



**FATİH SULTAN MEHMET VAKIF ÜNİVERSİTESİ
LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ
BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI
BİLGİSAYAR MÜHENDİSLİĞİ PROGRAMI**

**EVRIŞİMSEL SINIR AĞLARI İLE CİLT HASTALIK
GÖRÜNTÜLERİNİN SINIFLANDIRILMASINDA TRANSFER
ÖĞRENME YÖNTEMİNİN ETKİNLİĞİNİN ARAŞTIRILMASI**

YÜKSEK LİSANS TEZİ

AYHAN SARI

İSTANBUL, 2022



**FATİH SULTAN MEHMET VAKIF ÜNİVERSİTESİ
LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ
BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI
BİLGİSAYAR MÜHENDİSLİĞİ PROGRAMI**

**EVRIŞİMSEL SİNİR AĞLARI İLE CİLT HASTALIK
GÖRÜNTÜLERİNİN SINIFLANDIRILMASINDA TRANSFER
ÖĞRENME YÖNTEMİNİN ETKİNLİĞİNİN ARAŞTIRILMASI**

YÜKSEK LİSANS TEZİ

**AYHAN SARI
(190221003)**

**Danışman
(Dr. Öğr. Üyesi Musa Aydın)**

**İkinci Danışman
(Dr. Öğr. Üyesi Ali Nizam)**

İSTANBUL, 2022

26/05/2022

LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ MÜDÜRLÜĞÜNE

Bilgisayar Mühendisliği Anabilim Dalı'nda 190221003 numaralı Ayhan Sarı'nın hazırladığı "Derin Öğrenme Kullanılarak Medikal Görüntü Bölütleme ve Sınıflandırma" konulu Yüksek Lisans tezi ile ilgili Tez Savunma Sınavı, 26/05/2022 Perşembe günü saat 13:00'da yapılmış, sorulara alınan cevaplar sonunda adayın tezinin **Kabulüne Oy Birliği** ile karar verilmiştir.

Düzeltilme verilmesi halinde:

Adı geçen öğrencinin Tez Savunma Sınavı .../.../20... tarihinde, saat da yapılacaktır.

Tez adı değişikliği yapılması halinde: Tez adının Evrışimsel Sinir Ağları ile Cilt Hastalık Görüntülerinin Sınıflandırılmasında Transfer Öğrenme Yönteminin Etkinliğinin Araştırılması şeklinde değiştirilmesi uygundur.

Jüri Üyesi	Karar
1. (Danışman) Dr. Öğr. Üyesi Musa AYDIN	KABUL
2. Prof. Dr. Ali Yılmaz ÇAMURCU	KABUL
3. Dr. Öğr. Üyesi Eyüp Emre ÜLKÜ	KABUL
4.
5.
6. (İkinci Danışman)* Dr. Öğretim Üyesi Ali NİZAM

*2. Danışman varsa doldurulması gerekmektedir.

ETİK BİLDİRİMİ

Bu tezin yazılmasında bilimsel ahlak kurallarına uyulduğunu, başkalarının eserlerinden yararlanılması durumunda bilimsel normlara uygun olarak atıfta bulunulduğunu, kullanılan verilerde herhangi bir tahrifat yapılmadığını, tezin herhangi bir kısmının bağılı olduğum üniversite veya bir başka üniversitedeki başka bir çalışma olarak sunulmadığını beyan ederim.

Ayhan SARI

TEŐEKKÜR

Yüksek lisans tezimin ilk aşamasından son aşamasına kadar destekleriyle her zaman yanımda olan, bilgi, birikim ve deneyimleriyle ilerlememde büyük katkı sağlayan ve yol gösteren, danışman hocam Dr. Öğr. Üyesi Musa AYDIN'a ve ikinci danışman hocam Dr. Öğr. Üyesi Ali NİZAM'a sonsuz teşekkürlerimi sunarım.

Ayhan SARI

EVRIŞİMSEL SİNİR AĞLARI İLE CİLT HASTALIK GÖRÜNTÜLERİNİN SINIFLANDIRILMASINDA TRANSFER ÖĞRENME YÖNTEMİNİN ETKİNLİĞİNİN ARAŞTIRILMASI

Ayhan Sarı

ÖZET

Yapay zekâ alanındaki ilerlemeler çeşitli hastalıkların tanısında, doktorlar ve alanındaki uzmanlara yardımcı olmakta ve hastalıkların erken teşhisinde tıbbi karar destek sistemleri olarak kullanılmasına olanak tanımaktadır. Medikal görüntülere göre hastalıkları hızlı ve yüksek başarımda tespit etmek için evrişimsel sinir ağları sıklıkla kullanılmaktadır. Dünya genelinde her yıl milyonlarca kişiye cilt kanseri tanısı konulmakta ve çok sayıda insan bu hastalıktan dolayı hayatını kaybetmektedir. Her hastalıkta olduğu gibi cilt hastalıklarında da erken tanı çok önemlidir. Bu nedenle, cilt hastalıklarının bilgisayar destekli makine öğrenmesi tabanlı algoritmalar kullanılarak yüksek doğrulukla tanımlanması çalışmaları önem kazanmıştır.

Bu tez çalışmasında, HAM10000 veri setinin yüksek doğrulukla sınıflandırılması için transfer öğrenme kullanılarak bir yöntem önerilmiştir. Önerilen yöntemde, ImageNet veri seti ile ön eğitilmiş modeller, cilt hastalıkları içeren geniş bir veri seti ile ara eğitimden geçirildikten sonra transfer edilerek HAM10000 veri setindeki cilt lezyonu görüntülerinin sınıflandırılması için kullanılmıştır. Önerilen yöntemin etkinliğini karşılaştırmalı olarak göstermek için, Xception ve DenseNet201 evrişimsel sinir ağı modelleri ayrı ayrı kullanılmıştır. Yapılan deneysel çalışmalarda, HAM10000 veri setindeki görüntülerin sayısı veri artırma (data augmentation) yöntemi ile artırılmıştır. Deneysel çalışmalarda, veri artımı olmadan ve veri artımı

yapılarak evrişimsel sinir ağı modelleri eğitilmiştir. Deneysel çalışmalar iki grup olarak organize edilmiştir. İlk grupta, modeller yalnızca ImageNet ağırlıkları kullanılarak HAM10000 veri seti ile eğitilmiş ve sınıflandırma yapılmıştır. Diğer grupta ise yeni bir yöntem önerilmiş ve önerilen yöntemde modeller, yine ImageNet ağırlıkları kullanılarak cilt hastalıkları içeren başka bir veri seti olan BCN20000 ile eğitildikten sonra, HAM10000 veri seti ile eğitilmek ve sınıflandırma yapmak için eğitilen ağın ağırlıkları transfer edilmiştir. Böylece önerilen ara öğrenme ile transfer öğrenme yöntemde yalnızca ImageNet ağırlıklarını kullanmak yerine, BCN20000 veri seti ile güncellenen ağırlıklar kullanılmıştır.

Yapılan çalışmada önerilen ara öğrenme ile transfer öğrenme yöntemi, yalnızca ImageNet ağırlıkları transfer edilerek gerçekleştirilen deneylere göre daha iyi sonuçlar vermiştir. *Test doğruluğu, kesinlik, duyarlılık ve f1-skoru* ölçütlerine göre, veri artırma uygulanmadığında, Xception modelinde %3.2 ila %3.6, DenseNet201 modelinde %3.4 ila %4.1 aralığında daha iyi sonuçlar elde edilmiştir. Veri artırma uygulandığında ise, Xception modelinde %0.8 ila %1.1, DenseNet201 modelinde %1.3 ila %1.5 aralığında daha iyi sonuçlar elde edilmiştir. Çalışmada en iyi sonuçlar, önerilen ara öğrenme ile transfer öğrenme yöntemi ile veri artırma tekniği uygulanarak eğitilmiş Xception evrişimsel sinir ağı modeli ile elde edilmiş ve sırasıyla *test doğruluğu, kesinlik, duyarlılık ve f1-skoru* ölçütlerine göre %87.9, %87.6, %87.9 ve %87.7 değerleri elde edilmiştir.

Anahtar kelimeler; derin öğrenme, evrişimsel sinir ağları, transfer öğrenme, cilt hastalıkları sınıflandırma, HAM10000

INVESTIGATION OF THE EFFICIENCY OF TRANSFER LEARNING METHOD IN CLASSIFICATION OF SKIN DISEASE IMAGES BY CONVOLUTIONAL NEURAL NETWORKS

Ayhan Sari

ABSTRACT

Advances in artificial intelligence help doctors and experts in the diagnosis of various diseases, and allow it to be used as medical decision support systems in the early diagnosis of diseases. Convolutional neural networks are frequently used to detect diseases quickly and with high performance according to medical images. Millions of people are diagnosed with skin cancer every year around the world, and many people die from this disease. As in any disease, early diagnosis is very important in skin diseases. For this reason, studies on identifying skin diseases with high accuracy using computer-assisted machine learning-based algorithms have gained importance.

In this thesis, a method is proposed using the transfer learning model to classify the HAM10000 dataset with high accuracy. In the proposed method, pre-trained models with the ImageNet dataset were transferred after interim training with a large dataset containing skin diseases and used for classification of skin lesion images in the HAM10000 dataset. To demonstrate the effectiveness of the proposed method, Xception and DenseNet201 convolutional neural network models are used separately. In the experimental studies, the number of images in the HAM10000 data set was increased by the data augmentation method. In experimental studies, convolutional neural network models were trained both without data augmentation and with data augmentation. Experimental studies were organized in two groups. In the first group, the models were trained and classified with the HAM10000 dataset using only

ImageNet weights. In the other group, a new method was proposed and after the models were trained with BCN20000, another data set containing skin diseases, using ImageNet weights, the weights of the trained network were transferred to train and classify with the HAM10000 dataset. Thus, instead of using only ImageNet weights, weights updated with the BCN20000 data set were used in the proposed transfer learning method with intermediate learning.

The intermediate learning and transfer learning method proposed in the study gave better results than the experiments in the first group, which were performed by transferring only ImageNet weights. According to the *test accuracy*, *precision*, *sensitivity*, and *f1-score* criteria, better results were obtained when data augmentation was not applied, ranging from 3.2% to 3.6% in the Xception model and from 3.4% to 4.1% in the DenseNet201 model. When data augmentation was applied, better results were obtained in the range of 0.8% to 1.1% in the Xception model, and between 1.3% and 1.5% in the DenseNet201 model. In the study, the best results were obtained with the Xception convolutional neural network model trained by applying the proposed intermediate learning and transfer learning method and the data augmentation technique, and according to the *test accuracy*, *precision*, *sensitivity* and *f1-score* criteria, respectively, 87.9%, 87.6%, 87.9% and 87.7% values were obtained.

Keywords; deep learning, convolutional neural networks, transfer learning, skin diseases classification, HAM10000

ÖNSÖZ

Bu çalışmada, cilt hastalıklarının sınıflandırılması ele alınmakta ve bunun için yeni bir yöntem önerilmektedir. Önerilen yöntem ile hastalık teşhisinin yüksek doğrulukta sınıflandırılması hedeflenmiştir. Önerilen yöntemde, evrimsel sinir ağları ve transfer öğrenme birlikte kullanılmış ve transfer öğrenmenin etkinliğinin artırılması amaçlanmıştır. Çalışma yöntem itibariyle kısmen ilk olma özelliği taşımakta ve gelecek çalışmalara ışık tutmaktadır.

Bu tezin hazırlanması sürecinde, danışman hocam Dr. Öğr. Üyesi Musa AYDIN ve ikinci danışman hocam Dr. Öğr. Üyesi Ali NİZAM başta olmak üzere, danışman görevlendirilmesinde yardımcı olan FSMVÜ Bilgisayar Mühendisliği bölüm başkanı Dr. Öğr. Üyesi Berna KİRAZ'a, alandaki bilgilerimi daha da geliştirmeme katkı sağlayan FSMVÜ Bilgisayar Mühendisliği bölümü öğretim üyelerine ve kaynak edinmede yardımcı olan FSMVÜ kütüphane personeline teşekkür ederim.

İÇİNDEKİLER

ÖZET.....	IV
ABSTRACT	VI
ÖNSÖZ.....	VIII
ŞEKİL LİSTESİ.....	XI
TABLO LİSTESİ	XIII
KISALTMALAR	XV
GİRİŞ	1
BİRİNCİ BÖLÜM.....	4
1. LİTERATÜR ARAŞTIRMASI.....	4
1.1. CİLT HASTALIKLARININ SINIFLANDIRILMASI.....	4
1.2. HAM10000 VERİ SETİNİN SINIFLANDIRILMASI.....	7
İKİNCİ BÖLÜM.....	12
2. EVRİŞİMSEL SİNİR AĞLARI.....	12
2.1. GİRDİ KATMANI	13
2.2. EVRİŞİM KATMANI.....	15
2.3. ORTAKLAMA (ALT ÖRNEKLEME) KATMANI	17
2.4. ÖZELLİK HARİTALARINI TAM BAĞLANTITILI KATMANA GİRDİ OLARAK HAZIRLAMA.....	18
2.4.1. Düzleştirme Katmanı.....	18
2.4.2. Genel Ortalama Ortaklama Katmanı	19
2.5. BIRAKMA KATMANI.....	21
2.6. TOPLU NORMALLEŞTİRME	22
2.7. AKTİVASYON FONKSİYONLARI	25
2.7.1. ReLU Aktivasyon Fonksiyonu.....	28
2.7.2. Softmax Aktivasyon Fonksiyonu	28
2.8. İLERİ YAYILIM İŞLEMİ.....	30
2.9. MALİYET (KAYIP) FONKSİYONLARI.....	31
2.10. OPTİMİZASYON İŞLEMLERİ.....	33
2.10.1. Adam Optimizasyon Algoritması	34
2.10.2. Tam Bağlantılı Katmanlarda Geri Yayılım	36

2.10.3.Evrifim Katmanlarında Geri Yayılım.....	37
ÜÇÜNCÜ BÖLÜM	40
3. YÖNTEM VE DENEYSEL ÇALIŞMALAR	40
3.1. VERİ SETLERİ.....	44
3.2. EVRİŞİMSEL SİNİR AĞI MODELLERİNİN SEÇİMİ	45
3.2.1. Xception	46
3.2.2. DenseNet201	48
3.3. EVRİŞİMSEL SİNİR AĞI MODELLERİNİN OLUŞTURULMASI VE EĞİTİLMESİ	49
3.3.1. Doğrudan ImageNet Ağırlıkları Kullanılan Deneyler İçin Modellerin Oluşturulması	49
3.3.1.1. Veri Hazırlama ve Artırma	50
3.3.1.2. Modellerin Eğitilmesi	51
3.3.2. Ara Öğrenme ile Transfer Öğrenme Yöntemi Uygulanan Deneyler İçin Modellerin Oluşturulması	53
3.3.2.1. Veri Hazırlama ve Artırma	54
3.3.2.2. Modellerin Eğitilmesi	55
3.4. DENEYSEL SONUÇLAR.....	57
3.4.1. Doğrudan ImageNet Ağırlıkları ile Gerçekleştirilen Transfer Öğrenme Deneylerine Ait Sonuçlar.....	59
3.4.1.1. Veri Artırma Uygulanmadan Elde Edilen Sonuçlar	59
3.4.1.2. Veri Artırma Uygulanarak Elde Edilen Sonuçlar.....	63
3.4.2. Önerilen Ara Öğrenme ile Transfer Öğrenme Yöntemiyle Gerçekleştirilen Deneylere Ait Sonuçlar.....	67
3.4.2.1. Veri Artırma Uygulanmadan Elde Edilen Sonuçlar	67
3.4.2.2. Veri Artırma Uygulanarak Elde Edilen Sonuçlar.....	71
3.4.3. Deneysel Sonuçların Özeti ve Literatürdeki Sonuçlar ile Karşılaştırma	75
SONUÇ VE TARTIŞMA	79
KAYNAKÇA.....	84

ŞEKİL LİSTESİ

Şekil 2.1: Evrişimsel Sinir Ağlarının Genel Yapısı [35]	13
Şekil 2.2: Çok katmanlı standart YSA yapısı [36]	14
Şekil 2.3: 3x3 boyutunda, üç kanallı bir RGB görüntü örneği	14
Şekil 2.4: RGB görüntünün tek bir kanalında gerçekleşen evrişim işlemi [37]	15
Şekil 2.5: Evrişim işlemi için 0 (sıfır) boşluk (zero padding) işlemi [34]	17
Şekil 2.6: Maksimum ortaklama ve Ortalama ortaklama işlemleri	18
Şekil 2.7: Düzleştirme işlemi	19
Şekil 2.8: Xception modelinde Düzleştirme ve GOO karşılaştırması	21
Şekil 2.9: Bırakma işleminden önce (solda) ve sonra (sağda) YSA [41]	22
Şekil 2.10: Tam bağlantılı katmanlarda Toplu Normalleştirme	23
Şekil 2.11: DenseNet ve Xception modellerinde Toplu Normalleştirme kullanımı ..	24
Şekil 2.12: TN ile ve TN olmadan MNIST veri seti deneyi [43]	25
Şekil 2.13: Düğüm yapısı ve aktivasyon fonksiyonunun işlevi [45]	26
Şekil 2.14: ReLU (Düzeltilmiş Doğrusal Birim) aktivasyon fonksiyonu [50]	28
Şekil 2.15: Softmax aktivasyon fonksiyonu ve olasılık vektörü [56]	30
Şekil 2.16: Softmax AF ve Cross-Entropy maliyet fonksiyonu ilişkisi	32
Şekil 2.17: Gradyan iniş algoritmasının yakınsaması [59]	33
Şekil 2.18: Tam bağlantılı katmanlarda geri yayılım işlemi	36
Şekil 2.19: X ve F arasındaki evrişim işlemi	37
Şekil 2.20: Geri yayılım işleminde X ve F değerlerinin güncellenmesi [63]	38
Şekil 3.1: ImageNet ağırlıkları ile doğrudan transfer öğrenme yönteminin HAM10000 veri seti ile uygulanmasına ait akış diyagramı (Ağırlıklar: ImageNet) ..	41
Şekil 3.2: Ara öğrenme ile transfer öğrenme yönteminin (önerilen) HAM10000 veri seti ile uygulanmasına ait akış diyagramı (Ağırlıklar: ImageNet + BCN20000)	43
Şekil 3.3: Yedi sınıflı HAM10000 veri seti örnek görüntüleri	45
Şekil 3.4: Sekiz sınıflı BCN20000 veri seti örnek görüntüleri	45
Şekil 3.5: Derinlemesine evrişim [73]	46

Şekil 3.6: Noktasal evrişim [73].....	47
Şekil 3.7: Xception ESA modelinin genel yapısı [74]	48
Şekil 3.8: DenseNet201 ESA modelinin genel yapısı [76].....	49
Şekil 3.9: HAM10000 veri setinin sınıflandırılmasında, doğrudan ImageNet ağırlıkları kullanılarak eğitilen Xception ve DenseNet201 ağ modelleri	50
Şekil 3.10: HAM10000 veri setinin sınıflandırılmasında, BCN20000 veri seti ile ara öğrenme gerçekleştirilip transfer edilen ağırlıklar kullanılarak eğitilen Xception ve DenseNet201 ağ modelleri	53
Şekil 3.11: Xception ve DenseNet201 val_accuracy / dönem grafiği.....	59
Şekil 3.12: Xception ve DenseNet201 val_loss / dönem grafiği	60
Şekil 3.13: Xception için karışıklık matrisi	60
Şekil 3.14: DenseNet201 için karışıklık matrisi	61
Şekil 3.15: Xception ve DenseNet201 val_accuracy / dönem grafiği.....	63
Şekil 3.16: Xception ve DenseNet201 val_loss / dönem grafiği	64
Şekil 3.17: Xception için karışıklık matrisi	64
Şekil 3.18: DenseNet201 için karışıklık matrisi	65
Şekil 3.19: Xception ve DenseNet201 val_accuracy / dönem grafiği.....	67
Şekil 3.20: Xception ve DenseNet201 val_loss / dönem grafiği	68
Şekil 3.21: Xception için karışıklık matrisi	68
Şekil 3.22: DenseNet201 için karışıklık matrisi	69
Şekil 3.23: Xception ve DenseNet201 val_accuracy / dönem grafiği.....	71
Şekil 3.24: Xception ve DenseNet201 val_loss / dönem grafiği	72
Şekil 3.25: Xception için karışıklık matrisi	72
Şekil 3.26: DenseNet201 için karışıklık matrisi	73

TABLO LİSTESİ

Tablo 2.1: Aktivasyon fonksiyonları ve matematiksel ifadeleri [45]–[47].....	27
Tablo 2.2: Adam optimizasyon algoritması parametreleri [62]	35
Tablo 3.1: Cilt hastalıkları içeren verisetleri ve hastalık sınıflarının örnek sayıları..	44
Tablo 3.2: Xception ESA modelinin temel özellikleri [75]	47
Tablo 3.3: DenseNet201 ESA modelinin temel özellikleri [75]	48
Tablo 3.4: HAM10000 veri seti için eğitim, doğrulama ve test kümesi oranları	50
Tablo 3.5: Veri artırma parametreleri	51
Tablo 3.6: Doğrudan ImageNet ağırlıkları kullanılarak gerçekleştirilen deney sayıları	52
Tablo 3.7: Doğrudan ImageNet ağırlıkları kullanılan deneylerde, modellerin HAM10000 veri seti ile eğitime hiper parametreleri.....	52
Tablo 3.8: BCN20000 veri seti için eğitim ve doğrulama kümesi oranları	55
Tablo 3.9: Ara öğrenme için ağ modellerinin BCN20000 veri seti ile eğitime hiper parametreleri.....	55
Tablo 3.10: Ara öğrenme ile transfer öğrenme yönteminde deney sayıları	56
Tablo 3.11: Ara öğrenme ile transfer öğrenme yönteminde modellerin HAM10000 veri seti ile eğitilmesi	57
Tablo 3.12: Karşılaştırılan alt deney grupları.....	57
Tablo 3.13: Xception için sayısal sonuçlar	61
Tablo 3.14: DenseNet201 için sayısal sonuçlar.....	62
Tablo 3.15: Xception – DenseNet201 performans sonuçları	62
Tablo 3.16: Xception için sayısal sonuçlar	65
Tablo 3.17: DenseNet201 için sayısal sonuçlar.....	66
Tablo 3.18: Xception – DenseNet201 performans sonuçları	66
Tablo 3.19: Xception için sayısal sonuçlar	69
Tablo 3.20: DenseNet201 için sayısal sonuçlar.....	70
Tablo 3.21: Xception – DenseNet201 performans sonuçları	70
Tablo 3.22: Xception için sayısal sonuçlar	73

Tablo 3.23: DenseNet201 için sayısal sonuçlar.....	74
Tablo 3.24: Xception – DenseNet201 karşılaştırması	74
Tablo 3.25: Deneysel çalışmalardan elde edilen sonuçlar	76
Tablo 3.26: Literatürdeki diğer çalışmalar ve sonuçları ile karşılaştırma.....	77

KISALTMALAR

Kısaltma	Açıklama
ESA	Evrişimsel Sinir Ağı
TÖ	Transfer Öğrenme
YSA	Yapay Sinir Ağı
DÖ	Derin Öğrenme
GOO	Genel Ortalama Ortaklama
RGB	Kırmızı – Yeşil - Mavi
AF	Aktivasyon Fonksiyonu
TN	Toplu Normalleştirme

GİRİŞ

Yapay zekanın kullanım alanları gün geçtikçe genişlemekte ve günümüzde çok ileri boyutlara ulaşmış durumdadır. Makine öğrenmesi, bilgisayarlı görü, metin ve ses işleme, doğal dil işleme, robotik, uzman sistemler, planlama gibi birçok alanda kullanılmaktadır. Günümüzde geleneksel makine öğrenmesi yöntemlerinin yanı sıra derin öğrenme yöntemleri de yapay zekâ alanında yaygın olarak kullanılan yöntemlerdir. Makine öğrenimi, verileri ayrıştırmak, bu verilerden öğrenmek ve öğrenilen bilgilere dayanarak bilinçli kararlar vermek için algoritmalar kullanırken, derin öğrenme, kendi başına öğrenebilen ve akıllı kararlar alabilen bir yapay sinir ağı oluşturmak için algoritmaları katmanlar halinde yapılandırır [1]. Bu nedenle derin öğrenme, yapay sinir ağları olarak adlandırılan beynin yapısı ve işlevinden esinlenen algoritmalarla ilgili makine öğreniminin bir alt alanıdır.

Derin öğrenme, büyük miktarda veri üzerinde karmaşık hesaplamalar gerçekleştirmek için kullanılır. Karmaşık, çok katmanlı derin sinir ağları, verilerin insan beynindeki nöronlara benzer düğümler arasında yüksek düzeyde bağlantılı yollarla iletilmesine izin vermek için inşa edilmiştir. Veriler bu bağlantılı yollarda daha soyut hale gelerek dönüşüm geçirmektedir. Derin öğrenmede kullanılan çok sayıda algoritma ve ağ mimarisi bulunmaktadır. Bu algoritmalar genel olarak şu şekilde sıralanabilir [2];

- Evrişimsel Sinir Ağları,
- Uzun Kısa Dönem Bellek Ağları,
- Tekrarlamalı Sinir Ağları,
- Üretici Çekişmeli Ağlar,
- Radyal Temel Fonksiyon Ağları,
- Çok Katmanlı Algılayıcılar,
- Kendi Kendini Düzenleyen Haritalar,
- Derin İnanç Ağları,

- Kısıtlı Boltzmann Makineleri,
- Otomatik Kodlayıcılar

Bilgisayarlı görü alanında yapılan çalışmalar, nesne tanıma, bölütleme ve sınıflandırma alanlarına yoğunlaşmaktadır. Evrişimsel Sinir Ağları (ESA), bilgisayarlı görü alanında en çok tercih edilen derin öğrenme yöntemlerinden biridir. Görüntü miktarının ve veri seti çeşitliliğinin artması da bu alanda yapılan çalışmaları destekler niteliktedir.

Makine öğreniminde ilgilenilen alanlarında birisi de Transfer (aktarım) Öğrenme yöntemidir. Transfer öğrenme, bir problemi çözerken kazanılan bilgiyi depolamaya ve bunu, farklı ancak ilgili bir probleme uygulamaya odaklanan, makine öğreniminde bir araştırma problemidir. Başka bir ifadeyle, halihazırda eğitilmiş modellerin ağırlıklarını farklı bir veri kümesinde kullanarak, o veri kümesindeki görüntüleri ilgili sınıflara ayıştırmaktır [3]. Örneğin, insan cilt dokusunu tanımayı öğrenirken edinilen bilgiler, cilt üzerindeki hastalıkları tanımaya çalışırken uygulanabilir. Yapay zekanın sağlık sektöründe kullanımı konusu, araştırmacıların yaygın ilgi alanlarından birisidir. Özellikle medikal görüntüler üzerinde yapılan sınıflandırma çalışmaları ile hekimlere yönelik tıbbi karar destek sistemlerinin oluşturulması konusunda adımlar atılmaktadır. Derin Öğrenme ve Transfer Öğrenme yaklaşımları kullanılarak medikal görüntüler üzerinde uzman gereksinimine ihtiyaç duyulmadan otomatik sınıflandırma çalışmalarının birçok medikal veri seti üzerinde yapıldığı görülmektedir.

Medikal görüntüler üzerindeki araştırma alanlarından birisi de deri üzerinde oluşan hastalıklarla ilgilidir. Deri üzerinde oluşan hastalıkların, kanser olan veya olmayan ve bunların çok sayıda alt türleri bulunmaktadır. Dünya sağlık örgütüne göre hem melanom dışı hem de melanom cilt kanserlerinin yayılım hızı son on yılda artmaktadır. 2017 yılı verilerine göre her yıl dünya çapında 2 ila 3 milyon arasında melanom dışı cilt kanseri ve 132.000 melanom cilt kanseri meydana gelmektedir [4]. Teşhis edilen her üç kanserden biri cilt kanseridir ve Cilt Kanseri Vakfı İstatistiklerine göre, her beş Amerikalıdan biri yaşamları boyunca cilt kanserine yakalanacaktır.

İstatistiklere bakıldığında [4] deri üzerinde oluşan hastalıklara bağlı olarak ölümlerde ve vakalarda çok büyük sayılara ulaşıldığı görülmektedir. Bu nedenle deri üzerinde oluşan hastalıkların ve lezyon tiplerinin erken aşamada teşhis edilmesi önem arz etmektedir. Aynı zamanda bu çalışma ve buna benzer çalışmalar ile hekimlere yardımcı tıbbi karar destek sistemlerinin oluşturulmasına katkı sağlanabilir. Bu tez çalışması kapsamında, cilt üzerinde oluşan hastalıkların sınıflandırılmasına yönelik olarak aşağıda açıklanan problemler araştırılması amaçlanmıştır:

- ESA kullanılarak cilt hastalıklarının sınıflandırılması,
- ESA ve Transfer Öğrenme yöntemleri kullanılarak, ImageNet veri seti ile ön eğitilmiş ESA modelleri transfer edildikten sonra, cilt hastalıkları içeren başka bir veri ile bir ara öğrenme aşamasından geçirildikten sonra başka hedef bir cilt hastalıkları veri setinin hastalık sınıflandırması üzerindeki sınıflandırma başarısına olan etkileri,
- Xception ve DenseNet201 ESA modellerinin sınıflandırma performanslarının hem çalışmanın amacı bakımından hem de iki modelin karşılaştırılması açısından çeşitli metrikler kullanılarak sayısal verilerle ifade edilmesi.

Tezin amacı doğrultusunda, tezin birinci bölümünde cilt hastalıklarının sınıflandırılmasına yönelik olarak, ESA'lar kullanılarak son yıllarda yapılan güncel hastalık sınıflandırma çalışmaları detaylı olarak sunulmaktadır. İkinci bölümde, ESA'yı oluşturan katmanlar, bu katmanlarda gerçekleşen işlemler ve kullanılan teknikler ve algoritmalar açıklanmaktadır. Üçüncü bölümde, cilt hastalıkları içeren literatürdeki güncel veri setleri ve bu veri setlerinden bu tez çalışmasında kullanılanlar açıklanmıştır. Bununla birlikte kullanılan yöntem, gerçekleştirilen deneyler ve deneylere ilişkin sonuçlar, karmaşıklık matrisi gibi çeşitli metriklerle ortaya konulmuştur. Tezin son bölümünde ise, elde edilen deneysel sonuçlar yorumlanarak, cilt hastalıklarının sınıflandırılmasına yönelik literatürdeki diğer çalışmalarla karşılaştırmalı olarak sunulmuştur.

BİRİNCİ BÖLÜM

1. LİTERATÜR ARAŞTIRMASI

Cilt lezyonlarının sınıflandırılmasına yönelik çalışmalar, cilt lezyonları bulunan görüntüler kullanılarak doğrudan sınıflandırma ve cilt lezyon görüntülerindeki lezyonların önce bölütlenmesi ve daha sonra sınıflandırması şeklinde iki ana kategoride incelenebilir. Bu alanda 2016'dan beri Uluslararası Cilt Görüntüleme İşbirliği (ISIC - International Skin Imaging Collaboration) adıyla yapılan yarışmalar, Uluslararası Dijital Cilt Görüntüleme Topluluğu (ISDIS - International Society for Digital Imaging of the Skin) tarafından desteklenmekte ve özellikle en ölümcül cilt kanseri türü olan melanom teşhisini geliştirmeye yönelik çalışmalar olarak varlığını sürdürmektedir [5]. ISIC arşivi [14], dermatologlar tarafından kalite kontrollü dermoskopik görüntülerin halka açık en büyük koleksiyonunu içermektedir. ISIC arşivi 2021 sonu itibariyle binlerce hastadan toplanan, toplam 69445 cilt hastalık görüntüsü içermektedir. HAM10000 veri seti ISIC-2018 ve sonraki yıllarda yapılan yarışmaların veri setlerinde yedi sınıflı 10015 örnek ile yer almaktadır. BCN20000 veri seti se ISIC-2019'da sekiz sınıflı 12413 örnek ile temsil edilirken, 2020'de dokuz sınıflı olarak (9. sınıfın hastalık türü bilinmiyor) 19424 örnek ile temsil edilmektedir. Veri setlerine ilişkin detaylar Bölüm 3.1'de açıklanmıştır. Bu bölümde, cilt hastalıklarının sınıflandırılması ile ilgili son dönemde yapılan çalışmalar incelenmiş ve kronolojik olarak sunulmuştur.

1.1.CİLT HASTALIKLARININ SINIFLANDIRILMASI

Tezin bu bölümünde cilt hastalıklarını içeren çeşitli veri setleri ve HAM10000 veri seti kullanılarak yapılan hastalık sınıflandırması ile ilgili son birkaç yılda yapılan çalışmalar detaylı bir şekilde sunulmuştur.

