

**T.C.**  
**TRAKYA ÜNİVERSİTESİ**  
**FEN BİLİMLERİ ENSTİTÜSÜ**

**SAĞLIK ALANINDA DERİN ÖĞRENME YÖNTEMLERİNİN İNCELENMESİ**

**Yaren ÖZSOY**

**YÜKSEK LİSANS TEZİ**

**BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI**

**Tez Danışmanı: Dr. Öğr. Üyesi Deniz TAŞKIN**

**EDİRNE 2022**

**YAREN ÖZSOY**'un hazırladığı “**Sağlık Alanında Derin Öğrenme Yöntemlerinin İncelenmesi**” başlıklı bu tez, tarafımızca okunmuş, kapsam ve niteliği açısından **Bilgisayar Mühendisliği** Anabilim Dalında bir **Yüksek lisans tezi** olarak kabul edilmiştir.

Jüri Üyeleri

İmza

Dr. Öğr. Üyesi Deniz TAŞKIN

.....

Dr. Öğr. Üyesi Cem TAŞKIN

.....

Dr. Öğr. Üyesi Selçuk YAZAR

.....

Tez Savunma Tarihi: 27 / 04 / 2022

Bu tezin Yüksek Lisans tezi olarak gerekli şartları sağladığımı onaylarım.

İmza

Dr. Öğr. Üyesi Deniz TAŞKIN

Tez Danışmanı

Trakya Üniversitesi Fen Bilimleri Enstitüsü onayı

Prof. Dr. Hüseyin Rıza Ferhat KARABULUT

Fen Bilimleri Enstitüsü Müdürü

**T.Ü.FEN BİLİMLERİ ENSTİTÜSÜ**  
**BİLGİSAYAR MÜHENDİSLİĞİ YÜKSEK LİSANS PROGRAMI**  
**DOĞRULUK BEYANI**

Trakya Üniversitesi Fen Bilimleri Enstitüsü, tez yazım kurallarına uygun olarak hazırladığım bu tez çalışmasında, tüm verilerin bilimsel ve akademik kurallar çerçevesinde elde edildiğini, kullanılan verilerde tahrifat yapılmadığını, tezin akademik ve etik kurallara uygun olarak yazıldığını, kullanılan tüm literatür bilgilerinin bilimsel normlara uygun bir şekilde kaynak gösterilerek ilgili tezde yer aldığını ve bu tezin tamamı ya da herhangi bir bölümünün daha önceden Trakya Üniversitesi ya da farklı bir üniversitede tez çalışması olarak sunulmadığını beyan ederim.

27 / 04 / 2022

*Yaren ÖZSOY*

*İmza*

Yüksek Lisans Tezi  
Sağlık Alanında Derin Öğrenme Yöntemlerinin İncelenmesi  
T.Ü. Fen Bilimleri Enstitüsü  
Bilgisayar Mühendisliği Anabilim Dalı

## ÖZET

Günümüzde hızla artan nüfus ile birçok alanda olduğu gibi sağlık alanında da iş yoğunluğu artmaktadır. Artan iş yoğunluğunun getirdiği yükü hafifletmek amacıyla sağlık alanında doktorlarımıza yardımcı olmak için hastalığın teşhisinde kullanılabilecek model ve algoritmalar geliştirilmektedir.

Hastalığın özellikle erken teşhis edilmesi Dünyada ve Türkiye’de ölüm oranını azaltmak için çok önemli etkidir. Bu yüzden derin öğrenme yöntemleri kullanarak birçok hastalığın tespit edilmesi sağlık açısından önem taşımaktadır.

Hastalıklar arasında en yaygın ve ölümlü sonuçlanan hastalıklardan biri zatürredir. Zatürre hastalığı, görüntüleme tekniği olan röntgen ile tespit edilmektedir. Bu tez çalışmasında da akciğer röntgenlerinden oluşan veri kümesi içerisinde etiketli veriler ile sınıflandırma modeli kullanılmıştır. Ayrıca bu alandaki mevcut ve benzer çalışmalar incelenmiştir. İncelenen çalışmalar doğrultusunda performans açısından verimli olduğunu düşündüğümüz model ve metotlar tercih edilmektedir. Sağlık alanında verilerin kısıtlı olmasından kaynaklı kullanılacak olan modellerin uygunluğu başarı oranınının daha yüksek olmasındaki temel faktördür.

İlk aşama olarak veri kümesi, zatürre hastası olan ve zatürre hastası olmayan insanların akciğer röntgen görüntülerinden oluşmaktadır. Röntgen görüntüleri zatürre ve normal olmak üzere etiketlenmiştir. Bu etiketli verileri ile AlexNet derin öğrenme modeli kullanılarak eğitim işlemi gerçekleştirilmiştir. İkinci aşama olarak zatürrenin sebebinin bakteri kaynaklı mı virüs kaynaklı mı olduğunun tespit edilmesi için veri seti

oluřturulmuřtur. Veri setinde daha 6nce kullanmıř olduėumuz veri setindeki r6ntgen g6r6nt6leri etiketlerine g6re ayrılarak hazırlanmıřtır. Daha sonra mevcut olan veri setindeki r6ntgen g6r6nt6lerinden zat6rre hastalığının nedeninin bakteri mi vir6s kaynaklı mı olduėunu anlamak amacıyla AlexNet derin 6ėrenme modeli kullanarak eėitim iřlemleri ger6ekleřtirilmiřtir. 6ç6nc6 ařama olarak normal, zat6rre ve covid19 hastalarının g6ėė6s r6ntgenlerinden oluřan veri seti kullanılmıř. Veri setleri AlexNet ve GoogLeNet derin 6ėrenme modelleri ile eėitilmiř ve sonu6ları karřılařtırılmıřtır. Modellerin eėitimleri ve testleri ařamasında Nvidia Digits kullanılmıřtır.

Yıl : 2022

Sayfa Sayısı : 124

Anahtar Kelimeler : Derin 6ėrenme, AlexNet, Derin 6ėrenme Modeli, Tıbbi Biliřim

Master Thesis  
Investigation of Deep Learning Methods in Healthcare  
Trakya University Institute of Natural and Applied Sciences  
Department of Computer Engineering

## **ABSTRACT**

Today, with a rapidly growing population, the intensity of work in the field of Health is increasing, as in many areas. In order to alleviate the burden of increasing labor intensity, models and algorithms are being developed that can be used to diagnose the disease to help our doctors in the field of Health.

Early detection of the disease is a very important factor for reducing mortality in the world and in Turkey. For this reason, detecting many diseases using deep learning methods is important for health.

Among diseases, one of the most common and fatal diseases is pneumonia. Pneumonia disease is detected by X-ray imaging technique. In this thesis, a classification model with labeled data was used in a dataset consisting of lung X-rays. In addition, current and similar studies in this area have been viewed. In line with the studies viewed, models and methods that we think are efficient in terms of performance are preferred. The suitability of the models to be used due to the limited data in the field of Health is the main factor in the higher success rate.

As a first stage, the data set consists of lung X-ray images of people with pneumonia and without pneumonia. X-ray images are labeled as pneumonia and normal. With this tag data, the training process was carried out using the AlexNet deep learning model. As a second stage, a data set was created to determine whether the cause of pneumonia was caused by bacteria or viruses. The X-ray images in the data set that we have been used before were prepared by separating them according to their labels. Then, training operations were carried

out using the AlexNet deep learning model to determine whether the cause of pneumonia disease was caused by bacteria or viruses from the X-ray images in the current data set. As a third stage, a data set consisting of chest X-rays of normal, pneumonia and covid19 patients was used. The datasets were trained with AlexNet and GoogLeNet deep learning models and their results were compared. Nvidia Digits were used during the training and testing phase of the models.

Year : 2022

Number of Pages : 124

Keywords : Deep Learning, AlexNet, Deep Learning Model, Medical Informatics

## TEŐEKKÜR

Yüksek lisans öğrenimim ve tez çalışmam boyunca bilgi ve tecrübeleriyle destek olan saygıdeğer danışman hocam Dr. Öğr. Üyesi Deniz TAŐKIN'a verdiği emek ve katkılarından dolayı teşekkür ederim. Ayrıca tüm hayatım boyunca bana her zaman destek olan aileme teşekkür ederim.



## İÇİNDEKİLER

ÖZET .....	iv
ABSTRACT .....	vi
TEŞEKKÜR .....	viii
KISALTMALAR DİZİNİ .....	xii
ŞEKİLLER DİZİNİ .....	xiii
ÇİZELGELER DİZİNİ .....	xvi
BÖLÜM 1 .....	1
GİRİŞ .....	1
BÖLÜM 2 .....	4
LİTERATÜR ARAŞTIRMASI .....	4
BÖLÜM 3 .....	9
MATERYAL METOD .....	9
3.1. Yapay Zekâ .....	9
3.1. Makine Öğrenmesi .....	9
3.1. Derin Öğrenme .....	10
3.2. Yapay Sinir Ağları .....	13
3.3. Evrişimsel Sinir Ağları .....	14
3.3.1. Giriş Katmanı .....	15
3.3.2. Evrişim Katmanı .....	16

3.3.3. Havuzlama Katmanı.....	17
3.3.4. Tam Bağlantılı Katman .....	19
3.3.5. ReLu (Rectified Linear Unit) Katmanı .....	20
3.3.6. Sınıflandırma Katmanı .....	22
3.4. AlexNet Modeli.....	22
3.5. GoogLeNet Modeli .....	23
3.6. Caffe.....	25
3.7. Nvidia Digits .....	25
3.8. Hiper Parametreler .....	25
3.8.1 Öğrenme Oranı (Learning Rate) .....	25
3.8.2 Devir Sayısı (Epoch).....	26
3.8.3 Mini-Batch Boyutu .....	26
3.8.4 Aktivasyon Fonksiyonları .....	26
3.9. Başarım Ölçüm ve Değerlendirme.....	28
BÖLÜM 4.....	31
DENEYLER VE SONUÇLAR.....	31
4.1. Veri Seti.....	31
4.2. Veri Seti Eğitim ve Test İşlemleri .....	36
4.3. Eğitim Süresi.....	37
4.4. Değerlendirme Ölçütleri ve Değerlendirmeleri .....	38
BÖLÜM 5 .....	54
TARTIŞMA VE SONUÇ .....	54
KAYNAKLAR.....	57
EKLER.....	66

ÖZGEÇMİŞ .....	107
----------------	-----

## KISALTMALAR DİZİNİ

<b>DN</b>	: Doğru Negatif
<b>DP</b>	: Doğru Pozitif
<b>ESA</b>	: Evrişimsel Sinir Ağları (Convolutional Neural Network)
<b>GPU</b>	: Grafik İşlem Birimi (Graphics Processing Unit)
<b>ReLU</b>	: Doğrultulmuş Lineer Birim (Rectified Linear Unit)
<b>YN</b>	: Yanlış Negatif
<b>YP</b>	: Yanlış Pozitif
<b>YSA</b>	: Yapay Sinir Ağı (Artificial Neural Networks)

## ŞEKİLLER DİZİNİ

Şekil 3.1. Makine öğrenmesi süreci .....	10
Şekil 3.2. Derin öğrenme hata oranları. ....	11
Şekil 3.3. Derin öğrenmenin yerini açıklayan venn şeması. ....	12
Şekil 3.4. Derin öğrenme mimarisi. ....	13
Şekil 3.5. Yapay sinir ağı örneği. ....	14
Şekil 3.6. Evrişimsel sinir ağının mimarisi. ....	15
Şekil 3.7. Konvolüsyon işlemi örneği. ....	17
Şekil 3.8. Maksimum havuzlama işlemi. ....	18
Şekil 3.9. Ortalama havuzlama işlemi.....	18
Şekil 3.10. Tam bağlantılı katmanının uygulanması. ....	19
Şekil 3.11. Drop out katmanının uygulanması. ....	20
Şekil 3.12. ReLu grafiği. ....	21
Şekil 3.13. ReLu işlemi örneği.....	21
Şekil 3.14. AlexNet mimarisi. ....	22
Şekil 3.15. GoogLeNet mimarisi. ....	23
Şekil 3.16. Başlangıç modülü. ....	24

Şekil 3.17. Boyut azaltmalı başlangıç modülü.....	24
Şekil 3.18. Step fonksiyonu grafiği.....	27
Şekil 3.19. Sigmoid fonksiyonu grafiği. ....	27
Şekil 3.20. Softmax fonksiyonu grafiği. ....	28
Şekil 3.21. Tanh fonksiyonu grafiği. ....	28
Şekil 4.1. Veri seti 1 için eğitim veri sayısı. ....	34
Şekil 4.2. Veri seti 2 için eğitim veri sayısı. ....	34
Şekil 4.3. Veri seti 3 için eğitim veri sayısı. ....	35
Şekil 4.4. Versiyon bilgileri.....	36
Şekil 4.5. Epoch değer grafiği.....	38
Şekil 4.6. Veri seti 1 model sonucu. ....	39
Şekil 4.7. Test çalışması örnek sonuçları. ....	40
Şekil 4.8. Normal kategorideki röntgenin test sonucu. ....	42
Şekil 4.9. Zatürre hastasının röntgeninin test sonucu. ....	42
Şekil 4.10. Veri seti 2 AlexNet modeli sonucu.....	43
Şekil 4.11. Virüs kaynaklı zatürre hastasının röntgeninin test sonucu.....	46
Şekil 4.12. Bakteri kaynaklı zatürre hastasının röntgenin test sonucu.....	46
Şekil 4.13. Veri seti 3 AlexNet model sonucu grafiği. ....	47
Şekil 4.14. Normal kategorisindeki röntgenin test sonucu. ....	48

Şekil 4.15. Zatürre hastasının röntgeninin test sonucu. ....	49
Şekil 4.16. Covid19 hastasının röntgenin test sonucu. ....	49
Şekil 4.17. Veri seti 3 GoogLeNet model sonuç grafiği. ....	50
Şekil 4.18. Normal kategorisindeki röntgenin test sonucu. ....	51
Şekil 4.19. Zatürre hastasının röntgeninin test sonucu. ....	52
Şekil 4.20. Covid19 hastasının röntgeninin test sonucu. ....	52

## ÇİZELGELER DİZİNİ

Çizelge 3.1. Karışıklık matrisi. ....	29
Çizelge 4.1. Veri seti 1 içerisindeki veri sayıları. ....	32
Çizelge 4.2. Veri seti 2 içerisindeki veri sayıları. ....	33
Çizelge 4.3. Veri seti 1 içerisindeki veri sayıları. ....	33
Çizelge 4.4. Veri setleri karşılaştırması. ....	35
Çizelge 4.5. Eğitim sürelerinin karşılaştırılması. ....	38
Çizelge 4.6. Veri seti 1'in karışıklık matrisi içindeki değerler. ....	41
Çizelge 4.7. Veri seti 1 için değerlendirme ölçütleri. ....	41
Çizelge 4.8. Veri seti 2'nin bakteri için karışıklık matrisi içindeki değerleri. ....	44
Çizelge 4.9. Veri seti 2'nin virüs için karışıklık matrisi içindeki değerleri. ....	45
Çizelge 4.10. Veri seti 2 bakteri için değerlendirme ölçütleri. ....	45
Çizelge 4.11. Veri seti 2 virüs için değerlendirme ölçütleri. ....	45
Çizelge 4.12. Veri seti 3 AlexNet modeli test sonuçları. ....	47
Çizelge 4.13. Veri seti 3 AlexNet modeli için değerlendirme ölçütleri. ....	48
Çizelge 4.14. Veri seti 3 GoogLeNet modeli test sonuçları. ....	50
Çizelge 4.15. Veri seti 3 GoogLeNet modeli için değerlendirme ölçütleri. ....	51



Çizelge 4.16. Veri seti 3 için değerlendirme karşılaştırması. ....	53
--	----

# BÖLÜM 1

## GİRİŞ

Derin öğrenme, makine öğrenmesinin bir alt dalıdır ve derin öğrenme ile makine öğrenmesi kıyaslandığında derin öğrenmenin çok katmanlı yapıya sahip olduğu görünmektedir. Çok katmanlı yapısıyla çalışma şekli olarak insan beyninin çalışma mantığını temel almıştır (Koitka & Friedrich, 2016). Derin öğrenme modellerinde karar verme sürecinde insan beyni gibi kavram hiyerarşisi sisteminden yararlanmaktadır (Goodfellow, Bengio & Courville, 2016). Bu özelliğinden dolayı derin öğrenme en büyük avantajlarından biri olan öğrenme aşamasında kullanacağı özellik veya özniteliklerin çıkarımlarını kendi kendine yapabilmektedir. Derin öğrenme modellerinde özellik ve özniteliklerin kurallar halinde önceden verilmemekte ve eğitim aşamasında çıkarımları yapmaktadır (Patterson & Gibson, 2017).

Derin öğrenmenin temelinde birçok yapay sinir ağları mevcuttur. Bu sebepten dolayı derin öğrenme modelleri çok katmandan oluşmaktadır. Çok katmanlı yapıdan oluşan derin öğrenme modelinde kullanılan katmanlar modelin eğitimi aşamasında öğrenme işlemi gerçekleştirmek amacıyla kullanılmaktadır (Korkmaz, 2019).

Yapay zekâ, insan beyninin çalışma mantığını temel alarak bilgisayarlara ve makinalara belirli algoritmalar aracılığıyla problemlerin çözümünü sağlamaktadır (Hashimoto, Rosman, Rus & Meireles, 2018). Yapay zekaya giderek artan ilgi derin öğrenme mimarilerinin ortaya çıkmasına sebep olmuş ve derin öğrenme en yaygın olarak kullanılan yapay zekâ algoritmaları arasında yerini almıştır. Derin öğrenme yaklaşımlarıyla birlikte endüstri, tıp, görüntü işleme, nesne tespiti, robotik, finans ve tahmin yürütme gibi birçok

farklı alanda ve disiplinde oluşan problemlerin ve sorunların çözümü üzerinde çalışmalar yapılmıştır (Doğan & Türkoğlu, 2019).

Özellikle yapay zekaya sağlık alanında olan ilgi her geçen gün hızla artmaktadır. Yapay zekanın özellikle de derin öğrenmenin görüntü işleme üzerindeki başarısı nedeniyle tıbbi görüntüleri işleme ve yorumlama aşamalarında tercih edilmektedir (Atlan & Pençe, 2021).

Tıp kendi içerisinde birçok farklı disiplinleri barındıran bir bilimdir. Tıp bilimine bakıldığında hasta psikolojisi, mahremiyeti gibi etik ve insani konularda otomatik bir karar sisteminden bahsetmek tam olarak pek mümkün değildir. Bu sebepten dolayı tıp biliminde de bazı konular otomatikleştirilemez. Tıbbi görüntüleme uygulamalarında yapay zekanın kullanım amacı doktor ve sağlık personellerinin işlerini otomatik olarak yapmak değil onlara yardımcı olarak geliştirilmektedir. Bu uygulamalarda amaç hastalığın teşhisinde kısa zamanda doğruluk oranı yüksek sonuçlar elde etmektir (Ranschaert, Morozov & Algra, 2019).

Hızlı veri depolaması, hesaplama işlemleri üzerinde güce sahip olması ve bununla birlikte veri girişlerinden yüksek seviyeli özellikler ve anlamsal yorumlama özelliğine sahip olan derin öğrenme yapay zekanın geleceğini şekillendirmeye olanak sağlayacak güçte olarak değerlendirilir (Ravi vd., 2016).

Enfeksiyon hastalıkları, insanların yaşamında tehdit oluşturabilecek ölçüde önemlidir. Enfeksiyon hastalıklar listesinde ilk sırada olan enfeksiyon hastalığı ise pnömonidir. Pnömoninin halk dilindeki karşılığı zatürre hastalığıdır (Akter, Shamsuzzaman & Jahan, 2015). Zatürre hastalığının teşhisinde en çok tercih edilen yöntem göğüs röntgen görüntüleridir. Göğüs röntgen görüntüleri ile akciğerlerde zatürre olup olmadığı tespit edilebilmekte ve tanı konulabilmektedir (Ayan & Ünver, 2019).

Derin öğrenme yöntemleri ile insanların kendi yaptığı sonuçlar kıyaslandığında, derin öğrenmenin nesne tanıma, bölütleme ve algılama vb. işlemlerin sonucunda daha iyi sonuçlar görülmektedir. Tıbbi görüntüleme alanında da derin öğrenme sayesinde algılama ve bölütlemede performans artışları olduğu görülmektedir (Wang vd., 2017).

Derin öğrenme teknikleriyle yapılan tıbbi görüntüleme işlemleri bu alanda yapılan çalışmalar arasında yer almaktadır. Tıbbi görüntülerden hastalık teşhisi yapabilmek amacıyla veya tıbbi görüntülerin kalitesini arttırmak için derin öğrenme tekniklerinde yararlanılmaktadır. Tıbbi görüntüleme alanında derin öğrenme teknikleriyle yapılmış olan çalışmalara görüntülerden doku ve hücre sınıflandırılması, görüntülerdeki olan gürültünün azaltılması, vücuttan alınmış olan görünün içerisinde iç kanama olup olmadığının tespit edilmesi, organ tespiti yapılması, vücutta oluşan lezyonların tespit edilmesi, görüntülerin çözünürlüğünün artırılması ve önemli hastalıklardan olan Alzheimer, zatürre, kanser gibi hastalıkların teşhisi örnek olarak verilebilmektedir. Derin öğrenmenin tıbbi görüntülerde kullanılmasının önemli nedenlerinde biri insan kaynaklı hataları önleyerek tespiti önemli olan hastalık türlerinde yanlış tespit konulmaması amaçlayıp bu sayede ölüm oranını düşürmektir (Gülgün, 2020).

Derin öğrenmenin yaygınlaşmasının nedenlerinden biri de hayatımızın giderek dijitalleşmesidir. Dijitalleşmeyle birlikte veri kümelerinde artış söz konusu olmuştur. Bunu artış ile büyük veri kavramı hayatımıza girmiştir. Büyük veri kavramı ile makine öğrenmesinin kullanımı artmış ve daha fazla yaygınlaşmıştır. Derin öğrenmenin yaygınlaşmasının bir diğer nedeni de günümüzde büyük modellerle çalışma ve hesaplama imkânı sunabilen makinelere sahip olabilmemizdir (İnik & Ülker, 2017).

## BÖLÜM 2

### LİTERATÜR ARAŞTIRMASI

Zou ve arkadaşlarının 2015 yılındaki yapmış oldukları çalışmalarında kapsül endoskopi görüntüleri üzerinde sindirim organının sınıflandırılmasını incelemişler ve bu konuda çalışmalarını gerçekleştirmişlerdir. Yapmış oldukları bu çalışmada mide, ince bağırsak ve kolon görüntüleri ile çalışmıştır. Eğitim aşaması 20 bin mide, 20 bin ince bağırsak ve 20 bin kolon görüntülerinden oluşurken test aşaması her bir sınıfın rastgele seçilmiş 5 bin görüntülerinden oluşmaktadır. Çalışmanın sonunda destek vektör sistemleri ile doğruluk oranını %90,31 elde edilirken CNN ile doğruluk oranı %95,52 olarak elde edilmiştir (Zou, Li, Wang, Yu & Li, 2015).

Saxena 2015 yılında yapmış oldukları çalışmada evrimsel sinir ağını kullanarak çekim masasını görüntüden kaldırmayı amaçlamışlardır. Yapıkları bu çalışmada test aşamasında 500 adet görüntü kullanmışlardır ve bu test sonucunda %99 başarı oranı elde etmişlerdir (Saxena, 2015).

Anthimopoulos ve arkadaşları 2016 yılında akciğer hastalıklarında farklı doku türlerinin sınıflandırılmasında CNN modelini kullanmış ve bu modeli önermişlerdir. Bu çalışmada doğruluk oranı %85,61 olarak gösterilmiştir (Anthimopoulos, Christodoulidis, Ebner, Christe & Mougiakakou, 2016).

Suzyli ve arkadaşları 2016 yılında yapmış oldukları çalışmada derin öğrenme yöntemlerinden olan Derin Konvolüsyon Sinir Ağları kullanılmıştır. Çalışmalarında 99 kitleye sahip olan ve 99 kitleye sahip olmayan yani sağlıklı mamografi görüntülerinden

oluşmaktadır. Yaptıkları çalışma sonucunda kitle sınıflandırmada doğruluk oranı %89,90 olarak tespit edildiğini göstermişlerdir (Suzuki, Zhang, Homma, Ichiji & Sugita, 2016).

Sarraf ve Tofighi 2016 yılında yapmış oldukları çalışmalarında Alzheimer hastalığına sahip olan beyin ve sağlıklı beyin görüntülerini ayırt etmek için CNN kullanmışlardır. Veri seti 28 Alzheimer hastası ve 15 sağlıklı beyin görüntülerinden oluşmaktadır. Veriler eğitim için %60, doğrulama için %20 ve test için %20 oranına ayrılarak kullanılmıştır. Çalışmanın sonunda %98,85 doğruluk oranı elde ettiklerini ifade etmişlerdir (Sarraf & Tofighi, 2016).

Dimililer ve Ilhan 2016 yılında yapmış oldukları çalışmadan yapay sinir ağları tabanlı bir sistem ile önerilmiştir. Bu çalışmadaki amaçları tümörlü ve tümörsüz olarak beyin MR görüntülerini sınıflandırabilmektedir. Beyin MR görüntülerinin ilk boyutları 256x256 iken daha sonra 64x64 boyutuna indirilmiştir. Buradaki amaç eğitim süresinin daha az sürmesi içindir. Beyin MR görüntüleri 64x64 olarak boyutlandırıldıktan sonra histogram eşitleme yöntemleri yapılarak ön işlemlerden geçirilmiştir. Çalışmada üç katman kullanılmıştır. Bu katmanlar giriş katmanı, gizli katman ve çıkış katmanlarından oluşmaktadır. Ön işlemden geçmiş olan beyin MR görüntülerine ardından etiketleme işlemi yapılmıştır. Etiketlemede tümöre sahip olan beyin MR görüntüleri [1,0] iken tümöre sahip olmayan beyin MR görüntüleri [0,1] olarak etiketlenmiştir. Bu çalışmanın sonunda sınıflandırma başarısı %90 olarak belirtilmiştir (Dimililer & Ilhan, 2016).

Ciampi ve arkadaşları 2017 yılında yapmış oldukları çalışmada akciğer kanseri bulunan nodüllerin tespiti için CNN modelini kullanmışlardır. Kullandıklarını veri seti için 2 farklı veri setlerinden oluştuğu belirtilmektedir. Çalışma sonucunda doğruluk oranı %79,5 olarak elde edilmiştir (Ciampi vd., 2017).

Zatürre hastalığının teşhis edilmesiyle ilgili daha önceden yapılmış olan çalışmalara mevcuttur. Bunlardan biri Wang'a ait olan çalışmadır. Wang çalışmasını önce RESNet-50 üzerinde geliştirmiş ardından önceden eğitilmiş olan modellerden AlexNet, GoogleNet, VGGNet-16 ve ResNet kullanmış ve elde ettiği sonuçları karşılaştırmıştır. AlexNet 0.54, GoogLeNet 0.59, VGGNet-16 0.51 ve ResNet-50 0.63 başarı oranlarını yaklaştırmıştır (Wang vd., 2017).

Ma ve arkadaşların 2017 yılında baş ve boyun kanserinin tespit edilmesi için derin öğrenme yönetimini önermişlerdir. Bu çalışmada derin öğrenme algoritmasının el ile belirtilerek öğrenmesine kıyasla daha ayırt edici olduğunu belirtmişlerdir. Bu çalışmada her bir piksel için tümörlü ve normal dokuyu sınıflandırmada evrişimli sinir ağları kullanmışlardır. Yapmış oldukları çalışmanın sonucunda %91,36 başarı oranı elde etmişlerdir (Ma vd., 2017).

İnik ve Ülker'in yapmış 2017 yılında derin öğrenme ve görüntü analizinde kullanılan derin öğrenme modelleri üzerine çalışmasında derin öğrenme tarihsel olarak incelenmiştir. Derin öğrenme mimarisi olan evrişimsel sinir ağı mimarisinin yapısı ve mimari içerisinde kullanılan katmanlar hakkında bilgi verip katmanların modelle olan etkileri gösterilmiştir. Bu çalışmalarında derin öğrenmenin popüler olmasını AlexNet modeli sağladığını belirtmektedir (İnik & Ülker, 2017).

Kumar ve arkadaşlarının 2017 yılında yapmış oldukları çalışması beyin MR görüntülerinde mevcut olan tümörlerin iyi huylu veya kötü huylu olup olmadığının ayrımın yapılmasıyla ilgilidir. Çalışmalarında ilk olarak beyin MR görüntülerinin uzantısını .mha'tan .jpg uzantılı formatta geçirmekle başlamaktadır. Ardından belirledikleri değerleri hesaplayarak öznitelik vektörünü elde etmişlerdir. Öznitelik vektörünün üzerinde genetik algoritma kullanarak çalışma yaptıktan sonra DVM kullanarak iyi huylu ve kötü huylu tümörleri sınıflandırmışlardır. İyi huylu 20 ve kötü huylu 5 beyin MR görüntülerinde sınıflandırmada doğruluk başarı oranını ortalama %89,56 olarak ifade etmişlerdir (Kumar, Dabas & Godara, 2017).

