



T.C.

AKSARAY ÜNİVERSİTESİ

SOSYAL BİLİMLER ENSTİTÜSÜ

YÖNETİM BİLİŞİM SİSTEMLERİ ANABİLİM DALI

**GÖRÜNTÜ SINIFLANDIRMASI İÇİN EVRİŞİMSEL SİNİR AĞLARINDA
PARAMETRELERİN SINIFLANDIRMA PERFORMANSINA ETKİSİ**

YÜKSEK LİSANS TEZİ

İBRAHİM AKSOY

DANIŞMAN

DOÇ. DR. KEMAL ADEM

AKSARAY 2023

T.C.
AKSARAY ÜNİVERSİTESİ
SOSYAL BİLİMLER ENSTİTÜSÜ
YÖNETİM BİLİŞİM SİSTEMLERİ ANABİLİM DALI

GÖRÜNTÜ SINIFLANDIRMASI İÇİN EVRİŞİMSEL SİNİR AĞLARINDA
PARAMETRELERİN SINIFLANDIRMA PERFORMANSINA ETKİSİ

YÜKSEK LİSANS TEZİ

İBRAHİM AKSOY

DANIŞMAN
DOÇ. DR. KEMAL ADEM

AKSARAY 2023

TELİF HAKKI VE TEZ FOTOKOPİ İZİN FORMU

Bu tezin tüm hakları saklıdır. Kaynak göstermek koşuluyla tezin teslim tarihinden itibaren(...) ay sonra tezden fotokopi çekilebilir.

YAZARIN

Adı : İbrahim
Soyadı : AKSOY
Bölümü : Yönetim Bilişim Sistemleri
İmza :
Teslim tarihi :

TEZİN

Türkçe Adı : Görüntü Sınıflandırması İçin Evrişimsel Sinir Ağlarında Parametrelerin Sınıflandırma Performansına Etkisi

İngilizce Adı : Effect of Parameters on Classification Performance in Convolutional Neural Networks for Image Classification

ETİK İLKELERE UYGUNLUK BEYANI

Tez yazma sürecinde bilimsel ve etik ilkelere uyduğumu, yararlandığım tüm kaynakları kaynak gösterme ilkelerine uygun olarak kaynakçada belirttiğimi ve bu bölümler dışındaki tüm ifadelerin şahsıma ait olduğunu beyan ederim.

Yazar Adı Soyadı: İbrahim AKSOY

İmza:



T.C.
AKSARAY ÜNİVERSİTESİ
SOSYAL BİLİMLER ENSTİTÜSÜ
JÜRİ ONAY SAYFASI

İbrahim AKSOY tarafından hazırlanan “Görüntü sınıflandırması İçin Evrişimsel Sinir Ağlarında Parametrelerin Sınıflandırma Performansına Etkisi” başlıklı tez çalışması aşağıdaki jüri tarafından oy birliği ile Aksaray Üniversitesi Yönetim Bilişim Sistemleri Anabilim Dalı’nda Yüksek Lisans Tezi olarak kabul edilmiştir.

İmza

Danışman: Doç. Dr. Kemal ADEM

Bilgisayar Mühendisliği, Sivas Bilim ve Teknoloji Üniversitesi

Üye: Dr. Öğr. Üyesi Selçuk KILIÇ

Yönetim Bilişim Sistemleri, Aksaray Üniversitesi

Üye: Dr. Öğr. Üyesi Özlem POLAT

Mekatronik Mühendisliği, Sivas Cumhuriyet Üniversitesi

Tez Savunma Tarihi: 22 / 06 / 2023

Sosyal Bilimler Enstitüsü Yönetim Kurulu’nun tarih ve sayılı kararı ile onaylanmıştır.

Sosyal Bilimler Enstitüsü Müdürü
Doç. Dr. Sevilay USLU DİVANOĞLU
İmza

TEŞEKKÜR

Çalışmamın başından itibaren her süreçte bana yol gösteren ve desteğini esirgemeyen danışmanım, değerli hocam Sayın **Doç. Dr. Kemal ADEM**'e, tezimin her aşamasında fedakârlıklarda bulunarak bana çalışma imkânı yaratan mesai arkadaşlarım **Dr. Halit ARSLAN** ve **Emre BÜYÜKTOKATLI**'ya, varlığıyla her daim huzur ve motivasyon kaynağım olan canım oğlum **Kutay**'a ve yüksek lisans eğitimimin başından sonuna kadar bana her zaman destek olan sevgili eşim **Şeyma AKSOY**'a sonsuz teşekkürlerimi sunarım.

AKSARAY ÜNİVERSİTESİ
SOSYAL BİLİMLER ENSTİTÜSÜ
GÖRÜNTÜ SINIFLANDIRMASI İÇİN EVRİŞİMSSEL SİNİR
AĞLARINDA PARAMETRELERİN SINIFLANDIRMA
PERFORMANSINA ETKİSİ
(Yüksek Lisans Tezi)

İbrahim AKSOY, Aksaray 2023

ÖZET

Günümüzde, görüntü işleme ve örüntü tanıma alanında büyük ilerlemeler kaydedilmekte ve bu alandaki başarıların arkasında evrişimsel sinir ağları büyük bir rol oynamaktadır. Biyolojik sinir ağlarının işleyişinden ilham alarak tasarlanan evrişimsel sinir ağları, görüntü verileri üzerinde nesne tanıma, sınıflandırma ve öznitelik çıkarma gibi görevleri başarıyla gerçekleştirmektedir. Ancak evrişimsel sinir ağlarının tasarlanması ve uygulanmasında karşılaşılan bazı zorluklar mevcuttur. Kullanılan evrişimsel sinir ağı modeline, veri setine ve donanımlara göre hiperparametrelerin optimizasyonu konusu bu zorlukların başında gelmektedir. Bu çalışmada iki farklı veri seti kullanılarak evrişimsel sinir ağı modellerinde bazı hiperparametrelerin sınıflandırma performansı üzerindeki etkileri incelenmiştir. Çalışmada incelenen hiperparametreler; eğitim tur sayısı, nöron sayısı, paket boyutu, aktivasyon fonksiyonları, optimizasyon algoritmaları ve öğrenme oranıdır. Keras kütüphanesinde yer alan evrişimsel sinir ağı modellerinde veri setleri üzerinde yapılan testler sonucunda en iyi performansı gösteren NASNetMobile ve DenseNet201 modelleri ile uygulamalar yapılmıştır. Hiperparametrelerin farklı aralıklardaki değerleri ile planlanan 65 farklı model için eğitim uygulanmış ve sonuçlar elde edilmiştir. Yapılan çalışmalar neticesinde doğruluk oranlarında NASNetMobile modelinde başlangıç değerine göre % 6,5, DenseNet201 modelinde ise başlangıç değerine göre % 11,55 yükselme gözlemlenmiştir.

Bilim Kodu: 92402, 92418, 92431, 92432

Anahtar Kelimeler: Görüntü Sınıflandırma, Hiperparametreler, Aktivasyon fonksiyonları, Optimizasyon Algoritmaları, Öğrenme Oranı, ESA

Sayfa Adedi: 56

Danışman: Doç. Dr. Kemal ADEM

AKSARAY UNIVERSITY
GRADUATE SCHOOL OF SOCIAL SCIENCES
EFFECT OF PARAMETERS ON CLASSIFICATION PERFORMANCE
IN CONVOLUTIONAL NEURAL NETWORKS FOR IMAGE
CLASSIFICATION

(M.A. THESIS)

İbrahim AKSOY, Aksaray 2023

ABSTRACT

Nowadays, great progress has been made in the field of image processing and pattern recognition, and convolutional neural networks play a major role behind the success in this field. Convolutional neural networks designed with inspiration from effectiveness of biological neural networks successfully implement processors such as object recognition, thinking, and feature extraction on image data. However, there are some difficulties encountered in the design and implementation of convolutional neural networks. Optimization of hyperparameters according to the convolutional neural network model, data set and hardware used is one of these difficulties. In this study, the effects of some hyper parameters on classification performance in convolutional neural networks models were investigated using two different datasets. Hyperparameters examined in the study; the number of epochs, the number of neurons, batch size, activation functions, optimization algorithms and learning rate. In the convolutional neural network models thosetake place in the Keras library, applications were made with the NASNetMobile and DenseNet201 models, which showed the best performance as a result of the tests made in the dataset. 65 different training tasks planned with the values of the hyperparameters in different intervals were applied and the results were obtained. As a result of the studies, it was observed that the accuracy rates increased by 6.5% compared to the initial value in the NASNetMobile model and by 11.55% compared to the initial value in the DenseNet201 model.

Science Code: 92402, 92418, 92431, 92432

Key Words: Image Classification , Hyperparameters, Activation Functions, Optimization Algorithms, Learning Rate, CNN

Page Number: 56

Supervisor: Assoc. Prof. Dr. Kemal ADEM

İÇİNDEKİLER

TELİF HAKKI VE TEZ FOTOKOPİ İZİN FORMU	i
ETİK İLKELERE UYGUNLUK BEYANI.....	ii
JÜRİ ONAY SAYFASI	iii
TEŞEKKÜR.....	iv
ÖZET.....	v
ABSTRACT	vi
TABLolar LİSTESİ	ix
ŞEKİLLER LİSTESİ.....	x
SİMGELER VE KISALTMALAR LİSTESİ	xii
1. GİRİŞ	1
1.1. ARAŞTIRMANIN AMACI.....	3
1.2. ARAŞTIRMANIN ÖNEMİ	3
2. KURAMSAL TEMELLER.....	5
3. MATERYAL VE YÖNTEM	9
3.1. VERİ SETLERİ	9
3.2. YAPAY SİNİR AĞLARI	10
3.3. DERİN ÖĞRENME	11
3.4. DERİN ÖĞRENME KÜTÜPHANELERİ	12
3.5. EVRİŞİMSEL SİNİR AĞLARI.....	13
3.5.1. Girdi Katmanı	13
3.5.2. Evrişim katmanı.....	14
3.5.3. Havuzlama Katmanı	15
3.5.4. Tam Bağlı Katman.....	15
3.6. EVRİŞİMSEL SİNİR AĞI MODELLERİ	15

3.6.1. DenseNet	16
3.6.2. NASNetMobile	16
3.7. EVRİŞİMSSEL SİNİR AĞLARINDA HİPERPARAMETRELER	17
3.7.1. Eğitim Tur (Epoch) Sayısı	18
3.7.2. Nöron Sayısı	18
3.7.3. Paket Boyutu (Batch Size).....	18
3.7.4. Aktivasyon Fonksiyonları.....	19
3.7.5. Optimizasyon Yöntemleri.....	20
3.7.6 Öğrenme Oranı (Learning Rate).....	20
4. BULGULAR	21
4.1. EVRİŞİMSSEL SİNİR AĞI MODELLERİ	21
4.2. HİPERPARAMETRELERİN UYGULANMASI.....	22
4.2.1. Eğitim Tur (Epoch) Sayısı	22
4.2.2. Nöron Sayısı	24
4.2.3. Paket Boyutu.....	25
4.2.4. Aktivasyon Fonksiyonları.....	26
4.2.5. Optimizasyon Algoritmaları	29
4.2.6 Öğrenme Oranları	31
5. SONUÇLAR	35
6. KAYNAKÇA	37
AKSARAY ÜNİVERSİTESİ TEZ DEĞERLENDİRME FORMU	43

TABLÖLER LİSTESİ

Tablo 1: Optimize edilebilecek hiperparametreler	17
Tablo 2: Aktivasyon Fonksiyonları	19
Tablo 3: ESA Modellerinin performansları.....	22
Tablo 4: Epoch sayısına göre eğitim sonuçları	23
Tablo 5: Nöron sayısına göre eğitim sonuçları	24
Tablo 6: Birinci veri seti ile yapılan paket boyutu eğitim sonuçları	25
Tablo 7: İkinci veri seti ile yapılan paket boyutu eğitim sonuçları	25
Tablo 8: Aktivasyon fonksiyonlarının sonuç performansları (İkinci veri seti)	27
Tablo 9: Aktivasyon fonksiyonlarının sonuç performansları (Birinci veri seti)	28
Tablo 10: Optimizasyon fonksiyonlarının sonuç performansları (İkinci veri seti)	30
Tablo 11: Optimizasyon fonksiyonlarının sonuç performansları (Birinci veri seti)	30
Tablo 12: RMSprop ve Adamax algoritmalarının öğrenme oranlarına göre sonuçları.....	32
Tablo 13: Adam ve Adagrad algoritmalarının öğrenme oranlarına göre sonuçları	32
Tablo 14: RMSprop ve Adam algoritmalarının öğrenme oranlarına göre sonuçları	32

ŞEKİLLER LİSTESİ

Şekil 1: Makine Öğrenmesi ve Derin Öğrenme Yöntemlerinin Karşılaştırılması.....	2
Şekil 2: Birinci veri setinden örnek görüntüler.....	9
Şekil 3: İkinci veri setinden örnek görüntüler	10
Şekil 4: Yapay Sinir Ağı Yapısı	11
Şekil 5: İlk ileri beslemeli ve çok katmanlı derin öğrenme modeli	12
Şekil 6: Evrişimsel Sinir Ağı Modeli.....	13
Şekil 7: Evrişim İşlemi	14
Şekil 8: Havuzlama işlemleri.....	15
Şekil 9: DenseNet Katman Yapısı	16
Şekil 10: NasNetMobile Mimarisi	17
Şekil 11: 4 no'lu işe ait epoch ve doğruluk oranı grafiği	23
Şekil 12: 8 no'lu işe ait epoch ve doğruluk oranı grafiği	23
Şekil 13: NASNetMobile modelinde nöron sayılarına göre doğruluk grafiği.....	24
Şekil 14: DenseNet201 modelinde nöron sayılarına göre doğruluk grafiği	24
Şekil 15: 23 no'lu iş eğitim ve test doğruluk grafiği	26
Şekil 16: 26 no'lu iş eğitim ve test doğruluk grafiği	26
Şekil 17: 19 no'lu iş eğitim ve test doğruluk grafiği	26
Şekil 18: 22 no'lu iş eğitim ve test doğruluk grafiği	26
Şekil 19: DenseNet201 modelinde aktivasyon fonksiyonlarının doğruluk grafiği.....	27
Şekil 20: 27 no'lu işe ait doğruluk ve kayıp değerleri grafiği	28
Şekil 21: 28 no'lu işe ait doğruluk ve kayıp değerleri grafiği	28
Şekil 22: NASNetMobile modelinde aktivasyon fonksiyonlarının doğruluk grafiği	29
Şekil 23: DenseNet201 modelinde optimizasyon algoritmalarının doğruluk grafiği	30
Şekil 24: NASNetMobile modelinde optimizasyon algoritmalarının doğruluk grafiği.....	31

Şekil 25: RMSprop 0,01 öğrenme oranı doğruluk grafiği	33
Şekil 26: RMSprop 0,0001 öğrenme oranı doğruluk grafiği	33



SİMGELER VE KISALTMALAR LİSTESİ

API	Application Programming Interface (Uygulama Programlama Ara Yüzü)
BSD	Berkeley Software Distribution Lisansı
CIFAR	Canadian Institute For Advanced Research
CPU	Merkezi İşlem Birimi
CUDA	Compute Unified Device Architecture (Bilgisayar Birleşik Cihaz Mimarisi)
ELU	Üstel Doğrusal Aktivasyon Fonksiyonu
EMNIST	Extended Modified National Institute of Standards and Technology
ESA	Evrişimsel Sinir Ağları
GPU	Grafik İşleme Birimi
ILSVRC	ImageNet Large Scale Visual Recognition Challenge
M - BESA	Maskeli Bölgesel Evrişimsel Sinir Ağları
MNIST	Modified National Institute of Standards and Technology
NAS	Neural Architecture Search (Nöral Mimari Arama)
RAM	Random Access Memory (Rastgele Erişimli Hafıza)
ReLU	Doğrultulmuş Doğrusal Aktivasyon Fonksiyonu
RMSprop	Root Mean Square Propagation (Kare Ortalamalarının Karekökü Yayılımı)
SGD	Stochastic Gradient Descent (Olasalıksal Eğitim Düşümü)
STL-10	Stanford Large-Scale 10
Tanh	Hiperbolik Tanjant Fonksiyonu
VGG 16	Visual Geometry Group 16 Evrişimsel Sinir Ağı Modeli
VGG 19	Visual Geometry Group 19 Evrişimsel Sinir Ağı Modeli
YBS	Yönetim Bilişim Sistemleri
YSA	Yapay Sinir Ağları

1. GİRİŞ

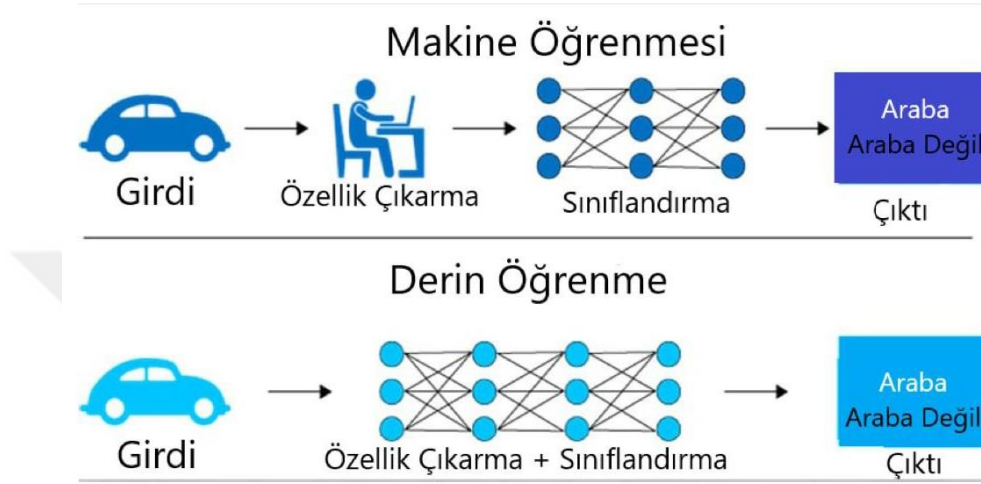
Bilgi çağı olarak nitelendirilebileceğimiz bu zamanda “Bilgi” kavramı hiç olmadığı kadar önemli ve ulaşılabilir bir seviyeye yükselmiştir. Makine öğrenmesi ve yapay zeka yöntemleri ile bilgi sadece insanlar tarafından değil, aynı zamanda bilgisayarlar tarafından da kullanılmaktadır. Yapay zeka kavramı günümüz dünyasında büyük bir ilgi ve heyecan uyandırmaktadır. Çünkü makinelerin insanoğlunun hayatına girmesinden bu yana, insan gibi düşünen, davranan ve kararlar veren robotlar hep hayal edilmiştir. Bu hayallerin izleri yıllar önce çekilen filmlerde ve yazılan kitaplarda bile görülebilir. Artık yapay zeka hayal dünyasının ötesine geçerek hayatın bir parçası konumuna gelmiştir. Akıllı ev sistemleri, yüz tanıma, sanal asistanlar, pazarlama, araç tanıma sistemleri gibi uygulamalar yapay zeka temelli teknolojilerine örnek olarak verilebilir.

