



**DAĞ ÇİÇEKLERİNİN DERİNLEMESİNE
AYRIŞTIRILABİLİR EVRİŞİMLİ YAPAY SİNİR
AĞLARIYLA MODELLENMESİ**

Turan Emre KÖKSAL

Yüksek Lisans Tezi

Bilgisayar Mühendisliği Anabilim Dalı

Prof. Dr. Abdulsamet HAŞILOĞLU

2019

**ATATÜRK ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ**

YÜKSEK LİSANS TEZİ

**DAĞ ÇİÇEKLERİNİN DERİNLEMESİNE AYRIŞTIRILABİLİR
EVRIŞİMLİ YAPAY SINIR AĞLARIYLA MODELLENMESİ**

TURAN EMRE KÖKSAL

BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI

**ERZURUM
2019**

HER HAKKI SAKLIDIR



T.C.
ATATÜRK ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ



TEZ ONAY FORMU

“DAĞ ÇİÇEKLERİNİN DERİNLEMESİNE AYRIŞTIRILABİLİR EVRİŞİMLİ
YAPAY SİNİR AĞLARIYLA SINIFLANDIRILMASI “

Prof. Dr. Abdulsamet HAŞILOĞLU danışmanlığında, Turan Emre KÖKSAL tarafından hazırlanan bu çalışma 25/07/2019 tarihinde aşağıdaki jüri tarafından Bilgisayar Mühendisliği Anabilim Dalı'nda Yüksek Lisans Tezi olarak ~~oybirliği/oy çaldığı~~ (.../...) ile kabul edilmiştir.

Başkan : Prof. Dr. Abdulsamet HAŞILOĞLU

İmza :

Üye : Dr. Öğr. Üyesi Gülşah Tümüklü ÖZYER

İmza :

Üye : Dr. Öğr. Üyesi Mete ÇELİK

İmza :

Yukarıdaki sonuç;

Enstitü Yönetim Kurulunun 23/08/2019 tarih ve 33.../83... nolu kararı ile onaylanmıştır.

Prof. Dr. Mehmet KARAKAN Y.
Enstitü Müdürü

Not: Bu tezde kullanılan özgün ve başka kaynaklardan yapılan bildirişlerin, çizelge, şekil ve fotoğrafların kaynak olarak kullanımı, 5846 sayılı Fikir ve Sanat Eserleri Kanunundaki hükümlere tabidir.

ÖZET

Yüksek Lisans Tezi

DAĞ ÇİÇEKLERİNİN DERİNLEMESİNE AYRIŞTIRILABİLİR EVRİŞİMLİ YAPAY SİNİR AĞLARIYLA MODELLENMESİ

Turan Emre KÖKSAL

Atatürk Üniversitesi
Fen Bilimleri Enstitüsü
Bilgisayar Mühendisliği Anabilim Dalı

Danışman: Prof. Dr. Abdulsamet HAŞILOĞLU

Dünyada 374.000'in üzerinde damarlı bitki türünün olduğu bilinmektedir. Bu bitki türlerinin yaklaşık 295.000 (%78) adedi ise çiçekli bitkilerden oluşmaktadır. Çiçek türlerinin belirlenmesi ve sınıflandırması botanik araştırmacılar için bile oldukça zor ve zaman alıcı bir işlem olup bilgisayarlı görme alanı için de zor problemdir. Bir görsel nesne tanıma problemi olan çiçek tanıma problemi, üzerinde çeşitli araştırma ve yayınların olduğu bir alandır. Bu çalışmada Doğu Anadolu Bölgesi'nde yetişen 14 bitki türünün mobil ortamda sınıflandırılabilmesi için Derin Öğrenme algoritmaları kullanılmıştır. Derinlemesine Ayırıştırılabilir Yapay Sinir Ağları ile eğitilen model ile mobil uygulama geliştirilmiştir. Artan veri boyutunun sınıflandırma üzerindeki etkisi incelenmiştir.

2019, 107 sayfa

Anahtar Kelimeler: Bitki Sınıflandırma, Derin Öğrenme, Derinlemesine Ayırıştırılabilir Yapay Sinir Ağları, MobileNet

ABSTRACT

Master Thesis

MODELLING OF MOUNTAIN FLOWERS WITH DEPTHWISE SEPARABLE CONVOLUTIONAL NEURAL NETWORKS

Turan Emre KÖKSAL

Atatürk Üniversitesi
Fen Bilimleri Enstitüsü
Bilgisayar Mühendisliği Anabilim Dalı

Danışman: Prof. Dr. Abdulsamet HAŞILOĞLU

It is known that there are over 374.000 vein plant species in the world. About 295,000 (78%) of these plant species are flowering plants. Identification and classification of flower species is a very difficult and time-consuming process, even for botanical researchers, and is a difficult problem for computer vision field. The flower recognition problem, which is a visual object recognition problem, is an area of research and publications on it. In this study, Artificial Neural Networks were used to classify 14 plant species grown in the Eastern Anatolia Region with Deep Learning algorithms. The mobile application has been developed with the model that is trained with Depthwise Separable Convolutional Neural Networks. The effect of increasing data size on classification was examined.

2019, 107 pages

Keywords: Flower Classification, Deep Learning, Depthwise Separable Convolutional Neural Networks, MobileNet

TESSSEKKÜR

Yüksek lisans eğitimim boyunca bilgi, birikim ve tecrübeleriyle akademik kariyerime yön veren ve maddi manevi desteklerini esirgemeyen çok kıymetli danışman hocam Sayın Prof. Dr. Abdulsamet HAŞILOĞLU'NA içten teşekkürlerimi sunarım.

Gerek ders döneminde, gerekse tez döneminde çalışmalarına ışık tutan ve her talebimde kıymetli zamanlarını ayıran çok kıymetli hocalarım Dr. Öğr. Üyesi Sayın Barış ÖZYER'e, Dr. Öğr. Üyesi Sayın Gülşah TÜMÜKLÜ ÖZYER'e, Dr. Öğr. Üyesi Sayın Levent BAYINDIR'a teşekkürlerimi sunarım.

Tez çalışmam için gerekli kaynağı temin eden ve beni her koşulda destekleyen Erzurum Orman Bölge Müdürü Sayın Şaban BIYIKLI'ya, Erzurum Orman Bölge Müdürlüğü Bilgi Teknolojileri ve İletişim Şube Müdürü Sayın Hikmet MUSTAFAOĞLU'na teşekkürlerimi sunarım.

Yazılım geliştirme aşamasında desteklerini esirgemeyen değerli meslektaşım Oğuzhan AKKAYA'ya teşekkürlerimi sunarım.

Ayrıca, tüm eğitim hayatım boyunca her koşulda yanımda olan kıymetli aileme teşekkürlerimi sunarım.

Turan Emre KÖKSAL

Haziran, 2019

İÇİNDEKİLER

ÖZET.....	i
ABSTRACT.....	ii
TEŞEKKÜR.....	iii
SİMGELER ve KISALTMALAR DİZİNİ.....	vi
ŞEKİLLER DİZİNİ.....	vii
ÇİZELGELER DİZİNİ.....	ix
1. GİRİŞ.....	1
2. KAYNAK ÖZETLERİ.....	6
2.1. Yapay Sinir Ağları.....	6
2.2. Derin Öğrenme.....	10
2.3. Evrişimli Yapay Sinir Ağları.....	13
2.3.1. Standart Evrişimli yapay sinir ağları (CNN).....	14
2.3.2. Derinlemesine ayrıştırılabilir evrişimli sinir ağları.....	18
2.3.3. EYSA ve DAEYSA' nın maliyet bazında karşılaştırılması.....	19
2.4. ImageNet Veri Seti ve ImageNet Büyük Ölçekli Görsel Tanıma Yarışması.....	23
2.5. Derin Öğrenme İle Görüntü Sınıflandırma Çalışmaları.....	25
2.5.1. AlexNet.....	26
2.5.2. ZFNet.....	27
2.5.3. VGGNet.....	28
2.5.4. GoogleNet.....	30
2.5.5. ResNet.....	32
2.5.6. ResNetXT.....	34
2.5.7. DenseNet.....	35
2.5.8. SE-Net.....	37
2.6. Derin Öğrenme Kütüphaneleri.....	38
2.6.1. TensorFlow.....	38
2.6.2. Keras.....	39
2.6.3. PyTorch.....	39
2.6.4. Caffe.....	39
2.6.5. Theano.....	40
2.6.6. DeepLearning4J.....	40

2.6.7.	Derin öğrenme kütüphanelerinin kıyaslanması	40
2.7.	Transfer Öğrenmesi	43
2.8.	Bilgisayarlı Görme Alanında Bitki Tanımlama Çalışmaları.....	45
3.	MATERYAL ve YÖNTEM	49
3.1.	Materyal	49
3.1.1.	Veri seti	50
3.2.	Yöntem.....	52
3.2.1.	Sistem altyapısı ve özet mimarisi	52
3.2.2.	Verilerin ön işlenmesi	53
3.2.3.	Web uygulaması	55
3.2.4.	Web hizmeti	62
3.2.5.	Görüntü verilerinin eğitilmesi	64
3.2.6.	Mobil Uygulama.....	70
4.	ARAŞTIRMA BULGULARI ve TARTIŞMA	73
4.1.	Ağ Genişliğinin Öğrenme Üzerindeki Etkisi	74
4.2.	Görüntü Çözünürlüğünün Öğrenme Üzerindeki Etkisi.....	83
4.3.	Maliyet Analizi	90
5.	SONUÇ	93
KAYNAKLAR		94
EKLER		97
1-	EK-1 Baz Metod Kodları	97
2-	EK-2 Depo Kodları	99
3-	EK-3 Yeniden Eğitici Python Kodu.....	104
ÖZGEÇMİŞ		107

SİMGELER ve KISALTMALAR DİZİNİ

°	Derece
px	Piksel

Kısaltmalar

BA	Bayes Ağı
BD	Boyutsal Düşürme
BV	Büyük Veri
ÇKA	Çoklu Katman Algılayıcı
DAEYSA	Derinlemesine Ayrıştırılabilir Evrişimli Yapay Sinir Ağları
DD	Dalgacık Dönüşümü
DDB	Doğrultulmuş Doğrusal Birim
DÖ	Derin Öğrenme
DSA	Derin Sinir Ağları
EYSA	Evrişimli Yapay Sinir Ağları
GIÜ	Grafik İşlem Ünitesi
GSEOM	Gri Seviyeli Eş Oluşum Matrisleri
IBÖGTY	ImageNet Büyük Ölçekli Görüntü Tanımlama Yarışması
MIÜ	Merkezi İşlem Ünitesi
MÖ	Makine Öğrenmesi
SGAPA	Savunma Gelişmiş Araştırma Projeleri Ajansı
TYSA	Tekrarlayan Yapay Sinir Ağları
YSA	Yapay Sinir Ağları
YZ	Yapay Zekâ

ŞEKİLLER DİZİNİ

Şekil 2.1. Biyolojik Sinir Ağı ve YSA.....	7
Şekil 2.2. Yapay Sinir Hücresinin Yapısı	7
Şekil 2.3. DÖ ve Geleneksel Algoritmaların Veri Boyutu Performansı	11
Şekil 2.4. DÖ Evrimi.....	12
Şekil 2.5. Klasik Bir DÖ Ağında Sınıflandırma	13
Şekil 2.6. EYSA Yapısı.....	14
Şekil 2.7. Evrişim İşlemi.....	16
Şekil 2.8. Havuzlama İşlemi	17
Şekil 2.9. Derinlemesine Evrişim ve Nokta Bazlı Evrişim İşlemi.....	19
Şekil 2.10. Evrişim İşlemi.....	20
Şekil 2.11. Derinlemesine Evrişim İşlemi.....	21
Şekil 2.12. Nokta Bazlı Evrişim İşlemi.....	22
Şekil 2.13. ImageNet Sunflower Görüntüleri	24
Şekil 2.14. AlexNet Mimarisi	26
Şekil 2.15. ZFNet Mimarisi	27
Şekil 2.16. VGGNet Mimarisi.....	28
Şekil 2.17. Farklı Derinliklerdeki Test Sonuçları	29
Şekil 2.18. Inception Katmanı.....	31
Şekil 2.19. GoogleNet Mimarisi	31
Şekil 2.20. Artık Blok	32
Şekil 2.21. ResNet Mimarisinin İlk 34 Katmanı.....	33
Şekil 2.22. ResNet Mimarisi Bloğu ve ResNetXT Mimarisi Bloğu.....	34
Şekil 2.23. 3 Yoğun Bloklulu DenseNet Ağı	35
Şekil 2.24. Farklı DenseNet Mimarileri.....	36
Şekil 2.25. SE-Net Mimarisi	37
Şekil 2.26. Resnet Bloğu ve Önerilen SE Bloğu.....	37
Şekil 2.27. Derin Öğrenme Kütüphanelerinin Google Arama Trendleri.....	42
Şekil 2.28. Toplam Yayın Sayılarına Göre Karşılaştırma.....	42
Şekil 2.29. Kütüphanelerin Github Popülaritesi	43
Şekil 2.30. Geleneksel MÖ ve Transfer Öğrenmesi Arasındaki Farklar	44
Şekil 3.1. Ağırlı Dağı Florası Verileri.....	50
Şekil 3.2. Veri Seti	51
Şekil 3.3. Sistem Özet Mimarisi	52
Şekil 3.4. Akış Diyagramı.....	53
Şekil 3.5. Ön İşleme Süreci.....	54
Şekil 3.6. Veri Tabanı Diyagramı	56
Şekil 3.6. Veri Tabanı Diyagramı	56
Şekil 3.7. Web Uygulaması Girişinin Farklı Boyutlardaki Ekranlarda Görünümü	57
Şekil 3.8. DataTable.js Kütüphanesi İle Tablolama	58
Şekil 3.9. Ağaç Yapısı ile Bitki Türünün Hiyerarşik Yapısı.....	59
Şekil 3.10. Yeni Bitki Ekleme Kullanım Senaryosu.....	61
Şekil 3.11. Bitki Görüntüleri Galerisi	62
Şekil 3.12. Web Servisi ile Sınıflandırılmayan Görüntülerin Sunucuya Gönderilmesi	63
Şekil 3.13. Isparta Gülü Bitkisi Üzerinde Standart Evrişim İşlemi	67
Şekil 3.14. Isparta Gülü Bitkisi Üzerinde Derinlemesine Evrişim İşlemi	67

Şekil 3.15. Isparta Gülü Bitkisi Üzerinde Nokta Bazlı Evrişim İşlemi	67
Şekil 3.16. Ayçiçeği 001 Görüntüsü ve Oluşan Öznitelik Vektörü	69
Şekil 3.17. Tensorboard ile Eğitim Sürecinin İzlenmesi	70
Şekil 3.18. Mobil Uygulama	72
Şekil 4.1. Ağ Genişliğinin Sınıflandırma Üzerindeki Etkisi.....	83
Şekil 4.2. Görüntü Çözünürlüğünün Sınıflandırma Üzerindeki Etkisi	90
Şekil 4.3 Eğitim ve Doğrulama Aşamasındaki Keskinlik Değer Grafiği	92
Şekil 4.4. Eğitim Süresince Kayıp Fonksiyonu Değer Grafiği	92



ÇİZELGELER DİZİNİ

Çizelge 2.1. Geleneksel Algoritmalar ve YSA Kıyaslaması.....	8
Çizelge 2.2. TYSA ve EYSA Kıyaslaması	9
Çizelge 2.3. DÖ Kütüphanelerinin Kıyaslanması.....	41
Çizelge 2.4. 4 Farklı Sınıflandırma Yönteminin Başarı Yönünden Karşılaştırılması.....	45
Çizelge 2.5. Renk Gruplarına Ayırmanın Sınıflandırma Üzerindeki Etkisi	48
Çizelge 3.1. Kullanılan Materyaller.....	49
Çizelge 3.2. 30 Katmanlı MobileNet Mimarisi.....	66
Çizelge 4.1. Veri Setlerinin Türlerine Dağılımı	74
Çizelge 4.2. 224px Çözünürlük ve 1,0 Ağ Genişliği Sonuçları (Orijinal Veri Seti).....	75
Çizelge 4.3. 224px Çözünürlük ve 1,0 Ağ Genişliği Sonuçları (Döndürülmüş Veri Seti)	76
Çizelge 4.4. 224px Çözünürlük ve 0,75 Ağ Genişliği Sonuçları (Orijinal Veri Seti).....	77
Çizelge 4.5. 224px Çözünürlük ve 0,75 Ağ Genişliği Sonuçları (Döndürülmüş Veri Seti)	78
Çizelge 4.6. 224px Çözünürlük ve 0,50 Ağ Genişliği Sonuçları (Orijinal Veri Seti).....	79
Çizelge 4.7. 224px Çözünürlük ve 0,50 Ağ Genişliği Sonuçları (Döndürülmüş Veri Seti)	80
Çizelge 4.8. 224px Çözünürlük ve 0,25 Ağ Genişliği Sonuçları (Orijinal Veri Seti).....	81
Çizelge 4.9. 224px Çözünürlük ve 0,25 Ağ Genişliği Sonuçları (Döndürülmüş Veri Seti)	82
Çizelge 4.10. 1,0 Ağ Genişliği ve 192px Çözünürlük Sonuçları (Orijinal Veri Seti).....	84
Çizelge 4.11. 1,0 Ağ Genişliği ve 192px Çözünürlük Sonuçları (Döndürülmüş Veri Seti)	85
Çizelge 4.12. 1,0 Ağ Genişliği ve 160px Çözünürlük Sonuçları (Orijinal Veri Seti).....	86
Çizelge 4.13. 1,0 Ağ Genişliği ve 160px Çözünürlük Sonuçları (Döndürülmüş Veri Seti)	87
Çizelge 4.14. 1,0 Ağ Genişliği ve 128px Çözünürlük Sonuçları (Orijinal Veri Seti).....	88
Çizelge 4.15. 1,0 Ağ Genişliği ve 128px Çözünürlük Sonuçları (Döndürülmüş Veri Seti)	89
Çizelge 4.16. Maliyet Analizi	91

1. GİRİŞ

Dünya’da 374.000’in üzerinde damarlı bitki türünün olduğu bilinmektedir. Bu bitki türlerinin yaklaşık 295.000 (%78) adedi ise çiçekli bitkilerden oluşmaktadır (Christenhusz and Byng 2016). Aynı türde çiçekler farklı bölgelerde yetişebildiği gibi bölgeye göre yetişen özel türde çiçek türleri de bulunmaktadır.

Çiçek türlerinin belirlenmesi ve sınıflandırması botanik araştırmacılar için bile oldukça zor ve zaman alıcı bir işlem olup bilgisayarlı görme alanı için de zor problemdir. Bu durumun temel nedeni farklı çiçek türlerinin benzer doku ve renk dağılımında olması ve görüntülerde arka planın karmaşık yapıda olmasıdır. Bölgesel olarak çiçek türlerinin belirlenmesi ve sınıflandırması o bölgeye ait bitki coğrafyası yapısının haritalanmasına ve muhtemel yetiştirebilecek yeni türlerin belirlenmesinde faydalı olacaktır. Bir görsel nesne tanıma problemi olan çiçek tanıma problemi, üzerinde çeşitli araştırma ve yayınların olduğu bir alandır. Genel nesne tanıma problemlerine göre (İnsan, hayvan, araba, plaka vb.), nesne şeklinden ziyade bitkinin doku ve renk bilgileri kullanılarak bitki tanıma problemi çözülmektedir (Russell and Norvig 2009).

Günümüz teknolojisinde bilgisayar yaşamımızın her alanında yer almaya başlamıştır. Gelişen teknoloji, gerçek dünyada var olan problemlerin çözümünü daha kolay hale getirmiştir. Teknolojinin gelişimiyle birlikte eğitim, sağlık, botanik, askeri vb. alanlarda daha az maliyet ve iş gücüyle daha iyi performans ortaya koyan ürünlerin geliştirilmesi mümkün olmuştur. Gelişen teknolojiyle birlikte, bilgisayarlar da insanlar gibi düşünüp karar verme yeteneğine sahip olmuştur. Günümüzde devasa boyutlara ulaşan veri ve gelişen teknolojiyle birlikte çok daha fazla verinin hızlı bir şekilde işlenmesine ve bilgiye dönüştürülmesine imkân sağlamaktadır. Özellikle son yetmiş yılda yaşanan gelişmelerle Yapay Zekâ – YZ (Artificial Intelligence - AI), Makine Öğrenmesi – MÖ (Machine Learning - ML) ve akabinde Derin Öğrenme – DÖ (Deep Learning - DL)

alanında çokça yol kat edilmiştir. Bilgisayarlı görme alanında nesnelerin tanınması problemlerinde yüksek başarı ortaya konulmuş, milyonlarca görüntü verisini kullanarak insan, hayvan, bitki gibi başka nesnelerin tanınması ve sınıflandırılmasında da kolaylık sağlamıştır.

YZ, farklı disiplinlerdeki problemlere sunduğu akılcı çözümler ile gün geçtikçe önemi artan bir bilim dalı olmuştur. YZ'nin bu denli önemli olmasının başlıca sebebi, matematiksel olarak ifade edilebilen her türlü probleme çözüm sunabilmesidir. 1950'lerden bu yana, gelişmekte olan YZ genel ve dar YZ olarak iki kategoriye ayrılabilir. Genel YZ, insan zekâsının tüm karakteristik özelliklerini kapsarken, dar YZ insan zekâsının sadece belli özelliklerini kapsamaktadır.

YZ alanı, akıllı varlıkları anlamaya çalışmaktadır. Bu yüzden, çalışmanın bir nedeni kendimiz hakkında daha fazla şey öğrenmektir. Fakat felsefe ve psikolojiden farklı olarak insan zekâsıyla da ilgilenen YZ, akıllı varlıklar üretmenin yanı sıra onları anlamaya çalışır. YZ'nin önemli olmasının bir başka nedeni de, bu inşa edilmiş akıllı varlıkların insanlığa katkı sunacak olmasıdır. YZ, 1950'lerde bile bu konuda birçok önemli ve etkileyici ürün üretmiştir. Hiç kimse geleceği ayrıntılı olarak tahmin edemese de insan zekâsına (veya daha iyisine) sahip olan bilgisayarların günlük yaşamımız ve gelecekteki uygarlık süreci üzerinde büyük etkisi olacaktır (Russell and Norvig 2009).

Teknolojinin gelişimiyle birlikte YZ'nin gelişimine katkı sunan birçok yeni teknoloji ortaya çıkmıştır. Ses algılama, doğal dil işleme gibi teknolojiler YZ'nin öğrenme yeteneklerini geliştirmiştir. Bu gelişmelerle birlikte literatüre MÖ kavramı kazandırılmıştır. MÖ, mevcut verilerden elde ettiği bilgi ve tecrübeleri insan zekâsının yapabileceği gibi sonraki işlemler için bilgi olarak kullanabilme yeteneğinin makinelere kazandırılmasıdır.

DÖ, karar ağacı öğrenme, endüktif mantık programlama, kümeleme, güçlendirme öğrenme ve Bayes Ağı – BA (Bayesian Network - BN) MÖ yöntemlerinden birisidir. DÖ, insan beyninin biyolojik yapısından ve işlevlerinden esinlenilmiş ve birçok nöronun birbirine bağlanmasından oluşmaktadır. DÖ yöntemleri kullanılarak milyonlarca veri kullanılarak insan algısının fark edemediği yapıdaki özellikler çözülüp nesnelere teşhis edilebilmektedir (Goodfellow, Bengio et al. 2016).

DÖ, bilgisayarın daha basit kavramlardan karmaşık kavramlar oluşturmasını imkân verir. DÖ ile köşeler ve konturlar gibi daha basit özniteliklerden, nesne parçalarının sırayla tanımlanması ve bu kavramların birleştirilerek nesnenin tamamen tanımlanması amaçlanmaktadır (Goodfellow, Bengio et al. 2016).

Yapay Sinir Ağları – YSA (Artificial Neural Network - ANN) beynin biyolojik yapısını taklit eden algoritmalarıdır. YSA, birden çok gizli katman ve birbirleriyle tamamen bağlı nöronlardan oluşmaktadır. Bu gizli katmanlar verilerin ağıdaki depolandığı kısımlardır. Her bir katman kendinden önceki katmandan elde ettiği bilgilerle, kendinden sonraki katman için veri sağlamaya görevlidir.

Bu tez ile DÖ teknikleri kullanılarak Doğu Anadolu Bölgesi'nde yetişen 14 bitki türünün mobil ortamda tanımlanması amaçlanmıştır. 8 farklı YSA incelenmiştir. 14 bitki türünün mobil ortamda tanınabilmesi için DÖ yöntemlerinden olan Derinlemesine Ayırılabilir Evrişimsel Yapay Sinir Ağı - DAEYSA (Depthwise Separable Convolutional Neural Network- DSCNN) eğitilmiştir.

Mevcut çalışmalarda kullanılan veri setleri genelde geniş bir coğrafyadan toplanan çok sayıda çiçek sınıfına sahiptir. Mevcut fazla sınıf ve fazla fotoğraf içeren veri setlerinin kullanımı teknolojik olarak çok büyük faydalar sağlasa da, yerel sorunlara çözüm

arayan araştırma ekiplerinin amacına hizmet etmemektedir. Dolayısıyla, çalışmanın ilk aşamasında Doğu Anadolu Bölgesi'nde bulunan özellikle bal ormanları çevresinde konumlanan bitki türlerinin taranması, fotoğraflanması ve veri setinin oluşturulması hedeflenmiştir. Bu veri seti 550 türe ait 2031 görüntüden oluşmaktadır. Öğrenme işleminin kısıtlamalarından ötürü, ön işleme aşamasından 20'den az fotoğrafa sahip türler projenin sonraki aşaması için ayrılmıştır. Ön işleme aşamasında veriler bitki türlerine göre etiketlenmiştir. Böylece eğitim ve test sürecinde kullanılan veri seti 14 türe ait 1215 görüntüden oluşmaktadır. Ayrıca, ön işleme aşamasında veri boyutunun öğrenme üzerine etkisini de araştırabilmek adına mevcut görüntüler 30 °, 60 °, 90 °, 120 °, 150 °, 180 °, 210 °, 240 °, 270 °, 300 ° ve 330 ° döndürülerek veri seti genişletilmiştir. Veri setinin genişletilmesine yönelik çalışmalar devam etmektedir. Oluşturulan veri setinde tür başına düşen fotoğraf sayısı 63 ile 96 arasında değişmektedir.

Yapılan bu çalışmalar sonucunda literatüre aşağıda verilen katkıların sağlanması hedeflenmiştir:

- Türkiye Doğu Anadolu Bölgesi'nde yetişen 14 cinse ait 1215 fotoğraftan oluşan veri seti oluşturulmuştur.
- Veri setinin genişletilebilmesi ve kategorize edilmiş bir şekilde Web'de barındırılabilmesi için ekran boyutundan bağımsız bir Web uygulaması geliştirilmiştir.
- DÖ tekniklerinden DAEYSA kullanılarak farklı hiper parametrelerle ağlar eğitilmiş ve farklı boyuttaki verilerin öğrenme doğruluğu üzerindeki etkisi incelenmiştir.
- Herhangi bir çevrimiçi bağlantıya ihtiyaç duymadan bitki sınıflandırması yapabilen bir mobil uygulama geliştirilmiştir.
- Oluşturulan veri setinin diğer çalışmalarda da kullanılabilmesi için bir web hizmeti oluşturulmuştur.

Genel başlıklarıyla bu tezin; ikinci bölümünde, literatür özeti verilecektir. Tezin üçüncü bölümünde, materyal ve yöntem ile birlikte Doğu Anadolu Bölgesi'nde yetişen bitki türlerinin mobil ortamda tanımlanabilmesi için geliştirmiş olduğumuz yazılım anlatılacaktır. Takip eden bölümlerde de yapılan çalışma ile ilgili deneysel sonuçlar, yorum ve değerlendirmelerden bahsedilecektir.



2. KAYNAK ÖZETLERİ

2.1. Yapay Sinir Ağları

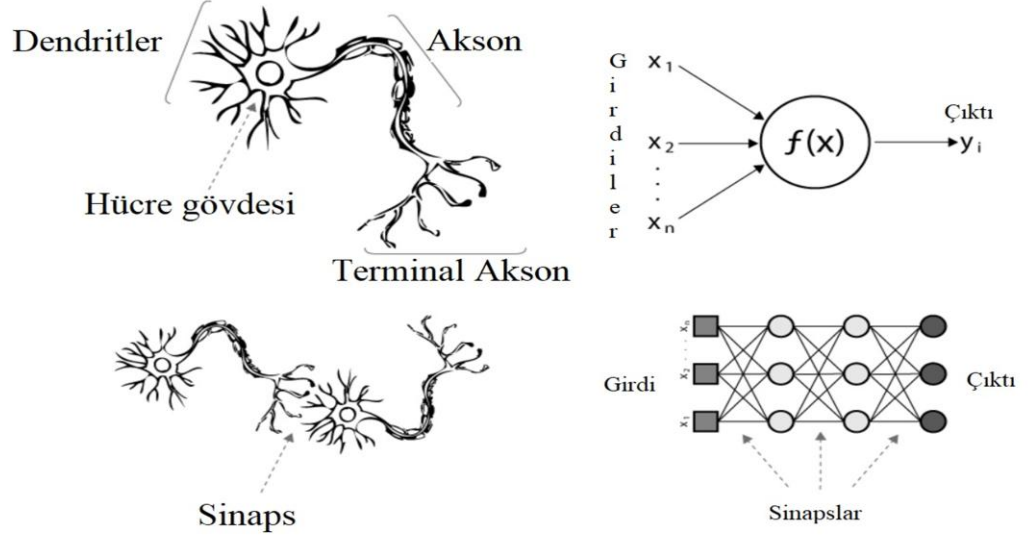
YSA, MÖ' de kullanılan ana araçlardan biridir. YSA, insan beyninin çalışma mantığını esas alır. 1950'lerden itibaren bir YZ yöntemi olarak sürekli gelişen YSA, araştırmacıların ilgi odağı olmuştur. 1943'de W. Pitts ve W. McCulloch tarafından ilk YSA modeli ortaya atılmıştır. McCulloch ve Pitts, insan beyninin aritmetik yetisini temel alarak, basit bir sinir ağı modeli geliştirmişlerdir (Oztemel 2010).

YSA'da asıl amaç kendisine verilen verilerden bilgi edinerek, çıkarımlarda bulunmak, genellemeler yapmak ve akabinde daha önce görmediği verileri edindiği tecrübelerden faydalanarak o veriler hakkında sonuç üretebilmesidir. YSA öğrenbilme özelliği sebebiyle günümüzde birçok alanda kullanılmakta ve mantıksal düzlemde ifade edilebilen problemleri başarı ile çözebilmektedir (Halit Ergezer 2003).

