

İSTANBUL TEKNİK ÜNİVERSİTESİ ★ FEN BİLİMLERİ ENSTİTÜSÜ

**DERİN ÖĞRENME İLE PNÖMONİ HASTALIĞINI
TESPİT ETME**

YÜKSEK LİSANS TEZİ

Gamze AKPOLAT

Elektronik ve Haberleşme Mühendisliği Anabilim Dalı

Telekomünikasyon Mühendisliği Programı

EYLÜL 2019

İSTANBUL TEKNİK ÜNİVERSİTESİ ★ FEN BİLİMLERİ ENSTİTÜSÜ

**DERİN ÖĞRENME İLE PNÖMONİ HASTALIĞINI
TESPİT ETME**

YÜKSEK LİSANS TEZİ

**Gamze AKPOLAT
504151312**

Elektronik ve Haberleşme Mühendisliği Anabilim Dalı

Telekomünikasyon Mühendisliği Programı

Tez Danışmanı: Dr.Öğr.Üyesi Hülya YALÇIN

EYLÜL 2019

İTÜ, Fen Bilimleri Enstitüsü'nün 504151312 numaralı Yüksek Lisans Öğrencisi Gamze AKPOLAT, ilgili yönetmeliklerin belirlediği gerekli tüm şartları yerine getirdikten sonra hazırladığı “Derin Öğrenme ile Pnömoni Hastalığını Tespit Etme” başlıklı tezini aşağıda imzaları olan jüri önünde başarı ile sunmuştur.

Tez Danışmanı : **Dr.Öğr.Üyesi Hülya YALÇIN**
İstanbul Teknik Üniversitesi

Jüri Üyeleri : **Prof. Dr. Zümray Dokur ÖLMEZ**
İstanbul Teknik Üniversitesi

Dr.Öğr.Üyesi Abdurrahim AKGÜNDOĞDU
İstanbul Üniversitesi

Teslim Tarihi : 20 Ağustos 2019
Savunma Tarihi : 3 Eylül 2019



ÖNSÖZ

Öncelikle tez hocam Sayın Dr.Öğr.Üyesi Hülya Yalçın'a vermiş olduğu destek ve katkılar için çok teşekkür ederim. Yüksek lisans eğitimim ve tez çalışmam sırasında imkan ve olanaklar sunan tüm yöneticilerime, desteklerini esirgemeyen ailem ve arkadaşlarıma sevgi, saygı ve teşekkürlerimi sunarım.

Eylül 2019

Gamze AKPOLAT





İÇİNDEKİLER

Sayfa

ÖNSÖZ.....	v
İÇİNDEKİLER	vii
KISALTMALAR	ix
ÇİZELGE LİSTESİ.....	xi
ŞEKİL LİSTESİ.....	xiii
ÖZET.....	xv
SUMMARY	xvii
1.GİRİŞ.....	1
1.1 Çalışmanın Amacı	1
1.2 Literatür Özeti	2
1.3 Hipotez	5
2. PNÖMONİ HASTALIĞI	7
2.1 Belirtileri	7
2.2 Teşhis	8
2.3 Tedavi.....	8
2.4 Akciğer iltihabının Röntgen Desenleri.....	8
3. MATERYAL VE METOT	11
3.1 Veri Kümesi İçeriği.....	11
3.2 Makine Öğrenmesi ile Model Oluşturma.....	11
3.3 Konvolüsyonel Sinir Ağları	13
3.3.1 Konvolüsyon Katmanı	14
3.3.2 Kaydırma Adımı	14
3.3.3 Piksel Ekleme.....	14
3.3.4 Havuzlama Katmanı.....	14
3.3.5 Önyargı ve Varyans	15
3.3.6 Uyum Durumları	16
3.3.7 Kayıp ve Maliyet Fonksiyonları	17
3.3.8 Ölçüm Formülleri.....	18
3.3.9 Derin Öğrenmede Kullanılan Hiper Parametreler	19
3.3.10 Aktivasyon Fonksiyonları	20
3.3.11 Regularizasyon ve Yöntemleri.....	25
3.3.12 Dereceli Alçalma Kavramı.....	26
3.3.13 Optimizasyon Algoritmaları	28
3.3.14 Toplu Normalleştirme	31
3.3.15 Ayrılabilir Konvolüsyon	32
3.4 Özellik Haritalarının Görselleştirilmesi	35
3.5 Veri Artırma İşlemi	35
4.DENEYLER	41
4.1 Model Tasarımları ve Parametre Değerleri	41
4.1.1 CNN Modeli.....	42
4.1.2 Inception Modeli	52

4.1.3 VGG16 Modeli.....	63
4.1.4 Modellerin Başarı Ölçüm Sonuçları ve Karşılaştırmaları	75
5.SONUÇ VE ÖNERİLER.....	77
KAYNAKLAR.....	79
EKLER.....	83
EK A: VGG16 Modelinin Diagramı	83
ÖZGEÇMİŞ.....	87



KISALTMALAR

CNN	: Konvolüsyonel Sinir Ağları'nı
RNN	: Tekrarlayan Sinir Ağları
DenseNet	: Yoğun Evrişim Ağı
DSC	: Derinlemesine Ayrılabilir Konvolüsyon
RGB	: Kırmızı-Yeşil-Mavi
BGR	: Mavi-Yeşil-Kırmızı
Dev	: Doğrulama veri kümesi
MSE	: Ortalama kare hatası
MAE	: Ortalama mutlak hata
TN	: Gerçek Olumsuz
TP	: Gerçek Olumlu
FN	: Yanlış Olumsuz
FP	: Yanlış Olumlu
ACC	: Doğruluk
TPR	: Gerçek Pozitif Oran
PPV	: Pozitif Öngörü Değeri
TNR	: Doğru Negatif Oran
F1	: F Skor
SDG	: Stochastic Gradient Descent
BGD	: Batch Gradient Descent
ADAM	: Adaptive Moment Estimation
RMSprop	: Root Mean Square Error Propabilit

ÇİZELGE LİSTESİ

	<u>Sayfa</u>
Çizelge 3. 1: Modelin Olasılıkları	19
Çizelge 4. 1: CNN modelinin parametre değerleri	44
Çizelge 4. 2: CNN modelinin başarı ölçüm sonuçları	44
Çizelge 4. 3: CNN modelinin veri artırma uygulanmamış başarı ölçüm sonuçları...	47
Çizelge 4. 4: Inception modelinin parametre değerleri	53
Çizelge 4. 5: Inception modelinin başarı ölçüm sonuçları	53
Çizelge 4. 6: Inception modelinin K=3 ve E=10 için parametre değerleri.....	56
Çizelge 4. 7: Inception modelinin K=3 ve E=10 için başarı ölçüm sonuçları.....	56
Çizelge 4. 8: Inception modelinin K=10 ve E=10 için parametre değerleri.....	58
Çizelge 4. 9: Inception modelinin K=10 ve E=10 için başarı ölçüm sonuçları.....	58
Çizelge 4. 10: Inception modelinin K=3 ve E=20 için parametre değerleri.....	60
Çizelge 4. 11: Inception modelinin K=3 ve E=20 için başarı ölçüm sonuçları.....	60
Çizelge 4. 12: VGG16 modelinin E=20 için parametre değerleri	64
Çizelge 4. 13: VGG16 modelinin E=20 için başarı ölçüm sonuçları	65
Çizelge 4. 14: VGG16 modelinin K=3 ve E=10 için parametre değerleri	67
Çizelge 4. 15: VGG16 modelinin K=3 ve E=10 için başarı ölçüm sonuçları	67
Çizelge 4. 16: VGG16 modelinin K=10 ve E=10 için parametre değerleri	69
Çizelge 4. 17: VGG16 modelinin K=10 ve E=10 için başarı ölçüm sonuçları	69
Çizelge 4. 18: VGG16 modelinin K=3 ve E=20 için parametre değerleri	71
Çizelge 4. 19: VGG16 modelinin K=3 ve E=20 için başarı ölçüm sonuçları	71
Çizelge 4. 20: Modellerin ölçüm sonuçlarının karşılaştırması	75



ŞEKİL LİSTESİ

	<u>Sayfa</u>
Şekil 2. 1: Hastalıklı akciğer görüntüleri.....	9
Şekil 2. 2: Sağlıklı bir akciğer görüntüsü	9
Şekil 3. 1 : Makine öğrenmesi ile Model oluşturma adımları	12
Şekil 3. 2 : Veri kümesi içeriğinin dağılım durumu	12
Şekil 3. 3: Maksimum havuzlama (Max pooling) işlemi	15
Şekil 3. 4: Veri kümesinde uyum durumları	16
Şekil 3. 5: Sigmoid aktivasyon fonksiyonunun grafiği	21
Şekil 3. 6: Tanh aktivasyon fonksiyonunun grafiği	22
Şekil 3. 7: ReLU aktivasyon fonksiyonunun grafiği	23
Şekil 3. 8: Leaky- ReLU aktivasyon fonksiyonunun grafiği.....	23
Şekil 3. 9: Degrade iniş ve küresel minimum	27
Şekil 3. 10: Optimizasyon algoritması ve küresel minimum noktası	29
Şekil 3. 11: Derinlemesine evrişim gösterimi.....	33
Şekil 3. 12: Noktasal evrişim.....	33
Şekil 3. 13: 256 çekirdekli noktasal konvolüsyon.....	34
Şekil 3. 14: Veri artırma işlemi uygulanacak bir akciğer röntgeni.....	36
Şekil 3. 15: Yatay çevirme ile veri artırma (Horizontal Flipping)	37
Şekil 3. 16: Rotasyon ile veri artırma işlemi	37
Şekil 3. 17: Parlaklık artırma ile veri artırma işlemi	38
Şekil 3. 18: Veri seyrekleştirme-gürültü ekleme ile veri artırma işlemi.....	39
Şekil 4. 1: Veri kümesinden örnek görüntüler.....	42
Şekil 4. 2: Sıfırdan oluşturulmuş CNN modelinin mimarisi	43
Şekil 4. 3: CNN modelinin Conv Blok mimarisi.....	43
Şekil 4. 4: CNN modelinin S-Conv Blok mimarisi	43
Şekil 4. 5: CNN modelinin Transaction Blok mimarisi	43
Şekil 4. 6: CNN modelinin Dense Fully Connected mimarisi	43
Şekil 4. 7: CNN model için çizdirilen confussion matrix.....	45
Şekil 4. 8: CNN modelin acc-epoch değişimi	45
Şekil 4. 9: CNN modelin loss-epoch değişimi.....	46
Şekil 4. 10: CNN modelinin ROC grafiği	46
Şekil 4. 11: CNN modelinin veri artırma işlemi uygulanmadan çizdirilen grafiği ...	47
Şekil 4. 12: CNN model veri artırma işlemi uygulanmamış acc-epoch değişimi	48
Şekil 4. 13: CNN model veri artırma işlemi uygulanmamış loss-epoch değişimi.....	48
Şekil 4. 14: CNN modelinin veri artırma işlemi uygulanmamış ROC grafiği	48
Şekil 4. 15: Sağlıklı verinin CNN model 1.katman farklı filtre görüntüleri.....	49
Şekil 4. 16: Pnömonili verinin CNN model 1.katman farklı filtre görüntüleri.....	50
Şekil 4. 17: Sağlıklı verinin CNN model 7.katman farklı filtre görüntüleri.....	50
Şekil 4. 18: Pnömonili verinin CNN model 7.katman farklı filtre görüntüleri.....	51
Şekil 4. 19: Sağlıklı verinin CNN model 9.katman farklı filtre görüntüleri.....	51
Şekil 4. 20: Pnömonili verinin CNN model 9.katman farklı filtre görüntüleri.....	52
Şekil 4. 21: Inception model için çizdirilen confussion matrix	54
Şekil 4. 22: Inception modelin acc-epoch değişimi.....	54

Şekil 4. 23: Inception modelin loss-epoch değişimi	54
Şekil 4. 24: Inception modelin ROC grafiği	55
Şekil 4. 25: Inception modelinin K=3 ve E=10 için confusion matrix	57
Şekil 4. 26: Inception modelinin K=3 ve E=10 için acc-epoch değişimi	57
Şekil 4. 27: Inception modelinin K=3 ve E=10 için loss-epoch değişimi	57
Şekil 4. 28: Inception modelinin K=3 ve E=10 için ROC grafiği	58
Şekil 4. 29: Inception modelinin K=10 ve E=10 için confusion matrix	59
Şekil 4. 30: Inception modelinin K=10 ve E=10 için acc-epoch değişimi	59
Şekil 4. 31: Inception modelinin K=10 ve E=10 için loss-epoch değişimi	59
Şekil 4. 32: Inception modelinin K=10 ve E=10 için ROC grafiği	60
Şekil 4. 33: Inception modelinin K=3 ve E=20 için confusion matrix	61
Şekil 4. 34: Inception modelinin K=3 ve E=20 için acc-epoch değişimi	61
Şekil 4. 35: Inception modelinin K=3 ve E=20 için loss-epoch değişimi	61
Şekil 4. 36: Inception modelinin K=3 ve E=20 için ROC grafiği	62
Şekil 4. 37: Inception modeli için sağlıklı ara filtre çıktıları	62
Şekil 4. 38: Inception modeli için pnömonili ara filtre çıktıları	63
Şekil 4. 39: VGG16 modeli için çizdirilen confusion matrix	65
Şekil 4. 40: VGG16 modeli acc-epoch değişimi	65
Şekil 4. 41: VGG16 modeli loss-epoch değişimi	66
Şekil 4. 42: VGG16 modeli ROC grafiği	66
Şekil 4. 43: VGG16 modeli K=3 ve E=10 için çizilen confusion matrix	68
Şekil 4. 44: VGG16 modeli K=3 ve E=10 için acc-epoch değişimi	68
Şekil 4. 45: VGG16 modeli K=3 ve E=10 için loss-epoch değişimi	68
Şekil 4. 46: VGG16 modeli K=3 ve E=10 için ROC grafiği	69
Şekil 4. 47: VGG16 modeli K=10 ve E=10 için çizilen confusion matrix	70
Şekil 4. 48: VGG16 modeli K=10 ve E=10 için acc-epoch değişimi	70
Şekil 4. 49: VGG16 modeli K=10 ve E=10 için loss-epoch değişimi	70
Şekil 4. 50: VGG16 modeli K=10 ve E=10 için ROC grafiği	71
Şekil 4. 51: VGG16 modeli K=3 ve E=20 için çizilen confusion matrix	72
Şekil 4. 52: VGG16 modeli K=3 ve E=20 için acc-epoch değişimi	72
Şekil 4. 53: VGG16 modeli K=3 ve E=20 için loss-epoch değişimi	72
Şekil 4. 54: VGG16 modeli K=3 ve E=20 için ROC grafiği	73
Şekil 4. 55: VGG16 modeli K=3 ve E=20 için sağlıklı ara filtre çıktıları	74
Şekil 4. 56: VGG16 modelinin K=3 ve E=20 için pnömonili ara filtre çıktıları	74
Şekil A. 1: VGG16 Modelinin Diagramı	86

DERİN ÖĞRENME İLE PNÖMONİ HASTALIĞINI TESPİT ETME

ÖZET

Dünya da en yaygın ve ölümlü sonuçlanabilir hastalıklar arasında yer alan pnömoni hastalığı, Türkiye’de de ölümlü sonuçlanan hastalıklar arasında üst sıralarda yer almaktadır. Erken teşhis ölüm oranını azaltmadaki öncelikli etkidir. Teşhis de en büyük yardımcı yöntem ise görüntüleme teknikleridir. Bu hastalığın teşhisinde en sık kullanılan yöntem ise görüntüleme tekniği olan röntgenlerdir.

Akciğer röntgeni ile hastalık tek başına çok büyük oranda tespit edilebilmektedir. Pnömoni ile benzer hastalıklar benzer desenler oluşturduğundan frontal yani önden çekilmiş görüntü, tanı ve tedavide tek başına yeterli olmayabilir. Lateral denilen yan taraftan çekilmiş görüntüler de gerekebilir.

Teşhiste doğruluk ve hızın artmasına katkıda bulunmak için birçok mühendislik uygulamalarına başvurulmaktadır. Bu uygulamalardan biri de makine öğrenmesi ile hastalık teşhis etmektir. Dünya da en popüler olan alanlardan biri veri bilimi ve makine öğrenmesi algoritmalarıdır. Medikal alanda bu konuya çok fazla ilgi gösterilmektedir. Gelişmiş ülkelerde makine öğrenmesi destekli hastaneler oluşturulmaktadır, var olan hastanelerin alt yapısı ise buna uygun hale dönüştürülmektedir.

Bu alana bu kadar yatırım yapılmasının başlıca nedenleri şunlardır;

- Makine öğrenmesi destekli hastaneler ile hastanın, şikayet ve belirtileri göz önünde bulundurularak, görüntüleme yöntemleri ve tahliller ile hastanın öncelik durumunu belirleme,
- Doğru ve hızlı aksiyon alınarak doktorların üzerindeki yükü azaltma,
- Yeterli hastane ve doktorun olmadığı bölgelerde, merkezi bir görüntüleme ve laboratuvarın olması, buraya gelen hastaların gerekli bilgilerinin alındıktan sonra verilerin makine öğrenmesi algoritmalarından geçirilip, kişinin hasta olup olmadığının tespitinin yapılması,
- Hastada görülen belirtilere göre kullanılabileceği ilaçların verilmesi,
- Tespit edilen hastalık için ilgili alana ve uzmana yönlendirilmesi,
- Veride yeterli kalite sağlandığında hastalıkların segmentasyonu ve detaylı analizlerinin yapılabilmesi.

Hedeflenen bu modern hastanelerin alt yapısının sağlanabilmesi öncelikli olarak geçmiş verilerin etiketlenmesi ve depolanmasına bağlıdır.

Bu çalışmanın öncülerinden olan The NIH Clinical Center akciğer röntgenlerini etiketlemiş ve hasta kimliğini koruyarak bu verilerin işlenebilmesi için veriler herkese açık hale getirmiştir. Klinik düzenli olarak sağladığı etiketli verilerin sayısını ve çeşidini artırmaktadır.

Bu tez çalışmasında derin öğrenme metodu kullanılarak röntgen görüntülerinden pnömoni hastalığını teşhis etmek için, kliniğin sağladığı veriler kullanılarak bir

sınıflandırma modeli oluşturulmuştur. Bu model oluşturulurken aynı ve benzer veri kümesi üzerinde yapılmış çalışmalar incelenmiş ve o çalışmaların üzerine ek çalışmalar yapılarak başarı oranları artırılmıştır. Performans ve hız sorununa yeni bir çözüm sunulmuştur.

Medikal veri kümelerinde genel olarak hasta etiketli veriler çoğunlukta olduğundan, bu tür veri setlerinde sağlıklı ve hasta etiketli veriler arasında ciddi bir dengesiz dağılım olur. Bu çalışmada kullanılan veri kümesinde de dengesiz dağılım mevcut olduğundan, bu sorun veri artırma yöntemi ile giderilmiştir. Veri artırma yöntemi sadece sağlıklı etiketli verilere uygulanmıştır. Veri hazırlama sürecinde, uygun olmayan veriler, kanal sayısı üç olmayan görüntüler gibi, veri kümesinden çıkarılmamış, bunun yerine uygun hale gelecek şekilde dönüştürülmüştür.

Makine öğrenmesinde karşılaşılan en büyük sıkıntılardan biri olan aşırı öğrenmeyi engellemek için verilerde normalizasyon, seyreltme, erken durdurma gibi tekniklere başvurulmuştur.

Modeller oluşturulurken konvolüsyonel sinir ağları ile derinlemesine ayrılabilir konvolüsyonel ağlar kullanılmıştır. Model çıktıları çeşitli başarı ölçüm yöntemleri ile incelenmiş olup, başarı oranının artırılması için değiştirilebilir hiper parametreler belirtilmiştir. Tüm çıktılar yorumlanmış ve ileride yapılacak çalışmalarda dikkate alınması gereken bulgular detaylıca açıklanmıştır.

DETERMINING OF PNEUMONIA DISEASE WITH DEEP LEARNING

SUMMARY

In the world, the most common and fatal diseases is the pneumonia disease, it is situated at the top among the fatal disease in Turkey. Early diagnosis and treatment are the primary factors in reducing the death rate. One of the most important methods of diagnosis is imaging techniques, which one is X-ray. This is the most commonly used method in the diagnosis of the disease. The disease can be detected correctly with using only chest x-ray in high probability.

Similar diseases with pneumonia create similar patterns in lungs. Because of that the frontal image may not be sufficient for diagnosis and detection alone. It may be required the lateral images for correct disease diagnosis.

In the medical field, many engineering applications are used to contribute to increase accuracy and speed in diagnosis. One of these applications is to diagnose various diseases and find the relationship between them through machine learning.

One of the most popular areas in the world is data science and artificial intelligence. Much attention has been paid to this issue in the medical field. In developed countries, hospitals with artificial intelligence support are formed. The infrastructure of the existing hospitals is transformed into the corresponding state.

The main reasons for investing in this area are as follows;

- With the help of artificial intelligence-supported hospitals, the patient's priority status is determined with imaging methods by assays and taking into account the symptoms.
- This algorithms taking correct and quick action, which is provides to reduce the overtime hours of doctors
- In a region where there is a central imaging and laboratory, but where there is no comprehensive hospital and a sufficient number of physicians, the data of the patients are passed through artificial intelligence algorithms and it is determined whether the person is ill or not,
- Giving the medicine to be used according to the symptoms seen in the patient,
- The patient is referred to the relevant field and specialist for the problem detected.
- When sufficient quality is achieved, segmentation and detailed analysis of diseases can be performed.

The provision of the infrastructure of these targeted modern hospitals depends primarily on the labeling and storage of historical data.

This clinic labeled lung X-rays and kept the data public for the purpose of processing this data while preserving patient identity. Clinical staff label data on a regular basis.

In this way, the number and variety of data is increasing. The clinic updates data sets periodically.

In this thesis, a classification model was formed by using the data provided by the clinic in order to diagnose pneumonia from the X-ray images using deep learning method.

In this model, the studies on the same data set were examined and additional studies were performed on these studies and the performance rates were increased. A new solution to the problem of performance and speed in model training is presented.

In general, patient-labeled data are predominant in medical data sets. In such data sets, there is a serious unbalanced distribution between healthy and unhealthy labeled data. Since the unbalanced distribution is present in the data set used in this study, this problem has been solved with the data increment method which is called data augmentation. The data enhancement method was applied only to healthy labeled data. In the process of preparing data, inappropriate data was transformed so that the number of channels was not subtracted from the data set, such as non-three images, instead being appropriate.

In order to prevent overfitting, which is one of the biggest problems encountered in machine learning, techniques such as normalization, dilution and early stop were applied.

When forming the model, convolutional neural networks and depthwise separable convolution has been used. One of the biggest challenges in machine learning is the need for a longer time requirement to teach, as well as the need for more computationally sophisticated computers due to processing and calculation overhead. The most important feature that distinguishes depthwise separable convolution from other convolutions is that it reduces the number of calculated parameters from millions number to thousands number and provides a great speed.

The model outputs have been investigated by various performance measurement methods and changeable hyperparameters have been stated to increase the performance rate. Some one of them are recall, precision, accuracy and f1 score.

All outputs were interpreted according to these measurement methods.

It is seen that the results of this study have higher performance rates compared to the previous studies. All methods and terms used in this study are explained firstly with cause and effect relationship and then step by step methods are explained. The findings to be taken into consideration in future studies are explained in detail.

In previous studies, CNN and RNN networks were formed with different models and parameters. Although the same neural networks and the same activation function were used in the same data set, better performance rates were obtained by the design of the intermediate layers and the differentiation of the hyper parameter values. This means that the performance ratio of the model can be higher with the appropriate hyper-parameter values. It is thought that performance rates can be increased to higher values with different classification methods.

In addition to this information, the images in the data set examined are frontal shots. That is, it consists only of views taken from the front. Some diseases can produce similar patterns in the lungs. In order to avoid confusion of different diseases with similar images, lung X-rays should consist of more extensive images called lateral. In machine learning algorithms, the distribution of images in the data set to be

trained should be balanced. Any tagged data should not be too dense from any other data. The number of different labeled data should be close to one another. In order to diagnose the disease, the symptoms, patient demographic information, X-rays and other diagnostic methods should be evaluated together. Therefore, this data set provided by the clinic and used in the thesis study will be more modern and accurate when enriched with other data of the patient.

Early diagnosis is of great importance for the treatment of this health problem. Many engineering applications are used to help improve accuracy and speed in diagnostics. One of these applications is to diagnose disease with machine learning. In this thesis, using deep learning methods, classification models will be developed to diagnose pneumonia from X-ray images.

In this thesis, the data set consisting of lung images of patients and healthy people was examined and models were formed to classify the diagnosis of disease with deep learning. The models created were based on previous studies and exceeded the achievements achieved.

Detected data imbalance in distribution when examining dataset data has been exceeded with the increase method. Although the data set was limited, 10% to 20% higher success was achieved than previous studies. In addition, the success of the data enhancement process on the model was measured by experiments. It is thought that the success rate in each model will increase when the number of data in the data set is increased to be balanced. Since the number of data was low, data with one channel number (one channel in grayscale) was converted to processable for each model by adding the number of channels (RGB three channels) instead of being removed from the cluster.

Early stop, data dilution and normalization procedures were applied to avoid excessive learning. In order to overcome the performance problem, in-depth separable convolution is used. With this convolution method, high computational processes were avoided and more performance and faster results were obtained. It has been observed that the success rate can be increased by changing the hyper parameters.

In addition, different classification methods can be used in imaging techniques to achieve higher success rates. In addition to the random division of the data set, the k-fold method was also applied and the results of the measurement were compared and the models for which the k-fold method was successful were determined.



1.GİRİŞ

Tıbbi görüntüleme teknikleri 19. yüzyıldan beridir hayatımızda yer almaktadır ve her geçen gün bu teknolojiler gelişmektedir. Medikal görüntüleme teknolojisinde genellikle elektromanyetik, parçacık, radyasyon kullanılmaktadır [1]. Canlıların biyolojik yapısından geçebilen radyasyonun zaman ve uzaydaki değişimine bağlı olarak şiddet dağılımının belirlenmesi ile görüntü oluşur [1]. Bu görüntüler ile canlının içyapısı hakkında bilgi edinilir. Bu bilgiler, canlının içyapısının nasıl çalıştığı, var olan veya oluşabilecek hastalıkların tanı ve tedavisinde kullanılmaktadır [1].

Medikal görüntüleme tekniklerinden biri olan röntgen, adını mucidi olan Alman fizikçi Wilhelm Conrad Röntgen'den almaktadır. Ünlü fizikçi, X ışınlarının bir doğru boyunca yansıma ve kırılmaya uğramadığını, elektrik veya manyetik alanların etkisi ile yön değiştirmedeğini ispatlayarak Nobel Bilim Ödülü almıştır. Röntgenin tıpta kullanımı ise, X ışınlarının canlı dokular tarafından homojen olmayan derecelerde emilmesine dayanır. Eşit olmayan bu geçişler, radyolojik gölgeler yani görüntüler meydana getirir.

1.1 Çalışmanın Amacı

Röntgen teknolojisi vücudun birçok bölümünü incelemede kullanılır. Bunlardan biri de akciğer sorunlarını tespit etmedir. Zatürre (Akciğer İltihabı), tüberküloz veya akciğer kanseri gibi sorunların belirtileri göğüs röntgenlerinde görülebilmektedir. Halk arasında zatürre olarak bilinen akciğer iltihaplanmasının tıbbi ismi pnömonidir [2]. Esasen, akciğerin en uç yapısını oluşturan ve havadaki oksijenin vücuda girdiği bölge olan hava keseciklerinin iltihaplanması hastalığıdır [3].

Pnömoni, Türkiye'de ölüme sebep olan hastalıklar arasında beşinci sırada yer alırken enfeksiyona bağlı ölümler arasında birinci sıradadır [2]. Aynı şekilde bu hastalık dünya çapında yaklaşık 450 milyon kişiyi etkilemektedir (nüfusun% 7'si) ve yılda yaklaşık 4 milyon ölümlerle sonuçlanmaktadır [2, 3].

Bu sađlık sorununun tedavisi iin erken teŖhis byk nem taŖıtmaktadır. TeŖhiste dođruluk ve hızın artmasına katkıda bulunmak iin birok mhendislik uygulamalarına baŖvurulmaktadır. Bu uygulamalardan biri de makine đrenmesi ile hastalık teŖhis etmektir.

Bu tez alıŖmasında derin đrenme metodları kullanılarak rntgen grntlerinden pnmoni hastalıđını teŖhis etmek iin sınıflandırma modelleri oluŖturulacaktır.

1.2 Literatr zeti

Pnmoni hastalıđını teŖhis etmek iin daha nce alıŖmalar yapılmıŖtır. İlgili veri kmesi zerinde yapılan alıŖmaların ilki Wang'a aittir [4]. Wang alıŖmasında veri kmesinin %70 'ini eđitim, %10'unu validasyon ve %20'sini test iin kullanmıŖtır. Yntem olarak Konvolsyonel Sinir Ađları'nı (CNN) uygulamıŖ, baŖarı oranını artırmak iin daha sonra ResNet-50 zerine yođunlaŖmıŖtır. Eđitim kmesini ImageNet zerinden eđitmiŖtir. Wang, gđs rntgeni veri kmesinde eŖitli modeller zerinde alıŖmıŖtır. Her bir model iin hiper parametreleri deđiŖtirerek eŖitli deneyler yapmıŖtır. Yaptıđı deneylerden en baŖarılı sonular alınmıŖtır. Kurduđu Cnn modelini ImageNet ile eđittiđinde; 0.75 accuracy, 0.66 precision, 0.93 recall, 0.77 f1 score, ve 0.63 auc deđerini elde etmiŖtir [4]. alıŖmalarında nceden eđitilmiŖ modellerden AlexNet, GoogLeNet, VGGNet-16 ve ResNet-50 kullanmıŖ, sonularını karŖılaŖtırmıŖtır. AlexNet ile 0.54, GoogLeNet ile 0.59, VGGNet-16 ile 0.51 ve en son ResNet-50 ile 0.63 baŖarı oranı yakalamıŖtır [4]. Bu modeller ile elde ettiđi auc deđerleri 0.51 ile 0.63 arasında deđiŖmiŖtir.

Wang'dan sonra aynı veri kmesi zerinden Li Yao alıŖma yapmıŖtır [5]. alıŖmasında Tekrarlayan Sinir Ađları'nı (RNN) kullanmıŖtır. Yao'da Wang gibi eđitim kmesini rastgele olarak %70 'ini eđitim, %10'unu validasyon ve %20'sini test iin ayırmıŖtır. Modelini OpenI veri kmesinde eđitmiŖtir. Sigmoid aktivasyon fonksiyonunu kullanmıŖtır. Li Yao'da kendisinden nce aynı veri kmesinde alıŖan Wang'ın alıŖmasını inceleyip, kendi bulguları ile karŖılaŖtırmıŖtır [4,5]. Li Yao alıŖmaları sonucunda 0.71 baŖarı oranı yakalamıŖtır [5]. Diđer verdiđi baŖarı lmleri; precision 0.67, recall 0.76 ve auc deđerini ise 0.713 olarak belirtilmiŖtir. Optimizer olarak Adam kullanmıŖ olup, đrenme adımı olarak 0.001 deđerini vermiŖtir. Yine aynı veri kmesi zerinde Pranav Rajpurkar'da alıŖmıŖtır [6]. Konvolsyonel Sinir Ađı'nın geliŖmiŖ versiyonu olan Yođun EvriŖim Ađı (Dense

Convolutional Network -DenseNet) ile oluşturulan modelini CheXNet olarak adlandırmıştır [6]. Şimdiye kadar yapılan ve yukarıda bahsedilen çalışmalardan, çalışmaya en yakın olanı CheXNet olduğundan bu çalışmayı biraz daha detaylandıracağım. CheXNet modelinde sigmoid aktivasyon fonksiyonu kullanılmıştır [6]. Ağın ağırlıkları, ImageNet'te önceden tanımlanmış bir modelin ağırlıkları ile hesaplanmıştır. İyileştirici fonksiyon (optimizer) olarak ADAM kullanılmıştır. Grup sayısı (batch size) 16, öğrenme oranı (learning rate) 0.001 ve epoch değerini 10 vermiştir. Bu parametreler ışığında elde ettiği başarı oranı 0.76 olmuştur [6]. Rajpurkar'da bu başarı oranını yakalayabilmek için kendisinden önceki ve yukarıda anlattığım Wang ve Li Yao'nun çalışmalarını incelemiş ve bu çalışmalardan elde edilen başarı oranının üzerine çıkabileceği bir model oluşturmuştur.

