

DERİN ÖĞRENME İLE PLANKTON SINIFLANDIRMASI

PLANKTON CLASSIFICATION WITH DEEP LEARNING

Betül SÖMEK

Doç. Dr. Seniha Esen YÜKSEL ERDEM

Tez Danışmanı

Hacettepe Üniversitesi

Lisansüstü Eğitim-Öğretim ve Sınav Yönetmeliğinin

Elektrik ve Elektronik Mühendisliği Anabilim Dalı için Öngördüğü

YÜKSEK LİSANS TEZİ olarak hazırlanmıştır.

2022

ÖZET

DERİN ÖĞRENME İLE PLANKTON SINIFLANDIRMASI

Betül SÖMEK

Yüksek Lisans, Elektrik Elektronik Mühendisliği

Danışman: Doç. Dr. Seniha Esen YÜKSEL ERDEM

Mart 2022, 108 sayfa

Planktonlar atmosferdeki oksijenin yaklaşık olarak yarısını üretmekten sorumlu olan, su ekosisteminde besin zincirinin en alt basamağında yer alan, dünyadaki yaşamın en önemli bileşenlerinden biridir. Plankton dağılımları, küresel ısınma gibi iklim değişiklikleri için önemli bir haberci olarak görülmektedir. Bundan dolayı dağılımlarını takip etmek, analizlerini yapmak ve su kalitesi hakkında bilgi sahibi olmak kritik bir öneme sahiptir. Plankton görüntüleme teknolojilerinin gelişmesiyle birlikte su ekosistemlerinden çok sayıda plankton görüntüsü elde edilmektedir. Geleneksel manuel sınıflandırma sistemleri, giderek büyüyen plankton veri kümesi sınıflandırma gereksinimlerini karşılayamamaktadır. Manuel sınıflandırma sistemleri, uzmanlık bilgisi gerektiren ve oldukça zaman alıcı olan yöntemlerdir. Bu nedenle, elde edilen verinin çok fazla sayıda olması ve planktonların dağılımlarının bilinmesinin önemli olması veriyi sınıflandıran otomatik sistemlere ihtiyacı günden güne arttırmaktadır. Tez kapsamında, Ulusal Veri Bilimi Kasesi olarak adlandırılan Kaggle platformunda düzenlenen, planktonları otomatik olarak algılamayı amaçlayan yarışmada yayınlanan 121 sınıfa ait 30.336 plankton görüntüsünden oluşan veri kümesinin 118 sınıfa ve 38 sınıfa sahip iki farklı sürümü kullanılmıştır. Bu çalışmada, plankton görüntülerini sınıflandırmak

iin derin evriřimsel sinir ađları olan Inceptionv3, InceptionResNetv2, DenseNet, ResNet-50, VGG-16 ađları; aktarım ğrenmesi, ince ayar, tm katmanların eđitilmesi yntemleri ile eđitilmiř ve veri arttırma ve n iřleme iřlemlerinin eđitime katkısı gzlemlenmiřtir. Kaggle veri kmesinin kullanılan iki farklı srmyle ađlarda eđitilen modellerde 118 sınıfa sahip ilk srmnde en yksek %78 bařarım, 38 sınıfa sahip ikinci srmnde %92 bařarım elde edilmiřtir. Ortalama topluluk yntemi kullanılarak ilk srmnde bařarımın %2, ikinci srmnde %1 artması sađlanmıřtır.

Anahtar Kelimeler: Plankton Sınıflandırılması, Derin ğrenme, Evriřimsel Sinir Ađı

ABSTRACT

PLANKTON CLASSIFICATION WITH DEEP LEARNING

Betül SÖMEK

Master of Science, Department of Electrical and Electronics Engineering

Supervisor: Doç. Dr. Seniha Esen YÜKSEL ERDEM

March 2022, 108 pages

Plankton which is at the bottom of the food chain in the aquatic ecosystem and is responsible for producing approximately half of the oxygen in the atmosphere, is one of the most important components of life on earth. Plankton distributions are seen as an important precursor for climate changes such as global warming. Therefore, it is very critical to follow their distribution, to analyze and to have information about water quality. With the development of plankton imaging technologies, a large number of plankton images are obtained from aquatic ecosystems. Traditional manual classification systems are unable to meet the ever-growing plankton dataset classification requirements. Manual classification systems are methods that require specialist knowledge and are very time consuming. For this reason, the large number of data obtained and the importance of knowing the distribution of plankton increase the need for automated systems that classify data day by day. Within the scope of the thesis, two different versions of 118 classes and 38 classes of the dataset consisting of 30,336 plankton images belonging to 121 classes published in the competition aiming to automatically detect plankton and organized on the Kaggle platform, called the National Data Science Bowl, were

used. In this study, deep convolutional neural networks namely Inceptionv3, InceptionResNetv2, DenseNet, ResNet-50, VGG-16 networks were trained to classify plankton images; transfer learning, fine tuning, training of all layers methods were used and the contribution of data augmentation and preprocessing on classification was observed. With the two different versions of the Kaggle dataset, in the models trained on the networks, the highest performance of 78% was achieved in the first version with 118 classes, and 92% performance in the second version with 38 classes. By using the average ensemble method, the performance increased by 2% in the first version and 1% in the second version of the dataset.

Keywords: Plankton Classification, Deep Learning, Convolutional Neural Network

TEŞEKKÜR

Tez çalışmam boyunca değerli katkılarını ve rehberliğini esirgemeyen, değerli fikirleriyle katkıda bulunan danışman hocam Doç. Dr. Seniha Esen Yüksel ERDEM'e saygı ve teşekkürlerimi sunarım.

Hayatım boyunca sonsuz sevgisini ve sabrını esirgemeyen, her an yanımda olan canım aileme en içten duygularıyla teşekkür ederim.

Son olarak tez çalışmam sırasında desteğini esirgemeyen, her anımda sevgisiyle ve sabrıyla bana eşlik eden sevgili eşim Süleyman Kazım SÖMEK'e sonsuz teşekkür ederim.

İçindekiler

ÖZET	i
ÖZ	iii
TEŞEKKÜR	v
İÇİNDEKİLER	vi
ŞEKİLLER DİZİNİ	x
ÇİZELGELER DİZİNİ.....	xiv
SİMGELER VE KISALTMALAR	xv
1. GİRİŞ	1
1.1. Tezin Amacı	2
1.2. Tezin Kapsamı ve Akışı	5
2. PLANKTON TANIMA	7
2.1. Plankton Görüntüleme	7
2.2. Plankton Taksonomisi	8
2.3. Plankton Veri Kümeleri.....	9
2.3.1. WHOI-Plankton	9
2.3.2. ZooScan	10
2.3.3. Kaggle Plankton.....	11
2.4. Plankton Tanıma Sistemi	15
2.4.1. Öznitelik Çıkarımı Tabanlı Plankton Tanıma	16
2.4.2. Evrişimli Sinir Ağı Tabanlı Plankton Tanıma.....	18
3. EVRİŞİMLİ SİNİR AĞLARI	23
3.1. Sinir Ağları Katmanları ve Yapıları	23
3.1.1. Evrişimsel Katmanlar	24
3.1.2. Havuz Katmanları	26
3.1.3. Tamamen Bağlı Katmanlar	27
3.1.4. Aktivasyon Fonksiyonu	29
3.1.5. CNN Mimarileri.....	33

3.2. Sinir Ağları Eğitimi.....	40
3.2.1. Kayıp fonksiyonu	40
3.2.2. Gradyan İniş ve Geri Yayılım Yöntemi	43
3.2.3. Kaybolan ve Patlayan Gradyan	44
3.3. Aşırı Uyumun Önlenmesi ve Sinir Ağları Optimizasyonu	45
3.3.1. Seyreltme	45
3.3.2. Veri Arttırma	45
3.3.3. Aktarım Öğrenmesi	46
4. DENEYLER ve SONUÇLAR	47
4.1. Veri Önışleme.....	47
4.2. Değerlendirme Metrikleri	51
4.3. Ortalama Topluluk Yöntemi	53
4.4. Deneyler	54
4.4.1. Deney Detayları	55
4.4.2. Deney-1	55
4.4.3. Deney-1 Sonuç	57
4.4.4. Deney-2.....	65
4.4.5. Deney-2 Sonuç	66
4.4.6. Deney-3	71
4.4.7. Deney-3 Sonuç	71
4.4.8. Deney-4.....	83
4.4.9. Deney-4 Sonuçları	83
4.4.10. Deney-5	84
4.4.11. Deney-5 Sonuçları	84
4.4.12. Deney-6.....	86
4.4.13. Deney-6 Sonuçları	86
5. SONUÇLAR.....	93
5.1. Gelecek Çalışmalar	96
KAYNAKLAR	100
TEZ ÇALIŞMASI ORİJİNALLİK RAPORU	109

ÖZGEÇMİŞ	110
----------------	-----



TABLolar

Tablo 1.1.	Kullanılan Veri Kümelerinin Özellikleri	3
Tablo 2.1.	Set-1 Veri Kumesinin Özellikleri	13
Tablo 2.2.	Set-2 Veri Kumesinin Özellikleri	13
Tablo 2.3.	Kaggle Veri Kumesindeki Sınıflar ve Her Sınıftaki Veri Sayısı (#)	14
Tablo 2.4.	Literatür Özet Tablosu	22
Tablo 3.1.	Ağların Karşılaştırma Tablosu	40
Tablo 4.1.	Set-1 Veri Kumesine Kısmi Veri Arttırım Uygulanmasının Sonuçları	48
Tablo 4.2.	Set-1 Veri Kumesine Tam Veri Arttırım Uygulanmasının Sonuçları	48
Tablo 4.3.	Set-1 Veri Kumesinde Elde Edilen Deney-1 Sonuçları	58
Tablo 4.4.	Inceptionv3 Ağıyla Eğitilen Ağların Deney-1 Eğitim Doğruluk ve Kayıp Sonuçları	58
Tablo 4.5.	Set-1 Veri Kumesinde Elde Edilen Deney-2 Sonuçları	66
Tablo 4.6.	Inceptionv3 Ağıyla Eğitilen Ağların Deney-2 Eğitim Doğruluk ve Kayıp Sonuçları	67
Tablo 4.7.	Set-2 Veri Kumesine Tam Veri Arttırım Uygulanmasının Sonuçları	71
Tablo 4.8.	Set-1 Veri Kumesinde Elde Edilen Ağ Sonuçları	73
Tablo 4.9.	Set-2 Veri Kumesinde Elde Edilen Ağ Sonuçları	73
Tablo 4.10.	Set-1 ve Set-2 Veri Kümelerinin Eğitim ve Doğrulama Doğruluk/Kayıp Sonuçları	73
Tablo 4.11.	Set-1 ve Set-2 Veri Kümelerinin Ortalama Modellerin Topluluğu Deney-4 Sonucu	84
Tablo 4.12.	Set-1 ve Set-2 Veri Kümelerinin Ortalama Modellerin Topluluğu Deney-4 Sonucu	84
Tablo 4.13.	Set-1 Veri Kumesinin DenseNet ağı ve Farklı Odak Kaybı Parametreleri ile Eğitilen Modellerinin Sonuçları	85

Tablo 4.14.	Set-1 Veri Kümesinin DenseNet ağı ve Farklı Odak Kaybı Parametre- leri ile Eğitilen Modellerinin Sonuçları	85
-------------	---	----



ŞEKİLLER

Şekil 2.1.	WHOI Plankton Veri Kümesinden Alınan Örnek Görüntüler	10
Şekil 2.2.	ZooSCan Plankton Veri Kümesinden Alınan Örnek Görüntüler	11
Şekil 2.3.	Kaggle Plankton Veri Kümesinden Alınan Örnek Görüntüler	12
Şekil 2.4.	Kaggle Plankton Veri Kümesindeki Görüntülerin Piksel Cinsinden Dağılım Grafiği	13
Şekil 3.1.	Örnek CNN Mimarisi [1]	23
Şekil 3.2.	Filtre sol üst köşedeki giriş vektörüne yerleştirildiğinde evrişim katmanının nasıl çalıştığına dair örnek [2].	25
Şekil 3.3.	3×3 Çekirdek (Kernel) için 2D-Evrişim-İşlemi örneği	25
Şekil 3.4.	Maksimum ve Ortalama Havuzlama Örneği	27
Şekil 3.5.	Tamamen bağlantılı bir katmandaki bağlantıların gösterimi.	28
Şekil 3.6.	Sigmoid fonksiyonu (sol) ve türevi (sağ) [3]	30
Şekil 3.7.	Tanh fonksiyonu (sol) ve türevi (sağ) [3]	31
Şekil 3.8.	Relu fonksiyonu (sol) ve türevi (sağ) [3]	32
Şekil 3.9.	Sızıntı Relu fonksiyonu (sol) ve türevi (sağ) [3]	33
Şekil 3.10.	ResNet50 Artık (Residual) Blok Örneği	35
Şekil 3.11.	Inceptionv3 Ağ Mimarisi [4]	37
Şekil 3.12.	VGG-16 Ağ Mimarisi	37
Şekil 3.13.	DenseNet Ağ Mimarisi	38
Şekil 3.14.	InceptionResNetv2 Ağ Mimarisi	39
Şekil 4.1.	Kaggle Plankton Veri Kümesine Uygulanan Çevirme Veri Arttırma Yöntemi. Sırayla Orijinal, Yatay Çevirme, Dikey Çevirme	49
Şekil 4.2.	Kaggle Plankton Veri Kümesine Uygulanan Rotasyon Veri Arttırma Yöntemi. Sırayla Orijinal, 90 Derece Rotasyon, 180 Derece Rotasyon, 270 Derece Rotasyon	50

Şekil 4.3.	Kaggle Plankton Veri Kümesine Uygulanan Yakınlaştırma Veri Arttırma Yöntemi. Sırayla Orijinal, 0.75, 1.25, 1.75 Yakınlaştırma	50
Şekil 4.4.	Boşluk Ekleme Yöntemi (Sol Sütun) ve Kare Kırpma Yöntemi (Sağ Sütun)	51
Şekil 4.5.	Gerçekleşen ve Tahmin Edilen Hata Matrisi	52
Şekil 4.6.	Ortalama Topluluk Örneği.....	54
Şekil 4.7.	Deney-1 Özet. Aktarım Öğrenmesi Yöntemi, İnce Ayar Yöntemi, Tüm Katmanların Eğitilmesi Yöntemi.....	56
Şekil 4.8.	Aktarım Öğrenmesi Yöntemi, Orijinal Veri Kümesi Eğitim Doğruluk ve Kayıp Sonucu	59
Şekil 4.9.	Aktarım Öğrenmesi Yöntemi, Kısmi Veri Arttırma Veri Kümesi Eğitim Doğruluk ve Kayıp Sonucu.....	60
Şekil 4.10.	Aktarım Öğrenmesi Yöntemi, Tam Veri Arttırma Veri Kümesi Eğitim Doğruluk ve Kayıp Sonucu.....	60
Şekil 4.11.	İnce Ayar Yöntemi, Orijinal Veri Kümesi Eğitim Doğruluk ve Kayıp Sonucu.....	61
Şekil 4.12.	İnce Ayar Yöntemi, Kısmi Veri Arttırma Veri Kümesi Eğitim Doğruluk ve Kayıp Sonucu	62
Şekil 4.13.	İnce Ayar Yöntemi, Tam Veri Arttırma Veri Kümesi Eğitim Doğruluk ve Kayıp Sonucu	62
Şekil 4.14.	Tüm Katmanların Eğitilmesi Yöntemi, Orijinal Veri Kümesi Eğitim Doğruluk ve Kayıp Sonucu.....	63
Şekil 4.15.	Tüm Katmanların Eğitilmesi Yöntemi, Kısmi Veri Arttırma Veri KÜmesi Eğitim Doğruluk ve Kayıp Sonucu	63
Şekil 4.16.	Tüm Katmanların Eğitilmesi Yöntemi, Tam Veri Arttırma Veri Kümesi Eğitim Doğruluk ve Kayıp Sonucu	64
Şekil 4.17.	Deney-2'nin Özeti. Aktarım Öğrenmesi Yöntemi, İnce Ayar Yöntemi, Tüm Katmanların Eğitilmesi Yöntemi.....	65
Şekil 4.18.	Aktarım Öğrenmesi Yöntemi, Tam Veri Arttırma / Boşluk Doldurma Veri Kümesi Eğitim Doğruluk ve Kayıp Sonucu.....	67

Şekil 4.19.	Aktarım Öğrenmesi Yöntemi, Tam Veri Arttırma / Kare Kırpma Veri Kümesi Eğitim Doğruluk ve Kayıp Sonucu	68
Şekil 4.20.	İnce Ayar Yöntemi, Tam Veri Arttırma / Boşluk Doldurma Veri Kümesi Eğitim Doğruluk ve Kayıp Sonucu	68
Şekil 4.21.	İnce Ayar Yöntemi, Tam Veri Arttırma / Kare Kırpma Veri Kümesi Eğitim Doğruluk ve Kayıp Sonucu	69
Şekil 4.22.	Tüm Katmanların Eğitilmesi Yöntemi, Tam Veri Arttırma / Boşluk Doldurma Veri Kümesi Eğitim Doğruluk ve Kayıp Sonucu	70
Şekil 4.23.	Tüm Katmanların Eğitilmesi Yöntemi, Tam Veri Arttırma / Kare Kırpma Veri Kümesi Eğitim Doğruluk ve Kayıp Sonucu	70
Şekil 4.24.	Set-1, InceptionResNetv2, Tüm Katmanların Eğitilmesi Yöntemi, Tam Veri Arttırma Veri Kümesi Eğitim Doğruluk ve Kayıp Sonucu	74
Şekil 4.25.	Set-1 DenseNet Yöntem-3 Tam Veri Arttırma Veri Kümesi Eğitim Doğruluk ve Kayıp Sonucu	75
Şekil 4.26.	Set-1, ResNet-50, Tüm Katmanların Eğitilmesi Yöntemi, Tam Veri Arttırma Veri Kümesi Eğitim Doğruluk ve Kayıp Sonucu	75
Şekil 4.27.	Set-1, VGG-16, Tüm Katmanların Eğitilmesi Yöntemi, Tam Veri Arttırma Veri Kümesi Eğitim Doğruluk ve Kayıp Sonucu	76
Şekil 4.28.	Set-2, Inceptionv3, Tüm Katmanların Eğitilmesi Yöntemi, Tam Veri Arttırma Veri Kümesi Eğitim Doğruluk ve Kayıp Sonucu	77
Şekil 4.29.	Set-2, InceptionResNetv2, Tüm Katmanların Eğitilmesi Yöntemi, Tam Veri Arttırma Veri Kümesi Eğitim Doğruluk ve Kayıp Sonucu	77
Şekil 4.30.	Set-2, DenseNet, Tüm Katmanların Eğitilmesi Yöntemi, Tam Veri Arttırma Veri Kümesi Eğitim Doğruluk ve Kayıp Sonucu	78
Şekil 4.31.	Set-2, ResNet, Tüm Katmanların Eğitilmesi Yöntemi, Tam Veri Arttırma Veri Kümesi Eğitim Doğruluk ve Kayıp Sonucu	78
Şekil 4.32.	Set-2, VGG-16, Tüm Katmanların Eğitilmesi Yöntemi, Tam Veri Arttırma Veri Kümesi Eğitim Doğruluk ve Kayıp Sonucu	79
Şekil 4.33.	Yanlış Sınıflandırılan Görüntü Örnekleri. (a) Gerçek Sınıf, (b) Tahmin Edilen Sınıf, (c) Gerçek Sınıf, (d) Tahmin Edilen Sınıf	80

Şekil 4.34.	Doğru Sınıflandırılan Görüntü Örnekleri	81
Şekil 4.35.	Copedod Sınıfı Hata Matrisi.....	82
Şekil 4.36.	Copedod Sınıfında Bulunan Karıştırılan Örnek Alt Gruplar	83
Şekil 4.37.	Sütunların Grup Bazında Toplanarak Gruplandığı 118x33 Hata Matrisi - Kısım 1	88
Şekil 4.38.	Sütunların Grup Bazında Toplanarak Gruplandığı 118x33 Hata Matrisi - Kısım 2	89
Şekil 4.39.	Her Bir Sınıfın Olasılık Değerinin Sınıf Sayısı ile Çarpılarak, Satır Ba- zında Gruplandırılıp Toplanarak Elde Edilen 33x33 Hata Matrisi	90
Şekil 4.40.	Her Bir Sınıf Grubunun Sahip Olduğu Toplam Veri Sayısına Bölünerek Elde Edilen 33x33 Hata Matrisi.....	91
Şekil 4.41.	33 Plankton Grubu ile Eğitilen Model Sonucunda Elde Edilen Hata Matrisi	92

SİMGELER VE KISALTMALAR

Kısaltmalar

ANN	Artificial Neural Network (Yapay Sinir Ağı)
SVM	Support Vector Machine (Destek Vektör Makinesi)
CNN	Convolutional Neural Networks (Evrişimsel Sinir Ağları)



1. GİRİŞ

Planktonlar, gelgitler ve akıntılar tarafından taşınan ve bu kuvvetlere karşı hareket edebilecek kadar iyi yüzemeyen organizmalar olarak tanımlanmaktadırlar [5]. Bilim insanları planktonları boyut, tür vb. gibi özelliklere göre farklı şekilde sınıflandırmaktadırlar. Fakat en temelde planktonlar iki sınıfa ayrılmıştır. Fitoplanktonlar (bitkiler) ve zooplanktonlar (hayvanlar) [5].

Fitoplanktonlar küçük fotosentetik canlılardır ve besin zincirinin en alt basamağında yer almaktadırlar [6]. Atmosferdeki oksijenin yaklaşık %70'inin okyanuslarda fitoplanktonlar sayesinde üretildiği kaydedilmiştir [7]. Fitoplanktonlar kendi besinlerini üretmek için tıpkı kara bitkileri gibi güneş ışığına ve besine ihtiyaç duyarlar ve bu yüzden suyun üst katmanlarında yer alırlar. Dengeli bir ekosistemde fitoplanktonlar suda yaşayan çok çeşitli canlılara besin sağlarlar. Zooplanktonlar küçük hayvanlardır ve fitoplankton ile beslenirler ve çoğu da büyük hayvanlar tarafından yenirler.

Planktonlar dünyadaki yaşamın en önemli bileşeni olarak kabul edilmektedir. Bunun sebebi, besin zincirinin en altında yer almaları ve dünya üzerindeki oksijenin büyük bir kısmını üretmeleridir. Ph seviyesi, tuzluluk oranı, sıcaklık ve besin konsantrasyonu vb. değişikliklere oldukça duyarlıdırlar. Örneğin suda çok fazla besin maddesi bulunduğunda alg patlamaları meydana gelir [8] ve birçok zooplankton türü fitoplankton yediği için, fitoplanktonun artışındaki değişimler zooplankton popülasyonlarını hızla etkileyebilir ve bu da daha sonra besin zinciri boyunca diğer türleri etkiler.

Plankton dağılımları küresel ısınma gibi iklim değişiklikleri için önemli bir haberci olarak görülmüştür. Bundan dolayı dağılımlarını takip etmek, analizini yapmak ve su kalitesi hakkında bilgi sahibi olmak kritiktir. Geçmiş yıllarda plankton numunelerinin elde edilmesi geleneksel yöntemlerle yapılmıştır. Sudan alınan her bir numune uzman kişiler tarafından laboratuvar ortamında incelenmiş ve analiz edilmiştir. Geleneksel yöntemler oldukça zaman alıcıdır ve yüksek düzeyde profesyonel bilgi gerektirmektedir. Son 20 yılda plankton görüntülerinin otomatik olarak çekilmesini sağlayan bir çok teknoloji geliştirilmiştir. 1992 yılında Video Plankton Kaydedici üretilmiştir [9] ve üretilen ilk yerinde otomatik tanımlama cihazı olmuştur. Üretilen bu cihaz modern yerinde plankton görüntüleme cihazlarına öncü olarak kabul edilmiştir. 2004 yılında [10], doğrudan suya yerleştirilen ve optik mikroskoplara dayanan “sualtı otomatik sayısal mikroskop görüntüleyici” geliştirilmiş ve bu sayede plankton görüntülerinin otomatik olarak çekilmesi sağlanmıştır. Üretilen cihaz yalnızca yüksek

özünürlük kořulları için uygundur. 10 ila 100 mm arasındaki mikrozooplankton ve fitoplanktonların uzun süreli izlenmesi için Olson *ve ark.* Görüntüleme Akışı Sitobot'u (Imaging FlowCytobot) kullanmıştır [11]. Planktonların yerinde gözlemlenmesi için etkili bir yöntem sağlayan FlowCAM, FastCam, CytoSense ve CytoSub [12] dahil olmak üzere planktonlar hakkında bilgi edinilmesini sağlayan benzer teknolojik sistemler mevcuttur. Bu cihazlar sayesinde dünyanın farklı su ekosistemlerinden günden güne sayısı artan plankton görüntüleri elde edilmektedir. Bu sebeple su altı görüntü veri kümesi oldukça hızlı bir şekilde büyümektedir. Elde edilen verinin çok fazla olması, veriyi tanımlayan, sınıflandıran ve analiz eden otomatik algılama sistemlerine ihtiyacı günden güne arttırmaktadır. Otomatik plankton sınıflandırma sistemleri kullanılarak, ekolojik ortamın en önemli değerlendirme faktörlerinden biri olan planktonların yaşam kořulları gözlemlenebilmektedir. Çünkü planktonların yaşam kořullarının gözlemlenmesi büyük ölçüde planktonların sınıflandırmasına bağıldır. Otomatik plankton sınıflandırma sistemleri ile birlikte su ekosistemi hakkında, uzman kişilere ihtiyaç duymadan ve hızlı bir şekilde su ekosistemi hakkında bilgi elde edilebilmektedir. Örneğin, düşük plankton seviyesi su ekosistemi için tehlike oluştururken, bolluğı da su ekosistemi için toksin etkisi yaratmakta ve yıkıcı etkilere neden olabilmektedir. Bu nedenle, görüntü verilerine dayalı olarak plankton sınıflandırması bir ekosistemdeki plankton popülasyonlarını tahmin etmek için çok kullanışlı bir sistemdir. Görüntü işleme teknolojilerini kullanarak elde edilen plankton görüntülerinin otomatik sınıflandırılmasına odaklanan çok sayıda çalışma vardır [13–16]. Bilgi işlem gücü, bellek kapasitesi, güç tüketimi, görüntü sensörü çözünürlüğü olmak üzere cihaz yeteneklerindeki hızlı gelişim ve derin öğrenme alanında çalışmaların yaygınlaşmasıyla birlikte derin sinir ağıları geleneksel yöntemlere kıyasla plankton sınıflandırma alanında son zamanlarda oldukça yaygın kullanılmaktadır.

1.1. Tezin Amacı

Bu tezin amacı, plankton görüntülerini alanında uzman kişilere ihtiyaç duymadan sınıflandırabilen, derin öğrenme tabanlı plankton sınıflandırma sistemi elde etmektir. Bu kapsamda Oregon Eyalet Üniversitesi'nde Hatfield Deniz Bilimleri Merkezi'nde çalışan bilim insanları tarafından etiketlenen Ulusal Veri Bilimi Kasesi olarak adlandırılan Kaggle plankton veri kümesi kullanılmıştır [17]. Bilim insanları yaklaşık 50 milyon plankton görüntüsünü 18 gün içinde elde etmişlerdir. Elde edilen bu veri kümesinin 30.336 resmi 2015 yılında Kaggle veri

yarışması için plankton veri kümesi olarak etiketlenmiştir. Veri kümesi, 30.336 görüntü ve 121 sınıf plankton çeşidi içermektedir.

Kaggle plankton veri kümesinden uzmanlar tarafından etiketlenemeyen görüntülere ait olan, bilinmeyen adlı sınıflar çıkarılmış ve geri kalan 118 sınıf eğitimlerde kullanılmıştır. 118 sınıfa sahip plankton veri kümesi tez kapsamında Set-1 olarak adlandırılmıştır. Ayrıca Kaggle plankton veri kümesinin ikinci alt kümesi olarak, Zheng *ve ark.* çalışmasında [18] ve benzer Kaggle plankton veri kümesi çalışmalarında kullanılan, plankton sınıflarında minimum 100 görüntüye sahip 38 sınıf seçilerek 14.374 görüntüden oluşan Kaggle plankton veri kümesinin bir alt kümesi kullanılmıştır [19], [20]. 38 sınıfa sahip plankton veri kümesi tez kapsamında Set-2 olarak adlandırılmıştır. Set-1 ve Set-2 veri kümeleri %70 eğitim, %15 doğrulama ve %15 test kümesi olarak üçe bölünmüştür. Tablo 1.1.'de kullanılan Set-1 ve Set-2 veri kümelerinin bilgileri paylaşılmıştır.

Tablo 1.1. Kullanılan Veri Kümelerinin Özellikleri

Küme Adı	#Toplam Veri	#Sınıf	#Eğitim Seti	#Doğrulama Seti	#Test Seti
Set-1	29419	118	20475	4472	4472
Set-2	14374	38	11464	1455	1455

Tez kapsamında, plankton sınıflandırma başarımını arttırmak ve uygulanan yöntemlerin etkisini gözlemlemek amacıyla kapsamlı deneyler gerçekleştirilmiştir. Toplamda altı deney yapılmıştır ve her bir deneyin amacı ve değerlendirmesi ile ilgili bilgiler paylaşılmıştır. Çalışma kapsamında Deney-1 olarak isimlendirilen deneyde, Set-1 (118 Sınıf) veri kümesine kısmi ve tam veri artırma olarak isimlendirilen iki farklı veri artırma yöntemi uygulanarak bu veri kümelerinin aktarım öğrenmesi, ince ayar ve tüm katmanların eğitilmesi yöntemleri kullanılarak eğitimi Inceptionv3 ağ modeli kullanılarak gerçekleştirilmiş ve sonuçlar elde edilmiştir. Deney-1 sonuçlarında en iyi başarıma sahip veri artırma yönteminin tam veri artırma yöntemi olduğu sonucu elde edilmiştir. Çalışma kapsamında Deney-2 olarak adlandırılan deneyde, plankton görüntülerinin farklı en-boy oranlarına sahip olmasından ve evrişimli sinir ağlarının sabit boyutta girdi görüntüsü kabul etmesinden kaynaklı çözünürlük kaybının giderilmesi amacıyla boşluk ekleme ve kare kırpma ön işlemleri uygulanmıştır ve elde edilen sonuçlar değerlendirilmiştir. Deney-2 kapsamında elde edilen sonuca göre uygulanan ön işleme yöntemleri elde edilen sonucu negatif etkilemiş ve en iyi sonuç ön işleme uygulanmamış orijinal veri kümesi eğitim sonucundan elde edilmiştir. Çalışma kapsamında Deney-3 olarak adlandırılan deneyde, Deney-1 ve Deney-2 kapsamında en iyi sonucu veren tam

veri arttırma uygulanmış Set-1 (118 Sınıf) ve Set-2 (38 Sınıf) veri kümeleri ve tüm katmanların eğitilmesi yöntemi kullanılarak Inceptionv3, InceptionResNetv2, DenseNet, ResNet, VGG-16 ağlarında eğitimler gerçekleştirilmiştir. Deney-3 kapsamında en iyi sonuç DenseNet ağında gerçekleşen eğitim sonucunda elde edilmiştir. Çalışma kapsamında Deney-4 olarak adlandırılan deneyde, Deney-3 kapsamında eğitilen modellere ortalama topluluk yöntemi uygulanmış ve her bir modelin ilgili sınıf için verdiği sonuçlar toplanarak ortalaması alınmış ve bu işlem her sınıfa uygulanmıştır. Deney-4 sonucunda Set-1 (118 Sınıf) veri kümesi sonucunda %2 başarımla %80 ağırlıklı F1-Skoru, Set-2 veri kümesi sonucunda %1 başarımla %93 ağırlıklı F1-Skoru elde edilmiştir. Çalışma kapsamında Deney-5 olarak adlandırılan deneyde, çapraz entropi kaybı ve odak kaybı fonksiyonlarının eğitimde kullanılmasıyla elde edilen sonuçlar karşılaştırılmış ve odak kaybı fonksiyonunun farklı hiperparametreleri uygulanarak eğitim gerçekleştirilmiştir. Deney-5 sonucunda test kümesinden elde edilen başarımda artış sağlanmamıştır, eğitim ve doğrulama doğruluk değerlerinde artış ve kayıp değerlerinde azalma gözlemlenmiştir. Deney-5 sonucunda odak kaybı fonksiyonunun eğitim başarımını arttırdığı fakat test başarımında herhangi bir etki göstermediği gözlemlenmiştir. Çalışma kapsamında Deney-6 olarak adlandırılan deneyde, 118 sınıfa ait DenseNet ağı üzerinde Deney-3 kapsamında Set-1 veri kümesi ile gerçekleştirilen deney sonucu elde edilen modelin hata matrisi hesaplanmıştır. Elde edilen 118x118 boyutundaki hata matrisi, aynı isim altında toplanan farklı gruplar sütun bazında toplanarak 118x33 matris elde edilmiş, sonrasında her bir sınıf içerdiği veri sayısı miktarıyla çarpılmıştır. Satır bazında da aynı isim altında toplanan farklı gruplar sütun bazında toplanarak 33x33 grup matrisi elde edilmiştir. Elde edilen 0-40 aralığındaki hata matrisi her bir ana sınıfın içerdiği toplam plankton sayısına bölünmüş ve 33 grup için 0-1 aralığında olasılık değerleri elde edilmiştir. Elde edilen bu hata matrisi sonucuyla, 33 grup olarak eğitimi gerçekleştirilen hata matrisi karşılaştırılmış ve 118 sınıftan 33 grup şekline dönüştürülerek elde edilen hata matrisinin benzer grup altındaki sınıfları daha az birbirine karıştırdığı ve daha iyi bir çözüm yolu olduğu sonucuna varılmıştır.

Tez kapsamında yapılan katkıların özeti aşağıda paylaşılmıştır:

- Tez kapsamında, dengesiz dağılımlı bir veri kümesi kullanıldığı için veri kümesi iyileştirilmeye çalışılmıştır. Veri arttırma yöntemi ve ön işleme yöntemleri uygulanmıştır.

- Önceden eğitilen evrişimli sinir ağı modelleri, aktarım öğrenmesi, ince ayar, tüm katmanların eğitilmesi yöntemleri olmak üzere toplamda üç yöntem kullanılarak ayrı ayrı eğitilmiş ve veri arttırma ve ön işleme işlemlerinin eğitime katkısı gözlemlenmiştir.
- Sınıflandırma modelinin başarımını arttırmak amacıyla ortalama topluluk öğrenimi yöntemi uygulanmıştır.
- Farklı kayıp yöntemlerinin plankton eğitimine etkisi gözlemlenmiştir.
- Benzer sınıfların gruplanması tekniği önerilmiş ve gruplanarak eğitilen model ile hata matrisleri karşılaştırılarak elde edilen sonuç analiz edilmiştir.

1.2. Tezin Kapsamı ve Akışı

Bu çalışma şu şekilde yapılandırılmıştır:

Bölüm 2’de; plankton görüntüleme sistemleri, plankton taksonomisi, plankton veri kümeleri ve plankton görüntü verilerinin sınıflandırılması ile ilgili literatür çalışmalarına odaklanmıştır. Bölüm 2.1.’de plankton görüntüleme sistemlerinde kullanılan yöntemler ve son zamanlarda kullanılan su altı görüntüleme sistemlerinden detaylı olarak bahsedilmiştir. Bölüm 2.2.’de plankton taksonomisinden bahsedilmiştir ve Kaggle plankton veri kümesinin plankton taksonomisi paylaşılmıştır. Bölüm 2.3.’te literatürde yaygın olarak kullanılan veri kümeleri incelenmiştir ve Kaggle plankton veri kümesinden detaylı olarak bahsedilmiştir. Bölüm 2.4.’de plankton sınıflandırmasının zorluklarından ve literatürde yer alan özellik tabanlı ve evrişimsel sinir ağı tabanlı çalışmalardan bahsedilmiştir.

Bölüm 3’te; evrişimli sinir ağları ve çalışma prensipleri, eğitim süreci ve bu süreçte öğrenme ve genelleme için eğitim sürecinin kullanılan tekniklerle birlikte nasıl çalıştığı ve evrişimli sinir ağlarının gerekliliği açıklanmıştır. Bölüm 3.1.’de sinir ağlarının katmanları ve yapıları detaylı olarak incelenmiştir. CNN mimarisinde kullanılan evrişimsel, havuz ve tamamen bağlı katmanlar incelenmiş ve aktivasyon fonksiyonların çeşitleri ve avantajları mercek altına alınmıştır. Tez kapsamında kullanılan ağlardan bu bölümde detaylı olarak bahsedilmiştir. Bölüm 3.2.’de sinir ağlarının eğitim aşamalarından ve eğitim sırasında kullanılan kritik öneme sahip kayıp fonksiyonu, gradyan iniş, geri yayılım ve kaybolan, patlayan gradyanlar detaylı olarak incelenmiştir.

Bölüm 4'te; Bölüm 4.1.'de tez kapsamında Kaggle plankton veri kümesine uygulanan veri ön işleme yöntemleri açıklanmıştır. Bölüm 4.2.'de başarımlı ölçütü olarak kullanılan değerlendirme metriklerinden detaylı olarak bahsedilmiştir. Bölüm 4.4.'te tez kapsamında gerçekleştirilen deney detaylarından ve gerçekleştirilen her bir deneyden ve sonuçlarından detaylı olarak bahsedilmiştir.

Bölüm 5'de elde edilen sonuçlar değerlendirilmiş ve gelecek çalışmalar ile ilgili analizler paylaşılmıştır.



2. PLANKTON TANIMA

Bu bölümde planktonlar ile ilgili detaylı bilgiler verilecektir. İlk olarak Bölüm 2.1.'de plankton görüntüleme sistemleri hakkında bilgi paylaşılacaktır. Sonrasında Bölüm 2.2.'de plankton taksanomisinin detaylarından, Bölüm 2.3.'te plankton veri kümeleri ve özelliklerinden, özellik tabanlı sınıflandırma sistemlerinden Bölüm 2.4.1.'de ve evrişimli ağ tabanlı sınıflandırma sistemlerinden Bölüm 2.4.2.'de detaylı olarak bahsedilecektir.

2.1. Plankton Görüntüleme

Su ekosistemini analiz edebilmek için suda görüntüleme oldukça önemlidir [12]. Suda görüntüleme, suda bulunan partiküllerin çeşitlilikleri, büyüklükleri ve sayıları vb. özelliklere sahip incelemeleri mümkün kılar. Planktonların dağılımının analizi su ekosisteminin sağlığı hakkında bilgi içerdiği için plankton türlerini belirlemek önemlidir. Plankton görüntüleme sistemleri, yerinde olan (in-situ) ve yerinde olmayan (ex-situ) sistemler şeklinde iki gruba ayrılır [21]. Yerinde olmayan (ex-situ) plankton görüntüleme sistemlerinde plankton görüntüsü planktonun doğal olarak bulunmadığı bir ortamda elde edilmektedir. Sudan numune alınmakta ve alınan numuneler laboratuvar ortamında incelenmekte ve plankton görüntüsü elde edilmektedir. Bu yöntem planktonun tespiti ve analizi için kullanılan en geleneksel yöntemdir. Optik mikroskop ile görüntüleme yöntemi, planktonları morfolojik özelliklerine dayanarak plankton türlerini tanımaya yönelik bir yöntemdir. Bilim insanları manuel olarak planktonları saymakta ve tanımlamaktadır. Optik mikroskoplar ile plankton tanıma süreci planktonların morfolojik yapısını ayırt edebilme uzmanlık bilgisini gerektirir ve oldukça zaman alıcıdır. Yerinde olan (in-situ) plankton görüntüleme sistemleri plankton algılaması ve görüntülemesini sağlayan sistemlerdir. Plankton numunesi toplamak amacıyla kullanılan cihazların farklı çevresel koşullarda ve boyutları değişkenlik gösteren partikülleri algılamak için gelişmiş bir optik sensöre ihtiyaçları vardır. Görüntü kalitesi büyük ölçüde aydınlatmadaki küçük farklılıklardan etkilenmektedir. Ayrıca görüntüleri işlemek ve analiz etmek için hızlı bir işlem gücüne ve yüksek depolama alanına sahip cihazlar olması gerekmektedir. Bu sorunlara çözüm niteliğinde plankton görüntülerini toplamak için yerinde ve laboratuvar da dahil olmak üzere Video Plankton Kaydedici (VPR) [22], Sualtı Video Profilleyici (UVP) [23], Gölge Görüntü Parçacı, Profil Oluşturma Değerlendirme Kaydedici (SIPPER) [24], Zooplankton Görselleştirme Sistemi (ZOOVIS) [25], Scripps Plankton Kamerası (SPC) [26],

Görüntüleme Akışı Sitobot (IFCB) [11], Yerinde Plankton Görüntüleme Sistemi (ISIIS) [27], ZooScan [16] vb. gibi birçok görüntüleme cihazı geliştirilmiştir. Kaggle plankton veri kümesi Yerinde Plankton Görüntüleme Sistemi (ISIIS) [27] cihazı kullanılarak oluşturulmuştur. Bu cihazlar oldukça fazla sayıda plankton görüntüsünü kısa süre içinde elde edebilmektedirler. Fakat elde edilen çok sayıda görüntünün elle sınıflandırılması oldukça zaman alıcı olmakla birlikte çok sayıda alanında uzman olan kişinin sınıflandırmaya zaman harcaması gerekliliğini beraberinde getirmektedir. Bu nedenle elde edilen görüntülerin sınıflandırılması için otomatik plankton sınıflandırma sistemleri oldukça gereklidir [28].