Bir diğer tanımla YSA; insan beynini temel alarak geliştirilmiş, ağırlıklı bağlantılar vasıtasıyla birbirine Tamamen Bağlı ve her biri kendi hafızasına sahip elemanlarından oluşan ve bilgiyi işleyip sonuç üreten yapılardır. Bir başka deyişle, biyolojik sinir ağları gibi çalışan bilgisayar yazılımlardır (Elmas 2003).

YSA biyolojik sinir sisteminden esinlenilerek geliştirilmiştir. Şekil 2.1'de görüldüğü üzere sinir hücreleri birbirleri ile sinapslar aracılığıyla iletişime geçerler. Sinir hücresi bilgiyi alıp işledikten sonra aksonlar vasıtasıyla ürettiği bilgiyi diğer sinir hücresine iletir. Yapay sinir hücreleri ise girdi olarak verilen bilgileri bir toplama fonksiyonu ile

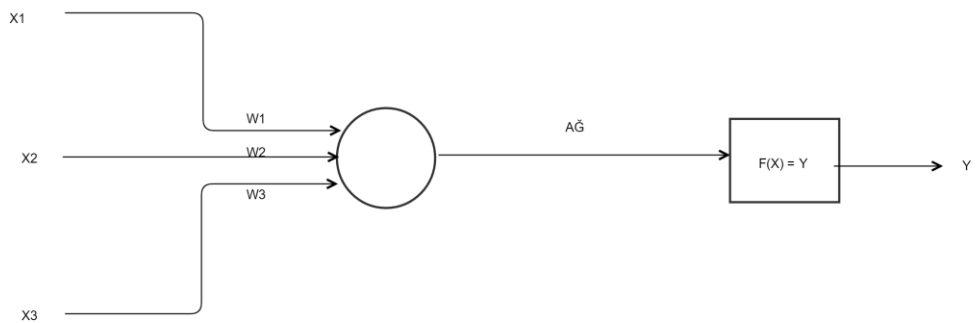
toplar ve aktivasyon fonksiyonu vasıtasıyla çıktı üretir. Üretilen bu çıktı ağdaki bağlantılar kanalıyla diğer hücelere gönderilir.



Şekil 2.1. Biyolojik Sinir Ağı ve YSA

Şekil 2.2’de gösterildiği gibi yapay sinir hücresi temel olarak şu bölümlerden oluşur:

- Girdi verisi
- Çıktı verisi
- Ağırlıklar
- Toplama
- Birleştirme Fonksiyonu



Şekil 2.2. Yapay Sinir Hücresinin Yapısı

YSA klasik algoritmalara göre bazı avantaj ve dezavantajlara sahiptir. Çizelge 2.1’de her iki tür için de kıyaslama sunulmuştur.

Çizelge 2.1. Geleneksel Algoritmalar ve YSA Kıyaslaması

Geleneksel Algoritmalar	YSA
Çıktılar, girdilere uygulanan katı kurallara göre şekillenir.	Girdilerden öğrenilenler neticesinde çıktılar şekillendirilir.
Bilgiler kesindir ve algoritmalar katı bir şekilde akışa uyar.	Tecrübelerden faydalanılır.
Yapılan hesaplamalar merkezidir. Eş zamanlı devam eder ve ardışıktır.	Yapılan hesaplamalar topludur. Eş zamanlı devam etmez ve öğrenme sonrasında paraleldir.
Hataya karşı tolerans bulunmaz.	Hatalara karşı toleranslıdır.
Görece daha hızlıdır.	Donanıma bağlı olarak nispeten daha yavaştır.

YSA farklı ilkelere dayanarak birçok türe ayrılmaktadır. YSA yapılarına, öğrenme algoritmalarına ve öğrenme zamanlarına göre farklılaşabilir. Her bir türün kendine özgü güçlü yanları bulunmaktadır. Ayrıca Çizelge 2.2’de Tekrarlayan Yapay Sinir Ağları - TYSA (RNN) ve Evrişimli Yapay Sinir Ağları – EYSA (Convolutional Neural Networks - CNN) kıyaslanmıştır.

Çizelge 2.2. TYSA ve EYSA Kıyaslaması

TYSA	EYSA
Sıralı veriler için uygundur.	Görseller gibi boyutsal veriler için uygundur.
TYSA, EYSA ile karşılaştırıldığında daha az özellik uyumluluğu içerir.	EYSA, TYSA'na göre daha güçlüdür.
İsteğe bağlı boyutlarda girdi alır ve çıktı üretir. Daha esnekler.	Belirli boyutlardaki girdileri alır ve belirli boyutlarda çıktı üretir.
TYSA ileri beslemeli sinir ağlarının aksine girdileri rastgele işlemek için dâhili hafızalarını kullanabilir.	EYSA, en az miktarda ön işleme kullanmak için tasarlanmış çok katmanlı algılayıcıların çeşitliliğine sahip ileri beslemeli bir YSA türüdür.
Ses ve yazı analizi için idealdir	Resim ve video gibi görsel verileri işlemek için idealdir.
TYSA zaman serisi bilgilerine göre oluşturulmuştur. Örneğin bir kullanıcının konuştuğu son cümle bir sonraki konuşmasındaki ilk cümleyi etkilemektedir.	Memelilerin görsel korteksinden esinlenilerek oluşturulmuştur.

YSA bilgisayarlı görme, makine çevirisi, konuşma tanıma, sosyal ağ filtreleme, spam mesaj filtreleme, video oyunları, nesne tanımlama ve tıbbi tanı gibi çeşitli alanlarda kullanılmaktadır. YSA sinyal işleme, kontrol sistemleri, örüntü tanıma, tıp, ses üretimi, ses tanıma ve ekonomi gibi alanlarda kullanılmaktadır (Laurene 1994).

YSA sınıflandırma yapmada, tahmin yürütmede, verileri ilişkilendirme, verileri yorumlamada ve verileri filtrelemede kullanılmaktadır (Oztemel 2010).

2.2. Derin Öğrenme

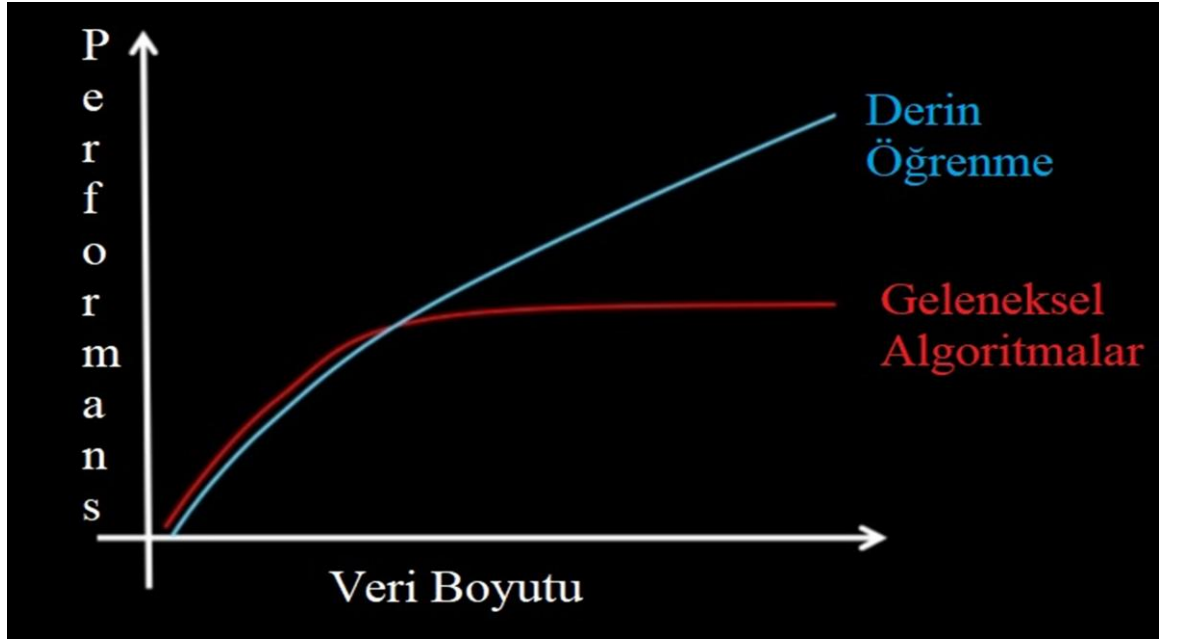
Gelişen teknolojiyle birlikte, YZ ve MÖ evriminin son halkası olan DÖ kavramı ortaya çıkmıştır. DÖ insan beyninin çalışma mantığını esas alan bir MÖ alt dalıdır. DÖ algoritmalarında sinir sisteminde olduğu gibi nöronlar birbiriyle iletişim halindedir. MÖ’de, görüntüdeki nesnelere tanımlamak, konuşmayı metne dönüştürmek; ürünleri, haberleri, yayınları kullanıcıların ilgi alanlarıyla eşleştirmek ve daha ilintili aramalar yapmak amaçlanır. Günümüzde, bu uygulamalar için DÖ denilen bir teknikten faydalanılmaktadır (LeCun, Bengio et al. 2015).

DÖ’de, büyük bir girdi kümesi bulunur. DÖ algoritması olması gereken çıktı ve tahmin edilen çıktı arasındaki farkı en aza indirmeye çalışır. Bu farkı azaltma çabaları neticesinde genel bir ilişki modeli oluşturur. Bu sürece öğrenme süreci denmektedir.

Günümüzde, tüm süreçler dijital ortama taşınmakta ve bu veriler dijital olarak saklanmaktadır. Teknolojinin gelişmesiyle birlikte, bilişim teknolojilerindeki ilerleme ve depolama alanlarının artması ve bununla ters orantılı olarak maliyetleri düşmüştür. Artık insanların yaptığı her bankacılık işlemi, her çevrimiçi görüşme, çektiği her fotoğraf, ziyaret ettiği her mekân bilinçli yahut bilinçsiz olarak kayıt altına alınmaktadır. Dolayısıyla saklanan veri boyutu gün geçtikçe artmaktadır. Bu hızlı artış literatüre Büyük Veri – BV (Big Data - BD) kavramını kazandırmıştır.

BV, geleneksel işleme tekniklerinin ve / veya algoritmaların üzerinde çalışmadığı yapısal ve yapılandırılmamış büyük karmaşık veri kümelerini ifade eder (Taylor-Sakya 2016).

DÖ, yeni bir alan olmasına rağmen, büyük verilerin işlenerek görüntü sınıflandırma, ses analizi, konuşma tanıma, video analizi ve doğal dil işleme gibi alanlarda başarılı sonuçlar vermesinden ötürü akademik çevrelerce ve endüstri toplumunda kendine hızla yer edinmiştir. Şekil 2.3'te geleneksel algoritmalara karşın DÖ algoritmalarının artan veri boyutuna göre başarısı gösterilmektedir.



Şekil 2.3. DÖ ve Geleneksel Algoritmaların Veri Boyutu Performansı (NG 2015)

Artan veri boyutuyla birlikte, gelişen teknoloji ve depolama ünitelerinin maliyetinin düşmesi ve asıl gelişme olan bilgisayarların Grafik İşlem Ünitesi- GIÜ (GPU - Graphical Processing Unit) birimlerinin bilgi işleme kabiliyetinin artmasıyla veri işleme ve öğrenme süreçleri kısalmıştır.

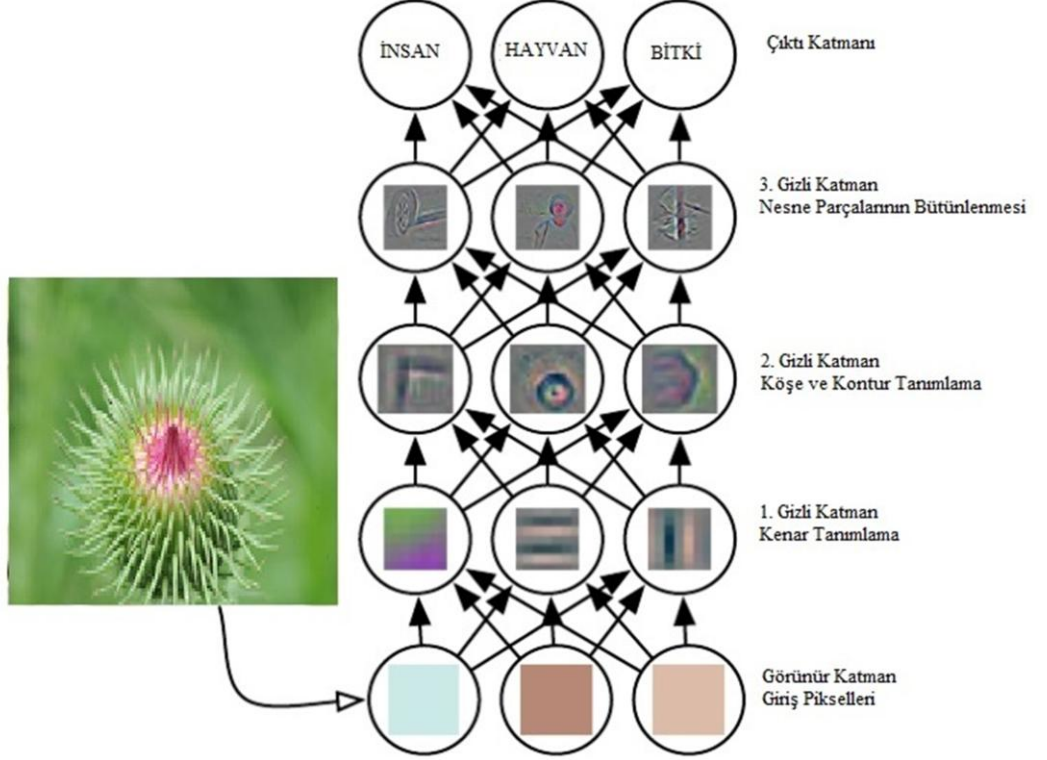
1950'lerde ortaya çıkan YZ kavramı, YSA ile MÖ'nin gelişimine önyak olmuştur. Artan veri boyutu ve gelişen teknolojiyle birlikte bu ağların daha da derinleşmesiyle birlikte DÖ kavramı ortaya çıkmıştır. Şekil 2.4'te DÖ'nin tarihi gelişimi gösterilmiştir.



Şekil 2.4. DÖ Evrimi (Nvidia 2016)

DÖ YSA'yı esas alır. Yeni bir kavram gibi sunulmasının nedeni, 30 yıl önceki gibi 2 veya 3 katmandan oluşan yapay sinir ağlarından ziyade 10 ile 100+ katmandan oluşan YSA ile daha da derinleşmiş olmasıdır.

Örneğin bir bitki fotoğrafının DÖ ağındaki süreci öncelikle giriş piksellerinden elde edilen tecrübelerin bir sonraki katmana aktarılmasıyla başlar. Daha sonra kenar tanıma yapılabilmesi için 2. katmanda bu tecrübeler kullanılır. Bir sonraki katmanda ise tanımlanmış olan kenarlardan köşe ve kontur tanımlaması yapılır. Bu süreçte nöronlar birbirlerine veri aktarımında bulunur. Sonraki katmanda ise nesne parçaları bütünlenir ve son katmana bilgi aktarılır. Son katmanda ise sınıflandırma yapılarak bitkinin hangi sınıfa ait olduğu belirlenir. Şekil 2.5'te klasik bir DÖ ağına çalışma mantığı gösterilmiştir.



Şekil 2.5. Klasik Bir DÖ Ağında Sınıflandırma

2.3. Evrişimli Yapay Sinir Ağları

YSA ağı türlerinden ileri beslemeli YSA veya çokça gizli katmanı bulunan ÇKA'lar Derin Yapay Sinir Ağları –DYSA (Deep Neural Networks DNN) olarak bilinir. EYSA ise bu türe dâhil olan ileri beslemeli derin sinir ağlarıdır (Liu, Fang et al. 2015).

Latince *convolvere* kelimesinden gelen convolution sözcüğü dilimize evrişim olarak geçmiştir. Matematiksel olarak bir evrişim değerinin üzerinden geçerek, iki fonksiyonun

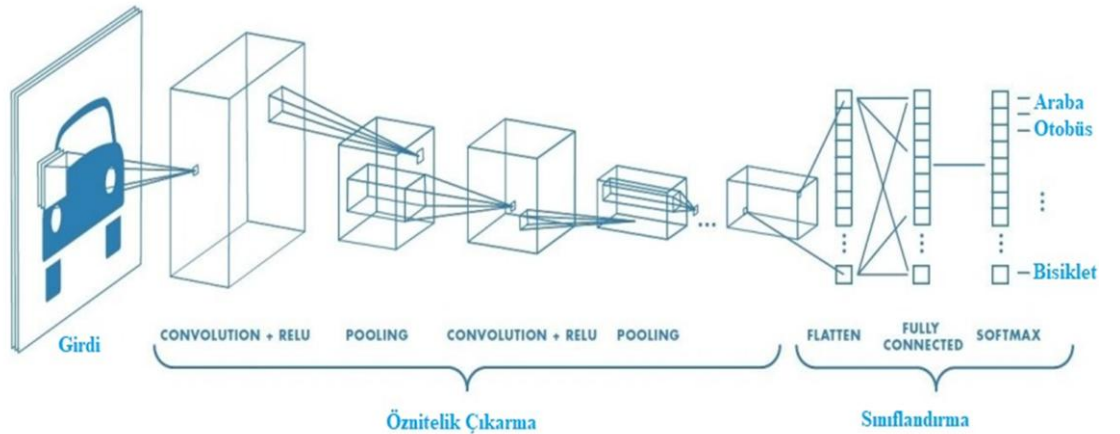
sonuçlarının ne kadar örtüştüğünü hesaplamak için kullanılır. Bu sebeple, evrişim işleminde 2 fonksiyon çarpma işlemine tabi tutulur.

EYSA, görüntüleri sınıflandırmak, benzer olanları eşleştirmek ve görüntü ve/veya videolardaki nesnelere tanımlayabilmek için kullanılan derin yapay sinir ağıdır. Tümörler, bitkiler, yüzler, trafik işaretleri gibi birçok nesneyi tanımlayabilen algoritmalarıdır.

Evrişimli ağların görüntü tanımadaki başarısı, DÖ'nin ilgi çekici olmaya başlamasının bir nedenidir. Otonom araçlar, uçangözler, tıbbi tanı ve tedavi uygulamaları, görme engelliler için üretilmiş uygulamalar gibi birçok alanda kullanılmakta olup gün geçtikçe bilgisayarlı görme alanındaki yerini güçlendirmektedir.

2.3.1. Standart Evrişimli yapay sinir ağı (CNN)

Genel olarak Standart EYSA 2 bölüme oluşur. İlk bölümde öznelik çıkarma işlemi yapılır. Her nöron, kendinden önceki nöronun ilgili kısmına bağlıdır ve yerel öznelik çıkarma işlemi yapar. İkinci bölümde ise sınıflandırma işlemi yapılır. Şekil 2.6'da EYSA yapısı gösterilmiştir.



Şekil 2.6. EYSA Yapısı (Shyamal Patel 2017)

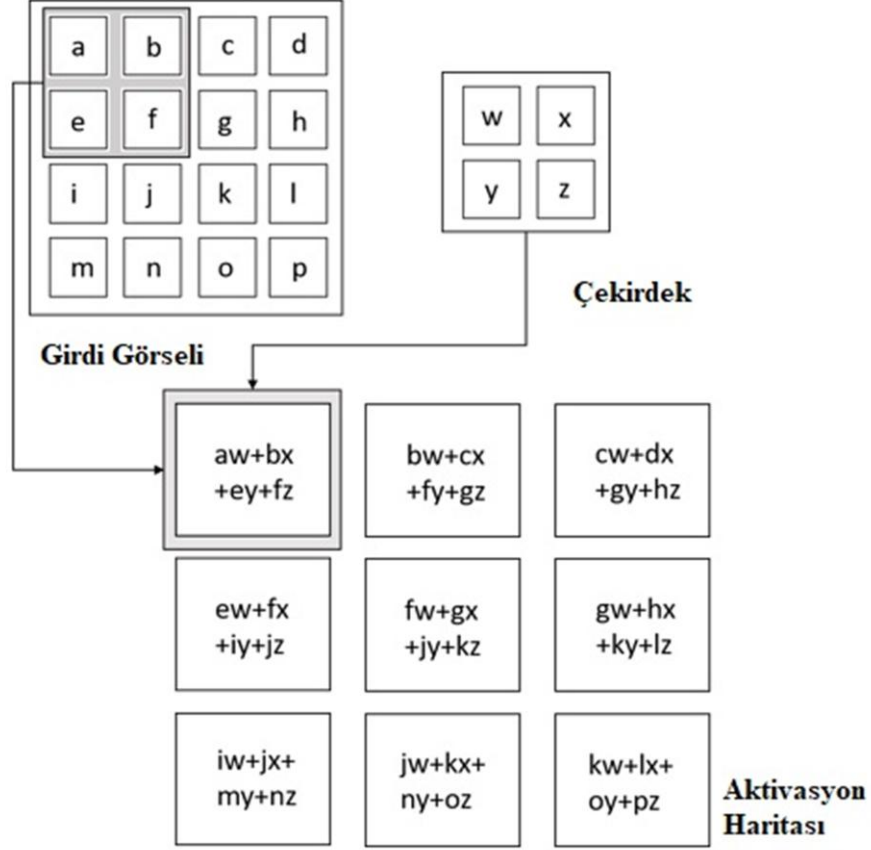
Evrişim katmanında ağın hesaplama işlemleri yapılır. EYSA'nın ana katmanıdır. Bu katmanda, iki matrisin çarpımı yapılır. Matrislerden birisi çekirdek olarak da bilinen öğrenilebilir özelliklerin matematiksel ifade edilmiş kümesidir. Diğer matris ise, girdiden kontrol edilmek üzere alınan küçük parçacıktır.

Devam eden süreçte çekirdek, girdi görüntüsünün genişliği ve yüksekliği boyunca kayar. Bu işlemle aktivasyon haritası olarak da bilinen görüntünün 2 boyutlu temsili gösterimi üretilir.

Girdi boyutları $W * W * D$, uzaysal boyut F , kayma boyutu S , dolgulama boyutu P ve çekirdek sayısı D_{out} olarak kabul edilirse çıktı boyutu aşağıdaki formülle hesaplanabilir (Goodfellow, Bengio et al. 2016).

$$W_{Out} = \frac{W - F - 2 * P}{S} + 1 \quad (2.5.1)$$

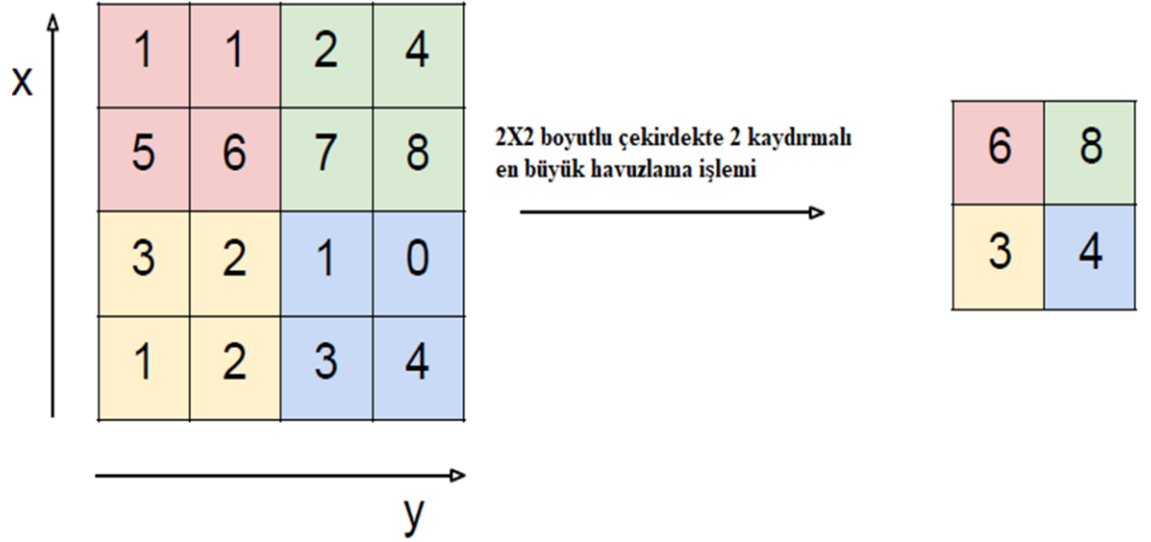
Şekil 2.7'de evrişim işlemi gösterilmiştir.



Şekil 2.7. Evrişim İşlemi (Goodfellow, Bengio et al. 2016)

Her evrişim katmanını takip eden bir aktivasyon fonksiyonu bulunur. Literatürde bazı kaynaklarda ayrı bir katman gibi sunulsa da, evrişim katmanının tamamlayıcı fonksiyonudur. Aktivasyon aşaması için basamak (step) fonksiyonu, doğrusal (linear) fonksiyon, Sigmoid fonksiyonu ve Doğrultulmuş Doğrusal Birim - DDB (Rectified Linear Units - RELU) fonksiyonu gibi aktivasyon fonksiyonları bulunmaktadır. DDB fonksiyonu doğrusal olmayan bir tahmin edici fonksiyondur.

Havuzlama işleminde amaçlanan uzaysal boyutun azaltılmasıdır. Bu azaltma ile yapılacak olan hesaplama ağırlık boyutlarının maliyetleri de düşürülmüş olur. Havuzlama işlemi için dikdörtgen komşuluk, L2 normlu dikdörtgen komşuluk, merkezi piksele olan uzaklığın ağırlıklı ortalaması gibi çeşitli fonksiyonlar bulunmaktadır. Bununla birlikte, en popüler havuzlama yöntemi komşuluğun en büyüğünü seçen en büyük havuzlama (Max Pooling) yöntemidir. Havuzlama işlemi her basamakta ayrı ayrı yapılmaktadır. Şekil 2.8’de 2 kaydırmalı havuzlama işlemi gösterilmiştir.



Şekil 2.8. Havuzlama İşlemi

Aktivasyon haritasının boyutları $W * W * D_l$, uzaysal boyut F_l , kayma boyutu S_l olarak kabul edilirse havuzlama işlemi sonucunda çıktı boyutu aşağıdaki formülle hesaplanabilir (Goodfellow, Bengio et al. 2016).

$$W_{Out} = \frac{W - F}{S} + 1 \quad (2.5.2)$$

Havuzlama katmanının devamında yer alan Tamamen Bağlı Katman (Fully Connected Layer) girdi ağırlıkları gizli katmanlara dağıtılarak ağırlık yükü hafifletilir. Bu aşamada

ağda ezberleme (overfitting) görülebilir. EYSA'da bu sorun Geri Yayılım (Backpropagation) algoritmasıyla çözülür.

Nihai süreçte çıktı üretilmeden evvel, tamamen bağlı katmandan gelen ağırlıklar Softmax katmanına iletilir. Softmax katmanı bir aktivasyon fonksiyonu olan Softmax fonksiyonunu kullanır. Softmax fonksiyonu kendisine gönderilen ağırlıkları kullanarak 0 ve 1 arasında olasılıksa bir değer üretir (Liu, Fang et al. 2015).

2.3.2. Derinlemesine ayrıştırılabilir evrişimli sinir ağları

Bölüm 2.3.1'de de bahsedildiği üzere evrişim işlemi YSA' da çok önemli bir işlemdir. Günümüzde genişletilmiş evrişim, transpoze edilmiş evrişim ve derinlemesine ayrıştırılabilir evrişim gibi birçok türü bulunmaktadır. DAEYSA da EYSA'nın bir türüdür.

Görüntü sınıflandırmada çeşitli EYSA kullanılmaktadır. Bunun 2 sebebi vardır:

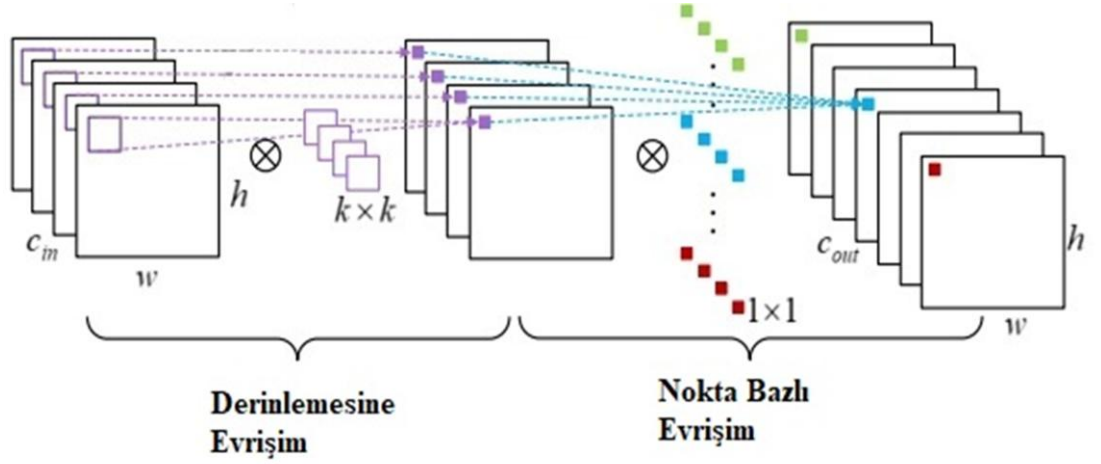
- Daha az parametreye sahiptirler ve bu da ezberleme riskini azaltır.
- Daha az hesaplama maliyetleri vardır ve bu da az kaynak kullanımı anlamına gelmektedir. Özellikle mobil uygulamalarda tercih edilmektedir (Howard, Zhu et al. 2017, R. Gavai, A. Jakhade et al. 2017) .

DAEYSA'da farklı olarak uzaysal bilginin genişlik ve yükseklik ve derinlik anlamında kanallara ayrılmasıdır. EYSA, özellik haritalarını giriş kanalları sayısı üzerinde birleştirirken, DAEYSA'da eklemeyen önce başka bir 1x1 evrişim işlemi gerçekleştirilmektedir.

TensorFlow ve Keras gibi DÖ kütüphanelerinde genellikle “ayrıştırılabilir evrişim”

olarak adlandırılır. Girdinin her bir kanalı üzerinde bağımsız olarak gerçekleştirilen derinlemesine evrişimi (Depthwise Convolution) takip eden bir nokta bazlı evrişim (Pointwise Convolution) yani 1×1 evrişim bulunmaktadır. Bu işlem, görüntü işleme topluluğunda genellikle “ayrılabilir evrişim (Spatial Convolution)” olarak da bilinen, uzamsal olarak ayrılabilir bir evrişimle karıştırılmamalıdır (Chollet 2016).

