

T.C.
YILDIZ TEKNİK ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

DERİN ÖĞRENME YÖNTEMİYLE X-IŞINI GÖRÜNTÜLERİNDEN
YAYGIN GÖĞÜS HASTALIKLARININ TESPİTİ

Tolga Saim BAŞÇETİN

DOKTORA TEZİ

Matematik Mühendisliği Anabilim Dalı

Matematik Mühendisliği Programı

Danışman

Prof. Dr. İbrahim EMİROĞLU

Ağustos, 2021

T.C.
YILDIZ TEKNİK ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

DERİN ÖĞRENME YÖNTEMİYLE X-IŞINI GÖRÜNTÜLERİNDEN
YAYGIN GÖĞÜS HASTALIKLARININ TESPİTİ

Tolga Saim BAŞÇETİN tarafından hazırlanan tez çalışması 10.08.2021 tarihinde aşağıdaki jüri tarafından Yıldız Teknik Üniversitesi Fen Bilimleri Enstitüsü Matematik Mühendisliği Anabilim Dalı, Matematik Mühendisliği Programı **DOKTORA TEZİ** olarak kabul edilmiştir.

Prof. Dr. İbrahim EMİROĞLU
Yıldız Teknik Üniversitesi
Danışman

Jüri Üyeleri

Prof. Dr. İbrahim EMİROĞLU, Danışman

Yıldız Teknik Üniversitesi

Prof. Dr. Mustafa SIVRİ, Üye

Yıldız Teknik Üniversitesi

Prof. Dr. Coşkun GÜLER, Üye

Yıldız Teknik Üniversitesi

Dr. Öğr. Üyesi Aysun GÜRAN, Üye

Doğuş Üniversitesi

Dr. Öğr. Üyesi Mustafa Zahid GÜRBÜZ, Üye

Doğuş Üniversitesi

Danışmanım Prof. Dr. İbrahim EMİROĞLU sorumluluğunda tarafımca hazırlanan Derin Öğrenme Yöntemiyle X-Işını Görüntülerinden Yaygın Göğüs Hastalıklarının Tespiti başlıklı çalışmada kaynaklardan aldığım bilgileri ana metin ve referanslarda eksiksiz gösterdiğimi, araştırma verilerine ve sonuçlarına ilişkin çarpıtma ve/veya sahtecilik yapmadığımı, çalışmam süresince bilimsel araştırma ve etik ilkelerine uygun davrandığımı beyan ederim. Beyanımın aksinin ispatı halinde her türlü yasal sonucu kabul ederim.

Tolga Saim BAŞÇETİN

İmza



Bu çalışma, Yıldız Teknik Üniversitesi Bilimsel Araştırma Proje Koordinatörlüğü'nün 2015-07-03-DOP01 numaralı projesi ile desteklenmiştir.

TEŐEKKÜR

Çalıőmam süresince sabırla bana destek olan sevgili aileme ve özellikle canım eőim Çiğdem BAŐÇETİN'e, yardımını ve bilgisini esirgemeyen tez danışmanım sayın Prof. Dr. İbrahim EMİROĐLU'na teşekkürü borç bilirim.

Tolga Saim BAŐÇETİN



İÇİNDEKİLER

SİMGE LİSTESİ	viii
KISALTMA LİSTESİ	ix
ÖZET	xvi
ABSTRACT	xviii
1 GİRİŞ	1
1.1 Literatür Özeti	1
1.2 Tezin Amacı	6
1.3 Hipotez	6
2 DERİN ÖĞRENME METOTLARI	7
2.1 Yapay Sinir Ağları.....	7
2.2 Gözetimli Öğrenme.....	9
2.3 İkili Sınıflandırma	9
2.4 Lojistik Regresyon	9
2.5 Kayıp ve Maliyet Fonksiyonu.....	10
2.6 Dereceli Azalma	11
2.7 Tek Katmanlı Yapay Sinir Ağı Modeli	12
2.8 İki Katmanlı Yapay Sinir Ağı Modeli	13
2.9 Aktivasyon Fonksiyonları	19
2.9.1 Sigmoid Fonksiyonu	19
2.9.2 Tanjant Hiperbolik Fonksiyonu	20
2.9.3 ReLU (Rectified Linear Unit) Fonksiyonu	21
2.9.4 Sızıntı (Leaky) ReLU Fonksiyonu.....	22
2.10 Evrişimsel Sinir Ağları	22
2.10.1 Dikey Sınır Belirleme	25
2.10.2 Dolgulama (Padding).....	29

2.10.3 Kaydırma (Stride)	30
2.10.4 Renkli Resimlerde Evrişim	32
2.10.5 Tek Katmanlı Evrişim İşlemi.....	36
2.10.6 Tek Katmanlı Evrişim İçin Parametre Sayısı	39
2.10.7 Tek Katmanlı Evrişimli Sinir Ağı.....	40
2.10.8 Havuzlama (Pooling) Katmanı	41
2.11 Klasik Evrişimli Sinir Ağları.....	43
2.11.1 LeNet – 5.....	43
2.11.2 AlexNet.....	44
2.11.3 VGG 16 ve VGG 19	46
2.11.4 ResNets.....	47
2.11.5 Ağ İçinde Ağ (Network in Network)	49
2.11.7 Transfer Öğrenme.....	55
2.11.8 MobileNet.....	57
2.11.9 EfficientNet.....	61
3 DERİN ÖĞRENMENİN RÖNTGEN GÖRÜNTÜLERİNE UYGULANMASI	63
3.1 Medikal Resimlerde Hastalık Tespiti.....	63
3.2 Göğüs Röntgenlerinde Hastalık Tespiti	66
3.3 Sınıf Dengesizliği (Class Imbalance)	68
3.4 Çoklu Sınıflandırma (Multi- Task)	72
3.5 Veri Seti Büyüklüğü.....	74
3.6 Medikal Görüntüler İçin Veri Artırma (Data Augmentation).....	75
3.7 Model İçin Test ve Eğitim Verisi Oluşturma	76
3.8 Model İçin Değerlendirme Metrikleri.....	78
3.8.1 Doğruluk (Accuracy)	78
3.8.2 Duyarlılık, Özgüllük ve Yaygınlık.....	80
3.8.3 Pozitif Prediktif Değer (PPD) ve Negatif Prediktif Değer (NPD)	81
3.8.4 Karışıklık Matrisi (Confusion Matrix)	82
3.8.5 AUC-ROC Eğrisi (Eğri Altındaki Alan-Alıcı işlem karakteristikleri / Receiver Operating Characteristic)	84

3.8.6 Güven Aralığı (Confidence Interval).....	87
4 VERİ KÜMESİ VE MODEL SONUÇLARI	89
4.1 Çalışılan Veri Kümesi.....	89
4.2 Geliştirilen Modelin Yapısı.....	104
4.2.1 Optimizasyon Algoritması Seçimi	105
4.2.2 Öğrenme Hızı ve Yığın Boyutu	105
4.2.3 Giriş Resminin Boyutu	105
4.3 Geliştirilen Modelin Sonuçları.....	109
4.3.1 Atelektazi	110
4.3.2 Kardiyomegali	111
4.3.3 Efüzyon	112
4.3.4 İnfiltrasyon (Infiltration).....	113
4.3.5 Kütle (Mass).....	114
4.3.6 Nodül (Nodule)	115
4.3.7 Zatürre (Pneumonia).....	116
4.3.8 Pnömotoraks (Pneumothorax).....	117
4.3.9 Konsolidasyon (Consolidation).....	118
4.3.10 Ödem (Edema)	119
4.3.11 Amfizem (Emphysema)	120
4.3.12 Fibroz (Fibrosis).....	121
4.3.13 Plevral Kalınlaşma (Pleural Thickening)	122
4.3.14 Fıtık (Hernia)	123
5 SONUÇ VE ÖNERİLER	126
5.1 Sonuçların Değerlendirilmesi.....	126
KAYNAKÇA	128
TEZDEN ÜRETİLMİŞ YAYINLAR	134

SİMGE LİSTESİ

S	Adım sayısı
w	Ağırlıklar
H	Boy
D	Derinlik
F×F	Filtre boyutu
K	Filtre sayısı
W	Genişlik
W×H×D	Girdi boyutu
P	Sıfır doldurma miktarı
q(x)	Tahmin edilen çıkışın gerçek değeri
p(x)	Tahmin edilen çıkışın olasılığı

KISALTMA LİSTESİ

AMESA	Adapte Edilmiş MobileNet Evrişimli Sinir Ağı
ESA	Evrişimli Sinir Ağı
FN	Yanlış Negatif
FP	Yanlış Pozitif
GYA	Geri Yayılım Algoritması
NIH	Amerika Birleşik Devletleri Ulusal Sağlık Enstitüsü
ReLU	Düzleştirilmiş Doğrusal Birim
SVM	Destek Vektör Makinesi
TN	Doğru Negatif
TP	Doğru Pozitif
YSA	Yapay Sinir Ağı

ŞEKİL LİSTESİ

Şekil 2. 1 Örnek Beyin Hücresi [25]	7
Şekil 2. 2 Perceptron [26]	8
Şekil 2. 3 Sigmoid Eğrisi	10
Şekil 2. 4 Dereceli Azalma [29]	11
Şekil 2. 5 Tek Katmanlı Yapay Sinir Ağı Modeli [29]	12
Şekil 2. 6 Yapay Sinir Ağı Kayıp Fonksiyonu [29]	13
Şekil 2. 7 İki Katmanlı Yapay Sinir Ağı [29]	13
Şekil 2. 8 Tek Katmanlı Sinir Ağında Hesaplama [29]	15
Şekil 2. 9 Bir Gizli Katmanlı Sinir Ağı Örneği [29]	15
Şekil 2. 10 Sigmoid Fonksiyonu [29]	19
Şekil 2. 11 Tanjant Aktivasyon Fonksiyonu [29]	20
Şekil 2. 12 ReLU Aktivasyon Fonksiyonu [29]	21
Şekil 2. 13 Sızıntı (Leaky) ReLU Aktivasyon Fonksiyonu [29]	22
Şekil 2. 14 64x64x3 Boyutunda Renkli Resim [29]	23
Şekil 2. 15 Çok Katmanlı Klasik Sinir Ağı [29]	23
Şekil 2. 16 1000x1000x3 Boyutunda Renkli Resim [29]	23
Şekil 2. 17 İki Boyutlu Evrişim Örneği [34]	24
Şekil 2. 18 Evrişim ile Özellik Belirleme [34]	24
Şekil 2. 19 Bir Resimdeki Yatay ve Dikey Sınırlar [34]	25
Şekil 2. 20 Dikey Kenar Belirleme Filtresi [34]	25
Şekil 2. 21 Aydınlik ve Karanlık Taraflı Örnek Resim [34]	26
Şekil 2. 22 Dikey Kenar Evrişim İşlemi [34]	26
Şekil 2. 23 Dikey Kenar Evrişim İşlemi 2 [34]	27
Şekil 2. 24 Sobel Filtre İşlemi İçin Giriş Resmi [34]	27
Şekil 2. 25 Dikey ve Yatay Eksen Sobel Filtreleri [34]	28
Şekil 2. 26 Dolgulama İşlemi Yapılmış Evrişim İşlemi [34]	29
Şekil 2. 27 Evrişim İşlemi [34]	30
Şekil 2. 28 İki Kaydırmalı Evrişim İşlemi [34]	31
Şekil 2. 29 İki Kaydırmalı Evrişim İşlemi Sonucu [34]	31
Şekil 2. 30 İki Boyutlu Siyah-Beyaz ve Üç Boyutlu Renkli Resim Örneği [34]	32
Şekil 2. 31 6x6x3 Boyutlu Renkli Resim Örneği [34]	33
Şekil 2. 32 Renkli Resmin Evrişim Örneği [34]	33

Şekil 2. 33 Renkli Resmin Evrişim Hesaplama Örneği [34]	34
Şekil 2. 34 Renkli Resmin Evrişim Hesaplama Örneği 2 [34].....	34
Şekil 2. 35 Renkli Resmin Evrişim Hesaplama Örneği 3 [34].....	35
Şekil 2. 36 Renkli Resmin Evrişim Hesaplama Örneği 4	35
Şekil 2. 37 Renkli Resme Uygulanan Evrişim Görseli [34]	37
Şekil 2. 38 Evrişimli Sinir Ağı İçin Girdi Karşılaştırması [34]	37
Şekil 2. 39 Evrişimli Sinir Ağı İçin Ağırlık Parametresi Karşılaştırması [34]	38
Şekil 2. 40 Tek Katmanlı Evrişimli Sinir Ağı Görseli	38
Şekil 2. 41 Tek Katmanlı Evrişimli Sinir Ağı Vektörel Gösterimi 1	39
Şekil 2. 42 Tek Katmanlı Evrişimli Sinir Ağı Vektörel Gösterimi 2	39
Şekil 2. 43 Evrişimli Sinir Ağı Filtresi İçin Parametre Sayısı Hesaplama 1	40
Şekil 2. 44 Evrişimli Sinir Ağı Filtresi İçin Parametre Sayısı Hesaplama 2 [34]	40
Şekil 2. 45 Evrişim İşlemi Örneği [34]	41
Şekil 2. 46 Tek Katmanlı Evrişimli Sinir Ağı Örneği [34]	41
Şekil 2. 47 Tek Katmanlı Evrişimli Sinir Ağı Örneği [40]	42
Şekil 2. 48 Ortalama Havuzlama Örneği [40].....	43
Şekil 2. 49 LeNet 5 Evrişimli Sinir Ağı Yapısı [34]	43
Şekil 2. 50 AlexNet Evrişimli Sinir Ağı Modeli [34]	44
Şekil 2. 51 AlexNet Evrişimli Sinir Ağı Şeması [34].....	45
Şekil 2. 52 VGG 16 Evrişimli Sinir Ağı Modeli [34]	46
Şekil 2. 53 VGG 16-19 Evrişimli Sinir Ağı Modelleri 2 [34]	47
Şekil 2. 54 Ana Yol (Main Path) [43]	47
Şekil 2. 55 Ana Yol ve Kısa Yol Aktarımı [43]	48
Şekil 2. 56 Düz Ağ Yapısı (Plain Network)	48
Şekil 2. 57 Artık Eklenmiş Ağ Yapısı (Residual Network).....	48
Şekil 2. 58 Düz Ağ Yapısında Katman Sayısı ile Eğitim Hatası İlişkisi [34]	49
Şekil 2. 59 ResNet Ağ Yapısında Katman Sayısı ile Eğitim Hatası İlişkisi [34].....	49
Şekil 2. 60 1x1 Evrişim İşlemi [34]	50
Şekil 2. 61 6x6x32 Boyutlu Resim İle 1x1 Filtrenin Evrişim İşlemi [34]	50
Şekil 2. 62 Çok Kanallı 3 Boyutlu Resim İle 1x1 Filtrenin Evrişim İşlemi [34].....	51
Şekil 2. 63 Inception Ağ Yapısı [34]	52
Şekil 2. 64 Inception Ağı İçin Parametre Sayısı Hesaplama [34]	53
Şekil 2. 65 Inception Ağı İçin Parametre Sayısı Hesaplama 2 [34].....	53
Şekil 2. 66 Inception Ağı Başlangıç Modülü [34]	54
Şekil 2. 67 Inception Ağı Yapısı [34].....	55

Şekil 2. 68 1000 Sınıflı Sinir Ağı [34]	56
Şekil 2. 69 Transfer Learning [34]	56
Şekil 2. 70 Transfer Learning Küçük Veri Seti [34]	56
Şekil 2. 71 Transfer Learning Büyük Veri Seti [34]	57
Şekil 2. 72 Renkli Resmin Evrişim Hesaplama Örneği [34]	58
Şekil 2. 73 Derinlemesine Ayrılabilir Evrişim [34]	58
Şekil 2. 74 Derinlemesine Ayrılabilir Evrişim 2 [34]	58
Şekil 2. 75 Noktasal (Pointwise) Evrişim [34]	59
Şekil 2. 76 Noktasal (Pointwise) Evrişim 2 [34]	59
Şekil 2. 77 MobileNet 1. Versiyonu [34]	60
Şekil 2. 78 MobileNet 2. Versiyonu [34]	60
Şekil 2. 79 MobileNet 2. Versiyon Blok Yapısı [34]	61
Şekil 2. 80 EfficientNet: Parametre Ayarlama [34]	62
Şekil 2. 81 EfficientNet Performans Karşılaştırması [50]	62
Şekil 3. 1 Deri Tümörü İçin Yapay Öğrenme [51]	63
Şekil 3. 2 Deri Tümörü İçin Yapay Öğrenme Modeli [51]	64
Şekil 3. 3 Deri Tümörü (Melanoma) Tahmin Karşılaştırması	64
Şekil 3. 4 Retinal Fundus Hastalığı [51]	65
Şekil 3. 5 Retinal Fundus Hastalık Tespiti [51]	65
Şekil 3. 6 Doku Hastalık Tespiti [54]	66
Şekil 3. 7 Göğüs Röntgeni [51]	66
Şekil 3. 8 Kitle Bulunan-Normal Göğüs Röntgenleri [51]	67
Şekil 3. 9 Göğüs Hastalıkları için Model Eğitimi [51]	67
Şekil 3. 10 Göğüs Hastalıkları için Model Çıktısı [51]	68
Şekil 3. 11 Kayıp Fonksiyonu İçin Model Çıktısı 1 [51]	69
Şekil 3. 12 Kayıp Fonksiyonu İçin Model Çıktısı 2 [51]	69
Şekil 3. 13 Çoklu Sınıflandırma	72
Şekil 3. 14 Veri Seti Karşılaştırması [51]	74
Şekil 3. 15 Veri Seti Boyutu Sorunu İçin Transfer Öğrenmesi [51]	74
Şekil 3. 16 Veri Seti Boyutu Sorunu İçin Transfer Öğrenmesi 2 [51]	75
Şekil 3. 17 Göğüs Röntgeni İçin Veri Artırma [51]	75
Şekil 3. 18 Veri Artırma Örneği – Kontrast Değişimi [51]	76
Şekil 3. 19 Veri Artırma Örneği – Dikey Eksende Çevirme [51]	76
Şekil 3. 20 Test -Eğitim Verisi [20]	77
Şekil 3. 21 Bir Hastaya Ait İki Farklı Röntgen Görüntüsü [51]	77

Şekil 3. 22 Bir Hastaya Ait Aynı Resimleri Ayırma [51].....	78
Şekil 3. 23 Karışıklık Matrisi 3.....	84
Şekil 3. 24 ROC Eğrisi İçin Model [51]	85
Şekil 3. 25 AUC- ROC Eğrisi Karşılaştırmaları	85
Şekil 3. 26 Eşik Değeri Gösterimi [51].....	86
Şekil 3. 27 Eşik Değeri Gösterimi 2 [51]	86
Şekil 3. 28 Eşik Değeri Gösterimi 3.....	87
Şekil 3. 29 Popülasyon Doğruluğu [51].....	87
Şekil 3. 30 Popülasyon İçin Örneklem Seçimi 1 [51]	88
Şekil 3. 31 Popülasyon İçin Örneklem Seçimi 2 [51]	88
Şekil 4. 1 Veri Setindeki Hastalıkların Dağılımı.....	90
Şekil 4. 2 Cinsiyete Göre Hastalıkların Dağılımı.....	91
Şekil 4. 3 Cinsiyete Göre Yaş Dağılımı.....	91
Şekil 4. 4 Hastalıkların Yaş ve Cinsiyete Göre Dağılımı -1	92
Şekil 4. 5 Hastalıkların Yaş ve Cinsiyete Göre Dağılımı -2	93
Şekil 4. 6 Hastaların Takip Sayılarına Göre Dağılımı	94
Şekil 4. 7 Tekil ve Çoklu Hastalıkların Ölüm Dağılımı	95
Şekil 4.8 Hastalıkların Birbirleri İle Olan Etkileşimi [4]	95
Şekil 4.9 Kardiyomegali 'nin Eşlik Ettiği Hastalık Dağılımı	96
Şekil 4. 10 Amfizemin Eşlik Ettiği Hastalık Dağılımı.....	96
Şekil 4. 11 Efüzyon 'un Eşlik Ettiği Hastalık Dağılımı.....	97
Şekil 4. 12 Nodülün Eşlik Ettiği Hastalık Dağılımı	97
Şekil 4. 13 Pnömotoraksın Eşlik Ettiği Hastalık Dağılımı.....	98
Şekil 4. 14 Atelektazi 'nin Eşlik Ettiği Hastalık Dağılımı.....	98
Şekil 4. 15 Plevral Kalınlaşmanın Eşlik Ettiği Hastalık Dağılımı	99
Şekil 4. 16 Kütlenin Eşlik Ettiği Hastalık Dağılımı	99
Şekil 4. 17 Ödemın Eşlik Ettiği Hastalık Dağılımı	100
Şekil 4. 18 Konsolidasyonun Eşlik Ettiği Hastalık Dağılımı.....	100
Şekil 4. 19 İnfiltration 'un Eşlik Ettiği Hastalık Dağılımı	101
Şekil 4. 20 Fibroz 'un Eşlik Ettiği Hastalık Dağılımı	101
Şekil 4. 21 Zatürrenin Eşlik Ettiği Hastalık Dağılımı.....	102
Şekil 4. 22 8 Adet Göğüs Hastalığınının X-Işını Gösterimi	102
Şekil 4. 23 Veri Setinden Alınan Örnek Resimler	103
Şekil 4. 24 Geliştirilen Modelin Adımları.....	104
Şekil 4. 25 Optimizasyon Algoritmaları Karşılaştırması	105

Şekil 4. 26 Geliştirilen Model Yapısı	106
Şekil 4. 27 MobileNet Mimarisi [64].....	108
Şekil 4. 28 Adapte Edilmiş MobileNet Gösterimi.....	108
Şekil 4. 29 Atelektazi Tanısı İçin Röntgen Görüntüsü [65].....	110
Şekil 4. 30 Kardiyomegali Tanısı İçin Röntgen Görüntüsü [65]	111
Şekil 4. 31 Efüzyon Tanısı İçin Röntgen Görüntüsü [65]	112
Şekil 4. 32 İnfiltrasyon Tanısı İçin Röntgen Görüntüsü [65]	113
Şekil 4. 33 Kütle Tanısı İçin Röntgen Görüntüsü.....	114
Şekil 4. 34 Nodül Tanısı İçin Röntgen Görüntüsü [65].....	115
Şekil 4. 35 Zatürre Tanısı İçin Röntgen Görüntüsü [65].....	116
Şekil 4. 36 Pnömotoraks Tanısı İçin Röntgen Görüntüsü [65].....	117
Şekil 4. 37 Konsolidasyon Tanısı İçin Röntgen Görüntüsü [65]	118
Şekil 4. 38 Ödem Tanısı İçin Röntgen Görüntüsü.....	119
Şekil 4. 39 Amfizem Tanısı İçin Röntgen Görüntüsü.....	120
Şekil 4. 40 Fibroz Tanısı İçin Röntgen Görüntüsü.....	121
Şekil 4. 41 Plevral Kalınlaşma Tanısı İçin Röntgen Görüntüsü.....	122
Şekil 4. 42 Fıtık Tanısı İçin Röntgen Görüntüsü	123

TABLO LİSTESİ

Tablo 2. 1 Evrişim Sonrası Çıkış Boyutu Hesaplama [34].....	30
Tablo 2. 2 Kaydırma Adımlı Evrişim Sonrası Çıkış Boyutu Tablosu [34]	32
Tablo 3. 1 Dengesiz Sınıf Dağılımı Örneği [51].....	70
Tablo 3. 2 Dengesiz Sınıf Dağılımı Örneği 2 [51].....	70
Tablo 3. 3 Dengesiz Sınıf Dağılımı Örneği 3 [51].....	71
Tablo 3. 4 Dengesiz Sınıf Dağılımı Örneği 4 [51].....	72
Tablo 3. 5 Çoklu Sınıflandırma 2 [51].....	73
Tablo 3. 6 Çoklu Sınıflandırma 3.....	73
Tablo 3. 7 Doğruluk Metriği İçin Örnek Hesaplama [51]	79
Tablo 3. 8 Örnek Model [51].....	81
Tablo 3. 9 Örnek Model 2 [51].....	82
Tablo 3. 10 Karışıklık Matrisi 1 [51]	83
Tablo 3. 11 Karışıklık Matrisi 2 [51]	83
Tablo 4. 1 Atelektazi Sınıfı İçin AUC Değer Karşılaştırması	110
Tablo 4. 2 Kardiyomegali Sınıfı İçin AUC Değer Karşılaştırması.....	111
Tablo 4. 3 Efüzyon Sınıfı İçin AUC Değer Karşılaştırması.....	112
Tablo 4. 4 İnfiltrasyon Sınıfı İçin AUC Değer Karşılaştırması.....	113
Tablo 4. 5 Kütle Sınıfı İçin AUC Değer Karşılaştırması.....	114
Tablo 4. 6 Nodül Sınıfı İçin AUC Değer Karşılaştırması	115
Tablo 4. 7 Zatürre Sınıfı İçin AUC Değer Karşılaştırması	116
Tablo 4. 8 Pnömotoraks Sınıfı İçin AUC Değer Karşılaştırması	117
Tablo 4. 9 Konsolidasyon Sınıfı İçin AUC Değer Karşılaştırması.....	118
Tablo 4. 10 Ödem Sınıfı İçin AUC Değer Karşılaştırması.....	119
Tablo 4. 11 Amfizem Sınıfı İçin AUC Değer Karşılaştırması.....	120
Tablo 4. 12 Fibroz Sınıfı İçin AUC Değer Karşılaştırması.....	121
Tablo 4. 13 Plevral Kalınlaşma Sınıfı İçin AUC Değer Karşılaştırması.....	122
Tablo 4. 14 Fıtık Sınıfı İçin AUC Değer Karşılaştırması.....	123
Tablo 4. 15 Tüm Sınıflar İçin AUC Değer Karşılaştırması 1	124
Tablo 4. 16 Tablo 4. 16 Tüm Sınıflar İçin AUC Değer Karşılaştırması 2.....	124
Tablo 4. 17 Tablo 4. 16 Tüm Sınıflar İçin AUC Değer Karşılaştırması 3.....	125

DERİN ÖĞRENME YÖNTEMİYLE X-IŞINI GÖRÜNTÜLERİNDEN YAYGIN GÖĞÜS HASTALIKLARININ TESPİTİ

Tolga Saim BAŞÇETİN

Matematik Mühendisliği Anabilim Dalı

Doktora Tezi

Danışman: Prof. Dr. İbrahim EMİROĞLU

Son yıllarda yaşadığımız pandemi ve bu salgına yol açan Kovid-19 virüsünün özellikle akciğerlere zarar vermesi ve bu tahribatın ve hastalık tanısının x-ray ışınları görüntüleri ile yapılabilmesi nedeni ile dikkatimizi daha çok göğüs hastalıkları ve özellikle akciğer hastalıklarına ve bu hastalıkların derin öğrenme metotları ile tespit edilmesi üzerinde yoğunlaştırdık. Yaptığımız çalışmada Amerikan Ulusal Sağlık Enstitüsüne ait Göğüs röntgen görüntüleri kullanılmıştır. Bu veri seti yaygın göğüs hastalıklarından oluşmaktadır. Bu hastalıklar Atelektazi, Kardiyomegali, Konsolidasyon, Ödem, Efüzyon, Amfizem, Fibroz, İnfiltrasyon, Kütle, Nodül, Plevral Kalınlaşma, Zatürre, Fıtık ve Pnömotoraks şeklindedir. Yaptığımız çalışmada ana hatlarını bu hastalıkların oluşturduğu hastalıkların teşhisleri yapılmıştır. Amerikan Ulusal Sağlık Enstitüsünün yayınladığı veri setinde birbirinden farklı 30805 hastanın 112120 adet röntgen görüntüsü bulunmaktadır [1]. Göğüs röntgen görüntülerini okumak uzmanlar için kolay gibi görünse de arkada yatan fizyolojik ve patolojik ilkeler ile anatomik yapılar, bu işlemi yoğun bilgi gerektiren zor ve karmaşık bir problem haline getirmektedir. Bu tür faktörler, göğüs röntgeni görüntülerini okumak için tutarlı ve otomatik bir teknik geliştirmenin zorluğunu artırırken, aynı zamanda tüm yaygın göğüs hastalıklarını da dikkate alır.

Uygulamanın mobil ve düşük kapasiteli bilgisayarlarda da etkili çalışabilmesi için temel yapı olarak MobileNet ve EfficientNet gibi mimariler temel alınarak yeni bir model geliştirilmiştir. Geliştirdiğimiz modeller ile elde ettiğimiz AUC değerleri şöyledir,

AMESA modelinde çalıştığımız sınıflar için elde edilen AUC değerleri, Atelektazi 0.76, Kardiyomegali 0.93, Konsolidasyon 0.70, Ödem 0.89, Efüzyon 0.85, Amfizem 0.83, Fibroz 0.80, İnfiltrasyon 0.68, Kütle 0.79, Nodül 0.67, Plevral Kalınlaşma 0.68, Zatürre 0.705, Pnömotoraks 0.870 ve Fıtık 0.69 dur.

ESA 1 modelinde çalıştığımız sınıflar için elde edilen AUC değerleri ise, Atelektazi 0.779, Kardiyomegali 0.92, Konsolidasyon 0.721, Ödem 0.875, Efüzyon 0.896, Amfizem 0.77, Fibroz 0.798, İnfiltrasyon 0.712, Kütle 0.802, Nodül 0.745, Plevral Kalınlaşma 0.778, Zatürre 0.705, Pnömotoraks 0.847 ve Fıtık 0.65 tir.

Anahtar Kelimeler: Derin Öğrenme, Evrişimli Sinir Ağı, Göğüs Hastalıkları Tespiti, Yapay Sinir Ağı, X-Işını Görüntüsü, Röntgen Görüntüsü

DETECTION OF COMMON THORAX DISEASES FROM X-RAY IMAGES WITH DEEP LEARNING METHOD

Tolga Saim BAŞÇETİN

Department of Mathematical Engineering

Doctor of Philosophy Thesis

Supervisor: Prof. Dr. İbrahim EMİROĞLU

Since the pandemic we have lived in in recent years and the Kovid-19 virus that caused this epidemic especially damage the lungs and this damage and disease diagnosis can be made with x-ray images, we paid more attention to chest diseases and especially lung diseases and to detect these diseases with deep learning methods. we focused on. Chest X-ray images of the American National Institute of Health were used in our study. This dataset consists of common chest diseases. These diseases are Atelectasis, Cardiomegaly, Consolidation, Edema, Effusion, Emphysema, Fibrosis, Infiltration, Mass, Nodule, Pleural Thickening, Pneumonia, Hernia and Pneumothorax. In our study, the diagnoses of the diseases, the main lines of which are formed by these diseases were made. In the data set published by the American National Institute of Health, there are 112120 x-ray images of 30805 patients who are different from each other [1]. Although it may seem easy for specialists to read chest X-ray images, the underlying physiological and pathological principles and anatomical structures make this a difficult and complex problem that requires extensive knowledge. Such factors increase the difficulty of developing a consistent and automated technique for reading chest X-ray images, while also taking into account all common chest diseases. Thus, with our study, it is aimed to

develop a deep learning-based computer application that will help experts interpreting these images. In order for the application to work effectively on mobile and low-capacity computers, a new model has been developed based on architectures such as MobileNet and EfficientNet. The AUC values we obtained with the models we developed are as follows:

AUC values obtained for the classes we studied in the AMESA model, Atelectasis 0.76, Cardiomegaly 0.93, Consolidation 0.70, Edema 0.89, Effusion 0.85, Emphysema 0.83, Fibrosis 0.80, Infiltration 0.68, Mass 0.79, Nodule 0.67, Pleural Thickening 0.68, Pneumonia 0.705, Pneumothorax 0.870 and Hernia 0.69.

The AUC values obtained for the classes we studied in the ESA 1 model are Atelectasis 0.779, Cardiomegaly 0.92, Consolidation 0.721, Edema 0.875, Effusion 0.896, Emphysema 0.77, Fibrosis 0.798, Infiltration 0.712, Mass 0.802, Nodule 0.745, Pneumothorax 0.847, Pneumonia 0.705, Pleural Thickening 0.778 and Hernia 0.65.

Keywords: Deep Learning, Convolutional Neural Network, Chest Disease Detection, Artificial Neural Network, X-Ray Image

1.1 Literatür Özeti

Brunese ve ark. üç aşamadan oluşan VGG-16 temelli bir ESA modeli geliştirmişlerdir. İlk aşamada akciğerde zatürre olup olmadığı tespit edilmektedir. İkinci aşamada Kovid-19 ile zatürre arasındaki ayırım yapılmaktadır. Son adımda, Kovid-19'un etkilenmiş akciğer bölgelerinden lokalize edilmesi sağlanmıştır. Sonuç olarak farklı kurumlara ait 6.523 göğüs röntgeni üzerinde yaptıkları deneysel analizler sonunda ortalama KOVİD-19 tespiti süresini yaklaşık 2,5 saniye ve ortalama doğruluk oranını da 0,97 olarak bulmuşlardır [2].

Gupta ve ark. yaptıkları çalışmada model olarak yığın şekilde entegre edilmiş bir ESA olan InstaCovNet-19'u kullanmıştır. Kullandıkları modeli desteklemek için daha önce eğitilmiş olan ResNet101, Xception, InceptionV3, MobileNet ve NASNet gibi modelleri kullanmıştır. Bu çalışmada röntgen görüntülerinden elde edilen anormallikler kullanılarak Kovid-19 ve zatürreyi tespit etmek amaçlanmıştır. Yapılan çalışmada üç sınıf için (Kovid-19, zatürre ve normal) %99.08 doğruluk elde ederken 2 sınıf (Kovid- NonKovid) için ise %99,53 doğruluk elde etmiştir [3].

Wang ve ark. göğüs radyografisinde toraks hastalıklarının teşhisini sağlamak amacıyla geliştirdikleri ESA ile ChestNet modelini önermişlerdir. Bu yapılan çalışmada 14 adet göğüs hastalığı tespiti için geliştirilen model sınıf başına ortalama 0.781 AOC değerine ulaşmıştır [4].

Rajpurkar ve ark. CheXNet ismini verdikleri 121 katmanlı ESA algoritması ile ChestX-ray14 veri seti üzerinde yaptıkları çalışmalar sonucunda elde ettikleri 0.435 (%95 CI 0.387, 0.481) F1 skoru ile radyologlardan ortalama 0.387 (%95 CI 0.330, 0.442) daha iyi bir performans göstermiştir [5].

Liu ve ark. geliştirdikleri ESA ile dengesiz ve az kategorideki x-ışını görüntülerinden çoklu tüberküloz belirtilerini sınıflandırmayı amaçlamaktadır. Geliştirdikleri model

ile yüksek sayıdaki tüberküloz veri seti üzerinde %85,68'lik bir doğruluk oranı elde etmişlerdir [6].

Yan ve ark. çalışmalarında, göğüs hastalıklarını sınıflandırmak ve şüpheli lezyon bölgelerini lokalize etmek için, sıkıştırma ve uyarma blokları, çoklu harita aktarımı ve maks - min havuzlama ile donatılmış zayıf denetimli DenseNet temelli bir derin öğrenme çerçevesi önermiştir. ChestX-ray14 veri seti üzerinde geliştirdikleri model 0.8302 AOC değerini elde etmiştir [7].

Abiyev ve ark. geliştirdikleri ESA ile zatürre, astım ve tüberküloz gibi yaygın akciğer hastalıklarının erken teşhis edilmesi üzerinde çalışmışlardır. Karşılaştırma yapabilmek için ESA temelli modellerini üç yapıda oluşturmuşlardır; Bunlar temel ESA(CNN), geri yayımlı sinir ağı (BPNN) ve rekabetçi sinir ağıdır (CpNN). Sırasıyla bu modeller için elde ettikleri doğruluk oranları şöyledir; %92.4, %80.04 ve %89.57 şeklindedir [8].

Madani ve ark. veri setlerindeki yetersiz sayıda örneklere dikkat çekerek bu verileri gerçeğe uygun bir şekilde çoğaltmanın yollarını aramışlar ve yaptıkları çalışmada da bunun için çekişmeli üretici ağı (GANs) kullanmışlardır. Elde ettikleri görüntülerdeki kardiyovasküler anaormallikleri sınıflandırmak için bir ESA kullanmışlardır. Elde ettikleri sonuçlar; Veri artırma olmadan %81.93, geleneksel veri artırma yöntemleriyle %83.12, GAN veri artırma yöntemiyle ise %84.19 olarak açıklanmıştır [9].

Jaiswal ve ark. göğüs röntgen görüntülerinden zatürrenin tespit ve lokalizasyonunu yapmak için geliştirdikleri modelde Mask-RCNN yapısını kullanmışlardır. Bu modeli ResNet50 ve ResNet101 mimarisi üzerinde geliştirmişlerdir [10].

Asif ve ark. Kovid-19 kaynaklı zatürre hastalığının teşhisi için göğüs röntgen görüntüleri üzerinde öğrenme aktarımı metodu ile geliştirilmiş Inception V3 tabanlı bir derin ESA modellerini kullanmışlardır. Veri setleri 864 adet Kovid-19, 1345 adet viral zatürre ve 1341 adet normal göğüs röntgen görüntülerinden oluşmaktadır. Çalışmaları %98'in üzerinde bir sınıflandırma doğruluğu elde etmiştir [11].

Basu ve ark. Kovid-19 tespiti için yapılan PCR testlerinin FP oranlarının yüksek olması ve bu sebeple radyolojik görüntülerden hastalık tespitinin önemine

vurgulayarak yaptıkları çalışmada derin ESA temelli bir model geliştirerek soruna çözüm bulmaya çalışmışlardır. Bununla birlikte, mevcut KOVID-19 veri setleri derin sinir ağlarını eğitmek için yetersiz olduğundan alan uzantısı aktarım öğrenimi (DETL) adı verilen yeni bir kavram önermişlerdir. Yaptıkları çalışma ile %90,13 ± 0,14 oranında bir genel doğruluk oranı elde etmişlerdir [12].

Yoo ve ark. göğüs röntgen görüntülerinden Kovid-19'u tespit etmek için derin öğrenme tabanlı bir karar ağacı sınıflandırıcısı kullanmanın uygulanabilirliğini araştırmışlardır. Önerilen sınıflandırıcı, her biri PyTorch kütüphanesine dayalı evrişim sinir ağına sahip bir derin öğrenme modeli tarafından eğitilen üç ikili karar ağacından oluşmaktadır. İlk karar ağacı, röntgen görüntülerini normal veya anormal olarak sınıflandırır. İkinci ağaç, tüberküloz belirtileri içeren anormal görüntüleri tanımlar, üçüncüsü ise Kovid-19'u tanılamaya çalışır. Birinci ve ikinci karar ağaçlarının doğrulukları sırasıyla %98 ve %80 iken, üçüncü karar ağacının ortalama doğruluğu %95'tir [13].

Ouchicha ve ark. yaptıkları çalışmada göğüs röntgen görüntülerinden Kovid-19 kaynaklı zatürre tanısı yapmak için derin evrişimli sinir ağı (ESA) modeli olan CVDNet isimli ResNET tabanlı bir model önermişlerdir. Bu çalışma, 219 Kovid-19, 1341 normal ve 1345 viral zatürre göğüs röntgeni görüntülerinin bir kombinasyonunu içeren bir veri seti üzerinde uygulanmıştır. Geliştirdikleri model Kovid-19 tespiti için %97,20'lik genel doğruluk oranı elde etmiştir. Üçlü sınıflandırmada ise (Kovid-19, zatürre ve normal) için ortalama %96,69 doğruluk oranı elde etmiştir [14].

Bhandary ve ark. yaptıkları çalışmalarında akciğer zatürresi ve kanseri incelemek için bir derin öğrenme modeli önermişlerdir. Çalışmalarında iki farklı derin öğrenme tekniği önermişlerdir. Değiştirilmiş AlexNet olarak adlandırdıkları ve önerdikleri ilk teknik ile göğüs röntgeni görüntülerini normal ve zatürre olarak iki sınıfa ayırmak için kullanmışlardır. Bu teknikte sınıflandırma işlemi Destek Vektör Makinesi (SVM) kullanılarak gerçekleştirmişler ve performans sonuçlarını da Softmax ile karşılaştırmışlardır. Yaptıkları çalışmanın performansını ayrıca AlexNet, VGG16, VGG19 ve ResNet50 gibi önceden eğitilmiş diğer derin öğrenme teknikleriyle doğrulamışlardır. Önerdikleri ikinci teknik ile akciğer kanseri

sınıflandırma doğruluğunu arttırmak amacıyla değiştirilmiş AlexNet modeline elle ayarlanmış ve öğrenilmiş özelliklerin birleşimi uygulanmıştır. Yaptıkları çalışma ile seri füzyon ve temel bileşen analizi tabanlı özellik seçimini kullanmışlardır. Yaptıkları çalışma ile %97,27 sınıflandırma doğruluğu elde etmişlerdir [15].

Jain ve ark. çalışmalarında Kovid-19'un sebep olduğu akciğer hastalıkları üzerinde yoğunlaşmışlardır. Yaptıkları çalışmada sağlıklı ve kovid-19'dan etkilenmiş hastaların röntgen görüntüleri üzerinde gerekli temizleme ve veri artırma işlemi sonrasında derin öğrenme tabanlı CNN modellerinden Inception V3, Xception ve ResNeXt'i kullanıp performanslarını karşılaştırmışlardır. Geliştirdikleri modeli analiz etmek için Kaggle'dan 6432 adet göğüs röntgeni kullanılmıştır. Bunun 5467'sini eğitim 965'ini ise doğrulama için kullanmışlardır. Yaptıkları çalışmada kullandıkları Xception modeli diğer modellere kıyasla %97,97 oranla en iyi sonucu vermiştir [16].

Chouhan ve ark. uzman ve deneyimsiz kişilerin zatürre tespitini kolaylaştırmak amacıyla geliştirdikleri modelde transfer öğrenme konseptini kullanmışlardır. Geliştirdikleri modelde ImageNet üzerinde önceden eğitilmiş farklı sinir ağı modelleri kullanılarak sınıflandırma yapmışlardır. Bu şekilde beş farklı model hazırlayıp performanslarını analiz etmişlerdir. Sonuç olarak eğittikleri modellerin çıktılarını birleştirerek bireysel modellerinden daha iyi performans gösteren bir topluluk modeli önermişlerdir. Modelleri Guangzhou Kadın ve Çocuk Tıp Merkezinden aldıkları veri seti üzerinde %99,62'lik duyarlılık ve %96,4 doğruluk elde etmiştir [17].

Abbas ve ark. Kovid19 göğüs röntgeni görüntülerinin sınıflandırılması için Ayırıştırma, Aktarma ve Oluşturma (DeTraC) adını verdikleri bir derin CNN modeli önermişlerdir. Bu model ile dünya çapındaki çeşitli hastanelerden topladıkları normal ve şiddetli akut solunum sendromu vakalarından Kovid-19 X-ray görüntülerinin saptanmasında %95,12'lik doğruluk (%97,91 duyarlılık ve %91,87 özgüllük) sağlamışlardır [18].

Rahman ve ark. yaptıkları çalışmada kronik bir akciğer hastalığı olan Tüberkülozu tespit etmeye çalışmışlardır. Geliştirdikleri model için önceden eğitilmiş başlangıç ağırlıklarından transfer öğrenimi için dokuz farklı derin CNN modeli ResNet18,

ResNet50, ResNet101, ChexNet, InceptionV3, Vgg19, DenseNet201, SqueezeNet ve MobileNet kullanılmışlar ve Tüberküloz ve olmayanları sınıflandırmak için eğitmiş, doğrulamış ve test etmişlerdir. Yaptıkları çalışmada üç farklı deney gerçekleştirmişlerdir bunlar, X-ışını görüntülerinin iki farklı U-net modeli kullanılarak bölümlere ayrılması, X-ışını görüntüleri kullanılarak sınıflandırma ve bölümlere ayrılmış akciğer görüntülerinin kullanılmasıdır. Yaptıkları çalışmada halka açık veri tabanlarından 3500 Tüberküloz ve 3500 normal göğüs röntgeni kullanmışlardır. Bu veri setini kullanarak tüberküloz tespitinde en iyi performans gösteren model ChexNet olmuş ve doğruluğu, kesinliği, duyarlılığı, F1 skoru ve özgüllüğü sırasıyla %96,47, %96,62, %96,47, %96,47 ve %96,51 olmuştur. Bununla birlikte, bölütlenmiş akciğer görüntülerini sınıflandırmak için kullandıkları DenseNet201 ise tüm X-ışını görüntülerinden daha iyi performans göstermiş ve doğruluk, kesinlik, duyarlılık, F1 skoru ve özgüllüğü sırasıyla %98,6, %98,57, %98,56, %98,56 ve %98,54 olmuştur [19].

Das ve ark. Kovid-19'u tespit edebilmek için Truncated Inception Net adını verdikleri derin öğrenme tabanlı bir CNN modeli önermişlerdir. Modellerinde Kovid-19 pozitif, Tüberküloz pozitif, Zatürre pozitif ve sağlıklı vakaların olduğu bir veri setinden faydalanmışlardır. Önerdikleri model Zatürre ve sağlıklı vakalardan Kovid-19 ayırmada %99,96 (AUC 1.0) doğruluk elde etmiştir. Ayrıca modelleri, Kovid-19 pozitif vakaları Zatürre, Tüberküloz ve sağlıklı görüntülerden ayırmada %99,92 (AUC 0,99) doğruluk elde etmiştir [20].

Palani ve ark. yaptıkları çalışma ile göğüs röntgenlerini işleyebilen ve farklı göğüs hastalıklarını tespit edebilen çeşitli derin öğrenme teknikleri hakkında karşılaştırmalı bir çalışma sunmuşlardır. Bunlar Atelektazi, Kardiyomegali, Konsolidasyon, Ödem, Efüzyon, Amfizem, Fibrozis, Fıtık, İnfiltrasyon, Kütle, Nodül, Zatürre, Pnömotoraks ve Plevral kalınlaşma olmak üzere toplam 14 farklı göğüs hastalığından oluşan x-ışını görüntüleridir. Kullandıkları model Residual Neural Network tabanlı bir derin öğrenme mimarisidir. Geliştirdikleri model %97,99 oranında eğitim doğruluğu elde etmiştir [21].