Shin ve Balasingham (2017) yapmış oldukları çalışma da görüntüler üzerinden renk ve şekil özelliklerini alarak Destek Vektör Makineleri ve CNN yöntemlerini kullanıp bu yöntemleri karşılaştırmışlardır. Veri setindeki benzer görüntüleri içerdiği için CVC-Clinic, ETIS-Larib ve Asu-Maya veri setlerini kullanarak çalışmalarını gerçekleştirmişlerdir. Bu veri setlerinde 428 adet görüntüyü test amaçlı kullanılmıştır. Shin ve Balasingham'ın yapmış oldukları çalışmanın sonunda destek vektör makineleri ile doğruluk oranı %84 iken CNN ile doğruluk oranı %91 olarak elde edilmiştir (Shin & Balasingham, 2017).

Deniz ve arkadaşları 2018 yılında daha önceden eğitilmiş olan AlexNet ve VGG-16 modellerine uyum sağlayan aktarım öğrenimleri ve derin özellik çıkarımı yöntemlerini değerlendirmişlerdir. Bu çalışma sonucunda görüntü düzeylerinde %90,5 ile %91,4 arasında bir doğruluk oranı elde etmişlerdir (Deniz vd., 2018).

Arı ve Hanbay beyin tümörünü tespit etmek için çalışma yapmışlardır. Bu çalışmada manyetik rezonans görüntüleri önceden bazı ön işlemlere sokulmuştur. Ardından evrimsel sinir ağına alternatif bir yöntem olarak tasarladıkları yöntem ile doğruluk oranında %97,18 başarı oranını elde etmişlerdir (Ari & Hanbay, 2018).

Siar ve Teshnehab (2019) yapmış oldukları çalışmada beyin tümörünü manyetik rezonans görüntüleme ile elde ettikleri görüntüleri kullanarak, evrimsel sinir ağı yöntemiyle tespit etmeye çalışmışlardır. Çalışmalarında toplamda 1666 görüntü eğitim amaçlı ve 226 görüntü test verisi kullanmışlardır. Beyin tümörünü tespit etmek amacıyla yaptıkları bu çalışmada CNN modelini kullanarak doğruluk oranı %98,67 başarıyı elde etmişlerdir (Siar & Teshnehab, 2019).

Ayan ve arkadaşları çalışmalarında zatürre hastalığının erken teşhisinin üzerinde çalışma yapmış ve erken teşhisinin önemini vurgulamışlardır. VGG-16 modelini ve Xception modelini kullanmış ve doğruluk oranlarını tespit etmişlerdir. Test aşamasında 624 adet göğüs kafesi görüntüleri ile VGG-16 modelinde %87, Xception modelinde %82 doğruluk oranına ulaşılmıştır (Ayan & Ünver, 2019).

Stephen ve arkadaşları zatürre teşhisinde kullanılan çalışmaların aksine göğüs kafesi görüntülerinden oluşan veri kümesini önceden oluşan değil sıfırdan yaratılan bir evrimsel sinir ağlarında eğitmişlerdir. Veri kümesinin veri artırma yöntemleri ile arttırmışlardır. Kullandıkları evrimsel sinir ağları modelinde 3x3'lük farklı sayıdaki filtrelerle 4 evrim katmanı ve her evrim katmanının ardından ReLu aktivasyon fonksiyonu ve maksimum havuzlama katmanı bulunmaktadır. İki tam bağlantı katmanı bu katmanları takip etmektedir. Tam bağlantı katmanında sigmoid fonksiyonu kullanılmıştır. Bu çalışmada zatürre veya normal olup olmadığının tespiti yapılmaya çalışılmıştır. Başarı oranı %93,73 oranı elde etmişlerdir (Stephen, Sain, Maduh ve Jeong, 2019).



Korkmaz 2019 yılında derin sinir ađlarını kullanarak alıřma yapmıř ve bu alıřmada kk ila moleklleriyle sınıflandırma iřlemi gerekleřtirerek sonuları gstermiřtir. Derin sinir ađlarının klasik makine đrenmesi yntemlerinden farklı olarak veri boyutu yksek olan verilerde alıřmanın performans aısından daha fazla avantajlı olduđunu belirtirmiř, derin sinir ađlarının avantajları zerinde durmuřtur. alıřmanın sonunda en iyi sonucu veren modellen dođruluk oranı %89 olarak elde etmiřtir (Korkmaz, 2019).

## BÖLÜM 3

### MATERYAL METOD

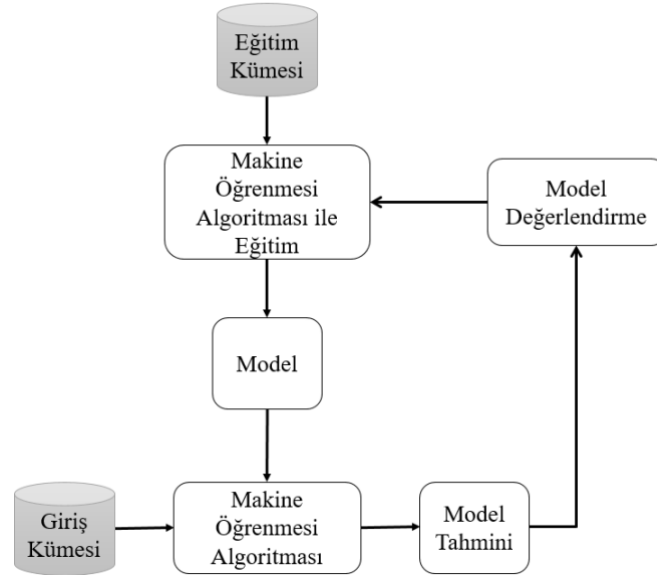
#### 3.1. Yapay Zekâ

Yapay zekâ insanların karar verme mekanizmasından yola çıkarak insan beynini model olarak alıp insan gibi düşünmesine olanak sağlayan ve akıllı kararlar almaya sağlayan teknolojidir. Yapay zekâ teknolojilerine yapay sinir ağları, makine öğrenmesi, derin öğrenme, bulanık mantık, genetik algoritmalar örnek olarak verilebilmektedir. Genel olarak kümeleme, tahmin etme, sınıflandırma işlemi yapmasının yanında bir takım farklı alanlarda da yapay zekâ algoritmaları kullanılmaktadır. Sadece insan beynini model olarak almayıp doğayı da model olarak alan yapay zekâ algoritmaları mevcuttur. Yapay zekâ optimizasyonu olarak kullanılmakta olan karınca kolonisi algoritması buna örnek olarak verilebilir (Balaban & Kartal, 2015).

#### 3.1. Makine Öğrenmesi

Yapay zekanın bir alt dalı olan makine öğrenmesinin temel amacı veriden bilgi elde etmektir. Veriden bilgi üretirken hesaplama işlemleri kullanılmaktadır. Doğal dil işleme, el yazısı tanıma, robotik çalışma alanlarında, ses ve konuşma tanıma işlemleri gibi birçok farklı alan ve farklı disiplinlerde makine öğrenmesi kullanılmaktadır (Peng, 2015). Makine öğrenmesi yöntemlerine rastgele orman, k-en yakın komşu, lojistik regresyon, karar ağacı, lineer regresyon ve basit bayes örnek olarak verilebilir. Rastgele orman yöntemi, denetimli makine öğrenmesi algoritması olup karar ağaçlarından oluşmaktadır. K-en yakın komşu yönteminde parametre mevcut değildir. Bu yöntemde sınıflandırma yaparken en yakın komşu sayısı kullanılmaktadır. Lojistik regresyon temel lojistik fonksiyonunu kullanmaktadır.

Burada bağımlı veya en az bir bağımsız değişkenlerin birbiri ile olan ilişkilerini tahmin etmek amacıyla kullanılmaktadır. Karar ağaçları, özelliklerin düğümlerle kararların ise dallar ile ifade edildiği bir yapıdır. Sınıflandırma yapılacak olan özelliği düğümler temsil ederken, dallar ise düğümlerin kararlarını veya değerleri göstermektedir. Lineer regresyon yöntemi, tahmin değerlerini modellemek amacıyla kullanılmakta olan bir makine öğrenmesi yöntemidir. Bu yöntemde bağımsız değişkenler kullanılmaktadır. Lineer regresyon yönetiminin en yaygın kullanımı içerdiği değişken ve tahminler arasındaki ilişkiyi göstermektedir. Basit Bayes bir denetimli makine öğrenmesi yöntemi olup Bayes teoremini kullanmaktadır (Mahesh, 2020). Şekil 3.1'de makine öğrenmesinin temel işleyişi gösterilmektedir (Gültepe, 2019).



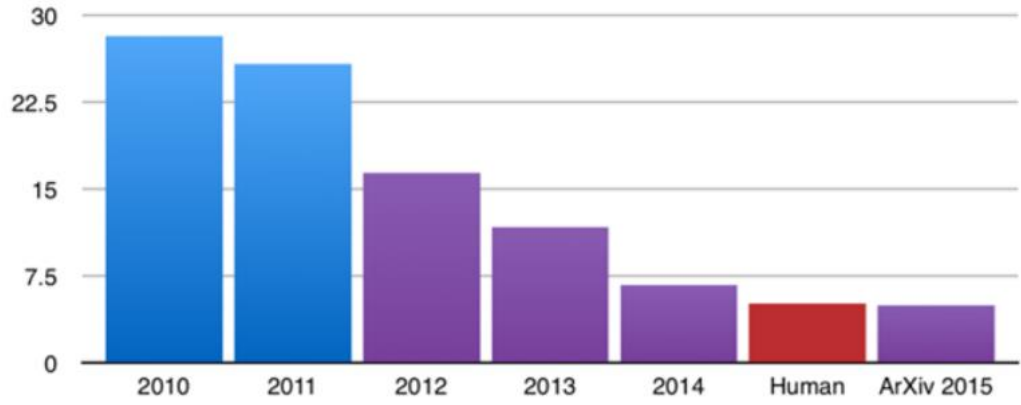
Şekil 3.1. Makine öğrenmesi süreci.

### 3.1. Derin Öğrenme

Bilim insanları tarafından derin öğrenme 2012 yılında büyük ilgi görmeye başlanmıştır. Derin öğrenmenin hayatımıza girmesiyle birlikte birçok farklı alanda ve disiplinde uygulamalar geliştirilmiştir. Derin öğrenme görüntü tanıma, nesne tanıma, ses tanıma, kanser hastalığının tanısını koyma gibi birçok farklı alanlarda karşımıza çıkmaktadır. Derin öğrenmenin bu alanlarda göstermiş olduğu problemlerin çözümündeki yüksek doğruluk oranları sayesinde derin öğrenme daha fazla yaygınlaşmış ve kullanımı artmıştır

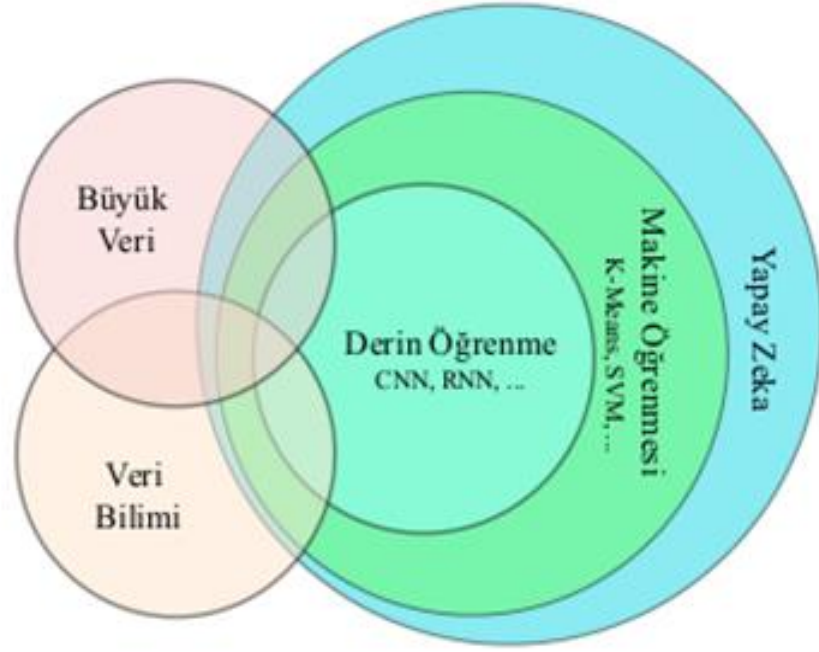
(Özcan, 2014). Derin öğrenmede, bilgisayara manuel olarak biçimsel ve matematiksel kurallar verilmez, her kavram temel kavramlara ilişkilendirilmektedir. Bu ilişkilendirme sonrasında kavram hiyerarşisi oluşmaktadır. Bu kavram hiyerarşisi öğrenmeyi sağlamaktadır. Kavram hiyerarşisi sayesinde basit kavramlardan faydalanılarak temel karmaşık kavramlar oluşturulmaktadır (Goodfellow, 2016).

Derin öğrenmenin ImageNet yarışmasında göstermiş olduğu hata oranları Şekil 3.2’de gösterilmektedir (İnik & Ülker, 2017).



Şekil 3.2. Derin öğrenme hata oranları.

Derin öğrenme makine öğrenmesinin bir alt sınıfıdır (Leng & Yu, 2014). Derin öğrenmenin yapay zekâ ve makine öğrenmesi arasındaki ilişkiyi göstermekte olan venn şeması Şekil 3.3’teki gibidir (Yurtsever, 2019).



Şekil 3.3. Derin öğrenmenin yerini açıklayan venn şeması.

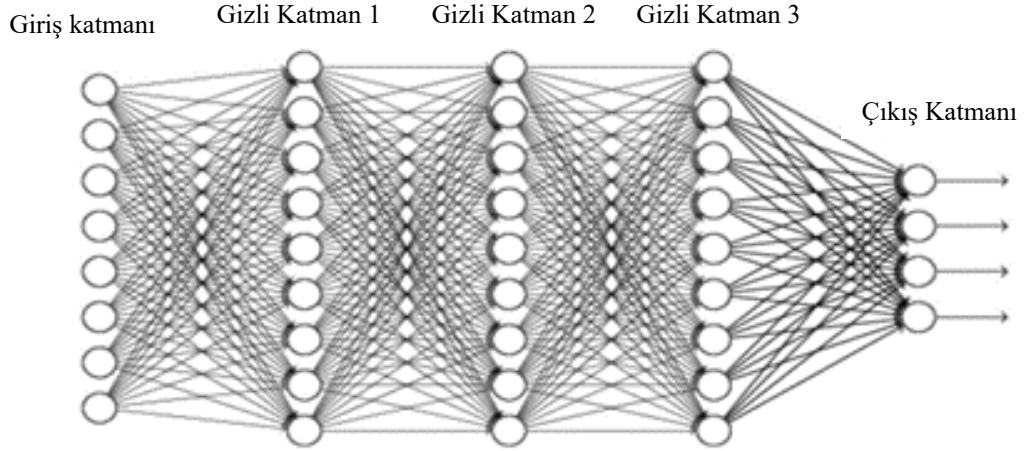
Şekil 3.3'te gösterildiği gibi yapay zekâ, makine öğrenmesini kapsarken makine öğrenmesi, derin öğrenmeyi kapsamaktadır. Buna ek olarak büyük veri ve veri biliminin kesişim kümeleri mevcut iken bu kümelerle yapay zekâ, makine öğrenmesi ve derin öğreneninde kesişimler kümeleri mevcuttur. Genel olarak yapay zekâ, makine öğrenmesi ve derin öğrenmeye bakıldığında en kapsamlı olarak yapay zekâ karşımıza çıkmaktadır (Yurtsever, 2019).

Veri işleme normal koşullar dahilinde verinin miktarına bağlı olarak çok zaman almaktadır. Bu yüzden derin öğrenmede miktarı fazla olan veriler ile çalışırken eğitim süresi kısa olamayabilir. Grafik İşlem Birimleri, (GPU) sayesinde bu eğitim sürelerini daha kısaltmak mümkündür. Bu yüzden derin öğrenmede GPU teknolojisi önemli rolle sahiptir (Shaikh, 2017).

Mevcut olan tek katmanlı görüntüler üzerinde çalışabilen sistemler ile çok katmanlı yapıya sahip olan derin öğrenme modelinin performansları kıyaslandığında derin öğrenme modelinin daha yüksek başarı oranına sahiptir. Derin öğrenmenin sisteme manuel olarak

verilmesi gereken özellikleri tek tek vermek yerine parametreleri kendi keşfederek kullanması en önemli özellikleri arasında yer almaktadır (Toğaçar & Ergen, 2019).

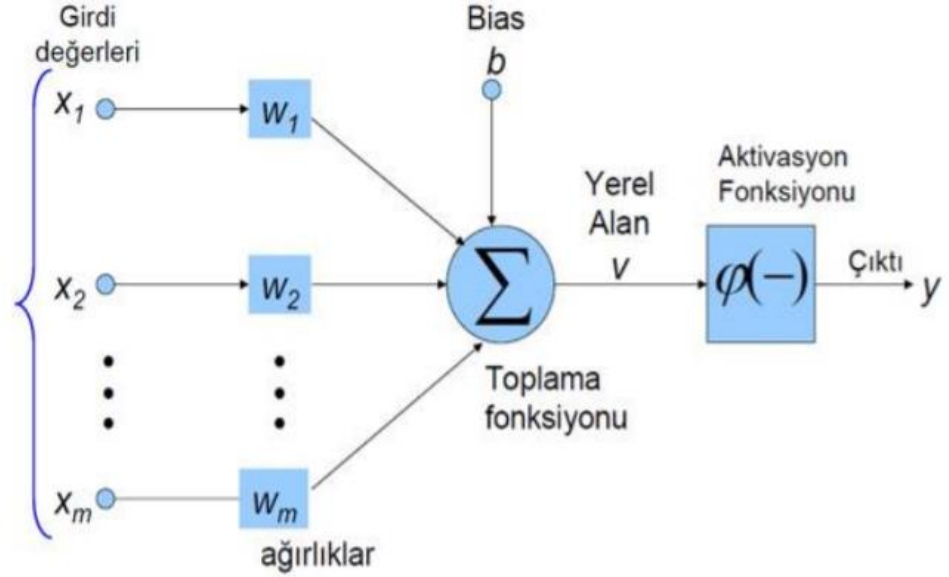
Veriden özellik çıkartma ve dönüştürmek için birden fazla doğrusal olmayan işlem birimi katmanını kullanmaktadır. Her katman ardışık olarak sıralanmıştır. Bu katmanlar önceki katmandaki çıktıyı girdi olarak almaktadır (Leng & Yu, 2014). Katmaların genel olarak yapısı üçe ayrılmaktadır. Bunlar girdi katmanı, gizli katman ve çıkış katmanıdır (Tüfekçi & Karpat, 2019). Derin öğrenme mimarisinin genel yapısı Şekil 3.4'te gösterilmektedir. Bu şekilde gizli katman olarak 3 adet gizli katman kullanılmıştır (Nielsen, 2015).



Şekil 3.4. Derin öğrenme mimarisini.

### 3.2. Yapay Sinir Ağları

Yapay sinir ağları ile birçok farklı alan ve farklı disiplinlerde geçmişten günümüze uygulamalar geliştirilerek problemlere çözüm aranmaktadır. Uygulama alanlarına organların modellenmesi, hastalık tanımlamaları, parmak izi tanıma, radar sinyallerinin sınıflandırılması, spam maillerinin filtrelenmesi, el yazısı tanımlama örnek olarak gösterilebilir. Yapay sinir ağı girdi değerlerinden, ağırlıklardan, toplama fonksiyonundan, aktivasyon fonksiyonundan ve çıktılardan meydana gelmektedir. Yapay sinir ağının örneği Şekil 3.5'te gösterilmektedir (Keskenler & Keskenler, 2017).



Şekil 3.5. Yapay sinir ağı örneği.

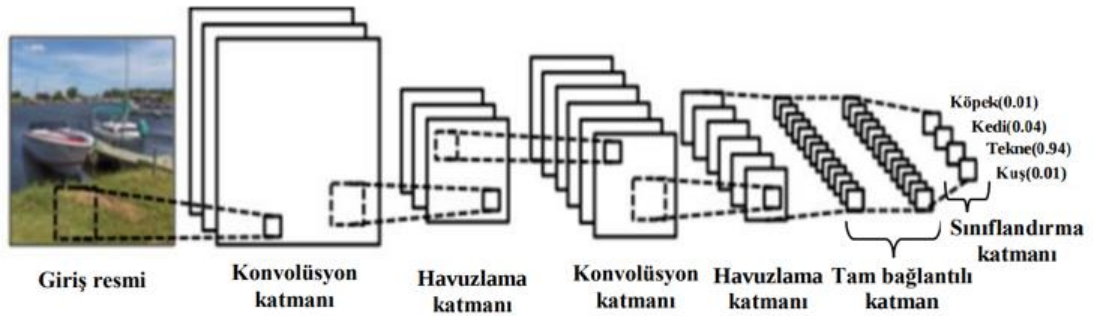
YSA çalışma mantığı olarak insan beyindeki nöronların birbirleri ile olan etkileşim hareketinin mantığını esas almaktadır (Chien, Ding & Wei., 2002). YSA model tanıma ve genelleme yapabilme yeteneğinden kaynaklı eğitilmiş olan ağa bilinmeyen girdi giriş olarak verildiğinde uygun olan çıktıyı verebilmektedir (Niebur & Germond, 1991).

### 3.3. Evrişimsel Sinir Ağları

Evrişimsel sinir ağları (ESA) çok katmanlı sinir ağlarından biridir. Görüntü tanıma, ses tanıma vb. birçok alanda ESA'dan yararlanılmaktadır. Özellikle görüntü tanıma alanında evrişimsel sinir ağları ile yapılmış çalışmalarda başarılı sonuçlar elde etmiştir (Tüfekçi & Karpat, 2019). Evrişimsel sinir ağları, günümüzde görüntü işlemede kullanılmasının yanı sıra doğal dil işleme alanında da sıkça kullanılmakta olan derin öğrenme modelidir (LeCun, Bottou, Bengio & Haffner, 1998).

Klasik yapay sinir ağları (YSA) ile ESA karşılaştırıldığında karşımıza çıkan en belirgin özellik, klasik sinir ağlarında vektörler kullanılırken ESA yapısında matrisler kullanılmaktadır. Matris kullanımının avantajı olarak komşu tabakalar arasında bilgi kaybolması engellenmektedir (Traore, Kamsu & Tangarab, 2018).

ESA modeli iki aşamadan oluşmaktadır. Birinci aşamada bağlantılı olan öznelikler belirlenir, ikinci aşamada ise sınıflandırma yapılmaktadır. Sınıflandırma yapılırken çok katmanlı sinir ağlarından yararlanılmaktadır. ESA modelinin ilk adımı olan öznelik haritası çıkartma işlemi filtrelerin sinir ağındaki giriş verisi içerisinde konvolüsyon uygulanmasıyla elde edilmektedir. ESA modelinde geri yayılım algoritması kullanılmaktadır. Geri yayılım algoritması kullanılarak modelin eğitimi boyunca hata durumunda filtrelerin parametreleri güncellenmektedir. Güncellenen parametreler ile model öğrenmeye devam etmektedir böylece filtreleri içerikleri öğrenilmiş olmaktadır. Bu işlemlerin ardından çıkarılmış olan öznelik haritaları havuzlama işlemi yapılmaktadır. Burada havuzlama işlemi yapılmasındaki temel amaç birinci aşamada çıkarılan özelliklerin çeşitliliğini ve boyutunu azaltmaktır. Ardından matris şeklinde olan haritalar vektör haline getirilerek sinir ağına girdi olarak verilir (Işık & Artuner, 2018). Evrişimsel sinir ağının genel mimari yapısı Şekil 3.6’da gösterilmektedir. Burada giriş resminin sırasıyla konvolüsyon katmanı, havuzlama katmanı, konvolüsyon katmanı, havuzlama katmanı, tam bağlantılı katman ve sınıflandırma katmanına girdiği görülmektedir (İnik & Ülker, 2017).



Şekil 3.6. Evrişimsel sinir ağının mimarisi.

### 3.3.1. Giriş Katmanı

Evrişimsel sinir ağı mimarisinin ilk katmanı giriş katmanıdır. Bu katmanda evrişimsel sinir ağının mimarisine giriş yapacak olan ham verinin ağa katılması sağlamaktadır. Ağa katılacak olan görüntünün boyutu önem taşımaktadır. Boyutun artışın performans ile doğru orantı gösterirken zaman açısından ters orantı göstermektedir. Bu açıdan giriş katmanı ağın



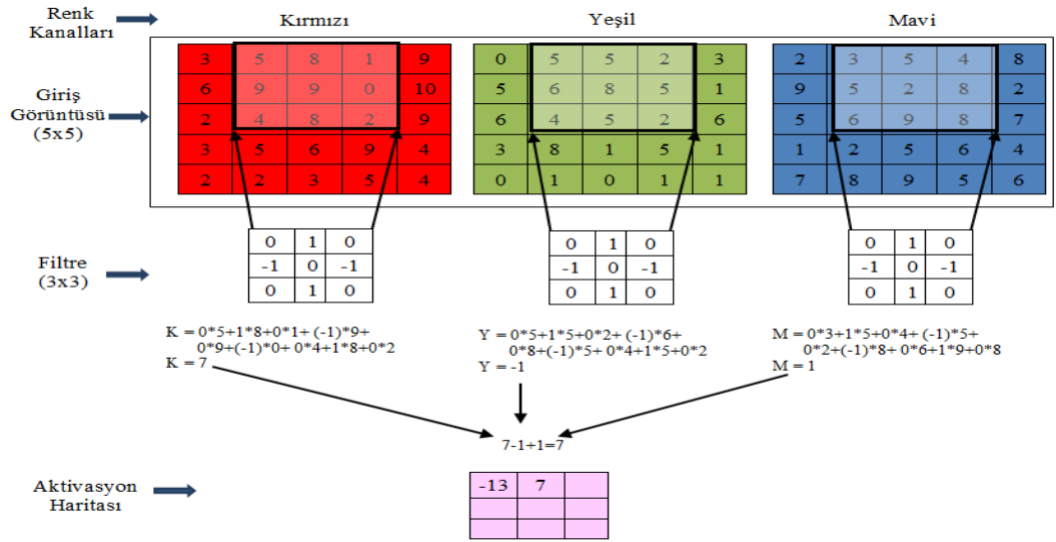
performans ve zaman deęerlendirmelerinde giriř katmanı önemli rol oynamaktadır (İnik & Ülker, 2017).

### **3.3.2. Evriřim Katmanı**

Evriřim katmanı genel olarak evriřimsel sinir aęı mimarisinin temelini oluřturan katmandır ve evriřim katman aynı zamanda dönüşüm katmanı olarak da adlandırılmaktadır (Doęan & Türkoęlu, 2019). Evriřim katmanında girdi görüntülerinden özellikler çıkarılmaktadır. Evriřim, görüntü üzerindeki girdi küçük kare içerisindeki verileri kullanarak görüntünün özellikleri öğrenmekte bu sayede piksel arasındaki uzamsal ilişkileri bozulmasına engel olunmaktadır. Her görüntünün piksel deęerleri matris olarak deęerlendirilmektedir ve piksel deęerleri 0 ve 1 deęerini almaktadır (Amir, 2019).

Görüntünün konvolse edilmesinden sonra oluřan halindeki filtreler ile görüntü aynı sayıda boyutta sahip olmaktadır. Boyutların hepsinin derinlikleri aynı olmasına raęmen genişlik ve yükseklik deęerleri daha küçüktür. Uzamsal boyutlar genellikle 3x3 ya da 5x5 olarak bilinmektedir. Uygulanmış olan filtre adedi ile çıktı derinlięi aynı deęerlere sahiptir (Goodfellow vd., 2016).

Giriř görüntüsü 5x5x3 olan bir görüntüye 3x3'lük filtre ile konvolüsyon işlemini yapıldığını Şekil 3.7'de gösterilmektedir.

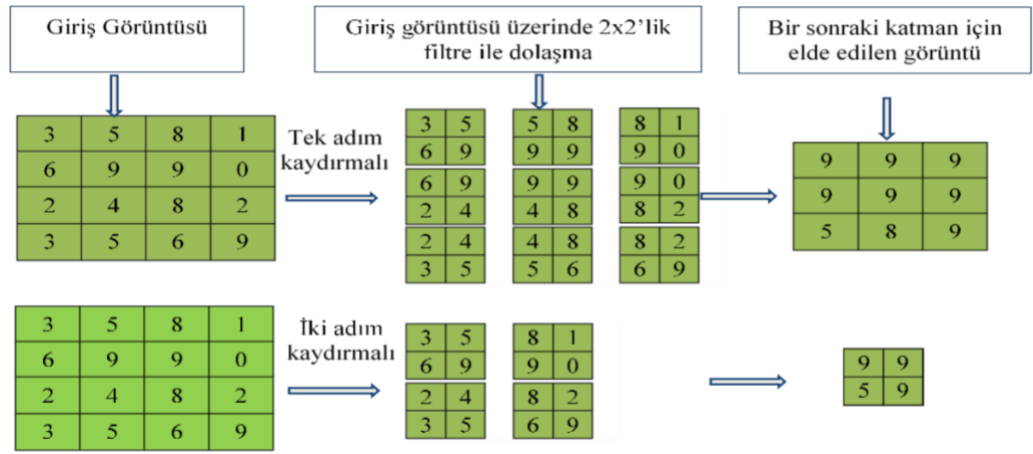


Şekil 3.7. Konvolüsyon işlemi örneği.