Şekil 2.9’da derinlemesine ayrıştırılabilir yapay sinir ağının yapısı gösterilmiştir.



Şekil 2.9. Derinlemesine Evrişim ve Nokta Bazlı Evrişim İşlemi

DAEYSA modellerine Google tarafından desteklenen Xception ve Mobilenet örnek verilebilir.

2.3.3. EYSA ve DAEYSA’ nın maliyet bazında karşılaştırılması

Örneğin girdi boyutu $G * G * K$ olan 3 boyutlu bir görüntüye C sayıda $F * F * K$ boyutlu çekirdekler uygulanır ve evrişim işlemi tamamlanırsa çıktı boyutu $A * A * C$ olmaktadır. Şekil 2.10’da evrişim işlemi temsili olarak gösterilmiştir.

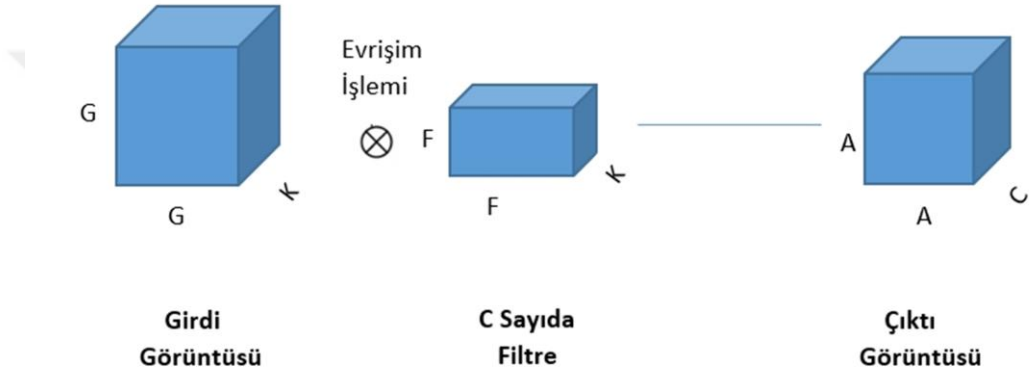
G : Girdi görüntüsünün eni ve boyu

F : Uygulanacak filtrenin eni ve boyu

K : Kanal Sayısı (RGB görüntüler için 3'tür)

C : Uygulanacak çekirdek (kernel veya filtre) sayısı

A : Çıktı görüntüsünün eni ve boyu



Şekil 2. 1. Evrişim İşlemi

Bir evrişimde yapılan çarpma işlemi filtre boyutuna eşittir.

$$\text{Çarpma Sayısı} = F * F * K \quad (2.5.3.1)$$

C sayıda filtre bulunduğu ve bu filtre çıktı görüntüsünde dikey ve yatay olarak kaydığından her evrişim için toplam çarpma sayısı;

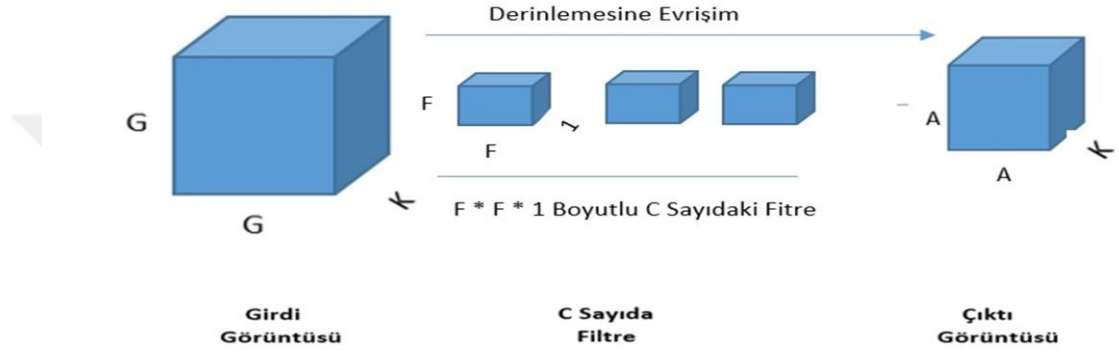
$$\text{Çarpma Sayısı} = C * A * A \quad (2.5.3.2)$$

Bu yüzden normal evrişim işleminde toplam çarpma sayısı;

$$\text{Toplam Maliyet} = C * K * A^2 * F^2 \quad (2.5.3.3)$$

olmaktadır (Chollet 2016, Howard, Zhu et al. 2017).

Yukarıda bahsedildiği gibi, derinlemesine ayrıştırılabilir yapay sinir ağları bir derin evrişim ve bir nokta bazlı evrişimden oluşmaktadır. Şekil 2.11’de derinlemesine evrişimli yapay sinir ağında derin evrişim işlemi temsili olarak gösterilmiştir.



Şekil 2. 7. Derinlemesine Evrişim İşlemi

Derinlemesine evrişim aşamasında, evrişim işlemi EYSA’ndaki gibi K farklı kanala değil sadece tek bir kanala uygulanmaktadır. Bu sebeple filtre boyutu;

$$\text{Filtre Boyutu} = F * F * 1 \quad (2.5.3.4)$$

olmaktadır. Bu sebeple bir evrişimdeki toplam çarpma sayısı

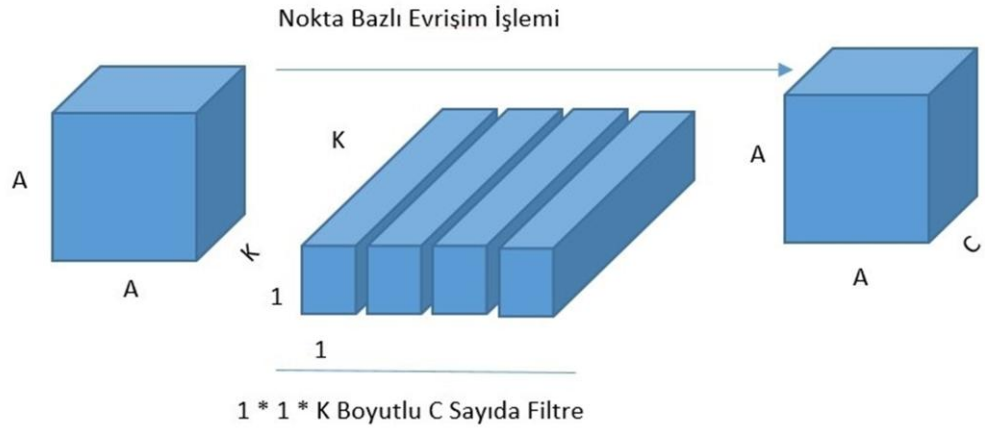
$$\text{Çarpma Sayısı} = F * F \quad (2.5.3.5)$$

olmaktadır. C sayıdaki filtre çıktı boyutunu tek tek gezdiğinden toplam çarpma işlemi sayısı;

$$\text{Çarpma Sayısı} = K * A^2 * F^2 \quad (2.5.3.6)$$

olmaktadır (Chollet 2016, Howard, Zhu et al. 2017).

Derinlemesine evrişimi takip eden nokta bazlı evrişimde ise 1×1 evrişim işlemi K kanallı filtreye uygulanmaktadır. Şekil 2.12’de DAEYSA nokta bazlı evrişim işlemi temsili olarak gösterilmiştir.



Şekil 2.12. Nokta Bazlı Evrişim İşlemi

Nokta bazlı evrişim aşamasında 1×1 evrişim işlemi K boyutlu filtreye uygulanmaktadır. Bu sebeple filtre boyutu;

$$\text{Filtre Boyutu} = 1 * 1 * K \quad (2.5.3.7)$$

olmaktadır. Bu sebeple bir evrişimdeki toplam çarpma sayısı

$$\text{Çarpma Sayısı} = 1 * K \quad (2.5.3.8)$$

olmaktadır. C sayıdaki filtre çıktı boyutunu tek tek gezdiğinden toplam çarpma işlemi sayısı;

$$\text{Çarpma Sayısı} = K * A^2 * C \quad (2.5.3.9)$$

olmaktadır.

DAEYSA, derinlemesine evrişimi takip eden nokta bazlı evrişim işleminden oluştuğundan;

$$\text{Toplam Maliyet} = (K * A^2 * F^2) + (K * A^2 * C) \quad (2.5.3.10)$$

olmaktadır (Chollet 2016, Howard, Zhu et al. 2017).

Toplam maliyet açısından kıyaslayacak olursak DAEYSA'nın EYSA'na oranı;

$$\text{Toplam Maliyet} = \frac{1}{C} + \frac{1}{F^2} \quad (2.5.3.11)$$

formülüyle bulunmaktadır.

Örneğin, filtre sayısı $C = 100$ ve filtre boyutu $F = 512$ alacak olursak, derinlemesine evrişimli yapay sinir ağlarının evrişimli sinir ağlarına göre yaklaşık 100 kat daha az hesaplama maliyeti bulunduğunu hesaplamaktayız.

2.4. ImageNet Veri Seti ve ImageNet Büyük Ölçekli Görsel Tanıma Yarışması

WordNet İngilizce dili için oluşturulmuş geniş bir leksik veri tabanıdır. İsimler, fiiller, sıfatlar ve zarflar için her biri ayrı bir kavramı ifade eden eş anlamlı kümeler halinde gruplandırılmıştır. Bu kümeler kavramsal-anlamsal ve leksik ilişkiler ile birbirine bağlanmaktadır. WordNet' te 100.000' in üzerinde veri kümesi bulunmakta ve bu kümelerin 80.000 adedini isimler oluşturmaktadır.

ImageNet veri tabanı, görüntü tanımlama yazılımlarında ve akademik arařtırmalarda kullanılmak üzere WordNet hiyerarřisine uygun olarak hazırlanmıř olan devasa bir veri tabanıdır. ImageNet veri tabanı günden güne geliřmekte ve her kümede 1000 görüntünün bulunması amaçlanmaktadır.

Güncel verilere göre 15 milyondan fazla görüntü verisi etiketlenerek veri tabanına dâhil edilmiřtir (Krizhevsky, Sutskever et al. 2012, Russakovsky, Deng et al. 2014).

ImageNet, “elma” ,“otomobil” gibi kategorilerden oluřan ve her kategorisinde yüzlerce görüntü baęlantısı bulunan bir veri setidir. ImageNet yaklařık 22.000 farklı kategoriye sahiptir. 2019 verilerine göre her düęümde 500’ün üzerinde görüntü verisi bulunmaktadır. řekil 2.13’te ImageNet’te bulunan ayçiçeęi bitkisine ait görüntüler gösterilmektedir.



řekil 2.13. ImageNet Sunflower Görüntüleri (ImageNet 2019)

Bu görüntüler indirilebilir durumda değildir. Görüntülere ait bağlantılar ve öngösterimler (thumbnail) ImageNet veri tabanında barınmaktadır. Bu görüntüler doğrudan ImageNet'e ait olmasa da ImageNet'ten ücretsiz olarak erişilebilir durumdadır (ImageNet 2016).

2010 yılından itibaren ImageNet tarafından, yazılımların nesnelere tespit etmek ve sınıflandırmak ImageNet Büyük Ölçekli Görsel Tanıma Yarışması – IBÖGTY (ImageNet Large Scale Vision Recognition Competition - ILSVRC) yarışması düzenlenmektedir. Çeşitli üniversitelerdeki görüntü işleme üzerine çalışan ekipler bu yarışmaya katılarak aşağıdaki alanlarda eğittikleri ağları yarıştırmaktadır.

- Görüntü Sınıflandırma: Görüntüde bulunan nesnenin hangi sınıfa ait olduğunu bulmaktır.
- Tekil Nesne Konumlandırma: Görüntü sınıflandırmaya ek olarak sınıflandırılan görüntünün bir örneğinin etrafına kutu çizmektir.
- Nesne tanımlama: Görüntü sınıflandırmaya ek olarak sınıflandırılan görüntünün etrafına kutu çizmektir.

2.5. Derin Öğrenme İle Görüntü Sınıflandırma Çalışmaları

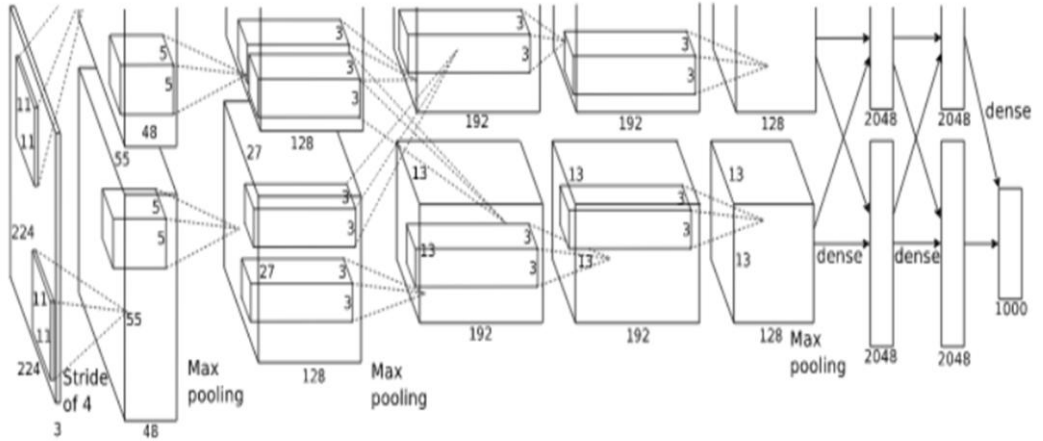
Aşağıda irdelenen tüm görüntü sınıflandırma çalışmalarında ImageNet veri setinin bir alt kümesi kullanılmıştır. Bu veri seti;

- Her kategorisinde 1000 görüntü bulunan yaklaşık 1000 kategoriden oluşmaktadır.

- Bu görüntülerin 1,2 milyon adedi ağı eğitimi için, 50.000 adedi doğrulama için ve 150.000 adedi test için kullanılmıştır.

2.5.1. AlexNet

Krizhevsky et al. 2012’de geliştirmiş oldukları AlexNet mimarisi DÖ dünyasında kilometre taşlarından sayılmaktadır. AlexNet mimarisi 5 evrişimli ve 3 tamamen bağlı katmandan oluşmaktadır. 11x11 boyutunda filtreler kullanılmış ve adım kayma sayısı 4 olarak belirlenmiştir Bu çalışmada aktivasyon fonksiyonu olarak DBB kullanılmış ve görüntü işleme alanında standartlaşmıştır. Bu çalışma ile aktivasyon fonksiyonu olarak Tanh fonksiyonunun yerine DBB fonksiyonunun tercih edilmesiyle benzer sonuçların 6 kat daha hızlı elde edildiği görülmüştür. Yerel tepki normalleştirme (Local Response Normalization) ve örtüşen havuzlamaların (Overlapping Pooling) kullanıldığı bu çalışmada kaymalar (stride) çekirdek boyutundan daha küçüktür (Krizhevsky, Sutskever et al. 2012). Şekil 2.14’te AlexNet mimarisi gösterilmiştir.



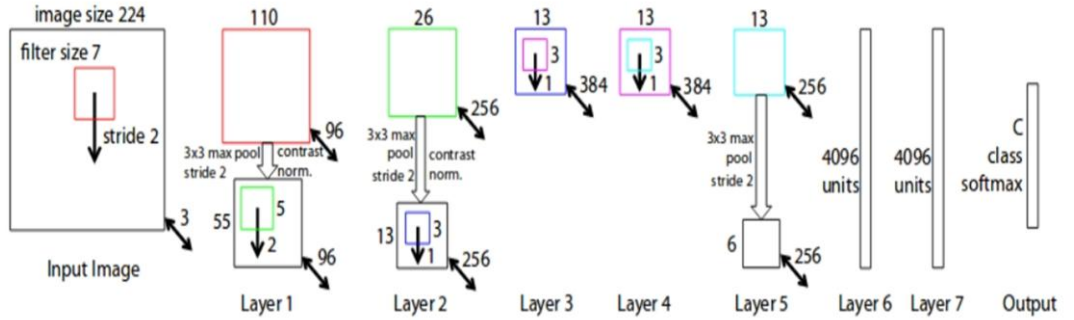
Şekil 2.14. AlexNet Mimarisi (Krizhevsky, Sutskever et al. 2012)

Yapılan çalışmada 1.2 milyon görüntüden 60 milyon adet parametre üretilmiştir. Ezberlemeyi engellemek için düşürme tamamen bağlı katmanlarda kullanılan düşürme

(Drop Out) yöntemi kullanılmıştır. Bu çalışma ile %15,3 hata oranını yakalamayı başaran Krizhevsky et al. 2012 IBÖGTY'nın sınıflandırma dalında birincisi olmuştur (Krizhevsky, Sutskever et al. 2012, Özkan İNİK 2017) .

2.5.2. ZFNet

2013'te Zeiler et al. tarafından geliştirilen bu modelde, AlexNet üzerinde bazı iyileştirmeler yapılmıştır. AlexNet'teki giriş katmanlarındaki $7 * 7$ ile $11 * 11$ boyutlu filtre boyutları $2 * 2$ ile $4 * 4$ olarak değiştirilmiştir. Havuzlama katmanında ise 2 adımlı kayma kullanılmıştır (Özkan İNİK 2017). Şekil 2.15'te ZFNet mimarisi gösterilmiştir.

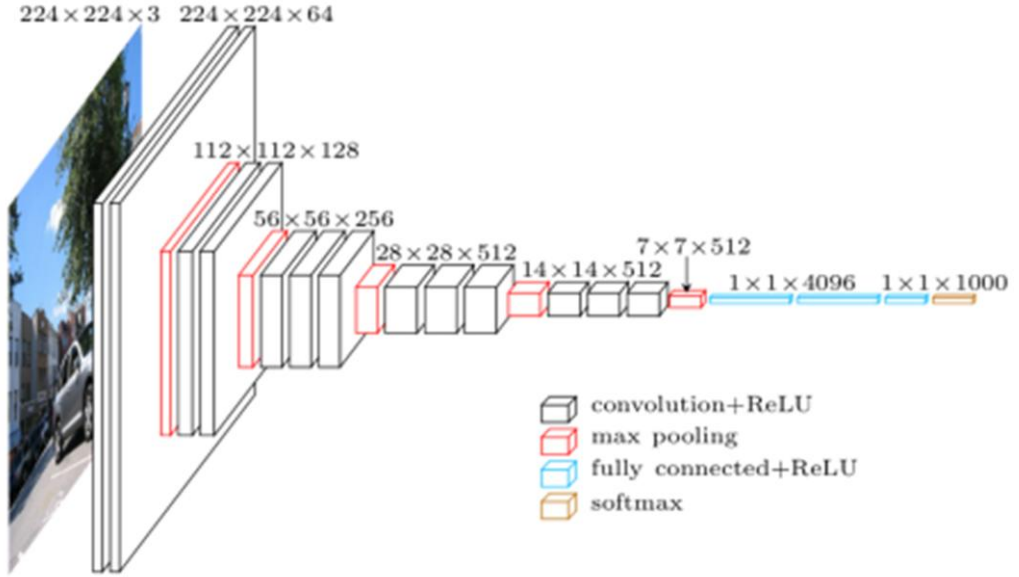


Şekil 2.15. ZFNet Mimarisi (Zeiler and Fergus 2013)

GTX580 Gİ kullanılan bu çalışmada eğitim süreci 12 gün sürmüştür. Bu model kullanılarak yapılan nesne tanımlamalarda hata oranı %11,2'ye düşürülmüştür. Bu model 2013 IBÖGTY'nın sınıflandırma dalında birincisi olmuştur (Zeiler and Fergus 2013).

2.5.3. VGGNet

Simonyan (2014) et al. tarafından geliştirilen bu modelde, derinlik kavramı ilk defa literatüre kazandırılmıştır. 3×3 boyutlu filtreler kullanılarak, girdi görüntüsü evrişim katmanından geçirilmektedir. Adım kaydırma işlemi 1 ile sınırlandırılmıştır. Boyutsal havuzlama işlemi 5 katmanlı en büyük havuzlama işlemiyle yapılmaktadır. Evrişim katmanını takiben tamamen bağlı katmanlar bulunmaktadır. Tüm gizli katmanlarda aktivasyon fonksiyonu olarak ReLu aktivasyon fonksiyonu kullanılmaktadır. Adım kaydırma işleminin küçük tutulmasıyla, nihai parametre sayısının azaldığı gözlemlenmiştir (Simonyan and Zisserman 2014). Bu ağ farklı derinliklerde test edilmiştir. VGGNet ağ mimarisi Şekil 2.16'da, farklı derinliklerdeki test sonuçları Şekil 2.17'de gösterilmiştir.



Şekil 2.16. VGGNet Mimarisi (Simonyan and Zisserman 2014)

ConvNet Configuration					
A	A-LRN	B	C	D	E
11 weight layers	11 weight layers	13 weight layers	16 weight layers	16 weight layers	19 weight layers
input (224×224 RGB image)					
conv3-64	conv3-64 LRN	conv3-64 conv3-64	conv3-64 conv3-64	conv3-64 conv3-64	conv3-64 conv3-64
maxpool					
conv3-128	conv3-128	conv3-128 conv3-128	conv3-128 conv3-128	conv3-128 conv3-128	conv3-128 conv3-128
maxpool					
conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256 conv1-256	conv3-256 conv3-256 conv3-256	conv3-256 conv3-256 conv3-256 conv3-256
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 conv1-512	conv3-512 conv3-512 conv3-512	conv3-512 conv3-512 conv3-512 conv3-512
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 conv1-512	conv3-512 conv3-512 conv3-512	conv3-512 conv3-512 conv3-512 conv3-512
maxpool					
FC-4096					
FC-4096					
FC-1000					
soft-max					

Şekil 2.17. Farklı Derinliklerdeki Test Sonuçları (Simonyan and Zisserman 2014)

A ve B ağlarında yerel tepki normalleştirilmenin gerekli olup olmadığı test edilmiştir. Bu test sonucunda A ve B ağlarında yaklaşık hata payı %10,4 olarak hesaplanmış ve yerel tepki normalleştirilmenin gereksiz olduğu sonucuna varılmıştır. Bu sebepler ilerleyen testlerde yerel tepki normalleştirme kullanılmamıştır. D ve C ağlarında ise 1x1 evrişim işleminin gerekli olup olmadığı test edilmiş ve sırasıyla %8,1 ve %8,5 hata payı

hesaplanmıştır. Dolayısıyla, devam eden testlerde daha büyük ağlarda 3x3 evrişim yapılmıştır. Araştırmacılar ayrıca farklı filtre boyutlarını test etmişler ve filtre boyutunu artırmanın ağın performansına olumlu etkisi olduğunu gözlemlemişlerdir. E ağında ise farklı boyutlardaki (256, 384, 512) girdiler test edilmiş ve hata payı %7,5'e düşürülmüştür. E ağında çoklu kırpma (Multiple Croppings) yöntemi uygulanarak hata payı %7,1'e kadar düşürülmüştür. D ve E ağında yapılan işlemler aynı anda uygulanarak nihai sonuçta hata payı %6,8'e kadar düşürülmüş ve bu sonuçla yarışmada 2.lik elde edilmiştir (Simonyan and Zisserman 2014). 2014 IBÖGTY' nın sınıflandırma dalında birincisi GoogleNet olmuştur.

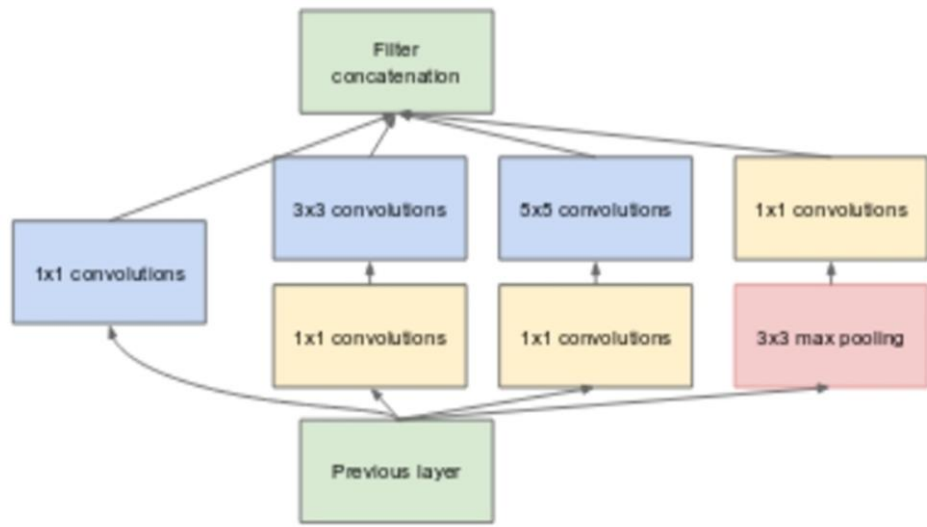
2.5.4. GoogleNet

Szegedy et al. tarafından geliştirilen bu ağ Google tarafından desteklenmektedir. Bu çalışmayla birlikte literatüre daha büyük ağların daha iyi sonuçlar verdiği sonucu kazandırılmıştır. Daha büyük ağlar elde etmek için iki yöntem bulunmaktadır. Bunlardan ilki ağın derinliğini artırmak, ikincisi ise genişliğini (her katmandaki işlem sayısı) artırmaktır. Ağın derinliğini artırmak parametre sayısının artışına sebep olmakta ve bu da ezberlemeye yol açmaktadır. Genişliği artırmak ise hesaplama maliyetindeki artışa yol açmaktadır (Szegedy, Liu et al. 2015).

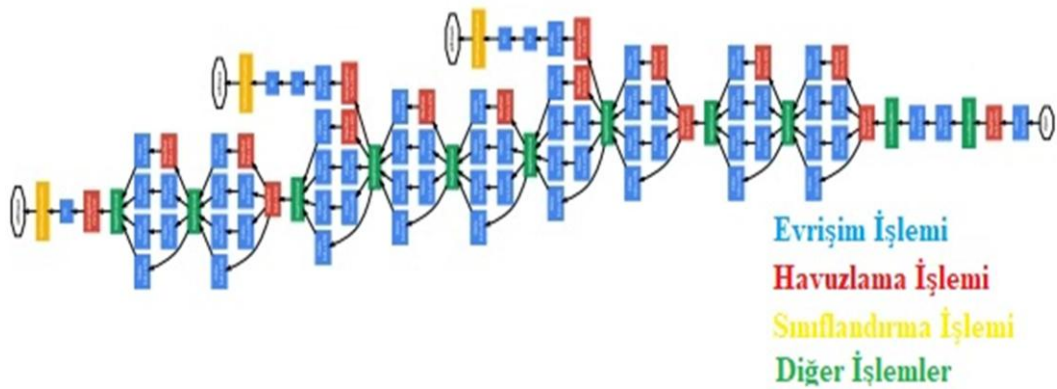
Bu mimari ardışık evrişim ve havuzlama katmanları düşüncesinden uzaklaşan ilk CNN modellerinden birisidir. Szegedy et al. artan maliyeti düşürmek için birbirine paralel olarak bağlı modüller kullanmıştır (Özkan İNİK 2017).

Ağda farklı olarak klasik evrişim katmanları inception katmanı adı verilen 3x3 evrişim, 5 x5 evrişim, 3x3 maksimum havuzlama ve 1x1 evrişimden oluşan katmanlar kullanılmıştır. Her evrişim birbirinden bağımsız çalışmakta ve bir önceki evrişimin aktivasyon haritası üzerinde işlem yapmaktadır Sürecin sonunda ise aktivasyon

haritaları birleştirilmektedir. Szegedy et al. artan maliyeti düşürmek için birbirine paralel olarak bağlı modüller kullanmıştır. Şekil 2.18’de inception katmanı, Şekil 2.19’da GoogleNet mimarisi gösterilmiştir.



Şekil 2.18. Inception Katmanı (Szegedy, Liu et al. 2015)



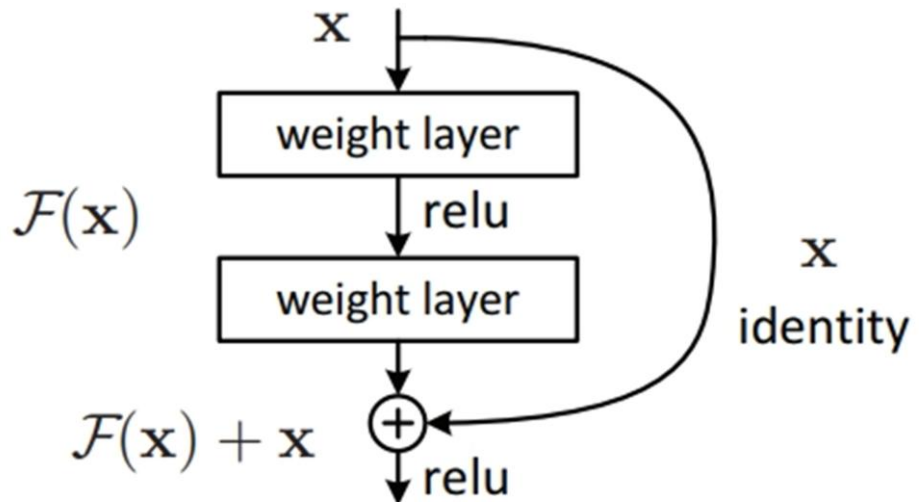
Şekil 2.19. GoogleNet Mimarisi (Szegedy, Liu et al. 2015)

GoogleNet mimarisi 2014 IBÖGTY'nın sınıflandırma dalında %6.67 hata payını yakalayarak birincisi olmuştur.

2.5.5. ResNet

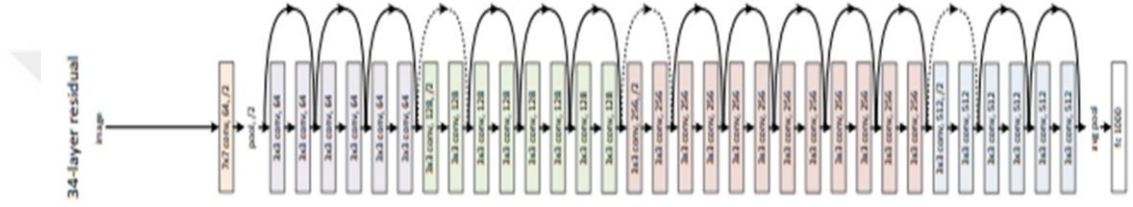
He et al. tarafından geliştirilen bu model Microsoft tarafından desteklenmektedir. ResNet mimarisi günümüze kadar tasarlanan mimariler arasında en derinidir. 152 katmandan oluşmaktadır. Bu mimaride artık (Residual) bloklar bulunmaktadır. Her artık blokta, x girişine müteakip evrişim-ReLu-evrişim katmanından sonra bir $F(x)$ sonucu ortaya çıkmaktadır. Daha sonra asıl x girişine eklenen bu $F(x)$ sonucuyla $H(x) = F(x) + x$ olarak ifade edilmektedir (He, Zhang et al. 2015, Özkan İNİK 2017).

