

T.C.
BOLU ABANT İZZET BAYSAL ÜNİVERSİTESİ
LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ
ELEKTRİK-ELEKTRONİK MÜHENDİSLİĞİ ANABİLİM
DALI



VERİ ÇOĞALTMA VE DERİN ÖĞRENME TEKNİKLERİYLE
MEDİKAL GÖRÜNTÜLERDEN OTOMATİK HASTALIK
TESPİTİ

YÜKSEK LİSANS TEZİ

TESLİME BAYIK

TEZ DANIŞMANI
PROF.DR KEMAL POLAT

BOLU, TEMMUZ - 2023

KABUL VE ONAY SAYFASI

TESLİME BAYIK tarafından hazırlanan “**Veri Çoğaltma Ve Derin Öğrenme Teknikleriyle Medikal Görüntülerden Otomatik Hastalık Tespiti**” adlı tez çalışması jürimiz tarafından Elektrik Elektronik Mühendisliği Anabilim Dalı’nda Yüksek Lisans olarak oy birliği/ oy çokluğuyla kabul edilmiştir.
27/07/2023

Jüri Üyeleri

İmza

Danışman
Prof. Dr. Kemal POLAT
Bolu Abant İzzet Baysal Üniversitesi

.....

Üye
Dr. Öğr. Üyesi Fatih KAYAALP
Düzce Üniversitesi

.....

Üye
Dr. Öğr. Üyesi Mithat ÖNDER
Bolu Abant İzzet Baysal Üniversitesi

.....

Lisansüstü Eğitim Enstitüsü Onayı

Prof. Dr. İbrahim KÜRTÜL
Lisansüstü Eğitim Enstitüsü Müdürü

ETİK BEYAN

Bolu Abant İzzet Baysal Üniversitesi, Lisansüstü Eğitim Enstitüsü Tez Yazım Kurallarına uygun olarak hazırladığım bu tez çalışmada;

- Tez içinde sunduğum verileri, bilgileri ve dokümanları akademik ve etik kurallar çerçevesinde elde ettiğimi,
- Tüm bilgi, belge, değerlendirme ve sonuçları bilimsel etik ve ahlak kurallarına uygun olarak sunduğumu,
- Tez çalışmada yararlandığım eserlerin tümüne uygun atıfta bulunarak kaynak gösterdiğimi,
- Kullanılan verilerde herhangi bir değişiklik yapmadığımı,
- Bu tezde sunduğum çalışmanın özgün olduğunu bildirir,

aksi bir durumda aleyhime doğabilecek tüm hak kayıplarını kabullendiğimi beyan ederim.

Teze ilişkin 22/08/2023 tarihinde Turnitin adlı intihal tespit programından enstitü müdürlüğünce belirlenen filtrelemeler uygulanarak alınmış olan benzerlik raporuna göre, tezin benzerlik oranı % 14 olarak tespit edilmiştir.

.....
TESLİME BAYIK

ÖZET

**VERİ ÇOĞALTMA VE DERİN ÖĞRENME TEKNİKLERİYLE
MEDİKAL GÖRÜNTÜLERDEN OTOMATİK HASTALIK TESPİTİ
YÜKSEK LİSANS TEZİ
TESLİME BAYIK
BOLU ABANT İZZET BAYSAL ÜNİVERSİTESİ
LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ
ELEKTRİK-ELEKTRONİK MÜHENDİSLİĞİ ANABİLİM DALI**

(TEZ DANIŞMANI: PROF.DR KEMAL POLAT)

BOLU, TEMMUZ - 2023

XV + 64

2019 yılında, Covid-19 hastalığının tüm dünyaya yayılması, yaşamı negatif olarak etkiledi. Covid-19 hastalığı, zatürre hastalığının bir çeşididir. Zatürre hastalarının, derin öğrenme yöntemiyle erken teşhisi, bu tez çalışmanın ana konusudur. Erken teşhis için bireylerin göğüs röntgen görüntüleri (X-Ray görüntüleri), bu tez çalışmada veri kümesi olarak kullanılmıştır. Kaggle veri sitesi üzerindeki, CoronaHack-Chest-Xray, bu tezde veri kümesi olarak kullanılmıştır. Veri kümesindeki görüntülerin; zatürre ve normal olarak, iki farklı sınıfta etiketli olduğu görülmüştür.

VGG-16, VGG-19, MobileNet, InceptionV3, Xception; derin öğrenme modelleri bu çalışmada, ayrı ayrı kullanılmıştır. Her bir model 4 farklı metot ile çalıştırılmıştır. Metotlar; 1- Veri artırımı yok ve çapraz doğrulama yok, 2- Veri artırımı yok, çapraz doğrulama var, 3-Veri artırımı var, çapraz doğrulama yok, 4- Veri artırımı var, çapraz doğrulama var. Tüm model ve metodu eşleşme kombinasyonları çalıştırıldığında, CoronaHack-Chest-Xray veri kümesi üzerinde, MobileNet modeli, veri artırımı yok-çapraz doğrulama yok metodu ile, eğitim kümesi %80 – test kümesi %20 olarak, en iyi başarı metrik sonuçları vermiştir. En iyi sonuca sahip modelin; görüntüleri, zatürre ve normal olarak 2 sınıfa ayırmasının, başarı metrikleri özetle şunlardır: Doğruluk metriği %99, kesinlik metriği %98, hatırlama metriği %98. Diğer başarı metrik sonuçları, karmaşıklık matrisi ve diğer model-metot kombinasyonlarının çalıştırılma başarı metrikleri bu tezde elde edilmiştir.

ANAHTAR KELİMELER: Derin Öğrenme, Veri Çoğaltma, Medikal Görüntüleme, Covid-19, Zatürre

ABSTRACT

AUTOMATIC DISEASE DETECTION FROM MEDICAL IMAGES WITH DATA AUGMENTATION AND DEEP LEARNING TECHNIQUES

MSC THESIS

TESLİME BAYIK

BOLU ABANT İZZET BAYSAL UNIVERSITY

INSTITUTE OF GRADUATE STUDIES

DEPARTMENT OF ELECTRICAL AND ELECTRONICS ENGINEERING

(SUPERVISOR: PROF.DR KEMAL POLAT)

BOLU, JULY 2023

XV + 64

In 2019, the spread of the Covid-19 disease worldwide negatively affected life. Covid-19 disease is a type of pneumonia disease. Early diagnosis of pneumonia patients with deep learning method is the main subject of this thesis study. Chest X-rays (X-Ray images) of individuals for early diagnosis were used as dataset in this thesis study. This study used coronaHack-Chest-Xray on the Kaggle data site as a dataset. The images in the dataset; It is labeled into two different classes, pneumonia and normal.

VGG-16, VGG-19, MobileNet, InceptionV3, and Xception; deep learning models were used separately in this thesis. Each model was run with 4 different methods. Methods; 1- No data augmentation and cross-validation, 2- No data augmentation, cross-validation, 3-Data augmentation, no cross-validation, 4-Data augmentation, cross-validation. When all model and method matching combinations were run, the MobileNet model gave the best success metric results on the CoronaHack-Chest-Xray dataset, with the no data augmentation-no cross-validation method, training set 80% – test set 20%. The model with the best result, The success metrics of separating the images into 2 classes, pneumonia and normal, are summarized as follows: the accuracy metric is 99%, the precision metric is 98%, and the recall metric is 98%. Other success metric results, complexity matrix, and running success metrics of other model-method combinations were obtained in this thesis.

KEYWORDS: Deep Learning, Data Augmentation, Medical Imaging, Covid-19, Pneumonia

İÇİNDEKİLER

Sayfa

KABUL VE ONAY SAYFASI	iii
ETİK BEYAN.....	iv
ÖZET.....	v
ABSTRACT	vi
İÇİNDEKİLER	vii
ŞEKİL LİSTESİ.....	ix
TABLO LİSTESİ	xii
FOTOĞRAF LİSTESİ	xiii
KISALTMA VE SEMBOLLER LİSTESİ	xiv
TEŞEKKÜR	xv
1. GİRİŞ.....	1
2. GENEL BİLGİLER	3
2.1 Çalışmanın Amacı Ve Literatür Taraması.....	3
2.2 Yapay Zekâ.....	7
2.2.1 Makine Öğrenmesi ve Derin Öğrenme.....	8
3. MATERYAL VE YÖNTEM	11
3.1 Kullanılan Veri kümesi (Veri Seti) Ve Kullanılan Kütüphaneler	11
3.2 Yöntem	13
3.3 Derin Öğrenme Ve Yapay Sinir Ağları	13
3.3.1 Yapay Sinir Ağları.....	14
3.4 Evrimsel Sinir Ağları (CNN) Ve Katmanları	16
3.4.1 Giriş Katmanı	17
3.4.2 Evrişim Katmanı.....	17
3.4.3 Havuzlama Katmanı	18
3.4.4 Düzleştirme.....	19
3.4.5 Tam Bağlantı Katmanı.....	19
3.4.6 Bırakma Katmanı.....	20
3.4.7 Sınıflandırma Katmanı	20
3.4.8 Aktivasyon Katmanı	20
3.4.8.1 Birim Basamak Fonksiyonu	20
3.4.8.2 Doğrusal Fonksiyon	20
3.4.8.3 ReLU Fonksiyonu	20
3.4.8.4 Sızıntı ReLU Fonksiyonu.....	20
3.4.8.5 Sigmoid Fonksiyonu	20
3.4.8.6 Hiperbolik Tanjant Fonksiyonu	21
3.4.8.7 Swish Fonksiyonu (Kendinden Geçitli).....	21
3.4.8.8 Softmax Fonksiyonu	21
3.4.9 Normalizasyon Katmanı	22

3.4.10 Kayıp Fonksiyonu	22
3.5 Kullanılan Derin Öğrenme Modelleri.....	22
3.5.1 VGG-16	22
3.5.2 VGG-19	23
3.5.3 MobileNet/MobileNetV1/MobileNetV2	24
3.5.4 GoogleNet (InceptionV3)	25
3.5.5 Xception.....	27
3.6 Veri Artırımı	27
3.7 Sinir Ağının Eğitilmesi	27
3.8 Sınıflandırıcı Performans Ölçütleri	28
3.8.1 Çapraz Doğrulama	28
3.8.2 Karmaşıklık Matrisi	28
4. BULGULAR	32
5. SONUÇ VE ÖNERİLER	59
5.1 Öneriler	59
5.2 Sonuçlar	60
6. KAYNAKLAR.....	62

ŞEKİL LİSTESİ

Sayfa

Şekil 2.1	Yapay zekâ ve alt bölümleri	8
Şekil 2.2	Makine öğrenmesi metotlarının sınıflandırılması(26)	9
Şekil 3.1	Veri kümesi içerisindeki bireylerin yüzdelik dağılımı	11
Şekil 3.2	Veri kümesi içerisindeki bireylerin sayısal dağılımı	12
Şekil 3.3	Veri kümesine uygulanan yöntem için işlem şeması.....	13
Şekil 3.4	Derin Öğrenme Genel Mimarisi	14
Şekil 3.5	Yapay Sinir Ağı Genel İşlevi.....	16
Şekil 3.6	Evrişimsel sinir ağları genel yapısı.....	16
Şekil 3.7	5×5 Boyutlu girilen bir matrisin 3×3 boyutlu filtreyle evrişimi	18
Şekil 3.8	4×4 Boyutlu girilen bir matrisin maksimum havuzlama işlemi	18
Şekil 3.9	4×4 Boyutlu girilen bir matrisin ortalama havuzlama işlemi	19
Şekil 3.10	3×3 Boyutlu girilen bir matrisin düzleştirme örneği şekli.....	19
Şekil 3.11	Aktivasyon fonksiyonu olan Sigmoid'in gösterimi.....	21
Şekil 3.12	Aktivasyon fonksiyonlarının grafikte toplu biçimde görünüşü.....	21
Şekil 3.13	Aktivasyon fonksiyonlarının genel gösterim şekilleri.....	22
Şekil 3.14	VGG-16 mimarisine sigmoid fonksiyonu eklenmiş görüntüsü.....	23
Şekil 3.15	MobileNet evrişim blok gösterimi.....	24
Şekil 3.16	MobileNetV2 evrişim blok gösterimi.....	25
Şekil 3.17	Naif Inception modülü gösterimi.....	26
Şekil 3.18	Boyut indirgemeli InceptionV3 modülü.....	26
Şekil 3.19	K=5 değeri için çapraz doğrulama şekli	28
Şekil 3.20	VGG-16 modeline ait bir ROC Eğrisi örneği.....	31
Şekil 4.1	Çapraz doğrulama olmadan veri kümesinin eğitim/testinin %80 bölünmesi	32
Şekil 4.2	Çapraz doğrulama olmadan veri kümesinin eğitim/testinin %70 bölünmesi	33
Şekil 4.3	Çapraz doğrulama olmadan veri kümesinin eğitim/testinin %50 bölünmesi	33
Şekil 4.4	MobileNet Modelinin Çapraz Doğrulama uygulanmamış Eğitim/test=0.2 seçilmiş grafiği	36
Şekil 4.5	VGG-16 Modeline veri artırımı uygulanmamış çapraz doğrulamalı grafik	39
Şekil 4.6	VGG-16 Modeline veri artırımı uygulanmış çapraz doğrulamalı grafik	39
Şekil 4.7	VGG-16 Modeline veri artırımı uygulanmamış çapraz doğrulama uygulanmamış grafik.....	40
Şekil 4.8	VGG-16 Modeline veri artırımı uygulanmış çapraz doğrulama uygulanmamış grafik.....	40
Şekil 4.9	VGG-19 Modeline veri artırımı uygulanmamış çapraz doğrulama uygulanmamış grafik.....	41
Şekil 4.10	VGG-19 Modeline veri artırımı uygulanmış çapraz doğrulama uygulanmamış grafik.....	41
Şekil 4.11	VGG-19 Modeline veri artırımı uygulanmamış çapraz doğrulama uygulanmamış grafik.....	42

Şekil 4.12	VGG-19 Modeline veri artırımı uygulanmış çapraz doğrulama uygulanmamış grafik.....	42
Şekil 4.13	MobileNet Modeline veri artırımı uygulanmış çapraz doğrulama uygulanmamış grafik.....	43
Şekil 4.14	MobileNet Modeline veri artırımı uygulanmamış çapraz doğrulama uygulanmamış grafik.....	43
Şekil 4.15	MobileNet Modeline veri artırımı uygulanmış çapraz doğrulama uygulanmamış grafik.....	44
Şekil 4.16	MobileNet Modeline veri artırımı uygulanmamış çapraz doğrulama uygulanmamış grafik.....	44
Şekil 4.17	InceptionV3 Modeline veri artırımı uygulanmış çapraz doğrulama uygulanmamış grafik.....	45
Şekil 4.18	InceptionV3 Modeline veri artırımı uygulanmamış çapraz doğrulama uygulanmamış grafik.....	45
Şekil 4.19	InceptionV3 Modeline veri artırımı uygulanmış çapraz doğrulama uygulanmamış grafik.....	46
Şekil 4.20	InceptionV3 Modeline veri artırımı uygulanmamış çapraz doğrulama uygulanmamış grafik.....	46
Şekil 4.21	Xception Modeline veri artırımı uygulanmış çapraz doğrulama uygulanmamış grafik.....	47
Şekil 4.22	Xception Modeline veri artırımı uygulanmamış çapraz doğrulama uygulanmamış grafik.....	47
Şekil 4.23	Xception Modeline veri artırımı uygulanmış çapraz doğrulama uygulanmamış grafik.....	48
Şekil 4.24	Xception Modeline veri artırımı uygulanmamış çapraz doğrulama uygulanmamış grafik.....	48
Şekil 4.25	VGG-16 Modeline veri artırımı uygulanmamış çapraz doğrulama uygulanmamış üçlü grafik.....	49
Şekil 4.26	VGG-16 Modeline veri artırımı uygulanmış çapraz doğrulama uygulanmamış üçlü grafik.....	49
Şekil 4.27	VGG-16 Modeline veri artırımı uygulanmamış çapraz doğrulama uygulanmamış üçlü grafik.....	50
Şekil 4.28	VGG-16 Modeline veri artırımı uygulanmış çapraz doğrulama uygulanmamış üçlü grafik.....	50
Şekil 4.29	VGG-19 Modeline veri artırımı uygulanmamış çapraz doğrulama uygulanmamış üçlü grafik.....	51
Şekil 4.30	VGG-19 Modeline veri artırımı uygulanmış çapraz doğrulama uygulanmamış üçlü grafik.....	51
Şekil 4.31	VGG-19 Modeline veri artırımı uygulanmamış çapraz doğrulama uygulanmamış üçlü grafik.....	52
Şekil 4.32	VGG-19 Modeline veri artırımı uygulanmış çapraz doğrulama uygulanmamış üçlü grafik.....	52
Şekil 4.33	MobileNet Modeline veri artırımı uygulanmış çapraz doğrulama uygulanmamış üçlü grafik.....	53
Şekil 4.34	MobileNet Modeline veri artırımı uygulanmamış çapraz doğrulama uygulanmamış üçlü grafik.....	53
Şekil 4.35	MobileNet Modeline veri artırımı uygulanmış çapraz doğrulama uygulanmamış üçlü grafik.....	54
Şekil 4.36	MobileNet Modeline veri artırımı uygulanmamış çapraz doğrulama uygulanmamış üçlü grafik.....	54

Şekil 4.37	InceptionV3 Modeline veri artırımı uygulanmamış çapraz doğrulama uygulanmış üçlü grafik.....	55
Şekil 4.38	InceptionV3 Modeline veri artırımı uygulanmış çapraz doğrulama uygulanmış üçlü grafik.....	55
Şekil 4.39	InceptionV3 Modeline veri artırımı uygulanmamış çapraz doğrulama uygulanmamış üçlü grafik.....	56
Şekil 4.40	InceptionV3 Modeline veri artırımı uygulanmış çapraz doğrulama uygulanmamış üçlü grafik.....	56
Şekil 4.41	Xception Modeline veri artırımı uygulanmış çapraz doğrulama uygulanmış üçlü grafik.....	57
Şekil 4.42	Xception Modeline veri artırımı uygulanmamış çapraz doğrulama uygulanmış üçlü grafik.....	57
Şekil 4.43	Xception Modeline veri artırımı uygulanmış çapraz doğrulama uygulanmamış üçlü grafik.....	58
Şekil 4.44	Xception Modeline veri artırımı uygulanmamış çapraz doğrulama uygulanmamış üçlü grafik.....	58



TABLO LİSTESİ

Sayfa

Tablo 3.1 Veri setindeki görüntü sayısındaki dağılımı	11
Tablo 3.2 Karmaşıklık Matrisi	29
Tablo 4.1 Modellerde kullanılan çapraz doğrulama uygulanan görüntüler	35
Tablo 4.2. En iyi Sonuç Veren MobileNet Modeline Ait Tablo	36



FOTOĞRAF LİSTESİ

Sayfa

Fotoğraf 4.1 Veri kümesindeki Normal bireylerin farklı boyutlardaki görüntüsü	34
Fotoğraf 4.2. Veri kümesindeki Zatürre bireylerin farklı boyutlardaki görüntüsü	34
Fotoğraf 4.3 En iyi sonuç veren MobileNet modeline ait Roc eğrisi	37
Fotoğraf 4.4 En iyi sonuç veren MobileNet modeline ait Roc eğrisi yakınlaştırılmış görüntüsü	37
Fotoğraf 4.5 En iyi sonuç veren MobileNet modeline ait Karmaşıklık Matrisi	38



KISALTMA VE SEMBOLLER LİSTESİ

AI	: Yapay zeka (Artificial Intelligence)
App	: Appendix
AUC	: Eğri Altında Kalan Alan (The area under curve)
CNN	: Evrişimsel Sinir Ağı (Convolutional Neural Network)
CV	: Çapraz Doğrulama (Cross-Validation)
CXR	: Göğüs Röntgeni (Chest X-ray)
CT	: Bilgisayarlı Tomografi (Computed Tomography)
DL	: Derin öğrenme (Deep Learning)
ESA	: Evrişimsel Sinir Ağı (Convolutional Neural Network)
FCN	: Tamamen Evrişimsel Ağ (Fully Connected)
GiÜ	: Grafik İşleme Ünitesi (Graphics Processing Unit-GPU)
GN	: Gerçek Negatif (True Negative-TN)
GP	: Gerçek Pozitif (True Positive-TP)
KM	: Karmaşıklık Matrisi (Confusion Matrix-CM)
LSTM	: Uzun Kısa Vadeli Bellek (Long-Short Term Memory)
ROC	: Alıcı İşletim Karakteristik Eğrisi (Receiver Operating Characteristic Curve)
RELU	: Doğrultulmuş Doğrusal Birim (Rectified Linear Unit-RELU)
SMOTE	: Sentetik Azınlık Aşırı Örnekleme Tekniği (Synthetic Minority Over-Sampling)
YN	: Yanlış Negatif (False Negative-FN)
YP	: Yanlış Pozitif (False Positive-FP)
YSA	: Yapay Sinir Ağları (Artificial Neural Networks-ANN)

TEŞEKKÜR

Öğrenim hayatımın bir kısmı olan lisansüstü eğitimimde;

Bana yardımcı olan maddi ve manevi destekleyen en başta anneme ve babama olmak üzere, iyi günde kötü günde benim hep destekleyen kardeşlerime,

Beni bugüne kadar desteleyip yönlendiren saygıdeğer değerli Hocam Prof. Dr. Kemal Polat’a,

Tezimi bitirmemde beni sürekli motive eden arkadaşlarıma,

Teşekkür ederim.

Not: Bu tez Kahramanmaraş merkezli depremde ölen insanlara ve vefat eden anneanneme adanmıştır. Mekanları cennet olsun...

Teslime BAYIK

1. GİRİŞ

Medikal alanda değişimler teknolojinin sağlık alanında da ilerlemesiyle önemli derece yol almıştır.

Manyetik Rezonans (MR), Ultrason, Röntgen gibi cihazların icadıyla birlikte medikal alanda insanlardan elde edilen verilerin(görüntülerin) işlenmesi, toplanması ve yorumlanması ciddi önem arz etmektedir. Dünyada her yıl milyonlarca insanın ölümüne sebep olan ve Dünya Sağlık Örgütü (WHO) tarafından 2019 yılı verilerine göre ilk 3 sırada yer alan solunum yolu hastalıkları olmuştur(1). Bu rahatsızlıkların başında ise zatürre hastalığı yer almakta olup bulaşıcı bir hastalık olarak bilinmektedir. Günümüzde zatürre hastalığının tedavisi hastanın hikâyesi ve muayene bulguları değerlendirmeye alınarak, akciğer grafisi çekilerek ve birtakım tahliller yapılarak teşhis ve tedavi aşaması uygulanır(2). 2019 yılının Kasım ayında Covid-19 hastalığının ortaya çıkmasıyla solunum yoluna bağlı ölümlerin ve yoğun bakıma giren hasta sayısında artış gözlenmiştir. Koronavirüs grip benzeri şikâyetlere neden bir virüs olarak tanımlanmıştır. Çin'inin Hubei eyaletinin Wuhan şehrinden başlayan ve tüm dünyaya yayılan Koronavirüs, insandan insana geçmesiyle birlikte soğuk algınlığı benzerliği olan Ortadoğu Solunum Sendromu (MERS) ve şiddetli Akut Solunum Sendromu (SARS) gibi enfeksiyonlara neden olduğu bildirilmiştir(3). Zatürre ve Covid-19 gibi insandan insana bulaşan hastalıkların tedavisi hayati önem arz etmiş ve tüm dünyayı pandemi sürecine sürüklemiştir. Zatürre daha yönetilebilen hastalık gibi görünmektedir. Bilgisayarla tanı konulabilen en iyi zatürre klinik yöntemi Göğüs Röntgeni görüntüsüdür. X-Ray ışını görüntülerinde bazen belirsiz olabilmekle kafa karıştırıcı olabilmektedir. Bu nedenle radyologlar arasında zatürre teşhisi konusunda farklı görüşlere neden olmuştur. Bunlara rağmen zatürre ve Covid-19 zatürresi maliyetsiz olması, hızlı değerlendirebiliyor oluşu, bilgisayar destekli teşhis ve tanılarda Röntgenin (CXR) ve Bilgisayar Tomografisinin (BT) önemini göstermektedir(4). Koronavirüsün tanımlanmasında RT-PCR testi kullanılsa da RT-PCR'ın ölçümlerinde %60 ve %90 arasında doğruluk verse de hastalığın belirli bir evresinde yanlış bilgi sağlayabilmektedir(5). Bunların yanı sıra gelişen yapay zekâ sayesinde bilgisayar destekli görüntülerin arşivlenmesi ve geliştirilmesi konusunda algoritmalar geliştirilip sunulmuştur(6). Bu çalışma Kaggle çalışma platformu kullanılmış olup, Kaggle'ın veri kümesinden faydalanılıp, bireylerin akciğer göğüs

röntgeni görüntülerinin bilgisayar ortamında verilerin işlenmesi ve verinin farklılaştırılmasını çoğaltılarak yorumlanması şeklinde çalışılmıştır. Tüm modellerin çalışması bittiğinde; modellere ait grafikler ve metrik sonuçları kaydedilmiştir. Kullandığım veri kümesinin verilerinin dengesizliğinden dolayı birçok dezavantajla karşılaşmış olup, birçok derin öğrenme modelini uygulayıp, MobileNet modelinin en iyi sonuç verdiği görülmüştür.



2. GENEL BİLGİLER

2.1 Çalışmanın Amacı Ve Literatür Taraması

Yapay zekâ günümüzde her alanda kendini göstermeye başlamıştır. Yapay zekâ ile bağlantılı makine öğrenimi ve derin öğrenme bunun en başında gelmektedir. Medikal alanlarda da öncelikle Bilgisayarlı Tomografi (BT), X-RAY (X ışını), Manyetik Rezonans (MR) kullanmasıyla elde edilen sonuçlar daha iyi değerlendirilmesi açısından bazı işlemlere tabi tutulmaktadır. Bu işlemlerde teşhis için önem arz etmektedir. Bu çalışmanın amacı, Kaggle (7) veri platformundaki 5910 göğüs röntgeni resimlerinin Python programlama dili kullanılarak, bir hastanın zatürre(pnömoni) olup-olmadığına karar veren derin öğrenme modellerini geliştirmektir. Bu sayede teşhisi koyacak doktora yardımcı olabilmesi ve ayrıca fazla tetkik yapılmadan daha az maliyetli kullanım sunarak teşhislerin doğruluk oranını en yükseğe taşıyabilmektir.

M. Toğaçar çalışmasında, çocuklarda büyük oranda görülen zatürre veri setini kullanmıştır. 1583'ü normal, 4226'sı zatürre(pnömoni) olmak üzere 2 sınıf oluşturup toplamda 5849 veri kullanılmıştır. Mimari olarak Alex-net seçilmiş ve farklı boyutlu filtrelerle denenmiştir. AlexNet'in boyutuna indirgenen görüntüler, Temel Bileşenler Analizi kullanarak öznetelik indirgemesi gerçekleştirmiştir. Sınıflandırma yöntemleri kıyaslanarak en iyi sınıflandırıcıyı bulmuşlardır. Temel bileşenler analizi kullanılarak elde edilen en iyi sonuçlar Destek vektör makinelerinde görülmüştür. Özellik çıkarımı kullanılarak farklı sınıflandırıcılarla Temel Bileşen Analizi'nin sonuçları üzerinde etkisi belirlenmiştir(8).

E. Ayan ve ark. çalışmasında, 2 Evrişimli sinir ağı (CNN) modeli VGG-16 ve Xception'ı eğitimi sırasında transfer öğrenimi ve ince ayar kullanarak elde edilmiştir. Test sonuçlarına göre VGG-16'da %87, Xception'da %82 elde edilirken normal vakalarda VGG-16, zatürre vakalarında Xception modelinin daha başarılı olduğu verisine ulaşmışlardır. 5856 X-Ray görüntüsü bulunan veri setinde farklı değişken çözünürlüğündeki görüntülerden 1583 normal, 4273 zatürre(pnömoni) şeklinde ayrılmıştır. Ve veri kümesi 3'e bölünerek(eğitim-doğrulama-test) ayrılmıştır. Transfer öğrenimiyle birlikte de veri büyütme (data augmentation) modifiye edilmiştir(9).

V. Chouhan ve ark. çalışmasında, 5232 görüntü veri kümesinden, 1349 görüntünün normal bireylerin, 2538 bakteriyel zatürre ve 1345 görüntü virüs zatürresine ait olduğu zatürreyi incelemişlerdir. Kullanılan ImageNet veri seti üzerinden önceden eğitilmiş mimariler AlexNet, DenseNet121, InceptionV3, GoogleNet ve ResNet18'den ResNet18'in diğerlerinden daha iyi sonuç vermiş olup, toplu öğrenme (ensemble) model ile karşılaştırıldığında toplu öğrenme (ensemble) modelin yaklaşık %96,4 doğruluk elde etmişlerdir(10).

