T.C.
ERCİYES ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ
MEKATRONİK MÜHENDİSLİĞİ ANABİLİM DALI

DERİN ÖĞRENME TABANLI NESNE TANIMA UYGULAMALARI

Hazırlayan
Eda GALİP

Danışman
Doç. Dr. Mehmet Bahadır ÇETİNKAYA

Yüksek Lisans Tezi

Ocak 2024
KAYSERİ

T.C.
ERCİYES ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ
MEKATRONİK MÜHENDİSLİĞİ ANABİLİM DALI

DERİN ÖĞRENME TABANLI NESNE TANIMA UYGULAMALARI

(Yüksek Lisans Tezi)

Hazırlayan
Eda GALİP

Danışman
Doç. Dr. Mehmet Bahadır ÇETİNKAYA

Ocak 2024
KAYSERİ

BİLİMSEL ETİĞE UYGUNLUK

Bu çalışmadaki tüm bilgilerin, akademik ve etik kurallara uygun bir şekilde elde edildiğini beyan ederim. Aynı zamanda bu kural ve davranışların gerektirdiği gibi, bu çalışmanın özünde olmayan tüm materyal ve sonuçları tam olarak aktardığımı ve referans gösterdiğimi belirtirim.

Eda GALİP

İmza

“Derin Öğrenme Tabanlı Nesne Tanıma Uygulamaları” adlı Yüksek Lisans Tezi, Erciyes Üniversitesi Lisansüstü Tez Önerisi ve Tez Yazma Yönergesi’ ne uygun olarak hazırlanmıştır.

Hazırlayan	**Danışman**
Eda GALİP	Doç. Dr. Mehmet Bahadır ÇETİNKAYA
İmza	İmza

Mekatronik Mühendisliği ABD Başkanı

Prof. Dr. Şahin YILDIRIM

İmza

Doç. Dr. Mehmet Bahadır ÇETİNKAYA danışmanlığında **Eda GALİP** tarafından hazırlanan **"Derin Öğrenme Tabanlı Nesne Tanıma Uygulamaları"** adlı bu çalışma jürimiz tarafından Erciyes Üniversitesi Fen Bilimleri Enstitüsü **Mekatronik Mühendisliği** Anabilim Dalında **Yüksek Lisans** tezi olarak kabul edilmiştir.

26.01.2024

JÜRİ:

Danışman : Doç. Dr. Mehmet Bahadır ÇETİNKAYA

Üye : Prof. Dr. Şahin YILDIRIM

Üye : Doç. Dr. Ali DURMUŞ

ONAY:

Bu tezin kabulü Enstitü Yönetim Kurulunun ……...… tarih ve ……..……sayılı kararı ile onaylanmıştır.

……. /……../ ……

Prof. Dr. Emel KIZILKAYA AYDOĞAN

Enstitü Müdürü

TEŞEKKÜR

Tez çalışmam sürecinde bilgi ve tecrübeleri ile bana yol gösteren ve her türlü desteğini esirgemeyen değerli danışman hocam sayın Doç. Dr. Mehmet Bahadır ÇETİNKAYA'ya göstermiş olduğu hoş görü ve sabırdan dolayı sonsuz teşekkürlerimi ve saygılarımı sunarım.

Çalışmalarım esnasında bilgi, tecrübe ve yardımlarını esirgemeyerek akademik gelişimime katkıda bulunan sayın hocam Öğr. Gör. Muhammed İŞÇİ'ye teşekkürlerimi ve saygılarımı sunarım.

Eğitim hayatım boyunca ilgisini ve önerilerini göstermekten kaçınmayan, varlığını her daim hissettiren, tecrübeleri ile yol gösteren sayın hocam Doç. Dr. Mehmet Naci EFE'ye teşekkürlerimi ve saygılarımı sunarım.

Hayatımın her anında yanımda olan, bana maddi ve manevi her ihtiyacımda destek veren, bugünlere gelmemde en büyük emeği olan, hayattaki en büyük şansım sevgili aileme teşekkürlerimi sunarım.

Eda GALİP

Ocak 2024, KAYSERİ

DERİN ÖĞRENME TABANLI NESNE TANIMA UYGULAMALARI
Eda GALİP

Erciyes Üniversitesi, Fen Bilimleri Enstitüsü
Yüksek Lisans Tezi, Ocak 2024
Danışman: Doç. Dr. Mehmet Bahadır ÇETİNKAYA

ÖZET

Bilgisayar destekli görüntü tanıma, yapay zekâ alanında yaşanan gelişmelerle birlikte son yıllarda önemli bir araştırma alanı haline gelmiştir. Derin öğrenme, yüksek boyutlu görüntülerin işlenmesinde doğru sonuçlar üretebilen en sağlam yaklaşımlardan biridir. Bu çalışmada açık kaynak kodlu Keras kütüphanesi ile sinir ağlarının oluşturulması hedeflenmiş ve ayrıca veri akış grafikleri kullanarak numerik hesaplamalar yapan TensorFlow açık kaynak kodlu makine öğrenmesi kütüphanesi üzerinde derin öğrenmeye dayalı analizler gerçekleştirilmiştir. ImageNet veri tabanından yararlanılarak internet tarayıcısı üzerinden 3900 adet görselden oluşan özel bir görüntü veri seti oluşturulmuştur. Oluşturulan bu veri seti üzerinde VGG16, VGG19, ResNet50, ResNet50V2, ResNet101, MobileNet ve DenseNet121 ağ modelleri için derin öğrenme tabanlı analizler gerçekleştirilmiştir. Gerçekleştirilen analizler Doğruluk Oranları ortalaması, Hata Oranları, Eğitim Süreleri ve Tahmin Değerleri ortalaması açısından mukayese edilmiştir. VGG19 ağ modeli 1081 adet nesne görselini doğru, 89 adet nesne görselini ise yanlış tahmin ederek, %97.8475 Doğruluk Oranı'na ve %92.3931 Tahmin Değeri oranına sahip olarak sistemin en başarılı ağ modeli olarak seçilmiştir, ResNet50V2 ağ modeli ise 387 adet nesne görselini doğru, 783 adet nesne görselini ise yanlış tahmin ederek, %45.3272 Doğruluk Oranına ve %33.0768 Tahmin Değeri oranına sahip olarak sistemin en başarısız ağ modeli olarak seçilmiştir. Sonuç olarak, farklı nesne grupları için derin öğrenme algoritmaları kullanılarak yüksek doğrulukta görüntü tanıma sonuçlarının elde edilebildiği görülmüştür.

Anahtar Kelimeler: Yapay zekâ, Derin öğrenme modelleri, Görüntü işleme uygulamaları, Görüntü tanıma uygulamaları

DEEP LEARNING BASED OBJECT RECOGNITION APPLICATIONS
Eda GALİP

Erciyes University, Graduate School of Natural and Applied Sciences
Master Thesis, January 2024
Supervisor: Assoc. Prof. Dr. Mehmet Bahadır ÇETİNKAYA

ABSTRACT

Computer-aided image recognition has become an important research area in recent years with the developments in the field of artificial intelligence. Deep learning is one of the most robust approaches that is able to produce accurate results in processing of high-dimensional images. In this study, it was aimed to create neural networks with the Keras open source library and also deep learning based analyzes were performed on the TensorFlow open source machine learning library which performs numerical calculations by using data flow charts. A special image dataset consisting of 3900 images was created from the ImageNet database via the internet browser. Deep learning based analyzes were carried out for VGG16, VGG19, ResNet50, ResNet50V2, ResNet101, MobileNet and DenseNet121 network models on this data set. The results have been compared in terms of mean accuracy rates, error rates, training times and mean prediction value rates. It was determined that the VGG19 network model predicted 1081 object images correctly and 89 object images incorrectly and it was chosen as the optimal network model of the system, with an accuracy rate of %97.8475 and a prediction value rate of %92.3931. ResNet50V2 network model predicted 387 object images correctly and 783 object images incorrectly and it was chosen as the worst network model of the system, with an accuracy rate of %45.3272 and a prediction value rate of %33.0768. Consequently, it was seen that the image recognition results with high accuracy were able to obtained for different object groups via deep learning algorithms.

Keywords: Artificial intelligence, Deep learning models, Image processing applications, Image recognition applications

İÇİNDEKİLER

DERİN ÖĞRENME TABANLI NESNE TANIMA UYGULAMALARI

BİLİMSEL ETİĞE UYGUNLUK ii

YÖNERGEYE UYGUNLUK iii

KABUL VE ONAY iv

TEŞEKKÜR v

ÖZET vi

ABSTRACT vii

İÇİNDEKİLER viii

KISALTMALAR x

TABLOLAR LİSTESİ xi

ŞEKİLLER LİSTESİ xiii

1. BÖLÜM

GENEL BİLGİLER ve LİTERATÜR ÇALIŞMASI

1.1. Genel Bilgiler 1

1.2. Tezin Amacı ve Önemi 3

1.3. Literatür Çalışması 4

2. BÖLÜM

YÖNTEM VE MATERYAL

2.1.Kavramsal Bilgiler 7

2.1.1. Yapay Zekâ 7

2.1.2. Makine Öğrenmesi 8

2.1.2.1. Denetimli Öğrenme 9

2.1.2.2. Denetimsiz Öğrenme ……..……...10
2.1.3. Derin Öğrenme ..…….…...10
2.1.4. Evrişimsel Sinir Ağları ……..…13
2.2. Yöntem ve Materyaller ……..16
2.2.1. VGG16 Ağ Modeli ……17
2.2.2. VGG19 Ağ Modeli ……18
2.2.3. ResNet50 Ağ Modeli ……..18
2.2.4. ResNet50V2 Ağ Modeli ……18
2.2.5. Resnet101 Ağ Modeli ……19
2.2.6. MobileNet Ağ Modeli ……19
2.2.7. DenseNet121 Ağ Modeli ……19

3. BÖLÜM

BULGULAR

3.1 Araştırma Modeli.....21
3.2 Sınıflandırma Ağ Modelleri 23
3.2.1. VGG16 ……23
3.2.2. VGG19 ……26
3.2.3. ResNet50 ……29
3.2.4. ResNet50V2 ……32
3.2.5. ResNet101 ……...……35
3.2.6. MobileNet ……...38
3.2.7. DenseNet121 ……...41
3.3. Test Verileri, Sonuçları ve Analizleri ……44

4. BÖLÜM

TARTIŞMA-SONUÇ ve ÖNERİLER

SONUÇ……...55

KAYNAKÇA 57

ÖZGEÇMİŞ.....61

KISALTMALAR

DL : Derin Öğrenme (Deep Learning)

CNN : Evrişimsel Sinir Ağları (Convolutional Neural Networks)

GAN : Çekişmeli Üretici Ağ (Generative Adriversal Network)

ML : Makine Öğrenmesi (Machine Learning)

YSA : Yapay Sinir Ağları

GİB : Grafik İşlem Birimleri

DSA : Derin Sinir Ağları

RNN : Tekrarlayan Sinir Ağı (Recurrent Neural Network)

RBM : Kısıtlı Boltmann Makineleri (Restricted Boltzmann Machies)

DBN : Derin İnanç Ağları (Deep Belief Nets)

KNN : K-En Yakın Komşu Algoritması (K Nearest Neighbor)

SVM : Destek Vektör Makineleri (Support Vector Machines)

DO : Drop Out

FC : Tam Bağlantı Katmanı

RGB : Red, Green, Blue

TABLOLAR LİSTESİ

Tablo 2.1. Farklı CNN Mimarileri Katman Sayıları ve Parametre Boyutları …….……20
Tablo 3.1. VGG16 Ağ Modeli Çıkış Parametreleri …….……23
Tablo 3.2. VGG16 Ağ Modeli Parametre Değerleri ……...……24
Tablo 3.3. VGG16 Ağ Modeli Eğitim Adımları ……...……...24
Tablo 3.4. VGG19 Ağ Modeli Çıkış Parametreleri ……..26
Tablo 3.5. VGG19 Ağ Modeli Parametre Değerleri ……26
Tablo 3.6. VGG19 Ağ Modeli Eğitim Adımları ……..27
Tablo 3.7. ResNet50 Ağ Modeli Çıkış Parametreleri ……29
Tablo 3.8. ResNet50 Ağ Modeli Parametre Değerleri ……29
Tablo 3.9. ResNet50 Ağ Modeli Eğitim Adımları ……...30
Tablo 3.10. ResNet50V2 Ağ Modeli Çıkış Parametreleri ……..32
Tablo 3.11. ResNet50V2 Ağ Modeli Parametre Değerleri ……..……32
Tablo 3.12. ResNet50V2 Ağ Modeli Eğitim Adımları ……...……33
Tablo 3.13. ResNet101 Ağ Modeli Çıkış Parametreleri ……..……35
Tablo 3.14. ResNet101 Ağ Modeli Parametre Değerleri ……35
Tablo 3.15. ResNet101 Ağ Modeli Eğitim Adımları ……...36
Tablo 3.16. MobileNet Ağ Modeli Çıkış Parametreleri ……...38
Tablo 3.17. MobileNet Ağ Modeli Parametre Değerleri ……..38
Tablo 3.18. MobileNet Ağ Modeli Eğitim Adımları ……39
Tablo 3.19. DenseNet121 Ağ Modeli Çıkış Parametreleri ……..41
Tablo 3.20. DenseNet121 Ağ Modeli Parametre Değerleri ……41
Tablo 3.21. DenseNet121 Ağ Modeli Eğitim Adımları ……...42
Tablo 3.22. VGG16 Ağ Modeline Ait Test Sonuçları ……..45
Tablo 3.23. VGG19 Ağ Modeline Ait Test Sonuçları ……..46
Tablo 3.24. ResNet50 Ağ Modeline Ait Test Sonuçları ……...47
Tablo 3.25. ResNet50V2 Ağ Modeline Ait Test Sonuçları ……..48
Tablo 3.26. ResNet101 Ağ Modeline Ait Test Sonuçları ……49
Tablo 3.27. MobileNet Ağ Modeline Ait Test Sonuçları ……50
Tablo 3.28. DenseNet121 Ağ Modeline Ait Test Sonuçları ……51

Tablo 3.29. Ağ Modellerine Ait Doğru Tahmin Edilen Görsel Sayılarının Kıyaslanması ……...52
Tablo 3.30. Ağ Modellerine Ait Yanlış Tahmin Edilen Görsel Sayılarının Kıyaslanması …….52
Tablo 3.31. Doğruluk Oranlarının Aritmetik Ortalamasının Kıyaslanması …….53
Tablo 3.32. Tahmin Değerlerinin Aritmetik Ortalamasının Kıyaslanması ……..54
Tablo 3.33. Tahmin Değerlerinin Aritmetik Ortalamasının Pasta Grafikte Gösterilmesi …….54

ŞEKİLLER LİSTESİ

Şekil 1.1. CNN'in farklı katmanlarda nesne ile ilgili oluşturduğu farklı temsiller …….2
Şekil 1.2. CNN'in genel mimarisi ……...……...2
Şekil 2.1. Yapay zekâ ve alt alanları hiyerarşisi gösterimi …….8
Şekil 2.2. Denetimli makine öğrenmesi süreci ……...9
Şekil 2.3. YSA tarafından algılanan yüz özellikleri hiyerarşisi örneği ……11
Şekil 2.4. 2 girişli ve 3 çıkışlı basit bir sinir ağı ve derin sinir ağının gösterimi ……12
Şekil 2.5. Temel bir CNN mimarisinin algoritmik yapısı ……13
Şekil 2.6. 3x3 Evrişimsel katmanın gösterimi ……14
Şekil 2.7. 2x2 Havuzlama işleminin gösterilmesi ……...14
Şekil 3.1. Veri sınıflandırılmasında kullanılan 13 farklı nesneye ait görsel ……21
Şekil 3.2. Veri seti içerisinden alınan örnek RGB görüntüleri ……22
Şekil 3.3. VGG16 ağ modeline ait Doğruluk Oranı grafiği ……25
Şekil 3.4. VGG16 ağ modeline ait Hata Oranı grafiği ……...25
Şekil 3.5. VGG19 ağ modeline ait Doğruluk Oranı grafiği ……28
Şekil 3.6. VGG19 ağ modeline ait Hata Oranı grafiği ……...28
Şekil 3.7. ResNet50 ağ modeline ait Doğruluk Oranı grafiği ……...…31
Şekil 3.8. ResNet50 ağ modeline ait Hata Oranı grafiği ……31
Şekil 3.9. ResNet50V2 ağ modeline ait Doğruluk Oranı grafiği ……34
Şekil 3.10. ResNet50V2 ağ modeline ait Hata Oranı grafiği ……34
Şekil 3.11. ResNet101 ağ modeline ait Doğruluk Oranı grafiği ……37
Şekil 3.12. ResNet101 ağ modeline ait Hata Oranı grafiği ……37
Şekil 3.13. MobileNet ağ modeline ait Doğruluk Oranı grafiği ……40
Şekil 3.14. MobileNet ağ modeline ait Hata Oranı grafiği ……40
Şekil 3.15. DenseNet121 ağ modeline ait Doğruluk Oranı grafiği ……..43
Şekil 3.16. DenseNet121 ağ modeline ait Hata Oranı grafiği ……43

1. BÖLÜM

GENEL BİLGİLER ve LİTERATÜR ÇALIŞMASI

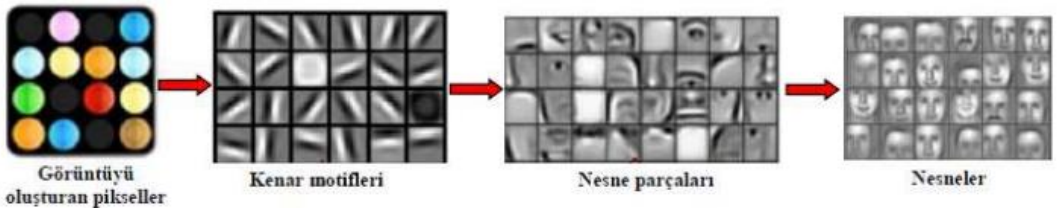
1.1. Genel Bilgiler

Derin Öğrenme (Deep Learning, DL), bir makine öğrenme tekniği olup, özellik ve görevleri doğrudan veri üzerinden öğrenme esasına dayanmaktadır. Sistematik bir yaklaşım olarak, derin öğrenme, yapay zekâ ve makine öğrenmesi alanlarının en temel seviyesinde konumlanmaktadır ve günümüzde yapay zekâ uygulamalarında en yaygın olarak benimsenen yaklaşımdır. Derin öğrenmeyi genel bir perspektifle ele alacak olursak, bu yöntemin temelde veriyi anlamlandırma amacına hizmet ettiği ifade edilebilir [1].

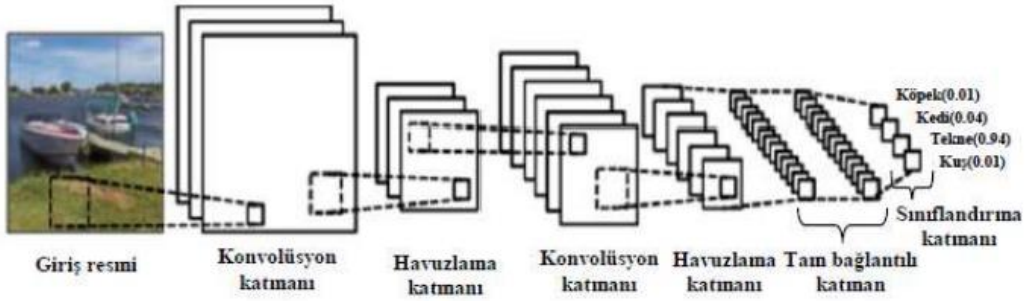
DL yöntemlerinin gelişmesiyle beraber görüntü analizleri, robotik ve otonom sistemler gibi birçok alanda kullanılmaya başlanmıştır. Bu tür problemlerin çözümünde yüksek doğruluk elde edilmesi yaygın bir kullanım alanı oluşmasını sağlamıştır.

Birincil amaç olarak görüntü sınıflandırma üzerine yoğunlaşan Evrişimsel Sinir Ağları (Convolutional Neural Networks, CNN) son yıllardır birçok alanda yoğun bir şekilde kullanılmaya başlanmıştır. Şekil 1.1'de görsel olarak temsil edildiği üzere, piksellerin kenar kombinasyonlarından türetilen desenler, bu desenlerin birleşimiyle birlikte nesne parçalarını meydana getirmekte ve bu parçaların bir araya gelmesiyle de nesnelerin bütününü şekillendirmektedir [1].

Şekil 1.2’de ise CNN’in genel mimarisi gösterilmektedir.



Şekil 1.1. CNN'in farklı katmanlarda nesne ile ilgili oluşturduğu farklı temsiller [2]



Şekil 1.2. CNN'in genel mimarisi [3]

Bu tez çalışmasında nesne tespiti için bir derin öğrenme algoritması koşturulacak olduğundan, CNN yöntemi kullanılması gerekmektedir.

CNN, özellikle görüntü tanıma ve nesne sınıflandırma modellerinin tasarımında kilit bir rol üstlenmektedir. İlk olarak görüntü sınıflandırmaya odaklanan bu ağ türü, geçtiğimiz yıllarda bu temel görevinin ötesine geçerek çeşitli alanlarda etkileyici bir kullanım potansiyeli elde etmiştir.

Bu tez çalışması için yapılan literatür araştırmalarında kullanılan CNN algoritmalarının öğrenme değerleri değişkenlik göstermektedir. Bu değişkenliklere karşın optimum bir öğrenme değerinde veri setleri eğitilmeye çalışılmıştır.

Teknolojinin gelişmesi ile beraber internetin kullanımının artması sonucunda farklı disiplinlere yönelik teknolojik tabanlı akıllı sistemler ortaya çıkmaya başlamıştır. İlk başta akıllı davranış sergileyen canlıların, bilgisayar ortamında modellenmesi söz konusuydu. Günümüzde bu eğilim gerçekleştirilen modellemelere yorumlama ve karar verme mekanizmalarının da entegre edilmesine doğru ilerlemiştir. Bir sistemin bütününü tahmin etmeye yönelik, kısıtlı bir veri kümesi üzerinden uygulamalar yapay zekâ temelli, özellikle robotik sistemlerde yaygın olarak kullanılmaya başlanmıştır.

