

T.C.
KÜTAHYA DUMLUPINAR ÜNİVERSİTESİ
LİSANÜSTÜ EĞİTİM ENSTİTÜSÜ
Bilgisayar Mühendisliği Anabilim Dalı

Yüksek Lisans Tezi

**GÖRÜNTÜ İŞLEME SÜREÇ VE SONUÇLARI AÇISINDAN GOOGLE VE
YEREL MAKİNE ÖĞRENMELERİNİN KIYASLANMASI**

Danışman:

Doç. Dr. Doğan AYDIN

Hazırlayan:

Zafer KAYA

Kütahya-2021

KABUL VE ONAY

Lisansüstü Eğitim Enstitüsü Müdürlüğüne,

Bu çalışma, jürimiz tarafından

Bilgisayar Mühendisliği Anabilim Dalında YÜKSEK LİSANS TEZİ olarak kabul edilmiştir.

Tez Jürisi	İmza	
	Kabul	Red
Doç. Dr. Doğan AYDIN (Danışman)		
Doç. Dr. Muhammed Gökhan CİNSDİKİCİ		
Dr. Öğr. Üyesi Durmuş ÖZDEMİR		

Onay

İmza

Prof. Dr. Şahmurat ARIK

Lisansüstü Eğitim Enstitüsü Müdürü

BİLİMSEL ETİK VE KURALLARA UYGUNLUK BELGESİ

Yüksek Lisans tezi olarak hazırladığım “**GÖRÜNTÜ İŞLEME SÜREÇ VE SONUÇLARI AÇISINDAN GOOGLE VE YEREL MAKİNE ÖĞRENMELEİNİN KİYASLANMASI**” adlı çalışmanın öneri aşamasından sonuçlandığı aşamaya kadar geçen süreçte bilimsel etiğe ve akademik kurallara özenle uyduğumu, tez içindeki tüm bilgileri bilimsel ahlak ve gelenek çerçevesinde elde ettiğimi, tez yazım kurallarına uygun olarak hazırladığımı, bu çalışmamda doğrudan veya dolaylı olarak yaptığım her alıntıya kaynak gösterdiğimi ve yararlandığım eserlerin kaynakçada gösterilenlerden oluştuğunu beyan ederim.

...../...../2021

Zafer KAYA

ÖZET

GÖRÜNTÜ İŞLEME SÜREÇ VE SONUÇLARI AÇISINDAN GOOGLE VE YEREL MAKİNE ÖĞRENMELEERİNİN KIYASLANMASI

KAYA, Zafer

Yüksek Lisans Tezi, Bilgisayar Mühendisliği Anabilim Dalı

Tez Danışmanı: Doç. Dr. Doğan AYDIN

Şubat, 2021 77 sayfa

Düşünen ve düşündükten sonra yorum, çıkarım ve kestirimde bulunan insanoğlunun doğal zekasını ve onun biyolojik yapısındaki sinir ağlarını taklit ederek oluşturulan yapay zeka, günümüzde çok yaygın hale gelmiştir. 2019 yılına ait verilere göre yüzde 44 artışla 35,8 milyar dolara ulaşan yapay zeka harcamaları, birçok teknoloji devi ve büyüklü-küçüklü birçok şirketin ilgi sahasındadır. Yapay zekanın en önemli çalışma alanlarından biri de yapay sinir ağları(YSA)dır. Yapay sinir ağları modelleriyle yapılan yapay zeka uygulamaları, bir kısım teknoloji devi tarafından ücretsiz olarak sunulsa da şirketlerin gizlilik, ticari sır ve güvenlik gibi endişelerine tam cevap olamamaktadır. Böyle bir ortamda yerel makinelerle şirketlerin yollarına devam etmesi gerekmektedir. Yerel makinelerle öğrenmelerin sağlanması şirketler için zaman, maliyet, güvenilirlik gibi kavramları düşünmeye zorlamaktadır. Bu çalışmada, şirketlere yol göstermesi için yerel makine ve Google Firebase makine öğrenmesi SSD algoritması üzerinden karşılaştırılmıştır. Deneyler robots.ox.ac.uk tarafından sağlanan çiçek veri kümesi kullanılarak gerçekleştirilmiştir. Veri seti 8 çiçek türü (ayçiçeği, çan çiçeği, çiğdem, iris çiçeği, karahindiba, kardelen, papatya ve rüzgar gülü) içermektedir. Bu çalışmada, kamuya açık Google Firebase ve yerel makine öğrenme kıyaslaması yapılarak kendi yerel makineleriyle yapay zeka uygulama yarışına katılmak isteyen şirketler için projeksiyon hedeflenmiştir.

Anahtar Kelimeler: Bitkilerin Sınıflandırması, Görüntü İşleme, Derin Öğrenme, Yapay Sinir Ağı, Google ve Yerel Makine Öğrenmesi

ABSTRACT**COMPARISON OF GOOGLE AND LOCAL MACHINE LEARNING IN
TERMS OF IMAGE PROCESSING PROCESS AND RESULTS****KAYA, Zafer****Master Thesis, Computer Engineering****Supervisor: Asist. Prof. Doğan AYDIN****February, 2021 77 pages**

Artificial intelligence, which is created by imitating the natural intelligence of human beings who think and interpret, infer and predict, and neural networks in its biological structure, has become very common today. Artificial intelligence spending, which reached 44.8 billion dollars with an increase of 44 percent according to the data of 2019, is in the interest of many technology giants and many big and small companies. One of the most important fields of study of artificial intelligence is artificial neural networks (ANN). Although artificial intelligence applications made with artificial neural network models are offered free of charge by some technology giants, they cannot fully respond to companies' concerns such as privacy, trade secret and security. In such an environment, companies need to move on with local machines. Providing learning with local machines forces companies to consider concepts such as time, cost, and reliability. In this study, local machine and Google Firebase machine learning were compared using SSD algorithm to guide companies. Experiments were carried out using the flower data set provided by robots.ox.ac.uk. The data set includes 8 flower types (sunflower, bellflower, crocus, iris flower, dandelion, snowdrop, daisy, and wind rose). In this study, projection was aimed for companies that want to participate in the artificial intelligence application race with their local machines by comparing public Google Firebase and local machine learning.

Key Words: Classification of Plants, Image Processing, Deep Learning, Artificial Neural Network, Google and Local Machine Learning

TEŞEKKÜR

Bu çalışmamda bana yardımcı olan başta danışman hocam Doç. Dr. Doğan AYDIN'a, desteğini her daim hissettiren abim Ahmet USLU'ya, babam Seydi KAYA'ya ve Annem Şengül KAYA'ya, eşim Kübra KAYA'ya ve aileme teşekkürü bir borç bilirim.



İÇİNDEKİLER

ÖZET.....	v
ABSTRACT	vi
TABLolar LİSTESİ.....	x
ŞEKİLLER LİSTESİ.....	xi
SİMGELER VE KISALTMALAR DİZİNİ	xiii
GİRİŞ	1

BİRİNCİ BÖLÜM

YAPAY ZEKA

1.1. Yapay Zeka ve Gelecek	4
1.2. Yapay Zeka Alanları	4
1.3. Yapay Sinir Ağları (YSA)	6
1.4. Aktivasyon Fonksiyonları	7
1.5. Nesne Tanıma	9

İKİNCİ BÖLÜM

EVRIŞİMLİ YAPAY SİNİR AĞLARI (CNN)

2.1. R-CNN Yapay Sinir Ağı	13
2.2. FAST R-CNN Yapay Sinir Ağı	14
2.3. FASTER R-CNN Yapay Sinir Ağı	15
2.4. SSD Yapay Sinir Ağı	17

ÜÇÜNCÜ BÖLÜM

NESNE TANIMA VE TEZ

3.1. Tensorflow Nesne Algılama API'si	20
3.2. Derin Öğrenme Kullanarak Bitkilerde Nesne Tanıma ile İlgili Çalışmalar ..	20
3.3. Tezin Amacı	22
3.4. Tezin Önemi	22

3.5. Tezin Kapsamı	22
3.6. Tezin Metodolojisi ve Organizasyonu	23

DÖRDÜNCÜ BÖLÜM

MATERYAL VE YÖNTEM

4.1. Materyal	25
4.2. Yöntem.....	28
5. BULGULAR	38
5.1. Resimlerin Deneysel Sonuçları.....	39
6. SONUÇLAR, TARTIŞMA ve ÖNERİLER	55
7. KAYNAKÇA	58

TABLÖLAR LİSTESİ

Tablo 1.1 Aktivasyon Fonksiyonları	9
Tablo 5.1 Ayçiçeği Veriler Tablosu	39
Tablo 5.2 Çançiçeği Veriler Tablosu	41
Tablo 5.3 Çiğdem Veriler Tablosu	42
Tablo 5.4 İris Çiçeği Veriler Tablosu	44
Tablo 5.5 Karahindiba Veriler Tablosu	46
Tablo 5.6 Kardelen Veriler Tablosu	47
Tablo 5.7 Papatya Veriler Tablosu	49
Tablo 5.8 Rüzgâr Gülü Veriler Tablosu	51



ŞEKİLLER LİSTESİ

Şekil 1.1 Yapay Zeka, Makine Öğrenmesi ve Derin Öğrenmenin Tarihsel Süreci	5
Şekil 1.2 Doğal ve Yapay Sinir Ağ Modelleri	6
Şekil 1.3 Yıllara Göre Nesne Tanıma Yüzdeleri	10
Şekil 1.4 Örnek Bölge Önerisi	11
Şekil 2.1 R-CNN Yapay Sinir Ağı	14
Şekil 2.2 Fast R-CNN Yapay Sinir Ağı	15
Şekil 2.3 R-CNN ile Fast R-CNN Karşılaştırma Grafiği	15
Şekil 2.4 Faster R-CNN Yapay Sinir Ağı	16
Şekil 2.5 Yapay Sinir Ağlarının Karşılaştırılması	16
Şekil 2.6 SSD Yapay Sinir Ağı	17
Şekil 4.1 SSD Yapay Sinir Ağı Örneği	25
Şekil 4.2 Örnek Çiçek Resimleri	27
Şekil 4.3 Tensorflow ve Diğer Kütüphaneler	30
Şekil 4.4 Eğitim Süreci Akış Şeması	31
Şekil 4.5 Platformların Benimsenme Grafiği	33
Şekil 4.6 Platformların Yıllara Göre Kıyaslanması	34
Şekil 4.7 Eğitim Süreci Akış Şeması	36
Şekil 5.1 Nesne Kayıp Grafiği	38
Şekil 5.2 Konum Kayıp Grafiği	38
Şekil 5.3 Ayçiçek-1 Tahmin Yüzdeleri	39
Şekil 5.4 Ayçiçek-2 Tahmin Yüzdeleri	40
Şekil 5.5 Ayçiçek-3 Tahmin Yüzdeleri	40
Şekil 5.6 Çan Çiçeği-1 Tahmin Yüzdeleri	41
Şekil 5.7 Çan Çiçeği-2 Tahmin Yüzdeleri	41
Şekil 5.8 Çan Çiçeği-3 Tahmin Yüzdeleri	42
Şekil 5.9 Çiğdem-1 Tahmin Yüzdeleri	43
Şekil 5.10 Çiğdem-2 Tahmin Yüzdeleri	43
Şekil 5.11 Çiğdem-3 Tahmin Yüzdeleri	43
Şekil 5.12 İris Çiçeği -1 Tahmin Yüzdeleri	44
Şekil 5.13 İris Çiçeği -2 Tahmin Yüzdeleri	45
Şekil 5.14 İris Çiçeği -3 Tahmin Yüzdeleri	45

Şekil 5.15 Karahindiba -1 Tahmin Yüzdeleri	46
Şekil 5.16 Karahindiba -2 Tahmin Yüzdeleri	46
Şekil 5.17 Karahindiba -3 Tahmin Yüzdeleri	47
Şekil 5.18 Kardelen -1 Tahmin Yüzdeleri	48
Şekil 5.19 Kardelen -2 Tahmin Yüzdeleri	48
Şekil 5.20 Kardelen -3 Tahmin Yüzdeleri	48
Şekil 5.21 Papatya -1 Tahmin Yüzdeleri	49
Şekil 5.22 Papatya -2 Tahmin Yüzdeleri	50
Şekil 5.23 Papatya -3 Tahmin Yüzdeleri	50
Şekil 5.24 Rüzgâr Gülü -1 Tahmin Yüzdeleri	51
Şekil 5.25 Rüzgâr Gülü -2 Tahmin Yüzdeleri	52
Şekil 5.26 Rüzgâr Gülü -3 Tahmin Yüzdeleri	52

SİMGELER VE KISALTMALAR DİZİNİ

<u>Kisaltmalar</u>	<u>Acıklama</u>
API	Application Programming Interface
ASCII	American Standard Code for Information Interchange
AutoML	Automated Machine Learning
AWS	Amazon Web Servisleri
Bmp	Bitmap
CNN	Convolution Neural Network
CPU	Central Processing Unit
CSV	Comma Separated Values
Faster RCNN	Faster Region Convolution Neural Network
GB	Gigabyte
Ghz	Gigaheartz
GPU	Graphical Processing Unit
Gif	Graphics Interchange Formatın
IDC	Internet Database Connector
JPEG	Joint Photographic Experts Group
MB	Megabyte
OpenCV	Open Source Computer Vision
RAM	Ramdom Access Memory (Rastgele Eriřimli Bellek)
RoI	Reigon of Interest (İlgi Alanı)
SSD	Solid State Disk
SVM	Support Vector Machine
Url	Uniform Resource Locator
YSA	Yapay Sinir Ağları
XML	Extensible Markup Language

GİRİŞ

İnsan; düşünen bir varlık olarak diğer canlılardan ayrılır. İnsanın düşünmesindeki araçları ise zekâ ve akıldır. Bütün insanlar, farklı oranlarda da olsa zekâ potansiyeline sahiptir. Akıl ise bu zekânın kullanımıyla ilgilidir. Konunun daha iyi anlaşılması için şeker ve suyun potansiyel olarak bulunması zekâyâ, bunun karıştırılıp şerbet haline getirilmesi ise akla benzetilebilir.

Son yıllarda ortaya çıkan birçok “akıllı” cihazın önemi buradan kaynaklanmaktadır. Birçok cihaz birçok veriyi alabilme, depolayabilme potansiyeline sahiptir. Ancak bu verilerden anlamlı bir sonuca varılmasının yolu bu tip cihazları “akıllı” hale getirmektedir.

“Akıllı” cihazların temelinde verilerin yorumlanması yatmaktadır. Ancak çok basit akıl yürütme işlemleri günümüzde önemini yitirmek üzeredir. Bugün daha karışık bilgiler arasından daha hızlı ve daha doğru kestirimler yapabilen cihazlar, yapılar, platformlar ön plana çıkmaktadır.

“Big data” diye tabir edilen büyük verilerin işlenmesi söz konusu olduğunda yapay zekâ kavramı devreye girmektedir. Zekâ, özünde insana ait bir kavramdır ve zekâ Türk Dil Kurumu tarafından “İnsanın düşünme, akıl yürütme, objektif gerçekleri algılama, yargılama ve sonuç çıkarma yeteneklerinin tamamı, anlak, dirayet, zeyreklik, feraset” olarak tarif edilmektedir (Türk Dil Kurumu, 2020). Tanımda, zekânın akıl yürütme işlevinden bahsedilmiştir. Burada kastedilen zekâ potansiyelinin açığa çıkarılmasıdır. “Zekî” bir insanın, zekâsını kullanarak, akıl yürüterek çıkardığı sonuçların makineler eliyle yapılması, yapay zekânın ilgi sahasıdır.

Akıl yürütme, insanın bilgi, beceri ve tecrübelerinden hareketle bir şeyle ilgili yaptığı çıkarım, kestirim, yorum ve sonuçtur. Akıl yürüten insan; algılar, hafızaya alır, anlar, öğrenir, sonuç çıkarır, karar verir, düşünce üretir, öneride bulunur, eyleme geçer. Bu haliyle insanın yaptığı akıl yürütmeler, insanı kanunen, hukuken, örfen, ahlaken, siyaseten, ekonomik olarak veya bir başka şekilde sınırlar, bağlar. Tıpkı insanı bağlayan akıl yürütmelerde olduğu gibi yapay zekânın akıl yürütmeler de birçok açıdan birçok şeyi etkilediği yapay zekânın hukuk, etik, ahlak ve kanun veya bir başka şey

karşısındaki konum ve durumlarıyla ilgili birçok tartışma söz konusudur. Ancak bu tartışmalar yapay zekânın hayatımızda kullanılmadığı anlamına gelmemektedir.





BİRİNCİ BÖLÜM

YAPAY ZEKA

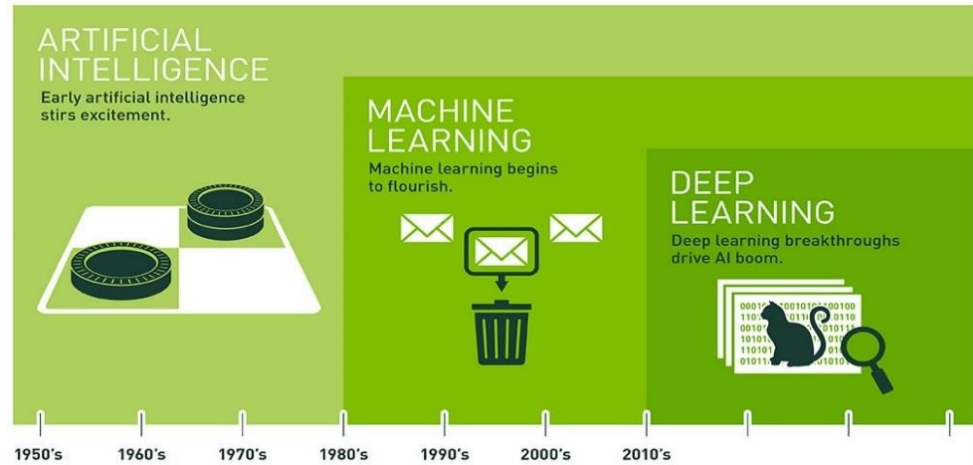
1.1. Yapay Zeka ve Gelecek

Dünyada yapay sinir ağlarıyla öğrenme sağlayan birçok platform ve ekip bulunmaktadır. Birçok şirket de yeni oyuncu olarak piyasaya giriş yapmaktadır. ‘Geleceği analiz eden’ araştırma ve danışmanlık şirketi Internet Database Connector (IDC), yapay zekaya yapılan küresel harcamaların, 2019’da yüzde 44 artışla 35,8 milyar dolara, 2022’de 79,2 milyar dolara ve 2023’te 98 milyar dolara ulaşacağını öngörmektedir (Internet Database Connector (IDC), 2019). Bu tahminlerden, gelecekle ilgili projeksiyonu olan her birey ve kurumun yapay zekâya dikkat etmek zorunda olduğu anlamak güç değildir.

1.2. Yapay Zeka Alanları

Yapay zeka (Artificial Intelligence), genel olarak doğal zekanın yapabileceği görevleri üstlenebilecek mekanizmalara verilen addır. İnsan aklını taklit üzerine kurulan yapay zeka, insanlarda zeka ile ilgili zihinsel fonksiyonları bilgisayar modelleri yardımıyla inceleyip formül haline getirmeyi ve bunları yapay sistemlere uygulamayı amaçlayan araştırma alanıdır. Yapay zeka, insanın düşünme yapısını anlamak ve bunun benzerini ortaya çıkaracak bilgisayar işlemlerini geliştirmeye çalışmak olarak tanımlanır. Daha geniş bir tanıma göre ise, yapay zeka; bilgi edinme, algılama, görme, düşünme ve karar verme gibi insan zekasına özgü kapasitelerle donatılmış bilgisayarlardır. Yapay zekanın uygulanma biçimleri açısından değerlendirildiğinde beş başlık altında incelenir: Bulanık mantık, yapay sinir ağları, genetik algoritma, uzman sistemleri ve karınca algoritmaları (Tülay Bozüyük, Caner Yağcı, İlhan Gökçe, Görkem Akar, 2005).

Yapay zeka ile ilgili farklı sınıflandırmalar mevcuttur. 1950’li yıllarda yapay zeka, etkileşimin arkasında insan zekası olup olmadığını ayırt etmeye göre tanımlanmışken, 1980’li yıllarda programlanmadığı halde sonuçlar çıkarabilen makine öğrenmesi devreye girmiştir. 2010 yılından itibaren verilerin çoğalmasıyla birlikte daha da karmaşık hale gelen veri ve durumların yordanması için derin öğrenme kavramı devreye girmiştir. Bu çalışmada; yapay zeka, makine öğrenmesi ve derin öğrenme arasındaki ilişki şöyle gösterilmiştir (Copeland, 2020):



Şekil 1.1 : Yapay Zeka, Makine Öğrenmesi ve Derin Öğrenmenin Tarihsel Süreci (Şener, 2020).

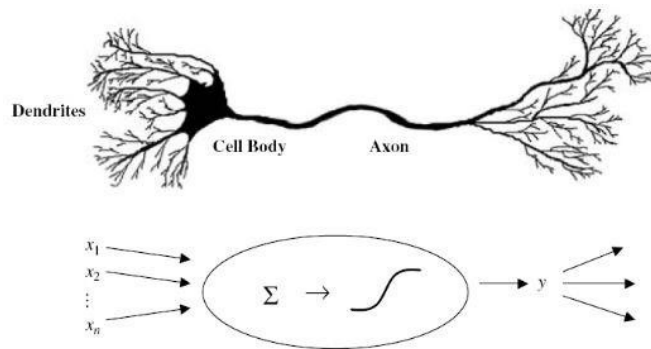
Bir başka araştırmacı Mijwel'e göre yapay zekâ; birçok işlemi ve çeşitli uygulamaları içeren bir isim ve temelde ikiye ayrılır: Makine öğrenimi ve derin öğrenme. Makine öğrenimi, örnekler ve verilerden öğrenilen algoritmaları birleştirir. Örnek olarak sunulan veri kümelerindeki değerleri tahmin etmeyi mümkün kılar. Sonuçların kalitesi, öğrenme sistemine verilen verilerin kalitesine bağlıdır. Derin öğrenme ise teknolojinin ham verilerden öğrenme kabiliyetine dayanır (Mijwel,2016).