Ağda $F(x)$ girişiyle her evrişim katmanından bir şeyler öğrenilmekte ve hata oranları ağ boyunca bir sonraki katmana iletilmektedir. Şekil 2.20'de artık blok temsili olarak gösterilmektedir.



Şekil 2.20. Artık Blok (He, Zhang et al. 2015)

İlk 34 katmanında 3,6 milyar toplama ve çarpma işlemi bulunan ResNet mimarisi Şekil 2.21’de gösterilmiştir.



Şekil 2.21. ResNet Mimarisinin İlk 34 Katmanı (He, Zhang et al. 2015)

Araştırmacılar tarafından ağa daha fazla katman eklemenin modelin keskinliğini artıracığı ve daha düşük hata payı alınacağı düşünülmüştür. Fakat burada 2 problemle karşılaşmıştır.

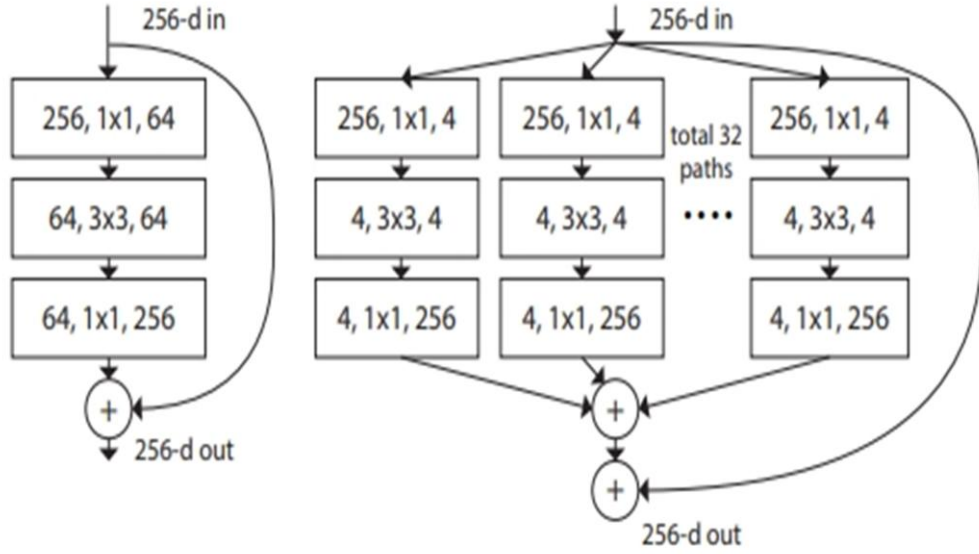
- 1- Sonsuz Gradyan Problemi (Vanishing gradient problem): Yığın normalizasyonu (Batch Normalization) gibi düzenleme fonksiyonlarıyla çözülmektedir.
- 2- Ağa daha fazla katman eklemenin ezberlemeye sebep olduğu ve ağın keskinliğini artırmadığı gözlemlendi.

ResNet mimarisi ResNet50, ResNet101 ve ResNet150 gibi çeşitli varyasyonlara sahiptir ve ön tanımlı ağırlıkları birçok kütüphane tarafından kullanılmaktadır. Facebook Detectron (Facebook 2018) projesi tarafından da kullanılan ResNet mimarisi bölütleme ve nesne tanımlama problemlerinde arka uç birim (backend) olarak kullanılmaktadır. ResNet mimarisi %3,57 hata payıyla 2015 IBÖGTY’nın sınıflandırma dalında birincisi olmuştur (He, Zhang et al. 2015).

2.5.6. ResNetXT

Xie et al. tarafından geliştirilen bu model, Facebook ArGe (Facebook Research) takımı tarafından desteklenmektedir. Inception (GoogleNet) topolojisine benzer olmakla birlikte ResNetXT mimarisi derinlik açısından farklı yaklaşımlar sunmaktadır. Ayrıca bu çalışma literatürde kardinalite (cardinality) kavramının kullanıldığı ilk çalışmadır.

Homojen ve çok bölümlü bir mimari olan çalışmada sadece birkaç hiper-parametre bulunmaktadır. Bu yöntem, kardinalite(dönüşümlerin boyutu) olarak adlandırdığımız yeni bir boyutun ortaya çıkmasına sebep olmuştur. Bu çalışmada derinlik ve genişlik boyutlarına ek olarak kardinalite de kullanılmaktadır(Xie, Girshick et al. 2016). Şekil 2.22’de ResNet ve ResNetXT mimarilerinin karşılaştırması



Şekil 2.22. ResNet Mimarisi Bloğu ve ResNetXT Mimarisi Bloğu (Xie, Girshick et al. 2016)

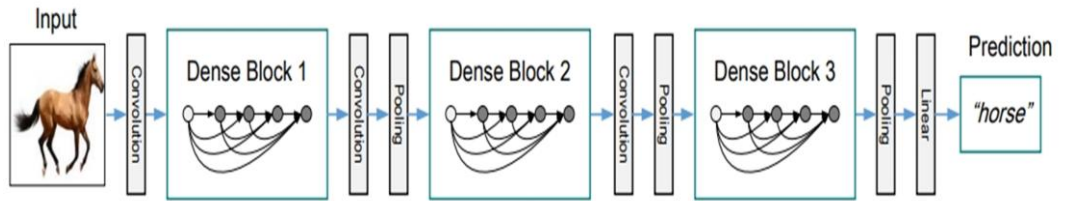
ResNetXT mimarisi %3,03 hata payıyla 2016 IBÖGTY'nın sınıflandırma dalında ikincisi olmuştur (Xie, Girshick et al. 2016).

2.5.7. DenseNet

Adını yoğun bağlı evrişimli sinir ağlarından alan DenseNet Huang et al. tarafından geliştirilmiştir. DenseNet birçok yönden ResNet' e benzese de 2 önemli değişiklikle ResNet' ten ayrılmaktadır.

- 1- ResNet' teki gibi öznitelik ağırlıkları birbirine eklenmeyip, öznitelikler birleştirilmektedir.
- 2- ResNet' te her katman bir önceki katmandan öğrenilen öznitelik bir önceki katmandan öğrenilen özniteliğe eklenmektedir. DenseNet' te ise farklı olarak her katmandan öğrenilen öznitelik, kendinden önceki tüm katmanlardan öğrenilen özniteliklerle birleştirilmektedir.

Huang et al. (2016) ağı yapısını bu cümlelerle özetlemektedir. “DenseNet katmanları, katman başına 12 filtre düşecek şekilde dar olarak inşa edilmiştir. Ağdaki “kolektif bilgiye” sadece küçük bir özellik haritası seti eklenmekte ve diğer özellik haritaları değiştirilmeden tutulmaktadır. Son katmanda ise sınıflandırıcı ile karar verilmektedir (Huang, Liu et al. 2016). Şekil 2.23' te 3 yoğun bloklulu DenseNet ağı gösterilmiştir.



Şekil 2.23. 3 Yoğun Bloklulu DenseNet Ağı (Huang, Liu et al. 2016)

Her blokta birden fazla evrişim bloğu bulunmaktadır. Bu evrişim blokları “Bottlenecks” olarak adlandırılmakta ve 3x3 evrişimlerden gelen anlamsız öznitelikleri azaltmaktadır (Huang, Liu et al. 2016).

Ağın derinliğine göre DenseNet-121, DenseNet-169, DenseNet-201, DenseNet-264 gibi farklı varyasyonlar bulunmaktadır. Şekil 2.24' te bu varyasyonlar gösterilmiştir.

Layers	Output Size	DenseNet-121	DenseNet-169	DenseNet-201	DenseNet-264
Convolution	112 × 112	7 × 7 conv, stride 2			
Pooling	56 × 56	3 × 3 max pool, stride 2			
Dense Block (1)	56 × 56	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 6$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 6$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 6$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 6$
Transition Layer (1)	56 × 56	1 × 1 conv			
	28 × 28	2 × 2 average pool, stride 2			
Dense Block (2)	28 × 28	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 12$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 12$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 12$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 12$
Transition Layer (2)	28 × 28	1 × 1 conv			
	14 × 14	2 × 2 average pool, stride 2			
Dense Block (3)	14 × 14	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 24$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 32$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 48$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 64$
Transition Layer (3)	14 × 14	1 × 1 conv			
	7 × 7	2 × 2 average pool, stride 2			
Dense Block (4)	7 × 7	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 16$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 32$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 32$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 48$
Classification Layer	1 × 1	7 × 7 global average pool			
		1000D fully-connected, softmax			

Şekil 2.24. Farklı DenseNet Mimarileri (Huang, Liu et al. 2016)

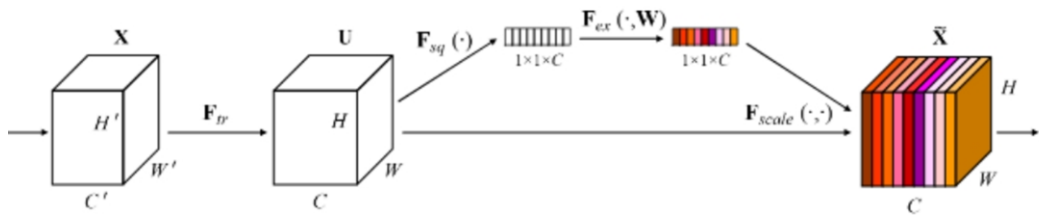
Yukarıdaki şemada 32 farklı özellik kullanılmıştır. Bu tür mimarilerde çok fazla eğitimin olması oluşan parametre sayısını azaltmaktadır. Örneğin, bir ResNet evrişim katmanı [96, 96, 96] boyutlu bir aktivasyon haritasına sahipken, DenseNet' in aynı şartlarda [32, 64, 96] boyutlu aktivasyon haritası içerdiği görülmüştür. Huang et al. (2016), her bir katmanın kendinden önceki tüm katmanlara ait özellikleri içerecek şekilde katmanları birleştirmenin gereksiz özelliklerin öğrenmesini engelleyeceğini sonuç olarak sunmaktadır (Huang, Liu et al. 2016).

Sonuç olarak, ResNet ağına göre aynı şartlar altında 3 kat daha az parametre üretilmektedir. ImageNet veri setinde ResNet' le yakın sonuçlara sahip olmasına

rağmen CIFAR 10, CIFAR 100, SVHN vb. diğer veri setlerinde gittikçe artan performanslar elde etmektedir (Huang, Liu et al. 2016).

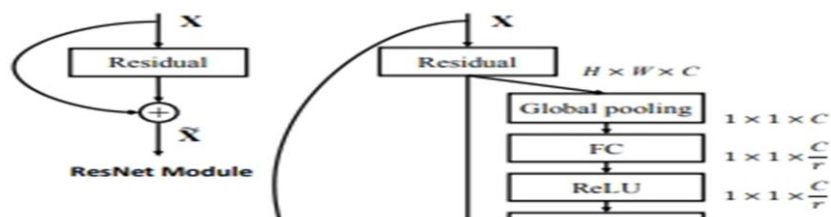
2.5.8. SE-Net

2017 yılına kadar yapılan tüm çalışmalarda doğruluğu artırmak için istifleme katmanlarında (stacking layers) iyileştirmeler yapılmıştır. Bu sebepler, evrişim işleminde yapılacak bir hata ile çok fazla veri kaybı olasıdır. Hu et al. yapmış oldukları çalışmada, “Aktivasyon haritasından tüm öğrendiklerim önemli mi?” sorusundan yola çıkmışlardır. Bu çalışmada, tamamen yeni bir mimari üretilmiştir. Kanallar arası



Şekil 2.25. SE-Net Mimarisi (Hu, Shen et al. 2017)

SE bloğu, her katmandaki önemli özellikleri ağırlandırarak bu ağırlıkları aktivasyon haritasına ekler. Sıkma (squeeze) aşamasında, küresel maksimum havuzlama işlemi yapılarak kanal bilgilerini toplanmakta ve olabilecek en düşük boyutta tutulmaktadır. Uyarma (excitation) işleminde ise hangi özelliğin önemli olduğuna karar verilir. Öğrenme işlemi ise 2 Tamamen Bağı katman arasında yerleştirilmiş ReLU katmanında yapılmaktadır. Aktivasyon fonksiyonu olarak kullanılan Sigmoid fonksiyonu, girdi verisine özgü kanal ağırlıklarını değerlendirmektedir (Hu, Shen et al. 2017, Duque, Santos et al. 2019). SE Şekil 2.26’da klasik ResNet bloğu ve önerilen SE-Net bloğu kıyaslanmıştır.



SE bloğu, SE-Net' e özgü değildir. Hu et al. tarafından ResNet, ResNeXt, Inception, Inception – Resnet gibi mimarilerde de test edilmiştir. Parametre, zaman ve maliyet bakımından cüzi denebilecek artışlar gözlenmiştir. SE-Net mimarisi %2,25 hata payıyla 2017 IBÖGTY'nın sınıflandırma dalında birinci olmuştur (Hu, Shen et al. 2017).

2.6. Derin Öğrenme Kütüphaneleri

DÖ'nin hızlı ve keskin sonuçlar vermesiyle birlikte, teknoloji dünyasında DÖ kütüphanelerin sayısı hızla artmaktadır. Bu bölümde çeşitli DÖ kütüphaneleri araştırılmış ve Bölüm 2.12'de çeşitli kıstaslara göre karşılaştırma yapılmıştır. Bu araştırmanın amacı, geliştirilecek projede olabilecek en iyi kütüphaneyi seçmektir. İncelenen tüm kütüphaneler açık kaynaklıdır. Python ve R gibi dilleri desteklemektedir.

2.6.1. TensorFlow

Tensorflow, GoogleBrain ekibi tarafından geliştirilmekte olan Python, C++ ve R gibi dillerle DÖ modelleri oluşturmaya olanak veren bir kütüphanedir. Tensorflow' un kullanıldığı bazı alanlar şunlardır:

1. Metin temelli uygulamalar: Dil tanımlama, metin özetleme
2. Görüntü tanımlama: Görüntü yakalama, yüz tanımlama, nesne tanımlama

3. Ses tanımlama
4. Zaman serisi analizleri
5. Video analizleri

Merkezi İşlem Ünitesi – MİÜ (Central Processing Unit - CPU) ve GIÜ ile çalışma olanağı sunmaktadır. Linux, MacOS, Microsoft ve Android işletim sistemleri üzerinde çalışabilmektedir (Tensorflow 2018).

2.6.2. Keras

Python dili ile geliştirilmiş olan Keras kütüphanesi, Tensorflow, Theano ve CNTK kütüphanelerinin en üst katmanında çalışmaktadır. Yeni başlayanlar için kullanıcı dostu arayüzler sunan Keras, “Makineler için değil, insanlar için API” sloganıyla DÖ kütüphaneleri arasında yerini almaktadır. Keras kütüphanesi modern EYSA ağırları modellerinden olan VGGNet, Inception, MobileNet gibi mimarileri desteklemektedir. Linux, MacOS ve Microsoft işletim sistemleri üzerinde çalışabilmektedir (Keras 2018).

2.6.3. PyTorch

2017 yılında Facebook tarafından geliştirilen PyTorch kütüphanesi Tensorflow, Theano ve CNTK gibi popüler kütüphanelerden farklı olarak, bünyesinde barındırdığı autograd bileşeniyle durgun graf(graph) yerine hareketli graf(graph) kullanmaktadır. Bu özelliği ile yazılan kodun derlenme aşamasında kontrol kullanıcıya bırakılmaktadır. Tensorflow kütüphanesine benzer olarak, görüntü tanımlama, ses tanımlama, metin tanımlama gibi tüm DÖ işlemleri desteklenmektedir. Linux, MacOS ve Microsoft işletim sistemleri üzerinde çalışabilmektedir (PyTorch 2018).

2.6.4. Caffe

Caffe kütüphanesi Berkeley Üniversitesi arařtırmacıları tarafından 2013 yılında geliştirilmiřtir. C, C++, Python ve Matlab dillerini destekleyen kütüphane ile basit regresyon, geniş yelpaze görüntü sınıflandırma, görüntü benzeřtirme siyam aęları, ses tanımlama ve robotik uygulamalarında kullanılmaktadır. Linux, MacOS ve Microsoft iřletim sistemleri üzerinde alıřabilmektedir (Berkeley 2014).

2.6.5. Theano

Theano kütüphanesi Montreal Üniversitesi arařtırmacıları tarafından 2017 yılında geliştirilmiřtir. Python diliyle geliştirilmiř olan bu kütüphanenin GIÜ desteęi buunmaktadır. Dinamik olarak C kodu üretebilmektedir. Linux, MacOS ve Microsoft iřletim sistemleri üzerinde alıřabilmektedir (Theano 2017).

2.6.6. DeepLearning4J

Skyminid mühendislik takımı tarafından 2014 yılında geliştirilmiřtir. Tensorflow gibi ok boyutlu diziler olan tensor'larla alıřabilmeyi desteklemektedir. Java ve C++ dilleri ile geliştirilmiřtir. MIÜ ve GIÜ ile alıřma olanaęı sunmaktadır. Linux, MacOS, Microsoft ve Android iřletim sistemleri üzerinde alıřabilmektedir (Deeplearning4j 2015).

2.6.7. Derin öğrenme kütüphanelerinin kıyaslanması

DÖ kütüphanelerinin karşılařtırması için Google arama trendleri, Github aktiviteleri, basılı yayınlar gibi kıstaslar kullanılmıřtır. Tüm bu veriler saęlayıcının kendi verisi olup, 01/04/2019 tarihi itibariyle elde edilmiřtir. izelge 2.4'te incelenen tüm kütüphanelere ait veriler sunulmuřtur.

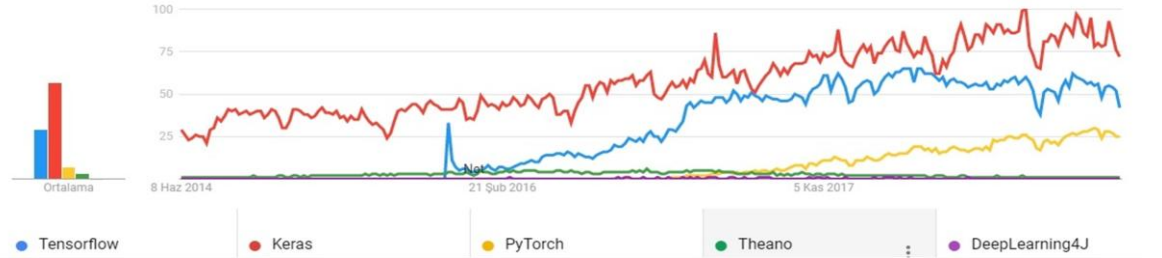
Çizelge 2.3. DÖ Kütüphanelerinin Kıyaslanması

Github Aktiviteleri	Google Arama Trendleri		Basılı Yayınlar			Destekçi	CUDA Desteği
	Takip Eden	Çalışan	Destekleyen	Son 5 Yıl	Arxiv Makaleleri		
8.584	75.270	2.018	45,07	336	385	1.339	Var
2.027	15.912	801	57,02	51	587	7.428	Var
1.262	6.905	1.000	20,13	110	57	231	Var
2.237	17.044	265	22,28	96	14	856	Var
591	2.484	332	46,53	42	58	653	Var
836	4.676	251	37,52	3	8	44	Var

	Yıldızlayan	128.937	41.697	28.702	28.253	8.815	10.861
İş İmkânı	Monster	2610	695	665	866	508	35
	Linkedin	1253	364	309	399	316	5
Kütüphane		Tensorflow	Keras	PyTorch	Caffe	Theano	DeepLearning4J

2.6.7.1. Google arama trendlerine göre kıyaslama

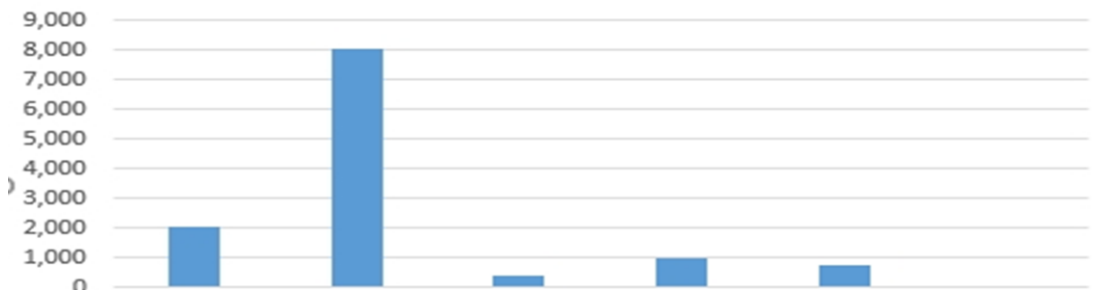
Son 5 yılda yapılan aramalara göre Tensorflow ve Keras kelimelerini içeren arama sayıları diğer kütüphanelerin önüne geçmektedir. Şekil 2.27’de Google arama trendleri gösterilmiştir.



Şekil 2.27. Derin Öğrenme Kütüphanelerinin Google Arama Trendleri (Google 2019)

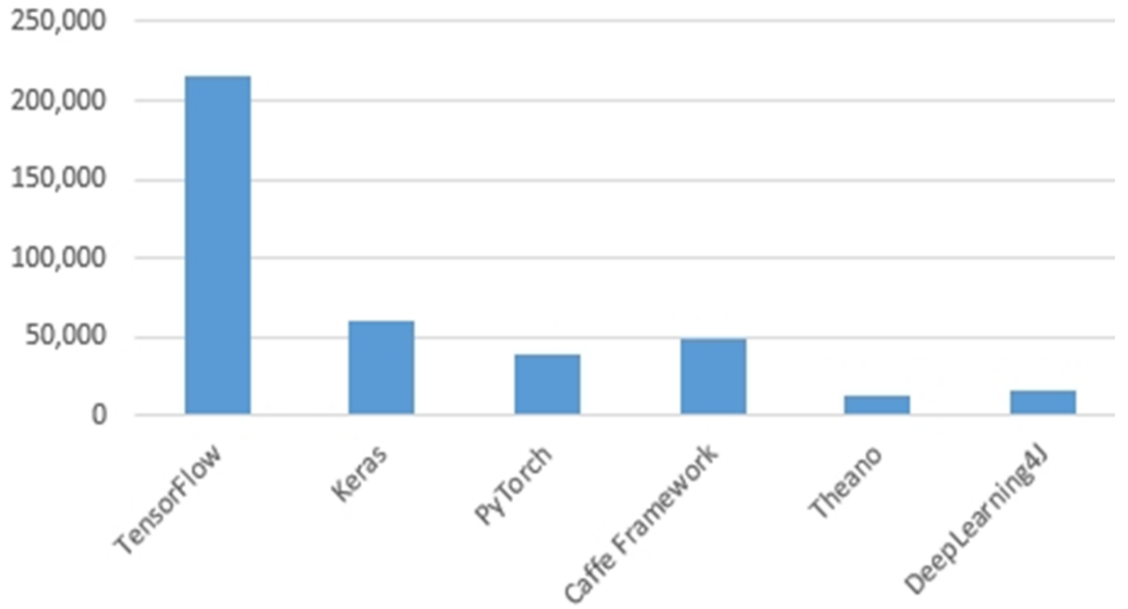
2.6.7.2. Basılı yayınlara göre kıyaslama

Amazon, Arxiv.org, ScienceDirect gibi akademik çalışmalar için kullanılan elektronik kütüphanelerde yayınlanan yayın sayılarına göre Tensorflow ve Keras kütüphaneler toplam yayın sayısı olarak öne çıkmaktadır. Şekil 2.28’de toplam yayın sayılarına göre kıyaslama gösterilmiştir.



2.6.7.3. Geliştirici ilgisine göre kıyaslama

Github geliştiricilerin projelerini yayınladıkları, yayınlanan açık kaynak projelere katkı sundukları bir topluluktur. Github’da yapılan taramalara göre yine Tensorflow ve Keras kütüphanelerinin geliştiricilerin ilgisini çektiği görülmektedir. İzleyen, takip eden, üzerinde çalışan ve destekleyen kişi sayısı verileri toplanarak Github popülaritesi elde edilmiştir. Şekil 2.29’ da Github popülaritesine göre kıyaslama gösterilmiştir.



Şekil 2.29. Kütüphanelerin Github Popülaritesi (Github 2019)

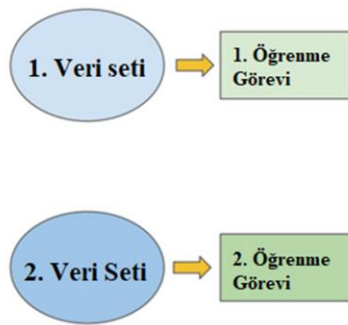
2.7. Transfer Öğrenmesi

Transfer öğrenmesi, bir konuda veya bir materyal grubuyla yapılan bir öğrenmeden elde edilen tecrübelerin başka bir konudaki performansı veya diğer ilgili materyallerle olan katkısını etkilediği zaman gerçekleşir (N. Perkins and Salomon 1999).

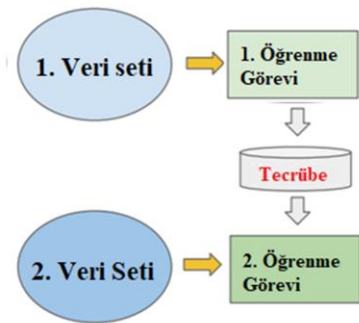
Transfer öğrenmesi, bir problem çözülürken elde edilen tecrübeyi veya bilgiyi saklayarak daha sonra bu bilgilerden faydalanarak benzer problemlerin çözümünü esas alan bir alandır. Transfer öğrenmesi, psikoloji literatüründe önemli bir yer edinmekte ve bu mantık MÖ'nde de kullanılmaktadır.

Transfer öğrenmesi, daha önceden belli amaçlar için eğitilmiş bir modelin benzer bir problemin çözümünden tekrar kullanılmasıdır. DÖ alanında bu kadar popüler olmasının sebebi, derin sinir ağlarını görece daha az veri ile eğitebilmemize olanak tanımaktadır. Örneğin, aynı cinse 2 bitki türünden birincisine ait sap uzunluğu, yaprak genişliği, renk gibi bilgiler öğrenildikten sonra bu bilgiler 2. türü tanımlarken kullanılabilir. Şekil 2.32' de klasik MÖ ve transfer öğrenme destekli MÖ karşılaştırılmıştır.

Geleneksel Makine Öğrenmesi



Transfer Öğrenmesi



Şekil 2.30. Geleneksel MÖ ve Transfer Öğrenmesi Arasındaki Farklar

2.8. Bilgisayarlı Görme Alanında Bitki Tanımlama Çalışmaları

Prasad et al. yapmış oldukları çalışmada, 102 kategoriden oluşan Oxford-102 (Nilsback and Zisserman 2006, Nilsback and Zisserman 2008) isimli veri setini ve 30 kategoriden oluşan KL University Flower Dataset (KLUFDF) veri setini kullanmışlardır. Bu veri setlerini birleştirerek, YSA, Adaboost, DYSA ve EYSA ile eğitmişlerdir. Evrişimli yapay sinir ağıyla eğitilen modelin eğitim ve test doğrulama sonuçlarının daha iyi olduğu gözlemlenmiştir. Diğer algoritmalara göre en iyi sonucu evrişimli yapay sinir ağlarıyla elde etmişlerdir. Bu çalışma sonucunda evrişimli yapay sinir ağının çiçeği doğru olarak sınıflandırma oranı %97,78 olarak gözlemlenmiştir (Prasad, et al. 2018). Çizelge 2.3'te bu çalışmaya ait sonuçlar sunulmuştur.

Çizelge 2.4. 4 Farklı Sınıflandırma Yönteminin Başarı Yönünden Karşılaştırılması

Sınıflandırıcı	Eğitim Veri Setiyle Test	Başka Veri Setiyle Test
Adaboost	%67.81	%62.91
YSA	%82.45	%78.63
DYSA	%89.74	%85.84
EYSA	%97.78	%93.98

Lakesar, çiçek fotoğrafları üzerine yapmış olduğu çalışmada Dalgacık Dönüşümü – DD (Wavelet Transform – WT) ve Gri Seviye Eş Oluşum Matrisleri – GSEOM (Gray Level Co-Occurance Matrix – GLCM) ile öznelik çıkarmış ve çıkarılan öznelikleri geri beslemeli YSA yapısı olan Çoklu Katman Algılayıcı – ÇKA (Multi Layer Perceptron - MLP) ile sınıflandırmıştır. Yapılan çalışmada çiçek görüntülerinin parametrelinin renk ve doku olduğu ve bu parametrelerin tek başına sınıflandırmada kullanılmasındansa, bu parametrelerin kombine edilerek kullanılmasının çiçek türünün daha iyi tanımlanmasında etki ettiği gözlenmiştir (Lakesar 2018).

Otomatik bitki tanımlama hem botanik sınıflandırma hem de bilgisayar bilimi için zor ve ilginç bir alandır. Pacifico et al. otomatik bitki tanımlama üzerine yapmış oldukları çalışmada karar ağaçları, NB sınıflandırıcı, K en yakın komşu (k=3, 4 ve 5 için), radyal temelli destek vektör makineleri ve lineer destek vektör makineleri yöntemlerine karşılık geri beslemeli ÇKA'yı test etmişlerdir. Bu çalışmada 100 bitkiden oluşan yaprak ve tohumların örnek görsellerini barındıran University Of California Irvine Iris veri seti kullanılmıştır. Özniteliklerin tek tek kullanılmasındansa ikili olarak kombine edilerek kullanılmasının daha iyi sonuçlar verdiği gözlemlenmiştir. Akabinde tüm özniteliklerin(yaprak açıklığı, yaprak şekli ve yaprak dokusu) kombine edilmesiyle en iyi sonuçlara ulaşılmıştır. Araştırma neticesinde ÇKA ile %98, 91 ortalama doğru sınıflandırma yapılmış ve ÇKA ile en iyi sonuçlar elde edilmiştir (Pacifico, Macario et al. 2018).

