

**ANKARA ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ**

YÜKSEK LİSANS TEZİ

DERİN ÖĞRENME YÖNTEMLERİ İLE BİTKİ SINIFLANDIRMA

Burak DUMAN

BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI

**ANKARA
2022**

Her hakkı saklıdır

ÖZET

Yüksek Lisans Tezi

DERİN ÖĞRENME YÖNTEMLERİ İLE BİTKİ SINIFLANDIRMA

Burak DUMAN

Ankara Üniversitesi
Fen Bilimleri Enstitüsü
Bilgisayar Mühendisliği Anabilim Dalı

Danışman: Doç. Dr. Recep ERYİĞİT

Bitki hastalıklarının çoğu zaman gözle görülebilir semptomlar gösterir ve günümüzde kabul edilen teknik, deneyimli bir bitki patoloğunun hastalığı, enfekteli bitki yapraklarını sadece gözlem yoluyla teşhis eder. Hastalık tanı sürecinin sadece gözle gözlem yapılması, zaman kaybı ve tanı başarısının patoloğun yetenekleriyle orantılı olduğu gerçeği, bu sorunun bilgisayar destekli tanı sistemleri için mükemmel bir uygulama alanı haline getirmektedir. Başarılı sonuçlara hızlı ve güvenilir ulaşmak için manuel öznitelik çıkarmanın kusursuz olması gereken klasik makine öğrenmesi yöntemleri yerine, ön işleme gerek duymayan ve başarılı bir sınıflandırma yapabilen bir modele ihtiyaç vardır. Bu çalışmada, bitki yaprak hastalıkları sınıflandırmasında derin öğrenme mimarileri incelenmiş ve bu konudaki önerilen modellerden olan Resnet50V2 ve MobileNet performansı karşılaştırılmıştır. Modelleri eğitmek için Kaggle üzerinden yayınlanan elma yaprakları veri seti kullanıldı. İki model için de aynı veri seti ve bilgisayar üzerinden işlem yapılmıştır. Resnet50V2'nin bu yapılan araştırmada MobileNet modeline göre daha iyi sonuç aldığı görülmüştür.

Temmuz 2022, 57 sayfa

Anahtar Kelimeler: Bitki Hastalıkları, Yaprak Resimleri, Derin Öğrenme, Resnet50V2, MobileNet

ABSTRACT

Master Thesis

PLANT CLASSIFICATION WITH DEEP LEARNING METHODS

Burak DUMAN

Ankara University
Graduate School of Natural and Applied Sciences
Department of Computer Engineering

Supervisor: Assoc. Prof. Dr. Recep ERYİĞİT

Plant diseases often show observable symptoms, and the currently recognized knowledgeable, experienced plant pathologist will diagnose the disease in infected plant leaves only by observation. The fact that the disease diagnosis process is only visually observed, the loss of time and the fact that the success of the diagnosis is proportional to the abilities of the pathologist makes this problem more suitable and usable for computer aided diagnosis systems. In order to achieve fast and reliable results that will be considered successful, there is a need for a model that does not require any prior work and can make a classification with a high success rate, instead of the constantly used standard machine learning methods where manual feature extraction must be flawless. In this study, deep learning architectures in the classification of plant leaf diseases were examined and the performance of Resnet50V2 and MobileNet, which are suggested models on this subject, were compared. The apple leaves veri seti published on Kaggle was used to train the models. The same data set and computer were used for both models. In this research, Resnet50V2 was found to have better results than the MobileNet model.

July 2022, 57 pages

Key Words: Plant disease, Leaf image, Deep learning, Resnet50V2, MobileNet

TEŞEKKÜR

Tez çalışmalarım kapsamında yol gösteren, bilgi birikimleriyle bana destek olan, tüm sabrıyla her zaman yanımda olan danışmanım Sayın Doç. Dr. Recep Eryiğit'e (Ankara Üniversitesi Bilgisayar Mühendisliği Anabilim Dalı) en derin sevgi ve saygılarımla teşekkürlerimi sunarım.

Hayatım süresince bütün tüm kararlarımda desteklerini esirgemeyen yanımda olan ve karşılaştığım zorluklar da bana yardımcı olan ablama, anneme ve babama tüm içtenliklerimle teşekkür ederim.

Tez çalışmamda benden yardımını esirgemeyen ve her konuda destek olan eşim Merve Duman'a ve birlikte bu çalışmayı hazırladığım, yeni doğan minik kızıma en derin sevgi ve saygılarımla teşekkür ederim.

Tez çalışmam boyunca benden yardımını esirgemeyen iş arkadaşlarım Hakkı Erduran ve Resul Emre Aygan'a desteklerinden dolayı teşekkürü borç bilirim.

Burak DUMAN
Ankara, Temmuz 2022

İÇİNDEKİLER

TEZ ONAY SAYFASI	
ETİK	i
ÖZET	ii
ABSTRACT	iii
TEŞEKKÜR	iv
KISALTMALAR DİZİNİ	vi
ŞEKİLLER DİZİNİ	vii
1. GİRİŞ	1
2. KURAMSAL TEMELLER ve KAYNAK ÖZETLERİ	4
2.1 Yapay Zekâ	4
2.2 Makine Öğrenmesi	5
2.3 Yapay Sinir Ağları	6
2.4 Derin Öğrenme	10
2.5 Tekrar Eden Sinir Ağları (RNN)	11
2.6 Uzun Kısa Süreli Hafıza Ağları (LSTM)	12
2.7 Sınırlı Boltzmann Makineleri (RBM)	14
2.8 Derin İnanç Ağları (DBN)	15
2.9 Derin Oto-Kodlayıcılar (DAE-AE)	17
2.10 Evrişimli Sinir Ağları Mimarisi	18
2.10.1 Evrişimli Sinir Ağları Katmanları	18
2.11 Konvolüsyonel Sinir Ağları (CNN)	25
2.11.1 VGG16	26
2.11.2 ResNet50	27
2.11.3 Inception V3	28
2.11.4 EfficientNet	28
2.11.5 MobilNet	30
2.11.6 DenseNet	31
2.12 Bitki Sınıflandırma	32
3. LİTERATÜR ARAŞTIRMASI	34
4. ARAŞTIRMA BULGULARI	38
4.1 Kullanılan Çevre Değişkenleri	38
4.2 Veri Seti	38
4.3 Kullanılan Çevre Değişkenleri	41
4.3.1 Veri seti bölümlendirme	42
4.3.2 Eğitim sonuçları	42
5. TARTIŞMA ve SONUÇ	50
KAYNAKLAR	51
ÖZGEÇMİŞ	57

KISALTMALAR DİZİNİ

BGR	Mavi Yeşil Kırmızı
CNN	Konvolüsyonel Sinir Ağları
GAP	Küresel Ortalama Havuzlama
GPU	Grafik İşleme Ünitesi
ILSVRC	ImageNet Büyük Ölçekli Görsel Tanıma Mücadelesi
RBM	Kısıtlı Boltzmann Makinesi
ReLU	Doğrultulmuş Lineer Birim
RGB	Kırmızı Yeşil Mavi
YSA	Yapay Sinir Ağları
ESA	Evrişimli Sinir Ağları



ŞEKİLLER DİZİNİ

Şekil 2.1 Yapay zekânın kronoloji gösterimi	5
Şekil 2.2 Biyolojik Nöron ve YSA	7
Şekil 2.3 Biyolojik ile Yapay Sinir Sistemi Karşılıkları	7
Şekil 2.4 Bir Yapay Nöron Modeli	8
Şekil 2.5 YSA Toplam Fonksiyonları	9
Şekil 2.6 YSA Transfer	9
Şekil 2.7 Yapay Sinir Ağları Algoritmalarının Kullanım Alanları	11
Şekil 2.8 RNN Katman Yapısı	11
Şekil 2.9 Tekrar Eden Sinir Ağlarında Kullanılan Matematik Algoritması.....	12
Şekil 2.10 LSTM mimarisi.....	13
Şekil 2.11 Boltzmann Makinesi	14
Şekil 2.12 Günlük olabilirlik fonksiyonu.....	15
Şekil 2.13 Enerji Fonksiyonu tanımı.....	15
Şekil 2.14 Model Parametrelerine göre $E(x,h)$ Fonksiyonu.....	15
Şekil 2.15 DBN Örneği	16
Şekil 2.16 Ağırlık Hesaplama Formülü(Tek RBM).....	16
Şekil 2.17 Ağırlık Hesaplama Formülü(Çoklu RBM)	16
Şekil 2.18 Sigmoid Fonksiyon Formülü	16
Şekil 2.19 Güncelleme Formülü	17
Şekil 2.20 AE yapısı	17
Şekil 2.21 DAE Yapısı.....	18
Şekil 2.22 5*5*3 Ölçütlere Sahip Görüntüye 3*3 Filtre Uygulanması	19
Şekil 2.23 Evrişimli Sinir Ağları için Giriş Görüntüsü.....	21
Şekil 2.24 Evrişimli Sinir Ağlarının Birinci Konvolüsyon Katmanından Sonra Oluşan Görüntüler	21
Şekil 2.25 Doğrultulmuş Lineer Birim Katmanı Çıkış Verisine Etkisi	22
Şekil 2.26 Bir ESA Modelinde Konvolüsyon ve ReLu Katmanının Etkisi	22
Şekil 2.27 5*5'lik Başlangıç Resmine 2*2 Filtre ile 1 ve 2 Basamak Kaymalı En Yüksek Havuzlama Süreci Uygulanması.....	24
Şekil 2.28 Evrişimli Sinir Ağları Modelinde Havuzlama Tabakasının Evvelki Tabakadan İletilen Resme Uygulanması.....	24
Şekil 2.29 DropOut Katmanın ESA ile Gerçekleştirilmesi.....	25
Şekil 2.30 VGG16 Katmanları	26
Şekil 2.31 Resnet50 Katman Yapısı	27
Şekil 2.32 Resnet50 Detay Mimarisi	27
Şekil 2.33 Inception V3Mimarisi	28
Şekil 2.34 EfficientNet Mimarisi	30
Şekil 2.35 MobilNet Mimarisi	31
Şekil 2.36 DenseNet-201 Mimarisi	32
Şekil 3.1 CNN Mimarileri ve XDNet Başarı Sonuçları.....	34
Şekil 3.2 CNN Mimarileri ve VGG-INCEP Başarı Sonuçları.....	35
Şekil 4.1 Veri Dağılımı	39
Şekil 4.2 Veri Seti Örneği – BGR.....	40
Şekil 4.3 Veri Seti Örneği – RGB.....	41
Şekil 4.4 Veri Seti Bölümlendirme	42
Şekil 4.5 Resnet50V2 Başarı Grafiği	43

Şekil 4.6 Resnet50V2 ile Eğitim Yapılan Örnek 9 Veri	43
Şekil 4.7 Resnet50V2 ile Şekil 4.6'daki Veri Setinin Başarı Oranları	44
Şekil 4.8 MobileNet Başarı Grafiği	44
Şekil 4.9 MobileNet ile eğitim Yapılan Örnek 9 Veri	45
Şekil 4.10 MobileNet ile Şekil 4.9'daki Veri Setinin Başarı Oranları.....	45
Şekil 4.11 Veri Seti Bölümlendirmeye Göre Sonuçlar	46
Şekil 4.12 ResNet50V2 ve MobileNet Mimarilerinin 35%-15% e Göre Başarı Tablosu.....	46
Şekil 4.13 MobileNet Eğitilebilir Parametre Sayısı	47
Şekil 4.14 ResNet50V2 Eğitilebilir Parametre Sayısı.....	47
Şekil 4.15 resNet50v2 ve MobileNet Dönem Süreleri.....	48
Şekil 4.16 CNN Mimarileri Başarı Grafiği	49
Şekil 4.17 CNN Mimarileri Parametre Sayıları	49



1. GİRİŞ

Yapay zekânın dayanakları 19. Yüzyılın sonlarına kadar uzanmaktadır. O zamanlar bir fikir olarak yer alan düşünebilir makineler üzerine çalışmalar başlatılmıştır. Ancak fikrin hayata geçirilme zamanı ikinci dünya savaşı sonrasına kadar sarkmıştır. Savaş sonrası ise yavaş yavaş gelişim görülmüştür.

Isaac Asimov, düşünebilen makineler dayanağını ilk ortaya atanlardandır. Isaac Asimov'un 1945 senesinde yazdığı Runaround isimli kısa öyküsünde ilk kez düşünebilen ve insana benzer eylemler gerçekleştiren robotları işlemiştir.

Alan Turing, bir savaş kahramanı olarak atfedilmektedir. Bunun sebebi ise, II. Dünya Savaşı esnasında Alman şifrelerinin kırılmasında anahtar bir role sahip olmasıdır. Bunun yanı sıra kaleme aldığı makalesinde "Makineler düşünebilir mi?" sorusu ile yapay zekâ düşüncesinin ilk adımlarını atmıştır.

Yapay zekâ, bilgisayar veya denetimindeki bir robotun farklı uygulamaları zeki canlılara denk şekilde yerine getirme becerisi olarak açıklanabilmektedir. Yapay Zekâ Sistemi ise mevcut şartları izleyerek ve bu izlemi öncesinde saptanmış değişkenler yönünden işleyerek çalışmaktadır. Başlıca görevleri problemlere çözüm önerisi getirmek ve mevcut vaziyete tepki hazırlamaktır.

Pek çok yöntem yapay zekânın üretimi için kullanılabilmektedir. Makine öğrenmesi ise bu yöntemlerden bir tanesidir. Yazılım aksiyonlarının alenen programlanmadan sonuçlarını ön görmeyi amaç edinmiş bir çalışma dalıdır. Makine öğrenmesi, aslında bilgisayarların tek başına öğrenebilmesini sağlamaktadır. Bu tarz bir öğrenme, küçük olmayan bilgi setlerini rahatlıkla çalıştıran modern bilgisayarların uygulama eforundan yararlanır. Bu öğrenme, girdi bilgilerini baz alarak istatistiksel çözümleme metoduyla çıktı bilgilerini ön görebilen ve taze girdi bilgileri geldikçe çıktıları güncelleme işlemleri üzerine kuruludur. Denetimli ve denetimsiz olmak üzere iki başlık altında sınıflandırılabilir.

Denetimli öğrenme ile bölümlendirme (classification) ve gerileme (regresyon) sorunları çözüme ulaşabilmektedir. Girdi verilerine ve muhtemel çıktı verilerine sahip, hususları ön görülen bilgi gruplarını kullanmayı içerir. Denetimsiz işlemleyiciler ise bir veri setindeki kümeler (classification) ulaşmak ve bölmek maksadıyla kullanılmaktadır. Denetimli işlemleyicilerin aksine belirsiz yapıdaki bilgi gruplarını kullanmayı içermektedir.

Derin Öğrenme kavramı, makine öğrenmesinin güncel bir kolu olarak konu edinilmiştir. Derin öğrenme genel anlamda tahmin işleminde kullanılan özneteliklerin kendisi tarafından belirlenen birden fazla tabakalı yapay sinir ağlarıdır. Birden fazla tabakalı yapay sinir ağları kullanarak, sorunların çözümlerini deneyim yöntemiyle ön gören bir çalışma dalı olarak tanımlanabilmektedir. Derin öğrenmenin geleneksel metotlardan ayrılmasının bazı sebepleri bulunmaktadır. İşin ehli bireylerle veya manuel kodlamaya gerek duymadan bilginin temsil edilme şeklini kendiliğinden öğrenebilmesi bu sebepler arasında sayılmaktadır. Derin öğrenme, cisim ve yüz teşhisi gibi birçok dalda geleneksel metotlardan başarılı sayılabilmektedir.

Yapay zekâ, makine öğrenmesi ve derin öğrenme birbirini tamamlayan parçalar olarak tanımlanabilmektedir. Bu durum, yapay zekânın kafatası, makine öğrenmesinin beyin, derin öğrenmenin ise beyincik olarak metaforlaştırılmasıyla açıklanabilir.

Evrişimli Sinir ağları, derin öğrenmenin bir alt dalıdır. Evrişimli Sinir ağları son yıllarda görüntü sağlık, tarım ve nesne sınıflandırmada önde gelen yöntemlerden biridir. Tarım alanında bitki tanıma/sınıflandırma, tohum sınıflandırma ve hastalık tespitinde kullanılmaktadır. Özellikle hastalık kritik ise, tehlikeli veya bulaşıcılığı varsa hastalık tespitinin hızlı yapılması çok önemlidir. Çünkü küçük bir alanda başlayan tehlikeli bir hastalık kısa sürede çok fazla miktarda bitkiye yayılabilir. Bu durum tarım bitkileri için verimde sorunlara hatta bitkilerin ölümüne dahi yol açabilir. Maddi kayıplara neden olabilir. Bu sebeple bitki hastalıklarının hızlı ve doğru olarak tespit edilebilmesi için derin öğrenme algoritmaları kullanılabilir.

Makine öğrenmesi vasıtasıyla bir dolu soruna seri çözümler getirilebilir. Türkiye, tarım ülkesi olarak atfedilmektedir. Bu sebeple bitkiler mühim yer tutmaktadır. Bu bağlamdaki bitkinin verim düzeyini arttırabilmek adına hastalıkların tanılanması ve seri çözümler getirilmesi önemli bir husustur. Doğru zamanda müdahale ve hastalığın tanılanmasının seri yapılabilmesi hem verimliliği arttırır hem de maddi zayıyatın önüne geçmektedir. Bazı hastalıklar gözle görülüp tanılanabilmesine rağmen bazı hastalıklar ancak ileri düzey analiz yöntemleriyle tespit edilebilmektedir.



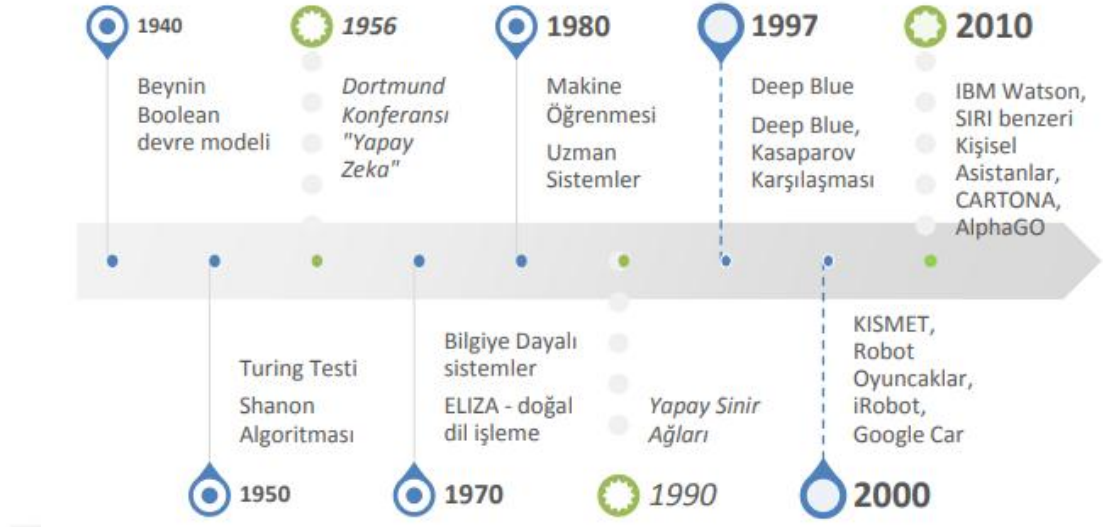
2. KURAMSAL TEMELLER ve KAYNAK ÖZETLERİ

Bu başlık altında yapay zekâ, makine öğrenmesi, sinir ağları ve derin öğrenme tekniklerinden en çok kullanılanlardan, tez için yapılan uygulamada kullanılan tekniklerden bahsedilecektir.

2.1 Yapay Zekâ

Bilgisayar destekli bir makinenin veya bilgisayarın, çoğunlukla insana has özellikler, çare arayışı, anlam bulma, tahmin ve önceki tecrübelerden öğrenme gibi yüksek mantık süreçlere ilişik işlevleri gerçekleştirme becerisi olarak açıklanmaktadır. (Nabiyev 2012) 1956 yılında Dortmund Konferansı'nda John McCarthy ve arkadaşları tarafından bir düşünce mektubu kaleme alınmıştır. Yapay zekâ kavramı ise ilk kez bu mektupta yer almıştır. Buna rağmen kavramın yaratıcısı olarak John McCarthy tanınmaktadır. (Alpaydın 2013) Yapay zekâyı, insana benzer akıllı makineler ya da akıllı bilgisayar tasarısı yapma mühendisliği olarak açıklamıştır.

Bütün bu tanımlar baz alınır, bilgileri ifadelendirmek, önceki bilgileri kullanarak güncel bilgi üretebilen yapıyı akıllı hale getirebilen bir bilim olarak tanımlayabilmek mümkündür.