Elhamdy ve ark. [6] 2017 yılında, örnek sayıları dengeli sayılabilecek üç sınıflı görüntü içeren ZF-35 veri setini (471 görüntü) kullanarak cilt hastalıklarını sınıflandırmak için yaptıkları çalışmada, transfer öğrenme yaklaşımıyla ImageNet ile ön eğitilmiş AlexNet [7] ESA modelini ImageNet [8] ağırlıkları ile birlikte kullanmışlardır. Gerçekleştirilen çalışmada görüntü kopyalamaya dayalı yeni bir görüntü artırma algoritması ile çeşitli görüntü artırma teknikleri uygulanmıştır. Önerilen modelde SGD optimizasyon algoritması kullanılmış ve %97.95 duyarlılık, %99.08 özgüllük ve %98.67 doğruluk oranları elde edilmiştir.

Hosny ve ark. [9] 2018 yılında, üç sınıfta dengeli örnek sayısına sahip toplam 200 görüntüden oluşan PH2 [10] veri setini kullandıkları çalışmada, çeşitli görüntü artırma teknikleri ve SGD optimizasyon algoritmasını kullanarak ImageNet ile ön eğitilmiş AlexNet ESA modelini kullanmışlardır. Önerilen model ile %98.61 ortalama doğruluk, %98.33 ortalama duyarlılık, %98.93 ortalama özgüllük ve %97.73 ortalama kesinlik oranları elde edilmiştir.

Dorj ve ark. [11] 2018 yılında internetten topladıkları dört tip cilt kanseri görüntüsü ile elde ettikleri veri setini (3753 görüntü) kullanarak ImageNet veri seti ile ön eğitilmiş AlexNet ESA modeli ile bir sınıflandırma çalışması yapmışlardır. Veri setinde bulunan kanser türlerinin örnek sayıları birbirine çok yakın oluşturulmuş ve veri seti yalnızca eğitim ve test olarak ayrıştırılmış, doğrulama için veri ayrılmamıştır. Bir ECOC SVM sınıflandırıcı kullanılarak yapılan sınıflandırma işleminde dört cilt kanseri türü için ortalama en yüksek %95.1, en küçük %91.8 doğruluk oranı elde edilmiştir.

Rodrigues ve ark. [12] ve ark. 2020 yılında, üç sınıflı PH2 (200 görüntü) [10] ve iki sınıflı ISIC-2016 (900 görüntü) [5] verisetlerini kullanarak yaptıkları çalışmada, ImageNet ile ön eğitilmiş çok sayıda son teknoloji (state of art) ESA modeli ve çok sayıda klasik sınıflandırıcı türünün kombinasyonunu kullanmışlardır. Çalışmada veri artırma tekniği uygulanmış ve ISIC-2016 için %96.805, PH2 için ise %93.167'lere varan oranda doğruluk değerleri elde edilmiştir.

Yılmaz ve Trocan [13] 2020 yılında lezyonları yalnızca iyi huylu ve kötü huylu şeklinde ikili sınıflandırma yapmak amacıyla yaptıkları çalışmayı, ISIC arşivinden

[14] edindikleri 21570 dermoskopik görüntü (19373 iyi huylu, 2197 kötü huylu) ile gerçekleştirmişlerdir. Yapılan çalışmada ImageNet [8] ile ön eğitilmiş AlexNet, GoogLeNet ve ResNet50 ESA modelleri karşılaştırmalı olarak sunulmuştur. Üç model arasında, ResNet50 modeli %92.81 doğruluk oranı ile en iyi performansı göstermiştir.

Sasikala ve ark. [15] 2020 yılında yaptıkları bir çalışmada, veri bilimcileri ve makine öğrenimi uygulayıcılarının çevrimiçi bir topluluğu olan Kaggle [16]'dan edindikleri 1800 iyi huylu ve 1800 kötü huylu olmak üzere toplam 3600 cilt hastalık görüntüsü ile yaptıkları sınıflandırma çalışmasında, ImageNet ile ön eğitilmiş dört ESA modeli ile deneyler gerçekleştirmişlerdir. AlexNet, VGG16, ResNet34 ve ResNet50 ESA modellerinin kullanıldığı çalışmada, Resnet34 modeli ile %90.12'lik doğrulama oranı elde edilmiştir.

Mukherjee ve Ganguly [3], 2020 yılında Kaggle'dan 12500 kedi ve köpek görüntüsü ile eğittikleri dört evrişim katmanı ve bir gizli katmandan oluşan klasik bir ESA modelini transfer ederek, internette edindikleri iki sınıflı 179 cilt hastalık görüntüsü üzerinde test etmişlerdir. Melanom ve melanom değil şeklinde sınıflandırma yapılan bu çalışmada, 0.90 duyarlılık, 0.34 özgüllük ve 0.60 doğruluk oranları elde edilmiştir.

Kassem ve ark. [17] 2020 yılında ISIC-2019 yarışmasının sekiz sınıftan oluşan 25331 cilt hastalık görüntüsünü hastalık sınıflarına sınıflandırmak için yaptıkları çalışmada, ImageNet ile ön eğitilmiş GoogLeNet (Inception V1) ESA modelini kullanmışlardır. Görüntü artırma ve azaltma teknikleri de kullanılarak gerçekleştirilen üç farklı deneyde, doğruluk, duyarlılık, özgüllük ve kesinlik ölçütleri bakımından sırasıyla, %94.92, %79.80, %97.00 ve %80.36 oranları elde edilmiştir.

Steppan ve Hanke [18] 2021 yılında, ISIC-2019 [5] yarışması veri seti ile birlikte PH2 [19][10], Light Field Image [20], SD-198 [21], MED-NODE [22] ve Seven-point criteria evaluation [23] veri setlerinin kombinasyonunu kullanarak oluşturdukları geniş veri setini, 29469 eğitim, 3279 doğrulama görüntüsü olacak şekilde ayırmışlardır. Test için etiketleri halka açık olmayan ISIC-2019'un 8238 görüntüden oluşan test setini kullanmışlardır. Görüntü ön işleme, gerçek zamanlı veri artırma, dengeli veri seti oluşturmak için yeniden örnekleme ve sınıf ağırlıkları

yöntemi, transfer öğrenme, erken durdurma, test zamanında veri artırma, eğitilmiş farklı modellerin birleştirilmesi gibi teknikler de bu çalışmada kullanılmıştır. Ön eğitilmiş Inception-ResNet-V2, SE-ResNeXt-101, NASNet-A-Large, EfficientNet-B4 ve EfficientNet-B5 ESA modelleri bu çalışmada dokuz sınıflı sınıflandırma için kullanılmıştır. Modellerin ortalama 0.634 dengeli çok sınıflı doğruluk oranı elde ettiği görülmüştür. Modellerin kombinasyonunda NASNet-A-Large modeli, elde edilen yetersiz genel doğruluk nedeniyle topluluğa dahil edilmemiş, EfficientNet tüm eğitilmiş ağ mimarilerinin en iyi sonuçlarını gösterse de, kombinasyon, SE-ResNeXt-101 ve Inception-ResNet-v2 modellerinden gelen tahminlerle herhangi tek bir modelin bağımsız olarak elde edebileceğinden daha yüksek ortalama doğruluk sağlamıştır. Birleştirilmiş modelin ortalama doğruluk değeri 0.923 olarak bulunmuştur.

Yapılan çalışmalar incelendiğine, çeşitli veri artırma tekniklerinin kullanıldığı ve birkaç çalışma hariç nispeten az sınıflı (iki, üç veya dört) veriler ile sınıflandırma yapıldığı görülmektedir. Ayrıca sınıf sayısının az olduğu çalışmalarda doğruluk, duyarlılık, özgüllük, kesinlik gibi değerlendirme ölçütlerinde elde edilen sınıflandırma sonuçlarının daha iyi olduğu görülmektedir.

1.2.HAM10000 VERİ SETİNİN SINIFLANDIRILMASI

Üçüncü bölümde detayları açıklanan HAM10000, halka açık olarak sunulmuş, yedi sınıfta cilt hastalık görüntüsü içeren ve sınıflardaki örnek sayısı bakımından dengesiz bir veri seti olarak karşımıza çıkmaktadır [24]. Veri setinin örnekler bakımından dengesiz olması bir dizi sorunu beraberinde getirmektedir. Yapılan çalışmalarda bu sorunların üstesinden gelmek için çeşitli yöntemler önerilmiş ve deneyler gerçekleştirilmiştir.

Younis ve ark. [25] 2019 yılında yaptıkları çalışmada, yedi sınıflı sınıflandırma için ImageNet ile ön eğitilmiş MobileNet ESA modelini kullanarak, eğitim verisinde örnek sayısı az olan sınıflarda veri büyütme yöntemi uygulamışlardır. Adam optimizasyon algoritması kullanılan çalışmada, kategorik doğruluk, ağırlıklı ortalama kesinlik ve ağırlıklı ortalama hatırlama (duyarlılık) ölçütleri bakımından sırasıyla %97.07 (sınıf bazında en yüksek), 0.90 (sınıf bazında 0.20 ile 0.97 arası) ve %91.34 (sınıf bazında en yüksek) oranları elde edilmiştir.

Rahman ve Ami [26] 2020 yılında yaptıkları çalışmada, yedi sınıflı sınıflandırma için ImageNet ile ön eğitilmiş üç bilinen model olan ResNet, DenseNet ve Xception modellerini kullanmıştır. Çalışmada öncelikle her bir model tek eğitilerek performansları ölçülmüş ve daha sonra üç model ağırlıklandırılmış topluluk yaklaşımıyla birleştirilerek bütünleşik bir sistem oluşturulmuştur. Çalışmada ayrıca, özellik çıkarımını iyileştirmek için görüntüler üzerinde gürültü azaltma tekniği uygulanmış, modelin çok fazla örnek içeren sınıfa yanlı öğrenmesinden kaçınmak için kayıp ağırlık yöntemi ve çalışma zamanı anlık veri büyütme işlemleri uygulanmıştır. Sonuç olarak, ağırlıklandırılmış ortalama topluluk değerleri bakımından 0.858 ağırlıklı ortalama doğruluk, 0.81 kesinlik, 0.83 f1 puanı, 0.97 özgüllük oranları elde edilmiştir.

Hosny ve ark. [27] 2020 yılında gerçekleştirdikleri bir çalışmada, yedi sınıflı sınıflandırma için ImageNet ile ön eğitilmiş AlexNet ESA modelini kullanmışlardır. Verileri eğitim ve test olarak bölmek için 10 katlı çapraz doğrulama tekniği kullanılmış ve üç farklı deney gerçekleştirilmiştir. Herhangi bir veri büyütme olmaksızın yapılan ilk deneyde %94.90 doğruluk, %82.17 duyarlılık, %97.03 özgüllük ve %82.17 kesinlik oranları elde edilmiştir. İkinci deneyde tüm sınıflardaki örnek sayısı, örnek sayısı en yüksek olan nevüs hastalık sınıfı ile eşitlenerek 6700 ± 5 olarak ayarlanmış ve yaklaşık 47 bin görüntü elde edilmiştir. Üçüncü deneyde ise nevüs sınıfı hariç diğer sınıflarda yüksek miktarda veri artırma uygulanmış ve yaklaşık 250 bin görüntü elde edilmiştir. İkinci ve üçüncü deneylerde test verisi orijinal görüntülerden değil, artırılmış görüntülerden elde edildiği için eğitim verileri ile daha çok benzerlik göstermektedir. Bu da daha yüksek doğruluk oranlarının elde edilmesini sağlamıştır. Ancak test verisinin orijinal olmaması bir eksiklik olarak görülebilir.

Sevli [28], 2021 yılında yaptığı çalışmada, yedi sınıflı sınıflandırma için öncelikle veri seti üzerinde kontrast iyileştirme ve keskinleştirme filtresi gibi ön işlemler gerçekleştirmiştir. Klasik ESA modeli ile gerçekleştirilen çalışmada, son teknoloji ESA modellerinin aksine, girdi boyutu 224×224 olarak değil 75×100 olarak ayarlanmış ve eğitim sırasında veri artırma tekniği kullanılmıştır. Sınıf bazında kesinlik, hatırlama ve f1 puanları bakımından sırasıyla, 0.30 - 0.91, 0.38 - 0.90 ve 0.45 - 0.90 aralığında oranlar elde etmişlerdir. Burada en yüksek oranın, en yüksek örnek sayısına sahip nevüs sınıfında gerçekleştiği, diğer sınıflarda oranların düşük olduğu

görülmektedir. Önerilen modelin önyargılı öğrendiği ve iyi genelleme yapamadığı söylenebilir.

Calderon, Sanchez, Castillo ve Arguello [29], 2021 yılında bir çift doğrusal ESA modeli yaklaşımıyla yedi sınıflı sınıflandırma çalışması yapmışlardır. Transfer öğrenme yönteminin de kullanıldığı çalışmada, ImageNet ile ön eğitilmiş ResNet50 ve VGG16 ve ResNet50 ve Xception modelleri paralel olarak eğitilmiş, evrişim katmanlarının çıktısında dış çarpım yapılarak çift doğrusal vektör elde edilmiş ve tam bağlantılı katmanlara girdi olarak verilmiştir. Çalışmada, %5 doğrulama (501 görüntü), %15 test (1502 görüntü) ve %80 (8012 görüntü) eğitim için ayrılmış, eğitim görüntülerine veri artırma uygulanmıştır. Veri artırma sonucunda toplam eğitim görüntüsü sayısı yaklaşık 40 bin olarak gerçekleşmiştir. Adam optimizasyon algoritması kullanılarak yapılan eğitim sonucunda, kesinlik, hatırlama ve f1 puanları açısından ResNet50 ve VGG16 çift doğrusal modeli ile sırasıyla 0.86, 0.77, 0.81 makro ortalama (7 sınıfın aritmetik ortalaması), 0.93, 0.93, 0.93 ağırlıklı ortalama (sınıflardaki örnek sayılarının ağırlığına göre) oranları elde edilirken, ResNet50 ve Xception modeli ile 0.79, 0.74, 0.76 makro ortalama, 0.93, 0.93, 0.93 ağırlıklı ortalama oranları elde edilmiştir.

Villa ve diğ. [30] 2021 yılında transfer öğrenme ve ince ayar optimizasyonuna dayalı yaptıkları çalışmada, yedi sınıflı sınıflandırma için veri setine veri artırma da uygulayarak, DenseNet-201, Inception-ResNet-V2 ve Inception-V3 son teknoloji ESA modelleri ile sınıflandırma yapmışlardır. Veri artırma için 6705 örnek görüntüye sahip nevüs sınıfı dışındaki sınıflarda, artırma ile görüntü sayısı olarak en yüksek örnek sayısına yakın değerlere getirilmiş ve toplam görüntü sayısı yaklaşık 43 bine getirilmiştir. Ayrıca görüntüler üzerinde kontrast iyileştirme, gürültü filtreleme gibi teknikler uygulanmıştır. Görüntülerin %75'i eğitim, %17'si doğrulama ve %8'i test verisi olarak ayrıştırılmıştır. Ancak burada dikkat çeken, test verisi başlangıçta orijinal verilerden ayrılmamış, veri artırma uygulandıktan sonra veri seti eğitim, doğrulama ve test gruplarına ayrıştırılmıştır. Önerilen modelde her bir ESA modeline, 512, 128, 64, 64 ve 32 düğümlü yoğun katmanlar (dense layer), her bir yoğun katmandan sonra toplu normalleştirme (batch normalization) ve ReLU AF katmanları eklenmiş ve çıkış katmanından sonra bir Softmax sınıflandırıcı ile sınıflandırma gerçekleştirilmiştir.

Farklı modellerle gerçekleştirilen deneylere göre kesinlik, hatırlama ve f1-puanı ölçütleri bakımından DenseNet-201 modeli için 0.98, Inception-ResNet-V2 modeli için 0.97, Inception-V3 modeli için 0.96 oranları elde edilmiştir.

Yao ve diğ. [31] 2021 yılında dengesiz veri seti sınıflandırması için yaptıkları çalışmada, modifiye edilmiş rastgele artırma adını verdikleri yeni bir veri artırma stratejisi önermişlerdir. Bununla birlikte eşit olmayan örneklem büyüklüğü ve sınıflandırma zorluğunun üstesinden gelmek için çoklu ağırlıklı odak kayıp fonksiyonu adını verdikleri bir yöntemi de tanıtmışlardır. Bu iki özelliği birleştirdikleri bir ESA modelinde fazla uydurmayı azaltmak için Dropout ve Dropblock katmanları kullanılan çalışmada, performans değerlendirmesi için ISIC-2018’de yer alan, etiketleri halka açık paylaşılmamış 1512 test görüntüsü kullanılmıştır. Önerilen yaklaşımda, RegNetY-3.2GF ESA modeli ile 0.864 dengeli test doğruluk oranı elde edilmiştir.

HAM10000 veri setinin yedi çeşit hastalık türüne sınıflandırılmasının dışında bazı çalışmalar melanom-melanom değil ya da iyi huylu-kötü huylu lezyonlar şeklinde ikili sınıflandırıldığı çalışmalar da bulunmaktadır.

Ali, Miah, Haque ve Rahman [32], 2021 yılında yaptıkları çalışma ile HAM10000 veri setinde yer alan cilt lezyonlarını iyi huylu ve kötü huylu olarak sınıflandırmak için ikili sınıflandırma gerçekleştirmişlerdir. ESA’larda yaygın olarak kullanılan transfer öğrenme yöntemine dayalı olarak gerçekleştirilen bu çalışmada, klasik bir ESA modeli önerilmiş ve bu model geniş bir medikal veri seti ile eğitilerek, daha sonra HAM10000 veri setinin ikili sınıflandırması için tekrar eğitilmiştir. Önerilen ESA modeli, literatürde yer alan önceki çalışmalardan ImageNet ile ön eğitilmiş AlexNet, ResNet, VGG-16, DenseNet ve MobileNet ESA modelleriyle karşılaştırılmıştır. Veri setinin hazırlanmasında, bulanık görüntüler veri setinden çıkarılmış, gürültülerin yok edilmesi için ön işleme yapılmıştır. Ayrıca bulanık, düşük kontrastlı, lezyonun yanında ben olan, renkli aydınlatma içeren vb. görüntüler manuel olarak veri setinden çıkarılmış, son durumda 6136 iyi huylu, 979 kötü huylu lezyon görüntüsünden oluşan bir veri seti elde edilmiş, aşırı uyumun üstesinden gelmek için veri artırma yöntemi uygulanmıştır. İkili sınıflandırma yapıldığı için softmax yerine sigmoid AF, optimize edici olarak Adam tercih edilmiştir. Farklı eğitim, test ve

doğrulama verisi oranlarıyla yapılan iki deneyde en yüksek, %96.57 kesinlik, %93.66 hatırlama, %95.09 f1 puanı, %91.43 test doğruluğu oranları elde edilmiştir.

Islam, Ali ve ark. [33], 2021 yılında HAM10000 veri setini iyi huylu ve kötü huylu lezyonlar olarak ikili sınıflandırmak için yaptıkları çalışmada filtreleme ve gürültü azaltma teknikleri uygulanmış, ayrıca fazla gürültü içeren görüntüler veri setinden çıkarılmıştır. Ayrıca, girdi boyutu 96x96 olarak ayarlanmış ve veri artırma teknikleri ile veri artırma uygulanmıştır. Klasik ESA modeli önerilen bu çalışmada Adam optimizasyon algoritması kullanılarak, %94.69 kesinlik, %94.13 hatırlama, %94.73 f1 puanı ve %90.93 test doğruluk oranları elde edilmiştir.

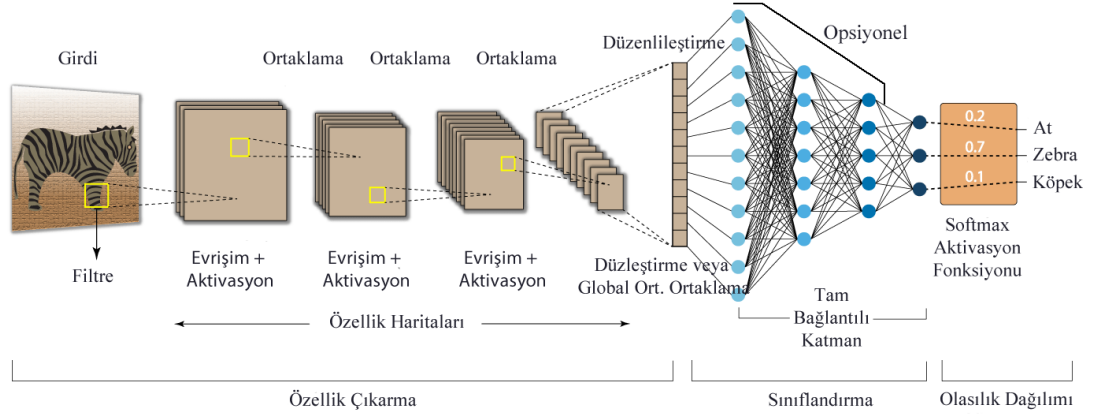
Literatüre bakıldığında cilt hastalıklarının veya kanser türlerinin sınıflandırılmasıyla ilgili çalışmaların büyük çoğunluğunun iyi huylu-kötü huylu veya melanom-melanom değil şeklinde ikili sınıflandırma olarak gerçekleştirildiği görülmüştür. Bunun dışında üç veya dört sınıflı ve dengeli sayıda örneğe sahip veri setleri üzerinde gerçekleştirilmiş çalışmalar da bulunmaktadır. Yedi, sekiz ve daha fazla hastalık türüne sınıflandırma ile ilgili çalışmaların diğerlerine göre daha az olduğu söylenebilir. Ayrıca ikili sınıflandırma yapılan çalışmalarında, yedi hastalık türüne sınıflandırma çalışmalarına göre daha yüksek başarı oranları elde edildiği görülmektedir. Bu, sınıf sayısı arttıkça sınıflandırma probleminin zorlaştığının bir göstergesi olarak düşünülebilir.

İKİNCİ BÖLÜM

2. EVRİŞİMSEL SİNİR AĞLARI

Derin öğrenme kavramı, makine öğrenmesinin bir alt alanı olarak, çok katmanlı YSA'nın inşa edilmesiyle ortaya çıkmıştır. Çeşitli DÖ modelleri bulunmakta ve bu modeller arasında ESA, ImageNet Büyük Ölçekli Görsel Tanıma Yarışması (ILSVRC, 2012) olarak bilinen nesne tanıma yarışmasında başarılı sonuçların elde edilmesinden sonra bilgisayarlı görme görevlerinde önemli bir yöntem olarak YSA modelleri arasındaki yerini almıştır [34] ve bu yarışma ile birlikte son yıllarda derin öğrenmeye olan ilgide ve bu alanda yapılan çalışmaların sayısında büyük artış yaşanmıştır.

ESA'lar, bir veya daha fazla evrişimsel katmana sahip olan ve esas olarak görüntü işleme, görüntü tanıma, sınıflandırma, bölütleme, nesne tespiti, yüz tanıma gibi alanlarda yaygın olarak kullanılan derin sinir ağlarıdır. Bu ağlar, basit olarak girdi katmanı (input layer), evrişim katmanları (convolve layers), düzleştirme katmanı (flatten, tam bağlantılı katmanlar (dense layers) ve çıktı katmanından (output layer) oluşmaktadır. Bunların dışında aktivasyon, normalleştirme, ortaklama (pooling), seyreltme/bırakma (dropout), genel ortalama ortaklama katmanları da kullanılmaktadır. Bir ESA'ya ait temel mimari yapı Şekil 2.1'de gösterilmiştir.

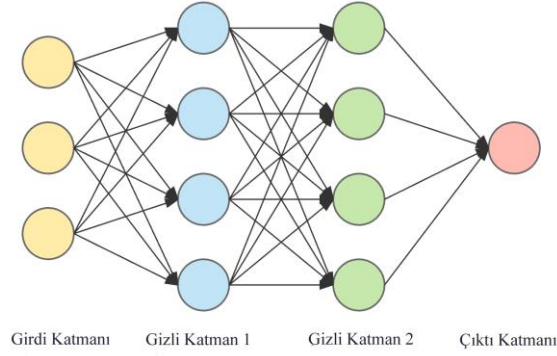


Şekil 2.1: Evrişimsel Sinir Ağlarının Genel Yapısı [35]

ESA, girdilerden özellik çıkarmak amacıyla en az bir adet evrişim katmanından oluşmaktadır. Standart YSA’larda evrişim katmanı bulunmaz. Girdi doğrudan gizli katmanlara (hidden layer) bağlanır (bkz. Şekil 2.2). Evrişim katmanlarında, girdi görüntüleri üzerinde filtrelerin dolaşması sonucu özellik çıkarımı gerçekleştirilmektedir. Geleneksel YSA’nda bir görüntü tanıma gerçekleştirebilmek için, girdiyi temsil eden özelliklerin YSA’ya bildirilmesi gerekmektedir. Doğru özelliklerin seçilmesi zor bir işittir. ESA’da özellik çıkarma işlemi evrişim katmanlarında otomatik olarak gerçekleştiği için, ham veri doğrudan giriş katmanına verilebilmektedir.

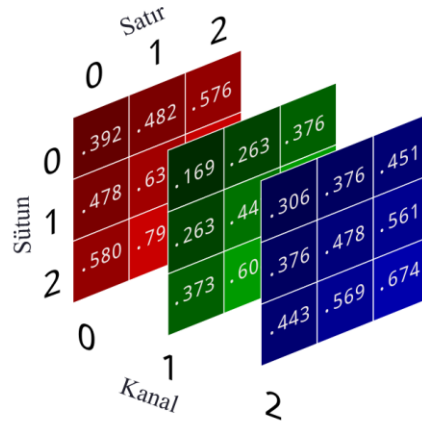
2.1. GİRDİ KATMANI

Standart çok katmanlı YSA, temel olarak üç katmandan meydana gelmektedir. Bunlar girdi katmanı (input layer), gizli katman/katmanlar (hidden layers) ve çıktı katmanıdır (output layer). Şekil 2.2’de çok katmanlı standart YSA yapısı gösterilmektedir. Veri setine göre değişen özelliklerin her biri farklı bir düğüm olarak girdi katmanında temsil edilir. Her bir girdinin ağırlık değeri vardır. Bu girdiler gizli katmandaki düğümlere bu ağırlıklar aracılığı ile bağlıdır. Girdiler için belirlenen ağırlık değerleri yapay sinir ağında o özelliğin önemini göstermektedir.



Şekil 2.2: Çok katmanlı standart YSA yapısı [36]

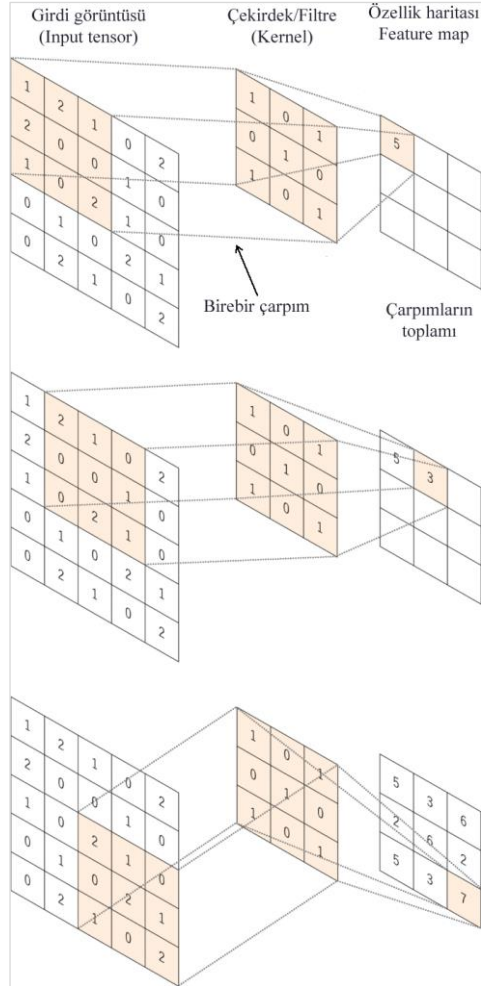
ESA'nın girdi katmanına verilen ham görüntüler tek kanallı (gri tonlamalı) veya üç kanallı (RGB) olabilir. 224×224 piksel boyutunda bir RGB görüntü $224 \times 224 \times 3$ (*Genişlik x Yükseklik x Kanal Sayısı*) kadar piksel değeri içerir. Her bir girdi görüntüsünü, piksel değerlerinden oluşan bir tensör olarak düşünebiliriz. Tensör çok boyutlu verileri simgelemek için kullanılır. Vektör ve matrislerin istenilen boyuta genelleştirilmesini ifade eder. RGB bir görüntünün her bir renk kanalını iki boyutlu bir matris (2D tensör) şeklinde ifade edebiliriz. 224×224 boyutunda 3 kanallı bir girdi görüntüsü için toplam piksel sayısı $224 \times 224 \times 3 = 150.528$ olur. Böylece $224 \times 224 \times 3$ boyutlarına sahip bir görüntü 150.528 elemandan oluşan bir vektör olarak temsil edilmiş olur. Her bir piksel değeri, varsayılan olarak 0 ile 255 arası tamsayı değerlerden oluşur. Şekil 2.3'de üç kanallı, 3×3 piksel boyutunda bir görüntünün normalize edilmiş (her bir piksel değeri 255'e bölünür) piksel değerleri temsili olarak gösterilmiştir.



Şekil 2.3: 3×3 boyutunda, üç kanallı bir RGB görüntü örneği

2.2. EVRİŞİM KATMANI

Evrifim iflem sreci, evrifim, adım (stride) ve dolgu (padding) gibi birkaç kavram ierir. ekirdek veya filtre olarak adlandırılan ađırlık matrisi, girdi grnts zerinde adım adım gezdirilerek, orijinal grntdeki piksel deđerleri (sayılar), ađırlık matrisinin kesişen konumdaki sayısı ile arpılır. Tm bu arpma iřlemlerinin toplamı, ekirdeđin normalleřtiricisine blnr ve sonu olarak ekirdeđin ortalandıđı konumda yeni oluřturulan grntye yerleřtirilir [37]. Bylece, ESA'larda evrifim iřlemi gerekleřmiř olur. řekil 2.4'de evrifim iřleminin ařamaları gsterilmektedir.



řekil 2.4: RGB grntnn tek bir kanalında gerekleřen evrifim iřlemi [37]

Evrişim işleminde çekirdek boyutu $kernel_size$ parametresi ile ifade edilir ve genellikle 1x1, 2x2, 3x3 veya daha büyük matris boyutlarında tercih edilmektedir. Şekil 2.4'deki gösterimde $kernel_size = 3 \times 3$ olarak alınmıştır. Bununla birlikte filtre sayısı, elde edilecek özellik haritası sayısını gösterir ve $filters$ parametresi ile ifade edilir. Bir diğer parametre, filtrenin bir seferdeki adım sayısını ifade eden $strides$ parametresidir [34]. Filtrenin tek seferde girdi görüntüsü üzerinde 1 adım hareket etmesi $strides = 1$ anlamına gelir. Şekil 2.4'deki gösterimde $strides = 1$ olarak alınmıştır.

Girdi görüntüleri genellikle kare matris olarak tercih edilir. Bir girdi görüntüsü $I = m \times m$ boyutunda, filtre sayısı $filters = k$, çekirdek boyutu $kernel_size = n \times n$ ve tek seferde adım sayısı $strides = s$ olmak üzere, her bir filtrenin RGB görüntünün sadece bir kanalı üzerindeki adım sayısı Denklem 2.1'deki gibi hesaplanır. Bu sonuç aynı zamanda elde edilen yeni görüntünün, bir diğer ifadeyle özellik haritasının genişlik ve yükseklik boyutlarını ifade eder.

