



**DERİN ÖĞRENME TABANLI ŞİDDETLİ  
FARENJİT TESPİTİ**

**2023  
YÜKSEK LİSANS TEZİ  
BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI**

**Tuğba TAK**

**Tez Danışmanı  
Doç. Dr. Hakan KUTUCU**

**DERİN ÖĞRENME TABANLI ŞİDDETLİ FARENJİT TESPİTİ**

**Tuğba TAK**

**Tez Danışmanı  
Doç. Dr. Hakan KUTUCU**

**T.C.  
Karabük Üniversitesi  
Lisansüstü Eğitim Enstitüsü  
Bilgisayar Mühendisliği Anabilim Dalında  
Yüksek Lisans Tezi  
Olarak Hazırlanmıştır**

**KARABÜK  
Ağustos 2023**

Tuğba TAK tarafından hazırlanan “DERİN ÖĞRENME TABANLI ŞİDDETLİ FARENJİT TESPİTİ” başlıklı bu tezin Yüksek Lisans Tezi olarak uygun olduğunu onaylarım.

Doç. Dr. Hakan KUTUCU .....

Tez Danışmanı, Yazılım Mühendisliği Anabilim Dalı

Bu çalışma, jürimiz tarafından Oy Birliği ile Bilgisayar Mühendisliği Anabilim Dalında Yüksek Lisans tezi olarak kabul edilmiştir. 04/08/2023

Ünvanı, Adı SOYADI (Kurumu)

İmzası

Başkan : Doç. Dr. İlker TÜRKER (KBÜ) .....

Üye : Doç. Dr. Hakan KUTUCU (KBÜ) .....

Üye : Dr. Öğr. Üyesi Bayram AKGÜL (BARÜ) .....

KBÜ Lisansüstü Eğitim Enstitüsü Yönetim Kurulu, bu tez ile, Yüksek Lisans derecesini onamıştır.

Prof. Dr. Müslüm KUZU .....

Lisansüstü Eğitim Enstitüsü Müdürü

*“Bu tezdeki tüm bilgilerin akademik kurallara ve etik ilkelere uygun olarak elde edildiğini ve sunulduğunu; ayrıca bu kuralların ve ilkelerin gerektirdiği şekilde, bu çalışmadan kaynaklanmayan bütün atıfları yaptığımı beyan ederim.”*

Tuğba TAK

## ÖZET

**Yüksek Lisans Tezi**

### **DERİN ÖĞRENME TABANLI ŞİDDETLİ FARENJİT TESPİTİ**

**Tuğba TAK**

**Karabük Üniversitesi**

**Lisansüstü Eğitim Enstitüsü**

**Bilgisayar Mühendisliği Anabilim Dalı**

**Tez Danışmanı:**

**Doç. Dr. Hakan KUTUCU**

**Ağustos 2023, 49 sayfa**

Günümüzde makine öğrenmesi ve özellikle derin öğrenme her alanda olduğu gibi, sağlık alanında da yaygın olarak kullanılmaya başlanmıştır. Derin öğrenmede de makine öğrenmesinde olduğu gibi kullanılan verilerin kalitesi, verilerin miktarının çok olması kadar önem arz eder. Aynı zamanda medikal verilere erişim hasta mahremiyeti ve verilerin toplanması gibi birçok nedenden ötürü zordur. Bu bağlamda küçük veri setleri ile eğitim yapılırken transfer öğrenimi uygulamak işleri kolaylaştıracaktır. Farenjit hastalığı, hemen hemen her yaş aralığında gözlemlenebilen bir hastalıktır. Bu çalışmadaki amacımız derin öğrenme tabanlı algoritmalar kullanılarak, küçük veri setlerine sahip olsak dahi kript içeren şiddetli farenjit hastalığının yüksek başarı oranları ile tespitini sağlamak ve aynı veri seti ile çalışılmış diğer algoritmalar ile performans metrikleri üzerinden kıyaslama yapmaktır. Bu çalışma geliştirilebilecek medikal uygulamalar için kullanılacak derin öğrenme algoritmalarının performansını karşılaştırdığından önem arz eder. Çalışmada kullanılan görüntü verilerinin az olması nedeniyle, ham veriler öncelikle geleneksel yöntemler ve afin

dönüşümleri kullanılarak artırılmış ve daha sonra artırılan görüntüler, derin öğrenme yaklaşımı olan GAN veri artırımı yöntemlerinden, CycleGAN algoritması ile eşlenmiş görüntüler elde edilecek şekilde tekrar çoğaltılmıştır. Sentetik görüntülerin her biri çalışmada kullanılmamıştır. Veri setine dahil edilecek görüntüler uzman bir tıp doktoru tarafından manuel şekilde sınıflandırılıp, seçilmiş ve görüntü kalitesine dikkat edilerek veri setine eklenmiştir. Çalışmada ResNet50, Inception-V3, EfficientNet grubu içerisinde olan EfficientNet B0, EfficientNet B1 ve EfficientNet B2 mimarileri kullanarak sınıflandırma yapılmıştır. Çalışmada Inception-V3 algoritması için ön eğitim, diğer tüm algoritmalar için transfer öğrenimi gerçekleştirilmiştir. Kullanılan ham veri seti, beş farklı geleneksel artırma yöntemi ile çoğaltılmıştır. Sağlıklı boğaz görüntüleri 1283 ve farenjitli boğaz görüntüleri 873 adet olarak belirlenmiştir. Ardından CycleGAN algoritmasından elde ettiğimiz veri doğruluğunun korunduğu ve görüntü kalitesi yüksek olan 68 adet sağlıklı ve 171 adet farenjitli sentetik görüntü seçilerek veri setine eklenmiştir. Veri setini üç bölüme ayırdık: Eğitim (0.8), doğrulama (0.1) ve test (0.1). Eğitimler tamamlandıktan sonra, Inception-V3 mimarisi için elde ettiğimiz test veri seti doğruluğu %97,51, ResNet50 mimarisi için elde ettiğimiz test veri seti doğruluğu %98,76, EfficientNet B0 mimarisi için elde ettiğimiz test veri seti doğruluğu %99,17, EfficientNet B1 mimarisi için elde ettiğimiz test veri seti doğruluğu %99,59 ve son olarak EfficientNet B2 mimarisi için elde ettiğimiz test veri seti doğruluğu %99,59 olarak gözlemlenmektedir. Mimariler hassasiyet, geri çağırma, AUC ve kayıp değerleri bakımından karşılaştırıldığında EfficientNet grubunun aynı veri seti ile çalışılan diğer mimariler ve bizim çalışmamızdaki ResNet50 ve Inception-V3 mimarilerine göre performans bakımından daha başarılı olduğu gözlemlenmiştir.

**Anahtar Sözcükler :** Makine öğrenmesi, Farenjit, ResNet50, InceptionV3, EfficientNet, CycleGan.

**Bilim Kodu :** 92431

## **ABSTRACT**

**M. Sc. Thesis**

### **DEEP LEARNING BASED SEVERE PHARYNGITIS DETECTION**

**Tuğba TAK**

**Karabük University**

**Institute of Graduate Programs**

**Department of Computer Engineering**

**Thesis Advisor:**

**Assoc. Prof. Dr. Hakan KUTUCU**

**August 2023, 49 pages**

Nowadays, machine learning and especially deep learning have become widely used in the field of health, as in every field. In deep learning, as in machine learning, the quality of the data used is as important as the quantity of data. At the same time, access to medical data is difficult for many reasons such as patient privacy and data collection. In this context, applying transfer learning when training with small data sets will make things easier. Pharyngitis is a disease that can be observed in almost every age range. Our aim in this study is to detect cryptic severe pharyngitis with high success rates using deep learning based algorithms, even with small data sets, and to compare performance metrics with other algorithms that have been studied with the same data set. This study is important because it compares the performance of deep learning algorithms that can be used for medical applications that can be developed. Due to the small amount of image data used in the study, the raw data was first augmented using traditional methods and affine transformations, and then the augmented images were re-duplicated to obtain mapped images with the CycleGAN algorithm, one of the GAN

data augmentation methods, which is a deep learning approach. Not all synthetic images were used in the study. The images to be included in the dataset were manually classified and selected by an expert medical doctor and added to the dataset by paying attention to image quality. In the study, classification was performed using ResNet50, Inception-V3, EfficientNet B0, EfficientNet B1 and EfficientNet B2 architectures within the EfficientNet group. In the study, pre-training was performed for the Inception-V3 algorithm and transfer learning was performed for all other algorithms. The raw data set was augmented with five different traditional augmentation methods. The number of healthy throat images was 1283 and the number of pharyngitis throat images was 873. Then, 68 healthy and 171 pharyngitis synthetic images were selected and added to the dataset, which preserved the accuracy of the data obtained from the CycleGAN algorithm and were of high quality. We divided the dataset into three parts: Training (0.8), validation (0.1) and testing (0.1). After training, we observe that the test data set accuracy for the Inception-V3 architecture is 97.51%, the test data set accuracy for the ResNet50 architecture is 98.76%, the test data set accuracy for the EfficientNet B0 architecture is 99.17%, the test data set accuracy for the EfficientNet B1 architecture is 99.59% and finally the test data set accuracy for the EfficientNet B2 architecture is 99.59%. When the architectures are compared in terms of precision, recall, AUC and loss values, it is observed that the EfficientNet group is more successful in terms of performance than the other architectures studied with the same dataset and the ResNet50 and Inception-V3 architectures in our study.

**Key Word** : Machine learning, Pharyngitis, ResNet50, InceptionV3, EfficientNet, CycleGan.

**Science Code** : 92431



## TEŐEKKÜR

Bu tez alıőmasının planlanmasında, araőtırılmasında, yürütülmesinde ve oluşumunda ilgi ve desteęini esirgemeyen, sayın hocam Do. Dr. Hakan KUTUCU'ya teőekkürlerimi sunarım.

Ayrıca tüm yaşamım boyunca aldığım her kararda beni destekleyen, akıl hocam olan dedem Hasan Tahsin TAK'a ve őefkatini üzerimden eksik etmeyen canım annem Sevgi TAK'a içtenlikle teőekkür ederim.

Akademik olarak hemen her adımımı danıőabildiğim, pratik zekasına güvendiğim yüksek lisans dönem arkadaşım Arő. Gör. Muhammed Yusuf KÜÇÜKKARA'ya samimiyetle teőekkür ederim.

Bu stresli dönemimde her an yanımda olan, beni sakinleőtiren en yakın dostum aynı zamanda niőanlım Fatih SAVAŐ'a ve bu tezi yazmama yol aan, beni dürüstlüęü ve sevgisiyle büyüleyen, varlığını ve desteęini tam kalbimde hissettiğim, en büyük sırdaőım biricik babam İdris TAK'a sevgilerimi sunarım.

## İÇİNDEKİLER

	<u>Sayfa</u>
KABUL.....	ii
ÖZET.....	iv
ABSTRACT.....	vi
TEŞEKKÜR.....	viii
İÇİNDEKİLER .....	ix
ŞEKİLLER DİZİNİ.....	xiii
ÇİZELGELER DİZİNİ .....	xiviv
KISALTMALAR DİZİNİ.....	xvv
BÖLÜM 1 .....	1
GİRİŞ .....	1
BÖLÜM 2 .....	2
YAPAY ZEKA, MAKİNE ÖĞRENMESİ VE DERİN ÖĞRENME.....	2
2.1. MAKİNE ÖĞRENİMİ TÜRLERİ .....	4
2.1.1. Denetimli Öğrenme .....	4
2.1.2. Denetimsiz Öğrenme .....	5
2.1.3. Yarı Denetimli Öğrenme .....	5
2.1.4. Takviyeli Öğrenme .....	5
2.2. MODEL OLUŞTURMA AŞAMALARI.....	6
2.2.1. Veri Ön İşleme.....	6
2.2.1.1. Girdiyi Belirli Bir Çözünürlüğe Göre Yeniden Boyutlandırma .....	7
2.2.1.2. Beyazlatma Dönüşümü .....	7
2.2.1.3. Veri Artırma.....	7
2.2.2. Mimari Seçimi .....	7
2.2.2. Hiper Parametre Ayarları ve Metrikler.....	8
2.3. MODEL GENELLENEBİLİRLİĞİ NASIL SAĞLANIR? .....	9
2.3.1. Aşırı Uyum (Overfitting) Engelleme Yöntemleri.....	10

	<b><u>Sayfa</u></b>
2.3.1.1. Transfer Öğrenimi (Transfer Learning) .....	10
2.3.1.2. Bırakma (Dropout).....	12
2.3.1.3. Ön Eğitim (Pre-Training).....	13
2.3.1.4. Erken Durdurma (Early Stopping).....	13
BÖLÜM 3 .....	15
FARENJİT .....	15
3.1. LİTERATÜR TARAMASI.....	15
BÖLÜM 4 .....	20
VERİ ARTIRMA .....	20
4.1. GELENEKSEL VERİ ARTIRIMI.....	21
4.1.1. Geometrik Dönüşümler .....	21
4.1.1.1. Çevirme (Flipping).....	22
4.1.1.2. Renk Alanı .....	22
4.1.1.3. Döndürme (Rotation) .....	22
4.1.1.4. Yakınlaştırma (Zoom).....	23
4.2. GAN TABANLI VERİ ARTIRMA .....	23
4.2.1. CycleGAN .....	24
BÖLÜM 5 .....	26
DNN ALGORİTMALARI.....	26
5.1. RESNET .....	26
5.2. INCEPTION .....	28
5.3. EFFICIENTNET .....	29
BÖLÜM 6 .....	30
METODOLOJİ .....	30

	<b><u>Sayfa</u></b>
6.1. VERİ SETİ .....	30
6.2. RESNET50 DENEYSEL ORTAM.....	32
6.2.1. Deneysel Sonuçlar .....	33
6.3. INCEPTION-V3 DENEYSEL ORTAM.....	34
6.3.1. Deneysel Sonuçlar .....	35
6.4. EFFICIENTNET B0 DENEYSEL ORTAM .....	36
6.4.1. Deneysel Sonuçlar .....	37
6.5. EFFICIENT B1 DENEYSEL ORTAM .....	38
6.5.1. Deneysel Sonuçlar .....	39
6.6. EFFICIENT B2 DENEYSEL ORTAM .....	40
6.6.1. Deneysel Sonuçlar .....	40
BÖLÜM 7 .....	42
TARTIŞMA VE SONUÇLAR .....	42
KAYNAKLAR .....	45
ÖZGEÇMİŞ .....	49

## ŞEKİLLER DİZİNİ

### Sayfa

Şekil 2.1. Yapay zeka ve Derin öğrenme arasındaki hiyerarşi.....	2
Şekil 2.2. Yapay sinir ağı şeması .....	3
Şekil 2.3. Makine öğrenmesi ve derin öğrenme özellik çıkarımı.....	3
Şekil 2.4. Denetimli, denetimsiz ve yarı denetimli öğrenme. ....	5
Şekil 2.5. Takviyeli öğrenme temsili.....	6
Şekil 2.6. Regresyon ve sınıflandırma problemleri için aşırı uyum, uyumsuzluk ve optimal durum şeması .....	10
Şekil 2.7. Aşırı uyum (overfitting) problemi.....	10
Şekil 2.8. Geleneksel makine öğrenimi ve transfer öğrenimi uygulanması.....	11
Şekil 2.9. Bırakma (dropout) temsili .....	13
Şekil 3.1. Bulanık Mamdani yöntemi ve e-öğrenme sisteminin entegre edilmesi ...	17
Şekil 3.2. Çalışma özeti şeması .....	18
Şekil 3.3. Çalışmada kullanılan aygıt ve kullanımının gösterimi.....	19
Şekil 3.4. Çalışmada kullanılan üç katmanlı sinir ağı temsili .....	19
Şekil 4.1. Veri artırma taksonomisi.....	21
Şekil 4.2. GAN konsepti çizimi.....	23
Şekil 4.3. Döngü tutarlılık kaybı şeması .....	24
Şekil 4.4. CycleGAN ile üretilmiş at ve zebra örneği .....	25
Şekil 5.1. CIFAR-10 ile eğitilmiş 20 ve 56 katmanlı düz ağların eğitim ve test hataları.....	27
Şekil 5.2. Artık blok örneği .....	27
Şekil 5.3. ResNet34 mimarisi.....	27
Şekil 5.4. Google Inception-V3 mimarisi .....	28
Şekil 5.5. EfficientNet B0 temsili .....	29
Şekil 6.1. Geleneksel veri artırma yöntemleri.....	31
Şekil 6.2. Veri setinde kullanılan gerçek ve sentetik görüntüler.....	32
Şekil 6.3. ResNet50 eğitim grafikleri.....	34
Şekil 6.4. Inception-V3 eğitim grafikleri .....	36
Şekil 6.5. EfficientNet B0 eğitim grafikleri .....	37
Şekil 6.6. EfficientNet B1 eğitim grafikleri .....	39

**Sayfa**

Şekil 6.7. EfficientNet B2 eğitim grafikleri ..... 41

## ÇİZELGELER DİZİNİ

	<b><u>Sayfa</u></b>
Çizelge 6.1. Veri seti örnek sayıları .....	31
Çizelge 6.2. ResNet50 hiper parametre denemeleri.....	33
Çizelge 6.3. ResNet50 performans metrikleri.....	34
Çizelge 6.4. Inception-V3 hiper parametreleri ve sonuçlar .....	35
Çizelge 6.5. Inception-V3 performans metrikleri .....	36
Çizelge 6.6. EfficientNet B0 hiper parametreleri.....	37
Çizelge 6.7. EfficientNet B0 performans metrikleri .....	38
Çizelge 6.8. EfficientNet B1 hiper parametreleri.....	39
Çizelge 6.9. EfficientNet B1 performans metrikleri .....	39
Çizelge 6.10. EfficientNet B2 hiper parametreleri.....	40
Çizelge 6.11. EfficientNet B2 performans metrikleri .....	41
Çizelge 7.1. İlgili alanda çalışılan mimari performans metrikleri .....	44