Bu tez çalışmasında ise, birden çok nesneyi derin öğrenme algoritmalarından 7 farklı ağ modeli (VGG16, VGG19, ResNet50, ResNet5022, ResNet101, MobileNet, DenseNet121) kullanılarak eğitimler gerçekleştirilmiştir. Eğitilen ağ modelleri üzerindeki Doğruluk Oranları (Accuracy), Hata Oranları (Loss), Eğitim Süreleri (Training Time), Tahmin Değerleri (Prediction Value Rates) mukayeseleri yapılarak analizler gerçekleştirilmiştir.

Simülasyon ortamında gerçekleştirilecek olan çalışmada, bir görüntü analiz edilerek bu görüntünün hangi nesneye ait olduğunu geçmiş verileri kullanarak tahmin edebilen sistemlerin geliştirilmesi ve eğitilen ağ modellerinden analizi gerçekleştirilerek en verimli ağ modelinin seçilmesi hedeflenmektedir.

1.2. Tezin Amacı ve Önemi

Sayısal görüntü işleme uygulamalarının temel unsurları nesne tanıma ve nesne tespiti olarak sınıflandırılabilir. Uzun yıllar boyunca nesne tanıma ve nesne tespiti üzerinde çalışmalar yapılmıştır. Nesne tanıma ve nesne tespiti için farklı yöntemler ve farklı algoritmalar geliştirilmiştir. Sayısal görüntülerdeki nesnelerin hızlı tespitini gerçekleştiren ilk algoritma Viola ve Jones tarafından geliştirilmiştir [4]. Yapay Zekâ tekniklerinin ve Görüntü İşleme tekniklerinin gelişmesi ile, yüz tanıma, nesne tanıma, algılama, öğrenme gibi yöntemler geliştirilmiştir. Geliştirilen bu yöntemler robotik, siber güvenlik, savunma, medikal ve daha birçok alanda kullanılmaya başlanmıştır.

Bu tez çalışmasında ise yapay zekâ sistemlerinden derin öğrenme algoritmaları ve görüntü tanıma tekniklerinin bir arada kullanımı gerçekleştirilmiştir. Belirlenmesi zor nesnelerin bilgisayar programları ile tanımlanabilmesi, algılanabilmesi ve veri tabanına aktarımına yönelik çalışmalar yapılmış olup çeşitli ağ modellerinin analizleri gerçekleştirilmiştir. Tez çalışması kapsamında gerçek zamanlı takip, birden fazla objenin ayırt edilebilmesi, görüntüde tek bir obje varsa hareketinin tahmini, bulunan objelerin gösterilmesi ve objenin ne olduğunun DL yöntemleri ile belirlenmiş ve farklı ağ modelleri üzerindeki değişimleri analiz edilmiştir.

Bir nesnenin şekli ve ismi, yapay zekâ teknikleri ile öğrenilip, çözüldükten sonra ona benzer nesneler, yüzler daha kolay ve daha yüksek doğrulukta bulunabilmektedir. Bu

sayede belirli bir giriş veri seti bulunan birçok nesnenin bulunması, tahmin algoritması geliştirilmesi kolaylaşacaktır.

Bu tez çalışmasındaki elde edilen bilgiler, savunma sanayinde, güvenlik ve gizlilik hususunda çalışma yapan portallarda, kolluk kuvvetleri ve güvenlik güçleri tarafından aranan şahısların kamera yoluyla tespitinde kullanılabilecektir. Aynı zamanda virüs, bakteri, patojen madde gibi algılanması birkaç ayrı veri ile mümkün olan biyolojik nesnelerin tespitinde de kullanılabilecektir.

1.3. Literatür Çalışması

Yapay zekâ kavramı, McCharthy tarafından 1950'li yıllarda ilk kez ortaya atılmıştır [5]. İnsan sinir sisteminden ilham alarak geliştirilen yapay sinir ağları, çok katmanlı yapıya sahip denetimli öğrenme algoritması ile Ivakhnenko ve Lapa tarafından 1965 yılında ilk defa literatüre sunulmuştur [6]. Derin öğrenmenin temel mimarisi olan "neocognitron" yapısını öneren Fukushma, denetimsiz bir öğrenme yapısına sahip olması nedeniyle bu yapının önemli bir adım olduğunu belirtmiştir [7]. Veri kümelerinin çeşitlenmesi ve modern derin öğrenme modellerinin güçlenmesiyle birlikte, modelden bağımsız, tamamen veri odaklı yaklaşımların popülerliği artmaktadır. Derin Öğrenme yöntemleri genel mimarileri kullanarak verilerden öğrenen ve özellikle denetlenen problemlerde üstün performans sergileyen yapılardır. Daha fazla parametre ve karmaşık topolojilerin göz önüne alındığı durumlarda, doğru ağırlık kombinasyonlarını bulmak için gereken hesaplama çabası önemli ölçüde artmaktadır. Bu bağlamda, büyük öğrenme problemlerine çözüm sağlayabilen geri yayılım algoritması, derin öğrenme yöntemlerinde oldukça önemli bir yer tutmaktadır [8]. Derin Öğrenme yöntemleriyle ilgili literatürde oldukça çeşitli çalışmalar vardır. Shone ve arkadaşları tarafından Ağ saldırı tespit sistemleri aracılığıyla izinsiz giriş tespiti için bu riskleri gideren yeni bir derin öğrenme tekniği sunulmuştur [9]. Derin öğrenme, tıbbi görüntü işleme alanında hızla evrim geçiren bir teknoloji olup, çeşitli klinik uygulamalarda gösterdiği performans iyileştirmeleri ile dikkat çekmektedir [10]. Söz konusu çalışmada, derin öğrenme yöntemlerinin temel prensipleri ele alınmış ve bu prensiplerin farklı tıbbi uygulamalardaki görüntü bölütlemeye nasıl uygulandığına dair başarılı örneklerle birlikte geniş bir perspektif sunulmuştur. [11] numaralı çalışmada mekânsal ve zamansal

korelasyonları doğası gereği dikkate alan, derin öğrenme tabanlı yeni bir trafik akışı tahmin yöntemi önerilmektedir.

LeCun ve ekibi, 1998 yılında rakamları tanımak ve sınıflandırmak için tasarladıkları LeNet-5 modeli ile elde ettikleri başarılarla birlikte, CNN'in tekrar ön plana çıkmasını sağlamışlardır [12]. Böylelikle literatürde oldukça geniş çaplı çalışmalar gerçekleştirilmiştir. [13] numaralı çalışmada patates yaprağında meydana gelebilecek hastalıkların tespitinde bir CNN modeli kullanılmıştır. Kullanılan algoritmalar sırasıyla AlexNet, VGGNet, ResNet, LeNet ve Sequential modelleridir. [14] numaralı çalışmadaki araştırmanın temel hedefi, kirazların görsel tespitini gerçekleştirmek ve bu meyvelerin sınıflandırılması için etkin bir sistem sunmayı amaçlayan CNN algoritmasını geliştirmek olarak belirlenmiştir.

Özellikle derin öğrenme ve evrişimli sinir ağlarının (CNN'lerin), çeşitli bilgisayarlı görme problemlerinde, nesne algılama ve görüntü sınıflandırma gibi görevlerde oldukça etkili olduğu birçok çalışma tarafından belirtilmiştir. Bu alandaki en başarılı CNN mimarilerinden biri, ImageNet Büyük Ölçekli Görsel Tanıma Yarışması (ILSVRC) 2014'ün kazananı olan Inception modelidir, hem doğruluk hem de hesaplama maliyeti açısından önemli avantajlar sunmaktadır. Görüntü tanıma performansını önemli ölçüde artıran diğer bir başarılı mimari ise ILSVRC 2015'in galibi olan Residual Network (ResNet) mimarisidir.

Bitki hastalıklarının otomatik tespiti, hassas tarım, yüksek verimli bitki fenotipi analizi, akıllı sera yönetimi ve benzeri alanlarda kullanılan otomatik teknikler, günümüzde önemli bir araştırma ve uygulama alanı haline gelmiştir. Bu bağlamda, özellikle bitki hastalıklarının tespiti konusunda geliştirilen otomatik yöntemlere olan talep artmıştır. [15] numaralı çalışmanın odaklandığı konu, 15200 bitki yaprağı görüntüsü içeren geniş kapsamlı bir veri kümesi üzerinde gerçekleştirilen bir ResNet34 modelinin eğitimi ve bu modelin bitki hastalıklarının sınıflandırılması görevini başarıyla yerine getirmesi üzerinedir. [16] numaralı çalışmada ise, yumurtalık tümörlerinin iyi huylu ve kötü huylu olmak üzere iki kategoriye ayrılması için kullanılan derin öğrenme mimarisi olarak, yüksek doğruluk değerine (%97.5) sahip özel bir veri kümesi temel alınarak Inception-ResNet v2 mimarisi önerilmiştir. Aynı zamanda Lin ve arkadaşları Inception-ResNet temelli optimize edilmiş bir versiyon olan CIR-Net adlı bir otomatik kromozom

sınıflandırma yaklaşımı geliştirmişlerdir [17]. Bilgisayar, mobil ve gömülü görme alanlarında kullanılan CNN'ler, nesne algılama ve tanıma konularında önemli birer derin öğrenme modeli olarak öne çıkmaktadır. Bu bağlamda, MobileNet, özellikle hafif ve etkili bir model olmasıyla dikkat çeken bir derin öğrenme yaklaşımı olarak öne çıkmaktadır. Xu ve arkadaşları Akustik sahne sınıflandırması için MobileNet yapısını uygulamışlardır [18]. 2017 yılının sonlarında ise derin öğrenme modellerine yeni bir bakış sunan Hinton ve arkadaşlarının dinamik yönlendirme algoritması ile kapsül ağı modeli tanıtılmıştır [19].

2. BÖLÜM

YÖNTEM VE MATERYAL

2.1. Kavramsal Bilgiler

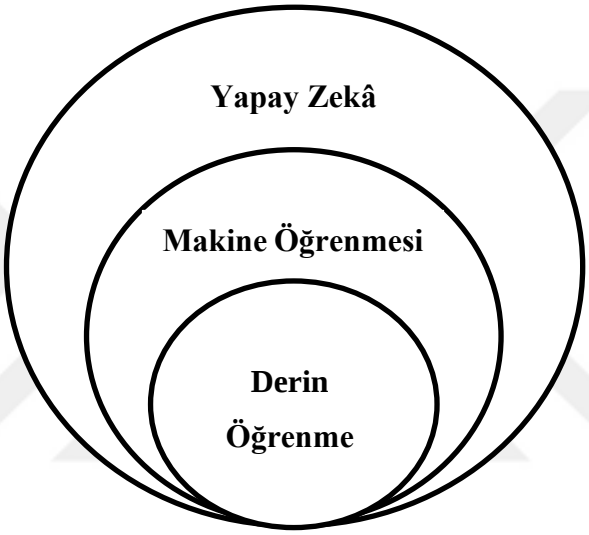
2.1.1. Yapay Zekâ

İnsan zekâsı, çeşitli bilişsel, duygusal ve sosyal yetenekleri içeren karmaşık bir bilişsel kapasitedir. Yapay zekâ ise makinelere zeki davranış kapasitesi kazandırma hedefini gerçekleştirmeye yönelik bir çabadır.

Alan Turing'in 1950'li yıllarda öne sürdüğü düşünebilen makineler konseptinden bu yana, çeşitli ülkelerde insan zekâsına özgü yeteneklerin ve bu yeteneklerle ilişkilendirilen davranışların, örneğin görme, konuşma ve anlama gibi, makine sistemleri tarafından da gerçekleştirilebilmesine dair çalışmalar, yapay zekâ olarak adlandırılmıştır.

İnsan beyni, matematiksel işlemleri zorluk derecelerine bağlı olarak birkaç dakika veya daha uzun bir sürede gerçekleştirebilirken; görmeyi, konuşmayı, anlamayı, yorumlamayı ve kavramayı içeren olayları çok kısa bir sürede başarabilir. Aynı zamanda, edinilen deneyimlerden elde ettiği bilgileri kullanarak, yeni durumlar karşısında adaptif davranış kalıpları geliştirebilir. Örneğin, sıcak bir cisimle temas ettiğinde elinin yanma tecrübesini yaşayan birey, bu bilgiyi sonraki davranışlarına entegre edebilir. Bilgisayarlar, insanlara göre çok daha karmaşık ve üst düzey matematiksel hesaplamaları son derece kısa sürede gerçekleştirme yeteneğine sahiptir. Bununla birlikte, insanlara kıyasla görece daha basit olan görme, konuşma gibi becerilere sahip değillerdir. Yapay zekâ çalışmaları, insanların deneyimlerini edinerek otomatikleştirdiği bu davranışların bilgisayarlar tarafından da gerçekleştirilebilme yeteneğini kazanma amacını taşımaktadır. Yapay zekâ tüm bu alanda yapılan çalışmalara verilen genel bir isimlendirme olmuştur. Şekil 2.1, yapay zekâ ve

onun alt alanlarını genelden özele doğru bir hiyerarşi içinde göstermektedir. Makinelere düşünebilme kabiliyeti kazandırma çabaları genellikle yapay zekâ gelişmeleri olarak adlandırılmaktadır. Ancak, bu çabaların daha spesifik bir alt alanını kapsayan çalışmalar ve algoritmalar, genel olarak makine öğrenmesi (Machine Learning, ML) olarak adlandırılmaktadır. Özellikle, derin öğrenme alt alanında konumlanan ve makine öğrenmesinin en dikkat çekici uygulamalarından birini temsil eden CNN, bu hiyerarşik yapı içinde önemli bir konumu işgal etmektedir.



Şekil 2.1. Yapay zekâ ve alt alanları hiyerarşisi gösterimi

2.1.2. Makine Öğrenmesi

Bilgisayar bilimlerinde, yapay zekanın öğrenme ve tanıma görevlerini yerine getirebilmesi amacıyla geliştirilen uygulamalar, genel olarak ML olarak adlandırılmaktadır. Makine Öğrenmesi, bilgisayarın öğrenme yeteneğini kullanarak, en uygun algoritmaları oluşturarak bilgisayarın öğrendiği bilgiyi genelleme ve tahmin yetenekleriyle kullanabilmesini hedefler. Bu algoritmalar, klasik bilgisayar programlaması için belirli talimatları sıralı olarak uygulama yaklaşımının aksine, eğitim setinde karşılaştığı örneklerden öğrenir ve bu öğrenilen bilgiyi yeni durumlara uyarlayabilir. Bu sayede, algoritmaların geliştirilmesiyle bilgisayarlara öğrenme yeteneği kazandırılması sürecine Makine Öğrenmesi terimiyle atıfta bulunulmaktadır [20].

Geleneksel programlamada, makineye belirli kurallar ve veri setleri sağlanır, ardından belirli bir çıktı elde etmek üzere tasarlanmıştır. Ancak, makine öğrenmesinde, örüntüleri

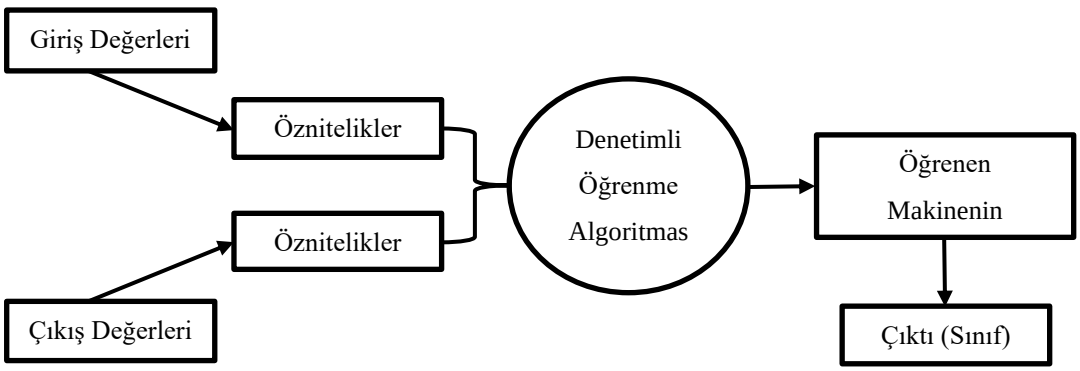
ve verileri makineye sağlarız ve ardından makineden bu verilere dayanarak genel bir kural veya model türetmesini bekleriz, yani makinenin mevcut bilgisi temelinde örüntülerden çıkarımlar yapması beklenir.

2.1.2.1. Denetimli Öğrenme

Denetimli öğrenme, makineye belirli bir problem durumunu deneyimleyip bilinen sonuca yönlendiren süreci, insan öğrenmesiyle benzer bir şekilde benimsetmeyi amaçlamaktadır.

Denetimli öğrenme, makine öğrenimi ve yapay zekanın bir alt kategorisi olarak bilinir. Bu öğrenme türü, etiketli veri kümelerini kullanarak algoritmaları eğiterek verileri sınıflandırma veya doğru tahminlerde bulunma amacını taşır. Giriş verileri modelle beslendikçe, model ağırlıklarını çapraz doğrulama sürecinin bir parçası olarak ayarlar, bu süreç uygun şekilde tamamlandığında model eğitilmiş olur. Denetimli öğrenme, gerçek dünya uygulamalarında, örneğin kuruluşların spam mesajları gelen kutusundan ayrı bir klasöre taşıma gibi çeşitli sorunların geniş kapsamlı çözümüne katkı sağlar.

Makine öğrenmesinde denetimli öğrenme süreci genellikle iki temel problem durumunda kullanılır. Eğer eldeki veri seti sayısal değerlerden oluşuyorsa, yani makine sayısal bir çıktı değeri öğrenmeye çalışıyorsa, bu durumda tahmin çalışması gerçekleştirilir. Ancak, makine kategorik değerleri çıktı olarak öğreniyorsa, sınıflandırma çalışması uygulanır. Denetimli öğrenme sürecinin genel işleyişini görselleştiren Şekil 2.2'de, giriş ve çıkış değerleri öznitelik olarak algoritmayı besler. Algoritma, çıkış değerini bilerek örüntüyü öğrenir ve tahminde bulunur; bu tahminin doğruluğu değerlendirilir ve algoritma geliştirilir.



Şekil 2.2. Denetimli makine öğrenmesi süreci

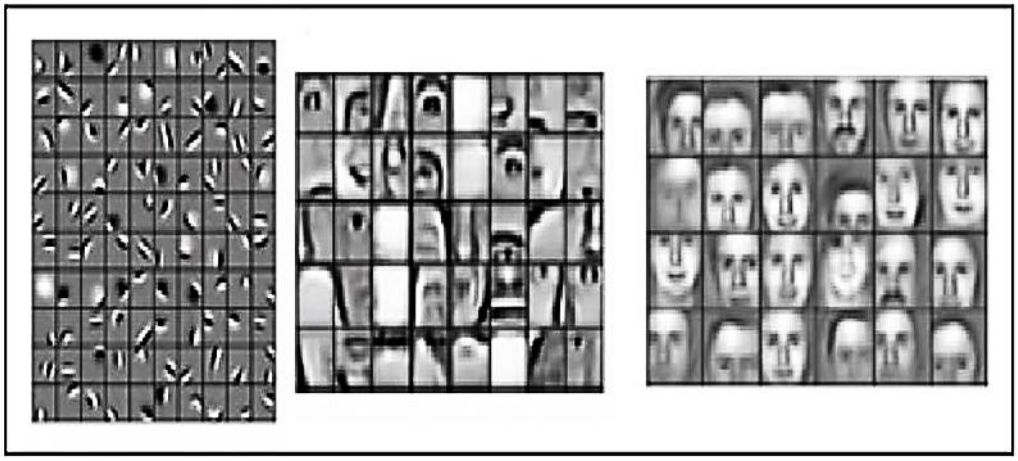
2.1.2.2. Denetimsiz Öğrenme

Denetimsiz makine öğrenmesi algoritmalarında, öğrenme sürecinin bir denetim mekanizması olmaksızın, makineye yalnızca giriş verileri sunulur. Çıkış verileri, etiketsiz bir şekilde temin edilir ve bu algoritmaya dayalı olarak, giriş özniteliklerini kullanarak bilgisayarın öğrenme sürecinde bir örüntü geliştirmesi ve nihayetinde bu örüntüye uygun bir çıktı üretmesi beklenir. Bu bağlamda, genellikle kategorize edilmemiş bir veri kümesi algoritmaya sunularak, verilerin anlamlı bir bütün olarak kategorize edilmiş çıktısı elde edilir; bu sürece kümeleme adı verilmektedir.

2.1.3. Derin Öğrenme

Günümüzde, modern teknikler kullanılarak özellik çıkartımı süreçleri otomatik bir biçimde icra edilmektedir. İllüstrasyon amacıyla, bir DL algoritması tarafından gerçekleştirilen yüz tespiti örneğinde, başlangıçta, bir veri kümesi üzerinde kenarların belirlenmesi adımı yapılmaktadır. Ardından, ikinci aşamada, burun, gözler gibi çizgilerin saptanması işlemi icra edilir. Son adımda ise, yüz gibi karmaşık yapıya sahip nesnelerin tanımlanması gerçekleşir. Her bir katman, önceki katmanın temsil ettiği veriye bağlı olarak eğitilmektedir. Bu katmanlar uygun bir şekilde birleştirildiğinde, yüz özellikleri hiyerarşik bir yapı içinde öğrenilmektedir.