M. Farooq ve Abdul Hafeez çalışmalarında, açık erişimli kaynak kullanarak hazırlanan model de Covid-19'u zatürre vakalarından ayırmak ve doğru bir evrişimsel sinir ağı kullanmışlardır. 4 sınıfa ayrılan Covid-19 veri setinde önceden eğitilmiş ResNet-50 mimarisine ince ayar yapmak için 3 adımlı teknik sunmuşlardır. Covid-Resnet giriş görüntülerine aşamalarına farklı boyutlardaki piksel olarak her aşamada ince ayar yapıp, sadece 41 tekrarda(epoch) %96,23 doğruluk başarısı elde etmişlerdir(11).

Ioannis D. Apostolopoulos ve ark. çalışmalarında, bakteriyel zatürre (pnömoni), onaylanmış Covid-19 ve normal hastalar değerlendirilmiş olup, 2 farklı veri seti üzerinde çalışmışlardır. Veri kümesinde anormallik tespiti için de öğrenim aktarımı (transfer learning) kullanmışlardır. İlk veri setinde toplamda 1428 görüntü mevcut olup, 224'ü Covid-19, 700'ü bakteriyel zatürreye (pnömoniyeye) aittir. 2. veri setinde 1442 görüntü olup, 224'ü Covid-19, 714 bakteriyel ve viral zatürreye(pnömoni) olup, MobileNetV2'den de ulaşılan doğruluk değeri 2. veri seti için %96,78 bulunmuştur(12).

T. Öztürk ve ark. çalışmalarında, bir derin öğrenme modeli olarak DarkCovid-Net'i geliştirip, kullanmışlardır. Toplamda 125 Covid-19 pozitif, 500 zatürre(pneumonia) ve 500 bulgu yok (no-findings) olan görüntüleri kullanmışlardır. Aynı ayrı elde ettikleri doğruluk sıralaması 2'li sınıf için %98 iken 3'lü sınıf için %87.02 olmuştur(13).

T. Rahman ve ark. çalışmasında, aktarım öğrenimi için önceden eğitilmiş dört farklı derin Evrişimli Sinir Ağı (CNN): AlexNet, ResNet18, DenseNet201 ve SqueezeNet kullanılmıştır. Bakteriyel, viral ve normalden oluşan toplam 5247 Kaggle göğüs röntgeni zatürresi(pnömonisi) veri tabanı kullanılarak göğüs röntgeni görüntüleri transfer öğrenmeye dayalı sınıflandırma için önceden işlenip ve eğitilmiştir. 5247 göğüs röntgeni görüntüsünden 3906 görüntü zatürreden (pnömoni) etkilenen farklı deneklerden (bakteriyel pnömoni için 2561 görüntü ve

viral pnömoni için 1345 görüntü) ve 1341 görüntü normal deneklerden alınmıştır. Bu veri seti eğitim ve test setine bölünmüştür. 3 sınıflandırma şeması bildirmişlerdir: normal ve pnömoni, bakteriyel ve viral pnömoni ve normal, bakteriyel ve viral pnömoni. Üç sınıflandırma şeması için, DenseNet201'in hem eğitim hem de test için en yüksek doğruluğu vermiştir. Testin doğruluğu normal ve pnömoni sınıflaması için %98, normal, bakteriyel ve viral pnömoni sınıflaması için %93,3, bakteriyel ve viral pnömoni sınıflaması için %95 olarak bulmuşlardır. DenseNet201'in diğer algoritmalarından daha iyi performans gösterdiği açıktır. Veri büyütme ve Evrişimsel sinir ağları ile birlikte 5-kat çapraz doğrulama kullanılmıştır (14).

L. Brunesea ve ark. çalışmasında, 2 farklı veri setini birleştirerek 6.523 göğüs röntgenini incelemiştir. Yaklaşık 2.5 saniyede Covid-19'u tespit etmişlerdir. VGG16'dan yararlanarak transfer öğrenmeye dayalı bir değerlendirme yapmışlardır. Bunun yanında Grad-Cam algoritmasını da kullanmışlardır. 3 aşamalı bir yöntem önerilip, 2 Model kullanılmıştır. İlk modelde veri kümesi sağlıklı (3520) ve akciğer hastalıklı (3003) (Covid-19 dahil) olarak 2'ye ayırmışlardır. 2.modelde de akciğer hastalıklı (2753) ve Covid-19(250) olmak üzere 2'ye ayrılarak değerlendirilmiş. Sonuçta:1. modelde %96 doğruluk elde edilirken 2. model için %98 doğruluk elde edilmiştir(15).

N. Albert çalışmasında, Kaggle platformundaki 5910 görüntüden oluşan veri kümesini kullanmıştır. Kullanılan veri kümesi dengesiz olduğu için veri sayısı az olan görüntüleri Covid-19 etiketi altında değerlendirmiştir. 3 çalışmada ayrı değerlendirmelerde bulunmuştur. Genelde aktivasyon katmanı olarak da softmax yerine sigmoid kullanmıştır.

DeneyA'da pikselleri [0,1] arasında ölçeklendirerek Densenet121'in ve Resnet50'nin 2 versiyonunu ImageNet'in önceden eğitilmiş, ağırlıklarını kullanarak eğitmişlerdir. Ayrıca optimizasyon olarak tekrar(epoch)=30 olacak şekilde Adam+AmS Grad optimizasyonu ile kullanmışlardır.

Deney B'de normalde 36 eğitim(train), 10 doğrulama (validation), 12 test diye ayrılan Covid-19 veri kümesi Keras kütüphanesinin ImageDataGenerator'ü kullanılarak eğitim (train) ve doğrulama (validation) için 15'er artırılmış görüntü, test içinse 12 artırılmış görüntü kullanılmıştır. Deney A'dan farklı olarak Densenet121 ve Resnet50 modellerinin yalnızca 1 versiyonun eğitmişlerdir. Yine Adam+AmS Grad optimizasyonu ile kullanmışlardır.

Deney C’de ise ön işleme uygulanarak 30 ve 40 boyutlarında kırılarak tekrar 768×768 piksele uyarlanmıştır. Son olarak görüntüyü geliştirmek için Kontrastlı Sınırlı Uyarlamalı Histogram Eşitlemesi kullanmıştır. ResNet daha iyi sonuç vermiştir(16).

Y. Oh ve ark. çalışmasında, Röntgen (CXR) görüntülerindeki potansiyel biyobelirteçleri araştırmışlar ve Covid-19 için ayırıcı özellik olabileceğini düşünmüşlerdir. Bunun içinde 2 farklı veri kümesi kullanarak total veri kümesi 5 sınıfa ayırarak hesaplayan büyük olmayan veri kümeleriyle eğitilebilen yama tabanlı bir derin sinir ağı modeli önemiştir. 15.043 görüntüsü olan (Covid-19 dahil) veri kümesi eğitim (train), doğrulama (validation) ve test olmak üzere 3’e ayırmışlardır. Modeli eğittikten sonra, karar akciğerlerdeki rastgele konumlardaki birden fazla yamadan yapılan çoğunluk oylamasına dayandırılarak verilmiştir. Ayrıca, yerel yama tabanlı yaklaşıma uyarlanmış yeni bir olasılıksal Grad-Cam belirginlik haritası önemiştir. Sonuçlar SOTA yöntemiyle karşılaştırılmalı sonuçlar sağladığı ve radyolojik bulgularla güçlü bir şekilde ilişkili yorumlanabilir belirginlik haritaları oluşturduğunu göstermiştir (17).

S. VJ ve ark. çalışmasında, 4 farklı veri setini kullanarak CXR görüntülerinin normal, zatürre ve Covid-19 olacak şekilde sınıflandıran yeni bir evrişimsel sinir ağı modeli ve derin evrişimli üretken düşman ağları (DCGAN) önermektedir. Çalışılan modelde, önceden üzerinde çalışılmış (AlexNet ve GoogleNet)’den daha iyi sonuç veren 8 evrişimli katman, 4 Maksimum havuzlama ve 2 tam bağlı katman içermektedir. Kullandıkları bu yöntemde en iyi sonucu %98,6 doğruluk oranıyla CoronaHack veri kümesinden elde etmişlerdir. Çalışmalarında zaman harcanması, çok etiketli sınıflandırma gibi eksiklikler olduğundan evrişimsel sinir ağlarına (CNN’e) ek bir katman ekleyebileceklerini ve DCGAN’ları doğru bir tahminle veri kümesini büyötmek için uygun hale getirebileceklerinden bahsetmişlerdir(18).

F. Bozkurt, X-ray görüntülerini (Covid-19/normal/viral zatürre(pnömoni)) 3 sınıfa ayırmış olup, önceden kullanılan 11 farklı yöntem denemiştir. Buna göre en başarılı sonucun %97.17 ile DenseNet 121 modeli olduğunu görürken en az başarılı doğruluk değeri sıg-tam bağlı ağ modeli ile olduğunu incelemiştir(19).

F. Bozkurt ve ark. çalışmasında, 3 sınıflı veri kümesinde totalde 3886 X-ray görüntüsü bulunan veri setinde, son zamanlarda ünlenmiş VGG-19, Inception-V3 ve Resnet50 gibi modellerin yanında derin evrişimli sinir ağı modeli önerip

kullanmıştır. Bu derin evrişimli sinir ağı modelinden yaklaşık %96 oranında doğruluk elde edilmiştir(20).

A. Yılmaz çalışmasında, X-ray görüntülerinden oluşan 3 veri kümesini (dataseti) kullanılıp, birincisi de 50, 2. cisinde 1125, 3.cüsünde 5856 veri yer almaktadır. Bu veriler %80 eğitim %20 test olarak ayrılmış olup, evrişimli sinir ağları (CNN) ile birlikte tekrarlayan sinir ağları (SRN) mimarisi de aynı problem çözümü için kullanılmıştır. Ayrıca beş evrişim kanalına sahip olan çok kanallı evrişimsel sinir ağları mimarisi sunumu ve bu kanalların ürettiği sonuçlar arasında en belirgin özellik filtrelerini seçmek için önerilen çok kanallı evrişimsel sinir ağları mimarisinde kanal seçim formülü ortaya konmaktadır(21).

A. Narin ve ark. çalışmasında, X-Ray görüntülerinde Covid-19 hastalığının normalden farkı, Covid-19'un zatürreden farkı ve Covid-19'un diğer ikisinden farkı incelenmiş olup, en yüksek doğruluk değeri Resnet50 modelinden elde edilmiştir ve doğruluk değerleri %90'nın üzerinde bulunmuştur(22).

E. Khan ve ark. çalışmalarında, Covid-19'u ayırabilmek için, önceden eğitilmiş 3 farklı model (EfficientNetB1, NasNetMobile ve MobileNetV2) metotlar kullanılarak sonuçlar elde edilmiştir. Veri artırımı kullanılmış olup, 2 farklı eğitim stratejisi kullanılmıştır. Ayrıca hiper parametreler ince ayarlar ile düzenlenmiştir. En iyi sonuç, EfficientNetB1'de elde edilirken, önerilen teknik, Covid-19, viral zatürre(pnömoni), akciğer opaklığı ve normal içerikli 4 sınıflı %96.13 doğrulukla başarıyla sınıflandırılmaktadır(23).

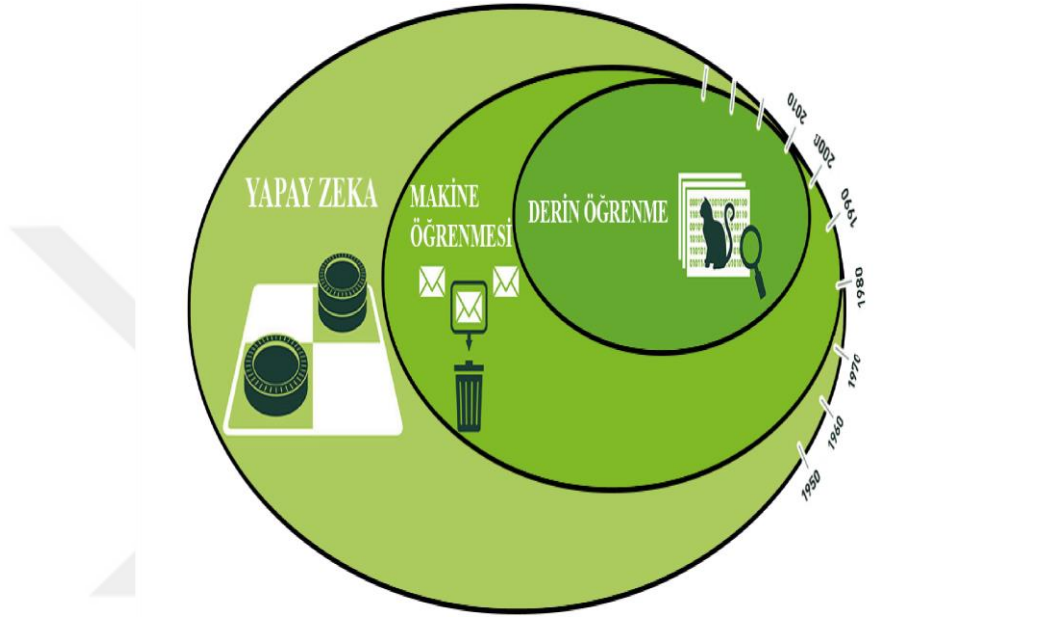
Z. Cao ve ark. çalışmalarında, 3 kategorili 219 Covid-19, 1300 zatürre(pnömoni), 1300 normal göğüs röntgenini ölçmek için BND-VGG-19 yöntemini kullanmışlardır. 2 GPU kullanılan deneyde 40 tekrar (epoch) için sadece 40 dakika kadar bir zaman harcanmıştır. DeneySEL sonuç olarak BND-VGG-19'da doğruluk değeri %95,48 iken BND-VGG-16 için doğruluk değeri %81,10'da kalmıştır(24).

2.2 Yapay Zekâ

Yapay zekâ, insanların her türlü davranışsal/düşünsel hareketlerinin bazılarının bilgisayara ve teknolojiye uyarlanması şeklinde yorumlanabilmektedir. Yapay zekâ günümüzde birçok alanda bizimle birlikte olmaya devam etmektedir. Günlük hayatta kullandığımız parasal işlemlerden, seslerden, görüntülerden,

sağlıksal alanlarda olmak üzere her an bizimledir. Teknolojinin bu derecede ilerlemesi yapay zekânın da daha ileri boyutlara ulaşacağını bir göstergesi olmuştur.

Yapay zekâ daha genel bir kavram olup, makine öğrenimini kapsamaktadır. Makine öğrenimi de derin öğrenmeyi kapsamaktadır.



Şekil 2.1 Yapay zekâ ve alt bölümleri

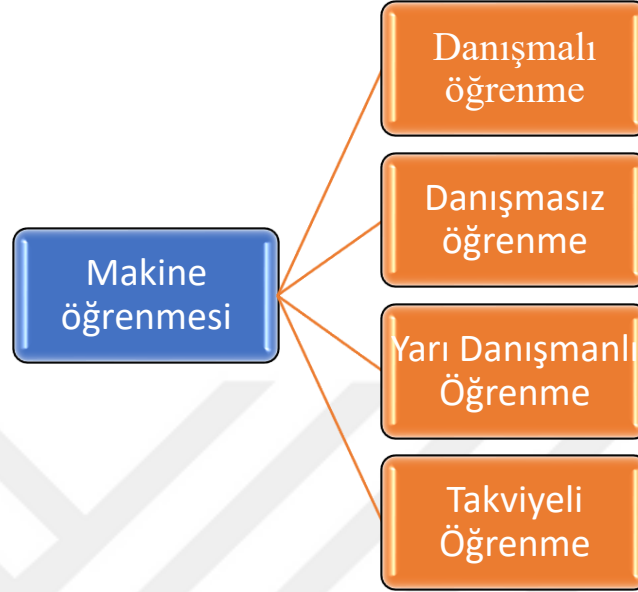
2.2.1 Makine Öğrenmesi ve Derin Öğrenme

Makine öğrenmesi çok büyük miktardaki verilerin işlenmesinin ve analizinin yapılmasının mümkün olduğunun az olmasıyla birlikte geçmişteki verileri kullanarak gelecekteki tahminleri yapmamızı sağlayan ve bu problemleri çözmeyi sağlayan alan olarak tanımlanabilmektedir. Bu sayede daha yeni modeller oluşturulmasına olanak sağlar(25).

Makine öğrenmesindeki uygulamaların adımları aşağıdaki gibi olmaktadır:

- 1) Problem verinin ne olduğu ve toplanması
- 2) Verilerin uygun bir şekilde hazır hale getirilmesi
- 3) Modelin eğitime tabi tutulması
- 4) Modelin değerlendirilmesi
- 5) En iyi sonuç için performansı artırmak

Makine öğrenmesi metotları 4 çeşittir. En çok kullanılan biri danışmanlı(denetimli) diğeri de danışmansız(denetimsiz) öğrenmedir(26).



Şekil 2.2 Makine öğrenmesi metotlarının sınıflandırılması(26)

Danışmanlı Öğrenme

Danışmanlı öğrenme, herhangi bir örneğe ait seçilen sınıf değerinin olumlu diğer kalanlarının olumsuz örnek olarak verildiği, önceden hiçbir sınıfa dahil olmamış yeni bir örneğin ise olumlu ya da olumsuz değerlendirilmesinde sınıf tanıyıcı olarak tanımlanabilen öğrenme çeşididir.

Danışmansız Öğrenme

Danışmansız öğrenmede, danışmanlı öğrenmenin aksine, girilen değer ile model arasındaki örüntüyü bulmaya amaçlar. Bu öğrenme çeşidinde çıkış değeri ile giriş değeri arasında eşleşme aranmaz.

Yarı Danışmanlı Öğrenme

Yarı Danışmanlı Öğrenmede, sistemin girilen değerlerindeki etiketlenmemiş veriler ve beraber az miktarda etiketlenmiş veriler girilmektedir.

Bu öğrenme çeşidinde etiketlenmiş verinin azlığında ile etiketlenmemiş veriyi elde etmenin basit olduğu durumlarda kullanılmaktadır.

Takviyeli (Pekiřtirmeli) Öğrenme

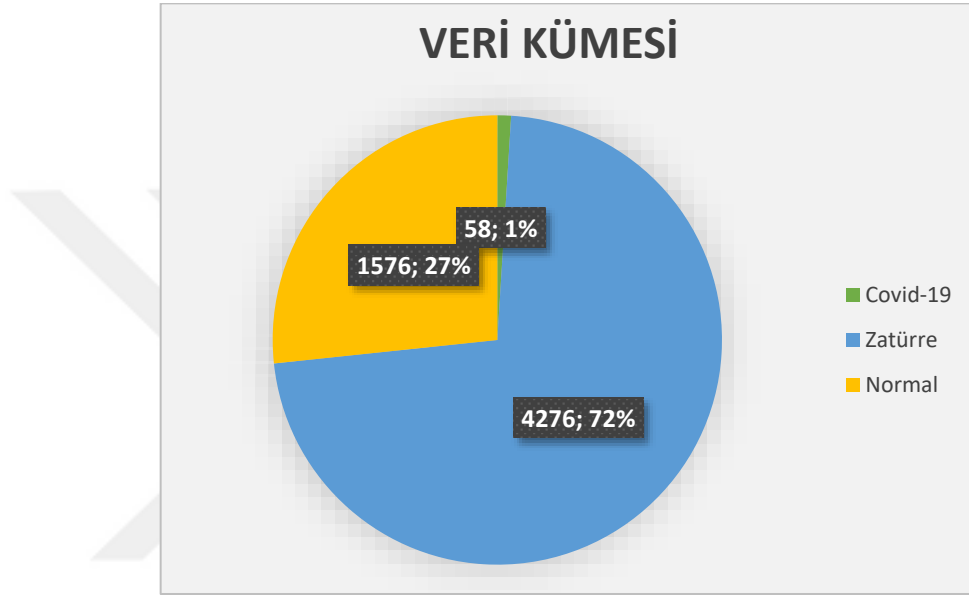
Takviyeli öğrenmede, sisteme verilen giriş değeriyle birlikte beklenen çıkış değeri ile hesaplanan çıkış değeri arasındaki farkı geri cevaplandırmasıyla bilinen öğrenme çeşididir.(26)



3. MATERYAL VE YÖNTEM

Bu tezde, araştırma Lenovo ThinkPad markalı AMD Ryzen 5 4500U birlikte Radeon Graphics 2.38 GHz işlemciye sahip, 8 GB Ram takılı olup, 64 bit işletim sistemine sahip bilgisayarda gerçekleştirilmiştir.

3.1 Kullanılan Veri kümesi (Veri Seti) Ve Kullanılan Kütüphaneler

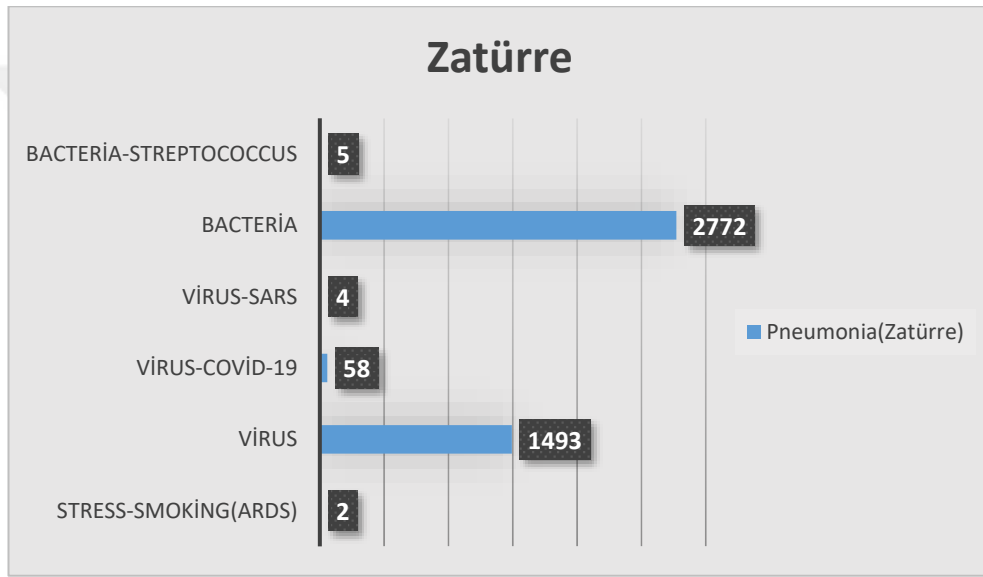


Şekil 3.1 Veri kümesi içerisindeki bireylerin yüzdelik dağılımı

Tablo 3.1 Veri setindeki görüntü sayısındaki dağılımı

Sınıf	Sınıf1 Kategorisi	Sınıf2 Kategorisi	Resim Sayısı
Normal			1576
Pneumonia	Stress-smoking	ARDS	2
Pneumonia	Virüs		1493
Pneumonia	Virüs	COVID-19	58
Pneumonia	Virüs	SARS	4
Pneumonia	Bacteria		2772
Pneumonia	Bacteria	Streptococcus	5

Kaggle veri platformunda bulunan CoronaHack-Chest-X-ray veri kümesinden alınan görüntülerde toplamda 5910 veri bulunmaktadır. Tabloda da gösterildiği üzere 1576 Normal bireylerin, 4334 Zatürre (pneumonia) hastalıklı bireylere ait olduğu görülmektedir. Bu veri kümesinde 58 Covid-19 verisi bulunmaktadır. Veri kümesinin veri dağılımına bakıldığında Covid-19 verisi, toplam veri kümesi verisinin yaklaşık %1'i kadar olduğu için bir dengesizlik yaratmaktadır. Bu nedenle de bu tez çalışmasında 2 ayrı sınıf şeklinde (Pneumonia ve Normal) olarak incelenecektir.



Şekil 3.2 Veri kümesi içerisinde zatürre bireylerin sayısal dağılımı

Kaggle veri platformunda bulunan CoronaHack-Chest-X-ray veri kümesinden alınan görüntülerde toplamda 5910 veri bulunmaktadır. Bu veriler işlenirken aşağıda bulunan kütüphaneler kullanılmıştır:

1. Verilerin hazırlanması ve tanınması için:
Numpy Kütüphanesi
Pandas Kütüphanesi
2. Veri Görselleştirilmesi için:
Matplotlib
3. Makine Öğrenmesi için:
Scikit-Learn

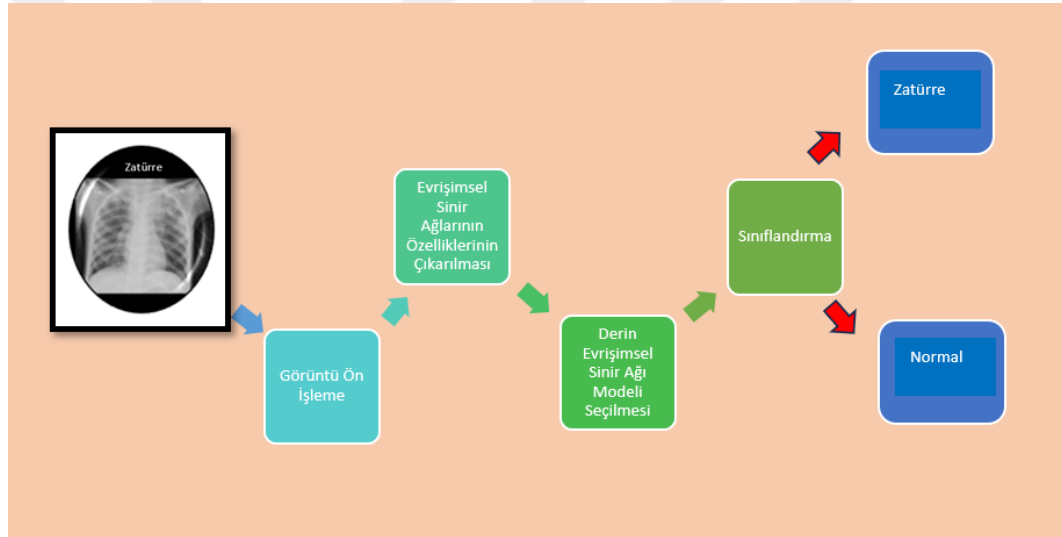
4. Derin Öğrenme için:

Keras

TensorFlow

3.2 Yöntem

Yöntem olarak normalde uygulanan derin öğrenme de bulunan evrişimsel sinir ağlarına ait mimariler kullanılmıştır. Önerilen yöntem görüntüsü 0-1 arasında normalizasyonuna gerek kalmayan görüntüler işlenmiştir. Veriler VGG-16, VGG-19, InceptionV3, Xception, MobileNet gibi mimariler uygulanmıştır.



Şekil 3.3 Veri kümesine uygulanan yöntem için işlem şeması

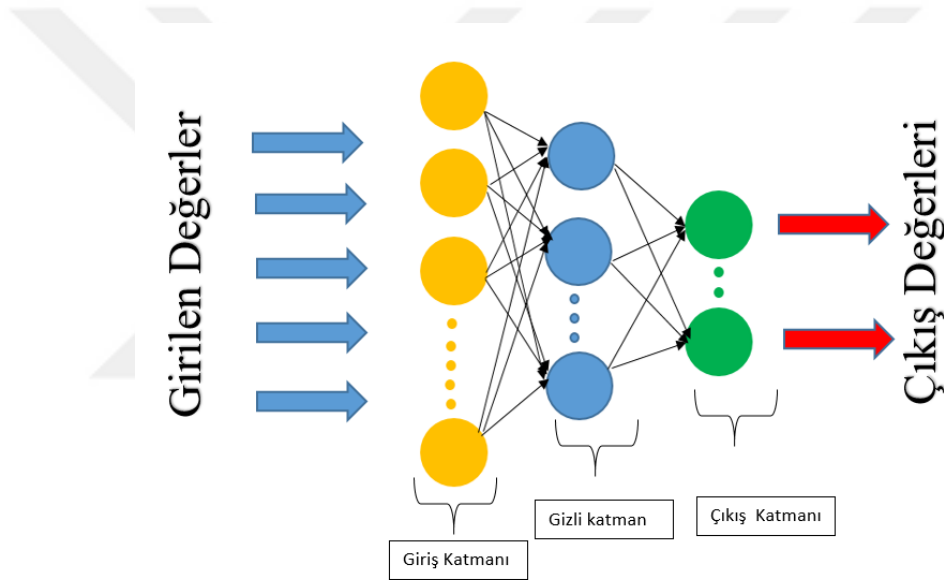
3.3 Derin Öğrenme Ve Yapay Sinir Ağları

Derin öğrenme, yapay zekânın bir yöntemi olup, makine öğreniminin bir alt dalıdır. Bu öğrenme önceki öğrenilenleri ya da girilenleri kullanarak sonuçlandırmasıyla ve ilişkilendirmesiyle bağlı olan sonucu göstermiştir.

Derin öğrenmedeki “derin” kelimesi yapay sinir ağındaki girdi ve çıktı katmanları arasındaki katmanların çok sayıda olmasından kaynaklanmaktadır. Genellikle derin öğrenme modelleri Evrişimsel Sinir Ağları’na (ESA) dayanmaktadır(27).