Makine öğrenmesi yapılandırılmış verilerle sonuç üretir. Derin öğrenme ise yapay sinir ağlarıyla insandaki sinir ağlarını taklit ederek her bir veri için farklı çıkarımda bulunur: Makine öğrenmesi, analitik model inşasını otomatikleştiren bir veri analizi yöntemidir. Sistemlerin verilerden öğrenebileceği, kalıpları tanımlayabileceği ve en az insan müdahalesiyle kararlar alabileceği fikrine dayanan bir yapay zekâ dalıdır. Klasik programlamadaki gibi tüm kurallar programda belirtilmez. Makine öğrenmesi algoritmaları kuralları verilerden öğrenirler. Klasik istatistik yöntemlerinin yetersiz kaldığı büyük verilerde iyi sonuçlar vermektedir. Derin öğrenme, özel bir makine öğrenmesi çeşididir. Makine öğrenmesi, yapılandırılmış verilerle kendini geliştirerek istenen sonucu verebilmek için insan müdahalesi olmadan kendini değiştirebilen algoritmalarıdır. Derin öğrenme, her bir aldığı veriye farklı bir çıkarım yapar, kullandığı algoritma ağı yapay sinir ağıdır. Yapay sinir ağı, insan beynindeki sinirlerden esinlenmiştir, beyinde mevcut olarak bulunan sinir ağlarını taklit ederek öğrenmeye çalışırlar (Murat Altun, Kemal Kayıkçı, Sezgin Irmak, 2019, s. 338, 339).

1.3. Yapay Sinir Ağları (YSA)

Yapay sinir ağları, öğrenme yolu ile yeni bilgiler üretebilme, yeni bilgiler keşfetme gibi işleri dışarıdan destek almadan otomatik olarak yapabilme amacıyla geliştirilen bilgisayar sistemleridir. Yapay sinir ağları, olayın örneklerinden yaptıkları genelleme ve çıkarımlarla, her girdi setine karşı bir çıktı seti üretebilir. Bu bağlamda yapay sinir ağları, biyolojik beyin fonksiyonlarının ürünü olan örnekleri kullanarak hem öğrenebilen hem de yeni çevresel koşullara vereceği tepkileri belirleyen bilgisayar sistemleridir; kendi deneyimlerini oluşturma ve karar verme kapasitesine sahiptirler. Yapay sinir ağları, birbirlerine bağlı ve paralel olarak çalışabilen yapay hücrelerden oluşmaktadır. Her yapay hücre bağlantısı bir ağırlık değerine sahiptir ve öğrenme yolu ile elde edilen bilgiler bu bağlantı değerlerinde saklanarak bir hafıza oluşturmaktadır (Öztemel, 2006, s. 29, 30).

YSA, insan beyninin çalışma mekanizmasını taklit ederek beynin öğrenme, hatırlama genelleme yapma yolu ile yeni bilgiler türetebilme gibi temel işlevlerini gerçekleştirmek üzere geliştirilen mantıksal yazılımlardır. YSA biyolojik sinir ağlarını taklit eden sentetik yapılardır (Uslu, 2015). Yapay sinir ağları kavramı biyolojik modellere dayandığı için, önce bir sinir hücresi olan bir doğal sinir hücresinin yapısına ve çalışma şekline genel bir bakış getirilmelidir. Nöron, sinir sisteminin temel birimi olup, uyarıları emme ve aktarma görevi görür. Hem çevrenin algılanmasına hem de canlı bir varlık içinde iletişimin yapılmasına izin verirler. Nöronlar duyu organlarda, omurilikte ve beyinde bulunur ve insan beyinde en az 14 milyar nöron bulunur (Mijwel, 2017, s. 1).



Şekil 1.2 : Doğal ve Yapay Sinir Ağ Modelleri (Uslu, 2015).

1.4. Aktivasyon Fonksiyonları

Yapay sinir ağları eğitilirken tıpkı biyolojik sinir ağlarındaki yapılar dikkate alınarak hesaplamalar yapılır. Yapay sinir ağlarında sıkça kullanılan başlıca aktivasyon fonksiyonlarına bakmak gerekir. Yapay sinir ağları eğitirken aktivasyon kodları sık sık kullanılır. Aktivasyon fonksiyonların amacı; weight ve bias değerlerini ayarlamaktır. “ $y=x.w + b$ ” işlemini yaptıktan sonra bu değeri aktivasyon fonksiyonundan geçirilir. X değeri bağımsız değişken girdiler, y ise x değerini bağlı bağımlı değişken çıktıları verir. w ağırlık parametresi ağırlığı, b ise bias değerini temsil eder x’in y’ye yakınlığını ifade eder. Değer sıfıra ne kadar yakınsa x, y’ye o kadar yakın demektir. Fonksiyonlar tensörlere etki eden doğrusal olmayan işlemlerdir. Bunlar nöronların tetiklenmesini olanak sağlayacaktır. Burada yüksek bir değer geldiğinde fonksiyonunda yüksek bir sinyal vermesini sağlamayı amaçlamaktadır.

1.4.1. Sigmoid Fonksiyonu

Aktivasyon kodları arasında ilk kullanılan fonksiyonlardan biridir. Aldığı değerleri 0 ile 1 arasına sıkıştırır ve yüksek değer geldiğinde bire yakın olacak biri geçen ve sıfırdan küçük bir değeri olmayacaktır. Yüksek değerlerle bire yakın olacak ve daha büyük sinyaller oluşturacaktır.

1.4.2. Tanh Fonksiyonu

Sigmoid fonksiyonuna göre daha çok değer alabilen bir fonksiyondur. Daha hızlı öğrenmeyi sağlar ve sınıflama için daha geniş aralığa sahip olduğundan verimliliği artırır. Bu fonksiyon, gelen değerleri -1 ile 1 arasına sıkıştırır. Uçlardaki gradyanların ölmesi problemi, bu fonksiyonun bazen tercih edilirliliğini düşürür.

1.4.3. ReLU Fonksiyonu

Günümüzde çok kullanılan fonksiyonlardandır. Bu fonksiyon gelen değerlere pozitif ya da negatif olarak bakar. Eğer gelen değer negatif ise 0 pozitif ise değeri olduğu gibi gönderir. Diğer iki fonksiyona göre daha etkilidir. Yaklaşık 6 kata kadar daha fazla hızda hesaplama yapabilir. Bu da çok fazla eğitimin olduğu modeller için çok büyük bir sayıya tekabül eder. ReLU fonksiyonu biyolojik nöronlara daha benzer yapıda çalışır. Biyolojik nöronlarda belirli bir seviyenin altında gelen sinyallere tepki vermez. Bir örnek vermek gerekirse, elimize batırdığımız bir iğnenin şiddetinin az

olması sinyalin beyne gitmesini engeller ve parmağımızda acı hissetmeyiz. Buna biyolojide eşik değer denir. Gelen sinyal eşik değerini geçmezse vücut tepki vermez. Bu fonksiyon da tıpkı bu biyolojik eşikte olduğu gibi belli bir değerden sonra tepki verir. Bu fonksiyonun dezavantajı ise, 0 değerinde gelen sinyallere ne yapacağının belirsiz olması ve bazı sinyallerin işlenememesidir (ölü nöron).

1.4.4. Leaky ReLU Fonksiyonu

ReLU fonksiyonunun dezavantajlı olduğu durumlar için geliştirilmiş bir fonksiyondur. ReLU fonksiyonundan farklı olarak negatif değerlere 0.01 gibi küçük bir değeri döndürür ve ölü nöron oluşmasına engel olur. Bu da fazladan işlem yaptırdığından ReLU fonksiyonuna göre işlemcinin hızını yavaşlatır ve öğrenmeyi geciktirir.

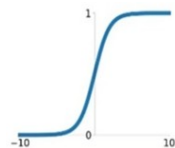
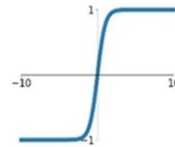
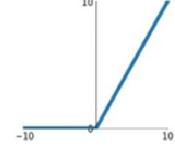
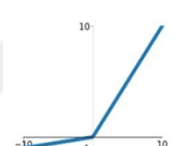
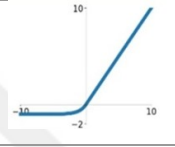
1.4.5. ELU Fonksiyonu

ReLU fonksiyonunun getirdiği bütün avantajlar mevcuttur. Orta noktası 0 olduğundan ölü nöronlar oluşmaz. Sıfırda ve pozitif değerlerde kendi değerini, sıfırdan $\alpha (e^x - 1)$ küçük değerlerde fonksiyonunu çalıştırır.

1.4.6. Maxout Fonksiyonu

Bu fonksiyon, ReLU fonksiyonu ve Leaky ReLU fonksiyonunu genelleştirir. Fonksiyon her iki fonksiyonu çalıştırdığı için ReLU fonksiyonuna göre çok daha yavaş olacaktır.

Tablo 1.1 : Aktivasyon Fonksiyonları.

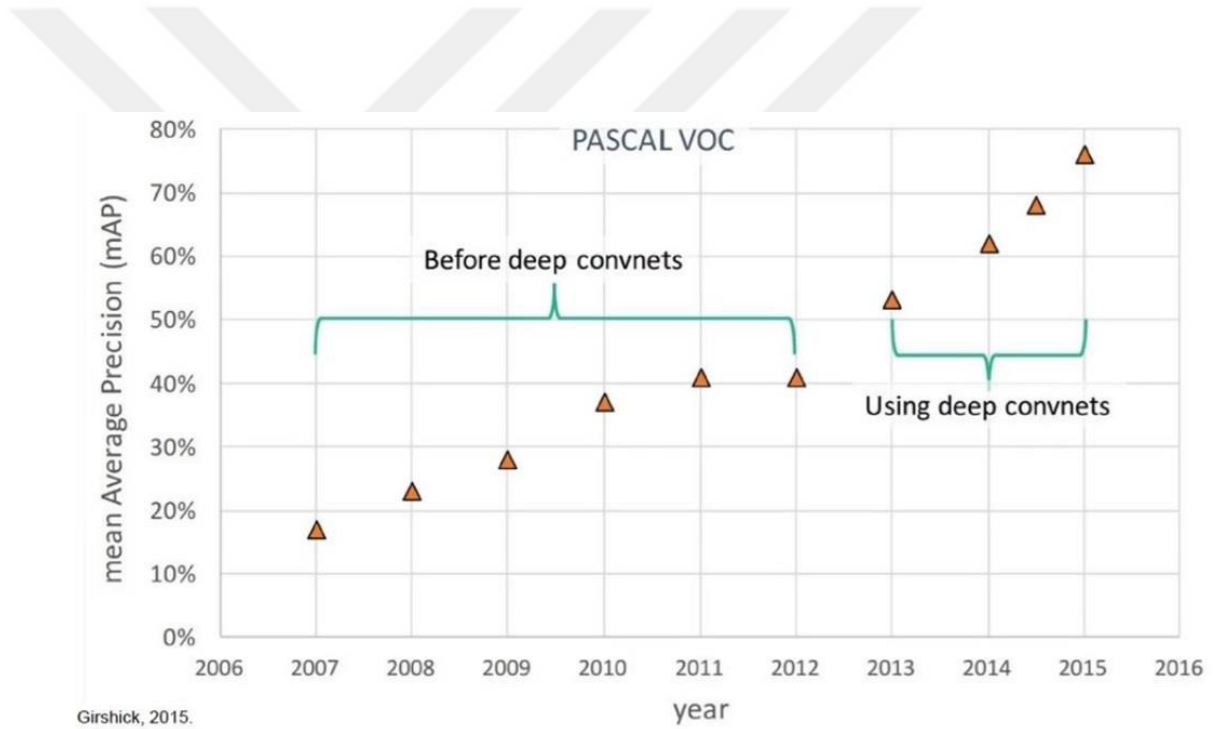
Fonksiyon	Matematiksel İfadesi	Grafiği
Sigmoid	$\sigma(x) = \frac{1}{1 + e^{-x}}$	
Tanh	$\tanh(x)$	
ReLU	$\max(0, x)$	
Leaky ReLU	$\max(0.1x, x)$	
ELU	$\begin{cases} x & x \geq 0 \\ \alpha(e^x - 1) & x < 0 \end{cases}$	
Maxout	$\max(w_1^T x + b_1, w_2^T x + b_2)$	

1.5. Nesne Tanıma

Nesne tanımda, bazı kategoriler temel alınarak nesne tanıma işlemi yapılır. Bir görüntüde belirlenen kategorilerdeki nesnelerin varlığı belirlenmeye çalışılır. İşlem temelde, görüntüde bir ev aranıyorsa o evi bularak sınırlarını belirlemesini işlemidir. Görüntüde ayrıca evin dışında başka nesneler varsa onların da belirtilmesi gerekmektedir. Hatta bir görüntüde birden fazla ev varsa makine öğrenmesinin bütün evleri göstermesi beklenir. Kısaca öğrenme, bütün nesnelerin yerlerini tespit etmelidir. Diğer sınıflandırma modellerinden farklı olarak görüntüde insanı da tanımaya çalışılıyorsa birden fazla kategori aynı görüntü içerisinde bulunabilir. Bir resimde hem ev hem de insanı aranıyorsa yapay sinir ağına hem ev hem de insan resimlerini ayrı ayrı öğretilmelidir. Nesne tanıma resim üzerinde çerçeveler çizerek ve bu çerçeveleri konvansiyonel sinir ağında işleme tabi tutarak bu çerçeve içerisinde aradığı nesne bir ev veya insan ise ev ya da insan olup olmadığını kontrol etmektedir. Eğer çerçevede ev ya

da insan bulamazsa arka plan sınıfı diye bir sınıf oluşturup orayı arka plan sınıfı olarak kaydeder. Bu şekilde çerçeveyi bütün resim üzerinde gezdirerek arama işlemi yapılır. Model iyi eğitildiği ise çerçeveler içerisinde bulduğu ev ya da insan görüntülerini belirleyip çerçeveler işaretlenir.

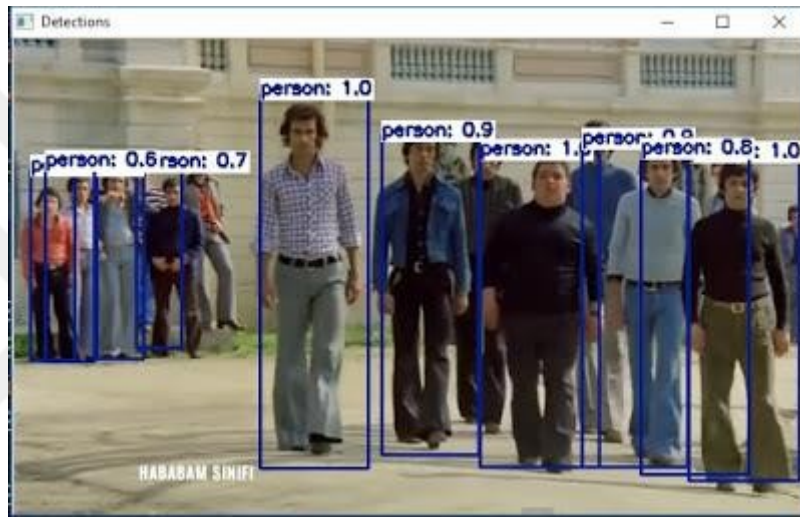
Görüntü içinden nesne tanıma yüzdeleri gün geçtikçe artmaktadır. Yapay sinir ağları kullanılarak nesne tanımda ilk yıllarda neredeyse gelişme olmamıştır. Ancak Alexnet birlikte konvansiyonel sinir ağlarının yaygınlaştığı ve nesne tanıma isabet oranlarında hızlı bir artış olduğu gözlenmiştir. Şekil 1.3'teki bilgi, günümüz modellerinde %90 ve üzeri skorlara ulaştığını göstermektedir. Yıllara göre nesne tanıma yüzdeleri kısaca şöyledir:



Şekil 1.3 : Yıllara Göre Nesne Tanıma Yüzdeleri (Alake, 2020).

Nesne tanımda, bölgenin belirlenmesi önemlidir. Modelden istenen farklı boyutlarda çerçeveler çizip görüntü üzerinde belirli aralıklarla gezdirilmesi ve elde ettiği verileri yapay sinir ağından geçirerek nesne tanıma işlemi yapılması beklenmektedir. Nesnelerin görüntü içerisinde büyük ya da küçük olması durumu göz önüne alındığında hem büyük çerçeveler hem küçük çerçevelerle arama yapılması doğru karar vermesi açısından önemlidir. Konvansiyonel sinir ağlarında milyonlarca işlem gerektirir. Her

çerçeve konvansiyonel sinir ağından geçirilerek değerlendirileceği için işlemler kolay işlemler olmayacaktır. Burada bir nesne için bile görüntü tanıma yapmak için çok güçlü bilgisayarlara ihtiyaç duyulacağı gerçeği göz ardı edilemez. Bu olay imkânsıza yakın bir durumdur. Görüntü bazı bölgelere ayrılarak bu bölgeler içerisinde arama yaparak daha az işlemle nesne tanıma yapılabilir. Görüntüde bazı renkler bir arada bulunması göz önüne alınarak bölgelerde nesne bulunma olasılığı fazla olan bölge sayısı 2000-2500 civarına indirilir. Böylece çerçevenin tarayacağı alan sayısı daha aza indirilmiş olacaktır.



Şekil 1.4 : Örnek Bölge Önerisi.

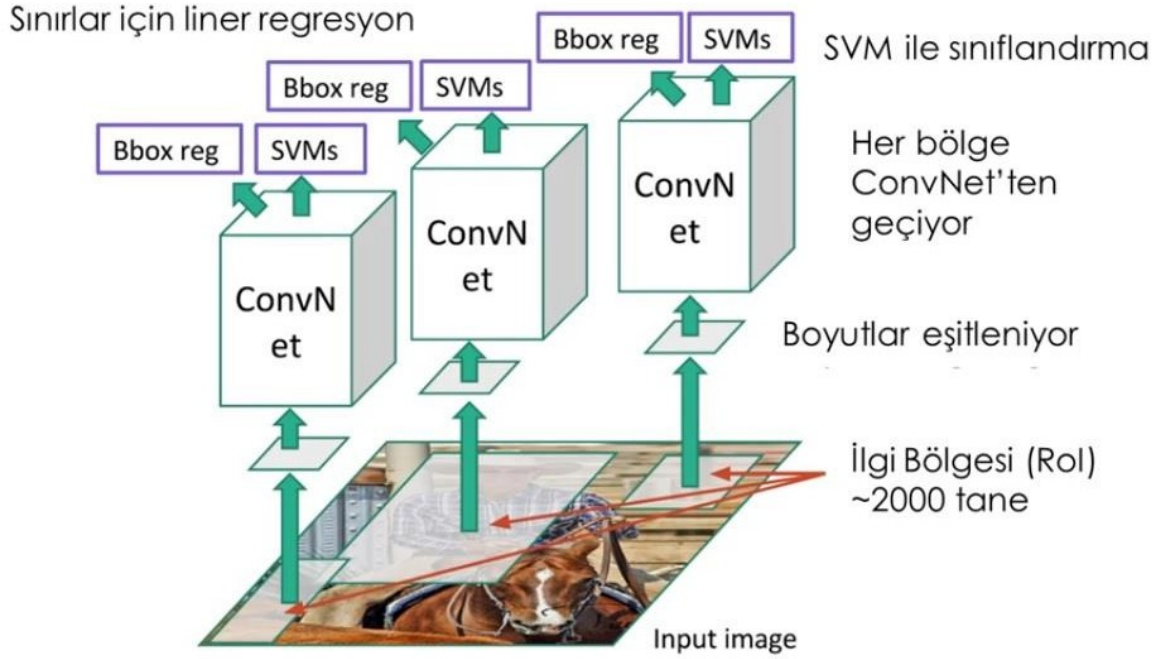
Görüntü içerisinde nesne tanımda en sık kullanılan modellere bakıldığında, R-CNN, Faster R-CNN, Fast, Faster ve SSD modelleri ön plana çıkmaktadır.