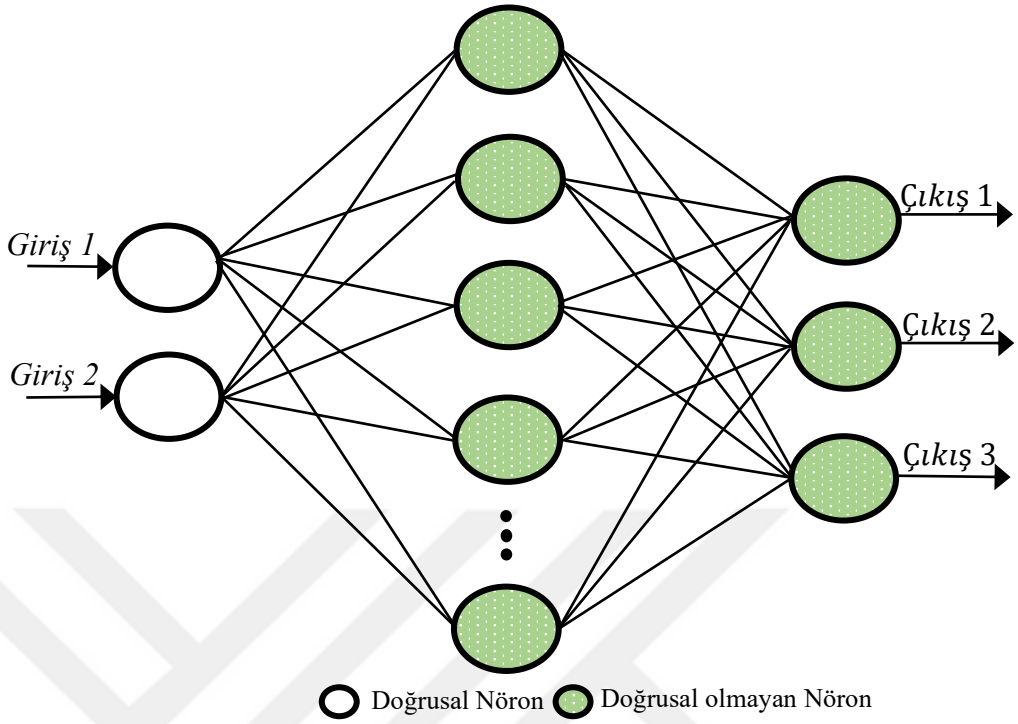
Derin öğrenmede öne çıkan çalışmalar bilgisayar görüsü problemleri temel alınarak başlamıştır. Büyük veri ve hesaplama gücünün gelişmesi ile CNN kullanımı ilgi çekici hale gelmiştir. CNN'den oluşan bu mimariler; nesne tespiti, sınıflandırma ve segmentasyon gibi bilgisayar görüsü görevlerinde yüksek performans göstermiştir. Bu modeller sonuç itibariyle birer sinir ağı olsa da klasik Yapay Sinir Ağları (YSA)'na göre daha karmaşık ve veriye ait önemli ayırt edici özelliklerin otomatik olarak öğrenilmesi konusunda çok daha başarılı olmuşlardır. CNN tarafından algılanan yüz özellikleri hiyerarşik olarak öğrenilmesinde katmanlardaki görselleştirilmiş örneği Şekil 2.3'te görülmektedir.



Şekil 2.3. YSA tarafından algılanan yüz özellikleri hiyerarşisi örneği

DL'nin farklı alanlarda artan kullanımının sebebi, donanımsal uygunluk (Grafik İşlem Birimleri, GİB), büyük veri, yeni algoritmalar, bulut sağlayıcılar ve yazılım araçlarındaki gelişmeler olmuştur [21, 22]. Buna ek olarak, büyük şirketlerin, DL konusuna yeni katılan bireyler ve gruplarla, kendi problem setlerinde kullanılmak üzere oluşturdukları modelleri paylaşma eğiliminde oldukları gözlemlenmektedir ki bu modeller genellikle büyük veri setleri üzerinde eğitilmiştir.

Derin Sinir Ağları (DSA), Şekil 2.4'te görüldüğü gibi çok katmanlı algılayıcılardaki gizli katman sayısının artmasıyla ortaya çıkmıştır. Gizli katman sayısı ve katmanlarda kullanılan nöron sayısı arttıkça, modelin derinlik özelliği de artmaktadır. DL'nin anlaşılması için temeli olan yapay sinir ağlarının anlaşılması önem arz etmektedir.



Şekil 2.4. 2 girişli ve 3 çıkışlı basit bir sinir ağı ve derin sinir ağının gösterimi

Veri bilimcileri ve geliştiriciler, büyük çaplı ve kompleks veri kümelerini değerlendirerek, genellikle insanlardan daha hızlı ve daha doğru bir şekilde metin, ses veya görüntü verileri üzerinde karmaşık ve doğrusal olmayan görevleri gerçekleştirmek amacıyla DL yazılımlarını kullanmaktadır [23].

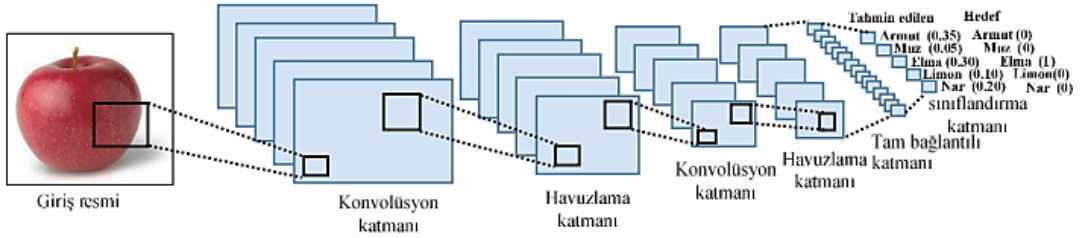
DL mimarilerinin inşası sırasında kullanılacak algoritmaların seçimi büyük bir öneme sahiptir. Bu algoritmalar, veri türü, boyutu, yapısı, hacmi ve kullanılacak parametreler gibi faktörlere bağlı olarak değişiklik gösterir. Başlıca kullanılan algoritmalar şunlardır:

- Evrişimsel Sinir Ağları yani ESA Convolutional Neural Network — CNN,
- Tekrarlayan Sinir Ağı yani Recurrent Neural Network-RNN,
- Kısıtlı Boltzmann Makineleri yani Restricted Boltzmann Machines-RBM
- Derin İnanç Ağları yani Deep Belief Nets-DBN

Bu aşamadan sonra tez kapsamında faydalanılan CNN algoritması anlatılmaya devam edilecektir.

2.1.4. Evrişimsel Sinir Ağları

DL algoritmaları arasında en yaygın olarak bilineni CNN'dir, özellikle görüntü sınıflandırma problemlerinde yaygın bir şekilde kullanılmaktadır. CNN, her bir katmanda genellikle 3x3, 5x5, 7x7 gibi boyutlara sahip filtrelerin (çekirdeklerin) matris üzerinde işlem gördüğü bir yapıya sahiptir. Ardından, bu filtrelerin çıktıları üzerinde havuzlama (pooling) işlemi uygulanmaktadır. Havuzlama, çekirdek içerisindeki verileri filtreleyerek matrisi yeniden ölçeklendirir. Sıkça tercih edilen havuzlama yöntemi genellikle "max pooling" olarak adlandırılmakta olup, bu yöntemde matris içindeki en yüksek değer seçilmektedir. Şekil 2.5'te CNN mimarisinin algoritmik yapısı gösterilmiştir.



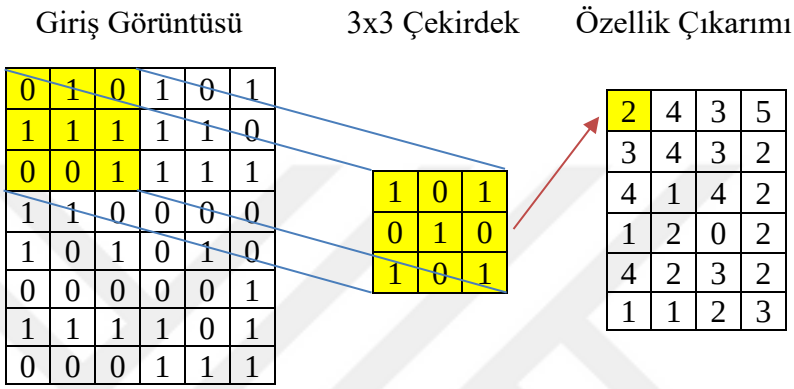
Şekil 2.5. Temel bir CNN mimarisinin algoritmik yapısı

CNN, Derin Öğrenme sinir ağı mimarilerinden biridir ve Bilgisayarla Görme alanında yaygın bir şekilde uygulanır. Bilgisayarlı görme, bir bilgisayarın görsel verileri anlama ve yorumlama yeteneğini geliştirmeyi amaçlayan bir Yapay Zekâ alt alanıdır.

Makine Öğrenimi bağlamında, Yapay Sinir Ağları başarılı bir performans sergileyen önemli bir paradigma sunmaktadır. Sinir Ağları, çeşitli veri türlerini işlemek için, örneğin görüntü, ses ve metin verileri üzerinde etkili bir şekilde kullanılabilmektedir. Farklı türdeki Sinir Ağları, belirli amaçlar için özelleştirilmiştir; örneğin, Tekrarlayan Sinir Ağları (LSTM), sözcük dizilerini tahmin etmek amacıyla kullanılırken, Evrişimli Sinir Ağları, özellikle görüntü sınıflandırması gibi görsel veri analizleri için uygundur.

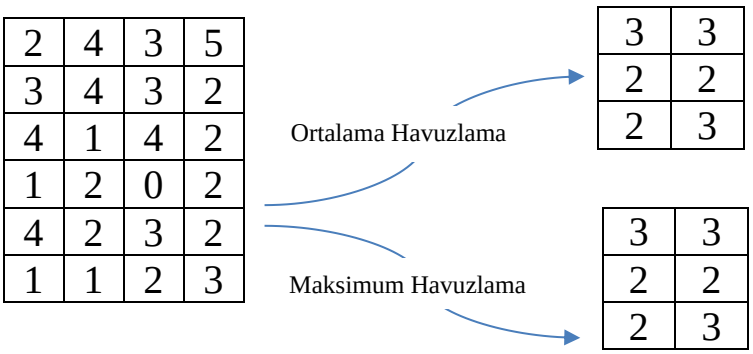
CNN'nin işlemlerini gerçekleştirdiği görüntü, matris formatında temsil edilmektedir. Örneğin, Şekil 2.6'da 6x8 boyutunda bir görüntü matrisi ve 3x3 boyutunda bir çekirdek matrisi görülmektedir. Filtre uygulaması sayesinde, belirli özellikler üzerinde karşılaştırmalar yapılarak görüntüden anlamlı veriler elde edilmektedir. Algoritma tarafından otomatik olarak seçilen kütüphaneler, çekirdek filtrelerini belirli parametrelere göre ayarlayabilmektedir. Parametrelerin doğru bir şekilde tanımlanması ise kritik bir

öneme sahiptir. Bu işlem, görüntü üzerinde belirli özelliklerin vurgulanmasını amaçlamaktadır. Bu süreç sonunda, görüntünün hangi sınıfa ait olduğu olasılıkla tespit edilir. Bu işlemler, diğer görüntüler üzerinde de benzerlikler kurularak genelleştirilmeye çalışılır. Belirlenen filtre, görüntü üzerinde kaydırılarak resmin tamamına uygulanır ve bu kaydırma işlemi, belirli bir adım sayısına göre gerçekleştirilir, bu adıma "stride" denilmektedir [24].



Şekil 2.6. 3x3 Evrişimsel katmanın gösterimi

Havuzlama katmanında, düzenli aralıklarla CovNet'lere eklenir ve temel işlevi hacim boyutunu azaltmaktır. Bu işlem, hesaplamaların hızlanmasına, bellek kullanımının azalmasına ve aşırı uyumu önlemeye yönelik avantajlar sağlar. Havuzlama katmanları genellikle maksimum havuzlama ve ortalama havuzlama olmak üzere iki yaygın türü içerir.



Şekil 2.7. 2x2 Havuzlama işleminin gösterilmesi

İki fonksiyonun (f & g) sonlu aralıktaki [0, t] evrişimi 2.1 numaralı formülde gösterilmiştir;

$$[f * g](t) = \int_0^t f(\tau) g(t - \tau)\, d_\tau \tag{2.1}$$

Burada $[f * g](t)$, f ve g fonksiyonlarının evrişimi anlamına gelmektedir.

Evrişim daha çok sonsuz bir aralıkta alınır, formül 2.2 ve 2.3'te gösterilmiştir [25]:

$$[f * g](t) = \int_{-\infty}^{\infty} f(\tau) g(t - \tau)\, d_\tau \tag{2.2}$$

$$= \int_{-\infty}^{\infty} g(\tau) f(t - \tau)\, d_\tau \tag{2.3}$$

Görüntü tanıma, görüntü sınıflandırma ve bilgisayarlı görme uygulamaları için, özellikle büyük veri setleri mevcut olduğunda son derece doğru sonuçlar sağlayabilen Evrişimli Sinir Ağları (CNN'ler) önemli bir rol oynamaktadır. CNN'ler, nesne verilerini çok katmanlı bir yapı içerisinde işlerler ve bu süreçte nesnenin özelliklerini ardışık iterasyonlarla öğrenirler. Bu doğrudan ve derin öğrenme yaklaşımı, manuel özellik çıkarma (özellik mühendisliği) ihtiyacını ortadan kaldırarak, özellikle büyük veri setlerinde yüksek doğruluk elde etme potansiyeli sunmaktadır.

Temel avantajlardan bir diğeri ise, parametre paylaşımıdır. CNN ile öğrenilen parametreler bir sonraki katmana girdi olarak paylaşılır. Bu sayede, aynı ağırlıklar katmandan tekrar kullanılabilir ve bu durum katmanlarda tekrar öğrenmeye ihtiyaç duyulmadığı anlamına gelir. Bu, daha sonraki katmanların daha karmaşık özellikleri öğrenmesine olanak tanır [26].

Hiper-parametreler ise, modeli tasarlayan kişinin belirlediği, probleme, veri setine ve diğer çeşitli faktörlere bağlı olarak değişen parametrelerdir [27].

Oluşturulan modelde kullanılacak hiper-parametreler, derin öğrenme uygulamalarındaki temel problemlerden biridir. Bu parametrelerin ve değerlerinin seçimi, veri türüne, programcının tecrübesine, yeni çalışmalara, farklı uygulamalarda elde edilen sonuçlara vb. bağlı olarak değişir. Model tasarlanırken kullanılabilecek temel hiper-parametreler şunlardır:

- Ağda Kullanılacak Çekirdek Boyutu
- Parti Büyüklüğü Sayısı (Batch-size)
- Çevrim Sayısı
- Katman Sayısı
- Kullanılacak Aktivasyon Fonksiyonu
- Optimizasyon Algoritması
- Öğrenme Oranı (Learning Rate)
- Seyreltme (Drop Out) Değeri
- Seyreltme Kullanılacak Yerler
- Veri Setinin Büyüklüğü

Bu aşamadan sonra, tez aşamasında kullanılan CNN ağ modelleriyle ilgili bilgiler sunulacaktır.

2.2. Yöntem ve Materyaller

DL yöntemi, yapay zekanın alt alanlarından birini temsil etmektedir. Bu metot, yapay sinir ağlarının geliştirilmesi yoluyla matematiksel ve istatistiksel hesaplamaların gerçekleştirilerek, probleme ait verilerin işlenmesini içermektedir. İşlenen veriler, yapay sinir ağının en üst katmanında çeşitli metrikler kullanılarak bir tahmine dönüştürülür. Bu tahminler, çeşitli değerlendirme metrikleriyle analiz edilerek, tahminleme sürecinin başarısı ölçülmektedir [28]. Yapay sinir ağlarının eğitimi, aynı zamanda model eğitimi olarak bilinir ve bu süreç, problem türüne bağlı olarak değişkenlik gösterir. Bu tezde, özellikle bilgisayarlı görü alanına odaklanılmıştır. Literatürde "CNN" olarak adlandırılan yapay sinir ağlarından türeyen algoritmalar ve mimariler, görüntü tabanlı problemlerin çözümünde yaygın olarak kullanılmaktadır.

CNN, giriş değeri olarak bir görüntü alır. Sinir ağına sunulan bu görüntü, sinir ağının mimarisine tabi olarak katmanlar arasından geçirilerek, hesaplamalar gerçekleştirilir.

Bu çalışmada, açık kaynak kodlu kütüphaneden oluşan Keras Kütüphanesi ile sinir ağları oluşturulması hedeflenmiş, veri akış grafikleri kullanarak numerik hesaplamalar yapan

TensorFlow kütüphanesi üzerinde çalıştırılmıştır. Bilgisayarlı görüntü sistemlerinde etkin çözümler üreten nesne algılama uygulamaları analiz edilmiştir.

Veri setinin hazırlanması esnasında ImageNet görsel veri tabanından ve internet tarayıcısından faydalanılmış olup, toplamda 3900 adet görsel kullanılmıştır.

Audi, Papatya, Karahindiba, İnsan, Hyundai Crete, Mahindra Scorpio, Rolls Royce, Gül, Ay Çiçeği, Swift, Tata Safari, Toyota İnnova ve Lale görseli kullanılarak toplam 13 farklı veri sınıfına ayrılmıştır. Her bir veri sınıfında 300 adet görsel bulunmaktadır.

Eğitimini yaptığımız yapay sinir ağı olan evrişimsel sinir ağlarından elde edilen çıktı, 13 adet sınıftan birisi olacaktır.

Hazırlanan veri seti 7 farklı ağ modeli (VGG16, VGG19, ResNet50, ResNet5022, ResNet101, MobileNet, DenseNet121) ile eğitilmiştir.

Sınıflandırma işleminin ardından elde edilen sonuçlar belirli bir yüzdelik oranına sahip olmak zorundadır. Bu yüzdelik oran bize tahmin işleminin ne kadar başarılı ya da başarısız olduğunu kıyaslamasında öneme sahiptir.

Başarım oranları farklı metrik ölçüm yöntemleri barındırabilir. Üzerinde çalıştığımız modeller ‘Doğruluk Oranları’, ‘Hata Oranları’ ve ‘Eğitim Süreleri’ ve ‘Tahmin Değerleri’ metrikleri ile kıyaslanmıştır.

Eğitim esnasında kullanılan CNN Ağ modelleri şu şekildedir;

2.2.1. VGG16 Ağ Modeli

VGG-16, temel bir sinir ağı modelidir ve diğer modellerden önemli bir farkı, evrişim katmanlarının 2'li veya 3'lü olarak kullanılmasıdır. Tam bağlantı (FC) katmanında 7x7x512=4096 nöronlu bir öznitelik vektörüne dönüştürülür. İki FC katmanı çıkışında, 1000 sınıflı softmax başarımı hesaplanır. Bu modelin toplamda yaklaşık 138 milyon parametresi bulunmaktadır. Diğer modellerde olduğu gibi, girişten çıkışa doğru matrislerin yükseklik ve genişlik boyutları azalırken derinlik değeri (kanal sayısı) artmaktadır.

2.2.2. VGG19 Ağ Modeli

VGG19, 16 konvolüsyon katmanı ve 3 tam bağlı katmandan oluşmaktadır. VGG19'da, 5 havuzlama ve son katman olarak softmaxtan oluşmaktadır. Yaklaşık 144 milyon parametre içermektedir.

2.2.3. ResNet50 Ağ Modeli

ResNet50, ImageNet veri kümesi üzerinde eğitilmiş olan ve 50 katmanlı bir derin öğrenme ağıdır. ImageNet, görüntü tanıma yarışmaları için oluşturulan 20 binden fazla kategoriye ait 14 milyondan fazla resmi içeren bir görüntü veri tabanını temsil etmektedir. ResNet50'nin mimarisi, evrişimli sinir ağlarının geliştirilmiş bir versiyonunu sunmaktadır. Geleneksel ardışık ağ mimarisinden ayrılan bu yapı, başarım oranlarını artırmak amacıyla bazı katmanlar arasındaki geçişlerdeki değişimleri dikkate almadan bir sonraki alt katmana geçiş yapmaktadır. Bu model, derin ağlar yakınsamaya başladığında ortaya çıkan evrişimli sinir ağlarının performans düşümü sorununu çözmeyi hedeflemektedir. ResNet50 mimarisinde, 177 katmanlı bir ağ bulunmakta olup, katmanlar arası bağlantıların konfigürasyonu belirli bir yapıya sahiptir [29]. Giriş katmanı, 224x224x3 boyutunda bir görüntüyü işlemek üzere tasarlanmıştır.

2.2.4. ResNet50V2 Ağ Modeli

ResNet50V2 modeli, dört blok ve toplamda 50 katmandan oluşan bir derin öğrenme mimarisini içermektedir. Bu bloklar, sırasıyla B1, B2, B3 ve B4 olarak adlandırılmıştır. B1 bloğu içerisinde 3 adet, B2 bloğu içerisinde 4 adet, B3 bloğu içerisinde 6 adet ve B4 bloğu içerisinde 3 adet olmak üzere toplam 16 adet birbirine bağlı blok bulunmaktadır. Her bir blokta 3 adet evrişim katmanı yer almaktadır, bu da toplamda 48 adet evrişim katmanını bloklar içinde barındırmaktadır. Giriş katmanını takip eden, boyutları 7x7 ve kaydırmalı olan 2 evrişim katmanı ile birleştirilen 1000 nöronlu tam bağlı katman, modelin toplamda 50 adet katman içermesini sağlamaktadır. Bir bloktan diğerine geçişlerde girdi boyutları yarıya düşerken, özellik haritası sayısı iki katına çıkmaktadır.

2.2.5. ResNet101 Ağ Modeli

3 katlı blok kullanarak 101 katlı ve 152 katlı ResNet'ler oluşur. 50/101/152-katmanlı ResNet'ler, 34-katmanlı olanlardan önemli ölçüde daha iyidir. Ağlar aracılığıyla, bozulma sorunlarına karşı etkin bir önlem alınarak, artan derinlik seviyeleriyle elde edilen doğruluk değerleri önemli ölçüde artmıştır.