Yapay zeka kavramının temel unsurlarından biri olan öğrenme işleminde genellikle yapay sinir ağları kullanılmaktadır. İnsan beyninin çalışma prensiplerinden esinlenerek tasarlanan yapay sinir ağları (YSA), karmaşık problemleri çözebilen, örüntüleri tanıyabilen ve tahminlerde bulunabilen sistemlerin geliştirilmesini sağlamaktadır (Öztemel, 2003). YSA’nın en önemli özelliklerinden biri ise verilerden öğrenebilme kabiliyetidir. Etkili bir öğrenme sağlayabilmek için YSA çok miktarda veriye ihtiyaç duyar. Bu bağlamda YSA’nın tarihi çok daha eskilere dayansa da kullanılabilecek veri miktarının artmasıyla günümüzde daha etkin şekilde kullanılmaya başlamıştır. Cep telefonu, tablet gibi mobil cihazlara kamera donanımları eklenmesi, güvenlik kameralarının yaygınlaşması ve insansız hava araçları tarafından sağlanan görüntüler sayesinde görüntü veri kümeleri de oldukça genişlemiş durumdadır. Özellikle görüntü işleme yöntemleri ile geliştirilen yapay zeka uygulamalarının birçok sektörde aktif olarak kullanıldığı görülmektedir. Üretim ve savunma sanayi, güvenlik sistemleri, sağlık, elektronik, tasarım, mimari gibi birçok alanda kullanılan görüntü işleme uygulamalarında temel maksadın görüntü içindeki istenilen nesnelerin tespit edilmesi veya görüntünün sınıflandırılması olduğu söylenebilir (Santaş ve Gülesin, 2011). Nesne tespiti için zaman içerisinde farklı algoritmalar ve yöntemler geliştirilmiştir. Nesne tespiti, takibi ve sınıflandırma işlemlerinde en çok kullanılan yöntemlerden biri derin öğrenmedir.

Derin öğrenmenin temelini çok katmanlı yapay sinir ağları oluşturmaktadır. Klasik makine öğrenmesi yöntemlerinde öznitelik çıkarma ve seçme işlemleri araştırmacılar tarafından yapılmakta iken derin öğrenme yöntemlerinde ise katmanlar boyunca öznitelikler ve ağırlıkları model tarafından belirlenir. Araştırmacıları özellik çıkarım yükünden kurtarması ve birbiriyle

bağlantılı katmanlar sayesinde daha kullanışlı öznitelikler elde etmesi derin öğrenme yöntemlerine popülerlik kazandırmıştır (Aktürk ve Serbest, 2022).

Derin öğrenmenin ilgi çekici bir diğer özelliğinin verilerdeki gizli ilişkileri keşfetmesi olduğu söylenebilir. Büyük miktarlarda veriyi işleyerek, örnekler veya durumlar arasında beklenmedik bağlantılar kurması, araştırmalarda yenilikçi çözümler sunabilme potansiyelini ortaya koymaktadır.



Şekil 1: Makine Öğrenmesi ve Derin Öğrenme Yöntemlerinin Karşılaştırılması (Daşgın, 2022)

Derin öğrenme modelleri etkili çözümler üretebilmek için büyük miktarda veriye ve aynı zamanda hesaplama gücüne ihtiyaç duyarlar. Donanımların gelişmesiyle hesaplama gücü her geçen gün artarken, büyük verilerin elde edilme yöntemleri zaman zaman etik ve gizlilik tartışmalarını beraberinde getirmektedir. Derin öğrenme modellerinde problemin kendisi, donanım, veri seti, veri setinin çeşitliliği ve büyüklüğü gibi değişkenlere bağlı olarak farklı başarı performansları elde edilmesi olağandır. Bu sebeple derin öğrenmede problemlere göre farklı alanlarda özelleşmiş yöntemler bulunmaktadır.

Evrişimsel sinir ağları (ESA) görüntü işleme alanında en başarılı ve en çok tercih edilen derin yöntemidir (Alzubaidi ve Ark., 2021). ESA biyolojik görme sisteminin işleyişinden ilham alınarak geliştirilen çok katmanlı bir yapıya sahiptir. Hem insan öğrenmesinde hem de ESA’da model ne kadar fazla ve çeşitli örnek ile çalışırsa genelleme yapma ve ayırt etme özellikleri o derecede gelişir. Örneğin alanında deneyimli ve çok miktarda radyoloji görüntüsü ile çalışmış bir hekim, bu görüntülerle daha az çalışmış birine göre yeni bir radyoloji görüntüsünde hastalıklara dair izleri daha hızlı ve yüksek kesinlikle ayırt edebilir. ESA’da da bu durum benzerdir. Veri setindeki görüntülerin sayısı ve çeşitliliğinin ağırlık öğrenme kapasitesini etkilediği bilinmektedir.

Duman ve Akın (2019), çalışmalarında CIFAR, MNIST ve ImageNet gibi kapsamlı ve büyük görüntü veri setlerinin kullanıma sunulmasıyla ESA'nın yükseliş döneminin başladığını belirtmişlerdir. ImageNet Büyük Ölçekli Görsel Tanıma Yarışması'nı (ImageNet Large Scale Visual Recognition Challenge – ILSVRC) 2012 yılında ESA modeli Alexnet algoritması kazanmış ve ilerleyen yıllarda başarısı giderek artmıştır. ESA görüntü tanıma ve sınıflandırma problemlerinde elde ettiği yüksek başarı performansları sayesinde birçok alanda en çok tercih edilen derin öğrenme modeli olmuştur. Literatür incelendiğinde ESA'nın görüntü işlemenin yanında doğal dil işleme ve zaman serisi gibi diğer veri türlerinde de kullanıldığı görülmektedir (Onan, 2020).

ESA'nın göstermiş olduğu bu yükseliş ve başarının yanı sıra probleme uygun bir sinir ağı modelinin oluşturulması ve bu model için en iyi hiperparametrelerin seçimi oldukça karmaşık bir konudur. En uygun hiperparametreleri belirlemek için “İzgara Arama”, “Rastgele Arama”, “Genetik Algoritma”, “Bayesian Eniyileme” gibi yöntemler bulunmaktadır. Ancak bu yöntemleri model üzerinde uygulamak ve ideal parametreleri belirlemek, zaman alıcı ve maliyetli işlemlerdir. Bu nedenle uygulanan modellerde hiperparametre seçim işlemi daha çok deneme yanılma yöntemine dayanmaktadır (Smith, 2018). Model tasarlamada karşılaşılan bu zorluklar sonucunda, kullanılan parametreler zaman içerisinde kalıplaşmıştır. Ancak her problem ve veri birbirinden farklı olması nedeniyle bir modelin tüm problemler üzerinde aynı sonucu vermesi beklenmemelidir.

1.1. Araştırmanın Amacı

Sağladığı başarıların neticesinde ESA modelleri sürekli geliştirilmeye devam etmektedir. Açık kaynaklı Keras kütüphanesi bu alanda en geniş ve güncel uygulamaları bünyesinde barındıran kütüphanelerden biridir. Bu çalışmada Keras kütüphanesindeki ESA modelleri kullanılmıştır. Seçilen modelde, paket boyutu (batch size), eğitim tur (epochs) sayısı, optimizasyon algoritmaları, aktivasyon fonksiyonları, öğrenme oranı (learning rate) ve nöron sayısı hiperparametrelerinin model başarısına etkilerini gözlemleyebilmek için 117 farklı eğitim işi planlanmıştır.

Bu çalışmada, ESA modellerinde hiperparametrelerin farklı değerlerinin modelin sınıflandırma performansı üzerindeki etkisinin gözlemlenmesi amaçlanmıştır.

1.2. Araştırmanın Önemi

ESA nesne tespitinde sıklıkla kullanılan bir derin öğrenme yöntemidir. Temel yapıları aynı olsa da kendine özgü farklılıkları ve özellikleri olan birçok ESA modeli geliştirilmektedir. Bir sinir

ağını tasarlamak ve oluşturmak uzmanlık gerektiren ve kolay olmayan bir iştir. Bu sebeple çoğu kişi çalışmalarında transfer öğrenme modellerini kullanmayı tercih etmektedir. Derin öğrenme kütüphanelerinin katkısıyla transfer öğrenme modelleri kullanmak araştırmacılara oldukça kolaylık sağlar ve zaman kazandırmaktadır.

Transfer öğrenme modeller kullanıcılar pratiklik sağlasa da her durumda ve her veriyle aynı performansı göstermesi beklenemez. Kullanılan veri setlerindeki görüntüler boyutlarına, renklerine, perspektiflerine göre çeşitlilik gösterebilir. Modelin performansını problem ve veri özelinde en üst seviyeye çıkarmak için bazı parametrelerin uygun şekilde seçilmesi gerekmektedir. Bu işleme hiperparametre optimizasyonu denilmektedir (Feurer ve Hutter, 2019). Ancak parametreler ve alabilecekleri değerler göz önüne alındığında, çok sayıda olasılık arasından en uygun parametre değerlerini belirlemek karmaşık bir süreç olarak karşımıza çıkmaktadır.

ESA modellerinde probleme ve veri setine göre hiperparametrelerin nasıl etkiler ortaya çıkardığını belirleyerek, araştırmacılara parametrelerin seçimi konusunda ışık tutabilecek olması bakımından bu çalışmanın önemli olduğu düşünülmektedir.

2. KURAMSAL TEMELLER

Dijital dünyada her geçen gün artan kullanıcı sayısı, daha büyük veri kümeleri, daha kolay erişilebilir büyük depolama alanları ve her türlü veriyi daha hızlı işleyebilen daha güçlü donanımlara sahip makineler, derin öğrenmenin yükselişindeki önemli etkenler olmuştur. Hiyerarşik öğrenme veya derin makine öğrenmesi gibi farklı isimlerle anılan derin öğrenme, yeni bir makine öğrenmesi araştırma alanıdır ve makine öğrenmesinin temel amaçlarından biri yapay zekaya yakınlaşmaktır (Deng ve Yu, 2014).

1950'de İngiliz matematikçi Alan Turing tarafından bir makinenin herhangi bir insanla yazılı konuşma yapması üzerine bir çalışma yapıldı. Turing testi olarak adlandırılan bu testte beş dakikalık yazılı konuşma sonucunda, konuşmayı gerçekleştiren kişi karşıdakinin bir insan olduğuna ikna olursa makine testi geçecekti. Böylece makinenin düşünebildiği sonucuna varılır (wikipedia.org, 2023).

Makine öğrenmesi alanında ilk çalışmalardan birini gerçekleştiren de Arthur Samuel'dir. Samuel çalışmasında dama oynayabilecek bir program algoritması oluşturdu. Bu algoritma iyi hamlelerle kötü hamleleri ayırt edebilmesi ve geçmiş hareketleri kontrol ederek kendini geliştirmesi sebebiyle derin öğrenme alanında önemli bir yere sahiptir (Wiederhold ve McCarthy, 1992).

Derin öğrenme yöntemleri başlangıçta yalnızca bir fikir olsa da sonraki yıllarda veri miktarlarının artması ve bilgisayar donanımlarının gelişmesiyle bu durum değişmiştir. Günümüzde elle yapılan işlemlerin yerini derin öğrenme yaklaşımlarının almasıyla nesne tespiti ve nesne takibi çalışmaları büyük bir ivme kazanmıştır (Demirel, 2023). Nesne tespiti ya da nesne algılama; görüntüdeki bir veya daha fazla sınıfa ait nesnelerin bilgisayarlar tarafından tespit edilmesi işlemidir (Amit ve Ark., 2020). Nesne tespiti; yüz tanımadan ev robotlarına, hava araçlarından sürücüsüz arabalara kadar gerçek dünyada birçok uygulama geliştirme imkânı sunar. Derin öğrenme yöntemleri ile farklı alanlarda çok sayıda görüntü sınıflandırma ve nesne tespiti çalışması yapılmıştır (Adem, 2022; Adem ve Ark., 2023; Yurdusev ve Ark., 2023; Yılmaz ve Ark., 2023).

Toğaçar ve arkadaşları (2019), çalışmalarında ESA'yı kullanarak zatürre hastalığının erken teşhisi için tanıma sistemi geliştirmişlerdir. Erişime açık akciğer röntgen görüntüleri kullanarak AlexNet modeli ile geliştirdikleri tanıma sistemiyle % 95,8 sınıflandırma başarısı elde etmişlerdir. Zatürre gibi ölümcül bir hastalık için hızlı ve bu doğruluk oranında erken teşhisin oldukça önemli olduğu vurgusunu yapmışlardır.

Salur ve arkadaşları (2019), X-ray cihazlarındaki görüntülerde tehdit oluşturabilecek nesnelerin tespiti için bir çalışma yapmışlardır. Çalışmada Alexnet ESA modeli 12000 görüntü ile eğitilerek gerçek zamanlı çalışabilen bir sistem geliştirilmiştir. Geliştirilen sistemin tehdit unsuru içeren nesneleri % 96,9 doğruluk oranıyla ortalama 12 milisaniyede tespit ettiği belirtilmiştir.

Hekim ve arkadaşları (2020), 1200 elma görüntüsü üzerinde ESA kullanarak yaptıkları çalışmalarında sağlıklı ve çürük elmaları sınıflandırmışlardır. Çalışmada % 74 olan doğruluk oranı, ESA tarafından desteklenen Yapay Arı Kolonisi (Artificial Bee Colony - ABC) ve Parçacık Sürü Optimizasyonu (PSO) kullanılarak % 98,33'e kadar yükseltilmiştir.

Bayram (2020), kamera görüntülerinden otomatik olarak plaka tanıyabilme üzerine yaptığı çalışmasında maskeli bölgesel evrişimsel sinir ağı (M - BESA) kullanarak % 98,46 doğruluk ile plaka tanıma işlemi gerçekleştirmiştir. Ortalama plaka tanıma süresi ise 0,4 saniye olarak belirtilmiştir.

Karakurt ve İşeri (2022), patoloji görüntülerinden kanserli ve kansersiz hücreleri ayırt etmek için ESA kullanarak yaptıkları çalışmalarında, 0,8775 doğruluk ve 0,8238 hassasiyet değerleri ile sınıflandırma performansı elde etmişlerdir. Araştırmacılar tarafından tasarlanan 3 evrişim, 3 havuzlama ve 200 nörona sahip bir tam bağlı katmandan oluşan ESA modeli, 40000 görüntü ile eğitilerek bu sonuçlara ulaşılmıştır.

Ülkemizde ve dünyada farklı alanlarda nesne tespiti ile ilgili çok sayıda çalışma yapılmıştır ve her geçen gün daha yüksek başarılarla literatüre yeni çalışmalar eklenmeye devam etmektedir.

Nesne tespiti ve görüntü sınıflandırma çalışmalarında en çok tercih edilen derin öğrenme yöntemlerinden biri ESA'dır (Li ve Ark., 2021). ESA'nın tarihine baktığımızda aslında YSA'nın ortaya çıkışı ile bağdaştırılabilir. Bununla birlikte LeCun ve arkadaşları tarafından 1988 yılında geliştirilen LeNet mimarisi ilk ESA modeli olarak kabul edilmektedir (Li ve Ark., 2021). İlk yıllarda bu model posta kodlarını, rakamları ve el yazılarını tanımak için kullanılmıştır. 2012 yılında ImageNet yarışmasında elde ettiği başarının ardından görüntü tanıma alanında giderek artan bir popülerlik kazanarak, günümüzde bu alanda en çok kullanılan model olmuştur (O'Shea ve Nash, 2015).

ESA, yapısında birçok parametreyi barındırır. Bu parametreler her modele, probleme veya veri setine göre farklılık göstereceğinden, parametre seçimi ESA'da başlı başına bir iş haline gelmiştir. Uygulamada kullanılacak en uygun hiper-parametreleri bulmak için çeşitli çalışmalar yapılmış ve bazı yöntemler geliştirilmiştir. Bu yöntemlerden temel olan 4 tanesi; elle arama,

ızgara arama, rastgele arama ve genetik algoritma olarak listelenebilir. Literatürde arama yöntemleri üzerinde çalışmalar yapıldığı görülmektedir.

Smith (2017) tarafından derin sinir ağlarında öğrenme oranları ile ilgili yapılan çalışmada, öğrenme oranlarını belirlemek döngüsel bir yaklaşım ortaya atılmıştır. Çalışmada ResNet ve DenseNet modellerinde CIFAR-10 ve CIFAR-100 veri setleriyle, Alexnet ve GoogLeNet modellerinde ise ImageNet veri seti ile döngüsel öğrenme oranları olarak adlandırılan bu yaklaşım test edilmiştir. Smith çalışmada ortaya atılan bu yaklaşımın ideal öğrenme oranlarının belirlenmesindeki varsayımları azaltarak sinir ağları eğiten kişiler için pratik bir yöntem olacağını savunmuştur.

Bircanoğlu ve Arıca (2018), çalışmalarında yaygın olarak kullanılan Doğrusal, Sigmoid, Tanh, Hard Sigmoid, Softsign, ReLU, Softplus, ELU, SeLU, Swish aktivasyon fonksiyonları ve Kare aktivasyon fonksiyonunun sınıflandırma performanslarına etkilerini karşılaştırmışlardır. MINST, Fashion MNIST, CIFAR-10, Boston Evleri, Ames Evleri, IMDB ve Reuters veri setleri ile yapılan deneysel çalışmalar sonucunda veri setine göre en iyi performans gösteren aktivasyon fonksiyonunun değişiklik gösterdiğini ancak tüm veri kümelerinde ReLU aktivasyon fonksiyonu ile genel olarak iyi sonuçlar elde edildiği belirtilmiştir. Ayrıca bu çalışmada ilk defa aktivasyon fonksiyonu olarak kullanılan Kare fonksiyonunun da diğerlerine göre daha başarılı olduğu ifade edilmiştir.

Liashchynskyi ve Liashchynskyi (2019) bu arama yöntemlerinin karşılaştırılması üzerine yaptıkları çalışmalarında; ızgara arama yönteminin bir hiper-parametre uzayında eksiksiz bir arama yaptığını ancak buna bağlı olarak da çok maliyetli olduğunu belirtmekte ve arama alanının büyük olduğu durumlarda bu yöntemi önermemektedirler. Rastgele arama yönteminin daha hızlı olduğunu ancak bu yönteminde en iyi sonuçları garanti edemediğini ifade etmektedirler. Genetik algoritma yönteminin ise çalıştırılmasının uzun zaman aldığını ancak bazı değişkenlerle bu durumun kontrol edilebilir olduğunu söylemektedirler. Çalışmada, özellikle parametre seçeneğinin fazla olduğu durumlarda genetik algoritma yönteminin tercih edilebilir olduğu sonucuna varmışlardır.