Akciğer röntgenleri üzerinde farklı veri kümesinde olsa da Rahib H. Abiyev'de çalışmıştır [7]. Çalışmalarında çeşitli akciğer hastalıklarını sınıflandırmaktadır. Veri kümesinde 112120 görüntü bulunmaktadır. Çalışmalarında veri kümesini rastgele %70 eğitim, %20 test ve %10 validasyon olarak ayırmıştır. Abiyev, CNN ,Vgg16 ve VGG19 modellerini kullanmıştır. Aktivasyon fonksiyonu olarak Cnn modelinde ReLu ve Softmax, Vgg16 ve Vgg19 modelinde ise Sigmoid kullanmıştır. Çeşitli öğrenme adımı ve epoch değerleri vererek elde ettiği en başarılı sonuçlar şu şekildedir; Cnn modelinde (112120 adet görüntü içeren kümede) %92, Vgg16 modelinde (8100 adet görüntü içeren kümede) %86, Vgg19 modelinde (8100 adet görüntü içeren kümede) %92 olmuştur [7].

John R. Zech , Abiyev'in çalıştığı veri kümesinin üzerine, farklı hastanelerin ortak kullanım için yayınladığı görüntüleri de ekleyerek, daha geniş bir veri kümesi oluşturmuştur [8]. National Institutes of Health Clinical Center adlı sağlık merkezi 30.805 kişiden 112120, Mount Sinai Hospital adlı hastane ise 12.904 kişiden 42396 ve Indiana University Network for Patient Care adlı hastane ise 3.683 kişiden 3807 akciğer röntgenini yayınlamıştır. Zech bu görüntüleri birleştirerek 158323 görüntü üzerinde çalışmıştır. Zech, veri kümesinin diğer geleneksel çalışmalar gibi rastgele eğitim, test ve validasyon olacak şekilde üç farklı parçaya bölmüştür.

Çalışmasını Cnn modeli ile oluşturmuştur. Aktivasyon fonksiyonu olarak sigmoid kullanmış olup, öğrenme adımı olarak 0.01 vermiş, ağırlıkları ImageNet ile güncellemiş ve modelini 10 epoch eğitmiştir [8]. Her bir kurumdan aldığı veri

setlerinde hem ayrı ayrı hem de kombinasyonları olacak şekilde modelini çalıştırmıştır. Örneğin NIH ve MSH kurumlarının verisini ayrı ayrı çalıştırdığı gibi bu iki kurumun verilerini birleştirip bir de o şekilde modelini çalıştırmıştır. Hiper parametrelerinin yanı sıra veri setleri kombinasyonları ile yaptığı deneylerde auc değeri 0.69 ile 0.93 arasında değişmiştir. Acc değeri 0.25 ile 0.74 , Recall değeri 0.95 ile 0.97, Precision 0.011 ile 0.48 değeri arasında değişmiştir [8].

Okeke Stephen da akciğer röntgen görüntüleri üzerinde çalışmıştır [9]. Çalışmasında pnömonili ve sağlıklı verileri Cnn modeli ile sınıflandırmaya çalışmıştır. Çalışmasında 5,856 adet akciğer röntgen görüntüsü bulunmaktadır [9]. Veri dağılımı dengesiz olduğundan sentetik veri artırma işlemi ile veri dağılımını dengeli hale getirmiştir. Veri artırma işleminde rotasyon, parlaklık artırma, ters çevirme gibi yöntemleri kullanmıştır. Veri kümesini eğitim, test ve validasyon olacak şekilde bölmüştür [9]. Modelin de 4 conv2d katmanı, 2 tamamen bağlı katman kullanmıştır. Her katman arasında aktivasyon fonksiyonu olarak ReLu kullanmış son bağlı katman çıkışında ise sigmoid aktivasyon fonksiyonunu tercih etmiştir. Modelin de aşırı öğrenmeyi engellemek için drop out yani veri seyreltme işlemini 0.5 oranında uygulamıştır. Modelini farklı parametreler vererek 100 epoch eğitmiştir. Yaptığı deneylerde acc değeri 0.85 ile 0.95 arasında değişmiş olup elde edilen minimum loss değeri ise 0.18 olmuştur [9].

Ivo M. Baltruschat ve arkadaşları, The National Institutes of Health (NIH) Clinical Center merkezinin sağladığı 112120 adet akciğer röntgen görüntüsünde 14 farklı hastalığın sınıflandırılmasını, ResNet50 ve Resnet38 modelinin farklı hiper parametreleri ile çalışmıştır [10]. Modelinde 8 conv2d, 1 tamamen bağlı , 2 havuzlama (ortalama ve maksimum) katmanı kullanılmıştır [10]. Aktivasyon fonksiyonu olarak sigmoid tercih edilmiştir. ImageNet'te eğitilmiş bir ResNet-50 ağının ağırlıkları başlangıç noktası olarak kullanılmıştır. Optimizasyon fonksiyonu olarak Adam tercih edilmiştir. Öğrenme adımı 0.01 ile 0.001, batch size 8 ile 16 verilmiştir [10]. Çalışmanın farklı hiper parametreler ile yapılan deneylerinde tüm hastalıkları sınıflandırmadaki ortalama auc 0.71 ile 0.83, sadece pnömonili veriler de 0.66 ile 0.76 arasında değişmiştir [10]. Benzer çalışmaları ResNet38 ve ResNet101 de uygulamıştır. ResNet38 ile elde edilen ortalama auc 0.80, ResNet50 ile 0.79, ResNet101 ile 0.78 olmuştur [10]. Bu modellerde pnömonili verilerin auc değeri

ResNet38 için 0.714, ResNet50 için 0.703 ve ResNet101 için 0.694 olmuştur. Baltruschat, ResNet modellerinden en iyi sonucu ResNet38 ile almıştır [10].

Jonathan Rubin ve arkadaşları da akciğer röntgen görüntüleri üzerinde çalışmıştır. Diğer çalışmalardan farklı olarak bu ekip frontal dediğimiz önden çekilmiş görüntülerin yanı sıra lateral olarak adlandırılan sağ ve sol taraftan çekilmiş görüntüleri de kullanmışlardır. Çalıştıkları veri kümesi, MIMIC-CXR adı verilen 473064 göğüs röntgeni ve 63.478 hastadan toplanan 206574 radyoloji raporundan oluşan bugüne kadar yayınlanmış en büyük göğüs röntgeni veri kümesidir [11]. Çalışmalarında bir radyolojik muayeneden elde edilen hem ön hem de yanal akciğer röntgen (chest x ray – CXR) görüntülerini aynı anda işleyerek rutin klinik uygulamaları taklit eden yeni bir DualNet mimarisi kurmayı hedeflemişlerdir [11]. 14 farklı hastalığı Cnn modeli kullanarak sınıflandırmışlardır. Dolayısıyla, 14 çıkışın her birine bir sigmoid işlemi uygulayıp her bir ağı eğitmek için bir ikili çapraz-entropi kaybı fonksiyonu kullanmışlardır. Hem ön hem de yanal röntgen görüntüleri, giriş olarak ayrı evrişimsel sinir ağları ile işleyip, çıktılarını son bir sınıflandırma daha yapmak için tamamen bağlı bir katman ile birleştirmişlerdir. Veri kümesi ya da veri kümesi diğer çalışmalardaki gibi %70 eğitim, %10 validasyon ve %20 test olacak şekilde rastgele bölünmüştür. Bu veri kümesi içerisindeki pnömonili verinin sayısı 25,025 olarak belirtilmiştir. Modeli geliştirme için PyTorch kullanılmış ve modeller 8 Nvidia Titan Xp GPU üzerinde veri paralelligi kullanılarak eğitilmiştir. Kurulan model de tüm hastalıklar için ulaşılan en iyi ortalama auc değeri 0.70 olmuştur. Sadece pnömonili veriler için elde edilen auc değeri 0.61 olmuştur [11].

1.3 Hipotez

Bu tez çalışmasının üzerine kurulduğu hipotez, yukarıda incelenen çalışmalarda, kullanılmış olan çeşitli makine öğrenmesi algoritmaları ile verilen parametreler ve değerlerle elde edilen başarı oranının üzerine çıkabilecek yeni modeller oluşturmaktır.



2. PNÖMONİ HASTALIĞI

Pnömoni, akciğerlerin bir veya her iki lobunun, hava keseleri veya alveollerinin sıvı (ödem) toplaması veya iltihaplanması durumudur [12,13]. Pnömoni bakteriyel, viral veya mantar kaynaklı olabilir.

Akciğer enfeksiyonuna neden olan mikrop tipi ile hastanın yaş ve genel sağlık durumu, pnömoninin hastadaki risk faktörünü belirlemektedir [12,14]. Bu hastalık yoğun olarak çocuk ve 65 yaş üzeri yetişkinlerde görülmektedir [15]. Akciğer iltihabı, çocuk ve yaşlıların yanı sıra, kalp yetmezliği, diyabet, KOAH (kronik obstrüktif akciğer hastalığı) gibi belirli rahatsızlıkları olan ya da bağışıklık sistemini zayıflatabilecek (HIV, AIDS, kemoterapi (kanser tedavisi) veya organ, kan, kemik iliği kök hücre nakli vb. gibi) hastalık ya da tedavi süreçleri içerisindeki insanlarda daha yüksek risk taşımaktadır [13,16,17].

2.1 Belirtileri

Semptomları arasında balgam çıkarma, ateş, üşüme ve solunum zorluğu ile öksürük yer almaktadır [18]. Ateş, pek çok başka yaygın hastalıkta meydana geldiğinden tanı koymada çok spesifik bir semptom değildir ama hastalık belirtileri arasında yer almaktadır. Öksürük ise, 2 aylıktan küçük çocuklarda genellikle yoktur [19]. Çocuklarda daha ciddi belirtiler arasında mavi renkte cilt, anne sütü ile beslenenlerde emme isteksizliği, vücutta kasılmalar, devam eden kusma, ateş veya azalmış bilinç düzeyi bulunabilir [19,20]. Bakteriyel ve viral pnömoni vakaları genellikle benzer semptomlarla ortaya çıkar [20]. Bazı nedenler klasik fakat spesifik olmayan klinik özelliklerle ilişkilidir. Legionella'nın neden olduğu zatürree karın ağrısı, ishal veya konfüzyon ile ortaya çıkabilir [18]. Streptococcus pneumoniae'nin neden olduğu pnömoni paslı renkli balgamla ilişkilidir [18]. Klebsiella'nın neden olduğu zatürree kanlı balgam ile ilişkilidir [18, 19]. Kanlı balgam (hemoptizi olarak da bilinir) ayrıca tüberküloz, Gram negatif pnömoni, akciğer apseleri ve daha sık akut bronşitte de görülebilir [20]. Mycoplasma pneumoniae'nin neden olduğu pnömoni, boyundaki lenf bezlerinin şişmesi, eklem ağrısı veya orta kulak enfeksiyonu ile birlikte ortaya

çıkabilir [20]. Viral pnömoni bakteriyel pnömoniden daha yaygın şekilde hırıltılı olarak ortaya çıkar [19,20].

2.2 Teşhis

Teşhis sıklıkla semptomlara, fiziki muayeneye ve detaylı testlere dayanır [13,18,19]. Pnömoni hastalığı ile benzer bulgular taşıyan pek çok hastalık vardır, örneğin pnömonit. Doğru teşhis için belirtiler, testler ve hasta geçmişi bir bütün olarak incelenmelidir [13,18,19,20,21]. Zatürree teşhisi koymak için uzman doktor tarafından hastanın tıbbi geçmişi gözden geçirilmeli, fiziksel muayene edilmeli (nefes alış verişinin dinlenilmesi gibi) daha sonra hasta; kan ve balgam testi, akciğer röntgeni vb. gibi detaylı testlerden geçirilmelidir [20,21]. Bu bilgiler doktorun zatürre tipini belirlemede yardımcı etkenlerdir [13,18]. Akciğer röntgeni hastalığın tanısını koymada ve hastalığın hangi aşamada olduğunu belirlemede asıl rolü oynamaktadır [22].

2.3 Tedavi

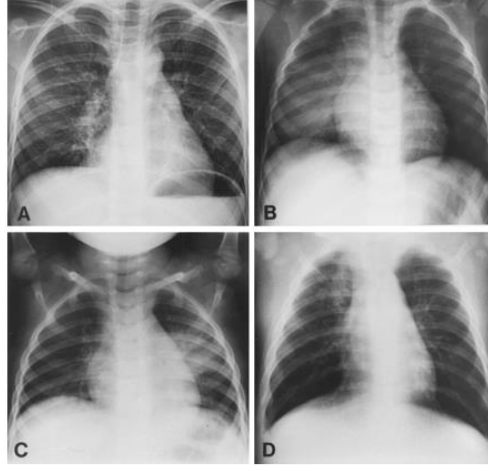
Zatürrenin tedavisi için öncelikle altta yatan nedenler incelenmelidir [18]. Detaylı testlere rağmen bazı vakalarda hastalığa neden olabilecek herhangi bir etken madde bulunamayabilir [18]. Bakteriyel kaynaklı olanlar genellikle antibiyotik ile tedavi edilebilir [23,24]. Oksijen seviyeleri düşükse oksijen tedavisi kullanılabilir [18,23]. Tedaviler Pnömoni tipine göre değişiklik göstermektedir, bu yüzden türünü doğru tespit etmek gerekir.

Tedavinin yanı sıra hastalığa yakalanmamak için önlemlerde alınmalıdır. Zatürre bulaşıcıdır [2]. Bazı pnömoni türlerini önleyen aşılarda mevcuttur [2,18]. Diğer önleme yöntemleri arasında el yıkama, sağlıklı beslenme, sigara içmemek ve temiz hava sahasında bulunmak sayılabilir [12,13,18,19].

2.4 Akciğer iltihabının Röntgen Desenleri

Mantar ve parazit türü yaygın olmasa da, bakteri veya virüs kaynaklı olanlar çok yaygındır [16]. Bunun yanı sıra hem bakteri hem de virüs kaynaklı olan karışık enfeksiyonlar da ortaya çıkmıştır [13]. Her pnömoni tipinin röntgendeki deseni farklılaşmaktadır. Fakat bu farklılıklar genelde tek bakışta ayırt edilebilecek şekilde

değildir. Aşağıda Şekil 2.1’de bakteriyel ve viral pnömoni röntgen görüntüleri ile Şekil 2.2 ‘ de sağlıklı bir akciğer görüntüsü görülmektedir [25,26].



Şekil 2. 1: Hastalıklı akciğer görüntüleri

Şekil 2.1’ de A. Her iki lobda bulunan bakteriyel Pnömoni görüntüsü, B. Sağ alt lobda bakteriyel kaynaklı alveoler değişiklikler olan Pnömoni görüntüsü, C. Rinovirüs ile ilişkili sol alt lobda alveoler değişiklikler olan Pnömoni görüntüsü, D. Parainfluenza 2 ve İnsan Herpes Virüsü ile ilişkili sağ üst lobda alveoler değişiklikler olan Pnömoni görüntüsü.



Şekil 2. 2: Sağlıklı bir akciğer görüntüsü



3. MATERYAL VE METOT

3.1 Veri Kümesi İeriđi

Veri kümesi pnömoni hastalığına sahip olan ve olmayan insanların akciđer röntgen görüntülerini içermektedir. Veri kümesinin içeriđi yaş, cinsiyet, hastalık geçmiři vb. gibi bilgiler ile desteklenmemiştir. Tamamen anonimleştirilmiştir. Burada veri kümesi hakkında sağlanan tek bilgi, görüntülerin tamamının farklı kişilerden alındığı yani kendi içinde tekil olduđu bilgisidir. Veri kümesi, yaklaşık 6000 farklı kişiden alınmış görüntüleri içermekte olup, toplam klasör boyutu 5,5 GB'ın üzerindedir. Veri kümesindeki görüntüler yüksek çözünürlüklü olup, jpeg formatındadır.

Veri kümesi The NIH Clinical Center tarafından paylaşılmıştır [27]. Paylaşılan veri kümesi bakteriyel ve viral pnömoni ile sağlıklı akciđer görüntülerini içermektedir. Veri kümesi içerisinde farklı pnömoni çeşitleri olsa da etiketleme olarak bir ayırım yapılmamış, Pnömoni veya Normal şeklinde etiketlenmiştir.

Veri sayısı az olduğundan daha alt sınıflara bölmek başarı oranının daha da düşmesine neden olacağından yeterli veri sayısı elde edilene kadar önceliđin sağlıklı ve pnömoni ayırımının yüksek başarı oranı ile yapılmasını sağlamaktır. Sınıflandırmanın yapılabilmesi için röntgen görüntüleri ilgili klinik doktorları tarafından etiketlenmektedir. Etiketlenen görüntü sayısı arttıkça veri kümesinin sürümleri ve boyutları deđişmektedir. Klinik, doktorların hastalar için daha iyi teşhis ve hızlı karar vermesinde makine öğrenmesi algoritmalarının yardımcı olacağını savunmaktadır.

3.2 Makine Öğrenmesi ile Model Oluşturma

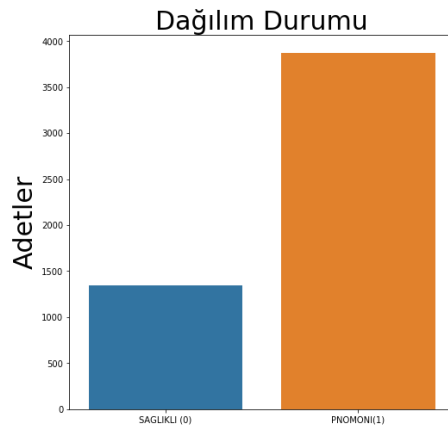
Makine öğrenmesi ile model oluşturma birkaç adımdan oluşur. Öncelikle bir problem belirlenir, daha sonra makine öğrenmesi ile ne yapılmak istendiđi belirlenir. Problem tanımı yapıldıktan sonra veri kümesi hazırlanır. Hazırlanan veri kümesinin içeriđi incelenir. Öznitelik yani girdi deđerleri oluşturulur. Veri kümesinde verinin temiz yani anlamlı olması beklenir. Boş deđerler varsa ya da veri kirli ise önce veri temizleme işlemleri yapılır. Daha sonra veri kümesindeki verilerin içeriđinin

araştırılması yapılır .Araştırma adımı istatistiksel ve görselleştirme tekniklerinin kullanılması işlemidir. Bu adımın yapılması problemin nasıl çözüleceği hakkında detaylı bilgi verir. Ayrıca uygulanacak modeli belirleme aşamasıdır. Veri kümesinin içeriğini, dağılımını görme ise verinin dengeli ya da dengesiz olup olmadığı hakkında bilgi verir. Bu bilgi ise verinin test, eğitim, validasyon gibi bölme oranlarının belirlenmesini sağlar. Veri araştırma adımından sonra model (sınıflandırma ya da regresyon) belirlenir. Model belirlendikten sonra veri kümesi kullanılan algoritmaya göre hesaplanır. Hesaplama adımında, algoritmanın parametrik değerleri değiştirilerek uygun ve yeterli görülen başarı oranı yakalanmaya çalışılır. Daha sonra model uygulamaya alınır. Şekil 3.1’ de makine öğrenmesi ile model oluşturmanın süreçleri gösterilmiştir.



Şekil 3. 1 : Makine öğrenmesi ile Model oluşturma adımları

Bu çalışmadaki problem tanımımız, pnömoni hastalığı ve tespitidir. Çalışma python dili ile pandas, numpy, matplotlib, scikit-learn, seaborn, keras, tensorflow vb gibi kütüphaneler kullanılarak yapılmıştır. Veri kümesi test, eğitim, validasyon olmak üzere üç parçaya ayrılmıştır. %70’i eğitim, %10 validasyon, %20 test datası olarak belirlenmiştir. Veri medikal veri olduğundan dengesiz bir dağılıma sahiptir, yani Şekil 3.2’de etiketlere baktığımızda hastalıklı etiketi, sağlıklı etiketinden çok daha fazladır. Bu sorun tüm medikal verilerde yaşanmaktadır. Bu durumda veri modelini iyi seçmek daha da önem kazanır. Aşağıda verinin dağılımı çizdirilmiştir.



Şekil 3. 2 : Veri kümesi içeriğinin dağılım durumu

Genel dağılıma baktığımızda pnömoni olanların sayısı sağlıklı olanların yaklaşık 2,5 katı görünüyor. Bu bilgiyi ve genel veri kümesi adetinin az olduğunu göz önünde bulundurduğumuzda overfitting yani aşırı öğrenme gerçekleştirmeden dengeli bir hesaplama yapabilecek bir model oluşturmamız gerekiyor. Bu veri kümesine uygun olan ve görüntü işlemede çok iyi sonuçlar veren Konvolüsyonel Sinir Ağları (CNN) ve Derinlemesine Ayrılabilir Konvolüsyon (DSC) kullanılacaktır.

3.3 Konvolüsyonel Sinir Ağları

Konvolüsyonel sinir ağları, bilgisayar görüntüleme alanında yoğun olarak kullanılan sinir ağı çeşitlerinden biridir. Adını, oluşturduğu gizli katmanların türünden alır. Bir CNN'in gizli katmanları tipik olarak evrişimsel/konvolüsyonel (convolutional), havuzlama (pooling), tamamen bağlı (fully-connected) ve normalizasyon (normalization) katmanlarından oluşur.

CNN görüntü sınıflandırmaları bir girdi imajı alır, işler ve belirli kategoriler altında sınıflandırır. Bilgisayarlar bir giriş görüntüsünü piksel dizisi olarak görür ve imaj çözünürlüğüne dayanır. İmaj çözünürlüğü (h) yükseklik \times (w) genişlik \times (d) boyut matris dizisinden oluşur. Burada boyut üç değerini alırsa RGB dediğimiz red-green-blue renklerinden oluşan çok kanallı (multi-channel), boyut bir değerini aldığınd ise Grayscale dediğimiz tek kanallı görüntüyü ifade etmektedir.

Burada boyut görüntü işlemede önem kazanmaktadır. Görüntü veri kümesinde boyutları üç olmayan yani tek kanallı (grayscale) verilenleri çok kanallı (multi-channel) yapıya çevirmek gerekir ya da bu verileri veri temizleme aşamasında veri kümesinden çıkarmak gerekir. Üzerinde çalıştığımız veri kümesinde bazı görüntüler tek kanallı boyutta verilmiştir, veri sayısı az olduğundan bu görüntüleri çıkarmak yerine boyut dönüşümü uygulanmıştır. Boyut dönüşümü uygulanırken kullanılan kütüphanelerde burada önem taşımaktadır. Çünkü numpy, matplotlib kütüphaneleri RGB byte sıralamasını kullanırken OpenCV kütüphanesi BGR byte sıralamasını kullanmaktadır. Bu ayrıntı göz önünde bulundurulmalıdır. CNN modelini detaylandırmadan önce bu modelde kullanılan terimler ve hesaplamaları nedeni ile birlikte incelemek gerekir.

3.3.1 Konvolüsyon Katmanı

Konvolüsyon ya da evrişim, bir giriş görüntüsünden özellikleri alan ilk katmandır. Evrişim, küçük girdi verisi kareleri kullanarak ve görüntü özelliklerini öğrenerek pikseller arasındaki ilişkiyi korur. Çıktı boyutunun hesaplaması Denklem 3.1'de verilmiştir.

Giriş görüntüsünün; h:yükseklik, w:genişlik, d: derinlik bilgisini verir.

Filtrenin ise; f_h : yükseklik, f_w : genişlik, f_d : derinlik bilgisini verir.

İmaj matrisinin boyutu: $(h \times w \times d)$

Filtrenin boyutu: $(f_h \times f_w \times d)$

Çıktıların boyutu: $(h - f_h + 1) \times (w - f_w + 1) \times 1$ (3.1)

3.3.2 Kaydırma Adımı

Giriş matrisinde kaydırılan piksel sayısıdır. Stride yani kaydırma adım sayısı olarak bilinir. Burada verilen değere göre giriş matrisinde filtreler kaydırılır. Genellikle s ile belirtilir. Bu çıkış boyutunu etkileyen bir parametredir.

3.3.3 Piksel Ekleme

Evrişim işlemi uygulandıktan sonra giriş ile çıkış arasında boyut farkı oluşabilir. Bu boyut farkı piksel ekleme yöntemi ile değiştirilebilir. Giriş matrisine ekstra piksel ekleyerek, eklenen pikseller sıfır olmalıdır (zero padding), boyut yeniden düzenlenebilir. Genellikle p ile belirtilir. Piksel ekleme değeri aşağıdaki Denklem 3.2 ile belirlenir. f burada filtre boyutunu verir.

$$p = (f-1) / 2 \quad (3.2)$$

Piksel ekleme ve kaydırma adımı işlemleri uygulandığında çıkış matrisinin boyutu aşağıdaki Denklem 3.3 ile bulunur:

$$[(w - f + 2p)/s] + 1 \quad (3.3)$$

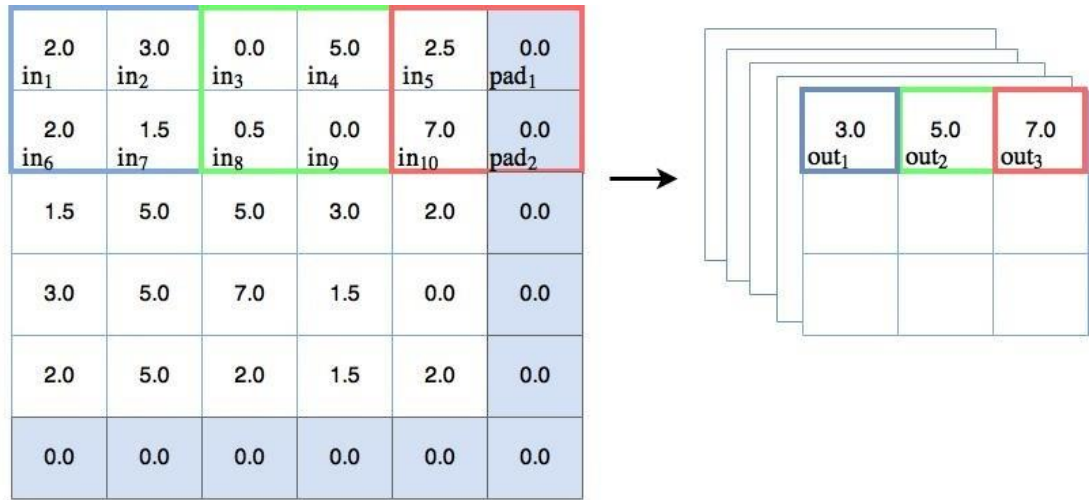
3.3.4 Havuzlama Katmanı

Havuz katmanları bölümü, görüntüler çok büyük olduğunda parametre sayısını azaltır. Mekansal havuzlama ayrıca her haritanın boyutsallığını azaltan ancak önemli bilgileri koruyan ve alt örnekleme olarak da adlandırılan bir yöntemdir. Üç farklı havuzlama yöntemi bulunmaktadır.

Maksimum Havuzlama (Max Pooling): Maksimum havuzlama, düzeltilmiş özellik haritasındaki en büyük öğeyi alır. Yani filtre imaj üzerinde gezdirilirken, filtreyi kapsayan en büyük değer alınır. Bu çalışmada bu teknik kullanılmıştır. Aşağıda Şekil 3.3 'te hesaplamaların nasıl yapıldığının bir örneği verilmiştir [28].

Ortalama Havuzlama (Average Pooling): Özellik haritasındaki tüm öğelerin ortalamasının alınmasıdır.

Toplam Havuzlama (Sum Pooling): Özellik haritasındaki tüm öğelerin toplamı toplam havuzlama olarak adlandırılır.



Şekil 3. 3: Maksimum havuzlama (Max pooling) işlemi

3.3.5 Önyargı ve Varyans

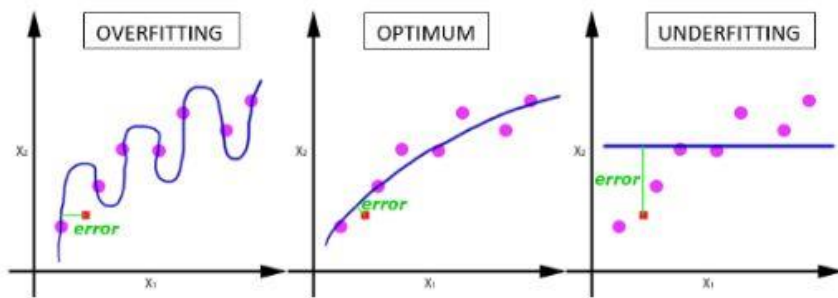
Önyargı-Yanlı (Biased): b ile gösterilir ve modelin yanlılığını ifade eder. Ayrıca girdi değerleri ile ağırlık çarpımından oluşan lineer doğrunun kaymasını yani ötelenmesini sağlayarak en uygun modeli oluşturmaya katkı sağlamaktadır. Bunun yanı sıra, önyargı yani Bias'ın en büyük katkılarından biri, girdi sıfır olduğunda ağırlık ne olursa olsun sonucun sıfıra gitmesini önleyerek, bir miktar öteleme yapıp, modelin bir sonraki yinelemede öğrenmesinin devam etmesini sağlamaktır. Önyargı hatası, ortalamada tahmin edilen değerlerin ne kadarının gerçek değerden farklı olduğunu ölçmek için kullanışlıdır [29].

Varyans: σ^2 ile gösterilir ve değişimi ifade eder. Oluşturulan modeldeki eğitim kümesi üzerinde hesaplanan tahminin, farklı veri kümesindeki değişen miktarını ifade eder [29].

3.3.6 Uyum Durumları

Sinir ağlarında model oluşturulurken veri kümesi genelde 3 gruba ayrılır. Bunlar eğitim (training), doğrulama (validasyon, geliştirme - dev), test veri setlerinden oluşmaktadır. Training/ Eğitim kümesi: Modeli eğittiğimiz veri kümesidir. Dev/ Doğrulama kümesi: Modeli eğittikten sonra, parametreleri değiştirerek performansının ölçüldüğü veri kümesidir. Test kümesi: Nihai modelin değerlendirildiği, nasıl çalıştığının gözlemlendiği veri kümesidir. Genellikle veri setlerinin büyüklükleri büyükten küçüğe şu şekilde sıralanır: Training, Test, Dev.

Oluşturulmak istenen modelde hedef optimum durumu bulabilmektir. Optimum durum, düşük hata oranının yakalanmasıdır. Yani hem eğitim hem de doğrulama veri kümesinde hata oranları küçükse, model verilere uygun şekilde uyarlanmış, düşük varyans ve düşük ön yargıya sahip demektir. Doğrulama kümesindeki hata, eğitim kümesindeki hatadan çok büyük ise model overfit yani aşırı uyumu, ezberlemiş demektir. Bu durumlarda varyans yani değişkenlik yüksek olmaktadır. Hem eğitim hem de doğrulama veri kümesinde hata yüksek ise model uygun değildir, yüksek önyargıya sahip demektir. Eğitim kümesinde hata yüksek ama doğrulama kümesinde hata çok daha yüksek ise hem önyargı hem de değişkenlik çok yüksek demektir. Bu durumlar ise underfitting yani yetersiz uyum olarak adlandırılır. Şekil 3.4'te mor noktalarla belirtilen veri kümesinde, mavi çizgi ile belirtilen modellerin veri kümesine uyum durumları gösterilmiştir [30].



Şekil 3. 4: Veri kümesinde uyum durumları

Modelin yüksek önyargıya sahip olduğu durumlarda, daha büyük sinir ağları kullanılabilir ya da modeller daha uzun süre eğitilebilir. Model yüksek varyansa sahip ise, varyansı azaltmak için daha fazla veri kullanılabilir, farklı sinir ağı mimarileri denenebilir ya da düzenleme – regularize etme tekniği kullanılabilir. Overfitting- aşırı uyumu azaltılması gereken durumlarda kullanılan bir diğer teknik,

cross-validation yani çapraz doğrulamadır. Bu teknik aynı zamanda, modelin hiper parametrelerini belirlemede ve hangi parametrelerin en düşük test hatasına neden olacağı yani hangi parametrelerin daha uygun olacağını belirlemede de kullanılabilir.