Guan ve ark. yaptıkları çalışmada 14 yaygın göğüs hastalığı üzerinde çalışmışlardır. CRAL ismini verdikleri modelde önerdikleri yapı iki kısımdan oluşmaktadır. Bunlar

özellik öğrenme ve dikkat öğrenme modülü olarak isimlendirilmiştir. Yaptıkları çalışma ile ortalama 0.816 AUC değeri elde etmişlerdir [22].

Gündel ve ark. yaptıkları çalışma ile DNetLoc ismini verdikleri model ile 14 yaygın göğüs hastalığı için ortalama 0.841 AUC değeri elde etmişlerdir [23].

1.2 Tezin Amacı

Salgın hastalıkların arttığı günümüzde erken ve etkili tedavi için hızlı teşhisin önemi oldukça büyüktür. Bu çalışmanın amacı göğüs röntgen görüntülerini yorumlayarak yaygın göğüs hastalıklarını hızlı ve doğru bir şekilde tespit edip ilgili uzmanlara yardımcı olacak derin öğrenme tabanlı bir bilgisayar uygulaması geliştirmektir.

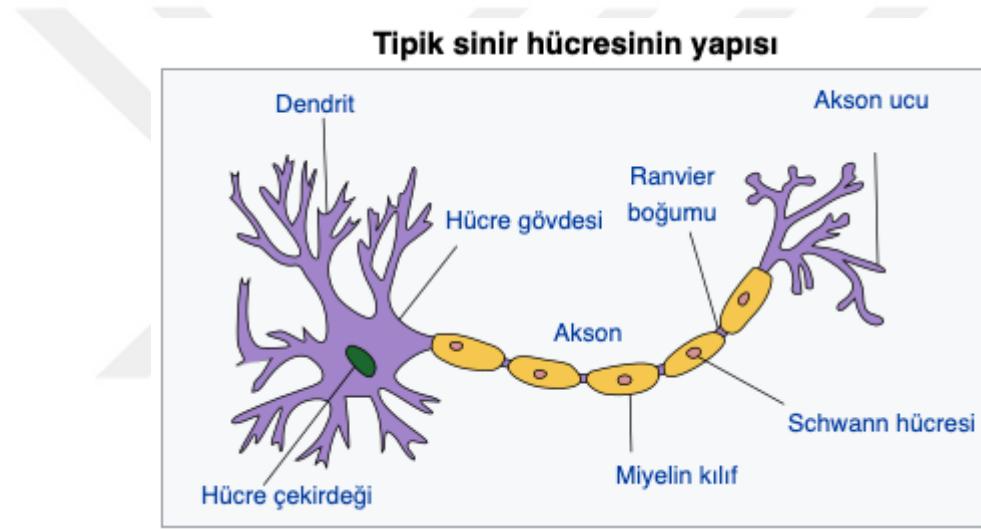
1.3 Hipotez

14 Yaygın göğüs hastalığının tespiti için iki adet model geliştirilmesi yapılmıştır. Bunlardan ilki transfer öğrenme metoduyla daha önce farklı veri setleri üzerinde eğitilerek geliştirilen modelleri baz almaktadır. Bu modeller kendi veri setimiz için adapte edilerek verdikleri sonuçlar değerlendirilir ve uygun olan modele kendi ESA modelimiz eklenir. Daha sonra modelin resimlerde hastalık ile ilgili alanlara yoğunlaşmasını sağlamak için üçüncü bir ESA katmanı daha eklenir. Böylece daha hızlı öğrenen ve daha iyi sınıflandıran bir modele ulaşılması hedeflenmiştir.

İkinci modelde çeşitli ESA yapıları üzerinde veri seti için çalışmalar yaparak en uygun katman yapısının elde edilmesi amaçlanmıştır. Elde edilen modeller uzmanlar için kullanılabilir bir uygulamaya dönüştürülerek hastalık teşhisinde yardımcı bir araç olarak kullanılması hedeflenmiştir.

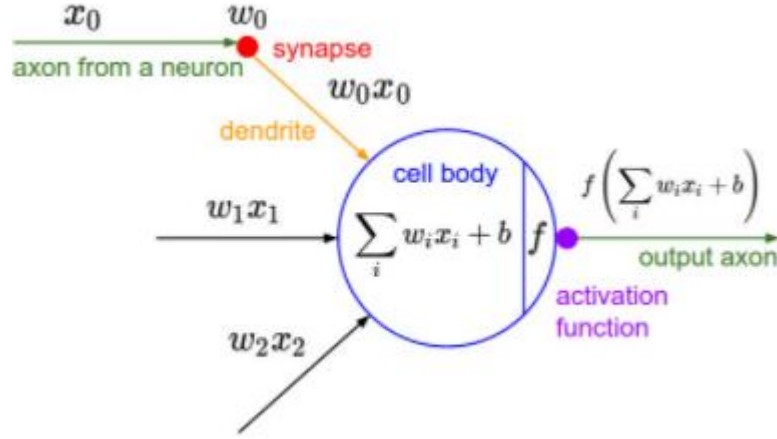
2.1 Yapay Sinir Ağları

Yapay sinir ağları beyinde bulunan nöronların çalışma biçiminden esinlenerek tasarlanmış bilgi işlemeye yarayan bir algoritmadır. Beyin hücrelerini oluşturan nöron adı verilen hücrelerin çalışma biçimlerinin dijital olarak modellenmesi ile geliştirilmiştir.



Şekil 2. 1 Örnek Beyin Hücresi [25]

Bir insanın sinir sisteminde yaklaşık 86 milyar nöron olduğu tahmin edilmektedir. Ve bu sinir sistemini oluşturan bir nöron yaklaşık $10^{14} - 10^{15}$ arasında diğer nöronlar ile bağlantılı olabilmektedir. Nöronlar dendritlerden aldığı bilgiyi çekirdeğinde işledikten sonra Akson boyunca tek yönlü olarak çıkış verisi olarak gönderir. Akson adı verilen çıkışın üstü "Miyelin Kılıf" adı verilen bir tabaka ile kaplıdır ve bu kısım çıkıştaki yayılım hızına etki eden bir tür direnç gibi görev alır.



Şekil 2. 2 Perceptron [26]

İşte bu yapının modellenmesi ile ortaya çıkan ve bir yapay sinir ağının en küçük parçasını temsil eden yapıya "perceptron" denir [27]. Frank Rosenblatt tarafından ilk defa 1958 yılında tanımlanmıştır [28]. Bu perceptron tıpkı bir nöron gibi girdileri alır ve bir çıktıyı üretir/üretir. Bundan sonra bu yapıya nöron demeye devam edeceğiz. Her nöron, girdi olarak bir dizi gerçek değerli sayı alan, bunlar üzerinde bazı hesaplamalar yapan ve tek bir çıktı değeri üreten küçük bir hesaplama birimidir. Nöron içinde yapılan işlem şöyle ifade edilir,

$$z = w \cdot x + b \quad (2.1)$$

Nöronun çıkışında üretilen sinyal, w ağırlıklı z girdilerinin toplamının bir fonksiyonudur. Burada w , ağırlık olarak adlandırılır ve girilen verilerin önemine göre değeri ayarlanarak nöronun çekirdeğine iletilmesini sağlar. Gerçek hayattaki öğrenme gibi yaşayıp tecrübe ettikçe nöronlar arası bağlantılar ayarlanır ve hatta yeni bağlantılar oluşur. Benzer şekilde gerçek hayattaki gibi YSA da yer alan nöronlar arasındaki bağlantının ayarlanması da w ile olur.

Nöronun çıkışında üretilen veriye y dersek, f ile gösterdiğimiz fonksiyon doğrusal değildir ve Aktivasyon Fonksiyonu olarak adlandırılır.

$$y = a = f(z) \quad (2.2)$$

Aktivasyon fonksiyonu nöronun çekirdeğinde üretilen sinyalin çıkışa ne oranda aktarılacağına karar veren bir mekanizmadır [24].

2.2 Gözetimli Öğrenme

Gözetimli öğrenme, elimizdeki giriş değerlerine karşılık gelen çıkış değerlerinde sahip olduğumuz bir öğrenme türüdür. Böylece etiketlenmiş verileri kullanarak etiketlenmemiş verileri sınıflandırabiliriz.

$$\{x(1), \dots, x(m)\} \rightarrow \{y(1), \dots, y(m)\} \quad (2.3)$$

Örneğin özelliklerine göre sınıflandırılmış ev değerleri ve hangi nesneye ait olduğu bilinen resimler gibi veri setleri gözetimli öğrenmeye örnek verilebilir [24].

2.3 İkili Sınıflandırma

Etiketlemek istediğimiz verinin sınıf sayısının iki olduğu sınıflandırma şeklindedir. Örneğin sonucun 0 ya da 1 olma durumu ya da bir resmin insan olup olmadığı gibi sınıflandırma şeklindedir. Matematiksel gösterimi şöyledir,

$$(x, y) \quad x \in \mathbb{R}^{n_x} \quad y \in \{0,1\} \quad (2.4)$$

m örnek sayısı olmak üzere,

$$m = \{(x^1, y^1), (x^2, y^2), \dots, (x^m, y^m)\} \quad (2.5)$$

Şeklinde ifade edilir [24].

2.4 Lojistik Regresyon

Regresyon, iki ve daha fazla değişken arasındaki matematiksel ilişkiyi denklemlerle ifade etmek ve değişkenlerin birbirlerinden etkilenme biçimini ve büyüklüğünü ortaya koymak için yararlanılan bir istatistiksel yöntemdir. Makine öğrenmesinde ikili sınıflandırmayı yapabilmek için kullandığımız regresyon modeline Lojistik Regresyon adı verilir. Verilen x girdileri için elde edeceğimiz y çıktıları bir olasılık olarak elde etmemizi sağlar.

$$x, x \in \mathbb{R} \quad \text{ve} \quad y = P(y = 1|x), \quad 0 \leq y \leq 1 \quad (2.6)$$

Daha önce açıkladığımız üzere $y = w^T \cdot x + b$ denkleminde y değeri eksi ve artı çok büyük değerler alabilmektedir.

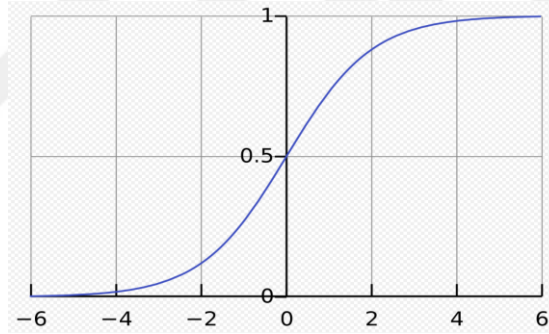
Bu durumu düzeltmek ve sonucu 0 ile 1 arasında sabitleyebilmek için Sigmoid Fonksiyonundan faydalanılır. Eğer bu fonksiyonu $\sigma(z)$ ile ifade edersek,

$$\sigma(z) = \frac{1}{1 + e^{-z}} \quad (2.7)$$

Şeklinde yazılır. Bu durumda y çıktımız,

$$y = \sigma(w^T \cdot x + b) \quad (2.8)$$

Olacak ve grafik gösterimi ise,



Şekil 2. 3 Sigmoid Eğrisi

Şekilden de görüldüğü gibi çıkışı 0 ile 1 arasında sıkıştıran bir yapıda olacaktır [24].

2.5 Kayıp ve Maliyet Fonksiyonu

Modellediğimiz sistemin tek bir çıktı için başarısını ölçmek için kullandığımız fonksiyona kayıp fonksiyonu ve tüm çıktılar için başarı ölçümünde kullandığımız fonksiyona da maliyet fonksiyonu denir. Eğer çıktımız \hat{y} ve gerçek değerimizde y ise kayıp fonksiyonunu şöyle tanımlayabiliriz,

$$y = \sigma(w^T \cdot x + b) \quad (2.9)$$

$$L(\hat{y}, y) = \frac{1}{2} (\hat{y} - y)^2 \quad (2.10)$$

$$L(\hat{y}, y) = -(y \log \hat{y} + (1 - y) \log(1 - \hat{y})) \quad (2.11)$$

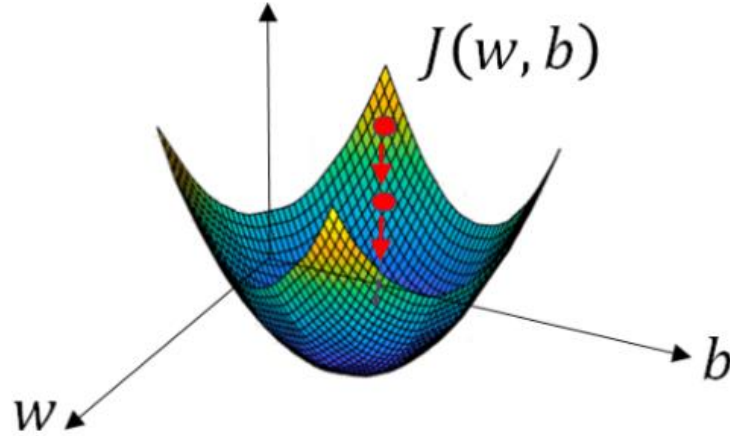
Sistemimizin m örnekten oluştuğunu düşünürsek maliyet fonksiyonunu da şöyle tanımlayabiliriz,

$$J(w, b) = \frac{1}{m} \sum_{i=1}^m L(\hat{y}^{(i)}, y^{(i)}) = -\frac{1}{m} \sum_{i=1}^m [y \log \hat{y} + (1 - y) \log(1 - \hat{y})] \quad (2.12)$$

Denklemdaki i harfi, örneğin sıra sayısını ifade etmektedir. [24]

2.6 Dereceli Azalma

Kurduğumuz modelin toplam hatasını içeren maliyet fonksiyonu yani (2.12) denklemini minimum yapan w ve b değerlerini bulmaya çalışırız. Bunu yaparken kullandığımız en önemli yöntemlerden birisi de Gradyan Azalması da denilen Dereceli Azalma yöntemidir.



Şekil 2. 4 Dereceli Azalma [29]

Grafikten de görüleceği üzere J maliyet (toplam hata) fonksiyonunu en aza indirecek şekilde kırmızı yolu takip ederek en iyi w ve b değerlerini elde etmek için maliyet fonksiyonunun w' ya göre türevini alarak mevcut w değerlerini güncellemeliyiz.

$$w := w - \alpha \frac{\partial J(w, b)}{\partial w} \quad (2.13)$$

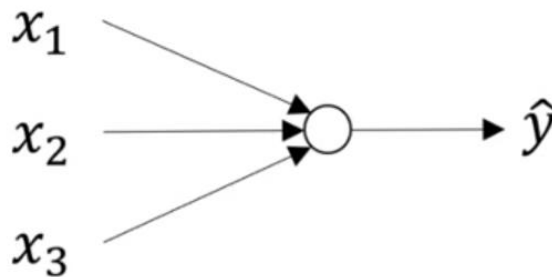
Aynı şekilde b ön yargı (bias) değerlerini de güncellemek için,

$$b := b - \alpha \frac{\partial J(w, b)}{\partial b} \quad (2.14)$$

Denklemini kullanırız. Buradaki α değeri öğrenme hızı olarak adlandırılır ve istenilen parametrenin ne kadar hızlı şekilde minimuma yaklaşacağını ayarlamak için kullanılır. Eğer bu parametreye büyük değerler verirsek global minimumu atlayabilir ya da bunun tersine küçük verirsek parametrenin çok daha yavaş bir şekilde minimuma gitmesine neden olabiliriz [24].

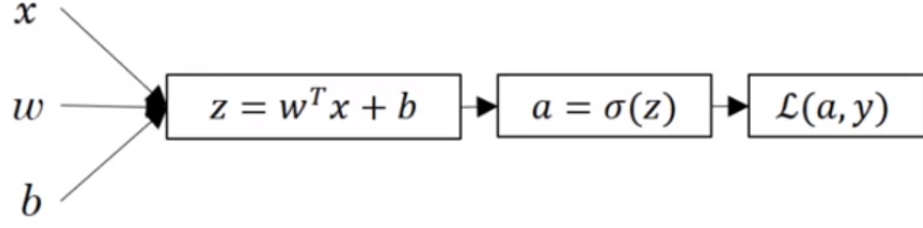
2.7 Tek Katmanlı Yapay Sinir Ağı Modeli

Daha önce 2.1 de incelediğimiz Yapay Sinir Ağına geri dönecek olursak aşağıda 3 girdisi (x_1, x_2 ve x_3) ve 1 çıktısı (\hat{y}) olan tek katmanlı bir yapay sinir ağı görülmektedir,



Şekil 2. 5 Tek Katmanlı Yapay Sinir Ağı Modeli [29]

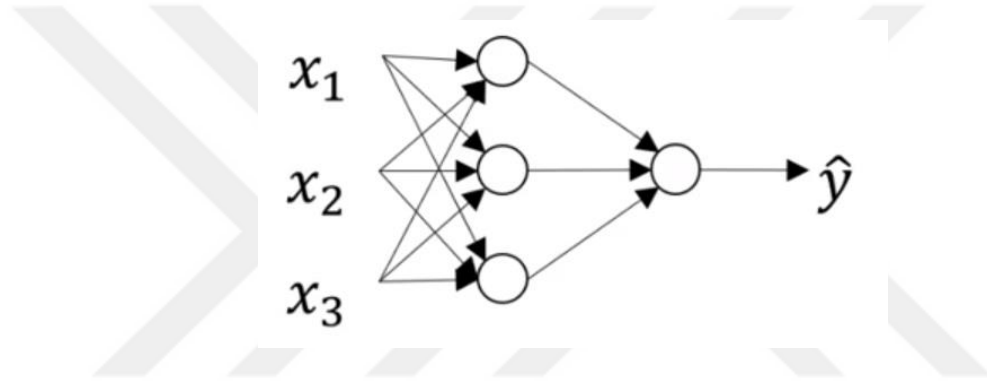
Bu ağı gözetimli öğrenme modeli için kullanılan lineer olmayan regresyon amaçlı tasarlandığını kabul edip bu sinir ağının tüm girdilerini x olarak düşünecek olursak ve $\hat{y} = a$ dersek L Kayıp fonksiyonunu şöyle gösterebiliriz, [24]



Şekil 2. 6 Yapay Sinir Ağı Kayıp Fonksiyonu [29]

2.8 İki Katmanlı Yapay Sinir Ağı Modeli

Bu sefer şekil 2.5'deki yapay sinir ağına bir ara katman daha eklersek 3 girdisi (x_1, x_2 ve x_3) ve 1 çıktısı (\hat{y}) olan iki katmanlı bir yapay sinir ağı elde ederiz,



Şekil 2. 7 İki Katmanlı Yapay Sinir Ağı [29]

Bu gösterimde x_1, x_2 ve x_3 'ün oluşturduğu ilk bölüme girdi katmanı (Input Layer) adı verilir ve fakat bu katman sinir ağının katman sayısına dahil edilmez. 3 adet düğümden oluşan ikinci bölüm ise gizli katman (Hidden Layer) olarak tanımlanır. Son kısımda yer alan düğümün bulunduğu katman ise çıkış katmanı (Output Layer) olarak tanımlanır ve \hat{y} değeri bu katmandan üretilir. Böylece ilk katman sıfırncı katman olarak tanımlandığından bu yapıya 2 katmanlı bir sinir ağı diyebiliriz. Denetimli öğrenme uygulaması yapan bir sinir ağı x girdilerinden oluşan eğitim kümesini ve \hat{y} olan sınıf değerlerini içerir. Fakat gizli katman olarak tanımladığımız 3 düğümden oluşan orta kısım bu eğitim kümesinin girdileri olan x değerlerinin bir kümesi olarak ifade edilmez ve bu sebeple bu alana gizli katman denilmektedir.

Girdi katmanı olan x_1, x_2 ve x_3 'ün oluşturduğu ilk bölümde $x_1, x_2, x_3 = a^{[0]}$ dersek

gizli katmanı bu durumda $a^{[1]}$ olarak tanımlayabiliriz. Bu aynı zamanda 3 düğümünden oluşan 3 boyutlu bir vektördür ve şu şekilde ifade edilir,

$$\mathbf{a}^{[1]} = \begin{bmatrix} \mathbf{a}_1^{[1]} \\ \mathbf{a}_2^{[1]} \\ \mathbf{a}_3^{[1]} \end{bmatrix} \quad (2.15)$$

Böylece çıkış katmanını $a^{[2]}$ olarak ifade edebilir ve şu şekilde yazabiliriz,

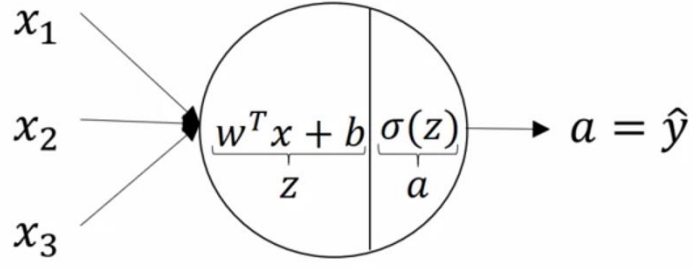
$$a^{[2]} = \hat{y} \quad (2.16)$$

Bu durumda yine çıkış değerine $\hat{y} = a^{[2]}$ dersek kayıp fonksiyonu olan L şöyle tanımlanır,

$$\begin{aligned} z^{[1]} = W^{[1]}x + b^{[1]} \rightarrow a^{[1]} = \sigma(z^{[1]}) \rightarrow z^{[2]} = W^{[2]}x + b^{[2]} \rightarrow \\ a^{[2]} = \sigma(z^{[2]}) \rightarrow L(a^{[2]}, y) \end{aligned} \quad (2.17)$$

(2.15) denklemindeki $a^{[1]}$ ifadesi ilk katmandaki çıktıyı, $a^{[2]}$ ifadesi ise ikinci katmandaki çıktıyı ifade etmektedir. Bu durumda kayıp fonksiyonu L, gerçek olması gereken y ifadesi ile $a^{[2]}$ arasındaki başarının ölçümü olarak tanımlanır.

Şimdi şekil (2.5) de tanımladığımız tek katmanlı sinir ağının hesaplamalarına daha yakından bakalım,



Şekil 2. 8 Tek Katmanlı Sinir Ağında Hesaplama [29]

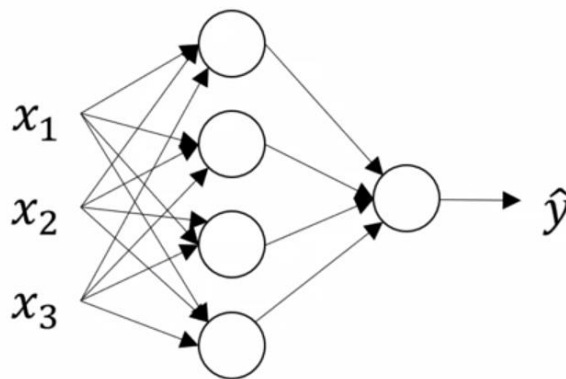
Eğer tek katmanlı sinir ağımızın ilk düğümü olan hücreesine yakından bakarsak şekilden de görüleceği üzere ilk olarak z değeri hesaplanır.

$$z = w^T \cdot x + b \quad (2.18)$$

Daha sonra da z 'yi aktivasyon fonksiyonundan geçirerek a değerini buluruz,

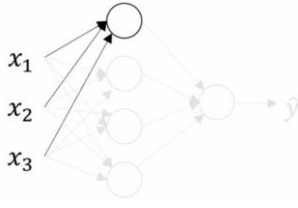
$$a = \sigma(z) \quad (2.19)$$

Bu hesaplama diğer çok katmanlı sinir ağlarında görülen tüm düğümlerde her seferinde tekrar uygulanır. Şimdi bir gizli katmanlı aşağıdaki yapı için uygulanan hesaplamalara bakalım,



Şekil 2. 9 Bir Gizli Katmanlı Sinir Ağı Örneği [29]

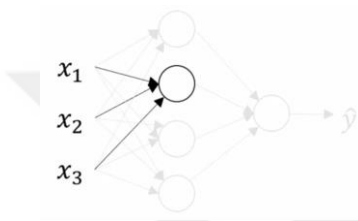
Şimdi bu gizli katmandaki ilk düğüme bakacak olursak ilgili hesaplamalar aşağıdaki gibi olacaktır,



$$z_1^{[1]} = W_1^{[1]T} x + b_1^{[1]} \quad (2.20)$$

$$a_1^{[1]} = \sigma(z_1^{[1]}) \quad (2.21)$$

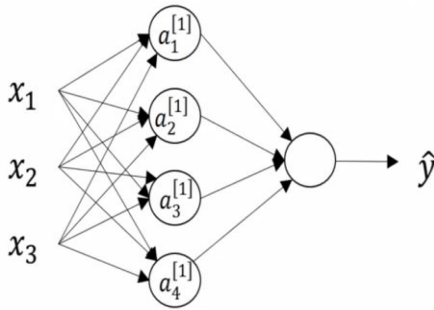
Burada yer alan üst indis hesaplanan katmanın hangisi olduğunu, alt indis ise bu katmandaki düğüm konumunu belirtmektedir. İkinci düğümdeki hesaplamayı da aşağıdaki gibi yazabiliriz,



$$z_2^{[1]} = W_2^{[1]T} x + b_2^{[1]} \quad (2.22)$$

$$a_2^{[1]} = \sigma(z_2^{[1]}) \quad (2.23)$$

Böylece 4 düğüm için yapılan hesaplamalar şöyle olacaktır,



$$z_1^{[1]} = W_1^{[1]T} x + b_1^{[1]}, a_1^{[1]} = \sigma(z_1^{[1]}) \quad (2.24)$$

$$z_2^{[1]} = W_2^{[1]T} x + b_2^{[1]}, a_2^{[1]} = \sigma(z_2^{[1]}) \quad (2.25)$$

$$z_3^{[1]} = W_3^{[1]T} x + b_3^{[1]}, a_3^{[1]} = \sigma(z_3^{[1]}) \quad (2.26)$$

$$z_4^{[1]} = W_4^{[1]T} x + b_4^{[1]}, a_4^{[1]} = \sigma(z_4^{[1]}) \quad (2.27)$$

Tüm bu hesaplamalar kodlanırken her düğüm için bir döngü yazmak yerine bu yapıyı vektör haline getirirsek hesaplamalarımız hem daha hızlı sonuç verecek hem de etkili olacaktır.

Şimdi ilk olarak ağırlık değerlerimizi vektör formunda yazalım ve buna W diyelim,

$$W^{[1]} = \begin{bmatrix} W_1^{[1]T} \\ W_2^{[1]T} \\ W_3^{[1]T} \\ W_4^{[1]T} \end{bmatrix} \quad (2.28)$$

Girdilerimizi ise X olarak gösterelim,

$$X = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} \quad (2.29)$$

Yanlılık olarak adlandırılan bias değerini b olarak gösterelim,

$$b^{[1]} = \begin{bmatrix} b_1^{[1]} \\ b_2^{[1]} \\ b_3^{[1]} \\ b_4^{[1]} \end{bmatrix} \quad (2.30)$$

Çıkış değerlerimizi ise $Z^{[1]}$ olarak gösterelim,

$$Z^{[1]} = \begin{bmatrix} z_1^{[1]} \\ z_2^{[1]} \\ z_3^{[1]} \\ z_4^{[1]} \end{bmatrix} \quad (2.31)$$

Sigmoid aktivasyon fonksiyonu çıktılarını $a^{[1]}$ olarak gösterelim,

$$a^{[1]} = \begin{bmatrix} a_1^{[1]} \\ a_2^{[1]} \\ a_3^{[1]} \\ a_4^{[1]} \end{bmatrix} \quad (2.32)$$

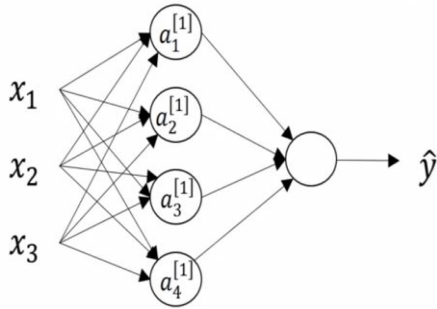
Böylece yukarıda yaptığımız hesaplama işlemi vektörel olarak yeniden yazarsak,

$$Z^{[1]} = \begin{bmatrix} W_1^{[1]T} \\ W_2^{[1]T} \\ W_3^{[1]T} \\ W_4^{[1]T} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} + \begin{bmatrix} b_1^{[1]} \\ b_2^{[1]} \\ b_3^{[1]} \\ b_4^{[1]} \end{bmatrix} = \begin{bmatrix} W_1^{[1]T} x + b_1^{[1]} \\ W_2^{[1]T} x + b_2^{[1]} \\ W_3^{[1]T} x + b_3^{[1]} \\ W_4^{[1]T} x + b_4^{[1]} \end{bmatrix} = \begin{bmatrix} z_1^{[1]} \\ z_2^{[1]} \\ z_3^{[1]} \\ z_4^{[1]} \end{bmatrix} \quad (2.33)$$

Son olarak bu çıktının sigmoid aktivasyon fonksiyonuna uygulanmış hali olan a değerlerini gösterecek olursak,

$$a^{[1]} = \begin{bmatrix} a_1^{[1]} \\ a_2^{[1]} \\ a_3^{[1]} \\ a_4^{[1]} \end{bmatrix} = \sigma(z^{[1]}) \quad (2.34)$$

Böylece vektörel gösterimi ile ağın son hali şöyle olur,



$$Z^{[1]} = W^{[1]T} x + b^{[1]} \quad (2.35)$$

$$a^{[1]} = \sigma(Z^{[1]}) \quad (2.36)$$

$$Z^{[2]} = W^{[2]T} a^{[1]} + b^{[2]} \quad (2.37)$$

$$a^{[2]} = \sigma(Z^{[2]}) \quad (2.38)$$

(2.20) denkleminde x yerine $a^{[0]}$ de yazabiliriz. Böylece çıktımız olan \hat{y} değeri,

$$\hat{y} = a^{[2]} \quad (2.39)$$

Şeklinde tanımlanır [24].

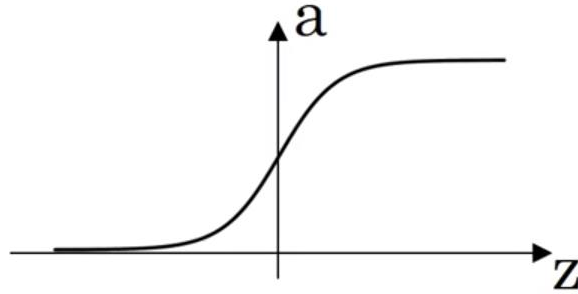
2.9 Aktivasyon Fonksiyonları

Yukarıdaki örneklerde de daha önce kullanmış olduğumuz (Şekil 2.3 Sigma Aktivasyon Fonksiyonu: σ) aktivasyon fonksiyonu bir nöronun ya da bizim kullanımımıza göre düğümün aktif olup olmayacağına karar veren ve çıkışın hangi sınıf değerine ya da öğrenilen bir özelliğe ait olduğunu belirtmek için kullandığımız bir fonksiyondur.

Eğer bir aktivasyon fonksiyonu kullanmasaydık sürekli olarak doğrusal bir çıkış üreten bir sinir ağına sahip olurduk. Bu durumda sinir ağıımız tıpkı lineer regresyon davranacak ve doğrusal olmayan fonksiyonların karşılık geldiği özellikleri öğrenemeyecekti. Özellikle bir veya daha fazla gizli katmana sahip olduğumuz yapılarıdaki ara katmanlarda lineer olamayan aktivasyon fonksiyonları kullanmamız kuracağımız modelin verilen girdiden daha fazla özellik öğrenmesine katkıda bulunur. Bununla birlikte çıktı katmanında nihai sınıflandırma esnasında lineer aktivasyon fonksiyonu kullanmak daha doğrudur.

Şimdi çok kullanılan bazı aktivasyon fonksiyonlarını inceleyelim.

2.9.1 Sigmoid Fonksiyonu



Şekil 2. 10 Sigmoid Fonksiyonu [29]

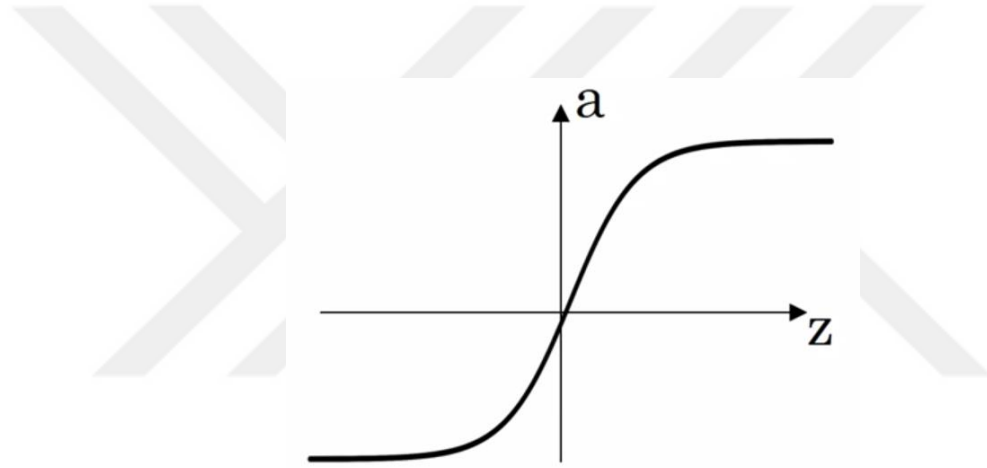
Bu grafiğe ait olan sigmoid fonksiyonu şöyle tanımlanır,

$$g(z) = \frac{1}{1 + e^{-z}} \quad (2.40)$$

Bu fonksiyon ismini grafikten de görüleceği üzere S harfine (Sigma) benzerliğinden dolayı almıştır. 0 ile 1 arasında değerler alır ve sinir ağlarında genellikle ikili (binary) çıktı alınması gereken durumlarda kullanılır.

Z eksenini boyunca ilerlendiğinde artık a değerinin fazlaca değişmediğini ve 1 değerine yakınsadığı görülmektedir. Bu durum bu bölgelerde alınan türevin çok küçük veya 0'a yakınsaması sonucunu doğurur ki bu duruma gradyanların ölmesi ya da kaybolması (vanishing gradient) adı verilir. Bu durumda geri yayılım esnasında ortaya çıkan öğrenmenin yavaşlamasına veya durmasına sebebiyet verir. Ayrıca lokal minimuma yakalanma olasılığı da çok fazladır.

2.9.2 Tanjant Hiperbolik Fonksiyonu



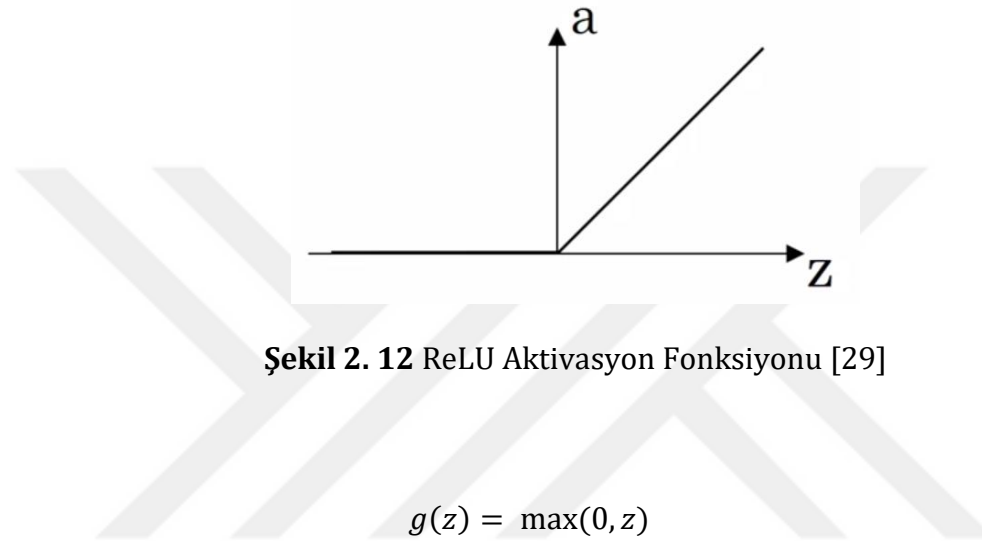
Şekil 2. 11 Tanjant Aktivasyon Fonksiyonu [29]

$$g(z) = \frac{e^z - e^{-z}}{e^z + e^{-z}} \quad (2.41)$$

Görüldüğü üzere tanjant hiperbolik fonksiyonu da sigmoid fonksiyonuna çok benzerlik göstermektedir. Bu fonksiyon sigmoid fonksiyonundan farklı olarak +1 ile -1 aralığında bulunmaktadır.

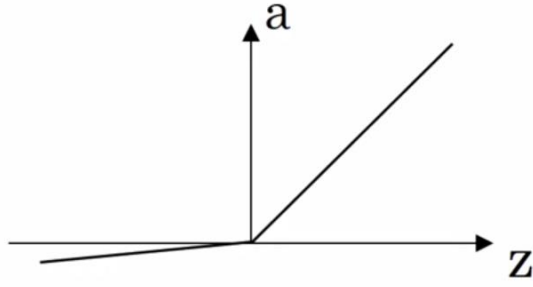
Bu fonksiyonda görüleceği üzere z değerinin büyümesi veya küçülmesi durumlarında fonksiyonun uç kısımlarında türevlerin sıfıra yakınsaması sorunu devam etmektedir. Fakat sigmoid fonksiyonundan farklı olarak daha büyük bir aralığa sahip olmasından dolayı geri yayılım esnasında daha iyi öğrenme ve sınıflandırma sonucu elde edilebilmektedir.

2.9.3 ReLU (Rectified Linear Unit) Fonksiyonu



ReLU fonksiyonu 0 ile $+\infty$ arasında değerler alabilir bu haliyle Sigmoid ve Tanjant fonksiyonlarından farklı olarak sinir ağındaki tüm nöronları aynı anda aktif etmez. Örneğin değeri negatif olan nöronlar aktive olmazlar. Bu ise hesaplama, hız ve verimlilik açısından bizlere avantaj sağlar. Sinir ağı içinde düşük aktivasyon yükü getirmesinden dolayı çok katmanlı sinir ağlarında daha çok tercih edilir. Fakat geri yayılım sırasında bu fonksiyonunda negatif kısımlarında sıfır değer alan kısımlarının türevlerinin de sıfır olması öğrenmenin olmayacağını ve parametrelerin güncellenemeyeceği sonucunu doğurur. Bu problemin üstesinden gelebilmek için Sızıntı (Leaky) ReLu geliştirilmiştir.

2.9.4 Sızıntı (Leaky) ReLU Fonksiyonu



Leaky ReLU

Şekil 2. 13 Sızıntı (Leaky) ReLU Aktivasyon Fonksiyonu [29]

$$g(z) = \max(0.01z, z) \quad (2.43)$$

Bu fonksiyona ReLU dan farklı olarak negatif değerlerin sıfır olmasını engellemek için 0.01 değerinde ve sızıntı olarak adlandırılan bir çarpan eklenmiştir. Böylece negatif değerler sıfıra yakın fakat sıfır olmadıkları için geri yayılım esnasında öğrenme gerçekleşmektedir.

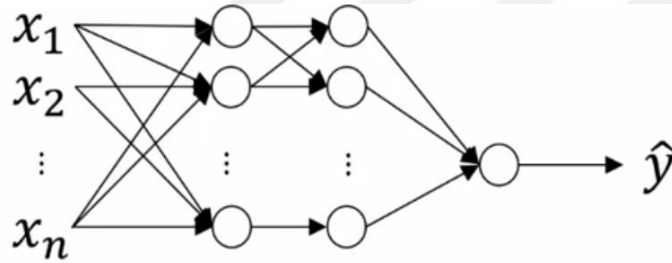
2.10 Evrişimsel Sinir Ağları

Günümüzde artan teknoloji ile birlikte kullandığımız verilerin de çeşitliliği, boyutu ve kalitesi de artmaktadır. Tamda bu noktada kullanmış olduğumuz yapay sinir ağları bu verileri sınıflandırma ve işleme konusunda yetersiz kalmaya başlamıştır. Bunu aşağıdaki örnek ile açıklamaya çalışalım,



Şekil 2. 14 64x64x3 Boyutunda Renkli Resim [29]

Şekil 2.14 de 64x64x3 boyutunda bir kedi resmi görülmektedir. Bunu klasik bir sinir ağı ile çözmek istediğimizi varsayalım,



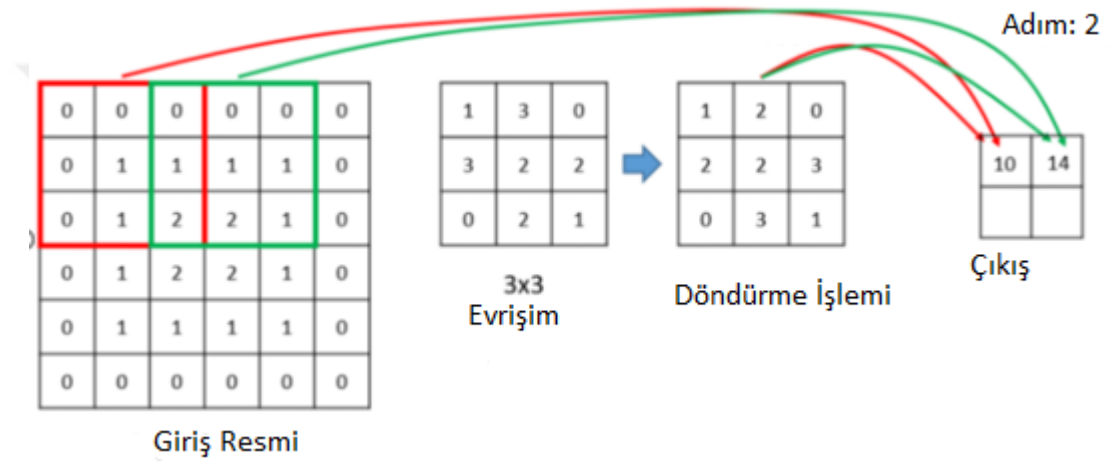
Şekil 2. 15 Çok Katmanlı Klasik Sinir Ağı [29]

Bu durumda elimizdeki 64x64x3 boyutundaki resmi düzleştirerek sinir ağına girdi olarak vermek istersek, vereceğimiz x girdisinin boyutu 12288 olacaktır. Yani 12288 adet x girdisine ($x_1, x_2, \dots, x_{12288}$) sahip oluruz. Bu yalnızca 64x64 boyutunda renkli bir resim için geçerlidir. Eğer inceleyeceğimiz resmin boyutu biraz daha artıp 1000x1000x3 boyutunda bir resim olursa,



Şekil 2. 16 1000x1000x3 Boyutunda Renkli Resim [29]

Bu durumda ağırlık parametresi olarak kullandığımız W 'nın boyutu 3 milyona olacaktır. Tabii ki günümüzde bundan çok daha boyutlu resimler ve veriler mevcuttur. İşte bu durumun üstesinde gelebilmek için 1998 yılında Yann LeCun ve ark. tarafından yayınlanan makalede [31] evrişimsel sinir ağları kullanılmıştır. Evrişimsel sinir ağları ilk olarak Nörofizyolog olan David Hubel ve Torsten Wiesel'in [32][33] 1959 yılında kediler üzerinde yaptıkları çalışmalar ile görüntü tanımda kullanılmış ve sonrasında Yann LeCun ve arkadaşlarının yaptıkları çalışmalar ile daha da geliştirilmiştir.



Şekil 2. 17 İki Boyutlu Evrişim Örneği [34]

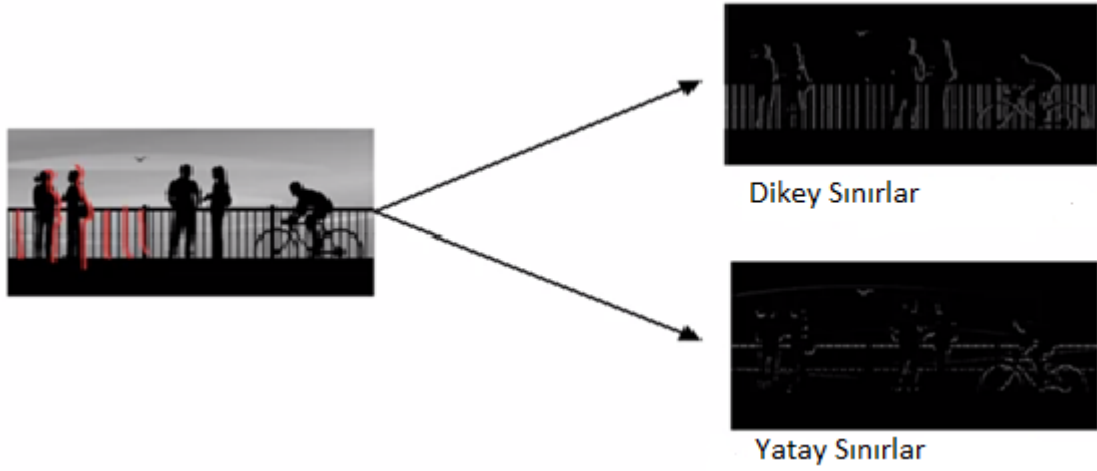
Yukarıda evrişim işleminin nasıl yapıldığına dair bir örnek görüntü sunulmuştur. Burada giriş resmimiz üzerine bizim belirleyeceğimiz boyutlarda bir filtre seçerek giriş resmi ile bu filtreyi evrişim işlemine tabi tutarız. Bu sayede filtreye bağlı olarak giriş resmimize ait çeşitli özellikler elde ederiz.



