



T.C.
NECMETTİN ERBAKAN
ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ



**Evrşimsel Sinir Ağları ile Kelebek Türlerinin
Tespiti**

Mustafa Serhat ÖZDEMİR

YÜKSEK LİSANS TEZİ

Bilgisayar Mühendisliği Anabilim Dalı

**Temmuz-2024
KONYA
Her Hakkı Saklıdır**

TEZ KABUL VE ONAYI

Mustafa Serhat ÖZDEMİR tarafından hazırlanan “Evrşimsel Sinir Ağları ile Kelebek Türlerinin Tespiti” adlı tez çalışması 01/07/2024 tarihinde aşağıdaki jüri tarafından oy birliği ile Necmettin Erbakan Üniversitesi Fen Bilimleri Enstitüsü Bilgisayar Mühendisliği Anabilim Dalı’nda YÜKSEK LİSANS TEZİ olarak kabul edilmiştir.

Jüri Üyeleri

Başkan

Prof. Dr. Mustafa Servet KIRAN

Danışman

Doç. Dr. Hüseyin HAKLI

Üye

Dr. Öğr. Üyesi Murat KARAKOYUN

İmza

.....

.....

.....

Fen Bilimleri Enstitüsü Yönetim Kurulu’nun/.../20.. gün ve sayılı kararıyla onaylanmıştır.

Prof. Dr. Şerife Yurdağül KUMCU
FBE Müdürü

TEZ BİLDİRİMİ

Bu tezdeki bütün bilgilerin etik davranış ve akademik kurallar çerçevesinde elde edildiğini ve tez yazım kurallarına uygun olarak hazırlanan bu çalışmada bana ait olmayan her türlü ifade ve bilginin kaynağına eksiksiz atıf yapıldığını bildiririm.

DECLARATION PAGE

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

İmza

Mustafa Serhat ÖZDEMİR

Tarih: 01.07.2024

ÖZET

YÜKSEK LİSANS TEZİ

EVRIŞİMSEL SİNİR AĞLARI İLE KELEBEK TÜRLERİNİN TESPİTİ

Mustafa Serhat ÖZDEMİR

**Necmettin Erbakan Üniversitesi Fen Bilimleri Enstitüsü
Bilgisayar Mühendisliği Anabilim Dalı**

Danışman: Doç. Dr. Hüseyin HAKLI

2024, 57 Sayfa

Jüri

Doç. Dr. Hüseyin HAKLI

Prof. Dr. Mustafa Servet KIRAN

Dr. Öğr. Üyesi Murat KARAKOYUN

Kelebek türlerinin sınıflandırılması, ekosistemlerin sağlığını ve sürdürülebilirliğini korumak, çevresel değişiklikleri izlemek, bilimsel araştırmaları desteklemek ve koruma çabalarına yön vermek açısından kritik bir rol oynar. Bu nedenle, kelebeklerin doğru ve hızlı bir şekilde tanımlanması, doğa koruma ve çevre yönetimi çalışmalarının önemli bir parçasıdır. Bu bağlamda yapılan çalışmada Türkiye’de yaygın olarak görülen 49 farklı kelebek türüne ait 5624 görüntüden oluşan bir veri seti hazırlanmıştır. Veriler Train-Validation-Test Split yöntemiyle bölünmüş, ardından sekiz farklı Evrişimli Sinir Ağı kullanılarak öğrenme aktarımı yöntemi ile eğitilmiş, son olarak bu ağların doğruluk, kayıp, kesinlik, duyarlılık ve F1-skoru değerlendirme kriterleri karşılaştırılmıştır. İlk yöntemde ağların kendi içinde yer alan katmanlar kullanılmış sadece sonuna sınıflandırma katmanı eklenmiştir. İkinci yöntemde ise ağların kendi katmanlarına ek olarak yeni katmanlar eklenerek ağların öğrenmesinin iyileştirilmesi amaçlanmıştır. Bu bağlamda sekiz ağ incelendiğinde test görüntüleri üzerinde en başarılı sonuçları veren ağın katman eklemesiz yöntemde %91,09 oranı ile DenseNet201 ağında elde edilirken, katman eklenerek yapılan yöntemde bu oran %94,33 ile DenseNet201 ağı olmuştur. Ağ sonuçları incelendiğinde ikinci yöntem sonunda DenseNet201 test için ayrılan 247 görüntünün 233 adetinin sınıfını doğru tahmin etmeyi başarmıştır.

Anahtar Kelimeler: Derin Öğrenme, Evrişimli Sinir Ağları, Kelebek Türlerinin Sınıflandırılması, Makine Öğrenmesi, Öğrenme Aktarımı

ABSTRACT

MS THESIS

**CLASSIFICATION OF BUTTERFLY SPECIES WITH A MODEL BASED ON
CONVOLUTIONAL NEURAL NETWORKS**

Mustafa Serhat Özdemir

**THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCE OF
NECMETTİN ERBAKAN UNIVERSITY
THE DEGREE OF MASTER OF SCIENCE
IN COMPUTER ENGINEERING**

Advisor: Assoc. Prof. Dr. Hüseyin HAKLI

2024, 57 Pages

Jury

**Assoc. Prof. Dr. Hüseyin HAKLI
Prof. Dr. Mustafa Servet KIRAN
Dr. Lecturer Murat KARAKOYUN**

The classification of butterfly species plays a critical role in preserving the health and sustainability of ecosystems, monitoring environmental changes, supporting scientific research, and guiding conservation efforts. Therefore, the accurate and rapid identification of butterflies is an essential part of nature conservation and environmental management efforts. In this context, a study was conducted to prepare a dataset consisting of 5624 images of 49 different butterfly species commonly found in Turkey. The data was split using the Train-Validation-Test Split method, then trained using eight different Convolutional Neural Networks (CNNs) through transfer learning, and finally, these networks were compared based on accuracy, loss, precision, recall, and F1-score evaluation criteria. In the first method, only the classification layer was added to the existing layers of the networks. In the second method, new layers were added to the networks in addition to their existing layers to improve learning. In this context, when eight networks were examined, the network that produced the most successful results on test images was DenseNet201 with a rate of 91,09% in the method without added layers, while in the method with added layers, this rate was 94,33% with the DenseNet201 network. Upon examining the network results, it was found that the second method successfully predicted the class of 233 out of the 247 images set aside for testing.

Keywords: Butterfly Species Classification, Convolutional Neural Networks, Deep Learning, Machine Learning, Transfer Learning

ÖNSÖZ

Bu tez çalışmasında derin öğrenmenin alt dallarından biri olan Evrişimsel Sinir Ağlarının farklı mimarileri, hazırlanan veri seti üzerine uygulanmıştır. Uygulama sonucunda eğitilen ağların performansları karşılaştırmalı olarak incelenmiştir.

Tez çalışmasının fikrinin oluşmasında bana desteği olan sayın Prof. Dr. Musa DİKMENLİ hocama, kelebek fotoğraflarının teminini sağladığım doğa fotoğrafçılarına ve web sayfalarına, çalışmaların planlanmasında, araştırılmasında, yürütülmesinde ve oluşturulması aşamasında bana desteğini esirgemeyen danışman hocam sayın Doç. Dr. Hüseyin HAKLI'ya, yüksek lisans eğitim sürecim boyunca bana maddi ve manevi desteğini esirgemeyen aileme teşekkürlerimi sunarım.

Mustafa Serhat ÖZDEMİR
KONYA-2024

İÇİNDEKİLER

ÖZET	iv
ABSTRACT.....	v
ÖNSÖZ	vi
İÇİNDEKİLER.....	vii
ŞEKİLLER LİSTESİ	ix
ÇİZELGELER LİSTESİ	xi
SİMGELER VE KISALTMALAR.....	xii
1. GİRİŞ.....	1
1.1 Tez Çalışmasına Giriş.....	1
1.2 Tez Çalışmasının Literatüre Katkısı	3
1.3 Tez Çalışmasının Organizasyonu	4
2. KAYNAK ARAŞTIRMASI	5
2.1 Makine Öğrenmesi (ML) Yöntemleri Kullanılarak Yapılan Kelebek Sınıflandırma Çalışmaları	6
2.2 Yapay Sinir Ağları (ANN) Kullanılarak Yapılan Kelebek Sınıflandırma Çalışmaları	9
2.3 Aşırı Öğrenme Makineleri (ELM) Kullanılarak Yapılan Kelebek Sınıflandırma Çalışmaları	10
2.4 Derin Öğrenme Yöntemleri Kullanılarak Yapılan Kelebek Sınıflandırma Çalışmaları	11
3. MATERYAL VE YÖNTEM.....	16
3.1 Veri Setinin Oluşturulması	16
3.2. Evrişimli Sinir Ağları (CNN) ve Katman Yapısı.....	19
3.2.1. Giriş Katmanı.....	20
3.2.2. Evrişim Katmanı	21
3.2.3. Aktivasyon Fonksiyonu Katmanı (Activation layer):.....	23
3.2.4. Ortaklama Katmanı (Average Pooling):.....	24
3.2.5. Düzleştirme Katmanı (Flattening Katmanı):	25
3.2.6. Tam Bağlantı Katmanı (Fully Connected Layer):	25
3.2.7. Batch Normalizasyon Katmanı (Batch Normalization Layer):	25
3.2.8. Seyreltme Katmanı (Dropout Layer):	26
3.3. Evrişimli Sinir Ağı Mimarileri	26
3.3.1. VGG-16 (Visual Geometry Group-16) Mimarisi	26
3.3.2. VGG-19 (Visual Geometry Group-19) Mimarisi	27

3.3.3. Xception (Extreme Inception) Mimarisi.....	27
3.3.4. InceptionV3 (Inception Version 3) Mimarisi	28
3.3.5. NASNetLarge (Neural Architecture Search Network) Mimarisi	28
3.3.6. InceptionResNetV2 (Inception Residual Network Version 2) Mimarisi.....	28
3.3.7. ResNet152V2 (Residual Network 152 Version 2) Mimarisi.....	29
3.3.8. DenseNet201 (Dense Convolutional Network 201) Mimarisi	29
3.4 CNN’de Öğrenme Aktarımı (Transfer Learning) Yöntemi Nasıl Uygulanır?	31
3.5 Katman Eklemeden Yapılan Öğrenme Aktarımı Yönteminin Uygulanması	33
3.6 Katman Eklenerek Yapılan Öğrenme Aktarımı Yönteminin Uygulanması	35
4. ARAŞTIRMA SONUÇLARI VE TARTIŞMA.....	37
4.1. Uygulama Sonucunda Elde Edilen Sonuçlar ve Sonuçların Karşılaştırılması	37
4.1.1. Katman Eklemeden Yapılan Öğrenme Aktarımında Elde Edilen Sonuçlar .	37
4.1.2. Katman Eklenerek Yapılan Öğrenme Aktarımında Elde Edilen Sonuçlar..	42
4.2 Katman Eklemeden ve Katman Eklenerek Yapılan Öğrenme Aktarımı Yöntemlerinin Karşılaştırılması.....	46
5. SONUÇLAR VE ÖNERİLER.....	51
5.1 Sonuçlar	51
5.2 Öneriler	51
6. KAYNAKLAR	53

ŞEKİLLER LİSTESİ

Şekil 3.1. Veri Önışleme Adımları

Şekil 3.2. Veri Setinin Train-Validation-Test Split Yöntemi ile Ayrılması

Şekil 3.3. Basit Bir Evrişimli Sinir Ağında Modelin Katman Yapısı

Şekil 3.4. Örnek Bir Evrişim İşlemi

Şekil 3.5. Sıfır Piksel Ekleme İşlemi Örneđi

Şekil 3.6. Piksel Ekleme İşlemi Örneđi

Şekil 3.7. Aktivasyon Fonksiyonlarının Karşılaştırmalı Olarak Gösterimi

Şekil 3.8. Maksimum Ortaklama ve Ortalama Ortaklama İşleminin Gösterimi

Şekil 3.9. Düzleştirme İşlemi

Şekil 3.10. VGG-16 Ağının Katman Yapısı

Şekil 3.11. VGG-19 Ağının Katman Yapısı

Şekil 3.12. InceptionV3 Ağının Katman Yapısı

Şekil 3.13. DenseNet201 Ağının Katman Yapısı

Şekil 3.14. Öğrenme Aktarımı ve İnce Ayar

Şekil 3.15. Katman Eklemeden Yapılan Öğrenme Aktarımı Yönteminin CNN Ağ Yapısı

Şekil 3.16. Katman Eklenerek Yapılan Öğrenme Aktarımı Yönteminin CNN Ağ Yapısı

Şekil 4.1. Katman Eklemeden Yapılan Öğrenme Aktarımında Elde Edilen Eğitim Verileri Üzerindeki Başarı Grafiđi

Şekil 4.2. Katman Eklemeden Yapılan Öğrenme Aktarımında Elde Edilen Eğitim Verileri Üzerindeki Kayıp Grafiđi

Şekil 4.3. Katman Eklemeden Yapılan Öğrenme Aktarımında Elde Edilen Doğrulama Verileri Üzerindeki Başarı Grafiđi

Şekil 4.4. Katman Eklemeden Yapılan Öğrenme Aktarımında Elde Edilen Doğrulama Verileri Üzerindeki Kayıp Grafiđi

Şekil 4.5. DenseNet201'in Katman Eklemeden Öğrenme Aktarımı Yönteminde Elde Edilen Karmaşıklık Matrisi

Şekil 4.6. Katman Eklenecek Yapılan Öğrenme Aktarımında Elde Edilen Eğitim Verileri Üzerindeki Başarı Grafiği

Şekil 4.7. Katman Eklenecek Yapılan Öğrenme Aktarımında Elde Edilen Eğitim Verileri Üzerindeki Kayıp Grafiği

Şekil 4.8. Katman Eklenecek Yapılan Öğrenme Aktarımında Elde Edilen Doğrulama Verileri Üzerindeki Başarı Grafiği

Şekil 4.9. Katman Eklenecek Yapılan Öğrenme Aktarımında Elde Edilen Doğrulama Verileri Üzerindeki Kayıp Grafiği

Şekil 4.10. DenseNet201'in Katman Eklenecek Öğrenme Aktarımı Yönteminde Elde Edilen Karmaşıklık Matrisi

Şekil 4.11. Ağların Katmansız ve Katmanlı Yöntemle Elde Edilen Eğitim Verileri Üzerindeki Başarısının Karşılaştırılması

Şekil 4.12. Ağların Katmansız ve Katmanlı Yöntemle Elde Edilen Doğrulama Verileri Üzerindeki Başarısının Karşılaştırılması

Şekil 4.13. Ağların Katmansız ve Katmanlı Yöntemle Elde Edilen Test Doğruluğu Performanslarının Karşılaştırılması

Şekil 4.14. Ağların Katmansız ve Katmanlı Yöntemle Elde Edilen Kesinlik Performanslarının Karşılaştırılması

Şekil 4.15. Ağların Katmansız ve Katmanlı Yöntemle Elde Edilen Duyarlılık Performanslarının Karşılaştırılması

Şekil 4.16. Ağların Katmansız ve Katmanlı Yöntemle Elde Edilen F1-Skoru Performanslarının Karşılaştırılması

ÇİZELGELER LİSTESİ

Çizelge 3.1. Veri Setinde Bulunan Tür Adları ve Görüntü Sayıları

Çizelge 3.2. Çalışmada Kullanılan CNN Mimarileri ve ImageNet Üzerindeki Parametreleri

Çizelge 3.3. Katman Eklemeden Yapılan Öğrenme Aktarımında Kullanılan Hiper Parametreler

Çizelge 4.1. Katman Eklemeden Yapılan Öğrenme Aktarımı Yönteminde Elde Edilen Başarı ve Kayıp Değerleri

Çizelge 4.2. Katman Eklemeden Yapılan Öğrenme Aktarımı Yönteminde Ağlarda Elde Edilen Değerlendirme Kriterlerinin Karşılaştırılması

Çizelge 4.3. Katman Eklenerek Yapılan Öğrenme Aktarımı Yönteminde Elde Edilen Başarı ve Kayıp Değerleri

Çizelge 4.4. Katman Eklenerek Yapılan Öğrenme Aktarımı Yönteminde Ağlarda Elde Edilen Değerlendirme Kriterlerinin Karşılaştırılması

SİMGELER VE KISALTMALAR

Simgeler

-

Kısaltmalar

ANN:	Yapay Sinir Ağı (Artificial Neural Network)
BLS:	Branch Length Similarity
CBIR:	İçerik Tabanlı Görüntü Benzerliği (Content Based Image Retrieval)
CNN:	Evrışimsel Sinir Ağı (Convolutional Neural Network)
CPU:	Merkezi İşlem Birimi (Central Processing Unit)
DCNN:	Derin Evrışimsel Sinir Ağı (Deep Convolutional Neural Network)
ELM:	Aşırı Öğrenme Makinesi (Extreme Learning Machine)
GF:	Gabor Filtreleri (Gabor Filters)
GLCM:	Gri Seviyeli Eş-oluşum Matrisi (Gray-Level Co-Occurrence Matrix)
GPU:	Grafik İşlem Birimi (Graphics Processing Unit)
HoMSC:	Histograms of Multi-Scale Curvature
HSV:	Hue Saturation Intensity
ILSVRC:	ImageNet Büyük Ölçekli Görsel Tanıma Yarışması (ImageNet Large Scale Visual Recognition Challenge)
KNN:	K-En Yakın Komşu (K-Nearest Neighbors)
LBP:	Yerel İkili Desenler (Local Binary Pattern)
MLR:	Çok Değişkenli Lojistik Regresyon (Multivariable Logistic Regression)
RS:	Kaba Küme (Rough Set)
SIFT:	Ölçekle Değişmez Unsur Dönüşümü (Scale Invariant Feature Transform)
SVM:	Destek Vektör Makineleri (Support Vector Machine)
TPU:	Tensor İşleme Birimi (Tensor Processing Unit)
UBN:	Unit Branching Network

1. GİRİŞ

1.1 Tez Çalışmasına Giriş

Günümüzde, görüntü işleme ve makine öğrenmesi tekniklerinin gelişmesiyle birlikte, doğal yaşamın çeşitli yönlerini incelemek ve anlamak için bilgisayar destekli yöntemlerin kullanımı önemli ölçüde artmıştır. Bu tekniklere bir örnek, Evrimsel Sinir Ağları (Convolutional Neural Network- CNN) tarafından gerçekleştirilen görüntü sınıflandırma ve tanıma işlemidir. Kelebekler, doğanın renkli ve zarif canlıları olarak biyoçeşitlilikte önemli bir yere sahiptir. Binlerce farklı türüyle dünyanın dört bir yanına yayılmışlardır. Hem bilim insanları hem de doğa fotoğrafçıları için büyük bir ilgi odağı olmuşlardır. Ancak, kelebeklerin tanımlanması ve sınıflandırılması, geleneksel olarak morfolojik özelliklerinin incelenmesi, uzun ve zahmetli bir süreci beraberinde getirmiştir. Kelebeklerin tür tespitinde, kanat özellikleri, renk, desen, şekil gibi görsel faktörler dikkate alınmaktadır. Ancak, bu yöntemlerin doğruluğu, maliyeti ve zaman alıcılığı gibi farklı sorunlarla karşılaşmaktadır. Son yıllarda, teknolojinin gelişimi ile kelebeklerin otomatik tanımlanması ve sınıflandırılması için yeni yaklaşımlar geliştirilmiştir.

Yapay zekâ, makine öğrenmesi ve derin öğrenme gibi terimler hayatımızın her alanında karşımıza çıkmaktadır. Yapay zekâ, bilgisayar sistemlerinde insanlara has olan tanıma, kavrama, öğrenme gibi yetenekleri taklit ederek bu yeteneklerini modellemesi anlamına gelmektedir (Yılmaz, 2021). Makine öğrenmesi; çevresel ortamdan öğrenerek insan zekasını taklit etmek üzere tasarlanmış hesaplamalı algoritmaların evrilen bir dalıdır (El Naqa and Murphy, 2015). Derin öğrenme ise nesne tanıma, konuşma tanıma ve doğal dil işleme gibi alanlarda sıklıkla kullanılan Yapay Sinir Ağlarından (Artificial Neural Network- ANN) meydana gelen makine öğrenmesinin alt dallarından biridir (Kaya, 2019). Bu tanımlardan da anlaşılacağı üzere bu üç alan birbirleriyle ilişki içerisindedir. Derin öğrenme, temel olarak veri tabanlı öğrenme yaklaşımıdır ve büyük veri kümeleri üzerinde çalışır. Derin öğrenme, özellikle görüntü, dil ve ses gibi yüksek boyutlu verilerde başarılı sonuçlar verir. Bu nedenle bilgisayarlı görü sistemleri, ses tanıma, doğal dil işleme, otonom araçlar, insansız hava araçları, tıbbi görüntü analizleri ve oyunlar gibi birçok alanda sıklıkla kullanılmaktadır (Goodfellow et al., 2016). Derin öğrenme algoritmaları, insan serebral korteksinin işlevini taklit edecek şekilde tasarlanmıştır. Bu algoritmalar, derin sinir ağlarının, yani birçok gizli katmana sahip sinir ağlarının

temsilleridir. CNN, çok fazla parametreye sahip olan büyük veri kümelerini eğitebilen derin öğrenme algoritmalarından biridir (Chauhan et al., 2018).

Derin öğrenmenin tarihçesi ve gelişimi

- Derin öğrenmenin tarihi, ANN'in 1940'lı yıllarına dayanmaktadır. Ancak bu dönemde ANN yeterince gelişmemiştir ve derin öğrenmenin temeli atılmamıştır.
- 1980'li yıllara kadar yeterince ilerleme kaydedilemedi fakat bu yıldan itibaren ANN modelleri derinleştirildi ve daha etkili sonuçlar elde edilmeye başlandı. Fakat bu modellerin eğitilmesi zordu ve bu nedenle verimli sonuçlar elde edilemiyordu.
- 2000'li yıllarından başından itibaren, ANN modelleri ve derin öğrenme teknikleri sadece akademik çalışmalarda kullanılmaya başlandı.
- 2006 yılına gelindiğinde Hinton ve arkadaşları tarafından "Geri Yayılım (back propagation)" adı verilen bir eğitim algoritması geliştirildi. Bu sayede derin öğrenmede daha doğru ve hızlı sonuçlar alınmaya başlandı.
- 2010'lu yıllardan itibaren Derin Öğrenme tekniklerinde özellikle görüntü sınıflandırma ve doğal dil işleme alanlarında çok büyük gelişmeler yaşanmaya başladı. Büyük veri kümelerinin ortaya çıkmasıyla birlikte, gelişmiş eğitim algoritmalarının kullanılmaya başlanmasıyla derin öğrenme uygulama alanları genişledi.
- Günümüzde gittikçe gelişen derin öğrenme teknikleri yüz tanıma, nesne tanıma, doğal dil işleme, ses işleme ve tıbbi teşhis gibi birçok alanda kullanılmaktadır (Goodfellow et al., 2016).

Bu bilgiler ışığında derin öğrenmenin bir alt dalı olan CNN mimarileri kullanılarak fotoğrafçılık alanında kelebek türü sınıflandırmada yaşanan problemlerin üstesinden gelmek amacıyla bir çalışma yapılma gereği duyulmuştur. Tez çalışmasında, kelebek sınıflandırılma için CNN mimarilerini kullanarak karşılaştırmalı bir yaklaşım sunulmuştur. Bu doğrultuda farklı CNN mimarilerinin transfer-learning (öğrenme aktarımı) yöntemi ile önceden eğitilmiş modellerden ağırlıkları alınmıştır. Daha sonra doğruluğu arttırmak için mimariler üzerinde katman ekleme, hiper parametre ayarı gibi iyileştirmeler yapılmıştır. Kelebeklerin tür tespiti ve sınıflandırılması için doğru ve hızlı bir sonuç elde etme yöntemi ortaya sunulmuştur.