Şekil 2.1 Yapay zekânın kronoloji gösterimi (Arslan 2020)

2.2 Makine Öğrenmesi

Makine Öğrenmesi, ilgili sorunu, o soruna has bilgiye kıyasla tasarlayan bilgisayar işlemleyicilerinin yaygın adı olarak tanımlanmaktadır. Hâlihazırdaki bilgi grubu ve mevcut işlemleyici ile oluşturulan bir tasarıdır. Maksimum verimi sağlamak amacıyla kurulmaktadır. Bu nedenle çok fazla makine öğrenmesi tekniği geliştirilmiştir. Bunlardan bazıları, k-en yakın komşu işlemleyicisi, karar ağaçları, k-ortalamlar işlemleyicisi, lojistik regresyon analizi ve YSA'lardır. Bu tekniklerin bir bölümü ön görme ve tahmin, bir kısmı ise bölümlendirme becerisine sahiptir. Bu tekniklerde öğrenme izlemleri üç kısımda incelenmektedir. Bunlar, denetimli, denetimsiz ve pekiştirmeli olarak gruplanmaktadır. Denetimli öğrenmede, geliştirilen teknikle birlikte bir kısım girdi verisine binaen amaçlanan veriler tahsis edilir.

Bu durum denetimsiz öğrenmede ise farklılık göstermektedir. Denetimsiz öğrenme de amaçlanan veriler yoktur. Yalnızca girdi verileri ile aralarındaki bağı gösterilmesi amaçlanır. Bu bağlar vasıtasıyla en benzer değerler kümelenir. Yeni alınan girdi verileri bu gruplardan hangisiyle bağlantılı ise o gruba dâhil olmaktadır. Pekiştirmeli öğrenmede ise amaçlanan çıktıyı iletme için bir danışman yerine, kazanılan çıktının

tahsis edilen girdiye karşılık değerlendirici bir unsur kullanılmamaktadır. (Atalay ve Çelik 2017)

2.3 Yapay Sinir Ağları

İnsan beyninin devasa nitelikleri üzerinde pek çok çalışma yapılmıştır. Beynin nörofiziksel durumundan etkilenerek matematiksel modelin ilk adımları atılmaya başlanmıştır. Pek çok suni hücre ve ağ modelleri geliştirilmiştir. Bunun sebebi ise beynin tüm davranışlarını ve fiziki komponentlerin bir bütün olarak modellenmesi gerektiğidir. Bu sebeple Yapay Sinir Ağları olarak adlandırılan, güncel bilgisayar işlemleyicilerinin hesaplama metodundan farklı olarak bir bilim dalı doğmuştur. (Ataseven 2013).

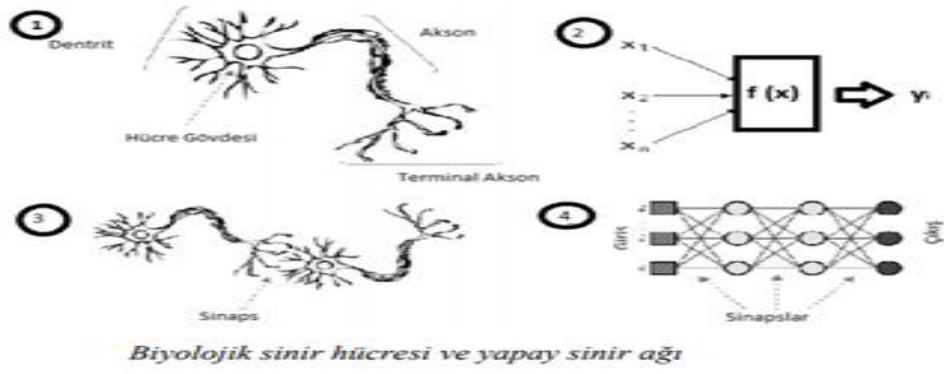
Özetle tanımlanacak olur ise beynin fonksiyonlarını yerine getirme metodunu modellemek için hazırlanan bir sistemdir. Yapay Sinir Ağları, suni sinir hücrelerinin farklı biçimlerde birbirleri ile bağlanmasıdır. Çoğu zaman tabakalar halinde sıralanır. Yapay Sinir Ağları, beynin veri işleme metoduna uygundur. Aynı zamanda öğrenme sürecinden sonra veriyi muhafaza etme ve genelleme becerisine sahip benzer dağılım göstermiş bir işlemcidir. (Anonim 2022a)

YSA, turing makineleriyle inşasına başlanan, yapay zeka bağlamında araştırılması en çok yapılan konudur. Aynı zamanda bütünüyle insan beyninden esinlenerek oluşturulmuştur.

Bir sinir ağı, veriyi muhafaza etmek ve işlevsel hale getirmeye yatkın paralel dağılımlı işlemcidir. İnsan beyniyle iki farklı biçimde benzeşme gösterir.

Veri, öğrenme prosesi vasıtasıyla ağ aracılığı ile kazanılır. (Ergezer vd. 2003)

Nöronlar arası bağlantılardaki kuvvetlerinden, veriyi muhafaza etmek amacıyla yararlanır. (Haykin 1999)



Şekil 2.2 Biyolojik Nöron ve YSA

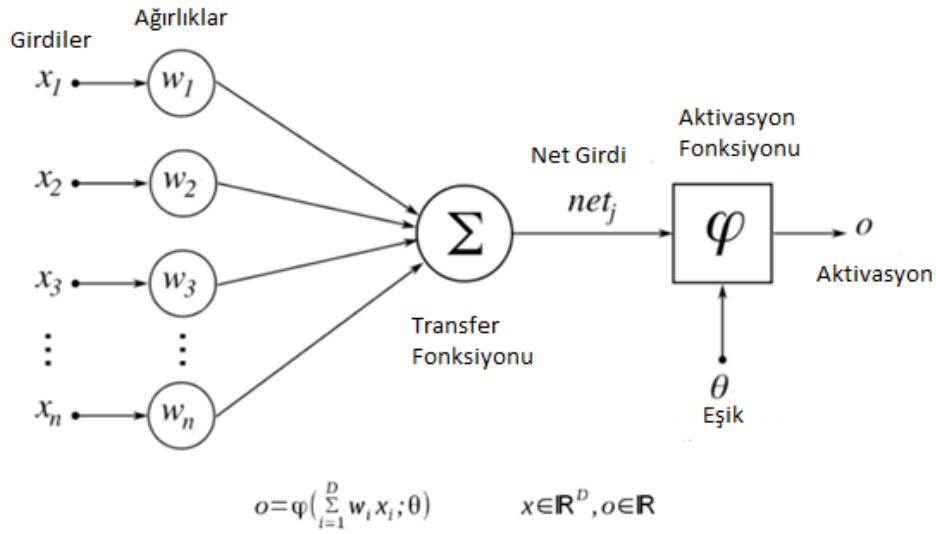
Biyolojik sinir sistemi elemanları ve yapay sinir sisteminde karşılıkları

Biyolojik Sinir Sistemi	Yapay Sinir Sistemi
Nöron	İşlemci Elemanı
Dentrit	Toplama Fonksiyonu
Hücre Gövdesi	Transfer Fonksiyonu
Aksonlar	Yapay Nöron Çıkışı
Sinapslar	Ağırlıklar

Şekil 2.3 Biyolojik ile Yapay Sinir Sistemi Karşılıkları

Kendi başına akıllı kararlar alabilen ve öğrenebilen YSA oluşturmak için katmanlardaki algoritmalar derin öğrenme ile yapılandırılır. YSA üzerindeki çalışan tüm algoritma ve modeller ile kümesi ile derin öğrenmeye geçiş yapılmıştır.

Bir sinir hücresinin yapay nöron matematiksel modeli şekilX deki gibi gösterilmektedir.



Şekil 2.4 Bir Yapay Nöron Modeli (Akbaş 2018)

Sinir hücresinin bölümleri Şekil 2.4’de gösterilmiştir. Bu bölümlerin görevleri aşağıdaki gibidir.

İşlemci Elemanı: İşlemci elemanları nöronlara iletilen verilerdir. Bu işlemci elemanlarına gelen veriler nöronun çekirdeğine yönlendirilir.

Kütleler: Yapay sinir temel birimlerine(hücrelerine) iletilen bilgiler işlemci elemanları ile çekirdeğe gelmeden evvel iletilerin kütlesiyle çarpılır. Bu bilgi çekirdeğe iletilir. Böylece işlemci elemanlarının meydana gelen çıktıya etkisi hesaplanabilmektedir.

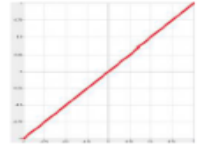
Birleştirme Fonksiyonu: Yapay bir sinir hücresinde kütlelerle çarpılarak gelen işlemci elemanları toplanarak o birime olan net girdi miktarını saptamayı sağlar.

Transfer fonksiyonu: Tüm işlemci elemanlarının kütleli toplamı ve ardından çıkış değeri meydana getiren ve ardındaki tabakaya iletim sağlar.

Yapay Nöron Çıkışı: Transfer fonksiyonundan iletilen sonuç, ilgili hücredeki çıktı değeridir. Hücrelerde çok sayıda girdi bulunabilirken, yalnızca bir adet çıktı bulunabilir. İlgili çıktı çok sayıda hücreyle ilişkilendirilebilir (Anonim 2021a).

Toplam $Net = \sum_{i=1}^N X_i * W_i$	Ağırlık değerleri girdiler ile çarpılır ve bulunan değerler birbirleriyle toplanarak Net girdi hesaplanır.
Çarpım $Net = \prod_{i=1}^N X_i * W_i$	Ağırlık değerleri girdiler ile çarpılır ve daha sonra bulunan değerler birbirleriyle çarpılarak Net Girdi Hesaplanır.
Maksimum $Net = \text{Max}(X_i * W_i)$	n adet girdi içinden ağırlıklar girdilerle çarpıldıktan sonra içlerinden en büyüğü Net girdi olarak kabul edilir.
Minimum $Net = \text{Min}(X_i * W_i)$	n adet girdi içinden ağırlıklar girdilerle çarpıldıktan sonra içlerinden en küçüğü Net girdi olarak kabul edilir.
Çoğunluk $Net = \sum_{i=1}^N \text{Sgn}(X_i * W_i)$	n adet girdi içinden girdilerle ağırlıklar çarpıldıktan sonra pozitif ile negatif olanların sayısı bulunur. Büyük olan sayı hücrenin net girdisi olarak kabul edilir.
Kümülatif Toplam $Net = \text{Net}(\text{eski}) + \sum_{i=1}^N X_i * W_i$	Hücreye gelen bilgiler ağırlıklı olarak toplanır. Daha önce hücreye gelen bilgilere yeni hesaplanan girdi değerleri eklenerek hücrenin net girdisi hesaplanır.

Şekil 2.5 YSA Toplam Fonksiyonları

Doğrusal (Lineer) Aktivasyon Fonksiyonu		$F(\text{NET}) = A * \text{NET}$ (A sabit bir sayı)	Doğrusal problemler çözmek amacıyla aktivasyon fonksiyonu doğrusal bir fonksiyon olarak seçilebilir. Toplama fonksiyonundan çıkan sonuç, belli bir katsayı ile çarpılarak hücrenin çıktısı olarak hesaplanır.
Adım (Step) Aktivasyon Fonksiyonu		$F(\text{Net}) = \begin{cases} 1 & \text{if Net} > \text{Eşik Değer} \\ 0 & \text{if Net} \leq \text{Eşik Değer} \end{cases}$	Gelen Net girdinin belirlenen bir eşik değerin altında veya üstünde olmasına göre hücrenin çıktısı 1 veya 0 değerini alır.
Sigmoid Aktivasyon Fonksiyonu		$F(\text{Net}) = \frac{1}{1 + e^{-\text{Net}}}$	Sigmoid aktivasyon fonksiyonu sürekli ve türevi alınabilir bir fonksiyondur. Doğrusal olmayışı dolayısıyla yapay sinir ağı uygulamalarında en sık kullanılan fonksiyondur. Bu fonksiyon girdi değerlerinin her biri için 0 ile 1 arasında bir değer üretir.
Tanjant Hiperbolik Aktivasyon Fonksiyonu		$F(\text{Net}) = \frac{e^{\text{Net}} + e^{-\text{Net}}}{e^{\text{Net}} - e^{-\text{Net}}}$	Tanjant hiperbolik fonksiyonu, sigmoid fonksiyonuna benzer bir fonksiyondur. Sigmoid fonksiyonunda çıkış değerleri 0 ile 1 arasında değişirken hiperbolik tanjant fonksiyonunun çıkış değerleri -1 ile 1 arasında değişmektedir.
Eşik Değer Fonksiyonu		$F(\text{Net}) = \begin{cases} 0 & \text{if Net} \leq 0 \\ \text{Net} & \text{if } 0 < \text{Net} < 1 \\ 1 & \text{if Net} \geq 1 \end{cases}$	Gelen bilgilerin 0 dan küçük-eşit olduğunda 0 çıktısı, 1 den büyük-eşit olduğunda 1 çıktısı, 0 ile 1 arasında olduğunda ise yine kendisini veren çıktılar üretilebilir.
Sinüs Aktivasyon Fonksiyonu		$F(\text{Net}) = \sin(\text{Net})$	Öğrenilmesi düşünülen olayların sinüs fonksiyonuna uygun dağılım gösterdiği durumlarda kullanılır.

Şekil 2.6 YSA Transfer (Anonim 2021b)

2.4 Derin Öğrenme

Suni sinir ağlarının gelişmiş bir yaklaşımı olarak tanımlanmaktadır. Tek katmanlı suni ağlardan ziyade çok katmanlı suni ağların oluşumuyla alakalıdır. Sonrasında ise derin öğrenme işlemleyicilerin oluşumuna doğru bir gelişim göstermiştir. Suni sinir ağlarının ana hattını oluşturan insan beyni matematiksel modeli, çok fazla yerde kullanılmasına vesile olmuştur. Sinyal işleme, ses işleme, gruplandırma, tanı ve teşhis alanları örnek olarak verilebilmektedir. Günümüzde pek çok işlem bilgisayarlar vasıtasıyla yapılmaktadır. Bu da iş gücü, zaman ve maliyet açısından oldukça tasarruf sağlamaktadır. Bahsi geçen tasarruf sürecinde ise yapay sinir ağları aktifleşerek gerçek geri bildirimler vermiş ve kabul edilmiştir. Birçok teknoloji şirketi suni sinir ağlarıyla ilgilenmektedir. Bunun yanı sıra ortaya çıkardıkları yapay sinir ağı kümeleriyle bütünleşmiş ürünler kullanmaktadırlar. (Krizhevsky vd. 2012)

Geoffrey Hinton ve Ruslan Salakhutdinov'ın kaleme aldığı makalede ilk defa derin öğrenme işlemleyicilerinden bahsedilmiştir. (Hinton vd. 2006) İlerleyen zamanlarda ise bu kavram çok daha derin bir şekilde incelenmiş ve farklı detaylar ortaya çıkmıştır. (Goodfellow vd. 2016)

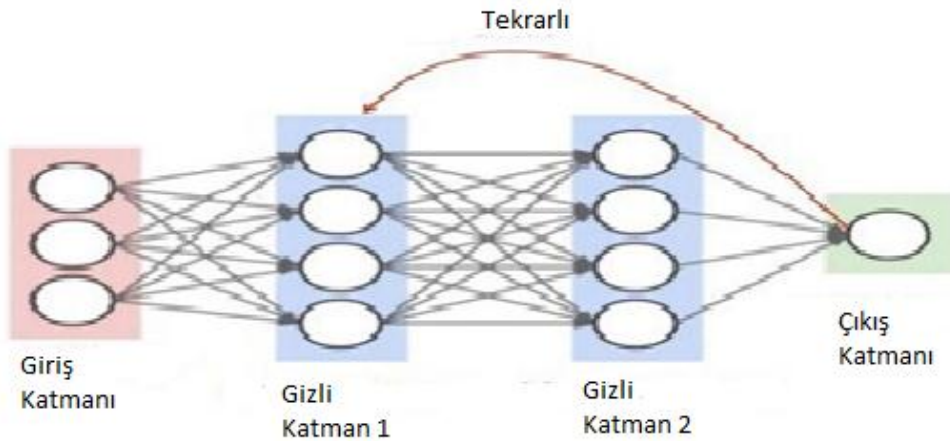
Bahsi geçen işlemleyicilerde inşa edilen çok katmanlı bir suni sinir ağı yapısı bulundurulur. Sözü edilen yapıda görüntüye ilişkin nitelik sonuçları sinir ağı tabakaları içerisinde oluşturulmaktadır. Tabakalı yapı esas özellikleri gizil tabakada çıkararak ağırlık değerliklerini öne sürmektedir. Görüntünün sahip olduğu kenar ve nitelikler konvolüsyon işlemi ile belirlenmektedir. Ağda yer alan öteki tabakalar bu nitelikleri alarak bir sonraki alt tabakaya aktarmaktadır. İlk tabaka girdi tabakası, son tabaka ise sınıflandırma katmanı olarak nitelendirilmektedir. Ara tabakadaki işlemlerde ise konvolüsyon, maxpooling, dropout, normalizasyon, Relu, Softmaxlayer, Fullconnectedlayer gibi katmanlar yer almaktadır. (Schmidhuber 2015) (Guo vd. 2016)

Algoritma Adı	Başarılı Kullanım Alanları
RNN	Konuşma Tanıma, Borsa Tahmini, Film Altyazı Tahmini
LSTM	Metin Üretimi, El Yazısı Tanıma
RBM	Konuşma İyileştirme İşlemi
DBN	Görüntü Tanımlama, Video Dizileri, Hareket Yakalama, Konuşma Tanıma
DAE-AE	Doğrusal Olmayan Boyutların Azaltılması, Tavsiye İnternet Motorlarının Tasarımı, Özellikleri Sınıflandırma, Görüntü Tanımlama
CNN	Yüz Tanımlama, Ses Tanımlama, Dil İşleme, Hastalık Tanıma

Şekil 2.7 Yapay Sinir Ağları Algoritmalarının Kullanım Alanları

2.5 Tekrar Eden Sinir Ağları (RNN)

Temel Yinelen Ağ (Simple Recurrent Network-SRN), Jeff Elman aracılığıyla geliştirilmiştir. Jeff Elman'ın tasarladığı simülasyonun konusu ise cümle yapısıdır. Burada yararlanılan bütün kelimeler için saklı şablonların üzerinde yaklaşık örüntü gruplama neticesinde ad ve eylem grupları düzgün bir biçimde ayrılmıştır. Tekrarlayan Sinir Ağları, bir adet giriş tabakasına, iki adet saklı tabakaya ve bir adet de çıkış tabakasına sahiptir. Bahsi geçen bu tabakaların hepsi birbirinden bağımsız bir sistemle çalışmalarını sürdürmektedir. Bu sebeple her bir tabakanın farklı bir görevi bulunmaktadır. (Şeker vd. 2017)



Şekil 2.8 RNN Katman Yapısı

Tekrar eden sinir ağlarında kullanılan matematiksel algoritması Şekil 2.9’da gösterilmiştir.

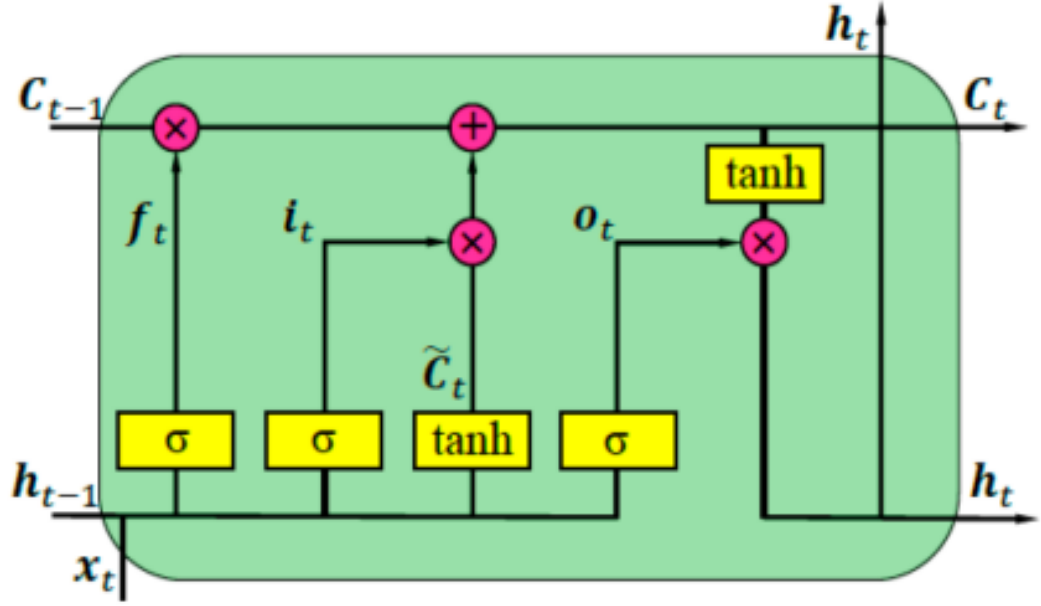
$$\mathbf{h}_t = \phi(W\mathbf{x}_t + U\mathbf{h}_{t-1}),$$

Şekil 2.9 Tekrar Eden Sinir Ağlarında Kullanılan Matematik Algoritması

$\mathbf{h}_t$, t zamanındaki gizli katmanın neticesidir. $\mathbf{x}_t$ girdisi ise W ağırlık değeriyle çarpılarak bir sonuç elde edilir. t-1 zamanında, içerik bölgesinde tutulan $\mathbf{h}^*(t-1)$ alanının değeriyle U ağırlık bilgisi çarpılır ve $W*\mathbf{x}_t$ ile toplamı hesaplanır. Bu alanda kütle matrisi, anlık ve bir önceki değerin neticeye etkisi değerlendirilir. İlgili işlemler neticesinde oluşan kusur hesaplanır ardından geri yayılım ile yeni kütle bilgileri yeniden şekillendirilir. Geri yayılım süreci önemsenmeyecek derecede azalana dek sürer. $W*\mathbf{x}_t + U*\mathbf{h}_{(t-1)}$ toplamı transfer fonksiyonuna dahil edilir. Bu sayede fazla büyük ya da fazla küçük değerler makul bir alalıkta yer alabilir. Böylelikle doğrusal olmayan bir çıktı sağlanmış olur(Anonim 2021c).

