöntemi bulabilmek üzere, her iki model hem küçük veri setinin hem de büyük veri setinin eğtim verilerinin üzerinde farklı zaman sürelerde ayrı ayrı eğitilmiştir.

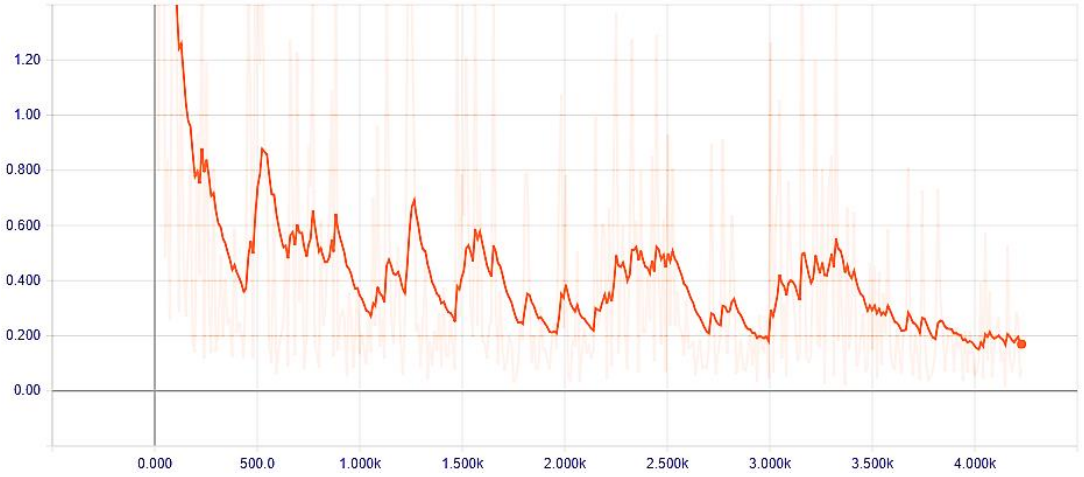
7.1.1. Birinci veri setinin üzerinde modellerin eğitilmesi

Her iki model bu veri setinin 543 eğitim verileri (görüntülerini) kullanılarak eğitilmiş ve modellerin eğitim süresi ve eğitim hızları incelenmiştir.

7.1.1.1. Faster-R-CNN Inception v2 modelin eğitilmesi

Bu model toplam 6981 eğitim adıma kadar, 22 saat boyunca eğitilmiştir. Bu modelin eğitmesi daha yavaş olmuştur, her adım için yaklaşık 9 ile 17 saniye zaman almıştır. Fakat bu modelin eğitilmesi oldukça iyi bir şekilde ilerlemiş, kayıp değerleri (loss value) başta yüksek iken eğitimin başlarında çok ani azalmıştır.

Faster-R-CNN Inception v2 modeli eğitirken TensorBoard tarafından oluşturulan Toplam Kayıp değerlerinin grafiği şekil 7.1'de gösterilmiştir.

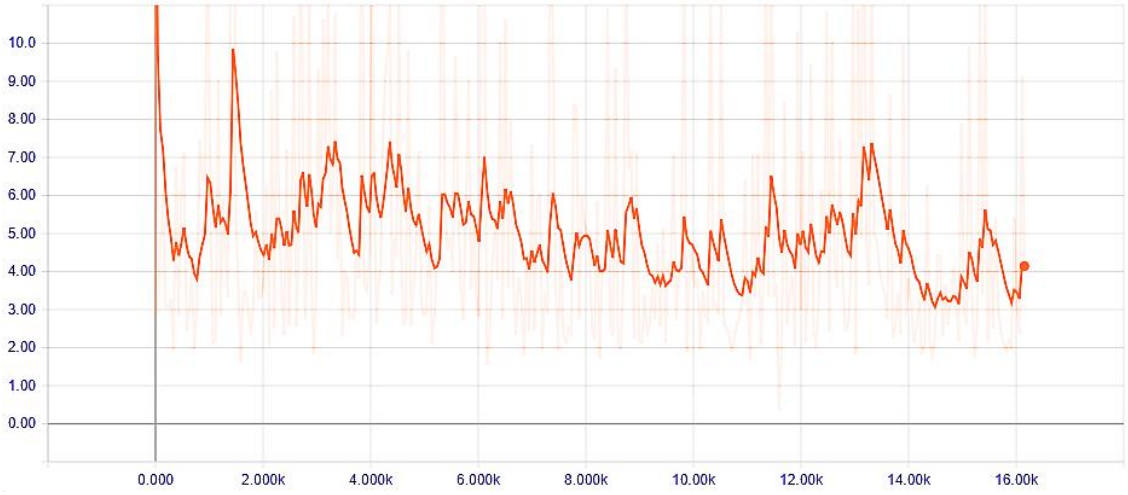


Şekil 7.1. Birinci veri seti için Faster-RCNN Inception v2 modelin Toplam Kayıp değerlerinin grafiği

7.1.1.2. SSD MobileNet v1 modelin eğitilmesi

SSD_MobileNet v1, toplam 16229 eğitm adıma kadar, 12 saat boyunca eğitilmiştir. Bu modelin Faster-R-CNN Inception v2 model'den daha çok hızlı bir şekilde eğitimi gerçekleştirilmiştir, her adım için yaklaşık 2 ile 3 saniye zaman almıştır. Ancak, iyi bir şekilde eğitilemediği gözlenmiştir. Çünkü, daha yüksek kayıp değerleri ile başlamış ve bu değerler diğer modele göre çok yavaş azalmıştır. Kayıp değerlerin azalabilmesi için binlerce eğitim adımına gerek duyulmuştur. Bu yüzden daha iyi sonuçlar elde edebilmek için bu model önceki modele göre daha fazla adımda eğitilmiştir.

Şekil 7.2'de SSD_MobileNet v1 modeli eğitirken TensorBoard tarafından oluşturulan Toplam Kayıp değerlerinin grafiği gösterilmiştir.



Şekil 7.2. Birinci veri seti için SSD_MobileNet v1 modelin Toplam Kayıp değerlerinin grafiği

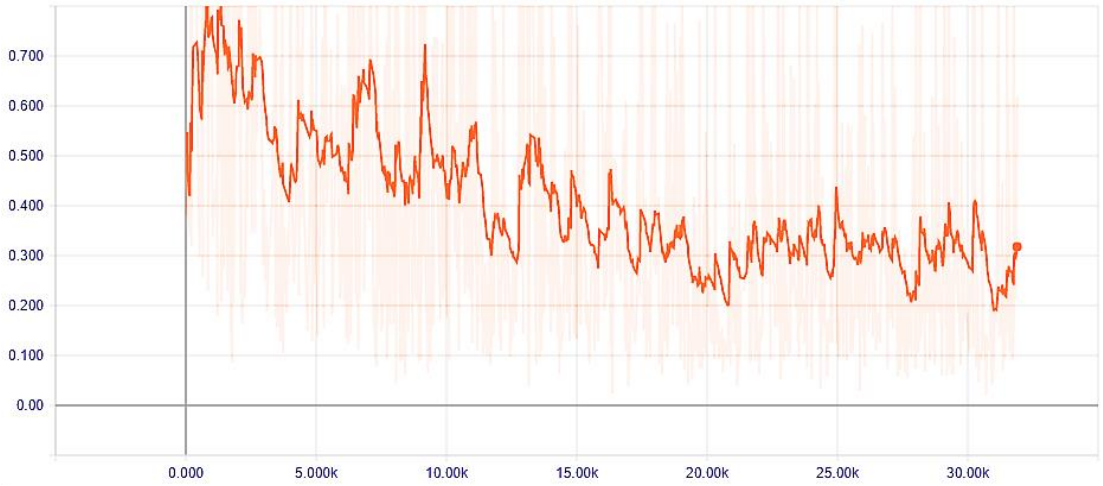
7.1.2. İkinci veri setinin üzerinde modellerin eğitilmesi

Her iki model bu veri setinin 4171 adet eğitim verileri (görüntüleri) kullanılarak eğitilmiş ve modellerin eğitim süresi ve eğitim hızları incelenmiştir.