### 3.3.3. Havuzlama Katmanı

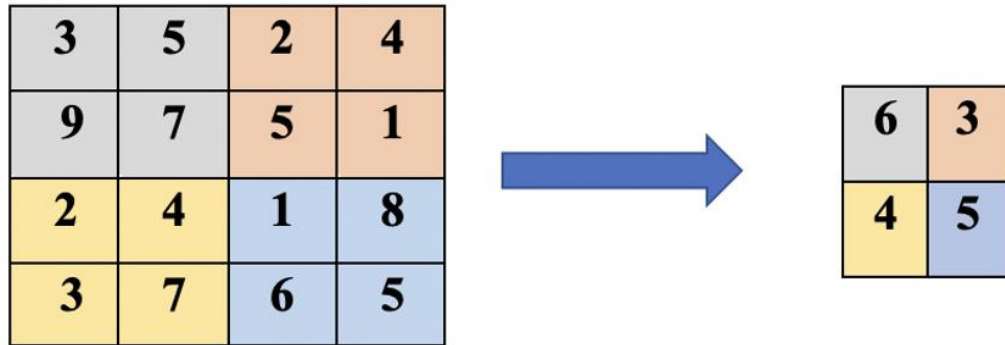
Havuzlama katmanı genel olarak model içerisinde ReLu katmanından sonra karşımıza çıkmaktadır. Havuzlama katmanında yapılan işlemler derinlik boyutunu herhangi bir değişime sebep olmaz ancak görüntünün boyutunu azaltmaktadır. Havuzlama katmanını temel amacı, kendisinden sonra gelecek olan konvolüsyon katmanına giriş olarak verilecek görüntünün boyutunu azaltmaktır. Görüntü boyutunun azaltılması sonucunda bilgi kaybı olmaktadır. Bu bilgi kaybının model için avantajları oluşmaktadır. Bu avantajlardan biri kendisinden sonra gelecek olan ağ katmanındaki hesaplama yükünün azalması, ikinci olarak ise modelin ezber yapmasının engellenmesidir. Havuzlama katmanı daha iyi performans göstermeyi sağladığı için tercih edilmektedir fakat ESA modelinde havuzlama katmanı mecburi değildir, isteğe bağlı kullanılmaktadır. Bazı mimarilerde bu yüzden havuzlama katmanı kullanılırken bazı mimarilerde ise kullanılmamaktadır. Havuzlama işlemlerine kullanılan yöntemlere örnek olarak maksimum havuzlama ve minimum havuzlama verilmektedir. Maksimum havuzlama işlemi, görüntünün filtre üzerinde olan karşılıklarının maksimum olan değerinin alınması işlemidir. Ortalama havuzlama işlemi ise görüntünün filtre üzerinde olan karşılıklarının ortalamasının alınması işlemidir. Genellikle performans açısından daha iyi sonuç verdiği için maksimum havuzlama tercih edilmektedir. Maksimum havuzlama katmanının bir örnek uygulaması Şekil 3.8’de gösterilmektedir. Buradaki örnekte

giriş görüntüsü 4x4'lük olup 2x2 filtre kullanılarak tek adım kaydırmalı maksimum havuzlama işlemi ve iki adım kaydırmalı maksimum havuzlama işlemi gösterilmektedir. Giriş görüntüsü tek adım kaydırmayla 2x2'lik filtreler ile dolaşarak 9 adet 2x2'likten oluşan piksel değerleri elde edilmektedir. Elde edilen 2x2'lik piksel değerlerinden maksimum değer alınarak bir sonraki katman için görüntü elde edilmektedir. 4x4'lük giriş görüntüsü tek adımlı kaydırma ile 3x3'lük görüntü oluştururken iki adımlı kaydırma ile 2x2'lik görüntü oluşmaktadır (İnik & Ülker, 2017).



Şekil 3.8. Maksimum havuzlama işlemi.

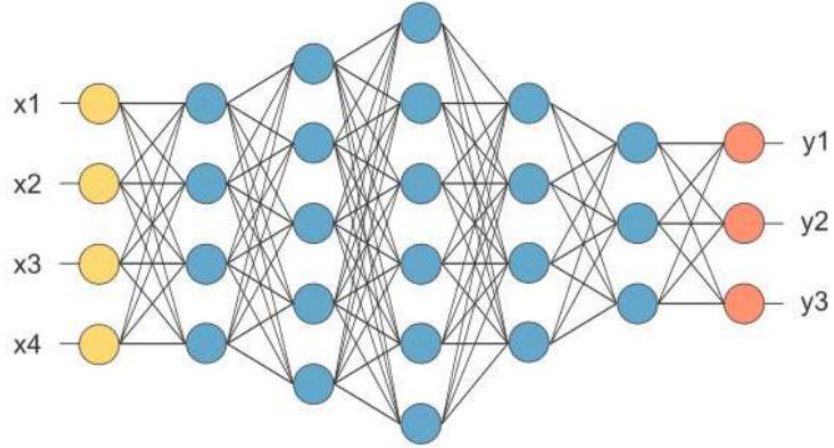
Ortalama havuzlama işleminin uygulanmasının matematiksel örneği Şekil 3.9'da gösterilmektedir. Burada değerlerin ortalamaları alınarak bir sonraki katman için görüntü oluşturulmaktadır. Matematiksel olarak verilen örnekte 4x4'lük görüntü maksimum havuzlama işleminin ardından 2x2'lik görüntü oluşturmaktadır (Bayram, 2020).



Şekil 3.9. Ortalama havuzlama işlemi.

### 3.3.4. Tam Bağlantılı Katman

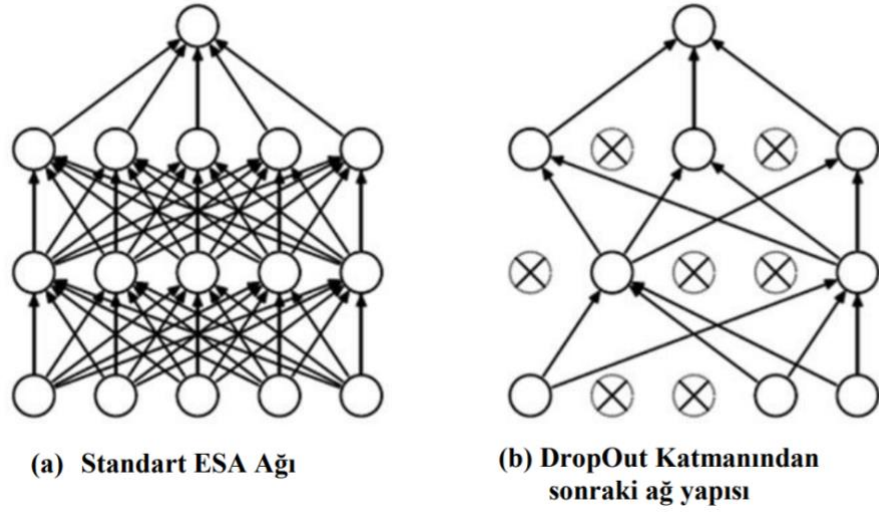
ESA mimarisinde konvolüsyon katmanı, ReLu katmanı ve havuzlama katmanının arkasından tam bağlantılı katman gelmektedir. Tam bağlantılı katman ESA mimarisinde kendisinden önce gelen katmanların tüm alanlarına bağlanmaktadır. ESA mimarisinde bulunan son katmanın üretmiş olduğu değer (matris boyutu) ve bu katmanda bulunan matris boyutunun çarpımı ağırlık matrisini oluşturmaktadır. Bu nedenle adına tam bağlantılı katman denilmektedir (İnik & Ülker, 2017). Tam bağlantılı katmanın önce gelen katmanlarının hepsine bağlandığını Şekil 3.10'da gösterilmektedir (Sallam, 2018).



Şekil 3.10. Tam bağlantılı katmanın uygulanması.

Ağ içerisinde kullanımı genellikle sonda bulunmaktadır (Claesson & Hansson, 2017). Tam bağlantılı katmandan sonra evrişim katmanı kullanılmamaktadır (Çoşkun, Yıldırım, Uçar & Demir, 2017).

ESA büyük veriler ile model eğitimi yaptığından ağ da ezberlemeler oluşabilir. Bu ezberleme yapması modelde istenen bir durum değildir. Ağın ezber yapmasını engellemek amacıyla bırakma katmanı kullanılmaktadır (Srivastava vd., 2014). Standart ESA ağı ile bırakma katmanının kullanılmasıyla oluşturulmuş ağın genel yapısı Şekil 3.11'de gösterilmektedir (İnik & Ülker, 2017).



Şekil 3.11. Drop out katmanının uygulanması.

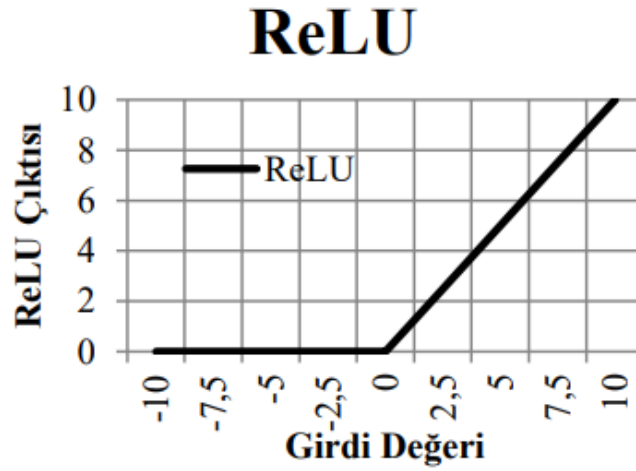
Bu katmanda modelin eğitim aşamasında bırakma uygulamasının kullanımı sıklıkla. Aşırı uyum göstermesinin azaltmak için bırakma oranı kullanılmaktadır. Burada eğitim sırasında düğümler bazı ihtimallere göre devre dışı bırakılmaktadır. Devre dışı bırakılan düğümlerin giriş ve çıkış bağlantıları da kesilir, devre dışı bırakılmaktadır. Devre dışı bırakılan düğümlerin bağlantıları sayesinde model üzerinde hesaplama miktarları azalır. Bu da performans açısından olumlu katkı sağlamaktadır. Modelle bırakma yöntemi ile eğitim sırasında düğümün bağlantıları koparılsa da test sırasında tüm düğümler aktifleştirilir (Srivastava vd., 2014).

### 3.3.5. ReLu (Rectified Linear Unit) Katmanı

ReLu doğrultulmuş doğrusal birim anlamına gelmektedir. ReLu katmanında yapılacak olan işlemler piksel başına uygulanmaktadır. Özellik haritası üzerindeki tüm negatif piksel değerleri sıfır ile yer değiştirir. Her evrişim işleminin ardından ReLu işlemi yapılmaktadır (Amir, 2019).

Aktivasyon fonksiyonlarını bu katman içerisinde gerçekleştirilir. Verilere aktivasyon fonksiyonu uygulandıktan sonra bir değer elde edilir. ReLu katmanı içerisinde kullanılacak birden fazla aktivasyon fonksiyonu vardır. Aktivasyon fonksiyonlarına örnek olarak hiperbolik tanjant fonksiyonu, step fonksiyonu, sinüs fonksiyonu, eşik değeri

fonksiyonları verilebilir. Derin öğrenme ağlarında genellikle kullanılmakta olan aktivasyon fonksiyonu  $f(x) = \max(0, x)$ 'dir (Yang, Nguyen, San, Li & Krishnaswamy). ReLU katmanında en çok kullanılan maksimum fonksiyonu, negatif sayılar girdiler için 0 değerini almakta iken pozitif sayılı girdiler için  $x$  değerini almaktadır. Şekil 3.12'de ReLU'da kullanımı en yaygın olan  $f(x) = \max(0, x)$  fonksiyonun grafiği gösterilmektedir (Sertkaya, 2018). Şekil 3.13'te ReLU işleminin örneği verilmiştir (Sallam, 2018).



Şekil 3.12. ReLU grafiği.

15	20	-10	35
18	-110	25	100
20	-15	25	-10
101	75	18	23

→

15	20	0	35
18	0	25	100
20	0	25	0
101	75	18	23

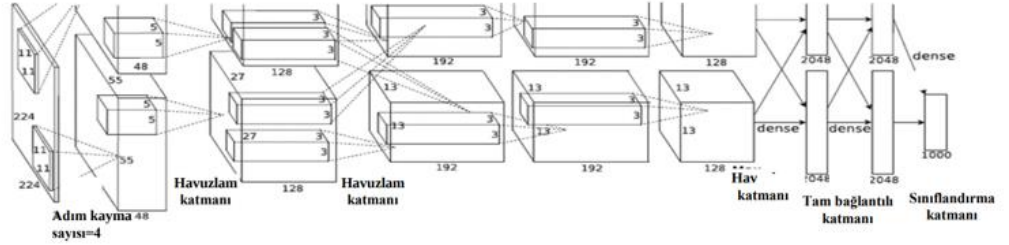
Şekil 3.13. ReLU işlemi örneği.

### 3.3.6. Sınıflandırma Katmanı

Tam bağlantılı katmanın ardından sınıflandırma katmanı gelmekte ve evrimsel sinir ağı mimarisinin sonunda bulunmaktadır. Sınıflandırma işlemleri derin öğrenme mimarilerinde bu katman içerisinde gerçekleştirilmektedir. Sınıflandırma katmanının çıkış değeri sınıflandırma yapılan nesne sayısına denk olmaktadır. Sınıflandırma katmanı içerisinde farklı sınıflandırıcılar tercih edilebilir. Sınıflandırma işleminde nesnelere 0-1 aralığında belirli bir değerde çıktı üretmektedir. Değerin 1'e yakın olması ağın tahmin ettiği nesne olmasıdır (İnik & Ülker, 2017).

### 3.4. AlexNet Modeli

AlexNet adını 2012 ImageNet yarışmasında duyuran bir derin öğrenme modelidir. AlexNet derin öğrenme modeli bir milyondan fazla görüntü ile eğilmiş ve sınıflandırmada yüksek başarı oranları göstermiştir. Katman yapısı olarak, konvansiyonel katman, havuzlama katmanı, tam bağlantılı katman ve sınıflandırma katmanından oluşmaktadır. AlexNet mimarisinin katman yapısı Şekil 3.14'te gösterilmiştir (İnik & Ülker, 2017).



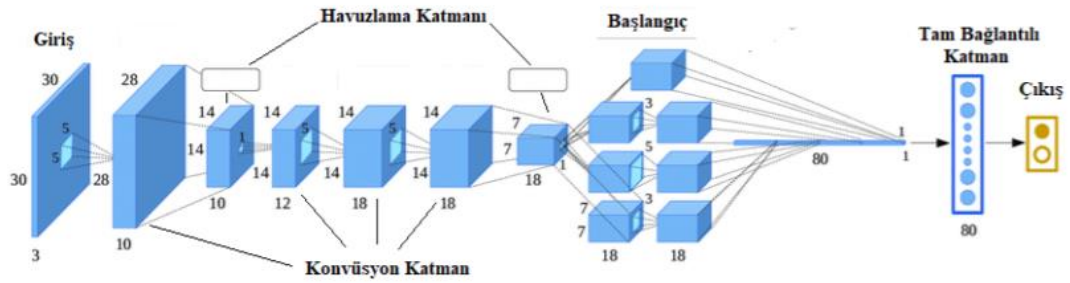
Şekil 3.14. AlexNet mimarisi.

AlexNet derin öğrenme modelinde kullanılan katmanlardan veri birbirine aktarılmaktadır. Veri miktarı arttıkça işlemlerin tamamlanma süresinde artış meydana gelmektedir. Derin öğrenmede mimarisi kullanılarak yapılan çalışmalarda yüksek zaman kaybının önüne geçmek için GPU (Grafical Processing Unit) tercih edilmektedir. GPU kullanılmasıdaki temel amaçlardan biri aynı anda birden fazla iş ve işlem yapabilmeye olanak sağlamasıdır (Krizhevsky, Sutskever & Hinton vd., 2012)

Çalışmada kullanılmış olan AlexNet derin öğrenme modeli mimarisinin uygulamadan alınmış olan görsel çıktısı Ek-1’de gösterilmiştir.

### 3.5. GoogLeNet Modeli

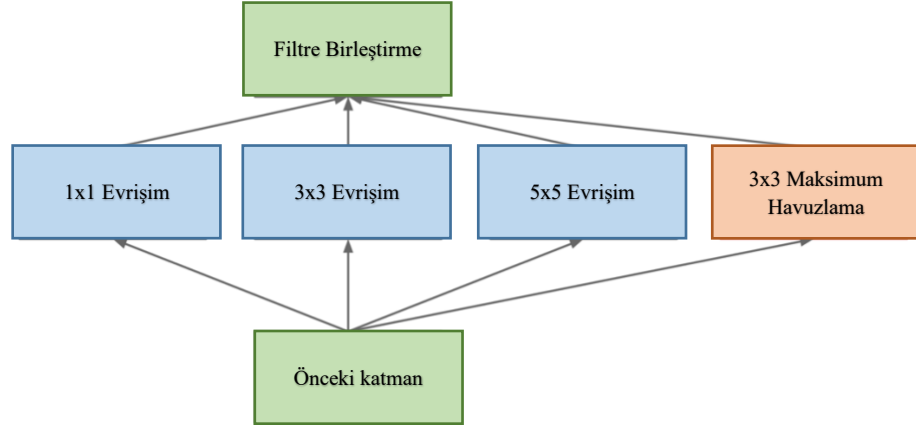
GoogLeNet 2014 yılında ImageNet yarışmasında birinci olmuş bir derin öğrenme modelidir. 2012 yılında yapılmış olan ImageNet yarışmasında %89,06 başarı oranına sahipken 2014 yılında bu başarı oranını artırarak %93,33’e yükseltmiştir. GoogLeNet 22 katmandan oluşan bir mimariye sahiptir. Yüksek boyutlu görüntülerin aşırı yüklenmesini önlemek amacıyla “1x1, 3x3, 5x5” gibi farklı boyutlardaki filtreleri aynı zaman diliminde uygulanmaktadır (Szegedy vd., 2015). Boyut azaltmak amacıyla kullanılan bu filtrelerin bir sonucu olarak GoogLeNet derin öğrenme modelinde AlexNet derin öğrenme modeline göre parametre açısından 12 kat az parametre kullanılmaktadır (Doğan & Türkoğlu, 2019) Şekil 3.15.’te GoogLeNet mimarisi gösterilmektedir (Toğaçar vd., 2019).



Şekil 3.15. GoogLeNet mimarisi.

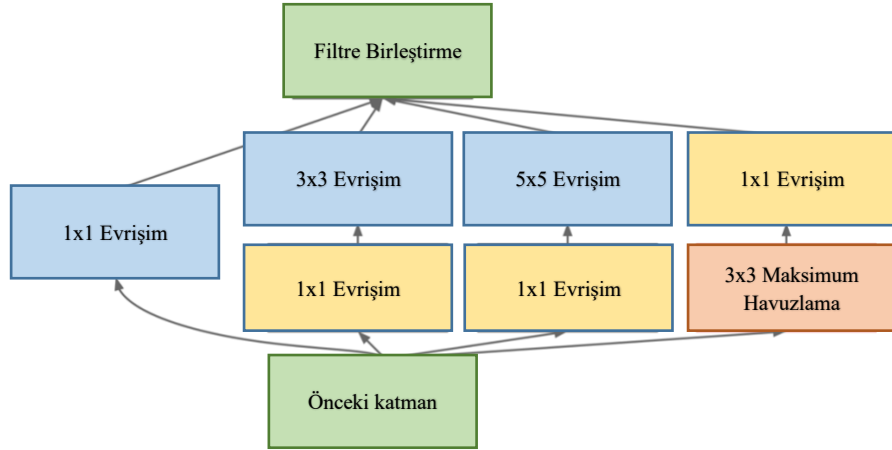
GoogLeNet derin öğrenme modeli mimarisinde gösterilmekte olan başlangıç modülünün iç yapısı Şekil 3.16’da gösterilmektedir. Bu başlangıç modülünde 1x1 evrişim işlemi, 3x3 evrişim işlemi, 5x5 evrişim işlemi ve 3x3 maksimum havuzlama işlemi yapılmaktadır.





Şekil 3.16. Başlangıç modülü.

GoogLeNet derin öğrenme modeli mimarisinde kullanılmakta olan başlangıç modülü işlemlerin sayısında artışa sebep olduğu için boyutu azaltılmış başlangıç modülü kullanılarak işlem sayısında azalama sağlanmıştır. Bu boyut azaltmalı başlangıç modülünde 3x3 evrişim işlemi ve 5x5 evrişim işleminden önce 1x1 evrişim işlemi yapılmaktadır. Boyut azaltmalı başlangıç modülü Şekil 3.17’de gösterilmektedir.



Şekil 3.17. Boyut azaltmalı başlangıç modülü.

Bu çalışmada kullanılan GoogLeNet derin öğrenme modeli mimarisi Ek-2 de gösterilmiştir.

### 3.6. Caffe

Caffe Kaliforniya Üniversitesi, Berkeley tarafından geliştirilmiştir ve açık kaynak kodludur. Caffe derin öğrenme kütüphanelerinden biridir. Görüntü işleme üzerine yazılmıştır. Programlama dillerinden olan python dili ile çalışmaktadır. Caffe kütüphanesinin en çarpıcı özelliklerinden birisi daha önceden eğitilmiş olan modelleri barındırmasıdır (Jia vd., 2014).

İlk olarak görüntü işlemek amacıyla tasarlanmış olan Caffe daha sonradan robotik, ses tanıma, sinir bilimi ve astronomi içinde kullanılmaya başlanmıştır. Caffe'nin hızla yaygınlaşması ile farklı alanlarda da kullanılması beklenmektedir (Uçar & Bingöl, 2018).

### 3.7. Nvidia Digits

Nvidia digits veri kümelerini yönetebilen, çoklu GPU sistemlerinde sinir ağları tasarlayıp geliştirme ortamı sunan, görseller ile performansı gerçek zamanlı olarak izleme imkânı sunan etkileşimli derin öğrenme eğitim sistemidir. Ağın eğitimi başladığında sonra DIGITS modelin mevcut kaybını ve doğruluğunu çizmeye başlar. Bu sayede eğitimin sadece sonunda grafiği görmek yerine aşamalarında da görmemize imkân sağlar (Monn, 2017).

### 3.8. Hiper Parametreler

Hiper parametreleri iki kategoriye ayırmak mümkündür. Birincisi modellere özgü hiper parametrelerden olan aktivasyon fonksiyonları ve dropout katman sayılarıdır. Bu hiper parametreler model hiper parametreleri olarak adlandırılmaktadır. İkinci hiper parametre kategorisi ise iyileştirici (optimizer) hiper parametredir. İyileştirici hiper parametreler modelinin eğitim aşamasında ve modelin optimizasyon aşamasından kullanılmakta olan değişkenlerdir. İyileştirici hiper parametrelere epoch (devir sayısı), öğrenme oranı (learning rate) ve mini-batch boyutundan oluşmaktadır (Sertkaya, 2018).

#### 3.8.1 Öğrenme Oranı (Learning Rate)

Öğrenme oranı, evrimsel sinir ağı modelinin eğitim sürecinde kayıp fonksiyonundan alınan değerlere göre modelin ağırlık değerlerine göre geri yayılım

algoritmaları ile optimize edilir. Elde edilen ağırlık değerlerinin belli değerler ile güncellenir ve bu değerlere öğrenme oranı olarak adlandırılmaktadır. Modelin eğitim sürecinde bu oran değişken olarak kullanılabilir bunun temel sebeplerinden biri öğrenme oranı çok yüksek olduğunda optimum değerlerden uzaklaşmak, çok düşük seçildiğinde ise zaman açısından fazla kayıp olmasıdır. (Barın, 2021)

### **3.8.2 Devir Sayısı (Epoch)**

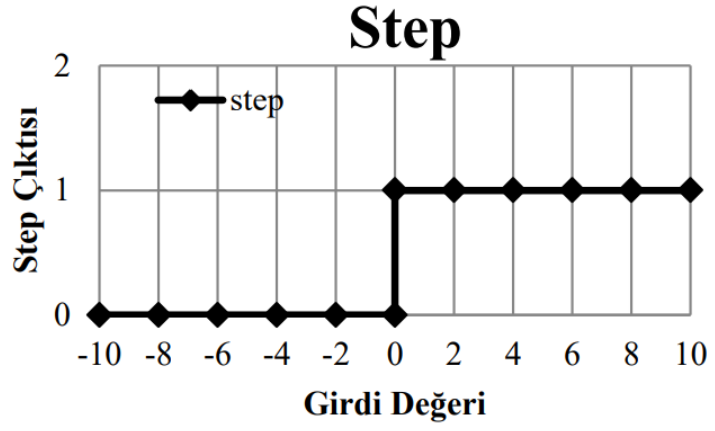
Derin öğrenme modelinde eğitim aşamasında veri setinin eğitim sayısı epoch ile ifade edilmektedir. Modelin eğitiminin bir kez olması yeterli olmamakla birlikte model yinelemeli olarak eğitilmez. Modelin ilk epoch değerinde hata değeri fazla iken daha sonraki epoch değerlerinde hata oranının azalması beklenmektedir. Epoch değeri belirlenirken çok yüksek bir değer seçilmesi modelin eğitim sırasında ezberleme yapmasına neden olacağından dolayı epoch değerinin seçilmesi önemli bir husustur (Barın, 2021).

### **3.8.3 Mini-Batch Boyutu**

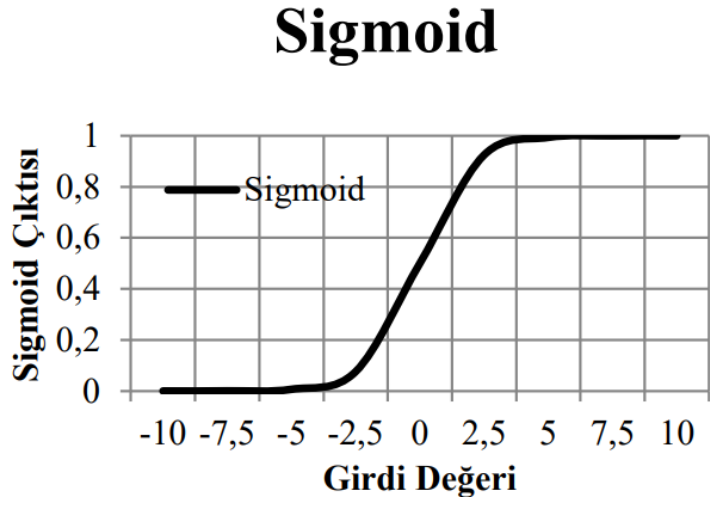
Mini-Batch derin öğrenme modellerinde tek bir adımda tüm verilerin eğitilmesi yerine veri setindeki verileri bölerek elde edilen parçalar üzerinde eğitim yapılmasıdır. Batch değeri veri setindeki verilerin sayısı ile 1 değeri arasında bir değer almaktadır. Batch değeri 1 ise her bir adımda sadece tek 1 veri ile eğitim işlemi yapılmaktadır (Tan, 2019).

### **3.8.4 Aktivasyon Fonksiyonları**

Yapay sinir ağlarından aktivasyon fonksiyonları doğrusal ifadeleri doğrusal olmayan ifadelere dönüştürme işleminde kullanılır. Aktivasyon fonksiyonlarına örnek olarak ReLu fonksiyonu, Step fonksiyonu, Softmax fonksiyonu, Tanh fonksiyonu, Sigmoid fonksiyonu örnek olarak verilebilir. Şekil 3.18’de Step fonksiyonun grafiği, Şekil 3.19’da Sigmoid fonksiyonun grafiği, Şekil 3.20’de Softmax fonksiyonun grafiği ve Şekil 3.21’de Tanh fonksiyonun grafiği gösterilmektedir (Sertkaya, 2018).

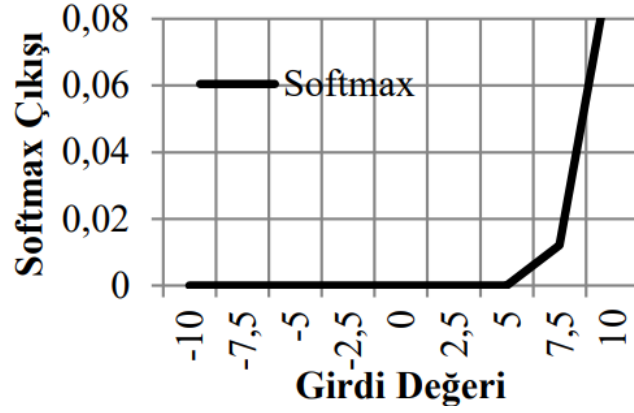


Şekil 3.18. Step fonksiyonu grafiği.



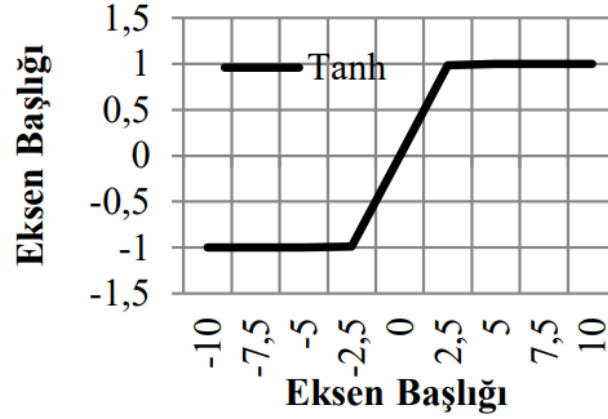
Şekil 3.19. Sigmoid fonksiyonu grafiği.

## Softmax



Şekil 3.20. Softmax fonksiyonu grafiği.

## Tanh



Şekil 3.21. Tanh fonksiyonu grafiği.

### 3.9. Başarım Ölçüm ve Değerlendirme

Sınıflandırma işleminin gerçekleşmesi amacıyla eğitilen modeller, eğitimlerini tamamladıktan sonra test aşamasına geçmektedir. Testler gerçekleştikten sonra birtakım ölçümler kullanılarak sınıflandırma için başarımları elde edilmektedir (Baratloo vd., 2015).

