



YAPAY ZEKA TABANLI DRONE OPTİMİZASYONU

Enes CENGİZ

**YÜKSEK LİSANS TEZİ
ELEKTRİK ELEKTRONİK MÜHENDİSLİĞİ ANA BİLİM DALI**

**GAZİ ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ**

HAZİRAN 2020

Enes CENGİZ tarafından hazırlanan “YAPAY ZEKA TABANLI DRONE OPTİMİZASYONU” adlı tez çalışması aşağıdaki jüri tarafından OY BİRLİĞİ ile Gazi Üniversitesi Elektrik Elektronik Mühendisliği Ana Bilim Dalında YÜKSEK LİSANS TEZİ olarak kabul edilmiştir.

Danışman: Prof. Dr. Cemal YILMAZ

Elektrik-Elektronik Mühendisliği Ana Bilim Dalı, Gazi Üniversitesi

Bu tezin, kapsam ve kalite olarak Yüksek Lisans Tezi olduğunu onaylıyorum.

.....

Başkan: Prof. Dr. Recep DEMİRCİ

Bilgisayar Mühendisliği Ana Bilim Dalı, Gazi Üniversitesi

Bu tezin, kapsam ve kalite olarak Yüksek Lisans Tezi olduğunu onaylıyorum.

.....

Üye: Doç. Dr. Hamdi Tolga KAHRAMAN

Yazılım Mühendisliği Ana Bilim Dalı, Karadeniz Teknik Üniversitesi

Bu tezin, kapsam ve kalite olarak Yüksek Lisans Tezi olduğunu onaylıyorum.

.....

Tez Savunma Tarihi: 09/06/2020

Jüri tarafından kabul edilen bu çalışmanın Yüksek Lisans Tezi olması için gerekli şartları yerine getirdiğini onaylıyorum.

.....
Prof. Dr. Sena YAŞYERLİ
Fen Bilimleri Enstitüsü Müdürü

ETİK BEYAN

Gazi Üniversitesi Fen Bilimleri Enstitüsü Tez Yazım Kurallarına uygun olarak hazırladığım bu tez çalışmada;

- Tez içinde sunduğum verileri, bilgileri ve dokümanları akademik ve etik kurallar çerçevesinde elde ettiğimi,
- Tüm bilgi, belge, değerlendirme ve sonuçları bilimsel etik ve ahlak kurallarına uygun olarak sunduğumu,
- Tez çalışmada yararlandığım eserlerin tümüne uygun atıfta bulunarak kaynak gösterdiğimi,
- Kullanılan verilerde herhangi bir değişiklik yapmadığımı,
- Bu tezde sunduğum çalışmanın özgün olduğunu,

bildirir, aksi bir durumda aleyhime doğabilecek tüm hak kayıplarını kabullendiğimi beyan ederim.

.....
Enes CENGİZ
09/06/2020

YAPAY ZEKA TABANLI DRONE OPTİMİZASYONU

(Yüksek Lisans Tezi)

Enes CENGİZ

GAZİ ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

Haziran 2020

ÖZET

Son yıllarda derin öğrenme algoritmalarının kullanımında önemli bir artış göze çarpmaktadır. Uygulamalarda derin öğrenme modellerinden konvolüsyon sinir ağı (CNN) özellikle görüntü işlemede insanlar ve araçlar gibi önemli nesneleri diğer nesnelerden ayırmak için sıklıkla kullanılmaktadır. 2012'de ImageNet Büyük Ölçekli Görsel Tanıma Yarışması'ndan (ImageNet Large Scale Visual Recognition Competition, ILSVRC) sonra, uygulamalarda CNN kullanımı oldukça yaygınlaşmaktadır. Görüntü işleme donanımlarının gelişmesiyle görüntü işleme süreci önemli ölçüde azaltılmaktadır. Bu gelişmeler sayesinde derin öğrenme üzerine yapılan çalışmaların performansı artmaktadır. Kuş bakışı farklı yüksekliklerden drone ile çekilen görüntülerden nesne tespiti oldukça zor olmaktadır. Bu çalışmada, dronlar tarafından çekilen videoda nesneleri (insan, otomobiller, bisikletler ve motosikletler) tespit etmek için bir sistem geliştirilmiştir. Sinir ağının eğitimi ve test edilmesi için, farklı yüksekliklerden kuş bakışı çekilen görüntülerden oluşan bir veri seti hazırlanmıştır. Çalışmada 2015 yılında ImageNet yarışmasında birinci olan Residual Networks (ResNet-50) ağırlıkları kullanılan RetinaNet modeli tercih edilmiştir. Sinir ağını eğitmek için, kullanıcılarına Nvidia Tesla K80 GPU desteği sunan Google'ın açık kaynak ortamı Google Colabratory kullanılmıştır. Bu ortamda, sinir ağının eğitim ve test süreçleri hızlı bir şekilde tamamlanmıştır. Eğitilmiş modelde, nesnelerin tespit işlemi için farklı eşik değerleri kullanılarak performans karşılaştırması yapılmıştır. Bu tez çalışmasının sonuçlarına göre, belirlenen nesnelerin tespitinde f1 skor değeri, 0.4 eşik değeri kullanılarak 0.906 olarak belirlenmiştir.

Bilim Kodu : 90542
Anahtar Kelimeler : Derin Öğrenme, CNN, Nesne Tespiti
Sayfa Adedi : 62
Danışman : Prof. Dr. Cemal YILMAZ

ARTIFICIAL INTELLIGENCE BASED DRONE OPTIMIZATION

(M. Sc. Thesis)

Enes CENGİZ

GAZİ UNIVERSITY

GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES

June 2020

ABSTRACT

It has attracted attention that significant increase in the use of deep learning algorithms recently. Convolutional neural network (CNN) from deep learning models is frequently used in applications, particularly in image processing to distinguish significant objects such as people and vehicles from other objects. After ImageNet Large Scale Visual Recognition Competition (ILSVRC) at 2012, usage of CNNs in applications has become notably widespread. Image processing process has been significantly reduced with the development of image processing equipment. Thanks to these developments, the performance of studies on deep learning have increased. Object detection has been quite difficult since bird's eye view images are taken by drone from different altitudes. In this study, a system has developed to detection objects (human, automobiles, bicycles and motorcycles) in video taken by drones. A dataset consisting of a bird's eye images taken from different altitudes has prepared in order to the training and testing of the neural network. In the study, the RetinaNet model, in which uses weights of Residual Networks (ResNet-50) was the first in the ImageNet competition in 2015, has preferred. Google Colab, the open source platform of Google, which offers Nvidia Tesla K80 GPU support to its users, is used to train the neural network. In this platform, training and testing procedures of the neural network have completed quickly. In the trained model, performance comparison is made by using different threshold values for objects detection process. According to results from this thesis study, f1 score value in specified objects detection is determined as 0.906 by using of 0.4 threshold value.

Science Code : 90542

Key Words : Deep Learning, CNN, Object Detection

Page Number : 62

Supervisor : Prof. Dr. Cemal YILMAZ

TEŞEKKÜR

Çalışmalarım boyunca tecrübelerinde yararlandığım, yanında çalışmaktan onur duyduğum ve tezin yürütülmesinde yol gösteren danışmanım sayın Prof. Dr. Cemal YILMAZ'a,

Araştırmanın planlanmasından yazımına kadar tüm aşamalarında yardımlarını esirgemeyen, beni bilgi ve birikimleriyle yönlendiren Karadeniz Teknik Üniversitesi öğretim üyesi sayın Doç. Dr. Hamdi Tolga KAHRAMAN'a ve Afyon Kocatepe Üniversitesi öğretim üyesi sayın Dr. Öğr. Üyesi Fatih BAYRAM'a,

Bugünlere gelmemde büyük emekleri olan, bana insani ve ahlaki değerler kazandıran anneme ve babama,

Desteğini hiçbir zaman esirgemeyen ve zor günlerimde yanımda olan eşime, bu tezin hazırlanma sürecinde dünyaya gelen ve bana çalışma azmi veren oğluma,

Sonsuz teşekkürler sunarım.

İÇİNDEKİLER

	Sayfa
ÖZET	iv
ABSTRACT.....	v
TEŞEKKÜR.....	vi
İÇİNDEKİLER	vii
ÇİZELGELERİN LİSTESİ.....	ix
ŞEKİLLERİN LİSTESİ	x
RESİMLERİN LİSTESİ	xii
SİMGELER VE KISALTMALAR.....	xiii
1. GİRİŞ.....	1
2. KONVOLÜSYON SİNİR AĞLARI (CNN)	9
2.1. CNN Mimarisinin Katmanları.....	11
2.1.1. Konvolüsyon katmanı	11
2.1.2. Havuzlama katmanı (Pooling)	15
2.1.3. Aktivasyon katmanı	16
2.1.4. Düzleştirme katmanı (Flatten).....	19
2.1.5. Tam bağlantılı katman.....	20
2.1.6. Normalizasyon katmanı	20
2.1.7. Dropout katmanı.....	20
2.2. CNN Mimariler	20
2.2.1. LeNet-5 mimarisi	22
2.2.2. AlexNet mimarisi	22
2.2.3. VGGNet mimarisi	23

	Sayfa
2.2.4. Inception mimarisi	24
2.2.5. ResNet mimarisi.....	26
3. ÖĞRENME AKTARIMI (TRANSFER LEARNING) YÖNTEMİ.....	31
3.1. İnce Ayar (Fine-Tuning)	32
4. MATERYAL VE METOT	35
4.1. Veri Setleri	37
4.1.1. Stanford veri seti	37
4.1.2. Tasarlanan veri seti	39
4.2. Çalışmada Kullanılan CNN Model	44
4.3. TensorFlow Kütüphanesi	47
4.4. Eğitim ve Test Süreci	48
4.5. Kayıp Fonksiyonları.....	50
4.6. Hata Matrisinin Çıkarılması (Confusion Matrix).....	51
5. SONUÇ VE ÖNERİLER	55
KAYNAKLAR	57
ÖZGEÇMİŞ	61

ÇİZELGELERİN LİSTESİ

Çizelge	Sayfa
Çizelge 4.1. Stanford drone veri seti genel bilgileri	39
Çizelge 4.2. Tasarlanan veri setinin genel bilgileri.....	41
Çizelge 4.3. Etiketlenen görüntülerden bazısının uygun formda yazımı.....	43
Çizelge 4.3. Hata matrisinin çıkarılması.....	51
Çizelge 4.4. Eşik değerleriyle hata matrisinin belirlenmesi	52
Çizelge 4.5. Farklı eşik değerlerine göre kesinlik, duyarlılık ve f1 skor değerleri.....	53

ŞEKİLLERİN LİSTESİ

Şekil	Sayfa
Şekil 2.1. Örnek bir RGB görüntüsü.....	10
Şekil 2.2. CNN mimarisinin genel görünüşü	10
Şekil 2.3. Bir görüntüye konvolüsyon katmanının uygulanması.....	12
Şekil 2.4. Giriş görüntüsüne konvolüsyon katmanı uygulanarak çıkış görüntüsünün bir pikselinin elde edilmesi.....	12
Şekil 2.5. Giriş görüntüsünün ve uygulanacak filtrenin matris gösterimi	13
Şekil 2.6. Kenarlara sıfır ekleme yöntemi gösterimi	14
Şekil 2.7. Giriş görüntüsüne uygulanan maksimum ve ortalama havuzlama katmanı ...	15
Şekil 2.8. ReLU aktivasyon fonksiyonu	17
Şekil 2.9. Sigmoid aktivasyon fonksiyonu	17
Şekil 2.10. Tanh aktivasyon fonksiyonu.....	18
Şekil 2.11. Sızıntılı (Leaky) ReLU aktivasyon fonksiyonu	19
Şekil 2.12. Düzleştirme (Flatten) katmanının uygulanışı	19
Şekil 2.13. ImageNet yarışmasında birinci olan CNN modelleri ve top-5 hata oranları	21
Şekil 2.14. LeNet-5 CNN mimarisi	22
Şekil 2.15. AlexNet CNN mimarisi	23
Şekil 2.16. VGGNet CNN mimarisi	23
Şekil 2.17. Inception CNN mimarisi.....	25
Şekil 2.18. ResNet CNN mimarisi.....	27
Şekil 2.19. Derin bir CNN modeli	28
Şekil 3.1. Geleneksel öğrenme ve transfer öğrenme yöntemlerinin genel gösterimi	31
Şekil 3.2. İnce ayar (fine tuning) yöntemleri	33
Şekil 4.1. Derin öğrenme süreci.....	35

Şekil	Sayfa
Şekil 4.2. Kullanılan modelde öğrenme aktarımının gerçekleştirilmesi.....	36
Şekil 4.3. Etiketlenen sınıfın koordinatları ve olması gereken format şekli.....	43
Şekil 4.4. ResNet modelinin ilk 34 katmanının gösterimi	45
Şekil 4.5. Google Colaboratory çalışma ortamı.....	48
Şekil 4.6. Kullanılan mimarinin regresyon ve sınıflandırma çıkışının gösterimi	50
Şekil 4.7. (a). Sınıflandırma kaybı, (b). Regresyon kaybı	51



RESİMLERİN LİSTESİ

Resim	Sayfa
Resim 4.1. Stanford veri setinden görüntüler	38
Resim 4.2. Tasarlanan veri setinden görüntüler.....	40
Resim 4.3. VIA aracılığıyla görüntü etiketleme işlemi	42



SİMGELER VE KISALTMALAR

Bu çalışmada kullanılmış simgeler ve kısaltmalar, açıklamaları ile birlikte aşağıda sunulmuştur.

Simgeler

Açıklamalar

m

Metre

F

Filtre sayısı

P

Dolgu sayısı

w

Ağırlık

l

Öğrenme hızı

Kısaltmalar

Açıklamalar

CNN

Convolution Neural Network

CSV

Virgülle Ayrılmış Değerler

DenseNet

Dense Convolutional Network

IoU

Intersection over Union

ILSVRC

ImageNet Large Scale Visual Recog. Competition

İHA

İnsansız Hava Aracı

ResNet

Residual Neural Network

ReLU

Rectified Linear Units

R-CNN

Region with Convolutional Neural Networks

YOLO

You Only Look Once

1. GİRİŞ

Hedef tespiti ve takibi, bilgisayarla görü alanında son zamanlarda en çok ilginin gösterildiği ve çalışmanın yapıldığı problemlerden biri olarak karşımıza çıkmaktadır [1]. Teknolojinin gelişimiyle birlikte dronlar askeri ve sivil uygulamalarda çok yaygın şekilde kullanılmaktadır. Dronlar günümüzde birçok zorlu görevi üstlenir hale gelerek problemlerin çözümüne katkı sağlamaktadır. Havadan nesne ve insan tespiti, arama-kurtarma ve tarım alanlarının verimlilik analizi gibi uygulamalarda yer alan dronlar başka birçok alanda da kullanılmaktadır. Drondan aktarılan hava görüntülerinde şüpheli nesne veya insan tespiti askeri uygulamalarda birçok kolaylıklar sağlamaktadır. Bir bölgenin güvenliğini sağlamak için dronlarla o bölgeyi kontrol ederek oluşabilecek tehditlerin önüne geçilebilmektedir. Tehdit eden nesne veya insan ise bunun dronla tespit ve takip edilmesi gerekmektedir. Bu tespit ve takip işleminin ardından uygun pozisyon alınarak tehdit unsuruna müdahale edilmektedir. Bu işlemlerin askeri alanda uygulanmasından ötürü kontrolü çok hassas bir şekilde olmaktadır. Bu nedenle havadan hedef tespiti ve takibi araştırma alanı olarak karşımıza çıkmaktadır.

Havadan algılanacak olan nesnenin, yüksek görüş mesafesi nedeniyle geleneksel hedef algılama algoritmalarıyla tespiti ve takibi problem teşkil etmektedir. Bu problem, farklı ortamların görüntü gürültüsünden, kamera çekim açılarından ve ortamdaki ışık değişimlerinden kaynaklanmaktadır [2]. Bir hava görüntüsünden nesne, insan tespit ve takip etme görevi, bu nedenlerden dolayı geleneksel hedef algılama algoritmalarıyla zorlu olmaktadır. Bu zorlu görevin üstesinden gelebilmek için yakın zamanda derin öğrenme yöntemleri sunulmuştur [3].

Yapay zekanın alt dallarından biri olan derin öğrenmenin çalışma mantığı, insan beynindeki sinir ağlarının nöronuna benzemektedir. Derin öğrenme modelleri, bir görevi yapmak için eğitim veri setiyle eğitilmektedir. Başlangıçta herhangi bir bilgiye sahip olmayan bu modeller eğitim sonucunda istenilen bilgiler ile eğitilmektedir. Bu derin öğrenme modelleri, temelde, bilgisayarın bir insan gibi görmesini ve görselleştirmesini sağlayan bilgisayar görme alanında kullanılmaktadır. Derin öğrenme algoritması olan konvolüsyon sinir ağları (CNN) nesne tespit uygulamalarında sıkça kullanılmaktadır. CNN tabanlı yaklaşımlar daha

iyi öğrenme kapasitesine sahip olduğundan havadan çekilen görüntülerde nesne algılama performansını artırabilmektedir.

Son yıllarda, veri setlerinin iyileştirilmesi ve görüntü işleme donanımlarının teknolojik gelişimiyle birlikte, veri güdümlü CNN'ler görüntü bilgisinin çıkarılmasında, nesne algılamada büyük ilerleme kaydetmiştir.

1990'ların sonundan itibaren özellikle yakın zamanda derin öğrenme yöntemleri kullanılarak nesne, insan algılama üzerine önemli çalışmalar yapılmaktadır. Son zamanlarda CNN modellerinin geliştirilmesi ile genel nesne sınıflandırması, nesne tespiti ve hatta insan tespiti gibi çeşitli görme görevlerine başarıyla uygulanmaktadır. Dronlar aracılığıyla belirli bir yükseklikten elde edilen hava görüntüleri ile eğitilen CNN'ler nesne tespitinde kullanılmaktadır. Tespit işlemi yapılırken modelden yüksek performans sergilemesi istenmektedir. Sabit bir yükseklikten elde edilen görüntüler ile eğitilen sinir ağı, uygulama esnasında farklı yüksekliklerde kullanılmasıyla olumsuz sonuçlar verebilmektedir. Bu farklı yüksekliklerde tespit edilecek görüntünün boyutunun değişmesinden kaynaklanmaktadır. Modelin performansında oldukça düşük sonuçlar veren bu durum uygulamada hiç istenmeyen bir problemdir. Bu tez çalışmasında farklı yüksekliklerden drone aracılığıyla insan, araba, bisiklet ve motosiklet tespiti için CNN modeli olan Residual sinir ağı kullanılmıştır. Kısaca ResNet olarak bilinen bu model oldukça derin ağlarda yüksek performansla çalışmaktadır. Kullanılan ResNet ağının eğitimi için biri açık kaynak olan Stanford drone veri seti, diğeri kendi oluşturduğumuz drone veri seti olmak üzere iki farklı veri seti kullanılmıştır. Farklı yüksekliklerden elde edilen drone görüntüleriyle sinir ağı eğitilmiş ve ardından teste tabii tutulmuştur. Test işlemi yapılırken farklı yüksekliklerden elde edilmiş görüntüler kullanılmıştır. Tespit işleminde farklı eşik değerlerinde performans sonuçları karşılaştırılmış ve uygun olan eşik değeri belirlenmiştir.

Derin öğrenme tabanlı nesne, insan tespiti ile ilgili yerli ve yabancı literatürde yapılmış çalışmalardan bazıları aşağıda kısaca anlatılmıştır.

[4]'te yapılan çalışmada DenseNet tabanlı CNN modeli kullanılmıştır. Bu modelle tahıl alanlarında oluşan devedikenlerini tespit ederek tahıl oluşumunda verimliliği artırmaya yönelik uygulama yapılmıştır. Veri seti olarak biri kış buğdayı diğeri bahar arpası olan iki farklı tahıl alanı kullanılmıştır. Drone aracılığıyla bu iki alandan 10m ve 50m

yüksekliklerden çekilen görüntülerle CNN modeli eğitilmiş ve ardından test işlemi uygulanarak karşılaştırmalar yapılmıştır. Karşılaştırmaya göre DenseNet modelinin başarımları drone aracılığıyla 10m'den çekilen görüntülerden %95 olurken 50m'den çekilen görüntülerden elde edilen başarımları ise %97 olarak gözlemlenmiştir.

[5]'te dronlar ile gerçek zamanlı nesne algılama görevi için bir model mimarisi önerilmektedir. Yapılan çalışmada farklı görüntü boyutlarında algılama performansını artırmak ve sınıf dengesizliği probleminden kaçınmak için özellik piramit ağı sunulmuştur. Tespit modeli için omurga olarak ResNet ve MobileNet adlı iki sinir ağı kullanılmıştır. Çalışmada yayalar, araçlar, bisikletler vb. farklı nesneleri içeren VisDrone'18 veri setinden faydalanılmıştır. ResNet ile birleştirilen modelde algılama doğruluğu açısından daha iyi sonuçlar sağlanmasına rağmen, MobileNet ile birleştirildiğinde algılama doğruluğundan ödün vermeden gerçek zamanlı hızlarla çalıştığı görülmüştür.

[6]'da sabit olmayan bir kamera ile havadan farklı yüksekliklerde çekilen videolarda insan tespiti için üç farklı derin öğrenme modelini birleştiren öğrenme yöntemi kullanılmaktadır. Bu modellerin biri denetimli konvolüsyon sinir ağı {supervised convolutional neural network (S-CNN)}, bir diğeri daha önceden eğitilmiş CNN, üçüncüsü ise ağırlıkları hassas bir şekilde ayarlanmayan hiyerarşik aşırı öğrenme makinesidir. Modeller, açık kaynak UCF-ARG hava veri seti üzerinde eğitilmiş ve test işlemi yapılmıştır. Yapılan çalışmanın deneysel sonuçlarında önerilen yöntemin havadan insan tespiti için performans sonuçları gösterilmiştir. Sonuçlara göre önceden eğitilmiş CNN %98.09 ortalama doğrulukla tespit işlemini gerçekleştirmiştir. S-CNN, soft-max ile %95,6, destek vektör makineleri ile %91,7 ortalama doğruluk sağlamıştır. Hiyerarşik aşırı öğrenme makinesi ise %95,9 ortalama doğrulukla tespit işlemini gerçekleştirmiştir.