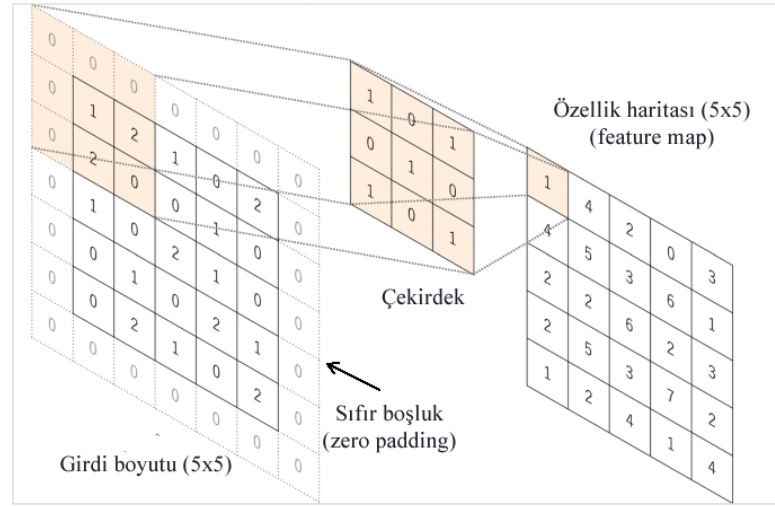
$$Adım\ sayısı = \left(\frac{(m - n)}{s} + 1 \right) \times \left(\frac{(m - n)}{s} + 1 \right) \quad (2.1)$$

Denklem 2.1'e göre k adet filtrenin görüntü üzerinde gezinmesi sonucunda Denklem 2.2'deki kadar yeni görüntü (özellik haritası) elde edilmiş olur.

$$Özellik\ haritası\ sayısı = \left(\frac{(m - n)}{s} + 1 \right) \times \left(\frac{(m - n)}{s} + 1 \right) \times k \quad (2.2)$$

Girdi görüntüsünün kanal sayısı ile filtrenin kanal sayısının eşit olmasına dikkat edilmelidir. Eğer üç kanallı bir görüntüye evrişim işlemi uygulanacaksa, filtre de üç boyutlu olmalıdır. Denklem 2.1'deki işlem üç kanal için gerçekleştirildiğinde, toplam adım sayısı üç katına çıkar. Evrişim sürecindeki bir diğer parametre dış boşluk sayısını ifade eden $padding$ parametresidir. Denklem 2.2'ye bakıldığında çıktı boyutu girdi boyutuna göre küçülmektedir. Bu parametre ile girdi matrisinin dışına 0 eklenir. Burada amaç, çıktı boyutunun girdi boyutu ile aynı kalmasını sağlamaktır. Eğer

$padding = 1$ olarak ayarlanırsa, girdi matrisinin her bir satır ve sütununa 0 eklenir. Şekil 2.5’de $padding = 1$ olarak ayarlandığında girdi matrisinin dışına 0 eklenme işlemi ve elde edilen özellik haritası boyutunun girdi boyutu ile aynı elde edildiği gösterilmiştir.



Şekil 2.5: Evrişim işlemi için 0 (sıfır) boşluk (zero padding) işlemi [34]

2.3. ORTAKLAMA (ALT ÖRNEKLEME) KATMANI

Ortaklama katmanı, ağdaki parametre ve hesaplama sayısını azaltmak için girdi görüntüsünün uzamsal boyutunu kademeli olarak düşürür. Burada asıl amaç hesaplama süresini düşürmek ve daha düşük donanıma sahip bilgisayarlarla model eğitimi gerçekleştirebilmektir. Ancak ortaklama işlemi, özellik çıkarma işleminde bilgi kaybına neden olur. Renkli görüntüler için her bir renk kanalında bağımsız olarak gerçekleşir. Ortaklama işleminin yaygın olarak kullanılan maksimum ortaklama (maximum pooling) ve ortalama ortaklama (average pooling) gibi çeşitleri bulunmaktadır. En yaygın kullanılan şekli maksimum ortaklamadır.

Ortaklama işlemi, filtre boyutu olarak $kernel_size$, adım olarak $stride$ ve ortaklama yöntemi parametrelerine sahiptir. Maksimum ortaklamanın iki varyasyonu sıklıkla kullanılmaktadır: $stride = 2, kernel_size = 3 \times 3$ ve $stride = 2, kernel_size = 2 \times 2$. Evrişim katmanlarına benzer şekilde, filtreler girdi üzerinde gezerek çıktılar elde

ederler. Maksimum ortaklamada, filtrenin anlık olarak girdi üzerinde bulunduğu konumdaki değerlerden en büyük olanı alınırken, ortalama ortaklamada değerlerin aritmetiksel ortalaması alınır ve yeni özellik haritasının (çıkıtı) oluşmasında karşılık gelen ilgili bölgeye yerleştirilir. Maksimum ve ortalama ortaklama işlemlerinin gerçekleşme aşamaları $stride = 2$, $kernel_size = 3 \times 3$ parametre değerlerine göre Şekil 2.6'da gösterilmektedir.



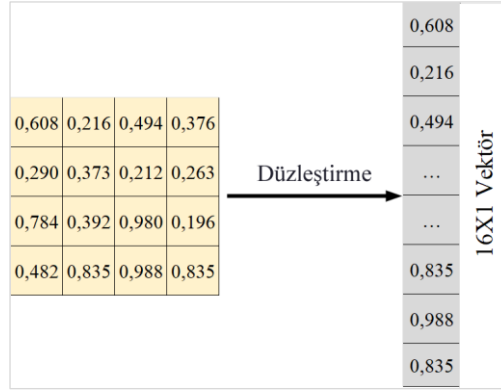
2.4. ÖZELLİK HARİTALARINI TAM BAĞLANTITILİ KATMANA GİRDİ OLARAK HAZIRLAMA

Evrişim sonunda elde edilen özellik haritaları matris formatında verilerden oluşmaktadır. Verilerin tam bağlantılı katmanlara (fully connected layers) iki veya üç boyutlu matris biçiminde değil, bir boyutlu vektör biçiminde verilmesi gerekmektedir. Vektörleştirme için yaygın olarak kullanılan katmanlar düzleştirme (flattening) ve global ortalama ortaklama (global average pooling - GOO) katmanlarıdır.

2.4.1. Düzleştirme Katmanı

Düzleştirme işlemi sonucunda, tam bağlantılı katmanlar için çok büyük bir girdi vektörü elde edilmiş olur. Son ortaklama katmanından sonra elde edilen özellik haritalarının $7 \times 7 \times 2048$ olduğu varsayıldığında, bu gösterim bir girdi görüntüsü için 7×7 boyutunda 2048 adet özellik haritasının çıkarıldığını ifade eder. Düzleştirme

işlemi sonucunda özellik haritası, $7 \times 7 \times 2048 = 100352$ elemandan oluşan bir vektöre dönüştürülmüş olur. Bu sayı girdi boyutuna ve oluşturulan ESA modeline göre değişkenlik gösterir. Şekil 2.7’de iki boyutlu 4×4 ’lük bir özellik haritasının bir boyutlu vektör biçiminde düzleştirilmiş biçimini gösterilmektedir.



Şekil 2.7: Düzleştirme işlemi

2.4.2. Genel Ortalama Ortaklama Katmanı

Geleneksel ESA’lar, ağın alt katmanlarında evrişim gerçekleştirir. Sınıflandırma için, son evrişim katmanının özellik haritaları vektörleştirilir ve varsa tam bağlantılı katmanlara beslenir, ardından bir softmax sınıflandırıcı katman gelir. Bu yapı, geleneksel sinir ağı sınıflandırıcıları ile evrişimsel yapı arasında köprü kurar. Evrişim katmanlarını öznitelik çıkarıcılar olarak ele alır ve elde edilen öznitelikler geleneksel bir şekilde sınıflandırılır. Tam bağlantılı katmanlar aşırı uydurmaya eğilimlidir. Bu durum, ağın genelleme kabiliyetini engeller.

GOO katmanı, sınıflandırma görevinin karşılık gelen her bir kategorisi için, son evrişim katmanının devamında özellik haritası oluşturmaktadır. Özellik haritalarının üzerine tam bağlı katmanlar eklemek yerine, her bir özellik haritasının ortalaması alınır ve ortaya çıkan vektör doğrudan softmax sınıflandırıcı katmana bağlanır. Tam bağlantılı katmanlara göre GOO’nun bir avantajı, özellik haritaları ve kategoriler arasındaki uygunluk durumunu güçlendirerek evrişim yapısına daha özgü bir yöntem olmasıdır. Böylece özellik haritalarına olan güven artmaktadır. Bu

kullanım şeklinin dışında, ESA modellerinin eğitiminde GOO katmanından sonra opsiyonel olarak tam bağlantılı katmanlar kullanılabilir. GOO'nun diğer bir avantajı, bu katmanda optimize edilecek bir parametrenin olmamasıdır, bu durum aşırı uyumdan kaçınmayı sağlar. Ayrıca uzamsal bilgiyi özetler, bu nedenle girdinin uzamsal dönüşümleri için daha sağlamdır. GOO, özellik haritalarını kategorilerin güven haritaları olmaya açıkça zorlayan yapısal bir düzenleyici olarak görülebilir [38]. GOO'nun tam bağlı katmanlarla birlikte kullanıldığı ağ modellerinde, genel olarak tam bağlı katmanlardan önce bir kez uygulanır ve avantajları şunlardır [39]:

- (1) Öğrenilebilir parametrelerin sayısını azaltır,
- (2) ESA'nın değişken büyüklükteki girdileri kabul etmesini sağlar.

Gösterge olarak, n örnek sayısı, w genişlik, h yükseklik ve m özellik haritası sayısı olmak üzere, düzleştirme katmanı (n, w, h, m) şekline sahip tensörü girdi olarak alır ve bunu tek boyutlu vektöre dönüştürür. Toplam parametre sayısı, Denklem 2.3' deki gibi ifade edilir.

$$\text{Toplam parametre sayısı} = n, w \times h \times m \quad (2.3)$$

Böyle bir kullanım şekli, eğitim veri kümesine aşırı uydurma riski taşır. Aşırı uydurmayı önlemek için tam bağlı katmanlardan sonra bırakma katmanları (bkz. Bölüm 2.5) kullanılır. GOO katmanı, farklı olarak her bir uzamsal boyut 1 olana kadar, uzamsal boyutlar üzerinde ortalama ortaklama uygular ve diğer boyutları değiştirmeden bırakır. GOO ile toplam parametre sayısı, Denklem 2.4' deki gibi ifade edilir.

$$\text{Toplam parametre sayısı} = n, 1 \times 1 \times m \quad (2.4)$$

Her iki yöntemin karşılaştırılması amacıyla, örneğin Xception [40] ESA modelinin girdi şekli $224 \times 224 \times 3$ olarak alındığında, bir softmax sınıflandırıcısından önce son evrişim katmanı düzleştirildiğinde 100352 parametre, GOO yöntemi kullanıldığında ise yalnızca 2048 parametre elde edilmektedir. Karşılaştırma detayı Şekil 2.8'de gösterilmektedir. Sağladığı birçok avantaj nedeniyle bu tez çalışmasında oluşturulan ağ modellerinde GOO katmanının kullanımı tercih edilmiştir.

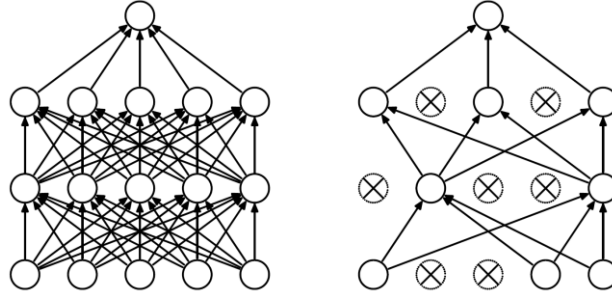
block14_sepconv2 (SeparableConv (None, 7, 7, 2048))	3159552	block14_sepconv1_act[0][0]
block14_sepconv2_bn (BatchNorm (None, 7, 7, 2048))	8192	block14_sepconv2[0][0]
block14_sepconv2_act (Activation (None, 7, 7, 2048))	0	block14_sepconv2_bn[0][0]
=====		
Total params: 20,861,480		
Trainable params: 20,806,952		
Non-trainable params: 54,528		
=====		
<pre> 1 x = base_model.layers[-1].output # en son katmanın çıkışı 2 # Flatten 3 flat = tf.keras.layers.Flatten() 4 print(flat(x).shape) 5 6 # Global Average Pooling 7 gap = tf.keras.layers.GlobalAveragePooling2D() 8 print(gap(x).shape) 9 </pre>		
(None, 100352)		
(None, 2048)		

Şekil 2.8: Xception modelinde Düzleştirme ve GOO karşılaştırması

2.5. BIRAKMA KATMANI

Aşırı uydurma (overfitting), ESA'larda önemli bir sorundur. Bir ESA modeli, eğitim verilerindeki ayrıntıları ve gürültüyü, modelin yeni veriler üzerindeki performansını olumsuz yönde etkileyecek ölçüde öğrendiğinde, aşırı uydurma gerçekleşir. Bu, eğitim verilerindeki gürültü veya rastgele dalgalanmaların model tarafından kavram veya etiket olarak alındığı ve öğrenildiği anlamına gelir. Buradaki sorun, bu kavramların yeni verilere uygulanmaması ve modellerin genelleme yeteneğini olumsuz etkilemesidir. Aşırı uydurma sorunu, sınıflardaki örnek sayılarının birbirinden çok farklı olduğu veya bazı sınıflara ait örnek sayılarının çok yetersiz olduğu dengesiz veri setlerinin eğitiminde sıklıkla görülür. Modelin eğitimi sırasında, eğitim başarı oranı yükseliyor ve eğitim hata oranı düşüyorken, doğrulama başarı oranı yükselmiyor ve doğrulama hata oranı aşırı yükselmeye başlamışsa, modelin aşırı uydurma sürecine girmiş olduğu anlaşılır.

Bırakma (dropout), aşırı uydurma sorunu ile mücadele etmenin yöntemlerinden biridir. Bir sinir ağındaki bazı düğümler rastgele devre dışı bırakılır. Bir düğüm devre dışı bırakıldığı zaman gelen ve giden bağlantılarıyla birlikte geçici olarak ağdan çıkarılır [41]. Şekil 2.9'da bir YSA'nın standart ve bırakma uygulandıktan sonraki durumları gösterilmektedir.



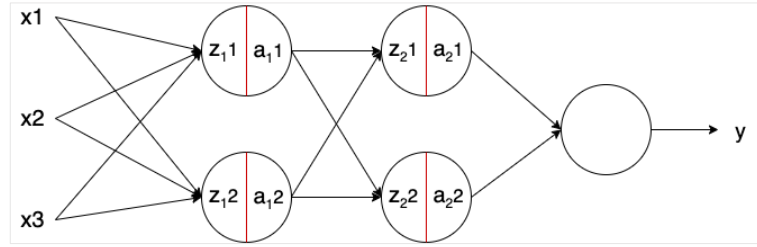
Şekil 2.9: Bırakma işleminden önce (solda) ve sonra (sağda) YSA [41]

Bırakma oranı genellikle 0.5 olarak alınır ve bırakılan düğümler ileri geçiş katkıda bulunmazlar ve geri yayılıma katılmazlar. Her bir bırakma işlemi, yakınsama için gereken yineleme sayısını yaklaşık olarak iki katına çıkartmaktadır [7]. Bırakma katmanı, oluşturulan sinir ağı modellerinde çoğunlukla tam bağlantılı katmanlarla birlikte kullanılır. Aşırı uydurmayı önlemeye yönelik olarak Bölüm 2.4.2’de açıklanan GOO katmanı kullanılabileceği gibi, bırakma (dropout) katmanı da kullanılabilir. Oluşturulan modelde GOO katmanının kullanımı tercih edilmişse, iyi bir model öğrenmesi için gizli katman kullanımına olan ihtiyaç kısmen ortadan kalkabileceğinden dolayı, bırakma katmanı eklenmesine olan gereksinimde azalmaktadır. Bu, bırakmadan kaynaklı bilgi kaybının önlenmesini sağlar. Bu tez çalışması kapsamında bırakma katmanının kullanılması tercih edilmemiştir.

2.6. TOPLU NORMALLEŞTİRME

Normalleştirme (normalization), sayısal verileri şeklini bozmadan ortak bir ölçüğe getirmek için kullanılan bir veri ön işleme aracıdır. Bir veri kümesinde tüm özellikler aynı aralıkta olmayabilir. Normalleştirme, tüm girdilerin dağılımını *ortalama = 0* ve *standart sapma = 1* olacak şekilde dönüştürür. Böylece değerlerin çoğu -1 ile +1 aralığına getirilir. Normalleştirme işleminin ham verilere uygulanması işleminde, RGB görüntüleri girdi katmanına verilmeden önce 255’e bölünerek normalize edilir ve her bir kanalın her bir piksel değeri [0-255] aralığından [0-1] aralığına indirgenmiş olur. Toplu normalleştirme (batch normalization), ham girdilere uygulanmak yerine ağırlık çeşitli katmanlarında kullanılır. Sinir ağlarının eğitimini daha hızlı ve kararlı hale getiren algoritmik bir yöntemdir.

Toplu normalleştirme, gelişmiş ESA modellerinde evrişim katmanlarında sıklıkla kullanılarak, model mimarilerinin bir parçası durumundadır. Bununla birlikte opsiyonel olarak çok katmanlı standart YSA'ları oluşturun tam bağlantılı katmanlarda da kullanılabilir. Tam bağlantılı katmanlarda düğümlerin çıkışında AF'den önce uygulanır. Şekil 2.10'da TN'nin tam bağlantılı katmanlarda uygulanışı gösterilmektedir.



Şekil 2.10: Tam bağlantılı katmanlarda Toplu Normalleştirme

Şekil 2.10'da düğümlerin ortasındaki kırmızı çizgiler TN'nin uygulandığı konumu göstermektedir. Burada x_i girdileri, z düğümlerin çıktılarını, a AF'nin çıktılarını ve y ağın çıktısını göstermek üzere düğüm ve TN çıktıları ve AF girdi hesapları Denklem 2.5 - 2.7'deki gibi yapılır.

$$z = g(w, x) \quad (2.5)$$

$$z^N = \left(\frac{z - m_z}{s_z} \right) \cdot \gamma + \beta \quad (2.6)$$

$$a = f(z^N) \quad (2.7)$$

Denklem 2.5' de, g düğümlerin lineer dönüşümünü, w ağırlıkları, Denklem 2.6' da z^N TN çıktısını, m_z düğümlerin çıktılarının ortalamasını ve s_z standart sapmasını, γ ve β TN'nin öğrenme parametrelerini, Denklem 2.7' de f AF'nu gösterir. Beta ve gama parametreleri sırasıyla ortalama ve standart sapmayı kaydırır. Böylece, toplu normalleştirmenin bir katman üzerindeki çıktıları, ortalama beta ve standart gama sapması olan bir dağılımla sonuçlanır. Bu değerler, düğümlerin ağırlıkları gibi diğer

öğrenme parametreleri ve eğitim dönemleri (epochs) ve üzerinden öğrenilerek model kaybının azaltılması hedeflenir.

TN'nin evrişim katmanlarında kullanımında ise, evrişim katmanlarından sonra ve aktivasyon katmanlarından önce kullanımı tercih edilir. Şekil 2.11'de Xception ve DenseNet201 ESA modellerinin orijinal durumundaki katman yerleşimlerinden birer kesit gösterilmektedir. Her iki modelde de görüldüğü üzere evrişim katmanlarından sonra TN ve devamında aktivasyon (ReLU) katmanı bulunmaktadır.

input_3 (InputLayer)	[(None, 224, 224, 3) 0
block1_conv1 (Conv2D)	(None, 111, 111, 32) 864
block1_conv1_bn (BatchNormaliza	(None, 111, 111, 32) 128
block1_conv1_act (Activation)	(None, 111, 111, 32) 0
block1_conv2 (Conv2D)	(None, 109, 109, 64) 18432
block1_conv2_bn (BatchNormaliza	(None, 109, 109, 64) 256
block1_conv2_act (Activation)	(None, 109, 109, 64) 0
block2_sepconv1 (SeparableConv2	(None, 109, 109, 128) 8768
block2_sepconv1_bn (BatchNormal	(None, 109, 109, 128) 512
block2_sepconv2_act (Activation	(None, 109, 109, 128) 0

a) Xception

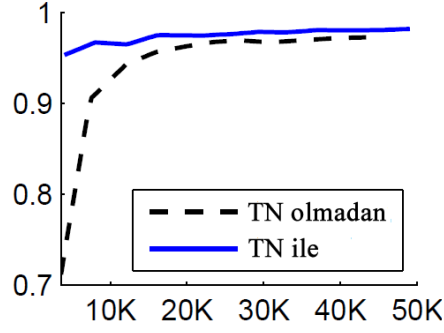
conv5_block30_1_conv (Conv2D)	(None, 7, 7, 128) 233472
conv5_block30_1_bn (BatchNormal	(None, 7, 7, 128) 512
conv5_block30_1_relu (Activatio	(None, 7, 7, 128) 0
conv5_block30_2_conv (Conv2D)	(None, 7, 7, 32) 36864
conv5_block30_concat (Concatena	(None, 7, 7, 1856) 0
conv5_block31_0_bn (BatchNormal	(None, 7, 7, 1856) 7424
conv5_block31_0_relu (Activatio	(None, 7, 7, 1856) 0
conv5_block31_1_conv (Conv2D)	(None, 7, 7, 128) 237568

b) DenseNet201

Şekil 2.11: Xception ve DenseNet201 modellerinde Toplu Normalleştirme kullanımı

Derin Sinir ağların eğitimi sırasında, ağ parametrelerindeki değişiklik nedeniyle ağ aktivasyonlarının dağılımında değişiklikler meydana gelir. Bu sorun dahili ortak değişken kayması (internal covariate shift) olarak adlandırılmakta ve daha düşük öğrenme oranları ve daha dikkatli parametre ayarı ile başlatma gerektirmekte, ayrıca eğitimi yavaşlatmakta ve zorlaştırmaktadır. Eğitim sırasında her katman, ileri yayılım aşamasında oluşan hata için kendini düzeltmeye çalışır. Ancak bu işlem için her katman ayrı ayrı hareket eder. Bir önceki katman hata düzeltmesi yaptıktan sonra, bir sonraki katman bu düzeltmeye göre yeni çıktılar üretmelidir. Bu, bir önceki katmandan sonra bir katmanın öğrenmeye başlaması problemini sunar, yani 3. katman 2. katman bitirdikten sonra, 4. katman 3. katman bitirdikten sonra vb. öğrenmeye başlar. Sonuç olarak, çok fazla katmana sahip mevcut derin sinir ağlarında eğitme süreci çok uzun zaman alır [42]. Bahsedilen dahili ortak değişken kaymasını azaltmak için toplu normalleştirme kullanılır. TN, çok daha yüksek öğrenme oranları

kullanılmasına ve başlatma ayarlarının daha kolay yapılmasına olanak tanır [43]. Ayrıca bazı durumlarda bırakma (dropout) ihtiyacını ortadan kaldıran bir düzenleyici görevi görür [41]. TN'nin bir ağıın eğitimi üzerindeki hız ve test başarısı yönünden etkisini Şekil 2.12'de görebiliriz.



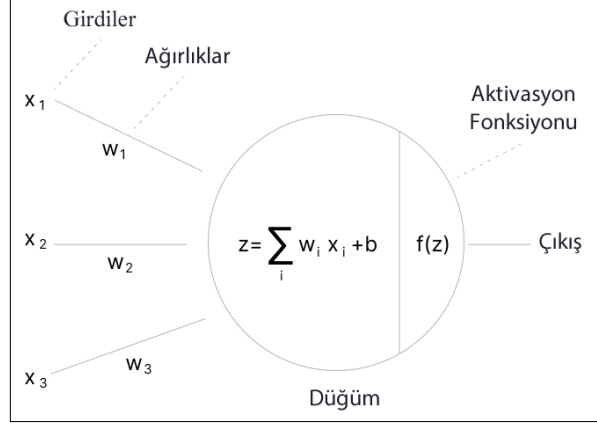
Şekil 2.12: TN ile ve TN olmadan MNIST veri seti deneyi [43]

Burada MNIST [44] el yazısı karakter veri seti ile TN kullanılarak ve TN kullanılmadan bir ağıın eğitimi gerçekleştirilmiştir. Şekle bakıldığında, TN ile yapılan deneyde modelin daha kısa sürede kısmen daha yüksek test doğruluk oranına ulaştığı görülmektedir. Tez kapsamında kullanılan ESA modellerinin evrişim bloklarında TN katmanları bulunmaktadır. Bu kullanımlara ilave olarak ek bir TN katmanı kullanılmamıştır.

2.7. AKTİVASYON FONKSİYONLARI

YSA'larının görüntü, video, ses ve yazı gibi doğrusal olmayan, karmaşık bilgilerden öğrenmesi istendiğinde aktivasyon fonksiyonlarına ihtiyaç duyulmaktadır. AF, gerekli dönüşümü yaparak daha karmaşık görevlerin öğrenilebilmesini sağlar. AF, ağırlıklı toplamı hesaplayarak ve buna eşik değeri (bias: aktivasyon fonksiyonunun sağa veya sola ötelenmesini sağlar, aynı zamanda aşırı uydurmayı önlemek adına hesaplanan çıktıya eklenen sabit veya rastgele bir değerdir [36]) ekleyerek bir düğümün aktive edilip edilmeyeceğine karar verir. Ayrıca, bir düğümün çıktısının doğrusal olmayan yapıya dönüştürülmesini sağlar. AF olmayan bir YSA, esasen sadece bir doğrusal regresyon modeli olarak ifade edilebilir. Girdi değerleri x ,

ağırlıklar w , yanlılık değeri b ve çıkış değeri z olmak üzere, bir düğümdeki AF'nun çıkış oluşturma işlevi Şekil 2.13'de gösterilmektedir.



Şekil 2.13: Düğüm yapısı ve aktivasyon fonksiyonunun işlevi [45]

Aktivasyon fonksiyonlarının bir diğer önemli özelliği türevlenebilir olmasıdır [37]. Elde edilen çıktı değeri ile gerçek değer arasındaki farkı ifade eden hata değerinin çok büyük olması, modelin öğrenme düzeyinin düşük olduğunu gösterir. Bu nedenle, hata değerinin düşürülmesi için w ve b değerlerinin güncellenmesi gerekir. Bunun için uygulanan yöntemlerden birisi geri yayılım (back-propagation) algoritmasıdır. Aktivasyon fonksiyonlarının türevlenebilir olması, ağda geriye doğru yayılırken w değerlerine göre hata gradyanlarını hesaplamak, en iyilemeyi gerçekleştirmek ve ardından hatayı azaltmak için gradyan inişi (gradient descent) veya başka herhangi bir optimizasyon algoritması kullanarak w değerlerini güncellemeyi sağlar. Literatürde çok sayıda AF yer almaktadır. Bu fonksiyonların birçoğu ve bunların matematiksel ifadeleri, Tablo 2.1'de gösterilmektedir.

Tablo 2.1: Aktivasyon fonksiyonları ve matematiksel ifadeleri [45]–[47]

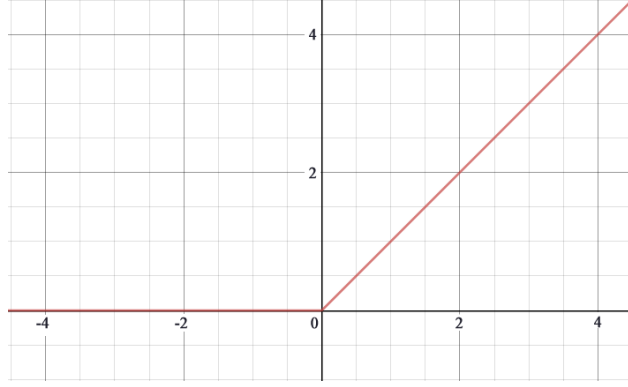
Aktivasyon Fonksiyonu	Matematiksel İfade
Binary Step (İkili Basamak)	$f(x) = \begin{cases} 0, & x < 0 \\ 1, & x \geq 0 \end{cases}$
Lineer (Doğrusal)	$f(x) = x$
Sigmoid / Logistic	$f(x) = \frac{1}{1 + e^{-x}}$
Tanh (Hyperbolic Tangent – Hiperbolik Tanjant))	$f(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$
ReLU (Rectified Linear Unit – Doğrultulmuş Doğrusal Birimi)	$f(x) = \begin{cases} \max(0, x), & x \geq 0 \\ 0, & x < 0 \end{cases}$
Leaky ReLU (Sızdıran ReLU)	$f(x) = \max(0.1x, x)$
ELU (Exponential Linear Units – Üstel Doğrusal Birimler)	$f(x) = \begin{cases} x, & x \geq 0 \\ \alpha(e^x - 1), & x < 0 \end{cases}$
Parametric ReLU (Parametrik ReLU)	$f(x) = \max(ax, x)$
Softplus	$f(x) = \log(1 + \exp^x)$
Softmax	$f(x_i) = \frac{\exp(x_i)}{\sum_{j=1}^n \exp(x_j)}$

Aktivasyon fonksiyonlarının seçimi çözülmekte olan problem türüne göre olmalıdır. Sigmoid AF, ikili sınıflandırma problemlerinde sıklıkla kullanılmaktadır [36][32]. Ancak kaybolan gradyan sorunu ile karşılaşılabilir. Softmax AF, en yüksek olasılığa sahip hedef sınıf ile birlikte her sınıfın olasılık değerlerini döndürdüğü için çok sınıflı problemlerde, neredeyse tüm DÖ modellerinin çıktı katmanlarında kullanılır [47]. Tanh AF Sigmoid'e göre daha iyi eğitim performansı sağlamasına rağmen, kaybolan gradyan sorununu çözememiştir [36][47]. Softplus AF, derin sinir ağlarının stabilizasyonunu ve performansını artırarak, yumuşatma ve sıfır olmayan gradyan özelliklerine sahip ReLU AF'nun yumuşak bir versiyondur. Teorik olarak ReLU'ya göre avantajlı olması beklenebilir ancak deneysel olarak değildir. ReLU'nun Softplus ve Tanh AF'na göre daha iyi sonuçlar verdiği [48], Sigmoid ile karşılaştırıldığında daha az sürede (eğitim dönem sayısı) daha iyi performans gösterdiği görülmüştür [49].

Avantajlarından dolayı tez kapsamında kullanılan ESA modellerinin çeşitli katmanlarında kullanılan AF ReLU olarak bırakılmış ve değiştirilmemiştir.

2.7.1. ReLU Aktivasyon Fonksiyonu

ReLU AF, yakınsamayı çok daha hızlı gerçekleştirmesi nedeniyle, günümüzde neredeyse tüm DÖ modelleri tarafından en çok tercih edilen AF'dur [37]. Tez kapsamında kullanılan ESA modellerinin evrişim katmanlarında çok sayıda ReLU AF kullanılmaktadır. Bu fonksiyonun grafiği Şekil 2.14'deki gibidir.



Şekil 2.14: ReLU (Düzeltilmiş Doğrusal Birim) aktivasyon fonksiyonu [50]

Tablo 2.1'de görüldüğü gibi ReLU, $x < 0$ olduğunda 0 sonucunu verirken, $x \geq 0$ olduğunda ise doğrusal bir fonksiyon üretir. ReLU yalnızca belirli sayıda düğümü etkinleştirdiğinden hesaplama açısından verimlidir. Ayrıca kayıp fonksiyonunun minimuma doğru gradyan inişinin yakınsamasını hızlandırır. ReLU, ESA'ların tam bağlantılı ve evrişim katmanlarında etkinleştirme fonksiyonu olarak kullanılabilir gibi, ağın son katmanında sınıflandırma fonksiyonu olarak da kullanılabilir [50].

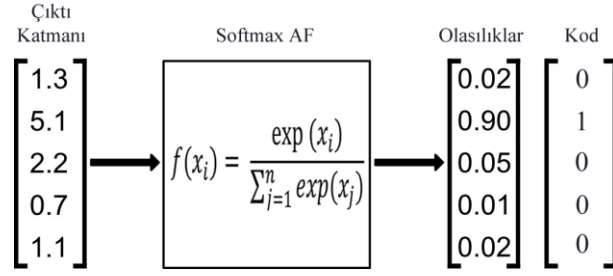
2.7.2. Softmax Aktivasyon Fonksiyonu

Çok sınıflı sınıflandırma işlemlerinde, reel sayılar vektöründen olasılık dağılımını hesaplamak için kullanılır. En yüksek olasılığa sahip hedef sınıf ile her bir sınıfın olasılıklarını döndürdüğü çok sınıflı modellerde kullanılır. Olasılıkların toplamı

1'e eşit olan 0 ile 1 değer aralığında değerlerden oluşan bir çıktı üretir. Neredeyse tüm son teknoloji (state of art) derin öğrenme mimarilerinin çıktı katmanlarında kullanılır [38], [51]–[55]. Bu tez çalışması kapsamında da sınıflandırma için çıktı katmanında softmax kullanımı tercih edildi.

Softmax, çoklu sigmoidlerin bir kombinasyonu olarak da ifade edilebilir. Sigmoid AF kolay anlaşılır olmakla birlikte, çoğunlukla sığ ağlarda, ikili sınıflandırma problemlerinde, olasılığa dayalı çıktıyı tahmin etmek için başarıyla uygulanır. Örneğin kalem içeren bir girdi görüntüsünün yüzde kaç olasılıkla kalem olarak tanımlanacağını tahmin etmesi isteniyorsa sigmoid kullanılabilir. Böyle bir durumda sigmoid AF, [0,1] aralığında bir değer üretmiş olacak ve üretilen değer girdinin kalem olma ve olmama olasılığını ifade edecektir. Sigmoid AF'nun belirgin bazı olumsuzlukları bulunmaktadır. Bunlar, x 'in bazı bölgeler dışında kalan değerleri için y 'nin çok küçük tepkiler vermesi ve 0'a yakınsaması nedeniyle kaybolan gradyan sorunu ve yavaş yakınsama gibi sorunlarla karşılaşmaktadır [47].