Model performansının doğruluğunu ölçümleyesilerden en sık kullanılanlara; duyarlılık(precision), anma(recall) ve F1 Skoru (F1 Score) örnek olarak verilebilir (Stisen vd., 2015).

Sınıflandırmanın sonuçlarını değerlendirmek için kullanılan YP, YN, DP ve DN değerleri her bir sınıf için doğru ve yanlış sınıflandırılmış olan örnek sayılarını bir matris kullanmaktadır. Bu matrisi karışıklık matrisi adı verilmektedir. Çizelge 3.1’de karışıklık matrisi verilmektedir (Er & Çiğ, 2020).

Çizelge 3.1. Karışıklık matrisi.

<b>Karışıklık Matrisi</b>	<b>Sınıf</b>	<b>Tahmin</b>
YP (Yanlış Pozitif)	Negatif	Pozitif
YN (Yanlış Negatif)	Pozitif	Negatif
DP (Doğru Pozitif)	Pozitif	Pozitif
DN (Doğru Negatif)	Negatif	Negatif

YP (Yanlış Pozitif) negatif sınıf içerisinde yer alırken tahminin pozitif olması, YN (yanlış negatif) pozitif sınıf içerisinde yer alırken tahminin negatif olması iken DP (doğru pozitif) pozitif sınıf içerisinde yer alırken tahminin pozitif olması, DN (doğru negatif) negatif sınıf içerisinde yer alırken tahminin negatif olmasıdır (Er & Çiğ, 2020).

Doğruluk, doğru olarak sınıflandırılmış olan sınıfların yani doğru pozitif ve doğru negatif sınıfların (DP ve DN), tüm sınıflara (DP, DN, YP ve YN) oranı ile elde edilmektedir. Doğruluk, doğru pozitif oran ve yanlış pozitif oran eşitlikleri eşitlik 3.1, eşitlik 3.2 ve eşitlik 3.3’te gösterildiği gibidir (Cengil & Çınar, 2020).

$$\text{Doğruluk} = \frac{|DN|+|DP|}{|YN|+|YP|+|DN|+|DP|} \quad (3.1)$$

$$\text{Doğru Pozitif Oran} = \frac{|DP|}{|YN|+|DP|} \quad (3.2)$$

$$\text{Yanlış Pozitif Oran} = \frac{|YP|}{|YP|+|DN|} \quad (3.3)$$

Doğru pozitif oranı, gerçek pozitiflerin sayısının ne kadarının pozitif sınıf olarak etiketlendiğini vermektedir. Doğru pozitif sınıfının (DP), yanlış negatif sınıf (YN) ile doğru pozitif sınıfın (DP) toplamına bölünmesiyle elde edilmektedir (Polat, Mehr & Cetin, 2017).

ROC eğrisi, doğru pozitif ve yanlış pozitif oranlarının yatay ve dikey ekseninde bulunduğu bir eğridir (Metz, 2006). ROC eğrisinde x ve ye değerlendirilirken sol üst köşeye yakın olan test için daha kullanışlı olarak kabul edilmektedir (Dirican, 2001).

Hassasiyet, her bir sınıf için sınıflandırıcının etkinliğini değerlendirmek amacıyla yapılmaktadır. Hassasiyet pozitif sınıfa ait verilerin gerçek verilerine oranıyla elde edilmektedir (Er & Çiğ, 2020).

Kesinlik (P) pozitif olarak yapılan bir tahminin doğru olma olasılığının tahmin etmekte olan bir ölçümdür. Kesin doğru pozitif sınıfın (DP), yanlış pozitif sınıf (YP) ile doğru pozitif sınıfın (DP) toplamına bölünmesiyle elde edilmektedir. Kesinlik eşitliliği eşitlik 3.4'te verildiği gibidir (Er & Çiğ, 2020).

$$\text{Kesinlik} = \frac{|DP|}{|YP|+|DP|} \quad (3.4)$$

Özgüllük doğru negatif sınıfın (DN), doğru negatif sınıf (DN) ve yanlış pozitif sınıfla toplamına bölünmesiyle elde edilmektedir. F1 skoru ise doğru pozitif sınıfın 2 ile çarpılmasının, 2 doğru pozitif sınıfla yanlış pozitif sınıfın ve yanlış negatif sınıfın toplamına bölünmesiyle elde edilmektedir. F1 skorunu sınıflar arasında denge bulunmadığı durumlarda da kullanılan bir ölçüt iken özgüllük gerçekte negatif olan verilerin oranı elde etmek amacıyla kullanılmaktadır. Özgüllük ve F1 skoru eşitliliği eşitlik 3.5 ve 3.6'da verildiği gibidir (Stanford,2021).

$$\text{Özgüllük} = \frac{|DN|}{|DN|+|YP|} \quad (3.5)$$

$$\text{F1 Skoru} = \frac{2|DP|}{2|DP|+|YP|+|YN|} \quad (3.6)$$

Bu çalışmada doğruluk, doğru pozitif oran, yanlış pozitif oran ve kesinlik değerleri hesaplanıp değerlendirilmiştir.

## BÖLÜM 4

### DENEYLER VE SONUÇLAR

#### 4.1. Veri Seti

İlk veri seti zatürre hastalığına sahip olan insanların akciğer röntgenlerinden ve zatürre hastalığına sahip olmayan insanların akciğerlerinden oluşmaktadır. Veri seti 1 zatürre ve normal olarak sınıflandırılmıştır. Veri setinde yer almakta olan akciğer röntgenlerinin hepsi anonim olup herhangi bir kişisel bilgi içermemektedir.

Veri seti 1 içerisinde yer alan akciğer röntgenleri kaggle adlı web sayfasından elde edilmiştir. Alınan veriler üzerinde herhangi bir değişiklik yapılmamıştır. Kaggle adlı web sitesinden alınan verilerin kaynağı <https://data.mendeley.com/datasets/rscbjbr9sj/2> Citation: [http://www.cell.com/cell/fulltext/S0092-8674\(18\)30154-5](http://www.cell.com/cell/fulltext/S0092-8674(18)30154-5) olarak belirtilmiştir ve veri seti açık kaynaklıdır.

Veri setinde yer alan zatürre hastası insanların akciğer röntgenlerinde zatürrenin birden fazla sebepten kaynaklı olduğunu görmekteyiz. Birinci veri kümesindeki zatürre ve normal olarak ayrılmıştır. İkinci veri seti olarak birinci veri kümesindeki zatürre olanların içerisinde elde edilerek hastaları etiket adından virüs ve bakteri olarak ayrılmıştır, virüs ve bakteri olmak üzere ikiye ayrılan veri setini elde edilmiştir. İkinci veri setimizde verileri üzerinde bir değişiklik yapılmamış sadece etiketler göz önüne alınarak klasörde değişiklik yapılmış, bakteri ve virüs olarak klasör yapısı oluşturulmuştur. Veri 2 içerisindeki sınıflandırma bakteri ve virüs olarak oluşturulmuştur. Röntgen görüntüleri üzerinde herhangi bir bozulma durumu olmamıştır.



Veri setleri üzerinde çalışma yaptıktan sonra tez çalışmamızı genişleterek normal, zatürre ve covid19 hastalarından oluşan veri setini kaggle web sitesinden ilgili alandan alınmıştır. Veri setinin kaynağı: <https://www.kaggle.com/prashant268/chest-xray-covid19-pneumonia>'dir. Bu veri seti açık kaynaklıdır. Veriler anonim olup herhangi bir kişisel bilgi içermemektedir. Bu veriler veri seti 3'ü oluşturmuş olmaktadır.

Veri seti 1, normal ve zatürre olarak gruplandırılmış olan veri setidir. Veri seti 2, veri seti 1 içindeki zatürre hastalarının görüntülerinden elde edilmiş ve gruplandırılmış olan virüs ve bakteri gruplarından oluşmaktadır. Veri seti 3, kaggle web sitesinden ikinci olarak elde ettiğimiz verilerdir ve normal, zatürre ve covid19 hastalarından oluşmaktadır. Veri setleri içerisinde yer alan röntgen sayıları aşağıdaki çizelge 4.1, çizelge 4.2 ve çizelge 4.3'te verilmiştir.

Veri seti 1, veri seti 2 ve veri seti 3 içerisinde yer alan röntgen görüntülerinin formatları JPEG formatındadır.

Veri seti 1 ve 2 de verilerimiz eğitim, doğrulama ve test olarak gruplanmıştır. Veri seti 3 ise eğitim ve test olarak gruplanmıştır. Veri seti 3 içerisindeki eğitim ve test dağılımı %70 ve %30 olarak belirlenmiştir. Veri seti 3, doğrulama verilerini eğitim içerisinde elde etmektedir. Çizelge 4.1'de veri seti 1, çizelge 4.2'de veri seti 2 ve çizelge 4.3'te veri seti 3 için olan veri sayıları gösterilmektedir.

Çizelge 4.1. Veri seti 1 içerisindeki veri sayıları.

<b>Veri Seti 1</b>	<b>Toplam Veri Sayısı</b>	<b>Eğitim Veri Sayısı</b>
Normal	1583	1341
Zatürre	4273	3875

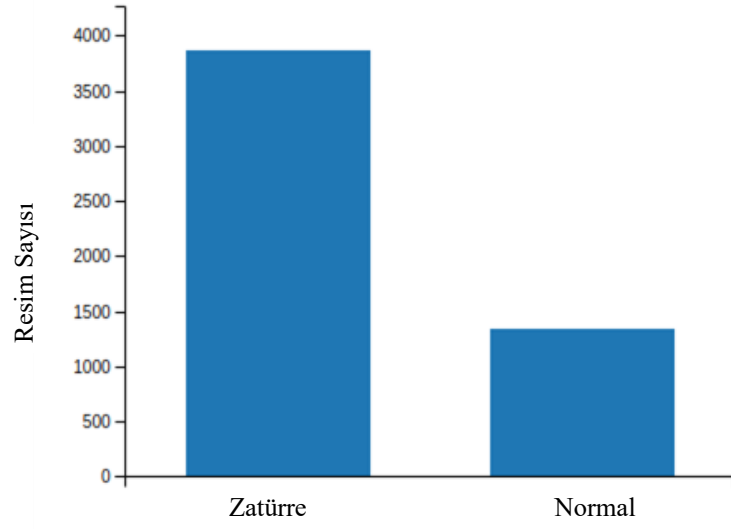
Çizelge 4.2. Veri seti 2 içerisindeki veri sayıları.

<b>Veri Seti 2</b>	<b>Toplam Veri Sayısı</b>	<b>Eğitim Veri Sayısı</b>
Virüs	1523	1345
Bakteri	2802	2530

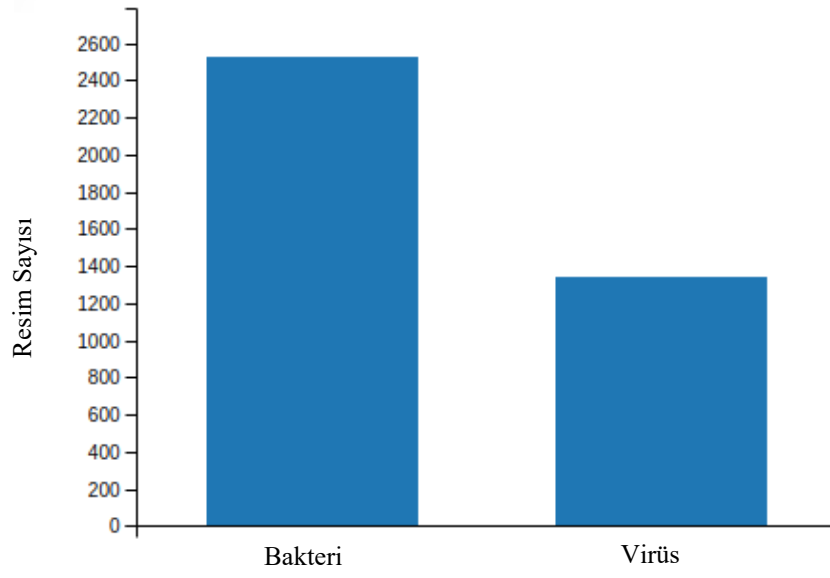
Çizelge 4.3. Veri seti 1 içerisindeki veri sayıları.

<b>Veri Seti 3</b>	<b>Toplam Veri Sayısı</b>	<b>Eğitim + Doğrulama Veri Sayısı</b>
Normal	1508	781
Zatürre	4273	3228
Covid19	576	418

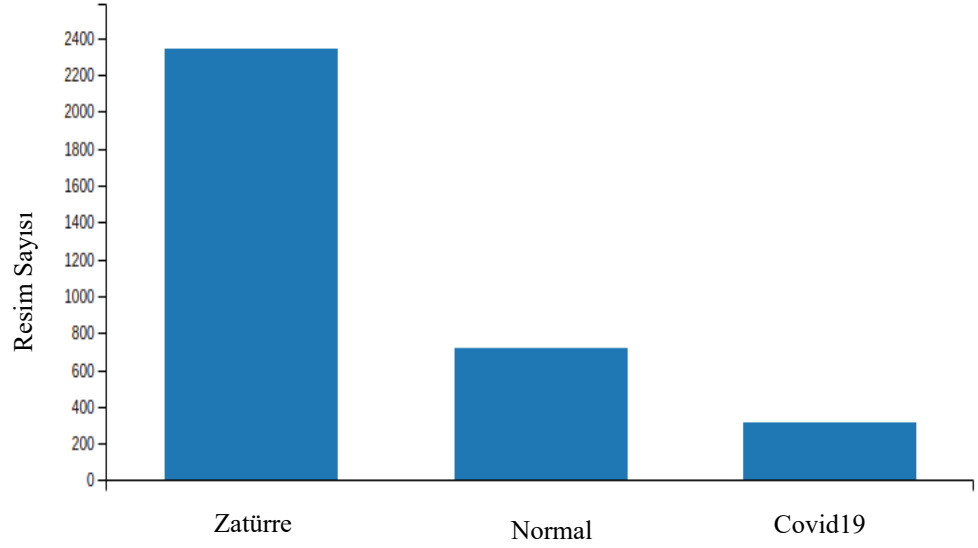
Nvidia Digits kurulumundan sonra localhost ile bağlanıp veri setlerinin tanımlamaları yapıldı. Veri setleri içerisindeki eğitim sayılarının grafik sonuçları Şekil 4.1, Şekil 4.2 ve Şekil 4.3'te gösterilmiştir.



Şekil 4.1. Veri seti 1 için eğitim veri sayısı.



Şekil 4.2. Veri seti 2 için eğitim veri sayısı.



Şekil 4.3. Veri seti 3 için eğitim veri sayısı.

Veri setleri genel olarak değerlendirildiğinde veri seti 1 ve veri seti 2’de sınıflandırma yapılırken ikili sınıflandırma yapılmış olup veri seti 3 için üçlü sınıflandırma yapılmıştır.

Veri setlerin içerdikleri veri sayıları bakımından karşılaştırıldığında en çok veriyi veri seti 3 içerirken en az veriyi veri seti 2 içermektedir. Veri setlerinin toplam veri sayısının karşılaştırmalı gösterimi çizelge 4.4’teki gibidir.

Çizelge 4.4. Veri setleri karşılaştırması.

Veri Seti	Toplam Veri Sayısı
Veri seti 1	5856
Veri seti 2	4325
Veri seti 3	6357

## 4.2. Veri Seti Eğitim ve Test İşlemleri

Modelin eğitim ve test süresine bilgisayarda kurulu olan Ubuntu 16.04 üzerinde gerçekleştirilmiştir. Bilgisayara kurulumu yapılan Nvidia Digits ile veri seti 1, veri seti 2 ve veri seti 3 için eğitim ve test işlemleri yapılmıştır. Digits versiyonu 6.1.1'dir. Eğitim aşamasından önce Nvidia Digits kurulumu tamamlanmış olup gerekli işlemler yapıldıktan sonra localhost üzerinden eğitim ve test işlemleri yapılmıştır. Eğitim sürecinde derin öğrenme kütüphanelerinden biri olan Caffe kullanılmıştır. Kullanılmış olan Caffe versiyonu 0.15.14'tür. Modellerin eğitim ve test aşamasında versiyonların bilgileri localhost üzerinden doğrulanmış ve Şekil 4.4'te gösterilmektedir.

```
Server: yaren-Lenovo-Z50-70
DIGITS version:
6.1.1
Caffe version:
0.15.14
Caffe flavor:
NVIDIA
```

Şekil 4.4. Versiyon bilgileri.

Şekil 4.4'te versiyon bilgileri içerisinde yer almakta olan server bilgisi Nvidia Digits'in kurulu olduğu bilgisayarın adını göstermektedir.

Eğitim aşaması tamamlandıktan sonra test işlemlerinin gerçekleştirilmesi işlemleri de localhost üzerinden yapılmıştır.

### 4.3. Eğitim Süresi

Nvidia Digits ile ilk olarak veri setleri tanımlanmaktadır. Veri seti 1, veri seti 2 ve veri seti 3 ayrı ayrı tanımlanmıştır. Veri setleri tamamlandıktan sonra modellerin ayarlanarak eğitim aşamalarına geçilmektedir. Seçilen veri seti ve modelle eğitim başladığı anda itibaren grafik oluşmaya başlamaktadır. Modelin eğitim süreci bitince grafiğin tamamı tamamlanmış olmaktadır.

Zatürre ve normal olarak gruplandırılmış olan veri seti olan ve 5856 veriden oluşan veri seti 1'in AlexNet derin öğrenme modeli ile eğitilmesi işlemi toplamda 21 saat 7 dakika sürmüştür.

Virüs ve bakteri olarak gruplandırılmış olan veri seti olan ve 4325 veriden oluşan veri seti 2'in AlexNet derin öğrenme modeli ile eğitilmesi işlemi toplamda 12 saat 57 dakika sürmüştür.

Zatürre, Covid19 ve normal olarak gruplandırılmış olan veri seti olan ve 6357 veriden oluşan veri seti 3'in AlexNet derin öğrenme modeli ile eğitilmesi işlemi toplamda 19 saat 4 dakika sürmüştür.

Zatürre, Covid19 ve normal olarak gruplandırılmış olan veri seti olan ve 6357 veriden oluşan veri seti 3'in GoogLeNet derin öğrenme modeli ile eğitilmesi işlemi toplamda 1 gün 18 saat sürmüştür.

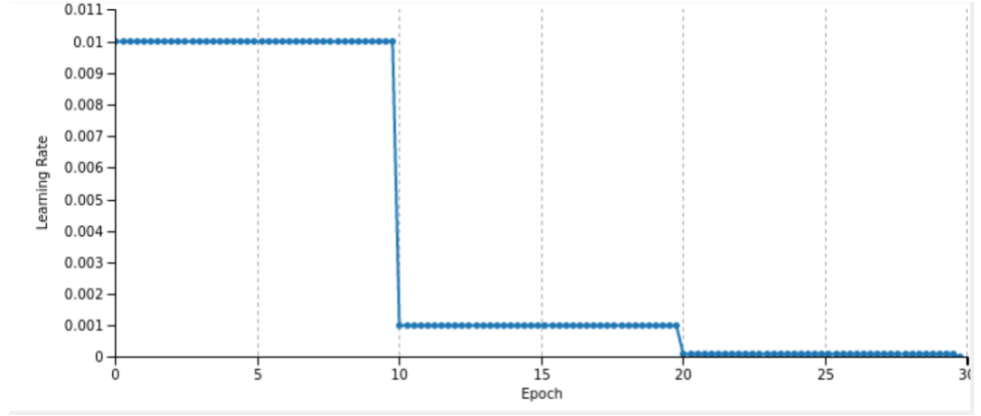
Eğitim süreleri genel olarak karşılaştırıldığında en uzun süren veri seti 3'ün GoogLeNet derin öğrenme modeli ile eğitim aşamasıdır. Bu modelde normal, zatürre ve covid19 ayrımı yapılmaktadır. En kısa süren ise veri seti 2'in AlexNet derin öğrenme modeli ile eğitim aşamasıdır. Bu modelde zatürre hastası röntgenleri içerisinde virüs ve bakteri ayrımı yapılmaktadır. Hem AlexNet hem de GoogLeNet ile eğitilen veri seti 3 ile yapılmış olan çalışmada eğitim süreleri karşılaştırıldığında AlexNet derin öğrenme modelinin eğitim süresinin daha az olduğu görülmektedir. Eğitim sürelerinin karşılaştırmalı hali çizelge 4.5'te gösterilmektedir.

Çizelge 4.5. Eğitim sürelerinin karşılaştırılması.

Veri Seti	Model	Süre
Veri Seti 1	AlexNet	21 saat 7 dakika
Veri Seti 2	AlexNet	12 saat 57 dakika
Veri Seti 3	AlexNet	19 saat 4 dakika
Veri Seti 3	GoogLeNet	1 gün 18 saat

#### 4.4. Değerlendirme Ölçütleri ve Değerlendirmeleri

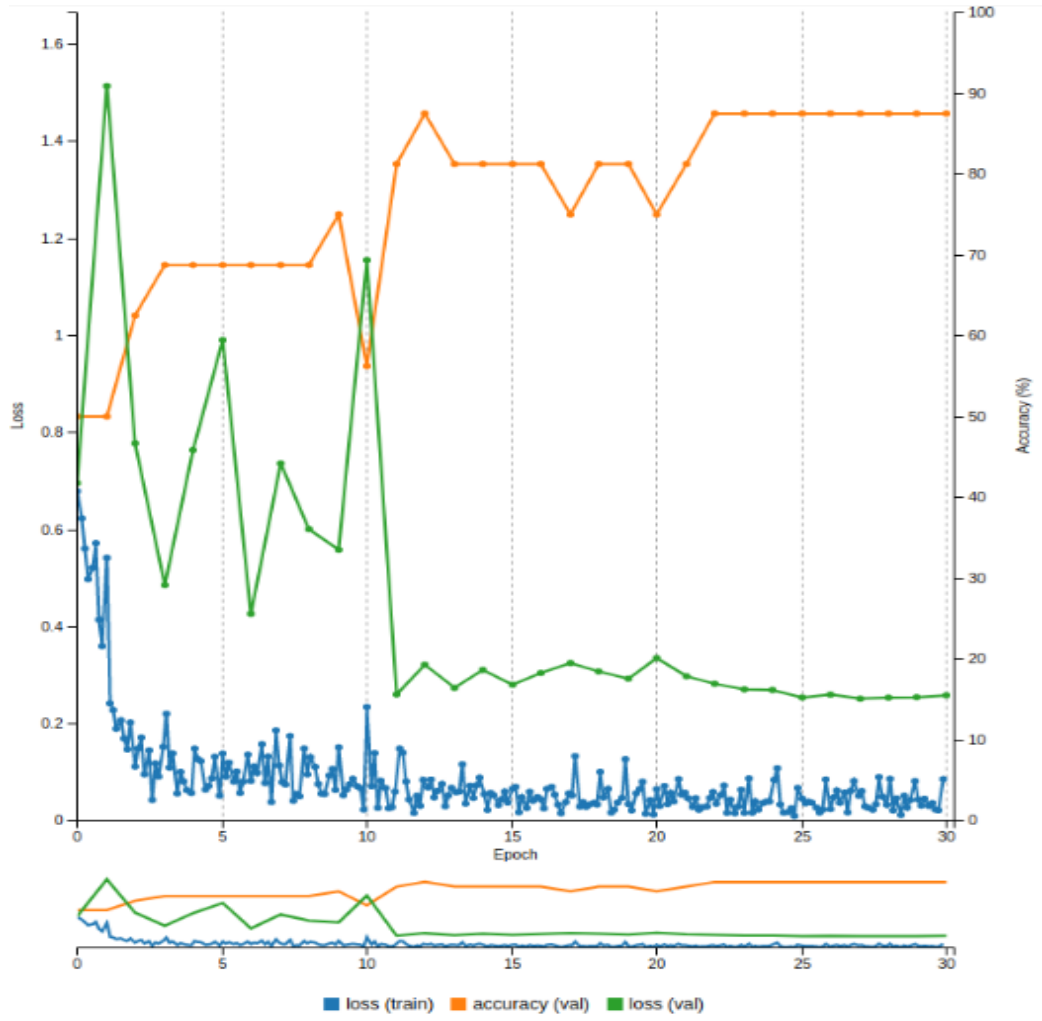
Çalışmada kullanılmış olan modellerin epoch değerlerinin gösteren grafik Şekil 4.5'te gösterildiği gibidir.



Şekil 4.5. Epoch değer grafiği.

Bu çalışmada modelin eğitim aşamasında epoch değeri 30 seçilmiştir. Epoch değeri modelin eğitim sayısını belirlediği için bize modelimizin 30 kez eğitimden geçtiğini göstermektedir. Şekil 4.9 epoch değerinin 0'dan başlayıp 30'a varana kadar oran öğrenme oranları gösterilmektedir. Bu öğrenme oranı modelin başında 0.01 değerinde iken ilerleyen epoch değerlerinde giderek azalmıştır.

Veri seti 1, zatiirre ve normal sınıflandırmasını yapmaktadır. Veri seti AlexNet derin öğrenme modeli ile eğitilmiştir. Modelin eğitiminin sonunda elde ettiğimiz grafik aşağıdaki Şekil 4.6'daki gibidir.



Şekil 4.6. Veri seti 1 model sonucu.

Modelin eğitimini gerçekleştirdikten sonra tek tek seçerek test yapabileceğimiz gibi toplu sınıflandırma işlemi testi de gerçekleştirilmiştir. Bunun için test klasörü içerisindeki verileri .txt formatın liste oluşturup çoklu test için .txt dosyasının yolu verilmiştir. Veri seti 1 için yapılan çoklu sınıflandırma sonuçlarının ilk 10 değer örneği Şekil 4.7’de verilmiştir.



Path	Top predictions			
1 /home/yaren/chest_xray/chest_xray/test/ALL/IM-0001-0001.jpeg	NORMAL	99.86%	PNEUMONIA	0.14%
2 /home/yaren/chest_xray/chest_xray/test/ALL/IM-0003-0001.jpeg	NORMAL	97.16%	PNEUMONIA	2.84%
3 /home/yaren/chest_xray/chest_xray/test/ALL/IM-0005-0001.jpeg	NORMAL	98.93%	PNEUMONIA	1.07%
4 /home/yaren/chest_xray/chest_xray/test/ALL/IM-0006-0001.jpeg	NORMAL	99.71%	PNEUMONIA	0.29%
5 /home/yaren/chest_xray/chest_xray/test/ALL/IM-0007-0001.jpeg	NORMAL	99.82%	PNEUMONIA	0.18%
6 /home/yaren/chest_xray/chest_xray/test/ALL/IM-0009-0001.jpeg	NORMAL	99.38%	PNEUMONIA	0.62%
7 /home/yaren/chest_xray/chest_xray/test/ALL/IM-0010-0001.jpeg	NORMAL	99.32%	PNEUMONIA	0.68%
8 /home/yaren/chest_xray/chest_xray/test/ALL/IM-0011-0001-0001.jpeg	NORMAL	97.18%	PNEUMONIA	2.82%
9 /home/yaren/chest_xray/chest_xray/test/ALL/IM-0011-0001-0002.jpeg	PNEUMONIA	84.82%	NORMAL	15.18%
10 /home/yaren/chest_xray/chest_xray/test/ALL/IM-0011-0001.jpeg	NORMAL	96.93%	PNEUMONIA	3.07%

Şekil 4.7. Test çalışması örnek sonuçları.

Şekil 4.7’de gösterilmiş olan test çalışması örnek olup veri seti 1’in ilk 10 test sonucunu göstermektedir. YP, YN, DP ve DN değerleri hesaplanması için bu test her veri seti ve model için ayrı ayrı oluşturulup hesaplanmıştır. İlk olarak verilen 10 örnekte 1 tahminde yanlış olup zatürre olmayan veri için %84,82 zatürre olarak ifade etmiş iken diğer 9 örnek veride normal olan (zatürre olmayan) veriler için yüksek oranlarda normal olarak ifade etmiştir.

Veri seti 1’in içerisinde yer alan test verileri ile modelimiz test edildikten sonra sonuçları sonucunda YP, YN, DP ve DN değerleri elde edilmiştir. Sonuçlar çizelge 4.6’da gösterilmiştir.

Çizelge 4.6. Veri seti 1'in karışıklık matrisi içindeki değerler.

	Sayıları
YP	10
YN	64
DP	170
DN	380

Normal ve zatürre hastası olarak gruplandırılmış olan veri setinde DP, DN, YP ve YN değerleri hesaplanırken normal röntgenler 1, zatürre hastalığına sahip olan röntgenler 0 değerlerini aldıkları kabul edilmiştir. Bu doğrultuda DP sınıfı gerçekte normal olup tahminin de normal olarak yapıldığı görüntüler, DN sınıfı zatürre hastalığına sahip olanlar olup tahmininin zatürre hastası olarak yapılanlar, YP sınıfı zatürre hastalığına sahip iken normal olarak tahmin edilenler ve YN sınıfı gerçekte normal olup tahmininin zatürre hastası olarak yapılanlardır.

Çizelge 4.6'ya göre doğruluk, pozitif doğru oranı, negatif doğru oranı, kesinlik ve hassasiyet değerleri hesaplanmıştır. Çizelge 4.7'de hesaplama değerleri yer almaktadır.

Çizelge 4.7. Veri seti 1 için değerlendirme ölçütleri.