2.2. Plankton Taksonomisi

Taksonomi organizmaları ortak özelliklere dayalı adlandıran, tanımlayan ve sınıflandıran bir bilimdir [29]. Takson, oluşturulan gruba verilen isimdir ve taksonlar daha küçük taksonomik gruplara ayrılabilir. Türler hiyerarşik olarak en alt grupta bulunur ve daha alt gruba bölünemez. Su akıntısı sonucu serbestçe sürüklenen mikroorganizmalara plankton denilir ve planktonlar gelecekte muhtemelen genetik bilgilere dayalı farklı taksonlar şeklinde ifade edilecektir; fakat şu anda çoğunlukla planktonların ayrımında dış özellikleri ve ekosistemdeki rolleri kullanılmaktadır. Plankton taksonlarının sınıflandırılmasında kullanılan tipik özellikler, doku, şekil, boyut, kalonilerde bulunuyor olmaları ve kamçılı olup olmadıkları gibi özelliklerdir.

Fitoplanktonlar ve zooplanktonlar plankton gruplarından ikisidir. Fitoplanktonlar güneş ışığı kullanarak, karada bulunan bitkiler gibi fotosentez yapar ve oksijen ve besin üretirler. Zooplanktonlar ise fitoplanktonlar ile beslenen birçok farklı büyüklükte ve çok sayıda alt türler içeren çeşitli bir gruptur. Su ekosisteminde bulunan diğer canlılar da zooplankton yiyerek beslenir. Fitoplanktonlar besin zincirinin en alt seviyesinde yer aldığı ve oksijen üretiminde dünyaya önemli bir katkı sağladığı için büyük öneme sahiptir. Fitoplanktonların türlerinin belirlenmesi, tanımlanması ve isimlendirilmesini sağlayan taksonomi bilimi de bu yüzden önemlidir. Gelişen teknoloji ile birlikte fitoplankton çeşitliliğinin incelenmesi hala aktif bir çalışma alanıdır. Bugüne kadar hala bir çok fitoplankton türü taksonomik olarak sınıflandırılmayı beklemektedir. Fitoplanktonlar, ekosistemde yer aldıkları role göre sınıflandırılır. Şekline ve boyutuna (Örn. Nanoplanktonlar (2-20 μm), mikrop planktonlar (20-200 μm) vb.) göre de sınıflandırılırlar. 1980'li yılların sonunda yaklaşık 4 bin farklı deniz planktonu türü önceden sınıflandırılmıştır [30].

2.3. Plankton Veri Kümeleri

Plankton popülasyonlarını analiz etmek için kullanılan geleneksel sistemler ve yöntemler zaman alıcıdır. Geleneksel sistemler için geliştirilmiş yaklaşım su altı kamerasıdır. Geliştirilen su altı kamera sistemleriyle kısa zamanda su altından yüksek çözünürlüklü birçok görüntü elde edilebilmektedir. Elde edilen görüntüler plankton türleri ve dağılımları için analiz edilebilmektedir. Bu bölümde farklı teknolojiler kullanılarak toplanan WHOI plankton, ZooScan ve Kaggle plankton veri kümelerinden detaylı olarak bahsedilecektir. Tez kapsamında yapılan çalışmalarda Kaggle plankton veri kümesi kullanılmıştır.

2.3.1. WHOI-Plankton

Woods Hole Oceanographic Institution (WHOI) olarak adlandırılan enstitüde araştırmacılar tarafından sağlanan sınıf etiketlerine göre düzenlenmiş WHOI-Plankton veri kümesi [31] milyonlarca mikroskobik deniz planktonu görüntüsü içermektedir. Görüntüler akış görüntüleme sitobot'u (Imaging FlowCytobot (IFCB)) [11] sistemi tarafından yakalanmıştır. IFCB plankton görüntüleme sistemi WHOI'de tasarlanmış ve üretilmiştir. WHOI-Plankton veri kümesi yaklaşık 3.4 milyon plankton görüntüsünden oluşmaktadır. 103 plankton sınıfına sahiptir. WHOI-Plankton veri kümesinde planktonlar dengesiz dağılıma sahiptir. Veri kümesinin yaklaşık %85'i dino30, cylindrotheca, rhizosolenia ve chaetoceros dahil olmak üzere altı nadir taksonda yoğunlaşmıştır. Bu durum plankton taksonlarının deniz ortamında dengesiz olarak dağıldığının bir kanıtı niteliğindedir. WHOI-Plankton veri kümesi¹ açık kaynaklıdır, 2006-2014 yılları arasında her bir yıl için paylaşılan veri kümesine erişilebilir. WHOI-Plankton veri kümesinden alınan örnek görüntüler Şekil 2.1.'de paylaşılmıştır.

¹WHOI-Plankton veri kümesi web sayfası:

<https://darchive.mblwhoilibrary.org/handle/1912/7341>

<https://whoigit.github.io/whoi-plankton/>



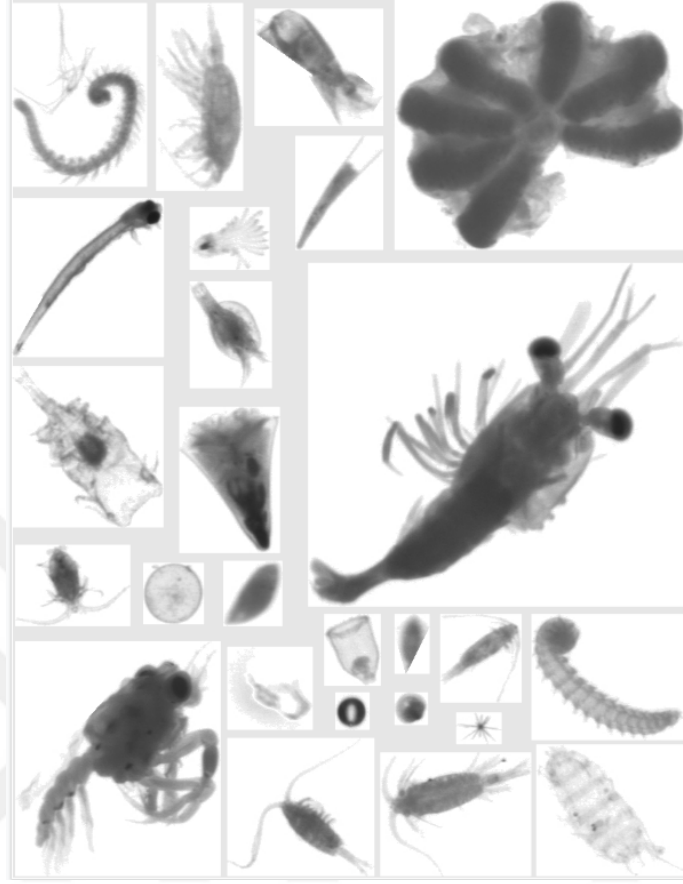
Şekil 2.1. WHOI Plankton Veri Kümesinden Alınan Örnek Görüntüler

2.3.2. ZooScan

ZooScanNet veri kümesi² ZooScan sistemi³ tarafından Ağustos 2007 ve 8 Ekim 2008 tarihleri arasında Fransa'nın Villefranche-sur-mer Körfezi'nden toplanan 1.433.278 görüntüden ve 93 taksondan oluşan bir veri kümesidir [16]. Çoğu kategori zooplankton ve zooplankton yumurtalarıdır; kalan kategoriler zooplankton olmayan ve görece kötü odaklı görüntülerdir. ZooScanNet veri kümesinden örnek plankton görüntüleri Şekil 2.2.'de paylaşılmıştır.

²ZooScanNet veri kümesi web sayfası: <https://www.seanoe.org/data/00446/55741/>

³ZooScanNet sistemi web sayfası: <http://www.zooscan.com>

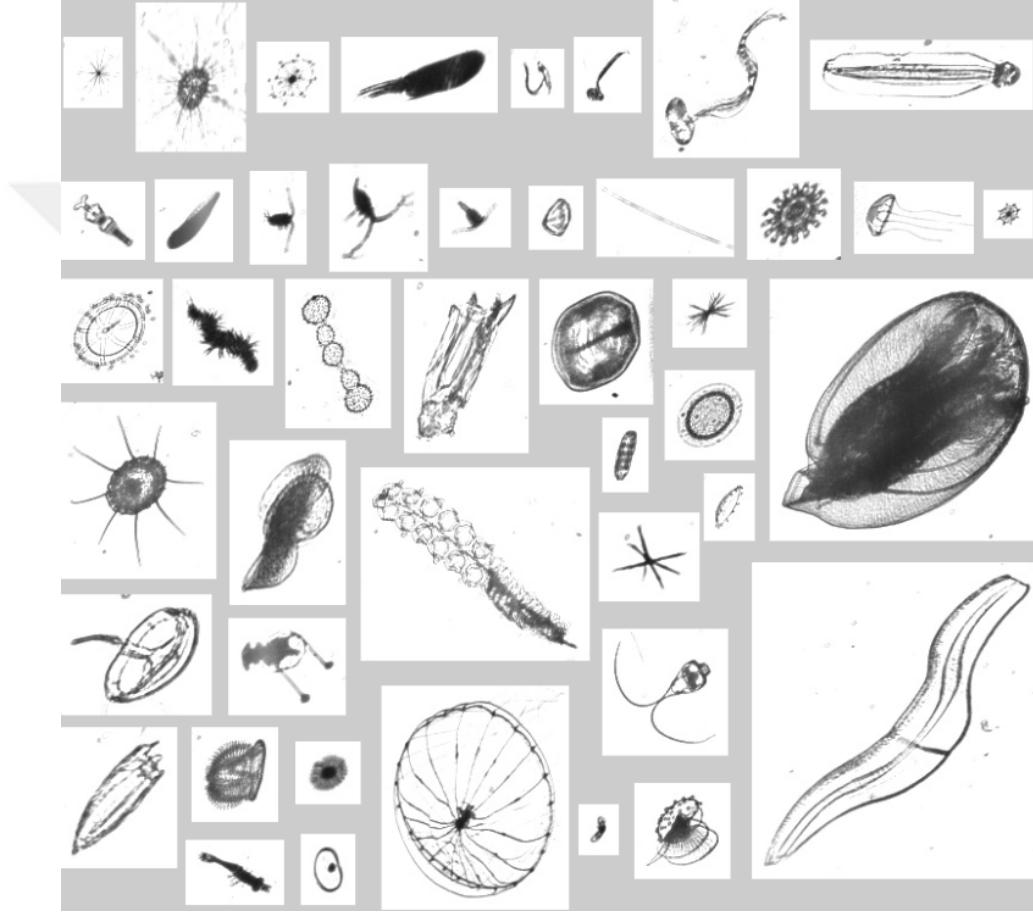


Şekil 2.2. ZooScan Plankton Veri Kümesinden Alınan Örnek Görüntüler

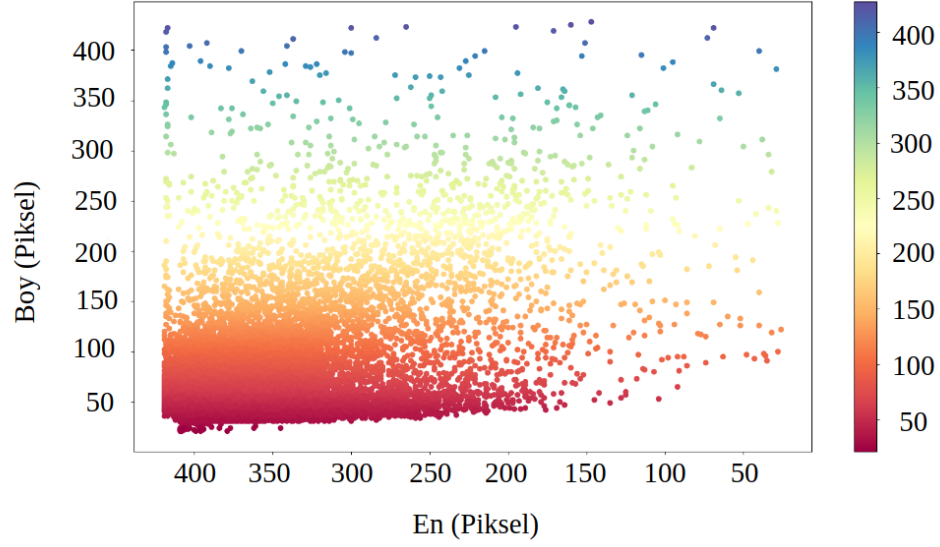
2.3.3. Kaggle Plankton

Oregon Eyalet Üniversitesi'nin Deniz Bilimleri Merkezi'nde bilim insanları 18 günlük bir süre içinde yaklaşık 50 milyon plankton fotoğrafı elde ettiler. Bu veri kümesi yerinde plankton görüntüleme sistemi (ISIIS) teknolojisi kullanılarak elde edilen büyük ölçekli bir veri kümesidir. Elde edilen bu görüntülerin 30.336'sı uzmanlar tarafından etiketlenmiştir. 2015 yılında Ulusal Veri Bilimi Kasesi (National Datascience Bowl) [17] olarak adlandırılan Kaggle veri bilimi yarışmasında organizatörler tarafından 30.336 etiketli ve 130.400 etiketsiz test görseli kullanıma sunuldu. Veri kümesi, tek hücreli en küçük protisler, kopedodlar ve lavra balıkları dahil birçok farklı tür içeren 121 farklı türden oluşmaktadır. Veri kümesi 3 boyutlu uzayda farklı yönlerden türlerin çekilmesiyle elde edilmiştir. Elde edilen görüntüler gri tonlamalı, beyaz zemin üzerine siyah rengindedir ve çok değişken boyutlara sahiptir. Kaggle plankton veri kümesinin örnek görüntüleri Şekil 2.3.'te paylaşılmıştır. Kaggle plankton veri

kümesinde sınıflar dengesiz dağılmıştır. Tablo 2.3.'te görüleceği gibi bazı sınıflar 1000'den fazla görüntü içermektedirken bazı sınıflar yalnızca bir görüntü içermektedir. Plankton görüntülerinin boyutlarının piksel cinsinden yükseklik ve genişlik cinsinde dağılım grafiği Şekil 2.4.'te verilmiştir. Grafikteki her nokta bir görüntüyü temsil etmektedir. Renk, çevreleyen alandaki yoğunluğu gösterir. Şekilde görüldüğü gibi plankton görüntülerinin boyutları arasında belirgin farklılık vardır.



Şekil 2.3. Kaggle Plankton Veri Kümesinden Alınan Örnek Görüntüler



Şekil 2.4. Kaggle Plankton Veri Kümesindeki Görüntülerin Piksel Cinsinden Dağılım Grafiği

Tez kapsamında Kaggle plankton veri kümesinin iki farklı sürümü kullanılmıştır. Kullanılan ilk sürümde, 121 sınıf içeren veri kümesinin 'bilinmeyen' isimli üç sınıfı, bir uzmanın makul bir kesinlikle sınıflandıramadığı örnekler içerdiğinden veri kümesinden çıkarılmıştır. Kullanılan ilk sürüm, Set-1 olarak isimlendirilmiştir. Set-1 ile ilgili detaylar Tablo 2.1.'de paylaşılmıştır. Kullanılan ikinci sürüm, Set-2 olarak isimlendirilmiştir. Set-2, Kaggle plankton veri kümesinin [18] yazarları tarafından seçilen bir alt kümesidir. En az 100 örnek bulunduran plankton sınıfları seçilmiştir. Set-2 ile ilgili detaylar Tablo 2.2.'de paylaşılmıştır.

Tablo 2.1. Set-1 Veri Kümesinin Özellikleri

Set-1 Kaggle Veri Kümesi				
#Toplam Veri	#Sınıf	#Eğitim Seti	#Doğrulama Seti	#Test Seti
29419	118	20475	4472	4472

Tablo 2.2. Set-2 Veri Kümesinin Özellikleri

Set-2 Kaggle Veri Kümesi				
#Toplam Veri	#Sınıf	#Eğitim Seti	#Doğrulama Seti	#Test Seti
14374	38	11464	1455	1455

Tablo 2.3. Kaggle Veri Kümesindeki Sınıflar ve Her Sınıftaki Veri Sayısı (#)

Sınıf İsmi	#	Sınıf İsmi	#
acantharia_protist	889	hydromedusae_haliscera_small_sideview	9
acantharia_protist_big_center	13	hydromedusae_liriope	19
acantharia_protist_halo	71	hydromedusae_narco_dark	23
amphipods	49	hydromedusae_narcomedusae	132
appendicularian_fritillarida	16	hydromedusae_narco_young	336
appendicularian_slight_curve	532	hydromedusae_other	12
appendicularian_s_shape	696	hydromedusae_partial_dark	190
appendicularian_straight	242	hydromedusae_shapeA	412
artifacts	393	hydromedusae_shapeA_sideview_small	274
artifacts_edge	170	hydromedusae_shapeB	150
chaetognath_non_sagitta	815	hydromedusae_sideview_big	76
chaetognath_other	1934	hydromedusae_solmaris	703
chaetognath_sagitta	694	hydromedusae_solmundella	123
chordate_type1	77	hydromedusae_typeD	43
copepod_calanoid	681	hydromedusae_typeD_bell_and_tentacles	56
copepod_calanoid_eggs	173	hydromedusae_typeE	14
copepod_calanoid_eucalanus	96	hydromedusae_typeF	61
copepod_calanoid_flatheads	178	invertebrate_larvae_other_A	14
copepod_calanoid_friillyAntennae	63	invertebrate_larvae_other_B	24
copepod_calanoid_large	286	jellies_tentacles	141
copepod_calanoid_large_side_antennatucked	106	polychaete	131
copepod_calanoid_octomoms	49	protist_dark_center	108
copepod_calanoid_small_longantennae	87	protist_fuzzy_olive	372
copepod_cyclopoid_copilia	30	protist_noctiluca	625
copepod_cyclopoid_oithona	899	protist_other	1172
copepod_cyclopoid_oithona_eggs	1189	protist_star	113
copepod_other	24	pteropod_butterfly	108
crustacean_other	201	pteropod_theco_dev_seq	13
ctenophore_cestid	113	pteropod_triangle	65
ctenophore_cydippid_no_tentacles	42	radiolarian_chain	287
ctenophore_cydippid_tentacles	53	radiolarian_colony	158
ctenophore_lobate	38	shrimp_caridean	49
decapods	55	shrimp-like_other	52
detritus_blob	363	shrimp_sergestidae	153
detritus_filamentous	394	shrimp_zoea	174
detritus_other	914	siphonophore_calycophoran_abyllidae	212
diatom_chain_string	519	siphonophore_calycophoran_rocketship_adult	135
diatom_chain_tube	500	siphonophore_calycophoran_rocketship_young	483

echinoderm_larva_pluteus_brittlestar	36	siphonophore_calycophoran_sphaeronectes	179
echinoderm_larva_pluteus_early	92	siphonophore_calycophoran_sphaeronectes_stem	57
echinoderm_larva_pluteus_typeC	80	siphonophore_calycophoran_sphaeronectes_young	247
echinoderm_larva_pluteus_urchin	88	siphonophore_other_parts	29
echinoderm_larva_seastar_bipinnaria	385	siphonophore_partial	30
echinoderm_larva_seastar_brachiolaria	536	siphonophore_physonect	128
echinoderm_seacucumber_auricularia_larva	96	siphonophore_physonect_young	21
echinopluteus	27	stomatopod	24
ephyra	14	tornaria_acorn_worm_larvae	38
euphausiids	136	trichodesmium_bowtie	708
euphausiids_young	38	trichodesmium_multiple	54
fecal_pellet	511	trichodesmium_puff	1979
fish_larvae_deep_body	10	trichodesmium_tuft	678
fish_larvae_leptocephali	31	trochophore_larvae	29
fish_larvae_medium_body	85	tunicate_doliolid	439
fish_larvae_myctophids	114	tunicate_doliolid_nurse	417
fish_larvae_thin_body	64	tunicate_partial	352
fish_larvae_very_thin_body	16	tunicate_salp	236
heteropod	10	tunicate_salp_chains	73
hydromedusae_aglaura	127	unknown_blobs_and_smudges	317
hydromedusae_bell_and_tentacles	75	unknown_sticks	175
hydromedusae_h15	35	unknown_unclassified	425
hydromedusae_haliscera	229		

2.4. Plankton Tanıma Sistemi

Plankton görüntülerinin elde edilme aşaması su ekosisteminin dinamik doğası gereği ve planktonların mikroskobik boyutu nedeniyle zordur. Elde edilen plankton görüntülerinin manuel olarak sınıflandırılması zaman alıcı, uzmanlık bilgisi gerektiren, pahalı ve hataya açık bir çözümdür. Bu sebeplerden dolayı birçok araştırmacı bilgisayarlı görü tekniklerini kullanarak bu süreci otomatik hale getirecek yöntemler araştırmıştır. Otomatik plankton sınıflandırması [8, 18, 32, 33] üç temel nedenden dolayı zor kabul edilmektedir:

(i) Plankton görüntüleri genellikle düşük çözünürlüktedir, benzeyen türler arasında düşük çözünürlükten dolayı bilgi kaybı olmakta ve bu da ayrımı zorlaştırmaktadır. Uzmanlar tarafından manuel sınıflandırma da bu nedenden dolayı hata payına sahiptir.

(ii) Taksonomiye özel zorluklar sunan çok çeşitli filogenetik türleri içerir.

(iii) Uzmanlar tarafından manuel olarak sınıflandırılan ve sonrasında otomatik plankton sınıflandırma sistemi için eğitim, doğrulama ve test kısmında kullanılacak olan plankton veri kümeleri oldukça dengesiz bir veri dağılımına sahiptir.

Otomotik plankton sınıflandırmasında çalışmalar temel olarak 2 farklı yaklaşıma sahiptir:

(1) Özniteliklerin el yapımı tanımlayıcılar ile çıkarılmasından sonra sınıflandırmanın destek vektör makinesi, rassal orman vb. yöntemlerle gerçekleştirilmesi [8, 18].

(2) Evrişimli sinir ağları temelli derin öğrenme yaklaşımlarıyla sınıflandırmanın gerçekleştirilmesi [32–34].

2.4.1. Öznitelik Çıkarımı Tabanlı Plankton Tanıma

Evrişimli sinir ağları (CNN) son zamanlarda plankton tanıma sistemi de dahil bir çok problemin çözümünde yaygın olarak kullanılmaktadır. Plankton tanıma sistemlerinin evrişimli sinir ağ tabanlı uygulamaları Bölüm 2.4.2.'de detaylı olarak ele alınmıştır. Evrişimli sinir ağları yaygınlaşmadan önce otomatik plankton sınıflandırma sistemleri özellik tabanlı yöntemlerle ele alınmıştır. Özellik tabanlı yöntemler belirli bir örneğin geometrik şekil, alan, uzunluk, genişlik, renk histogramı gibi hesaplanan bir dizi tanımlayıcı özelliği üzerinde çalıştıkları için "öznitelik çıkarımı tabanlı" olarak adlandırılmışlardır. Elde edilen öznitelik vektörleri daha sonra destek vektör makineleri (SVM) [35], rassal ormanlar [36], k-en yakın komşular, çok katmanlı algılayıcı (MLP) [36] gibi özellik tabanlı sınıflandırıcılara girdi olarak sağlanır.

Otomatik plankton tanıma sistemlerinde, öznitelik çıkarımı tabanlı ilk göze çarpan uygulamalardan biri Tang ve arkadaşlarının 1998 yılında yaptığı çalışmadır [15]. Tang ve arkadaşları, geleneksel değişmez moment özellikleri ve Fourier sınır tanımlayıcıları gri ölçekli morfolojik granülometrelerle birleştirerek plankton görüntülerinin hem şekil hem de doku bilgilerini elde eden öznitelik vektörü elde etmişlerdir. Altı farklı plankton taksonu içeren yaklaşık 2000 görüntüde %95 oranında sınıflandırma doğruluğu elde etmişlerdir. Birlikte oluşturma matrisi doku tanımlayıcılarından biridir. Bu tanımlayıcı kullanılarak 2005 yılında Hu ve Davis tarafından otomotik plankton tanıma sistemi elde etmek amacıyla destek vektör makinesi (SVM) eğitilmiştir [37]. Li ve diğerleri plankton sınıflandırmasının; sınıf içi çeşitlilik, sınıflar arası büyük benzerlik ve gürültülü veriler sebebiyle zor bir problem olduğundan bahsetmiştir [38]. Yapılan çalışmada daha düşük boyutluluğa indirgenmiş tanımlayıcı kullanılmıştır ve sınıflandırıcı olarak ayırıştırma (diskriminant) analiz tekniğine dayalı

ikili sınıflandırmanın ayrıştırılmasından ve istenmeyen bilgiler çıkarıldıktan sonra füzyon kuralı kullanılarak birleştirilmesinden oluşan bir çözüm geliştirilmiştir. Ellen *ve diğerleri* [39], zooplankton görüntüleri üzerinde birkaç farklı yöntem denemişlerdir ve aralarında en iyi başarıyı sağlayan yöntemi bulmaya çalışmışlardır. Zooplankton veri kümesinde en iyi performans gösteren algoritmalar gradyan destekli rastgele orman sınıflandırması (GBC) ve destek vektör makinesi-radyal tabanlı fonksiyon (SVM-RBF) olarak belirlemişlerdir. İki algoritmanın birleştirilmesiyle zooplankton sınıflandırma sonucunun daha da iyileştiğini gözlemlemişlerdir [39]. Bi *ve diğerleri* [40], bulanık sulardan elde edilen görüntülerde her bir görüntü içindeki plankton nesnelerini bulmak için bir segmentasyon prosedürü geliştirmiş ve sınıflandırma için el yapımı özellikler kullanarak yarı otomatik plankton tanıma yaklaşımı geliştirmişlerdir. Chang *ve diğerleri* [41], iki özellik çıkarma yöntemi olan hızlandırılmış sağlam özellikler-temel bileşen analizi (SURF-PCA) ve yerel ikili görüntü-temel bileşen analizini (LPB-PCA) fitoplanktonu tanıma amacıyla birleştirmişlerdir. Algoritmada hesaplama karmaşıklığını azaltmak ve yüksek sınıflandırma başarıyı elde etmek için temel bileşen analizi (PCA) algoritmasını kullanmışlardır. Deneyleri birkaç sınıf içeren küçük bir veri kümesi üzerinde gerçekleştirmişlerdir. Thi-Thu-Hong Phan *ve diğerleri* [42] fitoplankton sınıflandırması için farklı özellik tiplerini ve sınıflandırıcıları incelemişlerdir. Kullanılan veri kümesi, biyologlar tarafından etiketlenen her bir sınıfın 100 örnek içerdiği 7 sınıftan oluşmaktadır. El yapımı özellikler olarak plankton uzunluğu, iç yapısı, klorofil pigmenti vb. özellikler kullanılmıştır. K en yakın komşular, destek vektör makineleri, rassal orman sınıflandırıcıları kullanılmıştır. Yapılan tüm deney sonuçlarında rassal orman sınıflandırıcısı %98.24'lük bir puanla en iyi doğruluğa sahip sınıflandırıcı olarak tespit edilmiştir. Gloria Bueno *ve diğerleri* [43] okyanustaki en yaygın plankton türü olan planktonları sınıflandırma amacıyla farklı el yapımı tanımlayıcıları ve sınıflandırıcıları incelemişlerdir. İspanya'da toplanan, her sınıfta ortalama 100 örnek bulunan ve toplam 80 sınıftan oluşan plankton veri kümesi kullanmışlardır. Morfolojik, istatistiksel ve dokusal tanımlayıcılara ek olarak yerel ikili örüntü (LBP) ve Gabor uygulamasından faydalanmışlardır. Torbalama karar ağaçları ve tanımlayıcı kombinasyonları kullanarak %98.11 sınıflandırma doğruluğu elde etmişlerdir. Zheng *ve diğerleri* [18], özelliklerin kombinasyonuna dayalı çoklu çekirdek öğrenme sınıflandırıcısına dayanan otomatik plankton görüntü sınıflandırma sistemi önermiştir. Görüntüler ilk olarak kalitelerini arttırabilmek amacıyla ön işleme tabi tutulmuştur. Geometrik, doku ve yerel özellikler ile öznetelikler elde edilmiş sonrasında çoklu çekirdek sınıflandırma için en uygun öznetelikleri bulmak amacıyla öznetelik seçimi gerçekleştirilmiştir.

Özellik tabanlı yöntemler tarafından çıkarılan özellikler veri kümesi bağımlıdır. Bu durum

veri kümesi değıştikçe algoritmanın ve çıkarılan özelliklerin değışmesini gerektirmektedir. Bundan dolayı esnek bir yapıya sahip değildir. Esnek otomatik plankton tanıma sistemleri evrişimli sinir ağıları ile yapılmaktadır. Bu husustan bir sonraki bölümde detaylı olarak bahsedilmiştir.

2.4.2. Evrişimli Sinir Ağı Tabanlı Plankton Tanıma

Görüntülerdeki planktonik parçacıkları otomatik olarak tespit eden ve kaydeden sistemler son 20 yılda hızla gelişim göstermiştir. Bu sistemlere Video Plankton Kaydedici [44], FlowCytobot [11], FlowCam [45] ve ZooProcess [13] vb. ekipmanlar örnek verilebilir. Geliştirilen teknolojik ekipmanlar sayesinde plankton veri hacmi hızla büyümektedir. Planktonlar besin zincirinin en alt basamağında yer almakta ve öncelikle su ekosistemindeki yaşam için dolaylı olarak da diğer ekosistemdeki yaşamlar için kritik öneme sahiptir. Bundan dolayı elde edilen verilerin en iyi şekilde kullanılarak en yüksek doğruluğa sahip plankton sınıflandırıcısı elde edebilmek araştırmacılar için bir araştırma konusu olmuştur. Plankton tanıma ile ilgili birçok çalışma mevcuttur. Son zamanlarda plankton tanıma çalışmalarındaki en son eğilim derin öğrenmeye dayanmaktadır [46]. Sualtı görüntü analizi için en çok kullanılan yöntemlerden biri olan derin öğrenme sualtı deniz canlılarını algılama ve tanıma [47] [48] için evrişimli sinir ağıları (CNN) kullanılmaktadır. Evrişimli sinir ağıları (CNN) [49] optimal görüntü özelliklerini ve sınıflandırma ağırlıklarını öğrenebilen, pikseller arasında uzamsal bağlam ve ağırlık paylaşımını birleştiren insan görsel algılama mekanizmasına benzeyen çok katmanlı sinir ağılarıdır [19]. Evrişimli sinir ağılarının Bölüm 2.4.1.'de bahsedilen öznetelik çıkarımı tabanlı yöntemlere göre büyük bir avantajı üstün öznetelik çıkarma yeteneğidir. Öznetelik çıkarımı tabanlı yöntemlerin veri kümesi bağımlı olan ve alanında uzmanlık gerektiren öznetelik çıkarma ihtiyaçlarına kıyasla evrişimli sinir ağıları (CNN) daha esnek ve kullanılabilir bir yöntem sağlamaktadır. Bu açıdan evrişimli sinir ağı tabanlı derin öğrenme yaklaşımının öznetelik çıkarımı tabanlı yaklaşımlardan daha üstün olduğu gösterilmiştir [50].

Derin öğrenme yöntemi kullanılarak sualtı görüntü analizinin yapıldığı en eski çalışmalar 2015 yılına kadar uzanmaktadır. 2015 yılında Kaggle platformunda plankton görüntü tanıma sürecini otomatikleştirmek amacıyla bir veri bilimi yarışması olan Ulusal Veri Bilimi Kasesi [17] düzenlenmiştir. Bu yarışmanın amacı planktonun gri tonlamalı görüntülerini 121 sınıftan birine en yüksek doğruluğu elde ederek sınıflandırmaktır. Yarışmada kullanılan Kaggle

veya NDSB [17] diye adlandırılan plankton veri kümesi, Florida Boğazı'nda ISIIS [27] kullanılarak toplanan veri kümesinin bir alt kümesidir. Kaggle plankton veri kümesi [51] sınıf dengesizliği veri sorunu olan, farklı plankton türlerine karşılık gelen 121 sınıf, 30.336 gri tonlamalı etiketli veriden oluşmaktadır. Yaklaşık 130.400 görüntü etiketsiz bir şekilde test verisi olarak paylaşılmıştır.

Yarışmanın kazananı [52], sualtı kamerası ile elde edilen plankton görüntülerinin doğru bir şekilde sınıflandırılmasını gerçekleştirmek amacıyla DeepSea adlı bir derin öğrenme mimarisi geliştirmiştir. DeepSea, döngüsel çeşitliliğe karşı dayanıklılığını arttırmak için tasarlanmış ve mevcut ağ mimarilerine kolayca eklenebilen dört yeni katman kullanarak rotasyona karşı daha dayanıklı bir çerçeve sunmuştur. Bu çalışmada veri kümesi, doğrulama ve eğitim kümesi olarak sırasıyla 3,037 ve 27,299 görüntüye bölünmüştür. Etiketsiz olarak paylaşılan 130.400 görüntü test verisi olarak kullanılmıştır. DeepSea mimarisi ile birçok model eğitilmiş ve en iyi sonuç yaklaşık 40 farklı modelin topluluğundan elde edilmiştir. Bu çalışma ile, doğrulama kümesinde yaklaşık %82 oranında doğruluk ve %98'in üzerinde ilk 5 doğruluk elde ederek yarışmada ilk sırada yer almışlardır. 2016 yılında Py ve diğerleri [53] plankton sınıflandırması için GoogleNet'ten ilham alan derin sinir ağı modeli önermiştir. Önerilen modelde, derin sinir ağlarında derine inerken başarımların azaltılması amacıyla üç yaklaşım sunmuşlardır. Bu çalışma ile, her bir evrişim katmanı için öğrenme kapasitesi c-metriği ölçümüyle hesaplanmış ve daha karmaşık yapıları öğrenebilmesi için belirli bir seviyede öğrenme kapasitesi garanti etmişlerdir. İkinci katkı olarak, ilk evrişimli katmanın alıcı alanının giriş görüntüsünden daha büyük olmaması sağlanmıştır. Bu özellik, ağı bilgi kaybını azaltmış ve üst düzey özellikleri öğrenmesini sağlamıştır. Son katkı olarak çeşitli boyutlardaki plankton görüntülerinin sınıflandırma başarımını arttırmak için başlangıç katmanı geliştirmişler ve bu katman ile 32x32, 48x48, 64x64, 96x96 ve 128x128 boyutlarında giriş görüntüsünden çok ölçekli özellikler çıkarmışlardır. Py ve diğerleri [53] elde ettikleri modelin deneylerini Kaggle plankton veri kümesinde gerçekleştirmiş ve softmax kayıp metriği kullanarak diğer ağlar ile sonuçlarını karşılaştırmışlardır. Önerilen model, GoogleNet ve VggNet'den daha düşük softmax kaybına sahiptir. Cui ve Lui [54], otomatik plankton sınıflandırma sistemine derin kalıntı ağlarını Kaggle veri kümesi kullanarak uygulamışlardır. Deneysel sonuçlarda ilk 5 doğruluğu 0.958 ve ilk 1 doğruluğu 0.731 olarak belirtmişlerdir ve derin kalıntı ağlarının daha iyi bir şekilde genelleştirme yaptığını ve derin sinir ağlarının önemini vurgulamışlardır. Jiangpeng Yan, Xiu Li ve Zuoying Cui [55] plankton sınıflandırması için farklı ağ mimarilerini keşfederek daha verimli model elde etmeye odaklanmıştır. CaffeNet, VggNet-19 ve ResNet gibi değişen sayıda katmana sahip ağlar kullanmışlardır.

1000 sınıf içeren ILSVRC2012 veri kümesi ile önceden eğitilmiş modeller kullanarak ince ayar yöntemi uygulamışlar ve Kaggle veri kümesi üzerinde sınıflandırma sonuçlarını iyileştirmeyi amaçlamışlardır. Paylaşılan deney sonuçlarına göre, daha küçük (yalnızca 1,5 MB), daha hızlı (32.2 fps) ve %96'lık bir ilk 5 doğruluğa sahip daha verimli bir ağ mimarisi elde etmişlerdir.

Lumini ve Nanni [19], 2019 yılında yaptıkları çalışmada farklı derin öğrenme yöntemlerinin birleştirilmesine dayanan otomatik bir plankton tanıma sistemi hakkında detaylı bir çalışma sunmuştur. Bu çalışmada, derin öğrenilmiş birçok model kullanılarak ince ayar yöntemi ve aktarım öğrenmesi yöntemleri ayrı ayrı incelenmiştir. Sınıflandırıcı topluluğu tasarlanması için, farklı eğitim yöntemleri ve yaklaşımlarını değerlendirmişlerdir. Çalışma kapsamında; AlexNet, GoogLeNet, InceptionV3, VGG16, VGG19, ResNet50, ResNet101, DenseNet ve SqueezeNet olmak üzere toplam dokuz farklı ağ ve füzyonlarını karşılaştırmışlardır. Eğitim veri sayısının az olması nedeniyle aşırı uyumu (overfitting) önlemeye yardımcı olan aktarım öğrenmesi yöntemini kullanmışlardır. Görüntülerde ön işleme yöntemleri olarak kare ve boşluk ekleme yöntemlerini kullanmışlardır. Ayrıca aktarım öğrenmesinde en iyi özellik vektörlerini çıkaracak katmanları seçmek için destek vektör makinesi kullanan "sıralı kayan ileri seçim" (SFFS) yöntemini [56] kullanmışlardır. Deneyleri, plankton veri kümesi olarak en iyi bilinen Kaggle, ZooScan ve WHOI veri kümelerinde gerçekleştirmişlerdir. Kaggle veri kümesi, 121 plankton sınıfından oluşmaktadır. Fakat bu çalışmada, [18] çalışmasında kullanıldığı gibi 100'den daha fazla görüntüye sahip 38 sınıf seçilerek 14.374 görüntüden oluşan Kaggle veri kümesinin bir alt kümesi kullanılmıştır. Seçilen sınıfların her birinde min 108, max 1979 görüntü bulunmaktadır. Bu yöntemle kontrollü bir şekilde seçilen sınıf eğitimlerinde başarımlar oldukça pozitif etkilenmiştir. Kaggle veri kümesi ile araştırma yapılırken bu husus göz önüne alınarak sonuçlar karşılaştırılmalıdır. Bu çalışmada, deneyler sonucunda en iyi ağın DenseNet olduğuna karar verilmiş ve topluluk yönteminin başarımlarını arttırdığı gözlemlenmiştir. Deney sonuçlarında F1-Skor parametresi kullanılarak karşılaştırma yapılmıştır.