Gülcü ve Kuş (2019), tarafından yayınlanan “Konvolüsyonel Sinir Ağlarında Hiper-Parametre Optimizasyonu Yöntemlerinin İncelenmesi” adlı çalışmada ise genetik algoritma, parçacık sürü optimizasyonu diferansiyel evrim ve bayes optimizasyonu yöntemlerini ESA üzerinde incelemişlerdir. Çalışmalarında CIFAR10, CIFAR100, EMNIST, MNIST, STL-10, UCL-50 gibi çok sayıda veri seti kümesi ile yaptıkları testler sonucunda en iyi hiperparametre setini seçmek için bir genelleme yapmanın mümkün olmadığını, hiperparametre optimizasyonunun

probleme ve veri setine göre değişebileceğini söylemişlerdir.

Seyyarer ve arkadaşları (2020) tarafından yayınlanan çalışmada Caltech 101 ve Caltech 256 veri setleri kullanılarak optimizasyon algoritmalarının sınıflandırma başarısındaki performansları karşılaştırılmıştır. Çalışmada 64x64 görüntü boyutu, ReLU aktivasyon fonksiyonu ve cross entropi hata fonksiyonu ile yapılan uygulamalarda optimizasyon algoritmalarının sınıflandırmadaki başarı oranları; sgd % 64.5, momentum % 85.56, adam % 92.31, adagrad % 71.25, rmsprop % 40.26 ve adadelta % 86.88 olarak bulunmuştur. Ayrıca büyük veri setlerinde hatayı en aza indirmek için gardient descent tabanlı optimizasyon algoritmaları (SGD, Momentum, Adam) kullanılması önerilmiştir.

Kendal ve Castelli (2020), çalışmalarında ESA'da paket boyutu ve öğrenme oranının ağın performansına etkilerini incelemişlerdir. ImageNet ağırlıklarına sahip VGG 16 modelini, paket boyutunu 16, 32, 64, 128, 256 değerlerinde Adam ve SGD optimizasyon algoritmalarıyla 0,001 ve 0,0001 öğrenme oranlarında test etmişlerdir. Çalışmada yapılan uygulamalar neticesinde öğrenme oranı ile paket boyutu arasında kuvvetli bir ilişki olduğu ve büyük parti boyutunda yüksek öğrenme oranlarının, düşük parti boyutunda ise düşük öğrenme oranlarının daha iyi performans gösterdiği sonucuna ulaşmışlardır.

Kılıçarslan ve arkadaşları (2021) tarafından aktivasyon fonksiyonları üzerine yapılan çalışmada sabit ve eğitilebilir aktivasyon fonksiyonları incelenmiştir. Çalışmada yaygın olarak kullanılan Sigmoid, Tanh, ReLU, Softplus ve Swish gibi sabit aktivasyon fonksiyonları ve literatürde yeni yer almaya başlayan LReLU, ELU, SELU ve RSigELU eğitilebilir aktivasyon fonksiyonları tanıtılmıştır. Yapılan incelemeler sonucunda sigmoid ve Tanh aktivasyon fonksiyonları dışında kaybolan gradyan probleminin üstesinden gelindiği ve ağı sürekli olarak eğitildiği ifade edilmiştir.

Adem (2022), çalışmasında FELU, ELU ve ReLU aktivasyon fonksiyonlarının birleşimi olan P+ FELU aktivasyon fonksiyonu etkilerini incelemiştir. MNIST, CIFAR-10 ve CIFAR-100 veri setleri ile çalışarak kıyaslamalar yapmıştır. Esnek özelliklere sahip P+ FELU aktivasyon fonksiyonunun kaybolan gradyanı etkili biçimde önleyebileceği belirtilmiştir. Çalışmada önerilen aktivasyon fonksiyonunun ELU, SELU, MPELU, TReLU, ReLU ve FELU aktivasyon fonksiyonlarından daha iyi performans gösterdiği sonucuna ulaşılmıştır.

3. MATERİYAL VE YÖNTEM

3.1. Veri Setleri

Bu çalışmada iki farklı veri seti kullanılmıştır. Veri setleri çeşitli alanlarda birçok içeriğe sahip Kaggle platformundan edinilmiştir. Kullanılan veri setleri üzerinde herhangi bir ön işlem veya düzenleme yapılmamıştır.

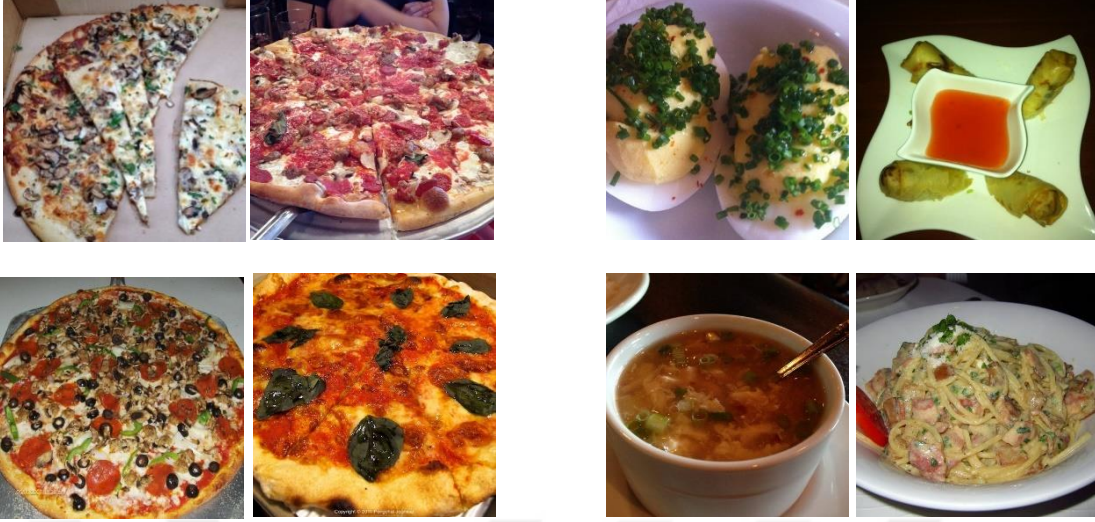
Birinci veri seti, Kaggle’da “Dogs & Cats Images” ismiyle yayınlanan veri setidir ve farklı açılardan çekilmiş kedi ve köpek görüntülerini içermektedir (Kaggle, 2018). Veri seti “cats” ve “dogs” olmak üzere iki sınıftan ve toplam 10000 görüntüden oluşmaktadır. Bu veri setine ait örnek görüntüler Şekil 2’de görülmektedir.



Şekil 2: Birinci veri setinden örnek görüntüler

Görüntülerin 8000 tanesi eğitim veri kümesine, 2000 tanesi ise test veri kümesine iki sınıfa eşit dağılımlı olarak ayrılmıştır.

İkinci veri seti de yine kaggle.com sitesinde “pizza classification data” ismiyle yayınlanan bir veri setidir (Kaggle, 2014). Farklı açılardan çekilmiş pizza ve pizza haricindeki yemekleri içeren görüntülerden oluşmaktadır. Görüntüler kenar uzunluğu en fazla 512 piksel olmak üzere farklı boyutlardadır. Veri seti “pizza” ve “not_pizza” olmak üzere iki sınıftan oluşmaktadır. Toplam 1966 görüntü 983’er adet olarak iki sınıfa eşit dağılmış durumdadır. Bu veri setine ait örnek görüntüler Şekil 3’te görülmektedir.



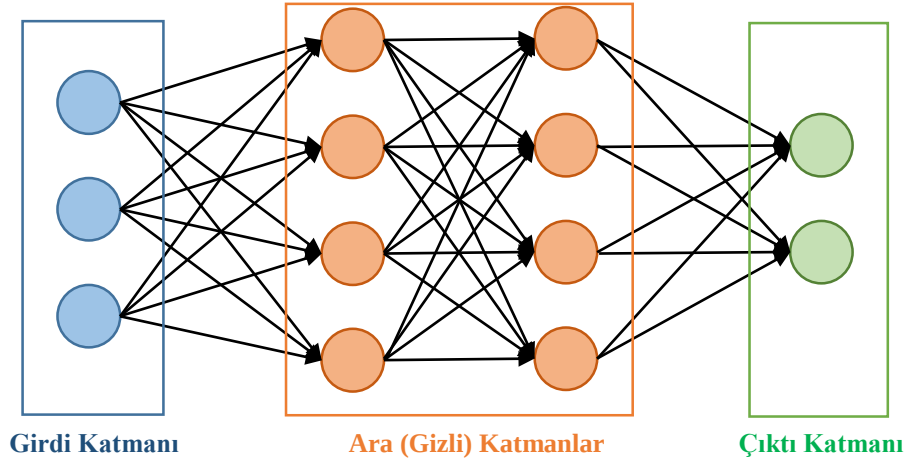
Şekil 3: İkinci veri setinden örnek görüntüler

Veri setindeki görüntülerin 1600 tanesi eğitim kümesine, 366 tanesi ise test veri kümesine ayrılmış durumdadır.

3.2. Yapay Sinir Ağları

Yapay sinir ağları (YSA) insan beyninin öğrenme yönteminden esinlenilerek geliştirilmiş bilgisayar yazılımlarıdır. İlk yapay sinir ağı modeli Warren McCulloch ve Walter Pitts tarafından 1943 yılında yapılan bir çalışmayla ortaya çıkarılmıştır. İlerleyen yıllarda YSA üzerine deneysel ve teorik çalışmalar yapılmakla birlikte 1990'lı yıllarda bilgisayar donanımlarının gelişmesiyle YSA daha uygulanabilir ve popüler hale gelmiştir (Keskenler ve Keskenler, 2017).

Biyolojik sinir sisteminde olduğu gibi YSA'nda da bilgileri işleyen ve birbirine aktaran sinir hücreleri vardır. YSA bu yapay sinir hücrelerinin birbirlerine katmanlar halinde bağlanmasıyla oluşur. YSA girdi katmanı, ara (gizli) katmanlar ve çıktı katmanı olmak üzere 3 temel katmandan oluşur. YSA'nın çalışma prensibi en sade şu şekilde açıklanabilir; girdi katmanından alınan bilgiler ara katmanlara iletilir ve burada bilgiler işlenerek çıktı katmanına gönderilir (Öztemel, 2003). Burada bilgiyi işleme kavramı ağı girilen bilgilerin ağırlık değerleri kullanılarak çıktıya dönüştürülmesidir. Ağı sunulan her veri örneği için ağırlıklar değiştirilir ve doğru ağırlık değerleri bulunmaya çalışılır. Bu işlem ağı eğitilmesi olarak tanımlanmaktadır.



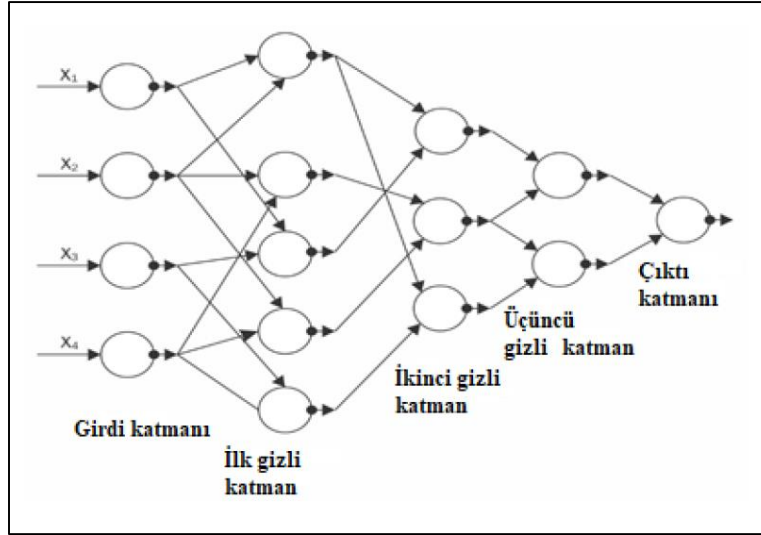
Şekil 4: Yapay Sinir Ağı Yapısı

YSA tıpkı insan beyni gibi verilen bilgilerden öğrenme, genelleme yapma ve yeni bilgiler türetme özelliklerine sahiptir (Öztürk ve Şahin, 2018). Literatür incelendiğinde yapay sinir ağları ile neredeyse her alanda çok çeşitli çalışmalar yapıldığı görülmektedir. Özellikle sağlık, endüstri, elektronik, uzay bilimleri, finans, askeri ve robotik alanlarında YSA etkin olarak kullanılmaktadır (Kaya, 2018). Teknolojide kullanılan cihazların donanımları ve yapay sinir ağı algoritmaları geliştikçe ilgi daha da artacaktır.

3.3. Derin Öğrenme

Derin öğrenme çok katmanlı yapay sinir ağlarının daha gelişmiş bir alt türüdür. Derin öğrenme modelleri çok sayıda katmandan oluşur ve katmanlar kendinden öncekilerin çıktılarını alarak özellik çıkarmayı amaçlar. Endüstri, akademi, sağlık, tarım vb. pek çok alanda ilgi görmüş ve çok sayıda uygulama geliştirilmiştir. Klasik makine öğrenmesi yöntemleri ile karşılaştırıldığında derin öğrenme doğrusal olmayan problemleri çözmede daha yeteneklidir (Hao ve Ark., 2016).

Ivakhnenko ve Lapa (1966) tarafından geliştirilen ilk ileri beslemeli ve çok katmanlı derin öğrenme modelinden günümüze derin öğrenme modellerinde ciddi ilerlemeler kaydedilmiştir (Şeker ve Ark., 2017).



Şekil 5: İlk ileri beslemeli ve çok katmanlı derin öğrenme modeli

Zaman içerisinde farklı veri kaynakları ve problemler için farklı derin öğrenme mimarileri de geliştirilmiştir. Örneğin görüntü tanıma için günümüzde en popüler yöntem ESA iken, konuşma tanıma veya el yazısı analizi gibi sıralı görevler için tekrarlayan sinir ağları (RNN) daha uygun görülmektedir. Ancak derin öğrenme modellerinde katman sayısı ve veri miktarı arttıkça hesaplama yükü de artmaktadır. Grafik işlemcilerin gelişmesiyle bu problem büyük oranda aşılsa da daha iyi bir derin öğrenme modeli elde etmek ve ideal optimizasyonu sağlamak için gerekli kaynaklara ulaşmak zor olabilmektedir.

3.4. Derin öğrenme Kütüphaneleri

Derin öğrenme kütüphaneleri; makine öğrenmesi, resim ve ses işleme, doğal dil işleme ve diğer pek çok alanda kullanılan derin öğrenme modellerinin tasarımı, eğitimi ve uygulanması için kullanışlı araç ve kaynaklar sağlar. Bu kütüphaneler kullanıcılara öğrenme algoritmaları oluşturmak için önemli kolaylıklar sağlar (İnik ve Ülker, 2017). Derin öğrenme modellerinin tasarlanması ve eğitimi için açık kaynak kodlu birçok kütüphane geliştirilmiştir. En popüler derin öğrenme kütüphaneleri şunlardır;

- **TensorFlow:** Google tarafından geliştirilen açık kaynaklı bir derin öğrenme kütüphanesidir. Gelişmiş veri akışı grafikleri ve farklı cihazlardaki esnek kullanım özellikleriyle öne çıkar.
- **Keras:** Python dilinde geliştirilmiş yüksek düzey bir derin öğrenme kütüphanesidir. TensorFlow ve Theano üzerinde çalıştırılabilen uygulama programlama ara yüzleri (API) sayesinde modellerin kolayca oluşturulmasına ve eğitilmesine olanak tanır.
- **PyTorch:** Facebook tarafından 2016 yılından beri geliştirilen açık kaynaklı bir derin öğrenme kütüphanesidir. Dinamik hesaplama yöntemleri kullanarak esnek modeller

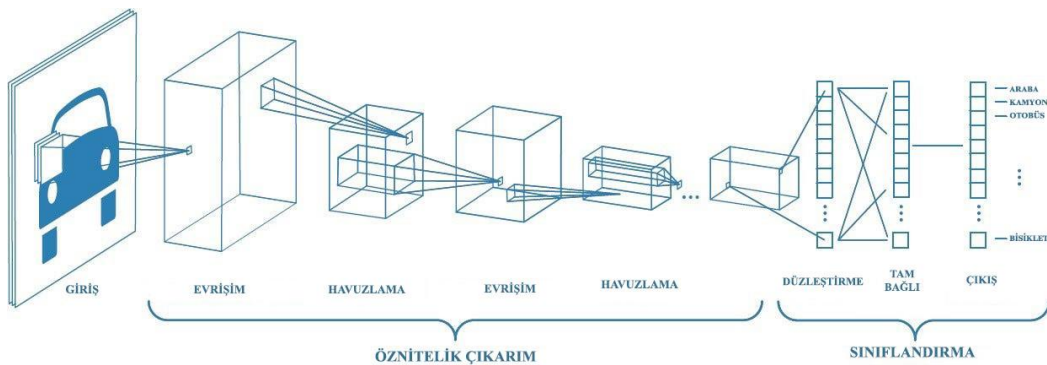
oluşturma imkânı sunar. PyTorch birçok kütüphaneyle entegre şekilde çalışabilir.

- **Caffe:** NVIDIA'nın desteğiyle Kaliforniya Üniversitesi tarafından geliştirilen açık kaynaklı bir platformdur. Bilgisayar Birleşik Cihaz Mimarisi (CUDA) kütüphanesi kullanılarak C dilinde geliştirilmiştir (Uçar ve Bingöl, 2018). Ayrıca GPU hızlandırmasına uyumludur.
- **Theano:** Montreal Üniversitesi tarafından geliştirilen ve Python dilini destekleyen açık kaynaklı bir yazılımdır. Theano temel olarak derin öğrenme modelleri oluşturmayı kolaylaştıran bir matematiksel ifade kütüphanesidir (Aktürk ve Serbest, 2022). Hızlı numPY hesaplama özellikleri ile az sayıda satırla modellerin oluşturulmasına olanak tanır.
- **Torch:** Derin öğrenme için bir dizi araç ve kütüphane sunan Berkeley Software Distribution (BSD) lisansına sahip açık kaynaklı bir platformdur. Hem CPU, hem GPU üzerinde çalışma özelliğine sahiptir.

Çalışmamızda alt kütüphane olarak TensorFlow, sağladığı kolaylıklar ve güncel olması sebebiyle üst düzey kütüphane olarak da Keras kütüphanesi kullanılmıştır.

3.5. Evrişimsel Sinir Ağları

ESA, kazandığı popülerlik ve elde ettiği başarı oranları sebebiyle pek çok alanda farklı uygulamalarda kullanılmaktadır. Bu yoğun kullanım neticesinde ise zaman içerisinde çok sayıda ESA modeli geliştirilmiştir (Şeker ve Ark., 2017). ESA temel olarak 4 katmandan oluşur. ESA modellerinde bu katmanların sayıları ve dizilişleri birbirinden farklı olabilir (O'Shea ve Nash, 2015).