Veri Seti 1- AlexNet	Doğruluk	Doğru Pozitif Oran	Yanlış Pozitif Oran	Kesinlik	Özgüllük	F1 Skoru
Değer	0,8814	0,7264	0,0256	0,9444	0,9743	0,8212

Veri seti 1'in AlexNet derin öğrenme modeli ile yapılmış olan çalışmanın test aşamasında kullanılmış olan sağlıklı bir insanın röntgen görüntüsünün test sonuçları Şekil 4.8'deki gibidir.



### Predictions

NORMAL	99.79%
PNEUMONIA	0.21%

Şekil 4.8. Normal kategorideki röntgenin test sonucu.

Şekil 4.11’de görünmekte olan sağlıklı akciğer röntgenin test sonucu %99,79 oranla normal ve %0,21 oranla zatürre olarak sınıflandırıp doğru sonuç vermiştir.

Veri seti 1’in AlexNet derin öğrenme modeli ile yapılmış olan çalışmanın test aşamasında kullanılmış olan zatürre hastası bir insanın röntgen görüntüsünün test sonuçları Şekil 4.9’daki gibidir.



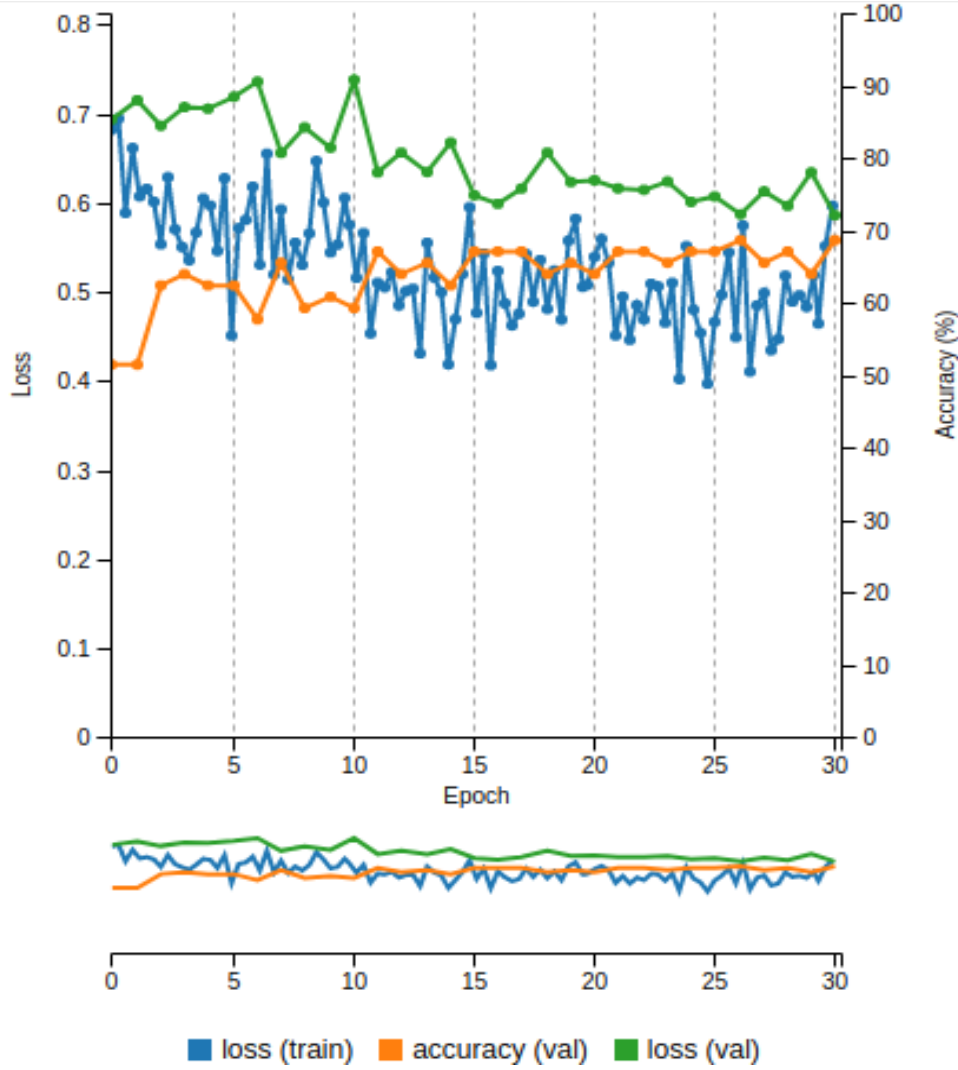
### Predictions

PNEUMONIA	94.5%
NORMAL	5.5%

Şekil 4.9. Zatürre hastasının röntgeninin test sonucu.

Şekil 4.9’da görünmekte olan zatürre hastalığına sahip olan akciğer röntgeninin test sonucu %99,5 oranla zatürre ve %5,5 oranla normal olarak sınıflandırıp doğru sonuç vermiştir.

Veri seti 2 virüs ve bakteri sınıflandırmasını yapmaktadır. Veri seti AlexNet derin öğrenme modeli ile eğitilmiştir. Modelin eğitiminin sonunda elde ettiğimiz grafik aşağıdaki Şekil 4.10’daki gibidir.



Şekil 4.10. Veri seti 2 AlexNet modeli sonucu.

Veri seti 2’in içerisinde yer alan test verileri ile modelimiz test edildikten sonra sonuçları sonucunda YP, YN, DP ve DN değerleri elde edilmiştir. Sonuçlar çizelge 4.6 ve çizelge 4.8’de gösterilmiştir. Bakteri için oluşturulmuş karışıklık matrisinde bakteri 1

değerini alırken virüs 0 değerini almaktadır. Virüs için oluşturulmuş karışıklık matrisinde virüs 1 değerini alırken bakteri 0 değerini almaktadır. Bu doğrultuda çizelge 4.6 da gösterilmekte olan, veri seti 2'nin bakteri için karışıklık matrisi içindeki değerleri aşağıdaki gibi oluşturulmuştur.

- DP: Bakteri sınıfında yer almakta olan ve tahminin bakteri olarak yapılması
- DN: Virüs sınıfında yer almakta olan ve tahminin virüs olarak yapılması
- YP: Virüs sınıfında yer almakta olan fakat tahminin bakteri olarak yapılması
- YN: Bakteri sınıfında yer almakta olan fakat tahminin virüs olarak yapılması

Çizelge 4.9'da gösterilmekte olan veri seti 2'nin virüs için karışıklık matrisi içindeki değerleri aşağıdaki gibi oluşturulmuştur.

- DP: Virüs sınıfında yer almakta olan ve tahminin virüs olarak yapılması
- DN: Bakteri sınıfında yer almakta olan ve tahminin bakteri olarak yapılması
- YP: Bakteri sınıfında yer almakta olan fakat tahminin virüs olarak yapılması
- YN: Virüs sınıfında yer almakta olan fakat tahminin bakteri olarak yapılması

Çizelge 4.8. Veri seti 2'nin bakteri için karışıklık matrisi içindeki değerleri.

	<b>Sayıları</b>
YP	29
YN	11
DP	246
DN	117

Çizelge 4.9. Veri seti 2'nin virüs için karışıklık matrisi içindeki değerleri.

	<b>Sayıları</b>
YP	11
YN	29
DP	117
DN	246

Çizelge 4.8'e göre doğruluk, pozitif doğru oranı, negatif doğru oranı, kesinlik ve hassasiyet değerleri hesaplanmıştır. Çizelge 4.10'da hesaplama değerleri yer almaktadır. Çizelge 4.9'a göre doğruluk, pozitif doğru oranı, negatif doğru oranı, kesinlik ve hassasiyet değerleri hesaplanmıştır. Çizelge 4.11'de hesaplama değerleri yer almaktadır.

Çizelge 4.10. Veri seti 2 bakteri için değerlendirme ölçütleri.

<b>Veri Seti 2- AlexNet</b>	<b>Doğruluk</b>	<b>Doğru Pozitif Oran</b>	<b>Yanlış Pozitif Oran</b>	<b>Kesinlik</b>
Değer	0,9007	0,9571	0,1986	0,8945

Çizelge 4.11. Veri seti 2 virüs için değerlendirme ölçütleri.

<b>Veri Seti 2- AlexNet</b>	<b>Doğruluk</b>	<b>Doğru Pozitif Oran</b>	<b>Yanlış Pozitif Oran</b>	<b>Kesinlik</b>
Değer	0,9007	0,8013	0,0428	0,9140

Test aşamasında veri seti 2 içerisinde yer alan test verilerinden etiketleri göz önüne alınarak 1 bakteri kaynaklı zatürre hastası ve 1 adet virüs kaynaklı zatürre hastası

röntgenlerinin AlexNet derin öğrenme modeli ile yapılmış olan çalışmasının test sonuçları Şekil 4.11 ve Şekil 4.12'deki gibidir.



Şekil 4.11. Virüs kaynaklı zatürre hastasının röntgeninin test sonucu.

Şekil 4.11'de görünmekte olan virüs kaynaklı zatürre hastalığına sahip olan akciğer röntgeninin test sonucu %80,05 oranla virüs ve %19,95 oranla bakteri olarak sınıflandırıp doğru sonuç vermiştir.

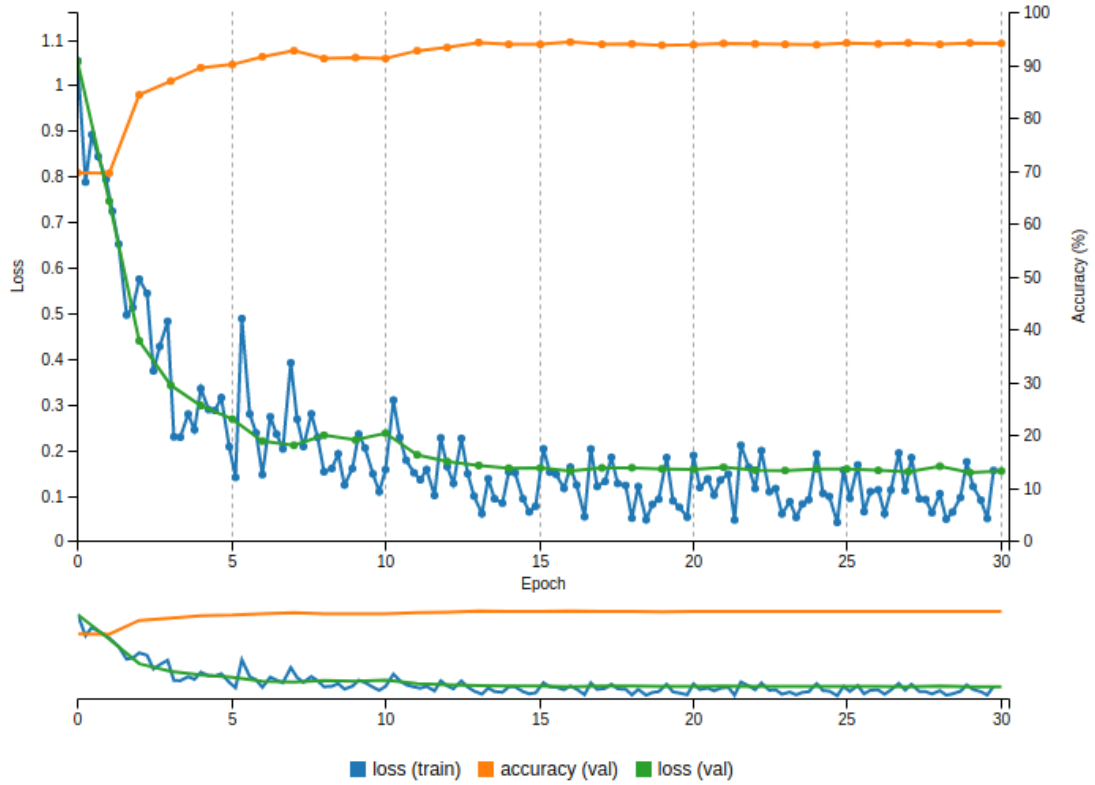


Şekil 4.12. Bakteri kaynaklı zatürre hastasının röntgeninin test sonucu.

Şekil 4.12'te görünmekte olan bakteri kaynaklı zatürre hastalığına sahip olan akciğer röntgeninin test sonucu %91,28 oranla bakteri ve %8,72 oranla virüs olarak sınıflandırıp doğru sonuç vermiştir.

Veri seti 3, normal, zatürre ve covid19 sınıflandırmasını yapmaktadır. Veri seti önce AlexNet derin öğrenme modeli ile eğitilmiştir. Ardından aynı veri seti GoogLeNet derin öğrenme modeli eğitilmiştir.

AlexNet derin öğrenme modeli ile öğretilen modelin eğitiminin sonunda elde ettiğimiz grafik aşağıdaki Şekil 4.13'teki gibidir.



Şekil 4.13. Veri seti 3 AlexNet model sonucu grafiği.

Veri seti 3'in içerisinde yer alan test verileriyle birlikte AlexNet derin öğrenme modeli ile eğitilmiş olan modelimiz test edildikten sonra sonuçları elde edilmiştir. Sonuçlar çizelge 4.12'de gösterilmiştir.

Çizelge 4.12. Veri seti 3 AlexNet modeli test sonuçları.

Tahmin	Gerçek			Toplam
	Normal	Zatürre	Covid19	
Normal	598	51	12	661
Zatürre	28	1087	11	1126
Covid19	1	7	135	143
Toplam	627	1145	158	1930

Çizelge 4.12'ye göre doğruluk, hata oranı, negatif doğru oranı, kesinlik ve hassasiyet değerleri hesaplanmıştır. Çizelge 4.13'te hesaplama değerleri yer almaktadır.



Çizelge 4.13. Veri seti 3 AlexNet modeli için değerlendirme ölçütleri.

Veri Seti 3- AlexNet	Doğruluk	Doğru Pozitif Oran	Yanlış Pozitif Oran	Kesinlik
Değer	0,9523	0.9537	0,0276	0,9046

Test aşamasında veri seti 3 içerisinde yer alan test verilerinden etiketleri göz önüne alınarak 1 adet sağlıklı, 1 zatürre hastası ve 1 adet covid19 hastası röntgenlerinin test sonuçları Şekil 4.14, Şekil 4.15 ve Şekil 4.16’da gösterilmektedir. Bu görüntüler AlexNet derin öğrenme modeli ile eğitilmiş olan modele ait testin sonuçlarıdır.



Şekil 4.14. Normal kategorisindeki röntgenin test sonucu.

Şekil 4.14’te görünmekte olan sağlıklı akciğer röntgenin test sonucu %99,32 oranla normal, %0,67 oranla zatürre ve %0,01 oranla covid19 olarak sınıflandırıp doğru sonuç vermiştir.



#### Predictions

PNEUMONIA	99.83%
COVID19	0.12%
NORMAL	0.04%

Şekil 4.15. Zatürre hastasının röntgeninin test sonucu.

Şekil 4.15'te görünmekte olan zatürre hastalığına sahip olan akciğer röntgeninin test sonucu %99,83 oranla zatürre, %0,12 covid19 ve %0,04 oranla normal olarak sınıflandırıp doğru sonuç vermiştir.



#### Predictions

COVID19	99.99%
PNEUMONIA	0.01%
NORMAL	0.0%

Şekil 4.16. Covid19 hastasının röntgeninin test sonucu.

Şekil 4.16'da görünmekte olan covid19 hastalığına sahip olan akciğer röntgeninin test sonucu %99,99 oranla covid19, %0,01 zatürre ve %0,0 normal olarak sınıflandırıp doğru sonuç vermiştir.

Veri seti 3'ün GoogLeNet derin öğrenme modeli ile eğitildikten sonra modelin sonuç grafiği Şekil 4.17'de gösterilmektedir.



Şekil 4.17. Veri seti 3 GoogLeNet model sonuç grafiği.

GoogLeNet modeli ile eğitilen modeline yapmış yapılan test sonuçları çizelge 4.14'te gösterilmektedir.

Çizelge 4.14. Veri seti 3 GoogLeNet modeli test sonuçları.

Tahmin	Gerçek			Toplam
	Normal	Zatürre	Covid19	
Normal	594	48	3	645
Zatürre	27	1094	7	1128
Covid19	6	3	148	157
Toplam	627	1145	158	1930

Çizelge 4.14'e göre doğruluk, hata oranı, negatif doğru oranı, kesinlik ve hassasiyet değerleri hesaplanmıştır. Çizelge 4.15'te hesaplama değerleri yer almaktadır.

Çizelge 4.15. Veri seti 3 GoogLeNet modeli için değerlendirme ölçütleri.

Veri Seti 3- GoogLeNet	Doğruluk	Doğru Pozitif Oranı	Yanlış Pozitif Oran	Kesinlik
Değer	0,9564	0,9473	0,0391	0,9209

Test aşamasında veri seti 3 içerisinde yer alan test verilerinden etiketleri göz önüne alınarak 1 adet sağlıklı, 1 zatürre hastası ve 1 adet covid19 hastası röntgenlerinin test sonuçları Şekil 4.18, Şekil 4.19 ve Şekil 4.20’de gösterilmektedir. Bu görüntüler GoogLeNet derin öğrenme modeli ile eğitilmiş olan modele ait testin sonuçlarıdır.



#### Predictions

NORMAL	99.47%
PNEUMONIA	0.52%
COVID19	0.01%

Şekil 4.18.Normal kategorisindeki röntgenin test sonucu.

Şekil 4.18’de görünmekte olan sağlıklı akciğer röntgenin test sonucu %99,47 oranla normal, %0,52 zatürre ve %0,01 covid19 olarak sınıflandırıp doğru sonuç vermiştir.



#### Predictions

PNEUMONIA	98.41%
NORMAL	1.58%
COVID19	0.01%

Şekil 4.19. Zatürre hastasının röntgeninin test sonucu.

Şekil 4.19’da görünmekte olan zatürre hastalığına sahip olan akciğer röntgeninin test sonucu %98,41 oranla zatürre, %1,58 normal ve %0,01 covid19 olarak sınıflandırıp doğru sonuç vermiştir.



#### Predictions

COVID19	99.96%
PNEUMONIA	0.03%
NORMAL	0.01%

Şekil 4.20. Covid19 hastasının röntgeninin test sonucu.

Şekil 4.20’de görünmekte olan covid19 hastalığına sahip olan akciğer röntgeninin test sonucu %99,96 oranla covid19, %0,03 zatürre ve %0,01 normal olarak sınıflandırıp doğru sonuç vermiştir.

Veri seti 3 için ayrı ayrı yapılan AlexNet derin öğrenme modelinin değerlendirme sonuçları ile GoogLeNet derin öğrenme modelinin değerlendirme sonuçlarının karşılaştırmaları tablosu çizelge 4.14’teki gibidir.

Çizelge 4.16. Veri seti 3 için değerlendirme karşılaştırması.

<b>Veri Seti 3</b>	<b>Doğruluk</b>	<b>Doğru Pozitif Oran</b>	<b>Yanlış Pozitif Oran</b>	<b>Kesinlik</b>
<b>AlexNet</b>	0,9523	0.9537	0,0276	0,9046
<b>GoogLeNet</b>	0,9564	0,9473	0,0391	0,9209

Çalışmada kullanılmış olan AlexNet ve GoogLeNet derin öğrenme modellerine eğitim aşamasından sonra uygulanmış olan test işleminin bir örneği AlexNet derin öğrenme modeli için Ek-3'te, GoogLeNet derin öğrenme modeli için Ek-4'te verilmiştir.

## BÖLÜM 5

### TARTIŞMA VE SONUÇ

Bu tez çalışmasında AlexNet derin öğrenme modelinden kullanarak göğüs röntgenlerinden zatürre hastası olup olmadığı tespiti gerçekleştirilmiştir. Veri setinde yer alan etiketleri göz önüne alarak 2. veri seti hazırlanmıştır. İkinci veri setinde zatürrenin kaynağının virüs mü bakteri mi olduğuna göre hazırlanmıştır. Ardından bu veri seti de AlexNet derin öğrenme modeli ile eğilmiştir. Bu sayede zatürrenin bakteri kaynaklı mı virüs kaynaklı mı olduğu tespit edilmiştir.

Literatür araştırması sırasında veri setlerinin sağlık alanında kısıtlı olduğu görülmüştür. Bu yüzden açık kaynak olan veri seti üzerinde çalışma yaptıktan sonra etiketleri baz alınarak ikinci veri seti ile daha alt kırılım elde edilmiştir. Nvidia Digits ile modelin eğitilmesi sayesinde modelin performansını gerçek zamanlı izlenmiştir. Kayıp ve doğruluk değerleri grafik üzerinde gözlenmiştir.

Bu tez çalışması veri seti üzerinden yapılmış olan çalışma ve alt kırılım elde edilmesiyle veri setlerinin kısıtlı olduğu alanlarda farklı bir bakış açısı sunmuştur. AlexNet derin öğrenme modeli ile zatürre tespiti ve zatürre sebebinin (virüs ve bakteri kaynaklı) tespitindeki başarı oranları belirlenmiştir.

Zatürre hastalığının teşhis derin öğrenme yöntemleri kullanarak teşhis edilebilmesi için literatürde yapılmış olan farklı çalışmalar mevcuttur. Bu çalışmalarda birçok farklı veri seti ve metotlar tercih edilmiştir. Wang'ın yapmış olduğu çalışma bunlardan biridir. AlexNet derin öğrenme modeli ile yapmış olduğu çalışma başarı olarak 0,54 olarak belirtilmiştir (Wang vd., 2017).

Bu tez çalışmasında ilk olarak var olan bir veri setinden yeni kategorileri elde ederek veri seti oluşturulmuş ve bu iki veri seti kıyaslanmıştır. Bu çalışmanın ardından diğer veri seti ile AlexNet ve GoogLeNet derin öğrenme modelleri karşılaştırılmıştır.

Veri setleri içerdikleri veri setleri bakımından kıyaslandığında en çok veri içeren veri seti 3, en az veri içeren veri seti 2'dir. Bununla birlikte veri setlerinin eğitim sürecinde de en uzun süren eğitim süreci 1 gün 18 saat ile veri seti 3'ün GoogLeNet derin öğrenme modeli eğitimi olmuştur. Veri setleri içerisinde verinin sayısı genellikle eğitim süresiyle doğru orantılı olmaktadır. Veri seti 2 'in AlexNet derin öğrenme modeliyle eğitim süresi 12 saat 57 dakika sürmesini bunu göstermektedir. Fakat veri sayısı olarak daha fazla olan veri seti 3'ün AlexNet derin öğrenme modeli ile eğitim süresi 19 saat 4 dakika sürerken veri seti 1'in AlexNet derin öğrenme modeli eğitim süresi 21 saat 7 dakika sürmüştür. Buradan yola çıkarak eğitim süresini etkileyen faktörlerden biri olarak veri setindeki verilerin sayısı olduğu görülmektedir. Buna ek olarak aynı veri setini (veri seti 3) hem AlexNet derin öğrenme modeli ile hem de GoogLeNet derin öğrenme modeli ile eğitilip eğitim süreleri karşılaştırıldığında AlexNet derin öğrenme modelinin GoogLeNet derin öğrenme modeline göre daha kısa sürdüğü görülmektedir.

Veri seti 1 ile AlexNet derin öğrenme modeli üzerinde çalışma yapılmıştır. Karmaşıklık matrisinde normal 1, zatürre 0 olarak kabul edilmiş ve değerlendirme ölçütleri hesaplanmıştır. Doğruluk 0,8814, doğru pozitif oran 0,7264, yanlış pozitif oran 0,0256 ve kesinlik değeri 0,9444 olarak elde edilmiştir.

Veri seti 1 ile veri seti 2'nin test sonuçları kıyaslandığında görüntülerin aynı olmasına rağmen farklı kategorilere ayrılarak sayılarının azalmasının sonuçlarını görmekteyiz. Veri setimiz içerisindeki sayıların azalmasının sonucunun yansıması olarak kesinlik değerinin azaldığı görülmektedir. Veri seti 1 ile AlexNet derin öğrenme modelinde kesinlik değeri 0,9444 iken veri seti 2 ile yapılan AlexNet derin öğrenme modellerinde bakteri 1 olarak kabul edilen hesaplama kesinlik değeri 0,8945, virüs 1 olarak kabul edilen hesaplamada kesinlik değeri 0,9140 olarak elde edilmiştir. Kullanılan diğer değerlendirme ölçütlerinde (doğru pozitif oran, yanlış pozitif oran ve doğruluk) değişimler görülmüştür. Veri setlerinin içerisindeki veri sayılarının azalması ile ilgili bir orantı sonuçlara yansımamıştır.



Veri seti 2’de aynı veri setinin bakteri ve virüs ayrımı yapmak için iki adet karışıklık matrisi oluşturularak doğruluk değerleri karşılaştırılmıştır. Bakteri için oluşturulan karışıklık matrisinde bakteri 1 virüs 0 olarak, virüs için oluşturulan karışıklık matrisinde bakteri 0 virüs 1 olarak kabul edilmiştir. Bakteri için doğruluk 0,9007, doğru pozitif oran 0,9571, yanlış pozitif oran 0,1986, kesinlik 0,8945 olarak elde edilirken virüs için doğruluk 0,9007, doğru pozitif oran 0,8013, yanlış pozitif oran 0,0428, kesinlik 0,9140 olarak elde edilmiştir.

Veri seti 3’ü %70 eğitim ve %30 test olarak ayarlanıp hem AlexNet hem de GoogLeNet derin öğrenme modeli ile eğitilip test edilmiştir. Test sonuçlarına göre her iki model içinde de karışıklık matrisi oluşturulmuştur. Karışıklık matrisleri baz alınarak değerlendirme ölçütleri hesaplanmış ve sonuçlar karşılaştırılmıştır. Test sonuçlarına göre doğruluk oranı değerleri birbirlerine çok yakın olmakla birlikte doğruluk için GoogLeNet derin öğrenme modeli daha verimli sonuç vererek 0,9564 değeri elde edilmiştir. Doğru pozitif oran doğruluk oranının aksine AlexNet derin öğrenme modelinde daha verimli sonuç vererek 0,9537 değeri elde edilmiştir. Yanlış pozitif oran değeri iki modelde de yakın olmakla birlikte AlexNet derin öğrenme modelinde daha yüksek çıkarak 0,0391 değeri elde edilmiştir. Kesinlik, GoogLeNet derin öğrenme modelinde daha verimli çıkarak 0,9209 değeri elde edilmiştir. Veri seti 3 için elde edilen değerlendirme ölçütlerine göre genel bakıldığında AlexNet ve GoogLeNet derin öğrenme modellerinin sonuçları birbirlerine çok yakın olarak çıkmaktadır.

Yapılmış olan bu çalışmada veri setleri 2 farklı modelle eğitilmiş ve sonuçları karşılaştırılmış. Ayrıca veri setindeki verilerin sayısının buna etkisi de göz önüne alınarak buna ek olarak farklı karşılaştırma sonuçları elde edilmiştir. Bu karşılaştırma sonuçlarını, verileri, derin öğrenme modellerini, eğitim sürelerini ve değerlendirme ölçütlerini göz önüne alınarak gelecek çalışmalarda daha yüksek doğruluk, kesinlik, doğru pozitif oran gibi değerlendirme ölçütlerinde başarı oranını yükseltilebilir.

## KAYNAKLAR

Akpınar, B. (2019). *Görüntü sınıflandırma için derin öğrenme ile bayesçi derin öğrenme yöntemlerinin karşılaştırılması*. Afyon Kocatepe Üniversitesi/Fen Bilimleri Enstitüsü, Afyonkarahisar.

Akter, S., Shamsuzzaman, S. M., & Jahan, F. (2014). Community acquired bacterial pneumonia: aetiology, laboratory detection and antibiotic susceptibility pattern. *Malays J Pathol*, 36(2), 97-103.

Amir, A. (2019). Convolutional Neural Network (CNN) with Practical Implementation 25 Nisan 2021 tarihinde <https://medium.com/machine-learning-researcher/convlutional-neural-network-cnn-2fc4faa7bb63> adresinden erişildi.

Anthimopoulos, M., Christodoulidis, S., Ebner, L., Christe, A., & Mougiakakou, S. (2016). Lung pattern classification for interstitial lung diseases using a deep convolutional neural network. *IEEE transactions on medical imaging*, 35(5), 1207-1216.

Ari, A., & Hanbay, D. (2018). Deep learning based brain tumor classification and detection system. *Turkish Journal of Electrical Engineering & Computer Sciences*, 26(5), 2275-2286.

Atlan, F., & Pençe, İ. (2021). Yapay Zekâ ve Tıbbi Görüntüleme Teknolojilerine Genel Bakış. *Acta Infologica*, 5(1), 207-230.

Ayan, E., & Ünver, H. M. (2019, April). Diagnosis of pneumonia from chest x-ray images using deep learning. In 2019 Scientific Meeting on Electrical-Electronics & Biomedical Engineering and Computer Science (EBBT) (pp. 1-5). Ieee.

Uçar A. & Bingöl, M. S. (2018). Derin öğrenmenin Caffe kullanılarak grafik işleme kartlarında değerlendirilmesi. *Dicle Üniversitesi Mühendislik Fakültesi Mühendislik Dergisi*, 9(1), 39-49.

Balaban, M. E. & Kartal, E. (2015). *Veri Madenciliği ve Makine Öğrenmesi*. İstanbul: Çağlayan Kitabevi.

Baratloo, A., Hosseini, M., Negida, A., & El Ashal, G. (2015). Part 1: simple definition and calculation of accuracy, sensitivity and specificity.

Barın, S. (2021). *Deri lezyonlarının derin öğrenme yöntemleri ile segmentasyonu*. Afyon Kocatepe Üniversitesi/Fen Bilimleri Enstitüsü, Afyonkarahisar.