2.6 Uzun Kısa Süreli Hafıza Ağları (LSTM)

Uzun Kısa Süreli Hafıza Ağları yapısı verinin hücre içi ya da dışına çıkışını kontrol eden kapılardan meydana gelmektedir. Güncel verilerin hafızaya gireceği vakti kontrol eden unsur girdi kapılarıdır. Bunun yanı sıra hali hazırdaki verinin hatırlanmamasını, yeni bilgilerin ise anımsanmasını unutma-hafıza kapısı kontrol etmektedir. Çıkış kademesi ise hücredeki verinin ne zaman kullanılacağını kontrol eder. (Anonim 2022b)



Şekil 2.10 LSTM mimarisi

LSTM mimarisi, Şekil 2.10’da gösterildiği gibi art arda bloklardan yapıların meydana gelir. İçeriğinde kullanılan unut, girdi ve çıktı olacak şekilde 3 farklı katmana sahiptir. Mimaride ilk olarak girdi olarak X_t ve h_{t-1} bilgilerinden faydalanarak silecek bilgilere karar verilir.

Unut tabakasında (f_t) formül eşitlik 1 kullanılarak elde edilir ve aktarım fonksiyonu yerine sigmoid fonksiyonu tüketilir.

$$f_t = \sigma (W_{f,x} * X_t + W_{f,h} * h_{t-1} + b_f) \quad (1)$$

Bir sonraki basamakta girdi tabakası çalışır, veriler seçilir ve öncelikle (i_t) formül (2) kullanılarak sigmoid fonksiyonu ile veriler güncel hale getirilir. Ardından formül (3) ile ortaya çıkarılacak yeni veri aday veriler tanh fonksiyonu kullanılarak seçilir.

$$i_t = \sigma(W_{i,x} * X_t + W_{i,h} * h_{t-1} + b_i) \quad (2)$$

$$\tilde{C}_t = \tanh(W_{c,x} * X_t + W_{c,h} * h_{t-1} + b_c) \quad (3)$$

Eşitlik 4 ile tarafından yeni veriler belirlenir.

$$C_t = C_{t-1} * f_t + i_t * \tilde{C}_t \quad (4)$$

Çıktı katmanında eşitlik 5 ve 6 ile çalışılarak çıktı verileri belirlenir.

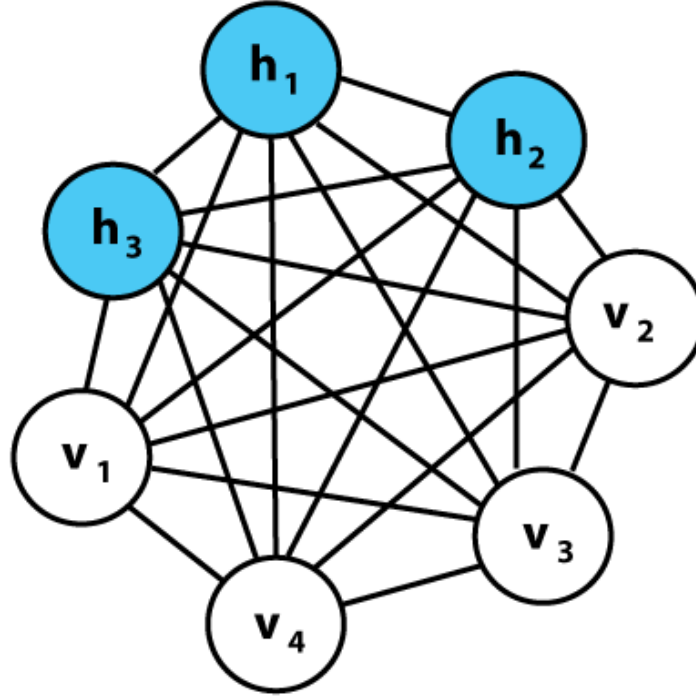
$$o_t = \sigma(W_{o,x} * X_t + W_{o,h} * h_{t-1} + b_o) \quad (5)$$

$$h_t = o_t * \tanh(C_t) \quad (6)$$

Yukarıdaki proses tekerrür ettirilerek kütle parametreleri (W) ve bias parametreleri (b) doğru eğitim değerleri ile LSTM çıktı değerleri arasındaki farkı azaltacak biçimde çalışır. (Anonim 2021d).

2.7 Sınırlı Boltzmann Makineleri (RBM)

Boltzmann makinesi, kümeleme, regresyon ve nitelik öğrenimi işlerini yapmaktadır. Girdi bilgi seti üzerindeki istatistiksel dağılımını öğrenebilmektedir. Bu makineler, tahmini yineleyen bir sinir ağı tipidir. (Anonim 2022c)



Şekil 2.11 Boltzmann Makinesi (Anonim 2022c)

Gizli değişkenler dahil edildiğinde, girdi özelliklerinin marjinal dağılımı, görünür özelliklerin ve her bir gizli değişkenin birleşme olasılığının toplamıdır. $H = (h_1, h_2, \dots, h_n)$ n tane gizli değişken içerir.

Gizli değişken tabanlı modellerin tanımlanması için ilk ve dikkat edilmesi gereken adım günlük olabilirlik fonksiyonudur.

$$\nabla_{\theta} \mathcal{L} = - \sum_h p(h|x) \nabla_{\theta} E(x, h) + \sum_{x,h} p(h, x) \nabla_{\theta} E(x, h)$$

Şekil 2.12 Günlük olabilirlik fonksiyonu

RBM'nin önemli bir özelliği, aynı katmandaki düğümler arasında bağlantı olmamasıdır. Gizli değişkenler, verilen girdi özelliklerinde koşullu olarak bağımsızdır ve bunun tersi de geçerlidir. RBM de gizli değişken tabanlı model tanımlanmasından sonraki adım ise enerji fonksiyonu tanıımıdır. Enerji fonksiyonu tanıımı şekil Y deki gibidir.

$$p(h|x) = \prod_{i=1}^n p(h_i|x) \quad ; \quad p(x|h) = \prod_{j=1}^m p(x_j|h)$$

Şekil 2.13 Enerji Fonksiyonu tanıımı

Bir diğer adım ise model parametrelerine göre $E(x,h)$ 'nin türetilmesi adıımıdır. Gerekli türevler ve adımlardan sonra Şekil Z deki gibi bir formülü elde ederiz.

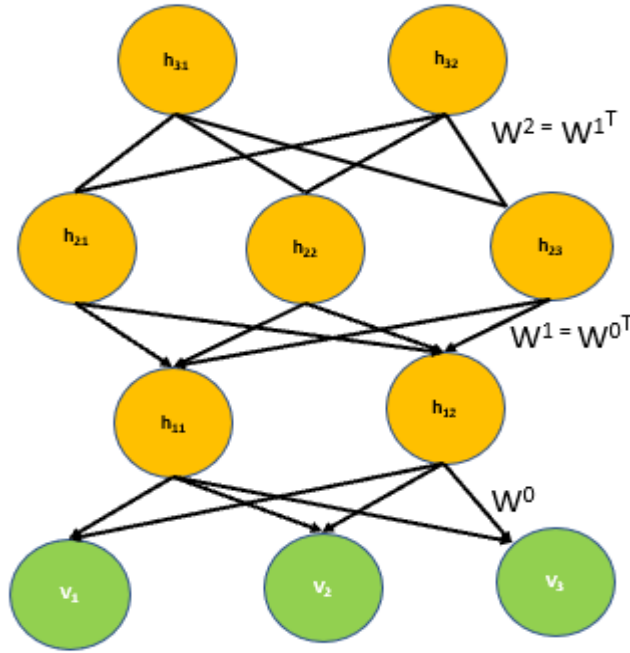
$$\nabla_{w_{ij}} \mathcal{L}_L = \langle x_j h_i \rangle_{\text{veri}} - \langle x_j h_i \rangle_{\text{model}}$$

Şekil 2.14 Model Parametrelerine göre $E(x,h)$ Fonksiyonu (Anonim 2021e)

2.8 Derin İnanç Ağları (DBN)

Temeli pek çok RBM'den oluşmaktadır. Derin İnanç Ağları çok sayıda alt kola ayrılmıştır. Bu her kolun kullanım hedefi farklılık göstermektedir. Bu sebeple farklı

sahalarda farklı biçimlerde çalışmalar yapılmaktadır. (Anonim 2022d)



Şekil 2.15 DBN Örneği (Anonim 2022d)

Tek RBM ye sahip DBN’lerin ağırlık güncelleme denklemi formülü Şekil 2.15’de gösterilmiştir. Eğer değilse Şekil 2.16’daki gibi hesaplanır.

$$w_{ij}(t+1) = w_{ij}(t) + \eta \frac{\partial \log(p(v))}{\partial w_{ij}}$$

Şekil 2.16 Ağırlık Hesaplama Formülü(Tek RBM)

$$\Delta w_{ij} \propto \langle v_i h_j \rangle_{\text{data}} - \langle v_i h_j \rangle_{\text{reconstruction}}$$

Şekil 2.17 Ağırlık Hesaplama Formülü(Çoklu RBM)

DBN’lerde kullanılan sigmoid fonksiyonu formülü şekilde 2.17’deki gibi gösterilmiştir.

$$p(h_j = 1 | \mathbf{V}) = \sigma(b_j + \sum v_i w_{ij}).$$

Şekil 2.18 Sigmoid Fonksiyon Formülü

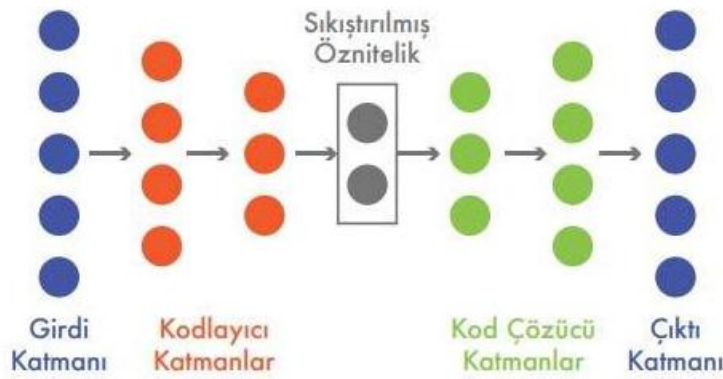
DBN’lerde kullanılan gizli birimler verildiğinde görünür birimleri paralel olarak güncellenmesi Şekil 2.19’deki gibi bir formül kullanılarak yapılmalıdır.

$$p(v_i = 1 | \mathbf{H}) = \sigma(a_i + \sum_j h_j w_{ij}).$$

Şekil 2.19 Güncelleme Formülü (Anonim 2021f)

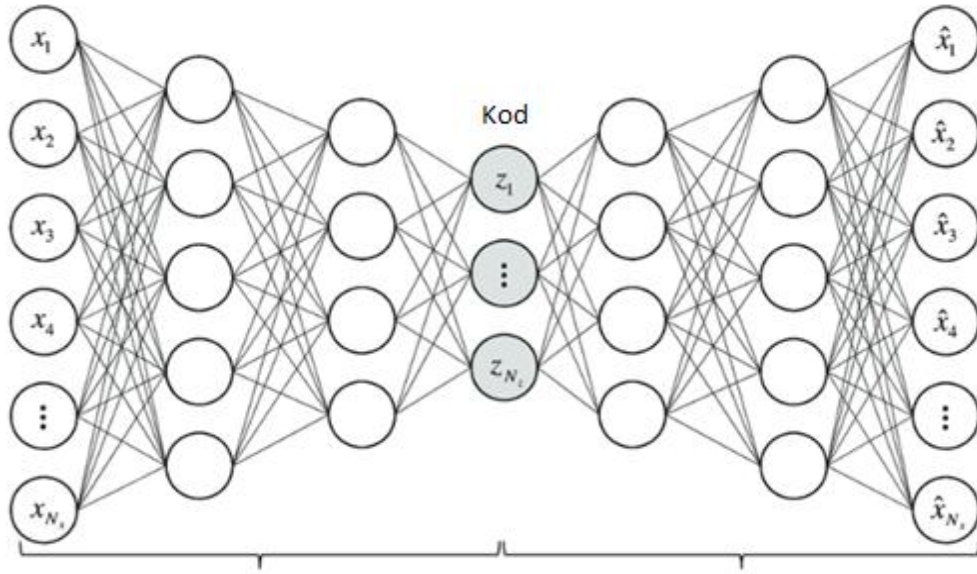
2.9 Derin Oto-Kodlayıcılar (DAE-AE)

Oto Kodlayıcılar (AE), Diabolo ağı olarak da bilinmektedir. Denetimsiz öğrenme için kullanılan hususi suni bir sinir ağıdır. Oto-Kodlayıcılar, giren veri tabakasındaki değerleri çıktı veri tabakasına nüshalanan bir sinir ağı olarak açıklanmaktadır. Özetle, girdi katmanındaki bilgiyi çıktı katmanında yeniden oluşturmaktadır. Girdi bilgilerinin bastırılmış görünümünden en başarılı niteliklerin kavranmasını amaçlayan ileri beslemeli bir sinir ağıdır. (Anonim 2022e)



Şekil 2.20 AE yapısı (Anonim 2022e)

Derin oto kodlayıcılar, her tabakadaki çıktı verilerinin sıralı tabakasının girişlerine bağlandığı oto kodlayıcıların çok katmanlarından oluşan bir sinir ağıdır.



Şekil 2.21 DAE Yapısı (Anonim 2022f)

2.10 Evrişimli Sinir Ağları Mimarisi

Evrişimli Sinir Ağları (ESA) Mimarisi derin öğrenmede kullanılan ana mimari olarak kabul edilir. Evrişimli sinir ağları mimarisi için ilk olarak konvolüsyon (Convulation) ve havuzlama (Pooling) katmanları yer alır. Sonrasında ise Tam Bağlı tabaka ortaya çıkar, ardından gelen ise Sınıflandırma tabakası olmalıdır. Giriş verileri evrişimli sinir ağlarına iletdikten sonra tüm katmanlardan adım adım geçtikten sonra eğitim süreci tamamlanır. Sonuç ile istenilen sonuç arasındaki fark kadar hata meydana gelir. Hatanın azaltılması için geriye yayılım algoritması yardımıyla bütün ağırlıkların değerleri güncellenir. Her geriye dönüşte değerler güncellenerek hataların azaltılması sağlanmaktadır.

2.10.1 Evrişimli Sinir Ağları Katmanları

Giriş Katmanı (Input Layer)

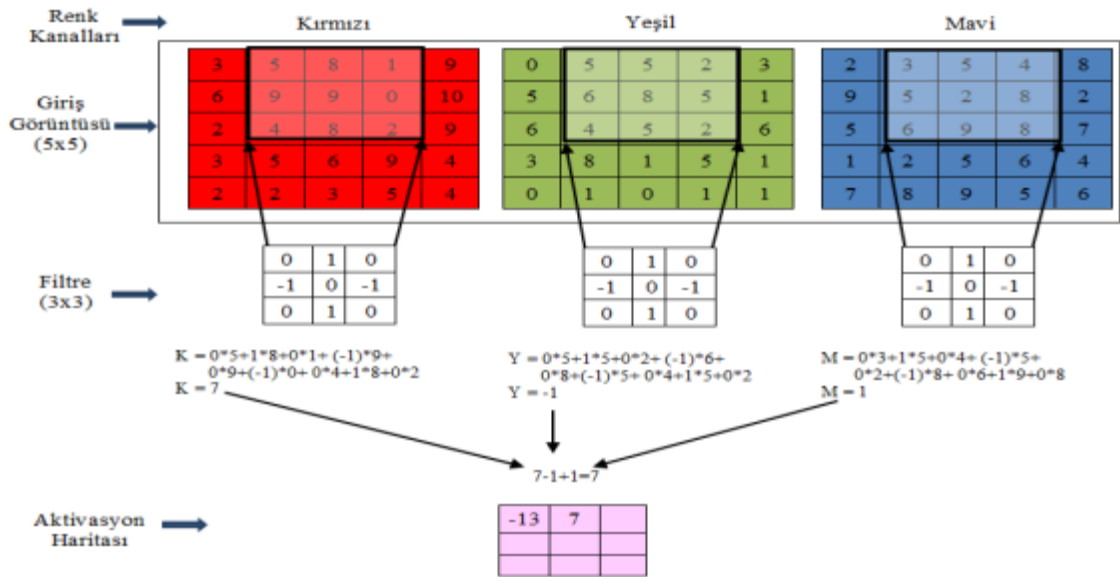
Evrişimli sinir ağlarının ilk katmanı olarak çalışmaktadır. Katmanda veri işlenmemiş olarak ağa iletilmektedir. Geliştirilecek modelin başarı oranı için verinin ölçütüne dikkat edilmelidir. Giriş verisinin ölçütünün büyük olması fazla bellek(ram) tüketimine yol

açar. Buna ek olarak ağın toplam eğitime süresini arttırır. Ayrıca her bir verinin test edilme süresine uzatabilir. Bu durumlara rağmen ağ başarı oranını iyileştirebilir. Giriş verisinin boyutunun düşük seçilmesi hem fazla bellek kullanımını ve toplamda eğitimin fazla zaman almasını düşürebilir. Lakin oluşturulacak bu ağın derinliğinde düşme görülebilir. Ek olarak bu durumdan kaynaklı performansı beklenilenden az olabilir. Veri olarak özellikle veri analizinde ağ derinliğini artı ve eksi noktaları düşünerek seçmek önem arz eder. Ek olarak donanımsal tüketim maliyeti ve kurulacak başarılı ağ için optimize edilmiş boyut seçilmesi önerilir.