3.3.7 Kayıp ve Maliyet Fonksiyonları

Kayıp (Loss) fonksiyonları: Tek bir eğitim örneğine göre tanımlanmıştır. Tek bir eğitim örneğinde modelin ne kadar iyi çalıştığını ölçer [31]. Matematiksel formu, Denklem 3.5' e aktivasyon fonksiyonunun uygulanıp Denklem 3.6 formunu elde ettikten sonra bu değerın Denklem 3.4'te yerine yazılıp, logaritmasının alınarak Denklem 3.7 formunun elde edilmesi şeklindedir.

$$z=w^t x + b \quad (3.4)$$

$$z=y \quad (3.5)$$

$$\bar{y}=g(y) \quad (3.6)$$

$$L(\bar{y}, y)= - (y \log \bar{y} + (1- y) \log(1- \bar{y})) \quad (3.7)$$

Maliyet (Cost) Fonksiyonları: Tüm eğitim kümesinde modelin ne kadar iyi çalıştığını ölçen fonksiyonlardır [31]. Mümkün olduğunca küçük olması istenilir. Formülü Denklem 3.8' de verilmiştir.

$$J(w,b)=\frac{1}{m} \sum_{i=1}^m L(\bar{y}_i, y_i) \quad (3.8)$$

Çapraz entropi (Cross Entropy), bir sınıflandırma modelini eğitmek için bir kayıp fonksiyon olabilir. Çapraz entropi, modelin öngörüsünü gerçek olasılık dağılımı olan etiketle karşılaştırır. Tahmin giderek daha doğru hale geldikçe, çapraz entropi azalır. Cross Entropy Loss'un bir diğer adı 'Negative Log Likelihood' dur. Formülü Denklem 3.9' da verilmiştir.

$$C= \left(-\frac{1}{m}\right) \sum [y \ln(\bar{y}) + (1 - y) \ln(1 - \bar{y})] \quad (3.9)$$

İkili Çapraz Entropi (Binary Cross-Entropy), çapraz entropi formülünün iki olasılık içerdiği durumdur.

Kayıp fonksiyonları, sınıflandırma ve regresyon olmak üzere ikiye ayrılır. Yukarıda açıklanan ve bu çalışmada da kullanılan fonksiyonlar sınıflandırma kayıpları olarak adlandırılır. Makine öğrenmesinde çok sık kullanılan bazı kayıp-hata fonksiyonları daha vardır. Aşağıda verilenler ise regresyon kayıpları olarak adlandırılır.

Mean Squared Error (MSE - Quadratic Loss - L_2 Loss): Adından da anlaşılacağı gibi, Ortalama hata karesi, tahminler ve gerçek gözlemler arasındaki karesel farkın ortalaması olarak ölçülür [31]. Yönlerine bakılmaksızın yalnızca ortalama hata büyüklüğü ile ilgilidir. Ancak, kare alınması nedeniyle gerçek değerlerden uzak olan tahminler, daha az sapma tahminlerine kıyasla ağır bir şekilde cezalandırılır. Matematiksel formülü Denklem 3.10' da verildiği gibidir:

$$MSE = \frac{1}{n} \sum_{i=1}^n (y_i - \bar{y}_i)^2 \quad (3.10)$$

Mean Absolute Error (MAE - L_1 Loss): Öte yandan, ortalama mutlak hata, tahminler ve gerçek gözlemler arasındaki mutlak farklılıkların toplamının ortalaması olarak ölçülür [31]. MSE gibi, yönleri dikkate almadan hatanın büyüklüğünü ölçer. MSE'den farklı olarak MAE, gradyanları hesaplamak için doğrusal programlama gibi daha karmaşık araçlara ihtiyaç duyar. Matematiksel formülü Denklem 3.11' de verilmiştir.

$$MAE = \frac{1}{n} \sum_{i=1}^n |y_i - \bar{y}_i| \quad (3.11)$$

3.3.8 Ölçüm Formülleri

Modelin ne kadar iyi çalıştığını ölçmek için bazı oran ve olasılıklardan faydalanılır. Çizelge 3.1 ile oluşturulan modelin yaptığı tahminler ile verilerin gerçekte olan durumları verilmiştir. Verinin gerçek değeri/etiketi olumsuz iken modelimiz olumlu buluyorsa veya gerçekte olmayanı modelimiz var olarak etiketliyorsa bu 'pozitif yanlış - false positive' ya da 'tip 1 hata - type 1 error' olarak adlandırılır. Benzer şekilde verinin gerçek değeri/etiketi olumlu iken modelimiz olumsuz buluyorsa bu 'negatif yanlış - false negative' ya da 'tip 2 hata - type 2 error' olarak adlandırılır. Bu terimler ve terimlerin oranı modelimizin performansını belirlemede büyük önem taşımaktadır [32].

Doğruluk (Accuracy): Doğruluk, makine öğrenme modelinin ne kadar iyi olduğunu inceleyen bir ölçüdür. Dolayısıyla, doğruluk, doğru tahmin edilen sınıfların öngörülen toplam sınıflara oranıdır, formülü Denklem 3.12'de verilmiştir. En iyi modeli belirlemede tek ölçü olmamalıdır [33].

$$\text{Doğruluk , ACC} = \frac{tp+tn}{tp+tn+fn+fp} \quad (3.12)$$

Çizelge 3. 1: Modelin Olasılıkları

		Tahmin Edilen	
		Olumsuz	Olumlu
Actual- Gerçek Durum	Olumsuz	Doğru Negatif (tn)	Yanlış Pozitif (fp – Tip-1 Hata- Yanlış Alarm)
	Olumlu	Yanlış Negatif (fn - Tip-2 Hata)	Doğru Pozitif (tp)

Doğru Pozitif Oran (Recall, (TPR)): Pozitif tahminlerin kaç tanesinin doğru olduğunun ölçüsü olarak tanımlanmaktadır [33]. Fomülü Denklem 3.13'te verilmiştir.

$$\text{Hassasiyet, TPR} = \frac{tp}{tp+fn} \quad (3.13)$$

Pozitif Öngörü Değeri (Precision, positive predictive value (PPV)): Toplam pozitif durumdan, kaç tane pozitifin doğru tahmin edildiğini ölçer [33]. Fomülü Denklem 3.14'te verilmiştir.

$$\text{Pozitif Öngörü Değeri, PPV} = \frac{tp}{tp+fp} \quad (3.14)$$

Özgüllük - Doğru Negatif Oran (Specificity, true negative rate): Doğru şekilde tanımlanmış olan gerçek negatiflerin oranını ölçer [33]. Fomülü Denklem 3.15'te verilmiştir.

$$\text{Doğru Negatif Oran, TNR} = \frac{tn}{tn+fp} \quad (3.15)$$

F1 Skor: F-skor değerlendirme ölçütü hem pozitif öngörü değerini hem de gerçek pozitif oranını sınıflandırma etkinliğinin bir ölçüsü olarak birleştirir. Yani bu iki ölçümün harmonik ortalamasıdır [33]. Fomülü Denklem 3.16'da verilmiştir.

$$F_1 = \frac{2.PPV.TPR}{PPV+TPR} = \frac{2 tp}{2tp+fp+fn} \quad (3.16)$$

3.3.9 Derin Öğrenmede Kullanılan Hiper Parametreler

Grup Boyutu (Batch Size): Tüm veri kümesi tek seferde modeli beslemek için çok büyüktür. Bu yüzden, tek seferde beslemek yerine birkaç küçük parçaya bölmek ve birkaç seferde beslemek daha pratiktir. Grup işleminde, veri kümesi grup boyutu

olarak verilen değere göre parçalara ayrılır ve her yinelemede modelin eğitimi bu parçalar ile gerçekleştirilmektedir. Bu verilen boyut değerinin büyük olmasının bir avantajı, daha doğru ve uygun gradyan değerinin hesaplanmasına olanak sağlaması ve linerizasyonu azaltmasıdır [34].

Eğitim Tur Sayısı (Epoch): Tüm veri kümesinin sınır ağında aynı anda eğitilmez. Yukarıda belirtildiği gibi küçük parçalara ayrılan veriler, bu parçalar ile sırasıyla eğitime katılır. İlk parça eğitilir, model de başarı ölçümü yapılır, geriye yayılım (backpropagation) ile ağırlıklar güncellenir. Bir sonraki parça ile model tekrar eğitilir ve ağırlıklar güncellenir böylece model için en uygun ağırlık (w) değerleri hesaplanır. Bu eğitim adımlarının her birine epoch denilmektedir. İlk adımlarda başarı oranı genel de düşük olur, adım sayısı arttıkça model öğrenir ve başarı oranı da yükselir [34].

Öğrenme Hızı (Learning rate): Makine öğrenmesinde ağırlıkların, parametrelerin güncellenmesi işlemi geriye yayılım ile yapılmaktadır. Bu işlemde zincir kuralı ile geriye doğru gidilirken türev alarak fark bulunur, bulunan fark belirlenen öğrenme hızı değeri ile çarpılır, çıkan değer ağırlık değerlerinden çıkarılır, böylece güncel ağırlık değerleri hesaplanmış olur [34].

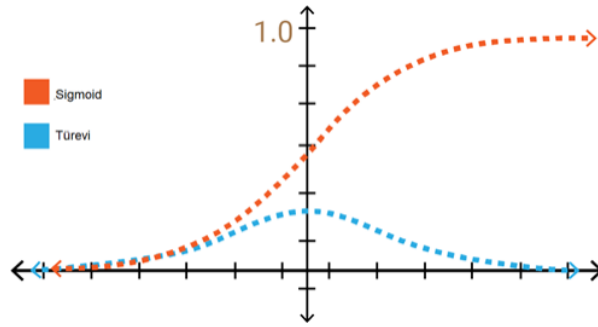
3.3.10 Aktivasyon Fonksiyonları

Aktivasyon fonksiyonları doğrusal olmayan özellikleri sinir ağlarımıza tanıtır. Temel amaçları sinir ağındaki bir düğümün giriş sinyalini bir çıkış sinyaline dönüştürmektir. Buradaki çıkış sinyali bir sonraki katmanda girdi olarak kullanılır. Aktivasyon fonksiyonunun uygulanmadığı durumlarda, çıkış sinyali bir derecelik polinom olan doğrusal bir fonksiyon meydana getirir [35,36]. Polinom derecesi düşük olan fonksiyonların öğrenme gücü de düşük olur. Aktivasyon fonksiyonu uygulanmayan modeller, başarı oranı ve performansı düşük olan bir lineer regresyon modeli oluşturur [35,36,37]. En çok kullanılan aktivasyon fonksiyonları şu şekildedir: Sigmoid, tanh, relu ve softmax.

Sigmoid Aktivasyon Fonksiyonu: Bu aktivasyon fonksiyonu çeşidi, ikili sınıflandırmada çıktı katmanında kullanılır. Çıktı değerlerini, 0 ile 1 değer aralığına dönüştürür, olasılık vermez, değerler toplamı 1 değildir. Çıktı değerlerinden en büyük değer en büyük sayıyı alır ve 1'e yakınsar. Grafiğin uçlarında türev sıfıra yakın olmaktadır, bu durumda degrade iniş parametrelerinin çok yavaş

güncellenmesine neden olur. Ayrıca çıktı sıfır merkezli olmadığından degrade güncellemelerinin farklı yönlerde çok ileri gitmesine neden olur, bu durumda optimizasyonu zorlaştırır. Bu aktivasyon fonksiyonunun bir diğer olumsuz yönü vanishing gradient dediğimiz probleme neden olmaktadır.

Sigmoid fonksiyonu girişinin mutlak değeri ($|x|$) büyüdükçe türev sıfıra yaklaşmaktadır. Sinir ağlarının gradyanları, geri yayılım kullanılarak bulunur [38,39]. Geri yayılma ise ağın türevlerini, zincir kuralı kullanılarak, son katmandan ilk katmana hareket ettirerek bulur. Türev sıfıra yaklaştıkça son katmandan ilk katmana doğru gidilmeye çalışıldığında ilk katmana ulaşılan kadar değer sıfıra çok yaklaşır, bazen katman sayısı çok fazla olduğunda ilk katmana ulaşmak mümkün olmayabilir. Bu ilk katmanlar, girdi verilerinin ana unsurlarını tanıma özelliğini taşıdığından, tüm ağın genel olarak yanlış modellenmesine yol açabilir. Sonuç olarak, kayıp fonksiyonun gradyanlarının sıfıra yaklaşması durumu sinir ağının eğitilmesini zorlaştırır [38,40]. Grafiği Şekil 3.5'te verilmiştir.

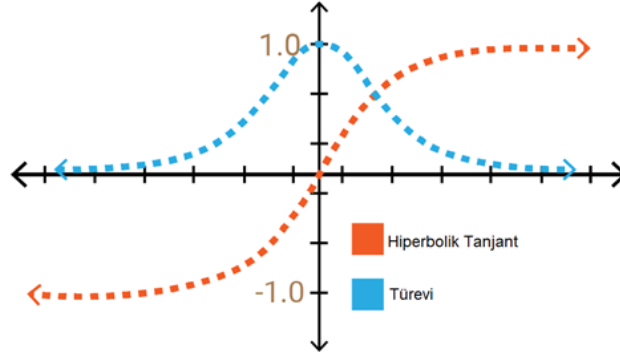


Şekil 3. 5: Sigmoid aktivasyon fonksiyonunun grafiği

Matematiksel formu Denklem 3.17'de verilmiştir.

$$f(x) = \frac{1}{1+\exp(-x)} \quad , \quad 0 < f(x) < 1 \quad (3.17)$$

Tanh (Hiperbolik Teğet) Aktivasyon Fonksiyonu: Bu aktivasyon fonksiyonunda da değerler en uç noktadayken türevler sıfıra yaklaştığından parametreler yavaş güncellenir. Sıfır merkezli olduğundan Sigmoid'e göre daha iyi çalışır ama gradyan kaybolma sorunu bu yöntemde de mevcuttur [41]. Grafiği Şekil 3.6'da verilmiştir.



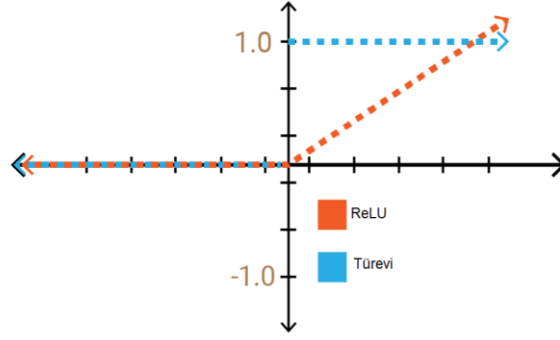
Şekil 3. 6: Tanh aktivasyon fonksiyonunun grafiği

Matematiksel formu Denklem 3.18' de verilmiştir.

$$f(x) = \frac{1 - \exp(-2x)}{1 + \exp(-2x)} = \frac{1 - e^{-2x}}{1 + e^{-2x}}, \quad -1 < f(x) < 1 \quad (3.18)$$

ReLU Aktivasyon Fonksiyonu: Sigmoid ve tanh aktivasyon fonksiyonlarına göre daha az matematiksel işlemler ile daha hızlı sonuçlar alınmasını sağlar. Bu yüzden diğer ikisine göre daha fazla tercih edilir. Fakat bu aktivasyon fonksiyonunun da olumsuz bir özelliği vardır. Grafiğinden görüldüğü üzere işlem yükünden kurtardığı sıfır değer bölgesinde, gradyan hesaplamalarında kullanılan türevlerin sonucu da burada sıfırdır [41].

ReLU ve çeşitleri diğer aktivasyon fonksiyonlarından çok daha hızlıdır. Sıfır bölgesinde yaşanan sorun Leaky ReLU ile çözümlenmiştir. Bu aktivasyon fonksiyonlarının sınırlayıcı özelliği, sadece sinir ağı modelinin gizli katmanları içinde kullanılması gerektiğidir [35]. Dolayısı ile modelde kullanıldığında çıktı katmanları için, sınıflandırma probleminde Softmax aktivasyon fonksiyonu, regresyon probleminde basit doğrusal bir fonksiyon kullanılmalıdır [36]. Bu tez çalışmasında ReLU kullanılmıştır, problemimiz sınıflandırma olduğundan çıktı katmanında da softmax aktivasyon fonksiyonu kullanılmıştır.

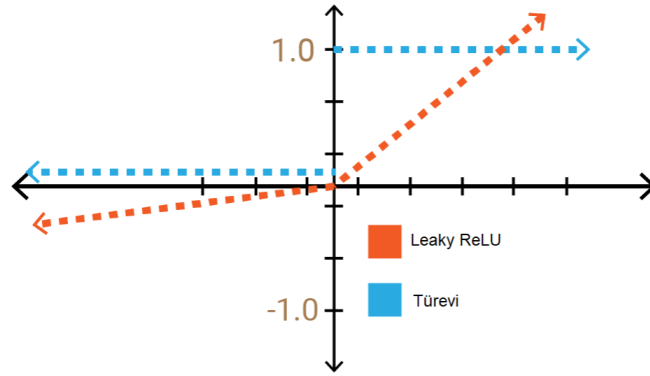


Şekil 3. 7: ReLU aktivasyon fonksiyonunun grafiği

Matematiksel formu Denklem 3.19 ve Denklem 3.20’de verilmiştir.

$$f(x) = \max(0, x) \quad (3.19)$$

$$x < 0 \rightarrow f(x) = 0, \quad x > 0 \rightarrow f(x) = x \quad (3.20)$$



Şekil 3. 8: Leaky- ReLU aktivasyon fonksiyonunun grafiği

Sıfırdan farklı ama sıfıra yakın değer yani sızıntı sayesinde sıfır bölgesinde de öğrenme devam ettirilmiş olur [41].

Softmax Aktivasyon Fonksiyonu: Sigmoid ile benzer bir yapısı vardır, sınıflayıcı olarak çok iyi bir performans gösterir. Çıktı değerlerini (0 – 1 aralığında) olasılık değerlerine dönüştürür. Yani çıktı değerlerinin toplamı 1’ e eşittir. İkiiden fazla sınıflamanın yapıldığı durumlarda tercih edilir [41].

Softmax Aktivasyon Fonksiyonunun Türetilmesi: Softmax aktivasyon fonksiyonu, sınıf etiketleri üzerinde kategorik bir dağılımı temsil etmek için derin sinir ağının çıkış katmanında, etiketlere ait, her giriş elemanının olasılıklarını elde etmek için kullanılır [35,42].

Kategorik dağılım Bernoulli dağılımının çok sınıflı genelleşmesidir. Bernoulli dağılımı, tek bir deneyin sonucunu modelleyen veya rastgele bir değişkenin iki sonucunu alan tek bir gözlemini modelleyen ayrık bir olasılık dağılımıdır. Kategorik dağılım doğal olarak Bernoulli dağılımını ikiden fazla sonucu olan deneylere genişletir [42].

X_i girişlerinde şartlandırılmış- koşullu Y_i çıktısının Bernoulli dağılımı:

$$Y_i | X_{1,i}, X_{2,i}, \dots, X_{m,i} \sim \text{Bernoulli}(P_i) \quad (3.21)$$

$$E[Y_i | X_{1,i}, X_{2,i}, \dots, X_{m,i}] = P_i \quad (3.22)$$

$$\Pr(Y_i = y | X_{1,i}, X_{2,i}, \dots, X_{m,i}) = \{ P_i \rightarrow y=1, 1 - P_i \rightarrow y=0 \} \quad (3.23)$$

$$\Pr(Y_i = y | X_{1,i}, X_{2,i}, \dots, X_{m,i}) = P_i^y (1 - P_i)^{(1-y)} \quad (3.24)$$

Bernoulli sonuçlarının log oranlarını lineer kestiriciye bağlayan link fonksiyonu logit fonksiyonudur:

$$\text{logit}(E[Y_i | X_{1,i}, X_{2,i}, \dots, X_{m,i}]) = \text{logit}(P_i) = \ln(P_i / (1 - P_i)) \quad (3.25)$$

$$\text{logit}(E[Y_i | X_{1,i}, X_{2,i}, \dots, X_{m,i}]) = \text{logit}(P_i) = \beta_0 + \beta_1 X_{1,i} + \dots + \beta_m X_{m,i}$$

Bu denklemi üstel hale getirip düzenlersek;

$$E[Y_i | X_i] = P_i = \text{logit}^{-1}(\beta X_i) = 1 / (1 + e^{-\beta X_i}) \quad (3.26)$$

$$\ln \Pr(Y_i=1) = \beta_1 X_i - \ln Z \quad (3.27)$$

$$\ln \Pr(Y_i=2) = \beta_2 X_i - \ln Z \quad (3.28)$$

...

$$\ln \Pr(Y_i=K) = \beta_K X_i - \ln Z \quad (3.29)$$

Yukarıdaki denklemler sistemindeki $\ln Z$ terimi normalizasyon faktörü, Z ise bölümlenme fonksiyonu olarak bilinir.

$$Y_i | X_i \sim \text{Kategorik}(P_i) \quad (3.30)$$

Her iki tarafı yeniden düzenlersek;

$$\sum_{k=1}^K \Pr(Y_i = k) = 1 \quad (3.31)$$

$$1 = \sum_k \Pr(Y_i = k) = \sum_{k=1}^K \frac{1}{Z} e^{\beta_k X_i} \quad (3.32)$$

Yukarıdaki denklemin sağ tarafındaki eşitliğe Gibbs Ölçümü denir ve Softmax fonksiyonunu istatistiksel mekanik ile birleştirir.

$$Z = \sum_{k=1}^K e^{\beta k x_i} \quad (3.33)$$

Denklemler sistemi şu hale gelir:

$$\Pr(Y_i=1) = \frac{e^{\beta 1 X_i}}{\sum_{k=1}^K e^{Z_k}} \quad (3.34)$$

$$\Pr(Y_i=2) = \frac{e^{\beta 2 X_i}}{\sum_{k=1}^K e^{Z_k}} \quad (3.35)$$

...

$$\Pr(Y_i=K) = \frac{e^{\beta K X_i}}{\sum_{k=1}^K e^{Z_k}} \quad (3.36)$$

Yukarıdaki denklemin sağ tarafı softmax fonksiyonunu verir. Genelleştirilmiş softmax fonksiyonu;

$$\sigma(Z)_j = \frac{e^{Z_j}}{\sum_{k=1}^K e^{Z_k}}, \quad j=1,2,\dots,K \quad (3.37)$$

3.3.11 Regularizasyon ve Yöntemleri

Regularizasyon, model veya verilerde ki önemli özellikleri kaybetmeden, varyansa odaklanıp, varyansı düşürmeyi sağlamaktadır. Bu yöntem, oluşturulmuş modelin, eğitim verilerinden farklı bir veri kümesi için iyi sonuçlar vermediği ya da overfitting (ezberleme) sorunlarında, modeli geliştirmek için kullanılmaktadır [43].

Bu çalışmada üç farklı regularize etme yöntemi kullanılmıştır. Bunlar; Veri Seyreltme (Dropout), Veri Artırma (Data Augmentation), Erken Durdurma (Early Stopping) [43].

Veri Seyreltme (Dropout): Google'ın sinirsel ağlarda aşırı uyumu azaltmak için, eğitim verilerinde karmaşık eş uyarlamaları önleyerek patenti alınmış bir düzenleme tekniğidir. Eğitim sırasında, bazı katman çıktıları rasgele göz ardı edilir veya bırakılır. Dropout, ağda tüm gizli katmanlara veya giriş katmanına uygulanabilir ama çıkış katmanına kesinlikle uygulanamaz. Bu çalışmada, overfitting

yani ezberlemeyi engellemek için, gizli katmanlarda rasgele olarak belirli oranlarda dropout uygulanmıştır.

Veri Artırma (Data Augmentation): Genellikle görüntü sınıflandırma modeli oluşturulurken, çeşitli nedenlerden dolayı gerekli veriler bulunamadığında kullanılan bir tekniktir. Görüntüyü ters çevirme, belirli oranda rasgele döndürme, renk-parlaklık ile değişiklik yapma vb. gibi işlemler uygulanarak veri artırma- veri örneği oluşturma uygulanır. Bu çalışmada stabil olmayan verilerde belirli oranı yakalayabilmek için veri çoğaltma işlemi uygulanmıştır. Bunlar potansiyel olarak daha fazla eğitim verisi elde edilmesini sağlayarak, aşırı öğrenmeyi (overfitting) azaltmaktadır.

Erken Durdurma (Early Stopping) : Büyük bir ağı eğitirken, eğitim sırasında modelin genelleştirmeyi durduracağı ve eğitim veri kümesindeki istatistiksel gürültüyü öğrenmeye başlayacağı bir nokta olacaktır. Eğitim veri kümesinin bu şekilde donatılması, genelleme hatasının artmasına neden olacak ve bu da modeli yeni veriler üzerinde öngörülerde bulunma konusunda daha az kullanışlı hale getirecektir. Buradaki zorluk, ağı, girdilerden çıktılara haritalamayı öğrenebilecek kadar uzun süre eğitmek, ancak modeli eğitim verilerinin üstüne getirecek kadar uzun süre eğitmemek. Ağ çok daha iyi bir hale gelse de, yani eğitim kümesindeki hatalar azaldığında, eğitim sırasında bir noktada gerçekte tekrar daha kötüleşmeye başlar, yani görülmeyen örneklerdeki hata artar. Modelin doğrulama veri kümesindeki performansı düşmeye başlarsa (örneğin, kayıp artmaya başlar veya doğruluk azalmaya başlar), eğitim süreci durur. Eğitimin durdurulduğu andaki model daha sonra kullanılır ve iyi genelleme performansına sahip olduğu bilinmektedir. Bu prosedür erken durdurma olarak adlandırılır ve en yaygın kullanılan sinir ağı düzenlenmesi biçimlerinden biridir. Bu çalışmada da erken durdurma yöntemi kullanılmıştır.

3.3.12 Dereceli Alçalma Kavramı

Dereceli alçalma ya da degrade inişleri olarak bilinen bu işlemler, w ve b parametrelerinin, maliyet fonksiyonunun en aza indirileceği şekilde öğrenilmesine yardımcı olan bir tekniktir [37]. Degrade iniş, optimizasyon yapmak için en popüler algoritmalarından biridir ve sinir ağlarını optimize etmek için en yaygın kullanılan yöntemdir. Bir sonraki Optimizasyon algoritmaları kısmında detaylandırılacaktır. Stochastic, mini-batch, batch olmak üzere üç farklı türü vardır. Stochastic Gradient

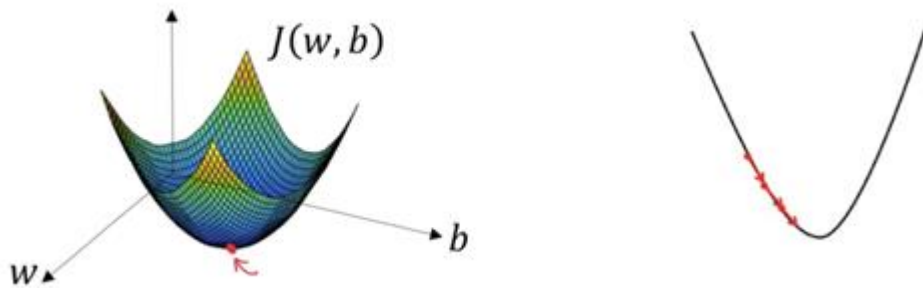
Descent (SDG) denilen yöntemde, her bir eğitim verisi üzerinden maliyet hesaplanır ve ona göre geri yayılım (back propagation) ve parametrelerin güncellemeleri yapılır.

Batch Gradient Descent (BGD) yönteminde ise tüm eğitim verileri vektörize hâlde maliyet fonksiyonuna katkıda bulunur ve ona göre aynı işlem yapılır. Mini-batch Gradient Descent ise ikisinin arasında kalan bir yöntemdir. Yani ne bir ne de tüm veriler üzerinden eğitim yapar. Veri sayısı azsa, genel kabul olarak 2500 kabul edilir, direkt olarak batch yaparak, hızdan kaybetmeden gürültüsüz (smooth) bir cost-iteration değişimine sahip olur. Veri sayısı 2500 den çok daha fazla ise mini-batch önerilir, çünkü çok sayıda veriden oluşan vektörize işlem demek çok fazla hesaplama yükü demektir bu durumda hızdan kaybettirir [37,44].

Gradyan iniş adımları şu şekilde yapılır:

- 1) w ve b başlatılır,
- 2) En dik yokuştan aşağı yönde adım atılır,
- 3) Küresel minimum bulunana kadar bir üstteki durum tekrarlanır.

Aşağıda ulaşılmak istenen, Şekil 3.9'da gösterilen ve kırmızı nokta ile işaretlenen küresel minimuma ulaşılma adımları gösterilmiştir. Burada ilerleme adımını öğrenme hızı denilen α hiper parametresi belirlemektedir. Çok büyük seçilirse hızlı bir iniş gerçekleştirilir ve küresel minimum kaçırılabilir. Çok küçük seçilirse öğrenme çok yavaş gerçekleşeceğinden küresel minimuma ulaşma hızı çok fazla düşer [44].



Şekil 3. 9: Degrade iniş ve küresel minimum

Matematiksel gösterimi:

$$z = w^t x + b \quad (3.4)$$

$$z = y \quad (3.5)$$

$$y' \text{ ye aktivasyon fonksiyonunun uygulanması : } \bar{y} = g(y) \quad (3.6)$$

Tek bir eğitim örneğindeki kayıp:

$$L(\bar{y}, y) = - (y \log \bar{y} + (1 - y) \log(1 - \bar{y})) \quad (3.7)$$

Tüm eğitim örneğindeki kayıp:

$$J(w, b) = \frac{1}{m} \sum_{i=1}^m L(\bar{y}_i, y_i) \quad (3.38)$$

$$J(w^{[1]}, b^{[1]}, w^{[2]}, b^{[2]}, \dots, w^{[m]}, b^{[m]}) = \frac{1}{m} \sum_{i=1}^m L(\bar{y}_i, y_i) \quad (3.39)$$

$$w := w - \alpha \frac{\partial J(w, b)}{\partial w} \quad (3.40)$$

w: degrade iniş sonrası güncellenmiş ağırlık değeri

$$b := b - \alpha \frac{\partial J(w, b)}{\partial b} \quad (3.41)$$

b: degrade iniş sonrası güncellenmiş bias değeri

3.3.13 Optimizasyon Algoritmaları

Makine öğrenmesi, özellikle derin öğrenme metodu işlemlerinin altında optimizasyon problemlerini çözmeye yatkındır [32,42]. Optimizasyon, uyumlaşma, uyumlaştırma anlamındadır [45].

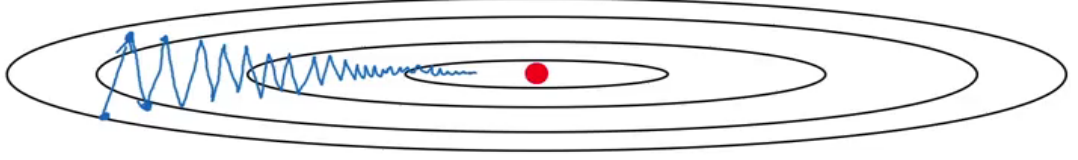
Matematik ve matematiksel programlamalarda optimizasyon terimi; bir gerçek fonksiyonu minimize ya da maksimize etmek amacı ile gerçek ya da tamsayı değerlerini tanımlı bir aralıkta seçip fonksiyona yerleştirerek sistematik olarak bir problemi tanıma, inceleme ya da çözmeye işlemlerini ifade eder [45].

Makine öğrenmesinin temelinde doğrusal olmayan problemler vardır dolayısı ile doğrusal olmayan problemlerin çözümünde ise optimum değeri bulmak için optimizasyon yöntemlerini kullanır [34].

Derin öğrenmede genelde Stochastic Gradient Descent, Momentum, RMSProp, Adam vb. gibi optimizasyon algoritmaları kullanılmaktadır. Bu algoritmalar kendi içerisinde başarı ve hız bakımından ayrışmakta, farklılaşmaktadır [34].

Momentum ile Gradient Descent Optimizasyon Algoritması: Optimizasyon algoritması kullanımındaki amaç küresel minimuma ulaşarak, ağırlıkları güncellemektir [42].

Şekil 3.10'da verilen, kırmızı noktanın küresel minimumu temsil ettiği varsayıldığında, bu noktaya ulaşmak için dikey yönde daha yavaş, yatay yönde ise daha hızlı öğrenmek gerekir. Şekil 3.10'da gösterimi verilmiştir [42].