[7]'de nesne algılama işlevi için You Only Look Once (YOLO) ve FAST R-CNN (Region with Convolutional Neural Networks) ağları kombine edilerek kullanılmıştır. YOLO ile Fast R-CNN birleştirilerek performanstaki artış gözlemlenmiştir. Çeşitli modellerin Fast R-CNN ile birleştirilip etkisi incelendiğinde Fast R-CNN'nin diğer versiyonları düşük bir fayda sağlarken, YOLO önemli bir performans artışı sağlamaktadır. Çalışmada farklı veri setleri kullanılarak gerçek zamanlı olarak nesne algılama işlevi bu ağlar ile karşılaştırılmıştır.

[8]'de nesne tespiti için derin CNN modellerden biri olan VGG ağı kullanılmıştır. VGG ağını eğitebilmek için oldukça uzun süreye ihtiyaç vardır. Ayrıca bu model kullanılarak yapılan tespit işleminde nesnelerin yerini belirlemede bazı sorunlar yaşanmaktadır. Çalışmada bu sorunu ortadan kaldırmak ve başarımı artırmak için bir mimari önerilmektedir. Önerilen mimari ile sisteme yeni katmanlar eklenerek modelin performansı artırılmak istenmiştir. VGG mimarisinde tam bağlantı katmanlarının hemen öncesine geliştirilen ResNet bağlantısı eklenmiştir. Deneysel çalışma olarak 20 görüntü kategorisi olan ve yaklaşık 20 bin eğitim ve test görüntüsü içeren PASCAL VOC2012 veri seti kullanılarak yapılmıştır. Geliştirilmiş ResNet ağıyla, katman boyutunu ve eğitim hatasını azaltmak için kısayol yöntemi kullanılmıştır. Deneysel sonuçlara göre geliştirilmiş konvolüsyon mimarisi, nesneleri tespit etmede belirgin bir hassasiyet elde etmektedir. Önerilen yöntemin, ortalama 85.8 mAP hassasiyet ile nesneleri tespit ettiği görülmüştür.

[9]'da insansız hava araçları aracılığıyla hareketli kameralar tarafından elde edilen videolardan görüntüde değişiklik tespiti için bir sistem sunulmaktadır. Geliştirilen sistem art arda görüntü sıralama tekniklerinin, CNN'e dayanan derin bir öğrenme modeli ile birleşimine dayanmaktadır. Çalışmada farklı ortamlardaki değişiklikleri tespit etmek için modelin hassasiyeti gelişmiş yöntemlerle karşılaştırılmıştır. Değişiklik algılama algoritmalarının eğitimi için farklı senaryolardan etiketlenmiş görüntülerden oluşan Change Detection 2014 veri kümesi kullanılmıştır [10]. Eğitim için dinamik arka plan, gelip giden nesne hareketi ve kötü hava koşullarının bulunduğu görüntüler modele giriş olarak verilmiştir. Böylece algoritmanın gerçek uçuş koşullarında olduğu gibi arka plandaki değişikliklere karşı performansını daha doğru test etmek mümkün olmuştur. Test sonuçlarına göre sistemin görüntü tespitinden elde ettiği ortalama f1 ölçüm skoru %97'yi aşmıştır. Ayrıca gelip giden (aralıklı) nesne hareketinde ise %99'un üzerinde bir hassasiyetle ölçüm gerçekleştirilmiştir.

[11]'de iki farklı derin öğrenme modelinin araç tespitinde başarı performans karşılaştırması yapılmıştır. Drone ile elde edilen hava görüntüleri kullanılarak sistem eğitilmiş ve ardından test işlemi yapılmıştır. Tespit edilecek olan nesnelerin küçük olması, görüntü başına düşen yüksek sayıda nesne gibi belirli kısıtları olan iki drone görüntüleme veri tabanında iki nesne algılama algoritmasının (YOLOv3 ve Faster R-CNN) kapsamlı deneysel karşılaştırması gerçekleştirilmiştir. Penn State Üniversitesi ve Stanford Üniversitesinin veri setleri ile YOLOv3 ve Faster R-CNN nesne algılama algoritmaları başarımlarına göre Penn

Üniversitesi veri setinde Faster R-CNN ortalama Intersection over Union (IoU) değeri 0,95 iken YOLOv3 veri setinde 0,48, Standford Üniversitesi veri setinde ise Faster R-CNN ortalama IoU değeri 0.93 iken YOLOv3 veri setinde 0,89 değeri elde edilmiştir.

[12]'de kızılötesi görüntülerinden hedef tanıma ve koordinatlarının tanımlanması için küresel ortalama havuzlama ile bir CNN model önerilmiştir. Çalışmada önerilen bu CNN temelli model ile makine öğrenme yöntemi olan destek vektör makineleri ile karşılaştırma yapılmıştır. Karşılaştırma sonucunda CNN temelli model ile insan tespit yaklaşımlarının destek vektör makineleri temelli yaklaşımlara göre daha iyi hedef tespit ettiği görülmüştür.

[13]'te insansız hava araçlarının karaya inişlerine yardımcı olmak için pistlerin doğru bir şekilde tespitine ve lokalizasyonuna odaklanılmaktadır. Çalışmada pist tespiti ile ilgili önceki basit görüntü işleme algoritmalarının aksine derin öğrenme mimarilerine dayanan pist tespit algoritması geliştirilmiştir. Tespit edilen pistin yerini belirlemek için hem geleneksel çizgi algılama algoritmaları hem de derin öğrenme modelleri kullanılmıştır. Çalışmada görüntülerden özellikler elde etmek için kullanılan dört CNN modelinin karşılaştırılması verilmiştir. Kullanılan VGG16, ResNet50, ResNet152 ve DenseNet161 CNN modelleri arasında Resnet50 ve Resnet152 modelleri başarı oranına göre neredeyse aynı performansta olup diğerlerine göre daha iyi sonuç vermiştir. Tespit etme hızına göre ise ResNet50 modelinin daha hızlı sonuç verdiği görülmüştür. Modelde pist sınıflandırması yaklaşık %97'lik bir doğrulukla elde edilirken, pistlerin koordinatları ortalama 0,8 IoU değeri ile belirlenmiştir

[14]'te havadan çekilen görüntülerdeki yayaları tespit etmek için derin öğrenme modeli kullanılmıştır. Çalışmada kullanılan derin öğrenme modelleri PASCAL VOC 2007 veri setinde yüksek hassasiyet göstermelerine rağmen, hava görüntülerindeki yayaların tespiti için yüksek algılama performansına sahip değildir. Bunun üstesinden gelmek için bu çalışmada görüntü bölümü ve birleşme noktası dönüşümü kullanılmıştır. Çalışmanın deneysel sonuçlara bakıldığında bu tür ön işleme yöntemleri, bazı derin öğrenme modelleri için tespit oranlarını önemli ölçüde artırabildiğini göstermiştir.

[15]'te çalışmada havadan gemi tespiti için derin öğrenme modeli önerilmiştir. Gemi tespit işlemi, önerilen model dışında 3 farklı yöntem (CFAR tabanlı metot, YOLOv2 ve VGG16 derin öğrenme modelleri) ile karşılaştırılmıştır. Önerilen modelde önce derin öğrenme

modeli olan ResNet açık kaynak veri seti PASCAL VOC 2007'de eğitilmiştir. Daha sonra model, SAR gemi veri setinin eğitim setiyle transfer öğrenim yapılarak eğitilmiştir. Modelin test işlemi SAR gemi veri setinin test seti ile yapılarak sonuçlar ortaya konulmuştur. Önerilen model ile diğer modeller ortalama hassasiyet ve eğitim süresi bakımından karşılaştırılmıştır. Ortalama hassasiyete göre karşılaştırma yapıldığında önerilen model %94,7 ortalama hassasiyetle diğer modellere göre daha iyi performans ortaya koymuştur. Eğitim süresine göre karşılaştırıldığında ise YOLOv2 önerilen modele göre daha kısa sürede eğitimi tamamlamış olsa da ortalama hassasiyet olarak daha geri planda kalmıştır.

[16]'da son yıllarda oldukça popüler olan derin öğrenme modellerinin eğitim süreleri üzerine bir çalışma yapılmıştır. Doğruluk ve hız performansı derin öğrenme modelleri için oldukça önemlidir. Öğrenme modellerinde eğitim sürelerinin kısa olması istenmektedir. Bu çalışmada farklı öğrenme ağları kullanılarak TensorFlow'da CIFAR-10 veri seti ile eğitim yapılmıştır [17]. Ayrıca derin öğrenme işlemi 3 farklı donanım platformunda (Intel Xeon E5-2630, NVIDIA TITAN X ve NVIDIA Tesla K20c) gerçekleştirilmiştir. Inception-v3, Resnet50 ve Resnet101 derin öğrenme algoritmaları 3 farklı donanım platformunda veri seti ile eğitilerek eğitim süreleri karşılaştırılmıştır. Test sonuçlarına bakıldığında ResNet50 algoritmasının diğer mimarilere göre daha kısa sürede eğitim işlemini tamamladığı sonucuna varılmıştır.

[18]'de havadan Seoul National Üniversitesi'nde çekilmiş olan görüntülerle eğitilen derin öğrenme modeliyle yaya ve araba sınıflarının tespit ve takip işlemi gerçekleştirilmiştir. Çalışmada VGG16 ve MobileNet derin öğrenme modelleri veri setleriyle eğitilerek karşılaştırılmıştır. Karşılaştırma sonucuna göre MobileNet modeli VGG16'a göre daha hızlı tespit ettiği görülmüştür. Çalışmada ayrıca derin öğrenme modeli COCO veri setiyle eğitilmiştir [19]. Çalışmanın test bölümünde aktarılanlara göre; Seoul National Üniversitesi'nden alınan görüntülerle eğitilen model açık kaynak COCO veri setiyle eğitilen modelden daha iyi sonuçlar vermiştir.

[20]'de insansız hava aracı ile havadan araç tespiti gerçekleştiren bir çalışma yapılmıştır. Araç tespit işlemi, derin öğrenme modelleri ile gerçekleştirilip çalışma performansı karşılaştırılmıştır. Öğrenme modellerini eğitmek için çalışmanın yazarları kendi veri setini ve açık kaynak olarak sunulan Stanford Üniversitesinin kampüsünden drone ile çekilen Stanford drone veri setini kullanmışlardır. Modeli kendi veri setleriyle eğitmek için

görüntüleri önceden etiketleyerek eğitime hazırlamışlardır. Tespit edilmesi istenen araçlar, veri seti içerisindeki görüntülerde tek tek işaretlenerek etiketleme işlemi yapılmıştır. Araç tespitinde kullanılan derin öğrenme modelleri (SSD, YOLO ve YOLOv2) her iki veri seti ile eğitilmiş ve ardından teste tabii tutulmuştur. Test sonucunda öğrenme modellerinin her iki veri setiyle araç tespit başarı oranları ve bir görüntüdeki araçların tespitinin ortalama çalışma zamanı verilmiştir. Sonuçlara göre YOLOv2, araç tespit etme başarısı ve görüntülerdeki ortalama tespit hızı bakımından diğer modellere göre daha iyi performans sergilediği görülmüştür. Kendi oluşturdukları veri setiyle eğitilmiş YOLOv2 modeli %77,12 başarı oranı elde ederken Stanford Drone veri setiyle %68 başarı oranı elde edilmiştir. Bir görüntüde belirlenen sınıf tespiti için ortalama çalışma zamanı kendi veri setleri ile 0,048 saniye olurken Stanford Drone veri seti ile 0,052 saniye olmuştur.

[21]'de insansız hava aracı (İHA) ile araç tespiti için CNN tabanlı derin öğrenme modelleri kullanılmıştır. Dört farklı derin öğrenme mimarisi kullanılarak kapsamlı bir karşılaştırma yapılmıştır. Sistemin çalışmasını sağlamak için önce veri seti oluşturulmuş ve ardından öğrenme modelleri bu veri setiyle eğitilmiştir. Çalışma Intel CPU i5-2520m, Odroid UX4 ve Raspberry Pi 3 platformlarında uygulanmıştır. Sonuçlara bakıldığında DroNet CNN modeli, farklı platformlarda saniyede 5-18 görüntü işleme kapasitesine ve %95 algılama doğruluğuna sahip olduğu görülmüştür.

Bu tez çalışmasının birinci bölümünde tez çalışmasının konusu olan havadan insan, araba, bisiklet ve motosiklet tespitinin önemi ve derin öğrenme konuları kısaca anlatılmıştır. Ayrıca literatürde konu ile ilgili daha önce yapılan diğer akademik çalışmalardan bahsedilmiştir.

İkinci bölümünde CNN'den ve onun çalışma mantığından bahsedilmektedir. CNN'ni oluşturan katmanlar tanıtarak ağı eğitimi sürecinde nasıl bir süreçten geçtiği aktarılmaktadır. Ayrıca ILSVRC yarışmasında başarılı olan CNN modeller hakkında bilgiler verilmektedir.

Üçüncü bölümde öğrenme aktarımı yöntemi hakkında bilgiler verilmektedir. Yöntemin uygulanışında kullanılan ince ayardan (fine tuning) bahsedilmektedir.

Dördüncü bölümde tez çalışması için yapılan işlemler materyal ve metot başlığı altında aktarılmıştır. Bu bölümde modelin eğitilmesi için kullanılan veri setleri ve onların uygun

forma getirilme işlemi anlatılmıştır. Bu bölümde; kullanılan model, derin öğrenme için açık kaynak olan Tensorflow kütüphanesi ve eğitimin yapıldığı Google Colab platformu hakkında bilgiler verilmiştir. Ayrıca eğitimin sonucunda ortaya çıkan sınıflandırma ve regresyon kayıp fonksiyonları ve hata matrisi çıkartılarak modelin performansı ortaya konulmuştur.

Beşinci bölüm de ise tez çalışmasının dördüncü bölümünde elde edilen sonuçların ne tür anlamlar içerdiği açıklanmış ve gelecek çalışmalar için bazı önerilerde bulunulmuştur.



2. KONVOLÜSYON SİNİR AĞLARI (CNN)

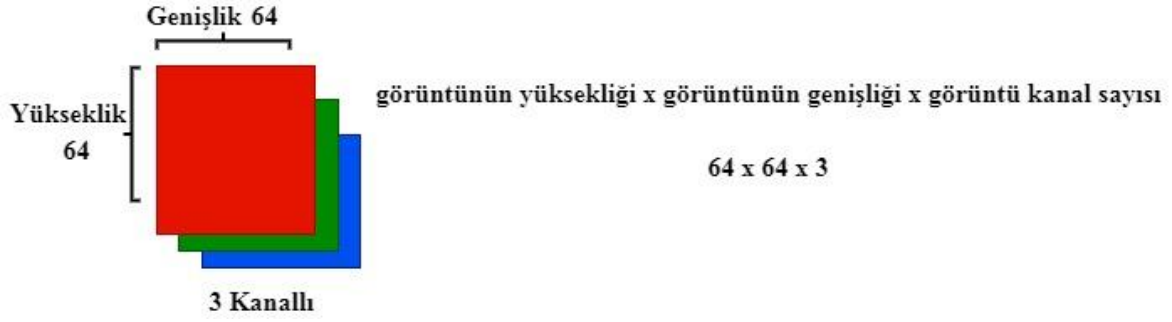
Yapay zekâ, insanların ve makinelerin yetenekleri arasındaki boşluğu kapatırken hızlı bir büyümeye ortaya koymaktadır. Araştırmacılar ve bu alanda ilerlemek isteyen kişiler, yapmak istediği işleri gerçekleştirmek için bu alanın çeşitli yönleri üzerinde çalışmaktadırlar. Bu alanların çoğunun yolu bilgisayarlarla görü alanıyla kesişmektedir.

Bilgisayarla görü alanı, makinelerin dünyayı insanlar gibi görmesini, benzer bir şekilde algılamasını sağlamaktadır. Görüntü ve video tanıma, görüntü analizi ve sınıflandırması, doğal dil işleme gibi çok sayıda görev için kullanılmaktadır. Son zamanlarda derin öğrenme ile bilgisayarlı görme alanındaki gelişmeler CNN üzerine inşa edilmekte ve mükemmelleştirilmektedir. CNN, özellikle görüntü verilerini işlemek için tasarlanmış görüntü tanıma ve işlemede kullanılan çok katmanlı bir yapay sinir ağı türüdür [23].

Sinir ağı, insan beynindeki nöronların çalışmasından sonra şekillendirilen bir donanım ve/veya yazılım sistemidir. Geleneksel sinir ağları görüntü işleme için ideal kullanıma uygun değildir ve düşük çözünürlüklü görüntülerle beslenmektedir. CNN; nöronları, insan beynindeki görsel uyaranları işlemekte sorumlu alan olan frontal lobun yaptığı gibi düzenlemektedir.

CNN modeline giriş olarak verilen görüntüler bazı filtrelerden geçirilerek çıkışa doğru aktarılmaktadır. Konvolüsyon, 1D’de (konuşma işlemede), 2D’de (görüntü işlemede), 3D’de (video işlemede) çalışabilmektedir. 2D konvolüsyon, görüntü işleme alanında özellik çıkarımda kullanılan ve aynı zamanda CNN’nin temel bloğu olduğundan önem arz etmektedir.

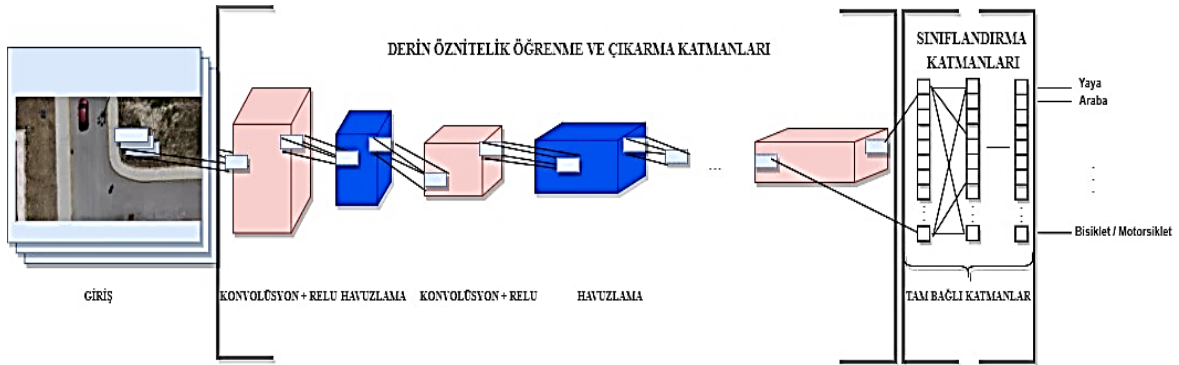
Sinir ağının girişindeki görüntü değişken bir matris olarak düşünülmektedir. Bu matrisin boyutu (görüntünün yüksekliği) x (görüntünün genişliği) x (görüntü kanal sayısı) olarak ifade edilmektedir. Gri tonlamalı bir görüntünün kanal sayısı 1 iken, renkli bir görüntünün (RGB) kanalı sayısı 3 olmaktadır. Şekil 2.1’de 3 kanallı örnek bir RGB görüntüsü bulunmaktadır.



Şekil 2.1. Örnek bir RGB görüntüsü

Görüntü, piksel piksel görüntüde var olan renklerin sayısal değerlerine dönüştürülerek matris formatına çevrilmektedir. Bu şekilde oluşturulan matris görüntüyü sayısal olarak temsil edecek hale gelmektedir. Girişte uygulanan görüntüye belirli filtreler uygulanarak giriş görüntüsünün farklı detayları ortaya konulmaktadır. Kullanılan filtrelerden bazıları şunlardır; bulanıklaştırma filtresi, keskinleştirme filtresi, kabartma filtresi, kenar algılama filtresi vb. filtrelerdir [24]. Kullanılan bu filtreler yardımıyla CNN modelinin derin öz nitelik bilgilerinin çıkartılması sağlanmaktadır.

Şekil 2.2’de CNN mimarisinin girişine uygulanan görüntüde istenilen nesnenin sınıflandırma aşamasını genel bir mimariyle aktarılmaktadır.



Şekil 2.2. CNN mimarisinin genel görünüşü

CNN bir girdi görüntüsünde yer alan görüntüdeki çeşitli nesnelere önem (öğrenilebilir ağırlıklar ve sapmalar) veren ve bir nesneyi diğerinden ayırt edebilen öğrenme algoritmasıdır. CNN’de gereken ön işleme, diğer sınıflandırma algoritmalarına kıyasla çok daha düşüktür.

Etkili CNN mimarilerinin nasıl tasarlanacağını öğrenmek için yapılacak en yararlı iş, başarılı modelleri incelemektir. 2012 yılından itibaren CNN'lerin ILSVRC yarışmasında kullanılmasıyla birlikte bu alanda başarılı çalışmalar ortaya çıkmaktadır. Dev teknoloji şirketlerinin de bu yarışmaya dahil olmasıyla birlikte popülerliği gün geçtikçe artmaktadır. ILSVRC yarışması hem bilgisayar görme görevleri için en son teknolojinin hızla ilerlemesine hem de CNN modelleri mimarisinde genel yeniliklerin gelişmesine neden olmaktadır [3,22].