Konvolüsyon Katmanı (Convulation Layer)

Dönüşüm katmanı olarak bilinen bu katman evrişimli sinir ağlarının temelini oluşturur. Dönüştürme yöntemi belli bir süzgecin bütün şekil üzerinde gezdirilmesi sürecine dayanmaktadır. Bu nedenle süzgeçler katmanlı yapının ayrılmaz bir parçasıdır. Süzgeçler 2*2, 3*3, 5*5 gibi ölçülerde seçilebilir. Süzgeçler, giriş katmanından iletilen verilere konvolüsyon sürecinden geçerek çıkış görüntüsünü(verisini) meydana getirir. Bu süreç sonunda aktivasyon tasarımı ortaya çıkar. Aktivasyon tasarımı, tüm süzgeçlerin kendine mahsus niteliklerin yer aldığı bölgelerdir. Evrişimli sinir ağları'nın çalışması süresinde bu süzgeçlerin katsayıları, eğitim yığınındaki her öğrenme tekrarlamasına göre değişebilir. Bu süreçte sistem, verinin bölgelerinin önemli olduğu ortaya çıkar ve bu sayede niteliklerinin ortaya konulması sağlanır. Rastgele seçilen süzgecin biçim üzerinde gerçekleştirilmesi Şekil 2.22'de yer almaktadır. Şekil 2.22'den anlaşıldığı üzere girişte RGB renklerine sahip ve 5*5'lik matris olarak varsayılırsa giriş veri ölçütü 5*5*3 şeklinde görülür. Konvolüsyon yöntemini kullanmak için 3*3 boyutuna sahip süzgeçle beraber giriş verisi kullanarak sağ ve sol yönüne göre belirli Adım(Stride) ilerletilerek güncelleme yapılır. Bu güncelleme sırasında matrisin uç noktalarına gelindiğinde bir adım aşağı inip süreç tekrar devam edilir. Bu gezinme sürecinde şekil matrisinin tamamında tekrarlanır. Süzgeç katsayıları tüm renk kanalındaki renkler için renk bazlı değerlerle çarpılır. Daha sonrasında bu çapılan değerler ilgili değerler toplanır. Tüm kanallar üzerinden her bir üç kanal ile süreç tekrar yapılmasının ardından üç değer toplanarak Aktivasyon planı oluşturulur. Renk kanalı matrisine gerçekleştirilen süzgeç katsayılarının birbirinin aynısı olmaması muhtemeldir. İlgili süzgeç katsayılarındaki değişiklikler çalışmacılar aracılığıyla numuneleri elverişli

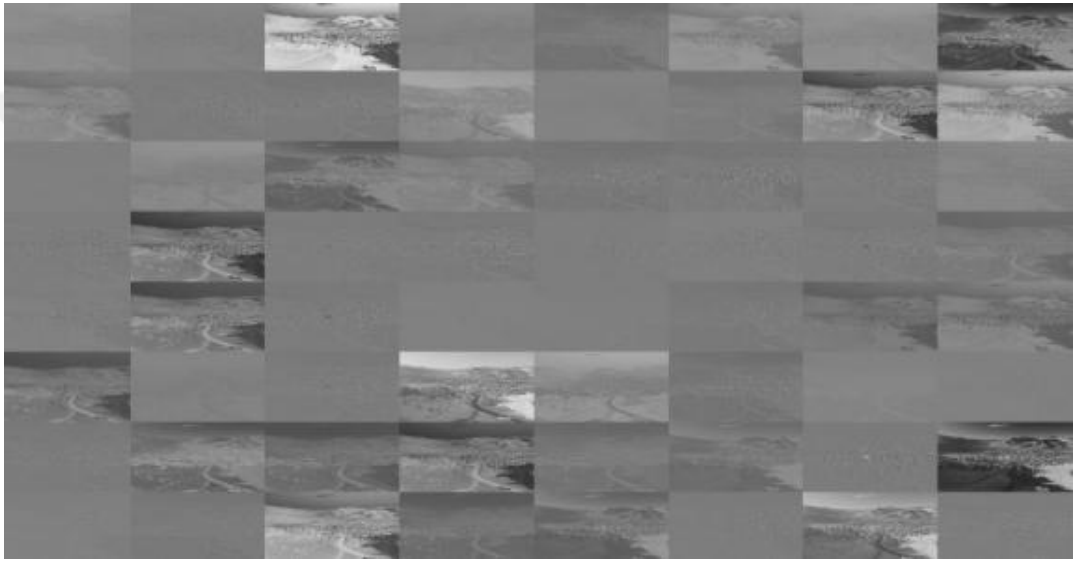
olacak biçimde ortaya çıkarılır. Aktivasyon planındaki veriler, çıkış ebatı ve giriş ebatı ortasında farklı olmayan yoğunluk aralığını yakalamak amacıyla normalleştirilir. İlgili normalleştirme sürecinde süzgeç katsayılarının toplamı, bulunan değerlere (renk kanalının tamamı için) bölünmesiyle sonuç üretilir. Süzgeç katsayılarının toplamı eğer sıfır olursa, Şekil 2.22’de yer alan resimdeki gibi işlem yapılmaz. Şekil 2.23 ve 24’te konvolüsyon katmanının evrişimli sinir ağları ile nasıl gösterileceği hakkında bilgi vermektedir. Ağda yer alan birinci katmanda (konvolüsyon) 3x3 değerlerinde ölçütlere sahip ve toplam 64 süzgeçten oluşan bir evrişimli sinir ağları oluşturulmuştur. Süzgeçlerin Şekil 2.23’te yer alan resim ile giriş görüntüsüne gerçekleştirilmesi işlemi sonucunda Şekil 2.2’de yer alan çıktılar meydana gelmiştir. Her bir süzgecin giriş görüntüsüne göre farklı etkilere sahip olduğunu Şekil 2.24’te çizilmiştir.



Şekil 2.22 5*5*3 Ölçütlere Sahip Görüntüye 3*3 Filtre Uygulanması



Şekil 2.23 Evrişimli Sinir Ağları için Giriş Görüntüsü

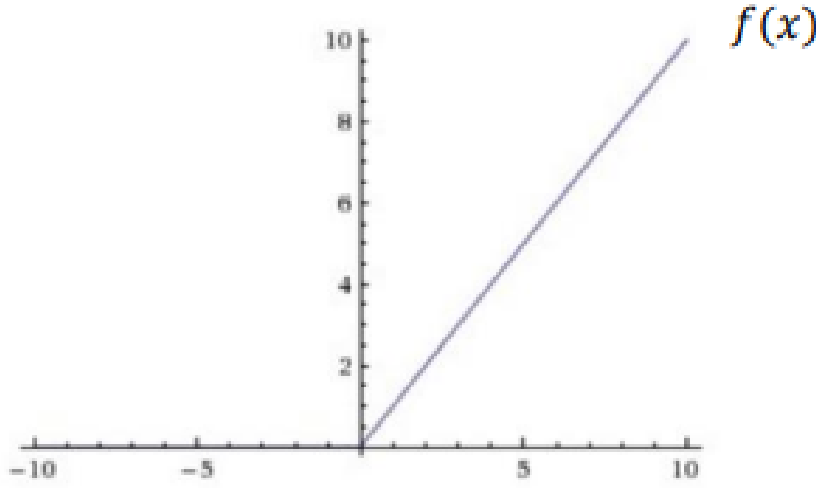


Şekil 2.24 Evrişimli Sinir Ağlarının Birinci Konvolüsyon Katmanından Sonra Oluşan Görüntüler

Doğrultulmuş Lineer Birim Katmanı (ReLU)

Konvolüsyon katmanından son yer alan katmandır. ESA nöron sonuçları için kullanılan doğrultucu elaman olarak adlandırılır. Matematiksel gösterimi Şekil 2.25 resminde çizilmiştir. Aktifleştirme katmanı adında da kullanılır. Giriş verisine etki eder ve eski olarak sıfırdan küçük değer varsa bu değeri sıfır yapar. Konvolüsyon katmanındaki işlemlerin uygulanması sebebiyle ağ doğrusaldır. İlgili derin ağı doğrusal olmayan bir hale getirilmek istenirse ReLu katmanı gerçekleştirilmelidir. ReLu katmanı ağın öğrenmesini hızlandırır.

$$f(x) = \begin{cases} 0 & \text{eğer } x < 0 \\ x & \text{eğer } x \geq 0 \end{cases}$$



Şekil 2.25 Doğrultulmuş Lineer Birim Katmanı Çıkış Verisine Etkisi

ReLU katmanının kullanılması ve etkisi Şekil 2.26'da yer almaktadır. Şekil 2.26'da ise giriş görüntüsü, konvolüsyon katmanı görüntüsü ve ReLu katmanı görüntüleri yer almaktadır.



Şekil 2.26 Bir ESA Modelinde Konvolüsyon ve ReLu Katmanının Etkisi

Havuzlama Katmanı (Pooling Layer)

Bu katman çoğunlukla düzleştirilmiş doğrusal birim katmanının ardından gelir. Birincil hedefi, ardından gelen tabaka için başlangıç ölçüsünü azaltmaktır. Bu yöntem veride derinlik ebatına etki etmez. Havuzlama tabakası nihayetinde katmanda bilgi eksikliğine neden olur. Bu şekilde bir kayıp şebeke iki sebepten kaynaklı olarak yararlıdır. İlki, ardından gelen ağ tabakaları için daha eksik miktarda sayımlama külfeti oluşturur. Diğeri ise modelin ezberlememesini sağlar. Birinci adımda ortaya çıkarılan konvolüsyon prosesine benzer şekilde, havuzlama tabakasında da belirli süzgeçler

belirlenir. Süzgeçler şekilde belirli basamak atlamak değerine oranla dolaştırılarak şekildeki piksellerin en fazla orandaki değerleri ya da ortalamadaki değerleri elde ederek süreç işletilir. Normalde en yüksek miktardaki havuzlama, yüksek başarılı performans meydana getirmesi sebebiyle seçilir. Havuzlama işlemi, konvolüsyon tabakası neticesinde elde edilen süzgeç sayısında şekillerin tamamı için uygulanır. Evrişimli sinir ağlarında Havuzlama tabakası tercihe göre şekillenir. Birtakım mimariler bu süreci başarıyla tamamlayamaz. Havuzlama süreci işlemi ile ilgili örnekleme Şekil 2.27’de gösterilmiştir. Şekil 2.27’de başlangıç şekil ebatı 4*4 ve süzgeç ebatı 2*2 olacaktır. Tek basamak kaydırılarak elde edilen şeklin boyutu 3*3 olarak elde edilir. Havuzlama süreci neticesinde elde edilen şeklin ebatı Eşitlik 2’ye göre hesaplanır.

$$\text{Elde edilen şeklin boyutu} = G_2 \times Y_2 \times D_2 \quad (2)$$

$$G_2 = (G_1 - F) / A + 1 \quad (3)$$

$$Y_2 = (H_1 - F) / A + 1 \quad (4)$$

$$D_2 = D_1 \quad (5)$$

G_1 = Başlangıç şekil ebatının genişliği

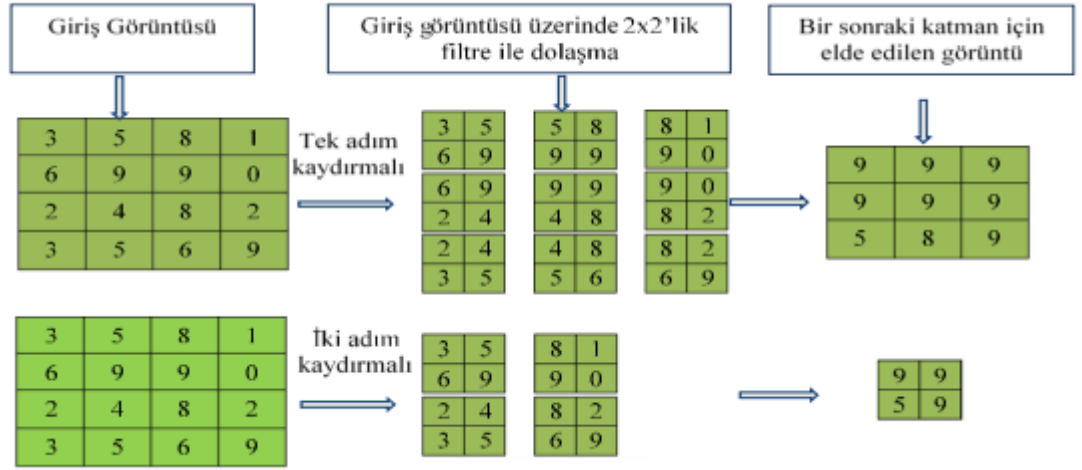
Y_1 = Başlangıç şekil ebatının yüksekliği

D_1 = Başlangıç şekil ebatının derinliği

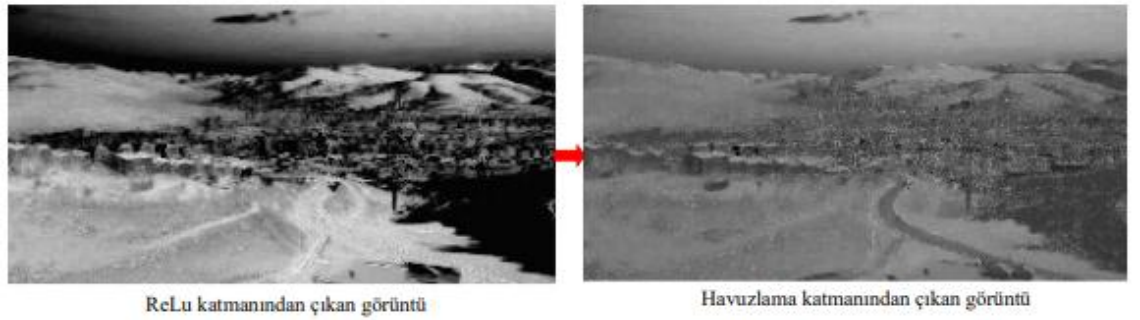
F = Süzgeç boyutu

A = Basamak adeti

Havuzlama prosesinde genellikle F ve A değerleri 2 olarak tercih edilir. Havuzlama tabakasının güncel bir resme uygulanması aşağıdaki şekilde gösterilmiştir. İlk şekil soldaki şekil düzleştirilmiş doğrusal birim katmanı neticesinde alınmış, üçüncü şekilse havuzlama tabakası neticesinde alınmıştır.



Şekil 2.27 5*5'lik Başlangıç Resmine 2*2 Filtre ile 1 ve 2 Basamak Kaymalı En Yüksek Havuzlama Süreci Uygulaması



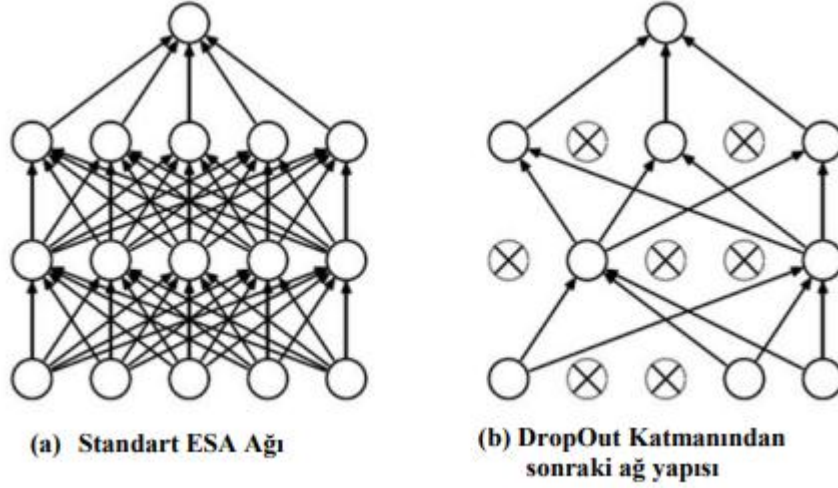
Şekil 2.28 Evrişimli Sinir Ağları Modelinde Havuzlama Tabakasının Evvelki Tabakadan İletilen Resme Uygulanması

Tam Bağlantılı Katman (Fully Connected Layer)

ESA'da kullanılan mimarilerde sıklıkla gelen konvolüsyon katmanı, ReLu katmanı ve havız katmanı sonrasında kullanılan bir katmandır. İlgili katmanda katman sayısı değişiklik gösterebilir ve diğer önceki katman alanlarını kullanır. Evrişimli sinir ağları mimarisinde bitiş katmanının üretmiş olduğu matris ölçütleri $25*25*256=160000*1$ ilgili katmanın ölçütü de $4096*1$ belirlenirse $160000*4096$ ağırlık matrisi meydana gelir. İlgili katmanda ilk ölçütte yer alan nöronlar ile diğerleri eşleşir ve bu nedenle bu ismi almıştır.

DropOut Katmanı

ESA'nda bazen ezberleyen ağlar ile karşılaşılabilir. Bu durumu birden fazla sebebi vardır. Özellikle büyük veriler ile eğitim gerçekleştirildiği zaman rastlanabilir. Ezberlemeyi engellemek veya azaltmak için dropout katmanı kullanılır. İlgili katmanda ağdaki bazı düğümler kaldırılarak gerçekleştirilir. Şekil 2.29 (a)'da ağın değiştirilmemiş hali varken, (b)'de ilgili katman gerçekleştirildikten sonraki hali vardır.



Şekil 2.29 DropOut Katmanının ESA ile Gerçekleştirilmesi

Sınıflandırma Katmanı(Classification Layer)

Bir sonraki katman olarak sınıflandırma katmanı gelir ve katmanda birbirine benzemeyen sınıflandırıcılar kullanılır. Mimarilerde sınıflandırma yapmak için kullanılır. Çıkışında elde edilecek çıktı, sınıflandırmada kullanılan eleman sayısına denktir. Bir önceki katmanın çıktısı 4096 ise, değerlere göre 4096×15 'lik bir matris ağırlık matrisi olarak kabul edilir. bu çıkış değerine göre sınıflandırma katmanı için 4096×15 ağırlık matrisi elde edilir. Softmax en çok kullanılan ve yaygın olan sınıflandırıcıdır. (İnik ve Ülker 2017).

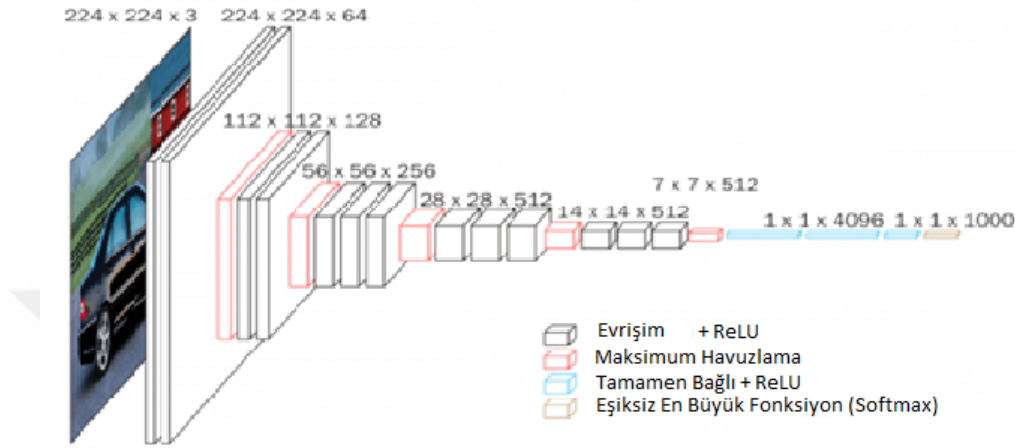
2.11 Konvolüsyonel Sinir Ağları (CNN)

CNN, bir veya birden fazla konvolüsyonel tabakalardan, aşağı örnekleme tabakasından ve olağan çok tabakalı bir sinir ağı gibi bir ya da birden fazla tabakalardan oluşmaktadır. Konvolüsyonel Sinir Ağlarının, eşit ölçüdeki gizli birimle tümüyle bağlı

ağlardan daha eksik miktardaki eğitime ve değişkene ihtiyaç duyulması avantaj olarak sayılmaktadır.

En çok kullanılan CNN modelleri aşağıdaki gibidir.