İKİNCİ BÖLÜM
EVİRİŞİMLİ YAPAY SİNİR AĞLARI (CNN)

2.1. R-CNN Yapay Sinir Ağı

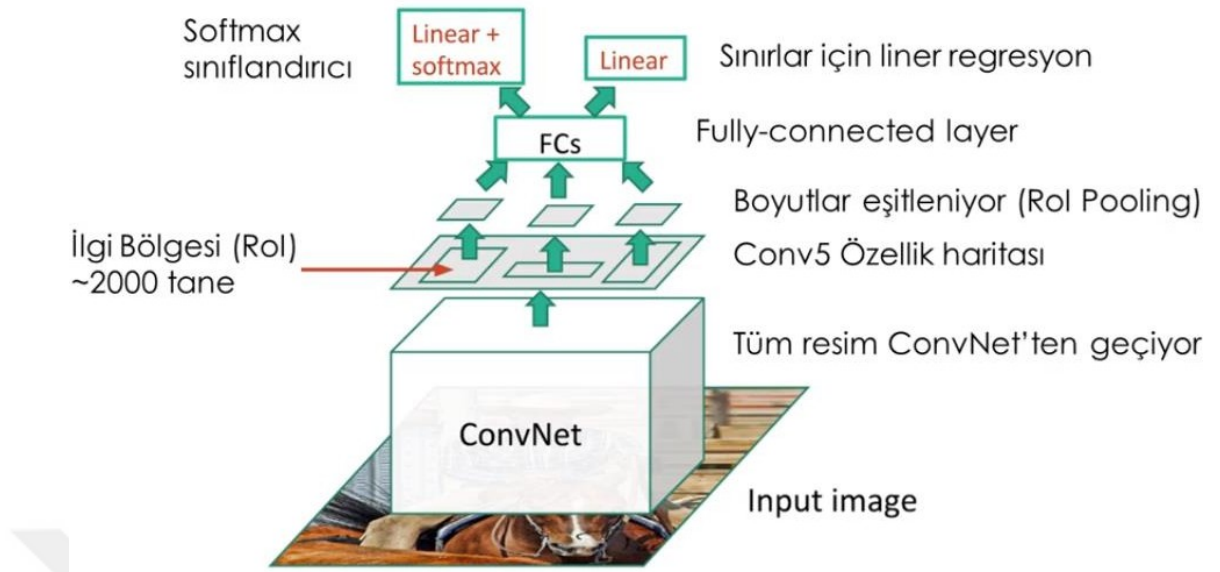
Makine öğrenmesi, görüntüde bölgeyi tanımladıktan sonra nesneyi evrişimli sinir ağına gönderir. Görüntünün birçok bölgesinde çerçeveler oluşturulacak nesne arama işlemi çok zor olacağından bölge önerisiyle belirlenen 1000-2000 civarında bölgede arama yapma işlemi daha kolay olacaktır (Şeker, 2008). Belirlenen bölgelere ilgi bölgesi (ROI) denilmektedir. Bu bölgelerin farklı farklı boyutları da olacaktır. Bu çerçeveleri yapay sinir ağından geçirebilmek için bölgelerin sınır uzunluklarını eşit yapmak gerekir. Sınıflandırma yapan konvansiyonel sinir ağları hep belirli büyüklükte giriş değeri (input) alır. Alınan bu bölgeler, tek tek konvansiyonel sinir ağından geçirilir. Bu bölgede sınıflandırma yapmak için Destek Vektör Makinesi (Support Vector Machine, SVM) kullanılır. SVM sınıflandırma için kullanılan basit ve etkili bir yöntemdir ve bir düzlemde bulunan iki grup arasında sınır çizerek nesneleri gruplandırmayı sağlar (Şeker, 2008). Ayrıca lineer regresyon sistemi ile de nesnenin sınırları belirlenir. Lineer regresyon, bir veya birden fazla bağımsız değişken ile başka bir bağımlı değişken arasındaki bağlantıyı modellemek için kullanılan bir yöntemdir. R-CNN, belirlenen bölgede bir nesne arayarak bulduğu nesnenin sınıfını döndürür. Aynı zamanda nesnenin yerini gösteren bir çerçeve çizer. Çok yavaş çalışan bir sinir ağıdır. Önerilen bölgeler alınarak tek tek konvansiyonel sinir ağından geçirilir. Bu işlemin yaklaşık 2000 tane olduğu göz önüne alındığında görüntü işlemenin neden yavaş olduğu anlaşılır. Sonuç olarak hem eğitim aşaması yavaş hem de tanıma işlemi yavaş olan bir model ortaya konur. Bu da gerçek zamanlı bir nesne tanıma yapma işlemini zorlaştırır.



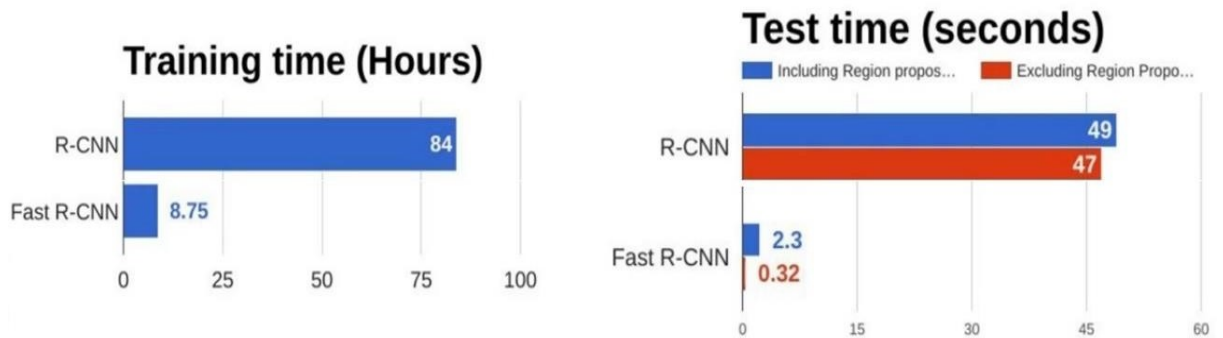
Şekil 2.1 : R-CNN Yapay Sinir Ağı (Sachan, 2018).

2.2. FAST R-CNN Yapay Sinir Ağı

R-CNN in yavaşlığını gidermek için geliştirilmiş bir sinir ağıdır. Faster R-CNN, R-CNN gibi benzer yapıları kullansa da hızlandırmak için farklı teknikler kullanır. Görüntüde bölge önerisi yapmak yerine görüntü konvansiyonel sinir ağından geçirilir. Orijinal görüntüye uyan yüksek çözünürlüklü bir özellik haritası çıkarılır. Bu harita üzerinde yaklaşık 2000 tane ROI oluşturulur. Önerilen bölge sayısı kadar konvansiyonel sinir ağından geçirmek yerine bir defa geçirildiği için R-CNN sistemine göre çok daha hızlı şekilde çalışmaktadır. Daha sonra önerilen bölgelerin boyutları eşitlenmektedir. Bölgeler tam bağlantılı katmandan (fully-connected layer) geçilerek sınıflandırma yapılmaktadır. Daha sonra tespit edilen nesnenin sınırları belirlenir. Fast R-CNN’de sınıflandırma işlemi için softmax katmanı kullanılmaktadır. Şekil 2.3’de görüldüğü üzere, Fast R-CNN, R-CNN’ e göre çok daha hızlıdır. Eğitim sürecinde yaklaşık 9-10 katı kadar daha hızlı olan Fast R-CNN, test sürecinde de yine 24 katı kadar hıza ulaştığı görülmektedir:



Şekil 2.2 : Fast R-CNN Yapay Sinir Ağı (Sachan, 2018).

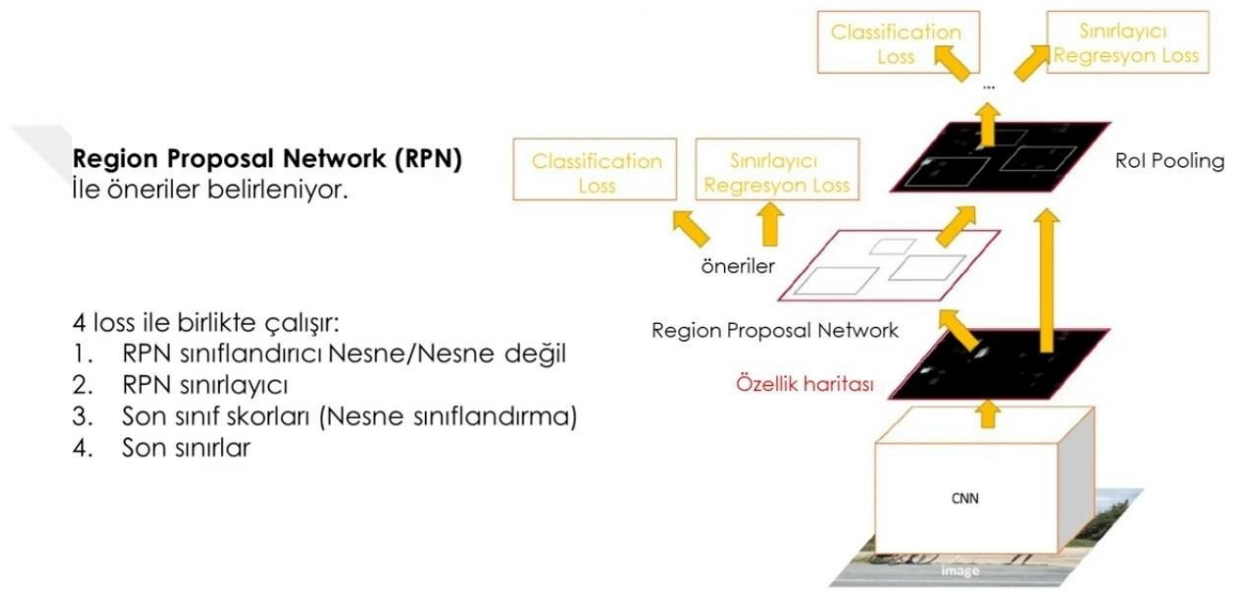


Şekil 2.3 : R-CNN ile Fast R-CNN Karşılaştırma Grafiği (Gandhi, 2018).

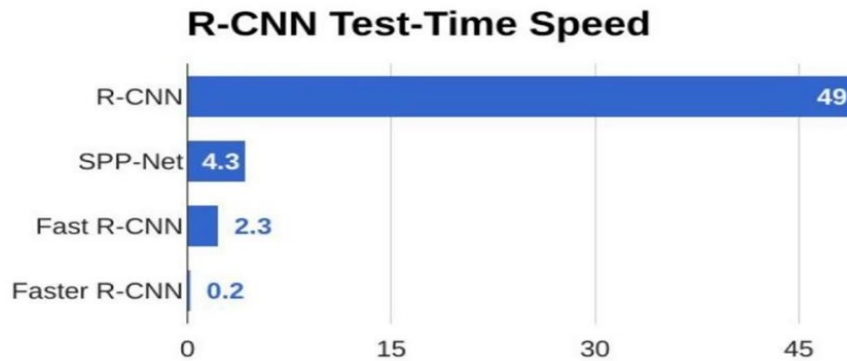
2.3. FASTER R-CNN Yapay Sinir Ağı

Görüntü, bölge önerisi yapmak yerine görüntüyü konvansiyonel sinir ağından geçirir. Orijinal görüntüye uyan yüksek çözünürlüklü bir özellik haritası çıkarılır. Seçici kaynak bulma yöntemi ile bölge önerisi almak yerine ayrı bölge önerisi verecek ayrı bir sinir ağı oluşturulur. Bölge önerileri artık bu ağı üzerinden yapılır. Bu aşamadan sonra önerilen bölgelerin boyutları eşitlenmektedir. Bölgeler tam bağlantılı katmandan geçilerek sınıflandırma yapılır ve tespit edilen nesnenin sınırları belirlenir. Artık hem

bölge önerisi ağını hem de normal konvansiyonel işlem yaptığımız ağ da eğitilmek durumundadır. Bölge önerisinde bulunan ağın görevi çerçeve içerisinde nesne araması yapmak ve çerçeve boyutunu belirlemektir. Asıl ağ ise sınıflandırma yaparak ve baktığı bölge içerisinde nesnenin sınırlarını belirlemektir. Şekil 2.5'deki yapay sinir ağlarının karşılaştırma grafiği incelendiğinde Faster R-CNN diğerlerine göre çok hızlı sonuçlar verdiği görülmektedir.



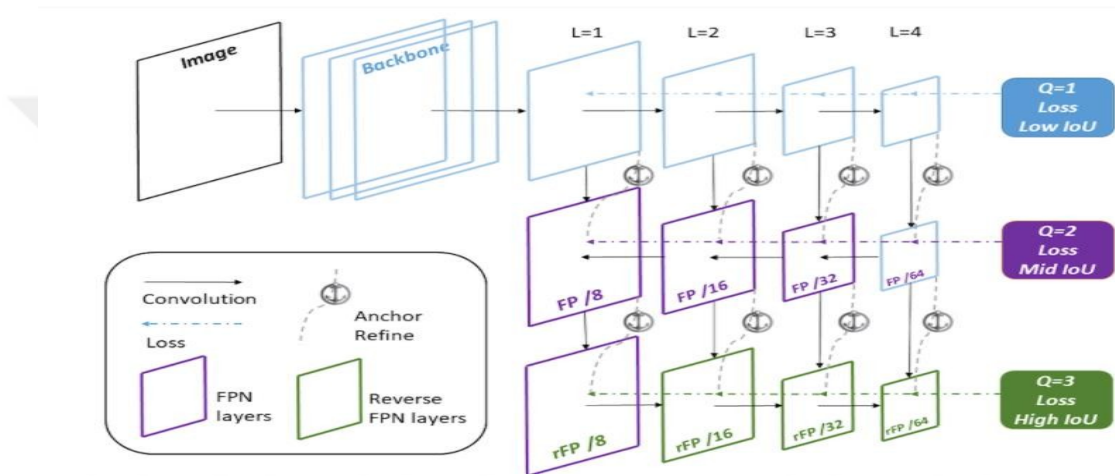
Şekil 2.4 : Faster R-CNN Yapay Sinir Ağı (Gandhi, 2018).



Şekil 2.5 : Yapay Sinir Ağlarının Karşılaştırılması (Gandhi, 2018).

2.4. SSD Yapay Sinir Ağı

Görüntü için bölge önerisi yapmak yerine görüntü konvansiyonel sinir ağından geçirir. Farklı boyutlarda özellik haritaları çıkarır. Bütün resim karelere ayrılıp her kare için hem sınırlar hem de sınıflandırmalar belirlenir. Bu kareler her aktivasyon haritasında olduğu için hem küçük nesneleri hem de büyük nesneleri tanımaya imkân verir. Şekil 2.6'da da SSD yapısının mimarisi verilmiştir:



Şekil 2.6 : SSD Yapay Sinir Ağı (Tekin, 2020).

SSD Yapay Sinir Ağında, eğitim esnasında doğru olan sınırlar ile tahmin edilen sınırlar karşılaştırılır. Yüzde 50 üzerinde tahminde bulunanlara olumlu olarak işaretleme yapılır. Her bölge için farklı işlemler yapmak yerine konvansiyonel sinir ağı içerisinde tek seferde yapılmaktadır. Ağ, giriş (input) olarak bir görüntü alırken çıktı (output) olarak bir tensör vermektedir. Özetle SSD modeli, belirlenen bütün bölgeler için tek tek tahmin yapmak yerine tüm tahminleri bir defa CNN ağından geçirir.

SSD ile Faster R-CNN günümüzde en çok kullanılan sistemlerdir. Daha isabetli tahminler yapmak istenirse Faster R-CNN kullanılması daha uygun olacaktır. Fakat daha hızlı tahminlerde bulunmasını istenen durumlarda SSD modeli daha uygun olacaktır. SSD görüntüyü sadece bir defa sinir ağından geçirdiği için isabeti daha düşük tahminde bulunur. Fakat Faster R-CNN'ye göre daha hızlı veriler verecektir (Cebeci, 2020).

Google Tensorflow görüntü içerisinde nesne tanımada kullanılan R-CNN, Faster R-CNN, Fast, Faster ve SSD modelleri dâhil farklı öğrenme modellerini (SSD Lite, RFCN gibi) içerir. Tensorflow'un dünyada çok yaygın biçimde kullanıldığı bilinmektedir. Tensorflow yayınlandığından bugüne kadar 150 milyon indirmeye ulaştığı kayıtlıdır.





ÜÇÜNCÜ BÖLÜM
NESNE TANIMA VE TEZ

3.1. Tensorflow Nesne Algılama API'si

Tensorflow, hazır olarak Google tarafından eğitilmiş modellere ulaşmayı sağlamaktadır. Model eğitmek günümüz bilgisayarları ile uzun süre alacağından bu kütüphane ile birçok eğitim daha kısa sürecektir. Tensorflow Object Detection API birçok farklı modeli farklı hızlarda ve isabet oranlarıyla paylaşmaktadır. Tensorflow, “https://github.com/tensorflow/models/tree/master/research/object_detection” adresinden Tensorflow Object Detection API hakkında birçok bilgiyi paylaşmaktadır. Kısaca Tensorflow Google'ın daha önceden eğittiği modellerle nesne tespit etme ve isimlendirmemizi sağlayan araçtır. Hazır eğitilmiş modelleri kullanıcıların hizmetine sunan Tensorflow, günlük hayatta sıkça karşılaşılan 90 tane nesne üzerinden eğitilen COCO veri seti ve içerdiği nesneleri barındırır. Her bir model yaklaşık 200 bin resim üzerinden eğitilen görüntülerden oluşmaktadır.

Google'ın paylaştığı veriler, SSD modellerinin Faster R-CNN' ye göre daha hızlı olsa da isabet konusunda daha geride kaldığını göstermektedir. Eğer cep telefonu uygulaması gibi daha düşük çözünürlükte işlemler yapılmak istenirse SSD modellerini, daha isabetli sonuçlar beklendiği durumlarda ise Faster R-CNN kullanmak avantajlı olacaktır (Shi, Rathod, & a-googler, 2020).

3.2. Derin Öğrenme Kullanarak Bitkilerde Nesne Tanıma ile İlgili Çalışmalar

Konvansiyonel sinir ağları kullanılarak çiçek türleri sınıflandırması yapılan bir çalışmada (Ghtairi, 2019), sınıflandırma amacıyla kullanılan derin öğrenme algoritmalarından biri olan CNN, 46 çiçek görüntüsüne uygulanmıştır. Model, renkli ve gri ölçekli olmak üzere iki farklı şemada tasarlanmıştır. Renkli görüntüler göz önüne alındığında, önerilen modelin %84'lük bir sınıflandırma başarı oranına ulaştığı gözlemlenmiştir. Gri tonlamalı görüntülerde ise %47'lik bir başarı oranına ulaşılmıştır. Model, gri tonlamalı görüntüler için daha kısa zamanda değerlendirme işlemi gerçekleştirmekte ancak renkli görüntüler için daha yüksek başarı oranı sağlamaktadır. CNN algoritmasında, konvansiyonel katmanın renkli görüntülerde daha belirgin ve daha fazla özelliği olması nedeniyle, algoritmanın renkli görüntülerde gri tonlamalı görüntülere kıyasla daha başarılı olduğu gözlemlenmiştir. Çalışma sonucunda, çiçek

sınıflandırma ile uğraşan araştırmacılara daha hızlı ve daha başarılı bir sınıflandırma modeli sunulmuştur (Ghtairi, 2019).

Nkemelu ve arkadaşları (Daniel Nkemelu, Daniel Omeiza, Nancy Lubalo, 2018), bitki fidelerinin sınıflandırılmasında destek vektör makinesi ve k-en yakın komşu algoritmaları ile derin öğrenme modellerinden CNN ile çalışmışlardır. 4 bin 275 farklı resimde 12 tür ve yaklaşık 960 fideden oluşan bir veri kümesi kullanmışlardır. Geleneksel algoritmalara göre CNN modelinin daha başarılı olduğunu gözlemlemişlerdir. CNN verimini optimize etmek ve verimliliğini artırmak için önemli bir potansiyelinin olduğunu bulmuşlardır.

Bitkilerin doğal ortamında tanımlamasını amaçlayan bir diğer çalışmada ise, renk, doku ve şekil özniteliklerinin çıkarılması için en iyi yöntemin bulunması amaçlanmıştır. Çalışmada Kuzey Kore coğrafyasında yetişen 300 türe ait 29 bin 463 bitki görüntüsü 2011-2014 yılları arasında toplanmış ve veri kümesi oluşturulmuştur. Öznitelik çıkarma aşamasında, bitkinin morfolojik özellikleri dikkate alınmış ve Watershed algoritması ile 47 bitki görüntüsü eşit parçalara bölünmüştür. Ayrıca, bitki görüntülerinin renk, doku ve şekil öznitelikleri çıkarıldıktan sonra çok katmanlı tanımlama yöntemi ile kombine edilerek sunucuda saklanmıştır. Çalışmada yüzde 91,26 en iyi 1 başarı oranı, yüzde 97,40 en iyi 6 başarı oranı elde edilmiştir (Lee, Hyo-Haeng - Hong, Kwang-Seok, 2017).

Köksal, “Dağ çiçeklerinin derinlemesine ayrıştırılabilir evrişimli sinir ağları modellenmesi” adlı tezinde Doğu Anadolu Bölgesi’nde bulunan 14 bitki türünü derin öğrenme yöntemleriyle sınıflandırmıştır. Veri setindeki görseller döndürmenin sınıflandırma başarısını iki kat artırdığını gözlemlemiştir (Köksal, 2019).

Muhammad ve arkadaşlarının (2018) yaptığı bir çalışmada CNN, Alexnet ve GoogleNet meyve sınıflandırma işlemi yapılmıştır. Üç modelde yüzde 100 tanıma gerçekleştirememiştir. Ancak tanıma yüzdesi ve hızı bakımından diğerlerine göre daha iyi sonuçlar verdiğini bulmuşlardır (Nur Azida Muhammad, Amelina Ab Nasir, Zaidah Ibrahim, Nurbaity Sabri, 2018).

Gurnani ve arkadaşlarının (2017) çalışmalarında farklı çiçeklerin sınıflandırılması için derin bir öğrenme ağı geliştirmişlerdir. Bunun için Oxford Üniversitesi’nden görsel geometri grubunun 102 çiçek kategorisinde 8 bin 189 çiçek verisini kullanmışlardır. Yazarların metotları temel olarak görüntü bölümlendirme ve sınıflandırma olarak iki bölüme ayrılmaktadır. Sınıflandırma amacıyla iki farklı

konvansiyonel sinir ağı mimarisi olan GoogleNet ve AlexNet'i karşılaştırmışlardır. Her iki mimaride de aynı hiper parametreleri koruyarak GoogleNet'in Top-1 ve Top-5 doğruluğunun sırasıyla yüzde 47,15 ve yüzde 69,17 olduğunu, AlexNet'in ise Top-1 ve Top-5'in doğruluğunun yüzde 43,39 ve yüzde 68,68 olduğunu gözlemlemişlerdir. Bu sonuçlar, yüzde 0,98'lik rasgele sınıflandırma doğruluğuyla karşılaştırıldığında oldukça iyidir (Ayesha Gurnani , Viraj Mavani , Vandit Gajjar , Yash Khandhediya, 2017).

3.3. Tezin Amacı

Bu tezde; Google makine öğrenmesi ile yerel makine öğrenmesini kıyaslanarak her iki platformun avantaj ve dezavantajlarını ortaya koymak amaçlanmıştır. Ticari sır, müşteri bilgilerinin güvenliği ve mahremiyeti, veri güvenliği, veri işleme maliyetleri, kamuya açık kaynakların ücretsiz kullanımı, hız, ekip çalışmasına uygunluk, sürekli ve anında iş birliği, pratiklik, kullanışlılık, güncelleme ve üçüncü parti yazılımları açısından her iki platformun artı ve eksi yönleri sonuç bölümünde tartışılacaktır.