2.2.6. MobileNet Ağ Modeli

Google tarafından sunulan ve sınıflandırıcıların eğitimi amacıyla geliştirilen bir bilgisayarlı görme modeli olan MobileNet, parametre sayısını önemli ölçüde azaltmak için derin evrişim katmanlarını kullanır. Bu sayede, hafif bir derin sinir ağı elde edilir. MobileNet, Tensorflow'un ilk mobil bilgisayarlı görme modeli olarak dikkat çeker. Derinlik bakımından ayrılabilir evrişimlerde her bir girdi kanalına tek bir filtre uygulanır. Ardından noktasal evrişimlerle (1 × 1 evrişimler) derinlik bazlı katmanların çıktısının doğrusal kombinasyonu oluşturulur. Derinlik bakımından ayrılabilir katmanlar, tipik evrişim katmanlarının işlevini neredeyse taklit eder; ancak işlem daha hızlı ve küçük bir farkla gerçekleştirilir. Bu yaklaşım, model boyutunu minimuma indirir ve hesaplama gücü taleplerini azaltır. Softmax katmanını besleyen tam bağlantılı katman haricinde, tüm katmanlardan sonra bir batch normalleştirme ve ReLU doğrusal olmayan aktivasyon katmanı gelir. MobileNet, derinlikli ve noktasal evrişimler dışında 28 katmana sahiptir. MobileNet mimarisi, derinlemesine ayrılabilir evrişim işlemini kullanarak parametre sayısını, işlem hacmini ve model karmaşıklığını azaltmış ve bu sayede donanımsal olarak kısıtlı olan mobil ve gömülü cihazlarda kullanılabilir hale getirilmiştir [30].

2.2.7. DenseNet121 Ağ Modeli

DenseNet modeli, girdi olarak 224 x 224 boyutundaki görüntüleri kabul eder ve yaklaşık 7 milyon eğitilebilir parametre içerir. DenseNet121, eğimin kaybolma sorununu aşarak yeterli derinlikte ağların oluşturulmasına olanak tanırken, diğer ağlara kıyasla daha az parametreye sahip olması sayesinde hızlı çalışabilme avantajı sağlar. Geleneksel bir ileri beslemeli Evrişimli Sinir Ağı (CNN), her evrişimli katmanın (ilk katman haricindeki) çıktısını, önceki evrişimli katmandan alarak bir sonraki evrişimli katmana aktaran bir yapıya sahiptir. Bu yapı, her katman için birbirine doğrudan bağlantılar sağlar. Dolayısıyla, "L" katmanı için "L" adet doğrudan bağlantı bulunur; her biri bir önceki katman ile bir sonraki katman arasında yer alır. Genel olarak ResNet mimarisine benzer

bir yapı kullanılmakla birlikte, DenseNet mimarisinde bir katmandan çıkan özellikler, alt katmanların tamamına giriş olarak sağlanır [31].

Tablo 2.1.'de Farklı CNN mimarilerinin katman sayıları ve parametre boyutları gösterilmiştir.

Tablo 2.1. Farklı CNN Mimarileri Katman Sayıları ve Parametre Boyutları

CNN Ağ Modeli	**Katman Sayısı**	**Parametreler (Milyon)**	**Boyut (MB)**
AlexNet	8	60	-
VGGNet-16	23	138	528
VGGNet-19	26	143	549
Inception-V1	27	7	-
Inception-V3	42	27	93
ResNet-152	152	50	132
ResNet-101	101	44	171
InceptionResNetV2	572	55	215
MobileNet-V1	28	4.2	16
MobileNet-V2	28	3.37	14
EfficientNet B0	-	5	-

3. BÖLÜM

BULGULAR

3.1 Araştırma Modeli

Bu çalışmada, her birisi derin öğrenme uygulamalarına yönelik geliştirilmiş olan açık kaynak kodlu Tensorflow ve Keras kütüphaneleri kullanılmıştır.

Tensorflow, 2015 yılında Google tarafından geliştirilen ve çok sayıda renkli görüntü içeren bir kütüphanedir [32].

Keras kütüphanesi, içerdiği çok sayıda işlevsel fonksiyon sayesinde etkin derin öğrenme modelleri oluşturmamızı ve bunları eğitmemizi sağlayan bir kütüphanedir [33].

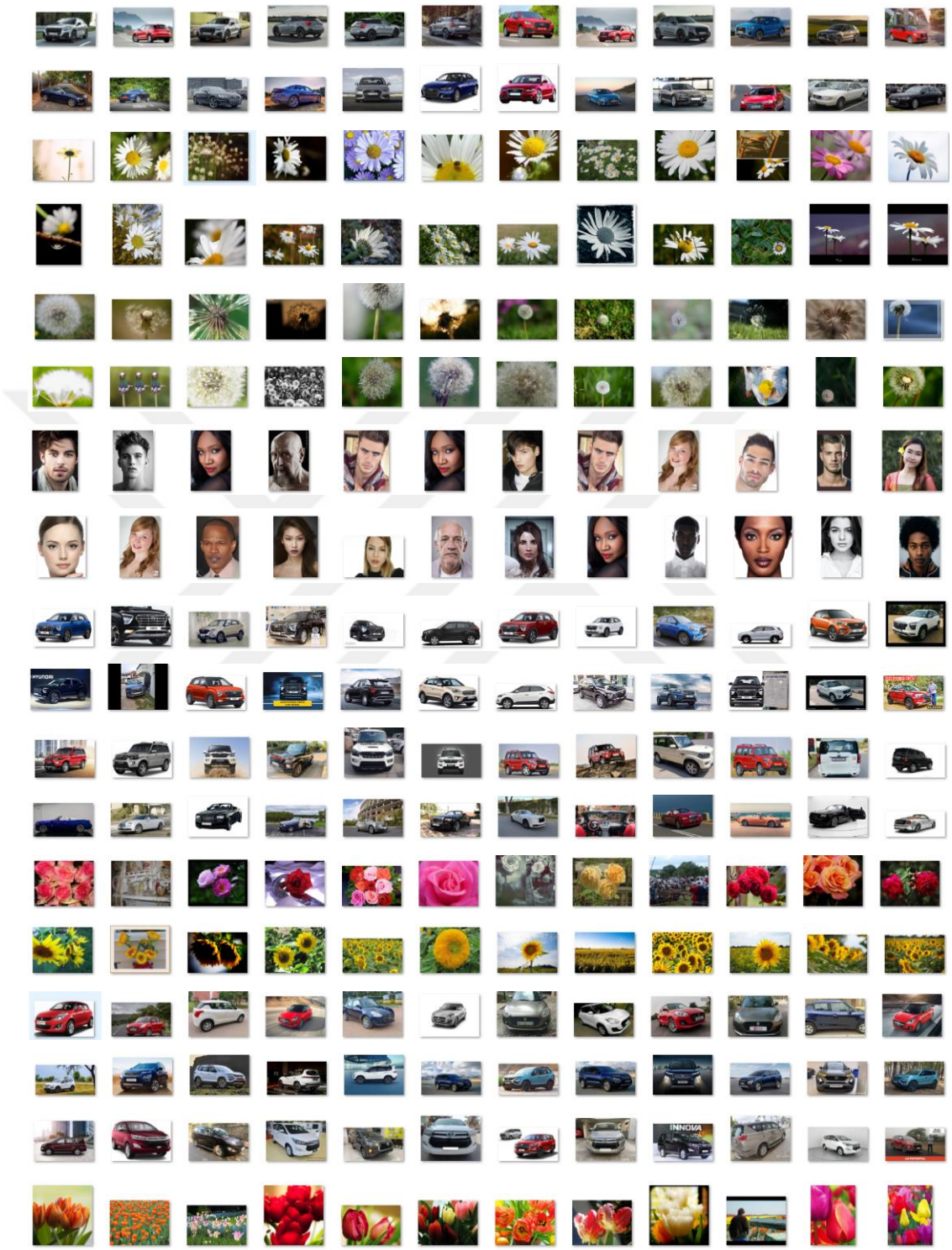
Bu çalışma kapsamında, veri seti olarak ImageNet veri setinden faydalanılarak internet tarayıcısından toparlanmış özel bir veri seti hazırlanmıştır.

Audi, Papatya, Karahindiba, İnsan, Hyundai Creta, Mahindra Scorpio, Rolls Royce, Gül, Ay Çiçeği, Swift, Tata Safari, Toyota İnnova ve Lale görseli kullanılarak 13 farklı veri sınıfı (Şekil 3.1) oluşturulmuştur. Oluşturulan hep bir veri sınıfı için 300 adet görsel hazırlanmış olup, 13 adet veri sınıfı toplamında 3.900 adet görsel kullanılmıştır.



Şekil 3.1. Veri sınıflandırılmasında kullanılan 13 farklı nesneye ait görsel

Veri seti içerisinde bulunan bazı örnek RGB görüntüler Şekil 3.2 ile gösterilmiştir.



Şekil 3.2. Veri seti içerisinden alınan örnek RGB görüntüleri

3.2 Sınıflandırma Ağ Modelleri

Bu tez çalışması kapsamında gerçekleştirilen derin öğrenme yapısında öğrenme veya sınıflanma ağ modelleri olarak VGG16, VGG19, ResNet50, ResNet5022, ResNet101, MobileNet ve DenseNet121 seçilmiştir.

Derin öğrenme algoritmalarının konfigürasyonu ve yürütülmesi aşamalarında, bilgisayar donanımı ve yazılım özellikleri hayati bir rol oynamaktadır. Bu çalışmada, HP ProBook 450 G8 Notebook sistem modeline sahip; Intel(R) Iris(R) Xe Graphics 32 bit ekran kartı, Intel(R) Core(TM) i5-1135G7 @ 2.40GHz (8 CPUs) işlemci, 8 GB RAM, 64 bit işletim sistemine sahip bir bilgisayar kullanılmıştır. Bu bağlamda, tüm simülasyonlar, Windows 10 işletimi üzerinde kurulan yüksek performanslı Visual Studio Code ortamında Python programlama dilinde geliştirilmiş ve eğitilmiştir. Ağ modellerinin eğitimi sırasında Önceden Eğitilmiş Model (pre-trained model) kullanımı nedeniyle, parametrelerin tamamı eğitilmemiştir; bununla birlikte, çıkış katmanlarını oluşturan parametreler eğitimden geçirilmiştir.

3.2.1. VGG16

Ağ modelinin eğitilmesi esnasında önceden eğitilmiş model kullanılmış olup, çıkış katmanlarını oluşturan parametreler eğitilmiştir. Tablo 3.1'de katman türü (Layer Type), çıkış parametreleri (Output Shape) gösterilmektedir.

Tablo 3.1. VGG16 Ağ Modeli Çıkış Parametreleri

Katman Türü	Çıkış	Parametreler
VGG16 (Functional)	(–, 512)	14.714.688
Flatten (Flatten)	(–, 512)	0
Dense (Dense)	(–, 512)	262.656
Dense_1 (Dense)	(–, 1024)	525.312
Dense_2 (Dense)	(–, 512)	524.800
Dense_3 (Dense)	(–, 13)	6.669

VGG16 ağ modelinin eğitilmesinde Toplam Parametreler (Total Params): 16.034.125, Eğitilebilir Parametreler (Trainable Params): 1.319.437 şeklinde olup parametre verileri Tablo 3.2'de gösterilmiştir.

Tablo 3.2. VGG16 Ağ Modeli Parametre Değerleri

Toplam Parametreler	16.034.125 (61.17 MB)
Eğitilebilir Parametreler	1.319.437 (5.03 MB)
Eğitilemeyen Parametreler	14.714.688 (56.13 MB)

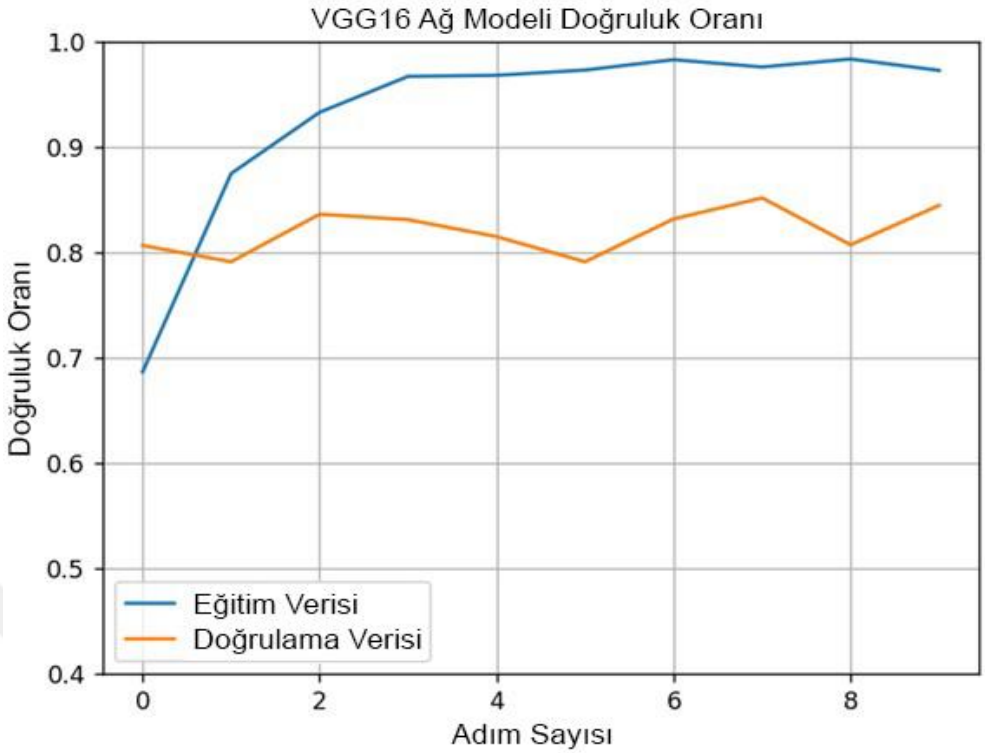
VGG16 Ağ Modeli; 10 adım sayısında (Epoch), 0.001 Öğrenme Oranı ile eğitilmiş olup; 1. Adım sayısında Hata Oranı: 1.2575, Doğruluk Oranı: 0.6865 seviyelerindeyken, 10. Adım sayısında Hata Oranı: 0.0776, Doğruluk Oranı: 0.9728 seviyelerine yükselmiştir.

Eğitimin 1. adım sayısındaki Doğruluk Oranı %68.65 seviyelerindeyken, 10. adım sayısının sonunda %97.28 seviyelerine yükselmiştir. Eğitilen model 10. adım sayısının sonunda %28.63 Doğruluk Oranı'nı arttırmıştır.

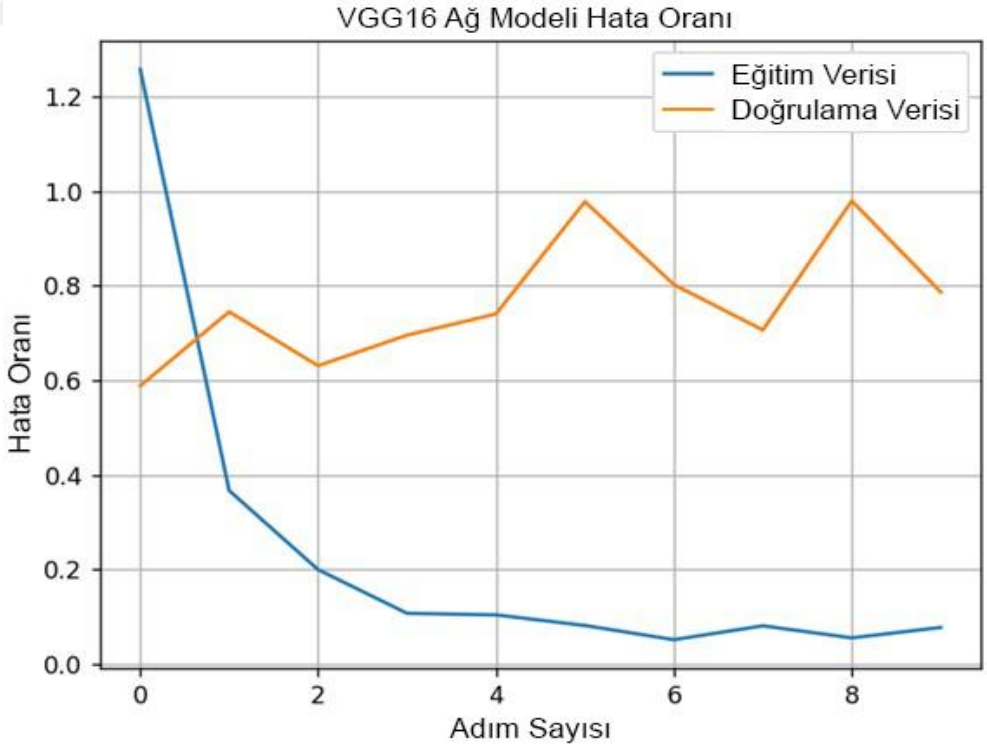
Bu ağ modelinin Eğitim Süresi 91 dk 55 sn sürmüştür. Eğitilen ağ modeline ait adımlar Tablo 3.3'te gösterilmiştir

Tablo 3.3. VGG16 Ağ Modeli Eğitim Adımları

Adım Sayısı	**Öğrenme Oranı**	**Eğitim Süresi (sn)**	**Hata Oranı**	**Doğruluk Oranı**
1	0.001	536	1.2575	0.6865
2	0.001	546	0.3670	0.8748
3	0.001	546	0.1998	0.9330
4	0.001	664	0.1074	0.9671
5	0.001	526	0.1041	0.9681
6	0.001	539	0.0816	0.9731
7	0.001	538	0.0517	0.9829
8	0.001	539	0.0810	0.9760
9	0.001	541	0.0555	0.9836
10	0.001	540	0.0776	0.9728



Şekil 3.3. VGG16 ağ modeline ait Doğruluk Oranı grafiği



Şekil 3.4. VGG16 ağ modeline ait Hata Oranı grafiği

Şekil 3.3'teki Doğruluk Oranı Grafiği ve Şekil 3.4'teki Hata Oranı grafiklerinin analizi, Doğruluk Oranı grafiğinde eğitim verisinin (Train) parabolik bir artış gösterdiğini, doğrulama verisinin (Validation) ise istikrarlı bir düzeyde artış ve azalış sergilediğini, bu süreçte belirli bir değeri koruma eğiliminde olduğunu ortaya koymaktadır. Hata Oranı grafiğinde ise eğitim verisinin parabolik bir azalış gösterdiği, sistem üzerindeki Hata Oranı'nı düşürdüğü ve Doğruluk Oranı'nı artırdığı gözlemlenmektedir. Modelin doğruluğu oldukça yüksek olup, %97.28 seviyelerindedir. Bu durum, modelin etkili bir şekilde öğrenmiş olduğunu ve test verilerinde başarı elde ettiğini göstermektedir.

3.2.2. VGG19

Ağ modelinin eğitilmesi esnasında önceden eğitilmiş model kullanılmış olup, çıkış katmanlarını oluşturan parametreler eğitilmiştir. Tablo 3.4'te katman türü ve çıkış parametreleri gösterilmektedir.

Tablo 3.4. VGG19 Ağ Modeli Çıkış Parametreleri

Katman Türü	**Çıkış**	**Parametreler**
VGG19 (Functional)	(–, 512)	20.024.384
Flatten (Flatten)	(–, 512)	0
Dense (Dense)	(–, 512)	262.656
Dense_1 (Dense)	(–, 1024)	525.312
Dense_2 (Dense)	(–, 512)	524.800
Dense_3 (Dense)	(–, 13)	6.669

VGG19 ağ modelinin eğitilmesinde Toplam Parametreler: 21.343.821, Eğitilebilir Parametreler: 1.319.437 şeklinde olup parametre verileri Tablo 3.5'te gösterilmiştir.

Tablo 3.5. VGG19 Ağ Modeli Parametre Değerleri

Toplam Parametreler	21.343.821 (81.42 MB)
Eğitilebilir Parametreler	1.319.437 (5.03 MB)
Eğitilemeyen Parametreler	20.024.384 (76.39 MB)

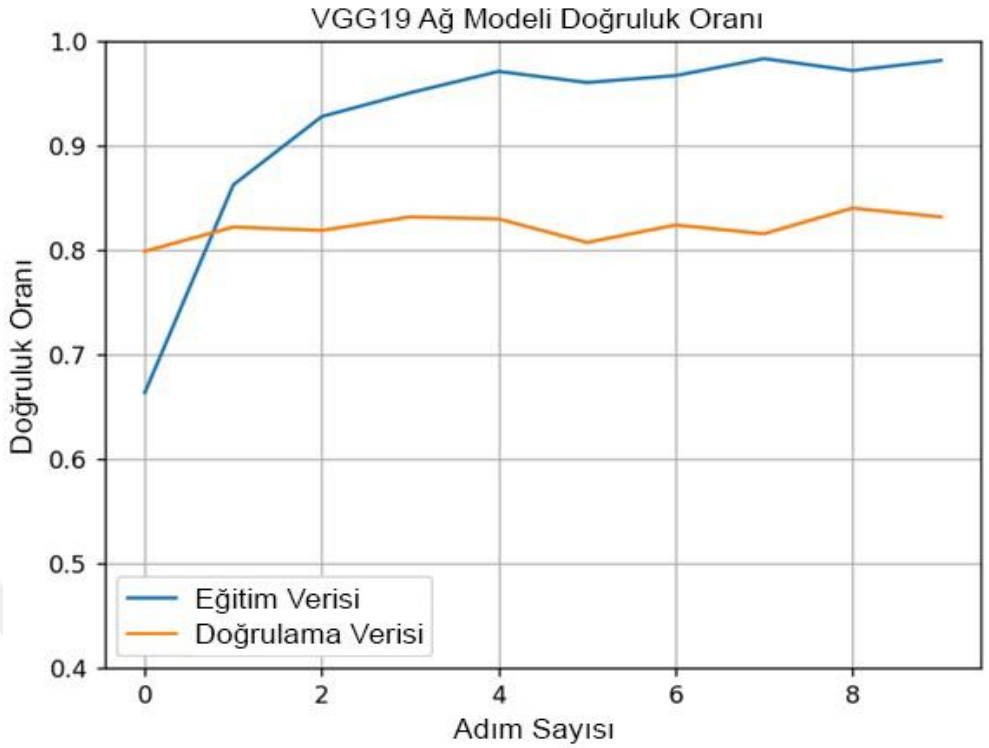
VGG19 Ağ Modeli; 10 adım sayısında, 0.001 Öğrenme Oranı ile eğitilmiş olup; 1. adım sayısında Hata Oranı: 1.3342, Doğruluk Oranı: 0.6636 seviyelerindeyken, 10. adım sayısında Hata Oranı: 0.0619, Doğruluk Oranı: 0.9815 seviyelerine yükselmiştir.

Eğitimin 1. adım sayısında Doğruluk Oranı %66.36 seviyelerindeyken, 10. adım sayısının sonunda %98.15 seviyelerine yükselmiştir. Eğitilen model 10. adım sayısının sonunda %31.79 Doğruluk Oranı'nı arttırmıştır.