## KISALTMALAR DİZİNİ

### KISALTMALAR

AI	:	Artificial Intelligence (Yapay Zekâ)
CNN	:	Convolution Neural Network (Konvolüsyonel Sinir Ağları)
DL	:	Deep Learning (Derin Öğrenme)
GAN	:	Generative Adversarial Networks (Çekişmeli Üretici Ağlar)
GPU	:	Graphics Processing Unit (Grafik İşlemci Ünitesi)
ILSVRC:		ImageNet Large Scale Visual Recognition Challenge (ImageNet Büyük Ölçekli Görsel Tanıma Yarışması)
K-NN	:	K-Nearest Neighbours (K- En Yakın Komşu Algoritması)
ML	:	Machine Learning (Makine Öğrenmesi)
MSE	:	Mean Squared Error (Ortalama Karesel Hata)
NN	:	Neural Networks (Sinir Ağları)
RGB	:	Red Green Blue (Kırmızı Yeşil Mavi)
RST	:	Rapid Strep Test (Hızlı Boğaz Testi)
YSA	:	Yapay Sinir Ağları



## BÖLÜM 1

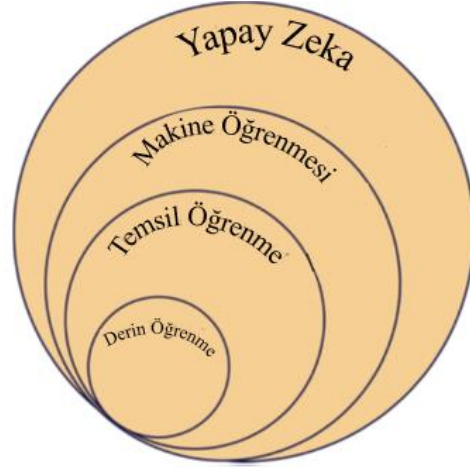
### GİRİŞ

Yapay zekanın günlük hayatımıza girmesiyle birlikte birçok farklı alanda kullanılmaya başlanması da kaçınılmazdır. Medikal alanda da iş yükünü hafifletmek amacıyla yapay zeka üzerine yapılan çalışmalar mevcuttur [1]. Makine öğrenmesi çalışılırken veri setinin büyüklüğü ağın performansını doğrudan etkilemektedir. Çalışmalarda kullanılan veri setleri milyonlarca veri içerebilir. Örneğin ImageNet veri seti 1000 kategoriden oluşan ve 1.2 milyon veri içeren bir görüntü veri kümesidir [2]. Fakat medikal verilere ulaşım hem hastanın haklarının ihlal edilemeyeceğinden hem de hastalığın özellikle dönemsel olmamasından dolayı zordur. Bu nedenle küçük ve orta boyutlu veri setleri ile eğitim gerçekleştirilebilmesi önem kazanmaktadır. Küçük veri setleri ile çalışırken kullanılan ağın önceden eğitilmiş olması ağın başarısını artırmaktadır. Alanda yaptığımız literatür taramasında çalışmamıza temel olan ve çalışmamızda, ham veri setini açık şekilde paylaşarak kullanmamıza olanak sağlayan 2020 yılında yapılan çalışmada [3], farenjit hastalığının tespiti Resnet-50 (%95,3), Inception-V3 (%94,9), MobileNet-V2 (%93,7) mimarileri ile çalışılmış ve belirtilen doğruluk oranları ile başarı sağlanmıştır [3]. 2022 yılında yapılan bir çalışmada ise konvolüsyonel sinir ağları (CNN) ve Bulanık Mamdani yöntemi kullanılarak %95,3'lük doğrulukla bir başarı elde edilmiştir [4]. Yine 2022 yılında yapılan diğer bir çalışmada ise farenjit hastalığı VGG16, VGG19, DenseNet, AlexNet ve ResNet-152 mimarileri kullanılarak artırılmış görüntüler ile VGG16 ve VGG19 mimarilerinde %98'lik bir doğruluk oranı ile başarı sağlanmıştır [5]. Çalışmamızda medikal uygulamalarda kullanılması açısından ResNet-50, Inception-V3 ve EfficientNet B0, EfficientNet B1 ve EfficientNet B2 mimarileri kullanıldı. EfficientNet mimarisinin çalışma prensibi gereği genişlik, yükseklik ve derinliğin farklı olarak kullanılabilirdiği B0-B7 arasında değişen sekiz mimarisi vardır. Mimarilerde ölçek arttıkça ağın karmaşıklığı da artmaktadır [6]. Bu sebeple çalışmada kullanılan her mimari kendi özellikleri temel alınarak hiper parametre ve veri ön hazırlığı yapılarak kullanılmıştır.

## BÖLÜM 2

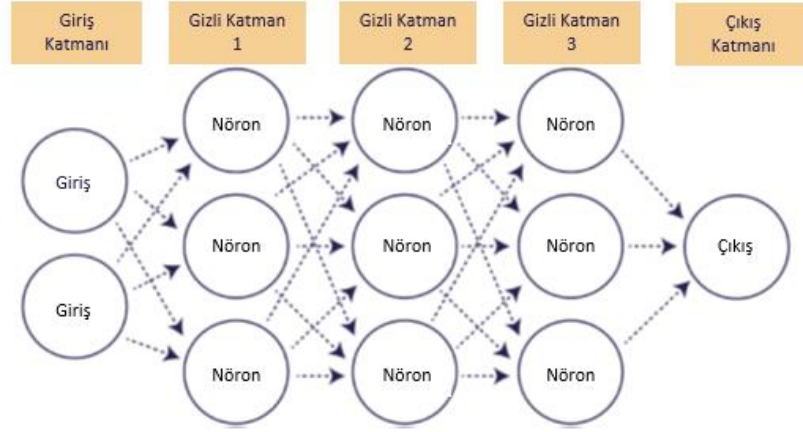
### YAPAY ZEKA, MAKİNE ÖĞRENMESİ VE DERİN ÖĞRENME

Yapay zeka, insan zekasını taklit ederek sorun çözme odaklı davranabilen ve bilgisayar biliminin geniş bir alanını temsil eden bir terimdir. Makine öğrenimi yapay zekanın bir alt dalıdır. Makine öğrenimini ise bir yapay zeka (AI) modeli tasarlamak adına veri öğrenen, algoritma tasarlayan ve eğiten bir teknik olarak tanımlayabiliriz. Makine öğrenimin bir alt dalı olan yapay sinir ağları (YSA) ise insan beynindeki nöronlar ve nöronlar arası iletişimden ilham alır. Yapay sinir ağları, girdi katmanı bir veya birden fazla gizli katman ve bir çıktı katmanından oluşur. Derin öğrenme algoritmaları verileri kendi kendine öğrenmeyi sağlayan özellik çıkarma yeteneğine sahip algoritmalarıdır [1]. Şekil 2.1’de Yapay zeka ve Derin öğrenme arasındaki hiyerarşiyi anlatan bir Ven şeması paylaşılmıştır.



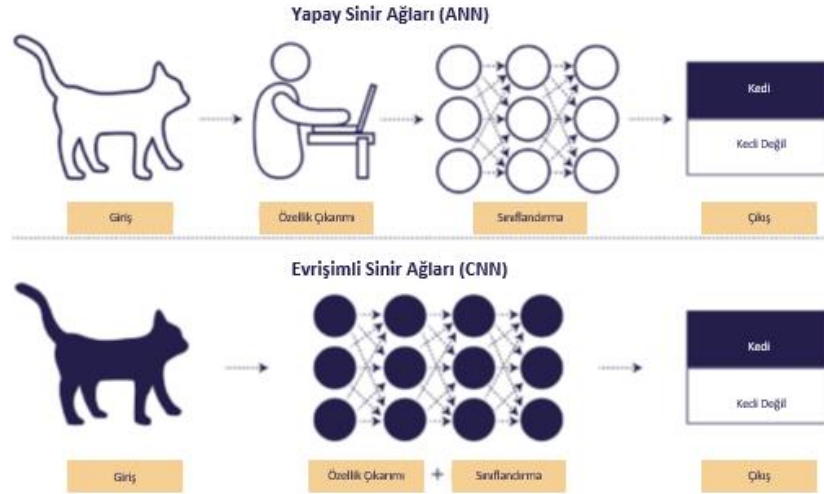
Şekil 2.1. Yapay zeka ve derin öğrenme arasındaki hiyerarşi [1].

Yapay sinir ağlarında veriler, nöronlar aracılığı ile aktarılır. Girdi katmanı, çıktı katmanı ve üç gizli katmandan oluşan bir yapay sinir ağı Şekil 2.2’de şematize edilmiştir.



Şekil 2.2. Yapay sinir ağı şeması[1].

Yapay sinir ağlarının ilk zamanlarında özellik çıkarımı araştırmacılar tarafından yapılmaktaydı. Derin öğrenme algoritmalarının verilerden otomatik olarak özellik çıkarımı yapması, derin öğrenme algoritmalarının en önemli avantajlarından sayılmaktadır [1].



Şekil 2.3. Makine öğrenmesi ve derin öğrenme özellik çıkarımı [1].

## 2.1. MAKİNE ÖĞRENİMİ TÜRLERİ

Makine öğrenmesi ve derin öğrenmenin de tabanlarının dayandığı sinir ağları, beynin davranışlarını taklit eden bir sistem olarak tanımlanmıştır [7]. Fakat farklı bir çalışmada bu taklit işleminin insan beynindeki nöron bağlantılarını tam olarak anlamlandırmadan gerçekleştirildiği de bu bilgiye eklenmiştir [8]. Makine öğrenmesi (ML) algoritmaları bir girdi veri kümesini istenen bir çıktı veri kümesine eşlemenin yollarını üretir. Bu eşleme işlemini, algoritma daha fazla veri örneğine maruz bırakılarak yaklaşımdaki belirsizliğin azalması beklenir. Öğrenme işleminin gerçekleştirilebildiği dört yaklaşım mevcuttur [9]. Yaklaşımlar şu şekilde sıralanabilir: denetimli, denetimsiz, yarı denetimli öğrenme ve takviyeli öğrenme.

### 2.1.1. Denetimli Öğrenme

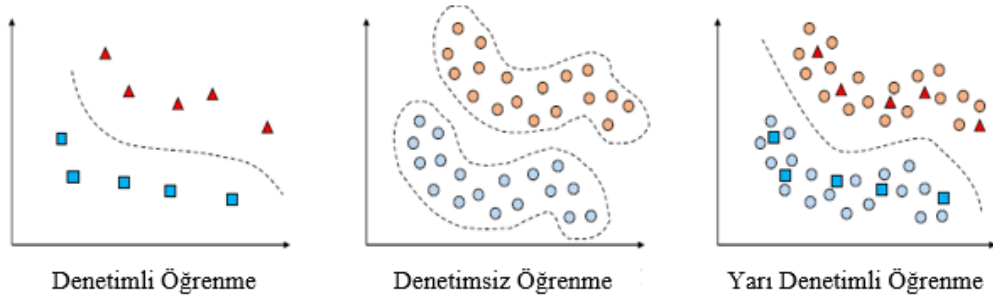
Verimli şekilde öğrenme gerçekleştirebilmek adına verilerin etiketlenmesini kapsar. Bu durumu insanoğlunun bir rehber eşliğinde çevresini anlamlandırabilmesine benzetebiliriz. Örneğin  $x$  verileri için istenen çıktının  $y$  olduğu bir durumda denetimli algoritma  $f$  işlevi kullanarak çıktıya yaklaşımaya çalışır. Kayıp fonksiyonu olarak da adlandırılan bu yaklaşım  $x$  ve  $y$  arasındaki farkın hesaplanması ile sağlanır. Farkı hesaplamak için doğruluk veya hata ölçütleri kullanılabilir [9]. Denetimli öğrenme teknikleri genellikle sınıflandırma ve regresyon problemleri için kullanılır. Sınıflandırma problemleri bir veya birden fazla etikete sahip verilerden oluşabilir. Ağın çıkış katmanı toplam sınıf sayısı ile aynı nöron sayısına sahip olmalıdır. Amaç fonksiyonunda çıkış vektörüyle istenen çıkış arasındaki olasılık dağılımı hesaplanır. Tek sınıflı problemler için bu olasılığı hesaplamak oldukça hızlıdır fakat bir örneğin birden fazla sınıfa ait olduğu durumlarda çok etiketli durumlar olarak adlandırılmaktadır ve bu durum işleri karmaşıklaştırabilir. Kayıp fonksiyonları ikili çapraz entropi veya kategorik çapraz entropi olarak seçilebilir. Regresyon problemleri için ise genellikle ortalama kare hata kaybı kullanılır [10].

### 2.1.2. Denetimsiz Öğrenme

Denetimsiz öğrenme veriler üzerinde etiketleme olmadan ve insan denetiminin minimumda olduğu öğrenme türüdür. Çocukların gördükleri yeni nesnelere iletişim çabasında nesnenin anlamlandırılabilmesi için geçmişte karşılaştıkları nesnelere ile bağ kurma çabasına benzetilebilecek olan bu öğrenme türü, veri toplamanın veri etiketlemeden daha kolay olduğu durumlarda kullanılabilir. Denetimli öğrenmede olasılık tahmini yapılmaya çalışılırken denetimsiz öğrenme örüntü tanıma amacı ile hareket eder. Örüntü tanıma ile hareket eden bu algoritmalar genellikle kümeleme problemlerini çözmede kullanılabilir [9].

### 2.1.3. Yarı Denetimli Öğrenme

Yarı denetimli öğrenme etiketlenmiş ve etiketlenmemiş verilerin birlikte kullanılmasıyla oluşur. İnsanlar denetimli ve denetimsiz öğrenmenin ikisine de maruz kaldıklarından öğrendikleri denetimli verilerden çıkarım yeteneğine sahip olurlar [9]. Şekil 2.4'te etiketlenmiş veriler kare ve üçgen şeklinde temsil edilmiş, etiketlenmemiş veriler ise daire şeklinde temsil edilmiştir.

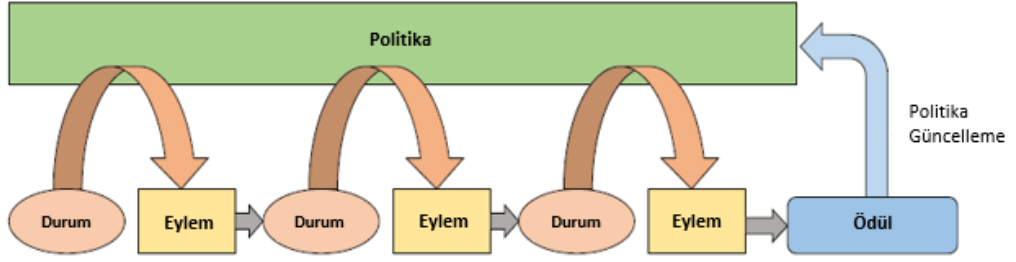


Şekil 2.4. Denetimli, denetimsiz ve yarı denetimli öğrenme [9].

### 2.1.4. Takviyeli Öğrenme

Takviyeli öğrenme yönteminin mantığı, insanoğlunun aldığı kararlar sonucunda başarı ya da başarısızlıkla sonuçlanan eylemlerinden çıkardığı bilgilere dayanır. Kararlar bir eylem dizisini oluşturur bu eylemlerin sonucunda elde edilen deneyimler gelecekte

daha iyi kararlar alınmasını sağlar. Takviyeli öğrenme algoritmalarında eylem seçme yöntemlerine politikalar adı verilir. Şekil 2.5'te sonucunda ödülün elde edildiği, politikalar tarafından belirlenen bir dizi eylemin art arda gelmesiyle bu öğrenme temsil edilmiştir [9].



Şekil 2.5. Takviyeli öğrenme temsili [9].

## 2.2. MODEL OLUŞTURMA AŞAMALARI

Makine öğrenmesinde model oluşturulurken geçilen aşamalı yollar vardır. Ağın iyileştirilmesini sağlamak adına amaç fonksiyonu tanımlanması, ağ mimarisi oluşturma ve öğrenme stratejisi kullanmak gerekir [10]. Makine öğrenimi türlerinde ağın öğrenme stratejileri açıklanmıştır. Model oluşumunda ağ mimarisi oluşturmadan önce verilerin toplanması, bu verilerin ön işlemeden geçmesi, kullanılacak olan mimaride amaç fonksiyonu doğrultusunda eğitimin tamamlanabilmesi açısından hiper parametre ayarlarının yapılması gerekir. Ağın performansının değerlendirilebilmesi açısından, bu işlemler tamamlandıktan sonra ağın doğrulanması gereklidir. Veri toplanması aşaması bölüm 6'da detaylı incelenecektir.

### 2.2.1. Veri Ön İşleme

Mimari hazırlanmadan ve veri algoritmaya beslenmeden önce verinin karakteristik özelliklerini kaybetmeyecek şekilde bir ön işlemeden geçirilmesi gerekir. En yaygın kullanılan ön işleme tekniklerinin bazılarını bu bölümde inceleyeceğiz.

### **2.2.1.1. Girdiyi Belirli Bir Çözünürlüğe Göre Yeniden Boyutlandırma**

Çoğu makine öğrenmesi (ML) algoritması mimarisi gereği optimize çalıştığı bir çözünürlük boyutuna sahiptir. Ağın gereksinimlerine göre veri beslenmeden önce veri çalışılacak ağın ölçeklendirme sınırlamasına göre ayarlanmalıdır. Bu teknik en yaygın kullanılan ön işleme tekniklerinden biridir [10].