Derin öğrenme, makine öğrenmesiyle bağlantılı olup, her bir makinedeki geri yayılım algoritmasını kullanarak büyük verileri keşfetmeye yarayan bir öğrenmedir.

Derin öğrenme, çoklu işlem katmanından oluşan birden çok soyutlama düzeyine sahip verilerin temsillerinin öğrenmesine olanak sağlar ve bunu hesaplamak için parametrelerini değiştirerek, hesaplayabilir. Derin evrişimli ağlar sayesinde, görüntü, video, konuşma ve sesin işlenmesine neden olurken, tekrarlayan ağlar sayesinde de metin ve konuşma gibi verilere ışık tutmuştur(28).



Şekil 3.4 Derin Öğrenme Genel Mimarisi

3.3.1 Yapay Sinir Ağları

YSA (Yapay Sinir Ağları), mimari ve şekilsel insan beynini modelleyerek öğrenmeyi amaçlayan yapay zekâ yöntemlerinden biri olmuştur. İçerisindeki nöronlar yardımıyla bilgiyi kendi hafızasında barındırmakta ayrıca ağırlıklar sayesinde katmanlarda bulunan nöronlar birbiriyle bağlanmaktadır(29). YSA, birçok denemeye uygun olması, karar verme süreçleri bakımından faydalı olması sebebiyle günümüzde birçok alanda tercih edilmeye olanak sağlamaktadır.

Yapay sinir ağıları beynin bilgileri işleme sürecinde önemli rol oynayan nöronların birbiriyle olan etkileşimleriyle oluşturulan sonuç sürecine benzer şekilde üretilmeye çalışılmıştır. Bir nöron temel olarak şunlardan oluşmaktadır: Çekirdek, dendritler, hücre gövdesi, akson, akson uçları(sinaps).

Bir yapay nöron ise ağırlıklar, Sapma (bias), Toplama Fonksiyonu, Aktivasyon Fonksiyonu ve çıktılarından oluşmaktadır.

- Ağırlıklar

Bir a girilen değerinin, w_1 ağırlığıyla bağlandığında, girilen değer geldiğinde nörona bir ağırlık ile çarpılır. $a \times w_1$ olarak ifade edilir.

- Sapma (Bias)

Girilen değere uygulanan ağırlıktan farklı olarak ek olarak uygulanan doğrusal değere sapma (bias) denir. Ağırlıklar çarpılmış bir şekilde geldikten sonra ağırlık katsayı değerlerini belirlemek için kullanılmaktadır. Sapma değeriyle birlikte girilen değer $a \times w_1 + \text{bias}$ şeklinde ifade edilir.

- Toplama Fonksiyonu (Net Fonksiyon)

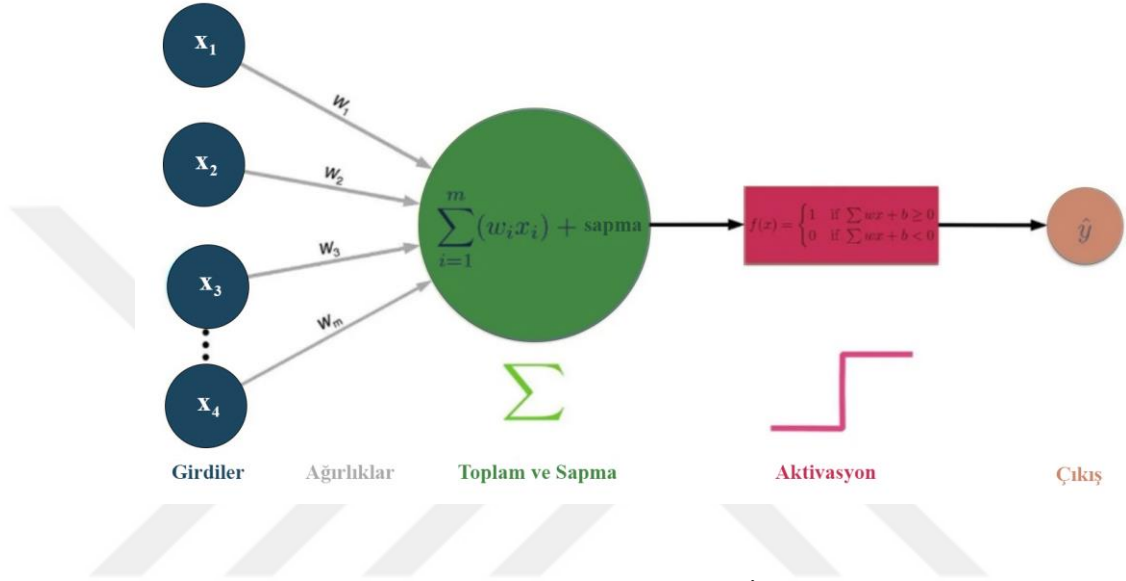
Toplama Fonksiyonu kendine gelen girilen değer en son sapma halinin eklenmesiyle Aktivasyon Fonksiyonuna hepsini toplayıp gönderen net girilenler toplamı fonksiyondur.

- Aktivasyon Fonksiyonu

Girilen değerlerin, çıkış değeri özelliğini kazanmasını sağlayan doğrusal bileşenli fonksiyondan doğrusal bileşenli olmayan fonksiyon çıktısı üreten fonksiyondur. Aktivasyon fonksiyonu sonucu: $f(a \times w_1 + b)$ olarak ifade edilir. Çoklu girdi değerleri olduğu için ve bu değerlere karşılık ayrı ağırlıklar denk geldiğinden dolayı $u = \sum w * x + b$ şeklinde ifade edilir. Aktivasyon fonksiyonu $f(u)$ oluşur ve $y = f(u)$ olarak ifade edilir. En yaygın kullanılanlar: sigmoid, ReLU ve Softmax fonksiyonlarıdır.

- Çıktılar

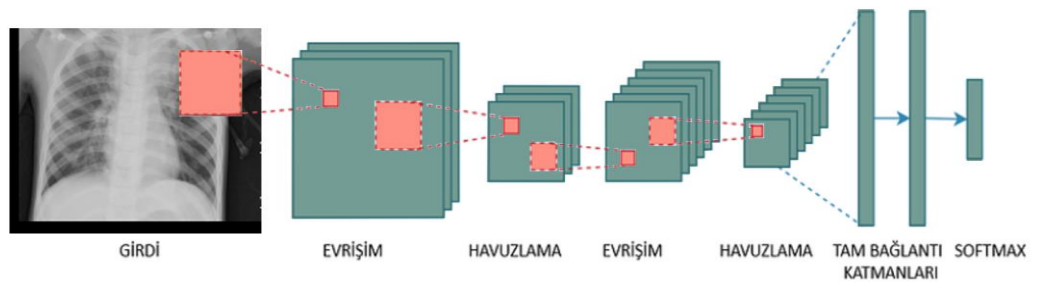
Aktivasyon katmanından sonra önceden girilen değere göre sonucu oluşturulan yerdir.



Şekil 3.5 Yapay Sinir Ağı Genel İşlevi

3.4 Evrişimsel Sinir Ağları (CNN) Ve Katmanları

Evrişimsel sinir ağları ileri yönlü sinir ağları olarak bilinse de az parametreye ihtiyaç duyması, kullanım şeklinin kolay oluşu sebebiyle evrişimsel sinir ağları (CNN'ler) tercih edilmektedir. Genelde 4-5 katmandan oluşsalar da 5 katmandan daha fazla da eklenebilme özelliğine sahiptirler.



Şekil 3.6 Evrişimsel sinir ağları genel yapısı

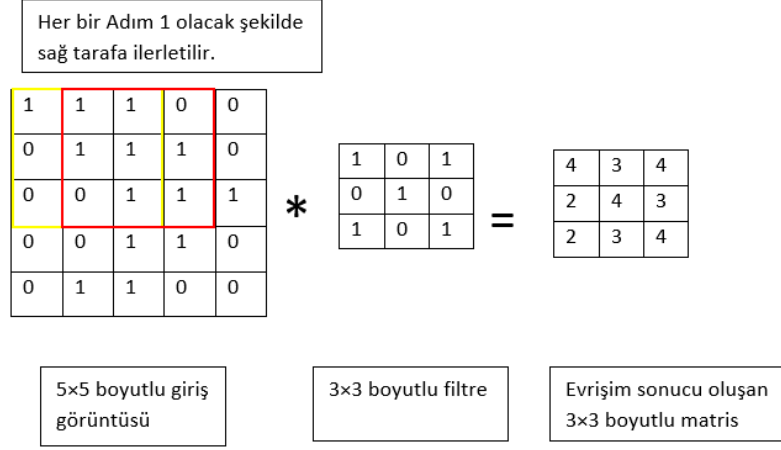
3.4.1 Giriş Katmanı

Giriş katmanı genelde $224 \times 224 \times 3$ şeklinde boyutlu matrisinden oluşmaktadır. Eğer 3 boyutlu olacak ise geçerlidir.

3.4.2 Evrişim Katmanı

CNN'de görüntüyü ele alan ilk katmandır. Evrişim sağlayarak diğer katmanlarla bağlantı oluşturur.

Bu katmanda evrişim için filtreler(kernel) farklı boyutlarda $2 \times 2, 3 \times 3$, vb gibi boyutlarda ağırlık matrisleri üzerinde işleme tabi tutulur. Bu sayede hem eğitim hemde filtrelerde öğrenme meydana gelmektedir. Filtreleme işlemi sol en üst köşesinden başlayarak, filtrenin her bir adımı sağa doğru kaydırılarak işlem gerçekleştirilir. Filtre ağırlık matrisi ve girdi matrisinin filtre ile birbiriyle örtüşen kısımdaki değerler çarpılarak yeni bir matris elde edilir (Hadamard Çarpımı). Görüntüde en sağa gelindiğinde bir basamak aşağıda en sol tarafa kaydırma işlemine sayesinde çarpım işlemine devam edilir. Daha sonra çarpılan değerler toplanarak matris doldurulur. Kaydırma(stride) genelde 1 olarak kabul görsede, tasarlayan kişiye göre bu değişmektedir. Bazende farklı uyumsuzluklar sonucunda görüntünün dış kısmı doldurulmaktadır. Yeni piksel ekleme(padding) denilen bu işlemde eğer '0' seçilirse sıfır dolgusu denilerek çıkış boyutunu değiştirmeden boyutunu kontrol edilebilmesine olanak sağlamaktadır.



Şekil 3.7 5×5 Boyutlu girilen bir matrisin 3×3 boyutlu filtreye evrişimi

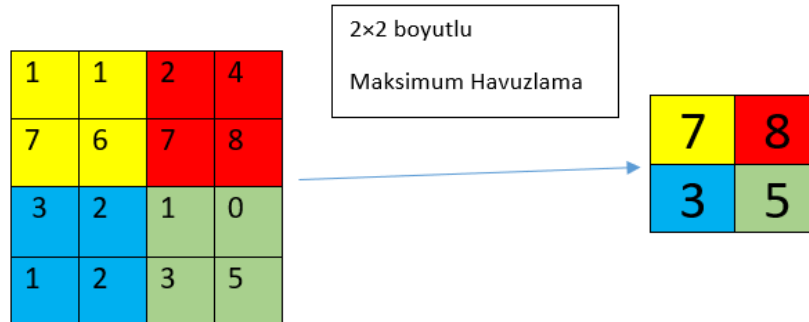
3.4.3 Havuzlama Katmanı

Havuzlama katmanında değişik boyutlu filtreler kullanılabilir. Bunlar 2×2 boyutlu olabileceği gibi 3×3 boyutlu da olabilmektedir. En yaygın olanları ortalama ve maksimum havuzlama katmanlarıdır(30).

Ortaklama veya Havuzlama katmanı olarak bilinen bu katmanda hesaplama yükünü ve temel amaç olan parametre sayısını azaltmak için işlem kolaylığı sağlamaktır.

- Maksimum Havuzlama

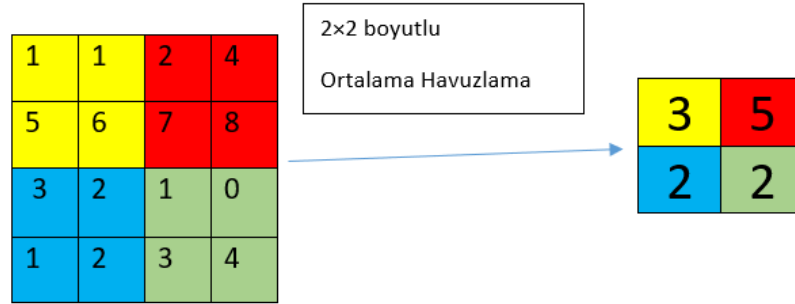
Bu Havuzlama türünde, evrişim işleminden sonra gerçekleşen katman olup, filtre işlemiyle birlikte matrisin kesiştiği piksel noktaları arasındaki maksimum değer seçilerek işlem yapılır.



Şekil 3.8 4×4 Boyutlu girilen bir matrisin maksimum havuzlama işlemi

- Ortalama Havuzlama

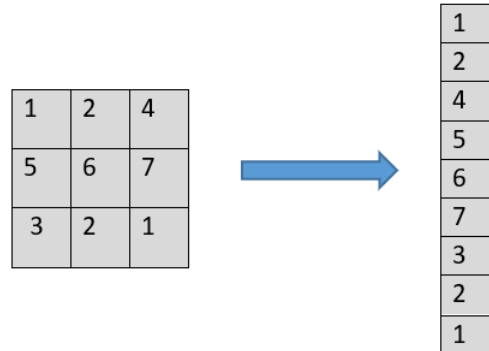
Bu katmanda, ortalama havuzlama olduđu takdirde filtre matrisin diğeri matrisle kesiřtiđi yerlerdeki piksel deęerlerinin ortalaması alınarak boyutu azaltılan matrise dñnüştüđü yerdir.



Şekil 3.9 4×4 Boyutlu girilen bir matrisin ortalama havuzlama işlemi

3.4.4 Düzleştirme

Havuzlama ve evriřim katmanlarından gelen matrislerin verilerini sıkıştırarak kullanmaktadır. Daha sonra bu sinir ađının girdisi řeklinde olacaktır(31). Bir matrisin skaler bir vektöre dñnüştüđü yer olarak da söylenebilir.



Şekil 3.10 3×3 Boyutlu girilen bir matrisin düzleştirme örneđi řekli

3.4.5 Tam Bađlantı Katmanı

Önceden elde edilen verilerin tek bir matrise dñnüşürüldüđü yer olarak tanımlanmaktadır. Sınıflandırma yaparken derin özellikler bu katmanda belirlenmektedir(27). Diğeri sinir ađlarına giriş olarak verilen katmandır.

3.4.6 Bırakma Katmanı

Bu katmanda amaç, yapay sinir ağının aşırı uyumuna karşı geliştirilmiştir. Bu yüzden de bazı nöronlar atılarak işlevini yitirmektedir. Atılan bu nöronlar döngüye katılamayarak ağın geri yayılmaya katılmasına engel olmaktadır. Özellikle eğitim verilerinde karmaşık birlikte uyarlamaları önlemek için ‘bırakma’ kullanılarak modelde ezberleme (overfitting) durumunu azaltabilir.

Yalnız evrişimli katmanlarda, sinir ağındaki parametre sayısında ciddi bir azalma olduğundan, evrişimli katmanlarda ‘bırakma’ çok daha az avantajlıdır(32).

3.4.7 Sınıflandırma Katmanı

Son katman olarak bilinmektedir. Genelde softmax/sigmoid aktivasyon fonksiyonundan sonra gelmektedir.

3.4.8 Aktivasyon Katmanı

3.4.8.1 Birim Basamak Fonksiyonu

Bu fonksiyon 0’dan önceki değerler için türevlenemez olduğu için kabul görsede 0’dan sonraki her birim için 1 sabit değerini aldığı için bu fonksiyon birim basamak fonksiyonu olarak adlandırılmaktadır.

3.4.8.2 Doğrusal Fonksiyon

Her bir değer için sabit oranlı sayılar aldığından ve türevi sabit olduğu durumundan dolayı çoklu katmanlara ait çalışmada kullanılamaz.

3.4.8.3 ReLU Fonksiyonu

Doğrusal (Linear) fonksiyonuyla aynı gibi görünmesine karşı $[0, +\infty]$ değerler almaktadır. Türevlenebiliyor olabilmesine karşın 0’da türevinin 0 oluşu başka fonksiyonlara yöneltilmesine sebep olmuştur.

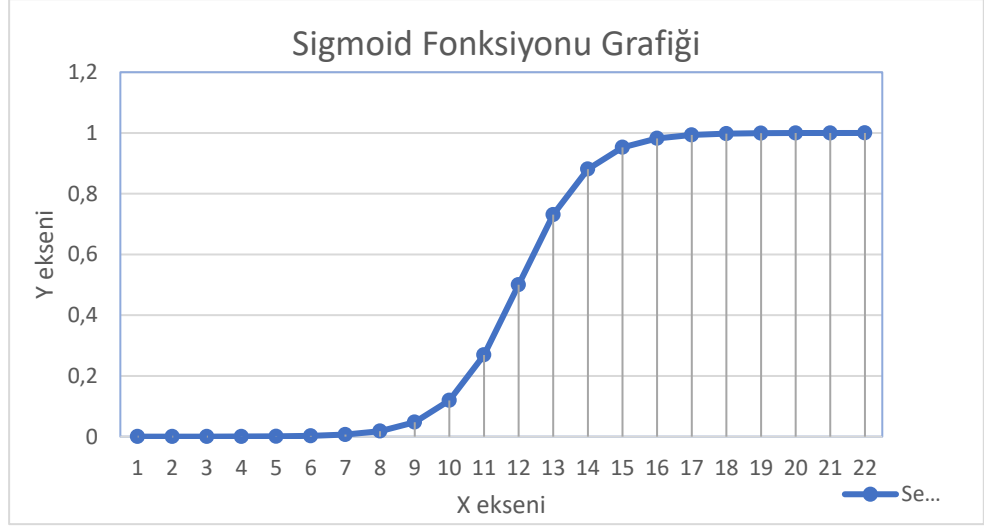
3.4.8.4 Sızıntı ReLU Fonksiyonu

Genelde, sızıntı değerinin 0,01 olarak verilmişken, başka değerlerle de değiştirilirse ismi değiştirilmektedir. ReLU’dan farklı olarak ReLU’nun türevlenemez olduğu noktaları ya da bölgeyi türevlenebilir hale getirebilmesiyle öne çıkmaktadır.

3.4.8.5 Sigmoid Fonksiyonu

Sigmoid, genelde $[0-1]$ aralığında değerler üretmektedir. Softmax’dan farklı olarak olasılık dağılımının olmayışı ve veri boyutunun Softmax’a göre bir boyut daha düşük olmasıyla bilinmektedir. Ayrıca Sigmoid fonksiyonun sınıflayıcı olarak iyi olmasına rağmen uçlarda türevlenebilir olmasının az olması sebebiyle çok sınıflı problemde tercih edilmemektedir.

$$\text{Sigmoid}(x) = \frac{1}{(1+e^{-x})}$$



Şekil 3.11 Aktivasyon fonksiyonu olan Sigmoid'in gösterimi

3.4.8.6 Hiperbolik Tanjant Fonksiyonu

Sigmoid fonksiyonuna benzerliğiyle ön plana çıksa da daha dik yerlerde türevlenebiliyor oluşu sebebiyle sigmoid fonksiyonunu geçmektedir.

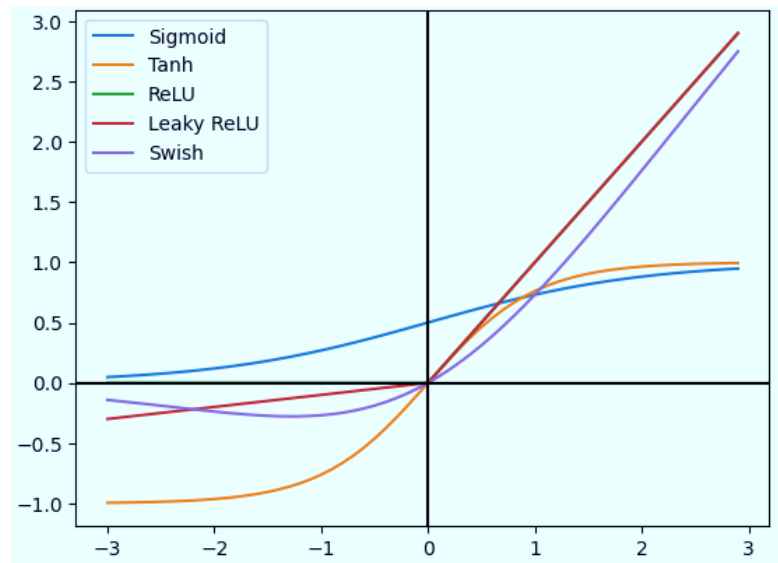
3.4.8.7 Swish Fonksiyonu (Kendinden Geçitli)

Swish fonksiyonu, Sızıntı ReLU (Leakly ReLU)'ya benzese de negatif değerler bölümünde türevlenebilir doğrusal değerler aldığı için daha kullanışlıdır.

3.4.8.8 Softmax Fonksiyonu

Her bir çıkış sonucuna (bir fonksiyona girişine karşı ürettiği cevap) karşı olasılık değeri üretir ve ürettiği bu olasılık değerlerinin toplamı her zaman 1'e eşittir. Bunun içinde sınıflandırma da en çok tercih edilenler arasındadır.

Çoklu sınıflandırma en çok kullanılan fonksiyondur.(33)



Şekil 3.12 Aktivasyon fonksiyonlarının grafikte toplu biçimde görünüşü

Doğrusal fonksiyonu	$f(x) = x$	$(-\infty, \infty)$
Basamak Fonksiyonu	$f(x) = \begin{cases} 0, & x < 0 \\ 1, & x \geq 0 \end{cases}$	$\{0,1\}$
Sigmoid fonksiyonu	$f(x) = \sigma(x) = \frac{1}{1 + e^{-x}}$	$(0,1)$
Hiperbolik Tanjant Fonksiyonu	$f(x) = \tanh(x) = \frac{(e^x - e^{-x})}{(e^x + e^{-x})}$	$(-1,1)$
RELU	$f(x) = \begin{cases} -x, & x < 0 \\ x, & x \geq 0 \end{cases}$	$[0, \infty]$
Leakly(Sızıntı)RELU	$f(x) = \begin{cases} 0.01, & x < 0 \\ x, & x \geq 0 \end{cases}$	$(-\infty, \infty)$
Swish Fonksiyonu	$f(x) = 2x\sigma(\beta x) = \begin{cases} \beta = 0 \text{ için } f(x) = x \\ \beta \rightarrow \infty \text{ için } f(x) = 2\max(0, x) \end{cases}$	$(-\infty, \infty)$

Şekil 3.13 Aktivasyon fonksiyonlarının genel gösterim şekilleri

3.4.9 Normalizasyon Katmanı

Genel bir ölçülendirme sağlayarak normalize edilmesi sağlanmaktadır. Normalleştirilmenin temel amacı sayısal değerler için, tek tip bir ölçek sağlamaktır.

3.4.10 Kayıp Fonksiyonu

Evrişimsel Sinir Ağlarında kayıp fonksiyonu (loss function) tahmin eden değer ile gerçek değer arasındaki farkı temel olarak kullanılan bir fonksiyon çeşididir.

3.5 Kullanılan Derin Öğrenme Modelleri

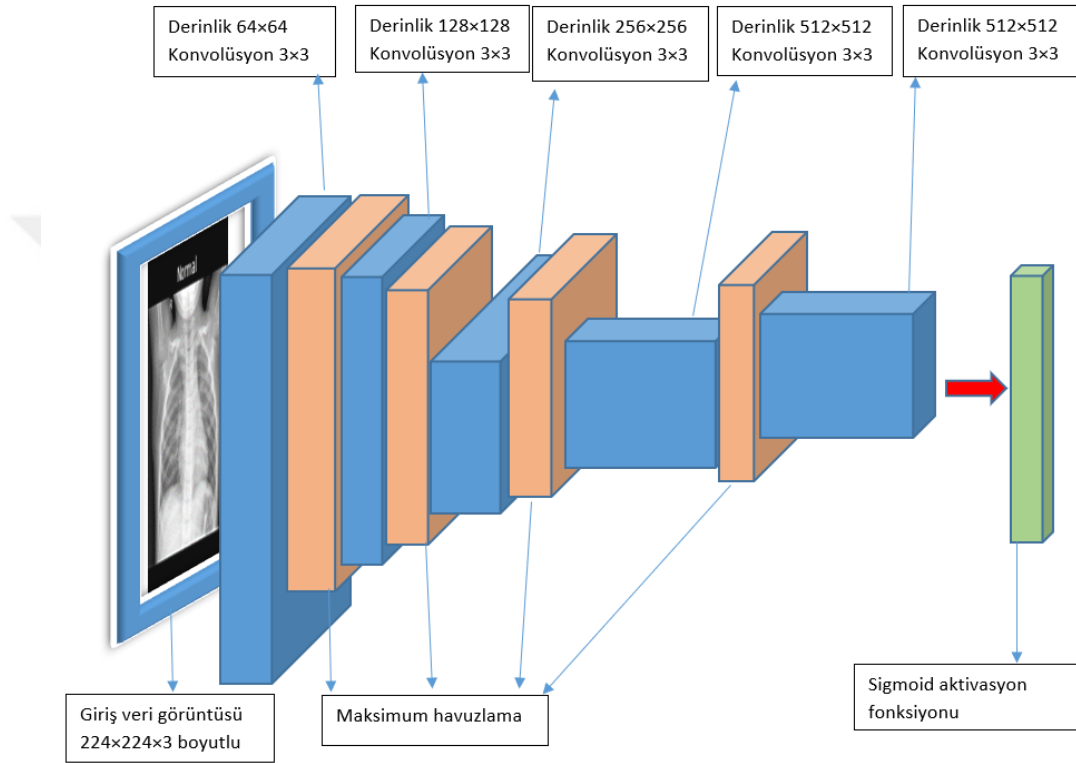
3.5.1 VGG-16

VGG-16, ILSVRC (Büyük Ölçekli Görsel Tanıma) 2014 yarışmasında ikinci olarak dereceye giren evrişimsel sinir ağından oluşan bir ağ modeli sunmuşlardır. LeCun ve arkadaşlarının klasik CovNet'inden ayrılmayarak derinliği daha artırmışlardır(28).

VGG-16, 224×224 RGB sabit boyutlu giriş görüntüsünü kullanmaktadır. Görüntüyü bir evrişim yığınının geçirilir ve 3×3 boyutlu filtreler kullanılır. Konfigürasyonların birinde de 1×1 evrişim filtreleri de kullanılmaktadır. Evrişim adımı 1 piksele sabitlenir fakat dolgu için 3×3 evrişimli katmanlar için 1 pikseldir.

Uzamsal Havuzlama, takip eden 5 maksimum havuzlama katmanı tarafından gerçekleştirilmektedir.

Maksimum havuzlama ise 2 adımla 2×2 piksellik bir pencere üzerinde yapılır. Katman yığınlarını 3 tam bağlı katman izler, ilk 2'sinin her bir 4096 kanala sahip olup 3'sü 1000 kanal içerir ve en son katman Softmax katmanıdır. Tüm gizli katmanlar doğrusal olamayan ReLU'lar ile donatılmıştır ve ağlarda (biri hariç) Yerel Yanıt Normalleştirme kullanılmamıştır. VGG-16'da %89 doğruluğu yakalamışlardır(34).



Şekil 3.14 VGG-16 mimarisine sigmoid fonksiyonu eklenmiş görüntüsü

Bu tezde, toplamda yaklaşık 138 milyon parametresi bulunan VGG-16 mimarisi sonuna aktivasyon fonksiyonu 'sigmoid' seçilmesi parametre sayısını yaklaşık 136 milyona denk gelmiştir.

3.5.2 VGG-19

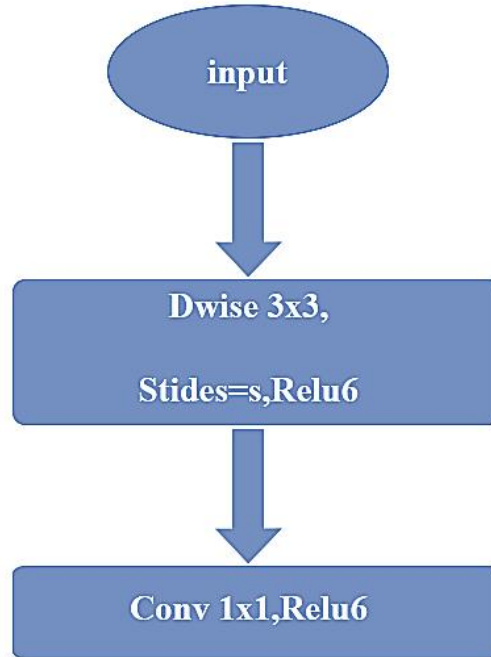
VGG-16'ya benzer olarak bulunan ve 19 katmana kadar eklenerek yapılmıştır. Toplamda 144 milyon parametresi bulunan VGG-19, maksimum havuzlama katmanına ayrıca eklenmesiyle elde edilmiştir(34).