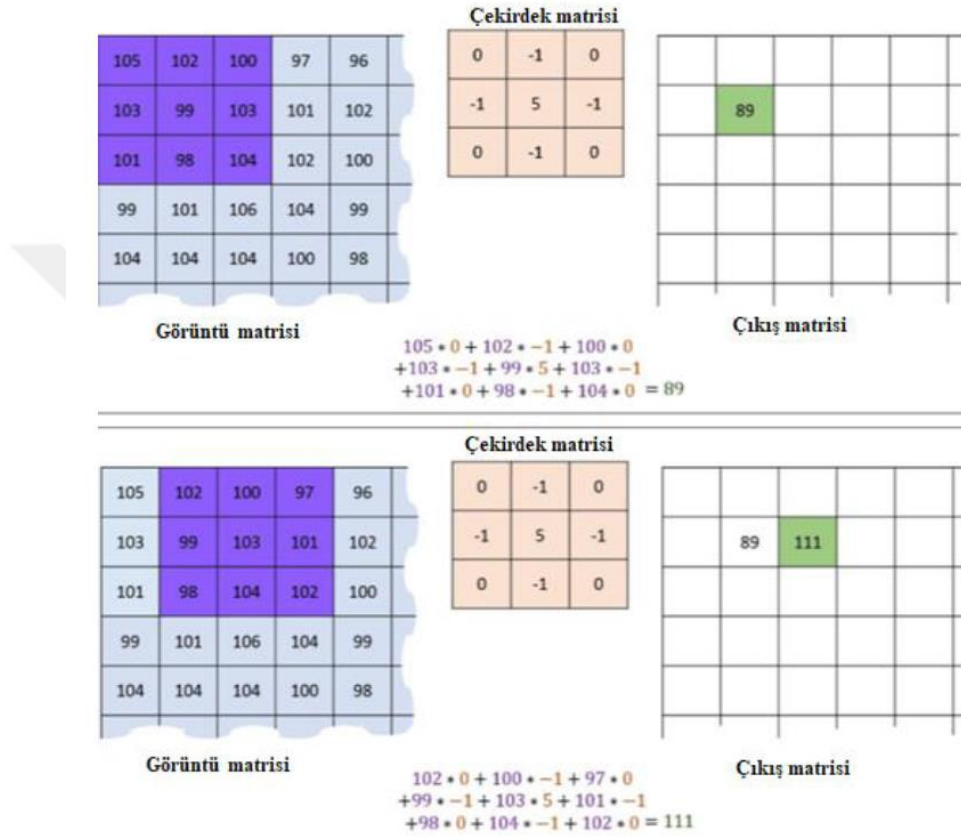
Şekil 6: Evrişimsel Sinir Ağı Modeli

3.5.1. Girdi Katmanı

YSA'nın diğer formlarında olduğu gibi, görüntünün sayısal değerlerinin tutulduğu ve sisteme verildiği katmandır. Bu sayısal değerler matrisler halinde tutulur ve görüntünün özelliklerine göre matrisin boyutları değişebilir.

3.5.2. Evrişim katmanı

Yapının temel katmanıdır ve girdiden öznitelikler çıkarmayı amaçlar. Bu işlem, bir filtre ya da kernel'ın, girdi olarak verilen bir görüntü üzerinde gezdirilmesiyle gerçekleştirilir (Deng ve Yu, 2014). Örneğin 3x3 boyutlarında bir filtre, görüntünün her bir pikselinde küçük bir pencere boyunca gezdirilerek evrişim işlemi uygulanır. Şekil 7'de örnek evrişim işlemleri görülmektedir.



Şekil 7: Evrişim İşlemi (Arslan, 2021)

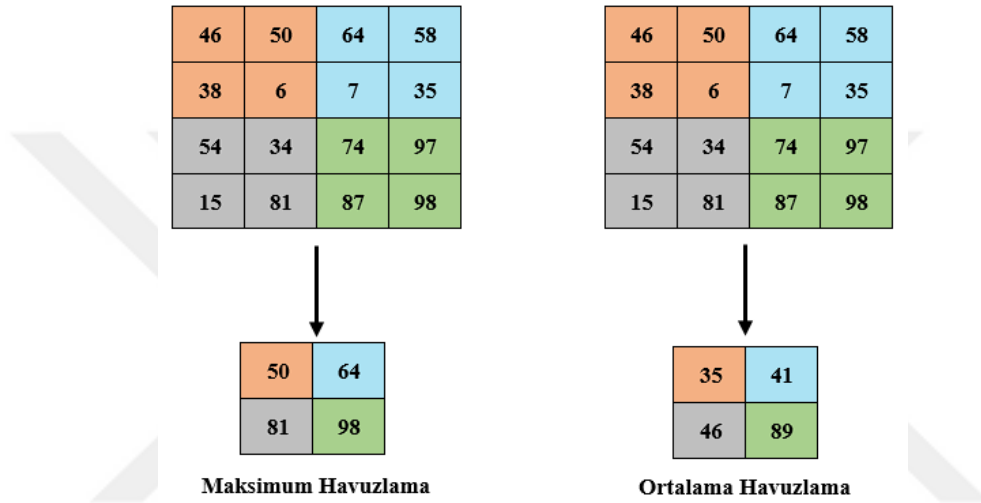
Şekil 7'de görüldüğü gibi görüntüdeki değerler ile filtre üzerinde örtüşen değerler çarpılarak toplanır ve çıkış matrisine eklenir. Kenarlardaki piksellerin değerlerini hesaplamak içinse görüntünün tüm kenarlarına sıfır eklenerek hesaplama işlemi yapılarak yine çıkış matrisine eklenir. Oluşturulan bu çıkış matrisine öznitelik haritası da denilir (Uçar ve Bingöl, 2018). Bu işlem çok katmanlı veriler için genişletilebilir. Örneğin renkli görüntülerden oluşan girdiler üç katmandan oluşacağı için filtreler de üç katmanlı olmalıdır. Bu durumda filtrenin her katmanı girdinin ilgili katmanında gezdirilir (Kurt, 2018). Elde edilen öznitelik haritaları aktivasyon fonksiyonundan geçirilerek sonraki evrişim katmanının girişine verilir.

Uygulanan filtreler farklı boyutlarda ve farklı şekillerde olabilir. Evrişim işleminde birden fazla

filtre kullanılabilir ve her filtre için bir evrişim işlemi yapılandırılabilir (Dumoulin ve Visin, 2016). Bu işlemler neticesinde görüntüden düşük veya yüksek seviyeli özellikler çıkarılmış olur.

3.5.3. Havuzlama Katmanı

Evrişim katmanlarından sonra elde edilen çıkış matrislerinin boyutunu indirmek ve işlemleri hızlandırmak için havuzlama katmanları kullanılır. Havuzlama işlemleri maksimum havuzlama (maxpooling) ve ortalama havuzlama (average pooling) yöntemleri ile gerçekleştirilir. Havuzlama katmanında uygulanan bu yöntemlere ait örnek işlemler Şekil 8’de görülmektedir.



Şekil 8: Havuzlama işlemleri

Şekil 8’de görüldüğü gibi her iki yöntemde de girdi matrisi parçalara ayrılır ve belirlenen adım sayısı ile hesaplama işlemi yapılır. Maksimum havuzlamada her parça matrisin içerisindeki en yüksek değer alınırken, ortalama havuzlamada matrisin içindeki değerlerin ortalaması alınır.

3.5.4. Tam Bağlı Katman

Evrişim ve havuzlama katmanlarında elde edilen sonuçlar bu katmana girdi olarak alınır. Tam bağlı katman standart bir yapay sinir ağında bulunan aynı görevleri yerine getirerek ve sınıflandırma için kullanılacak aktivasyonlardan sınıf puanları üretmeye çalışır.

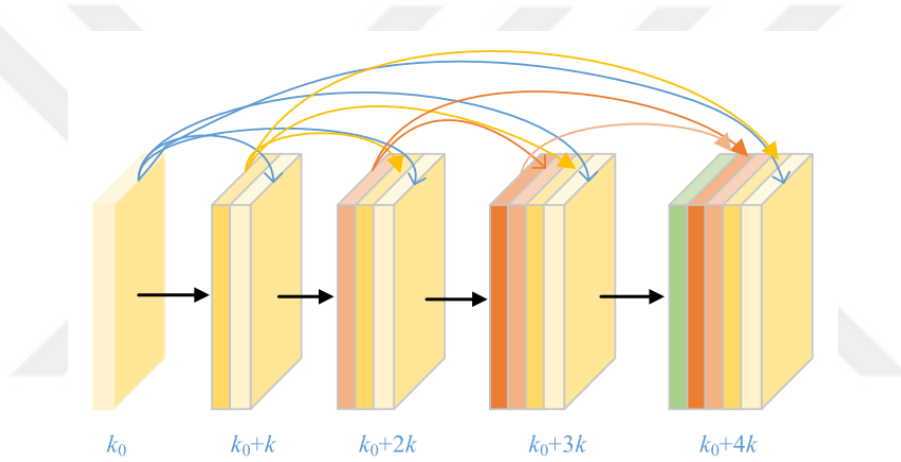
3.6. Evrişimsel Sinir Ağı Modelleri

ESA’nın yükselişine 2012 yılında sunulan AlexNet modeli öncülük etmiştir. Daha sonra ESA mimarileri kullanılarak verimli ve kullanışlı modeller tasarlanmıştır (Tuba ve Ark., 2021). Tasarımcılar tarafından bu modeller her geçen gün geliştirilmeye ve yeni modeller sunulmaya devam etmektedir. Bununla birlikte popüler modellerin bir araya getirilerek kullanıma sunulduğu kütüphaneler de mevcuttur. Yeni bir model tasarlamaya göre daha kolay kullanım imkanı sunan bu kütüphaneler, özellikle sağlık, tarım, üretim ve sanayi gibi alanlarda ESA ile

yapılan çalışmalarda yaygın olarak kullanılmaktadır. Keras kütüphanesi bu anlamda oldukça güncel ve popüler ESA modelleri sunan bir kütüphanedir. Bu çalışmada, Keras kütüphanesindeki 27 ESA modeli üzerinde deneysel çalışmalar yapılmıştır. Bu modellerde iki veri seti ile eğitimler gerçekleştirilmiş ve en yüksek doğruluk oranına ulaşılan DenseNet201 ve NASNetMobile modellerine alt bölümlerde yer verilmiştir.

3.6.1. DenseNet

DenseNet mimarisinin temelini evrişim katmanlarında yer alan ve modele adını veren dense blokları oluşturmaktadır. Hiyerarşik yapıya sahip olan geleneksel ESA modellerinden farklı olarak DenseNet, her katmanın sadece kendinden önceki katmanın çıktıları yerine kendinden önceki tüm katmanların çıktıları ile beslenmesidir. DenseNet mimarisinin katman yapısı Şekil 9'da görülmektedir.

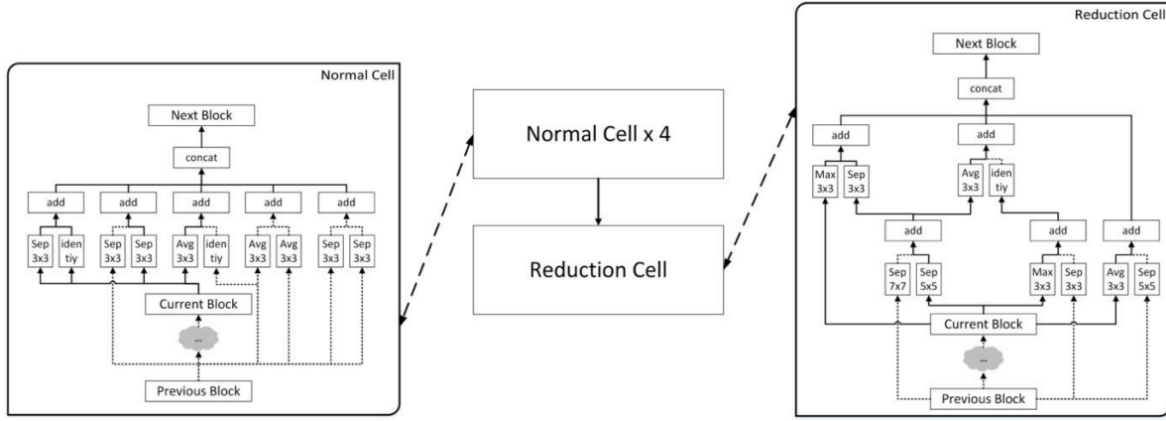


Şekil 9: DenseNet Katman Yapısı (Wang ve Zhang, 2020)

Şekil 9'da görüldüğü gibi DenseNet mimarisinde tüm bloklar birbirine bağlıdır. DenseNet'in farklı derinliklere sahip modelleri bulunmaktadır. Bu modeller DenseNet121, DenseNet169 ve DenseNet201'dir. Modellerin sonundaki sayılar ağıın derinliğini ifade etmektedir. Bu bağlamda DenseNet201 modeli 201 katmandan oluşmaktadır (Adhinata ve Ark., 2021).

3.6.2. NASNetMobile

NASNetMobile Google tarafından Nöral Mimari Arama (Neural Architecture Search - NAS) yönteminin temelinde derin öğrenme yoluyla 2017 yılında tasarlanmış bir yapay sinir ağı modelidir. (Adedoja ve Ark., 2022). NASNet pekiştirmeli öğrenme ile iyileştirilebilen hücrelerden oluşturulmuştur. Bu hücrelerde birkaç tane evrişim ve havuzlama işlemleri gerçekleştirilir ve ağıın kapasitesine göre birçok kez tekrarlanır. NASNetMobile modelinin mimarisi Şekil 10'da verilmiştir.



Şekil 10: NasNetMobile Mimarisi (Daşgın, 2023)

Şekil 10’da görüldüğü gibi 12 hücreden oluşan NASNet Mobile versiyonu 769 katmana, 224x224 giriş boyutuna ve 5,3 milyon parametre ile 564 milyon çarpma hacmine sahiptir. (Saxen ve Ark., 2019). Bu model, görüntü sınıflandırma, nesne algılama ve nesne takibi gibi görsel işleme görevlerinde kullanılır. NASNetMobile, hafif ve küçük boyutlu olmasına rağmen ImageNet veri kümesindeki sınıflandırma görevinde diğer popüler mimarilerle rekabet edebilecek yüksek bir doğruluk oranına sahiptir.

3.7. Evrimsel Sinir Ağlarında Hiperparametreler

Hiperparametreler modelin eğitimde kullanılan önceden belirlenmiş kontrol parametreleridir (Doğan, 2021). ESA modellerinde kullanılan parametrelerin bazıları katmanlara özeldir bazıları ise modelin ortak parametreleri olarak kabul edilir. Bu parametreler ve katmanlara göre gruplandırılmaları Tablo-1’de görülmektedir.

Tablo 1: Optimize edilebilecek hiperparametreler (Gülcü ve Kuş, 2019)

Evrişim	Havuzlama	Tam bağı	Genel
• Filtre boyutu • Filtre sayısı • Adım (stride) • Dolgu (padding) • Aktivasyon fonksiyonu • Katman sayısı • Başlangıç katman sayısı	• Filtre boyutu • Adım (stride) • Ortaklama alg. • Katman sayısı	• Nöron sayısı • Katman sayısı • Başlangıç katman sayısı	• Optimizasyon metodu • Batch boyutu • Öğrenme oranı • Başlangıç öğrenme oranı • Seyreltme oranı (dropout) • Seyreltme aktif • Düzenleştirme metodu • Düzenleştirme oranı • Ağırlık başlatıcı (weight initializer) • Ağırlık düşürücü • Ağırlık çarpanı • Ağırlık normalizasyonu • Ağırlık ceza değeri

			• İterasyon sayısı • Momentum • Yanlılık aktif (bias) • Yanlılık başlangıç • Yanlılık başlangıç oranı • Gauss oranı
--	--	--	--

3.7.1. Eğitim Tur (Epoch) Sayısı

Bir sinir ağı modeli eğitilirken, eğitim veri kümesi genellikle birden çok kez ağıdan geçirilir. Ağın öğrenme algoritmasında tüm eğitim veri kümesinin kaç defa çalıştırılacağını belirleyen parametreye eğitim tur (Epoch) sayısı denir (Kurt, 2018).

Epoch sayısı bir ile sonsuz arasında bir değer alabilir. Literatürde epoch sayıları genellikle 100, 200, 500 gibi yüksek değerlerden seçilmektedir (Brownlee, 2018). Ancak epoch sayısının yüksek olması ağın daha iyi eğitileceği ya da daha iyi başarı değerleri elde edeceği anlamına gelmez. Bununla birlikte epoch sayısı ne kadar fazlaysa modelin eğitimi için gereken süre de o kadar fazla olacaktır (Hastomo ve Ark., 2021). Diğer parametrelerde olduğu gibi epoch sayısını seçmek için de sabit kurallar yoktur. En uygun epoch sayısını belirlemek için farklı değerler denemek gereklidir. Bu da zor ve maliyetli bir iştir.

3.7.2. Nöron Sayısı

Sinir ağlarında nöron sayısı, ağın karmaşıklığı ve öğrenme yeteneği üzerinde önemli etkiye sahip parametrelerden biridir. Nöron sayısı arttıkça, ağın öğrenme kapasitesinin de artması beklense de bu her zaman geçerli değildir. Ağın mimarisinde gereğinden fazla nöron kullanmak aşırı uydurmaya (overfitting) ve genelleme yeteneğinin zayıflamasına yol açabilir (Zhang ve Ark., 2021). Ayrıca nöron sayısının fazla olması ağın boyutunu ve hesaplama yükünü de artırır. Donanım kapasitesinin düşük olması halinde bu durum zaman açısından negatif etkiye sebep olur. Nöron sayısının gerekenden az olması durumunda ise modelin veri kümesini temsil edememe sorunu ortaya çıkar (Ünal ve Ark., 2022).

Modelin katmanlarındaki nöron sayısının ağın performansını etkilediği bilinmektedir. Ancak nöron sayısının ağın performansına etkisi ile ilgili öngöründe bulunmak, dolayısıyla ideal nöron sayısını belirlemek zordur. Çünkü katman ve nöron sayısı ile birlikte ağın performansını etkileyen birçok parametre vardır (Kızrak, 2018).

3.7.3. Paket Boyutu (Batch Size)

Sinir ağı modeli eğitilirken tüm eğitim verisini aynı anda ağıdan geçirmek tercih edilen bir yöntem değildir. Bunun yerine veri seti parçalar halinde ileri ve geri yayımlı olarak adım adım

ağdan geçirilir. Her eğitim turunda veri setinden alınan bu parçalardaki veri miktarına ya da örnek sayısına paket boyutu (Batch Size) denilmektedir (Smith, 2018).

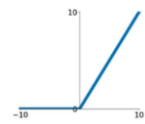
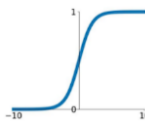
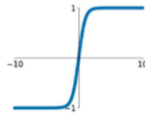
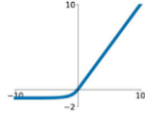
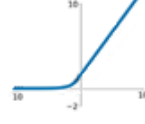
Literatürdeki çalışmalarda paket boyutun genellikle 32, 64 ve 128 örnekten oluşturulduğu görülmektedir (Brownlee, 2018). Paket boyutları sabit veya eğitim turuna göre değişken olarak da ayarlanabilir. Sabit ayarlanan paket boyutları da veri setindeki örnek sayısına bağlı olarak eşit olmayabilir. Genellikle son paketin boyutu diğerlerinden farklı olur. Bu durumu ortadan kaldırmak için veri setindeki örnek sayısı paket sayısına bölünerek paket boyutu ayarlanabilir.