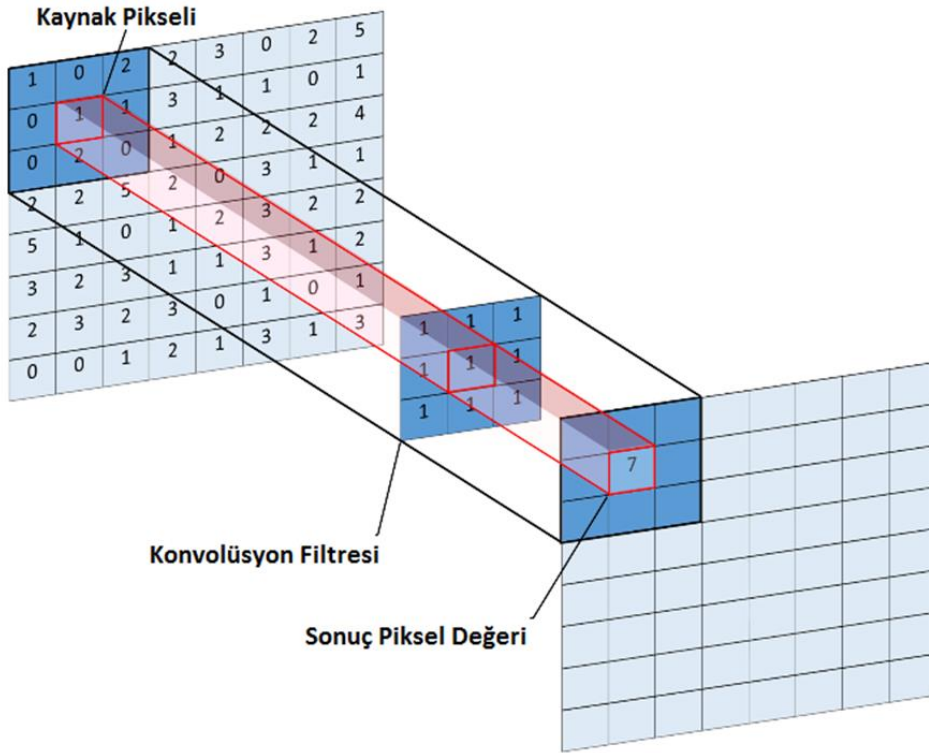
2.1. CNN Mimarisinin Katmanları

Bir sinir ağında, girişe uygulanan görüntü, çoklu nöronlara sahip olan katmanlardan geçirilerek çıkışa aktarılmaktadır. Her bir nöron önceki ve sonraki katmanlardaki tüm nöronlara bağlanmaktadır. CNN modellerinde kullanılan birçok katman bulunmaktadır. Bunlar; konvolüsyon, havuzlama, aktivasyon, düzleştirme (flatten) tam bağlantılı, normalizasyon ve dropout katmanlarıdır [25].

2.1.1. Konvolüsyon katmanı

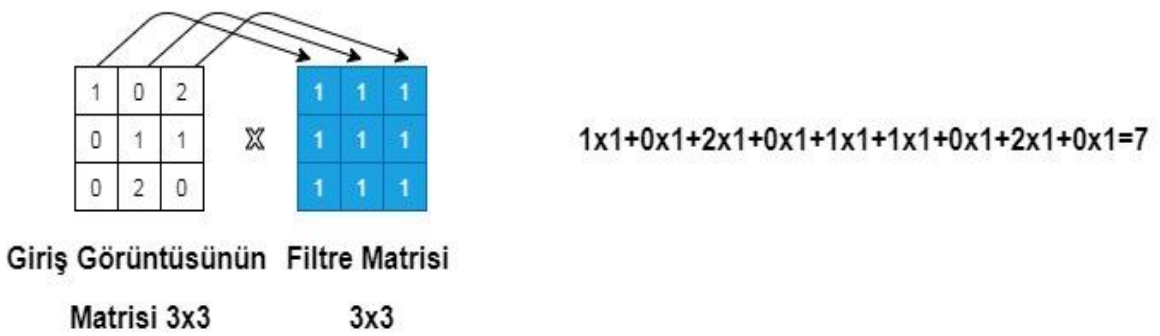
Konvolüsyon katmanı görüntü işlemede, görüntünün sayısal matrisleri üzerinde filtre matrislerin dolaştırılarak çarpılması işlemidir. Bu işlem ile görüntü üzerindeki farklı önemli detayların ortaya çıkarılmasına yardımcı olunmaktadır. Filtre matrisleri yani kernellerin parametreleri, derin öğrenme modellerinde klasik görüntü işlemeden farklı olarak programcı tarafından oluşturulmamaktadır. Oluşturulan CNN model kendi kendine öğrenerek en uygun özellikleri seçmektedir.

Görüntüye konvolüsyon katmanı uygulanırken birden çok filtreden geçirilerek farklı özellikler ortaya çıkarılmaktadır Şekil 2.3'te CNN modelinde bir görüntüye konvolüsyon katmanının uygulanışını göstermektedir.



Şekil 2.3. Bir görüntüye konvolüsyon katmanının uygulanması

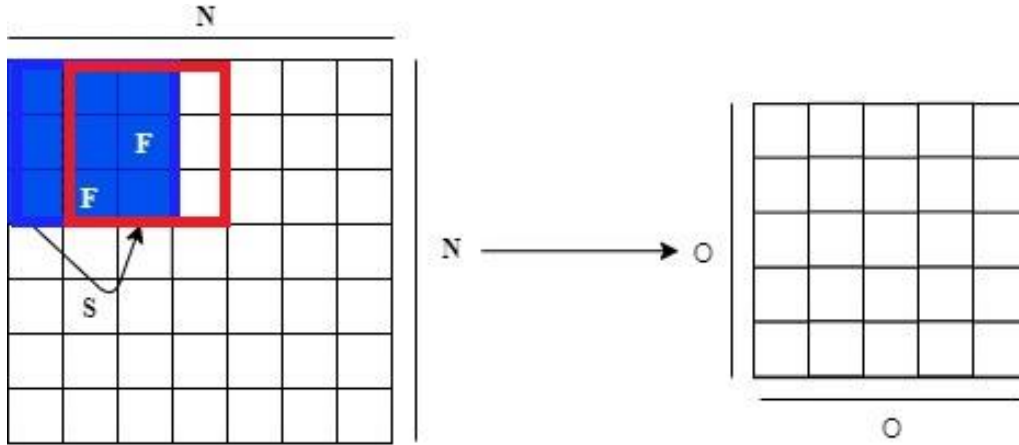
Giriş görüntüsüne uygulanan filtrenin boyutu kadar giriş görüntüsünde matris seçilerek çarpma işlemi yapılır ve toplanır. Sonuç çıkış görüntüsüne aktarılarak çıkış görüntüsü elde edilmektedir. Şekil 2.4'te giriş görüntüsünün bir filtreden geçirilerek çıkış görüntünün bir pikselinin oluşturulması aktarılmıştır.



Şekil 2.4. Giriş görüntüsüne konvolüsyon katmanı uygulanarak çıkış görüntüsünün bir pikselinin elde edilmesi

Atlama aralığı (stride)

Görüntüye filtre uygulanırken atlama aralığı çıkış görüntüsünün oluşturulmasında önem arz etmektedir. Şekil 2.5'te giriş görüntüsünün matrisi ve uygulanan filtre görülmektedir. Örnek olarak 7x7 boyutunda giriş görüntüsü ve 3x3 boyutunda filtre ele alınmaktadır.



Şekil 2.5. Giriş görüntüsünün ve uygulanacak filtrenin matris gösterimi

Çıkış görüntüsü denklemi;

$$O = \frac{N-F}{S} + 1 \quad (2.1)$$

Eş. 2.1 ile ifade edilmektedir. Çıkış görüntüsünün sonucu bir kare matris olarak tanımlanmıştır. Çıkış görüntüsünün boyutu $O \times O$ boyutunda bir kare matris, giriş görüntüsü $N \times N$ 'lik bir kare matris, uygulanacak olan filtrenin boyutu $F \times F$ 'lik bir kare matris, filtrenin görüntüde uygulanırken atlama aralığı ise S ile ifade edilmektedir.

Atlama aralığının değişmesiyle çıkış görüntüsünün boyutu da değişmektedir.

Çıkış görüntüsü;

$S=1$ iken; 5×5 'lik kare matris

$S=2$ iken; 3×3 'lük kare matris

Eş. 2.1 ile belirlenmektedir.

Sıfır ekleme (zero padding)

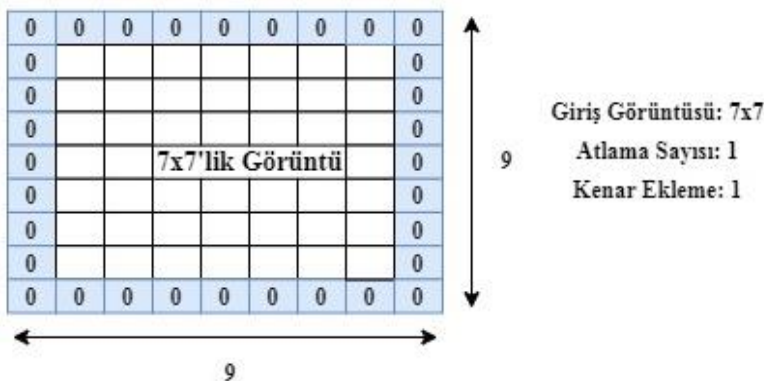
Giriş görüntü matrisine direkt filtre uygulanması durumunda görüntünün kenarları ihmal edilmiş olmaktadır. Görüntünün kenar piksellerinde önemli bilgiler içerebileceğinden bu istenmeyen bir durumdur. Bunun önüne geçmek için kenar ekleme işlemi uygulanmaktadır. Kenar ekleme işlemi için görüntünün kenarlarına sıfır eklemesi (padding) yapılmaktadır. Kenar ekleme yöntemi uygulanmadığında orjinal görüntünün boyutu azalmakta ve yine aynı şekilde bu süreç devam etmektedir. Görüntüye konvolüsyon katmanı uygulandıktan sonra elde edilecek matris boyutunun başlangıçtaki boyutuyla aynı olması istenirse yine kenar ekleme yöntemi uygulanması gerekmektedir. İlk görüntüyle çıkış görüntüsünün boyutunu aynı olması istendiğinde eklenecek olan kenar sayısı Eş. 2.2 ile bulunmaktadır. Sıfır kenar dolgu sayısı,

$$P = \frac{F-1}{2} \quad (2.2)$$

Kenar dolgu sayısı filtre sayısına bağlı olarak değişmekte olup Eş. 2.2. ile belirlenmektedir. F , filtre sayısı ifade etmektedir. Eklenecek olan kenar sayısı da işleme sokularak çıkış görüntüsünün boyutu Eş. 2.3'teki denklem ile belirlenmektedir.

$$\text{Çıkış Görüntüsü} = \frac{N-F+2P}{S} + 1 \quad (2.3)$$

Bir katmana giriş yapan görüntüsünün matrisine bir filtre uygulandığında çıkış görüntüsünün matris boyutunun elde edilmesi için genel denklem olarak Eş. 2.3 kullanılmaktadır. Şekil 2.6'da 7x7'lik matrise sıfır kenar eklemesi gösterilmektedir.



Şekil 2.6. Kenarlara sıfır ekleme yöntemi gösterimi

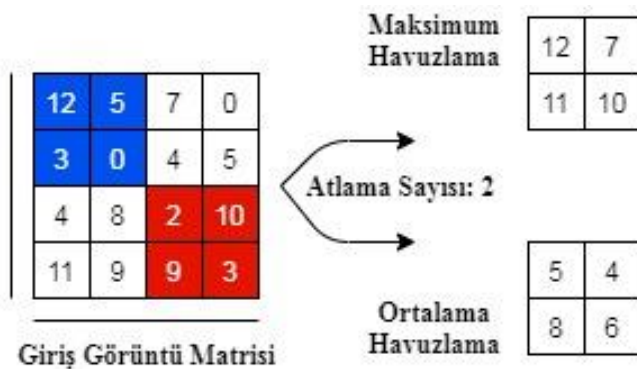
Şekil 2.6'daki 7x7 boyutunda görüntüye 3x3'lük filtre uygulaması gerçekleştirildiğinde çıkış görüntüsünün boyutları Eş. 2.3'ten hesaplanmaktadır. Böylece kenarlara sıfır ekleme yöntemi ile çıkış görüntüsünün matris boyutu başlangıçtaki görüntünün boyutuyla aynı olması sağlanabilmektedir.

2.1.2. Havuzlama katmanı (Pooling)

Havuzlama katmanı olarak bilinen bu katman genellikle konvolüsyon işlemi ardından tercih edilen bir katmandır [26]. Aynı zamanda aşağı örnekleme katmanı olarak da adlandırılmaktadır. Havuzlama katmanında bir önceki katman olan konvolüsyon katmanında gerçekleştiği gibi gelen görüntü üzerinde değişik boyutlarda filtreler gezdirilerek işlemler yapılmaktadır. Bu katmandaki filtrelerin konvolüsyon katmanında uygulanan filtrelerden farkı ise maksimum veya ortalama alan filtreler olmasındandır. Yeni oluşan görüntü matris boyutu orijinal görüntünün matris boyutundan daha küçük olmaktadır. Böylelikle ağdaki işlenen veri miktarı da düşürülmüş olmaktadır. Havuzlama katmanında kullanılan en popüler iki tür bulunmaktadır. Bunlar maksimum ve ortalama havuzlamadır.

Maksimum havuzlamada görüntüde en baskın karakter hangisiyse o söz sahibi olmaktadır. Görüntü matrisinde herhangi belirgin bir özelliğin varlığı güçlü ise maksimum havuzlamada tespit edilmektedir.

Ortalama havuzlamada ise görüntüde baskın bir noktanın olması bu katman için çok önemli değildir. Görüntüde herhangi bir özellik belirgin olsa bile diğer özellikler de dikkate alınarak ortalama hesaplanmaktadır. Şekil 2.7'de ortalama ve maksimum havuzlama katmanının uygulanışı gösterilmektedir.



Şekil 2.7. Giriş görüntüsüne uygulanan maksimum ve ortalama havuzlama katmanı

Maksimum havuzlamada ilgili matrisin en büyük değeri çıkışa aktarılırken ortalama havuzlama matrisin ortalaması alınarak çıkışa aktarılmaktadır. Ortalama değer çoğu durumda görüntüde olmayan değerlere karşılık geldiği için çok sık tercih edilmemektedir. Maksimum havuzlamada matrislerin en büyük değerleri en önemli piksellere karşılık geldiği için çok sık tercih edilen metot olarak karşımıza çıkmaktadır.

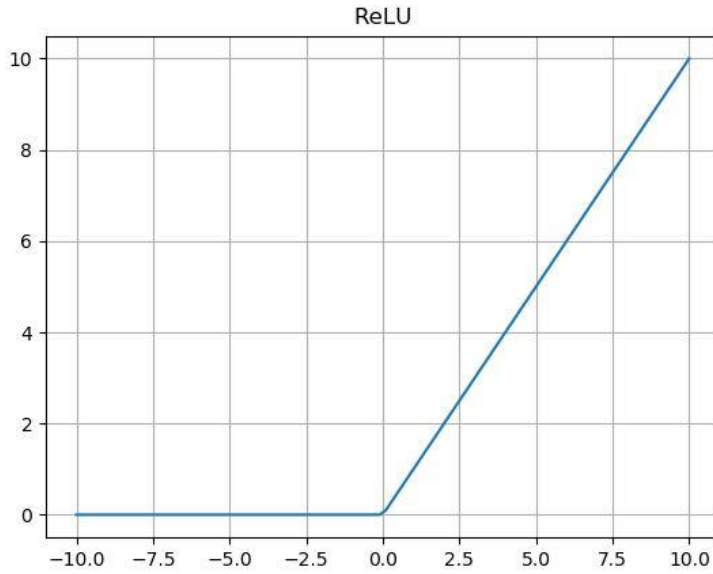
2.1.3. Aktivasyon katmanı

Aktivasyon katmanları sinir ağlarında olmazsa olmaz katmanlar arasında kabul edilen bir diğer katmandır. Bu katman genellikle düzleştirme (flatten) veya konvolüsyon katmanlarından sonra kullanılmaktadır. Aktivasyon katmanları nöronların çıktısını düzenlemeye yardımcı olmaktadır. Sinir ağlarında birçok farklı aktivasyon fonksiyonu kullanılmaktadır. Farklı sinir ağlarında farklı aktivasyon fonksiyonları tercih edilebilmektedir. Belli başlı aktivasyon fonksiyonları; doğrultulmuş lineer birim (rectified linear units, ReLU), sigmoid, tanh ve sızıntılı doğrultulmuş lineer birim (leaky ReLU) fonksiyonlarıdır.

Doğrultulmuş lineer birim aktivasyon fonksiyonu (ReLU)

CNN modellerinde en çok tercih edilen aktivasyon fonksiyonudur. Fonksiyonda negatif değerler 0'a dönüştürülerek eğitim yapılmaktadır. Bundan dolayı fonksiyonun kullanımı negatif değerlerin fazla olduğu modelde tavsiye edilmemektedir. Şekil 2.8'de ReLU aktivasyon fonksiyon grafiği, Eş. 2.5'te ReLU aktivasyon fonksiyon denklemi gösterilmektedir.

$$f(x) = \begin{cases} 0 & \text{if } x < 0 \\ x & \text{if } x \geq 0 \end{cases} \quad (2.5)$$

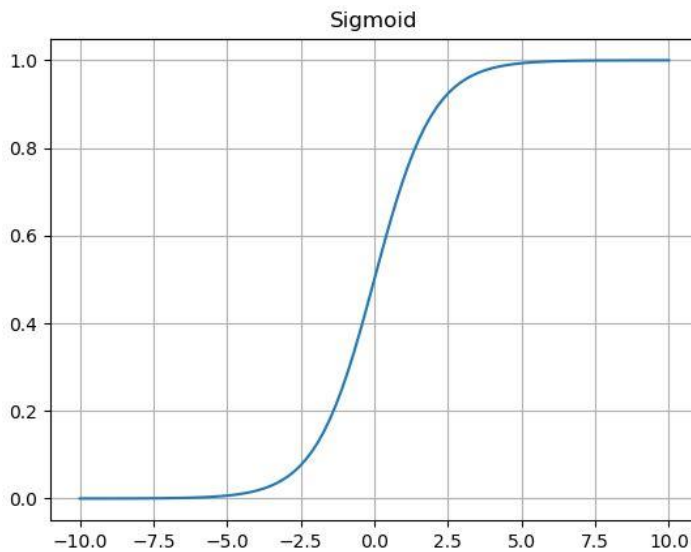


Şekil 2.8. ReLU aktivasyon fonksiyonu

Sigmoid fonksiyonu

Sigmoid fonksiyonu ReLU fonksiyonuna göre az tercih edilmektedir. Çünkü sinir ağlarının eğitimi esnasında gradyanların kaybolmasına neden olmaktadır. Sinir ağları kısa bir sürede öğrenemez hale gelmektedir. Fonksiyonun yaptığı işlem, giriş değerini 0 ile 1 arasına getirmektir. Sigmoid fonksiyon denklemi Eş. 2.6'da fonksiyon grafiği de Şekil 2.9'da aktarılmıştır.

$$f(x) = \frac{1}{1+e^{-x}} \quad (2.6)$$

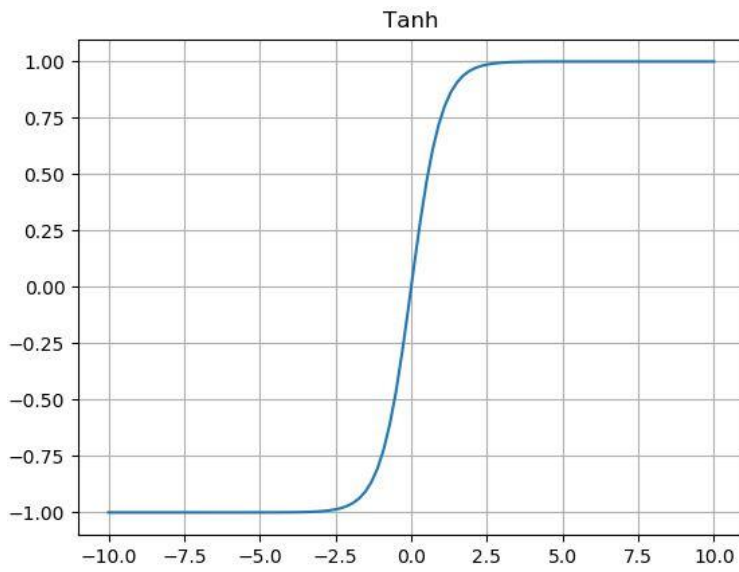


Şekil 2.9. Sigmoid aktivasyon fonksiyonu

Tanh fonksiyonu

Tanh fonksiyonu sigmoid fonksiyonunun sıfır eksenini etrafında simetrik hale getirilmesiyle oluşturulmuştur. Sigmoid fonksiyonun göre simetrik olmasından dolayı daha çok kullanışlıdır fakat bu fonksiyon da pek tercih edilen bir fonksiyon değildir. Tanh fonksiyonu Eş. 2.7’de, fonksiyon grafiği ise Şekil 2.10’da aktarılmıştır.