### **2.2.1.2. Beyazlatma Dönüşümü**

Veri dağılımına göre bir görüntüyü normalleştirme yöntemi olan beyazlatma dönüşümü, ağın çok büyük veya çok küçük gradyanlardan kaçınılması amacı ile kullanılır.

### **2.2.1.3. Veri Artırma**

Performans iyileştirme bakımından yapılan ön işlemlerden bir diğeri veri artırma işlemidir. Veri tipine bağlı kalarak, verinin karakteristik özellikleri ve kalitesi korunmaya çalışılarak çeşitli teknikler ile veri artırımı yapılabilir. En yaygın kullanılan veri artırma yöntemleri afin veya afin olmayan döndürme, öteleme, yakınlaştırma, çevirme, perspektif bozma gibi dönüşümlerdir. Gürültü eklemek veya azaltmak gibi veri artırma yöntemleri de kullanılabilir [10]. Veri artırma yöntemleri bölüm 4'te detaylandırılmıştır.

## **2.2.2. Mimari Seçimi**

Makul derecede veri elde edildikten sonra mimari tasarlamak ya da mimari seçmek için akılda olması gereken üç temel unsur vardır bunlar giriş tipi, amaç fonksiyonu, veri karmaşıklığıdır. Giriş tipi verinin yapısını tanımlar. Veri tek boyutlu bir tensör olabilirken, gri tonlamalı bir görüntü verisi iki boyutlu bir tensör üretir ve hatta görüntü verisinin RGB formatında olması durumunda üç boyutlu tensörler giriş tipi olabilir. Amaç fonksiyonları çıkış tensörünün şeklini belirlemektedir. Mimari seçiminde çok önemli rol oynayan bu etkenler sınıflandırma, regresyon veya segmentasyon gibi problemlerin çözümünde rol oynar. Veri karmaşıklığı ise diğeri bir faktördür. RGB

formatında olan verileri sınıflandırmak gri tonlamalı verileri sınıflandırmaktan daha karmaşıktır ve bu durum verilerin karmaşıklığından dolayı ağında daha derin olmasını gerektirir [10].

### 2.2.3. Hiper Parametre Ayarları ve Metrikler

Algoritma mimarisinde performans büyük ölçüde hiper parametrelere bağlıdır. Mimariler kabul edilebilen sonuçları üretmeden önce fazlaca eğitim verisi ve zamana ihtiyaç duyar fakat karar sürecinde çok az veya hiç içgörü sağlamayan ‘kara kutu’ gibidirler [11]. Özellikle hiper parametre ayarları konusunda deneysel davranılmak zorundadır. Hiper parametrelerle ilgili kararlar doğrulama seti üzerindeki performans gözlemlenerek verilir. Mimari, eğitim veri seti üzerinde farklı parametreler kullanılarak tekrar ve tekrar eğitilir. En iyi sonuç veren hiper parametre kombinasyonları test veri seti üzerinde test edilerek performans metrikleri ile mimari değerlendirilir. Özetlenecek olursa eğitim veri seti mimarinin geri yayılım ile eğitilmesi için, doğrulama veri seti mimarinin hiper parametre ayarlarının sağlanması için ve son olarak mimarinin eğitimde görmediği test veri seti ise modelin performans metrikleri ile değerlendirilmesi için kullanılır. Mimaride ana performans metriği doğruluktur fakat bu metrik genel performans için geçerlidir. Doğruluk metriği tek başına ağın her sınıfı için yeterli bir ölçüm değildir. Sınıflar için daha hassas metrikler mevcuttur bunlar kesinlik, geri çağırma, f-skoru, hassasiyet metrikleri olarak özetlenebilir [10]. Bir sınıflandırma örneği verecek olursak tıbbi bir alanda hastalık teşhisi için hem yanlış pozitif hem de yanlış negatif soruna neden olur. Yanlış pozitif hastalığın olmadığı bir durumda hastanın gereksiz şekilde tedavi almasına, yanlış negatif ise hasta olan bir bireyin sağlıklı olarak sınıflandırıldığından tedavi alamamasına neden olur. Bu sebeple sınıflar arası performans yine probleme bağlıdır ve performans metrikleri soruna özgü şekilde önem kazanmaktadır [8]. Buna ek olarak bu performans metrikleri daha hassas metriklerdir ve ağın her sınıfındaki veriler için bire karşı hepsi mantığına dayanır [10].

- Gerçek Pozitif (TP): Bir örneğin doğru bir şekilde bir sınıfa ait sınıflandırılmasıdır.
- Yanlış Pozitif (FP): Bir örneğin yanlış bir şekilde bir sınıfa ait sınıflandırılmasıdır.



- Gerçek Negatif (TN): Bir örneğin doğru bir şekilde bir sınıfa ait olmadığına sınıflandırılmasıdır.
- Yanlış Negatif (FN): Bir örneğin yanlış bir şekilde bir sınıfa ait olmadığına sınıflandırılmasıdır.

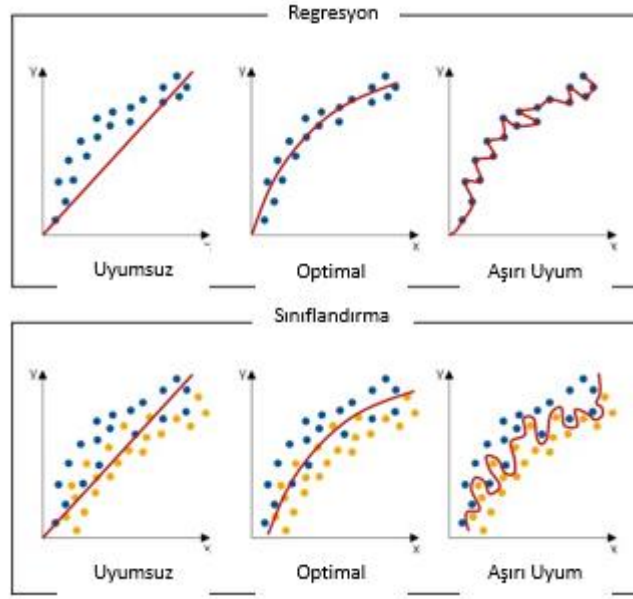
$$\text{Kesinlik (P)} = \frac{TP}{TP + FP}, \quad (2.1) \text{ Kesinlik}$$

$$\text{Duyarlılık (R)} = \frac{TP}{TP + FN}, \quad (2.2) \text{ Duyarlılık}$$

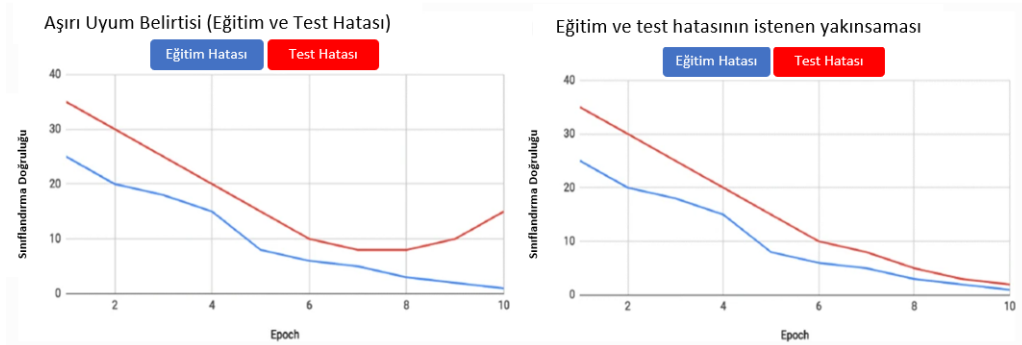
$$\text{F-Skoru} = \frac{2PR}{P + R}. \quad (2.3) \text{ F-skoru}$$

### 2.3. MODEL GENELLENEBİLİRLİĞİ NASIL SAĞLANIR?

Genelleştirilebilir model eğitim verilerinde sağladığı performansı test verilerinde de sağlayabilen modeldir [8]. Model eğitim verileri için yeterince uyarlanmadıysa performansı hem eğitim hem de test verileri için düşük performans gösterecektir. Aksine eğitim verileri için fazla uyarlanmışsa aşırı uyum (overfitting) problemi yaşayacaktır. Aşırı uyum ya da aşırı uydurma modelin eğitim verilerinde yüksek performans göstermesine fakat test setinde aksine düşük performans göstermesine sebep olur. Şekil 2.9’da regresyon ve sınıflandırma problemleri için optimal durum, yetersiz durum ve aşırı uyum için şematik bir gösterim vardır [12]. Aşırı uyumun eğitimde fark edilebilmesi adına her bir dönem (epoch) için eğitim ve doğrulama doğruluğu çizmek etkili bir yöntemdir. Bu yöntemde her dönem (epoch) için doğruluk veya hatada çizilen eğrilerin birbirine paralel olması beklenir fakat şekil 2.10’da eğitim ve test hatalarının aynı anda azalması beklenirken eğitim hatası azalmaya devam etmiş test hatası ise yaklaşık 8. dönem (epoch) gerçekleştikten sonra artmaya başlamıştır bu durum bize aşırı uyum (overfitting) problemi yaşandığının sinyalini verir [13]. Modelin genellelenebilirliğinin sağlanması açısından aşırı uyum (overfitting) probleminin oluşmasını engellemek önemlidir bunun için bazı çok bilinen yöntemler vardır.



Şekil 2.6. Regresyon ve sınıflandırma problemleri için aşırı uyum, uyumsuzluk ve optimal durum şeması [12].



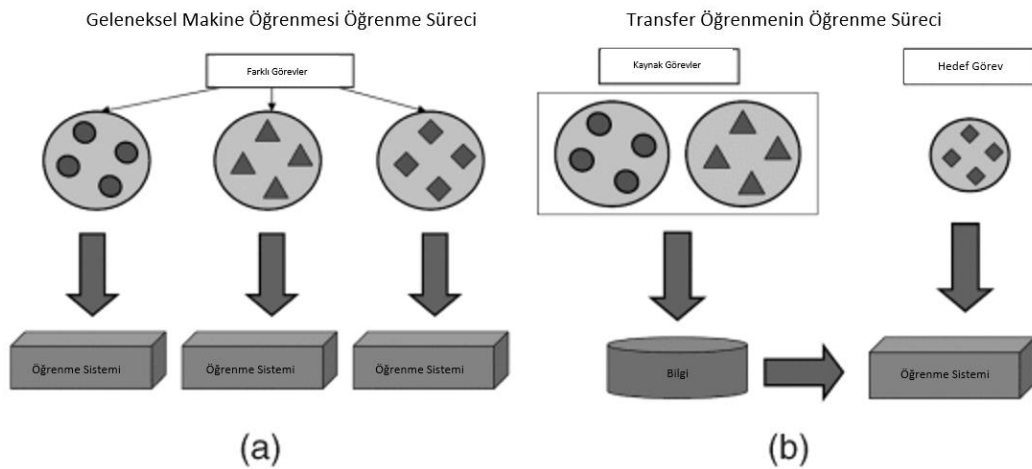
Şekil 2.7. Aşırı uyum (overfitting) problemi [13].

### 2.3.1. Aşırı Uyum (Overfitting) Engelleme Yöntemleri

#### 2.3.1.1. Transfer Öğrenimi (Transfer Learning)

Eğitim ve test verilerinin aynı alandan kullanılması girdi özellik uzayının ve veri dağılım özelliklerinin aynı olmasını sağlar. Fakat bu durum her zaman mümkün olmayabilir. Eğitim sırasında modelin veri sınıflarının ayırt edici özelliklerini öğrenmesi beklenir fakat bunun yerine konu dışı özellikleri de öğrenebilir [8]. Aşırı

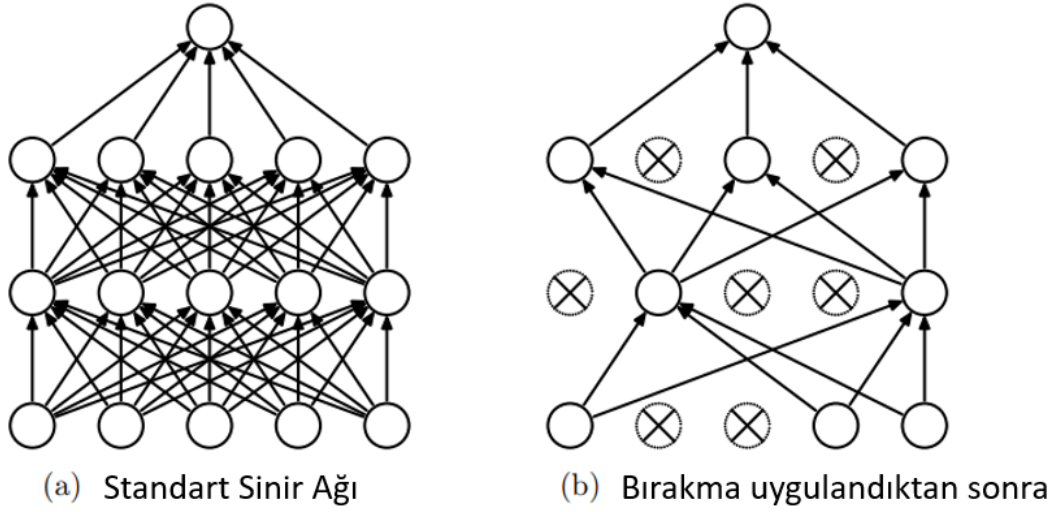
uydurma yüksek varyanslı modellerde yapısal karmaşıklığın bir sonucu olarak eğitim verilerindeki ayrıntılara ve gürültüye odaklanıldığında ortaya çıkar [7]. Bu durum genellikle eğitim verilerinin sınırlı olmasından kaynaklanır. Eğitim verilerine ulaşımın zor olduğu durumlarda farklı alanlardan daha basit şekilde elde edilebilen verilerle eğitilmiş yüksek performanslı mimarilerin oluşturulmasına ihtiyaç duyulabilir işte bu metodolojiyi transfer öğrenme olarak adlandırırız. Transfer öğrenimi bir alanda bilgi sahibi olan bir kişinin ilgili alanda bilgi aktarımı gerçekleştirmesi gibidir. Makine öğrenimi alanında incelenecek olursa bol miktarda etiketli verinin bulunduğu bir ürünün incelemelerinden metin üzerinden yorum tahmini yapılacak olsun. Eğitim ve hedef verilerinin her ikisi de bu alana ait olduğu durumda iyi tahmin verileri elde edilecektir. Fakat eğitim verilerinin ve hedef verinin tamamen farklı alanlardan alındığı durumda elde edilen tahmin değerleri önemli derecede düşecektir. Bir ürünün incelemesinde alanlar tamamen aynı olmasa da bir dizi ortak özelliğe sahip bulunduğu göz önüne alınırsa ki bu; ürünün incelemesindeki yapılan yorumun aynı dilde olması, yorumun metin biçiminde olması ya da her iki metninde ürün incelemesi içermesi gibi örneklendirilebilir, tahmin sonuçlarını iyileştirmek için kullanılabileceği anlamına gelir. Büyük veri setlerinin yaygınlaşmasıyla birlikte hedef etki alanları ilgili olan ancak tamamen aynı olmayan mevcut veri kümelerini kullanmak transfer öğreniminin de yaygınlaşmasına olanak vermiştir [14]. Geleneksel makine öğrenmesinde farklı hedef alanları kendi içlerinde öğrenme gerçekleştirilerek oluşturulurdu fakat transfer öğrenimde farklı hedef alanları birbiri ile ilgili olduğu sürece tamamen aynı olmasa bile kullanılabilir. Şekil 2.11’de bu durum temsil edilmiştir [15].



Şekil 2.8. Geleneksel makine öğrenimi ve transfer öğrenimi uygulanması. a) Geleneksel makine öğrenimi b) Transfer Learning [15].

### 2.3.1.2. Bırakma (Dropout)

Derin sinir ağı çok sayıda doğrusal olmayan katman içerir. Bu durum ağı girdi ve çıktılar arasındaki fazla karmaşık olan özellikleri öğrenmesine olanak sağlar. Fakat ağı çok sayıda parametreye sahip olması ağı kullanımını yavaşlatmaktadır aynı zamanda gereğinden fazla özellik çıkarımı aşırı uydurmaya sebep olacağından bırakma yöntemi kullanılabilir bir tekniktir. Bırakma (dropout) olarak adlandırılan bu tekniğin asıl çalışma mantığı eğitim sırasında bağlantılarıyla birlikte birimleri rastgele olarak sinir ağından düşürmeye dayanır. Bir birimin ağıdan düşürülmesi olarak kastedilen şey birime gelen ve giden bütün bağlantıların ağıdan geçici olarak çıkarılmasıdır. Şekil 2.12’de standart bir sinir ağı ve bırakma (dropout) uygulandıktan sonraki hali şematize edilmiştir. Bu birimler rastgele seçilirler. Rastgele seçimde birimler diğer birimlerden bağımsız sabit bir  $p$  olasılığı ile bulunmaktadır.  $p$  olasılığı deneysel olarak seçilebilir ya da optimal değer olarak görülen 0,5 kullanılabilir. Bu şekilde Bırakma (dropout) ile inceltirilmiş ağı elde edilmiş olur ve parametrelerin azalmasına yardımcı olur. Diğer düzenleme yöntemlerine göre, özellikle sınıflandırma problemlerinde daha etkili sonuçlara sahip olduğu kanıtlanmıştır. Eğitimde ileri ve geri yayılım inceltirilmiş ağı üzerinden yapılır. Büyük öğrenme oranları ile kullanılması tavsiye edilen bu yöntem tek başına iyi sonuçlar elde etse bile öğrenme oranının yüksek tutulmasıyla daha geniş keşif yapacaktır. Öğrenme oranının azaltılmasıyla birlikte daha kısa adımlar atar böylece daha az keşif yapar [16].