3.7.4. Aktivasyon Fonksiyonları

ESA modellerinde bir katmandaki nöronlarda üretilen çıktı değerleri sonraki katmana iletilirken aktivasyon fonksiyonları kullanılır. Aktivasyon fonksiyonları; katmanlardaki çıktı değerlerinin eşik değerleri belirlenerek sonraki katmana iletilip ileilmeyeceğine, dolayısıyla yapay sinir hücresinin aktif olup olmayacağına karar vermek için kullanılır (Ser ve Bati, 2019).

ESA genellikle doğrusal olmayan sınıflandırılmalarda kullanıldığından, aktivasyon fonksiyonları da doğrusal olmayan fonksiyonlardan seçilir. Bu fonksiyonlar genellikle $[-1,1]$ veya $[0,1]$ aralığında çıktı değerleri üretir. Keras kütüphanesinde yer alan ve en çok kullanılan aktivasyon fonksiyonlarının matematiksel formülleri ve çıktı grafikleri Tablo-2’de verilmiştir.

Tablo 2: Aktivasyon Fonksiyonları (Opengeni, 2023)

Aktivasyon Fonksiyonu	Matematiksel Formülü	Grafik
ReLU	$\max(0, x)$	
Sigmoid	$\sigma(x) = \frac{1}{1 + e^{-x}}$	
Tanh	$\tanh(x)$	
ELU	$\begin{cases} x & x \geq 0 \\ \alpha(e^x - 1) & x < 0 \end{cases}$	
Softplus	$\ln(1 + e^x)$	

3.7.5. Optimizasyon Yöntemleri

Yapay sinir ağlarında, ağın verdiği çıkış değeri ile gerçek değer arasındaki en iyi farkı bulmak için optimizasyon yöntemleri kullanılır. Literatürde bu optimizasyon yöntemlerine gradyan iniş (Gradient Descent) de denilmektedir (Seyyarer ve ark., 2020).

ESA uygulamalarında amaç çoğunlukla hata oranını en aza indirmek değil en iyi genellemeyi yapabilmektir. ESA modelinde kullanılan optimizasyon algoritmasının seçimi bu genellemeyi yapmada etkili olacağı bilinmektedir. Derin öğrenmede optimizasyon yöntemi olarak en çok kullanılan algoritmalar SGD, Adagrad, RMSProp, Adadelta, Adam ve Adamax'tır (Sun, 2019).

Seyyarer ve arkadaşları (2020) çalışmalarında optimizasyon algoritmalarının kıyaslamalarını yapmışlardır. Çalışmada ağın yapısına ve veri setinin büyüklüğüne göre kullanılan algoritmalar farklı başarımlar elde etmişlerdir.

3.7.6 Öğrenme Oranı (Learning Rate)

Öğrenme oranı, optimizasyon algoritmalarında yakınsama hesaplamalarında kullanılan bir katsayıdır. Öğrenme oranı sinir ağlarının eğitiminde oldukça önemli parametrelerden biridir (Smith, 2017). Bu oranın fazla büyük belirlenmesi yakınsama amacına ulaşmayı başaramayacağı hatta ıraksamaya neden olacağı, çok küçük belirlenmesi ise küçük adımlarla daha iyi bir yakınsama sağlayacağı bilinmektedir (Bengio, 2012). Ancak öğrenme oranı küçüldükçe optimizasyon algoritması daha küçük adımlarla ilerleyeceği için eğitim süresi uzayacaktır. Optimizasyon algoritmalarında öğrenme oranının tam olarak ayarlanması oldukça zordur (Ser ve Bati, 2019).

Öğrenme oranı bir değere sabitlenebilir veya eğitim sırasında belirlenen bir oranda değişen olarak ayarlanabilir. Smith (2017) ise sinir ağları eğitiminde döngüsel öğrenme oranları üzerine yaptığı araştırmasında; öğrenme oranını belirli sınırlar içinde döngüsel olarak değişen bir yöntemle yapmıştır. Söz konusu çalışmada bu yöntemin daha iyi düzeyde sınıflandırma doğruluğu sağladığı ve öğrenme oranı ayarlama işini ortadan kaldırdığı belirtilmektedir.

4. BULGULAR

Çalışmada kullanılan ESA modellerinin tasarlanması ve eğitim işlerinin uygulanmasında Google Colab platformu kullanılmıştır. Transfer öğrenme modellerin test edilmesi ve ikinci veri seti ile yapılan uygulamalar, standart Colab üyeliği ile sunulan 12.7 GB RAM, 15 GB GPU ve 72.8 GB disk alanı donanımlarıyla gerçekleştirilmiştir. Birinci veri seti ile gerçekleştirilen uygulamalar ise Colab Pro üyeliğine sağlanan 85 GB RAM, 40 GB Tesla A100 GPU ve 170 GB disk alanı ile çalıştırılmıştır.

4.1. Evrimsel Sinir Ağı Modelleri

ESA karmaşık problemler için etkili çözümler sunsa da bir sinir ağı modeli tasarlamak ve onu oluşturmak uzmanlık gerektiren zor bir iştir. Derin öğrenme kütüphaneleri bu anlamda sinir ağı modellerini çok daha kolay şekilde kullanılmasını sağlar. Ayrıca bu alandaki birçok kütüphanenin açık kaynaklı olması da erişilebilirlik açısından oldukça avantajlıdır. Keras kütüphanesi birçok ESA modelini kullanım kolaylığı ile kullanıcılarına sunan en popüler kütüphanelerdendir. Bu modeller isteğe bağlı olarak ImageNet veri seti ile önceden eğitilmiş ağırlıklarla da kullanılabilir. Çalışmada kullanılacak modeli belirlemek için veri setlerimizle Keras kütüphanesindeki 27 model üzerinde performans testleri yapılmıştır. Parametrelerin etkilerinin incelenebilmesi amacıyla, modeller Keras kütüphanesi tarafından sağlanan eğitilmiş ağırlıklar olmadan veri setleri ile eğitilmiştir. Bu eğitimlerde belirlenen parametreler aşağıdaki gibidir;

- Eğitim tur sayısı (Epoch) = 10
- Paket boyutu = 32
- Aktivasyon fonksiyonu = ReLU,
- Optimizasyon algoritması = Adam
- Öğrenme oranı = 0,001

Diğer parametreler modelde varsayılan değerlerle kullanılmıştır. İki veri seti için de test edilen modellerin performans değerleri Tablo 3'te verilmiştir.

Tablo 3: ESA Modellerinin performansları

Birinci Veri Seti Sonuçları					İkinci Veri Seti Sonuçları				
Sıra	Model	Doğruluk Değeri	Epochs	Eğitim Süresi	Sıra	Model	Doğruluk Değeri	Epochs	Eğitim Süresi (Sn)
1	NASNetMobile	0.6175	10	273.46	1	DenseNet201	0.6687	10	66.10
2	ResNet101V2	0.6167	10	267.62	2	InceptionResNetV2	0.6675	10	85.16
3	DenseNet169	0.6162	10	278.22	3	NASNetMobile	0.6667	10	70.95
4	ResNet50	0.6150	10	257.73	4	DenseNet121	0.6562	10	70.56
5	DenseNet201	0.6113	10	280.25	5	ResNet152V2	0.6438	10	95.95
6	ResNet152V2	0.6100	10	279.85	6	ResNet101V2	0.6375	10	68.71
7	DenseNet121	0.6087	10	282.77	7	DenseNet169	0.6313	10	68.42
8	ResNet50V2	0.6012	10	258.21	8	ResNet50	0.6187	10	63.61
9	InceptionResNetV2	0.5900	10	286.43	9	ResNet50V2	0.6125	10	62.77
10	VGG16	0.5838	10	256.08	10	InceptionV3	0.5250	10	61.22
11	VGG19	0.5038	10	264.16	11	EfficientNetB6	0.4750	10	72.89
12	ResNet152	0.5038	10	285.43	12	EfficientNetB3	0.4750	10	70.71
13	ResNet101	0.5038	10	274.85	13	VGG19	0.4750	10	54.21
14	Xception	0.5038	10	258.15	14	VGG16	0.4750	10	63.87
15	EfficientNetB7	0.5038	10	306.26	15	EfficientNetB0	0.4750	10	68.57
16	EfficientNetB2	0.5038	10	268.53	16	EfficientNetB1	0.4750	10	67.11
17	EfficientNetB1	0.5038	10	283.10	17	EfficientNetB2	0.4750	10	59.88
18	EfficientNetB0	0.5038	10	261.37	18	ResNet152	0.4750	10	92.00
19	MobileNet	0.4963	10	251.58	19	ResNet101	0.4750	10	79.33
20	MobileNetV3Large	0.4963	10	270.40	20	EfficientNetB7	0.4750	10	75.78
21	MobileNetV3Small	0.4963	10	251.78	21	EfficientNetB4	0.4750	10	77.36
22	MobileNetV2	0.4963	10	261.31	22	MobileNetV3Small	0.4750	10	58.07
23	EfficientNetB6	0.4963	10	278.30	23	MobileNetV3Large	0.4750	10	63.32
24	EfficientNetB5	0.4963	10	292.48	24	MobileNet	0.4750	10	51.30
25	EfficientNetB4	0.4963	10	293.12	25	MobileNetV2	0.4750	10	47.62
26	EfficientNetB3	0.4963	10	271.77	26	EfficientNetB5	0.4750	10	74.51
27	InceptionV3	0.4963	10	261.88	27	Xception	0.4750	10	64.10

Tablo 3’te görüldüğü gibi birinci veri seti ile yapılan uygulamalarda en yüksek doğruluk oranı “NASNetMobile” modeli ile elde edilmiştir. İkinci veri seti ile yapılan uygulamalarda ise en iyi sınıflandırma performansı “DenseNet201” modeli ile elde edilmiştir. İki veri setinde en iyi performansı gösteren modeller ile çalışmamıza devam edilmiştir.

4.2. Hiperparametrelerin Uygulanması

ESA’da kullanılan kontrol parametreleri Tablo 1’de verilmişti. Bu çalışmada epoch sayısı, paket boyutu, aktivasyon fonksiyonu, optimizasyon yöntemleri, öğrenme oranı ve nöron sayısı parametreleri üzerinde deneysel çalışmalar yapılmıştır. Bu çalışmalar, İki veri seti içinde en yüksek doğruluk oranı elde edilen modellerde ayrı ayrı yapılmıştır. Modellerin eğitiminde Keras tarafından sağlanan önceden ImageNet ile eğitilmiş ağırlıklar kullanılmamıştır.

4.2.1. Eğitim Tur (Epoch) Sayısı

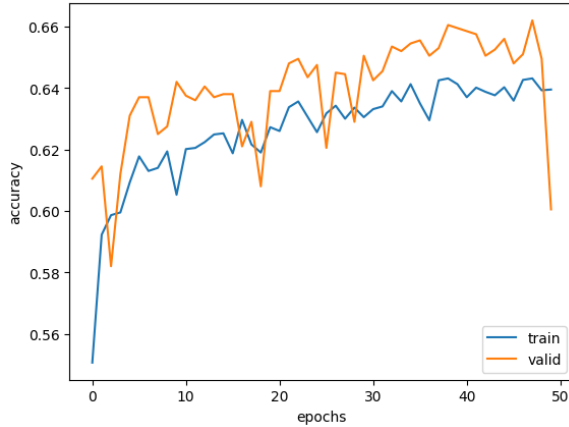
Eğitim tur sayısı tüm veri setinin algoritma üzerinden kaç defa geçirileceğini gösteren parametredir. Çalışmamızda 27 model arasından en iyi performansı gösteren modeller 5, 10, 25 ve 50 epoch sayıları ile eğitilmiştir. Bu eğitimlerde öğrenmeyi ideal noktada bitiren

“EarlyStopping” metodu kullanılmamış ve öğrenmenin son epoch’a kadar sürmesine izin verilmiştir. NASNetMobile ve DenseNet201 modellerinin farklı epoch değerleri ile eğitimleri sonucunda elde edilen performans değerleri Tablo 4’te verilmiştir.

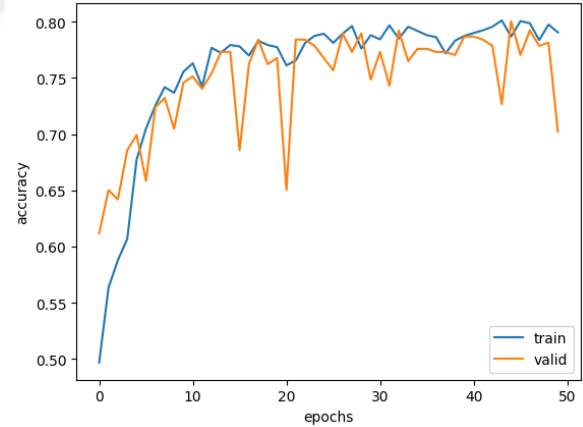
Tablo 4: Epoch sayısına göre eğitim sonuçları

Birinci Veri Seti						İkinci Veri Seti					
No	Model	Epochs	Loss	Doğruluk Değeri	Süre (Saniye)	No	Model	Epochs	Loss	Doğruluk Değeri	Süre (Saniye)
1	NASNetMobile	5	0,6936	0,5615	359	5	DenseNet201	5	0,5441	0,6639	73
2	NASNetMobile	10	0,6407	0,62	528	6	DenseNet201	10	0,5221	0,7459	174
3	NASNetMobile	25	0,6337	0,627	1296	7	DenseNet201	25	0,5268	0,724	350
4	NASNetMobile	50	0,6534	0,6005	2536	8	DenseNet201	50	0,5673	0,7022	776

Tablo 4’e bakıldığında birinci veri seti üzerinde NASNetMobile modeli ile yapılan çalışmalarda en yüksek doğruluk oranının 25 Epoch değerinde elde edildiği, ikinci veri seti üzerinde DenseNet201 modeli ile yapılan eğitimlerde ise en yüksek doğruluk oranının 10 epoch ile elde edildiği görülmektedir. Şekil 11 ve 12’de iki veri seti ile 50 epoch değerinde yapılan eğitimlere ait eğitim ve test doğruluk oranları grafiği verilmiştir.



Şekil 11: 4 no'lu işe ait epoch ve doğruluk oranı grafiği



Şekil 12: 8 no'lu işe ait epoch ve doğruluk oranı grafiği

Şekil 11’e bakıldığında 4 numaralı işte yaklaşık 45. eğitim tur sayısından sonra test doğruluk oranının aniden düştüğü görülmektedir. Aşırı uydurma (Overfitting) ihtimali göz önünde bulundurularak bu veri seti ile yapılan çalışmalar da 50 epoch ile sınırlandırılmıştır. Şekil 12’de görüldüğü üzere 8 numaralı işin eğitim doğruluk oranının belli bir epoch sayısından sonra yatay olarak ilerlemeye devam etmektedir. Bu veri seti nispeten daha küçük olmasına bağlı olarak, ağın artık daha fazla öğrenemeyeceğine karar verilmiş ve çalışmalar 50 epoch ile

sınırlandırılmıştır. Sonraki eğitimlerde epoch sayısı olarak modellerin en yüksek başarımları elde ettiği değerler kullanılmıştır.

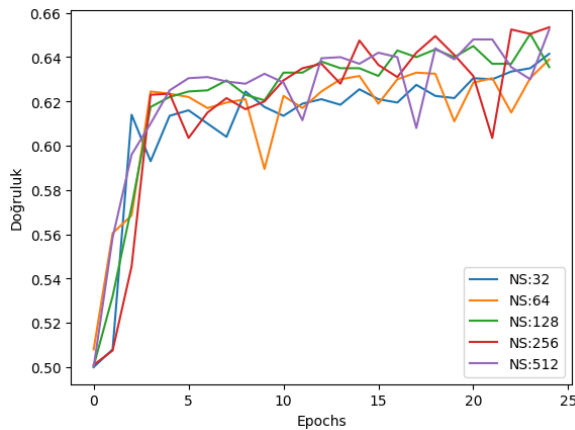
4.2.2. Nöron Sayısı

Nöron sayısı ağıın büyüklüğünü ve kapasitesini belirleyen önemli faktörlerdendir. İki model için de ideal epoch sayısını belirledikten sonra farklı nöron sayıları ile eğitimler gerçekleştirildi. Nöron sayısı olarak 32, 64, 128, 256 ve 512 değerleri üzerinde denemeler yapılmıştır. Tablo 5'te görüldüğü üzere her iki model içinde en yüksek doğruluk oranını sağlayan nöron sayısı 256'dır.

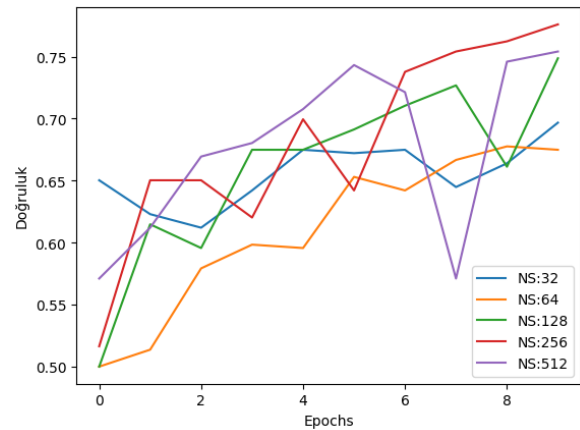
Tablo 5: Nöron sayısına göre eğitim sonuçları

Birinci Veri Seti							İkinci Veri Seti						
No Model		Epoch	Nöron Sayısı	Loss	Doğruluk Değeri	Süre	No Model		Epoch	Nöron Sayısı	Loss	Doğruluk Değeri	Süre
9	NASNetMobile	25	32	0,5255	0,6315	808	14	DenseNet201	10	32	0,5255	0,6967	112
10	NASNetMobile	25	64	0,5047	0,639	806	15	DenseNet201	10	64	0,5047	0,6749	104
11	NASNetMobile	25	128	0,4897	0,6355	805	16	DenseNet201	10	128	0,4897	0,7486	104
12	NASNetMobile	25	256	0,5244	0,6535	804	17	DenseNet201	10	256	0,5244	0,776	103
13	NASNetMobile	25	512	0,5126	0,6525	806	18	DenseNet201	10	512	0,5124	0,7541	104

Tablo 5'te nöron sayısının düşük olduğu 9 ve 14 numaralı işlerde başarımın diğerlerine kıyasla düşük olduğu görülmektedir. 256 ve 512 nöron sayısı ile yapılan eğitimlerde ise birbirine yakın ve daha yüksek doğruluk oranları elde edilmiştir. Şekil 13 ve 14'te farklı nöron sayıları ile yapılan eğitimlere ait doğruluk grafikleri verilmiştir.