Şekil 3. 10: Optimizasyon algoritması ve küresel minimum noktası

Matematiksel gösterimi:

t iterasyon ile d_w ve d_b değerleri hesaplanır,

$$V_{dw} = \beta * V_{dw} + (1-\beta) * d_w \quad (3.42)$$

$$V_{db} = \beta * V_{db} + (1-\beta) * d_b \quad (3.43)$$

Ağırlıklar güncellenir,

$$w = w - \alpha * V_{dw} \quad (3.44)$$

$$b = b - \alpha * V_{db} \quad (3.45)$$

Burada bulunan α ve β hiper parametrelerdir. α değeri, öğrenme hızıdır, ayarlanabilir bir parametredir. d_w ve d_b 'nin denklemdeki rolü ise momentum sağlamaktır, V_{dw} ve V_{db} ise hız sağlamaktadır. β değeri ise sürtünme görevi görür, limitin üzerinde hızlanmayı önler. Degrade inişlerin küresel minimayı kaçırarak kadar hızlanmasını istemiyoruz ve bu nedenle sürtünme görevi eklenir.

RMSprop (root mean square error propability) Optimizasyon Algoritması: Benzer şekilde küresel minimayı bulmak için kullanılan bir diğer optimizasyon algoritmasıdır.

Matematiksel gösterimi:

t iterasyon ile d_w ve d_b değerleri hesaplanır,

$$S_{dw} = \beta_2 * S_{dw} + (1-\beta_2) * d_w^2 \quad (3.46)$$

$$V_{db} = \beta_2 * S_{db} + (1-\beta_2) * d_b^2 \quad (3.47)$$

Ağırlıklar güncellenir,

$$w = w - \alpha * (d_w / S_{dw}) \quad (3.48)$$

$$b = b - \alpha * (d_b / S_{db}) \quad (3.49)$$

Bu optimizasyon algoritmasında, dikey yönde eğim daha dik olup, büyük bir S_{db} değeri ile sonuçlanır. Dikey yönde yavaş öğrenme istendiği için, d_b 'yi S_{db} 'im güncelleme adımına bölmek, b 'de daha küçük bir değişikliğe neden olacaktır. Benzer şekilde, S_{dw} değeri yatay yönde daha hızlı öğrenmeye neden olacak ve böylece algoritmayı daha hızlı hale getirecektir.

ADAM (Adaptive Moment Estimation) Optimizasyon Algoritması: Adam ya da uyarlanabilir moment tahmini olarak bilinen bu optimizasyon algoritması adından da anlaşılacağı üzere momentum ve RMSprop optimizasyon algoritmalarının birleşimidir. En yaygın kullanılan ve en hızlı sonuç veren algoritmalarından biridir. Bu çalışmada Adam optimizasyon algoritması kullanılmıştır.

Bu algoritmada kullanılan hiper parametreler şunlardır:

Öğrenme hızı (learning rate) : α , ayarlanabilir bir parametredir,

Momentum terimi: β_1 , yaygın olarak 0.9 seçilir,

RMSprop terimi: β_2 , yaygın olarak 0.999 seçilir,

$\epsilon = 10^{-8}$

Matematiksel gösterimi:

$$V_{dw} = \beta_1 * V_{dw} + (1 - \beta_1) * d_w \quad (3.50)$$

$$V_{db} = \beta_1 * V_{db} + (1 - \beta_1) * d_b \quad (3.51)$$

$$S_{dw} = \beta_2 * S_{dw} + (1 - \beta_2) * d_w^2 \quad (3.52)$$

$$S_{db} = \beta_2 * S_{db} + (1 - \beta_2) * d_b^2 \quad (3.53)$$

Bias düzeltmesi uygulanır,

$$V_{dw}^{\text{corrected}} = V_{dw} / (1 - \beta_1^t) \quad (3.54)$$

$$V_{db}^{\text{corrected}} = V_{db} / (1 - \beta_1^t) \quad (3.55)$$

$$S_{dw}^{\text{corrected}} = S_{dw} / (1 - \beta_2^t) \quad (3.56)$$

$$S_{db}^{\text{corrected}} = S_{db} / (1 - \beta_2^t) \quad (3.57)$$

Ağırlıklar Güncellenir,

$$w = w - \alpha * (V_{dw}^{\text{corrected}} / S_{dw}^{\text{corrected}} + \epsilon) \quad (3.58)$$

$$b = b - \alpha * (V_{db}^{\text{corrected}} / S_{db}^{\text{corrected}} + \epsilon) \quad (3.59)$$

Spesifik olarak, algoritma gradyan ve kare gradyanın üstel bir hareketli ortalamasını hesaplar ve beta1 ve beta2 parametreleri bu hareketli ortalamaların bozulma oranlarını kontrol eder. Epsilon, sifıra bölünmeyi engellemek için kullanılır ve çok küçük bir değer alır.

3.3.14 Toplu Normalleştirme

Toplu normalleştirme (batch normalization), sinir ağlarının performansını ve dengesini artırmak için uygulanan bir tekniktir [45]. Aynı zamanda bu teknik, daha gelişmiş derin öğrenme mimarilerinin pratikte çalışmasını sağlar.

Toplu normalleştirme, her bir fully-connected layer'dan sonra, aktivasyon fonksiyonu ve dropout işleminden önce uygulanmalıdır. Bu işlemin arkasındaki amaç ağ eğitimi optimize etmektir.

Toplu normalleştirme, gizli birim değerlerinin çevresinde kaydığı miktarı (kovaryans değişimi) azaltır. Ayrıca, toplu normalleştirme bir ağın her katmanının diğer katmanlardan bağımsız olarak biraz daha fazla öğrenmesini sağlar. Hafif bir düzenleme etkisine sahip olduğundan aşırı uyumu (overfitting) azaltır.

Seyreltme (dropout) işlemine benzer şekilde, her gizli katmanın aktivasyonlarına biraz gürültü ekler. Önceden eğitilmiş bir ağa bir parti normu eklenirse, önceden eğitilmiş ağırlıkları değiştirir, çünkü ortalamayı çıkarır ve aktivasyon katmanları için standart sapma ile böler. Bunun olması istenilmez, çünkü önceden eğitilmiş ağırlıkların aynı kalması gerekir. Dolayısı ile bir toplu iş norm katmanı yerleştirilir ve çıktılarının değişimini geri almak için gama ve beta bulunur [46].

Toplu normalleştirme şu şekilde yapılır:

$$\mu = \frac{1}{m} \sum_{i=1}^m Z^{[i]} \quad (3.60)$$

$$\sigma^2 = \frac{1}{m} \sum_{i=1}^m (Z^{[i]} - \mu)^2 \quad (3.61)$$

$$Z_{\text{norm}}^{[i]} = \frac{Z^{[i]} - \mu}{\sqrt{\sigma^2 + \epsilon}} \quad (3.62)$$

$$\hat{Z}^{[i]} = \gamma Z_{\text{norm}}^{[i]} + \beta \quad (3.63)$$

Yukarıdaki formüllerde kullanılan γ ve β öğrenilebilir parametrelerdir.

Bir sinir ağının kararlılığını artırmak için parti normalizasyonu, parti ortalamasını çıkartarak ve parti standart sapmasına bölünerek önceki bir aktivasyon katmanının çıkışını normalleştirir. Bununla birlikte, rastgele başlatılan bazı parametreler ile bu aktivasyon çıkış ölçeklerinin / kaymalarının ardından, bir sonraki katmandaki ağırlıklar artık optimum değildir. SGD (Stokastik gradyan inişi), kayıp fonksiyonunu en aza indirmenin bir yoluysa, bu normalleşmeyi geri alır. Sonuç olarak, parti normalizasyonu, her katmana iki eğitilebilir parametre ekler, böylece normalleştirilmiş çıktı bir standart sapma parametresi (gama) ile çarpılır ve ortalama bir parametre (beta) eklenir. Başka bir deyişle, toplu normalleştirme SGD'nin, tüm ağırlıkları değiştirerek ağın dengesini kaybetmek yerine, her aktivasyon için sadece bu iki ağırlığı değiştirerek denormalizasyonu yapmasını sağlar [45,46].

3.3.15 Ayrılabilir Konvolüsyon

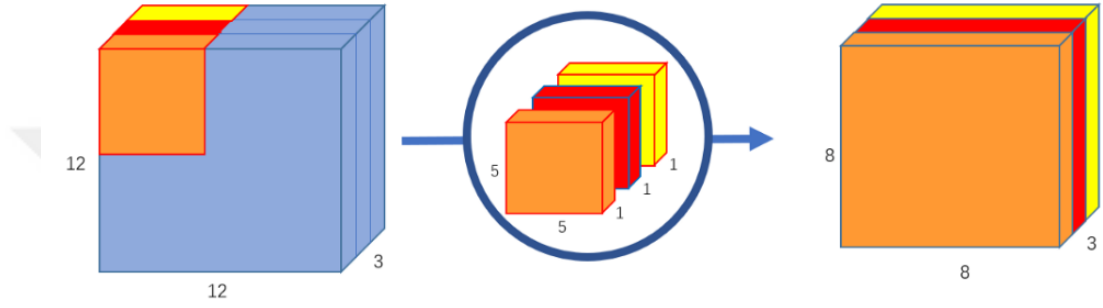
Mekansal-Uzamsal ayrılabilir evrişim (spatial separable convolution): Bir görüntünün ve çekirdeğin genişlik ve yükseklik olmak üzere mekansal boyutlarıyla ilgilenir [47,48]. Uzamsal olarak ayrılabilir bir evrişim, bir çekirdeği iki küçük çekirdeğe böler. En yaygın durum, 3x3 çekirdeği 3x1 ve 1x3 çekirdeğe bölmektir [49]. Bu şekilde bölündüğünde, 9 çarpım ile bir evrişim yapmak yerine, aynı etkiyi elde etmek için her biri 3 çarpım ile iki evrişim (toplamda 6 çarpım) yaparak aynı sonuca ulaşmış olur. Daha az çarpma ile hesaplama karmaşıklığı azalır ve ağ daha hızlı çalışır [50].

Derinlemesine Ayrılabilir Dönüşümler (Depthwise Separable Convolutions): Uzamsal olarak bölünebilen konvolüsyonların aksine, derinlemesine ayrılabilen konvolüsyonlar, iki küçük çekirdeğe çarpanlara giremeyen çekirdeklerle çalışır [49]. Bu nedenle, daha yaygın olarak kullanılır. Bu, keras.layers.SeparableConv2D veya tf.layers.separable_conv2d'de görülen ayrılabilir evrişim türüdür [50].

Derinlemesine ayrılabilir evrişim, uzaysal boyutların yanı sıra derinlik boyutu yani kanal sayısı ile de ilgilenmektedir. Bir giriş görüntüsünün RGB olarak adlandırılan 3 kanalı olabilir. Birkaç konvolüsyondan sonra, bir görüntünün birden fazla kanalı olabilir. Her kanalı o görüntünün özel bir yorumu olarak görüntüleyebilirsiniz; Örneğin, kırmızı kanal her pikselin kırmızılığını, mavi kanal her pikselin maviliğini, yeşil kanal ise her pikselin yeşilliğini yorumluyor. 64 kanallı bir görüntünün, bu görüntünün 64 farklı yorumu vardır. Derinlemesine ayrılabilir evrişim, bir çekirdeği

iki evrişim yapan iki ayrı çekirdekten oluşturmaktadır, bunlar: derinlemesine evrişim ve noktadan evrişim [50].

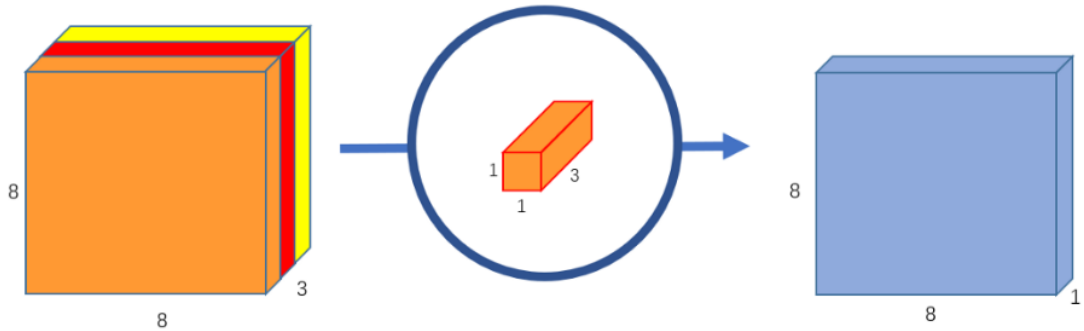
Derinlikli Konvolüsyon (Depthwise Convolution): Bu evrişim kısmı, derinlemesine evrişimin ilk aşamasıdır. Aşağıda $12 \times 12 \times 3$ şeklinde alınan bir görüntü, $5 \times 5 \times 1$ çekirdeğinden geçirilerek, $8 \times 8 \times 3$ şeklinde bir imaj elde edilir. Her $5 \times 5 \times 1$ çekirdeği, görüntünün 1 kanalını yineler (not: 1 kanal, tüm kanallar değil), her 25 piksel grubunun skaler ürünlerini alarak $8 \times 8 \times 1$ görüntü verir. Bu görüntülerin birlikte istiflenmesi $8 \times 8 \times 3$ görüntü oluşturur.



Şekil 3. 11: Derinlemesine evrişim gösterimi

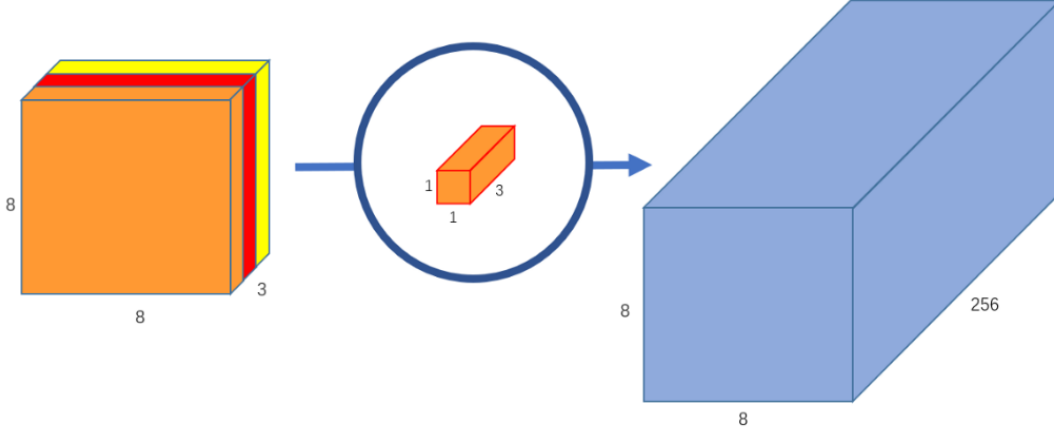
Şekil 3.11’ de $12 \times 12 \times 3$ görüntüyü $8 \times 8 \times 3$ görüntüye dönüştürmek için 3 çekirdek kullanır [50].

Noktasal Konvolüsyon (Pointwise Convolution): İlk kısımda derinlikli konvolüsyon $12 \times 12 \times 3$ görüntüsünü $8 \times 8 \times 3$ görüntüsüne dönüştürür, bu kısımda ise her görüntünün kanal sayısı artırılır. Noktasal evrişim, 1×1 çekirdeği veya her bir noktayı yinleyen bir çekirdeği kullanması nedeniyle adlandırılır. Bu çekirdek, giriş görüntüsünün sahip olduğu birçok kanalın derinliğine sahiptir. Burada kanal sayısı 3 olur. $8 \times 8 \times 1$ imajı elde etmek için, $8 \times 8 \times 3$ boyutundaki görüntü boyunca $1 \times 1 \times 3$ çekirdeği yinelenir.



Şekil 3. 12: Noktasal evrişim

Şekil 3.12’de 3 kanal görüntüsünü 1 kanal görüntüsüne dönüştürülmesi gösterilmiştir [50]. 8x8x256 şeklindeki son görüntüyü elde etmek için her biri 8x8x1 görüntü veren 256 tane 1x1x3 çekirdek oluşturulur.



Şekil 3. 13: 256 çekirdekli noktasal konvolüsyon

Şekil 3.13’te 256 çekirdekli noktasal konvolüsyonun, 256 kanallı görüntü çıktısı oluşturması gösterilmiştir [50].

Özet ile şu şekilde karşılaştırılabilir;

normal konvolüsyon işlevi: $12 \times 12 \times 3 \rightarrow (5 \times 5 \times 3 \times 256) \rightarrow 8 \times 8 \times 256$ ise,

ayrılabilir konvolüsyon işlevi: $12 \times 12 \times 3 \rightarrow (5 \times 5 \times 1 \times 1) \rightarrow (1 \times 1 \times 3 \times 256) \rightarrow 8 \times 8 \times 256$

olarak gösterebiliriz.

Bu çalışmada yukarıda anlatılan normal konvolüsyonel sinir ağları yerine derinlemesine ayrılabilir sinir ağları kullanılmıştır. Derinlemesine ayrılabilir bir evrişim yaratmanın amacı hız ve performanstan kazanç sağlayarak verimliliği artırmaktır. Makine öğrenmesinde yaşanan en büyük sıkıntılardan biri öğretmek için uzun süre gereksinimin yanı sıra, işlem ve hesaplama yükünden dolayı donanımsal olarak daha kapsamlı bilgisayarlara ihtiyaç duyulmasıdır. Burada bu konvolüsyon ile hesaplanan parametre sayısını milyonlardan binlere düşürülerek büyük bir hız kazanılır. Ayrılabilir konvolüsyon da, görüntü sadece bir kez derinlemesine evrişimde dönüştürülür. Ardından, dönüştürülmüş görüntü çekilir ve 256 kanala genişletilir. Görüntüyü tekrar tekrar dönüştürmek zorunda kalmadan, hesaplama gücünden tasarruf sağlanır, performans artırılır.

3.4 Özellik Haritalarının Görselleştirilmesi

Aktivasyon ya da özellik haritası, belirli evrişim katmanının bir çıktısıdır. CNN'nin görselleştirilmesi için aktivasyon haritalarından faydalanılır [51]. Derin CNN modelin giriş görüntüsünü nasıl sınıflandırabildiğini anlamak ve konvolüsyon ağının görüntülerde aradığı şeyleri öğrenmek için, modelin ara katmanları, filtre çıktıları görselleştirilmektedir. Bu sayede modelin giriş görüntüsünün ara katmanlarda nasıl görüldüğü ve her filtrenin yanıt vereceği kalıplar hakkında bilgi edinilmektedir [52]. Böylece, bu katmanların çalışması hakkında daha fazla şey öğrenilebilir. Konvolüsyon katmanlarının filtre çıktıları incelendiğinde, farklı katmanlardaki farklı filtrelerin, görüntünün farklı bölümlerini vurgulamaya veya etkinleştirmeye çalıştığı anlaşılmaktadır. Bazı filtreler kenar dedektörleri olarak işlev görürken, bazıları merkez kısmı gibi rötgen görüntüsünün belirli bir bölgesinde yoğunlaşır, bir kısmı ise arka plan dedektörleri gibi davranır [51,52]. Evrişim katmanlarının bu davranışını başlangıç katmanlarında görmek daha kolaydır, çünkü ağ derinleştikçe evrişim çekirdeği tarafından yakalanan desen giderek daha da seyrekleşir ve çıktıların görüntüsünün giriş görüntüsünden tamamen farklılaşabildiği görülmektedir.

Bu tez çalışmasında incelenen her modelin ara filtrelerinin çıktıları görselleştirilerek bu filtrelerin uygulanan modellerde görüntüleri sınıflandırmak için neyi aradığı incelenmiştir.

3.5 Veri Artırma İşlemi

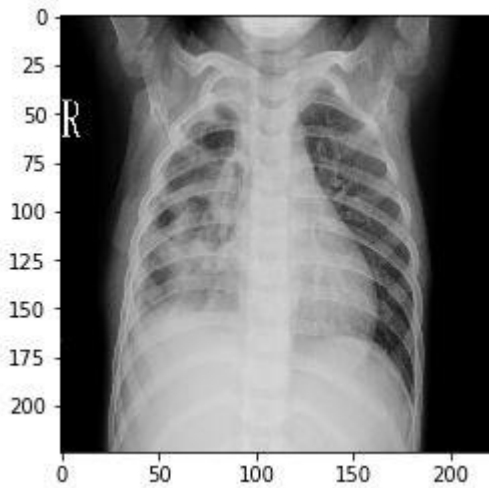
Görüntü verilerini artırma, veri kümesinde görüntülerin değiştirilmiş sürümlerini oluşturarak bir eğitim veri kümesinin boyutunu yapay olarak genişletmek için kullanılacak bir tekniktir [53]. Derin öğrenme sinir ağı modellerini, daha fazla veri konusunda eğitmek, daha yetenekli modellere neden olabilir ve veri artırma teknikleri, uygun modellerin yeni görüntüler için öğrendiklerini genelleştirme becerisini geliştirebilecek görüntüler üzerinde çeşitlilikler yaratabilir [53,54]. Keras derin öğrenme sinir ağı kütüphanesi, ImageDataGenerator sınıfı aracılığıyla görüntülerde veri artırma sağlanabilir.

Veri artırma işlemi, ters çevirme, kaydırma, gürültü ekleme, yakınlaştırma, bulanıklaştırma vb. görüntü işleme alanındaki bir dizi işlemi içerir. Amaç, eğitim veri kümesini yeni ve makul örneklerle genişletmektir [53]. Bu, eğitim kümesi

görüntülerinin model tarafından görülmesi muhtemel olan varyasyonları anlamına gelir [53,54]. Örneğin, bir kedinin resminin yatay bir şekilde çevrilmesi, fotoğrafın soldan veya sağdan çekildiği için mantıklı gelebilir. Bir kedinin fotoğrafının dikey olarak çevrilmesi anlamsızdır ve muhtemelen modelin baş aşağı bir kedinin fotoğrafını görmesi pek mümkün olmadığından uygun olmayabilir. Bu nedenle, bir eğitim veri kümesi için kullanılan spesifik veri artırma tekniklerinin seçiminin dikkatli bir şekilde ve eğitim veri kümesi bağlamında ve problemin alanı bilgisi dahilinde seçilmesi gerekmektedir [53,54].

Görüntü veri artırma işlemi genellikle eğitim veri kümesine uygulanır, doğrulama veya test veri kümesine uygulanmaz. Bu, görüntü yeniden boyutlandırma ve piksel ölçeklendirme gibi veri hazırlığından farklıdır; modelle etkileşime giren tüm veri kümelerinde tutarlı bir şekilde gerçekleştirilmeleri gerekir. Keras derin öğrenme kütüphanesi, bir modeli eğitirken veri artırmayı otomatik olarak kullanma yeteneği sağlar. Bu, ImageDataGenerator sınıfını kullanarak elde edilir. İlk olarak, sınıf başlatılabilir ve veri artırma türlerinin yapılandırması, sınıf yapıcısının argümanlarıyla belirlenir.

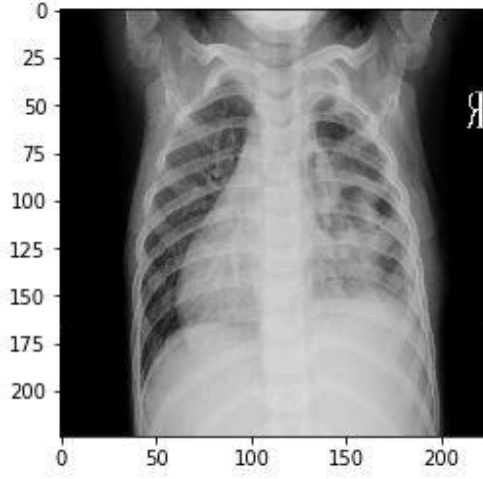
Şekil 3.14'te bu tez çalışmasında uygulanmış veri artırma işlemlerine göre çıktısı elde edilecek bir röntgen görüntüsü verilmektedir.



Şekil 3. 14: Veri artırma işlemi uygulanacak bir akciğer röntgeni

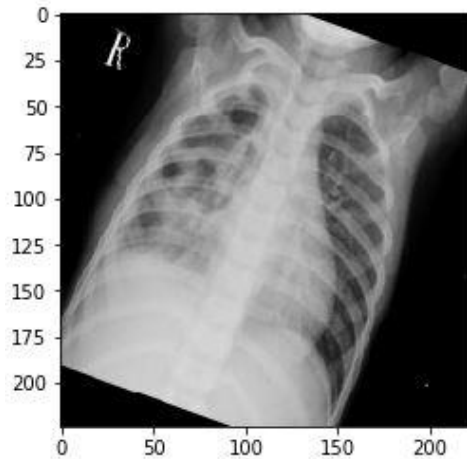
Yatay veya Dikey Çevirme : Bir görüntü çevirme, sırasıyla dikey veya yatay çevirme durumunda piksel satırlarını veya piksel sütunlarını ters çevirmek anlamına gelir. Çevirerek (Flipping) veri artırma işlemi, ImageDataGenerator sınıf yapıcısına bir bool horizontal_flip veya vertical_flip argümanı ile belirlenir [53]. Elimizdeki

medikal görüntüler için yatay çevirme uygun fakat dikey çevirme uygun değildir. Şekil 3.15'te veri kümesimizde röntgen görüntüsünün `horizontal_flip` argümanı, yatay çevrim ile veri artırılması gösterilmiştir.



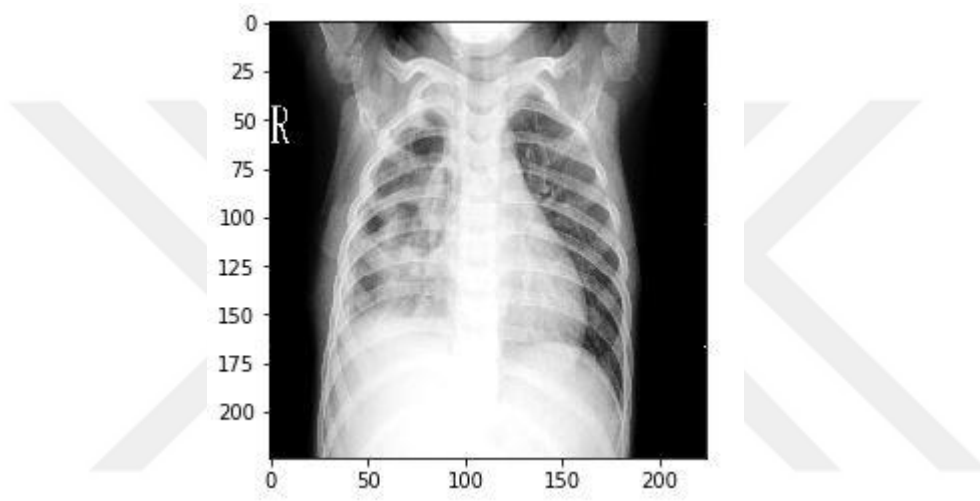
Şekil 3. 15: Yatay çevirme ile veri artırma (Horizontal Flipping)

Rastgele Rotasyon Verme: Bir rotasyon artırması rastgele görüntüyü 0 ila 360 derece arasında, belirli bir derece saat yönünde döndürür. Dönme büyük olasılıkla pikselleri görüntü çerçevesinin dışına döndürür ve doldurulması gereken piksel verisi olmayan karenin alanlarını bırakır. Veri kümesi için mantıklı olan rotasyon miktarına göre bu değer verilmelidir. Üzerinde çalışılan medikal veriler için 10 ile 20 derece arası rotasyon klinik hekimleri tarafından uygun görülmüştür. Bu tez çalışmasında ise veriler arasındaki dağılım dengesini sağlamak için veri artırma işlemi uygulanmış olup rotasyon olarak 20 derece verilmiştir. Şekil 3.16'da rotasyon işlemi uygulanmış bir röntgen görüntüsü gösterilmiştir.



Şekil 3. 16: Rotasyon ile veri artırma işlemi

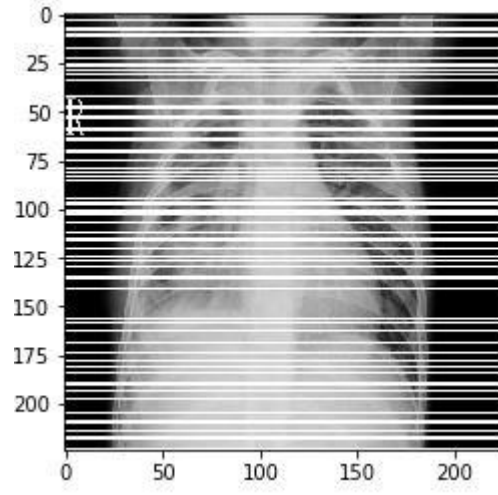
Rastgele Parlaklık Artırma: Görüntünün parlaklığı, resimleri rasgele karartmak, görüntüleri parlatmak veya her ikisiyle de artırılabilir. Amaç, bir modelin farklı aydınlatma seviyelerinde eğitilmiş görüntüler arasında genelleştirilmesine izin vermektir. Bu, aydınlık miktarını seçmek için bir yüzdeyi temsil eden bir değişken olarak minimum ve maksimum aralığı belirten ImageDataGenerator () yapıcısına brightness_range argümanını belirleyerek elde edilebilir. 1.0'dan küçük değerler, görüntüyü karartır, örn. [0.5, 1.0], 1.0'dan büyük değerler görüntüyü aydınlatır, örn. [1.0, 1.5], burada 1.0 parlaklık üzerinde hiçbir etkisi yoktur. Şekil 3.17'de parlaklığı artırılarak elde edilmiş görüntü gösterilmektedir.



Şekil 3. 17: Parlaklık artırma ile veri artırma işlemi

Gauss Gürültüsü: Sıfır ortalamaya sahip olan Gauss gürültüsü, esas olarak tüm frekanslarda veri noktalarına sahiptir ve yüksek frekans özelliklerini etkili bir şekilde bozar. Bu aynı zamanda daha düşük frekans bileşenlerinin (genellikle amaçlanan verilerinizin) de çarpıtıldığı anlamına gelir, ancak sinir ağınız bundan geçmiş olmayı öğrenebilir. Doğru miktarda gürültü eklemek, öğrenme yeteneğini artırabilir.

Veri Seyrekleştirme-Bulanıklaştırma: Görüntülerin bir kısmını çıkarma ya da görüntüyü bulanıklaştırma işlemleri de veri artırmayı sağlamaktadır. Sonuçta bu görüntülerde işlemek istenilen görüntülerin oluşabilecek varyasyonları arasındadır. Şekil 3.18'de görüntü gürültü eklenip, veri seyrekleştirme ve bulanıklaştırma uygulanarak elde edilmiş bir veridir.



Şekil 3. 18: Veri seyrekleştirme-gürültü ekleme ile veri artırma işlemi





4.DENEYLER

4.1 Model Tasarımları ve Parametre Değerleri

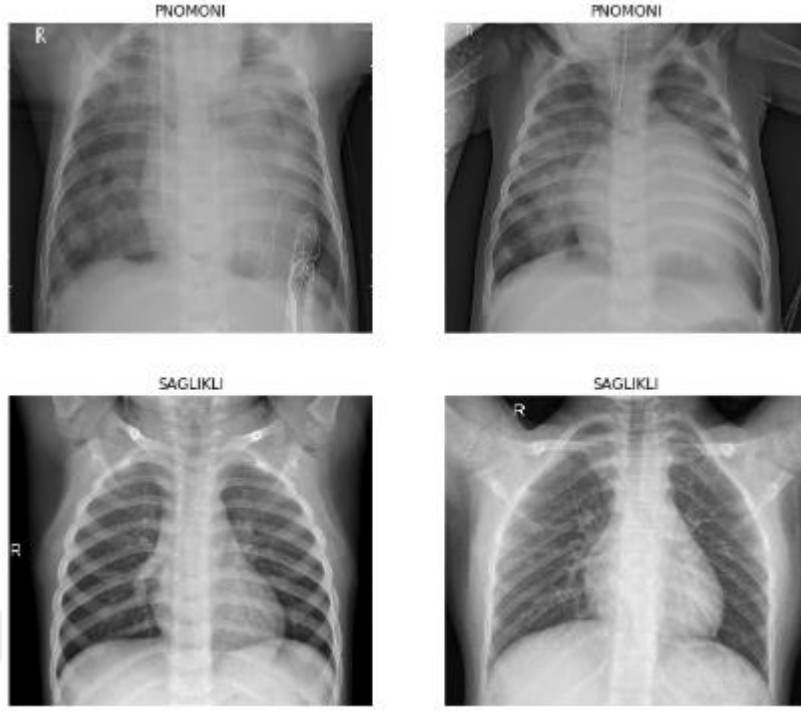
Üzerinde çalışılacak olan veri kümesi sağlıklı ve pnömoni hasta etiketlerinden oluşmaktadır. Normal olan veriler sıfır (0), Pnömoni olan veriler bir (1) olarak etiketlenmiştir. Bu tez çalışmasında dört farklı model kurulmuştur. Her bir model için ayrıca k-fold uygulanarak çapraz doğrulamanın etkisi ve veri artırma işleminin başarı oranlarına katkısı gözlemlenmiştir. Daha önce anlatıldığı gibi veri dağılımı hem dengesiz hem de kirliliği içermektedir. İlgili verileri kullanabilmek için öncelikle verilerin içerisinde kanalı bozulmuş veriler tespit edilmiştir.