3.5.3 MobileNet/MobileNetV1/MobileNetV2

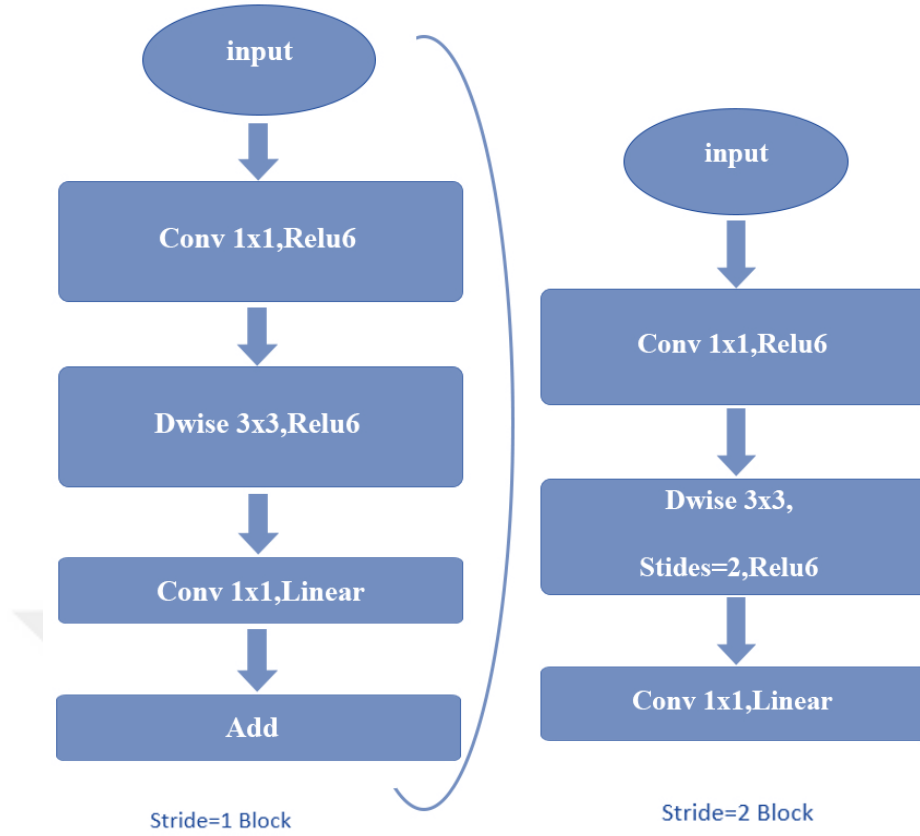
Mobil ve kaynak kısıtlı ortamlar için özel olarak tasarlanmış yeni bir mimaridir.

Bu mimaride, ihtiyaç duyulan işlem ve bellek sayısı önemli ölçüde azaltarak, aynı doğrulukla, mobil uyarlanmış bilgisayar görme modelleri için en son teknolojiyi zorlamaktadır. Yeni bir katman modülü olarakda, doğrusal dar boğazlı ters çevrilmiş artığa ya da kalıntıya dayanmaktadır. Ağ tasarımı MobileNetV1'e dayandırılmaktadır(35). MobileNetV2'de 32 filtreli ilk tam evrişim katmanı ve ardından 19 artık darboğaz katmanından oluşur. Düşük hassasiyetli hesaplama ile kullanıldığında sağlamlığı nedeniyle doğrusal olmayanlık olarak da ReLU'lar kullanılmaktadır.

Modern ağlar için standart olduğu gibi her zaman 3×3 'lük çekirdek boyutunu ve eğitim sırasında 'bırakma' ve 'toplu normalleştirme'yi kullanmışlardır.(36)



Şekil 3.15 MobileNet evrişim blok gösterimi

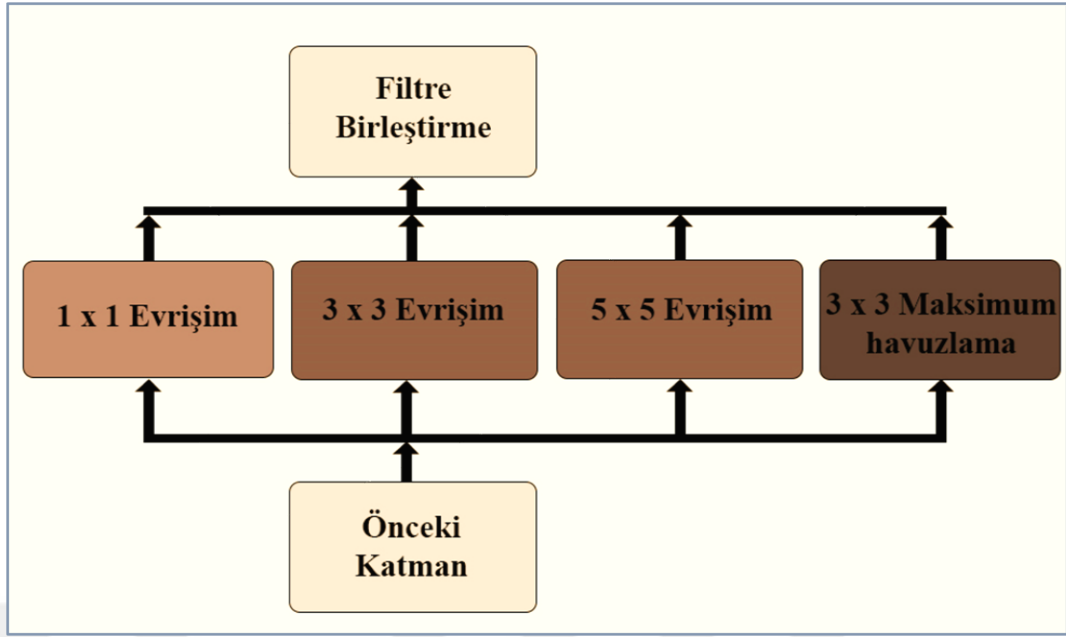


Şekil 3.16 MobileNetV2 evrişim blok gösterimi

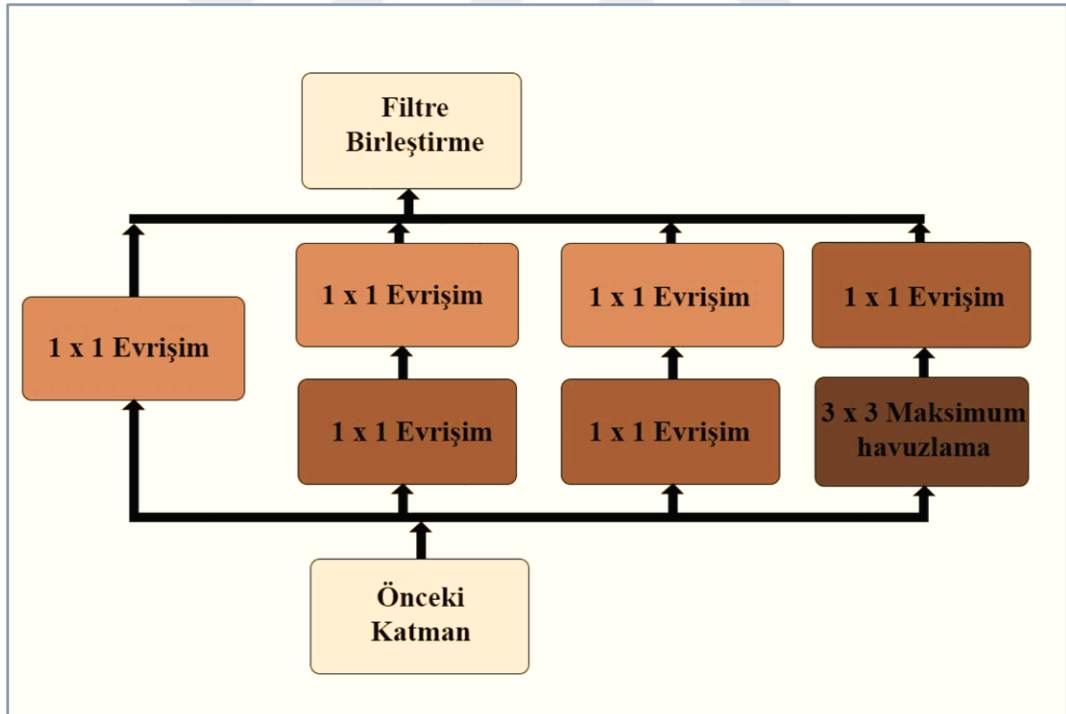
3.5.4 GoogleNet (InceptionV3)

Google tarafından ilk olarak ILSVRC yarışmasında birinci olan ve ilk Inception modülü sadece 4 milyon parametre ve %6.67 en yüksek 5 hata oranıyla bulunmuştur. İlk Inception modelinde derinliği ve genişliği artırmak için hazırlanmış olup, toplamda 9 Inception bloğu ve 100'ü aşan katman bulundurmaktadır(37).

Ayrıca ilk Inception modelleri, 1×1 , 3×3 , 5×5 filtre ve 3×3 maksimum ortaklama katmanlarından oluşmaktadır. Bu yapıya naif Inception modülü denir(5). InceptionV3 mimarisi 11 Inception modülünden oluşmakta olup her modelinde havuzlama, evrişim ve aktivasyon fonksiyonları ReLU'lar kullanılmaktadır(4). InceptionV3 mimarisinde, tüm bu katmanların, bir sonraki aşamanın girişini belirleyen çıkış vektörüne birleştirilmiş çıkış filtresinin kombinasyonu ile belirlenir(37).



Şekil 3.17 Naif Inception modülü gösterimi



Şekil 3.18 Boyut indirgemeli InceptionV3 modülü

3.5.5 Xception

Xception mimarisi, InceptionV3 mimarisine benzer ve yaklaşık aynı parametrelere sahip olmasına karşın performans kazanımlarının artan kapasiteden değil, model parametrelerinin daha verimli bir şekilde kullanılmasındandır. Inception V3'ün aşırı modeli örnek alındığı için aşırı başlangıç anlamına gelen Xception adı verilmiştir. Xception mimarisi 36 evrişimli katmana sahiptir. 36 evrişimli katman, ilk ve son modüller haricinde tümü etraflarında doğrusal artık bağlantılara sahip, 14 modül halinde yapılandırılmıştır. Ayrıca evrişimsel tabanı bir lojistik regresyon katmanı izlemektedir. İsteğe bağlı olarak da lojistik regresyon katmanından önce tam bağlı katmanlarda eklenebilir. JFT ve ImageNet veri kümelerinde başarılar yakalayan Xception, ImageNet üzerinde en yüksek-1 doğruluğuna 0,790 ve en yüksek-5 doğruluğuna 0,945'e ulaşmışlardır(38).

3.6 Veri Artırımı

Genelde yapay sinir ağlarının işlenmesinden başarılı sonuçlar elde etmek için çok iyi eğitilmesi gerekmektedir(39). Veri artırımı genelde yeterli görüntünün bulunmadığı veri setlerinde bir kullanılan yöntemdir. Veri artırımındaki kullanmamızın asıl amacı düzenlileştirme (regülerization) sağlamaktır. Bu sayede ezberleme (overfitting) olarak karşımıza çıkan veri tekrar öğrenmesinin üstesinden gelinmiş olmaktadır.

Veri artırımında birçok yol izlenmektedir. Bunlardan başlıcaları: Döndürme, yakınlaştırma, uzaklaştırma, bulanıklığı azaltma, çevirme, kaydırma, parlaklığı ayarlama, yeniden boyutlandırma gibi seçeneklerdir. Bu tezde yeniden boyutlandırma, kırpma, döndürme, yakınlaştırma gibi veri artırım yöntemleri kullanılmıştır.

3.7 Sinir Ağının Eğitilmesi

Eğitim için bu veri seti seçilmiştir. Toplamda 20 ve üzeri tekrarlarda çalışılmıştır. İyileştirme olması için 'Adam' Algoritması seçilmiş olup, eğitim katsayısı 0.0001 olarak kullanılmıştır.

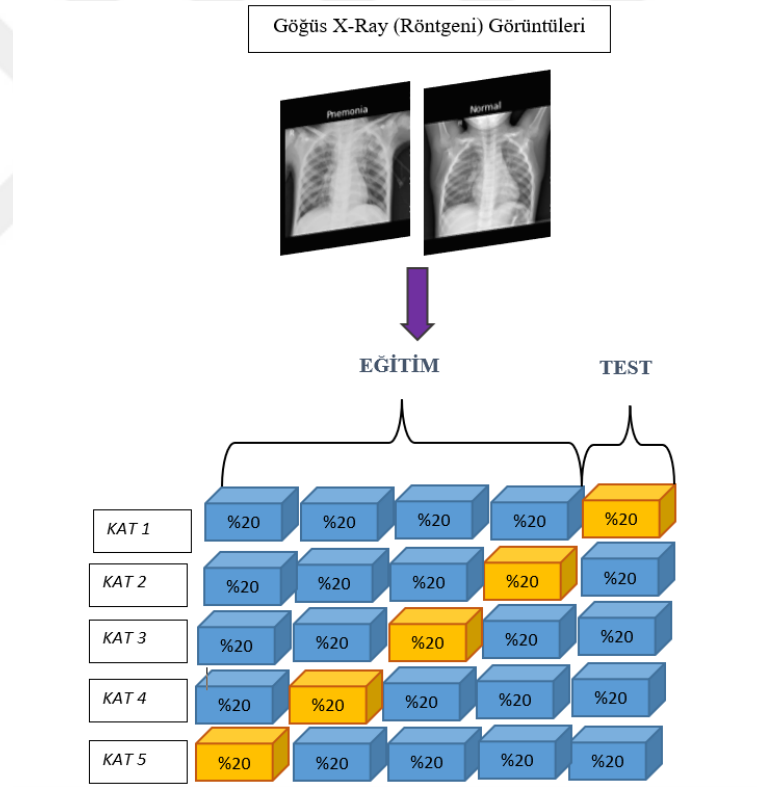
Ağ çıkışında ise 2 sınıftan birine ait olan sınıflandırma problemlerinde kullanılan ikili çapraz entropisi (binary cross entropy) uygulanmıştır.

3.8 Sınıflandırıcı Performans Ölçütleri

Sınıflandırıcı performans ölçütleri, sınıflandırma yapılması için değerlendirilmek üzere bazı kriterlerden oluşmaktadır.

3.8.1 Çapraz Doğrulama

Çapraz doğrulama, sınıflandırmanın güvenliğini ve etkinliğini artırmak için kullanılan bir yöntemdir. Tüm veri kümesini belirli sayıda alt kümelere ayrılmasıyla başlamaktadır. Her bir parametre 'k' ile gösterilerek her bir ayrılan set için işlem yapılır. Örneğin, $k=5$ seçilirse her işlemde 1. alt küme test olarak ayrılırken kalan 4 küme eğitim kümesi olarak kullanılır. Bu işlem 5 defa tekrarlanır. Sonuçta, elde edilen test kümelerine ait sonuçların ortalaması alınarak işlem tamamlanmış olur.



Şekil 3.19 K=5 değeri için çapraz doğrulama şekli

3.8.2 Karmaşıklık Matrisi

Bu yöntem makine öğrenmesindeki algoritmaların performanslarını ölçmek için bir ölçüt olarak genelde kullanılan bir yöntemdir.

Sonuçları değerlendirmek için Doğruluk, hatırlama, kesinlik, F-1 ölçütü, duyarlılık, özgünlük, Cohen Kappa gibi kriterler değerlendirmeye alınmıştır.

Tablo 3.2 Karmaşıklık Matrisi

	Tahmini Pozitif Sınıf	Tahmini Negatif Sınıf
Gerçek Pozitif Sınıf	TP	FN
Gerçek Negatif Sınıf	FP	TN

- TP (Doğru Kabul-True Positive):
Modelin tahmini pozitif iken, gerçekte de pozitif olan durumdur.
- FN (Yanlış Red-False Negative):
Modelin tahmini negatif iken, gerçekte de pozitif olan durumdur.
- FP (Yanlış Kabul-False Positive):
Modelin tahmini pozitif iken, gerçekte de negatif olan durumdur.
- TN (Doğru Kabul-True Positive):
Modelin tahmini negatif iken, gerçekten de negatif olan durumdur.

Doğruluk (Accuracy) veri kümesindeki doğru tahmin örnek sayısının veri kümesindeki tüm örnek sayısına oranı şeklindedir.

$$\text{Doğruluk (Accuracy)} = \frac{(TP+TN)}{(TP+TN+FP+FN)} \quad \text{Denklem (3.8.2.1)}$$

Kesinlik (Precision) sistemin doğru olarak tahmin ettiği pozitif örnek sayısının, doğru olarak tahmin edilen pozitif örnek sayısı ile yanlış olarak pozitif sınıflandırıldığı örnek sayısına bölümüdür.

$$Kesinlik(Precision) = \frac{TP}{(TP+FP)} \quad \text{Denklem (3.8.2.2)}$$

Hatırlama (Recall) sistemin doğru olarak tahmin ettiği pozitif örnek sayısının, doğru olarak tahmin edilen pozitif örnek sayısı ile yanlış olarak pozitif sınıflandırıldığı örnek sayısına oranıdır.

$$Hatırlama(Recall) = \frac{TP}{(TP+FN)} \quad \text{Denklem (3.8.2.3)}$$

F-1 ölçütü, kesinlik (precision) ve hatırlama (recall) arasındaki dengeye göre ayarlanmaktadır. Kullanılan yöntemde harmonik ortalama yardımcı olmaktadır.

$$F1 \text{ ölçütü} = \frac{2*Hatırlama*Kesinlik}{Kesinlik+Hatırlama} \quad \text{Denklem (3.8.2.4)}$$

F1 skoru, aralarında en güvenilir sonuçları gösterendir. Her bir veri seti için bu kriterler değişse de genelde doğruluk ve F-1 ölçütü değerlendirilirken bazılarında ise hassasiyet ve hatırlama kriterleri göz önüne alınarak değerlendirilebilir.

Duyarlılık (Sensitivity) değeri gerçek pozitif değerlerin, gerçek pozitif değerler ile yanlış negatif durumların oranlanmasıyla bulunmaktadır.

$$Duyarlılık(Sensitivity) = \frac{TP}{(TP+FN)} \quad \text{Denklem (3.8.2.5)}$$

Özgünlük (Specifity) gerçek negatif değerlerin, gerçek negatif değerler ile yanlış pozitif durumların toplamına bölünmesiyle bulunmaktadır.

$$Özgünlük(Specifity) = \frac{TN}{(TN+FP)} \quad \text{Denklem (3.8.2.6)}$$

Cohen Kappa Katsayısı (Cohen Kappa's Score) iki gözlemci arasındaki değişimi ya da farkı hesaplayarak ölçen bir uyum yöntemidir.

Cohen Kappa Katsayısı, -1 ile 1 arasında değerler alırken -1 olduğunda sonucun birbirleriyle uyumsuz olduğu, 0' da ise uyumun şansa bağlı bir durum iken +1' de ise birbirleriyle aşırı uyumlu olduğunu ifade etmektedir(40).

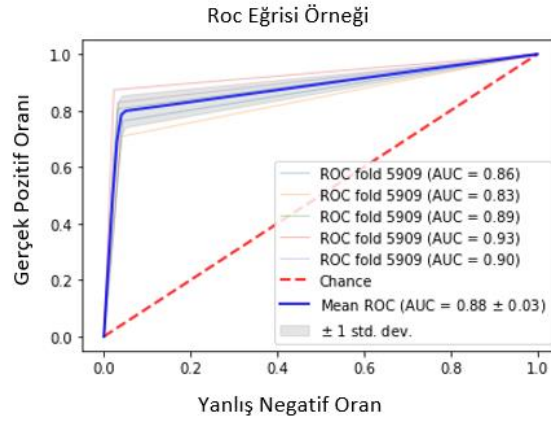
Kappa Katsayısı bulunurken θ_1 , iki değerlendirici için gözlenen uyumların toplam oranıyken, θ_2 , bu uyumunun şansa bağlı çıkma olasılığının belirtilmesidir. Aşağıda bulunan denklem formülü ile hesaplanmaktadır.

$$k = \frac{\theta_1 - \theta_2}{1 - \theta_2} \quad \text{Denklem (3.8.2.7)}$$

$$\theta_1 = \frac{TP + TN}{TP + TN + FP + FN}$$

$$\theta_2 = \frac{[(TP + TN)(TP + FP)] + [(TN + FN)(TN + FP)]}{(TP + TN + FP + FN)^2}$$

Son olarak değerlendirme kriterlerinden biri olarak Alıcı İşletim Karakteristik Eğrisi (Roc-Curve) incelenmiştir. ROC olasılık eğrisi olup, altındaki alan AUC değerini vermektedir. ROC eğrisi altında alan arttıkça sonuç performansı güçlenmektedir. Aşağıdaki şekilde ROC eğrisine örnek olarak verilmektedir.



Şekil 3.20 VGG-16 modeline ait bir ROC Eğrisi örneği

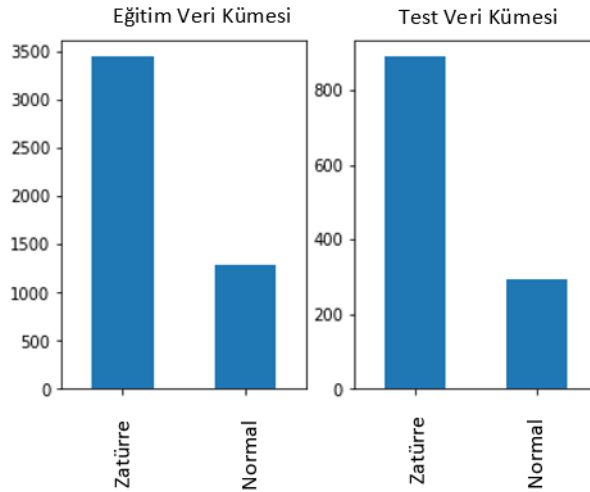
4. BULGULAR

Verilerin işlenip ortaya konması zaman almaktadır. Aşağıdaki tablolarda en iyi sonuç veren MobileNet modelinin diğer modellerle kıyaslanması ve sonuçları içeren bilgiler aşağıdaki gibi görülmektedir.

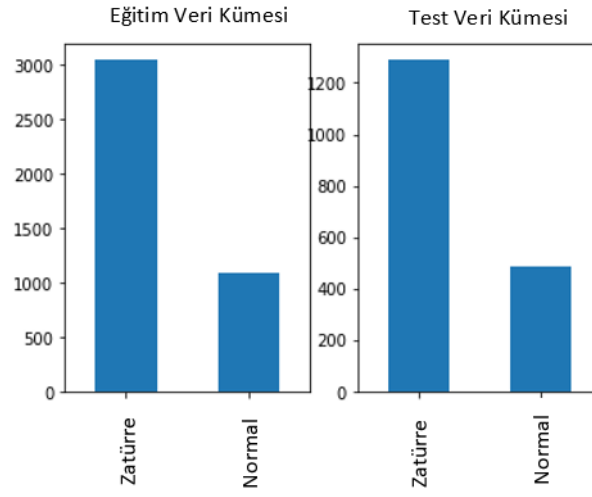
Modeller kullanılırken derin öğrenme tabanlı, Keras kütüphanesi tarafından desteklenen ImageDataGenerator fonksiyonu, görüntüleri daha sonra eğitmesi için verilen parametreleri kullanıp, hazır hale getirmektedir. Bu fonksiyon sayesinde modeller entegre edilmiştir.

Modellere ilk eğitim/test oranı %80/%20 olarak seçilmiş olup, Şekil 4.1’de veri kümesindeki sayıları grafikte gösterilmiştir. Modellere ikinci eğitim/test oranı %70/%30 olarak seçilmiş olup, Şekil 4.2’de veri kümesindeki sayıları grafikte gösterilmiştir. Modellere üçüncü eğitim/test oranı %50/%50 olarak seçilmiş olup, Şekil 4.3’de veri kümesindeki sayıları grafikte gösterilmiştir.

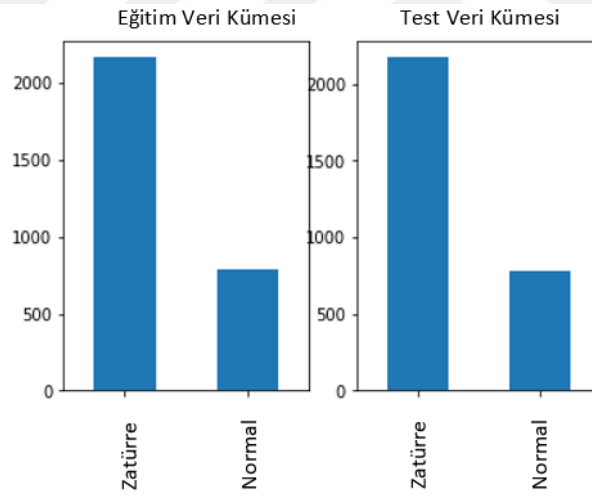
En iyi sonuç veren MobileNet modelinin veri kümesi üzerindeki eğitim ve test dağılımları Şekil 4.1’de olduğu gibi %80 şeklinde ayarlanmıştır.