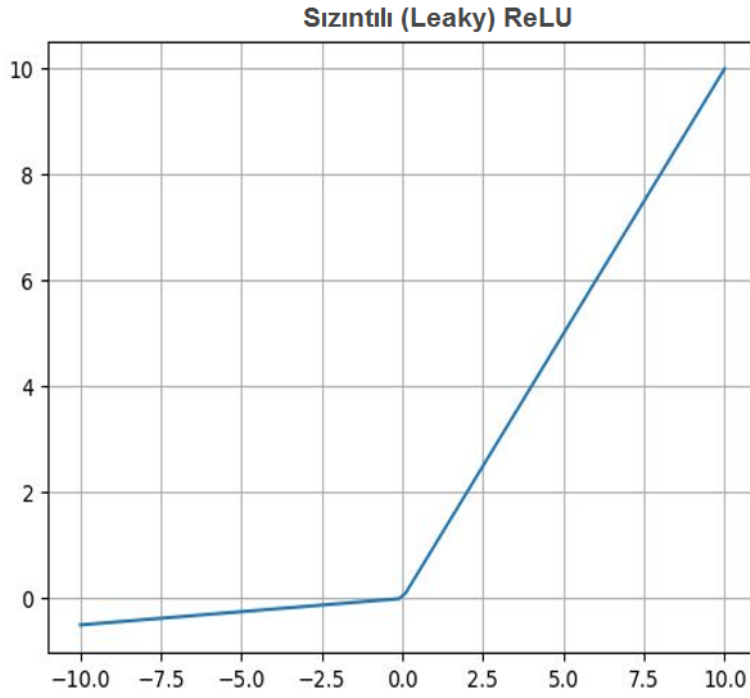
$$f(x) = \tanh(x) = \frac{2}{1+e^{-2x}} - 1 \quad (2.7)$$



Şekil 2.10. Tanh aktivasyon fonksiyonu

Sızıntılı (leaky) ReLU fonksiyonu

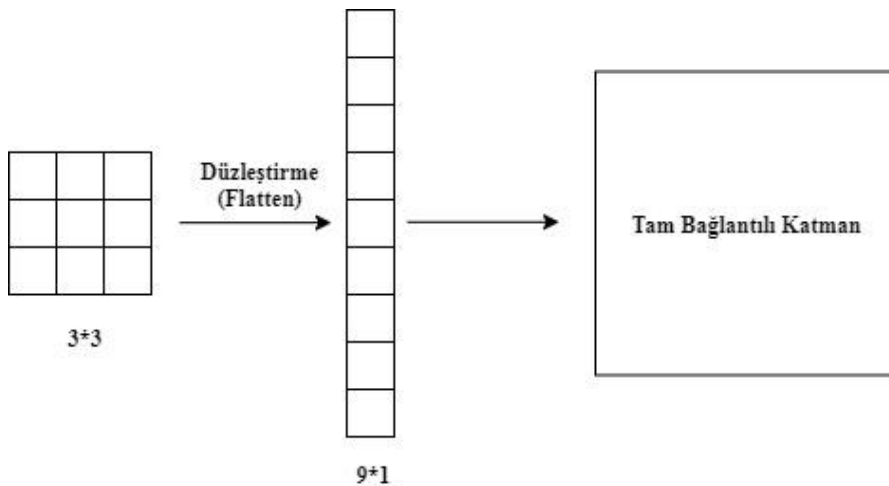
ReLU fonksiyonu diğer fonksiyonlara göre daha avantajlıdır. Sigmoid ve tanh fonksiyonlarına göre yapay sinir ağının çok hızlı yakınsayıp kısa zamanda yüksek performans elde edilmesini sağlamaktadır. Fakat kendi içerisinde bu fonksiyonun da bazı eksiklikleri vardır. Bunlardan en önemlisi eğitim esnasında nöronun ölmesidir. Bu da eğitim sırasında istenmeyen bir durumdur ve kayıpların artmasına sebebiyet vermektedir. Bu dezavantajı ortadan kaldırmak için sızıntılı ReLU fonksiyonu önerilmiştir. Bu fonksiyon ile negatif değerler sıfır değerine değil de çok küçük değerlere çevrilmektedir. Şekil 2.11’de sızıntılı ReLU fonksiyonun grafiği gösterilmektedir.



Şekil 2.11. Sızıntılı (Leaky) ReLU aktivasyon fonksiyonu

2.1.4. Düzleştirme katmanı (Flatten)

Düzleştirme katmanının görevi, sinir ağının en sonunda yer alan ve önemli bir katman olan tam bağlantılı katmanının girişindeki verileri hazırlamaktır. Bu sinir ağındaki veriler ise konvolüsyon ve havuzlama katmanından gelen matrislerin tek boyutlu diziye çevrilmiş halidir. Kısaca bu katman matris formatındaki verileri düzleştirmek için kullanılmaktadır. Düzleştirme, bir araya toplanan özellik haritası matrisinin tek bir sütuna dönüştürülmesini içerir. Düzleştirme işlemi basitçe Şekil 2.12’de gösterilmektedir.



Şekil 2.12. Düzleştirme (Flatten) katmanının uygulaması

2.1.5. Tam bağlantılı katman

Yapay sinir ağlarında gizli katman olarak adlandırılırken konvolüsyon sinir ağlarında tam bağlantılı katman olarak kullanılmaktadır. Sinir ağının en sonunda yer alan tam bağlantılı katmanın girişi düzleştirme katmanının çıkışıdır. Yani bu katmana giriş olarak tek vektörlü bir matris verilmektedir. Bu katmanda özellik analizinden girdiler alınarak doğru etiketi tahmin etmek için ağırlıklar uygulanmaktadır. Tam bağlı katmanın çıkışında tespit edilecek sınıflar aktarılmaktadır.

2.1.6. Normalizasyon katmanı

Derin öğrenme mimarilerinde yığın normalleştirme (batch normalization) genellikle bir katmandan hemen sonra aktivasyon fonksiyonu çağrılmadan önce kullanılmaktadır. Buradaki amaç çıktıları normalize ederek regularizasyon sağlamak ve yapay sinir ağının aşırı öğrenmesini engellemektir [27]. Bu katman CNN'nin eğitimi sırasında yok olma gradyanına direnç vermektedir. Böylece modelin eğitim süresi azalmakta ve model daha iyi performans göstermektedir. Çalışmada yığın normalleştirme işleminin model başarısına etkisi gösterilmektedir [28].

2.1.7. Dropout katmanı

Derin öğrenme katmanlarında öğrenimi gerçekleştirilmiş bir ağda aşırı öğrenmeyi yani ezberlemeyi (overfit) ortadan kaldırmak için kullanılan bir katmandır. Ağın ezberlemesi engellenerek ağın çalışma performansını artırmaktadır. Bu katman her iterasyonda belirli bir oranda nöronun rastgele ihmal edilmesini sağlamaktadır. Dropout 0 ile 1 arasında değer almaktadır. Belirlenen değer ile katmanın her seferinde farklı nöronları aktif olmaktadır. [29,30].

2.2. CNN Mimariler

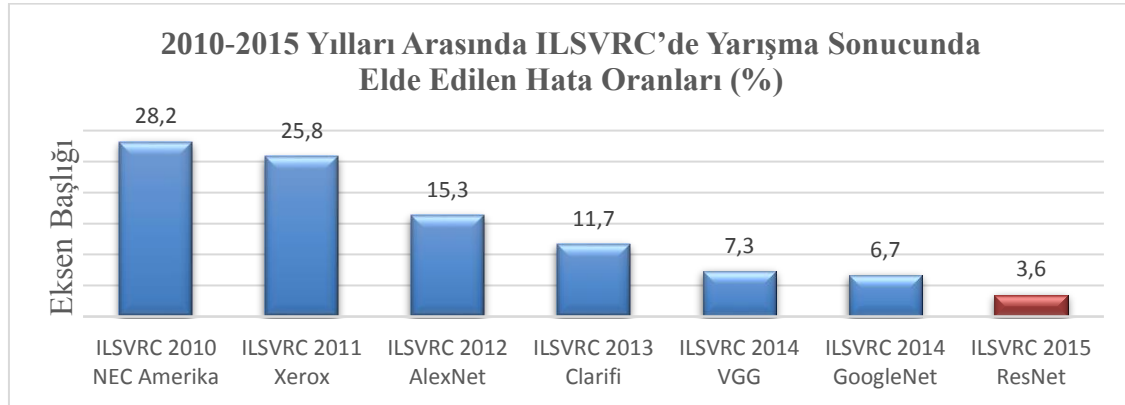
Bir CNN'nin mimarisi, performansını ve verimliliğini belirlemede önemli bir faktördür. Katmanların yapılandırılma şekli, katmanlarda hangi öğelerin kullanıldığı ve bunların nasıl tasarlandığı genellikle çeşitli görevleri gerçekleştirebilme hızını ve doğruluğunu etkilemektedir.

ImageNet yarışması

ImageNet, görsel nesne tanıma yazılımının araştırılmasında kullanılmak üzere tasarlanmış görsel bir veri tabanıdır. ImageNet, CNN'yi nesne algılama konusunda eğitmek için özel olarak tasarlanmış 14 milyondan fazla görüntüye sahiptir.

2010 yılından itibaren, proje ILSVRC adlı her yıl bir yarışmaya ev sahipliği yapmaktadır. Katılımcılar, verilen görüntülerdeki nesneleri doğru bir şekilde algılamaya ve sınıflandırmaya çalışan yazılım programlarıyla yarışmaya katılım sağlamaktadır. CNN mimarileri bu yarışma ile tüm dünyaya tanıtılmıştır. 2012’de CNN modellerinin ImageNet yarışmasına dahil edilmesiyle bu alanda başarılı modellerin ortaya çıkma süreci hızlanmıştır. Yarışmada katılan modellerin karşılaştırılması için hata oranları dikkate alınmaktadır. İlk 5 hata (top-5 error) oranı terimi, ImageNet yarışmasında öğrenme modellerini karşılaştırma yöntemini ifade etmektedir. Hedef etiket, modelin ilk 5 tahmininden biriye, modelin belirli bir görüntüyü doğru bir şekilde sınıflandırdığı kabul edilmektedir.

Şekil 2.13’te ImageNet yarışmasında yıllara göre birinciliği elde eden CNN modelleri ve onların ilk-5 hata oranları birlikte verilmiştir.

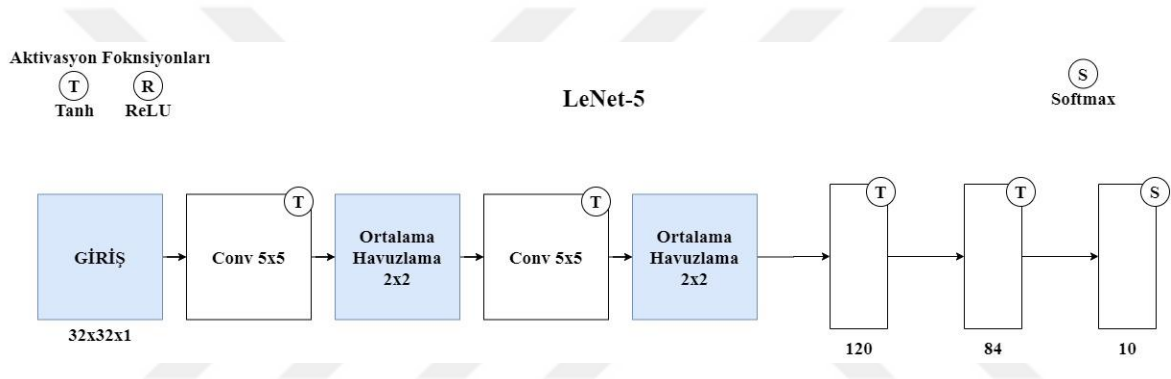


Şekil 2.13. ImageNet yarışmasında birinci olan CNN modelleri ve top-5 hata oranları

Genellikle ILSVRC'den önce CNN'lerin ilk başarılı ve önemli uygulaması olarak tanımlanan LeNet-5 kabul edilmektedir. Daha sonra ILSVRC için geliştirilen CNN modelleri (AlexNet, VGG, Inception ve ResNet) literatüre katkı sağlamaktadır.

2.2.1. LeNet-5 mimarisi

Yaygın olarak bilinen CNN'nin ilk başarılı uygulaması olarak LeNet-5 kabul edilmektedir. Sistem, elle yazılmış bir karakter tanıma probleminde kullanılmak üzere geliştirilmiştir. MNIST standart veri kümesinde test edilmiş ve yaklaşık %99,2 sınıflandırma doğruluğu elde edilmiştir. Çalışma MNIST veri kümesindeki giriş görüntülerin boyutu olan 32x32 şeklinde gri tonlamalı görüntülere sahiptir. Model toplam 7 katmandan oluşturulmuştur. Model, bir konvolüsyon katmanı ve ardından alt örnekleme katmanı olarak adlandırılan ortalama havuzlama katmanı önermektedir [31]. Şekil 2.14'te LeNet-5 modelinin mimarisi verilmiştir.

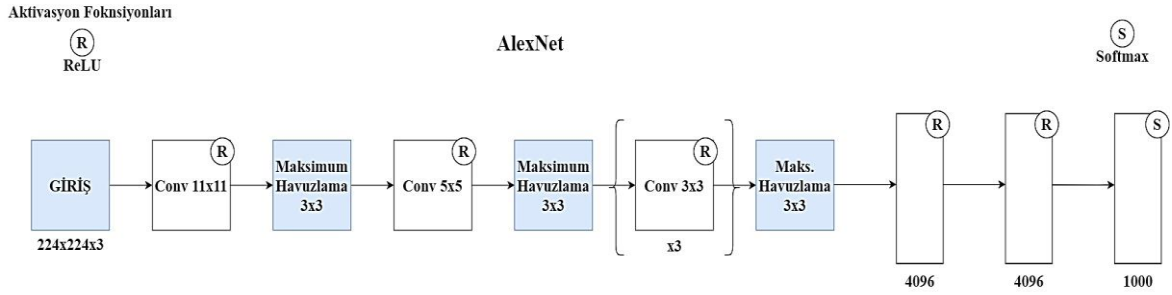


Şekil 2.14. LeNet-5 CNN mimarisi

Mimari ilk gizli katman olarak az sayıda filtre kullanmaktadır. Özellikle her biri 5x5 piksel boyutunda altı filtre bulunmaktadır. Havuzlama işleminden sonra başka bir konvolüsyon katmanını mimaride karşımıza çıkmaktadır. Önceki konvolüsyon katmanından daha küçük bir boyuttadır ve 5x5 piksel boyutunda 16 filtreye sahiptir.

2.2.2. AlexNet mimarisi

Alex Krizhevsky, Ilya Sutskever ve Geoffrey Hinton tarafından ortaya konulan derin öğrenme algoritması olan AlexNet 2012 yılında literatüre kazandırılmıştır [32]. ImageNet yarışmasında AlexNet modelinin geliştirilmesinden önceki görevler bilgisayar görüş yöntemlerinin oldukça uzağındaydı. AlexNet, alandaki CNN modelinin kapasitesini başarılı bir şekilde gösterdi ve daha sonraki yıllarda ILSVRC yarışmasında gösterilen daha fazla iyileştirme ve yenilikle sonuçlanan bir ateş yaktı. Modelde giriş görüntüsü olarak 224x224 boyutunda üç renk kanallı görüntüler kullanılmıştır. Şekil 2.15'te AlexNet modelinin mimarisi verilmiştir.



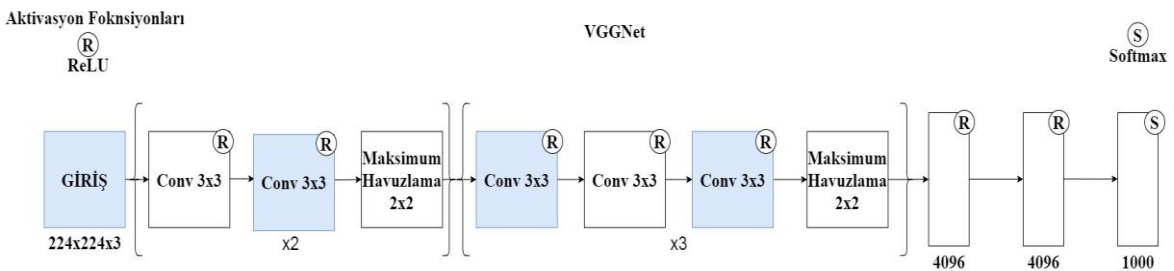
Şekil 2.15. AlexNet CNN mimarisi

AlexNet modelinin temel yönleri şu şekilde özetlenebilir;

- Konvolüsyon katmanından sonra ve ReLU aktivasyon fonksiyonu kullanılmıştır
- Ortalama havuzlama yerine maksimum havuzlama yöntemi tercih edilmiştir.
- Tamamen bağlı katmanlar arasına aşırı öğrenmeyi engellemek için dropout düzenleyici kullanılmıştır.

2.2.3. VGGNet mimarisi

VGG mimarisi adını çalışmaların yapıldığı Oxford'daki Visual Geometry Group laboratuvarının baş harflerinden almaktadır. Özellikle, LeNet-5'teki büyük boyutlu filtrelerden ve AlexNet'teki daha küçük ama yine de nispeten büyük filtrelerden farklı olan, 3x3 ve 1x1 boyutunda filtreler kullanılmıştır. Önceki modellerden farklı olarak daha fazla sayıda filtre kullanılmıştır. Model 13 konvolüsyon katmanı ve 3 tam bağlı katmandan oluşmaktadır. Bu modelde AlexNet'in aktivasyon fonksiyonu olarak kullanmış olduğu ReLU aktivasyon fonksiyonunu kullanmaya devam edilmiştir. ImageNet veri tabanında %89 doğruluk yakalamış bir derin öğrenme algoritmasıdır [33,34]. Şekil 2.16'da VGGNet modelinin mimarisi verilmiştir.



Şekil 2.16. VGGNet CNN mimarisi

VGGNet modelinin temel yönleri şu şekilde özetlenebilir;

- 3x3 ve 1x1 gibi çok küçük konvolüsyon filtreleri kullanılmıştır.
- 2x2 boyutlarında maksimum havuzlama adımları vardır.
- Konvolüsyon + havuzlama bloğunun tekrar etmektedir.
- VGG-16 ve VGG-19 adında sırasıyla 16 ve 19 katmanlı iki farklı modeli bulunmaktadır.

2.2.4. Inception mimarisi

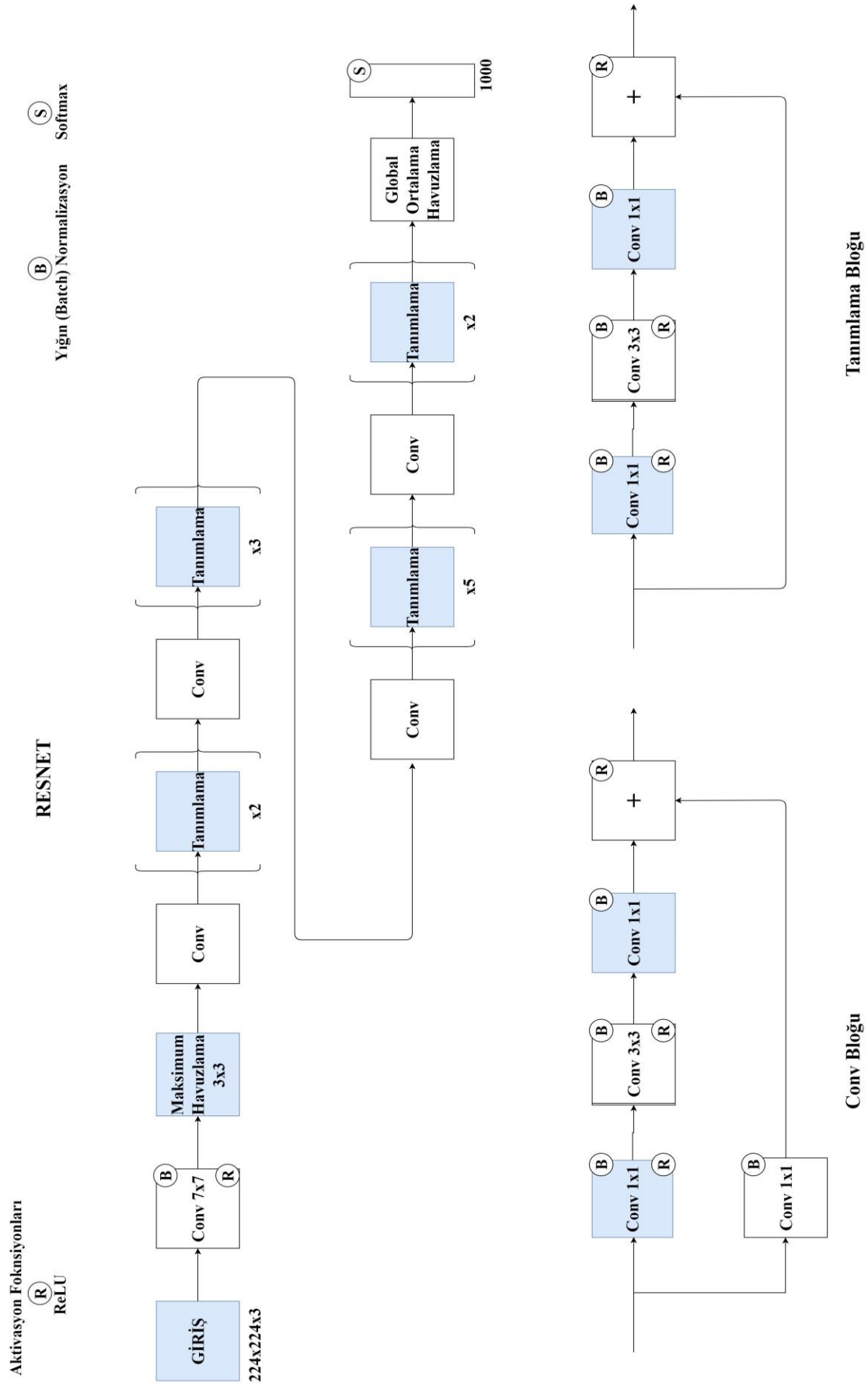
AlexNet modeline göre 12 kat daha az parametreye sahip olan Inception modeli 22 katmandan (havuzlama katmanları da dahil edildiğinde 27 katman) oluşmaktadır. Inception modelinin birden çok versiyonu bulunmaktadır. Fakat Inception-v1 modeli ILSVRC14 yarışmasında GoogleNet'i takım ismi olarak kullanılmıştır. R-CNN yöntemi nesne tanıma işleminde Inception-v1 modelinden faydalanmaktadır [35]. Şekil 2.17'de Inception modelinin mimarisi verilmiştir.

Inception modelinin temel yönleri şu şekilde özetlenebilir;

- Boyut küçültme ve ReLU için 1x1 boyutunda 128 filtre kullanılmıştır.
- Toplamda 5M parametreye sahiptir.
- Çıkışında %70 oranında düşürülmüş dropout katmanı bulunmaktadır.
- Bir tam bağlı katman ile 1024 elemanlı vektör oluşturulmaktadır.