Bayram, F. (2020). Derin öğrenme tabanlı otomatik plaka tanıma. *Politeknik Dergisi*, 23(4), 955-960.

Cengil, E., & Çınar, A. Göğüs Verileri Metrikleri Üzerinden Kanser Sınıflandırılması. *Dicle Üniversitesi Mühendislik Fakültesi Mühendislik Dergisi*, 11(2), 513-519.

Chien, S. I. J., Ding, Y., & Wei, C. (2002). Dynamic bus arrival time prediction with artificial neural networks. *Journal of transportation engineering*, 128(5), 429-438.

Ciampi, F., Chung, K., Van Riel, S. J., Setio, A. A. A., Gerke, P. K., Jacobs, C., ... & Van Ginneken, B. (2017). *Towards automatic pulmonary nodule management in lung cancer screening with deep learning*. *Scientific reports*, 7(1), 1-11.

Claesson, L., & Hansson, B. (2017). *Deep learning methods and applications- Classification of traffic signs and detection of Alzheimer's disease from images*. Chalmers University of Technology Department of Signals and Systems Division of Signal Processing and Biomedical Engineering, Sweden.

Coşkun, M., Yıldırım, Ö., Uçar, A. & Demir, Y. (2017). An overview of popular deep learning methods. *European Journal of Technique (EJT)*, 7(2), 165-176.

Deng, L., & Yu, D. (2014). Deep learning: methods and applications. *Foundations and trends in signal processing*, 7(3–4), 197-387.

Deniz, E., Şengür, A., Kadiroğlu, Z., Guo, Y., Bajaj, V., & Budak, Ü. (2018). Transfer learning based histopathologic image classification for breast cancer detection. *Health information science and systems*, 6(1), 1-7.

Dimililer, K., & İlhan, A. (2016). Effect of image enhancement on MRI brain images with neural networks. *Procedia Computer Science*, 102, 39-44.

Dirican, A. (2001). Tanı testi performanslarının değerlendirilmesi ve kıyaslanması. *Cerrahpaşa Tıp Dergisi*, 32(1), 25-30.

Doğan, F., & Türkoğlu, İ. (2019). Derin öğrenme modelleri ve uygulama alanlarına ilişkin bir derleme. *Dicle Üniversitesi Mühendislik Fakültesi Mühendislik Dergisi*, 10(2), 409-445.

ER, M. B., & Harun, Ç. İ. Ğ. (2020). Türk Müziği Uyarımları Kullanılarak İnsan Duygularının Makine Öğrenmesi Yöntemi İle Tanınması. *Gazi Üniversitesi Fen Bilimleri Dergisi Part C: Tasarım ve Teknoloji*, 8(2), 458-474.

Gülgün, O. D. (2020). *Derin Öğrenme Yöntemleri Kullanılarak Tıbbi Görüntülerden Teşhis*. Mersin Üniversitesi Den Bilimleri Enstitüsü Bilgisayar Mühendisliği Anabilim Dalı, Mersin

Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep learning*. MIT press.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., ... & Bengio, Y. (2014). Generative adversarial nets. *Advances in neural information processing systems*, 27.

Gültepe, Y. (2019). Makine öğrenmesi algoritmaları ile hava kirliliği tahmini üzerine karşılaştırmalı bir değerlendirme. *Avrupa Bilim ve Teknoloji Dergisi*, (16), 8-15.

Hashimoto, D. A., Rosman, G., Rus, D., & Meireles, O. R. (2018). Artificial intelligence in surgery: promises and perils. *Annals of surgery*, 268(1), 70.

Işık, G., & Artuner, H. (2018, May). A dataset for Turkish dialect recognition and classification with deep learning. In 2018 26th Signal Processing and Communications Applications Conference (SIU) (pp. 1-4). IEEE.

Jia, Y., Shelhamer, E., Donahue, J., Karayev, S., Long, J., Girshick, R., ... & Darrell, T. (2014, November). Caffe: Convolutional architecture for fast feature embedding. In Proceedings of the 22nd ACM international conference on Multimedia (pp. 675-678).

Keskenler, M. F., & Keskenler, E. F. (2017). Geçmişten günümüze yapay sinir ağları ve tarihçesi. *Takvim-i Vekayi*, 5(2), 8-18.

Korkmaz, S. (2019). Small Drug Molecule Classification Using Deep Neural Networks. *Turkiye Klinikleri Journal of Biostatistics*, 11(2).

Koitka, S., & Friedrich, C. M. (2016, September). Traditional Feature Engineering and Deep Learning Approaches at Medical Classification Task of ImageCLEF 2016. In CLEF (Working Notes) (pp. 304-317).

Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25, 1097-1105.

Kumar, S., Dabas, C., & Godara, S. (2017). Classification of brain MRI tumor images: a hybrid approach. *Procedia computer science*, 122, 510-517.

LeCun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11), 2278-2324.

Mahesh, B. (2020). Machine learning algorithms-a review. *International Journal of Science and Research (IJSR)*. [Internet], 9, 381-386.

Ma, L., Lu, G., Wang, D., Wang, X., Chen, Z. G., Muller, S., ... & Fei, B. (2017, March). Deep learning based classification for head and neck cancer detection with hyperspectral imaging in an animal model. In *Medical Imaging 2017: Biomedical Applications in Molecular, Structural, and Functional Imaging* (Vol. 10137, p. 101372G). International Society for Optics and Photonics.

Metz, C. E. (2006). Receiver operating characteristic analysis: a tool for the quantitative evaluation of observer performance and imaging systems. *Journal of the American College of Radiology*, 3(6), 413-422.

Monn, Dominic. (2017). How to use NVIDIA DIGITS for image classification 20 Nisan 2021 tarihinde <https://medium.com/@dmonn/how-to-use-nvidia-digits-for-image-classification-79600acbe0db> adresinden erişildi.

Niebur, D., & Germond, A. J. (1991, July). Power flow classification for static security assessment. In *Proceedings of the First International Forum on Applications of Neural Networks to Power Systems* (pp. 83-88). IEEE.

Nielsen, M. A. (2015). *Neural networks and deep learning* (Vol. 25). San Francisco, CA: Determination press.

Özcan, H. (2014). *Çok düşük çözünürlüklü yüz imgelerinde derin öğrenme uygulamaları*. Deniz Harp Okulu Komutanlığı/Deniz Bilimleri ve Mühendisliği Enstitüsü, İstanbul.

Saxena, H., (2015). DICOM Image Segmentation with CNNs in Tensorflow 20.07.2021 tarihinde <https://github.com/harsh1795/CNN-DICOM-Segmentation> adresinden erişildi.

Sertkaya, M. E. (2018). *Derin öğrenme tekniklerinin biyomedikal imgeler üzerine uygulamaları*. Fırat Üniversitesi / Fen Bilimleri Enstitüsü, Elazığ.

Siar, M., & Teshnehlal, M. (2019, October). Brain tumor detection using deep neural network and machine learning algorithm. In 2019 9th International Conference on Computer and Knowledge Engineering (ICCKE) (pp. 363-368). IEEE.

Stephen, O., Sain, M., Maduh, U. J., & Jeong, D. U. (2019). An efficient deep learning approach to pneumonia classification in healthcare. *Journal of healthcare engineering*, 2019.

Suzuki, S., Zhang, X., Homma, N., Ichiji, K., Sugita, N., Kawasumi, Y., ... & Yoshizawa, M. (2016, September). Mass detection using deep convolutional neural network for mammographic computer-aided diagnosis. In 2016 55th Annual Conference of the Society of Instrument and Control Engineers of Japan (SICE) (pp. 1382-1386). IEEE.

Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., ... & Rabinovich, A. (2015). Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 1-9).

S. Akter, Shamsuzzaman, and F. Jahan, "Community Acquired Pneumonia," pp. 1–5, 2015.

Peng, H. (2015). *Air quality prediction by machine learning methods*. University of British Columbia, Columbia.

Toğaçar, M., & Ergen, B. (2019). Biyomedikal Görüntülerde Derin Öğrenme ile Mevcut Yöntemlerin Kıyaslanması. *Fırat Üniversitesi Mühendislik Bilimleri Dergisi*, 31(1), 109-121.

Toğaçar, M., Ergen, B., & Sertkaya, M. E. (2019). Zatürre Hastalığının Derin Öğrenme Modeli ile Tespiti. *Fırat University Journal of Engineering*, 31(1).

Traore, B. B., Kamsu-Foguem, B., & Tangara, F. (2018). *Deep convolution neural network for image recognition*. *Ecological Informatics*, 48, 257-268.

Tüfekçi, M., & Karpat, F. (2019). Derin Öğrenme Mimarilerinden Konvolüsyonel Sinir Ağları (CNN) Üzerinde Görüntü İşleme-Sınıflandırma Kabiliyetinin Arttırılmasına

Yönelik Yapılan Çalışmaların İncelenmesi. In International Conference on Human-Computer Interaction, Optimization and Robotic Applications (pp. 28-31).

İnik, Ö., & Ülker, E. (2017). Derin öğrenme ve görüntü analizinde kullanılan derin öğrenme modelleri. *Gaziosmanpaşa Bilimsel Araştırma Dergisi*, 6(3), 85-104.

Patterson, J., & Gibson, A. (2017). Deep learning: A practitioner's approach. " O'Reilly Media, Inc."

Polat, H., Mehr, H. D., & Cetin, A. (2017). Diagnosis of chronic kidney disease based on support vector machine by feature selection methods. *Journal of medical systems*, 41(4), 55.

Ranschaert, E. R., Morozov, S., & Algra, P. R. (2019). Artificial Intelligence in Medical Imaging: Opportunities, Applications and Risks. Cham: Springer International Publishing.

Ravi, D., Wong, C., Deligianni, F., Berthelot, M., Andreu-Perez, J., Lo, B., & Yang, G. Z. (2016). Deep learning for health informatics. *IEEE journal of biomedical and health informatics*, 21(1), 4-21.

Sallam, M. K. (2018). *Deep Structure Base On Convolutional Neural Networks For Identificatio Of Chest Diesases*. Near East University/Computer Engineering, KKTC.

Sarraf, S., & Tofighi, G. (2016, December). Deep learning-based pipeline to recognize Alzheimer's disease using fMRI data. In 2016 future technologies conference (FTC) (pp. 816-820). IEEE.

Sertkaya, M. E. (2018). *Derin öğrenme tekniklerinin biyomedikal imgeler üzerine uygulamaları*. Fırat Üniversitesi/Fen Bilimleri Enstitüsü, Elazığ.

Shaikh, F (2017). Why are GPUs necessary for training Deep Learning models? 20 Nisan 2021 tarihinde <https://www.analyticsvidhya.com/blog/2017/05/gpus-necessary-for-deeplearning/> adresinden erişildi.



Shin, Y., & Balasingham, I. (2017, July). Comparison of hand-craft feature based SVM and CNN based deep learning framework for automatic polyp classification. In 2017 39th annual international conference of the IEEE engineering in medicine and biology society (EMBC) (pp. 3277-3280). IEEE.

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. (2014). Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1), 1929-1958.

Stanford (2021). 6 Nisan 2022 tarihinde <https://stanford.edu/~shervine/l/tr/teaching/cs-229/cheatsheet-machine-learning-tips-and-tricks#classification-metrics> adresinden erişildi.

Stisen, A., Blunck, H., Bhattacharya, S., Prentow, T. S., Kjærgaard, M. B., Dey, A., ... & Jensen, M. M. (2015, November). Smart devices are different: Assessing and mitigating mobile sensing heterogeneities for activity recognition. In *Proceedings of the 13th ACM conference on embedded networked sensor systems* (pp. 127-140).

Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., ... & Rabinovich, A. (2015). Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 1-9).

Tan, Z. (2019). *Derin öğrenme yardımıyla araç sınıflandırma*. Fırat Üniversitesi / Fen Bilimleri Enstitüsü, Elazığ.

Wang, X., Peng, Y., Lu, L., Lu, Z., Bagheri, M., & Summers, R. M. (2017). Chestx-ray8: Hospital-scale chest x-ray database and benchmarks on weakly-supervised classification and localization of common thorax diseases. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 2097-2106).

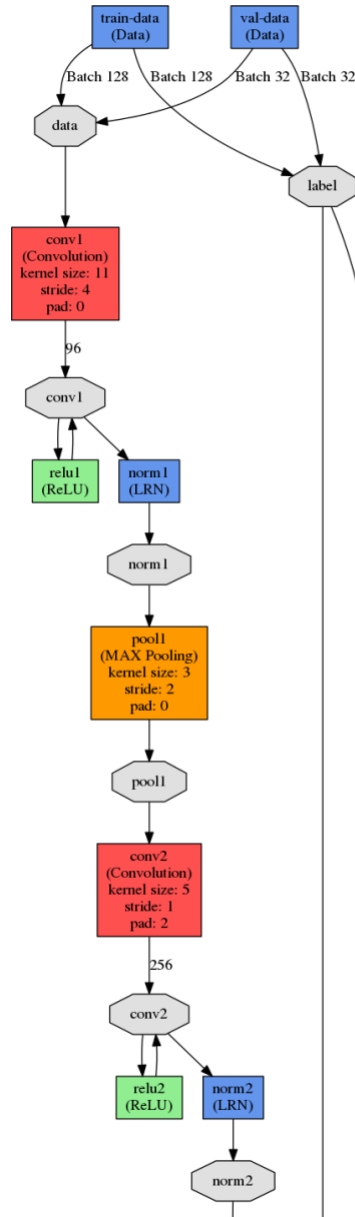
Yang, J., Nguyen, M. N., San, P. P., Li, X., & Krishnaswamy, S. (2015, July). Deep convolutional neural networks on multichannel time series for human activity recognition. In *Ijcai* (Vol. 15, pp. 3995-4001).

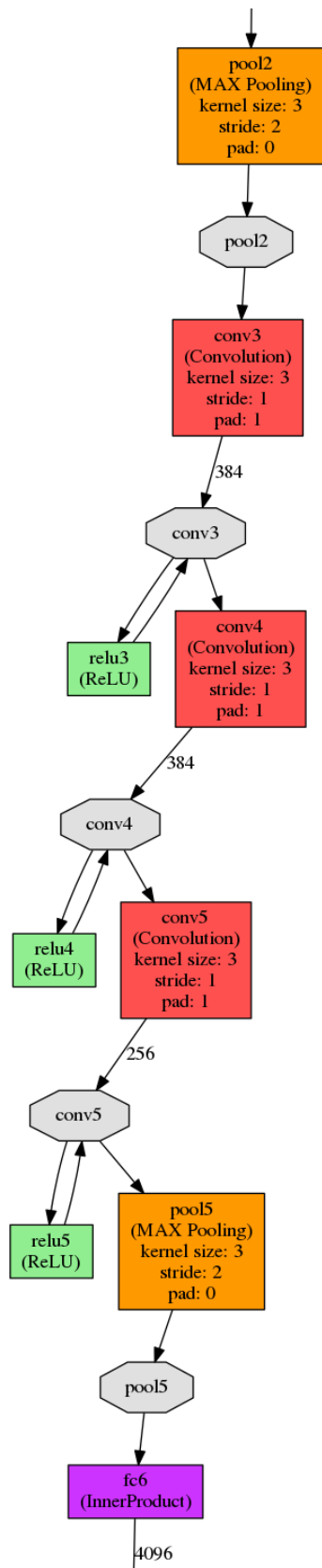
Yurtsever, U. (2019). *Bölütlenmiş histopatolojik görüntüler üzerinde derin öğrenme yöntemiyle kolon kanseri tespiti*. Sakarya Üniversitesi / Fen Bilimleri Enstitüsü, Sakarya.

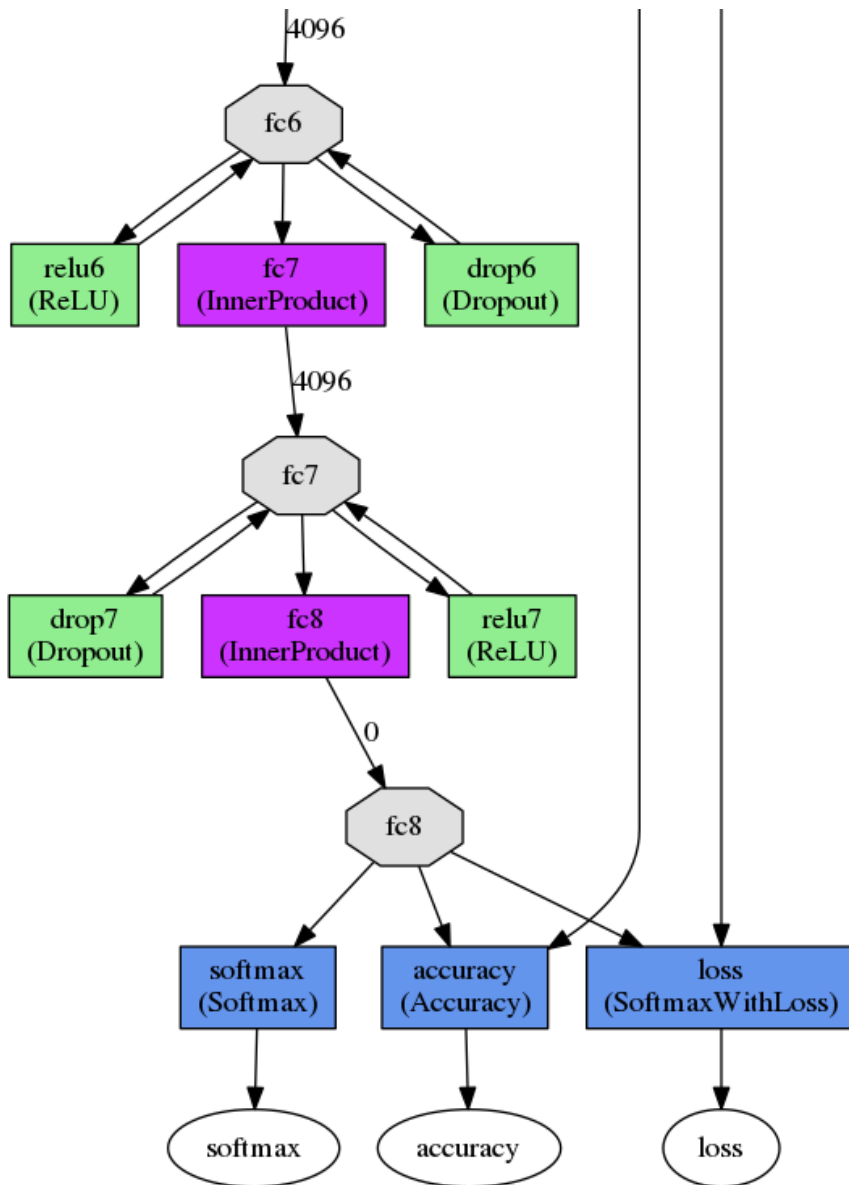
Zou, Y., Li, L., Wang, Y., Yu, J., Li, Y., & Deng, W. J. (2015, July). Classifying digestive organs in wireless capsule endoscopy images based on deep convolutional neural network. In 2015 IEEE International Conference on Digital Signal Processing (DSP) (pp. 1274-1278). IEEE.

# EKLER

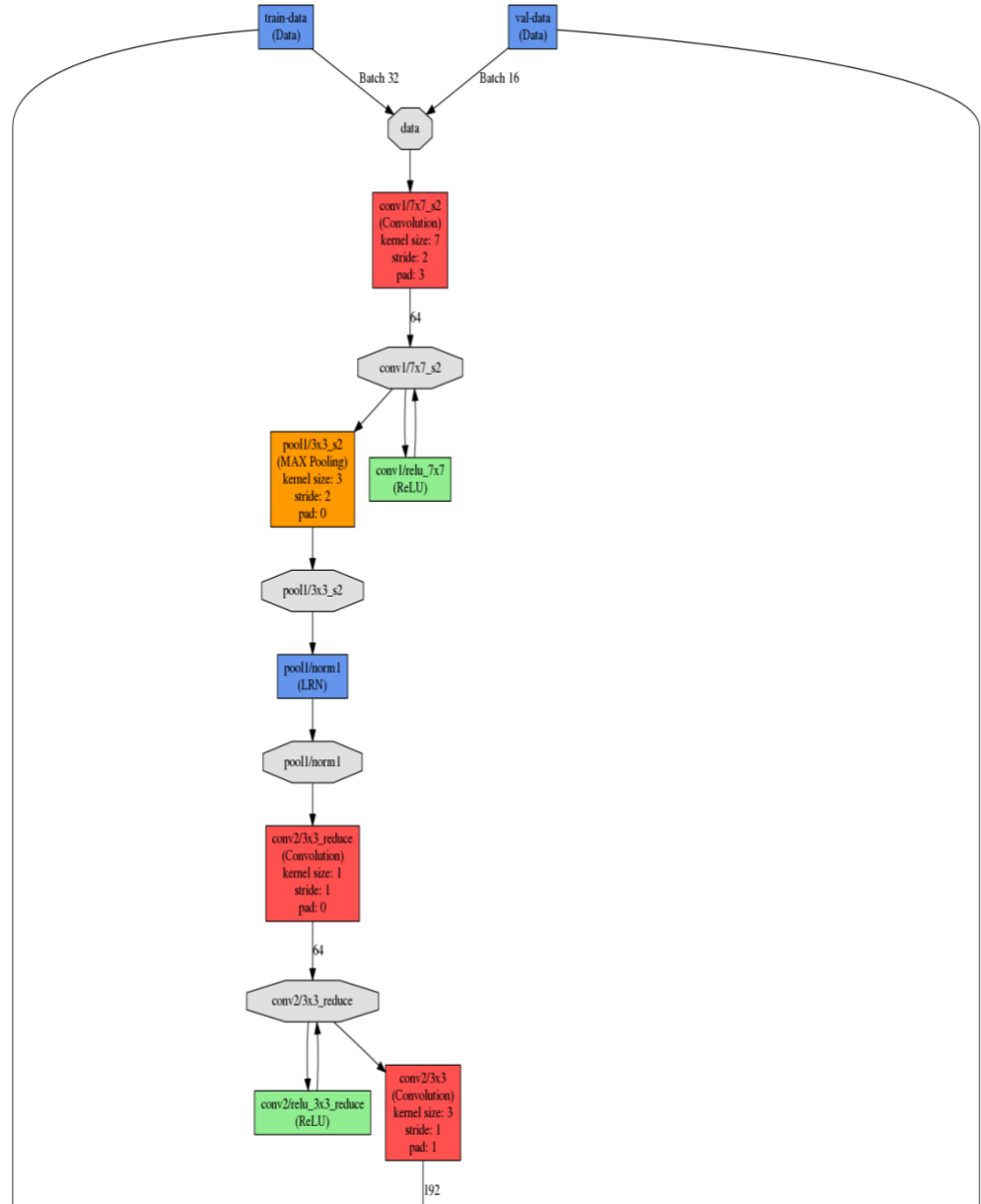
EK-1 AlexNet derin öğrenme modeli

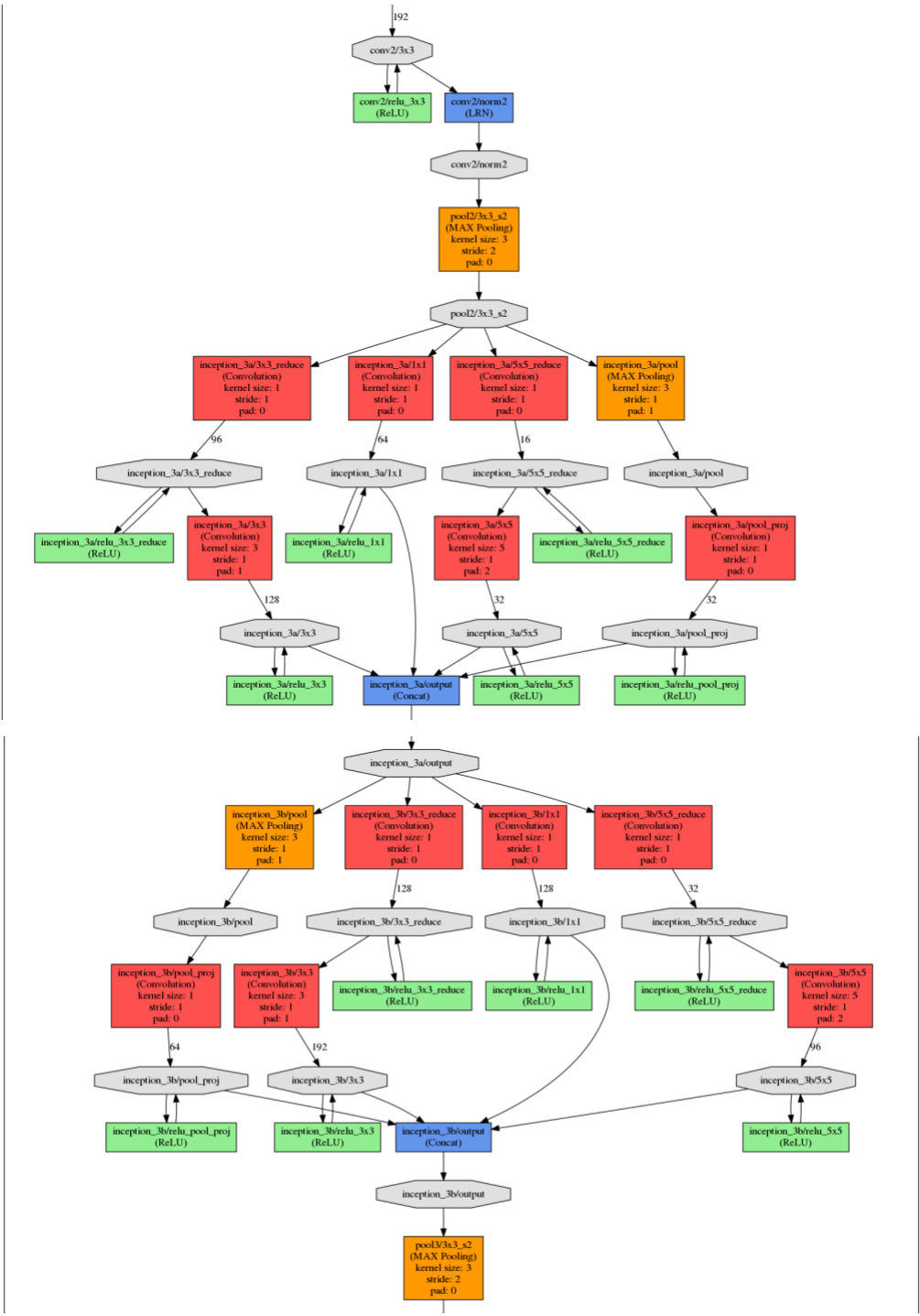


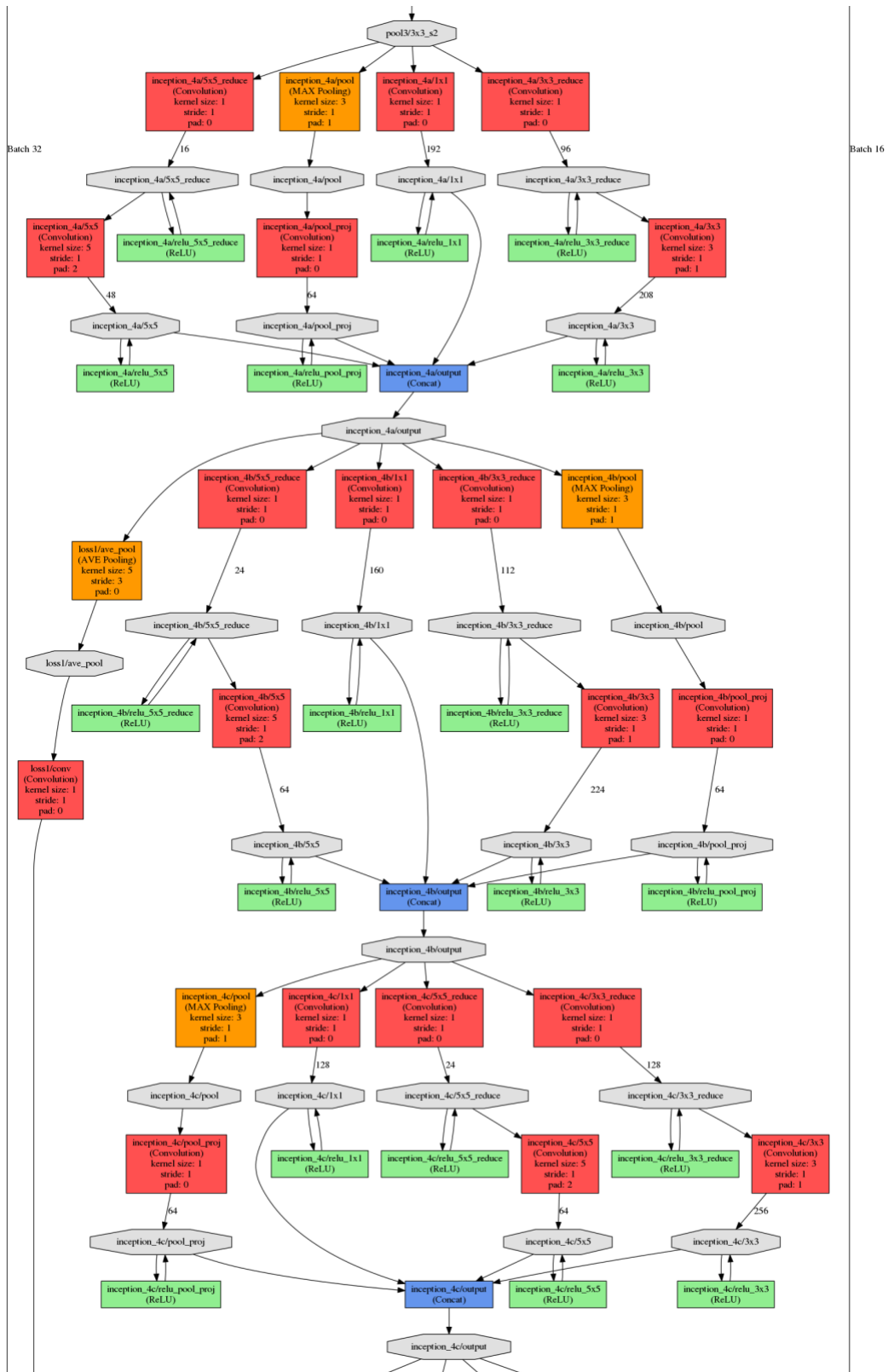




## EK-2 GoogLeNet derin öğrenme modeli mimarisi

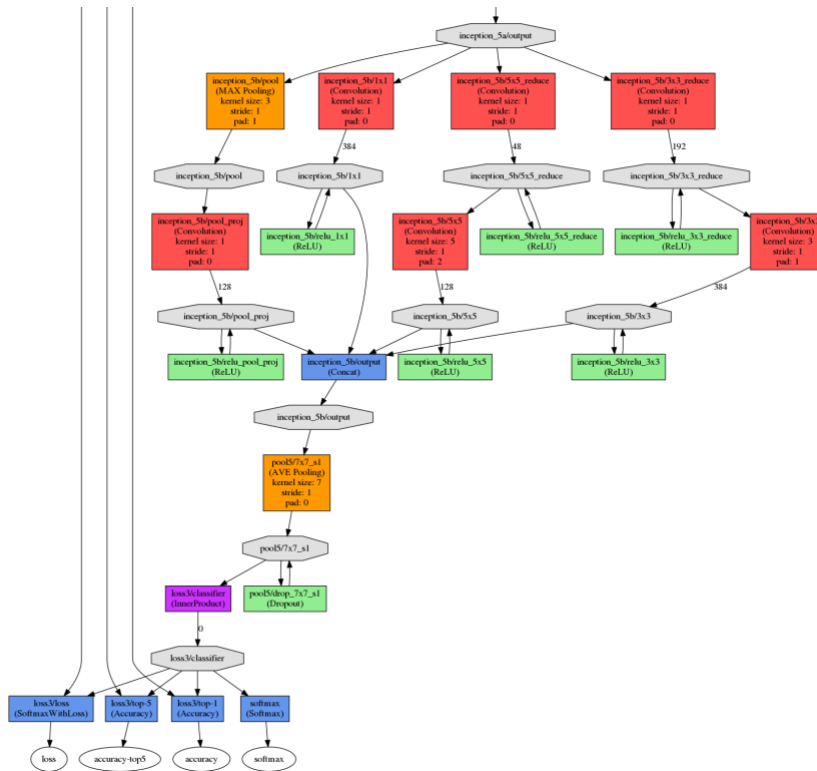










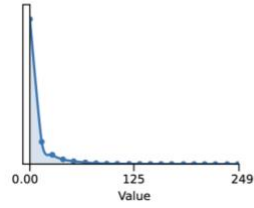


### EK-3 AlexNet ile eğitilmiş modelin test sonucu

Tanım	İstatistik	Görselleştirme
Veri Activation	<p><b>Data shape:</b> [ 3 227 227]  <b>Mean:</b> -22.368935  <b>Std deviation:</b> 39.97535</p>	
conv1 Weights (Convolution layer) 34,944 learned parameters	<p><b>Data shape:</b> [96 3 11 11]  <b>Mean:</b> -7.777329e-05  <b>Std deviation:</b> 0.011250892</p>	