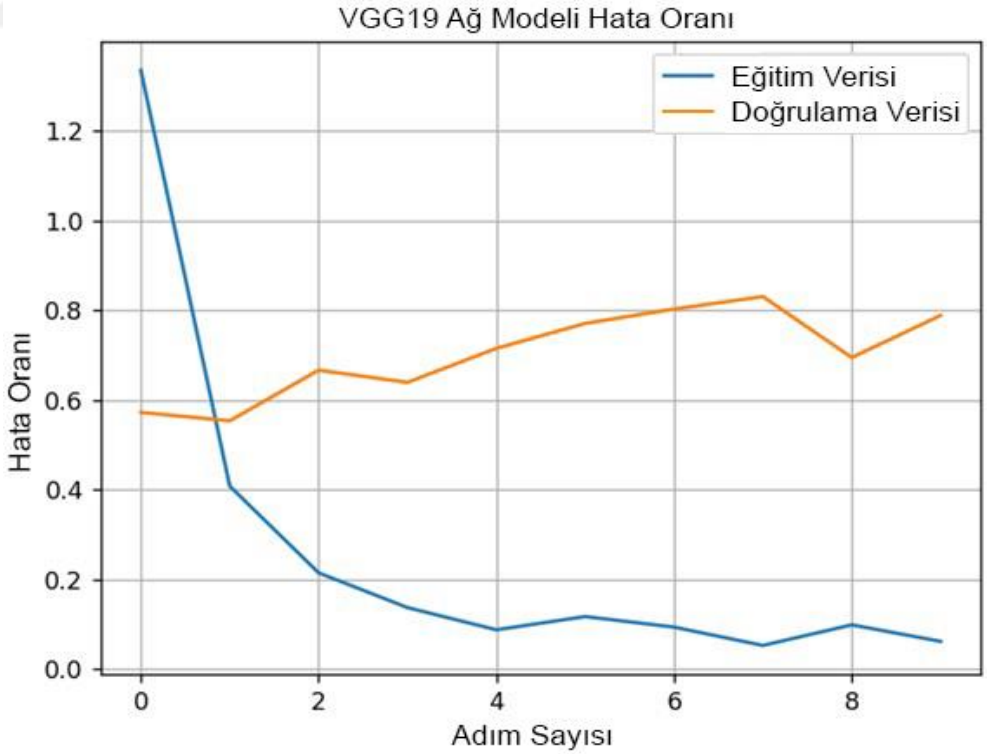
Bu ağ modelinin Eğitim Süresi 113 dk 50 sn sürmüştür. Eğitilen ağ modeline ait adımlar Tablo 3.6'da gösterilmiştir.

Tablo 3.6. VGG19 Ağ Modeli Eğitim Adımları

Adım Sayısı	Öğrenme Oranı	Eğitim Süresi (sn)	Hata Oranı	Doğruluk Oranı
1	0.001	689	1.3342	0.6636
2	0.001	688	0.4078	0.8624
3	0.001	683	0.2146	0.9278
4	0.001	682	0.1371	0.9505
5	0.001	681	0.0877	0.9710
6	0.001	680	0.1172	0.9604
7	0.001	682	0.0936	0.9670
8	0.001	682	0.0528	0.9832
9	0.001	682	0.0987	0.9718
10	0.001	681	0.0619	0.9815



Şekil 3.5. VGG19 ağ modeline ait Doğruluk Oranı grafiği



Şekil 3.6. VGG19 ağ modeline ait Hata Oranı grafiği

Şekil 3.5'te sunulan Doğruluk Oranı Grafiği ile Şekil 3.6'daki Hata Oranı grafiklerinin değerlendirilmesi sonucunda, Doğruluk Oranı grafiğinde eğitim verisinin parabolik bir artış gösterdiğini, doğrulama verisinin ise stabil bir durumu sürdürmeye çalıştığını göstermektedir. Hata Oranı grafiğinde ise eğitim verisinin parabolik bir azalış sergilediği, bu sayede sistem Hata Oranı'nı düşürerek Doğruluk Oranı'nı artırdığı görülmektedir. Modelin doğruluğu oldukça yüksek olup, %98.15 seviyelerindedir. Model, etkili bir şekilde öğrenmiş ve test verilerinde başarı elde etmiştir.

3.2.3. ResNet50

Ağ modelinin eğitilmesi esnasında önceden eğitilmiş model kullanılmış olup, çıkış katmanlarını oluşturan parametreler eğitilmiştir. Tablo 3.7.'de katman türü ve çıkış parametreleri gösterilmektedir.

Tablo 3.7. ResNet50 Ağ Modeli Çıkış Parametreleri

Katman Türü	**Çıkış**	**Parametreler**
ResNet50 (Functional)	(–, 2048)	23.587.712
Flatten (Flatten)	(–, 2048)	0
Dense (Dense)	(–, 512)	1.049.088
Dense_1 (Dense)	(–, 1024)	525.312
Dense_2 (Dense)	(–, 512)	524.800
Dense_3 (Dense)	(–, 13)	6.669

ResNet50 ağ modelinin eğitilmesinde Toplam Parametreler: 25.693.581, Eğitilebilir Parametreler: 2.105.869 şeklinde olup parametre verileri Tablo 3.8'de gösterilmiştir.

Tablo 3.8. ResNet50 Ağ Modeli Parametre Değerleri

Toplam Parametreler	25.693.581 (98.01 MB)
Eğitilebilir Parametreler	2.105.869 (8.03 MB)
Eğitilemeyen Parametreler	23.587.712 (89.98 MB)

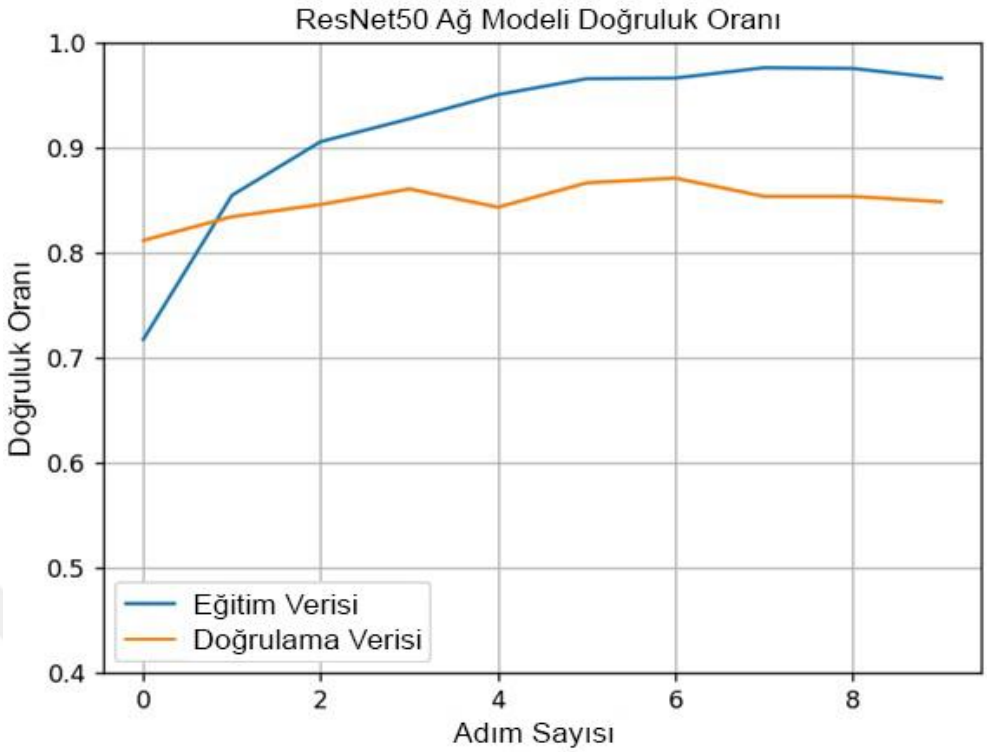
ResNet50 Ağ Modeli; 10 adım sayısında, 0.001 Öğrenme Oranı ile eğitilmiş olup; 1. adım sayısında Hata Oranı: 0.8439, Doğruluk Oranı: 0.7174 seviyelerindeyken, 10. adım sayısında Hata Oranı: 0.0978, Doğruluk Oranı: 0.9662 seviyelerine yükselmiştir.

Eğitimin 1. adım sayısında Doğruluk Oranı %71.74 seviyelerindeyken, 10. adım sayısının sonunda %96.62 seviyelerine yükselmiştir. Eğitilen model 10. adım sayısı sonunda %24.88 Doğruluk Oranı'nı arttırmıştır.

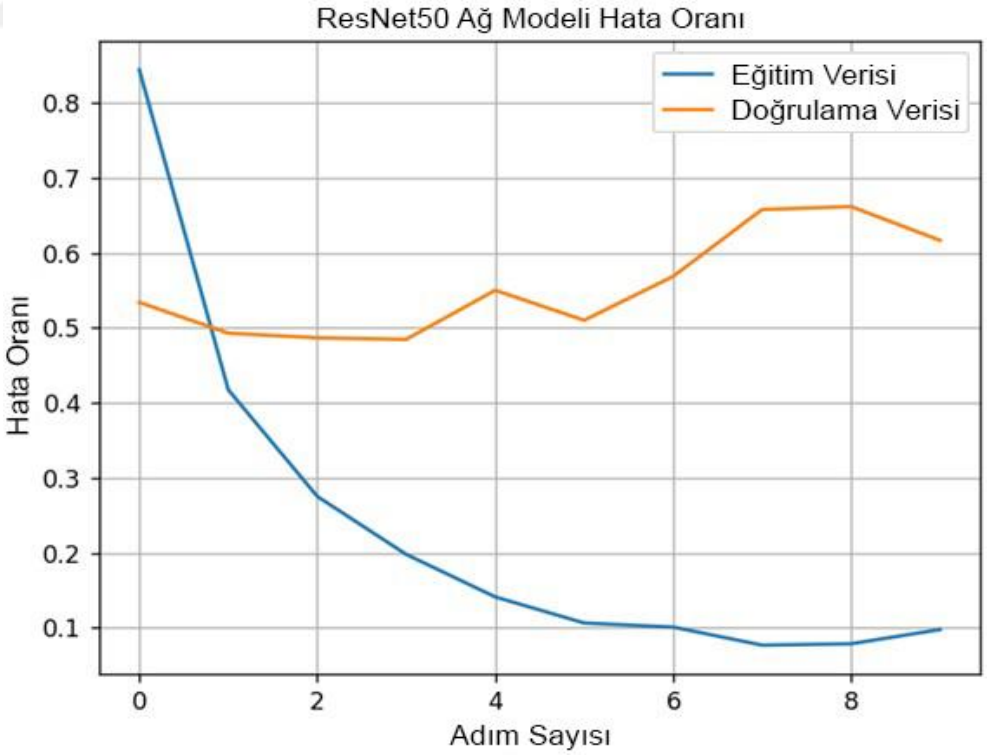
Bu ağ modelinin Eğitim Süresi 42 dk 38 sn sürmüştür. Eğitilen ağ modeline ait adımlar Tablo 3.9'da gösterilmiştir.

Tablo 3.9. ResNet50 Ağ Modeli Eğitim Adımları

Adım Sayısı	Öğrenme Oranı	Eğitim Süresi (sn)	Hata Oranı	Doğruluk Oranı
1	0.001	248	0.8439	0.7174
2	0.001	255	0.4174	0.8547
3	0.001	259	0.2753	0.9058
4	0.001	256	0.1980	0.9275
5	0.001	256	0.1413	0.9505
6	0.001	263	0.1066	0.9657
7	0.001	256	0.1011	0.9663
8	0.001	254	0.0771	0.9762
9	0.001	254	0.0789	0.9755
10	0.001	257	0.0978	0.9662



Şekil 3.7. ResNet50 ağ modeline ait Doğruluk Oranı grafiği



Şekil 3.8. ResNet50 ağ modeline ait Hata Oranı grafiği

Şekil 3.7’deki Doğruluk Oranı Grafiği ve Şekil 3.8'deki Hata Oranı grafiklerinin incelenmesi sonucunda, Doğruluk Oranı grafiğinde eğitim verisinin parabolik bir artış sergilediği tespit edilmiştir. Hata Oranı grafiğinde ise eğitim verisinin parabolik bir azalış gösterdiği ve bu durumun sistem üzerindeki Hata Oranı’nı düşürerek Doğruluk Oranı’nı artırdığı gözlemlenmiştir. Modelin doğruluğu oldukça yüksek olup, %96.62 seviyelerindedir. Model, etkili bir öğrenme süreci geçirmiş ve test verilerinde başarı elde etmiştir.

3.2.4. ResNet50V2

Ağ modelinin eğitilmesi esnasında önceden eğitilmiş model kullanılmış olup, çıkış katmanlarını oluşturan parametreler eğitilmiştir. Tablo 3.10’da katman türü ve çıkış parametreleri gösterilmektedir.

Tablo 3.10. ResNet50V2 Ağ Modeli Çıkış Parametreleri

Katman Türü	**Çıkış**	**Parametreler**
ResNet50V2 (Functional)	(–, 2048)	23.564.800
Flatten (Flatten)	(–, 2048)	0
Dense (Dense)	(–, 512)	1.049.088
Dense_1 (Dense)	(–, 1024)	525.312
Dense_2 (Dense)	(–, 512)	524.800
Dense_3 (Dense)	(–, 13)	6.669

ResNet50V2 ağ modelinin eğitilmesinde Toplam Parametreler: 25.70.669, Eğitilebilir Parametreler: 2.105.869 şeklinde olup parametre verileri Tablo 3.11’de gösterilmiştir.

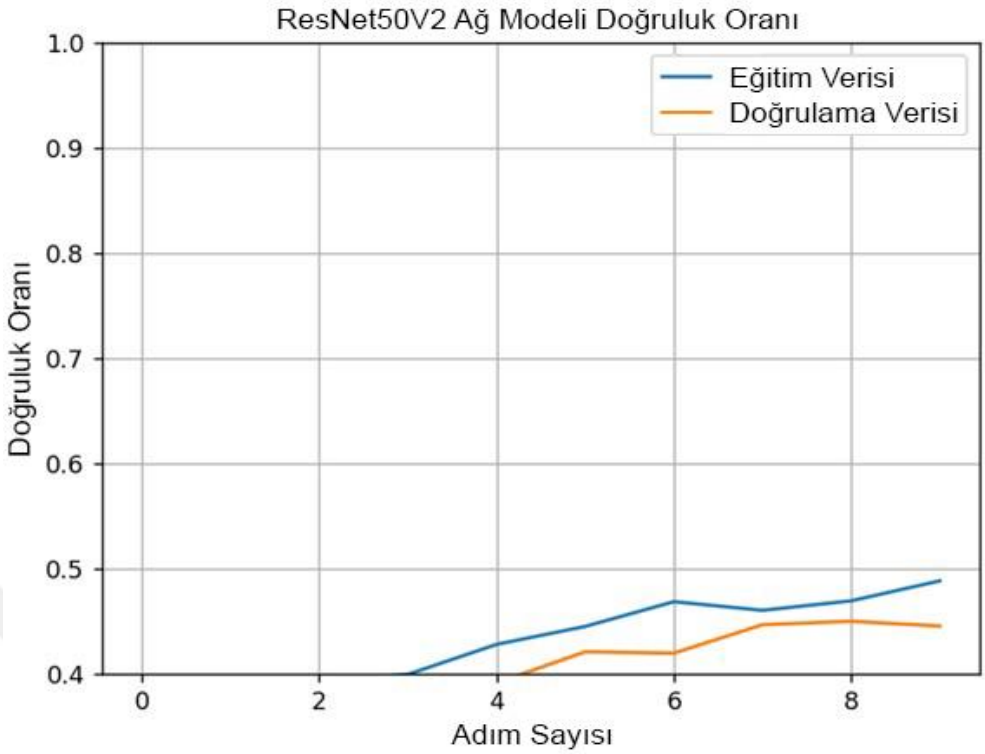
Tablo 3.11. ResNet50V2 Ağ Modeli Parametre Değerleri

Toplam Parametreler	25.670.669 (97.93 MB)
Eğitilebilir Parametreler	2.105.869 (8.03 MB)
Eğitilemeyen Parametreler	23.564.800 (89.89 MB)

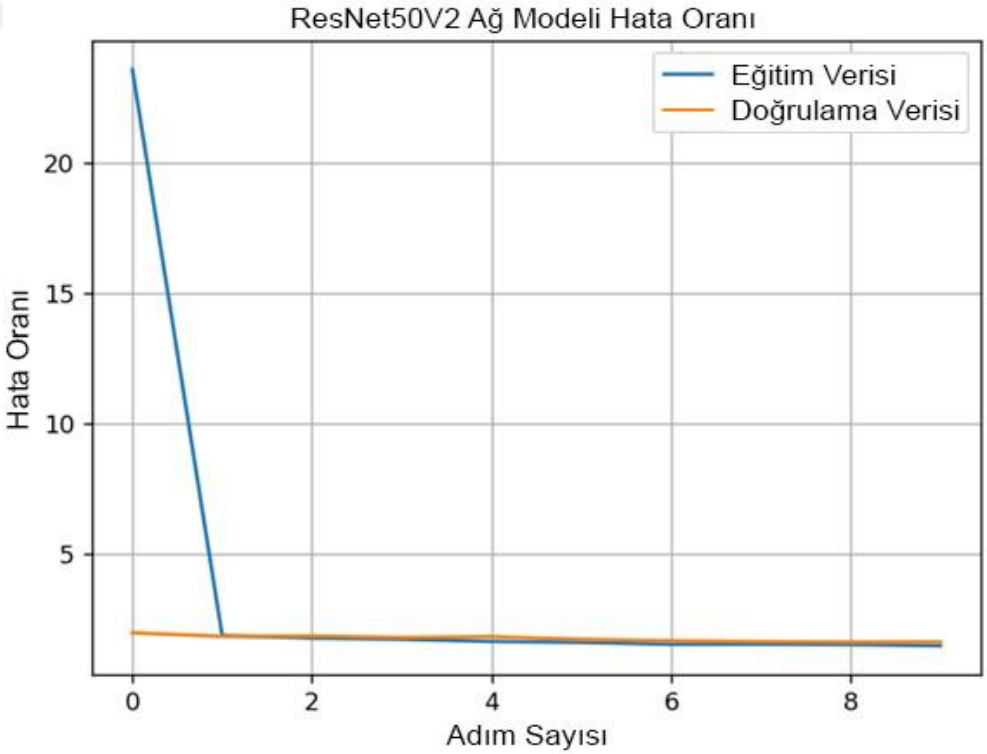
ResNet50V2 ağ modeli; 10 adım sayısı süresince, 0.001 Öğrenme Oranı ile eğitilmiştir. 1. adım sayısında Hata Oranı: 23.5926, Doğruluk Oranı: 0.2326 seviyelerinde iken, 10. adım sayısında bu değerler sırasıyla 1.4612 ve 0.4885 seviyelerine yükselmiştir. Eğitimin 1. adım sayısında gözlemlenen %23.26 seviyelerindeki Doğruluk Oranı, 10. adım sayısının sonunda %48.85 seviyelerine yükselmiştir. Dolayısıyla, eğitilen model, 10. adım sayısı sonunda %25.59'luk bir Doğruluk Oranı artışı elde etmiştir. Bu ağ modelinin Eğitim Süresi toplamda 31 dk 29 sn sürmüştür. Eğitilen ağ modeline ait adımlar Tablo 3.12'de ayrıntılı olarak gösterilmiştir.

Tablo 3.12. ResNet50V2 Ağ Modeli Eğitim Adımları

Adım Sayısı	Öğrenme Oranı	Eğitim Süresi (sn)	Hata Oranı	Doğruluk Oranı
1	0.001	194	23.5926	0.2326
2	0.001	196	1.8535	0.3588
3	0.001	198	1.7526	0.3937
4	0.001	193	1.7070	0.3995
5	0.001	196	1.6256	0.4281
6	0.001	194	1.5889	0.4451
7	0.001	181	1.5138	0.4687
8	0.001	180	1.5161	0.4604
9	0.001	179	1.5067	0.4695
10	0.001	178	1.4612	0.4885



Şekil 3.9. ResNet50V2 ağ modeline ait Doğruluk Oranı grafiği



Şekil 3.10. ResNet50V2 ağ modeline ait Hata Oranı grafiği

Şekil 3.9’da sunulan Doğruluk Oranı Grafiği ve Şekil 3.10'daki Hata Oranı grafiklerinin analizi sonucunda, Doğruluk Oranı grafiğinde eğitim verisinin 5. adım sayısından itibaren %40 seviyelerine ulaşarak lineer bir artış sergilediğini göstermektedir. Ancak, bu seviyede yeterli Doğruluk Oranı'na ulaşılamamış ve sistem tam anlamıyla eğitilememiştir. Hata Oranı grafiğinde ise eğitim verisinin lineer bir azalış gösterdiği, ancak 1. adım sayısından sonra stabilizasyon çabası içinde olduğu gözlemlenmektedir. Sistem üzerindeki Hata Oranı’nı azaltmış olmakla birlikte, Doğruluk Oranı’nda yeterli düzeyde artış sağlanamamıştır. Modelin performansı istenilen seviyede olmayıp, Doğruluk Oranı %48.85 seviyelerindedir. Modelin etkili bir şekilde öğrenmediği ve test verilerinde yeterli başarı elde edilemediği sonucuna varılmıştır.

3.2.5. ResNet101

Ağ modelinin eğitilmesi esnasında önceden eğitilmiş model kullanılmış olup, çıkış katmanlarını oluşturan parametreler eğitilmiştir. Tablo 3.13’te katman türü ve çıkış parametreleri gösterilmektedir.

Tablo 3.13. ResNet101 Ağ Modeli Çıkış Parametreleri

Katman Türü	**Çıkış**	**Parametreler**
ResNet101 (Functional)	(–, 2048)	42.658.176
Flatten (Flatten)	(–, 2048)	0
Dense (Dense)	(–, 512)	1.049.088
Dense_1 (Dense)	(–, 1024)	525.312
Dense_2 (Dense)	(–, 512)	524.800
Dense_3 (Dense)	(–, 13)	6.669

ResNet101 ağ modelinin eğitilmesinde Toplam Parametreler: 44.764.045, Eğitilebilir Parametreler: 2.105.869 şeklinde olup parametre verileri Tablo 3.14’te gösterilmiştir.