2.2.5. ResNet mimarisi

Residual Network olarak bilinen ResNet modeli ILSVRC 2015 yarışmasında birinciliği elde etmiştir. Bu modelde AlexNet, VGGNet, Inception modellerinin kullandığı ardışık yapıdan farklı bir yapı kullanılmıştır. Diğer modellere göre ResNet ağı daha derin özelliğe sahiptir. Derinliğin artmasıyla birlikte hedef fonksiyona daha iyi yaklaşılr ve daha yüksek ayırt edici güç ile daha iyi öznitelik sunumu oluşturulmaktadır. 2015 yılında yayınlanan makalede belirtildiği üzere 152 katmanlı ResNet modeli o ana dek ImageNet üzerinde kullanılan en derin ağ özelliğini taşımaktadır. ResNet'te eğitim hatasını azaltmak için çok sayıda katman içeren artık bloklar kullanılmıştır. ResNet CNN modeli genel olarak; konvolüsyon katmanı, aktivasyon katmanı, havuzlama (alt örnekleme) katmanı ve tamamen bağlı katman gruplarından meydana gelmektedir [36]. Şekil 2.18'de ResNet modelinin mimarisi verilmiştir.



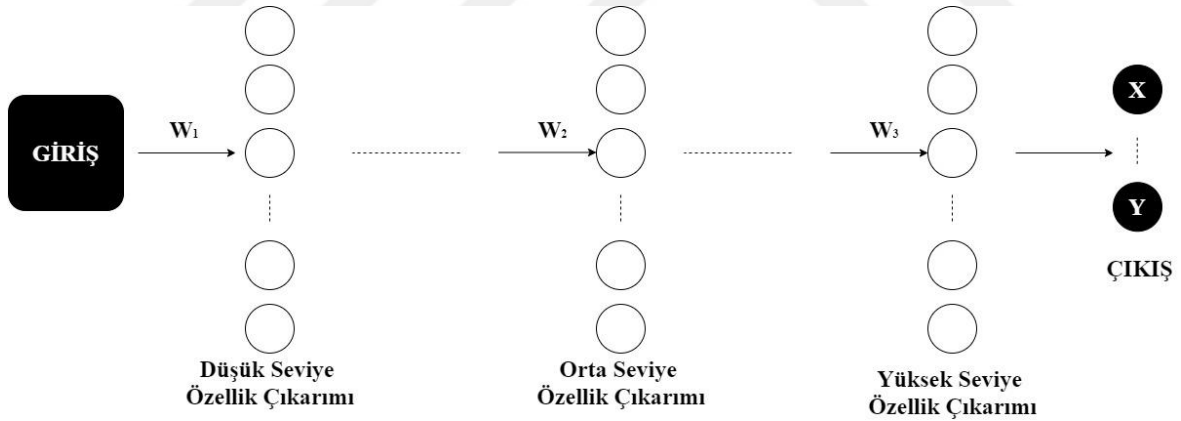
Şekil 2.18. ResNet CNN mimarisi

ResNet modelinin temel yönleri şu şekilde özetlenebilir;

- Model girişine 224x224x3 boyutunda görüntüler uygulanmaktadır.
- Modelde hata oranı insanın hata eşiği olan %5-10'nun altına gelerek %3,57 olarak belirlenmiştir.
- ResNet modeli CNN ağlarının performans düşümü (degradation) problemini çözme hedeflemektedir.
- Bilgilerin gözden kaybolma (vanishing gradient) problemini çözmek için ResNet modeli geliştirilmiştir.

Bilgilerin gözden kaybolması (vanishing gradient)

Derin öğrenme işlemlerinde CNN ağında derinlik arttıkça öğrenilen özelliklerde bazı kaybolmalar ortaya çıkmaktadır. Yani kısaca derinlik arttıkça öğrenme başarısı artmamaktadır. Bir ağın derinliği sistem donanımına ve veri setine bağlıdır. Şekil 2.19'da çok katmanlı derin bir CNN modeli gösterilmektedir.



Şekil 2.19. Derin bir CNN modeli

Şekil 2.19'da gösterilen girişe uygulanan görüntüde bazı sınıfların sınıflandırılmasını sağlayan derin CNN model gösterilmiştir. Girişe yakın katmanlarda görüntünün sınıflandırma için düşük seviyede özellik çıkarımı yapılabilmektedir. Çıkarılan düşük seviye özelliklerden bazıları köşeler kenarlar, dikey ve yatay çizgilerdir. Modelin orta katmanlarında biraz da üst seviye özellikler çıkartılmaktadır. Sonlara doğru da sınıflandırılacak olan nesnenin belli başlı özellikleri çıkartılmaktadır.

CNN derinleştikçe düşük seviye özelliklerin çıkarımında bilgi kaybı denilen vanishing gradient olayı olmaktadır. Gradient genel olarak;

$$\text{Gradient} = \frac{\partial \text{loss}}{\partial w} \quad (2.8)$$

Eş. 2.8 ile hesaplanmaktadır. Burada w , ağırlıklara, loss ise kayıpların karşılık gelmektedir. Sinir ağlarında öğrenilen şey aslında ağırlıklardır. Başlangıçta belirlenen ağırlıklar modelde katmanlardan geçerek ve belirli bir kayıp hesaplanarak güncellenmektedir.

Güncel ağırlık;

$$w = w - l \times \text{gradient} = w - l \times \frac{\partial \text{loss}}{\partial w} \quad (2.9)$$

Eş. 2.9 ile belirlenmektedir. Eş. 2.9'da l , öğrenme oranına karşılık gelmektedir. Buna göre kaybın ağırlığa göre türevinin hesaplanmasında sinir ağının derinliği ön plandadır. Derinlik arttıkça ilk ağırlıkların gradientin değeri oldukça düşük olacaktır. Gradientin ağırlıklar ile hesaplanması

$$\frac{\partial \text{loss}}{\partial w_1} = \frac{\partial \text{loss}}{\partial w_n} \times \frac{\partial w_n}{\partial w_{n-1}} \times \frac{\partial w_{n-1}}{\partial w_{n-2}} \dots \frac{\partial w_2}{\partial w_1} \quad (2.10)$$

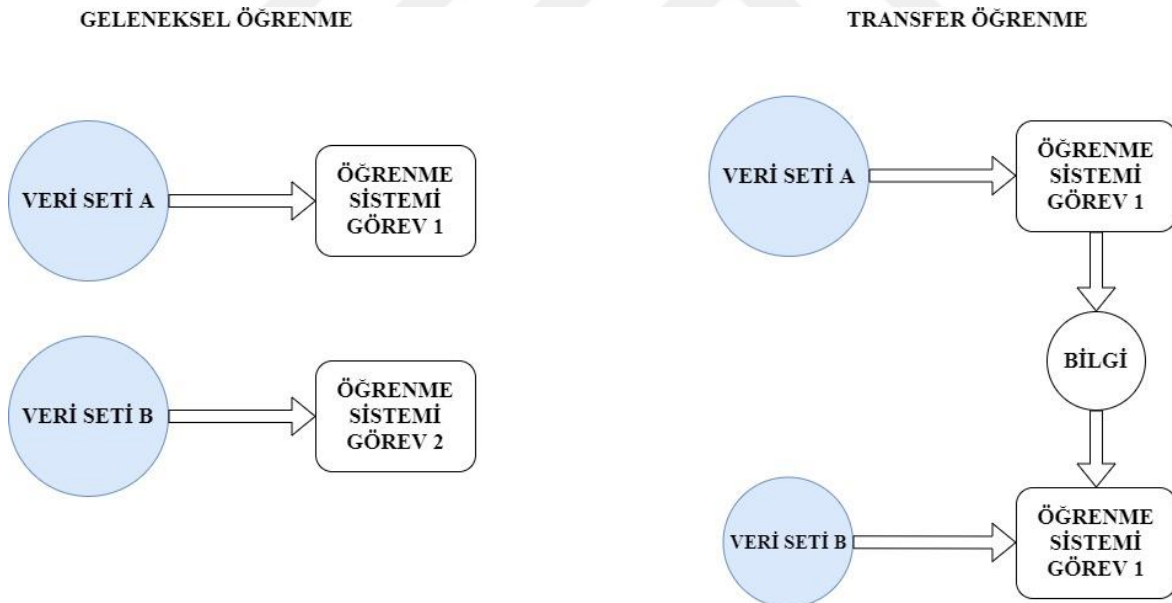
Eş. 2.10 ile belirlenmektedir. Düşük değerde olan gradientin öğrenme oranıyla çarpılıp ilk ağırlık değerinden çıkartılmasıyla güncelleniş ağırlık değeri bulunmaktadır. Derin ağlarda öğrenme oranının gradient ile çarpımı sonuncunun oldukça düşük değerde kalmasından dolayı güncellenmiş olan ağırlık değeri yaklaşık olarak ilk ağırlığa eşit kalmış olacaktır. Bu durumda ağırlık daha önceden öğrendiğini güncellenmemiş olacaktır. Sinir ağında bu işlemin en çok etkilediği kısım, ağırlık giriş kısmına yakın olan ağırlıklardır.



3. ÖĞRENME AKTARIMI (TRANSFER LEARNING) YÖNTEMİ

Öğrenme aktarımı aynı insanlar gibi bir problemi çözerken elde ettiği bilgiyi saklayıp, başka bir problem ile karşılaştığında o bilgiyi kullanan makine öğrenme tekniğidir. Öğrenme aktarımı ile daha önceden elde edilen bilgiler kullanılarak daha az eğitim seti ile yüksek oranda performans elde edilebilmektedir. Derin öğrenme modellerini eğitmek için gereken büyük kaynaklar veya derin öğrenme modellerinin eğitildiği büyük ve zorlu veri setleri göz önüne alındığında, öğrenme aktarımı derin öğrenmede oldukça popüler bir alan olarak karşımıza çıkmaktadır.

Öğrenme aktarımında, önce bir görevi gerçekleştirmek için temel veri kümesi ile bir temel ağ eğitilmektedir. Ardından öğrenilen özellikler yeniden hedef veri kümesi ve görev üzerinde eğitilerek ikinci bir hedef ağa aktarılmaktadır. Şekil 3.1’de geleneksel öğrenme yöntemiyle transfer öğrenme yönteminin basit bir çalışma süreci gösterilmektedir.



Şekil 3.1. Geleneksel öğrenme ve transfer öğrenme yöntemlerinin genel gösterimi

Geleneksel öğrenme tamamen belirli görevlere yönelik çalışmaktadır. Bir model belirlenen veri setiyle eğitilip bir görev için kullanılmaktadır. Bir modelden diğerine aktarılabilecek hiçbir bilgi yoktur. Transfer öğrenmede yani öğrenme aktarımında ise yeni modelleri eğitmek için daha önce eğitilmiş modellerden bilgiler (özellikler, ağırlıklar vb.) kullanılabilir. Ayrıca yeni görevi gerçekleştirmek için daha az veri setine ihtiyaç duyulmaktadır [38].

Öğrenme aktarımı için iki ortak yaklaşım vardır. Bunlardan biri gelişmiş model yaklaşımı diğeri ise önceden eğitilmiş model yaklaşımıdır.

Gelişmiş model yaklaşımı

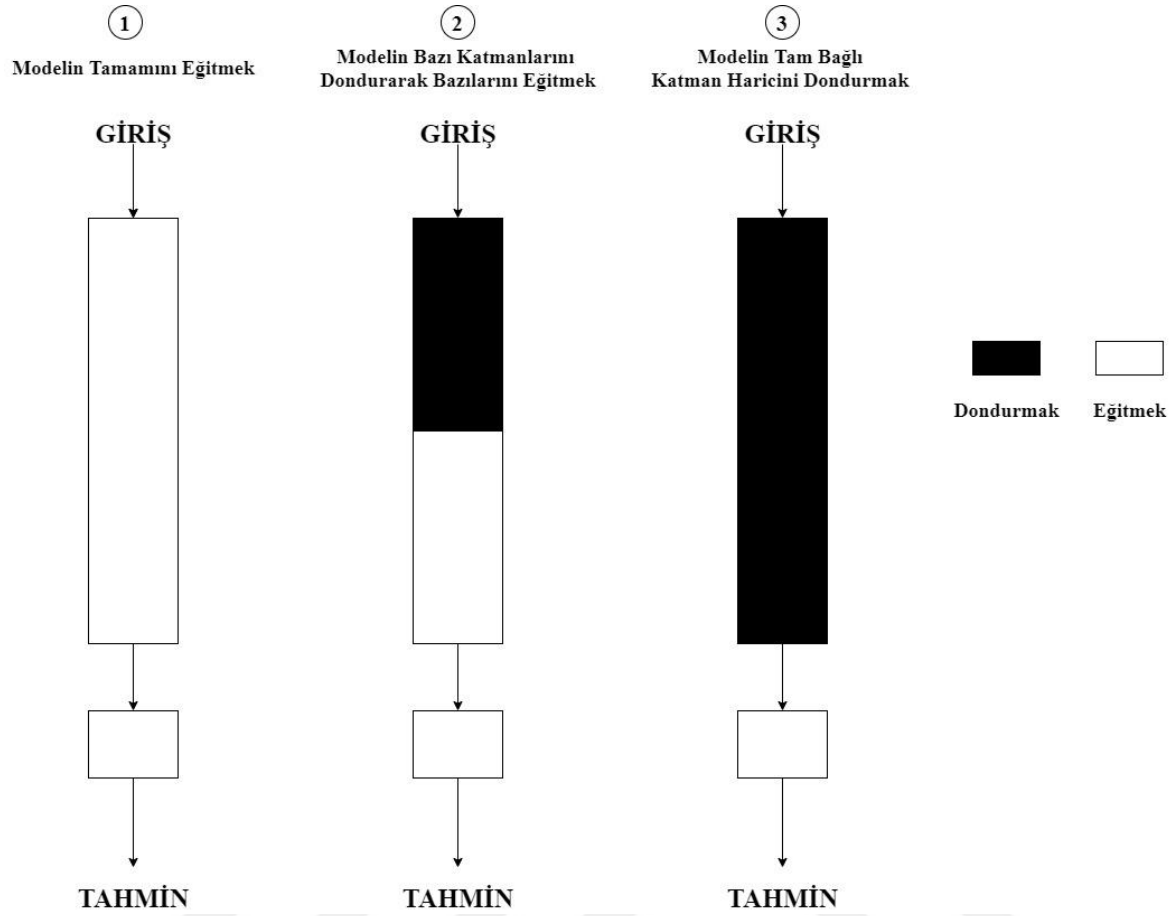
Model yaklaşımında ilk önce girişte uygulanan görüntünün çıkışta başarılı bir şekilde tespit edecek olan bir tahmin modellemesi belirlenir. Bazı özelliklerin öğrenebilmesi için saf bir modelden daha iyi şekilde olması için model geliştirilmelidir. Önceden belirlenip geliştirilmiş olan model daha sonra ilgilenilen ikinci bir görevdeki model için başlangıç noktası olarak kullanılmaktadır. Ayrıca eğitilmiş modelin ya tamamını ya da belirli bir kısmı kullanılabilir.

Önceden eğitilmiş model yaklaşımı (pre-trained model)

Bu model yaklaşımında öncelikle mevcut modellerden önceden eğitilmiş bir kaynak modeli seçilmektedir. Bunun için birçok araştırma kurumu seçilebilecek model havuzuna büyük ve zorlu veri setleri ile eğitilmiş modeller sunmaktadır. Önceden eğitilmiş model daha sonra ilgilenilen ikinci bir görevdeki model için başlangıç noktası olarak kullanılmaktadır. Bu eğitilmiş modelin tamamının mı yoksa bir kısmının mı kullanılacağı tekniğe bağlı olarak belirlenmektedir.

3.1. İnce Ayar (Fine-Tuning)

Konvolüsyon katmanlarının ağırlıkları, aynı mimariye sahip önceden eğitilmiş bir CNN'nin ağırlıkları ile başlatılabilir. Önceden eğitilmiş ağ, farklı bir uygulamadan devasa bir etiketlenmiş veri seti ile üretilmektedir. Bir CNN'yi önceden eğitilmiş bir ağırlık setinden eğitmeye ince ayar denir ve çeşitli uygulamalarda başarıyla kullanılmıştır [39]. Şekil 3.2'de ince ayar yapma yöntemleri gösterilmiştir.



Şekil 3.2. İnce ayar (fine tuning) yöntemleri

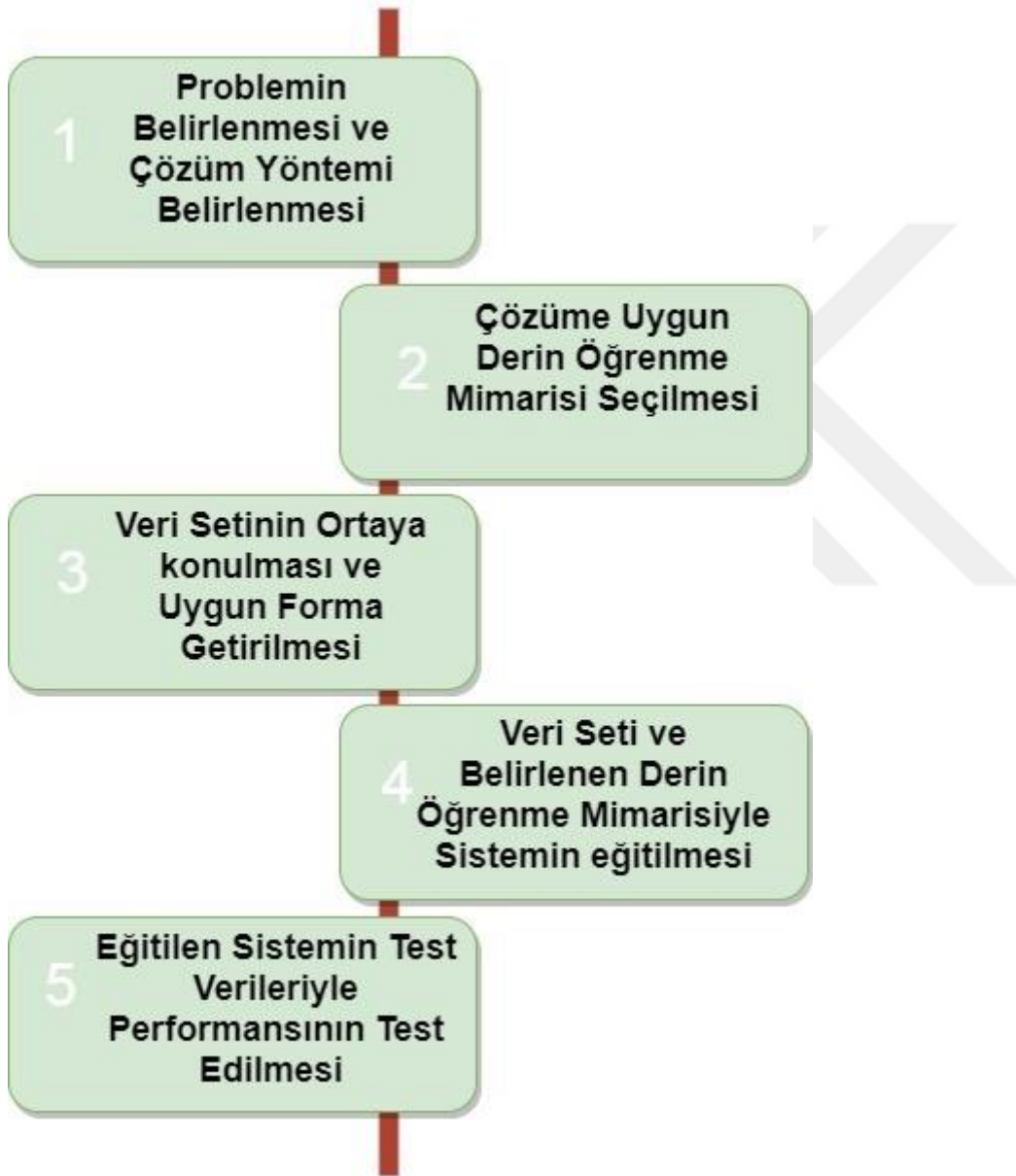
Uygulaması basit olan Strateji 3'ün aksine, Strateji 1 ve Strateji 2, konvolüsyon bölümde kullanılan öğrenme oranına dikkat edilmesi gerekir. Öğrenme hızı, ağırlıkların ne kadar ayarladığınızı kontrol eden hiper bir parametredir. CNN tabanlı önceden eğitilmiş bir model kullanırken, yüksek öğrenme oranları önceki bilgileri kaybetme riskini arttırdığı için küçük bir öğrenme oranı kullanmak akıllıca olmaktadır.

İnce ayar, ağırlıkların önceden eğitilmiş bir ağdan eğitmek istediğimiz ağa aktarılması ile başlamaktadır. Farklı olarak veri setindeki sınıfların sayısının tam bağlı katmandaki düğüm sayısıdır. Yaygın olarak kullanılan bir uygulama, önceden eğitilmiş CNN'nin son tam bağlı katmanının, yeni hedef uygulamadaki sınıf sayısı kadar nöron içeren yeni tam bağlı bir katmanla değiştirilmesidir.