Tüm görüntülerin kanal boyutları incelenmiştir. Bir kanallı görüntülere kanal eklenerek kanal sayısı üçe çıkarılmıştır. Kullanılan kütüphanelerin kanal bit sıralaması göz önünde bulundurularak düzenlemeler yapılmıştır. Opencv kütüphanesinin kullanıldığı yerlerde BGR dönüşümü, numpy kullanılan yerlerde RGB dönüşümü yapılmıştır.

Pandas kütüphanesi kullanılarak eğitim veri kümesinin dağılımı incelenmiştir. Eğitim kümesinin dağılımında, normal etiketli verilerin az, pnömoni etiketli verilerin ise dağılımında yoğunluk görülmüştür. Bu veri dağılım dengesizliğini önleyebilmek için daha önce anlatılan veri artırma yöntemleri uygulanmıştır. Bu yöntem sadece normal verilere uygulanarak, veri dağılımı dengesizliği çözümlenmeye çalışılmıştır.

Görüntüler (224, 224) olacak şekilde yeniden boyutlandırılmış olup, görüntü pikselleri normalize edilmiştir. Veri hazırlığı tamamlandıktan sonra modeller oluşturulmaya başlanmıştır.

Aşağıda Şekil 4.1'de veri kümesinden alınmış sağlıklı ve pnömonili görüntüler verilmiştir.

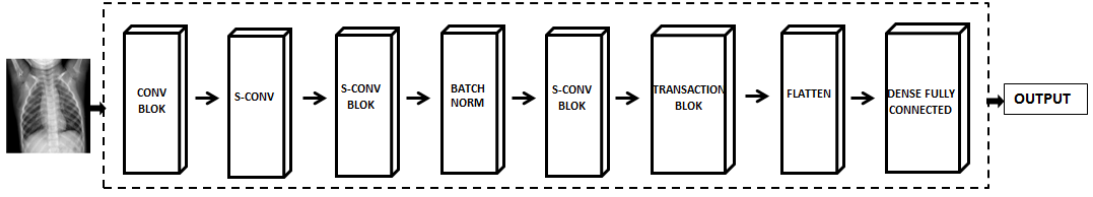


Şekil 4. 1: Veri kümesinden örnek görüntüler

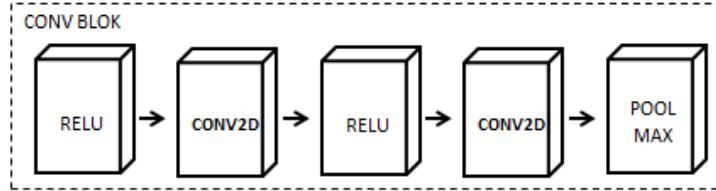
4.1.1 CNN Modeli

CNN ile model oluşturma aşamasında veri kümesi rastgele olarak, %70 eğitim, %20 test, %10 doğrulama olmak üzere üçe bölünmüştür.

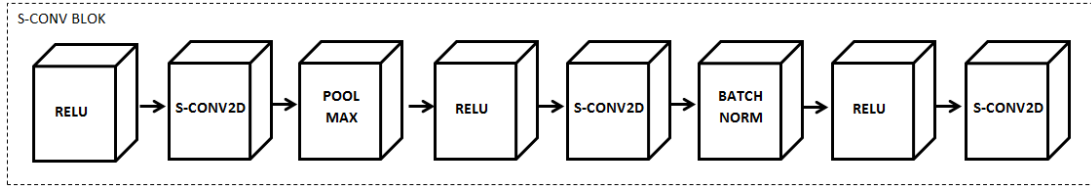
Girdi olarak (224, 224, 3) boyutlu görüntüler kullanılmıştır. Bu çalışma girdi katmanı, dört ara-gizli katman, üç tamamen bağlı katmandan oluşturulmuştur. Her bir ara katmanda havuzlama olarak max-pooling kullanılmıştır. Piksel ekleme yapılmamış olup, aynı bırakılmıştır. Üçüncü ve dördüncü katmanlarda toplu normalleştirme işlemi uygulanmıştır. En son ki tamamen bağlı katmanda yani çıktı katmanında aktivasyon fonksiyonu olarak softmax diğer katmanların tamamında ise ReLu kullanılmıştır. İlk tamamen bağlı katmanda 0.7, ikinci tamamen bağlı katmanda ise 0.5 oranında veri seyreltme uygulanmıştır. Model tasarlandıktan sonra ağırlık başlatma işlemi uygulanmıştır. İlk iki konvolüsyonun ağırlığı, ImageNet ağırlıkları ile başlatılmıştır. Optimizasyon işlemi olarak Adam kullanılmıştır. Öğrenme hızı olarak 0.001 , erken durdurma değeri olarak 5 tanımlanmıştır. Grup sayısı (batch size) 16, eğitim tur sayısı (epoch) ise 20 olarak belirlenmiştir. Oluşturulan modelin mimarisi aşağıdaki Şekil 4.2, Şekil 4.3, Şekil 4.4, Şekil 4.5, ve Şekil 4.6'da gösterilmiştir. Parametre değerleri ise Çizelge 4.1'de verilmiştir.



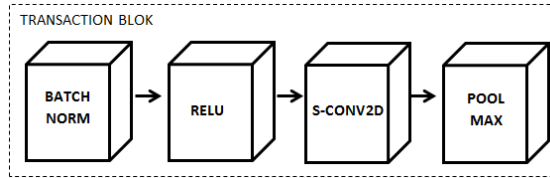
Şekil 4. 2: Sıfırdan oluşturulmuş CNN modelinin mimarisi



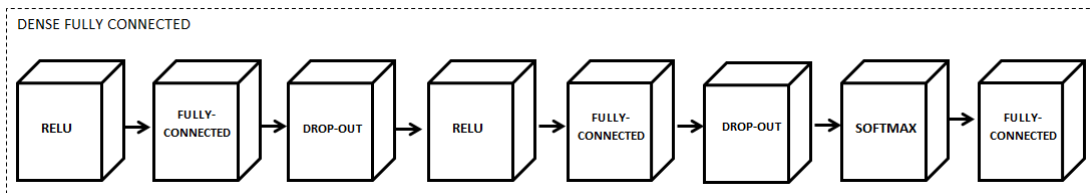
Şekil 4. 3: CNN modelinin Conv Blok mimarisi



Şekil 4. 4: CNN modelinin S-Conv Blok mimarisi



Şekil 4. 5: CNN modelinin Transaction Blok mimarisi



Şekil 4. 6: CNN modelinin Dense Fully Connected mimarisi

Oluşturulan modelin parametre ayarları doğrulama veri kümesinde düzenlendikten sonra, test veri kümesinde çalıştırıldı. Test veri kümesinde yaklaşık %83 doğruluk oranı yakalandı. Medikal veri dengesiz olduğundan diğer başarı oranlarını da detaylı incelemek gerekir. Bu yüzden gerçek pozitif oran ve pozitif öngörü değerlerine de bakmak gerekir. İncelediğimizde bu değerlerin 0.98 ve 0.80 olduğunu görüyoruz. Bu

iki deęerin harmonik ölçümü olan F1 skoruna baktığımızda ise bu deęerin 0.88 hesaplandığını görüyoruz.

Aynı zamanda derinlemesine ayrılabilir evrişimin zaman açısından ciddi performans sağlamıştır. Oluşturulan bu makine öğrenmesinin, ortalama bir bilgisayar modelinde, yaklaşık olarak 7 dakika da tamamlanmıştır. Modelin başarı ölçüm deęerleri ise Çizelge 4.2’de verilmiştir.

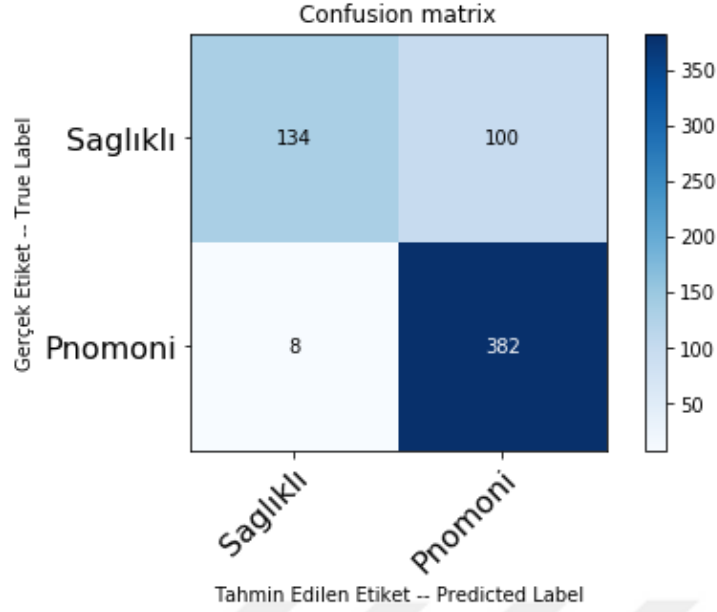
Çizelge 4. 1: CNN modelinin parametre deęerleri

Parametreler	Deęerler
Convolution Layer	10
Max Pooling Layer	4
Drop-Out Layer	2
Global Average Pooling Layer	--
Batch Normalization	4
Fully Connected Layer	3
Optimization Model	ADAM
Learning Rate	0.001
Activation F.	Relu-Softmax
Early Stopping	5
Batch Size	16
Epoch	20

Çizelge 4. 2: CNN modelinin başarı ölçüm sonuçları

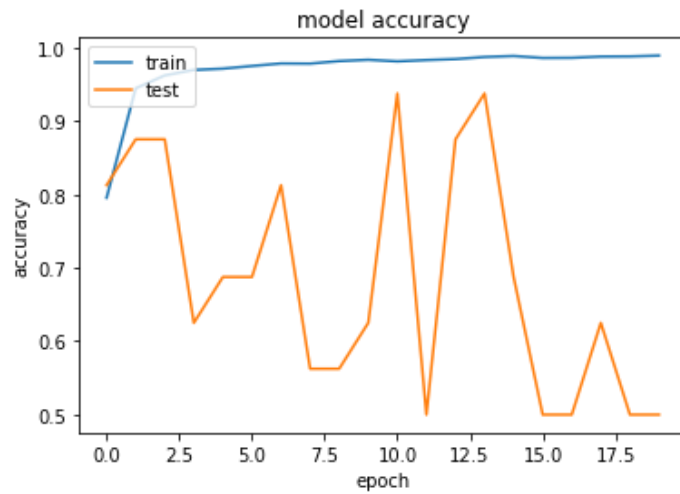
Ölçüm Parametreleri	Deęerler
Recall	% 97,9
Precision	% 79,3
Acc	% 83
Tpr	% 97,9
Fpr	% 42,7
Ppv	% 79,3
Tnr	% 57,3
F1 Skor	% 87,6

Oluşturulan CNN modelinin etkinliğini ölçmek, Çizelge 4.2’de verilen ölçüm deęerlerini daha detaylı incelemek ve doğru-yanlış tahmin adetlerini görebilmek için karışıklık matrisi olarak bilinen confusion matrix çizdirilmiştir. Şekil 4.7’de gerçekte olan deęerlerine karşılık modelin tahmin ettiği doğru, yanlış deęerler verilmiştir.

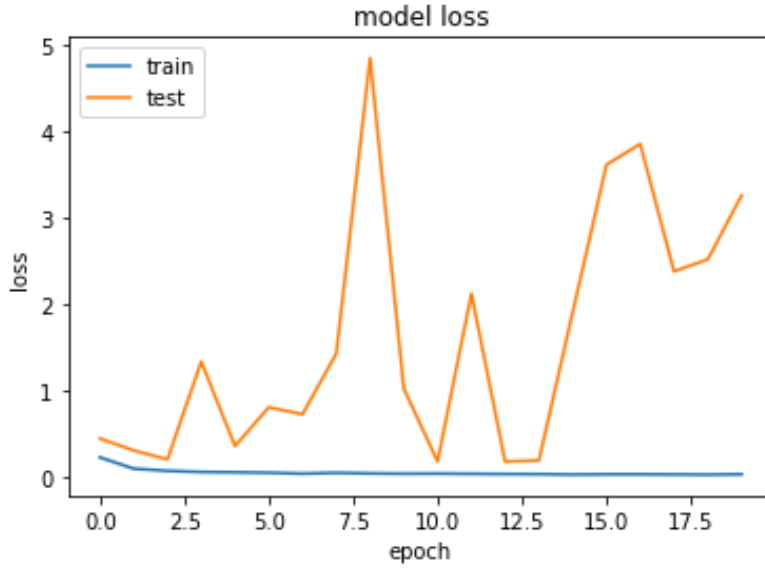


Şekil 4. 7: CNN model için çizdirilen confusion matrix

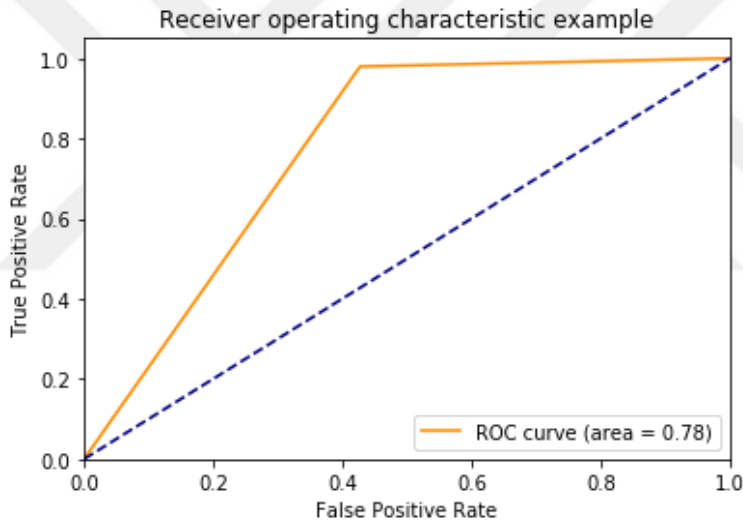
CNN modelin epoch değerine göre doğruluk-accuracy değerinin değişimi Şekil 4.8'de gösterilmiştir. Epoch değeri 20 verilmesine rağmen 10 ve 13 değerlerinin bu model için uygun olduğu görülmektedir. Şekil 4.9 incelendiğinde ise loss değerinin yine bu epoch değerlerinde dah düşük olduğu görülmektedir. Şekil 4.10'da ise ROC yani doğru pozitifler ile yanlış pozitiflerin oranının yer aldığı grafik görülmektedir. ROC eğrisi 'x' eksenine yaklaştıkça başarının seviyesi düşer. Sistemin başarısı, ROC eğrisinin altında kalan alan yani AUC ile ölçülmektedir. Bu alan değeri ne kadar büyük ise sistemin güvenilirlik değeri de o kadar yüksek olmaktadır, ilgili grafiğin Auc değeri 0.78 dir.



Şekil 4. 8: CNN modelin acc-epoch değişimi



Şekil 4. 9: CNN modelin loss-epoch değişimi



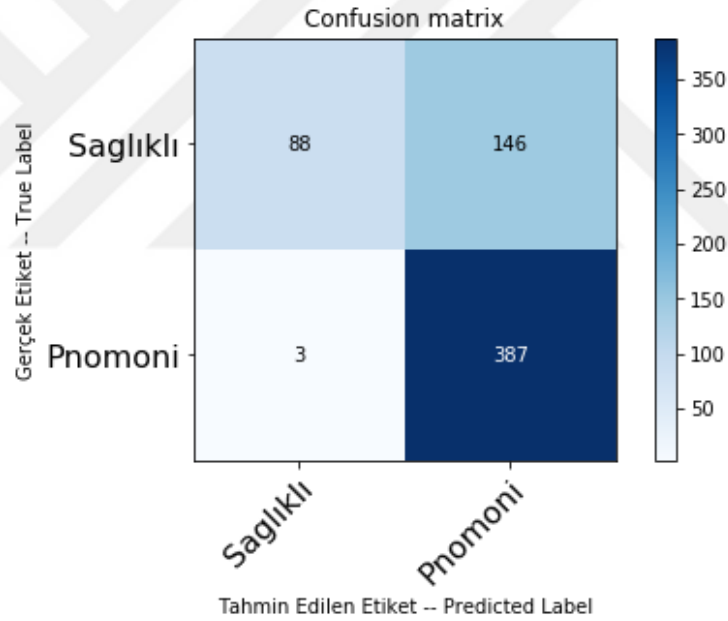
Şekil 4. 10: CNN modelinin ROC grafiği

Oluşturulan model de sentetik veri artırma işlemi uygulanarak yukarıdaki sonuçlar elde edilmiştir. Aynı model sentetik veri artırma yani data augmentation yapılmadan aynı parametreler ile tekrar çalıştırılmıştır. Buradaki amaç veri artırma işleminin modele katkısını belirlemektir. Çizelge 4.3 ile bu deneyin detaylı sonuçları verilmiştir. Sonuçlar incelendiğinde sağlıklı verileri doğru tahmin etme başarısının çok düştüğü görülmektedir, Şekil 4.11’de confusion matrix verilmiştir. Yani sağlıklı verileri de pnömonili olarak tahmin etmeye başlamıştır. Burada veri dağılımının sonuca etkisinin yanı sıra veri artırma işleminin modeli ne kadar iyileştirdiği görülebilmektedir. Aynı zamanda veri sayısının fazla olmasının modelin başarı

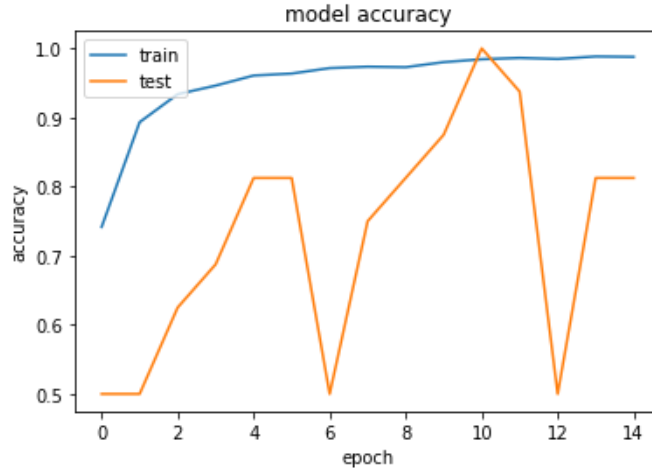
oranını artırdığı da görülmektedir. Aynı zamanda Şekil 4.12 ve Şekil 4.13’de verilen accuracy ve loss grafikleri de incelendiğinde modelin iyi öğrenemediği görülmektedir. Şekil 4.14’te ROC eğrisine bakıldığında ise auc değerinin 0.69 yani yaklaşık %13 düştüğü görülmektedir.

Çizelge 4. 3: CNN modelinin veri artırma uygulanmamış başarı ölçüm sonuçları

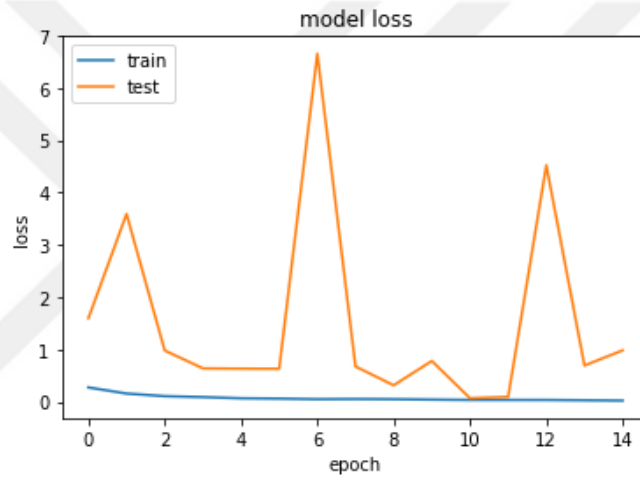
Ölçüm Parametreleri	Değerler
Recall	% 99
Precision	% 72
Acc	% 76
Tpr	% 99
Fpr	% 62,4
Ppv	% 72,6
Tnr	% 37,6
F1 Skor	% 83,9



Şekil 4. 11: CNN modelinin veri artırma işlemi uygulanmadan çizdirilen grafiği



Şekil 4. 12: CNN modelinin veri artırma işlemi uygulanmamış acc-epoch değişimi



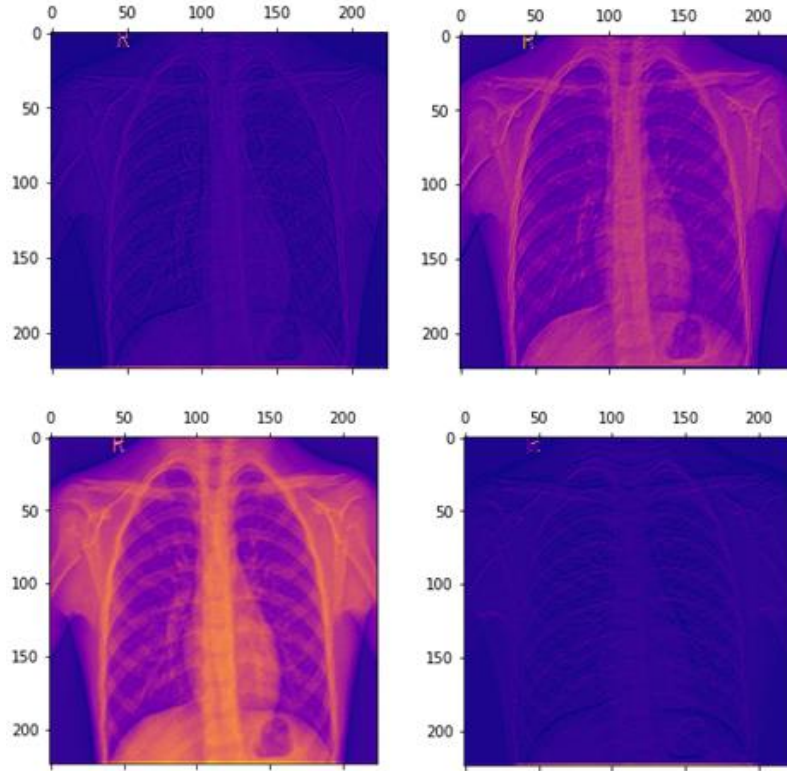
Şekil 4. 13: CNN modelinin veri artırma işlemi uygulanmamış loss-epoch değişimi



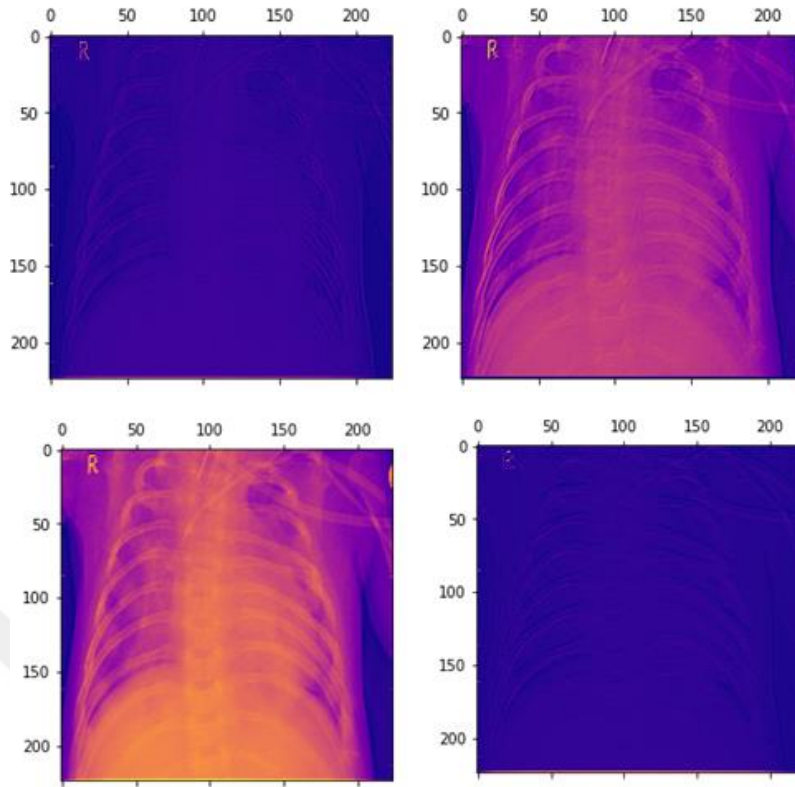
Şekil 4. 14: CNN modelinin veri artırma işlemi uygulanmamış ROC grafiği

Veri artırma işlemi uygulanan CNN modelinin ne öğrendiğini detaylı incelemek için sağlıklı ve pnömonili verilerin ayrı ayrı 1, 7 ve 9. katmanlarındaki filtreleri görselleştirilmiştir.

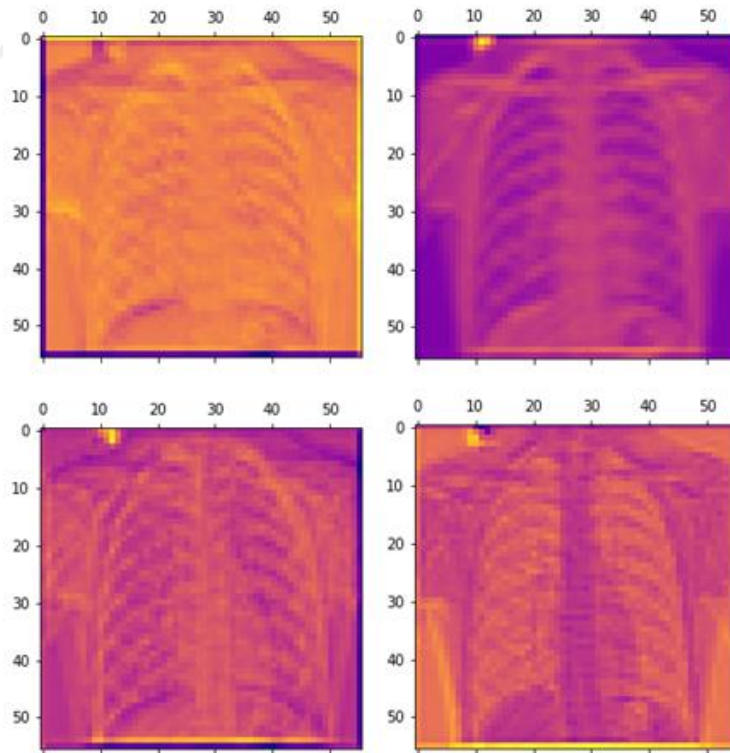
Oluşturulan CNN modelin birinci katmanın boyutu şu şekildedir; (1, 224, 224, 64). Bu model birinci katmanında 64 tane filtreye sahiptir. Bu 64 filtreden aşağıdaki Şekil 4.15'te gösterilen 4 farklı görüntü alınmıştır. Görüntüler 0, 1, 62 ve 63. filtrelere aittir. Aynı filtrelerin hasta verilerindeki çıktıları da Şekil 4.16'da gösterilmiştir. Birinci katmandaki bu filtrelere baktığımızda renklere ve akciğere daha çok odaklandığını görebiliyoruz. Oluşturulan CNN modelin yedinci katmanın boyutu şu şekildedir; (1, 56, 56, 256). Bu model yedinci katmanında 256 tane filtreye sahiptir. Bu 256 filtreden aşağıdaki Şekil 4.17'de gösterilen 4 farklı görüntü alınmıştır. Görüntüler 1, 5, 250 ve 255. filtrelere aittir. Aynı filtrelerin hasta verilerindeki çıktıları da Şekil 4.18'de gösterilmiştir. Oluşturulan CNN modelin dokuzuncu katmanın boyutu ise (1, 56, 56, 256) şeklindedir. Bu model dokuzuncu katmanında 256 tane filtreye sahiptir. Bu 256 filtreden aşağıdaki Şekil 4.19'da gösterilen 4 farklı görüntü alınmıştır. Görüntüler 0, 64, 128 ve 254. filtrelere aittir. Aynı filtrelerin hasta verilerindeki çıktıları da Şekil 4.20'de gösterilmiştir.



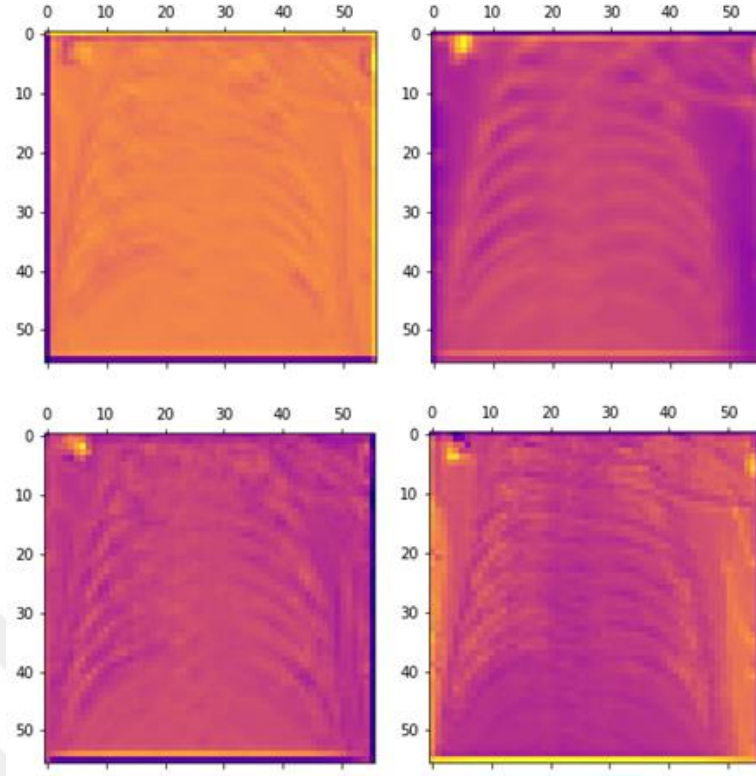
Şekil 4. 15: Sağlıklı verinin CNN model 1.katman farklı filtre görüntüleri



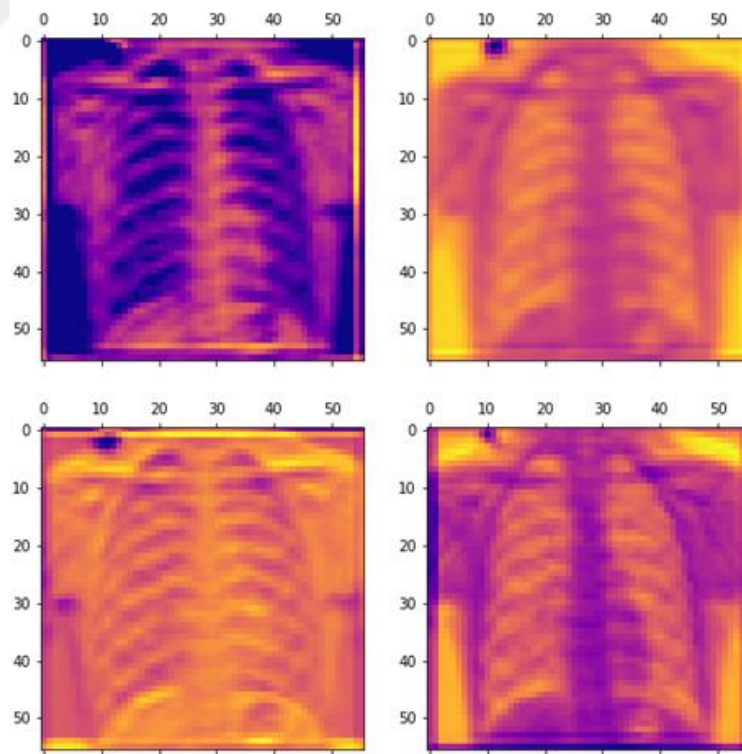
Şekil 4. 16: Pnömonili verinin CNN model 1.katman farklı filtre görüntüleri



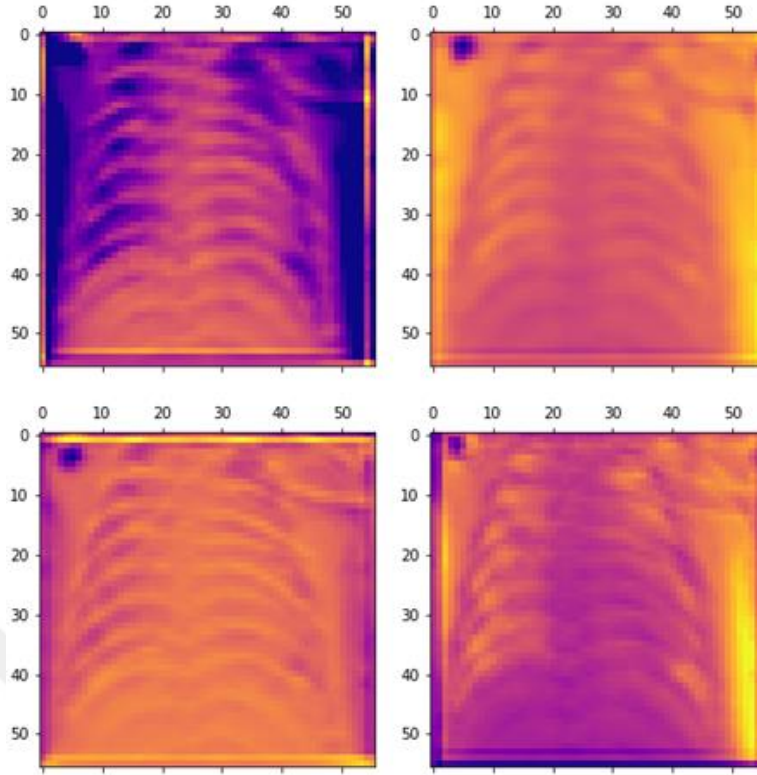
Şekil 4. 17: Sağlıklı verinin CNN model 7.katman farklı filtre görüntüleri



Şekil 4. 18: Pnömonili verinin CNN model 7.katman farklı filtre görüntüleri



Şekil 4. 19: Sağlıklı verinin CNN model 9.katman farklı filtre görüntüleri



Şekil 4. 20: Pnömonili verinin CNN model 9.katman farklı filtre görüntüleri

4.1.2 Inception Modeli

Inception modeli 2 farklı şekilde kullanılmıştır. İlk aşamada veri kümesi rastgele olarak, %70 eğitim, %20 test, %10 doğrulama olmak üzere üçe bölünmüştür. İkinci aşamada ise k-fold yöntemi kullanılmıştır. Her iki yöntemin çıktıları karşılaştırılmıştır.