Bitkiler üzerine yapılan bir diğer çalışmada ise 102 kategorili ve 8189 görselden oluşan Oxford-102(Nilsback and Zisserman 2006, Nilsback and Zisserman 2008) bitki veri seti kullanılmıştır. Bölütleme (segmentation) aşamasında görsellerin arka planları yeni bir yaklaşım olan renk ayrıştırma yöntemiyle temizlenmiş ve gürültü sıfıra indirgenmiştir (Mavani, Gurnani et al. 2017). Bu çalışmada, sınıflandırma aşaması için evrişimli YSA mimarilerinden olan GoogleNet ve AlexNet mimarileri kullanılmıştır. Sınıflandırma sonuçları GoogleNet için en iyi 1 keskinlik değeri %47,15 ve en iyi 5 keskinlik değeri %69,17 ölçülürken, AlexNet için en iyi 1 keskinlik değeri %43,39 ve en iyi 5 keskinlik değeri %68,68 olarak ölçülmüştü. Rastgele sınıflandırma sonucu ise keskinlik oranı %98 olarak ölçülmüştür (Gurnani, Mavani et al. 2017).

Bitkilerin doğal ortamında tanımlamasını amaçlayan bir diğer çalışmada ise, renk doku ve şekil özniteliklerinin çıkarılması için en iyi yöntemin bulunması amaçlanmıştır. Bu çalışmada Kuzey Kore coğrafyasında yetişen 300 türe ait 29463 bitki görüntüsü 2011 - 2014 yılları arasında toplanmış ve veri kümesi oluşturulmuştur. Öznitelik çıkarma aşamasında, bitkinin morfolojik özellikleri dikkate alınmış ve Watershed algoritması ile

bitki görüntüsü eşit parçalara bölünmüştür. Ayrıca, bitki görüntülerinin renk, doku ve şekil öznitelikleri çıkarıldıktan sonra, çok katmanlı tanımlama yöntemi ile kombine edilerek sunucuda saklanmıştır. Bu çalışma ile %91,26 en iyi 1 başarı oranı, %97,40 en iyi 6 başarı oranı elde edilmiştir (Lee and Hong 2017).

Bitki yapraklarının tanımlanması üzerine yapılan bir diğer çalışmada ise, başarılı bir yaprak tanıma sistemi olan LeafSnap uygulamasından esinlenerek Android işletim sisteminde çalışan bitki tanımlama sistemi geliştirmiştir. Bu çalışmada Fransa bölgesinde yetişen 126 ağaç türüne ait görüntüleri içeren ImageCLEF2012 veri seti kullanılmıştır. Veri setindeki verilerin 2500 adedi eğitim için, 2370 adedi ise test için ayrılmıştır. Yapılan çalışmada, renkli yaprak görüntüsü girdi olarak kabul edilmiş, görüntüler ön işleme sürecinde siyah beyaza dönüştürülmüştür. Eşik tabanlı algoritma ile az gürültü içeren binary görüntü elde edilmiş ve top-hat dönüşümü ile kök, sap gibi detaylar görüntüden çıkarılmıştır. Görüntü kırılarak asıl objeye ulaşılmıştır. En iyi sınıflandırma sonucunun phog-tophat-cropped öznitelik birleşimiyle olduğu görülmüştür (Zhao, Ma et al. 2015).

Yanıkoglu vd. yaptıkları çalışmada yerel öznitelikleri kullanarak bitki görüntüsü tanımlama sistemi geliştirmiştir. Dense SIFT öznitelikleri ve kelime torbası yaklaşımı ile betimlenen görüntüler, destek vektör makinaları ile sınıflandırılmıştır. Bu çalışmada yenilik olarak, ilk etapta görüntüleri renklerine göre alt-kümelere ayırarak problemin karmaşıklığın azaltılması hedeflenmiştir. 2221 adet görüntü ve 161 çiçek türü içeren ImageClef yarışması veri seti kullanılmıştır. Yapılan denemelerde bitki sınıflandırması başarısı ön kümeleme ile %42.68, renk gruplarına ayrılmamış görüntüler üzerinde aynı betimleme ve sınıflandırma yöntemleri ile %18 olarak bulunmuştur. İncelenen görüntü Dense SIFT yöntemi ile parçalara ayrılmış, her bir parçadan elde edilen SIFT betimleyicileri K-Ortalama Kümeleme yöntemi ile belli sayıdaki görsel kelimeye indirgenmiştir. Daha sonra görüntü, kelime torbası yaklaşımı (Bag of Visual Words (BOVW)) denilen yöntemle, bu kelimelerin görülme histogramı ile belirlenmiştir. Son olarak da destek vektör makineleri yöntemiyle sınıflandırma işlemi yapılmıştır.

Görüntüler 7 renk grubuna ayrılmıştır(Kırmızı, Yeşil, Beyaz, Sarı, Mavi, Mor-Pembe, Çoklu-Renk). Ancak, elde edilen başarı oranı hedeflenenin çok altında (%52) bulunmuştur. Bunun sebepleri incelendiğinde, bazı renk gruplarının başarı ile belirlenmesine rağmen, karışık renkli ve yeşil renkli çiçeklere sahip türlerin gruplanmasının başarılı olmadığı gözlenmiştir. Yaşanan bu başarısızlık üzerine, bir ön çalışma olarak, çiçek rengi mevsimsel vb. Değişiklik göstermeyen ve yeşil çiçekli olmayan bitkiler kalacak şekilde veri seti indirgenmiştir. Elde edilen başarı sonuçları Çizelge 2,5'te sunulmuştur (Yildiran, Yanikoglu et al. 2014).

Çizelge 2.5. Renk Gruplarına Ayırmanın Sınıflandırma Üzerindeki Etkisi

Sınıf	Renk Sınıflandırma Başarısı	Bitki Sınıflandırma Başarısı	Genel Başarı Oranı
<i>Kırmızı</i>	%86.3	%68.1	%63.63
<i>Mavi</i>	%94.8	%48.7	%46.15
<i>Mor & Pembe</i>	%95.8	%29.8	%28.47
<i>Beyaz & Sarı</i>	%99.3	%47.4	%47.11
Renk Gruplaması ile Başarı Oranı	%97.4	-	%42.68
Renk Gruplaması olmadan Başarı Oranı	-	-	%18

3. MATERYAL ve YÖNTEM

Tezin bu bölümünde, tez kapsamında kullanılan materyal ve yöntemler açıklanmıştır.

3.1. Materyal

Tez kapsamında, web uygulaması Visual Studio 2010 Ultimate geliştirme ortamında C# ve Asp.Net dilleri ve Twitter Bootstrap 3 kütüphanesi kullanılmıştır. Web uygulamasının veri tabanı SQL Server 2012 R2 geliştirme ortamında SQL diliyle geliştirilmiştir. Web hizmeti katmanında Sublime Text 3 metin editörü kullanılmış ve proje PHP dili ile geliştirilmiştir. DÖ kütüphanesi olarak Google tarafından geliştirilen açık kaynak kodlu TensorFlow kütüphanesi ve API olarak Keras kütüphanesi kullanılmıştır. DÖ modeli olarak ImageNet veri setiyle DAEYSA ile eğitilmiş MobileNet modeli kullanılmıştır. DÖ sürecinde PyCharm geliştirme ortamı ve Python dili kullanılmıştır. Mobil uygulama katmanında Android işletim sisteminde çalışan Java diliyle hazırlanmış bir mobil uygulama bulunmaktadır. Çizelge 3.1'de kullanılan materyaller gösterilmiştir.

Çizelge 3.1. Kullanılan Materyaller

Uygulama Katmanı	Dil	Geliştirme Ortamı	Kütüphane
Web Arayüzü	HTML	Visual Studio 2010 Ultimate	Twitter Bootstrap 3.x
Web Uygulaması	C#, Asp.Net	Visual Studio 2010 Ultimate	.Net
Veri Tabanı	SQL	SQL Server 2012 R2	.Net

Web Hizmeti	PHP	Sublime Text 3	-
Ağın Eğitimi	Python	PyCharm	Tensorflow Keras
Mobil Uygulama	Java	Android Studio	-

3.1.1. Veri seti

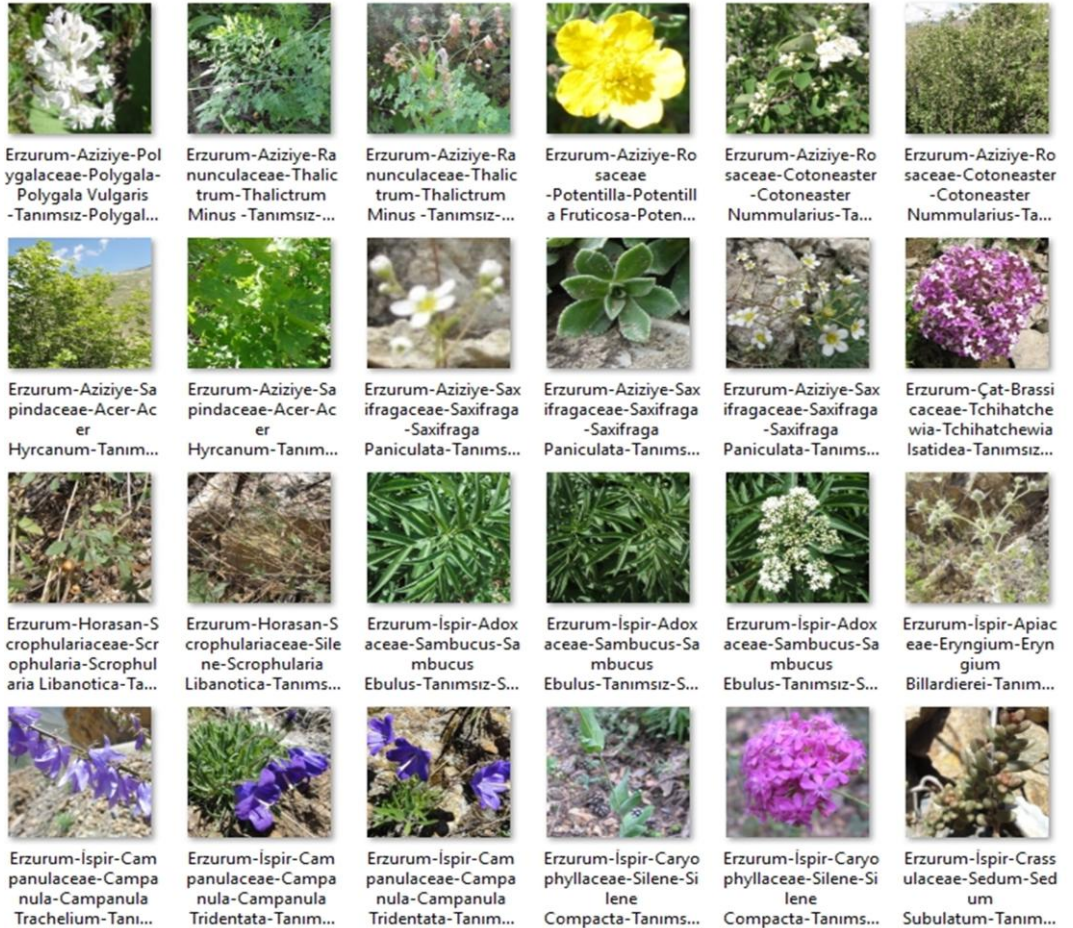
Çiçek görüntülerinin sınıflandırılması için yapılan çalışmalarda, Oxford-17 ve Oxford-102 gibi hazır veri setleri kullanılmaktadır. Tez çalışması kapsamında, Erzurum Orman Bölge Müdürlüğü yetki ve sorumluluk sınırları içerisinde kalan Erzurum, Erzincan, Ardahan, Kars, Iğdır ve Ağrı illerinden çiçek görüntüleri toplanmıştır. Bu çalışma Erzurum Orman Bölge Müdürlüğü'nde çalışan sözleşmeli Orman Mühendisleri ve tez çalışmasının bu bölümünde görev alan Atatürk Üniversitesi Bilgisayar Mühendisliği Bölümü'ndeki lisans öğrencileri tarafından gerçekleştirilmiştir.

Bu verilere ek olarak, Türkoğlu et al. tarafından yapılan "Ağrı Dağı'nın Florası(Türkoğlu and Zeynalov 2016)" isimli çalışmada bulunan 512 görüntü de kendilerinden rica edilmiş ve akademik çalışmalarda kullanılmak kaydıyla dijital ortamda teslim alınmıştır. Şekil 3.1'de Ağrı Dağı Florası verileri gösterilmiştir.

Bitki No.	FAMILYA	Familya- Türkçe adı	BITKİNİN ADI	TÜRKÇE ADI	ÖMRÜ	HAYAT FORMU	ÇİÇEKLENME AYLARI	YÜKSEKLİK	ENDEMİZM	FITOCOGRAFYA	B 10	AĞRI DAĞI
1. a, b	Grimmiaceae	Karayosunları	Grimmia orbicularis Bruch ex Wilson	Karayosunu	Çok yıllık			2000-2800		Avrupa-Sibirya		
2	Amaranthaceae	Tilkikuyruğuğiller	Amaranthus retroflexus L.	Kırmızı köklü horoz ibiği	Tek yıllık	otsu	5-8	700-1900				
3	Apiaceae	Maydanozğiller	Anthriscus sylvestris (L.) Hoffm.	Gımgım, gımi	çok yıllık	otsu	5-7	800-2200				
4	Apiaceae	Maydanozğiller	Anthriscus nemorosus (Bieb.) Spengel	Gımgım, Gımi	çok yıllık	otsu	4-7	800-1700		Avrupa-Sibirya		
5.a, b	Apiaceae	Maydanozğiller	Carum carvi L.	Kır kimyonu, Tarak otu	çok yıllık	otsu	5-6	1200				
6	Apiaceae	Maydanozğiller	Conium maculatum L.	Baldıran	tek yıllık	otsu	5-8	800-2000				
7.a, b	Apiaceae	Maydanozğiller	Eryngium billardieri Delar.	Boğa dikeni, Çakır dikeni	çok yıllık	otsu	7-8	1400-2500		İran-Turan		
8	Apiaceae	Maydanozğiller	Heracleum sphandylum	Tavşancıl otu, Baldırgan	çok yıllık	otsu	5-6	1000-1900		Avrupa-Sibirya		
9.a, b	Apiaceae	Maydanozğiller	Ferula communis L.	Deli çadır çok yıllık	çok yıllık	otsu	6-7	1200		İran-turan		
10.a, b	Apiaceae	Maydanozğiller	Ferula orientalis L. subsp. Orientalis	Eşek çarşın	çok yıllık	otsu	7-8	1100-2500				
11	Apiaceae	Maydanozğiller	Malabaila secacul Banks & Sol	Kelemekşir, manda	çok yıllık	otsu	5-6	1000-2000		İran-turan		
12.a, b	Apiaceae	Maydanozğiller	Torilis arvensis L.	Pıtrak	tek yıllık	otsu	6-8	600-12000		Akdeniz		
13.a, b	Apiaceae	Maydanozğiller	Torilis leptophylla (L.) Reichb. Pıtrak,	Pıtrak, çocukboğan	tek yıllık	otsu	4-8	900-2300				
14	Apiaceae	Maydanozğiller	Tordylium hassliquistiae DC.	Geyik otu	tek yıllık	otsu	4-5	800-1800				
15.a, b	Apiaceae	Maydanozğiller	Zosima absinthifolia (Vent.) Link.	Peynir otu, Aypeli	çok yıllık	otsu	6-7.	1400			B9 var	
16.a, b	Asteraceae	Topluççekğiller	Aster amellus L. subsp. ibericus (STEV.) Avestian	Kafkas yıldızı	Çok yıllık	otsu	6-8	1500-2300		Karadeniz	var	

Şekil 3.1. Ağrı Dağı Florası Verileri

2 yılı aşkın süreçte sürekli olarak güncellenen veri seti, 550 türe ait 2031 görüntüden oluşmaktadır. 20' den az görüntü içeren türler, öğrenme aşaması için yetersiz olacağından daha sonraki çalışmalarda kullanılmak üzere tez kapsamı dışında tutulmuştur. Çiçek görüntüleri genellikle bal ormanlarının konumlandığı coğrafyalardan toplanmış ve konum - zaman üstverileriyle (metadata) birlikte kayıt altına alınmış olup, diğer akademik çalışmalarda da (örneğin, Doğu Anadolu Bal Haritası'nın çıkartılması) kullanılabilmesi amaçlanmıştır. Bu çalışmayla birlikte eğitim ve test sürecinde kullanılan veri seti 14 türe ait 1215 görüntüden oluşmaktadır. Veri setinin genişletilmesine yönelik çalışmalar devam etmektedir. Oluşturulan veri setinde tür başına düşen fotoğraf sayısı 63 ile 96 arasında değişmektedir. Şekil 3.2'de veri setinde bulunan bazı türlere ait görüntüler gösterilmiştir.



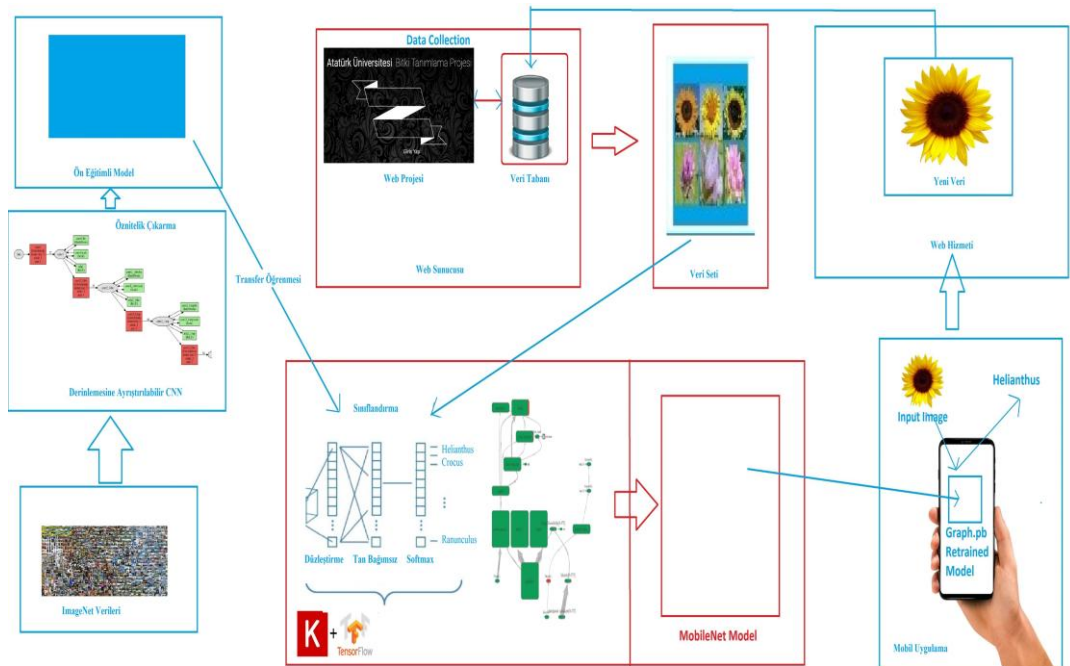
Şekil 3.2. Veri Seti

3.2. Yöntem

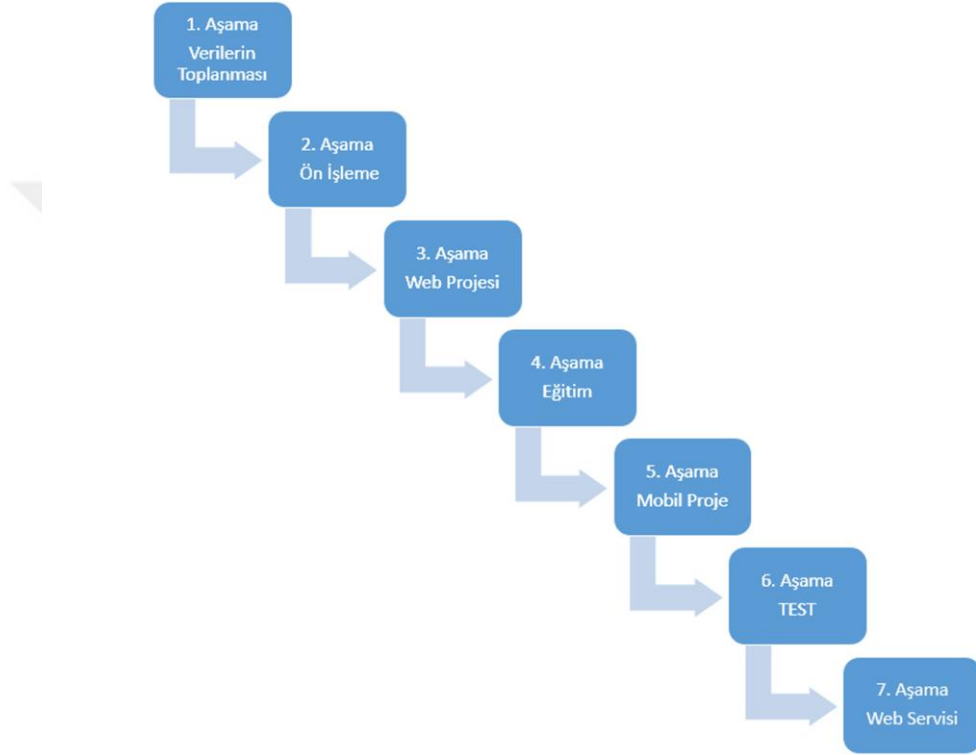
Tez çalışması kapsamında gerçekleştirilen proje 6 aşamadan oluşmaktadır. İlk aşamada Bölüm 3.1.1’ de değinilen veri seti hazırlanmıştır. İkinci aşamada veriler ön işleme sürecine tabi tutulmuştur. Üçüncü aşamada ise verilerin muhafaza edilebilmesi için bir web uygulaması hazırlanmıştır. Dördüncü aşamada ise sunucu ve mobil uygulama arasında veri alışverişini sağlayacak bir web hizmeti yazılmıştır. 5. aşamada veriler derinlemesine ayrıştırılabilir CNN ağıyla eğitilmiş ve MobileNet modeli elde edilmiştir. 6. ve son aşamada ise model, hazırlanan mobil uygulamanın içine gömülmüştür.

3.2.1. Sistem altyapısı ve özet mimarisi

Şekil 3.3’te sistemin özet mimarisi gösterilmiştir. Devam eden bölümlerde sistem özet mimarisi, Şekil 3.4’te akış diyagramı gösterilmiştir.



Şekil 3.3. Sistem Özet Mimarisi



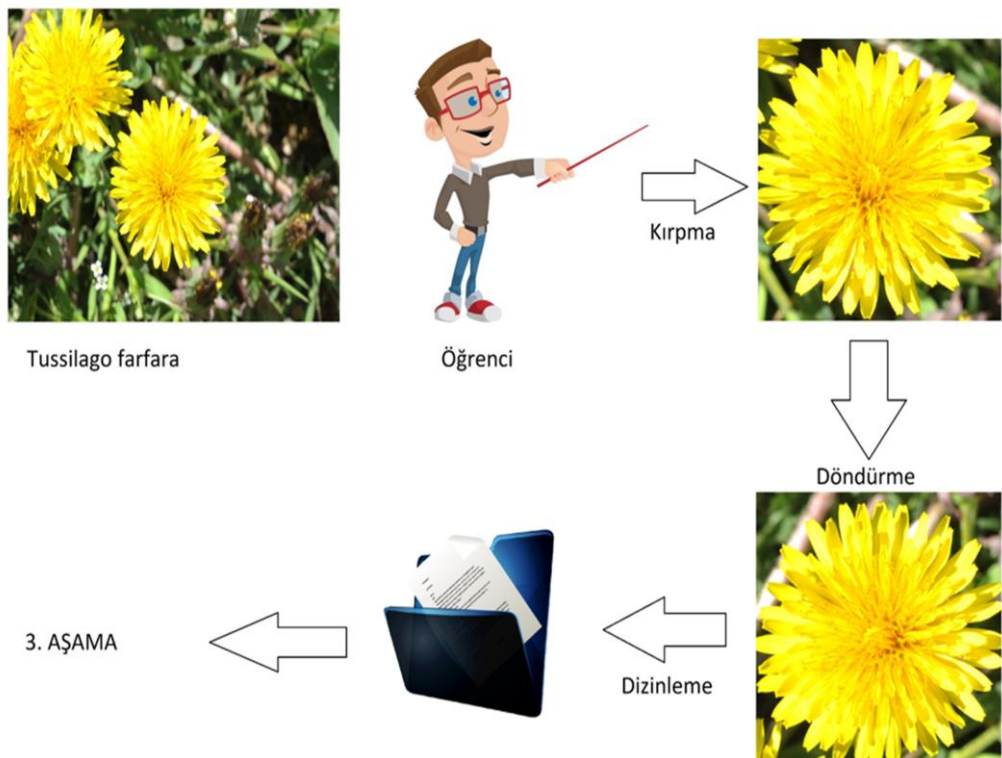
Şekil 3.4. Akış Diyagramı

3.2.2. Verilerin ön işlenmesi

Görüntü sınıflandırma algoritmalarının başarısını etkileyen en önemli etkenlerden biri de veri ön işlemedir. Görüntü verisi ne kadar sade ve kaliteli olursa, eğitim süreci ve buna bağlı olarak sınıflandırma doğruluğu artmaktadır. Görüntü verilerinde sınıflandırılmak istenen asıl nesnenin yanında alakasız ve net olmayan veri varsa, sınıflandırma süreci de o derece zor olur. Bu tür alakasız verilere genel olarak gürültü denmektedir.

Gürültülü veri, veri kalitesini etkileyen yaygın sorunlardan biridir. Özellikle eğitim verilerinin yanlış etiketlenmesi sınıflandırma problemlerinde yıkıcı sonuçlara sebep olmaktadır (García-Gil, Luengo et al. 2017). Gürültülü verideki gürültünün temizlenmemesi veri analizini engellediğinden, veri temizliğinin önemli bir hedefidir (Xiong, Pandey et al. 2006).

Sahadan toplanan verilerde asıl bitkiyle birlikte çekilmiş otsu bitkiler ve geniş açı fotoğraflama yönteminin sebep olduğu ağaç, bina, gökyüzü gibi alakasız nesnelere bulunmaktadır. Bu tür gürültülerin sınıflandırma sürecine olumsuz etki sağlayacağı aşikâr olduğundan, görüntüler el ile kırılarak gürültüden arındırılmış ve etiketlenmiştir. Ayrıca, ön işleme aşamasında veri boyutunun öğrenme üzerine etkisini de araştırabilmek adına mevcut görüntüler 30 °, 60 °, 90 °, 120 °, 150 °, 180 °, 210 °, 240 °, 270 °, 300 ° ve 330 ° döndürülerek veri seti genişletilmiştir. Bu işlemle, 14 türe ait 1215 gürültüsüz görüntüden oluşan veri seti 14580 görüntüye ulaşmıştır. Şekil 3.5'te ön işleme süreci gösterilmiştir.

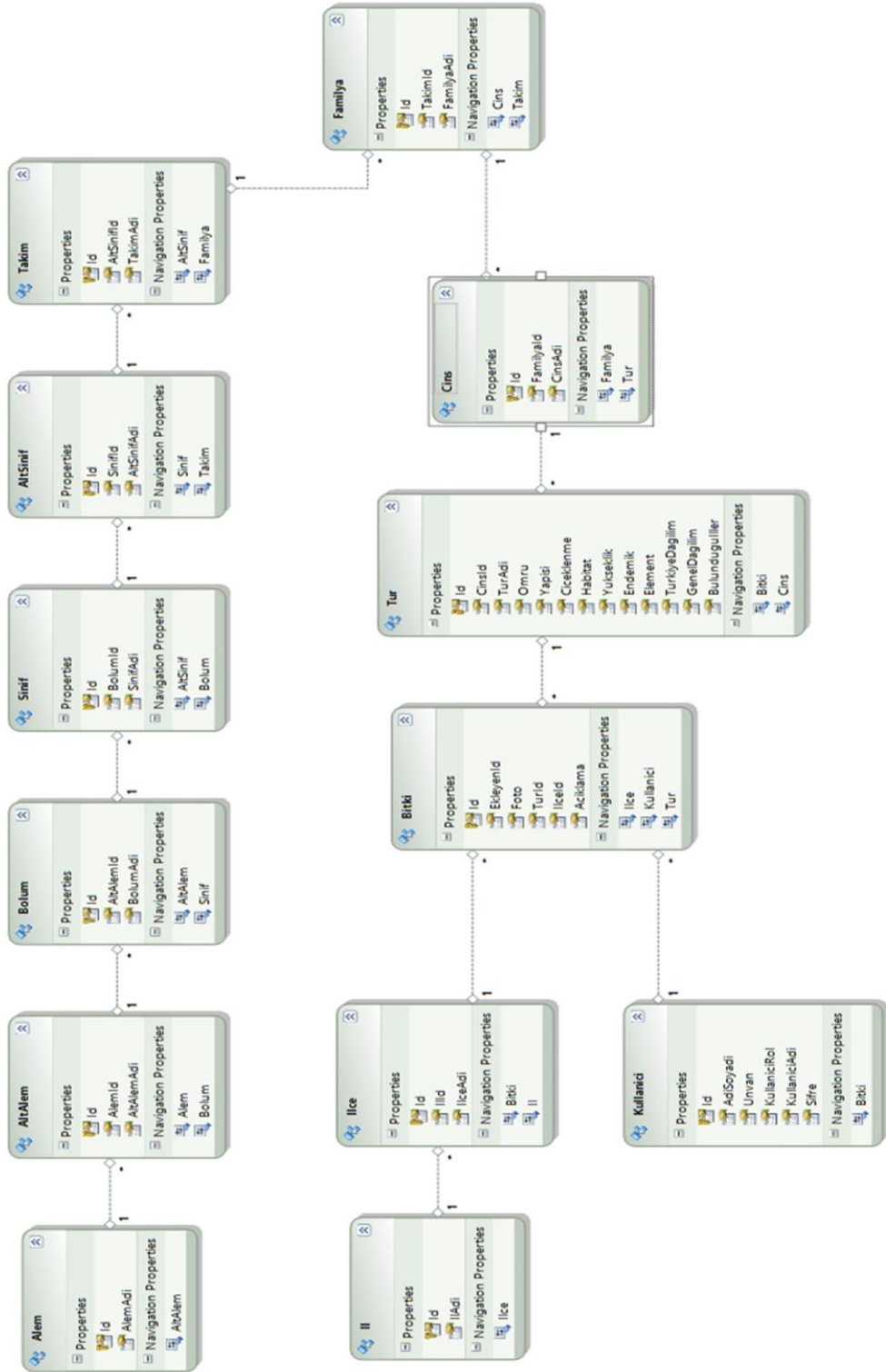


Şekil 3.5. Ön İşleme Süreci

3.2.3. Web uygulaması

Görüntü toplama aşamasında çok fazla kişinin (öğrenciler, orman mühendisleri) çalışması sebebiyle olası veri kaybını engellemek ve verileri alanında uzman kişilere etiket ettirmek amacıyla bir web uygulaması geliştirilmiştir. Bu uygulama web ve mobil ortamda çalıştığı için sahada çekilen fotoğrafların anlık olarak etiketlenerek veri tabanına aktarılmasına imkân tanımaktadır.