7.1.2.1. Faster-R-CNN Inception v2 modelin eğitilmesi

Bu modelin kayıp değeri en düşük olan 32029 eğitim adımına kadar toplam 96 saat boyunca eğitilmiştir. Bu veri seti için bu modelin eğitimi yavaş olmuştur. Her adım için yaklaşık 9 ile 12 saniye zaman almıştır. Fakat bu modelin eğitilmesi oldukça iyi bir şekilde ilerlemiş, kayıp değerleri yüksek iken eğitimin başlarında çok ani azalmıştır.

Faster-R-CNN Inception v2 modeli eğitirken TensorBoard tarafından oluşturulan Toplam Kayıp değerlerinin grafiği şekil 7.3'te gösterilmiştir.

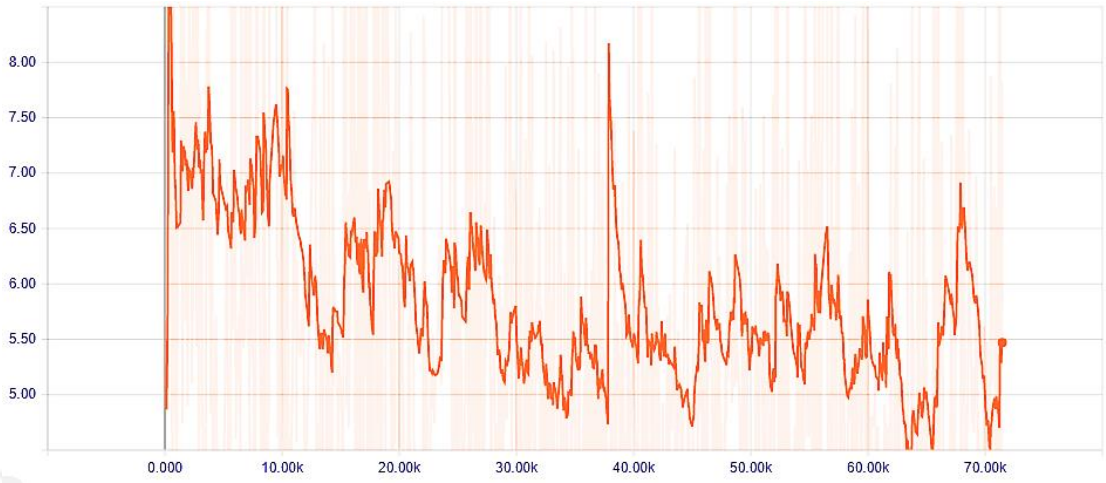


Şekil 7.3. İkinci veri seti için Faster-RCNN Inception v2 modelin Toplam Kayıp değerlerinin grafiği

7.1.2.2. SSD MobileNet v1 modelinin eğitilmesi

SSD_MobileNet v1, toplam 7187 eğitim adımına kadar, 50 saat boyunca eğitilmiştir. Bu model Faster-R-CNN Inception v2 modelinden daha çok hızlı bir şekilde eğitilmiştir. her adım için yaklaşık 2 ile 3 saniye zaman almıştır. Ancak, iyi bir şekilde eğitilememiştir. Çünkü, daha yüksek kayıp değerleri ile başlamış ve bu değerler diğer modele göre çok yavaş azalmıştır. Kayıp değerlerin azalabilmesi için binlerce eğitim adımına gerek duyulmuştur. Bu yüzden daha iyi sonuçlar elde edebilmek için bu model önceki modele göre daha fazla adımda eğitilmiştir. Şekil 7.4'te SSD_MobileNet v1

modeli eğitirken TensorBoard tarafından oluşturulan Toplam Kayıp değerlerinin grafiği gösterilmiştir.



Şekil 7.4. İkinci veri seti için SSD_MobileNet v1 modelin Toplam Kayıp değerlerinin grafiği

7.2. Modellerin Test Edilmesi

Modeller her iki veri setinin eğitim verileri kullanılarak ayrı ayrı eğitildikten sonra, modellerin son sonuçlarını elde etmek için veri setlerinin test veriler üzerinde modeller ayrı ayrı test edilmiştir.

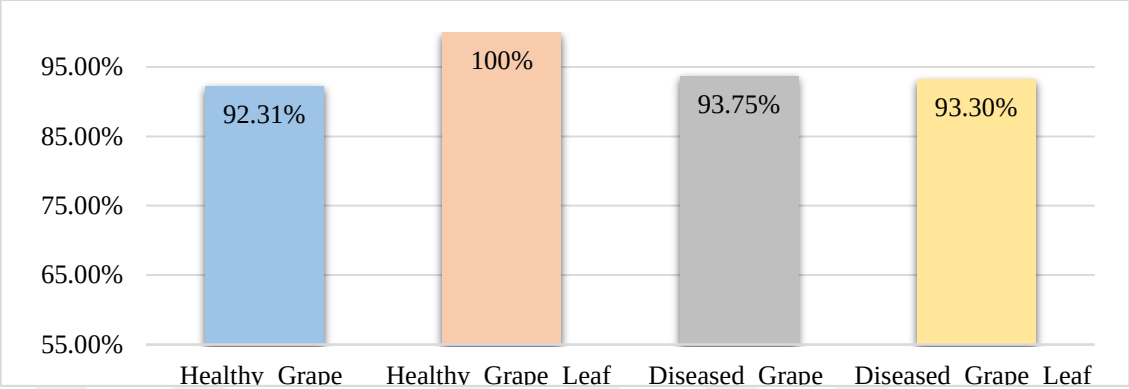
7.2.1. Birinci veri seti için modellerin test sonuçları

Modeller, 26 tane Healthy_Grape, 16 tane Diseased_Grape, 41 tane Healthy_Grape_Leaf ve 30 tane Diseased_Grape_Leaf görüntüsü içeren **113** görüntü üzerinde test edilmiştir. Modellerin gerçek zamanlı olarak gösterdiği performansı incelemek için, bu görüntüleri mümkün olduğunca doğal (görüntülerin farklı kameralar tarafından alınan farklı çözünürlüklerde farklı boyutları sahip olmaları) tutulmaya çalışılmıştır. Bazı görüntülerin arka planları aynıyken bazılarının çok gürültülü arka planları vardır.