2.11.1 VGG16



Şekil 2.30 VGG16 Katmanları

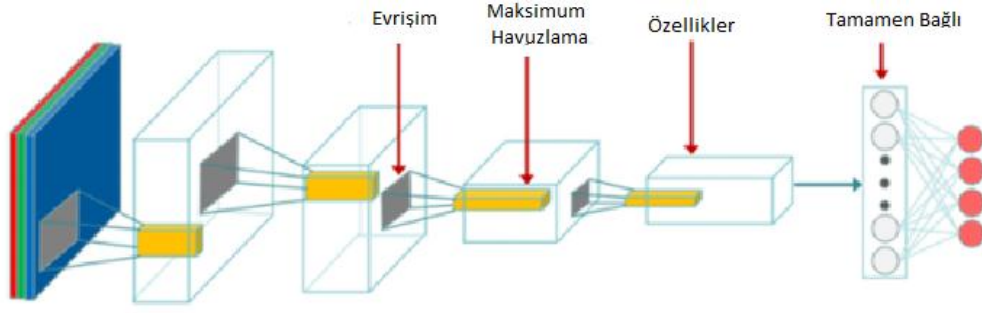
Bütün içi beyaz, kenarları siyah olan dikdörtgenler, onarılmış doğrusal birim (veya ReLU) olan doğrusal olmayan aktivasyon işleviyle beraber evrişim tabakalarını sembolize etmektedir. (Anonim 2022g)

Kenarları siyah olan 13 adet, kırmızı olan 5 adet dikdörtgen mevcuttur. Özetle 5 azami havuz tabakası ve 13 evrişim katmanı yer almaktadır. Bütün bunların beraberinde, karşılıklı olarak bütünüyle bağlı 3 tabakayı sembolize eden kenarları mavi 3 adet dikdörtgen bulunmaktadır. (Anonim 2022g)

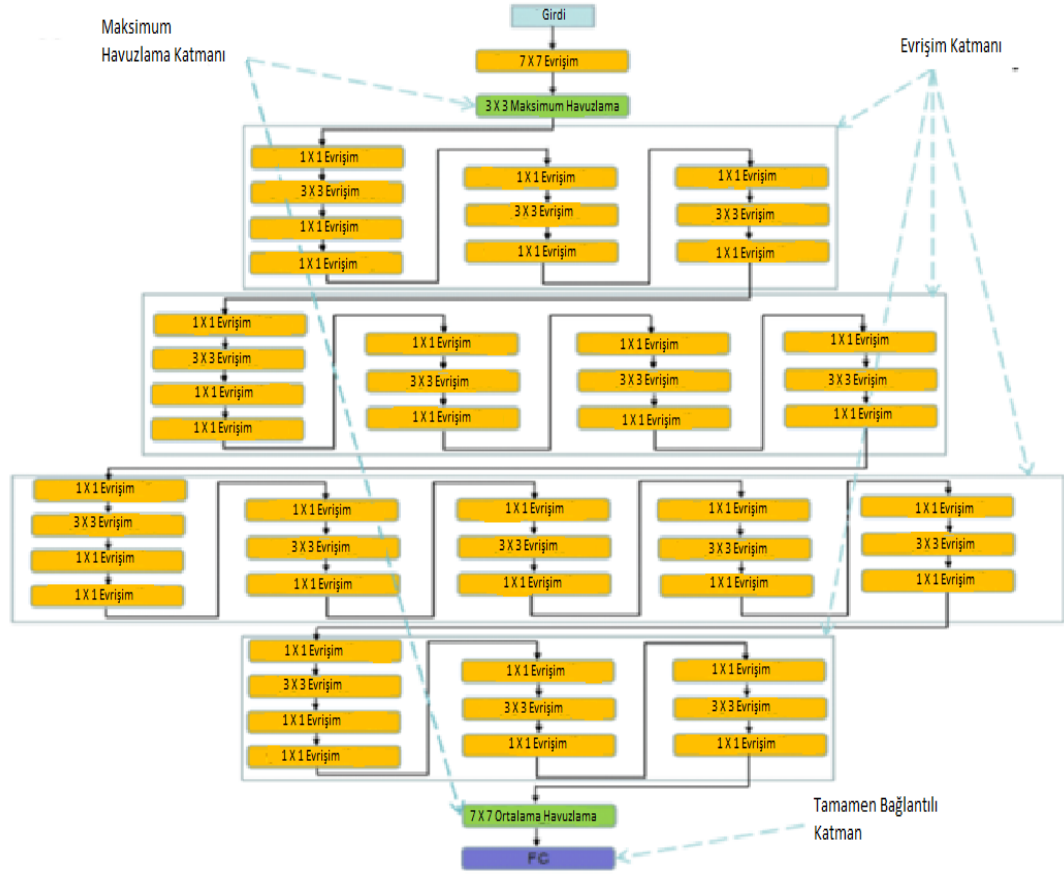
Ayarlanabilir parametreleri mevcut tabaka sayısı 16'dır. Sözü edilen tabakaların 13'ü evrişim tabakası, 3'ü ise bağlantılı tabakalardır. Bu sebeple VGG-16 ismini almıştır.

Imagenet veri kümesinde görüntü kategorisi başına 1000 çıktıya sahip bir softmax katmanı var. (Anonim 2022g)

2.11.2 ResNet50



Şekil 2.31 Resnet50 Katman Yapısı (Anonim 2022h)



Şekil 2.32 Resnet50 Detay Mimarisi (Anonim 2022h)

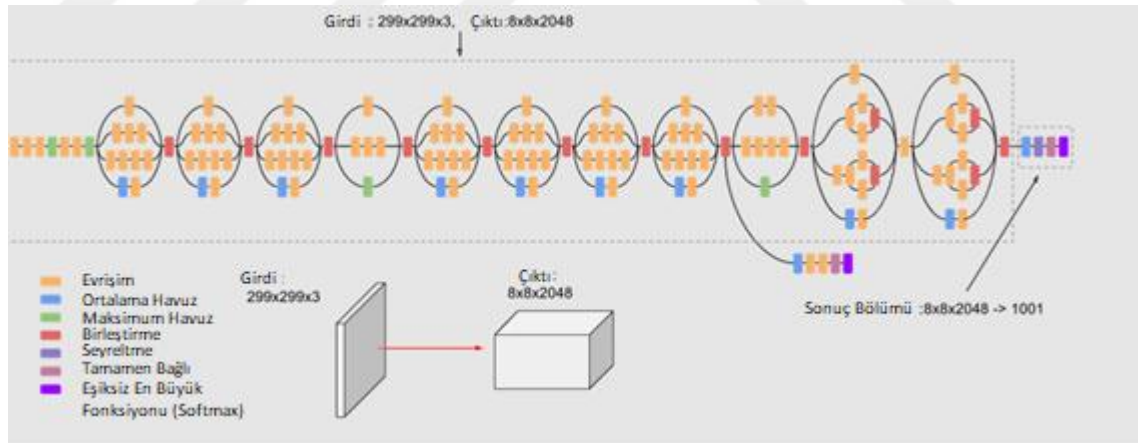
Görselde gösterildiği gibi ResNet'in tanıttığı bu yapı yardımıyla aşırı derin sinir ağlarını eğitime olanağı bulundu. Böylece ağır çok sayıda tabaka barındırabildiği ve kusursuz verim alınabildiğine tanık olunmuştur. (Anonim 2022h)

ResNet'in öncelikli olarak görüntü tanıma vazifelerinde uygulanmıştır. ResNet50 mimarisinde, azami havuz tabakaları, çoklu konveksiyon tabakaları ve tam bağıntılı bir tabaka içermektedir. (Anonim 2022h)

2.11.3 Inception V3

ImageNet bilgi takımında %75 üzerinde daha çok bir doğruluk payına sahip olduğu bilinen ve geniş kitlelerce kullanılan bir tür resim algılama modelidir. Denklemler, azami havuzlama, birbirine bütünüyle bağlı tabakalar dahil bakışumlu ve bakışumsuz yapı taşlarından oluşmaktadır. 48 tabaka derinliğe sahiptir. Aynı zamanda görüntüleri 1000 nesne kategorisinde sınıflandırabilmektedir. Bu ağ pek çok farklı görüntüler için verimli nitelik gösterimleri öğrenmiştir. Birçok resim boyutu mevcuttur. Ayrıca giriş için 299X299 bir görüntü boyutu mevcuttur (Barratt ve Sharma 2018).

GoogleNet, Google tarafından geliştirilmiş, 2014 ILSVRC'yi kazanan bir CNN mimarisidir. Belirli zaman aralıklarıyla model güncelleme ve geliştirmeler aldı. Bu modelin dört versiyonu vardır. (Barratt ve Sharma 2018).



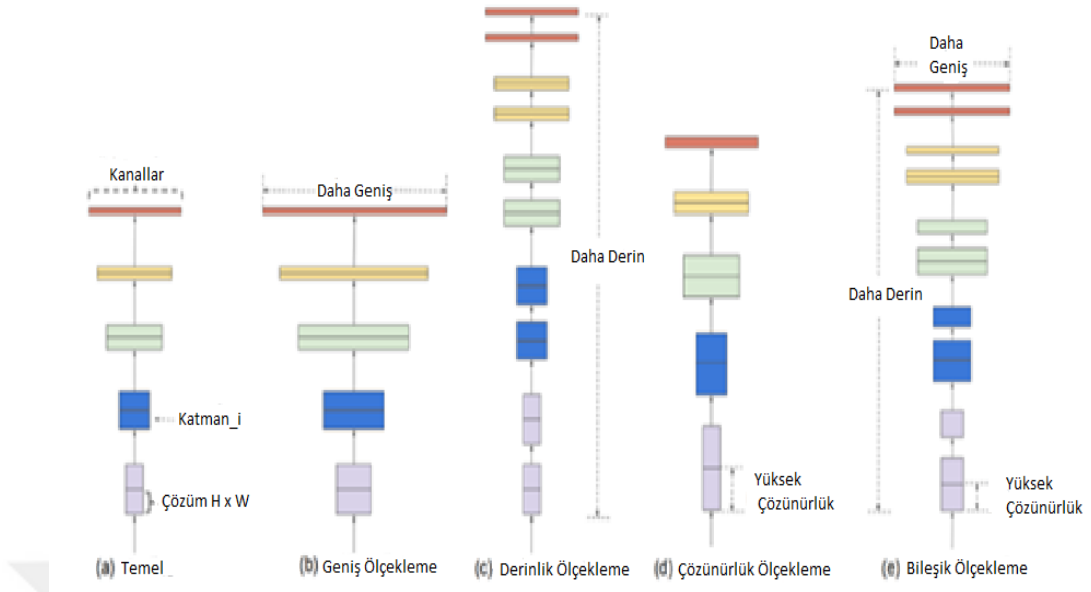
Şekil 2.33 Inception V3Mimarisi (Anonim 2022i)

2.11.4 EfficientNet

ImageNet veri grubundaki yer alan modellerle birlikte doğru orantılı olarak başarı yüzdesi de artış göstermiştir. 2012 yılından itibaren ise karışık bir hale bürünmüştür. Fakat pek çok hesaplama zorluğu bağlamında tesir gösterememektedir. EfficientNet modeli, gruplama sorunda 66 M parametresiyle %84,4 oranıyla doğruluk payı

göstermektedir. Bu sebeple son teknoloji modeller arasında yer edinmektedir. CNN modelleri arasında sınıflandırmak mümkündür. B0 ila B7 arasında 8 modelden oluşmaktadır. Model numarası artış gösterdikçe hesap edilen parametre sayısı pek artış göstermez. Ancak doğruluk oranı yadsınamaz bir şekilde büyümektedir. Öteki CNN modellerinden ayrı olarak ReLU etkinleştirme fonksiyonu yerine Swish isimli yeni bir fonksiyon kullanılmaktadır (Tan ve Le 2019).

Nispeten küçük modellerle daha işlevli modeller geliştirmek, derin öğrenme mimarilerinin amacı olarak sayılmaktadır. EfficientNet modelini öteki son teknoloji modellerden ayıran belirgin nitelikler bulunmaktadır. Genişliği, derinliği ve çözünürlüğü denk bir şekilde derecelendirirken modelin küçülmesiyle daha verimli neticeler kazanılmaktadır. Bileşik ölçekleme metodundaki başlangıç, durağan bir köken sınırlaması altında temel ağı çeşitli ölçeklendirme boyutları arasındaki bağıntıyı keşfetmek için bir ızgara aramaktır. Böylece derinlik, genişlik ve çözünürlük için elverişli bir ölçekleme etkeni saptanır. Bulunan katsayılar sonrasında temel ağı amaçlanan ağı ölçeklendirmek için uygulanır (Tan ve Le 2019).

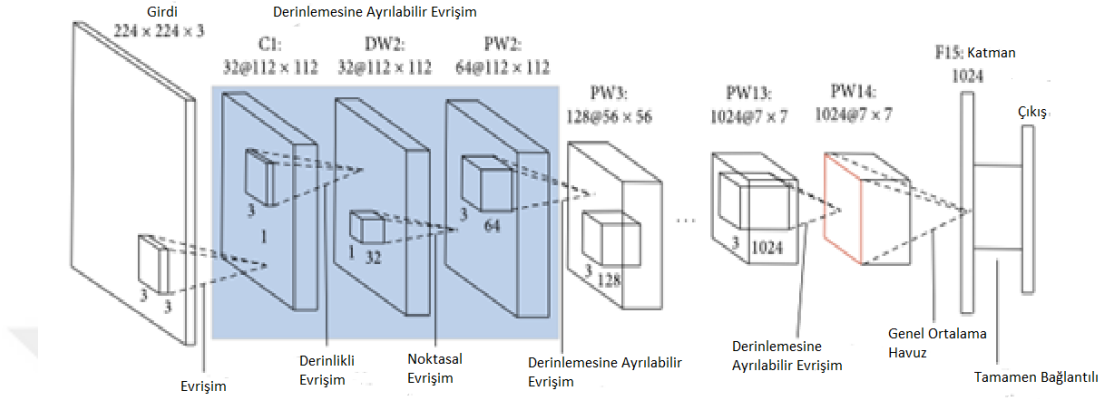


Şekil 2.34 EfficientNet Mimarisi (Anonim 2022j)

2.11.5 MobilNet

MobilNet modeli, mobil ve gömülü görme işlemleri için tanıtılmıştır. Derinlik bağlamında ise temeli ayrılabilir evrişimlere kadar uzanmaktadır. Bu model, oldukça yalın bir sinir ağı modeli olarak tanımlanmaktadır. Pek çok kıyaslanan bilindik modellerin yanı sıra daha küçük boyutta ve hızlı performansa sahiptir. Derinlik bağlamında ayrılabilir evrişimlerde her girdi kısmına yalnızca bir adet filtre işlenir. Sonrasında ise nokta esaslı evrişim katmanlarında 1×1 evrişimler kullanılmaktadır. Bu şekilde derinlik esaslı tabakaların çıktısının doğrusal kombinasyonu kurulmaktadır. Derinlik bağlamında ayrılabilir tabakalar, karakteristik evrişim tabakalarının işlemlerini neredeyse kopyalamaktadır. Fakat daha seri ve ufak bir ayırımla (sıradan evrişim, çıktı niteliklerine filtre uygular ve bütünleştirir, fakat bu yöntem derinlik bağlamından ayrılabilir evrişimde iki farklı tabakaya bölünür, filtreleme ve bütünleştirme için iki ayrı tabaka) işlem uygulanır. Bahsi geçen yaklaşım, modelin boyutunu minimize eder ve hesaplama zorluk istemlerini düşürür. Softmax tabakasını pekiştiren tam bağlantılı tabaka haricinde bütün tabakaların ardından batchnorm ve ReLU doğrusal olmayan aktivasyon tabakası gelerek çıktıyı sınıflandırır. Aynı zamanda bu model, noktasal ve derinlikli evrişimli tabakalar dışında toplamda 28 tabakaya sahiptir. Modelin yapısı,

ayrılabilir evrişim uygulamasından yararlanmıştır. Bu sayede değişken sayısını, model güçlüğü ve işlem hacmini azaltmıştır. Sonuç olarak ise belirli zorluklara sahip gömülü ve mobil aygıtlarda işlevsel hale getirilmiştir (Howard vd 2017).



Şekil 2.35 MobilNet Mimarisi (Anonim 2022k)

2.11.6 DenseNet

DenseNet, en son edinilen DL teknolojilerinden biri olarak tanımlanmaktadır. Görsel nesne teşhisi sorunları için kullanılmaktadır. Yapısı esasta ResNet'e benzerlik göstermektedir. Ancak ayrıştığı yönler mevcuttur. ResNet, mimarisi bakımından bir önceki tabakayı bir sonraki tabakayla bütünleştiren bir toplama metodu kullanır. DenseNet ise bir önceki tabakanın çıktısını sonraki tabakayla bütünleştirir.

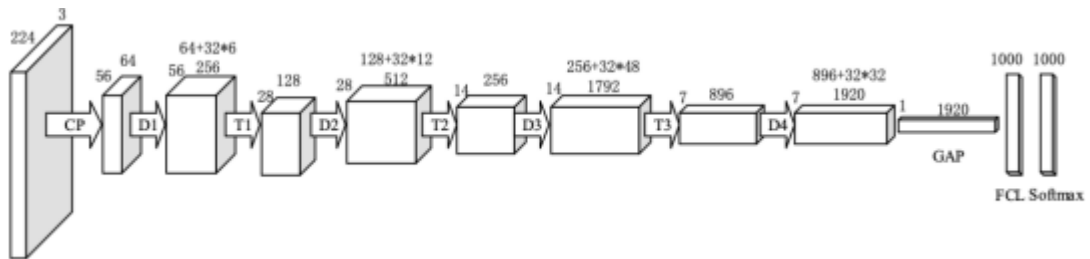
DenseNet'in ILSVRC testinin neticeleri, yoğun (dense) yapısının daha az değişkenle birlikte ResNet'le emsal bir performans sergilemektedir (Valev vd. 2018). DenseNet'in üç ayrı uyarlaması bulunmaktadır. Bunlar DenseNet-121, DenseNet-160 ve DenseNet-201'dir. Uyarlamalarda bahsi geçen sayılar modellerin ağındaki mevcut tabaka sayısını göstermektedir. DenseNet-201 ve DenseNet-160, 3x3 ve 1x1 tabaka adetleri haricinde DenseNet-121 ile benzerlik göstermektedir. DenseNet-201, sırası ile Dense Block 3 ve 4'te 48 ve 32 alt blok içerir (Wang ve Zhang 2020).

Wang ve Zhang (2020) vasıtasıyla, önceden eğitilmiş bilindik CNN modellerin kıyaslama verilerine dayanarak saptamada bulunulmuştur. Bu sayede DenseNet'in

başka modellere kıyasla daha üstün başarı gösterdiği belirlenmiştir. Aynı zamanda bu çalışma ile birlikte, gruplandırma hizmeti için DenseNet-201 modelinin öteki DenseNet modellerine göre daha yüksek başarı oranına sahip olduğu saptanmıştır. Bu sebeple bu tez çalışmasında tavsiye edilen hibrit model için DenseNet-201'in tercih edilme sebebi derin öznitelik çıkarma bağlamıdır.

DenseNet-201 modeli öncesinde eğitilmiş CNN modeli olarak tanımlanmaktadır. 201 tabaka derinliğine sahiptir. Aynı zamanda bir milyonu aşkın görüntü barındıran ImageNet bilgi tabanı kullanılmıştır. Görüntüleri 1000 grupta (mesela pek çok nesne ve hayvan) kategorize edilebilmektedir. DenseNet-201, $224 \times 224 \times 3$ görüntü giriş boyutundadır. İlk sıradaki 224 kısmı genişliği, diğer 224 kısmı yüksekliği ve 3 değeri ise derinliği (renk kanalları sayısı) temsil etmektedir (Peng vd. 2020).

Evrişimsel Sinir Ağlarında ağ, girdi görüntülerinin sıralı bir biçimini meydana getirir. Daha derindeki tabakalar, önceki tabakaların alt seviye niteliklerinden yararlanılarak meydana getirilen üst seviye nitelikleri kapsamaktadır. Çalışmada bilgi setinin örneklerine ulaşmak için ağın sonunda yer alan tam bağlantılı tabakadan önce yer alan küresel ortalama havuzlama tabakası “Global Average Pooling” (GAP) üzerindeki etkinleştirmeler kullanılmıştır. GAP, 1920 nitelik sağlamıştır. DenseNet-201'in genel yapısı şekil üzerinden örneklendirilmiştir.



Şekil 2.36 DenseNet-201 Mimarisi (Wang ve Zhang 2020)

2.12 Bitki Sınıflandırma

Bitkilerin yaşamdaki rolünün önemi yadsınamaz bir gerçektir. En mühim kaynaklardan sayılmaktadır. Bahsi geçen kaynakların yararlı bir biçimde ileriye götürülmesi hayati etkinliklerin önemi bağlamında zaruri bir konudur.

Bilhassa bazı bitkilerin himaye altına alınması ve hayat çemberinde yer edinmesi gerekmektedir. Bu bitkilerin adının duyulması insanların istikbali için önemli bir konu olarak yer edinmektedir. Bitki yapraklarının gruplandırılması bu bağlamda mühimdir. Bitkileri tasvir edecek nitelik ve biçimlerin çıkarılması, bitkiyi açıklayıcı faktörlerin saptanması gerekmektedir. Her bir grup için nitelikler saptanmalı ve o grubu simgeleyecek en öncelikli tercihler yapılmalıdır (şekil, kıvrım, boyut vs.) (Kulkarni vd. 2013).

Bilgisayarlı görü yöntemi, gruplandırma ve tanımlama işlemlerinde başarılı sayılmaktadır.



3. LİTERATÜR ARAŞTIRMASI

Xiaofei Chao ve arkadaşlarının çalışmasında makalesinde sararmış, kurumuş, gri nokta, kahverengi nokta, çevresi sarı için delikli ve sağlıklı yaprak gibi türler üzerinde toplamda 2970 yaprak verisi üzerinden eğitimler yapılmıştır. Veri sayısı az olduğundan ve veri dağılımı düzgün olmadığı için veri augmentation ile verileri düzenli hale getirilmiştir. Doğru dağılımlı toplamda 24,976 veri olarak veriler düzenlenmiştir. XDNet modeli ile 16 dönem ile 0.9835 gibi büyük bir başarı oranı yakalanmıştır. CNN mimarileri ile XDNet başarı oranları Şekil 3.1’de gösterilmiştir. Tezin yazıldığı çalışmada eğer veri augmentation ile veri sayısı artırılırsa ve model katmanında düzenlemeler yapılırsa başarının artacağı anlaşılmıştır (Chao vd. 2020).