Tablo 3.14. ResNet101 Ağ Modeli Parametre Değerleri

Toplam Parametreler	44.764.045 (170.76 MB)
Eğitilebilir Parametreler	2.105.869 (8.03 MB)
Eğitilemeyen Parametreler	42.658.176 (162.73 MB)

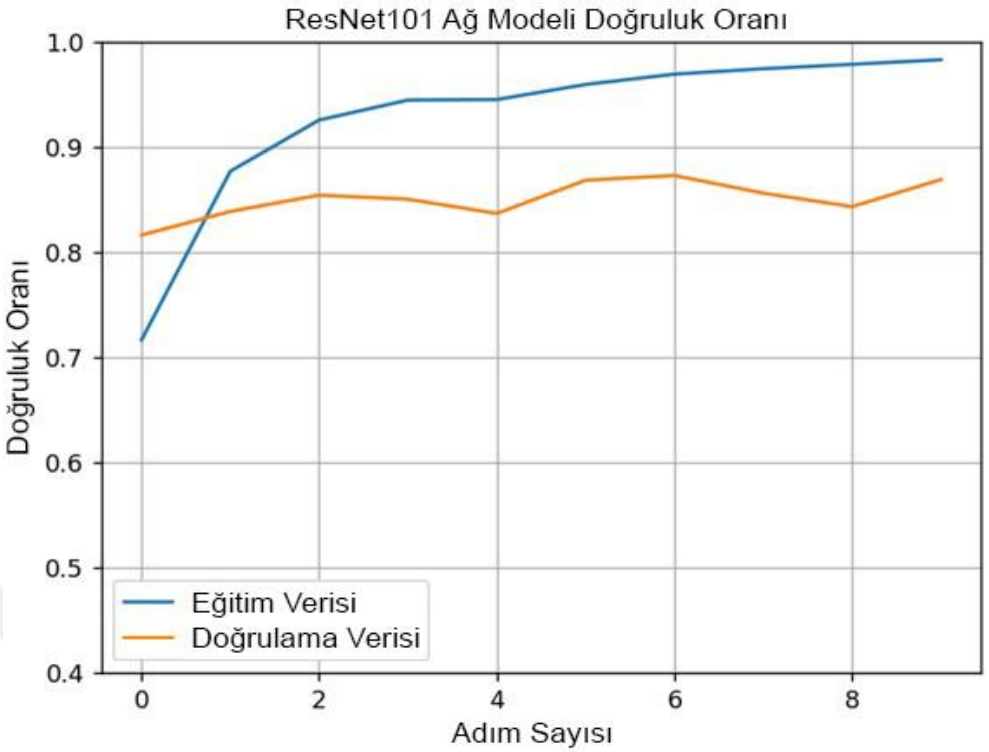
ResNet101 Ağ Modeli; 10 adım sayısında, 0.001 Öğrenme Oranı ile eğitilmiş olup; 1. adım sayısında Hata Oranı: 0.8051, Doğruluk Oranı: 0.7166 seviyelerindeyken, 10. adım sayısında Hata Oranı: 0.0578, Doğruluk Oranı: 0.9831 seviyelerine yükselmiştir.

Eğitimin 1. adım sayısında Doğruluk Oranı %71.66 seviyelerindeyken, 10. adım sayısının sonunda %98.31 seviyelerine yükselmiştir. Eğitilen model 10. adım sayısı sonunda %26.65 Doğruluk Oranı'nı arttırmıştır.

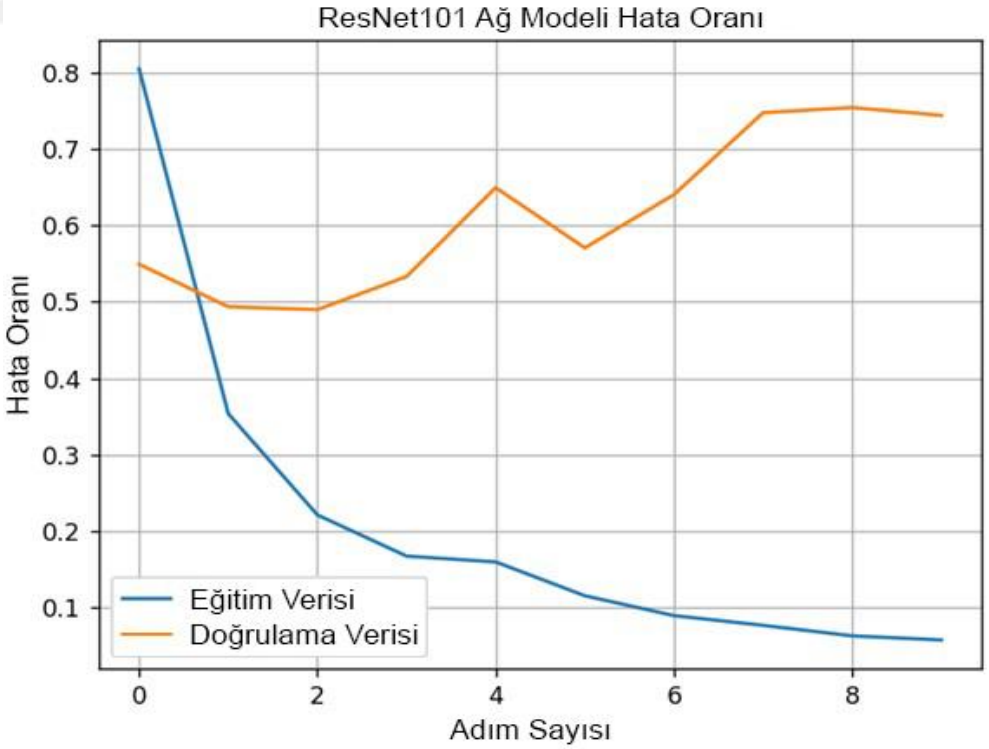
Bu ağ modelinin Eğitim Süresi 147 dk 1 sn sürmüştür. Eğitilen ağ modeline ait adımlar Tablo 3.15'te gösterilmiştir.

Tablo 3.15. ResNet101 Ağ Modeli Eğitim Adımları

Adım Sayısı	Öğrenme Oranı	Eğitim Süresi (sn)	Hata Oranı	Doğruluk Oranı
1	0.001	814	0.8051	0.7166
2	0.001	768	0.3539	0.8771
3	0.001	2013	0.2214	0.9257
4	0.001	760	0.1675	0.9447
5	0.001	759	0.1599	0.9452
6	0.001	772	0.1156	0.9596
7	0.001	730	0.0896	0.9694
8	0.001	728	0.0769	0.9745
9	0.001	737	0.0631	0.9787
10	0.001	740	0.0578	0.9831



Şekil 3.11. ResNet101 ağ modeline ait Doğruluk Oranı grafiği



Şekil 3.12. ResNet101 ağ modeline ait Hata Oranı grafiği

Şekil 3.11'de gösterilen Doğruluk Oranı Grafiği ve Şekil 3.12'deki Hata Oranı grafiklerinin analizi sonucunda, Doğruluk Oranı grafiğinde eğitim verisinin parabolik bir artış sergilediğini ortaya koymaktadır. Hata Oranı grafiğinde ise eğitim verisinin parabolik bir azalış gösterdiği gözlemlenmiştir. Bu durum, sistem üzerindeki Hata Oranı'nı azaltarak Doğruluk Oranı'nda artış sağladığını göstermektedir. Modelin doğruluğu oldukça yüksek olup, %98.31 seviyelerindedir. Modelin etkili bir öğrenme süreci geçirmiş olduğu ve test verilerinde başarı elde ettiği sonucuna varılmıştır.

3.2.6. MobileNet

Ağ modelinin eğitilmesi esnasında önceden eğitilmiş model kullanılmış olup, çıkış katmanlarını oluşturan parametreler eğitilmiştir. Tablo 3.16'da katman türü ve çıkış parametreleri gösterilmektedir.

Tablo 3.16. MobileNet Ağ Modeli Çıkış Parametreleri

Katman Türü	**Çıkış**	**Parametreler**
MobileNet_1.00_224 (Functional)	(–, 1024)	3.228.864
Flatten (Flatten)	(–, 512)	0
Dense (Dense)	(–, 512)	524.800
Dense_1 (Dense)	(–, 1024)	525.312
Dense_2 (Dense)	(–, 512)	524.800
Dense_3 (Dense)	(–, 13)	6.669

MobileNet ağ modelinin eğitilmesinde Toplam Parametreler: 4.810.445, Eğitilebilir Parametreler: 1.581.581 şeklinde olup parametre verileri Tablo 3.17'de gösterilmiştir.

Tablo 3.17. MobileNet Ağ Modeli Parametre Değerleri

Toplam Parametreler	4.810.445 (18.35 MB)
Eğitilebilir Parametreler	1.581.581 (6.03 MB)
Eğitilemeyen Parametreler	3.228.864 (12.32 MB)

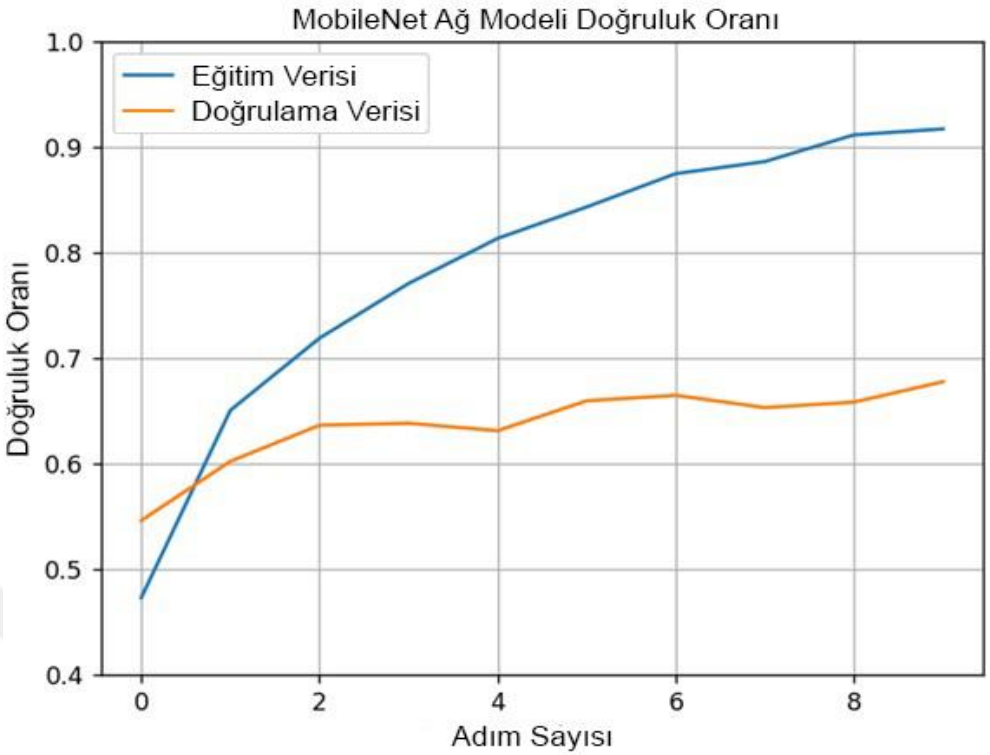
MobileNet Ağ Modeli; 10 eğitim sayısında, 0.001 Öğrenme Oranı ile eğitilmiş olup; 1. adım sayısında Hata Oranı: 1.5292, Doğruluk Oranı: 0.4730 seviyelerindeyken, 10. adım sayısında Hata Oranı: 0.2389, Doğruluk Oranı: 0.9172 seviyelerine yükselmiştir.

Eğitimin 1. adım sayısındaki Doğruluk Oranı %47.30 seviyelerindeyken, 10. adım sayısının sonunda %91.72 seviyelerine yükselmiştir. Eğitilen model 10. adım sayısı sonunda %44.42 Doğruluk Oranı'nı arttırmıştır.

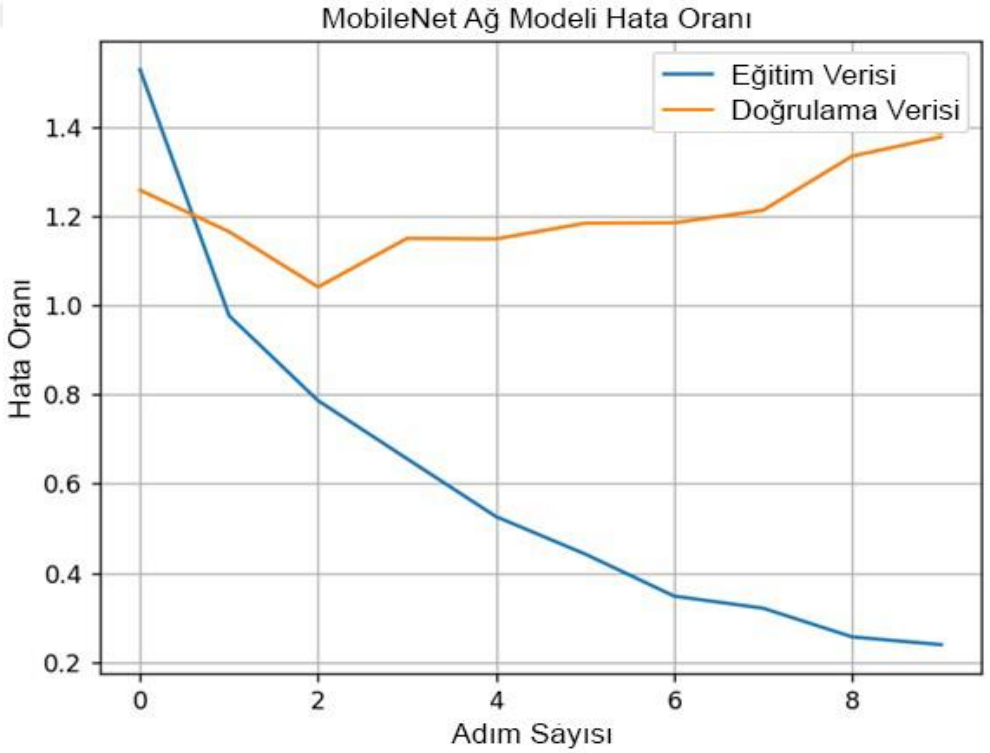
Bu ağ modelinin Eğitim Süresi 10 dk 7 sn sürmüştür. Eğitilen ağ modeline ait adımlar Tablo 3.18'de gösterilmiştir.

Tablo 3.18. MobileNet Ağ Modeli Eğitim Adımları

Adım Sayısı	Öğrenme Oranı	Eğitim Süresi (sn)	Hata Oranı	Doğruluk Oranı
1	0.001	61	1.5292	0.4730
2	0.001	59	0.9772	0.6506
3	0.001	60	0.7865	0.7189
4	0.001	61	0.6561	0.7707
5	0.001	61	0.5262	0.8134
6	0.001	61	0.4424	0.8432
7	0.001	61	0.3479	0.8748
8	0.001	61	0.3207	0.8863
9	0.001	61	0.2564	0.9116
10	0.001	61	0.2389	0.9172



Şekil 3.13. MobileNet ağ modeline ait Doğruluk Oranı grafiği



Şekil 3.14. MobileNet ağ modeline ait Hata Oranı grafiği

Şekil 3.13'te yer alan Doğruluk Oranı Grafiği ve Şekil 3.14'te yer alan Hata Oranı grafikleri incelendiğinde Doğruluk Oranı grafiğinde eğitim verisi parabolik olarak artış sağlamıştır. Hata Oranı grafiğinde ise eğitim verisi parabolik olarak azalış göstermiştir. Sistem üzerindeki Hata Oranı'nı azaltmış, Doğruluk Oranı'nda artış sağlamıştır. Model doğruluğu %91.72 seviyelerindedir. Diğer ağ modellerine oranla daha düşük doğruluğa sahiptir. Model öğrenmiş olup, test verilerinde genellikle başarı sağlamıştır.

3.2.7. DenseNet121

Ağ modelinin eğitilmesi esnasında önceden eğitilmiş model kullanılmış olup, çıkış katmanlarını oluşturan parametreler eğitilmiştir. Tablo 3.19'da katman türü ve çıkış parametreleri gösterilmektedir.

Tablo 3.19. DenseNet121 Ağ Modeli Çıkış Parametreleri

Katman Türü	Çıkış	Parametreler
DenseNet121 (Functional)	(–, 1024)	7.037.504
Flatten (Flatten)	(–, 1024)	0
Dense (Dense)	(–, 512)	524.800
Dense_1 (Dense)	(–, 1024)	525.312
Dense_2 (Dense)	(–, 512)	524.800
Dense_3 (Dense)	(–, 13)	6.669

DenseNet121 ağ modelinin eğitilmesinde Toplam Parametreler: 8.619.085, Eğitilebilir Parametreler: 1.581.581 şeklinde olup parametre verileri Tablo 3.20'de gösterilmiştir.

Tablo 3.20. DenseNet121 Ağ Modeli Parametre Değerleri

Toplam Parametreler	8.619.085 (32.88 MB)
Eğitilebilir Parametreler	1.581.581 (6.03 MB)
Eğitilemeyen Parametreler	7.037.504 (26.85 MB)

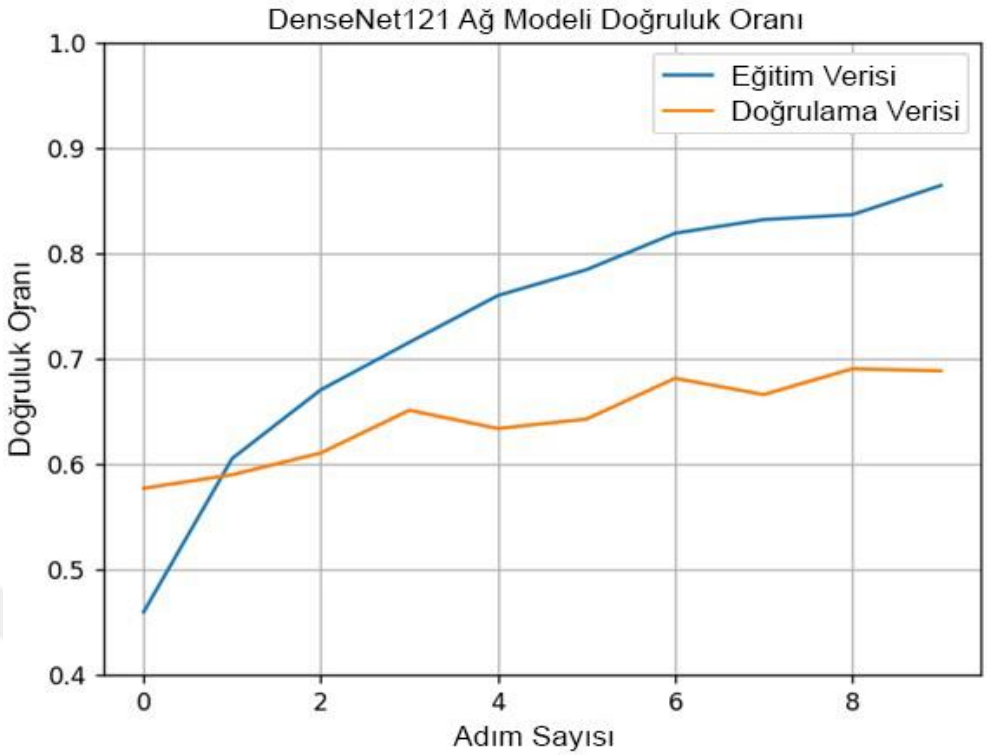
DenseNet121Ağ Modeli; 10 adım sayısında, 0.001 Öğrenme Oranı ile eğitilmiş olup; 1. adım sayısında Hata Oranı: 1.9603, Doğruluk Oranı: 0.4596 seviyelerindeyken, 10. adım sayısında Hata Oranı: 0.3722, Doğruluk Oranı: 0.8645 seviyelerine yükselmiştir.

Eğitimin 1. Adım sayısındaki Doğruluk Oranı %45.96 seviyelerindeyken, 10. adım sayısının sonunda %86.45 seviyelerine yükselmiştir. Eğitilen model 10. adım sayısı sonunda %40.49 Doğruluk Oranı'nı arttırmıştır.

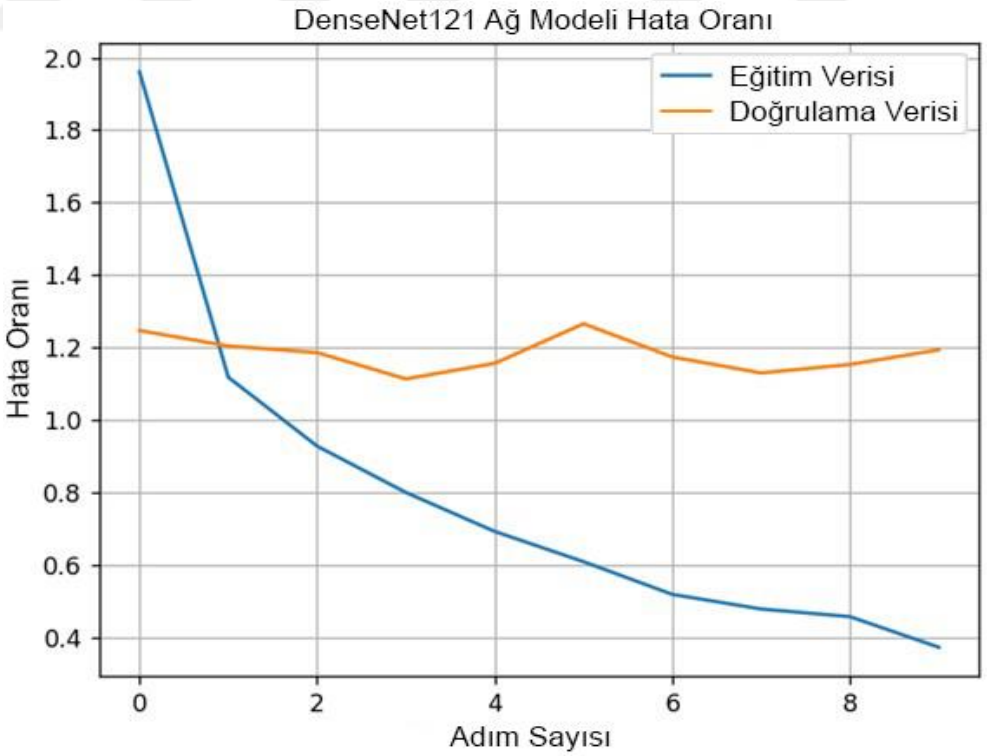
Bu ağ modelinin Eğitim Süresi 32 dk 3 sn sürmüştür. Eğitilen ağ modeline ait adımlar Tablo 3.21'de gösterilmiştir.