Şekil 2.9. Bırakma (dropout) temsili [16].

### 2.3.1.3. Ön Eğitim (Pre-training)

Ön eğitim transfer öğrenimi ile çok karıştırılan bir kavramdır. Ön eğitimde mimari oluşturulur daha sonra büyük bir veri seti ile eğitilir. Ön eğitimde sadece mimarinin ağırlıkları kullanılırken transfer öğreniminde ağırlıkların yanı sıra ağ mimarisi de aktarılmalıdır. Ön eğitim tekniği bu anlamda transfer öğreniminden ayrılmaktadır [13].

### 2.3.1.4. Erken Durdurma (Early Stopping)

Erken durdurma, eğitimin ilerleyişine göre farklı ölçümlerle aşırı uydurmadan şüphelenildiği durumlarda eğitimin durdurulmasıdır. Erken durdurma aşırı uydurmayı ortadan kaldıran bir yöntem değildir aksine oluşmasından şüphelenildiği durumlarda eğitimin durdurulmasını kapsar [7]. Hem aşırı uydurmanın önlenmesi hem de eğitim süresinin azaltılması önemli iki etkisidir. Fakat hesaplamaların maliyetini azaltmak pahasına modelin genellenebilirliği bozulmamalıdır. Bu durumda eğitimin ne zaman durdurulması gerektiğine karar verebilmek önem arz eder. Eğitim genelliğini kaybetmese, aşırı uyum belirtileri göstermese dahi sonsuza dek süremeyeceğinden ve çok erken bir durdurma olduğunda eğitimin test ya da doğrulama veri setinde başarısı düşük olacağından ne zaman durdurulacağı kararı önemlidir. Bu yöntemin yaygın kullanımı, yöntemin anlaşılmasının kolaylığı ve uygulanmasının pratikliğindedir.

Gelişmiş bir yöntem olarak eğitimin sürekli kaydedilmesi ve kaydedilen eğitimden sonra çapraz doğrulama yapılarak en iyi sürümün saklanması amaçlar fakat bu şekilde erken durdurmanın iki dezavantajı ortaya çıkar bunlar eğitimin çok uzun sürebileceği ve sürekli kaydedilmesi gerektiğidir. Bu yöneme göre daha geleneksel farklı bir yaklaşımda mevcuttur ki bu yöntemde ortalama karesel hatanın (MSE), eğitim ve doğrulama setleri için manuel kontrolü sağlanarak eğitimin durdurulmasıdır. Bir diğer yöntem ise ortalama karesel hatanın eğitim setinde gelişmediğinde durdurulmasıdır. Bu gibi manuel yöntemlerle durdurmada yöntemin karşılaştırılmaz olduğu bilinmektedir. Erken durdurmada ilk kural maksimum dönem (epoch) sayısını belirlemektir. Çoğu erken durdurma 1000 dönemden önce gerçekleşmektedir. Dönem sayısını yüksek tutmak bu durumdan dolayı önemlidir. İkincil kurallarda ise manuel yöntemlerden esinlenerek yaklaşım yapılmıştır. Doğrulama hatasının belirli bir eşik üzerine çıkması veya eğitim eğrisi gürültüsünün öğrenme etkilerini aştığında eğitim durdurulması gibi birçok farklı kuralla yöntemler geliştirilmiştir. İlk kural her zaman sabittir ancak ikincil kurallar duruma göre uygulanabilir [17].

## BÖLÜM 3

### FARENJİT

Farenjit, boğaz arkasındaki farenks bölgesinin iltihaplanması olarak tanımlanan bir hastalıktır. Farenks bölgesi bademcik adı verilen bir dizi organın bir araya gelmesiyle oluşur. Bu bölge enfeksiyöz etmenlere ki bunlar bakteri ve virüslerdir, ilk karşılaşma ve savunmayı yapan bölgedir. Bakteriler ya da virüsler ağızdan veya burundan solunum şeklinde vücuda girdiğinde genellikle bademciklerin iltihaplandığı gözlemlenir. Viral ve bakteriyel olmak üzere iki çeşit farenjit vardır. Semptomları arasında çoğunlukla boğaz ağrısı, hafif ateş, yutkunma güçlüğü vardır. Farenjit hastalığının viral ya da bakteriyel olup olmadığı semptomlar kontrol edilerek ve gözle görülerek anlaşılmaz. Viral farenjit, bakteriyel olana göre daha yaygın görünür. Çocukların %60 ile 75'i, yetişkinlerin ise %75 ile 90'ında hastalık teşhis edilmiştir. Bakteriyel farenjit ise çoğu zaman A grubu streptokoklardan oluşur. Bakteriyel farenjit yetişkinlerde, %10 ile %25 arası ve çocuklarda %25 ile %40 aralığında gelişen bir durumdur. Hastalığın bakteriyel mi yoksa viral mi olduğu hızlı strep testi (RST) adı verilen bir yöntemle boğazdan sürüntü alınarak ağız sıvısının kontrolü ile tespit edilebilir. Fakat her iki durumda da bademciklerde şişme ve kızarıklık durumu gözlemlenir. Hastalık daha da şiddetlendiğinde bademcikler kript adı verilen kirli beyaz iltihaplı alan (eksüda) ile kaplı olabilir [18].

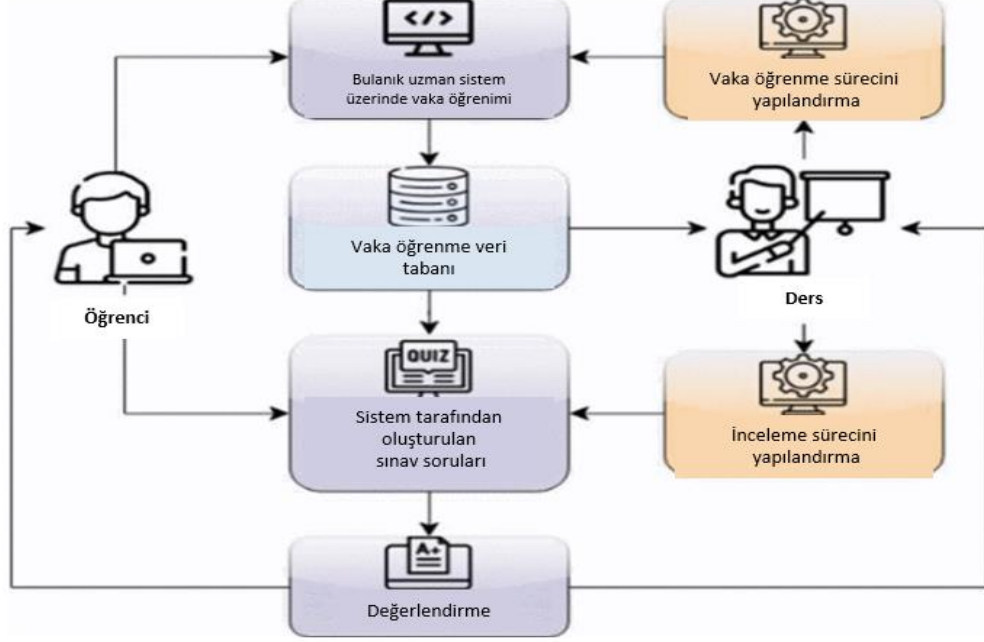
#### 3.1. LİTERATÜR TARAMASI

Wijayanto vd.'nin yaptığı çalışmada VGG-16, VGG-19, DenseNet121, AlexNet ve ResNet152 gibi farklı sinir ağı mimarileri kullanılarak en yüksek 0.98'lik bir doğruluk başarısı yakalanmıştır. Bu çalışmanın başlangıcında, ham veri seti olarak Yoo vd.'nin çalışmasındaki [3], farenjit ve sağlıklı görüntüler kullanılmış, sınırlı veri sebebiyle veriler döndürme, ters çevirme ve yakınlaştırma teknikleri kullanılarak artırılmıştır. Dengeli bir veri seti elde edilmeye çalışılmıştır. Bu çalışmada uygulanan mimarilerde

Adam iyileştirici (optimizer) yöntemi kullanılmış ve mimarinin ara katmanlarında ReLU aktivasyon fonksiyonu, son tam bağlı katmanlarında ise Softmax aktivasyon fonksiyonları ile mimariler başarı ile uygulanmıştır. Veriler ham olarak kullanıldığında %87,67 (VGG-16), %84,93 (VGG-19), %87,67 (DenseNet121), %86,30 (AlexNet) ve %69,86 (ResNet152) başarı oranları elde edilmiş olmasına rağmen veriler artırıldığında %96,51 (VGG-16), %97,67 (VGG-19), %93,02 (DenseNet121), %87,21 (AlexNet) ve son olarak %83,72 86 (ResNet152) mimarilerin doğruluk başarılarının arttığı da gözlemlenmiştir [5].

Istiadi vd.'nin yaptığı çalışmada Bulanık Mamdani yöntemi kullanılarak hastalığın tedavisinde antibiyotik reçete edilip edilmeyeceğine karar verilebilmesini amaçlamışlardır. Bu çalışma vaka çalışmasına dayalıdır. Bir e-öğrenme sistemi üzerinden sorulan sorulara verilen cevaplar üzerinden çalışma yapılarak hastanın viral ve bakteriyel farenjit semptomları toplanmıştır. Toplanan veriler uzmanlar tarafından doğrulama süreci uygulandıktan sonra iki önemli parametre elde edilmiştir. Ateş ve baş ağrısının viral ve bakteriyel farenjitte en çok belirtilen iki semptom olduğu gözlemlenmiştir. Yaş ve diğer değişkenler eklenerek Bulanık Mamdani yöntemi uygulanmıştır ve bu çalışmada görüntü veri seti ile çalışılmamıştır [4].

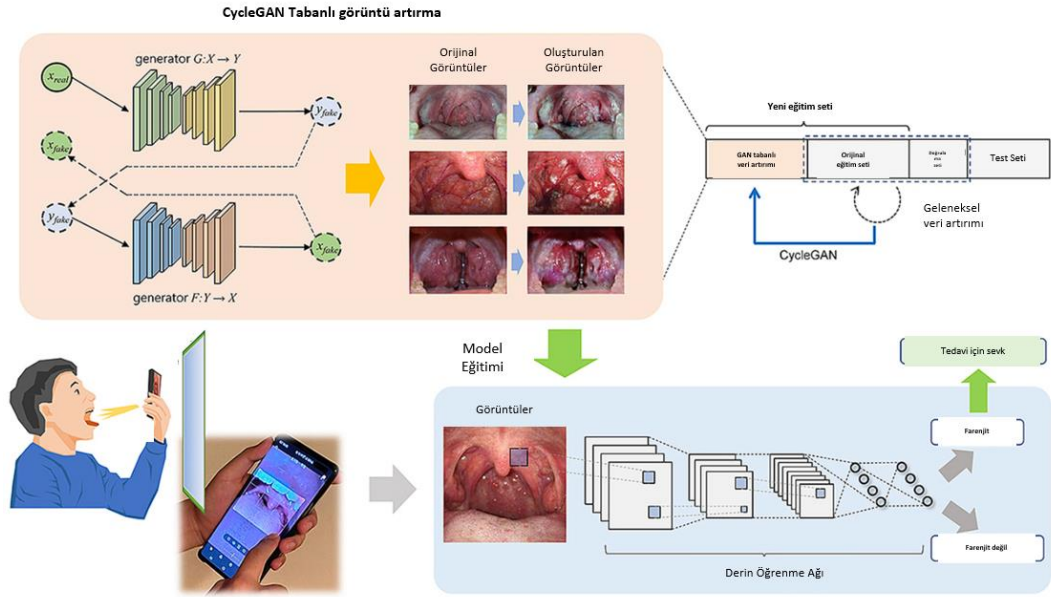




Şekil 3.1. Bulanık Mamdani yöntemi ve e-öğrenme sisteminin entegre edilmesi [4].

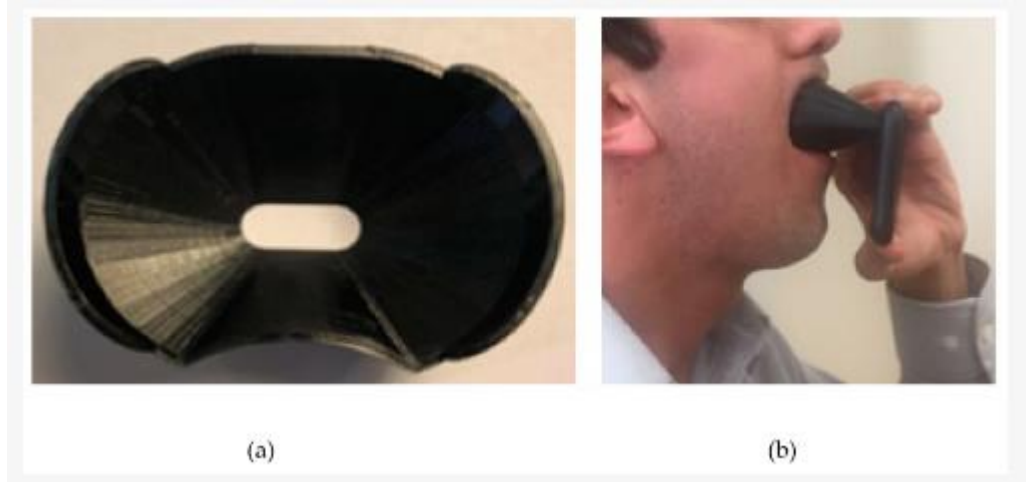
Yoo vd.'nin yaptığı bu çalışmada ise veri toplama aşamasında internet üzerinden kişinin kendi çektiği boğaz görüntüleri kullanılarak çalışma yapılmıştır. Farenjit ve normal boğaz görüntülerine dayanan bu veriler uzman hekimler tarafından sınıflandırılmış ve veri seti farenjitli 131 ve normal 208 boğaz görüntüsü şeklinde elde edilmiştir. Sınırlı sayıda veriden dolayı CycleGAN kullanılarak veri artırımı uygulanmıştır. CycleGAN mimarisi ile artırmadan önce veriler afin dönüşümlerle artırılmıştır. Artırma işlemleri sonunda toplamda 1500 farenjitli boğaz görüntüsü ve 1500 normal boğaz görüntüsü elde edilmiştir. Çalışmada ResNet50, Inception-V3, MobileNet-V2 mimarileri ile afin artırma yöntemlerinin kullanıldığı veriler ve GAN tabanlı artırılmış verilerinde olduğu sentez bir veri seti üzerinde eğitimler yapılmıştır. Elde edilen sonuçlar afin yani temel veri artırımından sonra %94,8 (ResNet50), %94,7 (Inception-V3) ve %94,3 (MobileNet-V2) doğrulukla ölçülmüş ve GAN tabanlı artırımıyla elde edilen sentez veri setinde doğruluk oranları %98,5 (ResNet50), %97,8 (Inception-V3) ve %96,9 (MobileNet-V2) olarak gözlemlenmiştir. Bu oranlar ortalama alınmaksızın ulaşılabilen en yüksek değerlerdir. Eğitimler için Adam iyileştirici (optimizer) kullanılmış Google Colab Pro ortamında eğitimler

gerçekleştirilmiştir. CycleGAN ile elde edilen görüntülerin tamamı dahil edilerek ve tüm görüntülerin 256 x 256 genişlik ve yükseklikte olduğu çözünürlüğe ayarlanarak eğitimler yapılmış, mimari modelleri ön eğitime tabi tutulmuştur [3].



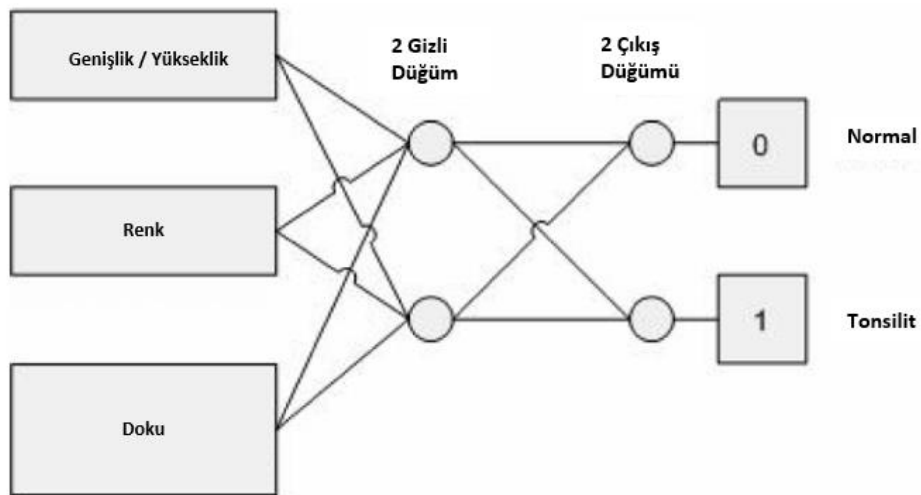
Şekil 3.2. Çalışma Özeti şeması [3].

Askarian vd.'nin yaptığı farklı bir çalışmada ise sağlıklı ve hastalıklı boğazdan görüntü alabilmek açısından bir aygıt tasarlanıp kullanılmıştır. Aygıt akıllı telefon kamerası ile çekilecek fotoğrafların net çıkabilmesi açısından ışığı dengelemek amaçlı tasarlanmıştır. Toplanan veri görüntülerine RGB kanallar üzerinden renk dönüşümü uygulanmış ve Matlab üzerinde çalışılmıştır. K-NN algoritması kullanılarak sınıflandırma yapılmış ve algoritma doğruluğu %93,75 olarak ölçülmüştür [19].