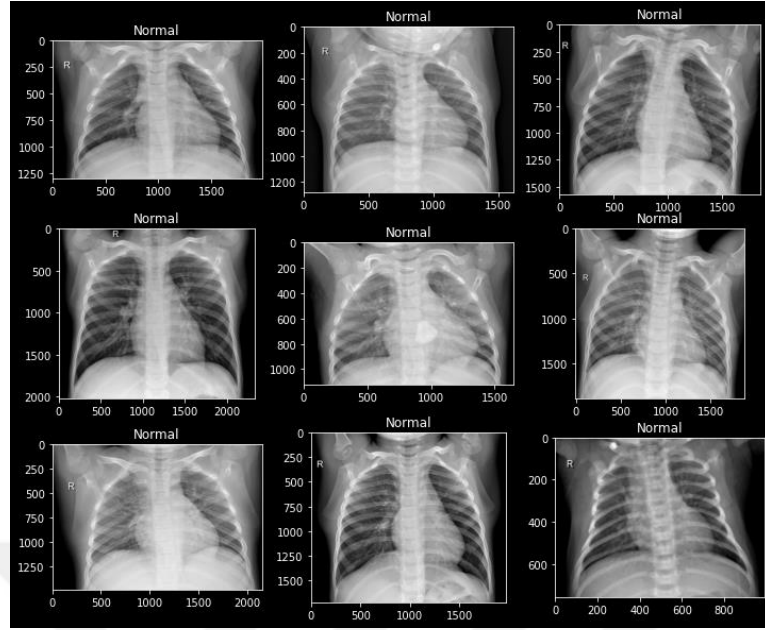
Şekil 4.1 Çapraz doğrulama olmadan veri kümesinin eğitim/testinin %80 bölünmesi



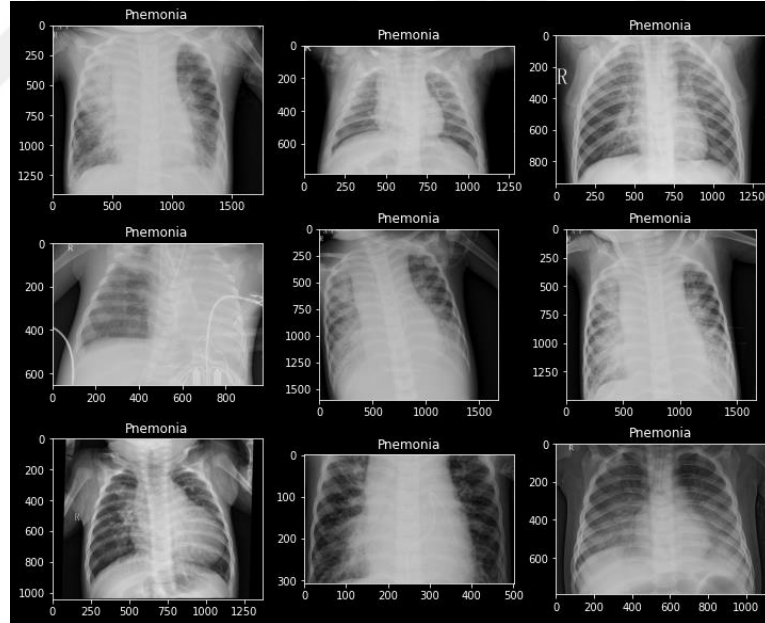
Şekil 4.2 Çapraz doğrulama olmadan veri kümesinin eğitim/testinin %70 bölünmesi



Şekil 4.3 Çapraz doğrulama olmadan veri kümesinin eğitim/testinin %50 bölünmesi



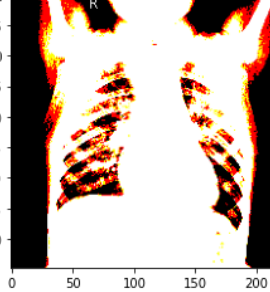
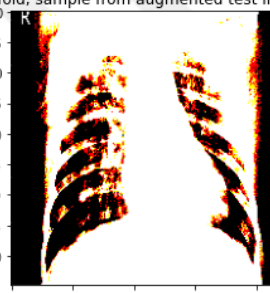
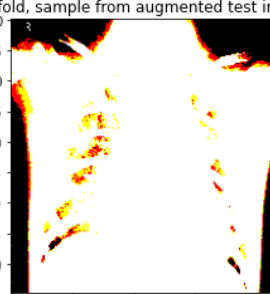
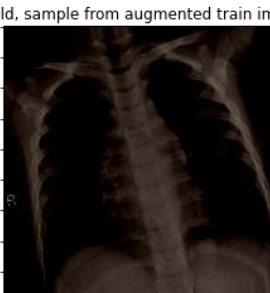
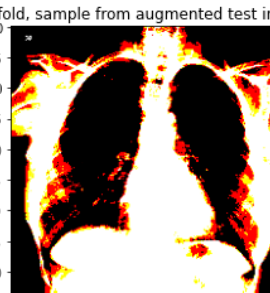
Fotoğraf 4.1 Veri kümesindeki Normal bireylerin farklı boyutlardaki görüntüsü



Fotoğraf 4.2. Veri kümesindeki Zatürre bireylerin farklı boyutlardaki görüntüsü

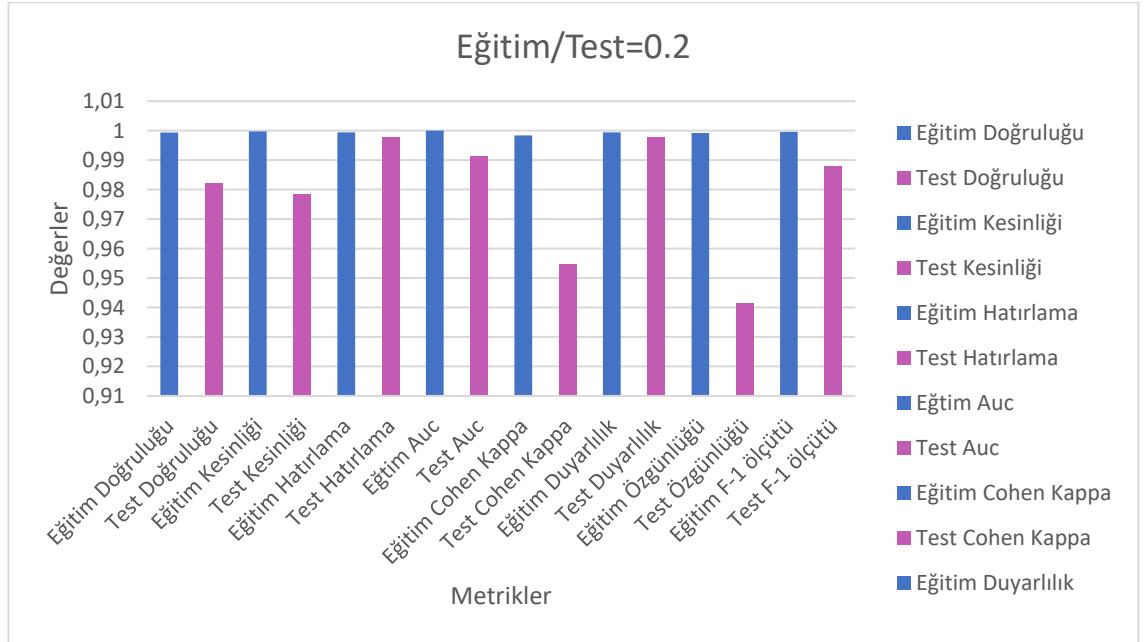
Fotoğraf 1 ve Fotoğraf 2’de Veri kümesi içerisinde farklı bireylerin görüntüleri: Yeniden boyutlandırma oranı=1/255., Kırpma =0.1, Döndürme=10, Yakınlaştırma=0.1 olacak şekilde ayarlanmıştır. Bu ayarlar tekrardan istenirse düzenlenebilmektedir.

Tablo 4.1 Modellerde kullanılan apraz doęrulama uygulanan grntler

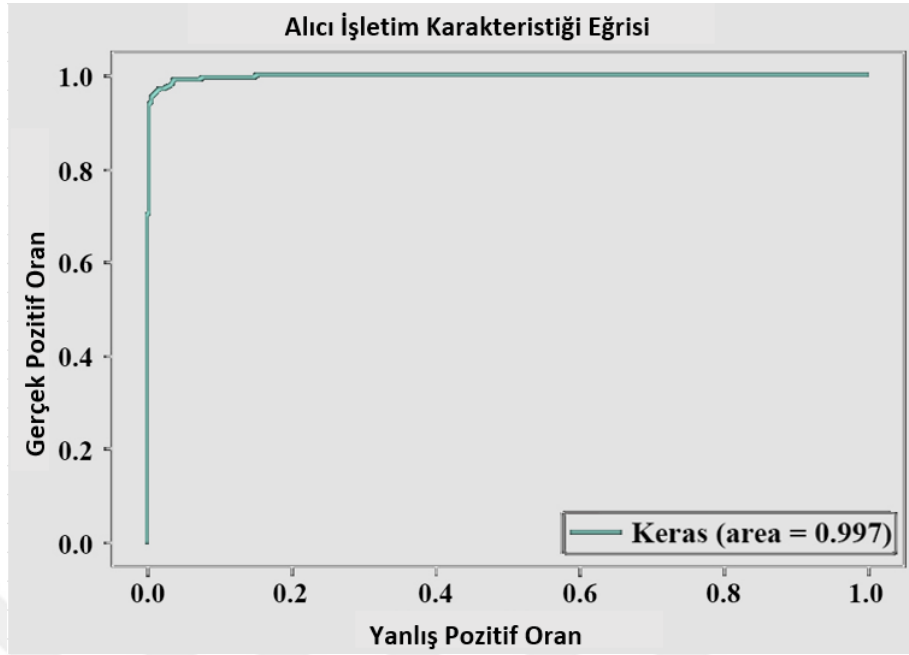
	EęİTİM GRNTLERİ	TEST GRNTLERİ
K=1 DEęERİ	1. fold, sample from augmented train images 	1. fold, sample from augmented test images 
K=2 DEęERİ	2. fold, sample from augmented train images 	2. fold, sample from augmented test images 
K=3 DEęERİ	3. fold, sample from augmented train images 	3. fold, sample from augmented test images 
K=4 DEęERİ	4. fold, sample from augmented train images 	4. fold, sample from augmented test images 
K=5 DEęERİ İİN	5. fold, sample from augmented train images 	5. fold, sample from augmented test images 

Tablo 4.2. En iyi Sonuç Veren MobileNet Modeline Ait Tablo

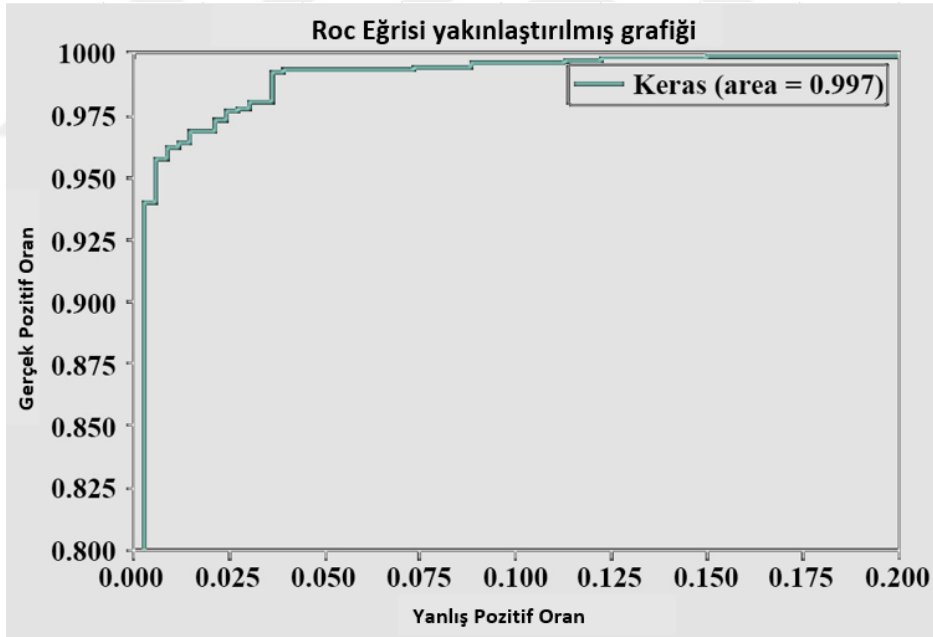
	Model=MobileNet
	Veri Artırımı=Yok
	eğitim/test=0.2
Eğitim Doğruluk	1
Test Doğruluk	0.983
Eğitim Kesinlik	1
Test Kesinlik	0.983
Eğitim Hatırlama	1
Test Hatırlama	0.992
Eğitim Auc	1
Test Auc	0.994
Eğitim Cohen Kappa	1
Test Cohen Kappa	0.957
Eğitim Duyarlılık	1
Test Duyarlılık	0.992
Eğitim Özgünlük	1
Test Özgünlük	0.957
Eğitim F-1 ölçütü	1
Test F-1 ölçütü	0.988



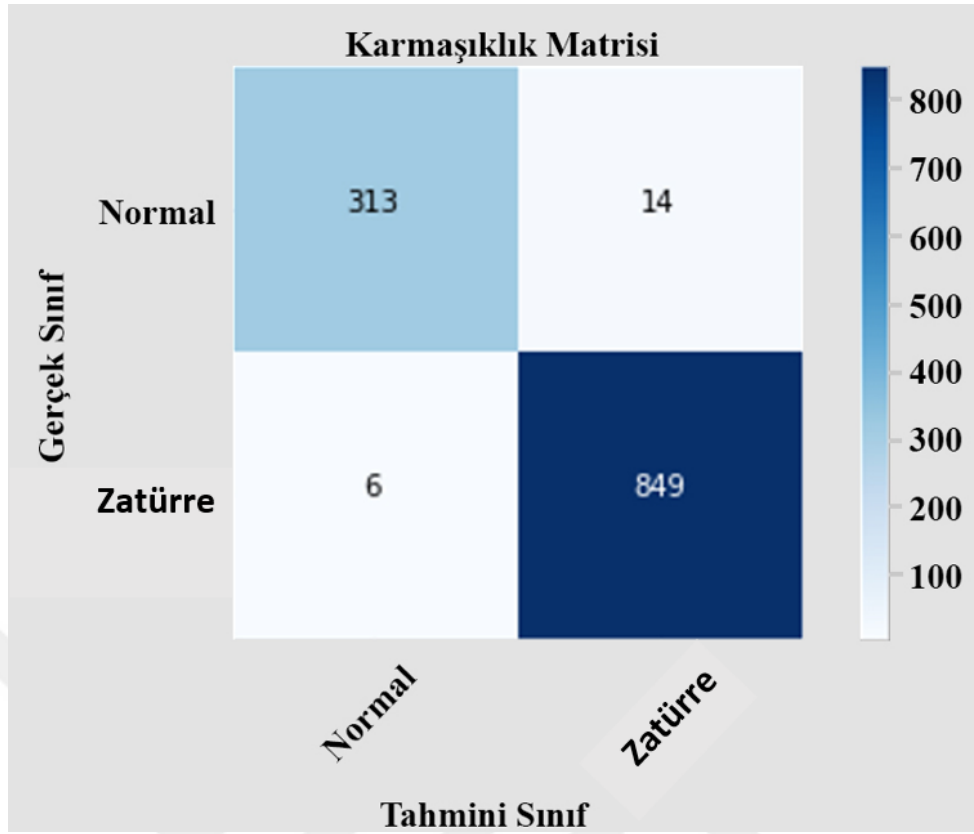
Şekil 4.4 MobileNet Modelinin Çapraz Doğrulama uygulanmamış Eğitim/test=0.2 seçilmiş grafiği



Fotoğraf 4.3 En iyi sonuç veren MobileNet modeline ait Roc eğrisi



Fotoğraf 4.4 En iyi sonuç veren MobileNet modeline ait Roc eğrisi yakınlaştırılmış görüntüsü

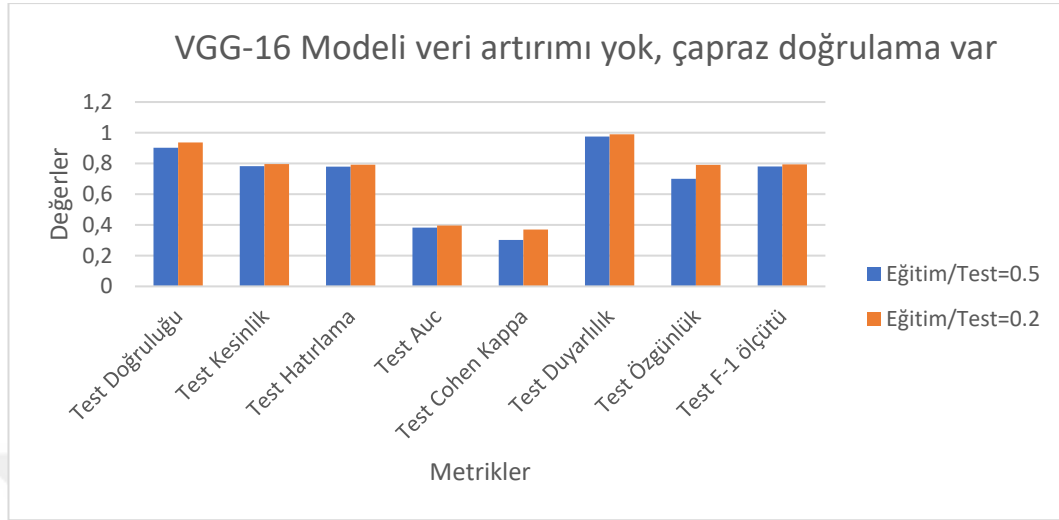


Fotoğraf 4.5 En iyi sonuç veren MobileNet modeline ait Karmaşıklık Matrisi

Fotoğraf 4.5’de gösterildiği gibi 313 birey normaldir ve zatürre hastalığı yoktur. Yani 313 gerçek veriyi model 313’ünün de doğru tahminde bulunmaktadır. Matris de olan 14 birey ise zatürre kabul edilmesine rağmen gerçekte normal bireyler olarak görülmektedir. Matris de 6 olan birey sayısı ise zatürre hastalığına sahip fakat tahminen normal olarak göstermektedir. Matris de bulunan son 849 birey ise gerçekte zatürre hastalığına sahip ve tahminen de hasta olarak kabul edilmektedir. Bu matrise bakıldığında amaca göre hangi skor değerinin yüksek olması isteniyorsa ona göre tercih yapılmalıdır.

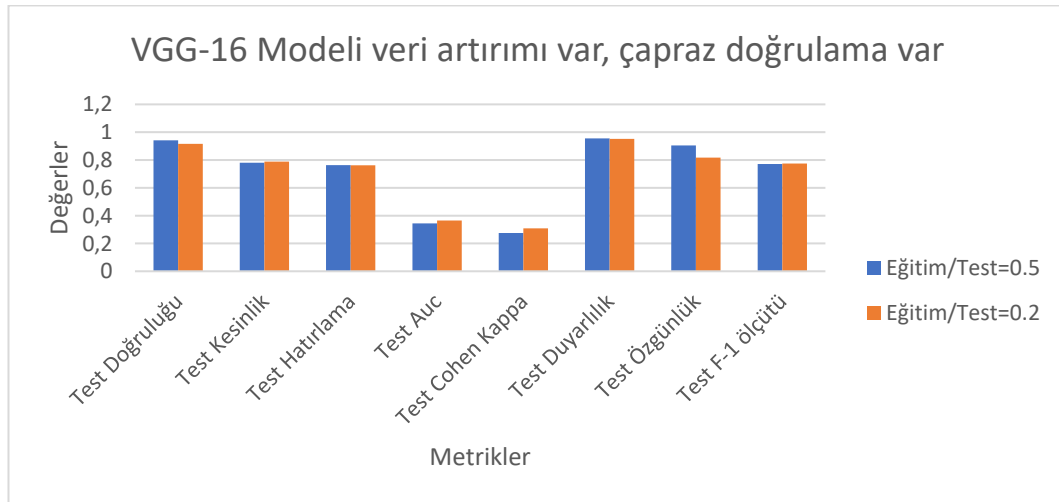
Diğer modellerden elde edilen sonuçlar aşağıdaki tabloda gösterilecektir. Sırasıyla VGG-16, VGG-19, MobileNet, InceptionV3, Xception modellerinin sonuçları mevcuttur. Bu bölümde 2’ye ayrılacak sonuçlar için her model için çapraz doğrulamalı (cvli) ve çapraz doğrulamasız (cvsiz) olarak değerlendirilecektir. Ayrıca her bölüm için de veri artırımı var ya da yok şeklinde de ayrılacaktır.

- **VGG-16 Modelinin Çapraz Doğrulama Uygulanmış Görüntüleri**



Şekil 4.5 VGG-16 Modeline veri artırımı uygulanmamış çapraz doğrulamalı grafik

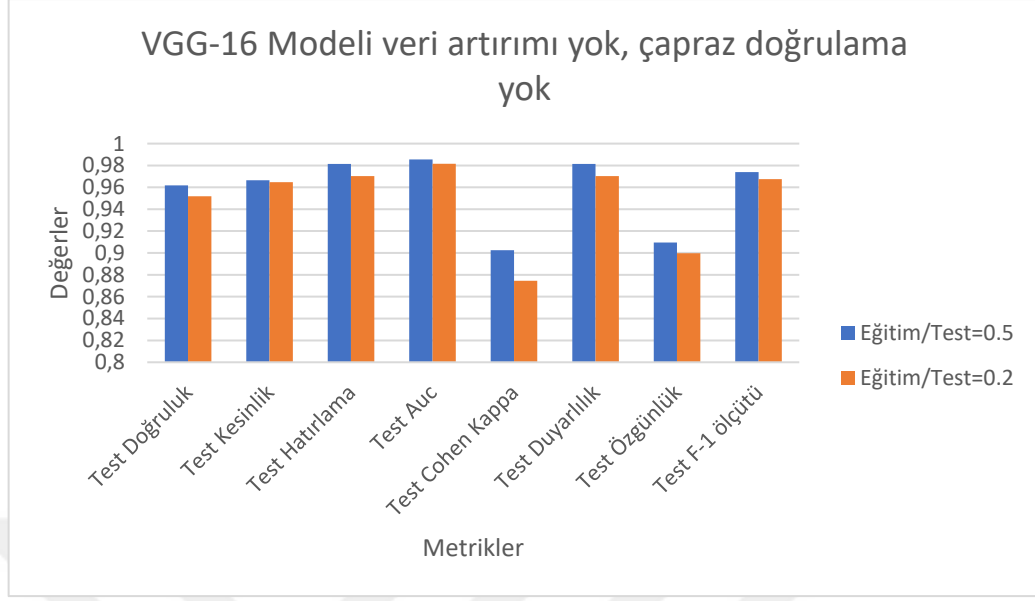
Şekil 4.5’de Eğitim/test oranı=0,2 olanların ortalamada daha iyi performans verdiği grafikten görülmektedir.



Şekil 4.6 VGG-16 Modeline veri artırımı uygulanmış çapraz doğrulamalı grafik

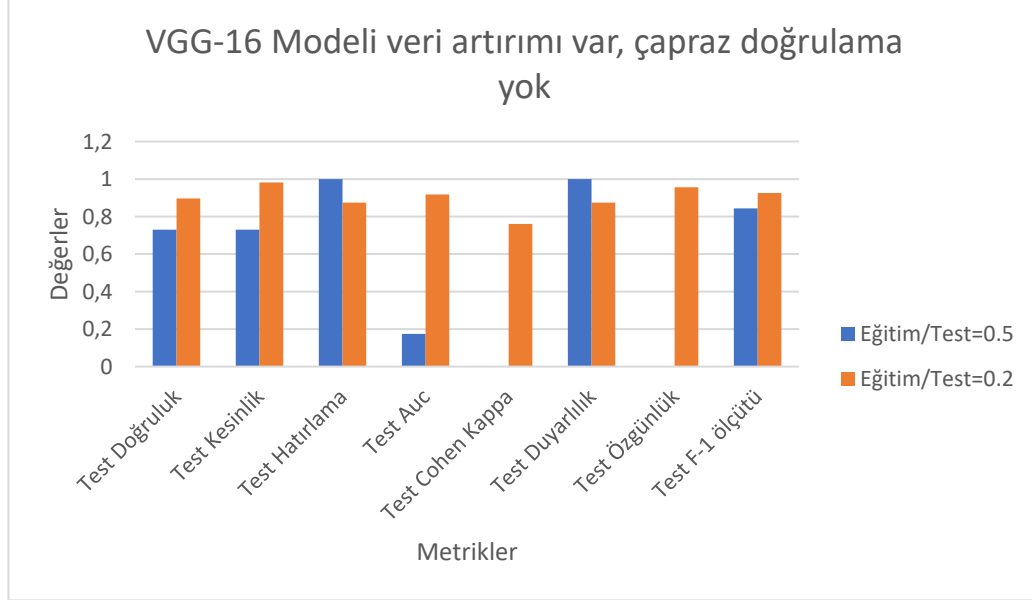
Şekil 4.6’da Eğitim/test oranı=0,2 olanların ortalamada daha iyi performans verdiği grafikten görülmektedir.

- **VGG-16 Modeline Çapraz Doğrulama Uygulanmamış Görüntüleri**



Şekil 4.7 VGG-16 Modeline veri artırımı uygulanmamış çapraz doğrulama uygulanmamış grafik

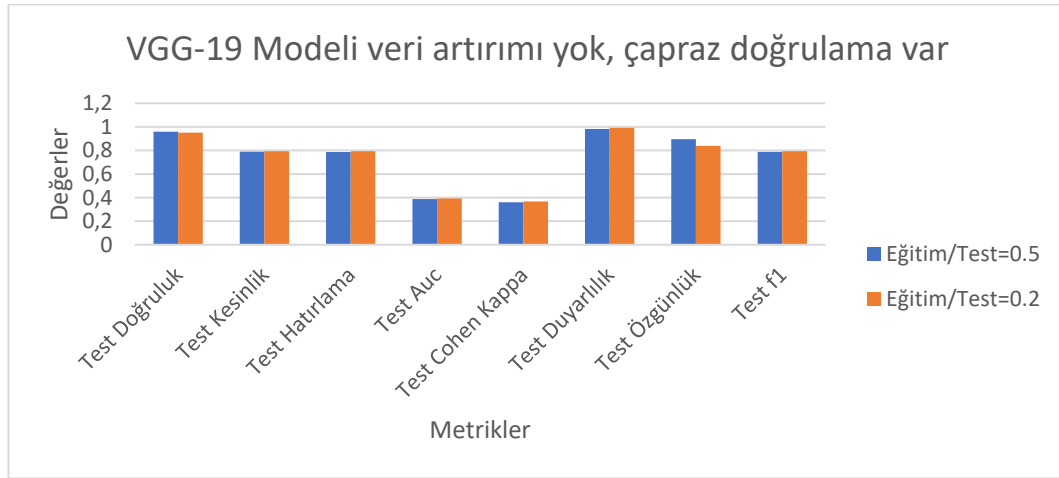
Şekil 4.7’de Eğitim/test oranı=0,5 olanların ortalamada daha iyi performans verdiği grafikten görülmektedir.



Şekil 4.8 VGG-16 Modeline veri artırımı uygulanmış çapraz doğrulama uygulanmamış grafik

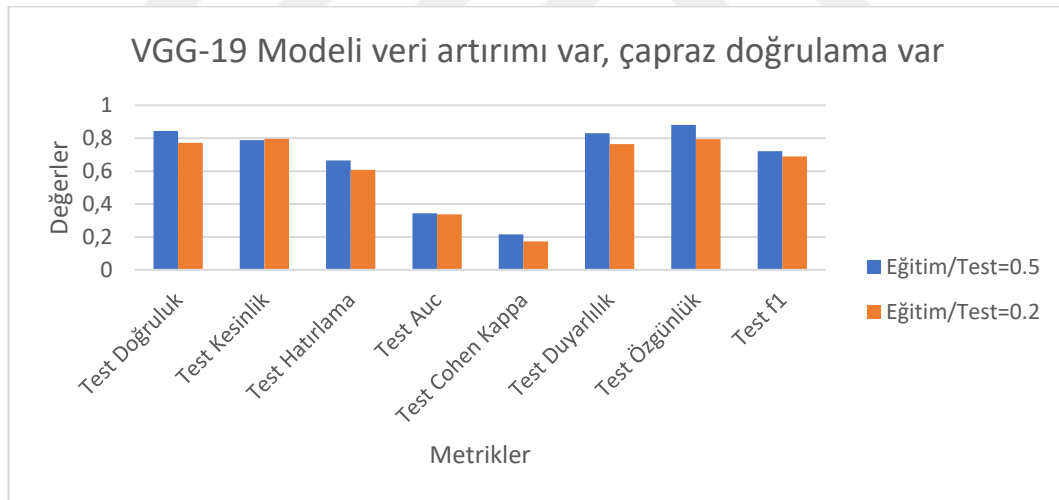
Şekil 4.8’de Eğitim/test oranı=0,2 olanların ortalamada daha iyi performans verdiği grafikten görülmektedir.

- VGG-19 Modeline Çapraz Doğrulama Uygulanmış Görüntüleri



Şekil 4.9 VGG-19 Modeline veri artırımı uygulanmamış çapraz doğrulama uygulanmış grafik

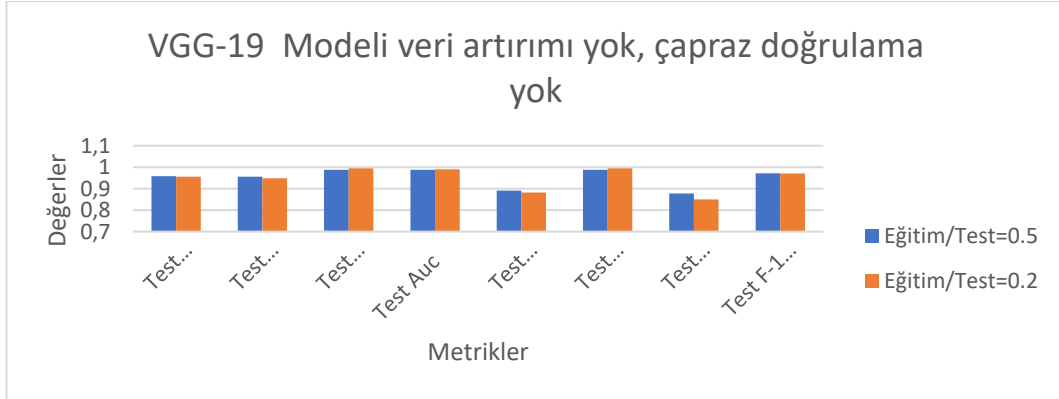
Şekil 4.9’da Eğitim/test oranı=0,2 olanların ortalamada daha iyi performans verdiği grafikten görülmektedir.