Lee ve diğerleri [33], 3 milyondan fazla veriye sahip ve sınıf dengesizliği olan WHOI plankton veri kümesinde sınıf dengesizliği probleminin üstesinden gelmek amacıyla aktarım öğrenmesi tabanlı evrişimli sinir ağı modeli önermiştir. Sınıf dengesizliği problemini çözmek amacıyla veri kümesinde bulunan büyük veriye sahip sınıflar için veri eşikleme ile sınıf normalleştirilmiş veriler oluşturmuşlar ve normalleştirilen verilerle evrişimli sinir ağı modelini eğitmişlerdir. Bu şekilde, küçük boyutlu sınıfların doğruluk oranlarını arttırmışlardır. Eğitilen modeli kullanarak orijinal tam verilerle evrişimli sinir ağını tekrar eğitmişlerdir. Önerilen

model ile hem az veriye sahip olan sınıflarda, hem de fazla veriye sahip olan sınıflarda başarımlar korunmuştur. Dai ve diğerleri [57], 2017 yılında yaptıkları çalışmada evrişimli sinir ağına girdi olarak orijinal görüntüyü ve iki ön işleme işlemine tabi tutulmuş versiyonunu alan hibrit 3 kanallı CNN yöntemi önermişlerdir. WHOI-plankton veri kümesinin 1000'den fazla örnek içeren 30 sınıfını alarak yeni bir veri alt kümesi oluşturmuşlardır. Görüntülerden yerel ve genel özellikler elde etmek amacıyla ön işleme yöntemlerini kullanmışlardır. Elde ettikleri hibrit yapıyı AlexNet ve GoogleNet üzerinde deneyerek yaklaşık %1 başarımlar artışı bildirmişlerdir.

Plankton sınıfları farklı şekillere sahiptir ve benzer şekillere sahip olan sınıflar dokuda farklılık göstermektedir. Bu nedenle Jinna Cui, Bin Wei ve diğerleri [58], plankton sınıflandırmasında doğruluk oranını arttırmak için birden fazla özellik kullanımının faydalı olabileceğini düşünmüştür. Plankton sınıflandırması için görüntü ön işleme yöntemlerinden biri olan Gauss filtreleme yöntemi ile elde edilen farklı girdileri birleştirmişlerdir. Elde edilen görüntüleri, AlexNet evrişimli sinir ağına eğitmişlerdir. Önerilen model %94.32'lik bir sınıflandırma başarımına sahiptir. Ön işleme uygulanmamış orijinal görüntüler üzerinde başarımı %93.58 olarak elde etmişlerdir. Model azınlık sınıflarında daha iyi performans göstermiştir.

Farklı boyutlardaki görüntülerde plankton tanıma [59] adlı çalışma 2021 yılında yayımlanmıştır. Evrişimli sinir ağı tabanlı yöntemler girdi olarak sabit boyutlu görüntü kabul etmektedir. Bu nedenle, plankton tanıma yöntemlerinin çoğunluğu ilk aşama olarak görüntüyü yeniden boyutlandırmaktadır. Fakat bu yöntem en boy oranı çok çeşitli boyutlarda olan plankton veri kümesinde bilginin kaybolmasına neden olmaktadır. Bu çalışmada plankton görüntü boyutundaki en-boy oranındaki aşırı çeşitliliği ele alan çeşitli yaklaşımlar incelenmiştir ve dört yöntem denenmiştir: uzamsal piramid havuzu, meta veri dahil etme, yama kırpma ve çoklu akış (multistream) ağları. Çoklu akış ağları ve yama kırpma yöntemlerini birleştirerek daha yüksek doğruluk oranına sahip sonuçlar elde etmişlerdir. Tablo 2.4.'te yayın adı, kullanılan veri kümesi, veri sayısı, sınıf sayısı, sonucun değerlendirildiği metrik ve sonuç bilgilerini içeren literatür özet tablosu paylaşılmıştır.

Tablo 2.4. Literatür Özet Tablosu

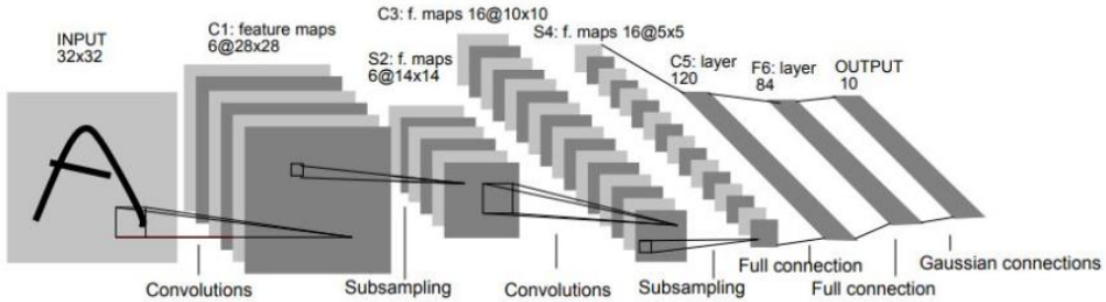
Yayın Adı	Kullanılan Veri Kümesi	#Veri	#Sınıf	Sonuç Tipi	Sonuç
[58]	WHOI	30.000 3.6 milyon	30 103	Doğruluk	%96.58 %94.32
[57]	WHOI	30.000	30	Doğruluk	%96.3
[33]	WHOI	3.6 milyon	103	Doğruluk	%92.8
[54]	Kaggle-plankton	30.336	121	Top-1 Doğruluk	%73.1
[19]	Kaggle-plankton	14.374	38	F-Ölçütü	%92.6
	ZooSCan	3771	20		%90
	WHOI	6600	22		%95
[53]	Kaggle-plankton	30.336	121	Softmax Kaybı	0.61
[55]	Kaggle-plankton	30.336	121	Top-1 Doğruluk	%76.4
[18]	Kaggle-plankton	14.374	38	F-Ölçütü	%76.9
	ZooSCan	3771	20		%82.1
	WHOI	6600	22		%88.3

Bu bölümde plankton görüntüleme sistemlerinden, plankton taksonomisinden ve plankton tanıma için kullanılan veri kümeleri ve plankton sınıflandırması ile ilgili yapılan çalışmalardan bahsedilmiştir. Plankton sınıflandırma amacıyla izlenen adımlar özetle: plankton veri kümesinin belirlenmesi, plankton veri kümesinin iyileştirilmesi için görüntüye ön işleme uygulanması, sınıflandırma algoritmasının seçimi, seçilen modelin belirlenen veri kümesiyle eğitilmesi, eğitim sonucunda elde edilen modelin değerlendirilmesi için metrik belirlenmesi ve belirlenen metrik ile test kümesi kullanılarak sonuçların değerlendirilmesi adımlarından oluşmaktadır.

3. EVRİŞİMLİ SİNİR AĞLARI

3.1. Sinir Ağları Katmanları ve Yapıları

Karmaşık problemleri çözmek için yaygın olarak kullanılan evrişimli sinir ağları derin özellik çıkarmada ve sınıflandırmada kullanılan derin öğrenme yaklaşımıdır. Evrişimli sinir ağları ilk olarak 1980 yılında Fukushima [60] tarafından kavramsallaştırıldı. Fukushima temel bir görüntü tanıma sinir ağı olan neocognitron'u icat etti. Yann LeCun ve arkadaşları [61], bu çalışmanın üzerine inşa ederek geri yayılım öğrenme algoritmasını uyguladı ve evrişimli sinir ağlarının temelini atmış oldu. Evrişimli sinir ağları geleneksel yöntemlerde elle tasarlanan özellikleri/filtreleri yeterli eğitimle öğrenme yeteğine sahiptir. Görüntü içeriği için daha yüksek temsiller çıkarılmasına izin veren evrişimli sinir ağları, görüntü özelliklerinin uzmanlar tarafından tanımlandığı klasik yöntemlerin aksine, görüntünün ham piksellerini alır, modeli eğitir ve daha iyi sınıflandırma için özellikleri otomatik çıkarır.



Şekil 3.1. Örnek CNN Mimarisi [1]

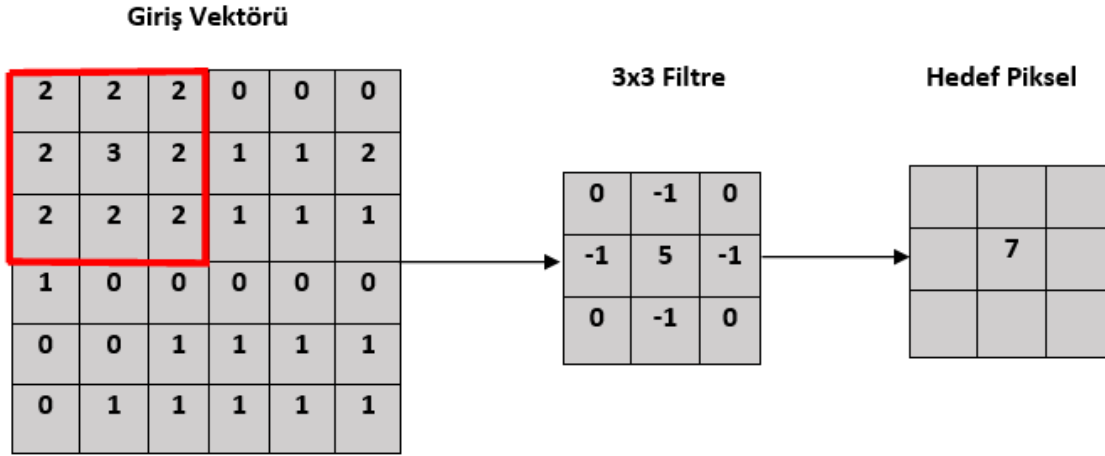
Evrişimli sinir ağı çeşitli sıralı katmanlardan oluşmaktadır. Şekil 3.1.'de örnek katman yapısı gösterilmiştir. Evrişim katmanları, havuz katmanları ve tam bağlantılı katmanlar evrişimli ağlarda kullanılan ortak katmanlardır. Evrişimli sinir ağları insan görme sisteminden esinlenmiştir. Farklı katman türleri görsel kortekste bulunan çeşitli hücrelere benzetilebilir [62]. Bu bölümde CNN'ler tarafından kullanılan katmanlar, fonksiyonlar ve ağ yapılarından detaylı olarak bahsedilecektir.

3.1.1. Evrişimsel Katmanlar

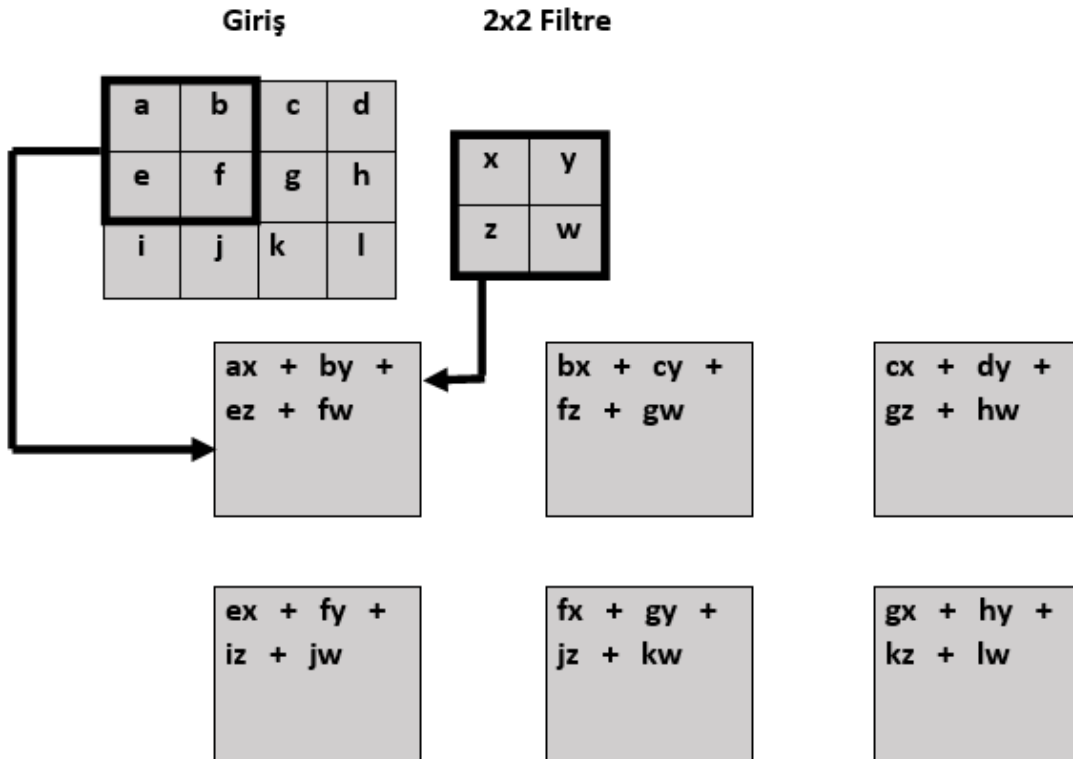
Evrişim katmanları, evrişimli sinir ağlarında mimariye de adını veren en önemli katman türüdür. Evrişimli sinir ağları, katmanlarından en az birinde evrişimsel katman olarak adlandırılan evrişim işlemi kullanılarak veri işlenmesi için tasarlanmıştır. İki boyutlu görüntüler için evrişim işlemi denklem 1’de paylaşılmıştır. Denklem 1’de yer alan parametrelerde $s(i, j)$: piksel satırı i ve piksel sütunu j ’de bir evrişim işlemi, K : evrişimsel filtre, I : giriş görüntüsü, m : filterinin satır sayısı ve n : filtrenin sütun sayısını ifade etmektedir.

$$s(i, j) = (I * K)(i, j) = \sum_{m=-k}^k \sum_{n=-k}^k I(m, n)K(i - m, j - n) \quad (1)$$

Renkli görüntülerin kırmızı, yeşil ve mavi olmak üzere üç kanalı bulunmaktadır. Her bir kanal iki boyutlu bir matris ile tanımlanmaktadır. Evrişimli katmanın girişi üç boyutlu bir matristir. Çıktısı iki boyutlu özellik haritaları ve bu katman için filtre sayısı n ise $n \times 3$ boyutlu matristir. Her filtre bir özellik haritası üretmektedir. Filtreler giriş görüntüsünün boyutundan daha küçük boyuta sahiptir, giriş görüntüsü boyunca sol üstten başlayarak sağ alta doğru kayar [63]. Şekil 3.2.’de filtre giriş vektörünün sol üst kısmında çizdirilmiştir. 3×3 filtre kullanılarak filtre sol üst köşedeki giriş vektörüne yerleştirildiğinde evrişim katmanının nasıl çalıştığına dair örnek Şekil 3.2.’de verilmiştir. Yalnızca bir giriş verisi kanalına sahip tek bir 3×3 filtre için 2D evrişim işlemi örneği Şekil 3.3.’te verilmiştir. Filtre pencere boyunca hareket ettirilerek azaltılmış uzamsal genişleme çıktısı üretilir.



Şekil 3.2. Filtre sol üst köşedeki giriş vektörüne yerleştirildiğinde evrişim katmanının nasıl çalıştığına dair örnek [2].



Şekil 3.3. 3×3 Çekirdek (Kernel) için 2D-Evrişim-İşlemi örneği

Modelin karmaşıklığını azaltmak amacıyla çıktıyı optimize etmek için evrişim katmanları üç hiperparametre kullanır.

Derinlik (depth): Derinliğin kontrol edilebilmesi avantajlı olmasına rağmen azaltılmasıyla birlikte nöron sayısı en aza indirgenir ve modelin örüntü tanımadaki etkinliğinin azaltılmasına sebebiyet verebilir.

Adım (step): Adım filtrenin girdi üzerinde ne kadar kaydırılarak hareket ettirildiğinde ilgili olan parametredir. Ne kadar yüksek olursa çıktı için daha az örtüşme miktarı ve daha az uzamsal boyutluluk elde edilmesini sağlar. Ne kadar düşük olursa filtre daha az kaydırılır ve daha yüksek uzamsal boyutluluk elde edilir.

Sıfır Doldurma (Zero padding): Giriş sınırını doldurmanın ve çıktı hacminin (uzaysal boyutluluğun) kontrolünün temel yöntemidir [2].

Evrişim katmanının çıktısının uzamsal boyutu denklem 2 ile hesaplanmaktadır.

$$UzamsalBoyut = \frac{(V - R) + 2z}{S + 1} \quad (2)$$

Denklem 2’de V değeri giriş hacim boyutunu, R değeri alıcı alan boyutunu, z değeri sıfır dolgu miktarını ve S değeri adım miktarını temsil etmektedir.

3.1.2. Havuz Katmanları

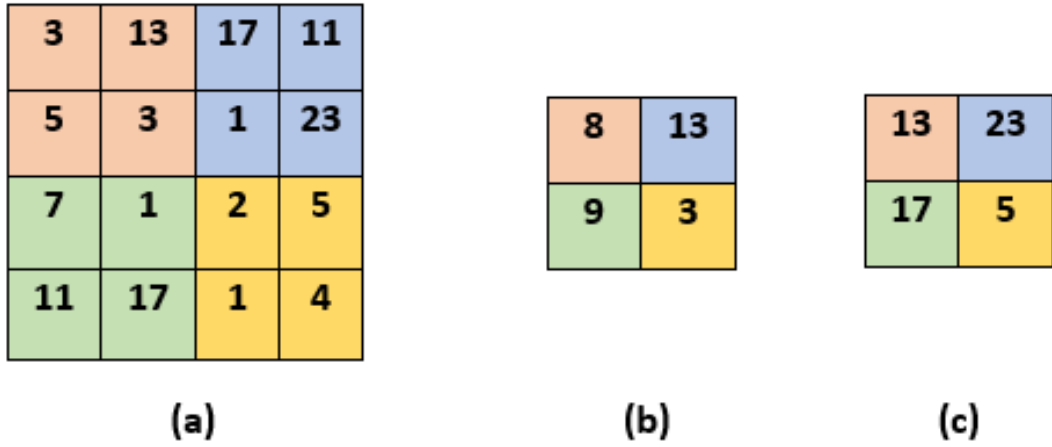
Havuzlama katmanı[1] özellik haritalarının boyutsallığını azaltmada kritik rol oynamaktadır. Bu katman sayesinde parametre miktarı ve hesaplama süresi azaltılır ve sonraki katman özellik haritalarının daha küçük bir versiyonunu girdi olarak alır. Maksimum havuzlama ve ortalama havuzlama en çok kullanılan iki yöntemdir. Maksimum havuzlama, filtrenin alıcı alanında bulunan maksimum değeri hesaplar ve ortalama havuzlama, alıcı alanındaki değerlerin ortalamasını hesaplar. Ortalama havuzlama denklem 3 ile ifade edilebilir:

$$s_i = \frac{1}{n} \sum_{j \in R_j}^n h_j \quad (3)$$

burada h , alt bölgedeki R_j özellik eşlemesinden bir pikseldir ve n , alt örneklemenin gerekli olduğu alt bölgedeki özellik sayısıdır. Maksimum havuzlama denklem 4 ile ifade edilebilir:

$$s_i = \max_{i \in R_j} h_i \quad (4)$$

Şekil 3.4.'te yer alan (a) görüntü piksel değerlerini, (b) her bir renkli bölgedeki ortalama havuzlama sonucu, (c) her bir renkli bölgedeki maksimum havuzlama sonucunu ifade etmektedir.

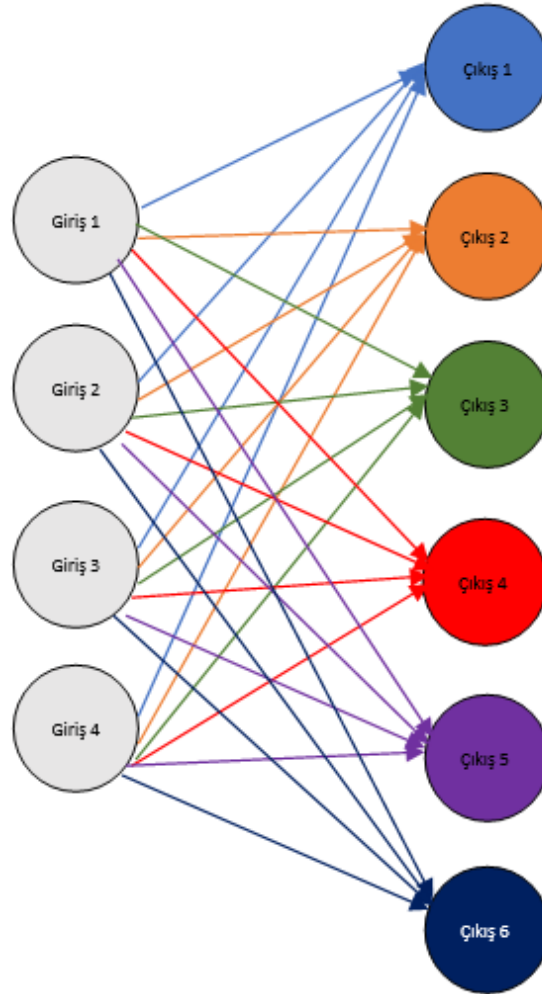


Şekil 3.4. Maksimum ve Ortalama Havuzlama Örneği

3.1.3. Tamamen Bağlı Katmanlar

Tamamen bağlı bir katmanın görevi evrişimli katman ve havuzlama katmanından girdi kabul etmek ve nihai sınıflandırma kararını veya her bir sınıf etiketi için nihai olasılıkları tahmin etmektir. Tam bağlantılı katmandaki bağlantılar eğitilebilir ağırlıklara sahiptir ve çıkış düğümünün çıkış değeri, girişlerinin ve yanlılık (bias) değerinin ağırlıkları toplamıdır. Evrişimli katman ve havuzlama katmanı bir görüntüden öznetelikleri çıkarır ve sonrasında tamamen bağlı katmanlar, çıkarılan bu özneteliklere dayalı olarak sınıflandırma gerçekleştirir. Bu nedenle bir CNN'in son katmanlarında tamamen bağlı katmanlar bulunmaktadır. Evrişimli katmanlardan elde edilen çıktı yüksek seviyedeki özellikleri temsil eder, anlamlı düşük boyutludur. Eklenen tamamen bağlı bir katman, bu özelliklerin doğrusal olmayan kombinasyonlarının

da öğrenilmesini sağlar. Tam bağlantılı katmandaki bağlantıların gösterimi Şekil 3.5.'de sunulmuştur. Şekil 3.5.'de görüldüğü gibi katmandan katmana tüm olası bağlantılar mevcuttur, giriş vektörünün her girişi çıkış vektörünün her çıkışını etkilemektedir. Ancak tüm ağırlıklar tüm çıkışları etkilemez. Şekill 3.5.'deki çizgilere bakıldığında mavi çizgilerin katmanın ilk nöronunu temsil ettiği görülür. Bu nöronun ağırlıkları sadece Çıkış 1'i etkilemektedir. Çıkış 2, 3, 4, 5 ve 6'yı etkilemez.



Şekil 3.5. Tamamen bağlantılı bir katmandaki bağlantıların gösterimi.

3.1.4. Aktivasyon Fonksiyonu

Aktivasyon fonksiyonları ağı doğrusal olmayan problemleri çözme imkanı veren, sinir ağı'nın ana parçasıdır. Aktivasyon fonksiyonunun olmadığı veya doğrusal aktivasyon fonksiyonunun olduğu durumlarda sinir ağı sadece doğrusal problemleri çözebilir. Doğrusal olmayan aktivasyon fonksiyonu ile birlikte sinir ağları yeterli katman ve nörona sahip olduğu sürece herhangi bir doğrusal olmayan işlevi öğrenebilir. Aktivasyon fonksiyonunun türevlenebilir olması gerekir. Türevlenebilir olmadığı durumlarda, geri yayılım algoritması gradyanı hesaplayamadığı için çalışmayacaktır. Aktivasyon fonksiyonunun seçimi, sinir ağının kapasitesi ve performansı üzerinde büyük bir etkiye sahiptir ve farklı aktivasyon fonksiyonları kullanılabilir. Bu bölümde farklı aktivasyon fonksiyonlarından bahsedilecek, avantajları ve dezavantajları incelenecektir.

Sigmoid Fonksiyonu

Sigmoid fonksiyonu herhangi bir değeri girdi olarak alır ve çıktı olarak 0 ile 1 arasındaki değerleri verir. Girdi ne kadar büyükse 1'e daha yakın olur, ne kadar küçükse 0'a o kadar yakın olur. Matematiksel olarak sigmoid fonksiyonunun orijinali denklem 5 ve türevi denklem 6 ile ifade edilmektedir:

$$f(x) = \frac{1}{1 + e^{-x}} \quad (5)$$

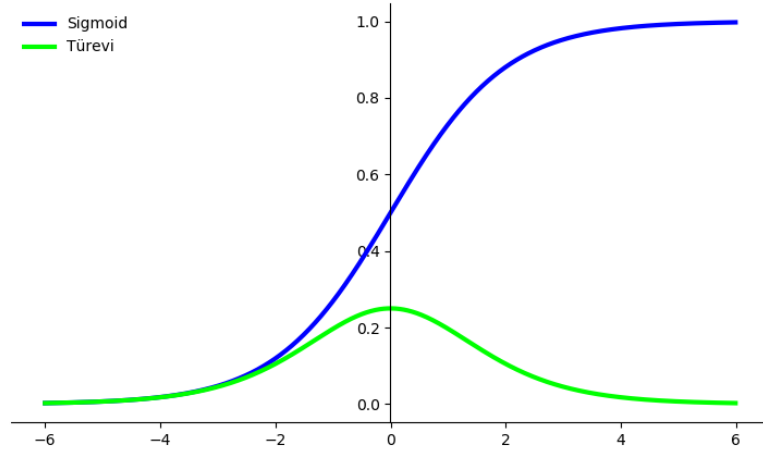
$$f'(x) = f(x)(1 - f(x)) \quad (6)$$

$$x \in (-\infty, \infty) \quad (7)$$

$$f(x) \in (0, 1) \quad (8)$$

Sigmoid fonksiyonu, denklem 7'de verilen aralıkta bir x değerini girdi olarak alır ve aldığı bu girdi değerini denklem 8'de verilen birim aralığındaki değere dönüştürür. Sigmoid fonksiyonu çıktı olarak olasılık tahmini yapacak modeller için yaygın olarak kullanılır. Herhangi bir şeyin olasılığı 0 ile 1 değeri arasında olduğundan sigmoid fonksiyonu avantaja sahip bir fonksiyondur. Sigmoid fonksiyonu türevlenebilir bir fonksiyondur, düzgün bir gradyan sağlamak ve sıçramaları önlemektedir. Şekil 3.6.'da sigmoid fonksiyonu ve türevinin grafiği verilmiştir. Türev grafiğinde gradyan değerleri sadece -3 ve 3 arasında önemlidir bu aralık

dışındaki değerler çok küçük gradyanlara sahiptir. Gradyan değeri 0'a yaklaştıkça ağ öğrenmeyi durdurmakta ve kaybolan gradyan (vanishing gradient) problemine sebep olmaktadır. Sigmoid fonksiyonunun bir diğer dezavantajı fonksiyon çıktısının orijin noktasına göre simetrik olmamasıdır. Nöron çıktıları aynı işarete sahiptir ve bu durum ağın eğitimini daha kararsız ve zor hale getirmektedir.



Şekil 3.6. Sigmoid fonksiyonu (sol) ve türevi (sağ) [3]

Tanh (Hiperbolik Tanjant) Fonksiyonu

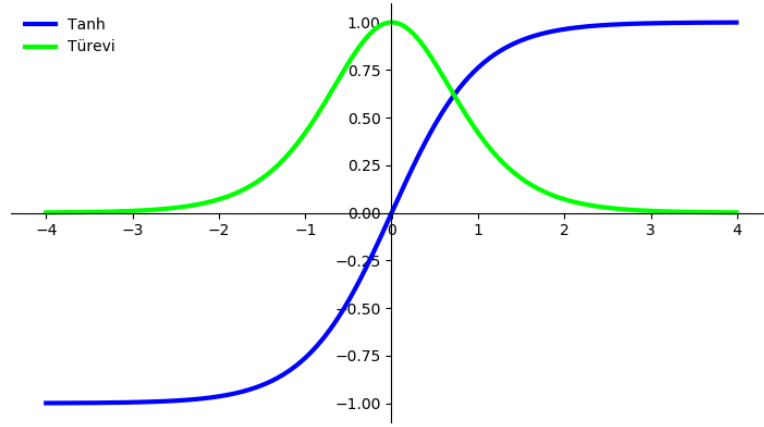
Tanh fonksiyonu sigmoid fonksiyonuna benzer şekilde S şekline sahiptir. Herhangi bir girdi değeri aldığı anda -1 ve 1 çıkış aralığında değer vermektedir. Çıkış değeri girdi büyüdükçe 1'e küçüldükçe -1'e yaklaşmaktadır. Matematiksel olarak tanh fonksiyonunun orijinali denklem 9 ve türevi denklem 10 ile ifade edilmektedir:

$$f(x) = \frac{2}{1 + e^{-2x}} - 1 \quad (9)$$

$$f'(x) = 1 - f(x)^2 \quad (10)$$

Tanh fonksiyonun orijinali ve türevi Şekil 3.7.'de verilmiştir. Tanh fonksiyonu orijin merkezlidir. Çıktı değerleri negatif, pozitif veya nötr şekilde eşlenebilir. Bu durum bir sonraki katman için verileri merkezlemesi sebebiyle öğrenmeyi daha kolay hale getirmektedir. Tanh fonksiyonunun gradyan grafiği sigmoid fonksiyonunun gradyan grafiğine benzemekte ve

aynı şekilde kaybolan gradyan problemine sebep olmaktadır. Tanh fonksiyonunun sigmoid fonksiyonuna göre en önemli avantajı orijin merkezli olmasıdır. Gradyan grafikleri benzer şekilde olsa da tanh fonksiyonu sigmoid'e göre daha tercih edilebilir konumdadır.



Şekil 3.7. Tanh fonksiyonu (sol) ve türevi (sağ) [3]

Doğrultulmuş Lineer Birim (Rectified Linear Unit - Relu) Fonksiyonu

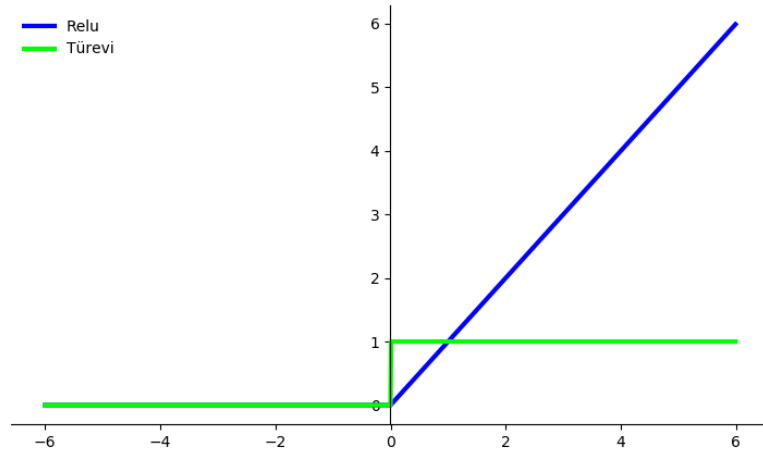
Relu, rektifiye edilmiş, doğrultulmuş doğrusal birim anlamına gelmektedir. İsminden dolayı doğrusal bir işlev izlenimi vermesine rağmen Relu türev işlevine sahiptir. Nöronlar aynı anda etkinleştirilmemektedir. Sadece doğrusal dönüşümün çıktısı 0'dan küçükse devre dışı bırakılmaktadır. Matematiksel olarak Relu fonksiyonunun orijinali denklem 11 ve türevi denklem 12 ile ifade edilmektedir:

$$f_{ReLU}(x) = \max(0, x) \quad (11)$$

$$f'_{ReLU}(x) = \begin{cases} 0, & x < 0 \\ 1, & x \geq 0 \end{cases} \quad (12)$$

Relu fonksiyonun orijinali ve türevi Şekil 3.8.'de verilmiştir. Fonksiyonun avantajlarından biri belirli sayıda nöronu aktive edip diğer nöronları pasifleştirdiğinden hesaplama açısından çok daha verimlidir. Relu'nun diğer bir avantajı, satüre olmama özelliği sayesinde, kayıp

fonksiyonunun global minimuma doğru gradyan iniş yakınsamasını hızlandırmasıdır. Fonksiyonun dezavantajlarından biri grafiğin negatif kısmında gradyanının sıfır olmasıdır. Bu nedenle geri yayılım sürecinde bazı nöronların ağırlıkları ve yanlılıkları (bias) güncellenememektedir. Bu durum aktive olmayan ölü nöronlara neden olmaktadır. Yaşanılan bu soruna “ölen Relu (dying Relu)” sorunu denilmektedir.



Şekil 3.8. Relu fonksiyonu (sol) ve türevi (sağ) [3]

Sızıntı Relu (Leaky Relu) Fonksiyonu

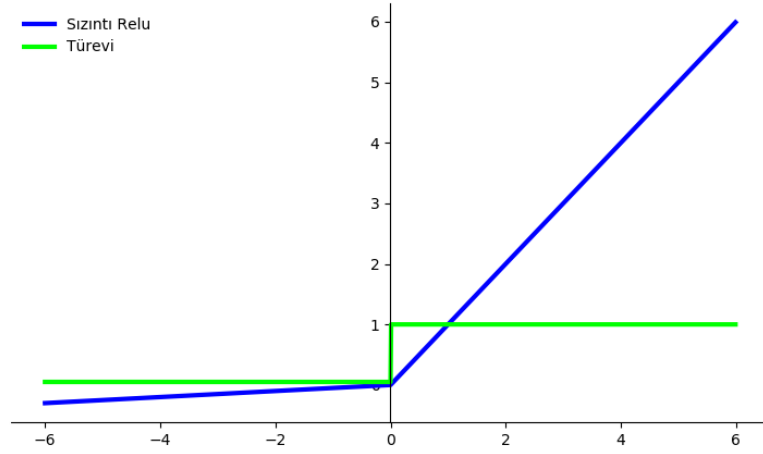
Sızıntı Relu, Relu’nun sahip olduğu “ölen Relu” problemini çözmek amacıyla Relu’nun geliştirilmiş bir versiyonudur. Matematiksel olarak sızıntı Relu fonksiyonunun orijinali denklem 13 ve türevi denklem 14 ile ifade edilmektedir:

$$f_{LReLU}(x) = \begin{cases} \alpha x, & x < 0 \\ x, & x \geq 0 \end{cases} \quad (13)$$

$$f'_{LReLU}(x) = \begin{cases} \alpha, & x < 0 \\ 1, & x \geq 0 \end{cases} \quad (14)$$

Sızıntı Relu fonksiyonunun orijinali ve türevi Şekil 3.9.’da verilmiştir. Sızıntı Relu negatif giriş değerleri için geri yayılımı etkinleştirmektedir. Relu’dan farklı olarak yapılan bu değişimle negatif değerlerin gradyanları sıfır olmayan bir değer olarak elde edilmektedir. Dezavantajı

tahminlerin negatif giriş değerleri için tutarlı olmama ihtimaline sahip olmasıdır. Negatif değerleri için elde edilen gradyan, model parametrelerinin öğrenilmesini zaman alıcı hale getiren küçük bir değerdir.



Şekil 3.9. Sızıntı Relu fonksiyonu (sol) ve türevi (sağ) [3]

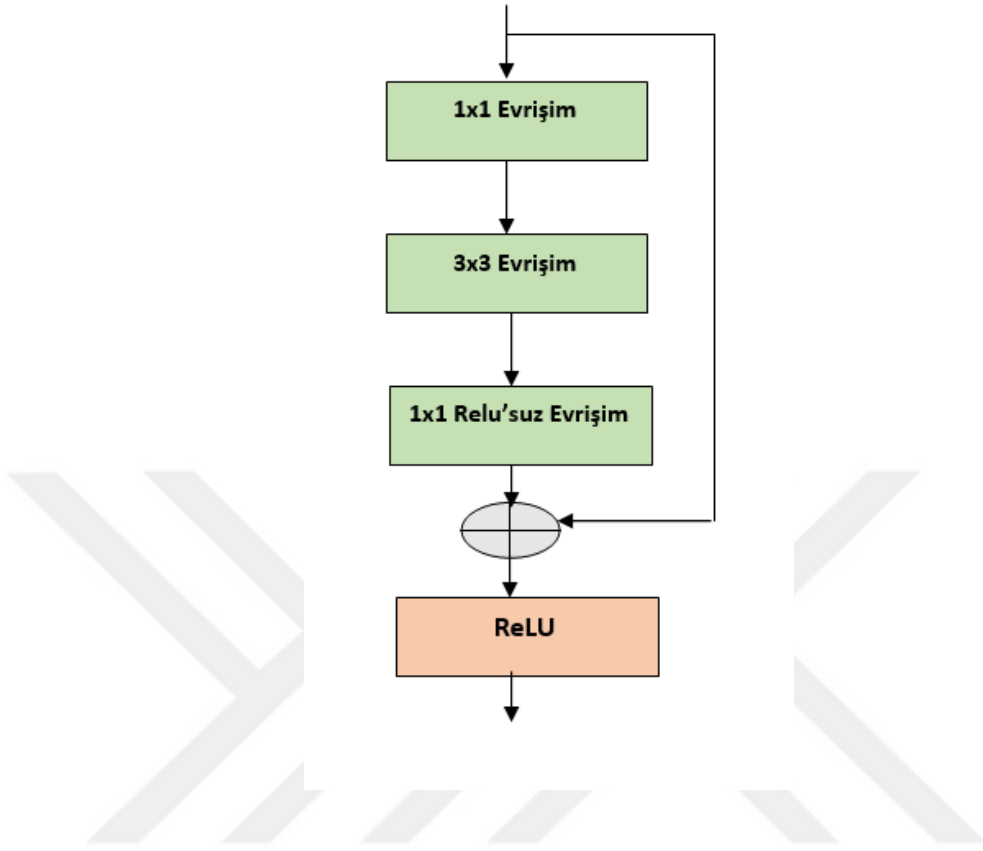
3.1.5. CNN Mimarileri

Evrişimli sinir ağı inşa etmek, oluşturmak oldukça zor bir iştir [64]. Bireysel veri kümeleri gözlemlenen özellikler itibariyle farklılık gösterdiğinden ilerlemek için altın bir kural yoktur. Bu sebeple belirli bir ağ yapısına ihtiyaç vardır. Evrişimli sinir ağı inşa etmek için mevcut çözümlerden gelen belirli kurallar vardır, bu çözümler zaman içinde ispatlanmıştır. Tipik bir evrişimli sinir ağı yapısı evrişimli katman, havuzlama katmanı ile başlar ardından tam bağlantılı katman ile devam eder. Son olarak çıktı katmanı ile tamamlanır. Evrişim katmanlarında çoğunlukla 3x3, 5x5, 7x7 boyutlarında filtreler kullanılır. Aktivasyon fonksiyonları derin ağlarda daha iyi performans gösterdiği bilinen Relu fonksiyonunun modifikasyonlarıdır. Evrişimli sinir ağlarının en yaygın örneklerinden bazıları ResNet-50 [65], Inceptionv3 [4], VGG-16 [66], DenseNet [67], InceptionResNetv2 [68] detaylı olarak incelenecektir.