Şekil 13: NASNetMobile modelinde nöron sayılarına göre doğruluk grafiği



Şekil 14: DenseNet201 modelinde nöron sayılarına göre doğruluk grafiği

Şekil 13'de verilen NASNetMobile grafiğinin daha istikrarlı iken Şekil 14'te verilen DenseNet201 modelinin doğruluk grafiğinin daha dalgalı olduğu görülmektedir. Bu durumun sebebi olarak model mimarilerini değil, veri setlerinin büyüklüklerini ele almak daha doğru olacaktır. Bu bağlamda nöron sayısından bağımsız olarak daha büyük veri setleriyle daha tutarlı

bir eğitim yapıldığı söylenebilir. Çalışmada bu aşamadan sonra modeller 256 nöron olarak tasarlanmış ve eğitimlere bu değerle devam edilmiştir.

4.2.3. Paket Boyutu

ESA modelleri eğitilirken veri setinin tamamını kullanarak tek seferde eğitim işini gerçekleştirmek cihaz performansı açısından mantıklı bir işlem değildir. Bunun yerine veri setini parçalara ayırarak eğitime vermek çalışmalarda tercih edilen bir yöntemdir. Bu çalışmada önceki aşamalarda paket boyutu 32 olarak belirlendiğinden, ikinci veri seti ile yapılan eğitimler her epoch 50 iterasyonda, birinci veri setiyle yapılan eğitimler ise her epoch için 250 iterasyonda gerçekleşmiştir. Paket boyutu 8, 16, 32 ve 64 olarak değiştirilerek her iki veri seti de belirlenen modellerle eğitimden geçirilerek sonuçları Tablo 6 ve 7’de verilmiştir.

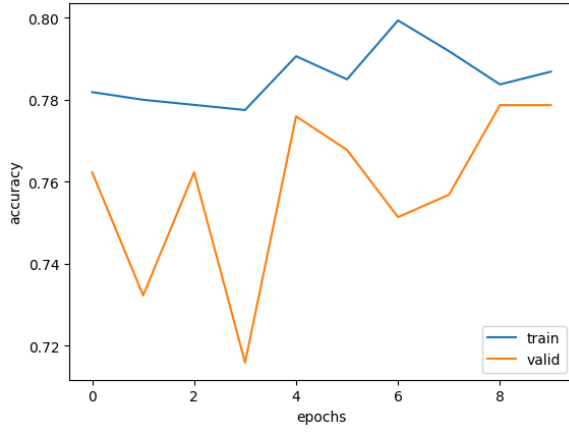
Tablo 6: Birinci veri seti ile yapılan paket boyutu eğitim sonuçları

No	Model	Epochs	Nöron Sayısı	Paket Boyutu	Loss	Doğruluk Değeri	Süre (Saniye)
19	NASNetMobile	25	256	8	0,5986	0,6825	1287
20	NASNetMobile	25	256	16	0,6342	0,625	1206
21	NASNetMobile	25	256	32	0,6532	0,6015	1083
22	NASNetMobile	25	256	64	0,6369	0,6355	1181

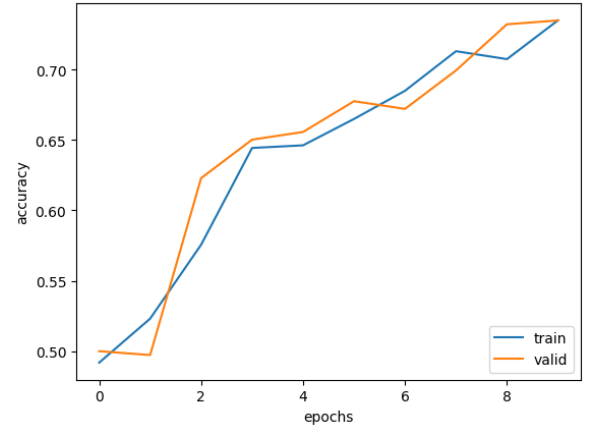
Tablo 7: İkinci veri seti ile yapılan paket boyutu eğitim sonuçları

No	Model	Epochs	Nöron Sayısı	Paket Boyutu	Loss	Doğruluk Değeri	Süre (Saniye)
23	DenseNet201	10	256	8	0,5017	0,7787	194
24	DenseNet201	10	256	16	0,5165	0,7568	178
25	DenseNet201	10	256	32	0,5205	0,7541	180
26	DenseNet201	10	256	64	0,5603	0,735	196

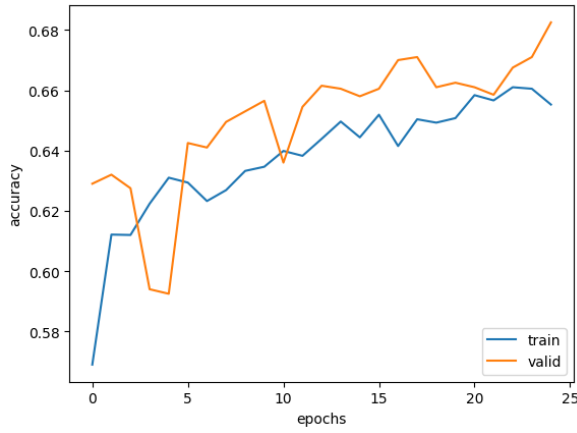
Tablo 6 ve 7 incelendiğinde en yüksek performansın paket boyutu 8 olan 19 ve 23 numaralı işlerde elde edildiği görülmektedir. Paket boyutu 8 olarak ayarlandığında, ikinci veri seti ile yapılan eğitimler her epoch 200 iterasyonda, birinci veri setiyle yapılan eğitimler ise her epoch için 1000 iterasyonda gerçekleşmiştir. Paket boyutunun küçülmesi ağına daha fazla sayıda eğitimden geçirilmesi anlamına gelmektedir. Bu durumun modelin performansında belli artış sağladığı gözlemlenmiştir. Ancak Tablo 6’ya bakıldığında paket boyutu küçüldükçe iterasyon sayısının artmasından dolayı eğitim süresinin de uzadığı görülmektedir. Daha büyük veri setleri ile yapılan çalışmalarda bu durumun dezavantaj oluşturabileceği de değerlendirilmelidir. Çalışmada kullanılan en düşük paket boyutu olan 8 değeri ile elde edilen sonuçlar Şekil 15 ve 17’de, en yüksek paket boyutu olan 64 değeri ile yapılan eğitimlerin doğruluk grafikleri ise Şekil 16 ve 18’de verilmiştir.



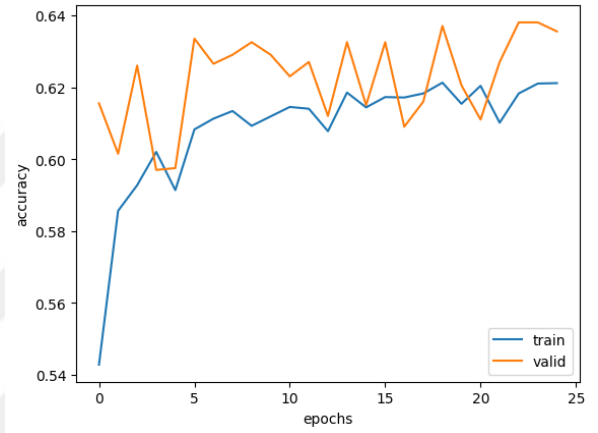
Şekil 15: 23 no'lu iş eğitim ve test doğruluk grafiği



Şekil 16: 26 no'lu iş eğitim ve test doğruluk grafiği



Şekil 17: 19 no'lu iş eğitim ve test doğruluk grafiği



Şekil 18: 22 no'lu iş eğitim ve test doğruluk grafiği

Şekil 15'de görüldüğü üzere eğitim doğruluk oranının daha tutarlı, test doğruluk oranının ise dalgalı bir grafik çizmesi paket boyutunun küçük olmasına bağlanabilir. Bu durum test için verilen paket boyutu küçük olduğunda ayırt edilebilirliği yüksek veya düşük olan örnek sayılarının dengesiz dağılım ihtimalinin yüksek olmasıyla açıklanabilir. Şekil 16'da paket boyutu 64 olan aynı veri setiyle yapılan işin sonuçlarına bakıldığında ise eğitim ve test doğruluklarının birbirine daha yakın ve tutarlı bir artış içinde olduğu görülebilir.

Veri setinin daha büyük ve eğitim tur sayısının daha fazla olduğu 19 ve 22 no'lu işlerin her ikisinde de test doğruluk grafiklerinde dalgalanmanın fazla olduğu gözlemlenmiştir. Sonuç olarak her iki veri seti ve modelde en iyi performansın elde edildiği paket boyutu 8 olduğundan çalışmada bu aşamadan sonraki modellerin eğitiminde paket sayısı 8 olarak kullanılmıştır.

4.2.4. Aktivasyon Fonksiyonları

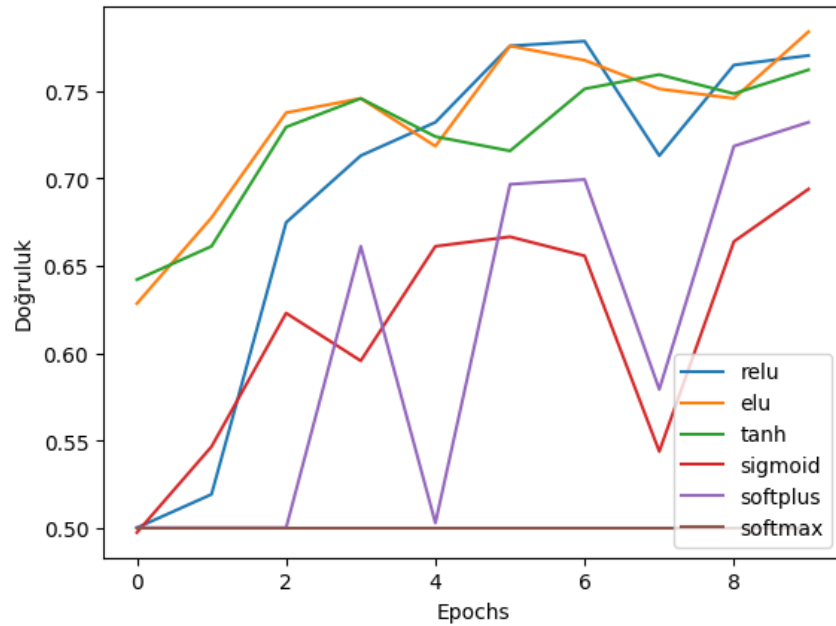
Aktivasyon fonksiyonları derin öğrenme mimarilerinde ağa gelen bilgileri işleyerek en uygun çıktının elde edilmesinde önemli bir role sahiptir. Bu bağlamda aktivasyon fonksiyonunun seçimi de önem arz etmektedir. Çalışmada kullanılan modellerde başlangıçta aktivasyon fonksiyonu olarak ara katmanlarda ReLU, çıkışta ise Softmax aktivasyon fonksiyonu

kullanılmıştır. Ara katmanlarda ise altı farklı aktivasyon fonksiyonu kullanılarak eğitim işleri yapılmış ve sonuçların doğruluk oranları Tablo 8 ve 9’da verilmiştir.

Tablo 8: Aktivasyon fonksiyonlarının sonuç performansları (İkinci veri seti)

No	Model	Epochs	Nöron Sayısı	Paket Boyutu	Aktivasyon Fonksiyonu	Doğruluk Değeri	Süre (Saniye)
27	DenseNet201	10	256	8	ELU	0.7842	139.74
28	DenseNet201	10	256	8	ReLU	0.7705	142.23
29	DenseNet201	10	256	8	Tanh	0.7623	129.66
30	DenseNet201	10	256	8	Softplus	0.7322	131.92
31	DenseNet201	10	256	8	Sigmoid	0.6940	137.09
32	DenseNet201	10	256	8	Softmax	0.5000	140.85

Tablo 8’de görüldüğü üzere ikinci veri seti ve DenseNet201 modeli ile en iyi doğruluk oranı ELU aktivasyon fonksiyonu ile elde edilmiştir. Başlangıç olarak belirlediğimiz ve çalışmalarda en çok kullanılan ReLU aktivasyonu negatif değerlerin çıktılarını sıfır olarak verir. ELU ise negatif değerlerde sıfırdan farklı çıktı verdiği için ReLU’ya göre daha yumuşak geçiş sağlar ve ReLU’ya göre aşırı uydurma konusunda daha avantajlıdır denilebilir. Şekil 19’da DenseNet201 modelinde kullanılan aktivasyon fonksiyonlarının doğruluk grafikleri verilmiştir.



Şekil 19: DenseNet201 modelinde aktivasyon fonksiyonlarının doğruluk grafiği

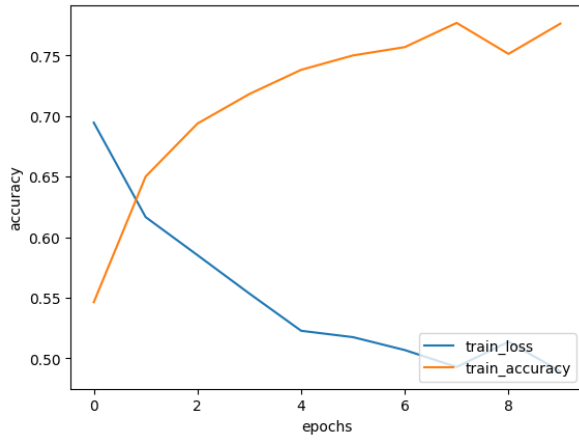
ELU ve Tanh aktivasyon fonksiyonlarıyla daha az dalgalanma ile öğrenme süreci gerçekleştirildiği Şekil 19 üzerindeki grafikte de gözlemlenebilir.

NASNetMobile modeli üzerinde farklı aktivasyon fonksiyonları ile yapılan eğitimlerin sonuçları Tablo 9’da verilmiştir.

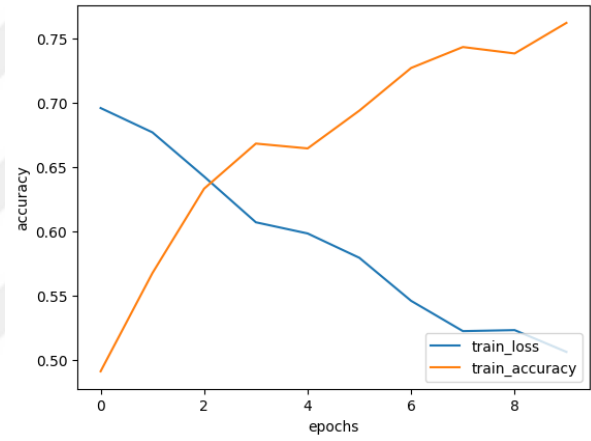
Tablo 9: Aktivasyon fonksiyonlarının sonuç performansları (Birinci veri seti)

No	Model	Epochs	Nöron Sayısı	Paket Boyutu	Aktivasyon Fonksiyonu	Doğruluk Değeri	Süre (Saniye)
33	NASNetMobile	25	256	8	Softplus	0.6480	1018
34	NASNetMobile	25	256	8	ReLU	0.6480	1127
35	NASNetMobile	25	256	8	Sigmoid	0.6365	1024
36	NASNetMobile	25	256	8	ELU	0.6070	1129
37	NASNetMobile	25	256	8	Tanh	0.5000	1078
38	NASNetMobile	25	256	8	Softmax	0.5000	1013

Tablo 9’da görüldüğü gibi bu eğitimlerde en performansı gösteren Softplus ve ReLU aktivasyon fonksiyonları ile yapılan eğitimlerin doğruluk grafikleri ise Şekil 20 ve 21’de verilmiştir.

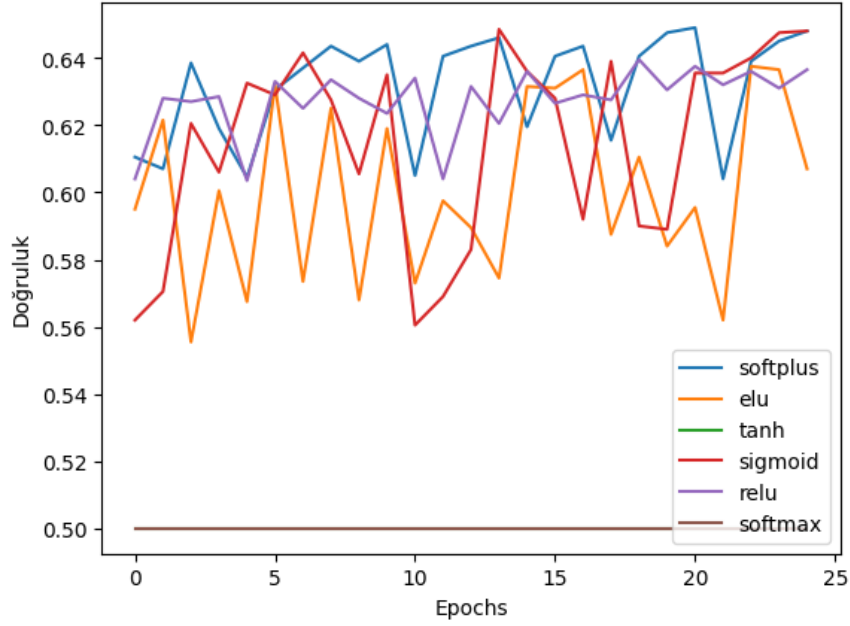


Şekil 20: 33 no'lu işe ait doğruluk ve kayıp değerleri grafiği



Şekil 21: 34 no'lu işe ait doğruluk ve kayıp değerleri grafiği

NASNetMobile modelinde birinci veri seti ile yapılan eğitimlerde Softplus ve ReLU aktivasyon fonksiyonları ile aynı doğruluk oranları elde edilse de Şekil 20 ve 21 incelendiğinde doğruluk ve kayıp değerleri grafiklerinin farklı olduğu görülmektedir. Softplus aktivasyon fonksiyonu yumuşak geçişlerle en üst noktaya gelirken, ReLU aktivasyon fonksiyonunun hem doğruluk hem kayıp değerlerinde daha sert düşüşler ve yükselişler olduğu görülür. Çalışmada test edilen tüm aktivasyon fonksiyonlarına ait grafikler ise Şekil 22’de verilmiştir.



Şekil 22: NASNetMobile modelinde aktivasyon fonksiyonlarının doğruluk grafiği

Şekil 22 incelendiğinde Softplus ve ReLU aktivasyon fonksiyonların daha tutarlı bir eğitim süreci olduğunu, diğer aktivasyon fonksiyonlarının daha dalgalı bir grafiği olduğunu görebiliriz. Softplus aktivasyon fonksiyonunda giriş değerlerinin logaritmaları alınarak ağırlıklar hesaplanır. Softplus, ReLU'nun aksine negatif değerli girişlere karşı duyarsız değildir ve pozitif değerler üretir (Sun ve ark., 2020). Bu özelliği ile Softplus aktivasyon fonksiyonunun giriş değerleri pozitif olan durumlarda daha tutarlı çalışması öngörülebilir.

ReLU aktivasyon fonksiyonu ile iki model için de iyi sonuçlar elde edilse de genelleme yapabilmek için daha fazla model ve veri seti ile test edilmesi gereklidir. Aktivasyon fonksiyonlarının özellikleri dikkate alınarak farklı fonksiyonları deneyip, veri setine ve modele en uygun olanına karar verilmelidir.