Şekil 2. 18 Evrişim ile Özellik Belirleme [34]

Şekil 2.18 den de görüleceği üzere bir yüz resmine çeşitli filtreler ile evrişim işlemi, uygularsak her katmanda bu resme ait yatay ve dikey sınırlar ve yüze ait diğer

detaylar öğrenilebilir. Veya sadece aşağıdaki gibi bir resimdeki yatay ve dikey sınırları görmek isteyebiliriz [30].



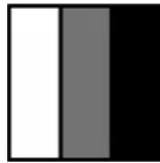
Şekil 2.19 Bir Resimdeki Yatay ve Dikey Sınırlar [34]

2.10.1 Dikey Sınır Belirleme

Evrişim işleminde bir resmin sınırlarını belirlemek çok önemlidir. Bir resmin içerisinde birden fazla obje olabileceğinden bu objeleri ayırmak ve tespit edebilmek için sınırlarını tespit etmeliyiz. Şimdi dikey sınır belirleme işlemini inceleyelim, Bu işlemi yapabilmek için yaygın olarak kullanılan ve Prewitt adı verilen şu dikey kenar belirleme filtresini kullanabiliriz [35],

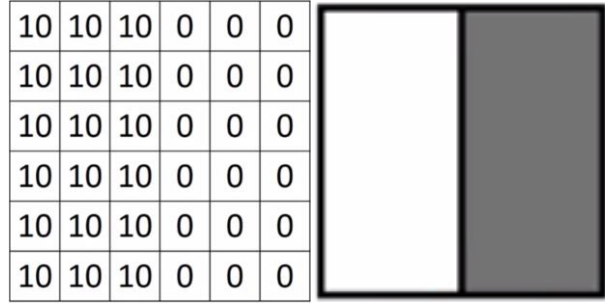
$$\begin{bmatrix} 1 & 0 & -1 \\ 1 & 0 & -1 \\ 1 & 0 & -1 \end{bmatrix} \quad (2.44)$$

Bu filtrenin görünümünü şu şekilde belirtebiliriz,



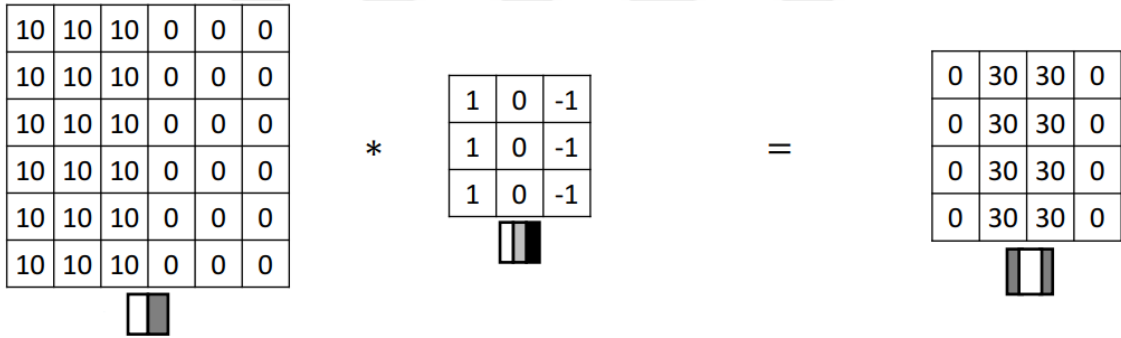
Şekil 2.20 Dikey Kenar Belirleme Filtresi [34]

Şimdi bunu aşağıdaki gibi bir tarafı aydınlık diğer tarafı karanlık olan bir resim üzerine uygulayalım,



Şekil 2. 21 Aydınlık ve Karanlık Taraflı Örnek Resim [34]

$$\begin{array}{cccccc}
 10 & 10 & 10 & 0 & 0 & 0 \\
 10 & 10 & 10 & 0 & 0 & 0 \\
 10 & 10 & 10 & 0 & 0 & 0 \\
 10 & 10 & 10 & 0 & 0 & 0 \\
 10 & 10 & 10 & 0 & 0 & 0 \\
 10 & 10 & 10 & 0 & 0 & 0
 \end{array}
 \begin{array}{ccc}
 * & 1 & 0 & -1 \\
 & 1 & 0 & -1 \\
 & 1 & 0 & -1
 \end{array}
 =
 \begin{array}{cccc}
 0 & 30 & 30 & 0 \\
 0 & 30 & 30 & 0 \\
 0 & 30 & 30 & 0 \\
 0 & 30 & 30 & 0
 \end{array}
 \quad (2.45)$$



Şekil 2. 22 Dikey Kenar Evrişim İşlemi [34]

Bu işlemi resimdeki sayıları yer değiştirerek (dikey simetri uygulayarak) da uygulayabiliriz,

X ve y ekseninde dikey ve yatay sınırları belirlemek için kullanacağımız filtreleri şu şekilde yazabiliriz,

-1	0	+1
-2	0	+2
-1	0	+1

G_x

+1	+2	+1
0	0	0
-1	-2	-1

G_y

Şekil 2. 25 Dikey ve Yatay Eksen Sobel Filtreleri [34]

Bu durumda her bir matris için filtre hesabı şu şekilde olur,

$$|G_x| = |-P_1 + P_3 - 2P_4 + 2P_6 - P_7 + P_9| \quad (2.48)$$

$$|G_y| = |P_1 + 2P_2 + P_3 - P_7 + 2P_8 - P_9| \quad (2.49)$$

Bu filtre kullanılarak oluşturulan pikselin değeri,

$$|G| = |G_x| + |G_y| \quad (2.50)$$

İle hesaplanabilir. Bu hesap için kullanılan diğer formül ise

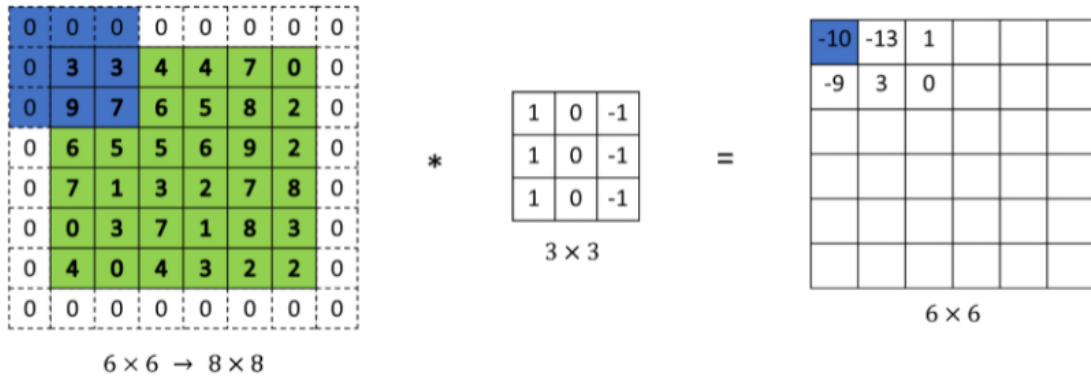
$$|G| = \sqrt{G_x^2 + G_y^2} \quad (2.51)$$

Şeklinde ifade edilir. [30]

2.10.2 Dolgulama (Padding)

Yukarıdaki işlemlere dikkat edilirse şekil 2.23 de 6x6 boyutlu bir giriş resmine 3x3 boyutlu bir filtre ile evrişim işlemi uygulanırsa sonuçta 4x4 boyutunda bir resim ortaya çıkmaktadır. İşte bu çıkış boyutunu ayarlayabilmek için kullandığımız yöntemlerden biri piksel ekleme olarak da adlandırılan dolgulama (padding) işlemidir.

Eğer bu giriş resminin her bir yanına 1 dolgulama yaparak yani sıfır piksel ekleyerek genişletecek ve daha sonra 3x3 lük bir filtre ile evrişim işlemine sokarsak,



Şekil 2. 26 Dolgulama İşlemi Yapılmış Evrişim İşlemi [34]

Görüleceği üzere çıkış matrisimizin boyutu değişmemektedir. Bu işlemi matematiksel olarak ifade edip formülleştirmek için şu tanımlamaları yapalım,

Dolgulama Sayısı: p

Filtre Boyutu: f

Giriş Matrisinin Boyutu: n

Bu durumda çıkış matrisinin boyutunu şu şekilde yazabiliriz,

$$n - f + 1 \times n - f + 1 \quad (2.52)$$

Böylece yapılan işlemleri şu şekilde toparlayabiliriz,

Tablo 2. 1 Evrişim Sonrası Çıkış Boyutu Hesaplama [34]

Çıkış Resminin Boyutu	Filtre Boyutu	Dolgulama	Çıkış Resminin Boyutu
$n \times n$	$f \times f$	p	$n - f + 1 \times n - f + 1$
6×6	3×3	0	6×6

Eğer evrişim işleminde giriş resminin boyutu değişmiyor ise bu evrişim işlemine Geçerli Evrişim (Valid Convolutions), giriş resminin boyutu çıkışta aynı olacak şekilde yapılan dolgulama ile oluşturulan evrişime ise Aynı Evrişim (Same Convolutions) adı verilir [30].

2.10.3 Kaydırma (Stride)

Tıpkı dolgulama(padding) gibi kaydırma(stride) da evrişimsel sinir ağlarında kullanılan ve çıkış resminin boyutunu etkileyen temel yapılardan biridir. Şimdi bu işlemin nasıl yapıldığını inceleyelim,

Aşağıdaki şekildeki gibi 7x7 boyutunda bir giriş resmi ile 3x3 boyutunda bir filtreyi evrişim işlemine tabi tutalım,

2	3	7	4	6	2	9
6	6	9	8	7	4	3
3	4	8	3	8	9	7
7	8	3	6	6	3	4
4	2	1	8	3	4	6
3	2	4	1	9	8	3
0	1	3	9	2	1	4

 *

3	4	4
1	0	2
-1	0	3

 =

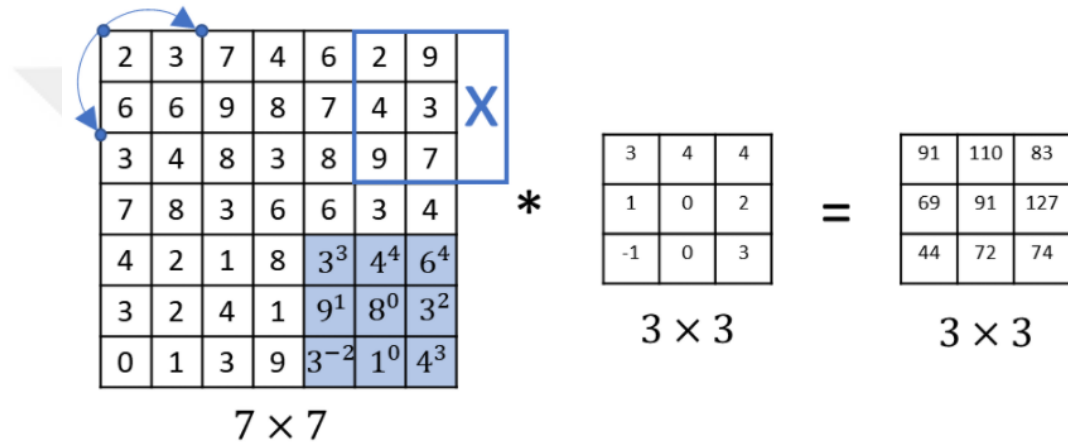
Şekil 2. 27 Evrişim İşlemi [34]

Eğer bu işlemi kaydırması 2 olacak şekilde yapmak istersek filtreyi resim üzerindeki ilk yerleşimden sonra sağa doğru 2 kez kaydıldıktan sonra yeniden işleme sokmalıyız,

2 ³	3 ⁴	7 ⁴	4	6	2	9	2	3	7 ³	4 ⁴	6 ⁴	2	9
6 ¹	6 ⁰	9 ²	8	7	4	3	6	6	9 ¹	8 ⁰	7 ²	4	3
3 ⁻¹	4 ⁰	8 ³	3	8	9	7	3	4	8 ⁻¹	3 ⁰	8 ³	9	7
7	8	3	6	6	3	4	7	8	3	6	6	3	4
4	2	1	8	3	4	6	4	2	1	8	3	4	6
3	2	4	1	9	8	3	3	2	4	1	9	8	3
0	1	3	9	2	1	4	0	1	3	9	2	1	4

Şekil 2. 28 İki Kaydırmalı Evrişim İşlemi [34]

Bu şekilde işleme devam edersek aşağıdaki sonucu elde ederiz,



Şekil 2. 29 İki Kaydırmalı Evrişim İşlemi Sonucu [34]

Bu şekilde yapılan işleme istersek dolgulama da ekleyebiliriz. Nihai olarak kaydırma adımını da içeren ve çıktı boyutunu veren (2.52) denklemini şu şekilde düzenleyebiliriz,

Kaydırma (stride): s

$$\frac{n-f+2p}{s} + 1 + 1 \times \frac{n-f+2p}{s} + 1 \quad (2.53)$$

Böylece daha önce oluşturduğumuz tablo (2.1)'i aşağıdaki gibi yeniden yazabiliriz,

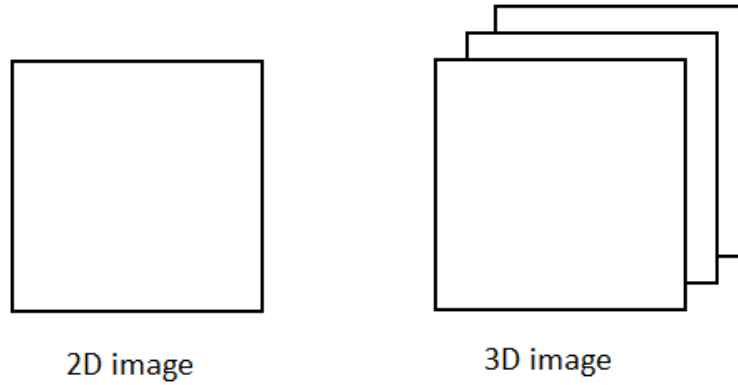
Tablo 2. 2 Kaydırma Adımlı Evrişim Sonrası Çıkış Boyutu Tablosu [34]

Çıkış Resminin Boyutu	Filtre Boyutu	Dolguam a	Kaydırma a	Çıkış Resminin Boyutu
$n \times n$	$f \times f$	p	s	$\frac{n - f + 2p}{s} + 1 \times \frac{n - f + 2p}{s} + 1$
7×7	3×3	0	2	3×3

Böylece girişi 7x7 olan bir resme 0 dolgulama ve 2 kaydırma ile 3x3'lük bir filtreyi evrişim işlemine sokarsak sonuçta 3x3'lük bir resim elde edilmektedir [30].

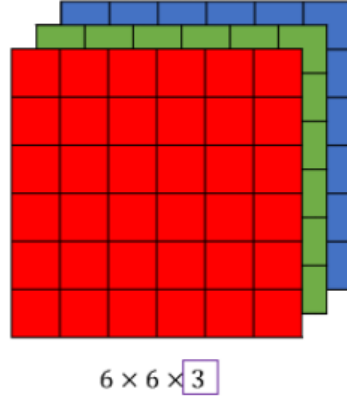
2.10.4 Renkli Resimlerde Evrişim

Daha önce yaptığımız evrişim işlemleri 2 boyutlu siyah-beyaz (gray scale) resimler üzerindedir. Şimdi 3 boyutlu resimler üzerinde de bu işlemin nasıl olacağını inceleyelim,



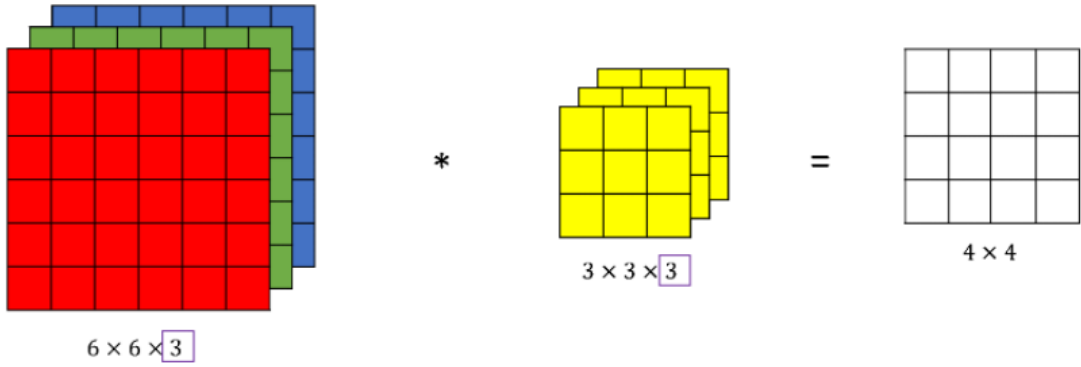
Şekil 2. 30 İki Boyutlu Siyah-Beyaz ve Üç Boyutlu Renkli Resim Örneği [34]

Renkli resimlerin iki boyutlu siyah-beyaz resimlerden farklı olarak her bir renk kanalına (kırmızı-yeşil-mavi) karşılık gelecek şekilde 3 boyutlu bir yapıya sahiptir.



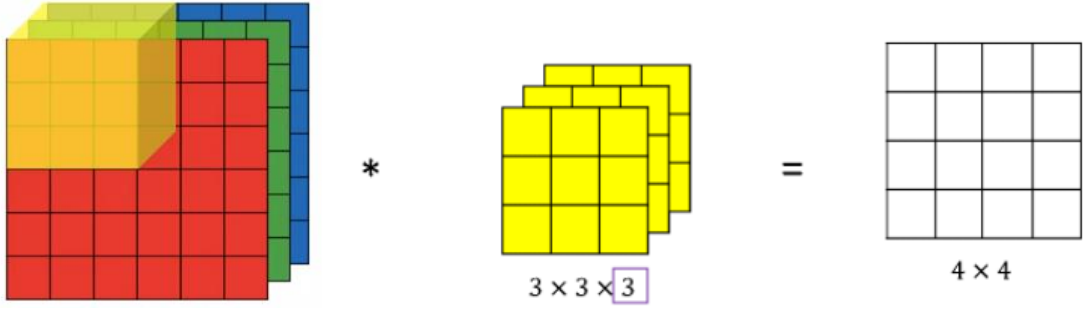
Şekil 2. 31 6x6x3 Boyutlu Renkli Resim Örneği [34]

(2.31) da ki şekilde görüleceği üzere 6x6 boyutlu bir renkli resim RGB [37] olarak adlandırılan 3 adet 6x6 boyutlu bir yapıya sahiptir. Bu 6x6x3 boyutlu renkli resmi 3x3 boyutlu bir filtre ile evrişim işlemine sokabilmemiz için bu filtrenin de resimde olduğu gibi 3 boyuta yani kanal başka bir ifade ile kanal sayısına sahip olması gereklidir.



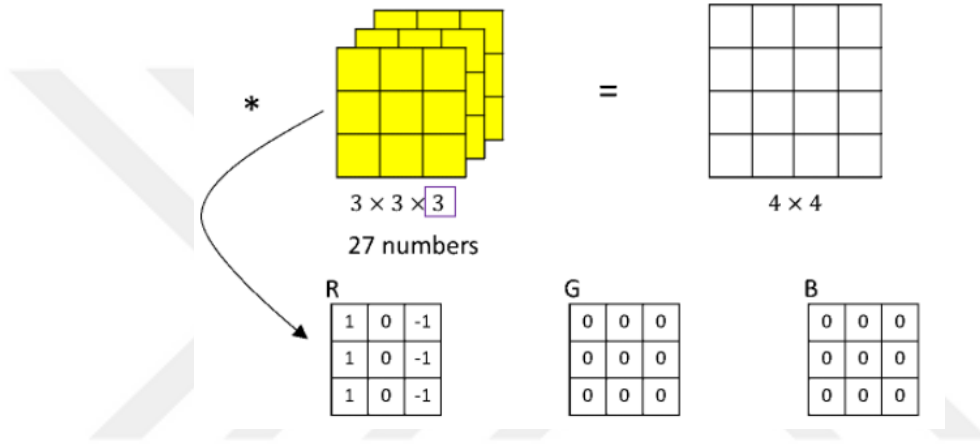
Şekil 2. 32 Renkli Resmin Evrişim Örneği [34]

Bu evrişim işlemi yapılırken filtrenin her bir kanalı (tabakası) resmin karşılık gelen kanalları ile üst üste getirilerek çarpılır ve çıkan sonuçlar toplanır.



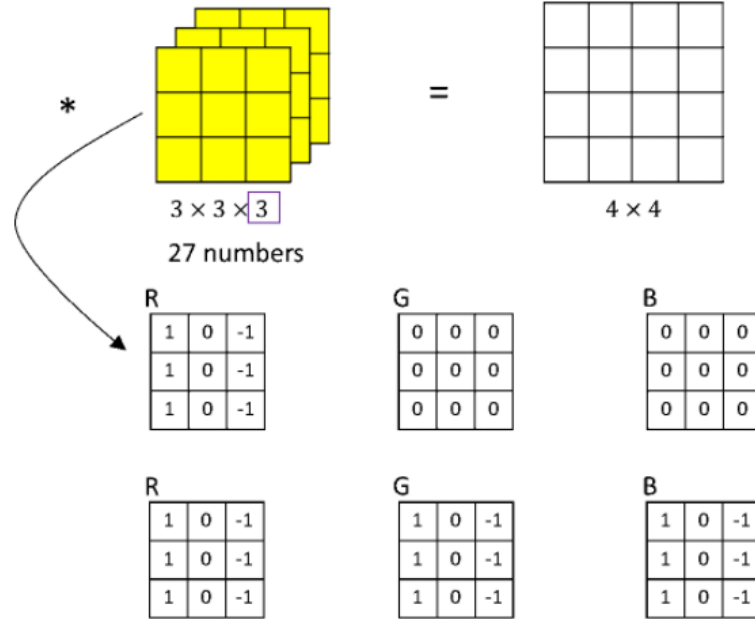
Şekil 2. 33 Renkli Resmin Evrişim Hesaplama Örneği [34]

Elimizdeki filtrenin de 3 kanalı olması sayesinde dilersek resmimizin kırmızı kanalındaki dikey ve yatay çizgileri tespit edebiliriz.



Şekil 2. 34 Renkli Resmin Evrişim Hesaplama Örneği 2 [34]

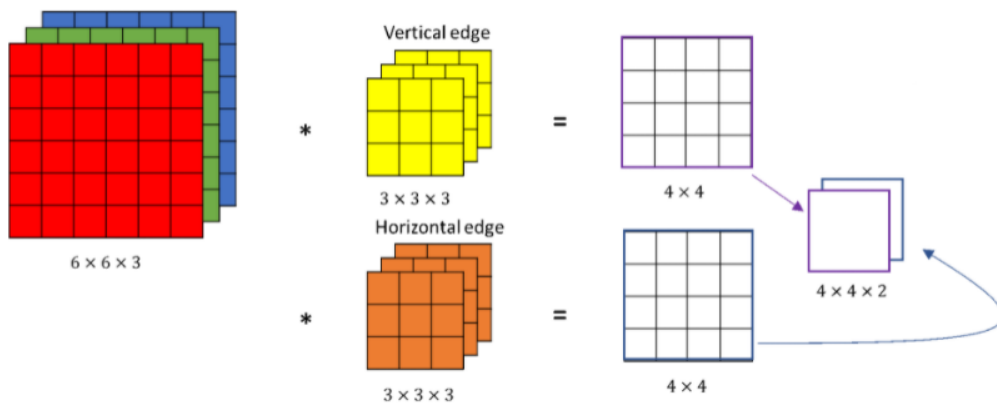
Bunun için filtremizdeki kırmızı kanala karşılık gelen ilk bölüme şekil (2.34) de görüleceği üzere karşılık gelen sınır değer özellikleri yazılır ve diğer renk kanallarıyla ilgilenmiyorsak bu kanallara karşılık gelen değerler sıfır ile doldurularak işlem yapılırsa resmimizde sadece kırmızı kanaldaki dikey ya da yatay özellikleri tespit etmiş oluruz.



Şekil 2. 35 Renkli Resmin Evrişim Hesaplama Örneği 3 [34]

Ya da şekil (2.35) deki gibi tüm kanallardaki yatay veya dikey özellikleri tespit etmek istersek o zaman bütün filtrelerdeki kanalları aynı şekilde seçerek evrişim işlemine tabi tutabiliriz.

Evrişim işleminin bize sağladığı kolaylıklar bunlarla sınırlı değildir, dilersek bir resimdeki tüm kanallarda hem yatay hem dikey hem de daha fazla özellik tespit etmek için filtre sayımızı artırabiliriz.



Şekil 2. 36 Renkli Resmin Evrişim Hesaplama Örneği 4

Şekil (2.36) da görüleceği üzere renkli resmimizde hem yatay hem de dikey özellik tespit etmek için iki farklı 3 boyutlu filtre kullanılmıştır. Çıkan sonuç resmimizin

boyutu $4 \times 4 \times 2$ şeklinde olmaktadır. En sondaki 2 sayısı kullanılan filtre sayısıdır. Bu örnekte 3×3 boyutunda yatay ve dikey özellik belirleyen iki adet filtre kullanılmıştır.

Bu çıkış resminin boyutunu da matematiksel olarak ifade etmek istersek,

l : Katman Sırası

$f^{[l]}$: l . Katmanda Kullanılan Filtre Boyutu

$s^{[l]}$: l . Katmanda Kullanılan Kaydırma Sayısı

$p^{[l]}$: l . Katmanda Kullanılan Dolgulama Sayısı

$n_c^{[l]}$: l . Katmandaki Resimde kullanılan kanal sayısı

n_H : Resmin yüksekliği

n_W : Resmin genişliği

Bu tanımlamalar ile birlikte giriş resmimiz,

Giriş Resmi: $n_H^{[l-1]} \times n_W^{[l-1]} \times n_c^{[l-1]}$

İse çıkış resminin boyutu,

$$n_H^{[l]} = \frac{n_H^{[l-1]} + 2p^{[l]} - f^{[l]}}{s^{[l]}} + 1, n_W^{[l]} = \frac{n_W^{[l-1]} + 2p^{[l]} - f^{[l]}}{s^{[l]}} + 1 \quad (2.54)$$

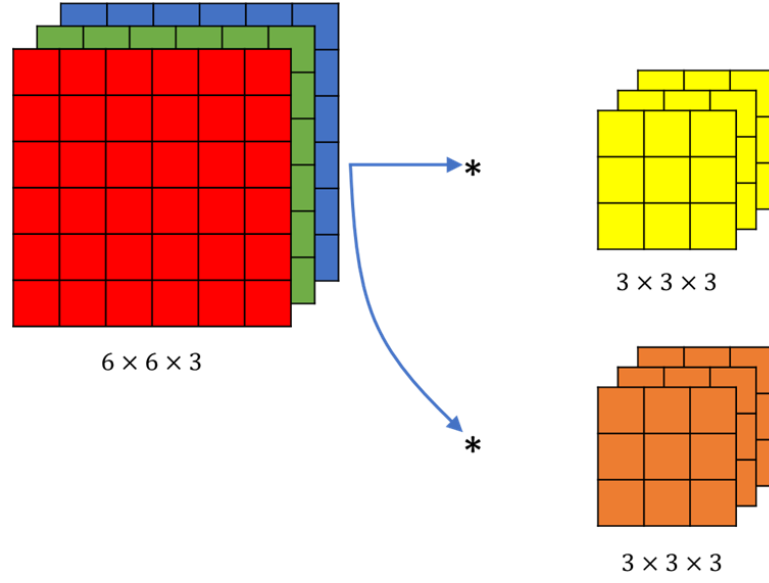
$$\text{Çıkış Resmi: } n_H^{[l]} \times n_W^{[l]} \times n_c^{[l]} \quad (2.55)$$

şeklinde elde edilir [30].

2.10.5 Tek Katmanlı Evrişim İşlemi

Şekil 2.33 deki gibi tasarlamış olduğumuz evrişimli yapıyı bir sinir ağı şeklinde ifade etmeye çalışalım,

Elimizdeki evrişimli sistemin şu şekilde olduğunu düşünersek,

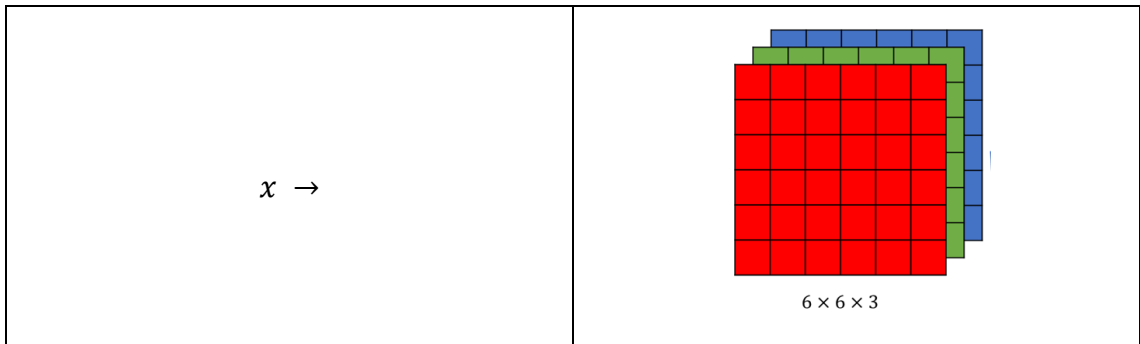


Şekil 2. 37 Renkli Resme Uygulanan Evrişim Görseli [34]

Elimizdeki bu yapıyı daha önce incelediğimiz klasik sinir ağı modeline dönüştürebilmek için önceden yazdığımız nöronlar için kullanılan temel hesaplama formülümüzü yeniden hatırlayalım,

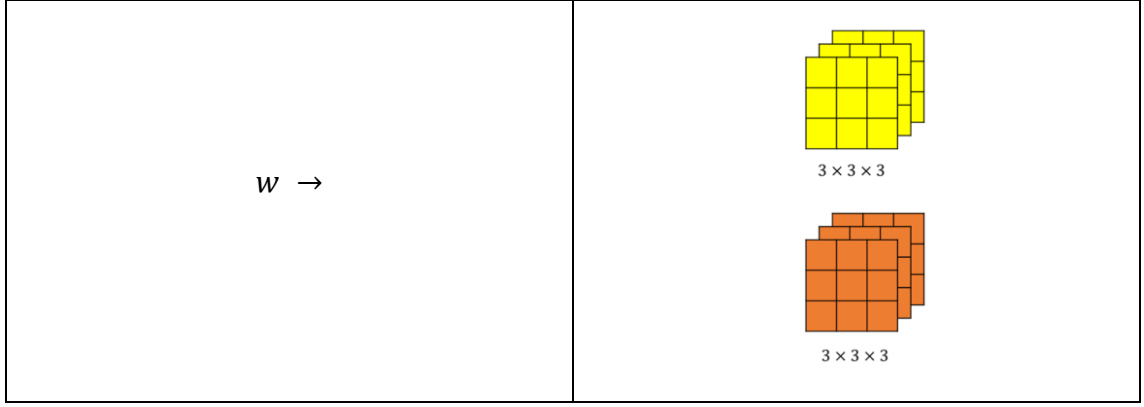
$$z = w^T \cdot x + b \quad (2.18)$$

Bunun için denklem 2.18 deki sinir ağı modelimizin bir nöronu için yapılan hesaplamayı göz önüne aldığımızda buradaki giriş verisi olan x ,



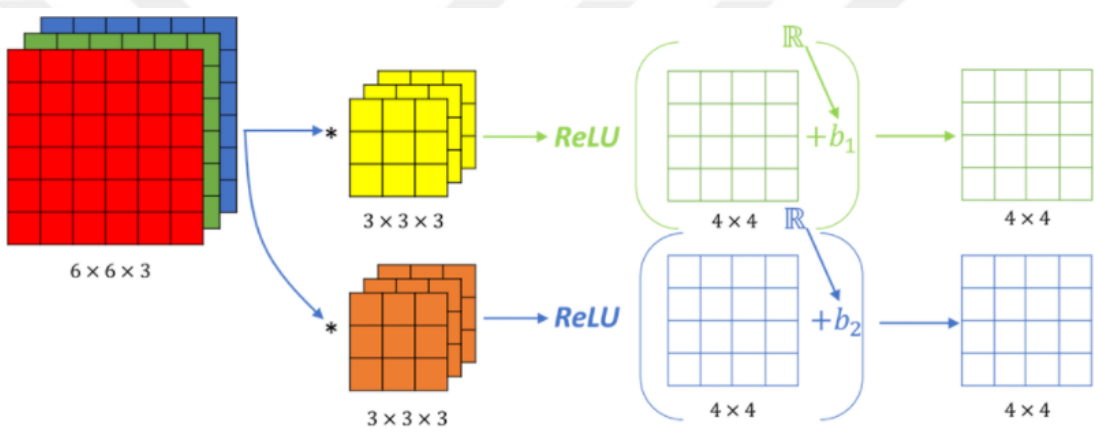
Şekil 2. 38 Evrişimli Sinir Ağı İçin Girdi Karşılaştırması [34]

Ağırlık parametremiz olan w ise,



Şekil 2. 39 Evrişimli Sinir Ağı İçin Ağırlık Parametresi Karşılaştırması [34]

Bu durumda modelimize eksik olarak sadece yanlılık şeklinde ifade edilen bias terimlerini eklemek kalmıştır,



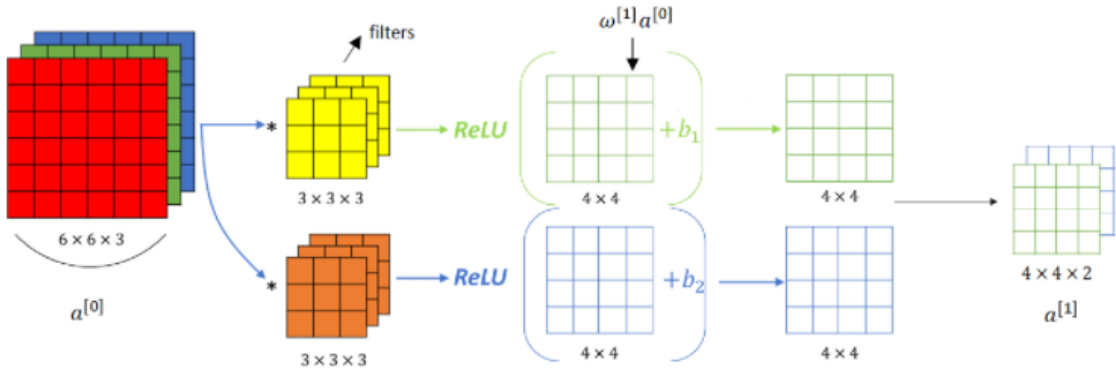
Şekil 2. 40 Tek Katmanlı Evrişimli Sinir Ağı Görseli

Şekil 2.40'daki gibi her bir filtreye karşılık gelecek şekilde b_1 ve b_2 yanlılık terimlerini ekleyerek modelimizi tamamlamış olduk. Bu yazdıklarımızı vektörel formda ifade edecek olursak,

$$z_1^{[1]} = W_1^{[1]T} x + b_1^{[1]} \quad (2.20)$$

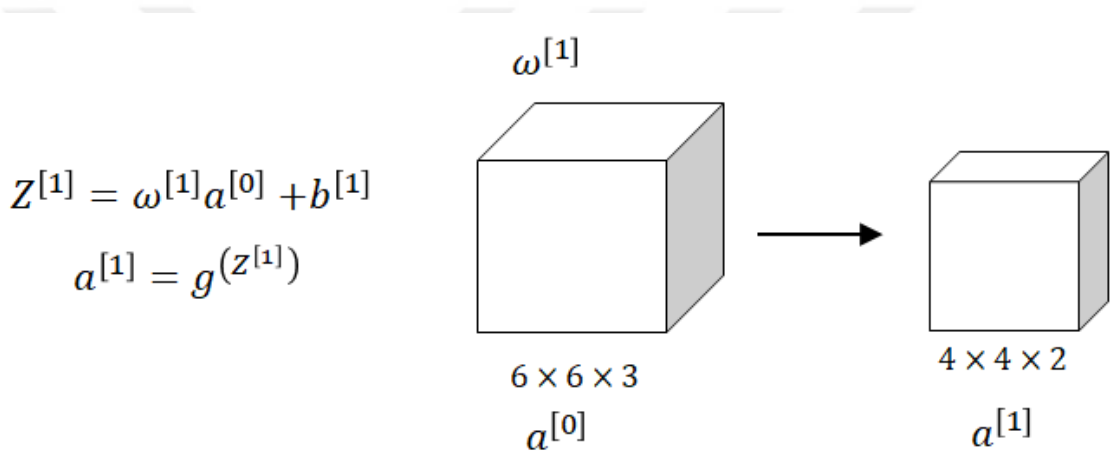
$$a_1^{[1]} = \sigma(z_1^{[1]}) \quad (2.21)$$

2.20 ve 2.21 denklemlerine uygun olarak şu şekilde gösterebiliriz,



Şekil 2. 41 Tek Katmanlı Evrişimli Sinir Ağı Vektörel Gösterimi 1

2.21 denklemindeki Sigmoid (σ) olarak özelleştirilmiş aktivasyon fonksiyonunu g şeklinde değiştirip genelleştirirsek,

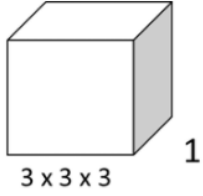


Şekil 2. 42 Tek Katmanlı Evrişimli Sinir Ağı Vektörel Gösterimi 2

Şekil 2.42 de ki gibi ifade edebiliriz [30].

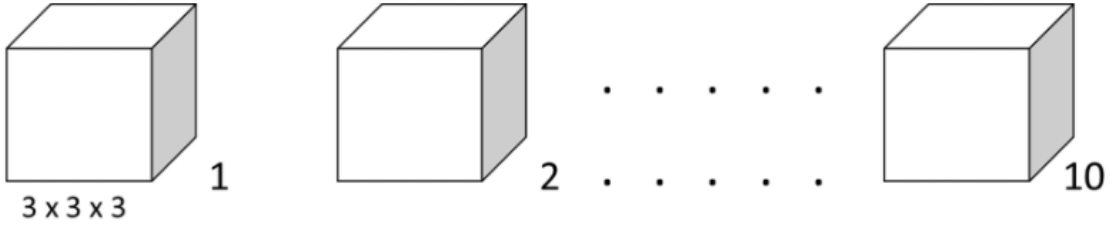
2.10.6 Tek Katmanlı Evrişim İçin Parametre Sayısı

Şekil 2.41 deki yapıyı düşünürsek elimizde 3x3x3 boyutunda toplam 2 adet filtre vardır. Bu filtreleri öğrenilmesi gereken parametreler yani w değeri olarak tanımlamıştık. Ayrıca bir de bunlara yanlılık terimi b_1 ve b_2 de eklemiştik. Bu durumda sadece tek bir filtre için düşünürsek,

	$3*3*3 = 27$ parametre $+ b_1$ (bias)yanlılık $= 28$ Parametre
---	--

Şekil 2. 43 Evrişimli Sinir Ağı Filtresi İçin Parametre Sayısı Hesaplama 1

Şekil 2.43'den görüldüğü üzere birinci filtre için 3x3x3 boyuttan 27 parametre ve 1 adette yanlılık (bias) terimi eklendiğinde toplam 28 parametre sayısı olmaktadır. Eğer 2 yerine 10 adet filtremiz olduğunu düşünürsek,

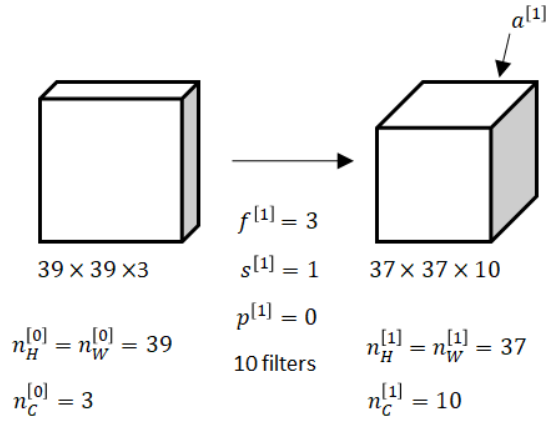


Şekil 2. 44 Evrişimli Sinir Ağı Filtresi İçin Parametre Sayısı Hesaplama 2 [34]

Bu sefer parametre sayımız $28 * 10$ filtreden toplam 280 olur. Buradaki önemli noktalardan biri de giriş resmimizin boyutu ne olursa olsun öğrenilecek parametre sayısı sadece uygulanan filtresin sayısına ve boyutuna bağlı olmasıdır. Bu durumda evrişimli sinir ağlarının klasik sinir ağlarına göre önemli bir üstünlüğü olarak öne çıkmaktadır. Ayrıca bu yapıda overfitting olarak adlandırılan aşırı öğrenme sorunu da azalmaktadır [30].

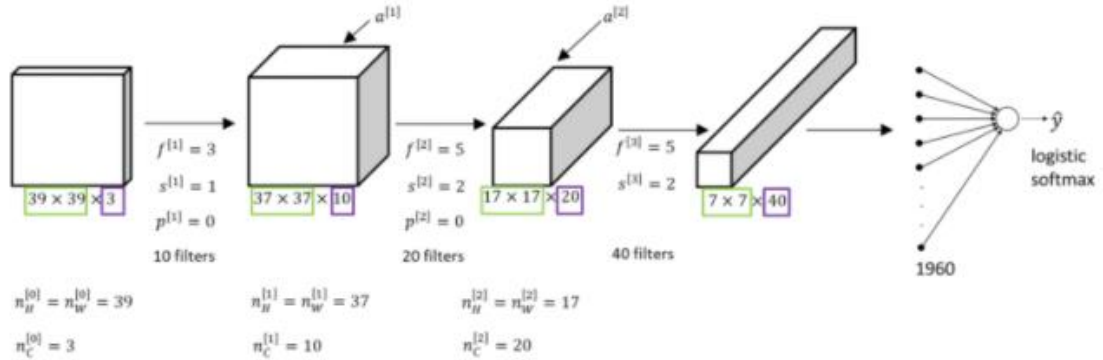
2.10.7 Tek Katmanlı Evrişimli Sinir Ağı

Şimdi giriş resmimizin boyutunu $39*39*3$ olarak alalım. Buna $3*3$ boyutunda 10 adet filtreyi 1 kaydırmalı ve sıfır dolgulu olacak şekilde evrişim işlemine tabi tutalım.



Şekil 2. 45 Evrişim İşlemi Örneği [34]

2.54 ve 2.55 numaralı denklemler uyarınca çıkış matrisimiz 37x37x10 boyutunda olacaktır. Bu şekilde devam ederek iki kez daha farklı boyut ve sayılarda filtre ile evrişim işlemi yapıp bunu klasik sinir ağına bağladığımızda sonuçta tek katmanlı evrişimli bir sinir ağı oluşmuş olacaktır,



Şekil 2. 46 Tek Katmanlı Evrişimli Sinir Ağı Örneği [34]

Son matrisimizin boyutu 7x7x40 olduğundan bunu klasik bir sinir ağına verebilmek için öncelikle düzleştirmemiz gerekmektedir. Düzleştirildiğinde sonuç, $7*7*40 = 1960$ olur. Böyle girdi boyutu 1960 olan klasik bir sinir ağı bağlantısı yapılarak sonuçların sınıflandırılması sağlanmış olur [30].

2.10.8 Havuzlama (Pooling) Katmanı

Evrişimli sinir ağları giriş resminin boyutlarını küçültmek için evrişim işleminden daha çok havuzlama katmanlarını kullanmaktadır. Bu sayede bu katmanlar hem hesaplamayı hızlandırır hem de bazı özelliklerin daha iyi öğrenilmesini sağlarlar.

Bu işlem ilk defa [38], Yann LeCun ver ark. tarafından [39] ve [31] numaralı makalelerde kullanılmıştır.

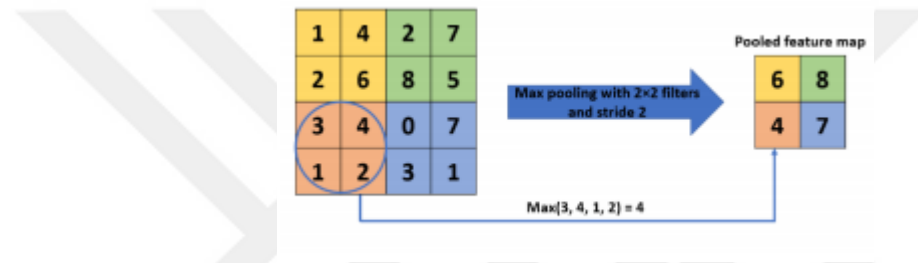
Genellikle iki çeşit havuzlama metodu kullanılır,

- 1) Maksimum Havuzlama (Max Pooling)
- 2) Ortalama Havuzlama (Average Pooling)

Şimdi öncelikle en büyük havuzlamaya bakalım.

En Büyük Havuzlama

Bunun için 4x4 boyutunda bir giriş resmimize 2x2 boyutunda ve 2 kaydırma değerine sahip bir filtre uyguladığımızı düşünelim.



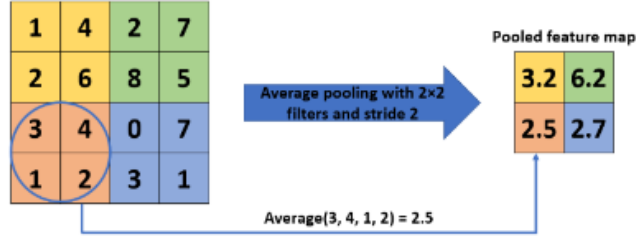
Şekil 2. 47 Tek Katmanlı Evrişimli Sinir Ağı Örneği [40]

Filtremizi giriş resmimize yerleştirdiğimiz karşılık gelen 2x2'lik alandaki en büyük değer alınarak çıktı resminin karşılık gelen alanına yazılır. Daha sonra kaydırma adımı 2 seçildiğinden filtremiz 2 adım sağa getirildiğinde bu sefer de bu bölgedeki en büyük değer alınır. İşte bu şekilde yapılan işlem sonucunda 2.54 denklemini uyarınca 2x2'lik bir çıktı elde edilir.