Şekil 4.10 VGG-19 Modeline veri artırımı uygulanmış çapraz doğrulama uygulanmış grafik

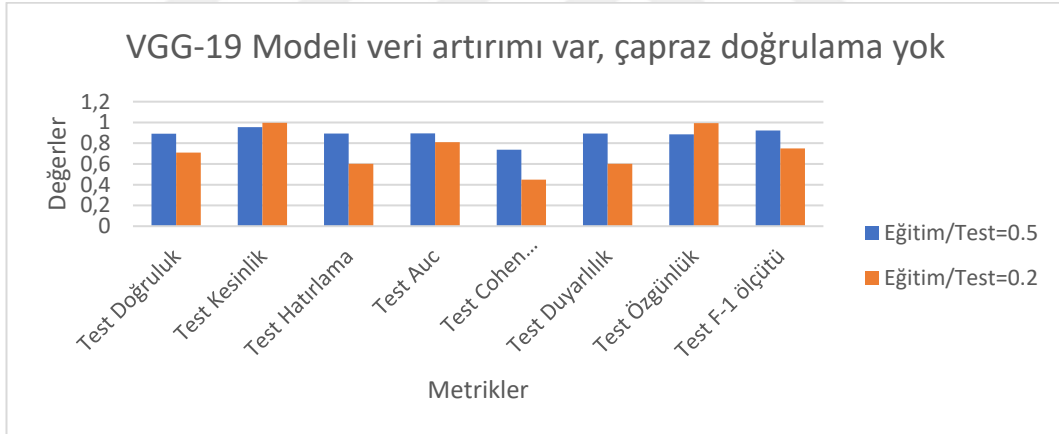
Şekil 4.10’da Eğitim/test oranı=0,5 olanların ortalamada daha iyi performans verdiği grafikten görülmektedir

- **VGG-19 Modeline Çapraz Doğrulama Uygulanmamış Görüntüleri**



Şekil 4.11 VGG-19 Modeline veri artırımı uygulanmamış çapraz doğrulama uygulanmamış grafik

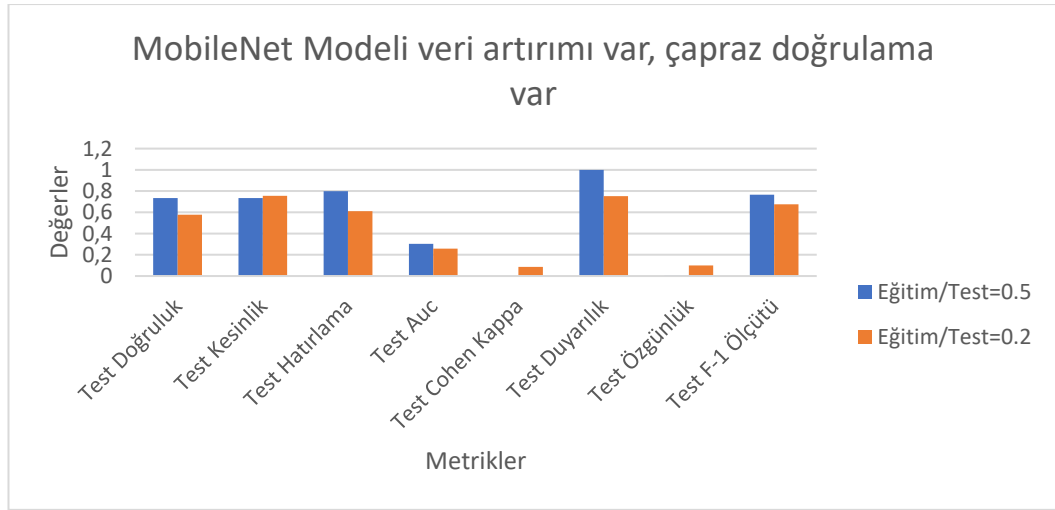
Şekil 4.11’de Eğitim/test oranı=0,2 olanların ortalamada daha iyi performans verdiği grafikten görülmektedir.



Şekil 4.12 VGG-19 Modeline veri artırımı uygulanmış çapraz doğrulama uygulanmamış grafik

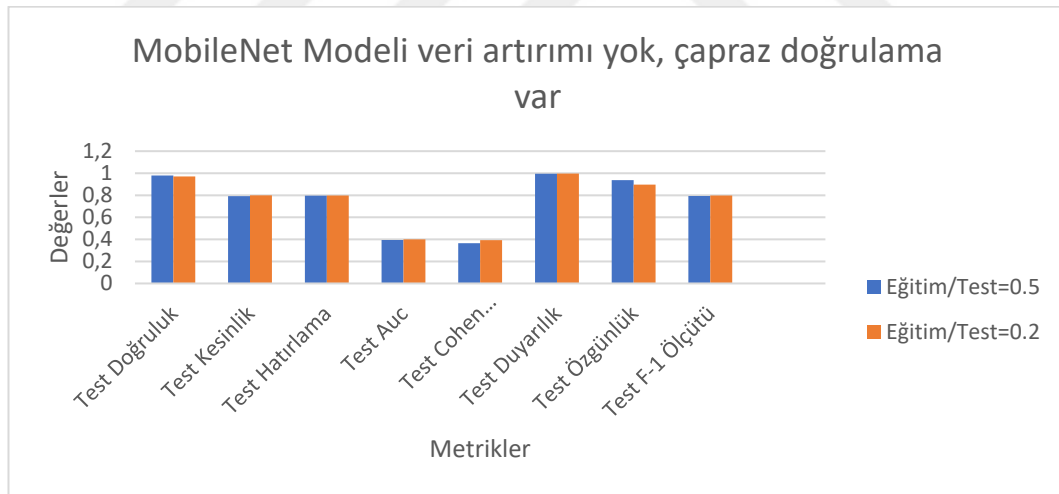
Şekil 4.12’de Eğitim/test oranı=0,5 olanların ortalamada daha iyi performans verdiği grafikten görülmektedir

- **MobileNet Modeline Çapraz Doğrulama Uygulanmış Görüntüleri**



Şekil 4.13 MobileNet Modeline veri artırımı uygulanmış çapraz doğrulama uygulanmış grafik

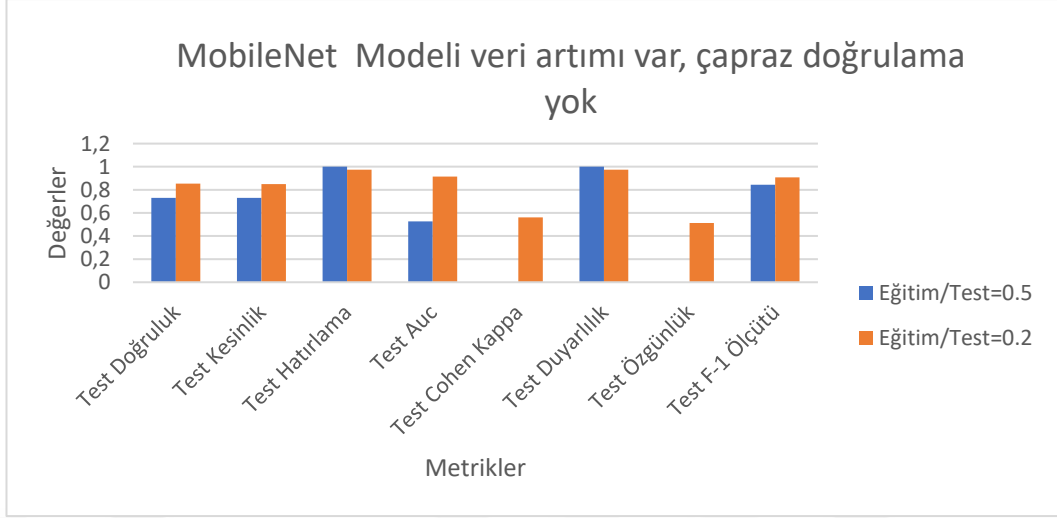
Şekil 4.13’de Eğitim/test oranı=0,2 olanların ortalamada daha iyi performans verdiği grafikten görülmektedir.



Şekil 4.14 MobileNet Modeline veri artırımı uygulanmamış çapraz doğrulama uygulanmış grafik

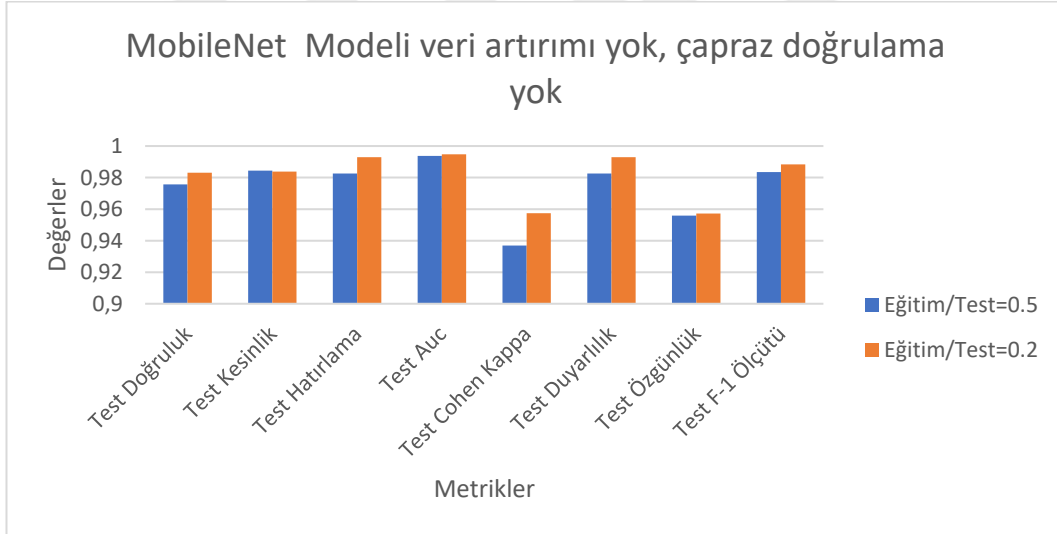
Şekil 4.14’de Eğitim/test oranı=0,5 ve 0,2 olanların yaklaşık eşite dercede daha iyi performans verdiği grafikten görülmektedir.

- **MobileNet Modeline Çapraz Doğrulama Uygulanmamış Görüntüleri**



Şekil 4.15 MobileNet Modeline veri artırımı uygulanmış çapraz doğrulama uygulanmamış grafik

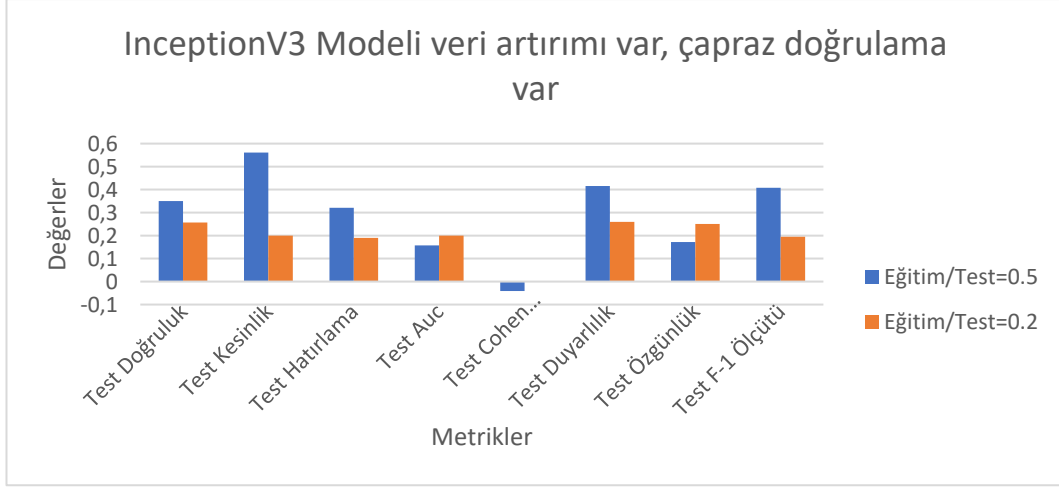
Şekil 4.15’de Eğitim/test oranı=0,2 olanların ortalamada daha iyi performans verdiği grafikten görülmektedir.



Şekil 4.16 MobileNet Modeline veri artırımı uygulanmamış çapraz doğrulama uygulanmamış grafik

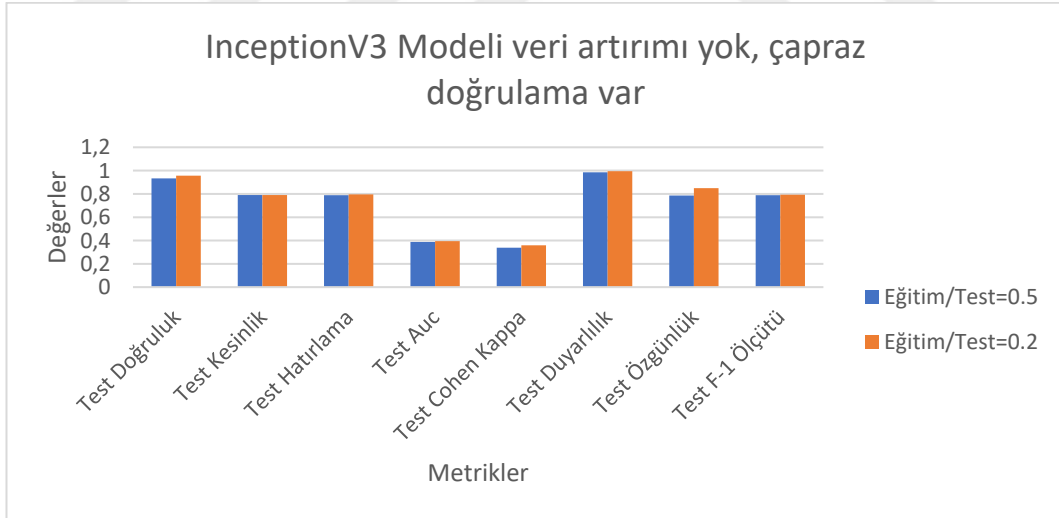
Şekil 4.16’da Eğitim/test oranı=0,2 olanların ortalamada daha iyi performans verdiği grafikten görülmektedir.

- InceptionV3 Modeline Çapraz Doğrulama Uygulanmış Görüntüleri



Şekil 4.17 InceptionV3 Modeline veri artırımı uygulanmış çapraz doğrulama uygulanmış grafik

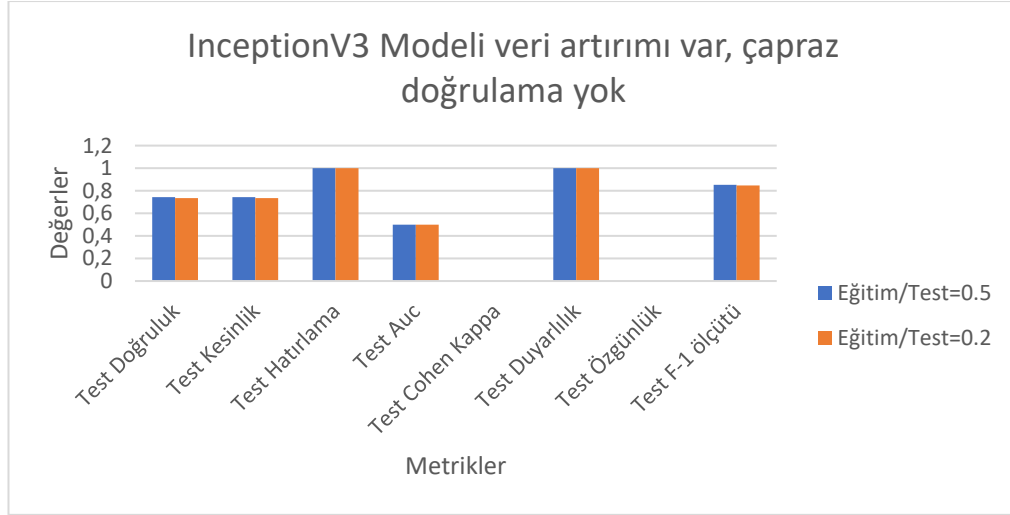
Şekil 4.17’de Eğitim/test oranı=0,2 olanların ortalamada daha iyi performans verdiği grafikten görülmektedir.



Şekil 4.18 InceptionV3 Modeline veri artırımı uygulanmamış çapraz doğrulama uygulanmış grafik

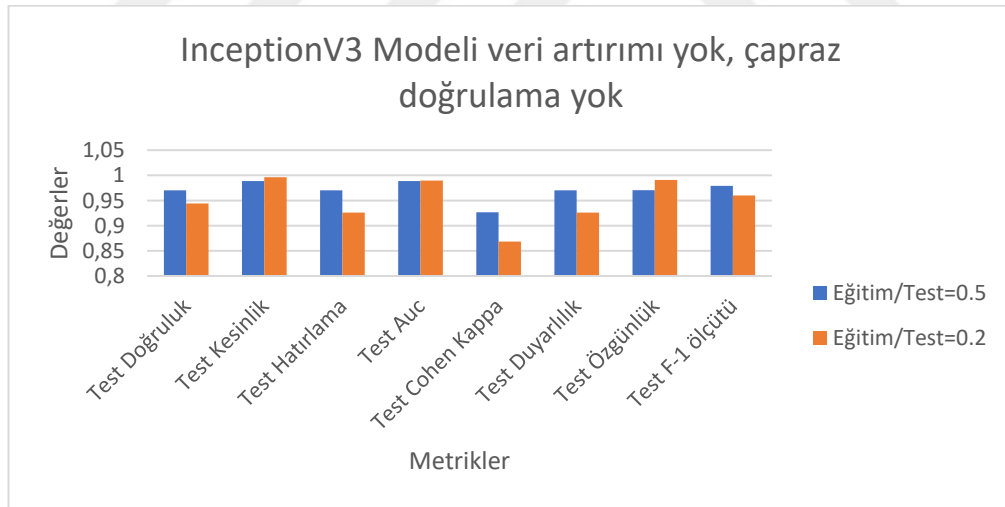
Şekil 4.18’de Eğitim/test oranı=0,2 olanların ortalamada daha iyi performans verdiği grafikten görülmektedir.

- **InceptionV3 Modeline Çapraz Doğrulama Uygulanmamış Görüntüleri**



Şekil 4.19 InceptionV3 Modeline veri artırımı uygulanmış çapraz doğrulama uygulanmamış grafik

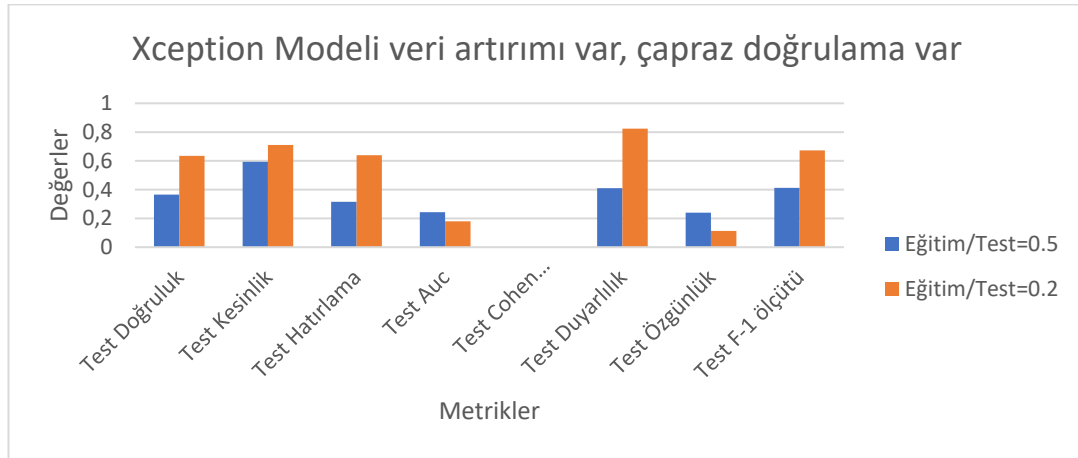
Şekil 4.19’da Eğitim/test oranı=0,2 olanların ortalamada daha iyi performans verdiği grafikten görülmektedir.



Şekil 4.20 InceptionV3 Modeline veri artırımı uygulanmamış çapraz doğrulama uygulanmamış grafik

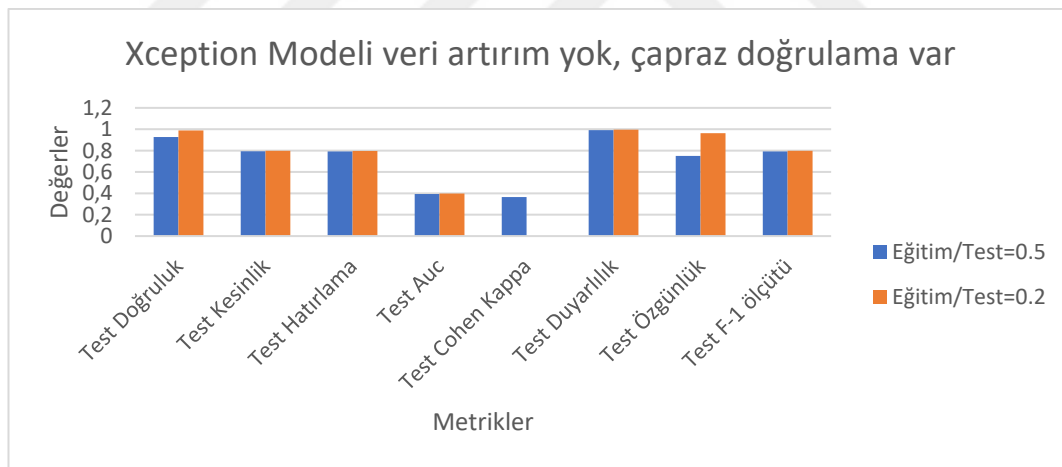
Şekil 4.20’de Eğitim/test oranı=0,5 olanların ortalamada daha iyi performans verdiği grafikten görülmektedir.

- Xception Modeline Çapraz Doğrulama Uygulanmış Görüntüleri



Şekil 4.21 Xception Modeline veri artırımı uygulanmış çapraz doğrulama uygulanmış grafik

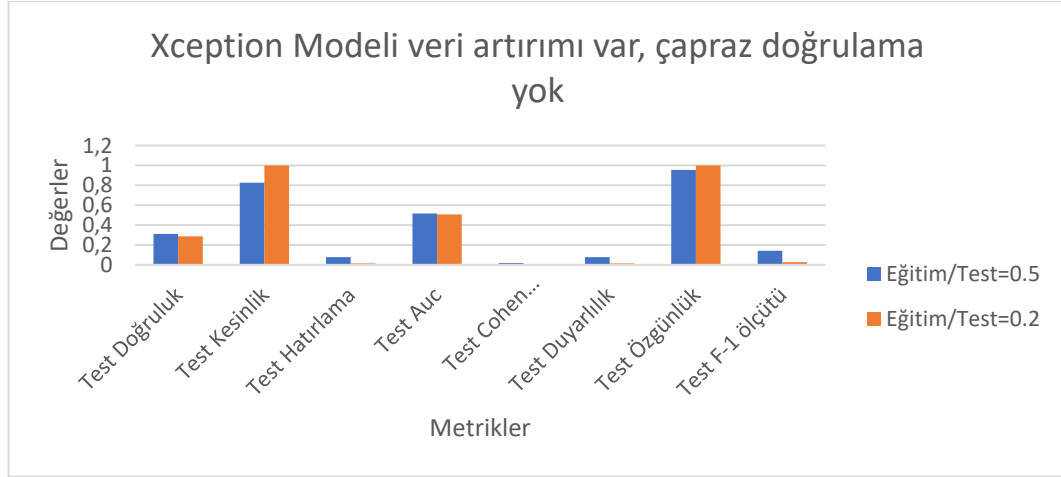
Şekil 4.21’de Eğitim/test oranı=0,2 olanların ortalamada daha iyi performans verdiği grafikten görülmektedir.



Şekil 4.22 Xception Modeline veri artırımı uygulanmamış çapraz doğrulama uygulanmış grafik

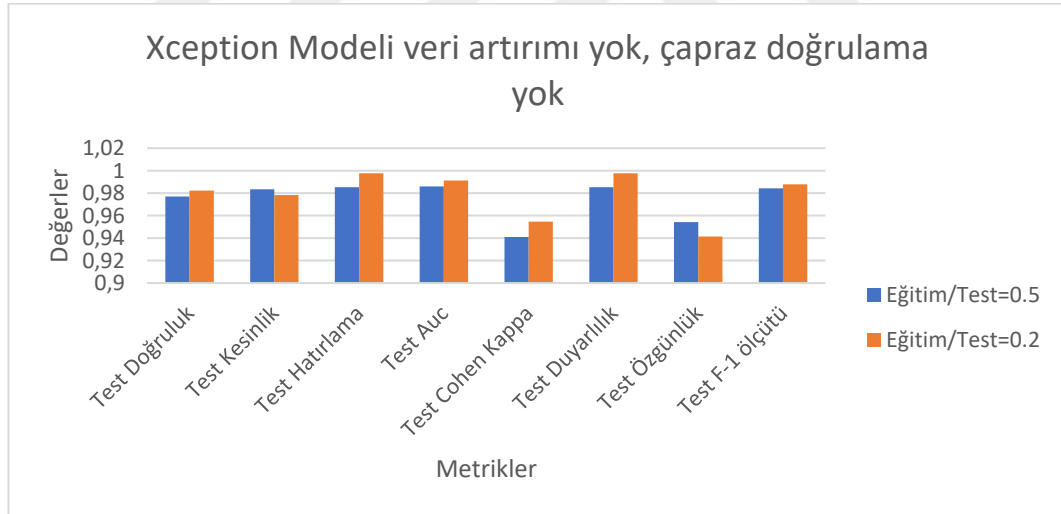
Şekil 4.22’de Eğitim/test oranı=0,5 olanların ortalamada daha iyi performans verdiği grafikten görülmektedir.

- **Xception Modeline Çapraz Doğrulama Uygulanmamış Görüntüleri**



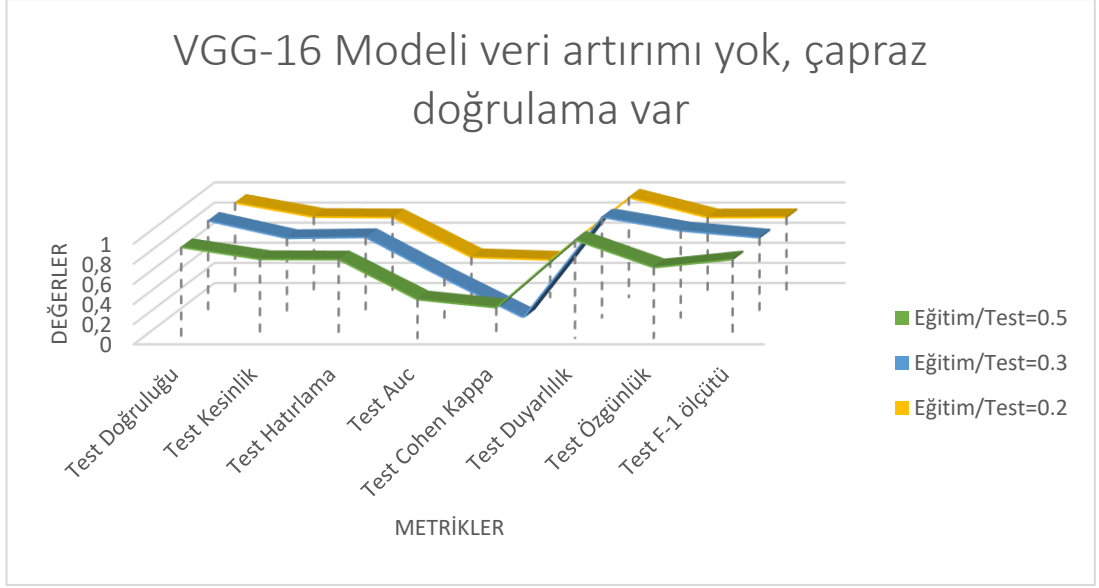
Şekil 4.23 Xception Modeline veri artırımı uygulanmış çapraz doğrulama uygulanmamış grafik

Şekil 4.23’de Eğitim/test oranı=0,5 olanların ortalamada daha iyi performans verdiği grafikten görülmektedir.



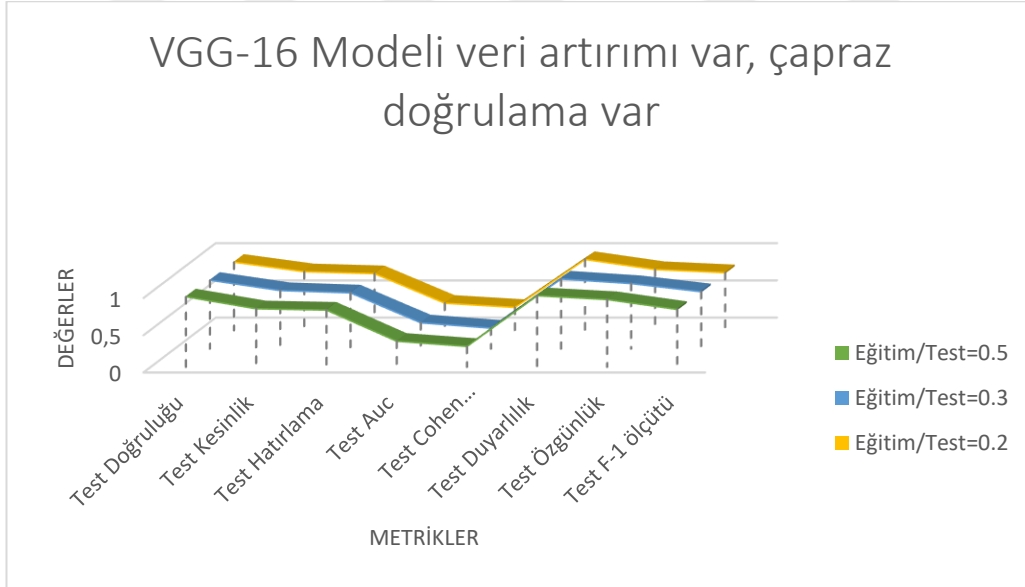
Şekil 4.24 Xception Modeline veri artırımı uygulanmamış çapraz doğrulama uygulanmamış grafik

Şekil 4.24’de Eğitim/test oranı=0,2 olanların ortalamada daha iyi performans verdiği grafikten görülmektedir.