Şekil 3.3. Çalışmada kullanılan aygıt ve kullanımının gösterimi [19].

Jirawanitcharoen vd.'nin yaptığı çalışmada bademcik büyüklüğü, bademcik rengi ve bademcik yüzeyinin iltihaplı alan içerip içermemesi üzerinden üç faktör kullanarak sinir ağı eğitilmiş, toplamda 30 sağlıklı ve iltihaplı alan içeren boğaza sahip hastalıklı boğaz görüntüleri ile üç katmanlı bir sinir ağında sınıflandırma işlemi gerçekleştirilmiştir [20].



Şekil 3.4. Çalışmada kullanılan üç katmanlı sinir ağı temsili [20].

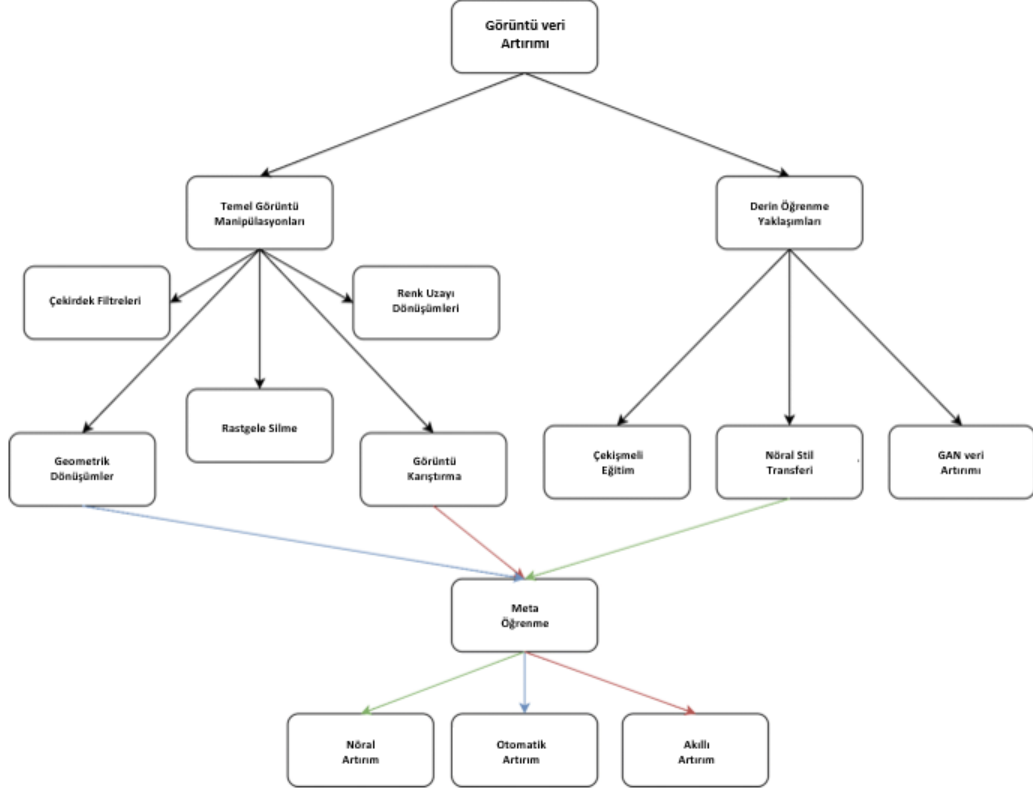
## BÖLÜM 4

### VERİ ARTIRMA

Veri artırma, veri setinde bulunan mevcut verilerin değiştirilmiş kopyalarını oluşturmak ve bu şekilde veri setini zenginleştirip, geliştirmek adına yapılan dönüştürme işlemleridir. Veri kalitesi ve varyans korunduğu sürece, veri kapsamı ne kadar geniş olursa modelin doğru tahmin yapmadaki performansı o kadar yüksek olacaktır aynı zamanda veri artırma makine öğrenmesi eğitimlerinde aşırı uydurmayı önlemek için kullanılan önemli bir yöntemdir. Aşırı uydurma eğitim verilerinin rastlantısal özelliklerini öğrenmek olarak adlandırılabilir. Makine öğrenmesinde genellenebilirlik öncelikli amaçtır. Aşırı uydurma modelin genellenebilirliğini zayıflatır. Modelin eğitimde gördüğü eğitim verileri ve hiç görmediği test verilerinin karşılaştırılmasıyla modelin performansı ve genellenebilirliği ortaya çıkmış olur [12].

Yeni veriler eklemeyen eğitim için kullanılacak olan verilerin miktar ve çeşitliliğini önemli ölçüde genişletmeyi sağlayan veri artırma, girdiye ve amaca bağlı olarak uygulanacak farklı stratejilere göre farklı performans sergileyecektir [21].

Özellikle görüntü veri kümeleri üzerinde uygulanan çok çeşitli veri artırma yöntemleri mevcuttur. Bunlar temel görüntü manipülasyonları geleneksel ya da klasik artırma olarak da adlandırılan ilk yöntem ve derin öğrenme yaklaşımları olarak ikiye ayrılabilir. Şekil 4.1’de görüntü verisi artırmanın bir taksonomisi paylaşılmıştır [13].



Şekil 4.1. Veri artırma taksonomisi [13].

Şekil 4.1’de kullanılan renkli çizgiler meta-öğrenmenin hangi sınıftan geldiğini belirtmektedir. Örneğin Geometrik dönüşümler meta-öğrenmede otomatik artırma kapsamındadır. Şekildeki her bir sınıf bu tezde ayrı ayrı incelenmeyecektir. Shorten vd.’nin yaptığı çalışmada bu incelemeler mevcuttur [13].

## 4.1. GELENEKSEL VERİ ARTIRIMI

### 4.1.1. Geometrik Dönüşümler

Verinin özelliğine göre her artırma işlemi her veride kullanılamaz çünkü bu durum veri güvenliğini etkileyebilir. Kedi ve köpek resimleri içeren bir veri setinde döndürme veya çevirme artırma yöntemlerini kullanmak faydalı olabilir. Fakat rakamlardan

oluşan bir veri setinde döndürme veri artırma yöntemini kullanmak, 6 ve 9 rakamlarında olacağı gibi verinin etiketinin yanlış sınıflandırılmasına yol açabilir. Bu durum veri güvenliğini etkiler [13]. Bu tez çalışmasında farenjit görüntü veri kümesi kullanıldığından insan fizyolojisine uygun düşecek yöntemler kullanılmıştır.

#### **4.1.1.1. Çevirme (Flipping)**

Yatay eksen ve dikey eksen üzerinde yapılan çevirme işlemlerini kapsar. Dikey eksene göre yatay eksen üzerinde döndürme işlemi daha fazla tercih edilmektedir [13].

#### **4.1.1.2. Renk Alanı**

Görüntü verilerinde boyut, yükseklik x genişlik x renk kanalları şeklinde kodlanır. Renk kanalları da yine RGB şeklinde üç kanaldan oluşur. Basit renk değişimleri R, G veya B kanallarından herhangi birinin değişimi ile oluşur. Seçilen kanal yalıtılır ve diğer kanallar 2 sıfır matris eklenerek temsil edilir. Bunun yanı sıra görüntü verisinin parlaklığını artırmak veya azaltmakta önemli bir stratejidir. Parlaklık artırma veya azaltma yöntemi görüntünün RGB değerlerini değiştireceğinden bu yöntem de renk alanı içerisinde incelenir [13].

#### **4.1.1.3. Döndürme (Rotation)**

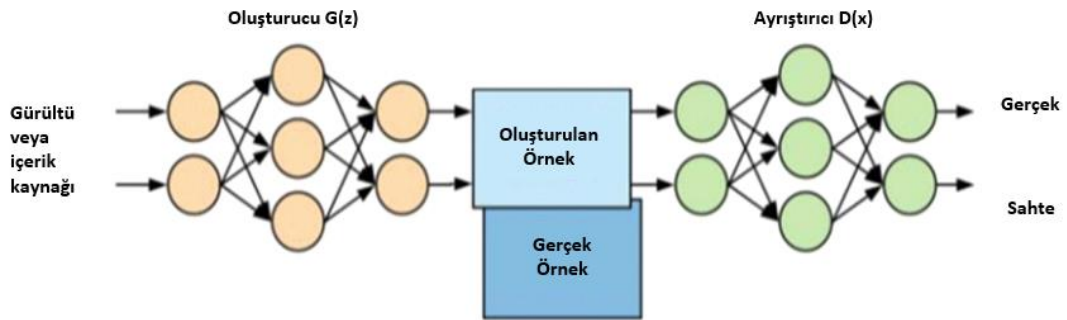
Görüntünün  $1^\circ$  ve  $359^\circ$  arasında kendi ekseninde sağ veya sola döndürülmesi ile kullanılan bir yöntemdir. 1 ile 20 ve -1 ile -20 dereceleri arasında yapılan dönüşümler veri etiketinin etkilenmemesi açısından güvenli kabul edilir. Döndürme derecesi arttıkça etiket korunamayacağından veri güvenliği bozulabilir bu duruma dikkat edilmelidir [13].

#### 4.1.1.4. Yakınlaştırma (Zoom)

Görüntü yakınlaştırılması veya uzaklaştırılmasını kapsar [22].

## 4.2. GAN TABANLI VERİ ARTIRMA

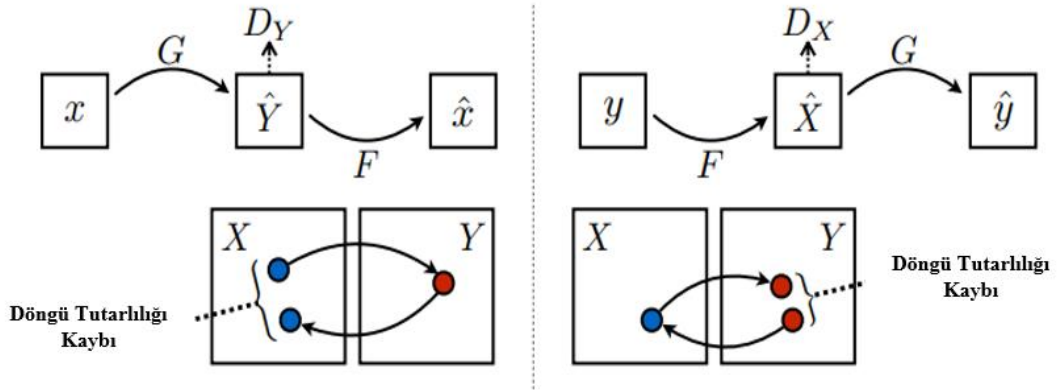
Veri artırma için kullanılan diğer bir yöntem üretken modellemedir. Üretken modelleme mevcut veri setinin ortak özellikleri esas alınarak sentetik veri üretmeye dayanır. GAN'ler var olan tek üretken modelleme tekniği değildir fakat sonuç kalitesi ve hız bakımından ön planda bulunurlar [13]. GAN mimarisi ilk olarak Ian Goodfellow tarafından önerilmiştir [23]. GAN'in çalışma mantığını anlamak açısından en çok verilen anekdot bir polis ve bir kalpazanın arasındaki ilişkidir. Kalpazan olarak örneklendirdiğimiz jeneratör ağı bir çeşit veri girdisi alır. Bu veri vektör, resim, metin gibi farklı formatlarda olabilir. Amaç kalpazanın, polis olarak örneklendirdiğimiz (discriminator) yani ayırıştırıcı ağın üretilen paranın gerçek mi sahte mi olduğunu anlayamayacağı şekilde para üretmeyi öğrenmesidir. Üretici ağ (generator) üretilen örneklerin eğitim setinden mi yoksa üretici ağ tarafından mı oluşturulduğunun anlaşılmayacağı şekilde veri üretmeyi öğrenir [13].



Şekil 4.2' de Üretici ağ sentetik veri üretir ayırıştırıcı ağ ise gördüğü verinin gerçek mi yoksa sahte mi olduğunu değerlendirir.

#### 4.2.1. CycleGAN

CycleGAN bir görüntü sınıfından diğer görüntü sınıfına eşlenmiş veri üretmek amacıyla kullanılır. İleri ve geri tutarlılık kaybı fonksiyonları ile örneğin bir at görüntü verisini, bir zebra görüntü verisine eşlemeyi öğrenir. Üretici ağ (generator) at verisini alıp zebra verisine eşlemeyi öğrenir. Ayrıştırıcı ağ ise üretilmiş zebra verisinin orijinal zebra görüntüsü olup olmadığını anlayamaz. Bu işlemden sonra at verisinden üretilen zebralar yine farklı bir ağdan geçirilerek tekrar at verisine çevrilir. Bu ayrımcı kayıplarının her ikisi de döngü tutarlılığı açısından toplanır. Bu durum şu şekilde de özetlenebilir, bir dilden diğer dile çeviri yapılırken yapılan çevirinin eski haline çevrilmesi durumunda orijinalliğini sürdürmesi gerekir ki çeviri doğru yapılmış olsun [13]. Şekil 4.3'te  $x$  ve  $y$  görüntü kümesi sınıflarını,  $G$  ve  $F$  oluşturuçu ağları ve  $D$  ayrıştırıcıyı temsil etmektedir.



Şekil 4.3. Döngü tutarlılık kaybı şeması [25].





Şekil 4.4. CycleGAN ile üretilmiş at ve zebra örneği [25].

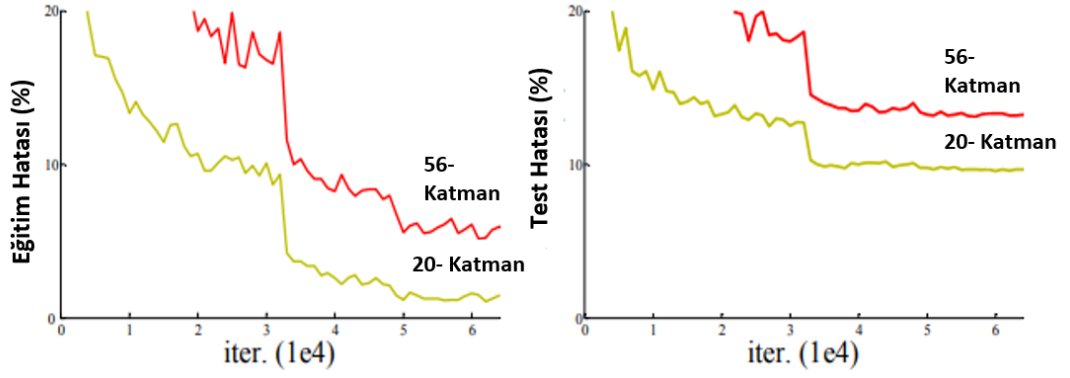
Şekil 4.4'te girdi verisi RGB kanallı bir at görüntüsüdür. Bu veri CycleGAN algoritması ile zebra verisine eşlenmiştir.

## BÖLÜM 5

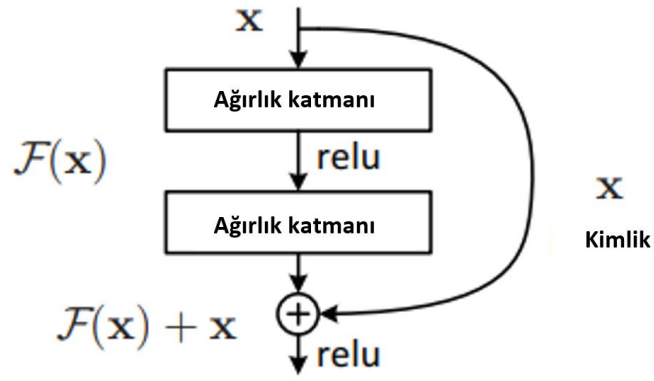
### DNN ALGORİTMALARI

#### 5.1. RESNET

ILSVRC yarışmasını 2015 yılında kazanan evrişimli sinir ağı (CNN) modeli olan ResNet (Artık Ağlar) [26], ResNet-18, ResNet-34, ResNet-50, ResNet-101, ResNet-152 gibi katman sayısına göre adlandırılmış birden fazla mimari modeline sahiptir. Derin öğrenmede fazla katmanlı olan derin sinir ağları verinin karmaşık özelliklerini çıkarmada büyük başarı sağlayabilir. Sığ sinir ağlarına kıyasla daha iyi sonuçlar göstermesi muhtemeldir [27]. Fakat ağın derinliği fazlalaştırıldığında eğitim doğruluğunun belli bir noktadan sonra doyuma ulaşp hızla düşeceği gözlemlenir. Fazla katman kullanılarak eğitim ve test hatalarındaki değişimin gözlemlenebileceği şekil 5.1’de 56 katmana sahip olan ağın eğitim ve test hatasının, 20 katmana sahip olan sığ ağdan daha yüksek olduğu gözlemlenmektedir. Bu olay bozulma sorunu olarak adlandırılır buna çözüm olarak ise artık blok katmanları önerilmiştir. Şekil 5.2’de tipik bir artık blok örneği bulunmaktadır [26]. Artık blok, iki evrişimli katman arasında parametre kullanılmadan önceki katmanın çıktısının bir sonraki blok girdisi olarak kullanılmasını sağlayan bir kısayol olarak düşünülebilir [28].