Softmax AF her sınıfın olasılığını döndürür. Çıktı katmanı üç nörondan oluşacak şekilde oluşturulmuş bir DÖ ağ modeli için, nöronların çıktı değerlerinin [1.8, 0.9, 0.68] olduğu varsayılırsa, nöronlara softmax fonksiyonu uygulandığında [0.58, 0.23, 0.19] sonucu elde edilecektir. Üç nöronun olasılık değerlerinin toplamının 1 olduğu görülmektedir. Burada elde edilen olasılık vektörü 1 ve 0 olarak kodlandığında, vektörün en büyük olasılık değeri için 1, diğer olasılık değerleri için 0 döndürür ([1 0 0]). Dolayısıyla [0.58, 0.23, 0.19] olasılık vektörü için, indeks 0'daki değere tam ağırlık verilir ve indeks 1 ve indeks 2'deki değerlere ağırlık verilmez. Bu durumda çıktı tanımı, indeks 0'daki değere karşılık gelen sınıf olacaktır. Bu sınıf bir araba, ev, çiçek, hastalık vb. türüne ait olabilir. Yine değerleri rastgele oluşturulmuş beş nöronlu çıkış için grafiksel olarak gösterilirse, softmax AF'nun çıktı nöronlarından ürettiği olasılık vektörü ve kodlanmış hali Şekil 2.15'de gösterilmektedir.



Şekil 2.15: Softmax aktivasyon fonksiyonu ve olasılık vektörü [56]

2.8. İLERİ YAYILIM İŞLEMİ

Bir x girdi vektöründen/matrisinden YSA'nın tahmini sonucu elde edilen \hat{y} çıktısının elde edilmesinde, akış girdi katmanından çıkış katmanına doğru hareket eder. Bu işlem süreci ileri yayılım (forward-propagation) olarak tanımlanır. Standart YSA'larda x girdisi ile ağa gönderilen bilgi, gizli katmanlardaki düğümlere ve oradan çıkış katmanına doğru yayılır. Örneğin bir görselin piksel değerlerinin normalize edilmiş vektörü ağa girdi olarak verildiğinde, ağırlık değerleri (Şekil 2.2'de gösterilen oklar) ile çarpılarak gizli katmanlardaki düğümlerin girdileri elde edilir. Ağırlık değerleri rastgele başlatılacağı gibi, ön eğitilmiş bir modelin ağırlık değerleri ile de başlatılabilir. Düğümlerin girdi değerleri, seçilen bir AF'na girdi olarak verilir ve düğümün çıktısı elde edilir. Daha sonra bu çıktı varsa diğer ağırlık değerleri ile çarpılır ve bir sonraki gizli katmandaki bir düğümün girdi değeri elde edilir. Süreç bu şekilde YSA'ndaki tüm gizli katman düğümlerinden geçerek devam eder. Özellik sayısı n olan bir x girdisi için ($i \in 1...n$), bir düğümün çıktı değeri (bkz. Şekil 2.13) Denklem 2.8'deki gibi hesaplanır.

$$z = \sum_{i=1}^n x_i w_i + b, \quad a = f(z) \quad (2.8)$$

Burada, f AF'yi, b eşik değerini (bias), w ağırlıkları gösterir. Ağırlık değerleri rastgele başlatılabileceği gibi farklı yöntemler kullanılarak da başlatılabilir.

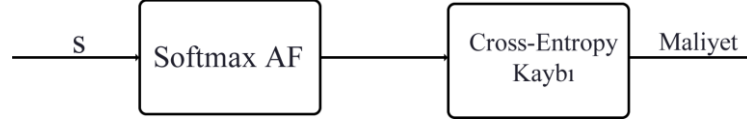
ESA'larda ise standart YSA'lardan farklı olarak, x girdisi öncelikle evrişim katmanlarından geçer ve özellik haritaları oluşturulur. Standart YSA'larda ağırlık değerlerini girdi katmanı ve gizli katmanlardaki düğümler arası bağlantılar ile gizli katmanlardaki düğümler ve çıktı katmanı arasındaki bağlantılar oluştururken, ESA'larda evrişim katmanlarındaki ağırlık değerlerini oluşturulan özellik haritaları (matrisler) ve özellik haritalarının oluşturulmasını sağlayan filtre matrislerinin değerleri oluşturur (bkz. bölüm 2.2 ve 2.3). Filtrelerin değerleri başlangıçta rastgele veya farklı yöntemler kullanılarak da oluşturulabilir.

2.9. MALİYET (KAYIP) FONKSİYONLARI

Kayıp ya da hata fonksiyonu olarak da adlandırılan maliyet fonksiyonları, ileri yayılım yoluyla ESA veya standart YSA'ların tahmin değerleri ile temel gerçek etiketleri arasındaki uyumluluğu ölçer [39]. Maliyetin ölçülmesinin nedeni, tahminin gerçek değere ne kadar yaklaştığının gözlemlenebilmesidir. Tüm hata değerlerinin ortalaması maliyeti oluşturur. Problemin tipine bağlı olarak kullanılacak çok sayıda maliyet fonksiyonu bulunmaktadır. Problem tipleri genel olarak regresyon, ikili sınıflandırma ve çok sınıflı sınıflandırma olabilir. Çok sınıflı sınıflandırma problemlerinde en yaygın kullanılan maliyet fonksiyonu, çok sınıflı çapraz entropi (Multi-class Cross-entropy Loss) fonksiyonudur. Bunun dışında diğer problemlerde kullanılan, ortalama kare hatası (Mean Squared Error Loss), ortalama mutlak hata (Mean Absolute Error Loss), ikili çapraz entropi (Binary Cross-Entropy Loss) gibi fonksiyonlar maliyet fonksiyonlarından bir kaçıdır. Bu tez çalışmasında çok sınıflı sınıflandırma problemi gerçekleştirildiği için çok sınıflı çapraz entropi maliyet fonksiyonu kullanılmaktadır.

Çok sınıflı (kategorik) çapraz entropi maliyet fonksiyonu, softmax kaybı olarak da adlandırılır. Çünkü bu fonksiyon bir softmax AF ve bir çapraz-entropi kaybının birleşimidir. Çok sınıflı sınıflandırma problemlerinde, yapay sinir ağ modellerinin çıktı katmanında oluşan değerler kullanılarak softmax AF ile her bir sınıf için $[0,1]$ aralığında olasılık değerleri üretilir. Evrişim katmanlarından sonra, GOO katmanı ile elde edilen vektörler S olmak üzere, maliyet değeri hesaplanmadan önce bir AF ile S değerine dönüşüm uygulanır. Böylece her bir i sınıfı için a_i aktivasyon değeri elde

edilir. Her bir a_i aktivasyon değerinin olasılık vektöründeki karşılığı softmax AF ile elde edilir ve elde edilen olasılık değerleri çapraz entropi yönteminde kullanılarak maliyet hesaplanır. Softmax AF ve çapraz entropi kaybı arasındaki ilişki Şekil 2.16'daki gibi gösterilebilir.



Şekil 2.16: Softmax AF ve Cross-Entropy maliyet fonksiyonu ilişkisi

Softmax AF'nun gelen aktivasyon değerlerine göre sınıfsal olasılık hesabı Denklem 2.9' daki gibi hesaplanır.

$$p_i = \frac{\exp(a_i)}{\sum_{j=1}^c \exp(a_j)} \quad (2.9)$$

Burada c veri setindeki sınıf sayısı, a_i i sınıfı için aktivasyon değeri ve p_i ise i sınıfı için softmax ile elde edilen olasılık değeridir. Her bir sınıf için elde edilen olasılık değerleri olasılık vektörünü yani modelin tahmin ederi \hat{y}_i vektörünü ifade eder. Gerçek değer ise y_i ile gösterildiğinde, bir girdi için kategorik çapraz entropi maliyet hesabı Denklem 2.10' daki gibi yapılır [57].

$$L(\hat{y}, y) = -y_i \log(\hat{y}_i) - (1 - y_i) \log(1 - \hat{y}_i) \quad (2.10)$$

Buradaki hesaplama bir girdi için maliyeti gösterir. Tüm örnekler için ortalama maliyet hesabı Denklem 2.11'deki gibi hesaplanır.

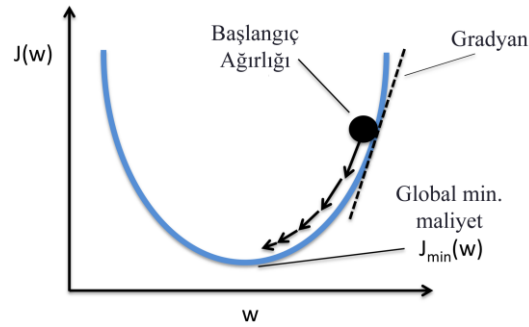
$$J(W, b) = -\frac{1}{N} \sum_{i=1}^N \sum_{j=1}^C L(\hat{y}_j, y_j) \quad (2.11)$$

Bu denklemde, N örnek sayısını, C özellik (sınıf) sayısını ve J ortalama maliyeti, W ağırlık matrisini ve b eşik vektörünü (bias) gösterir. Maliyet hesabı yapılırken başka aktivasyon ve maliyet fonksiyonları birlikte kullanılabilir.

2.10. OPTİMİZASYON İŞLEMLERİ

ESA'lar oluşturulduktan sonra modelin eğitimine başlamadan önce yapılması gereken en önemli işlemlerden birisi optimizasyon yönteminin seçilmesidir. Optimizasyon işlemi, modelin eğitimi sırasında oluşabilecek hataları en düşük seviyeye indirmek için kullanılmaktadır [58]. Bir ağın nasıl öğrendiğinin anahtarıdır ve minimum hata noktasını bulan süreci ifade eder [37]. Optimizasyon yöntemlerinin temelinde gradyan inişi (gradient descent) yöntemi bulunmaktadır. Gradyan inişi, maliyet fonksiyonunun değerini en aza indirgeyerek minimum hatayı ve en yüksek doğruluğu bulmak için kullanılır.

Derin öğrenmede ağ parametrelerinin güncellenmesi geri yayılım işlemi ile yapılmaktadır. Geri yayılım işleminde, geriye doğru türev alınarak farkın bulunması ve bulunan fark değerinin öğrenme oranı (*learning rate*) parametresiyle çarpılması, elde edilen sonucun ağırlık değerlerinden çıkarılarak yeni ağırlık değerinin hesaplanmasıyla yapılmaktadır. Bu işlem sürecine zincir kuralı da denilmektedir. Bu işlem esnasında kullanılan *learning rate* (öğrenme oranı/hızı) parametresi sabit, adım adım artan (örneğin belli bir öğrenme adımına kadar 0.001, daha sonra 0.01 gibi) bir değer olarak veya *momentum* (gradyan inişini hızlandıran ve salınımları azaltan parametre) değerine bağlı olarak belirlenebileceği gibi, adaptif algoritmalar tarafından ağın eğitimi sırasında öğrenilebilir. Şekil 2.17'de bir gradyan iniş algoritmasının yakınsaması gösterilmektedir.



Şekil 2.17: Gradyan iniş algoritmasının yakınsaması [59]

Burada, J maliyet fonksiyonu, w seçilen herhangi bir başlangıç noktası olmak üzere, $J(w)$ değerini minimuma yaklaştırmak için her adımda w değerinin güncellenmesi

gerekir. En temel anlamda gradyan inişin matematiksel ifadesi Denklem 2.12'deki gibidir.

$$W_{yeni} = W_{eski} - \alpha \frac{\partial J(W_{eski})}{\partial W_{eski}} \quad (2.12)$$

Şekil 2.17'deki ilk gradyan değeri (kesikli çizgi), başlangıç noktasına çizilen teğetin eğimini ifade eder. Bu aynı zamanda o noktadaki türevine eşittir. Katsayı α ise öğrenme katsayısını gösterir. Şekilde w küçülürken $J(w)$ 'de küçülmektedir. Bu yüzden noktanın eğimi pozitiftir. Eğimi pozitif olan noktanın türevi de pozitif olacağından dolayı, $\partial J / \partial W_{eski}$ pozitif değerinin α 'nın başındaki negatif (-) işareti ile çarpımı negatif olacağı için, yeni elde edilen w değeri mevcut w değerinden küçük olur. Bu da global ya da yerel minimuma yaklaşmayı sağlar. Eğer eğim negatif olursa, α 'nın başındaki negatif ile çarpımı pozitif olacağı için güncellenmiş olan yeni w ağırlık değeri mevcut w ağırlık değerinden büyük olur ve aynı şekilde minimum hata noktasına yaklaşılmış olur. Bu işlem sürekli tekrar ederek devam eder. Çok sayıda optimizasyon algoritması bulunmaktadır. Bunlardan en popüler olanları Adam, GD, SGD, Mini-batch GD, RMSProb, AdaGrad, AdaDelta, AdaMax, Nadam şeklinde sıralanabilir.

2.10.1. Adam Optimizasyon Algoritması

Adam (Adaptive Moment) optimizasyon algoritması, uygulama kolaylığı, hesaplama açısından verimlilik, büyük veri kümeleri ve yüksek boyutlu parametre uzayları ile makine öğrenimi problemlerine yönelik olması, az bellek gerektirmesi, durağan olmayan amaçlar ve çok gürültülü problemler için uygunluğu, hiper parametrelerin sezgisel olarak yorumlanabilmesi ve genellikle çok az ayarlama gerektirmesi, AdaGrad ve RMSProb'un yeteneklerini birleştirmesi gibi nedenlerden dolayı son yıllarda derin öğrenme modellerinde en sık tercih edilen optimizasyon algoritması konumundadır [60][61]. Bu avantajlarından dolayı bu tez kapsamında gerçekleştirilen deneysel çalışmalarda Adam optimizasyon algoritması kullanılmıştır. Adam optimizasyon algoritması birkaç parametre ile kontrol edilir. Bu parametrelerin Keras kütüphanesindeki isimleri ve varsayılan değerleri Tablo 2.2.'de gösterilmiştir.

Tablo 2.2: Adam optimizasyon algoritması parametreleri [62]

Parametre	Açıklama	Varsayılan Değer
$learning_rate (\alpha)$	Öğrenme oranı (ilk adım boyutu)	0.001
$beta_1 (\beta_1)$	Birinci an tahminleri için üstel azalma oranı. Değer aralığı: [0,1)	0.9
$beta_2 (\beta_2)$	İkinci an tahminleri için üstel azalma oranı. Değer aralığı: [0,1)	0.999
$epsilon (\epsilon)$	Sayısal kararlılık için küçük bir sabit	10^{-7}

Adam optimizasyon algoritmasının parametre güncellemesi adımları Denklem 2.13, 2.14, 2.15, 2.16 ve 2.17' deki gibidir [60].

$$m_t = \beta_1 m_{t-1} + (1 - \beta_1) g_t \quad (2.13)$$

Burada m_t , geçmiş gradyan değerlerinin üssel olarak azalan bir ortalaması, g_t ise t zaman adımıdaki gradyan değeridir.

$$v_t = \beta_2 v_{t-1} + (1 - \beta_2) g_t^2 \quad (2.14)$$

Burada v_t , geçmiş kare gradyan değerlerinin üssel olarak azalan bir ortalamasıdır. Bu değerlerin güncelleme kuralında kullanılmasıyla ilgili sorun, algoritma ilk çalıştırıldığında m ve v 0 vektörleri olarak başlatıldığı için 0'a eğimli olmaları sorunu vardır. Bu nedenle Adam optimizasyon Denklem 2.15 ve 2.16' daki yanlılık düzeltmesini getirmiştir.

$$\hat{m}_t = m_t / \beta_1^t \quad (2.15)$$

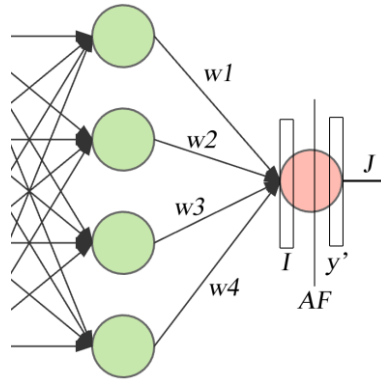
$$\hat{v}_t = v_t / \beta_2^t \quad (2.16)$$

Burada \hat{m}_t eşik (yanlılık, bias) düzeltmeli ilk an tahminini, \hat{v}_t ise eşik düzeltmeli ikinci ham anlık tahmini ifade eder. Son olarak parametre güncellemesi Denklem 2.17'deki gibi gerçekleşir.

$$W_t = W_{t-1} - \alpha \frac{\hat{m}_t}{\sqrt{\hat{v}_t}} + \varepsilon \quad (2.17)$$

2.10.2. Tam Bağlantılı Katmanlarda Geri Yayılım

Bir standart YSA'da veya ESA'da hesaplanan maliyet (hata, kayıp) değerine göre (bkz. Bölüm 2.8) gizli katmanlar arası veya evrişim katmanlarındaki ağırlıkların güncellenmesi için geri yayılım (back-propagation) yöntemi kullanılır. İleri yayılım işleminde ağırlıkların güncellenmesi yeni girdilerle gerçekleşirken, geri yayılımda ise ileri yayılım sonucu elde edilen hata değerine göre gerçekleşir. Farklı optimizasyon algoritmaları bulunsa da, optimizasyon işleminin temelinde zincir kuralına göre kısmi türev alma ve bu şekilde ağda geri yayılım süreci yatmaktadır. Denklem 2.12'de ifade edildiği gibi ağırlıkların güncellenmesi işleminde, hesaplanan toplam hata geriye doğru yayılarak, her bir ağırlığın toplam hataya olan etkisi kısmi türev ve zincir kuralı kullanılarak hesaplanır [36]. Çıktı katmanındaki bir düğüm için, bu düğüme gelen ağırlıklardan birinin zincir kuralına göre güncellenmesi işlemi Şekil 2.18'deki gibi gösterilebilir. Burada çok katmanlı YSA'nın son gizli katmanı ve bağlı oldukları çıkış düğümlerinden birisi gösterilmektedir. Şekildeki J hesaplanan maliyet, \hat{y} (veya y') ağın tahmini, I düğümün girdisi ve $w1$, $w2$, $w3$ ve $w4$ ise düğüme bağlı ağırlık değerleridir.



Şekil 2.18: Tam bağlantılı katmanlarda geri yayılım işlemi

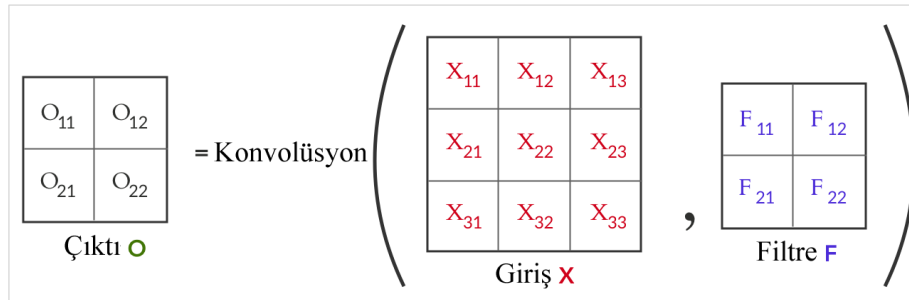
Geri yayılım işleminde zincir kuralına göre kısmi türevlerin çarpımı yoluyla örneğin $w1$ ağırlığının güncellenmesi Denklem 2.18'e göre gerçekleştirilir.

$$\frac{dJ}{dw1} = \frac{dJ}{\hat{y}} * \frac{d\hat{y}}{dI} * \frac{dI}{dw1} \quad (2.18)$$

Burada ağırlık $w1$ değerinin toplam hata üzerindeki etkisi elde edilmiş olur ve $w1$ ağırlığı elde edilen $\frac{dJ}{dw1}$ değerinin gradyan inişi (gradyan descent) yöntemine göre Denklem 2.12, Adam optimizasyon algoritmasına göre ise Denklem 2.17'de kullanılmasıyla güncellenmiş olur. Aynı katmanda bulunan diğer ağırlıklar ile önceki katmanlarda bulunan ağırlıkların güncellenmesi de, benzer şekilde zincir kuralına göre kısmi türevlerin çarpımı yoluyla gerçekleştirilir.

2.10.3. Evrişim Katmanlarında Geri Yayılım

Evrişim katmanlarının ileri geçişi evrişimleri kullandığı gibi, geri yayımları da evrişimleri kullanır. Şekil 2.19'da gösterilen bir X girdisinin, bir F filtresi ile evrişim işlemi O çıktısını verir.



Şekil 2.19: X ve F arasındaki evrişim işlemi

Burada O_{11} , O_{12} , O_{13} ve O_{14} değerleri Denklem 2.19 - 2.22' deki gibi hesaplanır.

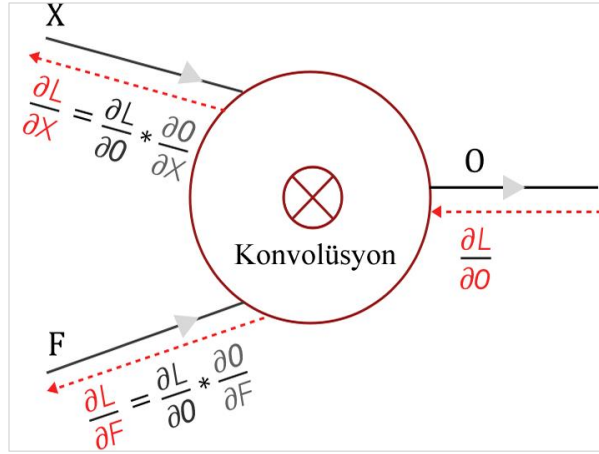
$$O_{11} = X_{11}F_{11} + X_{12}F_{12} + X_{21}F_{21} + X_{22}F_{22} \quad (2.19)$$

$$O_{12} = X_{12}F_{11} + X_{13}F_{12} + X_{22}F_{21} + X_{23}F_{22} \quad (2.20)$$

$$O_{21} = X_{21}F_{11} + X_{22}F_{12} + X_{31}F_{21} + X_{32}F_{22} \quad (2.21)$$

$$O_{22} = X_{22}F_{11} + X_{23}F_{12} + X_{32}F_{21} + X_{33}F_{22} \quad (2.22)$$

Denklemler ileri yayılım ile elde edilen sonuçları verir (sayısal hesaplama için Şekil 2.4'e bkz.). Geri yayılımda evrişim işlemini gerçekleştirmek için, tam bağlantılı katmanlardaki geri yayılıma benzer şekilde, bir sonraki katmandan çıktı O 'ya göre kayıp gradyanı dL/dO olarak alınır ve zincir kuralı ve geri yayılım kullanılarak Şekil 2.20'deki işlemler gerçekleşir.



Şekil 2.20: Geri yayılım işleminde X ve F değerlerinin güncellenmesi [63]

Burada dO/dX , dO/dF yerel gradyanları ve dL/dO diğer katmanlara geri yayılması gereken önceki katmandaki kayıptır. Bu değişkenlere göre hesaplanan dL/dF ve α öğrenme katsayısı kullanılarak F filtresinin güncellenmesi Denklem 2.23' deki gibi gerçekleştirilir.

$$F_{yeni} = F - \alpha \frac{dL}{dF} \quad (2.23)$$

F filtresinin her bir elemanı için güncellenmiş değeri bulmak için gerekli olan dL/dF değerini hesaplamak amacıyla Şekil 2.20'deki gibi gerçekleştirilecek zincir kuralı

işlemi, Denklem 2.24' deki gibi genelleştirilebilir. Burada M, F filtresinin eleman sayısını gösterir.

$$\frac{dL}{dF_i} = \sum_{k=1}^M \frac{dL}{dO_k} * \frac{dO_k}{dF_i} \quad (2.24)$$

F filtresinin güncellenmesine benzer şekilde X matrisinin güncellenmesi Denklem 2.25' deki gibi genelleştirilebilir.

$$\frac{dL}{dX_i} = \sum_{k=1}^M \frac{dL}{dO_k} * \frac{dO_k}{dX_i} \quad (2.25)$$

ÜÇÜNCÜ BÖLÜM

3. YÖNTEM VE DENEYSEL ÇALIŞMALAR

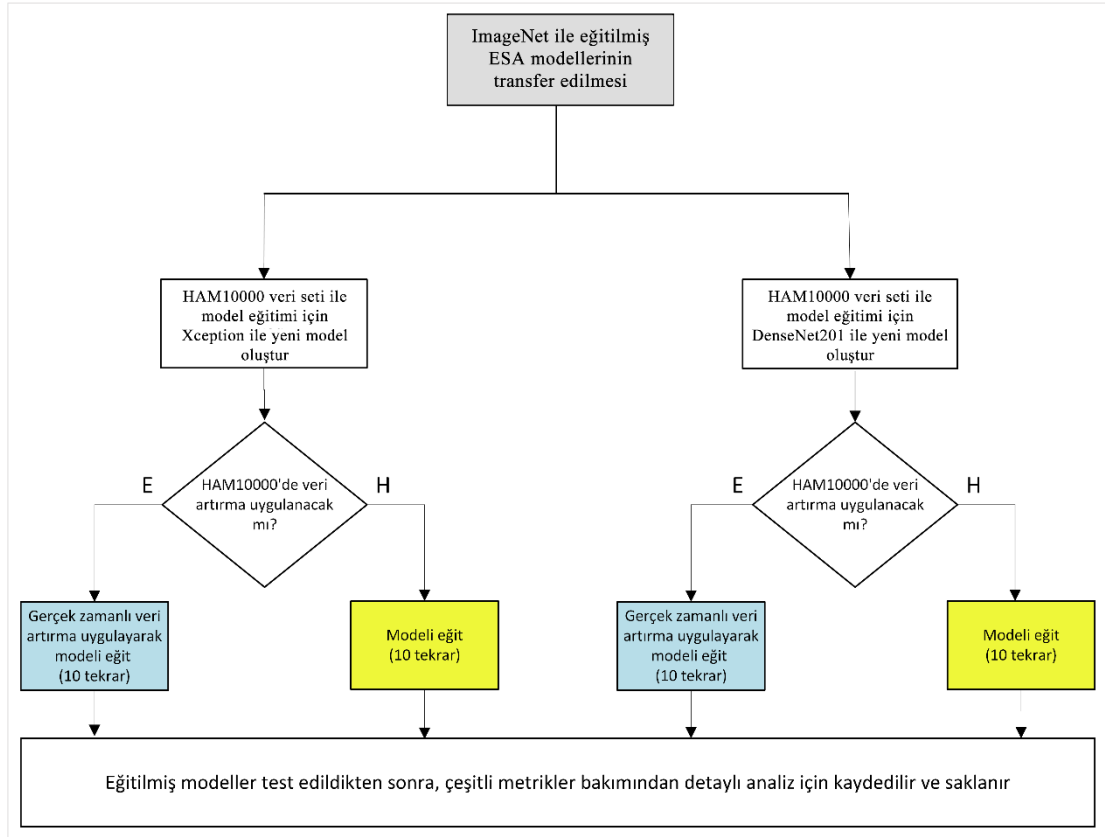
Bu tez çalışması kapsamında, ImageNet veri seti ile ön eğitilmiş gelişmiş iki ESA modeli, HAM10000 veri setinin yedi cilt hastalığı türüne sınıflandırılması amacıyla kullanılmıştır. Bu çalışmada kullanılan derin sinir ağı modelleri Keras'ta geliştirilmiştir. Keras [64], Python dili ile yazılmış, makine öğrenimi platformu TensorFlow [65] üzerinde çalışan bir derin öğrenme API'sidir. Hızlı uygulama geliştirmeye odaklanılarak geliştirilmiş basit, esnek ve güçlü bir yapıya sahiptir. Modellerin eğitilmesi, Google Colaboratory (Colab) [66] ortamında gerçekleştirildi.

Yapılan deneysel çalışmalarda, içerisinde herhangi bir medikal görüntü barındırmayan ImageNet veri setinin ön eğitilmiş (pre-trained) ağırlıklarının transfer edilmesi sonrasında elde edilen başarımla içerisinde tamamen cilt hastalık görüntüleri bulunan BCN20000 veri setinin ön eğitilmiş ağırlıklarının transfer edilerek HAM10000 veri setinin sınıflandırılması ve bu iki yöntemin karşılaştırılması amaçlanmıştır.

Tezin amacı doğrultusunda seçilen her iki model ile (Xception, DenseNet201) gerçekleştirilen deneyler iki ana gruba ayrılmıştır. ImageNet ağırlıkları transfer edilerek doğrudan model eğitimi ve sınıflandırma gerçekleştirilen birinci grup deneylerde (bkz. Bölüm 3.3.1), her iki model HAM10000 veri seti ile eğitilmiş ve bu veri seti sınıflandırılmıştır. İş akışı Şekil 3.1'de gösterilmiş ve işlem basamakları şu şekilde gerçekleştirilmiştir:

- 1) ImageNet veri seti ile ön eğitilmiş Xception ve DenseNet201 modellerinin ağırlıklarıyla birlikte alınması,
- 2) Alınan ESA modellerinin son katmanına özellik haritalarının vektörleştirilmesi için global ortalama ortaklama (GOO) katmanı (bkz. Bölüm

- 2.4), 7 düğümlü çıkış katmanı ve Softmax aktivasyon fonksiyonu (bkz. Bölüm 2.7) katmanı eklenerek modellerin nihai olarak oluşturulması,
- 3) Hem veri artırma uygulanarak 10 kez, hem de veri artırma uygulanmadan 10 kez olmak üzere her iki ESA modelinin HAM10000 veri seti ile 20'şer kez eğitilmesi,
 - 4) Her bir eğitim işleminden sonra eğitilen modellerin, test edilmek ve çeşitli metrikler bakımından sonuçlarını elde etmek için diske kaydedilmesi



Şekil 3.1: ImageNet ağırlıkları ile doğrudan transfer öğrenme yönteminin HAM10000 veri seti ile uygulanmasına ait akış diyagramı (Ağırlıklar: ImageNet)

İkinci deney grubunda (bkz. Bölüm 3.3.2) ise, yalnızca ImageNet veri setinden gelen ağırlıklarını kullanmak yerine bu ağırlıkları güncelleyen ara öğrenme ile transfer öğrenme yöntemi önerilmiş ve deneyler gerçekleştirilmiştir. Önerilen yöntem kapsamında, her iki model ImageNet ağırlıkları kullanılarak öncelikle sekiz cilt hastalık türü içeren BCN20000 veri seti ile eğitilmiş ve eğitilen her iki modelin

ağırlıkları kaydedilmiştir. Daha sonra kaydedilen her iki model transfer edilerek HAM10000 veri seti ile tekrar eğitilmiş ve bu veri seti sınıflandırılmıştır. Önerilen ara öğrenme ile transfer öğrenme yöntemine ilişkin iş akışı Şekil 3.2’de gösterilmiş ve işlem basamakları şu şekilde gerçekleştirilmiştir.

- 1) ImageNet veri seti ile ön eğitilmiş Xception ve DenseNet201 modellerinin ağırlıklarıyla birlikte alınması,
- 2) Alınan ESA modellerinin son katmanına özellik haritalarının vektörleştirilmesi için global ortalama ortaklama (GOO) katmanı (bkz. Bölüm 2.4), 7 düğümlü çıkış katmanı ve Softmax aktivasyon fonksiyonu (bkz. Bölüm 2.7) katmanı eklenerek modellerin nihai olarak oluşturulması,
- 3) Veri artırma uygulanarak 8 sınıflı BCN20000 veri seti her iki ESA modelinin 1’er kez eğitilmesi ve eğitilmiş modellerin diske kaydedilmesi,
- 4) BCN20000 ile ara eğitimden geçirilerek ağırlıkları güncellenmiş her iki ESA modelinin tekrar transfer edildikten sonra son iki katmanı çıkarılarak, 8 düğümlü çıkış katmanı yerine 7 düğümlü çıkış katmanı ve Softmax aktivasyon fonksiyonu katmanı eklenmesi,
- 5) Hem veri artırma uygulanarak 10 kez, hem de veri artırma uygulanmadan 10 kez olmak üzere her iki ESA modelinin HAM10000 veri seti ile 20’şer kez eğitilmesi,
- 6) Her bir eğitim işleminden sonra eğitilen modellerin, test edilmek ve çeşitli metrikler bakımından sonuçlarını elde etmek için diske kaydedilmesi

Çalışmada kullanılan BCN20000 ve HAM10000 veri setlerine ait detaylar ile bu iki veri setinin diğer bazı cilt hastalıkları veri setlerinden farklarına ilişkin açıklamalar Bölüm 3.1’de detaylı olarak gösterilmiştir. Yapılan deneysel çalışmalarda, ImageNet ağırlıkları doğrudan transfer edilerek gerçekleştirilen ilk grup deneylerde yalnızca ImageNet ağırlıkları kullanılırken, diğer deney grubunda ise BCN20000 ile güncellenmiş (ara öğrenme gerçekleştirilmiş) ImageNet ağırlıkları kullanılmıştır. Burada amaç, cilt hastalık görüntüleri içeren veri seti ile eğitilmiş modellerin, cilt hastalıkları içeren başka bir hedef veri setinin sınıflandırılma başarısındaki etkisini çeşitli metrikler bakımından ortaya çıkarılmasıdır. Sonuçları daha net ortaya koymak

için her iki deney grubunda, iki farklı ESA modeli ile HAM10000 veri setinde hem veri artırma (data augmentation) uygulanarak hem de veri artırma uygulanmadan farklı alt grupta deneyler gerçekleştirilmiştir.