1.2 Tez Çalışmasının Literatüre Katkısı

Makro fotoğrafçılık ile ilgilenen fotoğrafçılar ve doğa araştırmacıları, kelebek fotoğraflarını sınıflandırırken tür tespiti yapmakta zorluklar yaşamaktadırlar. Literatür incelendiğinde kelebek türlerinin sınıflandırılması ile ilgili birçok çalışma tespit edilmiştir. Bu çalışmalar kaynak araştırması bölümünde detaylandırılarak açıklanmıştır. Fakat çalışmalar incelendiğinde bazı noktalarda eksiklikler ve problemler olduğu görülmüştür. Bu eksiklikler şu şekilde listelenebilir;

- I. Yapılan çalışmalarda sınıflandırma doğruluklarının çok değişkenlik gösterdiği görülmüştür.
- II. Kelebek sınıflandırma için veri seti elde etmek oldukça zordur. Öğrenmenin yüksek olması için çok sayıda fotoğrafa ihtiyaç duyulmuştur. Çalışmalarda görüntü ve tür sayısının genellikle kısıtlı olduğu saptanmıştır.
- III. Genellikle modelin performansını arttırmak için tek bir model veya birkaç model üzerine odaklanılmıştır.

Bu gibi eksikliklerin üstesinden gelebilmek amacıyla böyle bir sınıflandırma çalışması yapma gereği duyulmuştur. Yapılan çalışmanın literatüre katkıları şu şekilde sıralanabilir;

- I. Oluşturulan veri seti fazla sayıda tür ve görüntüden meydana gelmektedir. Çok sayıda tür olmasına rağmen yüksek doğruluk oranları elde edilmiştir.
- II. 8 farklı CNN mimarisi kullanılarak sınıflandırmaya farklı ve karşılaştırmalı bir yaklaşım yapılmıştır. Transfer öğrenme, ince ayar, hiper parametre ayarı gibi ayarlamalar yapılarak öğrenme iyileştirilmiştir.
- III. Katman eklenmeden öğrenme aktarımı, katman ekleyerek hiper parametre ayarı yapılmış öğrenme aktarımı şeklinde iki farklı yöntem karşılaştırmalı olarak ele alınmıştır.

Bu sayede literatürde kelebek sınıflandırma ile ilgili yapılmış olan çalışmalara farklı bir yaklaşım sunulması amaçlanmıştır. Ayrıca TÜBİTAK'ın Bilim Teknik dergisinin eki olarak yayınladığı Türkiye'nin Kelebekleri adlı ekte yer alan Türkiye'de en yaygın görülen kelebek türleri kullanılmıştır (Bilim ve Teknik, 2011). Bu sayede sınıflandırma için hangi türlerin kullanılacağı belirlenmiştir. Çalışma bu türler üzerinde gerçekleştirilmiştir.

1.3 Tez Çalışmasının Organizasyonu

Tez çalışmasının organizasyon adımları şu şekildedir:

1. Giriş bölümünde tez çalışmasına giriş, tez çalışmasının literatüre katkısı ve tez çalışmasının organizasyonu bölümleri bulunmaktadır.
2. Kaynak araştırması bölümünde Makine öğrenmesi, ANN, Aşırı Öğrenme Makinesi (Extreme Learning Machine- ELM) ve Derin Öğrenme gibi yöntemler kullanılarak sınıflandırma ve kelebek sınıflandırma alanında yapılmış çalışmalarda kullanılan yöntemler ve elde edilen sonuçlar sunulmuştur.
3. Materyal ve Yöntem bölümünde sırasıyla veri setinin oluşturulması, CNN ve CNN'de katman yapısı, çalışmada kullanılan CNN mimarileri, CNN'de öğrenme aktarımı nasıl uygulanır, katman eklemeyen öğrenme aktarımı yönteminin uygulaması ve son olarak katman eklenerek öğrenme aktarımı yönteminin uygulaması bölümleri yer almaktadır.
4. Araştırma sonuçları ve tartışma bölümünde katman eklemeyen yapılan öğrenme aktarımı yönteminde elde edilen sonuçlar, katman eklenerek yapılan öğrenme aktarımı yönteminde elde edilen sonuçlar ve son olarak iki yöntemde elde edilen sonuçların karşılaştırıldığı bölümlerden oluşmaktadır.
5. Sonuçlar ve öneriler bölümünde elde edilen sonuçlar ve gelecek için önerilerde bulunularak tez çalışmasının organizasyonu tamamlanmıştır.

2. KAYNAK ARAŞTIRMASI

Kelebek, böceklerin, pul kanatlılar veya kelebekler (Lepidoptera) takımının kanatlı fertlerine verilen genel addır. Dünya üzerinde güncel araştırmalar sonucunda yaklaşık olarak 15000 ile 21000 arasında kelebeğin var olduğu tahmin edilmektedir (Stork, 2018). Kelebekler böcekler arasında en önemli türlerden biridir. Çünkü araştırma konularında çok fazla ele alınırlar. Aynı zamanda kelebeklerin tür sayılarının oldukça fazla olması, renk ve desenlerinin çeşitliliği nedeniyle kelebeklerin sınıflandırılması, oldukça zor ve düşük doğruluk problemlerine sahiptir (Zhao et al., 2019a). Kelebek türleri arasındaki ayırım yapmak bu alanda uzmanlık ve zaman gerektiren bir süreç olduğu için kelebek sınıflandıran uygulamalar bu süreci kolaylaştıracaktır. Son yıllarda, kelebeklerin sınıflandırılması gittikçe daha fazla araştırmacının dikkatini çekmiştir ve kelebek sınıflandırma uygulamaları hakkında birçok araştırma makalesi yazılmıştır. Birçok farklı yöntem uygulanarak tespit için çözümler aranmıştır. Bu yöntemlerin başında Makine öğrenmesi, ANN, ELM ve derin öğrenme gelmektedir. Bundan sonraki bölümde literatürde yer alan makine öğrenmesi, ANN, ELM ve derin öğrenme yöntemleri ile yapılmış kelebek sınıflandırma alanındaki çalışmalar detaylı bir şekilde incelenmiştir.

Bu bağlamda bu tez çalışmasında gerçekleştirilen uygulamalar sayesinde literatürde yer alan çalışmalara ek olarak şu gibi katkılarından bahsedilebilir;

1. 5624 görüntü ve 49 türden oluşan bir veri seti kazandırılmıştır. Bu veri seti, gelecekteki çalışmalar için referans niteliğinde olup, diğer araştırmacılar tarafından da kullanılabilir.
2. 8 farklı CNN mimarisi ile karşılaştırmalı bir yaklaşım uygulanmıştır. Öğrenme aktarımı, ince ayar, hiper parametre ayarı gibi ayarlamalar yapılarak öğrenme iyileştirilmiştir.
3. Öğrenme Aktarımı yöntemi katman eklemeyen ve katman eklemeli şekilde iki farklı yöntem ile karşılaştırmalı bir şekilde ağlara uygulanmıştır.
4. Öğrenme aktarımı yöntemi kullanılarak daha az veri ile hızlı ve yüksek performanslar elde edilmiştir.

2.1 Makine Öğrenmesi (ML) Yöntemleri Kullanılarak Yapılan Kelebek Sınıflandırma Çalışmaları

J. Wang ve arkadaşları yapmış olduğu çalışmada İçerik Tabanlı Görüntü Benzerliği (Content Based Image Retrieval- CBIR) yöntemi kullanarak 259 kelebek türünden 1327 görüntü üzerinde tür tespiti çalışması yapmıştır. Çalışma sonucunda %84 doğruluğa ulaşmıştır. Yapmış olduğu testler sonucunda tür çeşitliliği, görüntü kalitesi ve çözünürlüğün sistemin başarısını çok etkilediğini, kelebeklerde tür tanımlamada kanat üstü şekil yapısının, renk ve doku özelliklerinden daha önemli olduğunu tespit etmiştir (Wang et al., 2012).

S.-H. Kang ve arkadaşları entropi tabanlı Unit Branching Network'de (UBN) Branch Length Similarity (BLS) kullanarak *Colias erate*, *Parnassius bremeri*, *Eurema hecabe*, *Gonepteryx rhamni*, ve *Papilio maackii* türlerinde 50 görüntü üzerinde çalışmıştır. Tür tanımlaması için kanat şeklini kullanarak, kanat üzerinde x-y koordinatları oluşturmuşlardır. 749 piksele bölerek, 749 adet x-y çiftini 15 gruba ayırmıştır. Gruplar için entropi profillerini hesaplamışlardır ve profilleri ağırlıklandırmışlardır. Profiller, sırayla birleştirilerek bir kanadın şekli için tek bir ağırlıklı BLS entropi profili oluşturmuştur. Ağırlıklı BLS profilleri arasındaki korelasyon katsayısı ile istatistiksel karşılaştırmalar yapmışlardır. Deneysel sonuçlarda, bu yöntemin kelebek türü tanımlama için istatistiksel olarak başarılı olduğunu göstermişlerdir (Kang et al., 2012a).

S. N. A. Hassan arkadaşlarıyla birlikte, kelebek ve çekirge türlerinin renk ve şekil özelliklerine göre otomatik tür tanımlaması yapabilen bir çalışma yapmışlardır. Yaptıkları çalışmada kullandıkları 150 kelebek ve 150 çekirge görüntüsünden 100 görüntüyü test için kullanmışlardır. RGB görüntüleri 100x100 piksel olarak yeniden boyutlandırıp daha sonrasında gri tonlama görüntülere dönüştürmüşlerdir. Ölçekle Değişmez Unsur Dönüşümü (Scale Invariant Feature Transform- SIFT) kullanarak görüntülerden özellik çıkarımı yapmışlardır. Daha sonrasında Destek Vektör Makineleri (Support Vector Machine- SVM) sınıflandırıcısı kullanarak kelebekleri ve çekirgeleri sınıflandırmışlardır. Çekirgelerde %98 doğruluk elde ederken, kelebekler için %84 doğruluk değeri elde etmişlerdir. Genel sınıflandırma doğruluğu ile %92 olarak belirlenmiştir (Hassan et al., 2014).

L. Kaycı ve Y. Kaya yapmış oldukları çalışmada kelebek türlerini sınıflandırmada Gri Seviyeli Eş-oluşum Matrisi (Gray-Level Co-Occurrence Matrix-GLCM) ve Çok Değişkenli Lojistik Regresyon (Multivariable Logistic Regression-MLR) algoritmalarıyla Pieridae ailesine ait 19 farklı türden toplam 190 kelebek görüntüsü kullanılmıştır. Kelebek türlerinin yüzey dokusu özelliklerini kullanarak tanımlama yapmayı amaçlamışlardır. On bir adet dokusal özellik çıkarımı dört yönde (0° , 45° , 90° ve 135°) ve mesafelerde ($d = 1, 2, 3$ ve 4 piksel) tekstür ortalaması ile karakterize edilmiştir. GLCM+MLR'nin 10 kaç çapraz doğrulama yöntemi ile uyguladıkları çalışma sonucunda %96,3 gibi yüksek bir doğruluk oranı elde etmişlerdir (Kaycı and Kaya, 2014).

Y. Kaya ve L. Kaycı yine birlikte yapmış oldukları bir diğer çalışmada geleneksel yöntemlerin aksine Kaba Küme (Rough Set- RS) ve GLCM kullanarak yeni bir yaklaşım geliştirmişlerdir. Bu çalışmada Pieridae ailesinden 19 türden 190 kelebek görüntüsü kullanılmıştır. GLCM+RS yöntemi ile yapılmış bu çalışmada tanımlama doğruluğu %89,47 oranında elde edilmiştir (Kaya et al., 2015).

F. Li ve Y. Xiong yapmış oldukları kelebek sınıflandırma çalışmasında görüntülerin şekil ve doku yapılarını incelemişlerdir. Histograms of Multi-Scale Curvature (HoMSC) ve GLCM sırasıyla kelebekleri şekil ve doku yapısını açıklamak için kullanılmıştır. Sınıflandırma sonucu elde etmek için ise K-En Yakın Komşu (K-Nearest Neighbors- KNN) sınıflandırıcısını kullanmışlardır. Veri seti olarak 50 kelebek türünden toplamda 750 görüntü kullanılmıştır. Sınıflandırma sonucunda %98 doğruluk elde edilmiştir (Li and Xiong, 2018).

Y. Kartika ve arkadaşları yaptıkları çalışmada renk özellik ve doku özellik çıkarımının kombinasyonu yöntemi benimsenmiştir. Hue Saturation Intensity (HSV) renk uzayı kullanılarak renk özelliklerinden 72 özellik, Yerel İkili Desenler (Local Binary Pattern- LBP) kullanılarak doku özelliklerinden 328 özellik çıkarılmıştır. Renk ve doku özellik çıkarımı birleştirilerek sonuçlar sınıflandırılmıştır. 10 türden 890 görüntü üzerinde yapmış olduğu sınıflandırma çalışmaları sonucunda %72 ortalama doğruluk elde edilmiştir (Kartika et al., 2018).

Y. Kartika yapmış olduğu bir diğer çalışmada LBP ve bölge tabanlı sınıflandırma algoritması ile bir yöntem geliştirmiştir. LBP kullanılarak doku özelliklerini 256 özellik

olarak çıkarmış, bölge tabanlı sınıflandırma ile şekil özelliklerini çıkarmıştır. Görüntüleri normalize ederek 256x160 piksel boyutlarında kullanmıştır. Sınıflandırma yapabilmek için ise SVM yöntemini kullanmıştır. 10 türden 890 görüntü üzerinde yapmış olduğu çalışma sonucunda doku özelliklerinde %68, şekil özelliklerinde %49 doğruluk elde ederken, ikisinin birleştirilmesiyle uygulanan yöntemde %66 doğruluk elde etmiştir (Yudha Kartika, 2018).

X. Ankang ve arkadaşları GLCM ve ağırlıklı bir KNN sınıflandırıcısını kullanmışlardır. 10 farklı türden 150 görüntü için yapmış oldukları çalışma sonucunda görüntülerin önce ön işleme yaparak 840x600 piksel boyutuna dönüştürmüş, daha sonra gri seviyeli tonlama haline çevirmişlerdir. Daha sonra 16 bloğa ayrılan görüntülerin son 3 bloğu seçilerek doku özellikleri çıkarılmıştır. Çıkarılan özellikler homojenlik, enerji, kontrast ve korelasyona dayanmaktadır ve 2 uzaklık ve 4 yönlendirmeye göre belirlenmiştir. Daha sonra ağırlıklı KNN sınıflandırma algoritması uygulanarak sınıflandırma yapılmıştır. Sınıflandırma sonucunda tam görüntülerden en düşük %44 en yüksek %80 sınıflandırma doğruluğu elde etmişlerdir. Ağırlıklı KNN ile yapmış oldukları sınıflandırma sonucunda ise en yüksek %98 doğruluğa ulaşmışlardır (Xue et al., 2019).

R. Andrian ve arkadaşları yapmış oldukları çalışmada Gita Persada Kelebek Parkı'nda doğal yollarla üreyen kelebek yetiştirme merkezindeki kelebek üzerinde çalışmalar yapmışlardır. 6 farklı türden (*Centhosia penthesilea*, *Papilio memnon*, *Papilio nephelus*, *Pachliopta aristolochiae*, *Papilio peranthus* ve *Troides helena*) toplamda 600 üst kanat görüntüsünü veri olarak kullanmışlardır. İlk olarak görüntü ön işleme adımları olan, ölçeklendirme, segmentasyon ve gri tonlama gibi yöntemler uygulamışlardır. GLCM yöntemi kullanarak (0° , 45° , 90° ve 135°) açılarda ve piksel mesafesi (d) = 1 kullanılarak görüntülerin açısal ikinci moment, kontrast, homojenlik ve korelasyon özellikleri çıkarılmıştır. Sınıflandırma yöntemi olarak ise KNN kullanılarak 1, 3, 5, 7, 9, 11, 13, 15, 17, 19, 21 ve 23 gibi k değerleriyle denemeler yapılmıştır. Yapılan çalışmalar sonucunda en yüksek doğruluk oranı 90° açındaki $k = 5$ değeriyle %91,1 olarak tespit edilmiştir (Andrian et al., 2019).

2.2 Yapay Sinir Ağları (ANN) Kullanılarak Yapılan Kelebek Sınıflandırma Çalışmaları

Kelebek sınıflandırmada ANN birçok çalışmada tercih edilmiştir. S.-H. Kang ve arkadaşları yaptıkları çalışmada, 7 farklı kelebek türünden toplamda 268 görüntü ile çalışmışlardır. Kelebek kanatlarının sol bölümünü kullanarak 379x567 piksel boyutlarında siyah-beyaz görüntüye dönüştürmüşlerdir. Daha sonra Prewitt kenar dedektörü ve kanatların sınırlarının hesaplanması için 360 eşit piksele bölünmüştür. Sınıflandırma için giriş, gizli ve çıkış katmanlarından oluşan yapay sinir ağında sırasıyla 360,17 ve 7 nörondan meydana gelmiştir. BLS entropi profili ve ANN kullanılarak yapılan çalışmada ağı eğitmek için geri yayılım algoritması kullanılmıştır. Aşırı öğrenmeyi engellemek için ise Çapraz-Doğrulama yöntemi uygulanmıştır. Bu çalışmanın sonucunda BLS entropi ağıyla eğitilen model en yüksek %86 doğruluk değerine ulaşmıştır (Kang et al., 2012b).

Y. Kaya ve L. Kaycı yaptıkları çalışmada GLCM ve ANN kullanarak 14 farklı kelebek türü üzerinde 140 görüntüde sınıflandırma çalışması yapmışlardır. İlk olarak görüntüleri 256x256 piksel boyutlarına dönüştürmüşlerdir. Daha sonra görüntülerden beş farklı dokusal özellik çıkarımı yapmışlardır. Bu özellikler korelasyon, kontrast, enerji, entropi ve homojenliktir. Ardından görüntüleri RGB kanallarının ortalama gri seviyesi bulunmuştur. Bu özelliklerin ortalama değeri alınarak ANN'e giriş olarak kullanılmıştır. Sınıflandırma sonucunda test verisi için en yüksek %92,85 doğruluk elde edilmiştir (Kaya and Kayci, 2014).

S.-H. Kang yapmış olduğu bir diğer çalışmada ANN kullanarak farklı açılardan kelebek türü tanımlama yapmaya çalışmıştır. BLS'den aldığı özellikleri yine ANN'e giriş olarak verilmiştir. 15 farklı kelebek türünden toplam 150 görüntü kullanılmıştır. Görüntüler işlenerek 379x567 piksele dönüştürülerek işlenmiştir. Sınıflandırma için kullanılan sinir ağı giriş, gizli katman ve çıkış olmak üzere üç katmandan meydana gelmiştir. Bir adet giriş vektörü, 97 adet gizli katman ve 15 adet çıkış katmanı kullanılmıştır. Eğitimler sonucunda en düşük %89,6 ve en yüksek %100 doğruluk elde edilmiştir (Kang et al., 2014).

Y. Kaya ve arkadaşları yapmış olduğu bir diğer çalışmada ise LBP ve ANN kullanarak Papilionidae ailesinden 5 türe ait toplamda 50 görüntü içeren bir veri seti kullanmışlardır. Görüntüler 256x256 piksel olarak yeniden boyutlandırılmıştır. LBP matrisinden önce işlenmiş olarak beş boyutlu dokusal özellik kümesi kullanılmıştır. Özellik matrisi sınıflandırma için ise ANN'e 10 kat çapraz doğrulama işlemi yapılmıştır. Yapılan işlemler sonucunda en yüksek %98 doğruluk değerine ulaşmışlardır (Yılmaz Kaya et al., 2015).

Alhady ve arkadaşları yaptıkları çalışmada LBP ve ANN kullanmışlardır. Hypolimnas bolina ve Ideopsis vulgaris adlı iki türe ait 78 görüntüden oluşan bir veri seti kullanmışlardır. Görüntüler önce RGB'den gri tonlamalı görüntüye dönüştürülmüş ardından LBP'ye dönüştürülmüştür. LBP tanımlayıcısı görüntüyü 9 piksel içeren bölgelere böler. Her hücre için ikili sayı hesaplandıktan sonra histogram oluşturulmuştur. Daha sonra bu histogramdan ortalama, enerji, sapma, korelasyon ve entropi özellikleri hesaplanmıştır. ANN, Giriş katmanı, 105 nöron içeren gizli katman ve iki sınıftan oluşan çıktı katmanından meydana gelmiştir. Sınıflandırma sonucunda Ideopsis vulgaris türünde %90, Hypolimnas bolina türünde %100 doğruluk elde etmişlerdir (Alhady and Kai, 2018).

2.3 Aşırı Öğrenme Makineleri (ELM) Kullanılarak Yapılan Kelebek Sınıflandırma Çalışmaları

Bir diğer kelebek sınıflandırma yöntemlerinden olan ELM ile de çeşitli çalışmalar yapılmıştır. Bu çalışmaların ilki Y. Kaya ve arkadaşları tarafından, Papilionidae ailesine ait 5 türden 50 görüntüye Gabor Filtreleri (Gabor Filters- GF) uygulayarak özellik çıkarımı yapıp ardından ELM yöntemleri ile bilgisayarlı görü kullanarak bir çalışma yapmışlardır. Görüntüler 256x256 piksele dönüştürülmüş, ardından 35 görüntü eğitim 15 görüntü ise test için kullanılmıştır. Farklı aktivasyon fonksiyonları ile yapılan çalışmada en başarılı sonuçları Sinüs aktivasyon fonksiyonu vermiştir. ANN, SVM, Bayesian Network (BN), Kısmi Karar Algoritması (PART), Lojistik Regresyon (LR), J48 Karar Ağacı Algoritması gibi yöntemlerle ELM'nin etkinliğini kanıtlamak amacıyla 10 kat çaprazlama ile deneysel olarak uygulanmıştır. Deney sonucunda ELM yönteminin kelebek sınıflandırmak için en iyi sonucu verdiğini %97 gibi yüksek bir doğrulukla elde etmişlerdir (Kaya et al., 2013).

Y. Kaya ve arkadaşları yapmış olduğu bir diğer çalışmada ise 19 farklı kelebek türünden 190 fotoğraf üzerinde deneyler yapmıştır. GLCM+LBP ve GLCM+ELM olmak üzere iki farklı kombinasyon ile karşılaştırmalar yapmışlardır. GLEM+ELM ile yapılan yöntemde %98,25 doğruluk elde edilirken, GLEM+LBP yöntemi ile %96,49 doğruluk elde edilmiştir (Yılmaz Kaya Lokman Kayci and Ertuğrul, 2014).

Ö. F. Ertuğrul ve arkadaşları Texture Energy Measure (TEM) filtreleme yöntemi kullanılarak özellik çıkarımı yapmıştır daha sonrasında KNN, SVM ve ELM sınıflandırma için kullanılmıştır. 190 görüntüden oluşan 19 tür üzerinde çalışma yapmışlardır. Görüştüler 512x512 piksel boyutlarına yeniden boyutlandırılmıştır. Yapılan çalışma sonucunda TEM+ELM yönteminde %99,47, TEM+SVM yönteminde %98,16, TEM+ELM yönteminde ise %99,26 doğruluk değeri elde etmişlerdir (Ertugrul et al., 2015).