Inception modeli; 94 konvolüsyon (conv2d) ve toplu normalleştirme (batch norm), 2 tamamen bağlı (fully connected) katmandan oluşmaktadır. Havuzlama olarak ortalama ve maksimum (10 average pooling, 4 max pooling, 1 global average) kullanılmıştır. Model tasarlandıktan sonra ağırlık başlatma işlemi uygulanmıştır. İlk konvolüsyonun ağırlığı, ImageNet ağırlıkları ile başlatılmıştır. Optimizasyon işlemi olarak Adam kullanılmıştır. Öğrenme hızı olarak 0.001 , erken durdurma değeri olarak 5 tanımlanmıştır. Grup sayısı (batch size) 16, eğitim tur sayısı (epoch) ise 20 olarak belirlenmiştir. Parametre değerleri ise Çizelge 4.4'te verilmiştir. İlgili parametre değerlerine karşılık alınan ölçüm değerleri ise Çizelge 4.5'te verilmiştir.

Sonuçlar incelendiğinde fpr ölçüm parametresi dışında diğer tüm parametreler CNN modelinde elde edilenlerden daha iyi değildir. İyileşme sadece 0,034 fark ile fpr'da görülmüştür. Bu iyileşme farkı da yeterince yüksek değildir. Bu modelin Şekil

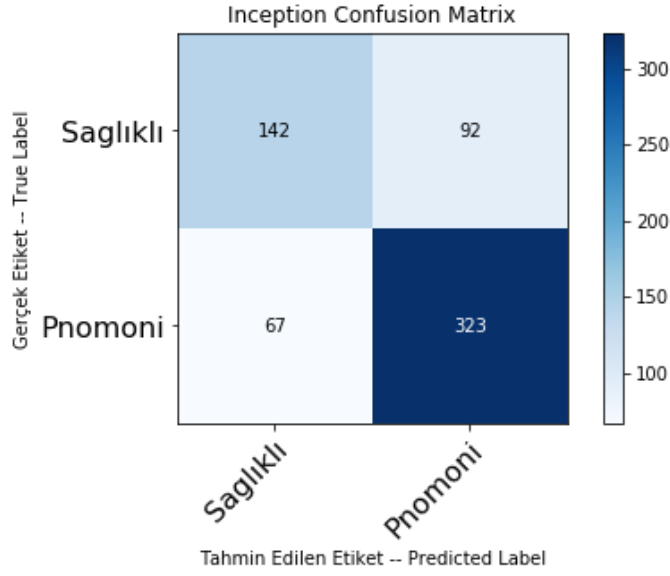
4.21’de verilen confusion matrix deęerleri incelendięinde bir önceki modele göre pnömonili verileri sağlıklı olarak tahmin etmeye başladığını ve bu hatalı tahminlerinde artış olduęu görölmektedir. Şekil 4.22 ve Şekil 4.23 ‘te verilen accuracy-loss ve epoch deęişimlerine bakıldığında yine bir önceki modele göre düşüşler görölmektedir. Şekil 4.24’te verilen roc grafięine baktığımızda ise auc deęerininde bir önceki modele göre düştüęü görölmektedir.

Çizelge 4. 4: Inception modelinin parametre deęerleri

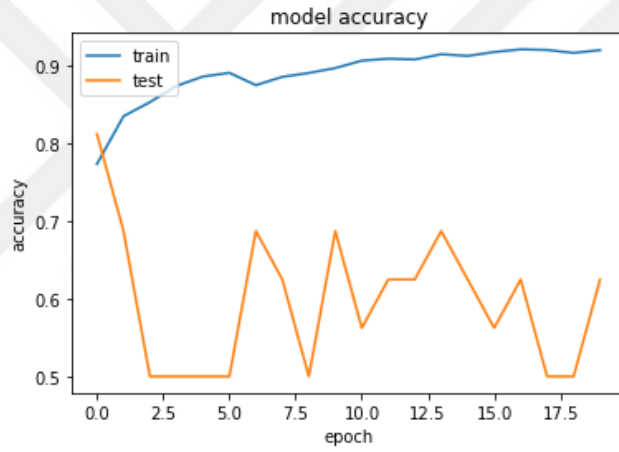
Parametreler	Deęerler
Convolution Layer	94
Max Pooling Layer	4
Average Pooling Layer	10
Drop-Out Layer	--
Global Average Pooling Layer	1
Batch Normalization	94
Fully Connected Layer	2
Optimization Model	ADAM
Learning Rate	0.001
Activation F.	Relu-Softmax
Early Stopping	5
Batch Size	16
Epoch	20

Çizelge 4. 5: Inception modelinin başarı ölçüm sonuçları

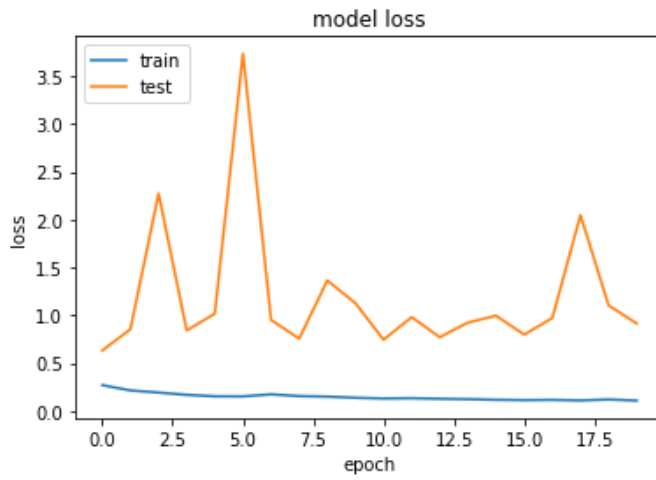
Ölçüm Parametreleri	Deęerler
Recall	% 82,8
Precision	% 77,8
Acc	% 74,5
Tpr	% 82,8
Fpr	% 39,3
Ppv	% 77,8
Tnr	% 60,7
F1 Skor	% 80,2



Şekil 4. 21: Inception model için çizdirilen confusion matrix

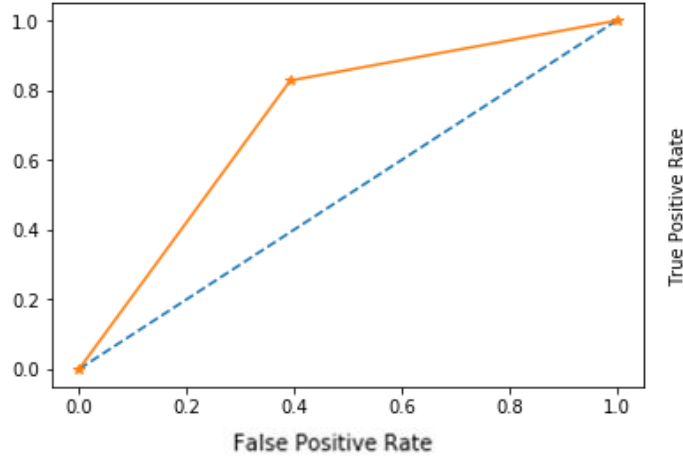


Şekil 4. 22: Inception modelin acc-epoch değişimi



Şekil 4. 23: Inception modelin loss-epoch değişimi

Şekil 4.24 de verilen grafiğin AUC değeri: 0.7175 'dir.



Şekil 4. 24: Inception modelin ROC grafiği

Aynı model için ikinci aşamada k-fold uygulanmıştır. K değeri ve epoch değişiminin modele yansması ölçülmüştür. Önce epoch değeri 10 alınıp k değeri sırası ile 3 ve 10 verilmiştir. Daha sonra epoch değeri 20 yapıp k değeri 3 alınmıştır. Bu üç deneyin ölçüm parametre ve çıktıları sırası ile Çizelgelerde detaylıca verilmiştir. Epoch değeri 10 iken k değerini artırmak bazı ölçüm parametrelerinde iyileştirmeyi sağlarken fpr ölçüm parametre değerinde olumsuz etkiye neden olmuştur.

K değerini sabit tutup epoch değerini 10'dan 20'ye artırmakta aynı şekilde bazı parametrelerde iyileşme sağlarken auc değerinde ciddi bir artışa neden olmadığı gibi fpr değerini de bir önceki deneylere göre iyileştirmemiştir. Üstelik modeli daha uzun süre eğittiğimizden zamandan da tasarruf edilmemiştir. Ayrıca model için erken durdurma değeri 5 verilmiştir. Bu erken durdurma sayesinde model 10 epoch'tan sonra ciddi şekilde performansı düşmeye başlayınca düşüş ilerlemeden modeli durdurmuştur, Şekil 4.27, Şekil 4.31 ve Şekil 4.35'te verilmiştir.

Bu model için epoch değerinin 20 yerine 10 verilmesi ile daha iyi sonuçlar sağlanmıştır, Şekil 4.26, Şekil 4.30 ve Şekil 4.34 'te bu değişimler görülmektedir. Deneylerin parametre değerleri Çizelge 4.6, Çizelge 4.8 ve Çizelge 4.10'da verilmiştir. Ölçüm sonuçları ise Çizelge 4.7, Çizelge 4.9 ve Çizelge 4.11'de verilmiştir. K=3, E=10 parametre değerleri verilen deneyde auc ve fpr değeri diğer deneylere göre daha iyi sonuç vermiştir, Şekil 4.28, Şekil 4.32 ve Şekil 4.36'da roc eğrileri verilmiştir. K=3, E=10 olan deneyde pnömonili olup sağlıklı tahmin edenlerin sayısı, E=20 parametresi verilen deneyden fazla bulunmuştur. E=20 verilen

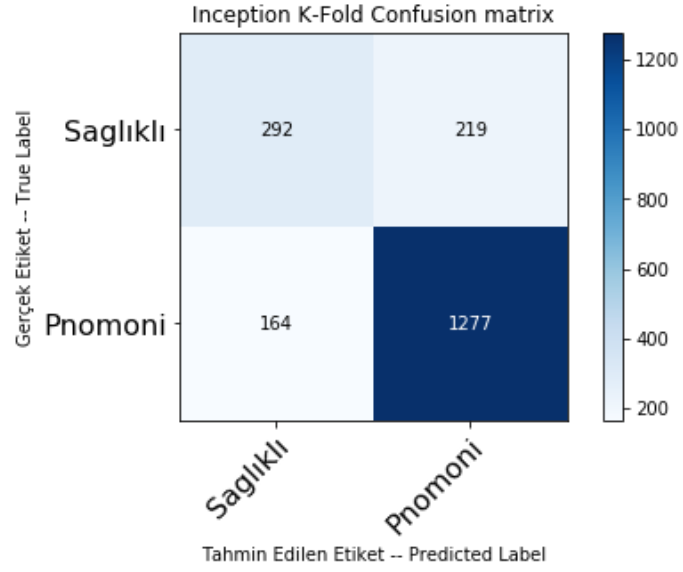
deneyde ise sağlıklı olup pnömonili tahmin edilenlerin sayısı diğer deneylere göre fazla bulunmuştur. Ölçüm sonuçlarının adet bazında değerleri Şekil 4.25, Şekil 4.29 ve Şekil 4.33'te verilmiştir.

Çizelge 4. 6: Inception modelinin K=3 ve E=10 için parametre değerleri

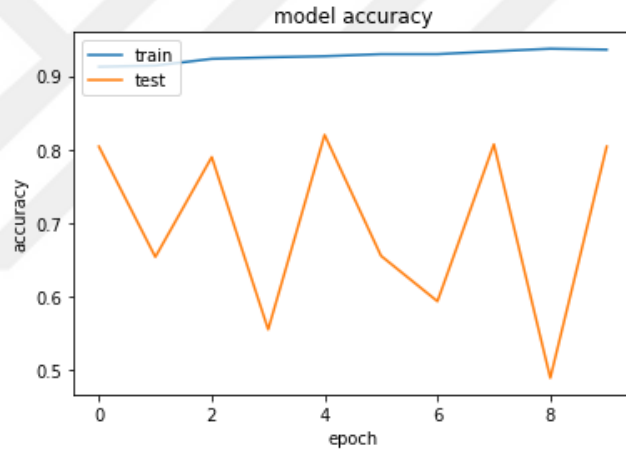
Parametreler	Değerler
Convolution Layer	94
Max Pooling Layer	4
Average Pooling Layer	10
Drop-Out Layer	--
Global Average Pooling Layer	1
Batch Normalization	94
Fully Connected Layer	2
Optimization Model	ADAM
Learning Rate	0.001
Activation F.	Relu-Softmax
Early Stopping	5
Batch Size	16
Epoch	10
K-Fold	3

Çizelge 4. 7: Inception modelinin K=3 ve E=10 için başarı ölçüm sonuçları

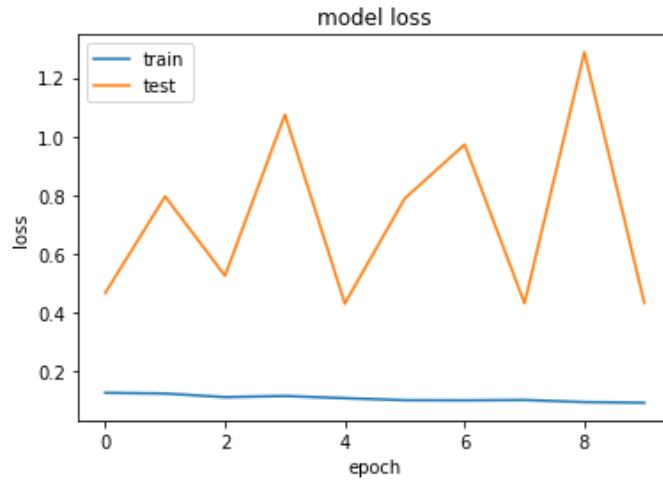
Ölçüm Parametreleri	Değerler
Recall	% 88,6
Precision	% 85,4
Acc	% 80,4
Tpr	% 88,6
Fpr	% 42,9
Ppv	% 85,4
Tnr	% 57,1
F1 Skor	% 87



Şekil 4. 25: Inception modelinin K=3 ve E=10 için confusion matrix

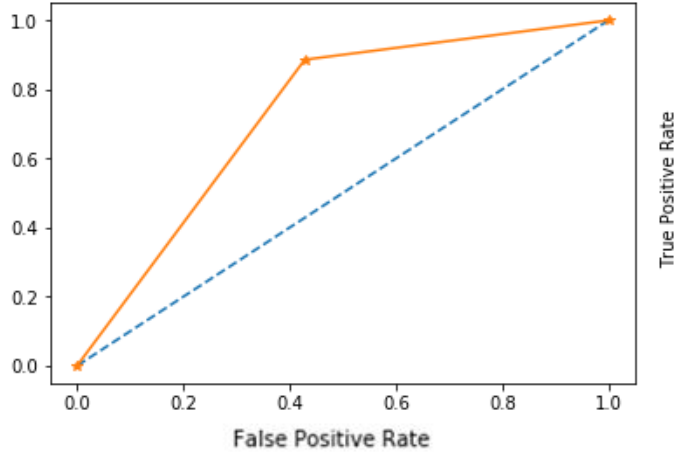


Şekil 4. 26: Inception modelinin K=3 ve E=10 için acc-epoch değişimi



Şekil 4. 27: Inception modelinin K=3 ve E=10 için loss-epoch değişimi

Şekil 4.28’de verilen grafiğin AUC değeri: 0.7288’ dir.



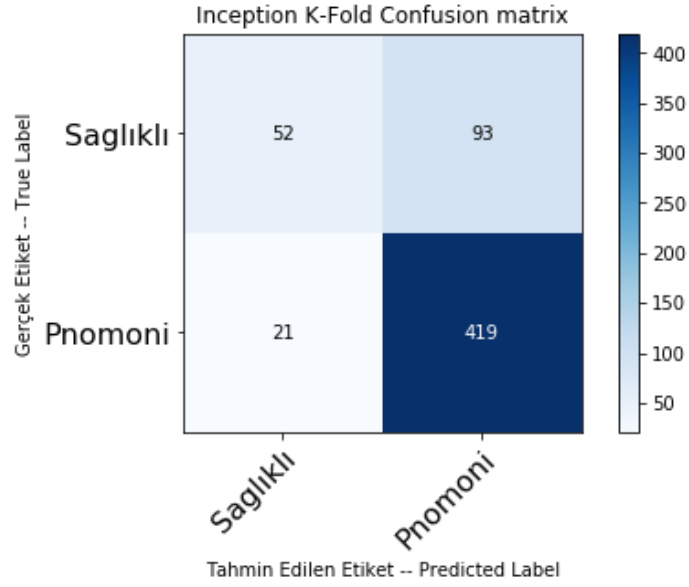
Şekil 4. 28: Inception modelinin K=3 ve E=10 için ROC grafiği

Çizelge 4. 8: Inception modelinin K=10 ve E=10 için parametre değerleri

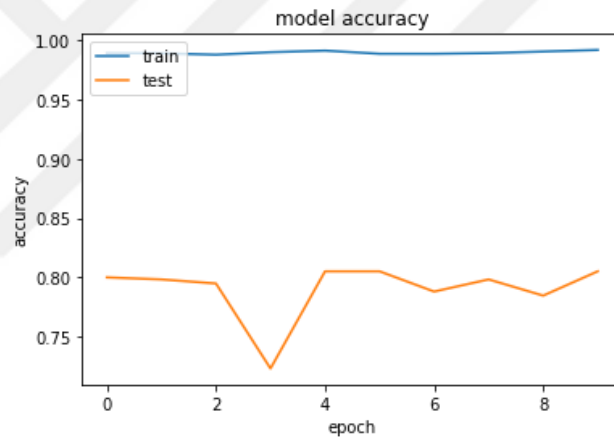
Parametreler	Değerler
Convolution Layer	94
Max Pooling Layer	4
Average Pooling Layer	10
Drop-Out Layer	--
Global Average Pooling Layer	1
Batch Normalization	94
Fully Connected Layer	2
Optimization Model	ADAM
Learning Rate	0.001
Activation F.	Relu-Softmax
Early Stopping	5
Batch Size	16
Epoch	10
K-Fold	10

Çizelge 4. 9: Inception modelinin K=10 ve E=10 için başarı ölçüm sonuçları

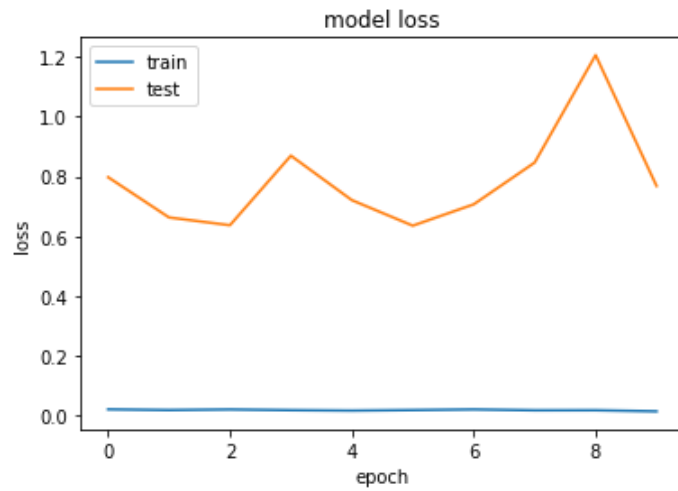
Ölçüm Parametreleri	Değerler
Recall	% 95,2
Precision	% 81,8
Acc	% 80,5
Tpr	% 95,2
Fpr	% 64,1
Ppv	% 81,8
Tnr	% 35,9
F1 Skor	% 88



Şekil 4. 29: Inception modelinin K=10 ve E=10 için confusion matrix

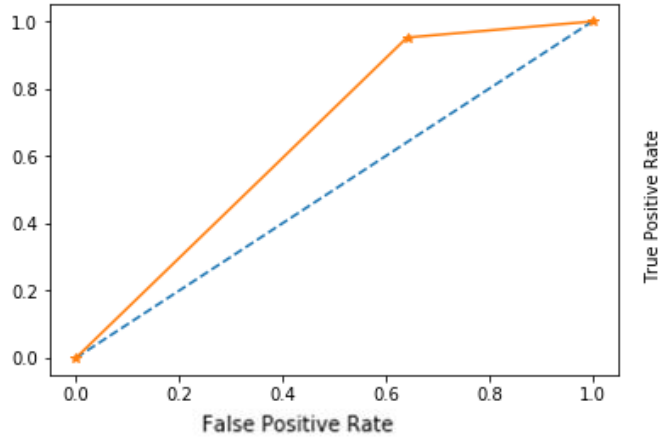


Şekil 4. 30: Inception modelinin K=10 ve E=10 için acc-epoch değişimi



Şekil 4. 31: Inception modelinin K=10 ve E=10 için loss-epoch değişimi

Şekil 4.32’de verilen grafiğin AUC Değeri: 0.6554’dir.



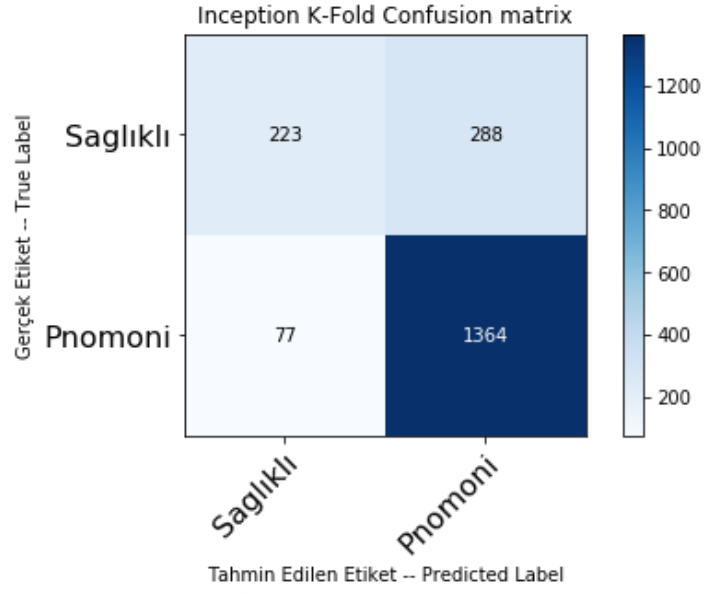
Şekil 4. 32: Inception modelinin K=10 ve E=10 için ROC grafiği

Çizelge 4. 10: Inception modelinin K=3 ve E=20 için parametre değerleri

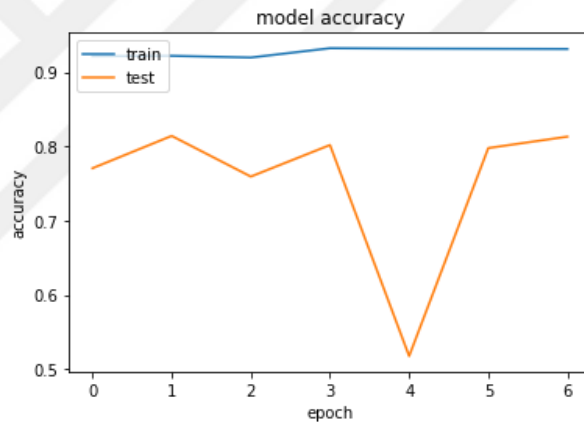
Parametreler	Değerler
Convolution Layer	94
Max Pooling Layer	4
Average Pooling Layer	10
Drop-Out Layer	--
Global Average Pooling Layer	1
Batch Normalization	94
Fully Connected Layer	2
Optimization Model	ADAM
Learning Rate	0.001
Activation F.	Relu-Softmax
Early Stopping	5
Batch Size	16
Epoch	20
K-Fold	3

Çizelge 4. 11: Inception modelinin K=3 ve E=20 için başarı ölçüm sonuçları

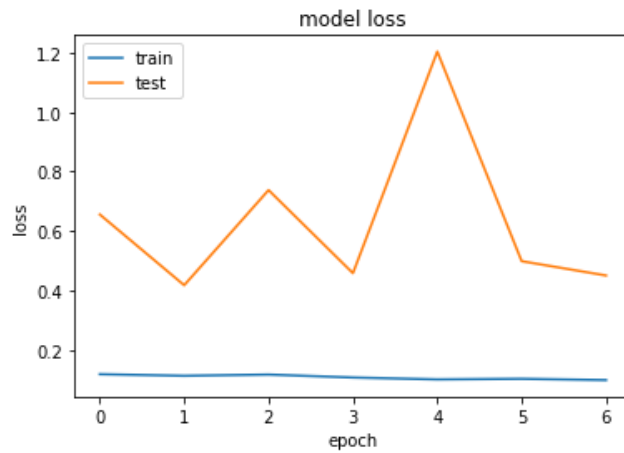
Ölçüm Parametreleri	Değerler
Recall	% 94,7
Precision	% 82,6
Acc	% 81,3
Tpr	% 94,7
Fpr	% 56,4
Ppv	% 82,6
Tnr	% 43,6
F1 Skor	% 88,2



Şekil 4. 33: Inception modelinin K=3 ve E=20 için confusion matrix

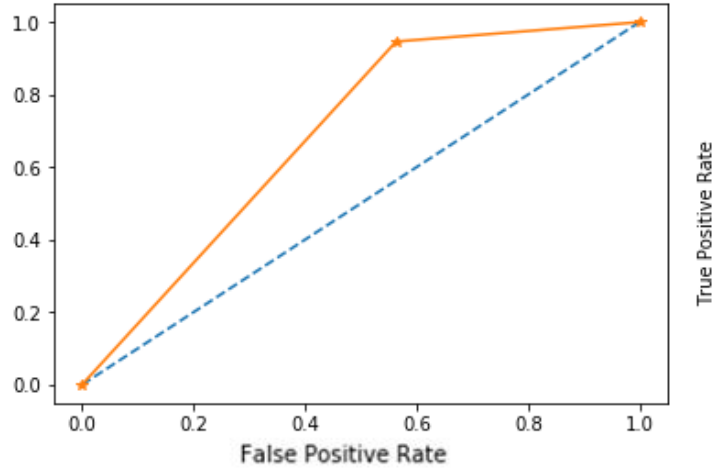


Şekil 4. 34: Inception modelinin K=3 ve E=20 için acc-epoch değişimi

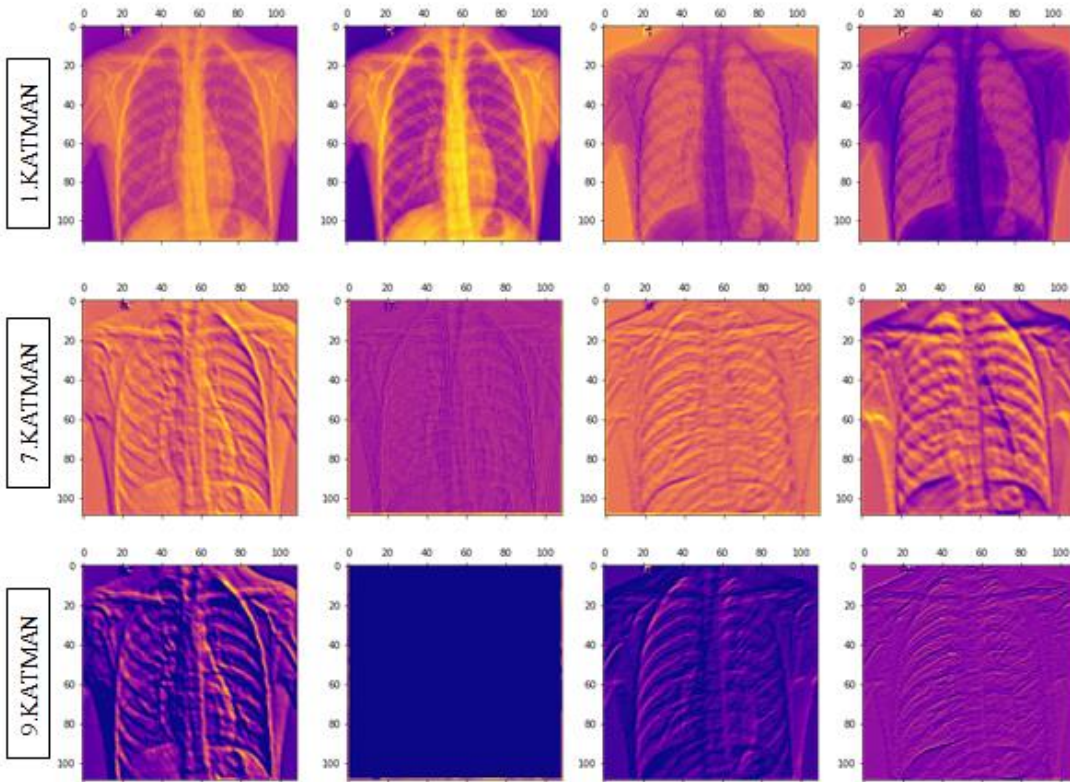


Şekil 4. 35: Inception modelinin K=3 ve E=20 için loss-epoch değişimi

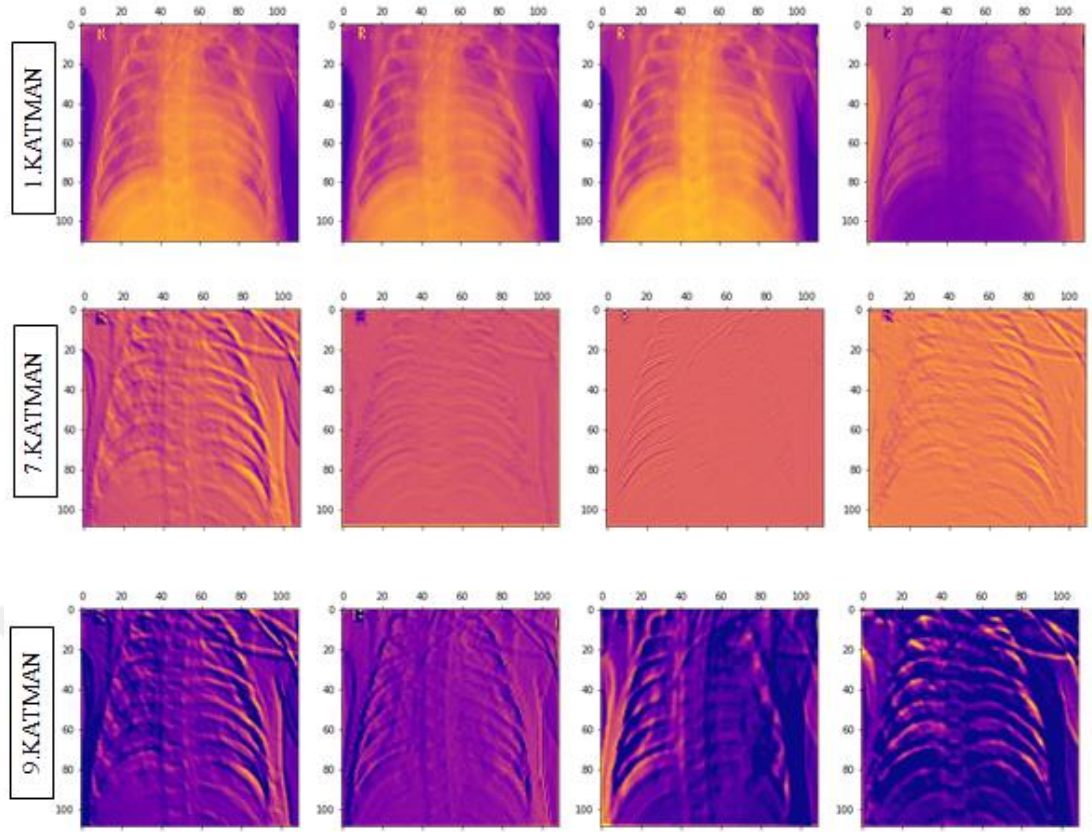
Şekil 4.36 Roc grafiğinin AUC Değeri: 0.6915'dir.