7.2.1.1. Faster-RCNN Inception v2 modelin test sonuçları

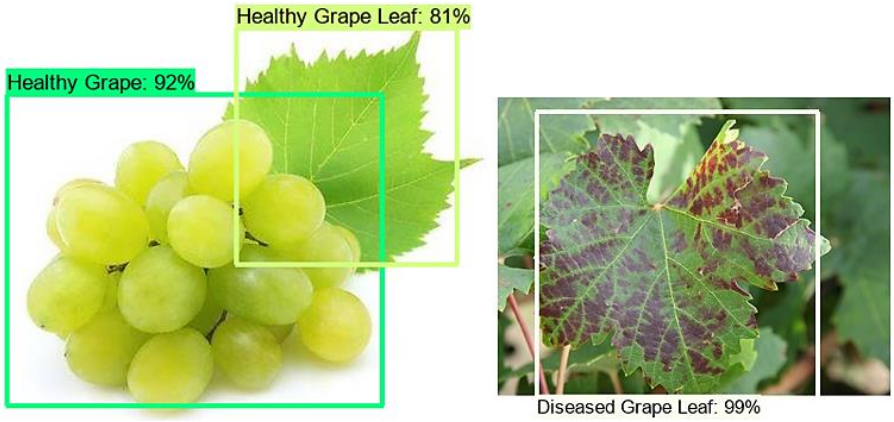
İlk olarak, tüm test görüntüler Faster-R-CNN Inception v2 modelinde tek tek test edilmiş ve oldukça iyi sonuçlar elde edilmiştir. Bu model tüm test görüntülerinin neredeyse %95.57'sini doğru şekilde sınıflandırabilmiştir. Şekil 7.5'teki grafikte

gösterildiği gibi, bu model toplam test görüntülerinden büyük bir yüzdesini doğru şekilde sınıflandırabilmiştir. Healthy_Grape görüntülerinin %92.31'ini, Healthy_Grape_Leaf görüntülerinin %100'ünü, Diseased_Grape görüntülerinin %93.75'ini ve Diseased_Grape_Leaf görüntülerinin %93.3'ünü doğru şekilde sınıflandırabilmiştir.



Şekil 7.5. Birinci veri seti için Faster-RCNN Inception v2 modelin test sonuçlar grafiği

Faster-R-CNN Inception v2 modeli kullanılarak %78 ile %99 arasında sınıflandırma doğruluğu elde edilmiştir. Bu model oldukça yüksek bir sınıflandırma doğruluğuna sahiptir. Ancak işlem süresi daha yavaş olmuştur. Yani bu model her görüntü için ortalama 25-30 saniye işlem süresi almıştır. Bu modelin test sırasında elde edilen sınıflandırma doğruluğunun bir örneği şekil 7.6'de sunulmuştur.

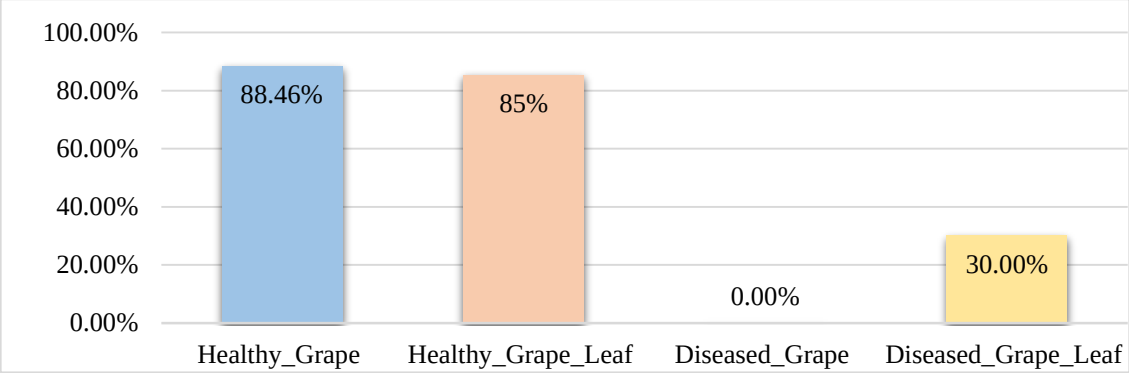


Şekil 7.6. Birinci veri seti için Faster-RCNN Inception v2 modelin test doğruluk öreği

7.2.1.2. SSD_MobileNet v1 modelin test sonuçları

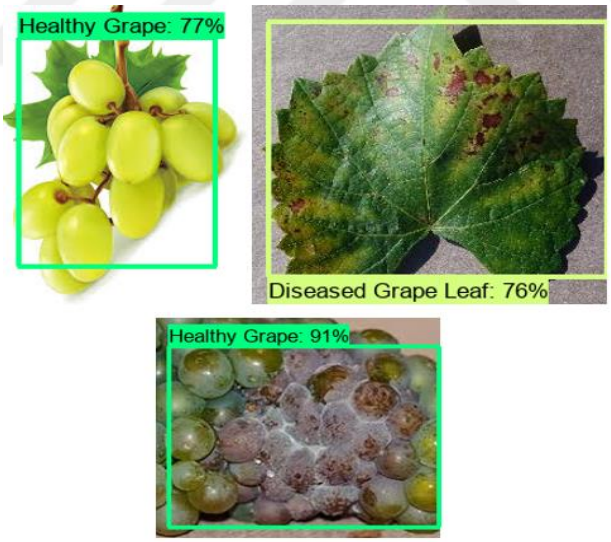
Aynı 113 adet test görüntüsü SSD_MobileNet v1 modelinde test için kullanılmıştır. Ancak bu modelin gösterdiği sonuçlar yeterince iyi olmamıştır. Bu model daha iyi işlem süresi ile tüm test görüntülerinin neredeyse %59.9'unu doğru şekilde sınıflandırabilmiştir. Model her görüntü için ortalama 14-15 saniye işlem süresi almıştır.

Şekil 7.7'deki grafikte gösterildiği gibi, bu model toplam test görüntülerinin daha düşük bir yüzdesini doğru şekilde sınıflandırabilmiştir. Sonuç olarak test görüntülerdeki, Healthy_Grape görüntülerinin %88.46'sını, Healthy_Grape_Leaf görüntülerinin %85'ini, Diseased_Grape görüntülerinin sadece %1'ini ve Diseased_Grape_Leaf görüntülerinin %30'unu doğru şekilde sınıflandırabilmiştir.



Şekil 7.7. Birinci veri seti için SSD_MobileNet v1 modelin test sonuçlar grafiği

Bu modelin test sırasındaki sınıflandırma doğruluğunun bir örneği şekil 7.8'da sunulmuştur.