2.4 Derin Öğrenme Yöntemleri Kullanılarak Yapılan Kelebek Sınıflandırma Çalışmaları

Kelebek sınıflandırma için kullanılan yöntemlerden birisi de derin öğrenmedir. Bu yöntem özellikle derin öğrenme tekniklerinin gelişmesiyle günümüzde oldukça popüler hale gelerek çalışmalarda sıklıkla kullanılmaya başlanmıştır. L.-Q. Zhu ve arkadaşları kelebek türlerini sınıflandırmak amacıyla Derin Evrimsel Sinir Ağı (Deep Convolutional Neural Network- DCNN) ve SVM kullanarak bir yöntem geliştirmişlerdir. 22 farklı kelebek türünden 1301 görüntü kullanarak, görüntüleri 227x227 piksel olarak yeniden boyutlandırmışlardır. Özellik çıkarımı için AlexNet modelinin ilk tam bağlantılı katmanı kullanılmıştır. Daha sonra çıkarılan özelliklerle Chi-kare SVM sınıflandırma için kullanılmıştır. Yapmış oldukları sınıflandırma sonucunda %100 doğruluk elde etmişlerdir ve görüntüden tür sınıflandırma işlemi 200 milisaniye sürmüştür (Le-Qing Zhu Meng-Yuan Ma and Wang, 2017).

J. A. Carvajal ve arkadaşları yapmış olduğu çalışmada 15 kelebek türünden 2110 görüntü kullanmışlardır. Yapmış oldukları çalışmada AlexNet, VGG-F, VGG-16 çok

derin ve VGG-19 çok derin ağırları kullanılmıştır. Ve yapılan sınıflandırma için görüntüler ön işleme yöntemleri ile AlexNet için 227x227, VGG16 ve VGG19 için ise 224x224 piksel boyutlarında kullanılmıştır. Yapılan sınıflandırma sonucunda %92'nin üzerinde doğruluk elde edilmiştir (Carvajal et al., 2017).

Q. Chang ve arkadaşları Kuzey Amerika'da yer alan 450 kelebek ve 186 güve türünden toplamda 14270 görüntü üzerinde sınıflandırma çalışması yapmayı amaçlamışlardır. Veri setini eğitim, test ve doğrulama için üçe ayırmışlardır. Sınıflandırma için Inception-v3, VGG-19, Resnet-18 ve Resnet-34 gibi dört farklı CNN mimarisini kullanmıştır. Bu dört modelden en iyi sonuçları Resnet-18'in verdiğini tespit etmişlerdir. Ardından hiper parametre ayarlamalarıyla 0,01 öğrenme oranında en iyi sonucu elde etmişlerdir. En yüksek doğruluk değeri olarak %92,6'lık yüksek bir başarı elde etmişlerdir (Chang et al., 2017).

L. Nie ve arkadaşları 82 kelebek türünden 10881 görüntü üzerinde AlexNet, VGGNet ve ResNet gibi mimarilerle sınıflandırma için çalışmalar yapmışlardır. Ağırlıkları İmagenet veri seti ile önceden eğitilmiş modellerden çekmişlerdir. Daha sonra modeller üzerinde ince ayar işlemleri yapmışlardır. Yaptıkları sınıflandırma sonucunda en yüksek doğruluğu Resnet ile %95,2 olarak elde etmişlerdir (Nie et al., 2017).

B. Liang ve arkadaşları yaptıkları çalışmada 94 kelebek türünden toplamda 1048 görüntü ile sınıflandırma çalışması yapmıştır. Yaptıkları sınıflandırmada YOLO algoritması kullanmışlardır. Görüntüler üzerinde veri artırma işlemleri yaparak görüntü sayısını arttırmışlardır. Sınıflandırma sonucunda %98,3 doğruluk elde etmişlerdir (Liang et al., 2020).

L. Zhu ve P. Spachos 10 türden 832 görüntü üzerinde çalışarak bir Android uygulaması geliştirmişlerdir. Temel Bileşen Analizi ve SVM ile 4-Konvülsiyonlu CNN ile önceden eğitilmiş VGG16 modelini kullanmışlardır. Yaptıkları eğitim aşamalarında farklı aktivasyon fonksiyonları, farklı batch boyutları gibi denemeler yapılarak sonuçlar iyileştirilmiştir. En yüksek doğruluk olarak ise %98,44 gibi yüksek bir değer elde etmişlerdir (Zhu and Spachos, 2019).

N. N. K. Arzar ve arkadaşları 4 farklı kelebek türünden toplamda 120 adet görüntü ile sınıflandırma çalışması yapmışlardır. Görüntüleri ilk olarak 224x224 piksel boyutlarına dönüştürmüş ve görüntüleri GoogleNet mimarisi kullanarak eğitmişlerdir. Modeli iyileştirmek amacıyla son üç katmanını tam bağlantı, softmax ve sınıflandırma katmanı ile değiştirmişlerdir. Sınıflandırma sonucunda %97,5 doğruluğa ulaşmışlardır (Kamaron Arzar et al., 2019).

R. Zhao ve arkadaşları 111 farklı kelebek türünden 5695 görüntü ile Faster R-CNN kullanılarak bir sınıflandırma çalışması yapmışlardır. Çalışmada özellik çıkarımı için Faster R-CNN'den Relu, konvolüsyon ve havuzlama katmanları kullanılarak gerçekleştirilmiştir. Modelin hiper parametre ayarları olarak 0,001 öğrenme oranı, 0,9 momentum ve 5000 eğitim tekrarı yapılmıştır. %70,4 doğruluk elde edilmiştir (Zhao et al., 2019b).

Z. Lin ve arkadaşları yaptıkları çalışmada 56 kelebek türünden 24836 görüntü kullanmıştır. Bu görüntülere çeşitli görüntü işleme yöntemleri eklenerek görüntü sayısı 173852 ye çıkartmışlardır. Görüntülerin %75'ini test için kalan %25'lik bölümünü doğrulama verileri için ayırmıştır. Sınıflandırma için S-CNN mimarisini kullanmışlardır. Yapılan sınıflandırma işlemi sonucunda %94,17 doğrulama doğruluğu ve %93,36 test doğruluğu elde etmişlerdir (Lin et al., 2020).

A. S. Almryad ve H. Kutucu yaptıkları çalışmada 10 farklı kelebek türüne ait 17769 görüntüde VGG16, VGG19 ve Resnet ile denemeler yapmışlardır. Veri setini %80 eğitim ve %20 test verisi olarak ayırmışlardır. Resnet ile eğitim verilerinde yüksek doğruluk elde etmelerine rağmen, test verilerinde istenilen doğruluğu elde edememişlerdir. En yüksek test doğruluğu VGG-16 modeli ile %79,5 olarak elde edilmiştir (Almryad and Kutucu, 2020).

S. N. Hegde ve arkadaşları yapmış oldukları sınıflandırma çalışmasında CNN'nin VGG16 mimarisini kullanarak 10 kelebek türü üzerinde 832 görüntüde bir sınıflandırma çalışması yapmışlardır. Yaptıkları çalışmada görüntülerin %85'ini eğitim için %15'ini ise test için olmak üzere ayırmışlardır. Yaptıkları sınıflandırma sonucunda %92 gibi yüksek bir doğruluk elde etmişlerdir (Samruddhi N Hegde et al., 2020).

D. Xin ve arkadaşları 30 türden 1299 görüntü üzerinde üç farklı modeli birleştirerek bir sınıflandırma önermişlerdir. Bunlar Sıkıştırma ve Uyarılma , Mekânsal Dikkat ve Genişletilmiş Artıklı Ağ kullanmışlardır. Mimari olarak ise CNN'nin Resnet50 mimarisini kullanmışlardır. Ve bu sınıflandırmalar sonucunda 300 eğitim boyunca eğitim işlemini devam ettirmişlerdir. En yüksek %95,6 doğruluğa ulaşmışlardır (Xin et al., 2020).

A. Tan ve arkadaşları geliştirdikleri model ile arka planı karışık olan görüntülerde kelebek sınıflandırmak için bir yöntem uygulamışlardır. Monarch kelebeği, Argynnis hyperbius, Polygonia c-aureum, Danaus genutia ve Papilio machaon türlerinden toplamda 1188 görüntü üzerinde çalışma yapmışlardır. R-CNN ve FCM-KM yöntemlerinin birleşimine dayanan bir yöntem önermişlerdir. Görüntülerin 913 tanesini eğitim için, geriye kalan 275 tanesini ise test için kullanmışlardır. Yaptıkları sınıflandırma yöntemi sonucunda %83,62 doğruluk elde etmişlerdir (Tan et al., 2020).

T. Hari yapmış olduğu çalışmada Hindistanda yer alan kelebek türlerinden 315 tanesinden toplamda 34024 tür üzerinde sınıflandırma çalışması yapmıştır. ResNet-18, ResNet-34, ResNet-50, ResNet-121, ResNet-152, Alex-Net, DenseNet-121, DenseNet-161, VGG-16, VGG-19, SqueezeNet-v1.1 gibi farklı CNN mimarilerinden toplamda 11 tanesi ile karşılaştırmalı bir yaklaşım yapmıştır. Bu çalışma sonucunda ResNet-152 mimarisine yaptığı sınıflandırmada en yüksek doğruluk değerlerine birinci eğitimde %94,44, üçüncü eğitimde %98,46 ve beşinci eğitim sonucunda %99,09 olarak oldukça yüksek bir doğruluk elde etmişlerdir (Theivaprakasham, 2021).

T. Çakar ve arkadaşları çalışmalarında önerilen modele göre, otomatik taksonomik kelebeklerin sınıflandırılmasını incelemişlerdir. CNN uygulamaları çalışmada kullanılan 6 kelebek türünün 7148 fotoğrafına uygulanmıştır. Verilerin %80'ini eğitim için, %20'sini ise test için kullanmışlardır ve çalışma sonunda %92,73 doğruluk derecesi elde etmişlerdir (Tek and Emel, 2020).

B. Elmas'ın yaptığı çalışmada ise Türkiye'deki 9 aile ve 416 kelebek türü incelenmiştir. 13528 görüntü toplanmış veri artırma yöntemleriyle veri sayısı 67640'a çıkmıştır. Kelebek türlerini sınıflandırmak için yedi adet CNN kullanmışlardır. Tür sayısının fazlalığından dolayı iki basamaklı ağ modeli kullanmışlardır. Birinci basamakta

%95,88, ikinci basamakta ise %91,99 ile %100 arasında başarı oranı elde etmişlerdir (Elmas, 2021).

D. Ruaika ve Shofwatul Uyun yapmış oldukları çalışmada 75 farklı kelebek türü üzerinden toplamda 9285 adet eğitim, 375 doğrulama ve 375 test görüntüsü üzerinde çalışmışlardır. Yaptıkları çalışmada CNN mimarilerinden VGG16, VGG19, MobileNet, Xception, InceptionV3 ve Resnet50'yi kullanmışlardır. Bu ağlarla birlikte AdaDelta, SGD, RMSprop, Adam gibi farklı optimizasyon türleri ile denemeler yapmışlardır. Yaptığı çalışma sonucunda en yüksek %95 doğruluğa ulaşmıştır (Ruaika, 2024).



3. MATERYAL VE YÖNTEM

Materyal ve Yöntem bölümünde, ilk olarak veri setinin nasıl oluşturulduğuna, ardından basit bir CNN mimarisinin hangi katmanlardan oluştuğuna ve bu katmanların ne işe yaradığına, son olarak ise çalışmada kullanılan CNN mimarilerinden bahsedilmiştir.

3.1 Veri Setinin Oluşturulması

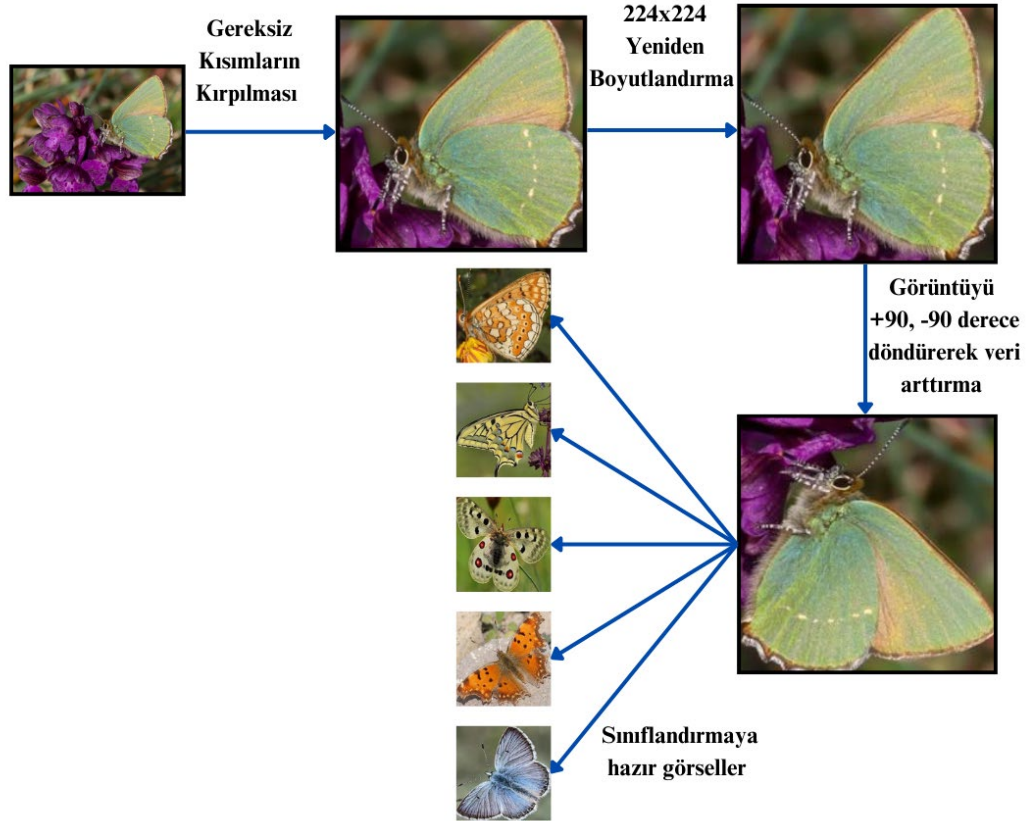
Veri setlerinin oluşturulması aşamasında meydana gelen en büyük problemlerden birisi de oluşturulan veri setlerinin güvenilir ve doğru olup olmadığıdır. Kelebek fotoğraflarında kullanılan veri setleri genellikle doğa fotoğrafçıları tarafından çekilmiş olup, çeşitli fotoğrafçılık sitelerinde sınıflandırılmış başlıklar altında yer almaktadırlar. Çalışmada kullanılan fotoğrafların büyük bir çoğunluğu bu sitelerden, yakın çevrede tanınan doğa fotoğrafçılarından ve çekmiş olduğum fotoğraflardan oluşmaktadır. Yurt içi ve yurt dışından olmak üzere toplamda beş farklı siteden toplanmıştır (“Kelebek-Türk Türkiye’nin Kelebekleri,” 2024) (“euroButterflies,” 2024) (“Butterflies, Butterfly Photo Gallery,” 2024) (“Butterflies of France,” 2024) (“Moths and Butterflies of Europe and North Africa,” 2024). Veri seti hazırlanırken, Tübitak dergisinin yayınlamış olduğu ekte, Türkiye’de en sık görülen kelebekler listesinde yer alan 52 tür dikkate alınmıştır (Bilim ve Teknik, 2011). Veri setlerinin toplandığı siteler incelendiğinde üç türe ait yeterli sayıda fotoğraf elde edilemediği için bu üç tür çıkarılmıştır. Ayrıca bu türlere ait görüntü sayıları eşit olmadığı için veri setine veri artırma işlemi uygulanarak eksik verilerin giderilmesi sağlanmıştır. Bu sayede daha dengeli bir veri seti oluşturularak eğitim sırasında modelin aşırı öğrenme yapmasının önüne geçilmiştir. Veri setinde yer alan görseller eğitim, test ve doğrulama verileri olarak kendi içerisinde üç ayrı bölüme ayrılmıştır. Veri setinde yer alan tür adları, eğitim, doğrulama ve test görsel sayıları aşağıdaki Çizelge 3.1’de listelenmiştir.

Çizelge 3.1. Veri Setinde Bulunan Tür Adları ve Görüntü Sayıları

İndis	Bilimsel Adı	Türkçe Adı	Eğitim	Doğrulama	Test	Toplam Görüntü
1	<i>Aglais urticae</i>	Aglais	85	15	6	106
2	<i>Anthocharis cardamines</i>	Turuncu Süslü	124	15	5	144
3	<i>Aporia crataegi</i>	Alç Kelebeği	120	15	5	140
4	<i>Argynnis niobe</i>	Niyobe	68	15	5	88
5	<i>Argynnis pandora</i>	Bahadır	99	15	5	119
6	<i>Brintesia circe</i>	Kara Murat	42	15	5	62
7	<i>Callophrys rubi</i>	Zümrüt	130	15	5	150
8	<i>Carcharodus alceae</i>	Hatmi Zıpzıp	118	15	5	138
9	<i>Celastrina argiolus</i>	Kutsalmavi	90	15	5	110
10	<i>Chazara briseis</i>	Cadı	68	15	5	88
11	<i>Coenonympha pamphilus</i>	Küçük Zıpzıp Perisi	109	15	5	129
12	<i>Colias crocea</i>	Sarı Azamet	133	15	5	153
13	<i>Euphydryas aurinia</i>	Nazuğum	110	15	5	130
14	<i>Glaucopsyche alexis</i>	Karagözmavisi	132	15	5	152
15	<i>Gonepteryx rhamni</i>	Orakkanat	95	15	5	115
16	<i>Inachis io</i>	Tavuskelebeği	84	15	5	104
17	<i>Iphichlides podalirius</i>	Erik Kırlangıçkuyruğu	121	15	5	141
18	<i>Issoria lathonia</i>	İspanyol Kraliçesi	119	15	5	139
19	<i>Kirinia roxelana</i>	Ağaç Esmeri	50	15	5	70
20	<i>Lampides boeticus</i>	Lampides	88	15	5	108
21	<i>Limenitis reducta</i>	Akdeniz Hanmelikelebeği	120	15	5	140
22	<i>Lycaena phlaeas</i>	Benekliakır Güzeli	124	15	5	144
23	<i>Lycaena thersamon</i>	Küçükateşgüzeli	104	15	5	124
24	<i>Maniola jurtina</i>	Çayıresmeri	54	15	5	74
25	<i>Melanargia larissa</i>	Anadolumelikesi	114	15	5	134
26	<i>Melitaea cinxia</i>	İparhan	121	15	5	141
27	<i>Melitaea didyma</i>	Benekli İparhan	127	15	5	147
28	<i>Nymphalis antiopa</i>	Sarı Bandlı Kadife	56	15	5	76
29	<i>Papilio machaon</i>	Kırlangıçkuyruk	107	15	5	127
30	<i>Pararge aegeria</i>	Karanlık Orman Esmeri	67	15	5	87
31	<i>Parnassius apollo</i>	Apollo	95	15	5	115
32	<i>Pieris brassicae</i>	Büyük Beyazmelek	83	15	5	103
33	<i>Plebejus argus</i>	Çokgözlü Esmer	117	15	5	137
34	<i>Pleris rapae</i>	Küçük Beyaz Melek	97	15	5	117
35	<i>Polygonia egea</i>	Anadolu Yırtıkırtığı	66	15	5	86
36	<i>Polyommatus amandus</i>	Çokgözlü Amanda	110	15	5	130
37	<i>Polyommatus anteros</i>	Çokgözlü Balkanmavisi	73	15	5	93
38	<i>Polyommatus icarus</i>	Çokgözlü Mavi	130	15	5	150
39	<i>Polyommatus semiargus</i>	Mazarinmavisi	89	15	5	109
40	<i>Pontia edusa</i>	Yeni Benekli Melek	109	15	5	129
41	<i>Pseudochazara anthelea</i>	Anadolu Yalancıcadısı	29	15	5	49
42	<i>Pseudophilotes vicrama</i>	Himalayamavisi	79	15	5	99
43	<i>Pyrgus sidae</i>	Sarıbandlı Zıpzıp	70	15	5	90
44	<i>Satyrrium spini</i>	Güzel Sevbeni	81	15	5	101
45	<i>Spialia orbifer</i>	Kızıl Zıpzıp	109	15	5	129
46	<i>Thymelicus sylvestris</i>	Sarı Antenli Zıpzıp	91	15	5	111
47	<i>Vanessa atalanta</i>	Atalanta	64	15	5	84
48	<i>Vanessa cardui</i>	Dikenkelebeği	110	15	5	130
49	<i>Zerynthia cerisyi</i>	Ormanfistosu	61	15	5	81
-	-	-	4643	735	246	5624

Veriler çeşitli kaynaklardan indirildiği için ilk olarak piksel değerleri kaynaklara göre değişiklik göstermektedir. Görseller içerisinde gereksiz olarak yer alan birçok nesne

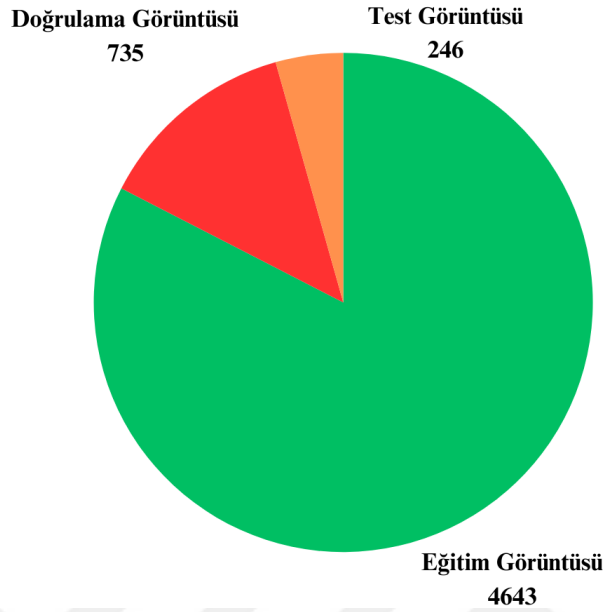
bulunmaktadır. Bu gibi gereksiz gürültüler sınıflandırma ve modelin eğitim sürecinde modeli olumsuz etkilemesi nedeniyle görüntülere ön işleme adımları uygulanmıştır. Görüntü ön işlemede uygulanan adımlar örnek bir fotoğraf üzerinde aşağıdaki Şekil 3.1’de gösterilmiştir.



Şekil 3.1. Veri Ön İşleme Adımları

Ayrıca makine öğrenmesi yöntemlerinde veri seti ayırma yöntemleri, modelin eğitim, doğrulama ve test aşamalarında kullanılmak üzere veri kümesini uygun şekilde bölmek için kullanılır. Train-Test Split, Train-Validation-Test Split, Cross-Validation, Temporal Split, Group-Based Splitting gibi farklı veri seti ayırma yöntemleri vardır. Yapılan çalışmada Train-Validation-Test Split yöntemi tercih edilmiştir. Bu yöntemin tercih edilmesinin sebebi ise modelin gerçek dünya verilerinde nasıl performans sergilediğini görmektir. Eğitim seti, modelin öğrenmek için kullandığı verileri sağlar, doğrulama seti, modelin ayarlanması ve iyileştirilmesi için kullanılır, test seti ise modelin nihai performansını değerlendirmeyi sağlar. Oluşturulan 5624 fotoğraflık veri seti literatürdeki çalışmalar incelenerek Şekil 3.2’de gösterildiği gibi 4643 adeti test verileri

olarak, 735 adet görüntü doğrulama verisi olarak ve 246 görüntüde test verisi olarak ayrılmıştır.