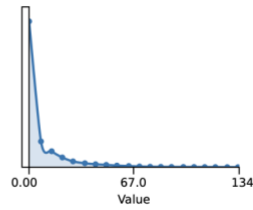
**conv1**  
Activation

**Data shape:** [96 55 55]  
**Mean:** 9.039242  
**Std deviation:** 17.63301



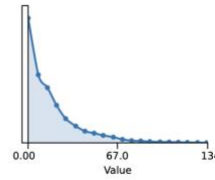
**norm1**  
Activation

**Data shape:** [96 55 55]  
**Mean:** 8.4103155  
**Std deviation:** 15.271057



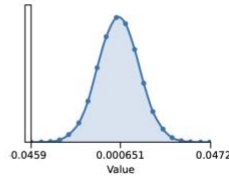
**pool1**  
Activation

**Data shape:** [96 27 27]  
**Mean:** 18.278778  
**Std deviation:** 19.85657



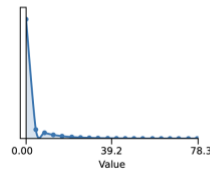
**conv2**  
Weights (Convolution layer)  
307,456 learned parameters

**Data shape:** [256 48 5 5]  
**Mean:** -0.00034594053  
**Std deviation:** 0.010241163



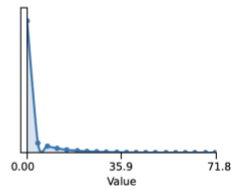
**conv2**  
Activation

**Data shape:** [256 27 27]  
**Mean:** 2.3917944  
**Std deviation:** 6.014185



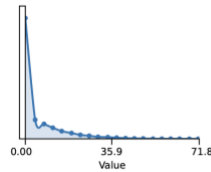
**norm2**  
Activation

**Data shape:** [256 27 27]  
**Mean:** 2.3694775  
**Std deviation:** 5.9191346



**pool2**  
Activation

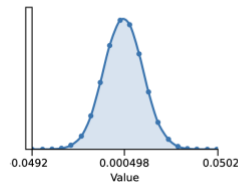
**Data shape:** [256 13 13]  
**Mean:** 5.5203648  
**Std deviation:** 8.982546



**conv3**

Weights (Convolution layer)  
885,120 learned parameters

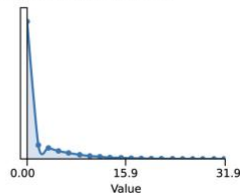
**Data shape:** [384 256 3 3]  
**Mean:** -0.000134227  
**Std deviation:** 0.010052162



**conv3**

Activation

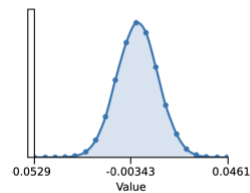
**Data shape:** [384 13 13]  
**Mean:** 1.8083082  
**Std deviation:** 3.575287



**conv4**

Weights (Convolution layer)  
663,936 learned parameters

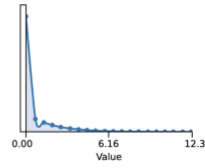
**Data shape:** [384 192 3 3]  
**Mean:** -0.00012327243  
**Std deviation:** 0.010051482



### conv4

Activation

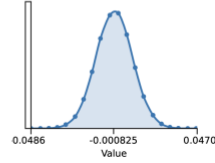
**Data shape:** [384 13 13]  
**Mean:** 0.65870166  
**Std deviation:** 1.2885044



### conv5

Weights (*Convolution layer*)  
442,624 learned parameters

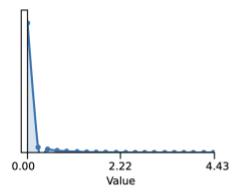
**Data shape:** [256 192 3 3]  
**Mean:** -0.0005277033  
**Std deviation:** 0.010103202



### conv5

Activation

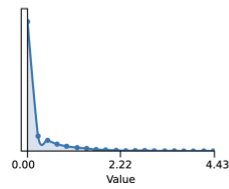
**Data shape:** [256 13 13]  
**Mean:** 0.07845267  
**Std deviation:** 0.2730345



### pool5

Activation

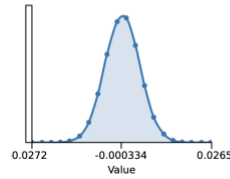
**Data shape:** [256 6 6]  
**Mean:** 0.23847978  
**Std deviation:** 0.49210525



### fc6

Weights (*InnerProduct layer*)  
37,752,832 learned parameters

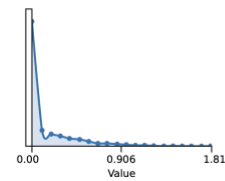
**Data shape:** [4096 9216]  
**Mean:** -3.3038472e-05  
**Std deviation:** 0.004996552



### fc6

Activation

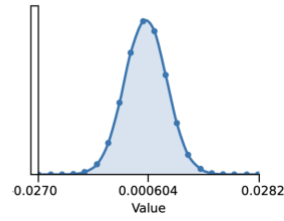
**Data shape:** [4096]  
**Mean:** 0.14660437  
**Std deviation:** 0.24808902



**fc7**

Weights (*InnerProduct layer*)  
16,781,312 learned parameters

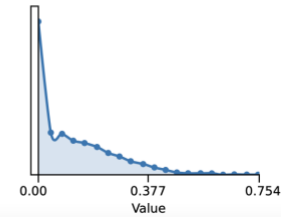
**Data shape:** [4096 4096]  
**Mean:** -1.8873347e-05  
**Std deviation:** 0.005000188



**fc7**

Activation

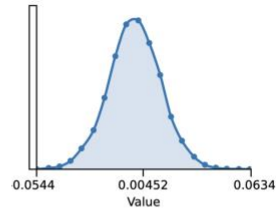
**Data shape:** [4096]  
**Mean:** 0.12128945  
**Std deviation:** 0.12928557



**fc8**

Weights (*InnerProduct layer*)  
12,291 learned parameters

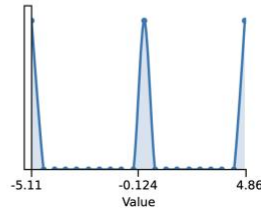
**Data shape:** [ 3 4096]  
**Mean:** -2.3828043e-05  
**Std deviation:** 0.013851054



**fc8**

Activation

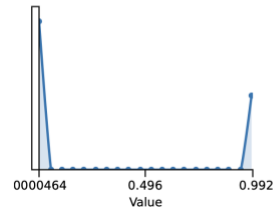
**Data shape:** [3]  
**Mean:** -0.059980232  
**Std deviation:** 4.0708237



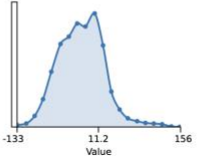

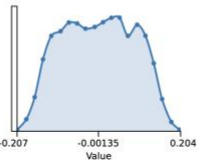

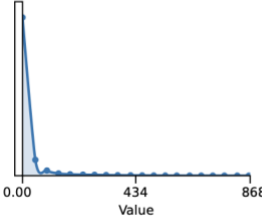
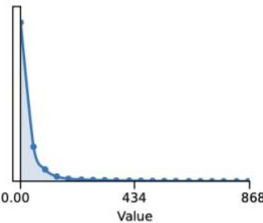
**softmax**

Activation

**Data shape:** [3]  
**Mean:** 0.3333333  
**Std deviation:** 0.46556833



## EK-4 GoogLeNet eğitilmiş modelin test sonucu

Description	Statistics	Visualization
<b>data</b> Activation	<b>Data shape:</b> [ 3 224 224] <b>Mean:</b> -17.895254 <b>Std deviation:</b> 38.778965  A histogram showing the distribution of activation values for the 'data' layer. The x-axis is labeled 'Value' and ranges from -133 to 156, with a tick mark at 11.2. The y-axis represents frequency. The distribution is roughly bell-shaped and centered around 11.2.	 A grayscale chest X-ray image showing the ribcage and lungs. The letter 'R' is visible in the top left corner, indicating the right side of the patient.
<b>conv1/7x7_s2</b> Weights (Convolution layer) 9,472 learned parameters	<b>Data shape:</b> [64 3 7 7] <b>Mean:</b> -0.002052563 <b>Std deviation:</b> 0.08566093  A histogram showing the distribution of weights for the 'conv1/7x7_s2' layer. The x-axis is labeled 'Value' and ranges from -0.207 to 0.204, with a tick mark at -0.00135. The y-axis represents frequency. The distribution is roughly bell-shaped and centered around -0.00135.	 A grid of small images showing the learned weights for the 'conv1/7x7_s2' layer. Each small image represents a 7x7 kernel filter.
<b>conv1/7x7_s2</b> Activation	<b>Data shape:</b> [ 64 112 112] <b>Mean:</b> 24.574303 <b>Std deviation:</b> 58.068806  A histogram showing the distribution of activation values for the 'conv1/7x7_s2' layer. The x-axis is labeled 'Value' and ranges from 0.00 to 868, with a tick mark at 434. The y-axis represents frequency. The distribution is highly skewed towards zero, with a sharp peak at the origin and a long tail extending to 868.	
<b>pool1/3x3_s2</b> Activation	<b>Data shape:</b> [64 56 56] <b>Mean:</b> 41.080765 <b>Std deviation:</b> 70.612015  A histogram showing the distribution of activation values for the 'pool1/3x3_s2' layer. The x-axis is labeled 'Value' and ranges from 0.00 to 868, with a tick mark at 434. The y-axis represents frequency. The distribution is highly skewed towards zero, with a sharp peak at the origin and a long tail extending to 868.	

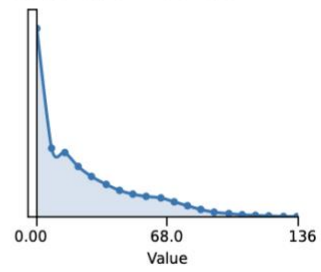
## pool1/norm1

Activation

**Data shape:** [64 56 56]

**Mean:** 24.645573

**Std deviation:** 25.619362



---

## conv2/3x3\_reduce

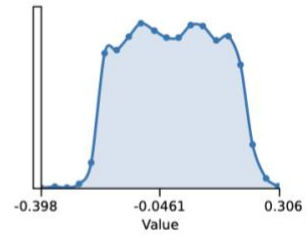
Weights (*Convolution layer*)

4,160 learned parameters

**Data shape:** [64 64 1 1]

**Mean:** -0.008278498

**Std deviation:** 0.12602821



---

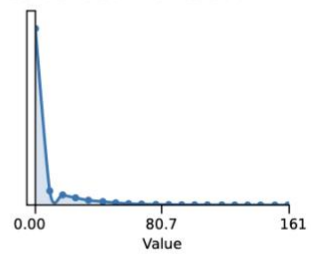
## conv2/3x3\_reduce

Activation

**Data shape:** [64 56 56]

**Mean:** 6.4850993

**Std deviation:** 15.168826

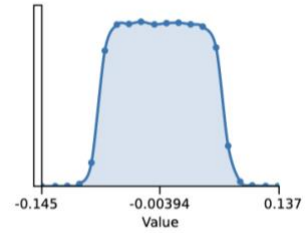




### conv2/3x3

Weights (*Convolution layer*)  
110,784 learned parameters

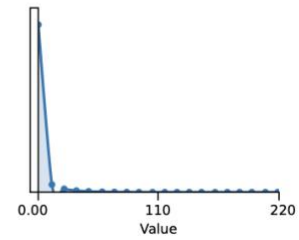
**Data shape:** [192 64 3 3]  
**Mean:** -0.0029397216  
**Std deviation:** 0.04195285



### conv2/3x3

Activation

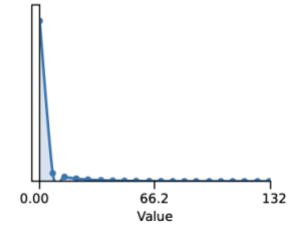
**Data shape:** [192 56 56]  
**Mean:** 2.5757284  
**Std deviation:** 8.505596



### conv2/norm2

Activation

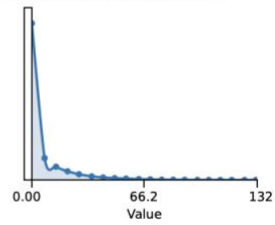
**Data shape:** [192 56 56]  
**Mean:** 2.5027902  
**Std deviation:** 8.002811



### pool2/3x3\_s2

Activation

**Data shape:** [192 28 28]  
**Mean:** 6.8322234  
**Std deviation:** 13.0830555

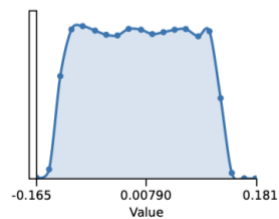


### inception\_3a/1x1

Weights (*Convolution layer*)

12,352 learned parameters

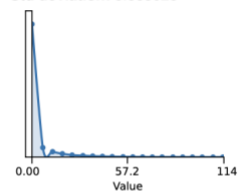
**Data shape:** [ 64 192 1 1]  
**Mean:** -0.0025707202  
**Std deviation:** 0.07202534



### inception\_3a/1x1

Activation

**Data shape:** [64 28 28]  
**Mean:** 3.2978923  
**Std deviation:** 8.883028

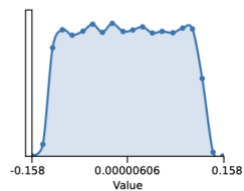


### inception\_3a/3x3\_reduce

Weights (*Convolution layer*)

18,528 learned parameters

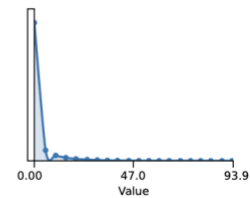
**Data shape:** [ 96 192 1 1]  
**Mean:** -0.0023974087  
**Std deviation:** 0.07129951



### inception\_3a/3x3\_reduce

Activation

**Data shape:** [96 28 28]  
**Mean:** 2.3064036  
**Std deviation:** 6.159713

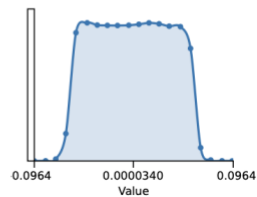


### inception\_3a/3x3

Weights (*Convolution layer*)

110,720 learned parameters

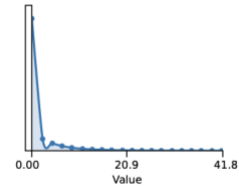
**Data shape:** [128 96 3 3]  
**Mean:** -0.0011211493  
**Std deviation:** 0.033865724



### inception\_3a/3x3

Activation

**Data shape:** [128 28 28]  
**Mean:** 1.5562347  
**Std deviation:** 3.7397554

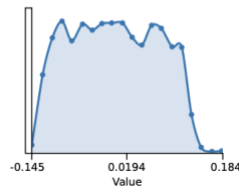


---

### inception\_3a/5x5\_reduce

Weights (Convolution layer)  
3,088 learned parameters

**Data shape:** [ 16 192 1 1]  
**Mean:** -0.0014082548  
**Std deviation:** 0.071288414

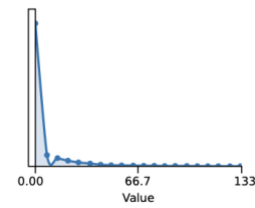


---

### inception\_3a/5x5\_reduce

Activation

**Data shape:** [16 28 28]  
**Mean:** 5.981681  
**Std deviation:** 14.548469

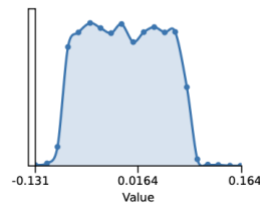


---

### inception\_3a/5x5

Weights (Convolution layer)  
12,832 learned parameters

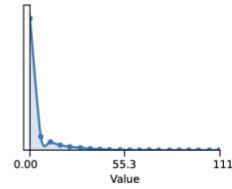
**Data shape:** [32 16 5 5]  
**Mean:** -0.0015073745  
**Std deviation:** 0.05043197



### inception\_3a/5x5

Activation

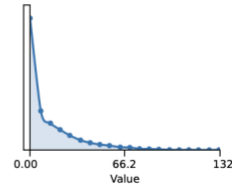
**Data shape:** [32 28 28]  
**Mean:** 4.2742486  
**Std deviation:** 9.519524



### inception\_3a/pool

Activation

**Data shape:** [192 28 28]  
**Mean:** 13.935849  
**Std deviation:** 18.22946

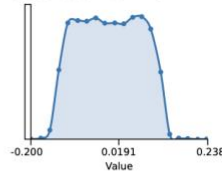


### inception\_3a/pool\_proj

Weights (Convolution layer)

6,176 learned parameters

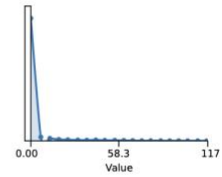
**Data shape:** [ 32 192 1 1]  
**Mean:** -0.00348071  
**Std deviation:** 0.07221921



### inception\_3a/pool\_proj

Activation

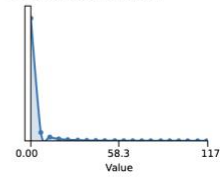
**Data shape:** [32 28 28]  
**Mean:** 3.1810956  
**Std deviation:** 11.563463



### inception\_3a/output

Activation

**Data shape:** [256 28 28]  
**Mean:** 2.5345085  
**Std deviation:** 7.471051

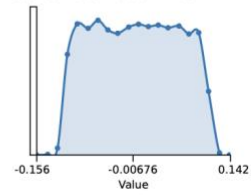


### inception\_3b/1x1

Weights (Convolution layer)

32,896 learned parameters

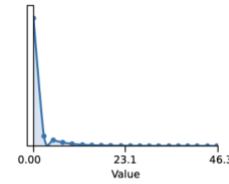
**Data shape:** [128 256 1 1]  
**Mean:** -0.0024240543  
**Std deviation:** 0.061722912



### inception\_3b/1x1

Activation

**Data shape:** [128 28 28]  
**Mean:** 1.2737334  
**Std deviation:** 3.2854455



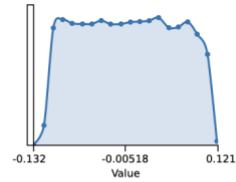
---

### inception\_3b/3x3\_reduce

Weights (Convolution layer)

32,896 learned parameters

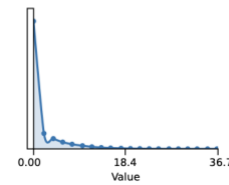
**Data shape:** [128 256 1 1]  
**Mean:** -0.001694982  
**Std deviation:** 0.061562456



### inception\_3b/3x3\_reduce

Activation

**Data shape:** [128 28 28]  
**Mean:** 1.7939389  
**Std deviation:** 3.5414712

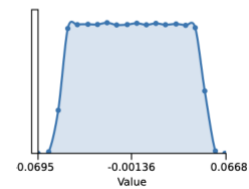


### inception\_3b/3x3

Weights (Convolution layer)

221,376 learned parameters

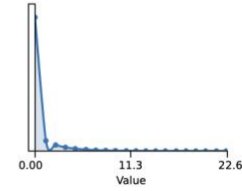
**Data shape:** [192 128 3 3]  
**Mean:** -0.0008129702  
**Std deviation:** 0.029185113



### inception\_3b/3x3

Activation

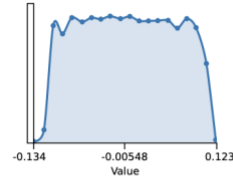
**Data shape:** [192 28 28]  
**Mean:** 0.6129294  
**Std deviation:** 1.564551



### inception\_3b/5x5\_reduce

Weights (Convolution layer)  
8,224 learned parameters

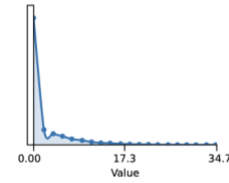
**Data shape:** [32 256 1 1]  
**Mean:** -0.0017812442  
**Std deviation:** 0.061924648



### inception\_3b/5x5\_reduce

Activation

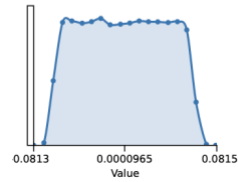
**Data shape:** [32 28 28]  
**Mean:** 2.1111677  
**Std deviation:** 3.9460719



### inception\_3b/5x5

Weights (Convolution layer)  
76,896 learned parameters

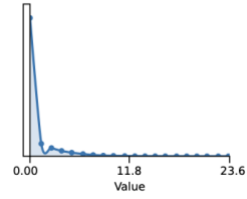
**Data shape:** [96 32 5 5]  
**Mean:** -0.0008932093  
**Std deviation:** 0.034998547



### inception\_3b/5x5

Activation

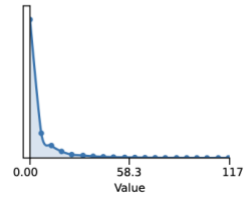
**Data shape:** [96 28 28]  
**Mean:** 0.8184957  
**Std deviation:** 1.8634665



### inception\_3b/pool

Activation

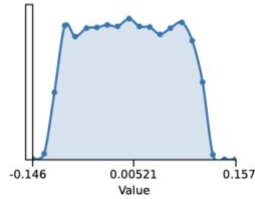
**Data shape:** [256 28 28]  
**Mean:** 5.8506265  
**Std deviation:** 11.181611



### inception\_3b/pool\_proj

Weights (*Convolution layer*)  
16,448 learned parameters

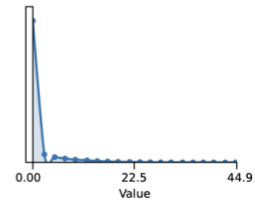
**Data shape:** [ 64 256 1 1]  
**Mean:** -0.0022668487  
**Std deviation:** 0.061968714



### inception\_3b/pool\_proj

Activation

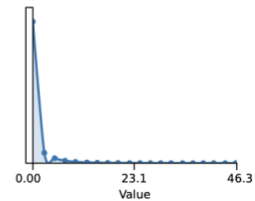
**Data shape:** [64 28 28]  
**Mean:** 1.3437214  
**Std deviation:** 3.6932013



### inception\_3b/output

Activation

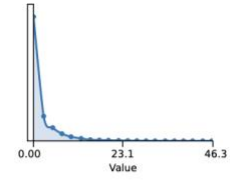
**Data shape:** [480 28 28]  
**Mean:** 0.927696  
**Std deviation:** 2.5431721



### pool3/3x3\_s2

Activation

**Data shape:** [480 14 14]  
**Mean:** 2.3226414  
**Std deviation:** 3.9730065

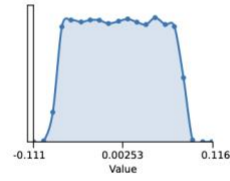


### inception\_4a/1x1

Weights (Convolution layer)

92,352 learned parameters

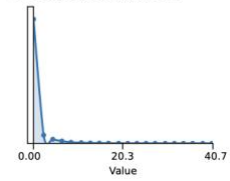
**Data shape:** [192 480 1 1]  
**Mean:** -0.0015967966  
**Std deviation:** 0.0452065



### inception\_4a/1x1

Activation

**Data shape:** [192 14 14]  
**Mean:** 0.7899622  
**Std deviation:** 2.2707515

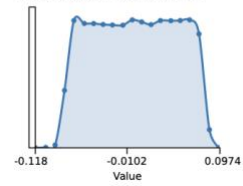


### inception\_4a/3x3\_reduce

Weights (Convolution layer)

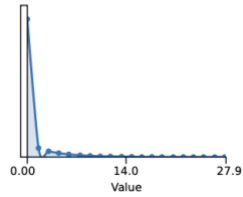
46,176 learned parameters

**Data shape:** [ 96 480 1 1]  
**Mean:** -0.001561822  
**Std deviation:** 0.045422442

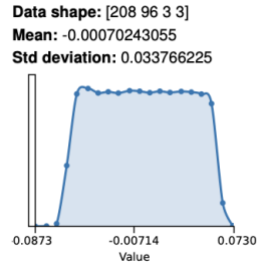




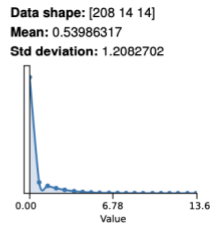
**inception\_4a/3x3\_reduce**  
Activation  
Data shape: [96 14 14]  
Mean: 0.8923662  
Std deviation: 2.4299374



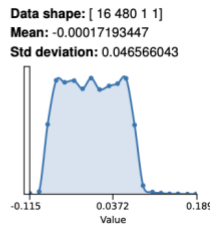
**inception\_4a/3x3**  
Weights (Convolution layer)  
179,920 learned parameters



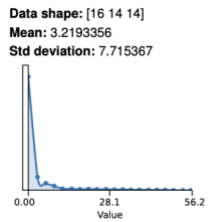
**inception\_4a/3x3**  
Activation



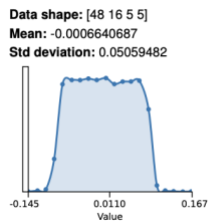
**inception\_4a/5x5\_reduce**  
Weights (Convolution layer)  
7,696 learned parameters



**inception\_4a/5x5\_reduce**  
Activation



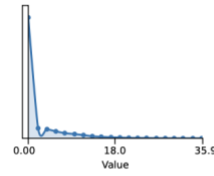
**inception\_4a/5x5**  
Weights (Convolution layer)  
19,248 learned parameters



### inception\_4a/5x5

Activation

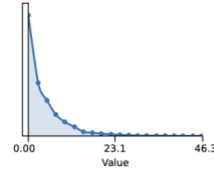
Data shape: [48 14 14]  
Mean: 2.1662664  
Std deviation: 4.326077



### inception\_4a/pool

Activation

Data shape: [480 14 14]  
Mean: 4.450139  
Std deviation: 5.340689

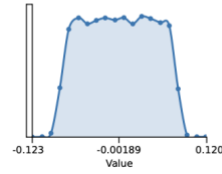


### inception\_4a/pool\_proj

Weights (Convolution layer)

30,784 learned parameters

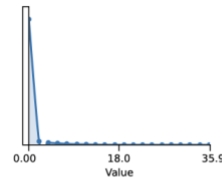
Data shape: [ 64 480 1 1]  
Mean: -0.001913647  
Std deviation: 0.045173433