4.2.5. Optimizasyon Algoritmaları

Optimizasyon algoritmaları sinir ağlarının eğitiminde önemli rol oynayan parametrelerden biridir. Bu çalışmada Adadelta, Adagrad, Adam, Adamax, Nadam, SGD ve RMSprop olmak üzere yedi farklı optimizasyon algoritmalarıyla model eğitimleri yapılmış sınıflandırma performansları incelenmiştir. Farklı optimizasyon yöntemleriyle yapılan eğitimlerin sonuçları Tablo 10 ve 11'de verilmiştir.

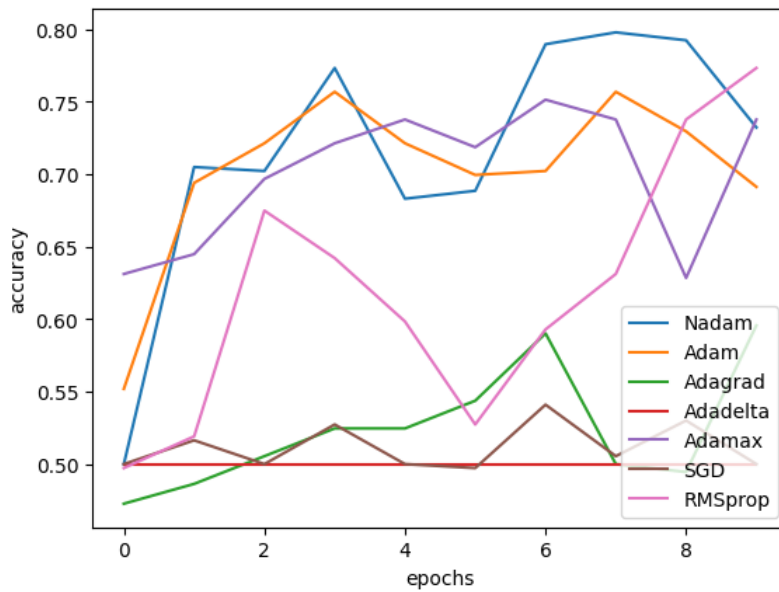
Tablo 10: Optimizasyon fonksiyonlarının sonuç performansları (İkinci veri seti)

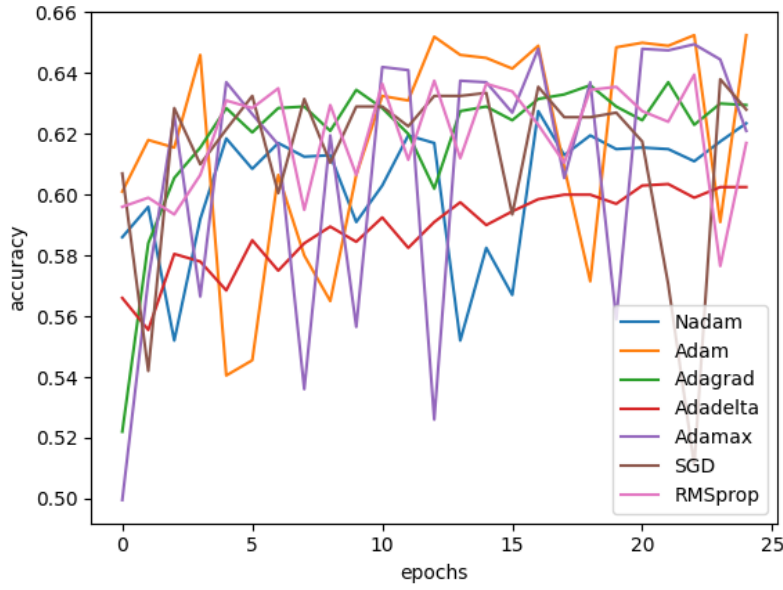
No	Model	Epochs	Nöron Sayısı	Paket Boyutu	Aktivasyon Fonksiyonu	Optimizasyon Fonksiyonu	Doğruluk Değeri	Süre (Sn.)
39	DenseNet201	10	256	8	ELU	RMSprop	0.7732	127.90
40	DenseNet201	10	256	8	ELU	Adamax	0.7377	116.71
41	DenseNet201	10	256	8	ELU	Nadam	0.7322	128.04
42	DenseNet201	10	256	8	ELU	Adam	0.7319	116.34
43	DenseNet201	10	256	8	ELU	Adagrad	0.5956	115.91
44	DenseNet201	10	256	8	ELU	Adadelata	0.5000	115.12
45	DenseNet201	10	256	8	ELU	SGD	0.5000	116.27

Tablo 11: Optimizasyon fonksiyonlarının sonuç performansları (Birinci veri seti)

No	Model	Epochs	Nöron Sayısı	Paket Boyutu	Aktivasyon Fonksiyonu	Optimizasyon Fonksiyonu	Doğruluk Değeri	Süre (Sn.)
46	NASNetMobile	25	256	8	Softplus	Adam	0.6525	1029.47
47	NASNetMobile	25	256	8	Softplus	Adagrad	0.6295	1026.14
48	NASNetMobile	25	256	8	Softplus	SGD	0.6280	1034.05
49	NASNetMobile	25	256	8	Softplus	Nadam	0.6235	1041.37
50	NASNetMobile	25	256	8	Softplus	Adamax	0.6210	1025.61
51	NASNetMobile	25	256	8	Softplus	RMSprop	0.6170	1034.03
52	NASNetMobile	25	256	8	Softplus	Adadelata	0.6025	1025.97

Tablo 10 ve 11’de görüldüğü gibi DenseNet201 modeliyle yapılan eğitimlerde RMSprop algoritması ile, NASNetMobile modeliyle yapılan eğitimlerde ise Adam algoritması ile en yüksek doğruluk değerleri elde edilmiştir. Tablo 10’da verilen 44 ve 45 numaralı işlerde SGD ve Adadelata algoritmaları ile ilgili modelde öğrenme gerçekleşmemiştir.

**Şekil 23:** DenseNet201 modelinde optimizasyon algoritmalarının doğruluk grafiği



Şekil 24: NASNetMobile modelinde optimizasyon algoritmalarının doğruluk grafiği

Şekil 24'e bakıldığında RMSprop algoritması daha tutarlı bir grafik çizerken, Adam algoritmasının daha dalgalı bir eğitim süreci olduğu görülmektedir. RMSprop algoritması, gradyanların karelerinin hareketli ortalamasını kullanır. Gradyanların büyük değerler alması durumunda öğrenme hızını otomatik olarak düşürerek daha kararlı bir eğitim sağlaması öngörülür (Ruder, 2016). Daha dalgalı bir yapıda olsa da NASNetMobile modelinde Adam algoritması daha yüksek doğruluk oranı sağlamıştır.

Adam algoritması gradyanların hareketli ortalamalarını hesaplayarak ağırlıkların güncellenmesini sağlar. Adam aslında momentum ve RMSprop yöntemlerinin güçlü yönlerini birleştirerek oluşturulan bir algoritmadır (Ruder, 2016). Bu özellikleri sayesinde Adam algoritması oldukça popüler bir optimizasyon yöntemidir ve birçok çalışmada da başarılı sonuçlar elde edilmiştir. Ancak çalışmamızda yapılan eğitim işlerinin sonuçlarında da görüldüğü üzere farklı modeller ve veri setleri üzerinde en iyi performansın elde edildiği optimizasyon algoritmaları değişiklik göstermektedir. Dolayısıyla herhangi bir optimizasyon algoritması hakkında genelleme yapılamamıştır. Veri setlerine ve modellere en uygun optimizasyon yöntemini bulmak için farklı algoritmaları denemek önerilir.

4.2.6 Öğrenme Oranları

Öğrenme oranı, ağırlık güncellemelerinin ne kadar büyüklükte adımlarla yapılacağını belirler ve 0 ile 1 arasında değer alır. Çok büyük öğrenme oranlarının ağırlık istikrarsız bir eğitim süreci yaşamasına, çok küçük öğrenme oranlarının ise eğitim sürecinin yavaşlamasına sebep olacağı ve donanım ihtiyacını artıracığı bilinmektedir (Dikici, 2021). Bu durumlardan

kaçınmak için çalışmamızda 0,01 ile 0,0001 arasında öğrenme oranları uygulanmıştır. DenseNet201 modelinde en yüksek başarıyı gösteren RMSprop ve Adamax optimizasyon algoritmalarında belirlenen öğrenme oranları test edilerek sonuçlar Tablo 12’de verilmiştir.

Tablo 12: RMSprop ve Adamax algoritmalarının öğrenme oranlarına göre sonuçları

No	Optimizasyon	Öğrenme Oranı	Doğruluk Değeri	Süre (Sn)	No	Optimizasyon	Öğrenme Oranı	Doğruluk Değeri	Süre (Sn)
53	RMSprop	0.01	0,5027	141	56	Adamax	0.01	0,7213	115
54	RMSprop	0.001	0,7787	132	57	Adamax	0.001	0,7322	106
55	RMSprop	0.0001	0,6749	131	58	Adamax	0.0001	0,6913	106

Tablo 12’de görüldüğü gibi her iki algortmada da en iyi performansa 0,001 öğrenme oranları ile ulaşılmıştır. NASNetMobile modelinde ise en yüksek başarıya sahip Adam ve Adagrad optimizasyon algoritmaları ile öğrenme oranları test edilerek sonuçlar Tablo 13 ‘de verilmiştir.

Tablo 13: Adam ve Adagrad algoritmalarının öğrenme oranlarına göre sonuçları

No	Optimizasyon	Öğrenme Oranı	Doğruluk Değeri	Süre (Sn)	No	Optimizasyon	Öğrenme Oranı	Doğruluk Değeri	Süre (Sn)
59	Adam	0.01	0,5	967	62	Adagrad	0.01	0,6275	1020
60	Adam	0.001	0,6615	971	63	Adagrad	0.001	0,6385	1064
61	Adam	0.0001	0,638	971	64	Adagrad	0.0001	0,5985	1005

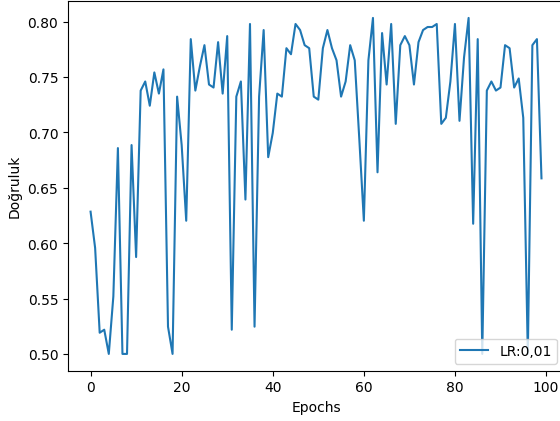
Tablo 13’te görüldüğü gibi Adam ve Adagrad algoritmaları ile en iyi performans 0,001 öğrenme oranı ile elde edilmiştir. Farklı öğrenme oranları kullanılan dört optimizasyon yönteminde de en iyi sonuçlar 0,001 öğrenme oranı ile elde edilmiştir. Bu sonuçların ardından öğrenme oranlarının etkisini daha iyi gözlemleyebilmek için modellerde belirlenen en iyi optimizasyon algoritmaları ve aynı parametrelerle 100 epoch’ta tekrar eğitilmiştir. Bu eğitimlerden sonra elde edilen sonuçlar Tablo 14’te yer almaktadır.

Tablo 14: RMSprop ve Adam algoritmalarının öğrenme oranlarına göre sonuçları

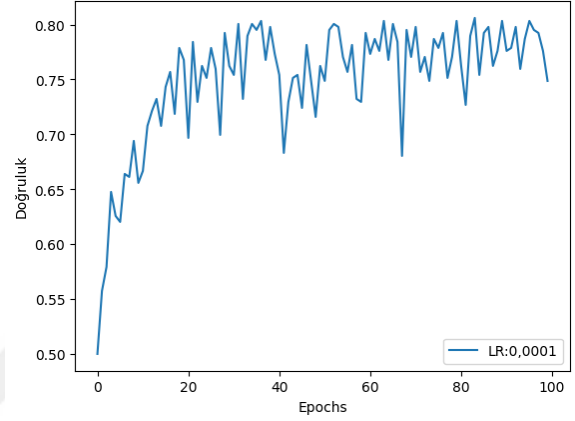
No	Model	Veri Seti	Epochs	Optimizasyon Fonksiyonu	Öğrenme Oranı	Doğruluk Değeri	Süre (Sn.)
65	DenseNet201	pizza classification data	100	RMSprop	0.01	0,6585	1251
66	DenseNet201	pizza classification data	100	RMSprop	0.001	0,7860	1269
67	DenseNet201	pizza classification data	100	RMSprop	0.0001	0.7486	1249
68	NASNetMobile	Dogs & Cats Images	100	Adam	0.01	0,6	1028
69	NASNetMobile	Dogs & Cats Images	100	Adam	0.001	0,6286	1033
70	NASNetMobile	Dogs & Cats Images	100	Adam	0.0001	0,6786	1011

Tablo 14’te görüldüğü gibi 100 epoch ile yapılan eğitimlerde RMSprop algoritması en yüksek doğruluk performansını yine 0,001 öğrenme oranı ile elde etmiştir. Adam algoritmasında ise 100 epoch ile yapılan eğitimlerde önceki işlere göre daha düşük öğrenme oranında daha iyi

sonuç verdiği gözlemlenmiştir. 54 ve 66 numaralı işlerin sonuçları incelendiğinde öğrenme oranı aynı olmasına rağmen 66 numaralı işin 100 epoch ile eğitilmesinin etkisiyle doğruluk oranının bir miktar yükseldiği görülmektedir. RMSprop algoritmasının yüksek öğrenme oranı ile yapılan eğitiminin doğruluk grafiği Şekil 25'te, düşük öğrenme oranı ile yapılan eğitiminin doğruluk grafiği ise Şekil 26'da verilmiştir.



Şekil 25: RMSprop 0,01 öğrenme oranı doğruluk grafiği



Şekil 26: RMSprop 0,0001 öğrenme oranı doğruluk grafiği

Şekil 25 ve 26'da verilen doğruluk/epoch grafiklerine bakıldığında öğrenme oranı büyük olduğunda eğitimin daha dalgalı bir yapıda ilerlediği, öğrenme oranı düştükçe dalgalanmanın azaldığı daha tutarlı bir yapıya büründüğü görülebilir.

Özetle bu çalışmada, Keras kütüphanesinde yer alan 27 ESA modeli arasından seçilen NASNetMobile modeli birinci veri seti, DenseNet201 modeli ikinci veri seti kullanılarak farklı parametre değerleriyle eğitilmiş ve farklı sonuçlar elde edilmiştir. Bunlar:

Birinci veri seti ve NASNetMobile modeliyle 10 epoch sayısında, ikinci veri seti ve DenseNet201 modeliyle 25 epoch sayısında en iyi performansa ulaşılmıştır. Nöron sayısı 32, 64, 128, 256 ve 512 olarak denenmiş, her iki model ve veri seti için en iyi performans değerleri 256 nöron sayısı ile elde edilmiştir. Paket boyutu parametresi 8, 16, 32 ve 64 değerleri ile değiştirilerek her iki model ve veri setinde paket boyutu 8 olarak ayarlandığında en iyi performans değerlerine ulaşılmıştır. Ara katmanlarda ELU, ReLU, Tanh, Softplus, Sigmoid, Softmax aktivasyon fonksiyonları ile yapılan eğitimlerde en iyi performans değerleri DenseNet201 modelinde ELU aktivasyon fonksiyonu, NASNetMobile modelinde ise Softplus aktivasyon fonksiyonu ile elde edilmiştir. Optimizasyonda Adadelta, Adagrad, Adam, Adamax, Nadam, SGD ve RMSprop olmak üzere yedi farklı algoritma ile çalışılmış ve DenseNet201 modelinde RMSprop, NASNetMobile modelinde Adam algoritması en iyi performansı

sağlayan algoritmalar olmuştur. Elde edilen en iyi optimizasyon algoritmalarının öğrenme oranlarında değişiklik yapılarak 0,01, 0,001, 0,0001 öğrenme oranlarıyla yeniden eğitilmiş ve iki algoritma için de en yüksek doğruluk değerleri 0,001 öğrenme oranıyla sağlanmıştır.



5. SONUÇLAR

Derin öğrenme yöntemleriyle görüntü işleme ve örüntü tanımada ESA'lar önemli bir yer tutmaktadır. Görüntü verileri üzerinde örüntü tanıma, sınıflandırma ve öznitelik çıkarma gibi görevleri başarıyla gerçekleştiren ESA, nesne tespiti alanında kaydedilen ilerlemelerde büyük rol almaktadır. Bu tez çalışmasının amacı, ESA modellerindeki hiperparametreleri değiştirerek nesne tespiti uygulamalarında sınıflandırma performansına nasıl etki ettiğini gözlemlemektir. Bu etkileri daha net görebilmek adına farklı iki veri seti ile NASNetMobile ve DenseNet201 ESA modelleri ile parametreler değiştirilerek 65 farklı eğitim gerçekleştirilmiştir. Yapılan eğitimler sonucunda elde edilen bulgular tablolar ve grafiklerle görselleştirilerek sunulmuştur. Belirlenen ESA modellerinde eğitim tur sayısı, nöron sayısı, paket boyutu, aktivasyon fonksiyonları, optimizasyon algoritmaları ve öğrenme oranları parametrelerinin değerleri değiştirilerek modeller eğitilmiş ve ortaya çıkan sonuçlar incelenmiştir.

Yapılan eğitimlerde ideal eğitim tur sayısının modele ve veri setine göre farklı değerlerle ulaşılabileceği görülmüştür. Çalışmada kullanılan veri setleri ele alındığında, büyük veri setlerini daha yüksek epoch değerleri eğitmek gerektiği söylenebilir. Nöron sayılarıyla ilgili yapılan deneylerde ise her iki modelde de nöron sayısını artırmanın ağırlık öğrenme kapasitesini, dolayısıyla sınıflandırma performansını artırdığı görülmüştür. Paket boyutu ile ilgili yapılan ayarlamalarda ise her iki model için de paket boyutu azaldıkça doğruluk oranının arttığı gözlemlenmiştir. Ancak paket boyutunun küçük olduğunda iterasyon sayısı büyük olacağından eğitim süresinin de uzayacağı değerlendirilmelidir. Aktivasyon fonksiyonları ve optimizasyon algoritmaları ile yapılan deneyler sonucunda modellere veya veri setine göre bir nitelendirme yapılamamıştır. Bu parametrelerin özellikleri dikkate alınarak farklı değerleri deneyip, veri setine ve modele en uygun olanına karar verilmelidir. Son olarak öğrenme oranlarının büyük değerler aldığı eğitimde eğitimin daha istikrarsız bir grafik çizdiği, küçük öğrenme oranlarında daha tutarlı kararlı bir eğitim gerçekleştiği görülmüştür. Ancak küçük öğrenme oranlarının özellikle büyük veri setlerinde eğitimin yavaşlamasına sebep olabileceği de göz önünde bulundurulmalıdır.