3.4. Tezin Önemi

Yapay zekâ; günümüzde bireysel, toplumsal, siyasal, hukuki, ticari vb. açılardan insan hayatını etkilemektedir ve gelecekte bu etki katlanarak büyüyecektir. Karlılık, sürdürülebilirlik, güvenlik, güncellenebilirlik gibi bir ticari işletme için önemli olan birçok hususu da göz önünde bulundurarak bir ticari işletme için Google makine öğrenmesi mi yoksa yerel makine öğrenmesinin mi daha tercih edilebilir olduğunu ortaya koymak, birçok işletmeye yol gösterebilir.

3.5. Tezin Kapsamı

Bu tezde, aynı veriler kullanılarak Google makine öğrenmesi ve yerel makine öğrenmesi kıyaslanmıştır. Her iki platformun öğrenmelerden sonra bulundukları tahminler üzerinden bir kıyaslama yapılmıştır. Tez sonuçları okunurken ticari işletmelerin ihtiyaçlarına göre dikkate alınan ve alınmayan hususlar olacaktır. Bu hususlar araştırma sonuçlarının farklılaşmasına sebep olabileceği göz önünde bulundurulmalıdır.

3.6. Tezin Metodolojisi ve Organizasyonu

Bu tez Giriş, Materyal ve Yöntem, Bulgular ve Sonuç-Tartışma bölümlerinden oluşmaktadır. Giriş bölümünde yapay zekâ, sinir ağları, yapay sinir öğrenmeleri, yerel makine ve Google öğrenmeleri, öğrenme modelleri ve derin öğrenme ile bitki nesnelerini tanımayla ilgili literatür verilmiştir. Çalışmanın Materyal ve Yöntem bölümünde veri setlerinin seçimi ve aşamaları, modellerin eğitilmesi anlatılmıştır. Bulgular bölümünde yerel makine ve Google öğrenme skorları verilmiştir. Sonuç ve tartışma bölümünde ise yerel makine ve Google skorları önce kendi aralarında sonra literatürdeki diğer çalışmalarla kıyaslanmıştır.

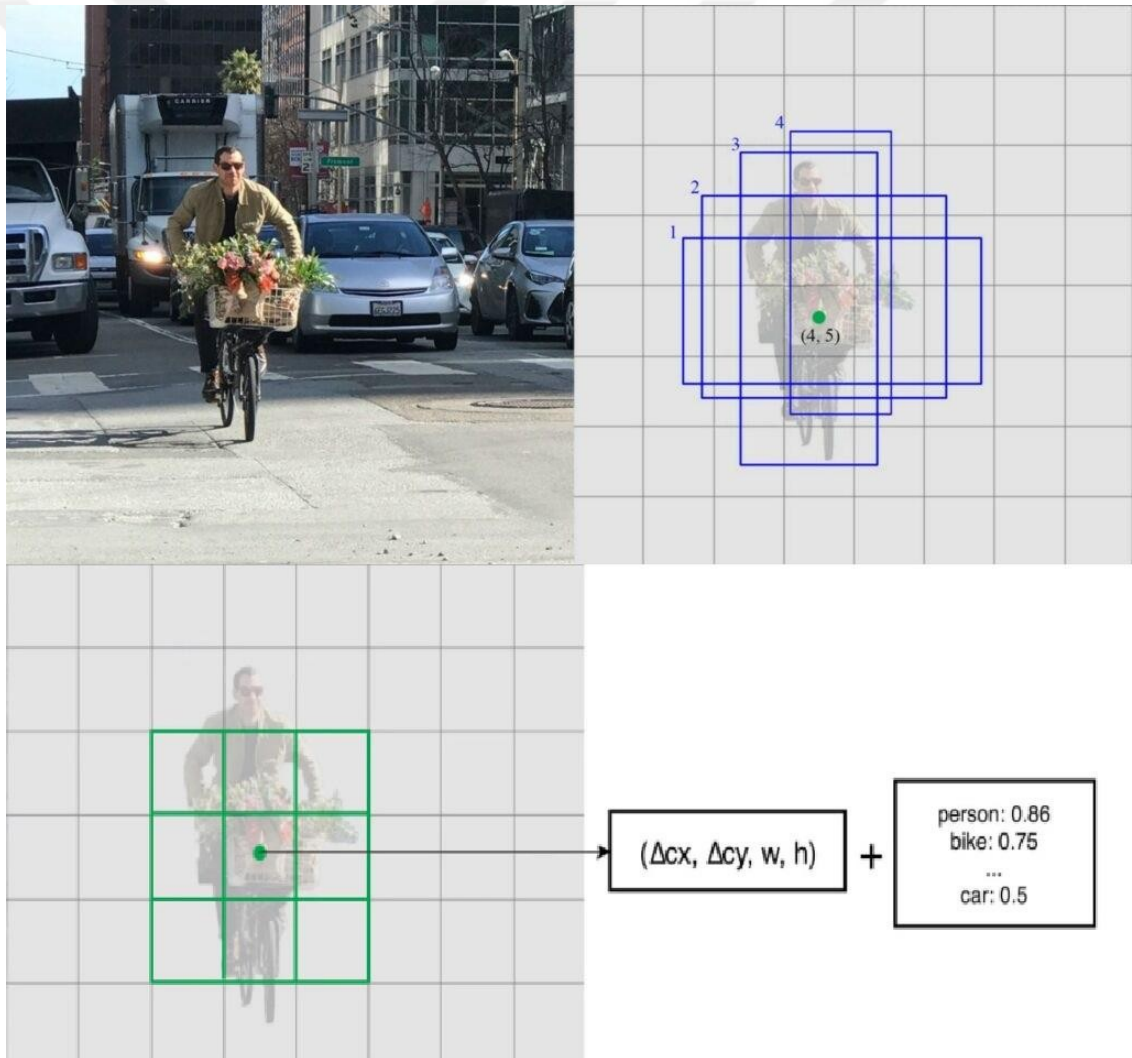




DÖRDÜNCÜ BÖLÜM
MATERYAL VE YÖNTEM

4.1. Materyal

Bu çalışmada, yerel makine eğitiminde Google tarafından geliştirilen Tensorflow kütüphanesi kullanılmıştır. Program için gerekli kodlar Python dilinde yazılmış ve derleyici olarak Python diline özel geliştirilmiş olan Pycharm kullanılmıştır. Daha önceden eğitilmiş COCO kümelerinden “ssd_mobilenet_v1_ppn_coco” kullanılmıştır. Çünkü bu tez çalışması Google Firebase mobil cihazlar için geliştirilmiş olduğundan yerel makinede de mobil cihazlar için uygun olan bu küme tercih edilmiştir. İsaletli veri için Faster R-CNN modelleri, hızlı veri için SSD modelleri daha uygun olduğu için böyle bir tercih yapılmıştır.



Şekil 4.1 : SSD Yapay Sinir Ağı Örneği (Tekin, 2020).

Ssd modelinde Şekil 4.1'deki ızgara yapılarında sınırlayıcı kareler vardır. Bu sayede resimdeki nesnelerin bulunduğu gerçek bölgeleri tahmin etmeye çalışır. Şekil 11'de görüldüğü üzere tahmin etmeye çalıştığı objelere bir yüzde atayarak hangi objelerin daha yüksek tahminle belirlediği görülmektedir. Araba olma ihtimalini %50 ile belirlemiş olsa da insan ve bisiklet tahminleri daha yüksek olduğunu belirlemiştir. Bu bağlamda tezimizde eğitilen çiçek kategorilerinden hangisinin daha yüksek yüzdeyle belirlenmesi daha fazla önem arz etmektedir. Bu nedenle ssd modeli üzerinde bir çalışma yapılmıştır.

Makine öğrenmesi, görüntü işleme, veri işleme, grafik oluşturma işlemleri için özel bir kütüphane olan Anaconda kullanılmıştır.

Yerel makine özellikleri ise şöyledir:

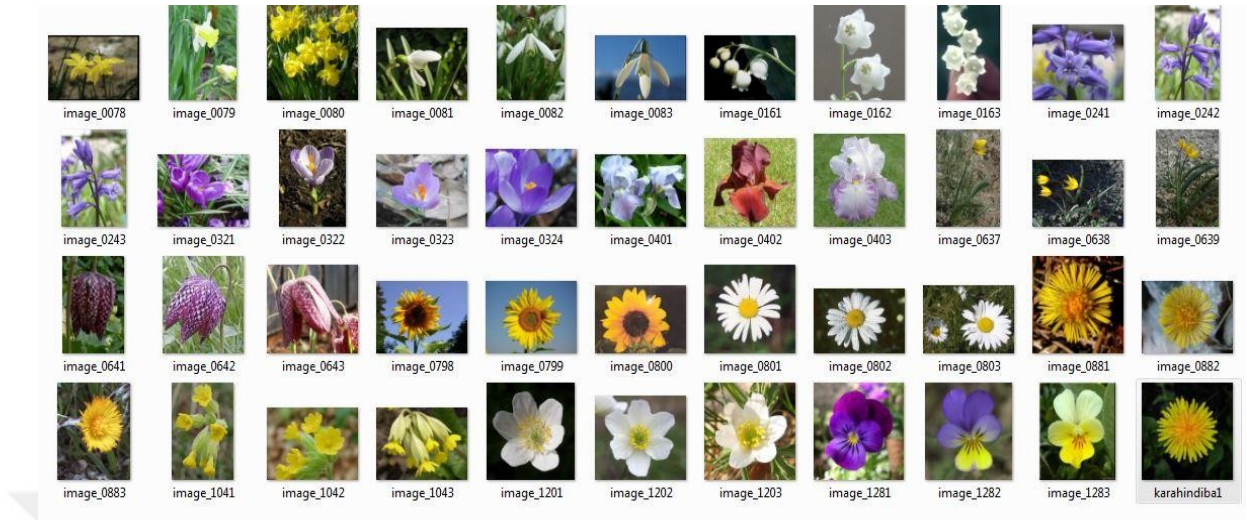
Eğitilen makinenin özellikleri işletim sistemi Windows 7(64 bit, yapı 7601) işletim sistemi, işlemci özellikleri: Intel Xeon (CPU E5-2620 v4 2.10GHz (32 CPUs)), bellek 65534 MB özelliklerine sahiptir. Yerel makine sadece işlemci işe koşularak eğitilmiştir. Yerel makine bu alan için kullanılacak iyi özelliklere sahiptir.

Bu tez çalışmasında yerel eğitilen makinede 8 adet çiçek türü üzerinden bir çalışma gerçekleştirilmiştir. Her tür çiçek için 80 adet birbirinden farklı resimlerle eğitim sağlanmıştır. Google Firebase üzerinden ise yerel makinede eğitilen 8 tane çiçek türünün 80 adet resimleri alınıp yaklaşık 1 saatlik bir eğitim gerçekleştirilmiştir. Google Firebase, spark planı üzerinde gerçekleştirdiği eğitimin süresini kendisi belirlenmiş ve yaklaşık 1 saat süren bir eğitim yapmıştır. Bu eğitimler sonucunda elde edilen modellere eğitimde kullanılmayan ve her çiçek türü için üçer adet çiçek teste tabi tutulmuştur.

4.1.1. Veri Kümesi

Karşılaştırılması yapılacak olan yerel makine ile Google Firebase de modeller oluşturulup test edilmiştir. Bu modeller oluşturulurken; 8 adet çiçek türü üzerinden bir çalışma gerçekleştirilmiştir. Her çiçek türü için 80 adet resim kullanılmıştır. Bu eğitimler sonucunda elde edilen modellere eğitimde kullanılmayan ve her çiçek türü için üçer adet çiçek teste edilmiştir. Eğitim sürecinde ayçiçeği, çan çiçeği, çiğdem çiçeği, düğün çiçeği, iris çiçeği, karahindiba, kardelen, lale, müge çiçeği, nergis, öksürük otu, papatya ve menekşe çiçeği resimlerinden oluşan veriler kullanılmıştır. Resimler

"<https://www.robots.ox.ac.uk/~vgg/data/flowers/> " sitesinden indirilmiştir. Resimlere ait bazı örnekler Şekil 4.2'de verilmiştir.



Şekil 4.2 : Örnek Çiçek Resimleri.

Veri kümesi içerisinde sadece bu 8 türün seçilmesinin nedeni, veri kümesinde birbirine benzer şekillere sahip olan çiçek türleri olması ve eğitilen başka bir çiçek türüne benzemeyen ve kendine mahsus görüntüsü olan çiçekler olmalarıdır. Bu durum eğitilen modelin ayırt etme gücünü test edecektir. Ayrıca bu veri setindeki çiçeklerin boyutu küçüktür. Makine öğrenmesi için verilecek resim birçok yönden, ışığın miktarı ve geliş yönü, varsa farklı renklerinden oluştuğundan makine öğrenmesi yapmak için elverişlidir. Makine öğrenme esnasında veri kümeleri ne kadar çok veriyi makineye sağlarsa model o derece güçlü olacaktır. Kısacası modeli eğitirken kullanılan veriler nasılsa, test aşamasında kullanılacak veriler de benzer veriler olmalıdır. Eğitim verileri bulanık ve düşük çözünürlüklü resimlerden oluşuyor ise bulanık ve düşük çözünürlüklü test verilerinde daha yüksek tahminlerde bulunacaktır.

4.1.2. Labellmg Programı ile Görüntü Etiketleme

Yerel makine de model oluşturma sürecinin en uzun sürecek kısmı resim etiketleme sürecidir. Resimleri etiketlenecek resimlerin yüzde 20'si test için yüzde 80'ini eğitim için kullanılmıştır. Her iki kısımda da birbirine benzeyen resimler kullanılırsa modelin gücünü azalacağından her bir tür için birçok açıdan çekilmiş birbirinden farklı 80 resim kullanılmıştır. Resimleri etiketleme ve xml dosyalarına ayırmak için Labellmg (Tzutalin, 2020) programı kullanılmıştır. Labellmg açık kaynaklı

kodlu ve etiketleme için kullanılan bir programdır. Etiketleme yapılırken Türkçe karakter kullanılmamıştır.

4.1.3. Automl Vision Edge ile Görüntü Etiketleme

AutoML Vision Edge'e veriler veri kümesi şeklinde yüklenir. Yüklenen resimler jpeg, png, gif, bmp, ico biçiminde olmalıdır. Her resim 30 MB veya daha küçük olması gerekir. AutoML Vision Edge ön işleme sırasında birçok resmi küçülterek yükler. Bunun nedeni yüksek çözünürlüğe sahip verilerin doğruluk avantajını yükseltmeyeceğidir ki bu kendi dokümanlarında belirtmektedir (google, 2020). Her etiket için en az 100 resim istemektedir. Ücretsiz (Spark Planı) model oluşturma sırasında en fazla 1000 resim yüklenmesine izin vermektedir. Daha fazla resim için ücretli planları da mevcuttur. Eğitim verilerini yükleme için üç seçenek sunmaktadır:

1. **Yapısal Arşiv:** Resimleri her biri bir etiketin adını taşıyan ve o etikete örnek olan resimleri içeren dizinler halinde düzenlenip dizin yapısını bir zip dosyası halinde sisteme yüklenir. Dizin adları etiket ismi de olacağından 32 ASCII karakter uzunluğunda olmalıdır. Yalnızca alfa sayısal karakterler ve alt çizgi karakteri (_) kullanılmalıdır.
2. **CSV Dizini ile Bulut Depolama:** Egzersiz resimleri Google Cloud Storage'a yüklenir ve her bir resmin URL'sini ve CSV dosyaları şeklinde kullanılabilecektir.
3. **Etiketsiz Görüntüler:** Resimler tek tek ya da toplu halde yüklendikten sonra Firebase konsolunda etiketleme yapılır (Train an image labeling model with AutoML Vision Edge, 2020). Bu tez çalışmasında da etiketsiz olarak konsol aracılığıyla sisteme yüklenmiş ve Google Firebase konsolunda etiketleme işlemi yapılmıştır.

4.2. Yöntem

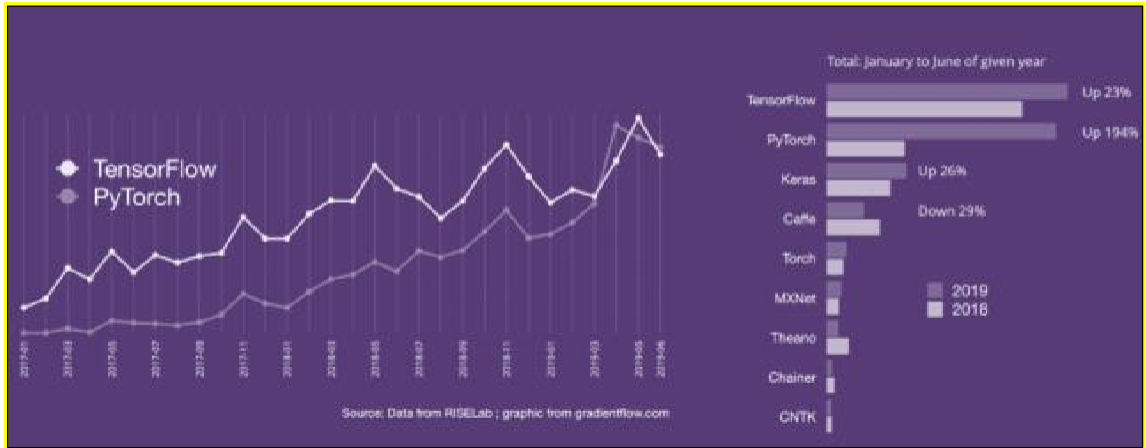
4.2.1. Yerel Makinede Anaconda ve Python Kurulumu

Bu tez çalışmasında modellerin eğitimi ve kullanılması sırasında yerel makinelerde Python kodları kullanılmıştır. Anaconda, bir Python dağıtımıdır.

Anaconda, makine öğrenmesi, görüntü işleme, veri işleme, grafik oluşturma için kullanılacak birçok kütüphaneyi kurulu şekilde getirdiğinden ihtiyaç olan birçok kütüphaneyi kurmak yerine Anaconda kurulumu gerçekleştirilmiştir. Anaconda kendi sitesi üzerinden(<https://www.anaconda.com/>) Windows kurulumu yapılmıştır. Python kullanımı için özel olarak tasarlanmış Pycharm programı da üzerinden model yazılmıştır. Interpreter olarak Anaconda dosyası gösterilerek program yazılmıştır.

4.2.2. Tensorflow

Google tarafından derin öğrenme uygulamaları için geliştirilmiş açık kaynaklı bir kütüphanedir. Tensorflow, bir tek API üzerinden masaüstü, sunucu veya mobil cihazdaki bir veya daha fazla veya CPU'ya dağıtma olanağı sağlamaktadır. Dünya çapında birçok firmayla çalışan Tensorflow'u kurmak için iki seçenek bulunmaktadır. Birincisi CPU, diğeri ise GPU kurulumu yapmaktır. CPU kurulumunda modellerinizin eğitimin sadece işlemci hızınız belirleyici olurken GPU kurulumunda ekran kartınızın markası ve özellikleri önemli olacaktır. GPU ile eğitilen eğitimler eş zamanlı olarak ekran kartı da kullanılacağından çok daha fazla hız sağlayacaktır. Bu tez çalışmasında Google Cloud platformuyla karşılaştırmalar yapılacağından gereksinimler en asgari düzeyde tutulmasına özen gösterilmiş olup sadece CPU ile kurulum yapılmıştır. Veriler de CPU ile geliştirilen model üzerinden verilecektir. RISELab sitesinden alınan Şekil 4.3'de görüldüğü gibi Tensorflow benzer olduğu birçok kütüphaneye göre daha başarılıdır. Şekil 4.3'deki rakamlar popüler e-baskı hizmeti arXiv.org'da yayınlanan makalelerin basit tam metin aramalarına dayanmaktadır. Spesifik olarak, her bir çerçeveden bahseden(tam metin aramasında) makale sayısını yansıtmaktadır. Ocak 2019'dan Haziran 2019'un sonuna kadar yaklaşık 1800 makalede Tensonflow'dan bahsedilmiştir. Ayrıca %23 lük bir artış gözükmetedir. (Lorica, 2019).



Şekil 4.3 : Tensorflow ve Diğer Kütüphaneler (Lorica, 2019).

4.2.3. Sanal Ortam (Virtual Environment)

Python programında sanal bir ortam hazırlamak için Virtual Environment oluşturulmalıdır.. Çok fazla proje için kullanılacak paket çakışmaları ve versiyon uyumsuzluklarına engel olmak için her biri birbirinden bağımsız Python ortamları oluşturmak gerekir. Örneğin iki farklı projenizde farklı versiyonlarda Python kullanıyorsa bu çakışmadan dolayı doğabilecek problemleri farklı ortamlar oluşturarak üstesinden gelinebilir. Sanal ortam oluşturmak daha kolay yönetim kolaylığı sağlamaktadır. Bir sunucu üzerinden projelerin birbirine karışmamasını ve silinen bir projeden dolayı diğerinin etkilenmemesini sağlamaktadır. Kurulan sanal ortam üzerinden iş yapmak için o ortam aktif edilmelidir. Komut ekranında kurulumu yapılmıştır.

4.2.4. Kütüphaneler

Opencv: Gerçek zamanlı uygulamalara odaklı görüntü işleme kütüphanesidir. Birçok alanda kullanılmakla beraber derin öğrenme alanında birçok fonksiyonu kolaylık yapılmasına olanak sağlayan kütüphanedir. Birçok yazılım dilinde ara yüzlere sahiptir. Ayrıca Windows, Linux, Mac OS, iOS ve Android'i destekler.

Imageio: Resimler üzerinde işlem yapmayı sağlayacak kütüphanedir.