ResNet-50

ResNet ağı artık ağ anlamına gelmektedir. He ve diğerleri tarafından 2015 yılında 'Görüntü Tanıma için Derin Kalıntı Öğrenme' [65] başlıklı araştırma makalesinde tanıtılan yenilikçi

bir ağıdır. ImageNet test kümesinde %3.57 hata oranıyla ILVRSC yarışmasının sonucunda ilk sırada yer almıştır. ResNet'in farklı katman sayısına sahip, aynı konsept üzerinde çalışan birçok çeşidi bulunmaktadır. ResNet-50, elli sinir ağı katmanına sahip bir ağıdır. ResNet ağının tasarımıyla birlikte derin sinir ağlarında daha iyi bir performans elde etmek mümkün hale gelmiştir. Derin evrişimli sinir ağlarında katman sayısı arttıkça farklı katmanlar çeşitli görevler için eğitilebildiğinden bu görev farklılaşması karmaşık sorunların daha verimli bir şekilde çözülmesine yardımcı olur. Sinir ağının katman sayısı arttıkça doğruluk seviyesi önce doygun hale gelir ve sonrasında düşebilir. Derin ağlar ile eğitilen bir modelin performansı eğitim ve test verilerinde bozulabilmektedir. Yaşanılan bu problem aşırı uyumdan (overfitting) kaynaklı bir sonuç değildir. Bu problem kaybolan veya patlayan gradyandan (vanishing or exploding gradients) kaynaklanmaktadır. ResNet ağının çözüm bulduğu kısım bu probleme ışık tutmaktadır. ResNet ağı modelin doğruluğunu arttırmak için artık bloklardan faydalanmaktadır ve artık blokların merkezinde kısayol bağlantıları bulunmaktadır. Kullanılan bu yöntemler sayesinde ResNet ağı ile birlikte parametre sayısı ve hesaplama karmaşıklığının artmaması sağlanmış ve kaybolan gradyan sorunu azaltılmıştır. Şekil 3.10.'da örnek artık blok gösterilmiştir. Kısayol bağlantısı bir veya daha fazla katmanı atlamaktadır ve ardından ağa Şekil 3.10.'da gösterildiği gibi eklenmektedir. Kısayol bağlantısı ile birlikte, daha iyi performans sağlayan daha derin modeller oluşturmak mümkün hale getirilmiştir.



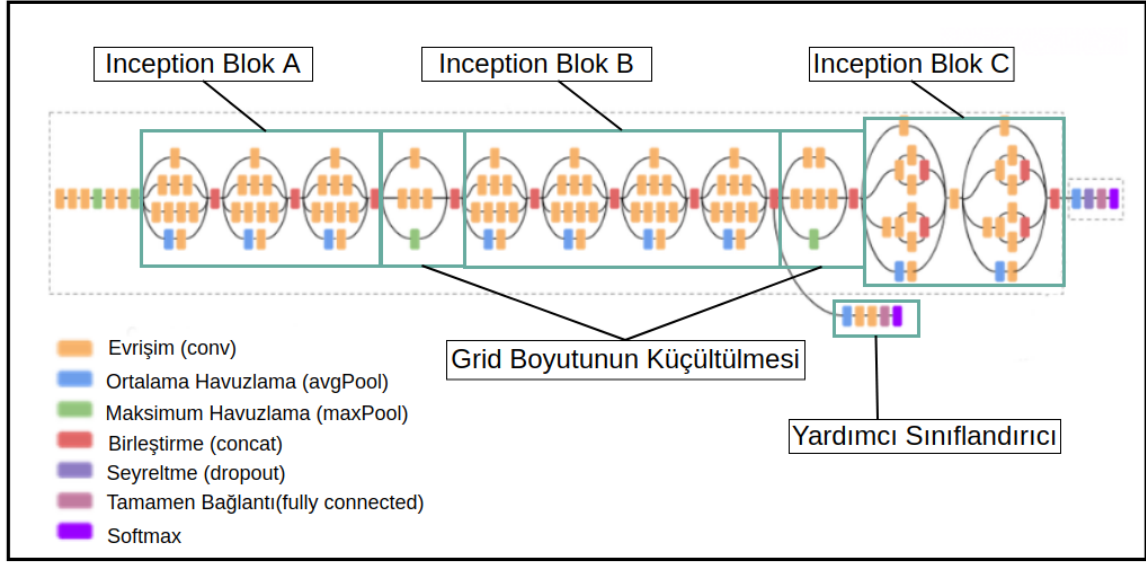
řekil 3.10. ResNet50 Artık (Residual) Blok Örneęi

Inceptionv3

Inception ağı CNN'in gelişmesinde önemli bir rol oynamıştır. Öncesinde yapılan çalışmalarda evrişimli sinir ağları daha iyi performans elde edebilmek için katmanlarını derinleştirmiştir. Inception ağı hız ve başarımı aynı anda sağlamak amacıyla farklı yöntemler uygulamıştır. Inceptionv3, temel olarak önceki ağ mimarilerini değiştirerek daha az hesaplama gücü kullanmaya odaklanmıştır. 2015 yılında 'Bilgisayarlı Görü için Inception Mimarisini Yeniden Düşünmek' başlıklı makalede önerilmiştir [4]. Inceptionv3 ağının mimarisinin aşamalarından detaylı olarak aşağıda bahsedilmiştir:

- Faktörleştirilmiş Evrişimler: Ağda yer alan parametrelerin sayısını azaltmaktadır. Bu nedenle ağın verimliliğine katkı sağlamaktadır.
- Daha Küçük Evrişimler: Kullanılan filtre boyutunun değiştirilmesi daha hızlı eğitim elde edilmesine olanak sağlar.
- Asimetrik Evrişimler: Asimetrik evrişimler, görüntünün x ve y eksenlerini ayrı ayrı olarak çalışır. $(1 \times n)$ filtresinden önce bir $(n \times 1)$ filtre ile bir evrişim gerçekleştirilmesi örnek olarak verilebilir. 5×5 evrişimi, 1×5 evrişim ve ardından 5×1 evrişim ile değiştirilebilir.
- Yardımcı Sınıflandırıcı: Eğitim sırasında katmanlar arasına yerleştirilen küçük bir evrişimdir. Oluşan kayıp ana ağ kaybına eklenir.
- Grid Boyutunun Küçültülmesi: Özellik haritalarının grid boyutunu azaltmak için geleneksel olarak maksimum havuzlama veya ortalama havuzlama kullanılır. Inceptionv3 modelinde, grid boyutunun azaltılması amacıyla ağ filtrelerinin aktivasyon boyutu genişletilir. Örneğin, k filtreli $n \times n$ grid, $2k$ filtreli $n/2 \times n/2$ gridle sonuçlanır. İki paralel evrişim bloğu ve havuzlama birleştirilerek yapılır.

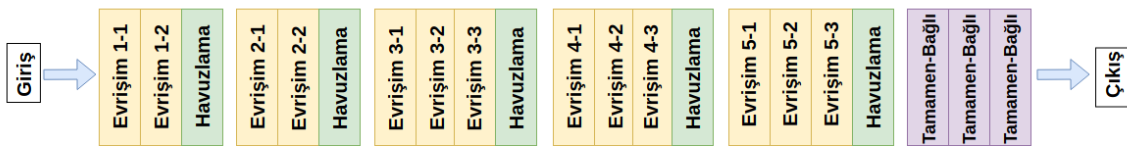
Şekil 3.11.'de Inceptionv3 ağ mimarisi gösterilmiştir. Inceptionv3 ağını oluşturan birimler renklerle belirtilmiş ve her bir bloğun içerdiği birimler görselleştirilmiştir.



Şekil 3.11. Inceptionv3 Ağ Mimarisi [4]

VGG-16

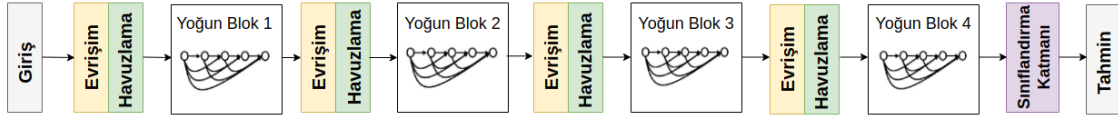
VGG-16, Simonyan ve Zisserman [66] tarafından geliştirilmiştir. 16 katmandan oluşan temel bir sıralı evrişimli ağ modelidir. 2014 yılında ImageNet yarışmasında (challenge) uygulanmıştır. ImageNet [69] 1000 sınıfa ait 14 milyondan fazla görüntüye sahip bir veri kümesidir. VGG-16 ağı, ImageNet üzerinde %92,7 ilk 5 test doğruluğu elde etmiştir. VGG-16 ağı, büyük çekirdek boyutlu filtreleri birinci ve ikinci katmanında birden fazla 3x3 çekirdek boyutunda filtreyle değiştirerek AlexNet ağı üzerinde iyileştirme yapmıştır. Şekil 3.12.'de VGG-16 ağının mimarisi verilmiştir. VGG-16 mimarisi; evrişim (convolution), havuzlama (pooling), tamamen-bağlı (fully-connected) katmanlarından oluşmaktadır.



Şekil 3.12. VGG-16 Ağ Mimarisi

DenseNet

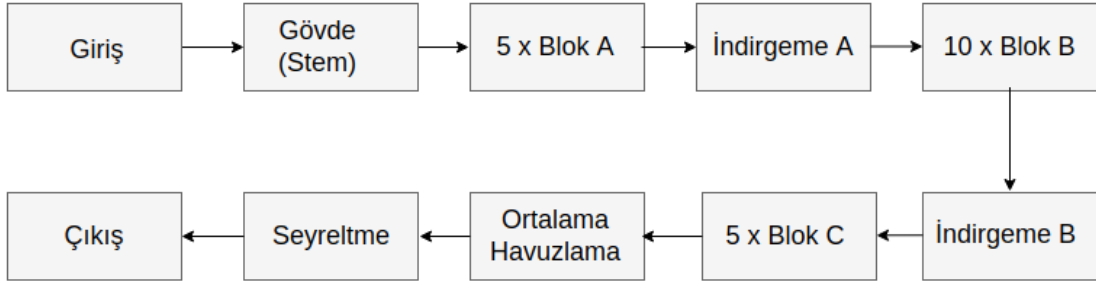
CNN'lerin derinleşmesiyle ortaya çıkan problemi giderebilmek amacıyla DenseNet ağı [67] önerilmiştir. Problemin ana nedeni giriş katmanından çıkış katmanına kadar olan bilgi yolunun oldukça büyük olması ve bilgilerin diğer tarafa ulaşmadan önce kaybolabilmesidir. DenseNet ağı, diğer mimarilerde tanıtılan katmanlar arasındaki bağlantı modelini basite indirmiştir. DenseNet ağında maksimum bilgi ve gradyan akışı sağlanarak sorun giderilmiştir. Bu akışı sağlamak amacıyla her katman doğrudan birbirine bağlanmıştır. DenseNet yazarları, oluşturdukları mimariyi derinleştirmek yerine özelliklerin yeniden kullanımı yoluyla ağın potansiyelinden yararlanmışlardır. Şekil 3.13.'te DenseNet ağ mimarisi verilmiştir. DenseNet mimarisinin verildiği Şekil 3.13.'te verilen yoğun bloklar (dense blocks), n adet yoğun katmandan oluşmaktadır. Blok içinde yer alan her bir yoğun katman, önceki katmanlardan özellik haritalarını almaktadır ve çıktılarını tüm katmanlara iletmektedir. Yoğun katman (dense layer), 1×1 ve 3×3 olmak üzere iki farklı evrişim işleminden oluşmaktadır. 1×1 evrişim işleminde özellik çıkarımı yapılmaktadır ve çıkarılan özelliklerin 3×3 evrişim işlemi ile derinlik/kanal sayısı azaltılmaktadır.



Şekil 3.13. DenseNet Ağ Mimarisi

InceptionResNetv2

Inception ve artık (residual) ağ mimarileri düşük hesaplama maliyetine ve yüksek performansla sahip mimarilerdir. InceptionResNetv2 ağı [68], bu iki mimarinin birleştirilmesinden oluşmaktadır. Inception ve ResNet bloğunda, çok boyutlu evrişim filtreleri artık bağlantılarıyla birleştirilmiştir ve bu bağlantıların kullanılması derin yapıların neden olduğu bozulma problemini ortadan kaldırarak eğitim süresini azaltmıştır. Şekil 3.14.'de inceptionResNetv2 ağ mimarisi özetlenmiştir. InceptionResNetv2 ağ mimarisinde verilen A, B, C blokları [68] çalışmasında şema olarak verilen inceptionResNet A, B ve C bloklarını ifade etmektedir. Mimaride gövde (stem) katmanı girdi olarak $299 \times 299 \times 3$ boyutundaki görüntüyü almaktadır ve $35 \times 35 \times 384$ boyutlu bir çıktı üretmektedir.



Şekil 3.14. InceptionResNetv2 Ağ Mimarisi

Tablo 3.1.'de ağların parametre sayısı, ilk-1 (top-1) doğruluk oranı ve ağların yayımlanma yılı paylaşılmıştır. VGG-16 ağı 2014 yılında ortaya çıkmış ve o yıla göre derin ağ olarak kabul edilmiştir. VGG-16 modelinin avantajlarından biri kıyaslama yapmak için iyi bir mimari olmasıdır, dezavantajı ise 138 milyon parametreye sahip olması ve bu nedenle eğitiminin yavaş olmasıdır. VGG-16 ağının geleneksel ağ mimarisinin aksine, ResNet-50 mimarisi artık (residual) modüller ve kısayol bağlantıları kullanarak derinleşen ağlarda yaşanan problemlere ışık tutmuştur. ResNet-50 ağı, VGG-16 ağına göre daha derin bir model olmasına rağmen Tablo 3.1.'de paylaşıldığı gibi 26 milyon parametre sayısına sahiptir. Bunun temel sebebi, ağın son katmanında tamamen bağlı (fully-connected) katman yerine küresel ortalama havuzlama (global average pooling) katmanı kullanmasıdır. Inceptionv3 ağı, hız ve başarımı ayna anda sağlayan yaklaşımlar geliştirmiştir. 27 milyon parametre sayısına sahiptir ve ImageNet verisi üzerinde %78.8 ilk-1 (Top-1) doğruluk değerine sahiptir. ResNet-50'ye göre başarımlar artışı sağlamıştır. InceptionResNetv2, artık ağ ve inception ağ mimarilerini birleştirerek doğruluk değeri daha yüksek olan fakat aynı zamanda hesaplama yükü daha yüksek olan bir mimari geliştirmiştir. ResNet-50 ve Inceptionv3 modellerinin sahip olduğu parametre sayısının yaklaşık iki katı parametreye sahiptir. InceptionResNetv2, diğer ağlara kıyasla, ImageNet veri kümesi ile yapılan testlerde %80.1 ilk doğruluk oranıyla en yüksek doğruluğa sahip ağ olmuştur. DenseNet-121 ağı, ResNet mimarisinin katmanlar arasında yoğun (dense) bağlantılar içeren bir evrimi olarak kabul edilmektedir. ImageNet veri kümesi üzerinde %74.98 ilk-1 doğruluk oranına ve 8 milyon parametre sayısı ile diğer modellere kıyasla oldukça düşük bir parametre sayısına sahiptir.

Tablo 3.1. Ağların Karşılaştırma Tablosu

Model İsmi	Parametre Sayısı	İlk-1 (Top-1) Doğruluk	Yıl
VGG-16	138 milyon	%74.4	2014
ResNet-50	26 milyon	%77.15	2015
Inceptionv3	27 milyon	%78.8	2015
DenseNet-121	8 milyon	%74.98	2016
InceptionResNetv2	55.8 milyon	%80.1	2016

3.2. Sinir Ağları Eğitimi

Evrışimli sinir ağları eğitimi ile ilgili aşamalar bu bölümde özetlenmiştir. Makine öğreniminde, denetimli öğrenme, denetimsiz öğrenme ve pekiştirmeli öğrenme olarak üç temel öğrenme şekli vardır [70]. Sınıflandırma problemi için en yaygın kullanılan yöntem her girdi değerine karşılık gelen istenen çıktı ile etiketlendiği bir eğitim veri kümesi ile öğrenmenin gerçekleştiği denetimli öğrenmedir. Ağın tahmininin mümkün olduğunca bu değerlere yakın olması beklenir.

Bir evrışimli sinir ağı, bir kayıp fonksiyonunu yinelemeli olarak minimum seviyeye indirerek eğitir. Sinir ağının çıkışı ile beklenen çıkış arasındaki fark kayıp fonksiyonu kullanılarak belirlenir. Fark kayıp fonksiyonu kullanılarak hesaplandıktan sonra farkı en aza indirecek şekilde ağ parametreleri ayarlanır. Farkı minimum seviyeye indirmek için kullanılan yöntemlerden biri gradyan iniş (gradient descent) yöntemidir. Gradyan iniş yöntemi hesaplama açısından pahalı olabilir. Geri yayılım yöntemi [71], ağdaki tüm ağırlıklar ve sapmalar için gradyanın kısmi türevlerini verimli bir şekilde hesaplar. Kaybolan ve patlayan (vanishing-/exploding) gradyanlar gradyan iniş ile ilgili sinir ağlarının eğitimi sırasında karşılaşılan bir problemdir [61].

3.2.1. Kayıp fonksiyonu

Evrışimli sinir ağı, bir kayıp fonksiyonunu en aza indirgeyerek eğitilir. Ağ üzerinden eğitim örneği geçirilir ve çıktı, beklenen çıktı ile karşılaştırılır. Ağ çıkışında bulunan değer ile beklenen değer arasındaki farkı tanımlamak için kayıp fonksiyonu kullanılır. Kullanılan kayıp

fonksiyonu eğitimin başarımını önemli ölçüde etkilemektedir. Kayıp fonksiyonları sınıflandırma ve regresyon kayıp fonksiyon olmak üzere iki gruba ayrılır [72]. Sınıflandırma kayıp fonksiyonunda sonlu kategoriler kümesinden bir değer tahmin etmeye çalışılırken, regresyonun amacı, bir dizi parametreye dayalı sürekli bir değeri tahmin etmektir. Bu tezin kapsamı nedeniyle sınıflandırma probleminde kullanılacak olan sınıflandırma kayıp fonksiyonları detaylı olarak verilmiştir. Kayıp fonksiyonu ve maliyet fonksiyonu benzer gözükmelerine rağmen aralarında belirgin bir fark bulunmaktadır. Kayıp fonksiyonu tek bir eğitim örneği için hesaplanırken, maliyet fonksiyonu tüm eğitim veri kümesi üzerinde hesaplanan ortalama kayıptır [73].

Çapraz Entropi Kaybı

Çapraz entropi kaybı (cross entropy loss), görüntü sınıflandırması, algılaması ve segmentasyonu için yaygın olarak kullanılan bir kayıp fonksiyonudur. Denklem 16'da çapraz entropi kaybının formülü verilmiştir. Denklem 15'de p parametresi $y=1$ etiketli sınıf için modelin tahmini olasılığını temsil etmektedir. Denklem 16'da çapraz entropi kaybının formülü verilmiştir.

$$p_t = \begin{cases} p, & y = 1 \\ 1 - p, & \text{diğer} \end{cases} \quad (15)$$

$$CE(p_t) = -\log(p_t) \quad (16)$$

Odak Kaybı

Odak kaybı (focal loss), sınıf dengesizliğine sahip veri kümelerinde iyi performans gösteren, 2017 yılında [74] makalesinde yayınlanan kayıp fonksiyonudur. Odak kaybı, çapraz entropi kayıp fonksiyonunun gelişmiş versiyonudur. Odak kaybı fonksiyonu, kolay bir şekilde sınıflandırılan örneklerin ağırlığını azaltarak ağırlık zor örneklerin eğitime odaklanmasını sağlamaktadır. Böylece, kolay örnekler eğitimi baskılayamamakta ve zor sınıflandırılan örneklerle odaklanabilmektedir. Odak kaybı, nadir sınıflara yüksek ağırlık verirken baskın sınıflara daha küçük ağırlıklar vermektedir. Pozitif örnekler, hedef sınıfı veya temel gerçekler gibi ön plan bilgilerini ifade etmekteyken, negatif örnekler, hedef olmayan sınıf veya arka plan bilgilerini ifade etmektedir. α ve γ katsayıları odak kaybının hiperparametreleridir. α parametresi,

pozitif/negatif örneklerin dengesini sağlamaya yardımcı olurken, γ parametresi kolay/zor örneklerin dengesini sağlamaya yardımcı olmaktadır [74].

Denklemler 17’de p_t parametresi küçük olduğunda, modülasyon faktörü 1’e yaklaşmaktadır ve kayıp etkilenmemektedir. p_t parametresi 1’e yaklaştığında, modülasyon faktörü 0’a gider ve iyi sınıflandırılmış örnekler için kayıp azalmaktadır.

$$FL(p_t) = -\alpha_t(1 - p_t)^\gamma \log(p_t) \quad (17)$$



3.2.2. Gradyan İniş ve Geri Yayılım Yöntemi

Gradyan iniş, fonksiyonun yerel minimumunu tahmin etmek için kullanılan sayısal bir yöntemdir. Bu yöntem kayıp fonksiyonunun yerel minimumunu tahmin etmek için kullanılabilir. Rastgele seçilen bir başlangıç noktasının konumu yinelemeli olarak güncellenir. Yinelemeyi durdurmanın koşulu, maksimum yineleme sayısına erişmiş olmak veya iki adım arasındaki konum değişikliğinin belirlenen bir eşikten az olup olmadığını kontrol etmektir. Öğrenme oranı (learning rate) adı verilen değişken η tarafından gradyan algoritması kontrol edilebilir. Bu değerin küçük olduğu durumlarda adım uzunluğu azalacak ve adım sayısı artacağından gradyan algoritması uzun sürecektir, bu değer büyük olduğunda adım uzunluğu artacak ve adım sayısı azalacağından gradyan algoritması kısa sürecektir. Bir pozisyonun güncellenmesi şu şekilde ifade edilebilir:

$$v' = v - \eta \nabla C \quad (18)$$

Bu denklemde belirtilen v ifadesi değişkenlerin bir vektörüdür ve ∇C , maliyet fonksiyonunun v 'ye göre türevidir. Gradyan iniş yöntemi büyük veri kümeleri için uygun değildir, zaman alıcıdır. Bunun nedeni maliyet fonksiyonunun her bir örnekle hesaplanmasının gerekliliğidir. Gradyan inişin bu probleminin çözülmesi amacıyla Stokastik gradyan iniş (SGD) adı verilen düzeltilmiş bir yöntem önerilmiştir [73]. Önerilen bu yöntemle veri kümesi, rastgele seçilen örneklerle birkaç gruba bölünür. Eğitim bölünen gruplarla grup grup işlenerek ve her bir grup için maliyet fonksiyonu hesaplanarak ve gradyanlar güncellenerek gerçekleştirilir. Eğitim verisi bitene kadar bu süreç devam eder. Geliştirilen stokastik gradyan iniş yöntemine rağmen gradyan yöntemi hesaplama karmaşıklığına sahip ve bu nedenden dolayı da oldukça yavaş bir algoritmadır [73].

Geri yayılım algoritması, sinir ağlarını eğitmek için yaygın olarak kullanılan bir algoritmadır. Öğrenme hızındaki artış sayesinde evrişimli sinir ağlarının yeni yollar kullanmasına imkan tanımıştır. Karmaşık fonksiyonların kısmi türevlerini hesaplamak için zincir kuralı yöntemini kullanan geri yayılım algoritması, birden fazla aşamadan oluşmaktadır. Bu algoritmanın ilk aşaması, ileri geçiş olarak adlandırılır ve bu aşamada giriş görüntüsü ağın giriş katmanına uygulanır. Çıkış katmanına kadar denklem 19 her katman için hesaplanır. Denklem 19'da verilen parametrelerin açıklaması şu şekildedir; $f(x)$ model çıktısı, x model girdisi, w ağırlık parametresi, b ön yargı (bias) değeri, σ aktivasyon fonksiyonudur.

$$f(x) = \sigma(w \cdot x^T + b) \quad (19)$$

Maliyet fonksiyonunun değeri δ hesaplanır. δ , son katmandan ilk katmana doğru hata denklem 20 formülüyle geri yayılır. Bu aşama geriye doğru geçiş olarak adlandırılmaktadır. Denklem 20 formülünde yer alan l , katman numarasıdır.

$$\delta^{x,l} = ((w^{l+1})^T \delta^{x,l+1}) o(\sigma' z^{x,l}) \quad (20)$$

$(w^{l+1})^T$ ifadesi $(l+1)$. katmandaki ağırlık vektörünün transpozunu ifade etmektedir, δ yayılan hata vektörünü, σ aktivasyon fonksiyonunu, z nöronların ağırlıklı girdisini ifade etmektedir. İşlemler arasındaki o sembolü iki vektörün eleman bazında çarpımıdır. Her bir katman için ağırlık ve Bayes formülü denklem 21’de verilmiştir.

$$\begin{aligned} w^l &= w^l - \frac{n}{m} \sum_x \delta^{x,l} (a^{x,l-1})^T \\ b^l &= b^l - \frac{n}{m} \sum_x \delta^{x,l} \end{aligned} \quad (21)$$

η öğrenme oranı (learning rate), m küme boyutu (batch size), $a^{x,l}$ ifadesi aktivasyon fonksiyonudur [73].

3.2.3. Kaybolan ve Patlayan Gradyan

Kaybolan gradyan problemi, ağın öğrenilemez hale geldiği ağ eğitimi problemidir [75]. Ağ eğitiminde geri yayılım algoritması çıkış katmanından giriş katmanına doğru ilerledikçe gradyanlar genellikle küçülür ve sıfıra yaklaşır. Bu sebeple de ilk katmanların ağırlıklarını güncellemeden bırakması sonucu ortaya çıkar. Model ağırlıkları, kaybolan gradyan probleminde eğitim sırasında 0 değeri alabilmektedir. Kaybolan gradyan probleminde gradyan hiçbir zaman optimum seviyeye ulaşamaz. Gradyan probleminin birçok sebebi olabilir. Relu aktivasyon fonksiyonunun kullanımının diğer geleneksel aktivasyon fonksiyonlarının kullanımıyla karşılaştırıldığında kaybolan gradyan problemine sebep olma olasılığı daha düşüktür [76].

Patlayan gradyan probleminde kaybolan gradyan probleminin aksine bazı durumlarda geri yayılım algoritması ilerledikçe gradyanlar daha da büyür. Bunun sonucu olarak çok büyük ağırlık güncellemeleri olur ve gradyan inişi farklılaşır. Model ağırlıkları, patlayan gradyan probleminde eğitim sırasında NaN değeri alabilmektedir. Patlayan gradyan sorununa önerilen çözüm, ağ eğitimini büyük ağırlıklar atamak için cezalandıran düzenleştirme (regularization) işlevidir. Bu bölümde bahsedilen gradyan problemleri çok sayıda katmana sahip, çok derin ağ mimarilerinde karşımıza çıkmaktadır [77].

3.3. Aşırı Uyumun Önlenmesi ve Sinir Ağları Optimizasyonu

Aşırı uyum, modelin eğitim verilerine fazla uyması durumunda ortaya çıkan bir problemdir. Model gördüğü verileri çok iyi tahmin ederken, görmediği verilere karşı düzgün bir şekilde çalışmaz. Bu durum yeterli veriye sahip olunmadığında, karmaşık bir model kullanıldığında veya model çok uzun süre eğitildiğinde meydana gelir. Model veri kümesi içindeki gürültüyü veya alakalı olmayan bilgileri öğrenmeye ve ezberlemeye başlar, böylece model aşırı uyum sağlamış olur ve yeni verilere genelleme yapamaz. Bir modelin görmediği veride genelleme yapabilmesi istenen bir durumdur. Bu durumu sağlamak için birçok yöntem kullanılır. Bu bölümde aşırı uyumu önleme yöntemleri olan seyreltme (dropout) [78], veri arttırma (data augmentation) [79], aktarım öğrenmesi (transfer learning) [80] yöntemleri detaylı olarak incelenecektir.

3.3.1. Seyreltme

Aşırı uyumu engelleyen etkili yöntemlerden biri seyreltme yöntemidir [78]. Bu yöntem tam bağlantılı katmanlarda uygulanır. Eğitim sırasında bazı nöronlar, rastgele olarak belirli bir olasılıkla her örnek için devredışı bırakılır. Bırakılan düğümler ileri veya geri yayılımda göz ardı edilmektedir. Seyreltme için yaygın olarak 0.5 değeri kullanılmaktadır. Bu yöntem ağ yakınsamasını önemli ölçüde yavaşlatır, ancak aşırı uyumu önlemede etkilidir.

3.3.2. Veri Arttırma

Aşırı uyumu önleyecek bir başka yöntem de veri arttırma yöntemidir. Orijinal örneklere dönüşüm uygulayarak eğitim veri kümesini genişletmeyi sağlamaktadır. Eğitim veri kümesinin

yeterli miktarda olmasını sağlamak, girdi ve çıktı değişkenleri arasındaki baskın ilişkinin aydınlatılmasına fırsat sağlayarak modelin doğruluğunu arttırabilir. Arttırım uygulanacak veri kümesinin temiz ve ilgili verilerden oluşması önemlidir. Aksi takdirde temiz olmayan verilere uygulanacak arttırım ile birlikte daha fazla karmaşıklık elde ederek aşırı öğrenmenin desteklenmesi söz konusu olabilir. Uygulanacak dönüşümlere kaydırma, kırpma, döndürme, gürültü ekleme, renk tonu ve doygunluğun ayarlanması vb. yöntemler örnek verilebilir.

3.3.3. Aktarım Öğrenmesi

Aktarım öğrenmesi, daha önceden eğitilmiş bir modelin yeni problem üzerinde yeniden kullanılması olarak ifade edilebilir. Önceden eğitilen model genellikle ImageNet gibi büyük bir veri kümesi üzerinde eğitilir. Önceden eğitilen modelin alt katmanları (mevcut verilerin temel yapısını öğrenmesi beklenen) yeniden kullanılabilir ve daha uygun olan yüksek seviyeli özellikleri öğrenmek için üst katman değiştirilebilir [81]. Bu yöntem kullanılarak halihazırda eğitilmiş bir modelin bilgisi aktarım öğrenimi boyunca farklı bir probleme aktarılır. Aktarım öğrenmesi yöntemiyle evrişimli sinir ağının evrişim katmanları, görüntü verilerinden özellik çıkarımı için kullanılarak sınıflandırma yöntemleri için kullanılabilir [80]. Modeller için gereken eğitim süresini azaltmak amacıyla aktarım öğrenmesi faydalı olabilir [82]. Bir evrişimli sinir ağı modelinin önceden eğitilmiş evrişim filtreleri ile eğitilmesi, ağın evrişim filtrelerini öğrenmesi gerekmediğinden, ağ yakınsamasını önemli ölçüde hızlandırabilir. Bir ağın ince ayarının yapılması, önceden eğitilmiş evrişim filtreleri ile bir ağın eğitilmesi işlemine denir. Bir ağda ince ayar yapmak, evrişimli sinir ağlarına aktarım öğreniminin nasıl uygulandığı olarak açıklanmaktadır.

4. DENEYLER ve SONUÇLAR

4.1. Veri Önışleme

Tez kapsamında veri kümesinin iki farklı versiyonu kullanılmıştır. Kullanılan ilk versiyonda 121 sınıf içeren veri kümesinin 'bilinmeyen' isimli üç sınıfı, bir uzmanın makul bir kesinlikle sınıflandıramadığı örnekler içerdiğinden veri kümesinden çıkarılmıştır. Kullanılan ilk versiyon, Set-1 olarak isimlendirilmiştir. 118 sınıfta toplamda 29419 plankton görüntüsü bulunmaktadır. Veri kümesi %75 oranında eğitim kümesi, %15 doğrulama kümesi ve %15 test kümesi olarak bölünmüştür. Kullanılan ikinci versiyon, Set-2 olarak isimlendirilmiştir. Set-2, Kaggle plankton veri kümesinin [18] yazarları tarafından seçilen bir alt kümesidir. En az 100 örnek bulunduran plankton sınıfları seçilmiştir. Toplamda 38 sınıf için 14374 görüntü bulunmaktadır. İlk versiyonda olduğu gibi veri kümesi %75 oranında eğitim kümesi, %15 doğrulama kümesi ve %15 test kümesi olarak bölünmüştür. Kaggle plankton veri kümesi ile ilgili detaylar ve Set-1, Set-2 veri kümeleri ile ilgili açıklamaların yer aldığı tablolar Bölüm 2.3.3.'te paylaşılmıştır.

Kaggle veri kümesi sınıf dengesizliği olan bir veri kümesidir. Eğitim örnekleri bazı sınıflar için sınırlıdır ve 118 farklı olası taksonomik sınıf etiketi bulunmaktadır. Bu sebeplerden dolayı aşırı uyum (overfitting) bu veri kümesi için önemli bir problemdir. Bu problemlere ışık tutmak amacıyla denenen yöntemler bu bölümde detaylı anlatılmıştır.

Sınırlı veri problemine sunulan çözümlerden biri veri arttırma (data augmentation) yöntemidir. Tez kapsamında iki farklı yöntemle veri arttırma uygulanmıştır. Kısmi veri arttırma yöntemi ve tam veri arttırma yöntemi.

Kısmi veri arttırma olarak isimlendirilen yöntemde, 118 sınıfın her biri için sınıfın 60 olarak belirlenen sınırın üstünde veya altında olduğu kontrol edilmiştir. Sınıfın içerdiği görüntü sayısı 60'dan düşükse veri arttırma uygulanmıştır. Değilse o sınıf için veri arttırma uygulanmamıştır. Veri arttırma uygulanan ve veri sayısı 60'dan az olan her bir sınıfın veri sayısı kısmi veri arttırma yöntemiyle yaklaşık 6 katına çıkmıştır. Kısmi veri arttırım sonucu elde edilen veri kümesi %70 eğitim, %15 doğrulama, %15 test kümesi olarak bölünmüştür. Kısmi veri arttırımıyla ilgili detaylar Tablo 4.1.'de paylaşılmıştır.

Tam veri arttırma yönteminde, sınıfların ne kadar veriye sahip olduğundan bağımsız her sınıf için veri arttırma uygulanmıştır. Her bir sınıfın veri sayısı tam veri arttırma yöntemi

uygulandığında yaklaşık 6 katına çıkmıştır. Tam veri arttırım sonucu elde edilen veri kümesi %70 eğitim, %15 doğrulama, %15 test kümesi olarak bölünmüştür. Tam veri arttırımıyla ilgili detaylar Tablo 4.2.'de paylaşılmıştır.

Veri Arttırma Yöntemleri

Görüntüler yatay ve dikey olarak tersine çevirme işlemine tabi tutularak yatay ve dikey veri arttırma yöntemi uygulanmıştır. Yatay ve dikey olarak çevirme işleminin uygulandığı plankton görüntüleri Şekil 4.1.'de verilmiştir.

Görüntülere rotasyon veri arttırımı uygulanmıştır. Her bir görüntü 90, 180, 270 derece döndürülerek farklı açılardan da görüntünün elde edilmesi sağlanmıştır. Rotasyon işleminin uygulandığı plankton görüntüleri Şekil 4.2.'de verilmiştir.

Odaklanma 0.75, 1.25 ve 1.5 katsayılarında uygulanarak verilerin farklı ölçeklerde de elde edilmesi sağlanmıştır. Odaklanma işleminin uygulandığı plankton görüntüleri Şekil 4.3.'te verilmiştir. Odaklanma yöntemi her görüntüye uygulanmamıştır. Bunun sebebi bazı planktonların odaklanma sebebiyle kendisinden farklı bir türe benzemesinin önlenmesinin istenmesidir. Sadece odaklanma sonrası en ve yüksekliği orijinal görüntüyle aynı olan planktonlar için odaklanma veri arttırımı kullanılmıştır.

Tablo 4.1. Set-1 Veri Kümesine Kısmi Veri Arttırım Uygulanmasının Sonuçları

Set-1 Veri Kümesi ve Kısmi Veri Arttırma Yöntemi				
#Toplam Veri	#Sınıf	#Eğitim Seti	#Doğrulama Seti	#Test Seti
36456	118	27512	4472	4472

Tablo 4.2. Set-1 Veri Kümesine Tam Veri Arttırım Uygulanmasının Sonuçları

Set-1 Veri Kümesi ve Tam Veri Arttırma Yöntemi				
#Toplam Veri	#Sınıf	#Eğitim Seti	#Doğrulama Seti	#Test Seti
133103	118	124159	4472	4472

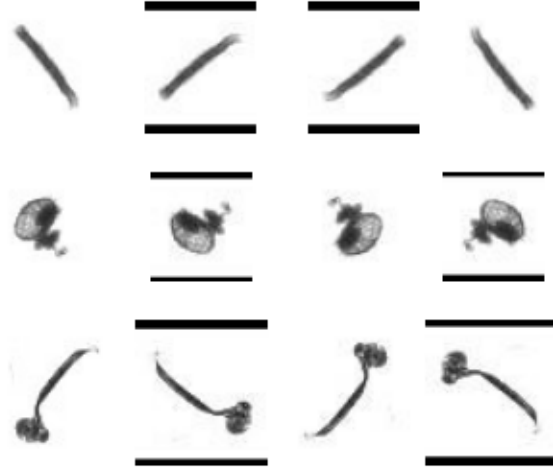
Geleneksel bir evrişimli sinir ağı, giriş görüntülerinin önceden tanımlanmış bir boyuta sahip olmasını gerektirmektedir. Tez kapsamında kullanılan modellerde evrişimli sinir ağının giriş boyutu 224x224 boyutundadır. Plankton görüntülerinin çok farklı boyutlara sahip olduğu düşünüldüğünde tüm görüntülerin aynı boyuta boyutlandırılması bilgi kaybına neden olmaktadır. Bu kaybın yaşanmaması için iki farklı yöntem uygulanmıştır. Boşluk ekleme yöntemi ve kare kırpma işlemi.

Boşluk ekleme işleminde plankton görüntüsü evrişimli sinir ağının istediği boyutlara getirilmesi için etrafına beyaz pikseller yerleştirilerek elde edilir. Plankton görüntüsünün 224x224 boyutundan daha büyük olduğu durumlarda en-boy oranı (aspect ratio) göz önünde bulundurularak yeniden boyutlandırılır.

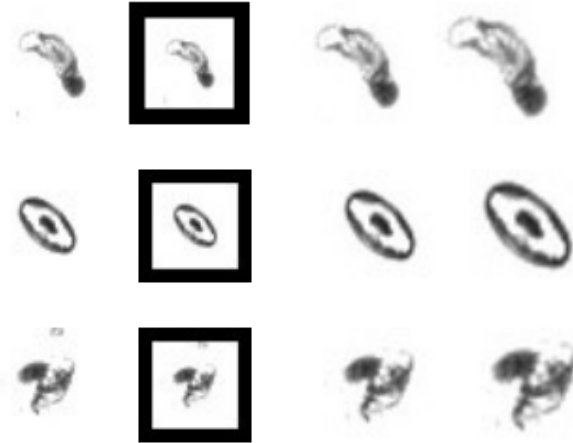
Kare kırpma işleminde en-boy oranı (aspect ratio) gözetilerek 224x224 boyutunda yeniden boyutlandırılan görüntü merkeze alınarak biküçük interpolasyonu kullanılarak yeniden boyutlandırılır. Boyutlandırma işlemi sırasında var olan plankton görüntüsü beklenen değerden büyükse en-boy oranı gözetilerek yeniden boyutlandırılır. Boşluk ekleme ve kırpma işlemlerinin uygulandığı plankton görüntüleri Şekil 4.4.'te verilmiştir.



Şekil 4.1. Kaggle Plankton Veri Kümesine Uygulanan Çevirme Veri Arttırma Yöntemi. Sırayla Orijinal, Yatay Çevirme, Dikey Çevirme



Şekil 4.2. Kaggle Plankton Veri Kümesine Uygulanan Rotasyon Veri Arttırma Yöntemi. Sırayla Orijinal, 90 Derece Rotasyon, 180 Derece Rotasyon, 270 Derece Rotasyon



Şekil 4.3. Kaggle Plankton Veri Kümesine Uygulanan Yakınlaştırma Veri Arttırma Yöntemi. Sırayla Orijinal, 0.75, 1.25, 1.75 Yakınlaştırma



Şekil 4.4. Boşluk Ekleme Yöntemi (Sol Sütun) ve Kare Kırpma Yöntemi (Sağ Sütun)

4.2. Değerlendirme Metrikleri

Plankton sınıflandırma sistemlerinin başarımını değerlendirmek için çeşitli metrikler kullanılmaktadır. Kesinlik, geri çağırma, F1-Skor, ilk-1 (top-1) ve ilk-5 (top-5) metrikleri bu metriklerden bazılarıdır. Metrikler 0 ile 1 arasında değerler üretmektedir. Üretilen değer 1'e yakınlığı modelin ne kadar iyi olduğunu, 0'a yakınlığı modelin ne kadar kötü olduğunu göstermektedir.