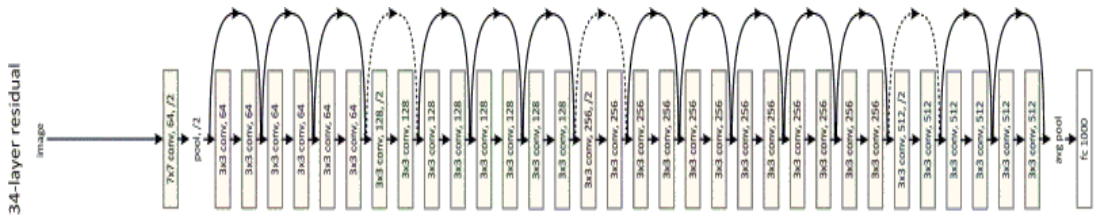


Şekil 5.1. CIFAR-10 ile eğitilmiş 20 ve 56 katmanlı düz ağların eğitim ve test hataları [26].



Şekil 5.2. Artık blok örneği [26].

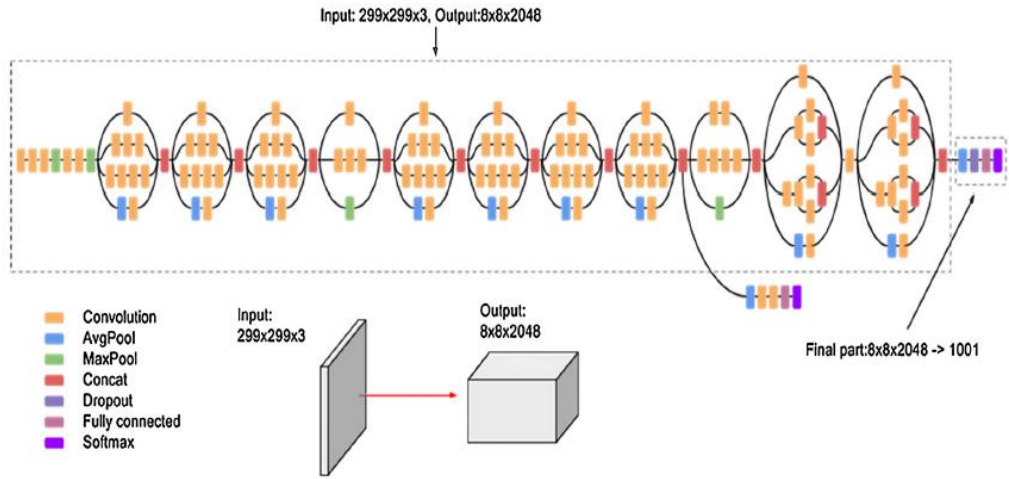
ResNet50 mimarisi, ResNet34 mimarisindeki her artık blok arasında kalan iki evrişimli katmanın üç evrişimli katman olacak şekilde yeniden şekillendirilmiş halidir. Şekil 5.3'te ResNet34 mimarisinin temsili bulunmaktadır [26].



Şekil 5.3. ResNet34 mimarisi [26].

## 5.2. INCEPTION

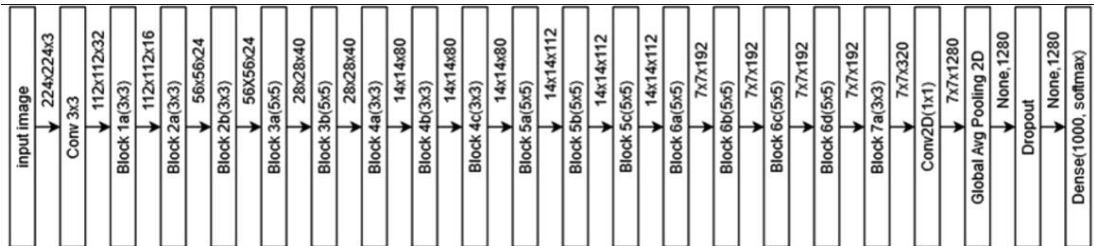
Inception mimarisi, CNN tabanlı evrimsel bir mimaridir. 2015 yılında kendi alanında görüntü sınıflandırması için, alınan ilk 5 hata oranında %3,46'lık hata ile en doğru modeller arasında bulunmaktaydı [29]. Bu hata, model ImageNet [30] veri seti ile eğitildikten sonra elde edilmiştir. Inception mimarisi ilk olarak Google beyin ekibi tarafından geliştirilmiştir. 2014 yılında tanıtılan GoogleNet [31]'in bir parçası olan Inception-V3'ün görüntü tanıma performansı, 2012 yılında önerilen AlexNet [32] mimarisine göre daha yüksektir. Inception mimarisi 22 katmandan oluşur. Maksimum havuzlama katmanlarında, öznetelik çıkarımı için kullandığı 1x1, 3x3, 5x5 gibi farklı filtre boyutlarına sahiptir [33]. Inception mimarisinin temel mantığı aynı seviyede birden fazla çekirdeğe sahip olmasıdır [34]. Google, Inception mimarisindeki parametreleri azaltmak amacı ile önce Inception-V2 [35], daha sonra Inception-V3 [35] mimarilerini piyasaya sürmüştür. Inception-V3 mimarisinde 48 katman bulunmaktadır. Model: evrişim bloğu, Inception modülleri ve sınıflandırıcıdan oluşur [33].



Şekil 5.4. Google Inception-V3 mimarisi [36].

### 5.3. EFFICIENTNET

EfficientNet mimarileri bu çalışmada kullanılan tüm mimariler gibi evrişimli sinir ağı (CNN) modelleri olarak adlandırılabilir. EfficientNet sekiz farklı modelden oluşan bir gruptur ilk model B0 ve son model B7 şeklinde adlandırılmıştır. Ağ içinde B0'dan B7'ye tasarımda bir bileşik katsayı kullanılarak ağ genişliği (katman başına özellik harita sayısı), ağ derinliği (katman sayısı) ve çözünürlüğü eşit şekilde ölçeklendirilerek oluşturulmuştur. Evrişimli sinir ağı modellerinden farklı olarak aktivasyon fonksiyonu olarak katmanlar arası ReLU aktivasyon fonksiyonu yerine Swish aktivasyon fonksiyonu kullanılır [37]. Ölçeklendirme B0 mimarisine uygulandıktan sonra aynı işlem tekrarlanarak sırayla B1-B7 mimarileri elde edilmiştir [38]. Ölçeklendirme yönteminin mantığı girdi görüntüsü ne kadar büyükse ağın genişlik ve derinliğinin de o denli büyük olmasına dayanmaktadır [39]. B0 mimarisi 5.3 milyon parametreye sahiptir ve 224 x 224 giriş çözünürlüğü sınırlandırmasına sahiptir. B7 olan son mimari ise 66 milyon parametreye sahiptir ve 600 x 600 çözünürlüklü görüntülere kadar çıkabilmektedir [38]. EfficientNet mimarileri genel olarak MBConv olarak adlandırılan ters çevrilmiş boğaz bloklar kullanır. Ters çevrilmiş boğaz (MBConv) blokları ilk olarak MobileNetV2 mimarisinde tanıtılmıştır. Bu şekilde mimari küçültülmüş fakat derinlik, genişlik ve çözünürlük ölçeklendirildiğinden daha verimli bir model oluşumu sağlanmıştır [40]. Şekil 5.5'te EfficientNet B0 mimarisinin şematik bir gösterimi mevcuttur.



Şekil 5.5. EfficientNet B0 temsili [38].

## BÖLÜM 6

### METODOLOJİ

#### 6.1. VERİ SETİ

Bu çalışmada kullanılan ham veri seti Yoo vd.'nin çalışmasında [3] kullandığı, 131 farenjit ve 208 normal olmak üzere iki sınıftan oluşan boğaz görüntüleridir. Görüntüler RGB formatındadır. Yoo vd.'nin yaptığı çalışmada veri toplama süreci olarak internet tabanlı veri toplama yöntemi tercih edilmiştir. İnternet üzerinden herkesin erişebileceği Naver Korea (<https://kin.naver.com>) sitesi ve Yahoo Japan (<https://chiebukuro.yahoo.co.jp>) sitesi dahil olmak üzere farklı sosyal soru-cevap siteleri üzerinden kişinin kendi çektiği boğaz görüntüleri toplanmıştır. Toplanan veri kümesine ek olacak şekilde Google üzerinden anonim boğaz görüntüleri, “tonsilit”, “tonsillofarenjit”, “eksüdatif” gibi anahtar kelimeler yardımıyla bulunmuştur. Bu tezde kullanılan ham veri seti öncelikle geleneksel veri artırma yöntemleri kullanılarak artırılmıştır. Bu aşamada geleneksel veri artırma yöntemi olarak 5 farklı yöntem kullanılmıştır. Bunlar afin artırma yöntemleri olarak da geçen: döndürme (+5°), döndürme (-5°), parlaklık verme, çevirme (y ekseninde simetrik), ve yakınlaştırma (zoom) yöntemleridir. Ham veri geleneksel artırma ile veri seti, 1283 normal ve 873 farenjitli halini almıştır. Geleneksel veri artırma yönteminden sonra görüntüler, veri setinde denge problemini aşmak adına, derin öğrenme yaklaşımı olan GAN tabanlı veri artırma yöntemlerinden CycleGAN kullanılarak verilerin eşlenmiş hali elde edilmeye çalışılmıştır. CycleGAN algoritması ile sentetik veri elde etme sürecinde Tensorflow'un eğitici içeriği [25], kullanılarak kod elde edilmiştir. Eğitim 100 dönem (epoch) olarak gerçekleştirilmiş ve her dönem (epoch) yaklaşık olarak 397 saniye sürmüştür. Colab ortamının GPU kullanımını sınırlı tutması sebebiyle deneysel ortam olarak NVIDIA Quadro P5000 GPU desteğine sahip bir sunucu bilgisayarda eğitim gerçekleştirilmiştir ve eğitim yaklaşık 12 saat sürmüştür. Görüntüler 256 x 256 x 3 boyutlarında yerel olarak kaydedilmiştir. Bu ortamda kullanılan Tensorflow sürümü

2.10.0'dır. Eşlenmiş olarak elde edilen sentetik görüntüler veri seti kalitesini korumak adına, her bir görüntü kullanılmamış ve uzman bir hekim tarafından sınıflandırılmıştır. Toplamda 68 normal ve 171 adet farenjitli görüntü, veri setine dahil edilmiştir. Veri setinin nihai hali çizelge 6.1'de her iki sınıf için olacak şekilde mevcuttur.

Çizelge 6.1. Veri seti örnek sayıları

	Normal	Farenjit
<b>Geleneksel + Ham Veri</b>	1283	873
<b>CycleGan</b>	68	171
<b>Toplam</b>	1351	1044



Şekil 6.1. Geleneksel veri artırma yöntemleri. a) Renk alanı parlaklık artırma, b) Döndürme (-5), c) Döndürme (+5), d) Çevirme, e) Yakınlaştırma (Zoom)

Şekil 6.1'de bu tezde kullanılan geleneksel veri artırımı örnekleri gösterilmektedir. Görüntü etiketlerinin güvenliği açısından insan fizyolojisine uygun olabilecek dönüşümler kullanılmıştır.



Şekil 6.2. Veri setinde kullanılan gerçek ve sentetik görüntüler. (Veri seti içindeki gerçek görüntüler a), b) ve CycleGAN ile oluşturulmuş sentetik veriler c), d)).

## 6.2. RESNET50 DENEYSSEL ORTAM

Çalışmamız Google Colab ve yerel bir sunucu bilgisayar olmak üzere iki farklı ortamda gerçekleştirilmiştir. Her iki ortamda da GPU desteği alınarak çalışma tamamlandı. Tensorflow versiyonu 2.12.0 olan Google Colab ortamına uyumlu şekilde, yerel sunucu bilgisayarda ise 2.10.0 versiyonu tercih edilmiştir. Veri seti Google Colab ortamı için Google Drive üzerinden çekildi. Veri setinin, eğitim (0.8), doğrulama (0.1) ve test (0.1) için bölünebilmesi açısından SplitFolders kütüphanesi kullanılmıştır. Görüntüler RGB kanalında, 224 x 224 x 3 boyutlarında kullanılmıştır. ResNet50 mimarisinin gerektirdiği üzere bu ölçeklendirme kullanılmıştır. Farklı denemelerle boyutlandırma açısından ResNet mimarisinde 224 x 224 girdi boyutunun en uygun değer olduğu gözlemlenmiştir [26]. Transfer Öğrenimi gerçekleştirmek adına ImageNet ile son katmana kadar eğitilen modelde ara katmanlarda aktivasyon fonksiyonu olarak ReLU kullanıldı. Sınıflandırma gereği son bağlı katmanda ise



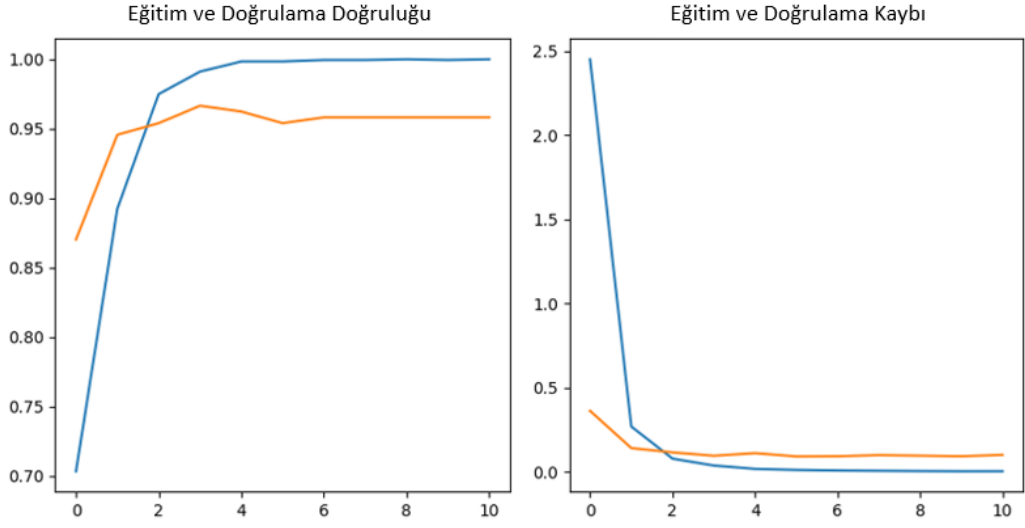
Sigmoid aktivasyon fonksiyonu kullanılmıştır. Mimaride öğrenme oranı  $1e-4$ , iyileştirici (optimizer) olarak Adam tercih edildi. Aşırı uydurmanın oluşmasını engellemek amacıyla hem erken durdurma (early stopping) hem de bırakma (dropout) teknikleri uygulandı. Erken durdurmanın birincil koşulu olan dönem (epoch) sayısının yüksek tutulması açısından mimari 150 dönem (epoch) olacak şekilde ayarlandı. Parametre değerleri kesinleştirilmeden önce bir dizi hiper parametre denemesi gerçekleştirilmiştir. Çizelge 6.2’de hiper parametre denemeleri mevcuttur.

Çizelge 6.2. ResNet50 hiper parametre denemeleri

Öğrenme oranı	Parti boyutu	Dönem	Eğitim doğruluğu	Doğrulama doğruluğu	Test doğruluğu	Doğrulama kaybı	Test kaybı
1e-2	32	15	0.9833	%96,65	%96,68	%31,93	%11,44
1e-2	64	8	0.9890	%96,23	%96,68	%20,25	%24,06
1e-2	128	9	0.9916	%94,98	%95,44	%48,30	%18,39
1e-3	32	8	0.9854	%94,56	%97,93	%92,84	%21,17
1e-3	64	11	0.9974	%97,07	%97,51	%23,28	%16,27
1e-3	128	9	0.9984	%96,65	%97,93	%65,74	%27,22
1e-4	32	17	0.9718	%95,82	%96,27	%23,33	%26,59
1e-4	64	18	1.0000	%97,07	%97,93	%7,85	%6,42
1e-4	128	11	1.0000	%95,82	%98,76	%10,04	%3,84
1e-5	32	14	1.0000	%96,23	%97,93	%8,55	%7,47
1e-5	64	22	1.0000	%96,23	%96,68	%7,19	%7,20
1e-5	128	15	1.0000	%96,23	%97,10	%6,91	%7,49
1e-6	32	52	1.0000	%95,40	%96,27	%7,77	%7,47
1e-6	64	48	0.9974	%94,98	%95,85	%8,80	%8,51
1e-6	128	39	0.9922	%94,98	%95,02	%10,35	%10,98

### 6.2.1. Deneysel Sonuçlar

ResNet50 mimarisinin eğitimi 11. dönemde tamamlandı. Doğrulama veri setinde %95,82 ve test veri setinde %98,76 doğruluk gözlemlendi. Her dönem (epoch) için eğitim yaklaşık 12 saniye sürmüştür.