Şekil 4. 36: Inception modelinin K=3 ve E=20 için ROC grafiği



Şekil 4. 37: Inception modeli için sağlıklı ara filtre çıktıları



Şekil 4. 38: Inception modeli için pnömonili ara filtre çıktıları

Inception modelinde birinci katman (1, 111, 111, 32), yedinci katman (1, 109, 109, 64) ve dokuzuncu katman (1, 109, 109, 64) şeklindedir. İlk katmanın 32, yedinci ve dokuzuncu katmanın 64 filtresi bulunmaktadır. İlk katmandan (0,1,22,31), yedinci katmandan (0,10,25,63), dokuzuncu katmandan ise (0,35,55,61) nolu filtrelerin görüntüleri incelenmiştir. Her bir katmanda her bir filtrenin farklı noktalara odaklandığı görülmektedir. Aynı pnömonili veya sağlıklı görüntü ile farklı katmanlarda farklı filtrelerden bambaşka çıktılar elde edilmektedir. İlk katmanlarda filtre çıktılarından bir akciğer röntgen görüntüsü incelendiği anlaşılabilirken, daha derin katmanlardaki filtrelerin çıktıları daha da karmaşık görülmektedir.

4.1.3 VGG16 Modeli

Vgg16 modeli 2 farklı şekilde kullanılmıştır. İlk aşamada veri kümesi rastgele olarak, %70 eğitim, %20 test, %10 doğrulama olmak üzere üçe bölünmüştür. İkinci aşamada ise k-fold yöntemi kullanılmıştır. Her iki yöntemin çıktıları karşılaştırılmıştır. Vgg16 modeli; 13 konvolüsyon (conv2d) ve 2 tamamen bağlı (fully connected) katmandan oluşmaktadır. Havuzlama olarak ortalama ve maksimum (5 max pooling, 1 global

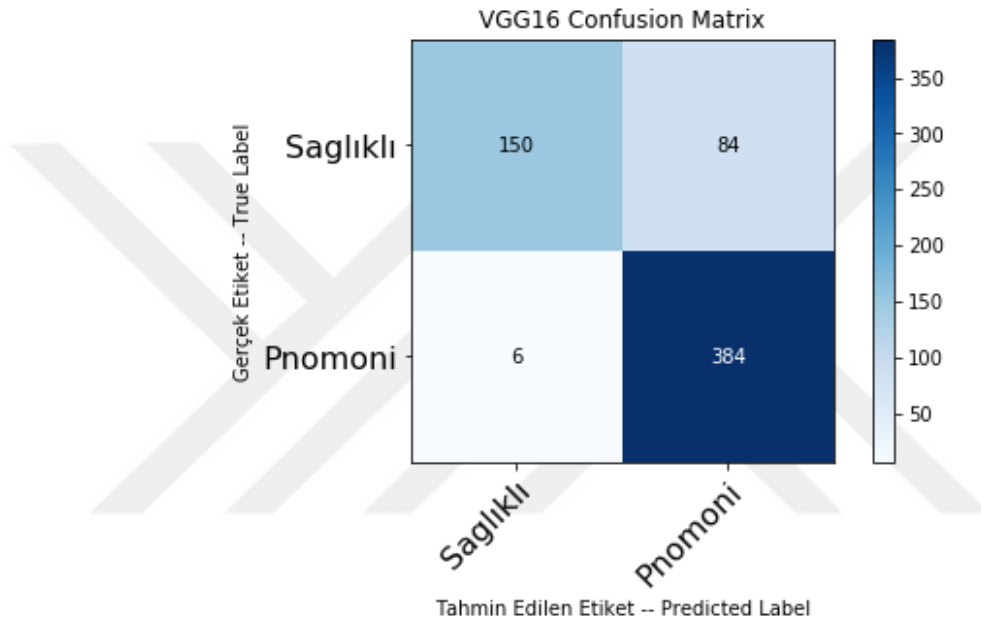
average) kullanılmıştır. Model tasarlandıktan sonra ağırlık başlatma işlemi uygulanmıştır. İlk konvolüsyonun ağırlığı, ImageNet ağırlıkları ile başlatılmıştır. Optimizasyon işlemi olarak Adam kullanılmıştır. Öğrenme hızı olarak 0.001 , erken durdurma değeri olarak 5 tanımlanmıştır. Grup sayısı (batch size) 16, eğitim tur sayısı (epoch) ise 20 olarak belirlenmiştir. Modelin diagramı ekler kısmına konulmuştur. Parametre değerleri ise Çizelge 4.12’de verilmiştir. İlgili parametre değerlerine karşılık alınan ölçüm değerleri ise Çizelge 4.13’te verilmiştir. Gerçek değer ile tahmin edilen değerlerin adet bazında karşılaştırılması için Şekil 4.39’da confusion matrix verilmiştir. İncelendiğinde gerçekte pnömonili olan verilerden 6 tanesini sağlıklı olarak yanlış tahmin ettiği görülmektedir. Diğer modellerde bu değer çok daha fazla iken bu modelde hatalı tahminlerin sayısı ciddi oranda düşmüştür. Bu model ile yapılan ilk deney ile elde edilen sonuçlar, diğer tüm deneylerden daha başarılı olmuştur. Acc değeri 0.85 , roc – auc değeri ise yaklaşık 0.82 ve fpr değeri ise 0.35’e ulaşmıştır, Şekil 4.42’de auc değeri verilmiştir.. Acc ve loss değerlerinin epoch ile değişimi ise Şekil 4.40 ve Şekil 4.41’de verilmiştir.

Çizelge 4. 12: VGG16 modelinin E=20 için parametre değerleri

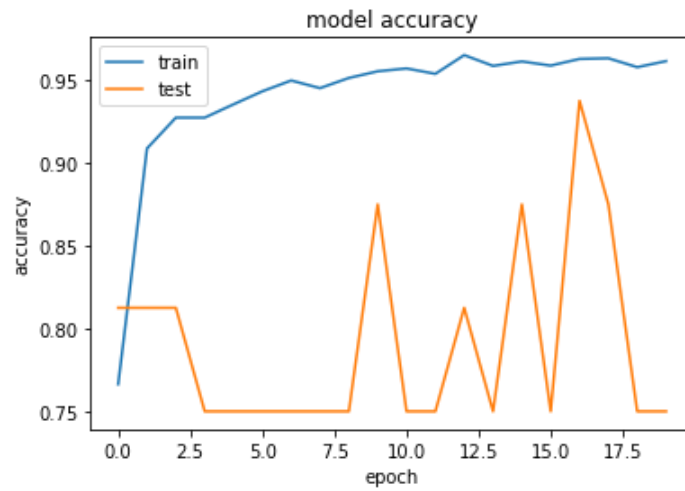
Parametreler	Değerler
Convolution Layer	13
Max Pooling Layer	5
Drop-Out Layer	--
Global Average Pooling Layer	1
Batch Normalization	--
Fully Connected Layer	2
Optimization Model	ADAM
Learning Rate	0.001
Activation F.	Relu-Softmax
Early Stopping	5
Batch Size	16
Epoch	20
K-Fold	--

Çizelge 4. 13: VGG16 modelinin E=20 için başarı ölçüm sonuçları

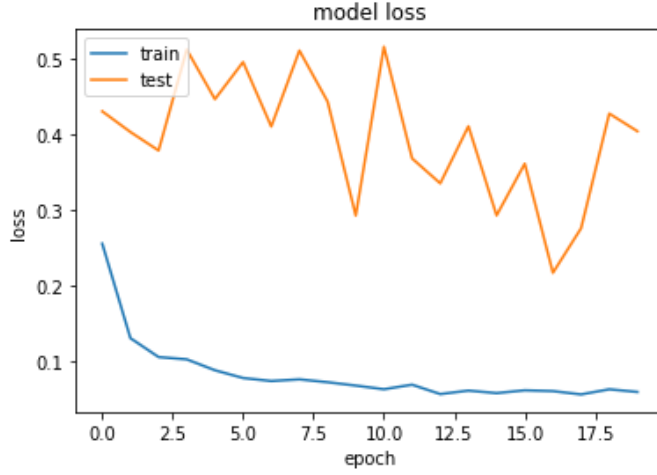
Ölçüm Parametreleri	Değerler
Recall	% 98,5
Precision	% 82,1
Acc	% 85,6
Tpr	% 98,5
Fpr	% 35,9
Ppv	% 82,1
Tnr	% 64,1
F1 Skor	% 89,5



Şekil 4. 39: VGG16 modeli için çizdirilen confusion matrix

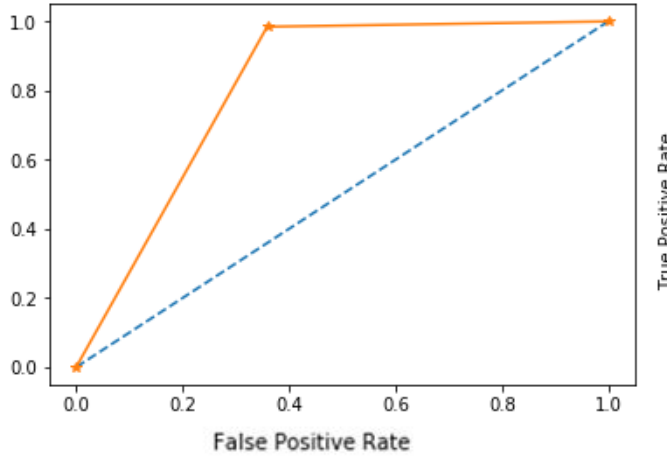


Şekil 4. 40: VGG16 modeli acc-epoch değişimi



Şekil 4. 41: VGG16 modeli loss-epoch değişimi

Şekil 4.42 Roc grafiğinin AUC Değeri: 0.8128 'dir.



Şekil 4. 42: VGG16 modeli ROC grafiği

Vgg16 modeli üzerinde k- fold yöntemi de çalışılmıştır. K-fold yönteminin ilk deneyinde K=3, E=10 , ikinci deneyinde ise K=10, E=10 ve son deneyinde ise K=3, E=20 verilmiştir. K ve E (epoch) değerinin model üzerindeki etkisi ölçümlenmiştir. İlk deneyde acc=0.96, fpr=0.055 ve roc-auc değeri 0.95'e ulaşmıştır, Şekil 4.46'da verilmiştir. Tüm parametrelerde ciddi oranda iyileşme görülmüştür. İkinci deneyde ise acc=0.97, fpr=0.034 ve roc-auc değeri 0.97'ye ulaşmıştır, Şekil 4.50'de verilmiştir. Bir önceki deneye göre tüm parametrelerde iyileşme görülmüştür. Şu ana kadar yapılan deneyler arasında en iyi sonucu veren model ve parametre değerleri bu olmuştur. Son deneyde ise acc=0.94, fpr=0.025 ve roc-auc değeri 0.95'e ulaşmıştır, Şekil 4.54'te verilmiştir. Acc ve roc-auc değerinde düşüş ama fpr değerinde iyileşme

görülmüştür. Fakat epoch 20 verildiğinden zamandan performans kaybına neden olmuştur. İlgili deneylerin detaylı parametre değerleri Çizelge 4.14, Çizelge 4.16 ve Çizelge 4.18’de verilmiştir. Ölçüm sonuçları ise Çizelge 4.15, Çizelge 4.17 ve Çizelge 4.19’da verilmiştir.

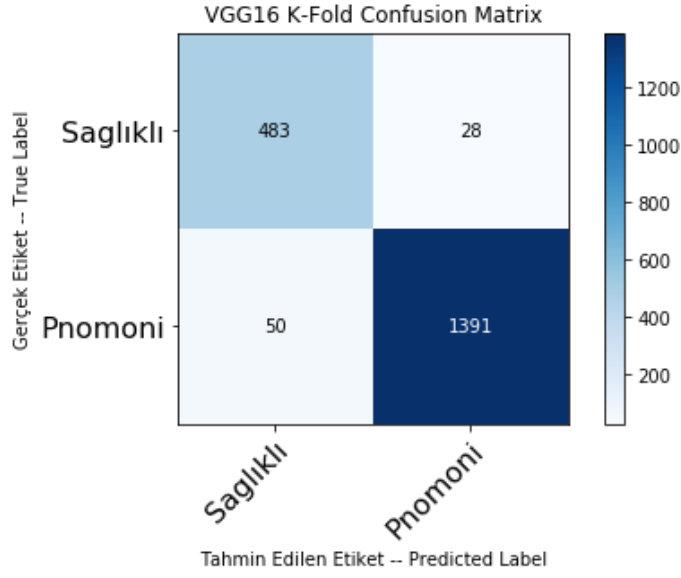
Vgg16 modelini, veri kümesi rastgele bölünme yöntemi ile çalıştırmaktansa, k- fold yöntemi ile çalıştırmak çok daha başarılı sonuçlar sağlamıştır. K değerini düşük verip epoch değerini yüksek vermek ya da k değerini yüksek verip epoch değerini düşük vermek yerine k değerini ortalama bir değer olarak 10 vermek daha başarılı sonuçlar sağlamıştır.

Çizelge 4. 14: VGG16 modelinin K=3 ve E=10 için parametre değerleri

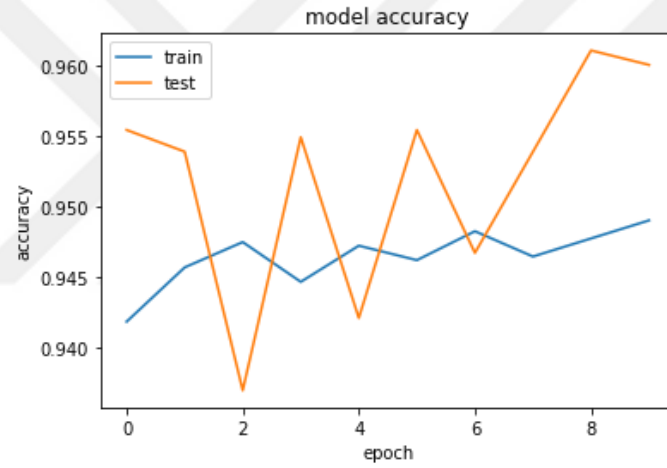
Parametreler	Değerler
Convolution Layer	13
Max Pooling Layer	5
Drop-Out Layer	--
Global Average Pooling Layer	1
Batch Normalization	--
Fully Connected Layer	2
Optimization Model	ADAM
Learning Rate	0.001
Activation F.	Relu-Softmax
Early Stopping	5
Batch Size	16
Epoch	10
K-Fold	3

Çizelge 4. 15: VGG16 modelinin K=3 ve E=10 için başarı ölçüm sonuçları

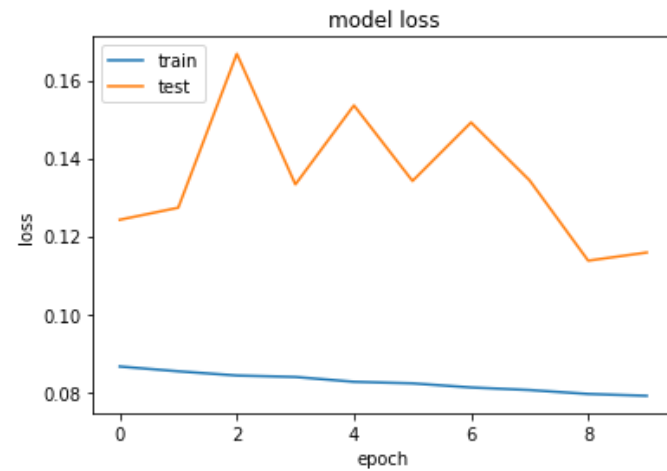
Ölçüm Parametreleri	Değerler
Recall	% 96,5
Precision	% 98
Acc	% 96
Tpr	% 96,5
Fpr	% 55
Ppv	% 98
Tnr	% 94,5
F1 Skor	% 97,3



Şekil 4. 43: VGG16 modeli K=3 ve E=10 için çizilen confusion matrix

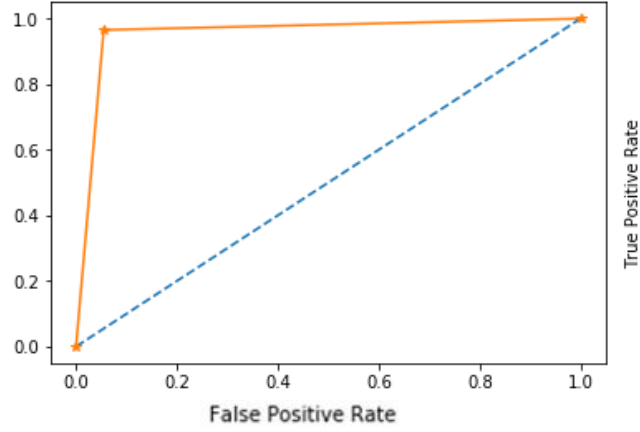


Şekil 4. 44: VGG16 modeli K=3 ve E=10 için acc-epoch değişimi



Şekil 4. 45: VGG16 modeli K=3 ve E=10 için loss-epoch değişimi

Şekil 4.46’da verilen ROC grafiğinin AUC Değeri: 0.9553’dir.



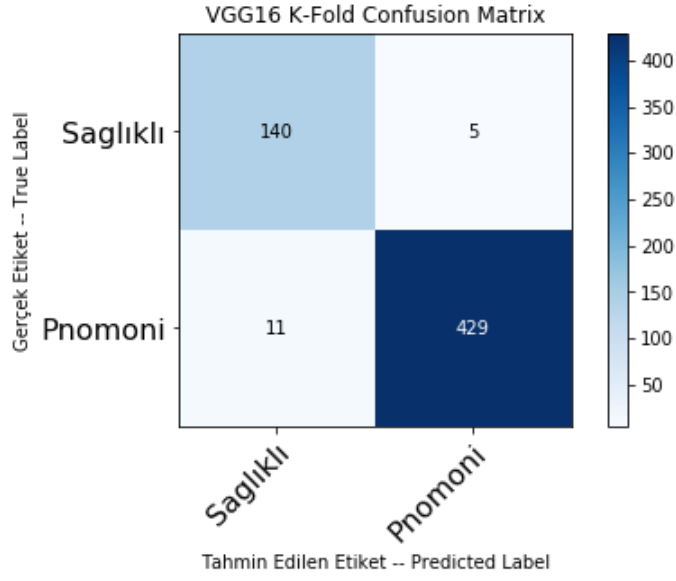
Şekil 4. 46: VGG16 modelinin K=3 ve E=10 için ROC grafiği

Çizelge 4. 16: VGG16 modelinin K=10 ve E=10 için parametre değerleri

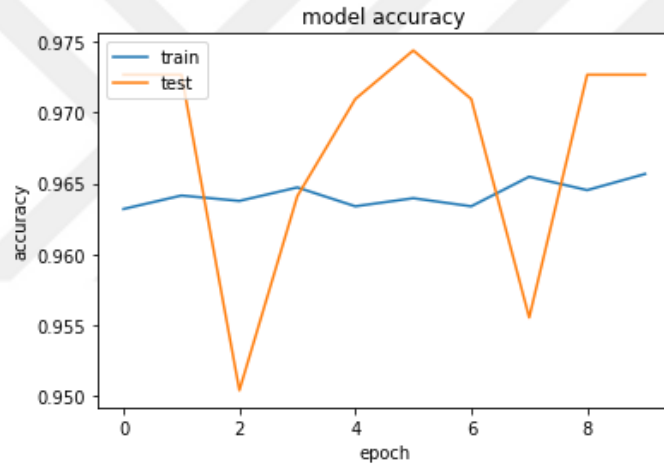
Parametreler	Değerler
Convolution Layer	13
Max Pooling Layer	5
Drop-Out Layer	--
Global Average Pooling Layer	1
Batch Normalization	--
Fully Connected Layer	2
Optimization Model	ADAM
Learning Rate	0.001
Activation F.	Relu-Softmax
Early Stopping	5
Batch Size	16
Epoch	10
K-Fold	10

Çizelge 4. 17: VGG16 modelinin K=10 ve E=10 için başarı ölçüm sonuçları

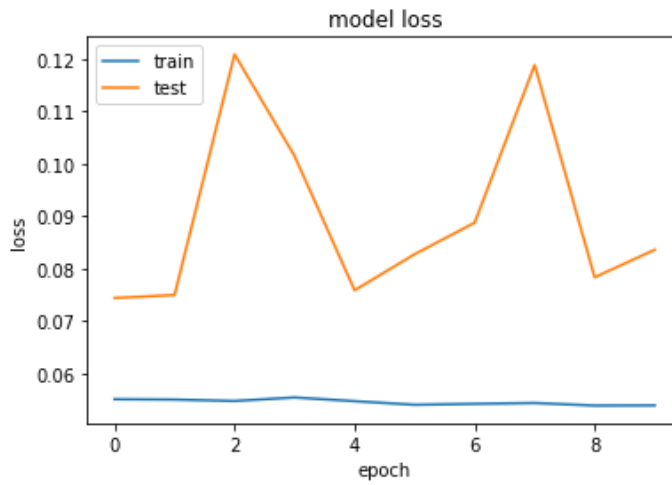
Ölçüm Parametreleri	Değerler
Recall	% 97,5
Precision	% 98,8
Acc	% 97,3
Tpr	% 97,5
Fpr	% 3,4
Ppv	% 98,8
Tnr	% 96,6
F1 Skor	% 98,2



Şekil 4. 47: VGG16 modeli K=10 ve E=10 için çizilen confusion matrix

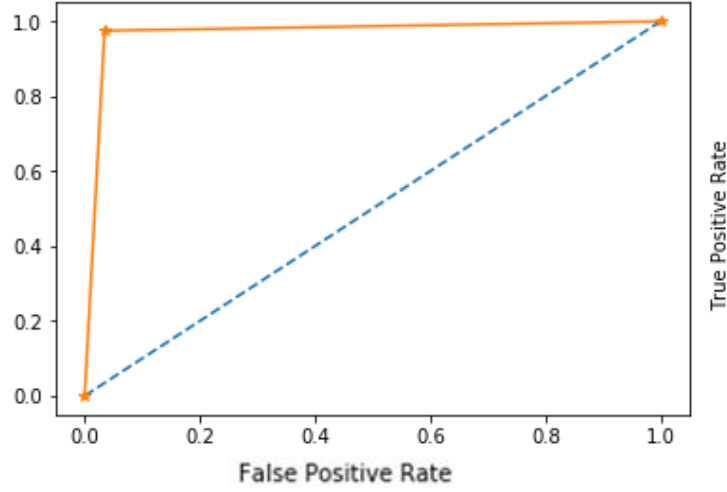


Şekil 4. 48: VGG16 modeli K=10 ve E=10 için acc-epoch değişimi



Şekil 4. 49: VGG16 modeli K=10 ve E=10 için loss-epoch değişimi

Şekil 4.50’de verilen ROC grafiğinin AUC Değeri: 0.9703’dir.



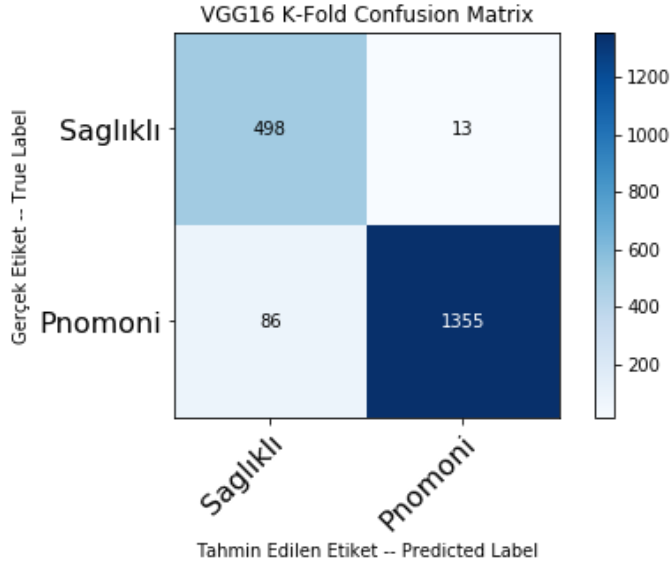
Şekil 4. 50: VGG16 modeli K=10 ve E=10 için ROC grafiği

Çizelge 4. 18: VGG16 modelinin K=3 ve E=20 için parametre değerleri

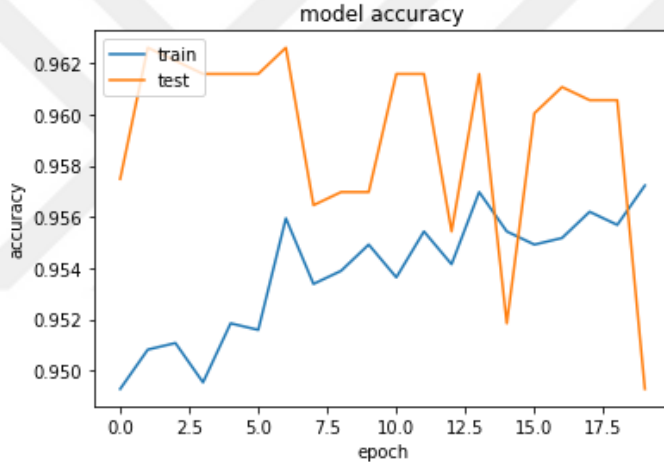
Parametreler	Değerler
Convolution Layer	13
Max Pooling Layer	5
Drop-Out Layer	--
Global Average Pooling Layer	1
Batch Normalization	--
Fully Connected Layer	2
Optimization Model	ADAM
Learning Rate	0.001
Activation F.	Relu-Softmax
Early Stopping	5
Batch Size	16
Epoch	20
K-Fold	3

Çizelge 4. 19: VGG16 modelinin K=3 ve E=20 için başarı ölçüm sonuçları

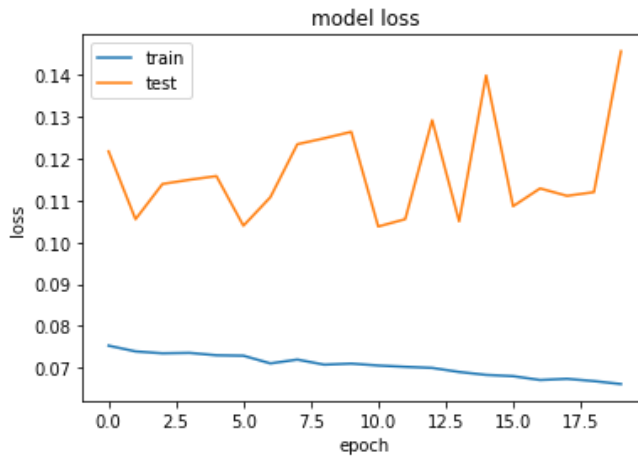
Ölçüm Parametreleri	Değerler
Recall	% 94
Precision	% 99
Acc	% 94,9
Tpr	% 94
Fpr	% 2,5
Ppv	% 99
Tnr	% 97,5
F1 Skor	% 96,5



Şekil 4. 51: VGG16 modeli K=3 ve E=20 için çizilen confusion matrix

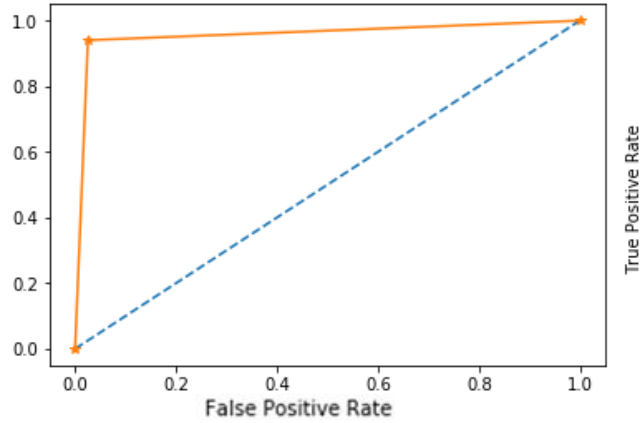


Şekil 4. 52: VGG16 modeli K=3 ve E=20 için acc-epoch değişimi



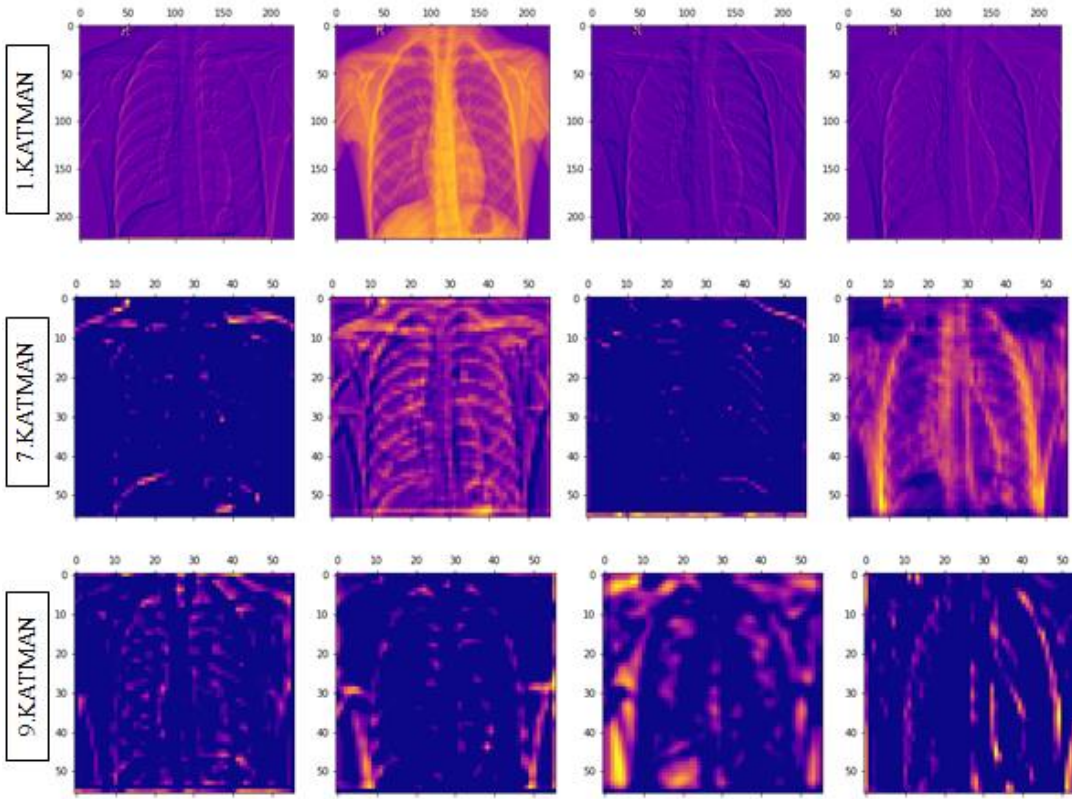
Şekil 4. 53: VGG16 modeli K=3 ve E=20 için loss-epoch değişimi

Şekil 4.54'te verilen ROC grafiğinin AUC Değeri: 0.9574'dir.