Tablo 3.21. DenseNet121 Ağ Modeli Eğitim Adımları

Adım Sayısı	Öğrenme Oranı	Eğitim Süresi (sn)	Hata Oranı	Doğruluk Oranı
1	0.001	188	1.9603	0.4596
2	0.001	192	1.1175	0.6054
3	0.001	193	0.9275	0.6705
4	0.001	193	0.7994	0.7156
5	0.001	193	0.6919	0.7601
6	0.001	193	0.6084	0.7846
7	0.001	194	0.5181	0.8192
8	0.001	192	0.4777	0.8321
9	0.001	193	0.4566	0.8368
10	0.001	192	0.3722	0.8645



Şekil 3.15. DenseNet121 ağ modeline ait Doğruluk Oranı grafiği



Şekil 3.16. DenseNet121 ağ modeline ait Hata Oranı grafiği

Şekil 3.15'te yer alan Doğruluk Oranı Grafiği ve Şekil 3.16'da yer alan Hata Oranı grafiklerinin değerlendirilmesi sonucunda, Doğruluk Oranı grafiğinde eğitim verisinin parabolik bir artış sergilediği tespit edilmiştir. Hata Oranı grafiğinde ise eğitim verisinin parabolik bir azalış gösterdiği gözlemlenmiştir. Bu durum, sistem üzerindeki Hata Oranı'nı azaltarak Doğruluk Oranı'nda artış sağladığını göstermektedir. Modelin doğruluğu %86.45 seviyelerindedir ve diğer ağ modellerine kıyasla daha düşük bir doğruluk düzeyine sahiptir. Ancak, modelin öğrenme sürecini başarıyla tamamladığı ve genellikle test verilerinde başarılı sonuçlar elde ettiği belirlenmiştir.

3.3. Test Verileri, Sonuçları ve Analizleri

Veri Setini oluşturan toplam 3.900 adet nesne görselinin %70'i ağ modellerinin eğitilmesinde kullanılmış olup geriye kalan %30'luk kısmı test verilerinde kullanılmıştır. Test verilerinde kullanılan %30'luk kısım her bir nesne için 90 adet görseli oluşturmaktadır.

Tablo 3.22'de VGG16 Ağ modeline ait test sonuçlarını gösterilmektedir. Oluşturulan test sonuçları neticesinde toplam 1.170 adet görsel analiz edilmiş olup, 1.048 adet görsel doğru tahmin edilmiş, 122 adet görsel ise yanlış tahmin edilmiştir. Doğru tahmin edilen görsel sayısının, test edilen görsel sayısına oranı ile sistemin Tahmin Değerleri hesaplanmıştır. Sistemin Tahmin Değerleri'nin aritmetik ortalaması ve Doğruluk Oranları'nın aritmetik ortalaması alınmış Tablo 3.22'de gösterilmiştir. VGG16 Ağ modelinde test edilen 13 farklı nesne sınıfına ait Tahmin Değerleri'nin ortalaması %89.5726; Doğruluk Oranları'nın ortalaması ise %97.9019 şeklinde bulunmuştur

Tablo 3.22. VGG16 Ağ Modeline Ait Test Sonuçları

Nesne Tipi	Test Edilen Görsel Sayısı	Doğru Tahmin Edilen Görsel Sayısı	Yanlış Tahmin Edilen Görsel Sayısı	Tahmin Değerlerinin Ortalaması (%)	Doğruluk Oranlarının Ortalaması (%)
Audi	90	86	4	95.5555	98.8357
Papatya	90	89	1	98.8888	99.1611
Karahindiba	90	82	8	91.1111	99.9666
İnsan	90	90	0	100	99.9263
Hyundai Creta	90	83	7	92.2222	96.1359
Mahindra Scorpio	90	76	14	84.4444	98.5986
Rolls Royce	90	57	33	63.3333	94.9773
Gül	90	83	7	92.2222	97.7677
Ay Çiçeği	90	68	22	75.5555	95.0068
Swift	90	81	9	90	98.2386
Tata Safari	90	76	14	84.4444	95.9204
Toyota İnnova	90	88	2	97.7777	98.9308
Lale	90	89	1	98.8888	99.2597
ORTALAMA				**89.5726**	**97.9019**

Tablo 3.23'te VGG19 Ağ modeline ait test sonuçlarını gösterilmektedir. Oluşturulan test sonuçları neticesinde toplam 1.170 adet görsel analiz edilmiş olup, 1.081 adet görsel doğru tahmin edilmiş, 89 adet görsel ise yanlış tahmin edilmiştir. Doğru tahmin edilen görsellerin sayısının, test edilen görsel sayısına oranı ile sistemin Tahmin Değerleri hesaplanmıştır. Sistemin Tahmin Değerleri'nin aritmetik ortalaması ve Doğruluk Oranları'nın aritmetik ortalaması alınmış Tablo 3.23'te gösterilmiştir. VGG19 Ağ modelinde test edilen 13 farklı nesne sınıfına ait Tahmin Değerleri'nin ortalaması %92.3931; Doğruluk Oranları'nın ortalaması %97.8475 şeklinde bulunmuştur.

Tablo 3.23. VGG19 Ağ Modeline Ait Test Sonuçları

Nesne Tipi	Test Edilen Görsel Sayısı	Doğru Tahmin Edilen Görsel Sayısı	Yanlış Tahmin Edilen Görsel Sayısı	Tahmin Değerlerinin Ortalaması (%)	Doğruluk Oranlarının Ortalaması (%)
Audi	90	87	3	96.6666	99.3510
Papatya	90	87	3	96.6666	99.2573
Karahindiba	90	87	3	96.6666	99.4139
İnsan	90	88	2	97.7777	98.8579
Hyundai Creta	90	84	6	93.3333	97.7633
Mahindra Scorpio	90	87	3	96.6666	98.2582
Rolls Royce	90	80	10	88.8888	97.4656
Gül	90	84	6	93.3333	97.8030
Ay Çiçeği	90	79	11	87.7777	95.4057
Swift	90	68	22	75.5555	94.9786
Tata Safari	90	79	11	87.7777	94.7761
Toyota İnnova	90	84	6	93.3333	98.9285
Lale	90	87	3	96.6666	99.7593
ORTALAMA				**92.3931**	**97.8475**

Tablo 3.24'te ResNet50 Ağ modeline ait test sonuçlarını gösterilmektedir. Oluşturulan test sonuçları neticesinde toplam 1.170 adet görsel analiz edilmiş olup, 1.004 adet görsel doğru tahmin edilmiş, 166 adet görsel ise yanlış tahmin edilmiştir. Doğru tahmin edilen görsellerin sayısının, test edilen görsel sayısına oranı ile sistemin Tahmin Değerleri hesaplanmıştır. Sistemin Tahmin Değerleri'nin aritmetik ortalaması ve Doğruluk Oranları'nın aritmetik ortalaması alınmış Tablo 3.24'te gösterilmiştir. ResNet50 Ağ modelinde test edilen 13 farklı nesne sınıfına ait Tahmin Değerleri'nin ortalaması %85.8119; Doğruluk Oranları'nın ortalaması %95.1419 şeklinde bulunmuştur.

Tablo 3.24. ResNet50 Ağ Modeline Ait Test Sonuçları

Nesne Tipi	Test Edilen Görsel Sayısı	Doğru Tahmin Edilen Görsel Sayısı	Yanlış Tahmin Edilen Görsel Sayısı	Tahmin Değerlerinin Ortalaması (%)	Doğruluk Oranlarının Ortalaması (%)
Audi	90	81	9	90	96.0406
Papatya	90	86	4	95.5555	98.6805
Karahindiba	90	85	5	94.4444	98.3555
İnsan	90	89	1	98.8888	98.8190
Hyundai Creta	90	70	20	77.7777	90.3153
Mahindra Scorpio	90	77	13	85.5555	93.5257
Rolls Royce	90	75	15	83.3333	94.1458
Gül	90	76	14	84.4444	94.7048
Ay Çiçeği	90	33	57	36.6666	88.5586
Swift	90	90	0	100	97.6062
Tata Safari	90	84	6	93.3333	91.3456
Toyota İnnova	90	68	22	75.5555	94.9524
Lale	90	90	0	100	99.7950
ORTALAMA				**85.8119**	**95.1419**

Tablo 3.25'te ResNet50V2 Ağ modeline ait test sonuçlarını gösterilmektedir. Oluşturulan test sonuçları neticesinde toplam 1.170 adet görsel analiz edilmiş olup, 387 adet görsel doğru tahmin edilmiş, 783 adet görsel ise yanlış tahmin edilmiştir. Doğru tahmin edilen görsel sayısının, test edilen görsel sayısına oranı ile sistemin Tahmin Değerleri hesaplanmıştır. Sistemin Tahmin Değerleri'nin aritmetik ortalaması ve Doğruluk Oranları'nın aritmetik ortalaması alınmış Tablo 3.25'te gösterilmiştir. ResNet50V2 Ağ modelinde test edilen 13 farklı nesne sınıfına ait Tahmin Değerleri'nin ortalaması %33.0768; Doğruluk Oranları'nın ortalaması ise %45.3272 şeklinde bulunmuştur.

Tablo 3.25. ResNet50V2 Ağ Modeline Ait Test Sonuçları

Nesne Tipi	Test Edilen Görsel Sayısı	Doğru Tahmin Edilen Görsel Sayısı	Yanlış Tahmin Edilen Görsel Sayısı	Tahmin Değerlerinin Ortalaması (%)	Doğruluk Oranlarının Ortalaması (%)
Audi	90	52	38	57.7777	50.2321
Papatya	90	63	27	70	61.4962
Karahindiba	90	33	57	36.6666	58.1707
İnsan	90	45	45	50	74.0474
Hyundai Creta	90	2	88	2.2222	38.3328
Mahindra Scorpio	90	9	81	10	48.5549
Rolls Royce	90	0	90	0	0
Gül	90	12	78	13.3333	40.2480
Ay Çiçeği	90	28	62	31.1111	48.7333
Swift	90	4	86	4.4444	23.2502
Tata Safari	90	41	49	45.5555	37.9013
Toyota İnnova	90	52	38	57.7777	51.5414
Lale	90	46	44	51.1111	56.7464
ORTALAMA				**33.0768**	**45.3272**

Tablo 3.26'da ResNet101 Ağ modeline ait test sonuçlarını gösterilmektedir. Oluşturulan test sonuçları neticesinde toplam 1.170 adet görsel analiz edilmiş olup, 1.066 adet görsel doğru tahmin edilmiş, 104 adet görsel ise yanlış tahmin edilmiştir. Doğru tahmin edilen görsel sayısının, test edilen görsel sayısına oranı ile sistemin Tahmin Değerleri hesaplanmıştır. Sistemin Tahmin Değerleri'nin aritmetik ortalaması ve Doğruluk Oranları'nın aritmetik ortalaması alınmış Tablo 3.26'da gösterilmiştir. ResNet101 Ağ modelinde test edilen 13 farklı nesne sınıfına ait Tahmin Değerleri'nin ortalaması %91.1110; Doğruluk Oranları'nın ortalaması %97.7262 şeklinde bulunmuştur.

Tablo 3.26. ResNet101 Ağ Modeline Ait Test Sonuçları

Nesne Tipi	Test Edilen Görsel Sayısı	Doğru Tahmin Edilen Görsel Sayısı	Yanlış Tahmin Edilen Görsel Sayısı	Tahmin Değerlerinin Ortalaması (%)	Doğruluk Oranlarının Ortalaması (%)
Audi	90	83	7	92.2222	98.0898
Papatya	90	88	2	97.7777	99.0185
Karahindiba	90	87	3	96.6666	99.2655
İnsan	90	84	6	93.3333	99.6493
Hyundai Creta	90	82	8	91.1111	96.6642
Mahindra Scorpio	90	83	7	92.2222	98.9096
Rolls Royce	90	63	27	70	93.2390
Gül	90	80	10	88.8888	97.5930
Ay Çiçeği	90	77	13	85.5555	95.7458
Swift	90	85	5	94.4444	97.7193
Tata Safari	90	84	6	93.3333	95.9975
Toyota İnnova	90	80	10	88.8888	98.9590
Lale	90	90	0	100	99.5910
ORTALAMA				**91.1110**	**97.7262**

Tablo 3.27'de MobileNet Ağ modeline ait test sonuçlarını gösterilmektedir. Oluşturulan test sonuçları neticesinde toplam 1.170 adet görsel analiz edilmiş olup, 733 adet görsel doğru tahmin edilmiş, 437 adet görsel ise yanlış tahmin edilmiştir. Doğru tahmin edilen görsel sayısının, test edilen görsel sayısına oranı ile sistemin Tahmin Değerleri hesaplanmıştır. Sistemin Tahmin Değerleri'nin aritmetik ortalaması ve Doğruluk Oranları'nın aritmetik ortalaması alınmış Tablo 3.27'de gösterilmiştir. MobileNet Ağ modelinde test edilen 13 farklı nesne sınıfına ait Tahmin Değerleri'nin ortalaması %62.6495; Doğruluk Oranlarının ortalaması ise %88.2940 şeklinde bulunmuştur.

Tablo 3.27. MobileNet Ağ Modeline Ait Test Sonuçları

Nesne Tipi	Test Edilen Görsel Sayısı	Doğru Tahmin Edilen Görsel Sayısı	Yanlış Tahmin Edilen Görsel Sayısı	Tahmin Değerlerinin Ortalaması (%)	Doğruluk Oranlarının Ortalaması (%)
Audi	90	66	24	73.3333	89.2249
Papatya	90	58	32	64.4444	91.7259
Karahindiba	90	71	19	78.8888	91.9634
İnsan	90	72	18	80	96.8049
Hyundai Creta	90	25	65	27.7777	78.4166
Mahindra Scorpio	90	68	22	75.5555	91.5981
Rolls Royce	90	56	34	62.2222	89.3328
Gül	90	65	25	72.2222	78.4161
Ay Çiçeği	90	29	61	32.2222	82.7972
Swift	90	52	38	57.7777	91.6760
Tata Safari	90	46	44	51.1111	88.8917
Toyota İnnova	90	66	24	73.3333	93.5189
Lale	90	59	31	65.5555	83.4556
ORTALAMA				**62.6495**	**88.2940**

Tablo 3.28'de DenseNet121 Ağ modeline ait test sonuçlarını gösterilmektedir. Oluşturulan test sonuçları neticesinde toplam 1.170 adet görsel analiz edilmiş olup, 658 adet görsel doğru tahmin edilmiş, 512 adet görsel ise yanlış tahmin edilmiştir. Doğru tahmin edilen görsel sayısının, test edilen görsel sayısına oranı ile sistemin Tahmin Değerleri hesaplanmıştır. Sistemin Tahmin Değerleri'nin aritmetik ortalaması ve Doğruluk Oranları'nın aritmetik ortalaması alınmış Tablo 3.28'de gösterilmiştir. DenseNet121 Ağ modelinde test edilen 13 farklı nesne sınıfına ait Tahmin Değerleri'nin ortalaması %56.2392; Doğruluk Oranları'nın ortalaması %85.4810 şeklinde bulunmuştur.

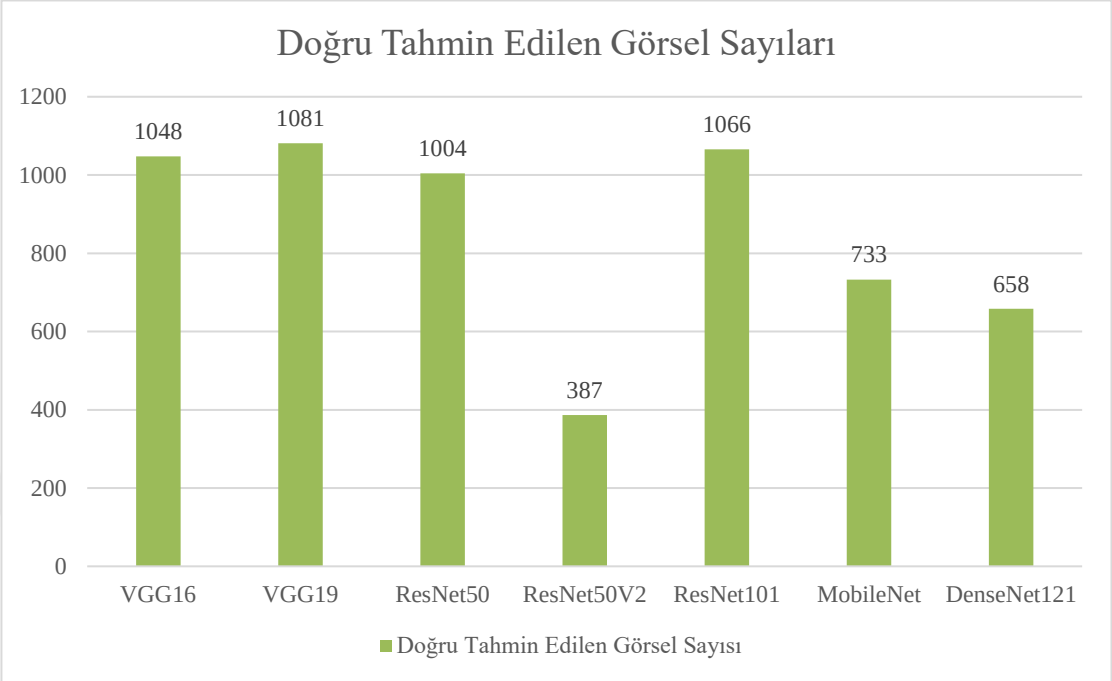
Tablo 3.28. DenseNet121 Ağ Modeline Ait Test Sonuçları

Nesne Tipi	**Test Edilen Görsel Sayısı**	**Doğru Tahmin Edilen Görsel Sayısı**	**Yanlış Tahmin Edilen Görsel Sayısı**	**Doğru Tahmin Edilen Görsel Sayısının Test Edilen Görsel Sayısına Oranı (%)**	**Ortalama Doğruluk Değeri (%)**
Audi	90	65	25	72.2222	80.5727
Papatya	90	54	36	60	90.9151
Karahindiba	90	74	16	82.2222	95.0094
İnsan	90	69	21	76.6666	93.3708
Hyundai Creta	90	34	56	37.7777	81.9858
Mahindra Scorpio	90	47	43	52.2222	82.9679
Rolls Royce	90	18	72	20	61.0136
Gül	90	71	19	78.8888	94.3236
Ay Çiçeği	90	2	88	2.2222	91.9587
Swift	90	48	42	53.3333	80.8328
Tata Safari	90	48	42	53.3333	81.6378
Toyota İnnova	90	66	24	73.3333	87.2868
Lale	90	62	28	68.8888	89.3786
ORTALAMA				**56.2392**	**85.4810**

Bundan sonraki kısımda test edilen ağ modellerine ait kıyaslama grafikleri verilecektir.

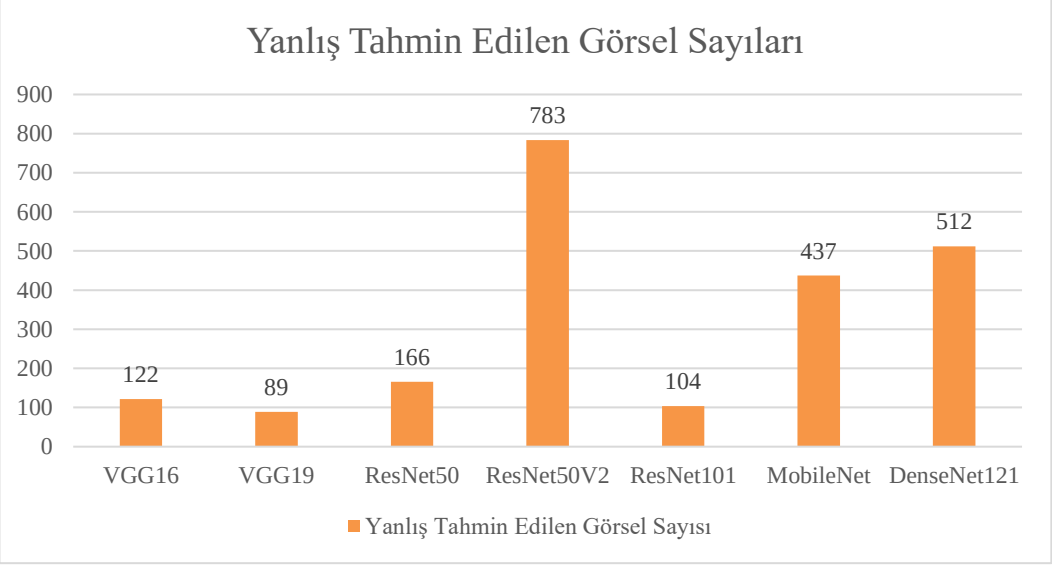
Tablo 3.29'da farklı ağ modellerinde test edilen 13 farklı nesnenin doğru tahmin edilen görsel sayıları karşılaştırılmıştır. VGG19 ağ modeli en fazla doğru tahmin edebilen ağ modeli olmuştur. ResNet50V2 ağ modeli ise en az doğru tahmin edebilen ağ modeli olmuştur. Doğru tahmin edilen görsel sayılarına göre eğitilen ağ modelleri kıyaslandığında VGG19 ağ modeli en başarılı ağ modeli olarak belirlenirken, ResNet50V2 ağ modeli en başarısız ağ modeli olarak belirlenmiştir.