Bitkilerin sınıflandırılması için 2009'da ortaya atılan ve halen kabul gören Evrimsel Kapalı Tohumlu Gelişimi Topluluğu (APG) III sistemi kullanılmıştır (Phylogeny and Bremer 2009). Bu sistem temel alınarak bitkiler âlem, alt âlem, bölüm, sınıf, alt sınıf, takım, familya, cins ve tür olmak üzere 9 katmanlı mimariyle sınıflandırılmıştır. Projenin veri tabanı katmanı da bu prensiplere bağlı kalınarak oluşturulmuştur. Son katmanda tür tablosuna bağlanan bitki tablosu il ve ilçe tablolarıyla konum verilerini tutmaktadır. Ayrıca, bitki tablosuna bağlı kullanıcı tablosuyla bitki fotoğrafını Şekil 3.6'da veri tabanı diyagramı gösterilmiştir.



Şekil 3.6. Veri Tabanı Diyagramı

Tür tablosunda Latince tür adı, bitkinin ömrü (çok yıllık, tek yıllık), yapısı(otsu, odunsu vb), çiçeklenme süresi, habitatu (kayalık, çayırılık vb.), görüldüğü rakım, endemik bitki olup olmadığı, kökeni (Avrupa-Sibirya, İran-Turan vb.), Türkiye’deki dağılımı, dünyadaki dağılımı ve bulunduğu iller gibi veriler tutulmaktadır. Bitki tablosunun bu denli detaylı planlanmasının sebebi, oluşturulan verinin daha sonraki akademik çalışmalarda da kullanılabilmesine olanak sağlanmasının istenmesidir.

Proje ara yüzünün ekran boyutundan bağımsız olması önemli bir kıstastır. Genellikle Web geliştiriciler uygulamanın çalışacağı cihazın boyutuna, hedef kitlenin yoğun olarak kullandığı cihaz boyutuna ya da dünya genelinde kabul görmüş cihaz boyutuna göre karar vermektedirler (Budak and Gezer 2016).

Web için ayrı, mobil için ayrı proje geliştirmek proje maliyetini artırmaktadır. Bu sebeple, ekran boyutundan bağımsız uygulamalar günümüzde oldukça popülerdir. Web projesinin ara yüzünün standartlara uygun olması ve sahada çalışan kişilerin kullandıkları tüm cihazlarda çalışabilmesi için önyüz katmanında Twitter firmasına ait Bootstrap 3.x kütüphanesi kullanılmıştır. Şekil 3.7’de Twitter Bootstrap ile hazırlanmış giriş ekranınının 17.3 inç dizüstü bilgisayar ekranında ve 5.6 inç mobil ekranda görünümü gösterilmiştir.



Şekil 3.8. Web Uygulaması Girişinin Farklı Boyutlardaki Ekranlarda Görünümü

Ayrıca sıralama, arama, gruplama gibi sunucuya iş yükü getirecek işlemlerin önyüzde yapılması sağlanmıştır. Örneğin tablolama işlemi için 3. Parti ücretsiz Javascript kütüphanelerinden Datatable.js kullanılmıştır. Bir diğer işlev olan bitkilerin harita üzerine konumlandırılması için yine sunucu katmanında değil, önyüz katmanında 3. Parti ücretsiz Javascript kütüphanelerinden JVectorMap.js kullanılmıştır. Şekil 3.8’ de datatable ile bitki türlerinin tablolaması gösterilmiştir.

Tür Listesi										
										Yeni Tür Ekle
Search:										
Show 10 v entries										
Bağlı Oluştuğu Alem	Bağlı Oluştuğu Bölüm	Bağlı Oluştuğu Sınıf	Bağlı Oluştuğu Alt Sınıf	Bağlı Oluştuğu Takım	Bağlı Oluştuğu Familya	Bağlı Oluştuğu Cins	Tür Adı	Bu Türdeki Bilgi Sayısı	İşlem	
Plantae	Tracheobionta	Magnoliopsida	Rosidae	Sapinales	Asteraceae	Acer	Acer hyrcanum	2		
Plantae	Tracheobionta	Magnoliopsida	Asteridae	Asterales	Asteraceae	Achillea	Achillea millefolium	3		
Plantae	Tracheobionta	Magnoliopsida	Rosidae	Rosales	Rosaceae	Agmonia	Agmonia eupatorioides	2		
Plantae	Tracheobionta	Magnoliopsida	Caryophyllales	Caryophyllales	Caryophyllaceae	Agrostemma	Agrostemma githago	2		
Plantae	Tracheobionta	Magnoliopsida	Dilleniidae	Malvales	Malvaceae	Alecea	Alecea striata	2		
Plantae	Tracheobionta	Magnoliopsida	Rosidae	Rosales	Rosaceae	Alchemilla	Alchemilla caucasica	3		
Plantae	Tracheobionta	Magnoliopsida	Asteridae	Lamiales	Borraginaceae	Alliaria	Alliaria orientalis	2		
Plantae	Tracheobionta	Magnoliopsida	Liliidae	Liliales	Liliaceae	Allium	Allium akaka	3		
Plantae	Tracheobionta	Magnoliopsida	Dilleniidae	Primiales	Primulaceae	Anagallis	Anagallis nemina	2		
Plantae	Tracheobionta	Magnoliopsida	Dilleniidae	Primiales	Primulaceae	Androsace	Androsace armenica	1		

Şekil 3.9. DataTable.js Kütüphanesi İle Tablolama

Bitki fotoğraflarının hiyerarşik yapısının çıkartılması oluşturulacak olan veri seti için anlamlı olacağından, ön yüzde ağaç yapısı kullanılarak genel hiyerarşi yansıtılmıştır. Şekil 3.9'da bu yapı gösterilmiştir.



Şekil 3.10. Ağaç Yapısı ile Bitki Türünün Hiyerarşik Yapısı

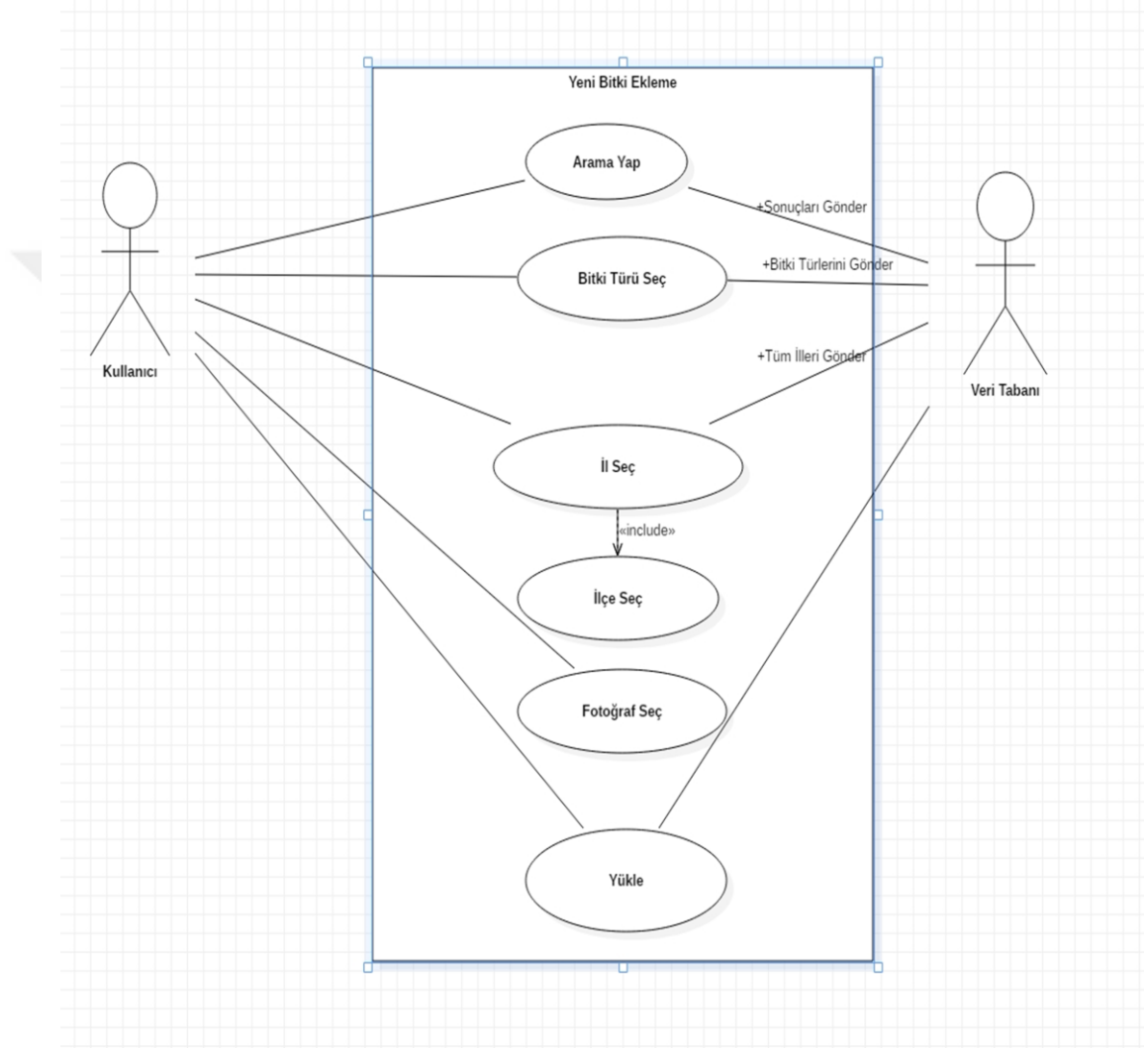
Projenin arka uç (backend) kodlaması C# dili ile ASP.Net MVC4 tasarım desenine bağlı kalınarak geliştirilmiştir. ASP.Net MVC4 tasarım deseninin seçilmesindeki ana sebep, projenin ileriye dönük olmasının istenmesidir. Birden fazla yazılımcının çalıştığı projelerde uygulanması elzem olan MVC tasarım deseniyle proje veri tabanı (model), ara yüz (view) ve kod (controller) katmanlarına ayrılmıştır.

Arka uç katmanında, depo (repository) deseni kullanılmıştır. Depo deseni ile veri erişim katmanı (data access layer) standart hale getirilmiştir. Hazırlanan kod taslağı ile aynı kodların tekrar tekrar yazılması engellenmiş ve veri tabanındaki CRUD (Create, Read, Update, Delete) fonksiyonları tek merkezden kullanılmıştır. Ayrıca, bu yapılanma ile veri enjeksiyonu(data injection) ve test aşamaları daha hızlı bitirilmiştir. Ek-1’de baz metod ve Ek-2’de depo kodları sunulmuştur.

Arka uç katmanında bitki görüntülerinin yüklenmesi aşamasında, bitkinin sadece türü seçilmekte ve türün silsile ile bağlı olduğu cins, bölüm vb. bilgiler otomatik olarak tanımlanmaktadır. Kullanıcı yeni bitki eklerken, bitki türünü seçtikten sonra bitkinin nerede fotoğraflandığını seçip, bitki görüntüsünü yükleyerek bitkiyi kaydedebilmektedir.

Bu işleyle, kullanıcı dostu bir tasarım politikası izlenmiş ve verinin çatallaşması engellenmiştir. Bir başka deyişle aynı bitki türüne ait çiçeklenme süresi, habitat vb. verilerin aynı olması sağlanmıştır.

Bitki görüntüsünün boyutu hususunda herhangi bir kısıtlama yapılmamış ve görüntü post metoduyla sisteme yüklenmektedir. Şekil 3.10’da bitki ekleme sürecinin kullanım senaryosu diyagramı (Use case diagram) gösterilmiştir.



Şekil 3.11. Yeni Bitki Ekleme Kullanım Senaryosu

Bitki görselleri galeri tasarımıyla kullanıcıya gösterilmektedir. Galeri ile bitki görüntüsü listeleme işlemi tür, cins, bölüm vb. alt katmanlarda yapıldığı gibi genel olarak tüm veriler için de listeleme yapılmaktadır. Bu aşamada çok fazla sayıda görüntünün sunucudan çağırılması ihtimaliyle sayfanın yüklenme süresinin artması olağandır. Sayfa yüklenme süresini düşürmek için görüntüler orijinal boyutuyla değil Twitter Bootstrap

kütüphanesinin 12' lik ızgara sisteminde 3 ızgaraya tekabül edecek şekilde 300 x 300 px boyutuyla gösterilmektedir. Şekil 3.11'de Ranunculus Acraeus türüne ait bitki görüntüleri galerisi gösterilmiştir.



Şekil 3.12. Bitki Görüntüleri Galerisi

Web ve veri tabanı verileri için tek sunucu kullanılmıştır. HP Proliant DL-380 G7 model sunucu üzerine Windows Server 2012 R2 yazılımı kurulmuştur. Diğer tüm gerekli bileşenler kurulduktan sonra turbo hızlandırma (turbo boost) sistemi devreye alınmıştır. Turbo hızlandırma ile Intel Xeon 5560 serisi işlemcilerin hızı 2.8 GHz'den 3.2 GHz'e çıkarılmıştır. Sunucunun dışarıya açılabilmesi için Vodafone firmasından sabit IP hizmeti satın alınmıştır. Projenin ilerleyen aşamalarında, sunucunun bakım ve soğutma maliyetlerinin işgücü kaybına sebep olduğu görülmüş olup, sunucu ve barındırma hizmeti satın alınarak tüm veriler İstanbul merkezli SSD destekli sunuculara gönderilmiştir. Web projesi XXX.238.171.194 nolu IP' de, veri tabanı ise XXX.238.171.194 nolu IP' de barınmaktadır.

3.2.4. Web hizmeti

Projenin mobil uygulama katmanından gelen görüntülerin, daha sonraki dönemlerde eğitim verisi olarak kullanılabilmesi için mobil uygulama ve sunucu arasında veri taşıma görevi yapacak bir web hizmeti geliştirilmiştir. Web hizmeti, PHP dili ile Sublime Text 3 editörü kullanılarak geliştirilmiştir.

Mobil uygulamadan gönderilen görüntü sınıflandırılabilirse sunucudaki bitki görselleri dizinindeki ilgili sınıfa, sınıflandırılmadıysa sınıflandırılmayan görseller dizinine taşınmaktadır. Sınıflandırılmayan görseller daha sonra botanik uzmanı desteğiyle etiketlenmektedir. Ayrıca, bu web hizmeti sunucuda biriken görüntü sayısı 1000' e erişince yöneticiyi uyarmakta ve görüntüler kullanılarak tekrar eğitim yapılması beklenmektedir. Şekil 3.12'de Web hizmetinin sınıflandırılmayan görüntüyü sunucuya göndermesi için hazırlanan kod bloğu gösterilmiştir.

```

if(isset($_POST["encoded_string"])){

    $encoded_string = $_POST["encoded_string"];
    $image_name = $_POST["image_name"];

    $decoded_string = base64_decode($encoded_string);

    $path = 'images/siniflandirilmayan'.$image_name;

    $file = fopen($path, 'wb');

    $is_written = fwrite($file, $decoded_string);
    fclose($file);

    if($is_written > 0) {

        $connection = mysqli_connect('localhost', 'root', '', 'ylplantapp');
        $query = "INSERT INTO rawphotos(name,dizin) values('$image_name','$path')";

        $result = mysqli_query($connection, $query) ;

        if($result){
            echo "success";
        }else{
            echo "failed";
        }

        mysqli_close($connection);
    }
}

```

Şekil 3.13. Web Servisi ile Sınıflandırılmayan Görüntülerin Sunucuya Gönderilmesi

3.2.5. Görüntü verilerinin eğitilmesi

3.2.5.1. Keras ve Tensorflow

Tensorflow, Google Brain ekibi tarafından geliştirilen, sayısal hesaplama ve büyük ölçekli DÖ problemleri için çözümler sunan bir kütüphanedir. TensorFlow, çeşitli MÖ ve DÖ mimarilerini ve algoritmalarını bir arada barındırılmaktadır.

Keras, TensorFlow gibi kütüphanelerin üst seviye API' si olarak kullanılmaktadır. TensorFlow' un karmaşık yapısını kolaylaştırmak için resmi olarak TensorFlow tarafından desteklenmekte ve 1.12 sürümünden itibaren TensorFlow ile birlikte, "Makineler için değil, insanlar için makine öğrenmesi" sloganıyla sunulmaktadır. Keras, yapay sinir ağları, maliyet fonksiyonları, optimize ediciler, aktivasyon fonksiyonları gibi bileşenleri bünyesinde barındırarak yeni modeller oluşturmaya olanak sağlamaktadır.

3.2.5.2. MobieNet mimarisi

MobileNet, Mobil ve gömülü görüntü işleme uygulamaları için derinlemesine ayrıştırlabilir evrişimler kullanan aerodinamik bir mimariye dayanmaktadır (Howard, Zhu et al. 2017).

Projenin bu aşaması Google tarafından geliştirilen açık kaynak kodlu TensorFlow kütüphanesi ve kullanıcı dostu ara yüz sunan Keras Api'si kullanılarak Python diliyle bitki görüntülerini sınıflandırabilecek bir derinlemesine ayrıştırılabilir ağ olan MobileNet ağı eğitilmiştir. MobileNet emsallerine kıyasla daha küçük boyuta sahiptir. Tez çalışmasında kullanılan sürümü sadece 17,7 megabayttır.

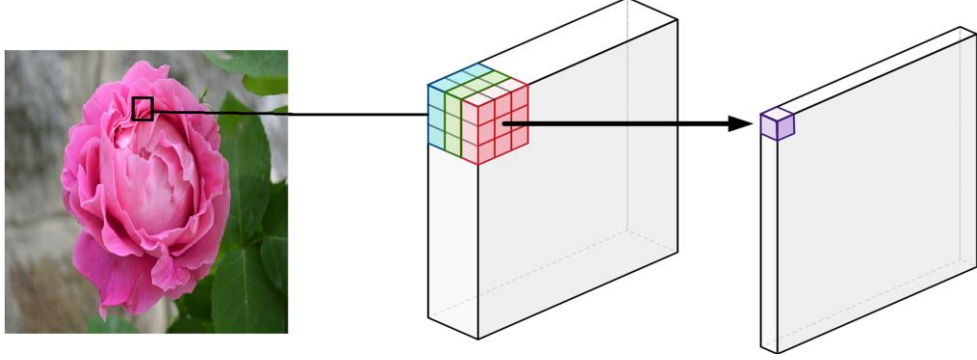
MobileNet için derinlemesine evrişim her giriş kanalına tek bir filtre uygulamaktadır. Derinlemesine evrişimi işleminden çıkan çıktıları birleştirmek için 1 x 1 evrişim işlemi Nokta bazlı evrişim tarafından uygulanmaktadır. Standart evrişim işleminde, tek bir adımda hem filtreleme hem de birleştirme işlemi yapılmaktadır. Derinlemesine ayrıştırılabilir evrişim bu işlemi iki katmana bölmektedir. Bu da hesaplama maliyetini ve model boyutunu önemli ölçüde azaltma etkisine sahiptir (Howard, Zhu et al. 2017).

MobileNet yukarıda anlatılan standart evrişim, derinlemesine evrişim ve nokta bazlı evrişim katmanlarının bir araya gelmesiyle oluşmaktadır. Tezin bu aşamasında öznetelik çıkarma ve sınıflandırma aşaması için 30 katmanlı bir MobileNet ağı eğitilmiştir. Çizelge 3.2' de MobileNet mimarisi gösterilmiştir.

Çizelge 3.2. 30 Katmanlı MobileNet Mimarisi

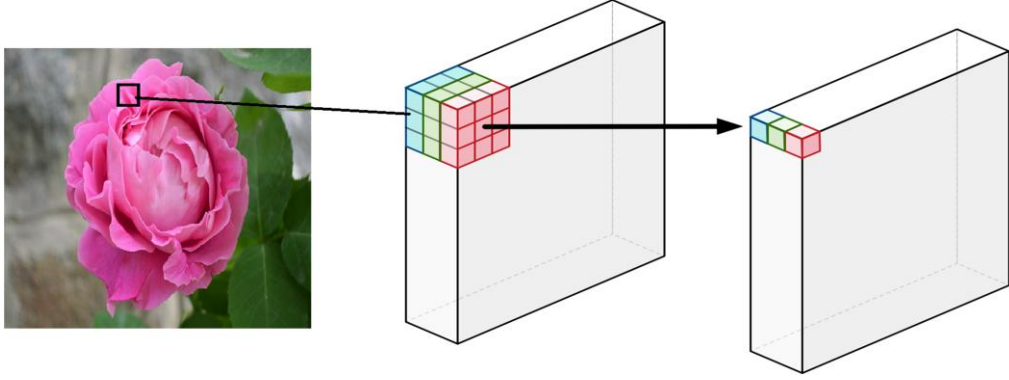
Katman	Kayma	Filtre Boyutu	Girdi Boyutu
Standart Evrişim	2	3×3×3×32	224×224×3
Derinlemesine Evrişim	1	3×3×32	112×112×32
Nokta Bazlı Evrişim	1	1×1×32×64	112×112×32
Derinlemesine Evrişim	2	3×3×64	112×112×64
Nokta Bazlı Evrişim	1	1×1×64×128	56×56×64
Derinlemesine Evrişim	1	3×3×128	56×56×128
Nokta Bazlı Evrişim	1	1×1×128×128	56×56×128
Derinlemesine Evrişim	2	3×3×128	56×56×128
Nokta Bazlı Evrişim	1	1×1×128×256	28×28×128
Derinlemesine Evrişim	1	3×3×256	28×28×256
Nokta Bazlı Evrişim	1	1×1×256×256	28×28×256
Derinlemesine Evrişim	2	3×3×256	28×28×256
Nokta Bazlı Evrişim	1	1×1×256×512	14×14×256
5 x Derinlemesine Evrişim	1	3×3×512	14×14×512
5 x Nokta Bazlı Evrişim		1×1×512×512	14×14×512
Derinlemesine	2	3×3×512	14×14×512
Havuzlama	1	1×1×512×1024	7×7×512
Derinlemesine	2	3×3×1024	7×7×1024
Havuzlama	1	1×1×1024×1024	7×7×1024
Ortalama Havuzlama	1	Havuzlama 7×7	7×7×1024
Tamamen Bağlı	1	1024×1000	1×1×1024
Softmax	1	Sınıflandırıcı	1×1×1000

Bölüm 2.3.2’de anlatıldığı üzere, derinlemesine ayrıştırılabilir evrişimli ağlarda her evrişim işlemi 3 rengin kombine edilerek uygulanmasından ziyade sadece tek bir renge uygulanır. Bu işlem sayesinde girdi kanallarının filtrelenme süreci etkilenmektedir. Şekil 3.13’te Isparta gülü bitkisi üzerinde standart evrişim işlemi gösterilmiştir.



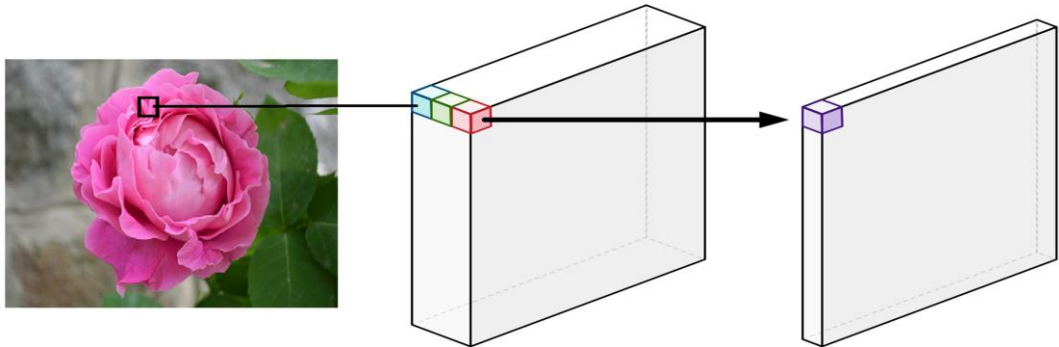
Şekil 3.14. Isparta Gülü Bitkisi Üzerinde Standart Evrişim İşlemi

Şekil 3.14'te Isparta gülü bitkisi üzerinde derinlemesine evrişim işlemi gösterilmiştir.



Şekil 3.15. Isparta Gülü Bitkisi Üzerinde Derinlemesine Evrişim İşlemi

Şekil 3.15'te Isparta gülü bitkisi üzerinde nokta bazlı evrişim işlemi gösterilmiştir.



Şekil 3.16. Isparta Gülü Bitkisi Üzerinde Nokta Bazlı Evrişim İşlemi

3.2.5.3. Eğitim Süreci

Tezin bu aşamasında, yukarıda anlatılan 30 katmanlı ağın en üst katmanı olan Softmax katmanı ayrılmış ve bu katmandan önceki katmanlar ImageNet verileriyle eğitilmiştir. Transfer öğrenmesi felsefesinden faydalanılarak, daha önce benzer problemler için eğitilmiş ön eğitilmiş bir MobileNet ağı kullanılmıştır. Son katman olan sınıflandırıcı Softmax katmanı ise kendi veri setimiz ile eğitilmiş ve farklı MobileNet hiper parametrelerinin sınıflandırma üzerindeki etkisi gözlemlenmiştir.

MobileNet 2 hiper parametre ile özelleştirilebilmektedir.

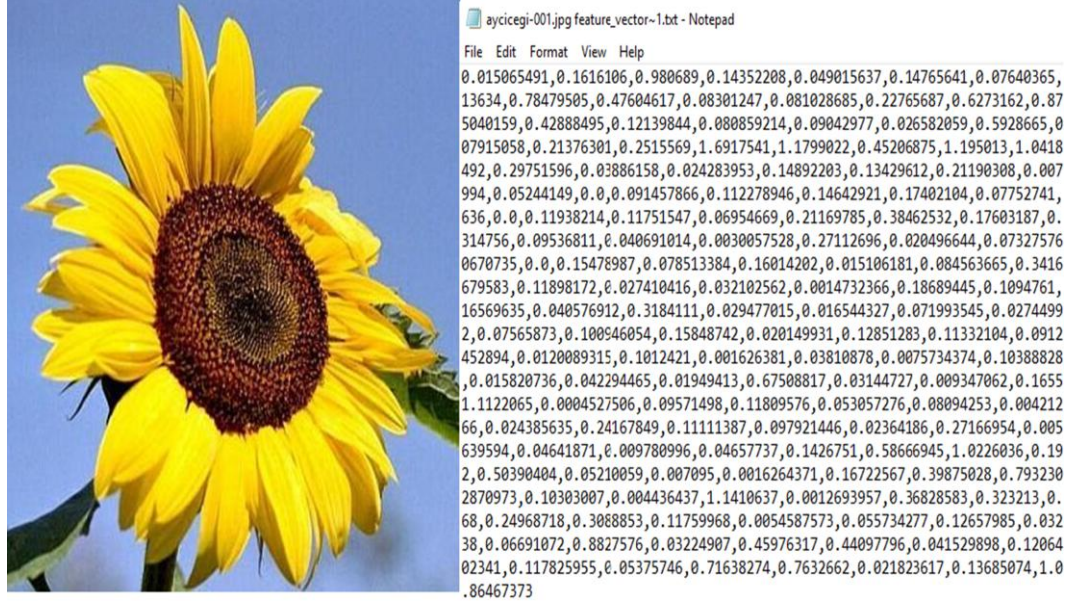
- 1- Girdi görüntüsünün çözünürlüğü: 128, 160, 192, 224 px
- 2- Gizli katmanlardaki nöron sayısı: 0,25, 0,50, 0,75, 1,0

İlk aşamada, 14 türe ait 1215 bitki görüntüsü etiketlenmiş alt klasörlere ayrılmıştır. Eğitim sürecinde tüm hiper parametre kombinasyonları denenmiş ve bu klasörler dizini Ek-3' te kodu sunulan yeniden eğitici Python koduna gönderilmiştir.

Ağın sadece son katmanı eğitildiğinden, eğitim süreci GTX 960M DDR5 GIÜ ile 14 dakika sürmüştür. Eğitim sürecinin sonunda asıl sınıflandırmayı yapan ve DÖ literatüründe Bootleneck olarak adlandırılan her görüntü için öznitelik vektörleri hesaplanmıştır.

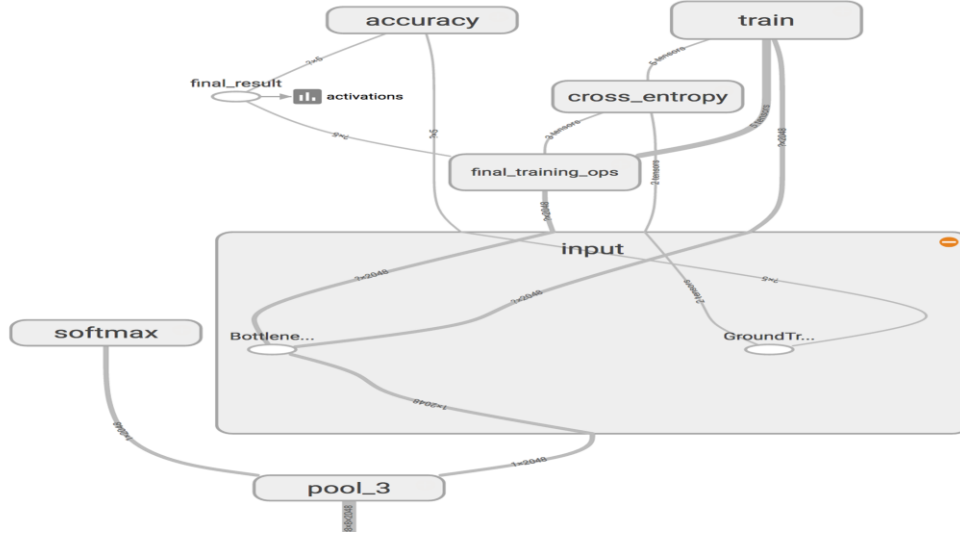
Eğitim boyunca her görüntü birçok kez kullanılmaktadır. Öznitelik vektörü arkasındaki katmanların hesaplanması için ciddi kaynağa ve zamana ihtiyaç duyulmaktadır. Fakat transfer öğrenme yönteminde bu katmanların çıktıları değişmediğinden, önbelleklenip

tekrar tekrar kullanılmıştır. Şekil 3.16’da 001 nolu ayçiçeği görüntüsü için hazırlanan öznitelik vektörü gösterilmiştir.