Protopuf: Google'ın kendi içindeki veri iletişiminde de sıklıkla kullanıldığı veri transfer protokolüdür. Yapılandığı verilerin serileştirilmesini sağlar. Xml'e göre daha küçük ve hızlı bir yapıdadır.

Pillow: Pillow, PIL olarak da geçen, Python Image Library kütüphanesidir. Güncel Python sürümlerinde stabil olarak çalışmaktadır. Pillow kütüphanesi ile yeni resim dosyaları yaratabilir, var olanları düzenleyebilir, birden fazla resim dosyasını birbiriyle birleştirebilir, resimleri RGB olarak 3 katmana ayırabilir ve daha pek çok işlem yapabilirsiniz. Pillow günümüzdeki popüler jpg, png gibi pek çok resim formatını destekler.

Lxml: Etiket isimleri ve özelliklerini yaparken kullanılan Python API'leri için özel geliştirilmiş bir kitaplıktır.

Matplotlib: Python veri görselleştirme için kullanılan bir kütüphanedir. 2 ve 3 boyutlu çizimlerde kullanılmaktadır.

Pandas: Veri keşfi ve analizi sırasında kullanılan bir Python modülüdür.

4.2.5. Yerel Makine Sistemi ve Eğitim Kümesi

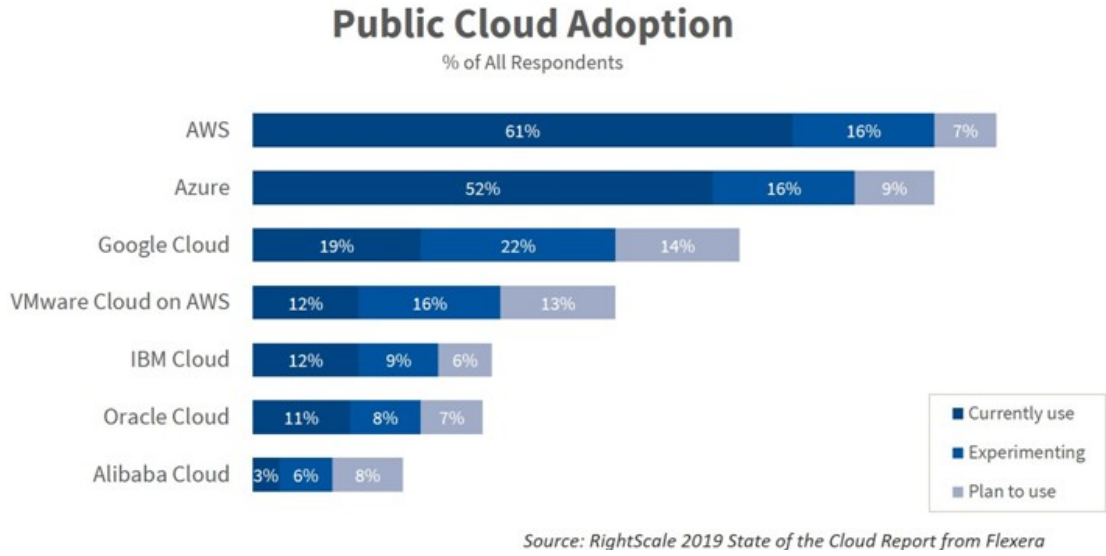


Şekil 4.4 : Eğitim Süreci Akış Şeması.

Yerel makinede sisteminde model oluşturma adımları Şekil 4.4'te gösterilmektedir. Şekil 4.4'te görüldüğü gibi model oluşturma işlemi 4 adımdan oluşur. Birinci aşamada bir Python dağıtımı olan Anaconda kurulumu yapılmıştır. Anaconda, makine öğrenmesi, görüntü işleme, veri işleme, grafik oluşturma için kullanılacak birçok kütüphaneyi kurulu şekilde getirdiğinden ihtiyaç olan birçok kütüphaneyi kurmak yerine Anaconda kurulumu gerçekleştirilmiştir. Anaconda kurulumundan sonra, Google tarafından derin öğrenme uygulamaları için geliştirilmiş açık kaynaklı bir kütüphane olan Tensorflow kurulumu gerçekleştirilmiştir. Tensorflow, bir tek API üzerinden masaüstü, sunucu veya mobil cihazdaki bir veya daha fazla veya CPU'ya dağıtma olanağı sağlamaktadır. Çok fazla proje için kullanılabilecek paket çakışmaları ve versiyon uyumsuzluklarına engel olmak için ise Virtual Environment kurulumu yapılmıştır. Proje esnasında pillow, xml, matplotlib ve pandas kütüphaneleri de ayrıca kurulmaktadır. Projenin 2. Aşamasında eğitim esnasında kullanılacak resimler toplanmıştır. Resimler üzerinde nesne algılaması yapılacağından dolayı resimler üzerindeki çiçeklerin yerlerinin belirtilmesi işlemi olan resim etiketleme yapılmıştır. Resim etiketleme üzerine yazılmış olan labellmg programı kullanılmıştır. 3. aşmada ise oluşturulan xml dosyaları csv dosyalarına dönüştürülerek Tensorflow'un veri depolaması için Tfrecord dosyaları oluşturulmuştur. Daha sonrasında Label Map oluşturularak eğitim gerçekleştirilmiştir. Son aşamada ise, eğitim esnasında kullanılan resimlerin sayısına ve kalitesine bağlı olarak eğitim süresi belirlenmiştir. Model resim, video ve web cam araçlarıyla test edilerek kullanıma hazır hale getirilmiştir. Oluşturan model belirlenen resimler üzerinden test edilmiştir.

4.2.6. Bulut Platformları

Bulut bilişim, işletmelerin çalışma prensiplerinde ve düzenlerinde yavaş yavaş kalıcı yenilikler yapan teknolojilerdendir. Bulut bilişiminde dünyada önde gelen dört tane platform öne çıkmaktadır. Bunlardan en çok bilinen ve kullanılanları Amazon Web Servisleri (AWS), Microsoft Azure, Google Cloud ve Huawei Cloud'tur. Şekil 4.5'te 2019 yılındaki piyasa durumu gösterilmiştir.



Şekil 4.5 : Platformların Benimsenme Grafiği (Dignan, 2020).

Bu platformlar, işletmelerin genel bulut ihtiyaçlarına en uygun çözümü sunmaya çalışmaktadırlar. Fakat her platformun kendisine göre güçlü ve zayıf yönleri mevcuttur.

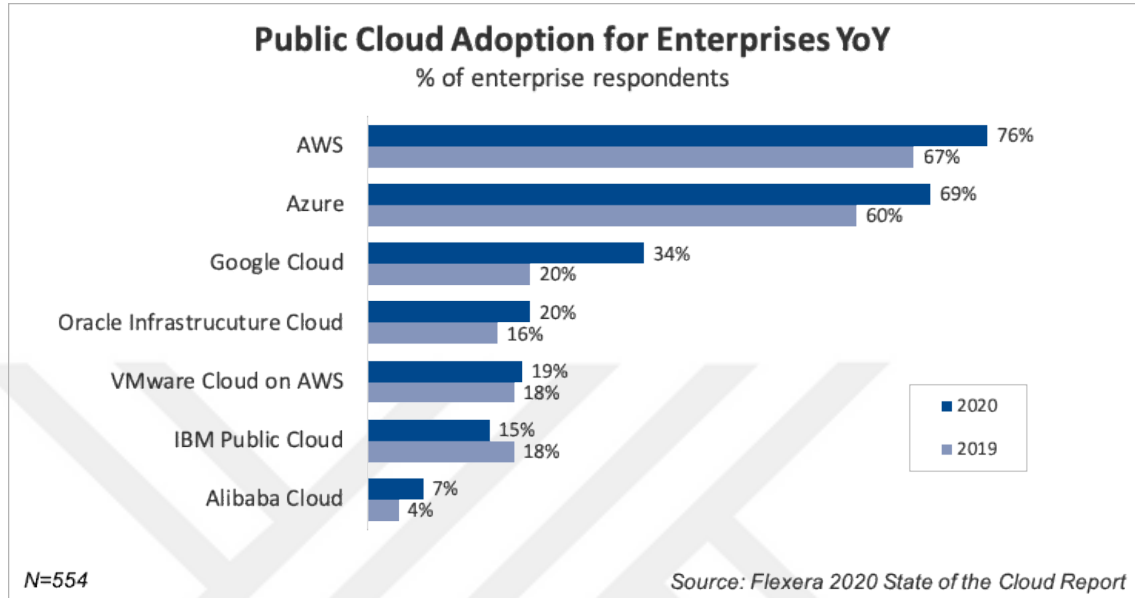
Amazon Web Servisleri (AWS): Bu alanın en eski ve en büyük pazar payına sahip platformudur. Bulut bilişim pazarı gün geçtikçe payını artırmaya rağmen Amazon'un payı giderek azalmaktadır. Dünya çapında kullanılabilir hizmet veren AWS, dünyanın en geniş veri merkezleri ağına da sahibidir. Hala birçok kullanıcı tarafından maliyet sorunlarına rağmen tercih edilmektedir.

Azure: Microsoft'un dünyada yaygın olarak kullanılan kendi yazılım çözümleriyle uyumluluğu ve bu konuda uzmanlığının bilinmesi Azure'ın en güçlü yanıdır. Microsoft'un hâlihazırdaki programlarını kullanan işletmeler, Azure'ın kurumsal yapısının çok daha yeni ve oturmamış olmasına rağmen birçok indirimle birlikte Azure'ı tercih etmektedirler.

Huawei: Bu alandaki son oyuncudur, zor ve olağan dışı iş senaryolarını ele almak için gayet kullanışlı ve değişen şartlara uygun dağıtım modelleri sunmaktadır. 5G ve AI gibi yeni lider teknolojilerin yanı sıra dijital dönüşüm konusunda yardımcı olan uygulamalarını 170'ten fazla ülkede faaliyet göstermektedir. Marka bilinirliği artırma yolunda olan Huawei bir Çin ürünüdür.

Google: Büyük veri, mantıksal analiz ve makine öğrenimi gibi alanlarda uzmanlığı diğerlerine göre daha fazladır. Platform dengeli bir ölçeklendirme ve yükleme

özelliklerine sahiptir. AWS ve Azure’a göre ürün çeşidi az olması ve veri merkezleri nedeniyle diğerlerine göre piyasada daha az yer kaplamaktadır. Şekil 4.6’daki grafikte bulut platformlarını üzerinde yıllara göre kurumsal katılımcı oranları gösterilmiştir.



Şekil 4.6 : Platformların Yıllara Göre Kıyaslanması (Weins, 2020).

4.2.7. Google Firebase

Firebase, 2011 yılında Envolv şirket tarafından kurulan bir platformdur. Web ve mobil uygulamalarda sunucular tarafında geliştiriciye sunulan verilerin gerçek zamanlı ve senkron bir şekilde tutulmasını sağlar. 2014 yılında Google’ın satın alınmış ve adı Google Firebase olarak değiştirilmiştir. Uygulama geliştirme ihtiyaçlarının tamamını karşılama amacı olan Google Firebase, ücretsiz kullanım (Spark) hizmetinin yanında ücretli (Flame ve Blaze) hizmetleriyle de kapsamlı yönetim ve ek özellikler de sunmaktadır.

Google Firebase, geliştiricilere kolay ve çözüm odaklı birçok özellik sunmaktadır. Bu özelliklerden öne çıkanları şöyle sıralamak mümkündür:

1. Gerçek Zamanlı Bir Veri Tabanı
2. Depolama
3. Makine Öğrenme Kitleri

4. Uygulama Analizi
5. Reklam Araçları
6. Kullanıcı Etkinlikleri (Açıkgöz, 2018)

Google Firebase sağladığı araçlarla tek platform üzerinde kullanmak isteyenler için yine bir başka Google ürünü olan Gmail hesabı üzerinden hizmet veren bir platformdur.

4.2.8. AutoML Vision Edge ile Modeli Eğitme

Makine öğrenme teknolojisinin günümüzde çok fazla ilerlemesindeki şüphesiz en önemli parametre, ileri teknolojik donanımların gelişmiş olmasıdır. Makine öğrenmesi teknolojisi çok fazla hesaplama yaparak karışık işlemleri çözmede etkililiğini her geçen gün daha fazla göstermektedir. Mobil cihazlarda bu ilerleyen teknolojiyle beraber daha güçlü ve daha verimli algoritmaları barındırmaya başlamıştır. Mobil cihazlarda makine öğrenme teknolojisi birçok alanda kullanılmaktadır:

Akıllı Asistan Sistemleri: Siri, Cortana, Google Assistant gibi.

Akıllı Cevaplama: Whatsapp gibi sohbet programlarına dahil edilen, gelen bir mesaja belirli kalıplar içerisinde cevap veren mesaj şablonudur.

Filtreler: Snapchat, Instagram vb. sosyal medya uygulamalarında kamera karşısındaki insanın yüzlerini, gözlerini tespit ederek gözlük, şapka, göz rengi değiştirme gibi 3D filtreler, uygulamak için kullanılır.

Bixby Vision-Google Lens: Mobil cihazınızın kamerasını nesneler tanımak ve etiketleme olarak kullandırmayı amaçlayan uygulamalardır.

Firestore ML Kit de mobil geliştiricilerin uygulamalarına makine öğrenme yeteneklerini eklemelerini kolaylaştıran yazılım geliştirme kitidir (Software Development Kit, SDK). Firestore özel görüntü sınıflandırma modelleri için makine öğrenmesini kullanan modeli AutoML modelidir. AutoML Kiti temel olarak mobil uygulamanıza makine öğrenme yeteneklerini dâhil etme ve kullanma karmaşıklığını en aza indiren bir sistemdir.



Şekil 4.7 : Eğitim Süreci Akış Şeması.

Bu tezde, AutoML modeli Google bulut servisleri ve Firebase kullanılarak geliştirilmiş olup izlenen aşamalar Şekil 4.7’de gösterilmiştir. 1. Aşamada Gmail hesabına kayıtlı olarak firebase.google sitesindeki konsol üzerinden bir proje oluşturulmuştur. 2. Aşamada ise yerel makine eğitiminde kullanılan veriler bir araya getirilerek etiketleme işlemi gerçekleştirilmiştir. Yüklenen resimler 30 MB altında resimler olup her çiçek türü için belirlenen sayıda çiçek girişi yapılmıştır. 3. Aşamada ise siteye yüklenen resimler uygun etiketlerle eğitilmesi sağlanmıştır. Verilere uygun olarak eğitim sayısı ve süresince eğitim gerçekleştirilmiştir. Son aşama olan modelin oluşturulması aşamasında ise model puan eşiği, model performansı gibi bilgiler görüntülenmiştir. Model bu aşamadan sonra kullanıma hazır hale getirilmiştir. AutoML Kiti üzerinden yapılan eğitimlerle oluşan model belirlenen verilerle test edilmiştir.

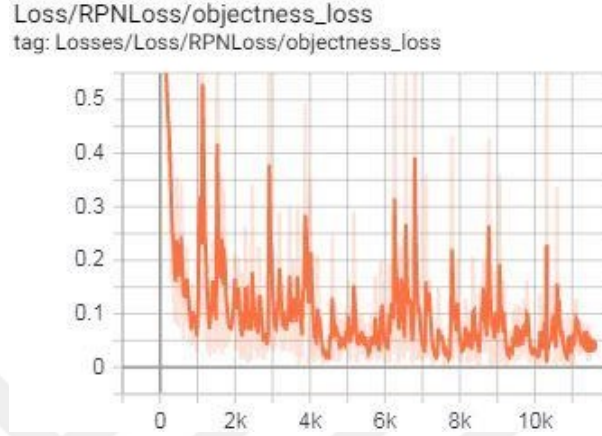


BEŞİNCİ BÖLÜM

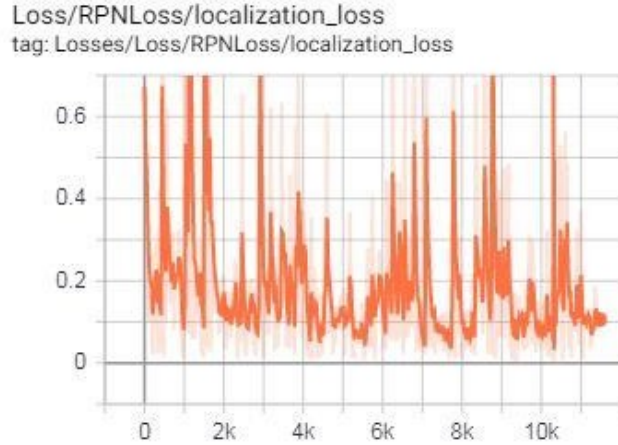
BULGULAR

5. BULGULAR

Yerel makinede 95 bin 73 tane eğitim gerçekleştirilirken Şekil 5.1'deki sonuç grafikleri elde edilmiştir. Ortalama olarak verilen çiçek sayısı başına yaklaşık 74 adet eğitim gerçekleştirilmiştir. Eğitim sonucunda kayıp (loss) değerlerine ait bilgiler Şekil 5.2'da gösterilmiştir.



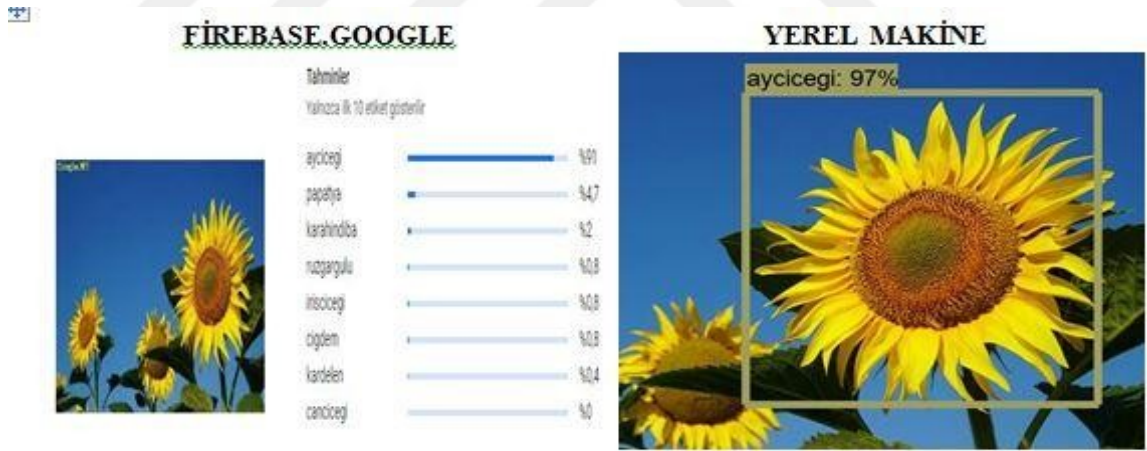
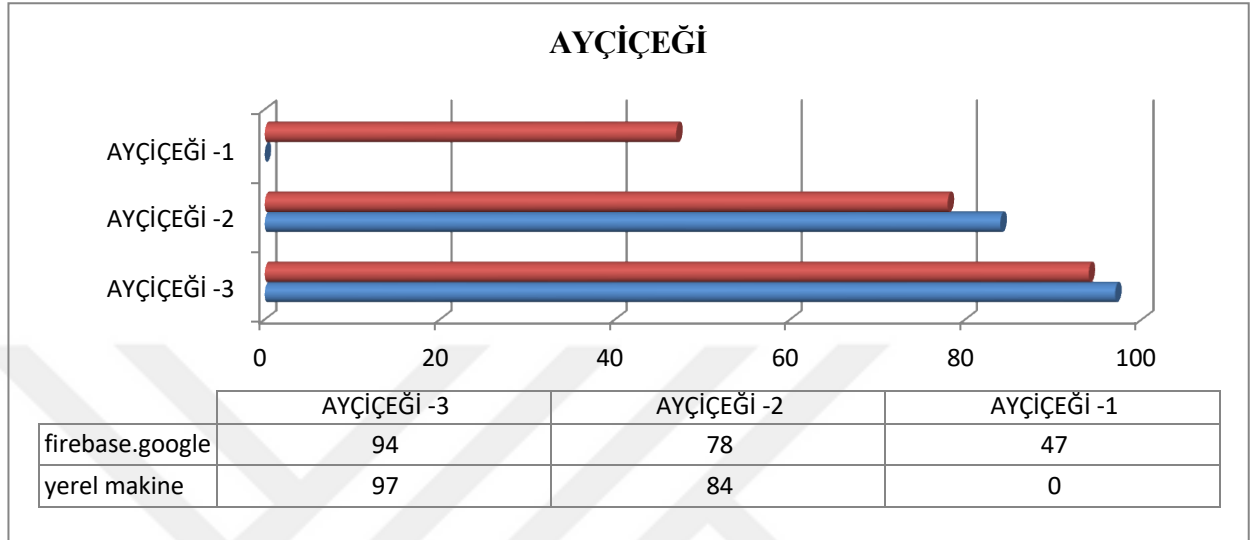
Şekil 5.1 : Nesne Kayıp Grafiği.



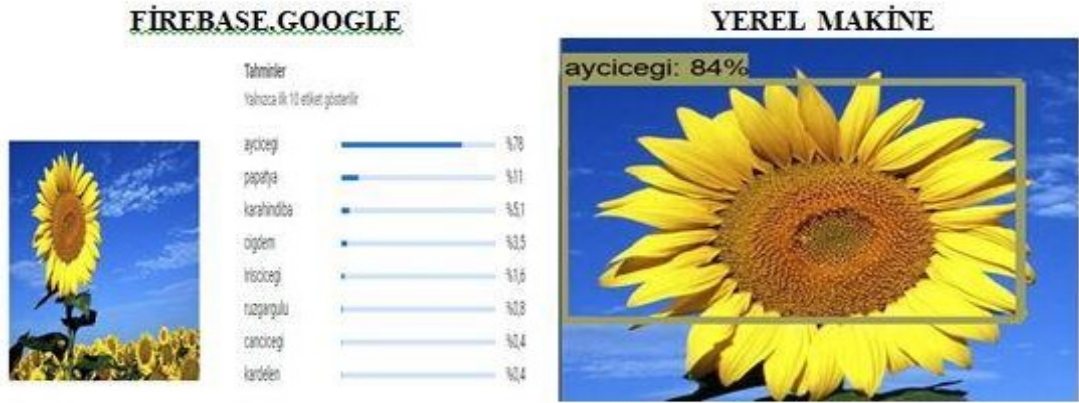
Şekil 5.2 : Konum Kayıp Grafiği.