Bu işlem bize resmimizde özellik açısından vurgu yapılan alanların alınmasını garantiler. Ayrıca bu işlem esnasında öğrenilmesi gereken bir parametre de olmayışı ve bu işlemin pratikte iyi sonuçlar vermesi bu metodun kullanılmasını yaygınlaştırmıştır [40].

Ortalama Havuzlama

Bu işlem de maksimum havuzlamaya benzer şekilde çalışır ancak karşılık geldiği alanın maksimum değerini almak yerine ortalama değerlerini hesaplar.



Şekil 2. 48 Ortalama Havuzlama Örneği [40]

Bu işlem uygulamada maksimum havuzlamaya göre daha a kullanılır.

Görüleceği üzere havuzlama işleminde kullanılan parametreler şunlardır,

f : Filtre Boyutu

s : Kaydırma Adımı

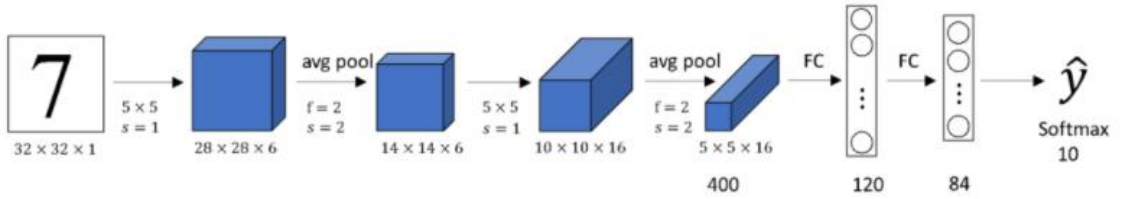
p : Dolgulama

Bu parametreleri belirleyip havuzlama yöntemini de seçtikten sonra yapılması gereken şey 2.54 denklemine göre çıktıyı hesaplamak olacaktır [40].

2.11 Klasik Evrişimli Sinir Ağları

2.11.1 LeNet – 5

Yann LeCun ve ark. [31] tarafından siyah beyaz el yazsısı rakamlarını tanımak için 1998 yılında geliştirilmiştir.



Şekil 2. 49 LeNet 5 Evrişimli Sinir Ağı Yapısı [34]

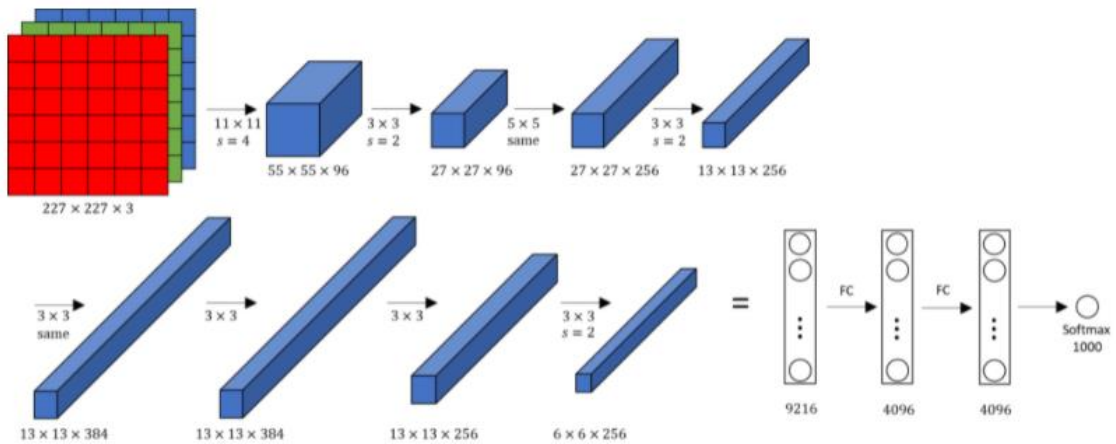
Giriş resimleri 32x32x1 boyutlu siyah beyaz resimlerden oluşmaktadır. Daha sonra bu resme 6 tane 5x5 boyutlu filtre 1 kaydırma değerine sahip olacak şekilde bir evrişim işlemi uygulanır. Daha sonra 28x28x6 boyutlu çıktıya 2x2 boyutlu filtre ile ortalama havuzlama işlemi uygulanır. Bu sefer çıktımızın boyutu 14x14x6 olur. Bu çıktıya 5x5 boyutlu 16 adet filtre 1 kaydırma değeri ile evrişim işlemine tabi tutulur. Son durumda oluşan çıktının boyutu 10x10x16 olur ve bu sefer 2x2 boyutlu filtre ile

tekrar ortalama havuzlama işlemine tabi tutulur. Tüm bu işlemlerin sonunda $5 \times 5 \times 16$ boyutunda bir çıktıya sahip oluruz. Daha sonra bu çıktıyı düzleştirerek önce 120 daha sonra bu da 84 uzunluklu nöronlu tam bağlantılı klasik sinir ağına bağlanır. En sonda 0'dan 9'a kadar 10 adet rakam bulunduğundan 10 farklı sınıflandırma yapan Softmax fonksiyonuna bağlanır.

Görüleceği üzere bu ağ yapısında sırasıyla evrişim - havuzlama - evrişim - havuzlama - tam bağlı katman - tam bağlı katman şeklinde bir sinir ağı yapısı kullanılmıştır. Bu yapıda soldan sağa doğru gidildikçe resim boyutu azalmakta fakat kanal sayısı artmaktadır. Bu şekilde daha derine gidildikçe resim hakkında daha detaylı özellikler öğrenilmektedir. Ayrıca bu sinir ağında yaklaşık 60.000 öğrenilmesi gereken parametre vardır ve bu klasik ağlar ile karşılaştırıldığında 1-100 milyon parametrenin yanında oldukça düşüktür [34].

2.11.2 AlexNet

Bu ağ yapısı Alex Krizhevsky ve ark. [41] tarafından 2017 yılında geliştirilmiştir.



Şekil 2. 50 AlexNet Evrişimli Sinir Ağı Modeli [34]

Bu modelde giriş olarak $227 \times 227 \times 3$ boyutlu renkli resimler kullanılmaktadır. Bu modelde sırasıyla,

1. Evrişim 1

Giriş resmi 11×11 boyutunda 96 adet filtre ile $s=4$ kaydırma parametresiyle evrişim işlemine sokulur. Bunun sonucunda $55 \times 55 \times 96$ boyutlu bir çıktı elde edilir.

2. Maksimum Havuzlama 1

55x55x96 boyutlu çıktımıza 3x3 boyutlu bir filtre ile kaydırma parametresi $s=2$ olacak şekilde maksimum havuzlama işlemi uygulanır. Böylece 27x27x96 boyutlu bir çıktı elde edilir.

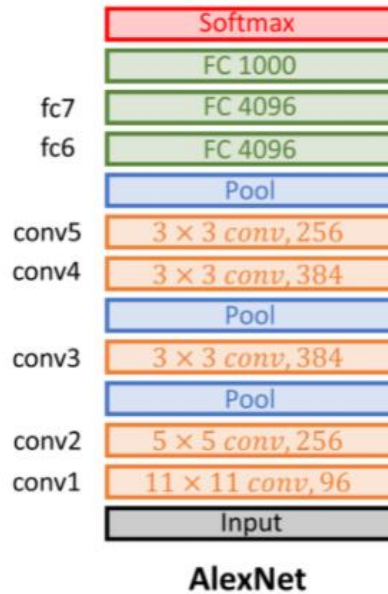
3. Evrişim 2

Daha sonra 27x27x96 boyutlu çıktımıza 5x5 boyutlu 256 adet filtre kaydırma parametresi $s=1$ aynı(same) olacak şekilde evrişim işlemi yaptırılır ve 27x27x256 boyutlu bir çıktı elde edilir.

4. Maksimum Havuzlama 2

Daha sonra 27x27x256 boyutlu çıktımıza 3x3 boyutlu bir filtreyi $s=2$ olacak şekilde maksimum havuzlama işlemine tabi tutarız. Bu işlemin sonunda 13x13x256 boyutlu bir çıktı elde ederiz.

Benzer şekilde diğer adımlarda uygulanarak 6x6x256 boyutlu çıktımız tam bağlı klasik sinir ağına bağlanarak sonunda 1000 sınıf için düzenlenmiş Softmax fonksiyonuna aktarılır.

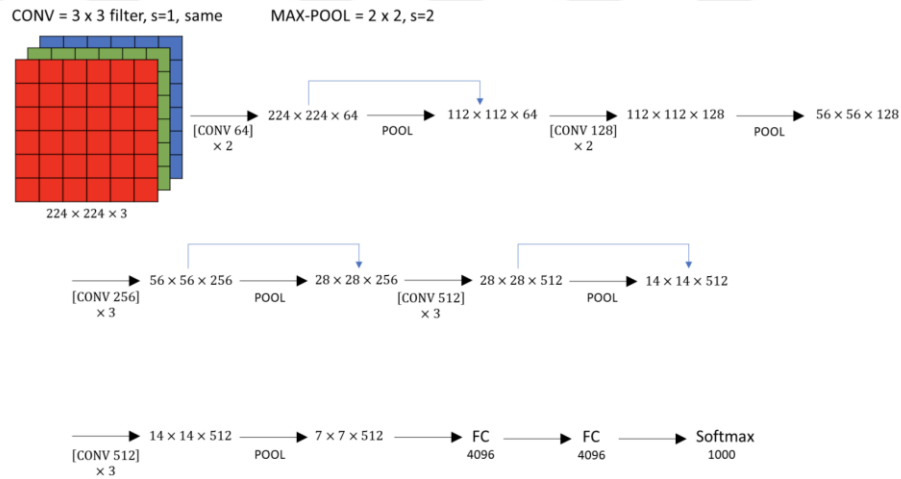


Şekil 2. 51 AlexNet Evrişimli Sinir Ağı Şeması [34]

Şekil 2.51 de yapılan tüm bu işlemler özet olarak sıralı şekilde gösterilmiştir. Bu model her ne kadar LeNet modeline benzese de ondan çok daha büyüktür. LeNet 60000 parametre içerirken AlexNet modelinde yaklaşık 60 milyon parametre mevcuttur. AlexNet modelinin bir başka öne çıkan yanı ise LeNet modelinden farklı olarak ReLu aktivasyon fonksiyonunu kullanarak daha iyi ve hızlı optimizasyon değerlerine ulaşmış olmasıdır. ImageNet ILSVRC yarışmasında sınıflandırma doğruluk oranını %74,3'ten %83,6'ya çıkararak önemli bir başarı elde etmiştir. [34]

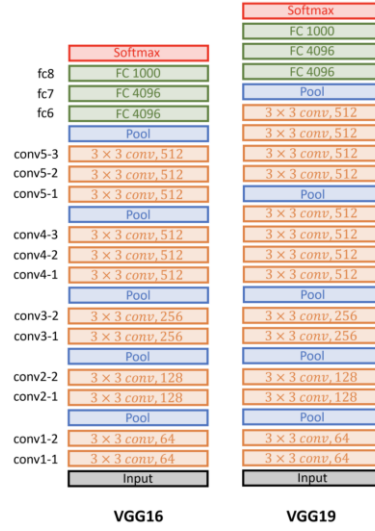
2.11.3 VGG 16 ve VGG 19

Karen Simonyan ve Andrew Zisserman tarafından [42] 2015 yılında geliştirilen bu model 224x224x3 giriş boyutu kullanmaktadır. 12 evrişim ve 4 havuzlama işlemi içeren model yapısında 3 adet de tam bağlı sinir ağı bulundurulur.



Şekil 2. 52 VGG 16 Evrişimli Sinir Ağı Modeli [34]

Bu model diğerlerinden farklı olarak evrişim esnasında 2 veya 3 kanallı filtreler kullanır. Diğerleri gibi soldan sağa gidildikçe boyut azalırken kanal sayısı artmaktadır. Bu modelin parametre sayısı ise yaklaşık 138 milyondur.

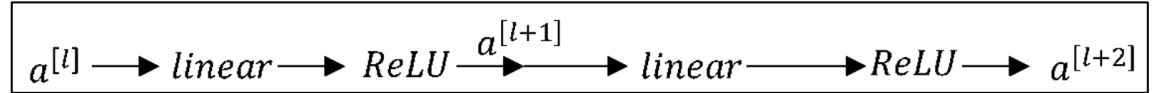


Şekil 2. 53 VGG 16-19 Evrişimli Sınır Ağ Modelleri 2 [34]

VGG 16 modelinden daha büyük olan VGG 19 modelinde ise diğerinden farklı olarak 16 katman yerine 19 katman kullanılmıştır. Şekil 2.53 de iki modelin karşılaştırılması görülmektedir [30].

2.11.4 ResNets

Kaiming He ve ark. [43] tarafından 2015 yılında geliştirilen bu model ile giriş katmanında elde edilen ve artık (residual) denen çıktı değerlerini ağın ilerleyen katmanlarına iletilerek eğitim esnasında türevlerin sıfırlanması sorununun önüne geçmiştir.



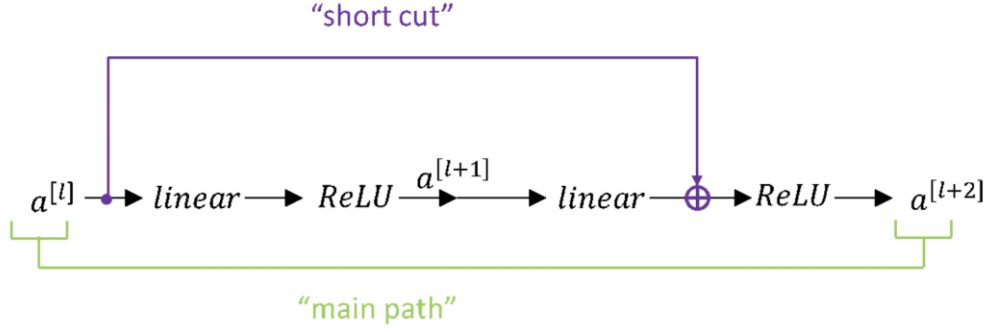
Şekil 2. 54 Ana Yol (Main Path) [43]

Daha önce incelediğimiz ve [43] çalışmasında Ana Yol(Main Path) olarak adlandırılan klasik ağ yapısı Şekil 2.54 de görüldüğü gibidir. Bu modelde $a^{[l+1]}$ ifadesini şu şekilde tanımlamıştık,

$$a^{[l+1]} = g(z^{[l+2]}) \quad (2.56)$$

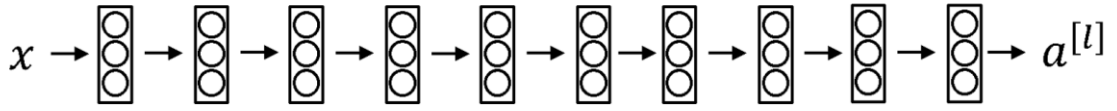
Kullanmış olduğumuz ResNet modelinde bu yapı Şekil 2.55 deki gibi olduğundan 2.56 denkleminin yerine,

$$a^{[l+1]} = g(z^{[l+2]} + a^{[l]}) \quad (2.57)$$

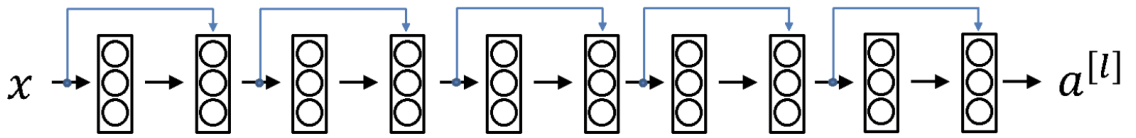


Şekil 2. 55 Ana Yol ve Kısa Yol Aktarımı [43]

$a^{[l]}$ değerinin eklenmiş hali kullanılır. Bu yapıyı şu gösterimle de ifade etmek mümkündür,

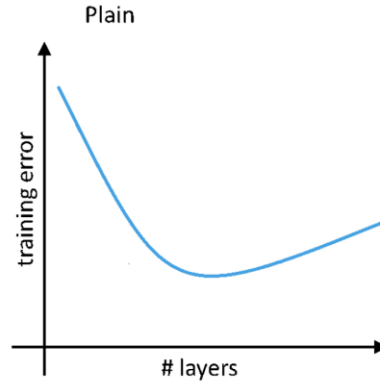


Şekil 2. 56 Düz Ağ Yapısı (Plain Network)



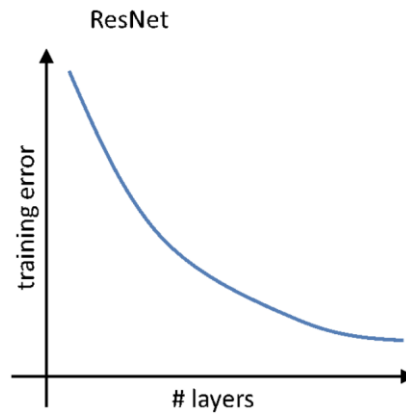
Şekil 2. 57 Artık Eklenmiş Ağ Yapısı (Residual Network)

Şekil 2.54'deki düz bir ağı eğitmek için kullandığımız gradient azalması yöntemi ile artan eğitim sayısı ve katman sayısı ile eğitim hatasının azaldığını gözlemleriz fakat bir süre sonra bu azalma sonlanarak bu sefer artış eğilimi gösterir.



Şekil 2. 58 Düz Ağ Yapısında Katman Sayısı ile Eğitim Hatası İlişkisi [34]

Fakat şekil 2.55'deki gibi bir ResNet ağ yapısı kullandığımızda artan katman sayısı ile birlikte hata oranının da hiç artmadan azaldığı görülmüştür.



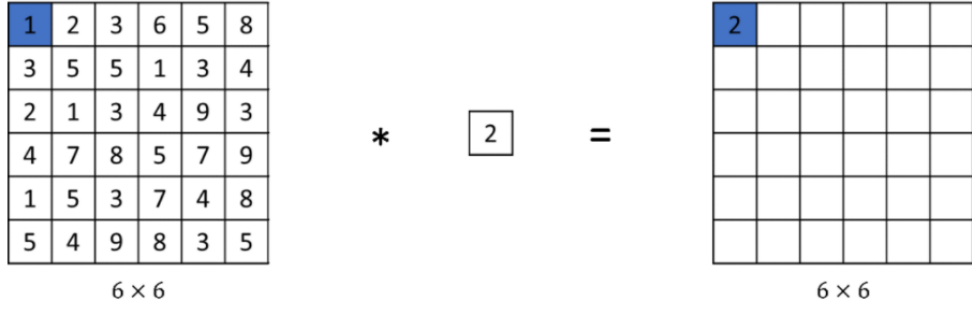
Şekil 2. 59 ResNet Ağ Yapısında Katman Sayısı ile Eğitim Hatası İlişkisi [34]

Böylece ResNet ağlarında 100 katman sayısına kadar çıkılarak model geliştirme çalışmaları yapılabilmektedir [30].

2.11.5 Ağ İçinde Ağ (Network in Network)

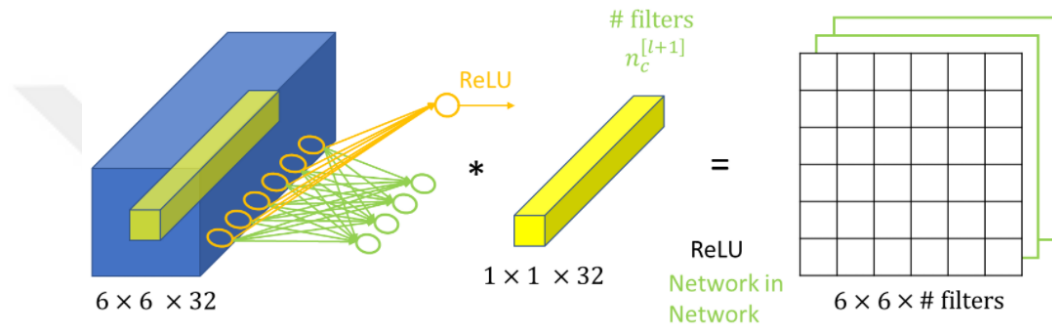
1x1 Evrişim

1x1 evrişim işlemi aslında bir sayı ile giriş vektörünü çarpma işlemidir.



Şekil 2. 60 1x1 Evrişim İşlemi [34]

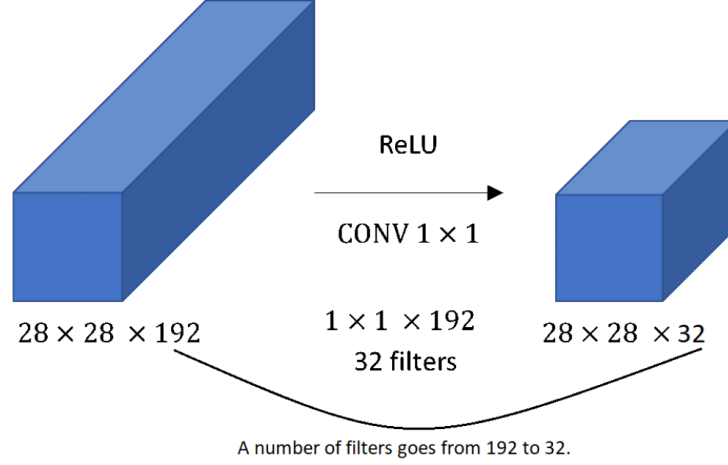
Eğer giriş resmimizin boyutu 6x6 yerine 6x6x32 olsaydı,



Şekil 2. 61 6x6x32 Boyutlu Resim İle 1x1 Filtrenin Evrişim İşlemi [34]

Bu durumda filtremiz de giriş resmimiz ile uyumlu olması için 1x1x32 boyutlu olmalıdır. Filtremizdeki tek bir 1x1x32'lik dilim resimde karşılık gelen 1x1x32'lik kısım ile karşılıklı olarak tek bir nöron olarak ifade edilir. Bu durum sanki tam bağlı bir ağ gibi görünmektedir. Çıktımız 6x6x Filtre Sayısı şeklinde olmaktadır. Bu yapı 1x1 evrişim işlemi yada ağ içinde ağ (network in network) şeklinde isimlendirilir [44].

Şimdi bu işlemin yüksek kanal sayılı giriş resimlerinde nasıl faydalı olduğuna bakalım,



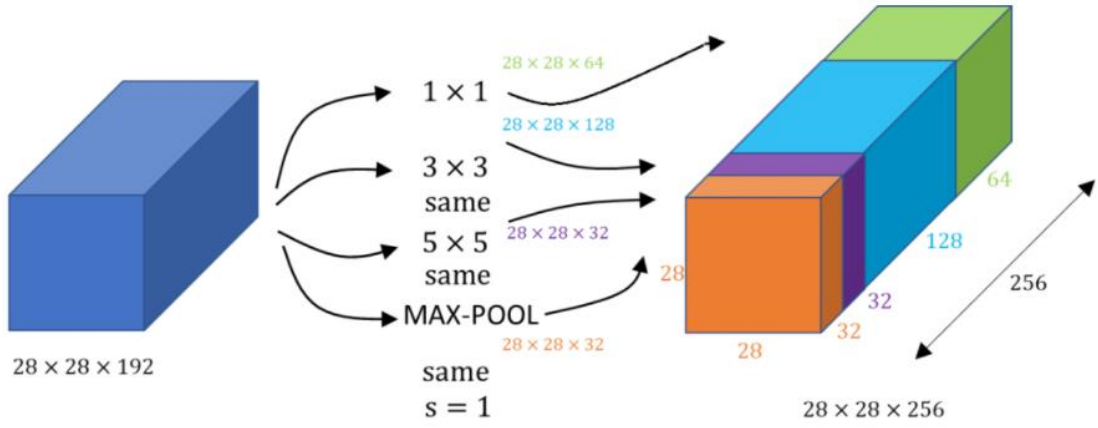
Şekil 2. 62 Çok Kanallı 3 Boyutlu Resim İle 1x1 Filtrenin Evrişim İşlemi [34]

Şekil 2.62 de görüldüğü üzere 192 gibi çok sayılabilecek bir kanal sayısına sahip $28 \times 28 \times 192$ boyutlu bir resim ile $1 \times 1 \times 192$ boyutlu 32 adet filtreyi evrişim işlemine sokarsak çıkışta $28 \times 28 \times 32$ boyutlu bir resim elde ederiz. Böylece tek bir hamlede herhangi bir havuzlama katmanı kullanmadan çıkış resminin boyutlarını küçülterek hesaplanacak parametre sayısını da azaltabiliyoruz. Filtre sayısını ayarlayarak çıkış boyutunu istediğimiz gibi değiştirebiliriz [30].

2.11.6 Inception Network

Szegedy ve ark. [45] tarafından geliştirilen bu model yapısı ağ içinde ağ olarak isimlendirdiğimiz modeli farklı filtre boyutları için kullanarak bir ağ yapısı oluşturur. Bu modelde giriş resmimize sadece 1×1 boyutunda bir filtre uygulamak yerine 3×3 , 5×5 gibi farklı boyutlarda filtreleri 1×1 evrişim işlemine benzer şekilde kullanıp her bir filtrenin sonuçlarını çıkışta bir araya getiririz. Böylece birden fazla filtreye ait çeşitli özelliklere ait olan bir çıkış verisi elde edilir.

Şimdi bu işlemi örnek bir giriş resmi üzerinde gösterelim,



Şekil 2. 63 Inception Ağ Yapısı [34]

Şekil 2.63 de görüldüğü üzere $28 \times 28 \times 192$ boyutlu bir giriş resmine ilk önce 1×1 boyutlu 64 adet filtreyle evrişim işlemi uyguluyoruz. Sonuçta $28 \times 28 \times 64$ boyutunda bir çıktımız oluyor.

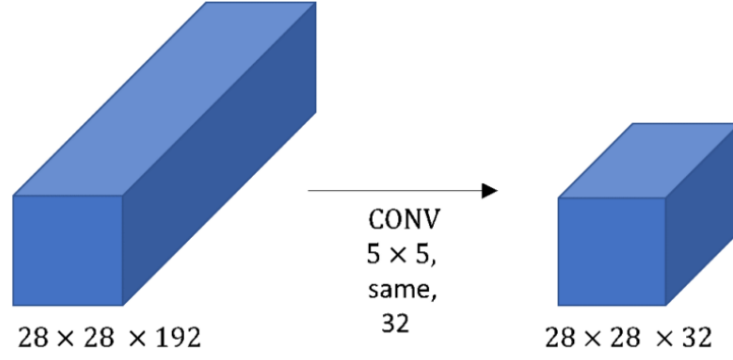
Daha sonra $3 \times 3 \times 192$ boyutlu 128 adet filtre ile dolgulaması aynı olacak şekilde yeniden evrişim işlemine tabi tutuluyor ve bu sefer $28 \times 28 \times 128$ boyutunda bir çıktımız oluyor.

Üçüncü olarak $5 \times 5 \times 192$ boyutlu aynı dolgulamalı olacak şekilde 32 adet filtre ile evrişim işlemine tabi tutuluyor ve $28 \times 28 \times 32$ boyutlu bir çıktı elde ediliyor.

Son olarak 32 kere aynı dolgulamalı ve 1 kaydırma adımlı olacak şekilde maksimum havuzlama işlemi uygulanıyor ve $28 \times 28 \times 32$ boyutlu bir çıktı elde ediliyor.

Böylece tek bir filtre boyutu seçip ona göre boyut ayarlama işlemleri yapmak yerine birden fazla filtreyi girişe benzer şekilde uygulayıp çıktıları bir araya getirip bağlayabiliriz. Çıktıları bir araya getirdiğimizde $28 \times 28 \times 256$ boyutlu bir resim elde edilir. Fakat burada birden fazla boyutta filtre kullanımının sonucu olarak parametre hesaplama maliyeti karşımıza çıkmaktadır.

Örneğin sadece 5×5 boyutlu filtre kullandığımız kısım için bir hesaplama yapacak olursak,



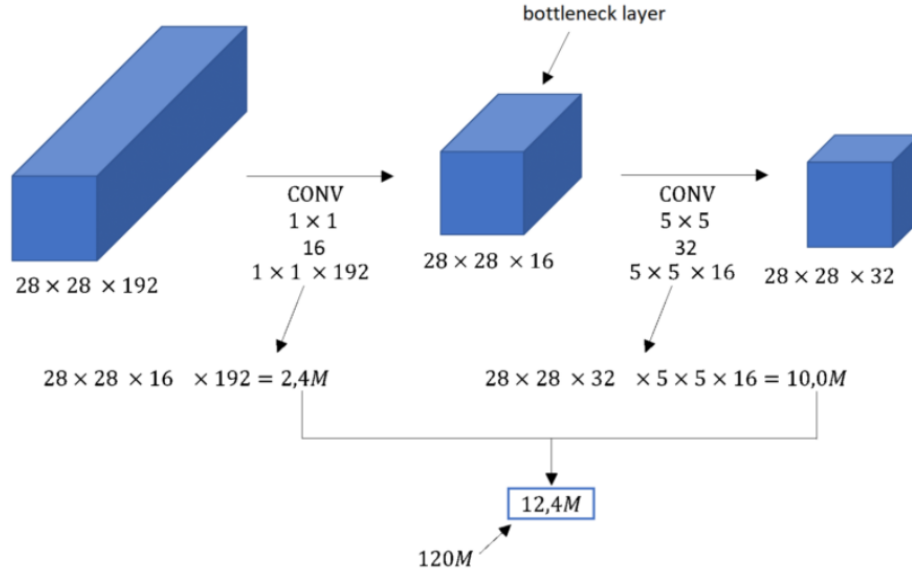
Şekil 2. 64 Inception Ağı İçin Parametre Sayısı Hesaplama [34]

Şekil 2.64 de görüleceği üzere $5 \times 5 \times 192$ boyutlu 32 adet filtreyi $28 \times 28 \times 192$ boyutlu giriş resmi ile evrişim işlemine soktuğumuzda $28 \times 28 \times 32$ boyutlu bir çıktı elde edilmektedir.

$$\text{Parametre Sayısı} = 5 * 5 * 192 * 28 * 28 * 32 \cong 120 \text{ Milyon}$$

Şeklinde olmaktadır. Bu her ne kadar günümüz bilgisayarları için hesaplanabilir bir miktar olsa da yine de oldukça fazladır.

Bu durumun üstesinden gelmek için yukarıda bahsettiğimiz 1×1 'lik evrişim işlemi tüm bu evrişimlerin önüne getirerek etkili şekilde faydalanabiliriz.

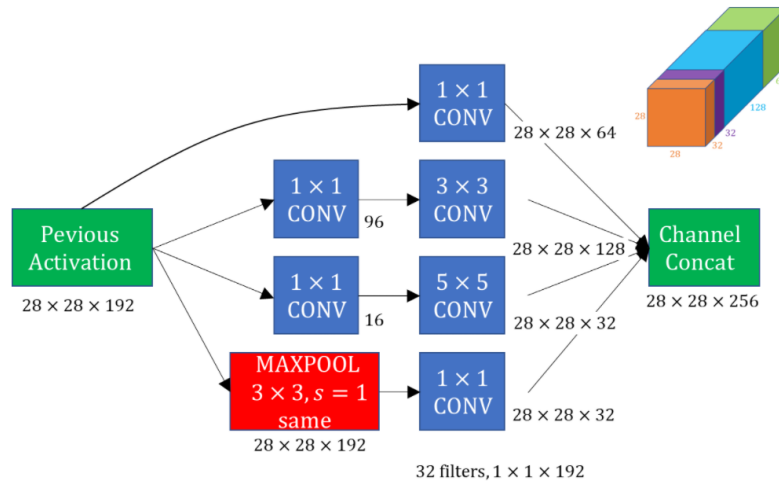


Şekil 2. 65 Inception Ağı İçin Parametre Sayısı Hesaplama 2 [34]

Bu sefer $1 \times 1 \times 192$ boyutlu 16 filtreyi $28 \times 28 \times 192$ boyutlu aynı giriş resmiyle evrişim işlemine soktuktan sonra diğerin farklı olarak ikinci evrişim işlemi birinci işlemin sonucuna uyguluyoruz. İlk evrişim işleminden elde ettiğimiz $28 \times 28 \times 16$ boyutlu çıktıya $5 \times 5 \times 16$ boyutlu 32 adet filtreyi evrişim işlemine tabi tutarsak şekil 2.65'deki gibi $28 \times 28 \times 256$ boyutlu bir çıktı elde edilir. Ve boyutun ciddi anlamda düştüğü ilk katmana yani $1 \times 1 \times 192$ boyutunda filtre kullanarak elde edilen ilk kısma dar boğaz (bottleneck) adı verilir.

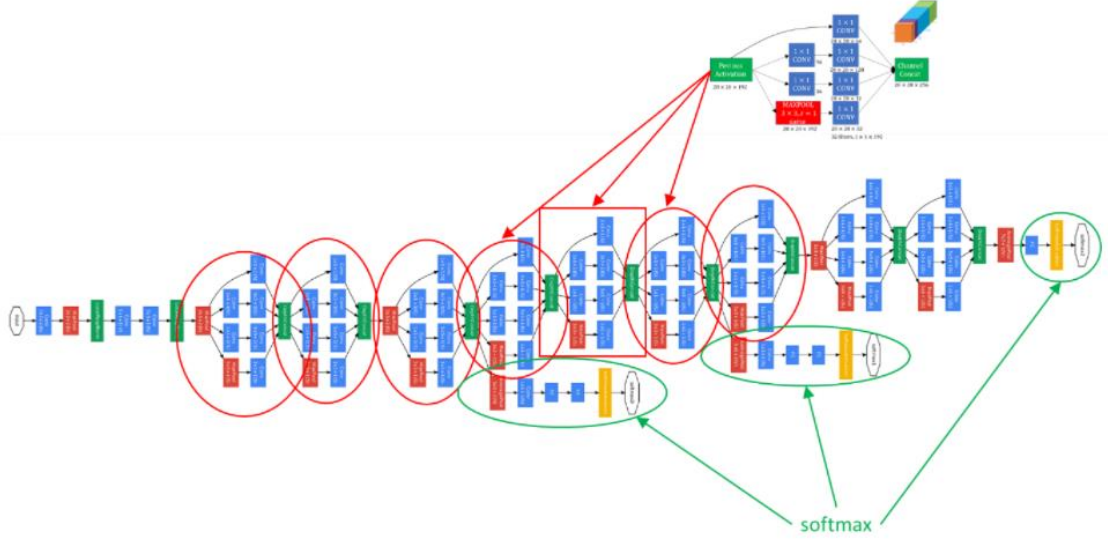
Yukarıda yaptığımız şekilde bu ağ içinde bir parametre hesabı yapıldığında yaklaşık 12,4 milyon parametre sayısı elde edilir. Bu değer diğer parametre sayısından yaklaşık 10 kat daha azdır ve bu hesaplama maliyetini ciddi oranda düşürür.

Tüm bunları gösterdikten sonra şimdi tek bir inception ağ modülü yapısını aşağıdaki gibi gösterebiliriz,



Şekil 2. 66 Inception Ağı Başlangıç Modülü [34]

Tam bir inception ağ yapısı şekil 2.66 da gösterilen birçok ağ modülünün bir araya gelmesi ile oluşur.



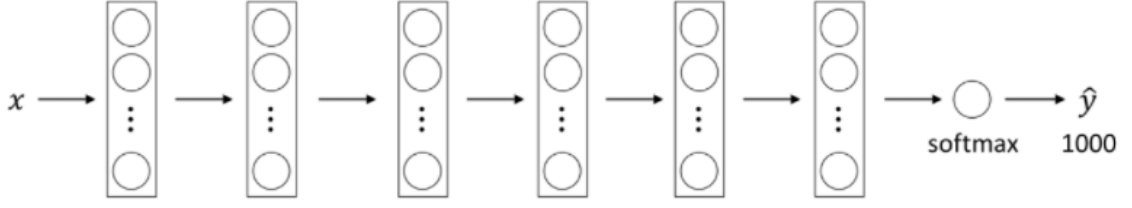
Şekil 2. 67 Inception Ağı Yapısı [34]

Şekil 2.66'daki başlangıç ağ modüllerinin bir araya gelerek oluşturdu yapılar şekil 2.67 de gösterilmiştir. Bu ağ yapısında ilave yan dallar görülmektedir. Bunlar tam bağlı katmanlardan oluşur ve çıkışta sınıflandırma yapmak için softmax fonksiyonu kullanır. Bunun amacı istenilen özneliklerin ara katmanlarda hesaplanıyor olduğundan emin olmak içindir.

2.11.7 Transfer Öğrenme

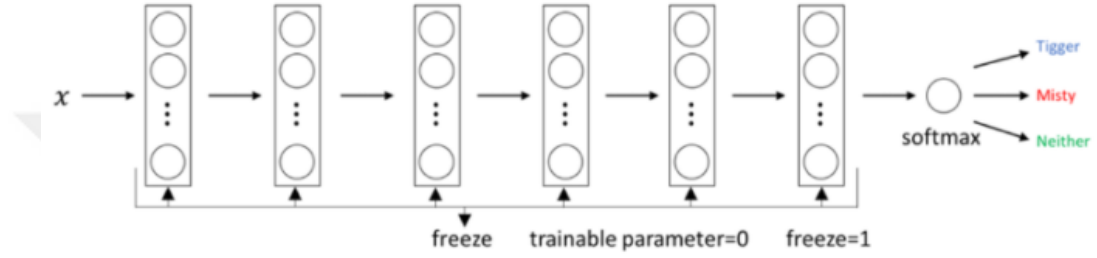
Kurduğunuz bir derin öğrenme modelini yeni bir eğitim seti üzerinde eğitmek için güçlü ekran kartlarına (GPU) ve haftalar hatta aylar süren zamana ihtiyaç duyarsınız. Bunun yerine başkaları tarafında bu süreçleri başarıyla tamamlamış ve iyi sonuç vermiş ağların öğrenilmiş parametrelerini (ağırlıkları) indirerek kendi modelinize transfer edebilirsiniz. İşte bu öğrenme şekline transfer öğrenme denilmektedir. [46]

İki farklı kediyi ayırmak için bir sinir ağı modeli geliştirdiğimizi düşünelim. Bu kedilerin adı Misty ve Tigger olsun. Üçüncü sınıfı da bunlardan hiçbiri şeklinde seçelim. Bunun için elimizde kısıtlı bir eğitim verisi olduğunu da düşünürsek sıfırdan bir eğitim yapmak yerine daha çok veri ile eğitilmiş ve başarı olmuş bir modelin ağırlık parametrelerini kullanmak daha doğru olacaktır.



Şekil 2. 68 1000 Sınıflı Sinir Ağı [34]

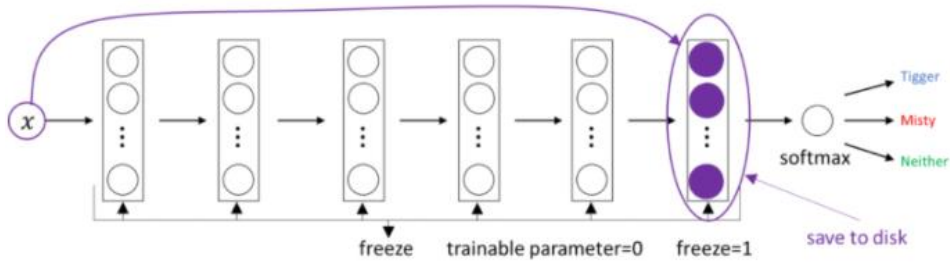
Örneğin şekil 2.68'deki gibi 1000 sınıftan oluşan bir veri seti üzerinde eğitilmiş olan bir sinir ağının ağırlıklarını kullanabiliriz.



Şekil 2. 69 Transfer Learning [34]

Bu durumda yapmamız gereken ilk şey şekil 2.69'daki gibi referans aldığımız ağın 1000 sınıflık softmax çıktısını kaldırarak kendi 3 sınıflı softmax fonksiyonumuzu sisteme yerleştirmeliyiz.

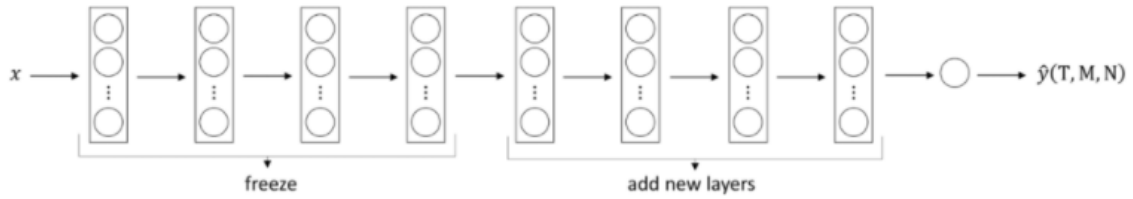
Daha sonra referans aldığımız sinir ağı modelinin katmanlarını dondurarak yeniden eğitime tabi tutmuyoruz. Fakat bu modelin sonuna kendi eklediğimiz yeni katmanları ekleyerek eğitilmiş olan eski ağırlıkları kendi sistemimiz için yeniden güncelleyebiliriz.



Şekil 2. 70 Transfer Learning Küçük Veri Seti [34]

Şekil 2.70'deki gibi son kısma kendimize ait bir katman ekleyebilir referans aldığımız modelin ağırlıklarını kendi veri setimiz ile güncelleyerek yeni ağırlıkları diske kaydedebiliriz.

Eğer veri setimizde yeterince örnek var ise bu sefer dondurulması gereken katmanları daha az seçerek işlem yapabiliriz,



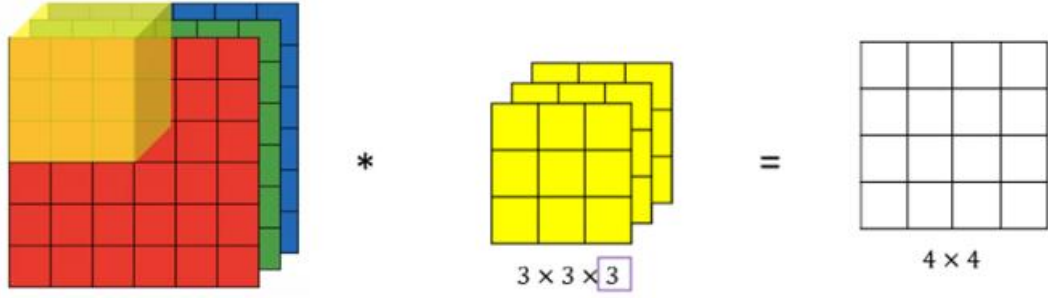
Şekil 2. 71 Transfer Learning Büyük Veri Seti [34]

Şekil 2.71'deki gibi referans olarak aldığımız veri setinin daha az katmanını dondurup diğer katmanlar için ve sonradan eklediğimiz katmanlar ile birlikte kurduğumuz yeni sistemi kendi veri setimiz için yeniden eğiterek ağırlık parametrelerimizi güncelleyebiliriz. Bu modelde de softmax fonksiyonunu kendi sistemimizin çıktılarına göre ayarlamamız gerekmektedir [30].

2.11.8 MobileNet

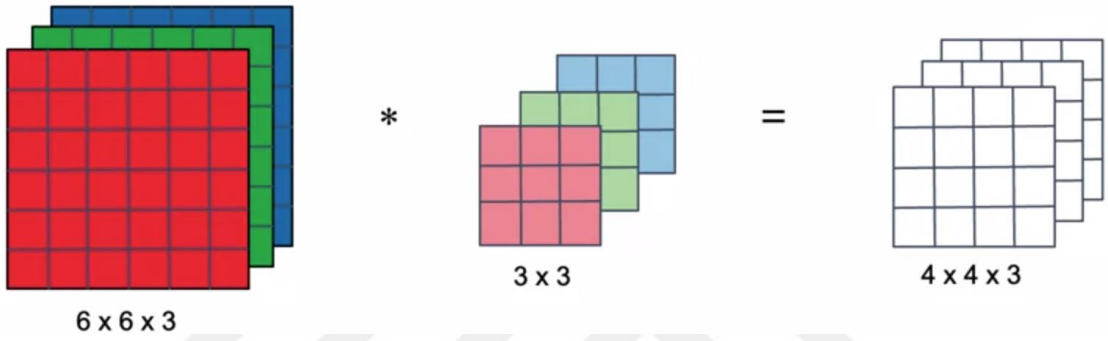
Bilindiği üzere sinir ağlarını eğitmek için çok güçlü CPU ve GPU donanımları gerektirmektedir. Bu donanımlar üzerinde aylarca eğitilen modellerin son kullanıcı için erişime açılması da oldukça fazla bellek ve bilgisayar kaynağına ihtiyaç duymaktadır. Eğer oluşturduğunuz modelin son kullanıcıların mobil cihazlarında ve daha az güçlü CPU ve GPU içeren bilgisayarlarında da çalışmasını istiyorsak o zaman tercih edeceğimiz model Howard ve ark. [47] tarafından geliştirilen MobileNet olmalıdır.

Bu model Derinlemesine Ayrılabilir Evrişim (Depthwise Separable Convolutional) ve Noktasal (Pointwise) işlem olmak üzere iki kısımdan oluşmaktadır. Bu iki işlemi yapmadan önce şekil 2.72'deki normal evrişim işlemini tekrar gösterelim,



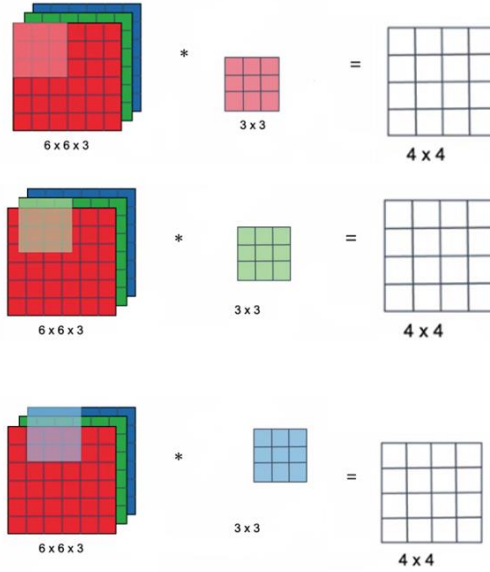
Şekil 2. 72 Renkli Resmin Evrişim Hesaplama Örneği [34]

Derinlemesine ayrılabilir evrişim işlemi ise şu şekilde gösterilir,



Şekil 2. 73 Derinlemesine Ayrılabilir Evrişim [34]

Burada şekil 2.72'den farklı olarak 3x3x3 boyutlu bir filtre yerine 3x3x1 boyutlu 3 filtre kullanmaktayız. Ayrıca kullandığımız her bir filtre resimde karşılık gelen kanal ile evrişim işlemine tabi olmaktadır.