Şekil 3.17. Ayçiçeği 001 Görüntüsü ve Oluşan Öznitelik Vektörü

Eğitim aşaması TensorFlow kütüphanesinin izleme ekranı olan TensorBoard ile izlenmiştir. Yukarıda bahsedildiği gibi önceden ImageNet verileriyle eğitilmiş olan katmanlar görüntü sınıflandırmaya yarayan öznitelikleri barındırmaktadır. Şekil 3.17’de eğitim süreci ve önceden eğitilmiş katmanların Softmax katmanına bağlantısı gösterilmiştir.



Şekil 3.18. Tensorboard ile Eğitim Sürecinin İzlenmesi

3.2.6. Mobil uygulama

Gelişen teknolojiyle birlikte mobil cihazlar yaşantımızda önemli bir yer tutmaktadır. Sağlık, finans, eğitim vb. amaçlar için çeşitli uygulamalar kullanılmaktadır. Dolayısıyla, geliştirilen bitki tanıma uygulamasının sahada çalışan orman mühendisleri ve botanik alanında çalışan uzmanlarca etkin bir şekilde kullanılabilmesi için mobil uygulama geliştirmenin önemi görülmüştür.

Google tarafından geliştirilen Android ve Apple tarafından geliştirilen IOS' un mobil pazarda hâkimiyet kurdukları bilinmektedir. Geliştirilecek olan uygulamanın hedef kitleye hitap etmesi açısından mobil pazardaki kullanıcı sayıları dikkate alınmıştır.

Dünya genelinde tüm mobil cihaz kullanıcılarının %74,45'i Android tercih ederken, %22,85'i IOS' u tercih etmektedir. Bada, MeeGo, Symbian gibi diğer mobil işletim sistemlerini tercih edenlerin oranı ise %2.7'dir (Casserly 2019).

Kullanıcı sayıları dikkate alınarak projenin mobil katmanının Android'de yazılması uygun görülmüştür. Mobil uygulama için Android Studio geliştirme ortamı ve Java dili kullanılmıştır.

Mobil uygulama kullanıcı dostu bir ara yüze sahip olacak şekilde tasarlanmıştır. Sol tarafta açılır kapanır hamburger menü, sağ tarafta ise ayarlar menüsü bulunmaktadır. Ayrıca, kullanıcıların geliştiricilerle iletişime geçebilmesi için iletişim ikonu bulunmaktadır. Sol hamburger menüde bitki tanımlama fonksiyonu bulunmaktadır.

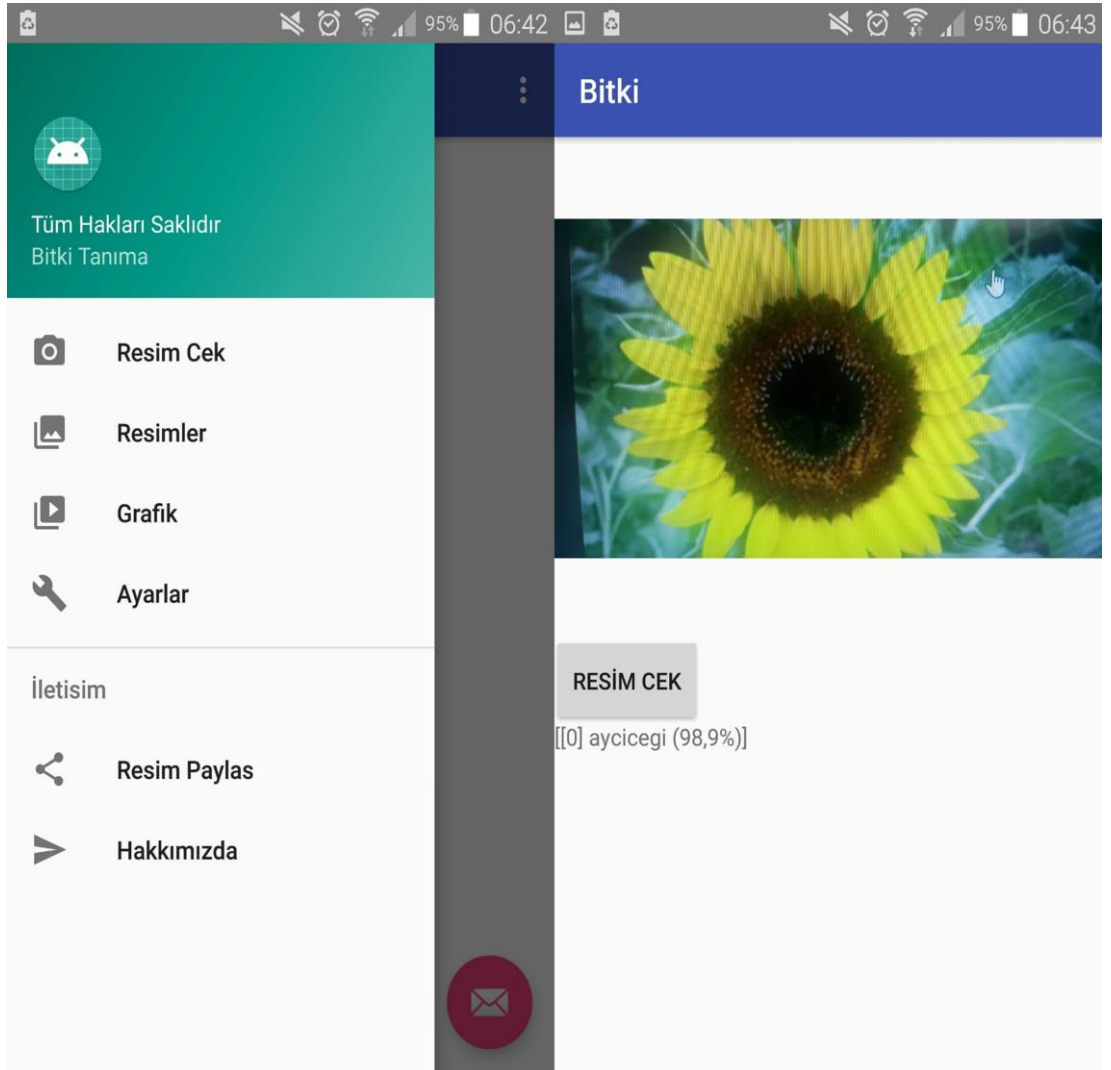
Uygulama fotoğraf çekmek işlevi için Android Kamera Api'sini, çekilen fotoğrafı kaydedebilmek için ise Android Depolama Api'sini kullanmaktadır. Dolayısıyla uygulama kurulduktan sonra bu iki Api için kullanıcıdan izin istemektedir. Uygulama Android 6.x sürümlerinde çalışmaktadır.

Bitki tanımlama işlevinin herhangi bir ağ bağlantısına ihtiyaç duymamaktadır. En iyi eğitim sonuçlarını elde ettiğimiz döndürülmüş verileri de içeren veri setiyle 1,0 ağ genişliği ve 224px çözünürlükle eğitilen ağdan elde edilen model mobil uygulamaya gömülmüştür.

Kullanıcının çektiği görüntü anlık olarak modele gitmekte ve bu modelden eşleştirilen sonuç yine aynı ekranda kullanıcıya gösterilmektedir. Sonuç gösterilirken en iyi tahmin oranına sahip bitkinin id' si, adı ve tahmin oranı gösterilmektedir.

Geliştirilen web uygulamasındaki bitkilere ait Latince tür adı, bitkinin ömrü (çok yıllık, tek yıllık), yapısı(otsu, odunsu vb), çiçeklenme süresi, habitatu (kayalık, çayırılık vb.), görüldüğü rakım, endemik bitki olup olmadığı, kökeni (Avrupa-Sibirya, İran-Turan vb.), Türkiye’deki dağılımı, Dünya’daki dağılımı ve bulunduğu iller gibi verilerin bitki id’ si ile web uygulamasından alınarak mobil uygulamada gösterilmesi planlanmaktadır.

Şekil 3.18’de, solda mobil uygulamanın genel ara yüzü ve sağda çekilen görüntüden bitki tanıma işlevinin görüntüsü sunulmuştur.



Şekil 3.19. Mobil Uygulama

4. ARAŞTIRMA BULGULARI ve TARTIŞMA

Yapılan çalışmada DAEYSA ile Doğu Anadolu Bölgesi'nde yetişen 14 türe ait bitki türü sınıflandırılmıştır. Ağ genişliğinin, görüntü çözünürlüğünün ve veri boyutunun sınıflandırma başarısı üzerindeki etkisini incelemek adına çeşitli yöntemler kullanılmıştır.

Girdi çözünürlüğü sabit tutularak, ağ genişliği 4 farklı seçenekle kodlanmış ve ağın genişlemesi ya da daralmasının sınıflandırma başarısı üzerindeki etkisi incelenmiştir.

Ağ genişliği sabit tutularak, girdi görüntüsünün çözünürlüğü 4 farklı seçenekle kodlanmış ve çözünürlük boyutunun sınıflandırma başarısı üzerindeki etkisi incelenmiştir.

Veri boyutunun öğrenme üzerindeki etkisini inceleyebilmek için asıl veri setindeki görüntülerin döndürülmesiyle elde edilen veri seti de kullanılmıştır. Çizelge 4.1'de kullanılan veri setlerinin veri sayısı, eğitim için ayrılan veri sayısı ve test için ayrılan veri sayısı listelenmiştir.

Çizelge 4.1. Veri Setlerinin Türlerine Dağılımı

Tür Adı	Orijinal Veri Seti		Döndürme Sonucu Elde Edilen Veri Seti	
	Eğitim Verisi	Test Verisi	Eğitim Verisi	Test Verisi
Helianthus	49	8	676	8
Ranunculus	75	8	988	8
Crocus	99	9	1287	9
Astragalus membranaceus	115	6	1446	6
Anemone coronaria L	56	7	749	7
Galanthus nivalis	18	10	326	10
Bellis perennis	49	9	687	9
Tulipa gerneirana	116	7	1469	7
Viola reichenbachiana	78	8	1024	8
Iris versicolor	70	9	939	9
Tussilago farfara	65	14	934	14
Hyacinthus orientalis	96	7	1229	7
Fritillaria imperialis	77	11	1045	11
Lilium	132	7	1661	7
Toplam	1095	120	14460	120

Eğitim aşaması için Çizelge 4.1’de ayrıntıları verilen 2 veri seti kullanılmış ve hep aynı model GİB ile eğitilmiştir. Eğitim aşaması için orijinal veri setinden 1095 görüntü, döndürülmüş verileri de içeren veri setinden 14460 görüntü kullanılmıştır. Test aşamasında ise orijinal veri setinden 120, döndürülmüş verileri de içeren veri setinden 120, Google görsel aramalardan da 120 görüntü kullanılmıştır.

4.1. Ağ Genişliğinin Öğrenme Üzerindeki Etkisi

Ağ genişliğinin öğrenme üzerinde etkisini araştırmak için 224 px girdi çözünürlüğüne karşın 1,0 ağ genişliği, 0,75 ağ genişliği, 0,50 ağ genişliği ve 0,25 ağ genişliği test edilmiş ve sonuçları aşağıda sunulmuştur.

4.1.1. 224px çözünürlük 1,0 ağ genişliği

Eğitim için orijinal veri seti kullanılmıştır. Ağ, 224px çözünürlük ve 1,0 ağ genişliği parametreleriyle eğitilmiştir. Çizelge 4.2’de bu hiper parametrelerle alınan test sonuçları gösterilmiştir.

Çizelge 4.2. 224px Çözünürlük ve 1,0 Ağ Genişliği Sonuçları (Orijinal Veri Seti)

Tür Adı	Veri Setinden Ayrılan Test Görüntüleri	Başarı Oranı (%)	Google’dan Rastgele Seçilmiş Test Görüntüleri	Başarı Oranı (%)
Helianthus	8	45,6	8	41,7
Ranunculus	8	42,7	8	38,0
Crocus	9	41,5	9	39,3
Astragalus membranaceus	6	38,1	6	36,2
Anemone coronaria L	7	41,9	7	40,6
Galanthus nivalis	10	42,3	10	40,8
Bellis perennis	9	47,2	9	44,6
Tulipa gerneirana	7	51,0	7	48,2
Viola reichenbachiana	8	41,9	8	38,7
Iris versicolor	9	42,1	9	40,3
Tussilago farfara	14	42,3	14	39,1
Hyacinthus orientalis	7	42,5	7	36,7
Fritillaria imperialis	11	42,8	11	38,5
Lilium	7	53,2	7	50,2
Toplam/Ortalama	120	43,9	120	40,9

Bu hiper parametreler kullanılarak orijinal veri setiyle eğitilen modelde kendi veri setimizden ayrılan test görüntüleri ile ortalama %43,9; Google' dan rastgele seçilen görüntülerle yapılan testlerde ise ortalama %40,9 başarılı sınıflandırma yapıldığı görülmüştür.

Eğitim için orijinal verilerin döndürülmesiyle elde edilen veri seti kullanılmıştır. Ağ, 224px çözünürlük ve 1,0 ağ genişliği parametreleriyle eğitilmiştir. Çizelge 4.3'te bu hiper parametrelerle alınan test sonuçları gösterilmiştir.

Çizelge 4.3. 224px Çözünürlük ve 1,0 Ağ Genişliği Sonuçları (Döndürülmüş Veri Seti)

Tür Adı	Veri Setinden Ayrılan Test Görüntüleri	Başarı Oranı (%)	Google'dan Rastgele Seçilmiş Test Görüntüleri	Başarı Oranı (%)
Helianthus	8	92,4	8	91,8
Ranunculus	8	90,7	8	89,9
Crocus	9	89,5	9	88,1
Astragalus membranaceus	6	89,2	6	88,3
Anemone coronaria L	7	90,3	7	89,4
Galanthus nivalis	10	92,1	10	90,4
Bellis perennis	9	93,7	9	91,8
Tulipa gerneirana	7	94,8	7	92,6
Viola reichenbachiana	8	89,5	8	93,5
Iris versicolor	9	91,3	9	90,6
Tussilago farfara	14	91,7	14	89,5
Hyacinthus orientalis	7	92,2	7	90,4
Fritillaria imperialis	11	92,3	11	91,7
Lilium	7	95,1	7	93,6
Toplam/Ortalama	120	91,8	120	90,8

Bu hiper parametreler kullanılarak döndürülmüş verileri de içeren veri setiyle eğitilen modelde kendi veri setimizden ayrılan test görüntüleri ile ortalama %91,8; Google' dan rastgele seçilen görüntülerle yapılan testlerde ise ortalama %90,8 başarılı sınıflandırma yapıldığı görülmüştür.

Ağ hiper parametreleri sabit kalınca artan veri boyutunun sınıflandırma üzerinde pozitif etkisi olduğu gözlemlenmiştir.

4.1.2. 224px çözünürlük 0,75 ağ genişliği

Eğitim için orijinal veri seti kullanılmıştır. Ağ, 224px çözünürlük ve 0,75 ağ genişliği parametreleriyle eğitilmiştir. Çizelge 4.4'te bu hiper parametrelerle alınan test sonuçları gösterilmiştir.

Çizelge 4.4. 224px Çözünürlük ve 0,75 Ağ Genişliği Sonuçları (Orijinal Veri Seti)

Tür Adı	Veri Setinden Ayrılan Test Görüntüleri	Başarı Oranı (%)	Google'dan Rastgele Seçilmiş Test Görüntüleri	Başarı Oranı (%)
Helianthus	8	43,8	8	37,8
Ranunculus	8	41,4	8	35,9
Crocus	9	39,9	9	36,7
Astragalus membranaceus	6	36,7	6	36,4
Anemone coronaria L	7	39,5	7	38,2
Galanthus nivalis	10	40,5	10	38,4
Bellis perennis	9	46	9	41,8
Tulipa geneirana	7	48,1	7	46,3
Viola reichenbachiana	8	40,5	8	37,2
Iris versicolor	9	41,2	9	38,5
Tussilago farfara	14	41,7	14	36,2
Hyacinthus orientalis	7	40,9	7	33,7
Fritillaria imperialis	11	40,9	11	37,1
Lilium	7	50,2	7	48,4
Toplam/Ortalama	120	42,2	120	38,7

Bu hiper parametreler kullanılarak orijinal veri setiyle eğitilen modelde kendi veri setimizden ayrılan test görüntüleri ile ortalama %42,2; Google' dan rastgele seçilen

görüntülerle yapılan testlerde ise ortalama %38,7 başarılı sınıflandırma yapıldığı görülmüştür.

Eğitim için orijinal verilerin döndürülmesiyle elde edilen veri seti kullanılmıştır. Ağ, 224px çözünürlük ve 0,75 ağ genişliği parametreleriyle eğitilmiştir. Çizelge 4.5'te bu hiper parametrelerle alınan test sonuçları gösterilmiştir.

Çizelge 4.5. 224px Çözünürlük ve 0,75 Ağ Genişliği Sonuçları (Döndürülmüş Veri Seti)

Tür Adı	Veri Setinden Ayrılan Test Görüntüleri	Başarı Oranı (%)	Google'dan Rastgele Seçilmiş Test Görüntüleri	Başarı Oranı (%)
Helianthus	8	91,8	8	90,4
Ranunculus	8	89,2	8	88,2
Crocus	9	88	9	86,8
Astragalus membranaceus	6	87,8	6	86,4
Anemone coronaria L	7	89,1	7	87,9
Galanthus nivalis	10	91,5	10	89,2
Bellis perennis	9	91,9	9	90,2
Tulipa gerneirana	7	93,2	7	90,4
Viola reichenbachiana	8	88,1	8	92
Iris versicolor	9	90	9	88,9
Tussilago farfara	14	90,2	14	87,6
Hyacinthus orientalis	7	91,7	7	88,5
Fritillaria imperialis	11	90,9	11	90,8
Lilium	7	94,2	7	92,7
Toplam/Ortalama	120	90,5	120	89,2

Bu hiper parametreler kullanılarak döndürülmüş verileri de içeren veri setiyle eğitilen modelde kendi veri setimizden ayrılan test görüntüleri ile ortalama %90,5; Google' dan rastgele seçilen görüntülerle yapılan testlerde ise ortalama %89,2 başarılı sınıflandırma yapıldığı görülmüştür.

Ağ hiper parametreleri sabit kalınca artan veri boyutunun sınıflandırma üzerinde pozitif etkisi olduğu gözlemlenmiştir.

4.1.3. 224px çözünürlük 0,50 ağ genişliği

Eğitim için orijinal veri seti kullanılmıştır. Ağ, 224px çözünürlük ve 0,50 ağ genişliği parametreleriyle eğitilmiştir. Çizelge 4.6'da bu hiper parametrelerle alınan test sonuçları gösterilmiştir.

Çizelge 4.6 224px Çözünürlük ve 0,50 Ağ Genişliği Sonuçları (Orijinal Veri Seti)

Tür Adı	Veri Setinden Ayrılan Test Görüntüleri	Başarı Oranı (%)	Google'dan Rastgele Seçilmiş Test Görüntüleri	Başarı Oranı (%)
Helianthus	8	39,5	8	33,1
Ranunculus	8	37,3	8	31,7
Crocus	9	35,2	9	32,4
Astragalus membranaceus	6	32,4	6	32,4
Anemone coronaria L	7	35,1	7	34,5
Galanthus nivalis	10	36,5	10	34,9
Bellis perennis	9	41,4	9	37,2
Tulipa gerneirana	7	43,7	7	42,1
Viola reichenbachiana	8	36,2	8	33,0
Iris versicolor	9	36,9	9	34,6
Tussilago farfara	14	37,1	14	32,2
Hyacinthus orientalis	7	36,3	7	29,5
Fritillaria imperialis	11	36,5	11	33,2
Lilium	7	45,8	7	44,8
Toplam/Ortalama	120	37,8	120	34,7

Bu hiper parametreler kullanılarak orijinal veri setiyle eğitilen modelde kendi veri setimizden ayrılan test görüntüleri ile ortalama %37,8; Google' dan rastgele seçilen görüntülerle yapılan testlerde ise ortalama %34,7 başarılı sınıflandırma yapıldığı görülmüştür.

Eđitim için orijinal verilerin döndürülmesiyle elde edilen veri seti kullanılmıřtır. Ađ, 224px çözünürlük ve 0,50 ađ geniřliđi parametreleriyle eđitilmiřtir. Çizelge 4.7’de bu hiper parametrelerle alınan test sonuçları gösterilmiřtir.

Çizelge 4.7. 224px Çözünürlük ve 0,50 Ađ Geniřliđi Sonuçları (Döndürölmüş Veri Seti)

Tür Adı	Veri Setinden Ayrılan Test Görüntüleri	Başarı Oranı (%)	Google’dan Rastgele Seçilmiş Test Görüntüleri	Başarı Oranı (%)
Helianthus	8	87,2	8	86,2
Ranunculus	8	85,4	8	84,2
Crocus	9	83,5	9	82,8
Astragalus membranaceus	6	83,9	6	82,5
Anemone coronaria L	7	85,3	7	83,1
Galanthus nivalis	10	87,8	10	85,1
Bellis perennis	9	87,4	9	86,7
Tulipa gerneirana	7	89,0	7	86,4
Viola reichenbachiana	8	83,8	8	88,6
Iris versicolor	9	85,4	9	84,8
Tussilago farfara	14	86,7	14	83,1
Hyacinthus orientalis	7	87,1	7	84,1
Fritillaria imperialis	11	85,2	11	86,3
Lilium	7	90,0	7	88,8
Toplam/Ortalama	120	86,2	120	85,2

Bu hiper parametreler kullanılarak döndürölmüş verileri de içeren veri setiyle eđitilen modelde kendi veri setimizden ayrılan test görüntüleri ile ortalama %86,2; Google’ dan rastgele seçilen görüntülerle yapılan testlerde ise ortalama %85,2 başarılı sınıflandırma yapıldığı görölmüşür.

Ağ hiper parametreleri sabit kalınca artan veri boyutunun sınıflandırma üzerinde pozitif etkisi olduğu gözlemlenmiştir.

4.1.4. 224px çözünürlük 0,25 ağ genişliği

Eğitim için orijinal veri seti kullanılmıştır. Ağ, 224px çözünürlük ve 0,25 ağ genişliği parametreleriyle eğitilmiştir. Çizelge 4.8’de bu hiper parametrelerle alınan test sonuçları gösterilmiştir.

Çizelge 4.8. 224px Çözünürlük ve 0,25 Ağ Genişliği Sonuçları (Orijinal Veri Seti)

Tür Adı	Veri Setinden Ayrılan Test Görüntüleri	Başarı Oranı (%)	Google’dan Rastgele Seçilmiş Test Görüntüleri	Başarı Oranı (%)
Helianthus	8	28,2	8	22,2
Ranunculus	8	26,9	8	20,2
Crocus	9	24,1	9	21,8
Astragalus membranaceus	6	21,4	6	21,5
Anemone coronaria L	7	24,3	7	23,1
Galanthus nivalis	10	25,7	10	23,1
Bellis perennis	9	30,1	9	26,7
Tulipa gerneirana	7	31,3	7	31,4
Viola reichenbachiana	8	24,3	8	22,6
Iris versicolor	9	24,9	9	23,4
Tussilago farfara	14	26,2	14	21,2
Hyacinthus orientalis	7	25,9	7	18,8
Fritillaria imperialis	11	25,7	11	22,1
Lilium	7	34,3	7	33,4
Toplam/Ortalama	120	26,7	120	23,7

Bu hiper parametreler kullanılarak orijinal veri setiyle eğitilen modelde kendi veri setimizden ayrılan test görüntüleri ile ortalama %26,7; Google’ dan rastgele seçilen görüntülerle yapılan testlerde ise ortalama %23,7 başarılı sınıflandırma yapıldığı görülmüştür.

Eđitim için orijinal verilerin döndürülmesiyle elde edilen veri seti kullanılmıřtır. Ađ, 224px çözünürlük ve 0,25 ađ geniřliđi parametreleriyle eđitilmiřtir. Çizelge 4.9'da bu hiper parametrelerle alınan test sonuçları gösterilmiřtir.

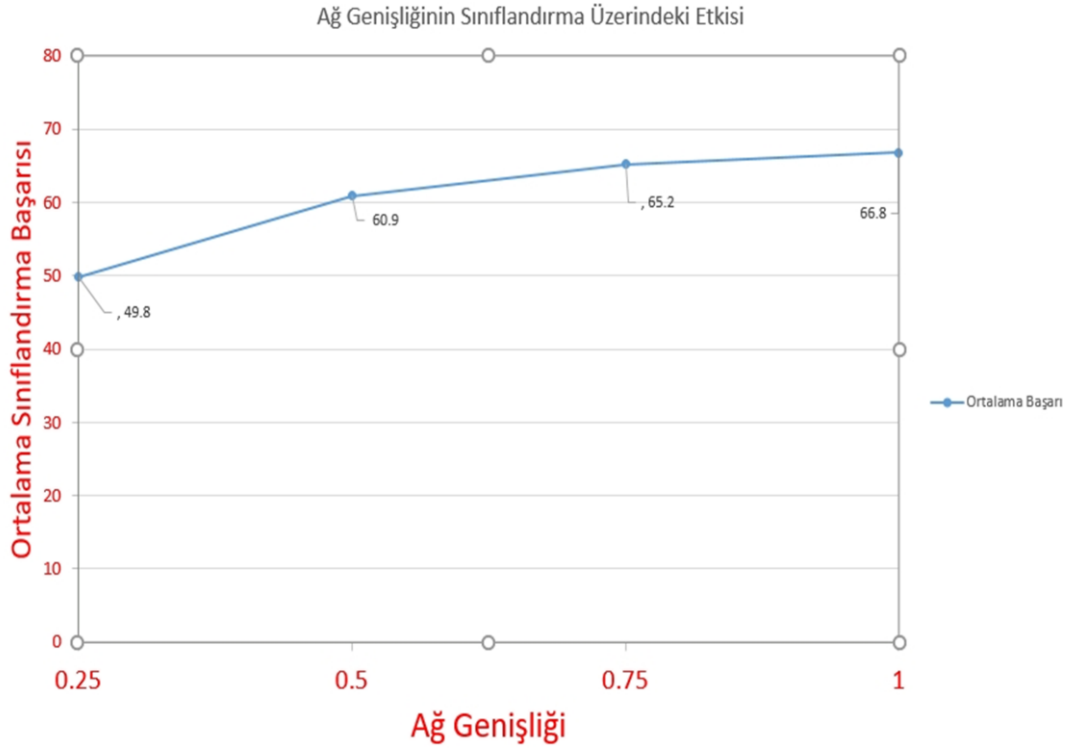
Çizelge 4.9. 224px Çözünürlük ve 0,25 Ađ Geniřliđi Sonuçları (Döndürülmüř Veri Seti)

Tür Adı	Veri Setinden Ayrılan Test Görüntüleri	Başarı Oranı (%)	Google'dan Rastgele Seçilmiř Test Görüntüleri	Başarı Oranı (%)
Helianthus	8	76,2	8	75,2
Ranunculus	8	74,4	8	73,3
Crocus	9	72,5	9	71,6
Astragalus membranaceus	6	72,9	6	71,7
Anemone coronaria L	7	74,3	7	72,8
Galanthus nivalis	10	76,8	10	74,1
Bellis perennis	9	76,4	9	75
Tulipa gerneirana	7	78,2	7	75,6
Viola reichenbachiana	8	72,9	8	77,2
Iris versicolor	9	74,5	9	73,5
Tussilago farfara	14	75,4	14	72,4
Hyacinthus orientalis	7	75,1	7	73,8
Fritillaria imperialis	11	74,8	11	75
Lilium	7	76,2	7	76,1
Toplam/Ortalama	120	75	120	74,1

Bu hiper parametreler kullanılarak döndürülmüř verileri de içeren veri setiyle eđitilen modelde kendi veri setimizden ayrılan test görüntüleri ile ortalama %75; Google'dan rastgele seçilen görüntülerle yapılan testlerde ise ortalama %74,1 başarılı sınıflandırma yapıldıđı görülmüřtür.

Ađ hiper parametreleri sabit kalınca artan veri boyutunun sınıflandırma üzerinde pozitif etkisi olduđu gözlemlenmiřtir.

Nihai olarak orijinal veri seti ve döndürülmüş veri seti sonuçlarının ortalaması alınmıştır. Şekil 4.1’ de görüldüğü üzere 224px çözünürlük sabit olmak kaydıyla, ağ genişliğinin kademeli olarak artırılmasının sınıflandırma başarısı üzerinde keskin artışlara neden olmaktadır.



Şekil 4.1. Ağ Genişliğinin Sınıflandırma Üzerindeki Etkisi

4.2. Görüntü Çözünürlüğünün Öğrenme Üzerindeki Etkisi

Bölüm 4.1’de tespit edildiği üzere, ağ genişliğinin artması sınıflandırma sonuçlarını olumlu etkilemektedir. Bu sebeple, araştırmanın bu bölümünde en iyi sonuçları veren 1,0 ağ genişliği sabit tutulmuştur. Görüntü çözünürlüğünün öğrenme üzerindeki etkisini araştırmak için 192px, 160px ve 128px görüntü çözünürlükleriyle test edilmiş ve sonuçları aşağıda sunulmuştur. 224px görüntü çözünürlüğü Bölüm 4.1.1’ de test edildiği için tekrar test edilmemiş ve aynı sonuçlar kullanılmıştır.

4.2.1. 1,0 ağ genişliği 192px çözünürlük

Eğitim için orijinal veri seti kullanılmıştır. Ağ, 1,0 ağ genişliği ve 192px çözünürlük parametreleriyle eğitilmiştir. Çizelge 4.10'da bu hiper parametrelerle alınan test sonuçları gösterilmiştir.