4. MATERYAL VE METOT

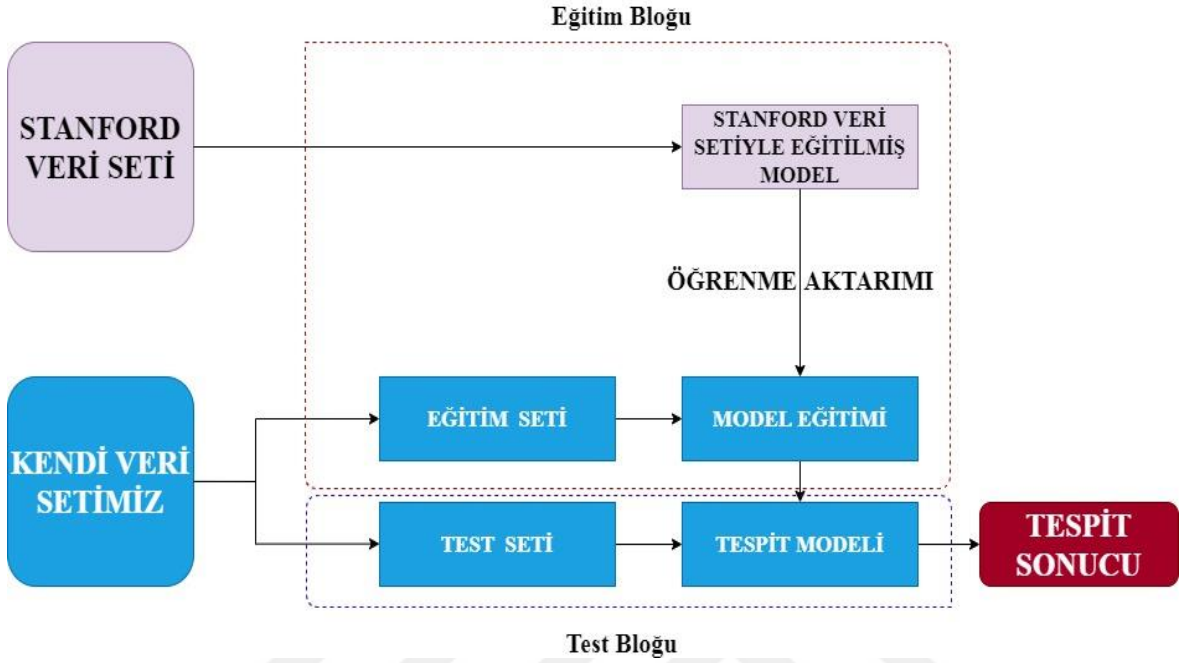
Bu kısımda derin öğrenme CNN tabanlı model ile insan/araç sınıflandırması yapılan sistemin süreç akışı aktarılmaktadır. Problemin çözümü için gerçekleştirilen adımlar genel olarak Şekil 4.1’de gösterilmektedir.



Şekil 4.1. Derin öğrenme süreci

Problemin belirlenmesinin ardından drone aracılığıyla elde edilen görüntüler CNN tabanlı ResNet-50'nin ağırlıklarını kullanan RetinaNet modeli ile eğitilmiştir. Böylece bu görüntülerle ağırlıklar yeniden güncellenmiştir. Eğitim işlemi için önceden eğitilmiş

ağırlıklar kullanılarak başarımlar artırılmak istenmiştir. Modeli eğitmek için kullanılan öğrenme aktarımı Şekil 4.2’de gösterilmektedir.



Şekil 4.2. Kullanılan modelde öğrenme aktarımının gerçekleştirilmesi

Model performansını artırmak için iki farklı veri seti kullanılmıştır. Modelde eğitimin yapılabilmesi için görüntüler uygun forma getirilerek model girişine uygulanmaktadır. Kendi veri setimizi uygun forma getirme işlemi uygulanan modelden modele farklılıklar göstermektedir.

Bu bölüm içeriğinde aktarılabilecek bilgiler şu şekildedir;

- Veri setlerinin tanıtımı ve nasıl oluşturulduğu
- Kendi veri setimizin etiketleme aşaması (model için uygun forma getirme işlemi)
- Kullanılacak olan model ve kullanım nedenleri
- Modelin eğitimi ve ardından eğitilmiş ağı test süreci
- Ağı eğitimi sırasında ortaya çıkan kayıp fonksiyonları
- Test işleminin ardından karşımıza çıkan sonuçların hata matrisi şeklinde aktarımı

4.1. Veri Setleri

Veri setlerinin çözmek istediğimiz probleme uygun olması büyük önem taşımaktadır. Veriler problemle uyumlu değilse terabaytlarca veriye sahip olmamızın herhangi bir önemi yoktur. İlk adım problemin çözümü için doğru veriyi toplama olmalıdır ve bunu başarana kadar kendimizi sürekli olarak bu adıma geri döndürmeliyiz.

İnsan/araç sınıflandırmak ve sınıfların yerinin tespit edilebilmesi için iki farklı veri seti kullanılmıştır. Bunlardan birisi açık kaynak olan Stanford Üniversitesi kampüsünde drone ile çekilen veri setidir. Diğer veri seti ise kendimizin oluşturduğu Afyon Kocatepe Üniversitesi kampüsünde drone ile çekilen görüntüler görüntülerden oluşmaktadır.

4.1.1. Stanford veri seti

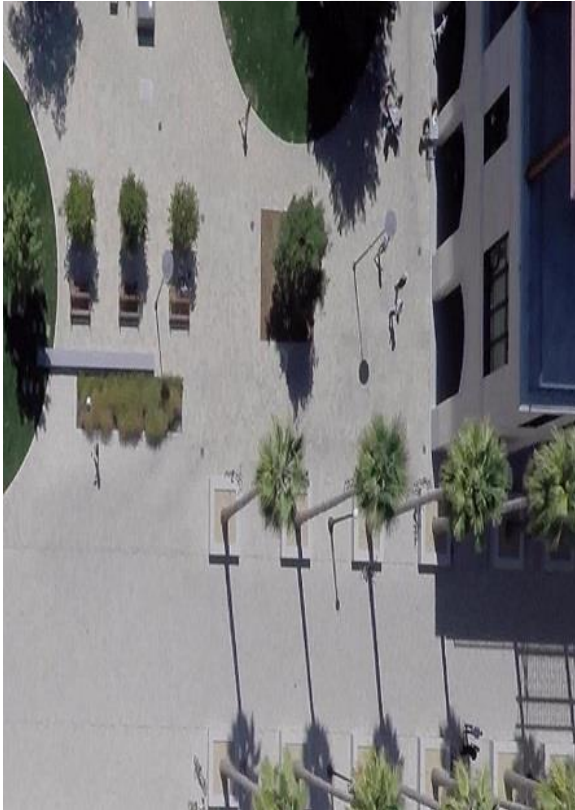
Veri seti Stanford Üniversitesi kampüsünün farklı alanlarında drone ile çekilen görüntülerden oluşmaktadır. Veri seti açık kaynak olmasından dolayı birçok akademik çalışmada karşımıza çıkmaktadır. Resim 4.1’de Stanford Üniversitesinin veri setinde yer alan görüntülerden bazıları gösterilmektedir.



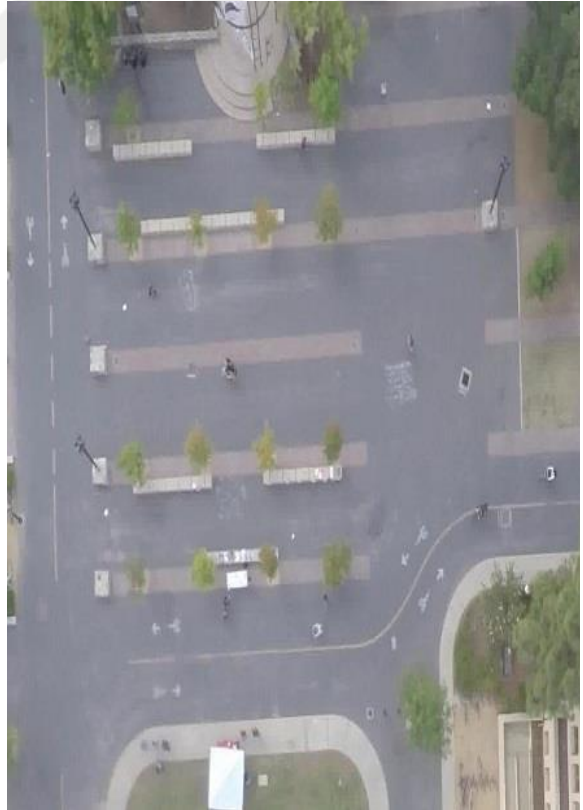
1.1.jpg



1.2.jpg



1.3.jpg



1.4.jpg

Resim 4.1.Stanford veri setinden görüntüler

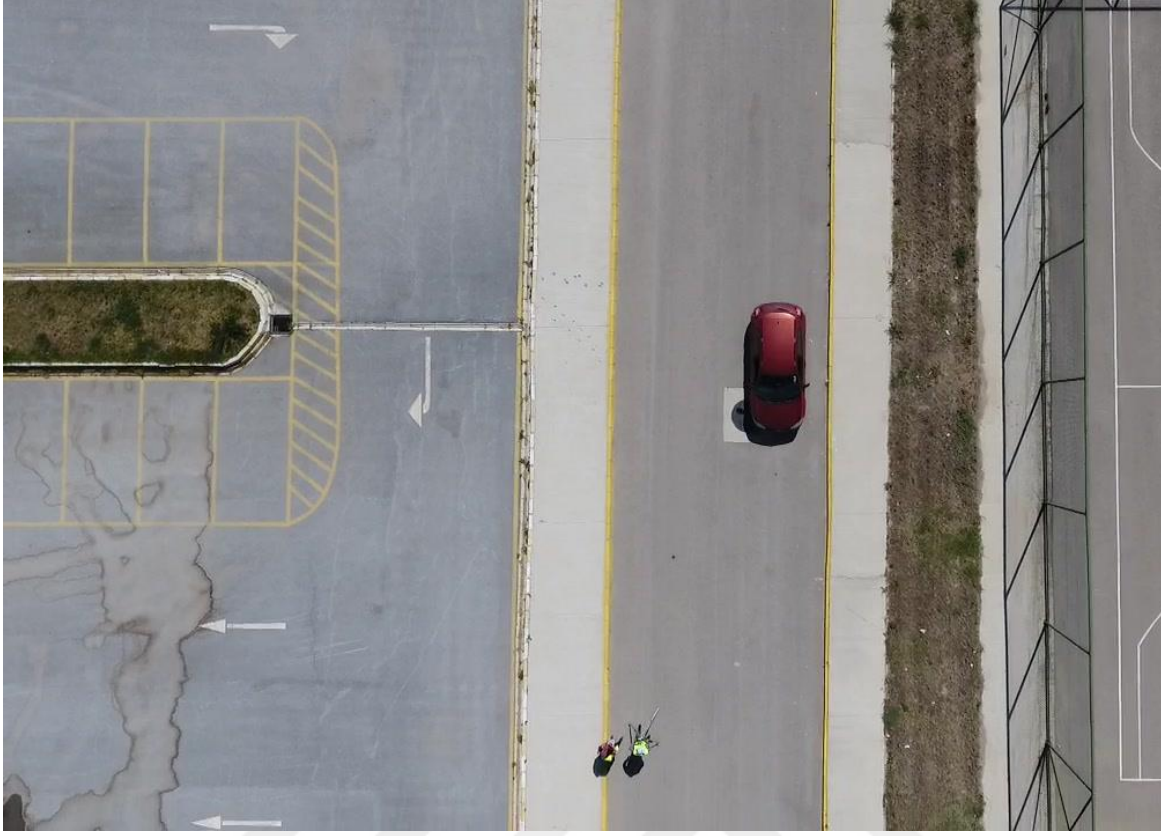
Veri seti üniversite kampüsünde farklı yüksekliklerden drone ile çekilen görüntülerden oluşmaktadır. Eğitim görüntüleri 50 metreden 115 metreye kadar olan belirli yüksekliklerden elde edilmiştir. Test görüntüleri ise 55 metreden 115 metreye kadar olan yüksekliklerden elde edilmiştir. Veri setinde yer alan görüntülerin dağılımı Çizelge 4.1’de verilmektedir

Çizelge 4.1. Stanford drone veri seti genel bilgileri

Görüntü Bilgisi	Görüntü Sayısı	Ortalama Çekim Yüksekliği
Kütüphane Bölgesi	587	83,66 m
Dinlenme Alanı 1	226	115,35 m
Kavşak Noktası	879	51,72 m
Kapılar Bölgesi	459	53,25 m
Dinlenme Alanı 2	370	76,86 m
Üniversite İçi Yol	70	55,79 m

4.1.2. Tasarlanan veri seti

Afyon Kocatepe Üniversitesi kampüsünden çekilen drone görüntülerinden oluşan veri setimiz insan ve araba sınıflarından oluşmaktadır. Eğitim ve test veri seti için oluşturulan görüntülerin yükseklikleri 25m’den 45m’e kadardır. Resim 4.2’de Afyon Kocatepe Üniversitesinin veri setinde yer alan görüntülerden bazıları gösterilmektedir.



1.1.jpg



1.2.jpg

Resim 4.2. Tasarlanan veri setinden görüntüler

Drone ile kuş bakışı bir şekilde çekilen videolardan elde edilen görüntüler sonucunda veri seti 518 adet görüntüden oluşmaktadır. Veri setinde yer alan görüntülerin dağılımı Çizelge 4.2’de verilmektedir.

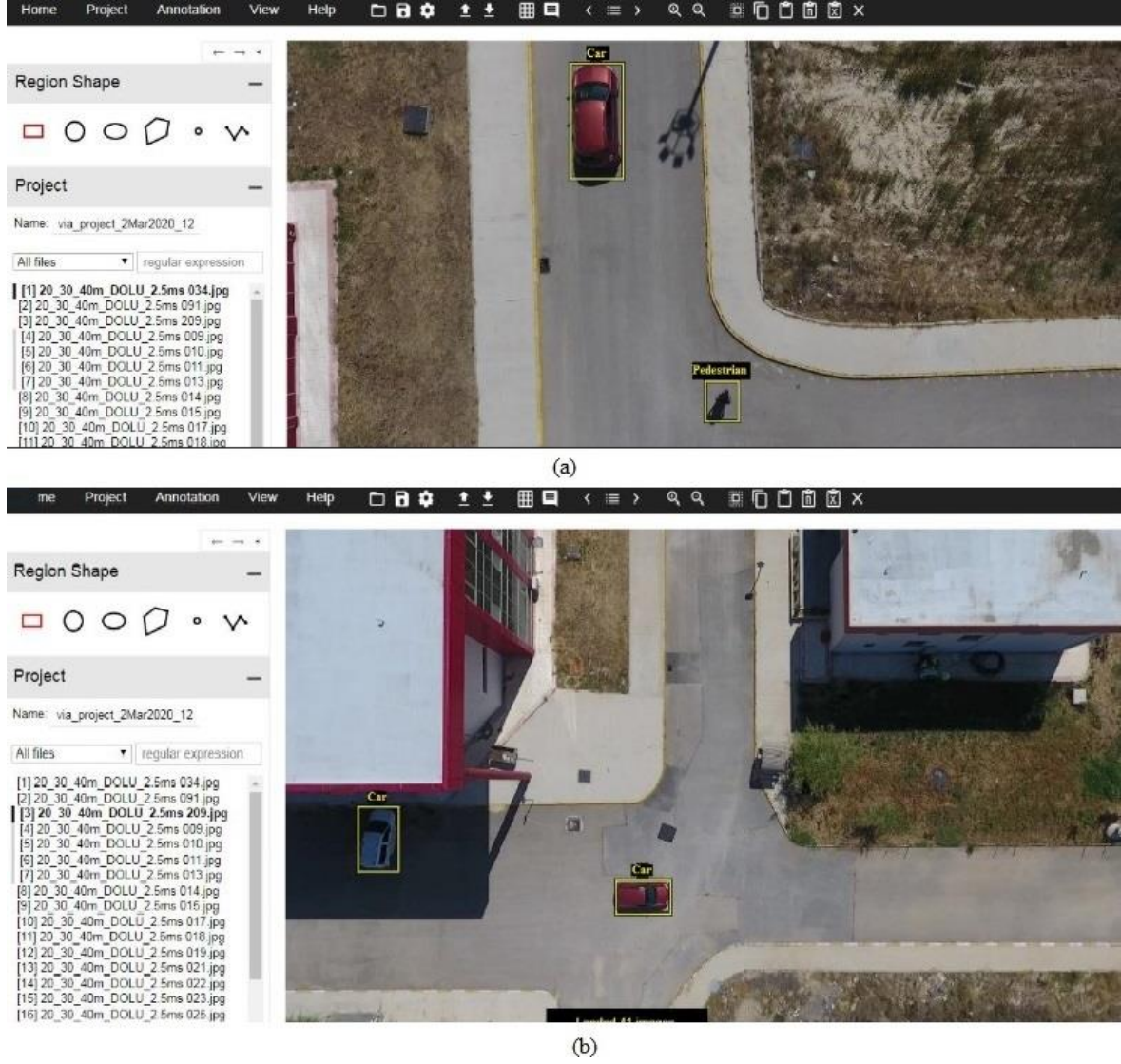
Çizelge 4.2. Tasarlanan veri setinin genel bilgileri

Sınıf	Her Sınıfın Toplam Görüntü Sayısı
Yaya	337
Araba	480

Çizelge 4.2’e göre Afyon Kocatepe Üniversitesi kampüsünden elde edilen görüntüler içerisinde 337 yaya 480 araba bulunmaktadır. Bu görüntüler tezin amacına yönelik drone aracılığıyla havadan kuş bakışı farklı yüksekliklerden elde edilmiştir. Görüntülerin büyük bir kısmı 1280x720 boyutunda olup bazıları da 2704x1520 boyutundadır.

Veri setlerinin etiketlemesi

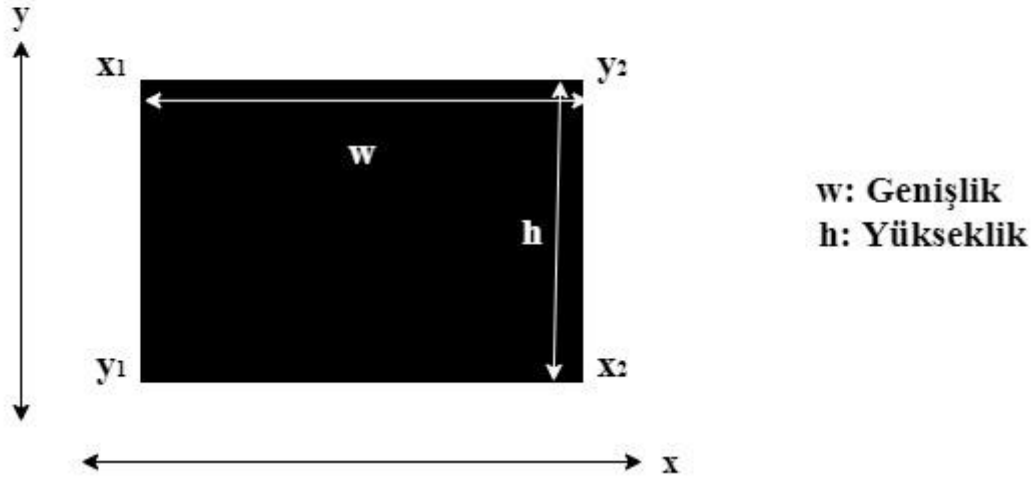
Belirli nesnelerde eğitimin yapılabilmesi için öncelikle veri setinin oluşturulması gerekmektedir. Ardından oluşturulmuş veri setindeki görüntülerin hepsinin tek tek etiketlenmesi gerekmektedir. Oluşturduğumuz veri setimizi etiketlemek için VGG Image Annotator (VIA) kullanılmıştır. Bu, herhangi bir kurulum gerektirmeyen ve yalnızca web tarayıcısında çalışan hafif, bağımsız ve çevrimdışı bir yazılım paketidir. VIA ile görüntülerin etiketlenme işlemlerinden sonra json veya csv düz metin formatında dosya oluşturulabilmektedir. Çalışmada VIA ile csv uzantılı dosya oluşturulmuştur ve eğitim işlemi için bu dosya kullanılmıştır. VIA’dan başka birçok etiketleme programı ve ara yüzü bulunmaktadır. VIA’nın tercih edilme nedeni ara yüzünün daha basit ve kullanışlı olmasıdır. Resim 4.3’te VIA ile veri setinde yer alan görüntülerin etiketleme işlemi gösterilmektedir.



Resim 4.3. VIA aracılığıyla görüntü etiketleme işlemi

Veri setinde yer alan görüntülerde sadece arabaları, sadece yayaları ve hem yayaları hem de arabaları içeren örnekler bulunmaktadır. Resim 4.3.'e göre (a)'da bir insan ve bir araba etiketlenmesi (b)'de ise iki adet araba etiketlenmesi yapılmıştır. VIA aracılığıyla tüm veri setindeki görüntüler etiketlendikten sonra cvs formatında çıktısı alınır. Oluşturulan csv formatında işaretlenen sınıfın koordinatları ve sınıfı belirlidir. Kullanılan RetinaNet modelinde tüm etiketlerin aynı formatta olması gerekmektedir. Şekil 4.3'te işaretlenen görüntünün olması gereken format şekli ve işaretlenen sınıfın koordinat düzleminde gösterimi aktarılmıştır.

Dosya Yolu/Görüntü Adı.jpg,x1,y1,x2,y2,Sınıf Adı



Şekil 4.3. Etiketlenen sınıfın koordinatları ve olması gereken format şekli

Eğitimde kullanılan etiketlenen görüntü dosyası belirli şekilde olması gerekmektedir. Şekil 4.3'te bu format şekli ve sırası görülmektedir. Dikdörtgen şekli ile işaretlenen görüntü sınıfları (yaya ve araba) bu formata göre dikdörtgenin kenarlarının apsisi ve ordinatı olacak şekilde verilmeli. Yani x_1, y_1, x_2, y_2 şeklinde sıralı olması gerekmektedir. VIA ile dikdörtgen şekli kullanılarak yapılan etiketleme işleminin sonucunda x (görüntünün merkez), y (görüntünün merkez), w, h formatında çıktı alınmaktadır. Görüntülerin boyut bilgileri (genişlik ve yükseklik) de dikkate alınarak x_1, y_1, x_2, y_2 formatına dönüşümler yapılmıştır ve modelin eğitimde bu format kullanılmıştır. Çizelge 4.3.'te bazı etiketlenen görüntülerin uygun formda yazılmış hali verilmektedir.