Model	Giriş Boyutu	Toptan Boyut (Batch)	Eğitim Zamanı /Min	Parametreler	En İyi Başarı %	Ortalama Başarı %
Inception-v3	299	16	76.85	21,815,078	97.13	96.59
DenseNet-201	224	16	111.17	18,333,510	98.31	97.50
MobileNet	224	16	30.42	3,235,014	97.47	96.52
VGG-16	224	16	31.72	134,285,126	96.29	95.85
Xception	299	16	133.43	20,873,774	98.48	97.77
VGG-INCEP	224	32	144.27	148,714,950	97.64	97.20
XDNet (Our work)	224	16	86.67	10,159,518	98.82	98.35

Şekil 3.1 CNN Mimarileri ve XDNet Başarı Sonuçları

Peng Jiang ve arkadaşlarının çalışmasında ise sararmış, kurumuş, gri nokta, kahverengi nokta, çevresi sarı için delikli ve sağlıklı yaprak gibi türler üzerinde toplamda 2029 yaprak verisi üzerinden eğitimler yapılmıştır. Burada VGGNet-16 ve InceptionV3 bir arada kullanılarak en yüksek başarı yüzdesine ulaşılmıştır. Başarı olarak 0,9714 olarak hesaplanmıştır. Diğer mimarilere göre başarı sonuçları Şekil 3.2’de gösterilmiştir. Eğer hazır modeller kullanırken başarılı iki mimariyi birlikte kullanarak mimari oluşturulsaydı daha yüksek başarı oranı bu tez çalışmasında sağlanırdı. (Jiang vd. 2019)

Model	Giriş Boyutu	Başarı (%)
AlexNet	227	95.78
GoogLeNet	224	94.85
InceptionV3	232	95.49
ResNet-101	224	91.16
ResNet-50	224	92.43
ResNet-34	224	93.17
ResNet-18	224	93.64
VGGNet-16	224	96.10
VGG-INCEP(Our work)	224	97.14

Şekil 3.2 CNN Mimarileri ve VGG-INCEP Başarı Sonuçları

Bin Liu ve arkadaşlarının çalışmasında sararmış, kurumuş, kahverengi nokta, çevresi sarı için delikli ve sağlıklı yaprak gibi türler üzerinde toplamda 13.689 yaprak verisi üzerinden eğitimler yapılmıştır. Mimariler 15 dönem üzerinden eğitilmiştir. AlexNet, GoogLeNet, ResNet-20, VGGNet-16 gibi hazır mimarilere ek olarak ayrı bir öznel model oluşturulmuştur. Tüm eğitim için ortamlar aynı olacak şekilde çalışma yapılmıştır. Sırasıyla eğitim başarı sonuçları 91.19%, 95.69%, 92.76%, 96.32%, 97.62% olarak belirlenmiştir(Liu vd 2017). Özgün yazılan modelden diğerlerine göre daha iyi bir sonuç alındığı görülmüştür. Tez çalışmasından da iyi bir sonuç alınmasının sebebi veri sayısının fazla olması ve modelin daha iyi yaratılması olarak belirlenmiştir.

Saraansh Baranwal ve arkadaşlarının çalışmasında ise elma yaprağı kabuğu tabakası hastalığı, kurumuş, siyah noktalı ve sağlıklı yaprak olmak üzere 4 farklı kategoride toplamda 1526 yaprak verisi üzerinden eğitimler yapılmıştır. Train ve teste tüm verilerin 80%-20% oranla ayrılmıştır. Dönem olarak 40 belirlenmiştir. Optimizasyon algoritması olarak Adam kullanılmıştır. Öznel bir model oluşturulup eğitim yapıldığında 0,9854 gibi bir başarı alınmıştır (Baranwal vd. 2019). Tez çalışmasında dönem sayısı artırılıp teste ayrılan yüzde düşürülüp eğitilen veriye verilseydi başarı oranının artacağı anlaşılmıştır.

S.P. Mohanty ve arkadaşlarının “Görüntü Tabanlı Bitki Hastalıkları Tespiti için Derin Öğrenmeyi Kullanma” başlıklı makalesi (Mohanty vd. 2016), bu alanda yürütülen çeşitli araştırma türlerinin temeli olmuştur. Bunun Plant Village veri setindeki tüm sınıflar için son derece doğru sonuçların gözlemlendiği ilk makalelerden biri olduğu açıktır. Bu yazıda tasarlanan modeller, modern CNN mimarilerinin hünerlerini tanıttı. Ayrıca, burada kullanılan veri seti üç kategoriye ayrılmıştır – RGB görüntüler, Tek

renkli görüntüler ve Bölümlenmeli görüntüler. Her üç tür veriyi (biri birer) işlemek için ayrı yaklaşımlar kullanılmış ve %85,53 ile %99,34 arasında değişen nihai doğruluk elde edilmiştir.

Vijai Singh ve AK Mishra araştırmalarında (Singh ve Misra 2017) bitki hastalıklarının tespiti için çok ilginç bir yaklaşım ortaya koydu. Aynısını yapmak için, görüntü bölümlendirme kritik bir kısımdır. Genetik algoritma kullanarak görüntü sınıflandırma görevini yerine getirdiler. K-ortalama kümeleme, sınıflandırma amacıyla kullanılır. Genetik algoritma kullanılarak doğruluk %86,54'ten %93,63'e yükseldi. %95.71'lik en iyi doğruluk, sınıflandırma için Görüntü Bölümleme ve Destek Vektör Makinesi (SVM) ile birlikte Genetik algoritma kullanılarak elde edildi.

CNN, özellikle günümüz görüntü işleme yeteneklerindeki artış nedeniyle, görüntü sınıflandırması için yaygın olarak kullanılmaktadır. S.Sladojevic ve ark. (Sladojevic vd.2016), daha büyük, geliştirilmiş bir veri seti oluşturmak için Image Augmentation tekniği kullanıldı. CaffeNet kullanılarak on üç farklı bitki hastalığı türü sınıflandırılmıştır (Jia vd. Kasım), beş evrimsel katmana ve ardından üç tam bağlantılı katmana sahip bir derin CNN türüdür. Gerekli model düzenlemelerinden sonra %96,3 doğruluk elde etmişlerdir.

LG Nachtigall ve arkadaşları (Nachtigall vd 2016) yakın zamanda kullanılan derin CNN'nin insan uzmanlardan daha iyi performans gösterebileceğine dair bir hipotez önerdi. Bunu kanıtlamak için, yazarlar üç sınıflandırıcının doğruluğunu karşılaştırdılar: Çok seviyeli algılayıcı (MLP), AlexNet mimarisi ve 2539 görüntüden oluşan bir veri kümesi üzerinde eğitimleri yapıldı. MLP %77,3 doğrulukla çok kötü performans gösterirken, AlexNet %97,3 doğrulukla uzmanların %96 doğruluğunu geçebildi.

CNN'lerin yüksek karmaşıklığı nedeniyle, bir ağı eğitmek haftalar alabilir. Bu dezavantaja karşı koymak için A. Ramcharan ve arkadaşları (Ramcharan vd 2017) (Panwar vd. 2018) bir transfer öğrenme yaklaşımı kullanmıştır. Önceden eğitilmiş bir Inception v3'ün evrim katmanlarını kullandılar. Sınıflandırma üç farklı yolla yapıldı:

orijinal softmax başlangıç katmanı v3, SVM ve K-en yakın komşular (KNN), $K=3$. KNN kullanılarak %73, SVM sınıflandırıcı kullanılarak %91 doğruluk elde edilmiştir. Bir başka araştırma makalesi, K.P. Ferentinos (Ferentinos, K.P., 2018) daha önce tartışılan makaleye göre artan iyileştirmeler vaat ediyor. Bu amaçla Plant Village veri seti ve gerçek dünya görüntüleri dahil olmak üzere 87.848 görüntüden oluşan açık bir veri seti ile birlikte üst düzey donanım kurulumu kullanıldı.



4. ARAŞTIRMA BULGULARI

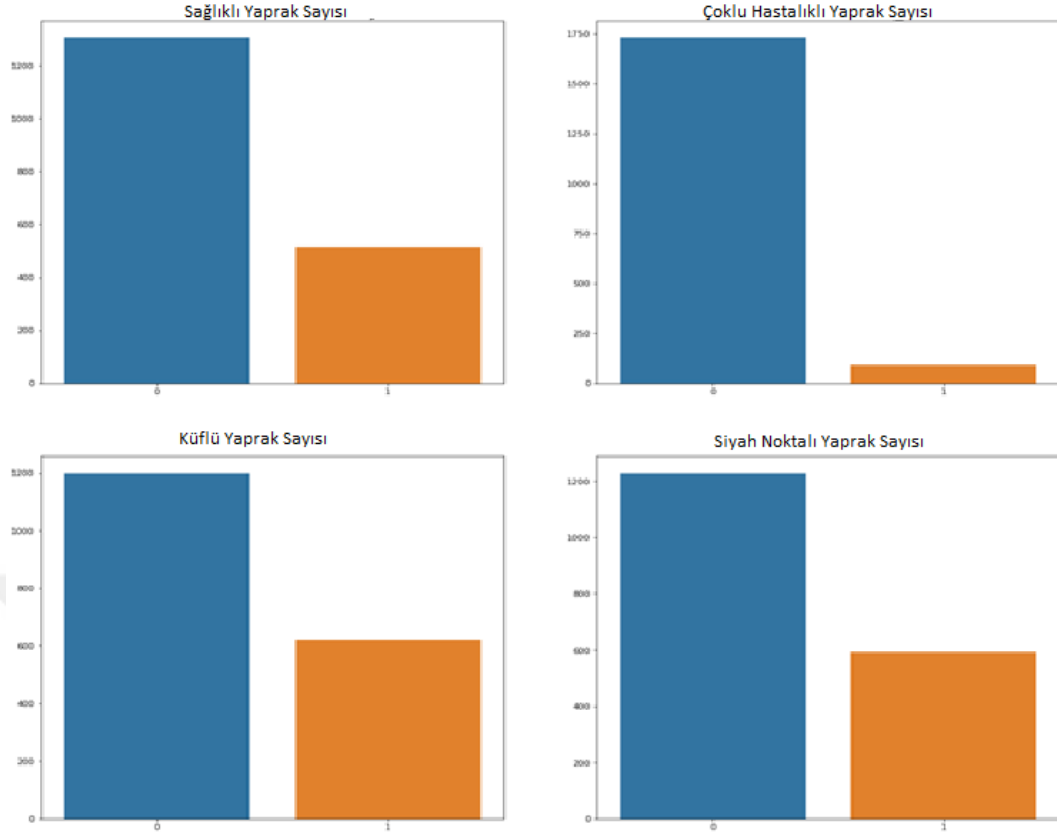
Belirli sayıdaki veri seti üzerinden elma yapraklarındaki hastalık tespiti için en efektif modeli üretme ve doğruluk oranını artırmak amacıyla işlemler yapılmıştır.

4.1 Kullanılan Çevre Değişkenleri

Bu çalışmada kullanılan tüm modeller GPU/CPU desteği ile derlenmiştir. Tüm deneysel çalışmalar, 9. Nesil Intel® Core™ i7 9750H ve 16 GB DDR4-SDRAM ile 6GB belleğe sahip NVIDIA® GeForce® GTX 1660 Ti ile çalışan Windows 10 işlemi sistemi üzerinden gerçekleştirilmiştir. Tüm kodlar Python dilinde yazılmış açık kaynaklı bir derin sinir ağı kütüphanesi olan Keras 2.3.1 çerçevesi ile gerçekleştirilmektedir. Python geliştirici ortamı olarak Spyder 5.1.5 sürümü kullanılmıştır.

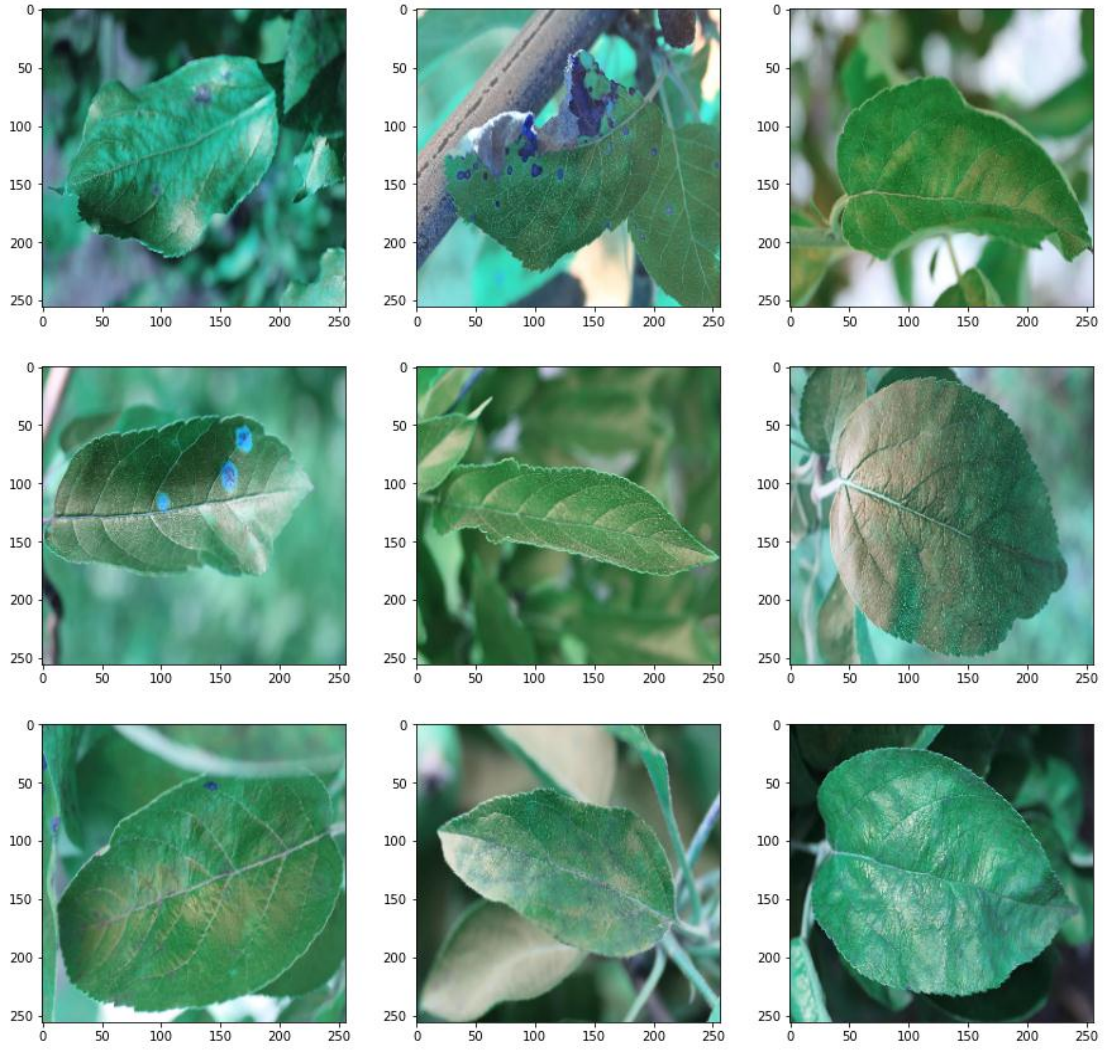
4.2 Veri Seti

Veri seti olarak internet üzerinden alınan 3642 elma yaprağı verisinden veri seti oluşturulmuştur. Eğitilen veri ve Validasyon serisi için 1821 veri, 1821 de Test verisi olarak kullanılmıştır. Veriler sağlıklı elma yaprağı (516), çoklu hastalıklı elma yaprağı (91), küflü hastalıklı elma yaprağı (622) ve siyah lekeli hastalıklı elma yaprağından (592) oluşmaktadır.



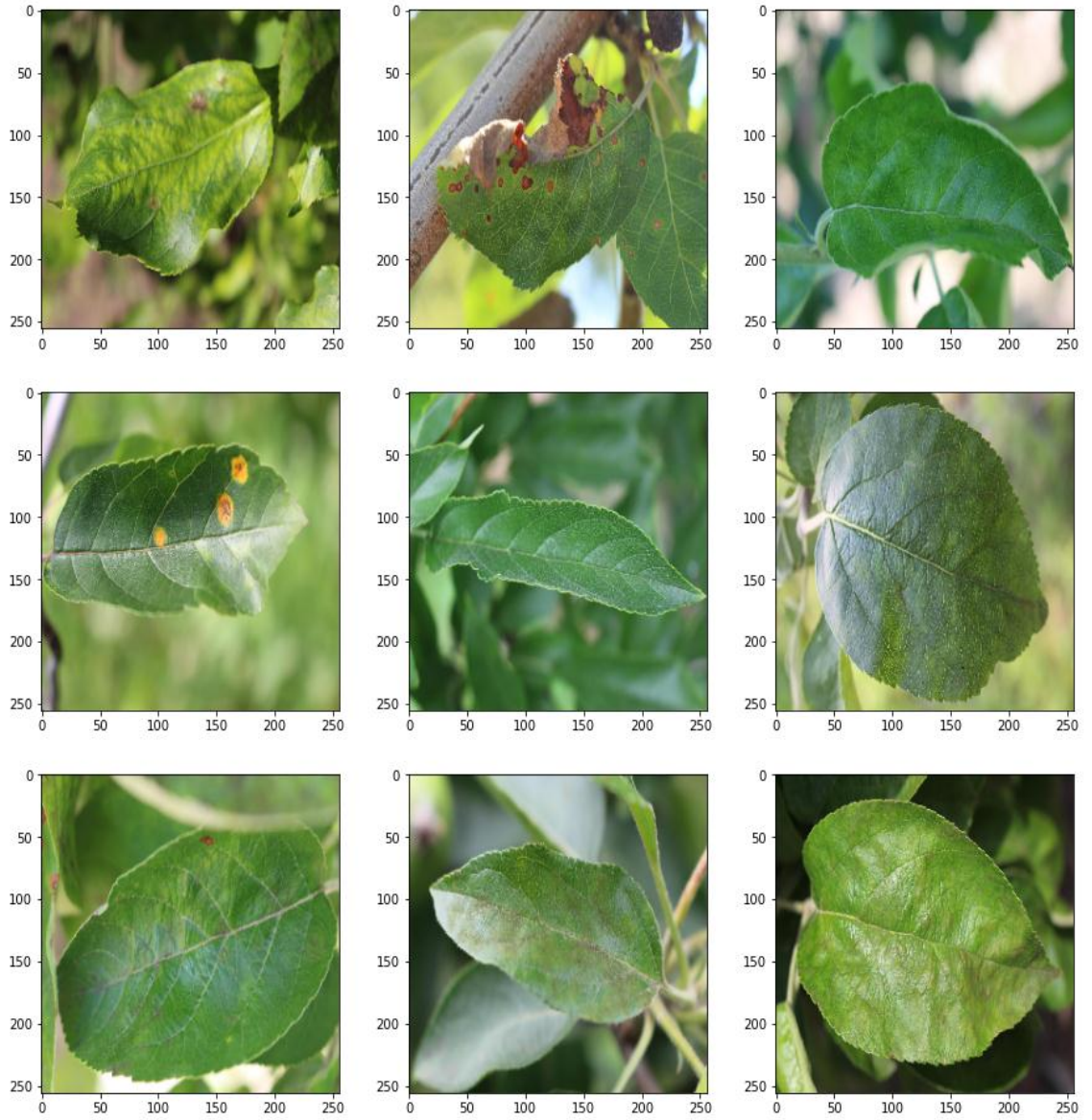
Şekil 4.1 Veri Dağılımı

Veri setinin BGR olarak belirli örnekleri aşağıdaki gibidir. Burada veri setinden küflü yaprak, sağlıklı yaprak, çok hastalıklı yaprak ve siyah lekeli yapraklar mevcuttur. Resim boyutlarını 256X256 olarak gösterilmesi sağlanmıştır.



Şekil 4.2 Veri Seti Örneği – BGR

Aynı veri seti yaprak örneğinin RGB gösterimi ise aşağıdaki gibidir. Burada renkteki farklılıklar göz ile daha net görülebilmektedir.



Şekil 4.3 Veri Seti Örneği – RGB

4.3 Kullanılan Çevre Değişkenleri

Bu çalışmada kullanılan tüm modeller GPU/CPU desteği ile derlenmiştir. Tüm deneysel çalışmalar, 9. Nesil Intel® Core™ i7 9750H ve 16 GB DDR4-SDRAM ile 6GB belleğe sahip NVIDIA® GeForce® GTX 1660 Ti ile çalışan Windows 10 işlemi sistemi üzerinden gerçekleştirilmiştir. Tüm kodlar Python yazılım dili ile hazırlanmıştır. Kodların yazımında açık kaynaklı kütüphane olan Keras 2.3.1 (bir derin sinir ağı kütüphanesi) çerçevesi ile gerçekleştirilmektedir.

4.3.1 Veri seti bölümlendirme

Net sonuçlar elde edebilmek için veri seti üzerinde bölümlenmeler yapılmıştır. Yapılan ve buna göre sonuçları çıkartılan veri seti bölümlenmesi Şekil 4.4'teki gibi yapılmıştır. Tüm bölümlendirme test verileri için ayrılan sayılar sabit tutulmuştur. Eğitilen veri ve Validasyon serisi için ayrılan veri sayıları üzerinde değişiklikler yapılmış ve bunun eğitim üzerindeki etkileri test edilmiştir.