Şekil 7.8. Birinci veri seti için SSD_MobileNet v1 modelin test doğruluk örneği

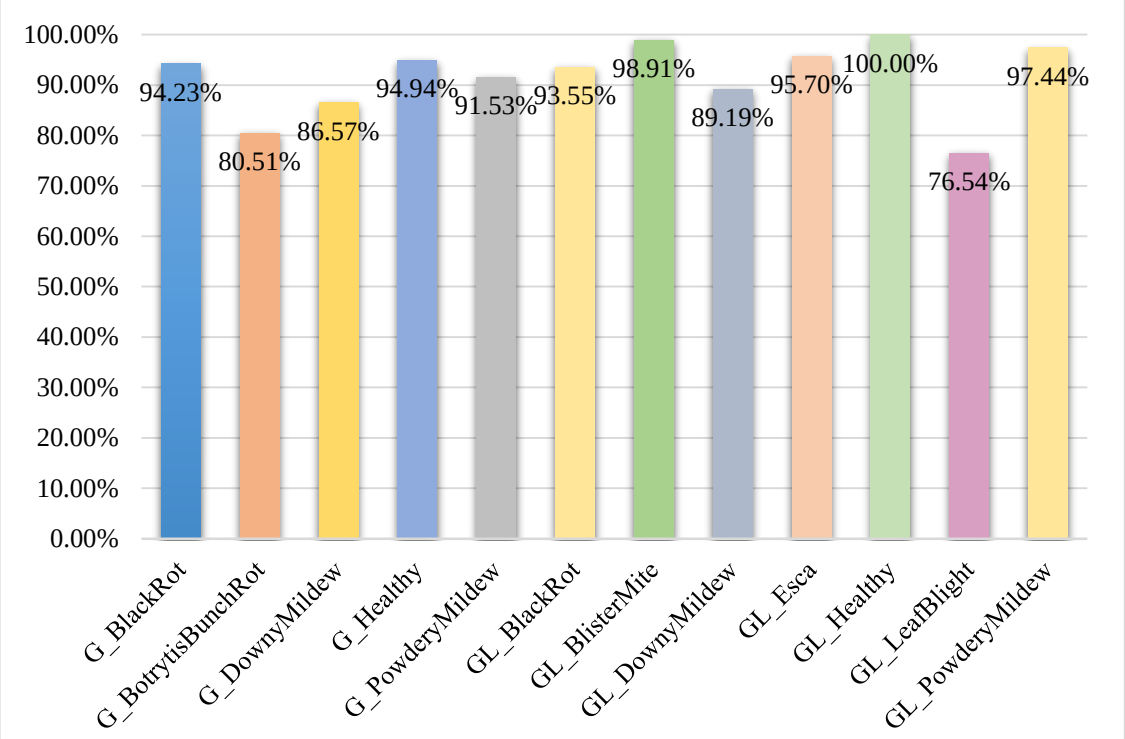
7.2.2. İkinci veri seti için modellerin test sonuçları

Modeller, 99 adet GL_Healthy, 93 adet GL_BlackRot, 92 adet GL_BlisterMite, 74 adet GL_DownyMildew, 93 adet GL_Esca, 81 adet GL_LeafBlight, 78 adet GL_PowderyMildew, 79 adet G_Healthy, 104 adet G_BlackRot, 118 adet

G_BotrytisBunchRot, 67 adet G_DownyMildew ve 59 adet G_PowderyMildew görüntüsü içeren **1037** görüntü üzerinde test edilmiştir.

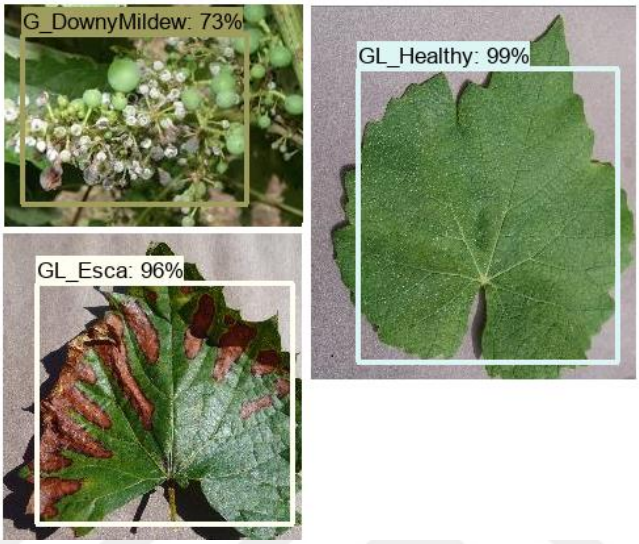
7.2.2.1. Faster-RCNN Inception v2 modelin test sonuçları

İlk olarak, tüm test görüntüler Faster-RCNN Inception v2 modelinde tek tek test edilmiştir, ve oldukça çok iyi sonuçlar elde edilmiştir. Bu model tüm test görüntülerinin %91,61'sini doğru şekilde sınıflandırabilmiştir. Şekil 7.9'daki grafiğinde gösterildiği gibi, bu model toplam test görüntülerinden büyük bir yüzdesini doğru şekilde sınıflandırabilmiştir. Sonuç olarak G_BlackRot görüntülerinin %94.23'ünü, G_BotrytisBunchRot görüntülerinin %80.51'ini, G_DownyMildew görüntülerinin %86.57'sini, G_Healthy görüntülerinin %94.94'ünü, G_PowderyMildew görüntülerinin %91.53'ünü, GL_BlackRot görüntülerinin %93.55'ini, GL_BlisterMite görüntülerinin %98.91'ini, GL_DownyMildew görüntülerinin %89.19'unu, GL_Esca görüntülerinin %95.70'ini, GL_Healthy görüntülerinin %100'ünü, GL_LeafBlight görüntülerinin %76.54'ünü ve GL_PowderyMildew görüntülerinin %97.44'ünü doğru sınıflandırabilmiştir.



Şekil 7.9. İkinci veri seti için Faster-RCNN Inception v2 modelin test sonuçlar grafiği

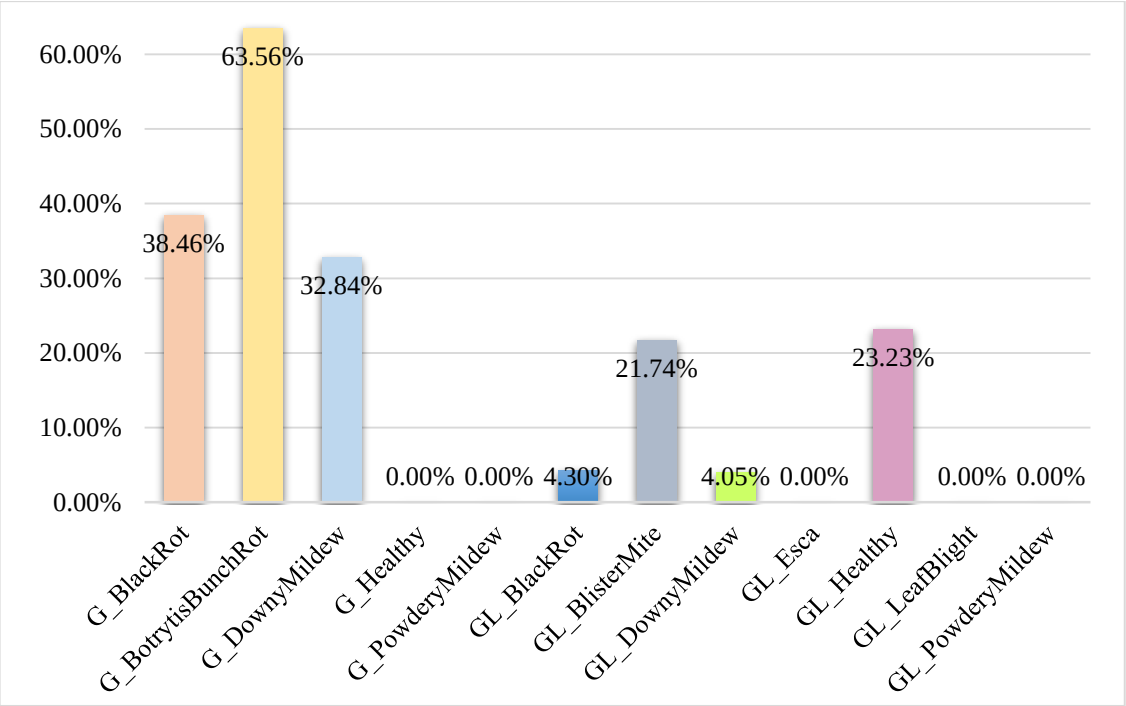
Bu veri setinin çoğu test görüntüsü için Faster-RCNN Inception v2 modeli %70 ile %99 arasında doğru sınıflandırma yapmıştır. Bu model oldukça yüksek bir sınıflandırma doğruluğuna sahip olmuştur, ancak işlem süresi yavaş kalmıştır. Yani bu model her görüntü için ortalama 25-30 saniye işlem süresi almıştır. Bu modelin test sırasında elde edilen sınıflandırma doğruluklarının örnekleri şekil 7.10'da sunulmuştur.