Şekil 6.3. ResNet50 eğitim grafikleri

Çizelge 6.3. ResNet50 performans metrikleri

RESNET-50	Kesinlik	Duyarlılık	AUC
Doğrulama	%94,96	%97,78	%99,62
Test	%98,54	%99,26	%99,91

### 6.3. INCEPTION-V3 DENEYSSEL ORTAM

Inception-V3 mimarimizde Tensorflow sürümü şu anda güncel versiyon olan 2.12.0'dır. Colab ortamında yine GPU desteği alınarak gerçekleştirilen eğitimimiz, veri setinin çekilebilmesi adına Google Drive kullanılarak yapılmıştır. Verilerin manuel olarak ayrılmasındansa dengeli bir veri seti oluşturmak adına SplitFolders kütüphanesinden yararlanılmıştır. Eğitim (0.8), doğrulama (0.1), test (0.1) olmak üzere üç bölüme ayrılan veri seti 1915 eğitim, 239 doğrulama ve test için 241 adet RGB formatında görüntü içermektedir. Inception-V3 mimarisinde görüntü boyutları 256 x 256 x 3 olarak ayarlanmıştır. Kullanılan parti boyutu (batch size) değeri bu mimaride 32 olarak belirlenmiştir. Bu mimari diğer eğitimlerimizden farklı olarak ön eğitime tabi tutulmuştur. Ağın ön eğitimi Yoo vd.'nin yaptığı çalışmada [3] ek olarak verilmiş olan kod baz alınarak eğitim tamamlanılmıştır. ImageNet ile eğitim yapılmayıp hazır

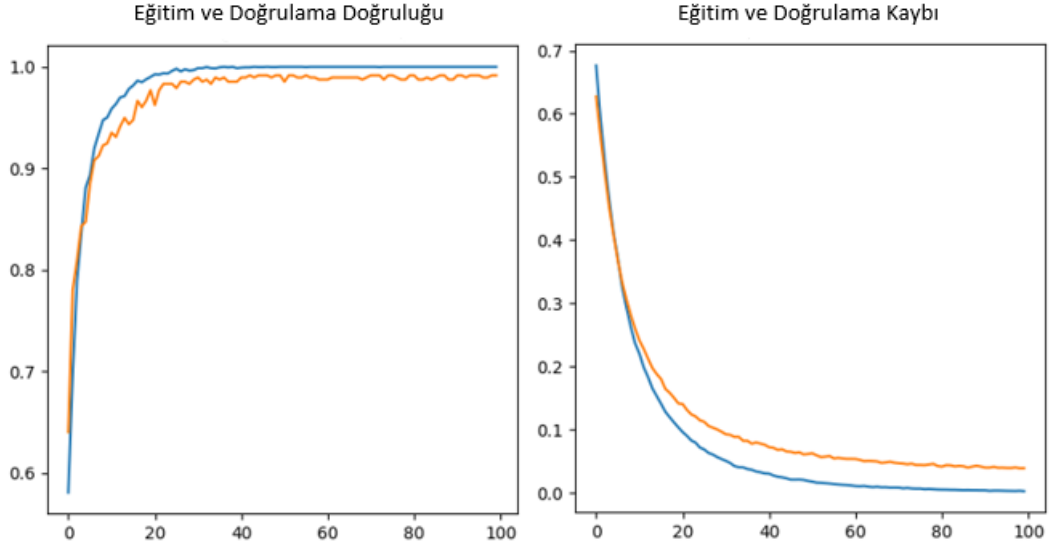
ağırlıklar kullanılmıştır. Ön eğitimde de transfer öğrenimine benzer şekilde ağ son katmana kadar dondurulup eğitim tamamlanmıştır. Giriş katmanı 1024 ve çıkış katmanı 2 olacak şekilde eğitim tamamlanmıştır. Ara katmanlarda diğer mimarilere benzer şekilde ReLU aktivasyon fonksiyonu ve çıkış katmanında Sigmoid aktivasyon fonksiyonu kullanılmıştır. Mimaride bırakma (dropout) tekniği kullanılmış ve değeri 0.6 olarak ayarlanmıştır. Öğrenme oranı (learning rate) diğer mimarilerimizden farklı olarak  $1e-6$  olacak şekilde kullanılmıştır. Erken durdurma tekniği kullanılmamış ve eğitim 100 dönem (epoch) olacak şekilde ayarlanmıştır. Adam iyileştirici (optimizer) kullanılarak modelin genellebilirliği sağlanmaya çalışılmıştır.

Çizelge 6.4. Inception-V3 hiper parametreleri ve sonuçlar

Öğrenme oranı	Parti boyutu	Dönem	Eğitim doğruluğu	Doğrulama doğruluğu	Test doğruluğu	Doğrulama kaybı	Test kaybı
<b>1e-6</b>	32	100	1.0000	%99,16	%97,51	%3,88	%4,27

### 6.3.1. Deneysel Sonuçlar

Inception-V3 mimarisinde ön eğitim yapıldığından ve erken durdurma kullanılmadığından eğitim dönem (epoch) sayısı 100 olarak belirlenmiştir. Her dönem (epoch) yaklaşık 12 saniye sürmüştür. Doğrulama veri setinde %99,16 ve test veri setinde %97,51 doğruluk oranı gözlemlenmiştir.



Şekil 6.4. Inception-V3 eğitim grafikleri

Çizelge 6.5. Inception-V3 performans metrikleri

INCEPTION-V3	Kesinlik	Duyarlılık	AUC
Doğrulama	%99,16	%99,16	%99,92
Test	%97,51	%97,51	%99,91

#### 6.4. EFFICIENTNET B0 DENEYSEL ORTAM

EfficientNet B0 mimarisinde Tensorflow sürümü olarak şu an da güncel versiyon olan 2.12.0 kullanılmıştır. Eğitimler Colab ortamında GPU desteği alınarak tamamlanmıştır. Veri seti Google Drive üzerinden çekilmiştir. Eğitimde kullanılan veri seti dengeli şekilde bölünebilmek adına SplitFolders kütüphanesi kullanılmıştır. Eğitim (0.8), doğrulama (0.1), test (0.1) olacak şekilde veri seti üç parçaya rastgele olarak bölünmüştür. EfficientNet B0 mimarisinin çözünürlük kısıtı olan giriş verilerinin 224 x 224 x 3 boyutlandırması yapılmıştır. Model ImageNet ile eğitilerek transfer öğrenimi ile mimari Keras üzerinden çekilmiştir. Transfer öğrenimi gerçekleştirilirken model son katmana kadar dondurulmuştur. Yine EfficientNet mimarisinin kısıtı olan aktivasyon fonksiyonu Swish aktivasyonu olarak tercih

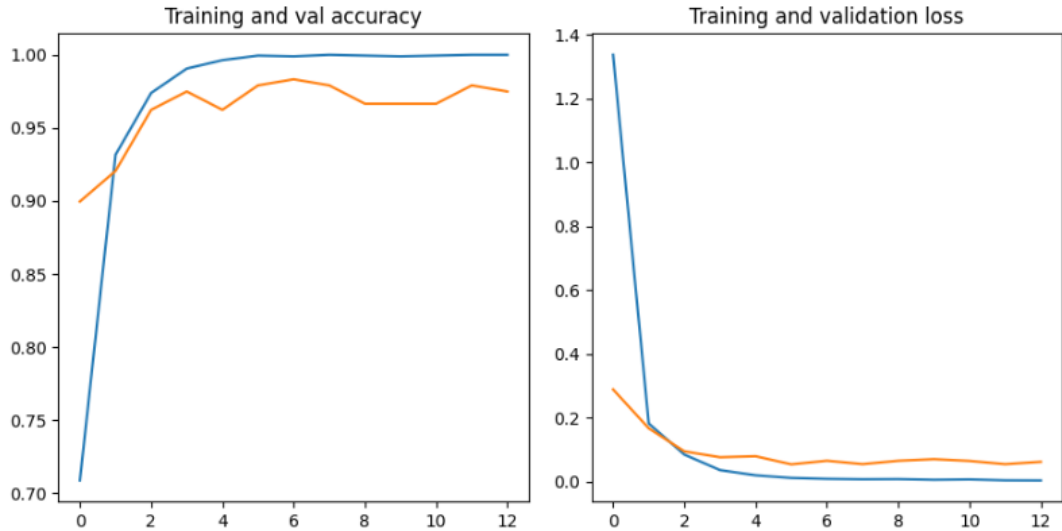
edilmiştir. Son tam bağlı katmanda ise aktivasyon fonksiyonu diğer çalışılan mimariler ile aynı şekilde Sigmoid olarak ayarlanmıştır. Model üzerinde bırakma (dropout) uygulanmıştır. Adam iyileştirici (optimizer) ile eğitim yapılmıştır. Eğitimde erken durdurma (early stopping) doğrulama setinin yedi dönem (epoch) boyunca düşmemesi üzerine uygulanmıştır. Erken durdurma gereği eğitim dönem (epoch) sayısı 150 olarak belirlenmiştir. Parti boyutu (batch size) değeri 128 ve öğrenme oranı (learning rate)  $1e-4$  en uygun değerler olarak gözlemlenmiştir.

Çizelge 6.6. EfficientNet B0 hiper parametreleri

Öğrenme oranı	Parti boyutu	Dönem	Eğitim doğruluğu	Doğrulama doğruluğu	Test doğruluğu	Doğrulama kaybı	Test kaybı
$1e-4$	128	13	1.0000	%97,49	%99,17	%6,11	%1,59

#### 6.4.1. Deneysel Sonuçlar

EfficientNet B0 mimarisinde eğitim 13. dönemde tamamlandı. Doğrulama veri setinde %97,49 ve test veri setinde %99,17 doğruluk oranı gözlemlendi. Her dönem yaklaşık 11 saniye sürmüştür.



Şekil 6.5. EfficientNet B0 eğitim grafikleri

Çizelge 6.7. EfficientNet B0 performans metrikleri

<b>EFFICIENTNET B0</b>	<b>Kesinlik</b>	<b>Duyarlılık</b>	<b>AUC</b>
<b>Doğrulama</b>	%96,40	%99,26	%99,88
<b>Test</b>	%98,55	%100,00	%100,00

## 6.5. EFFICIENTNET B1 DENEYSEL ORTAM

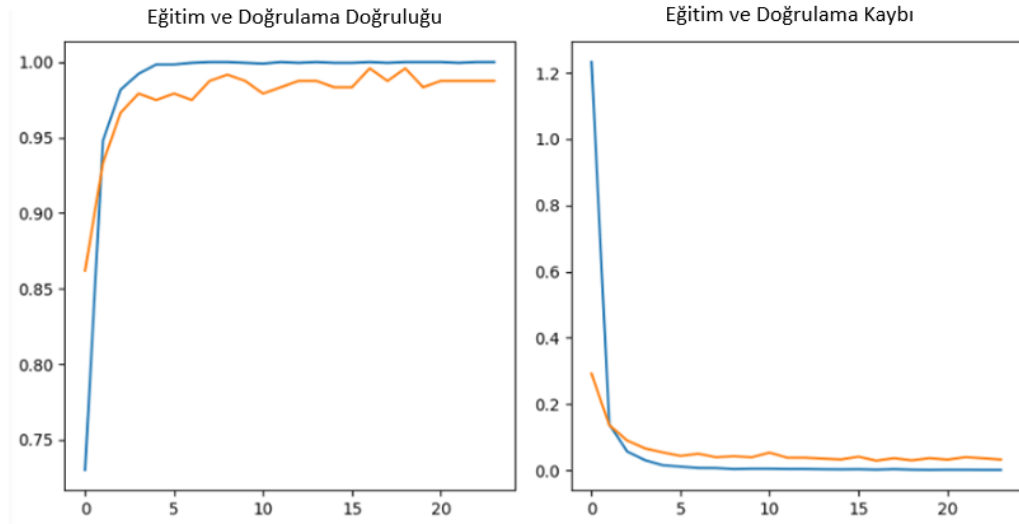
EfficientNet B1 mimarisinde veri seti her mimaride olduğu gibi Google Drive aracılığı ile çekilmiştir. GPU desteği ile yine Colab ortamında eğitim gerçekleştirilmiştir. Veri seti SplitFolders kütüphanesi ile rastgele olarak üç gruba ayrılmış, eğitim (0.8), doğrulama (0.1), test (0.1) olacak şekilde veri seti mimariye beslenmiştir. Toplam 2395 veri, eğitim için 1915, doğrulama için 239, test için ise 241 görüntü RGB formatta mimariye beslenmektedir. Bu oran her mimaride aynı kullanılmıştır. Eğitimde transfer öğrenmenin gerçekleştirilebilmesi adına model ImageNet ile eğitilmiş bu eğitim son katmana kadar dondurularak yapılmıştır. EfficientNet B1 mimarisinin giriş verilerinin çözünürlük sınırlandırması 240 x 240 x 3 olduğundan veriler bu şekilde boyutlandırılmıştır. Aktivasyon fonksiyonu olarak ara katmanlarda Swish aktivasyon fonksiyonu tercih edilmiştir. Son tam bağlı katmanda ise yine her çalışılan modelde olduğu gibi Sigmoid aktivasyon fonksiyonu kullanılmıştır. Eğitimde bırakma (dropout) 0.5 olarak ayarlanmıştır. İyileştirici olarak Adam kullanılmış, öğrenme oranı 1e-4 olarak optimal bulunmuştur. Parti boyutu (batch size) değeri 128 olarak ayarlanmıştır. Doğrulama kaybının 7 dönem (epoch) boyunca azalmadığı sürece eğitimin durdurulacağı kriteri erken durdurma (early stopping) şartı olarak belirlenmiştir. Erken durdurma kullanmanın gereği olacak şekilde mimari eğitiminde 150 dönem (epoch) eğitim yapılacağı belirlenmiştir. Ortam özellikleri olarak Tensorflow 2.12.0 sürümündedir.

Çizelge 6.8. EfficientNet B1 hiper parametreleri

Öğrenme oranı	Parti boyutu	Dönem	Eğitim doğruluğu	Doğrulama doğruluğu	Test doğruluğu	Doğrulama kaybı	Test kaybı
1e-4	128	24	1.0000	%98,74	%99,59	%3,22	%3,17

### 6.5.1. Deneysel Sonuçlar

EfficientNet B1 mimarisinde eğitim 24. dönemde tamamlanmıştır. Doğrulama veri setinde %98,74 ve test veri setinde %99,59 doğruluk oranı gözlemlenmiştir. Eğitimde her dönem yaklaşık 11 saniye sürmüştür.



Şekil 6.6. EfficientNet B1 eğitim grafikleri

Çizelge 6.9. EfficientNet B1 performans metrikleri

EFFICIENTNET B1	Kesinlik	Duyarlılık	AUC
Doğrulama	%97,83	%100,00	%99,95
Test	%100,00	%99,26	%99,60

## 6.6. EFFICIENTNET B2 DENEYSSEL ORTAM

Son olarak EfficientNet B2 mimarisinde Tensorflow sürümü 2.12.0'dır. Colab ortamında eğitim tamamlanmıştır. Veri seti diğer çalışılan algoritmalarındaki gibi Google Drive üzerinden çekilmiştir. Veri seti SplitFolders kütüphanesi kullanılarak eğitim (0.8), doğrulama (0.1) ve test (0.1) olarak üç parçaya bölünmüştür. Mimaride erken durdurma (early stopping) 7 dönem (epoch) boyunca doğrulama kaybının azalmadığı durumda uygulanmıştır. Eğitimde erken durdurma kullanılması sebebiyle dönem (epoch) 150 olarak ayarlanmıştır. Mimari transfer öğrenimi sebebiyle Keras üzerinden çekilmiştir ve ImageNet ile son katmana kadar eğitimi yapılmıştır. Ara katmanlarda EfficientNet mimarisi gereği aktivasyon fonksiyonu olarak Swish aktivasyonu kullanılmıştır. Son tam bağlı katmanda ise diğer çalışılan algoritmalarda olduğu gibi Sigmoid aktivasyon fonksiyonu kullanılmıştır. EfficientNet B2 algoritmasının veri girişindeki kısıtı sebebiyle veri seti 260 x 260 x 3 boyutlarında mimariye beslenmiştir. Bırakma (dropout) bu mimaride de 0.5 değerinde uygulanmıştır. İyileştirici (optimizer) olarak Adam ve öğrenme oranı 1e-4 olarak ayarlanmıştır. Parti boyutu (batch size) değeri 128'dir.

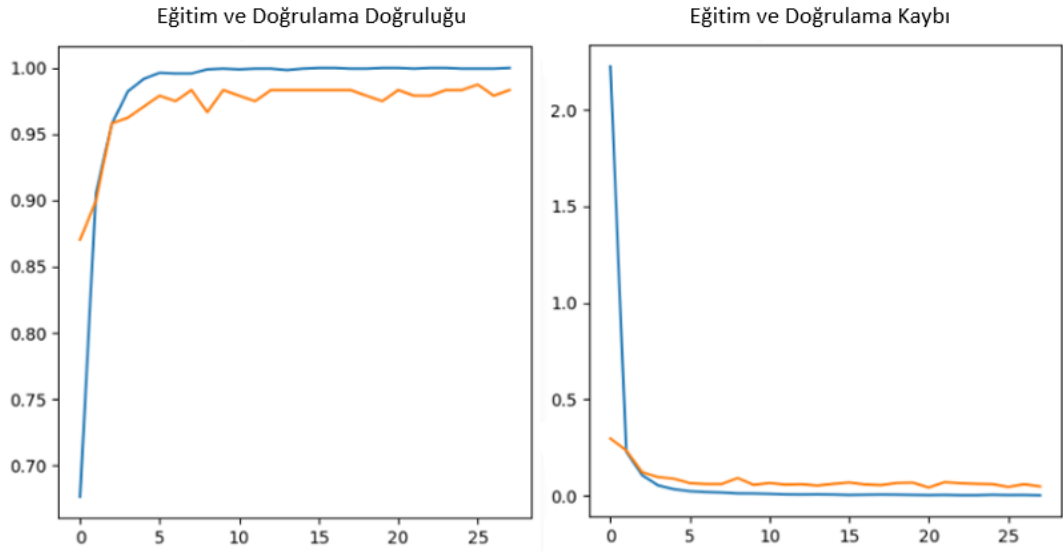
Çizelge 6.10. EfficientNet B2 hiper parametreleri

Öğrenme oranı	Parti boyutu	Dönem	Eğitim doğruluğu	Doğrulama doğruluğu	Test doğruluğu	Doğrulama kaybı	Test kaybı
1e-4	128	28	1.000	%98,33	%99,59	%4,77	%2,09

### 6.6.1. Deneysel Sonuçlar

EfficientNet B2 mimarisinde eğitim 28. dönemde tamamlandı. Doğrulama veri setinde %98,33 ve test veri setinde %99,59 doğruluk oranı gözlemlenmiştir. Her dönem yaklaşık 13 saniye sürmüştür.