5.1. Resimlerin Deneysel Sonuçları

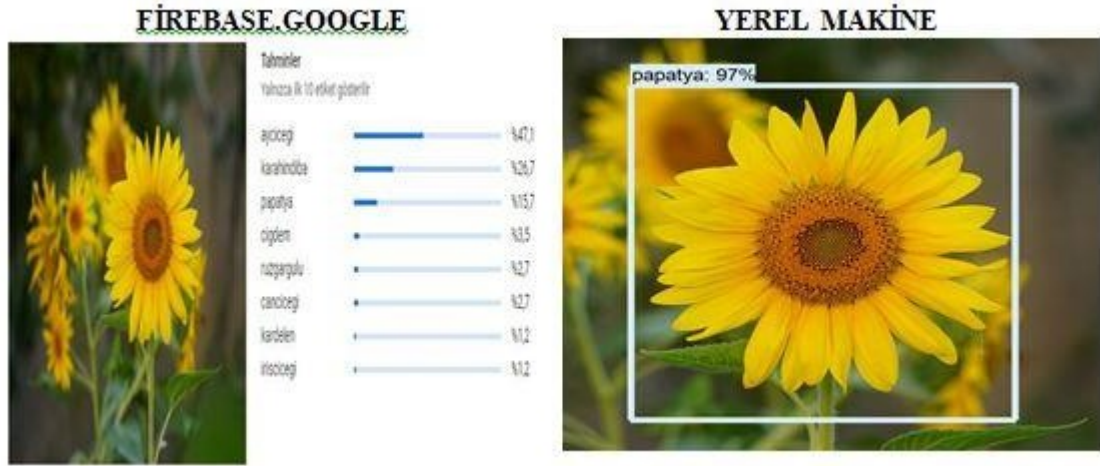
Tablo 5.1 : Ayçiçeği Veriler Tablosu.



Şekil 5.3 : Ayçiçek-1 Tahmin Yüzdeleri.

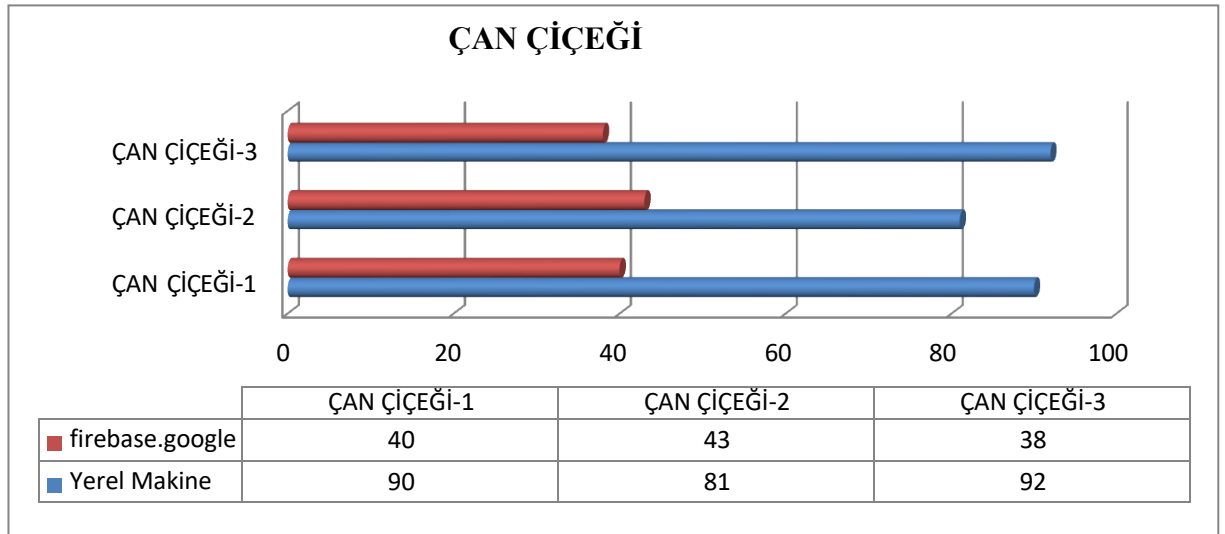


Şekil 5.4 : Ayçiçek-2 Tahmin Yüzdeleri.



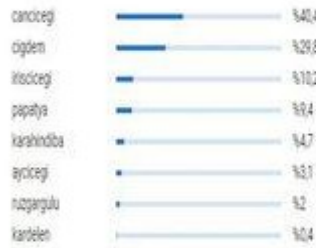
Şekil 5.5 : Ayçiçek-3 Tahmin Yüzdeleri.

Ayçiçek veri setinde üç tane rastgele alınan ayçiçeği resimleri hem firebase üzerinden eğitilen hem de yerel makinelerde eğitilen modeller üzerinde test edilmiştir ve Tablo 5.1'deki sonuçlar elde edilmiştir. Şekil 5.3 ve Şekil 5.4'deki ilk iki resimde çok belirgin bir fark olmasa da Şekil 5.5'deki üçüncü resimde yerel makinede eğitilen model %97'lik bir oranla "papatya" tahmininde bulunurken firebase üzerinden eğitilen model %47,1'lik bir oranla "ayçiçeği" tahmininde bulunmuştur. Şekil 5.5' de yerel makine ayçiçeğini papatyadan ayırt edememiştir. Bu durumda düşük olsa da firebase doğru tahmin de bulunmuştur.

Tablo 5.2. Çançiçeği Veriler Tablosu.**FİREBASE.GOOGLE**

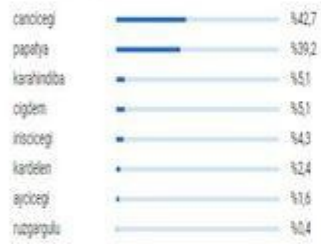
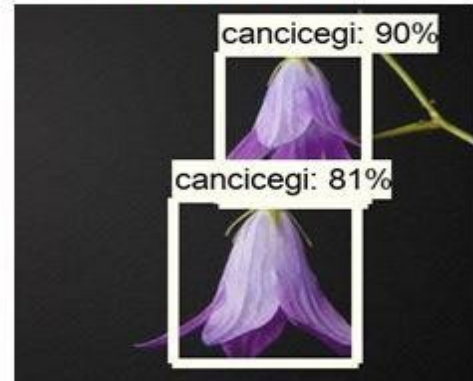
Tahminler

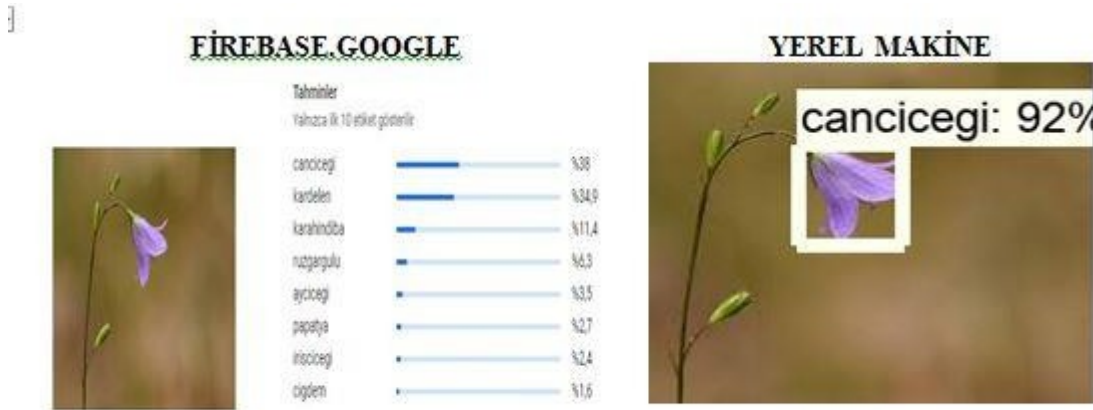
Yalnızca ilk 10 etiket gösterilir

**YEREL MAKİNE****Şekil 5.6 :** Çan Çiçeği-1 Tahmin Yüzdeleri.**FİREBASE.GOOGLE**

Tahminler

Yalnızca ilk 10 etiket gösterilir

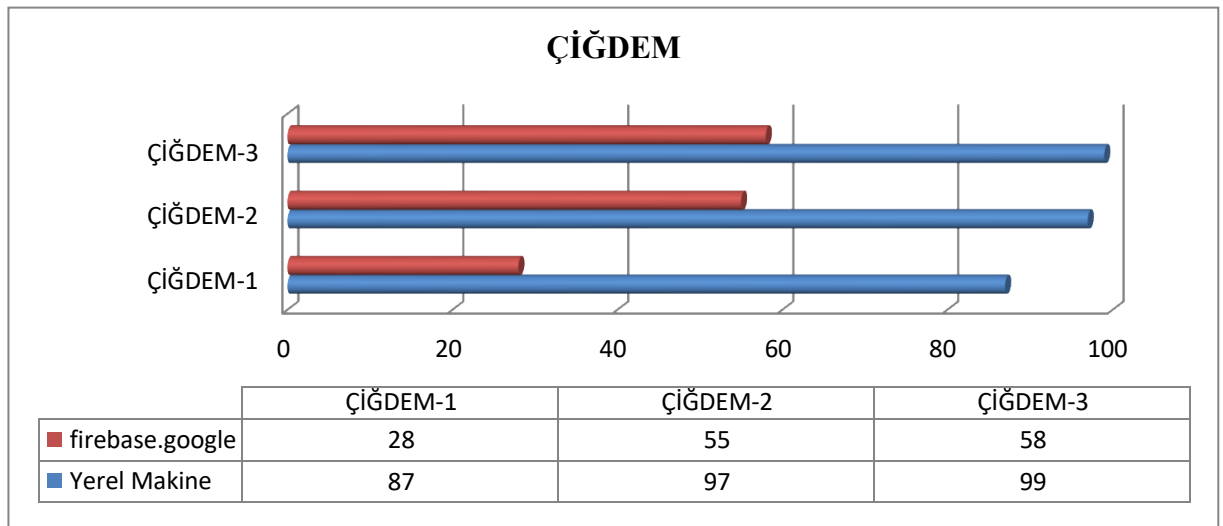
**YEREL MAKİNE****Şekil 5.7 :** Çan Çiçeği-2 Tahmin Yüzdeleri.

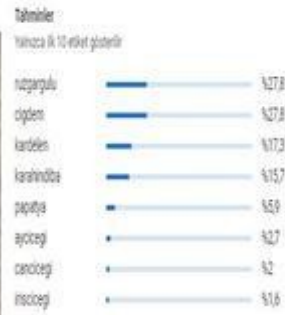
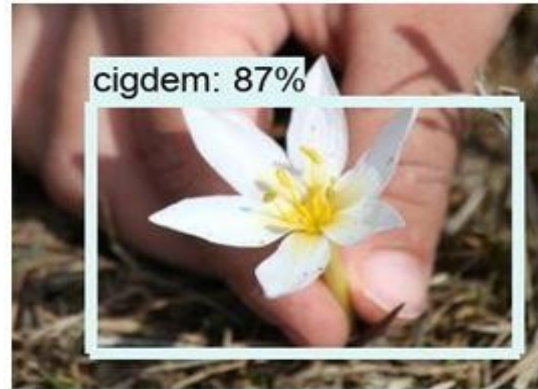
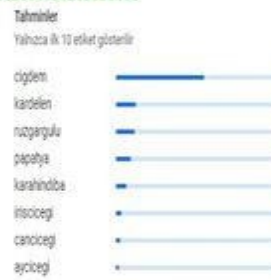
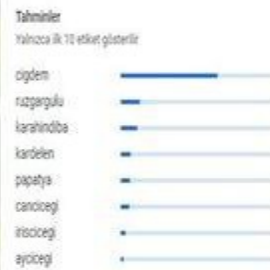


Şekil 5.8 : Çan Çiçeği-3 Tahmin Yüzdeleri.

Çan çiçeği veri setinde üç tane rastgele alınan ayçiçeği resimleri hem firebase üzerinden eğitilen hem de yerel makinelerde eğitilen modeller üzerinde test edilmiştir ve Tablo 5.2'deki sonuçlar elde edilmiştir. Şekil 5.6, Şekil 5.7 ve Şekil 5.8'deki gösterilen üç resimde de yerel makine modeli %80'in üzerinde bir oranla doğru tahminde bulunurken firebase modeli %45'in altında bir başarı oranı ile tahminde bulunmuştur.

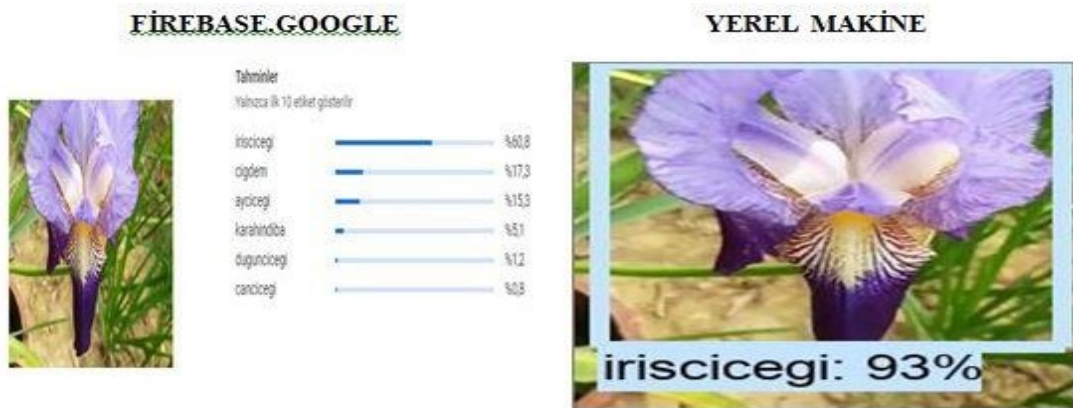
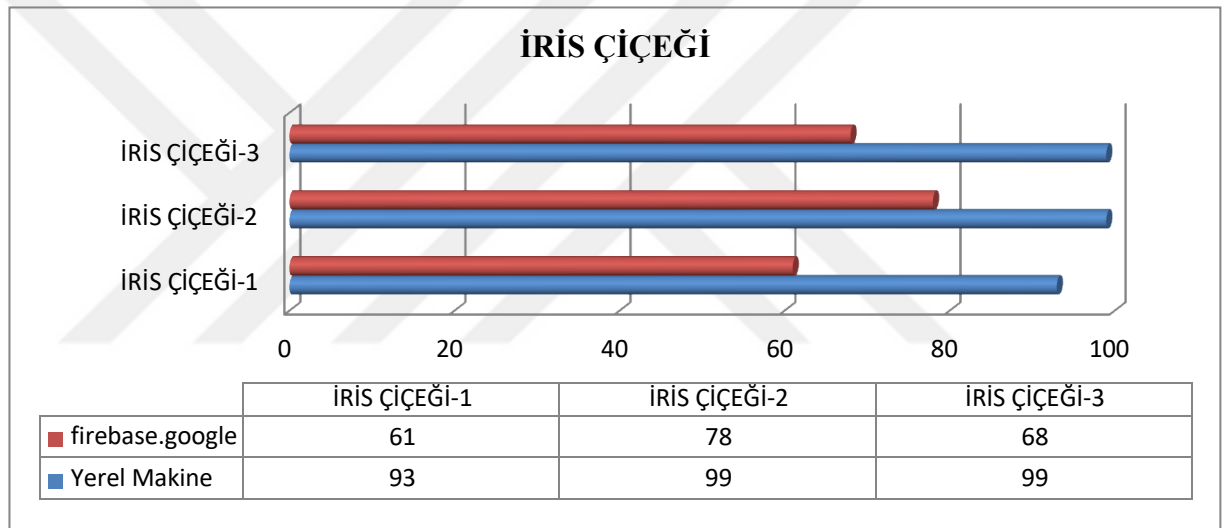
Tablo 5.3 : Çiğdem Veriler Tablosu.



FİREBASE.GOOGL**YEREL MAKİNE****Şekil 5.9 : Çiğdem-1 Tahmin Yüzdeleri.****FİREBASE.GOOGL****YEREL MAKİNE****Şekil 5.10 : Çiğdem-2 Tahmin Yüzdeleri.****FİREBASE.GOOGL****YEREL MAKİNE****Şekil 5.11 : Çiğdem-3 Tahmin Yüzdeleri.**

Çiğdem veri setinde üç tane rastgele alınan çiğdem resimleri hem firebase üzerinden eğitilen hem de yerel makinelerde eğitilen modeller üzerinde test edilmiştir ve Tablo 5.3 'teki sonuçlar elde edilmiştir. Şekil 5.9'da da görüldüğü gibi ilk çiçek örneğinde yerel makine modeli %85'in üzerinde bir başarı oranı ile tahminde bulunurken firebase modeli %27,8 ile yanlış bir tahmin ile "rüzgar gülü" ve aynı yüzdelikli fakat başarılı bir şekilde "çiğdem" tahmininde bulunmuştur. Şekil 5.10 ve Şekil 5.11'de görüldüğü gibi diğer iki çiçekte de firebase, %50'nin üzerinde bir başarı oranı ile tahminde bulunurken yerel makine modeli %85'in üzerinde bir başarı oranı ile tahminde bulunmuştur.

Tablo 5.4 : İris Çiçeği Veriler Tablosu.



Şekil 5.12 : İris Çiçeği -1 Tahmin Yüzdeleri.

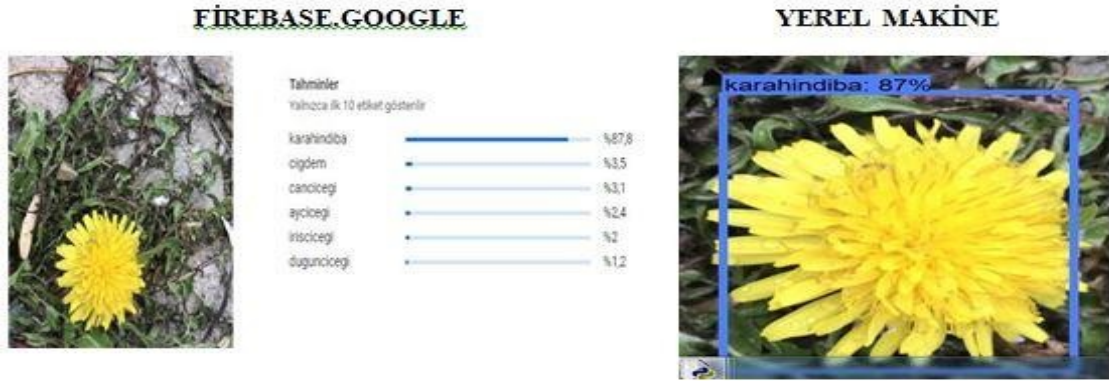
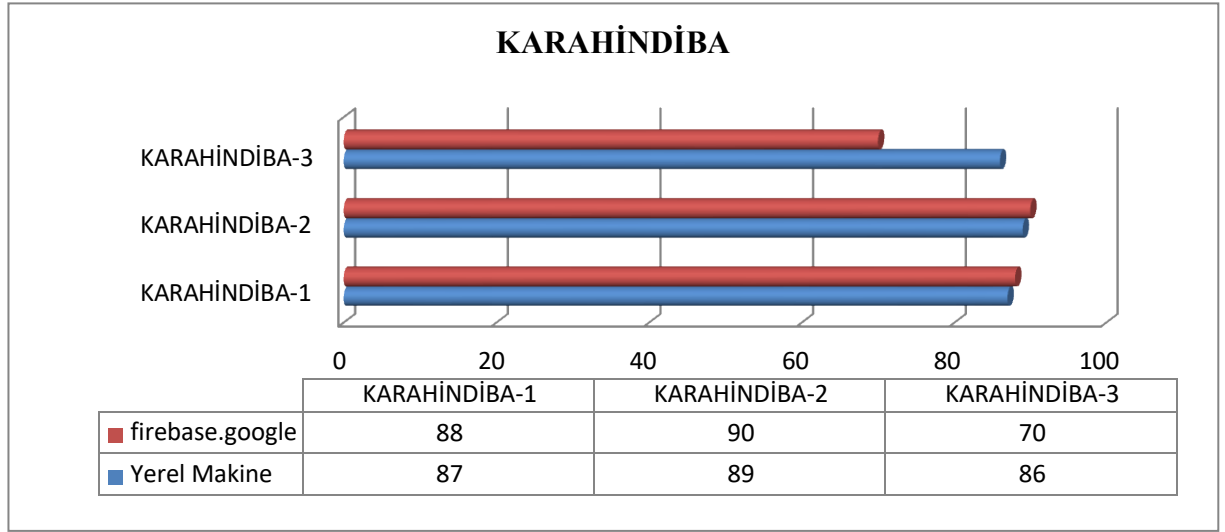
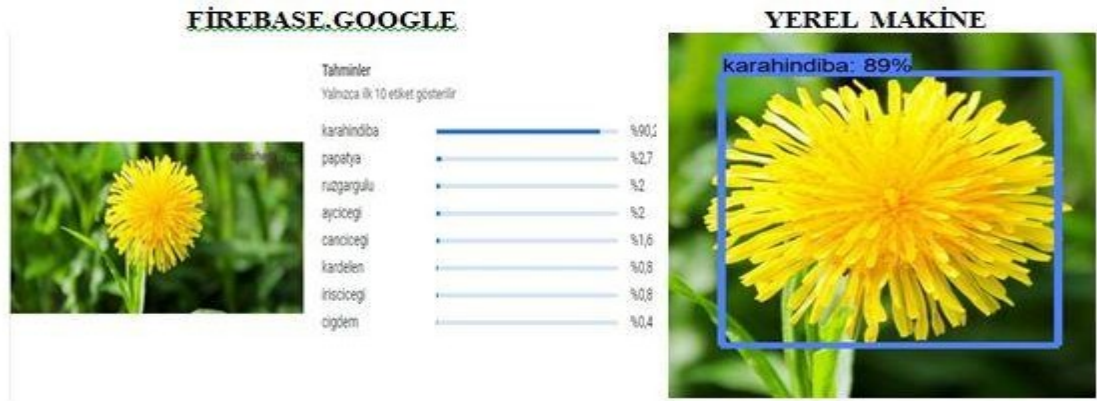


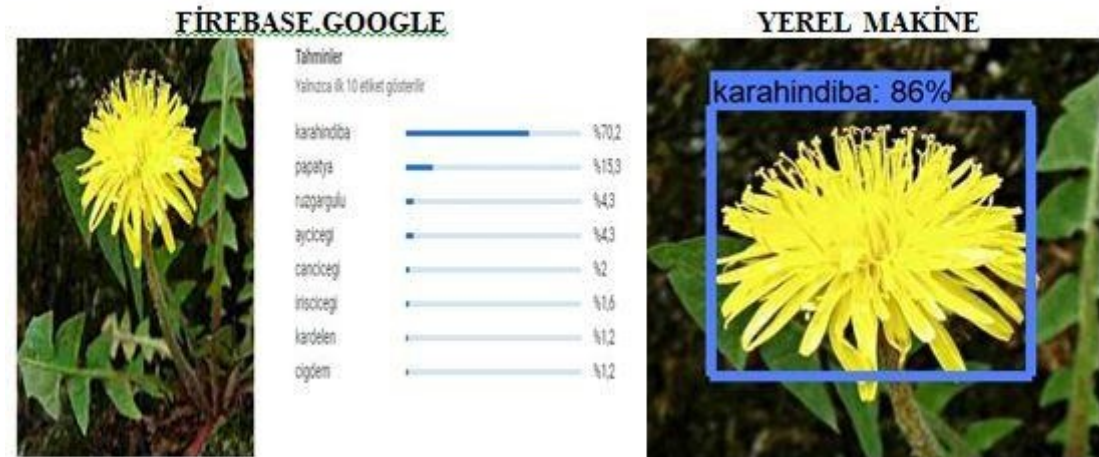
Şekil 5.13 : İris Çiçeği -2 Tahmin Yüzdeleri.