Şekil 2. 74 Derinlemesine Ayrılabilir Evrişim 2 [34]

Şekil 2.73’de bahsedilen bu işlem görülmektedir. Bu işlemin sonucu şekil 2.74’de görüldüğü gibi 4x4x3 şeklinde olmaktadır.

Şimdi bu yapı için karşımıza çıkan hesaplama maliyetine yani parametre sayısına bakalım,

$$Parametre Sayısı = filtre Boyutu \times Filtre Pozisyonları \times Filtre Sayısı \quad (2.57)$$

$$Parametre Sayısı = 3 \times 3 \times 4 \times 4 \times 3 = 432$$

Görüldüğü üzere hesaplanması gereken parametre sayımız 432 olmaktadır.

Şimdi de MobileNet’i oluşturan ikinci kısım olan Noktasal (Pointwise) evrişimi inceleyelim.



Şekil 2. 75 Noktasal (Pointwise) Evrişim [34]

Noktasal evrişim işleminde Şekil 2.72’den görüleceği üzere ilk işlem olan derinlemesine ayrılabilir evrişim çıktısı olan 4x4x3 boyutlu resme 1x1x3 boyutlu 5 adet filtre ile evrişim işlemi uygulanır.



Şekil 2. 76 Noktasal (Pointwise) Evrişim 2 [34]

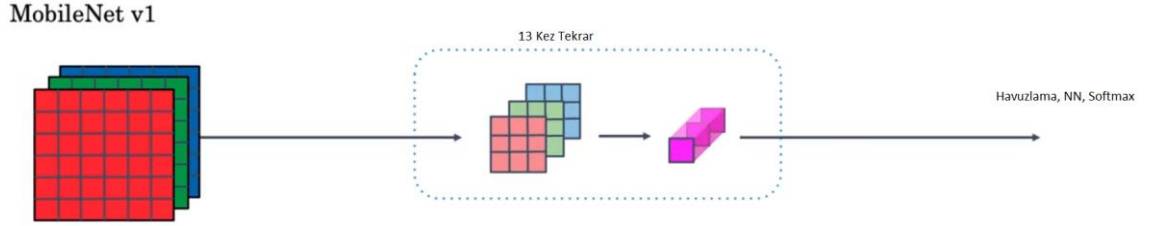
Bu işlem daha önce inceldiğimiz 1x1 evrişim işlemine benzer şekilde yapılır ve sonuçta şekil 2.75’deki gibi 4x4x5 boyutunda bir çıktı elde edilir. Bu adımında parametre sayısını 2.57 denkleminde hesaplırsak,

$$Parametre Sayısı = 1 \times 1 \times 3 \times 4 \times 4 \times 5 = 240$$

Böylece noktasal evrişim adımından elde ettiğimiz parametre sayısı da 240 olur. Bu durumda tüm modelin hesaplanması gereken parametre sayısı $432+240 = 672$

Olmaktadır. Normal evrişim işleminde hesapladığımız parametre sayısının 2160 olduğu düşünülduğünde yaklaşık yüzde 31 oranında daha az parametre hesaplamış olmaktadır.

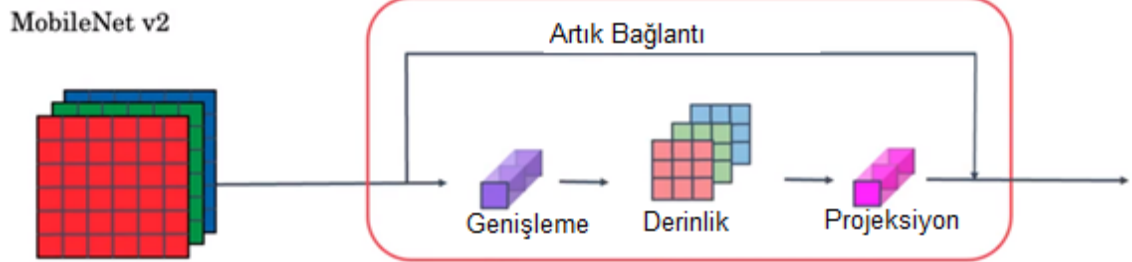
Bu iki adımı anlattıktan sonra MobileNet'in ilk versiyonunu şu şekilde gösterebiliriz,



Şekil 2. 77 MobileNet 1. Versiyonu [34]

Şekil 2.74 de görülen derinlemesine ayrılabilir evrişim ve noktasal işlem 13 kez tekrarlandıktan sonra bir havuzlama sonrasında tam bağlı bir sinir ağı ve en son softmax fonksiyonuna bağlanır. Böylece bu yapı ile daha az parametre hesabı yapılarak daha verimli bellek kullanımı ve hızlı çalışan bir ağ yapısı elde edilir.

Daha sonra 2019 yılında Sandler ve ark. [48] tarafından yapılan çalışmalar ile bu ağ yapısı daha da verimli hale getirilerek ikinci versiyon olarak tanıtılmıştır.



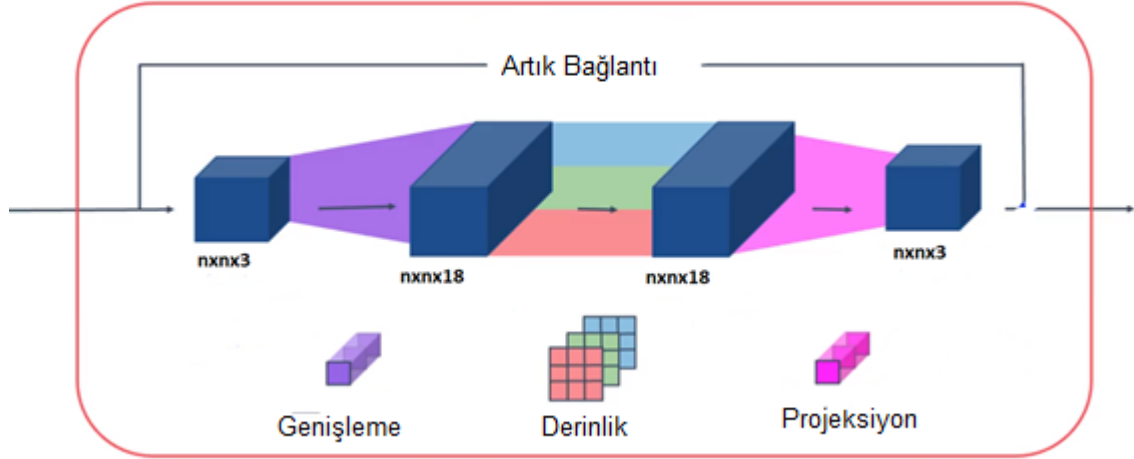
Şekil 2. 78 MobileNet 2. Versiyonu [34]

Şekil 2.78 de görülen MobileNet'in ikinci versiyonunda yapılan birinci değişiklik residual [49] olarak adlandırılan ve daha önce bahsettiğimiz artık çıktıyı girişten çıkış kısmına aktarmaktır. İkinci değişiklik ise derinlemesine ayrılabilir evrişimden önce gelen ve genişleme (expansion) olarak adlandırılan kısımdır. Bu modelde son

kısımda havuzlama, tam bağılı sinir ağı ve softmax fonksiyonu kullanır. Ancak bu kısımlar birinci versiyondaki gibi 13 kere değil 17 kere tekrar eder.

Şimdi ikinci versiyondaki değişen blok yapısına detaylı bakalım,

MobileNet v2 Bottleneck

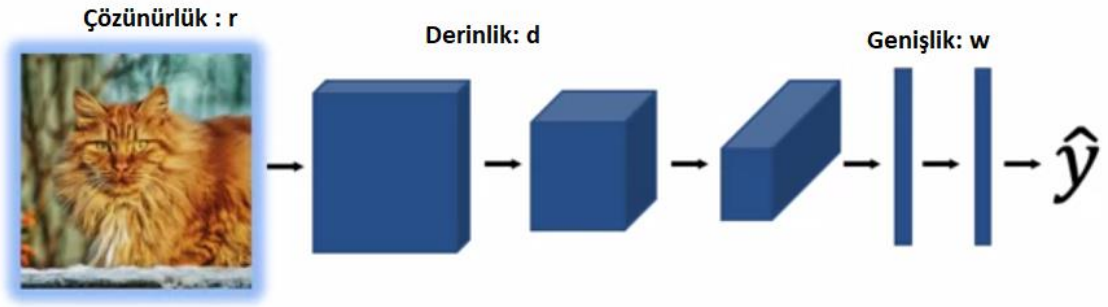


Şekil 2. 79 MobileNet 2. Versiyon Blok Yapısı [34]

Şekil 2.79'daki ikinci versiyona ait blokta görüleceği üzere giriş kısmı direkt olarak kalıntı ağılardaki gibi çıkışa aktarılıyor. Daha sonra $n \times n \times 3$ boyutlu giriş resmi 18 adet $1 \times 1 \times 3$ boyutlu filtre ile genişletiliyor. Bu genişleme sayesinde giriş resmine ait daha fazla öznelik öğrenmesi gerçekleşiyor. Fakat bu kısımdaki bellek kullanımı ve parametre sayısı artıyor. Bunun üstesinden gelmek için derinlemesine ayrılabilir evrişim işleminden sonra 3 adet $1 \times 1 \times 3$ boyutlu filtre ile noktasal evrişim işlemi yapılarak boyutun tekrar küçültülmesi sağlanıyor. Bu sayede bir bloğun fazlaca bellek kullanmadan diğer bloğa hızlı bir şekilde bağlanması sağlanmış oluyor. Son yapılan işlem aynı zamanda daha öncede gördüğümüz şekliyle darboğaz (bottleneck) olarak adlandırılmaktadır [30].

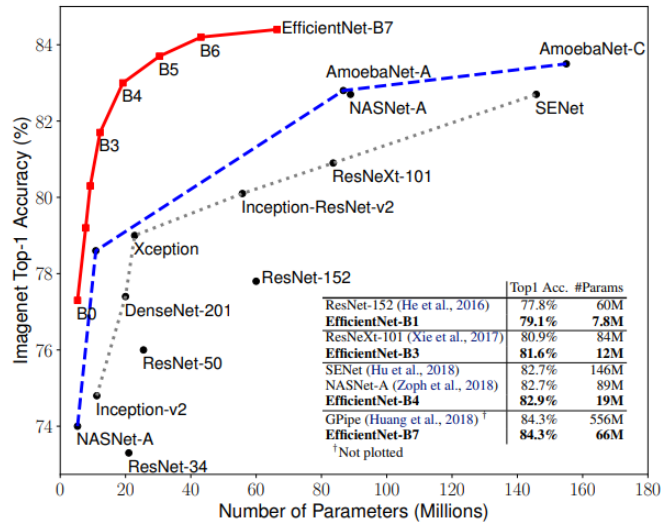
2.11.9 EfficientNet

2019 yılında Tan ve Le [50] tarafından geliştirilen EfficientNet, MobileNet için geliştirilen bir mimarinin özelleştirilmiş bir mobil cihazda kullanabilmeniz için gerekli olan derinlik d , genişlik w ve çözünürlük r arasındaki ilişkiyi verir.



Şekil 2. 80 EfficientNet: Parametre Ayarlama [34]

Bu şekilde kendiniz için en uygun giriş çözünürlüğünü, ağ derinliğini ve genişliğini ayarlayabilirsiniz.



Şekil 2. 81 EfficientNet Performans Karşılaştırması [50]

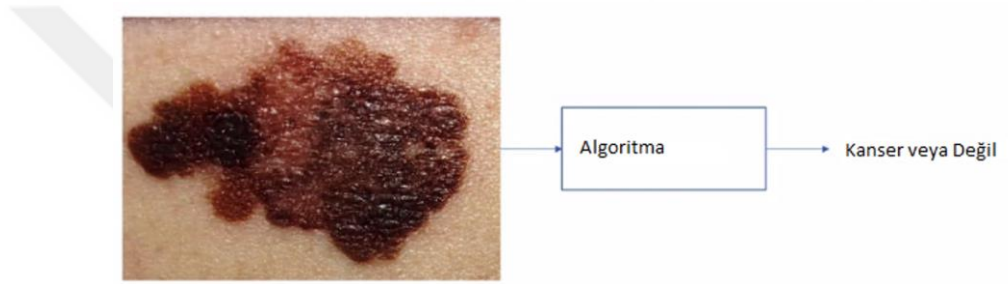
Şekil 2.81'den de görüleceği üzere bu ayarlama yapıldığında eğitilecek daha az parametre ile daha iyi doğruluk oranları elde edilmiştir [30].

3

DERİN ÖĞRENMENİN RÖNTGEN GÖRÜNTÜLERİNE UYGULANMASI

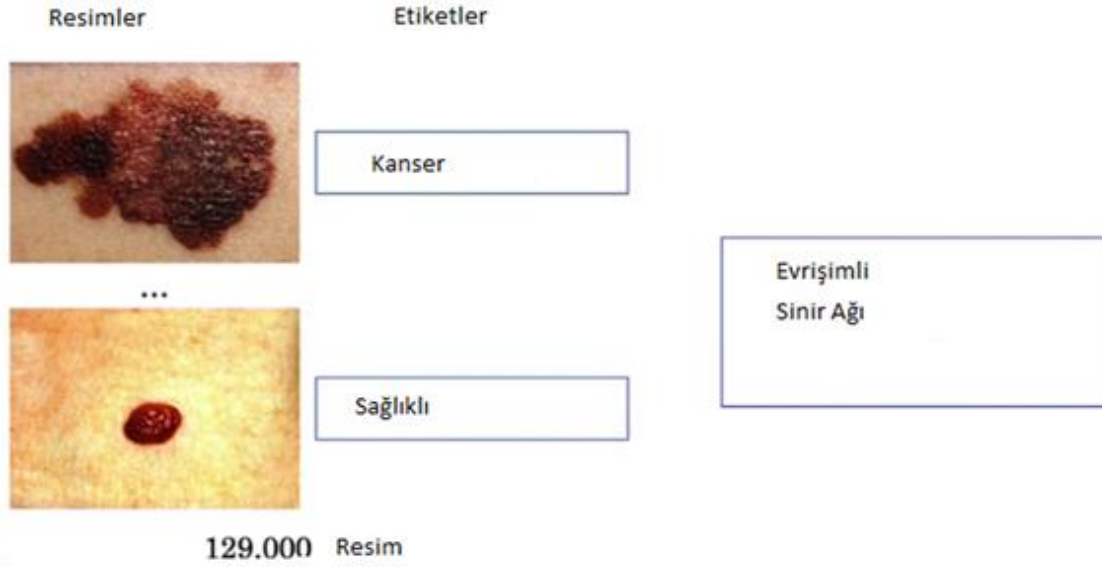
3.1 Medikal Resimlerde Hastalık Tespiti

Yapay öğrenme geliştikçe kullanım alanları da artmaya başlamıştır. Özellikle Tıp alanında kullanımı da giderek yaygınlaşmaya başlamıştır. Göğüs röntgeni için kullanımına geçmeden önce diğer başka Tıp alanlarındaki yayın kullanımına göz atalım.



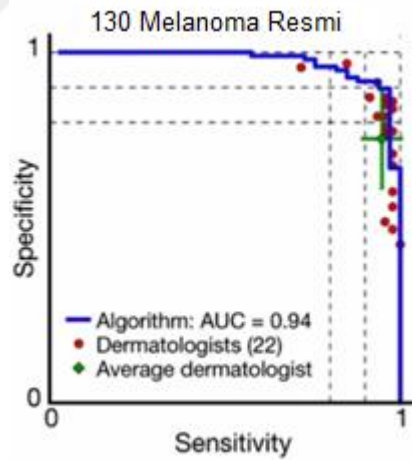
Şekil 3. 1 Deri Tümörü İçin Yapay Öğrenme [51]

İlk örneğimiz Cildiye (Dermatoloji) ile ilgili. Örneğin şekil 3.1 de insan cildi üzerinde oluşan bir ben görünmektedir. Örneğin bunun için sağlıklı ve tümörlü verileri içeren bir eğitim kümesi oluşturularak hastalığın tespitini yapacak evrişimli bir derin sinir ağı modeli geliştirilebilir.



Şekil 3. 2 Deri Tümörü İçin Yapay Öğrenme Modeli [51]

Cilt tümörleri konusunda yapılan çalışmalarda geliştirilen bu algoritmalar ile hekimlerin buldukları sonuçlar karşılaştırıldığında,

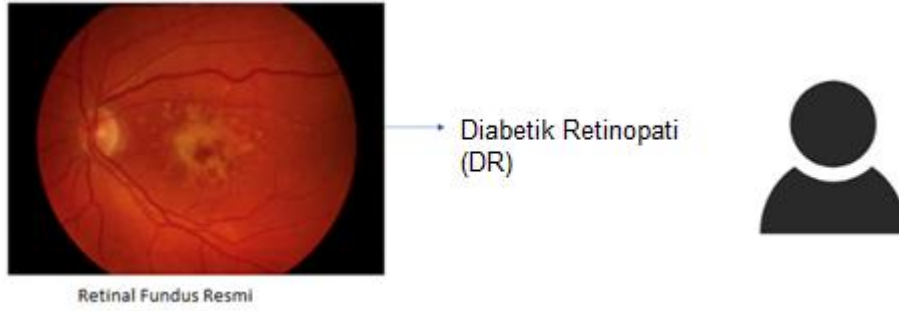


Şekil 3. 3 Deri Tümörü (Melanoma) Tahmin Karşılaştırması

Şekil 3.3'ten de görüleceği üzere Melanoma olarak adlandırılan kötü huylu tümörün hekimler ve algoritma yaptıkları tahminler birbirine oldukça yakındır [52].

İkinci örneğimizde ise göz hastalıkları ile ilgili. Diyabetin sebep olduğu ve retinaya zarar veren diyabetik retinopati hastalığının teşhisi için gözün arka tarafını

fotoğraflayan ve retinal fundus olarak isimlendirilen bölüme ait fotoğraflar kullanılarak 2016 yılında bir başarılı bir çalışma yapıldı [53].



Şekil 3. 4 Retinal Fundus Hastalığı [51]

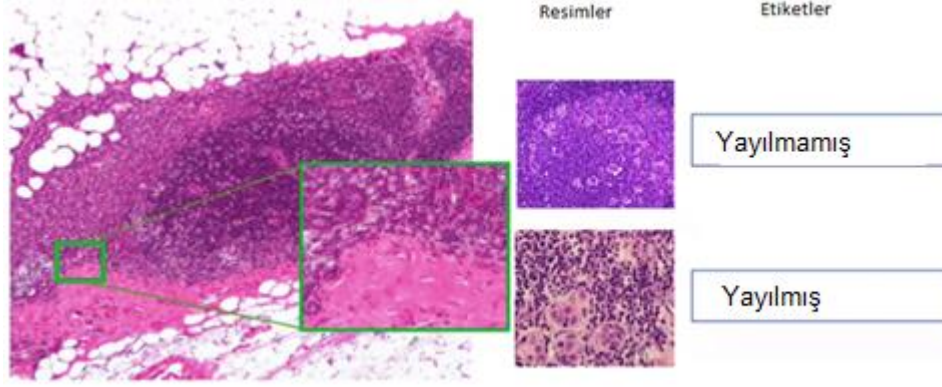
Bu hastalığı belirleyebilmek için eğitilen modelde %30 hasta olan 128 binden fazla resim kullanılmıştır.



Şekil 3. 5 Retinal Fundus Hastalık Tespiti [51]

Elde edilen sonuçlar en az uzmanların yaptığı tahminler kadar iyi bulunmuştur.

Üçüncü örneğimiz ise Doku Hastalıkları (Histopatoloji) ile ilgilidir. Burada uzmanlar mikroskop altındaki dokuları inceleyerek burada yer alan zararlı olabilecek hücrelerin ayırt etmeye çalışırlar [51].



Şekil 3. 6 Doku Hastalık Tespiti [54]

Bu alanda da 2018 yılında yapılan yapay öğrenme çalışmaları ile elde edilen sonuçların uzmanların elde ettikleriyle karşılaştırılabilir seviyede olduğu gözlemlenmiştir. [55]

3.2 Göğüs Röntgenlerinde Hastalık Tespiti

Göğüs röntgeni 1 yılda çekilen yaklaşık 2 milyar örnek resim ile tıpta en yaygın olarak kullanılan tanısal görüntüleme tekniklerinin başında gelmektedir.



Şekil 3. 7 Göğüs Röntgeni [51]

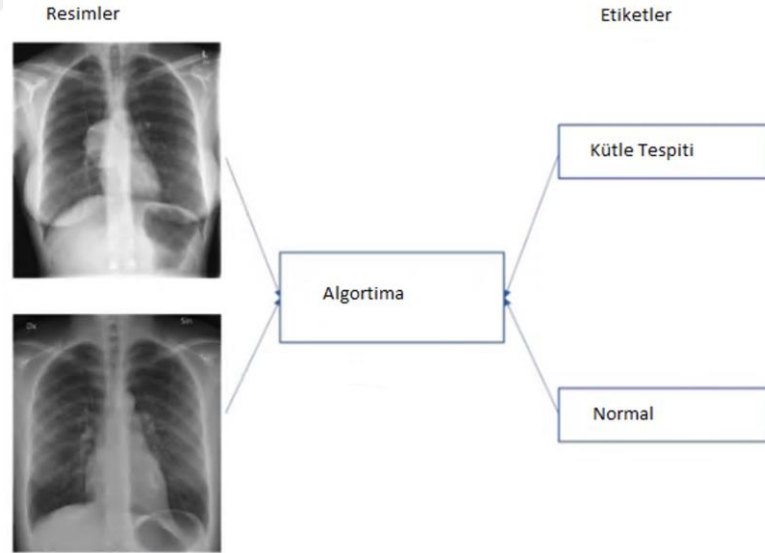
Şekil 3.7 de görülen göğüs röntgeni ile uzmanlar başta zatürre ve akciğer kanseri olmak üzere birçok yaygın akciğer ve göğüs hastalıklarını tespit edebilirler. Bunlar yaygın olarak tespit edilen ve kütle olarak isimlendirilen hastalık akciğerde

meydana gelen 3cm'den büyük lezyon veya doku hasarlarının oluşmasıyla meydana gelmektedir.



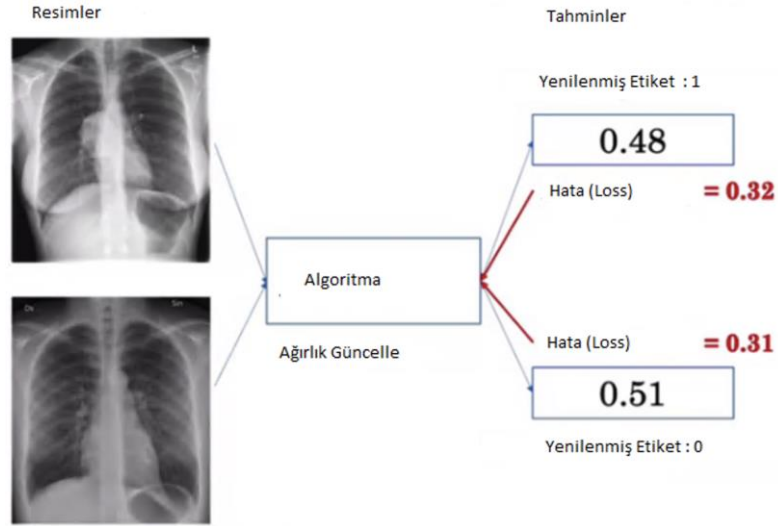
Şekil 3. 8 Kitle Bulunan-Normal Göğüs Röntgenleri [51]

Örneğin şekil 3.8'deki gibi kütle içeren bir göğüs röntgeni ile normal göğüs röntgenlerini alarak bir model oluşturabilir ve modele kütle içeren göğüs röntgenlerini öğretebiliriz.



Şekil 3. 9 Göğüs Hastalıkları için Model Eğitimi [51]

Bu işlemin sonunda modele gir olarak verdiğimiz bir röntgen görüntüsü için aşağıdaki gibi bir çıktı elde ederiz,



Şekil 3. 10 Göğüs Hastalıkları için Model Çıktısı [51]

Algoritma, görüntünün bir kütle içermeye olasılığı olan birden küçük sayılar biçiminde bir çıktı üretir. Belirleyeceğimiz eşik değerine göre ya da 1'e en yakın olan değere göre etiketleme yapabilir veya modelimizi tekrar eğiterek ağırlıklarımızı güncelleyebiliriz [51].

3.3 Sınıf Dengesizliği (Class Imbalance)

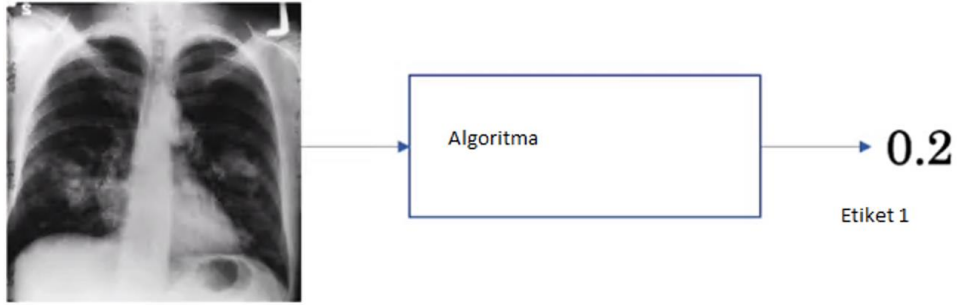
Tıbbi görüntüleme oluşturduğumuz modelimizin eğitimi için genelde 3 temel zorluk kaşımıza çıkmaktadır. Bunlar veri setimizde yer alan sınıfların dengesiz dağılımları, çoklu etiketleme görevleri ve veri setinin boyutudur.

Bunlardan sınıf dengesizliği olarak isimlendirdiğimiz birinci zorluk veri setinde bulunan sağlıklı ve hastalıklı görüntü sayılarının eşit olmamasından kaynaklanmaktadır. Genelde sağlıklı görüntü sayısı hastalıklı resimlerden oldukça fazla sayıda olmaktadır.

Bu durumu ortadan kaldırmak için neler yaptığımıza bir bakalım. Öncelikle algoritmamızın ürettiği çıktı için hesapladığımız Binary Cross-Entropy olarak adlandırılan kayıp fonksiyonunu tanımlayalım,

$$L(X, y) = \begin{cases} -\log P(Y = 1|X), & y = 1 \\ -\log P(Y = 0|X), & y = 0 \end{cases} \quad (3.1)$$

Kayıp fonksiyonumuzu bu şekilde tanımladıktan sonra modelimizin örnek olarak ürettiği iki çıktı için kayıp fonksiyonunu tanımlayalım,



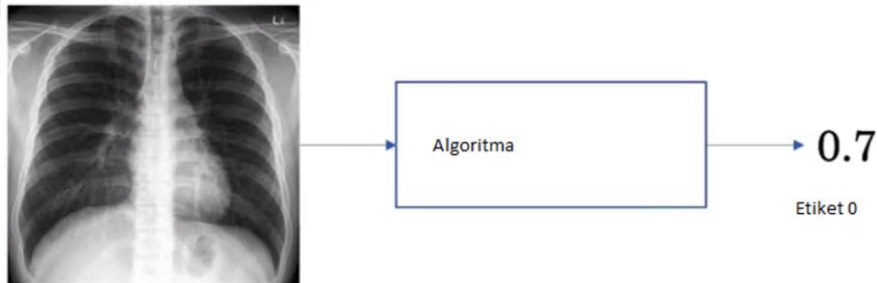
Şekil 3. 11 Kayıp Fonksiyonu İçin Model Çıktısı 1 [51]

Şekil 3.11'deki modelimize verdiğimiz resmin 0.2 olasılıkla bir numaralı etikete sınıflandırıldığını düşünelim. Bu durumda kayıp fonksiyonu,

$$y = 1 \text{ olduğu için, } L = -\log P(Y = 1|X)$$

$$L = -\log 0.2 = 0.7$$

Görüleceği üzere ilk çıktımızın kayıp değeri 0,7 olarak hesaplanır. İkinci çıktıya bakarsak,



Şekil 3. 12 Kayıp Fonksiyonu İçin Model Çıktısı 2 [51]

İkinci resmimizin model çıktısı ise 0.7 olasılıkla sıfır numaralı etikete atanıştır. Bu durumda kayıp fonksiyonu,

$$y = 0 \text{ olduğu için, } L = -\log P(Y = 0|X)$$

$$L = -\log(1 - 0.7)$$

$$L = -\log 0.3 = 0.52$$

İkinci girdi olarak verilen resmimizin kayıp fonksiyonu da 0,52 olarak hesaplanır. Buradan hareketle sınıf dağılımı dengesiz olarak verilen bir veri için ne yapılabileceğine bakalım,

Tablo 3. 1 Dengesiz Sınıf Dağılımı Örneği [51]

Örnekler	Yeniden Örnekleme
P1 Normal	P3 Normal
P2 Normal	P6 Normal
P3 Normal	P1 Normal
P4 Kütle	P8 Normal
P5 Normal	P7 Kütle
P6 Normal	P4 Kütle
P7 Kütle	P7 Kütle
P8 Normal	P4 Kütle

Tablo 3.1'deki gibi 6 tane normal, 2 tane kütle içeren 8 adet örneğe sahip olduğumuzu düşünelim.

Tablo 3. 2 Dengesiz Sınıf Dağılımı Örneği 2 [51]

Örnekler	Tahmin Olasılıkları	Hata (Loss)
P1 Normal	0.5	0.3
P2 Normal	0.5	0.3
P3 Normal	0.5	0.3
P4 Kütle	0.5	0.3
P5 Normal	0.5	0.3
P6 Normal	0.5	0.3
P7 Kütle	0.5	0.3
P8 Normal	0.5	0.3

Bu resimlerin çıktı olasılıkları 0.5 ve kayıp değerleri 0.3 olmaktadır. Bu durumda bu veri setini şu şekilde dengeleyebiliriz,

$$\text{Kütle içeren toplam örneklerin toplam kaybı} = 2 * 0.3 = 0.6$$

$$\text{Normal örneklerin toplam kaybı} = 6 * 0.3 = 1.8$$

Şimdi bu dengesizliği çözmek için 3.1 denklemini yeniden düzenlememiz gerekmektedir,

$$L(X, y) = \begin{cases} w_p x - \log P(Y = 1|X), & y = 1 \\ w_n x - \log P(Y = 0|X), & y = 0 \end{cases} \quad (3.2)$$

Veri setini dengelemek için kayıp değerini uygun şekilde oluşturduğumuz ağırlıklar ile çarpılır. Kütle içeren örneklerin ağırlıklandırmasına w_p ve normallerinkine ise w_n dersek bu değerler,

$$w_p = \frac{\text{Toplam Kütle İçeren Örnek Sayısı}}{\text{Toplam Örnek}}, w_p = \frac{6}{8} \quad (3.3)$$

$$w_n = \frac{\text{Toplam Normal Örnek Sayısı}}{\text{Toplam Örnek}}, w_n = \frac{2}{8} \quad (3.4)$$

Şeklinde elde edilir.

Tablo 3.3 Dengesiz Sınıf Dağılımı Örneği 3 [51]

Örnekler	Tahminler	Hata (Loss)
P1 Normal	0.5	$2/8 \times 0.3 = 0.075$
P2 Normal	0.5	$2/8 \times 0.3 = 0.076$
P3 Normal	0.5	$2/8 \times 0.3 = 0.077$
P4 Kütle	0.5	$6/8 \times 0.3 = 0.225$
P5 Normal	0.5	$2/8 \times 0.3 = 0.075$
P6 Normal	0.5	$2/8 \times 0.3 = 0.076$
P7 Kütle	0.5	$6/8 \times 0.3 = 0.225$
P8 Normal	0.5	$2/8 \times 0.3 = 0.076$

Daha sonra bu değerleri ilgili çıktılarının kayıp değerleri ile çarpılır. Bu durumda,

$$\text{Kütle içeren toplam örneklerin toplam kaybı} = 2 * 0.225 = 0.45$$

$$\text{Normal örneklerin toplam kaybı} = 6 * 0.075 = 0.45$$

Bu şekilde yeniden ağırlıklandırma ile örneklerin sınıf içinde dengeli şekilde dağılması sağlanabilir.

Bir başka dengeleme şekli ise yeniden örnekleme (resampling) dir. Burada sınıf içerisindeki örnekleri uygun şekilde yeniden örnekleyerek dengeli hale getirebiliriz.

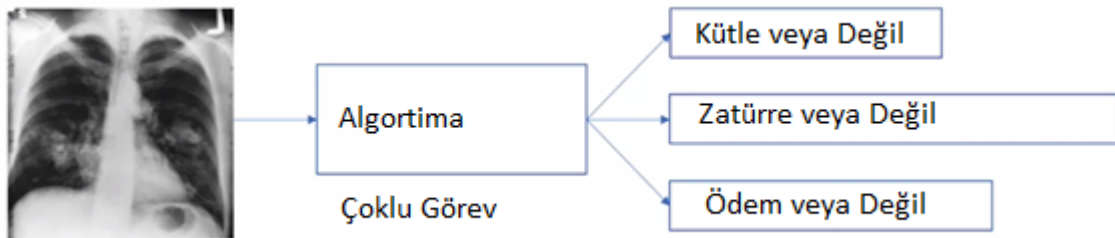
Tablo 3. 4 Dengesiz Sınıf Dağılımı Örneği 4 [51]

Örnekler		Yeniden Örnekleme
P1 Normal	Normal 4, Kütle 4 Örnek	P3 Normal
P2 Normal		P6 Normal
P3 Normal		P1 Normal
P4 Kütle		P8 Normal
P5 Normal		P7 Kütle
P6 Normal		P4 Kütle
P7 Kütle		P7 Kütle
P8 Normal		P4 Kütle

Bu durumu düzeltmek için öncelikle normal ve kütle içeren örneklerimizi kendi içinde gruplandırıyoruz. Daha sonra veri setimizin yarısını normal yarısını kütle içerecek şekilde yeniden düzenliyoruz. Bu düzenleme sırasında fazla örnek içeren resimlerden bazıları kullanılamayabilir ve az örnek içeren sınıftan birden fazla resim kopya olarak kullanılabilir. Bu durum istediğimiz veri setinin büyüklüğüne bağlıdır. Eğer veri setimizin büyüklüğünün 8 adet olmasını istiyorsak tablo 3.4 deki gibi bir düzenleme yapabiliriz. [51]

3.4 Çoklu Sınıflandırma (Multi- Task)

Daha önceki örneklerde hep kütle ve normal görüntüleri içeren ikili sınıflandırma yapmıştık. Gerçek hayatta durum bundan daha karmaşıktır özellikle uzmanlar göğüs hastalıklarında bir x-ışını röntgen görüntüsünden birden fazla hastalık tespiti yapabilmektedirler.



Şekil 3. 13 Çoklu Sınıflandırma

Bunu gerçekleştirebilmek için tüm bu hastalıkları tek seferde öğrenen bir model geliştirmeliyiz. Örneğin 3 adet hastalık sınıfına sahip olduğumuzu düşünelim.

Bunlar kütle, zatürre ve ödem olsun. Veri setimizdeki resimlerin bu hastalıklara ait olma (1 veya 0) değerleri ve ikili sınıflandırma olasılıklarına sahip olalım.

Tablo 3. 5 Çoklu Sınıflandırma 2 [51]

Örnekler (Kütle, Zatürre, Ödem)	Tahmin Olasılıkları	L (X,y)
P1 0, 1, 0	0.3, 0.1, 0.8	
P2 0, 0, 1	0.1, 0.1, 0.8	
P3 0, 1, 1	0.2, 0.2, 0.7	
P4 1, 0, 1	0.6, 0.3, 0.8	
P5 1, 1, 1	0.7, 0.7, 0.9	
P6 1, 0, 0	0.8, 0.1, 0.2	
P7 0, 1, 1	0.3, 0.9, 0.8	
P8 0, 0, 0	0.1, 0.1, 0.2	

Bu durumda ikili sınıflandırma için kullandığımız kayıp fonksiyonunu yeniden düzenlememiz gerekecektir.

Bunun için yapmamız gereken şey bir resme ait her bir hastalık için verilen kayıp değerlerini toplamaktır.

$$L(X, y) = L(X, y_{k\ddot{u}tle}) + L(X, y_{zat\ddot{u}rre}) + L(X, y_{\ddot{o}dem}) \quad (3.5)$$

Tablo 3. 6 Çoklu Sınıflandırma 3

Örnekler (Kütle,Zatürre,Ödem)	Tahmin Olasılıkları	Hata (Loss)
P1 0, 1, 0	0.3, 0.1, 0.8	0.15 + 1.00 + 0.70
P2 0, 0, 1	0.1, 0.1, 0.8	0.05 + 0.05 + 0.10
P3 0, 1, 1	0.2, 0.2, 0.7	0.10 + 0.70 + 0.15
P4 1, 0, 1	0.6, 0.3, 0.8	0.22 + 0.15 + 0.10
P5 1, 1, 1	0.7, 0.7, 0.9	0.15 + 0.15 + 0.05
P6 1, 0, 0	0.8, 0.1, 0.2	0.10 + 0.05 + 0.10
P7 0, 1, 1	0.3, 0.9, 0.8	0.15 + 0.05 + 0.10
P8 0, 0, 0	0.1, 0.1, 0.2	0.05 + 0.05 + 0.10

Buradan tekli hastalıklar için kayıp fonksiyonumuzu da örneğin sadece kütle için kayıp fonksiyonu şu şekilde yazılabilir,

$$L(X, y_{k\u00fctle}) = \begin{cases} w_{p,k\u00fctle} x - \log P(Y = 1|X), & y = 1 \\ w_{n,k\u00fctle} x - \log P(Y = 0|X), & y = 0 \end{cases} \quad (3.6)$$

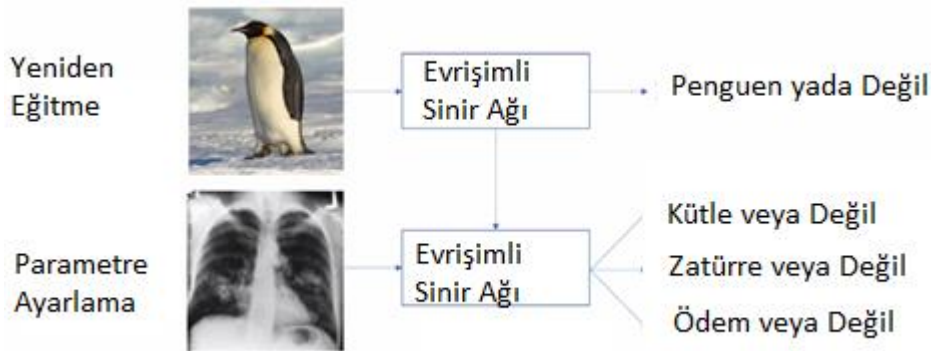
w_p Pozitif ve w_n negatif ağırlıkları gösterecek şekilde denklem 3.6 ile diğer hastalıklara ait kayıp değerleri de elde edilebilir [51].

3.5 Veri Seti Büyüklüğü

Röntgen görüntüleriyle çalışılırken karşılaşılan bir diğer zorlukta veri setinin boyutundan kaynaklanmaktadır. Normal resim sınıflandırma algoritmaları milyonlarca resim ile eğitilirken x-ışını röntgen görüntüleri için bu sayı 10 bin ile 100 bin arasında kalmaktadır.



Bu durumun üstesinden gelmek için daha önce milyonlarca resim içeren gündelik nesnelere üzerinden eğitilmiş olan algoritmaların ağırlıklarını kullanabiliriz.



Bunu ikinci bölümde transfer öğrenmesi başlığı altında açıklamıştık. Bu durum özellikler aylar süren eğitim sürecinden bizleri kurtardığı gibi aynı zamanda veri setinin az olduğu durumlarda da oldukça işimizi kolaylaştırmaktadır.

Şekil 3.16'da hayvanlar için eğitilmiş bir modele sahip olduğumuzu düşünelim.



Şekil 3. 16 Veri Seti Boyutu Sorunu İçin Transfer Öğrenmesi 2 [51]

Bu durumda modelimizin ilk katmanları bu hayvanlara ait genel sayılabilecek köşe ve kıvrımlar gibi özellikleri öğrenmektedir. Öğrenilen bu özellikler röntgen görüntülerindeki akciğer ve diğer iç organların sınır ve şekillerini tespit etmek içinde kullanılabilir. Katmanlar derinleştikçe ilgili hayvana ait detaylar örneğin göz rengi, kafa şekli vb. detaylar öğrenilir. Bu sebeple ilk katmanları dondurup diğer katmanlardaki ağırlıkları ince ayar (fine tuning) da denilen işlem ile yeniden güncelleyerek modeli kullanabiliriz. Veya ilk katmanları dondurmamak yerine tüm katmanlar için ince ayar yapmakta bir diğer seçenektir. Her iki durumda da bu bize hem zaman kazandıracak hem de veri setinin az olması problemini çözecektir [51].

3.6 Medikal Görüntüler İçin Veri Artırma (Data Augmentation)

Veri setini büyüklüğü probleminin üstesinden gelmenin bir diğer yolu da veri artırmadır. Bu işlem için bir resmin rengini, parlaklığını değiştirebilir, bir ekseninde sağa, sola aşağı veya yukarı döndürebilir veya resmin belli kısımlarına yakınlaştırma vb. yöntemler uygulayarak birden fazla resim üretebiliriz.



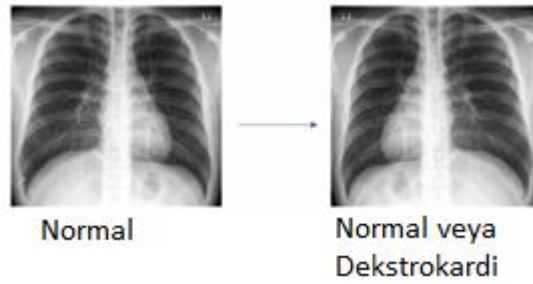
Şekil 3. 17 Göğüs Röntgeni İçin Veri Artırma [51]

Ancak göğüs röntgen görüntülerinde bu işlemi yaparken çok dikkatli olmamız gerekmektedir. İlk resimden ürettiğimiz ikinci resmin hastalık sınıfının korunması gerekmektedir. Özellikle parlaklığı artırma işlemi bu resimler için oldukça yaygındır. Bu işlem hastalık sınıfında herhangi bir değişikliğe de yol açmaz.



Şekil 3.18 Veri Artırma Örneği – Kontrast Değişimi [51]

Fakat aynı resmi dikey ekseninde sağa veya sola çevirme işlemi uygularsak,

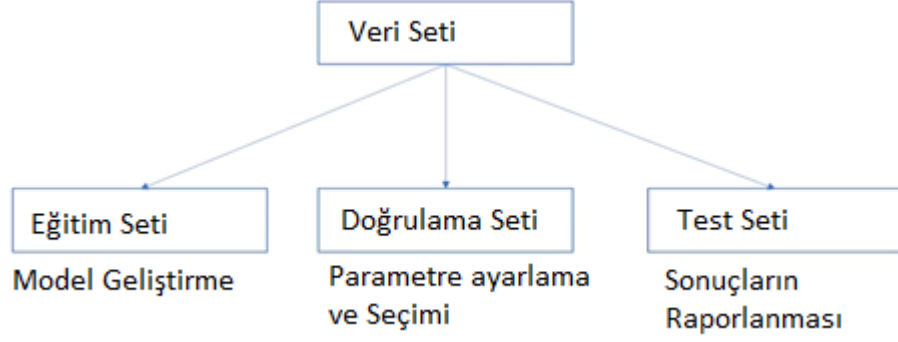


Şekil 3.19 Veri Artırma Örneği – Dikey Ekseninde Çevirme [51]

Şekil 3.19'daki gibi bir işlem yapıldığında resimde görülen kalbin yer değiştirdiği görülmektedir. Bu durum kalbin sol taraf yerin sağ tarafta yer aldığı dekstrokardi adı verilen nadir bir kalp rahatsızlığı olarak tanımlanır. Bu ise artık birinci resimdeki sınıfın korunmadığı anlamına gelir ve bu işlem veri artırma olarak kullanılamaz [51].

3.7 Model İçin Test ve Eğitim Verisi Oluşturma

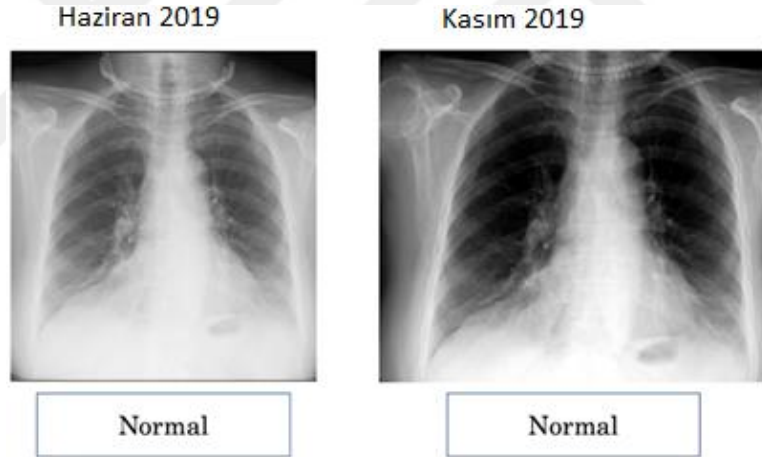
Veri setimizi modelimizde kullanabilmek için onu test ve eğitim olmak üzere iki ana parçaya ayırmamız gereklidir. Ayrıca eğitim setini de model geliştirme ve model doğrulama (validation) olmak üzere ikiye böleriz.