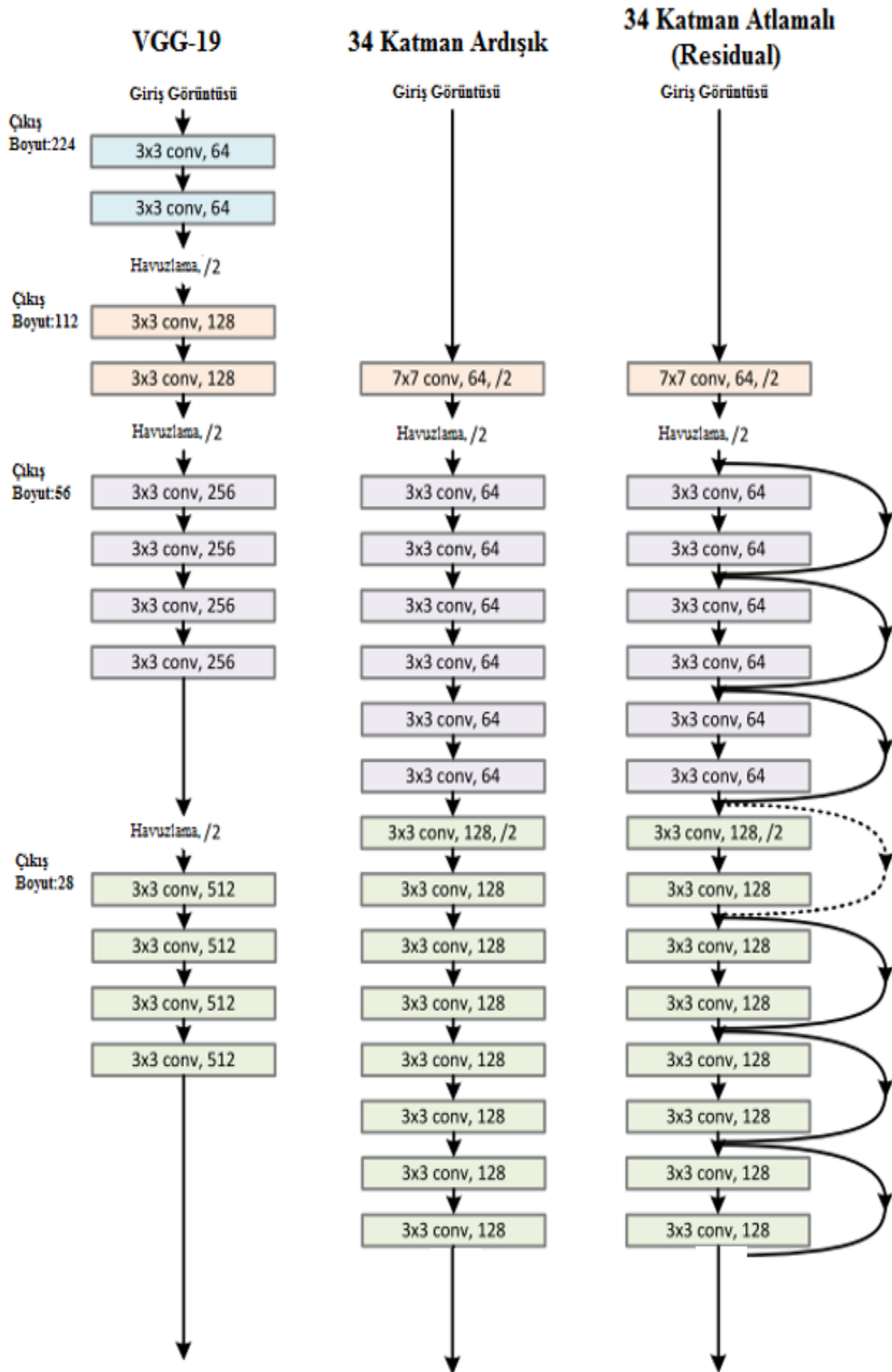
Çizelge 4.3. Etiketlenen görüntülerden bazısının uygun formda yazımı

Etiketlenen görüntülerin uygun formatta yazımı
imgs1/train1/20_30_40m_BOS_3.5ms 012.jpg,764,200,801,251,Pedestrian
imgs1/train1/DJI_0029_29.6mIrtifa 057.jpg,318,300,367,411,Car
imgs1/train1/DJI_0032-40m-720 087.jpg,1127,7,1158,30,Pedestrian
imgs1/train1/20_30_40m_DOLU_3.5ms 138.jpg,134,402,192,507,Car

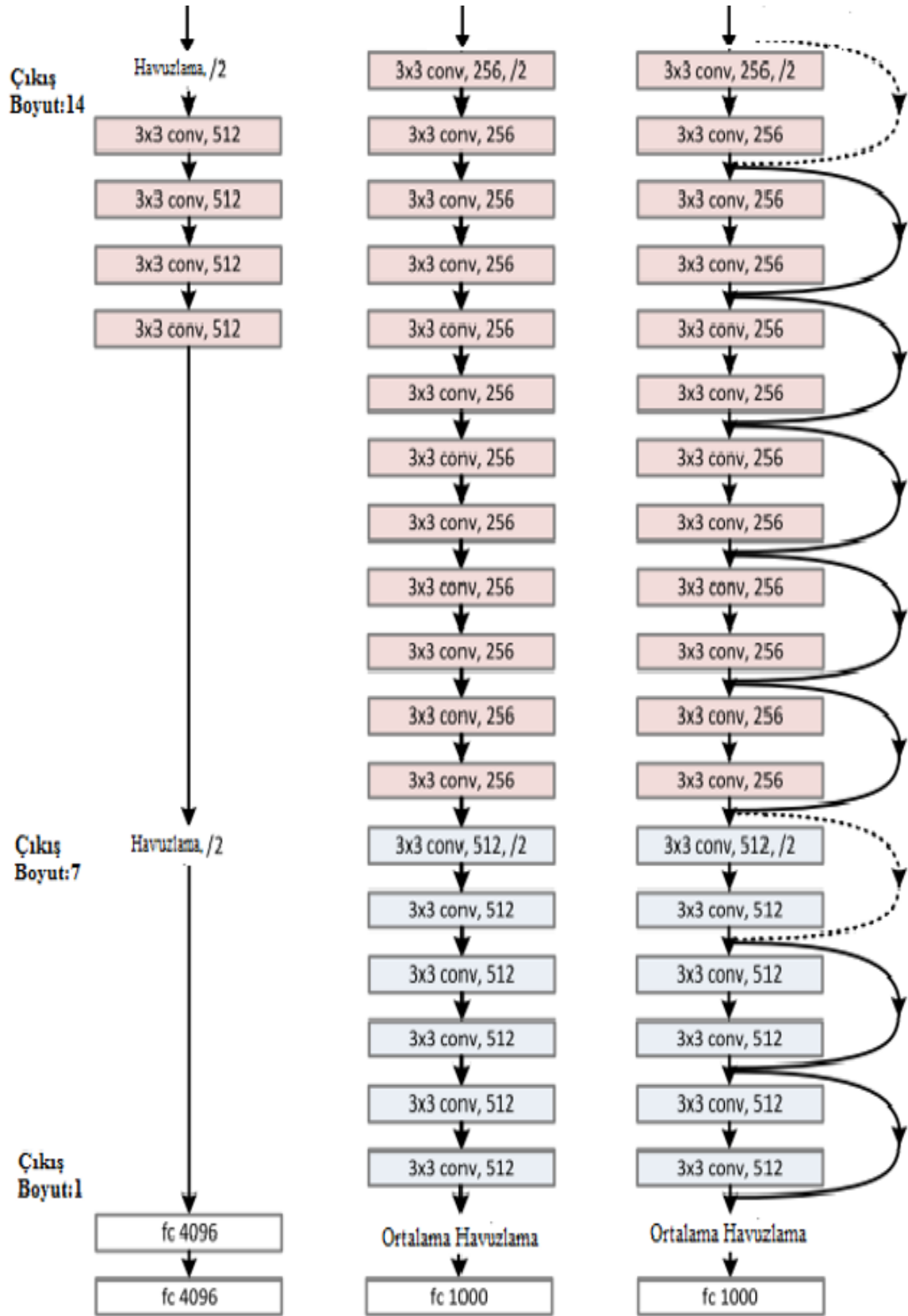
4.2. Çalışmada Kullanılan CNN Model

Nesne tespitiinde başarılı sonuçlar alabilmek için kullanılan model çok önemlidir. Kullanılan veri setinin büyüklüğü model seçimini etkilemektedir. Çalışmada kullanılan veri setlerinde özellikle Stanford veri setinin büyüklüğünden dolayı ve eğitim esnasındaki kayıpları (gradient kaybolmasını) önlemek için geliştirilmiş olan ResNet CNN modeli kullanılmıştır. Ayrıca bu model ağı literatürde yapılan çalışmalarda görüldüğü gibi yüksekten nesne tespiti ve sınıflandırma problemlerinde başarılı sonuçlar sergilemektedir. Şekil 4.4'te kullanılan ResNet modelin ilk 34 katmanı ve Vgg-19 sinir ağı karşılaştırma amaçlı verilmektedir.





Şekil 4.4. ResNet modelinin ilk 34 katmanının gösterimi



Şekil 4.4.(devamı) ResNet modelinin ilk 34 katmanının gösterimi

Şekil 4.4'te gösterilen 34 katman atlamalı (residual) ağ ResNet sinir ağına karşılık gelmektedir. 34 katman ardışık ağ ise şekildeki ResNet ağının katmanlarının ardışık şekilde

sıralanmış biçimidir. Derin öğrenmede oldukça derin katmanların ardışık olmasından ziyade atlamalı olması tercih edilmektedir. Bu tür ağlar bilgi kaybının önüne geçilmesi için kullanılmaktadır. Vgg-19 yerine ResNet sinir ağının tercih edilme nedeni de ResNet'in derin ağlarda Vgg-19'a göre başarılı sonuçlar vermesindendir.

Nesne tespiti için ResNet-50 sinir ağının önceden eğitilmiş ağırlıklarını kullanan RetinaNet modeli kullanılmıştır. Bu model nesne boyutlarının oldukça küçük olduğu hava görüntülerden oluşan veri kümesinde nesneleri başarıyla algılayabilmektedir. RetinaNet bir omurga ve farklı görevlere özgü iki alt ağdan oluşan birleşik ağıdır. Omurga, tüm girdi görüntüsü üzerinde bir dönüşüm özelliği haritasının hesaplanmasından sorumludur. İlk alt ağ, omurga çıkışında sınıflandırma yapmaktadır; İkinci alt ağ konvolüsyon sınırlayıcı kutu regresyonu gerçekleştirmektedir. Sinir ağının eğitim sürecinde sınıflandırma ve regresyon kaybı şeklinde iki farklı kayıp fonksiyonu karşımıza çıkmaktadır.

4.3. TensorFlow Kütüphanesi

Tez çalışmasında derin öğrenme kütüphanesi olarak Google tarafından geliştirilen tensorflow kullanılmıştır. Tensorflow, sayısal hesaplama ve büyük ölçekli derin öğrenimi için açık kaynaklı bir kütüphanedir. Tensorflow tarafından eğitilmiş modeller ile kendi görüntümüzde nesne sınıflandırılması yapılmaktadır. Tensorflow bizlere bu işlem için bazı modeller önermektedir.

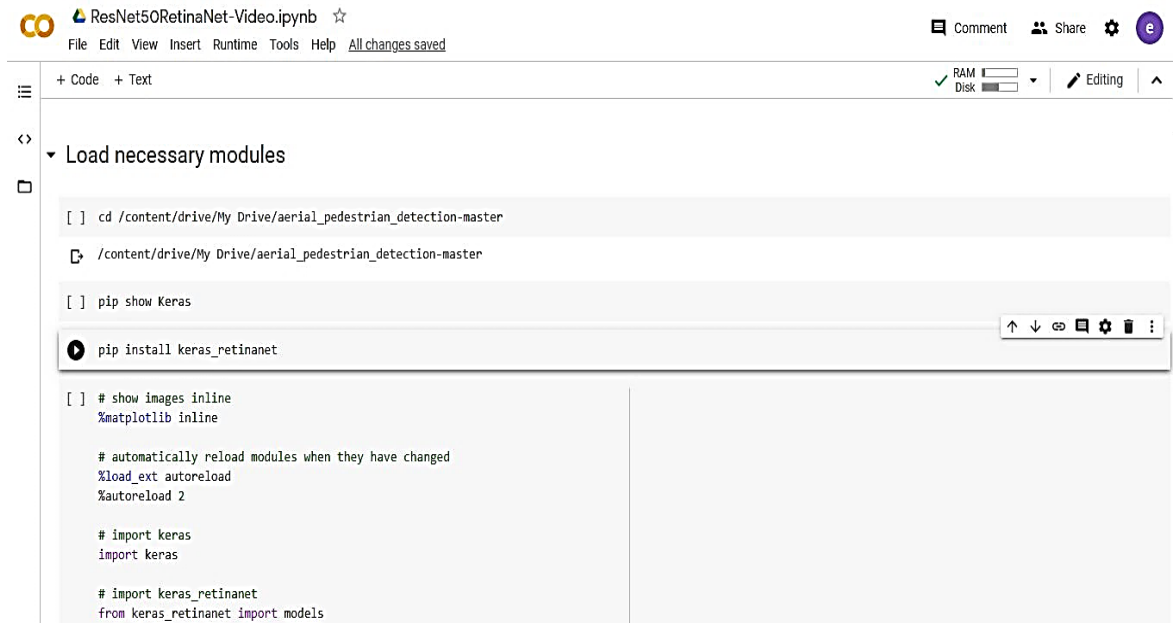
Tensorflow tarafından COCO veri seti ile eğitilmiş olan SSD ve RCNN modeller ile sınıflandırma yapılmaktadır. Tez çalışmasında sınıflandırma performansını daha yüksek seviyelere çıkarmak için Stanford veri seti ve kendi oluşturduğumuz veri seti üzerinde yeni bir model eğitmemiz gerekmektedir.

Sınıflandırma işleminin yapılması için modeli sıfırdan oluşturmak yerine önceden eğitilmiş hazır modeli kullanılmıştır. Tensorflow üzerinden hem Stanford veri setini hem de kendi oluşturmuş olduğumuz veri seti eğitilmiştir. Bu çalışmada daha önceden eğitilmiş olan modelleri kendi çalışmamızda kullanılan veri setleriyle eğitilerek başarı sonuçları gözlemlenmiştir.

4.4. Eğitim ve Test Süreci

Veri setleri belirlenip uygun forma getirildikten sonra model eğitimi yapılabilmektedir. Modelin veri setleriyle eğitimi oldukça fazla zaman almaktadır. Bu süreçte yüksek performanslı işlemcilerle ihtiyaç duyulmaktadır.

Çalışmada modelin eğitilmesi ve test edilmesi için Google Colaboratory sistemi kullanılmıştır. Kısa adıyla Colab ile Keras, Tensorflow, Pytorch, OpenCV gibi kütüphaneler kullanılarak derin öğrenme çalışmaları geliştirilmektedir. Diğer bulut sistemlerinden farklı olarak Colab ücretsiz GPU desteği sağlamaktadır. Colab sisteminde NVIDIA Tesla K80 grafik işlemcisi bulunmaktadır. Böylece büyük veri setlerinde daha çok hesaplama yapılacağından hızlı bir şekilde sonuçları sunabilmektedir. Şekil 4.5'te Colab'ın çalışma ortamı verilmektedir.



Şekil 4.5. Google Colaboratory çalışma ortamı

ResNet ağını omurga olarak belirlediğimiz çalışmada RetinaNet modeli ile eğitimde toplam 36 milyon 486 bin 682 parametre kullanılmıştır. Bu parametrelerin 106 bin 240 tanesi eğitim sürecinde eğitilemedi. Geriye kalan parametreler eğitilmiştir.

Derin öğrenmede bir model tasarlanırken belirli parametreler bulunmaktadır. Bu parametrelerin tercihi başlangıçta açık ve kesin değildir. Bu nedenle tasarlayan kişiye

bırakılmaktadır. Parametrelerin ne olması gerektiği probleme ve veri setine göre değişiklik göstermektedir. Bu parametrelere literatürde hiper-parametre denilmektedir. Modeli eğitimde kullanırken ihtiyaç duyulan bazı önemli hiper-parametreler şunlardır.

Veri seti

Veri seti olarak Stanford drone seti ve kendi oluşturduğumuz veri seti kullanılmıştır. Eğitim ve test işlemlerinde bu veri setlerindeki görüntülerden faydalanılmıştır.

Öğrenme hızı (learning rate)

Derin öğrenmede ağırlıkların güncellenmesi geriye yayılım işlemiyle olmaktadır. Bu güncelleme işleminde geriye doğru türev alınarak elde edilen fark değerinin öğrenme hızıyla çarpılması sonucunda yeni ağırlıklar belirlenmektedir. Çalışmada eğitim sürecinde öğrenme hızı parametresinin değeri $1e-5$ olarak belirlenmiştir.

Eğitim tur (epoch) sayısı

Modelin eğitimi esnasında veri setlerindeki görüntülerin tamamı aynı anda eğitilmemektedir. Veriler belirli sayıda parçalar halinde eğitime katılmaktadırlar. Belirlenen eğitim tur sayısı kadar eğitim devam etmektedir. Eğitimin ilk turunda başarımlar oldukça düşüktür ve turlar devam ettikçe başarımlar artmaktadır. Belirli bir süre sonra modelin öğrenmesi oldukça yavaşlayacaktır. Çalışmada eğitim için belirlenen eğitim tur sayısı 50'dir. Bu parametre için farklı değerler de alınabilirdi fakat ağırlık eğitimi için tur sayısının 50 olması yeterli görülmüştür.

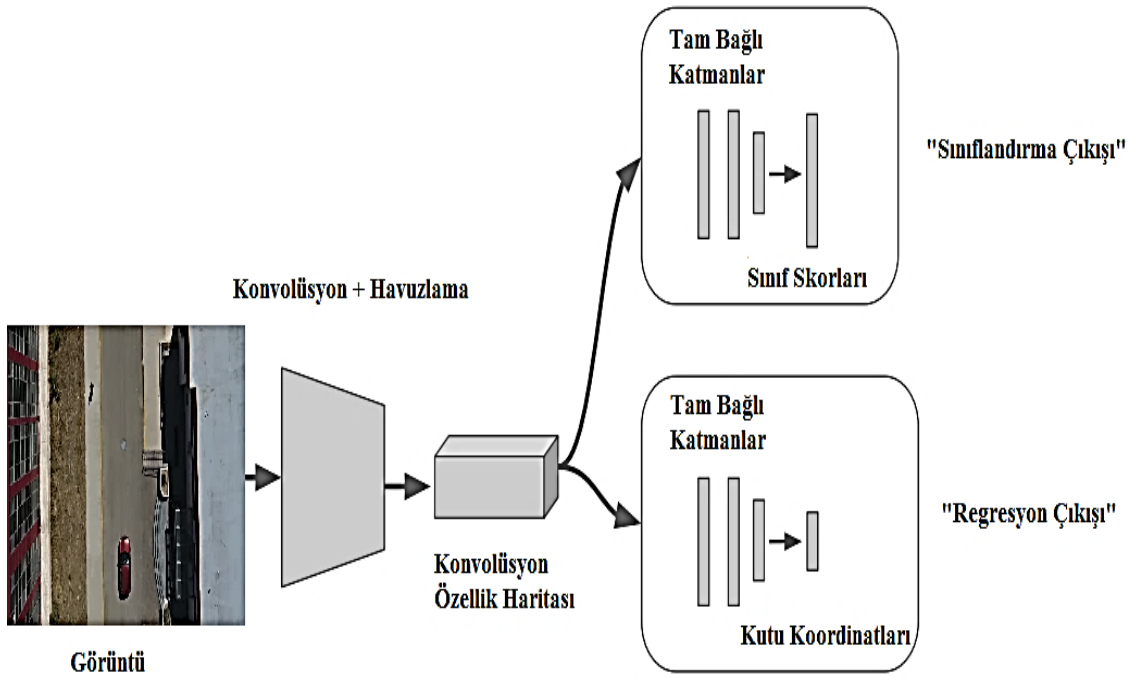
Adım sayısı

Gelenek olarak, eğitim tur sayısı başına düşen adım sayısı eğitim veri setindeki görüntü sayısının batch size'a bölünmesiyle hesaplanmaktadır. Çalışmada sinir ağının eğitimi için bir eğitim tur sayısı başına düşen adım sayısı 2000 olarak belirlenmiştir.

Modelin eğitilmesinin ardından performansının test edilmesi gerekmektedir. Test işlemi için Stanford drone setinin test veri seti ve kendi oluşturmuş olduğumuz test veri seti kullanılmıştır.

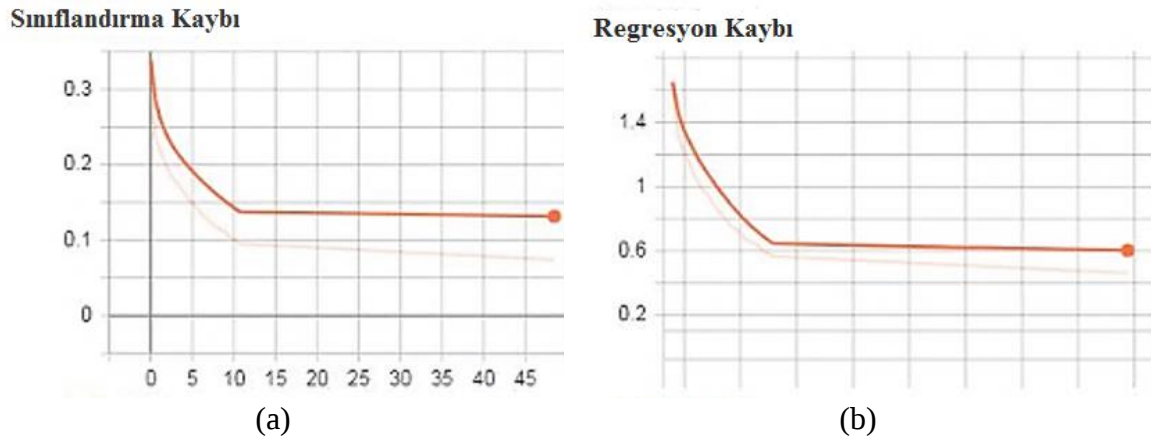
4.5. Kayıp Fonksiyonları

Nesne sınıflandırma işleminde kullanılmakta olan derin ağ mimarileri çok çıkışlı ağ yapısına sahiptirler. Bu yapıda derin öğrenme modeli nesne tanıma işlemi için nesnenin sınıfını ve konumunu belirlemesi gerekmektedir. Her bir çıktının kendine ait bir hata fonksiyonu bulunmaktadır. Bu hata fonksiyonları eş zamanlı bir şekilde optimize edilmeye çalışılmaktadır. Şekil 4.6'da çalışmada kullanılan giriş görüntüsünün nesne tanımda kullanılan ağ mimarisi gösterilmektedir.