Bahsedilen yaklaşımların çok sınıflı sınıflandırmaya göre farklı ortalama alma yaklaşımları bulunmaktadır. Kullanılan veri kümesinin özellikleri değerlendirilerek, ortalama alma yaklaşımlarının hangisinin plankton sınıflandırma algoritmalarını değerlendirmek için daha uygun olacağına karar verilmelidir. Bu bölümde ağırlıklı F1-Skor metriği ve hesaplanan diğer metrikler detaylı bir şekilde açıklanacaktır. Modelin doğruluğunu kontrol etmek için kullanılan dört temel yaklaşım Şekil 4.5.'de gösterilmiştir.

		Tahmin Edilen	
		Pozitif	Negatif
Gerçekleşen	Pozitif	Doğru Pozitif (DP)	Yanlış Pozitif (YP)
	Negatif	Yanlış Negatif (YN)	Doğru Negatif (DN)

Şekil 4.5. Gerçekleşen ve Tahmin Edilen Hata Matrisi

- Doğru Negatif (DN): Gerçekte negatif olan ve model tarafından negatif olarak sınıflandırılanlardır.
- Doğru Pozitif (DP): Gerçekte pozitif olan ve model tarafından pozitif olarak sınıflandırılanlardır.
- Yanlış Negatif (YN): Gerçekte pozitif olan ve model tarafından negatif olarak sınıflandırılanlardır.
- Yanlış Pozitif (YP): Gerçekte negatif olan ve model tarafından pozitif olarak sınıflandırılanlardır.

Kesinlik (Precision): Sınıflandırıcı tarafından gerçekten kaç tane pozitif örneğin pozitif olarak tahmin edildiğinin ölçüsüdür. Pozitif tahminelerin doğruluk oranıdır.

$$K = \frac{DP}{DP + YP} \quad (22)$$

Geri Çağırma (Recall): Sınıflandırıcı tarafından elde edilen tüm pozitif tahminlerden doğru pozitif tahminlerin sayısını ölçen bir ölçümdür. Doğru tanımlanmış pozitiflerin oranıdır.

$$G = \frac{DP}{DP + YN} \quad (23)$$

F1-Skor: Geri çağırma ve kesinlikten elde edilen sonuçların harmonik ortalaması hesaplanarak elde edilir.

$$F1 = 2 * \frac{K * G}{K + G} \quad (24)$$

Makro Ortalama

Makro ortalama, sayısız ortalama alma işlemleri arasında en temel olanıdır. Makro ortalama alınacak değerlendirme metriğinin aritmetik ortalaması (ağırlıksız ortalama) alınarak hesaplanır. Makro ortalama ile tüm sınıflar nihai elde edilen sonuç metriğine eşit katkıda bulunmaktadır. Tez kapsamında kesinlik, geri çağırma ve F1-Skor metriklerinin makro ortalaması hesaplanmıştır.

Ağırlıklı Ortalama

Ağırlıklı ortalama [83] yöntemi ile her sınıftaki örnek sayısı önem kazanmaktadır. Bu nedenle sınıflar, o sınıflardaki örnek sayısına göre ağırlıklandırılır ve her bir metriğin ortalaması alınır. Tez kapsamında kesinlik, geri çağırma ve F1-Skor metriklerinin ağırlıklı ortalaması hesaplanmıştır.

İlk-1 (Top-1) Doğruluk

İlk-1 doğruluk; en yüksek olasılık değerine sahip olan tahminin, hedef etiket (target label) ile eşleştiği örneklerin oranını ölçmektedir.

İlk-5 (Top-5) Doğruluk

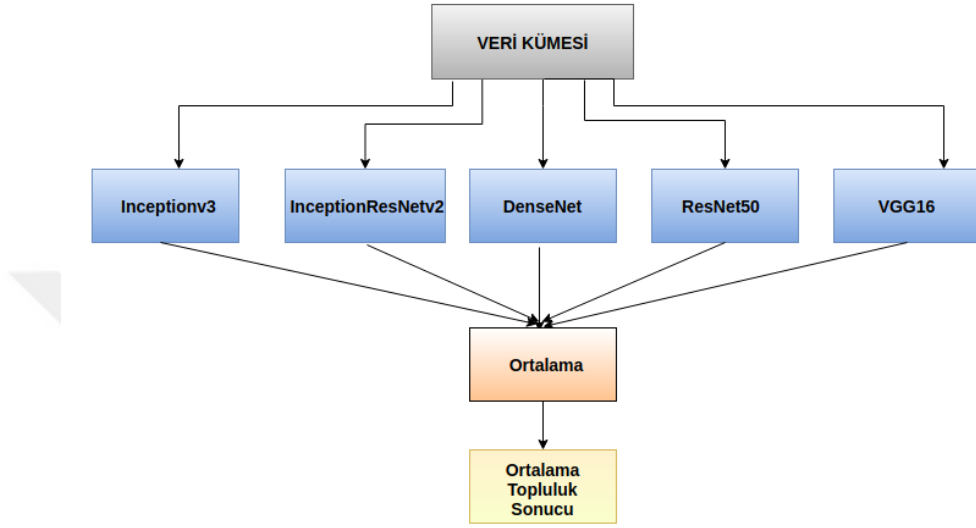
İlk-5 doğruluk; en yüksek 5 olasılık değerine sahip olan tahminin, hedef etiket (target label) ile eşleştiği örneklerin oranını ölçmektedir.

Tez çalışmasında gerçekleştirilen deneylerde kesinlik, geri çağırma, F1-Skor metriklerinin ağırlıklı ortalaması ve makro ortalaması hesaplanmıştır. Ayrıca her bir deneyde eğitilen modellerin ilk-1 (top-1) ve ilk-5 (top-5) doğruluk yüzdeleri hesaplanmıştır.

4.3. Ortalama Topluluk Yöntemi

Ortalama topluluk yöntemi, toplulukta bulunan her bir modelin nihai tahmine eşit miktarda katkıda bulunduğu bir topluluk öğrenme yaklaşımıdır. Toplulukta bulunan her bir modelin

ilgili sınıf için yaptığı tahminler birleştirilerek ortalaması alınır. Bu yöntem kullanılarak belirli tek bir modele göre daha yüksek sınıflandırma doğruluğu elde edilir. Şekil 4.6.'da tez kapsamında kullanılan ortalama topluluk yöntemi verilmiştir.



Şekil 4.6. Ortalama Topluluk Örneği

4.4. Deneyler

Tez kapsamında yapılan deneyler ve elde edilen sonuçlar bu bölümde detaylı olarak ele alınacaktır.

- Deney-1’de Bölüm 4.1.’de detaylı olarak bahsedilen ve oluşturulan 3 veri kümesi farklı eğitim yöntemleriyle Inceptionv3 ağı kullanılarak elde eğitilmiştir. Deney-1’in sonucunda en iyi sonucu veren veri kümesi diğer deneylerde kullanılmıştır.
- Deney-2’de Deney-1 sonucunda karar verilen veri kümesine Bölüm 4.1.’de bahsedilen boşluk ekleme ve kare kırpma işlemleri uygulanmış ve farklı eğitim yöntemleriyle eğitilerek sonuçlar elde edilmiştir. Deney-2 sonucunda en iyi sonucu veren yöntem diğer deneylerde kullanılmıştır.

- Deney-3’de, Set-1 (118 Sınıf) ve Set-2 (38 Sınıf) veri kümeleri, tam veri arttırma ve tüm katmanların eğitilmesi yöntemleri kullanılarak Inceptionv3, InceptionResNetv2, DenseNet, ResNet-50, VGG-16 ağlarında eğitilmiştir. Alınan sonuçlar paylaşılmıştır.
- Deney 4’de Set-1 (118 Sınıf) ve Set-2 (38 Sınıf) veri kümelerinin Deney-3 kapsamında ilgili yöntemlerle eğitilmesi sonucu elde edilen modeller ortalama topluluk yöntemiyle birleştirilmiştir.
- Deney-5’de, çapraz entropi kaybı ve odak kaybı fonksiyonlarının eğitimde kullanılmasıyla elde edilen sonuçlar karşılaştırılmış ve odak kaybı fonksiyonunun farklı hiperparametreleri uygulanarak eğitim gerçekleştirilmiştir.
- Deney-6’da, 118 sınıfa ait DenseNet ağı üzerinde Deney-3 kapsamında Set-1 veri kümesi ile gerçekleştirilen deney sonucu elde edilen modelin hata matrisi hesaplanmıştır. Elde edilen 118x118 hata matrisi gruplanarak 33x33 hata matrisine dönüştürülmüştür. 33 grup olarak eğitilen modelin hata matrisi sonucuyla elde edilen sonuç karşılaştırılmıştır.

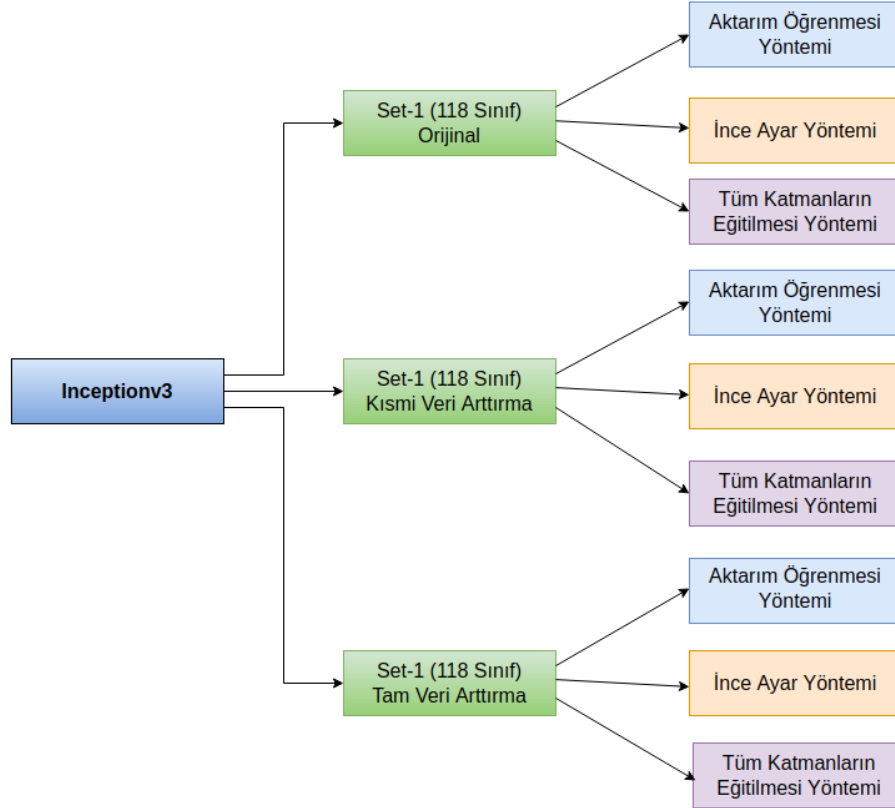
4.4.1. Deney Detayları

Deneyler kapsamında yapılan tüm eğitimler esnasında eğitim veri kümesi ve doğrulama veri kümesi kullanılmıştır. Eğitimde doğrulama doğruluğu her bir iterasyonda hesaplanarak 30 iterasyonda en yüksek doğrulama doğruluğuna sahip ağırlık kaydedilmiş ve testlerde kullanılmıştır. Eğitim toplamda 30 iterasyonda gerçekleştirilmiştir. Girdi görüntü büyüklüğü 224x224’tür. Öğrenme oranı (learning rate) 0.00001, küme boyutu (batch size) 32 ve seyreltme (dropout) 0.5 olarak kullanılmıştır. Kayıp fonksiyonu olarak kategori çapraz entropi ve optimizasyon olarak Adam optimizasyon yöntemi kullanılmıştır. Eğitimde her bir iterasyonda 859 ayri iterasyon kullanılmıştır (adım/iterasyon).

4.4.2. Deney-1

Deney 1’de, Bölüm 4.1.’de detaylı olarak bahsedilen 118 sınıftan oluşan Set-1 veri kümesi Orijinal, Kısmi Veri Arttırma ve Tam Veri Arttırma olmak üzere üç farklı versiyonuyla Inceptionv3 ağına eğitilmiştir. Her bir veri kümesi kullanılarak üç farklı yöntem ile Inceptionv3 ağına eğitim gerçekleştirilmiştir. Kullanılan üç farklı yöntem detaylı açıklanmıştır:

Aktarım öğrenmesi yöntemi ağıın sadece son katmanının değiştirilerek eğitilmesi, **ince ayar yöntemi** ağıın katman sayısının %30 oranında aktif edilerek eğitilmesi ve **tüm katmanların eğitilmesi yöntemi** ağıın tüm katmanlarının aktif edilerek eğitilmesi olarak ifade edilmiştir. Kademeli olarak katmanlar aktif edilerek katmanların eğitime dahil edilmesinin başarıma etkisi gözlemlenmiştir. Deney 1, Şekil 4.7.'de görselleştirilmiştir.



Şekil 4.7. Deney-1 Özet. Aktarım Öğrenmesi Yöntemi, İnce Ayar Yöntemi, Tüm Katmanların Eğitilmesi Yöntemi

Eğitim, keras kütüphanesi kullanılarak Inceptionv3 ağıında aktarım öğrenmesi, ince ayar ve tüm katmanların eğitilmesi olmak üzere üç farklı yöntem ve üç farklı veri kümesi ile 9 farklı kez gerçekleştirilmiştir.

4.4.3. Deney-1 Sonuç

Aktarım öğrenmesi, ince ayar ve tüm katmanların eğitilmesi yöntemleri ve her bir yöntem için orijinal, kısmi veri arttırma, tam veri arttırma veri kümeleriyle gerçekleştirilen dokuz farklı deneyin deney sonuçları Tablo 4.3.'te paylaşılmıştır. Tabloda her bir deney için kesinlik, geri çağırma, F1-Skoru metrikleri makro ortalama ve ağırlıklı ortalama yöntemleri kullanılarak ayrı ayrı hesaplanmıştır. Tablo 4.4.'de eğitim ve doğrulama doğruluk, kayıp değerleri paylaşılmıştır. Ayrıca eğitilen modelin, test veri kümesi kullanılarak test edilmesi sonucu elde edilen ilk-1 (Top-1) ve ilk-5 (Top-5) sonuçları her bir deney için paylaşılmıştır. Doğrulama doğruluğu, eğitim esnasında doğrulama veri kümesi üzerinde her bir iterasyonda hesaplanan doğruluk değeridir. Eğitim sırasında her bir iterasyonda doğrulama doğruluğu hesaplanır ve gerçekleştirilen 30 iterasyonda en yüksek doğruluk değerine sahip doğruluğun olduğu iterasyonun doğruluk ve kayıp değerleri ve eğitilen ağırlık kaydedilir. Modelin testi kaydedilen ağırlık ile gerçekleştirilmiştir..

Set-1 (118 Sınıf) veri kümesinde, Inceptionv3 ağı kullanılarak gerçekleştirilen Deney-1 kapsamında en iyi sonuç tam veri arttırma uygulanmış Set-1 veri kümesinin tüm katmanların eğitilmesi yöntemi kullanılması ile elde edilmiştir. Tabloda, her bir sütunda alınan en iyi sonuç koyu renk ile gösterilmiştir. Tablo 4.3. ve Tablo 4.4.'te yer alan sonuçlar değerlendirildiğinde yöntemler arasında en iyi sonucun, tüm katmanların eğitilmesi yöntemi ve tam veri arttırma ile elde edilmiş veri kümesinden elde edildiği görülmektedir. Kısmi veri arttırma ile elde edilmiş veri kümesinde kesinlik olarak yaklaşık %1 iyi sonuç elde edilmesine rağmen diğer değerlendirme metriklerinde tam veri arttırma ile elde edilmiş veri kümesi ile en iyi sonuçlar elde edilmiştir. Doğrulama doğruluğu, Tablo 4.4.'teki en düşük doğrulama değerine göre yaklaşık 0.06 oranında artmıştır. Ağırlıklı F1-Skoru Tablo 4.3.'teki en düşük ağırlıklı F1-Skoruna göre yaklaşık %7 artış sağlamıştır. İlk-1 (Top-1) ve İlk-5 (Top-5) doğruluk değerleri %77.4 ve %97.0 ile tüm katmanların eğitilmesi yöntemi ve tam veri arttırma ile elde edilen veri kümesinden elde edilmiştir.

Tablo 4.3. Set-1 Veri Kümesinde Elde Edilen Deney-1 Sonuçları

		Makro Ortalama			Ağırlıklı Ortalama		
		Kesinlik	Geri Çağırma	F1-Skoru	Kesinlik	Geri Çağırma	F1-Skoru
Aktarım Öğrenmesi Yöntemi	Orijinal	55%	50%	51%	70%	71%	70%
	Kısmi Veri Arttırma	61%	57%	57%	71%	71%	70%
	Tam Veri Arttırma	61%	51%	53%	71%	71%	70%
İnce Ayar Yöntemi	Orijinal	60%	55%	56%	72%	73%	72%
	Kısmi Veri Arttırma	59%	61%	59%	73%	73%	72%
	Tam Veri Arttırma	58%	56%	56%	74%	75%	74%
Tüm Katmanların Eğitilmesi Yöntemi	Orijinal	62%	57%	57%	74%	74%	73%
	Kısmi Veri Arttırma	64%	62%	62%	75%	75%	75%
	Tam Veri Arttırma	63%	62%	62%	77%	77%	77%

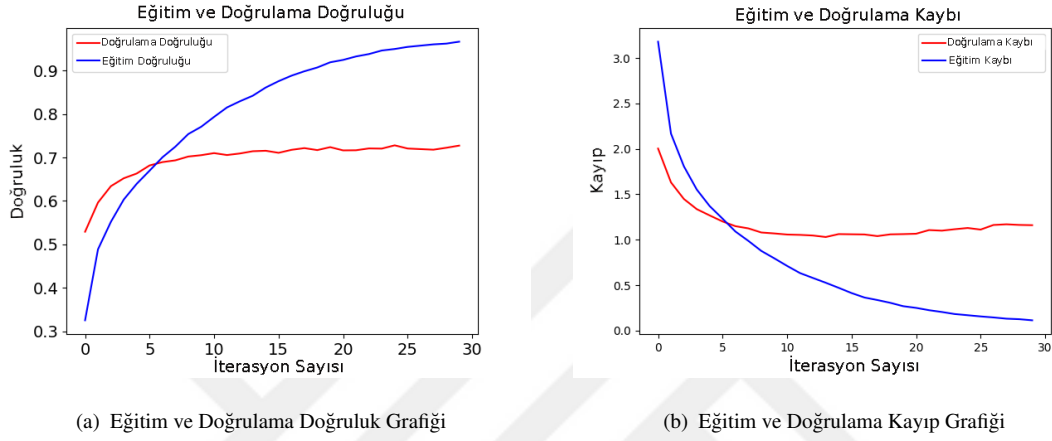
Tablo 4.4. Inceptionv3 Ağıyla Eğitilen Ağların Deney-1 Eğitim Doğruluk ve Kayıp Sonuçları

		Eğitim		Doğrulama		Doğruluk	
		Doğruluk	Kayıp	Doğruluk	Kayıp	İlk-1 (Top-1)	İlk-5 (Top-5)
Aktarım Öğrenmesi Yöntemi	Orijinal	0.95	0.17	0.73	1.13	71.4%	93.8%
	Kısmi Veri Arttırma	0.92	0.27	0.72	1.07	71.0%	93.7%
	Tam Veri Arttırma	0.73	0.91	0.72	0.96	70.9%	93.9%
İnce Ayar Yöntemi	Orijinal	0.98	0.05	0.74	1.53	72.8%	93.9%
	Kısmi Veri Arttırma	0.97	0.08	0.74	1.40	72.9%	94.7%
	Tam Veri Arttırma	0.81	0.63	0.76	0.91	74.6%	95.6%
Tüm Katmanların Eğitilmesi Yöntemi	Orijinal	0.98	0.05	0.76	1.64	74.3%	94.3%
	Kısmi Veri Arttırma	0.85	0.46	0.77	0.92	75.4%	95.8%
	Tam Veri Arttırma	0.86	0.44	0.79	0.83	77.4%	97.0%

Deney-1 kapsamında gerçekleştirilen her bir eğitim sonucunda elde edilen eğitim ve doğrulama doğruluğu ve kaybı grafikleri paylaşılmıştır. Gerçekleştirilen eğitimlerde, doğrulama doğruluğu her bir iterasyonda hesaplanarak 30 iterasyonda en yüksek doğrulama doğruluğuna sahip ağırlık kaydedilmiştir. Eğitim toplamda 30 iterasyonda gerçekleştirilmiştir. Eğitimde her bir iterasyonda 859 ayrı iterasyon kullanılmıştır (adım/iterasyon).

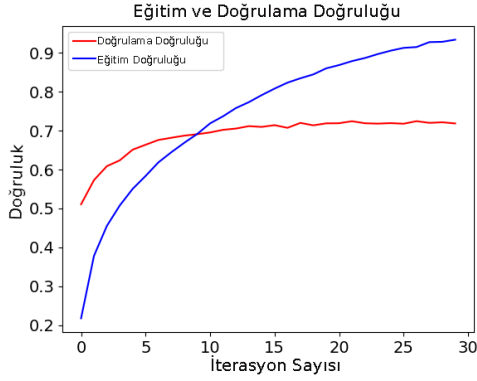
Şekil 4.8.'de aktarım öğrenmesi yöntemi kullanılarak orijinal Set-1 (118 Sınıf) veri kümesinin Inceptionv3 ağında eğitilerek elde edilen eğitim ve doğrulama doğruluğu ve kaybı grafikleri paylaşılmıştır. Orijinal Set-1 veri kümesi 118 sınıftan oluşmaktadır ve dengesiz dağılıma sahip bir veri kümesidir. Bir sınıf 3 örnek içerirken diğer bir sınıf 1000 örnek içerebilmektedir. Bu durum modelin istenilen yüksek doğrulama doğruluğuna ulaşmasını zor

hale getirmektedir. Şekil 4.8.'de eğitim ve doğrulama doğruluğu arasındaki fark yaklaşık 6. epoch'tan sonra artmıştır. Bu durum modelin yeteri kadar öğrenemediği anlamına gelmektedir. Doğrulama doğruluğu 0.7 civarında sabitlenmiş ve eğitim boyunca artmamıştır. Benzer şekilde, doğrulama ve eğitim kaybının birbirine paralel olması gerekirken 6.epoch'tan sonra doğrulama kaybı artmıştır.

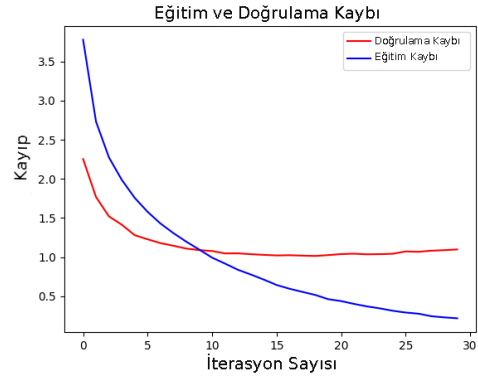


Şekil 4.8. Aktarım Öğrenmesi Yöntemi, Orijinal Veri Kümesi Eğitim Doğruluk ve Kayıp Sonucu

Veri sayısı 60'dan az olan sınıflara veri arttırımı uygulanarak elde edilen kısmi veri kümesi aktarım öğrenme yöntemi kullanılarak Inceptionv3 ağında eğitilmiştir. Şekil 4.9.'da elde edilen eğitim ve doğrulama doğruluğu ve kaybı grafikleri paylaşılmıştır. Eğitim ve doğrulama doğrulukları arasındaki fark, Şekil 4.8.'e göre azalmıştır. Kayıplar arasındaki fark da yine aynı şekilde azalmış ve birbirine daha çok yaklaşmıştır. Kısmi veri arttırma yöntemi ile eğitilen model, orijinal veri kümesi ile eğitilen modele göre daha iyi sonuç elde etmiştir.



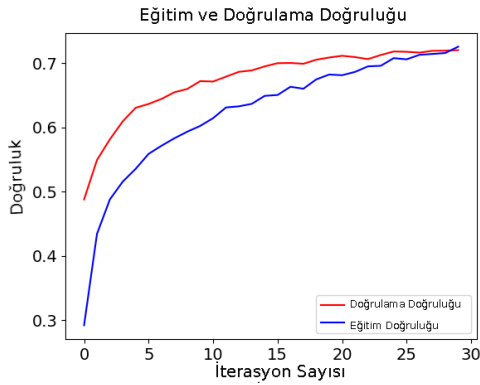
(a) Eğitim ve Doğrulama Doğruluk Grafiği



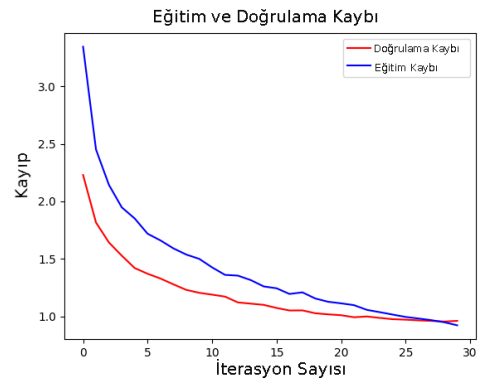
(b) Eğitim ve Doğrulama Kayıp Grafiği

Şekil 4.9. Aktarım Öğrenmesi Yöntemi, Kısmi Veri Arttırma Veri Kümesi Eğitim Doğruluk ve Kayıp Sonucu

Veri sayısına bakılmaksızın, sınıflara veri arttırımı uygulanarak elde edilen tam veri arttırma ile elde edilen veri kümesi aktarım öğrenme yöntemi kullanılarak Inceptionv3 ağında eğitilmiştir. Şekil 4.10.'da elde edilen eğitim ve doğrulama doğruluğu ve kaybı grafikleri paylaşılmıştır. Eğitim ve doğrulama doğrulukları arasındaki fark, Şekil 4.8. ve Şekil 4.9.'a göre azalmıştır. Kayıplar arasındaki fark da yine aynı şekilde azalmış ve birbirine daha çok yaklaşmıştır. Tam veri arttırma yöntemi uygulanarak elde edilen veri kümesi ile eğitilen model, orijinal ve kısmi veri arttırma uygulanan veri kümeleri ile eğitilen modellere göre daha iyi sonuç elde etmiştir.



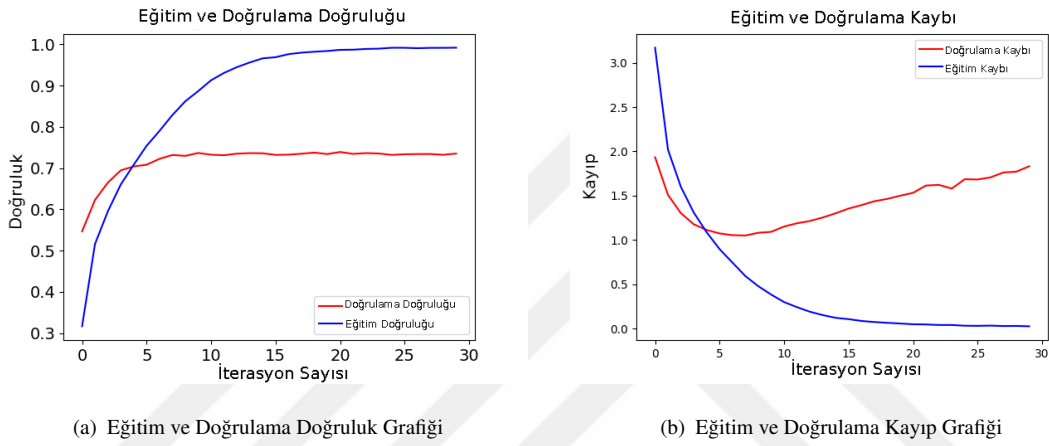
(a) Eğitim ve Doğrulama Doğruluk Grafiği



(b) Eğitim ve Doğrulama Kayıp Grafiği

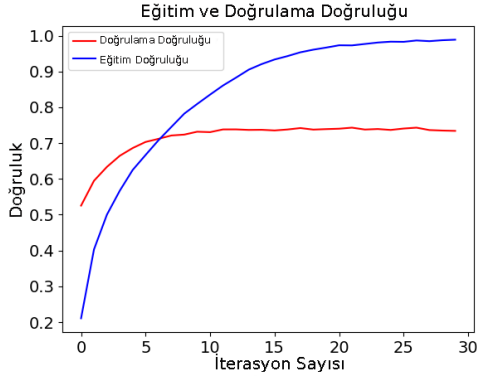
Şekil 4.10. Aktarım Öğrenmesi Yöntemi, Tam Veri Arttırma Veri Kümesi Eğitim Doğruluk ve Kayıp Sonucu

Şekil 4.11.'de ince ayar yöntemi kullanılarak orijinal Set-1 (118 Sınıf) veri kümesinin Inceptionv3 ağında eğitilerek elde edilen eğitim ve doğrulama doğruluğu ve kaybı grafikleri paylaşılmıştır. Aktarım öğrenmesinde orijinal veri kümesi ile eğitilen modelin sonuçlarına benzer şekilde eğitim ve doğrulama doğruluğu arasındaki fark yaklaşık 4. iterasyondan sonra giderek artmıştır. Doğrulama kaybı eğitim kaybına göre oldukça yüksektir ve bu durum modelin iyi öğrenemediği anlamına gelmektedir. İyi öğrenen bir modelde beklenen davranış, eğitim ve doğrulama doğruluk ve kayıp değerlerinin birbirine paralel şekilde artması ve azalmasıdır.

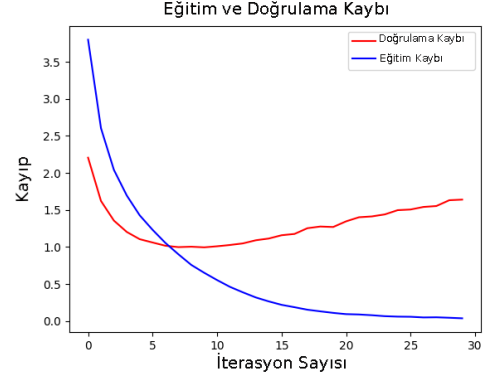


Şekil 4.11. İnce Ayar Yöntemi, Orijinal Veri Kümesi Eğitim Doğruluk ve Kayıp Sonucu

Şekil 4.12.'de ince ayar yöntemi kullanılarak kısmi veri arttırımı uygulanmış Set-1 (118 Sınıf) veri kümesinin Inceptionv3 ağında eğitilerek elde edilen eğitim ve doğrulama doğruluğu ve kaybı grafikleri paylaşılmıştır. Eğitim ve doğrulama doğrulukları arasındaki fark Şekil 4.11.'e göre azalmıştır. Benzer şekilde eğitim ve doğrulama kaybı arasındaki fark da azalmıştır. Bu durum kısmi veri arttırma yöntemi uygulanarak elde edilen kısmi veri kümesi kullanılarak eğitilmiş modelin daha iyi sonuç verdiğini göstermektedir.



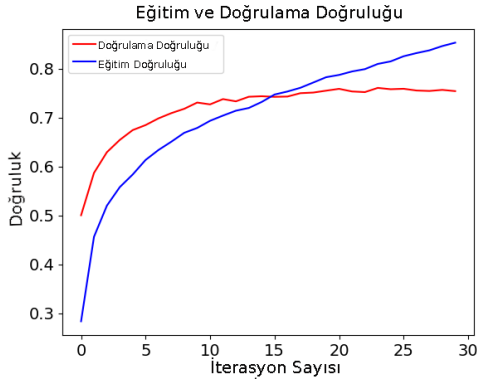
(a) Eğitim ve Doğrulama Doğruluk Grafiği



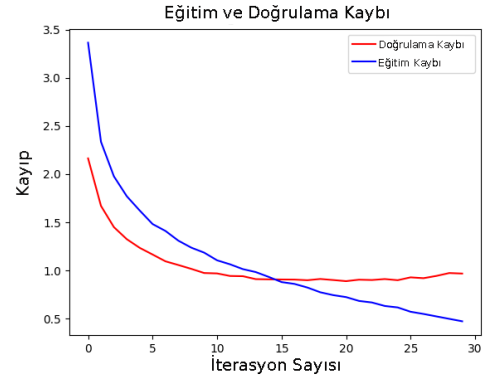
(b) Eğitim ve Doğrulama Kayıp Grafiği

Şekil 4.12. İnce Ayar Yöntemi, Kısmi Veri Arttırma Veri Kümesi Eğitim Doğruluk ve Kayıp Sonucu

Şekil 4.13.'te ince ayar yöntemi kullanılarak tam veri arttırımı uygulanmış Set-1 (118 Sınıf) veri kümesinin Inceptionv3 ağında eğitilerek elde edilen eğitim ve doğrulama doğruluğu ve kaybı grafikleri paylaşılmıştır. Eğitim ve doğrulama doğruluğu arasındaki fark oldukça azalmış ve doğrulama doğruluğu 0.7'nin üzerine çıkmıştır.



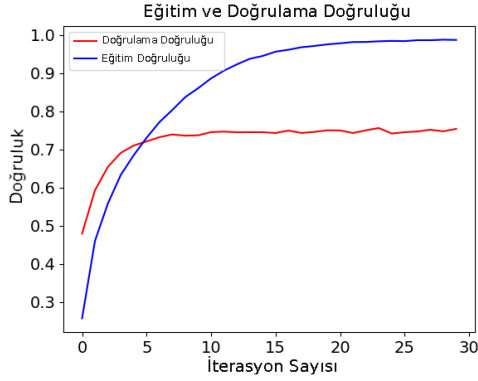
(a) Eğitim ve Doğrulama Doğruluk Grafiği



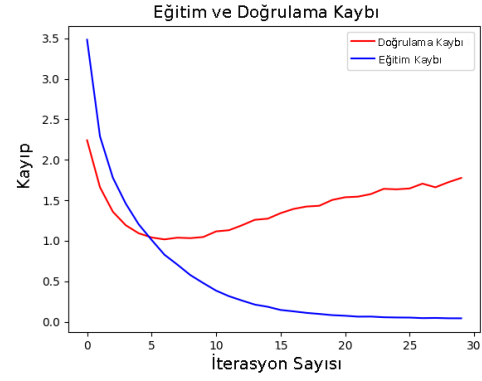
(b) Eğitim ve Doğrulama Kayıp Grafiği

Şekil 4.13. İnce Ayar Yöntemi, Tam Veri Arttırma Veri Kümesi Eğitim Doğruluk ve Kayıp Sonucu

Şekil 4.14.'te tüm katmanların eğitilmesi yöntemi kullanılarak orijinal Set-1 (118 Sınıf) veri kümesinin Inceptionv3 ağında eğitilerek elde edilen eğitim ve doğrulama doğruluğu ve kaybı grafikleri paylaşılmıştır. Eğitim ve doğrulama doğruluğu ve kaybı arasındaki fark azalmamış ve sabit olarak kalmıştır.



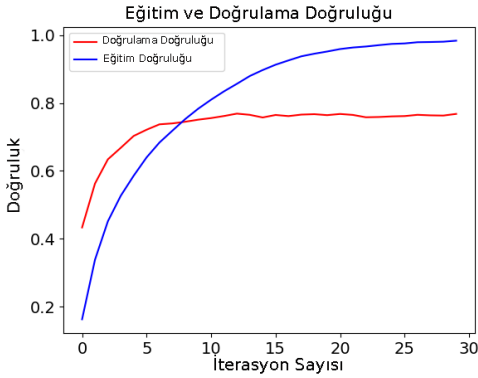
(a) Eğitim ve Doğrulama Doğruluk Grafiği



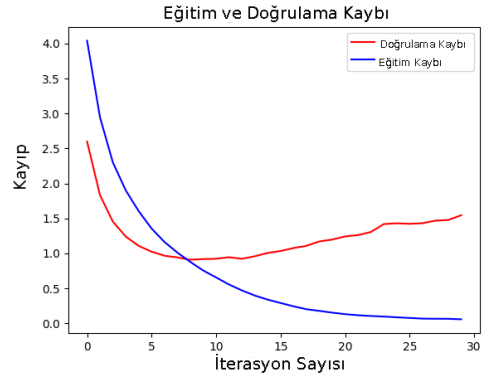
(b) Eğitim ve Doğrulama Kayıp Grafiği

Şekil 4.14. Tüm Katmanların Eğitilmesi Yöntemi, Orijinal Veri Kümesi Eğitim Doğruluk ve Kayıp Sonucu

Şekil 4.15.'de tüm katmanların eğitilmesi yöntemi kullanılarak kısmi veri artırma uygulanarak elde edilmiş Set-1 (118 Sınıf) veri kümesinin Inceptionv3 ağında eğitilerek elde edilen eğitim ve doğrulama doğruluğu ve kaybı grafikleri paylaşılmıştır. Eğitim ve doğrulama doğruluğu arasındaki fark orijinal veri kümesi ile eğitilen modelin sonucuna göre azalmıştır.



(a) Eğitim ve Doğrulama Doğruluk Grafiği

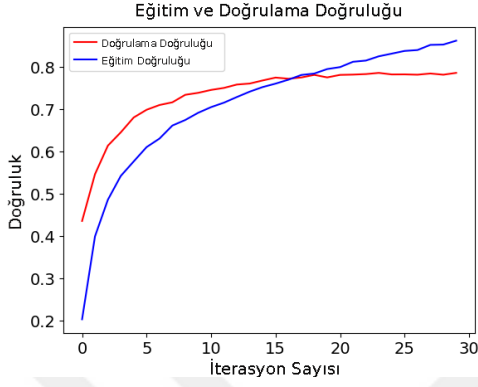


(b) Eğitim ve Doğrulama Kayıp Grafiği

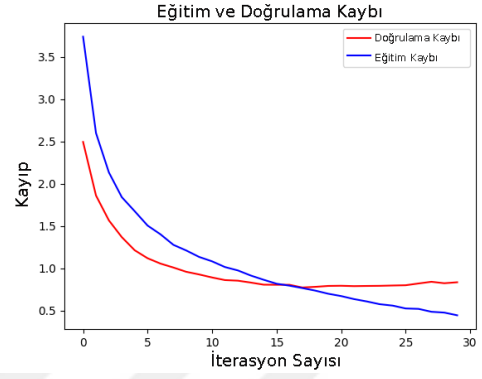
Şekil 4.15. Tüm Katmanların Eğitilmesi Yöntemi, Kısmi Veri Artırma Veri KÜmesi Eğitim Doğruluk ve Kayıp Sonucu

Şekil 4.16.'da tüm katmanların eğitilmesi yöntemi kullanılarak tam veri artırma uygulanarak elde edilmiş Set-1 (118 Sınıf) veri kümesinin Inceptionv3 ağında eğitilerek elde edilen eğitim ve doğrulama doğruluğu ve kaybı grafikleri paylaşılmıştır. Tam veri artırma yöntemi

modelin daha iyi öğrenmesini sağlamıştır. Doğrulama ve eğitim doğruluğu birbirini takip etmiştir.