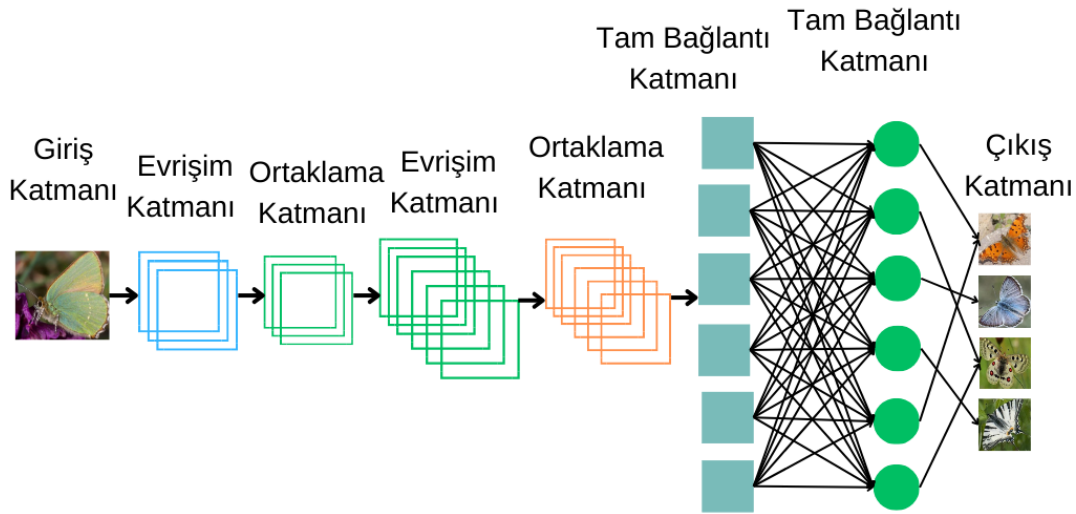


Şekil 3.2. Veri Setinin Train-Validation-Test Split Yöntemi ile Ayrılması

3.2. Evrişimli Sinir Ağları (CNN) ve Katman Yapısı

Derin Öğrenme algoritmaları, insan serebral korteksinin işlevini taklit edecek şekilde tasarlanmıştır. Bu algoritmalar, derin sinir ağlarının, yani birçok gizli katmana sahip sinir ağlarının temsilidir. Evrişimli Sinir Ağı (Convolutional Neural Network - CNN), derin öğrenme alanında yaygın olarak kullanılan ve özellikle görüntü işleme görevlerinde başarıyla uygulanan bir yapay sinir ağı türüdür. Görüntü sınıflandırma, nesne tanıma, yüz tanıma, video analizi gibi görsel veri analizi problemlerinde etkili bir şekilde kullanılır. CNN'lerin amacı genellikle karmaşık yapıdaki görüntülerde yer alan desenleri otomatik tanıyıp öğrenme mantığına göre sınıflandırma yapabilmektir. Basit bir Evrişimli Sinir Ağı temelde üç katmandan oluşur. Bu katmanlar giriş, gizli katmanlar ve çıkış katmanlarıdır. Giriş katmanında görüntü, video gibi giriş verileri yer alır. Gizli katmanlar bir çarpma veya başka bir nokta ürünü ile kırılan bir dizi kıvrımlı katmandan oluşur. Aktivasyon fonksiyonu olarak çıkışta genellikle RELU kullanılır ve takiben havuzlama, tam bağlı katman ve normalizasyon katmanı gibi gizli katmanlar yer alır. Son evrişim katmanının görevi, bir CNN mimarisinde öznitelik çıkarma sürecini

tamamlamaktır. Genellikle sınıflandırma ve benzeri görevler için kullanılan çıktıları üretir. Bu sayede öğrenme işlemi gerçekleştirilmiş olur.



Şekil 3.3. Basit Bir Evrişimli Sinir Ağında Modelin Katman Yapısı

Basit bir Evrişimli Sinir Ağı aşağıdaki katmanlardan oluşmaktadır.

- Giriş Katmanı—Sisteme verilen giriştir (Görüntü, ses veya videodur).
- Evrişim Katmanı — Parametreleri tespit etmeye yarar.
- Aktivasyon Katmanı — Sistemin lineer olmamasının tanımlanması.
- Ortaklama Katmanı — Ağırlığı düşürür ve uygunluğu kontrol eder.
- Tam Bağlantı Katmanı — Sınıflamada kullanılan Standart Sinir Ağı.
- Çıkış Katmanı—Sınıflandırma sonucunda elde edilen sınıflar.

3.2.1. Giriş Katmanı

Giriş katmanı, bir CNN’de en temel ve olması gereken katmanlardan ilkidir. Bu katman verilerin ağı girdi olarak kabul edildiği katmandır. Genellikle CNN’de görsel, ses, metin gibi girdiler kullanılır. Bu girdi verileri, genellikle sayısal veya vektör biçimindedir ve ağı çıktıları belirlemek için kullanılır. Giriş katmanı, girdi verilerinin boyutunu belirler. Örneğin, bir görüntü sınıflandırma probleminde, giriş katmanı, görüntülerin genişliği, yüksekliği ve kanal sayısı gibi boyutları belirler. Örneğin yapılan tez çalışmasında giriş görüntüleri 224x224x3 olarak yani genişlik 224 yükseklik 224 ve kanal sayısı 3 olarak belirlenmiştir. Giriş katmanı, yapay sinir ağlarının mimarisinde önemli bir rol oynar çünkü bu katman, ağı geri kalan kısmına girdi verilerini ileten ilk

adımdır. Girdi katmanının boyutu, ağın mimarisinin ve modelin başarımının belirlenmesinde önemli bir faktördür.

3.2.2. Evrişim Katmanı

Evrişim katmanında, girdide yer alan bir görüntüden özellik çıkarımının sağlandığı katmandır. Evrişim, alıcıda bulunan tüm pikselleri tek bir değere dönüştürür. Örneğin görüntüye evrişim uygulandığında görüntünün boyutu küçüldüğü gibi alandaki tüm pikselleri de bir araya getirilmiş olur. Bu katmanın son çıktısı bir vektördür. Çözülmesi gereken problemin türüne ve öğrenilmek istenen özelliklere göre farklı evrişim türleri kullanılabilir. Evrişim katmanında resmin gerçek boyutlarından daha küçük bir filtre resmin üstüne eklenir.

$$\begin{array}{|c|c|c|c|c|} \hline 3 & 1 & 5 & 2 & 2 \\ \hline 1 & 5 & 4 & 5 & 5 \\ \hline 2 & 1 & 1 & 3 & 4 \\ \hline 5 & 2 & 1 & 1 & 2 \\ \hline 1 & 5 & 2 & 3 & 1 \\ \hline \end{array} \quad * \quad \begin{array}{|c|c|c|} \hline 1 & 0 & -1 \\ \hline 1 & 0 & -1 \\ \hline 1 & 0 & -1 \\ \hline \end{array} \quad = \quad \begin{array}{|c|c|c|} \hline -4 & -3 & -1 \\ \hline 2 & -1 & -5 \\ \hline 4 & 1 & -3 \\ \hline \end{array}$$

5x5 3x3 3x3

Şekil 3.4. Örnek Bir Evrişim İşlemi

Şekil 3.4'te gösterildiği gibi 3×3 'lük bir filtre, 5×5 'lik bir görselin üzerine eklenir. Matrise filtre eklenirken çarpma işlemi 3×3 'lük matris sol üst köşeye denk gelecek şekilde ilk önce soldan sağa kaydırılarak, daha sonra da yukarıdan aşağıya kaydırılarak çarpma işlemleri yapılır. Örneğin 1. Satır 1. Sütündeki sayıyı bulmak için çarpma işlemi $(3 \times 1) + (1 \times 0) + (5 \times (-1)) + (1 \times 1) + (5 \times 0) + (4 \times (-1)) + (2 \times 1) + (1 \times 0) + (1 \times (-1)) = (-3)$ olarak bulunur. Bu işlem tüm satır ve sütunlara uygulanarak aşağıdaki filtrelenmiş olur. Sonuç olarak gereken özniteliklerin tanımlanması evrişim katmanında gerçekleştirilmiş olur. Ayrıca çıkış matrisinin boyutunun hesaplanmasında aşağıdaki formülü kullanılabilir. Burada giriş matrisinin boyutu ve filtre matrisinin boyutunun bilinmesi yeterlidir. 3.1'de yer alan formül ile çıkış matrisinin boyutu 3×3 olarak bulunmuş olur.

Şekil 3.5. Sıfır Piksel Ekleme İşlemi Örneği

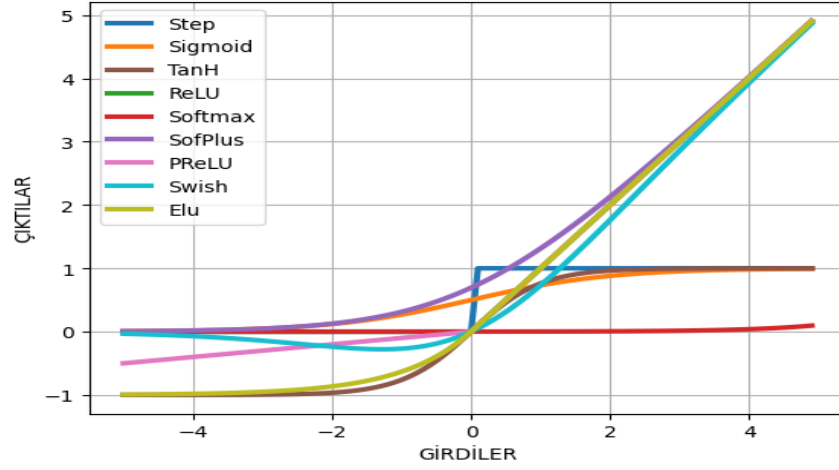
- 2) İkinci yöntem olarak ise giriş görüntüsüne ait piksel değerleri giriş görüntüsünün etrafına eklenerek oluşturulabilir.

3	3	3	1	5	2	2	2	2
3	3	3	1	5	2	2	2	2
3	3	3	1	5	2	2	2	2
1	1	1	5	4	5	5	5	5
2	2	2	1	1	3	4	4	4
5	5	5	2	1	1	2	2	2
1	1	1	5	2	3	1	1	1
1	1	1	5	2	3	1	1	1
1	1	1	5	2	3	1	1	1

Şekil 3.6. Piksel Ekleme İşlemi Örneği

3.2.3. Aktivasyon Fonksiyonu Katmanı (Activation layer):

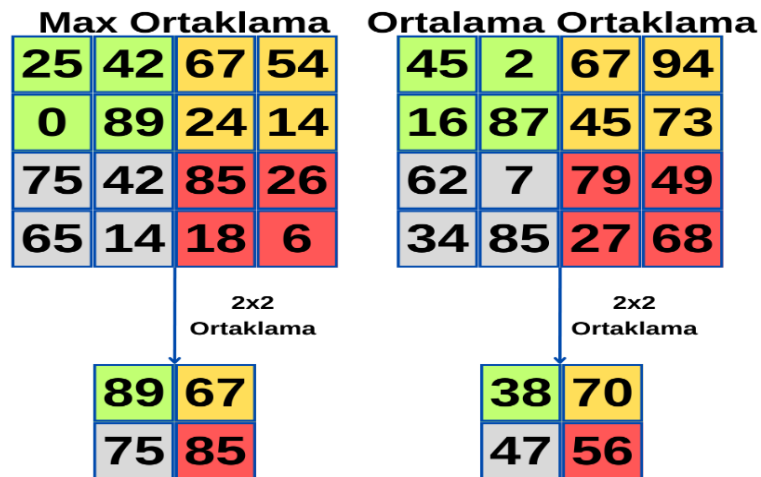
ANN’de doğrusal olmayan gerçek dünya özelliklerini tanıtmak için çıkış üretmeye yarayan fonksiyonlara denir. Genellikle yapmak istediğiniz işleme göre kullanılan aktivasyon fonksiyonları farklılık göstermektedir. Basamak (Step) fonksiyonu, Doğrusal (Linear) fonksiyon, Sigmoid fonksiyonu, Tanjant fonksiyonu, ReLU (Rectified Linear Unit) fonksiyonu, Softmax fonksiyonu, Softplus fonksiyonu, eLU (Exponential Linear Unit), PReLU (Parametric Rectified Linear Unit), Swish fonksiyonu gibi fonksiyonlar mevcuttur. Bu fonksiyonlar genellikle probleme göre değişiklik gösterir. Örneğin derin öğrenme problemlerinde genellikle ReLU fonksiyonu kullanılırken, çıkış katmanlarında genellikle softmax fonksiyonu kullanılır. Aktivasyon fonksiyonları, nöronun çıkışını belirli bir eşiğin üzerinde veya altında bir değere sınırlayabilir veya nöronun çıkışını bir aralıkta normalize edebilir. Aktivasyon katmanı, sinir ağının öğrenme sürecinde önemli bir rol oynar ve sinir ağının daha karmaşık ve esnek bir model haline gelmesini sağlar. Şekil 3.7’de aktivasyon fonksiyonuna gönderilmiş bir verinin girdileri ve çıktılarının grafiksel olarak karşılaştırılması gösterilmiştir.



Şekil 3.7. Aktivasyon Fonksiyonlarının Karşılaştırmalı Olarak Gösterimi

3.2.4. Ortaklama Katmanı (Average Pooling):

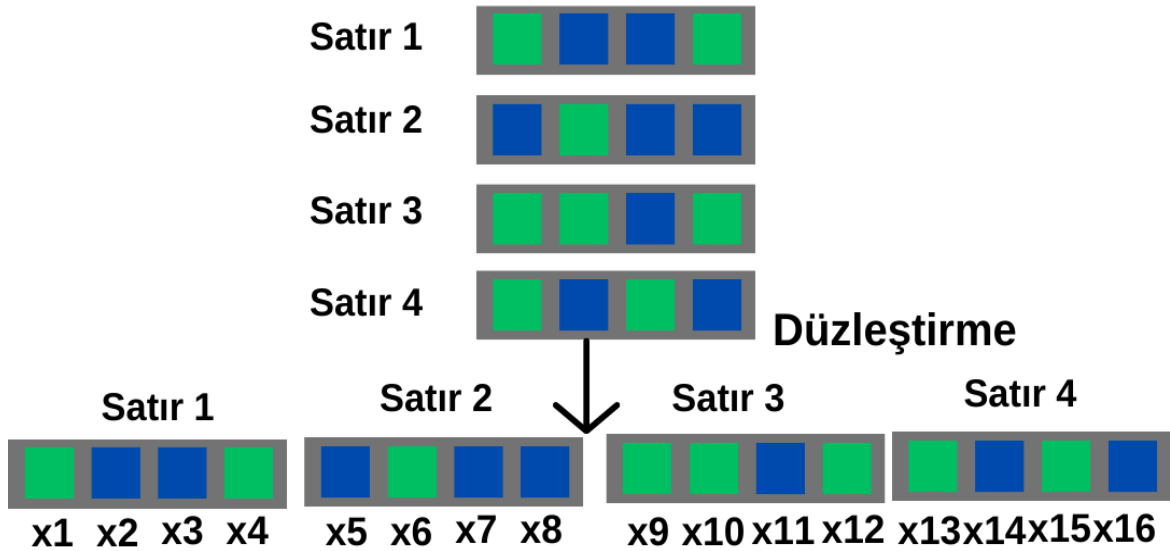
Ortaklama katmanının görevi, ağdaki çok çeşitli parametrelerin sayısını azaltmak için görüntünün uzamsal boyutunu kademeli olarak düşürmektir. Ortaklama katmanı, belirli bir bölge içindeki özelliklerin maksimum veya ortalama değerlerini hesaplayarak çalışır. CNN modellerinde genellikle kullanılan iki farklı ortaklama tekniği vardır. Bunlardan biri Maksimum diğeri de Ortalama ortaklamadır. Aşağıdaki örnekte 2x2 boyutunda bir çerçeve, 4x4 boyutunda bir resme uygulanmıştır (2'ser adım atlamış, 2x2 stride) ve çerçeve içindeki ortalama değer alınmıştır. Bu işlem tüm görsel uygulandıktan sonra daha küçük (2x2 boyutunda) bir görsel elde edilmiştir.



Şekil 3.8. Maksimum Ortaklama ve Ortalama Ortaklama İşleminin Gösterimi

3.2.5. Düzleştirme Katmanı (Flattening Katmanı):

CNN’de iki boyutlu veya üç boyutlu matris şeklinde elde edilmiş olan görüntüyü sınıflandırmak için tek boyutlu dizi haline getirilmesi gerekmektedir. Bu problemin çözülmesi için genellikle düzleştirme katmanı kullanılır. Şekil 3.9’da örnek bir CNN’de düzleştirme işlemi gösterilmiştir.



Şekil 3.9. Düzleştirme İşlemi

3.2.6. Tam Bağlantı Katmanı (Fully Connected Layer):

Tam bağlantı katmanında yapılan işlem düzleştirilmiş değerlerin alınıp ANN’e giriş olarak gönderilmesi işlemidir. Bu işlem sonucunda öğrenme işlemine başlanmış olur.

3.2.7. Batch Normalizasyon Katmanı (Batch Normalization Layer):

Verilerin standartlaştırılması için kullanılan katmandır. Tüm veriye değil de veriyi daha küçük parçalara ayırarak normalleştirme yapılır. Eğitim işleminin hızlandırılması ve yüksek öğrenme oranları kullanmayı sağlayarak öğrenmeyi kolaylaştırır. Genelde evrişim ve aktivasyon fonksiyonu katmanları arasında kullanılmaktadır.

3.2.8. Seyreltme Katmanı (Dropout Layer):

Seyreltme katmanı, öğrenme sırasında öğrenilen nöronların bir kısmının unutulması işlemidir. Modelde aşırı öğrenme sorununun olduğunun düşünüldüğü durumlarda seyreltme katmanı eklenerek bu sorunun önüne geçilir. Genellikle bu oran 0,2-0,5 arasında tercih edilir. Bu sayede öğrenme arttırılmış olur.

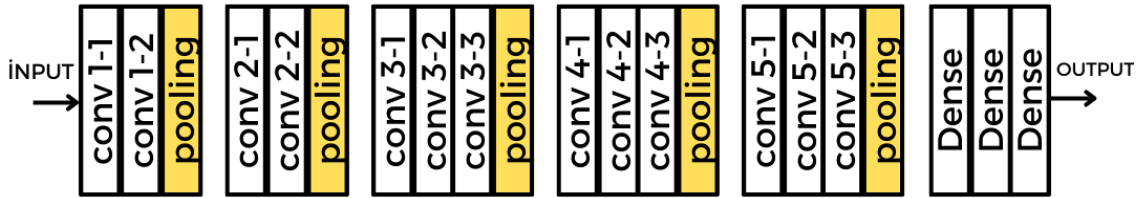
3.3. Evrişimli Sinir Ağı Mimarileri

CNN’de özelleştirilmiş problemlere çözümler üretmek amacıyla farklı mimariler geliştirilmiştir. Bu mimarilerin çoğu ImageNet veri seti kullanılarak eğitilmiş ve çeşitli sınıflandırma çalışmalarında kullanılmıştır. Yüksek doğruluk ve önceden eğitilmiş ağırlıkları kullanması nedeniyle CNN kullanılarak yapılan sınıflandırma çalışmalarında genellikle bu mimariler kullanılmaktadır. Özellikle büyük veri setlerinde eğitim yapılırken bu mimariler tercih edilir. Literatürde yer alan çalışmalar incelendiğinde özellikle öğrenme aktarımı yöntemi ile kullanıldıklarında doğrulukların gözle görülür şekilde arttığı görülmüştür. LeNet, VGG, ResNet, AlexNet, GoogleNet, MobileNet, EfficientNet gibi birçok CNN mimarisi mevcuttur. Bu çalışmada mimariler kullanılarak yapılan çalışmalar incelenmiştir. Ağların en son versiyonları olan sekiz ağ seçilerek eğitim işlemi bu ağlar üzerinde yapılmıştır. Bu mimariler kullanılarak ağlar önce öğrenme aktarımı yöntemi ile herhangi bir katman ekmeden eğitilmiş, ardından ağların doğruluklarının iyileştirilmesi amacıyla belirli katmanlar eklenmiştir. Ağlar tekrar öğrenme aktarımı yöntemi ile eğitilerek doğruluk değerleri iyileştirilmiştir.

3.3.1. VGG-16 (Visual Geometry Group-16) Mimarisi

VGG-16, Oxford Üniversitesi tarafından ImageNet Büyük Ölçekli Görsel Tanıma Yarışması (ImageNet Large Scale Visual Recognition Challenge – ILSVRC)’da yüksek performans gösteren bir CNN mimarisidir. 13 evrişim, 3 adet tam bağlantı katmanı olmak üzere toplam 16 katmandan oluşur. Giriş görüntüleri 224x224x3 boyutlarındadır. Özellikle basit katman yapısına sahip olmasından dolayı, görüntü tanıma, nesne sınıflandırma gibi bilgisayarlı görü uygulamalarında sıklıkla kullanılmaktadır. Model ardışık olarak yerleştirilmiş evrişim ve bunları takiben maksimum havuzlama katmanlarından oluşturulmuştur. 2014 yılında yapılan yarışmada Karen Simonyan ve

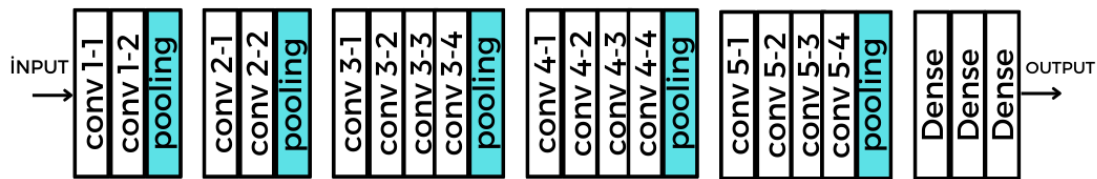
Andrew Zisserman VGG-16 iki farklı görevde en yüksek sırayı elde ederek 1000 sınıfa ait ImageNet veri setinde %92,7 doğruluk elde ederek başarısını ispatlamıştır (Simonyan and Zisserman, 2014). Şekilde 3.10'da VGG-16 ağıının hangi katmanlardan meydana geldiği gösterilmiştir.



Şekil 3.10. VGG-16 Ağıının Katman Yapısı

3.3.2. VGG-19 (Visual Geometry Group-19) Mimarisi

VGG-19, yine Oxford Üniversitesi tarafından geliştirilen VGG-16 ile benzer fakat daha derin bir versiyonudur. Aynı prensiplere dayanmaktadır. 16 adet evrişim ve 3 adet tam bağlantı katmanlarından meydana gelmektedir. Giriş görüntüsü 224x224x3 boyutlarındadır. Katman sayısının artması sayesinde daha karmaşık özelliklerin çıkarılmasını ve daha yüksek doğruluk elde edilmesini sağlar. 2015 yılında Karen Simonyan ve Andrew Zisserman yine ILSVRC yarışmasına yine birincilik sonucu elde etmişlerdir (Simonyan and Zisserman, 2014). Şekil 3.11'de VGG-19 ağıının hangi katmanlardan meydana geldiği gösterilmiştir.