Şekil 6.7. EfficientNet B2 eğitim grafikleri

Çizelge 6.11. EfficientNet B2 performans metrikleri

<b>EFFICIENTNET B2</b>	<b>Kesinlik</b>	<b>Duyarlılık</b>	<b>AUC</b>
<b>Doğrulama</b>	%97,12	%100,00	%99,91
<b>Test</b>	%100,00	%99,26	%99,98

## BÖLÜM 7

### TARTIŞMA VE SONUÇLAR

Bu çalışmada internet tabanlı, açık erişimli boğaz görüntülerinin olduğu farenjit ve normal görüntülerden oluşan iki sınıflı bir veri seti [3], kullanılmıştır. Veri seti 5 farklı geleneksel veri artırma yöntemi ile genişletildi. İnsan fizyolojisine ters düşecek afin yöntemler kullanılmamaya çalışılmıştır. Geleneksel dönüşümlerden sonra GAN tabanlı veri artırma yöntemi olarak CycleGAN tercih edilmiştir. Veri kümesinin kalitesini korumak en az veri kümesinin doğruluğunu korumak kadar önemli olacağından, her görüntünün CycleGAN algoritması tarafından eşlenmiş sürümü mevcut olsa dahi veri setine dahil edilmemiştir. CycleGAN algoritması tarafından üretilmiş sentetik görüntülerin 68 adet normal boğaz görüntüsü ve 171 adet farenjitli boğaz görüntüsü veri setine dahil edilecek şekilde uzman bir hekim tarafından uygun bulunmuştur. Açık erişimli olan Yoo vd.'nin yaptığı çalışmada [3] paylaşılan kodlardan gözlemlendiği üzere algoritma üzerinde ön eğitim yapılmıştır. Bu çalışmada ise transfer öğrenimi gerçekleştirerek ImageNet [30] veri seti üzerinde eğitilmiş mimarilere veri beslemesi yapılarak çalışılmıştır. ImageNet veri kümesi her ne kadar boğaz görüntüleri içermese dahi, uygulanan her mimaride performans artışı gözlemlenmektedir. Veri setini manuel olarak ayırmak veri setinin sınıflar arasında dengesini bozacağından bu çalışmada SplitFolders kütüphanesinden yararlanılmıştır. Mimariler gereği girdi veri boyutlandırması 224 x 224, 240 x 240, 256 x 256, 260 x 260 olacak şekilde farklı boyutlarda yapılmıştır. Eğitimlerde aşırı uyumu (overfitting) engellemek adına, bırakma (dropout) ve erken durdurma (early stopping) yöntemleri en uygun değerler bulunacak şekilde ampirik hareket edilerek uygulanmıştır. Erken durdurmanın uygulandığı her mimaride dönem (epoch) sayılarını yüksek tutmamıza rağmen, yöntemin çalışma mantığından ötürü mimarilerde eğitim hızlarının artması sağlanmıştır. Belirli sayıda dönem (epoch) ile çalışmak algoritmanın performansını artırmayacağı gibi, eğitim süresinin de artışına sebep olur. Çizelge 7.1'de yaptığımız çalışma ile diğer yapılan çalışmaların performans metrikleri açısından bir

karşılaştırması mevcuttur. Karşılaştırılan mimariler metriklerin sağlıklı kıyaslaması yapılabilmesi açısından aynı ham veri setini kullanan çalışmalardan seçilmiştir. Sonuçlarımıza göre;

1. ResNet50 algoritmasında mimariye uygun olarak girdi verilerinin boyutlandırılması ve transfer öğrenimi uygulanması bunun dışında hiper parametrelerin ampirik olarak ayarlanması algoritmanın performansını artırmıştır.
2. Eğitim, doğrulama ve test verilerinin manuel olarak ayrılması veri setinin sınıflar arası dengesini bozacağından kütüphane yardımıyla rastgele olarak ayrılması performansı etkilemektedir.
3. Inception-V3 algoritmasında algoritmaya transfer öğrenimi uygulanmasa dahi mimarideki hiper parametrelerin ampirik olarak ayarlanması ve yine mimarinin girdi verisi boyutuna dikkat edilerek algoritmaya beslenmesinin başarımı artırdığı gözlemlenmektedir.
4. EfficientNet B0, B1 ve B2 algoritmalarının karşılaştırılan diğer tüm mimarilere göre çok daha başarılı olduğu gözlemlenmektedir.
5. EfficientNet mimarilerinin kendi içlerinde karşılaştırmasında, bulduğumuz sonuçlara göre B1 ve B2 mimarisi ele alınırsa kesinlik (precision) ve duyarlılık (recall) tahminlerinin aynı olduğu gözlemlenmektedir. Bu çalışmada hastalık tespiti çalışıldığından sağlıklı bir bireyin hasta olarak sınıflandırılmaması ya da hasta olan bireyin sağlıklı olarak sınıflandırılmamasının elzem olduğu düşünülecek olursa yanlış negatif (FN) tahminlerin esas olduğu duyarlılık (recall) metriği ve yanlış pozitif (FP) tahminlerin esas olduğu kesinlik (precision) metriği ön plana çıkmaktadır. Bu durumda EfficientNet B1 ve B2 mimarilerinin duyarlılık ve kesinlik değeri açısından başarılı olduğu gözlemlenmektedir.

Çizelge 7.1. İlgili alanda çalışılan mimari performans metrikleri

Mimari	AUC	Doğruluk	Duyarlılık	Kesinlik
<b>ResNet50</b>	0.999	0.987	0.992	0.985
<b>Inception-V3</b>	0.999	0.975	0.975	0.975
<b>EfficientNet B0</b>	1.000	0.991	1.000	0.985
<b>EfficientNet B1</b>	0.996	0.995	0.992	1.000
<b>EfficientNet B2</b>	0.999	0.995	0.992	1.000
<b>ResNet50 [3]</b>	0.992	0.953	0.929	-
<b>Inception-V3 [3]</b>	0.987	0.949	0.929	-
<b>MobileNet V2[3]</b>	0.981	0.937	0.908	-
<b>VGG 16 [5]</b>	-	0.98	0.98	0.98
<b>VGG 19 [5]</b>	-	0.98	0.98	0.98
<b>DenseNet 121[5]</b>	-	0.94	0.93	0.93
<b>AlexNet [5]</b>	-	0.86	0.87	0.87
<b>ResNet 152 [5]</b>	-	0.82	0.83	0.84

## KAYNAKLAR

- [1] A. P. Susanto *et al.*, “Building an artificial intelligence-powered medical image recognition smartphone application: What medical practitioners need to know,” *Informatics Med. Unlocked*, vol. 32, no. July, p. 101017, 2022, doi: 10.1016/j.imu.2022.101017.
- [2] A. O. Topal, R. Chitic, and F. Leprévost, “One evolutionary algorithm deceives humans and ten convolutional neural networks trained on ImageNet at image recognition,” *Appl. Soft Comput.*, vol. 143, p. 110397, 2023, doi: 10.1016/j.asoc.2023.110397.
- [3] T. K. Yoo, J. Y. Choi, Y. Jang, E. Oh, and I. H. Ryu, “Toward automated severe pharyngitis detection with smartphone camera using deep learning networks,” *Comput. Biol. Med.*, vol. 125, p. 103980, Oct. 2020, doi: 10.1016/j.compbiomed.2020.103980.
- [4] Istiadi, E. B. Sulistiarini, R. Joegijantoro, A. N. Suksmawati, K. S. Nugroho, and I. Akbar, “An Integrated E-Learning Fuzzy Expert System for Acute Pharyngitis Diagnosis,” *2022 6th Int. Conf. Inf. Technol. Inf. Syst. Electr. Eng.*, pp. 1–6, 2023, doi: 10.1109/icitisee57756.2022.10057861.
- [5] I. Wijayanto, I. D. Irawati, F. Fahrozi, and S. Hadiyoso, “Computer Aided Detection System for Pharyngitis Based on Convolutional Neural Network,” *APWiMob 2022 - Proc. 2022 IEEE Asia Pacific Conf. Wirel. Mob.*, pp. 1–4, 2022, doi: 10.1109/APWiMob56856.2022.10014256.
- [6] M. A. Kızrak, Z. Müftüoğlu, and T. Yıldırım, “Limitations and challenges on the diagnosis of COVID-19 using radiology images and deep learning,” *Data Sci. COVID-19 Vol. 1 Comput. Perspect.*, pp. 91–115, 2021, doi: 10.1016/B978-0-12-824536-1.00007-1.
- [7] M. Vilares Ferro, Y. Doval Mosquera, F. J. Ribadas Pena, and V. M. Darriba Bilbao, “Early stopping by correlating online indicators in neural networks,” *Neural Networks*, vol. 159, pp. 109–124, 2023, doi: 10.1016/j.neunet.2022.11.035.
- [8] S. Nazir, D. M. Dickson, and M. U. Akram, “Survey of explainable artificial

- intelligence techniques for biomedical imaging with deep neural networks,” *Comput. Biol. Med.*, vol. 156, no. August 2022, p. 106668, 2023, doi: 10.1016/j.compbimed.2023.106668.
- [9] K. Santosh, N. Das, and S. Ghosh, “Introduction,” *Deep Learn. Model. Med. Imaging*, pp. 1–27, 2022, doi: 10.1016/b978-0-12-823504-1.00011-8.
- [10] K. Santosh, N. Das, and S. Ghosh, *Deep learning models*. 2022.
- [11] N. McCarthy *et al.*, “Enterprise imaging and big data: A review from a medical physics perspective,” *Phys. Medica*, vol. 83, no. December 2020, pp. 206–220, 2021, doi: 10.1016/j.ejmp.2021.04.004.
- [12] S. Uhlig, I. Alkhasli, F. Schubert, C. Tschöpe, and M. Wolff, “A review of synthetic and augmented training data for machine learning in ultrasonic non-destructive evaluation,” *Ultrasonics*, vol. 134, no. June 2022, 2023, doi: 10.1016/j.ultras.2023.107041.
- [13] C. Shorten and T. M. Khoshgoftaar, “A survey on Image Data Augmentation for Deep Learning,” *J. Big Data*, vol. 6, no. 1, 2019, doi: 10.1186/s40537-019-0197-0.
- [14] K. Weiss, T. M. Khoshgoftaar, and D. D. Wang, *A survey of transfer learning*, vol. 3, no. 1. Springer International Publishing, 2016.
- [15] S. J. Pan and Q. Yang, “A survey on transfer learning,” *IEEE Trans. Knowl. Data Eng.*, vol. 22, no. 10, pp. 1345–1359, 2010, doi: 10.1109/TKDE.2009.191.
- [16] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, “Dropout: A simple way to prevent neural networks from overfitting,” *J. Mach. Learn. Res.*, vol. 15, pp. 1929–1958, 2014.
- [17] A. Lodwich, Y. Rangoni, and T. Breuel, “Evaluation of robustness and performance of Early Stopping Rules with Multi Layer Perceptrons,” *Proc. Int. Jt. Conf. Neural Networks*, pp. 1877–1884, 2009, doi: 10.1109/IJCNN.2009.5178626.
- [18] G. Foucher and S. Faure, “What’s pharyngitis?,” *Actual. Pharm.*, vol. 61, no. 612, pp. 59–61, 2022, doi: 10.1016/j.actpha.2021.12.015.
- [19] B. Askarian, S. C. Yoo, and J. W. Chong, “Novel image processing method for detecting strep throat (Streptococcal pharyngitis) using smartphone,” *Sensors (Switzerland)*, vol. 19, no. 15, pp. 1–17, 2019, doi: 10.3390/s19153307.
- [20] K. Jirawanitcharoen, S. Kiattisin, A. Leelasantitham, and P. Chairapa, “A

- method of detecting tonsillitis images based on medical knowledge and neural network,” *Proc. - 2009 2nd IEEE Int. Conf. Comput. Sci. Inf. Technol. ICCSIT 2009*, pp. 125–128, 2009, doi: 10.1109/ICCSIT.2009.5234443.
- [21] F. Garcea, A. Serra, F. Lamberti, and L. Morra, “Data augmentation for medical imaging: A systematic literature review,” *Comput. Biol. Med.*, vol. 152, no. November 2022, 2023, doi: 10.1016/j.compbiomed.2022.106391.
- [22] M. Priya and M. Nandhini, “Detection of fetal brain abnormalities using data augmentation and convolutional neural network in internet of things,” *Meas. Sensors*, vol. 28, no. June, p. 100808, 2023, doi: 10.1016/j.measen.2023.100808.
- [23] I. Goodfellow *et al.*, “Generative adversarial networks,” *Commun. ACM*, vol. 63, no. 11, pp. 139–144, 2020, doi: 10.1145/3422622.
- [24] A. Mikołajczyk and M. Grochowski, “Data augmentation for improving deep learning in image classification problem,” in *2018 International Interdisciplinary PhD Workshop, IIPHDW 2018*, Jun. 2018, pp. 117–122, doi: 10.1109/IIPHDW.2018.8388338.
- [25] “CycleGAN | TensorFlow Core.” <https://www.tensorflow.org/tutorials/generative/cyclegan?hl=tr> (accessed May 28, 2023).
- [26] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, vol. 2016-Decem, pp. 770–778, 2016, doi: 10.1109/CVPR.2016.90.
- [27] S.-C. Huang and T.-H. Le, “Convolutional neural network architectures,” *Princ. Labs Deep Learn.*, pp. 201–217, 2021, doi: 10.1016/b978-0-323-90198-7.00001-x.
- [28] D. McNeely-White, J. R. Beveridge, and B. A. Draper, “Inception and ResNet features are (almost) equivalent,” *Cogn. Syst. Res.*, vol. 59, pp. 312–318, 2020, doi: 10.1016/j.cogsys.2019.10.004.
- [29] O. Russakovsky *et al.*, “ImageNet Large Scale Visual Recognition Challenge,” *Int. J. Comput. Vis.*, vol. 115, no. 3, pp. 211–252, 2015, doi: 10.1007/s11263-015-0816-y.
- [30] “ImageNet.” <https://www.image-net.org/> (accessed Jul. 19, 2023).
- [31] C. Szegedy *et al.*, “Going deeper with convolutions,” *Proc. IEEE Comput. Soc.*

- Conf. Comput. Vis. Pattern Recognit.*, vol. 07-12-June, pp. 1–9, 2015, doi: 10.1109/CVPR.2015.7298594.
- [32] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “ImageNet classification with deep convolutional neural networks,” *Commun. ACM*, vol. 60, no. 6, pp. 84–90, 2017, doi: 10.1145/3065386.
- [33] G. Meena, K. K. Mohbey, and S. Kumar, “Sentiment analysis on images using convolutional neural networks based Inception-V3 transfer learning approach,” *Int. J. Inf. Manag. Data Insights*, vol. 3, no. 1, p. 100174, 2023, doi: 10.1016/j.jjime.2023.100174.
- [34] N. D. Kathamuthu *et al.*, “A deep transfer learning-based convolution neural network model for COVID-19 detection using computed tomography scan images for medical applications,” *Adv. Eng. Softw.*, vol. 175, no. October 2022, p. 103317, 2023, doi: 10.1016/j.advengsoft.2022.103317.
- [35] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, “Rethinking the Inception Architecture for Computer Vision,” *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, vol. 2016-Decem, pp. 2818–2826, 2016, doi: 10.1109/CVPR.2016.308.
- [36] R. Mohanasundaram, A. S. Malhotra, R. Arun, and P. S. Periasamy, *Deep Learning and Semi-Supervised and Transfer Learning Algorithms for Medical Imaging*. Elsevier Inc., 2019.
- [37] M. Tan and Q. V. Le, “EfficientNet: Rethinking model scaling for convolutional neural networks,” *36th Int. Conf. Mach. Learn. ICML 2019*, vol. 2019-June, pp. 10691–10700, 2019.
- [38] K. Ali, Z. A. Shaikh, A. A. Khan, and A. A. Laghari, “Multiclass skin cancer classification using EfficientNets – a first step towards preventing skin cancer,” *Neurosci. Informatics*, vol. 2, no. 4, p. 100034, 2022, doi: 10.1016/j.neuri.2021.100034.
- [39] P. Yadav, N. Menon, V. Ravi, S. Vishvanathan, and T. D. Pham, “EfficientNet convolutional neural networks-based Android malware detection,” *Comput. Secur.*, vol. 115, 2022, doi: 10.1016/j.cose.2022.102622.
- [40] Ü. Atila, M. Uçar, K. Akyol, and E. Uçar, “Plant leaf disease classification using EfficientNet deep learning model,” *Ecol. Inform.*, vol. 61, no. October 2020, p. 101182, 2021, doi: 10.1016/j.ecoinf.2020.101182.



## ÖZGEÇMİŞ

Tuğba TAK, ilk ve orta öğrenimini İstanbul'da tamamladı. Sağmalcılar Anadolu Lisesi Fen Bilimleri Alanı'ndan 2012 yılında mezun oldu. Lisans öğrenimine 2013 yılında Karabük Üniversitesi Bilgisayar Mühendisliği Bölümü'nde başladı ve 2019 yılında mezun olduktan sonra 2020 yılında Karabük Üniversitesi Lisansüstü Eğitim Enstitüsü Fen Bilimleri Alanı Bilgisayar Mühendisliği Anabilim Dalı'nda lisansüstü öğrenimine başlamış, 2023 yılında öğrenimini tamamlamıştır.