Şekil 7.10. İkinci veri seti için Faster-RCNN Inception v2 modelin test doğruluk öreği

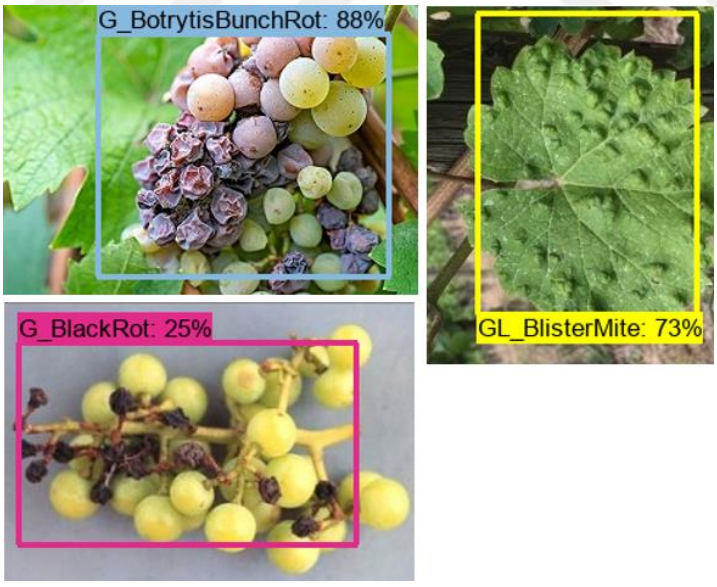
7.2.2.2. SSD_MobileNet v1 modelin test sonuçları

Aynı 1037 adet test görüntüsü SSD_MobileNet v1 modelinde tek tek test edilmiştir. Ancak bu modelin gösterdiği sonuçlar yeterince iyi olmamıştır. Bu model daha iyi işlem süresi ile tüm test görüntülerinin ancak %18'ini doğru şekilde sınıflandırabilmiştir. Her görüntü için ortalama 10- 14 saniye işlem süresi almıştır. Şekil 7.11'deki grafikte gösterildiği gibi, bu model toplam test görüntülerinin daha düşük bir yüzdesini doğru şekilde sınıflandırabilmiştir. Sonuç olarak G_BlackRot görüntülerinin %38.46'sını, G_BotrytisBunchRot görüntülerinin %63.56'sını, G_DownyMildew görüntülerinin %32.84'ünü, G_Healthy görüntülerinin %0.00'ını, G_PowderyMildew görüntülerinin %0.00'ını, GL_BlackRot görüntülerinin %4.30'unu, GL_BlisterMite görüntülerinin %21.74'ünü, GL_DownyMildew görüntülerinin %4.05'ini, GL_Esca görüntülerinin %0.00'ını, GL_Healthy görüntülerinin %23.23'ünü, GL_LeafBlight görüntülerinin %0.00'ını ve GL_PowderyMildew görüntülerinin %0.00'ını sınıflandırabilmiştir.



Şekil 7.11. İkinci veri seti için SSD_MobileNet v1 modelin test sonuçlar grafiği

Bu modelin test sırasında elde edilen sınıflandırma doğruluğunun örnekleri şekil 7.12'de sunulmuştur.



Şekil 7.12. İkinci veri seti için SSD_MobileNet v1 modelin test doğruluk öreği

7.3. Tartışmalar

7.3.1. Birinci veri setinin sonuçları için tartışma

Çizelge 7.1'de gösterildiği gibi, Faster-RCNN Inception v2 modeli, bu veri setinin test görüntülerini SSD_MobileNet v1 modelinden daha iyi sınıflandırabilmiştir. Bu, Faster-RCNN Inception v2 modelinin, görüntüler farklı şekillerde, boyutlarda, çözünürlüklerde ve arka planlarda olmasına rağmen, çok yüksek bir sınıflandırma doğruluğuna sahip 113 görüntüden 108 tanesini (%95.57) doğru şekilde sınıflandırabildiği anlamına gelmektedir. Ancak bu modelde her görüntünün işlenmesi daha uzun sürmüştür. Bu, Faster-R-CNN Inception modelinin bu çalışma için oldukça yararlı olabileceği anlamına gelmektedir.

Çizelge 7.1 Birinci Veri Seti için Modellerin Sonuç Karşılaştırmaları

Modeller	Görüntü Sınıfları	Toplam Görüntüler	Doğru Sınıflandırılmış olan	Toplam Görüntüler	Doğru Sınıflandırılmış olan	Toplam Yüzde
Faster-R-CNN Inception v2	Healthy_Grape	26	24	113	108	**%95.57**
	Healthy_Grape_Leaf	41	41			
	Diseased_Grape	16	15			
	Diseased_Grape_Leaf	30	28			
SSD_MobileNet v1	Healthy_Grape	26	23	113	67	**%59.29**
	Healthy_Grape_Leaf	41	35			
	Diseased_Grape	16	0			
	Diseased_Grape_Leaf	30	9			

Diğer taraftan, SSD_MobileNet v1 modeli tüm görüntüleri doğru şekilde sınıflandıramamıştır. 113 adet test görüntülerinden yanlızca 67 (%59.29) görüntüyü sınıflandırabilmiştir. Ancak bu model görüntüleri diğer modelden daha hızlı işlemiştir. Elde edilen sonuçlara göre, SSD_MobileNet v1 model bu çalışma için bşarılı olamamıştır.

7.3.2. İkinci veri setinin sonuçları için tartışma

Çizelge 7.2'de gösterildiği gibi, Faster-RCNN Inception v2 modeli, bu veri setinin test görüntülerini SSD_MobileNet v1 modelinden daha iyi sınıflandırabilmiştir. Faster-RCNN Inception v2 modeli bu veri setinin test görüntülerinin %91.61'ini yani 1037 görüntüden 950 tanesini doğru şekilde sınıflandırabilmiştir. Ancak bu modelin her görüntünün işlenmesi daha uzun sürmüştür.