(a) Eğitim ve Doğrulama Doğruluk Grafiği

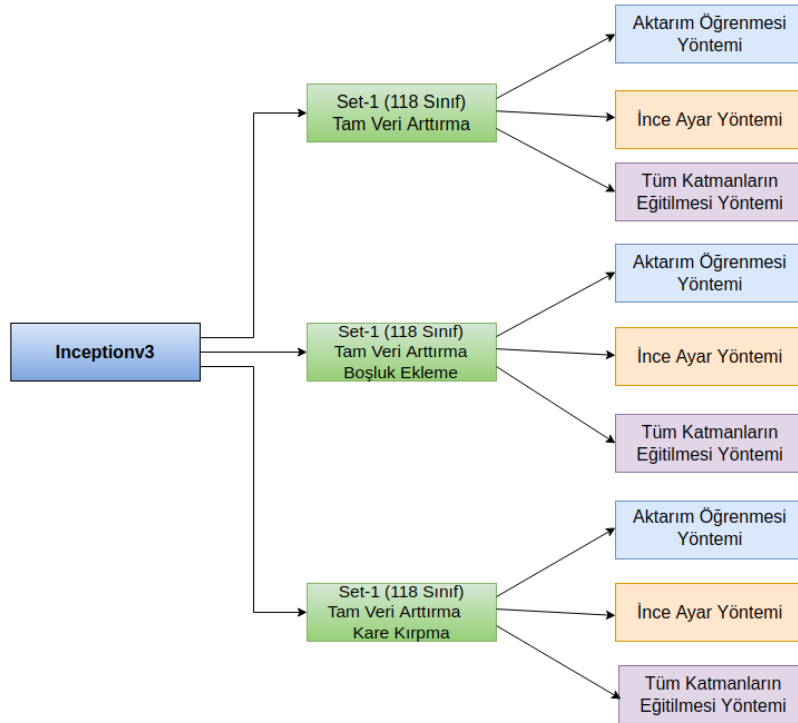


(b) Eğitim ve Doğrulama Kayıp Grafiği

Şekil 4.16. Tüm Katmanların Eğitilmesi Yöntemi, Tam Veri Arttırma Veri Kümesi Eğitim Doğruluk ve Kayıp Sonucu

4.4.4. Deney-2

Deney-1 sonucuna göre en iyi sonucu veren veri kümesi ve yöntem seçilmiştir. Tüm katmanların eğitilmesi yöntemi ve tam veri artırma ile elde edilmiş veri kümesi en iyi sonucu veren yöntem ve veri kümesi olmuştur. Plankton görüntüleri çok farklı en-boy oranına sahiptir. Evrişimli sinir ağlarının girdisi sabit boyutta, tez kapsamında 224x224, görüntü kabul ettiği için görüntüler yeniden boyutlandırılarak ağlara sokulmaktadır. Bu da plankton görüntülerinde bilgi kaybına neden olmaktadır. Bu bilgi kaybının önlenebilirliğinin araştırılması için boşluk ekleme ve kare kırpma yöntemleri plankton görüntülerine uygulanmıştır. Bu deneyde Inceptionv3 ağında Set-1 (118 Sınıf) tam veri artırma ile elde edilmiş veri kümesine kare kırpma ve boşluk ekleme ön işlemleri uygulanmıştır. Kare kırpma ve boşluk ekleme ön işleme yöntemleri ile elde edilen veri kümeleri, aktarım öğrenmesi, ince ayar ve tüm katmanların eğitilmesi yöntemleri kullanılarak eğitilmiş ve elde edilen sonuçlar paylaşılmıştır. Bölüm 4.1.'de ön işleme ayrıntılarından bahsedilmiştir. Deney 2, Şekil 4.17.'de görselleştirilmiştir.



Şekil 4.17. Deney-2'nin Özeti. Aktarım Öğrenmesi Yöntemi, İnce Ayar Yöntemi, Tüm Katmanların Eğitilmesi Yöntemi

4.4.5. Deney-2 Sonuç

Aktarım öğrenmesi, ince ayar ve tüm katmanların eğitilmesi yöntemleri ve her bir yöntem için Set-1 (118 Sınıf) tam veri arttırma ile elde edilmiş veri kümesinin orijinal, boşluk ekleme, kare kırpma veri kümeleriyle gerçekleştirilen dokuz farklı deneyin deney sonuçları Tablo 4.5.'de paylaşılmıştır. Tabloda her bir deney için kesinlik, geri çağırma, F1-Skoru metrikleri makro ortalama ve ağırlıklı ortalama yöntemleri kullanılarak ayrı ayrı hesaplanmıştır. Tablo 4.6.'da eğitim ve doğrulama doğruluk ve kayıp değerleri her bir deney için paylaşılmıştır. Ayrıca eğitilen modelin, test veri kümesi kullanılarak test edilmesi sonucu elde edilen ilk-1 (Top-1) ve ilk-5 (Top-5) sonuçları her bir deney için paylaşılmıştır.

Set-1 (118 Sınıf) tam veri arttırma ile elde edilmiş veri kümesinde Inceptionv3 ağı kullanılarak gerçekleştirilen Deney-2 kapsamında en iyi sonuç, ön işleme uygulanmamış orijinal Set-1 tam veri arttırma veri kümesinin tüm katmanlarının aktif edilerek Yöntem-3 ile eğitilmesi sonucu elde edilmiştir. Tabloda her bir metrik için elde edilmiş en iyi sonuçlar koyu renk ile gösterilmiştir. Deney-2 kapsamında paylaşılan tablolarda yer alan sonuçlara bakıldığında ön işleme işlemlerinin ağırlıklı ortalama kesinlik, geri çağırma, F1-Skoru metriklerini negatif yönde etkilediği görülmektedir. Makro ortalama ise kesinlik metriği dışındaki geri çağırma ve F1-Skoru metriklerini negatif yönde etkilediği gözlemlenmektedir. Eğitim ve doğrulama doğruluk değerlerinde en yüksek değere sahip sonucu orijinal veri kümesi ve tüm katmanların eğitilmesi yöntemi vermiştir. İlk-1 ve ilk-5 doğruluğa bakıldığında ise en iyi sonuç yaklaşık %2 farkla kare kırpma yönteminde elde edilmiştir.

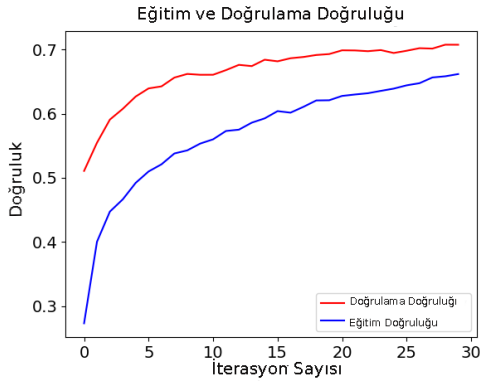
Tablo 4.5. Set-1 Veri Kümesinde Elde Edilen Deney-2 Sonuçları

		Makro Ortalama			Ağırlıklı Ortalama		
		Kesinlik	Geri Çağırma	F1-Skoru	Kesinlik	Geri Çağırma	F1-Skoru
Aktarım Öğrenmesi Yöntemi	Orijinal	55%	50%	51%	70%	71%	70%
	Boşluk Ekleme	53%	50%	50%	68%	69%	68%
	Kare Kırpma	57%	49%	50%	69%	70%	69%
İnce Ayar Yöntemi	Orijinal	58%	56%	56%	74%	75%	74%
	Boşluk Ekleme	56%	52%	53%	70%	71%	70%
	Kare Kırpma	59%	57%	57%	73%	73%	73%
Tüm Katmanların Eğitilmesi Yöntemi	Orijinal	63%	62%	62%	77%	77%	77%
	Boşluk Ekleme	65%	61%	61%	76%	76%	76%
	Kare Kırpma	61%	60%	59%	76%	77%	76%

Tablo 4.6. Inceptionv3 Ağıyla Eğitilen Ağların Deney-2 Eğitim Doğruluk ve Kayıp Sonuçları

		Eğitim		Doğrulama		Doğruluk	
		Doğruluk	Kayıp	Doğruluk	Kayıp	İlk-1 (Top-1)	İlk-5 (Top-5)
Aktarım Öğrenmesi Yöntemi	Orijinal	0.73	0.91	0.72	0.96	71.4%	93.8%
	Boşluk Ekleme	0.66	1.18	0.71	1.01	68.9%	93.5%
	Kare Kırpma	0.72	0.95	0.71	1.01	70.3%	93.9%
İnce Ayar Yöntemi	Orijinal	0.81	0.63	0.76	0.91	72.8%	93.9%
	Boşluk Ekleme	0.73	0.92	0.73	0.99	71.2%	94.4%
	Kare Kırpma	0.85	0.52	0.75	0.98	73.4%	95.6%
Tüm Katmanların Eğitilmesi Yöntemi	Orijinal	0.86	0.44	0.79	0.83	74.3%	94.3%
	Boşluk Ekleme	0.81	0.61	0.78	0.84	76.1%	96.4%
	Kare Kırpma	0.84	0.52	0.77	0.91	76.8%	96.8%

Şekil 4.18.'de aktarım öğrenmesi yöntemi kullanılarak tam veri artırma ve boşluk doldurma ön işlemi uygulanmış Set-1 (118 Sınıf) veri kümesinin Inceptionv3 ağında eğitilerek elde edilen eğitim ve doğrulama doğruluğu ve kaybı grafikleri paylaşılmıştır. Eğitim sonuçları ön işleme uygulanmamış Şekil 4.16.'ya göre kötüleşmiştir. Eğitim, doğrulama doğruluğu ve eğitim, doğrulama kaybı arasındaki fark artmıştır. Doğrulama doğruluğu azalmış ve doğrulama kaybı artmıştır.



(a) Eğitim ve Doğrulama Doğruluk Grafiği

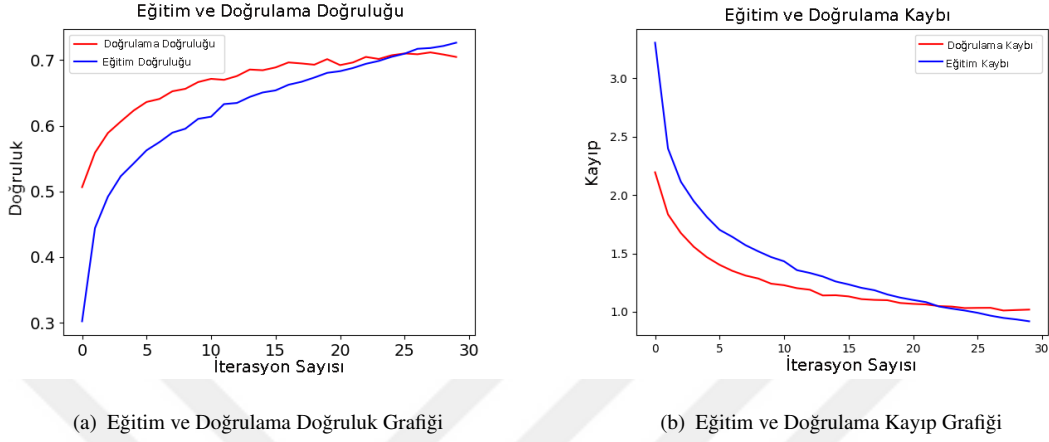


(b) Eğitim ve Doğrulama Kayıp Grafiği

Şekil 4.18. Aktarım Öğrenmesi Yöntemi, Tam Veri Arttırma / Boşluk Doldurma Veri Kümesi Eğitim Doğruluk ve Kayıp Sonucu

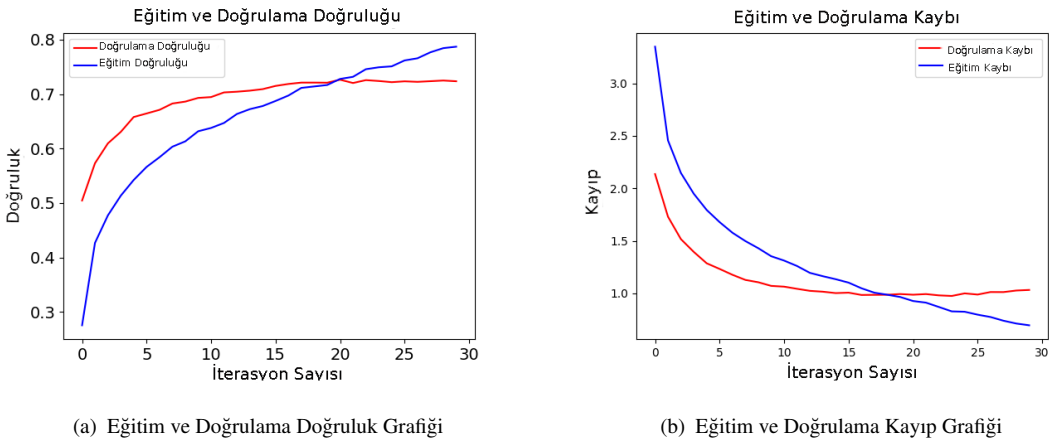
Şekil 4.19.'da aktarım öğrenmesi yöntemi kullanılarak tam veri artırma ve kare kırpma ön işleme uygulanmış Set-1 (118 Sınıf) veri kümesinin Inceptionv3 ağında eğitilerek elde edilen

eğitim ve doğrulama doğruluğu ve kaybı grafikleri paylaşılmıştır. Şekil 4.18.'e göre eğitim doğruluğu artmış ve kaybı azalmıştır. Doğrulama doğruluğu ve kaybı sabit kalmıştır.



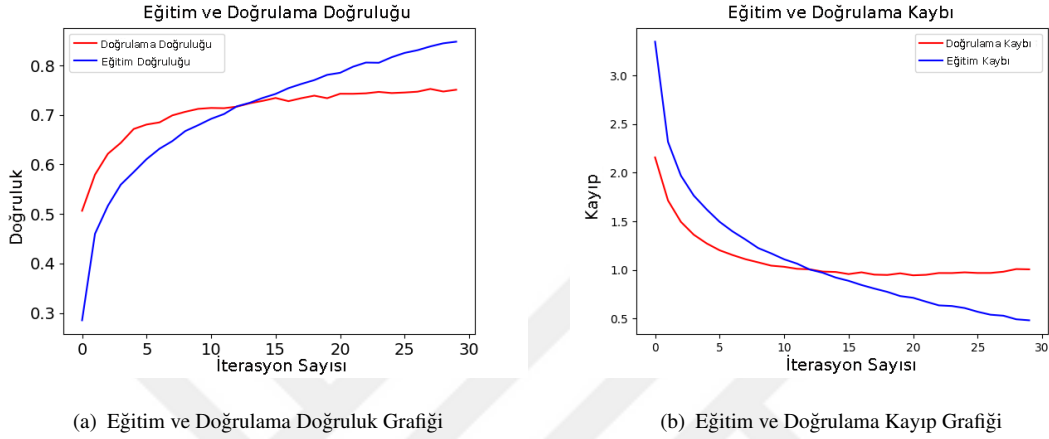
Şekil 4.19. Aktarım Öğrenmesi Yöntemi, Tam Veri Arttırma / Kare Kırpma Veri Kümesi Eğitim Doğruluk ve Kayıp Sonucu

Şekil 4.20.'de ince ayar yöntemi kullanılarak tam veri arttırma ve boşluk doldurma ön işlemi uygulanmış Set-1 (118 Sınıf) veri kümesinin Inceptionv3 ağında eğitilerek elde edilen eğitim ve doğrulama doğruluğu ve kaybı grafikleri paylaşılmıştır. Aynı veri kümesi fakat aktarım öğrenmesi yöntemi kullanan model ile elde edilen sonuçlara göre daha iyi sonuçlar elde edilmiştir. İnce ayar yöntemi aktarım öğrenmesi yöntemine göre iyi sonuç verse de, boşluk ekleme ön işlemi on işleme yapılmamış veri kümesi ile eğitilen modele göre %3 doğrulama kaybı yaşamıştır.



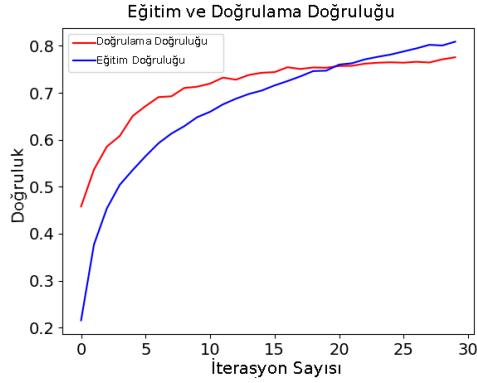
Şekil 4.20. İnce Ayar Yöntemi, Tam Veri Arttırma / Boşluk Doldurma Veri Kümesi Eğitim Doğruluk ve Kayıp Sonucu

Şekil 4.21.'de ince ayar yöntemi kullanılarak tam veri arttırma ve kare kırpma ön işleme uygulanmış Set-1 (118 Sınıf) veri kümesinin Inceptionv3 ağında eğitilerek elde edilen eğitim ve doğrulama doğruluğu ve kaybı grafikleri paylaşılmıştır. Şekil 4.20.'de elde edilen eğitim sonuçlarına göre doğrulama doğruluğu %2, eğitim doğruluğu %12 artmıştır.

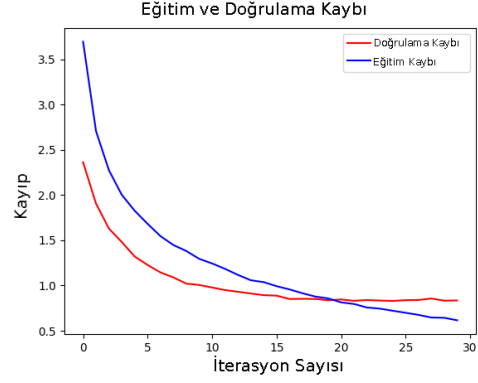


Şekil 4.21. İnce Ayar Yöntemi, Tam Veri Arttırma / Kare Kırpma Veri Kümesi Eğitim Doğruluk ve Kayıp Sonucu

Şekil 4.22.'de tüm katmanların eğitilmesi yöntemi kullanılarak tam veri arttırma ve boşluk doldurma ön işleme uygulanmış Set-1 (118 Sınıf) veri kümesinin Inceptionv3 ağında eğitilerek elde edilen eğitim ve doğrulama doğruluğu ve kaybı grafikleri paylaşılmıştır. %78 doğrulama doğruluğu ve %86 eğitim doğruluğu elde edilmiştir. Diğer ön işleme uygulanan eğitim sonuçlarına göre iyi olmasına rağmen, orijinal tam veri arttırımı uygulanarak eğitilmiş modelin doğrulama doğruluğunun altında kalmıştır.



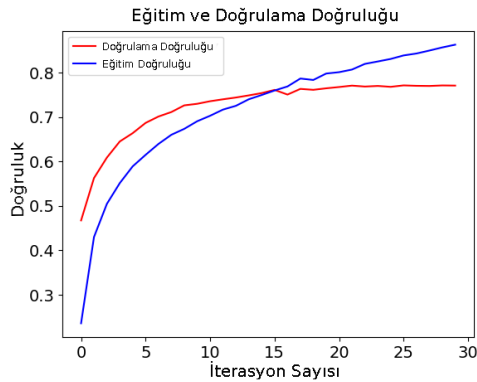
(a) Eğitim ve Doğrulama Doğruluk Grafiği



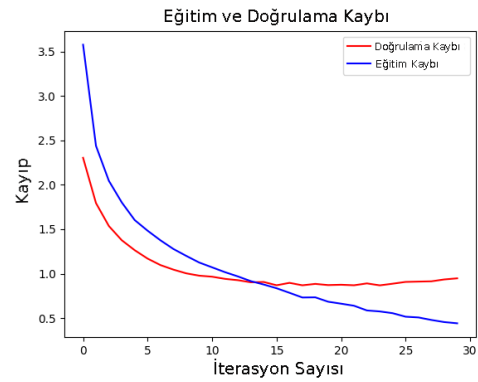
(b) Eğitim ve Doğrulama Kayıp Grafiği

Şekil 4.22. Tüm Katmanların Eğitilmesi Yöntemi, Tam Veri Arttırma / Boşluk Doldurma Veri Kümesi Eğitim Doğruluk ve Kayıp Sonucu

Şekil 4.23.'te tüm katmanların eğitilmesi yöntemi kullanılarak tam veri artırma ve kare kırpma ön işleme uygulanmış Set-1 (118 Sınıf) veri kümesinin Inceptionv3 ağında eğitilerek elde edilen eğitim ve doğrulama doğruluğu ve kaybı grafikleri paylaşılmıştır. Şekil 4.22.'de elde edilen doğrulama doğruluğu sonuçlarından %1 daha az doğruluk oranına sahiptir.



(a) Eğitim ve Doğrulama Doğruluk Grafiği



(b) Eğitim ve Doğrulama Kayıp Grafiği

Şekil 4.23. Tüm Katmanların Eğitilmesi Yöntemi, Tam Veri Arttırma / Kare Kırpma Veri Kümesi Eğitim Doğruluk ve Kayıp Sonucu

4.4.6. Deney-3

Deney-3 kapsamında, Deney-1 (Eğitim yöntemlerinin etkisinin incelendiği deney) ve Deney-2'de (Ön işleme yöntemlerinin etkisinin incelendiği deney) en iyi sonucu veren tüm katmanların eğitilmesi yöntemi ve tam veri attırma ile elde edilmiş veri kümesi kullanılmıştır. Deney-1 ve 2'den elde edilen en iyi veri küme seçimi olarak karar verilen tam veri arttırma ile elde edilmiş veri kümesi seçimi, Set-2 veri kümesine uygulanmıştır. Set-2 veri kümesinin tam veri arttırma uygulanarak elde edilen toplam veri sayısı, sınıf sayısı, eğitim, doğrulama ve test kümesinde bulunan veri sayısı miktarı Tablo 4.7.'de paylaşılmıştır.

Tam veri arttırma uygulanmış Set-1 (118 Sınıf) ve Set-2 (38 Sınıf) veri kümeleri kullanılarak InceptionResNetv2, DenseNet, ResNet-50, VGG-16 ağlarında eğitimler yapılmış ve eğitilen ağın tüm katmanları aktif edilerek, tüm katmanların eğitilmesi yöntemi ile eğitim gerçekleştirilmiştir. Inceptionv3 ağında Set-1 tam veri arttırma kümesi ile yapılan eğitim Deney-1'de Tablo 4.5.'te paylaşılmıştır. Inceptionv3 ağında Set-2 tam veri arttırma ile elde edilmiş veri kümesi kullanılarak tüm katmanların eğitilmesi yöntemi ile eğitim gerçekleştirilmiştir. Bu deneyde Deney-1 ve 2 sonucunda çıkarım yapılan en iyi yöntem ve veri kümesi olan tüm katmanların eğitilmesi yöntemi ve tam veri arttırımı ile elde edilmiş veri kümesi kullanılarak sonuçların diğer ağlarda elde edilmesi ve gözlemlenmesi hedeflenmiştir.

Tablo 4.7. Set-2 Veri Kümesine Tam Veri Arttırım Uygulanmasının Sonuçları

Set-2 Veri Kümesi ve Tam Veri Arttırma Yöntemi				
#Toplam Veri	#Sınıf	#Eğitim Seti	#Doğrulama Seti	#Test Seti
72501	38	69591	1455	1455

4.4.7. Deney-3 Sonuç

Tablo 4.8.'de 118 sınıfa sahip Set-1 tam veri arttırma uygulanmış veri kümesinin eğitim sonucunda elde edilen sonuçlar paylaşılmıştır. Tabloda her bir ağ için kesinlik, geri çağırma, F1-Skoru metrikleri makro ortalama ve ağırlıklı ortalama yöntemleri kullanılarak ayrı ayrı hesaplanmıştır. Inceptionv3 ağında %77, InceptionResNetv2, DenseNet ve ResNet-50 ağlarında %78 ağırlıklı F1-Skoru elde edilmiştir. VGG-16 ağında diğer ağlara göre %5'lik azalma

oranıyla %73'lük ağırlıklı F1-Skoru ve %3'lük bir azalma ile 0.76 doğrulama doğruluğu elde edilmiştir.

Tablo 4.9.'da 38 sınıfa sahip Set-2 tam veri arttırma uygulanmış veri kümesinin eğitim sonucunda elde edilen sonuçlar paylaşılmıştır. Tabloda her bir ağ için kesinlik, geri çağırma, F1-Skoru metrikleri makro ortalama ve ağırlıklı ortalama yöntemleri kullanılarak ayrı ayrı hesaplanmıştır. Set-2 veri kümesinden elde edilen sonuçlar Set-1 veri kümesine göre daha yüksek yüzdelere sahiptir. Bunun sebebi Set-2 veri kümesinin dengeli dağılıma sahip minimum 100 örnekten oluşan Set-1 veri kümesinin alt kümesine sahip olmasından kaynaklanmaktadır. InceptionResNetv2, DenseNet ve ResNet-50 ağlarında sırasıyla %91, %92, %92, %91 ağırlıklı F1-Skorları elde edilmiştir. VGG-16 ağında diğer ağlara göre daha düşük oranda ağırlıklı F1-Skoru elde edilmiştir. Set-1 veri kümesinde VGG-16 ve diğer ağlar arasında elde edilen yüzdesel fark Set-2'de elde edilen farktan yaklaşık %4 daha fazladır.

Tablo 4.10.'da eğitim ve doğrulama doğruluk ve kayıp değerleri Set-1 ve Set-2 veri kümesi ve her bir ağ için paylaşılmıştır. Set-1 veri kümesinde VGG-16 ağında 0.76, diğer ağlarda 0.79 doğrulama doğruluğu elde edilmiştir. Set-2 veri kümesinde VGG-16 ağında %92, diğer ağlarda %93 doğrulama doğruluğu elde edilmiştir. Tablo 4.10.'da eğitilen ağ modellerinin test veri kümesi üzerinde elde edilen ilk-1 ve ilk-5 doğruluk değerleri paylaşılmıştır. Set-1 (118 Sınıf) veri kümesinden elde edilen en yüksek ilk-1 (top-1) doğruluk %78.6 ile DenseNet ağından, ilk-5 (top-5) doğruluk %97.1 ile inceptionResNetv2 ağında elde edilmiştir. Set-2 (38 Sınıf) veri kümesinden elde edilen en yüksek ilk-1 doğruluk (top-1) %92.1 doğruluk değeri ile DenseNet ağından ve ilk-5 (top-5) doğruluk ise %99.1 ile ResNet-50 ağından elde edilmiştir.

Ağırlıklı ortalama ile elde edilen sonuçlar, dengesiz bir veri dağılımına sahip olan Set-1 (118 Sınıf) veri kümesinin değerlendirmesinde daha kritik bir role sahiptir. Bunun nedeni, ağırlıklı ortalama yöntemi ile sınıfların sahip olduğu örnek sayısına göre ağırlıklandırılmasıdır. Ağırlıklı ortalama yöntemi kullanılarak elde edilen kesinlik, geri çağırma ve F1-Skoru sonuçlarına bakıldığında en iyi sonucu veren ağ DenseNet ağıdır. Set-2 veri kümesinin ağırlıklı ortalama yöntemi ile elde edilen sonuçlarında da en iyi sonuçlar DenseNet ağına sahiptir. Bu deney sonrasında yapılacak olan deneylerde bu nedenle DenseNet ağı kullanılmıştır.

Tablo 4.8. Set-1 Veri Kümesinde Elde Edilen Ağ Sonuçları

	Makro Ortalama			Ağırlıklı Ortalama		
	Kesinlik	Geri Çağırma	F1-Skoru	Kesinlik	Geri Çağırma	F1-Skoru
Inceptionv3	%63	%62	%62	%77	%77	%77
InceptionResNetv2	%67	%66	%66	%78	%78	%78
DenseNet	%66	%61	%61	%78	%79	%78
ResNet50	%64	%63	%63	%77	%78	%78
VGG16	%57	%53	%53	%73	%74	%73

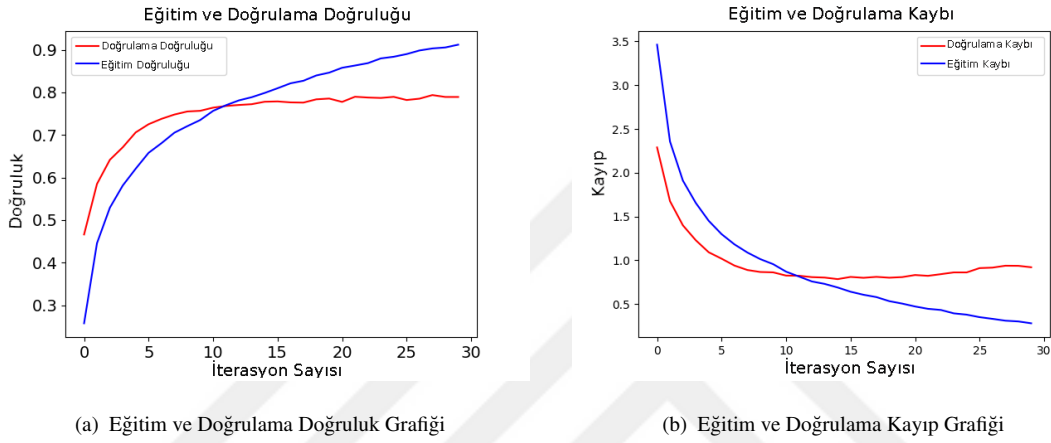
Tablo 4.9. Set-2 Veri Kümesinde Elde Edilen Ağ Sonuçları

	Makro Ortalama			Ağırlıklı Ortalama		
	Kesinlik	Geri Çağırma	F1-Skoru	Kesinlik	Geri Çağırma	F1-Skoru
Inceptionv3	%90	%88	%89	%92	%91	%91
InceptionResNetv2	%90	%89	%89	%92	%92	%91
DenseNet	%90	%90	%90	%92	%92	%92
ResNet50	%89	%89	%89	%92	%92	%92
VGG16	%90	%87	%88	%91	%91	%91

Tablo 4.10. Set-1 ve Set-2 Veri Kümelerinin Eğitim ve Doğrulama Doğruluk/Kayıp Sonuçları

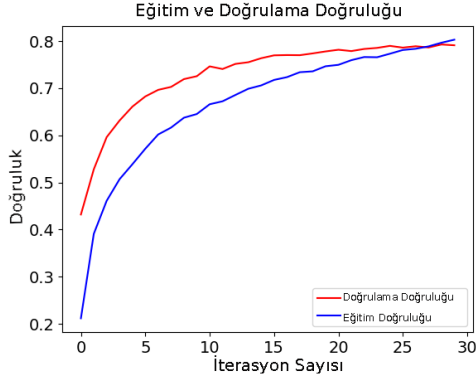
		Eğitim		Doğrulama		Doğrulama	
		Doğruluk	Kayıp	Doğruluk	Kayıp	İlk-1 (Top-1)	İlk-5 (Top-5)
Set-1 (118 Sınıf)	Inceptionv3	0.86	0.44	0.79	0.83	77.4%	97.0%
	InceptionResNetv2	0.90	0.32	0.79	0.94	78.4%	97.1%
	DenseNet	0.79	0.68	0.79	0.74	78.6%	97.0%
	ResNet-50	0.87	0.41	0.79	0.86	78.4%	97.0%
	VGG-16	0.78	0.73	0.76	0.84	74.3%	95.6%
Set-2 (38 Sınıf)	Inceptionv3	0.99	0.04	0.93	0.46	91.4%	98.9%
	InceptionResNetv2	0.99	0.03	0.93	0.45	91.6%	98.6%
	DenseNet	0.95	0.15	0.93	0.31	92.1%	98.8%
	ResNet-50	0.99	0.03	0.93	0.42	92.0%	99.1%
	VGG-16	0.94	0.20	0.92	0.32	91.0%	99.0%

Set-1 (118 Sınıf) veri kümesinin ağlarda gerçekleşen eğitimleri toplamda 30 iterasyon gerçekleştirilmiştir. Tüm eğitim boyunca en yüksek doğrulama doğruluğuna sahip model kaydedilmiştir. Şekil 4.24.'te tüm katmanların eğitilmesi yöntemi kullanılarak tam veri artırma uygulanmış Set-1 (118 Sınıf) veri kümesinin InceptionResNetv2 ağında eğitilerek elde edilen eğitim ve doğrulama doğruluğu ve kaybı grafikleri paylaşılmıştır. %79 doğrulama ve %90 eğitim doğruluğu elde edilmiştir.

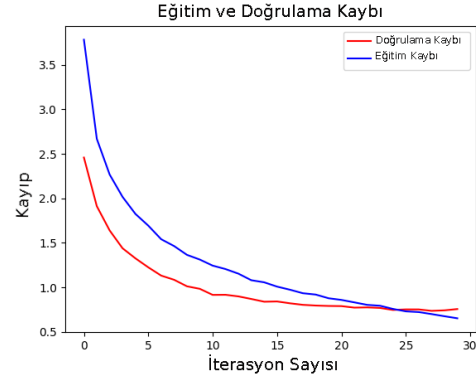


Şekil 4.24. Set-1, InceptionResNetv2, Tüm Katmanların Eğitilmesi Yöntemi, Tam Veri Artırma Veri Kümesi Eğitim Doğruluk ve Kayıp Sonucu

Şekil 4.25.'de tüm katmanların eğitilmesi yöntemi kullanılarak tam veri artırma uygulanmış Set-1 (118 Sınıf) veri kümesinin DenseNet ağında eğitilerek elde edilen eğitim ve doğrulama doğruluğu ve kaybı grafikleri paylaşılmıştır. %79 doğrulama ve %79 eğitim doğruluğu elde edilmiştir. Eğitim ve doğrulama doğruluğu ve kaybı arasındaki uzaklık iterasyon sayısı ilerledikçe azalmıştır.



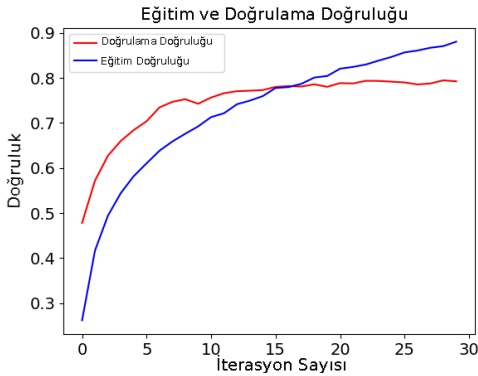
(a) Eğitim ve Doğrulama Doğruluk Grafiği



(b) Eğitim ve Doğrulama Kayıp Grafiği

Şekil 4.25. Set-1 DenseNet Yöntem-3 Tam Veri Arttırma Veri Kümesi Eğitim Doğruluk ve Kayıp Sonucu

Şekil 4.26.'da tüm katmanların eğitilmesi yöntemi kullanılarak tam veri arttırma uygulanmış Set-1 (118 Sınıf) veri kümesinin ResNet50 ağında eğitilerek elde edilen eğitim ve doğrulama doğruluğu ve kaybı grafikleri paylaşılmıştır. %79 doğrulama ve %87 eğitim doğruluğu elde edilmiştir. Eğitim ve doğrulama doğruluğu ve kaybı arasındaki uzaklık yaklaşık 15. iterasyona kadar azalmış sonrasında artarak devam etmiştir.



(a) Eğitim ve Doğrulama Doğruluk Grafiği

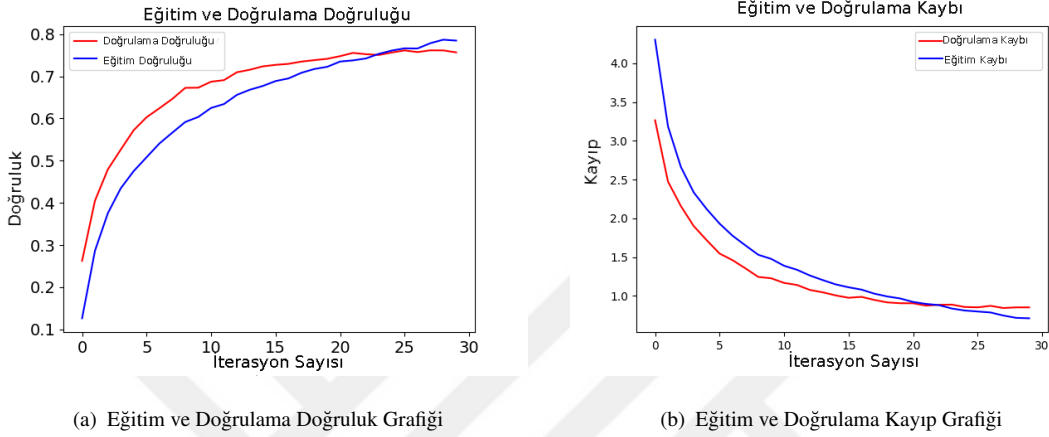


(b) Eğitim ve Doğrulama Kayıp Grafiği

Şekil 4.26. Set-1, ResNet-50, Tüm Katmanların Eğitilmesi Yöntemi, Tam Veri Arttırma Veri Kümesi Eğitim Doğruluk ve Kayıp Sonucu

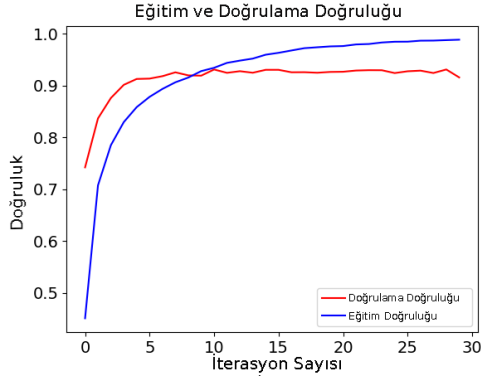
Şekil 4.27.'de tüm katmanların eğitilmesi yöntemi kullanılarak tam veri arttırma uygulanmış Set-1 (118 Sınıf) veri kümesinin VGG-16 ağında eğitilerek elde edilen eğitim ve doğrulama

doğruluğu ve kaybı grafikleri paylaşılmıştır. %76 doğrulama ve %78 eğitim doğruluğu elde edilmiştir. Eğitim, doğrulama doğruluğu ve kaybı birbirine yakın bir şekilde artmış ve azalmıştır. Aşırı uyum (overfitting) problemi görülmemiştir. Fakat diğer ağlara göre doğrulama doğruluğu %3 oranında daha düşük seviyede elde edilmiştir.

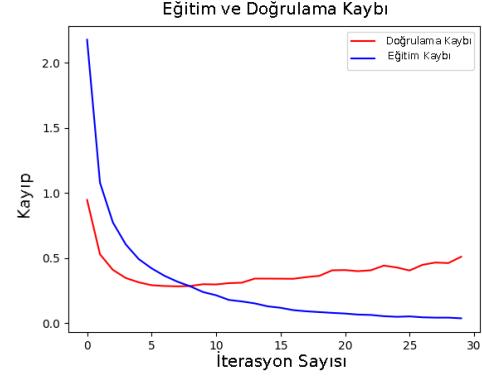


Şekil 4.27. Set-1, VGG-16, Tüm Katmanların Eğitilmesi Yöntemi, Tam Veri Arttırma Veri Kümesi Eğitim Doğruluk ve Kayıp Sonucu

Şekil 4.28.'de tüm katmanların eğitilmesi yöntemi kullanılarak tam veri arttırma uygulanmış Set-2 (38 Sınıf) veri kümesinin Inceptionv3 ağında eğitilerek elde edilen eğitim ve doğrulama doğruluğu ve kaybı grafikleri paylaşılmıştır. Toplamda 30 iterasyon ile gerçekleşen eğitimde %93 doğrulama doğruluğu ve %99 eğitim doğruluğu elde edilmiştir. Eğitim ve doğrulama doğruluğu yaklaşık olarak 10. iterasyonda birbirine yaklaştıktan sonra aralarındaki fark artarak devam etmiştir.



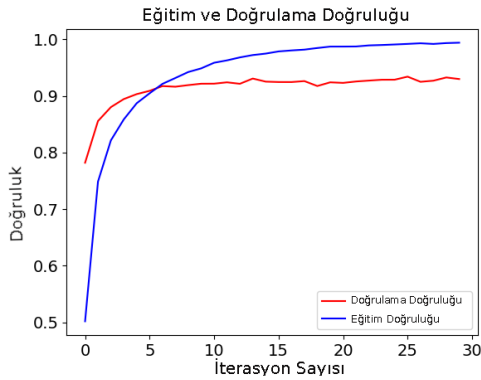
(a) Eğitim ve Doğrulama Doğruluk Grafiği



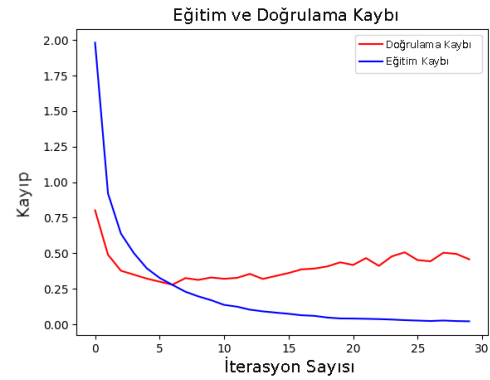
(b) Eğitim ve Doğrulama Kayıp Grafiği

Şekil 4.28. Set-2, Inceptionv3, Tüm Katmanların Eğitilmesi Yöntemi, Tam Veri Arttırma Veri Kümesi Eğitim Doğruluk ve Kayıp Sonucu

Şekil 4.29.'da tüm katmanların eğitilmesi yöntemi kullanılarak tam veri arttırma uygulanmış Set-2 (38 Sınıf) veri kümesinin InceptionResNetv2 ağında eğitilerek elde edilen eğitim ve doğrulama doğruluğu ve kaybı grafikleri paylaşılmıştır. Toplamda 30 iterasyon ile gerçekleşen eğitimde Şekil 4.28. ile benzer sonuçlar elde edilmiştir.