Şekil 3.2: Ara öğrenme ile transfer öğrenme yönteminin (önerilen) HAM10000 veri seti ile uygulanmasına ait akış diyagramı (Ağırlıklar: ImageNet + BCN20000)

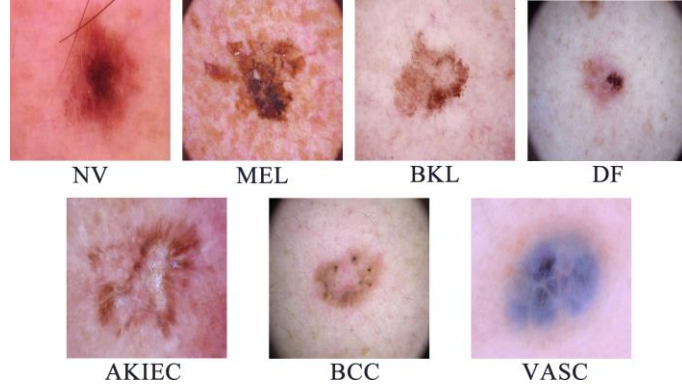
3.1. VERİ SETLERİ

Cilt hastalıkları ile ilgili çok sayıda veri seti bulunmaktadır. Bunlardan bazıları HAM10000 [24], BCN20000 [67], MSK[68], PH2 [10], 2018 JID Editorial Images, Sydney (MIA / SMDC), UDA [69], Dermofit [70], PAD-UFES-20 [71], Derm7pt [23] şeklinde sıralanabilir. Verisetlerinde melanocytic nevus (NV), melanoma (MEL), basal cell carcinoma (BCC), dermatofibroma (DF), benign keratosis (BKL), vascular lesion (VASC), squamous cell carcinoma (SCC) ve actinic keratosis (AKIEC) cilt hastalık türlerinden tümü ya da birkaçı belli sayılarda örneklerle temsil edilmektedir. Cilt hastalıkları içeren bazı veri setlerinde bulunan hastalık sınıfları ve her bir sınıftaki örnek sayıları Tablo 3.1’de gösterilmektedir.

Tablo 3.1: Cilt hastalıkları içeren verisetleri ve hastalık sınıflarının örnek sayıları

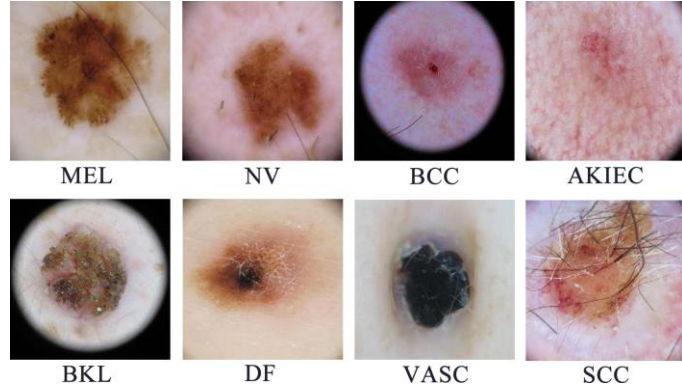
Veri seti	AKIEC	BCC	BKL	DF	MEL	NV	SCC	VASC	Toplam
BCN20000	737	2809	1138	124	2857	4206	431	111	12413
HAM10000	327	514	1099	115	1113	6705	0	142	10015
DERMOFIT	123	239	257	65	76	331	0	121	1212
DERM7PT	0	82	90	40	504	1150	0	58	1926
MSK	0	32	0	452	826	2241	0	0	3551
UDA	0	0	0	0	193	408	0	0	601
PH2	0	0	0	0	40	160	0	0	200

HAM10000 ve BCN20000 veri setlerinin etiketli sınıf sayısı bakımından zengin olduğu görülmektedir. Ancak sınıflardaki örnek sayılarında dengesizlik bulunmaktadır. Örneğin HAM10000 veri setinde melanocytic nevus (NV) hastalık türünde bulunan örnek sayısı toplam örnek sayısının %66.94’ünü, dermatofibroma (DF) sınıfındaki ise %1.14’ünü oluşturmaktadır. HAM10000 veri setine ait örnek görüntüler Şekil 3.3’ de gösterilmektedir.



Şekil 3.3: Yedi sınıflı HAM10000 veri seti örnek görüntüleri

BCN20000 veri seti nispeten daha dengeli olsa da, melanocytic nevus sınıfındaki örnek sayısı, toplam görüntü sayısının %33,88'ini, vascular lesion (vasc) sınıfındaki örnek sayısı ise yalnızca %0.89'unu oluşturmaktadır. BCN20000 veri seti, HAM10000' de olmayan squamos cell carcinoma cilt hastalık türüne ait 431 adet görüntü içermektedir. BC20000 veri setine ait örnek görüntüler Şekil 3.4' te gösterilmektedir.



Şekil 3.4: Sekiz sınıflı BCN20000 veri seti örnek görüntüleri

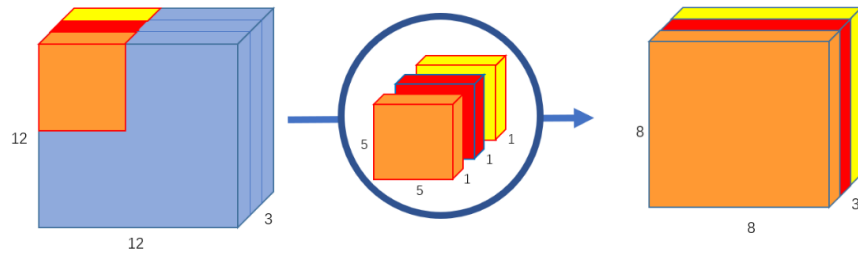
3.2. EVRİŞİMSSEL SİNİR AĞI MODELLERİNİN SEÇİMİ

Deneyisel çalışmalarda kullanılması planlanan ESA modellerinin belirlenmesinde, modellerin parametre sayıları, boyutu (MB) ve ön deneylerde gösterdikleri performanslar dikkate alındı. Fiziksel olarak düşük boyutlu ve parametre

sayısı bakımından da diğer birçok ESA modelinde daha az parametreye sahip modeller seçilerek, HAM10000 veri seti ile birkaç ön deney gerçekleştirildi. Sonuç olarak, Xception [72] ve DenseNet201 [55] modellerinin tüm belirlenen kriterler bakımından en uygun ve başarılı olarak en yüksek sonuç verdiği gözlemlenmiştir. Bu sebeple çalışmamızda bu iki ESA modeli kullanılmıştır.

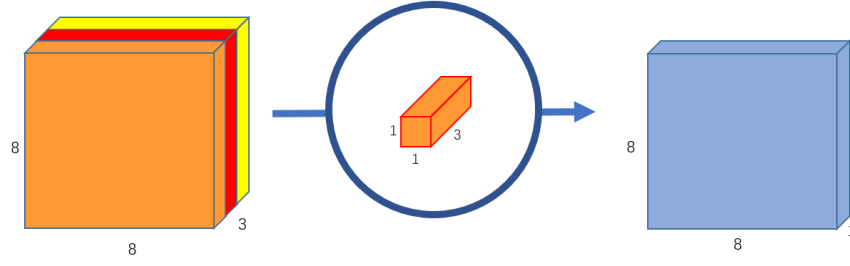
3.2.1. Xception

Xception ESA modeli, Google araştırmacıları tarafından geliştirilmiş, derinlemesine ayrılabilir evrişimler (Depthwise Separable Convolutions) içeren bir mimariye sahiptir. Derinlemesine ayrılabilir evrişim işlemleri, hesaplama süresi açısından çok daha verimli olduğu varsayılan klasik evrişimlere alternatif olarak sunulmaktadır. Yalnızca uzamsal boyutlarla değil, derinlik boyutuyla ve kanalların sayısı ile de ilgilenmektedir. Derinlemesine ayrılabilir evrişim, bir çekirdeğin (filtre) iki ayrı evrişim yapan iki farklı çekirdeğe bölünmesiyle gerçekleştirilir. Bu çekirdeklerden birisi derinlemesine evrişim yaparken, diğeri noktasal evrişim yapar. Derinlemesine evrişim Şekil 3.5’de gösterildiği gibi gerçekleşir. Şekilde 12x12 boyutundaki 3 renk kanallı temsili bir görüntü, öncelikle her bir renk kanalı için ayrı ayrı gezinen 5x5 boyutunda 1 kanallı çekirdeklerle evrişim işlemine tabi tutulmaktadır. Evrişim işlemi sonunda 8x8x3 biçiminde bir görüntü elde edilmiş olur.



Şekil 3.5: Derinlemesine evrişim [73]

Derinlemesine evrişim aşamasından sonra Şekil 3.6’da gösterildiği gibi gerçekleşen noktasal evrişim aşaması gerçekleşir.

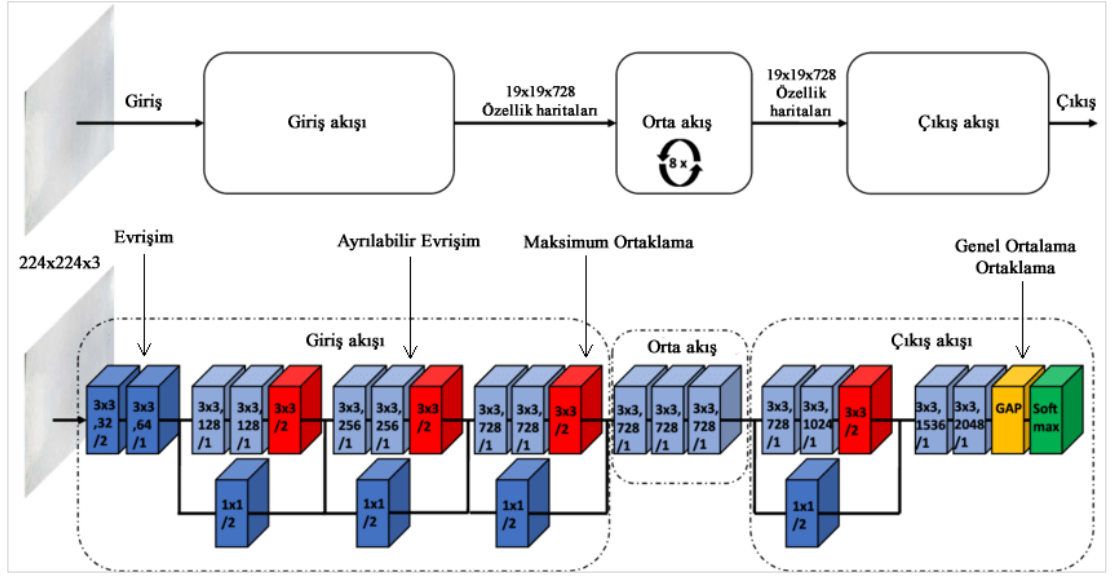


Şekil 3.6: Noktasal evrişim [73]

Bu aşamada, 1x1 boyutunda çekirdek kullanılır. Bu çekirdek giriş görüntüsünün sahip olduğu kanal derinliğine sahiptir ve Şekil 3.6'daki gösterimde 8x8x1 biçiminde bir görüntü elde etmek için 1x1x3 biçiminde noktasal çekirdek kullanılır. Şekil 3.5 ve 4.6'daki gösterimlere göre, derinlemesine ayrılabilir evrişim işleminde derinlemesine evrişim aşamasının sonunda elde edilen görüntü, noktasal evrişim aşamasından geçirilir ve 8x8x1 biçiminde nihai bir görüntü elde edilir. Derinlemesine ayrılabilir evrişim ile yapılan toplam çarpma işlemi sayısında, standart evrişim işlemine göre çok ciddi bir düşüş gerçekleşir. Böylece daha az hesaplama ile daha kısa sürede daha fazla girdi işlenebilir. Tablo 3.2'de Xcepiton modelinin temel özellikleri, Şekil 3.7'de yapısal şeması (şemaya TN katmanları dahil edilmemiş ve açıklamalar eklenmiştir), gösterilmektedir. Xcepiton model mimarisi, giriş, orta ve çıkış akışları olmak üzere üç kısımda düzenlenmiştir. Veriler önce giriş akışından, ardından sekiz kez tekrarlanan orta akıştan ve son olarak çıkış akışından geçer. Bütün standart evrişim ve ayrılabilir evrişim katmanlarını bir TN katmanı izler. Ayrılabilir evrişim katmanları, derinlik çarpanı olarak 1 kullanır.

Tablo 3.2: Xcepiton ESA modelinin temel özellikleri [75]

Boyut (MB)	Parametre Sayısı (~)	Katman Derinliği	Giriş Boyutu
88	23 milyon	126	224x224



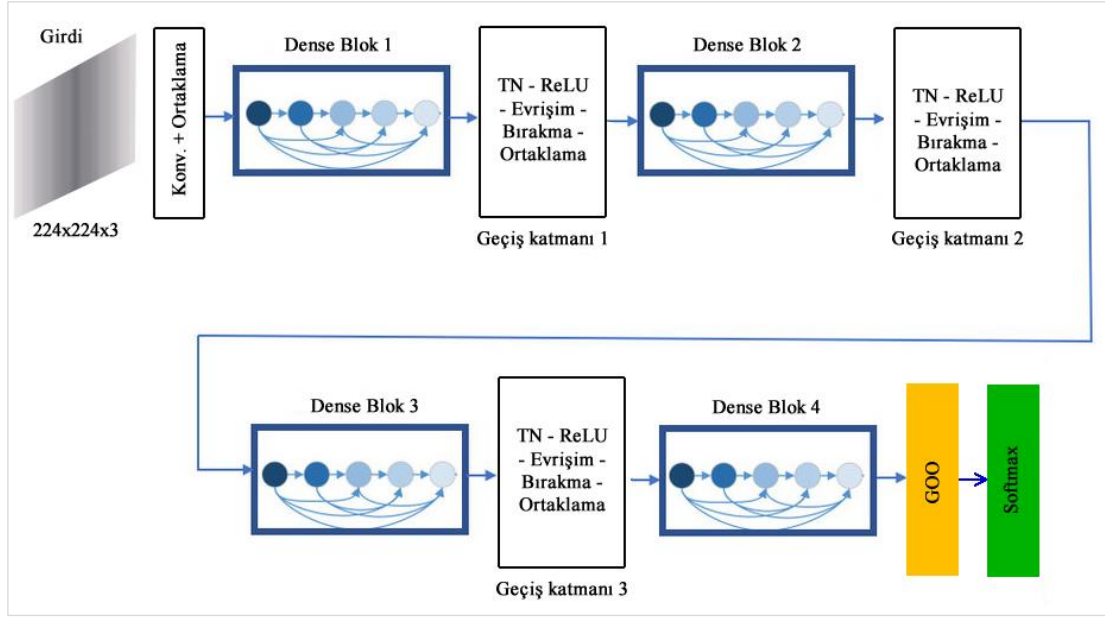
Şekil 3.7: Xception ESA modelinin genel yapısı [74]

3.2.2. DenseNet201

DenseNet ESA modelleri genel olarak giriş katmanı ve tahmin katmanları dışında Dense bloklar ve Dense bloklar arasında bulunan geçiş (transition) katmanlarından oluşur. Her bir geçiş katmanı, TN-ReLU-Evrişim katmanlarını içerir. Yoğun bloklarda farklı sayılarda filtrelerle evrişim işlemleri gerçekleştirilir. DenseNet modellerinin diğer ESA modellerinden en önemli yapısal farkı, bir yoğun blok içerisindeki her katman (ilk katmanlar hariç), o yoğun blok içerisinde bulunan tüm katmanlardan ek girdiler alır ve kendi özellik haritalarını sonraki katmanlara iletir [55]. Bu işlem için birleştirme (concatenate) yapılır. DenseNet201 modelinin temel özellikleri Tablo 3.3’de, genel yapısı (değiştirilmiş olarak) Şekil 3.8’de, gösterilmektedir. Bu modelin Xception’a kıyasla daha az parametre sayısına ve daha fazla katman derinliğine sahip olduğu görülmektedir.

Tablo 3.3: DenseNet201 ESA modelinin temel özellikleri [75]

Boyut (MB)	Parametre Sayısı (~)	Katman Derinliği	Giriş Boyutu
80	20 milyon	402	224x224



Şekil 3.8: DenseNet201 ESA modelinin genel yapısı [76]

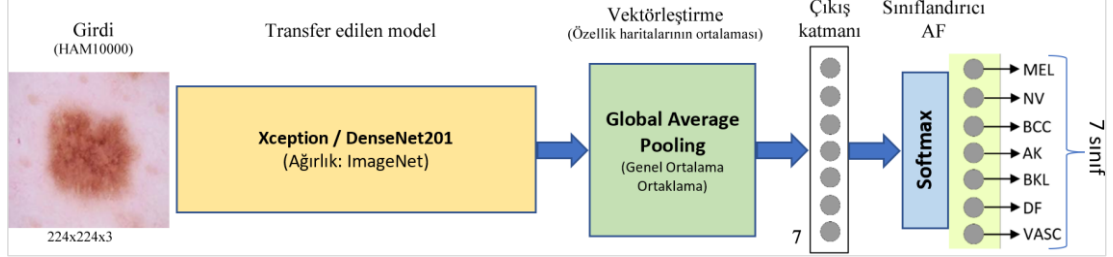
3.3. EVRİŞİMSSEL SİNİR AĞI MODELLERİNİN OLUŞTURULMASI VE EĞİTİLMESİ

Tez kapsamında oluşturulan modeller, doğrudan ImageNet ağırlıkları ile TÖ yöntemi uygulanarak eğitilen modeller ve transfer edilen modellerin ImageNet ağırlıkları BCN20000 veri seti ile bir ara eğitimden ve öğren işleminden geçirilerek uygulanan TÖ yöntemi ile eğitilen modeller olmak üzere iki ana kategoride oluşturulmuş ve eğitilmiştir.

3.3.1. Doğrudan ImageNet Ağırlıkları Kullanılan Deneyler İçin Modellerin Oluşturulması

Bu bölümde yapılan deneysel çalışmalarda, ImageNet veri seti ile ön eğitilmiş Xception ve DenseNet201 modellerinin ağırlıkları transfer öğrenme yaklaşımı ile transfer edilmiştir. Bu modeller kullanılarak oluşturulan yeni modeller HAM10000 veri seti ile eğitilmiştir. Xception ve DenseNet201 ağ modelleri kullanılarak oluşturulan yeni ağ modellerin blok şeması Şekil 3.9'daki gibidir. Burada HAM10000 veri setine ait girdi görüntüleri, ImageNet ağırlıkları ile transfer edilen ağ modellerine

girdi olarak verilmiş ve ağ modellerinin son katmanı bir GOO katmanı ile birleştirilmiştir. GOO çıkışı ise bir Softmax sınıflandırıcıya bağlanmıştır.



Şekil 3.9: HAM10000 veri setinin sınıflandırılmasında, doğrudan ImageNet ağırlıkları kullanılarak eğitilen Xception ve DenseNet201 ağ modelleri

3.3.1.1. Veri Hazırlama ve Artırma

HAM10000 veri setindeki görüntüler 600x450 piksel boyutunda olduğundan, öncelikle oluşturulan ağ modelinin girdi boyutuna uygun hale getirmek için yeniden boyutlandırılarak 224x224 boyutlarına ölçeklenmiştir. Veri setinden, gruplara ataması rastgele gerçekleştirilmiş on farklı eğitim, doğrulama ve test grubu oluşturulmuştur. Eğitim sırasında eğitim ve doğrulama verilerinde gerçek zamanlı veri artırma (eğitim görüntüleri model eğitiminden önce fiziksel olarak değil, model eğitimi sırasında her eğitim döneminden (epoch) önce gerçekleştirilir) uygulanmıştır. Test verisi üzerinde herhangi bir işlem yapılmamış ve orijinaliği korunmuştur. Veri setinin eğitim, doğrulama ve test olarak ayrıştırılması Tablo 3.4’de gösterilen oranlara göre gerçekleştirildi. Test verisi için ayrılan %20’lik bölümün dışında kalan verinin %10’u, toplam veri setinin %8’i doğrulama verisi olarak ayrılmıştır. Veri setinin bölünmesinde, sınıflardaki örnek sayılarının toplam içindeki oranı dikkate alınmıştır.

Tablo 3.4: HAM10000 veri seti için eğitim, doğrulama ve test kümesi oranları

Eğitim	Doğrulama	Test
%72	%8	%20
7215	800	2000

Veri artırma işlemi, özellikle sınıflardaki örnek sayısı bakımından dengesizlik olması veya sınıfları temsil edecek yeterli sayıda örnek bulunmaması durumlarında oldukça faydalıdır. Veri artırma ile fiilen yeni veriler toplamadan mevcut eğitim verilerinden biraz değiştirilmiş kopyalar ekleyerek veri miktarı yapay olarak artırılmaktadır. Dengesiz ve az örnek sayısına sahip çok sınıflı veri setlerinde karşılaşılan aşırı uydurma sorununa karşı yardımcı olur [32]. Gerçek zamanlı veri artırma ile, her eğitim döneminde her sınıftaki örnek sayısı kadar Tablo 3.5’de belirtilen veri artırma parametrelerine göre rastgele artırma işlemi uygulanmıştır. Veri artırma işleminde veri öncelikle yeniden ölçeklenmiş, döndürme, kaydırma, kırpma, yakınlaştırma ve çevirme gibi işlemler uygulanmıştır. Veri seti üzerinde ayrıca kontrast iyileştirme, gürültü azaltma, kırpma gibi ilave herhangi bir görüntü işleme veya geliştirme tekniği uygulanmadı.

Tablo 3.5: Veri artırma parametreleri

Parametre	Değer
Yeniden ölçekleme (<i>1.0/255</i>)	Var
Döndürme açısı	90°
Genişlik kaydırma aralığı	0.2
Yükseklik kaydırma aralığı	0.2
Kırpma aralığı	0.2
Yakınlaştırma aralığı	0.8 – 1.2
Yatay – dikey çevirme	Var

3.3.1.2. Modellerin Eğitilmesi

Her iki ESA modeli Şekil 3.9’daki gibi hem veri artırma uygulanarak hem de veri artırma uygulanmadan HAM10000’den rastgele oluşturulmuş 10 ayrı veri seti ile ayrı ayrı eğitilmiştir. Bu grupta gerçekleştirilen toplam deney sayısı Tablo 3.6’daki gibidir. Bu bölümdeki deneylerde her iki ESA modeli veri artırma uygulanarak ve uygulanmadan, HAM10000 veri setinin rastgele oluşturulmuş 10 ayrı eğitim ve doğrulama verileri ile eğitilmiş ve test edilmiştir. Toplamda gerçekleştirilen deney sayısı 40 olmuştur.

Tablo 3.6: Doğrudan ImageNet ağırlıkları kullanılarak gerçekleştirilen deney sayıları

Model	Veri artırma var	Veri artırma yok
Xception	10	10
DenseNet201	10	10

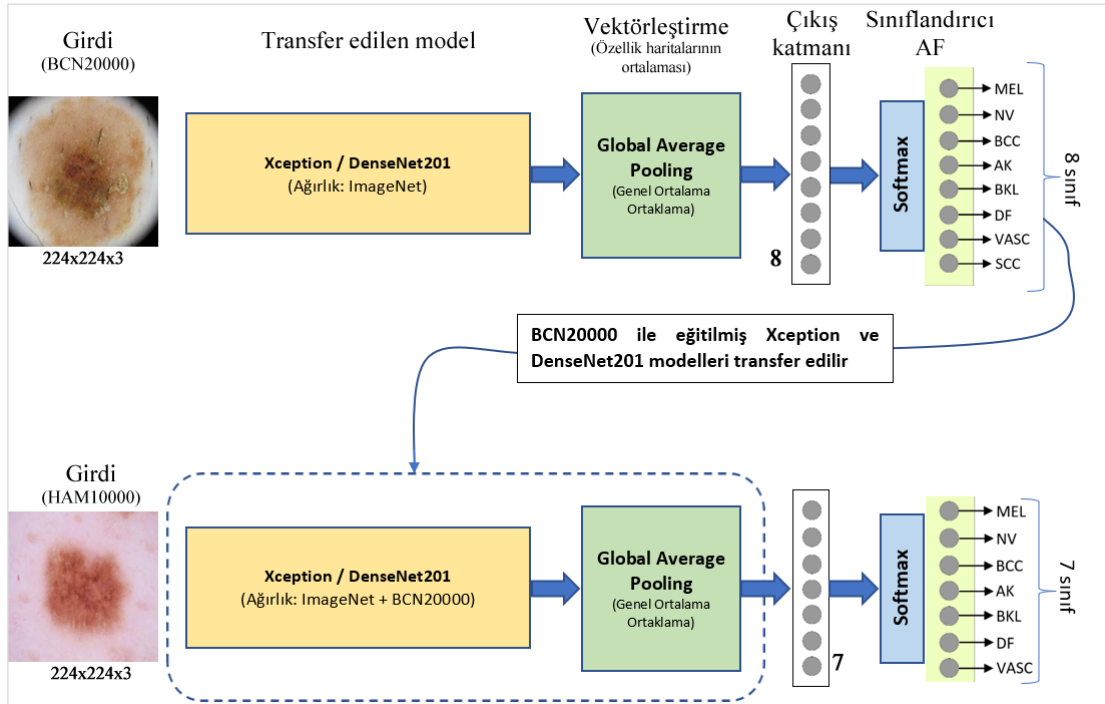
Değerlendirme için her bir onlu deney grubunun aritmetik ortalaması alındı. Oluşturulan ağ modellerinin eğitilmesi için kullanılan hiper parametreler ve değerleri Tablo 3.7’deki gibidir. Hataları azaltarak modelin daha iyi genelleme yapabilecek şekilde öğrenmesini sağlamak için Adam optimizasyon algoritması kullanılmıştır (bkz. Bölüm 2.10). Öğrenme oranı, optimizasyon işleminin hassasiyetini artırmak ancak öğrenme hızını da çok düşürmemek için 0.0001 olarak belirlenmiştir. Modellerin aşırı uydurmasını (overfitting) önlemek için, her eğitim dönemi sonunda doğrulama verisi ile yapılan test başarısında 8 dönem arka arkaya artış olmaması durumunda erken durdurma işlemi uygulanmıştır. Maliyet fonksiyonu olarak çok sınıflı sınıflandırmalarda yaygın kullanılan *categorical_crossentropy* fonksiyonu kullanılmıştır (bkz. Bölüm 2.9). Görüntülerin girdi boyutu, kullanılan ESA modellerinin girdi boyutu ile aynı olacak şekilde 224x224 olarak ayarlanmıştır. Son katmanda çok sınıflı sınıflandırma problemlerinde yaygın kullanılan softmax AF kullanılmıştır. Başlangıç ağırlıkları olarak ImageNet veri setinden gelen ağırlıklar kullanılmıştır.

Tablo 3.7: Doğrudan ImageNet ağırlıkları kullanılan deneylerde, modellerin HAM10000 veri seti ile eğitime hiper parametreleri

Parametre	Xception & DenseNet201
Optimizasyon algoritması	Adam
Öğrenme katsayısı (<i>learning_rate</i>)	0.0001
Erken durdurma (<i>val_accuracy</i>)	Var
Erken durdurma limiti (<i>patience</i>)	8
Maliyet fonksiyonu (<i>loss</i>)	<i>categorical_crossentropy</i>
Girdi boyutu (<i>input_shape</i>)	224x224
Sınıflandırıcı AF	Softmax
Ağırlık (weights)	ImageNet

3.3.2. Ara Öğrenme ile Transfer Öğrenme Yöntemi Uygulanan Deneyler İçin Modellerin Oluşturulması

İkinci gruptaki deneyler iki aşamada gerçekleştirilmiştir. Öncelikle, ImageNet veri seti ile ön eğitilmiş Xception ve DenseNet201 modelleri ağırlıklarıyla birlikte transfer edilmiş ve bu modeller BCN20000 veri seti ile tekrar eğitilerek ara öğrenme işlemi uygulanmıştır. Daha sonra, BCN20000 veri seti ile eğitilen bu ağ modellerinden transfer edilen ağırlıklar kullanılarak HAM10000 veri setinin eğitim ve test işlemleri gerçekleştirilmiştir (işlem adımları için Bölüm 3'e bkz.). Tez kapsamında önerilen ara öğrenme ile TÖ yöntemi uygulanarak gerçekleştirilen deneyler için oluşturulan ağ modellerinin blok şeması Şekil 3.10'daki gibidir.



Şekil 3.10: HAM10000 veri setinin sınıflandırılmasında, BCN20000 veri seti ile ara öğrenme gerçekleştirilip transfer edilen ağırlıklar kullanılarak eğitilen Xception ve DenseNet201 ağ modelleri

Burada BCN20000 veri setine ait girdi görüntüleri, ImageNet ağırlıkları ile transfer edilen ağ modellerine girdi olarak verilmiş ve ağ modellerinin son katmanı bir GOO katmanı ile birleştirilmiştir. GOO çıkışı ise bir Softmax sınıflandırıcıya bağlanmıştır. Daha sonra, elde edilen bu ağ modeli transfer edilerek, yedi sınıflı bir Softmax sınıflandırıcıya bağlanmıştır. Oluşturulan yeni model, HAM10000 veri seti ile eğitilerek bu veri seti sınıflandırılmıştır.

Eğitim işlemi için BCN20000 veri seti öncelikle eğitim ve doğrulama olarak iki gruba ayrıldı ve eğitim sırasında hem eğitim hem de doğrulama verilerine gerçek zamanlı veri artırma uygulandı. Bu tez çalışmasının amacı HAM10000 veri setindeki görüntüleri sınıflandırmak olduğu için BCN20000 veri seti ile modellerin eğitilmesi işlemi test verisi ayrılmamıştır. BCN20000 veri seti ile eğitilerek ara öğrenme gerçekleştirilmiş modellerin, HAM10000 veri setinin sınıflandırması için tekrar kullanılması aşamasında, doğrudan ImageNet ağırlıkları kullanılan (bkz. Bölüm 3.3.1) deneylerde HAM10000 veri seti için kullanılan 10 farklı eğitim, doğrulama ve test alt veri kümeleri kullanılmıştır.

3.3.2.1. Veri Hazırlama ve Artırma

BCN20000 veri setindeki görüntüler 1024x1024 piksel boyutunda olduğundan, öncelikle oluşturulan ağ modellerinin girdi boyutuna uygun hale getirmek için yeniden boyutlandırılarak 224x224'e küçültülmüştür. Hem BCN20000 hem de HAM10000 veri setleri için eğitim sırasında eğitim ve doğrulama veri kümelerinde gerçek zamanlı veri artırma uygulanmıştır. Gerçek zamanlı veri artırma ile, her eğitim döneminde her sınıftaki örnek sayısı kadar Tablo 3.5'de belirtilen parametrelere göre rastgele artırma işlemi uygulanmıştır. Veri seti üzerinde ayrıca kontrast iyileştirme, gürültü azaltma, kırpma gibi ilave herhangi bir görüntü işleme veya geliştirme tekniği uygulanmamıştır. BCN20000 veri setinin eğitim ve doğrulama olarak ayrıştırılmasında, her sınıftaki örnek sayısının başlangıçtaki oranları korundu ve veri setinin eğitim ve doğrulama oranları Tablo 3.8'de gösterilmiştir. Önerilen ara öğrenme ile transfer öğrenme yönteminin uygulanması için, BCN20000 veri seti ile modellerin eğitilmesi işlemi için test verisi ayrılmamıştır. Çünkü nihai olarak sınıflandırılıp test edilmek istenen veri seti HAM10000'dir. Eğitim işlemi için veri setinin %90'ı, test işlemi için %10'u

ayrılmıştır. Bu bölümde gerçekleştirilen deneylerde, HAM10000 veri seti için eğitim, doğrulama ve test bölümlenmesi, Bölüm 3.3.1’de gerçekleştirilen deneylerdeki gibidir ve detaylar Tablo 3.4’ de gösterilmiştir.

Tablo 3.8: BCN20000 veri seti için eğitim ve doğrulama kümesi oranları

Eğitim	Doğrulama
%90	%10
11172	1241

3.3.2.2. Modellerin Eğitilmesi

Öncelikle, transfer edilen Xception ve DenseNet201 modelleri BCN20000 veri seti ile gerçek zamanlı veri artırma uygulanarak birer kez eğitilmiştir (Şekil 3.10 üst satır). Modellerin eğitilmesi için kullanılan hiper parametreler ve değerleri Tablo 3.9’deki gibidir. BCN20000 veri seti ile modellerin ara eğitimden geçirilmesi için gerçekleştirilen deneysel çalışmalar için belirlenen hiper parametreler birinci grupta gerçekleştirilen deneylerle aynı olacak şekilde belirlenmiştir. Tek farklılık, erken durdurma parametresi 8’den 10’a yükseltilmiştir. BCN20000 veri seti daha fazla veriye sahip olduğu için erken durdurma parametresi yükseltilmiştir.