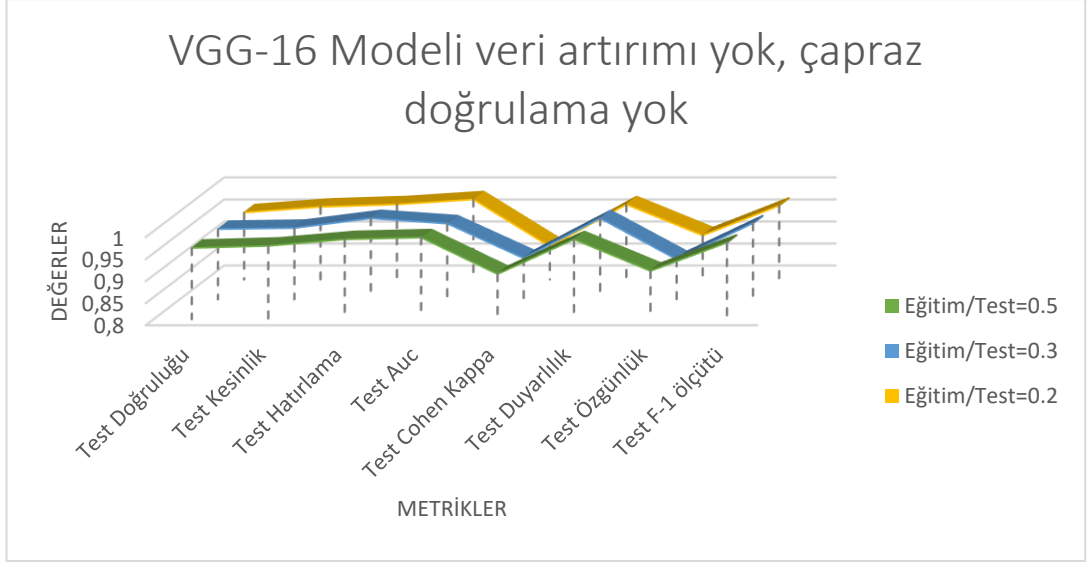
Şekil 4.25 VGG-16 Modeline veri artırımı uygulanmamış çapraz doğrulama uygulanmış üçlü grafik

Şekil 4.25’de VGG-16 modeli -veri artırımı yok ve çapraz doğrulama var, model-method kombinasyonu, farklı sına seti (hold-out) yüzdeleri (0,5/0,3/0,2) ile çalıştırılmıştır. Grafik incelendiğinde sına seti (hold-out)=0.2 parametresinin, 3 farklı sına seti yöntemi parametresi içinde, en iyi sonucu verdiği görülmüştür.



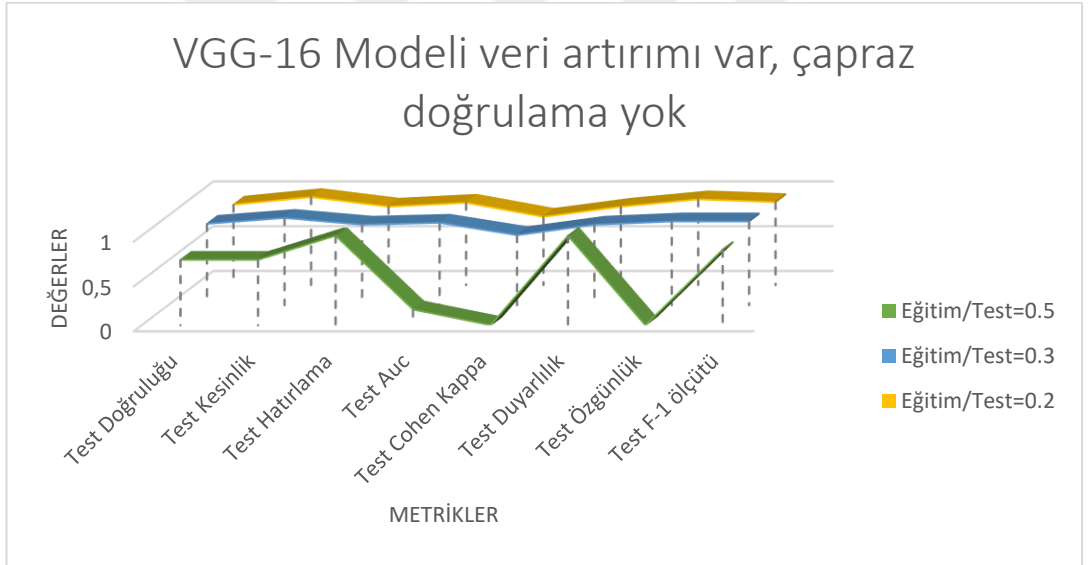
Şekil 4.26 VGG-16 Modeline veri artırımı uygulanmış çapraz doğrulama uygulanmış üçlü grafik

Şekil 4.26’da VGG-16 modelinin yaklaşık olarak 3 farklı sına seti yönteminde de aynı sonuçları vermiştir.



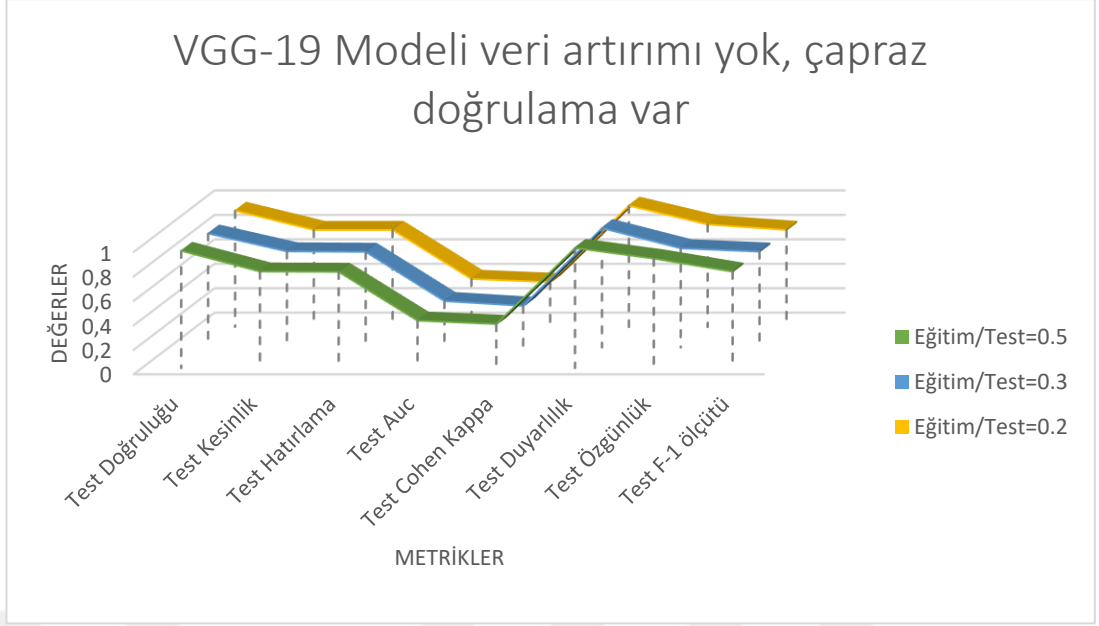
Şekil 4.27 VGG-16 Modeline veri artırımı uygulanmamış çapraz doğrulama uygulanmamış üçlü grafik

Şekil 4.27’de VGG-16 modelinin yaklaşık olarak 3 farklı sına seti yöntemi yönteminde de yaklaşık aynı sonuçları vermiştir.



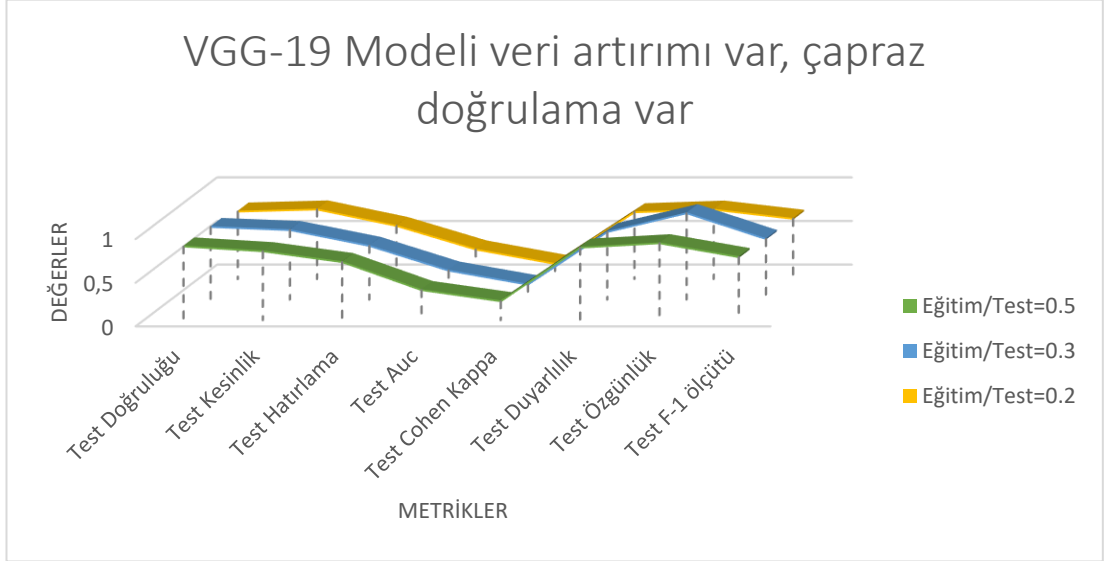
Şekil 4.28 VGG-16 Modeline veri artırımı uygulanmış çapraz doğrulama uygulanmamış üçlü grafik

Şekil 4.28’de VGG-16 modelinin yaklaşık olarak 3 farklı sına seti yöntemi yönteminde en iyi sonucu 0,2 olanda vermiştir.



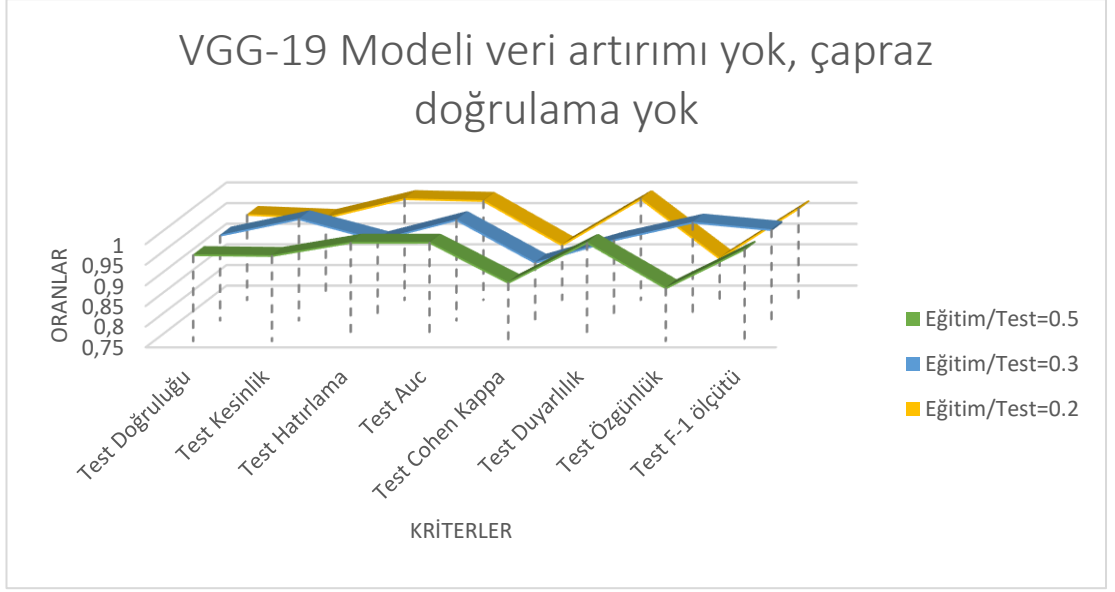
Şekil 4.29 VGG-19 Modeline veri artırımı uygulanmamış çapraz doğrulama uygulanmış üçlü grafik

Şekil 4.29'da VGG-19 modelinin yaklaşık olarak 3 farklı sınaama seti yönteminde de yaklaşık aynı sonuçları vermiştir.



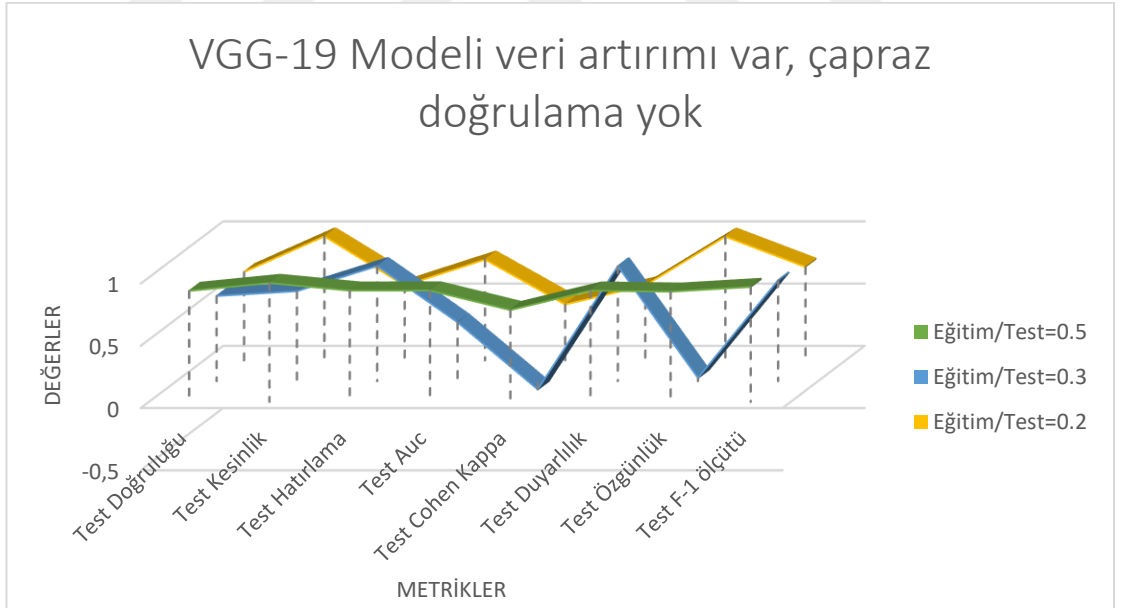
Şekil 4.30 VGG-19 Modeline veri artırımı uygulanmış çapraz doğrulama uygulanmış üçlü grafik

Şekil 4.30'da VGG-19 modelinin yaklaşık olarak 3 farklı sınaama seti yönteminde en iyi sonucu 0,3 olanda vermiştir.



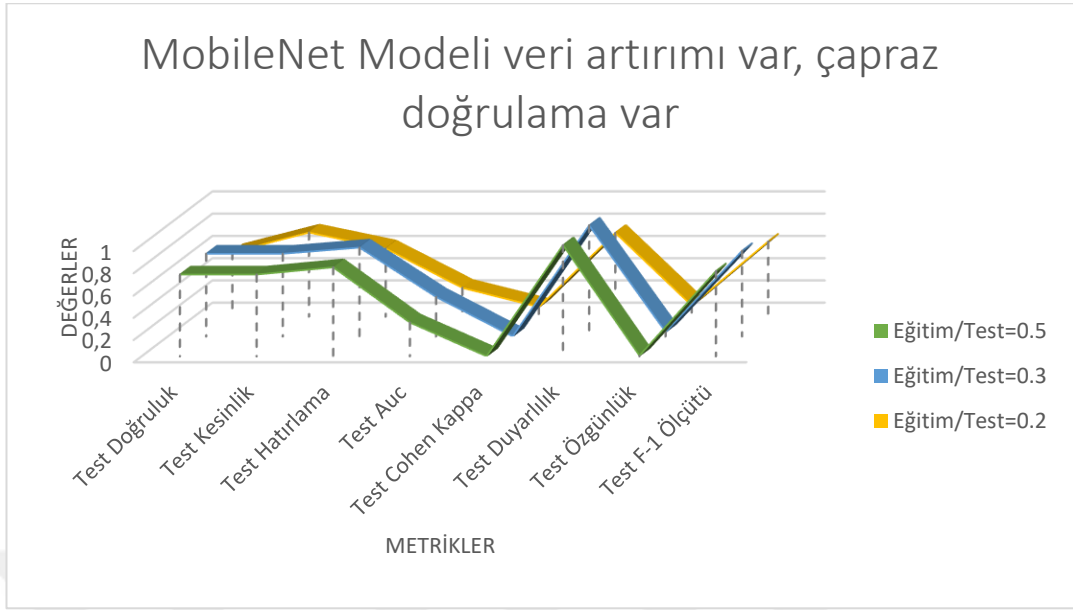
Şekil 4.31 VGG-19 Modeline veri artırımı uygulanmamış çapraz doğrulama uygulanmamış üçlü grafik

Şekil 4.3’de VGG-19 modelinin yaklaşık olarak 3 farklı sınaama seti yönteminde en iyi sonucu 0,3 olanda vermiştir.



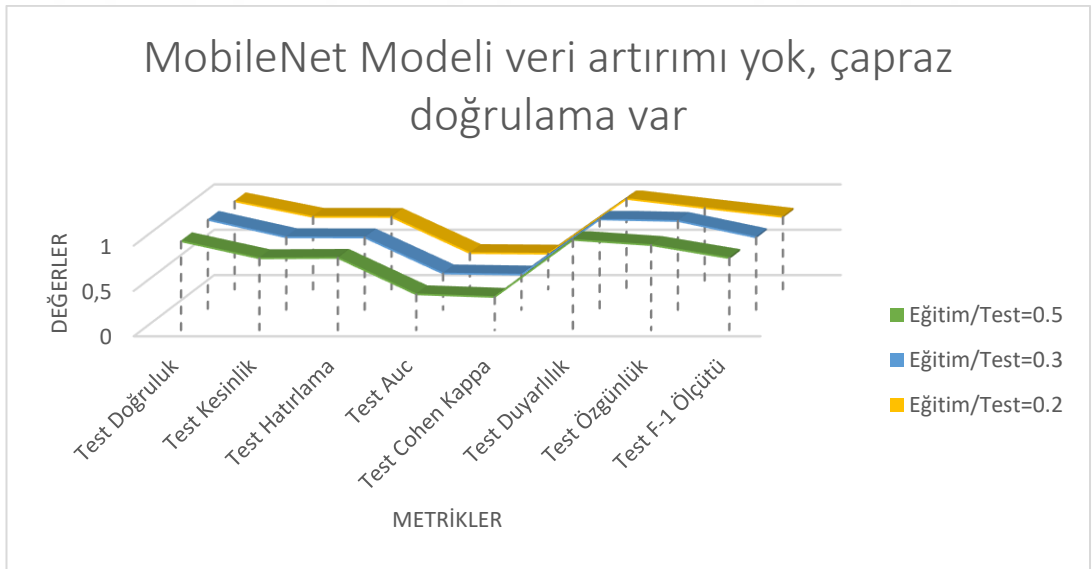
Şekil 4.32 VGG-19 Modeline veri artırımı uygulanmış çapraz doğrulama uygulanmamış üçlü grafik

Şekil 4.32’de VGG-19 modelinin yaklaşık olarak 3 farklı sınaama seti yönteminde en iyi sonucu 0,5 olanda vermiştir.



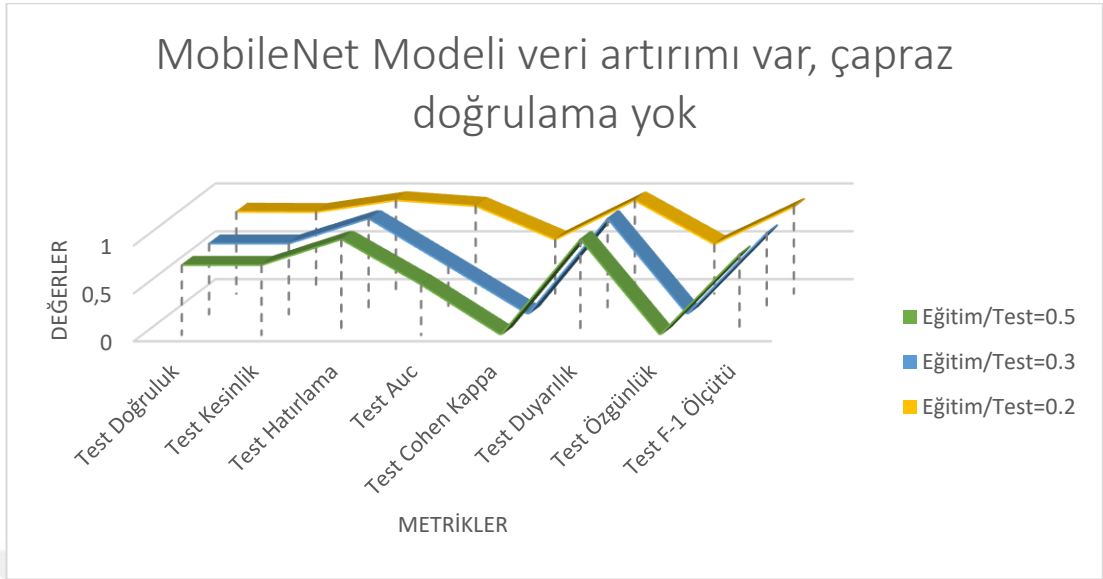
Şekil 4.33 MobileNet Modeline veri artırımı uygulanmış çapraz doğrulama uygulanmış üçlü grafik

Şekil 4.33’de MobileNet modelinin yaklaşık olarak 3 farklı sınaması yönteminde en iyi sonucu 0,5 olanda vermiştir.



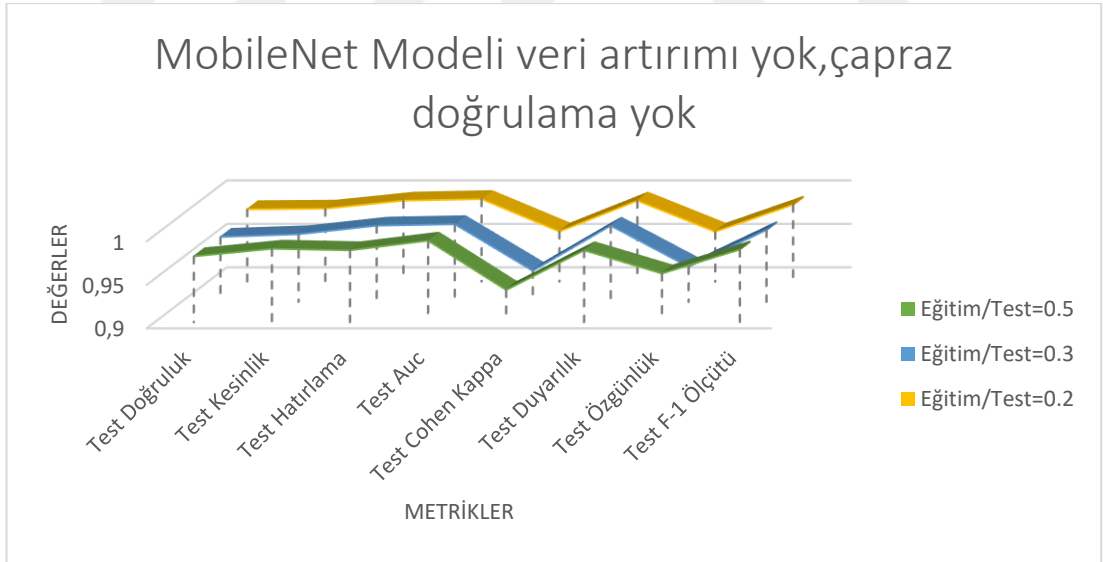
Şekil 4.34 MobileNet Modeline veri artırımı uygulanmamış çapraz doğrulama uygulanmış üçlü grafik

Şekil 4.34’de MobileNet modelinin yaklaşık olarak 3 farklı sınaması yönteminde de aynı sonuçları vermiştir.



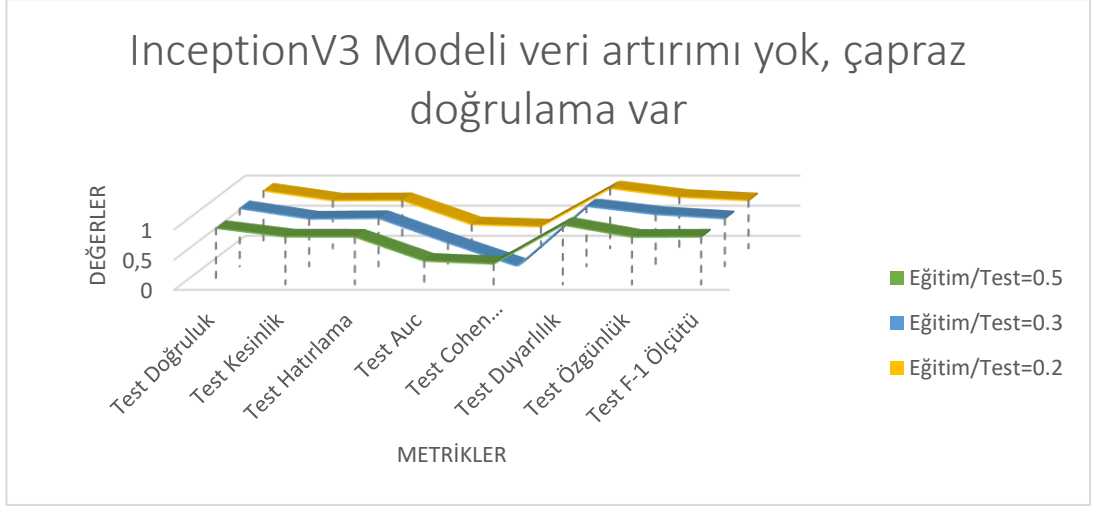
Şekil 4.35 MobileNet Modeline veri artırımı uygulanmış çapraz doğrulama uygulanmamış üçlü grafik

Şekil 4.35’de MobileNet modelinin yaklaşık olarak 3 farklı sınaama seti yönteminde en iyi sonucu 0,2 olanda vermiştir.



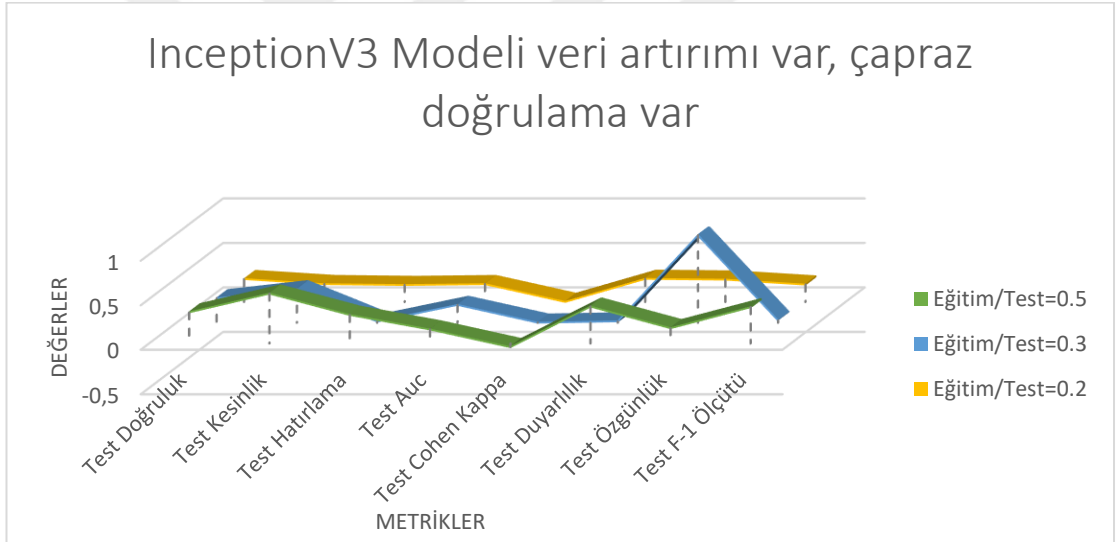
Şekil 4.36 MobileNet Modeline veri artırımı uygulanmamış çapraz doğrulama uygulanmamış üçlü grafik

Şekil 4.36’da MobileNet modelinin yaklaşık olarak 3 farklı sınaama seti yönteminde en iyi sonucu 0,2 olanda vermiştir.