Çizelge 7.2 İkinci Veri Seti için Modellerin Sonuç Karşılaştırması

Modeller	Görüntü Sınıfları	Toplam Görüntüler	Doğru Sınıflandırılmış olan	Toplam Görüntüler	Doğru Sınıflandırılmış olan	Toplam Yüzde
Faster-R-CNN Inception v2	G_Healthy	79	75	1037	950	**%91.61**
	G_BlackRot	104	104			
	G_BotrytisBunchRot	118	95			
	G_DownyMildew	67	58			
	G_PowderyMildew	59	54			
	GL_Healthy	99	99			
	GL_BlackRot	93	87			
	GL_BlisterMite	92	91			
	GL_DownyMildew	74	66			
	GL_Esca	93	89			
	GL_LeafBlight	81	62			
	GL_PowderyMildew	78	76			
SSD_MobileNet v1	G_Healthy	79	0	1037	187	**%18.03**
	G_BlackRot	104	40			
	G_BotrytisBunchRot	118	75			
	G_DownyMildew	67	22			
	G_PowderyMildew	59	0			
	GL_Healthy	99	23			
	GL_BlackRot	93	4			
	GL_BlisterMite	92	20			
	GL_DownyMildew	74	3			
	GL_Esca	93	0			
	GL_LeafBlight	81	0			
	GL_PowderyMildew	78	0			

Diğer taraftan, SSD_MobileNet v1 modeli tüm görüntüleri doğru şekilde sınıflandıramamıştır. 1037 test görüntüsünden yanlızca 187 tanesini (%18.03) sınıflandırabilmiştir. Ancak bu model görüntüleri diğer modelden daha hızlı işlemiştir. Bu model için elde edilen sonuçlara göre, SSD_MobileNet v1 modelin bu çalışma için yararlı olamadığı sonucuna varılmıştır.

8. SONUÇLAR VE ÖNERİLER

8.1. Sonuçlar

Üzüm ve Üzüm Yaprağının hastalıklarının tespit edilme işlemi, veri büyütme ve aktarımlı öğrenme tekniklerini ve iki farklı veri seti görüntüleri kullanılarak yapılmıştır. Bu çalışmada önceden eğitilmiş iki derin öğrenme modeli (SSD MobileNet v1 ve Faster R-CNN Inception v2) Tensorflow nesne algılama API'sinin üzerinde çalıştılarak tamamlanmıştır.

6.5.1. bölümde bahsedilen ilk veri seti sadece 4 sınıf (sağlıklı üzüm (Healthy_Grape), hastalıklı üzüm (Diseased_Grape), sağlıklı üzüm yaprağı (Healthy_Grape_Leaf) ve hastalıklı üzüm yaprağı (Diseased_Grape_Leaf)) içermektedir.

6.5.2. bölümde bahsedilen ikinci (büyük) veri seti ise 12 tane sınıf (sağlıklı üzüm yaprağı (GL_Healthy), üzüm yaprağının hastalıkları (GL_BlackRot, GL_BlisterMite, GL_DownyMildew, GL_Esca, GL_LeafBlight, GL_PowderyMildew), sağlıklı üzüm (G_Healthy), üzüm hastalıkları (G_BlackRot, G_BotrytisBunchRot, G_DownyMildew ve G_PowderyMildew)) içermektedir.

Birinci veri seti için, Faster-R-CNN Inception v2 modeli, %70 ile %99 arasında sınıflandırma doğruluğu ile tüm test görüntülerinin %95.57'sini doğru bir şekilde sınıflandırabilmiştir. Bu modelin doğruluk seviyesi daha yüksek ancak daha uzun işlem süresine sahip olmuştur. Bu modelin ortalama işlem süresi görüntü başına 25-30 saniye almıştır. SSD_MobileNet v1 modeli tüm görüntüleri doğru şekilde sınıflandıramamıştır. Bu model tüm test görüntülerinin ancak %59,29'unu doğru bir şekilde sınıflandırabilmiştir. Fakat verileri işleme zamanı daha hızlı olmuştur, her bir görüntüyü işlemesi yaklaşık 14-15 saniye sürmüştür.

İkinci veri seti için, Faster-R-CNN Inception v2 modeli, bu veri setinin tüm test görüntülerinin %91,61'sini %70-99'luk bir sınıflandırma doğruluğu ile sınıflandırmıştır. Bu modelin doğruluk seviyesi daha yüksek ancak daha uzun işlem süresine sahip olmuştur. SSD_MobileNet v1, tüm test resimlerinin sadece %18'ini doğru ancak daha iyi işlem süresiyle sınıflandırmıştır. Bazı sınıflar için bu model oldukça yüksek bir sınıflandırma doğruluğu sağlamıştır (%70 ile %90 arasında), ancak bazı sınıflar için çok zayıf sınıflandırma doğruluğu vermiştir (%0 ile %35 arasında) sağlamıştır. Ancak bu modellerde veri büyütme (Data Augmentation) süreci, sonuçların elde edilmesinde

beklenenin aksine doğruluk oranlarını düşürmüştür. Bu da aktarımlı öğrenmede veri sayısının çok olmasının sonuçlarda faydalı olmadığı anlamına gelmektedir.

Elde edilen sonuçlara göre, SSD_MobileNet v1 modelinin üzüm meyvesi ve yaprağı hastalıklarının sınıflandırmanın işlemi için yararlı olamayacağı ve başarısız olduğu sonucuna varılmıştır. Diğer tarafından, Faster-R-CNN Inception v2 modeli oldukça iyi bir sonuçlar sergilemiş ve daha başarılı bir model olduğu sonucuna varılmıştır.

8.2. Öneriler

Bu çalışmada verilerin insansız hava araçları (dron) ile alınması planlanmıştır. Fakat bitkilerin yapraklarını döktükten sonra veri toplanmaya başlandığı için dron kullanma imkanı bulunamıştır. İleriki çalışmalarda dron ile alınan verilerin anlık işlenmesi bu çalışmanın çok daha iyi uygulanır olmasını sağlayacaktır.

Derin sinir ağlarını sıfırdan oluşturmak ve eğitmek çok uzun sürdüğü için bu çalışmada aktarımlı öğrenme kullanılmıştır. İleriki çalışmalarda derin sinir ağlarını sıfırdan oluşturup eğiterek bu çalışmanın daha verimli sonuçlar vermesi amaçlanmaktadır.

Bu çalışma başka bitkilerin hastalıklarının tespit edilmesinde de kullanılabilir. Böylece sadece tek bitkinin değil çeşitli bitkilerin yetiştiği ortamlarda da hastalıkların tespitinin yapılması kolaylaşacaktır ve tarım alanında daha fazla katkı sağlayacaktır.

KAYNAKLAR

Amara, J., Bouaziz, B. and Algergawy, A., 2017, A Deep Learning-based Approach for Banana Leaf Diseases Classification, Stuttgart: BTW workshop, 79-88, *Bonn, Germany*.