Tablo 3.9: Ara öğrenme için ağ modellerinin BCN20000 veri seti ile eğitilme hiper parametreleri

Parametre	Xception & DenseNet201
Optimizasyon algoritması	Adam
Öğrenme katsayısı (<i>learning_rate</i>)	0.0001
Erken durdurma (<i>val_accuracy</i>)	Var
Erken durdurma limiti (<i>patience</i>)	10
Maliyet fonksiyonu (<i>loss</i>)	categorical_crossentropy
Girdi boyutu (<i>input_shape</i>)	224x224
Sınıflandırıcı AF	Softmax
Ağırlık (<i>weights</i>)	ImageNet

Her iki modelin BCN20000 veri seti ile eğitimi sonunda Xception modeli 45, DenseNet201 modeli 48 dönemde eğitimini tamamladı. Modellerin eğitiminde erken durdurma tekniği uygulandığı için modellerin eğitim dönem sayıları farklı çıkmıştır. Eğitim sırasında doğrulama başarısı *val_accuracy* metriği bakımından Xception modeli ile en yüksek 0.879, DenseNet201 modeli ile 0.873 değerleri elde edilmiştir. Kayıp değeri *val_loss* metriği bakımından Xception modeli ile en düşük 0.475, DenseNet201 modeli ile en düşük 0.520 değerleri elde edilmiştir. Xception modelinin daha düşük kayıp değerine ulaştığı görülmüştür. BCN20000 ile ara eğitim işleminden geçirilmiş ağ modelleri ile HAM10000 veri setini yedi cilt hastalığı türüne sınıflandırmak için, bu modeller HAM10000 veri seti ile tekrar eğitilmiş ve test edilmiştir (Şekil 3.10 alt satır). Her bir veri seti için, önerilen ara öğrenme ile transfer öğrenme yönteminde gerçekleştirilen deney sayıları Tablo 3.10’ daki gibidir.

Tablo 3.10: Ara öğrenme ile transfer öğrenme yönteminde deney sayıları

Model	BCN20000		HAM10000	
	Veri artırma var	Veri artırma yok	Veri artırma var	Veri artırma yok
Xception	1	-	10	10
DenseNet201	1	-	10	10

Birinci grupta yalnızca ImageNet ağırlıkları kullanılan deneylerde her iki model için 20’şer kez olmak üzere toplamda 40 model eğitimi gerçekleştirilmişken, önerilen yöntemde 42 kez model eğitimi işlemi gerçekleştirilmiştir. İki adet farklılık, modellerin BCN20000 veri seti ile ara eğitim aşamasında geçirilmesinden kaynaklanmaktadır.

HAM10000 veri setinin sınıflandırılmasında, değerlendirme için her bir onlu deney grubunun ortalaması alındı. Modellerin HAM10000 veri seti ile eğitilmesi için kullanılan hiper parametreler ve değerleri Tablo 3.11’deki gibidir. Doğrudan ImageNet ağırlıkları kullanılan deneylerdeki model eğitimi için hiper parametreleri gösteren Tablo 3.7 ile arasındaki tek fark ağırlık parametresindedir.

Tablo 3.11: Ara öğrenme ile transfer öğrenme yönteminde modellerin HAM10000 veri seti ile eğitilmesi

Parametre	Xception & DenseNet201
Optimizasyon algoritması	Adam
Öğrenme katsayısı (<i>learning_rate</i>)	0.0001
Erken durdurma (<i>val_accuracy</i>)	Var
Erken durdurma limiti (<i>patience</i>)	8
Maliyet fonksiyonu (<i>loss</i>)	categorical_crossentropy
Girdi boyutu (<i>input_shape</i>)	224x224
Sınıflandırıcı AF	Softmax
Ağırlık (<i>weights</i>)	İmagenet + bcn20000

3.4. DENEYSEL SONUÇLAR

Doğrudan ImageNet ağırlıkları transfer edilerek gerçekleştirilen dört alt deney grubuna ilişkin sonuçlar ile, BCN20000 veri seti ile ara öğrenme gerçekleştirilerek uygulanan transfer öğrenme (önerilen) yöntemindeki dört alt deney grubuna ilişkin sonuçlar karşılaştırmalı olarak sunulmuştur. Karşılaştırılan deney grupları ve her bir deney grubunda HAM10000 veri setinden oluşturulan on farklı eğitim, doğrulama ve test veri kümeleri ile gerçekleştirilen deney sayıları Tablo 3.12'deki gibidir. Karşılaştırılan alt deney grupları aynı renkle gösterilmiştir.

Tablo 3.12: Karşılaştırılan alt deney grupları

Model	Doğrudan ImageNet ile TÖ		Ara Öğrenme ile TÖ (Önerilen Yöntem)	
	Veri artırma YOK	Veri artırma VAR	Veri artırma YOK	Veri artırma VAR
Xception	10	10	10	10
DenseNet201	10	10	10	10

Çalışmada, performans değerlendirmesi için doğruluk (*accuracy*) ölçütünün yanı sıra, örnek sayısı bakımından dengesiz veri setlerinin değerlendirilmesinde yaygın olarak kullanılan kesinlik (*precision*), duyarlılık (*sensitivity*, *recall* olarak da adlandırılır), f1-skoru (*f1-score*) ölçütleri de kullanılmıştır. Değerlendirme metriklerine ait hesaplamalar Denklem 3.1, 3.2, 3.3 ve 3.4' teki gibidir.

$$\text{Doğruluk} = \frac{TP + TN}{TP + TN + FP + FN} \quad (3.1)$$

$$\text{Kesinlik} = \frac{TP}{TP + FP} \quad (3.2)$$

$$\text{Duyarlılık} = \frac{TP}{TP + FN} \quad (3.3)$$

$$\text{F1 - skoru} = \frac{2 * \text{Kesinlik} * \text{Duyarlılık}}{\text{Kesinlik} + \text{Duyarlılık}} \quad (3.4)$$

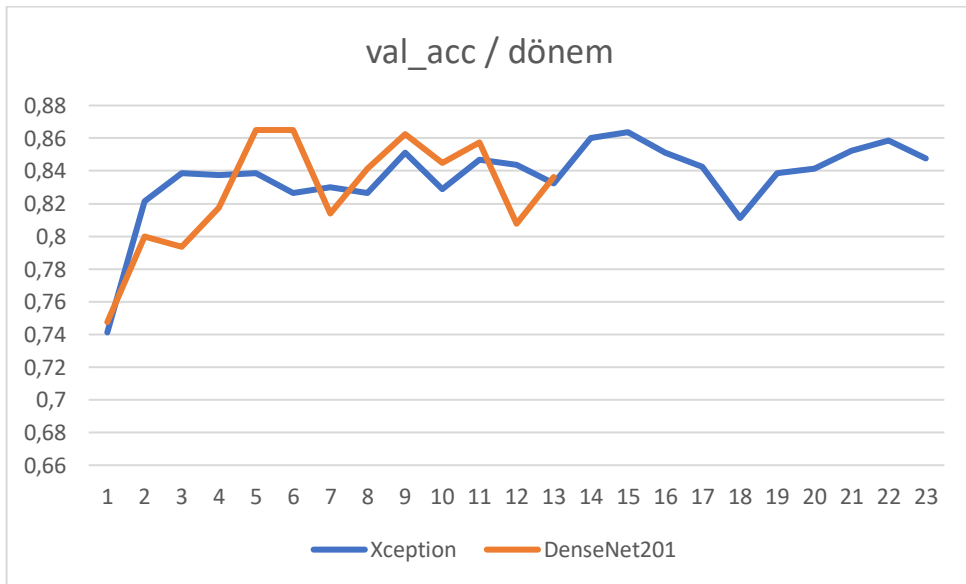
Burada TP ve TN tahmin edilenin gerçek ile aynı olduğu durumları, FP ve FN ise tahmin edilenin gerçekten farklı olduğu durumları gösterir. Doğruluk, duyarlılık ve f1-puanı ölçütlerinin hem makro (aritmetik) ortalamaları hem de ağırlıklı ortalamaları ayrı ayrı sunuldu ve her bir değerlendirme ölçütü, deney gruplarında gerçekleştirilen 10 farklı deneyin ortalamasına göre hesaplandı. Ayrıca gerçekleştirilen model eğitimi deneylerinin ilerleyişini göstermek üzere, her deney grubunda gerçekleştirilen 10 adet deney içerisinde örnek olarak seçilen beşinci deneye ait doğrulama başarımı (*val_accuracy*) / dönem (*epoch*) ve doğrulama kayıp değeri (*val_loss*) / dönem (*epoch*) grafikleri de sunuldu. Model eğitimlerinde erken durdurma tekniği uygulandığı için, grafiklerde Xception ve DenseNet201 modelleri için farklı dönem sayıları oluşmaktadır.

3.4.1. Doğrudan ImageNet Ağırlıkları ile Gerçekleştirilen Transfer Öğrenme Deneylerine Ait Sonuçlar

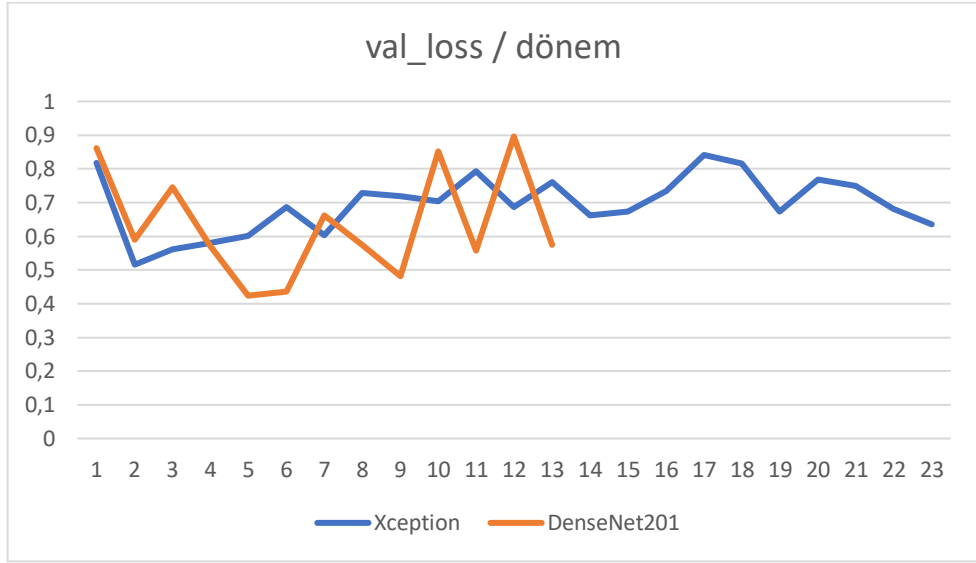
3.4.1.1. Veri Artırma Uygulanmadan Elde Edilen Sonuçlar

ImageNet ile ön eğitilmiş Xception ve DenseNet201 ESA modelleri kullanılarak oluşturulan ağ modellerinin, veri artırma uygulanmadan HAM10000 veri seti ile eğitilmesi ve bu veri setinin yedi hastalık türüne sınıflandırılmasına ilişkin eğitim sürecini gösteren grafikler Şekil 3.11 ve 3.12’ de gösterilmektedir. Xception modelinde ortalama eğitim dönemi sayısı 17.9, DenseNet201 modelinde 18.7 olarak gerçekleşmiştir. Eğitim dönemlerindeki farklılık, modellerin eğitiminde erken durdurma tekniği uygulanmış olmasından kaynaklanmaktadır.

Xception modelinde elde edilen en yüksek *val_accuracy* değeri 0.864, en düşük *val_loss* değeri 0.511, ortalama *val_accuracy* değeri 0.848 ve ortalama *val_loss* değeri ise 0.550 olarak gerçekleşmiştir. DenseNet201 modelinde ise elde edilen en yüksek *val_accuracy* değeri 0.878, en düşük *val_loss* değeri 0.424, ortalama *val_accuracy* değeri 0.856 ve ortalama *val_loss* değeri ise 0.527 olarak gerçekleşmiştir.

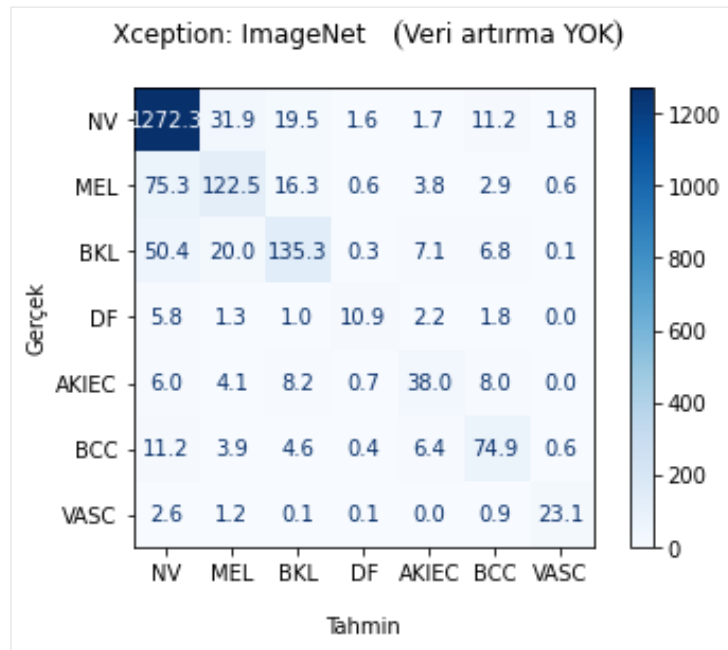


Şekil 3.11: Xception ve DenseNet201 *val_accuracy* / dönem grafiği



Şekil 3.12: Xception ve DenseNet201 val_loss / dönem grafiği

Her iki model için gerçekleştirilen 10 deneyin ortalaması ile elde edilen karışıklık matrisleri Şekil 3.13 ve 3.14’ de, değerlendirme ölçütlerine ilişkin veriler ise Tablo 3.13 ve 3.14’ de gösterilmektedir.

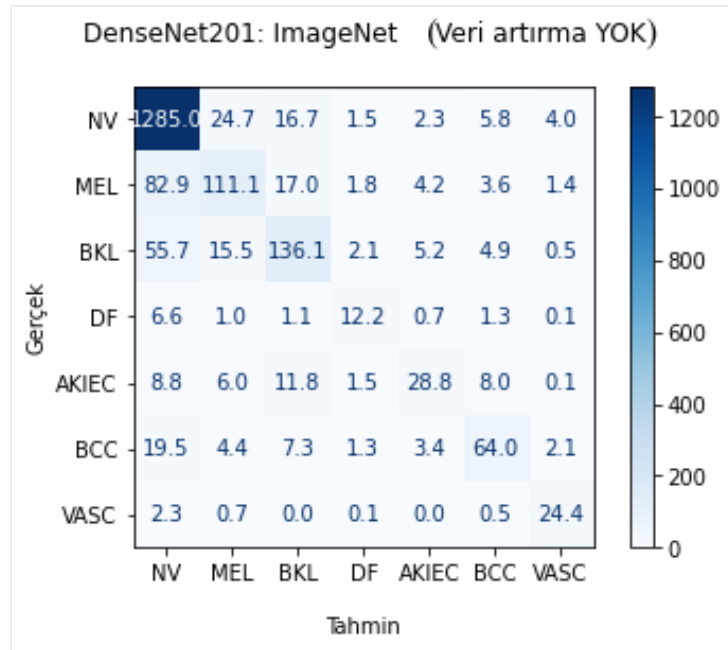


Şekil 3.13: Xception için karışıklık matrisi

Tablo 3.13: Xception için sayısal sonuçlar

	Kesinlik	Duyarlılık	F1-skoru
NV	0.894	0.949	0.921
MEL	0.663	0.552	0.602
BKL	0.731	0.615	0.668
DF	0.747	0.474	0.580
AKIEC	0.642	0.585	0.612
BCC	0.703	0.734	0.728
VASC	0.882	0.825	0.852
Doğruluk			0.839
Makro ort.	0.752	0,676	0.708
Ağırlıklı ort.	0.830	0.839	0.832

Xception modelinin sonuçlarında, kesinlik ölçütünde en yüksek değer 0.894 ile NV sınıfında, en düşük değer 0.642 ile AKIEC sınıfında elde edilmiştir. Duyarlılık ölçütünde en yüksek değer 0.949 ile NV sınıfında, en düşük değer 0.474 ile BKL sınıfında elde edilmiştir. F1-skoru bakımından en yüksek değer 0.921 ile NV sınıfında, en düşük değer 0.580 ile DF sınıfında elde edilmiştir. Elde edilen test doğruluğu ise 0.839 olarak gerçekleşmiştir.



Şekil 3.14: DenseNet201 için karışıklık matrisi

Tablo 3.14: DenseNet201 için sayısal sonuçlar

	Kesinlik	Duyarlılık	F1-skoru
NV	0.880	0.959	0.918
MEL	0.680	0.500	0.577
BKL	0.716	0.619	0.664
DF	0.595	0.530	0.561
AKIEC	0.646	0.443	0.526
BCC	0.726	0.627	0.673
VASC	0.748	0.871	0.805
Doğruluk			0.831
Makro ort.	0.713	0.650	0.675
Ağırlıklı ort.	0.819	0.831	0.821

DenseNet201 modelinin sonuçlarında, kesinlik ölçütünde en yüksek değer 0.880 ile NV sınıfında, en düşük değer 0.595 ile DF sınıfında elde edilmiştir. Duyarlılık ölçütünde en yüksek değer 0.959 ile NV sınıfında, en düşük değer 0.443 ile AKIEC sınıfında elde edilmiştir. F1-skoru bakımından en yüksek değer 0.918 ile NV sınıfında, en düşük değer 0.526 ile AKIEC sınıfında elde edilmiştir. Elde edilen *test doğruluğu* ise 0.831 olarak gerçekleşmiştir.

Ağ modellerinin doğrudan ImageNet ağırlıkları kullanılarak eğitilmesi deneylerinde veri artırma işlemi uygulanmadan elde edilen sonuçlarda, ortalama *test doğruluğu* bakımından Xception ile 0.839, DenseNet201 modelinde 0.831 değeri elde edilmiştir. Xception modeli ile %0.8 oranında daha iyi bir sonuç elde edilmiştir. Her iki modelin karşılaştırması Tablo 3.15’ deki gibidir.

Tablo 3.15: Xception – DenseNet201 performans sonuçları

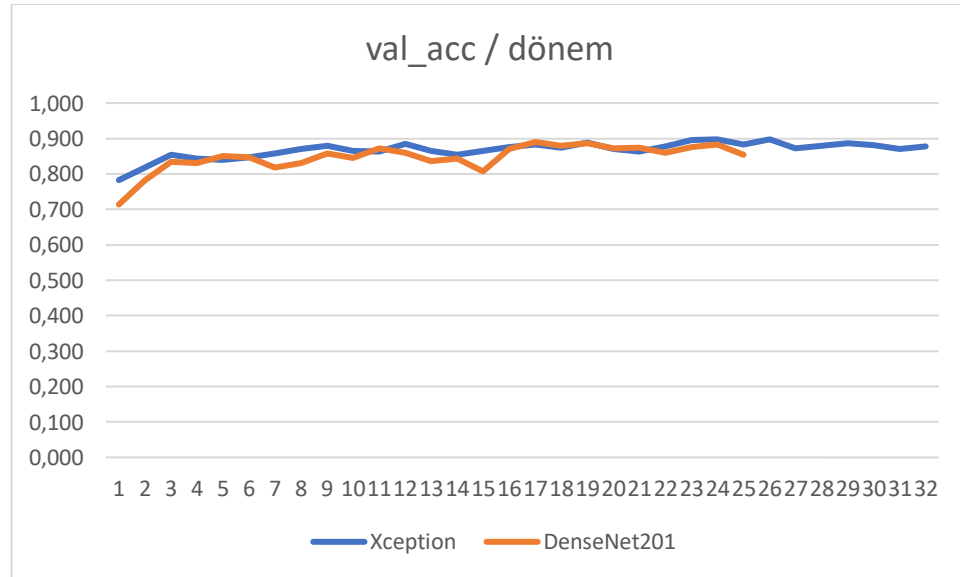
Model	Test doğruluğu	Kesinlik ort.		Duyarlılık ort.		F1-skoru ort.	
		Makro	Ağırlıklı	Makro	Ağırlıklı	Makro	Ağırlıklı
Xception	0.839	0.752	0.830	0.676	0.839	0.708	0.832
DenseNet201	0.831	0.713	0.819	0.650	0.831	0.675	0.821

Tablo 3.15’e göre, Xception modelinin DenseNet201 modeline göre bütün ölçütler bakımından daha iyi sonuç verdiği görülmektedir (bkz. Bölüm 3.4.3).

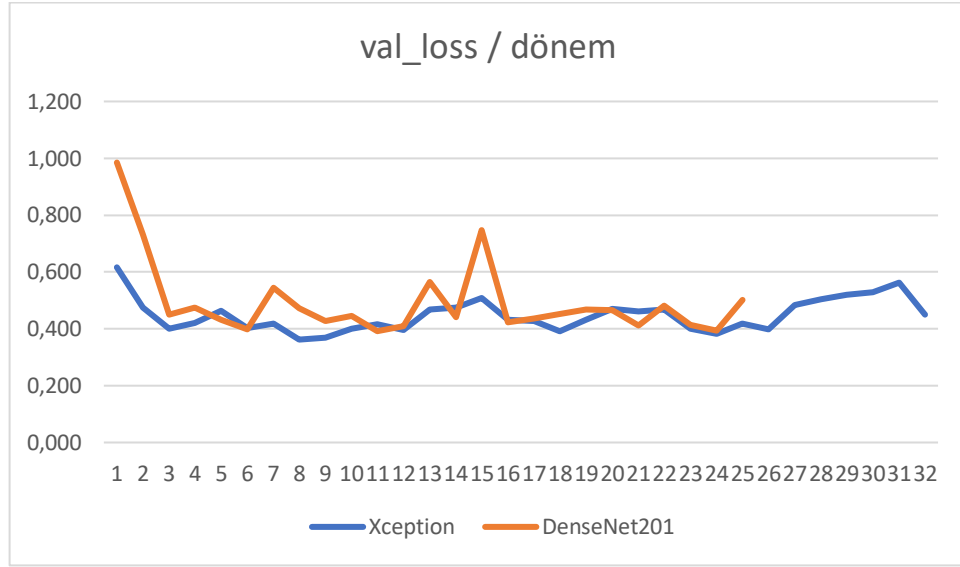
3.4.1.2. Veri Artırma Uygulanarak Elde Edilen Sonuçlar

ImageNet ile ön eğitilmiş Xception ve DenseNet201 ESA modelleri kullanılarak oluşturulan ağ modellerinin, veri artırma uygulanarak HAM10000 veri seti ile eğitilmesi ve bu veri setinin yedi hastalık türüne sınıflandırılmasına ilişkin eğitim sürecini gösteren grafikler Şekil 3.15 ve 3.16’ da gösterilmektedir. Xception modelinde ortalama eğitim dönemi sayısı 28.8, DenseNet201 modelinde 30.3 olarak gerçekleşmiştir. Eğitim dönemlerindeki farklılık, modellerin eğitiminde erken durdurma tekniği uygulanmış olmasından kaynaklanmaktadır.

Xception modelinde elde edilen en yüksek *val_accuracy* değeri 0.898, en düşük *val_loss* değeri 0.353, ortalama *val_accuracy* değeri 0.887 ve ortalama *val_loss* değeri ise 0.397 olarak gerçekleşmiştir. DenseNet201 modelinde ise elde edilen en yüksek *val_accuracy* değeri 0.898, en düşük *val_loss* değeri 0.384, ortalama *val_accuracy* değeri 0.879 ve ortalama *val_loss* değeri ise 0.423 olarak gerçekleşmiştir.

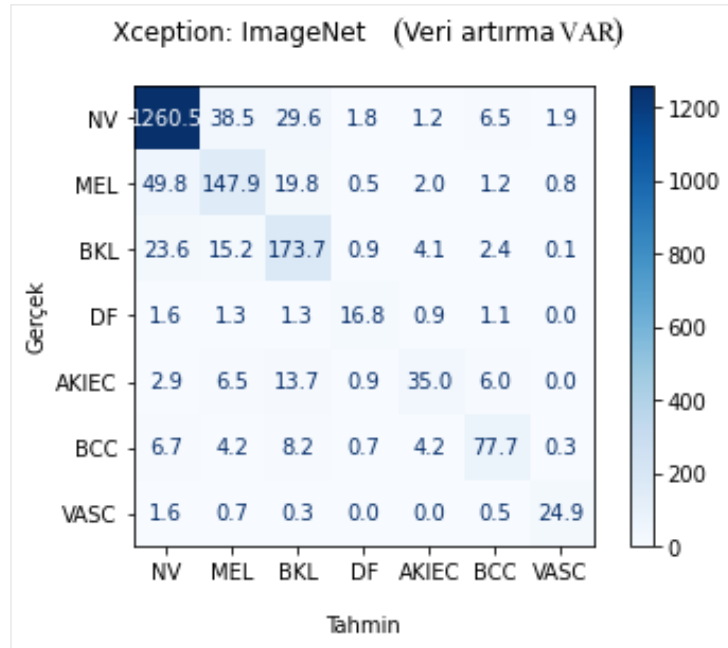


Şekil 3.15: Xception ve DenseNet201 *val_accuracy* / dönem grafiği



Şekil 3.16: Xception ve DenseNet201 val_loss / dönem grafiği

Her iki model için gerçekleştirilen 10 deneyin ortalaması ile elde edilen karışıklık matrisleri Şekil 3.17 ve 3.18’ de, değerlendirme ölçütlerine ilişkin veriler ise Tablo 3.16 ve 3.17’ de gösterilmektedir.

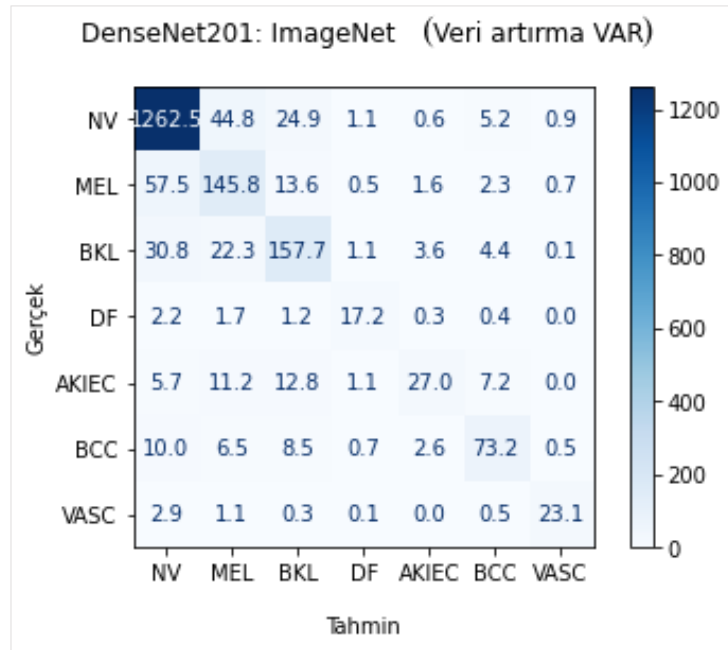


Şekil 3.17: Xception için karışıklık matrisi

Tablo 3.16: Xception için sayısal sonuçlar

	Kesinlik	Duyarlılık	F1-skoru
NV	0.936	0.941	0.938
MEL	0.690	0.666	0.678
BKL	0.704	0.790	0.745
DF	0.778	0.730	0.753
AKIEC	0.738	0.538	0.623
BCC	0.814	0.762	0.787
VASC	0.889	0.889	0.889
Doğruluk			0.868
Makro ort.	0.793	0.759	0.773
Ağırlıklı ort.	0.868	0.868	0.867

Xception modelinin sonuçlarında, kesinlik ölçütünde en yüksek değer 0.936 ile NV sınıfında, en düşük değer 0.690 ile MEL sınıfında elde edilmiştir. Duyarlılık ölçütünde en yüksek değer 0.941 ile NV sınıfında, en düşük değer 0.538 ile AKIEC sınıfında elde edilmiştir. F1-skoru bakımından en yüksek değer 0.938 ile NV sınıfında, en düşük değer 0.623 ile AKIEC sınıfında elde edilmiştir. Elde edilen *test doğruluğu* ise 0.868 olarak gerçekleşmiştir.



Şekil 3.18: DenseNet201 için karışıklık matrisi

Tablo 3.17: DenseNet201 için sayısal sonuçlar

	Kesinlik	Duyarlılık	F1-skoru
NV	0.920	0.942	0.931
MEL	0.625	0.657	0.640
BKL	0.720	0.717	0.718
DF	0.789	0.748	0.768
AKIEC	0.756	0.415	0.536
BCC	0.785	0.718	0.750
VASC	0.913	0.825	0.867
Doğruluk			0.853
Makro ort.	0.787	0.717	0.744
Ağırlıklı ort.	0.852	0.853	0.851

DenseNet201 modelinin sonuçlarında, kesinlik ölçütünde en yüksek değer 0.920 ile NV sınıfında, en düşük değer 0.625 ile MEL sınıfında elde edilmiştir. Duyarlılık ölçütünde en yüksek değer 0.942 ile NV sınıfında, en düşük değer 0.415 ile AKIEC sınıfında elde edilmiştir. F1-skoru bakımından en yüksek değer 0.931 ile NV sınıfında, en düşük değer 0.536 ile AKIEC sınıfında elde edilmiştir. Elde edilen *test doğruluğu* ise 0.853 olarak gerçekleşmiştir.

Ağ modellerinin doğrudan ImageNet ağırlıkları kullanılarak eğitilmesi deneylerinde veri artırma işlemi uygulanarak elde edilen sonuçlarda, ortalama *test doğruluğu* bakımından Xception ile 0.868, DenseNet201 modelinde 0.853 değeri elde edilmiştir. Xception modeli ile %1.5 oranında daha iyi bir sonuç elde edilmiştir. Her iki modelin karşılaştırması Tablo 3.18’ deki gibidir.

Tablo 3.18: Xception – DenseNet201 performans sonuçları

Model	Test doğruluğu	Kesinlik ort.		Duyarlılık ort.		F1-skoru ort.	
		Makro	Ağırlıklı	Makro	Ağırlıklı	Makro	Ağırlıklı
Xception	0.868	0.793	0.868	0.759	0.868	0.773	0.867
DenseNet201	0.853	0.787	0.852	0.717	0.853	0.744	0.851

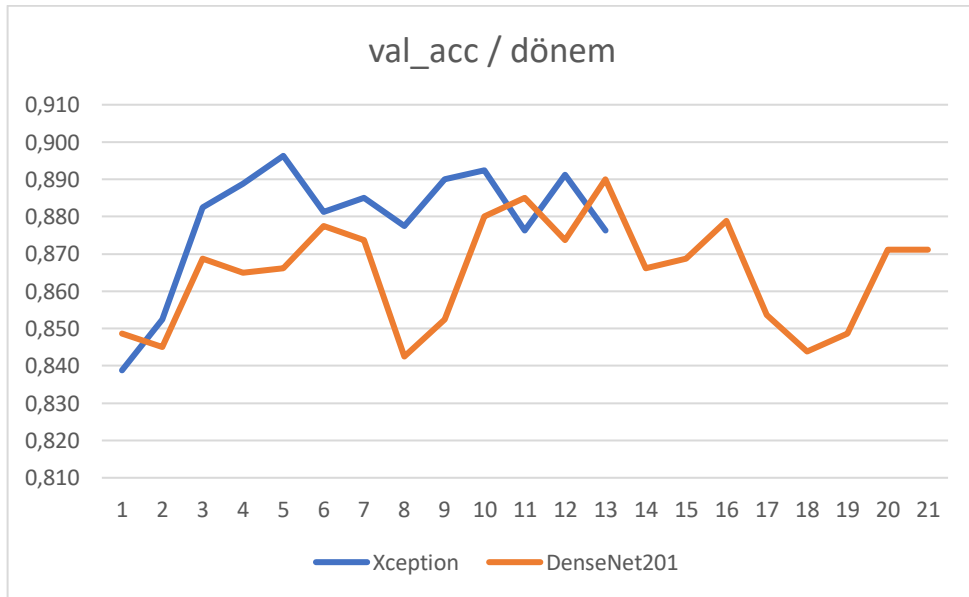
Tablo 3.18’e göre, Xception modelinin DenseNet201 modeline göre bütün ölçütler bakımından daha iyi sonuç verdiği görülmektedir (bkz. Bölüm 3.4.3).