Şekil 3.11. VGG-19 Ağıının Katman Yapısı

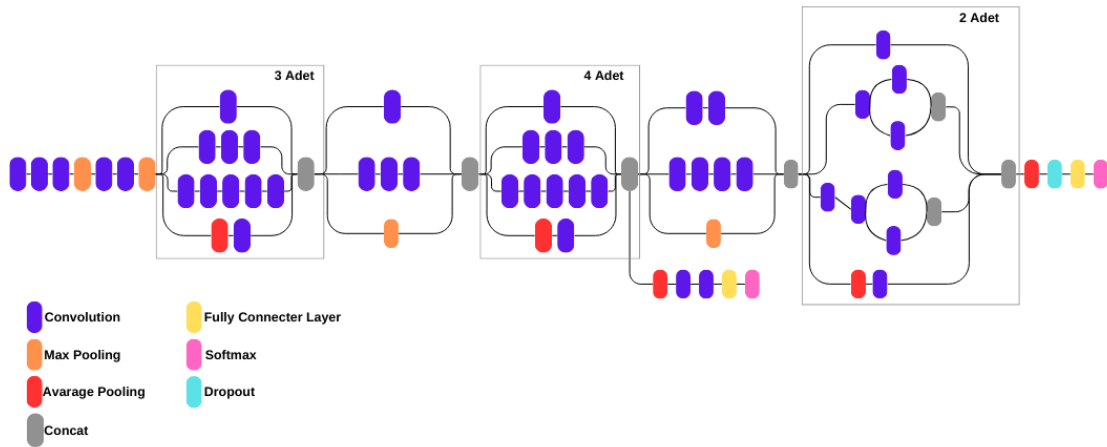
3.3.3. Xception (Extreme Inception) Mimarisi

Xception, Google tarafından 2017 yılında geliştirilen bir CNN mimarisidir. Xception, "Inception" mimarisinin geliştirilmiş bir versiyonudur ve özellikle görüntü sınıflandırma ve nesne tanıma gibi görevlerde yüksek performans sağlamak için tasarlanmıştır. Geleneksel evrişim katmanlarında hem derinlik hem de uzay boyutları

üzerinde evrişim işlemleri yapılırken, Xception'da her bir boyutta ayrı bir evrişim işlemi gerçekleştirilir. Xception, inception modellerine göre daha hafif ve daha verimli bir mimariye sahiptir. 350 milyon görüntüden ve 17.000 sınıftan oluşan eğitim sonucunda %94,5 doğruluk elde edilmiştir (Chollet, 2016).

3.3.4. InceptionV3 (Inception Version 3) Mimarisi

2014 yılından beri önerilmeye başlanmış olan InceptionV3, Google tarafından geliştirilen ILSVRC yarışmasında %96,5 gibi yüksek bir doğruluk elde etmiştir. 25 milyar gibi bir parametre sayısı mevcuttur. Bu parametre sayısına diğer mimarilere göre daha düşük parametre ile daha yüksek doğruluk elde edilmiştir (Szegedy et al., 2015).



Şekil 3.12. InceptionV3 Ağının Katman Yapısı

3.3.5. NASNetLarge (Neural Architecture Search Network) Mimarisi

NASNetLarge, Google tarafından geliştirilen bir CNN mimarisidir. ImageNet yarışmasında %96,2 gibi yüksek bir doğruluk elde etmiştir. Diğer NASNet'e oranla daha fazla parametre içermesi sebebiyle büyük veri kümelerinde yüksek doğruluk elde etmek amacıyla sıklıkla kullanılmıştır. 533 katman ve 88.9 milyon parametre içermektedir (Zoph et al., 2017).

3.3.6. InceptionResNetV2 (Inception Residual Network Version 2) Mimarisi

InceptionResNetV2, Christian Szegedy ve arkadaşları tarafından Inception ve ResNet'in birleşimi olarak geliştirilmiş bir mimaridir. Bu mimari, her iki mimarinin

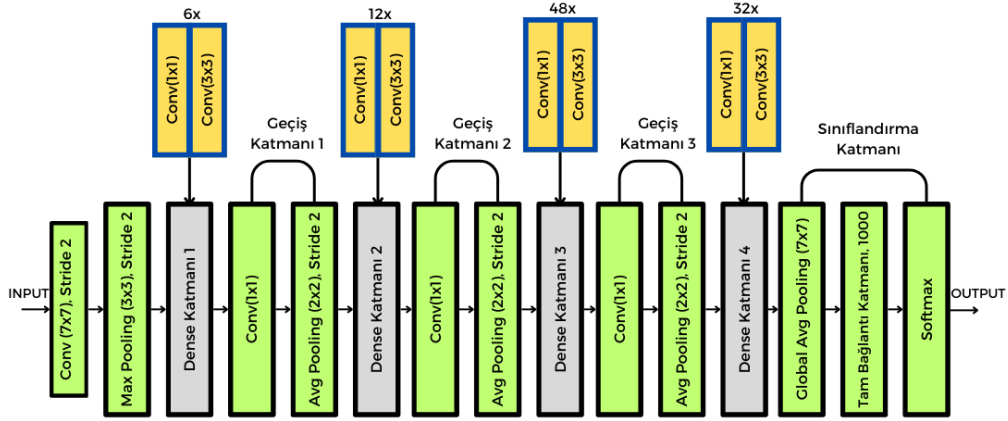
avantajlarını bir araya getirerek yeni ve daha güçlü bir yapı sunar. 2015 ILSVRC yarışmasında InceptionV3 ile benzer sonuçlar elde etmiştir. Fakat ardından ILSVRC 2012 görevinde üç rezidüel ve bir Inception-v4 birleşimi yaparak geliştirilmiş bu mimari ile ImageNet sınıflandırmasında %96,92 gibi yüksek bir doğruluk elde etmişlerdir (Szegedy et al., 2016).

3.3.7. ResNet152V2 (Residual Network 152 Version 2) Mimarisi

ResNet152V2, Microsoft Research tarafından geliştirilen bir CNN mimarisidir. ResNet mimarisinin bir varyantı olan bu model, "Residual Learning" adı verilen bir teknik kullanarak derin ağlarda daha iyi performans elde etmeyi hedefler. ResNet152V2, 152 katmandan oluşan çok derin bir ağıdır. Bu derinlik, modelin karmaşık özellikleri öğrenme kapasitesini artırır ve daha genellemeci bir model elde etmeye yardımcı olur. ImageNet ile yapılan sınıflandırma sonucunda %94,5 gibi yüksek bir doğruluk elde edilmiştir (He et al., 2016)

3.3.8. DenseNet201 (Dense Convolutional Network 201) Mimarisi

DenseNet201, orijinal DenseNet mimarisinin bir varyantıdır ve 201 katmandan oluşan bir derin öğrenme modelidir. Bu model, CIFAR-10, CIFAR-100, SVHN ve ImageNet gibi büyük veri kümeleri üzerinde eğitilmiş ve nesne sınıflandırma görevlerinde yüksek doğruluk elde etmiştir. Özellikle transfer öğrenme uygulamalarına oldukça başarılı sonuçlar vermektedir (Huang et al., 2016). Şekil 3.13'de DenseNet201 ağının katman yapısı gösterilmiştir. Şekilden de anlaşılacağı üzere Evrişim, maksimum havuzlama katmanı, çok katmanlı bir yapıya sahip Dense katmanı, Ortalama havuzlama katmanı, softmax gibi birçok katman içerecek bir yapıya sahiptir. Ayrıca geçiş katmanları ile öznitelik çıkarım işlemini gerçekleştiren bir yapıya sahiptir.



Şekil 3.13. DenseNet201 Ağının Katman Yapısı

Çizelge 3.2’de yapılan tez çalışmasında kullanılan CNN mimarilerine ait boyut (önceden eğitilmiş olan ağırlıkların megabayt cinsinden boyutu), En yüksek doğruluk (ImageNet ile eğitilmiş olan ağların elde etmiş oldukları en yüksek doğruluk değeri), parametre sayısı (ağırlık ve bias değerleri ile çarpılarak elde edilmiş olan parametre sayısı), derinlik (içerdiği katman sayısı), giriş görüntüsü boyutu (ağa girdi olarak verilecek görüntünün boyutları) verilmiştir.

Çizelge 3.2. Çalışmada Kullanılan CNN Mimarileri ve ImageNet Üzerindeki Parametreleri

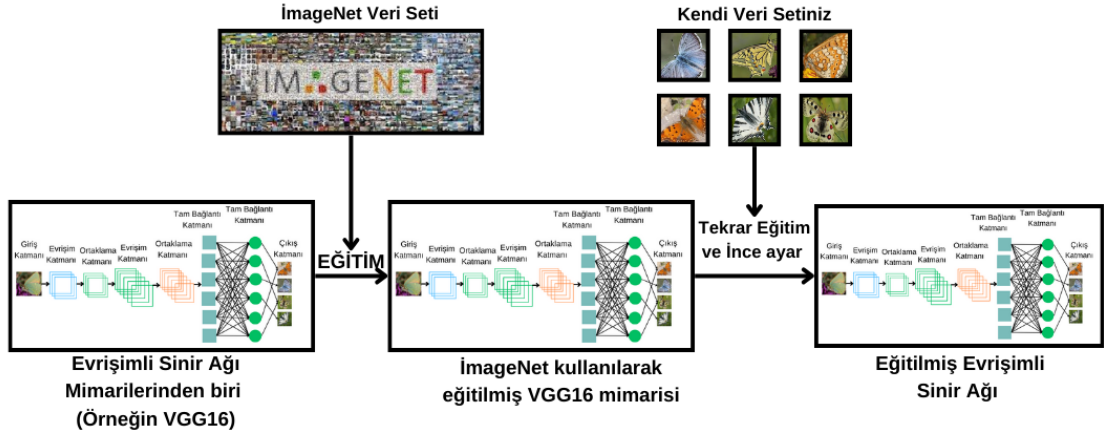
Mimari Özellik	Boyut (Size)	En Yüksek Doğruluk (ImageNet)	Parametre Sayısı	Derinlik	Giriş Görüntüsü Boyutu
VGG16	528	%90,1	138.4 Milyon	16	224x224
VGG19	549	%90,0	143.7 Milyon	19	224x224
Xception	88	%94,5	22.9 Milyon	81	299x299
InceptionV3	92	%93,7	92 Milyon	189	299x299
ResNet152V2	232	%94,2	60.4 Milyon	307	224x224
InceptionResNetV2	215	%95,3	55.9 Milyon	449	299x299
DenseNet201	80	%93,6	20.2 Milyon	402	224x224
NasNetLarge	343	%96,0	88.9 Milyon	533	331x331

Derin öğrenme veya CNN, birçok katmanı olan karmaşık bir yapıya sahiptir ve genellikle büyük miktarda veriyle eğitilirler. Geniş veri setleri ve hesaplama gücü gerektiren karmaşık yapıları nedeniyle genellikle Grafik İşlem Birimi (Graphics Processing Unit- GPU) veya özel donanımlar gibi yüksek performanslı bilgisayar sistemlerinde eğitilir. Yapay sinir ağları temelinden faydalanan üretilen yapay zekâ uygulamalarını eğitmek için kullanılan Tensor İşleme Birimi (Tensor Processing Unit-TPU), Merkezi İşlem Birimi (Central Processing Unit- CPU) ve GPU'lara göre 15 ila 30 kat daha hızlıdır (Kızrak, 2018). İşlemci gücünün yetersiz olduğu durumlarda yapay

bilgisayar ortamlarına ihtiyaç duyulabilir. 2015 yılında Google tarafından ilk TPU merkezi kurulmuştur. Google Colab TPU sayesinde çeşitli işlemciler ile yapay bir bilgisayar üzerinden bağlanarak, Google Drive'a yüklemiş olunan verilerin işlenmesine ve üzerinde eğitim yapılmasına olanak sağlar. 2024 yılı itibarıyla Google Colab üzerinde Python ve R programlama dillerinde eğitim yapabilen yedi farklı yapay donanım hızlandırıcı mevcuttur. Bunlar CPU, A100 GPU, L4 GPU, V100 GPU, T4 GPU, TPU, TPUv2'dir. Ayrıca yüksek ram seçeneği ile ram hızınızı da arttırmanıza olanak sağlar. Yapılan tez çalışmasında Python programlama dili, T4 GPU ve yüksek ram kullanılmıştır. Yapay zekâ ve derin öğrenmede sıklıkla kullanılan Keras kütüphanesi üzerinden ağların ilk ağırlıkları çekilerek eğitim işlemleri yapılmıştır.

3.4 CNN'de Öğrenme Aktarımı (Transfer Learning) Yöntemi Nasıl Uygulanır?

Makine öğrenmesi ve derin öğrenmede sıklıkla kullanılan bir yöntem olan Öğrenme aktarımı, ağın önceden eğitilerek bir problemi çözerken elde ettiği bilgiyi saklayıp daha sonra benzer bir problemle karşılaştığında aynı bilgiyi problemi çözmede kullanmasıdır. Derin öğrenmede bir ağın sıfırdan eğitilmesi çok uzun zaman alan bir süreçtir. Bu yüzden genellikle önceden benzer problemlerde kullanılmış olan ağlar alınarak öğrenme aktarımı yöntemi sıklıkla tercih edilir. Öğrenme aktarımında genellikle ImageNet yarışmasında eğitilmiş ağlar kullanılır. Bu ağlar sayesinde önceden eğitilmiş hazır halde bulunan ağırlıklar alınarak üzerine eğitilmek için kullanılacak veri seti eklenir. Bu yöntem sayesinde veri setinin daha kısa sürede, daha az veri ile ve daha iyi performans elde edilmesi sağlanmış olur. ImageNet veri setiyle önceden eğitilmiş bir modeli ele alalım. Örneğin, VGG16 modeli, önceden eğitilmiş ağırlık bilgilerini içerir ve Keras kütüphanesinde bulunur. VGG16'nın öğrenme aktarımı kullanılarak, tez çalışmasında kullanılan veri seti eğitilebilir. İstenildiğinde katmanlar eklenebilir, ince ayarlar yapılabilir ve hiper parametreler ayarlanabilir. Bu şekilde, modelin başarılı sonuçlar elde etmesi sağlanabilir.



Şekil 3.14. Öğrenme Aktarımı ve İnce Ayar

Öğrenme aktarımı yaparken dikkat edilmesi gereken adımlar vardır. Bu adımlar şu şekilde sıralanabilir:

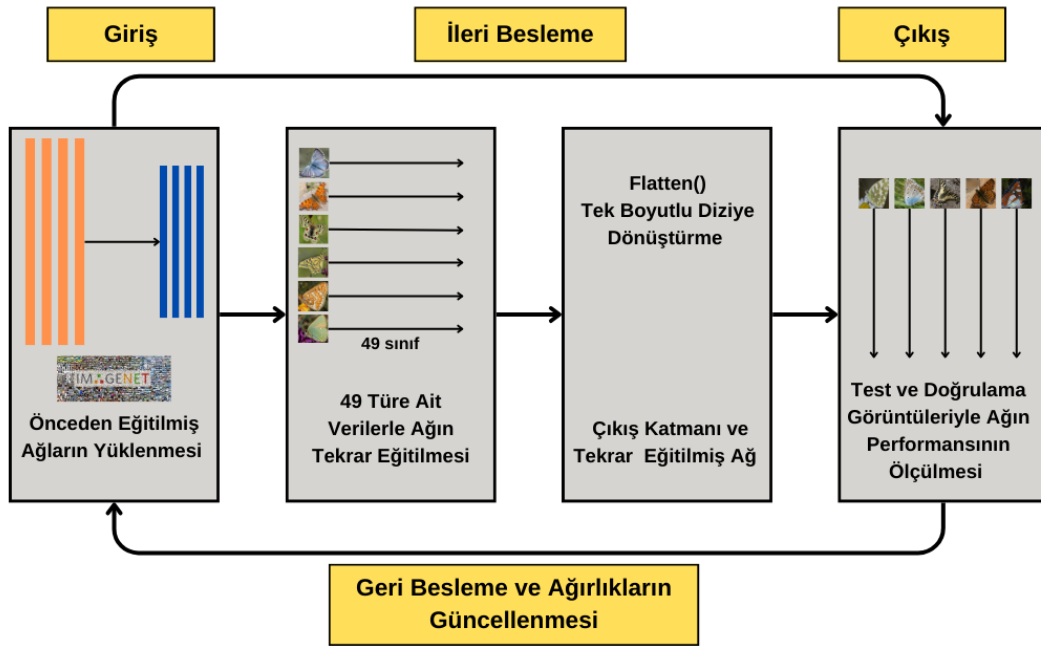
- 1) Gerçekleştirilecek göreve göre önceden eğitilmiş ağların belirlenmesi. Örneğin bu tez çalışmasında VGG16, VGG19, DenseNet201, ResNet152v2, NasNetLarge, Inceptionv3, InceptionResNetv2, Xception gibi genellikle görüntü sınıflandırmada kullanılan ağlar tercih edilmiştir.
- 2) Modelin oluşturulması. Bu seçilen ağların önceden ImageNet kullanılarak eğitilmiş ağırlıklarının çekilmesi sayesinde ağın daha kısa sürede eğitilmesi sağlanmıştır. Aksi takdirde çok daha uzun süren eğitim süreleri ile karşılaşmak kaçınılmaz olacaktır. Son katmanlarda ImageNet 1000 sınıflı bir eğitici olduğu için o sınıflandırma katmanı kaldırılarak 49 sınıflı bir sınıflandırma katmanı eklenmiştir.
- 3) Katmanların dondurulması. Öğrenme aktarımı yapılırken sıklıkla kullanılan yöntemlerden biridir. CNN’de genellikle giriş ve ara katmanlar kenar, doku, renk, desen gibi üst seviye özellikleri öğrenirken son katmanlar daha spesifik özellikleri öğrenir. Bundan dolayı genellikle çıkış hariç geri kalan katmanlar dondurularak eğitimin hızlanması sağlanır (Brownlee, 2018).
- 4) Eğitilebilir yeni katmanlar ekleyerek modelin iyileştirilmesi. Modeli yeniden eğitirken özellik çıkarımını yapacak olan katmanlar yeni eklenecek katmanlardır.
- 5) Son olarak, katmanları eklenen model eğitilir ve farklı parametreler denenerek hiper parametre ayarlamaları yapılır, böylece modelde iyileştirmeler sağlanır (Şener, 2023).

Tez çalışmasında iki farklı yöntem uygulanmıştır. Bu yöntemlerden ilki doğrudan CNN mimarilerini kullanarak herhangi bir katman eklemeyen girişten alınan görüntüleri eğiterek 49 sınıflı sınıflandırıcıya gönderilmiştir. İkinci yöntem olarak mimariler iyileştirme amacıyla belirli katmanla eklenerek mimarinin öğrenecek parametre sayısı artırılmış ve doğruluğun artırılması amaçlanmıştır.

3.5 Katman Eklemeyen Yapılan Öğrenme Aktarımı Yönteminin Uygulanması

- 1) Yapılan tez çalışmasında, ilk yöntem olarak, herhangi bir katman eklenmeden doğrudan ImageNet'ten önceden eğitilmiş ağlardan ilk ağırlık değerleri çekilmiştir.
- 2) Ardından öğrenme aktarımı yöntemi uygulanmıştır. Katman olarak sadece ağların içinde yer alan katmanlar kullanılmış fakat öğrenmenin hızlı olması için katmanlar dondurulmuştur. Modelin tekrar baştan sona eğitimi hem fazla süre hem de gereksiz yere parametre arttıracığından dolayı bu yöntem tercih edilmemiştir.
- 3) Eğitim için kullanılacak verilerin ilk olarak giriş görüntüleri kullanılan ağ için ideal olan Keras kütüphanesinde önerilen boyutlara yeniden boyutlandırılmıştır. Ardından çalışmada kullanılacak girdi olarak bu görüntüler gönderilmiştir.
- 4) Eğitilmiş ağın çıkışına düzleştirme katmanı eklenerek çıkış tek boyuta indirgenmiştir. Son olarak ağa 49 sınıflı “softmax aktivasyon fonksiyonu” kullanan çıkış katmanı eklenerek modelin katman yapısı sonlandırılmıştır.
- 5) Eğitim işleminde ağların doğrulukları karşılaştırılmıştır.

Şekil 3.15’de ilk yöntem kullanılarak tasarlanmış bir ağın yapısının nasıl olacağı gösterilmiştir. Tüm ağlar bu yöntem ile eğitilmiştir.



Şekil 3.15. Katman Eklemeden Yapılan Öğrenme Aktarımı Yönteminin CNN Ağ Yapısı

Ağın eğitimi sırasında kullanılan hiper parametreler ise Çizelge 3.3’de verilmiştir. Bu parametreler belirlenirken benzer çalışmalarda sıklıkla tercih edilen parametreler tercih edilmiştir.

- 1) Batch boyutu genellikle 2, 4, 8, 16, 32, 64, 128, 256, 512 gibi iki ve ikinin katları olarak tercih edilmektedir. Bu parametre modelin dahili parametrelerini güncellemeden önce üzerinde çalışacağı eğitim örnekleminin sayısını ifade eder. Genellikle işlemcinin 32 veya 64 gibi değerler tercih edilir. Bu çalışmada eğitim süresinin de artacağı göz önünde bulundurularak 32 değeri tercih edilmiştir.
- 2) Öğrenme oranı ağın eğitilmesi sırasında geriye yayılım yaparken türev alınarak bulunan fark değeriyle çarpılır ve bu şekilde yeni ağırlık değeri hesaplanır. Bu işlem eğitim süresince tekrar edilir. Genellikle 0.01 ile 0.0001 arasında tercih edilir. Bu çalışmada eğitimin tekrar sayısı fazla tutulacağından öğrenme oranı 0.0001 gibi düşük bir değer olarak tercih edilmiştir.
- 3) Optimizasyon algoritmaları doğrusal olmayan problemlerde optimum çözümün bulunması için kullanılırlar. Genellikle stochastic gradient descent, adagrad, adadelta, adam, adamax gibi farklı optimizasyon algoritmaları çalışmadaki probleme göre kullanılmaktadır. Literatürde bu tarz sınıflandırma problemlerinde genellikle Adam optimizasyon algoritması tercih edilmiştir. Bu yüzden bu çalışmada da Adam optimizasyon algoritması kullanılmıştır.

- 4) Eğitim tur sayısı (Epoch), ileri ve geri yayılım sonucunda her bir eğitim sonucunda modelin ağırlıklarının güncellenmesidir. Bu işlemin kaç kere tekrar edileceğini belirten sayıya eğitim tur sayısı (Epoch) denir. Zaten çalışmada transfer öğrenme yapıldığı ve öğrenmenin belirli bir süre sonra durağan hale geldiği göz önünde bulundurularak epoch sayısı 50 olarak belirlenmiştir.
- 5) Aktivasyon fonksiyonları doğrusal olmayan dönüşüm işlemlerinde kullanılırlar. Genellikle sınıflandırma çalışmalarında aktivasyon fonksiyonu olarak softmax tercih edildiği için bu çalışmada da softmax tercih edilmiştir. Bunun nedeni ise giriş ne olursa olsun 0 ile 1 arasında çıkış üretmesidir.
- 6) Patience modelin performansında iyileşme olup olmadığını kontrol eder. Bu çalışmada beş adım üst üste modelde iyileşme olmadığında modelin eğitimi durdurması böylece aşırı öğrenmenin önüne geçilmesi sağlanmıştır.
- 7) Loss fonksiyonu regresyon, sınıflandırma veya çoklu sınıflandırma problemlerinde modelin veri üzerindeki başarısını ölçmek için kullanılır. Çoklu sınıflandırmada genellikle categorical_crossentropy kullanıldığı için bu çalışmada da bu yöntem tercih edilmiştir.