(a) Eğitim ve Doğrulama Doğruluk Grafiği

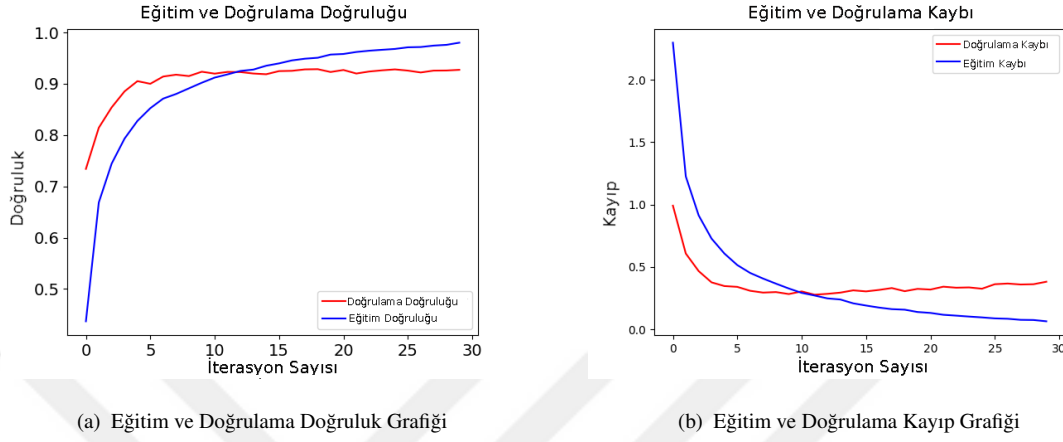


(b) Eğitim ve Doğrulama Kayıp Grafiği

Şekil 4.29. Set-2, InceptionResNetv2, Tüm Katmanların Eğitilmesi Yöntemi, Tam Veri Arttırma Veri Kümesi Eğitim Doğruluk ve Kayıp Sonucu

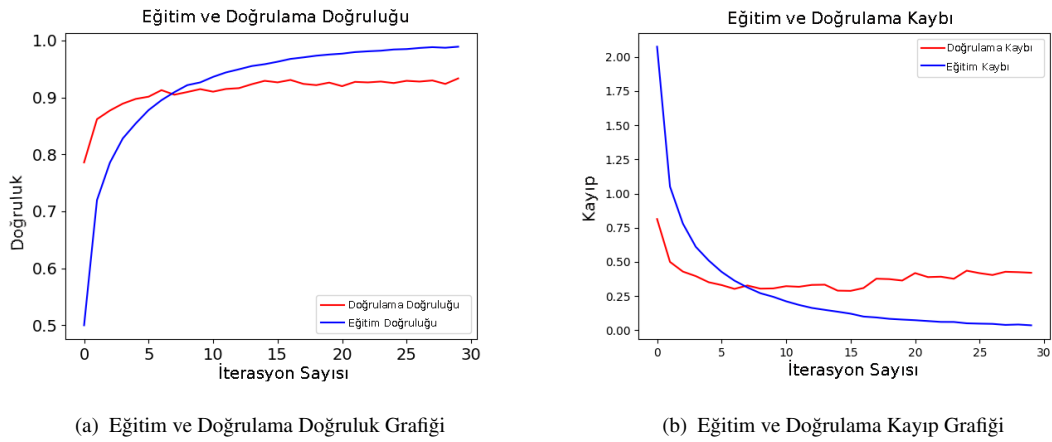
Şekil 4.30.'da tüm katmanların eğitilmesi yöntemi kullanılarak tam veri arttırma uygulanmış Set-2 (38 Sınıf) veri kümesinin DenseNet ağında eğitilerek elde edilen eğitim ve doğrulama doğruluğu ve kaybı grafikleri paylaşılmıştır. Toplamda 30 iterasyon ile gerçekleşen eğitimde

5. iterasyondan sonra doğrulama doğruluğu sabit kalmıştır. Doğrulama doğruluğu ve eğitim doğruluğu sırasıyla %93 ve %95 olarak elde edilmiştir.



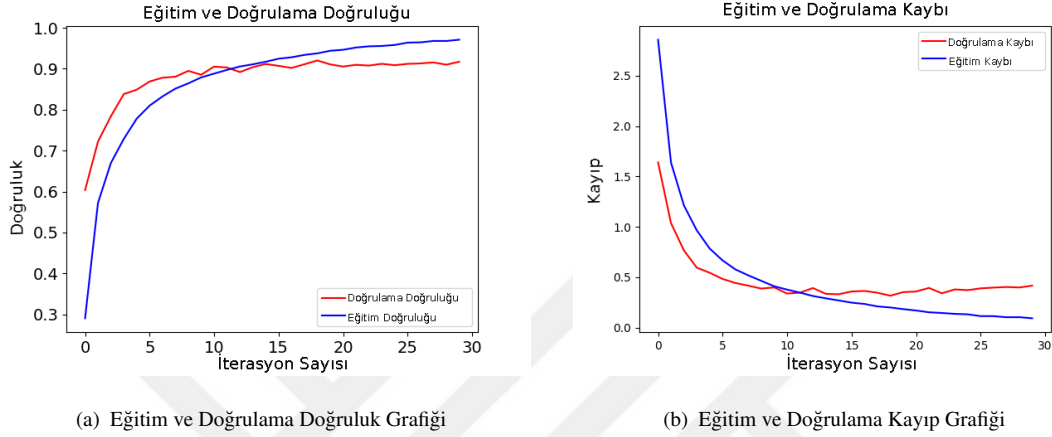
Şekil 4.30. Set-2, DenseNet, Tüm Katmanların Eğitilmesi Yöntemi, Tam Veri Arttırma Veri Kümesi Eğitim Doğruluk ve Kayıp Sonucu

Şekil 4.31.'de tüm katmanların eğitilmesi yöntemi kullanılarak tam veri arttırma uygulanmış Set-2 (38 Sınıf) veri kümesinin ResNet-50 ağında eğitilerek elde edilen eğitim ve doğrulama doğruluğu ve kaybı grafikleri paylaşılmıştır. Toplamda 30 iterasyon ile gerçekleşen eğitimde 7. iterasyondan sonra doğrulama doğruluğu sabit kalmıştır. Doğrulama doğruluğu ve eğitim doğruluğu sırasıyla %93 ve %99 olarak elde edilmiştir.



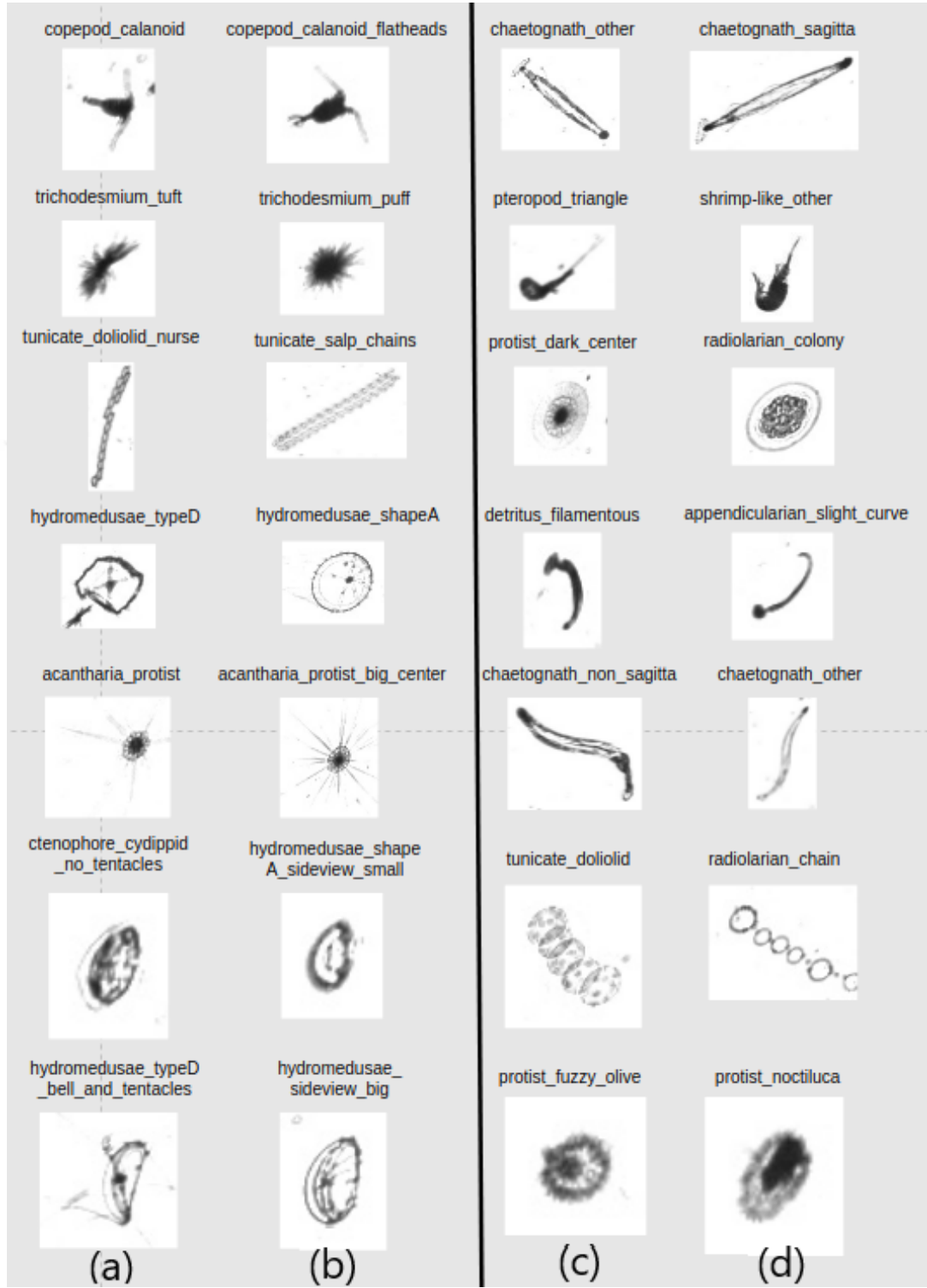
Şekil 4.31. Set-2, ResNet, Tüm Katmanların Eğitilmesi Yöntemi, Tam Veri Arttırma Veri Kümesi Eğitim Doğruluk ve Kayıp Sonucu

Şekil 4.32.'de tüm katmanların eğitilmesi yöntemi kullanılarak tam veri arttırma uygulanmış Set-2 (38 Sınıf) veri kümesinin VGG-16 ağında eğitilerek elde edilen eğitim ve doğrulama doğruluğu ve kaybı grafikleri paylaşılmıştır. Doğrulama doğruluğu ve eğitim doğruluğu sırasıyla %92 ve %94 olarak elde edilmiştir.

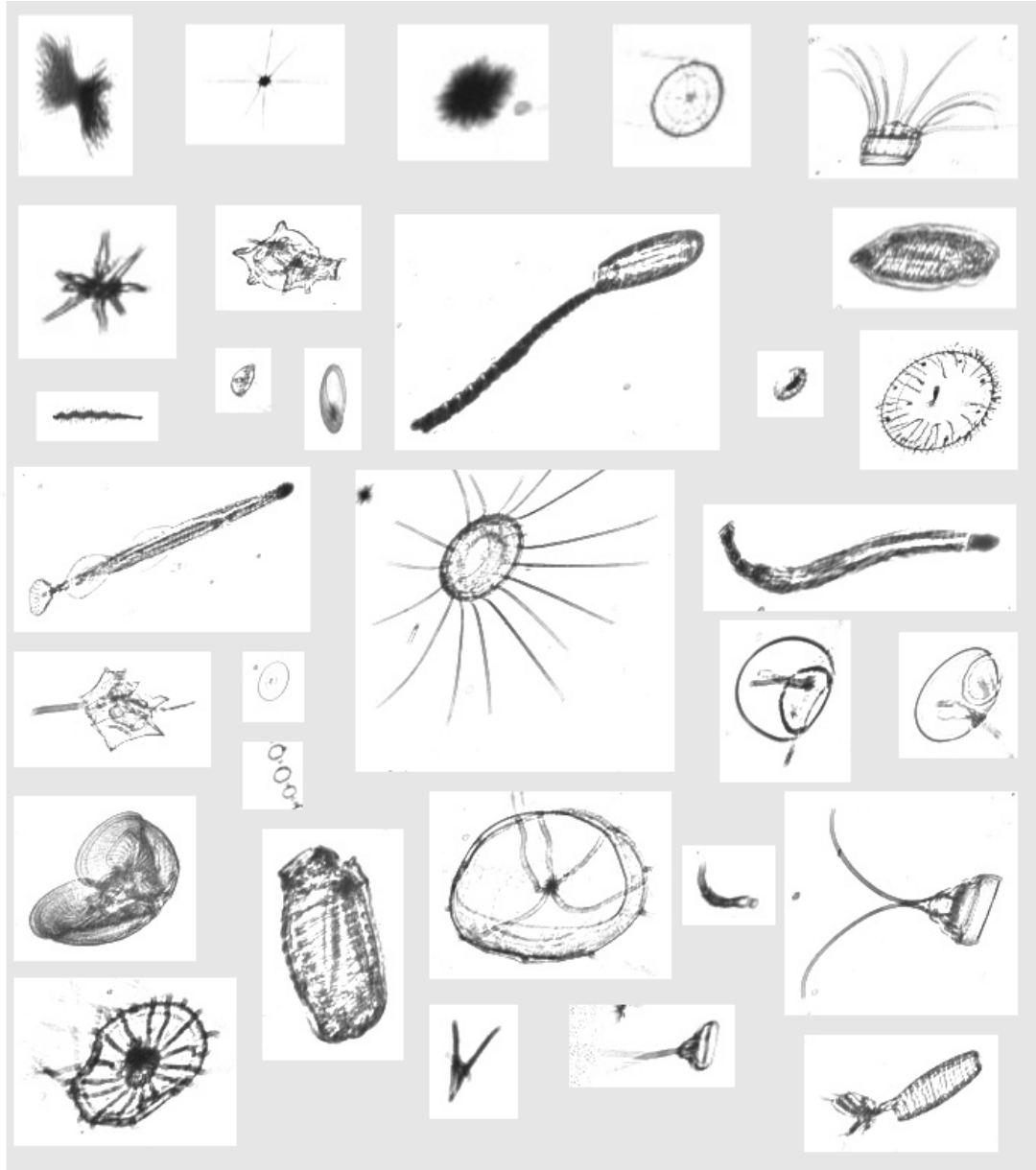


Şekil 4.32. Set-2, VGG-16, Tüm Katmanların Eğitilmesi Yöntemi, Tam Veri Arttırma Veri Kümesi Eğitim Doğruluk ve Kayıp Sonucu

Şekil 4.33.'te yanlış sınıflandırılan örneklerin sonuçları paylaşılmıştır. Şekile bakıldığında en çok karıştırılan örneklerin aynı sınıfın alt sınıfına sahip olduğu ve birbirine benzedikleri görülmektedir. Şekilde a ve c sütunları tahmin edilmesi beklenen gerçek sınıflara verilen örnekler iken, b ve d sütunları model tarafından tahmin edilen sınıf görüntüleridir. Şekil 4.34.'te test kümesinden doğru sınıflandırılan örnek görüntüler paylaşılmıştır.



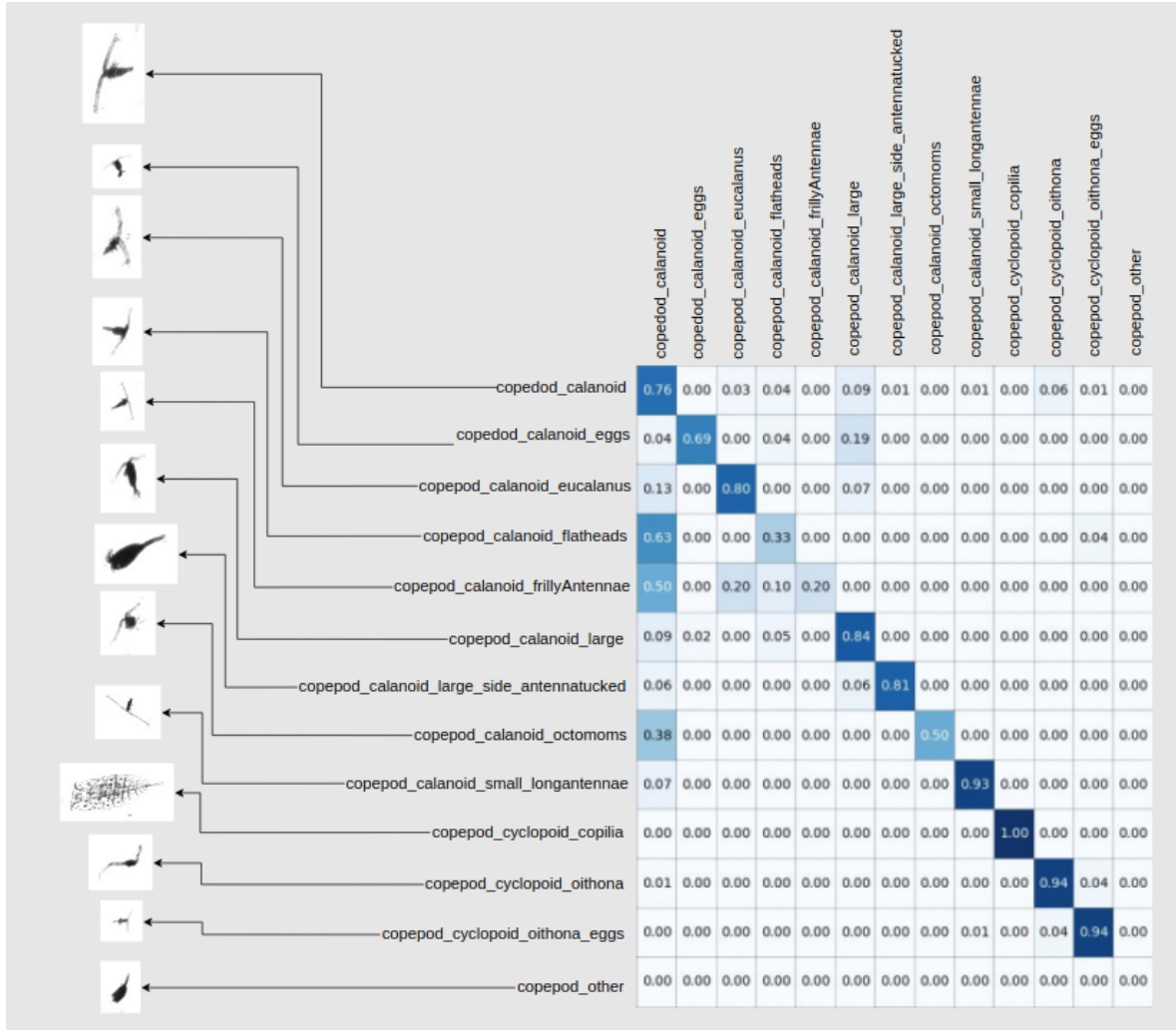
Şekil 4.33. Yanlış Sınıflandırılan Görüntü Örnekleri. (a) Gerçek Sınıf, (b) Tahmin Edilen Sınıf, (c) Gerçek Sınıf, (d) Tahmin Edilen Sınıf



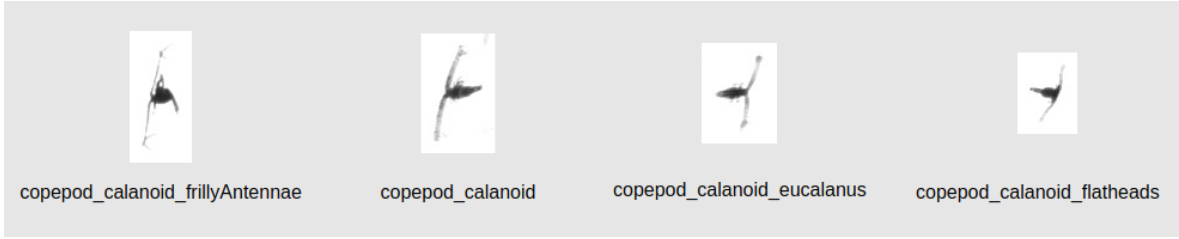
Şekil 4.34. Doğru Sınıflandırılan Görüntü Örnekleri

Şekil 4.35.'de en çok karıştırılan plankton sınıflarından biri olan copepod sınıflarının, Deney-3 kapsamında eğitilen DenseNet modelinin test veri kümesi ile testi sonucunda elde edilen hata matrisi paylaşılmıştır. Hata matrisinde değerler, 0 ile 1 aralığında aldığı değere göre açık maviden koyu maviye doğru renklendirilmiştir. Hata matrisinin sol tarafında her bir sınıftan, analiz edebilmek amacıyla örnek görüntü paylaşılmıştır. Her bir sınıfın görüntüsü incelendiğinde birbirine oldukça benzeyen sınıflar olduğu ve bu sınıfların model tarafından

kariřtırıldıđı hata matrisinde g zlemlenmiřtir. Hata matrisinde en  ok kariřtırılan sınıf olan copepod calanoid frillyAntennae ve model tarafından kariřtırılan diđer sınıfların g r nt leri řekil 4.36.'da paylařılmıřtır. řekil 4.36.'da g r ld đ   zere sınıflar k   k detaylar haricinde birbirine  ok benzemektedir. Bu durum, hata matrisinde copepod calanoid frillyAntennae sınıfının tahmin deđerinin d ř k bir deđer olan 0.20 olmasına sebebiyet vermiřtir. Hata matrisinde y ksek dođruluk oranına sahip lacivert rengine sahip sınıflara odaklanıldıđında bu sınıfların diđer sınıflardan ayırt edici bir řekilde farklı olduđu g zlemlenmiřtir.



řekil 4.35. Copepod Sınıfı Hata Matrisi



Şekil 4.36. Copepod Sınıfında Bulunan Karıştırılan Örnek Alt Gruplar

4.4.8. Deney-4

Deney-4 kapsamında ortalama topluluk yöntemi kullanılarak eğitilen modeller birleştirilmiştir. 118 sınıfa sahip Set-1 tam veri arttırma ile elde edilmiş veri kümesi Inceptionv3, InceptionResNetv2, DenseNet, ResNet-50 ve VGG-16 ağlarında eğitilmiş ve elde edilen modeller ortalama topluluk yöntemiyle birleştirmiştir.

38 sınıfa sahip Set-2 tam veri arttırma veri kümesi Inceptionv3, InceptionResNetv2, DenseNet, ResNet-50 ve VGG-16 ağlarında eğitilmiş ve elde edilen modeller ortalama topluluk yöntemiyle birleştirilmiştir.

4.4.9. Deney-4 Sonuçları

Deney-4 sonucunda elde edilen sonuçlar Tablo 4.11.'de paylaşılmıştır. Tabloda her bir topluluk modeli için kesinlik, geri çağırma, F1-Skoru metrikleri makro ortalama ve ağırlıklı ortalama yöntemleri kullanılarak ayrı ayrı hesaplanmıştır. Deney sonucunda Set-1 için %80 ağırlıklı F1-skoru, Set-2 için %93 ağırlıklı F1-skoru elde edilmiştir. Topluluk yöntemiyle elde edilen ağırlıklı F1-Skorunda, Set-1 topluluk modeliyle %2, Set-2 topluluk modeliyle %1'lik iyileşme gözlemlenmiştir. Tablo 4.12.'de ortalama topluluk yöntemiyle birleştirilen beş ağın model sonucu, test veri kümesi kullanılarak test edilmiş ve ilk-1 (top-1) ve ilk-5 (top-5) doğruluk değerleri elde edilmiştir. Set-1 veri kümesi için ilk-1 (top-1) doğruluk değeri %80.7 elde edilirken, ilk-5 (Top-5) topluluk değeri %98 olarak elde edilmiştir. Set-2 veri kümesinde ise %93.3 ve %99.2 ilk-1 (top-1) ve ilk-5 (top-5) doğruluk değerleri elde edilmiştir.

Tablo 4.11. Set-1 ve Set-2 Veri Kümelerinin Ortalama Modellerin Topluluğu Deney-4 Sonucu

	Makro Ortalama			Ağırlıklı Ortalama		
	Kesinlik	Geri Çağırma	F1-Skoru	Kesinlik	Geri Çağırma	F1-Skoru
Set-1 Topluluk Modeli	%66	%65	%65	%80	%81	%80
Set-2 Topluluk Modeli	%92	%90	%91	%93	%93	%93

Tablo 4.12. Set-1 ve Set-2 Veri Kümelerinin Ortalama Modellerin Topluluğu Deney-4 Sonucu

	Doğruluk	
	İlk-1 (Top-1)	İlk-5 (Top-5)
Set-1 Topluluk Modeli	%80.7	%98.0
Set-2 Topluluk Modeli	%93.3	%99.2

4.4.10. Deney-5

Deney-5 kapsamında, eğitim esnasında çapraz entropi kaybı yerine odak kaybı fonksiyonu kullanılmıştır. Tam veri arttırma ile elde edilen Set-1 (118 Sınıf) veri kümesi, tüm katmanların eğitilmesi yöntemi kullanılarak DenseNet ağında eğitilmiştir. Odak kaybı fonksiyonu dengesiz dağılıma sahip veri kümelerinde zor sınıflara odaklanarak, zor sınıflardaki başarıyı arttırmaktadır. Bu deney kapsamında dengesiz dağılıma sahip Set-1 (118 Sınıf) veri kümesi üzerinde, odak kaybının farklı hiperparametre değerlerinde eğitim gerçekleştirilmiştir ve odak kaybı fonksiyonunun başarıma etkisi gözlemlenmiştir. $\alpha = 0, 0.1, 0.25, 0.5, 0.75, 0.90, 1$ değerlerinde ve $\gamma = 0, 2$ değerlerinde deneyler gerçekleştirilmiştir.

4.4.11. Deney-5 Sonuçları

α parametresinin 1, γ parametresinin 0 olduğu durumda odak kaybı ile çapraz entropi kaybı matematiksel olarak eşit olmaktadır. Denklem 17’de odak kaybının matematiksel ifadesi verilmiştir. α parametresinin 1, γ parametresinin 0 olduğu deneye bakıldığında Tablo 4.10.’da çapraz entropi ile elde edilen sonuçların birebir aynısı elde edilmemiştir. Odak kaybıyla ile yapılan eğitim sonucunda, eğitim ve doğrulama doğruluğu %1 artış göstermiş ve eğitim kaybında %4 doğrulama kaybında %2 azalma gözlemlenmiştir. Makro ve ağırlıklı ortalama ile bulunan değerler, çapraz entropi ile makro ağırlıklı kesinlik değerindeki %2’lik düşüş haricinde birebir aynı olarak elde edilmiştir. Matematiksel olarak iki kayıp fonksiyonunun eşit

kabul edildiği α parametresinin 1, γ parametresinin 0 olduğu durumda eğitimde iyileşme gerçekleşmiştir. Test verisi ile elde edilen sonuçlarda kesinlik metriği haricinde bir değişiklik gözlemlenmemiştir. Deney sonuçları Tablo 4.13. ve Tablo 4.14.'te paylaşılmıştır. α 'nın farklı değerleriyle gerçekleşen deney sonuçlarında α 'nın 0.1 ve 0.90 arasındaki değerleri için kayıp ile doğru orantılı olduğu gözlemlenmiştir.

Tablo 4.13. Set-1 Veri Kümesinin DenseNet ağı ve Farklı Odak Kaybı Parametreleri ile Eğitilen Modellerinin Sonuçları

		Makro Ortalama			Ağırlıklı Ortalama		
α	γ	Kesinlik	Geri Çağırma	F1-Skoru	Kesinlik	Geri Çağırma	F1-Skoru
$\alpha = 0$	$\gamma = 2$	%63	%60	%60	%76	%77	%76
$\alpha = 0.1$	$\gamma = 2$	%55	%50	%50	%73	%75	%73
$\alpha = 0.25$	$\gamma = 2$	%59	%56	%55	%75	%76	%75
$\alpha = 0.5$	$\gamma = 2$	%64	%59	%59	%77	%77	%76
$\alpha = 0.75$	$\gamma = 2$	%64	%61	%62	%77	%78	%77
$\alpha = 0.90$	$\gamma = 2$	%63	%61	%61	%77	%78	%77
$\alpha = 1$	$\gamma = 0$	%64	%61	%61	%78	%79	%78

Tablo 4.14. Set-1 Veri Kümesinin DenseNet ağı ve Farklı Odak Kaybı Parametreleri ile Eğitilen Modellerinin Sonuçları

		Eğitim		Doğrulama		Doğruluk	
α	γ	Doğruluk	Kayıp	Doğruluk	Kayıp	İlk-1 (Top-1)	İlk-5 (Top-5)
$\alpha = 0$	$\gamma = 2$	0.75	0.64	0.79	0.59	%77	%96
$\alpha = 0.1$	$\gamma = 2$	0.69	0.12	0.77	0.10	%75	%95
$\alpha = 0.25$	$\gamma = 2$	0.74	0.21	0.78	0.18	%76	%96
$\alpha = 0.5$	$\gamma = 2$	0.75	0.33	0.79	0.28	%77	%97
$\alpha = 0.75$	$\gamma = 2$	0.76	0.41	0.79	0.39	%78	%97
$\alpha = 0.90$	$\gamma = 2$	0.76	0.45	0.79	0.43	%78	%97
$\alpha = 1$	$\gamma = 0$	0.80	0.64	0.80	0.72	%79	%97

4.4.12. Deney-6

Bu deney kapsamında, tam veri artırma uygulanmış Set-1 (118 Sınıf) veri kümesi ve tüm katmanların eğitilmesi yöntemiyle eğitilen DenseNet modelinin test veri kümesi üzerinde elde edilen hata matrisleri analiz edilmiş ve gruplama ile ilgili bir yaklaşım önerilmiştir. Ayrıca veri kümesi, 33 plankton grubu şeklinde gruplandırılarak eğitilmiş ve elde edilen model test veri kümesinde test edilerek hata matrisi elde edilmiştir.

118 sınıf ile eğitilen modelin hata marisinden, grup yapısı hata matrisi elde etmek amacıyla uygulanan adımlar şu şekildedir.

- 118 sınıfa sahip Set-1 veri kümesinin DenseNet modeli, test veri kümesinde test edilmiştir ve elde edilen hata matrisi 118x118 boyutunda elde edilmiştir.
- Elde edilen 118x118 hata matrisinin sütun kısmında yer alan sınıfları grup bazında toplanmış bu işlem sonucunda 118x33 matris elde edilmiştir.
- Elde edilen 118x33 boyutundaki hata matrisinin her bir satırı ilgili sınıfın veri sayısı ile çarpılmış ve sonrasında satırlar grup bazında toplanmıştır. Bu işlem sonucunda 33x33 hata matrisi elde edilmiştir.
- 33x33 hata matrisindeki her bir satır ilgili grubun içerdiği toplam veri sayısına bölünerek o grup ile ilgili tahmin elde edilmiştir.

118x118 hata matrisinden yukarıdaki işlemler uygulanarak elde edilen 33x33 hata matrisi ile, grup bazında 33 sınıf grubu ile eğitilen modelin hata matrisi karşılaştırılmış ve deney ile ilgili tüm sonuçlar Bölüm 4.4.13.'de paylaşılmıştır.

4.4.13. Deney-6 Sonuçları

Şekil 4.37.'de ve Şekil 4.38.'de 118x118 hata matrisindeki sütunlarda yer alan sınıfların grup bazında toplanarak gruplandığı ve sonucunda 118x33 boyutunda hata matrisinin elde edildiği hata matrisi görseli iki kısım olarak paylaşılmıştır. İlk kısım, 118x33 boyutundaki matrisin ilk 60 sınıfını, ikinci kısım ise son 58 sınıfını göstermektedir. 118x33 boyutuna sahip matrisin her bir satırının sınıf sayısı ile çarpıldığı ve aynı sütun sınıflarının toplanarak gruplandırıldığı gibi satırlarının da toplanarak gruplandırıldığı 33x33 hata matrisi Şekil 4.39.'da

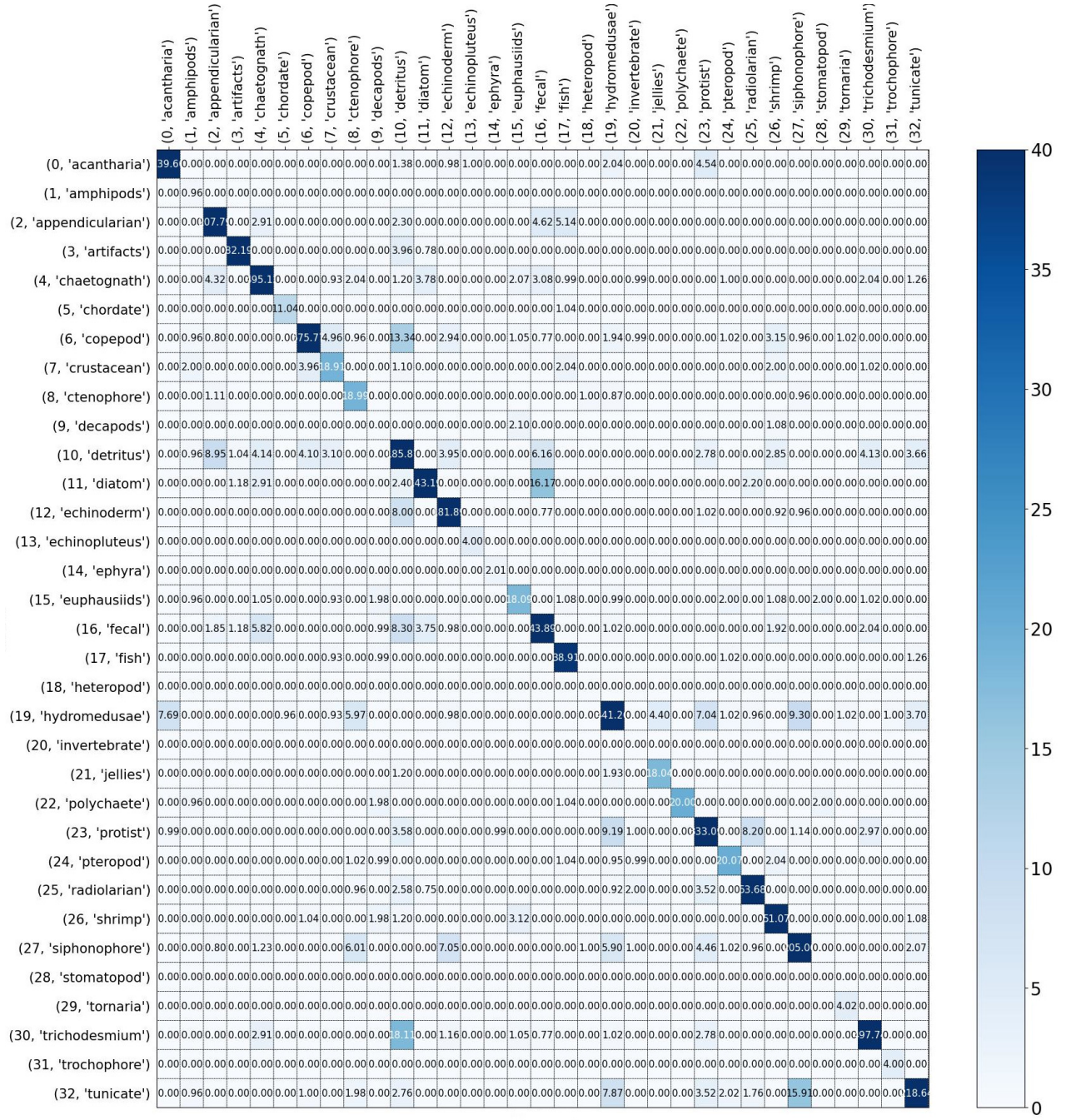
paylaşılmıştır. Elde edilen 33x33 matrisin her bir değerinin olasılık değerini bulmak amacıyla Şekil 4.39.'da yer alan her bir değer grupta yer alan toplam veri sayısına bölünmüştür ve bu işlem sonucunda Şekil 4.40.'da yer alan hata matrisi elde edilmiştir.

Şekil 4.40. ve Şekil 4.41. hata matrisleri karşılaştırıldığında, Şekil 4.40.'da yer alan ve 118 sınıfın gruplanması ile elde edilen hata matrisinin Şekil 4.41.'deki hata matrisine göre doğruluk oranlarının daha yüksek olduğu ve sınıflar arası karıştırma oranının daha düşük olduğu gözlemlenmiştir. Örneğin tornaria plankton sınıfı Şekil 4.41.'de 0.50 tahmin doğruluğuna sahipken, Şekil 4.40.'da 0.67 doğruluk tahminine sahiptir. Tahmin doğruluğu ortalama seviyede olan ve grup içi karışmaya meyil gösteren sınıflar için önerilen yöntem başarılı olmuştur. Önerilen yöntemin dezavantajlarından bir tanesi tahmin doğruluğu iyi seviyede olmayan, az veriye sahip ve karıştırılan sınıfları tamamen yok saymasıdır (Tahmin değeri 0 olmaktadır). Örneğin stomatopod sınıfı, Şekil 4.41.'de yer alan hata matrisinde 0.25 tahmin oranına sahipken bu oran Şekil 4.40.'da sıfırlanmıştır.