Şekil 3. 20 Test -Eğitim Verisi [20]

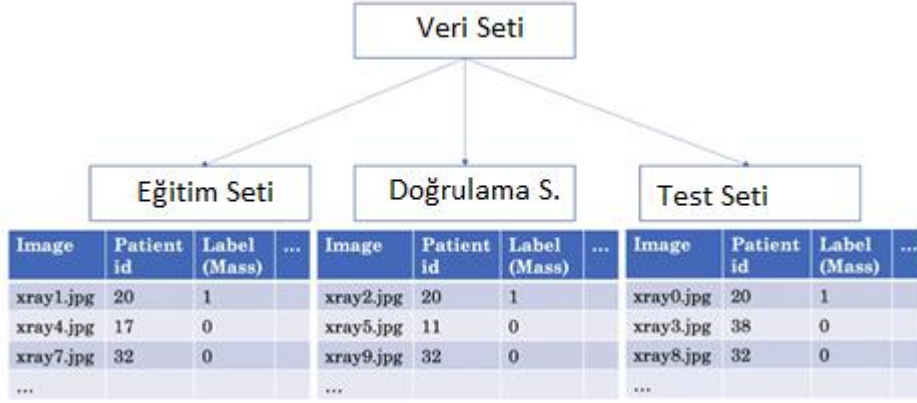
Böylece modelimizi geliştirmek için kullandığımız eğitim setinin sonuçlarını doğrulama seti ile ölçümleyebiliriz ve model parametreleri ile ilgili ince ayarlamalar yapabiliriz.

Ayrıca bu tarz medikal görüntülerde birtakım zorluklar da karşımıza çıkmaktadır.



Şekil 3. 21 Bir Hastaya Ait İki Farklı Röntgen Görüntüsü [51]

Örneğin şekil 3.21 da bir hastanın farklı zamanlarda çekilmiş iki röntgen görüntüsü görülmektedir. Bu röntgen görüntülerine dikkatli bakılırsa her ikisinde de hastanın aynı kolyeyi taktığı görülmektedir. Bu durum modelimizin bu durumu ezberleyerek her kolye tespit ettiği resim için benzer sonuçlar vermesine neden olabilir.



Şekil 3. 22 Bir Hastaya Ait Aynı Resimleri Ayırma [51]

Bu durumu. Düzeltmek için şekil 3.22 deki gibi aynı hastalara ait resimleri oluşturduğumuz eğitim ve test veri setlerine dağıtmalıyız. Ayrıca test ve eğitim veri setlerimizin bir hastalığa ait resimleri de %50 oranında yani eşit şekilde almış olması gereklidir [51].

3.8 Model İçin Değerlendirme Metrikleri

3.8.1 Doğruluk (Accuracy)

Bir modelin ne kadar iyi çalıştığını ölçmenin yollarından biri doğruluk metriğidir. Doğruluk şu şekilde tanımlanır.

$$\text{Doğruluk} = \frac{\text{Doğru Örnek Sayısı}}{\text{Toplam Örnek Sayısı}} \quad (3.7)$$

Fakat bu ölçme şekli özellikle tıbbi görüntülerde ve verilen dengeli dağıtılmadığı örnek kümelerinde oldukça sorunlu bir yaklaşım olarak karşımıza çıkabilir.

Bunu göstermek için 8 resimden oluşan bir veri setine sahip olduğumuzu düşünelim. Bu set içinde 6 adet resmin normal 2 adet resmin ise hastalıklı olarak etiketlendiğini düşünelim.

Tablo 3. 7 Doğruluk Metriği İçin Örnek Hesaplama [51]

Gerçek Değer	Model 2	Model 1
Normal	negatif	negatif
Normal	pozitif	negatif
Normal	negatif	negatif
Normal	negatif	negatif
Normal	negatif	negatif
Hasta	pozitif	negatif
Normal	negatif	negatif
Hasta	pozitif	negatif
Normal	pozitif	negatif
Normal	negatif	negatif

Bu veri seti için 2 adet model oluşturduğumuzu varsayalım. Bunlardan birincisi tüm resimleri normal olarak etiketlesin. Bu durumda birinci model için doğruluk,

$$\frac{8}{10} = 0.8$$

Şeklinde bulunur. Model 2 de birinci gibi 8 adet doğru etiketleme yapsın ancak hastalıklı resimleri de doğru olarak etiketlemiş olsun. Bu modelinde doğruluk değeri 0.8 çıkacaktır. Fakat açıkça görüleceği üzere model 2 model 1'den daha iyidir. Buradan da anlaşılacağı üzere bu metriği modellerimiz için kullanabilmemiz veri dağılımının durumuna bağlıdır.

Doğruluk metriği aynı zamanda doğru olma olasılığı olarak da tanımlanır.

$$Doğruluk = P(doğru) \quad (3.8)$$

$$Doğruluk = P(doğru \cap hasta) + P(doğru \cap normal) \quad (3.9)$$

Matematiksel olarak $P(A \cap B) = P(A|B)P(B)$ şeklinde ifade edildiğine göre,

$$Doğruluk = P(doğru | hasta)P(hasta) + P(doğru | normal)P(normal) \quad (3.10)$$

$$Doğruluk = P(+ | hasta)P(hasta) + P(- | normal)P(normal) \quad (3.11)$$

Şeklinde tanımlayabiliriz [51].

3.8.2 Duyarlılık, Özgüllük ve Yaygınlık

Duyarlılık

Duyarlılık (Sensitivity) gerçek pozitif oranı olarak da bilinir. Bir modelin bir hasta için hastalığı hangi oranla tespit ettiğı ya da modelin bir hastayı hastalığa sahip olarak sınıflandırma olasılığıdır.

Özgüllük

Özgüllük (Specificity) gerçek negatif oranı olarak da bilinir. Bir modelin bir hasta olmayan bir kişi için doğru yani negatif teşhis koyma oranını belirtir.

Yaygınlık

Yaygınlık (Prevalans) bir popülasyonda bir hastanın hastalığa sahip olma olasılığı şeklinde tanımlanır.

Şimdi bunları matematiksel olarak tanımlayalım,

$$Duyarlılık = P(+ | hasta) \quad (3.12)$$

$$Özgüllük = P(- | normal) \quad (3.13)$$

$$Yaygınlık = P(hasta) \quad (3.14)$$

Bu durumda denklem 3.11 de verdiğimiz doğruluk değerini şu şekilde yeniden ifade edebiliriz,

$$Doğruluk = Duyarlılık \times Yaygınlık + Özgüllük \times (1 - Yaygınlık) \quad (3.15)$$

Şimdi bunu bir örnek üzerinden gösterelim,

Tablo 3. 8 Örnek Model [51]

Gerçek Değer	Model
Normal	negatif
Normal	negatif
Hasta	pozitif
Normal	negatif
Normal	negatif
Hasta	negatif
Normal	negatif
Hasta	pozitif
Normal	pozitif
Normal	negatif

Tablo 3.7'deki gibi 2 adet hasta ve 8 adet normal olmak üzere toplam 10 adet veri içeren bir veri setinde çalışan modelimizin elde ettiği sonuçlar görülmektedir. Buradan hareketle yukarıda tanımladığımız metrikler için değerleri hesaplayacak olursak,

$$\text{Duyarlılık} = P(+ | hasta) = \frac{+ \text{ ve hasta}}{\text{hasta}} = \frac{2}{3} = 0.67$$

$$\text{Özgüllük} = P(- | normal) = \frac{- \text{ ve normal}}{\text{normal}} = \frac{6}{7} = 0.86$$

$$\text{Yaygınlık} = P(\text{hasta}) = \frac{\text{hasta}}{\text{normal}} = \frac{3}{10} = 0.3$$

Bu durumda doğruluk,

$$\text{Doğruluk} = 0.67 \times 0.3 + 0.86 \times (1 - 0.3) = 0.8$$

Şeklinde elde edilir [51].

3.8.3 Pozitif Prediktif Değer (PPD) ve Negatif Prediktif Değer (NPD)

Pozitif Prediktif Değer, bir testin hasta olarak bulduğu bireyler içinde gerçekten hasta olan bireylerin oranıdır.

Negatif Prediktif Değer, bir testin sağlıklı olarak bulduğu bireyler içinde gerçekten sağlıklı olanların yüzdesidir.

Tablo 3.7'deki gibi aynı veri setine sahip olduğumuzu düşünelim.

Tablo 3. 9 Örnek Model 2 [51]

Gerçek Değer	Model
Normal	negatif
Normal	pozitif
Hasta	pozitif
Normal	negatif
Normal	negatif
Hasta	negatif
Normal	negatif
Hasta	pozitif
Normal	pozitif
Normal	negatif

Bu durumda modelimizin sonuçlarına göre PPD ve NPD değerleri şu şekilde hesaplanır,

$$PPD = P(hasta | +) = \frac{+ ve hasta}{+} = \frac{2}{4} = 0.5$$

$$NPD = P(normal | -) = \frac{- ve normal}{-} = \frac{5}{6} = 0.83$$

Şeklinde bulunur [51].

3.8.4 Karışıklık Matrisi (Confusion Matrix)

Karışıklık matrisi PPD, NPD ve diğer metrikler arasındaki ilişkiyi kolayca görebileceğimiz ve sınıflandırmada çok sık kullanılan bir karşılaştırma tablosudur.

Bu tabloda satırlar temel gerçeğe (ground truth) ve sütunlar ise model tahminlerine karşılık gelmektedir.

Tablo 3. 10 Karışıklık Matrisi 1 [51]

Gerçek Değer				Model
Normal				negatif
Hasta				pozitif
Normal	Model Çıktısı			pozitif
Normal		pozitif	negatif	negatif
Normal	Hasta	2	1	negatif
Hasta	Normal	2	5	negatif
Normal				negatif
Hasta				pozitif
Normal				pozitif
Normal				negatif

Tablo 3.10'daki gibi bir veri setimiz ve model sonuçlarımız var ise karışıklık matrisini şu şekilde gösterebiliriz,

Tablo 3. 11 Karışıklık Matrisi 2 [51]

		Model Çıktısı		Model Çıktısı		
		pozitif	negatif		pozitif	negatif
Gerçek Değer	Hasta	Gerçek Pozitif (TP)	Yanlış Negatif (FN)	Hasta	2	1
	Normal	Yanlış Pozitif (FP)	Gerçek Negatif (TN)	Normal	2	5

Burada,

Gerçek Pozitif (TP): Gerçek değeri 1 (hasta) ve tahmin ettiğimiz değerin de 1 (hasta) olduğu örneklerdir.

Gerçek Negatif (TN): Gerçek değeri 0 (normal) ve tahmin ettiğimiz değerin de 0 (normal) olduğu örneklerdir.

Yanlış Pozitif (FP): Gerçek değeri 0 (normal) ancak tahmin ettiğimiz değerin 1 (hasta) olduğu örneklerdir.

Yanlış Negatif (FN): Gerçek değeri 1 (hasta) ancak tahmin ettiğimiz değerin 0 (normal) olduğu örneklerdir.

Tüm bu değerlerin karışıklık tablosundaki ilişkilerini aşağıdaki gibi gösterebiliriz,

Gerçek Değer		Model Çıktısı		$Duyarlılık = \frac{TP}{TP + FN}$
		pozitif	negatif	
	Hasta	Gerçek Pozitif (TP)	Yanlış Negatif (FN)	$Özgüllük = \frac{TN}{FP + TN}$
Normal	Yanlış Pozitif (FP)	Gerçek Negatif (TN)		
		$PPD = \frac{TP}{TP + FP}$ $NPD = \frac{TN}{TN + FN}$		

Şekil 3. 23 Karışıklık Matrisi 3

Bu şekle bakarak,

$$Duyarlılık = \frac{TP}{TP + FN} \quad (3.16)$$

$$Özgüllük = \frac{TN}{FP + TN} \quad (3.17)$$

$$PPD = \frac{TP}{TP + FP} \quad (3.18)$$

$$NPD = \frac{TN}{TN + FN} \quad (3.19)$$

Denklemlerini yazabiliriz.

3.8.5 AUC-ROC Eğrisi (Eğri Altındaki Alan-Alıcı işlem karakteristikleri / Receiver Operating Characteristic)

ROC eğrisi, geliştirilen tıbbi modelleri değerlendirmek için kullanılan en önemli araçlardan birisidir. ROC eğrisi belirli eşik (treshold) değerleri altında bir modelin duyarlılığının özgüllüğüne karşı görsel olarak çizilmesini sağlar. [56][57][58]

Geliştireceğimiz bir göğüs hastalıkları sınıflandırma modeli ile verilen bir röntgen görüntüsünün hastalıklı olma olasılığını alabiliriz. Böylece sistemden elde edilen bu değer belirleyeceğimiz bir sınır değer ve çalışma noktası ile bir teşhis yöntemi olarak kullanılabilir hale gelebilir.

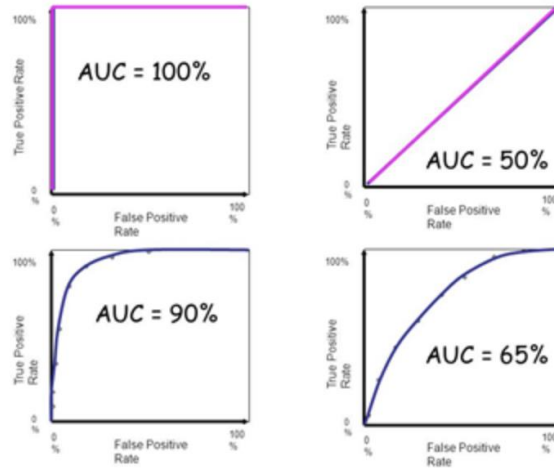


Şekil 3. 24 ROC Eğrisi İçin Model [51]

Örneğin eşik değerine t dersek ve modelimizin bir göğüs röntgeni için verdiği değer eşiğin üzerinde ise bunu pozitif olarak veya kişinin hastalığı olduğunu söyleyerek yorumlarız. Eğer verilen olasılık eşiğin altında ise bunu olumsuz ya da kişide hastalık yok şeklinde yorumlarız.

Örneğin t değerimiz 0.5 olsun. Eğer modelden elde edilen olasılık değeri 0.7 ise bu örneği pozitif (hasta) olarak sınıflandırırız. Ancak değerimiz 0.2 ise bu örneği negatif (normal) olarak sınıflandırırdık. Burada eşik seçimi çok önemlidir.

AUC, ROC eğrisi altında kalan alan anlamına gelmektedir. Bu alanın kapsamı AUC'dur. Kapsanan alan ne kadar büyükse, geliştirilen model o kadar iyi sınıflandırma yapıyor demektir. AUC için ideal değer 1'dir.



Şekil 3. 25 AUC- ROC Eğrisi Karşılaştırmaları

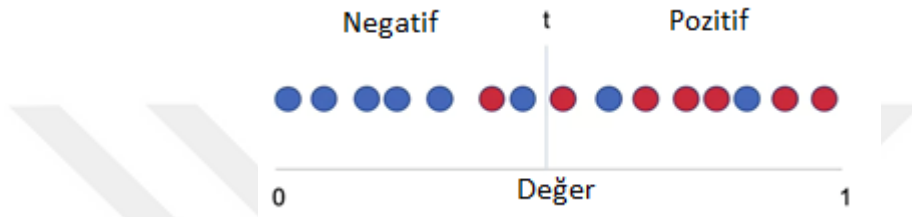
AUC alanı için şekil 3.25'e bakarak şunları söyleyebiliriz [59],

1. 0.90-1.00 = mükemmel
2. 0.80-0.90 = iyi
3. 0.70-0.80 = orta

4. 0.60-0.70 = zayıf
5. 0.50-0.60 = başarısız

Şimdi seçilen eşik değerinin sonuca olan etkisini inceleyelim,

Elimizde 15 adet resim içeren bir göğüs röntgeni veri olduğunu düşünelim. Bu görüntülerin her biri için modelimizden aldığımız sonuçları eğer eşik değerinden küçükse eşik değerinin soluna ve eşik değerinden büyük ise sağına yerleştirelim. Ayrıca bu resimlerin gerçek değerleri eğer normal ise mavi hasta ise kırmızı renkli olacak şekilde işaretleyelim.



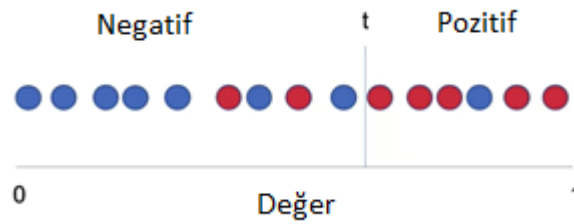
Şekil 3. 26 Eşik Değeri Gösterimi [51]

İlk durumda t eşik değerinin konumuna göre ölçülen metrikler,

$$Duyarlılık = P(+ | hasta) = \frac{6}{7} = 0.85$$

$$Özgüllük = P(- | normal) = \frac{6}{8} = 0.75$$

Şeklinde hesaplanır. Şimdi eşik değerinin yerini değiştirelim,

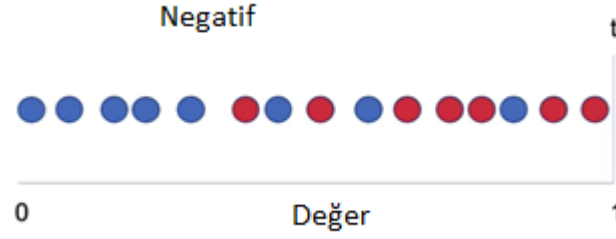


Şekil 3. 27 Eşik Değeri Gösterimi 2 [51]

Bu durumda ölçülen metrikler,

$$Duyarlılık = P(+ | hasta) = \frac{5}{7} = 0.71$$

$$Özgüllük = P(- | normal) = \frac{7}{8} = 0.88$$



Şekil 3. 28 Eşik Değeri Gösterimi 3

Son durumda ise eşik değerinin en sağa alınması sonuçları,

$$\text{Duyarlılık} = P(+ | hasta) = \frac{0}{7} = 0$$

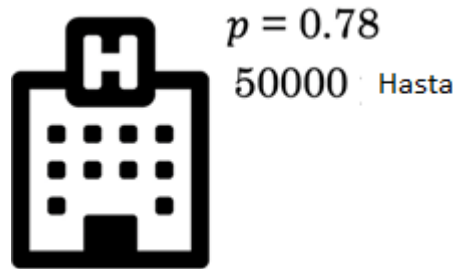
$$\text{Özgüllük} = P(- | normal) = \frac{8}{8} = 1$$

Görüldüğü şekilde değiştirir.

3.8.6 Güven Aralığı (Confidence Interval)

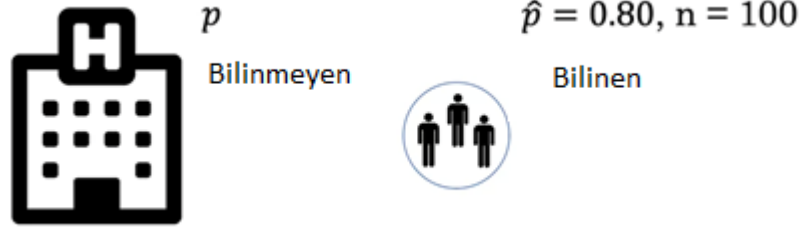
Tıbbi modelleri değerlendirmenin çok önemli bir başka yolu da güven aralığı ile örneklem yapmaktır. 50.000 kişiden oluşan bir hastanedeki tüm hastalar için geliştirdiğimiz model ile röntgen görüntülerini değerlendirip modelimizin performansını ölçmek istediğimizi düşünelim.

Bu ölçüm sonunda model doğruluğu $p=0.87$ olsun. Buna aynı zamanda popülasyon doğruluğu da denir.



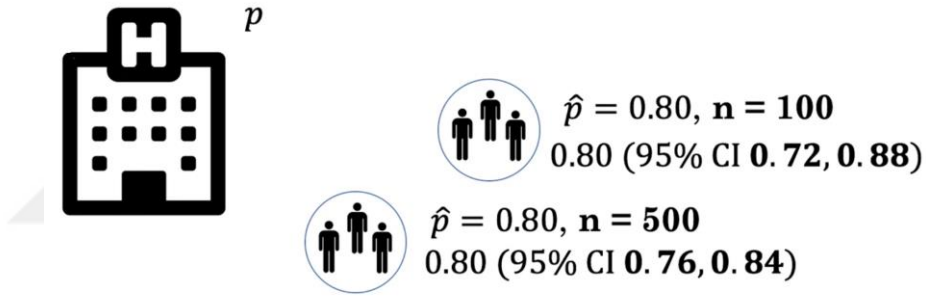
Şekil 3. 29 Popülasyon Doğruluğu [51]

Fakat gerçekte işler bu şekilde yürümez. Bütün hastalar için bir model değerlendirmesi yapılması birçok açıdan zor ve gereksiz olabilir. Böyle bir durumda yapılması gereken şey hedef hasta kümesi içerisinde belli bir oranda örnek alınarak o hastalar üzerinde test yapılması olacaktır.



Şekil 3. 30 Popülasyon İçin Örneklem Seçimi 1 [51]

Seçtiğimiz 100 kişilik hasta gurubu üzerinde yapılan model değerlendirmesi $p=0.80$ ve güven aralığımız %0,95 ise bu durumda tüm hastalar için genelleme yaparken popülasyon doğruluğu $p = [0.72, 0,88]$ aralığındadır diyebiliriz. Yani bulduğumuz sonuç tüm hasta popülasyonu için %95 oranında doğru %5 oranında ise hatalı çıkabilir demektir. Bulduğumuz aralığın değeri hem güven aralık yüzdesi hem de seçilen popülasyonun büyüklüğü ile ilişkilidir.



Şekil 3. 31 Popülasyon İçin Örneklem Seçimi 2 [51]

Şekil 3.31'deki gibi 100 yerine 500 kişilik bir örneklem gurubu oluşturup da $p=0.80$ bulduğumuzda %95 güven değeri ile bu seferki aralık $p= [0.76, 0,84]$ şeklinde olacaktır. Örneklem sayısı arttıkça aralık sayısı da azalmakta ve elde edilen sonuçlar gerçeğe daha yakın olmaktadır [60][61].

Bu bölümde çalışmış olduğumuz veri kümesinin tanıtımı yapılacak ve geliştirdiğimiz modelin sonuçları daha önce yapılan çalışmalar ile karşılaştırılacaktır.

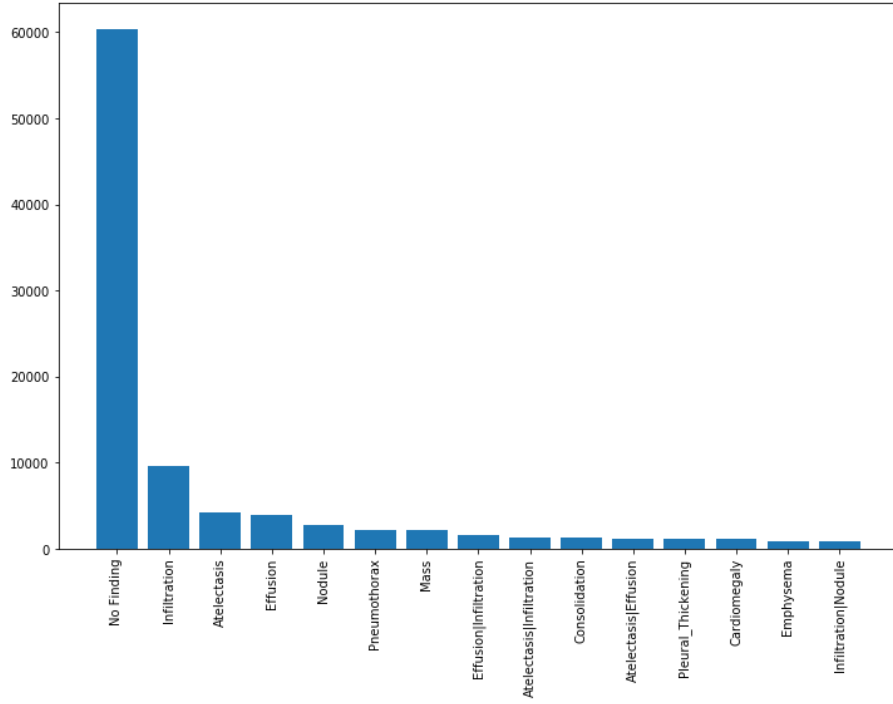
4.1 Çalışılan Veri Kümesi

Amerika Birleşik Devletleri Ulusal Sağlık Enstitüsü (NIH) [62] tarafından yayınlanan ve 30805 farklı hastanın 112120 adet önden çekilmiş etiketli göğüs röntgen görüntülerini içeren veri seti kullanılmıştır. Bu veri seti üzerinden tamamı gönüllü olarak katılan hastaların mahremiyet içerebilecek tüm bilgileri kaldırılmıştır [1].

Veri setinde yaygın göğüs hastalıklarından 13 tanesinin tekli ya da birlikte olabileceği şekilde kombine edilmiş etiketleme uygulanmıştır. Toplamda 14 sınıf için çalışma yaptığımız (1 sınıf hastalık bulunamadı) veri serindeki hastalıklar şöyledir,

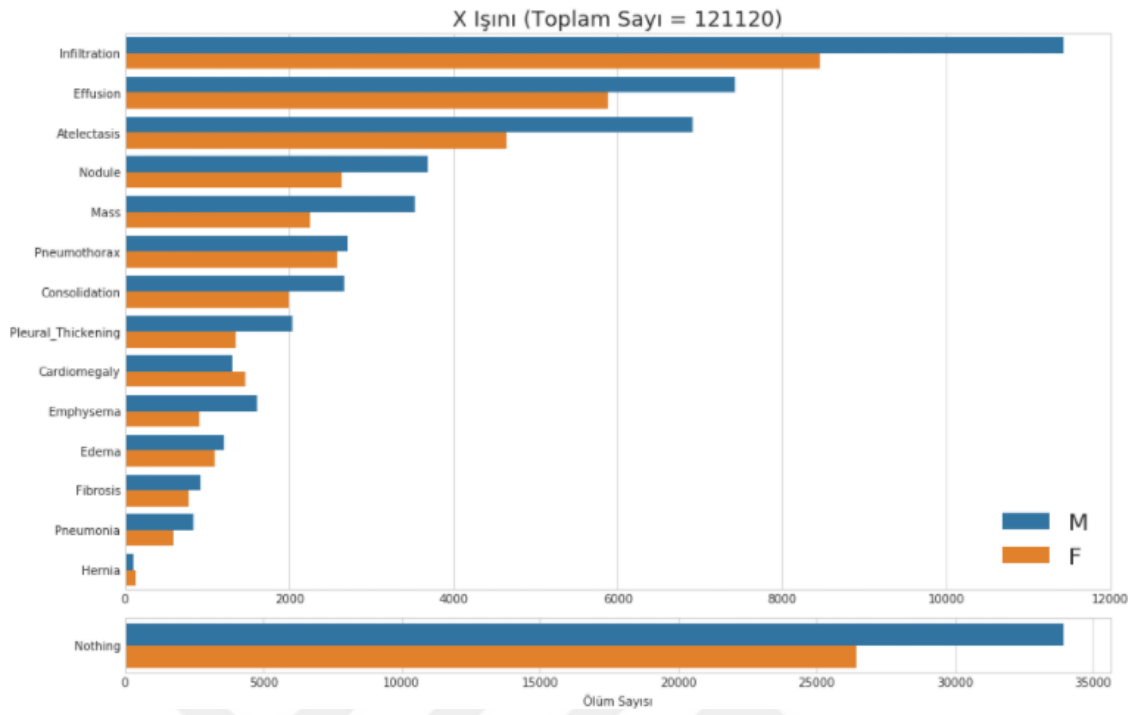
1. Atelektazi (Atelectasis)
2. Konsolidasyon (Consolidation)
3. İnfiltrasyon (Infiltration)
4. Pnömotoraks (Pneumothorax)
5. Ödem (Edema)
6. Amfizem (Emphysema)
7. Fibroz (Fibrosis)
8. Efüzyon (Effusion)
9. Zatürre (Pneumonia)
10. Plevral Kalınlaşma (Pleural Thickening)
11. Kardiyomegali (Cardiomegaly)
12. Nodül Kitlesi (Nodule Mass)
13. Kütle (Mass)
14. Fıtık (Hernia)

Şimdi bu hastalıkların veri setindeki görselleştirmelerine bakalım, hastalıkların veri setindeki dağılımları şöyledir,



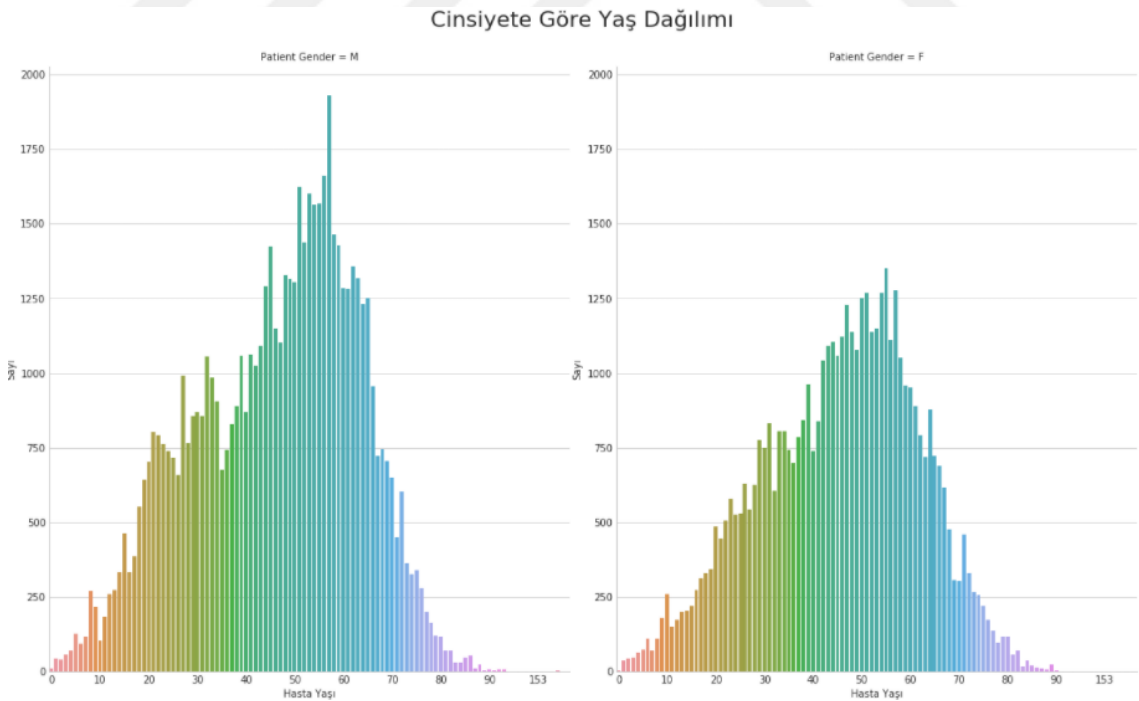
Şekil 4. 1 Veri Setindeki Hastalıkların Dağılımı

Hastaların cinsiyetine göre hastalık dağılımları,



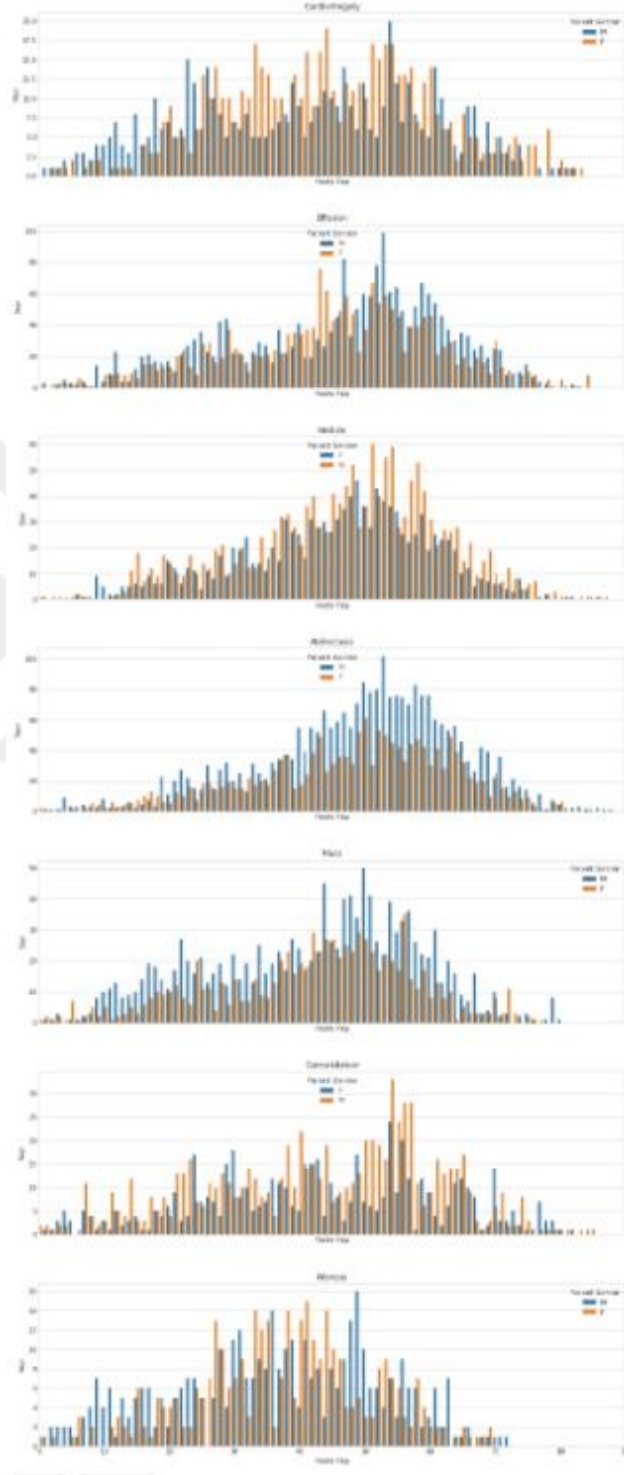
Şekil 4. 2 Cinsiyete Göre Hastalıkların Dağılımı

Cinsiyete göre yaş dağılımı,

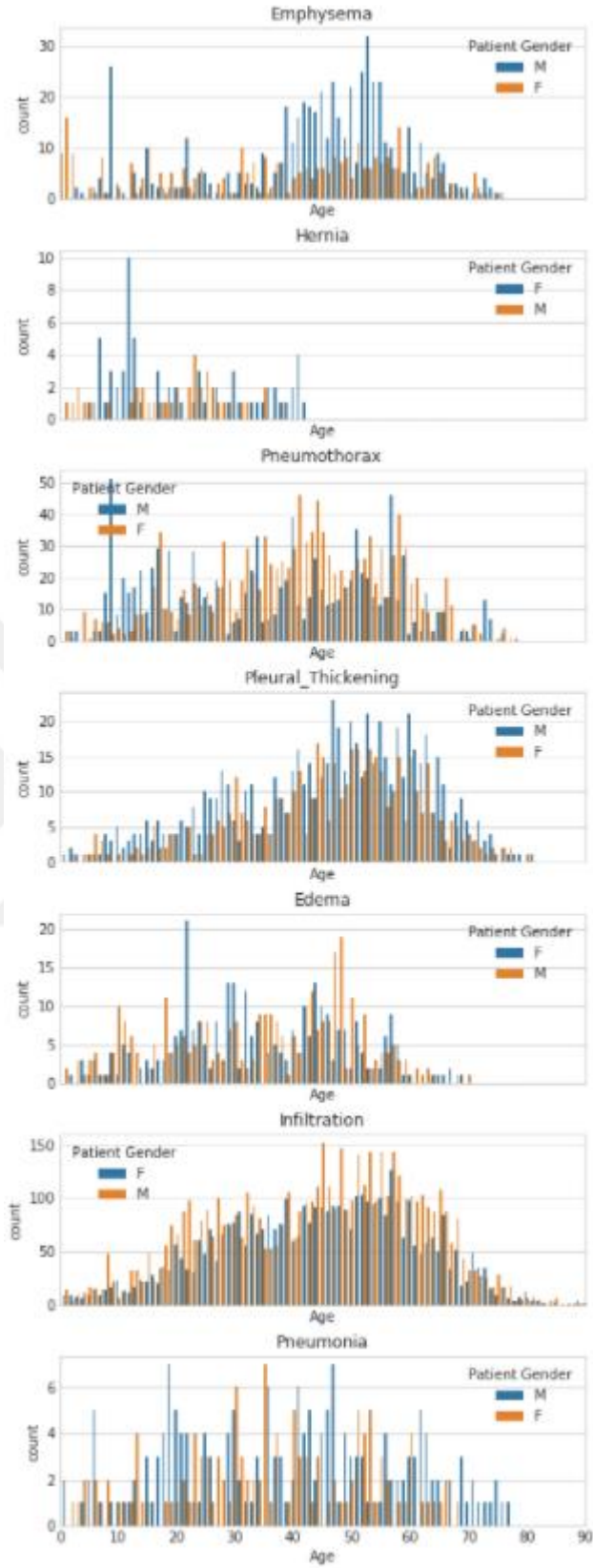


Şekil 4. 3 Cinsiyete Göre Yaş Dağılımı

Hastalıkların yaş ve cinsiyete göre dağılımı,

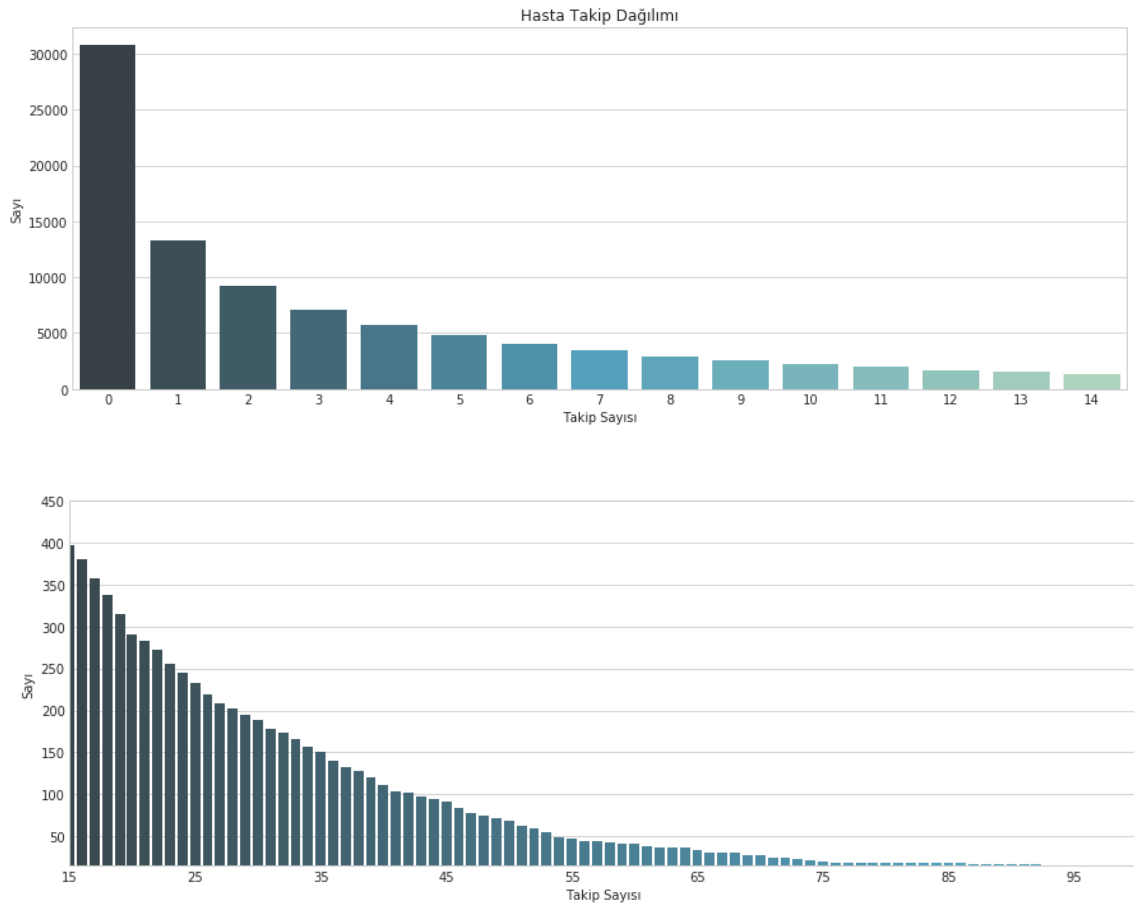


Şekil 4. 4 Hastalıkların Yaş ve Cinsiyete Göre Dağılımı -1



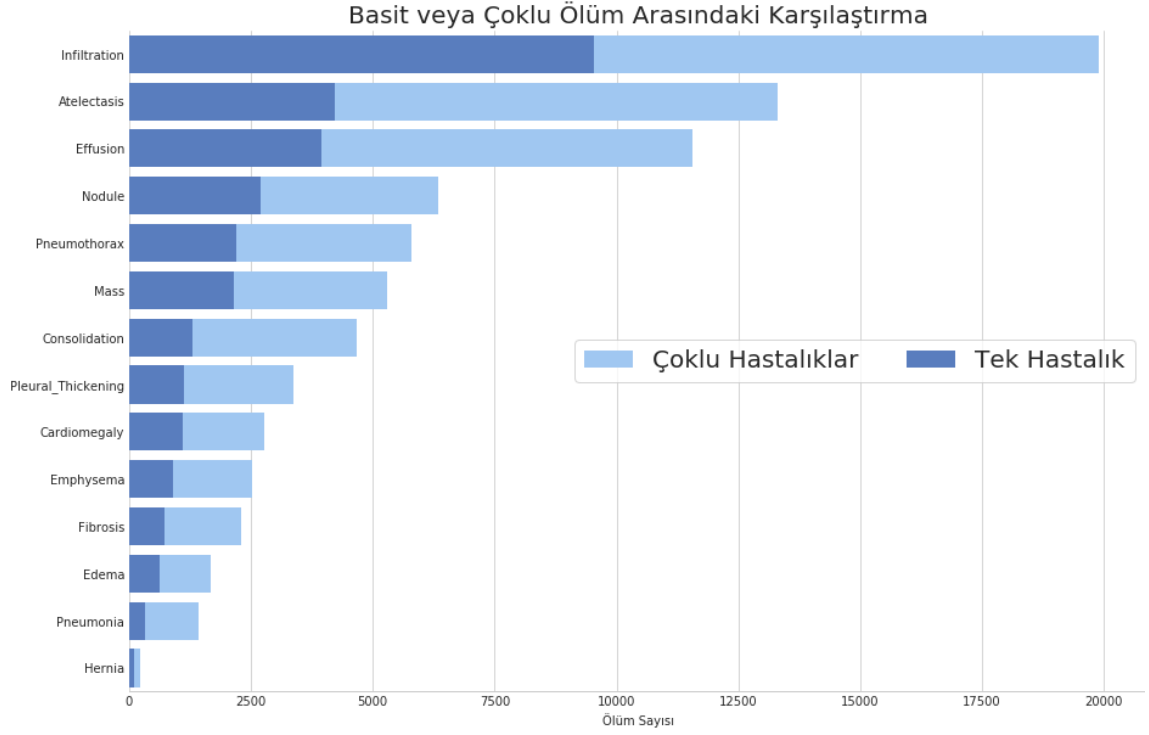
Şekil 4. 5 Hastalıkların Yaş ve Cinsiyete Göre Dağılımı -2

Takipli hastaların takip sayısına göre dağılımı,



Şekil 4. 6 Hastaların Takip Sayılarına Göre Dağılımı

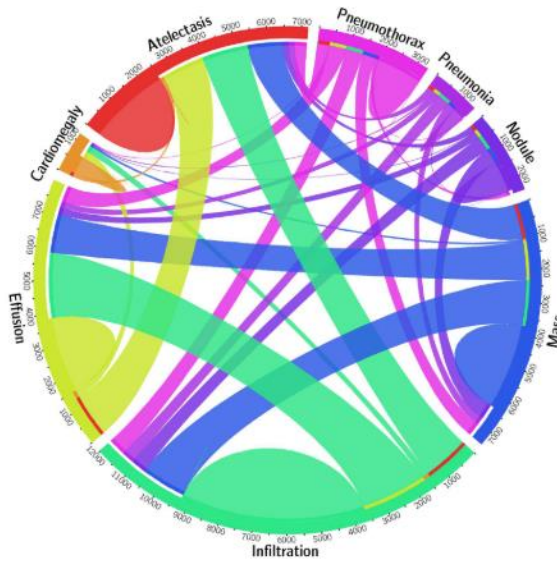
Veri setini incelediğimizde belirtmiş olduğumuz 14 hastalık hem tekil hem de çoğul olarak kullanılmıştır. Bu dağılım,



Şekil 4.7 Tekil ve Çoklu Hastalıkların Ölüm Dağılımı

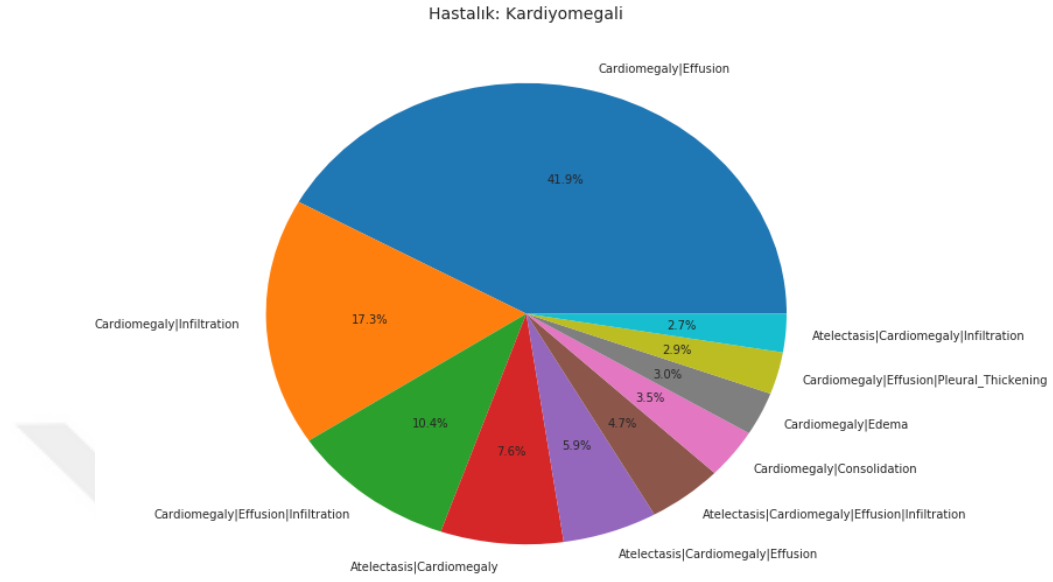
Şekil 4.7 de görüldüğü gibi olmaktadır.