Çizelge 3.3. Katman Eklemeden Yapılan Öğrenme Aktarımında Kullanılan Hiper Parametreler

Batch Boyutu	32
Learning Rate (Öğrenme Oranı)	0.0001
Optimizasyon Algoritması	Adam
Eğitim Tur Sayısı (Epoch)	50
Aktivasyon Fonksiyonu	Softmax
Patience	5
Loss Fonksiyonu	Categorical_crossentropy

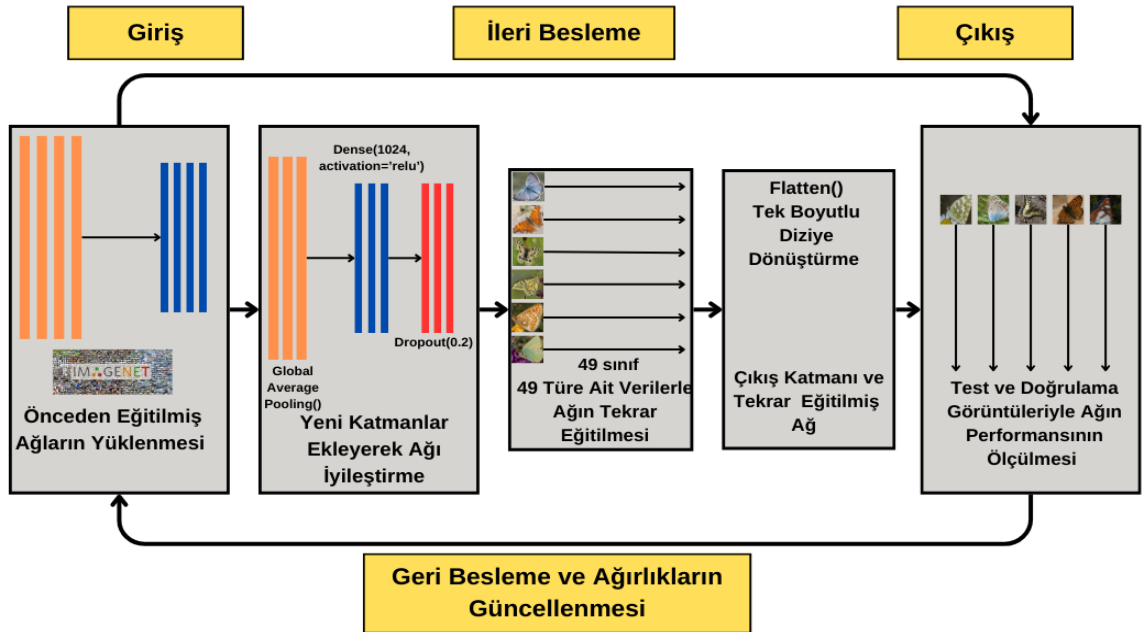
3.6 Katman Eklenerek Yapılan Öğrenme Aktarımı Yönteminin Uygulanması

İlk yöntem sonucunda bu veri setinde özellikle doğrulama ve test veri seti üzerinde değerlendirme kriterlerinin performanslarının literatürdeki diğer çalışmalara göre daha düşük kaldığı görülmüştür. Bu nedenle ağların iyileştirilmesine ve performanslarının artırılmasına yönelik çalışmalar yapılmıştır. Bu çalışmalar mevcut ağın üzerine yeni katmanlar eklenerek yapılmıştır. Şekil 3.16'da yeni yöntemin katman yapısı gösterilmiştir. Bu yöntem adım adım şu şekilde açıklanabilir.

- 1) İlk adımda ilk yöntemde olduğu gibi giriş görüntüleri kullanılan ağın önerilen giriş boyutlarına göre tekrar boyutlandırılarak veri girişi ağa gönderilmiştir. Ağa

gelen görüntüler evrişim ve tam bağlantı katmanlarından geçerek özellik çıkarımı yapılır.

- 2) Ardından ağdan çıkan çıkışlar Global Ortalama Havuzlama katmanına gönderilir. Bu katmanda her bir özellik haritasındaki tüm değerlerin ortalamasını alır ve bu şekilde her özellik haritasını tek bir değere indirir. Bu katman modelin aşırı öğrenmeye kaçmasını engelleyici bir katmandır.
- 3) Ardından 1024 katmanlı aktivasyon fonksiyonu Relu olan bir dense katmanı mevcuttur. Bu katman sayesinde negatif değerleri sıfıra çeker ve pozitif değerleri olduğu gibi bırakır. Bu, modelin doğrusal olmayan özellikleri öğrenmesini sağlar.
- 4) Daha sonra dropout katmanı eklenmiştir. Bu katman sayesinde öğrenilmiş nöronların %20 sini serbest bırakır. Dropout, bir regularizasyon tekniğidir ve aşırı öğrenmeyi (overfitting) önlemek için kullanılmıştır.
- 5) Son olarak eğitim sonunda yine 49 sınıflı çıktı katmanının bulunduğu ve aktivasyon fonksiyonunun softmax olduğu modelin her bir sınıfa ait olma olasılığını tahmin etmesini sağlayan sınılandırma katmanı eklenmiştir. Bu katman sonunda her sınıf için olasılık değeri üretilmiş olur. İkinci yöntemde ağın katman yapısı bu şekilde tamamlanmış olur.



Şekil 3.16. Katman Eklenerek Yapılan Öğrenme Aktarımı Yönteminin CNN Ağ Yapısı

Ağın eğitiminde kullanılan hiper parametreler yine birinci yöntemde olduğu gibi bırakılmıştır. Bu yüzden hiper parametre ile ilgili tekrar tanımlama gösterilmemiştir.

4. ARAŞTIRMA SONUÇLARI VE TARTIŞMA

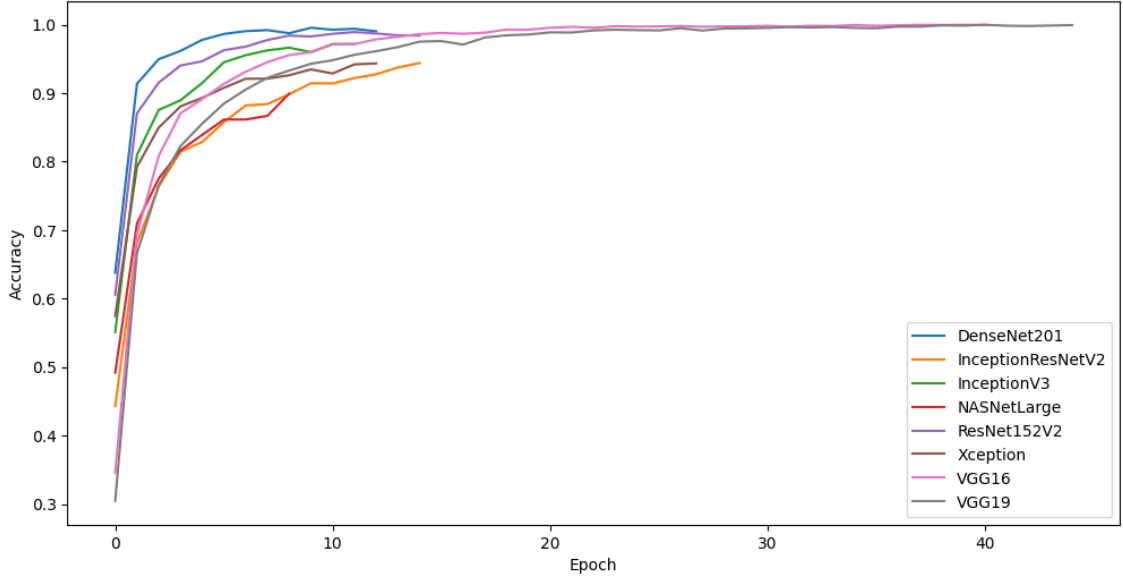
4.1. Uygulama Sonucunda Elde Edilen Sonuçlar ve Sonuçların Karşılaştırılması

Bu bölümde katman eklemeyen yapılan öğrenme aktarımı yöntemi ve katman eklenerek yapılan öğrenme aktarımı yöntemi sonucunda elde edilen ağların performansları incelenmiştir. Ardından iki yöntemin birbiriyle karşılaştırması yapılmıştır. Ağların artı ve eksi yönleri paylaşılmıştır.

4.1.1. Katman Eklemeyen Yapılan Öğrenme Aktarımında Elde Edilen Sonuçlar

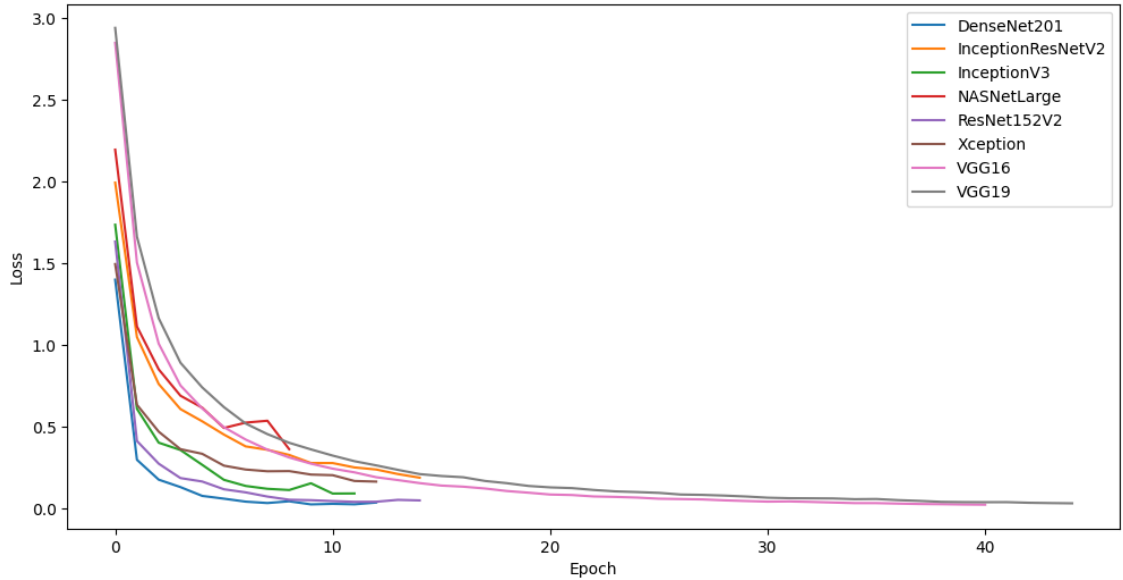
Katman eklemeyen yapılam öğrenme aktarımında elde edilen sonuçlara göre eğitim için ayrılan veri üzerindeki başarısı (accuracy) ve kayıp (loss) eğitim için ayrılan veri seti üzerindeki performans, doğrulama başarısı (validation_accuracy) ve doğrulama kaybı (validation_loss) eğitim esnasında modelin hiper parametrelerinin performansını değerlendirmek için kullanılan doğrulama veri seti üzerindeki performans, test başarısı (test_accuracy) ve test kaybı (test_loss) modelin eğitildikten sonra hiç görmediği veri seti üzerindeki performansı incelenmiştir. Ayrıca diğer değerlendirme kriterlerinden kesinlik (precision), duyarlılık (recall), f1-skoru (f1-score), hata oranı ve karmaşıklık matrisi (confusion matrix)'in iki yöntemde elde edilen en iyi ağ performanslarının karşılaştırılması da yapılmıştır.

İlk olarak Şekil 4.1'de eğitim sonucunda elde edilen veriler incelendiğinde katmansız öğrenme aktarımı yönteminde elde edilen doğruluk değerleri ağların hepsinin öğrenme yaptığını göstermektedir. DenseNet201'in diğer ağlara göre daha hızlı bir öğrenme yaptığı, NasNetLarge'ın öğrenmede daha yavaş kaldığı görülmektedir. Şekil 4.3 incelendiğinde bazı ağlarda öğrenme kriteri olan validation_loss düşmediği için öğrenmenin erkenden durdurulduğu gözlenmektedir. Bunun uygulanmasının sebebi ise modelin aşırı öğrenmeye kaçmasının önüne geçmektir.



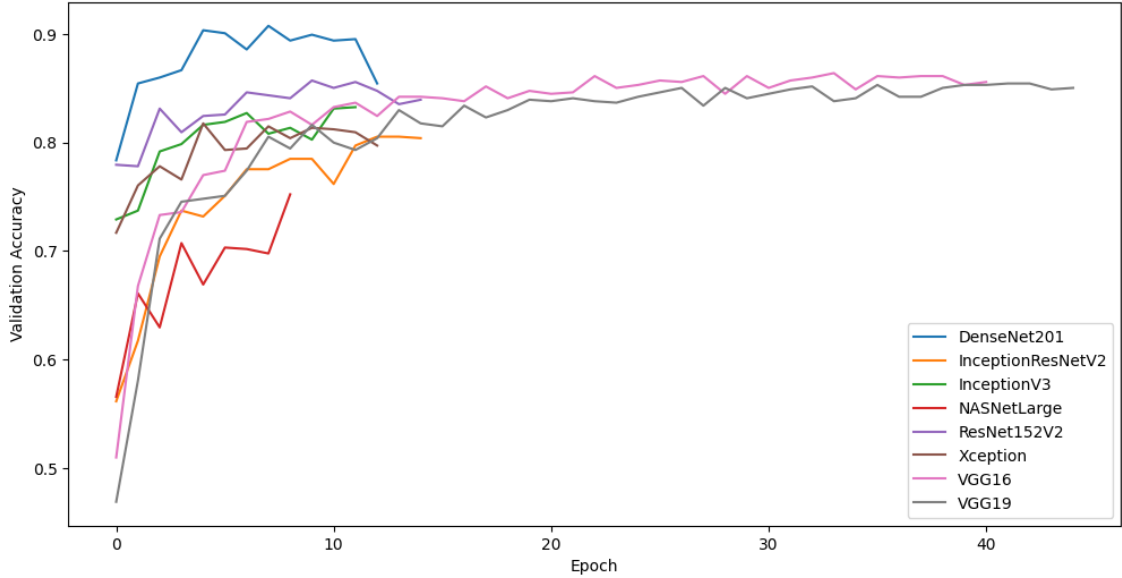
Şekil 4.1. Katman Eklemeden Yapılan Öğrenme Aktarımında Elde Edilen Eğitim Verileri Üzerindeki Başarı Grafiği

Yine birinci yöntem sonucunda elde edilen kayıp parametresinin Şekil 4.2’de yer alan grafiği incelendiğinde ise DenseNet201 ağının eğitim verileri üzerindeki loss değeri en hızlı düşmekte, VGG19 ağının ise en yavaş düşmekte olduğu görülmektedir. Bu da eğitim verileri üzerinde DenseNet201 ağının daha hızlı başarı elde ettiğinin göstergesidir.



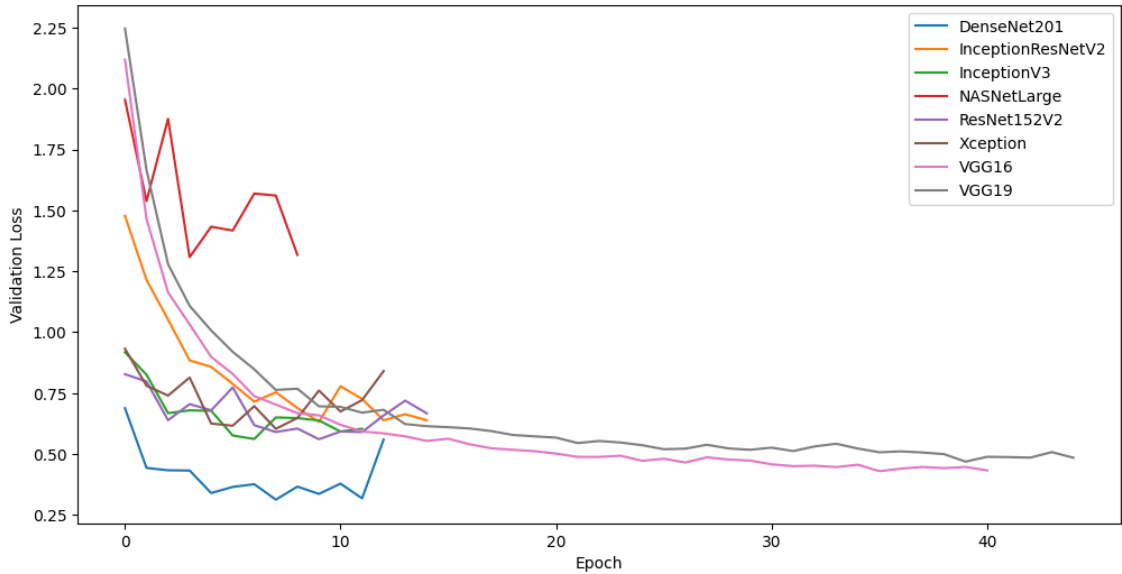
Şekil 4.2. Katman Eklemeden Yapılan Öğrenme Aktarımında Elde Edilen Eğitim Verileri Üzerindeki Kayıp Grafiği

Birinci yöntem sonucunda elde edilen doğrulama verileri üzerindeki ağların başarısı Şekil 4.3’de incelendiğinde en iyi performansı DenseNet201 ağında, en kötü performansın ise NasNetLarge ağında verdiği gözlenmiştir.



Şekil 4.3. Katman Eklemeden Yapılan Öğrenme Aktarımında Elde Edilen Doğrulama Verileri Üzerindeki Başarı Grafiği

Doğrulama verileri üzerindeki kaybı Şekil 4.4’de incelendiğinde en iyi sonucun DenseNet201’de en başarısız sonucun ise NasNetLarge ağında elde edildiği gözlenmiştir.



Şekil 4.4. Katman Eklemeden Yapılan Öğrenme Aktarımında Elde Edilen Doğrulama Verileri Üzerindeki Kayıp Grafiği

Üst bölümde grafiksel olarak gösterilmiş sonuçlar aşağıda yer alan Çizelge 4.1’de ağların eğitim, doğrulama ve test verileri üzerindeki en başarılı performansları gösterilmiştir. Bu verilere göre sekizinci eğitim sonucunda doğrulama doğruluğunda elde ettiği %90,73’lük ve test doğruluğunda elde ettiği %91,09’luk başarı oranıyla en başarılı ağın DenseNet201 ağı olduğu gözlenmiştir. Sekizinci eğitim sonunda en başarılı sonuçlara ulaşan DenseNet201 ağının hem en hızlı hem de en başarılı ağ olduğu gözlenmiştir. En başarısız ağın ise NasNetLarge ağı olduğu gözlenmiştir. Bu veriler dikkate alınırken en yüksek doğruluk ve en düşük kayıpların toplamının yüzdesel olarak %100 değerini geçmesinin sebebi ise sonuçların normalize edilmemiş olmasıdır. Hata oranı olarak verilmiş olan Çizelge 4.2’de yer alan veri normalize edilmiş olarak verilmiştir. Bu veriler dikkate alınabilir.

Çizelge 4.1. Katman Eklemeyen Yapılan Öğrenme Yönteminde Elde Edilen Başarı ve Kayıp Değerleri

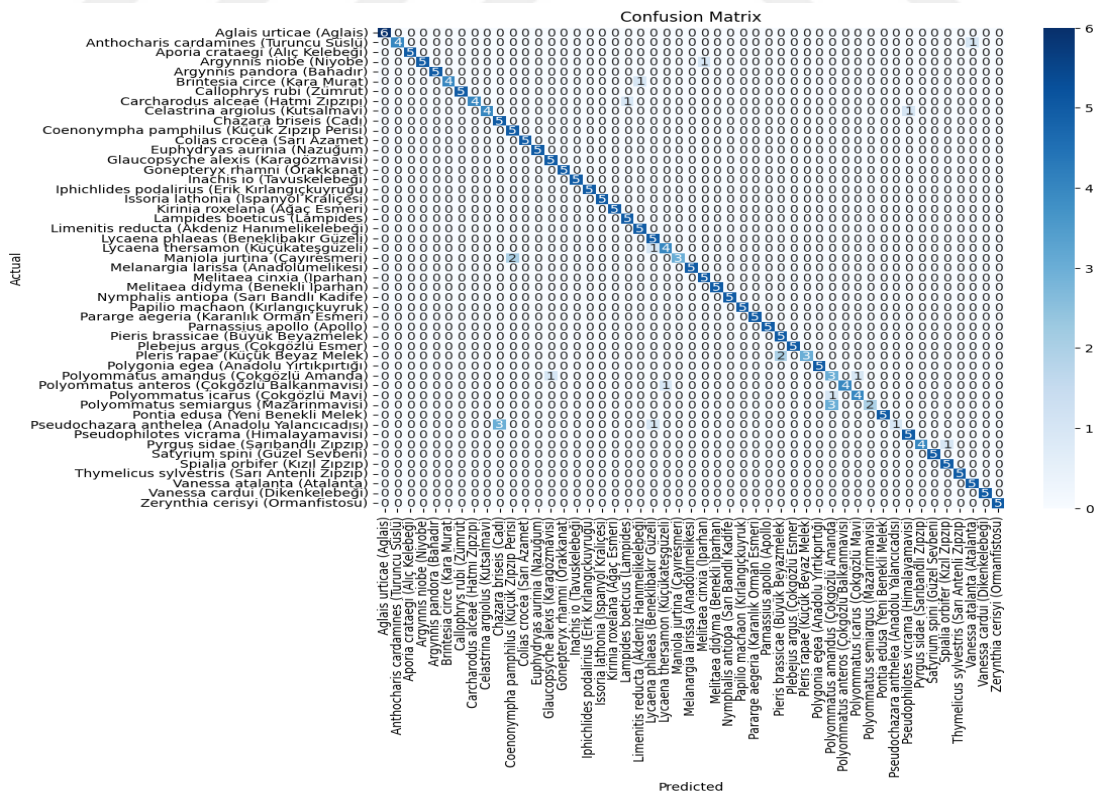
Ağ\Değerlendirme Kriteri	Eğitim Başarısı	Eğitim Kaybı	Doğrulama Başarısı	Doğrulama Kaybı	Test Başarısı	Test Kaybı
VGG16	0,9982	0,0330	0,8610	0,4294	0,8785	0,4625
VGG19	0,9987	0,0392	0,8528	0,4687	0,8623	0,5365
InceptionV3	0,9554	0,1379	0,8269	0,5616	0,8219	0,6933
Xception	0,9211	0,2277	0,8147	0,6032	0,8057	0,8179
DenseNet201	0,9920	0,0334	0,9073	0,3125	0,9109	0,3167
NasNetLarge	0,8164	0,6898	0,7070	1,3080	0,7166	1,7558
ResNet152V2	0,9931	0,0518	0,9046	0,2703	0,8462	0,6578
InceptionResNetV2	0,9144	0,2782	0,7847	0,6317	0,7733	0,8913

İlk yöntem sonucunda elde edilen kesinlik (precision), duyarlılık (recall), f1-skoru (f1-score) performansları ise Çizelge 4.2’de veri seti üzerindeki ortalamaları alınarak tablo halinde verilmiştir. Çizelgede en iyi sonuçlar sarı en başarısız sonuçlar kırmızı renkte gösterilmiştir. Bu sonuçlar incelendiğinde kesinlik parametresinde en başarılı sonucun DenseNet201’de en başarısız sonucun ise NasNetLarge ağına elde edildiği, duyarlılık parametresi incelendiğinde en başarılı sonucun DenseNet201’de en başarısız sonucun NasNetLarge’da, f1 skoru incelendiğinde ise en başarılı sonucun DenseNet201’de en başarısız sonucun yine NasNetLarge ağına elde edildiği gözlemlenmiştir.

Çizelge 4.2. Katman Eklemeden Yapılan Öğrenme Yönteminde Ağlarda Elde Edilen Değerlendirme Kriterlerinin Karşılaştırılması

Ağ Değerlendirme Kriteri	Kesinlik (Precision)	Duyarlılık (Recall)	F1-Skoru (F1-Score)	Hata Oranı (%)
VGG16	0,90	0,88	0,88	12,15
VGG19	0,89	0,86	0,86	13,77
InceptionV3	0,86	0,82	0,82	17,81
Xception	0,84	0,81	0,80	19,43
DenseNet201	0,93	0,91	0,91	8,91
NasNetLarge	0,75	0,72	0,71	28,34
ResNet152V2	0,88	0,85	0,84	15,38
InceptionResNetV2	0,85	0,77	0,76	22,67

Test görüntülerine uygulanan sınıflandırma sonucunda elde edilen en düşük hata oranı %8,91 ile DenseNet201 ağında 247 test görüntüsünde 225 doğru sınıflandırma yaparak elde edilmiştir. Bu sınıflandırmaya ait karmaşıklık matrisi Şekil 4.5’de verilmiştir. Karmaşıklık matrisinde yer alan verilere göre 49 sınıfın 22’sinde hatalı sınıflandırma yapıldığı gözlenmiştir. En fazla hatalı sınıflandırmanın yapıldığı tür beş görüntüden dördünün hatalı sınıflandırıldığı Pseudochazara anthelea (Anadolu Yalancıcadısı) türü olduğu saptanmıştır. Bu türleri Cadı olarak sınıflandırmıştır. Bu iki tür arasında oldukça fazla benzerlik olduğu için ağı öğrenmeyi doğru yaptığı bilgisi çıkarılabilir.