3.4.2. Önerilen Ara Öğrenme ile Transfer Öğrenme Yöntemiyle Gerçekleştirilen Deneylere Ait Sonuçlar

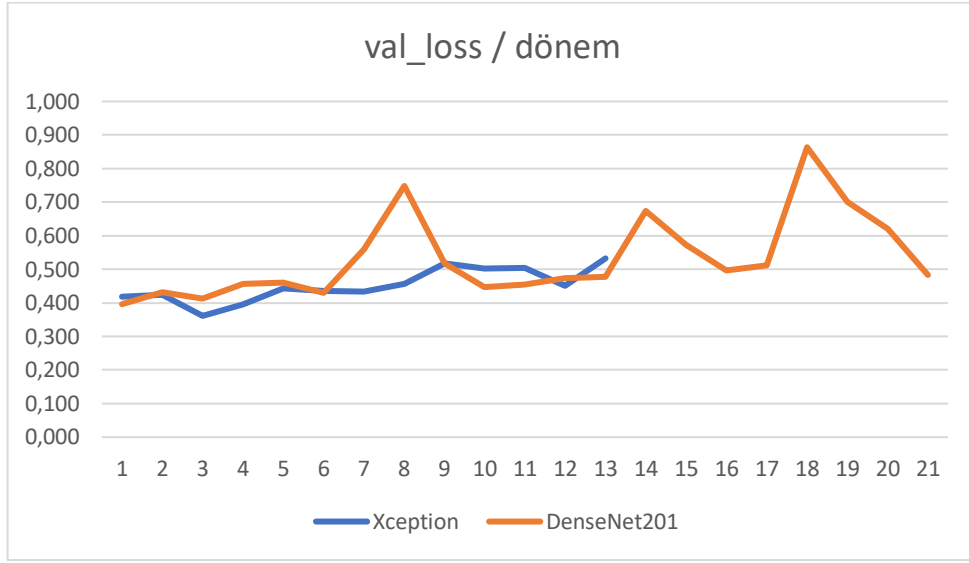
3.4.2.1. Veri Artırma Uygulanmadan Elde Edilen Sonuçlar

ImageNet ile ön eğitilmiş Xception ve DenseNet201 ESA modelleri kullanılarak oluşturulan ağ modellerinin, BCN20000 veri veri ile eğitildikten sonra transfer edilmesiyle, veri artırma uygulanmadan HAM10000 veri seti ile eğitilmesi ve bu veri setinin yedi hastalık türüne sınıflandırılmasına ilişkin eğitim sürecini gösteren grafikler Şekil 3.19 ve 3.20’de gösterilmektedir. Xception modelinde ortalama eğitim dönemi sayısı 18.0, DenseNet201 modelinde 16.6 olarak gerçekleşmiştir. Eğitim dönemlerindeki farklılık, modellerin eğitiminde erken durdurma tekniği uygulanmış olmasından kaynaklanmaktadır.

Xception modelinde elde edilen en yüksek *val_accuracy* değeri 0.899, en düşük *val_loss* değeri 0.361, ortalama *val_accuracy* değeri 0.881 ve ortalama *val_loss* değeri ise 0.425 olarak gerçekleşmiştir. DenseNet201 modelinde elde edilen en yüksek *val_accuracy* değeri 0.899, en düşük *val_loss* değeri 0.341, ortalama *val_accuracy* değeri 0.882 ve ortalama *val_loss* değeri ise 0.406 olarak gerçekleşmiştir.

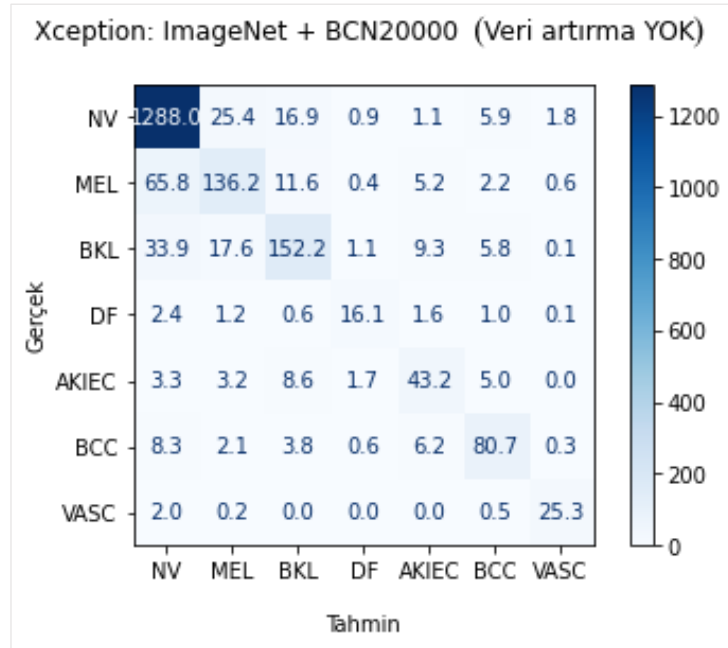


Şekil 3.19: Xception ve DenseNet201 *val_accuracy* / dönem grafiği



Şekil 3.20: Xception ve DenseNet201 val_loss / dönem grafiği

Her iki model için gerçekleştirilen 10 deneyin ortalaması ile elde edilen karışıklık matrisleri Şekil 3.21 ve 3.22’ de, değerlendirme ölçütlerine ilişkin veriler ise Tablo 3.19 ve 3.20’ de gösterilmektedir.

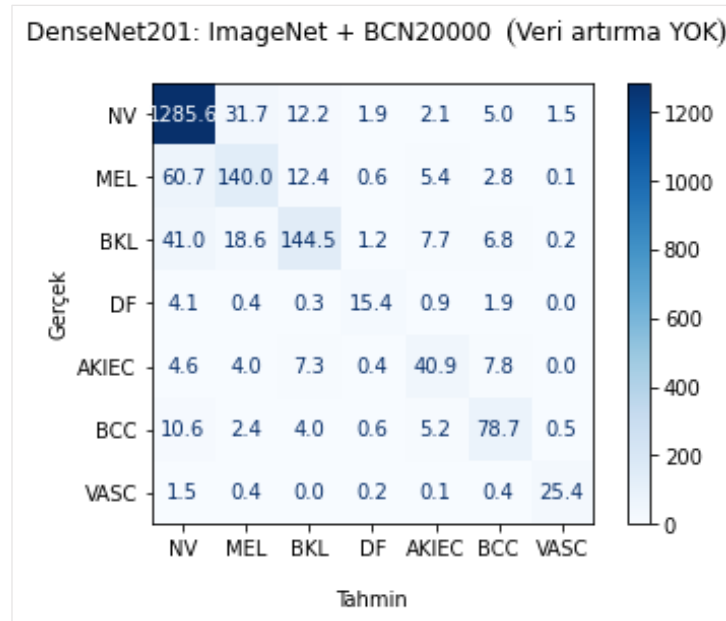


Şekil 3.21: Xception için karışıklık matrisi

Tablo 3.19: Xception için sayısal sonuçlar

	Kesinlik	Duyarlılık	F1-skoru
NV	0.918	0.961	0.939
MEL	0.733	0.614	0,668
BKL	0.786	0.692	0,736
DF	0.774	0.700	0,735
AKIEC	0.649	0.665	0.657
BCC	0.798	0.791	0.795
VASC	0.897	0.904	0.900
Doğruluk			0.871
Makro ort.	0.793	0.761	0.776
Ağırlıklı ort.	0.866	0.871	0.867

Xception modelinin sonuçlarında, kesinlik ölçütünde en yüksek değer 0.918 ile NV sınıfında, en düşük değer 0.649 ile AKIEC sınıfında elde edilmiştir. Duyarlılık ölçütünde en yüksek değer 0.961 ile NV sınıfında, en düşük değer 0.614 ile MEL sınıfında elde edilmiştir. F1-skoru bakımından en yüksek değer 0.939 ile NV sınıfında, en düşük değer 0.657 ile AKIEC sınıfında elde edilmiştir. Elde edilen test doğruluğu ise 0.871 olarak gerçekleşmiştir.



Şekil 3.22: DenseNet201 için karışıklık matrisi

Tablo 3.20: DenseNet201 için sayısal sonuçlar

	Kesinlik	Duyarlılık	F1-skoru
NV	0.913	0.959	0.936
MEL	0.709	0.631	0.667
BKL	0.800	0.657	0.721
DF	0.759	0.670	0.711
AKIEC	0.657	0.629	0.643
BCC	0.761	0.772	0.766
VASC	0.917	0.907	0.912
Doğruluk			0.865
Makro ort.	0.788	0.746	0.765
Ağırlıklı ort.	0.860	0.865	0.861

DenseNet201 modelinin sonuçlarında, kesinlik ölçütünde en yüksek değer 0.917 ile VASC sınıfında, en düşük değer 0.627 ile AKIEC sınıfında elde edilmiştir. Duyarlılık ölçütünde en yüksek değer 0.959 ile NV sınıfında, en düşük değer 0.629 ile AKIEC sınıfında elde edilmiştir. F1-skoru bakımından en yüksek değer 0.936 ile NV sınıfında, en düşük değer 0.643 ile AKIEC sınıfında elde edilmiştir. Elde edilen *test doğruluğu* ise 0.865 olarak gerçekleşmiştir.

Ara öğrenme ile TÖ uygulanarak gerçekleştirilen deneylerde veri artırma işlemi uygulanmadan elde edilen sonuçlarda, *test doğruluğu* bakımından Xception ile 0.871, DenseNet201 modelinde ise 0.865 değeri elde edilmiştir. Xception modeli ile %0.6 oranında daha iyi bir sonuç elde edilmiştir. Her iki modelin karşılaştırması Tablo 3.21’deki gibidir.

Tablo 3.21: Xception – DenseNet201 performans sonuçları

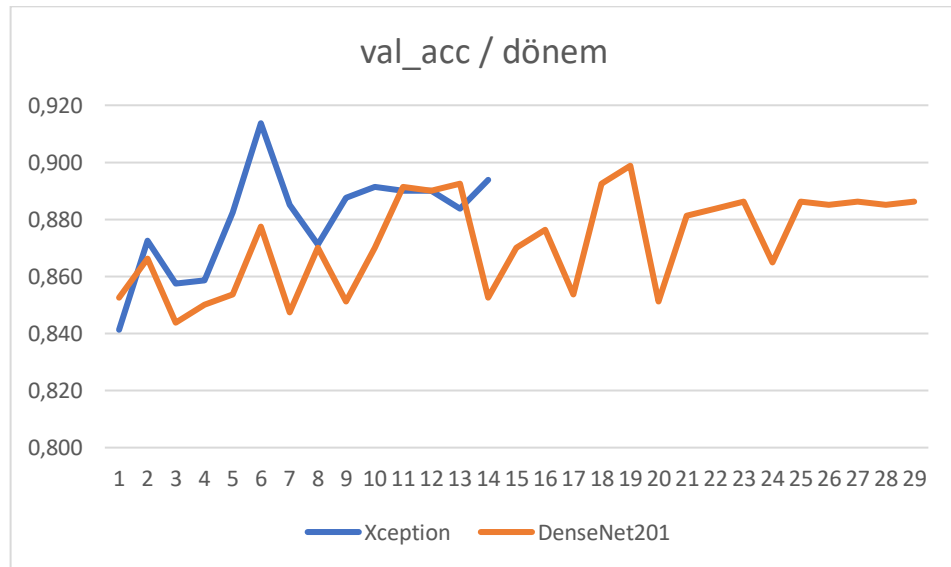
Model	Test doğruluğu	Kesinlik ort.		Duyarlılık ort.		F1-skoru ort.	
		Makro	Ağırlıklı	Makro	Ağırlıklı	Makro	Ağırlıklı
Xception	0.871	0.793	0.866	0.761	0.871	0.776	0.867
DenseNet201	0.865	0.788	0.860	0.746	0.865	0.765	0.861

Tablo 3.21’e göre, Xception modelinin DenseNet201 modeline göre bütün ölçütler bakımından daha iyi sonuç verdiği görülmektedir (bkz. Bölüm 3.4.3).

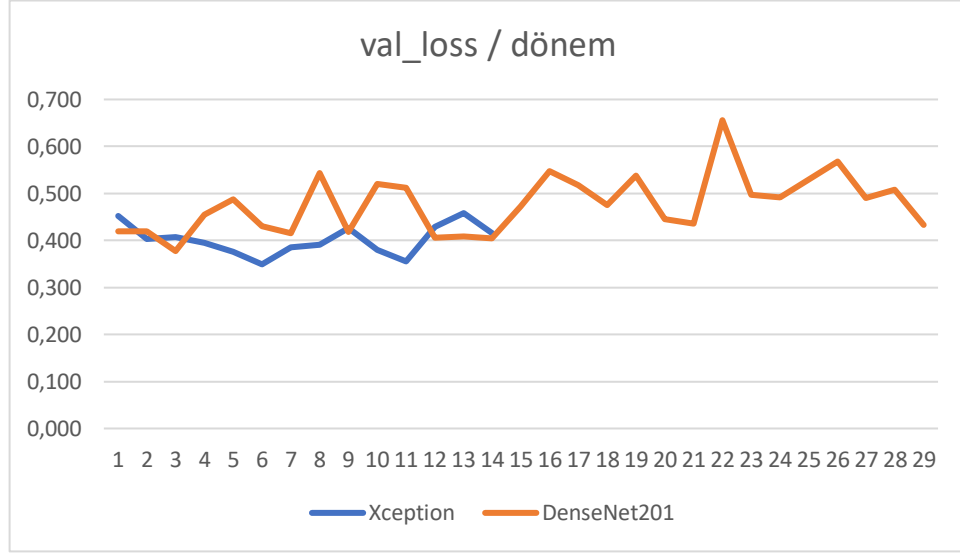
3.4.2.2. Veri Artırma Uygulanarak Elde Edilen Sonuçlar

ImageNet ile ön eğitilmiş Xception ve DenseNet201 ESA modelleri kullanılarak oluşturulan ağ modellerinin, BCN20000 veri seti ile eğitildikten sonra transfer edilmesiyle, veri artırma uygulanarak HAM10000 veri seti ile eğitilmesi ve bu veri setinin yedi hastalık türüne sınıflandırılmasına ilişkin eğitim sürecini gösteren grafikler Şekil 3.23 ve 3.24' de gösterilmektedir. Xception modelinde ortalama eğitim dönemi sayısı 21.5, DenseNet201 modelinde 21.3 olarak gerçekleşmiştir. Eğitim dönemlerindeki farklılık, modellerin eğitiminde erken durdurma tekniği uygulanmış olmasından kaynaklanmaktadır.

Xception modelinde elde edilen en yüksek *val_accuracy* değeri 0.914, en düşük *val_loss* değeri 0.349, ortalama *val_accuracy* değeri 0.888 ve ortalama *val_loss* değeri ise 0.390 olarak gerçekleşmiştir. DenseNet201 modelinde ise elde edilen en yüksek *val_accuracy* değeri 0.910, en düşük *val_loss* değeri 0.348, ortalama *val_accuracy* değeri 0.883 ve ortalama *val_loss* değeri ise 0.402 olarak gerçekleşmiştir.

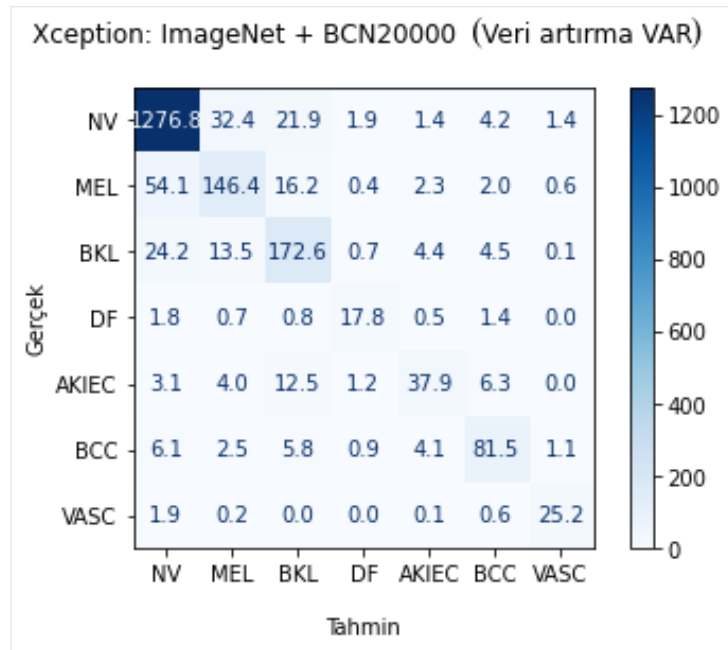


Şekil 3.23: Xception ve DenseNet201 *val_accuracy* / dönem grafiği



Şekil 3.24: Xception ve DenseNet201 val_loss / dönem grafiği

Her iki model için gerçekleştirilen 10 deneyin ortalaması ile elde edilen karışıklık matrisleri Şekil 3.25 ve 3.26’ da, değerlendirme ölçütlerine ilişkin veriler ise Tablo 3.22 ve 3.23’ de gösterilmektedir.

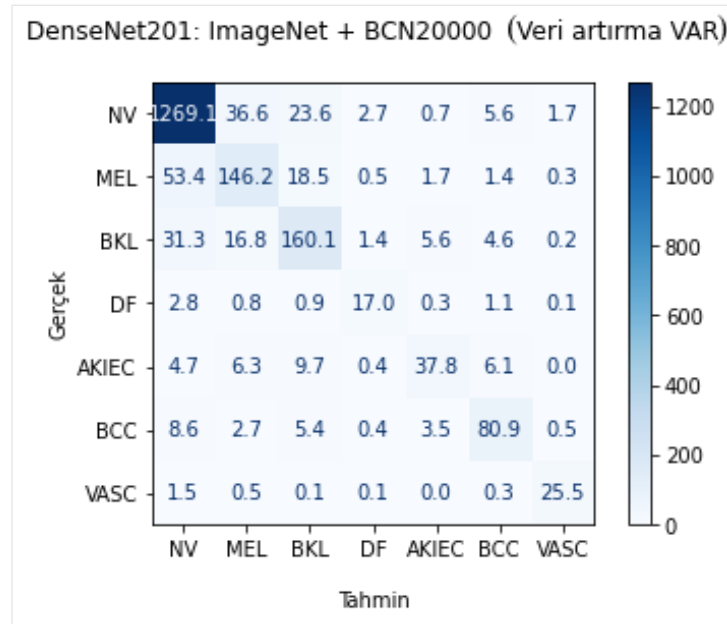


Şekil 3.25: Xception için karışıklık matrisi

Tablo 3.22: Xception için sayısal sonuçlar

	Kesinlik	Duyarlılık	F1-skoru
NV	0.933	0.953	0.943
MEL	0.733	0.659	0.694
BKL	0.751	0.785	0.767
DF	0.777	0.774	0.776
AKIEC	0.748	0.583	0.655
BCC	0.811	0.799	0.805
VASC	0.887	0.900	0.894
Doğruluk			0.879
Makro ort.	0.806	0.779	0.791
Ağırlıklı ort.	0.876	0.879	0.877

Xception modelinin sonuçlarında, kesinlik ölçütünde en yüksek değer 0.933 ile NV sınıfında, en düşük değer 0.733 ile MEL sınıfında elde edilmiştir. Duyarlılık ölçütünde en yüksek değer 0.953 ile NV sınıfında, en düşük değer 0.583 ile AKIEC sınıfında elde edilmiştir. F1-skorunda en yüksek 0.943 ile NV, en düşük 0.655 ile AKIEC sınıfında elde edilmiş ve modelin *test doğruluğu* ise 0.879 olarak gerçekleşmiştir.



Şekil 3.26: DenseNet201 için karışıklık matrisi

Tablo 3.23: DenseNet201 için sayısal sonuçlar

	Kesinlik	Duyarlılık	F1-skoru
NV	0.925	0.947	0.936
MEL	0.697	0.659	0.677
BKL	0.733	0.728	0.731
DF	0.756	0.739	0.747
AKIEC	0.762	0.582	0.660
BCC	0.809	0.793	0.801
VASC	0.901	0.911	0.906
Doğruluk			0.868
Makro ort.	0.798	0.765	0.780
Ağırlıklı ort.	0.865	0.868	0.866

DenseNet201 modelinin sonuçlarında, kesinlik ölçütünde en yüksek değer 0.925 ile NV sınıfında, en düşük değer 0.697 ile MEL sınıfında elde edilmiştir. Duyarlılık ölçütünde en yüksek değer 0.947 ile NV sınıfında, en düşük değer 0.582 ile AKIEC sınıfında elde edilmiştir. F1-skoru bakımından en yüksek değer 0.936 ile NV sınıfında, en düşük değer 0.660 ile AKIEC sınıfında elde edilmiştir. Elde edilen *test doğruluğu* ise 0.868 olarak gerçekleşmiştir.

Önerilen ara öğrenme ile TÖ uygulanarak gerçekleştirilen deneylerde veri artırma işlemi uygulanarak elde edilen sonuçlarda, *test doğruluğu* bakımından Xception ile 0.879, DenseNet201 modelinde 0.868 değeri elde edilmiştir. Xception modeli ile % 1.1 oranında daha iyi bir sonuç elde edilmiştir. Her iki modelin karşılaştırması Tablo 3.24’deki gibidir.

Tablo 3.24: Xception – DenseNet201 karşılaştırması

Model	Test doğruluğu	Kesinlik ort.		Duyarlılık ort.		F1-skoru ort.	
		Makro	Ağırlıklı	Makro	Ağırlıklı	Makro	Ağırlıklı
Xception	0.879	0.806	0.876	0.779	0.879	0.791	0.877
DenseNet201	0.868	0.798	0.865	0.765	0.868	0.780	0.866

Tablo 3.24’te görüldüğü gibi, Xception modelinin DenseNet201 modeline göre bütün ölçütler bakımından daha iyi sonuç verdiği görülmektedir (bkz. Bölüm 3.4.3).

3.4.3. Deneysel Sonuçların Özeti ve Literatürdeki Sonuçlar ile Karşılaştırma

Bu bölümde bu çalışmada elde edilen deneysel sonuçların, eğitim aşamasında elde edilen en yüksek doğrulama başarımı (*val_acc*), test aşamasında elde edilen sınıf bazında en yüksek kesinlik (*pre*), duyarlılık (*sen/rec*), f1-skoru (*f1-score*) değerleri ile bunların makro ve ağırlıklı ortalamaları ve test doğruluğu (*test_acc*) değerleri sunulmaktadır. Deneysel gruplarındaki sonuçların karşılaştırmalı özeti Tablo 3.25'te sunulmuştur. Sınıf bazında en yüksek *kesinlik*, *duyarlılık* ve *f1-skoru* değerleri büyük çoğunlukla önerilen ara öğrenme ile transfer öğrenme uygulanarak gerçekleştirilen deneylerde veri artırma tekniği uygulanarak elde edilmiştir. NV ve VASC sınıfında yüksek değerler elde edilirken, bunların dışındaki sınıflarda elde edilen oranlar genel olarak test doğruluğu değerinin altında kaldı. MEL ve AKIEC sınıflarının değerlerinin diğer sınıflara göre daha düşük olduğu görülmüştür.

Xception modelinin, DenseNet201 modeline göre genel olarak daha iyi sonuçlar ürettiği görülmüştür. Bu durumun, her iki ESA modelinin parametre sayılarındaki farklılıktan ve Xception modelinin derinlemesine ayrılabilir evrişim yönteminden kaynaklandığı düşünülmektedir. Xception modeli ile elde edilen *test_acc* değeri 0.879 olurken, DenseNet201 modeli ile 0.868 olmuştur. Böylece Xception, DenseNet201'e göre %1.1 daha iyi test başarısı göstermiştir. Eğitim sırasında elde edilen en yüksek *val_acc* değerleri önerilen yöntem ile gerçekleştirilen deneylerde veri artırma tekniği uygulanarak elde edilmiş ve Xception modeli ile 0.914, DenseNet201 ile 0.910 değerlerine ulaşılırken, ortalama *val_acc* değerleri Xception ile 0.888 ve DenseNet201 modeli ile 0.883 olarak gerçekleşmiştir.

Önerilen, ara öğrenme ile TÖ yöntemi kullanılarak oluşturulan Xception modelinde veri artırma uygulanmadan gerçekleştirilen deneylerde, doğrudan ImageNet ağırlıkları kullanılan deney grubuna göre %3.2, veri artırma uygulanarak ise %1.1 daha iyi *test_acc* değeri elde edilmiştir. DenseNet201 modelinde ise veri artırma uygulanmadan %3.4, veri artırma uygulanarak %1.5 daha iyi *test_acc* değerleri elde edilmiştir. Böylece, *test doğruluğu*, *kesinlik*, *duyarlılık* ve *f1-skoru* metrikleri bakımından, önerilen yöntem ile, doğrudan ImageNet ağırlıklarının transfer

edilmesiyle oluşturulan ESA modellerinin sınıflandırma başarısına göre daha iyi sonuçlar elde edildiği görülmüştür.

Tablo 3.25: Deneysel çalışmalardan elde edilen sonuçlar

Xception												
Sınıf	Doğrudan ImageNet ile TÖ						BCN20000 ile Ara Öğrenme ile TÖ (Önerilen Yöntem)					
	Veri Artırma Yok			Veri Artırma Var			Veri Artırma Yok			Veri Artırma Var		
	Pre	Rec	f1	Pre	Rec	f1	Pre	Rec	f1	Pre	Rec	f1
NV	0.894	0.949	0.921	0.936	0.941	0.938	0.918	0.961	0.939	0.933	0.953	0.943
MEL	0.663	0.552	0.602	0.690	0.666	0.678	0.733	0.614	0.668	0.733	0.659	0.694
BKL	0.731	0.615	0.668	0.704	0.790	0.745	0.786	0.692	0.736	0.751	0.785	0.767
DF	0.747	0.474	0.580	0.778	0.730	0.753	0.774	0.700	0.735	0.777	0.774	0.776
AKIEC	0.642	0.585	0.612	0.738	0.538	0.623	0.649	0.665	0.657	0.748	0.583	0.655
BCC	0.703	0.734	0.718	0.814	0.762	0.787	0.798	0.791	0.795	0.811	0.799	0.805
VASC	0.882	0.825	0.852	0.889	0.889	0.889	0.897	0.904	0.900	0.887	0.900	0.894
M.Ort.	0.752	0.676	0.708	0.793	0.759	0.773	0.793	0.761	0.776	0.806	0.779	0.791
A.Ort.	0.830	0.839	0.832	0.868	0.868	0.867	0.866	0.871	0.867	0.876	0.879	0.877
Val acc.	maks	0.864		0.898			0.899			0.914		
	ort	0.848		0.886			0.881			0.888		
Test acc.	0.839		0.868			0.871			0.879			
DenseNet201												
Sınıf	Doğrudan ImageNet ile TÖ						BCN20000 ile Ara Öğrenme ile TÖ (Önerilen Yöntem)					
	Veri Artırma Yok			Veri Artırma Var			Veri Artırma Yok			Veri Artırma Var		
	Pre	Rec	f1	Pre	Rec	f1	Pre	Rec	f1	Pre	Rec	f1
NV	0.880	0.959	0.918	0.920	0.942	0.931	0.913	0.959	0.936	0.925	0.947	0.936
MEL	0.680	0.500	0.577	0.625	0.657	0.640	0.709	0.631	0.667	0.697	0.659	0.677
BKL	0.716	0.619	0.664	0.720	0.717	0.718	0.800	0.657	0.721	0.733	0.728	0.731
DF	0.595	0.530	0.561	0.789	0.748	0.768	0.759	0.670	0.711	0.756	0.739	0.747
AKIEC	0.646	0.443	0.526	0.756	0.415	0.536	0.657	0.629	0.643	0.762	0.582	0.660
BCC	0.726	0.627	0.673	0.785	0.718	0.750	0.761	0.772	0.766	0.809	0.793	0.801
VASC	0.748	0.871	0.805	0.913	0.825	0.867	0.917	0.907	0.912	0.901	0.911	0.906
M.Ort.	0.713	0.650	0.675	0.787	0.717	0.744	0.788	0.746	0.765	0.798	0.765	0.780
A.Ort.	0.819	0.831	0.821	0.852	0.853	0.851	0.860	0.865	0.861	0.865	0.868	0.866
Val acc.	maks	0.878		0.898			0.899			0.910		
	ort	0.856		0.879			0.882			0.883		
Test acc.	0.831		0.853			0.865			0.868			

Bu bölümde ayrıca, literatürde HAM10000 veri setinin yedi hastalık türüne sınıflandırılmasıyla (iki, üç, dört hastalık türüne sınıflandırma ile ilgili çalışmalar dikkate alınmamıştır) ilgili yapılmış olan benzer çalışmaların sayısal sonuçlarının yanı sıra, kullanılan yöntem ve teknikler ile bu tez çalışmasında kullanılan yöntem ve tekniklerin benzerlik ve farklılıkları da sunulmuştur. Literatürde yer alan çalışmaların bir kısmı sınıflandırma görevi öncesi, lezyon sınırlarını ortaya çıkaran bölütleme ve daha sonra bölütlenmiş görüntüler kullanılarak sınıflandırma görevi gerçekleştirmektedir. Bu çalışma, ham görüntülerin bölütlenmeden doğrudan sınıflandırılmasına yönelik olduğu için bölütleme içeren çalışmalar karşılaştırmaya dahil edilmemiştir. Bu tez çalışmasının literatürdeki diğer çalışmalarla karşılaştırma detayları Tablo 3.26’da sunulmuştur.

Tablo 3.26: Literatürdeki diğer çalışmalar ve sonuçları ile karşılaştırma

Çiş.	Kesinlik		Duyarlılık		f1-skoru		Test doğr.	Açıklama	Gör. işl.
	Makro ort.	Ağır. ort.	Makro ort.	Ağır. ort.	Makro ort.	Ağır. ort.			
[27]	0.956	-	0.956	-	-	-	0.987	(AlexNet). Veri artırma eğitim öncesi fiziksel. Veri artırma miktarı çok fazla. Test verisi orijinal değil. İnce ayar ile son üç katman hariç dondurulmuş.	-
[30]	0.980	0.980	0.980	0.980	0.980	0.980	0.980	(Inception-ResNet-V2, Inception-V3, DenseNet-201). Veri artırma eğitim öncesi fiziksel. Test verisi orijinal değil. İnce ayar ile bazı özellik katmanlarında ağırlıklar değiştirilmiş ve yoğun katmanlar eklenmiş.	Var
[29]	0.860	0.929	0.770	0.930	0.810	0.932	0.932	(ResNet50 - VGG16, ResNet50 - Xception). Veri artırma eğitim öncesi fiziksel. Test verisinde sınıfların örnek sayıları oransal değil. İnce ayar ile katmanların %70-75’i dondurulmuş.	-

[25]	-	0.900	-	0.913	-	-	-	(MobileNet). Veri artırma eğitim öncesi fiziksel. Test verisinde sınıfların temsil oranları belirsiz. İnce ayar ile katmanların ile %70'i dondurulmuş.	-
Bu çalışma	0.806	0.876	0.779	0.879	0.791	0.877	0.879	(Xception, DenseNet201). Gerçek zamanlı veri artırma. Test verisinde sınıfların temsil oranları dengeli ve test verisi orijinal. Katmanlar üzerinde ince ayar yok, ara öğrenme var.	-
[77]	-	-	-	-	-	-	0.862	Veri artırma var.	-
[26]	-	0.810	-	-	-	0.83	0.852	Veri artırma eğitim sırasında gerçek zamanlı.	Var
[78]	-	-	-	-	-	-	0.851	Veri artırma eğitim sırasında gerçek zamanlı.	Var
[79]	0,582	0.890	0.604	0.830	0.523	0.830	0.832	Veri artırma eğitim öncesi fiziksel.	-
[80]	0.71	-	-	-	-	-	0.820	Veri artırma eğitim öncesi fiziksel.	-
[81]	-	-	-	-	-	-	0.775	Veri artırma eğitim sırasında gerçek zamanlı.	-
[82]	0.519	0.880	0.607	0.740	0.446	0.770	-	Veri artırma var.	Var
[83]	-	-	-	-	-	-	0.76	Veri artırma var.	-
[28]	0.611	-	0.604	-	0.597	-	-	Veri artırma eğitim sırasında gerçek zamanlı. Test verisinde sınıfların örnek sayıları oransal değil, rastgele. Girdi boyutu 75x100.	Var

* Birden fazla ESA modeli ile gerçekleştirilen çalışmalarda, koyu renkli olan ESA modelleri aynı çalışmada test edilen diğer ESA modelinden daha iyi performans göstermiştir.

Bu çalışma, literatürde aynı amaca yönelik yapılan diğer çalışmalarla karşılaştırıldığında, çalışmanın diğer birçok çalışmadan daha iyi *test doğruluğu* ve daha iyi ağırlıklı ortalama *kesinlik*, *duyarlılık* ve *f1-skoru* değerlerine ulaştığı görülmüştür.

SONUÇ VE TARTIŞMA

Bu tez çalışmasının amacı, transfer öğrenme tekniğini kullanarak cilt hastalık görüntülerinin doğru sınıflandırılma performansını geliştirmeye yöneliktir. Bu amaç için, yalnızca ImageNet ile ön eğitilmiş ESA modellerini kullanmak yerine, bu modellerin başka cilt hastalık görüntüleri içeren bir veri seti ile ara eğitimden geçirildikten sonra, cilt hastalık görüntüleri içeren diğer bir veri setinin sınıflandırılması sağlanmıştır. Böylece yalnızca ImageNet ağırlıkları kullanılarak yapılan sınıflandırma çalışmalarına göre, cilt hastalık görüntüleri içeren başka bir veri seti ile ara eğitimden geçirilmiş ESA modelleriyle yapılan çalışmaların performansı karşılaştırılabilmektedir.