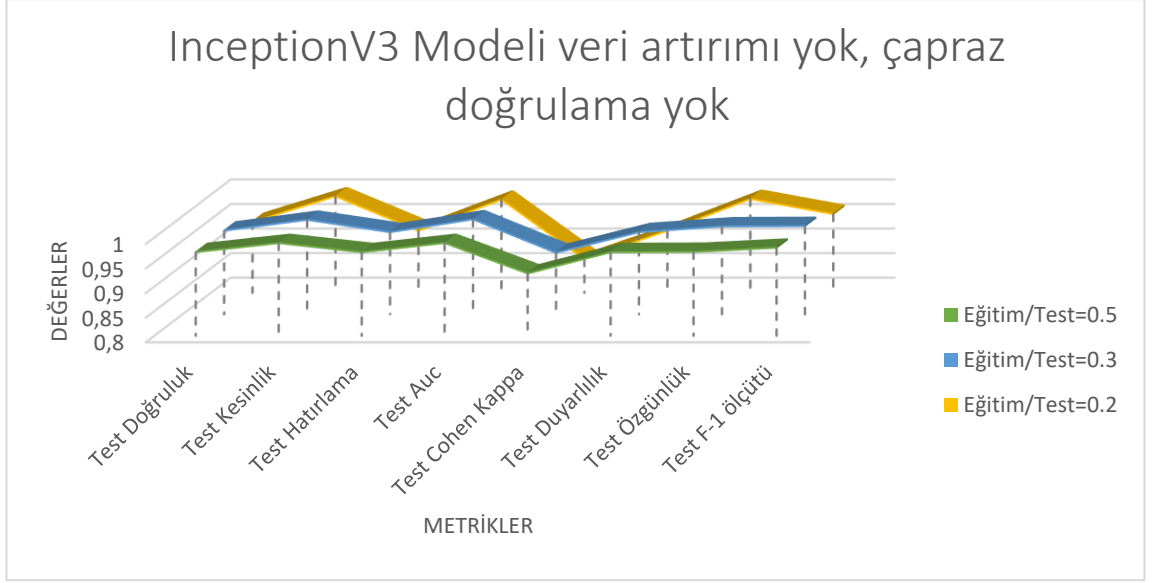
Şekil 4.37 InceptionV3 Modeline veri artırımı uygulanmamış çapraz doğrulama uygulanmış üçlü grafik

Şekil 4.37’de InceptionV3 modelinin yaklaşık olarak 3 farklı sınama seti yönteminde de aynı sonuçları vermiştir.



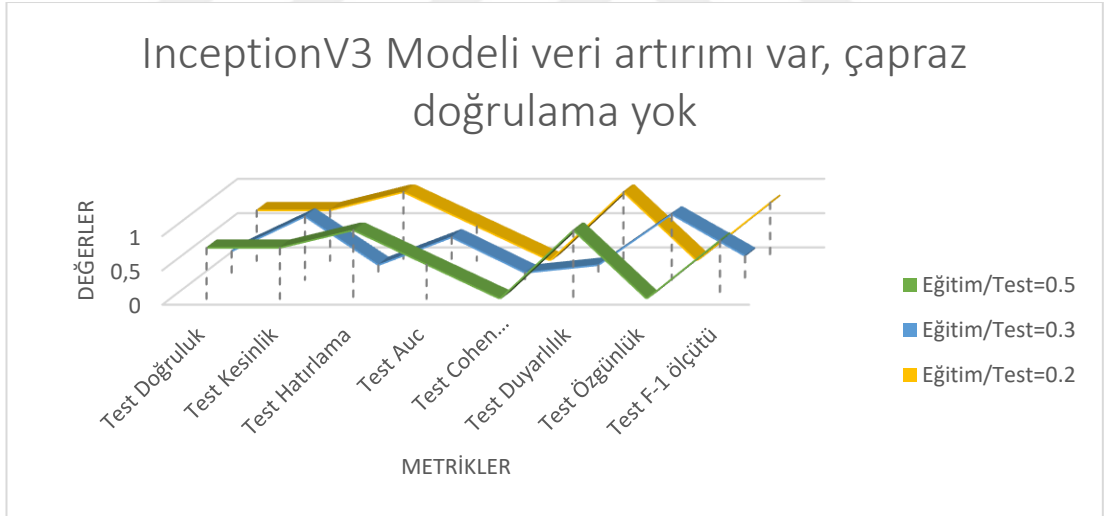
Şekil 4.38 InceptionV3 Modeline veri artırımı uygulanmış çapraz doğrulama uygulanmış üçlü grafik

Şekil 4.38’de InceptionV3 modelinin yaklaşık olarak 3 farklı sınama seti yönteminde de en iyi sonuçları 0,2 ve 0,5 vermiştir.



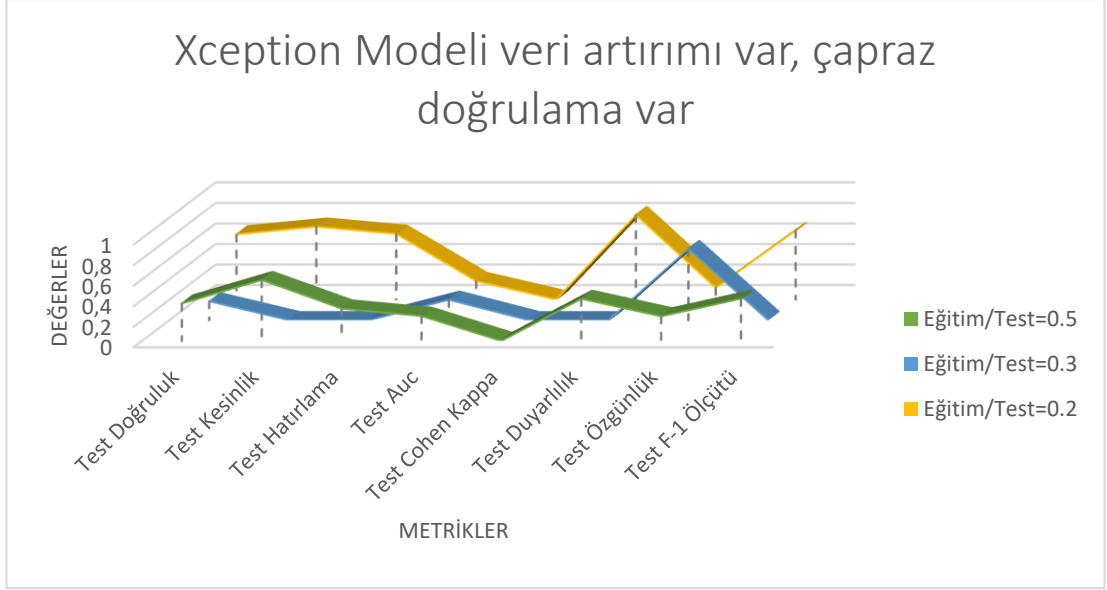
Şekil 4.39 InceptionV3 Modeline veri artırımı uygulanmamış çapraz doğrulama uygulanmamış üçlü grafik

Şekil 4.39’da InceptionV3 modelinin yaklaşık olarak 3 farklı sına seti yönteminde de en iyi sonucu 0,3 vermiştir.



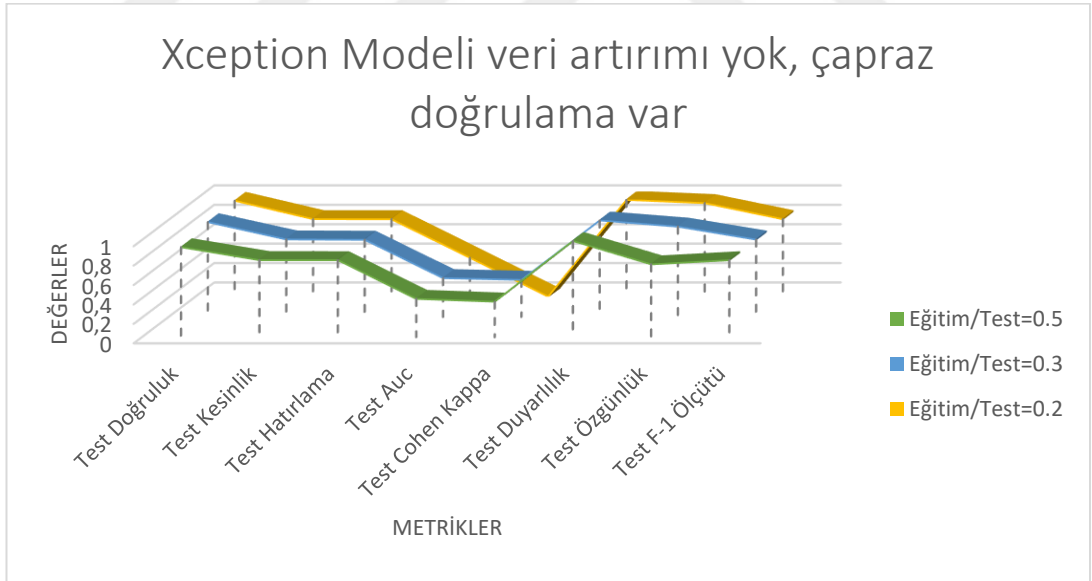
Şekil 4.40 InceptionV3 Modeline veri artırımı uygulanmış çapraz doğrulama uygulanmamış üçlü grafik

Şekil 4.40’da InceptionV3 modelinin yaklaşık olarak 3 farklı sına seti yönteminde de en iyi sonucu bulmak için hangi metriği baz alınır ona göre yorum yapılmalıdır.



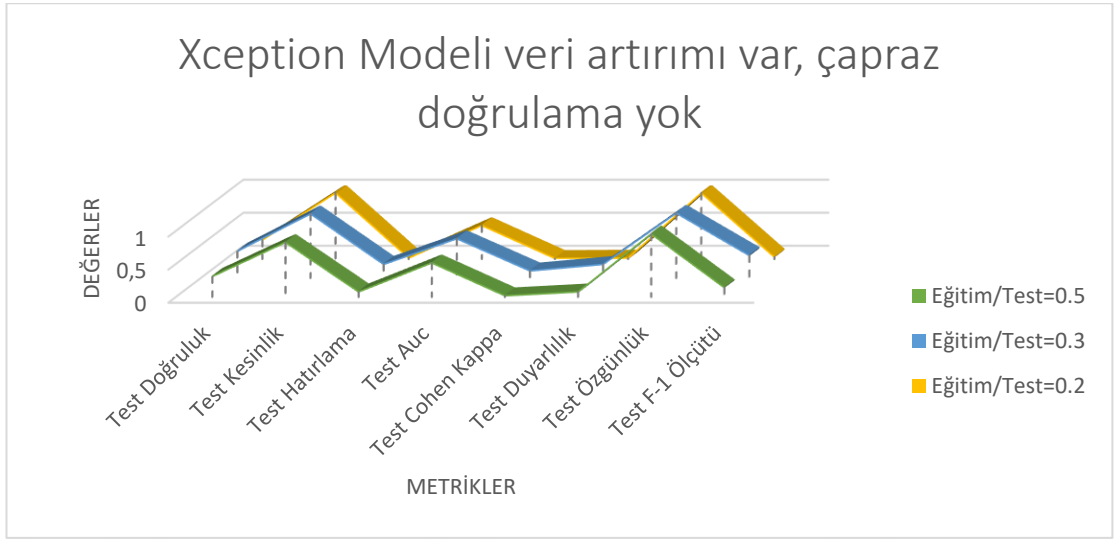
Şekil 4.41 Xception Modeline veri artırımı uygulanmış çapraz doğrulama uygulanmış üçlü grafik

Şekil 4.41’de Xception modelinin yaklaşık olarak 3 farklı sına seti yönteminde de en iyi sonucu 0,3 vermiştir.



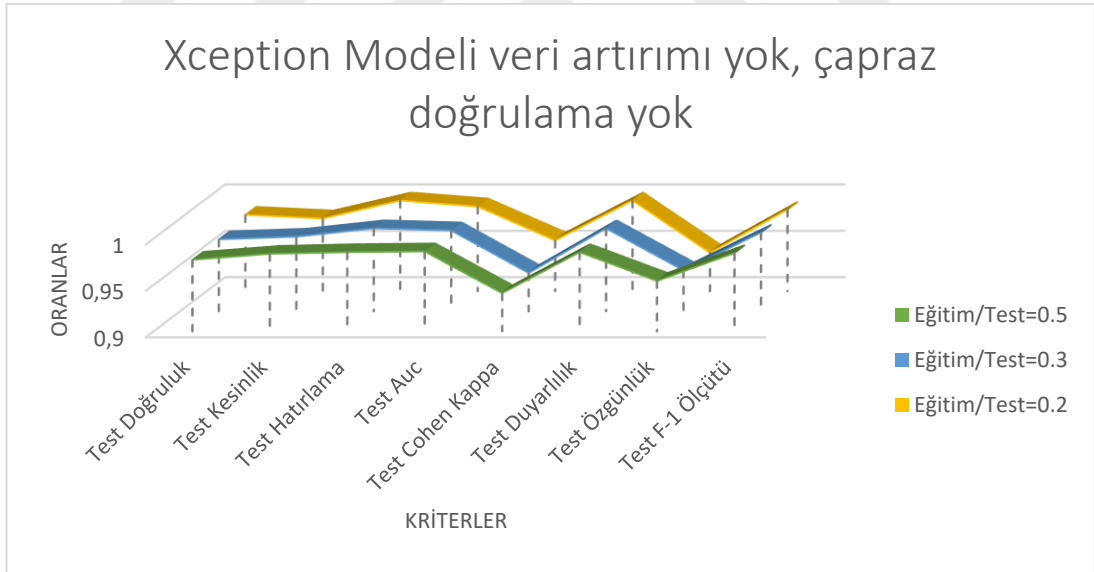
Şekil 4.42 Xception Modeline veri artırımı uygulanmamış çapraz doğrulama uygulanmış üçlü grafik

Şekil 4.42’de Xception modelinin yaklaşık olarak 3 farklı sına seti yönteminde de en iyi sonuçları 0,5 ve 0,2 olanlar vermiştir.



Şekil 4.43 Xception Modeline veri artırımı uygulanmış çapraz doğrulama uygulanmamış üçlü grafik

Şekil 4.43’de Xception modelinin yaklaşık olarak 3 farklı sınama seti yönteminde de aynı sonuçları vermiştir.



Şekil 4.44 Xception Modeline veri artırımı uygulanmamış çapraz doğrulama uygulanmamış üçlü grafik

Şekil 4.44’de Xception modelinin yaklaşık olarak 3 farklı sınama seti yönteminde de aynı sonuçları vermiştir.

5. SONUÇ VE ÖNERİLER

5.1 Öneriler

Veri kümesinin içeriğine ve içinde bulunan verilere göre değişebildiğinden her veri kümesi (yani veri kaynağı), çok ince araştırılarak çalışılması gerekir. Bu bölümde kullandığım veri kümesinin özelliğinden kaynaklanan bazı sorunlar ve neler yapılabileceğine dair önerilerin bulunduğu kısımdır.

1. Veri kümesi seçiminde daha dengeli dağılmış ya da homojenize olmuş veri kümesi/veri kümesiler seçilerek başlanabilir. Dengesiz veri setlerinde uygulanan SMOTE yöntemi bu veri kümesi içinde uygulanabilir.
2. Evrişimsel sinir ağları model seçilimi yapılırken: LeNet, DenseNet, RasNet, NasNet, ResNet-18, ResNet-50, SqueezeNet ya da EfficientNet kullanılabilir.
3. Ayrıca evrişimsel sinir ağları kullanımı daha yaygın olsada diğer derin öğrenme mimarisi olan LSTM de kullanılabilir.
4. Sonuçlar zatürre-normal şeklinde 2 'li (binary)sınıfa ayrılmış olup, sonuçlar 3'lü çoklu(multiclass)sınıfa ya da daha fazla ayrılarak işlem yapılabilir ve farklı sonuçlar elde edilebilir.
5. Modeller yerine öğrenim aktarımı (transfer learning) yapılarak veya modellere yeni katmanlar eklenerek kullanılabilir.
6. Kullandığım Kaggle Veri kümesinde Covid-19'u ayrıca ayırabilmek için Pnömoni (zatürre (4334)) verisini Bakteria(bakteri) strees-smooking+virüs olarak 2'ye ayrılabilir, daha sonraki aşamada Virüs kategorisi seçilerek, Virüs-Sars ve Covid-19 ayrımı yapılarak, Covid-19 verisi elde edilebilir.

5.2 Sonular

Veri kumesinde homojenize daėılım olmadıėından verilere birok iřlem uygulanmıřtır. Bunlar ierinde en nemli olanlarından biri dzeltmeye (reglarizasyon) yapmaya olanak veren veri artırımını olmuřtur. Bu sayede ezberleme (overfitting) durumuna karřı veri artırımını kullanımıdır. Bu alıřmada veri artırımını yapılmadan model alıřtırıldıėında %80/%20 eėitim/test oranında eėitim/test skorları olduka yksek geldiėinden ve hemen hemen birbirine ok yakın olduėundan hem bařarı metrikleri ok yksek alınmıř ve ezberleme (overfitting) gzlenmemiřtir. Bu sonular neticesinde, bu veri kumesi iin veri artırımını yapmaya gerek olmadıėı sylenebilir. Verinin yeteri kadar farklı olduėunu da sonuların yksek olmasından anlařılmıřtır. Bu veriye, veri artırımını yapılnca veriler ařırı farklılařma gstereceėinden dolayı, sonular daha dřk skor verecektir ki bu sonuları vermiřtir. Bu anlamda, bu veri kumesi iin veri artırımını yapmadan alıřtırmanın daha iyi olduėu gzlemlenmiřtir. Derin ėrenme modelleri olarak da InceptionV3, Xception, VGG-19, MobileNet, VGG-16 kullanılmıř olup en iyi sonular MobileNet’ de gzlenmiřtir. Sonular 2 sınıflı olduėu iin tm modellerde aktivasyon fonksiyonu olarak ‘softmax’ yerine deėiřtirilerek ‘sigmoid’ aktivasyon fonksiyonu seilmiřtir. alıřma sreleri olarak en hızlı VGG-16’da en yavař Xception ve MobileNet’de alıřmıřtır. Genelde tekrar sayısı (epoch) olarak 15, 20 ya da 30 seilse de bazı noktalarda erken durdurma (earlystopping) kullanılmıřtır. Birok derin ėrenme ktphanelerinden arasından Keras’ı seerek bu projede kullanıldı. Kullanmıř olduėum kolay ve efektif deėer sunan Keras’da optimizasyon olarak ‘Adam’ fonksiyonu seilmiřtir. Verileri Eėitim/test olarak sırasıyla %80, %70, %50 olarak denenmiřtir. En iyi sonular MobileNet modelinde, ‘Adam’ optimizasyon fonksiyonu, ‘sigmoid’ aktivasyon fonksiyonu, veri artırımını kapalı moddayken, Eėitim/test oranı %80/%20 parametreleriyle alınmıřtır. Doėruluk %98,30, Kesinlik%98,37, Hassasiyet%99,29, Auc %99,48, Cohen Kappa%95,74, Duyarlılık %99,29, zgnlk %95,71 ve F1-skoru %98,83, sonuları elde edilmiřtir. Ayrıca test kaybı da 0.073 olarak elde edilmiřtir.

Eđitim veri kümesinde daha fazla veri bulunduđu için apraz dođrulama olmayan veri sonuçları daha başarılı örnekler vermiştir.

apraz dođrulama olan verilerde dođrulama (validation) da uygulanmış olup en iyi sonuç Eđitim oranının en yüksek olanda incelenmiştir. Ayrıca apraz dođrulama uygulanarak, k değeri 5 olarak seçilmiş ve verilerin daha güvenilir çıkması için kullanılmıştır.



6. KAYNAKLAR

Bu tez çalışmasında Vancouver atıf sistemi kullanılmıştır.

1. “Global Health Estimates:Life expenctancy and leading causes of death and disability”[Internet].[ErişimTarihi:11Haziran2023]
<https://www.who.int/data/gho/data/themes/mortality-and-global-health-estimates>
2. “Zatürre(Pnömoni Nedir?) Zatürre Teşhis Ve Tedavisi Nasıl Yapılır?” [Internet]. [Erişim Tarihi:15Haziran2023].<https://www.memorial.com.tr/hastaliklar/zaturrepnomoni-nedir-zaturre-teshis-ve-tedavisi-nasil-yapilir>
3. “Covid-19 Bilgilendirme Platformu(Covid-19 Nedir?)” [Internet]. [Erişim Tarihi:10Haziran 2023].<https://covid19.saglik.gov.tr/TR-66300/covid-19-nedir-.html>
4. Darıcı M. Göğüs Kafesi Röntgen Görüntülerinde Derin Öğrenme Metoduyla Zatürre Hastalığının Tanısı(Yüksek Lisans Tezi). İstanbul Teknik Üniversitesi; 2020.
5. Üstündağ E.Evrişimli Sinir Ağları,Transfer Öğrenme Ve Destek Vektör Makineleri Kullanarak Akciğer Röntgen Görüntülerinden COvid-19 Sınıflandırılması,(Yüksek Lisans Tezi),Afyon Kocatepe Üniversitesi,2022 p.98.
6. Carlile M, Hurt B, Hsiao A, Hogarth M, Longhurst CA, Dameff C. Deployment of artificial intelligence for radiographic diagnosis of COVID-19 pneumonia in the emergency department. J Am Coll Emerg Physicians Open. 01 Aralık 2020;1(6):1459-64.
7. Kaggle.(2020)[Dataset]. CoronaHack -Chest X-Ray-Dataset. [Erişim Tarihi:11 Kasım 2020]. <https://www.kaggle.com/datasets/praveengovi/coronahack-chest-xraydataset>
8. Toğaçar M, Ergen B, Sertkaya ME, Programı BT, Meslek TB, Okulu Y, vd. Zatürre Hastalığının Derin Öğrenme Modeli ile Tespiti. C. 31, Fırat Üniversitesi Müh. Bil. Dergisi. 2019.
9. Ayan E.,Ünver H.M, Diagnosis of Pneumonia From Chest X-ray Images Using Deep learning(2019)IEE.sf:5
10. Chouhan V, Singh SK, Khamparia A, Gupta D, Tiwari P, Moreira C, vd. A novel transfer learning based approach for pneumonia detection in chest X-ray images. Applied Sciences (Switzerland). 02 Ocak 2020;10(2).
11. Farooq M,Hafeez A.,Covid-ResNet:A Deep Learning Framework For Screening Of Covid-19 From Radiographs(2020),sf:5.
12. Apostolopoulos ID, Mpesiana TA. Covid-19: automatic detection from X-ray images utilizing transfer learning with convolutional neural networks. Phys Eng Sci Med. 01 Haziran 2020;43(2):635-40. <https://doi.org/10.1007/s13246-020-00865-4>
13. Ozturk T, Talo M, Yildirim EA, Baloglu UB, Yildirim O, Rajendra Acharya U. Automated detection of COVID-19 cases using deep neural networks with X-ray images. Comput Biol Med. 01 Haziran 2020;121. <https://doi.org/10.1016/j.compbiomed.2020.103792>

14. Rahman T, Chowdhury MEH, Khandakar A, Islam KR, Islam KF, Mahbub ZB, vd. Transfer learning with deep Convolutional Neural Network (CNN) for pneumonia detection using chest X-ray. *Applied Sciences (Switzerland)*. 01 Mayıs 2020;10(9). <https://doi.org/10.3390/app10093233>
15. Brunese L, Mercaldo F, Reginelli A, Santone A. Explainable Deep Learning for Pulmonary Disease and Coronavirus COVID-19 Detection from X-rays. *Computer Methods Programs In Biomedicine* 01 Kasım 2020;196. <https://doi.org/10.1016/j.cmpb.2020.105608>
16. Albert N. Evaluation of Contemporary Convolutional Neural Network Architectures for Detecting COVID-19 from Chest Radiographs, College Of Engineering and Computer Science Syracuse University, NY, USA, 2020.
17. Oh Y, Park S, Ye JC. Deep Learning COVID-19 Features on CXR Using Limited Training Data Sets. *IEEE Trans Med Imaging*. 01 Ağustos 2020;39(8):2688-700.
18. V J S, D JF. Deep Learning Algorithm for COVID-19 Classification Using Chest X-Ray Images. *Comput Math Methods Med*. 2021;2021:9269173. <https://doi.org/10.1155/2021/9269173>
19. BOZKURT F. Derin Öğrenme Tekniklerini Kullanarak Akciğer X-Ray Görüntülerinden COVID-19 Tespiti. *European Journal of Science and Technology*. 15 Nisan 2021;149-56. <https://doi.org/10.31590/ejosat.898385>
20. Bozkurt F, Yağanoğlu M, Derin Evrişimli Sinir Ağları Kullanarak Akciğer X-Ray Görüntülerinden COVID-19 Tespiti, *Veri Bilimi*, Cilt 4, Sayı 2, 2021.
21. Yılmaz A. Diagnosing COVID-19 from X-Ray images with using multi-channel CNN architecture. *Journal of the Faculty of Engineering and Architecture of Gazi University*. 2021;36(4):1761-73. <https://doi.org/10.17341/gazimmfd.746883>
22. Narin A, İşler Y. Detection of new coronavirus disease from chest x-ray images using pre-trained convolutional neural networks. *Journal of the Faculty of Engineering and Architecture of Gazi University*. 2021;36(4):2095-107. <https://doi.org/10.17341/gazimmfd.827921>
23. Khan E, Rehman MZU, Ahmed F, Alfouzan FA, Alzahrani NM, Ahmad J. Chest X-ray Classification for the Detection of COVID-19 Using Deep Learning Techniques. *Sensors*. 01 Şubat 2022;22(3). <https://doi.org/10.3390/s22031211>
24. Cao Z, Huang J, He X, Zong Z. BND-VGG-19: A deep learning algorithm for COVID-19 identification utilizing X-ray images. *Knowl Based Syst*. 22 Aralık 2022;258. <https://doi.org/10.1016/j.knosys.2022.110040>
25. Prof.Dr. Diri B, Makine Öğrenmesine Giriş (Machine Learning -ML), Yıldız Teknik Üniversitesi, Bilgisayar Mühendisliği Bölümü, https://ytuce.maliayas.com/files/38dc2afc-Introduction_to_Machine_Learning_10.pdf
26. Bİlgin M, Makine Öğrenmesi, İstanbul, Papatya Yayıncılık Eğitim, 2018, xii,160sf, sf:14-17
27. Oğuz Ç, Covid-19 Olası Vakalarının Derin Öğrenme Teknikleri Kullanılarak Tespiti, (Yüksek Lisans Tezi), Atatürk Üniversitesi, 2021 sf:84.
28. Lecun Y, Bengio Y, Hinton G. Deep learning. *C*. 521, Nature. Nature Publishing Group; 2015. s. 436-44. <https://doi.org/10.1038/nature14539>

29. Yılmaz A, Kaya U, (2021), Derin Öğrenme (3. baskı). İstanbul: Inkilap Kitabevi Yayıncılık, sf:134
30. Arı A, Hanbay D. Bölgesel Evrişimsel Sinir Ağları Tabanlı MR Görüntülerinde Tümör Tespiti. Gazi Üniversitesi Mühendislik-Mimarlık Fakültesi Dergisi. 12 Eylül 2018;2018(18-2). <https://doi.org/10.17341/gazimmfd.460535>
31. Akdağlı E. “Evrişimsel Sinir Ağları(Convolutional Neural Network)” [Internet]. [Erişim Tarihi:13Haziran2023].<https://ece-akdagli.medium.com/evri%C5%9Fimsel-sinir-a%C4%9Flar%C4%B1-convolutional-neural-network-a6c9d8e666c4>
32. Hinton GE, Srivastava N, Krizhevsky A, Sutskever I, Salakhutdinov RR. Improving neural networks by preventing co-adaptation of feature detectors. 03 Temmuz 2012; Erişim adresi: <http://arxiv.org/abs/1207.0580>
33. Kızrak, A. (2019). “Derin Öğrenme İçin Aktivasyon Fonksiyonlarının Karşılaştırılması”[Internet][ErişimTarihi:11Haziran2023] <https://ayyucekizrak.medium.com/derin-%C3%B6%C4%9Frenme-i%C3%A7in-aktivasyon-fonksiyonlar%C4%B1n%C4%B1n-kar%C5%9F%C4%B1la%C5%9F%C4%B1r%C4%B1lmas%C4%B1-cee17fd1d9cd>.
34. Simonyan K, Zisserman A. Very Deep Convolutional Networks for Large-Scale Image Recognition. 04 Eylül 2014; Erişim adresi: <http://arxiv.org/abs/1409.1556>
35. Howard AG, Zhu M, Chen B, Kalenichenko D, Wang W, Weyand T, vd. MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications. 16 Nisan 2017; Erişim adresi: <http://arxiv.org/abs/1704.04861>
36. Sandler M, Howard A, Zhu M, Zhmoginov A, Chen LC. MobileNetV2: Inverted Residuals and Linear Bottlenecks. İçinde: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition. IEEE Computer Society; 2018. s. 4510-20.
37. Szegedy C, Liu W, Jia Y, Sermanet P, Reed S, Anguelov D, vd. Going Deeper with Convolutions. 16 Eylül 2014; Erişim adresi: <http://arxiv.org/abs/1409.4842>
38. Chollet F. Xception: Deep Learning with Depthwise Separable Convolutions. 07 Ekim 2016; Erişim adresi: <http://arxiv.org/abs/1610.02357>
39. Koç Kaan O,Python Üzerinden Derin Öğrenme Algoritmaları Kullanılarak Veri Görüntüsünden Cilt Hastalıklarının Tespit edilmesi,(Yüksek Lisans Tezi),Bolu Abant İzzet Baysal Üniversitesi,2019,sf:52
40. Kilic S. Kappa test. Journal of Mood Disorders. 2015;5(3):142. <https://doi.org/10.5455/jmood.20150920115439>