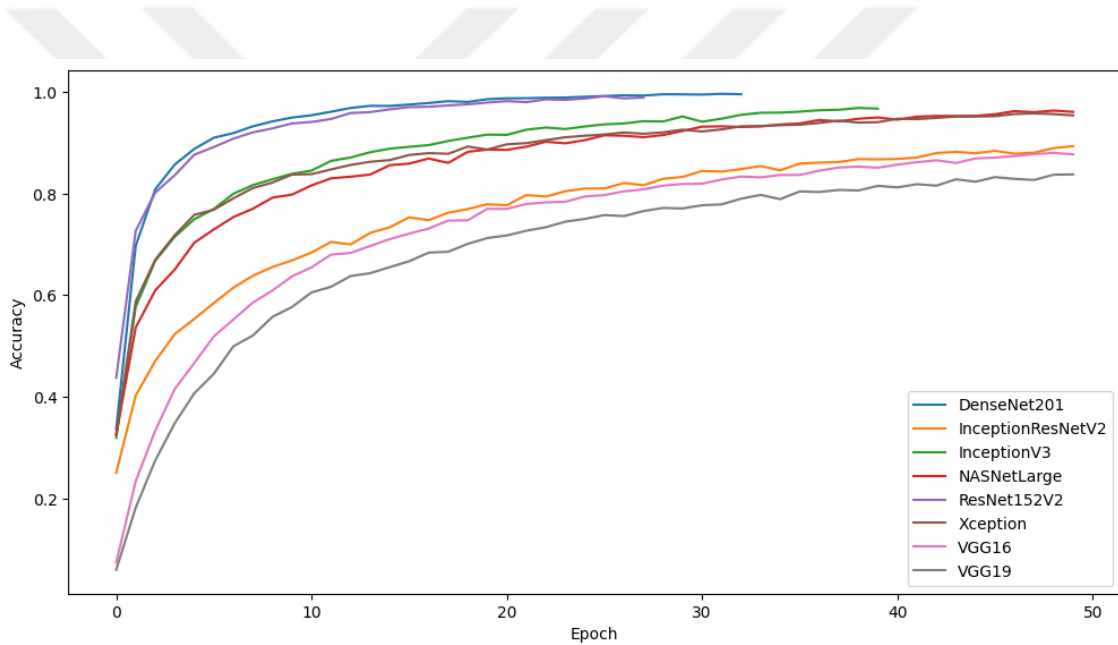


Şekil 4.5. DenseNet201’in Katman Eklemeden Öğrenme Aktarımı Yönteminde Elde Edilen Karmaşıklık Matrisi

4.1.2. Katman Eklenerek Yapılan Öğrenme Aktarımında Elde Edilen Sonuçlar

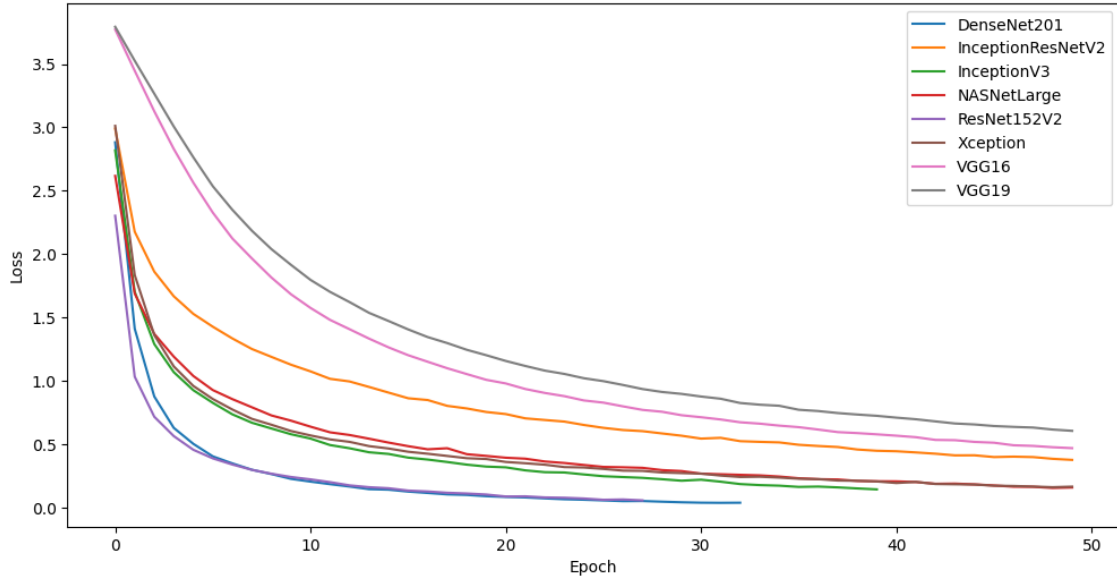
İkinci yöntemde elde edilen sonuçlara göre eğitim verileri üzerindeki başarı ve kayıp performansı, doğrulama verileri üzerindeki başarı ve kaybı, test verileri üzerindeki başarısı ve kaybı modelin eğitildikten sonra hiç görmediği veri seti üzerindeki performansı incelenmiştir. Ayrıca diğer değerlendirme kriterlerinden kesinlik, duyarlılık, fl-skoru, hata oranı ve karmaşıklık matrisi'nin karşılaştırılması da yapılmıştır.

Şekil 4.6'da eğitim verileri üzerindeki başarısı incelendiğinde, birinci yöntemde olduğu gibi, en hızlı ve en yüksek doğruluğa DenseNet201 ağına ulaştığı, en yavaş doğruluk artışının ise VGG19 ağına gözlendiği görülmüştür.



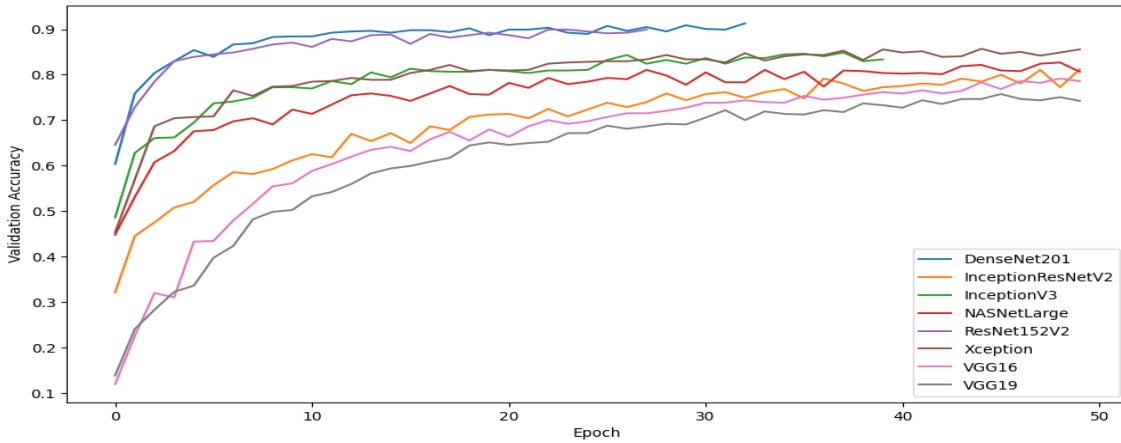
Şekil 4.6. Katman Eklenerek Yapılan Öğrenme Aktarımında Elde Edilen Eğitim Verileri Üzerindeki Başarı Grafiği

Şekil 4.7'de eğitim verileri üzerindeki kayıp grafiği incelendiğinde ise ilk başta ResNet152v2'nin daha keskin bir iniş yaptığı fakat ardından DenseNet201 ağına ResNet152v2 ağını geçtiği gözlenmiştir. En yüksek kayıp değeri ise VGG19 ağına olduğu gözlenmiştir.



Şekil 4.7. Katman Eklenecek Yapılan Öğrenme Aktarımında Elde Edilen Eğitim Verileri Üzerindeki Kayıp Grafiği

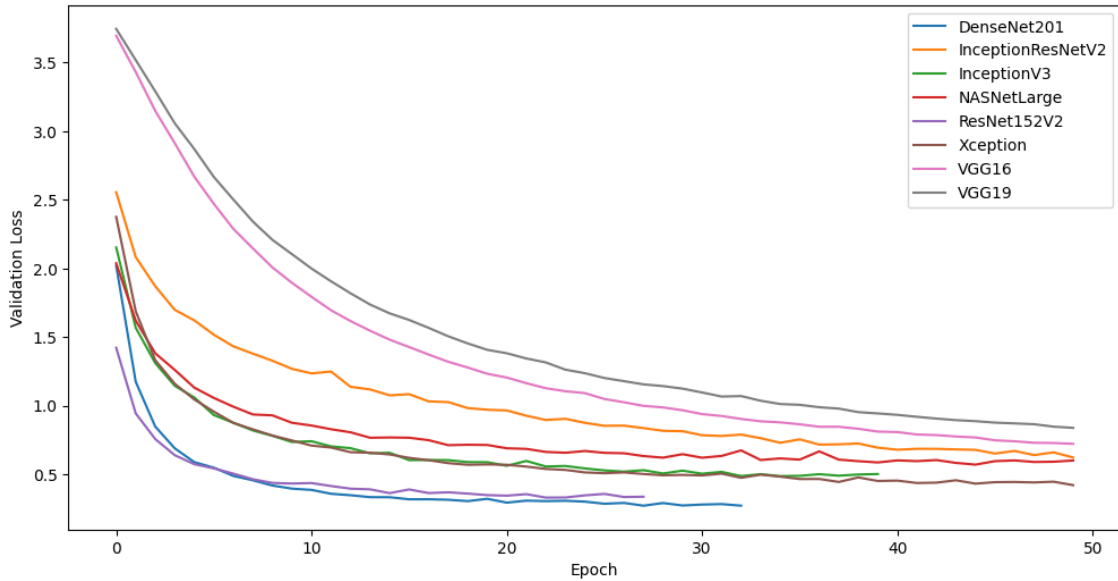
Şekil 4.8’de ağların doğrulama doğruluğu gösterilmiştir. Bu grafiğe göre ilk yöntemde ağın öğrenmesinin daha düzensiz olduğu ikinci yöntemde daha düzenli öğrenme yaptığı gözlenmiştir. İlk yöntemde olduğu gibi doğrulama verileri üzerindeki en yüksek doğruluk DenseNet201 ağında elde edilmiş, en düşük doğruluk ise VGG19 ağında elde edilmiştir.



Şekil 4.8. Katman Eklenecek Yapılan Öğrenme Aktarımında Elde Edilen Doğrulama Verileri Üzerindeki Başarı Grafiği

Şekil 4.9’da ise doğrulama verileri üzerindeki kayıp grafiği gösterilmiştir. Bu grafik incelendiğinde ise en düşük kayıp değerinin DenseNet201 ağında olduğu

gözlenmiş, en yüksek kayıp değerinin elde edildiği VGG19 ağı en başarısız ağ olduğu gözlenmiştir.



Şekil 4.9. Katman Eklenerek Yapılan Öğrenme Aktarımında Elde Edilen Doğrulama Verileri Üzerindeki Kayıp Grafiği

Çizelge 4.3’de yer alan sonuçlar incelendiğinde ilk yöntemde elde edilen performanslarla benzer şekilde DenseNet201 ağının eğitim, test ve doğrulama verilerinin tümünde en başarılı sonuçları verdiği gözlenmiştir. Bu da eklenen katmanların eğitim verileri üzerinde de olumlu performans sergilediğini göstermiştir. Ayrıca ilk yöntemde elde edilen en yüksek doğrulama başarısı %90,73’ten %91,28’e yükselmiştir. Doğrulama verileri üzerindeki kaybı %31,25’ten %27,06 ya düşmüştür. Aynı şekilde ağın test görüntüleri üzerindeki performansı incelendiğinde ise %91,09’dan %94,33 gibi oldukça yüksek bir başarıya ulaşılmıştır. En başarısız ağ ise ilk yöntemde olmuş olan NasNetLarge ağının aksine VGG19 ağı olduğu gözlemlenmiştir. Katman ekleme işleminin NasNetLarge ağının olumlu VGG19 ağını ise olumsuz yönde etkileyerek ağın öğrenme kapasitesini düşürdüğü gözlenmiştir. İlk yöntemde elde edilen düşük eğitim başarısı %81,64’ten %83,80’e çekilmiştir. Doğrulama verilerinin en düşük başarısı ise %70,70’ten %74,25’e yükselmiştir. Test verilerinin en düşük başarısı ise %71,66’dan %71,26’ya düşmüştür. Bu verilere göre en başarılı ağ DenseNet201, en başarısız ağ ise VGG19 olmuştur.

Çizelge 4.3. Katman Eklenerek Yapılan Öğrenme Aktarımı Yönteminde Elde Edilen Başarı ve Kayıp Değerleri

Ağ\Değerlendirme Kriteri	Eğitim Başarısı	Eğitim Kaybı	Doğrulama Başarısı	Doğrulama Kaybı	Test Başarısı	Test Kaybı
VGG16	0,8774	0,4691	0,7861	0,7111	0,7976	0,8760
VGG19	0,8380	0,6057	0,7425	0,8376	0,7126	1,2661
InceptionV3	0,9595	0,1730	0,8446	0,4857	0,8583	0,5386
Xception	0,9541	0,1653	0,8555	0,4201	0,8502	0,5134
DenseNet201	0,9956	0,0385	0,9128	0,2706	0,9433	0,2538
NasNetLarge	0,9610	0,1586	0,8065	0,5997	0,8421	0,5085
ResNet152V2	0,9855	0,0802	0,8991	0,3354	0,8866	0,3592
InceptionResNetV2	0,8935	0,3764	0,8119	0,6214	0,8421	0,5896

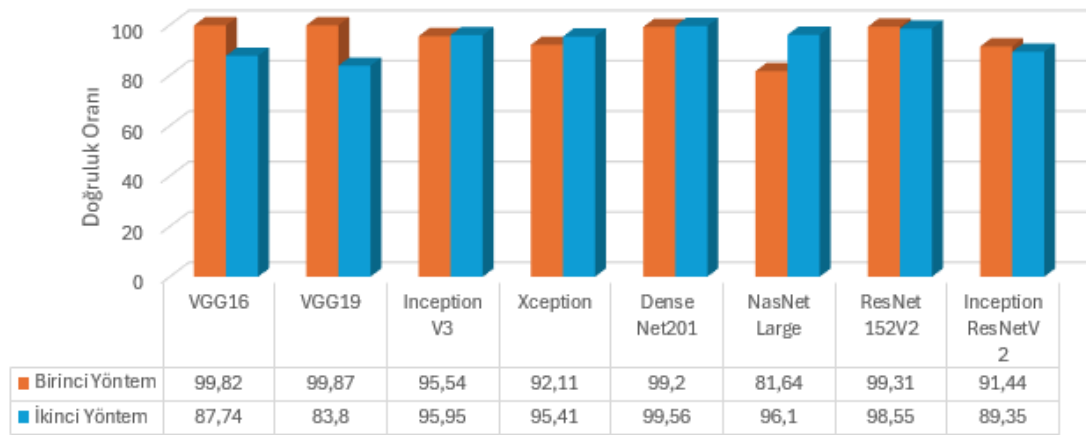
İkinci yöntem sonucunda elde edilen kesinlik, duyarlılık, f1-skoru performansları ise Çizelge 4.4’de veri seti üzerindeki ortalamaları alınarak tablo halinde verilmiştir. Bu sonuçlar incelendiğinde kesinlik parametresinde en başarılı sonucun DenseNet201’de en başarısız sonucun ise VGG19 ağında elde edildiği, duyarlılık parametresi incelendiğinde en başarılı sonucun DenseNet201’de en başarısız sonucun VGG19’da f1 skoru incelendiğinde ise en başarılı sonucun DenseNet201’de en başarısız sonucun yine VGG19 ağında elde edildiği gözlemlenmiştir.

Çizelge 4.4. Katman Eklenerek Yapılan Öğrenme Aktarımı Yönteminde Ağlarda Elde Edilen Değerlendirme Kriterlerinin Karşılaştırılması

Ağ\Değerlendirme Kriteri	Kesinlik (Precision)	Duyarlılık (Recall)	F1-Skoru (F1-Score)	Hata Oranı (%)
VGG16	0,81	0,80	0,78	20,24
VGG19	0,74	0,71	0,70	28,74
InceptionV3	0,87	0,86	0,85	14,17
Xception	0,85	0,85	0,84	14,98
DenseNet201	0,95	0,94	0,94	5,67
NasNetLarge	0,85	0,83	0,82	15,19
ResNet152V2	0,89	0,89	0,88	11,34
InceptionResNetV2	0,86	0,84	0,84	15,79

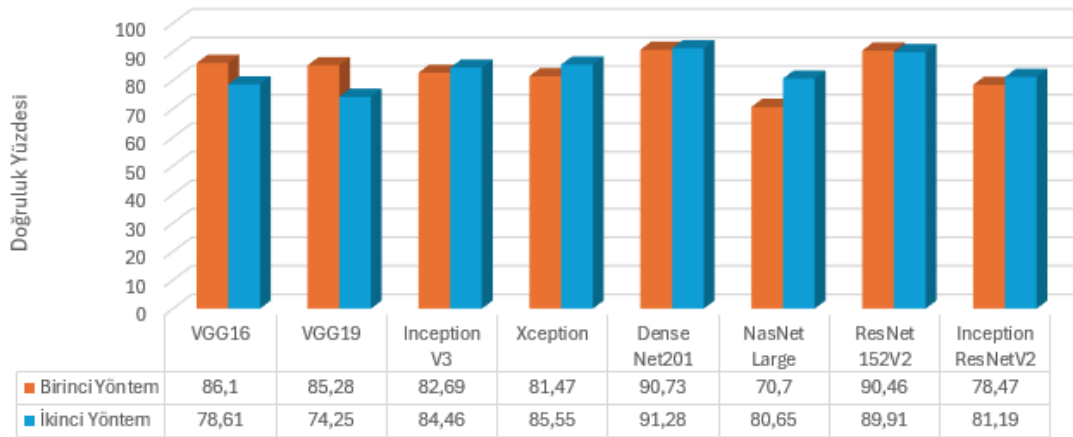
Son olarak test için verilerin sınıflandırılması aşamasında DenseNet201 ağının 247 test görüntüsü üzerinde sınıflandırma doğruluğu olarak 233 görüntüyü doğru sınıflandırmış. Yalnızca 14 görüntüyü yanlış sınıflandırmıştır. DenseNet201 ağı test görüntülerini sınıflandırmada olarak %94,33 doğruluk elde etmiş, sınıflandırma hatası olarak ise %5,67 gibi oldukça düşük bir orana düşmüştür. Ağın ilk duruma göre hata oranında %8,91’den %5,67’ye %3,24’lük düşme yaşanmıştır. Bu da katmanlı ağın daha iyi öğrendiğini göstermektedir. En başarısız ağ ise %28,74 hata oranı ile VGG19 ağı olmuştur. Test görüntülerine ait karmaşıklık matrisi Şekil 4.10’da verilmiştir.

ağlarında katmanlı yapı öğrenmeyi olumsuz bir şekilde etkileyerek performansı düşürmüştür. InceptionV3'te hemen hemen birbirine yakın doğruluklar elde edilmiştir. Xception'da ise %3,3'lük bir iyileşme gerçekleşmiştir. DenseNet201 ağında neredeyse %100'e yakın olan katmansız yöntemde %99,2 doğruluk elde edilirken, katmanlı yöntemde ise %99,56 gibi oldukça iyi bir doğruluk elde edilmiştir. NasNetLarge ağında ise katmanlı yöntem sonucunda katmansız yöntemle göre oldukça iyi bir performans göstermiştir. ResNet152V2 ağında yine iki yöntemde de yüksek doğruluklar elde edilmiştir. Son olarak InceptionResNetV2 ağında ağın öğrenmesi katmansız yöntemde katmanlı yöntemle göre daha yüksek olmuştur.



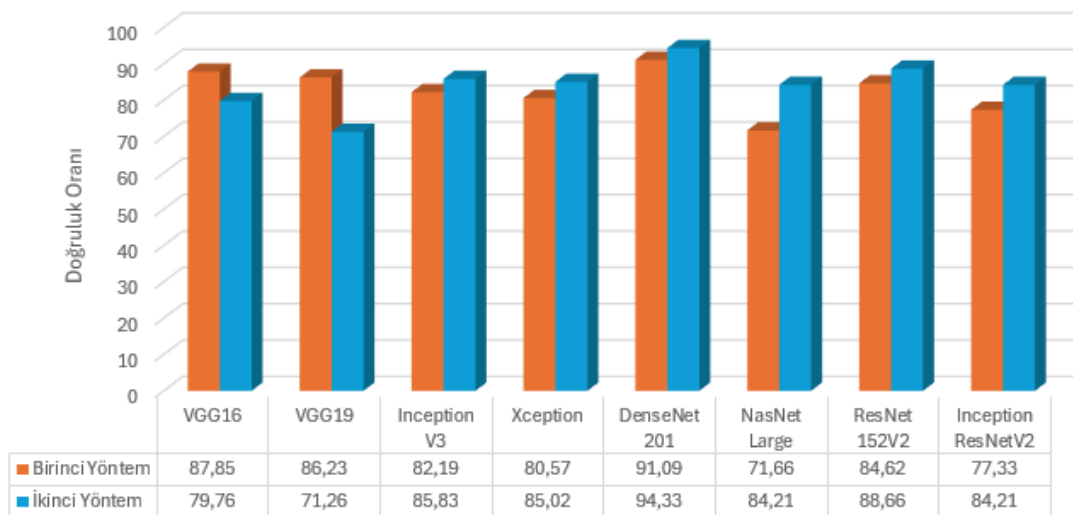
Şekil 4.11. Ağların Katmansız ve Katmanlı Yöntemle Elde Edilen Eğitim Verileri Üzerindeki Başarısının Karşılaştırılması

Doğrulama verilerinin performansı Şekil 4.12'de karşılaştırmalı olarak gösterilmiştir. Ağlar incelendiğinde sekiz ağın beşinde doğrulama verilerinin performansında artış görülmüştür. VGG modelleri birinci yöntemde daha yüksek doğruluk elde edilirken, ikinci yöntemde doğruluk oranları düşmüştür. Inception V3, Xception ve NasNetLarge, ikinci yöntemde daha yüksek doğruluk oranları elde edilmiştir. Özellikle NasNetLarge, ikinci yöntemde önemli bir doğruluk artışı göstermiştir. DenseNet201, her iki yöntemde de yüksek doğruluk oranlarına ulaşmış, bu da modelin kararlı performans sergilediğini göstermektedir. ResNet152V2 ve InceptionResNetV2 ağlarında iki yöntem arasında doğruluk oranları oldukça yakın olmuştur. Bu verilere göre en başarılı ağın DenseNet201 olduğu görülmektedir.