Amini, A., 2019, 6.S191: Introduction to Deep Learning, *MIT's introductory course on deep learning methods and applications.*, Massachusetts Institute of Technology, http://introtodeeplearning.com/, [Ziyaret Tarihi: 15 Haziran 2019].

Chen, S. W., Shivakumar, S. S., Dcunha, S., Das, J., Okon, E., Qu, C., Taylor, C. J. and Kumar, V., 2017,Counting Apples and Oranges With Deep Learning: A Data-Driven Approach, *IEEE Robotics and Automation Letters*, **2**(2), 781-788, doi: 10.1109/LRA.2017.2651944.

Chollet, F., 2017, Deep Learning with Python, *Manning Publications Company*.

Demchak, K., 2017, *Grape Disease - Black Rot* [online], https://extension.psu.edu/grape-disease-black-rot, [Ziyaret Tarihi: 3 Haziran 2019].

Doman, E., 2015, *7 Most Common Grapevine Diseases* [online], https://learn.winecoolerdirect.com/common-grapevine-diseases/, [Ziyaret Tarihi: 2 Haziran 2019].

Durmuş, H., Güneş, E. O. and Kırıcı, M., 2017 of Conference, Disease detection on the leaves of the tomato plants by using deep learning, *6th International Conference on Agro-Geoinformatics*, 1-5, doi: 10.1109/Agro-Geoinformatics.2017.8047016.

Ellis, M. A., 2008a, *Botrytis Bunch Rot or Gray Mold of Grape* [online], https://ohioline.osu.edu/factsheet/plpath-fru-03, [Ziyaret Tarihi: 3 Haziran 2019].

Ellis, M. A., 2008b, *Powdery Mildew of Grape* [online], https://ohioline.osu.edu/factsheet/plpath-fru-37, [Ziyaret Tarihi: 3 Haziran 2019].

Fuentes, A., Yoon, S., Kim, S. C. and Park, D. S., 2017,A Robust Deep-Learning-Based Detector for Real-Time Tomato Plant Diseases and Pests Recognition, *Sensors*, **17**(9), 2022. MDPI AG, doi: 10.3390/s17092022.

Goodfellow, I., Bengio, Y. and Courville, A. (2016). Convolutional Networks. Deep learning. London, England : MIT Press, Cambridge, Massachusetts.

Grabowski, M., 2018, *Alternaria leaf blight* [online], https://extension.umn.edu/diseases/alternaria-leaf-blight, [Ziyaret Tarihi: 4 Haziran 2019].

Hanson, A. M. J., Joy, A. and Francis, J., 2017,Plant Leaf Disease Detection using Deep Learning and Convolutional Neural Network, *International Journal of Engineering Science and Computing*, **7**(3), 5324-5328.

Horea, M. and Mihai, O., 2018,Fruit recognition from images using deep learning, *Acta Universitatis Sapientiae, Informatica*, **10**(1), 26-42, doi: https://doi.org/10.2478/ausi-2018-0002.

Howard, A. G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M. and Adam, H., 2017, MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications, https://arxiv.org/abs/1704.04861, [Ziyaret Tarihi: 18 Mart 2019].

Hughes, D. P. and Salathe, M., 2015, An Open Access Repository of Images on Plant Health to Enable the Development of Mobile Disease Diagnostics, https://arxiv.org/abs/1511.08060, [Ziyaret Tarihi: 24 Mart 2019].

Hunt, E. R. and Daughtry, C. S. T., 2018,What good are unmanned aircraft systems for agricultural remote sensing and precision agriculture?, *International Journal of Remote Sensing*, **39**(15-16), 5345-5376, doi: 10.1080/01431161.2017.1410300.

Ienco, D., Gaetano, R., Dupaquier, C. and Maurel, P., 2017,Land Cover Classification via Multi-temporal Spatial Data by Recurrent Neural Networks, *IEEE Geoscience and Remote Sensing Letters*, **14**, 1685-1689.

Jeon, W.-S. and Rhee, S.-Y., 2017,Plant Leaf Recognition Using a Convolution Neural Network, *The International Journal of Fuzzy Logic and Intelligent Systems*, **17**(1), 26-34, doi: IJFIS.2017.17.1.26.

Kakade, N. R. and Ahire, D. D., 2015,REAL TIME GRAPE LEAF DISEASE DETECTION, *International Journal of Advance Research and Innovative Ideas (IJARIIE)*, **1**(4), 598-610.

Kamilaris, A. and Prenafeta-Boldu, F. X., 2018,Deep learning in agriculture: A survey, *Computers and Electronics in Agriculture International Journal*, **147**, 70-90, doi: 10.1016/j.compag.2018.02.016.

Karunakaran, D., 2018a, *Deep learning series 1: Intro to deep learning* [online], https://medium.com/intro-to-artificial-intelligence/deep-learning-series-1-intro-to-deep-learning-abb1780ee20, [Ziyaret Tarihi: 19 Ekim 2018].

Karunakaran, D., 2018b, *Simple Image classification using deep learning—deep learning series 2* [online], https://medium.com/intro-to-artificial-intelligence/simple-image-classification-using-deep-learning-deep-learning-series-2-5e5b89e97926 [Ziyaret Tarihi: 19 Ekim 2018].

Kothavale, S. S. and Barbade, S. R., 2018,Grape Leaf Disease Detection using Image Processing, *International Journal for Scientific REsearch & Development (IJSRD)*, **6**(2), 2573-2574.

Kothawale, S. S., Barbade, S. R. and Mirajkar, P. P., 2018,Grape Leaf Disease Detection Using SVM Classifier, *International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)*, **6**(4), 3288-3294, doi: 10.15680/IJIRCCE.2018.0604018.

Lin, T.-Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P. and Zitnick, C. L., 2014, Microsoft COCO: Common Objects in Context, *Computer Vision – ECCV 2014*, Springer International Publishing, 740-755, *Cham*.

Liu, L., Ouyang, W., Wang, X., Fieguth, P., Chen, J., Liu, X. and Pietikäinen, M., 2018, Deep Learning for Generic Object Detection: A Survey, https://arxiv.org/abs/1809.02165, [Ziyaret Tarihi: 21 Mart 2019].

Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.-Y. and Berg, A. C., 2016, SSD: Single Shot MultiBox Detector, Springer International Publishing, 21-37, *ECCV*.

Lu, H., Fu, X., Liu, C., Li, L.-g., He, Y. and Li, N.-w., 2017,Cultivated land information extraction in UAV imagery based on deep convolutional neural network and transfer learning, *Journal of Mountain Science*, **14**(4), 731-741, doi: 10.1007/s11629-016-3950-2.

Lu, Y., Yi, S., Zeng, N., Liu, Y. and Zhang, Y., 2017,Identification of Rice Diseases using Deep Convolutional Neural Networks, *Neurocomputing*, **267**, 378-384.

Madden, L., Ellis, M., Lalancette, N., Hughes, G. and Wilson, L., 2000,Evaluation of a disease warning system for downy mildew of grapes, *Plant Dis*, **84**(5), 549-554, doi: 10.1094/PDIS.2000.84.5.549.

MathWorks, 2018a, *Convolutional Neural Network* [online], https://www.mathworks.com/solutions/deep-learning/convolutional-neural-network.html [Ziyaret Tarihi: 19 Ekim 2018].