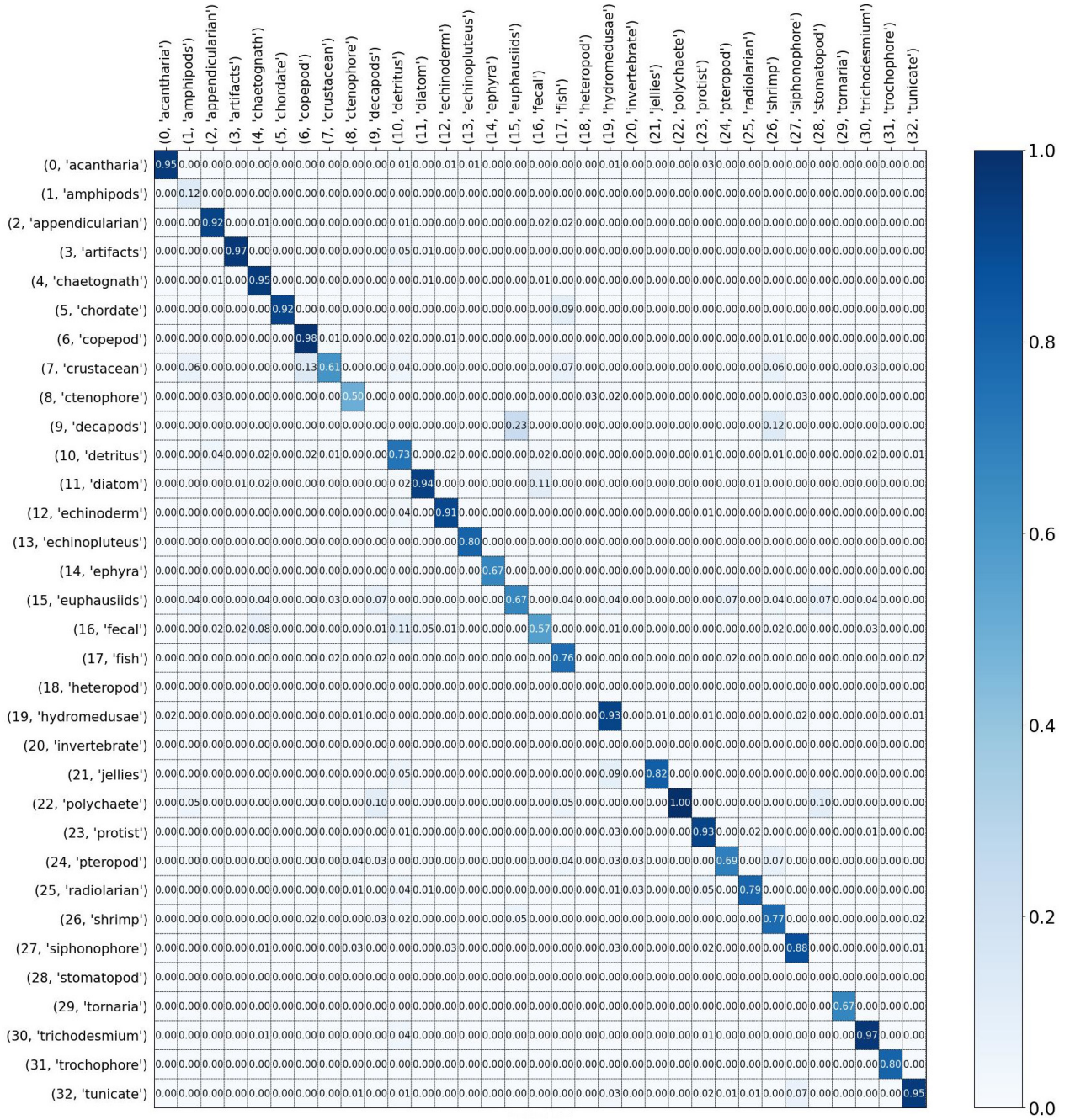
[illegible]

[illegible]

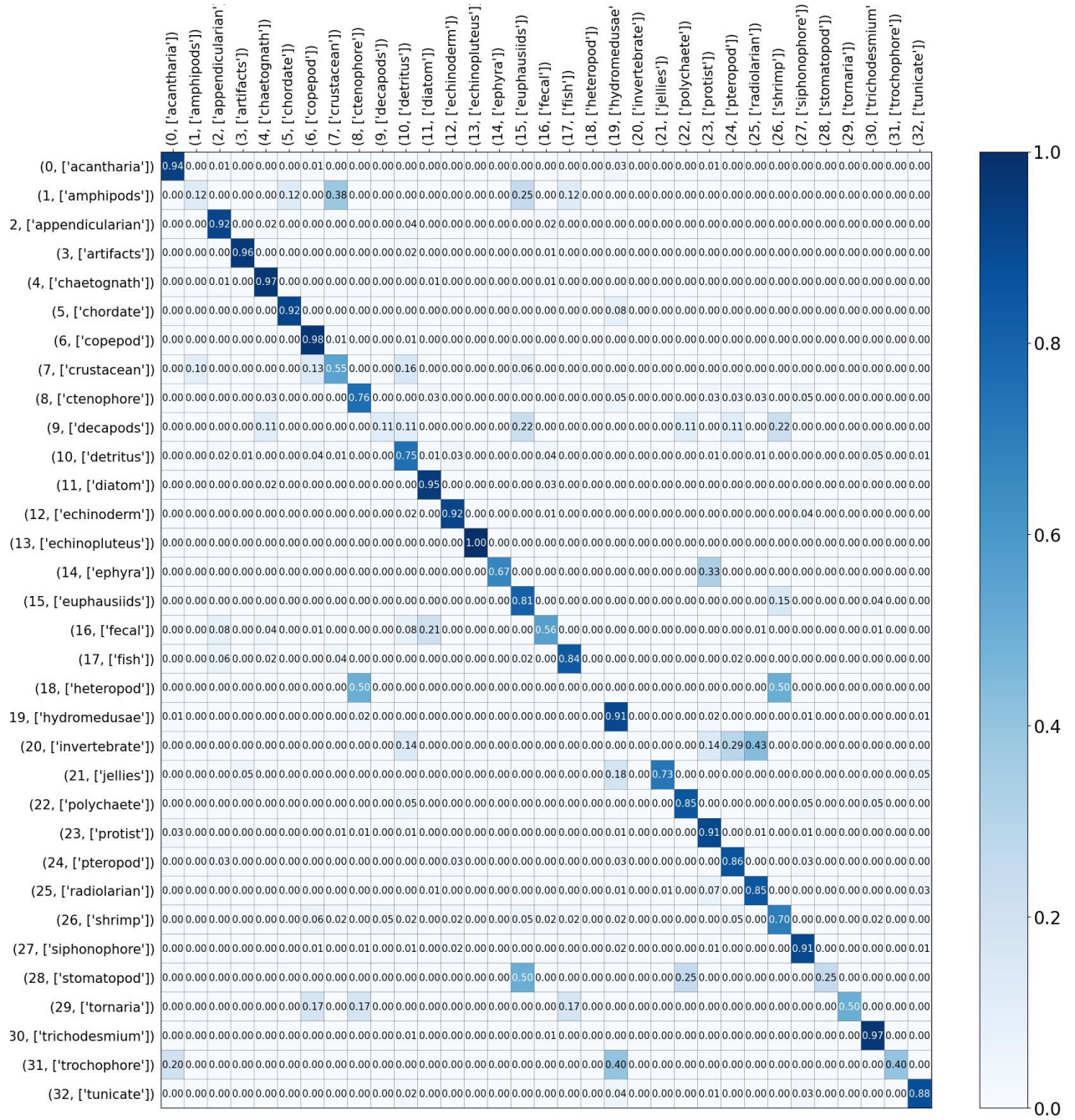
Sekil 4.38. Sütunların Grup Bazında Toplanarak Gruplandığı 118x33 Hata Matrisi - Kısım 2



Şekil 4.39. Her Bir Sınıfın Olasılık Değerinin Sınıf Sayısı ile Çarpılarak, Satır Bazında Gruplandırılıp Toplanarak Elde Edilen 33x33 Hata Matrisi



Şekil 4.40. Her Bir Sınıf Grubunun Sahip Olduğu Toplam Veri Sayısına Bölünerek Elde Edilen 33x33 Hata Matrisi



Şekil 4.41. 33 Plankton Grubu ile Eğitilen Model Sonucunda Elde Edilen Hata Matrisi

5. SONUÇLAR

Bu çalışmanın ana amacı, derin evrişimli sinir ağı tabanlı plankton sınıflandırmasının performansını raporlamak için farklı yöntemlerin ve evrişimli sinir ağı modellerinin geniş bir deneysel değerlendirmesini yapmaktır. Bu amaçla gerçekleştirilen katkılar şu şekilde özetlenebilir: Dengesiz dağılımlı bir veri kümesi olan Kaggle-plankton veri kümesi kullanıldığı için veri kümesi veri arttırma yöntemi ve ön işleme yöntemleri uygulanarak iyileştirilmeye çalışılmıştır. Önceden ImageNet veri kümesi [69] ile eğitilen evrişimli sinir ağı modellerini; aktarım öğrenmesi, ince ayar, tüm katmanların eğitilmesi yöntemleri olmak üzere toplamda üç yöntem kullanılarak veri kümeleri ile ayrı ayrı eğitilmiş ve veri arttırma ve ön işleme işlemlerinin eğitime katkısı gözlemlenmiştir. Sınıflandırma modelinin başarımını arttırmak amacıyla ortalama topluluk öğrenimi yöntemi uygulanmıştır. Farklı kayıp yöntemleri olan çapraz entropi ve odak kaybı kayıp yöntemlerinin plankton eğitime etkisi gözlemlenmiştir. Benzer sınıfların gruplanması tekniği önerilerek ve gruplanarak eğitilen model ile hata matrisleri karşılaştırılarak elde edilen sonuçlar analiz edilmiştir.

Plankton sınıflandırmasında Kaggle plankton veri kümesinin iki farklı sürümü kullanılmıştır. Kullanılan ilk sürümde 121 sınıfa sahip Kaggle plankton veri kümesinden, bilim insanları tarafından etiketlenemeyen ve bilinmeyen olarak adlandırılan üç sınıf çıkarılmış ve 118 sınıfa sahip, tez kapsamında Set-1 olarak isimlendirilen veri kümesi oluşturulmuştur. Bilinmeyen olarak adlandırılan üç sınıfın veri kümesinden çıkarılmasının sebebi hangi sınıfa ait olacağı bilinemediği ve bu sebeple doğru etiketlenemediği için eğitim başarısını negatif yönde etkileyebileceğinin düşünülmesidir. İkinci sürüm olarak [18] çalışmasında ve benzer Kaggle plankton veri kümesi çalışmalarında kullanıldığı gibi plankton sınıflarında minimum 100 görüntüye sahip 38 sınıf seçilerek 14.374 görüntüden oluşan Kaggle plankton veri kümesinin bir alt kümesi kullanılmıştır. Seçilen sınıfların her birinde min 108 ve maksimum 1979 görüntü bulunmaktadır. Set-2 olarak isimlendirilen veri kümesi diğer çalışmalarla yöntemin karşılaştırılması ve veri kümesinin farklı versiyonlarını değerlendirebilmek amacıyla kullanılmıştır. Tez kapsamında plankton sınıflandırması başarımını arttırmak amacıyla uygulanan yöntemler Set-1 veri kümesi üzerinde gerçekleştirilmiştir. Uygulanan yöntemlerden başarılı olduğuna karar verilen eğitim yöntemi ve veri arttırma yöntemi Set-2 veri kümesine ayrıca uygulanmıştır.

Deney-1 kapsamında Set-1 veri kümesi üç farklı eğitim yöntemi ve her bir yöntem için üç farklı veri kümesi seçeneğiyle eğitilmiştir. Set-1 veri kümesi sınıf dengesizliğine sahip bir

veri kümesi olması sebebiyle bazı sınıflar fazla veriye sahipken bazı sınıflar yetersiz sayıda veri içermektedir. Bu durum orijinal veri kümesi ile eğitildiğinde yeterli sayıda veriye sahip sınıflarda fazla başarımlar elde edilmesini sağlarken, yetersiz veriye sahip sınıflarda düşük başarımlar elde edilmesine neden olmaktadır. Ayrıca yetersiz sayıdaki veri eğitimde aşırı uyum durumuyla karşılaşılmasına sebep olmaktadır. Bu durumları önlemek amacıyla Set-1 veri kümesine veri arttırım yöntemleri uygulanmıştır. Uygulanan iki farklı veri arttırma yönteminden biri olan kısmi veri arttırma yönteminde, sınıflarda belirlenen değerin (60) altında veriye sahip olan sınıflara veri arttırma yöntemi uygulanmıştır. Belirli değerin üzerinde veriye sahip sınıflara veri arttırımı uygulanmamıştır. Uygulanan ikinci veri arttırma yöntemi olan tam veri arttırma yönteminde, sınıfların sahip olduğu veri sayısından bağımsız olarak tüm sınıflara veri arttırımı uygulanmıştır. Yatay, dikey olarak tersine çevirme, farklı açılarda uygulanan rotasyon, farklı katsayılarla uygulanan odaklanma Deney-1 kapsamında uygulanan veri arttırma yöntemleridir. Set-1 (118 Sınıf) veri kümesinde plankton sınıflandırması yapmak amacıyla gerçekleştirilen ilk eğitimlerde aktarım öğrenmesi yöntemi kullanılarak eğitim gerçekleştirilmiştir. Bu yöntem ile elde edilen başarımlar, %70 civarından daha yükseğe çıkamamıştır. Aktarım öğrenmesinde, ağların büyük veri kümelerinde önceden eğitilmiş ağırlıkları kullanılmış, sadece son katman değiştirilmiş ve plankton sınıflandırmasında kullanılan veri kümesinin sınıf sayısına uygun olacak hale getirilmiştir. Uygulanan ikinci eğitim yöntemi olan ince ayar yönteminde, önceden eğitilen ağın katman sayısının %30'u hesaplanarak ilgili katmanların eğitilebilir hale getirilmesi sağlanmıştır. Evrimsel sinir ağlarında ilk katmanlar veriye dair daha genel özellikleri öğrenirken, son katmanlar veriyle ilgili daha spesifik özellikleri öğrenmektedir. Bundan dolayı son katmanlardan geriye giderek her bir ağ için katmanların %70'inden itibaren eğitilebilir hale getirilmesinin sonucu nasıl etkileyeceği gözlemlenmek istenmiştir. Deney-1 kapsamında gerçekleştirilen ince ayar yöntemi sonucu, aktarım öğrenmesi sonucuna göre orijinal ve kısmi veri arttırma uygulanmış veri kümelerinde yaklaşık olarak %2, tam veri arttırma uygulanmış veri kümesinde %4 daha iyi başarımlar göstermiştir. Katmanların eğitilebilir hale gelmesinin başarıma olan etkisini gözlemlemek amacıyla tüm katmanların eğitilmesi yöntemi kullanılmış, tüm katmanlar eğitilebilir hale getirilmiş ve ince ayar yöntemi ile karşılaştırıldığında orijinal veri kümesinde %1, kısmi veri arttırma ve tam veri arttırma uygulanmış veri kümelerinde %3 başarımlar artışı gerçekleşmiştir. Deney-1 kapsamında elde edilen veri kümeleri ve yöntemler kullanılarak elde edilen eğitim sonuçları karşılaştırıldığında, en iyi sonuç aktarım öğrenmesi yöntemi ve tam veri arttırma uygulanarak elde edilen veri kümesi eğitimi sonucunda elde edilmiştir. Aktarım öğrenmesi yönteminin diğer yöntemlere göre başarısız olmasındaki en büyük sebebin ağların önceden

eğitilen veri kümesi (ImageNet) ile plankton veri kümesinin benzerlik oranının çok düşük olması sebebiyle olduğu düşünülmektedir.

Deney-2 kapsamında Deney-1 sonucunda en iyi sonucu veren Set-1 (118 Sınıf) tam veri artırma uygulanarak elde edilmiş veri kümesi kullanılmıştır. Plankton görüntülerinin çok farklı boyutlarda olması sebebiyle ve evrişimli sinir ağlarının sabit boyut kabul etmesi nedeniyle görüntülerde yaşanan bilgi kaybını önlemek için kare kırpma ve boşluk ekleme ön işleme yöntemleri kullanılmıştır. Boşluk ekleme yönteminde, ilgili plankton görüntüsünün evrişimli sinir ağının kabul ettiği görüntü boyutuna getirilmesi için etrafına gereken boyut kadar beyaz pikseller eklenmiştir. Kare kırpma yönteminde, evrişimli sinir ağına görüntüler verilmeden önce görüntülere ortalanarak kesme işlemi gerçekleştirilmiş ve elde edilen kare plankton görüntüsü, bikübik interpolasyonu kullanılarak 224x224 boyutu olan evrişimli sinir ağının girdi boyutuna göre yeniden boyutlandırılmıştır. Yöntemlerin ilgili ön işleme yöntemlerine olan katkısını görmek amacıyla ayrı ayrı uygulanmıştır. Elde edilen sonuçlarda, uygulanan ön işleme yöntemlerinin plankton sınıflandırmasının başarımına negatif etki ettiği gözlemlenmiştir. Ön işleme uygulamalarıyla farklı yöntemler ile elde edilen sonuçlarda %2-3 arasında kayıplar yaşanmıştır. Deney-2 sonucunda ön işleme yöntemlerinin sonuçları negatif etkilediğine karar verilerek en iyi yöntem olarak Yöntem-3 ve ön işleme uygulanmamış (orijinal) tam veri artırma ile elde edilmiş veri kümesi seçilerek sonraki deneyler bu yöntem ve veri kümesi ile gerçekleştirilmiştir.

Deney-3 kapsamında tüm katmanların eğitilmesi yöntemi ve tam veri artırma yöntemleri, Set-1 (118 Sınıf) ve Set-2 (38 Sınıf) veri kümesi ile beş farklı ağda eğitilmiştir. Inceptionv3, InceptionResNetv2, DenseNet, ResNet-50 ve VGG-16 ağlarında eğitim gerçekleştirilmiştir. Set-2 veri kümesi daha dengeli sınıflara sahip olduğundan, Set-1 veri kümesine göre daha yüksek başarımler elde edilmiştir. En iyi sonucu veren model, DenseNet modeli olarak gözlemlenmiştir. Set-1 veri kümesinde %78, Set-2 veri kümesinde %92 ağırlıklı F1-Skoru elde edilmiştir. Ayrıca Set-1 veri kümesinde %97, Set-2 veri kümesinde %99.1 ilk-5 (Top-5) skoru elde edilmiştir. Deney-4 kapsamında tüm ağlardan elde edilen başarımlar ortalama topluluk yöntemiyle birleştirilmiş ve sonuç Set1 veri kümesinde %80 ve Set-2 veri kümesinde %93 ağırlıklı F1-Skoru elde edilmiştir. Bu yöntem ile ağların başarımları birleştirilerek Set-1 topluluk modeliyle %2, Set-2 topluluk modeliyle %1 daha yüksek başarımlar elde edilmiştir. Ayrıca ilk-5 doğruluk oranı, Set-1 veri kümesinde %98, Set-2 veri kümesinde %99.2 olarak elde edilmiştir. Deney-5 kapsamında odak kaybı kayıp fonksiyonunun farklı hiperparametreleri ile eğitimler gerçekleştirilerek sonuçlar elde edilmiştir. Deney-5 sonucunda test kümesinden

elde edilen başarımda artış sağlanmamıştır, eğitim ve doğrulama doğruluk değerlerinde artış ve kayıp değerlerinde azalma gözlemlenmiştir. Deney-6 kapsamında, 118 sınıfa ait DenseNet ağı üzerinde Deney-3 kapsamında Set-1 veri kümesi ile gerçekleştirilen deney sonucu elde edilen modelin hata matrisi hesaplanmıştır. Elde edilen 118x118 boyutundaki hata matrisi, aynı isim altında toplanan farklı gruplar sütun bazında toplanarak 118x33 matris elde edilmiş, sonrasında her bir sınıf içerdiği veri sayısı miktarıyla çarpılmıştır. Satır bazında da aynı isim altında toplanan farklı gruplar sütun bazında toplanarak 33x33 grup matrisi elde edilmiştir. Elde edilen 0-40 aralığındaki hata matrisi her bir ana sınıfın içerdiği toplam plankton sayısına bölünmüş ve 33 grup için 0-1 aralığında olasılık değerleri elde edilmiştir. Elde edilen bu hata matrisi sonucuyla, 33 grup olarak eğitimi gerçekleştirilen hata matrisi karşılaştırılmış ve 118 sınıftan 33 grup şekline dönüştürülerek elde edilen hata matrisinin benzer grup altındaki sınıfları daha az birbirine karıştırdığı ve daha iyi bir çözüm yolu olduğu sonucuna varılmıştır.

Tez kapsamında yapılan katkılar şu şekilde özetlenebilir: Dengesiz dağılımlı bir veri kümesi kullanıldığı için veri kümesi iyileştirilmeye çalışılmıştır. Veri arttırma yöntemi ve ön işleme yöntemleri uygulanmıştır. Önceden eğitilen evrişimli sinir ağı modelleri, aktarım öğrenmesi, ince ayar, tüm katmanların eğitilmesi yöntemleri olmak üzere toplamda üç yöntem kullanılarak ayrı ayrı eğitilmiş ve veri arttırma ve ön işleme işlemlerinin eğitime katkısı gözlemlenmiştir. Sınıflandırma modelinin başarımını arttırmak amacıyla ortalama topluluk öğrenimi yöntemi uygulanmıştır. Farklı kayıp yöntemlerinin plankton eğitimine etkisi gözlemlenmiştir. Benzer sınıfların gruplanması tekniği önerilmiş ve gruplanarak eğitilen model ile hata matrisleri karşılaştırılarak elde edilen sonuç analiz edilmiştir.

5.1. Gelecek Çalışmalar

Plankton görüntülerinin farklı boyutlarda olması sebebiyle ve evrişimli sinir ağlarının sabit boyutlarda girdi görüntüsü kabul etmesinden dolayı plankton görüntüleri yeniden boyutlandırılırken görüntüde bilgi kaybı olmaktadır. Bu bilgi kaybının giderilmesi için plankton görüntülerine uygulanacak ön işleme yöntemi olarak kaybı önleyen farklı yöntemler uygulanarak sınıflandırma başarımına olan etkisi gözlemlenebilir. Bu yöntemlerden biri olduğu düşünülen süper çözünürlük (super resolution) yöntemlerinin kullanımı daha yüksek sınıflandırma başarımına katkı sağlayacaktır. İkinci olarak evrişimli sinir ağının girdi görüntüsünü farklı boyutlarda almasını sağlayan uzamsal piramid katmanı (spatial pyramid pooling)

mimariye dahil edilerek evriřimli sinir ađının grnty farklı boyutlarda girdi olarak alması sađlanabilir. Bu yntemle grntlerde daha dřk bilgi kaybı olduđundan sınıflandırmanın bařarımının artacađı dřnlmektedir.



EKLER

EK1: SÖZLÜK DİZİNİ

Aktarım Öğrenmesi	: Transfer Learning
Aşırı Uyum	: Overfitting
Atık	: Residual
Az Vuruşlu Öğrenme	: Few-shot Learning
Bilgisayarlı Görü	: Computer Vision
Boşluk Ekleme	: Padding
Çok Katmanlı Algılayıcı	: MultiLayer Perceptron
Çoklu Akış	: Multistream
Derin Öğrenme	: Deep Learning
Destek Vektör Makinesi	: Support Vector Machine
Doğrulama	: Validation
Doğru Negatif	: True Negatif
Doğru Pozitif	: True Positive
Dondurulmuş Katman	: Freezing Layer
El Yapımı Tanımlayıcılar	: Handcrafted Descriptors
Eşik Değeri	: Threshold
Evrişim	: Convolution
Evrişimsel Sinir Ağı	: Convolutional Neural Network
Geri Çağırma	: Recall
Gradyan İnişi	: Gradient Descent
Yerinde Plankton Görüntüleme Sistemi	: In-situ plankton imaging system
Yerinde Olmayan Plankton Görüntüleme Sistemi	: Ex-situ plankton imaging system
İnce Ayar	: Fine Tuning
Kaybolan Gradyan	: Vanishing Gradient
Kayıp Fonksiyonu	: Loss Function
Kesinlilik	: Precision
K-En Yakın Komşu	: K-Nearest Neighborhood
Küme Boyutu	: Batch Size
Makine Öğrenmesi	: Machine Learning
Maksimum Ortalama	: Maximum Pooling
Ortalama Topluluk Öğrenimi	: Average Ensemble Learning

Öğrenme Oranı	: Learning Rate
Ön İşleme	: Pre-Processing
Özellik	: Feature
Özellik Mühendisliği Tabanlı	: Feature-based Engineering
Pekiştirme	: Reinforcement
Rassal Orman	: Random Forest
Seyreltme	: Dropout
Sızıntı	: Leaky
Sinir Ağları	: Neural Networks
Ulusal Veri Bilimi Kasesi	: National Datasience Bowl
Uzamsal Piramid Ağı	: Spatial Pyramid Network
Veri Arttırma	: Data Augmentation
Veri Kümesi	: Dataset
Tam Bağlantılı Katman	: Fully Conncted Layer
Topluluk	: Ensemble
Torbalama Karar Ağacı	: Bagged Decision Tree
Yanlılık	: Bias
Yanlış Negatif	: False Negative
Yanlış Pozitif	: False Positive
Yerel İkili Örüntü	: Local Binary Pattern
Yoğun Katman	: Dense Layer

KAYNAKLAR

- [1] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, **1998**.
- [2] Keiron O’Shea and Ryan Nash. An introduction to convolutional neural networks. *arXiv preprint arXiv:1511.08458*, **2015**.
- [3] Jaroslav Bureš. *Classification of varying-size plankton images with convolutional neural network*. Ph.D. thesis, Brno University of Technology, Czech Republic, **2020**.
- [4] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2818–2826. **2016**.
- [5] Andrew S Brierley. Plankton. *Current Biology*, 27(11):R478–R483, **2017**.
- [6] Emilio Marañón, J Steele, A Thorpe, and K Turekian. Phytoplankton size structure. *Elements of physical oceanography: A derivative of the encyclopedia of ocean sciences*, 85, **2009**.
- [7] Pushpendra Kumar, Vedat Suat Erturk, Ramashis Banerjee, Mehmet Yavuz, and V Govindaraj. Fractional modeling of plankton-oxygen dynamics under climate change by the application of a recent numerical algorithm. *Physica Scripta*, 96(12):124044, **2021**.
- [8] Feng Zhao, Feng Lin, and Hock Soon Seah. Binary sipper plankton image classification using random subspace. *Neurocomputing*, 73(10-12):1853–1860, **2010**.
- [9] Kilian B Sullivan-Silva and Michael J Forbes. Behavioral study of zooplankton response to high-frequency acoustics. *The Journal of the Acoustical Society of America*, 92(4):2423–2423, **1992**.
- [10] Song J. Yu, L. and H. Tang. In situ capturing of plankton images. *Ocean Technology of China*, 25:47–49, **2004**.

- [11] Robert J Olson and Heidi M Sosik. A submersible imaging-in-flow instrument to analyze nano-and microplankton: Imaging flowcytobot. *Limnology and Oceanography: Methods*, 5(6):195–203, **2007**.
- [12] Michael E Sieracki, Mark Benfield, Allen Hanson, Cabell Davis, Cynthia H Pilskaln, David Checkley, Heidi M Sosik, Carin Ashjian, Phil Culverhouse, Robert Cowen, et al. Optical plankton imaging and analysis systems for ocean observation. *Proc. Ocean Obs*, 9:21–25, **2010**.
- [13] Philippe Grosjean, Marc Picheral, Caroline Warembourg, and Gabriel Gorsky. Enumeration, measurement, and identification of net zooplankton samples using the zooscan digital imaging system. *ICES Journal of Marine Science*, 61(4):518–525, **2004**.
- [14] HP Jeffries, MS Berman, AD Poularikas, C Katsinis, I Melas, K Sherman, and L Bivins. Automated sizing, counting and identification of zooplankton by pattern recognition. *Marine Biology*, 78(3):329–334, **1984**.
- [15] Xiaou Tang, W Kenneth Stewart, He Huang, Scott M Gallager, Cabell S Davis, Luc Vincent, and Marty Marra. Automatic plankton image recognition. *Artificial Intelligence Review*, 12(1):177–199, **1998**.
- [16] Gaby Gorsky, Mark D Ohman, Marc Picheral, Stéphane Gasparini, Lars Stemmann, Jean-Baptiste Romagnan, Alison Cawood, Stéphane Pesant, Carmen García-Comas, and Franck Prejger. Digital zooplankton image analysis using the zooscan integrated system. *Journal of Plankton Research*, 32(3):285–303, **2010**.
- [17] National data science bowl. <https://www.kaggle.com/c/datasciencebowl/data/>, (**Access date: 2015-11-26**).
- [18] Haiyong Zheng, Ruchen Wang, Zhibin Yu, Nan Wang, Zhaorui Gu, and Bing Zheng. Automatic plankton image classification combining multiple view features via multiple kernel learning. *BMC bioinformatics*, 18(16):1–18, **2017**.
- [19] Alessandra Lumini and Loris Nanni. Deep learning and transfer learning features for plankton classification. *Ecological Informatics*, 51:33–43, **2019**.

- [20] Alessandra Lumini, Loris Nanni, and Gianluca Maguolo. Deep learning for plankton and coral classification. *Applied Computing and Informatics*, **2020**.
- [21] Howard M Shapiro. *Practical flow cytometry*. John Wiley & Sons, **2005**.
- [22] Cabell S Davis, Fredrik T Thwaites, Scott M Gallager, and Qiao Hu. A three-axis fast-tow digital video plankton recorder for rapid surveys of plankton taxa and hydrography. *Limnology and Oceanography: Methods*, 3(2):59–74, **2005**.
- [23] Marc Picheral, Lionel Guidi, Lars Stemann, David Karl, Ghezlane (Rizlene) Id Daoud, and Gabriel Gorsky. The underwater vision profiler 5: An advanced instrument for high spatial resolution studies of particle size spectra and zooplankton. *Limnology and Oceanography: Methods*, 8:462–473, **2010**. doi: 10.4319/lom.2010.8.462.
- [24] Scott Samson, Thomas Hopkins, Andrew Remsen, Lawrence Langebrake, Tracey Sutton, and Jim Patten. A system for high-resolution zooplankton imaging. *IEEE Journal of Oceanic Engineering*, 26(4):671–676, **2001**.
- [25] MC Benfield, CJ Schwehm, and SF Keenan. Zoovis: a high resolution digital camera system for quantifying zooplankton abundance and environmental data. *Proc Am Soc Limnol Oceanogr*, pages 12–17, **2001**.
- [26] Jules S Jaffe. To sea and to see: That is the answer. *Methods in Oceanography*, 15:3–20, **2016**.
- [27] Robert K Cowen and Cedric M Guigand. In situ ichthyoplankton imaging system (isiis): system design and preliminary results. *Limnology and Oceanography: Methods*, 6(2):126–132, **2008**.
- [28] Norman MacLeod, Mark Benfield, and Phil Culverhouse. Time to automate identification. *Nature*, 467(7312):154–155, **2010**.
- [29] Elizabeth A Kellogg Peter F Stevens Walter S Judd, Christopher S Campbell and Michael J Donoghue. Plant systematics. *Sinauer Sunderland*, **2002**.
- [30] Alain Sournia, M-J Chrdtiennot-Dinet, and Michel Ricard. Marine phytoplankton: how many species in the world ocean? *Journal of Plankton Research*, 13(5):1093–1099, **1991**.

- [31] Eric C Orenstein, Oscar Beijbom, Emily E Peacock, and Heidi M Sosik. Whoi-plankton-a large scale fine grained visual recognition benchmark dataset for plankton classification. *arXiv preprint arXiv:1510.00745*, **2015**.
- [32] J Dai, R Wang, H Zheng, G Ju, and X Qiao. Zooplankonet: deep convolutional network for zooplankton classification. *oceans 2016*, **2016**.
- [33] Hansang Lee, Minseok Park, and Junmo Kim. Plankton classification on imbalanced large scale database via convolutional neural networks with transfer learning. In *2016 IEEE International Conference on Image Processing (ICIP)*, pages 3713–3717. IEEE, **2016**.
- [34] Jiuxiang Gu, Zhenhua Wang, Jason Kuen, Lianyang Ma, Amir Shahroudy, Bing Shuai, Ting Liu, Xingxing Wang, Gang Wang, Jianfei Cai, et al. Recent advances in convolutional neural networks. *Pattern Recognition*, 77:354–377, **2018**.
- [35] Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine learning*, 20(3):273–297, **1995**.
- [36] Shai Shalev-Shwartz and Shai Ben-David. *Understanding machine learning: From theory to algorithms*. Cambridge university press, **2014**.
- [37] Qiao Hu and Cabell Davis. Automatic plankton image recognition with co-occurrence matrices and support vector machine. *Marine Ecology Progress Series*, 295:21–31, **2005**.
- [38] Zhifeng Li, Feng Zhao, Jianzhuang Liu, and Yu Qiao. Pairwise nonparametric discriminant analysis for binary plankton image recognition. *IEEE Journal of Oceanic Engineering*, 39(4):695–701, **2013**.
- [39] Jeffrey Ellen, Hongyu Li, and Mark D Ohman. Quantifying california current plankton samples with efficient machine learning techniques. In *OCEANS 2015*. IEEE, **2015**.
- [40] Hongsheng Bi, Zhenhua Guo, Mark C Benfield, Chunlei Fan, Michael Ford, Suzan Shahrestani, and Jeffery M Sieracki. A semi-automated image analysis procedure for in situ plankton imaging systems. *PloS one*, 10(5):e0127121, **2015**.

- [41] Lin Chang, Ruchen Wang, Haiyong Zheng, Jialun Dai, and Bing Zheng. Phytoplankton feature extraction from microscopic images based on surf-pca. In *OCEANS 2016-Shanghai*, pages 1–4. IEEE, **2016**.
- [42] Emilie Poisson Caillault, André Bigand, et al. Comparative study on supervised learning methods for identifying phytoplankton species. In *2016 IEEE Sixth International Conference on Communications and Electronics (ICCE)*, pages 283–288. IEEE, **2016**.
- [43] Gloria Bueno, Oscar Deniz, Anibal Pedraza, Jesús Ruiz-Santaquiteria, Jesús Salido, Gabriel Cristóbal, María Borrego-Ramos, and Saúl Blanco. Automated diatom classification (part a): handcrafted feature approaches. *Applied Sciences*, 7(8):753, **2017**.
- [44] CS Davis, SM Gallager, MS Berman, LR Haury, and JR Strickler. The video plankton recorder (vpr): design and initial results. *Arch. Hydrobiol. Beih*, 36:67–81, **1992**.
- [45] Christian K Sieracki, Michael E Sieracki, and Charles S Yentsch. An imaging-in-flow system for automated analysis of marine microplankton. *Marine Ecology Progress Series*, 168:285–296, **1998**.
- [46] Md Moniruzzaman, Syed Mohammed Shamsul Islam, Mohammed Bennamoun, and Paul Lavery. Deep learning on underwater marine object detection: A survey. In *International Conference on Advanced Concepts for Intelligent Vision Systems*, pages 150–160. Springer, **2017**.
- [47] Hongwei Qin, Xiu Li, Zhixiong Yang, and Min Shang. When underwater imagery analysis meets deep learning: A solution at the age of big visual data. In *OCEANS 2015-MTS/IEEE Washington*, pages 1–5. IEEE, **2015**.
- [48] Dhruv Rathi, Sushant Jain, and S Indu. Underwater fish species classification using convolutional neural network and deep learning. In *2017 Ninth International Conference on Advances in Pattern Recognition (ICAPR)*, pages 1–6. IEEE, **2017**.
- [49] Jürgen Schmidhuber. Deep learning in neural networks: An overview. *Neural networks*, 61:85–117, **2015**.

- [50] Ali Sharif Razavian, Hossein Azizpour, Josephine Sullivan, and Stefan Carlsson. Cnn features off-the-shelf: an astounding baseline for recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 806–813. **2014**.
- [51] Robert K Cowen, S Sponaugle, K Robinson, and J Luo. Planktonset 1.0: Plankton imagery data collected from fg walton smith in straits of florida from 2014–06-03 to 2014–06-06 and used in the 2015 national data science bowl (ncei accession 0127422). *NOAA National Centers for Environmental Information*, **2015**.
- [52] Sander Dieleman, Jeffrey De Fauw, and Koray Kavukcuoglu. Exploiting cyclic symmetry in convolutional neural networks. In *International Conference on Machine Learning*, pages 1889–1898. PMLR, **2016**.
- [53] Ouyang Py, Hu Hong, and Shi Zhongzhi. Plankton classification with deep convolutional neural networks. In *2016 IEEE Information Technology, Networking, Electronic and Automation Control Conference*, pages 132–136. IEEE, **2016**.
- [54] Xiu Li and Zuoying Cui. Deep residual networks for plankton classification. In *OCEANS 2016 MTS/IEEE Monterey*, pages 1–4. IEEE, **2016**.
- [55] Jiangpeng Yan, Xiu Li, and Zuoying Cui. A more efficient cnn architecture for plankton classification. In *CCF Chinese Conference on Computer Vision*, pages 198–208. Springer, **2017**.
- [56] Pavel Pudil, Jana Novovičová, and Josef Kittler. Floating search methods in feature selection. *Pattern recognition letters*, 15(11):1119–1125, **1994**.
- [57] Jialun Dai, Zhibin Yu, Haiyong Zheng, Bing Zheng, and Nan Wang. A hybrid convolutional neural network for plankton classification. In *Asian Conference on Computer Vision*, pages 102–114. Springer, **2016**.
- [58] Jinna Cui, Bin Wei, Chao Wang, Zhibin Yu, Haiyong Zheng, Bing Zheng, and Hua Yang. Texture and shape information fusion of convolutional neural network for plankton image classification. In *2018 OCEANS-MTS/IEEE Kobe Techno-Oceans (OTO)*, pages 1–5. IEEE, **2018**.

- [59] Jaroslav Bureš, Tuomas Eerola, Lasse Lensu, Heikki Kälviäinen, and Pavel Zemčík. Plankton recognition in images with varying size. In *Pattern Recognition. ICPR International Workshops and Challenges. ICPR 2021. Lecture Notes in Computer Science*. Springer, Cham, **2021**.
- [60] Kunihiro Fukushima and Sei Miyake. Neocognitron: A self-organizing neural network model for a mechanism of visual pattern recognition. In *Competition and Cooperation in Neural Nets*, pages 267–285. Springer, **1982**.
- [61] Yann LeCun, Patrick Haffner, Léon Bottou, and Yoshua Bengio. Object recognition with gradient-based learning. In *Shape, Contour and Grouping in Computer Vision*, pages 319–345. Springer, **1999**.
- [62] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in Neural Information Processing Systems*, 25:1097–1105, **2012**.
- [63] Md Zahangir Alom, Tarek M Taha, Christopher Yakopcic, Stefan Westberg, Paheding Sidike, Mst Shamima Nasrin, Brian C Van Esesn, Abdul A S Awwal, and Vijayan K Asari. The history began from alexnet: A comprehensive survey on deep learning approaches. *arXiv preprint arXiv:1803.01164*, **2018**.
- [64] Hamed Habibi Aghdam and Elnaz Jahani Heravi. Guide to convolutional neural networks. *New York, NY: Springer*, 10(978-973):51, **2017**.
- [65] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778. **2016**.
- [66] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, **2014**.
- [67] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4700–4708. **2017**.
- [68] Christian Szegedy, Sergey Ioffe, Vincent Vanhoucke, and Alexander A Alemi. Inception-v4, inception-resnet and the impact of residual connections on learning. In *Thirty-first AAAI Conference on Artificial Intelligence*. **2017**.

- [69] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255. IEEE, **2009**.
- [70] Michael I Jordan and Tom M Mitchell. Machine learning: Trends, perspectives, and prospects. *Science*, 349(6245):255–260, **2015**.
- [71] David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. Learning representations by back-propagating errors. *Nature*, 323(6088):533–536, **1986**.
- [72] Vidya Muthukumar, Adhyayan Narang, Vignesh Subramanian, Mikhail Belkin, Daniel Hsu, and Anant Sahai. Classification vs regression in overparameterized regimes: Does the loss function matter? *Journal of Machine Learning Research*, 22(222):1–69, **2021**.
- [73] Michael A Nielsen. *Neural networks and deep learning*, volume 25. Determination Press San Francisco, CA, **2015**.
- [74] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2980–2988. **2017**.
- [75] Sepp Hochreiter, Yoshua Bengio, Paolo Frasconi, Jürgen Schmidhuber, et al. Gradient flow in recurrent nets: the difficulty of learning long-term dependencies, **2001**.
- [76] Xavier Glorot, Antoine Bordes, and Yoshua Bengio. Deep sparse rectifier neural networks. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pages 315–323. JMLR Workshop and Conference Proceedings, **2011**.
- [77] Razvan Pascanu, Tomas Mikolov, and Yoshua Bengio. On the difficulty of training recurrent neural networks. In *International Conference on Machine Learning*, pages 1310–1318. PMLR, **2013**.
- [78] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, 15(1):1929–1958, **2014**.

- [79] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical Image Computing and Computer-assisted Intervention*, pages 234–241. Springer, **2015**.
- [80] Eric C Orenstein and Oscar Beijbom. Transfer learning and deep feature extraction for planktonic image data sets. In *2017 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 1082–1088. IEEE, **2017**.
- [81] Jason Yosinski, Jeff Clune, Yoshua Bengio, and Hod Lipson. How transferable are features in deep neural networks? *arXiv preprint arXiv:1411.1792*, **2014**.
- [82] Francisco Caio Maia Rodrigues, Nina ST Hirata, Antonio A Abello, T Leandro, De La Cruz, Rubens M Lopes, and Roberto Hirata Jr. Evaluation of transfer learning scenarios in plankton image classification. In *VISIGRAPP (5: VISAPP)*, pages 359–366. **2018**.
- [83] Boaz Shmueli. Multi-class metrics made simple, part ii: the f1-score. *Retrieved from Towards Data Science: <https://towardsdatascience.com/multi-class-metrics-made-simplepart-ii-the-f1-score-ebe8b2c2ca1>*, **2019**.