Tezin amacı doğrultusunda, ImageNet ile ön eğitilmiş Xception ve DenseNet201 ESA modelleri kullanılarak, HAM10000 veri setine herhangi bir bölütleme işlemi uygulanmadan bu veri setinin yedi cilt hastalık türüne sınıflandırılmasında, yalnızca ImageNet ağırlıkları kullanılarak gerçekleştirilen ESA modelleri eğitim ve sınıflandırma sonuçları ile, BCN20000 veri seti ile eğitilerek bir ara öğrenme aşamasından geçirilmiş modellerin transfer edilmesiyle gerçekleştirilen (önerilen yöntem) ESA modelleri eğitim ve sınıflandırma sonuçları *test doğruluğu*, *kesinlik*, *duyarlılık* ve *f1-skoru* metriklerine göre karşılaştırılmıştır.

Deneyler veri artırma uygulanarak ve veri artırma tekniği uygulanmadan ayrı ayrı gerçekleştirilmiştir. Veri artırma uygulanmadan yapılan deneylerde, önerilen yöntem ile kullanılan Xception modeli ile, doğrudan ImageNet ağırlıkları ile TÖ yöntemiyle kullanılan Xception modeline göre, sırasıyla *test doğruluğu*, ağırlıklı ortalama *kesinlik*, *duyarlılık* ve *f1-skoru* ölçütlerine göre %3.2, %3.6, %3.2, %3.5 ve DenseNet201 modelinde ise %3.4, %4.1, %3.4 ve %4.0 oranında daha iyi sonuçlar elde edilmiştir. Veri artırma uygulanarak yapılan deneylerde ise, önerilen yöntem ile kullanılan Xception modeli ile, doğrudan ImageNet ağırlıkları ile TÖ yöntemiyle kullanılan Xception modeline göre %1.1, %0.8, %1.1, %1.0 ve DenseNet201 modelinde ise %1.5, %1.3, %1.5 ve %1.5 daha iyi sonuçlar elde edilmiştir. Bu sonuçlar, cilt hastalık görüntüleri içeren veri setleri ile ESA modellerinin ara eğitimden geçirildikten sonra, transfer öğrenme yaklaşımıyla transfer edilerek kullanılmasının,

modellerin yalnızca ImageNet ağırlıkları ile TÖ yöntemiyle kullanılan versiyonlarına göre sınıflandırma görevinde önemli oranlarda daha iyi performans gösterdiğini ortaya koymuştur.

Bu tez çalışması kapsamında, *Test doğruluğu*, ağırlıklı ortalama *kesinlik*, *duyarlılık* ve *f1-skoru* ölçütleri bakımından en iyi sonuç, önerilen yöntem olan ara öğrenme ile transfer öğrenme yöntemi kullanılarak ve veri artırma tekniği uygulanarak eğitilen Xception ESA modeli elde edilmiş ve değerler sırasıyla %87.9, %87.6, %87.9 ve %87.7 olarak gerçekleşmiştir. Bu sonuçların literatürdeki birçok çalışmaya göre daha iyi olduğu görülmüştür. Literatürdeki birkaç çalışmada, bu çalışmanın sonuçlarına göre daha iyi sınıflandırma sonuçları açıklanmış ancak bu çalışmalarda bazı teknik farklılıklar bulunmaktadır. [27] ve [30]'daki çalışmalarda, test verisi orijinal ham görüntülerden oluşmamakta, eğitim ve doğrulama verisi gibi artırma işlemi uygulanmış görüntülerden oluşmaktadır. Bu tez çalışmasında ise, test verisi en başta on farklı varyasyonla oluşturulmuş ve test verisine 224x224 piksel boyutlarına ölçeklendirmek dışında başka herhangi bir işlem uygulanmamıştır. Ayrıca [27]'daki çalışmada veri artırma miktarı yüksek miktarlarda gerçekleştirilmiş ve bir sınıfta 80 bin gibi çok yüksek görüntü sayılarına ulaşılmıştır. Bu çalışmada ise daha düşük sayılarda veri artırma uygulanmıştır. [30]'daki çalışmada ilave olarak görüntü işleme yapılmışken bu çalışmada görüntü işleme yapılmamıştır.

Her bir hastalık sınıfı için ayrılmış olan test görüntü sayısı, sayısal sonuçları etkilemektedir. Örneğin bu çalışmada, NV ve VASC sınıflarında yüksek değerler elde edilirken, diğer sınıflarda ve özellikle MEL ve AKIEC sınıflarında daha düşük sonuçlar elde edilmiştir. Eğer model, MEL ve AKIEC sınıflarında daha fazla eğitim verisiyle eğitilmiş ve daha az test verisi ile test edilmiş olsaydı daha iyi sınıflandırma sonuçlarının elde edilmesi beklenebilirdi. Ancak daha gerçekçi sonuçlar elde edebilmek adına her sınıftan test için ayrılan örnek sayısı, bu sınıfların HAM10000 veri setinde başlangıçtaki oranlarına göre belirlenmiştir. Fakat [29]'daki çalışmada bu oranlara dikkat edilmemiştir. Örneğin sınıflandırılma oranları genellikle daha yüksek elde edilen NV hastalık sınıfından %18 test verisi ayrılmışken, sınıflandırılma oranları daha düşük çıkan AKIEC ve MEL hastalık sınıflarından ise %12.5 ve %5.5 oranında test verisi ayrılmıştır. Temsil oranlarındaki bu dengesizlik test sonuçlarının daha

yüksek çıkmasını sağlasa da, bütün olarak HAM10000 veri setinin sınıflandırılmasında sonuçları gerçekten saptırmaktadır. [25]'deki çalışmada ise test verisinde sınıfları temsil eden görüntü oranları veya sayıları ifade edilmemiştir. Örneğin kesinlik (precision) ölçütünde NV hastalık sınıfı için oran çok yüksek elde edilmişken (%97), AKIEC, BCC, DF, MEL sınıflarında çok daha düşük değerler elde edilmiştir. Bu durum, veri setinden oluşturulan test grubunda, sınıfları temsil eden görüntü sayılarının başlangıçtaki oranları ile uyumlu olmadığını göstermiştir. Bunların dışında [25], [27], [29] ve [30]'daki çalışmalarda kullanılan ESA modellerinin özellik çıkarma katmanları değişen oranlarda dondurulmuşken, bu tez çalışmasında transfer edilen ESA modellerinin katmanları dondurulmamış, tüm katmanlar kullanılan veri setleri (ara eğitim için BCN20000, hedef sınıflandırma için HAM10000) ile yeniden eğitilmiştir.

Literatürdeki çalışmaların bir kısmında, görüntü artırma işlemi modellerin eğitimi öncesi fiziksel olarak, bazılarında ise eğitim ve doğrulama verilerinde gerçek zamanlı olarak model eğitimi sırasında gerçekleştirilmiştir. Bu çalışmada, gerçek zamanlı görüntü artırma tekniği uygulanmıştır. Görüntü artırma işleminde sınıfları temsil eden örnek sayılarında bir değişiklik yapılmamıştır. Bu yüzden bazı sınıfları temsil eden örnek sayıları, ESA'nın daha iyi eğitilmesi için gerekli olan görüntü sayılarından uzak kalmıştır. HAM10000 veri seti gibi, her bir hastalık sınıfını temsil eden örnek sayılarında önemli oransal farklılık olan veri setlerinde bu önemli bir sorundur. Gerçek zamanlı görüntü artırma kısmen bu sorunu çözmeye çalışsa da yeterli düzeyde olamamıştır. Bu nedenle [25], [27], [29] ve [30]'daki çalışmalarda olduğu gibi, eğitim verisinde çok yüksek veya yüksek düzeyde ve örnek sayılarının eşit veya çok yakın olmasını sağlayacak bir veri artırma işlemi, ESA modellerinin eğitimi öncesi fiziksel olarak gerçekleştirmek bu çalışma açısından çok daha iyi sınıflandırma sonuçlarının elde edilmesini sağlayabilir. Bununla birlikte, modellerin eğitimi sırasında gerçek zamanlı veri artırma tekniği uygulanırken, sınıfları temsil eden örnek sayılarını dengeleyici bir yöntemi teknik olarak gerçekleştirmek de, bu çalışma açısından çok daha iyi sınıflandırma sonuçlarının elde edilmesini sağlayabilir. Ayrıca, daha gerçekçi sınıflandırma sonuçları elde edebilmek için, veri artırma işlemi uygulanmadan önce veri setinden test verisinin rasgele veya k-katlı çapraz doğrulama

gibi yöntemler uygulayarak farklı alt kümelerini ayırmak ve test verisinin ham halini korumak daha doğru bir yaklaşım olarak kabul edilebilir.

Modellerin eğitimi ile ilgili grafiklere bakıldığında, önerilen yöntem ile gerçekleştirilen deneylerde, eğitim/dönem grafiğinde ilk eğitim dönemi (birinci epoch) sonunda öğrenme düzeyi daha yüksek değerlerden başlarken, kayıp/dönem grafiğinde ise kayıp değeri daha düşük değerlerden başlamaktadır. Bu beklenen bir sonuç olmuştur. Buna göre, ImageNet ile ön eğitilmiş ESA modellerinin BCN20000 veri seti ile eğitilerek bir ara öğrenme aşamasından geçirildikten sonra transfer edilerek kullanılması durumunda, uygun veri artırma tekniği uygulanarak HAM10000 veri setinin eğitilmesinde daha yüksek *val_acc* ve daha düşük *val_loss* değerleri elde edilmesi beklenmektedir. BCN20000 veri seti ile modellerin ara eğitiminde ve öğrenmesinde gerçekleştirilebilecek iyileşme ile, HAM10000 veri setinin sınıflandırılma performansında önemli bir iyileşme sağlanabilir. Bunlara ek olarak, HAM10000 veri setinde sınıflandırma sonuçları diğerlerine göre daha düşük çıkan hastalık sınıflarındaki görüntüler üzerinde gürültü azaltma, kontrast iyileştirme, filtreleme gibi ilave görüntü işleme teknikleri uygulamak bu sınıfların sonuçlarında iyileşme sağlayabilir.

Bu çalışmada önerilen yöntem, literatürde benzerine tam olarak rastlanılmamış olması itibariyle de bir ilk olma özelliği taşımaktadır. [3] ve [32]'de belirtilen çalışmalar ile kısmen benzerlikler göstermektedir. [3]'teki çalışmada, cilt hastalıkları içeren görüntüler yerine kedi-köpek görüntüleri içeren bir veri seti ile eğitilmiş klasik bir ESA modeli transfer edilmiş ve daha sonra cilt hastalıkları içeren küçük bir veri setini ikili olarak sınıflandırmak için kullanılmış ve düşük sonuçlar elde edilmiştir. [32]'deki çalışmada ise, yine klasik bir ESA modeli geniş bir medikal veri seti (kullanılan veri setleri ile ilgili bilgi verilmemiş) ile ön eğitilmiş ve HAM10000 veri setinin iyi huylu-kötü huylu lezyon olarak ikili sınıflandırılması için kullanılmıştır.

Gelecek çalışmalarda, araştırmacılar için kullanım kolaylığı sağlaması ve yüksek doğrulukta sınıflandırma sonuçları elde edebilmek için, cilt hastalıkları içeren geniş medikal bir veri seti ile ara öğrenme uygulanmış ESA modellerinin makine

öğrenmesi kütüphanelerinde kullanıma hazır versiyonlarının sunulmasına yönelik çalışmalar yapılabilir.

Bu tez çalışmasında önerilen yöntem ile, HAM10000 veri setinin yedi hastalık türüne sınıflandırılmasında literatürdeki benzer birçok çalışmaya göre daha iyi sonuçlar elde edilmiştir. Farklı veri artırma ve bazı görüntü işleme teknikleri uygulanarak gelecekte daha iyi sınıflandırma sonuçları elde edilebilir.

KAYNAKÇA

- [1] **Url-1** P. Grieve, “Deep learning vs. machine learning: What’s the difference?”, 23.01.2020. <<https://www.zendesk.com/blog/machine-learning-and-deep-learning/>>, erişim tarihi 20.04.2021.
- [2] **Url-2** A. Biswal, “Top 10 Deep Learning Algorithms You Should Know in 2022”, <<https://www.simplilearn.com/tutorials/deep-learning-tutorial/deep-learning-algorithm>>, erişim tarihi 10.05.2021.
- [3] **S. Mukherjee ve D. Ganguly**, “Transfer Learning in Skin Lesion Classification”, *Adv. Intell. Syst. Comput.*, c. 1255, sayı May, ss. 343–349, 2020, doi: 10.1007/978-981-15-7834-2_32.
- [4] **Url-3** WHO, “Radiation: Ultraviolet (UV) radiation and skin cancer”. <[https://www.who.int/news-room/questions-and-answers/item/radiation-ultraviolet-\(uv\)-radiation-and-skin-cancer](https://www.who.int/news-room/questions-and-answers/item/radiation-ultraviolet-(uv)-radiation-and-skin-cancer)>, erişim tarihi 15.03.2021.
- [5] **Url-4** K. Ahearn, “ISIC Challenge”, *International Skin Imaging Collaboration*. <<https://challenge.isic-archive.com/>>, erişim tarihi 10.06.2021.
- [6] **M. S. Elmahdy, S. S. Abdeldayem, ve I. A. Yassine**, “Low quality dermal image classification using transfer learning”, *2017 IEEE EMBS Int. Conf. Biomed. Heal. Informatics, BHI 2017*, ss. 373–376, 2017, doi: 10.1109/BHI.2017.7897283.
- [7] **A. Krizhevsky, I. Sutskever, ve G. E. Hinton**, “ImageNet classification with deep convolutional neural networks”, *Commun. ACM*, c. 60, sayı 6, ss. 84–90, 2017, doi: 10.1145/3065386.
- [8] **Url-5** L. Fei, L., Deng, J., Russakovsky, O., Berg, A. & Kai, “ImageNet Image Database”, *Stanford Vision Lab, Stanford University, Princeton University*. <<https://image-net.org/>>, erişim tarihi 28.04.2021
- [9] **M. M. Hosny, K.M., Kassem, M.A., & Foad**, “Skin Cancer Classification using Deep Learning and Transfer Learning”, *2018 9th Cairo Int. Biomed. Eng. Conf.*, ss. 90–93, 2018, doi: 10.1109/CIBEC.2018.8641762.
- [10] **Url-6** ADDI Project, “PH2 Dataset”, 2012. <<https://www.fc.up.pt/addi/team.html>>, erişim tarihi 25.06.2021.

- [11] **U. O. Dorj, K. K. Lee, J. Y. Choi, ve M. Lee**, “The skin cancer classification using deep convolutional neural network”, *Multimed. Tools Appl.*, c. 77, sayı 8, ss. 9909–9924, 2018, doi: 10.1007/s11042-018-5714-1.
- [12] **D. de A. Rodrigues, R. F. Ivo, S. C. Satapathy vd.**, “A new approach for classification skin lesion based on transfer learning, deep learning, and IoT system”, *Pattern Recognit. Lett.*, c. 136, ss. 8–15, 2020, doi: 10.1016/j.patrec.2020.05.019.
- [13] **E. Yilmaz ve M. Trocan**, *Benign and Malignant Skin Lesion Classification Comparison for Three Deep-Learning Architectures*, c. 12033 LNAI. Springer International Publishing, 2020.
- [14] **Url-7 K. Ahearn**, “ISIC Archive”. <<https://www.isic-archive.com/>>, erişim tarihi 10.05.2021.
- [15] **T. Sasikala, S., Kumar, S.A. vd.**, “Towards Improving Skin Cancer Detection Using Transfer Learning”, *Biosci. Biotechnol. Res. Commun.*, c. 13, sayı 11, ss. 55–60, 2020, doi: 10.21786/bbrc/13.11/13.
- [16] **Url-8 A. Goldbloom**, “Kaggle”, *Google LLC*. <<https://www.kaggle.com/>>, erişim tarihi 13.07.2021.
- [17] **M. A. Kassem, K. M. Hosny, ve M. M. Fouad**, “Skin Lesions Classification into Eight Classes for ISIC 2019 Using Deep Convolutional Neural Network and Transfer Learning”, *IEEE Access*, c. 8, ss. 114822–114832, 2020, doi: 10.1109/ACCESS.2020.3003890.
- [18] **J. Steppan ve S. Hanke**, “Analysis of skin lesion images with deep learning”, ss. 1–8, 2021, Available at: <http://arxiv.org/abs/2101.03814>.
- [19] **T. Mendonca, P. M. Ferreira vd.**, “PH2 - A dermoscopic image database for research and benchmarking”, *Proc. Annu. Int. Conf. IEEE Eng. Med. Biol. Soc. EMBS*, sayı July, ss. 5437–5440, 2013, doi: 10.1109/EMBC.2013.6610779.
- [20] **S. M. M. De Faria vd.**, “Light Field Image Dataset of Skin Lesions”, *Proc. Annu. Int. Conf. IEEE Eng. Med. Biol. Soc. EMBS*, ss. 3905–3908, 2019, doi: 10.1109/EMBC.2019.8856578.
- [21] **X. Sun, J. Yang, M. Sun, ve K. Wang**, “A benchmark for automatic visual classification of clinical skin disease images”, *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, c. 9910 LNCS, ss. 206–222, 2016, doi: 10.1007/978-3-319-46466-4_13.

- [22] **I. Giotis, N. Molders, S. Land, M. Biehl vd.**, “MED-NODE: A computer-assisted melanoma diagnosis system using non-dermoscopic images”, *Expert Syst. Appl.*, c. 42, sayı 19, ss. 6578–6585, 2015, doi: 10.1016/j.eswa.2015.04.034.
- [23] **J. Kawahara, S. Daneshvar vd.**, “Seven-Point Checklist and Skin Lesion Classification Using Multitask Multimodal Neural Nets”, *IEEE J. Biomed. Heal. Informatics*, c. 23, sayı 2, ss. 538–546, 2019, doi: 10.1109/JBHI.2018.2824327.
- [24] **Uri-9 R. C. & K. H. Tschandl P.**, “The HAM10000 dataset, a large collection of multi-source dermatoscopic images of common pigmented skin lesions”, *Harvard Dataverse*, 2018. <<https://dataverse.harvard.edu/dataset.xhtml?persistentId=doi:10.7910/DVN/DBW86T>>, erişim tarihi 11.05.2021.
- [25] **H. Younis, M. H. Bhatti, ve M. Azeem**, “Classification of skin cancer dermoscopy images using transfer learning”, *15th Int. Conf. Emerg. Technol. ICET 2019*, 2019, doi: 10.1109/ICET48972.2019.8994508.
- [26] **Z. Rahman ve A. M. Ami**, “A transfer learning based approach for skin lesion classification from imbalanced data”, *Proc. 2020 11th Int. Conf. Electr. Comput. Eng. ICECE 2020*, ss. 65–68, 2020, doi: 10.1109/ICECE51571.2020.9393155.
- [27] **K. M. Hosny, M. A. Kassem, ve M. M. Fouad**, “Classification of Skin Lesions into Seven Classes Using Transfer Learning with AlexNet”, *J. Digit. Imaging*, c. 33, sayı 5, ss. 1325–1334, 2020, doi: 10.1007/s10278-020-00371-9.
- [28] **O. Sevli**, “A deep convolutional neural network-based pigmented skin lesion classification application and experts evaluation”, *Neural Comput. Appl.*, c. 33, sayı 18, ss. 12039–12050, 2021, doi: 10.1007/s00521-021-05929-4.
- [29] **C. Calderón, K. Sanchez, S. Castillo, ve H. Arguello**, “BILSK: A bilinear convolutional neural network approach for skin lesion classification”, *Comput. Methods Programs Biomed. Updat.*, c. 1, s. 100036, 2021, doi: 10.1016/j.cmpbup.2021.100036.
- [30] **J. P. Villa-Pulgarin vd.**, “Optimized convolutional neural network models for skin lesion classification”, *Comput. Mater. Contin.*, c. 70, sayı 2, ss. 2131–2148, 2022, doi: 10.32604/cmc.2022.019529.
- [31] **P. Yao vd.**, “Single Model Deep Learning on Imbalanced Small Datasets for Skin Lesion Classification”, c. 10000, ss. 1–10, 2021, doi: 10.1109/TMI.2021.3136682.

- [32] **M. S. Ali, M. S. Miah, J. Haque, M. M. Rahman, ve M. K. Islam**, “An enhanced technique of skin cancer classification using deep convolutional neural network with transfer learning models”, *Mach. Learn. with Appl.*, c. 5, sayı April, s. 100036, 2021, doi: 10.1016/j.mlwa.2021.100036.
- [33] **M. K. Islam vd.**, “Melanoma Skin Lesions Classification using Deep Convolutional Neural Network with Transfer Learning”, *2021 1st Int. Conf. Artif. Intell. Data Anal. CAIDA 2021*, ss. 48–53, 2021, doi: 10.1109/CAIDA51941.2021.9425117.
- [34] **A. Patil ve M. Rane**, “Convolutional Neural Networks: An Overview and Its Applications in Pattern Recognition”, *Smart Innov. Syst. Technol.*, c. 195, ss. 21–30, 2021, doi: 10.1007/978-981-15-7078-0_3.
- [35] **Url-10 S. Visherut**, “What do you mean by Convolutional Neural Network?”, 2021. <<https://discuss.boardinfinity.com/t/what-do-you-mean-by-convolutional-neural-network/8533>>, erişim tarihi 18.04.2021.
- [36] **Z. Kuş**, “Mikrokanonikal optimizasyon algoritması ile konvolüsyonel sinir ağlarında hiper parametrelerin optimize edilmesi”, Fatih Sultan Mehmet Vakıf Üniversitesi, 2019.
- [37] **J. Wei, Di., Anurag, B., Wei**, *Deep Learning Essentials: Your hands-on guide to the fundamentals of deep learning and neural network modeling*. Packt Publishing (January 30, 2018), 2018.
- [38] **M. Lin, Q. Chen, ve S. Yan**, “Network in network”, *2nd Int. Conf. Learn. Represent. ICLR 2014 - Conf. Track Proc.*, ss. 1–10, 2014.
- [39] **R. Yamashita, M. Nishio, R. K. G. Do, ve K. Togashi**, “Convolutional neural networks: an overview and application in radiology”, *Insights Imaging*, c. 9, sayı 4, ss. 611–629, 2018, doi: 10.1007/s13244-018-0639-9.
- [40] **Url-11 keras.io**, “Xception”. <<https://keras.io/api/applications/xception/>>, erişim tarihi 05.07.2021.
- [41] **N. Srivastava, G. Hinton vd.**, “Dropout: A simple way to prevent neural networks from overfitting”, *J. Mach. Learn. Res.*, c. 15, ss. 1929–1958, 2014.
- [42] **Url-12 I. Rajagopal**, “Batch Normalization - Speed up Neural Network Training”, 2018. <https://medium.com/@ilango100/batch-normalization-speed-up-neural-network-training-245e39a62f85>, erişim tarihi 03.08.2021.
- [43] **S. Ioffe ve C. Szegedy**, “Batch normalization: Accelerating deep network training by reducing internal covariate shift”, *32nd Int. Conf. Mach. Learn. ICML 2015*, c. 1, ss. 448–456, 2015.

- [44] **Url-13** C. J. C. LeCun, Y., Cortes, C., Burges, “The MNIST Database”, *Courant Institute, NYU Corinna Cortes, Google Labs, New York Christopher J.C. Burges, Microsoft Research, Redmond.* <<http://yann.lecun.com/exdb/mnist/>>, erişim tarihi 20.05.2021.
- [45] **Url-14** P. Baheti, “Neural Network Activation Functions”, 2021. <<https://www.v7labs.com/blog/neural-networks-activation-functions>>, erişim tarihi 05.07.2021
- [46] **C. Goodfellow, I., Yoshua, B. & Aaron**, “Numerical Computation”, 2016. <https://www.deeplearningbook.org/contents/numerical.html>.
- [47] **C. Nwankpa, W. Ijomah, A. Gachagan, ve S. Marshall**, “Activation Functions: Comparison of trends in Practice and Research for Deep Learning”, ss. 1–20, 2018, Available at: <http://arxiv.org/abs/1811.03378>.
- [48] **Y. Glorot, X., Bordes, A., Bengio**, “Deep Sparse Rectifier Neural Networks”, *Appear. Proc. 14th Int. Conf. Artif. Intell. Stat.*, c. 15, 2011, doi: 10.1002/ecs2.1832.
- [49] **H. Zheng, Z. Yang, W. Liu, J. Liang, ve Y. Li**, “Improving deep neural networks using softplus units”, *Proc. Int. Jt. Conf. Neural Networks*, c. 2015-Septe, sayı July 2015, ss. 1–5, 2015, doi: 10.1109/IJCNN.2015.7280459.
- [50] **A. F. Agarap**, “Deep Learning using Rectified Linear Units (ReLU)”, sayı 1, ss. 2–8, 2018, Available at: <http://arxiv.org/abs/1803.08375>.
- [51] **V. Badrinarayanan, A. Kendall, ve R. Cipolla**, “SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation”, *IEEE Trans. Pattern Anal. Mach. Intell.*, c. 39, sayı 12, ss. 2481–2495, 2017, doi: 10.1109/TPAMI.2016.2644615.
- [52] **K. He, X. Zhang, S. Ren, ve J. Sun**, “Deep residual learning for image recognition”, *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, c. 2016-Decem, ss. 770–778, 2016, doi: 10.1109/CVPR.2016.90.
- [53] **K. Simonyan ve A. Zisserman**, “Very deep convolutional networks for large-scale image recognition”, *3rd Int. Conf. Learn. Represent. ICLR 2015 - Conf. Track Proc.*, ss. 1–14, 2015.
- [54] **C. Szegedy vd.**, “Going deeper with convolutions”, *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, c. 07-12-June, ss. 1–9, 2015, doi: 10.1109/CVPR.2015.7298594.

- [55] **G. Huang, Z. Liu vd.**, “Densely connected convolutional networks”, *Proc. - 30th IEEE Conf. Comput. Vis. Pattern Recognition, CVPR 2017*, c. 2017-Janua, ss. 2261–2269, 2017, doi: 10.1109/CVPR.2017.243.
- [56] **Url-15** D. Radečić, “Softmax Activation Function”, 2020, <<https://towardsdatascience.com/softmax-activation-function-explained-a7e1bc3ad60>>, erişim tarihi 13.09.2021.
- [57] **E. Apaydın**, *Introduction to Machine Learning*. MIT Press, 2014.
- [58] **E. Seyyarer, F. Ayata, T. Uçkan, ve A. Karci**, “Derin Öğrenmede Kullanılan Optimizasyon Algoritmalarının Uygulanması ve Kıyaslanması”, *Anatol. J. Comput. Sci.*, sayı 2, ss. 90–98, 2020.
- [59] **Url-16** G. Carter, “Best Optimization Gradient”. <<https://tr.pinterest.com/pin/424816177350692379/>>, erişim tarihi 07.12.2021
- [60] **D. P. Kingma ve J. L. Ba**, “Adam: A method for stochastic optimization”, *3rd Int. Conf. Learn. Represent. ICLR 2015 - Conf. Track Proc.*, ss. 1–15, 2015.
- [61] **Url-17** J. Brownlee, “Gentle Introduction to the Adam Optimization Algorithm for Deep Learning”, 2017. <<https://machinelearningmastery.com/adam-optimization-algorithm-for-deep-learning/>>, erişim tarihi 16.08.2021.
- [62] **Url-18** keras.io, “Adam optimizer”. <https://keras.io/api/optimizers/adam/>, erişim tarihi 15.07.2021.
- [63] **Url-19** P. Solai, “Convolutions and Backpropagations”, 2018. <<https://pavisj.medium.com/convolutions-and-backpropagations-46026a8f5d2c>>, erişim tarihi 12.09.2021.
- [64] **Url-20** Keras.io, “About Keras”, *Keras.io*. <<https://keras.io/about/>>, erişim tarihi 15.07.2021.
- [65] **Url-21** tensorflow.org, “Neden TensorFlow”, *tensorflow.org*. <<https://www.tensorflow.org/about>>, erişim tarihi 20.07.2021.
- [66] **Url-22** Google, “Google Colaboratory”, *Google LLC*. <<https://colab.research.google.com/>>, erişim tarihi 25.07.2021.
- [67] **J. Combalia, M., Codella, N., Rotemberg vd.**, “BCN20000: Dermoscopic Lesions in the Wild”, 2019. <https://arxiv.org/abs/1908.02288>.
- [68] **N. C. F. Codella vd.**, “Skin lesion analysis toward melanoma detection: A challenge at the 2017 International symposium on biomedical imaging (ISBI), hosted by the international skin imaging collaboration (ISIC)”, *Int. Symp. Biomed. Imaging*, 2018-April, ss. 168–172, doi: 10.1109/ISBI.2018.8363547.

- [69] **Url-23** “SONIC”, 2015. <<https://isic-archive.com/girder#collection/55943cff9fc3c13155bcad5e>>, erişim tarihi 14.07.2021.
- [70] **Url-24** J. Ballerini *vd.*, “Dermofit Image Library”, *The University of Edinburgh*, 2013. <<https://licensing.edinburgh-innovations.ed.ac.uk/i/software/dermofit-image-library.html>>, erişim tarihi 15.07.2021.
- [71] **A. G. C. Pacheco *vd.***, “PAD-UFES-20: A skin lesion dataset composed of patient data and clinical images collected from smartphones”, *Data Br.*, c. 32, s. 106221, 2020, doi: 10.1016/j.dib.2020.106221.
- [72] **F. Chollet**, “Xception: Deep learning with depthwise separable convolutions”, *Proc. - 30th IEEE Conf. Comput. Vis. Pattern Recognition, CVPR 2017*, c. 2017-Janua, ss. 1800–1807, 2017, doi: 10.1109/CVPR.2017.195.
- [73] **Url-25** C. F. Wang, “A Basic Introduction to Separable Convolutions”, *Towards Data Science*, 2018. <<https://towardsdatascience.com/a-basic-introduction-to-separable-convolutions-b99ec3102728>>, erişim tarihi 16.09.2021.
- [74] **E. Westphal ve H. Seitz**, “A machine learning method for defect detection and visualization in selective laser sintering based on convolutional neural networks”, *Addit. Manuf.*, c. 41, sayı March, s. 101965, 2021, doi: 10.1016/j.addma.2021.101965.
- [75] **Url-26** Keras.io, “Keras Applications”, *keras.io*. <<https://keras.io/api/applications/>>, erişim tarihi 18.07.2021>.
- [76] **O. Attallah**, “MB-ai-his: Histopathological diagnosis of pediatric medulloblastoma and its subtypes via ai”, *Diagnostics*, c. 11, sayı 2, 2021, doi: 10.3390/diagnostics11020359.
- [77] **T. Emara, H. M. Afify, F. H. Ismail, ve A. E. Hassanien**, “A modified inception-v4 for imbalanced skin cancer classification dataset”, *Proc. - ICCES 2019 2019 14th Int. Conf. Comput. Eng. Syst.*, ss. 29–33, 2019, doi: 10.1109/ICCES48960.2019.9068110.
- [78] **N. Gessert *vd.***, “Skin Lesion Diagnosis using Ensembles, Unscaled Multi-Crop Evaluation and Loss Weighting”, ss. 1–5, 2018, Available at: <http://arxiv.org/abs/1808.01694>.
- [79] **S. S. Chaturvedi *vd.***, “Skin lesion analyser: an efficient seven-way multi-class skin cancer classification using mobilenet”, *Adv. Intell. Syst. Comput.*, c. 1141, ss. 165–176, 2021, doi: 10.1007/978-981-15-3383-9_15.

- [80] **A. Bhardwaj ve P. P. Rege**, “Skin Lesion Classification Using Deep Learning”, *Lect. Notes Electr. Eng.*, c. 703, ss. 575–589, 2020, doi: 10.1007/978-981-15-8391-9_42.
- [81] **A. Pal, S. Ray, ve U. Garain**, “Skin disease identification from dermoscopy images using deep convolutional neural network”, ss. 9–10, 2018, Available at: <http://arxiv.org/abs/1807.09163>.
- [82] **R. Garg, S. Maheshwari, ve A. Shukla**, “Decision Support System for Detection and Classification of Skin Cancer Using CNN”, *Adv. Intell. Syst. Comput.*, c. 1189, ss. 578–586, 2021, doi: 10.1007/978-981-15-6067-5_65.
- [83] **M. A. A. Milton**, “Automated Skin Lesion Classification Using Ensemble of Deep Neural Networks in ISIC 2018: Skin Lesion Analysis Towards Melanoma Detection Challenge”, 2019, Available at: <http://arxiv.org/abs/1901.10802>.