Aşağıdaki resim ise veri setinde yer alan 8 önemli hastalığın çoklu etiketli görüntülerin oranı ve etiketlerin birlikte ortaya çıkma istatistiklerini göstermektedir [4].



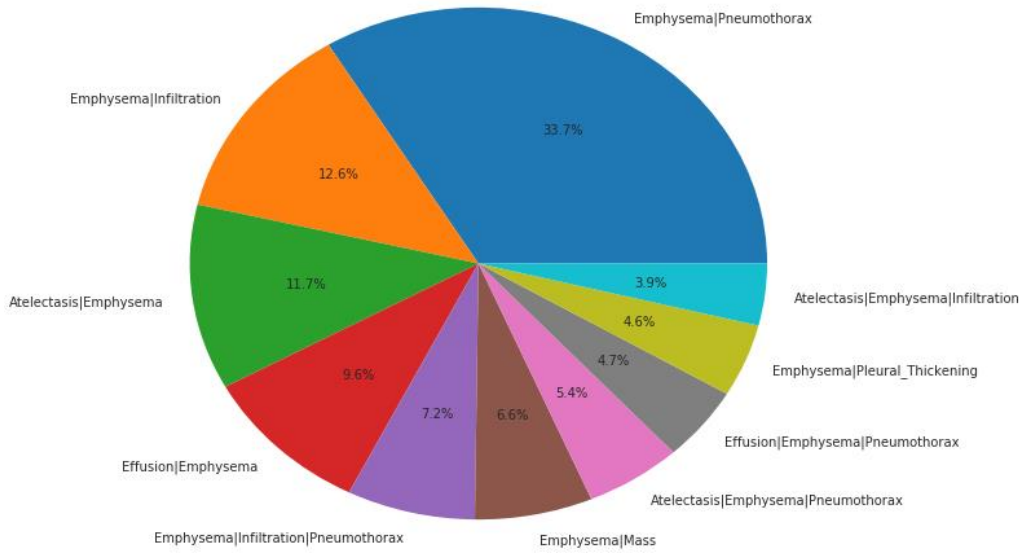
Şekil 4.8 Hastalıkların Birbirleri İle Olan Etkileşimi [4]

Şimdi bu hastalık gruplarını detaylı olarak ortaya koyalım. Öncelikle Kardiyomegali (Cardiomegaly) hastalığının eşlik ettiği ve tekil olarak yer aldığı hastalık grubuna bakalım,

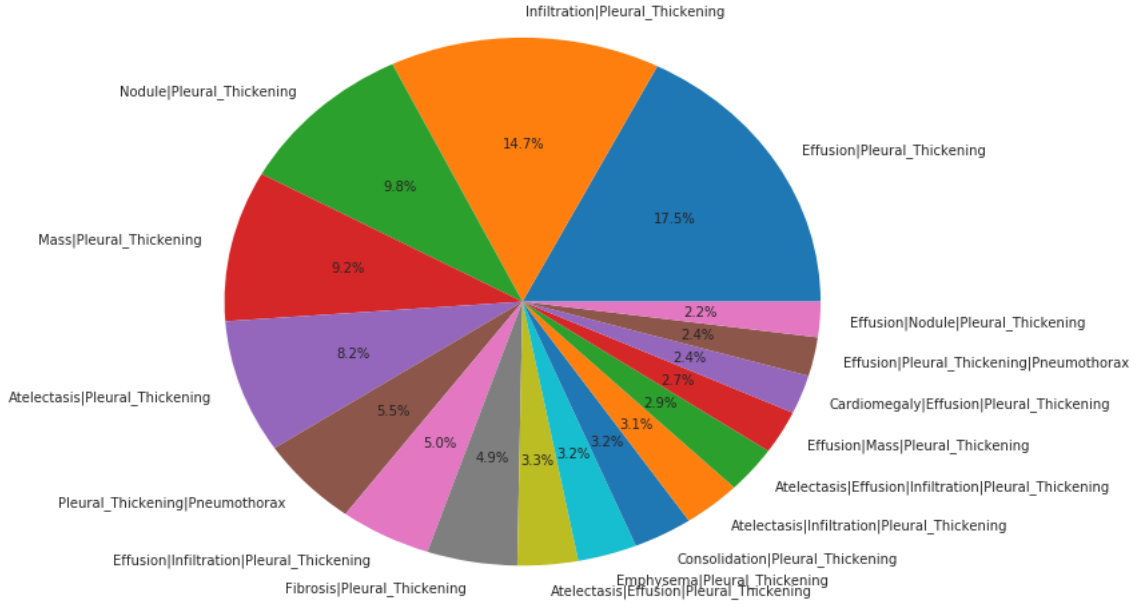


Şekil 4.9 Kardiyomegali 'nin Eşlik Ettiği Hastalık Dağılımı

Amfizem (Emphysema) hastalığının eşlik ettiği ve tekil olarak yer aldığı hastalık grubu,

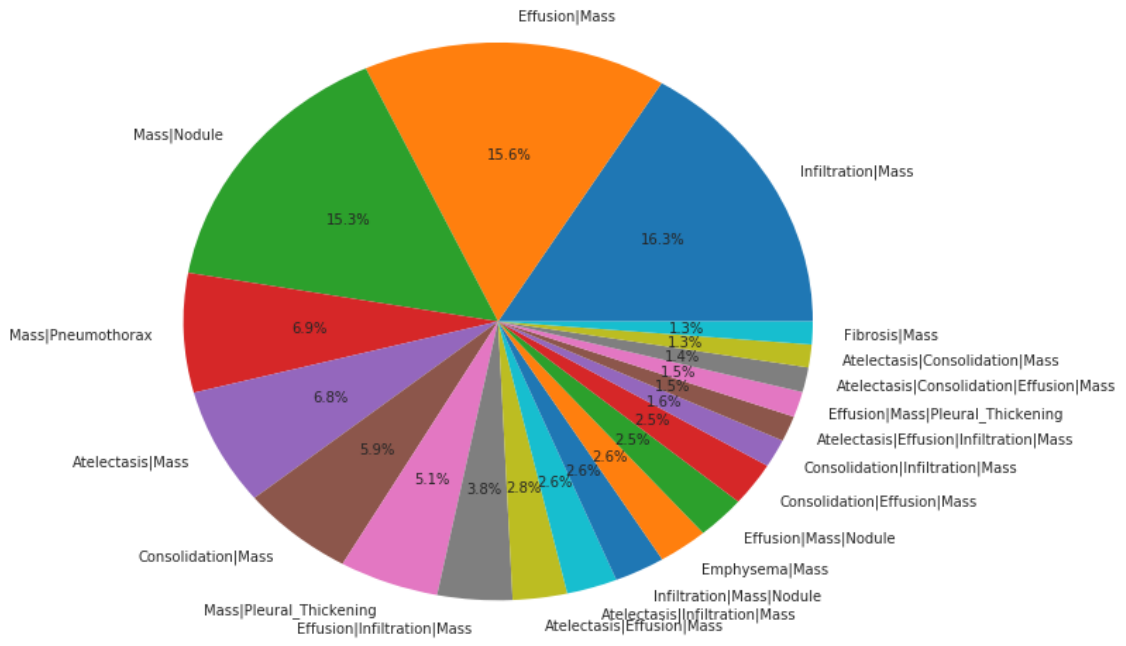


Şekil 4. 10 Amfizemin Eşlik Ettiği Hastalık Dağılımı



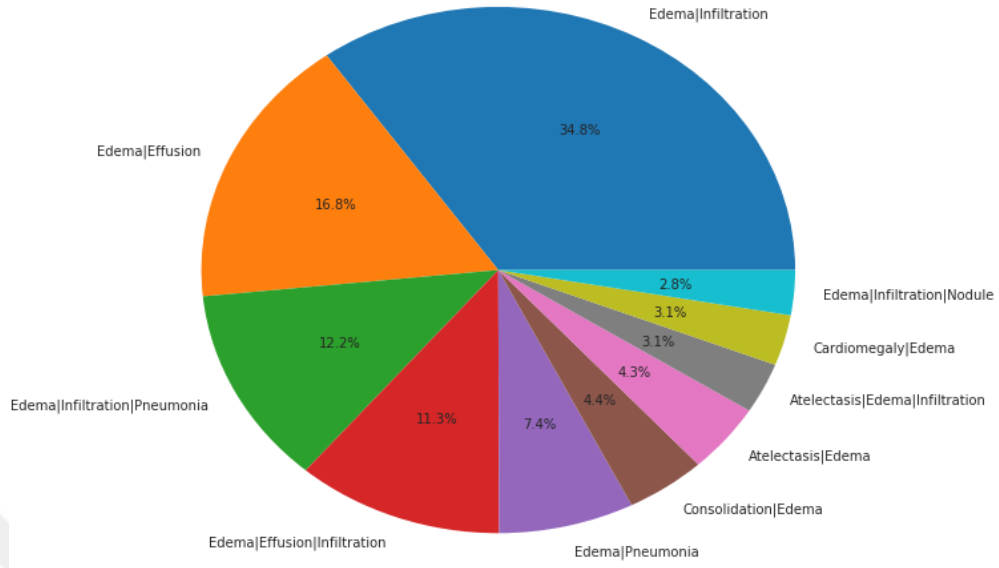
Şekil 4. 15 Plevral Kalınlaşmanın Eşlik Ettiği Hastalık Dağılımı

Kütle (Mass) hastalığının eşlik ettiği ve tekil olarak yer aldığı hastalık grubu,



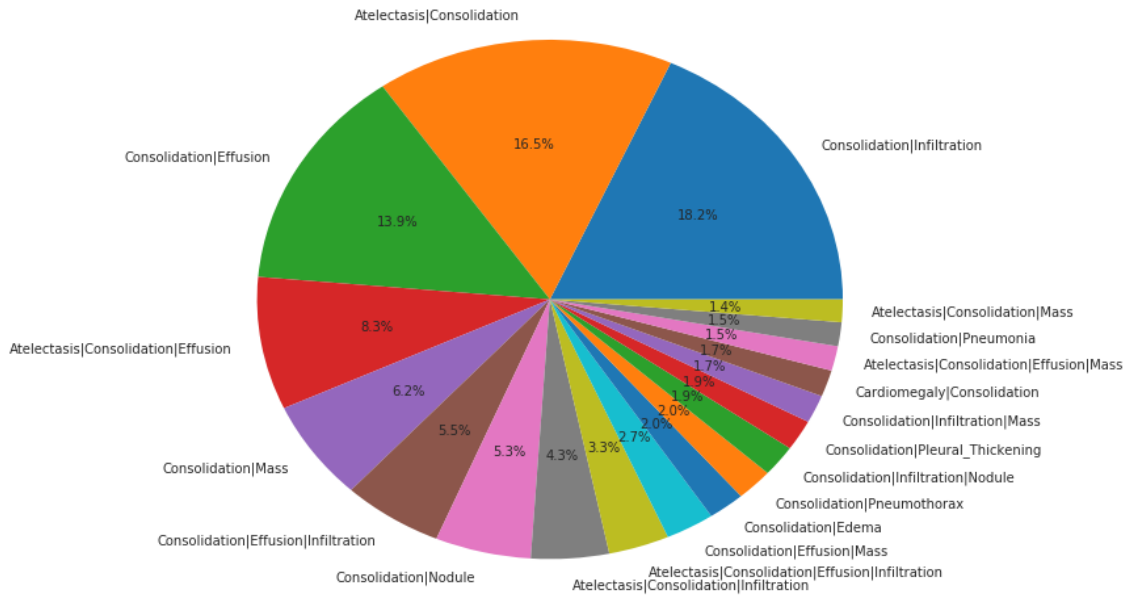
Şekil 4. 16 Kütle (Mass) Hastalığının Eşlik Ettiği Hastalık Dağılımı

Ödem (Edema) hastalığının eşlik ettiği ve tekil olarak yer aldığı hastalık grubu,

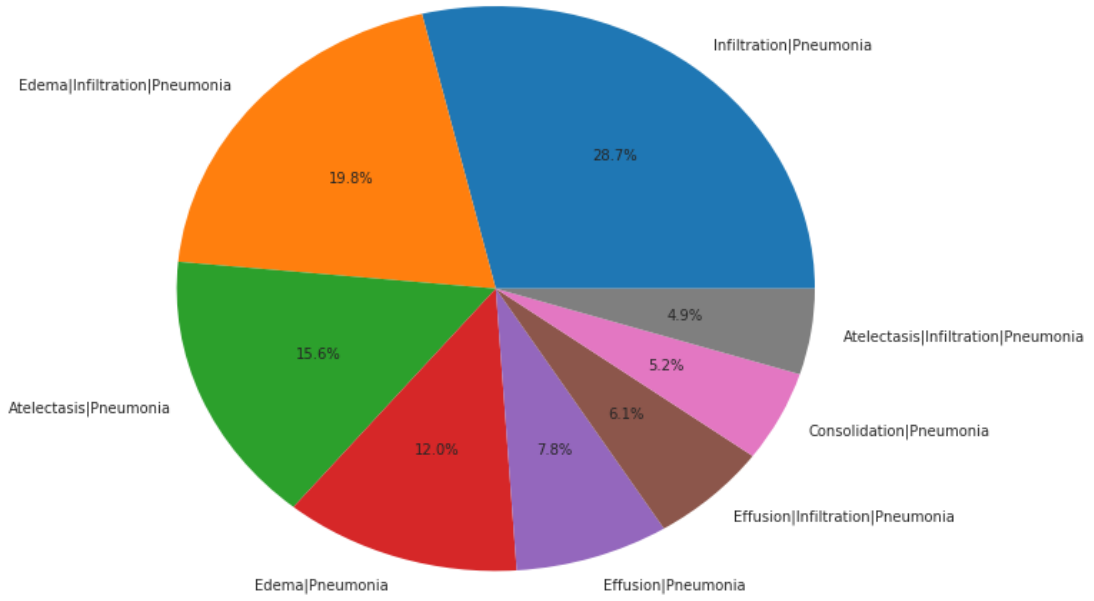


Şekil 4. 17 Ödemin Eşlik Ettiği Hastalık Dağılımı

Konsolidasyon (Consolidation) hastalığının eşlik ettiği ve tekil olarak yer aldığı hastalık grubu,



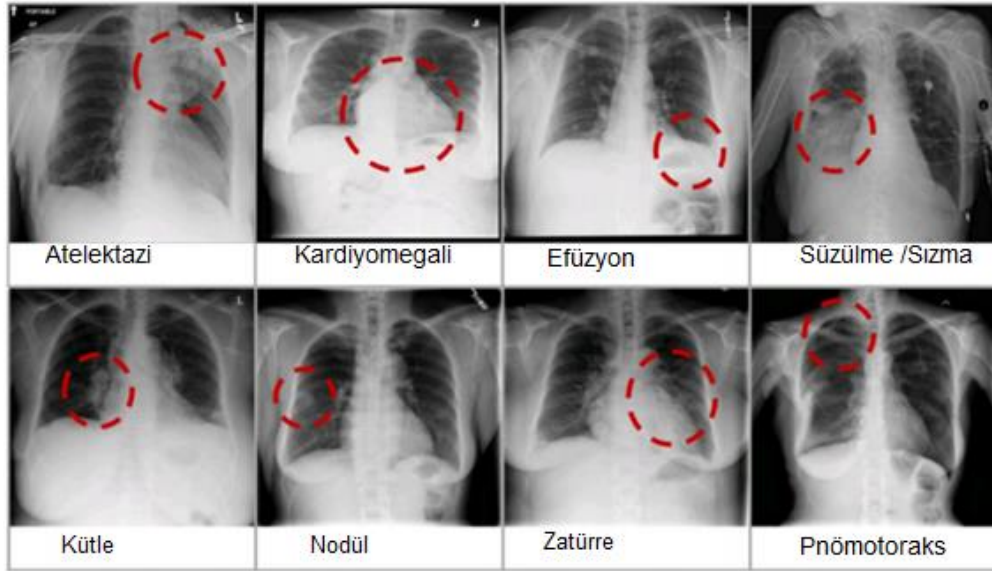
Şekil 4. 18 Konsolidasyonun Eşlik Ettiği Hastalık Dağılımı



Şekil 4. 21 Zatürrenin Eşlik Ettiği Hastalık Dağılımı

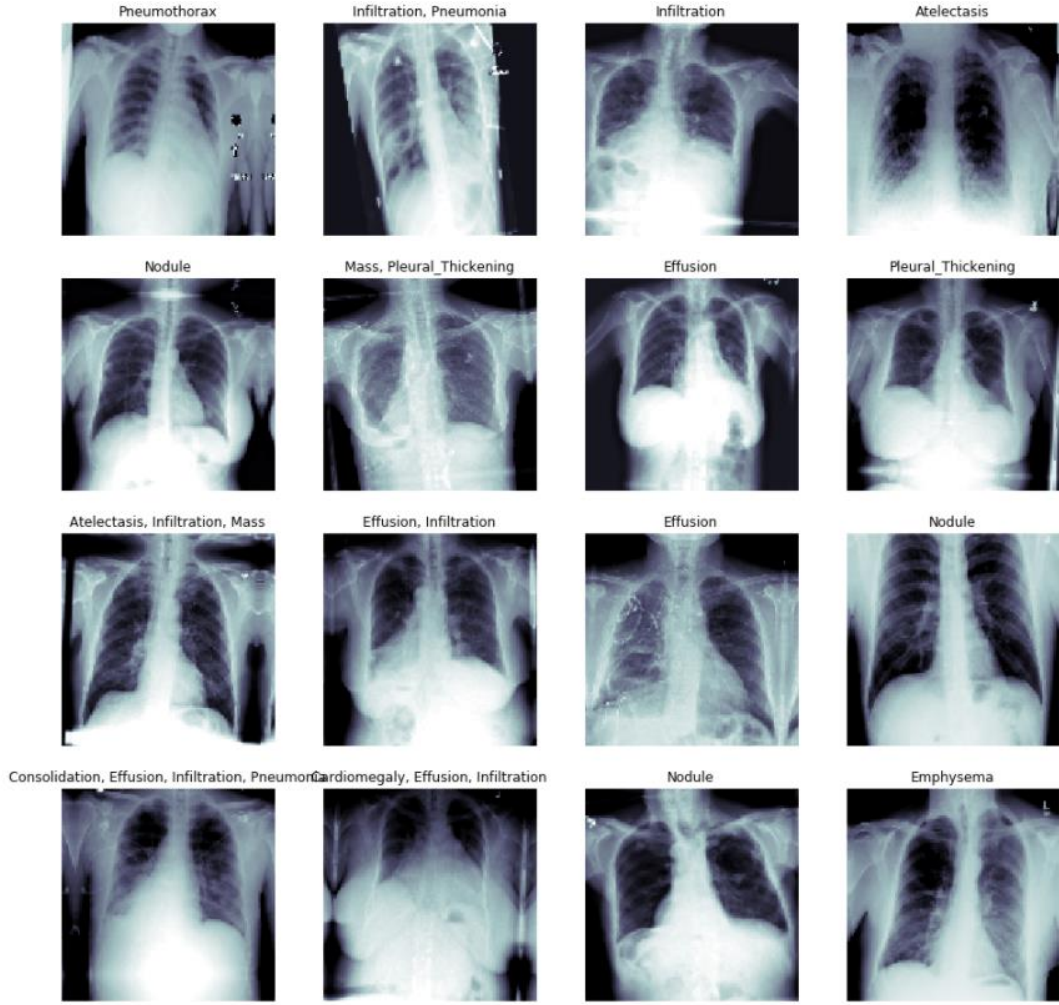
Fıtık (Hernia) için yeterli veri olmadığından bu dağılımda gösterilmemiştir. Grafiklerden de görüleceği üzere birçok hastalık iki veya daha fazla hastalığa da eşlik etmektedir.

En yaygın 8 hastalığın belirgin şekilde yer aldığı x-ışını görüntülerine bakalım [63],



Şekil 4. 22 8 Adet Göğüs Hastalığının X-Işını Gösterimi

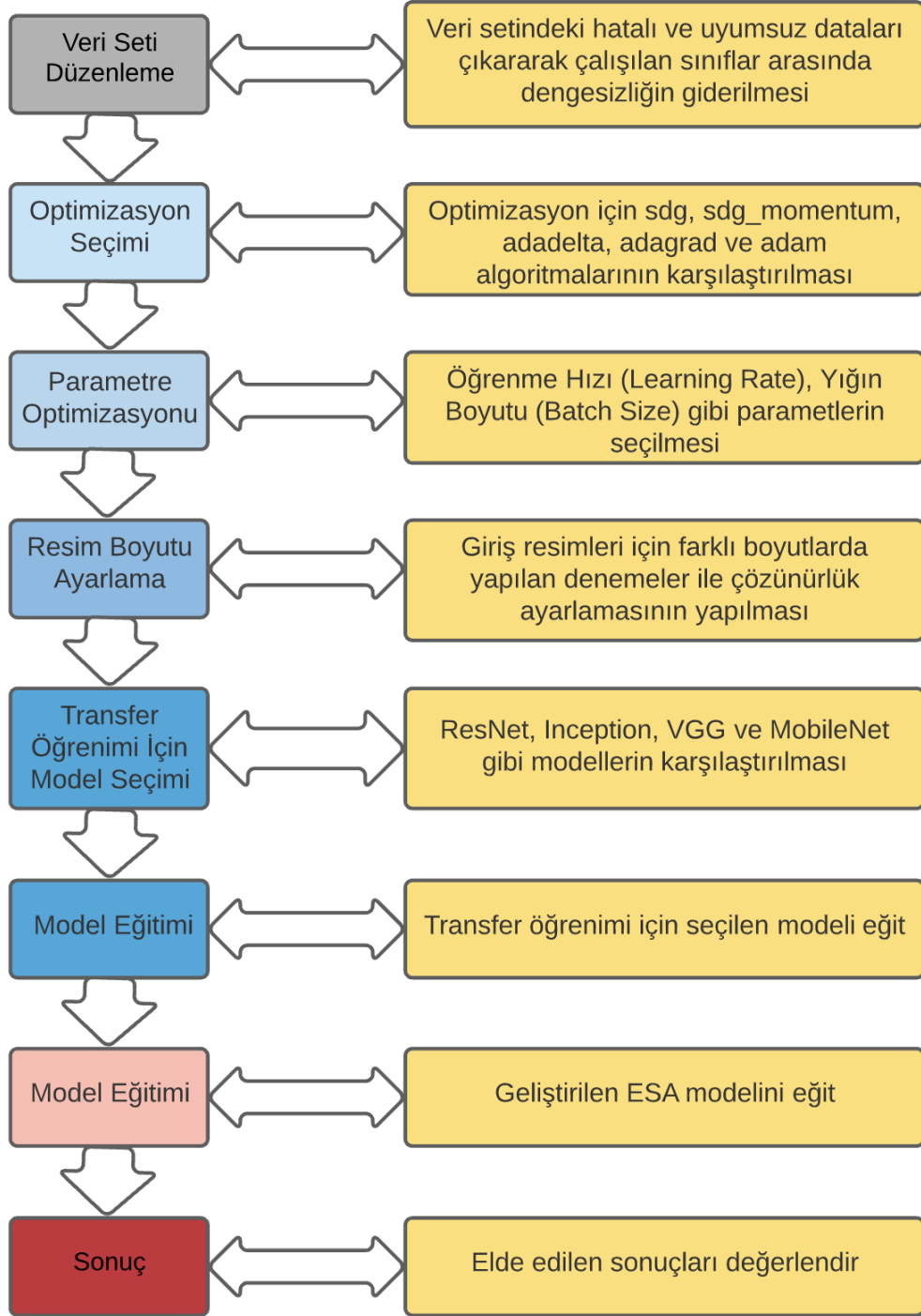
Veri setimizden alınmış rastgele resimler ile hastalıkların gösterimi şöyledir,



Şekil 4. 23 Veri Setinden Alınan Örnek Resimler

4.2 Geliştirilen Modelin Yapısı

Modelimizi geliştirirken uyguladığımız adımları şu şekilde gösterebiliriz,

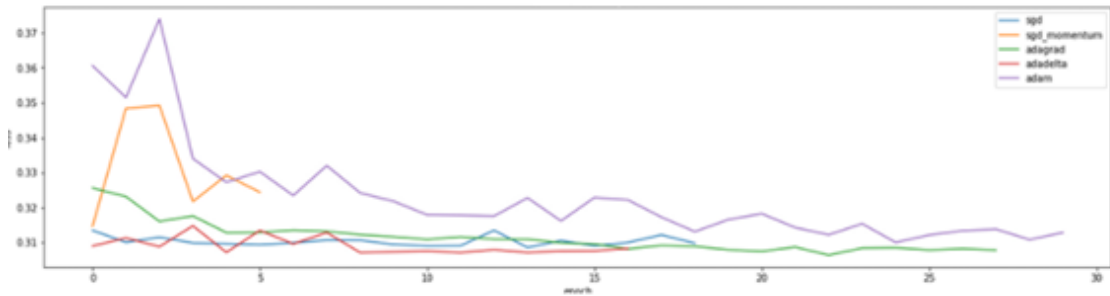


Şekil 4. 24 Geliştirilen Modelin Adımları

Modeli geliştirirken transfer öğrenimi için AlexNet, VGG, ResNet, GoogleNet, Inception ve MobileNet ağları ve bunların versiyonları eğitilerek bu ağların verdiği sonuçlar incelenmiştir.

4.2.1 Optimizasyon Algoritması Seçimi

Daha sonra optimizasyon algoritması seçimine geçilerek model için en hızlı şekilde kayıp değerini düşürecekimiz algoritmalar denenmiştir. Bu algoritmalar Adagrad, Adadelta, Adam, Sgd ve Sgd_Momentum dur.



Şekil 4. 25 Optimizasyon Algoritmaları Karşılaştırması

Yapılan karşılaştırmada algoritmalar arasında çok fazla fark olmasa da veri sayısı çok olan durumlarda en iyi performansı gösterdiği için optimizasyon algoritması olarak Adam seçilmiştir.

4.2.2 Öğrenme Hızı ve Yığın Boyutu

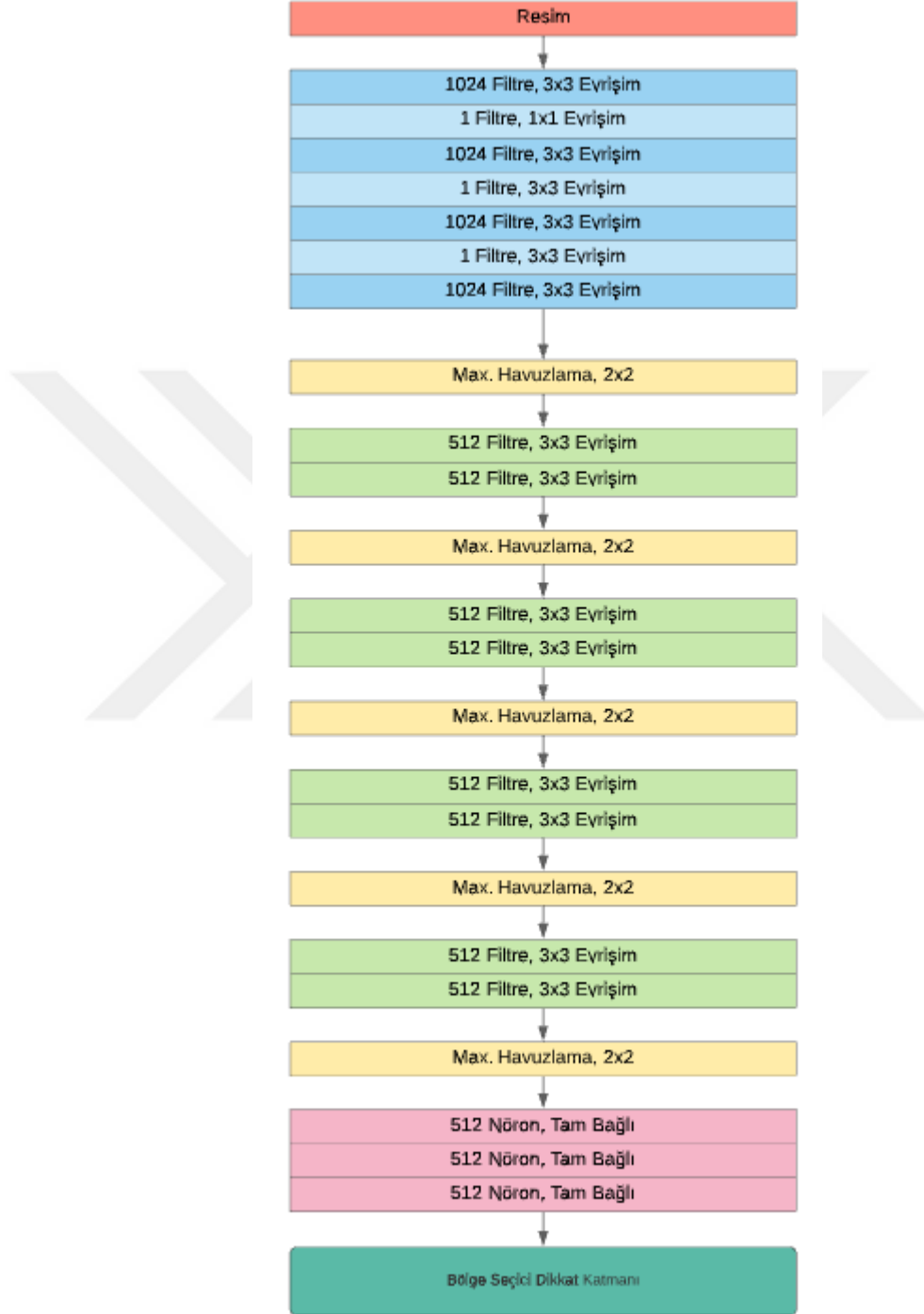
Yaptığımız denemelerde yığın boyutu olarak 1024 ve öğrenme katsayısını 0.05 seçerek çalıştığımızda daha iyi sonuçlar elde ettiğimizi gördük.

4.2.3 Giriş Resminin Boyutu

Modelimiz için verilen resimlerin boyutunu ayarlamak için 64, 128, 256, 512 ve 1024 gibi çözünürlükler için denemeler yapılmıştır. Elde ettiğimiz sonuçlar neticesinde en iyi değerleri 512x512 çözünürlüğü için elde ettiğimizden giriş resmi boyutu olarak bu değeri kullandık. Ayrıca giriş resmimizi 3 kanallı RGB renkli resim olarak aldık.

4.2.4 Geliştirilen Model Yapısı

Hazırladığımız modelin yapısı aşağıdaki şekilde verilmiştir. Kullanılan bu ilk modele ESA 1 adı verilmiştir.



Şekil 4. 26 Geliştirilen Model Yapısı

Bu ESA modelinde yapılan işlemler şu şekildedir,

1. Veri setinde uygun olmayan veriler temizlenerek sınıflar arası dengesizlik giderilir.
2. Verilen giriş resminin boyutları daha önce hesaplanarak bulunan optimum değere ayarlanır. (512, 512, 3)
3. Verilen giriş resmi 3x3 boyutunda 1024 filtre ile “aynı” dolgulama değeri ile evrişim işlemine tabi tutulur. Bu işlem 4 kere tekrarlanır.
4. 1024 filtre bir sonraki katmana bağlanmadan önce 1 adet 1x1’lik bir evrişim işlemi ile hem boyut düşürülmesi sağlanır hem de tam bağlı bir katman gibi görev alarak iki katman arasındaki tüm hücrelerin birbiri ile etkileşmesi sağlanmış olur. Bu işlem üç kere tekrarlanır.
5. Daha sonra 2x2 boyutlu filtre ile en büyük havuzlama işlemine tabi tutulur.
6. Elde edilen çıktı iki kere 3x3 boyutunda 512 adet filtre ile evrişim işlemine sokulur.
7. Bu çıktıya yeniden 2x2 boyutlu filtre en büyük havuzlama işlemi uygulanır.
8. 5 ve 6. Adımlar aynı şekilde dört kere tekrarlanır.
9. Son olarak evrişim işleminden elde edilen çıktı 512 adet 3 katmanlı tam bağlı sinir ağına aktarılır.
10. Sinir ağından gelen çıktılar bölge seçici dikkat katmanı adını verdiğimiz bir ESA yapısından geçirilerek sistemin resimle ilgili hastalıklı bölgelere yoğunlaşması sağlanır.
11. Aktivasyon fonksiyonu ReLU, optimizasyon algoritması ise Adam seçilmiştir.
12. Kayıp fonksiyonu hesaplamasında Categorical Cross Entropy kullanılmıştır.
13. Sınıflandırma için Softmax fonksiyonu kullanılmıştır.

4.2.5 Adapte Edilmiş MobileNet Yapısı

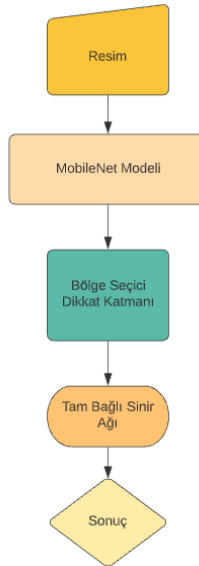
Çalıştığımız modeller içersin de parametre sayısının azlığı, taşınabilir olması ve diğer modellere göre de iyi sonuçlar vermesi açısından temel model olarak MobileNet seçilmiştir. Bu modele “Adapte Edilmiş MobileNet Evrişimli Sinir Ağı” (AMESA) ismi verilmiştir.

Aşağıda kullanılan modelin mimari yapısı görülmektedir [64],

Type / Stride	Filter Shape	Input Size
Conv / s2	$3 \times 3 \times 3 \times 32$	$224 \times 224 \times 3$
Conv dw / s1	$3 \times 3 \times 32$ dw	$112 \times 112 \times 32$
Conv / s1	$1 \times 1 \times 32 \times 64$	$112 \times 112 \times 32$
Conv dw / s2	$3 \times 3 \times 64$ dw	$112 \times 112 \times 64$
Conv / s1	$1 \times 1 \times 64 \times 128$	$56 \times 56 \times 64$
Conv dw / s1	$3 \times 3 \times 128$ dw	$56 \times 56 \times 128$
Conv / s1	$1 \times 1 \times 128 \times 128$	$56 \times 56 \times 128$
Conv dw / s2	$3 \times 3 \times 128$ dw	$56 \times 56 \times 128$
Conv / s1	$1 \times 1 \times 128 \times 256$	$28 \times 28 \times 128$
Conv dw / s1	$3 \times 3 \times 256$ dw	$28 \times 28 \times 256$
Conv / s1	$1 \times 1 \times 256 \times 256$	$28 \times 28 \times 256$
Conv dw / s2	$3 \times 3 \times 256$ dw	$28 \times 28 \times 256$
Conv / s1	$1 \times 1 \times 256 \times 512$	$14 \times 14 \times 256$
5× Conv dw / s1	$3 \times 3 \times 512$ dw	$14 \times 14 \times 512$
Conv / s1	$1 \times 1 \times 512 \times 512$	$14 \times 14 \times 512$
Conv dw / s2	$3 \times 3 \times 512$ dw	$14 \times 14 \times 512$
Conv / s1	$1 \times 1 \times 512 \times 1024$	$7 \times 7 \times 512$
Conv dw / s2	$3 \times 3 \times 1024$ dw	$7 \times 7 \times 1024$
Conv / s1	$1 \times 1 \times 1024 \times 1024$	$7 \times 7 \times 1024$
Avg Pool / s1	Pool 7×7	$7 \times 7 \times 1024$
FC / s1	1024×1000	$1 \times 1 \times 1024$
Softmax / s1	Classifier	$1 \times 1 \times 1000$

Şekil 4. 27 MobileNet Mimarisi [64]

Bu yapıya tam bağlı bir sinir ağı ekleyerek mevcut modele seçtiğimiz parametreler ile adapte ettik.



Şekil 4. 28 Adapte Edilmiş MobileNet Gösterimi

Burada yapılan işlemler şu şekildedir,

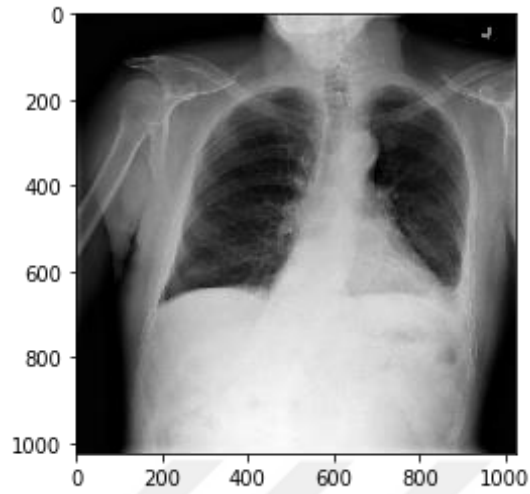
1. Giriş resmi daha önce optimum olarak bulunmuş değere göre ayarlanarak algoritmaya verilir.
2. Giriş resim boyutları (512, 512, 3) şeklindedir.
3. Öğrenme oranı 0.0005 seçilmiştir.
4. Ara katmanlarda aktivasyon fonksiyonu olarak ReLu kullanılmıştır. Optimizasyon algoritması olarak ta Adam seçilmiştir.
5. Kayıp fonksiyonu olarak Binary Cross Entropy kullanılmıştır.
6. Son katmanda ikili sınıflandırma için Sigmoid fonksiyonu kullanılmıştır.
7. MobileNet çıkış katmanında elde edilen değerler yeni sınıf yapısına göre yeniden ayarlanarak parametreler güncellenir.
8. Ana modelden gelen çıktılar bölge seçici dikkat katmanı adını verdiğimiz bir ESA yapısından geçirilerek sistemin resimle ilgili hastalıklı bölgelere yoğunlaşması sağlanır.
9. Bu yapıda MobileNet'e ait bir "Imagenet" gibi başlangıç ağırlığı kullanılmamıştır.

4.3 Geliştirilen Modelin Sonuçları

Şimdi geliştirilen modellerin her bir hastalık için bulduğu sonuçları diğer modellerin sonuçlarıyla karşılaştırarak detaylı olarak inceleyelim.

4.3.1 Atelektazi

Atelektazi için bulduğumuz sonuçlar şöyledir,



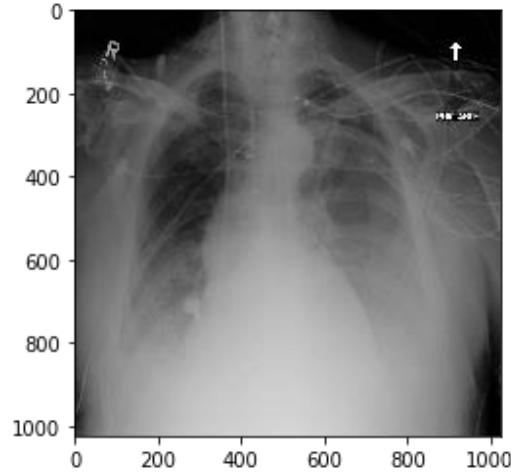
Şekil 4. 29 Atelektazi Tanısı İçin Röntgen Görüntüsü [65]

Tablo 4. 1 Atelektazi Sınıfı İçin AUC Değer Karşılaştırması

Çalışma	Model	Sonuç
Tez Çalışmamız	ESA 1	0.779
Tez Çalışmamız	AMESA	0.760
Wang ve ark. [63]	R-50	0.700
Gündel ve ark. [23]	D-121	0.767
Yao ve ark. [66]	-	0.733
Li ve ark. [67]	R-50	0.727
Li ve ark. [67]	D-121	0.728
Shen ve ark. [68]	-	0.766
Tang ve ark. [69]	-	0.756
Guan ve ark.[22]	CRAL (att1) R-50	0.777
Guan ve ark.[22]	CRAL (att2) R-50	0.779
Guan ve ark.[22]	CRAL (att1) D-121	0.781
Guan ve ark.[22]	CRAL (att2) D-121	0.781

Tez çalışmamız haricinde yapılan 11 adet modelin ortalama sonucu bu sınıf için 0.754 tür. Atelektazi sınıfında ESA 1 modeli için elde edilen 0.779 değeri ve AMESA modeli için bulunan 0.76 değeri ortalamanın üzerindedir.

4.3.2 Kardiyomegali



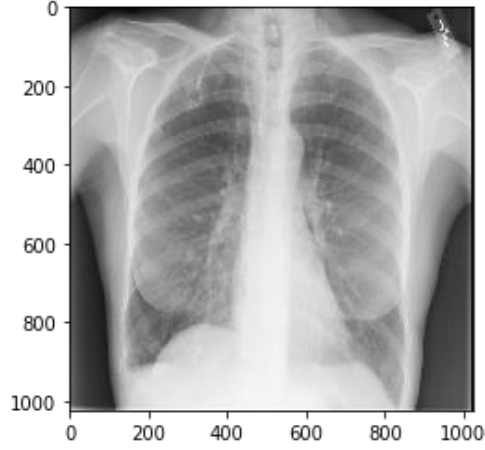
Şekil 4. 30 Kardiyomegali Tanısı İçin Röntgen Görüntüsü [65]

Tablo 4. 2 Kardiyomegali Sınıfı İçin AUC Değer Karşılaştırması

Çalışma	Model	Sonuç
Tez Çalışmamız	ESA 1	0.920
Tez Çalışmamız	AMESA	0.930
Wang ve ark. [63]	R-50	0.810
Gündel ve ark. [23]	D-121	0.883
Yao ve ark. [66]	-	0.856
Li ve ark. [67]	R-50	0.836
Li ve ark. [67]	D-121	0.848
Shen ve ark. [68]	-	0.801
Tang ve ark. [69]	-	0.887
Guan ve ark.[22]	CRAL (att1) R-50	0.879
Guan ve ark.[22]	CRAL (att2) R-50	0.875
Guan ve ark.[22]	CRAL (att1) D-121	0.883
Guan ve ark.[22]	CRAL (att2) D-121	0.880

Tez çalışmamız haricinde yapılan 11 adet modelin ortalama sonucu bu sınıf için 0.858 dir. Kardiyomegali sınıfında ESA 1 modeli için elde edilen 0,92 değeri ve AMESA modeli için bulunan 0,93 değeri ortalamanın üzerindedir.

4.3.3 Efüzyon



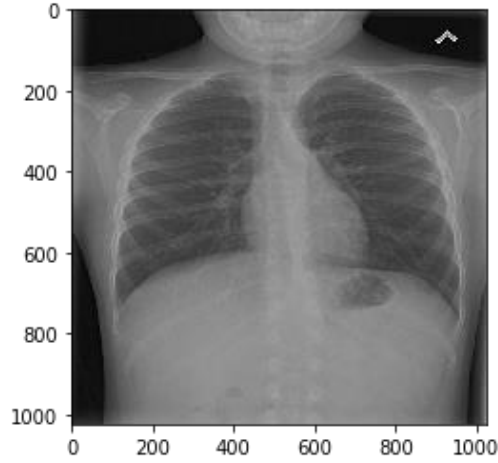
Şekil 4. 31 Efüzyon Tanısı İçin Röntgen Görüntüsü [65]

Tablo 4. 3 Efüzyon Sınıfı İçin AUC Değer Karşılaştırması

Çalışma	Model	Sonuç
Tez Çalışmamız	ESA 1	0.896
Tez Çalışmamız	AMESA	0.850
Wang ve ark. [63]	R-50	0.759
Gündel ve ark. [23]	D-121	0.828
Yao ve ark. [66]	-	0.806
Li ve ark. [67]	R-50	0.789
Li ve ark. [67]	D-121	0.782
Shen ve ark. [68]	-	0.797
Tang ve ark. [69]	-	0.819
Guan ve ark.[22]	CRAL (att1) R-50	0.824
Guan ve ark.[22]	CRAL (att2) R-50	0.826
Guan ve ark.[22]	CRAL (att1) D-121	0.831
Guan ve ark.[22]	CRAL (att2) D-121	0.829

Tez çalışmamız haricinde yapılan 11 adet modelin ortalama sonucu bu sınıf için 0.808 dir. Efüzyon sınıfında ESA 1 modeli için elde edilen 0,896 değeri ve AMESA modeli için bulunan 0,850 değeri ortalamanın üzerindedir.

4.3.4 İnfiltrasyon (Infiltration)



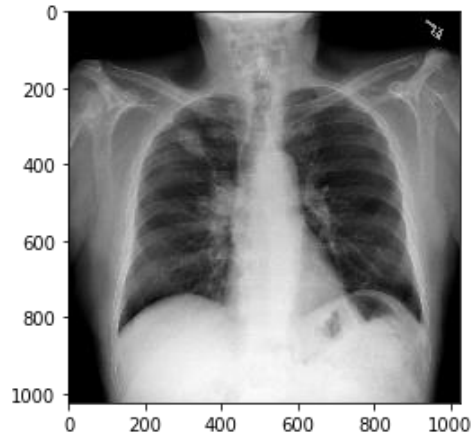
Şekil 4. 32 İnfiltrasyon Tanısı İçin Röntgen Görüntüsü [65]

Tablo 4. 4 İnfiltrasyon Sınıfı İçin AUC Değer Karşılaştırması

Çalışma	Model	Sonuç
Tez Çalışmamız	ESA 1	0.712
Tez Çalışmamız	AMESA	0.680
Wang ve ark. [63]	R-50	0.661
Gündel ve ark. [23]	D-121	0.709
Yao ve ark. [66]	-	0.673
Li ve ark. [67]	R-50	0.672
Li ve ark. [67]	D-121	0.645
Shen ve ark. [68]	-	0.751
Tang ve ark. [69]	-	0.689
Guan ve ark.[22]	CRAL (att1) R-50	0.694
Guan ve ark.[22]	CRAL (att2) R-50	0.695
Guan ve ark.[22]	CRAL (att1) D-121	0.697
Guan ve ark.[22]	CRAL (att2) D-121	0.702

Tez çalışmamız haricinde yapılan 11 adet modelin ortalama sonucu bu sınıf için 0.689 dur. İnfiltrasyon sınıfında ESA 1 modeli için elde edilen 0,712 değeri ortalamanın üzerindeyken AMESA modeli için bulunan 0.680 değeri ortalamanın altında kalmıştır.