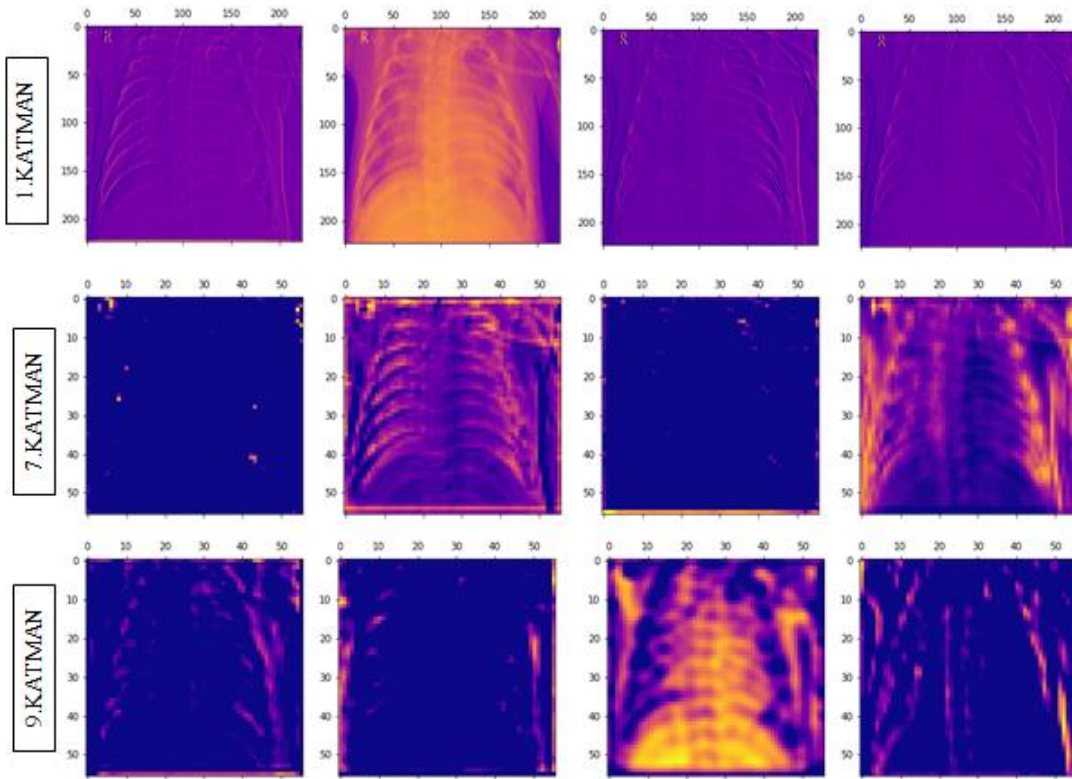


Şekil 4. 54: VGG16 modeli K=3 ve E=20 için ROC grafiği

Vgg16 modelinde en iyi sonuçları veren ikinci deneyin ara filtre görüntülerini incelediğimizde modelin ilerleyen katmanlarda diğer modellere göre daha kompleks çıktılar verdiği görülmektedir, Şekil 4.55 ve Şekil 4.56'da çıktı sonuçları verilmiştir. Bu modelde birinci katman (1, 224, 224, 64) şeklindedir, yani 64 filtresi bulunmaktadır. İlk katmanda (0,1,62,63) nolu filtreler görselleştirilmiştir. İlk katman görüntüleri tüm modellerde birbirine benzemektedir. Katmanlar derinleştikçe filtrelerin görüntülerde aradığı spesifik desenler görülmektedir. Yedinci katman (1, 56, 56, 256) şeklindedir, 256 filtresi bulunmaktadır. Bu katmanda (0,10,188,220) nolu filtreler görselleştirilmiştir. Hem sağlıklı hem de pnömonili veriler için bir önceki katmana göre çıktılar daha kompleks yapıdadır. Yedinci katmanda ilk filtrenin aradığı şey ile son filtrenin aradığı şey birbirinden tamamen farklıdır. İlk filtre imajın üst kısmını aynı tutarken orta kısmında yoğunlaştığı, son filtrenin ise daha çok renk odaklı olduğu görülmektedir. Dokuzuncu katmanın şeklide (1, 56, 56, 256)'dır. Bu katmanda (14,94,103,201) nolu filtreler görselleştirilmiştir. 103 nolu filtre dışındakiler kenarlara odaklanmışken, 103 nolu filtre renk ve orta kısma yoğunlaşmıştır.



Şekil 4. 55: VGG16 modeli K=10 ve E=10 için sağlıklı ara filtre çıktıları



Şekil 4. 56: VGG16 modeli K=10 ve E=10 için pnömonili ara filtre çıktıları

4.1.4 Modellerin Başarı Ölçüm Sonuçları ve Karşılaştırmaları

Bu tez çalışmasında derin öğrenme algoritmaları kullanılarak röntgen görüntülerinden pnömoni hastalığı tespit edilmeye çalışılmıştır. Yapılan çalışmalarda öncelikle modeller rastgele olarak %70 eğitim, %20 test, %10 validasyon olarak ayrılmıştır. Daha sonra aynı modeller için k-fold yöntemi çalışılmıştır. Rastgele veri kümesini bölme yönteminde sentetik veri artırma kullanılmıştır.

CNN modeli sıfırdan kurulmuş olup, bu model üzerinde veri artırma yönteminin başarısı ölçülmüştür. Ayrıca aynı modele k-fold yöntemi uygulanmış fakat bu yöntem yetersiz kalmıştır, model overfit olmuştur. CNN modeline uygulanan k – fold yönteminin sonuçlarına bu yüzden yer verilmemiştir.

Inception modeli öncelikle rastgele bölünmüş veri kümesindeki yöntem ile çalıştırılmıştır. Bu sonuçları CNN modeli ile karşılaştırdığımızda, CNN modelinin Inception modelinin ilk deneyinden daha başarılı sonuçlar sağladığını söyleyebiliriz. Aynı model üzerinde k- fold yöntemi uygulandığında recall, precision, acc, tpr vb. gibi ölçüm sonuçlarında iyileşmeler görünürken roc eğrisinin altında kalan ve modelin güvenilirliği hakkında bilgi veren auc değerinde ise ciddi bir artış görülmeyip bazı deneylerde bu değerde düşüşler gözlemlenmiştir.

Çizelge 4. 20: Modellerin ölçüm sonuçlarının karşılaştırması

		Recall	Precision	Acc	Tpr	Fpr	Ppv	Tnr	F1 Skor	Roc-Auc
CNN	Deney-1	0,979	0,793	0,83	0,979	0,427	0,793	0,573	0,876	0,78
CNN	Deney-2	0,99	0,72	0,76	0,99	0,624	0,726	0,376	0,839	0,69
Ineption	Deney-1	0,828	0,778	0,745	0,828	0,393	0,778	0,607	0,802	0,7175
Ineption	Deney-2	0,886	0,854	0,804	0,886	0,429	0,854	0,571	0,870	0,7288
Ineption	Deney-3	0,952	0,818	0,805	0,952	0,641	0,818	0,359	0,880	0,6554
Ineption	Deney-4	0,947	0,826	0,813	0,947	0,564	0,826	0,436	0,882	0,6915
VGG16	Deney-1	0,985	0,821	0,856	0,985	0,359	0,821	0,641	0,895	0,8128
VGG16	Deney-2	0,965	0,980	0,960	0,965	0,055	0,980	0,945	0,973	0,9553
VGG16	Deney-3	0,975	0,988	0,973	0,975	0,034	0,988	0,966	0,982	0,9703
VGG16	Deney-4	0,940	0,990	0,949	0,940	0,025	0,990	0,975	0,965	0,9574

VGG16 modeli incelendiğinde ise, rastgele bölünmüş veri kümesindeki yöntem ile CNN ve Inception modelinin sonuçlarından daha iyi bir sonuç elde edilmiştir. Aynı şekilde VGG16 modeline k-fold yöntemi uygulandığında sonuçlarda ciddi iyileşmeler görülmüştür. Acc değeri %97,3 ‘e çıkmış ve Fpr değeri de 0.025’e kadar düşmüştür. Deneylerin detaylı sonuçları Çizelge 4.20’de verilmiştir. Her bir ölçüm parametresinde en iyi değerler koyu mavi en kötü değerler açık mavi olacak şekilde

renklendirilmiştir. Çizelge dikkatli incelendiğinde VGG16 modelinde koyu mavi renk tonunun yoğunlaştığı görülmektedir.



5.SONUÇ VE ÖNERİLER

Pnömoni, Türkiye’de yaygın olan ve ölümlerle sonuçlanabilen sağlık problemleri arasında yer almaktadır. Bu hastalığın erken teşhisinde yararlanılan görüntüleme işlemlerinden biri akciğer görüntüleridir. Bu tez çalışmasında hasta ve sağlıklı insanların akciğer görüntülerinden oluşan veri kümesi incelenmiş olup, derin öğrenme ile hastalık tanısının sınıflandırılabilmesi için modeller oluşturulmuştur. Oluşturulan modeller, daha önce yapılmış çalışmaların üzerine kurulmuş olup, elde edilmiş olan başarıların üzerine çıkmıştır.

- Veri kümesi incelenirken dağılımdaki tespit edilen veri dengesizliği veri artırma yöntemi ile aşılmıştır. Veri kümesi kısıtlı olmasına rağmen bir önceki yapılan çalışmalardan %10 ile %20 daha yüksek başarı elde edilmiştir. Ayrıca veri artırma işleminin deneylerle model üzerindeki başarısı ölçülmüştür.
- Veri kümesindeki veri sayısı dengeli olacak şekilde artırıldığında her bir modeldeki başarı oranının artacağı düşünülmektedir.
- Veri sayısı az olduğundan, kanal sayısı bir olan (grayscale bir kanallı) veriler, kümeden çıkarılmak yerine kanal sayısı eklenerek (RGB üç kanallı) her model için işlenebilir hale dönüştürülmüştür.
- Aşırı öğrenmenin gerçekleşmemesi için erken durdurma, veri seyreltme ve normalizasyon işlemleri uygulanmıştır.
- Performans sorununu aşmak için derinlemesine ayrılabilir evrişim kullanılmıştır. Bu evrişim yöntemi ile yüksek hesaplama işlemlerinden kurtulmuş olup, daha performanslı ve hızlı sonuçlar alınabilmektedir.
- Hiper parametreler değiştirilerek başarı oranının daha da artırılabilindiği deneyler ile gözlemlenmiştir. Ayrıca görüntüleme tekniklerinde kullanılacak farklı sınıflandırma yöntemleri ile başarı oranları daha da yüksek değerlere ulaştırılabilmektedir.
- Veri kümesinin rastgele bölünmesi yönteminin yanı sıra k-fold yöntemi de uygulanarak ölçüm sonuçları karşılaştırılmış ve k-fold yönteminin başarılı olduğu modeller belirlenmiştir.

- Her modelin özellik haritaları (ara filtreleri), sağlıklı ve pnömonili veriler için görselleştirilerek algoritmaların görüntüleri nasıl öğrendiği, filtrelerin görüntülerde farklı katmanlarda nerelere yoğunlaştığı konusunda fikir edinilmiştir.

Bu bilgilerin yanı sıra, incelenen veri kümesindeki görüntüler frontal denilen önden çekilmiş görüntülerdir. Bu görüntüler ile benzer görüntülere sahip farklı hastalıkların karıştırılmaması için öncelikle görüntü içeriği lateral denilen daha kapsamlı görüntülerden oluşmalıdır. Oluşturulacak olan veri kümesindeki görüntülerin dağılımı dengeli olmalıdır. Ayrıca sağlık sorununu daha doğru tespit edebilmek için hastanın geçmişi ile veri kümesi zenginleştirilmelidir. İleride yapılacak çalışmalarda tespit edilen bu bulgular dikkate alınarak, kişilerin lateral, frontal görüntüleri ile demografik bilgileri ve hastalık geçmişi bir bütün olarak incelenecek olup, hepsini içeren bir model üzerinde çalışılacaktır.

KAYNAKLAR

- [1] **Yazgan, E.** (1992). Tıbbi Görüntüleme Teknikleri. EMO. Sayı:390. http://bbm.emo.org.tr/genel/katalog_detay.php?katalog=2&kayit=233
- [2] **Url-1** < www.toraks.org.tr/halk/News.aspx?detail=4298>, erişim tarihi 16.04.2019
- [3] **Aydogdu, M, Ozyilmaz, E, Aksoy, Handan, Gursel, G, and Ekim, Numan.** (2010). Mortality prediction in community-acquired pneumonia requiring mechanical ventilation; values of pneumonia and intensive care unit severity scores. *Tuberk Toraks*, 58(1):25-34.
- [4] **Bagheri M., Lu L., Lu Z., Summers R. M., Peng Y, Wang X.** (2017). Chestx-ray8: Hospital-scale chest x-ray database and benchmarks on weakly-supervised classification and localization of common thorax diseases. arXiv preprint arXiv:1705.02315, 2017.
- [5] **Bernard, D., Covington, B., Dagunts, D., Lyman, K., Poblentz, E., Yao, L.** (2017). Learning to diagnose from scratch by exploiting dependencies among labels. arXiv preprint arXiv:1710.10501, 2017.
- [6] **Bagul, A., Ball, R., Ding, D., Duan, T., Irvin, J., Lungren ,M. P., Langlotz, C., Mehta, H., Ng Andrew Y., Rajpurkar, P., Shpanskaya, K., Yang, B., Zhu, K.** (2017). CheXNet: Radiologist-Level Pneumonia Detection on Chest X-Rays with Deep Learning arXiv:1711.05225v3, 2017.
- [7] **Abiyev, Rahib H. , Sallam Ma'aitah , M.Khaleel.** (2018). Deep Convolutional Neural Networks for Chest Diseases Detection.
- [8] **Badgeley, M. A. , Costa, A. B., Liu, M., Oermann, E. K.,Titano, J. J., Zech, J.R.** (2018). Variable generalization performance of a deep learning model to detect pneumonia in chest radiographs: A cross-sectional study
- [9] **Jeong, D. , Maduh, U. J., Sain, M., Stephen , O.** (2019). An Efficient Deep Learning Approach to Pneumonia Classification in Healthcare
- [10] **Baltruschat, I. M. , Grass, M. , Knopp, T., Nickisch, H. , Saalbach, A.** (2019). Comparison of Deep Learning Approaches for Multi-Label Chest X-Ray Classification.
- [11] **Lee, K., Rubin, J., Sanghavi, D., Zhao, C., Qadir, A., Xu-Wilson, M.** (2018). Large Scale Automated Reading of Frontal and Lateral Chest X-Rays using Dual Convolutional Neural Networks.
- [12] **Bülbül, Y.** (2014). Türkiye’de Toplumda Gelişen Pnömoni Verileri.
- [13] **Url-2** < <https://www.nhlbi.nih.gov/health-topics/pneumonia>>, erişim tarihi 12.04.2019

- [14] **Ashby, B., Turkington, C.** (2007). The encyclopedia of infectious diseases (3rd ed.). New York: Facts on File. p. 242. ISBN 978-0-8160-6397-0. Retrieved 21 April 2019.
- [15] **Uçan, E. S.** (2010). Göğüs Hastalıklarında Aciller, Erişim 04 Nisan 2019. http://file.toraks.org.tr/TORAKSFD23NJKL4NJ4H3BG3JH/kisokulu5-ppt-pdf/eyup_sabri_ucan_gogus.pdf
- [16] **Bülbül, Y. , Özlü, T. , Özsu, S.** (2007). Ulusal Verilerle Toplum Kökenli Pnömoniler.
- [17] **Şen, Nazan, Özhan, M.H.** (2017). Pnömoni. Erişim 20 Nisan 2019. http://www.solunum.org.tr/TusadData/Book/555/1712017101925-Pnomoni_Kitabi.pdf
- [18] **Taşbakan, M. Sezai.** (2019). Pnömoniler. Erişim 22 Nisan 2019. http://file.atuder.org.tr/_atuder.org/fileUpload/vtUjrNAhLL1x.pdf
- [19] **Zatürre (Pnömoni) Hakkında Herşey.** (2019). Erişim 22 Nisan 2019. <https://www.toraks.org.tr/halk/News.aspx?detail=4716>
- [20] **Ruuskanen, O., Lahti, E., Jennings, LC., Murdoch, DR.** (2011). Viral pneumonia. Lancet.
- [21] **Feldman, C., Anderson R.** (2012). Antibiotic resistance of pathogens causing community-acquired pneumonia.
- [22] **Singh V, Aneja S.** (2011). Pneumonia – management in the developing world. Paediatric Respiratory Reviews.
- [23] **Waqas, Q.A., Abdullah, H.M.A., Khan, U.I., Oliver, T.** (2019). Human cytomegalovirus pneumonia in an immunocompetent patient: a very uncommon but treatable condition.
- [24] **Tanzella, G., Motos, A., Battaglini, D., Meli, A., Torres, A.** (2019). Optimal approaches to preventing severe community-acquired pneumonia.
- [25] **Virkki, R., Juven, T., Rikalainen, H., Svedström, E., Mertsola, J., Ruuskanen, O.** (2002). Differentiation of bacterial and viral pneumonia in children. Erişim 15 Nisan 2019. <https://thorax.bmj.com/content/57/5/438>
- [26] **Erdur., B.** (2011). Akciğer Grafisi Yorumlama. Erişim 12 Nisan 2019. http://file.atuder.org.tr/_atuder.org/fileUpload/cfgPfxtrY2F.pdf
- [27] **Url-3** < <https://www.nih.gov/news-events/news-releases/nih-clinical-center-provides-one-largest-publicly-available-chest-x-ray-datasets-scientific-community> >, erişim tarihi 17.04.2019
- [28] **What is max pooling in convolutional neural networks.** (2019). Erişim 28 Nisan 2019. <https://www.quora.com/What-is-max-pooling-in-convolutional-neural-networks>
- [29] **Ng, Andrew.** (2018) Machine Learning Yearning. Identifying Bias, Variance, and Data Mismatch Errors. (Vol. 41, pp.79-80)
- [30] **Ng, Andrew.** (2018) Machine Learning Yearning, Examples of Bias and Variance (Vol. 20, pp.41-45)
- [31] **Bengio, Y., Courville, A., Goodfellow, I.** (2017). Deep Learning. Conditional Log-Likelihood and Mean Squared Error. (Vol.5, pp.133-139)

- [32] **Burkov, A.** (2018). The Hundred-Page Machine Learning Book. Model Performance Assessment.(Vol.5, pp.13-19)
- [33] **Url-4** < https://en.wikipedia.org/wiki/Sensitivity_and_specificity>, erişim tarihi 28.04.2019
- [34] **Gibson, A. , Patterson, J.** (2017). Deep Learning A Practitioner’s Approach. Controlling Epochs and Mini-Batch Size. (Vol. 6, pp.273 – 275)
- [35] **Bengio, Y., Courville, A.,Goodfellow, I.** (2017). Deep Learning. Deep Feedforward Networks. (Vol.6, pp.168-197)
- [36] **Gibson, A., Patterson, J.** (2017). Deep Learning A Practitioner’s Approach. Activation Functions.(Vol.3, pp.93-95)
- [37] **Ng .Andrew.** (2018).Understanding mini-batch gradient descent. Erişim 14 Temmuz 2019. <https://cs230.stanford.edu/files/C2M2.pdf>
- [38] **Bengio, Y., Courville, A.,Goodfellow, I.** (2017). Deep Learning. Gradient-Based Optimization.(Vol.4, pp.82-93)
- [39] **Ashia C. Wilson, Rebecca Roelofs, Mitchell Stern, Nathan Srebro, Benjamin Recht.** (2017). The Marginal Value of Adaptive Gradient Methods in Machine Learning.
- [40] **Ruder, Sebastian.** (2017).An overview of gradient descent optimization algorithms
- [41] **Buduma,N.** (2017).Fundamentals of Deep Learning Designing Next-Generation Machine Intelligence Algorithms.Sigmoid, Tanh, and ReLU Neurons.(Vol.1, pp.13-17).
- [42] **Buduma,N.** (2017).Fundamentals of Deep Learning Designing Next-Generation Machine Intelligence Algorithms. Preventing Overfitting in Deep Neural Networks. (Vol.2, pp.34-37).
- [43] **Gibson, A. , Patterson, J.** (2017). Deep Learning A Practitioner’s Approach. How to Use Regularization. (Vol.6, pp.275-281)
- [44] **Trask, W. Andrew.** (2017) Grokking Deep Learning.Calculating Both direction and amount from error.(Vol.4, pp.56-66)
- [45] **Bengio, Y., Courville, A.,Goodfellow, I.** (2017). Deep Learning. Optimization for Training Deep Models.(Vol.8, pp.274-330).
- [46] **Bengio, Y., Courville, A.,Goodfellow, I.** (2017). Deep Learning. Batch Normalization.(Vol.8, pp.317-321)
- [47] **Ghosh, Tapabrata.** (2017). Towards a New Interpretation of Separable Convolutions.
- [48] **Url-5** < <https://towardsdatascience.com/review-xception-with-depthwise-separable-convolution-better-than-inception-v3-image-dc967dd42568> >, erişim tarihi 25.08.2019
- [49] **Chollet, François.** (2017). Xception: Deep Learning with Depthwise Separable Convolutions.
- [50] **Url-6** < <https://towardsdatascience.com/a-basic-introduction-to-separable-convolutions-b99ec3102728> > , erişim tarihi 25.08.2019

- [51] **Url-7** < <https://machinelearningmastery.com/how-to-visualize-filters-and-feature-maps-in-convolutional-neural-networks/> > , erişim tarihi 31.08.2019
- [52] **Chen., X. , Liu, C. , Qin, Z. , Yu, F.** (2018). How Convolutional Neural Networks See The World A Survey of Convolutional Neural Network Visualization Methods
- [53] **Perez, L., Wang, J.** (2017). The Effectiveness of Data Augmentation in Image Classification using Deep Learning.
- [54] **DeVries, T., Taylor, G.W.** (2017). Dataset Augmentation in Feature Space.

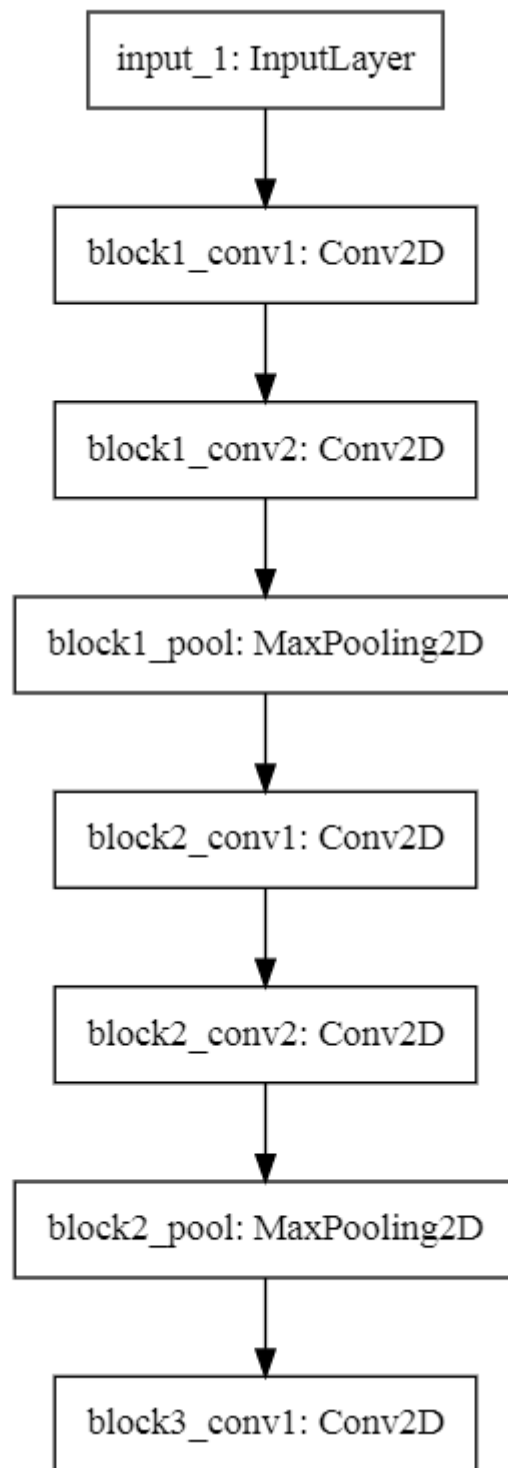


EKLER

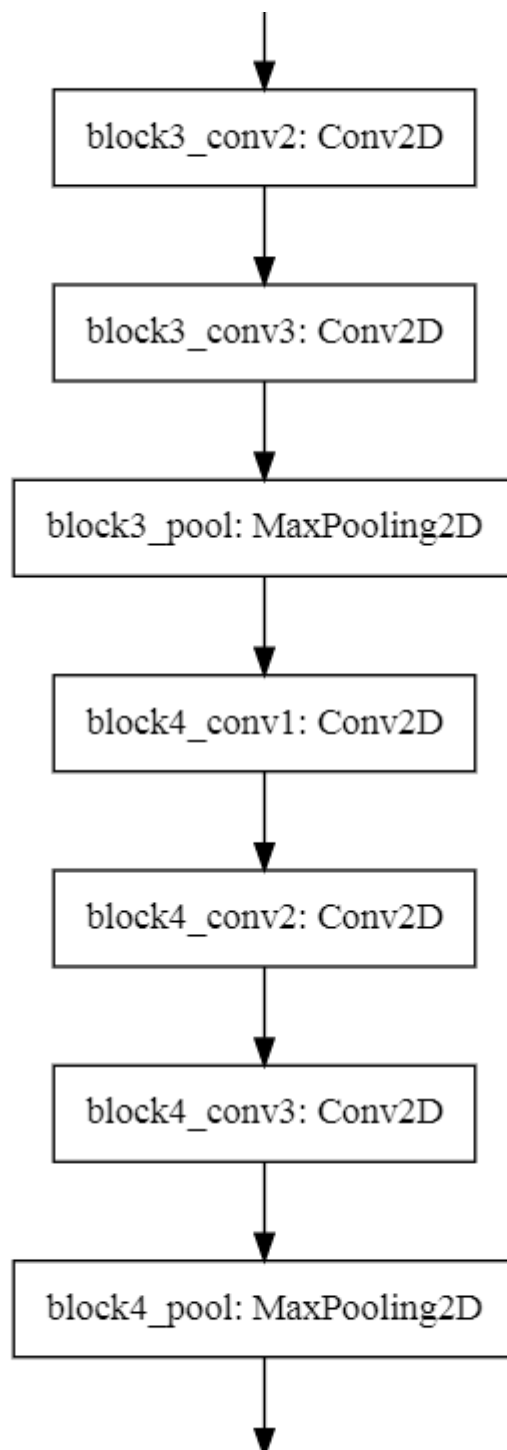
EK A: VGG16 Modelinin Diagramı



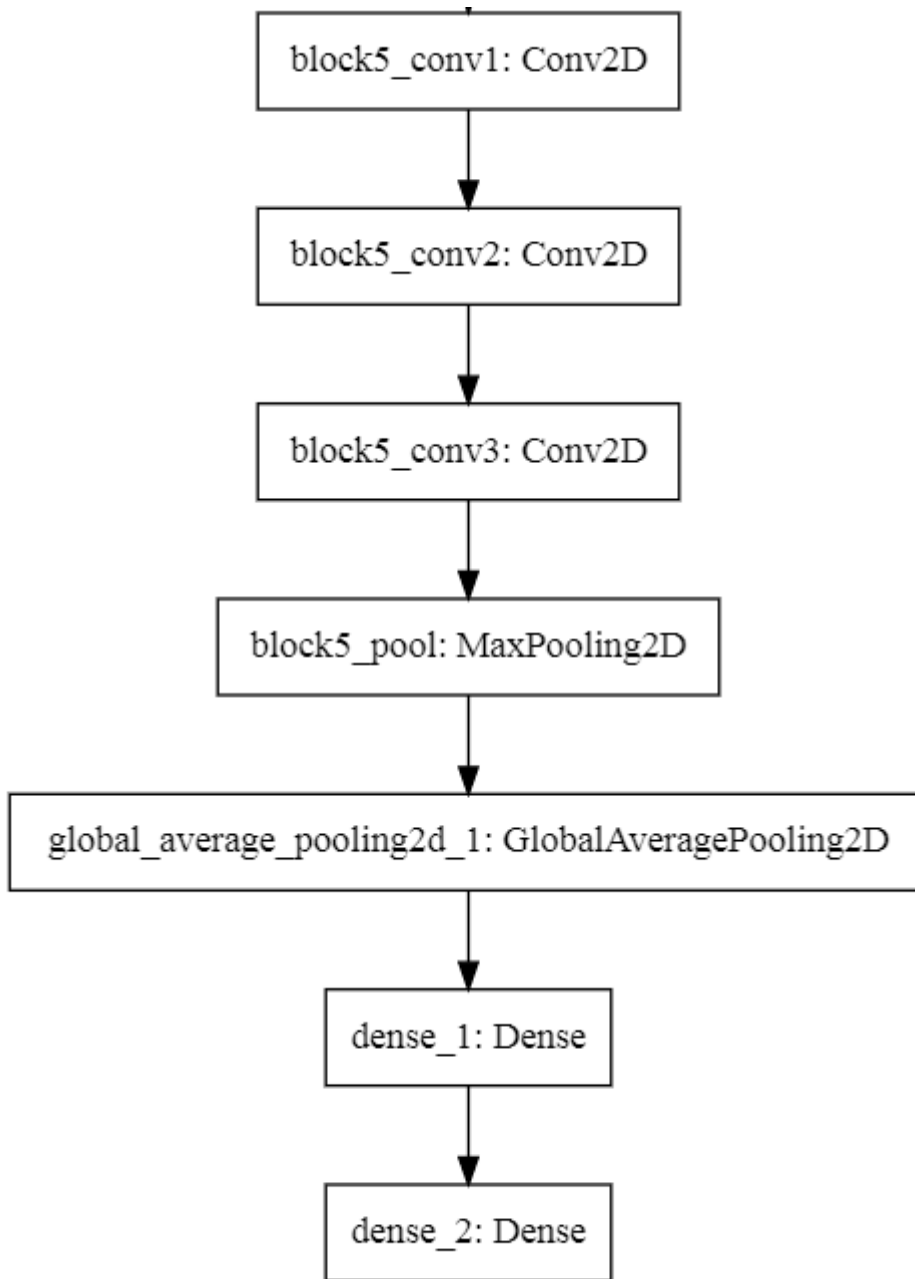
EK A:



EK A (Devam):



EK A (Devam):



Şekil A. 1: VGG16 Modelinin Diagramı

ÖZGEÇMİŞ

Ad-Soyad : Gamze Akpolat
Doğum Tarihi ve Yeri : 1990-Türkiye
E-posta : colabbasakt@gmail.com

ÖĞRENİM DURUMU:

- **Lisans** : 2015, İstanbul Üniversitesi, Mühendislik Fakültesi, Elektrik-Elektronik Mühendisliği
- **Yükseklisans** : 2019, İstanbul Teknik Üniversitesi, Elektronik ve Haberleşme Mühendisliği, Telekomünikasyon Mühendisliği

MESLEKİ DENEYİM VE ÖDÜLLER:

- 2015-2017 yılları arası özel sektör IT Veri Analisti Uzmanı
- 2017-2019 yılları arası özel sektör IT Raporlama İş Çözümleri ve Veri Yönetimi Uzmanı