### inception\_4a/pool\_proj

Activation

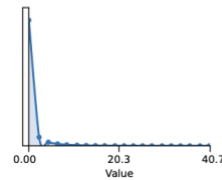
Data shape: [64 14 14]  
Mean: 0.584539  
Std deviation: 2.5541313



### inception\_4a/output

Activation

Data shape: [512 14 14]  
Mean: 0.79170996  
Std deviation: 2.3032472



### inception\_4b/1x1

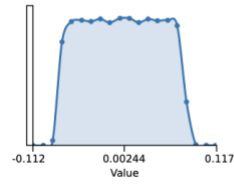
Weights (Convolution layer)

82,080 learned parameters

Data shape: [160 512 1 1]

Mean: -0.00061898713

Std deviation: 0.04366124



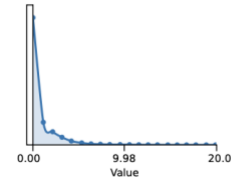
### inception\_4b/1x1

Activation

Data shape: [160 14 14]

Mean: 0.8849339

Std deviation: 1.5879064



### inception\_4b/3x3\_reduce

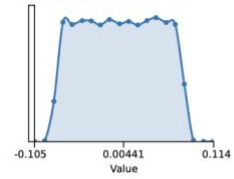
Weights (Convolution layer)

57,456 learned parameters

Data shape: [112 512 1 1]

Mean: -0.00019142541

Std deviation: 0.04385449



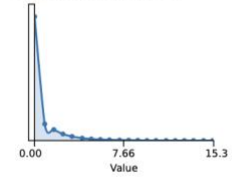
### inception\_4b/3x3\_reduce

Activation

Data shape: [112 14 14]

Mean: 0.7274809

Std deviation: 1.4613395



### inception\_4b/3x3

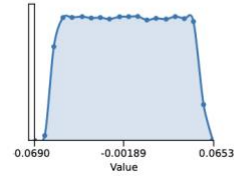
Weights (Convolution layer)

226,016 learned parameters

Data shape: [224 112 3 3]

Mean: -0.00037886773

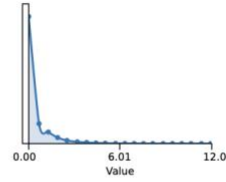
Std deviation: 0.031216636



### inception\_4b/3x3

Activation

**Data shape:** [224 14 14]  
**Mean:** 0.4760348  
**Std deviation:** 0.8939185

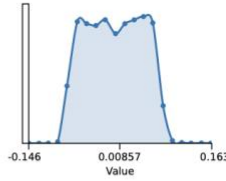


### inception\_4b/5x5\_reduce

Weights (Convolution layer)

12,312 learned parameters

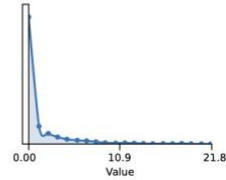
**Data shape:** [24 512 1 1]  
**Mean:** 4.9236503e-05  
**Std deviation:** 0.04453625



### inception\_4b/5x5\_reduce

Activation

**Data shape:** [24 14 14]  
**Mean:** 1.4782174  
**Std deviation:** 2.969285

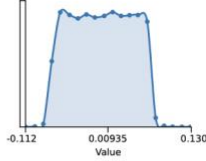


### inception\_4b/5x5

Weights (Convolution layer)

38,464 learned parameters

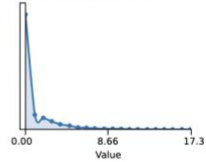
**Data shape:** [64 24 5 5]  
**Mean:** -0.00025531882  
**Std deviation:** 0.040950663



### inception\_4b/5x5

Activation

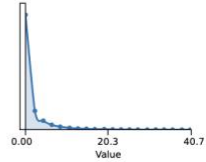
**Data shape:** [64 14 14]  
**Mean:** 1.0092539  
**Std deviation:** 1.8832924



### inception\_4b/pool

Activation

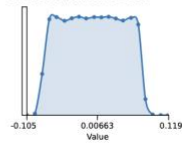
**Data shape:** [512 14 14]  
**Mean:** 1.7849181  
**Std deviation:** 3.502818



**inception\_4b/pool\_proj**

Weights (Convolution layer)  
32,832 learned parameters

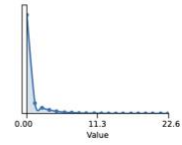
Data shape: [64 512 1 1]  
Mean: -0.0008131221  
Std deviation: 0.043950547



**inception\_4b/pool\_proj**

Activation

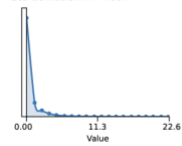
Data shape: [64 14 14]  
Mean: 0.7612627  
Std deviation: 1.7633915



**inception\_4b/output**

Activation

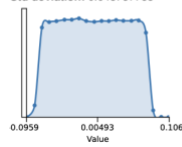
Data shape: [512 14 14]  
Mean: 0.70612156  
Std deviation: 1.4193717



**inception\_4c/1x1**

Weights (Convolution layer)  
65,664 learned parameters

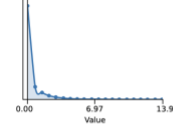
Data shape: [128 512 1 1]  
Mean: -0.00061838265  
Std deviation: 0.043757755



**inception\_4c/1x1**

Activation

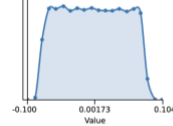
Data shape: [128 14 14]  
Mean: 0.53964937  
Std deviation: 1.1274307



**inception\_4c/3x3\_reduce**

Weights (Convolution layer)  
65,664 learned parameters

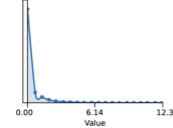
Data shape: [128 512 1 1]  
Mean: -0.0010266156  
Std deviation: 0.04394282



**inception\_4c/3x3\_reduce**

Activation

Data shape: [128 14 14]  
Mean: 0.40143406  
Std deviation: 0.9199226

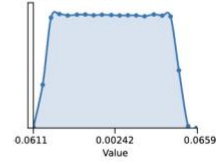


### inception\_4c/3x3

Weights (Convolution layer)

295,168 learned parameters

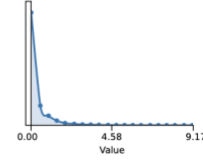
Data shape: [256 128 3 3]  
Mean: -0.00032584925  
Std deviation: 0.029202098



### inception\_4c/3x3

Activation

Data shape: [256 14 14]  
Mean: 0.36076802  
Std deviation: 0.67842776

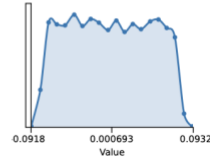


### inception\_4c/5x5\_reduce

Weights (Convolution layer)

12,312 learned parameters

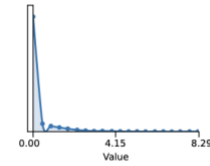
Data shape: [ 24 512 1 1]  
Mean: -0.0013061041  
Std deviation: 0.04397074



### inception\_4c/5x5\_reduce

Activation

Data shape: [24 14 14]  
Mean: 0.31569085  
Std deviation: 0.8262498

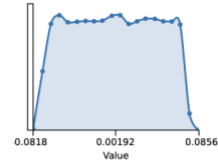


### inception\_4c/5x5

Weights (Convolution layer)

38,464 learned parameters

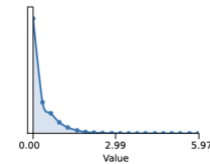
Data shape: [64 24 5 5]  
Mean: 0.00014289109  
Std deviation: 0.04021398



### inception\_4c/5x5

Activation

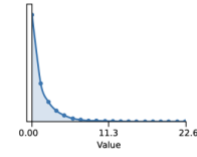
Data shape: [64 14 14]  
Mean: 0.34356257  
Std deviation: 0.4798329



### inception\_4c/pool

Activation

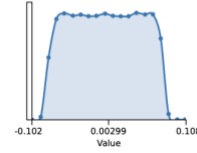
**Data shape:** [512 14 14]  
**Mean:** 1.5033742  
**Std deviation:** 2.0151002



### inception\_4c/pool\_proj

Weights (Convolution layer)  
32,832 learned parameters

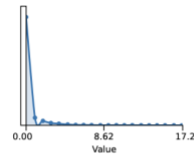
**Data shape:** [ 64 512 1 1]  
**Mean:** -0.0009249033  
**Std deviation:** 0.0438277



### inception\_4c/pool\_proj

Activation

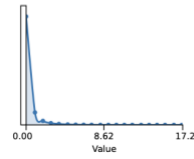
**Data shape:** [64 14 14]  
**Mean:** 0.47112805  
**Std deviation:** 1.4040155



### inception\_4c/output

Activation

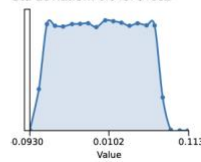
**Data shape:** [512 14 14]  
**Mean:** 0.41713268  
**Std deviation:** 0.9107561



### inception\_4d/1x1

Weights (Convolution layer)  
57,456 learned parameters

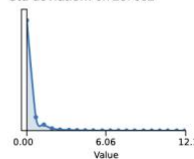
**Data shape:** [112 512 1 1]  
**Mean:** -0.00043029975  
**Std deviation:** 0.043761052

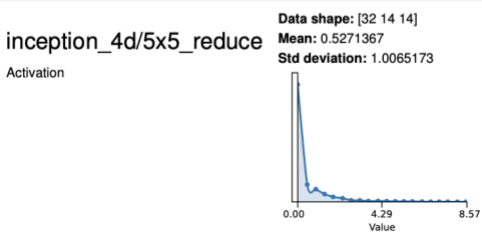
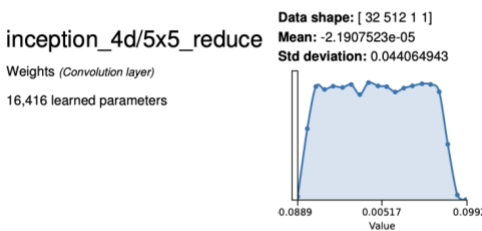
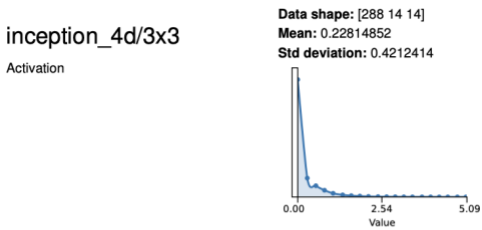
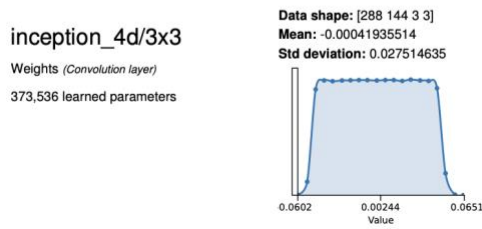
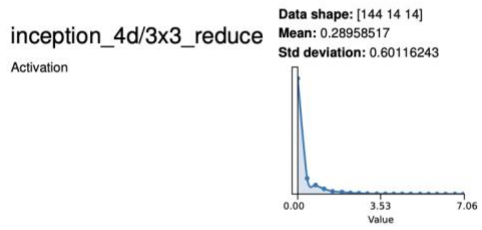
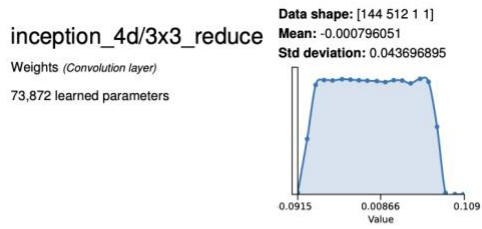


### inception\_4d/1x1

Activation

**Data shape:** [112 14 14]  
**Mean:** 0.32874033  
**Std deviation:** 0.7237652





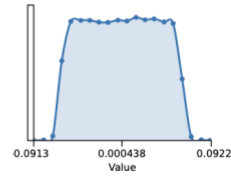


### inception\_4d/5x5

Weights (Convolution layer)

51,264 learned parameters

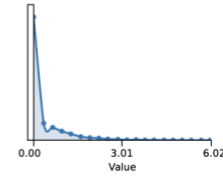
Data shape: [64 32 5 5]  
Mean: -0.00026910345  
Std deviation: 0.035205964



### inception\_4d/5x5

Activation

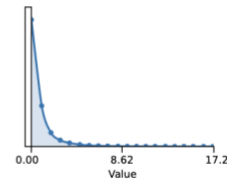
Data shape: [64 14 14]  
Mean: 0.36549985  
Std deviation: 0.66726404



### inception\_4d/pool

Activation

Data shape: [512 14 14]  
Mean: 0.8882514  
Std deviation: 1.298583

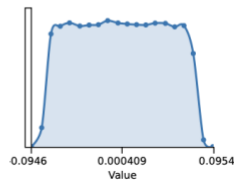


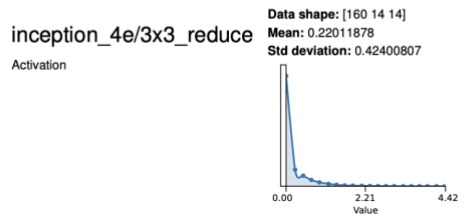
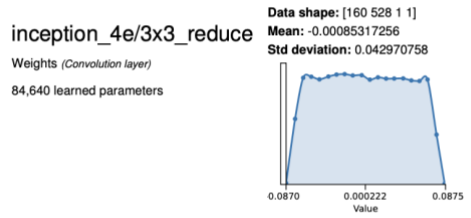
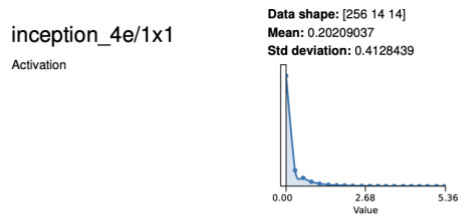
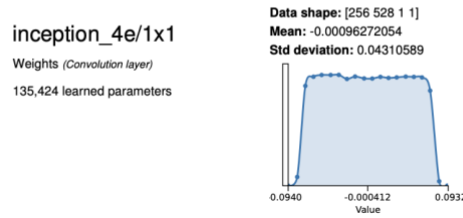
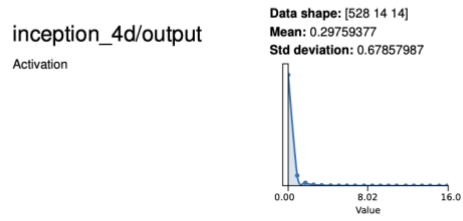
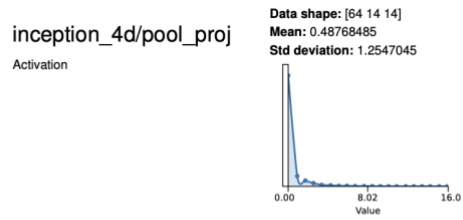
### inception\_4d/pool\_proj

Weights (Convolution layer)

32,832 learned parameters

Data shape: [ 64 512 1 1]  
Mean: -0.00084550097  
Std deviation: 0.04374968



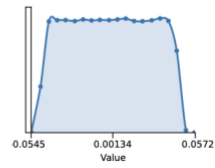


### inception\_4e/3x3

Weights (Convolution layer)

461,120 learned parameters

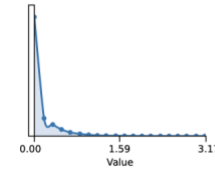
**Data shape:** [320 160 3 3]  
**Mean:** -0.0004705041  
**Std deviation:** 0.026061296



### inception\_4e/3x3

Activation

**Data shape:** [320 14 14]  
**Mean:** 0.13491228  
**Std deviation:** 0.24907498

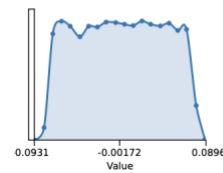


### inception\_4e/5x5\_reduce

Weights (Convolution layer)

16,928 learned parameters

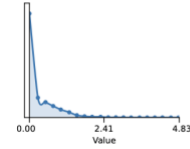
**Data shape:** [32 528 1 1]  
**Mean:** -0.00051787455  
**Std deviation:** 0.043011762



### inception\_4e/5x5\_reduce

Activation

**Data shape:** [32 14 14]  
**Mean:** 0.32649592  
**Std deviation:** 0.4924225

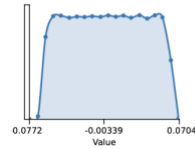


### inception\_4e/5x5

Weights (Convolution layer)

102,528 learned parameters

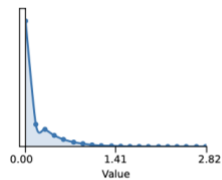
**Data shape:** [128 32 5 5]  
**Mean:** -0.0006365105  
**Std deviation:** 0.034992706



### inception\_4e/5x5

Activation

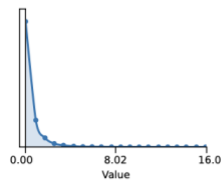
**Data shape:** [128 14 14]  
**Mean:** 0.17050566  
**Std deviation:** 0.27288306



### inception\_4e/pool

Activation

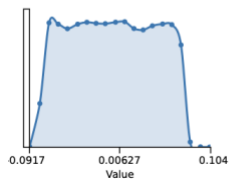
**Data shape:** [528 14 14]  
**Mean:** 0.59627265  
**Std deviation:** 0.9529171



### inception\_4e/pool\_proj

Weights (Convolution layer)  
67,712 learned parameters

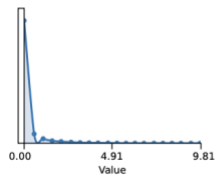
**Data shape:** [128 528 1 1]  
**Mean:** -0.0011263796  
**Std deviation:** 0.043213837



### inception\_4e/pool\_proj

Activation

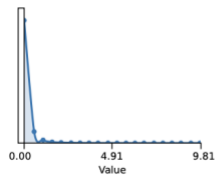
**Data shape:** [128 14 14]  
**Mean:** 0.25949958  
**Std deviation:** 0.7174088



### inception\_4e/output

Activation

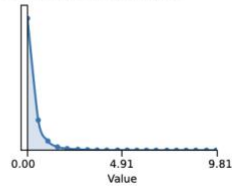
**Data shape:** [832 14 14]  
**Mean:** 0.18022564  
**Std deviation:** 0.4109248



### pool4/3x3\_s2

Activation

**Data shape:** [832 7 7]  
**Mean:** 0.36323142  
**Std deviation:** 0.5783168

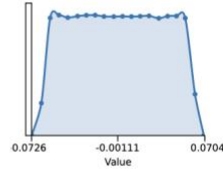


### inception\_5a/1x1

Weights (Convolution layer)

213,248 learned parameters

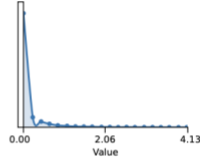
**Data shape:** [256 832 1 1]  
**Mean:** -0.0008922155  
**Std deviation:** 0.03427015



### inception\_5a/1x1

Activation

**Data shape:** [256 7 7]  
**Mean:** 0.13653573  
**Std deviation:** 0.35137543

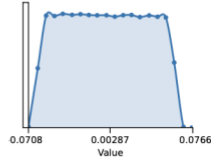


### inception\_5a/3x3\_reduce

Weights (Convolution layer)

133,280 learned parameters

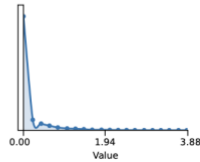
**Data shape:** [160 832 1 1]  
**Mean:** -0.0007951081  
**Std deviation:** 0.034298167



### inception\_5a/3x3\_reduce

Activation

**Data shape:** [160 7 7]  
**Mean:** 0.15690476  
**Std deviation:** 0.36884564

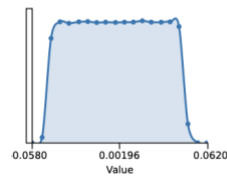


### inception\_5a/3x3

Weights (Convolution layer)

461,120 learned parameters

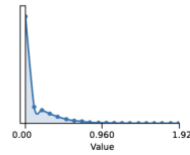
**Data shape:** [320 160 3 3]  
**Mean:** -0.0003928145  
**Std deviation:** 0.026063649



### inception\_5a/3x3

Activation

**Data shape:** [320 7 7]  
**Mean:** 0.12134089  
**Std deviation:** 0.19953889



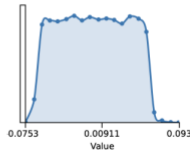
---

### inception\_5a/5x5\_reduce

Weights (Convolution layer)

26,656 learned parameters

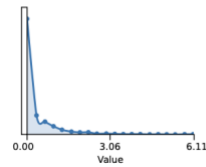
**Data shape:** [ 32 832 1 1]  
**Mean:** 2.9994337e-05  
**Std deviation:** 0.034715407



### inception\_5a/5x5\_reduce

Activation

**Data shape:** [32 7 7]  
**Mean:** 0.3800847  
**Std deviation:** 0.72519976



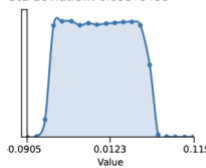
---

### inception\_5a/5x5

Weights (Convolution layer)

102,528 learned parameters

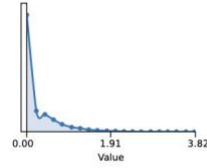
**Data shape:** [128 32 5 5]  
**Mean:** -0.0004929244  
**Std deviation:** 0.03515458



### inception\_5a/5x5

Activation

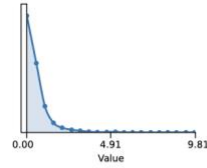
**Data shape:** [128 7 7]  
**Mean:** 0.26233685  
**Std deviation:** 0.40612432



### inception\_5a/pool

Activation

**Data shape:** [832 7 7]  
**Mean:** 0.6684669  
**Std deviation:** 0.7520578

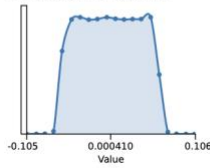


### inception\_5a/pool\_proj

Weights (Convolution layer)

106,624 learned parameters

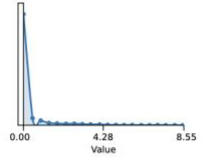
**Data shape:** [128 832 1 1]  
**Mean:** -0.0006880694  
**Std deviation:** 0.03454352



### inception\_5a/pool\_proj

Activation

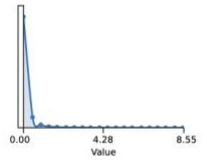
**Data shape:** [128 7 7]  
**Mean:** 0.34385258  
**Std deviation:** 0.9595086



### inception\_5a/output

Activation

**Data shape:** [832 7 7]  
**Mean:** 0.18194048  
**Std deviation:** 0.476856

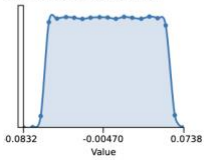


### inception\_5b/1x1

Weights (Convolution layer)

319,872 learned parameters

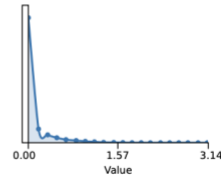
**Data shape:** [384 832 1 1]  
**Mean:** -0.0008401457  
**Std deviation:** 0.03427806



### inception\_5b/1x1

Activation

**Data shape:** [384 7 7]  
**Mean:** 0.11752018  
**Std deviation:** 0.25367314

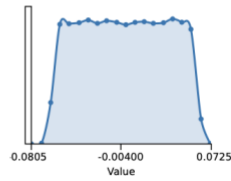


### inception\_5b/3x3\_reduce

Weights (Convolution layer)

159,936 learned parameters

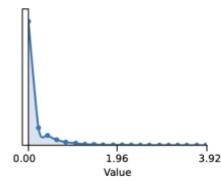
**Data shape:** [192 832 1 1]  
**Mean:** -0.00079563225  
**Std deviation:** 0.034287885



### inception\_5b/3x3\_reduce

Activation

**Data shape:** [192 7 7]  
**Mean:** 0.15973318  
**Std deviation:** 0.3280891

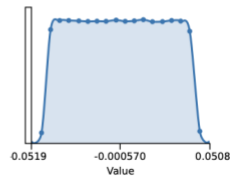


### inception\_5b/3x3

Weights (Convolution layer)

663,936 learned parameters

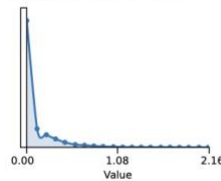
**Data shape:** [384 192 3 3]  
**Mean:** -0.0005784074  
**Std deviation:** 0.023808116



### inception\_5b/3x3

Activation

**Data shape:** [384 7 7]  
**Mean:** 0.11452977  
**Std deviation:** 0.21548146

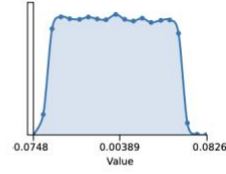




inception\_5b/5x5\_reduce

Weights (Convolution layer)  
39,984 learned parameters

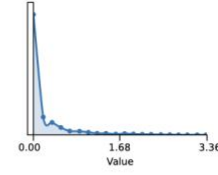
Data shape: [ 48 832 1 1]  
Mean: -0.00068781106  
Std deviation: 0.034367725



inception\_5b/5x5\_reduce

Activation

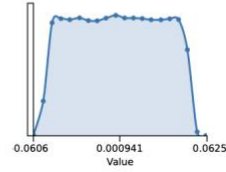
Data shape: [48 7 7]  
Mean: 0.22066309  
Std deviation: 0.42779827



inception\_5b/5x5

Weights (Convolution layer)  
153,728 learned parameters

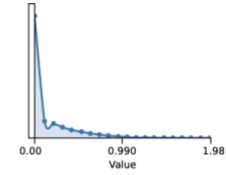
Data shape: [128 48 5 5]  
Mean: -0.0006025636  
Std deviation: 0.028545788



inception\_5b/5x5

Activation

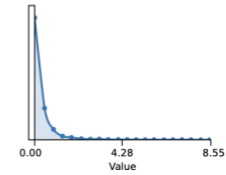
Data shape: [128 7 7]  
Mean: 0.15221946  
Std deviation: 0.24975577



inception\_5b/pool

Activation

Data shape: [832 7 7]  
Mean: 0.3794816  
Std deviation: 0.63577014

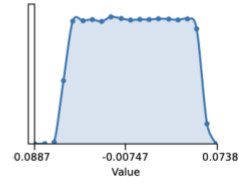


### inception\_5b/pool\_proj

Weights (Convolution layer)

106,624 learned parameters

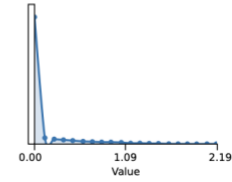
**Data shape:** [128 832 1 1]  
**Mean:** -0.001022546  
**Std deviation:** 0.03430906



### inception\_5b/pool\_proj

Activation

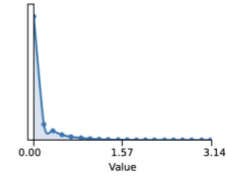
**Data shape:** [128 7 7]  
**Mean:** 0.1214796  
**Std deviation:** 0.2953759



### inception\_5b/output

Activation

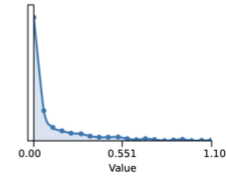
**Data shape:** [1024 7 7]  
**Mean:** 0.12123111  
**Std deviation:** 0.24574079



### pool5/7x7\_s1

Activation

**Data shape:** [1024 1 1]  
**Mean:** 0.12123111  
**Std deviation:** 0.18832587

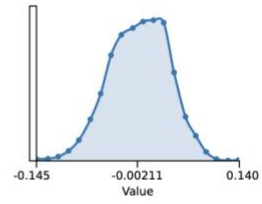


### loss3/classifier

Weights (*InnerProduct layer*)

3,075 learned parameters

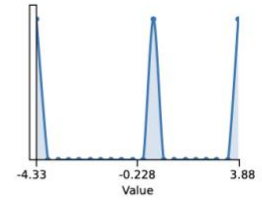
**Data shape:** [ 3 1024]  
**Mean:** 3.1904008e-06  
**Std deviation:** 0.0398968



### loss3/classifier

Activation

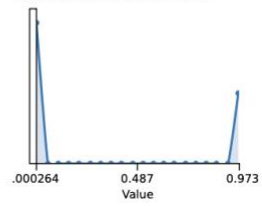
**Data shape:** [3]  
**Mean:** -0.055930167  
**Std deviation:** 3.3609653



### softmax

Activation

**Data shape:** [3]  
**Mean:** 0.3333333  
**Std deviation:** 0.45235813



## ÖZGEÇMİŞ

Yaren Özsoy, 2015 yılında Trakya Üniversitesi Bilgisayar Mühendisliği bölümünde lisans eğitimine başladı ve 2019 yılında mezun oldu. 2016 yılında Anadolu Üniversitesi Halkla İlişkiler ve Reklamcılık bölümünde lisans eğitimine başladı ve 2020 yılında mezun oldu. 2019 yılında Trakya Üniversitesi Bilgisayar Mühendisliği yüksek lisans eğitimine başladı. Lisans eğitimini tamamladıktan sonra 2019 yılında RS Positive Bilgi Teknolojileri ve Anonim Şirketinde yazılım geliştiricisi olarak, 2020 yılında Sensotek Otomasyon ve Bilgi Teknolojileri Şirketinde yazılım geliştiricisi ve proje yöneticisi olarak çalıştı. Ardından 2020 yılında Ekol Lojistik Yeni Nesil Teknoloji Şirketinde İş Zekâsı ve Veri Ambarı alanında çalıştı. 2022 Mart itibariyle Trakya Üniversitesi Bilgisayar Mühendisliği bölümünde araştırma görevlisi olarak çalışmaktadır.