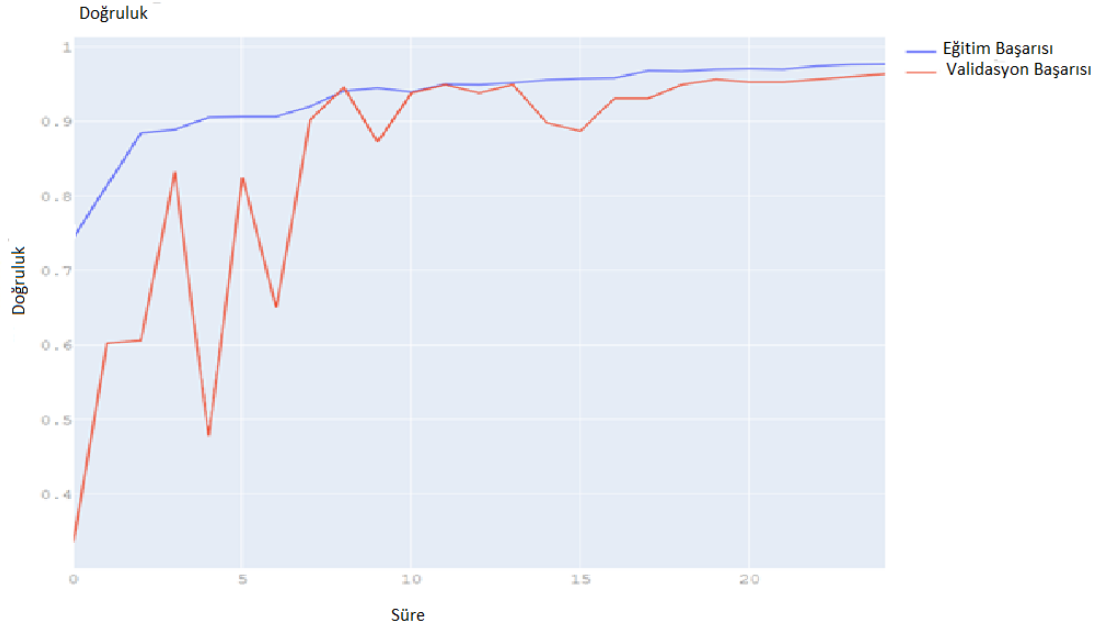
Data Bölümlendirme		Açıklaması
50%	25%-25%	Eğitim : 25% - Validasyon: 25%
	30%-20%	Eğitim : 30% - Validasyon: 20%
	35%-15%	Eğitim : 35% - Validasyon: 15%
50%		Test: 50%

Şekil 4.4 Veri Seti Bölümlendirme

4.3.2 Eğitim sonuçları

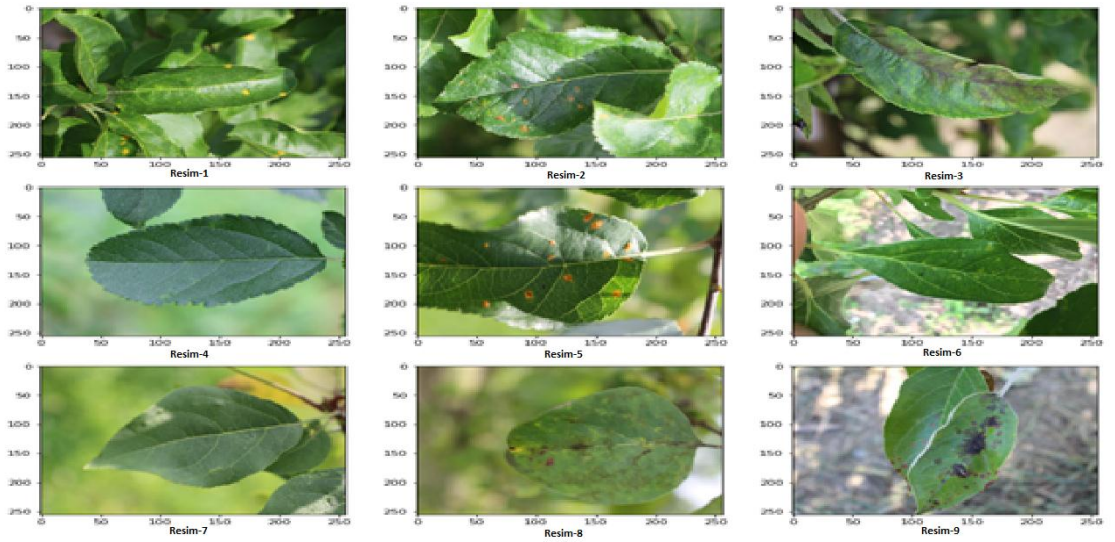
Bu çalışmada kullanılan veri setine en iyi sonuçları alabilmek için CNN mimarilerinden olan Resnet50V2 ve MobileNet ile çalışmalar yapılmıştır. Resnet50V2 ile daha iyi sonuçlar alındığı görülmüş ve daha iyi sonuçlar almak için optimize edici değiştirme, katman azaltıp ekleme işlemleri yapılmıştır.

Resnet50V2 ile alınan sonuçlar aşağıdaki şekilde gösterilmiştir. Resnet50V2 ile yaklaşık 0,9672 gibi sonuç elde edilmiştir.



Şekil 4.5 Resnet50V2 Başarı Grafiği

Resnet50V2 mimarisi ile yukarıdaki şekildeki sonucu almak için train için 1546 veri, test için 275 veri ayrılmıştır. Optimizasyon için Adam algoritması kullanılmıştır. Resnet50V2 dışında modele ek olarak GlobalAveragePooling2D ve 3 adet Dense katmanı eklenmiştir. Eğitim tamamlandıktan sonra daha önce eğitim aşamasında hiç sisteme gösterilmemiş test verileri ve bunların kısa sonucu şekillerde gösterilmiştir.

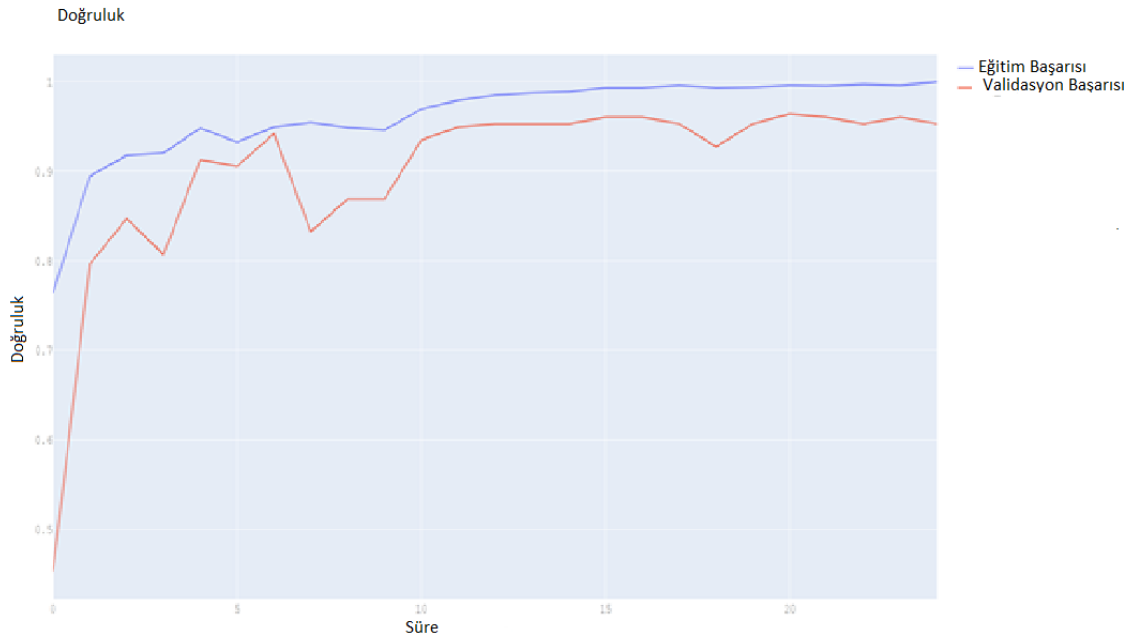


Şekil 4.6 Resnet50V2 ile Eğitim Yapılan Örnek 9 Veri

resim_id	Sağlıklı Yaprak(%)	Çok Hastalıklı (%)	Küflü Yaprak (%)	Siyah Benekli Yaprak (%)
Resim-1	0,02	1,02	98,93	0,03
Resim-2	0,01	0,23	99,75	0,01
Resim-3	0,03	1,09	0,08	98,8
Resim-4	99,93	0,01	0,04	0,02
Resim-5	0,01	0,02	99,96	0,01
Resim-6	97,8	1,95	0,05	0,2
Resim-7	99,94	0,01	0,01	0,04
Resim-8	0,05	0,04	0,01	99,9
Resim-9	0,02	2,46	0,01	97,51

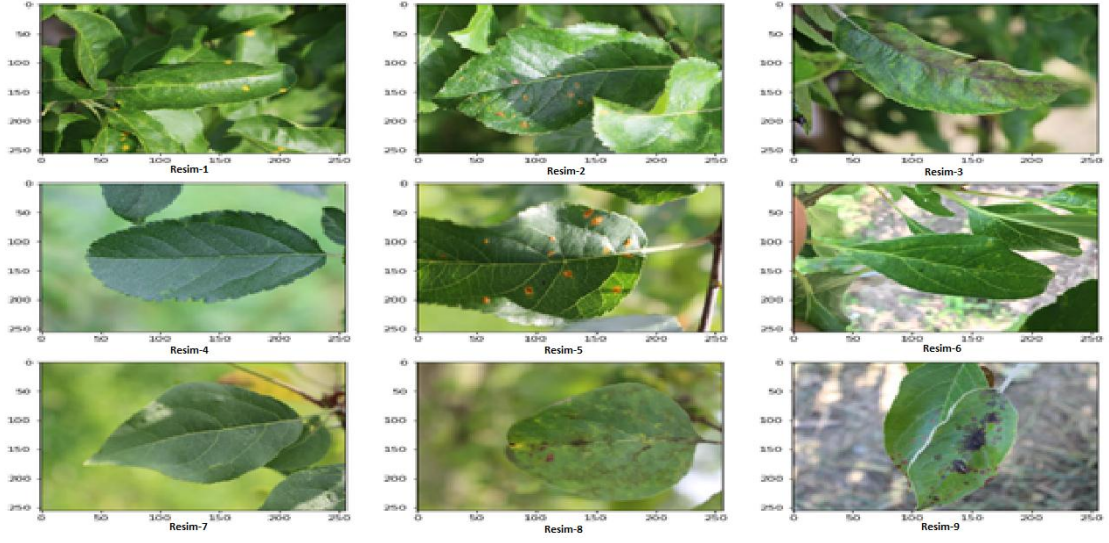
Şekil 4.7 Resnet50V2 ile Şekil 4.6'daki Veri Setinin Başarı Oranları

Resnet50V2 eğitiminden sonra MobileNet ile eğitim ve testi yapılmıştır. MobileNet ile yaklaşık 0.9635 gibi bir sonuç elde edilmiştir. Eğitim sırasında elde edilen başarı grafiği şekildeki gibidir.



Şekil 4.8 MobileNet Başarı Grafiği

MobileNet mimarisi ile yukarıdaki şekildeki sonucu almak için train için 1546 veri, test için 275 veri ayrılmıştır. Optimizasyon için Adam algoritması kullanılmıştır. MobileNet dışında modele ek olarak GlobalAveragePooling2D ve 3 adet Dense katmanı eklenmiştir. Eğitim tamamlandıktan sonra daha önce eğitim aşamasında hiç sisteme gösterilmemiş test verileri ve bunların kısa sonucu şekillerde gösterilmiştir.

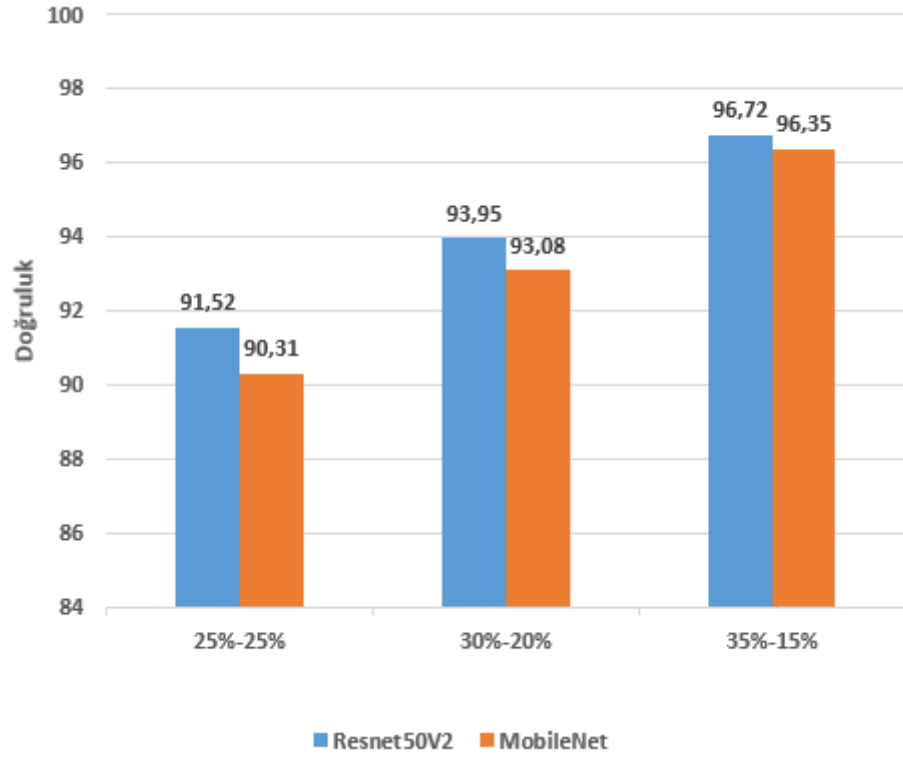


Şekil 4.9 MobileNet ile eğitim Yapılan Örnek 9 Veri

resim_id	Sağlıklı Yaprak(%)	Çok Hastalıklı (%)	Küflü Yaprak (%)	Siyah Benekli Yaprak (%)
Resim-1	0,01	0,6	99,38	0,01
Resim-2	0,01	36,65	63,32	0,02
Resim-3	0,02	0,09	0,01	99,88
Resim-4	99,94	0,02	0,02	0,02
Resim-5	0,01	0,04	99,91	0,04
Resim-6	99,53	0,02	0,4	0,05
Resim-7	98,48	0,76	0,64	0,12
Resim-8	0,07	0,09	0,05	99,79
Resim-9	0,05	98,83	0,11	1,01

Şekil 4.10 MobileNet ile Şekil 4.9'daki Veri Setinin Başarı Oranları

Şekil 4.4'e ResNet50V2 ile MobileNet mimarileri üzerinde testler yapılmıştır. En fazla başarılar Şekil 4.4'te yer alan üçüncü satırdaki eğitilen veri 35% validasyon serisi 15% olarak bölümlendirildiğinde elde edilmiştir. Bu testin sonuçları aşağıda verilmiştir.

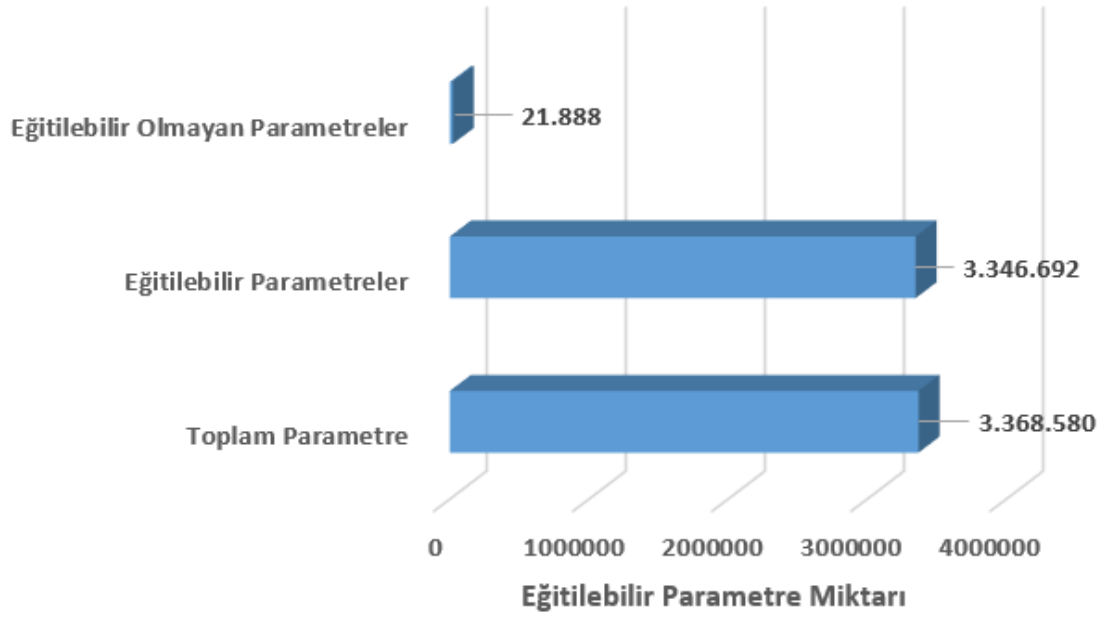


Şekil 4.11 Veri Seti Bölümlendirmeye Göre Sonuçlar

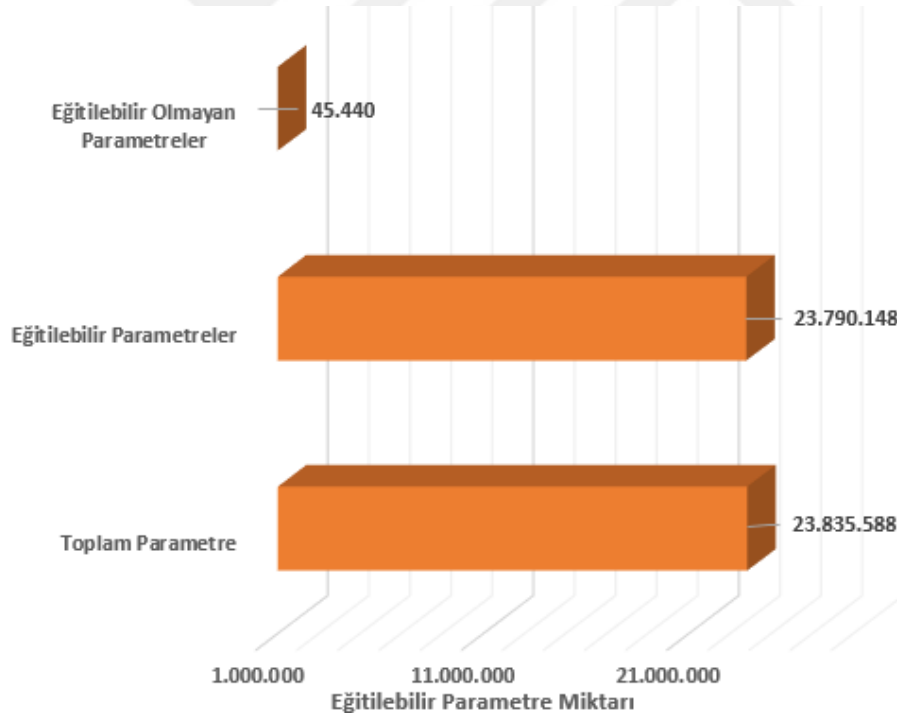
35%-15% oranına göre kayıp oranı ResNet50V2'nin MobileNet e göre daha az olduğu görülmüştür. Şekil 4.12'de başarı oranları ve kayıp oranları gösterilmiştir.