Şekil 5.14 : İris Çiçeği -3 Tahmin Yüzdeleri.

İris çiçeği veri setinde üç tane rastgele alınan iris çiçeği resimleri hem firebase üzerinden eğitilen hem de yerel makinelerde eğitilen modeller üzerinde test edilmiştir ve Tablo 5.4 'teki sonuçlar elde edilmiştir. Şekil 5.11, Şekil 5.12 ve Şekil 5.13'de de görüldüğü gibi firebase her üç çiçek örneğinde %60'ın üzerinde bir başarı ile tahminde bulunurken, yerel makine modeli %99 başarı oranı ile iris çiçeği tahmininde bulunmuştur.

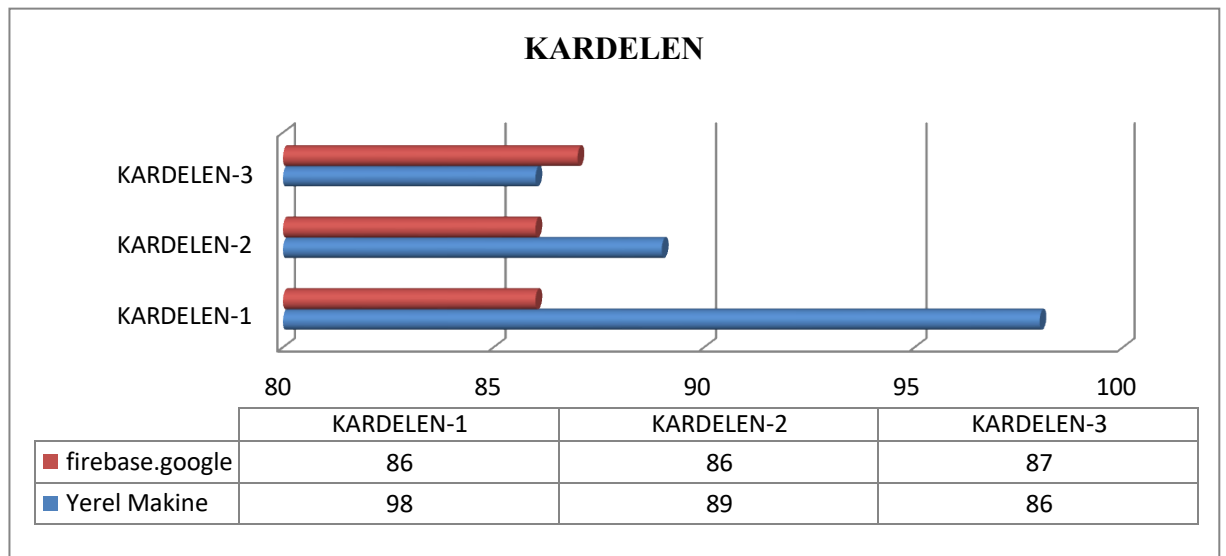
Tablo 5.5 : Karahindiba Veriler Tablosu.**Şekil 5.15 : Karahindiba -1 Tahmin Yüzdeleri.****Şekil 5.16 : Karahindiba -2 Tahmin Yüzdeleri.**

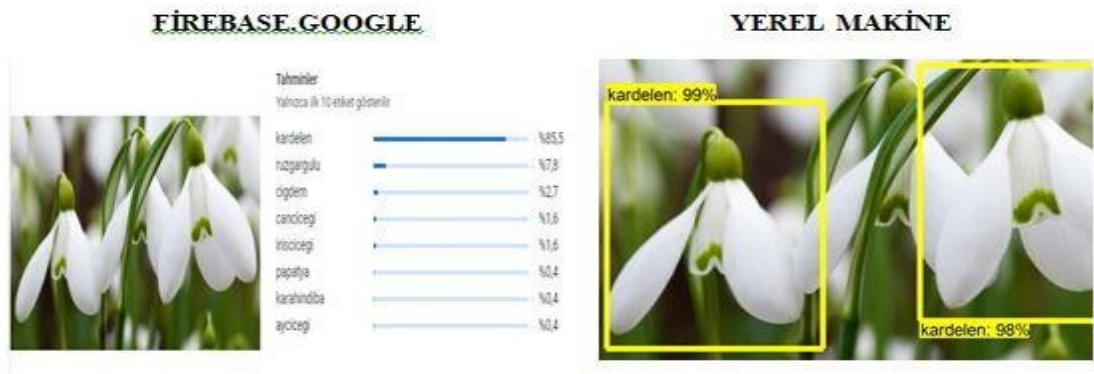


Şekil 5.17 : Karahindiba -3 Tahmin Yüzdeleri.

Karahindiba veri setinde üç tane rastgele alınan karahindiba resimleri hem firebase üzerinden eğitilen hem de yerel makinelerde eğitilen modeller üzerinde test edilmiştir ve Tablo 5.5 'deki sonuçlar elde edilmiştir. Şekil 5.15 ve Şekil 5.16'ta da görüldüğü gibi her iki modelin başarılı tahmin sonuçları birbirine yakındır. Şekil 5.17'de gösterilen üçüncü örnekte ise yerel makinede eğitilen model %86'lık bir başarı oranı sergileyerek firebase göre %16 oranında daha iyi bir tahminleme yaptığı görülmektedir.

Tablo 5.6 : Kardelen Veriler Tablosu.

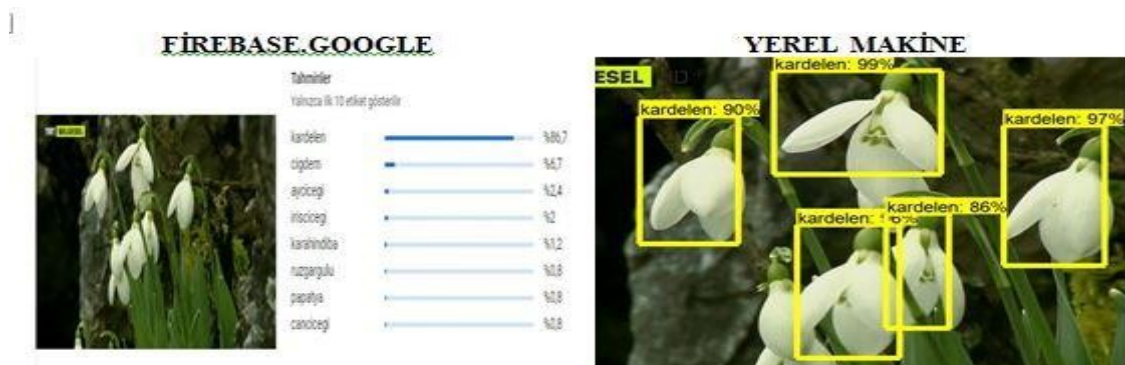




Şekil 5.18 : Kardelen -1 Tahmin Yüzdeleri.



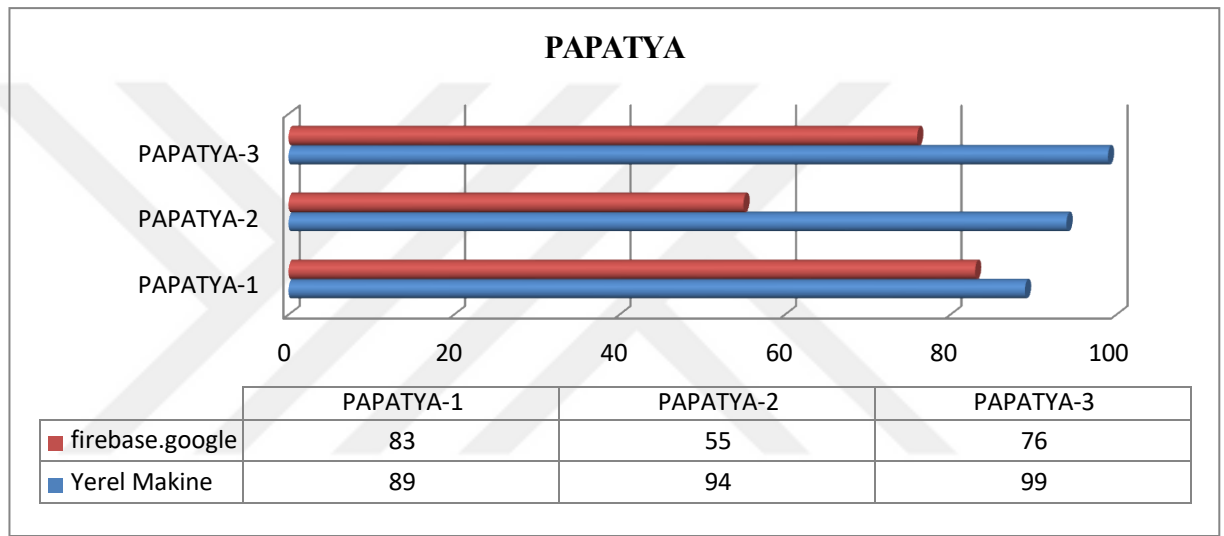
Şekil 5.19 : Kardelen -2 Tahmin Yüzdeleri.



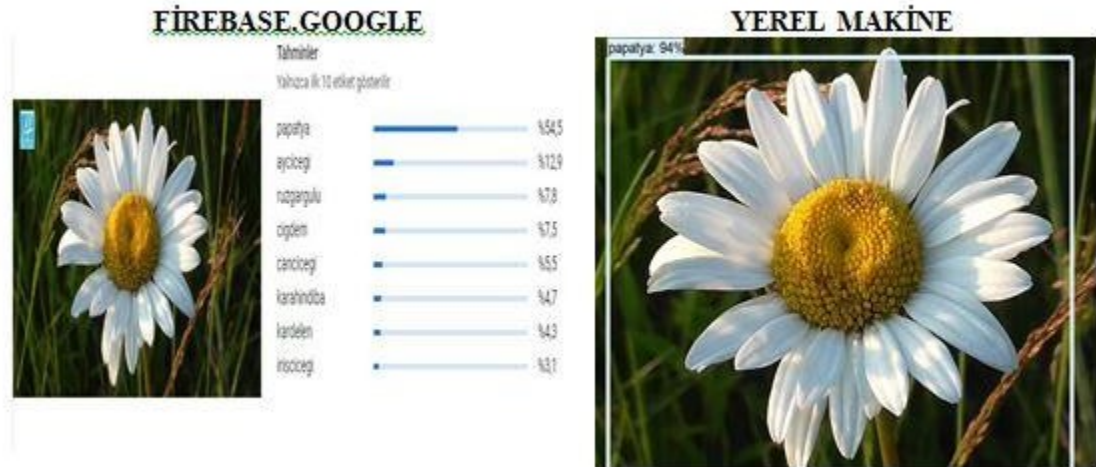
Şekil 5.20 : Kardelen -3 Tahmin Yüzdeleri.

Kardelen veri setinde üç tane rastgele alınan kardelen resimleri hem firebase üzerinden eğitilen hem de yerel makinelerde eğitilen modeller üzerinde test edilmiştir ve Tablo 5.6'daki sonuçlar elde edilmiştir. Şekil 5.18 ve Şekil 5.19'da da görüldüğü gibi her iki model de üç çiçek türünde %85'in üzerinde bir başarı ile tahminde bulunmuştur. Şekil 5.20'de yerel makine modeli %98'lik bir başarı oranı ile firebase modelinden %12'lik daha iyi bir tahminleme yapmıştır. Şekil 5.19 ve Şekil 5.20'de ise başarı oranları benzerdir.

Tablo 5.7 : Papatya Veriler Tablosu.



Şekil 5.21 : Papatya -1 Tahmin Yüzdeleri.



Şekil 5.22 : Papatya -2 Tahmin Yüzdeleri.

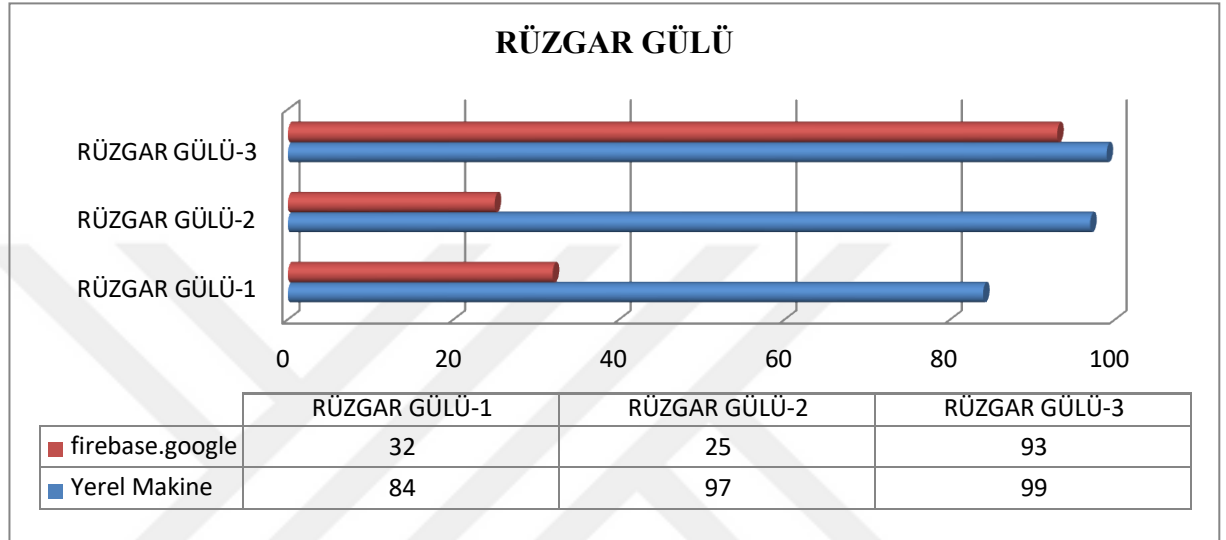


Şekil 5.23 : Papatya -3 Tahmin Yüzdeleri.

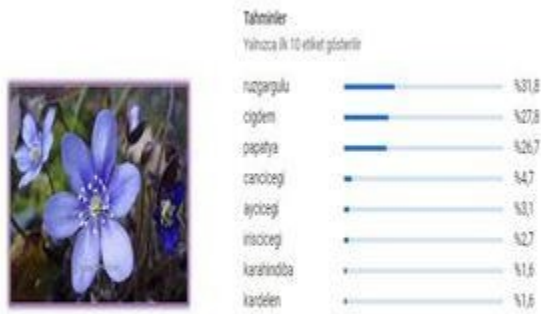
Papatya veri setinde üç tane rastgele alınan papatya resimleri hem firebase üzerinden eğitilen hem de yerel makinelerde eğitilen modeller üzerinde test edilmiştir ve Tablo 5.7'deki sonuçlar elde edilmiştir. Şekil 5.21 ve Şekil 5.23'te de görüldüğü gibi yerel makine modelinde %85 üzerinde başarılı bir tahminleme yaparken firebase üzerinden eğitilen model Şekil 5.21 ve Şekil 5.23'deki örneklerde %75'in üzerinde bir

başarı ile tahminde bulunmuştur. Fakat Şekil 5.22'deki örneğinde başarılı tahmin yüzdesi %55'te kalmıştır. Aynı resimde yerel makine modeli %94'lük bir başarı ile tahminleme yaptığı görülmekte olup çok daha iyi bir tahminlemede bulunduğu açıkça görülmektedir.

Tablo 5.8 : Rüzgâr Gülü Veriler Tablosu.



FIREBASE GOOGLE



YEREL MAKİNE



Şekil 5.24 : Rüzgâr Gülü -1 Tahmin Yüzdeleri.



Şekil 5.25 : Rüzgâr Gülü -2 Tahmin Yüzdeleri.



Şekil 5.26 : Rüzgâr Gülü -3 Tahmin Yüzdeleri.

Rüzgar gülü veri setinde üç tane rastgele alınan rüzgar gülü resimleri hem firebase üzerinden eğitilen hem de yerel makinelerde eğitilen modeller üzerinde test edilmiştir ve Tablo 5.8'deki sonuçlar elde edilmiştir. Şekil 5.24, Şekil 5.25 ve Şekil 5.26 'da da görüldüğü gibi üç çiçek örneğinde de yerel makinelerde %80'in üzerinde bir başarı oranı ile tahminde bulunulmuştur. Firebase sadece Şekil 5.26'daki örnekte %93 oranında başarılı bir şekilde "rüzgar gülü" tahmininde bulunabilmekte Şekil 5.24'te ve Şekil 5.25'teki örneklerde ise başarılı tahmin yüzdesi %35'in altına düşmektedir.

Bütün verilerden yola çıkarak bakıldığında yerel makine test edilen 24 çiçek resminden 23 tanesi doğru pozitif olmuştur. 1 tane çiçek resminde ise başka bir çiçek türü tahmininde bulunduğundan bir tanesi yanlış negatif olmuştur. Firebase ise 27 tane

iekten 6 tanesini %50 nin altında tahmin etmesine raėmen doėru iek tr olarak belirlemiřtir. Fakat 1 tane iek resmine bařka bir iek tr olarak tahmin etmiř ve yanlıř negatif olmuřtur. Bu durumda zetle hem yerel makinede hem de firebase de doėru pozitif 23 ve yanlıř negatif 1 olmuřtur. Bu deėerlerden yola ıkararak hem hassasiyetleri em geri aėırma yzdeleri ve hem de F1 puanı eřit olacaktır.





ALTINCI BÖLÜM
SONUÇLAR, TARTIŞMA ve ÖNERİLER

6. SONUÇLAR, TARTIŞMA ve ÖNERİLER

Ayçiçeği testinde ilk iki çiçek örneğinde her iki modelin başarısı benzer olmasına karşın, üçüncü test edilen ayçiçek resminde yerel makine papatyadan ayırt edememiştir. Firebase ise yüzde 47 oranında tahmin de bulunmuş ve ayçiçeğine benzeyen karahindiba ve papatya ile karıştırmıştır. Yerel makine ayçiçeği tahmin ettiği resimlerde yüksek oranlar elde ederken tahmin edemediği resimde olmuştur. Firebase tahmin sürecinde yeteri kadar yüzdelere çıkamamış olsa da tahminde başarılı olmuştur.

Çan çiçeğinde yerel makine test için verilen üç çiçekte de başarılı tahminlerde bulunmuştur. Firebase çiçekleri ayırt etmekte zorlanmıştır.

Çiğdem çiçeğinde yerel makine modeli test için verilen çiçek örneklerinde tahmin süresince başarılı sonuçlar verirken firebase başarılı sonuçlar verememiştir. Ayrıca eğitim esnasında da test için verilen çiçek örneklerinde de yüksek yüzdeli tahminlerde bulunamamıştır.

İris çiçeği testinde yerel makine modeli yüksek bir oranda doğru tahminde bulunmuştur. Firebase ise çiçekleri iris çiçeği olarak ayırt etmekte diğer çiçek türlerine nispeten daha başarılı olmuştur. Fakat bazı resim örneklerinde yine yerel makine modelinin daha iyi sonuçlar verdiği gözlemlenmiştir.

Karahindiba çiçeğinde yerel makine modeli çiçekleri ayırt ederek başarılı tahminde bulunmuştur. Firebase iki çiçek örneğinde az bir fark ile de olsa yerel makine modeline göre daha başarılı olmuş ve yüksek oranda tahminlerde bulunmuştur. Fakat bir resimde başarı oranı yerel makineye göre belirgin bir şekilde daha düşüktür.

Kardelen çiçeğinin test sonuçlarında her iki model de yüksek oranlarda tahmin de bulunarak ayırt etmede başarılı olmuşlardır.

Papatya çiçeğinde ise yerel makine modeli yüksek oranda tahminde bulunmuştur. Firebase ise sadece bir çiçekte düşük oranlı tahminde bulunmuş ve bu çiçeği ayçiçeğiyle karıştırmıştır.