Tablo 3.29. Ağ Modellerine Ait Doğru Tahmin Edilen Görsel Sayılarının Kıyaslanması



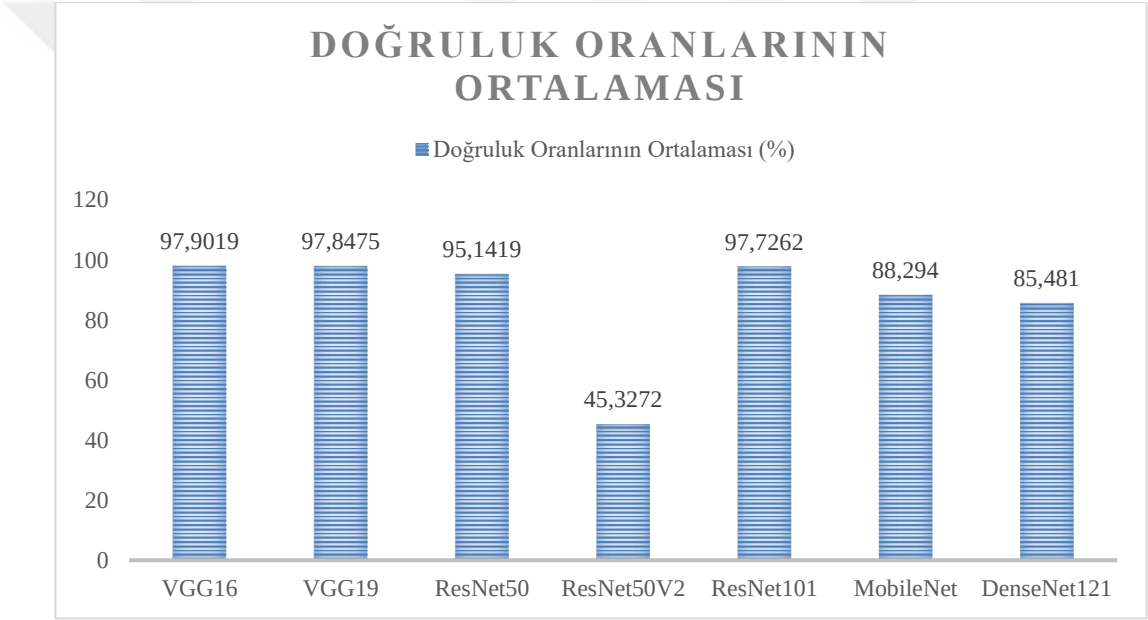
Tablo 3.30'da farklı ağ modellerinde test edilen 13 farklı nesnenin yanlış tahmin edilen görsel sayıları karşılaştırılmıştır. VGG19 ağ modeli en az yanlış tahminde bulunan ağ modeli olmuştur. ResNet50V2 ağ modeli ise en fazla yanlış tahminde bulunan ağ modeli olmuştur. Yanlış tahmin edilen görsel sayılarına göre eğitilen ağ modelleri kıyaslandığında VGG19 ağ modeli en başarılı ağ modeli olarak belirlenirken, ResNet50V2 ağ modeli en başarısız ağ modeli olarak belirlenmiştir.

Tablo 3.30. Ağ Modellerine Ait Yanlış Tahmin Edilen Görsel Sayılarının Kıyaslanması



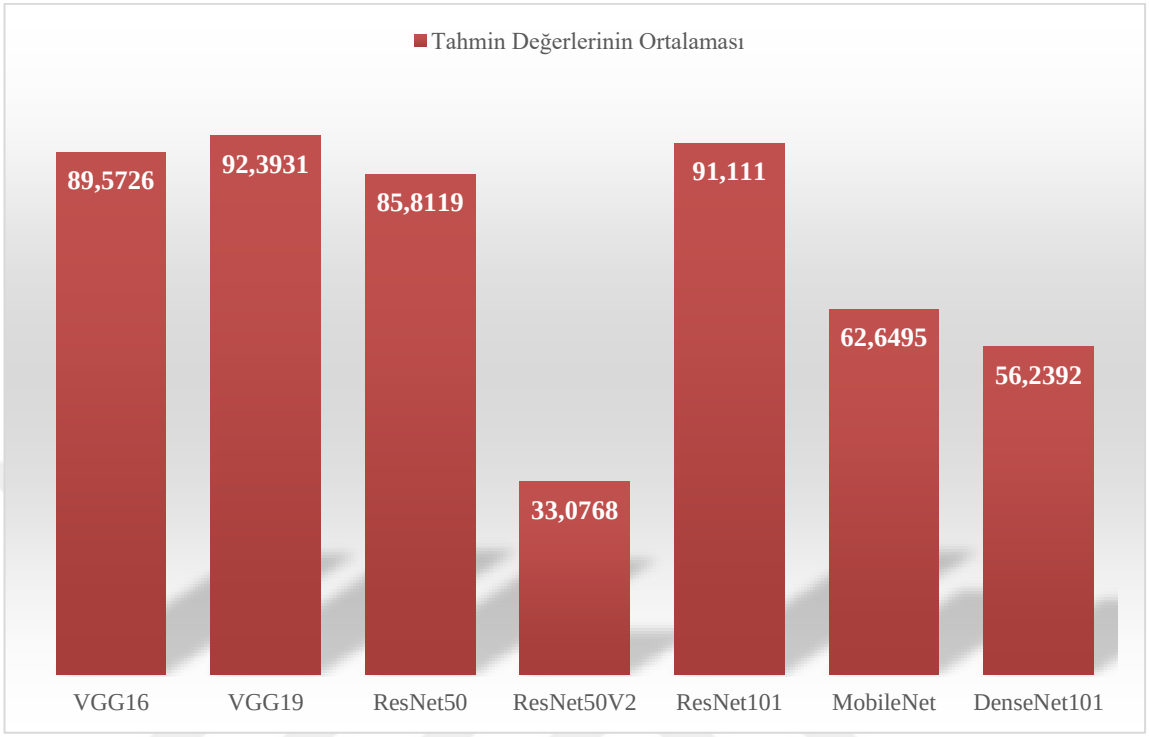
Tablo 3.31'de eğitilen ağ modellerinde test işlemi gerçekleştiren nesnelerin doğru tahmin edebilme oranlarının aritmetik ortalaması alınmış ve ilgili tabloda gösterilmiştir. Ağ modellerinin Doğruluk Oranları kıyaslandığında VGG16 Ağ Modeli %97.9019 gibi orana sahipken, VGG19 Ağ Modeli %97.8475 gibi bir orana sahiptir. Fakat VGG19 Ağ Modelinde 1081 görsel doğru analiz edilmişken, VGG16 ağ modelinde 1048 görsel doğru analiz edilmiştir. ResNet50V2 ağ modelinde ise %45.3272 gibi bir Doğruluk Oranı mevcut olup sistemin en başarısız ağ modeli olarak belirlenmiştir. Ağ modellerinin Doğruluk Oranları'nın ortalamasının kıyaslandığı Tablo 3.31'de gösterilmiştir.

Tablo 3.31. Doğruluk Oranlarının Aritmetik Ortalamasının Kıyaslanması

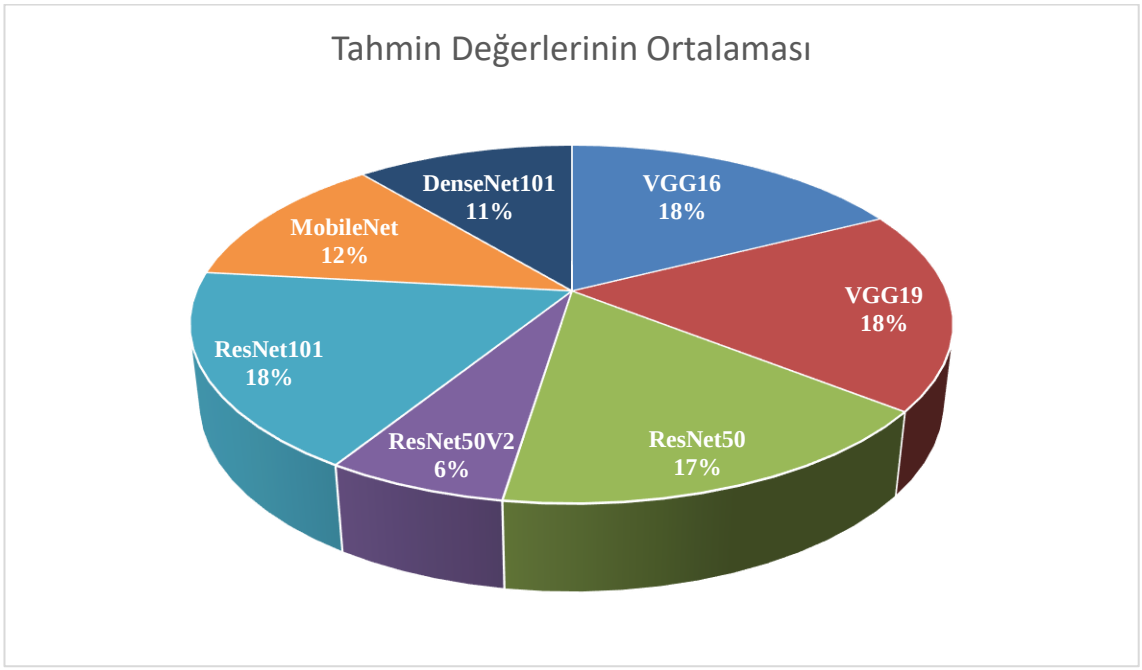


Ağ modellerinin Tahmin Değerleri'nin (doğru tahmin edilen görsel sayısının, test edilen görsel sayısına oranı) aritmetik ortalamasının kıyaslandığı grafikler Tablo 3.32 ve 3.33'te gösterilmiştir.

Tablo 3.32. Tahmin Değerlerinin Aritmetik Ortalamasının Kıyaslanması



Tablo 3.33. Tahmin Değerlerinin Aritmetik Ortalamasının Pasta Grafikte Gösterilmesi



4. BÖLÜM

TARTIŞMA-SONUÇ VE ÖNERİLER

Bilgisayar teknolojisinin hızla ilerlemesi ve yapay zekâ temelli nesne tanıma algoritmalarının gelişimi, literatürde önemli bir yer edinmiştir. Görüntü işleme uygulamalarında kullanılan nesne tanıma algoritmalarının sayısında belirgin bir artış gözlenmiştir. Yüksek boyutlu veri yapılarına sahip görüntü işleme uygulamalarında etkili sonuçlar elde edebilen güçlü yaklaşımların geliştirilmesi gerekmektedir.

Python programlama dilini kullanarak geliştirilen Derin Öğrenme algoritmaları, açık kaynak kodlu ve geniş kütüphane desteği ile birlikte, Büyük Veri içeren problemlerin çözümünde yaygın bir biçimde tercih edilmektedir.

Bu çalışmada, açık kaynak kodlu kütüphaneden oluşan Keras Kütüphanesi ile sinir ağları oluşturulması hedeflenmiş, veri akış grafikleri kullanarak numerik hesaplamalar yapan TensorFlow kütüphanesi üzerinde çalıştırılmıştır. Bilgisayarlı görüntü sistemlerinde etkin çözümler üreten nesne algılama uygulamaları analiz edilmiştir.

ImageNet veri setinden faydalanarak, internet tarayıcısından özel bir veri seti hazırlanmıştır. Hazırlanan veri seti üzerinde derin öğrenme modelleri ile çalışılmıştır. 7 farklı ağ modeli; VGG16, VGG19, ResNet50, ResNet5022, ResNet101, MobileNet ve DenseNet121 eğitilmiştir. Eğitilen modeller üzerindeki Doğruluk Oranı, Hata Oranı, Tahmin Değerleri ve Öğrenme Süreleri mukayeseleri yapılmıştır. Ağ modeller üzerinde analizler gerçekleştirilmiştir.

Eğitim üzerinde önceden eğitilmiş model kullanıldığı için ana katmanlar eğitilemez hale getirilmiştir. Model üzerindeki Öğrenme Oranı 0.001 şeklinde belirlenmiş olup 10 adım sayısında eğitilmiştir.

Eğitilen modeller ile birlikte 13 farklı nesnenin görselleri test uygulamaları gerçekleştirilmiştir. Gerçekleştirilen uygulamaların kıyaslamaları yapılmış ve analizleri gerçekleştirilmiştir.

Gerçekleştirilen analizler neticesinde sistem üzerinde 13 farklı nesneye ait toplam 3900 adet görsel kullanılmıştır. Kullanılan görsellerin %70'i öğrenme verisi olarak sistemde eğitilmiş, geriye kalan %30'luk görsel ise test verisi olarak kullanılmıştır.

Test sonuçlarında 1081 nesne görselinin doğru tahmin edilip, 89 nesne görselin yanlış tahmin edilmesi ile %97.8475 gibi bir Doğruluk Oranı ve %92.3931 gibi bir Tahmin Değeri ile analizleri bitiren VGG19 ağ modeli çalışmamızın en başarılı ağ modeli olarak seçilmişken, 387 nesne görselinin doğru tahmin edilip, 783 nesne görselinin yanlış tahmin edilmesi ile %45.3272 gibi bir Doğruluk Oranı ve %33.0768 gibi bir Tahmin Değeri ile analizleri bitiren ResNet50V2 ağ modeli çalışmamızın en başarısız ağ modeli olarak seçilmiştir.

Yapmış olduğumuz çalışma kapsamında, farklı nesne gruplarında derin öğrenme algoritmaları kullanılarak yüksek sonuçlar elde edilmiş, başarılı sonuçlar sergilendiği gözlemlenmiştir.

Evrişimsel Sinir Ağları özellikle görüntü tanıma ve görüntü işleme uygulamalarında kullanılmakta olup, güvenlik sanayii, savunma sanayii ve sağlık sektörlerinde oldukça sık kullanıldığı saptanmıştır.

KAYNAKÇA

1. LeCun, Bengio, Hinton, 2015. Deep learning. **Nature, 521**: 436-444

2. Lee, Grosse, Ranganath, Ng., 2009. Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations. 609-616. *Proceedings of the 26th Annual International Conference on Machine Learning*, 14-18 June 2009, Montreal Quebec Canada, ICML, 1285.

3. Wıldml, 2016. Understanding convolutional neural networks for nlp,. (Web Sayfası: **http://www.wildml.com/2015/11/understanding-convolutional-neural-networks-fornlp/**), (Erişim tarihi: Eylül 2023).

4. Viola, Jones., 2001. Rapid object detection using a boosted cascade of simple features. *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition CVPR 2001*, 08-14 December 2001, Kaunai HI USA, IEEE.

5. Rosenblatt, 1957. The perceptron a perceiving and recognizing automaton. **Cornell Aeronautical Laboratory**, 85-60-1.

6. Ivakhnenko, Lapa, 1965. Cybernetic predicting devices. **Purdue University School of Electrical Engineering**, 250.

7. Fukushima, 1980. Neocognitron: a self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. **Biological Cybernetics by Springer**, **36**: 193-202.

8. Rojas, 1996. The backpropagation algorithm. Neural networks, 149-182. In: A Systematic İntroduction. Springer, Berlin, Heidelberg.

9. Shone, Ngoc, Phai, Şi, 2018. A deep learning approach to network intrusion detection. **IEEE transactions on emerging topics in computational intelligence**, **2** (1): 41-50.

10. Haque, Neubert, 2020. Deep learning approaches to biomedical image segmentation. **Informatics in Medicine Unlocked**, **18**: 100297.

11. Lv, Duan, Kang, Li, Wang, 2014. Traffic flow prediction with big data: A deep learning approach. **IEEE Transactions on Intelligent Transportation Systems**, **16** (2): 865-873.

12. LeCun., Bottou., Bengio., Haffner., 1998. Gradient based learning applied to document recognition, **Proceeding of the IEEE**, **86** (11): 2278-2324.

13. Asif, Rahman, Hena, 2020. CNN based disease detection approach on potato leaves, 428-432. *In 2020 3rd International Conference on Intelligent Sustainable Systems (ICISS)*, 3-5 December 2020, Thoothukudi, India, IEEE.

14. Momeny, Jahanbakhshi, Jafarnezhad, Zhang, 2020. Accurate classification of cherry fruit using deep CNN based on hybrid pooling approach. **Postharvest Biology and Technology**, **166**: 111204.

15. Kumar, Arora, Sisodia, 2020. Resnet-based approach for detection and classification of plant leaf diseases, 495-502. *In 2020 international conference on electronics and sustainable communication systems (ICESC)*, 2-4 July 2020, Coimbatore, India, IEEE.

16. Kodipalli, Guha, Dasar, Ismail, 2022. An inception-ResNet deep learning approach to classify tumours in the ovary as benign and malignant. **Expert Systems**, e13215.

17. Lin, Zhao, Yang, Yin, Wang, Guo, Chen, Ma, Luo, Ding, Pang, 2020. Cir-net: automatic classification of human chromosome based on inception-resnet architecture. **IEEE/ACM Transactions on Computational Biology and Bioinformatics**, **19** (3): 1285-1293.

18. Xu, Lin, Yu, Tai, Chang, 2018. Acoustic scene classification using reduced MobileNet architecture. 267-270. *In 2018 IEEE International Symposium on Multimedia (ISM)*, 10-12 December 2018, Taichung, Taiwan, IEEE.

19. Sabour, Frosst, Hinton, 2017. Dynamic routing between capsules, 3859-3869. *31st Conference on Neural Information Processing Systems (NIPS'17)*, 04 December 2017, Long Beach CA USA.

20. Toraman, 2018. Derin öğrenme ile insansız hava aracı görüntülerinden Yaya Tespiti. **Journal of Aviation**, **2** (2): 64-69.

21. Viola, Jones, 2001. Rapid object detection using a boosted cascade of simple features. *Proceedings of the 2001 IEEE Computer Society Conference on*, 08-14 December 2001, Kaunai HI USA, IEEE.

22. Daş, Polat, Tuna, 2019. Derin öğrenme ile resim ve videolarda nesnelerin tanınması ve takibi. **Fırat Üniversitesi Mühendislik Bilimleri Dergisi**, **31(2)**: 571-58.1

23. Savaş, 2022. Derin öğrenme (deep learning) nedir? nasıl çalışır?. (Web sayfası: **https://www.gulaysavas.com.tr/derin-ogrenme-deep-learning-nedir-nasil-calisir/**), (Erişim tarihi: Eylül 2023).

24. Okumuş, 2019. Convolutional neural networks (evrişimsel sinir ağları). (Web sayfası: **https://medium.com/@rabiaokumus96/convolutional-neuralnetworks-evri%C5%9Fimsel-sinir-a%C4%9Flar%C4%B1-cceb887a2979**), (Erişim tarihi: Eylül 2023).

25.Weisstein, 2019. Convolution. (Web sayfası: **http://mathworld.wolfram.com/Convolution.html**), (Erişim tarihi: Eylül 2023).

26. Satar, 2018. Deep Learning Based Vehicle Make And Model Classification. Uludağ Üniversitesi Fen Bilimleri Enstitüsü, Yüksek Lisans Tezi, Bursa, 80 s.

27. Çarkacı, 2018. Derin öğrenme uygulamalarında en sık kullanılan hiperparametreler. (Web sayfası: **https://medium.com/deep-learning-turkiye/derin-ogrenmeuygulamalarinda-en-sik-kullanilan-hiper-parametreler-ece8e9125c4**), (Erişim tarihi: Eylül 2023)

28. Esen, Onan, 2022. Derin öğrenme yöntemleri ile bitki yaprakları üzerindeki hastalıkların sınıflandırılması. **Avrupa Bilim ve Teknoloji Dergisi**, **40**: 151-155.

29. He, Zhang, Ren, Sun, 2016. Deep residual learning for image recognition, 770-778. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 27-30 June 2016, Las Vegas, NV, USA.

30. Howard, Zhu, Chen, Kalenichenko, Wang, Weyand, Andreetto, Adam, 2017. MobileNets: efficient convolutional neural networks for mobile vision applications. (Web sayfası: **https://www.arxiv-vanity.com/papers/1704.04861/**), (Erişim tarihi: Eylül 2023)

31. Huang, Liu, Maaten, Weinberger, 2017. Densely connected convolutional networks, 2261-2269. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2017)*, 21-26 July 2017, Honolulu, HI, USA, 7506 pp.

32. Maptriks, 2020. Uzmanından örneklerle tensorflow nedir açıklıyoruz?, (Web sayfası: **https://maptriks.com/uzmanindan-orneklerle-tensorflow-nedir-acikliyoruz/#:~:text=TensorFlow%2C%20Mobil%20uygulamalardan%20IoT%20cihazlar%C4%B1na,sayesinde%20%C3%A7ok%C3%A7a%20kullan%C4%B1lan%20bir%20k%C3%BCt%C3%BCphanedir.),** (Erişim tarihi: Eylül 2023)

33. Doğan, 2020. Keras kütüphanesi nedir? derin öğrenme modeli oluşturma, (Web sayfası: **https://teknoloji.org/keras-kutuphanesi-nedir-derin-ogrenme-modeli-olusturma/**), (Erişim tarihi: Eylül 2023)

ÖZGEÇMİŞ

KİŞİSEL BİLGİLER

Adı Soyadı: Eda GALİP
Uyruğu: Türkiye (T.C)
Doğum Tarihi ve Yeri:
Medeni Durum:
E-mail:

EĞİTİM

Derece	Kurum	Mezuniyet Tarihi
Yüksek Lisans	Erciyes Üniversitesi, Mekatronik Mühendisliği	2024
Lisans	Erciyes Üniversitesi, Mekatronik Mühendisliği	2020
Lisans	Erciyes Üniversitesi, Endüstriyel Tasarım Mühendisliği	2020
Lise	Açık Öğretim Lisesi, Ankara	2015

İŞ DENEYİMLERİ

Yıl	Kurum	Görev
2022 - Halen	Türkiye İş Bankası A. Ş.	Memur
2021 - 2021	Ceotekmer Girişim Evi A. Ş.	Proje Uzmanı

YABANCI DİL

İngilizce

YAYINLAR

1. Çetinkaya, Galip, 2022. Image recognition based on deep learning, 1704 s. *12. Uluslararası Mühendislik Mimarlık ve Tasarım Kongresi*, 23-24-25 Aralık 2023, İstanbul, Türkiye