Şekil 4.12. Ağların Katmansız ve Katmanlı Yöntemle Elde Edilen Doğrulama Verileri Üzerindeki Başarısının Karşılaştırılması

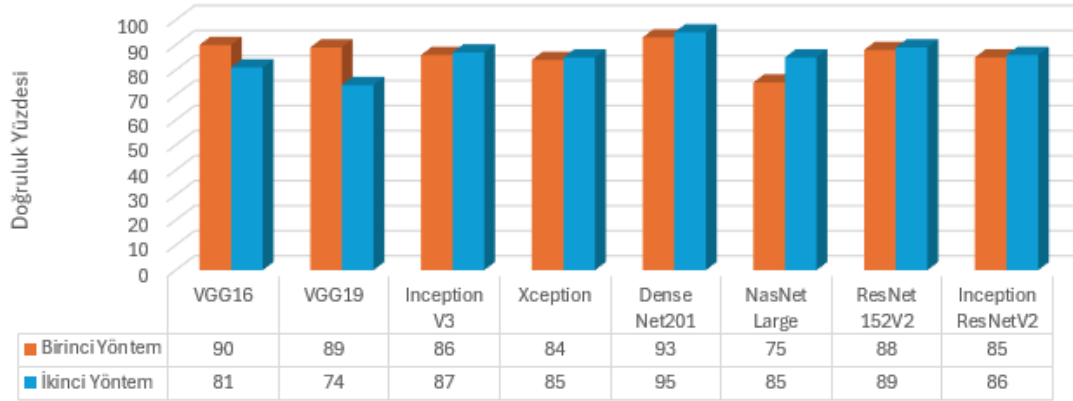
Test verileri üzerinde elde edilen sonuçları Şekil 4.13’de gösterilmiştir. Şekilde yer alan veriler incelendiğinde katman eklenerek yapılan yöntem sonucunda altı ağın sonuçlarında iyileşme gözlenmiştir. En yüksek test doğruluğu DenseNet201 ağında %94,33 olarak gözlenmiştir. VGG modellerinde birinci yöntemde daha yüksek doğruluk oranları elde edilirken, ikinci yöntemde bu doğruluk oranları düşmüştür. InceptionV3, Xception, DenseNet201, ResNet152V2, InceptionResNetV2 ve NasNetLarge ağlarında ikinci yöntemde yüksek doğruluk oranları elde edilmiştir. Özellikle DenseNet201 ve NasNetLarge, ikinci yöntemde önemli bir doğruluk artışı göstermiştir.



Şekil 4.13. Ağların Katmansız ve Katmanlı Yöntemle Elde Edilen Test Doğruluğu Performanslarının Karşılaştırılması

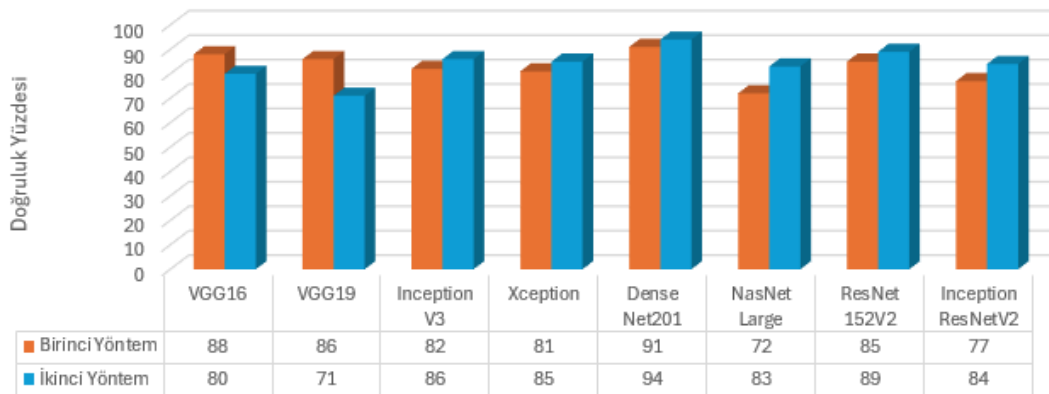
Şekil 4.14 incelendiğinde VGG mimarilerinde birinci yöntemde daha yüksek kesinlik oranları elde edilirken, ikinci yöntemde bu doğruluk oranları belirgin bir şekilde

düşmüştür. Inception V3, Xception, DenseNet201, NasNetLarge, ResNet152V2 ve InceptionResNetV2 ağlarında ikinci yöntemde daha yüksek kesinlik oranları elde edilmiştir. DenseNet201 her iki yöntemde de en yüksek kesinlik oranlarına ulaşmış, ikinci yöntemle daha iyi sonuç elde edilmiştir.



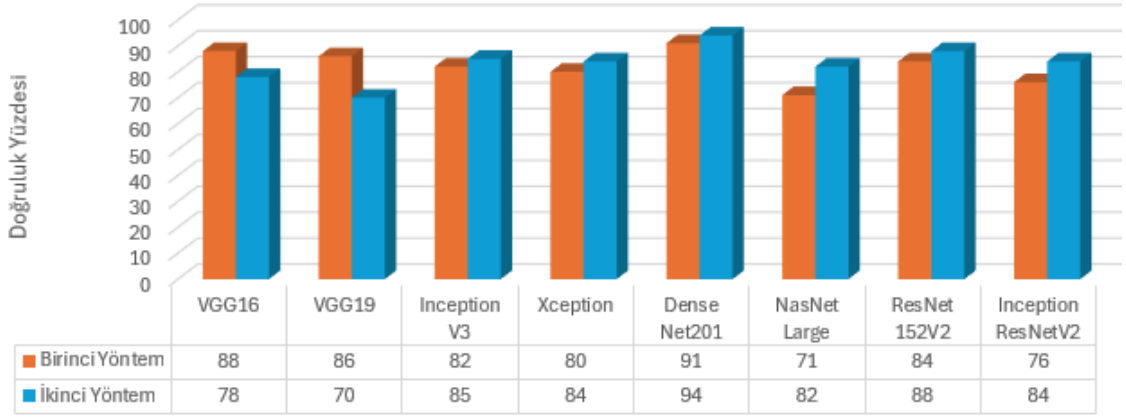
Şekil 4.14. Ağların Katmansız ve Katmanlı Yöntemle Elde Edilen Kesinlik Performanslarının Karşılaştırılması

Şekil 4.15 incelendiğinde VGG modelleri birinci yöntemde daha yüksek duyarlılık oranları elde edilirken, ikinci yöntemde bu duyarlılık oranları belirgin bir şekilde düşmüştür. Diğer ağların tümünde duyarlılık değeri olarak ikinci yöntemde daha yüksek performanslar elde edilmiştir. Duyarlılık parametresinde en yüksek başarı ise %94 ile DenseNet201 ağında elde edilmiştir.



Şekil 4.15. Ağların Katmansız ve Katmanlı Yöntemle Elde Edilen Duyarlılık Performanslarının Karşılaştırılması

Şekil 4.16’de F1-Skoru incelendiğinde yine VGG16 ve VGG19 ağı dışında tüm ağlarda iyileşme meydana gelmiştir. En yüksek F1-skoruna sahip olan ağ ilk ve ikinci yöntemde DenseNet201 olmuştur.



Şekil 4.16. Ağların Katmansız ve Katmanlı Yöntemle Elde Edilen F1-Skoru Performanslarının Karşılaştırılması

5. SONUÇLAR VE ÖNERİLER

5.1 Sonuçlar

Çeşitli kaynaklardan toplanan kelebek fotoğrafları üzerinde yapılan bu kelebek sınıflandırma çalışması sonucunda yüksek doğruluk değerlerine ulaşılmıştır. Kelebeklerin doğru sınıflandırılması, biyoçeşitliliğin korunması, nesli tehlike altında bulunan türlerin tespiti ve korunması, ekoturizm ve doğa fotoğrafçılığı, bilimsel çalışmalar gibi birçok alanda önemli bir yere sahiptir. Bu çalışma sayesinde bu türlerin tespitine önemli bir çözüm sunulmuştur.

Araştırma kapsamında VGG16, VGG19, InceptionV3, Xception, DenseNet201, NasNetLarge, ResNet152V2 ve InceptionResNetV2 gibi derin öğrenme modelleri iki farklı yöntemle eğitilmiş ve test edilmiştir. Çalışma öğrenme aktarımı yönteminin sınıflandırma çalışmalarında, özellikle önceden eğitilmiş ağlar kullanıldığında yüksek doğruluklar elde edilebildiği ortaya konmuştur. İlk yöntemde ağlara doğrudan sınıflandırma katmanı eklenerek öğrenme aktarımı yapılmıştır. İkinci yöntemde ise çeşitli katmanlar eklenerek ağların öğrenmesi arttırılmıştır. Yapılan çalışmalar sonucunda elde edilen verilere göre ilk yöntemde %91,09 ile DenseNet201 ağı olurken, ikinci yöntemde bu doğruluk %94,33'e yükselmiştir. DenseNet201 ağı 247 sınıflandırma test görüntüsünden 233 tanesini doğru sınıflandırmayı başarmıştır. Bu sonuçlara göre en başarılı sınıflandırma sonuçlarını verdiği tespit edilmiştir. Tüm ağlar incelendiğinde öğrenme aktarımı yönteminde ağlara eklenen yeni katmanların ağların performansını iyileştirdiği gözlemlenmiştir. Katman sayısının artması öğrenmeyi arttırırken, ağların parametre sayısının artması öğrenmeyi düşürmüştür.

Çalışmanın amaçlarından biri olan veri setlerinin yetersizliğinde karşılaşılan sorunların çözülmesi amaçlanmıştır. Bu bağlamda 49 türe ait 5623 görüntülü bir veri seti kazandırılmıştır. Bu sayede kelebek sınıflandırma alanında çalışma yapmayı amaçlayan araştırmacılar için yeni bir veri seti oluşturulmuştur. Araştırma sonucunda literatüre sunulan katkılar bu şekilde listelenebilir.

5.2 Öneriler

Araştırma sonuçlarının geliştirilmesi için farklı ve daha geniş veri setleri üzerinde tekrarlanarak sonuçların genelleştirilebilirliği test edilebilir. Yeni derin öğrenme

modelleri ve eğitim yöntemleri geliştirilerek mevcut bulgularla karşılaştırılabilir. Birinci ve ikinci yöntemin hibrit kullanımı ile performans artışı sağlanıp sağlanamayacağı araştırılabilir. Bu çalışma, kelebek sınıflandırma alanında önemli bulgular ortaya koymuş olup, gelecekteki araştırmalara ışık tutacak önerilerde bulunmaktadır. Derin öğrenme modellerinin kelebek sınıflandırma görevlerindeki başarısı, biyolojik araştırmalar ve doğa koruma projelerinde önemli katkılar sağlayabilir. Çalışma kapsamında ileride bu çalışmanın Türkiye’de yer alan tüm türleri kapsayan büyük bir veri seti oluşturularak ağların eğitilmesi amaçlanmaktadır. Proje bu şekilde tamamlanarak araştırmacıların kullanabileceği bir telefon uygulaması oluşturularak kullanıcının internet bağlantısı olmaksızın yüklenen fotoğrafın hangi türe ait olduğunu görebilecekleri bir uygulama geliştirmek amaçlanmaktadır. Bu sayede kullanıcılar internet erişiminin olmadığı yerlerde bile telefonlarına yükledikleri görüntülerin hangi sınıfa yüzde kaç oranında ait olduğunu görebileceklerdir.

6. KAYNAKLAR

- Alhady, S.S.N., Kai, X.Y., 2018. Butterfly Species Recognition Using Artificial Neural Network. pp. 449–457. doi:10.1007/978-981-10-8788-2_40
- Almryad, A.S., Kutucu, H., 2020. Automatic identification for field butterflies by convolutional neural networks. *Engineering Science and Technology, an International Journal* 23 1 , 189–195. doi:10.1016/j.jestch.2020.01.006
- Andrian, R., Maharani, D., Muhammad, M.A., Junaidi, A., 2019. Butterfly identification using gray level co-occurrence matrix (glcm) extraction feature and k-nearest neighbor (knn) classification. *Register: Jurnal Ilmiah Teknologi Sistem Informasi* 6 1 , 11. doi:10.26594/register.v6i1.1602
- Bilim ve Teknik, 2011. Türkiye'nin Kelebekleri.
- Brownlee, J., 2018. A Gentle Introduction to Transfer Learning for Deep Learning [WWW Document]. URL <https://machinelearningmastery.com/transfer-learning-for-deep-learning/> (accessed 5.15.24).
- Butterflies, Butterfly Photo Gallery [WWW Document], 2024. URL <https://www.naturephoto-cz.com/butterflies.html> (accessed 4.3.24).
- Butterflies of France [WWW Document], 2024. URL <https://www.butterfliesoffrance.com/index.htm> (accessed 4.3.24).
- Carvajal, J.A., Romero, D.G., Sappa, A.D., 2017. Fine-Tuning Based Deep Convolutional Networks for Lepidopterous Genus Recognition. pp. 467–475. doi:10.1007/978-3-319-52277-7_57
- Chang, Q., Qu, H., Wu, P., Yi, J., 2017. Fine-Grained Butterfly and Moth Classification Using Deep Convolutional Neural Networks. doi:10.13140/RG.2.2.22642.84161
- Chauhan, R., Ghanshala, K.K., Joshi, R.C., 2018. Convolutional Neural Network (CNN) for Image Detection and Recognition, in: 2018 First International Conference on Secure Cyber Computing and Communication (ICSCCC). IEEE, pp. 278–282. doi:10.1109/ICSCCC.2018.8703316
- Chollet, F., 2016. Xception: Deep Learning with Depthwise Separable Convolutions.
- El Naqa, I., Murphy, M.J., 2015. What Is Machine Learning?, in: *Machine Learning in Radiation Oncology*. Springer International Publishing, Cham, pp. 3–11. doi:10.1007/978-3-319-18305-3_1
- Elmas, B., 2021. Türkiye'deki Kelebek Türlerinin Basamaklı Evrişimli Sinir Ağları ile Sınıflandırılması. *Konya Journal of Engineering Sciences* 9 3 , 568–587. doi:10.36306/konjes.844847
- Ertugrul, O., Kaya, Y., Kayci, L., Tekin, R., 2015. A Vision System for Classifying Butterfly Species by using Law's Texture Energy Measures. *Int J Biomed Data Min* 1, 16–24.
- euroButterflies [WWW Document], 2024. URL <http://www.eurobutterflies.com/index.php?region=EU> (accessed 4.3.24).
- Goodfellow, I., Bengio, Y., Courville, A., 2016. *Deep Learning*. MIT Press.
- Hassan, S.N.A., Rahman, N.S.A., Htike, Z.Z., Win, S. Le, 2014. Advances in Automatic Insect Classification. *Electrical and Electronics Engineering: An International Journal* 3 2 , 51–63. doi:10.14810/elelij.2014.3204
- He, K., Zhang, X., Ren, S., Sun, J., 2016. Identity Mappings in Deep Residual Networks.
- Huang, G., Liu, Z., van der Maaten, L., Weinberger, K.Q., 2016. Densely Connected Convolutional Networks.
- Kameron Arzar, N.N., Sabri, N., Mohd Johari, N.F., Amilah Shari, A., Mohd Noordin, M.R., Ibrahim, S., 2019. Butterfly Species Identification Using Convolutional Neural Network (CNN), in: 2019 IEEE International Conference on Automatic

- Control and Intelligent Systems (I2CACIS). IEEE, pp. 221–224. doi:10.1109/I2CACIS.2019.8825031
- Kang, S.-H., Cho, J.-H., Lee, S.-H., 2014. Identification of butterfly based on their shapes when viewed from different angles using an artificial neural network. *J Asia Pac Entomol* 17 2 , 143–149. doi:<https://doi.org/10.1016/j.aspen.2013.12.004>
- Kang, S.-H., Jeon, W., Lee, S.-H., 2012a. Butterfly species identification by branch length similarity entropy. *J Asia Pac Entomol* 15 3 , 437–441. doi:10.1016/j.aspen.2012.05.005
- Kang, S.-H., Song, S.-H., Lee, S.-H., 2012b. Identification of butterfly species with a single neural network system. *J Asia Pac Entomol* 15 3 , 431–435. doi:10.1016/j.aspen.2012.03.006
- Kartika, D.S.Y., Herumurti, D., Yuniarti, A., 2018. Butterfly Image Classification Using Color Quantization Method on HSV Color Space and Local Binary Pattern. *IPTEK Journal of Proceedings Series 4 1* , 78. doi:10.12962/j23546026.y2018i1.3512
- Kaya, U., 2019. Derin öğrenme nedir?, in: *Derin Öğrenme*. Kodlab Yayincılık, pp. 1–2.
- Kaya, Y., Kayci, L., 2014. Application of artificial neural network for automatic detection of butterfly species using color and texture features. *Vis Comput* 30 1 , 71–79. doi:10.1007/s00371-013-0782-8
- Kaya, Yılmaz, Kayci, L., Ertugrul, O., Tekin, R., 2015. Identification of Butterfly Species with Rough Set Approach Based on Textural Features. *International Journal of Computer Vision, Machine Learning and Data Mining*.
- Kaya, Y., Kayci, L., Tekin, R., 2013. A Computer Vision System for the Automatic Identification of Butterfly Species via Gabor-Filter-Based Texture Features and Extreme Learning Machine : GF + ELM.
- Kaya, Yılmaz, Kayci, L., Uyar, M., 2015. Automatic identification of butterfly species based on local binary patterns and artificial neural network. *Appl Soft Comput* 28, 132–137. doi:10.1016/j.asoc.2014.11.046
- Kayci, L., Kaya, Y., 2014. A vision system for automatic identification of butterfly species using a grey-level co-occurrence matrix and multinomial logistic regression. *Zool Middle East* 60 1 , 57–64. doi:10.1080/09397140.2014.892340
- Kelebek-Türk Türkiye'nin Kelebekleri [WWW Document], 2024. URL <https://www.kelebek-turk.com/index.php> (accessed 4.3.24).
- Kızrak, A., 2018. Adım Adım Google Colab Ücretsiz TPU Kullanımı [WWW Document]. URL <https://ayyucekizrak.medium.com/adım-adım-google-colab-ücretsiz-tpu-kullanımı-621dc6e5487d> (accessed 5.15.24).
- Le-Qing Zhu Meng-Yuan Ma, Z.Z.P.-Y.Z.W.W.D.-D.W.D.-X.Z.X.W., Wang, H.-Y., 2017. Hybrid deep learning for automated lepidopteran insect image classification. *Orient Insects* 51 2 , 79–91. doi:10.1080/00305316.2016.1252805
- Li, F., Xiong, Y., 2018. Automatic identification of butterfly species based on HoMSC and GLCMoIB. *Vis Comput* 34 11 , 1525–1533. doi:10.1007/s00371-017-1426-1
- Liang, B., Wu, S., Xu, K., Hao, J., 2020. Butterfly Detection and Classification Based on Integrated YOLO Algorithm, in: *Genetic and Evolutionary Computing* . pp. 500–512. doi:10.1007/978-981-15-3308-2_55
- Lin, Z., Jia, J., Gao, W., Huang, F., 2020. Fine-grained visual categorization of butterfly specimens at sub-species level via a convolutional neural network with skip-connections. *Neurocomputing* 384, 295–313. doi:10.1016/j.neucom.2019.11.033
- Moths and Butterflies of Europe and North Africa [WWW Document], 2024. URL <https://www.leps.it/> (accessed 4.3.24).
- Nie, L., Wang, K., Fan, X., Gao, Y., 2017. Fine-Grained Butterfly Recognition with Deep Residual Networks: A New Baseline and Benchmark, in: *2017 International*

- Conference on Digital Image Computing: Techniques and Applications (DICTA). pp. 1–7. doi:10.1109/DICTA.2017.8227435
- Ruaika, D., 2024. Optimisation of Residual Network Using Data Augmentation and Ensemble Deep Learning for Butterfly Image Classification. *IJID (International Journal on Informatics for Development)* 12. doi:10.14421/ijid.2023.4038
- Samruddhi N Hegde, Manjesh R, P Sindhura, Reema Rodrigues, Sheethal A, 2020. Identification of Butterfly Species Using VGG-16. *International Research Journal of Modernization in Engineering Technology and Science* 02 05 , 387–391.
- Şener, M., 2023. Makine Öğrenmesinde Aktarımlı Öğrenme Yaklaşımı ve Aktarımlı Öğrenme Modelleri [WWW Document]. URL [https://blog.architect.com/blog/2468/makine-ogrenmesinde-aktarimli-ogrenme-yaklasimi-ve-aktarimli-ogrenme-modelleri/#:~:text=Aktarımlı%20Öğrenmede%20\(Transfer%20Learning\)%2C,başarı%20gösteren%20modeller%20elde%20edilir.](https://blog.architect.com/blog/2468/makine-ogrenmesinde-aktarimli-ogrenme-yaklasimi-ve-aktarimli-ogrenme-modelleri/#:~:text=Aktarımlı%20Öğrenmede%20(Transfer%20Learning)%2C,başarı%20gösteren%20modeller%20elde%20edilir.) (accessed 5.16.24).
- Simonyan, K., Zisserman, A., 2014. Very Deep Convolutional Networks for Large-Scale Image Recognition, in: *ICLR 2015*.
- Stork, N.E., 2018. How Many Species of Insects and Other Terrestrial Arthropods Are There on Earth? *Annu Rev Entomol* 63 1 , 31–45. doi:10.1146/annurev-ento-020117-043348
- Szegedy, C., Ioffe, S., Vanhoucke, V., Alemi, A., 2016. Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning.
- Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., Wojna, Z., 2015. Rethinking the Inception Architecture for Computer Vision.
- Tan, A., Zhou, G., He, M., 2020. Rapid Fine-Grained Classification of Butterflies Based on FCM-KM and Mask R-CNN Fusion. *IEEE Access* 8, 124722–124733. doi:10.1109/ACCESS.2020.3007745
- Tek, K., Emel, S., 2020. The Automatic Identification of Butterfly Species Using Deep Learning Methodologies.
- Theivaprakasham, H., 2021. Identification of Indian butterflies using Deep Convolutional Neural Network. *J Asia Pac Entomol* 24 1 , 329–340. doi:10.1016/j.aspen.2020.11.015
- Wang, J., Ji, L., Liang, A., Yuan, D., 2012. The identification of butterfly families using content-based image retrieval. *Biosyst Eng* 111 1 , 24–32. doi:10.1016/j.biosystemseng.2011.10.003
- Xin, D., Chen, Y.-W., Li, J., 2020. Fine-Grained Butterfly Classification in Ecological Images Using Squeeze-And-Excitation and Spatial Attention Modules. *Applied Sciences* 10 5 , 1681. doi:10.3390/app10051681
- Xue, A., Li, F., Xiong, Y., 2019. Automatic Identification of Butterfly Species Based on Gray-Level Co-occurrence Matrix Features of Image Block. *J Shanghai Jiaotong Univ Sci* 24 2 , 220–225. doi:10.1007/s12204-018-2013-y
- Yılmaz, A., 2021. Yapay zekâ nedir?, in: *Yapay Zekâ . Kodlab Yayincılık*, pp. 4–5.
- Yılmaz Kaya Lokman Kayci, R.T., Ertuğrul, Ö.F., 2014. Evaluation of texture features for automatic detecting butterfly species using extreme learning machine. *Journal of Experimental & Theoretical Artificial Intelligence* 26 2 , 267–281. doi:10.1080/0952813X.2013.861875
- Yudha Kartika, D.S., 2018. Local Binary Pattern Method and Feature Shape Extraction for Detecting Butterfly Image. *International Journal of GEOMATE* 15 50 . doi:10.21660/2018.50.IJCST21

- Zhao, R., Li, C., Ye, S., Fang, X., 2019a. Butterfly recognition based on Faster R-CNN, in: *Journal of Physics: Conference Series*. Institute of Physics Publishing. doi:10.1088/1742-6596/1176/3/032048
- Zhao, R., Li, C., Ye, S., Fang, X., 2019b. Butterfly Recognition Based on Faster R-CNN. *J Phys Conf Ser* 1176, 032048. doi:10.1088/1742-6596/1176/3/032048
- Zhu, L., Spachos, P., 2019. Butterfly Classification with Machine Learning Methodologies for an Android Application, in: *2019 IEEE Global Conference on Signal and Information Processing (GlobalSIP)*. IEEE, pp. 1–5. doi:10.1109/GlobalSIP45357.2019.8969441
- Zoph, B., Vasudevan, V., Shlens, J., Le, Q. V., 2017. Learning Transferable Architectures for Scalable Image Recognition.