MathWorks, 2018b, *What Is Deep Learning* [online], https://www.mathworks.com/discovery/deep-learning.html [Ziyaret Tarihi: 19 Ekim 2018].

McCool, C., Perez, T. and Upcroft, B., 2017,Mixtures of Lightweight Deep Convolutional Neural Networks: Applied to Agricultural Robotics, *IEEE Robotics and Automation Letters*, **2**, 1344-1351, doi: 10.1109/LRA.2017.2667039.

Mehdipour-Ghazia, M., Yanikoglu, B. A. and Aptoula, E., 2017,Plant identification using deep neural networks via optimization of transfer learning parameters, *Neurocomputing*, **235**, 228-235, doi: 10.1016/j.neucom.2017.01.018.

Minh, D. H. T., Ienco, D., Gaetano, R., Lalande, N., Ndikumana, E., Osman, F. and Maurel, P., 2018,Deep Recurrent Neural Networks for mapping winter vegetation quality coverage via multi-temporal SAR Sentinel-1, *IEEE Geoscience and Remote Sensing Letters*, **15**(3), 464-468, doi: arXiv:1708.03694.

Mishra, M., 2019, *Convolutional Neural Networks, Explained* [online], https://www.datascience.com/blog/convolutional-neural-network, [Ziyaret Tarihi: 16 Haziran 2019].

Mohanty, S. P., Hughes, D. P. and Salathe, M., 2016,Using Deep Learning for Image-Based Plant Disease Detection, *Frontiers in plant science*, **7**doi: 10.3389/fpls.2016.01419.

Mude, S., Naik, D. and Patil, A., 2017,LEAF DISEASE DETECTION USING IMAGE PROCESSING FOR PESTICIDE SPRAYING, *International Journal of Advance Engineering and Research Development*, **4**(4), 1129-1132.

Mugnai , L., Graniti, A. and Surico, G., 1999,Esca (black measles) and brown wood-streaking: Two old and elusive diseases of grapevines, *Plant Disease*, **83**(5), 404-418.

Namin, S. T., Esmaeilzadeh, M., Najafi, M., Brown, T. B. and Borevitz, J. O., 2018,Deep phenotyping: deep learning for temporal phenotype/genotype classification, *Plant Methods*, **14**(1), 66, doi: 10.1186/s13007-018-0333-4.

Nielsen, M., 2019, Neural Networks and Deep Learning, 224, http://neuralnetworksanddeeplearning.com, [Ziyaret Tarihi: 15 Haziran 2019].

Rahnemoonfar, M. and Sheppard, C., 2017,Deep Count: Fruit Counting Based on Deep Simulated Learning, *Sensors*, **17**(4), 905.

Rangarajan, A. K., Purushothaman, R. and Ramesh, A., 2018 of Conference, Tomato crop disease classification using pre-trained deep learning algorithm, *International Conference on Robotics and Smart Manufacturing (RoSMa2018)*, 1040-1047, doi: 10.1016/j.procs.2018.07.070.

Rathod, V., Wu, N. and Pkulzc, 2019, *Tensorflow detection model zoo (documentation)* [online], https://github.com/tensorflow/models/blob/master/research/object_detection/g3doc/detection_model_zoo.md, [Ziyaret Tarihi: 20 Mart 2019].

Ren, S., He, K., Girshick, R. and Sun, J., 2015,Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks, *Advances in Neural Information Processing Systems 28 (NIPS)*, 91-99.

Sharma, S., 2017, *What the Hell is Perceptron?* [online], The Fundamentals of Neural Networks https://towardsdatascience.com/what-the-hell-is-perceptron-626217814f53, [Ziyaret Tarihi: 12 Haziran 2019].

Sladojevic, S., Arsenovic, M., Anderla, A., Culibrk, D. and Stefanovic, D., 2016,Deep Neural Networks Based Recognition of Plant Diseases by Leaf Image Classification, *Computational Intelligence and Neuroscience*, 11, doi: 10.1155/2016/3289801.

Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J. and Wojna, Z., 2016, Rethinking the Inception Architecture for Computer Vision, *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2818 - 2826, doi: 10.1109/CVPR.2016.308.

Szegedy, C., Wei, L., Yangqing, J., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V. and Rabinovich, A., 2015, Going deeper with convolutions, *2015 IEEE*

Conference on Computer Vision and Pattern Recognition (CVPR), 1-9, doi: 10.1109/CVPR.2015.7298594.

The Royal Horticultural Society, 2019, *Grapevine blister mite* [online], https://www.rhs.org.uk/advice/profile?pid=789, [Ziyaret Tarihi: 3 Haziran 2019].

Tiwari, V. M. and Gupta, T., 2017,Plant Leaf Disease Analysis using Image Processing Technique with Modified SVM-CS Classifier, *International Journal of Engineering & management Technology*, **5**(1), 11-17.

Too, E. C., Yujian, L., Njuki, S. and Yingchun, L., 2019,A comparative study of fine-tuning deep learning models for plant disease identification, *Computers and Electronics in Agriculture*, **161**, 272-279.

Tzutalin, D., 2018, *LabelImg* [online], https://github.com/tzutalin/labelImg, [Ziyaret Tarihi: 13 February 2019].

Wang, Z., Hu, M. and Zhai, G., 2018,Application of Deep Learning Architectures for Accurate and Rapid Detection of Internal Mechanical Damage of Blueberry Using Hyperspectral Transmittance Data, *Sensors* **18**, 1126, doi: 10.3390/s18041126.

Wilcox, W. F., Gubler, W. D. and Uyemoto, J. K., 2015, Compendium of grape diseases, disorders, and pests, *Am Phytopath Society*.

Zhang, A., Lipton, Z. C., Li, M. and Smola, A. J., 2018, Dive into Deep Learning., http://en.diveintodeeplearning.org, [Ziyaret Tarihi: 19 Mart 2019].

ÖZGEÇMİŞ

KİŞİSEL BİLGİLER

Adı Soyadı	**:**	Shekofa Ghoury
Uyruğu	**:**	Afganistan
Doğum Yeri ve Tarihi	**:**	31 Aralık 1994 Afganistan
Telefon	**:**	0537 372 61 37
Faks	**:**	
E-Posta	**:**	shekofaghoury@hotmail.com

EĞİTİM

Derece		**Adı**	**İlçe**	**İl**	**Bitirme Yılı**
Lise	:	Durkhanai Lisesi School,	Kabil,	Kabil	2010
Üniversite	:	Kabul Polytechnic Üniversitesi,	Kabil,	Kabil	2015
Yüksek Lisans	:	Selçuk Üniversitesi,	Selçuklu,	Konya	2019
Doktora	:				

İŞ DENEYİMLERİ

Yıl	**Kurum**	**Görevi**
2016	Etisalat Afghanistan	Kıdemli ERP Asistanı
2015	Etisalat Afghanistan	Pazarlamacı
2014	SilverBullet IT services	Bilgisayar Teknisyeni
2014	Afghanistan Institute Banking and Finance	Stajyer
2013	Etisalat Afghanistan	Stajyer

YABANCI DİLLER

İngilizce, Türkçe, Urduca, Hintçe