4.3.5 Kütle (Mass)



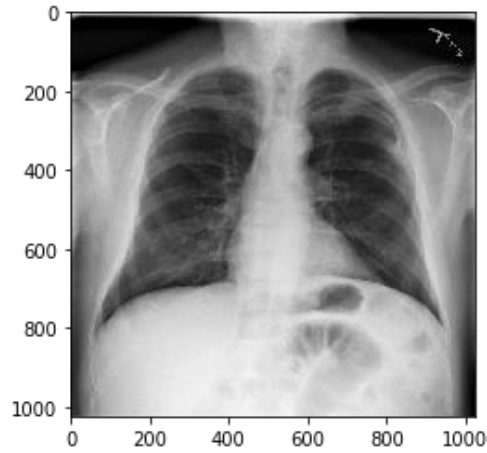
Şekil 4. 33 Kütle Tanısı İçin Röntgen Görüntüsü

Tablo 4. 5 Kütle Sınıfı İçin AUC Değer Karşılaştırması

Çalışma	Model	Sonuç
Tez Çalışmamız	ESA 1	0.802
Tez Çalışmamız	AMESA	0.790
Wang ve ark. [63]	R-50	0.693
Gündel ve ark. [23]	D-121	0.821
Yao ve ark. [66]	-	0.718
Li ve ark. [67]	R-50	0.776
Li ve ark. [67]	D-121	0.747
Shen ve ark. [68]	-	0.760
Tang ve ark. [69]	-	0.814
Guan ve ark.[22]	CRAL (att1) R-50	0.831
Guan ve ark.[22]	CRAL (att2) R-50	0.825
Guan ve ark.[22]	CRAL (att1) D-121	0.830
Guan ve ark.[22]	CRAL (att2) D-121	0.834

Tez çalışmamız haricinde yapılan 11 adet modelin ortalama sonucu bu sınıf için 0.786 dır. Kütle sınıfında ESA 1 modeli için elde edilen 0,802 değeri ve AMESA modeli için bulunan 0.790 değeri ortalamanın üzerindedir.

4.3.6 Nodül (Nodule)



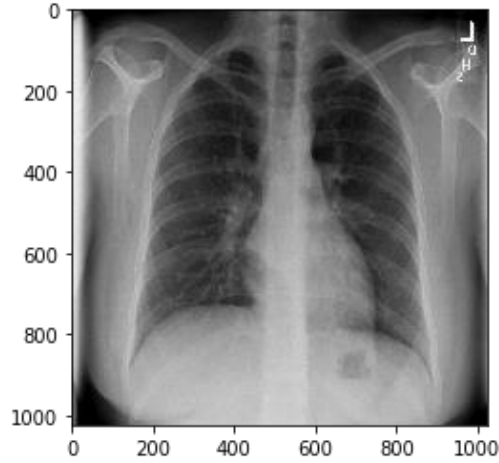
Şekil 4. 34 Nodül Tanısı İçin Röntgen Görüntüsü [65]

Tablo 4. 6 Nodül Sınıfı İçin AUC Değer Karşılaştırması

Çalışma	Model	Sonuç
Tez Çalışmamız	ESA 1	0.745
Tez Çalışmamız	AMESA	0.670
Wang ve ark. [63]	R-50	0.669
Gündel ve ark. [23]	D-121	0.758
Yao ve ark. [66]	-	0.777
Li ve ark. [67]	R-50	0.696
Li ve ark. [67]	D-121	0.702
Shen ve ark. [68]	-	0.741
Tang ve ark. [69]	-	0.755
Guan ve ark.[22]	CRAL (att1) R-50	0.766
Guan ve ark.[22]	CRAL (att2) R-50	0.765
Guan ve ark.[22]	CRAL (att1) D-121	0.764
Guan ve ark.[22]	CRAL (att2) D-121	0.773

Tez çalışmamız haricinde yapılan 11 adet modelin ortalama sonucu bu sınıf için 0.742 dur. Nodül sınıfında ESA 1 modeli için elde edilen 0,745 değeri ortalamanın üzerindeyken AMESA modeli için bulunan 0.670 değeri ortalamanın altında kalmıştır.

4.3.7 Zatürre (Pneumonia)



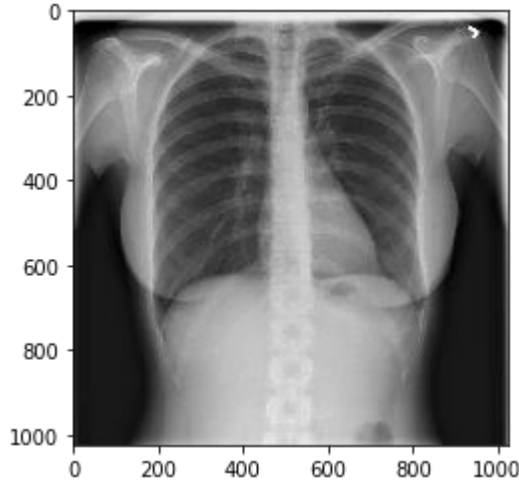
Şekil 4. 35 Zatürre Tanısı İçin Röntgen Görüntüsü [65]

Tablo 4. 7 Zatürre Sınıfı İçin AUC Değer Karşılaştırması

Çalışma	Model	Sonuç
Tez Çalışmamız	ESA 1	0.713
Tez Çalışmamız	AMESA	0.705
Wang ve ark. [63]	R-50	0.658
Gündel ve ark. [23]	D-121	0.731
Yao ve ark. [66]	-	0.684
Li ve ark. [67]	R-50	0.649
Li ve ark. [67]	D-121	0.632
Shen ve ark. [68]	-	0.778
Tang ve ark. [69]	-	0.729
Guan ve ark.[22]	CRAL (att1) R-50	0.726
Guan ve ark.[22]	CRAL (att2) R-50	0.720
Guan ve ark.[22]	CRAL (att1) D-121	0.725
Guan ve ark.[22]	CRAL (att2) D-121	0.729

Tez çalışmamız haricinde yapılan 11 adet modelin ortalama sonucu bu sınıf için 0.706 dır. Zatürre sınıfında ESA 1 modeli için elde edilen 0,713 değeri ortalamanın üzerindeyken AMESA modeli için bulunan 0.705 değeri ortalamanın altında kalmıştır.

4.3.8 Pnömotoraks (Pneumothorax)



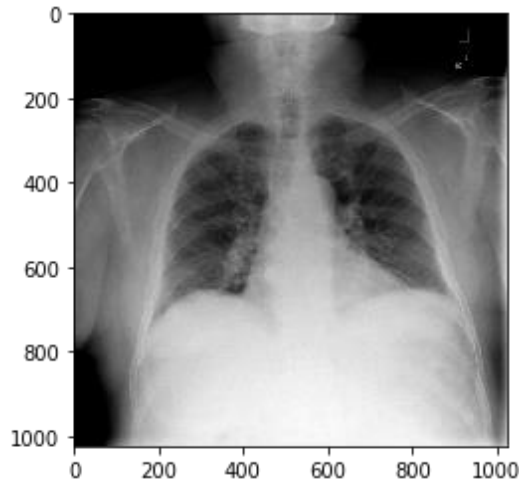
Şekil 4. 36 Pnömotoraks Tanısı İçin Röntgen Görüntüsü [65]

Tablo 4. 8 Pnömotoraks Sınıfı İçin AUC Değer Karşılaştırması

Çalışma	Model	Sonuç
Tez Çalışmamız	ESA 1	0.847
Tez Çalışmamız	AMESA	0.870
Wang ve ark. [63]	R-50	0.799
Gündel ve ark. [23]	D-121	0.846
Yao ve ark. [66]	-	0.805
Li ve ark. [67]	R-50	0.808
Li ve ark. [67]	D-121	0.802
Shen ve ark. [68]	-	0.800
Tang ve ark. [69]	-	0.850
Guan ve ark.[22]	CRAL (att1) R-50	0.858
Guan ve ark.[22]	CRAL (att2) R-50	0.852
Guan ve ark.[22]	CRAL (att1) D-121	0.866
Guan ve ark.[22]	CRAL (att2) D-121	0.857

Tez çalışmamız haricinde yapılan 11 adet modelin ortalama sonucu bu sınıf için 0.831 dir. Pnömotoraks sınıfında ESA 1 modeli için elde edilen 0,847 değeri ve AMESA modeli için bulunan 0.870 değeri ortalamanın üzerindedir.

4.3.9 Konsolidasyon (Consolidation)



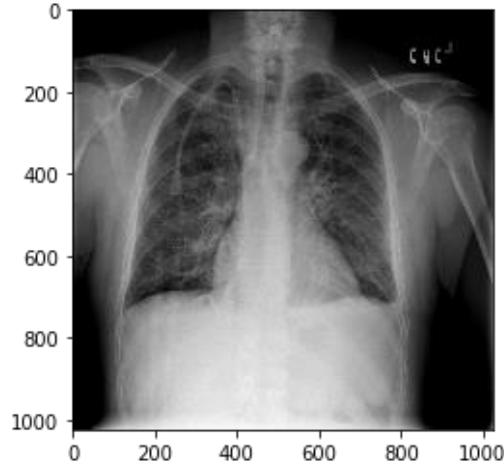
Şekil 4. 37 Konsolidasyon Tanısı İçin Röntgen Görüntüsü [65]

Tablo 4. 9 Konsolidasyon Sınıfı İçin AUC Değer Karşılaştırması

Çalışma	Model	Sonuç
Tez Çalışmamız	ESA 1	0.721
Tez Çalışmamız	AMESA	0.700
Wang ve ark. [63]	R-50	0.703
Gündel ve ark. [23]	D-121	0.745
Yao ve ark. [66]	-	0.711
Li ve ark. [67]	R-50	0.720
Li ve ark. [67]	D-121	0.727
Shen ve ark. [68]	-	0.787
Tang ve ark. [69]	-	0.728
Guan ve ark.[22]	CRAL (att1) R-50	0.758
Guan ve ark.[22]	CRAL (att2) R-50	0.751
Guan ve ark.[22]	CRAL (att1) D-121	0.758
Guan ve ark.[22]	CRAL (att2) D-121	0.754

Tez çalışmamız haricinde yapılan 11 adet modelin ortalama sonucu bu sınıf için 0.740 tır. Konsolidasyon sınıfında ESA 1 modeli için elde edilen 0,721 değeri ve AMESA modeli için bulunan 0.700 değeri ortalamanın altında kalmıştır.

4.3.10 Ödem (Edema)



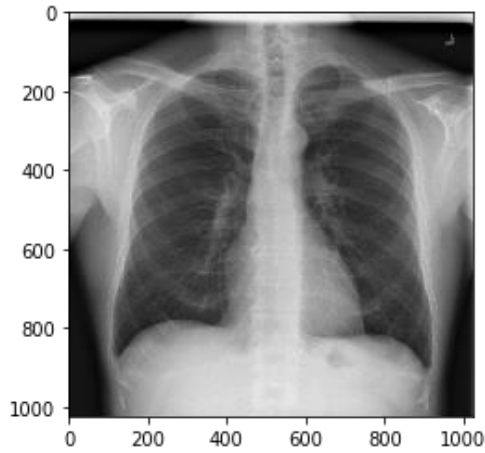
Şekil 4. 38 Ödem Tanısı İçin Röntgen Görüntüsü

Tablo 4. 10 Ödem Sınıfı İçin AUC Değer Karşılaştırması

Çalışma	Model	Sonuç
Tez Çalışmamız	ESA 1	0.875
Tez Çalışmamız	AMESA	0.890
Wang ve ark. [63]	R-50	0.805
Gündel ve ark. [23]	D-121	0.835
Yao ve ark. [66]	-	0.806
Li ve ark. [67]	R-50	0.806
Li ve ark. [67]	D-121	0.823
Shen ve ark. [68]	-	0.820
Tang ve ark. [69]	-	0.848
Guan ve ark.[22]	CRAL (att1) R-50	0.850
Guan ve ark.[22]	CRAL (att2) R-50	0.848
Guan ve ark.[22]	CRAL (att1) D-121	0.853
Guan ve ark.[22]	CRAL (att2) D-121	0.850

Tez çalışmamız haricinde yapılan 11 adet modelin ortalama sonucu bu sınıf için 0.831 tir. Ödem sınıfında ESA 1 modeli için elde edilen 0,875 değeri ve AMESA modeli için bulunan 0.890 değeri ortalamanın üzerindedir.

4.3.11 Amfizem (Emphysema)



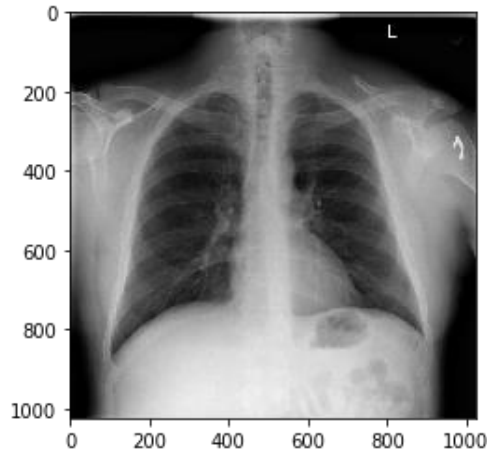
Şekil 4. 39 Amfizem Tanısı İçin Röntgen Görüntüsü

Tablo 4. 11 Amfizem Sınıfı İçin AUC Değer Karşılaştırması

Çalışma	Model	Sonuç
Tez Çalışmamız	ESA 1	0.770
Tez Çalışmamız	AMESA	0.830
Wang ve ark. [63]	R-50	0.833
Gündel ve ark. [23]	D-121	0.895
Yao ve ark. [66]	-	0.842
Li ve ark. [67]	R-50	0.888
Li ve ark. [67]	D-121	0.757
Shen ve ark. [68]	-	0.773
Tang ve ark. [69]	-	0.906
Guan ve ark.[22]	CRAL (att1) R-50	0.909
Guan ve ark.[22]	CRAL (att2) R-50	0.905
Guan ve ark.[22]	CRAL (att1) D-121	0.911
Guan ve ark.[22]	CRAL (att2) D-121	0.908

Tez çalışmamız haricinde yapılan 11 adet modelin ortalama sonucu bu sınıf için 0.867 dir. Amfizem sınıfında ESA 1 modeli için elde edilen 0,770 değeri ve AMESA modeli için bulunan 0.830 değeri ortalamanın altında kalmıştır.

4.3.12 Fibroz (Fibrosis)



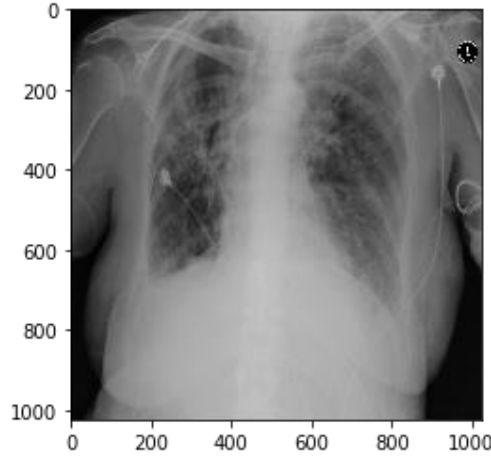
Şekil 4. 40 Fibroz Tanısı İçin Röntgen Görüntüsü

Tablo 4. 12 Fibroz Sınıfı İçin AUC Değer Karşılaştırması

Çalışma	Model	Sonuç
Tez Çalışmamız	ESA 1	0.798
Tez Çalışmamız	AMESA	0.800
Wang ve ark. [63]	R-50	0.786
Gündel ve ark. [23]	D-121	0.818
Yao ve ark. [66]	-	0.743
Li ve ark. [67]	R-50	0.771
Li ve ark. [67]	D-121	0.763
Shen ve ark. [68]	-	0.765
Tang ve ark. [69]	-	0.818
Guan ve ark.[22]	CRAL (att1) R-50	0.832
Guan ve ark.[22]	CRAL (att2) R-50	0.819
Guan ve ark.[22]	CRAL (att1) D-121	0.826
Guan ve ark.[22]	CRAL (att2) D-121	0.830

Tez çalışmamız haricinde yapılan 11 adet modelin ortalama sonucu bu sınıf için 0.797 dir. Fibroz sınıfında ESA 1 modeli için elde edilen 0,798 değeri ve AMESA modeli için bulunan 0.800 değeri ortalamanın üzerindedir.

4.3.13 Plevral Kalınlaşma (Pleural Thickening)



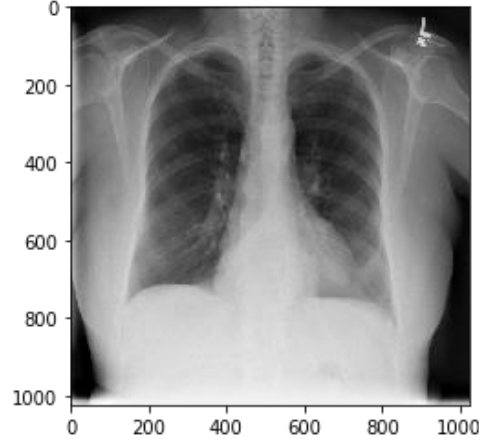
Şekil 4. 41 Plevral Kalınlaşma Tanısı İçin Röntgen Görüntüsü

Tablo 4. 13 Plevral Kalınlaşma Sınıfı İçin AUC Değer Karşılaştırması

Çalışma	Model	Sonuç
Tez Çalışmamız	ESA 1	0.778
Tez Çalışmamız	AMESA	0.680
Wang ve ark. [63]	R-50	0.684
Gündel ve ark. [23]	D-121	0.761
Yao ve ark. [66]	-	0.724
Li ve ark. [67]	R-50	0.737
Li ve ark. [67]	D-121	0.735
Shen ve ark. [68]	-	0.759
Tang ve ark. [69]	-	0.765
Guan ve ark.[22]	CRAL (att1) R-50	0.778
Guan ve ark.[22]	CRAL (att2) R-50	0.777
Guan ve ark.[22]	CRAL (att1) D-121	0.780
Guan ve ark.[22]	CRAL (att2) D-121	0.778

Tez çalışmamız haricinde yapılan 11 adet modelin ortalama sonucu bu sınıf için 0.752 dir. Plevral Kalınlaşma sınıfında ESA 1 modeli için elde edilen 0,778 değeri ortalamanın üzerindeyken AMESA modeli için bulunan 0.680 değeri ortalamanın altında kalmıştır.

4.3.14 Fıtık (Hernia)



Şekil 4. 42 Fıtık Tanısı İçin Röntgen Görüntüsü

Tablo 4. 14 Fıtık Sınıfı İçin AUC Değer Karşılaştırması

Çalışma	Model	Sonuç
Tez Çalışmamız	ESA 1	0.650
Tez Çalışmamız	AMESA	0.690
Wang ve ark. [63]	R-50	0.872
Gündel ve ark. [23]	D-121	0.896
Yao ve ark. [66]	-	0.775
Li ve ark. [67]	R-50	0.693
Li ve ark. [67]	D-121	0.653
Shen ve ark. [68]	-	0.748
Tang ve ark. [69]	-	0.875
Guan ve ark.[22]	CRAL (att1) R-50	0.906
Guan ve ark.[22]	CRAL (att2) R-50	0.908
Guan ve ark.[22]	CRAL (att1) D-121	0.918
Guan ve ark.[22]	CRAL (att2) D-121	0.917

Tez çalışmamız haricinde yapılan 11 adet modelin ortalama sonucu bu sınıf için 0.833 tür. Fıtık sınıfında ESA 1 modeli için elde edilen 0,650 değeri ve AMESA modeli için bulunan 0.690 değeri ortalamanın altında kalmıştır.

Elde edilen sonuçların toplu gösterimi şöyledir.

Tablo 4. 15 Tüm Sınıflar İçin AUC Değer Karşılaştırması 1

Çalışma	Atelektazi	Kardiyomegali	Efüzyon	İnfiltrasyon	Kütle
Tez Çalışmamız	0.779	0.92	0.896	0.712	0.802
Tez Çalışmamız	0.76	0.93	0.85	0.68	0.79
Wang ve ark. [61]	0.7	0.81	0.759	0.661	0.693
Gündel ve ark. [64]	0.767	0.883	0.828	0.709	0.821
Yao ve ark. [65]	0.733	0.856	0.806	0.673	0.718
Li ve ark. [66]	0.727	0.836	0.789	0.672	0.776
Li ve ark. [66]	0.728	0.848	0.782	0.645	0.747
Shen ve ark. [67]	0.766	0.801	0.797	0.751	0.76
Tang ve ark. [68]	0.756	0.887	0.819	0.689	0.814
Guan ve ark. [69]	0.777	0.879	0.824	0.694	0.831
Guan ve ark. [69]	0.779	0.875	0.826	0.695	0.825
Guan ve ark. [69]	0.781	0.883	0.831	0.697	0.83
Guan ve ark. [69]	0.781	0.88	0.829	0.702	0.834

Tablo 4. 16 Tablo 4. 16 Tüm Sınıflar İçin AUC Değer Karşılaştırması 2

Çalışma	Nodül	Zatürre	Pnömotoraks	Konsolidasyon
Tez Çalışmamız	0.745	0.713	0.847	0.721
Tez Çalışmamız	0.67	0.705	0.87	0.7
Wang ve ark. [61]	0.669	0.658	0.799	0.703
Gündel ve ark. [64]	0.758	0.731	0.846	0.745
Yao ve ark. [65]	0.777	0.684	0.805	0.711
Li ve ark. [66]	0.696	0.649	0.808	0.72
Li ve ark. [66]	0.702	0.632	0.802	0.727
Shen ve ark. [67]	0.741	0.778	0.8	0.787
Tang ve ark. [68]	0.755	0.729	0.85	0.728
Guan ve ark. [69]	0.766	0.726	0.858	0.758
Guan ve ark. [69]	0.765	0.72	0.852	0.751
Guan ve ark. [69]	0.764	0.725	0.866	0.758
Guan ve ark. [69]	0.773	0.729	0.857	0.754

Tablo 4. 17 Tablo 4. 16 Tüm Sınıflar İçin AUC Değer Karşılaştırması 3

Çalışma	Ödem	Amfizem	Fibroz	Plevral Kalınlaşma	Fıtık
Tez Çalışmamız	0.875	0.77	0.798	0.778	0.65
Tez Çalışmamız	0.89	0.83	0.8	0.68	0.69
Wang ve ark. [61]	0.805	0.833	0.786	0.684	0.872
Gündel ve ark. [64]	0.835	0.895	0.818	0.761	0.896
Yao ve ark. [65]	0.806	0.842	0.743	0.724	0.775
Li ve ark. [66]	0.806	0.888	0.771	0.737	0.693
Li ve ark. [66]	0.823	0.757	0.763	0.735	0.653
Shen ve ark. [67]	0.82	0.773	0.765	0.759	0.748
Tang ve ark. [68]	0.848	0.906	0.818	0.765	0.875
Guan ve ark. [69]	0.85	0.909	0.832	0.778	0.906
Guan ve ark. [69]	0.848	0.905	0.819	0.777	0.908
Guan ve ark. [69]	0.853	0.911	0.826	0.78	0.918
Guan ve ark. [69]	0.85	0.908	0.83	0.778	0.917

Karşılaştırma tablolarından da görüldüğü üzere modelimiz elimizdeki kısıtlı imkânlara rağmen başarılı sonuçlar elde etmiştir. Ayrıca modelimiz de kullandığımız algoritmalarından MobileNet temelli AMESA bellek kullanımı ve parametre sayısı açısından da oldukça başarılıdır.

5.1 Sonuçların Değerlendirilmesi

Yapılan bu çalışmada Amerika Birleşik Devletleri Ulusal Sağlık Enstitüsü (NIH) [62] tarafından yayınlanan ve 30805 farklı hastanın 112120 adet önden çekilmiş etiketli göğüs röntgen görüntülerini içeren veri seti kullanılmıştır. Kullanılan bu veri setinde öne çıkan ve en çok rastlanan 14 hastalık Atelektazi, Kardiyomegali, Konsolidasyon, Ödem, Efüzyon, Amfizem, Fibroz, İltihaplanma, Kütle, Nodül, Plevral Kalınlaşma, Zatürre, Pnömotoraks ve Fıtık değerlendirilmiştir. Öncelikle veri setindeki dengesizlikler giderilmiştir. Daha sonra transfer öğrenmesi yöntemi ile geliştirmek istediğimiz ilk model için farklı veri setleri üzerinde eğitilerek geliştirilmiş olan ResNet, VGG, Inception, Xception, DenseNet, EfficientNet ve MobileNet gibi ağlar değerlendirilerek parametre sayısı, taşınabilirlik (mobilité) ve model başarısı açısından en uygun olan modelin MobileNet olduğu tespit edilmiştir. Temel model seçildikten sonra veri seti üzerinde iyileştirme çalışmaları yapılmış ve çalışılan sınıflar için veri dengesizliği giderilmiştir. Sonrasında resim boyutu, öğrenme hızı, optimizasyon algoritması ve yığın boyutu gibi parametreler için uygun değer belirleme çalışmaları yapılmıştır. Uygun parametre çalışmaları tamamlandıktan sonra ilk olarak transfer öğrenimi yoluyla geliştirilen AMESA modeli için eğitim çalışmaları yapılarak uygun ağırlık oranları elde edilmiştir. Daha sonra çeşitli ESA modelleriyle çalışmalar yapılarak geliştirilen ESA 1 modeli için eğitim çalışmaları yapılarak uygun ağırlık oranları elde edilmeye çalışılmıştır.

AMESA modelinde çalıştığımız sınıflar için elde edilen AUC değerleri şöyledir, Atelektazi 0.76, Kardiyomegali 0.93, Konsolidasyon 0.70, Ödem 0.89, Efüzyon 0.85, Amfizem 0.83, Fibroz 0.80, İnfiltrasyon 0.68, Kütle 0.79, Nodül 0.67, Plevral Kalınlaşma 0.68, Zatürre 0.705, Pnömotoraks 0.870 ve Fıtık 0.69 dur.

ESA 1 modelinde çalıştığımız sınıflar için elde edilen AUC değerleri ise şöyledir, Atelektazi 0.779, Kardiyomegali 0.92, Konsolidasyon 0.721, Ödem 0.875, Efüzyon

0.896, Amfizem 0.77, Fibroz 0.798, İnfiltrasyon 0.712, Kütle 0.802, Nodül 0.745, Plevral Kalınlaşma 0.778, Zatürre 0.705, Pnömotoraks 0.847 ve Fıtık 0.65 tir.

ESA 1 modelinde elde edilen sonuçlar Atelektazi, Kardiyomegali, Efüzyon, İnfiltrasyon, Kütle, Nodül, Zatürre, Pnömotoraks, Ödem, Fibroz, Plevral Kalınlaşma, sınıfları için diğer modellerin ortalamasının üzerindeyken Konsolidasyon, Fıtık ve Amfizem sınıfları için ortalamasının altında kalmıştır.

AMESA modelinde elde edilen sonuçlar Atelektazi, Kardiyomegali, Efüzyon, Kütle, Ödem, Fibroz sınıfları için diğer modellerin ortalamasının üzerindeyken İnfiltrasyon, Nodül, Plevral Kalınlaşma, Zatürre, Konsolidasyon, Pnömotoraks, Fıtık ve Amfizem sınıfları için ortalamasının altında kalmıştır. Fakat Fıtık dışındaki sınıflarda alınan değerler ortalamaya oldukça yakın çıkmıştır.

ESA 1 modelinin başarısı yanında yüksek parametre sayısı ve çok uzun süren eğitimler süreleri göz önüne alındığında MobileNet temelli AMESA modelinin hızlı ve düşük parametrelili yapısı bu modeli daha ön plana çıkarmaktadır. Ayrıca AMESA modelinin düşük boyutlu yapısı bu algoritmayı taşınabilir cihazlara uygulama konusunda da oldukça elverişlidir.

Böylece Geliştirilen model ile yaygın göğüs hastalıklarının sınıflandırmasında kullanılabilir derecede iyi bir sınıflandırma başarısı elde edilmiştir. Bu sayede uzmanlar için hem bilgisayarlar hem de taşınabilir cihazlar için performanslı bir şekilde çalışan modeller geliştirmiş olduk.

Göğüs röntgenleri ile ilgili paylaşılacak veriler arttıkça modelimiz daha da çok örnek ile eğitilerek daha iyi sonuçlar verecektir.

- [1] “NIH Clinical Center provides one of the largest publicly available chest x-ray datasets to scientific community | National Institutes of Health (NIH).” <https://www.nih.gov/news-events/news-releases/nih-clinical-center-provides-one-largest-publicly-available-chest-x-ray-datasets-scientific-community> (accessed Dec. 17, 2020).
- [2] L. Brunese, F. Mercaldo, A. Reginelli, and A. Santone, “Explainable Deep Learning for Pulmonary Disease and Coronavirus COVID-19 Detection from X-rays,” *Comput. Methods Programs Biomed.*, vol. 196, p. 105608, 2020, doi: 10.1016/j.cmpb.2020.105608.
- [3] A. Gupta, Anjum, S. Gupta, and R. Katarya, “InstaCovNet-19: A deep learning classification model for the detection of COVID-19 patients using Chest X-ray,” *Appl. Soft Comput. J.*, vol. 99, p. 106859, 2020, doi: 10.1016/j.asoc.2020.106859.
- [4] H. Wang and Y. Xia, “ChestNet: A deep neural network for classification of thoracic diseases on chest radiography,” *arXiv*, pp. 1–8, 2018.
- [5] P. Rajpurkar *et al.*, “CheXNet: Radiologist-level pneumonia detection on chest X-rays with deep learning,” *arXiv*, pp. 3–9, 2017.
- [6] C. Liu *et al.*, “TX-CNN: Detecting tuberculosis in chest X-ray images using convolutional neural network,” *Proc. - Int. Conf. Image Process. ICIP*, vol. 2017-Septe, pp. 2314–2318, 2018, doi: 10.1109/ICIP.2017.8296695.
- [7] C. Yan, J. Yao, R. Li, Z. Xu, and J. Huang, “Weakly supervised deep learning for thoracic disease classification and localization on chest X-rays,” *arXiv*, pp. 103–110, 2018.
- [8] R. H. Abiyev and M. K. S. Ma’aitah, “Deep Convolutional Neural Networks for Chest Diseases Detection,” *J. Healthc. Eng.*, vol. 2018, 2018, doi: 10.1155/2018/4168538.
- [9] M. Moradi, A. Madani, A. Karargyris, and T. F. Syeda-Mahmood, “Chest x-ray generation and data augmentation for cardiovascular abnormality classification,” no. April 2020, p. 57, 2018, doi: 10.1117/12.2293971.
- [10] A. K. Jaiswal, P. Tiwari, S. Kumar, D. Gupta, A. Khanna, and J. J. P. C. Rodrigues, “Identifying pneumonia in chest X-rays: A deep learning approach,” *Meas. J. Int. Meas. Confed.*, vol. 145, pp. 511–518, 2019, doi: 10.1016/j.measurement.2019.05.076.
- [11] S. Asif, Y. Wenhui, H. Jin, Y. Tao, and S. Jinhai, “Classification of COVID-19 from Chest X-ray images using Deep Convolutional Neural Networks,” *medRxiv*, 2020, doi: 10.1101/2020.05.01.20088211.
- [12] S. Basu, S. Mitra, and N. Saha, “Deep learning for screening COVID-19 using chest X-ray images,” *medRxiv*, pp. 2521–2527, 2020, doi:

10.1101/2020.05.04.20090423.

- [13] S. H. Yoo *et al.*, “Deep Learning-Based Decision-Tree Classifier for COVID-19 Diagnosis From Chest X-ray Imaging,” *Frontiers in Medicine*, vol. 7, 2020, doi: 10.3389/fmed.2020.00427.
- [14] C. Ouchicha, O. Ammor, and M. Meknassi, “CVDNet: A novel deep learning architecture for detection of coronavirus (Covid-19) from chest x-ray images,” *Chaos, Solitons and Fractals*, vol. 140, 2020, doi: 10.1016/j.chaos.2020.110245.
- [15] A. Bhandary *et al.*, “Deep-learning framework to detect lung abnormality – A study with chest X-Ray and lung CT scan images,” *Pattern Recognit. Lett.*, vol. 129, pp. 271–278, 2020, doi: 10.1016/j.patrec.2019.11.013.
- [16] R. Jain, M. Gupta, S. Taneja, and D. J. Hemanth, “Deep learning based detection and analysis of COVID-19 on chest X-ray images,” *Appl. Intell.*, 2020, doi: 10.1007/s10489-020-01902-1.
- [17] V. Chouhan *et al.*, “A novel transfer learning based approach for pneumonia detection in chest X-ray images,” *Appl. Sci.*, vol. 10, no. 2, 2020, doi: 10.3390/app10020559.
- [18] A. Abbas, M. M. Abdelsamea, and M. M. Gaber, “Classification of COVID-19 in chest X-ray images using DeTraC deep convolutional neural network,” *medRxiv*, 2020, doi: 10.1101/2020.03.30.20047456.
- [19] T. Rahman *et al.*, “Reliable tuberculosis detection using chest x-ray with deep learning, segmentation and visualization,” *arXiv*, vol. 8, 2020, doi: 10.1109/access.2020.3031384.
- [20] D. Das, K. C. Santosh, and U. Pal, “Truncated inception net: COVID-19 outbreak screening using chest X-rays,” *Phys. Eng. Sci. Med.*, vol. 43, no. 3, pp. 915–925, 2020, doi: 10.1007/s13246-020-00888-x.
- [21] K. M. S. Palani, A. Kulkarni, A. Kochara, “Detection of Lung Diseases using Deep Learning,” *SSRN Electron. J.*, vol. 03024, pp. 1–7, 2020, doi: 10.2139/ssrn.3568730.
- [22] Q. Guan and Y. Huang, “Multi-label chest X-ray image classification via category-wise residual attention learning,” *Pattern Recognit. Lett.*, vol. 130, pp. 259–266, 2020, doi: 10.1016/j.patrec.2018.10.027.
- [23] S. Gündel, S. Grbic, B. Georgescu, S. Liu, A. Maier, and D. Comaniciu, “Learning to Recognize Abnormalities in Chest X-Rays with Location-Aware Dense Networks,” *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 11401 LNCS, pp. 757–765, Nov. 2018, doi: 10.1007/978-3-030-13469-3_88.
- [24] A. Y. Ng, “Deep Learning Specialization - Neural Networks Deep Learning, Erişim Tarihi: 05.10.2021.” <https://www.coursera.org/learn/neural-networks-deep-learning>.
- [25] “Structure Of A Neuron, Erişim Tarihi: 05.10.2021.” <https://owlcation.com/stem/Structure-of-a-Neuron>.

- [26] Q. J. Huang, "Designing DNN Accelerators, Eriřim Tarihi: 10.05.2021." https://inst.eecs.berkeley.edu/~eecs151/sp21/files/EECS251Lecture-JennyHuang_2021.pdf.
- [27] S. University, "CS231n Convolutional Neural Networks for Visual Recognition." <https://cs231n.github.io/neural-networks-1/> (accessed Dec. 13, 2020).
- [28] F. Rosenblatt, "The perceptron: A probabilistic model for information storage and organization in the brain," *Psychol. Rev.*, vol. 65, no. 6, pp. 386–408, 1958, doi: 10.1037/h0042519.
- [29] A. Y. Ng, "Lecture Notes - Neural Networks Deep Learning, Eriřim Tarihi: 05.10.2021." deeplearning.ai/.
- [30] A. Y. Ng, "Deep Learning Specialization - Convolutional Neural Networks, Eriřim Tarihi: 05.10.2021." <https://www.coursera.org/learn/convolutional-neural-networks>.
- [31] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proc. IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998, doi: 10.1109/5.726791.
- [32] D. H. Hubel, "Single Unit Activity In Striate Cortex Of Unrestrained Cats," *J. Physiol.*, vol. 147, no. 2, pp. 226–238, Sep. 1959, doi: 10.1113/jphysiol.1959.sp006238.
- [33] R. H. Wurtz and R. H. Wurtz, "Recounting the impact of Hubel and Wiesel," *J Physiol*, vol. 587, pp. 2817–2823, 2009, doi: 10.1113/jphysiol.2009.170209.
- [34] A. Y. Ng, "Lecture Notes - Convolutional Neural Networks, Eriřim Tarihi: 05.10.2021." deeplearning.ai/.
- [35] G. T. Shrivakshan, "A Comparison of various Edge Detection Techniques used in Image Processing," *Int. J. Comput. Sci. Issues*, vol. 9, no. 5, pp. 269–276, 2012.
- [36] I. Journal and O. F. S. Circuits, "Design of an Image Edge Detection Filter Using the Sobel Operator," vol. 23, no. 2, pp. 358–367, 1988.
- [37] S. Ssstrunk, R. Buckley, and S. Swen, "Standard RGB color spaces," *Final Progr. Proc. - IS T/SID Color Imaging Conf.*, pp. 127–134, 1999.
- [38] H. Gholamalinezhad and H. Khosravi, "Pooling Methods in Deep Neural Networks, a Review," 2020, [Online]. Available: <http://arxiv.org/abs/2009.07485>.
- [39] D. H. LeCun, Yann, B. Boser, J. S. Denker and L. D. J. R. E. Howard, W. Hubbard, "Handwritten Digit Recognition with a Back-Propagation Network," *Adv. Neural Inf. Process. Syst.*, vol. 39, no. 1 Pt 2, p. 149, 1990, doi: 10.1111/dsu.12130.
- [40] H. Gholamalinezhad and H. Khosravi, "Pooling Methods in Deep Neural Networks, a Review," 2020.
- [41] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," *Commun. ACM*, vol. 60, no. 6, pp. 84–90,

May 2017, doi: 10.1145/3065386.

- [42] K. Simonyan and A. Zisserman, "Very Deep Convolutional Networks For Large-Scale Image Recognition," 2015. Accessed: Jun. 29, 2021. [Online]. Available: <http://www.robots.ox.ac.uk/>.
- [43] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition." [Online]. Available: <http://image-net.org/challenges/LSVRC/2015/>.
- [44] M. Lin, Q. Chen, and S. Yan, "Network in network," *2nd Int. Conf. Learn. Represent. ICLR 2014 - Conf. Track Proc.*, pp. 1–10, 2014.
- [45] C. Szegedy *et al.*, "Going deeper with convolutions," *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, vol. 07-12-June, pp. 1–9, 2015, doi: 10.1109/CVPR.2015.7298594.
- [46] F. Zhuang *et al.*, "A Comprehensive Survey on Transfer Learning," *Proc. IEEE*, vol. 109, no. 1, pp. 43–76, 2021, doi: 10.1109/JPROC.2020.3004555.
- [47] A. G. Howard *et al.*, "MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications," *Comput. Vis. Pattern Recognit.*, Apr. 2017, [Online]. Available: <http://arxiv.org/abs/1704.04861>.
- [48] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L. C. Chen, "MobileNetV2: Inverted Residuals and Linear Bottlenecks," *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, pp. 4510–4520, 2018, doi: 10.1109/CVPR.2018.00474.
- [49] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," *Indian J. Chem. - Sect. B Org. Med. Chem.*, vol. 45, no. 8, pp. 1951–1954, Dec. 2015, doi: 10.1002/chin.200650130.
- [50] M. Tan and Q. V Le, "EfficientNet : Rethinking Model Scaling for Convolutional Neural Networks," *Int. Conf. Mach. Learn. 2019*, 2019.
- [51] P. R. Ng Andrew, "AI for Medicine Specialization - AI for Medical Diagnosis, Erişim Tarihi: 05.10.2021." <https://www.coursera.org/learn/ai-for-medical-diagnosis>.
- [52] A. Esteva *et al.*, "Dermatologist-level classification of skin cancer with deep neural networks," *Nature*, vol. 542, no. 7639, pp. 115–118, 2017, doi: 10.1038/nature21056.
- [53] R. Raman, S. Srinivasan, S. Virmani, S. Sivaprasad, C. Rao, and R. Rajalakshmi, "Fundus photograph-based deep learning algorithms in detecting diabetic retinopathy," *Eye*, vol. 33, no. 1, pp. 97–109, 2019, doi: 10.1038/s41433-018-0269-y.
- [54] "AI for Medical Diagnosis | Coursera." <https://www.coursera.org/learn/ai-for-medical-diagnosis> (accessed Jul. 11, 2021).
- [55] D. F. Steiner *et al.*, "Impact of Deep Learning Assistance on the Histopathologic Review of Lymph Nodes for Metastatic Breast Cancer," 2018. [Online]. Available: www.ajsp.com.

- [56] O. NA, "Receiver operating characteristic curves and their use in radiology," *Radiology*, vol. 229, no. 1, pp. 3–8, Oct. 2003, doi: 10.1148/RADIOL.2291010898.
- [57] S. H. Park, J. M. Goo, and C.-H. Jo, "Receiver Operating Characteristic (ROC) Curve: Practical Review for Radiologists," *Korean J. Radiol.*, vol. 5, no. 1, p. 11, 2004, doi: 10.3348/kjr.2004.5.1.11.
- [58] S. JA, "ROC analysis applied to the evaluation of medical imaging techniques," *Invest. Radiol.*, vol. 14, no. 2, pp. 109–121, 1979, doi: 10.1097/00004424-197903000-00002.
- [59] A. Deniz Ertorsun, B. Bağ, G. Uzar, M. Ali Turanoğlu Danışman, and A. Canan Yazıcı, "Roc (Receiver Operating Characteristic) Eğrisi Yöntemi İle Tanı Testlerinin Performanslarının Değerlendirilmesi," Accessed: Jul. 16, 2021. [Online]. Available: http://tip.baskent.edu.tr/kw/upload/464/dosyalar/cg/sempozyum/ogrsm_pzsnm12/10.2.pdf.
- [60] A. Hazra, "Using the confidence interval confidently," *J. Thorac. Dis.*, vol. 9, no. 10, pp. 4124–4129, Oct. 2017, doi: 10.21037/jtd.2017.09.14.
- [61] A. Majkowska *et al.*, "Chest Radiograph Interpretation with Deep Learning Models: Assessment with Radiologist-adjudicated Reference Standards and Population-adjusted Evaluation," *Radiology*, vol. 294, no. 2, pp. 421–431, Feb. 2020, doi: 10.1148/radiol.2019191293.
- [62] "National Institutes of Health (NIH) | Turning Discovery Into Health." <https://www.nih.gov/> (accessed Jul. 05, 2021).
- [63] X. Wang, Y. Peng, L. Lu, Z. Lu, M. Bagheri, and R. M. Summers, "ChestX-Ray8: Hospital-Scale Chest X-Ray Database and Benchmarks on Weakly-Supervised Classification and Localization of Common Thorax Diseases," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jul. 2017, pp. 3462–3471, doi: 10.1109/CVPR.2017.369.
- [64] Q. Niu, Y. Teng, and L. Chen, "Design of gesture recognition system based on Deep Learning," *J. Phys. Conf. Ser.*, vol. 1168, no. 3, 2019, doi: 10.1088/1742-6596/1168/3/032082.
- [65] X. Wang, Y. Peng, L. Lu, Z. Lu, M. Bagheri, and R. M. Summers, "ChestX-ray8: Hospital-scale Chest X-ray Database and Benchmarks on Weakly-Supervised Classification and Localization of Common Thorax Diseases." Accessed: Jan. 15, 2021. [Online]. Available: <https://uts.nlm.nih.gov/metathesaurus.html>.
- [66] L. Yao, J. Prosky, E. Poblentz, B. Covington, and K. Lyman, "Weakly Supervised Medical Diagnosis and Localization from Multiple Resolutions," *Comput. Vis. Pattern Recognit.*, Mar. 2018, Accessed: Jul. 06, 2021. [Online]. Available: <https://arxiv.org/abs/1803.07703v1>.
- [67] Z. Li *et al.*, "Thoracic Disease Identification and Localization with Limited Supervision," *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, pp. 8290–8299, 2018, doi: 10.1109/CVPR.2018.00865.

- [68] Y. Shen and M. Gao, "Dynamic Routing on Deep Neural Network for Thoracic Disease Classification and Sensitive Area Localization," in *International Workshop Machine Learning in Medical Imaging*, Aug. 2018, pp. 389–397, [Online]. Available: <http://arxiv.org/abs/1808.05744>.
- [69] Y. Tang, X. Wang, A. P. Harrison, L. Lu, J. Xiao, and R. M. Summers, "Attention-Guided Curriculum Learning for Weakly Supervised Classification and Localization of Thoracic Diseases on Chest Radiographs," in *International Workshop Machine Learning in Medical Imaging*, 2018, pp. 249–258.



TEZDEN ÜRETİLMİŞ YAYINLAR

Konferans Bildirileri

1. Başçetin T. S. , Emirođlu İ., “A New Model for The Detection of Breast Cancer Using Neural Networks”, International Conference on Applied Analysis and Mathematical Modeling (ICAAMM19), İstanbul, Türkiye, 10 - 13 Mart 2019, ss.212.

2. Başçetin T. S. , Emirođlu İ., “Comparison of Image Classification Models in Neural Networks”, International Conference on Applied Analysis and Mathematical Modeling (ICAAMM19), İstanbul, Türkiye 10 - 13 Mart 2019, ss.211.

Projeler

1. Yürütücü: Prof. Dr. İbrahim EMİROĐLU, Proje Kodu: 2015-07-03-DOP01, Proje Adı: Derin Öğrenme Yöntemiyle X-Işını Görüntülerinden Yaygın Göğüs Hastalıklarının Tespiti, Projedeki Görevi: Araştırmacı, YTÜ Bilimsel Araştırma Projeleri Doktora Projesi.