Çalışmada ESA modelleri üzerinde yapılan parametre değişiklikleri sonucunda; NASNetMobile modelinde başlangıçta % 61,75 olan doğruluk oranı % 68,25'e yükselerek % 6,5'lik bir artış sağlanmıştır. DenseNet201 modelinde ise başlangıçta % 66,87 olan doğruluk oranı % 78,42'ye yükselerek % 11,55'lik bir artış sağlanmıştır. Bu bulgular hiperparametrelerin ESA modellerinin sınıflandırma performansı üzerinde etkili olduğunu gösterse de çalışmamızın bazı sınırlılıkları da mevcuttur. ESA modellerinde kullanılan hiperparametrelerin yalnızca bir

kısının deęerleri belirli aralıklarda seilerek incelenmiřtir. Tm hiperparametrelerin alabileceęi tm deęerlerden oluřan olasılık uzayının sadece bir blm iin deneyler yapılmıřtır. Hiperparametrelerin yanında modelde kullanılan veri seti de performans zerinde etkilidir. İki veri seti ile modellerin eęitilmesi alıřmadaki bir bařka sınırlılık olarak ifade edilebilir.

Ynetim Biliřim Sistemleri'nde (YBS) veriler artık yapay zeka uygulamalarıyla iřlenerek, iřletmenin ynetim, analiz ve karar verme mekanizmalarını desteklemekte kullanılmaktadır (Dnerark ve Tecim, 2020). Deneysel alıřmalarımızda yapılan uygulamaların iřletmenin ynetim ve analiz srelerinde uygulamaya koyulamamıř olması da tez alıřmamızın sınırlılıklardandır.

alıřmanın devamında, yapılan uygulamalar bir iřletmenin ynetim srelerine dahil edilerek gerek hayatta etkileri gzlenebilir. Farklı hiperparametrelerin farklı deęer aralıklarında ve daha fazla veri seti ile ESA modelleri zerinde deneyler yapılabilir. Ayrıca grnt iřleme alanında yeni ve etkileyici bir yaklařım olan Vision Transformers mimarilerinde yer alan parametrelerin incelenmesi de alıřmanın devamına eklenebilir.

6. KAYNAKÇA

- Adedaja, A. O. & Owolawi, P. A. & Mapayi, T. & Tu, C. (2022). Intelligent Mobile Plant Disease Diagnostic System Using NASNet-Mobile Deep Learning. *IAENG International Journal of Computer Science*, 49(1), 216-231.
- Adem, K. (2022). P+ FELU: Flexible and trainable fast exponential linear unit for deep learning architectures. *Neural Computing and Applications*, 34(24), 21729-21740.
- Adem, K. (2022). Impact of activation functions and number of layers on detection of exudates using circular Hough transform and convolutional neural networks. *Expert Systems with Applications*, 203, 117583.
- Adem, K., Ozguven, M. M., & Altas, Z. (2023). A sugar beet leaf disease classification method based on image processing and deep learning. *Multimedia Tools and Applications*, 82(8), 12577-12594.
- Adhinata, F. D. & Rakhmadani, D. P. & Wibowo, M. & Jayadi, A. (2021). A deep learning using DenseNet201 to detect masked or non-masked face. *JUITA: Jurnal Informatika*, 9(1), 115-121.
- Aktürk, S. & Serbest, K. (2022). Nesne Tespiti İçin Derin Öğrenme Kütüphanelerinin İncelenmesi. *Journal of Smart Systems Research*, 3(2), 97-119.
- Alzubaidi, L., Zhang, J., Humaidi, A. J., Al-Dujaili, A., Duan, Y., Al-Shamma, O., & Farhan, L. (2021). Review of deep learning: Concepts, CNN architectures, challenges, applications, future directions, *Journal of big Data*, 8, 1-74.
- Amit, Y. & Felzenszwalb, P. & Girshick, R. (2020). Object detection. *Computer Vision: A Reference Guide*, 1-9.
- Arslan, E. (2021). *Evrişimli sinir ağı özelliklerine dayanan korelasyon filtreleme ve veri ilişkilendirme ile çoklu nesne takibi*. (Yayımlanmamış Yüksek Lisans Tezi). Bursa Uludağ Üniversitesi, Fen Bilimleri Enstitüsü, Bursa.
- Aytuğ, O. (2020). Evrişimsel Sinir Ağı Mimarilerine Dayalı Türkçe Duygu Analizi. *Avrupa Bilim ve Teknoloji Dergisi*, 374-380.
- Bayram, F. (2020). Derin öğrenme tabanlı otomatik plaka tanıma. *Politeknik Dergisi*, 23(4), 955-960.

- Bengio, Y. (2012). Practical recommendations for gradient-based training of deep architectures. *Neural Networks: Tricks of the Trade: Second Edition*, 437-478.
- Bircanoğlu, C. & Arıca, N. (2018). Yapay Sinir Ağlarında Aktivasyon Fonksiyonlarının Karşılaştırılması. In *2018 26th signal processing and communications applications conference (SIU)* (pp. 1-4), IEEE.
- Brownlee, J. (2018). What is the Difference Between a Batch and an Epoch in a Neural Network. *Machine Learning Mastery*, 20.
- Daşgın, R. (2022). *Kitlesel Çevrimiçi Ders Platformlarında Kurslara Yapılan Yorumların Metin Madenciliği Kullanılarak Duygu Analizi*. (Yayımlanmamış Yüksek Lisans Tezi). Aksaray Üniversitesi, Sosyal Bilimler Enstitüsü, Aksaray.
- Daşgın, A. (2023). *Covid19 Yayılımını Azaltmak İçin Yüz Maskesinin Evrimsel Sinir Ağı Modelleri İle Tespiti*. (Yayımlanmamış Yüksek Lisans Tezi). Aksaray Üniversitesi, Sosyal Bilimler Enstitüsü, Aksaray.
- Demirel, B. (2023). *Object Detection with Minimal Supervision*. (Yayımlanmamış Yüksek Lisans Tezi). Orta Doğu Teknik Üniversitesi, Fen Bilimleri Enstitüsü, Ankara.
- Deng, L. & Yu, D. (2014). Deep learning: methods and applications. *Foundations and trends® in signal processing*, 7(3-4), 197-387.
- Dikici, B. (2021). *Object Detection Using Convolutional Neural Network*. (Yayımlanmamış Yüksek Lisans Tezi). Gaziantep Üniversitesi, Elektrik ve Elektronik Mühendisliği, Gaziantep.
- Doğan, C. (2021). *İstatistiksel ve Makine Öğrenme ile Derin Sinir Ağlarında Hiper-Parametre Seçimi İçin Melez Yaklaşım*. (Yayımlanmamış Yüksek Lisans Tezi). Hacettepe Üniversitesi, Fen Bilimleri Enstitüsü, Ankara.
- Dönerçark, M. & Tecim, V. (2020). Kurumsal Karar Destek Sistemlerinde Yapay Zekâ Kullanımı: Tasarım Ve Uygulama. *Yönetim Bilişim Sistemleri Dergisi*, 6(2), 77-103.
- Duman, E. & Akın, F. (2019). VGG16-CNN Parametrelerinin Model Başarısına Etkilerinin Değerlendirilmesi, *Mardin Artuklu Uluslararası Multidisipliner Çalışmalar Kongresi*, Mardin, 19-21 Nisan.
- Dumoulin, V. & Visin, F. (2016). A guide to convolution arithmetic for deep learning. *CoRR*, vol. abs/1603.07285.

- Feurer, M. & Hutter, F. (2019). Hyperparameter optimization. *Automated machine learning: Methods, systems, challenges*, 3-33.
- Gülcü, A. & Kuş Z. (2019). Konvolüsyonel sinir ağlarında hiper-parametre optimizasyonu yöntemlerinin incelenmesi. *Gazi University Journal of Science Part C: Design and Technology*, 7(2), 503-522.
- Hao, X. & Zhang, G. & Ma, S. (2016). Deep learning. *International Journal of Semantic Computing*, 10(03), 417-439.
- Hastomo, W. & Karno, A. S. B. & Kalbuana, N. & Meiriki, A. (2021). Characteristic parameters of epoch deep learning to predict Covid-19 data in Indonesia. *Journal of Physics: Conference Series*, 1933(1), 1-4.
- Hekim, M., Cömert, O., & Adem, K. (2020). A hybrid model based on the convolutional neural network model and artificial bee colony or particle swarm optimization-based iterative thresholding for the detection of bruised apples. *Turkish Journal of Electrical Engineering and Computer Sciences*, 28(1), 61-79.
- Kaggle (2018). “Dogs & Cats Images”, <https://www.kaggle.com/datasets/chetankv/dogs-cats-images>, (Erişim tarihi: 17.03.2023).
- Kaggle (2014). “pizza classification data”, <https://www.kaggle.com/datasets/projectshs/pizza-classification-data>, (Erişim Tarihi: 01.03.2023).
- Kızrak, A. (2018). “Şu Kara Kutuyu Açalım: Yapay Sinir Ağları”, <https://ayyucekizrak.medium.com/şu-kara-kutuyu-açalım-yapay-sinir-ağları-7b65c6a5264a>, (Erişim tarihi: 12.04.2023).
- Kurt, F. (2018). *Evrışimli Sinir Ağlarında Hiper Parametrelerin Etkisinin İncelenmesi*. (Yayımlanmamış Yüksek Lisans Tezi). Hacettepe Üniversitesi, Fen Bilimleri Enstitüsü, Ankara.
- İnik, Ö. & Ülker, E. (2017). *Derin Öğrenmede Kullanılan Veri Setleri ve Yazılım Kütüphaneleri*. ISMSIT 2017, Tokat, 2-4 November 2017.
- Kandel, I. & Castelli, M. (2020). The effect of batch size on the generalizability of the convolutional neural networks on a histopathology dataset. *ICT express*, 6(4), 312-315.
- Karakurt, M. & İşeri, İ. (2022). Patoloji Görüntülerinin Derin Öğrenme Yöntemleri İle Sınıflandırılması. *Avrupa Bilim ve Teknoloji Dergisi*, (33), 192-206.

- Kaya, Ü. & Oğuz, Y. & Şenol, Ü. (2018). An Assessment of Energy Production Capacity of Amasra Town Using Artificial Neural Networks. *Turkish Journal of Electromechanics and Energy*, 3(1), 2226.
- Keskenler, M. F. & Keskenler, E. F. (2017). Geçmişten günümüze yapay sinir ağları ve tarihçesi. *Takvim-i Vekayi*, 5(2), 8-18.
- Kılıçarslan, S., Adem, K. & Çelik, M. (2021). An overview of the activation functions used in deep learning algorithms. *Journal of New Results in Science*, 10(3), 75-88.
- Li, Z. & Liu, F. & Yang, W. & Peng, S. & Zhou, J. (2021). A survey of convolutional neural networks: analysis, applications, and prospects. *IEEE transactions on neural networks and learning systems*. 33(12), 6999-7019.
- Liashchynskyi, P. & Liashchynskyi, P. (2019). Grid search, random search, genetic algorithm: a big comparison for NAS. *arXiv preprint arXiv:1912.06059*, 1-11.
- Onan, A. (2020). Evrişimli sinir ağı mimarilerine dayalı Türkçe duygu analizi. *Avrupa Bilim ve Teknoloji Dergisi*, 374-380.
- Opengenus (2023). “Aktivasyon Fonksiyonları”, <https://iq.opengenus.org/activation-functions-ml/>, (Erişim tarihi: 03.05.2023).
- O'Shea, K. & Nash, R. (2015). An introduction to convolutional neural networks. *arXiv preprint arXiv:1511.08458*.
- Öztemel, E. (2003). *Yapay sinir ağları*. Papatya Yayıncılık, İstanbul.
- Öztürk, K. & Şahin, M. E. (2018). Yapay sinir ağları ve yapay zekâ'ya genel bir bakış. *Takvim-i Vekayi*, 6(2), 25-36.
- Ruder, S. (2016). “An Overview Of Gradient Descent Optimization Algorithms”, <https://www.ruder.io/optimizing-gradient-descent/>, (Erişim tarihi 16.04.2023).
- Salur, M. U. & Aydın, İ. & Karaköse, M. (2019). Gömülü derin öğrenme ile tehdit içeren nesnelerin gerçek zamanda tespiti. *Dicle Üniversitesi Mühendislik Fakültesi Mühendislik Dergisi*, 10(2), 497-509.
- Samtaş, G. & Gülesin, M. (2011). Sayısal görüntü işleme ve farklı alanlardaki uygulamaları. *Ejovoc (Electronic Journal of Vocational Colleges)*, 2(1), 85-97.
- Saxen, F. & Werner, P. & Handrich, S. & Othman, E. & Dinges, L. & Al-Hamadi, A. (2019). Face attribute detection with mobilenetv2 and nasnet-mobile. *International Symposium*

on Image and Signal Processing and Analysis (ISPA) 2019, Dubrovnik, 23-25 September 2019.

- Ser, G. & Bati, C. T. (2019). Derin sinir ağıları ile en iyi modelin belirlenmesi: mantar verileri üzerine Keras uygulaması. *Yuzuncu Yıl University Journal of Agricultural Sciences*, 29(3), 406-417.
- Seyyarer, E. & Ayata, F. & Uçkan, T. & Karci, A. (2020). Derin öğrenmede kullanılan optimizasyon algoritmalarının uygulanması ve kıyaslanması. *Computer Science*, 5(2), 90-98.
- Smith, L. N. (2017). Cyclical learning rates for training neural networks. In *2017 IEEE winter conference on applications of computer vision (WACV)*, 464-472.
- Smith, L. N. (2018). A disciplined approach to neural network hyper-parameters: Part 1-learning rate, batch size, momentum, and weight decay. *arXiv preprint arXiv:1803.09820*.
- Sun, K. & Yu, J. & Zhang, L. & Dong, Z. (2020). A convolutional neural network model based on improved Softplus activation function. *International Conference on Applications and Techniques in Cyber Intelligence ATCI 2019: Applications and Techniques in Cyber Intelligence*, 7(pp. 1326-1335).
- Sun, R. (2019). Optimization for deep learning: theory and algorithms. *arXiv preprint arXiv:1912.08957*.
- Şeker, A. & Diri, B. & Balık H. (2017). Derin öğrenme yöntemleri ve uygulamaları hakkında bir inceleme. *Gazi Mühendislik Bilimleri Dergisi*, 3(3), 47-64.
- Ünal, T. & Çiftçi, Ü. & Urgan, N. N. (2022). Bir Gizli Katmanlı Yapay Sinir Ağlarında Optimal Nöron Sayısının İncelenmesi. *Süleyman Demirel Üniversitesi Fen Edebiyat Fakültesi Fen Dergisi*, 17(2), 303-325.
- Yılmaz, E. K., Adem, K., Kılıçarslan, S., & Aydın, H. A. (2023). Classification of lemon quality using hybrid model based on Stacked AutoEncoder and convolutional neural network. *European Food Research and Technology*, 1-13.
- Yurdusev, A. A., Adem, K., & Hekim, M. (2023). Detection and classification of microcalcifications in mammograms images using difference filter and Yolov4 deep learning model. *Biomedical Signal Processing and Control*, 80, 104360.

- Toğaçar, M. & Ergen, B. & Sertkaya, M. E. (2019). Zatürre Hastalığının Derin Öğrenme Modeli ile Tespiti. *Firat University Journal of Engineering*, 31(1).
- Tuba, E. & Baçanin, N. & Strumberger, I. & Tuba, M. (2021). Convolutional neural networks hyperparameters tuning. *Artificial intelligence: theory and applications* (pp. 65-84).
- Uçar, A. & Bingöl, M. S. (2018). Derin öğrenmenin Caffe kullanılarak grafik işleme kartlarında değerlendirilmesi. *Dicle Üniversitesi Mühendislik Fakültesi Mühendislik Dergisi*, 9(1), 39-49.
- Wang, S. H. & Zhang, Y. D. (2020). DenseNet-201-based deep neural network with composite learning factor and precomputation for multiple sclerosis classification. *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)*, 16(2s), 1-19.
- Wiederhold, G. & McCarthy, J. (1992). Arthur Samuel: Pioneer in machine learning. *IBM Journal of Research and Development*, 36(3), 329-331.
- Wikipedia (2023). “Turing testi”, https://tr.wikipedia.org/wiki/Turing_testi, (Erişim tarihi: 24.05.2023).
- Zhang, A. & Lipton, Z. C. & Li, M. & Smola, A. J. (2021). Dive into deep learning. *arXiv preprint arXiv:2106.11342*.

Aksaray Üniversitesi Tez Değerlendirme Formu		
Öğrencinin Adı Soyadı:		EVET
Kapak		
1	Tez Başlığı tutanaktaki başlıkla aynı mı?	
2	Kapaktaki ay ve yıl savunmaya girilen tarihle tutarlı mı?	
3	Kapak format kılavuzdaki kapak formatına uygun mu?	
4	Kapakta yazılan tüm yazılar doğru olarak verilmiş mi?	
İçindekiler		
5	Sayfa numaraları tam verilmiş mi?	
6	Şekil, Çizelge vb. listeleri verilmiş mi? Sıralaması doğru mu?	
7	Özet, Abstract, Giriş, Sonuçlar vb. bölümler var mı?	
8	Yazım hataları kontrolü yapıldı mı?	
Giriş		
9	Hazırlanan tezin önemini anlatıyor mu?	
10	İkinci ve Üçüncü dereceden başlık içermemeli kuralına uyuldu mu?	
Özet/Abstract		
11	Kılavuza uygun mu?	
12	Ay ve yıl savunmaya girilen tarihle tutarlı mı?	
13	Özet; tek sayfa, tek aralık, tek paragraf kuralına uygun olarak yazıldı mı?	
14	Bilim kodu, sayfa adedi, anahtar kelimeler ve tez danışmanı yazıldı mı?	
Kaynakça		
15	Kaynakların tamamına metin içinde atıf yapıldı mı?	
16	Kaynak formatı Kılavuzdaki kaynak formatına uygun olarak hazırlanmış mı?	
17	Atıf formatı kılavuzdaki atıf formatına uygun mu?	
Atıf Yöntemi APA 6 ☐ CMS ☐ İSNAD ☐		
Genel Değerlendirme		
18	Etik Beyan açıklaması okundu, uyuldu ve imzalandı mı?	
19	Kabul/Onay sayfası kılavuzdaki formata uygun olarak düzenlenmiş mi?	
20	Kabul /Onay sayfasında belirtilen oy birliği/oy çokluğu seçeneklerinden uygun olanı savunmayla tutarlı olacak şekilde belirlenmiş mi?	
21	Sayfa kenar boşluklar ve sayfa numaraları kılavuzdaki formatına uygun mu?	
22	Paragraf boşlukları ve metin satır aralığı kılavuza uygun olacak şekilde düzenlenmiş mi?	
23	Başlık yazımları kılavuzdaki başlık formatlarına uygun mu?	
24	Yazı tipi ve boyutu kılavuzdaki yazı tipi ve boyutu formatına uygun mu?	
25	Şekil, Çizelge vb. açıklama ve numaralandırmaları kılavuzdaki formata uygun olarak yazılmış mı?	

Bu tezin tarafımdan “Tez yazım kuralları” okunarak dikkatlice hazırlanmış olduğunu ve doğabilecek her türlü olumsuzluktan sorumlu olacağımı kabul ederim.

Öğrencinin imzası