Çizelge 4.10. 1,0 Ağ Genişliği ve 192px Çözünürlük Sonuçları (Orijinal Veri Seti)

Tür Adı	Veri Setinden Ayrılan Test Görüntüleri	Başarı Oranı (%)	Google'dan Rastgele Seçilmiş Test Görüntüleri	Başarı Oranı (%)
Helianthus	8	44,4	8	40,1
Ranunculus	8	41,5	8	36,5
Crocus	9	40,2	9	38,3
Astragalus membranaceus	6	37,1	6	35,2
Anemone coronaria L	7	40,5	7	39,6
Galanthus nivalis	10	41,3	10	39,1
Bellis perennis	9	46,1	9	43,0
Tulipa gerneirana	7	49,7	7	47,9
Viola reichenbachiana	8	40,8	8	37,7
Iris versicolor	9	41,5	9	39,8
Tussilago farfara	14	41,8	14	38,8
Hyacinthus orientalis	7	41,2	7	35,2
Fritillaria imperialis	11	41,3	11	37,7
Lilium	7	52,3	7	49,0
Toplam/Ortalama	120	42,8	120	39,8

Bu hiper parametreler kullanılarak orijinal veri setiyle eğitilen modelde kendi veri setimizden ayrılan test görüntüleri ile ortalama %42,8; Google' dan rastgele seçilen görüntülerle yapılan testlerde ise ortalama %39,8 başarılı sınıflandırma yapıldığı görülmüştür.

Eđitim için orijinal verilerin döndürülmesiyle elde edilen veri seti kullanılmıřtır. Ađ, 1,0 ađ geniřliđi ve 192px çözünürlük parametreleriyle eđitilmiřtir. Çizelge 4.11’de bu hiper parametrelerle alınan test sonuçları gösterilmiřtir.

Çizelge 4.11. 1,0 Ađ Geniřliđi ve 192px Çözünürlük Sonuçları (Döndürölmüş Veri Seti)

Tür Adı	Veri Setinden Ayrılan Test Görüntüleri	Başarı Oranı (%)	Google’dan Rastgele Seçilmiş Test Görüntüleri	Başarı Oranı (%)
Helianthus	8	91,3	8	90,7
Ranunculus	8	89,6	8	88,2
Crocus	9	88,4	9	87,3
Astragalus membranaceus	6	88,1	6	87,9
Anemone coronaria L	7	89,2	7	88,6
Galanthus nivalis	10	91,9	10	89,8
Bellis perennis	9	92,6	9	90,2
Tulipa gerneirana	7	93,7	7	91,0
Viola reichenbachiana	8	88,4	8	92,7
Iris versicolor	9	90,2	9	89,3
Tussilago farfara	14	90,6	14	88,3
Hyacinthus orientalis	7	91,1	7	89,7
Fritillaria imperialis	11	91,2	11	90,5
Lilium	7	94,2	7	92,2
Toplam/Ortalama	120	90,7	120	89,7

Bu hiper parametreler kullanılarak döndürölmüş verileri de içeren veri setiyle eđitilen modelde kendi veri setimizden ayrılan test görüntüleri ile ortalama %90,7; Google’ dan rastgele seçilen görüntülerle yapılan testlerde ise ortalama %89,7 başarılı sınıflandırma yapıldığı görölmüşür.

Ađ hiper parametreleri sabit kalınca artan veri boyutunun sınıflandırma üzerinde pozitif etkisi olduđu gözlemlenmiřtir.

4.2.2. 1,0 ağ genişliği 160px çözünürlük

Eğitim için orijinal veri seti kullanılmıştır. Ağ, 1,0 ağ genişliği ve 160px çözünürlük parametreleriyle eğitilmiştir. Çizelge 4.12’de bu hiper parametrelerle alınan test sonuçları gösterilmiştir.

Çizelge 4.12. 1,0 Ağ Genişliği ve 160px Çözünürlük Sonuçları (Orijinal Veri Seti)

Tür Adı	Veri Setinden Ayrılan Test Görüntüleri	Başarı Oranı (%)	Google’dan Rastgele Seçilmiş Test Görüntüleri	Başarı Oranı (%)
Helianthus	8	42,3	8	38,7
Ranunculus	8	39,6	8	35,2
Crocus	9	38,4	9	36,3
Astragalus membranaceus	6	35,1	6	33,2
Anemone coronaria L	7	38,2	7	37,6
Galanthus nivalis	10	39,3	10	37,8
Bellis perennis	9	44,6	9	41,2
Tulipa gerneirana	7	48,7	7	45,2
Viola reichenbachiana	8	38,4	8	35,7
Iris versicolor	9	39,2	9	37,3
Tussilago farfara	14	39,6	14	36,1
Hyacinthus orientalis	7	39,1	7	33,7
Fritillaria imperialis	11	39,2	11	35,5
Lilium	7	50,2	7	47,2
Toplam/Ortalama	120	40,8	120	37,9

Bu hiper parametreler kullanılarak orijinal veri setiyle eğitilen modelde kendi veri setimizden ayrılan test görüntüleri ile ortalama %40,8; Google’ dan rastgele seçilen görüntülerle yapılan testlerde ise ortalama %37,9 başarılı sınıflandırma yapıldığı görülmüştür.

Eđitim için orijinal verilerin döndürülmesiyle elde edilen veri seti kullanılmıştır. Ağ, 1,0 ağ genişliği ve 160px çözünürlük parametreleriyle eğitilmiştir. Çizelge 4.13'te bu hiper parametrelerle alınan test sonuçları gösterilmiştir.

Çizelge 4.13. 1,0 Ağ Genişliği ve 160px Çözünürlük Sonuçları (Döndürülmüş Veri Seti)

Tür Adı	Veri Setinden Ayrılan Test Görüntüleri	Başarı Oranı (%)	Google'dan Rastgele Seçilmiş Test Görüntüleri	Başarı Oranı (%)
Helianthus	8	89,4	8	88,9
Ranunculus	8	87,7	8	86,1
Crocus	9	86,5	9	85,0
Astragalus membranaceus	6	86,2	6	85,4
Anemone coronaria L	7	87,3	7	86,5
Galanthus nivalis	10	89,1	10	87,7
Bellis perennis	9	90,7	9	88,8
Tulipa geneirana	7	91,8	7	89,3
Viola reichenbachiana	8	86,5	8	90,5
Iris versicolor	9	88,3	9	87,2
Tussilago farfara	14	88,7	14	86,4
Hyacinthus orientalis	7	89,2	7	87,0
Fritillaria imperialis	11	89,3	11	88,4
Lilium	7	92,5	7	90,4
Toplam/Ortalama	120	88,8	120	87,6

Bu hiper parametreler kullanılarak döndürülmüş verileri de içeren veri setiyle eğitilen modelde kendi veri setimizden ayrılan test görüntüleri ile ortalama %88,8; Google' dan rastgele seçilen görüntülerle yapılan testlerde ise ortalama %87,6 başarılı sınıflandırma yapıldığı görülmüştür.

Ağ hiper parametreleri sabit kalınca artan veri boyutunun sınıflandırma üzerinde pozitif etkisi olduğu gözlemlenmiştir.

4.2.3. 1,0 ağ genişliği 128px çözünürlük

Eğitim için orijinal veri seti kullanılmıştır. Ağ, 1,0 ağ genişliği ve 128px çözünürlük parametreleriyle eğitilmiştir. Çizelge 4.14'te bu hiper parametrelerle alınan test sonuçları gösterilmiştir.

Çizelge 4.14. 1,0 Ağ Genişliği ve 128px Çözünürlük Sonuçları (Orijinal Veri Seti)

Tür Adı	Veri Setinden Ayrılan Test Görüntüleri	Başarı Oranı (%)	Google'dan Rastgele Seçilmiş Test Görüntüleri	Başarı Oranı (%)
Helianthus	8	40,6	8	36,7
Ranunculus	8	37,7	8	33,2
Crocus	9	36,7	9	34,3
Astragalus membranaceus	6	33,3	6	31,1
Anemone coronaria L	7	36,1	7	35,3
Galanthus nivalis	10	37,7	10	35,5
Bellis perennis	9	42,5	9	39,8
Tulipa gerneirana	7	46,2	7	43,2
Viola reichenbachiana	8	36,5	8	33,7
Iris versicolor	9	37,1	9	35,3
Tussilago farfara	14	37,3	14	34,1
Hyacinthus orientalis	7	37,5	7	31,7
Fritillaria imperialis	11	37,8	11	33,5
Lilium	7	48,2	7	45,9
Toplam/Ortalama	120	38,9	120	35,9

Bu hiper parametreler kullanılarak orijinal veri setiyle eğitilen modelde kendi veri setimizden ayrılan test görüntüleri ile ortalama %38,9; Google' dan rastgele seçilen görüntülerle yapılan testlerde ise ortalama %35,9 başarılı sınıflandırma yapıldığı görülmüştür.

Eđitim için orijinal verilerin döndürülmesiyle elde edilen veri seti kullanılmıřtır. Ađ, 1,0 ađ geniřliđi ve 128px çözünürlük parametreleriyle eđitilmiřtir. Çizelge 4.15'te bu hiper parametrelerle alınan test sonuçları gösterilmiřtir.

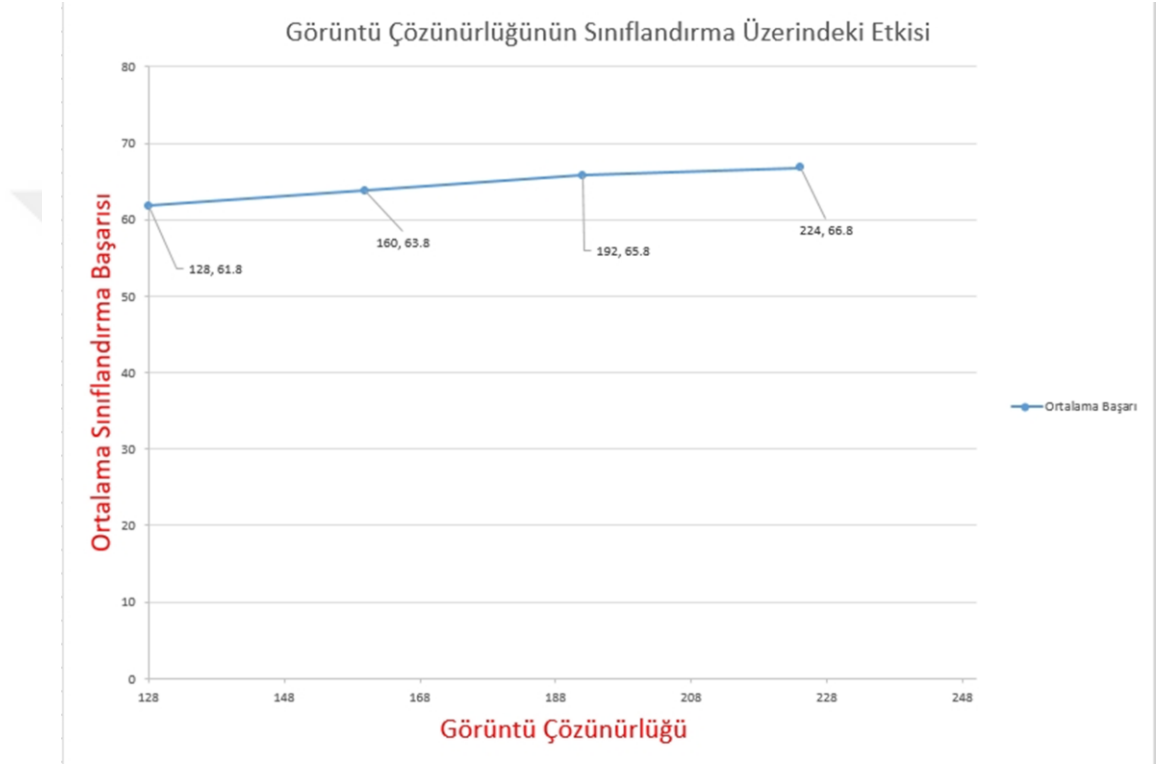
Çizelge 4.15. 1,0 Ađ Geniřliđi ve 128px Çözünürlük Sonuçları (Döndürölmüş Veri Seti)

Veri Setinden Ayrılan Test Görüntüleri	Başarı Oranı (%)	Google'dan Rastgele Seçilmiş Test Görüntüleri	Başarı Oranı (%)	Veri Setinden Ayrılan Test Görüntüleri
Helianthus	8	87,3	8	86,2
Ranunculus	8	85,7	8	84,7
Crocus	9	84,5	9	83,1
Astragalus membranaceus	6	84,2	6	83,3
Anemone coronaria L	7	85,3	7	84,5
Galanthus nivalis	10	87,1	10	85,1
Bellis perennis	9	88,7	9	86,4
Tulipa gerneirana	7	89,4	7	87,5
Viola reichenbachiana	8	84,5	8	88,3
Iris versicolor	9	86,3	9	85,6
Tussilago farfara	14	86,7	14	84,7
Hyacinthus orientalis	7	87,2	7	85,1
Fritillaria imperialis	11	87,3	11	86,8
Lilium	7	90,1	7	88,0
Toplam/Ortalama	120	86,7	120	85,6

Bu hiper parametreler kullanılarak döndürölmüş verileri de içeren veri setiyle eđitilen modelde kendi veri setimizden ayrılan test görüntüleri ile ortalama %86,7; Google' dan rastgele seçilen görüntülerle yapılan testlerde ise ortalama %85,6 başarılı sınıflandırma yapıldığı görölmüşür.

Ađ hiper parametreleri sabit kalınca artan veri boyutunun sınıflandırma üzerinde pozitif etkisi olduđu gözlemlenmiştir.

Nihai olarak orijinal veri seti ve döndürülmüş veri seti sonuçlarının ortalaması alınmıştır. Şekil 4.2’ de görüldüğü üzere 1,0 ağ genişliği sabit kalmak şartıyla, görüntü çözünürlüğünün kademeli olarak artırılmasının sınıflandırma başarısı üzerinde keskin olmayan artışlara neden olmaktadır.



Şekil 4.2. Görüntü Çözünürlüğünün Sınıflandırma Üzerindeki Etkisi

4.3. Maliyet Analizi

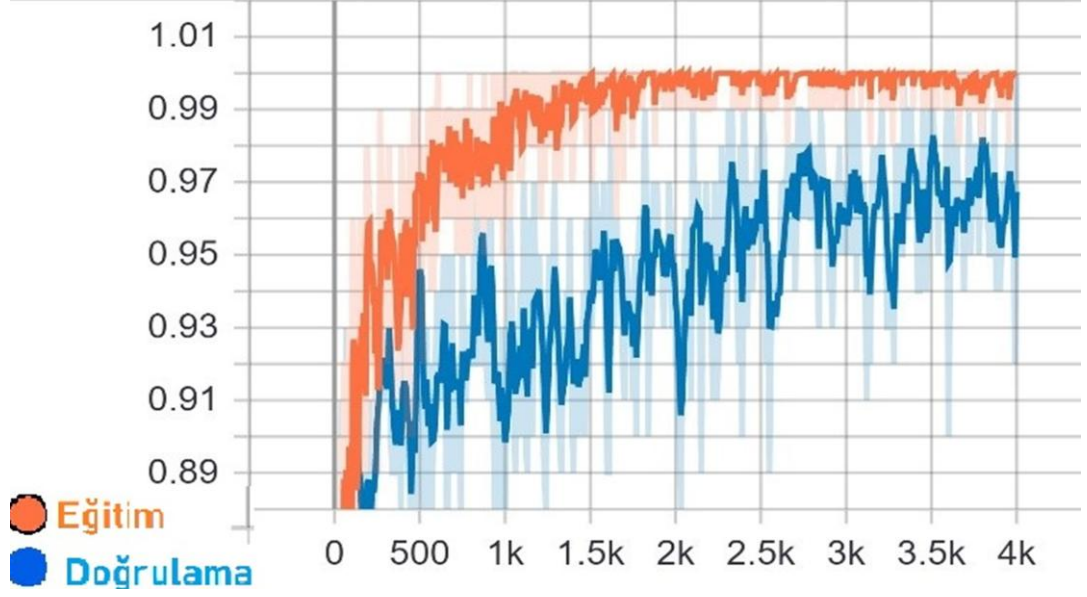
Çalışmanın bu aşamasında, farklı hiper parametrelerin maliyet analizi yapılmıştır. Farklı hiper parametreler kullanılarak üretilen parametre sayısı izlenmiş, öznelilik çıkarma aşamasındaki toplam-çarpım sayısı elde edilmiştir. Bulgular Çizelge 4.16’da verilmiştir.

Çizelge 4.16. Maliyet Analizi

Hiper Parametre	Veri Kümesi	Üretilen Parametre	Toplam Çarpım Sayısı
1,0 Ağ Genişliği 224px Çözünürlük	Orijinal	4,3 Milyon	582 Milyon
	Döndürülmüş	8,5 Milyon	824 Milyon
0,75 Ağ Genişliği 224px Çözünürlük	Orijinal	2,8 Milyon	346 Milyon
	Döndürülmüş	4,9 Milyon	729 Milyon
0,50 Ağ Genişliği 224px Çözünürlük	Orijinal	1,4 Milyon	168 Milyon
	Döndürülmüş	2,6 Milyon	349 Milyon
0,25 Ağ Genişliği 224px Çözünürlük	Orijinal	0,5 Milyon	52 Milyon
	Döndürülmüş	0,9 Milyon	96 Milyon
192px Çözünürlük 1,0 Ağ Genişliği	Orijinal	4,3 Milyon	481 Milyon
	Döndürülmüş	4,5 Milyon	946 Milyon
160px Çözünürlük 1,0 Ağ Genişliği	Orijinal	4,3 Milyon	296 Milyon
	Döndürülmüş	4,5 Milyon	548 Milyon
128px Çözünürlük 1,0 Ağ Genişliği	Orijinal	4,3 Milyon	191 Milyon
	Döndürülmüş	4,5 Milyon	339 Milyon

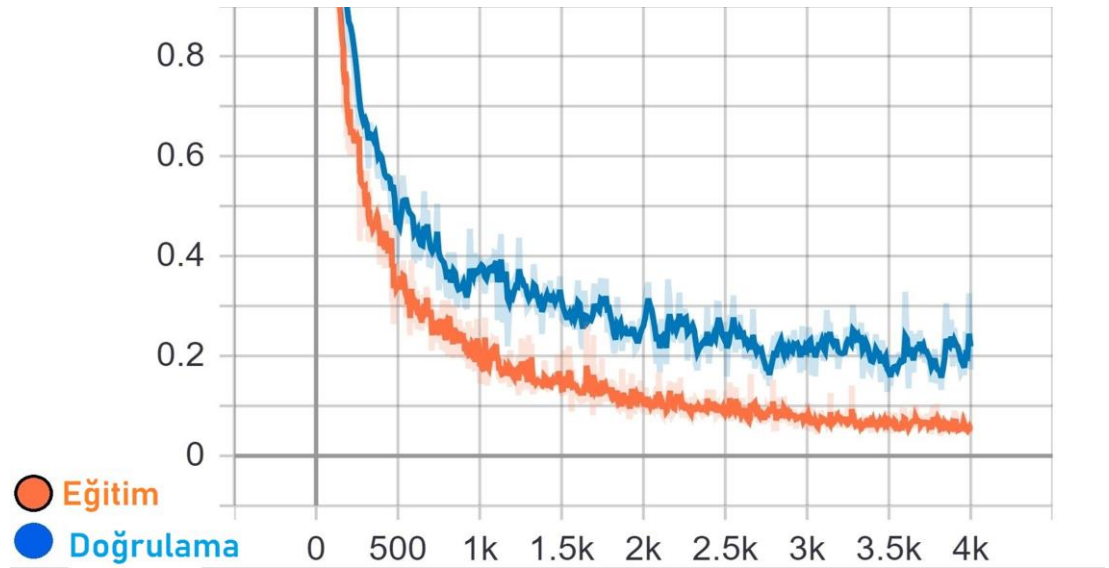
Çizelge 4.16’da görüldüğü gibi, ağ genişliği daraldıkça toplam çarpım sayısı azalmakta ve üretilen parametre sayısı da doğru orantılı olarak azalmaktadır. Buna rağmen, ağ genişliği en iyi seviyede sabitlenip, görüntü çözünürlüğü düşürülünce üretilen parametre sayısı değişmemekte fakat toplam çarpım sayısı görüntü çözünürlüğü ile doğru orantılı olarak düşmektedir.

En iyi sonuçları veren hiper parametreler (1,0 ağ genişliği, 224px çözünürlük) ile eğitim süreci izlenerek keskinlik değerleri elde edilmiştir. Eğitim doğruluğu, doğru sınıfta etiket edilmiş verilerin yüzdesini ifade ederken, doğrulama doğruluğu ise eğitim sonucundaki eşleşmenin keskinliğini ifade etmektedir. Şekil 4.3’te eğitim ve doğrulama aşamasındaki keskinlik grafiği sunulmuştur.



Şekil 4.3 Eğitim ve Doğrulama Aşamasındaki Keskinlik Değer Grafiği

Bununla birlikte, çapraz entropi, öğrenme süreci boyunca kayıp fonksiyonudur. Bu kıstas, öğrenme sürecinin nasıl verimli geçtiğini gösteren en önemli kıstastır. Düşük sayılar iyi bir öğrenme sürecini göstermektedir. Şekil 4.4'te eğitim sürecindeki kayıp fonksiyonu değerleri gösterilmiştir.



Şekil 4.4. Eğitim Süresince Kayıp Fonksiyonu Değer Grafiği

5. SONUÇ

Bu çalışmada Doğu Anadolu Bölgesi'nde bulunan 14 bitki türünü derin öğrenme yöntemleriyle sınıflandırma amaçlanmıştır.

Erzurum Orman Bölge Müdürlüğü tarafından sağlanan görüntüler kullanılarak bir veri seti oluşturulmuştur. Görüntü verilerinin sağlıklı olarak muhafaza edilmesi ve veri setinin sürekli geliştirilebilir olması için Web tabanlı bir uygulama geliştirilmiştir.

Eğitim aşamasında DAEYSA kullanılmış çeşitli parametreler test edilmiş ve sınıflandırma başarısı üzerindeki etkisi incelenmiştir. Yapılan çalışmalar neticesinde en iyi performans 1,0 ağ genişliği ile 224px görüntü çözünürlüğü parametreleriyle elde edilmiştir. Ağ genişliğinin azalmasının sınıflandırma başarısını keskin olarak azalttığı, görüntü çözünürlüğünü düşürmenin ise o kadar da keskin olmayan bir biçimde sınıflandırma başarısını azalttığı görülmüştür.

Veri boyutunun öğrenme üzerindeki etkisini araştırmak için mevcut veri setindeki görseller döndürülerek veri seti 11 kat artırılmış ve bu doğrultuda sınıflandırma başarısı yaklaşık 2 katına çıkmıştır. En iyi sonuçları veren parametrelerle eğitilen model, geliştirilen Android uygulaması içine gömülmüş ve mobil ortamda sınıflandırma yapılması sağlanmıştır.

KAYNAKLAR

- Amazon. (2019, 2019). "Amazon Book Search." from <https://www.amazon.com/s?k>.
- Arxiv.org. (2019). "Arxiv.org Search Results." Retrieved 2019, from <https://arxiv.org>.
- Berkeley. (2014). "Caffe." 2018, from <https://caffe.berkeleyvision.org/>.
- Budak, V. and M. Gezer (2016). Esnek Web Ara Yüzü Tasarlanması Üzerine Bootstrap İle Bir Uygulama.
- Cassery, M. (2019, 08/02/2019). "iPhone vs Android Market Share." Retrieved 08/02/2019, 2019, from <https://www.macworld.co.uk/feature/iphone/iphone-vs-android-market-share-3691861/>.
- Chollet, F. (2016) "Xception: Deep Learning with Depthwise Separable Convolutions."
- Christenhusz, M. and J. Byng (2016). "The number of known plant species in the world and its annual increase." *Phytotaxa* **261**: 201-217.
- Deeplearning4j. (2015). "Open-source distributed deep learning for the JVM." 2018, from <https://deeplearning4j.org/>.
- Duque, A. B., L. L. J. Santos, D. Macêdo and C. Zanchettin (2019) "Squeezed Very Deep Convolutional Neural Networks for Text Classification." arXiv e-prints.
- Elmas, Ç. (2003). Yapay sinir ağları:(kuram, mimari, eğitim, uygulama), Seçkin Yayıncılık.
- Facebook. (2018). "Facebook Research - Detectron." 2018, from <https://research.fb.com/downloads/detectron/>.
- García-Gil, D., J. Luengo, S. García and F. Herrera (2017). Enabling Smart Data: Noise filtering in Big Data classification.
- Github. (2019). "Github " Retrieved 01/04/2019, 2019, from <https://github.com/>.
- Goodfellow, I., Y. Bengio and A. Courville (2016). Deep Learning, The MIT Press.
- Google. (2019). "Google Trends." Retrieved 01/04/2019, 2019, from <https://trends.google.com/>.
- Gurnani, A., V. Mavani, V. Gajjar and Y. Khandhediya (2017) "Flower Categorization using Deep Convolutional Neural Networks."
- Halit Ergezer, M. D. v. E. Ö. (2003). "Yapay Sinir Ağları ve Tanıma Sistemleri." *PiVOLKA* **06**(2): 14-17.
- He, K., X. Zhang, S. Ren and J. Sun (2015) "Deep Residual Learning for Image Recognition." arXiv e-prints.
- Howard, A. G., M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto and H. Adam (2017) "MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications."
- Hu, J., L. Shen, S. Albanie, G. Sun and E. Wu (2017) "Squeeze-and-Excitation Networks." arXiv e-prints.
- Huang, G., Z. Liu, L. van der Maaten and K. Q. Weinberger (2016) "Densely Connected Convolutional Networks." arXiv e-prints.
- ImageNet. (2016). "About ImageNet." 2018, from <http://image-net.org/about-overview>.
- ImageNet. (2019). "ImageNet Data Set." from <http://image-net.org/explore.php?wnid=n11669921>.
- Keras. (2018). "You have just found Keras." 2018, from <https://keras.io/#you-have-just-found-keras>.
- Krizhevsky, A., I. Sutskever and G. E. Hinton (2012). ImageNet classification with deep convolutional neural networks. Proceedings of the 25th International Conference on Neural Information Processing Systems - Volume 1. Lake Tahoe, Nevada, Curran Associates Inc.: 1097-1105.
- Lakesar, A. (2018). A Review on Flower Classification Using Neural Network Classifier.

- Laurene, F. (1994). *Fundamentals of neural networks: architectures, algorithms, and applications*, Prentice-Hall, Inc.
- LeCun, Y., Y. Bengio and G. Hinton (2015). *Deep Learning*.
- Lee, H.-H. and K.-S. Hong (2017). "Automatic Recognition of Flower Species in the Natural Environment." *Image and Vision Computing* **61**.
- Liu, T., S. Fang, Y. Zhao, P. Wang and J. Zhang (2015) "Implementation of Training Convolutional Neural Networks." arXiv e-prints.
- Mavani, V., A. Gurnani and J. Shah (2017) "A Novel Approach for Image Segmentation based on Histograms computed from Hue-data." arXiv e-prints.
- N. Perkins, D. and G. Salomon (1999). *Transfer Of Learning*.
- NG, A. (2015). "What Data Scientists Should Know About Deep Learning." 1. 2018.
- Nilsback, M.-E. and A. Zisserman (2006). A Visual Vocabulary for Flower Classification. 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06). Z. Feng. New York, NY, USA, USA, IEEE. **2**: 1447-1454
- Nilsback, M.-E. and A. Zisserman (2008). Automated Flower Classification over a Large Number of Classes.
- Nvidia. (2016). "What's the Difference Between Artificial Intelligence, Machine Learning, and Deep Learning?" Retrieved 10/12/2018, 2018, from <https://blogs.nvidia.com/blog/2016/07/29/whats-difference-artificial-intelligence-machine-learning-deep-learning-ai/>.
- Oztemel, E. (2010). Yapay Sinir Ağları (In Turkish) (Artificial Neural Network).
- Özkan İNİK, E. Ü. (2017). "Derin Öğrenme ve Görüntü Analizinde Kullanılan Derin Öğrenme Modelleri." *Gaziosmanpasa Journal of Scientific Research* **6**: 85-104.
- Pacifico, L. D. S., V. Macario and J. F. L. Oliveira (2018). Plant Classification Using Artificial Neural Networks. 2018 International Joint Conference on Neural Networks (IJCNN).
- Phylogeny, G. and K. Bremer (2009). Angiosperm Phylogeny Group III (APG III). An update of The Angiosperm Phylogeny Group classification for the orders and families of flowering plants: APG III. *Botanical Journal of the Linnean Society*.
- Prasad, M. V. D., R. and S. Inthiyaz (2018). An efficient classification of flower images with convolutional neural networks.
- PyTorch. (2018). "PyTorch Documentation." 2018, from <https://pytorch.org/docs/stable/index.html>.
- R. Gavai, N., Y. A. Jakhade, S. A. Tribhuvan and R. Bhattad (2017). MobileNets for flower classification using TensorFlow.
- Russakovsky, O., J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg and L. Fei-Fei (2014) "ImageNet Large Scale Visual Recognition Challenge." arXiv e-prints.
- Russell, S. and P. Norvig (2009). *Artificial Intelligence: A Modern Approach*, Prentice Hall Press.
- ScienceDirect. (2019). "ScienceDirect Search Results." Retrieved 2019, from <https://www.sciencedirect.com/>.
- Shyamal Patel, J. P. (2017). "Introduction to Deep Learning: What Are Convolutional Neural Networks?", 2018, from <https://www.mathworks.com/videos/introduction-to-deep-learning-what-are-convolutional-neural-networks--1489512765771.html>.
- Simonyan, K. and A. Zisserman (2014) "Very Deep Convolutional Networks for Large-Scale Image Recognition." arXiv e-prints.

- Szegedy, C., W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke and A. Rabinovich (2015). Going deeper with convolutions.
- Taylor-Sakya, K. (2016). Big Data: Understanding Big Data.
- Tensorflow, G. B. (2018). "About Tensorflow." from <https://www.tensorflow.org/about>.
- Theano. (2017). "Theano at a Glance." 2018, from <http://deeplearning.net/software/theano/introduction.html#introduction>.
- Türkoğlu, M. and Y. Zeynalov (2016). Ağrı Dağı' nın Florası. Ankara, AJANS-TÜRK GAZETECİLER KAMERASI MATBAACILIK İNŞ. SAN. A.Ş. .
- Xie, S., R. Girshick, P. Dollár, Z. Tu and K. He (2016) "Aggregated Residual Transformations for Deep Neural Networks." arXiv e-prints.
- Xiong, H., G. Pandey, M. Steinbach and V. Kumar (2006). Enhancing data analysis with noise removal.
- Yildiran, T., B. Yanikoglu and E. Abdullah (2014). Plant identification using local invariants.
- Zeiler, M. D. and R. Fergus (2013) "Visualizing and Understanding Convolutional Networks." arXiv e-prints.
- Zhao, Z.-Q., L.-H. Ma, Y.-m. Cheung, X. Wu, Y. Tang and C. Chen (2015). "ApLeaf: An efficient android-based plant leaf identification system." *Neurocomputing* **151**: 1112-1119.