Model	Başarı Oranı	Kayıp Oranı
ResNet50V2	96.72	0.1735
MobileNet	96.35	0.1846

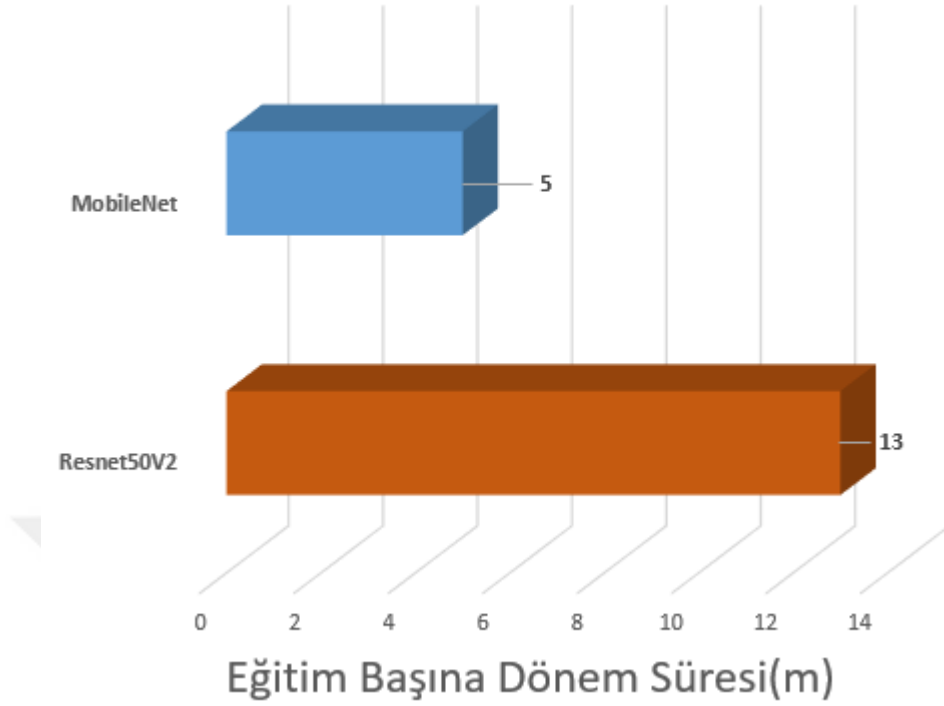
Şekil 4.12 ResNet50V2 ve MobileNet Mimarilerinin 35%-15% e Göre Başarı Tablosu



Şekil 4.13 MobileNet Eğitilebilir Parametre Sayısı



Şekil 4.14 ResNet50V2 Eğitilebilir Parametre Sayısı

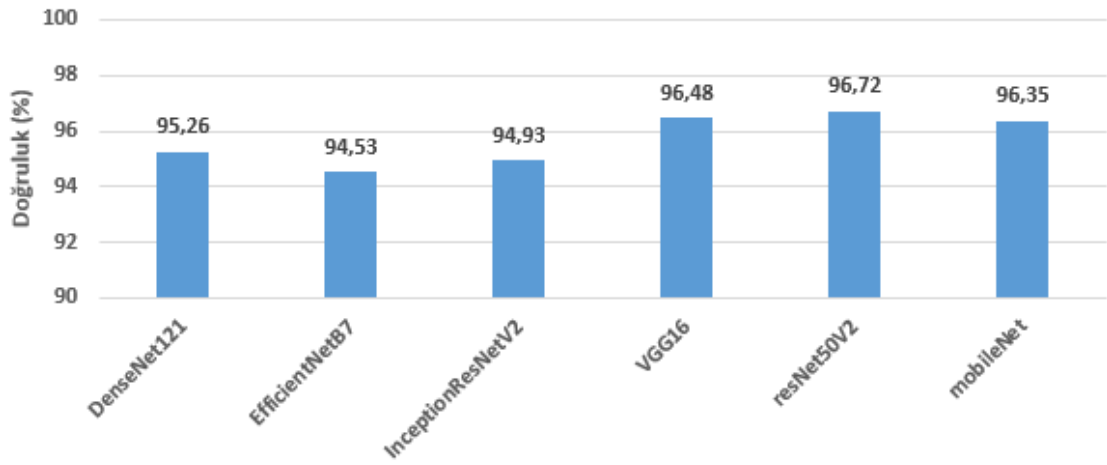


Şekil 4.15 resNet50v2 ve MobileNet Dönem Süreleri

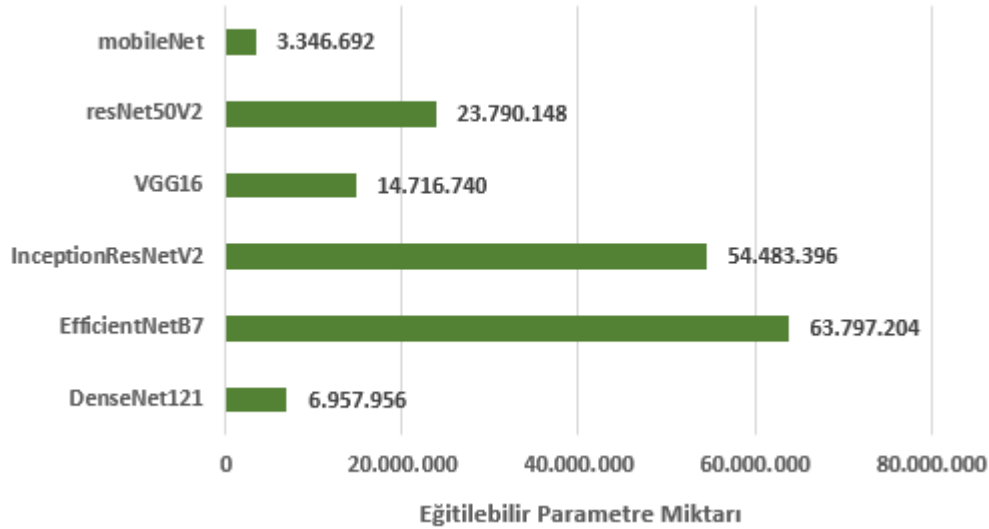
Bir sonraki analiz, Şekil 4.14 ve 4.15'teki gösterildiği gibi bir dönem için eğitilebilir parametre sayısı ve eğitim süresi açısından bu mimarilerin hesaplama maliyetini analiz edilmiştir. Şekil 4.14 ResNet50V2'deki eğitilebilir parametre sayısının MobileNet'e göre daha fazla parametreye sahip olduğu görülmüştür. Şekil 4.15'de ise parametre sayısına göre çalışma oranının daha kısa bir süre olduğu belirlenmiştir. ResNet50V2'nin hesaplama karmaşıklığının MobileNet'e göre daha iyi olduğu tespit edilmiştir.

MobileNet ile ResNet50V2 mimarileri dönem sayıları artırılarak da karşılaştırılmıştır. ResNet50V2'de dönem 25 yapılmış olsa bile maksimum değerlere 7.dönem'de ulaşıldığı görülmüştür. Şekil 3.4'de de grafikten görülebilir. MobileNet'de ise en yüksek başarı 6.dönem'de alınmaktadır. Şekil 4.5'de grafikten dönem ve başarıları görebilirsiniz. Dönem sayısı 7'nin altındaysa MobileNet başarıları ResNet50V2'den yüksek fakat 7 ve üzerine geçince ResNet50V2'nin daha başarılı olduğunu ortaya çıkmıştır.

MobileNet ile ResNet50V2 ile karşılaştırmalar yapıldıktan sonra diğer mimarilerde de en iyi durumlara göre aynı şartlarda eğitim sağlanmıştır. Test olarak tüm verinin 50% si, eğitilen veri için 35% ve validasyon serisi için 15% ayrılmıştır. Dönem olarak 25 belirlenmiştir. Optimizasyon algoritması olarak ADAM seçilmiştir. Ek olarak GlobalAveragePooling2D ve 3 adet Dense katmanları da eklenmiştir. Karşılaştırmalar Şekil 4.16 ve 14.7’de gösterilmiştir.



Şekil 4.16 CNN Mimarileri Başarı Grafiği



Şekil 4.17 CNN Mimarileri Parametre Sayıları

5. TARTIŞMA ve SONUÇ

Bu yüksek lisans tez çalışmasında bitki yapraklarındaki hastalıkları tespit etme, bunlarla ilgili kullanılan CNN mimarilerinin detaylı açıklaması, derin öğrenme algoritma ve mimarilerinden bahsedilmiştir.

Tez kapsamında, bitki yaprak hastalıklarında en çok kullanılan yöntemler ve CNN mimarileri incelenmiştir. Tezin asıl konusu olan elma yaprağı hastalık tespiti için CNN mimarileri arasında Resnet50V2 ile MobilNet mimari kullanılarak başarı yüzdeleri karşılaştırılmıştır. Eğitimlerde optimizasyon algoritması olarak Adam algoritması kullanılmıştır. Genel karşılaştırmalar ResNet50V2 ile MobileNet üzerinden yapılmıştır fakat diğer bazı hazır mimariler ile de başarı çalışması yapıp karşılaştırıldı.

Çalışma süresi bakımından ResNet50V2 ile MobileNet karşılaştırılıp en efektif olarak ResNet50V2 olduğu tespit edildi.

Eğitimde model oluşturulurken başarıyı artırmak için CNN mimarilerine ek Dense ve GlobalAveragePooling2D katmanları eklenmiştir. Ayrıca eğitilen veri ve validasyon serisine ayrılan veri sayılarına göre başarılar karşılaştırılmış ve başarı artırılması için uygun oran bulunmuştur. İki eğitimden de çıkan sonuçta başarı oranı yüksek olduğundan hastalığı tespit etmede manuel göz ile bu hastalıkları tespit etme ihtiyacını ortadan kaldırmıştır.

KAYNAKLAR

Akbaş, E., 2018. Derin Yapay Sinir Ağları ve Derin Öğrenme'ye Kısa Bir Giriş. Bilgisayar Mühendisleri Odası Semineri, Ankara.

Anonim 2021a. Web Sitesi: <https://www.veribilimiokulu.com/yapay-sinir-agiartificial-neural-network-nedir/#:~:text=Yapay%20sinir%20a%C4%9Flar%C4%B1%20insan%20beyni,%C3%B6%C4%9Frenme%20i%C5%9Flemi%20%C3%B6rnekleme%20kullan%C4%B1larak%20ger%C3%A7ekle%C5%9Ftirilir,> Erişim Tarihi: 02. 12.2021.

Anonim 2021b. Web Sitesi: <https://www.derinogrenme.com/2017/03/04/yapay-sinir-aglari>, Erişim Tarihi: 02. 12.2021.

Anonim 2021c. Web Sitesi: <https://medium.com/@hamzaerguder/recurrent-neural-network-nedir-bdd3d0839120#:~:text=RNN%2C%20bu%20long%2Dterm%20ba%C4%9F%C4%B1ml%C4%B1%C4%B1klar%C4%B1n,veri%20ile%20d%C3%B6ng%C3%BCye%20girerek%20i%C5%9Flenir,> Erişim Tarihi: 03. 12.2021.

Anonim 2021d. Web Sitesi: [https://dergipark.org.tr/en/download/article-file/878498#:~:text=LSTM%2C%20tekrarlayan%20sinir%20a%C4%9Flar%C4%B1%20mimarisinin,durumlar%C4%B1%20%C3%B6%C4%9Frenmesi%20ile%20dikkat%20%C3%A7ekmektedir.\)](https://dergipark.org.tr/en/download/article-file/878498#:~:text=LSTM%2C%20tekrarlayan%20sinir%20a%C4%9Flar%C4%B1%20mimarisinin,durumlar%C4%B1%20%C3%B6%C4%9Frenmesi%20ile%20dikkat%20%C3%A7ekmektedir.)) , Erişim Tarihi: 03.12.2021.

Anonim 2021e. Web Sitesi: <https://medium.com/datatype/restricted-boltzmann-machine-a-complete-analysis-part-3-contrastive-divergence-algorithm-3d06-bbeb10c>, Erişim Tarihi: 05. 12.2021.

Anonim 2021f. Web Sitesi: https://en.wikipedia.org/wiki/Deep_belief_network, Erişim Tarihi: 05. 12.2021.

Anonim 2022a. Web Sitesi: <http://www.akademiya-payzeka.com/DesktopDefault.aspx?tabindex=4&tabid=4>, Eriřim Tarihi: 02.03.2022.

Anonim 2022b. Web Sitesi: <https://medium.com/@batincangurbuz/uzun-k%C4%B1sa-d%C3%B6nemli-bellek-long-short-term-memory-lstm-c526980c28b1#:~:text=LS TM%20yap%C4%B1s%C4%B1%20bilginin%20h%C3%BCrenin%20i%C3%7 ine,yeni%20verilerin%20hat%C4%B1rlanmas%C4%B1n%C4%B1%20sa%C4% 9Flayarak%20denetler>, Eriřim Tarihi: 02.03.2022.

Anonim 2022c. Web Sitesi: <https://medium.com/@batincangurbuz/boltzman-makinesi-boltzmann-machines-bm-ve-k%C4%B1s%C4%B1tlanm%C4%B1%C5%9F-boltzmann-makineleri-restricted-boltzmann-6f2e0f839b05>, Eriřim Tarihi: 02.03.2022.

Anonim 2022d. Web Sitesi: <https://medium.datadriveninvestor.com/deep-learning-deep-belief-network-dbn-ab715b5b8afc>, Eriřim Tarihi: 05.03.2022.

Anonim 2022e. Web Sitesi: <https://medium.com/@batincangurbuz/derin-oto-kodlay%C4%B1c%C4%B1lar-denoising-autoencoders-dae-ve-oto-kodlay%C4%B1c%C4%B1lar-autoencoders-ae-bbd3b1389d60>, Eriřim Tarihi: 05.03.2022.

Anonim 2022f. Web Sitesi: <https://medium.com/@batincangurbuz/derin-oto-kodlay%C4%B1c%C4%B1lar-denoising-autoencoders-dae-ve-otokodlay%C4%B1c%C4%B1lar-autoencoders-aebbd3b1389d60#:~:text=Diabolo%20a%C4%9F%C4%B1%20olarak%20adland%C4%B1r%C4%B1lan%20Oto,veriyi%2C%20%C3%A7%C4%B1kt%C4%B1%20katman%C4%B1nda%20tekrar%20olu%C5%9Fturur>, Eriřim Tarihi: 05.03.2022.

Anonim 2022g. Web Sitesi: <https://medium.com/towards-artificial-intelligence/the-architecture-andimplementation-of-vgg-16-b050e5a5920b>, Eriřim Tarihi: 01.03.2022.

Anonim 2022h. Web Sitesi: <https://iq.opengenus.org/resnet50-architecture/#:~:text=ResNet50%20is%20a%20variant%20of,explored%20ResNet50%20architecture%20in%20depth>, Eriřim Tarihi: 02.03.2022.

Anonim 2022i. Web Sitesi: <https://iq.opengenus.org/inception-v3-model-architecture/>, Eriřim Tarihi: 02.03.2022.

Anonim 2022j. Web Sitesi: https://production-media.paperswithcode.com/methods/Screen_Shot_2020-06-06_at_10.45.54_PM.png, Eriřim Tarihi: 01.04.2022.

Anonim 2022k. Web Sitesi: <https://medium.com/analytics-vidhya/image-classification-with-mobilenet-cc6fbb2cd470>, Eriřim Tarihi: 21.03.2022.

Alpaydın, E. 2013. Yapay öğrenme. Boğaziçi Üniversitesi Yayınevi, 486, İstanbul.

Arslan, K. 2017. Eğitimde Yapay Zeka ve Uygulamaları. Batı Anadolu Eğitim Bilimleri Dergisi, 11(1), 71-88.

Atalay, M. ve Çelik, E. 2017. Büyük Veri Analizinde Yapay Zekâ ve Makine Öğrenmesi Uygulamaları. Mehmet Akif Ersoy Üniversitesi Sosyal Bilimler Enstitüsü Dergisi, 9(22), 156-172.

Ataseven, B. 2013. Yapay Sinir Ağları ile Öngörü Modellemesi. Öneri Dergisi, 10(39), 101-115.

Baranwal, S., Khandelwal, S., Arora, A. 2019. Deep learning convolutional neural network for apple leaves disease detection. In Proceedings of International Conference on Sustainable Computing in Science, Technology and Management (SUSCOM), Amity University Rajasthan, Jaipur-India.

Barratt S, Sharma R. 2018. A Note on the Inception Score,arXiv:1801.01973.

- Chao, X., Sun, G., Zhao, H., Li, M., He, D. 2020. Identification of apple tree leaf diseases based on deep learning models. *Symmetry*, 12(7), 1065.
- Ergezer, H., Dikmen M. Ve Özdemir E. 2003. Yapay Sinir Ağları Ve Tanıma Sistemleri. *Pivolka*, 2(6), 14-17.
- Ferentinos, K. P. 2018. Deep learning models for plant disease detection and diagnosis. *Computers and Electronics in Agriculture*, 145, 311-318.
- Goodfellow, I., Bengio, Y., Courville, A., Bengio, Y. 2016. *Deep learning*. MIT press, 767, Cambridge.
- Guo, Y., Liu, Y., Oerlemans, A., Lao, S., Wu, S., Lew, M. S. 2016. Deep learning for visual understanding: A review. *Neurocomputing*, 187, 27-48.
- Haykin, S. 1999. *Neural Networks: A Comprehensive Foundation*. Prentice Hall Inc., 842, New Jersey.
- Hinton, G. E., Osindero, S., ve Teh, Y. W. 2006. A fast learning algorithm for deep belief nets. *Neural computation*, 18(7), 1527-1554.
- Howard A. G., Zhu M., Chen B., Kalenichenko D., Wang W., Weyand T., Andreetto M., Adam H. 2017. MobileNets: Efficient convolutional neural networks for mobile vision applications.
- İnik, Ö., Ülker, E., 2017. Derin Öğrenme ve Görüntü Analizinde Kullanılan Derin Öğrenme Modelleri. *Gaziosmanpaşa Bilimsel Araştırma Dergisi*, 6(3), 85-104.
- Jia, Y., Shelhamer, E., Donahue, J., Karayev, S., Long, J., Girshick, R., Darrell, T. 2014, November. Caffe: Convolutional architecture for fast feature embedding. In *Proceedings of the 22nd ACM international conference on Multimedia*, 675-678.

- Jiang, P., Chen, Y., Liu, B., He, D., & Liang, C. 2019. Real-time detection of apple leaf diseases using deep learning approach based on improved convolutional neural networks. *IEEE Access*, 7, 59069-59080.
- Krizhevsky A., Sutskever I., Hinton G. E. 2012. Imagenet classification with deep convolutional neural networks. In *Advances In Neural Information Processing Systems*, 1097–1105.
- Kulkarni, A. H., Rai, H. M., Jahagirdar, K. A., Upparamani, 2013. A leaf recognition technique for plant classification using RBPNN and Zernike moments. *International Journal of Advanced Research in Computer and Communication Engineering*, 2(1), 984-988.
- Nabiyev, V. V. 2012. *Yapay Zekâ: İnsan-Bilgisayar Etkileşimi*. Seçkin Yayıncılık, 816, Ankara.
- Liu, B., Zhang, Y., He, D., & Li, Y. 2017. Identification of apple leaf diseases based on deep convolutional neural networks. *Symmetry*, 10(1), 11.
- Mohanty, S. P., Hughes, D. P., ve Salathé, M. 2016. Using deep learning for image-based plant disease detection. *Frontiers in plant science*, 7, 1419.
- Nachtigall, L. G., Araujo, R. M., Nachtigall, G. R., 2016. Classification of apple tree disorders using convolutional neural networks. In *Tools with Artificial Intelligence (ICTAI)*, 2016 IEEE 28th International Conference, 472-476.
- Panwar, D., Tomar, P., Singh, V. (2018). Hybridization of Cuckoo-ACO algorithm for test case prioritization. *Journal of Statistics and Management Systems*, 21(4), 539-546.

- Peng, X., Zhang, X., Li, Y. ve Liu, B. (2020). Research on image feature extraction and retrieval algorithms based on convolutional neural network. *Journal of Visual Communication and Image Representation*, 69, 102705.
- Ramcharan, A., Baranowski, K., McCloskey, P., Ahmed, B., Legg, J., Hughes, D. P. 2017. Deep learning for image-based cassava disease detection. *Frontiers in plant science*, 8, 1852.
- Schmidhuber, J. 2015. Deep learning in neural networks: An overview. *Neural networks*, 61, 85-117.
- Simonyan, K., & Zisserman, A. 2014. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*.
- Singh, V., & Misra, A. K. 2017. Detection of plant leaf diseases using image segmentation and soft computing techniques. *Information Processing in Agriculture*, 4(1), 41-49.
- Sladojevic, S., Arsenovic, M., Anderla, A., Culibrk, D., Stefanovic, D. 2016. Deep neural networks based recognition of plant diseases by leaf image classification. *Computational intelligence and neuroscience*.
- Şeker, A., Diri, B., Balık, H.H. 2017. Derin Öğrenme Yöntemleri ve Uygulamaları Hakkında Bir İnceleme. *Gazi Mühendislik Bilimleri Dergisi*, 3(3), 47-64.
- Tan, M., Le, Q.V., 2019. EfficientNet: Rethinking model scaling for convolutional neural networks. In: 36th Int. Conf. Mach. Learn. ICML 2019, 10691–10700, United States.
- Valev, K., Schumann, A., Sommer, L. ve Beyerer, J. (2018). A systematic evaluation of recent deep learning architectures for fine-grained vehicle classification. *International Society for Optics and Photonics*, 10649, 1064902.