Şekil 4.6. Kullanılan mimarinin regresyon ve sınıflandırma çıkışının gösterimi

Şekil 4.6'daki modelde, sınıflandırma çıkışı (classifier head) ve regresyon çıkışı (Regressor Head) adlı iki çıktıya sahiptir. Sınıflandırma çıkışı, nesnenin cinsini bulmaya çalışır ve hata fonksiyonu olarak çapraz entropi kullanır. Regresyon çıkışı, nesneyi çevreleyen dikdörtgenin köşe koordinatlarını bulmaya çalışır ve bunun için hata fonksiyonu olarak ortalama karesel hata kullanılır. Şekil 4.7'de sınıflandırma ve regresyon kaybı görsel olarak verilmektedir.



Şekil 4.7. (a). Sınıflandırma kaybı, (b). Regresyon kaybı

Şekil 4.7’de yer alan sınıflandırma ve regresyon kayıp grafiklerine bakıldığında beklenildiği gibi belirli bir eğitim turu boyunca azalan ve ardından sabitlenen grafik görülmektedir.

4.6. Hata Matrisinin Çıkarılması (Confusion Matrix)

Hata matrisi sınıflandırma problemine yönelik tahmin sonuçlarının bir özetidir. Doğru ve yanlış tahminlerin sayısı sayım değerleri ile özetlenir ve her sınıf tarafından ayrılmaktadır. Hata matrisi sınıflandırma modelimizin tahminlerde nasıl bir performans sergilediğini göstermektedir. Bize sadece bir sınıflandırıcı tarafından yapılan hatalar hakkında değil, daha da önemlisi yapılan hata türleri hakkında da bilgiler vermektedir. Çizelge 4.3’te hata matrisinin genel görünüşü verilmektedir.

Çizelge 4.3. Hata matrisinin çıkarılması

	Pozitif Tahmin Edilen Değerler	Negatif Tahmin Edilen Değerler
Pozitif Gerçek Değerler	Doğru Pozitif	Yanlış Negatif
Negatif Gerçek Değerler	Yanlış Pozitif	Doğru Negatif

Hata matrisinin belirlenmesi modelin çalışma performansı hakkında bilgiler vermektedir. Modelin görüntülerden istenilen sınıfları tespit edip edemediği bu hata matrisi ile gözlemlenmektedir.

Çizelge 4.4'te farklı eşik değerleri kullanılarak hata matrisi belirlenmiştir. Çalışmamızda bu ortaya koyulan tablodan yola çıkarak kesinlik, duyarlılık ve f1 skor değerleri hesaplanmaktadır.

Çizelge 4.4. Eşik değerleriyle hata matrisinin belirlenmesi

Eşik Değeri	Toplam Doğru Pozitif	Toplam Yanlış Pozitif	Toplam Yanlış Negatif	Toplam Yanlış Sınıf Eşleşmesi	Toplam Yerel Doğrular
0,3	6855	651	497	628	6867
0,4	6282	296	731	274	6867
0,5	5795	112	1058	119	6867

Çizelge 4.4'de görüldüğü gibi eşik değerinin yükselmesiyle birlikte modelin görüntülerde doğru pozitif, yanlış pozitif nesnelerin tespiti ve yanlış sınıfın eşleşmesi azalmaktadır. Yanlış negatif değeri ise eşik oranının artmasıyla artmaktadır.

Kesinlik değeri doğru sınıflandırılmış pozitif örneklerin toplam sayısını, tahmin edilen pozitif örneklerin toplam sayısına bölerek hesaplanmaktadır. Eş. 4.1'de kesinlik ölçütünün denklemi bulunmaktadır.

$$\text{Kesinlik} = \frac{\text{Doğru Pozitif}}{\text{Doğru Pozitif} + \text{Yanlış Pozitif}} \quad (4.1)$$

Eş. 4.1'e göre az sayıda yanlış pozitif değerin olması yüksek pozitif olarak etiketlenmiş bir örneğin gerçekten pozitif olduğunu göstermektedir.

Duyarlılık doğru sınıflandırılmış pozitif örneklerin toplam sayısının toplam pozitif gerçek örnek sayısına oranı olarak tanımlanmaktadır. Eş. 4.2'de duyarlılık ölçütünün denklemi bulunmaktadır.

$$\text{Duyarlılık} = \frac{\text{Doğru Pozitif}}{\text{Doğru Pozitif} + \text{Yanlış Negatif}} \quad (4.2)$$

F1 skor, duyarlılık ve kesinlik adında iki ölçümümüz olduğundan her ikisini de temsil eden bir ölçümün yapılmasına yardımcı olmaktadır. F1 skor aşırı değerleri daha fazla

cezalandırdığı için aritmetik ortalama yerine harmonik ortalamayı kullanarak hesaplanmaktadır. Eş. 4.3'te F1 skor ölçütünün denklemi bulunmaktadır.

$$F1 \text{ Skor} = 2 * \frac{\text{Kesinlik} * \text{Hassasiyet}}{\text{Kesinlik} + \text{Hassasiyet}} \quad (4.3)$$

F1 skor her zaman duyarlılık ve kesinlik ölçütlerinin değeri küçük olanına yakın sonuçlar vermektedir.

Çalışmada yapılan eğitim ve test işlemlerinin ardından hata matrisi ortaya çıkarıldı. Belirli hesaplamalar sonucunda bu üç ölçüt (kesinlik, duyarlılık ve f1 skor) çalışmaya aktarılmıştır. F1 skor değerini maksimum seviyeye çıkarmak için belirli eşik değerlerinde tek tek kesinlik ve duyarlılık değerleri hesaplanmıştır. F1 skoru maksimum yapan optimum eşik değeri belirlenmiştir. Çizelge 4.5'te farklı eşik değerlerine göre kesinlik, duyarlılık ve f1 skor değerleri verilmektedir.

Çizelge 4.5. Farklı eşik değerlerine göre kesinlik, duyarlılık ve f1 skor değerleri

Eşik Değeri	Kesinlik	Duyarlılık	F1 Skor
0,3	0,842	0,932	0,885
0,4	0,916	0,895	0,906
0,5	0,961	0,845	0,899

Çizelge 4.5'e göre eşik değeri 0,4 olduğunda f1 skor değeri maksimum değeri olan 0,906 olmaktadır. Eşik değeri 0,3 olarak belirlendiğinde duyarlılık 0,932 değerini, 0,5 olarak belirlendiğinde ise kesinlik 0,961 değerini aldığı görülmektedir.



5. SONUÇ VE ÖNERİLER

Drone ile hedef tespiti, hassas askeri uygulamalarda hedeflerin tespit edilmesi, düşmanların durum analizi, personel arama kurtarma, tarım ve hayvancılık izleme alanlarında son derece yüksek bir uygulama değerine sahiptir. Son zamanlarda derin öğrenme algoritmalarının da geliştirilmesiyle oldukça popüler olan olarak karşımıza çıkmaktadır. Bu çalışmada derin öğrenme ile önceden eğitilmiş sinir ağı üzerine Stanford drone seti ve kendi oluşturmuş olduğumuz hava görüntü setinin de eğitilmesiyle yeni bir yapı oluşturulmuştur. Afyon Kocatepe Üniversitesi kampüsünde drone ile kuş bakışı bir şekilde çekilen görüntülerden kendi veri setimizi oluşturduk. Veri setinde yer alan görüntülerin model için uygun forma getirilmesi gerekmektedir. Bu yüzden VIA görüntü etiketleme programı ile görüntülerin tek tek etiketleme işlemleri yapıldıktan sonra etiketlemeler modelin uygun formuna dönüştürülerek eğitim ve test işlemlerine hazır hale getirilmiştir. Farklı yüksekliklerden oluşturulmuş olan eğitim veri setiyle farklı yüksekliklerden insanları ve araçları tespit eden bir sistem geliştirilmiştir. Sistemde kendisine gösterilen insanları ve araçları başarılı bir şekilde tespit etmesi için derin öğrenme tabanlı CNN kullanılmıştır. Oldukça derin ağlarda başarılı sonuçlar veren ResNet-50 ağırlıklarını kullanan RetinaNet modeli kullanılmıştır. Sinir ağının eğitimi Google Colabatory üzerinde gerçekleştirilmiştir. Sinir ağının eğitim ve test işlemlerinin yapılması için toplamda 3795 görüntü içeren iki veri seti kullanılmıştır. Bu veri setinde yer alan görüntülerin yaklaşık %80'i eğitim işlemi için, geri kalan %20'lik kısmı da sistemin test işlemi için kullanılmıştır. Eğitim esnasında ortaya çıkan sınıflandırma ve regresyon kayıpları da incelendiğinde beklenildiği gibi eğitim tur sayısının artmasıyla birlikte kayıplar azalmaktadır ve bir süre sonra yaklaşık olarak sabitlenmektedir. Sistemin performansının ortaya konulması için farklı eşik değerleri (0,3,0,4 ve 0,5) uygulanarak hata matrisleri çıkartılmıştır. Hata matrislerinden yola çıkarak kesinlik, duyarlılık ve f1 skor değerleri belirlenmiştir. Buna göre 0,4 eşik değeriyle elde edilen nesne tespitlerinde 0,916 kesinlik, 0,895 duyarlılık ve 0,906 f1 skor değeri elde edilmiştir.

Yapılan çalışmayla görüntülerde yer alan belirli sınıfların tespiti ve koordinatlarının belirlenmesi sağlanmıştır. İleriye dönük çalışmalarda performans oranının artırılması için belirli işlemler yapılabilir. Bunlar;

- Özellikle modelin ezberlemesini engellemek için eğitim esnasında her bir eğitim turunda değerlendirme yapılarak hangi turda aşırı öğrenmeye gidildiği belirlemek.

- Çalışmanın kullanılacağı bölgenin kritik noktaları için zorlu görüntülerin veri setine eklenmesiyle modelin zor koşullarda da başarılı sonuçlar vermesi sağlanmak.
- Gerçek zamanlı yapılacak olan uygulamalarda eğitim süresinin ve hedef tespitin hızlı bir şekilde tamamlanabilmesi için yüksek hızlı GPU kullanılması performansın oldukça artmasını sağlamak.
- Veri setinde yer alan görüntülerin etiketleme işlemleri için daha basit alternatifler getirilerek veri setinin sinir ağı için daha kısa sürede hazır olmasını sağlamak.
- Sinir ağında başarılı şekilde eğitim yapılabilmesi için kullanılan filtreleri optimizasyon yöntemleri kullanılarak belirlemek.



KAYNAKLAR

1. Panchal, P., Prajapati, G., Patel, S., Shah, H., and Nasriwala, J. (2015). A review on object detection and tracking methods. *International Journal for Research in Emerging Science and Technology*, 2(1), 7-12.
2. Maraş, B., Arica, N., ve Ertüzün, A. B. (2017). Görsel Hedef Takibi Yöntemlerine Genel Bakış. *Elektrik Mühendisliği Odası Bilimsel Dergi*, 7(13), 5-16.
3. Shah, U., and Harpale, A. (2018). *A Review of Deep Learning Models for Computer Vision*. In 2018 IEEE Punecon, Pune, India, 1-6.
4. Sørensen, R. A., Rasmussen, J., Nielsen, J., and Jørgensen, R. N. (2017). *Thistle detection using convolutional neural networks*. In 2017 Efit Wcca Congress, Montpellier, France, 161-162.
5. İnternet: Vaddi, S., Kumar, C., and Jannesari, A. (2019). Efficient object detection model for real-time UAV applications. URL: <https://arxiv.org/abs/1906.00786>, Son Erişim Tarihi: 20.03.2020
6. AlDahoul, N., Sabri, M., Qalid, A., and Mansoor, A. M. (2018). Real-time human detection for aerial captured video sequences via deep models. *Computational intelligence and neuroscience*, 2018, 1-14.
7. Redmon, J., Divvala, S., Girshick, R., and Farhadi, A. (2016). *You only look once: Unified, real-time object detection*. In Proceedings of the IEEE conference on computer vision and pattern recognition, Las Vegas, USA, 779-788.
8. Haque, M. F., Lim, H. Y., and Kang, D. S. (2019). *Object Detection Based on VGG with ResNet Network*. In 2019 International Conference on Electronics, Information, and Communication, Auckland, New Zealand, 1-3.
9. García Rubio, V., Ferrán, R., Antonio, J., Menéndez García, J. M., Sánchez Almodóvar, N., Lalueza Mayordomo, J. M., and Álvarez, F. (2019). Automatic Change Detection System over Unmanned Aerial Vehicle Video Sequences Based on Convolutional Neural Networks. *Sensors*, 19(20), 4484-4499.
10. Goyette, N., Jodoin, P. M., Porikli, F., Konrad, J., and Ishwar, P. (2012). *Change detection. net: A new change detection benchmark dataset*. In 2012 IEEE computer society conference on computer vision and pattern recognition workshops, Providence, USA, 1-8.
11. İnternet: Ammar, A., Koubaa, A., Ahmed, M., and Saad, A. (2019). Aerial Images Processing for Car Detection using Convolutional Neural Networks: Comparison between Faster R-CNN and YoloV3. URL: <https://arxiv.org/abs/1910.07234>, Son Erişim Tarihi: 24.03.2020

12. d'Acremont, A., Fablet, R., Baussard, A., and Quin, G. (2019). *CNN-based target recognition and identification for infrared imaging in defense systems*. *Sensors*, 19(9), 2040-2055.
13. Akbar, J., Shahzad, M., Malik, M. I., Ul-Hasan, A., and Shafait, F. (2019). *Runway Detection and Localization in Aerial Images using Deep Learning*. In 2019 Digital Image Computing: Techniques and Applications, Perth, Australia, 1-8.
14. Chang, Y. C., Chen, H. T., Chuang, J. H., and Liao, I. C. (2018). *Pedestrian Detection in Aerial Images Using Vanishing Point Transformation and Deep Learning*. In 2018 25th IEEE International Conference on Image Processing, Athens, Greece, 1917-1921.
15. Li, Y., Ding, Z., Zhang, C., Wang, Y., and Chen, J. (2019). *SAR Ship Detection Based on Resnet and Transfer Learning*. In IGARSS 2019-2019 IEEE International Geoscience and Remote Sensing Symposium, Yokohama, Japan, 1188-1191.
16. Sundar, K. S., Bonta, L. R., Baruah, P. K., and Sankara, S. S. (2018). *Evaluating Training Time of Inception-v3 and Resnet-50,101 Models using TensorFlow across CPU and GPU*. In 2018 Second International Conference on Electronics, Communication and Aerospace Technology, Coimbatore, India, 1964-1968.
17. Internet: Krizhevsky, A., Nair, V., and Hinton, G. (2014). The cifar-10 dataset. URL: <https://www.cs.toronto.edu/~kriz/cifar.html>, Son Erişim Tarihi:15.03.2020
18. Song, H., Choi, I. K., Ko, M. S., Bae, J., Kwak, S., and Yoo, J. (2018). *Vulnerable pedestrian detection and tracking using deep learning*. In 2018 International Conference on Electronics, Information, and Communication, Honolulu, USA, 1-2.
19. Lin, T. Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., and Zitnick, C. L. (2014). *Microsoft coco: Common objects in context*. In European conference on computer vision, Cham, Switzerland, 740-755.
20. Tang, T., Deng, Z., Zhou, S., Lei, L., and Zou, H. (2017). *Fast vehicle detection in UAV images*. In 2017 International Workshop on Remote Sensing with Intelligent Processing, Shanghai, China, 1-5.
21. Kyrkou, C., Plastiras, G., Theocharides, T., Venieris, S. I., and Bouganis, C. S. (2018). *DroNet: Efficient convolutional neural network detector for real-time UAV applications*. In 2018 Design, Automation & Test in Europe Conference & Exhibition, Dresden, Germany, 967-972.
22. Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., and Berg, A. C. (2015). Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, 115(3), 211-252.
23. LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11), 2278-2324.

24. Çevik, K. K., ve Çakı, A. (2010). Görüntü İşleme Yöntemleriyle Araç Plakalarının Tanınarak Kapı Kontrolünün Gerçekleştirilmesi. *Afyon Kocatepe Üniversitesi Fen Ve Mühendislik Bilimleri Dergisi*, 10(1), 31-38.
25. Bayram, F. (2020). Derin Öğrenme Tabanlı Otomatik Plaka Tanıma. *Politeknik Dergisi. Baskıda*.
26. Yao, X., Yang, H., Wu, Y., Wu, P., Wang, B., Zhou, X., and Wang, S. (2019). Land Use Classification of the Deep Convolutional Neural Network Method Reducing the Loss of Spatial Features. *Sensors*, 19(12), 2792-2807.
27. Yavuz, S., ve Deveci, M. (2012). İstatiksel Normalizasyon Tekniklerinin Yapay Sinir Ağı Performansına Etkisi. *Erciyes Üniversitesi İktisadi ve İdari Bilimler Fakültesi Dergisi*, (40), 167-187.
28. İnternet: Ioffe, S., and Szegedy, C. (2015). Batch normalization: Accelerating deep network training by reducing internal covariate shift. URL: <https://arxiv.org/abs/1502.03167>, Son Erişim Tarihi: 24.03.2020
29. Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. (2014). Dropout: a simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, 15(1), 1929-1958.
30. Xiao, T., Li, H., Ouyang, W., and Wang, X. (2016). *Learning deep feature representations with domain guided dropout for person re-identification*. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, USA, 1249-1258.
31. LeCun, Y., Jackel, L. D., Bottou, L., Cortes, C., Denker, J. S., Drucker, H., and Vapnik, V. (1995). *Learning algorithms for classification: A comparison on handwritten digit recognition*, Neural Networks: the Statistical Mechanics Perspective, New Jersey, USA 261-276.
32. Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012). *Imagenet classification with deep convolutional neural networks*. In Advances in Neural Information Processing Systems, Nevada, USA, 1097-1105.
33. İnternet: Simonyan, K., and Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. URL: <https://arxiv.org/abs/1409.1556>, Son Erişim Tarihi: 24.03.2020
34. Chen, L. C., Papandreou, G., Kokkinos, I., Murphy, K., and Yuille, A. L. (2017). Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(4), 834-848.
35. Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., and Rabinovich, A. (2015). *Going deeper with convolutions*. In Proceedings of the IEEE conference on computer vision and pattern recognition, Boston, USA, 1-9.

36. He, K., Zhang, X., Ren, S., and Sun, J. (2016). *Deep residual learning for image recognition*. In Proceedings of the IEEE conference on computer vision and pattern recognition, Las Vegas, USA, 770-778.
37. Pan, S. J., and Yang, Q. (2009). A survey on transfer learning. *IEEE Transactions on knowledge and data engineering*, 22(10), 1345-1359.
38. Tan, C., Sun, F., Kong, T., Zhang, W., Yang, C., and Liu, C. (2018). *A survey on deep transfer learning*. In International conference on artificial neural networks, Rhodes, Greece, 270-279.
39. Sharif Razavian, A., Azizpour, H., Sullivan, J., and Carlsson, S. (2014). *CNN features off-the-shelf: an astounding baseline for recognition*. In Proceedings of the IEEE conference on computer vision and pattern recognition workshops, Columbus, USA 806-813.



ÖZGEÇMİŞ

Kişisel Bilgiler

Soyadı, adı : CENGİZ, Enes
 Uyuğu : T.C.
 Doğum tarihi ve yeri : 07.05.1993, Trabzon
 Medeni hali : Evli
 Telefon : 0 (272) 218 25 54
 Faks : 0 (272) 218 25 54
 e-mail : enescengiz@yandex.com



Eğitim

Derece	Eğitim Birimi	Mezuniyet Tarihi
Yüksek lisans	Gazi Üniversitesi / Elektrik Elektronik Mühendisliği	Devam ediyor
Lisans	Karadeniz Teknik Üniversitesi / Elektrik Elektronik Mühendisliği	2016
Lise	Maçka Mehmet Akif Ersoy Anadolu Lisesi	2011

İş Deneyimi

Yıl	Yer	Görev
2018-Halen	Afyon Kocatepe Üniversitesi	Araştırma Görevlisi
2016-2018	Astor Transformatör A.Ş.	Test Mühendisi

Yabancı Dil

İngilizce

Yayınlar

1. Cengiz, E., Yılmaz, C., Kahraman, H. T., and Bayram, F. (2019). *Pedestrian and Vehicles Detection with ResNet in Aerial Images*, 4th International Symposium on Innovative Approaches in Engineering and Natural Sciences, Samsun, Türkiye, 26-30.

2. Cengiz, E., Yılmaz, C., Yılmaz, E. N., and Kahraman, H. T. (2019). *Importance of Power Flow and Load Analysis in Pre-Installation Power Systems*, In 2019 3rd International Symposium on Multidisciplinary Studies and Innovative Technologies, Ankara, Türkiye, 1-4.

Hobiler

Futbol, Yüzme, Masa tenisi





GAZİ GELECEKTİR..