Rüzgar gülü testinde yerel makine modeli yüksek oranda tahminlerde bulunurken firebase modeli bir çiçekte yüksek oranda tahmin verebilmiş fakat diğer iki çiçek örneğinde diğer çiçeklerden ayırt edememiştir. Düşük tahminde bulunduğu çiçek örnekleri dikkate alındığında firebase modelinin renk ayırımında zorlandığını göstermektedir.

Genel sonuçlara bakıldığında yerel makine modeli ortalama %92 oranında başarılı tahminlerde bulunurken, firebase ortalama %67 oranında bir başarı ile tahminleme yapabilmektedir. Sonuç olarak yerel makine modeli yüksek yüzdelik olarak başarı sağlamıştır. Hassasiyetleri, geri çağırma yüzdeleri, gerçek tahminleri ve F1 puanı açısından herhangi bir farklılık olmadığı gözükmemektedir.

Diğer yandan yerel makine modeli ile bazı çiçek türlerinde Google Firebase'e göre daha yakın ve yüksek oranlar ile tahminleme yapılmıştır. Makine maliyeti göz önüne alındığında, sadece yerel makine işlemcisi 500-600 \$ maliyet sahip iken Google Firebase ile ücretsiz olan AutoML Kiti'nin Spark Planı kullanılarak benzer sonuçlar alındığı görülebilmektedir. Kendi planınızı oluşturmak istediğimizde ücretli olan Blaze Planı tercih eder ve eğitim verisi sayısını ve süresini artırdığımızda yine çok daha uygun maliyetler ile modelin başarı oranını artırabilmek mümkündür. Bütün bunlara ek olarak çalışanın emek maliyeti de düşünüldüğünde Google Firebase Platformunda model eğitmek daha uygun olduğu görülmektedir.

Günümüz bilişim teknolojilerinde en büyük ihtiyaçlardan birisi de yetişmiş insan gücüdür. Birçok çiçek türünde Google Firebase makine öğreniminde model oluştururken kullanıcı dostu ara yüzlere sahiptir ve çok fazla bilişim teknolojileri bilgisine sahip olmak gerekmemektedir. Sosyal medya hesaplarına resim yüklemekten çok daha zor değildir. Buna karşılık yerel makine eğitim sürecinde makine öğrenmesi alanında yetişmiş en az bir çalışana ihtiyaç duyulmaktadır. Bu alanda çalışmak isteyen kişi veya şirketler için herhangi bir ön yatırıma ihtiyaç duymadan Google Firebase üzerinden makine öğrenmesi hizmeti alabilmektedirler.

Çağımızda durağan teknolojik araçların artık kullanım sahası genişlemiş, sabit olmaktan çıkmaya başlamıştır. Farklı mekânlarda ve farklı cihazlarla çalışırken veriye erişme, transfer etme ve paylaşımının hızlı ve kolay hale gelmesini sağlayan en yenilikçi teknolojilerden biri de bulut bilişim teknolojileridir (Saritaş & Üner, 2013). Bulut hesabı üzerinden eğitilen modelde yerel makine modeline göre uzaktan erişim yollarıyla daha fazla kişinin daha kolay erişimi sağlanabilmektedir. Bu da ekip çalışmalarında verimliliği artıran diğer bir özellik olacaktır. Bulut hesaplarında eğitilen modellerin mobil uygulamalar, web siteleri, sosyal medya uygulamaları vb. kullanıcı yatkınlığına göre tercih edilmesini sağlamaktadır.

Yerel makine ile model oluřturma süreci verilerin toplanıp etiketlenme süreci dıřında GPU üzerinden dahi olsa belirli zaman almaktadır. Bu süreç, verilerin etiketlenmesi faktörüyle beraber Google Firebase' e göre çok daha uzun sürecektir.

Sonuç olarak daha önce belirtilen birçok durumdan farklı olarak veri güvenlięi ayrıca ele almak gerekecektir. Yapay zekânın kullanımlarının artması hem yerel makinelerde hem de bulut platformlarında yapılan makine öğrenme süreçleri ayrıca önem arz etmektedir.

Internet Database Connector (IDC)'ye göre; yapay zekâ ve makine öğrenimi kullanımının kurumsal kaynak planlaması, içerik yönetimi, iş birlięi ve kullanıcı verimlilięi gibi alanlarda ön plana çıktığı ve on yıl içinde "endüstrileri deęiřtiren rahatsız edici bir etki"ye ulaşacaktır. Perakende ve bankacılık sektörlerinde işlerin yarısının otomatik müşteri acenteleri ve uzman alışveriş danışmanlarına, ürün öneri sistemlerine doęru kayacağını vurgulayan araştırma, aynı zamanda yeni ortaya çıkacak otomatik tehdit istihbaratı, önleme sistemleri ve sahtekarlık analiz-soruřturmalarına harcanacağını ve yapay zekâ sistemlerinin önemli yatırımlar alacağını ifade etmektedir (Internet Database Connector (IDC), 2019). Türkiye Yapay Zekâ İnisiyatifi tarafından yayınlanan rapora göre, yapay zekânın önümüzdeki yıllarda etkileri muazzam olacaktır. Yapay zeka; saęlık, eğitim, güvenlik, turizm, inřaat, ulaşım, tekstil ve tarıma kadar birçok alanda daha akıllı, daha ergonomik ve daha verimli olarak kullanılır hale gelecektir. Özel sektörde, kamusal alanda, bireysel hayatta yapay zekâyla bir şekilde muhatap olmak zorunda kalınacaktır. Bundan dolayı yapay zekanın toplumsal, hukuki ve etik boyutlarını bugünden konuşmak gerekmektedir. Rapor, yapay zekâdaki bu gelişimin ulusal güvenlik meselesi haline dönüşeceğini ön görmektedir. Milyarlarca dolarlık şirketler olan IBM, Microsoft, Apple, Google, Facebook, Amazon, Baidu'nun yapay zekâ alanında rekabet halinde olduęunun vurgulandığı raporda, rekabet üstünlüğünün ancak yapay zekayla geleceęe hazırlanmakta olduęu vurgulanmaktadır. Raporun "Teknik Konular" bölümünde kamu sektörü ve özel sektörün veri ve hesaplama gücünün yapay zekâ için iki temel bileřen olduęu, özel sektörde verinin ticari sır teşkil etmesine rağmen ticari gizlilięi ve müşteri mahremiyetini ihlal etmeden verilerin açık veri haline getirilmesinin önemine dikkat çekilmektedir (Aksu, 2017). Bu açıdan yerel makineler, veri güvenlięini ön plana alan kişi ve kurumlar için kamuya açık Google Firebase'e göre daha avantajlı konumda olacaktır.

KAYNAKÇA

- Açıkgöz, F. P. (2018, 8 2). *Google Firebase Nedir?* <https://medium.com/:https://medium.com/furkanpacikgoz/google-firebase-nedir-ae013e495a74> adresinden alınmıştır
- Akın, H. L. (2020, 05 14). *Yapay Zekada Vücut ve Beyin Problemi*. cmpe.boun.edu.tr: <https://www.cmpe.boun.edu.tr/~akin/papers/beyin.pdf> adresinden alınmıştır
- Aksu, H. (2017). *Türkiye Yapay Zeka İnisiyatifi 1. Çalıştay Raporu*. GelecekHane A.Ş.
- Aksu, H. (2019). Pusula 20 Teknoloji ve Yayıncılık A.Ş.
- Alake, R. (2020, 07 14). *How Does AI Detect Objects? (Technical)*. 12 20, 2020 tarihinde towardsdatascience.com: <https://towardsdatascience.com/how-does-ai-detect-objects-technical-d8d63fc12881> adresinden alındı
- Anwer, A. M. (2017). *Derin Öğrenme Yöntemleri ile Göğüs Kanseri Teşhisi*. Ankara: Tür Hava Kurumu Üniversitesi Fen Bilimleri Enstitüsü (Yüksek Lisans Tezi).
- Aslantaş, A. (2019, Mart 4). *digitalage*. <https://digitalage.com.tr/makale/yapay-zeka-dijital-pazarlamayi-nasil-degistirecek/>: <https://digitalage.com.tr/makale/yapay-zeka-dijital-pazarlamayi-nasil-degistirecek/> adresinden alınmıştır
- Ayesha Gurnani , Viraj Mavani , Vandit Gajjar , Yash Khandhediya. (2017). Flower Categorization using Deep Convolutional Neural Networks. *arxiv.org*, 4321-4324.
- Cebeci, İ. (2020). *Bilgisayar Görüşü ile Yüz ve Nesne Tanıma | R-CNN, SSD, GANs*. www.udemy.com.
- Ceylan, A. (2020, 05 14). *Nasıl Düşünebildiğimizi Hiç Düşündünüz mü?* medium.com: <https://medium.com/dusunenbeyinler/nas%C4%B1ld%C3%BC%C5%9F%C3%BCnebildi%C4%9Finizi-hi%C3%A7d%C3%BC%C5%9F%C3%BCnd%C3%BCn%C3%BCz-m%C3%BC-56585737b61b> adresinden alınmıştır
- Copeland, M. (2020, 11 01). *What's the Difference Between Artificial Intelligence, Machine Learning and Deep Learning?* <https://blogs.nvidia.com:https://blogs.nvidia.com/blog/2016/07/29/whats-difference-artificial-intelligence-machine-learning-deep-learning-ai/> adresinden alınmıştır

- Çoruh, M. (2018). *İşletmelerde Bilişim Sistemleri Yönetimi*. e-Kitap Projesi.
- Daniel Nkemelu, Daniel Omeiza, Nancy Lubalo. (2018). Deep Convolutional Neural Network For Plant Seedlings Classification. *arxiv.org*, 1-5. *arxiv.org*. adresinden alınmıştır
- Dignan, L. (2020, 12 15). *Top cloud providers 2019: AWS, Microsoft Azure, Google Cloud; IBM makes hybrid move; Salesforce dominates SaaS*. 8 14, 2020 tarihinde *www.zdnet.com*: <https://www.zdnet.com/article/top-cloud-providers-2019-aws-microsoft-azure-google-cloud-ibm-makes-hybrid-move-salesforce-dominates-saas/> adresinden alındı
- Doğan, M. (2020). *Yapay Zeka Felsefesinde Bilinç Problemi*. Ankara: Ankara Yıldırım Beyazıt Üniversitesi Sosyal Bilimler Enstitüsü (Doktora Tezi).
- Gandhi, R. (2018, Haziran 9). <https://towardsdatascience.com>. 12 27, 2020 tarihinde R-CNN, Fast R-CNN, Faster R-CNN, YOLO — Object Detection Algorithms: <https://towardsdatascience.com/r-cnn-fast-r-cnn-faster-r-cnn-yolo-object-detection-algorithms-36d53571365e> adresinden alındı
- Ghtairi, E. S. (2019). *Konvolüsyonel Sinir Ağları Kullanılarak Çiçek Türlerinin Sınıflandırılması*. Konya: Konya Selçuk Üniversitesi Fen Bilimleri Enstitüsü (Yüksek Lisans Tezi).
- Google. (2020). Ocak 2020 tarihinde <https://www.google.co.uk/landing/now/>: <https://www.google.co.uk/landing/now/> adresinden alındı
- google. (2020). *AutoML Vision Edge ile bir nesne algılama modeli eğitin*. 5 27, 2020 tarihinde *firebase.google.com/*: <https://firebase.google.com/docs/ml/train-object-detector> adresinden alındı
- Goyal, L. C., & Jatav, P. K. (2012). Cloud Computing: an Overview and its Impact on. *INTERNATIONAL JOURNAL OF NEXT GENERATION COMPUTER APPLICATIONS*.
- Internet Database Connector (IDC)*. (2019, 09 04). Worldwide Spending on Artificial Intelligence Systems Will Be Nearly \$98 Billion in 2023, According to New IDC Spending Guide: <https://www.idc.com/getdoc.jsp?containerId=prUS45481219> adresinden alınmıştır

- Kerim Kürşat Çevik, Abdülkadir Çakır. (2010). Görüntü İşleme Yöntemleriyle Araç Plakalarının Tanınarak Kapı. *AKÜ Fen Bilimleri Dergisi*, 31-38.
- Kınalıoğlu, İ. H. (2019). Yapay Zeka Tabanlı Yöntemler Kullanılarak Futbol Müsabakalarının Sonuçlarının Kestirilmesi ve Hibrit Model Önerileri. İ. H. Kınalıoğlu içinde, *Yapay Zeka Tabanlı Yöntemler Kullanılarak Futbol Müsabakalarının Sonuçlarının Kestirilmesi ve Hibrit Model Önerileri*. Konya: Yüksek Öğretim Kurumu (YÖK).
- Köksal, T. E. (2019). *Dağ Çiçeklerinin Derinlenmesine Ayırıştırılabilir Evrişimli Yapay Sinir Ağlarıyla Modellenmesi*. Erzurum: Atatürk Üniversitesi Fen Bilimleri Enstitüsü (Yüksek Lisans Tezi).
- Köse, U. (2019). Yapay Zeka ve Geleceğin Siber Savaşları. *Bilim ve Teknik*.
- Kubota, T. (2017, 01 25). *Stanford University*. <https://www.stanford.edu/https://news.stanford.edu/2017/01/25/artificial-intelligence-used-identify-skin-cancer/> adresinden alınmıştır
- LaPlante, A. (2019). *Getting Started with*. oracle.com/data-science.
- Lee, Hyo-Haeng - Hong,Kwang-Seok. (2017). Automatic recognition of flower species in the natural environment. *Image and Vision Computing*, 61.
- Lorica, B. (2019, 07 28). *One simple graphic: Researchers love PyTorch and TensorFlow*. 12 28, 2020 tarihinde [www.oreilly.com:https://www.oreilly.com/content/one-simple-graphic-researchers-love-pytorch-and-tensorflow/](http://www.oreilly.com/https://www.oreilly.com/content/one-simple-graphic-researchers-love-pytorch-and-tensorflow/) adresinden alındı
- Low, C. (2016). [www.engadget.com](http://www.engadget.com/2016-09-29-google-opens-up-its-machine-learning-tricks-to-all.html). [www.engadget.com:https://www.engadget.com/2016-09-29-google-opens-up-its-machine-learning-tricks-to-all.html](http://www.engadget.com/2016-09-29-google-opens-up-its-machine-learning-tricks-to-all.html) adresinden alınmıştır
- Mijwel, M. M. (2016). *Yapay Zeka Nedir?* 5 14, 2020 tarihinde https://www.researchgate.net/https://www.researchgate.net/publication/323292529_Yapay_Zeka_Nedir adresinden alındı
- Mijwel, M. M. (2017, Eylül). *Yapay Sinir Ağlar Yapısı ve Fonksiyonu*. <https://www.researchgate.net/?enrichId=rgreq-ccf1ad9d9dbb587fcdf571512f701fbf->

XXX&enrichSource=Y292ZXJQYWdlOzMzMzA3Mzg2NDtBUzo1OTIxODIxMDQ4Mjk5NTlAMTUxODE5ODYyNDk3Ng%3D%3D&el=1_x_1&_esc=publicationCoverPdf adresinden alınmıştır

Murat Altun, Kemal Kayıkçı, Sezgin Irmak. (2019). Eğitimde Yapay Zeka Kavramı ve Kullanım Alanları. *XI. Uluslararası Eğitim Denetimi Kongresi* (s. 337-353). Antalya: XI. Uluslararası Eğitim Denetimi Kongresi Eğitim, Hukuk ve Yapay Zeka Bildiri Kitabı.

Nur Azida Muhammad, Amelina Ab Nasir, Zaidah Ibrahim, Nurbaity Sabri. (2018). Evaluation of CNN, Alexnet and GoogleNet for Fruit Recognition. *Indonesian Journal of Electrical Engineering and Computer Science*, 468 - 475.

Orhan, S. (2017). *Localization Of Certain Animal Species*. İzmir: İzmir Yüksek Teknoloji Enstitüsü Mühendislik ve Fen Bilimleri Enstitüsü (Yüksek Lisans Tezi).

Öztemel, E. (2006). *Yapay Sinir Ağları*. İstanbul: Papatya Yayıncılık.

Phillip Isola , Jun-Yan Zhu , Tinghui Zhou , Alexei A. Efros. (2018). Image-to-Image Translation with Conditional Adversarial Networks. *arxiv.org*, 1611.07004.

Ryan Kiros, R. S. (2014). Unifying Visual-Semantic Embeddings with Multimodal Neural Language Models. *arxiv.org*, 1411.2539.

Sachan, A. (2018, 02 05). *Zero to Hero: Guide to Object Detection using Deep Learning: Faster R-CNN, YOLO, SSD*. 12 20, 2020 tarihinde [cv-tricks.com: https://cv-tricks.com/object-detection/faster-r-cnn-yolo-ssd/](https://cv-tricks.com/object-detection/faster-r-cnn-yolo-ssd/) adresinden alındı

Sarıtaş, M., & Üner, N. (2013). Eğitimde Yenilikçi Teknolojiler: Bulut Teknolojisi. *Journal of Research in Education and Teaching*.

Sarıtaş, M., & Üner, N. (2013). Eğitimdeki Yenilikçi Teknolojiler: Bulut Teknolojisi. *Journal of Research in Education and Teaching*.

Shi, Y., Rathod, V., & a-googler. (2020, 06 28). *TensorFlow 1 Detection Model Zoo*. 12 27, 2020 tarihinde [https://github.com: https://github.com/tensorflow/models/blob/master/research/object_detection/g3doc/tfl_detection_zoo.md](https://github.com/tensorflow/models/blob/master/research/object_detection/g3doc/tfl_detection_zoo.md) adresinden alındı

- Şeker, Ş. E. (2008, 12 1). *SVM (Support Vector Machine, Destekçi Vektör Makinesi)*.
<http://bilgisayarkavramlari.sadievrenseker.com/>:
<http://bilgisayarkavramlari.sadievrenseker.com/2008/12/01/svm-support-vector-machine-destekci-vektor-makinesi/> adresinden alınmıştır
- Şener, S. (2020, 09 10). *Yapay Zeka, Makine Öğrenimi ve Derin Öğrenme Arasındaki Farklar*. 10 01, 2020 tarihinde www.endustri40.com:
<https://www.endustri40.com/yapay-zeka-makine-ogrenimi-ve-derin-ogrenme-arasindaki-farklar/> adresinden alındı
- Tag: *Single Shot Multibox Detecto*. (2020, 10 13). 12 25, 2020 tarihinde
<https://globalaihub.com/>: <https://globalaihub.com/tag/single-shot-multibox-detector/> adresinden alındı
- Tan, Z. (2019). *Derin Öğrenme Yardımıyla Araç Sınıflandırma*. Elazığ: Fırat Üniversitesi Fen Bilimleri Enstitüsü (Yüksek Lisans Tezi).
- Tekin, B. Y. (2020, 10 13). Tag: *Single Shot Multibox Detector*. 27 12, 2020 tarihinde
https://globalaihub.com: <https://globalaihub.com/tag/single-shot-multibox-detector/> adresinden alındı
- Tekin, B. Y. (2020, 10 3). Tag: *Single Shot Multibox Detector*. <https://globalaihub.com>:
<https://globalaihub.com/tag/single-shot-multibox-detector/> adresinden alınmıştır
- Tekin, B. Y. (2020, 10 13). Tag: *Single Shot Multibox Detector*. 12 25, 2020 tarihinde
https://globalaihub.com: <https://globalaihub.com/tag/single-shot-multibox-detector/> adresinden alındı
- Train an image labeling model with AutoML Vision Edge*. (2020, 04 08).
<https://firebase.google.com/>: https://firebase.google.com/docs/ml-kit/train-image-labeler?authuser=0#train_the_model adresinden alınmıştır
- Tülay Bozüyük, Caner Yağcı, İlhan Gökçe, Görkem Akar. (2005). *Yapay Zeka Teknolojilerinin Endüstrideki Uygulamaları*. İstanbul: Marmara Üniversitesi Teknik Bilimler Meslek Yüksek Okulu Elektrik Programı.
- Türk Dil Kurumu*. (2020, 05 12). Türk Dil Kurumu Sözlükleri: <https://sozluk.tdk.gov.tr/> adresinden alınmıştır
- Tzutalin, D. (2020, 11 15). *tzutalin labellmg*. <https://github.com>:
<https://github.com/tzutalin/labellmg> adresinden alınmıştır

- Uslu, M. (2015). *Yapay Sinir Ağları (YSA) Nedir ?* 11 01, 2020 tarihinde <https://kod5.org>: <https://kod5.org/yapay-sinir-aglari-ysa-nedir/> adresinden alındı
- Weins, K. (2020, 05 21). *Cloud Computing Trends: 2020 State of the Cloud Report*. www.flexera.com: <https://www.flexera.com/blog/industry-trends/trend-of-cloud-computing-2020/> adresinden alınmıştır
- Yannis M. Assael, Brendan Shillingford, Shimon Whiteson ,Nando de Freitas. (2016). LIPNET: END-TO-END SENTENCE-LEVEL LIPREADING. *arvix.org*, 1611.01599.





Zafer KAYA

GÖRÜNTÜ İŞLEME SÜREÇ VE SONUÇLARI
AÇISINDAN GOOGLE VE YEREL MAKİNE
ÖĞRENMELEİNİN KİYASLANMASI

Kütahya Dumlupınar Üniversitesi
Lisansüstü Eğitim Enstitüsü
Bilgisayar Mühendisliği Ana Bilim Dalı



ÖZGEÇMİŞ

Kişisel Bilgiler

Adı Soyadı : Zafer KAYA

Eğitim

Derece	Eğitim Birimi	Mezuniyet Tarihi
Yüksek Lisans	Dumlupınar Üniversitesi Bilgisayar Mühendisliği	
Lisans	Eskişehir Anadolu Üniversitesi Hukuk Fakültesi	
Lisans	Balıkesir Üniversitesi Bilgisayar ve Öğretim Teknolojileri Eğitimi	2011
Lise	Mimar Sinan Lisesi	2004

Uzmanlık Alanları

Görüntü İşleme, Nesne Tanıma, Programlama Dilleri